National Semiconductor

**PRELIMINARY**
April 2004
Revision 1.2

# PC87591L-N05
# LPC Mobile Embedded Controller

## General Description

The National Semiconductor PC87591L-N05 is a highly integrated embedded controller with an embedded RISC core and integrated advanced functions. This device is targeted for a wide range of portable applications that use the Low Pin Count (LPC) interface.

In this datasheet, references to the PC87591L-N05 include the PC87591L-**VPC**N05 and PC87591L-**SLC**N05.

The PC87591L-N05 incorporates the National CompactRISC™ CR16B core (a high-performance 16-bit RISC processor), on-chip ROM and RAM memories, system support functions and a Bus Interface Unit (BIU) that directly interfaces with external memory (such as flash) and I/O devices.

System support functions include: watchdog, PWM, timers, interrupt control, General-Purpose I/O (GPIO) with internal keyboard matrix scanning, PS/2® Interface, ACCESS.bus® interface and high-accuracy analog-to-digital (ADC) and digital-to-analog (DAC) converters for battery charging, system control, system health monitoring and analog controls.

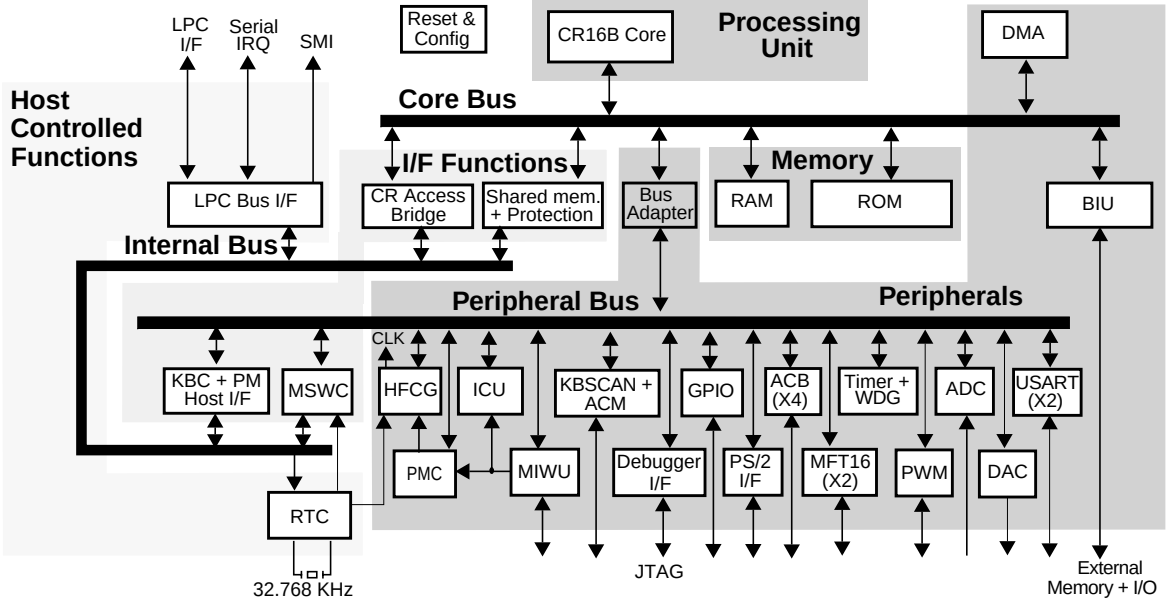
The PC87591L-N05 interfaces with the host via an LPC interface that provides the host with access to the following: Keyboard and embedded controller interface channels, integrated functions, Real-Time Clock (RTC) and BIOS firmware.

Like other members of the National SuperI/O family, the PC87591L-N05 is PC01 and ACPI compliant.

## Outstanding Features

- Host interface, based on Intel's *LPC Interface Specification Revision 1.1*, August 2002
- PC01 Rev 1.0, and ACPI 2.0 compliant
- 16-bit RISC core, with 2 Mbytes address space, running at up to 20 MHz
- JTAG-based debugger interface
- Shared BIOS flash memory (external)
- 92 GPIO ports (including keyboard scanning) with a variety of wake-up events
- Software- and hardware-controlled clock throttling, and extremely low current consumption in Idle mode
- 176-pin LQFP and FBGA packages

## Block Diagram



## Features

- Processing Unit
  - CompactRISC CR16B 16-bit embedded RISC processor core (the “core”)
  - Up to 2 Mbytes address space
- Internal Memory
  - 4 Kbytes of ROM
  - Boot block for CR16B
  - Memory contents protection
  - 4K of on-chip RAM
  - All memory types can hold both code and data
- Expansion Memory
  - Up to 2 Mbytes of code and data
  - Supports BIOS (flash) memory sharing with PC host
  - Boot block for host code
  - Hardware-protected boot zone with block protection
  - Supports external memory power-down mode
  - Field upgradable with flash or SRAM devices
  - Supports host-controlled code download and update
  - Bus Interface Unit (BIU)
    - Three address zones for static devices (SRAM, ROM, flash, I/O)
    - Configurable wait states and fast-read, single cycle bus cycles
    - 8- or 16-bit wide bus
- LPC System Interface
  - Synchronous cycles, up to 33 MHz bus clock
  - Serial IRQ
  - I/O and memory read and write cycles
  - Bootable memory support
  - Reset input
  - Base Address (BADDR) strap to determine the base address of the Index-Data register pair
  - $\overline{\text{LPCPD}}$ and $\overline{\text{CLKRUN}}$ support
  - FWH Transaction support
- Protection Function Support
  - Memory access protection

### Embedded Controller System Features

- Host Bus Interface (HBI)
  - Comprises three host interface channels, which are typically used for the KBC and ACPI Private or Shared EC channels
  - Includes one, 8042 KBC-standard, interface (legacy $60_{16}$, $64_{16}$)
  - Includes two PM interface ports (legacy $62_{16}$, $66_{16}$ and $68_{16}$, $6C_{16}$)
  - Provides ACPI Embedded Controller with either Shared or Private interface through the PM interface
  - Generates IRQ, SMI and SCI
  - Provides IRQ1 and IRQ12 support
  - Provides Fast Gate A20 and Fast Host Reset via firmware
- Interrupt Control Unit (ICU)
  - 31 maskable vectored interrupts (of which 26 are external)
  - General-purpose external interrupt inputs through MIWU
  - Enable and pending indication for each interrupt
  - Non-maskable interrupt input
- Multi-Input Wake-Up (MIWU)
  - Supports up to 32 wake-up or interrupt inputs
  - Generates wake-up event to PMC (Power Management Controller)
  - Generates interrupts to ICU
  - Provides user-selectable trigger conditions
- General-Purpose I/O (GPIO)
  - 92 port pins.
  - I/O pins individually configured as input or output
  - Configurable internal pull-up resistors
  - Special ports for internal keyboard matrix scanning
    - 16 open-collector outputs
    - Eight Schmitt inputs with internal pull-ups
  - Input for system On/Off switch
  - 17 external wake-up events
  - Low-cost external GPIO expansion through the BIU I/O Expansion protocol
- PS/2 Interface
  - Supports four external ports: Keyboard, mouse and two additional pointing devices
  - Supports byte-level handling via hardware accelerator
- Four ACCESS.bus (ACB) Interface modules. Each module:
  - Is Intel SMBus® and Philips $I^2C$® compatible
  - Is ACCESS.bus master and slave
  - Detects up to three simultaneous slave addresses
  - Supports polling and interrupt controlled operation
  - Generates a wake-up signal on detection of a Start Condition while in Idle mode
  - Has an optional internal pull-up on SDA and SCL pins
- Two Universal Synchronous/Asynchronous Receiver-Transmitter (USART) modules
  - A full-duplex USART channel
  - Programmable baud rate
  - Data transfer via interrupt or polling
  - Synchronous mode with either internal or external clock
  - 7-, 8- or 9-bit protocols.

## Features (Continued)

- Two 16-bit Multi Function Timer (MFT16) modules. Each module:
  - Contains two 16-bit timers
  - Supports Pulse Width Modulation (PWM), Capture and Counter
- Pulse Width Modulation (PWM) Module
  - Eight outputs
  - 8/16-bit duty cycle resolution
  - 8/16-bit common input clock prescaler
- Timer and Watchdog (TWM)
  - 16-bit periodic interrupt timer with 30 µs resolution and 5-bit prescaler for system tick and periodic wake-up tasks
  - 8-bit watchdog timer
- Analog to Digital Converter (ADC)
  - 14 channels (up to ten external and four internal), with 8-bit resolution
  - Sigma-delta technology for high noise rejection
  - Three voltage measurements every 100 ms
  - Internal voltage reference
- System Health Monitoring
  - Controlled by embedded controller
  - System Voltage Measurement
    - Up to ten external measurement points
    - Four internal measurement points
    - Smart power failure detection
- Digital to Analog Converter (DAC)
  - Four channels, 8-bit resolution
  - 1 µs conversion time for 50 pF load
  - Full output range from AGND to AVCC
- Analog Comparators Monitor (ACM)
  - Eight comparator inputs on KBD scan inputs
  - 6-bit input measurement resolution
  - Scan and Threshold modes
  - Supports low-current system wake-up
- Development Support Features
  - Interface to debugger via JTAG pins
    - ISE/ADB mode
    - On-board Debug mode
- CR16B Access to Host Controlled Functions
  - Enabled when host inactive

### Host Controlled Functions Features

- Supports *Microsoft® Advanced Power Management (APM) Specifications Revision 1.2*, February 1996
  - Generates the System Management Interrupt (SMI)
- PC01 and ACPI Compliant
  - PnP Configuration Register structure
  - Flexible resource allocation for all logical devices
    - Relocatable base address
    - 15 IRQ routing options
- Real-Time Clock (RTC)
  - DS1287 and MC146818 compatible
  - 242-byte battery backed-up CMOS RAM
  - Calendar including century and automatic leap-year adjustment (Y2K compliant)
  - Optional adjustment for daylight saving time
  - BCD or binary format for timekeeping
  - Three individually maskable interrupt event flags: periodic rates from 122 µs to 500 ms; time-of-day alarm, once-per-second to once-per-day
  - Double-buffer time registers
  - Alarm wake-up
- Mobile System Wake-Up Control (MSWC)
  - Wake-up on detection of $\overline{RI1}$, $\overline{RI2}$, $\overline{RING}$ activity
    - External modem ring on serial port
    - Ring pulse or pulse train on $\overline{RING}$ input signal
    - Software-controlled off events
  - Optional routing of power-up request on IRQ and/or SMI lines

### Clocking, Supply and Package Information

- Strap Inputs for operation control
  - ENV1-0 for IRE/OBD/DEV operating mode selection
  - SHBM for shared BIOS control
  - TRI-STATE® for ISE/ADB support
- Clocks
  - Single 32.768 KHz crystal oscillator
  - LPC clock, up to 33 MHz
  - On-chip high frequency clock generator
    - CPU clock 4-20 MHz
    - Software-controlled frequency generation
    - Multiplier source is the 32.768 KHz input
  - 32.768 KHz clock out
  - CR16B clock out
- Testability
  - XOR tree structure
    - Includes all device pins (except supply, analog and crystal oscillator pins)
    - Selected at power-up by strap input
  - TRI-STATE device pins, selected at power-up by strap input (TRIS)

## Features (Continued)

- Power Supply
  - 3.3V supply operation
  - 5V tolerance and back-drive protection on all pins (except LPC bus pins and keyboard scan inputs)
  - Separate supply for Host I/F ($V_{DD}$) and Embedded Controller functions ($V_{CC}$)
  - Separate pin for core voltage filtering ($V_{CORF}$)
  - Backup battery input for RTC, and wake-up configuration.
  - Reduced power consumption capability
  - Four power modes, switched by software or hardware
    - Active mode current (25 mA typ.)
    - Active mode executing WAIT (12 mA typ.)
    - Idle (10 µA typ.)
    - Power Off - for RTC and oscillator (0.9 µA typ.) from backup battery
  - Automatic wake-up on system events
- Package Options
  - 176-pin LQFP and FBGA packages

## Revision Record

| Revision Date | Status | Comments |
|---|---|---|
| March 16, 2003 | Revision 1.0 | Preliminary |
| July 17, 2003 | Revision 1.1 | Preliminary, second release |
| April 1, 2004 | Revision 1.2 | Preliminary, third release. List of changes:<br>- In the entire document:<br>- Removed PC97591L device<br>- Changed PC97591V to PC87591L-N05.<br>- In the entire document changed “security” to “protection”.<br>- Added clarification on XOR-Tree use.<br>- PTWRL, PTWRH and PNMR registers moved to “System Configuration Registers”.<br>- To PTWRL and PTWRH registers, added updating conditions (HOSTWAIT bit).<br>- In “Factory Parameters”, changed RevisionCode Interpretation.<br>- Updated the “Power Supply Current Consumption” for $I_{DD}$ and $I_{CC}$.<br>- Added “guaranteed by characterization” to $V_{BATDTC}$.<br>- Updated “Voltage Measurement” characteristics for: “Offset Error”, “Gain Error”, “Integral Non-linearity Error” and “Differential Non-linearity Error”.<br>- Added clarification for MFT16: “Slow Speed Clock” is “LFCLK”.<br>- Added clarification: “Input Clock” for TWD is “LFCLK”.<br>- Added specification for $V_{DD}$, $V_{CC}$ and $AV_{CC}$ Power Off Voltage ($V_{OFF}$).<br>- Separated $\overline{RI}$ timing from $\overline{RING}$ timing; added new values ($t_{LR}$ and $t_{HR}$) for $\overline{RING}$.<br>- Updated USART timing.<br>- Changed the “Header 2”, “Reserved” area at offset 21 to 3 bytes.<br>- Replaced TBD in the current consumption in the “Power Supply” features.<br>- Replaced TBD in the input capacitance of “Voltage Inputs”, “Temperature Inputs” and “ACM Inputs”.<br>- Replaced TBD in the “Package Thermal Information” for 176-Ball FBGA.<br>- In GPIO pin description, corrected pin numbers of IOPJ7-2.<br>- In the “Register List”, corrected the layout of PEWPU register.<br>- In the SZCFGn register, added clarification to WAIT, HOLD and BRE bits.<br>- In the “Register List”, removed duplicate TnCNT2 register.<br>- Corrected the reset value of DCRi (i = 0-7) registers.<br>- Added clarification for ACB: ACBnCTL1 register is cleared in Idle mode.<br>- Added clarification to CFGAE bit in CRSMAE register.<br>- Corrected type (WO) of WK_STATE register.<br>- Corrected the conditions for RTC oscillator active/disabled.<br>- Corrected the value of $t_{LW}$ in “Debugger Interface Timing”.<br>- In “LPC Signals Timing”, added minimum value for $t_{VAL}$ and removed $\overline{LPCPD}$, $\overline{RESET1-2}$ from “Inputs” timing diagram.<br>- Corrected the description of the calculation algorithm for “Checksum” and “XOR Checksum” in “Header 2” of the Booter program.<br>- Corrected the Booter memory resources limitations. |

# Table of Contents

**Features** ..... 2
Embedded Controller System Features ..... 2
Host Controlled Functions Features ..... 3
Clocking, Supply and Package Information ..... 3

**Revision Record** ..... 5

## 1.0 Introduction

1.1 DOCUMENT ORGANIZATION ..... 23
1.2 GENERAL DESCRIPTION ..... 23
1.2.1 System Connections ..... 23
1.2.2 Power Management ..... 23
1.2.3 Operating Environments ..... 25
1.3 INTERNAL ARCHITECTURE ..... 25
1.3.1 Processing Unit ..... 26
1.3.2 Bus Interface Unit and Memory Controller (BIU) ..... 26
1.3.3 Memory ..... 26
1.3.4 Peripherals ..... 26
1.3.5 Host-Controller Interface Modules ..... 27
1.3.6 Host-Controlled SuperI/O Modules and Host Interface ..... 28
1.4 OPERATING ENVIRONMENTS ..... 28
1.4.1 IRE Environment ..... 28
1.4.2 OBD Environment ..... 29
1.4.3 DEV Environment ..... 29
1.5 MEMORY MAP ..... 29
1.5.1 Core Address Domain Memory Map ..... 29
Register Abbreviations and Access ..... 32
Accessing Base Memory ..... 33
Accessing Expansion Memory ..... 33
Accessing I/O Expansion Space ..... 34
1.5.2 Host Address Domain Memory Map ..... 34
1.5.3 Core Access to Host Controlled Peripherals ..... 35

## 2.0 Signal/Pin Description and Configuration

2.1 CONNECTION DIAGRAMS ..... 36
2.2 BUFFER TYPES AND SIGNAL/PIN DIRECTORY ..... 38
2.2.1 ACCESS.bus Interface ..... 38
2.2.2 Analog Interface ..... 39
2.2.3 Clocks ..... 39
2.2.4 Core Bus Interface Unit (BIU) ..... 39
2.2.5 Development System Support ..... 40
2.2.6 General-Purpose I/O (GPIO) and Internal Keyboard Scan ..... 41
2.2.7 Host Interface ..... 42
2.2.8 Interrupt and Wake-Up Inputs (ICU and MIWU) ..... 44
2.2.9 Power and Ground ..... 44
2.2.10 PS/2 Interface ..... 45
2.2.11 Strap Configuration and Testing ..... 45

## Table of Contents (Continued)


2.2.12 Mobile System Wake-Up Control (MSWC) ..... 46
2.2.13 Timers and PWM ..... 46
2.2.14 Universal Synchronous/Asynchronous Receiver/Transmitter (USART) ..... 46
2.2.15 Internal Pull-Up and Pull-Down Resistors ..... 47
2.3 STRAP PINS ..... 48
Setting the Environment ..... 48
Other Strap Pin Settings ..... 48
System Load on Strap Pins ..... 48
Strap Pin Status Register (STRPST) ..... 48
2.4 ALTERNATE FUNCTIONS ..... 49
2.4.1 GPIO with Alternate Functions ..... 49
2.4.2 System Configuration Registers ..... 52
Register Map ..... 52
Module Configuration Register (MCFG) ..... 52
External Interrupts Configuration Register (EICFG) ..... 53
Protection Word Low Register (PTWRL) ..... 54
Protection Word High Register (PTWRH) ..... 55
Pin Multiplexing Register (PNMR) ..... 55
2.4.3 GPIO with Echo Configuration ..... 56
Input to Output Echo Enable Register 1 and 2 (IOEE1 and IOEE2) ..... 56

**3.0 Power, Reset and Clocks**

3.1 POWER ..... 58
3.1.1 Power Planes ..... 58
3.1.2 Power States ..... 58
3.1.3 Power Connection and Layout Guidelines ..... 59
3.2 RESET SOURCES AND TYPES ..... 61
3.2.1 $V_{PP}$ Power-Up Reset ..... 62
3.2.2 $V_{CC}$ Power-Up Reset ..... 62
3.2.3 Watchdog Reset and Debugger Interface Reset ..... 62
3.2.4 Warm Reset ..... 63
3.2.5 Host Domain Reset ..... 63
Using RESET2 Input ..... 63
Host Domain Reset Actions ..... 63
3.3 CLOCK DOMAINS ..... 64
Core Domain Clock ..... 64
LPC Clock ..... 64
RTC (32 KHz) Clock ..... 64
3.4 TESTABILITY SUPPORT ..... 64
3.4.1 ICT ..... 64
3.4.2 XOR-Tree Testing ..... 64

**4.0 Embedded Controller Modules**

4.1 BUS INTERFACE UNIT (BIU) ..... 67
4.1.1 Features ..... 67
4.1.2 Functional Description ..... 67
Interface ..... 67

## Table of Contents (Continued)


Static Memory and I/O Support ..... 67
Byte Access ..... 67
Clock and Bus Cycles ..... 68
4.1.3 Clock Cycles ..... 69
Optional Clock Cycles ..... 69
Other Clock Cycles ..... 69
Control Signals ..... 70
4.1.4 Early Write Bus Cycle ..... 70
4.1.5 Late Write Bus Cycle ..... 73
4.1.6 Normal Read Bus Cycle ..... 74
4.1.7 Fast Read Bus Cycle ..... 78
4.1.8 I/O Expansion Bus Cycles ..... 78
4.1.9 Development Support ..... 80
Bus Status Signals ..... 80
Core Bus Monitoring ..... 80
4.1.10 BIU Registers ..... 81
BIU Configuration Register (BCFG) ..... 81
I/O Zone Configuration Register (IOCFG) ..... 82
Static Zone Configuration Register (SZCFGn) ..... 83
4.1.11 Usage Hints ..... 84
4.2 DMA CONTROLLER (DMAC) ..... 85
4.2.1 Features ..... 85
4.2.2 Functional Description ..... 85
4.2.3 Channel Assignment in PC87591L-N05 ..... 86
4.2.4 Transfer Types ..... 86
Direct (Fly-By) Transfers ..... 86
Indirect (Memory-to-Memory) Transfers ..... 87
4.2.5 Bus Policy ..... 88
Intermittent Operation Mode ..... 88
Continuous Operation Mode ..... 88
4.2.6 Operation Modes ..... 89
Single Transfer Operation ..... 89
Double Buffer Operation ..... 89
Auto-Initialize Operation ..... 90
4.2.7 Software DMA Request ..... 90
4.2.8 DMAC Registers ..... 90
DMAC Register Map ..... 90
Device A Address Counter Register (ADCAn) ..... 91
Device A Address Register (ADRAn) ..... 91
Device B Address Counter Register (ADCBn) ..... 91
Device B Address Register (ADRBn) ..... 91
Block Length Counter Register (BLTCn) ..... 92
Block Length Register (BLTRn) ..... 92
DMA Control Register (DMACNTLn) ..... 92
DMA Status Register (DMASTATn) ..... 94
4.2.9 Usage Hints ..... 95
4.3 INTERRUPT CONTROL UNIT (ICU) ..... 96
4.3.1 Features ..... 96

## Table of Contents (Continued)


4.3.2 Non-Maskable Interrupt (NMI) ..... 96
External NMI Inputs ..... 96
Non-Maskable Interrupt Processing ..... 96
$\overline{\text{PFAIL}}$ Input ..... 96
4.3.3 Maskable Interrupts ..... 96
Maskable Interrupt Vectors ..... 97
Pending Interrupts ..... 97
Interrupt Priorities ..... 97
Power-Down Modes ..... 97
External Interrupt Inputs ..... 97
Interrupt Assignment ..... 97
4.3.4 ICU Registers ..... 99
ICU Register Map ..... 99
Interrupt Vector Register (IVCT) ..... 99
NMI Status Register (NMISTAT) ..... 99
Power Fail Interrupt Control and Status Register (PFAIL) ..... 100
Interrupt Status Register 0 (ISTAT0) ..... 100
Interrupt Status Register 1 (ISTAT1) ..... 101
Interrupt Enable and Mask Register 0 (IENAM0) ..... 101
Interrupt Enable and Mask Register 1 (IENAM1) ..... 101
Edge Interrupt Clear Register 0 (IECLR0) ..... 102
Edge Interrupt Clear Register 1 (IECLR1) ..... 102
4.3.5 Usage Hints ..... 102
Initializing ..... 102
Clearing ..... 102
Nesting ..... 102
4.4 MULTI-INPUT WAKE-UP (MIWU) ..... 103
4.4.1 Features ..... 103
4.4.2 Operation ..... 103
4.4.3 MIWU Registers ..... 106
MIWU Register Map ..... 106
Edge Detection Register (WKEDG1) ..... 107
Edge Detection Register (WKEDG2) ..... 107
Edge Detection Register (WKEDG3) ..... 107
Edge Detection Register (WKEDG4) ..... 107
Pending Register (WKPND1) ..... 107
Pending Register (WKPND2) ..... 108
Pending Register (WKPND3) ..... 108
Pending Register (WKPND4) ..... 108
Enable Register (WKEN1) ..... 108
Enable Register (WKEN2) ..... 108
Enable Register (WKEN3) ..... 108
Enable Register (WKEN4) ..... 108
Pending Clear Register (WKPCL1) ..... 109
Pending Clear Register (WKPCL2) ..... 109
Pending Clear Register (WKPCL3) ..... 109
Pending Clear Register (WKPCL4) ..... 109
4.4.4 Usage Hints ..... 109
4.5 GENERAL-PURPOSE I/O (GPIO) PORTS ..... 110
GPIO Port Functionality ..... 110
4.5.1 Features ..... 110

## Table of Contents (Continued)


4.5.2 GPIO Port Px ..... 111
Bidirectional Port with Alternate Function ..... 111
4.5.3 GPI Port Py ..... 113
Input Only Port with Alternate Function ..... 113
4.5.4 GPO Port Pz ..... 114
Output Only Port with Alternate Function ..... 114
4.5.5 GPIO Port Pw ..... 115
GPIO Port Signals Shared with Development System Signals ..... 115
4.5.6 GPIO Port Registers ..... 116
GPIO Register Map ..... 116
Port Alternate Function Registers (PxALT, PyALT and PzALT) ..... 116
Port Direction Registers (PxDIR and PwDIR) ..... 117
Port Data Output Register (PxDOUT, PzDOUT and PwDOUT) ..... 117
Port Data Input Registers (PxDIN, PyDIN and PwDIN) ..... 117
Port Weak Pull-Up Registers (PxWPU, PyWPU) ..... 118

4.6 PS/2 INTERFACE ..... 119
4.6.1 Features ..... 119
4.6.2 General Description ..... 119
4.6.3 Operating With the Shift Mechanism Disabled ..... 121
4.6.4 Operating With the Shift Mechanism Enabled ..... 121
Receive Mode ..... 123
Transmit Mode ..... 124
4.6.5 PS/2 Interface Registers ..... 125
PS/2 Register Map ..... 125
PS/2 Data Register (PSDAT) ..... 126
PS/2 Status Register (PSTAT) ..... 126
PS/2 Control Register (PSCON) ..... 127
PS/2 Output Signal Register (PSOSIG) ..... 128
PS/2 Input Signal Register (PSISIG) ..... 128
PS/2 Interrupt Enable Register (PSIEN) ..... 129

4.7 MULTI-FUNCTION 16-BIT TIMER (MFT16) ..... 130
4.7.1 Features ..... 130
4.7.2 Clock Source Unit ..... 131
Pre-Scaler ..... 131
External Event Clock ..... 131
Pulse Accumulate Mode ..... 131
Slow-Speed Clock (LFCLK) ..... 131
Counter Clock Source Select ..... 132
4.7.3 Timer/Counter and Action Unit ..... 132
Operation Modes ..... 132
Mode 1, PWM and Counter ..... 132
Mode 2, Dual Input Capture ..... 133
Mode 3, Dual Independent Timer ..... 134
Mode 4, Input Capture and Timer ..... 135
4.7.4 Timer Interrupts ..... 136
4.7.5 Timer I/O Functions ..... 137
4.7.6 Operation in Development System ..... 137
4.7.7 MFT16 Registers ..... 137
MFT16 Register Map ..... 137

## Table of Contents (Continued)


Timer/Counter Register 1 (TnCNT1) ..... 138
Reload/Capture Register A (TnCRA) ..... 138
Reload/Capture Register B (TnCRB) ..... 138
Timer/Counter Register 2 (TnCNT2) ..... 138
Clock Pre-Scaler Register (TnPRSC) ..... 139
Clock Unit Control Register (TnCKC) ..... 139
Timer Mode Control Register (TnCTRL) ..... 140
Timer Interrupt Control Register (TnICTL) ..... 141
Timer Interrupt Clear Register (TnICLR) ..... 142

4.8 PULSE WIDTH MODULATOR (PWM) ..... 143
4.8.1 Features ..... 143
4.8.2 Functional Description ..... 143
4.8.3 Cycle Time and Duty Cycle Calculation ..... 144
4.8.4 Power Modes ..... 144
4.8.5 PWM Registers ..... 144
PWM Register Map ..... 144
Clock Pre-Scaler Register (PRSC) ..... 144
Cycle Time Register (CTR) ..... 145
Duty Cycle Registers 0 to 7 (DCRi) ..... 145
PWM Polarity Register (PWMPOL) ..... 146
PWM Control Register (PWMCNT) ..... 146

4.9 UNIVERSAL SYNCHRONOUS/ASYNCHRONOUS RECEIVER-TRANSMITTER (USART) .. 147
4.9.1 Features ..... 147
4.9.2 Functional Overview ..... 147
4.9.3 Operation ..... 148
4.9.4 USART Registers ..... 154
USART Register Map ..... 154
Receive Data Buffer Register (UnRBUF) ..... 154
Transmit Data Buffer Register (UnTBUF) ..... 154
Baud Rate Pre-Scaler Register (UnPSR) ..... 154
Baud Rate Divisor Register (UnBAUD) ..... 155
Frame Select Register (UnFRS) ..... 155
Mode Select Register (UnMDSL) ..... 156
Status Register (UnSTAT) ..... 157
Interrupt Control Register (UnICTRL) ..... 158
4.9.5 Usage Hints ..... 159

4.10 TIMER AND WATCHDOG (TWD) ..... 160
4.10.1 Features ..... 160
4.10.2 Functional Description ..... 161
4.10.3 TWD Registers ..... 162
TWD Register Map ..... 162
Timer and Watchdog Configuration Register (TWCFG) ..... 162
Timer and Watchdog Clock Pre-Scaler Register (TWCP) ..... 163
TWD Timer 0 Register (TWDT0) ..... 164
TWDT0 Control and Status Register (T0CSR) ..... 164
Watchdog Count Register (WDCNT) ..... 165
Watchdog Service Data Match Register (WDSDM) ..... 165
4.10.4 Usage Hints ..... 165

4.11 ANALOG TO DIGITAL CONVERTER (ADC) ..... 166
4.11.1 Features ..... 166

## Table of Contents (Continued)


4.11.2 Functional Description ..... 166
4.11.3 Voltage Measurement ..... 167
4.11.4 ADC Operation ..... 168
Reset ..... 168
ADC Clock ..... 168
Initializing the ADC ..... 168
Enabling and Disabling the ADC ..... 169
Interrupt Structure ..... 169
ADC Operating Principles ..... 169
Reading Measurement Results ..... 170
Failure Detection ..... 170
4.11.5 ADC Registers ..... 171
ADC Register Map ..... 171
ADC Status Register (ADCSTS) ..... 171
ADC Configuration Register (ADCCNF) ..... 172
ADC Clock Control Register (ACLKCTL) ..... 172
ADC Delay Control Register (ADLYCTL) ..... 172
ADC Parameters Index Register (ADCPINX) ..... 173
ADC Parameters Data Register (ADCPD) ..... 173
Voltage Channel 1 Control Register (VCHN1CTL) ..... 174
Voltage Channel 1 Data Buffer (VCHN1DAT) ..... 174
Voltage Channel 2 Control Register (VCHN2CTL) ..... 175
Voltage Channel 2 Data Buffer (VCHN2DAT) ..... 175
Voltage Channel 3 Control Register (VCHN3CTL) ..... 176
Voltage Channel 3 Data Buffer (VCHN3DAT) ..... 176
4.11.6 Usage Hints ..... 176
Power Supply and Layout Guidelines ..... 176
Power Consumption ..... 176
Back-Drive Protection ..... 176
Measuring Out of Range Voltages ..... 176
Filtering the Noise on Voltage Input Signals ..... 177
Calculating the Voltage Channel Delay ..... 177
Thermistor-Based Temperature Measurement ..... 177
4.12 DIGITAL TO ANALOG CONVERTER (DAC) ..... 179
4.12.1 Features ..... 179
4.12.2 Functional Description ..... 179
4.12.3 D/A Conversion ..... 179
Output Signal ..... 179
Reference Voltage ..... 180
Conversion Time ..... 180
4.12.4 Operation ..... 181
Initializing the DAC ..... 181
Enabling and Disabling the DAC ..... 181
4.12.5 DAC Registers ..... 181
DAC Register Map ..... 181
DAC Control Register (DACCTRL) ..... 182
DAC Data Channel 0-3 Registers (DACDAT0-3) ..... 182
4.12.6 Usage Hints ..... 183
Power Consumption ..... 183
DAC Output Protection ..... 183
Output Voltage Accuracy ..... 183

## Table of Contents (Continued)


Output Settling Time ..... 183
Filtering Noise on Output Signals ..... 183

4.13 ACCESS.BUS (ACB) INTERFACE ..... 184
4.13.1 Features ..... 184
4.13.2 Functional Description ..... 184
"Acknowledge After Every Byte" Rule ..... 185
4.13.3 Master Mode ..... 186
4.13.4 Slave Mode ..... 188
4.13.5 Power-Down ..... 189
4.13.6 SDA and SCL Pin Configuration ..... 189
4.13.7 ACB Clock Frequency Configuration ..... 189
4.13.8 ACB Registers ..... 190
ACB Register Map ..... 190
ACB Serial Data Register (ACBnSDA) ..... 190
ACB Status Register (ACBnST) ..... 191
ACB Control Status Register (ACBnCST) ..... 192
ACB Control Register 1 (ACBnCTL1) ..... 193
ACB Own Address Register (ACBnADDR and ACBnADDR2) ..... 194
ACB Control Register 2 (ACBnCTL2) ..... 195
ACB Control Register 3 (ACBnCTL3) ..... 195
4.13.9 Usage Hints ..... 196

4.14 ANALOG COMPARATORS MONITOR (ACM) ..... 197
4.14.1 Features ..... 197
4.14.2 Functional Description ..... 197
4.14.3 Voltage Level ..... 198
4.14.4 ACM Operation ..... 199
Reset ..... 199
Sampling Delay ..... 199
Initializing the ACM ..... 199
Interrupt Structure ..... 200
ACM Operating Sequences ..... 200
4.14.5 ACM Registers ..... 201
ACM Register Map ..... 201
ACM Control and Status Register (ACMCTS) ..... 202
ACM Configuration Register (ACMCNF) ..... 203
ACM Timing Control Register (ACMTIM) ..... 204
Comparison Threshold Data Register (THRDAT) ..... 205
Comparison Result Register (CMPRES) ..... 205
Voltage Level Data Buffer - Input 0 through 7 (VOLDAT0-7) ..... 206
4.14.6 Usage Hints ..... 206
Voltage Level Burst ..... 206

4.15 ON-CHIP RAM ..... 207

4.16 ON-CHIP ROM ..... 207

4.17 POWER MANAGEMENT CONTROLLER (PMC) ..... 208
4.17.1 Features ..... 208
4.17.2 The Core Domain Power Modes ..... 208
4.17.3 Switching Between Power Modes ..... 209
Decreasing Power Consumption ..... 209

## Table of Contents (Continued)


Increasing Activity ..... 209
Power Mode Switch Protection ..... 210
4.17.4 The Power Management Controller Status Register (PMCSR) ..... 210
4.17.5 Usage Hints ..... 211
4.18 HIGH-FREQUENCY CLOCK GENERATOR (HFCG) ..... 212
4.18.1 Features ..... 212
4.18.2 Functional Description ..... 212
4.18.3 HFCG States ..... 213
PMC Enabled SuperI/O Disabled State ..... 213
SuperI/O Enabled State ..... 215
4.18.4 The Programmable Pre-Scaler: Core Domain Clock Generation ..... 215
4.18.5 State Transitions ..... 215
4.18.6 48 MHz Clock Monitor ..... 216
4.18.7 HFCG Registers ..... 216
HFCG Register Map ..... 216
HFCG Control Register 1 (HFCGCTRL1) ..... 217
HFCGM Low Value Register (HFCGML) ..... 218
HFCGM High Value Register (HFCGMH) ..... 218
HFCGN Value Register (HFCGN) ..... 218
HFCGI Low Value Register (HFCGIL) ..... 219
HFCGI High Value Register (HFCGIH) ..... 219
HFCG Pre-Scaler Register (HFCGP) ..... 219
HFCG Control Register 2 (HFCGCTRL2) ..... 220
4.19 THE DEBUGGER INTERFACE ..... 221
4.19.1 Features ..... 221
4.19.2 Structure ..... 221
4.19.3 Debugger Interface Functional Description ..... 222
Rx Data Link ..... 222
Tx Data Link ..... 222
Debugger Reset Circuit ..... 223
ISE Interrupt Control ..... 223
Clock Synchronization ..... 224
4.19.4 Test Access Port (TAP) ..... 224
TAP Signals ..... 225
TAP Controller ..... 225
4.19.5 TAP Instruction Register ..... 227
Design and Construction ..... 227
Instructions ..... 228
Debugger Interface Instructions ..... 228
4.19.6 TAP Data Registers, Debugger Interface ..... 229
Bit Arrangement and Mapping ..... 229
Functionality in Various TAP Controller States ..... 229
Debug Bypass Register (BYPASS) ..... 230
Debug Data Register (DBGDATA) ..... 230
Debug Abort Mask Register (DBGMASKS) ..... 230
4.19.7 Core Registers, Debugger Interface ..... 231
Core Register Map ..... 231
Debug Receive Data Registers 0, 2, 4, 6, 8, 10, 12 and 14 (DBGRXD0-14) ..... 231
Debug Receive Status Register (DBGRXST) ..... 232

## Table of Contents (Continued)


Debug Transmit Data Registers 0, 2, 4, 6, 8, 10, 12 and 14 (DBGTXD0-14) ..... 232
Debug Transmit Lock Register (DBGTXLOC) ..... 233
Debug Transmit Status Register (DBGTXST) ..... 233
Debug TINT Assert Register (DBGTINT) ..... 233
Debug Abort Generate Register (DBGABORT) ..... 234
Debug ISE Source Registers A (DBGISESRCA) ..... 234

4.19.8 Usage Hints ..... 235

4.20 DEVELOPMENT SYSTEM SUPPORT ..... 236

4.20.1 Features ..... 236

4.20.2 The ISE Interrupt ..... 236

4.20.3 Break Line and Reset Output Interrupt ..... 236

4.20.4 TRIS Strap Input Pin ..... 236

4.20.5 Freezing Events ..... 237

4.20.6 Monitoring Activity During Development ..... 237

Bus Status Signals ..... 237
Transaction Effect on the External Bus ..... 238
Pipe Status Signals (PFS and PLI) ..... 238

4.20.7 On-Chip Hardware Breakpoint ..... 238

4.20.8 CR16B Development Support Registers ..... 239

Debug Configuration Register (DBGCFG) ..... 239
Debug Freeze Enable Register (DBGFRZEN) ..... 240
Debug Freeze Enable Register2 (DBGFRZEN2) ..... 241

**5.0 Host Controller Interface Modules**

5.1 KEYBOARD AND MOUSE CONTROLLER INTERFACE ..... 242

5.1.1 Features ..... 242

5.1.2 General Description ..... 242

Host Addresses ..... 242
Core Interrupts ..... 242
Host Interrupts ..... 242
Keyboard/Mouse Channel (6016, 6416) ..... 243
Status Read ..... 244

5.1.3 Host Interface Registers ..... 245

Host Interface Register Map ..... 245
Data Out Buffer Register (DBBOUT, Legacy 6016) ..... 245
Status Register (STATUS, Legacy 6416) ..... 246
Data In Buffer Register (DBBIN, Legacy 6016) ..... 246
Command In Buffer Register (COMAND, Legacy 6416) ..... 246

5.1.4 Core Interface Registers ..... 247

Core Interface Register Map ..... 247
Host Interface Control Register (HICTRL) ..... 247
Host Interface IRQ Control Register (HIIRQC) ..... 248
Host Interface Keyboard/Mouse Status Register (HIKMST) ..... 249
Host Interface Keyboard Data Out Buffer Register (HIKDO) ..... 249
Host Interface Mouse Data Out Buffer Register (HIMDO) ..... 250
Host Interface Keyboard/Mouse Data In Buffer Register (HIKMDI) ..... 250

5.2 POWER MANAGEMENT (PM) CHANNELS ..... 251

5.2.1 Features ..... 251

5.2.2 General Description ..... 251

## Table of Contents (Continued)


Host Addresses ..... 252
Core Interrupts ..... 252
Host Interrupt Generation Modes ..... 253
Status Read ..... 255
Host Data Read from Host Interface Power Management Channel ..... 255

5.2.3 Core PM Registers ..... 256

Core PM Register Map ..... 256
Host Interface PM n Status Register (HIPMnST) ..... 256
Host Interface PM n Data Out Buffer (HIPMnDO) ..... 257
Host Interface PM n Data Out Buffer with SCI (HIPMnDOC) ..... 257
Host Interface PM n Data Out Buffer with SMI (HIPMnDOM) ..... 257
Host Interface PM n Data In Buffer (HIPMnDI) ..... 258
Host Interface PM n Data In Buffer with SCI (HIPMnDIC) ..... 258
Host Interface PM n Control Register (HIPMnCTL) ..... 259
Host Interface PM n Interrupt Control Register (HIPMnIC) ..... 260
Host Interface PM n Interrupt Enable Register (HIPMnIE) ..... 261

5.3 SHARED MEMORY AND PROTECTION ..... 262

5.3.1 Host Bus to Core Bus Access Translation ..... 262
5.3.2 Memory Mapping and Host Address Translation ..... 262
5.3.3 Indirect Memory Read and Write Transaction ..... 265
5.3.4 Locking Between Domains ..... 265
5.3.5 Host Access Protection ..... 266

Response to a Restricted Access ..... 267

5.3.6 Signaling Interface ..... 268
5.3.7 Shared Memory Host Registers ..... 268

Shared Memory Indirect Memory Address Register 0 (SMIMA0) ..... 268
Shared Memory Indirect Memory Address Register 1 (SMIMA1) ..... 269
Shared Memory Indirect Memory Address Register 2 (SMIMA2) ..... 269
Shared Memory Indirect Memory Address Register 3 (SMIMA3) ..... 269
Shared Memory Indirect Memory Data Register (SMIMD) ..... 269
Shared Memory Host Access Protect Register 1 and 2 (SMHAP1-2) ..... 270
Shared Memory Host Semaphore Register (SMHSEM) ..... 270

5.3.8 Shared Memory Core Registers ..... 271

Shared Memory Core Control and Status Register (SMCCST) ..... 271
Shared Memory Core Top Address Register (SMCTA) ..... 272
Shared Memory Host Semaphore Register (SMHSEM) ..... 272
Shared Memory Core Override Read Protect Registers 0-2 (SMCORP0-2) ..... 272
Shared Memory Core Override Write Protect Registers 0-2 (SMCOWP0-2) ..... 273

5.3.9 Usage Hints ..... 274

5.4 CORE ACCESS TO HOST-CONTROLLED MODULES ..... 275

5.4.1 Core Access to Host-Controlled Module Registers ..... 276

Indirect Host I/O Address Register (IHIOA) ..... 276
Indirect Host Data Register (IHD) ..... 276
Lock SuperI/O Host Access Register (LKSIOHA) ..... 277
SuperI/O Access Lock Violation Register (SIOLV) ..... 277
Core to SIB Modules Access Enable Register (CRSMAE) ..... 278
SIB Control Register (SIBCTRL) ..... 279

5.5 MOBILE SYSTEM WAKE-UP CONTROL (MSWC) ..... 280

5.5.1 Features ..... 280
5.5.2 Wake-Up Event Detection and Status Bits ..... 280

## Table of Contents (Continued)


5.5.3 Wake-Up Output Events ..... 283
5.5.4 Other MSWC Controlled Elements ..... 284
Host Configuration Address Selection ..... 284
Host Keyboard Fast Reset ..... 284
GA20 Pin Functionality ..... 285
5.5.5 MSWC Host Registers ..... 285
MSWC Host Register Map ..... 285
Wake-Up Event Status Register 0 (WK_STS0) ..... 286
Wake-Up Events Enable Register (WK_EN0) ..... 286
Wake-Up Configuration Register (WK_CFG) ..... 287
Wake-Up Signals Value Register (WK_SIGV) ..... 288
Wake-Up ACPI State Register (WK_STATE) ..... 288
Wake-Up Event Routing to SMI Enable Register 0 (WK_SMIEN0) ..... 289
Wake-Up Event Routing to IRQ Enable Register 0 (WK_IRQEN0) ..... 290
5.5.6 MSWC Core Registers ..... 291
MSWC Control Status Register 1 (MSWCTL1) ..... 291
MSWC Control Status Register 2 (MSWCTL2) ..... 292
MSWC Control Status Register 3 (MSWCTL3) ..... 293
Host Configuration Base Address Low (HCFGBAL) ..... 293
Host Configuration Base Address High (HCFGBAH) ..... 293
MSWC Interrupt Enable Register 2 (MSIEN2) ..... 294
MSWC Host Event Status Register 0 (MSHES0) ..... 294
MSWC Host Event Interrupt Enable Register (MSHEIE0) ..... 295
5.5.7 Usage Hints ..... 296
PWUREQ Output Connection ..... 296
RESET2 Events ..... 296

**6.0 Host-Controlled Modules and Host Interface**

6.1 DEVICE ARCHITECTURE AND CONFIGURATION ..... 297
6.1.1 Configuration Structure and Access ..... 297
The Index-Data Register Pair ..... 297
Banked Logical Device Registers Structure ..... 298
Standard Logical Device Configuration Register Definitions ..... 299
6.1.2 Standard Configuration Registers ..... 302
SuperI/O Control and Configuration Registers ..... 302
Logical Device Control and Configuration Registers ..... 302
Control ..... 302
Standard Configuration ..... 303
Special Configuration ..... 303
6.1.3 Default Configuration Setup ..... 303
6.1.4 Address Decoding ..... 303
6.1.5 Interrupt Serializer ..... 304
6.1.6 Protection ..... 304
6.1.7 LPC Interface ..... 304
LPC Transactions Supported ..... 304
Core Interrupt ..... 305
CLKRUN Functionality ..... 305
LPCPD Functionality ..... 305
6.1.8 SuperI/O Configuration Registers ..... 306
SuperI/O ID Register (SID) ..... 306

## Table of Contents (Continued)


SuperI/O Configuration 1 Register (SIOCF1) ..... 306
SuperI/O Configuration 5 Register (SIOCF5) ..... 307
SuperI/O Configuration 6 Register (SIOCF6) ..... 308
SuperI/O Revision ID Register (SRID) ..... 308
SuperI/O Configuration 8 Register (SIOCF8) ..... 309
SuperI/O Configuration 9 Register (SIOCF9) ..... 309
SuperI/O Configuration D Register (SIOCFD) ..... 309

6.1.9 Mobile System Wake-Up Control (MSWC) Configuration ..... 310
Logical Device 4 (MSWC) Configuration ..... 310

6.1.10 Keyboard and Mouse Controller (KBC) Configuration ..... 310
Logical Devices 5 and 6 (Mouse and Keyboard) Configuration ..... 310

6.1.11 Shared Memory Configuration ..... 311
Logical Device 15 ($0F_{16}$) (Shared Memory) Configuration ..... 311
Memory Range Programing ..... 311
Shared Memory Configuration Register ..... 312
Shared Memory Base Address High Byte Register ..... 313
Shared Memory Base Address Low Byte Register ..... 313
Shared Memory Size Configuration Register ..... 314

6.1.12 Real Time Clock (RTC) Configuration ..... 315
Logical Device 16 ($10_{16}$) RTC Configuration ..... 315
RAM Lock Register (RLR) ..... 315
Date Of Month Alarm Register Offset (DOMAO) ..... 316
Month Alarm Register Offset (MONAO) ..... 316
Century Register Offset (CENO) ..... 316

6.1.13 Power Management Interface Channel 1 Configuration ..... 317
Logical Device 17 ($11_{16}$) Power Management Channel 1 ..... 317

6.1.14 Power Management Interface Channel 2 Configuration ..... 317
Logical Device 18 ($12_{16}$) Power Management Channel 2 ..... 317

6.2 REAL-TIME CLOCK (RTC) ..... 318
6.2.1 Bus Interface ..... 318
6.2.2 RTC Clock Generation ..... 318
6.2.3 Internal Oscillator ..... 318
6.2.4 External Oscillator ..... 319
Connections ..... 320
Signal Parameters ..... 320
6.2.5 Timing Generation ..... 320
6.2.6 Timekeeping ..... 320
6.2.7 Updating ..... 321
6.2.8 Alarms ..... 321
6.2.9 Power Supply ..... 322
6.2.10 System Bus Lockout ..... 323
6.2.11 Power-Up Detection ..... 323
6.2.12 Oscillator Activity ..... 324
6.2.13 Interrupt Handling ..... 324
6.2.14 Battery-Backed RAMs and Registers ..... 325
6.2.15 RTC Registers ..... 325
Seconds Register (SEC) ..... 326
Seconds Alarm Register (SECA) ..... 327

## Table of Contents (Continued)


Minutes Register (MIN) ..... 327
Minutes Alarm Register (MINA) ..... 327
Hours Register (HOR) ..... 327
Hours Alarm Register (HORA) ..... 328
Day Of Week Register (DOW) ..... 328
Date Of Month Register (DOM) ..... 328
Month Register (MON) ..... 328
Year Register (YER) ..... 329
RTC Control Register A (CRA) ..... 329
RTC Control Register B (CRB) ..... 330
RTC Control Register C (CRC) ..... 331
RTC Control Register D (CRD) ..... 332
Date of Month Alarm Register (DOMA) ..... 332
Month Alarm Register (MONA) ..... 332
Century Register (CEN) ..... 333

6.2.16 BCD and Binary Formats ..... 333
6.2.17 Usage Hints ..... 333
6.2.18 RTC General-Purpose RAM Map ..... 333

**7.0 Device Specifications**

7.1 GENERAL DC ELECTRICAL CHARACTERISTICS ..... 334
7.1.1 Recommended Operating Conditions ..... 334
7.1.2 Absolute Maximum Ratings ..... 334
7.1.3 Capacitance ..... 335
7.1.4 Power Supply Current Consumption under Recommended Operating Conditions ..... 335
7.1.5 Voltage Thresholds ..... 335

7.2 DC CHARACTERISTICS OF PINS BY I/O BUFFER TYPES ..... 336
7.2.1 Input, CMOS Compatible with Schmitt Trigger ..... 336
7.2.2 Input, PCI 3.3V ..... 336
7.2.3 Input, SMBus Compatible ..... 336
7.2.4 Input, TTL Compatible ..... 337
7.2.5 Input, TTL Compatible with Schmitt Trigger ..... 337
7.2.6 Output, TTL Compatible Push-Pull Buffer ..... 337
7.2.7 Output, Open-Drain Buffer ..... 338
7.2.8 Output, PCI 3.3V ..... 338
7.2.9 Exceptions ..... 338
7.2.10 Terminology ..... 338

7.3 INTERNAL RESISTORS ..... 339
DC Test Conditions ..... 339
7.3.1 Pull-Up Resistor ..... 339
7.3.2 Pull-Down Resistor ..... 339

7.4 ANALOG CHARACTERISTICS ..... 340
7.4.1 ADC Characteristics ..... 340
Voltage Measurement ..... 340
7.4.2 ACM Characteristics ..... 341
7.4.3 DAC Characteristics ..... 341

7.5 PACKAGE THERMAL INFORMATION ..... 342

## Table of Contents (Continued)


7.6 AC ELECTRICAL CHARACTERISTICS ..... 343
7.6.1 AC Test Conditions ..... 343
Definitions ..... 343
7.6.2 Reset Timing ..... 345
7.6.3 Clock Timing ..... 346
..... 346
7.6.4 BIU Timing ..... 348
7.6.5 GPIO Port Timing ..... 352
7.6.6 PWM Timing ..... 353
7.6.7 MSWC Timing ..... 353
7.6.8 PS/2 Interface Timing ..... 354
7.6.9 ACCESS.bus Timing ..... 355
7.6.10 MFT16 Timing ..... 358
7.6.11 ICU/Development Timing ..... 359
7.6.12 Asynchronous Edge Detected Signals Timing ..... 360
7.6.13 Debugger Interface Timing ..... 361
7.6.14 USART Timing ..... 363
7.6.15 LCLK and RESET1-2 ..... 365
7.6.16 LPC and SERIRQ Signals ..... 366

**A. Register List**

A.1 CORE DOMAIN REGISTERS ..... 367
A.1.1 Module Configuration ..... 367
A.1.2 Bus Interface Unit (BIU) ..... 367
A.1.3 DMA Controller ..... 367
A.1.4 General-Purpose I/O (GPIO) Ports ..... 368
A.1.5 PS/2 Ports ..... 370
A.1.6 Host Interface (KBC, PM1 and PM2 Channels) ..... 370
A.1.7 Multi-Function Timer (MTF16) 1 ..... 371
A.1.8 Multi-Function Timer (MFT16) 2 ..... 371
A.1.9 Timing and Watchdog (TWD) ..... 372
A.1.10 Analog to Digital Converter (ADC) ..... 372
A.1.11 Digital to Analog Converter (DAC) ..... 372
A.1.12 ACCESS.bus Interface (ACB) 1 ..... 373
A.1.13 ACCESS.bus Interface (ACB) 2 ..... 373
A.1.14 ACCESS.bus Interface (ACB) 3 ..... 373
A.1.15 ACCESS.bus Interface (ACB) 4 ..... 374
A.1.16 Analog Comparators Monitor (ACM) ..... 374
A.1.17 Power Management (PM) ..... 374
A.1.18 High Frequency Clock Generator (HFCG) ..... 375
A.1.19 Development System Support ..... 375
A.1.20 Multi-Input Wake-Up (MIWU) ..... 375
A.1.21 Interrupt Control Unit (ICU) ..... 376
A.1.22 Debugger Interface ..... 376
A.1.23 Pulse Width Modulator (PWM) ..... 377

## Table of Contents (Continued)


A.1.24 Universal Synchronous/Asynchronous Receiver Transmitter (USART) 1 ..... 377
A.1.25 Universal Synchronous/Asynchronous Receiver Transmitter (USART) 2 ..... 378
A.1.26 Shared Memory Core ..... 378
A.1.27 Core Access to SuperI/O ..... 378
A.1.28 Mobile System Wake-Up Control (MSWC) ..... 379

A.2 HOST DOMAIN REGISTERS ..... 380
A.2.1 Configuration Registers ..... 380
Common SuperI/O Configuration ..... 380
Shared Memory ..... 380
RTC Configuration ..... 380
A.2.2 Host Runtime Registers ..... 381
Shared Memory Host ..... 381
MSWC Host Registers ..... 381
Host Interface (HI) ..... 381
Power Management Channel 1 ..... 382
Power Management Channel 2 ..... 382

A.3 CORE DOMAIN REGISTER LAYOUT ..... 383
A.3.1 Module Configuration ..... 383
RTC ..... 383
A.3.2 Bus Interface Unit (BIU) ..... 384
A.3.3 DMA Controller ..... 384
A.3.4 General-Purpose I/O (GPIO) Port ..... 384
A.3.5 PS/2 Interface ..... 385
A.3.6 Core Interface ..... 385
A.3.7 Multi-Function Timer (MFT16) ..... 386
A.3.8 Timing and Watchdog (TWD) ..... 386
A.3.9 Analog to Digital Converter (ADC) ..... 386
A.3.10 Digital to Analog (DAC) ..... 386
A.3.11 ACCESS.bus Interface (ACB) ..... 386
A.3.12 Analog Comparators Monitor (ACM) ..... 387
A.3.13 Power Management (PM) ..... 387
A.3.14 High-Frequency Clock Generator (HFCG) ..... 387
A.3.15 Development System Support ..... 387
A.3.16 Multi-Input Wake-Up (MIWU) ..... 387
A.3.17 Interrupt Control Unit (ICU) ..... 388
A.3.18 Debugger Interface ..... 388
A.3.19 Pulse with Modulator (PWM) ..... 388
A.3.20 Universal Synchronous/Asynchronous Receiver Transmitter (USART1 and USART2) ..... 389
A.3.21 Shared Memory Core ..... 389
A.3.22 Core Access to SuperI/O Modules ..... 389
A.3.23 MSWC ..... 389

A.4 HOST DOMAIN REGISTER LAYOUT ..... 390
Host Configuration Registers ..... 390
A.4.1 SuperI/O Configuration ..... 390

## Table of Contents (Continued)


A.4.2 Shared Memory Configuration ..... 390
A.4.3 RTC Configuration ..... 390
Host Runtime Registers ..... 391
A.4.4 Shared Memory Host ..... 391
A.4.5 MSWC Host ..... 391
A.4.6 Host Interface (HI) Registers ..... 391
A.4.7 RTC Registers ..... 392

**B. Software for Hardware Interface**

Information Block Access Index Register (IBAI) ..... 393
Information Block Data Register (IBD) ..... 393

**C. Booter Program**

C.1 BOOT DATA ..... 394
C.2 BOOT SEQUENCE ..... 396
C.3 RECOVERY MODE ..... 397
C.3.1 RS-232 Connection ..... 397
C.3.2 JTAG Connection ..... 398
C.4 MONITOR MEMORY WRITES ..... 398
C.5 EXTERNAL FLASH ERASE ..... 399
C.6 DEBUGGING CAPABILITIES OF THE BOOTER ..... 399

**Physical Dimensions** ..... 400

# 1.0 Introduction

## 1.1 DOCUMENT ORGANIZATION

This document describes the PC87591L-N05 architecture and device specifications. It is organized as follows:

Chapter 1 - Introduction, provides an overview of PC87591L-N05 modules, system connections, operating modes and configuration.

Chapter 2 - Signal/Pin Description and Configuration, lists the PC87591L-N05 pins and describes their functions and multiplexing options.

Chapter 3 - Power, Reset and Clocks, describes the PC87591L-N05 power supplies, clock scheme and reset sequence.

Chapter 4 - Embedded Controller Modules, describes the modules that comprise the CompactRISC core peripherals.

Chapter 5 - Host Controller Interface Modules, describes the modules and functions that interface core operation with the host.

Chapter 6 - Host-Controlled Modules and Host Interface, defines the configuration and control functions.

Chapter 7 - Device Specifications, defines the AC, DC and analog characteristics of the PC87591L-N05.

Appendix A - Summary of Registers, provides a composite listing of all relevant data on core domain registers and summarizes the registers' layouts.

Appendix B - This section includes directions for the software to handle some predetermined hardware interfaces and provides details of software-to-hardware interface conventions.

Appendix C - This section describes the Booter program.

## 1.2 GENERAL DESCRIPTION

The PC87591L-N05 is a highly integrated, embedded controller with an embedded RISC core and system functions. Targeted for a wide range of portable applications that use the Low Pin Count (LPC) interface, it also features a protection system and host BIOS firmware.

### 1.2.1 System Connections

Figure 1 shows the system connections of the PC87591L-N05 in a typical mobile PC application. The PC87591L-N05 requires few, if any, system glue elements. For a typical application, the PC87591L-N05 includes all required memory and peripherals on-chip. For more complex applications, it allows simple low-cost expansion, using its bus. Some of the features illustrated are mutually exclusive, depending on pin functions.

The major elements of the PC87591L-N05 are:

- Embedded Controller (EC) functions, which include: PS/2 devices, keyboard matrix, ACCESS.bus, timers, D/A and A/D converters and GPIO pins that can be assigned to various functions, as needed. External memory and peripheral devices may be added to extend the functionality of the on-chip resources.
- Host Processor interface based on the LPC bus and additional signals for interrupts and system power management
- Power Supplies for Host interface functions, EC and backup battery
- Clocks, using a 32.768 KHz crystal and optional clock output
- Strap inputs to initialize the PC87591L-N05 to different operation modes

In addition to the wide range of internal peripherals, the PC87591L-N05 provides hooks so that the system can be expanded in an easy and cost-effective manner, as follows:

- I/O expansion to support additional I/O port pins, using low-cost, standard 74HCxx devices or ASICs.
- On-chip memories may be expanded to interface with external RAM, flash or ROM devices.

### 1.2.2 Power Management

The PC87591L-N05 has an advanced power management scheme controlled by the host and/or the PC87591L-N05 firmware. The supported SuperI/O functions may be controlled directly by the host, using ACPI compliant schemes. The EC may also interface with the host via one or two communication channels. Power Management events are available for ACPI-compliant operation with the respective parts of the host chipset.

The PC87591L-N05 is designed to operate as the embedded controller of an ACPI-compliant system. It is equipped with various system power monitor and control functions and advanced means to control its own power consumption and power modes.

ACPI-compliant wake-up and sleep control are integrated into the PC87591L-N05. These functions are powered by $V_{CC}$.

The Keyboard Controller and the EC functions (core and associated peripherals) are powered by $V_{CC}$, which remains active as long as the system has a power source (e.g., main battery or outlet). Using $V_{CC}$, the core may be programed to monitor and control the system even when the host processor is turned off. To support this, the PC87591L-N05 is equipped with advanced means to control its power consumption: software controlled clock frequency throttling, the ability to disable mod-

## 1.0 Introduction (Continued)

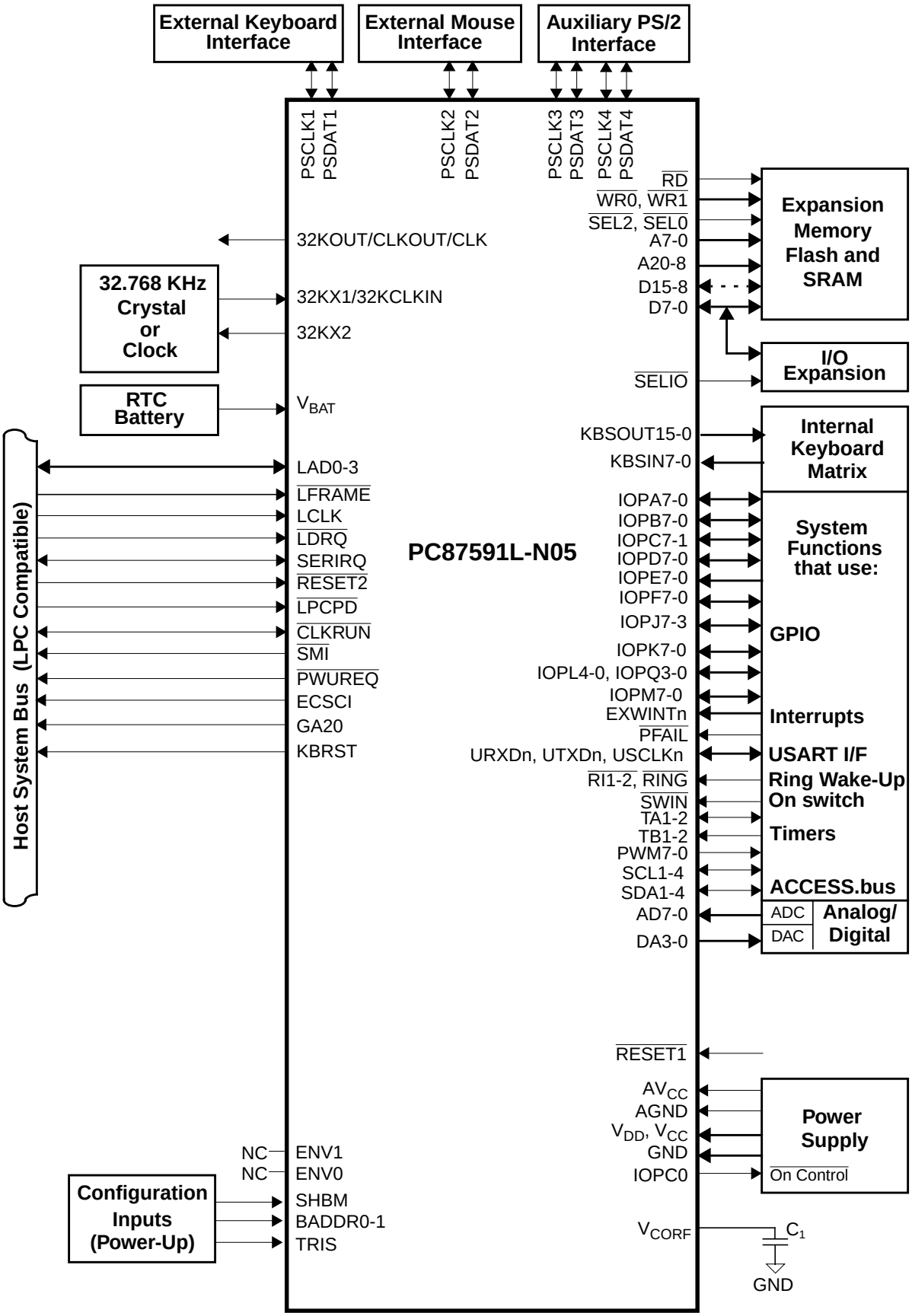



**Figure 1. PC87591L-N05 System Connection Diagram**

## 1.0 Introduction (Continued)

ules, and four power modes. These power modes include:

- Active Mode - Full functionality
- Active Mode Executing WAIT Instruction - Core execution and associated operations (such as memory access) are suspended
- Idle - Main clock is stopped, but the device can be woken up by internal or external events. The system tick timer is still operational and can be used to periodically wake up the device.
- Power Off - $V_{CC}$ is absent and only backup battery is available to supply the Real Time Clock (RTC) and retain the state of some memory and configuration elements.

### 1.2.3 Operating Environments

The PC87591L-N05 can operate in one of the following three environments:

- **Internal ROM Enabled (IRE)**. Used while the PC87591L-N05 operates in the production system and executes the application. The external ROM is the main source of code for the device.
- **On-Board Development (OBD)**. Used to debug the PC87591L-N05 code while it is mounted on its final production board. All pins have their IRE functionality. Interface to a debugger (running on the host) is through the JTAG-based debugger interface. OBD environment is binary and cycle-by-cycle compatible with IRE environment.
- **Development (DEV).** Used in Application Development Boards (ADB) or In System Emulators (ISE). In this environment, the external ROM is replaced with off-chip SRAM memory to allow flexible and fast development of application code. Some pins are allocated to development system use, and the GPIO functions associated with them are replicated using off-chip logic as part of the ADB system. DEV environment is binary and cycle-by-cycle compatible with OBD and IRE environments.

## 1.3 INTERNAL ARCHITECTURE

The PC87591L-N05 consists of several main components, divided into three groups:

- Core Domain
  - CR16B core processing unit
  - Bus Interface Unit and Memory Controller (BIU)
  - RAM and ROM memory
  - Core peripherals
- Host Domain
  - Host-Controlled functions
  - Host Interface
- Host-Controller Interface

The descriptions below detail the functions of the various blocks shown in Figure 2.

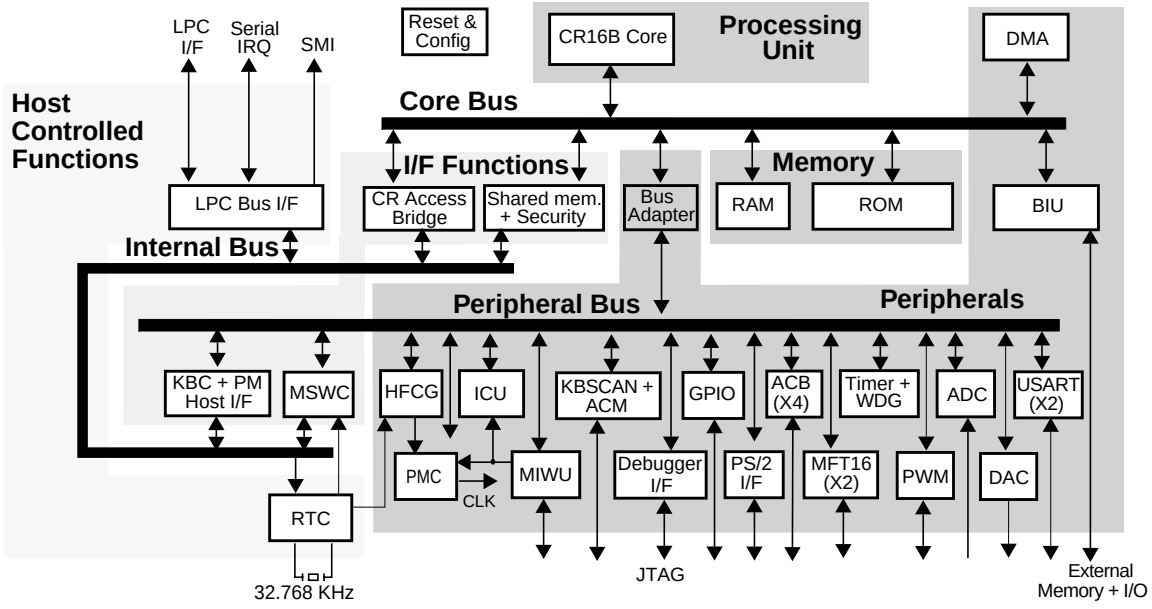



**Figure 2. PC87591L-N05 Functional Block Diagram**

# 1.0 Introduction (Continued)

## 1.3.1 Processing Unit

The CompactRISC CR16B core (referred to in this datasheet as the “core”) is an advanced, general-purpose 16-bit microprocessor core with a RISC architecture. The core is responsible for arithmetic and logic operations, as well as program control.

For more details about the core structure and instruction set, see *CR16B Programmer’s Reference Manual, Revision 2.2,* September 1999 (Literature Number: 633150-001)

## 1.3.2 Bus Interface Unit and Memory Controller (BIU)

The BIU enables access to off-chip memory and I/O devices. It is organized in zones, as follows:

- Zone 0 and 2 - Expansion memory (flash and/or SRAM). This memory may be used for the core code, data and/or the host BIOS program.
- Zone 1 - This zone is available for off-chip in DEV environment only and is used for emulating the operation of the on-chip base memory, using an off-chip SRAM. In IRE and OBD environments, the configuration of this zone should be the same as in DEV environment to enable cycle-by-cycle compatibility.
- I/O Zone - This zone can be used for I/O expansion. In DEV environment, it can be used to recreate GPIO signals, whose pins are used for the development system interface.

Configuration registers that control the bus transactions are associated with each zone and are part of the BIU module, See Section 4.1 on page 67 for details. For details about the link between DEV environment and the BIU, see Section 1.4.3 on page 29.

## 1.3.3 Memory

**RAM.** The 4096-byte on-chip RAM is mostly used for the storage of program variables and the stack. It can also store short programs used while the flash memory is being updated. Part of the on-chip RAM is reserved for use by the core development tools monitor program, TMON (part of the Booter). For information, see the *CompactRISC™ PC87591x Tmonlib Version 3.1.2.3 Release Letter*, March 2001.

**ROM.** The PC87591L-N05 is equipped with a small pre-programed ROM, which functions as a boot ROM.

**External Flash.** The PC87591L-N05 hardware arbitrates flash usage by the core firmware and the host processor BIOS program when Shared-BIOS configuration is selected. Flash sharing is based on in-parallel “cycle stealing” so both the host processor and the core can execute code in parallel from the same memory device. The host processor typically copies the flash contents to the host’s main memory (DRAM) on system boot to improve access time to it and enable execution when the flash’s contents is compressed. It is important to do this early in the boot process to reduce resource contention between the core and the host.

## 1.3.4 Peripherals

**The ICU** (Interrupt Control Unit) collects interrupts from various internal and external (through the MIWU) sources and uses the vectored interrupt mechanism to notify the core of events. It supports 31 maskable interrupt inputs (see Table 15 on page 97 for the interrupt assignment) and, via the $\overline{\text{PFAIL}}$ input, a Non-Maskable Interrupt (NMI).

**The MIWU** (Multi-Input Wake-Up) module enables collecting various internal and external interrupt sources (events), generates interrupts in Active mode and enables the PC87591L-N05 to return from Idle mode to Active mode. The core can separately enable or disable each wake-up conditions. The PC87591L-N05 has a total of 28 wake-up signals, some of which are grouped together to generate a single interrupt signal to the ICU.

**The PMC** (Power Management Controller) controls PC87591L-N05 power consumption according to the required activity level. Power consumption is adjusted by controlling the clock frequency and selective enabling/disabling of three power modes: Active, Idle and Power Off. Activity can be resumed by external events (through the MIWU) or internal events, such as a periodic wake-up.

**The Clock Generator** provides clocks for the various core-related on-chip modules. These clocks are generated directly from a 32.768 KHz crystal or from the on-chip High-Frequency Clock Generator (HFCG). The HFCG generates the high-frequency clock using the RTC’s 32.768 KHz clock signal as a reference. The PC87591L-N05 operation frequency is set by programing the HFCG registers. The PMC enables and disables high-frequency clock generation, according to the required power mode.

**The GPIO Ports** (General-Purpose Input/Output) module consists of up to 92 GPIO port signals that serve as an interface to and provide control for the PC system. Some of these GPIO port signals share their pins with an alternate function (see Table 6 on page 49), with which they may be mutually exclusive. When configured as inputs, some of these signals can interrupt the core when an event is detected, even if the device is in Idle mode. An example is the SWIN input, which is dedicated to the PC On/Off switch.

## 1.0 Introduction (Continued)

**Internal keyboard scanning** is supported by 16 open-drain output signals and eight input signals. Switch-based keyboard matrices are supported with CMOS Schmitt trigger inputs with internal pull-up resistors. For power efficiency, the inputs include an interrupt and a wake-up capability, so that pressing/releasing keys may be identified without scanning the keyboard matrix in either Active, Power Save or Idle modes. The keyboard interrupt is controlled by the MIWU.

**The PS/2 Interface** enables interface with industry-standard, PS/2-compatible keyboard, mouse and other pointing devices. The PC87591L-N05 supports up to four PS/2 devices via its dedicated 4-channel PS/2 interface module. Each channel has two quasi-bidirectional signals. The symmetric structure of the channels enables software-controlled interchanging of devices to channels.

The PC87591L-N05 includes a hardware accelerator, which allows the PS/2 channels to be controlled with minimal software overhead. It also eliminates the sensitivity to interrupt latency that characterized traditional solutions.

**The ACB Interface** is a two-wire serial interface compatible with the ACCESS.bus physical layer. It is also compatible with Intel's SMBus and Philips' $I^2C$. This module can serve as a bus master or slave and performs both transmit or receive operations. As a slave, it can respond to two assigned addresses, a global call address and an SMBus ARP address.

The PC87591L-N05 includes two ACB Interface modules, which allows operation on two isolated buses in the system.

**The USART** (Universal Synchronous Asynchronous Receiver Transmitter) gives full-duplex support for a wide range of software programmable baud rates and data formats. It handles automatic parity generation and several error detection schemes. It also supports DMA transfers, which provides fast processor-independent receive and transmit. The PC87591L-N05 includes two USART interfaces.

**The MFT16** (Multi-Function 16-Bit Timer) contains two 16-bit timers with a range of operation modes. These timers can operate, using several clock sources, in PWM, Capture or Counter mode to satisfy a wide range of application requirements.

The PC87591L-N05 includes two MFT16 modules, each of which may be assigned functions and configured independently.

**The TWD** (Timer and Watchdog) module has a 16-bit periodic interrupt timer that can be programed to generate interrupts at pre-defined intervals and an 8-bit watchdog timer that can reset the PC87591L-N05 whenever the software loses control of the processor.

The periodic timer is typically used as a system tick timer. This timer is fed by the 32.768 KHz clock. Thus its counting is not impacted by the setting of the HFCG, and it may continue to operate even in Idle mode. This enables it to serve as a periodic wake-up source during Idle mode.

**The PWM** (Pulse Width Modulator) module provides eight modulated output signals. All of these signals have the same (programmable) frequency and each signal has an individually programmable 8/16-bit duty cycle.

**The ADC** (Analog to Digital Converter) provides the PC87591L-N05 with an accurate means for measuring slowly changing voltages and temperature. The ADC module can measure up to ten external and four internal voltages with 8-bit resolution over a voltage range of 0 to 2.97V. It can measure temperature using thermistors.

**The DAC** (Digital to Analog Converter) has four channels of voltage output. Each of the four DAC channels has an 8-bit resolution with a full output range from AGND to $AV_{CC}$. The DAC provides a settling time of about 1 µs on a 50 pF load.

**The Debugger Interface** module provides a JTAG-based interface to a remote, host-based debugger. This interface enables device debugging while in OBD environment (i.e., in the final production board) or in DEV environment once in the development system.

### 1.3.5 Host-Controller Interface Modules

Chapter 5 on page 242 describes a set of modules that resides in the boundary between the host-controlled functions and the core-controlled functions. These modules are used for message communication, data exchange, memory access and generating power management events to the host. Chapter 5 also discusses the mechanism that enables the core to access the host-controlled peripherals.

**The Keyboard Controller, Power Management** module has three channels that are available for keyboard and power management (EC)-related host-controller communication.

The keyboard and mouse data channel (i.e., host legacy I/O addresses $60_{16}$ and $64_{16}$) is compatible with the legacy interface of keyboard controllers. It may be used with polling or interrupts. For use with interrupts, the module can generate the two legacy IRQ signals: IRQ1 and IRQ12. In addition, the PC87591L-N05 generates the gate A20 control signal (GA20 pin) and a soft reset signal ($\overline{\text{KBRST}}$ pin) to the host. Optionally, this $\overline{\text{KBRST}}$ reset signal can be used to prevent the host from accessing the shared flash when the PC87591L-N05 is not ready to perform shared memory access (i.e., during PC87591L-N05 boot-up). See "GA20 Pin Functionality" on page 285 and "Host Keyboard Fast Reset" on page 284 for details).

The PC87591L-N05 supports two Power Management channels, in compliance with ACPI requirements for Embedded Controller (EC) interface. This enables the PC87591L-N05 to implement an EC interface that operates in either shared or private modes. The number of Power Management channels in use and the addresses they respond to (i.e., legacy host I/O addresses $62_{16}$ and $66_{16}$) are configured in the Host Controlled Functions configuration space. These channels may generate

## 1.0 Introduction (Continued)

IRQ, $\overline{SMI}$ or $\overline{SCI}$ events to the host. The Power Management channels include a PC87570-compatible mode and an enhanced scheme that enables more efficient control by the core.

**Shared Memory and Protection** is supported between the host and the core. This sharing may be used for the support of a shared BIOS scheme, for protected information storage and/or for the PC87591L-N05 firmware update by the host. The expansion memory can be used as shared memory. The Shared Memory module provides means for the host to access the shared memory. It can also protected access to portions of the shared memory for read and/or write operations to allow reliable and tamper-protected storage and protected update.

**The Mobile System Wake-Up** module includes various system wake-up and power management services that may be handled either by the host or the core. The wake-up sources may be the RTC or external events such as ring detection on the $\overline{RING}$ input or modem RI inputs. The module provides hooks for ACPI-compliant drivers, which enable the drivers to handle wake-up events, change the system power state (including turning it off) and interface to the core firmware. Mask bits can be enabled to determine whether the core or the host handles each one of the events. In addition, this module provides status information about the host domain (e.g., reset input state and $V_{DD}$ supply status).

**The Core Access to Host-Controlled Peripherals** module enables core access to SuperI/O modules. It can interleave usage of a module with the host or take control of it and prevent any host access to that module.

### 1.3.6 Host-Controlled SuperI/O Modules and Host Interface

**The Host Interface** is based on Intel's Low Pin Count (LPC) interface, as defined in *LPC Interface Specification, Revision 1.1*. This interface enables the host to perform read and write cycles using I/O space accesses and memory space accesses and FWH transactions. Interrupts are sent to the host, using the serial IRQ protocol.

The PC87591L-N05 supports the advanced power management features of the LPC bus. The $\overline{SMI}$ signal may be sent to interrupt the host and put it in System Management Mode (SMM). The $\overline{PWUREQ}$ signal may be connected to one of the wake-up inputs of the host chipset and used to trigger an SCI event for various EC communication purposes. The PC87591L-N05 can operate with a slowed down or stopped LPC clock and can re-start the LPC clock as part of the system power management capabilities, using the $\overline{CLKRUN}$ signal. The $\overline{LPCPD}$ input enables turning off LPC bus supply while the PC87591L-N05 and some Host Controlled functions are operating.

**Host Configuration.** The PC87591L-N05 includes a set of global configuration register and seven logical devices, each with associated configuration registers.

The central configuration register set supports ACPI-compliant PnP configuration. The configuration registers are structured as a subset of the Plug and Play Standard registers defined in Appendix A of the *Plug and Play ISA Specification, Revision 1.0a* by Intel and Microsoft. All system resources assigned to the functional blocks (I/O address space and IRQ lines) are configured in and managed by the central configuration register set. In addition, some function-specific parameters are configurable through the configuration registers and distributed to the functional blocks through special control signals.

**The RTC** (Real Time Clock) has a low-power timekeeping mechanism that provides a time-of-day, year-2000-compatible calendar with a century counter and alarm features. It can work from either $V_{CC}$ or a backup battery, using an internal switch. Other features include three maskable interrupt sources and 242 bytes of general-purpose RAM. An external battery source maintains valid RAM and time during $V_{CC}$ failure. The RTC is software compatible with the DS1287 and MC146818.

## 1.4 OPERATING ENVIRONMENTS

On Power-Up reset, the ENV1-0 and TRIS input signals select one of the following operating environments:

- Internal ROM Enabled (IRE)
- On Board Development (OBD)
- Development (DEV)

See Section 2.3 on page 48 for more information about these pins and controlling the loads connected to them.

Code written for IRE environment is executable in all environments, since it is binary compatible. The execution time of code in on-chip base memory (in IRE environment) is identical to that in OBD and DEV environments; i.e., the operation is cycle-by-cycle compatible.

The PC87591L-N05 is factory tested to ensure that it operates in either IRE or OBD environment. Only selected parts are tested for operation in DEV environment.

### 1.4.1 IRE Environment

IRE environment is used for PC87591L-N05 operation in the production system and for normal execution of applications. The external flash is the main source of code for the device. In this environment, after reset, the PC87591L-N05 starts running the code written in the first address of the internal ROM.

The PC87591L-N05 is shipped with 4 Kbytes of on-chip boot code. The user is expected to use an external memory for most of the code and constant data.

To maximize on-chip ROM performance, configure the BIU as described in Section 4.1.11 on page 84.

## 1.0 Introduction (Continued)

See Figure 1 on page 24 for a system example in IRE environment. In this environment, the ENV0, ENV1 and TRIS strap pins do not need any external pull-up resistors.

### 1.4.2 OBD Environment

OBD environment is used for debugging the PC87591L-N05 firmware while it is mounted on its final production board. All pins have their IRE functionality, and the interface to a debugger running on the host is enabled using the JTAG based debugger interface. In OBD Environment, code is executed from the expansion memory. Breakpoints on data and code access may be applied using the core hardware breakpoint mechanism. The monitor stored in the on-chip ROM is used as part of the debugging environment.

OBD environment is binary and cycle-by-cycle compatible with IRE environment.

See Figure 4 on page 30 for a system example in OBD environment. In this environment, the ENV0 and TRIS strap pins are left unconnected, and ENV1 requires an external pull-up resistor.

### 1.4.3 DEV Environment

DEV environment is used in Application Development Boards (ADBs) or In System Emulators (ISEs). In this mode, the on-chip ROM and the external flash are replaced with off-chip SRAM memory to allow flexible and fast development of application code. Some pins are allocated for development system use, and the GPIO functions associated with the pins are replicated using off-chip logic as part of the ADB system. DEV environment is binary and cycle-by-cycle compatible with OBD and IRE environments.

In this environment, the pins of ports PH, PI, PJ, PK, PL and PM are allocated for the interface to the off-chip base memory, core status signals, reset output and a breakpoint input. The system may regain these ports using the I/O Expansion protocol and off-chip logic, while maintaining cycle-by-cycle and binary compatibility with IRE and OBD environments. Using the same software, this environment is binary and cycle-by-cycle compatible with IRE and OBD environments. All features of IRE environment can be implemented either directly or by using additional external logic.

See Figure 5 on page 31 for a system example in DEV environment. In this environment, the ENV0 strap pin needs an external pull-up resistor and the ENV1 and TRIS pins are left unconnected.

## 1.5 MEMORY MAP

The PC87591L-N05 has two address domains: core and host. The host address space is composed of the host I/O address space and the host memory space. Section 1.5.1 discusses the mapping of memories and peripherals into the core address space. Figure 3 shows the memory map and the shared access schemes that are possible. Section 1.5.2 discusses the mapping of the host address space and ways of accessing it.

The PC87591L-N05 enables several memories in the core address space to have restricted access to the host. These memories are referred to as “shared memory” or “shared BIOS”. In addition, the core can access the Host Controlled Functions. Section 1.5.2 and Section 1.5.3 discuss these instances of cross-domain access, respectively.

### 1.5.1 Core Address Domain Memory Map

The memory and I/O devices are directly mapped into the 2 Mbyte address space of the core. The core address space can include both code and data. However, access to data stored in the first 64 Kbytes of the address space is more efficient.

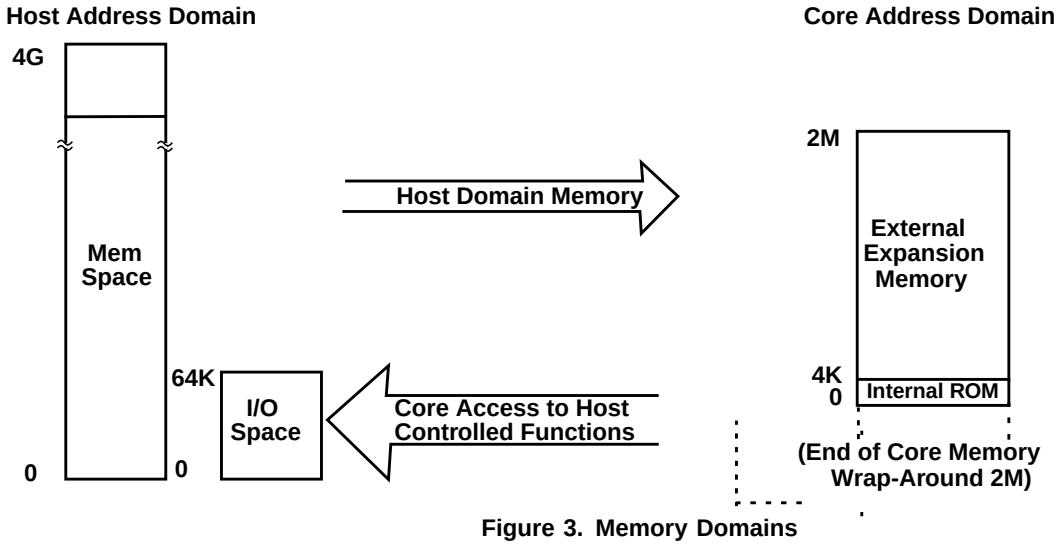



**Figure 3. Memory Domains**

## 1.0 Introduction (Continued)

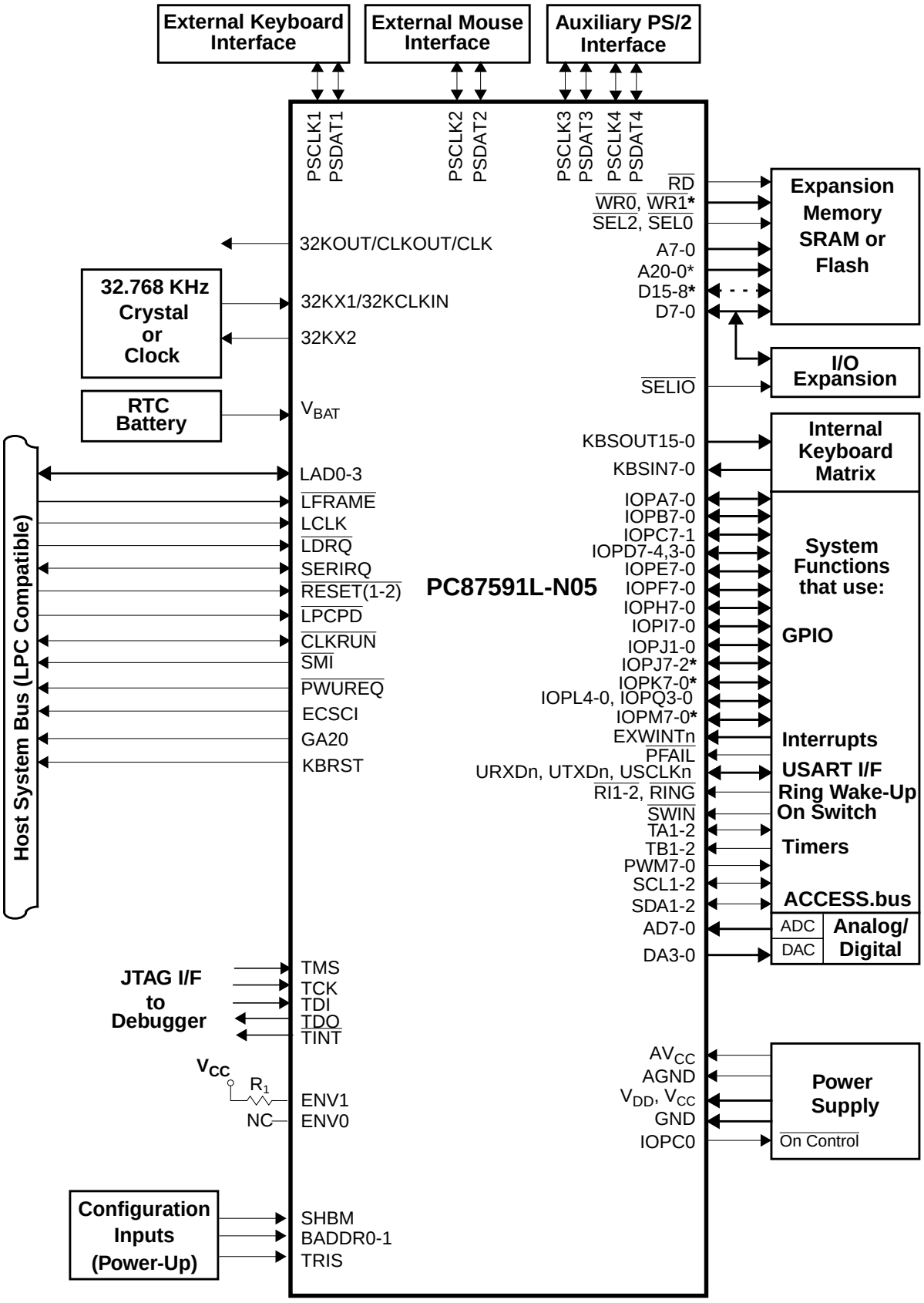



**Figure 4. OBD Environment PC87591L-N05 System Connection Diagram**

## 1.0 Introduction (Continued)

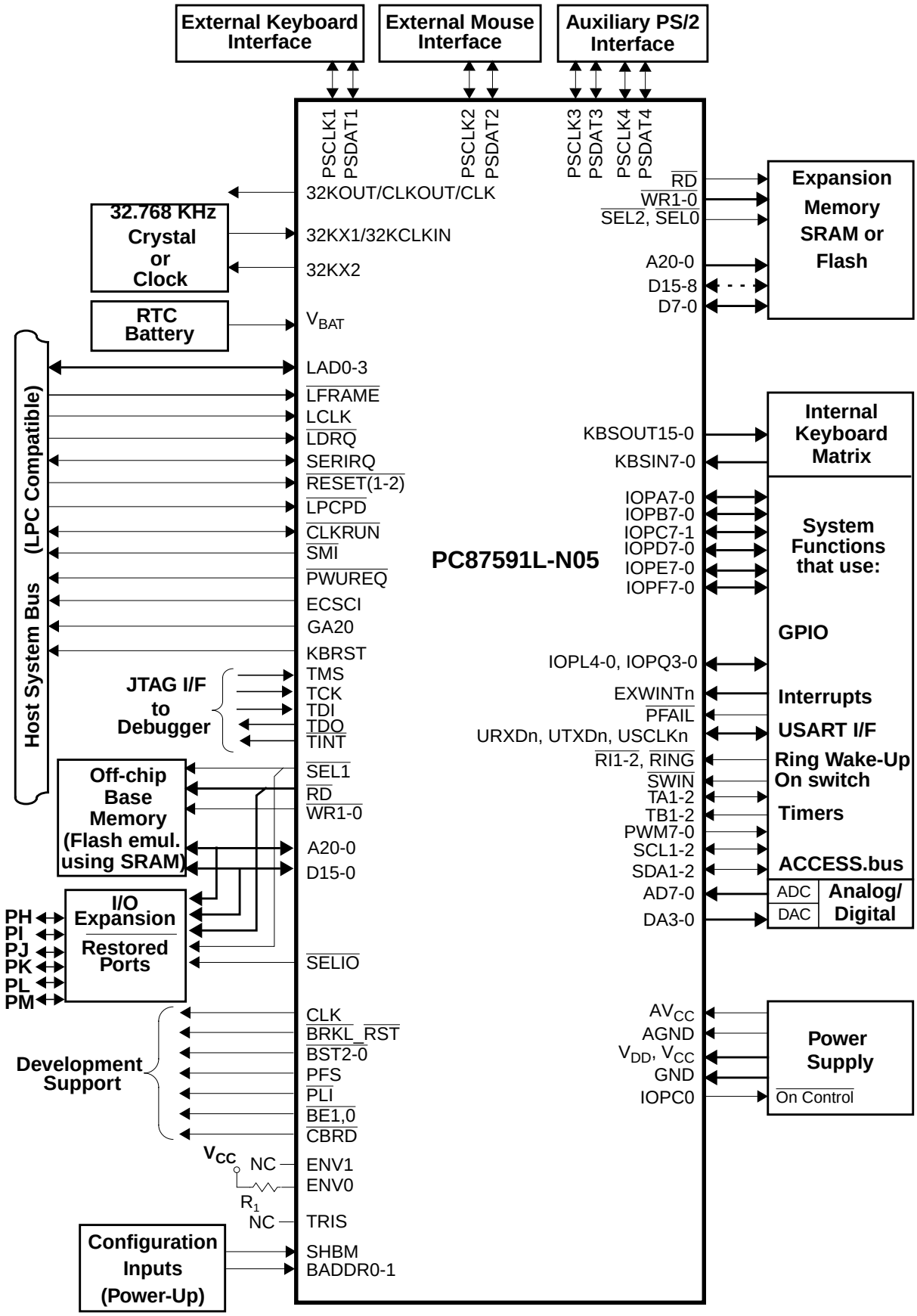



**Figure 5. DEV Environment PC87591L-N05 System Connection Diagram**

## 1.0 Introduction (Continued)

The core boot section is stored in the base memory. This memory is:

- On-chip ROM in IRE and OBD environments
- Off-chip memories (SRAM or flash memory) in DEV environment

The constant data and the remaining core code is stored in external expansion memory, which is one of the following:

- Flash memory in IRE and OBD environments
- SRAM or flash memory in DEV environment

The on-chip RAM and various peripherals are also mapped into the core address space.

Table 1 shows how the PC87591L-N05 memory and I/O devices are mapped in the core address space. Appendix A on page 367 shows the address map of the registers for the other modules.

Addresses not included in the following table or Appendix A are reserved. Attempts to access reserved addresses produce unpredictable results.

**Table 1. PC87591L-N05 Memory Map**

| Address | Size (Bytes) | Description | |
|---|---|---|---|
| | | Purpose | Environment |
| $00\ 0000_{16}$ – $00\ 0FFF_{16}$ | 4K | Base Memory | IRE & OBD - Internal ROM<br>DEV - External Base Memory (Zone 1) |
| $00\ 1000_{16}$ – $00\ DFFF_{16}$ | 52K | Expansion Memory (Zone 0 or Zone 2[1]) | |
| $00\ E800_{16}$ – $00\ F7FF_{16}$ | 4K | System RAM[2] | |
| $00\ F880_{16}$ – $00\ F883_{16}$ | 4 | Information Block Access Registers[3] | |
| $00\ F900_{16}$ – $00\ F90A_{16}$ | 11 | Shared BIOS and Protection Registers[4] | |
| $00\ F980_{16}$ – $00\ F98F_{16}$ | 16 | BIU Registers[4] | |
| $00\ FA00_{16}$ – $00\ FA7F_{16}$ | | DMA Controller Registers[4] | |
| $00\ FB00_{16}$ – $00\ FBFF_{16}$ | 256 | I/O Expansion[5] | |
| $00\ FC00_{16}$ – $00\ FFFF_{16}$ | 1K | On-Chip Module Registers[4] | |
| $01\ 0000_{16}$ – $1F\ FFFF_{16}$ | 1984K | Expansion Memory (Zone 0 or Zone 2[1]) | |

1. Zone 2 is enabled by bit 5 of Module Configuration Register (MCFG) (see Page 52). The size of zone 2 is selected by bits 0-1 of the PTWRH register (see Page 55).
2. The system RAM size is controlled by bit 7 of the PTWRH register (see Page 55).
3. See Appendix B on page 393 for details of the implemented registers.
4. See Appendix A on page 367 for details of the implemented registers.
5. See “Accessing I/O Expansion Space” on page 34.

### Register Abbreviations and Access

The following abbreviations are used to indicate the Register Type:

- R/W= Read/Write
- R= The Read portion of a register, where a read from a specific address returns the value of a specific register; a write to the same address is to a different register.
- W= The Write portion of a register as described above for ‘R’.
- RO= Read Only
- WO= Write Only
- R/W1C= Read/Write 1 to Clear. Writing 1 to a bit clears it to 0. Writing 0 has no effect.

Either byte-wide or word-wide transactions to any address within the memory address space may be used to access memory devices.

Only byte-wide transactions may be used to access byte-wide registers, and only word-wide transactions may be used to access word-wide registers. Attempts to read a write-only register or write to a read-only register cause unpredictable results.

Zeros must be written to reserved bits unless stated otherwise. Reading reserved bits returns an undefined value. When modifying a register with reserved bits, the data read from reserved a bit can be written back to it.

## 1.0 Introduction (Continued)

### Accessing Base Memory

The base memory is used for storing code and data required for basic boot operations. The rest of the code and data are stored in the expansion memory shared by the core firmware and host BIOS.

Figure 6 on page 33 shows how on-chip and off-chip base memory are mapped to the PC87591L-N05 address space.

**IRE and OBD Environments (On-Chip Base Memory).** In IRE and OBD environments, the on-chip ROM is used as base memory. The access time to it is controlled by BIU zone 1. To allow cycle-by-cycle compatibility with DEV environment, this zone should be programed in the same way for all environments. Thus to maximize on-chip ROM performance, configure BIU zone 1, as described in Section 4.1.11 on page 84.

**DEV Environment (Off-Chip Base Memory).** In DEV environment (when on-chip ROM is disabled), the boot code and constant data are stored in off-chip base memory. The size of the off-chip base memory is 4 Kbytes.

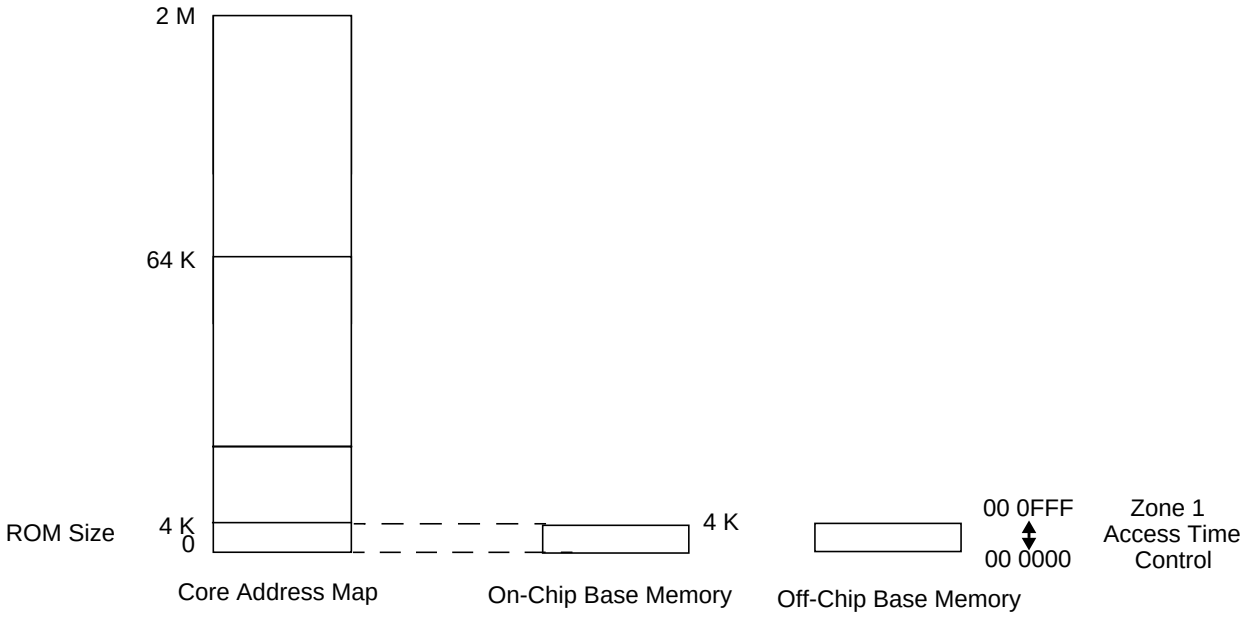



**Figure 6. Base Memory Address Mapping**

### Accessing Expansion Memory

The expansion memory of the PC87591L-N05 is divided into two zones when BIU zone 2 is enabled:

- Low Zone - for addresses with base memory in the range of 00 1000 to 03 FFFF. The access time to this zone is controlled by the BIU zone 2 configuration registers. The zone 2 memory can be configured for 64K, 128K, 192K or 256K ranges. The configuration of the zone 2 range is selected by bits 8-9 of Protection Word Low Register (PTWRL) (see Page 54).
- High Zone - for addresses with base memory in the range of 01 0000 to 1F FFFF. The access time to this zone is controlled by the BIU zone 0 configuration registers.

When BIU zone 2 is disabled, the whole expansion memory of the PC87591L-N05 is controlled by the BIU zone 0 configuration registers.

Access to expansion memory is enabled only after:

- The pins used for the memory interface are configured to operate as expansion memory interface signals (see Section 2.4 on page 49 for details on the alternate functions configuration).
- The BIU zone 0 register and/or zone 2 configuration registers (SZCFG0, SZCFG2), which control the memory access parameters (e.g., bus width and access time), is set to support the configuration in use.
- Zone 0 or zone 2 is selected for $00\ 1000_{16}$ – $00\ DFFF_{16}$ Expansion memory address range by bit 5 of Module Configuration Register (MCFG) (see Page 52).

The interface signals to the expansion memory are:

- 8-bit flash or SRAM: $\overline{SEL0}$, $\overline{SEL2}$, $\overline{RD}$, $\overline{WR0}$, D0-7 and A0-20 (fewer address lines may be used with a smaller flash).
- 16-bit flash or SRAM: $\overline{SEL0}$, $\overline{SEL2}$, $\overline{RD}$, $\overline{WR0\text{-}1}$, D0-15 and A1-20 (fewer address lines may be used with a smaller flash).

## 1.0 Introduction (Continued)

Figure 7 shows the Expansion Memory Address Range mapping to the core in the PC87591L-N05 device.

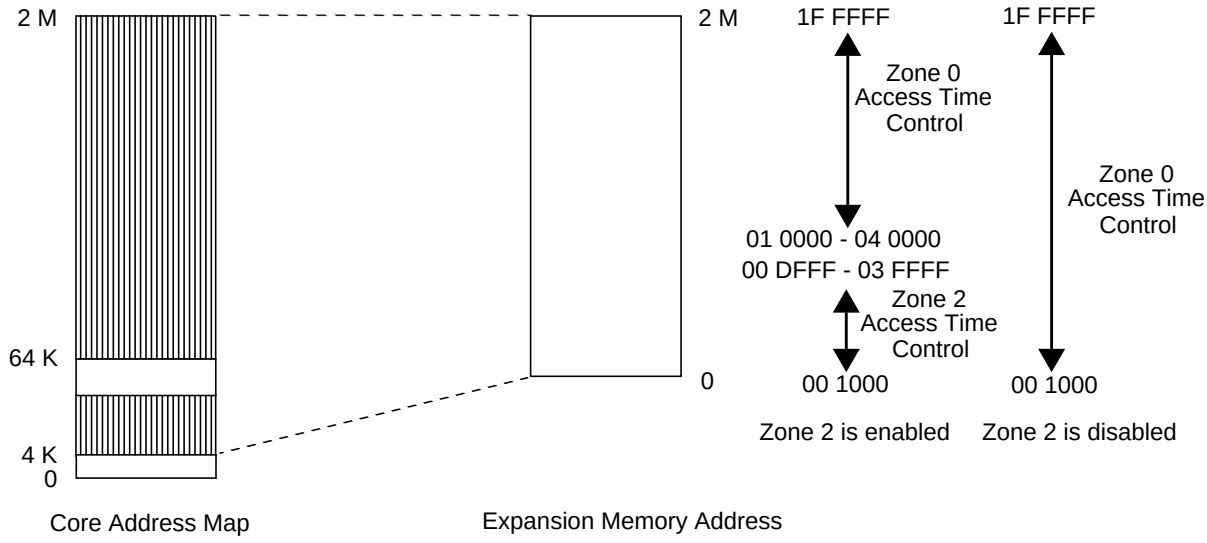



**Figure 7. Expansion Memory (Zone 0 and Zone 2) Address Range**

**External Memory Mapping into Shared BIOS Memory**

When the shared BIOS memory is enabled using the SHBM strap input (SHBM=1) or the Shared Memory configuration registers (LDN=10h), the expansion memory address range is mapped into the address range of the host. The PC87591L-N05 uses the wrap-around effect of the core address space (on a 2 Mbyte boundary) using “growing down” addresses, which enables the host to view the Expansion flash as a continuation of the BIOS Memory. See Section 5.3.2 on page 262 for details of the memory mapping scheme.

### Accessing I/O Expansion Space

The I/O expansion protocol enables implementing I/O devices or GPIO ports in the system, in addition to those available on-chip, for IRE, OBD and DEV environments. In addition, in DEV environment, some of the on-chip I/O port pins are used to interface with off-chip peripherals. In such a case, the I/O expansion protocol is used to implement the functionality of these I/O ports, using off-chip external logic. Access to these ports is through the same addresses used for the ports’ on-chip implementation.

The I/O expansion space is mapped to the address space 00 $FB00_{16}$ to 00 $FBFF_{16}$. This space is partitioned as follows:

- Addresses in the range 00 $FB00_{16}$ to 00 $FB22_{16}$ are used by GPIO ports PH, PI, PJ, PK, PL and PM (either on-chip or in their off-chip implementation, while the chip is in DEV environment).
- Address 00 $FBFE_{16}$ is used only in DEV environment by the MCFGSH register and must be written after each write to the MCFG with the same data written to the MCFG.
- Addresses in the range 00 $FBC0_{16}$ to 00 $FBFF_{16}$ are reserved for development board use.
- All other addresses may be used by the application for adding additional I/O elements.

The PC87591L-N05 accesses the off-chip I/O expansion using the I/O zone of the BIU. The zone select signal ($\overline{SELIO}$), address lines A0-7 and the $\overline{RD}$ and $\overline{WR0}$ signals are used to interface to the off-chip logic.

### 1.5.2 Host Address Domain Memory Map

The host address space includes memory space and I/O space.

The I/O space used by the PC87591L-N05 is configured through the PC87591L-N05 configuration registers. The configuration register address is defined by strap inputs (BADDR0-1) to be one of two fixed addresses or an address defined by the core using registers in the MSWC module.

When a Shared BIOS scheme is enabled via the SHBM strap input, the PC87591L-N05 is mapped to enable the host to boot from a shared flash device. The PC87591L-N05 supports either memory or FWH transactions for the host memory interface, with automatic selection between them (see Section 6.1.11 on page 311). Section 5.3 on page 262 discusses the mapping of memory between the host and core domains and the read and write access protection scheme from the host and core sides.

## 1.0 Introduction (Continued)

Following the boot process, the Shared Memory configuration registers (see Section 6.1.11 on page 311) enable setting memory sharing. The configuration setting includes defining the memory protocol in use (memory or FWH) and the address range used in the host address space. The configuration registers allow the defaults set by the SHBM strap input to be overridden; this enables using the shared memory for purposes other than system BIOS (e.g., PC87591L-N05 firmware update and protected storage of information).

### 1.5.3 Core Access to Host Controlled Peripherals

The core may access host domain devices through the Core to Host Controlled Functions access bridge. The bridge employs an indirect mapping scheme.

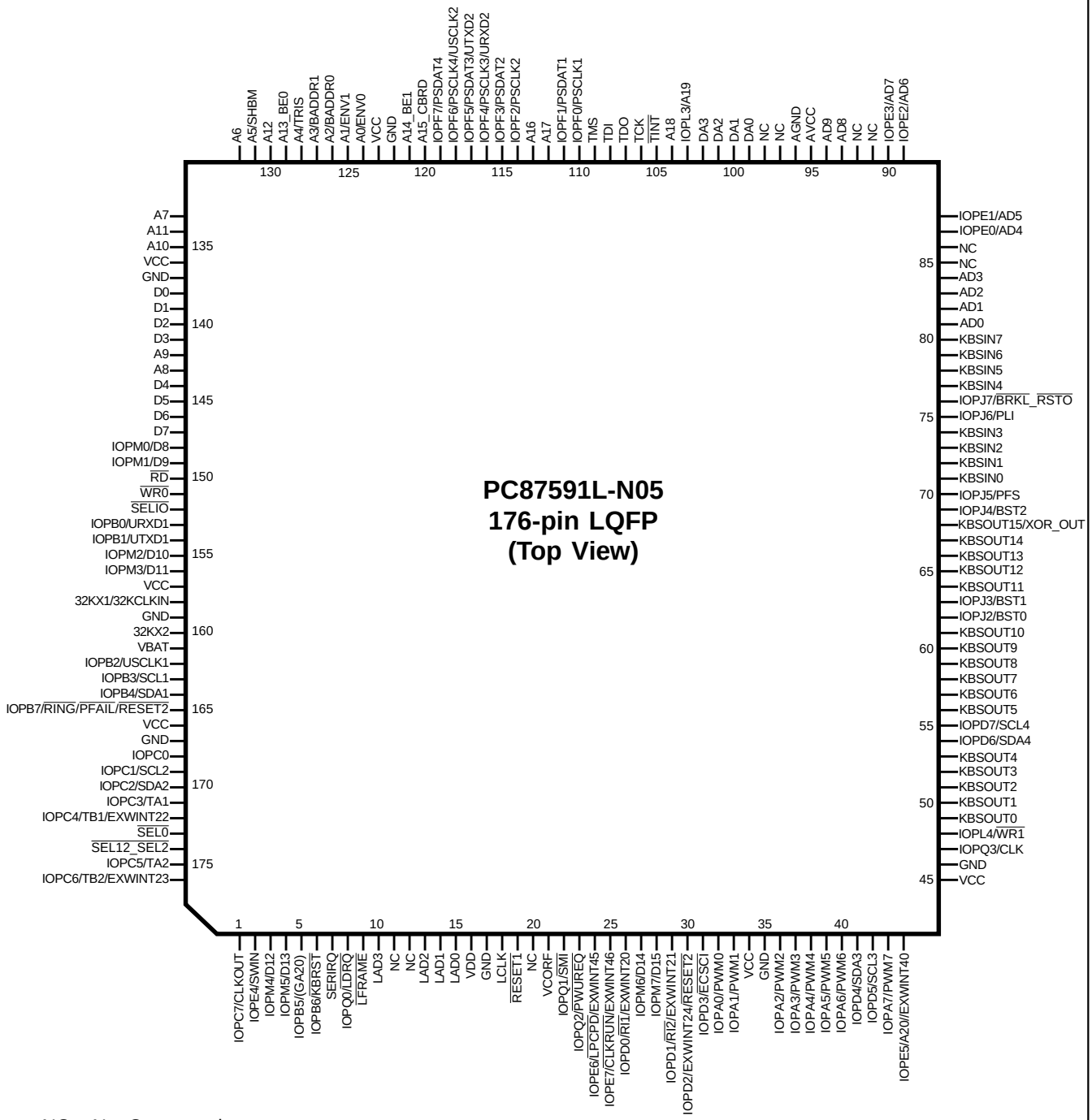
There is only a single set of peripheral registers for host and core use. The bus arbitration guarantees that only one of the two register accesses occurs at any given time, but this does not prevent problems that may be caused by conflicting write transactions. When such a case is expected, the core lock mechanism may be used to protect access to one or more of the devices. For security reasons, the lock may also be used to protect against host access to devices.

The core accesses a register by specifying the logical device and the offset of the register within the logical device. Note that the configuration registers' index and data registers are also handled as a logical device. The core triggers a read by writing 1 to the read start bit and waits for the bit to clear; it can then read the data from the data register. The core triggers a write by performing a write operation to the data register.

Section 5.4 on page 275 provides details of the bridge and its operation.

# 2.0 Signal/Pin Description and Configuration

## 2.1 CONNECTION DIAGRAMS




NC = Not Connected

**176-pin Low Profile Plastic Quad Flatpack (LQFP)**
**Order Number PC87591L-VPCN05**
**NS Package Number VPC176**

## 2.0 Signal/Pin Description and Configuration (Continued)

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | IOPC7/ CLKOUT | IOPC6/TB2/ EXWINT23 | SEL12_SEL2 | SEL0 | IOPC3/ TA1 | IOPC0 | IOPB3/ SCL1 | GND | RD | D6 | D3 | D0 | VCC | A10 | A7 |
| B | IOPM4/ D12 | IOPE4/ SWIN | IOPC5/ TA2 | IOPC4/TB1/ EXWINT22 | GND | IOPB4/ SDA1 | 32KX2 | IOPM3/ D11 | IOPB0/ URXD1 | IOPM1/ D9 | D5 | A9 | GND | A11 | A6 |
| C | IOPM5/ D13 | IOPB5/ (GA20) | IOPC2/ SDA2 | VCC | IOPB2/ USCLK1 | 32KX1/ 32KCLKIN | IOPM2/ D10 | IOPB1/ UTXD1 | WR0 | D7 | A8 | D1 | A2/ BADDR0 | A5/SHBM | A12 |
| D | IOPB6/ KBRST | LAD3 | SERIRQ | IOPC1/ SCL2 | IOPB7/RING/ PFAIL/RESET2 | VBAT | VCC | SELIO | IOPM0/ D8 | D4 | D2 | A1/ENV1 | GND | A4/TRIS | A13_BE0 |
| E | LFRAME | LAD2 | NC | IOPQ0/ LDRQ | | | | | | | | A14_BE1 | IOPF6/ PSCLK4/ USCLK2 | VCC | A3/ BADDR1 |
| F | LAD1 | GND | LAD0 | NC | | | | | | | | IOPF5/ PSDAT3/ UTXD2 | IOPF2/ PSCLK2 | A15_CBRD | A0/ENV0 |
| G | LCLK | VCORF | RESET1 | VDD | | | | | | | | A16 | IOPF1/ PSDAT1 | IOPF4/ PSCLK3/ URXD2 | IOPF7/ PSDAT4 |
| H | IOPM6/ D14 | IOPE6/LPCPD/ EXWINT45 | IOPQ1 SMI | NC | | | | | | | | TDI | IOPF0/ PSCLK1 | A17 | IOPF3/ PSDAT2 |
| J | IOPD3/ ECSCI | IOPM7/ D15 | IOPQ2/ PWUREQ | IOPE7/CLKRUN/ EXWINT46 | | | | | | | | A18 | TDO | TMS | TCK |
| K | IOPA2/ PWM2 | IOPA0/ PWM0 | IOPD0/RI1/ EXWINT20 | IOPD1/RI2/ EXWINT21 | | | | | | | | DA1 | IOPL3/A19 | TINT | DA3 |
| L | IOPA5/ PWM5 | GND | IOPD2/EXWINT24/ RESET2 | IOPA1/ PWM1 | | | | | | | | AGND | DA0 | DA2 | NC |
| M | IOPD4/ SDA3 | IOPA6/ PWM6 | VCC | IOPA3/ PWM3 | KBSOUT3 | KBSOUT5 | KBSOUT9 | KBSOUT11 | IOPJ4/ BST2 | KBSIN2 | KBSIN4 | AD0 | AVCC | NC | AD9 |
| N | IOPD5/ SCL3 | IOPA7/ PWM7 | IOPA4/ PWM4 | KBSOUT2 | IOPD7/ SCL4 | KBSOUT8 | IOPJ3/ BST1 | KBSOUT13 | KBSOUT14 | IOPJ5/ PFS | KBSIN3 | KBSIN5 | AD1 | AD8 | NC |
| P | IOPE5/A20/ EXWINT40 | GND | KBSOUT0 | IOPD6/ SDA4 | KBSOUT6 | KBSOUT10 | KBSOUT12 | KBSOUT15/ XOR_OUT | KBSIN1 | IOPJ7/ BRKL_RSTO | KBSIN6 | AD3 | IOPE0/ AD4 | IOPE3/ AD7 | NC |
| R | VCC | IOPQ3/CLK | IOPL4/WR1 | KBSOUT1 | KBSOUT4 | KBSOUT7 | IOPJ2/BST0 | KBSIN0 | IOPJ6/PLI | KBSIN7 | AD2 | NC | NC | IOPE1/ AD5 | IOPE2/ AD6 |

**PC87591L-N05**
**176-pin FBGA**
**(Top View)**

NC = Not Connected

**176-Pin Fine Pitch Ball Grid Array (FBGA)**
**Order Number PC87591L-SLCN05**
**NS Package Number SLC176A**

# 2.0 Signal/Pin Description and Configuration (Continued)

## 2.2 BUFFER TYPES AND SIGNAL/PIN DIRECTORY

The following sections contain detailed functional descriptions and electrical DC characteristics for the PC87591L-N05 signals. The pin multiplexing and function selection criteria are described in Section 2.4 on page 49. The signal DC characteristics are denoted by a buffer type symbol described briefly below and in further detail in Section 7.2 on page 336. The pin multiplexing information refers to two different types of multiplexing:

- MUX - Multiplexed, denoted by a slash (/) between pins in the diagram in Section 2.1 on page 36. Pins are shared between two different functions. Each function is associated with different board connectivity. Normally, the function selection is determined by the board design and can not be changed dynamically. The multiplexing options must be configured by the core in order to comply with the board implementation.
- MM - Multiple Mode, denoted by an underscore (_) between pins in the diagram in Section 2.1 on page 36. Pins have two or more modes of operation within the same function. These modes are associated with the same external (board) connectivity. Mode selection may be controlled by the device driver through the registers of the functional block and do not require a special setup. These pins are not considered multiplexed pins from the configuration perspective. The mode selection method as well as the signal specification in each mode are described in the functional description.

**Table 2. Buffer Types**

| Symbol | Description |
|---|---|
| $IN_{AC}$ | Input, analog to ACM |
| $IN_{AD}$ | Input, analog to ADC |
| $IN_{CS}$ | Input, CMOS compatible, with Schmitt Trigger |
| $IN_{OSC}$ | Input, from crystal oscillator (not characterized) |
| $IN_{PCI}$ | Input, PCI 3.3V |
| $IN_{SM}$ | Input, SMBus compatible |
| $IN_{T}$ | Input, TTL compatible |
| $IN_{TS}$ | Input, TTL compatible, with Schmitt Trigger |
| $IN_{ULR}$ | Input, power, resistor protected (not characterized) |
| $O_{p/n}$ | Output, push-pull output buffer that is capable of sourcing *p* mA and sinking *n* mA |
| $OD_{n}$ | Output, open-drain output buffer that is capable of sinking *n* mA |
| $O_{DA}$ | Output, analog from DAC |
| $O_{OSC}$ | Output, to crystal oscillator (not characterized) |
| $O_{PCI}$ | Output, PCI 3.3V |
| PWR | Power pin |
| GND | Ground pin |

### 2.2.1 ACCESS.bus Interface

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| SCL4-1 | 55, 42, 169, 163 | N5, N1, D4, A7 | I/O | $IN_{SM}/OD_{2}$ | $V_{CC}$ | **ACCESS.bus Serial Clock 1, 2, 3 and 4 Signals**. An internal pull-up for this pin is optional. |
| SDA4-1 | 54, 41, 170, 164 | P4, M1, C3, B6 | I/O | $IN_{SM}/OD_{2}$ | $V_{CC}$ | **ACCESS.bus Serial Data 1, 2, 3 and 4 Signals**. An internal pull-up for this pin is optional. |

# 2.0 Signal/Pin Description and Configuration (Continued)

## 2.2.2 Analog Interface

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| AD9-0 | 94-93, 90-87, 84-81 | M15, N14, P14, R15, R14, P13, P12, R11, N13, M12 | I | $IN_{AD}$ | $AV_{CC}$ | **Analog to Digital Converter Inputs.** |
| DA3-0 | 102-99 | K15, L14, K12, L13 | O | $O_{DA}$ | $AV_{CC}$ | **Digital to Analog Converter Outputs.** |

## 2.2.3 Clocks

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| 32KCLKIN | 158 | C6 | I | $IN_T$ | $V_{PP}$ | **32.768 KHz Clock Input.** |
| 32KX1 | 158 | C6 | I | $IN_{OSC}$ | $V_{PP}$ | **32.768 KHz Crystal Oscillator Input.** Input from external crystal oscillator circuitry. See Figure 105 on page 318. |
| 32KX2 | 160 | B7 | O | $O_{OSC}$ | $V_{PP}$ | **32.768 KHz Crystal Oscillator Output.** Output to external crystal oscillator circuitry. See Figure 105 on page 318. |
| CLK | 47 | R2 | O | $O_{1/2}$ | $V_{CC}$ | **Clock Output.** PC87591L-N05 Core Domain system clock. Available for all environments. |
| CLKOUT | 1 | A1 | O | $O_{2/4}$ | $V_{CC}$ | **Clock Output.** This pin may output either the 32.768 KHz clock or the Core Domain system clock (CLK). |

## 2.2.4 Core Bus Interface Unit (BIU)

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| A3-0 | 127-124 | E15, C13, D12, F15 | O | $O_{1/2}$ | $V_{CC}$ | **Address Bus, Bits 0-20.** Core external address bus. Affected or used by any transaction on the internal core bus, including DMA transactions and accesses to the internal memory. |
| A20-8 | 44, 103-104, 112-113, 120-121, 129-130, 134-135, 142-143 | P1, K13, J12, H14, G12, F14, E12, D15, C15, B14, A14, B12, C11 | | | | |
| A7-4 | 133-131, 128 | A15, B15, C14, D14 | O | $O_{2/12}$ | $V_{CC}$ | |
| D15-0 | 28-27, 4-3, 156-155, 149-148, 147-144, 141-138 | J2, H1, C1, B1, B8, C7, B10, D9, C10, A10, B11, D10, A11, D11, C12, A12 | I/O | $IN_T/O_{1/2}$ | $V_{CC}$ | **Data Bus, Bits 15-0.** Core external data bus. Used for core access to external memory or I/O devices. When accessing an 8-bit external memory or I/O device, only D7-0 should be used. |

## 2.0 Signal/Pin Description and Configuration (Continued)

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| $\overline{RD}$ | 150 | A9 | O | $O_{1/2}$ | $V_{CC}$ | **Read Control.** Core external read strobe. Can be used as output enable for external memory of I/O devices. |
| $\overline{SEL0}$ | 173 | A4 | O | $O_{1/2}$ | $V_{CC}$ | **Zone Select 0.** Chip-select signal for external memory devices mapped to zone 0. |
| $\overline{SEL12}$_ $\overline{SEL2}$ | 174 | A3 | O | $O_{1/2}$ | $V_{CC}$ | **Zone Select 1 and 2.** Chip-select signal for external memory devices mapped to zones 1 and 2. Zone Select 1 is available in DEV environment for off-chip emulation of the on-chip ROM. In IRE and OBD environments, SEL2 output is used for Zone Select 2 when zone 2 is enabled. |
| $\overline{SELIO}$ | 152 | D8 | O | $O_{1/2}$ | $V_{CC}$ | **I/O Zone Select.** Chip-select signal for external I/O devices mapped to this zone. Can be used for mapping off-chip I/O expansion devices to the core address space. |
| $\overline{WR1}$ | 48 | R3 | O | $O_{1/2}$ | $V_{CC}$ | **Write Control 1 and 0.** Indicates writes to bytes 1 and 0, respectively, of the BIU data bus. |
| $\overline{WR0}$ | 151 | C9 | | | | |

### 2.2.5 Development System Support

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| BE1-0 | 121,129 | E12, D15 | O | $O_{1/2}$ | $V_{CC}$ | **Byte Enable** bits 1 and 0 on monitor bus cycles. |
| $\overline{BRKL}$_$\overline{RSTO}$ | 76 | P10 | I | $IN_T/O_{1/2}$ | $V_{CC}$ | **Break Line and Reset Out.** When the Core bus is active, the pin is used as a $\overline{BRKL}$ input. It indicates to the core that a breakpoint is needed when the currently fetched instruction goes into execution. When the Core bus is not active, the signal is used as a $\overline{RSTO}$ output. It indicates to the system that an internal reset to the core and its peripherals occurred. |
| BST2-0 | 69, 63-62 | M9, N7, R7 | O | $O_{1/2}$ | $V_{CC}$ | **Bus Status** bits 2-0 on monitor bus cycles. In DEV environment, these pins allow monitoring of the external bus cycles. When OBR bit in BCFG register is set, the internal bus cycles are also visible on the external bus. See also Table 31 on page 237. |
| CBRD | 120 | F14 | O | $O_{1/2}$ | $V_{CC}$ | **Core Bus Read Status** on monitor bus cycles. Available in all modes. See Section 4.1.9 on page 80. |
| PFS | 70 | N10 | O | $O_{1/2}$ | $V_{CC}$ | **Pipe Flow Status.** |
| PLI | 75 | R9 | O | $O_{1/2}$ | $V_{CC}$ | **Pipe Long Instruction.** |
| TCK | 106 | J15 | I | $IN_{TS}$ | $V_{CC}$ | **JTAG Test Clock.** |
| TDI | 108 | H12 | I | $IN_T$ | $V_{CC}$ | **JTAG Test Data In.** |
| TDO | 107 | J13 | O | $O_{1/2}$ | $V_{CC}$ | **JTAG Test Data Out.** |
| $\overline{TINT}$ | 105 | K14 | O | $OD_2$ | $V_{CC}$ | **JTAG Test Interrupt.** |
| TMS | 109 | J14 | I | $IN_T$ | $V_{CC}$ | **JTAG Test Mode Select.** |

## 2.0 Signal/Pin Description and Configuration (Continued)

### 2.2.6 General-Purpose I/O (GPIO) and Internal Keyboard Scan

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| KBSIN7-0 | 80-77, 74-71 | R10, P11, N12, M11, N11, M10, P9, R8 | I | $IN_{CS}/IN_{AC}$ | $V_{CC}$ | **Keyboard Scan Inputs.** The input side of the internal keyboard scan lines. These inputs may be configured to work with either Schmitt trigger inputs, for operation with switch based keyboards, or with internal analog voltage comparators to interface to a resistive keyboard matrix. KBSIN7-0 are referenced to $V_{CC}$ and are designed to work using a 3.3V supply. See Section 4.14 on page 197 for more details. |
| KBSOUT15-0 | 68-64, 61-56, 53-49 | P8, N9, N8, P7, M8, P6, M7, N6, R6, P5, M6, R5, M5, N4, R4, P3 | O | $OD_6$ | $V_{CC}$ | **Keyboard Scan Outputs.** The output side of the internal keyboard scan lines. |
| IOPA7-0 | 43, 40-36, 33-32 | N2, M2, L1, N3, M4, K1, L4, K2 | I/O | $IN_{TS}/O_{3/6}$ | $V_{CC}$ | **General-Purpose I/O Ports.** The GPIO registers are accessible by the core for read, write and configuration. Each of these GPIO signals can be individually configured to be input or output. The pins may be used as GPIO or assigned to their respective alternate functions. See Section 2.4 on page 49 for GPIO pin assignment to alternate functions. See Section 4.5 on page 110 for further details on the GPIO pins and their functionality. |
| IOPD1-0 | 29, 26 | K4, K3 | | | | |
| IOPB7-3 | 165, 6-5, 164-163 | D5, D1, C2, B6, A7 | I/O | $IN_{TS}/O_{1/2}$ | $V_{CC}$ | |
| IOPC6-1 | 176-175, 172-169 | A2, B3, B4, A5, C3, D4 | | | | |
| IOPJ7-2 | 76-75, 70-69, 63-62 | P10, R9, N10, M9, N7, R7 | | | | |
| IOPL4-3 | 48, 103 | R3, K13 | | | | |
| IOPM7-0 | 28-27, 4-3, 156-155, 149-148 | J2, H1, C1, B1, B8, C7, B10, D9 | | | | |
| IOPQ2-1 | 23-22 | J3, H3 | | | | |
| IOPQ3 | 47 | R2 | O | $O_{1/2}$ | $V_{CC}$ | |
| IOPQ0 | 8 | E4 | I/O | $IN_{TS}/O_{PCI}$ | $V_{DD}$ | |
| IOPB2-0 | 162, 154-153 | C5, C8, B9 | I/O | $IN_{TS}/O_{2/4}$ | $V_{CC}$ | |
| IOPC7 | 1 | A1 | | | | |
| IOPD7-2 | 55-54, 42-41, 31-30 | N5, P4, N1, M1, J1, L3 | I/O | $IN_{TS}/O_{2/12}$ | $V_{CC}$ | |
| IOPF7-0 | 119-114, 111-110 | G15, E13, F12, G14, H15, F13, G13, H13 | | | | |

## 2.0 Signal/Pin Description and Configuration (Continued)

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| IOPC0 | 168 | A6 | O | $O_{3/6}$ | $V_{CC}$ | **General-Purpose Output Port IOPC0.** IOPC0 is are targeted for use as power supply control for the power supply unit. It is accessible by the core for write and configuration. On $V_{CC}$ Power-Up reset, the default state is TRI-STATE; it is not affected by other types of reset. |
| IOPE7-0 | 25-24, 44, 2, 90-87 | J4, H2, P1, B2, P14, R15, R14, P13 | I | $IN_{TS}$ | $V_{CC}$ | **General-Purpose Input Port.** These pins serve as input-only pins. The GPIO registers are accessible by the core for read and configuration. The pins may be used as GPIO or assigned to their respective alternate functions. IOPE3-0 and IOPE5 do not have an internal pull-up resistor option. Note that IOPE3-0 and IOPE7-6 are not 5V tolerant. See Section 2.4 on page 49 for GPIO pin assignment to alternate functions. See Section 4.5 on page 110 for further details on the GPIO pins and their functionality. |

### 2.2.7 Host Interface

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| $\overline{\text{CLKRUN}}$ | 25 | J4 | I/O | $IN_{PCI}/OD_6$ | $V_{DD}$ | **Clock Run.** Same as PCI $\overline{\text{CLKRUN}}$. When high, it indicates that the LPC clock will be slowed down or stopped. In this case, the PC87591L-N05 may pull it down to request full speed of the clock. |
| GA20 | 5 | C2 | O | $O_{1/2}$ | $V_{CC}$ | **Gate A20.** Implemented using IOPB5 port output. See **Section 5.5.4 on page 284** for signal operation and behavior when $V_{DD}$ is off. |
| $\overline{\text{KBRST}}$ | 6 | D1 | O | $O_{1/2}$ | $V_{CC}$ | **Keyboard Reset Output.** See **Section 5.5.4 on page 284** for signal operation and behavior when $V_{DD}$ is off. |
| LAD0-3 | 15-13, 10 | D2, E2, F1, F3 | I/O | $IN_{PCI}/O_{PCI}$ | $V_{DD}$ | **LPC Address-Data.** Multiplexed command, address bidirectional data and cycle status. |
| LCLK | 18 | G1 | I | $IN_{PCI}$ | $V_{DD}$ | **LPC Clock.** Practically the PCI clock (up to 33 MHz). |
| $\overline{\text{ECSCI}}$ | 31 | J1 | O | $O_{2/12}$ | $V_{CC}$ | **EC SCI.** Generates an Embedded Controller SCI interrupt to the chipset. This signal is typically connected to one of the chipset GPI inputs. |
| $\overline{\text{LDRQ}}$ | 8 | E4 | O | $O_{PCI}$ | $V_{DD}$ | **LPC DMA Request**. Encoded Bus Master request for LPC I/F. |
| $\overline{\text{LFRAME}}$ | 9 | E1 | I | $IN_{PCI}$ | $V_{DD}$ | **LPC Frame**. Low pulse indicates the beginning of a new LPC cycle or the termination of a broken cycle. |
| $\overline{\text{LPCPD}}$ | 24 | H2 | I | $IN_{PCI}$ | $V_{CC}$ | **Power Down.** Indicates that power will be shut off on the LPC interface. |
| $\overline{\text{RESET1}}$ | 19 | G3 | I | $IN_{PCI}$ | $V_{CC}$ | **Reset 1.** A falling edge on this signal starts a reset sequence of the PC87591L-N05. For details, see Section 3.2 on page 61. |
| $\overline{\text{RESET2}}$ | 30/165 | L3 (D5) | I | $IN_{CS}$ | $V_{DD}$ | **Reset 2.** A level low reset to the LPC interface and Host Controlled Function configuration registers and Shared Memory host registers. $\overline{\text{RESET2}}$ is either assigned to one of two optional pins or it is disabled. For details, see Section 3.2 on page 61. |

## 2.0 Signal/Pin Description and Configuration (Continued)

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| $\overline{\text{PWUREQ}}$ | 23 | J3 | O | $O_{1/2}$ | $V_{CC}$ | **Power-Up Request.** |
| SERIRQ | 7 | D3 | I/O | $IN_{PCI}/O_{PCI}$ | $V_{DD}$ | **Serial IRQ.** The interrupt requests are serialized over a single pin, where each IRQ level is delivered during a designated time slot. |
| $\overline{\text{SMI}}$ | 22 | H3 | I/O | $IN_{TS}/OD_{12}$ | $V_{CC}$ | **System Management Interrupt.** |

# 2.0 Signal/Pin Description and Configuration (Continued)

## 2.2.8 Interrupt and Wake-Up Inputs (ICU and MIWU)

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| EXWINT20<br>EXWINT21<br>EXWINT22<br>EXWINT23<br>EXWINT24<br>EXWINT40<br>EXWINT45<br>EXWINT46 | 26<br>29<br>172<br>176<br>30<br>44<br>24<br>25 | K3<br>K4<br>B4<br>A2<br>L3<br>P1<br>H2<br>J4 | I | $IN_{TS}$ | $V_{CC}$ | **External Wake-up/Interrupt Inputs.** Fed into the Multi-Input Wake-up Unit (MIWU) for wake-up or interrupt. Can be used as external source for wake-up or interrupt. For details on interrupt and wake-up event assignment, see Table 16 on page 103. |
| $\overline{PFAIL}$ | 165 | D5 | I | $IN_{CS}$ | $V_{CC}$ | **Power Fail.** Non-maskable external interrupt source. This assigned interrupt is non-maskable only after being enabled by software after reset. |
| SWIN | 2 | B2 | I | $IN_{CS}$ | $V_{CC}$ | **Power Switch Input.** Indicates a user request to turn the power on or off. This signal is connected to the Multi-Input Wake-up Unit (MIWU) for wake-up of the core domain and interrupt generation to the core for handling. For details on interrupt and wake-up event assignment, see Table 16 on page 103. |

## 2.2.9 Power and Ground

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| AGND | 96 | L12 | I | GND | | **Analog Ground.** Used as ground for the Analog-to-Digital Converter (ADC) and the Digital-to-Analog Converter (DAC). |
| $AV_{CC}$ | 95 | M13 | I | PWR | | **Analog 3.3V Power** supply. Used as power supply for the Analog-to-Digital Converter (ADC) and the Digital-to-Analog Converter (DAC). |
| GND | 17, 35, 46, 122, 137, 159, 167 | F2, L2, P2, A8, B5, B13, D13 | I | GND | | **Ground.** Serves for both on-chip logic, output drivers and back-up battery circuit. See Section 3.1.3 on page 59 for details on connections with AGND. |
| $V_{BAT}$ | 161 | D6 | I | $IN_{ULR}$ | | **Battery Power Supply.** Provides battery backup to the Mobile System Wake-Up Control registers, to the RTC and to the 32 KHz crystal oscillator when $V_{CC}$ is lost. The pin is connected to the internal logic through a series resistor for UL protection. |
| $V_{CORF}$ | 21 | G2 | I/O | PWR | | **On-Chip Power Converter Filter.** On-chip power converter output. Powers the internal logic of all the device modules. An external 1 µF ceramic filter capacitor must be connected between this pin and GND. |
| $V_{DD}$ | 16 | G4 | I | PWR | | **Digital 3.3V Power Supply.** Serves as power supply for the LPC interface and some of the host-controlled functions. |
| $V_{CC}$ | 34, 45, 123, 136, 157, 166 | A13, C4, D7, E14, M3, R1 | I | PWR | | **Standby Digital 3.3V Power Supply.** Serves as power supply for the LPC interface and some of the host-controlled functions. |

# 2.0 Signal/Pin Description and Configuration (Continued)

## 2.2.10 PS/2 Interface

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| PSCLK4-1 | 118, 116, 114, 110 | E13, G14, F13, H13 | I/O | $IN_T/O_{2/12}$ | $V_{CC}$ | PS/2 Channel 1 through 4 Clock signal. |
| PSDAT4-1 | 119, 117, 115, 111 | G15, F12, H15, G13 | I/O | $IN_T/O_{2/12}$ | $V_{CC}$ | PS/2 Channel 1 through 4 Data signal. |

## 2.2.11 Strap Configuration and Testing

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| BADDR1-0 | 127-126 | E15, C13 | I | $IN_{CS}$ | $V_{CC}$ | **I/O Base Address.** Sampled at $V_{CC}$ Power-Up reset to determine the base address of the configuration Index-Data register pair as follows:<br>• No pull-up resistor: $2E_{16}$-$2F_{16}$<br>• 10 KΩ external pull-up resistor on BADDR0: $4E_{16}$-$4F_{16}$<br>• 10 KΩ external pull-up resistor on BADDR1: Core defined<br>• 10 KΩ external pull-up resistor on BADDR0 and BADDR1: XOR Tree Test Mode. When selecting this mode TRIS must be 0 (left unconnected). |
| ENV1-0 | 125-124 | D12, F15 | I | $IN_{CS}$ | $V_{CC}$ | **Environment Select 1 and 0.** Sampled at $V_{CC}$ Power-Up reset to determine the device operation environment, IRE, OBD or DEV. Each pin is pulled to 0 (set environment to IRE = 00) by an internal resistor or set to 1 by an external 10 KΩ pull-up resistor.<br>For further details refer to Section 2.3 on page 48. |
| SHBM | 131 | C14 | I | $IN_{CS}$ | $V_{CC}$ | **Shared Host BIOS Memory.** Sampled at $V_{CC}$ Power-Up reset to determine the state of the shared Host BIOS memory. Pulled to 0 (disables the shared host BIOS memory) by an internal resistor or set to 1 by an external 10 KΩ pull-up resistor to enable the shared BIOS mode. |
| TRIS | 128 | D14 | I | $IN_{CS}$ | $V_{CC}$ | **TRI-STATE.** Forces the device to float all its output and I/O pins (except for DAC outputs and 32KX2) if an 10 KΩ external pull-up resistor is connected. Sampled at $V_{CC}$ Power-Up reset. |
| XOR_OUT | 68 | P8 | O | $O_{4/8}$ | $V_{CC}$ | **XOR Tree Output.** All the device pins (except power type and analog type pins) are internally connected in a XOR tree structure. |

# 2.0 Signal/Pin Description and Configuration (Continued)

## 2.2.12 Mobile System Wake-Up Control (MSWC)

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| $\overline{RING}$ | 165 | D5 | I | $IN_{CS}$ | $V_{CC}$ | **Telephone Line Ring.** Detection of a pulse train on this pin is a wake-up event that can activate the power-up request ($\overline{PWUREQ}$). The pin has a Schmitt Trigger input buffer, powered by $V_{CC}$. |
| $\overline{RI1}$ | 26 | K3 | I | $IN_{TS}$ | $V_{CC}$ | **Ring Indicator.** When low, it indicates that a telephone ring signal has been received by the modem. Ring signals are monitored during Power-Off for wake-up event detection. |
| $\overline{RI2}$ | 29 | K4 | | | | |

## 2.2.13 Timers and PWM

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| TA2 | 175 | B3 | I/O | $IN_{T,}$ $O_{1/2}$ | $V_{CC}$ | **Timer Pin A for Timers 2 and 1.** |
| TA1 | 171 | A5 | | | | |
| TB2 | 176 | A2 | I | $IN_T$ | $V_{CC}$ | **Timer Pin B for Timers 2 and 1.** |
| TB1 | 172 | B4 | | | | |
| PWM7-0 | 43, 40-36, 33-32 | N2, M2, L1, N3, M4, K1, L4, K2 | O | $O_{3/6}$ | $V_{CC}$ | **PWM Output 7-0.** |

## 2.2.14 Universal Synchronous/Asynchronous Receiver/Transmitter (USART)

| Signal | LQFP Pin(s) | FBGA Ball(s) | I/O | Buffer Type | Power Well | Description |
|---|---|---|---|---|---|---|
| UTXD2-1 | 117, 154 | F12, C8 | O | $O_{2/4}$ | $V_{CC}$ | **USART1 and 2 Transmit Data.** |
| URXD2-1 | 119, 153 | G14, B9 | I | $IN_T$ | $V_{CC}$ | **USART1 and 2 Receive Data.** |
| USCLK2-1 | 118, 162 | E13, C5 | I/O | $IN_{CS}/O_{2/4}$ | $V_{CC}$ | **USART Serial Clock.** May be used as input or output when the USART is configured to operate in its synchronous mode of operation. |

## 2.0 Signal/Pin Description and Configuration (Continued)

### 2.2.15 Internal Pull-Up and Pull-Down Resistors

The signals listed in Table 3 can optionally support internal pull-up (PU) and/or pull-down (PD) resistors. See Section 7.3 on page 339 for the values of each resistor type.

**Table 3. Internal Pull-Up and Pull-Down Resistors**

| Signal | LQFP Pin(s) | FBGA Ball(s) | Type | Comments |
|---|---|---|---|---|
| **ACCESS.bus (ACB)** | | | | |
| SCL4-1 | 55, 42, 169, 163 | N5, N1, D4, A7 | $PU_{80}$ | Programmable |
| SDA4-1 | 54, 41, 170, 164 | P4, M1, C3, B6 | $PU_{80}$ | Programmable |
| **General-Purpose I/O (GPIO) Ports** | | | | |
| IOPA7-0 | 43, 40-36, 33-32 | N2, M2, L1, N3, M4, K1, L4, K2 | $PU_{80}$ | Programmable |
| IOPB7-0 | 165, 6-5, 164-162, 154-153 | D5, D1, C2, B6, A7, C5, C8, B9 | $PU_{80}$ | Programmable |
| IOPC7-0 | 1,176-175, 172-168 | A1, A2, B3, B4, A5, C3, D4 | $PU_{80}$ | Programmable |
| IOPD7-0 | 55-54, 42-41, 31-29, 26 | N5, P4, N1, M1, J1, L3, K4, K3 | $PU_{80}$ | Programmable |
| IOPE7,6,4 | 25-24, 2 | J4, H2, B2 | $PU_{80}$ | Programmable |
| IOPF7-0 | 119-114, 111-110 | G15, E13, F12, G14, H15, F13, G13, H13 | $PU_{80}$ | Programmable |
| IOPQ2-0 | 23, 22, 8 | J3, H3, E4 | $PU_{80}$ | Programmable |
| **Internal Keyboard Scan (KBC)** | | | | |
| KBSIN7-0 | 80-77, 74-71 | R10, P11, N12, M11, N11, M10, P9, R8 | $PU_{80}$ | Programmable |
| **PS/2 Interface** | | | | |
| PSCLK4-1 | 118,116, 114,110 | E13, G14, F13, H13 | $PU_{80}$ | Programmable |
| PSDAT4-1 | 119,117, 115,111 | G15, F12, H15, G13 | $PU_{80}$ | Programmable |
| **Strap Configuration** | | | | |
| BADDR1-0 | 127-126 | E15, C13 | $PD_{80}$ | Strap[1] |
| ENV1-0 | 125-124 | D12, F15 | $PD_{80}$ | Strap[1] |
| TRIS | 128 | D14 | $PD_{80}$ | Strap[1] |
| SHBM | 131 | C14 | $PD_{80}$ | Strap[1] |

1. Active only during $V_{CC}$ Power-Up reset.

# 2.0 Signal/Pin Description and Configuration (Continued)

## 2.3 STRAP PINS

During $V_{CC}$ Power-Up reset, the ENV(0-1), TRIS, SHBM and BADDR strap input signals are sampled. Internal pull-down resistors set these signals to 0. These resistors are active only during $V_{CC}$ Power-Up reset. An external 10 KΩ resistor connected to $V_{CC}$ may be used to set them to 1.

### Setting the Environment

ENV0 and ENV1 determine the operating environment. Table 4 shows the settings allowed. Pulling both ENV0 and ENV1 to 1 produces unpredictable results. In IRE and OBD environments, the TRIS strap input may be used for floating all the device signals. In other cases it should be kept low.

**Table 4. Environment Pin Settings**

| Environment | ENV0 | ENV1 | TRIS |
|---|---|---|---|
| IRE | 0 | 0 | 0[1] |
| OBD | 0 | 1 | 0[1] |
| DEV | 1 | 0 | 0 |

1. When set to 1, the PC87591L-N05 is put in TRI-STATE mode.

Figures 1 on page 24, 4 on page 30 and 5 on page 31 demonstrate how to configure the PC87591L-N05 for IRE, OBD and DEV environments, respectively, using the ENV0-1 signals.

### Other Strap Pin Settings

Table 5 provides brief descriptions of other strap inputs. For details on SHBM and TRIS, see Section 5.3 on page 262 and Section 4.20.4 on page 236, respectively.

**Table 5. Other Strap Pin Settings**

| Strap Pin | Internal Pull-Down (0) | External Pull-Up (1) |
|---|---|---|
| BADDR1-0 | SuperI/O Configuration Base Address; see Table 37 on page 297 | |
| SHBM | Disables shared memory with host BIOS | Enables shared memory with host BIOS |
| TRIS | Normal operation | While in IRE and OBD environments, causes PC87591L-N05 to float its output and I/O signals for system test purposes and clip-on ISE use |

### System Load on Strap Pins

The loads on the strap pins should not cause the voltage on them to drop below $V_{IH}$ when the pins should be high (1), or to rise above $V_{IL}$ when they should be low (0). See Section 7.3.2 on page 339.

If the load caused by the system on the strap pins exceeds 10 µA, use either an external pull-down resistor to keep the pin at 0 or a pull-up resistor with lower resistance to keep the pin at 1.

To reduce power consumption, in Idle mode, pins with strap inputs on them and a signal function other than GPIO are put in TRI-STATE or drive the strap-pin value. For pins with strap inputs that function as GPIO, it is recommended that the application drive the strap value as output to the value defined by the strap pin. For pins with strap and address line functionality, when the address configuration is enabled, the signal is driven by the hardware to its strap value on reset.

### Strap Pin Status Register (STRPST)

The STRPST register is a byte-wide, read-only register. It enables the software to read the value set to strap pins during Power-Up reset. STRPST bits provide the value of their respective strap input. See Table 5 for bit details.

Location: 00 $FF12_{16}$

Type: RO

Reset Value: According to external straps

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | **BADDR1** | **BADDR0** | **SHBM** |
| Reset | | | | | | | | |

# 2.0 Signal/Pin Description and Configuration (Continued)

## 2.4 ALTERNATE FUNCTIONS

### 2.4.1 GPIO with Alternate Functions

The PC87591L-N05 uses the GPIO port pins to multiplex functions, thereby maximizing the device's flexibility. Table 6 lists all the pins with such alternate functions and summarizes their selection criteria. To select alternate pin functions, strap inputs and configuration registers are used as follows:

- The ports' Alternate Function Control register controls the IOPA, IOPB, IOPC, IOPD, IOPE, IOPF and IOPQ pins. Each of the ports' pins may be used as a GPIO port or its alternate function, based on the setting of the corresponding bit in PxALT register.
- The environment setting and MCFG bits control port IOPJ, IOPL and IOPM pins. In IRE and OBD environments, these pins may function either as GPIO or their alternate function (as indicated in the table). In DEV environment, they all function as their alternate functions.
- Interrupts may be enabled together with the use of the pin as input or with its alternate function when the alternate function is an input. Enabling the interrupt input is done via the port's alternate function or the respective bit in EICFG register.
- Strap inputs and their associated internal pull-down resistor function during $V_{CC}$ Power-Up reset.
- The Protection Word registers configure the use of $\overline{\text{RESET2}}$ input.
- The Pin Multiplexing register (PNMR) configures the use of USART2, A19 and A20 signals.

When a pin is used as GPIO and not as its alternate function, disable the alternate function in the relevant module's register.

**Table 6. Alternate Function Selection**

| LQFP/ FBGA Pin/Ball | GPIO | | Alternate Function 1 | | | Alternate Function 2 | | |
|---|---|---|---|---|---|---|---|---|
| | Name | Type | Module | Signal Name | Select | Module | Signal Name | Select |
| 32/K2 | IOPA0 | I/O | PWM | PWM0 | PAALT.0 | | | |
| 33/L4 | IOPA1 | I/O | | PWM1 | PAALT.1 | | | |
| 36/K1 | IOPA2 | I/O | | PWM2 | PAALT.2 | | | |
| 37/M4 | IOPA3 | I/O | | PWM3 | PAALT.3 | | | |
| 38/N3 | IOPA4 | I/O | | PWM4 | PAALT.4 | | | |
| 39/L1 | IOPA5 | I/O | | PWM5 | PAALT.5 | | | |
| 40/M2 | IOPA6 | I/O | | PWM6 | PAALT.6 | | | |
| 43/N2 | IOPA7 | I/O | | PWM7 | PAALT.7 | | | |
| 153/B9 | IOPB0 | I/O | USART1 | URXD1 | PBALT.0 | | | |
| 154/C8 | IOPB1 | I/O | | UTXD1 | PBALT.1 | | | |
| 162/C5 | IOPB2 | I/O | | USCLK1 | PBALT.2 | | | |
| 163/A7 | IOPB3 | I/O | ACCESS.bus | SCL1 | PBALT.3 | | | |
| 164/B6 | IOPB4 | I/O | | SDA1 | PBALT.4 | | | |
| 5/C2 | IOPB5 | I/O | GA20 and $\overline{\text{KBRST}}$ | (GA20) | PBALT.5=0[1] | | | |
| 6/D1 | IOPB6 | I/O | | KBRST | PBALT.6 | | | |
| 165/D5 | IOPB7 | I/O | MSWC | $\overline{\text{RING}}$ | PBALT.7 | ICU | $\overline{\text{PFAIL}}$/ $\overline{\text{RESET2}}$[2] | PBALT.7/ PTWRH. RST2EN=01 |
| 169/D4 | IOPC1 | I/O | ACCESS.bus | SCL2 | PCALT.1 | | | |
| 170/C3 | IOPC2 | I/O | | SDA2 | PCALT.2 | | | |

## 2.0 Signal/Pin Description and Configuration (Continued)

**Table 6. Alternate Function Selection (Continued)**

| LQFP/ FBGA Pin/Ball | GPIO | | Alternate Function 1 | | | Alternate Function 2 | | |
|---|---|---|---|---|---|---|---|---|
| | Name | Type | Module | Signal Name | Select | Module | Signal Name | Select |
| 171/A5 | IOPC3 | I/O | Timers | TA1 | PCALT.3 | | | |
| 172/B4 | IOPC4 | I/O | | TB1 | PCALT.4 | MIWU | EXWINT22 | PCALT.4 |
| 175/B3 | IOPC5 | I/O | | TA2 | PCALT.5 | | | |
| 176/A2 | IOPC6 | I/O | | TB2 | PCALT.6 | MIWU | EXWNT23 | PCALT.6 |
| 1/A1 | IOPC7 | I/O | Clocks | CLKOUT | PCALT.7 & MCFG.CLKOM | | | |
| 26/K3 | IOPD0 | I/O | MSWC | $\overline{RI1}$ | PDALT.0 | MIWU | EXWINT20 | PDALT.0 |
| 29/K4 | IOPD1 | I/O | | $\overline{RI2}$ | PDALT.1 | | EXWINT21 | PDALT.1 |
| 30/L3 | IOPD2 | I/O | Host I/F | $\overline{RESET2}$ | PTWRH. RST2EN=10 | | EXWINT24 | PDALT.2 |
| 31/J1 | IOPD3 | I/O | | $\overline{ECSCI}$ | PDALT.3 | | | |
| 41/M1 | IOPD4 | I/O | ACCESS.bus | SDA3 | PDALT.4 | | | |
| 42/N1 | IOPD5 | I/O | | SCL3 | PDALT.5 | | | |
| 54/P4 | IOPD6 | I/O | ACCESS.bus | SDA4 | PDALT.6 | | | |
| 55/N5 | IOPD7 | I/O | | SCL4 | PDALT.7 | | | |
| 87/P13 | IOPE0 | I | ADC Voltage | AD4 | PEALT.0 | | | |
| 88/R14 | IOPE1 | I | | AD5 | PEALT.1 | | | |
| 89/R15 | IOPE2 | I | | AD6 | PEALT.2 | | | |
| 90/P14 | IOPE3 | I | | AD7 | PEALT.3 | | | |
| 2/B2 | IOPE4 | I | MIWU | SWIN | PEALT.4 | | | |
| 44/P1 | IOPE5 | I | Core Bus I/F | A20 | (DEV Env. or MCFG. ENEMEM=1) and PNMR.A20=1 | MIWU | EXWINT40 | PEALT.5 |
| 24/H2 | IOPE6 | I | LPC Interface | LPCPD | PEALT.6 | | EXWINT45 | EICFG. EXWINT45 |
| 25/J4 | IOPE7 | I | | CLKRUN | PEALT.7 | | EXWINT46 | EICFG. EXWINT46 |
| 110/H13 | IOPF0 | I/O | PS2 Interface | PSCLK1 | PFALT.0 | | | |
| 111/G13 | IOPF1 | I/O | | PSDAT1 | PFALT.1 | | | |
| 114/F13 | IOPF2 | I/O | | PSCLK2 | PFALT.2 | | | |
| 115/H15 | IOPF3 | I/O | | PSDAT2 | PFALT.3 | | | |
| 116/G14 | IOPF4 | I/O | | PSCLK3 | PFALT.4 | USART2 | URXD2 | PNMR. ENUSART2 |
| 117/F12 | IOPF5 | I/O | | PSDAT3 | PFALT.5 | | UTXD2 | |
| 118/E13 | IOPF6 | I/O | | PSCLK4 | PFALT.6 | | USCLK2 | |
| 119/G15 | IOPF7 | I/O | | PSDAT4 | PFALT.7 | | | |

## 2.0 Signal/Pin Description and Configuration (Continued)

**Table 6. Alternate Function Selection (Continued)**

| LQFP/ FBGA Pin/Ball | GPIO | | Alternate Function 1 | | | Alternate Function 2 | | |
|---|---|---|---|---|---|---|---|---|
| | Name | Type | Module | Signal Name | Select | Module | Signal Name | Select |
| 62/R7 | IOPJ2 | I/O | Development System Observability | BST0 | DEV Env. | | | |
| 63/N7 | IOPJ3 | I/O | | BST1 | | | | |
| 69/M9 | IOPJ4 | I/O | | BST2 | | | | |
| 70/N10 | IOPJ5 | I/O | | PFS | | | | |
| 75/R9 | IOPJ6 | I/O | | PLI | | | | |
| 76/P10 | IOPJ7 | I/O | | BRKL_RSTO | | | | |
| 103/K13 | IOPL3 | I/O | Core BIU | A19 | (DEV Env. or MCFG. ENEMEM=1) and PNMR.A19=1 | | | |
| 48/R3 | IOPL4 | I/O | | WR1 | DEV Env. or (MCFG. ENEMEM=1 and MCFG. EXMEM16=1) | | | |
| 148/D9 | IOPM0 | I/O | | D8 | | | | |
| 149/B10 | IOPM1 | I/O | | D9 | | | | |
| 155/C7 | IOPM2 | I/O | | D10 | | | | |
| 156/B8 | IOPM3 | I/O | | D11 | | | | |
| 3/B1 | IOPM4 | I/O | | D12 | | | | |
| 4/C1 | IOPM5 | I/O | | D13 | | | | |
| 27/H1 | IOPM6 | I/O | | D14 | | | | |
| 28/J2 | IOPM7 | I/O | | D15 | | | | |
| 8/E4 | IOPQ0 | I/O | Host I/F | $\overline{\text{LDRQ}}$ | PQALT.0 | | | |
| 22/H3 | IOPQ1 | I/O | | $\overline{\text{SMI}}$ | PQALT.1 | | | |
| 23/J3 | IOPQ2 | I/O | | $\overline{\text{PWUREQ}}$ | PQALT.2 | | | |
| 47/R2 | IOPQ3 | O | Clocks | CLK | PQALT.3 | | | |
| 68/P8 | | | Internal Keyboard Scan | KBSOUT15 | | Testing | XOR_OUT | BADDR0=1 and BADDR1=1 |

1. GA20 is implemented using the GPIO.
2. $\overline{\text{RESET2}}$ use is orthogonal to the other alternate functions and is determined by the setting in PTWRH register (see Page 55), although it is recommended to use it in conjunction with an interrupt on the same pin. See “Using RESET2 Input” on page 63.

# 2.0 Signal/Pin Description and Configuration (Continued)

## 2.4.2 System Configuration Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| MCFG | Module Configuration Register | R/W |
| EICFG | External Interrupts Configuration Register | R/W |
| IOEE1 and IOEE2 | Input to Output Echo Enable Register 1 and 2 | R/W |
| PTWRL | Protection Word Low Register | R/W or RO |
| PTWRH | Protection Word High Register | R/W or RO |
| PNMR | Pin Multiplexing Register | R/W |

### Module Configuration Register (MCFG)

The MCFG register is a read/write, byte-wide register. It is used for global system configuration and setup.

Write operations to the MCFG register should write zeros to all reserved bits. On reset, non-reserved bits of MCFG are cleared to 0. MCFG can be written in Active mode only. Its contents is preserved in Idle mode.

In IRE and OBD environments, all MCFG fields should be used to designate associated pins as GPIO ports or their alternate functions. In DEV environment, the pins are always allocated for development system use. The I/O ports functionality can be implemented using off-chip logic.

To guarantee binary and cycle-by-cycle compatibility among the different environments, define the MCFG fields as required for IRE and OBD even when in DEV environment, and use the I/O Expansion protocol to build an off-chip implementation of the I/O ports when they are used by the application.

ADBs or ISE systems use the MCFG Shadow (MCFGSH, write only) register to select the functionality of the signal that reaches the user’s application.

All write operations to the MCFG must be immediately followed by a write to the MCFG Shadow (MCFGSH) register. This register is part of the development system and is accessed by an access to the I/O zone ($\overline{\text{SELIO}}$).

MCFG is loaded with either $40_{16}$ or $C0_{16}$ on reset. See the description of GTMON for behavior of bit 7 during reset.

Software should load MCFGSH with the MCFG register’s value after reset.

The format of MCFG (and MCFGSH) is as follows:

MCFG Location: 00 $FF10_{16}$

MCFGSH Location: 00 $FBFE_{16}$

Type: MCFG is R/W; MCFGSH is WO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **GTMON** | **HOSTWAIT** | **ENZONE2** | **CLKOM** | | **EXMEM16** | **ENEMEM** | **ENEIO** |
| Reset | See text | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **ENEIO (Enable Expansion I/O).** This bit enables the use of the I/O Expansion protocol for expanding the amount of GPIO pins available to the application. In IRE and OBD environments, when ENEIO is cleared, the associated pins are used as GPIO signals. When set, enables the use of BIU I/O zone (SELIO) for the interface with off-chip logic. Use the BIU I/O zone configuration to select the access parameters to the I/O Expansion logic and its bus width. |
| 1 | **ENEMEM (Enable Expansion Memory).** This bit enables the use of the expansion memory for expanding the amount of memory available to the application to more than what is provided on chip. When cleared, the associated pins are used as GPIO signals. When set, enables the use of BIU zone 0 (SEL0) and the associated address and data lines for the interface with flash or SRAM devices. Use BIU zone 0 configuration to select the access parameters to the expansion memory and its width. |

## 2.0 Signal/Pin Description and Configuration (Continued)

| Bit | Description |
|---|---|
| 2 | **EXMEM16 (16-bit-Wide Expansion Memory).** This bit enables the use of the 16-bit-wide expansion memory, when the ENEMEM is set. The bus width indicated in this register should be the same as the bus width defined in zone 0 and zone 2 of the BIU (SEL0, SEL2).<br>0: The External Memory to be eight bits wide<br>1: Enables the use of a 16-bit-wide External Memory |
| 4-3 | **CLKOM (Clock Out Mode).** This field selects the mode of operation of the CLKOUT signal. This field is in effect only when this signal is set for output.<br>**Bits 4 3 Description**<br>0 0: Reserved (default)<br>0 1: CLK - a clock at the core frequency is driven out<br>1 0: 32.768 KHz clock output<br>1 1: Reserved |
| 5 | **ENZONE2 (Enable Expansion Memory Zone 2).** When set, this bit enables the use of BIU Zone 2 (SEL2) and the associated address (selected by bits 0-1 of the PTWRH register; see Page 55) and data lines for the interface with flash or SRAM devices. Use BIU Zone 2 configuration to select the access parameters to the expansion memory and its width. When cleared, the associated address range is used with Zone 0 (SEL0).<br>0: Zone 2 (SEL2) is disabled. Zone 0 (SEL0) is used for 00 $1000_{16}$ – 00 $DFFF_{16}$ Expansion Memory address range.<br>1: Zone 2 (SEL2) is enabled. The Expansion Memory address range is selected by bits 0-1 of the PTWRH register. |
| 6 | **HOSTWAIT (Host Interface WAIT state).** The host interface is in wait state after host domain power-up until the bit is cleared by software (Booter). LPC transactions to the device during this state are extended by a valid SYNC field (long wait state, LAD3:0 are 0110b). This bit is a sticky bit; it is set by core domain reset, and remains set until explicitly cleared by software by writing 1b. Writing 0b to the bit has no effect. On clearing the HOSTWAIT bit, the host interface is released from wait state. |
| 7 | **GTMON (Go-to Target Monitor Set Flag).** This bit is set to indicate that the code should jump to the beginning of the Target Monitor (TMON). The reset routine of the application should check this bit and behave accordingly.<br>GTMON bit is used by the Booter firmware,; therefore it should not be modified by the application firmware (EC BIOS). Once cleared, this bit can not be set by software. |

### External Interrupts Configuration Register (EICFG)

The EICFG register is a read/write, byte-wide register. It enables the use of some of the external interrupts. The EICFG bits have impact only when the pin is configured to its GPIO function. EICFG is cleared ($00_{16}$) on reset.

Note that some of the pins that have an external interrupt (not controlled by the EICFG register) have an alternate function that is an input. When a pin's alternate function is selected, its interrupt function is also enabled.

The format of EICFG is as follows:

Location: 00 $FF00_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | **EXWINT46** | **EXWINT45** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **EXWINT45 (Enable EXWINT45).**<br>0: EXWINT45 input is blocked (disabled) while not being read as a GPIO (default)<br>1: EXWINT45 input is open; this enables the detection of changes on the input to trigger interrupts |

## 2.0 Signal/Pin Description and Configuration (Continued)

| Bit | Description |
|---|---|
| 1 | **EXWINT46 (Enable EXWINT46).**<br>0: EXWINT46 input is blocked (disabled) while not being read as a GPIO (default)<br>1: EXWINT46 input is open; this enables the detection of changes on the input to trigger interrupts |
| 7-2 | **Reserved.** |

### Protection Word Low Register (PTWRL)

This register can be updated only when HOSTWAIT bit in MCFG register is 1. After HOSTWAIT bit is cleared, PTWRL register becomes read only.

PTWRL Location:00 $FF06_{16}$

Type: R/W or RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **Force MBTA Zero** | **RTC Lock Default** | **Host Boot Block** | **Core Boot Block** | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

| Bit | Description |
|---|---|
| 3-0 | **Core Boot Block.** This field defines the size of the Core Boot Block<br>The boot block starts at address 00 0000 and ends at the address specified by this field.<br>**Bits 3 2 1 0 Description**<br>1 1 1 1: When in IRE and OBD environments, the core is kept in reset<br>1 1 1 0: 4K (default)<br>Other: Reserved |
| 4 | **Host Boot Block.** This bit defines the use and the size of a Host Boot Block area. Protection is provided by the Shared Memory function.<br>0: A 64 Kbyte Host Boot Block is specified<br>1: No Host Boot Block is available (default) |
| 5 | **RTC Lock Default.** This bit defines the reset value of the Lock RTC Host Access (LKRTCHA) bit in the Lock SuperI/O Host Access register (LKSIOHA). This enables preventing host access to the RTC at any time when the RTC is used by security applications running on the core.<br>0: RTC Lock is disabled on reset (LKRTCHA=0)<br>1: RTC Lock is enabled on reset (LKRTCHA=1) (default) |
| 6 | **Force MBTA Zero.** This bit controls the setting of the host boot block location. See “Shared Memory Core Top Address Register (SMCTA)” on page 272.<br>When expansion memory is used for a shared BIOS implementation perform the following:<br>• Set the Force MBTA Zero bit to 1<br>• Set the Host Boot Block bit to 1<br>• Locate the host boot block in the boot block of the flash device, used as expansion memory<br>0: MBSD field in SMCTA register is reset to the implemented on-chip ROM size<br>1: MBSD field in SMCTA register is reset to $0000_{16}$ (default) |
| 7 | **Reserved.** |

## 2.0 Signal/Pin Description and Configuration (Continued)

### Protection Word High Register (PTWRH)

This register can be updated only when HOSTWAIT bit in MCFG register is 1. After HOSTWAIT bit is cleared, PTWRH register becomes read only.

PTWRH Location:00 $FF08_{16}$

Type: R/W or RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **RAM Size** | **RST2EN** | | **Reserved** | | | **Zone 2 Memory Range** | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

| Bit | Description |
|---|---|
| 1-0 | **Zone 2 Memory Range (ZONE2MAP).** This field selects the Expansion Memory zone 2 address range.<br>**Bits 1 0 Description**<br>0 0: 64K zone2 memory address range enabled (default):<br>Zone 2: 00 $1000_{16}$ – 00 $DFFF_{16}$<br>Zone 0: 01 $0000_{16}$ – 1F $FFFF_{16}$<br>0 1: 128K zone2 memory address range enabled:<br>Zone 2: 00 $1000_{16}$ – 00 $DFFF_{16}$<br>Zone 2: 01 $0000_{16}$ – 01 $FFFF_{16}$<br>Zone 0: 02 $0000_{16}$ – 1F $FFFF_{16}$<br>1 0: 192K zone2 memory address range enabled:<br>Zone 2: 00 $1000_{16}$ – 00 $DFFF_{16}$<br>Zone 2: 01 $0000_{16}$ – 02 $FFFF_{16}$<br>Zone 0: 03 $0000_{16}$ – 1F $FFFF_{16}$<br>1 1: 256K zone2 memory address range enabled:<br>Zone 2: 00 $1000_{16}$ – 00 $DFFF_{16}$<br>Zone 2: 01 $0000_{16}$ – 03 $FFFF_{16}$<br>Zone 0: 04 $0000_{16}$ – 1F $FFFF_{16}$ |
| 4-2 | **Reserved.** |
| 6-5 | **RST2EN ($\overline{\text{RESET2}}$ Enable).** This field controls the use of the $\overline{\text{RESET2}}$ alternate function. The following options are supported in the PC87591L-N05:<br>**Bits 6 5 Description**<br>0 0: Reserved<br>0 1: $\overline{\text{RESET2}}$ is enabled on the IOPB7/$\overline{\text{RING}}$/$\overline{\text{PFAIL}}$/$\overline{\text{RESET2}}$ pin<br>1 0: $\overline{\text{RESET2}}$ is enabled on the IOPD2/EXWINT24/$\overline{\text{RESET2}}$ pin<br>1 1: $\overline{\text{RESET2}}$ input is disabled, and $\overline{\text{RESET2}}$ events will not be generated (default) |
| 7 | **RAM size.** This bit controls the internal RAM size:<br>0: 2K of the internal RAM is accessible in 00 $E800_{16}$ – 00 $F7FF_{16}$<br>1: 4K of the internal RAM is accessible in 00 $E000_{16}$ – 00 $F7FF_{16}$ (default) |

### Pin Multiplexing Register (PNMR)

PNMR Location: 00 $FF0A_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | **A20** | **A19** | **ENUSART2** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Bit | Description |
|---|---|
| 0 | **ENUSART2 (Enable USART2).** This bit selects either the USART2 or PS/2 Channels 3 and 4 to device pins. When set, this bit enables the use of USART2.<br>0: PS/2 Channels 3 and 4 or IOPF4-6 signals are selected, according to PFALT4-6 bits (default)<br>1: USART2 signals are selected |
| 1 | **A19 (Enable A19).** This bit selects between either A19 or IOPL3 to the device pin. When set to 1, A19 bit enables the use of A19.<br>0: IOPL3 signal is selected<br>1: A19 signal is selected (default) |
| 2 | **A20 (Enable A20).** This bit selects either A20 or IOPE5/EXWINT40 to the device pin. When set to 1, A20 bit enables the use of A20.<br>0: IOPE5/EXWINT40 signal is selected, according to PEALT5 bit (default)<br>1: A20 signal is selected |
| 7-3 | **Reserved.** |

### 2.4.3 GPIO with Echo Configuration

Some of GPIO pins may be paired to generate an input to output echoing with software masking. The input GPIO are input ports (Px or Py) with a wake-up function. The output ports are a Px type enhanced with the echo mechanism, as described in Section 4.5.2 on page 111. The input echoing is enabled only when the respective Echo Enable bit in IOEE1-2 registers is set. The input port should be configured to enable the interrupt function.

**Table 7. GPIO Echo Functions Routing and Echo Enable Bit Assignments**

| Output Port | Echo Enable Bit | Input Port |
|---|---|---|
| IOPA0 | EEPA0 bit in IOEE1 register | IOPC4/EXWINT22 |
| IOPA1 | EEPA1 bit in IOEE1 register | IOPE6/$\overline{\text{LPCPD}}$/EXWINT45 |
| IOPA2 | EEPA2 bit in IOEE1 register | IOPE7/$\overline{\text{CLKRUN}}$/EXWINT46 |
| IOPA3 | EEPA3 bit in IOEE1 register | IOPF6/PSCLK4 |
| IOPA4 | EEPA4 bit in IOEE1 register | IOPF7/PSDAT4 |
| IOPC0 | EEPC0 bit in IOEE2 register | IOPE4/SWIN |
| IOPB0 | EEPB0 bit in IOEE2 register | IOPD0/EXWINT20 |
| IOPB1 | EEPB1 bit in IOEE2 register | IOPD1/EXWINT21 |
| IOPB2 | EEPB2 bit in IOEE2 register | IOPC6/EXWINT23 |
| IOPD3 | EEPD3 bit in IOEE2 register | IOPD2/EXWINT24 |

#### Input to Output Echo Enable Register 1 and 2 (IOEE1 and IOEE2)

The IOEE1 and IOEE2 registers are read/write, byte-wide registers. They enable the echoing of some of the General-Purpose Input ports, equipped with wake-up, to a specified output. The IOEE bits have an impact only when the output GPIO is configured to its GPIO function. IOEE1 and IOEE2 are cleared ($00_{16}$) on reset.

The format of IOEE1 is:

IOEE1 Location: 00 $FF02_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | **EEPA4** | **EEPA3** | **EEPA2** | **EEPA1** | **EEPA0** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 2.0 Signal/Pin Description and Configuration (Continued)

| Bit | Description |
|---|---|
| 0 | **EEPA0.**<br>0: Echo Disabled (default)<br>1: Echo Enabled |
| 1 | **EEPA1.**<br>0: Echo Disabled (default)<br>1: Echo Enabled |
| 2 | **EEPA2.**<br>0: Echo Disabled (default)<br>1: Echo Enabled |
| 3 | **EEPA3.**<br>0: Echo Disabled (default)<br>1: Echo Enabled |
| 4 | **EEPA4.**<br>0: Echo Disabled (default)<br>1: Echo Enabled |
| 7-5 | **Reserved.** |

The format of IOEE2 is:

IOEE2 Location: 00 $FF04_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | **EEPC0** | **EEPD3** | **EEPB2** | **EEPB1** | **EEPB0** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **EEPB0.**<br>0: Echo Disabled (default)<br>1: Echo Enabled |
| 1 | **EEPB1.**<br>0: Echo Disabled (default)<br>1: Echo Enabled |
| 2 | **EEPB2.**<br>0: Echo Disabled (default)<br>1: Echo Enabled |
| 3 | **EEPD3.**<br>0: Echo Disabled (default)<br>1: Echo Enabled |
| 4 | **EEPC0.** |
| 7-5 | **Reserved.** |

See Table 7 on page 56 for the assignment of the Echo Enable bits to GPIO pairs.

# 3.0 Power, Reset and Clocks

## 3.1 POWER

### 3.1.1 Power Planes

The PC87591L-N05 has four power planes (wells), as shown in Table 8.

**Table 8. PC87591L-N05 Power Planes**

| Power Plane | Description | Power Pins | Ground Pins |
|---|---|---|---|
| Host Domain | Powers the LPC interface and Host Controlled Functions (except for RTC and MSWC) and some external signals[1] | $V_{DD}$ | GND |
| Suspend | Powers the core domain, the host-core interface, the MSWC and their external signals[1] | $V_{CC}$ | GND |
| Analog | Powers some internal analog circuits and their external signals[1] | $AV_{CC}$ | AGND |
| Core | Powers the internal (core) logic of all the device modules | $V_{CORF}$[2] | GND |
| Battery Backed | Powers the RTC, some MSWC registers, the 32.768 KHz clock/crystal oscillator signals and some other storage elements that should be preserved at all times[1] | $V_{PP}$[3] | GND |

1. See the tables in Section 2.2.1 on page 38 to Section 2.2.13, specifically the Power Well column, for how the PC87591L-N05 external interface signals are assigned to various power planes.
2. $V_{CORF}$ is generated from $V_{CC}$ by an on-chip power converter.
3. $V_{PP}$ is an internal power signal derived from $V_{CC}$ or $V_{BAT.}$ $V_{PP}$ is taken from $V_{CC}$ if it is greater than the minimum value defined; otherwise, $V_{BAT}$ is used as the $V_{PP}$ source. For more details on the switching between them, refer to Section 6.2.9 on page 322.

For correct PC87591L-N05 operation, $V_{CC}$ must be applied whenever $V_{DD}$ is applied. Apply $AV_{CC}$ at the same time that $V_{CC}$ is applied, protection is provided against rise-time differences only.

### 3.1.2 Power States

The PC87591L-N05 has three main power states:

- **Battery Fail**
  Host Domain, Suspend, Analog and Battery Backed power planes are all powered off (i.e., in this case $V_{DD}$, $V_{CC}$, $AV_{CC}$ and $V_{BAT}$ are all inactive).
- **Power Fail**
  Host Domain, Suspend and Analog power planes are powered off. Battery Backed power plane is powered on (i.e., in this case $V_{DD}$, $V_{CC}$ and $AV_{CC}$ are inactive; $V_{BAT}$ is active).
- **Power Applied**
  Suspend, Analog and Battery Backed power planes are powered on, Host Domain power plane may be on or off (i.e., in this case $V_{CC}$, $AV_{CC}$ are active, $V_{BAT}$ and $V_{DD}$ may be active or inactive).
  The Power Applied state has several sub-states, depending on the domain:
  — The host domain has Host Power On and Host Power Off states, according to the $V_{DD}$ status;
  — The core domain and host-core interface have Active and Idle states. For details, see Section 4.17 on page 208.

The following power states are illegal:

- Host domain on and Suspend and/or Analog off (i.e., $V_{DD}$ active and $V_{CC}$ and/or $AV_{CC}$ are inactive).
- Suspend on and Analog off (i.e., $V_{CC}$ active and $AV_{CC}$ inactive) and vice-versa.

## 3.0 Power, Reset and Clocks (Continued)

Table 9 summarizes the power states related to the PC87591L-N05 power planes.

**Table 9. PC87591L-N05 Power States and Related Power Planes**

| Power State | | Host Domain ($V_{DD}$) | Suspend ($V_{CC}$) | Analog ($AV_{CC}$) | Battery Backed ($V_{PP}$) | $V_{BAT}$ |
|---|---|---|---|---|---|---|
| Battery Fail | | - | - | - | - | - |
| Power Fail | | - | - | - | + | + |
| Power Applied | Active and Idle | x | + | + | + | x |
| | Host Power Off | - | + | + | + | x |
| | Host Power On | + | + | + | + | x |
| Illegal[1] | | + | - | - | x | x |
| | | x[2] | + | - | x | x |
| | | x | - | + | x | x |

1. Operation is not guaranteed.
2. X = Don't care.

Figure 8 shows the power state transitions.

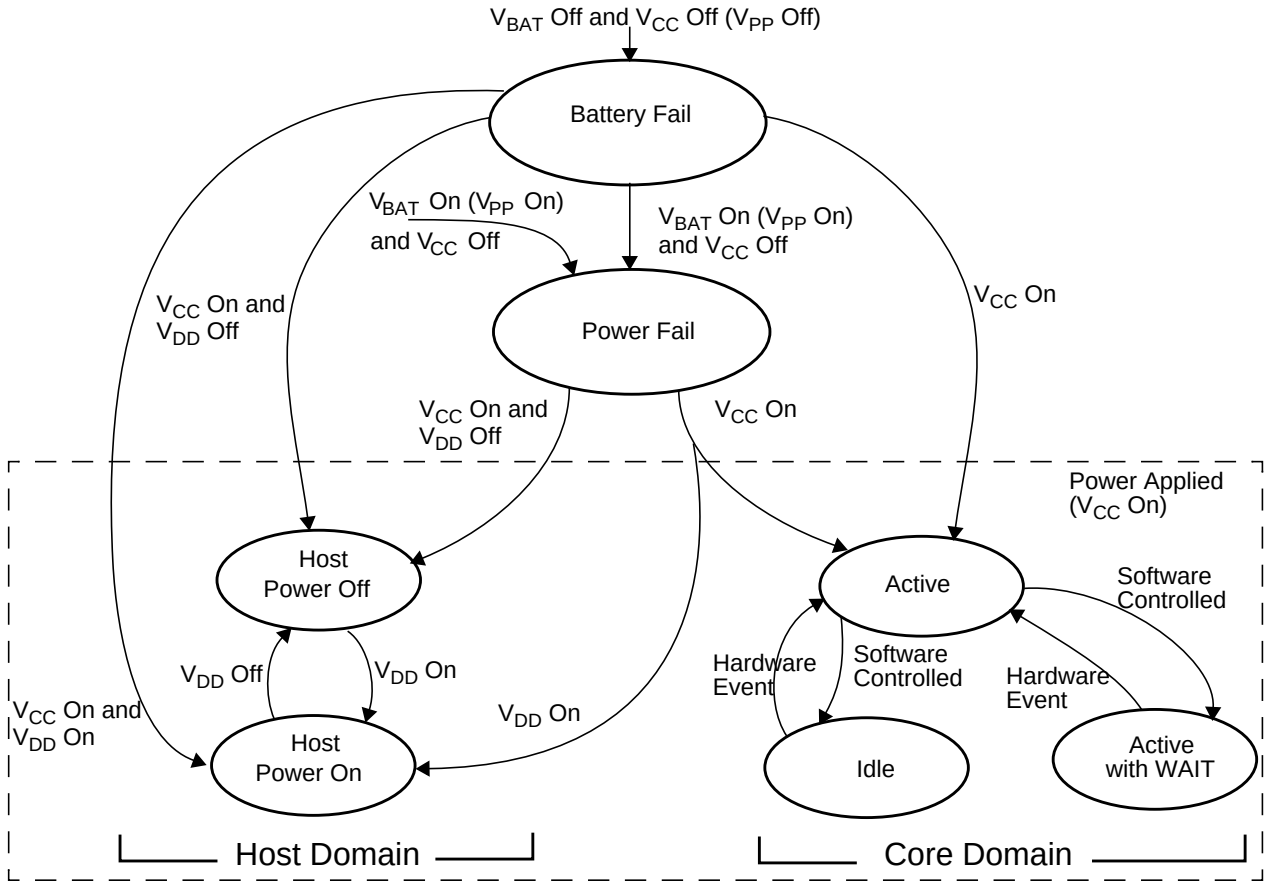



**Figure 8. Power State Transitions**

### 3.1.3 Power Connection and Layout Guidelines

The PC87591L-N05 requires a power supply voltage of 3.3V ± 10% for all the digital supplies ($V_{CC}$ and $V_{DD}$). The analog power supply requires a voltage of 3.3V ± 10% or 3.3V ± 5%. A 3.3V ± 5% analog supply is recommended since it improves the accuracy of measuring input voltage values close to the full-scale (see Section 7.4.1 on page 340).

Both $AV_{CC}$ and $V_{CC}$ should be applied for correct operation; and although the PC87591L-N05 is protected against damage if operated with only one of them ($AV_{CC}$ or $V_{CC}$), the ADC and DAC modules do not function correctly in such a case and may cause current leakage. Therefore, ADC and DAC modules should be left at their reset condition (disabled) or disabled

## 3.0 Power, Reset and Clocks (Continued)

by firmware until $AV_{CC}$ is within the limits specified above. In addition, the $AV_{CC}$ is internally isolated from $V_{CC}$ to enable $AV_{CC}$ to be externally filtered or driven by a low-noise power supply.

The PC87591L-N05 is designed to operate with a Lithium backup battery that can supply voltage up to 3.6V. The PC87591L-N05 includes an internal current-limiting resistor on the $V_{BAT}$ input as required to meet UL regulations.

$V_{DD}$, $V_{CC}$ and $V_{BAT}$ use a common ground return, named digital ground and marked GND. The analog circuits supplied by $AV_{CC}$ use a separate ground return, named analog ground and marked AGND. This ensures effective isolation of the analog modules from noise caused by the digital modules.

The following directives are recommended for the PC87591L-N05 power and ground connections (see Figure 9 for the power supply connections):

### Ground Connection

Use two ground planes, one for the digital signals (GND) and one for the analog signals (AGND). The following ground connections should also be made:

- The analog ground plane (AGND) should be connected at only one point to the digital ground plane (GND). This point should be located physically near the PC87591L-N05.
- The analog ground return pin of the PC87591L-N05 (AGND) should be connected to the analog ground plane.
- The decoupling capacitors of the analog supply ($AV_{CC}$) pin should be connected to the analog ground plane as close as possible to the AGND pin.
- The ground reference of the voltage input signals to the ADC module ($V_{IN}$0-9 in Figure 9) should be connected to the AGND plane close to the PC87591L-N05.
- The ground reference of the voltage output signals from the DAC module ($V_{OUT}$0-3 in Figure 9) should be connected to the AGND plane close to the PC87591L-N05.
- All GND pins of the PC87591L-N05 must be connected to the GND plane.
- The decoupling capacitors of the Suspend digital supply ($V_{CC}$) pin should be located near each $V_{CC}$-GND pin pair; one side of each capacitor should be connected to the ground plane.
- The ground reference of the voltage input signals to the ACM module ($KBV_{IN}$0-7 in Figure 9) should be connected to the GND plane close to the PC87591L-N05.
- The decoupling capacitor of the on-chip power converter ($V_{CORF}$) pin should be located near the PC87591L-N05; one of the capacitor's sides should be connected to the ground plane.
- The decoupling capacitors of the Host digital supply ($V_{DD}$) pin should be located near the $V_{DD}$-GND pair; one side of each capacitor should be connected to the ground plane.
- The ground return and the decoupling capacitor of the backup battery supply ($V_{BAT}$) pin should be located near the PC87591L-N05; one of the capacitor's sides should be connected to the ground plane.

Note that low-impedance ground layers improve noise isolation and reduce ground bounce problems.

### Power Connection

All PC87591L-N05 supply pins must be connected to the appropriate power plane, and decoupling capacitors must be used as recommended below.

The analog supply pin, $AV_{CC}$, should be connected to a low-noise power supply that has the same voltage as the digital supply (3.3V). If the $AV_{CC}$ pin is connected to the same power supply as the $V_{CC}$ pin, it is recommended to use an external LC or RC filter for the $AV_{CC}$ pin. An example of an LC filter [$L_1$ and ($C_1$+$C_2$)] is shown in Figure 9. An RC filter is obtained by replacing $L_1$ (in Figure 9) with a 10Ω resistor (not shown in the figure).

**Note:** If VBAT pins are not used, connect the pins to GND.

### Decoupling Capacitors

The following decoupling capacitors should be used to reduce power supply deeps, ground bounce and EMI (refer to Figure 9 for the position of capacitors, e.g., C1, C2, etc.):

- Suspend digital supply ($V_{CC}$): Place one 0.1 µF capacitor (C3) on *each* $V_{CC}$-GND pin pair as close as possible to the pin pair. Also, place one 10–47 µF $C_4$ tantalum capacitor on the common net as close as possible to the chip.
- On-chip power converter ($V_{CORF}$): Place one 1 µF ceramic capacitor (C8) on the $V_{CORF}$ pin pair as close as possible to the pin.
- Host digital supply ($V_{DD}$): Place one 0.1 µF capacitor (C5) on the $V_{DD}$-GND pin pair as close as possible to the pin pair. Also, place one 10-47 µF tantalum capacitor (C6) on the common net as close as possible to the chip.
- Backup battery ($V_{BAT}$): Place one 0.1 µF capacitor (C7) on the $V_{BAT}$ pin as close as possible to the pin.
- Analog supply ($AV_{CC}$): Place a 0.1 µF capacitor (C2) and a 10-47 µF tantalum capacitor (C1) on the $AV_{CC}$ pin as close as possible to the pin.

## 3.0 Power, Reset and Clocks (Continued)

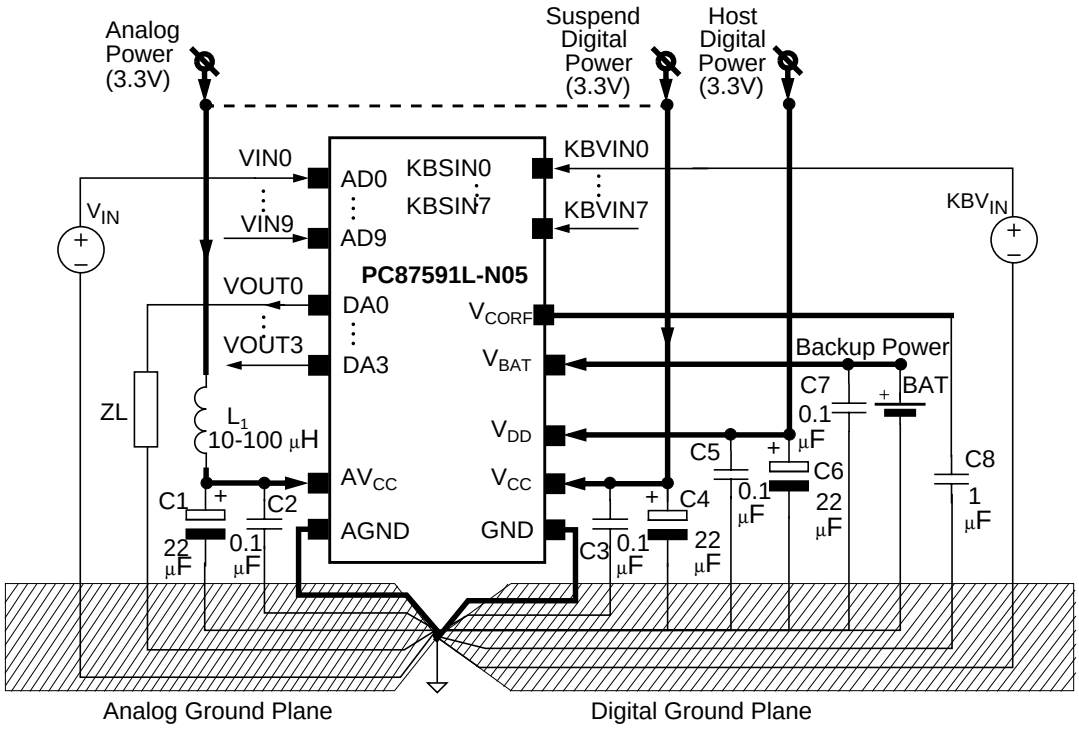



**Figure 9. PC87591L-N05 Power Supply Connection**

### 3.2 RESET SOURCES AND TYPES

The PC87591L-N05 has several input reset types:

- $V_{PP}$ Power-Up reset (for $V_{PP}$ supplied functions only)
  Activated when either $V_{CC}$ or $V_{BAT}$ is powered up after both have been off.
- $V_{CC}$ Power-Up reset
  Activated when $V_{CC}$ is powered up.
- Warm reset
  Activated on $\overline{\text{RESET1}}$ input falling edge.
- Watchdog and Debugger Interface resets:
  - — Watchdog reset
    Activated on request from the TWD module (watchdog signal is asserted); see Section 4.10 on page 160.
  - — Debugger Interface reset
    Activated on request from the Debugger Interface module used during debug and flash updates; see Section 4.19 on page 221.
- Host Domain reset
  This is divided into two sub-groups: Host Domain Hardware and Host Domain Software reset. Combinations consisting of the flowing events are used to trigger these reset operations:
  - — $\overline{\text{RESET1}}$ active
  - — When $V_{DD}$ is active, $\overline{\text{RESET2}}$ is active
  - — On $V_{DD}$ power-up
  - — A Software reset triggered by a write of 1 to bit 1 of SIOCF1 register in the SuperI/O Configuration registers; see Section 6.1.8 on page 306

## 3.0 Power, Reset and Clocks (Continued)

Unless otherwise noted, reset references throughout the PC87591L-N05 modules default to the following types:

- For $V_{PP}$ retained functions: $V_{PP}$ Power-Up reset
- For core domain functions and host-core interface functions: $V_{CC}$ Power-Up reset, Warm reset, Watchdog reset, Debugger Interface reset and Software reset
- For host domain functions: Host Domain reset

In DEV environment, the PC87591L-N05 outputs to the $\overline{\text{BRKL\_RSTO}}$ signal an indication that a reset occurred at the core domain. See Section 4.20.3 on page 236 for the implementation and usage of $\overline{\text{RSTO}}$.

The following sections detail the sources and effects of the various resets on the PC87591L-N05, per reset type.

### 3.2.1 $V_{PP}$ Power-Up Reset

$V_{PP}$ is an internal power signal derived from $V_{CC}$ and $V_{BAT}$. $V_{PP}$ Power-Up reset is generated by an internal circuit that detects the status of the $V_{PP}$ power. $V_{PP}$ Power-Up reset signal is active from the rising of $V_{PP}$ until the $V_{PP}$ power is detected as “good” (i.e., $V_{PP}$ is above $V_{BATDTC}$). When active, this signal resets all registers whose values are retained by $V_{PP}$.

For more details, see Section 6.2.9 on page 322.

### 3.2.2 $V_{CC}$ Power-Up Reset

$V_{CC}$ Power-Up reset is generated by an internal circuit. The PC87591L-N05 performs a $V_{CC}$ Power-Up reset when $V_{CC}$ power is applied. This reset is completed $t_{IRST}$ after the internal clocks have stabilized (see Section 7.6.2 on page 345).

If the 32 KHz crystal is disabled before $V_{CC}$ power-up, external devices should wait at least $t_{32KW}$ before accessing the PC87591L-N05. Any host processor access during this time results in:

- The host processor is stalled (by driving a “Long WAIT sync” response on the LPC bus) until after the reset process is completed (after the HOSTWAIT bit in MCFG register is set to 1 by the Booter firmware) and the bus request can be performed.
- If HRAPU bit in MSWCTL1 register is set (1), the host processor is reset by asserting $\overline{\text{KBRST}}$ until the internal reset is completed.

On $V_{CC}$ Power-Up reset, the PC87591L-N05 responds as follows:

- Enables the 32 KHz crystal, if it is disabled.
- Resets the High-Frequency Clock Generator (HFCG) to its default frequency.
- Loads default values to all registers whose values are retained by $V_{CC}$.
- Puts pins with strap options into TRI-STATE and enables the internal pull-downs on the strap pins.
- Samples the values of the strap pins.
- Resets the TAP controller of the Debugger Interface module.
- Resets the MSWC, excluding those MSWC registers whose values are retained by $V_{PP}$.
- Resets Port PC0.
- Carries out all the Warm reset actions (see below).

### 3.2.3 Watchdog Reset and Debugger Interface Reset

The PC87591L-N05 generates a Watchdog reset on request from the TWD module (i.e., a watchdog signal is asserted). It generates a Debugger Interface reset on request from the Debugger Interface module (reset command). During these resets, the PC87591L-N05 performs the $V_{CC}$ Power-Up reset actions, with the following exceptions:

- The PC87591L-N05 does not sample the value of any strap pin; instead, it maintains the configuration determined by the strap pins at $V_{CC}$ Power-Up reset.
- It does not reset the TAP controller.
- On Debugger I/F, reset PC0 is not reset (it is reset on Watchdog reset).
- It resets the HFCG to its default frequency.
- Some MSWC registers do not reset on Watchdog or Debugger I/F reset.

The reset periods are identical to the $V_{CC}$ Power-Up reset period.

## 3.0 Power, Reset and Clocks (Continued)

### 3.2.4 Warm Reset

Warm reset is activated on the falling edge of $\overline{\text{RESET1}}$ input. If Warm reset and $V_{CC}$ Power-Up reset occur at the same time, $V_{CC}$ power-up takes precedence.

During a Warm reset, the PC87591L-N05 responds as follows:

- Terminates core executed instructions
- Discards results not yet written to memory
- Eliminates any pending core interrupts and traps
- Clears the internal latch for the core domain's edge-sensitive external interrupt
- Deactivates the external bus control signals $\overline{\text{WR(0-1)}}$, $\overline{\text{SEL(0-2)}}$, $\overline{\text{SELIO}}$, $\overline{\text{RD}}$ and BST(0-2)
- Puts the address A(0-20) and data D(0-15) buses in TRI-STATE
- Switches to core domain's Active mode
- Loads default values into registers, with the exception of:
  - The Mobile System Wake-Up Control (MSWC) and RTC registers retained by $V_{PP}$
  - Some registers that are specifically noted to be cleared only by other events
- Resets the Debugger interface, except for TAP controller

During Warm reset, strap pins are not sampled and the configuration determined at $V_{CC}$ power-up is unaffected by subsequent Warm resets.

The PC87591L-N05 core domain and some parts of the host-core interface functions can operate when $\overline{\text{RESET1}}$ is still asserted (low). Some parts of the Host Domain Functions that are reset by the Host Domain reset (see following section) are kept reset as long as RESET1 is asserted.

### 3.2.5 Host Domain Reset

#### Using $\overline{\text{RESET2}}$ Input

The $\overline{\text{RESET2}}$ input signal is enabled as an alternate function on one of two pins. The RST2EN field of the "Protection Word High Register (PTWRH)" (see Page 55) is used to define whether the $\overline{\text{RESET2}}$ input is enabled and, if so, on which of the two pins.

Note that enabling $\overline{\text{RESET2}}$ input on either of the signals causes that pin to be used as an input GPIO with an interrupt input associated with it. It is recommended to use this interrupt function to interrupt the core when a $\overline{\text{RESET2}}$ event occurs and to use the interrupt routing to stop activities and resume default values to some system elements not directly reset by the event.

#### Host Domain Reset Actions

The reset actions on the host domain are broken into two categories that depend on the reset event source; some of the actions may happen for both these reset source types and thus are named Host Domain Reset.

**Host Domain Hardware Reset:** While $\overline{\text{RESET1}}$ is active, **or** $\overline{\text{RESET2}}$ is active **and** $V_{DD}$ is on or after $V_{DD}$ power-up.

Host Domain Hardware reset performs the following actions:

- It brings the LPC interface state machine to its inactive state.
- It resets all SuperI/O configuration registers except for those that are battery-backed (see Section 6.1.8 on page 306).
- It resets Shared Memory Host Controlled registers

**Host Domain Software Reset:** This reset is triggered by a write of 1 to bit 1 in SIOCF1 register in the SuperI/O Configuration registers.

Host Domain Software reset performs the following actions:

- It resets all SuperI/O configuration registers except those noted to be protected from software reset (i.e., bits that are locked from write accesses).
- It resets Mobile System Wake-Up Control (MSWC) bits and RTC bits marked to be reset by software.

Note that lock bits and memory protect bits are not reset by software reset; instead, they require a hardware reset to unlock them; this requirement protects these bits from any faulty or malicious software.

For more details, see Section 6.1.3 on page 303.

**Host Domain Reset:** This term is used when a bit is reset by either a Host Domain Hardware reset or Host Domain Software reset.

# 3.0 Power, Reset and Clocks (Continued)

## 3.3 CLOCK DOMAINS

The PC87591L-N05 has three clock domains, as shown in Table 10.

**Table 10. PC87591L-N05 Clock Domains**

| Clock Domain | Frequency | Source | Usage |
|---|---|---|---|
| **Core** | See Section 4.18 on page 212 | HFCG | Core domain and host-core interface |
| **LPC** | Up to 33 MHz | LPC clock input | LPC bus interface |
| **RTC** | 32 KHz | Clock input or on-chip oscillator[1] | RTC, HFCG, TWD, PMC, ACM |

1. See Section 6.2.3 on page 318 and Section 6.2.4 on page 319.

### Core Domain Clock

The core clock is sourced by the HFCG. The following section gives an overview of the clock domain.

On $V_{CC}$ Power-Up reset, the HFCG is set to generate a 4 MHz clock. The HFCG frequency can be modified by core firmware. See Section 4.18 on page 212.

On Watchdog reset (or Debugger interface reset), the HFCG is reset.

### LPC Clock

The LPC interface is driven by the LCLK input. LCLK frequency can be up to 33 MHz. The clock $\overline{\text{CLKRUN}}$ signal may be used as part of a power management scheme that slows down the clock; see “CLKRUN Functionality” on page 305.

### RTC (32 KHz) Clock

The RTC clock is the reference for the generation of all other clocks in the PC87591L-N05. The clock is enabled at $V_{CC}$ power-up and is kept active while $V_{PP}$ is active. Keeping the RTC clock active significantly reduces the PC87591L-N05 wake-up time. The RTC clock is used for time keeping in the RTC; it is also fed into the Power Management Controller (PMC), which provides the clock to other functions (such as TWD and ACM) that are active at all times (including Idle).

## 3.4 TESTABILITY SUPPORT

The PC87591L-N05 supports two testing techniques:

- In-Circuit Testing (ICT)
- XOR-Tree Testing

### 3.4.1 ICT

The In-Circuit Testing (ICT) technique, also known as “bed-of-nails”, injects logic patterns to the input pins of the devices mounted on the tested board. It then checks their outputs for the correct logic levels.

The PC87591L-N05 supports this testing technique by floating (TRI-STATE) all the device pins. This avoids back-driving the PC87591L-N05 pins by the ICT tester when a device normally controlled by PC87591L-N05 is tested (device inputs are driven by the ICT tester).

To enter TRI-STATE mode, the TRIS pin must be pulled up (by a 10 KΩ resistor to $V_{CC}$) in IRE and OBD environments, after $V_{CC}$ power supply is turned on. After the internal reset (see Section 7.6.2 on page 345) is completed, all the device output and I/O pins are floated (TRI-STATE). An exception to this are the power supply pins ($AV_{CC}$, $V_{CC}$, $V_{DD}$, $V_{CORF}$, $V_{BAT}$, AGND, GND), the DAC output pins (DA3-0) and the 32KX2 pin, which do not float in TRI-STATE mode.

### 3.4.2 XOR-Tree Testing

The PC87591L-N05 device mounted on the board can be tested using the XOR-Tree technique. This test also checks the correct connection of the device pins to the board.

To enter XOR-Tree mode, BADDR0 and BADDR1 pins must be pulled up (by a 10 KΩ resistor to $V_{CC}$) and the TRIS pin must be left unconnected after $V_{CC}$ power supply is turned on. After the internal reset (see Section 7.6.2 on page 345) is completed, the device pins (including BADDR0-1 and TRIS pins) are connected in a XOR-Tree configuration and are isolated from the internal PC87591L-N05 functions.

## 3.0 Power, Reset and Clocks (Continued)

In XOR-Tree mode, all PC87591L-N05 device pins are configured as inputs, except the last pin in the tree, which is the XOR_OUT output. The buffer type of the input pins participating in the XOR-Tree is $IN_T$ (Input, TTL compatible), regardless of the buffer type of these pins in normal device operation mode (see Section 2.2 on page 38). The input pins are chained through XOR gates, as shown in Figure 10. The power supply pins ($AV_{CC}$, $V_{CC}$, $V_{DD}$, $V_{CORF}$, $V_{BAT}$, AGND, GND), Analog pins (AD9-0, DA3-0) and Crystal Oscillator pins (32KX1, 32KX2) are excluded from the XOR tree.

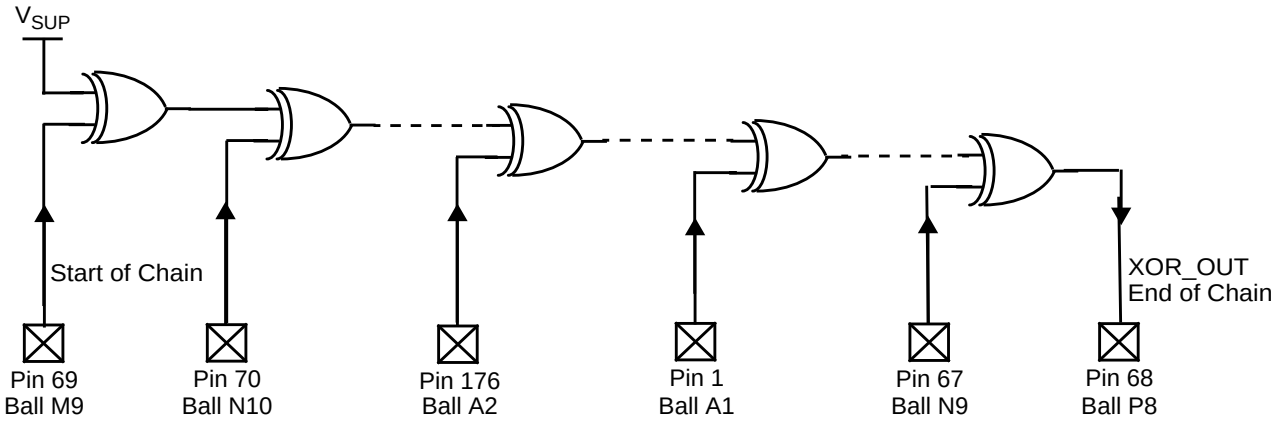



**Figure 10. XOR-Tree Chain (Simplified Diagram)**

In the 176-pin LQFP package, the XOR-Tree chain starts with pin 69 (see Figure 10), continues with pins 70 (the next pin in ascending order) through 176, goes to pin 1, and ends with XOR_OUT pin (68). In the 176-pin FBGA package, the XOR-Tree chain starts with ball M9 (see Figure 10) and ends with ball P8. For a detailed description of the XOR-Tree chain for both packages, see Table 11 on page 66. In the table, the chain direction is from top to bottom, and from left to right.

For correct XOR-Tree operation, **all** the XOR-Tree inputs must be set to a valid logic level (i.e., either below $V_{IL}$, or above $V_{IH}$; see Section 7.2 on page 336).

The maximum propagation delay through the XOR-Tree, from the start of the chain to the end of the chain (XOR_OUT) is TBD.

## 3.0 Power, Reset and Clocks (Continued)

**Table 11. XOR-Tree Pin Chaining**

| Pin/Ball[1] | Signal Name | Pin/Ball | Signal Name | Pin/Ball | Signal Name | Pin/Ball | Signal Name |
|---|---|---|---|---|---|---|---|
| | Start of Chain | 121/E12 | A14_BE1 | 165/D5 | IOPB7/RING /PFAIL/RESET2 | 32/K2 | IOPA0/PWM0 |
| 69/M9 | IOPJ4/BST2 | 124/F15 | A0/ENV0 | 168/A6 | IOPC0 | 33/L4 | IOPA1/PWM1 |
| 70/N10 | IOPJ5/PFS | 125/D12 | A1/ENV1 | 169/D4 | IOPC1/SCL2 | 36/K1 | IOPA2/PWM2 |
| 71/R8 | KBSIN0 | 126/C13 | A2/BADDR0 | 170/C3 | IOPC2/SDA2 | 37/M4 | IOPA3/PWM3 |
| 72/P9 | KBSIN1 | 127/E15 | A3/BADDR1 | 171/A5 | IOPC3/TA1 | 38/N3 | IOPA4/PWM4 |
| 73/M10 | KBSIN2 | 128/D14 | A4/TRIS | 172/B4 | IOPC4/TB1/EXWINT22 | 39/L1 | IOPA5/PWM5 |
| 74/N11 | KBSIN3 | 129/D15 | A13_BE0 | 173/A4 | SEL0 | 40/M2 | IOPA6/PWM6 |
| 75/R9 | IOPJ6/PLI | 130/C15 | A12 | 174/A3 | SEL12_SEL2 | 41/M1 | IOPD4/SDA3 |
| 76/P10 | IOPJ7/BRKL_RSTO | 131/C14 | A5/SHBM | 175/B3 | IOPC5/TA2 | 42/N1 | IOPD5/SCL3 |
| 77/M11 | KBSIN4 | 132/B15 | A6 | 176/A2 | IOPC6/TB2/EXWINT23 | 43/N2 | IOPA7/PWM7 |
| 78/N12 | KBSIN5 | 133/A15 | A7 | 1/A1 | IOPC7/CLKOUT | 44/P1 | IOPE5/A20/EXWINT40 |
| 79/P11 | KBSIN6 | 134/B14 | A11 | 2/B2 | IOPE4/SWIN | 47/R2 | IOPQ3/CLK |
| 80/R10 | KBSIN7 | 135/A14 | A10 | 3/B1 | IOPM4/D12 | 48/R3 | IOPL4/WR1 |
| 87/P13 | IOPE0/AD4 | 138/A12 | D0 | 4/C1 | IOPM5/D13 | 49/P3 | KBSOUT0 |
| 88/R14 | IOPE1/AD5 | 139/C12 | D1 | 5/C2 | IOPB5/(GA20) | 50/R4 | KBSOUT1 |
| 89/R15 | IOPE2/AD6 | 140/D11 | D2 | 6/D1 | IOPB6/KBRST | 51/N4 | KBSOUT2 |
| 90/P14 | IOPE3/AD7 | 141/A11 | D3 | 7/D3 | SERIRQ | 52/M5 | KBSOUT3 |
| 103/K13 | IOPL3/A19 | 142/B12 | A9 | 8/E4 | IOPQ0/LDRQ | 53/R5 | KBSOUT4 |
| 104/J12 | A18 | 143/C11 | A8 | 9/E1 | LFRAME | 54/P4 | IOPD6/SDA4 |
| 105/K14 | TINT | 144/D10 | D4 | 10/D2 | LAD3 | 55/N5 | IOPD7/SCL4 |
| 106/J15 | TCK | 145/B11 | D5 | 13/E2 | LAD2 | 56/M6 | KBSOUT5 |
| 107/J13 | TDO | 146/A10 | D6 | 14/F1 | LAD1 | 57/P5 | KBSOUT6 |
| 108/H12 | TDI | 147/C10 | D7 | 15/F3 | LAD0 | 58/R6 | KBSOUT7 |
| 109/J14 | TMS | 148/D9 | IOPM0/D8 | 18/G1 | LCLK | 59/N6 | KBSOUT8 |
| 110/H13 | IOPF0/PSCLK1 | 149/B10 | IOPM1/D9 | 19/G3 | RESET1 | 60/M7 | KBSOUT9 |
| 111/G13 | IOPF1/PSDAT1 | 150/A9 | RD | 22/H3 | IOPQ1/SMI | 61/P6 | KBSOUT10 |
| 112/H14 | A17 | 151/C9 | WR0 | 23/J3 | IOPQ2/PWUREQ | 62/R7 | IOPJ2/BST0 |
| 113/G12 | A16 | 152/D8 | SELIO | 24/H2 | IOPE6/LPCPD /EXWINT45 | 63/N7 | IOPJ3/BST1 |
| 114/F13 | IOPF2/PSCLK2 | 153/B9 | IOPB0/URXD1 | 25/J4 | IOPE7/CLKRUN /EXWINT46 | 64/M8 | KBSOUT11 |
| 115/H15 | IOPF3/PSDAT2 | 154/C8 | IOPB1/UTXD1 | 26/K3 | IOPD0/RI1/EXWINT20 | 65/P7 | KBSOUT12 |
| 116/G14 | IOPF4/PSCLK3 /URXD2 | 155/C7 | IOPM2/D10 | 27/H1 | IOPM6/D14 | 66/N8 | KBSOUT13 |
| 117/F12 | IOPF5/PSDAT3/UTXD2 | 156/B8 | IOPM3/D11 | 28/J2 | IOPM7/D15 | 67/N9 | KBSOUT14 |
| 118/E13 | IOPF6/PSCLK4 /USCLK2 | 162/C5 | IOPB2/USCLK1 | 29/K4 | IOPD1/RI2/EXWINT21 | 68/P8 | KBSOUT15/**XOR_OUT** |
| 119/G15 | IOPF7/PSDAT4 | 163/A7 | IOPB3/SCL1 | 30/L3 | IOPD2/EXWINT24 /RESET2 | | End of Chain |
| 120/F14 | A15_CBRD | 164/B6 | IOPB4/SDA1 | 31/J1 | IOPD3/ESCI | | |

1. “Pin” for the LQFP package; “Ball” for the FBGA package.

# 4.0 Embedded Controller Modules

## 4.1 BUS INTERFACE UNIT (BIU)

The BIU directly interfaces with a wide variety of devices, including ROM, SRAM and flash memory devices and I/O devices. It interfaces via address, data and control buses without the need for external glue logic.

The BIU also defines the access time to the on-chip ROM Main block to provide cycle-by-cycle compatibility between environments; see Section 1.4 on page 28 and “Accessing Base Memory” on page 33. The base memory is associated with zone 1 of the BIU.

### 4.1.1 Features

- Four address zones for static devices (SRAM, ROM, flash, I/O).
- Basic bus cycle: two clock cycles.
- Configurable fast read bus cycles with 1-cycle read duration.
- Wait states: configurable between zero and seven clock cycles.
- Hold cycles: configurable between zero and three clock cycles.
- I/O expansion support.
- Configurable burst on read.
- Burst read: one clock cycle.
- Configurable early write or late write.
- Bus width: configurable per zone - 16-bit or 8-bit.

### 4.1.2 Functional Description

#### Interface

The BIU interfaces between:

- Internal core bus
- External static memory
- Off-chip I/O (memory-mapped) devices

The BIU performs the following functions:

- Distinguishes between four static memory zones
- Selects the relevant configured parameters of the accessed zone (e.g., the number of wait states)
- Issues the appropriate bus cycle to access the zone

Each memory zone has a different address range and a set of parameters that define access to this zone. The set of parameters is software configurable.

#### Static Memory and I/O Support

The BIU accesses static memory devices (ROM, SRAM, flash and I/O devices) using static read and write bus cycles. The BIU can be configured to extend the bus cycles with wait cycles.

The BIU supports burst read bus cycles if the accessed zone is configured as burstable. (A burst-read bus cycle is an extension of the basic-read bus cycle in which additional data is accessed. A burst access usually requires only one clock cycle per additional data item. It may be extended by up to two clock cycles per additional data item.)

To support both I/O and static memory devices that require long hold times at the end of the access, the BIU can be configured to add up to three $T_{hold}$ clock cycles at the end of the bus cycle. In addition, the BIU can be configured to insert a $T_{idle}$ clock cycle between two consecutive accesses to different zones.

#### Byte Access

The internal core bus is 16-bits wide and supports byte and word transactions.

The BIU issues the appropriate bus cycle to access the right bytes, according to the core bus transaction and the memory device bus width. Table 12 and Table 13 summarize the details:

## 4.0 Embedded Controller Modules (Continued)

**Table 12. Bus Cycles of a 16-Bit Data Bus**

| Number of Bytes | Transferred Core Bus Bytes | | Address (LSB) | Data Bus Pins |
|---|---|---|---|---|
| 1 | | B0 | 0 | 0-7 |
| 1 | B1 | | 1 | 8-15 |
| 2 | B1 | B0 | 0 | 0-15 |
| 2 | B1 | | 1 | 8-15 |
| | | B0 | Then 0 | 0-7 |

**Table 13. Bus Cycles of an 8-Bit Data Bus**

| Number of Bytes | Transferred Core Bus Bytes | | Address (LSB) | Data Bus Pins |
|---|---|---|---|---|
| 1 | | B0 | 0 | 0-7 |
| 1 | B1 | | 1 | 0-7 |
| 2 | | B0 | 0 | 0-7 |
| | B1 | | Then 1 | 0-7[1] |
| 2 | B1 | | 1 | 0-7 |
| | | B0 | Then 0 | 0-7 |

1. Burst bus cycle, if burstable; otherwise, the core transaction is broken into “basic” bus cycles.

On write cycles of a single byte, the remaining eight bits of the bus are floating.

On read cycles of a single byte, the remaining eight bits of the bus are ignored. There is no need for external pull-up resistors.

### Clock and Bus Cycles

There are two types of bus cycles: data transfer and non-data transfer. Data transfer bus cycles cause transfer of data from or to the memory device. Non-data transfer bus cycles (described in Section 4.1.9 on page 80) are used for observability of internal bus transactions and do not involve data transfer from or to external devices.

There are four types of data transfer bus cycles:

- Early write
- Late write
- Normal read
- Fast read

The BIU uses EWR configuration bit in BCFG register to select the early or late write data transfer bus cycle. It uses FRE in SZCFGn register (where “n” refers to zone 0, 1 or 2) to select normal read or fast read data transfer bus cycles.

The basic late write bus cycle takes two clock cycles. The basic early write bus cycle takes three clock cycles. When the BIU uses the early write bus cycle, the $\overline{RD}$ signal is not required for interfacing with the memory device (with the exception of flash). On reset, the early write bus cycle is configured.

The basic normal read bus cycle takes two clock cycles. Fast read bus cycle always takes one clock cycle. On reset, the normal read bus cycle is configured.

**Notes**:

1. In the descriptions that follow, the “n” in $\overline{SELn}$ signal refers to two of the three available BIU select signals (numbered 0 for zone 0, or 12 for zones 1 and 2). The third signal is labelled $\overline{SELIO}$.
2. For all timing diagrams, the value of BST0-2 depends on the type of core bus transaction.
3. In the following paragraphs, SZCFGn refers to three of the four BIU zone configuration registers (n = 0, 1 or 2); the fourth configuration register is labelled IOCFG.

# 4.0 Embedded Controller Modules (Continued)

## 4.1.3 Clock Cycles

### Basic Bus Cycle

A basic bus cycle comprises one to three clock cycles (depending on the type of bus cycle). Adding extra wait or hold clock cycles extends the data transfer bus cycles. Every data transfer bus cycle has the T1 and T2 clock cycles, with the exception of the fast read bus cycle, which has only one clock cycle (T1-2).

**$T_{idle}$ Cycle** Clock cycles that are not used for bus cycles are called Idle clock cycles ($T_{idle}$). $T_{idle}$ cycles are added when the BIU does not need to generate a bus transaction or as specifically configured pauses between two consecutive transactions. When more than one $T_{idle}$ cycle is requested as a pause, the $T_{idle}$ cycles overlap and only one $T_{idle}$ cycle is added.

$T_{idle}$ clock cycles can be inserted between two consecutive accesses in different zones (to allow long hold times or buffer disable times). To do this, either program IPRE and/or IPST in SZCFGn register (or IPST in IOCFG register); see Figure 17 on page 75.

$T_{idle}$ clock cycles are also added between an early write and a read bus cycle, and between a late write and a fast read bus cycle; see Figure 22 on page 78.

**T1 Cycle** Every bus cycle starts with T1. In this clock cycle, the address of the selected device (either external or internal) is set on the address pins. Write bus cycles never drive data during T1.

**T2 Cycle** The read T2 bus cycles always sample the data at the end of T2.

The write T2 bus cycles always drive data during T2. If no $T_{hold}$ clock cycles follow, the data bus is put in TRI-STATE after the T2 cycle.

**T1-2 Cycle** The fast read T1-2 bus cycle is a one-cycle read transaction.

At the start of the clock cycle, the address of the selected device is set on the address pins, and the $\overline{SELn}$ and $\overline{RD}$ signals are activated. At the end of the clock cycle, the BIU samples the data.

**T3 Cycle** Early write bus cycles always have the T3 clock cycle. No other bus cycles have this clock cycle.

At the start of this clock cycle, $\overline{SELn}$ (or $\overline{SELIO}$) is deactivated; then $\overline{WR0\text{-}1}$ is deactivated. The address and data remains valid until T3 is completed. If no $T_{hold}$ clock cycles follow, the data bus is put in TRI-STATE after the T3 cycle.

### Optional Clock Cycles

The following clock cycles are optional in a data transfer bus cycle:

- TIW (Internal Wait)
- T2B (T2 burst)
- TBW (Burst Wait)
- $T_{hold}$

**TIW Cycle** Extend the basic data transfer bus cycle by adding wait clock cycles. To do this, program WAIT in SZCFGn register (or IOCFG register) with the required additional wait clock cycles. Wait clock cycles generated by this action are named TIW (internal wait). TIW cycles are added after T1 and followed by T2 cycles. Data is always driven during wait clock cycles of a write bus cycle.

**T2B Cycle** Data of read burst bus cycles is sampled at the end of T2B. If the TBW cycle is not configured, the address is changed at the start of T2B. Write bus cycles do not have this clock cycle.

**TBW Cycle** A burst bus cycle can be extended by one wait clock cycle, named TBW. This is done according to WBR in SZCFGn register. The address is changed at the start of TBW. Write bus cycles do not have this clock cycle.

**$T_{hold}$ Cycle** Hold cycles are added after T2 or T2B (if there is a burst bus cycle) or T3 (according to HOLD in SZCFGn or IOCFG register); the address and data (during a write bus cycle) are always valid during these cycles. The data bus is put in TRI-STATE after the last $T_{hold}$.

### Other Clock Cycles

**Special $T_{idle}$ Cycle** During $T_{idle}$ cycles, one of the $\overline{SEL0\text{-}2}$ signals and the $\overline{RD}$ signal may be activated for one clock cycle. This happens due to special activity on the internal core bus. To avoid contention on the memory bus, it is guaranteed that this clock cycle is followed by a sufficient number of $T_{idle}$ cycles before the next T1 cycle is performed.

The number of $T_{idle}$ cycles that follows is at least the number required by the selected zone as configured in HOLD field in SZCFGn register.

## 4.0 Embedded Controller Modules (Continued)

**Burst Read Cycles** A read bus cycle consisting of the basic bus cycle plus additional clock cycles called “burst bus cycles”. The burst bus cycles occur if the bus is burstable (BRE in SZCFGn register is 1), the configured bus width is eight bits and the core attempts to read a word. When the bus is not burstable (BRE in SZCFGn register is 0), the BIU issues two separate read bus cycles. Write bus cycles are never burstable, and the BIU always issues two separate write bus cycles.

### Control Signals

The write bus cycles use byte write qualifiers on $\overline{WR0\text{-}1}$ pins:

- They access an 8-bit wide memory on D0-7 data lines.
  One byte is accessed on basic bus cycles. Only the $\overline{WR0}$ pin is used as the byte write qualifier.
- They access a 16-bit wide memory on D0-15 data lines.
  Either one or two bytes are accessed on basic bus cycles. The $\overline{WR0}$ pin is used as an even byte (D0-7) write qualifier and $\overline{WR1}$ pin is used as an odd byte (D8-15) write qualifier.

### 4.1.4 Early Write Bus Cycle

If EWR in BCFG register is 1, the BIU uses early write bus cycles. This allows removal of the $\overline{RD}$ signal from the memory device interface. The basic early write bus cycle takes three clock cycles.

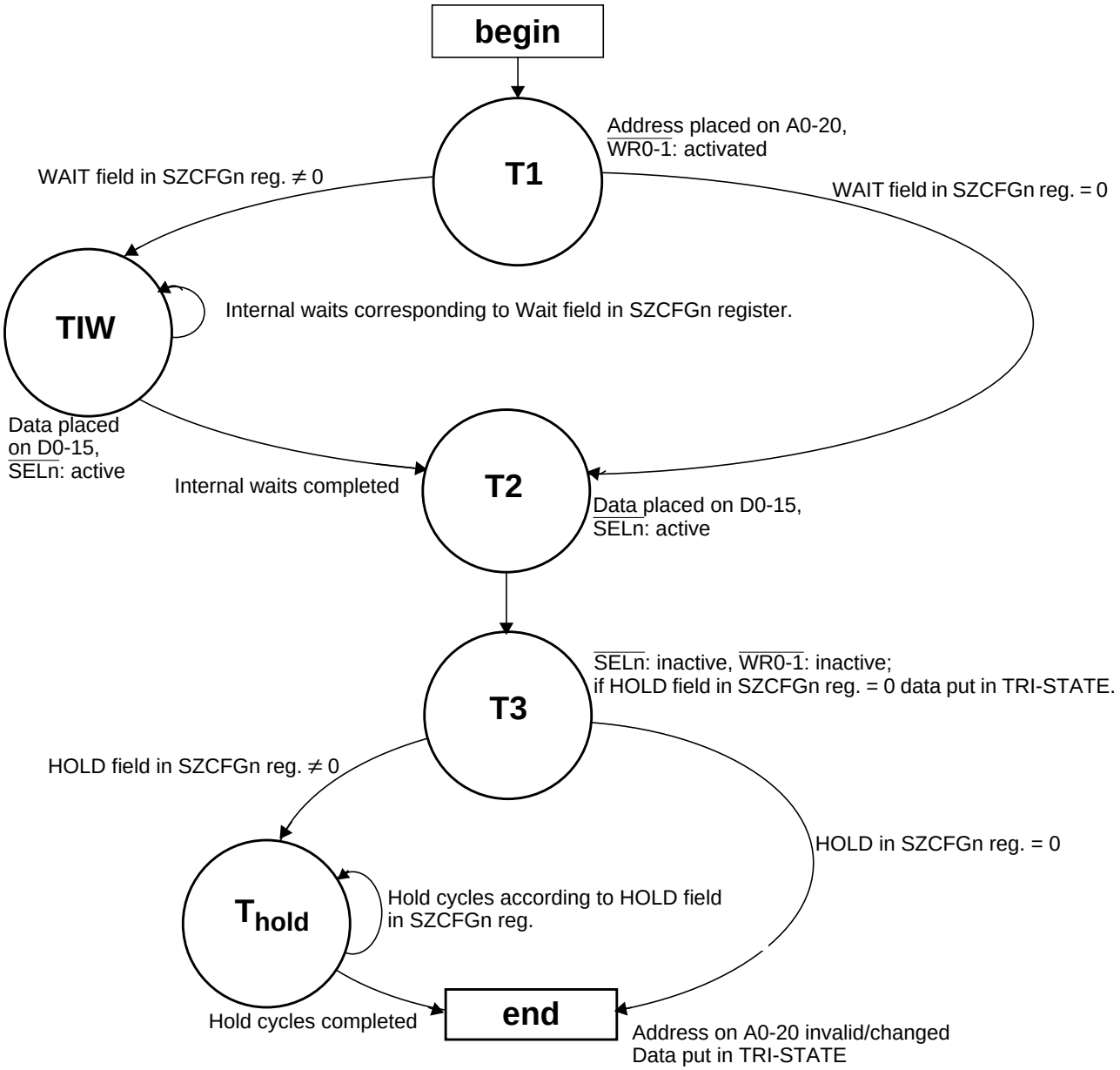
The cycle starts at T1; at this point, the data bus is in TRI-STATE, the address is placed on the address bus and $\overline{RD}$ is inactive. indicating that this is a write bus cycle. Then, $\overline{WR0\text{-}1}$ are activated.

At the first TIW or T2 (when there are no TIW cycles), the data is placed on the data bus and the $\overline{SELn}$ (or $\overline{SELIO}$) is activated. The bus transaction is terminated at T3; at this point, $\overline{SELn}$ (or $\overline{SELIO}$) becomes inactive. Then $\overline{WR0\text{-}1}$ become inactive and the data bus is put in TRI-STATE. The address remains valid until T3 is complete.

$T_{hold}$ clock cycles may follow T3, according to HOLD in SZCFGn or IOCFG registers (may be 0). The address and data remain valid until the end of the last $T_{hold}$ cycle. The data is put in TRI-STATE in the clock cycle after the last $T_{hold}$ or T3 (if no $T_{hold}$ cycle is configured); see Figures 11, 12 and 13.

If a read bus cycle immediately follows an Early Write bus cycle, an idle cycle is added between the two.

## 4.0 Embedded Controller Modules (Continued)




**Note:** References to SZCFGn also apply to the IOCFG register.
References to SELn also apply to the SELIO signal.

**Figure 11. Early Write Bus Cycle**

## 4.0 Embedded Controller Modules (Continued)

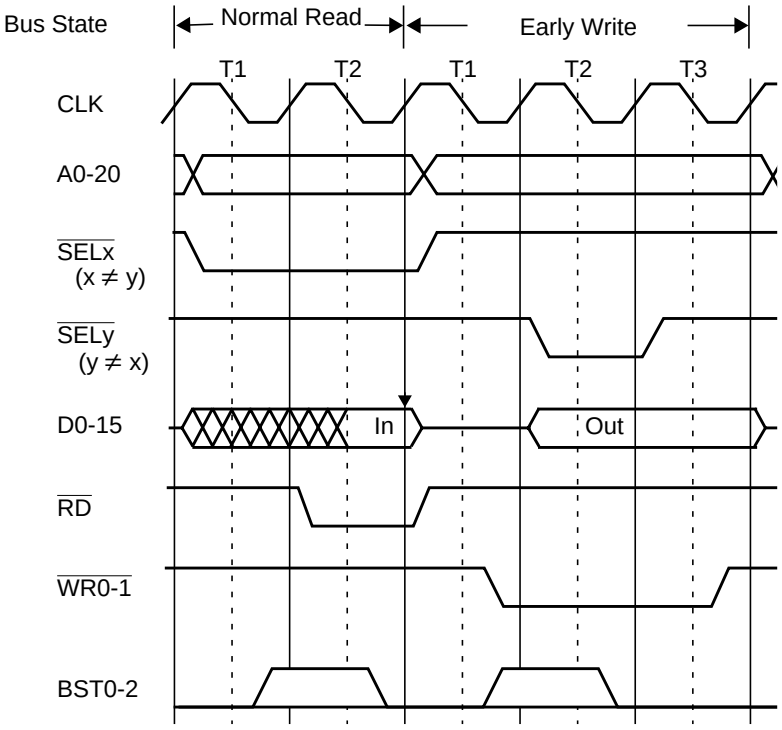



**Figure 12. Early Write Following Normal Read with 0 Wait**

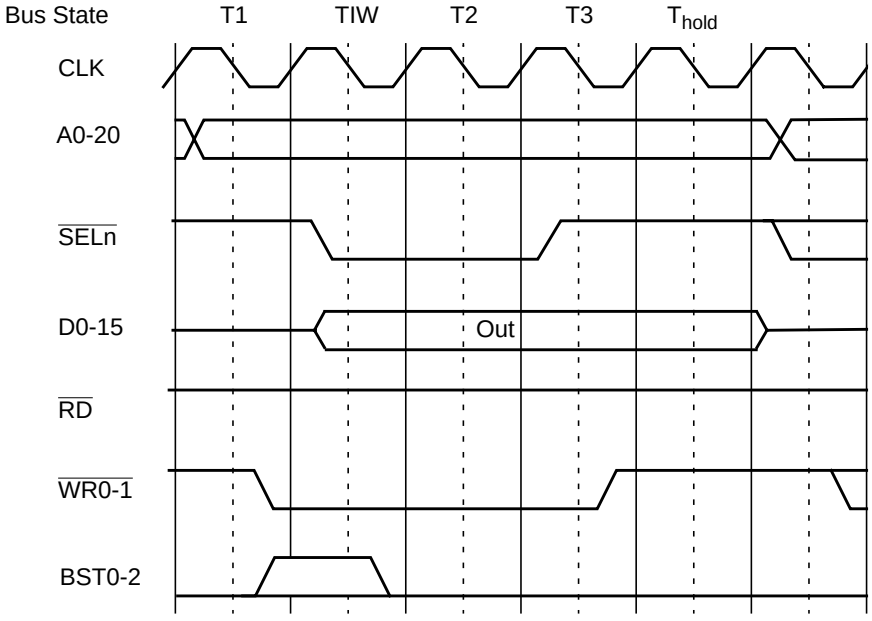



**Figure 13. Early Write Bus Cycle with 1 Internal Wait and 1 Hold**

## 4.0 Embedded Controller Modules (Continued)

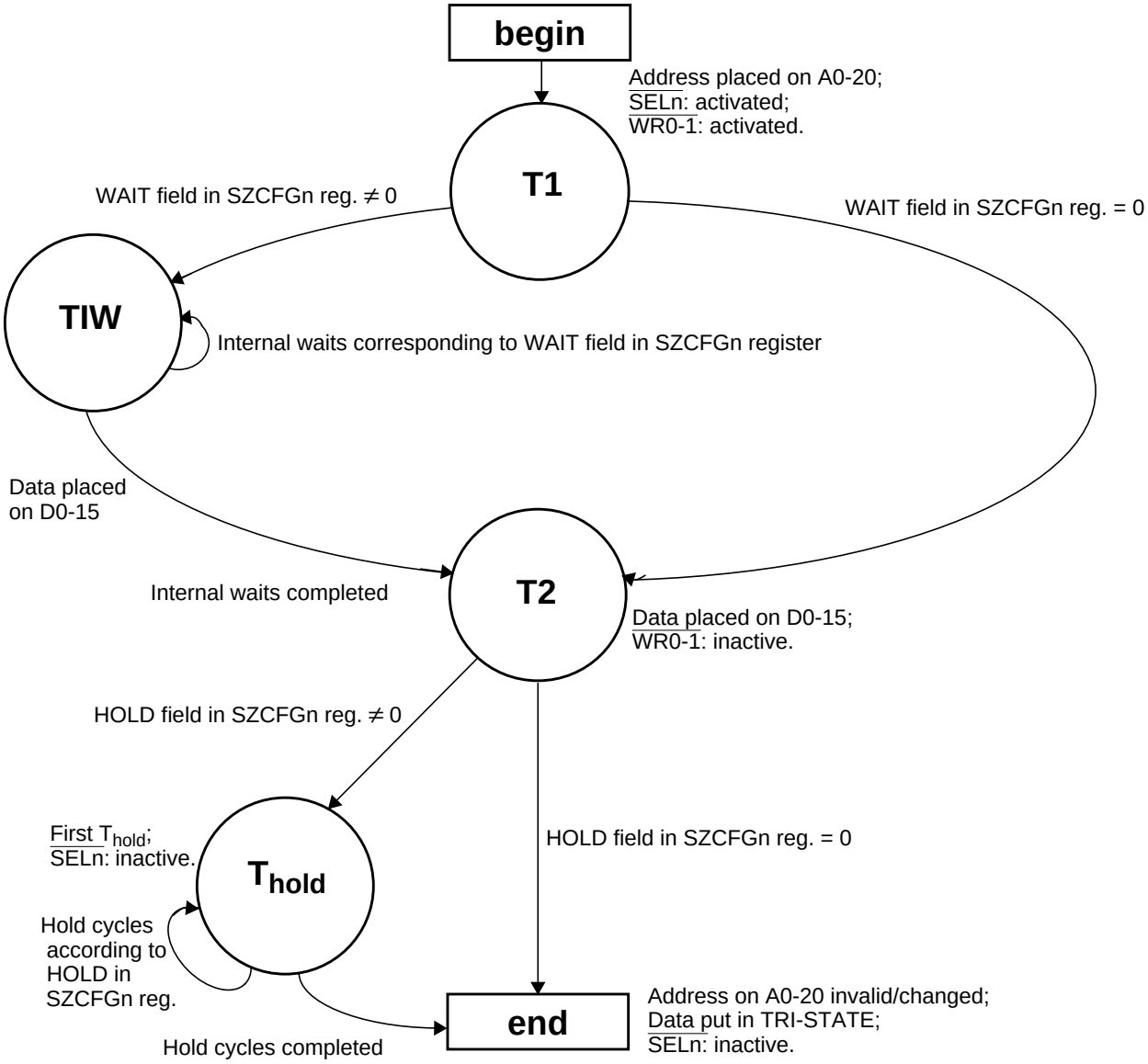
### 4.1.5 Late Write Bus Cycle

If EWR in BCFG register is 0, the BIU uses the late write bus cycle. The basic late write bus cycle takes two clock cycles. This write bus cycle requires the $\overline{RD}$ signal in the memory device interface.

A write bus cycle starts at T1. When the data bus is in TRI-STATE, the address is placed on the address bus and $\overline{SELn}$ (or $\overline{SELIO}$) is activated. Next, $\overline{WR0\text{-}1}$ are activated. $\overline{RD}$ is inactive to indicate this is a write transaction.

At the first TIW or T2 (when there are no TIW cycles), the data is placed on the data bus. The bus cycle is completed at T2; at this point, $\overline{WR0\text{-}1}$ are deactivated. The address and data remain valid until T2 is completed.

After T2, the number of $T_{hold}$ cycles specified by HOLD in SZCFGn register (may be 0) is added to the transaction. When $T_{hold}$ cycles are added, the address and data remain valid until the end of the last $T_{hold}$ cycle. $\overline{SELn}$ (or $\overline{SELIO}$) is deactivated on the first $T_{hold}$ cycle. When no $T_{hold}$ cycles are specified, $\overline{SELn}$ (or $\overline{SELIO}$) is deactivated in the clock cycle after T2 unless another read or write from the same zone follows. The data is put in TRI-STATE in the clock cycle after the last $T_{hold}$ or T2 (if no $T_{hold}$ cycle is configured); see Figures 14, 15 and 16.




**Note:** References to SZCFGn also apply to the IOCFG register.
References to $\overline{SELn}$ also apply to the $\overline{SELIO}$ signal.

**Figure 14. Late Write Bus Cycle**

## 4.0 Embedded Controller Modules (Continued)

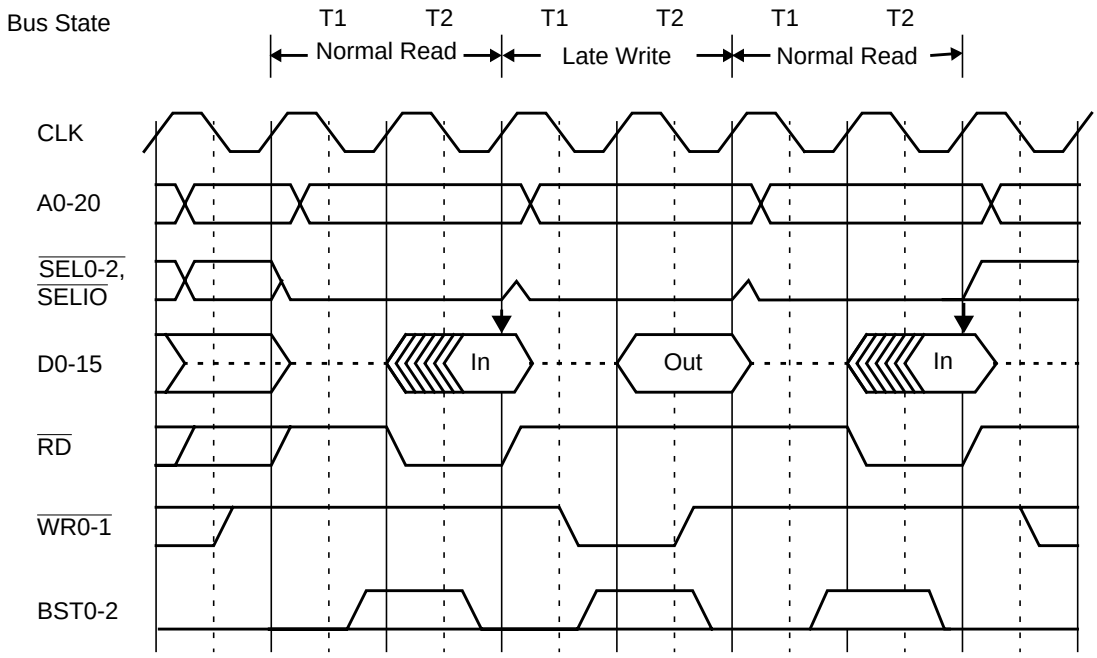



**Figure 15. Late Write Bus Cycle Between Normal Read Bus Cycles with 0 Wait**

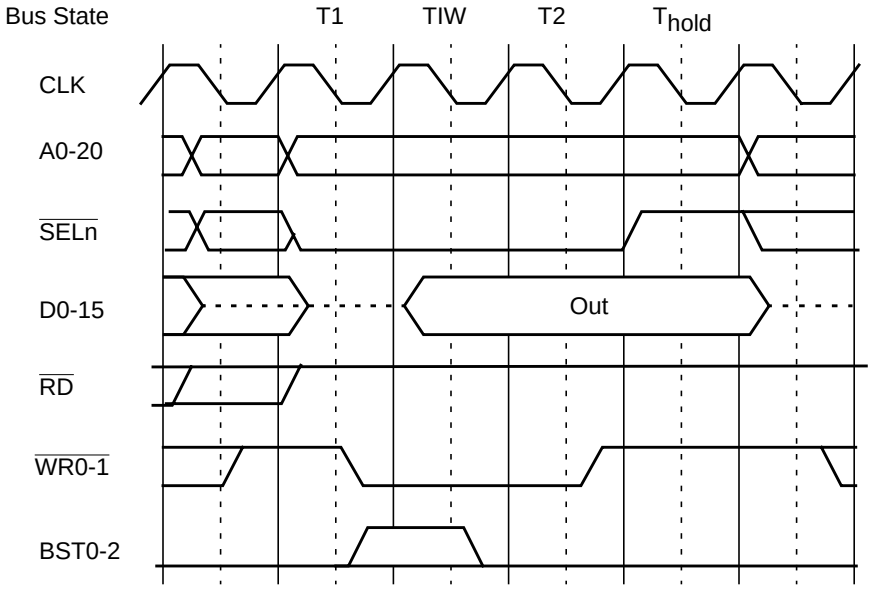



**Figure 16. Late Write Bus Cycle with 1 Internal Wait and 1 Hold**

### 4.1.6 Normal Read Bus Cycle

A read bus cycle starts at T1; at this point, the address is placed on the address bus, $\overline{SELn}$ (or $\overline{SELIO}$) is activated and $\overline{WR0\text{-}1}$ are inactive, indicating that this is a read bus cycle. The $\overline{RD}$ signal is activated on the first TIW or T2 (when there are no TIW cycles).

At the end of T2, the BIU samples the data on D0-7 or D0-15, according to BW signal in SZCFGn register. After T2, the number of $T_{hold}$ cycles specified by HOLD in SZCFGn register (may be 0) is added. $\overline{SELn}$ and $\overline{RD}$ are deactivated on the first $T_{hold}$ cycle. The address remains valid until the end of the last $T_{hold}$ cycle.

## 4.0 Embedded Controller Modules (Continued)

When no $T_{hold}$ cycles are specified, $\overline{SELn}$ is deactivated in the clock cycle that follows T2, unless another read from the same zone follows. The $\overline{RD}$ signal is always deactivated in the clock cycle following T2; see Figures 17, 18 and 19.

A burst bus cycle supplements the basic read bus cycle if the core attempts to access more bytes (i.e., a word) than the configured bus width (and BRE in SZCFGn register is set to 1). The burst bus cycle (T2B) follows T2 before the $T_{hold}$ cycles (if configured). A wait clock cycle (TBW) is added between T2 and T2B if WBR in SZCFGn register is set to 1.

The address of the burst bus cycle is changed on TBW (if configured) or T2B (if no TBW). At the end of T2B, data is sampled. The $\overline{RD}$ signal is activated during the burst bus cycle and is deactivated in the clock cycle following T2B; see Figures 20 and 21.

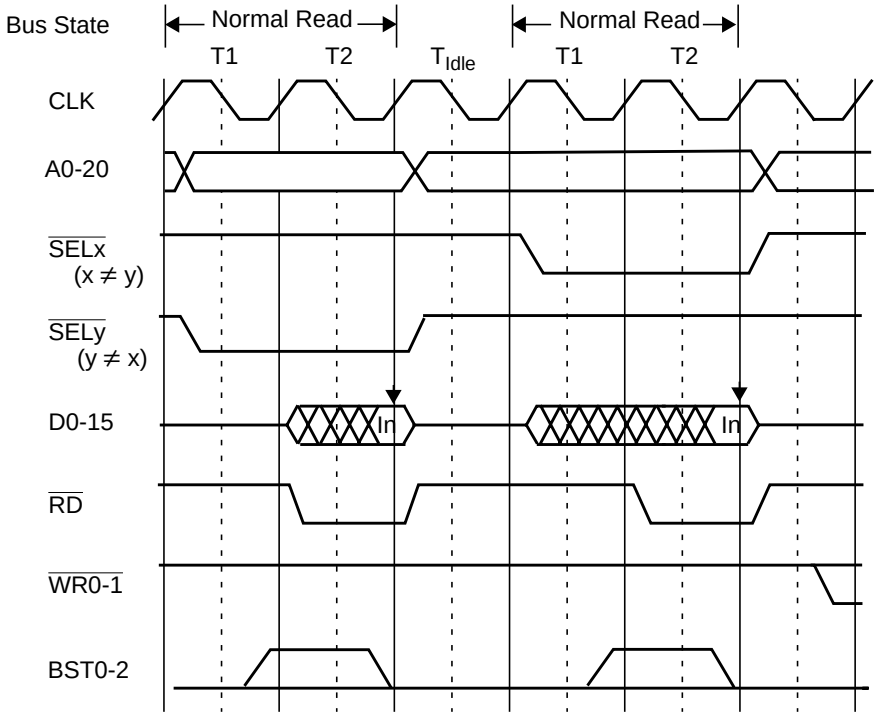



**Figure 17. Two Basic Normal Read Bus Cycles with Idle In Between (IPST Bit in SZCFGy Register = 1, IPRE Bit in SZCFGx Register = 1)**

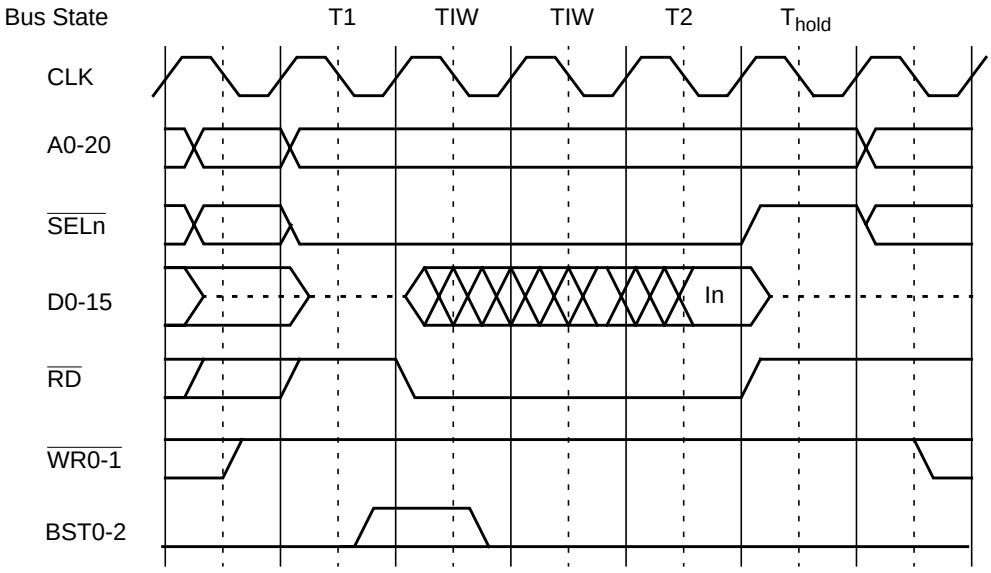



**Figure 18. Normal Read Bus Cycle with 2 Internal Waits and 1 Hold**

## 4.0 Embedded Controller Modules (Continued)

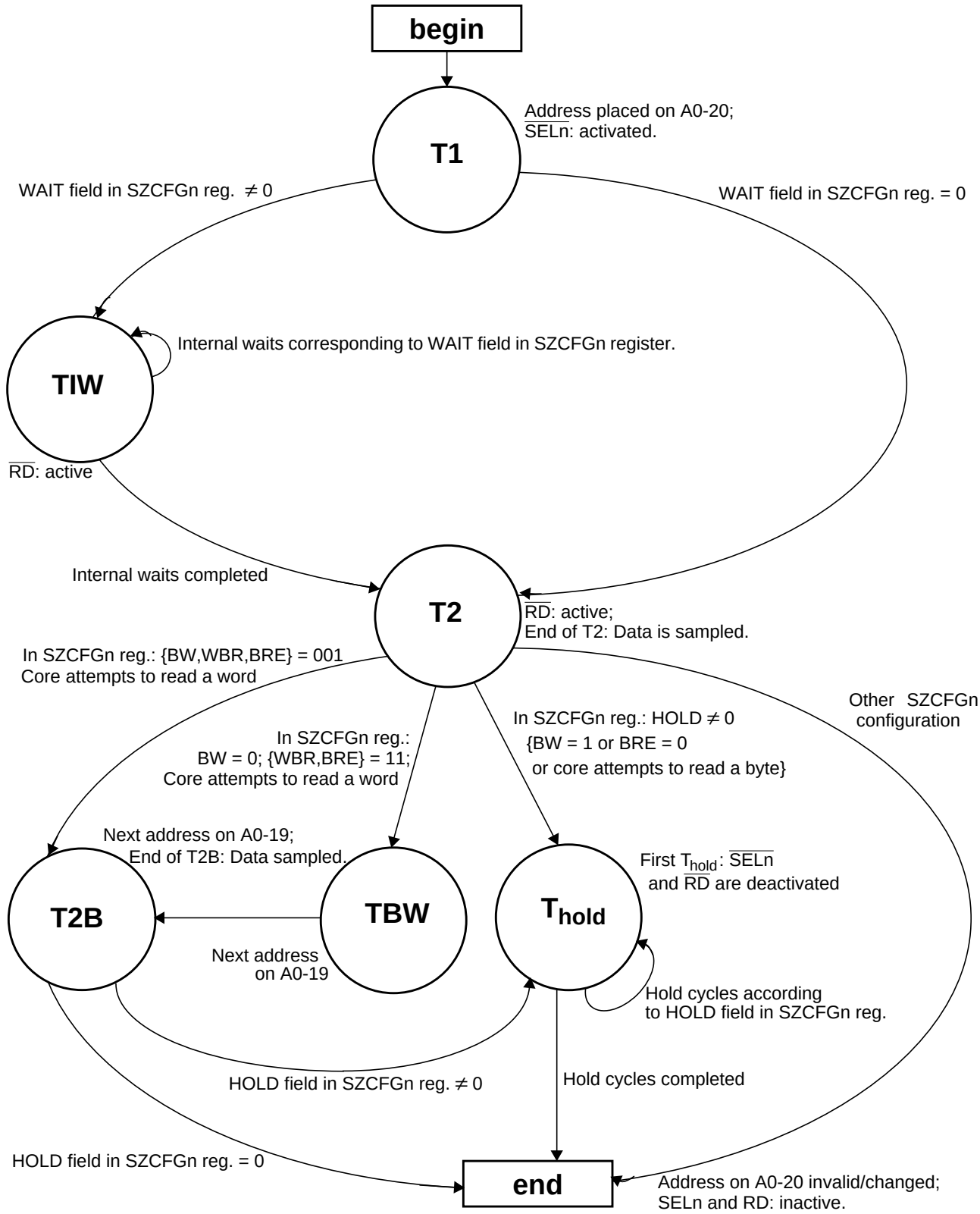



**Figure 19. Normal Read Bus Cycle**

**Note:** References to SZCFGn also apply to the IOCFG register.
References to SELn also apply to the SELIO signal.
TBW and T2B states do not exist in bus cycles of the IO zone.

## 4.0 Embedded Controller Modules (Continued)

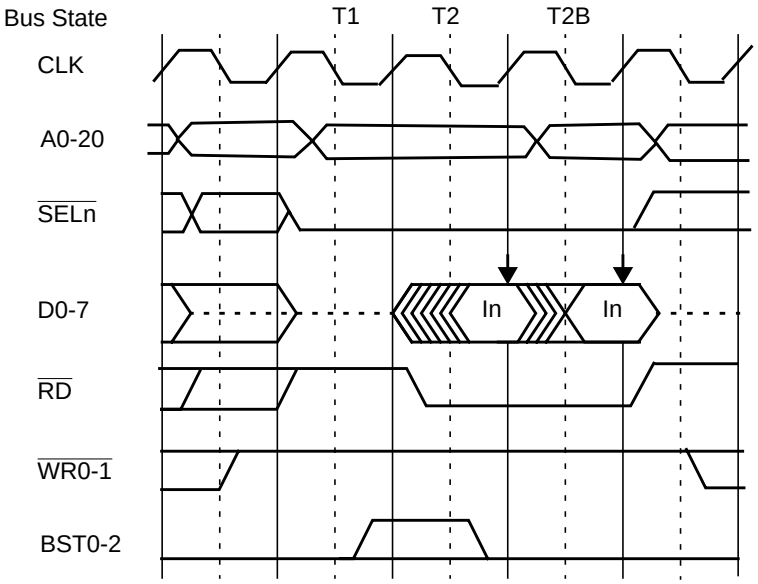



**Figure 20. Normal Read Bus Cycle with 0 Wait on Burst**

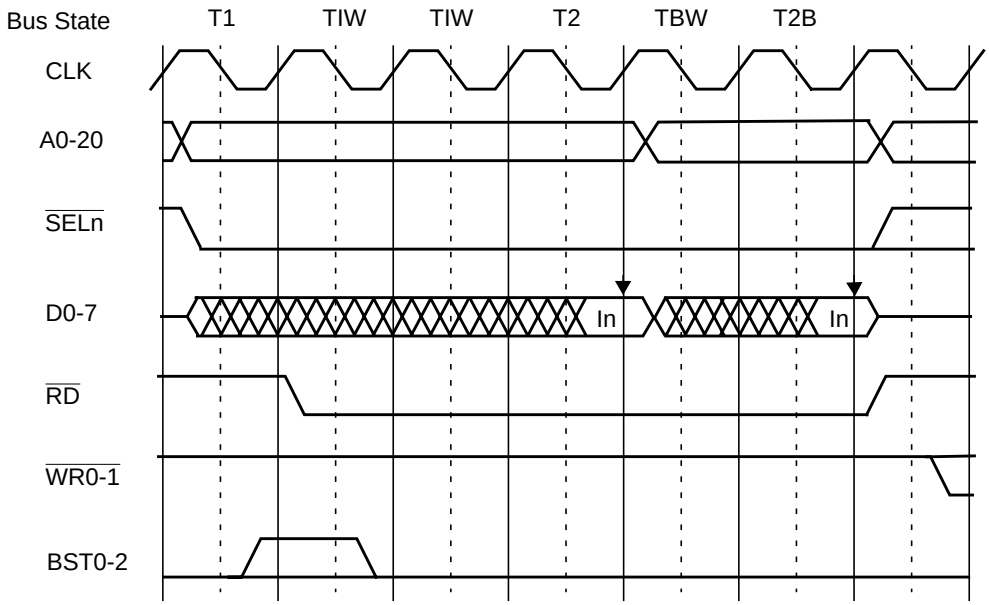



**Figure 21. Normal Read Bus Cycle with 2 Internal Waits and 1 Wait on Burst**

# 4.0 Embedded Controller Modules (Continued)

## 4.1.7 Fast Read Bus Cycle

When FRE bit in SZCFGn register is 1, the fast read bus cycle is enabled for zone n. The fast read bus cycle takes one clock cycle.

At the start of the T1-2 clock cycle, the address is placed on the address bus, and $\overline{SELn}$ and $\overline{RD}$ are activated. $\overline{WR0\text{-}1}$ are inactive, indicating a read bus cycle. At the end of the clock cycle, the BIU samples the data. $\overline{SELn}$ and $\overline{RD}$ are deactivated in the following clock cycle, unless another read from the same zone follows. If a write to the same zone follows and late write is configured, $\overline{SELn}$ remains activated. The address remains valid until the start of the clock cycle after the T1-2 clock cycle.

The fast read bus cycle cannot be extended by adding wait cycles (WAIT field in SZCFGn register is ignored during this bus cycle). Additionally, hold cycles cannot be added (HOLD field in SZCFGn register is also ignored). When a write bus cycle immediately precedes, in sequence, a fast read bus cycle, an idle clock cycle is forced between the two; see Figure 22.

Note that when the core attempts to access more bytes than the configured bus width (i.e., a word), the transaction is broken up into “basic” (T1-2) bus cycles.

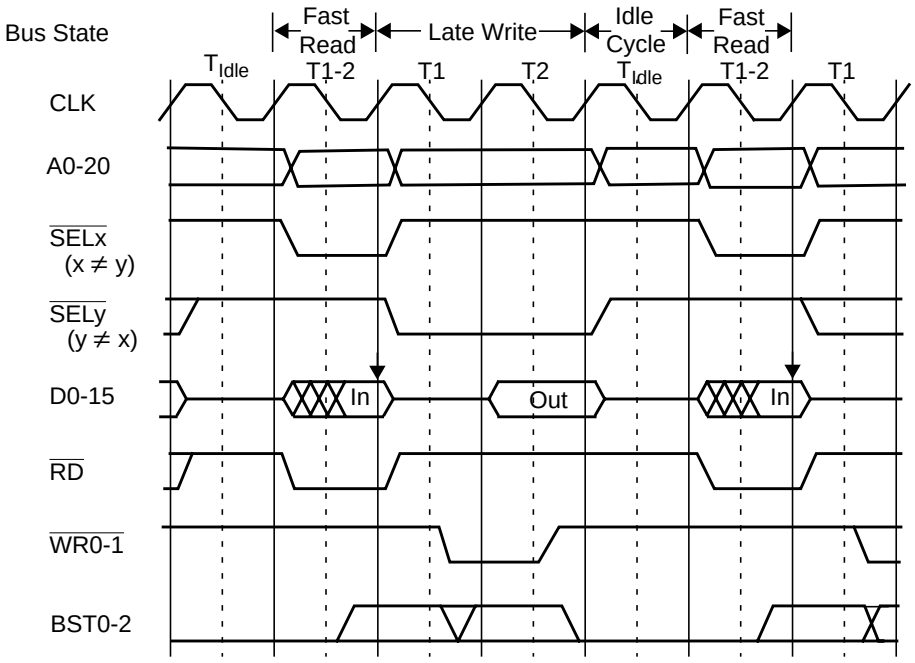



**Figure 22. Fast Read Bus Cycle**

## 4.1.8 I/O Expansion Bus Cycles

The I/O expansion bus cycles support the implementation of on-chip I/O port functionality (when the pins of the on-chip I/O ports are used to support DEV environment) and/or additional ports, using off-chip external logic.

I/O expansion bus cycles access the off-chip I/O device using the following signals:

- SELIO.
- Address lines A0-7.
- The $\overline{RD}$ and $\overline{WR0\text{-}1}$ signals may be used.

The design minimizes the off-chip logic required to implement the I/O ports. It is costly to implement a port with pins individually configured for input or output. Implementing ports for input only or output only reduces expenses.

I/O expansion bus cycle is not generated during an access to a port register if one of the following conditions occurs:

- A port pin is available on-chip.
- All port pins are inputs and the port is being written.

## 4.0 Embedded Controller Modules (Continued)

### I/O Expansion Read/Write Bus Cycle

These cycles are always preceded by a $T_{idle}$ clock cycle (see Figure 23). The I/O zone is not burstable.

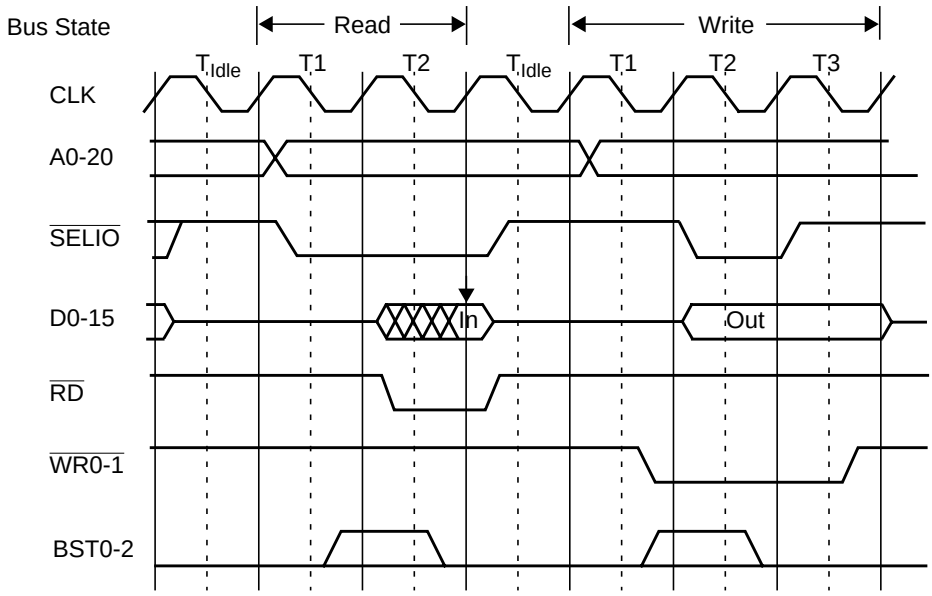
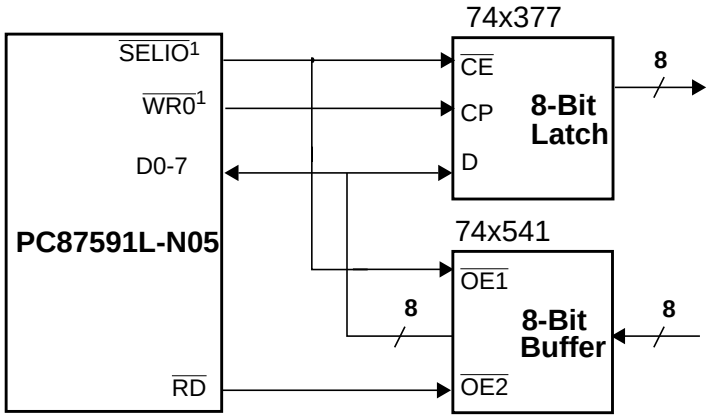



**Figure 23. I/O Expansion Bus Cycles (EWR bit in BCFG Register = 1)**

### I/O Expansion Example

Figure 24 shows an example of how two ports can be implemented off-chip, using I/O expansion. This example implements two 8-bit ports by connecting the SELIO, RD and WR0 pins to the latch/buffer controls.




1. This routing is for late write. If early write, SELIO is routed to CP and WR0 to CE. All other routing is unchanged.

**Figure 24. Example of an Implementation of Two Ports Using I/O Expansion**

# 4.0 Embedded Controller Modules (Continued)

## 4.1.9 Development Support

The BIU provides the following support for development systems.

### Bus Status Signals

The Bus Status BST0-2 signals indicate whether a transaction on the core bus was issued; they also indicate the transaction type; see Table 31 on page 237.

### Core Bus Monitoring

The core bus monitoring cycle is a non-data transfer bus cycle. It takes a single clock cycle - T1. On this cycle:

- The address pins display the address of the internal device accessed on the core bus.
- CBRD indicates the direction of the access (read or write).
- BE0-1 indicate which data bus bytes are accessed (lower or upper).
- BST0-2 display the core bus status.

The core bus monitoring cycle, as shown in Figure 25, is generated only when bit 1 (OBR) in BCFG register is 1.

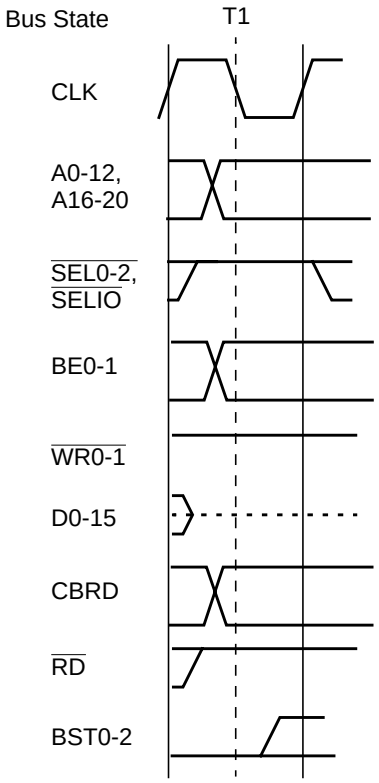



**Figure 25. Core Bus Monitoring Bus Cycle**

# 4.0 Embedded Controller Modules (Continued)

## 4.1.10 BIU Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

**BIU Register Map**

| Mnemonic | Register Name | Type |
|---|---|---|
| BCFG | BIU Configuration | R/W |
| IOCFG | I/O Zone Configuration | R/W |
| SZCFGn | Static Zone Configuration | R/W |

### BIU Configuration Register (BCFG)

The BCFG register controls the configuration of common features to all zones. On reset, this register is initialized to $07_{16}$.

Location: 00 $F980_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | **ISTL** | **OBR** | **EWR** |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 0 | **EWR (Early Write).**<br>0: Late write<br>1: Early write (default) |
| 1 | **OBR (Observability).** This bit determines if the address and status of internal accesses are observable.<br>0: Not observable (no toggle of external buses)<br>1: Observable (external bus toggle - default) |
| 2 | **ISTL (Internal Stall).** This bit determines if the internal bus is stalled while the BIU is busy.<br>0: Internal bus activity not stalled when BIU is busy<br>1: Stall internal bus activity when the external bus is busy (default) |
| 7-3 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## I/O Zone Configuration Register (IOCFG)

The IOCFG register controls the configuration of the I/O zone. On reset, it is initialized to $069F_{16}$.

Location: $00\ F982_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | **IPST** | **Res** | **BW** | **Reserved** | | **HOLD** | | **WAIT** | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 2-0 | **WAIT.** This field sets the number of TIW clock cycles that extend the bus cycle.<br>**Bits 2 1 0 Number**<br>0 0 0: None<br>0 0 1: One<br>0 1 0: Two<br>0 1 1: Three<br>1 0 0: Four<br>1 0 1: Five<br>1 1 0: Six<br>1 1 1: Seven (default) |
| 4-3 | **HOLD.** This field sets the number of $T_{hold}$ clock cycles.<br>**Bits 4 3 Number**<br>0 0: None<br>0 1: One<br>1 0: Two<br>1 1: Three (default) |
| 6-5 | **Reserved.** |
| 7 | **BW (Bus Width).** This bit sets the external bus width used for the I/O zone. It is initialized during reset to its default value.<br>0: 8-bit bus<br>1: 16-bit bus (default) |
| 8 | **Reserved.** |
| 9 | **IPST (Idle After Bus Cycle).** This bit determines if an idle cycle follows the current bus cycle when the next bus cycle is in a different zone.<br>0: No idle cycle inserted<br>1: Idle cycle inserted (default) |
| 15-10 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## Static Zone Configuration Register (SZCFGn)

The SZCFGn register (where n = 0, 1 or 2) controls the configuration of zone n. On reset, SZCFGn is initialized to $069F_{16}$.

Location: Zone 0 - $00\ F984_{16}$
Zone 1 - $00\ F986_{16}$
Zone 2 - $00\ F988_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | **FRE** | **IPRE** | **IPST** | **Res** | **BW** | **WBR** | **BRE** | **HOLD** | | **WAIT** | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 2-0 | **WAIT.** This field sets the number of TIW clock cycles that extend the bus cycle. This field is ignored in read transactions, when bit 11 (FRE) of this register is set to 1.<br>**Bits 2 1 0 Number**<br>0 0 0: None<br>0 0 1: One<br>0 1 0: Two<br>0 1 1: Three<br>1 0 0: Four<br>1 0 1: Five<br>1 1 0: Six<br>1 1 1: Seven (default) |
| 4-3 | **HOLD.** This field sets the number of $T_{hold}$ clock cycles. This field is ignored in read transactions, when bit 11 (FRE) of this register is set to 1.<br>**Bits 4 3 Number**<br>0 0: None<br>0 1: One<br>1 0: Two<br>1 1: Three (default) |
| 5 | **BRE (Burst Read Enable).** This bit is ignored in read transactions, when bit 11 (FRE) of this register is set to 1.<br>0: Disabled (default)<br>1: Enabled |
| 6 | **WBR (Wait on Burst Read).** This bit determines if a wait state (TBW) is added on a burst read transaction.<br>0: No TBW (default)<br>1: TBW |
| 7 | **BW (Bus Width).** This bit sets the external bus width used for the static zone. It is initialized during reset to its default value.<br>0: 8-bit bus<br>1: 16-bit bus (default) |
| 8 | **Reserved.** |
| 9 | **IPST (Idle After Bus Cycle).** This bit determines if an idle cycle follows the current bus cycle when the next bus cycle is in a different zone.<br>0: No idle cycle inserted<br>1: Idle cycle inserted (default) |

## 4.0 Embedded Controller Modules (Continued)

| Bit | Description |
|---|---|
| 10 | **IPRE (Idle Before Bus Cycle).** This bit inserts an idle cycle before the current bus cycle when this bus cycle is in a new zone.<br>0: No idle cycle inserted<br>1: Idle cycle inserted (default) |
| 11 | **FRE (Fast Read Enable).**<br>0: Disabled - Normal read bus cycle takes at least two clock cycles (default)<br>1: Enabled - Normal read bus cycle takes one clock cycle |
| 15-12 | **Reserved.** |

### 4.1.11 Usage Hints

The following usage hints help configure the BIU to maximize PC87591L-N05 performance and avoid contention on the data bus.

1. Memory Sections 0 and 1 (fast zone) and Section 2 (slow zone) can use a fast read bus cycle through the operation frequency of the PC87591L-N05; therefore, program SZCFG1 fields to be: WAIT=000, HOLD=00, BRE=0, WBE=0, BW=1, FRE=1.

   When Section 2 (slow zone) can operate with a fast read bus cycle, program SZCFG2 fields to be: WAIT=000, HOLD=00, BRE=0, WBE=0, BW=1, FRE=1.

   When Section 2 (slow zone) needs to operate with normal read and zero wait, program SZCFG2 fields to be: WAIT=000, HOLD=00, BRE=0, WBE=0, BW=1, FRE=0.
2. To avoid contention on the data bus when a read bus cycle (no $T_{hold}$ clock cycles) in one zone is followed by a read bus cycle in another zone, program IPST and IPRE in the different memory (I/O) zones as follows:

| Zone | IPRE | IPST |
|---|---|---|
| Zone 0 | 0 | 0 |
| Zone 1 | 1 | 0 |
| Zone 2 | 0/1[1] | 0 |
| Zone I/O | 1[2] | 0 |

1. Set IPRE when the zone is configured for fast read.
2. An IPRE is forced always for zone I/O.

# 4.0 Embedded Controller Modules (Continued)

## 4.2 DMA CONTROLLER (DMAC)

The DMAC transfers blocks of data between memory and I/O devices along four independent channels, with minimal intervention by the core. The source and destination addresses and the block size to be transferred may be defined for each of the channels.

### 4.2.1 Features

- Four Independent Direct Memory Access (DMA) channels.
- Direct (fly-by) and indirect (memory-to-memory) transfer types.
- Single-buffer, double-buffer and auto initialize operation modes.
- Fixed address (I/O device) or updated (memory device).
- Address update (increment or decrement) independent of the number of transferred bytes.
- Interrupt line for each channel.
- Programmable bus policy for each channel: continuous or intermittent use of the bus.
- Software DMA request for each channel.
- Maximum throughput in direct (fly-by) transfer:
  - — Intermittent: Every three clock cycles.
  - — Continuous: On internal core bus - every clock cycle.
    Otherwise - every two clock cycles.
- Maximum throughput in indirect (memory-to-memory) transfer:
  - — Intermittent: Every five clock cycles.
  - — Continuous: On internal core bus - every two clock cycles.
    Otherwise - every four clock cycles.

### 4.2.2 Functional Description

When transferring blocks of data using the DMAC, the source and destination addresses, as well as the block size and type of operation, are set up in advance by programing the appropriate control registers. Actual data transfers are handled by the DMAC channel in response to DMA transfer requests. On receiving a DMA transfer request ($\overline{\text{DMRQn}}$), if the channel is enabled, the DMAC performs the following operations:

1. Acquires control of the core bus according to the DMAC priority on the core bus.
2. Determines priority among the DMAC channels, one clock cycle before T1 of the DMAC transfer cycle. (T1 is the first clock cycle of the bus cycle.) Priority among the DMAC channels is fixed in descending order, with Channel 0 receiving the highest priority.
3. Executes data transfer bus cycle(s) according to the values stored in the control registers of the channel being serviced and according to the accessed memory address. It acknowledges the request during the bus cycle that accesses the requesting device.
4. If the transfer of a block is terminated, the DMAC does the following:
   a. Updates the termination flags.
   b. Generates an interrupt if enabled.
   c. Goes to step 6.
5. If $\overline{\text{DMRQn}}$ is still active and the Bus Policy is “continuous”, returns to step 3.
6. Relinquishes the internal core bus.

Each DMAC channel can be programed for direct (fly-by) or indirect (memory-to-memory) data transfer. Once a DMAC transfer cycle is in process, the next transfer request is sampled when the DMAC acknowledge is deactivated and subsequently, on the rising edge of each clock cycle.

The configuration of either address freeze or address update (increment or decrement) is independent of the number of transferred bytes, transfer direction or number of bytes in each DMAC transfer cycle. All these can be configured for each channel by programing the appropriate control registers.

Each DMAC channel has eight control registers. DMAC channels are described hereafter with the suffix “n”, where n represents the channel number in the register name (n = 0 to 3).

# 4.0 Embedded Controller Modules (Continued)

## 4.2.3 Channel Assignment in PC87591L-N05

Table 14 shows the assignment of the DMA channels to different tasks in the PC87591L-N05.

**Table 14. DMA Channel Assignment**

| Channel | Usage | Mode | Enable Control | Interrupt |
|---|---|---|---|---|
| 0 | USART1 Receive | Indirect | USART1 registers | DMA_INT0 |
| 1 | USART1 Transmit | Indirect | USART1 registers | DMA_INT1 |
| 2 | ACCESS.bus3 | Indirect | ACCESS.bus3 registers | DMA_INT2 |
| 3 | ACCESS.bus4 | Indirect | ACCESS.bus4 registers | DMA_INT3 |

## 4.2.4 Transfer Types

The DMAC uses two data transfer modes, direct (fly-by) and indirect (memory-to-memory). The choice of mode depends on the correlation between the source and destination bus lengths, the required bus performance and the peripheral structure (as indicated in Table 14).

### Direct (Fly-By) Transfers

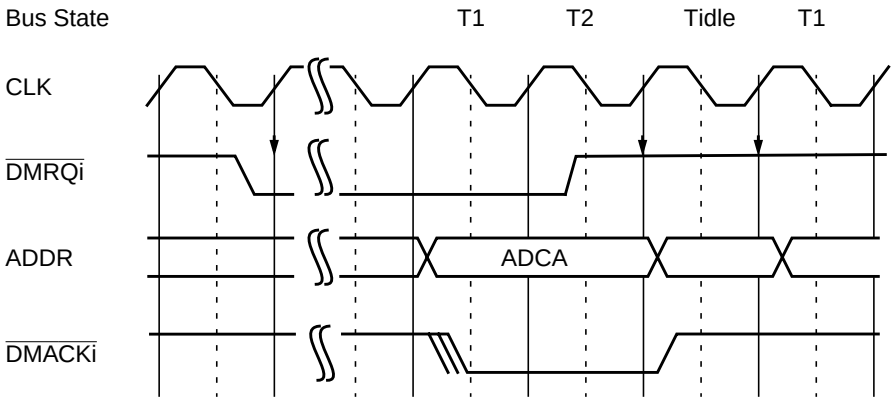



**Figure 26. DMAC Direct Bus Cycle Followed by a Core Bus Cycle**

In Direct mode, each data item is transferred using a single bus cycle without reading the data into the DMAC. It provides the fastest transfer rate, but it requires identical source and destination bus widths.

Data transfer cannot occur between two memory elements. One of the elements must be the I/O device that requested the DMA transfer. This device is referred to as the implied I/O device. The other element can be either memory or another I/O device and is referred to as the addressed I/O device.

The appropriate DMA acknowledge signal for each channel is asserted during the bus cycle.

If the bus policy is “intermittent”, maximum throughput is one transaction every three clock cycles. If bus policy is “continuous”, maximum throughput on the internal core bus is one transaction every two clock cycles.

Since only one address is required in Direct mode, this address is taken from the corresponding ADCAn counter. The DMAC Channel generates either a read or a write bus cycle according to the setting of DIR bit in DMACNTLn register.

When DIR bit is 0, a read bus cycle from the addressed device is performed, and the data is written to the implied I/O device. When DIR bit is 1, a write bus cycle to the addressed device is performed, and the data is read from the implied I/O device.

The configuration of either address freeze or address update (increment or decrement) is independent of the number of transferred bytes, transfer direction, or number of bytes in each DMAC transfer cycle. All these can be configured for each channel by programing the appropriate control register.

The number of bytes transferred in each cycle is taken from TCS bit in DMACNTLn register. After the data item has been transferred, the BLTCn counter is decremented by one. The ADCAn counter are updated according to INCAn field and ADA bit in DMACNTLn register.

## 4.0 Embedded Controller Modules (Continued)

### Indirect (Memory-to-Memory) Transfers

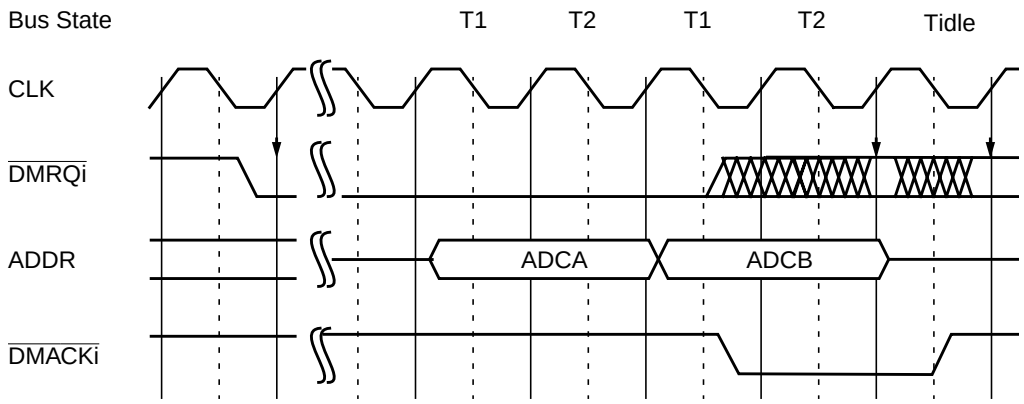



**Figure 27. Indirect Bus Cycle (DIR=0)**

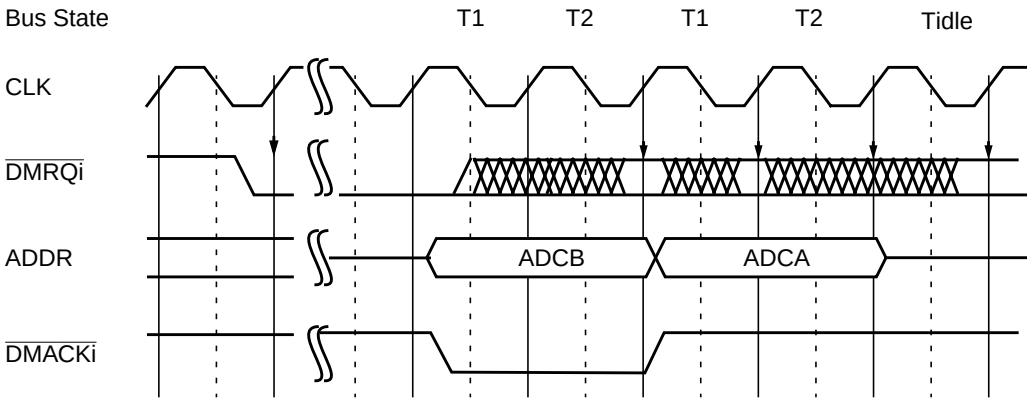



**Figure 28. Indirect Bus Cycle (DIR=1)**

In Indirect (Memory-to-Memory) mode, data transfers use two consecutive bus cycles. The data is first read into a temporary register and subsequently written into the destination. This mode is slower than Direct (Fly-By) mode, but it provides support for different source and destination bus widths.

Indirect mode also facilitates block transfers between two memory elements. Each element is an addressed device and can be either memory or an I/O device.

The appropriate DMA acknowledge signal for each channel is asserted during the Device B bus cycle.

If the bus policy is “intermittent”, maximum, throughput is every five clock cycles. If the bus policy is “continuous”, maximum throughput is every two clock cycles on the internal core bus (otherwise, it uses four clock cycles).

When DIR bit is 0, the first bus cycle reads data from the source using the ADCAn counter, and the second bus cycle writes the data into the destination using the ADCBn counter. When DIR bit is 1, the first bus cycle reads data from the source using the ADCBn counter, and the second bus cycle writes the data into the destination using the ADCAn counter.

The number of bytes transferred in each cycle is taken from TCS bit in DMACNTLn register. After the data item has been transferred, the BLTCn counter is decremented by one. The ADCAn and ADCBn counters are updated according to INCAn, INCBn, ADA and ADB fields in DMACNTLn register.

**Note:**

For transfer operations between two memory areas, see “Software DMA Request” on page 90.

# 4.0 Embedded Controller Modules (Continued)

## 4.2.5 Bus Policy

### Intermittent Operation Mode

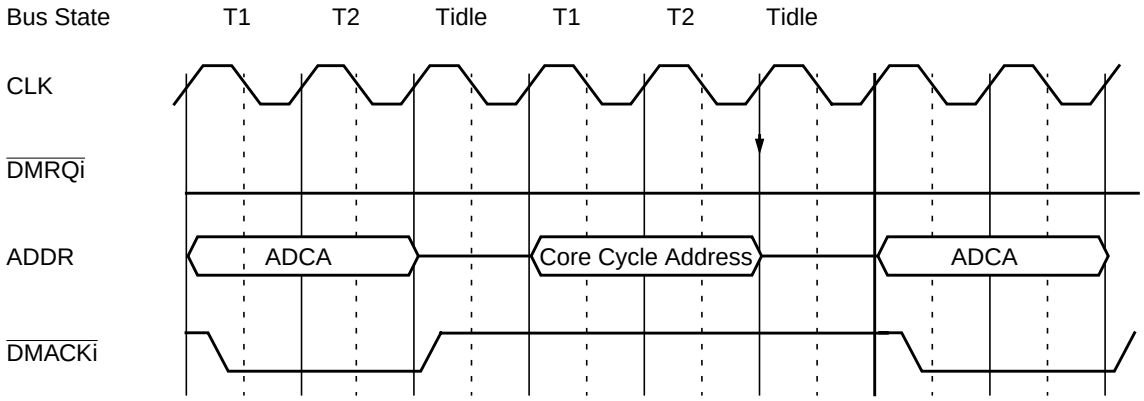



**Figure 29. DMAC Direct Bus Cycles in Intermittent Mode, DMRQ Asserted Constantly.**

When BPC bit in DMACNTLn is 0, channel n is in Intermittent mode. In this mode, the DMAC channel relinquishes the bus after each transaction, regardless of the state of its DMA request input. In this way, the DMAC gives the core (and other DMA channels) a chance to use the bus, even if a DMA device needs the bus for multiple transfers.

### Continuous Operation Mode

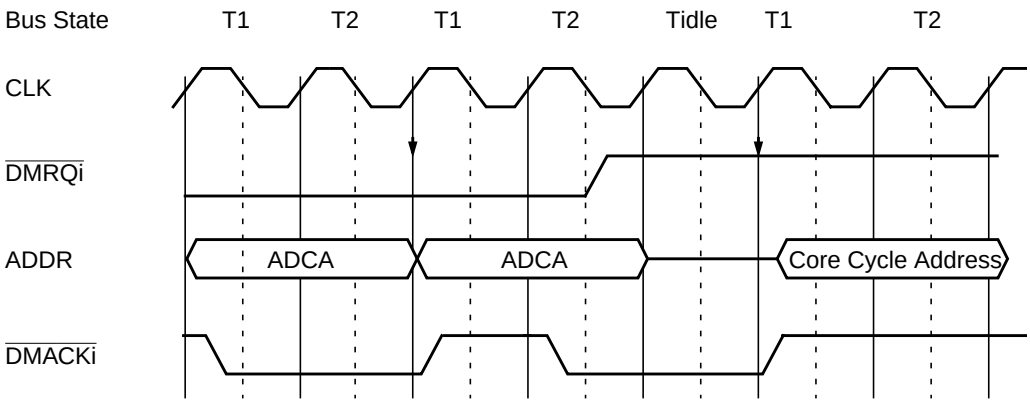



**Figure 30. DMAC Direct Bus Cycles in Continuous Mode**

When BPC bit in DMACNTLn register is 1, channel n is in Continuous mode. In this mode, a DMAC channel uses the bus continuously, as long as its request is active and BLTCn > 0. This allows the channel to utilize the full bandwidth of the bus.

The activity of this channel cannot be interrupted by any other internal bus master, including higher priority DMAC channels. It is the system designer's responsibility to limit the duration of the DMA request to prevent bus starvation.

# 4.0 Embedded Controller Modules (Continued)

## 4.2.6 Operation Modes

The DMAC operates in three different block transfer modes - single transfer, double buffer and auto-initialize. Select the appropriate mode according to the character of the block transfer.

### Single Transfer Operation

This mode provides the simplest way to accomplish a single block data transfer.

**Initialization**

1. Write the two block transfer addresses and byte count into the corresponding ADCAn, ADCBn and BLTCn counters, respectively. The BLTCn counter should be written last.
2. Program the OT bit for Non-Auto-Initialize mode, and clear EOVR bit in DMACNTLn register to 0. Clear to 0 VLD bit in DMASTATn register.
3. Set CHEN bit in DMACNTLn register to 1; the channel activates and responds to DMAC transfer requests.

**Termination**

When the BLTCn counter reaches 0:

- The transfer operation terminates.
- TC bit in DMASTATn register is set to 1, and CHAC is cleared to 0.
- A level interrupt is generated (if enabled by ETC bit in DMACNTLn register).

### Double Buffer Operation

This mode allows the software to set up the next block transfer specification while the current block transfer proceeds. This mode is used for preparing the next buffer for use in a multi-buffer operation (e.g., the alternate buffer in a double-buffer scheme).

**Initialization**

1. Write the two block transfer addresses and byte count into the ADCAn, ADCBn and BLTCn counters, respectively. The BLTCn counter should be written last.
2. Program OT bit in DMACNTLn register for Non-Auto-Initialize mode.
3. Set CHEN bit in DMACNTLn register to 1; the channel activates and responds to DMAC transfer requests.
4. While the current block transfer proceeds, write the addresses and byte count for the next block into the ADRAn, ADRBn and BLTRn registers. The BLTRn register should be written last.

**Continuation / Termination**

When the BLTCn counter reaches 0:

- TC bit in DMASTATn register is set to 1.
- A level interrupt is generated (if enabled by ETC bit in DMACNTLn register).
- The DMAC channel checks the value of VLD bit.

If VLD bit is 1:

- The channel copies the ADRAn, ADRBn and BLTRn values into ADCAn, ADCBn and BLTCn. The BLTCn counter should be written last.
- Clears VLD bit to 0.
- Becomes ready to start the next block transfer (on the next DMA request).

If VLD bit is 0:

- The transfer operation terminates.
- The channel sets OVR bit in DMASTATn register to 1.
- Clears CHAC bit to 0.
- A level interrupt is generated (if enabled by EOVR bit in DMACNTLn register).

**Note:**

ADCB and ADRBn are used only in indirect (memory-to-memory) transfer. In Direct (Fly-By) mode, the DMAC does not use them and therefore does not copy ADRBn into ADCBn.

### Auto-Initialize Operation

This mode allows the DMAC to continuously fill the same memory area without software intervention.

#### Initialization

1. Write the two block addresses and byte count into the ADCAn, ADCBn and BLTCn counters, respectively (the BLTCn counter should be written last); also write them to the ADRAn, ADRBn and BLTRn registers, respectively (the BLTRn counter should be written last).
2. Program OT bit in DMACNTLn register for Auto-Initialize mode.
3. Set CHEN bit in DMACNTLn register to 1; the channel activates and responds to DMAC transfer requests.

#### Continuation

When the BLTCn counter reaches 0:

- The contents of the ADRAn, ADRBn and BLTRn registers are copied to the ADCAn, ADCBn and BLTCn counters, respectively. The BLTCn counter should be written last.
- The DMAC channel checks the value of TC bit.

If TC bit is 1:

- OVR bit in DMASTATn register is set to 1.
- A level interrupt is generated (if enabled by EOVR bit in DMACNTLn register).
- The operation is repeated.

If TC bit is 0:

- TC bit in DMASTATn register is set to 1.
- A level interrupt is generated (if enabled by ETC bit in DMACNTLn register).
- The operation is repeated.

### 4.2.7 Software DMA Request

In addition to the $\overline{\text{DMRQn}}$ signals, a DMA transfer request can also be initiated by software. The software DMA transfer request is used for memory-to-memory block copying (in indirect transfers).

When SWRQ bit in DMACNTLn register is 1, the corresponding DMA channel receives a DMA transfer request. When SWRQ bit is 0, the software DMA transfer request of the corresponding channel is inactive.

For each channel, use the software DMA transfer request, only when the corresponding $\overline{\text{DMRQn}}$ signal is inactive.

### 4.2.8 DMAC Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### DMAC Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| ADCAn | Device A Address Counter Register | R/W |
| ADRAn | Device A Address Register | R/W |
| ADCBn | Device B Address Counter Register | R/W |
| ADRBn | Device B Address Register | R/W |
| BLTCn | Block Length Counter Register | R/W |
| BLTRn | Block Length Register | R/W |
| DMACNTLn | DMA Control Register | R/W |
| DMASTATn | Status Register | R/W |

**Notes:**

- Register names with the suffix n, where n = 0 to 3, are replicated for each channel.
- Unless stated otherwise, bits 21 to 31 are reserved in each of the following registers. Double-word (32-bit) registers may be accessed word-by-word (word aligned).

# 4.0 Embedded Controller Modules (Continued)

## Device A Address Counter Register (ADCAn)

A double-word, read/write register. Holds the current address of either the source data item or the destination location, according to DIR bit in DMACNTLn register. ADCAn is updated after each transfer cycle by INCAn, INCBn, ADA and ADB in DMACNTLn register.

Location: Channel 0 - 00 $FA00_{16}$
Channel 1 - 00 $FA20_{16}$
Channel 2 - 00 $FA40_{16}$
Channel 3 - 00 $FA60_{16}$

Type: R/W

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Device A Address Counter** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

## Device A Address Register (ADRAn)

A double-word, read/write register. Holds the starting address of either the next source data block, or the next destination data area, according to DIR bit in DMACNTLn register.

Location: Channel 0 - 00 $FA04_{16}$
Channel 1 - 00 $FA24_{16}$
Channel 2 - 00 $FA44_{16}$
Channel 3 - 00 $FA64_{16}$

Type: R/W

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Device A Address** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

## Device B Address Counter Register (ADCBn)

A double-word, read/write register. Holds the current address of either the source data item, or the destination location, according to DIR bit in DMACNTLn register. ADCBn is updated after each transfer cycle by INCAn, INCBn, ADA and ADB in DMACNTLn register. In Direct (Fly-By) mode, this register is not used.

Location: Channel 0 - 00 $FA08_{16}$
Channel 1 - 00 $FA28_{16}$
Channel 2 - 00 $FA48_{16}$
Channel 3 - 00 $FA68_{16}$

Type: R/W

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Device B Address Counter** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

## Device B Address Register (ADRBn)

A double-word, read/write register. Holds the starting address of either the next source data block or the next destination data area, according to DIR bit in DMACNTLn register. In Direct (Fly-By*)* mode, this register is not used.

Location: Channel 0 - 00 $FA0C_{16}$
Channel 1 - 00 $FA2C_{16}$
Channel 2 - 00 $FA4C_{16}$
Channel 3 - 00 $FA6C_{16}$

Type: R/W

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Device B Address** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

# 4.0 Embedded Controller Modules (Continued)

## Block Length Counter Register (BLTCn)

A double-word, read/write register. Holds the current number of DMA transfers to be executed in the current block. BLTCn is decremented by one after each transfer cycle. A DMA transfer may consist of one or two bytes according to TCS bit in DMACNTLn register.

Location: Channel 0 - 00 $FA10_{16}$
Channel 1 - 00 $FA30_{16}$
Channel 2 - 00 $FA50_{16}$
Channel 3 - 00 $FA70_{16}$

Type: R/W

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | | | | | | | | | | | **Block Length Counter** | | | | | | | | | | | | | | | |

**Note:** Writing 0 to Block Length Counter field of BLTCn initializes the DMA for $2^{21}$-1 transfers.

## Block Length Register (BLTRn)

A double-word, read/write register. Holds the number of DMA transfers to be executed in the next block. Writing this register, sets VLD bit in DMASTATn register to 1.

Location: Channel 0 - 00 $FA14_{16}$
Channel 1 - 00 $FA34_{16}$
Channel 2 - 00 $FA54_{16}$
Channel 3 - 00 $FA74_{16}$

Type: R/W

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | | | | | | | | | | | **Block Length** | | | | | | | | | | | | | | | |

**Note:** Writing 0 to Block Length field of BLTRn initializes the DMA for $2^{21}$-1 transfers.

## DMA Control Register (DMACNTLn)

A word-wide, read/write register that synchronizes the channel's operation with the programing of the block transfer parameters. On reset, the implemented bits are initialized to 0.

The format of the DMACNTLn register is shown below.

Location: Channel 0 - 00 $FA1C_{16}$
Channel 1 - 00 $FA3C_{16}$
Channel 2 - 00 $FA5C_{16}$
Channel 3 - 00 $FA7C_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **INCB** | | **ADB** | **INCA** | | **ADA** | **SWRQ** |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **BPC** | **OT** | **DIR** | **IND** | **TCS** | **EOVR** | **ETC** | **CHEN** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **CHEN (Channel Enable).** This bit must be set to enable DMA operation on this channel<br>0: Channel disabled (default)<br>1: Channel enabled<br>If CHEN bit in DMACNTLn register is cleared in all channels, the DMA clock is disabled to preserve power. |

## 4.0 Embedded Controller Modules (Continued)

| Bit | Description |
|---|---|
| 1 | **ETC (Enable Interrupt on Terminal Count).** This bit enables a level interrupt, when TC bit is set.<br>0: Interrupt masked (default)<br>1: Interrupt enabled |
| 2 | **EOVR (Enable Interrupt on OVR).** This bit enables a level interrupt, when OVR bit is set.<br>0: Interrupt masked (default)<br>1: Interrupt enabled |
| 3 | **TCS** (**Transfer Cycle Size).** This bit specifies the number of bytes transferred in each DMA transfer cycle. In Direct (Fly-By) mode, undefined results occur if TCS is not equal to the addressed memory bus width.<br>0: Byte wide transfer (default)<br>1: Word-wide (16-bit) transfer |
| 4 | **IND (Direct/Indirect Transfer).** This bit sets the Transfer Type.<br>0: Direct (Fly-By- default)<br>1: Indirect (Memory-to-Memory) |
| 5 | **DIR (Transfer Direction).** This bit specifies the direction of the transfer relative to Device A.<br>0: Device A (pointed to by ADCAn) is the source. In Fly-By mode, a read transaction is initialized.<br>1: Device A (pointed to by ADCAn) is the destination. In Fly-By mode, a write transaction is initialized. |
| 6 | **OT (Operation Type).**<br>0: Single-Buffer mode or Double-Buffer mode enabled (default)<br>1: Auto-Initialize mode enabled |
| 7 | **BPC (Bus Policy Control).** This bit sets the operation type, intermittent (cycle stealing) or continuous (burst).<br>0: Intermittent operation. DMAC channel n relinquishes the bus after each transaction even if the request is still asserted.<br>1: Continous operation. DMAC channel n uses the bus continuously as long as the request is asserted. This mode can only be used for SW DMA requests (i.e., when SWRQ is set). On HW DMA requests, BPC must be set to 0. |
| 8 | **SWRQ (Software DMA Request).**<br>0: Software DMA request is inactive (default)<br>1: Software DMA request is active |
| 9 | **ADA (Device A Address Control).** This bit enables Update of Device A Address.<br>0: ADCAn address unchanged (default)<br>1: ADCAn address incremented or decremented, according to INCA field |
| 11-10 | **INCA (Increment/Decrement ADCAn).**<br>**Bits 11 10 Description**<br>0 0: Increment ADCAn register by 1 (default)<br>0 1: Increment ADCAn register by 2<br>1 0: Decrement ADCAn register by 1<br>1 1: Decrement ADCAn register by 2 |
| 12 | **ADB (Device B Address Control).** This bit enables Update of Device B Address.<br>0: ADCBn address unchanged<br>1: ADCBn address incremented or decremented, according to INCB field |
| 14-13 | **INCB (Increment/Decrement ADCBn).**<br>**Bits 14 13 Description**<br>0 0: Increment ADCBn register by 1 (default)<br>0 1: Increment ADCBn register by 2<br>1 0: Decrement ADCBn register by 1<br>1 1: Decrement ADCBn register by 2 |
| 15 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## DMA Status Register (DMASTATn)

A byte-wide, read with write 1 to clear register that holds the status information for the DMAC channel. On reset, the implemented bits are initialized to 0. The reserved bits always return zero when read.

The format of the DMASTATn register is shown below.

Location: Channel 0 - 00 $FA1E_{16}$
Channel 1 - 00 $FA3E_{16}$
Channel 2 - 00 $FA5E_{16}$
Channel 3 - 00 $FA7E_{16}$

Type: R/W1C

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | **VLD** | **CHAC** | **OVR** | **TC** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **TC**[1] **(Terminal Count).** When set to 1, this bit indicates that the transfer was completed by a terminal count condition (BLTCn register reached 0). |
| 1 | **OVR**[1] (**Channel Overrun).**<br>• In double buffered operation (OT bit in DMACNTLn register is 0): OVR is set to 1 when the present transfer is completed (BLTC = 0), but the parameters for the next transfer (address and block length) are not valid.<br>• In auto initialize operation: (OT bit in DMACNTLn register is 1) OVR is set to 1 when the present transfer is completed (BLTC = 0), but TC bit is still set to 1 (e.g., the software did not serve the last interrupt).<br>• In single buffer operation, this bit is ignored. |
| 2 | **CHAC (Channel Active).** This bit continuously reflects the active or inactive status of the channel and is therefore read only. Data written to CHAC bit is ignored.<br>0: Indicates that the channel is inactive (default)<br>1: Indicates that the channel is active (CHEN bit in DMACNTLn register is 1 and BLTC > 0) |
| 3 | **VLD**[1] **(Transfer Parameters Valid).** This bit specifies whether the transfer parameters for the next block to be transferred are valid.<br>Writing to the BLTRn register sets this bit to 1.<br>It is cleared to 0 in the following cases:<br>• The present transfer is completed, and the ADRAn, ADRBn (Indirect mode only) and BLTR registers are copied to the ADCAn, ADCBn (Indirect mode only) and BLTCn registers, respectively.<br>• Writing 1 to VLD bit; (writing zero has no effect). |
| 7-4 | **Reserved.** |

1. The VLD, OVR and TC bits are sticky (once set by the occurrence of the specific condition, they remain set until explicitly cleared by software). These bits can be cleared individually by writing a value into the DMASTATn register with the bit positions to be cleared set to 1; writing 0 to these bits has no effect.

# 4.0 Embedded Controller Modules (Continued)

## 4.2.9 Usage Hints

- Do not write to ADCAn, ADCBn or BLTCn and do not change the value of TCS, IND, DIR, OT, ADA, INCA, ADB and INCB fields of DMACNTLn register while the associated channel is active (CHAC bit in DMASTATn register is 1). When initializing these registers, write to BLTCn register last, since writing to BLTCn register activates the channel immediately (if CHEN bit in DMACNTLn register is set to 1).
- The ADRAn, ADRBn and BLTRn registers store transfer parameters (source address, destination address and block length) for the next data block to be transferred, for either Auto-Initialize or Double-Buffer modes of operation. When initializing these registers, write the BLTRn register last, since this validates the next block's parameters (VLD bit in DMASTATn register is set to 1).
- The TCS bit in DMACNTLn register is programed according to the bus width of the devices. It determines how many bytes are transferred in each DMA bus cycle.
- The DMAC does not support non-aligned transfers. The values written to ADCAn, ADRAn, ADCBn and ADRBn must be multiples of the Transfer Cycle Size (as defined by TCS bit in DMACNTLn register).

# 4.0 Embedded Controller Modules (Continued)

## 4.3 INTERRUPT CONTROL UNIT (ICU)

The ICU has 31 channels. It interfaces between the different modules' interrupt requests and external interrupt requests and also generates the core interrupt. It generates both maskable and non-maskable interrupts. The ICU has a predetermined scheme that allocates priority.

### 4.3.1 Features

**Non-Maskable Interrupts (NMI)**

- Gathers all edge-triggered non-maskable interrupt sources
  - External Power Fail (PFAIL) interrupt source
- Holds the status of the current pending NMI requests
- Generates non-maskable interrupt (NMI) to the core

**Maskable Interrupts**

- 31 active high-level or edge-triggered interrupt sources
- Core vectored interrupt mode
- Fixed priority among interrupt sources
- Individual enable/disable for each interrupt source
- Polling support by an interrupt status register
- Clear registers for edge-triggered interrupts

### 4.3.2 Non-Maskable Interrupt (NMI)

The Interrupt Control Unit (ICU) gathers external non-maskable interrupt (NMI) sources and generates an NMI interrupt to the core when required.

#### External NMI Inputs

The ICU processes the $\overline{\text{PFAIL}}$ signal to send to the CR16B NMI input.

#### Non-Maskable Interrupt Processing

The CR16B core performs an "Interrupt Acknowledge" bus cycle when beginning to process a non-maskable interrupt. The address associated with this core bus cycle is within the internal core address space and may be monitored as a Core Bus Monitoring (CBM) clock cycle. For additional details, see "Core Bus Monitoring" on page 80 and Section 4.20.6 on page 237.

After reset, NMI is disabled and must remain disabled until the software initializes the interrupt table, interrupt base and the interrupt mode.

The PFAIL interrupt is enabled by setting ENLCK bit and remains enabled until a reset occurs. This allows the external NMI feature to be enabled only after the Interrupt Base Register (IMASK) and the Interrupt Stack Pointer (ISP), in the core, have been set up.

Alternatively, the external PFAIL interrupt can be enabled by setting EN bit, which remains enabled until an interrupt event or a reset occurs. The NMISTAT register holds the status of the current pending NMI request. When the bit in NMISTAT is set to 1, an NMI request to the core is issued. NMISTAT is cleared each time its contents are read. NMI handlers must read the NMISTAT register to allow new NMI events to occur.

Note that PFAIL status bit in NMISTAT register may be set as a result of transient enable conditions on $\overline{\text{PFAIL}}$. To avoid an interrupt to the core, after configuring the $\overline{\text{PFAIL}}$ input for operation, read the NMISTAT register and then enable $\overline{\text{PFAIL}}$ by writing 1 to EN bit in PFAIL register.

#### $\overline{\text{PFAIL}}$ Input

The $\overline{\text{PFAIL}}$ signal is an asynchronous input with Schmitt trigger characteristics and an internal synchronization circuit; therefore, no external synchronizing circuit is needed. The $\overline{\text{PFAIL}}$ signal generates an interrupt on its falling edge.

### 4.3.3 Maskable Interrupts

The ICU receives level or edge-triggered interrupt request signals from 31 sources and generates a vectored interrupt to the CR16B core when required. Priority among the interrupt sources (named INT1 to INT31) is fixed. Each interrupt source can be individually enabled or disabled under software control through:

- ICU interrupt enable registers
- Interrupt enable bits in the peripherals that request the interrupts.

Pending interrupts, enabled or disabled, can be polled using the Status registers. The CR16B core supports INT0, but the ICU reserves INT0 so that it is not connected to any interrupt source.

# 4.0 Embedded Controller Modules (Continued)

## Maskable Interrupt Vectors

Interrupt vector numbers are always positive and are in the range $10_{16}$ to $2F_{16}$. The IVCT register contains the interrupt vector of the enabled and pending interrupt with the highest priority. The interrupt vector $10_{16}$ corresponds to INT0 with the lowest priority; the vector $2F_{16}$ corresponds to INT31 with the highest priority.

The CR16B core performs an “Interrupt Acknowledge” bus cycle on receiving an enabled maskable interrupt request from the ICU. During the interrupt acknowledge cycle, a byte is read from address 00 $FE00_{16}$ (IVCT register). The byte read is used as an index in the Dispatch Table to determine the address of the interrupt handler.

Although INT0 is not connected to any interrupt source, the IVCT register can return the value $10_{16}$. This happens, for example, when the interrupt request is removed before the interrupt acknowledge cycle. The entry in the Dispatch Table should point to a default interrupt handler that handles this error condition.

## Pending Interrupts

Edge-triggered interrupts are latched by the Interrupt Status register. A pending edge-triggered interrupt is cleared by writing a ‘1’ to the respective bit in the Edge Interrupt Clear register, IECLR0 or IECLR1.

A pending level-triggered interrupt is cleared when the interrupt source is not active; note that the interrupt should be cleared at the device/module that drives it by clearing the event status bit.

Interrupt mask bits (IENAM register bits) and pending interrupt bits (ISTAT register bits), should be cleared to 0 only when interrupts are disabled; i.e., when bits I and/or E in PSR register (a core register) are 0. Bits in IENAM may be set at any time.

## Interrupt Priorities

The priorities of the maskable interrupts are hard-wired and thus fixed. The interrupts are named INT0 to INT31, where INT0 has the lowest priority and INT31 has the highest priority.

## Power-Down Modes

Interrupt sources that may generate unexpected interrupts in Idle mode should be masked before switching to the power-down mode.

## External Interrupt Inputs

When an MIWU input is disabled, and the respective WKOxx output at the MIWU is connected to the ICU, the MIWU input is fed directly to the ICU. In this case, the interrupt inputs are asynchronous. They are recognized by the PC87591L-N05 during cycles in which the input setup and hold time requirements are satisfied. To use an external interrupt that is shared with an I/O port, configure the I/O port to its alternate function (see Section 2.4 on page 49).

## Interrupt Assignment

Table 15 shows the mapping of the ICU maskable interrupts to different functions. For information on mask bits and the clear mechanism for the status bit in internal level interrupts, refer to descriptions of the module that is the interrupt source.

**Table 15. ICU Interrupt Assignments**

| INT Number | Source | Type | Details | Priority |
|---|---|---|---|---|
| INT0 | - | - | Error condition occurred (spurious interrupt) | Lowest |
| INT1 | External/MIWU | Level-High | External Interrupt EXWINT20 through the MIWU[1] | |
| INT2 | Internal | Level-High | Host I/F Keyboard/Mouse channel Output Buffer Empty | |
| INT3 | Internal | Level-High | Host I/F Power Management channel 1 or channel 2 Output Buffer Empty | |
| INT4 | Internal | Level-High | High-Frequency Clock Generator | |
| INT5 | Internal | Level-High | MIWU PSWUINT or WKINTD | |
| INT6 | External/MIWU | Level-High | External Interrupt EXWINT23 through the MIWU[1] | |
| INT7 | Internal | Level-High | MFT16 1 Interrupt (INT1 ORed with INT2) | |
| INT8 | Internal | Level-High | ADC interrupt (ADCI) | |
| INT9 | Internal | Level-High | ACCESS.bus 1 interrupt or ACCESS.bus 3 interrupt | |
| INT10 | Internal | Level-High | ACCESS.bus 2 interrupt or ACCESS.bus 4 interrupt | |

## 4.0 Embedded Controller Modules (Continued)

**Table 15. ICU Interrupt Assignments (Continued)**

| INT Number | Source | Type | Details | Priority |
|---|---|---|---|---|
| INT11 | MIWU | Level-High | Internal Keyboard Scan Interrupt (KBSINT from MIWU) or ACM Interrupt | |
| INT12 | Internal/MIWU | Level-High | MSWC interrupt through the MIWU | |
| INT13 | Internal/MIWU | Edge Rising | TWM system tick (T0OUT), through the MIWU[2] | |
| INT14 | External/MIWU | Level-High | SWIN input[1], through the MIWU | |
| INT15 | Reserved | | | |
| INT16 | Internal | Level-High | USART1 Interrupt (TX Int OR RX int) | |
| INT17 | External/MIWU | Level-High | External Interrupt EXWINT24 through the MIWU[1] | |
| INT18 | Internal | Edge Falling | PS/2 shift mechanism (PSINT3) | |
| INT19 | Internal | Edge Falling | PS/2 shift mechanism (PSINT2) | |
| INT20 | Internal | Level High / Edge Falling[3] | PS/2 shift mechanism (PSINT1) | |
| INT21 | External/MIWU | Level-High | External Interrupt EXWINT22 through the MIWU[1] | |
| INT22 | Internal | Level-High | MFT16 2 Interrupt (INT1 ORed with INT2) | |
| INT23 | Internal | Level-High | Shared Memory Interrupt | |
| INT24 | Internal | Level-High | Host I/F Keyboard/Mouse channel Input Buffer Full | |
| INT25 | Internal | Level-High | Host I/F Power Management channel 1 or channel 2 Input Buffer Full | |
| INT26 | Reserved | | | |
| INT27 | Internal | Level-High | DMA Channel 0 (USART1 Rx DMA) | |
| INT28 | Internal | Level-High | DMA Channel 1 (USART1 Tx DMA) or USART2 Interrupt (Tx Int **or** Rx int) | |
| INT29 | Internal | Level-High | DMA Channel 2 (ACCESS.bus 3 DMA) | |
| INT30 | Internal | Level-High | DMA Channel 3 (ACCESS.bus 4 DMA) | |
| INT31 | External/MIWU | Level-High | External Interrupt EXWINT21 through the MIWU[1] | Highest |

1. To enable the external interrupt, set the pin to its alternate function. When used as I/O port signals the External interrupt input is forced to 0
2. When in Active mode, it is advised to disable the T0OUT channel of the MIWU, this saves the need to clear the pending bit in the MIWU on each interrupt.
3. Falling Edge is enabled for this input when the PS/2 channel ‘Disabled Shift Mechanism Interrupts’ are enabled (DSMIE in PSIEN register = 1).

# 4.0 Embedded Controller Modules (Continued)

## 4.3.4 ICU Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### ICU Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| IVCT | Interrupt Vector | RO |
| NMISTAT | NMI Status | RO |
| PFAIL | Power Fail Interrupt Control and Status | R/W |
| ISTAT0 | Interrupt Status 0 | RO |
| ISTAT1 | Interrupt Status 1 | RO |
| IENAM0 | Interrupt Enable and Mask 0 | R/W |
| IENAM1 | Interrupt Enable and Mask 1 | R/W |
| IECLR0 | Edge Interrupt Clear 0 | WO |
| IECLR1 | Edge Interrupt Clear 1 | WO |

### Interrupt Vector Register (IVCT)

The IVCT register holds the vector number of the interrupt vector. IVCT is set to $10_{16}$ on reset.

Location: 00 $FE00_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **0** | **0** | **INTVECT** | | | | | |
| Reset | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 5-0 | **INTVECT (Interrupt Vector).** Contains the encoded value of the enabled pending interrupt with the highest priority; the valid values range from $10_{16}$ to $2F_{16}$. Valid during an interrupt acknowledge core bus cycle in which IVCT is read. It may contain invalid data when INTVECT is updated. |
| 7-6 | These bits return 0 when read. |

### NMI Status Register (NMISTAT)

The NMISTAT register holds the status of the current pending Non-Maskable Interrupt (NMI) request. This register is cleared on reset and each time its contents are read. Refer to the description of the PFAIL register below for additional details.

Location: 00 $FE02_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | | **EXT** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **EXT (External Non-Maskable Interrupt Request).**<br>0: No external non-maskable interrupt request occurred (default)<br>1: External non-maskable interrupt request occurred |
| 7-1 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## Power Fail Interrupt Control and Status Register (PFAIL)

The PFAIL register holds the current value of the PFAIL signal and controls the NMI interrupt generation based on a falling edge of the PFAIL signal. EN and ENLCK are cleared on reset. When writing to this register, all reserved bits must be written with 0 for the device to function properly.

Location: 00 $FE04_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | **ENLCK** | **PIN** | **EN** |
| Reset | x | x | x | x | 0 | 0 | x | 0 |

| Bit | Description |
|---|---|
| 0 | **EN (PFAIL Interrupt Enable).** An NMI interrupt is generated when this bit is set to 1 and the PFAIL signal changes its value from high to low. The bit is cleared by hardware on reset and whenever the interrupt occurs (i.e., when EXT bit in NMISTAT register is set). It can be set and cleared by software; however, software can set this bit only when EXT is cleared. This bit is ignored when ENLCK is set.<br>0: No NMI interrupt generated (default)<br>1: NMI interrupt generated |
| 1 | **PIN (PFAILPin Value).** Contains the current (non-inverted) PFAIL signal value. This bit is read only; data written to it is ignored. |
| 2 | **ENLCK PFAIL Interrupt Enable Lock.** When this bit is set to 1, the external PFAIL feature is enabled and locked; it cannot be cleared by software and can only be cleared by hardware on reset. After setting this bit, an NMI interrupt is generated every time the PFAIL signal changes its value from high to low. Note that when ENLCK is set, EN bit is ignored.<br>0: External PFAIL feature disabled (default)<br>1: External PFAIL feature enabled and locked |
| 7-3 | **Reserved.** |

## Interrupt Status Register 0 (ISTAT0)

This register indicates which maskable interrupts are pending regardless of the state of the corresponding IENA bits. ISTAT0 is cleared on reset.

Location: 00 $FE0A_{16}$

Type: RO

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **IST15-0** | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 15-0 | **IST15-0 (Interrupt Status).** Each bit indicates if an interrupt event was sent to the ICU; IST15 to IST0 correspond to INT15 to INT0, respectively. Since INT0 is not used, IST0 always reads 0. Each bit is encoded as follows:<br>0: Interrupt input to ICU not pending (default)<br>1: Interrupt input to ICU pending |

# 4.0 Embedded Controller Modules (Continued)

## Interrupt Status Register 1 (ISTAT1)

This register indicates which maskable interrupts are pending regardless of the state of the corresponding IENA bits.

Location: 00 $FE0C_{16}$

Type: RO

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **IST31-16** | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 15-0 | **IST31-16 (Interrupt Status).** Each bit indicates if an interrupt event was sent to the ICU; IST31 to IST16 correspond to INT31 to INT16, respectively. Each bit is encoded as follows:<br>0: Interrupt input to ICU not pending (default)<br>1: Interrupt input to ICU pending |

## Interrupt Enable and Mask Register 0 (IENAM0)

This register controls the enable/disable of the maskable interrupt sources INT0 to INT15. The register is cleared ($0000_{16}$) on reset.

Location: 00 $FE0E_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **IENA15-0** | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 15-0 | **IENA15-0 (Interrupt Enable).** Each bit enables or disables the corresponding interrupt request INT0 to INT15; e.g. IENA15 controls INT15. Since INT0 is not used, IENA0 has no effect on the operation of the ICU.<br>0: Interrupt disabled (default)<br>1: Interrupt enabled |

## Interrupt Enable and Mask Register 1 (IENAM1)

This register controls the enable/disable of the maskable interrupt sources INT16 to INT31. The register is cleared ($0000_{16}$) on reset.

Location: 00 $FE10_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **IENA31-16** | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 15-0 | **IENA31-16 (Interrupt Enable).** Each bit enables or disables the corresponding interrupt request INT16 to INT31; e.g. IENA16 controls INT16.<br>0: Interrupt disabled (default)<br>1: Interrupt enabled |

# 4.0 Embedded Controller Modules (Continued)

### Edge Interrupt Clear Register 0 (IECLR0)

The IECLR register is used to clear pending, edge-triggered interrupts.

Location: 00 $FE12_{16}$

Type: WO

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **IEC15-1** | | | | | | | | | | | | | | | **Res.** |

| Bit | Description |
|---|---|
| 0 | **Reserved.** |
| 15-1 | **IEC15-1 (Edge Interrupt Clear).** Each bit clears the corresponding bit in ISTAT0 register. Writing to the bit positions of level-triggered interrupts has no effect. Read always returns $FFFF_{16}$. IEC15 to IEC1 correspond to INT15 to INT1, respectively. Each bit is encoded as follows:<br>0: No effect<br>1: Pending edge-triggered interrupt cleared |

### Edge Interrupt Clear Register 1 (IECLR1)

The IECLR register is used to clear pending, edge-triggered interrupts.

Location: 00 $FE14_{16}$

Type: WO

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **IEC31-16** | | | | | | | | | | | | | | | |

| Bit | Description |
|---|---|
| 15-0 | **IEC31-16 (Edge Interrupt Clear).** Each bit clears the corresponding bit in ISTAT1 register. Writing to the bit positions of level-triggered interrupts has no effect. Read always returns $FFFF_{16}$. IEC31 to IEC16 correspond to INT31 to INT16, respectively. Each bit is encoded as follows:<br>0: No effect<br>1: Pending edge-triggered interrupt cleared |

## 4.3.5 Usage Hints

### Initializing

The recommended initialization sequence is:

1. Initialize both the INTBASE register and the interrupt stack pointer of the core.
2. Prepare the interrupt routines of the interrupts used.
3. Clear edge interrupt used.
4. Set relevant bits of the peripherals.
5. Set relevant bits in IENAM register.
6. Set PFAIL register.
7. Enable core interrupt.

### Clearing

Clearing an interrupt request before it is serviced may cause a spurious interrupt (i.e., when the core detects an interrupt not reflected by IVCT). Clear interrupt requests only when interrupts are disabled. Clear IENAM bits and ISTAT bits while the core interrupts are disabled (i.e., bits I and/or E in PSR register are cleared).

### Nesting

The IENAM registers can be used in interrupt handlers to allow interrupt nesting. When the core acknowledges an interrupt, it disables maskable interrupts by clearing bit I in PSR register and executes the interrupt service routine. This routine can enable nested interrupts by setting bit I in PSR register and can use the IENAM registers to control which interrupts are allowed.

# 4.0 Embedded Controller Modules (Continued)

## 4.4 MULTI-INPUT WAKE-UP (MIWU)

The Multi-Input Wake-Up Unit (MIWU) allows the PC87591L-N05 to exit Idle mode. In addition, it provides signal conditioning and grouping of external interrupt sources. It supports a total of 32 internal and/or external wake-up sources.

### 4.4.1 Features

- Supports up to 32 internal and/or external wake-up inputs
- Generates a wake-up signal
- Generates interrupt signals for:
  - — each of the 32 inputs
  - — one interrupt for each group of eight inputs e.g., I/O ports
- User-selectable trigger condition on each input:
  - — positive edge
  - — negative edge
- Individual enable and pending bits for each input
- Programmable bypass mode connects inputs to ICU without MIWU involvement

### 4.4.2 Operation

**Overview**

The Multi-Input Wake-Up Unit detects a valid software-selectable trigger condition on any of its inputs. On detection of a valid trigger condition, the MIWU generates a wake-up request and/or an interrupt request. The wake-up request is connected to the Power Management module (PMC) and may be utilized to exit the Idle mode and return to Active mode. The interrupt requests are used to signal to the Interrupt Control Unit (ICU) that either an edge-triggered external or internal interrupt condition has occurred. Figure 31 shows a block diagram of the Multi-Input Wake-Up module.

Note that the MIWU module is active while in Idle mode. Note, however, that because all device clocks are stopped in Idle mode, the detection of a trigger condition on an input, and the resulting set of the pending flag, are not synchronous to the system clock.

Table 16 lists the MIWU sources and interrupts used in the PC87591L-N05.

**Table 16. Input Assignments**

| Source | | Destination | |
|---|---|---|---|
| **Name** | **MIWU Input** | **Interrupt Name** | **MIWU Output** |
| PSCLK1 | WUI10 | PSWUINT | WKINTA |
| PSCLK2 | WUI11 | | |
| PSCLK3 | WUI12 | | |
| PSDAT1 | WUI13 | | |
| PSDAT2 | WUI14 | | |
| PSDAT3 | WUI15 | | |
| PSCLK4 | WUI16 | | |
| PSDAT4 | WUI17 | | |
| EXWINT20 | WUI20 | INT1 | WKO20 |
| EXWINT21 | WUI21 | INT31 | WKO21 |
| EXWINT22 | WUI22 | INT21 | WKO22 |
| EXWINT23 | WUI23 | INT6 | WKO23 |
| EXWINT24 | WUI24 | INT17 | WKO24 |
| SWIN | WUI25 | INT14 | WKO25 |
| MSWC Wake-Up | WUI26 | INT12 | WKO26 |
| T0OUT[1] | WUI27 | T0OUTINT | WKO27 |

## 4.0 Embedded Controller Modules (Continued)

**Table 16. Input Assignments (Continued)**

| Source | | Destination | |
|---|---|---|---|
| **Name** | **MIWU Input** | **Interrupt Name** | **MIWU Output** |
| KBSIN0 | WUI30 | KBSINT | WKINTC |
| KBSIN1 | WUI31 | | |
| KBSIN2 | WUI32 | | |
| KBSIN3 | WUI33 | | |
| KBSIN4 | WUI34 | | |
| KBSIN5 | WUI35 | | |
| KBSIN6 | WUI36 | | |
| KBSIN7 | WUI37 | | |
| EXWINT40 | WUI40 | MIWU2 | WKINTD |
| Reserved | WUI41 | | |
| Host Access Wake-Up[2] | WUI42 | | |
| ACCESS.bus 1 Wake-Up[1] | WUI43 | | |
| ACCESS.bus 2 Wake-Up[1] | WUI44 | | |
| EXWINT45 | WUI45 | | |
| EXWINT46 | WUI46 | | |
| Analog Comparators (ACMI) | WUI47 | | |

1. Program the input to detect the rising edge of the input event.
2. The wake-up input is triggered by a host access to one of a selected set of devices in the PC87591L-N05. See “Core Interrupt” on page 305. Configure the WUI26 for rising edge detection.

### Trigger Conditions

Through the WKEDGx registers, the trigger condition on the selected input signal can be selected as either positive edge (low-to-high transition) or negative edge (high-to-low transition).

### Pending Flags

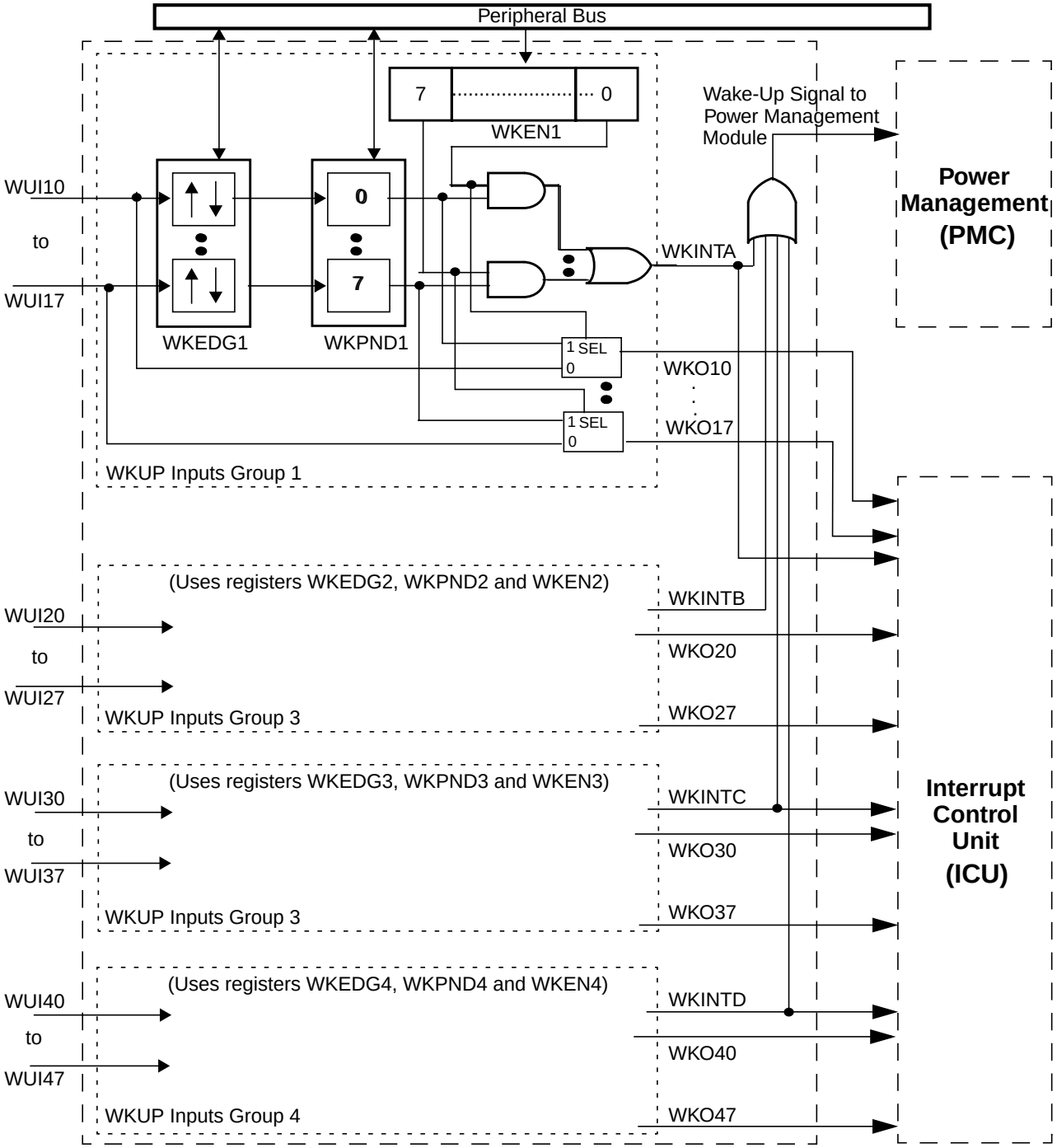
An occurrence of a trigger condition for the Multi-Input Wake-Up input is latched into the respective pending bit in WKPNDx register. The respective bits of WKPNDx are set on an occurrence of the selected trigger edge on the corresponding input signal.

Since the WKPNDx register holds a pending wake-up condition until it is cleared, the device does not enter IDLE mode if any wake-up bit is both enabled and pending. Consequently, the pending flags must be cleared before an attempt to enter Idle mode.

### Input Enable

The MIWU utilizes multiple Multi-Input Wake-Up signals. Set the appropriate bits in the WKENAx registers to select which particular wake-up signal causes the device to exit Idle mode.

## 4.0 Embedded Controller Modules (Continued)




**Note:** Not all MIWU-ICU connections are implemented; for details, see ICU interrupt assignments in Table 16 on page 103.

**Figure 31. Multi-Input Wake-Up Block Diagram**

### Interrupts

The combined output of all pending and enabled channels of the MIWU module generates the wake-up signal, which is fed into both the Power Management Control module (PMC) and the Interrupt Control Unit (ICU). Therefore, each wake-up of the device can be followed by a wake-up interrupt. Since the device can not enter the power reduction mode without having the core execute a “WAIT” instruction, a wake-up interrupt is needed to terminate the “WAIT” instruction on wake-up.

## 4.0 Embedded Controller Modules (Continued)

The MIWU outputs WKO10 through WKO37 may be connected to the Interrupt Control Unit (ICU) to generate an interrupt associated with the specific MIWU output. The WKOxx behaves as follows:

- When the respective WKENxx bit is cleared, the WUIxx is connected to the ICU directly (bypassing the edge detectors and pending bits). The ICU can be configured to use the signal as a level or edge triggered interrupt.
- When the respective WKENxx bit is enabled, the output of the pending bit, WKPDxx, is connected to WKOxx.

**Note:** To enable and disable ICU interrupts generated by their associated WKOxx output of the MIWU, use the ICU mask register in the ICU.

In addition, the MIWU provides four interrupt request lines: WKINTA, WKINTB, WKINTC and WKINTD (see Figure 31 on page 105). These are routed to the ICU (except WKINTB) and can request an interrupt if a valid trigger condition occurred on any of the enabled input sources within the group of eight inputs associated with the interrupt line. For a detailed summary of the interrupts available, see Table 16 on page 103.

### 4.4.3 MIWU Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

#### MIWU Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| WKEDG1 | Edge Detection Register | R/W |
| WKEDG2 | Edge Detection Register | R/W |
| WKEDG3 | Edge Detection Register | R/W |
| WKEDG4 | Edge Detection Register | R/W |
| WKPND1 | Pending Register | R/W |
| WKPND2 | Pending Register | R/W |
| WKPND3 | Pending Register | R/W |
| WKPND4 | Pending Register | R/W |
| WKEN1 | Enable Register | R/W |
| WKEN2 | Enable Register | R/W |
| WKEN3 | Enable Register | R/W |
| WKEN4 | Enable Register | R/W |
| WKPCL1 | Pending Clear Register | WO |
| WKPCL2 | Pending Clear Register | WO |
| WKPCL3 | Pending Clear Register | WO |
| WKPCL4 | Pending Clear Register | WO |

# 4.0 Embedded Controller Modules (Continued)

## Edge Detection Register (WKEDG1)

Byte-wide read/write register that configures the trigger condition of the input signals WUI10 to WUI17. The register is cleared on reset. This configures all associated input signals to be triggered on a rising edge.

Location: 00 $FFC0_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **WKED17-WKED10** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Edge Selection.** For inputs WUI10 through WUI17. Each bit is associated with one of eight inputs.<br>0: Low-to-High transition (default)<br>1: High-to-Low transition |

## Edge Detection Register (WKEDG2)

Byte-wide read/write register that configures the trigger condition of the input signals WUI20 to WUI27. The functionality of the register is identical to the WKEDG1 register described above.

Location: 00 $FFC2_{16}$

Type: R/W

## Edge Detection Register (WKEDG3)

Byte-wide read/write register that configures the trigger condition of the input signals WUI30 to WUI37. The functionality of the register is identical to the WKEDG1 register described above.

Location: 00 $FFC4_{16}$

Type: R/W

## Edge Detection Register (WKEDG4)

Byte-wide read/write register that configures the trigger condition of the input signals WUI40 to WUI47. The functionality of the register is identical to the WKEDG1 register described above.

Location: 00 $FFC6_{16}$

Type: R/W

## Pending Register (WKPND1)

Byte-wide read/write register that latches the occurrence of a selected trigger condition associated with the input signals WUI10 to WUI17. On reset, the WKPND1 register is cleared (0). This indicates that no occurrence of the selected trigger condition is pending.

**Note:** Only software can set the register bits; only the WKPCL1 register can clear them. Writing a 0 to any of the bits leaves their values unchanged. The WKPND1 register format is shown below:

Location: 00 $FFC8_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **WKPD17-WKPD10** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Wake-Up Pending.** If set, (1) indicates that a valid trigger condition occurred on the associated input. |

# 4.0 Embedded Controller Modules (Continued)

### Pending Register (WKPND2)

Byte-wide read/write register that latches the occurrence of a selected trigger condition associated with the input signals WUI20 to WUI27. For a detailed description of the register see the description of the WKPND1 register, above.

Location: 00 $FFCC_{16}$

Type: R/W

### Pending Register (WKPND3)

Byte-wide read/write register that latches the occurrence of a selected trigger condition associated with the input signals WUI30 to WUI37. For a detailed description of the register see the description of the WKPND1 register, above.

Location: 00 $FFD0_{16}$

Type: R/W

### Pending Register (WKPND4)

Byte-wide read/write register that latches the occurrence of a selected trigger condition associated with the input signals WUI40 to WUI47. For a detailed description of the register see the description of the WKPND1 register, above.

Location: 00 $FFD4_{16}$

Type: R/W

### Enable Register (WKEN1)

Byte-wide read/write register that enables the wake-up function of the associated input signal, WUI10 to WUI17. On reset, WDENA1 is cleared (0); this disables the associated input signals. The WKENA1 register format is shown below:

Location: 00 $FFD8_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **WKEN17-WKEN10** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Wake-Up Enable.** If set (1), a valid trigger condition on the associated input generates a wake-up signal or EXTINTx interrupt request. |

### Enable Register (WKEN2)

Byte-wide read/write register that enables the wake-up function of the associated input signal, WUI20 to WUI27. For a detailed description of the register, see the description of the WKEN1 register, above.

Location: 00 $FFDA_{16}$

Type: R/W

### Enable Register (WKEN3)

Byte-wide read/write register that enables the wake-up function of the associated input signal, WUI30 to WUI37. For a detailed description of the register, see the description of the WKEN1 register, above.

Location: 00 $FFDC_{16}$

Type: R/W

### Enable Register (WKEN4)

Byte-wide read/write register that enables the wake-up function of the associated input signal, WUI40 to WUI47. For a detailed description of the register, see the description of the WKEN1 register, above.

Location: 00 $FFDE_{16}$

Type: R/W

# 4.0 Embedded Controller Modules (Continued)

### Pending Clear Register (WKPCL1)

Byte-wide write-only register that controls the clearing (0) of the pending bits associated with the WUI10 through WUI17 inputs. This avoids potential hardware/software collisions during read-modify-write operations. The WKPCL1 register format is shown below:

Location: 00 $FFCA_{16}$

Type: WO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **WKCL17-WKCL10** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Clear Pending Flag.** If a 1 is written to any bit, the associated pending flag located in WKPND1 is cleared (0). Writing a 0 to any bit leaves the value of the corresponding pending flag unchanged. |

### Pending Clear Register (WKPCL2)

Controls the clearing (0) of the pending bits associated with the WUI20 through WUI27 inputs. For a detailed description of the register see, the description of the WKPCL1 register, above.

Location: 00 $FFCE_{16}$

Type: WO

### Pending Clear Register (WKPCL3)

Controls the clearing (0) of the pending bits associated with the WUI30 through WUI37 inputs. For a detailed description of the register, see the description of the WKPCL1 register, above.

Location: 00 $FFD2_{16}$

Type: WO

### Pending Clear Register (WKPCL4)

Controls the clearing (0) of the pending bits associated with the WUI40 through WUI47 inputs. For a detailed description of the register, see the description of the WKPCL1 register, above.

Location: 00 $FFD6_{16}$

Type: WO

## 4.4.4 Usage Hints

1. When changing an edge select, perform the following steps to avoid a spurious wake-up condition, which may occur as a result of the edge change:
   a. Clear the associated WKENxx bit.
   b. Select the required the edge in the WKEDGx register.
   c. Clear the associated WKPDxx bit.
   d. Re-enable the associated WKENxx bit.
2. The correct use of the Multi-Input Wake-Up circuit, which avoids false triggering of a wake-up condition, requires the following sequence of actions. Use the same procedure following a Reset since the wake-up inputs are left floating, producing unknown data on the MIWU input signals.
   a. If the input originates from an I/O port, write to the port alternate function and, if required, direction register to set the pin to interrupt inputs.
   b. Clear the WKENAx register or, if a WKOxx interrupt is used, disable the interrupt via the ICU.
   c. Write the WKEDGx register to select the desired type of edge sensitivity for each of the pins used.
   d. Clear the WKPNDx register to cancel any pending bits.
   e. Either set the WKENxx bits associated with the pins to be used, thus enabling them for the wake-up/interrupt function, or re-enable the interrupt via the ICU.
3. On Reset, the WKEDGx registers are configured to select positive edge sensitivity for all wake-up inputs. To change the edge sensitivity of an input signal while preventing the false triggering of a wake-up/interrupt condition, use the following procedure.
   a. Clear the WKENxx bit associated with the WUIxx input to disable that input.
   b. Write to the WKEDGx register to select the new type of edge sensitivity for the specific input.
   c. Clear the WKPDxx bit associated with the WUIxx input.
   d. Set the WKENxx bit associated with the WUIxx input to re-enable it.

# 4.0 Embedded Controller Modules (Continued)

## 4.5 GENERAL-PURPOSE I/O (GPIO) PORTS

The PC87591L-N05 includes four types of General-Purpose I/O (GPIO) ports: Px, Py, Pz and Pw.

- Px signals: Each signal is bidirectional and can be configured as input or output. An internal weak pull-up is provided to hold the pin high when used as an input or in an open-drain scheme.
- Py signals: Each signal is input only. An internal weak pull-up is provided to hold the pin high.
- Pz signals: Each signal is output only. It may be configured to work as totem-pole or in an open-drain scheme.
- Pw signals: Each signal is bidirectional and can be configured for input or output. The Pw pins may be shared with development system functions. These ports can be implemented off-chip in DEV environment using external logic.

The GPIO signals are organized in ports. Each port is either 8-bits or 16-bits wide. In ports where not all eight bits are used, some of the register's bits are reserved. Some GPIO signals share their pins with one or more alternate functions. A configuration bit selects which function is active (see Section 2.4 on page 49).

### GPIO Port Functionality

The PC87591L-N05 provides 92 GPIO pins. They are subdivided into the following groups:

- Ports IOPA(7-0), IOPB(7-0), IOPC(7-0), IOPD(7-0), IOPQ(2-0) and IOPF(7-0)
  These ports are on-chip, General-Purpose Input/Output (GPIO) ports (type Px).

  IOPC0 is reserved for power supply control use. Bit 0 of PCALT, PCDIR, PCWPU and PCDOUT are reset on $V_{CC}$ Power-Up reset and Watchdog reset only.

  IOPB5 and IOPB6 have an option for automatic TRI-STATE based on $\overline{\text{LPCPD}}$. See "MSWC Control Status Register 3 (MSWCTL3)" on page 293 for the enable function.

  IOPB6 is selected to its alternate function, by default (i.e., bit 6 is set to 1).

  Ports IOPA4-0, IOPB2-0, IOPC0 and IOPD3 have the option to echo the value of the associated input. For the exact echo matrix specifications, see Section 2.4.3 on page 56.

  Bit 5 of PBALT register, bit 0 of PCALT register and bits 4-7 of PDALT register are read only (RO) and return a value of zero.
- IOPE(7-0) and KBSIN(7-0)
  These are General-Purpose Input (GPI) ports (type Py). IOPE(3-0) and IOPE5 do not implement the pull-up function, and the respective bits in PEWPU are reserved. KBSIN has no alternate function; thus it has no PyALT register.
- KBSOUT(15-0), IOPQ3
  This is a 16-bit General-Purpose Output (GPO) port (type Pz). Since KBSOUT has no alternate functions, its alternate function register is not implemented. The reset value of KBSOUT register is $FFFF_{16}$. KBSOUT has open-drain output drivers.

  IOPQ3 is selected to its alternate function (CLK) by default (i.e., bit 3 of PQATL register is set to 1 after reset).
- Ports IOPJ(7-2), IOPL(4-3) and IOPM(7-0)
  These ports are on-chip, General-Purpose Input/Output (GPIO) ports (type Pw).
- When the analog function is enable, the Read function, for GPIO signals that are multiplexed with analog functions, is disabled; this affects signals KBSIN(7-0) and IOPE(3-0).
  When the BIU function is enabled, the Read function, for GPIO signals that are multiplexed with BIU signals, is disabled; this affects signals IOPL(4-3) and IOPM(7-0).

### 4.5.1 Features

- General-Purpose Input/Output (GPIO) Port (Px).
  - Each pin functions as input or output signal.
  - Direction register controls the port direction.
  - Weak pull-up.
  - Read-back on all registers.
- General-Purpose Input (GPI) Port (Py).
  - All port pins function as input signals.
  - Weak pull-up.
  - Read-back on all control registers.
- General-Purpose Output (GPO) Port (Pz).
  - All pins function as output signals.
  - Read-back on all registers.

## 4.0 Embedded Controller Modules (Continued)

- General-Purpose Input/Output (GPIO) Port (Pw).
  - — DEV environment support.
    - — Pins shared with DEV environment signals.
    - — Off-chip implementation supported in DEV environments.
    - — Binary and cycle-by-cycle compatibility between environments.
  - — Each pin functions as input or output signal.
  - — Direction register controls the port direction.
  - — Read-back on all registers.
- Each I/O pin can be configured as a GPIO port or as an alternate function.
  - — Some I/O or input pins can provide interrupt functions.

### 4.5.2 GPIO Port Px

#### Bidirectional Port with Alternate Function

The GPIO port enables access to input and output pins. It also controls the pin usage either as an I/O port or in its alternate function. Figure 32 shows this functionality.

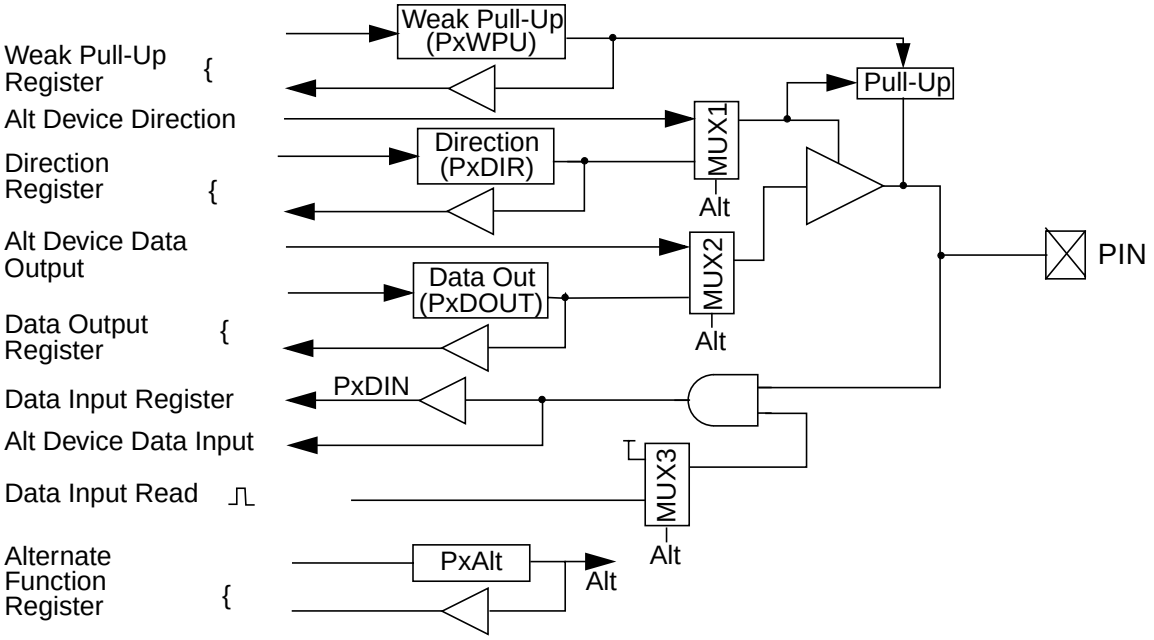



**Figure 32. GPIO Port Px Schematic Diagram**

#### Output Buffer

The output buffer is a TRI-STATE buffer. The output type (i.e., CMOS or TTL) and its driving capabilities are described in Section 2.2 on page 38.

#### Input Buffer

The I/O port input buffer characteristics are defined in Section 2.2 on page 38.

The input buffer has an enable input. When enabled, the buffer inputs the pin's logic level to the on-chip modules. When disabled, the input is blocked to prevent supply leakage currents.

#### Weak Pull-Up

The weak pull-up is enabled when the corresponding bit of PxWPU is set (1) and the pin is configured as an input port. When the pin is configured for input, this pull-up can prevent the input from being in an undefined state. When the pin is configured as an output port, this pull-up is disabled.

## 4.0 Embedded Controller Modules (Continued)

### Alternate Function

The PxALT controls the use of each of the port pins for GPIO or for the pin's respective alternate function.

When PxALT bit is cleared (0):

- The corresponding pin is used as a GPIO pin.
- The output buffer is controlled by the Direction and Data Output registers.
- The input buffer is routed to the Data Input register.
  In this case, the input buffer is blocked, except when the buffer is actually being read.
- The pull-up is enabled when both the PxWPU is set and the device puts the output buffer in TRI-STATE.

When a bit in PxALT is set (1):

- The corresponding pin is used for an alternate function (i.e., a signal from/to some other PC87591L-N05 module).
- The output buffer data and TRI-STATE are controlled by signals from the alternate module.
- The input buffer is always enabled when the alternate function is an input; therefore, to minimize current consumption, the signal should be held above $V_{CC}$–0.2 or below GND+0.2V.
- The pull-up is enabled when PxWPU is set and the device puts the output buffer in TRI-STATE.

### Port Direction

The Port Direction register (PxDIR) controls the direction of the port. If set (1), each bit in the register causes the corresponding port signal to serve as an output port, thus enabling the output buffer.

When cleared, the port serves as an input port signal, thus putting the output buffer in TRI-STATE.

If the corresponding bit in PxWPU is set, it also enables the pull-up.

### Data Output

The Data Output (PxDOUT) register holds the data to be driven onto the pin, when the respective pin is configured as GPIO and its direction is set as output.

### Data Input

The Data Input (PxDIN) register returns the current value/state of the pin. This register can always be read.

### Open Drain

To use the GPIO pin as an inverting open-drain output buffer, the software should clear the corresponding bit in PxDOUT register and then use PxDIR register to set the value to the port pin.

When the signal direction is set as output (1), a value of 0 is forced. When the direction is set for input (0), the signal is in TRI-STATE and is not forced low.

The internal weak pull-up can pull the signal high when it is not forced low, by writing (1) to the corresponding bit of PxWPU.

### Input Echo Function

Some of the Px pins can echo the value of an input pin. Figure 33 shows the modified structure of Px port pins that support the echo function. The input echo function may be used when the Px pin is configured to operate as a General-Purpose output pin.

Table 7 on page 56 defines pairs of input ports (Pi) and output ports (Po), and the Echo Enable bit associated with each pair. When the pair's Echo Enable bit is set, and if Po is configured as output, the value from the input bit (Pi) is output to the respective output port (Po). When the pair's Echo Enable bit is cleared, and if Po is configured as output, the value in PxDOUT is output to the respective output port.

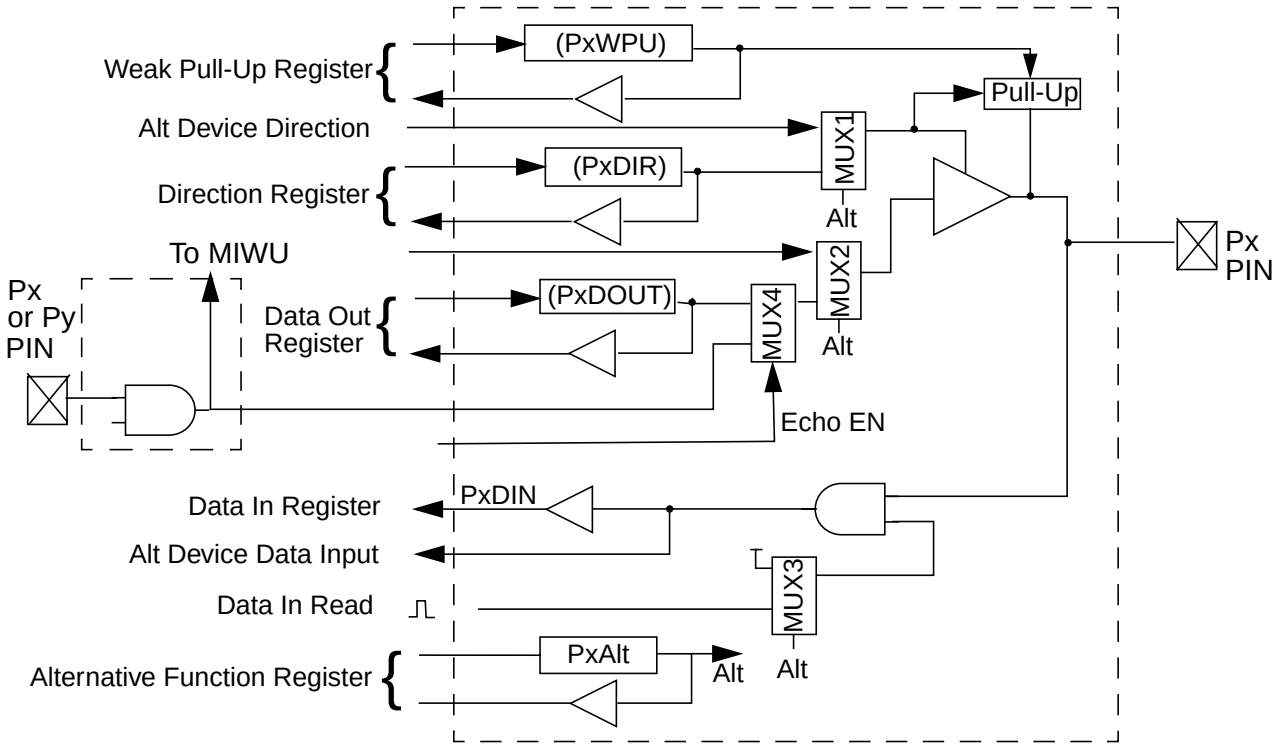



**Figure 33. GPIO Port Px Output with Echo Schematic Diagram**

### 4.5.3 GPI Port Py

#### Input Only Port with Alternate Function

The General-Purpose Input (GPI) port (Py) contains a subset of the GPIO functions. It can be used as a GPI port or as an input signal for an alternate function. Figure 34 shows its functionality.

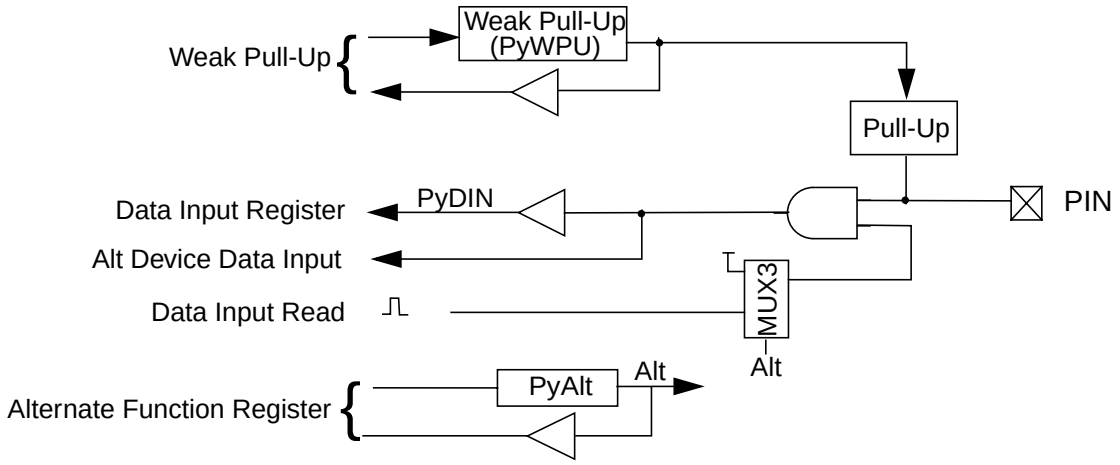



**Figure 34. GPI Port Py Schematic Diagram**

#### Input Buffer

The input buffer characteristics are defined in Section 2.2 on page 38. The input buffer has an enable input. When enabled, the buffer inputs the pin's logic level to the on-chip modules. When disabled, the input is blocked to prevent supply leakage currents.

**Weak Pull-Up**

The weak pull-up is enabled when the corresponding bit of PyWPU is set (1). This pull-up can prevent the input from being in an undefined state.

**Alternate Function**

PyALT controls the use of each of the port pins for GPI or for the pin's respective alternate function.

When a PxALT bit is cleared (0):

- The corresponding pin is used as a GPI pin.
- The input buffer is routed to the Data Input register.
  In this case, the input buffer is blocked, except when the buffer is actually being read.
- The pull-up is enabled when PyWPU is set.

When a bit in PxALT is set (1):

- The corresponding pin is used for an alternate function (i.e., a signal to some other PC87591L-N05 module).
- The input buffer is always enabled; therefore, to minimize current consumption, the signal should be held above $V_{CC}$–0.2 or below GND+0.2V.)
- The pull-up is enabled when both the PyWPU is set and the output buffer is put in TRI-STATE.

**Data Input**

The Data Input (PyDIN) register returns the current value/state of the pin. This register can always be read.

### 4.5.4 GPO Port Pz

### Output Only Port with Alternate Function

The General-Purpose Output (GPO) port (Pz) is a subset of the GPIO functions. It enables the port to be used as a GPO port or as an output signal for an alternate function. Figure 35 shows its functionality.

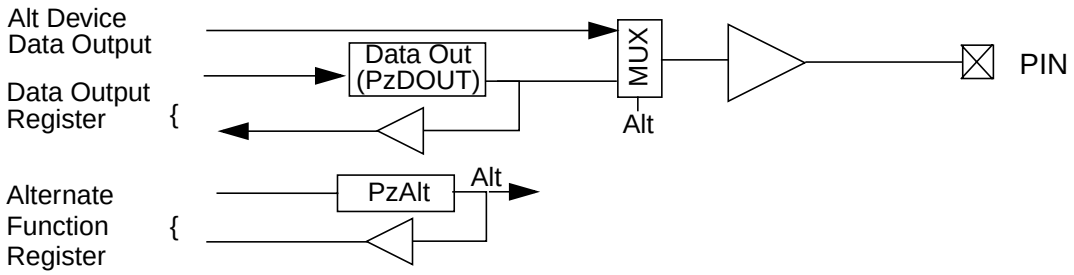



**Figure 35. GPO Port Pz Schematic Diagram**

**Output Buffer**

The output buffer is a TRI-STATE buffer. The output type (i.e., CMOS or TTL) and its driving capabilities are described in Section 2.2 on page 38.

**Alternate Function**

The PzALT controls the use of each of the port pins for GPO or for the pin's respective alternate function.

When a PzALT bit is cleared (0):

- The corresponding pin is used as a GPO pin.
- The output buffer is controlled by the Data Output register.

When a bit in PzALT is set (1):

- The corresponding pin is used for an alternate function (i.e., a signal from or to some other module of the PC87591L-N05).
- The output buffer data is controlled by signals coming from the alternate module.

**Data Output**

The Data Out (PzDOUT) register holds the data to be driven onto the pin.

# 4.0 Embedded Controller Modules (Continued)

## 4.5.5 GPIO Port Pw

### GPIO Port Signals Shared with Development System Signals

The Pw GPIO port enables access to input and output pins. Depending on the operating environments, some pins of the I/O ports may be dedicated to functions other than input or output. In this case, low-cost external logic can be used to perform the I/O functions with binary and cycle-by-cycle compatibility (see Section 4.1.8 on page 78 for details of the Expansion I/O protocol). The Alternate Functions Table (Table 6 on page 49) defines the environment in which each port is implemented off-chip and the pin that performs its alternate function.

The I/O Expansion protocol is used to access the off-chip implementation of the port's registers (PwDIR, PwDOUT, PwDIN), when the port pins are used by DEV environment. This enables binary compatibility between all environments.

To enable cycle-by-cycle compatibility in all environments, the access time to any of the registers is identical for on-chip and off-chip implementation of the ports (i.e., as configured for the BIU I/O zone).

Figure 36 shows its functionality.

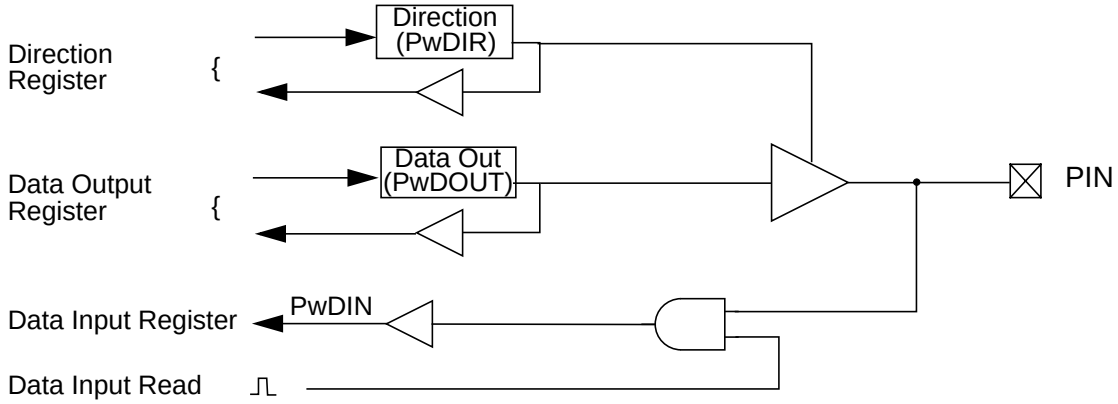



**Figure 36. GPIO Port Pw Schematic Diagram**

**Output Buffer**

The output buffer is a TRI-STATE buffer. Its output type (i.e., CMOS or TTL) and its driving capabilities are described in Section 2.2 on page 38.

**Input Buffer**

The I/O port input buffer characteristics are defined in Section 2.2 on page 38. The input buffer has an enable input. When enabled, the buffer inputs the pin's logic level to the on-chip modules. When disabled, the input is blocked to prevent supply leakage currents.

**Port Direction**

The Port Direction register (PwDIR) controls the direction of the port. When set (1), each bit in PwDIR register causes the corresponding port signal to serve as an output port, thus enabling the output buffer. When cleared, the port serves as an input port signal, thus putting the output buffer in TRI-STATE.

**Data Output**

The Data Output (PwDOUT) register holds the data to be driven onto the pin when the corresponding pin is set as GPIO and its direction is set as output.

**Data Input**

The Data Input (PwDIN) register returns the current value/state of the pin. This register can always be read.

**Open Drain**

To use the GPIO pin as an inverting open-drain output buffer, the software should clear the corresponding bit in Data Output (PwDOUT) register and then use the Direction register to set the value to the port pin.

When the signal's direction is set as output (1), a value of 0 is forced. When the direction is set for input (0), the signal is in TRI-STATE and is not forced low.

# 4.0 Embedded Controller Modules (Continued)

## 4.5.6 GPIO Port Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### GPIO Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| PxALT | Port Px Alternate Function | R/W |
| PyALT | Port Py Alternate Function | R/W |
| PzALT | Port Pz Alternate Function | R/W |
| PxDIR | Port Px Direction | R/W |
| PwDIR | Port Pw Direction | R/W |
| PxDOUT | Port Px Data Output | R/W |
| PzDOUT | Port Pz Data Output | R/W |
| PwDOUT | Port Pw Data Output | R/W |
| PxDIN | Port Px Data Input | RO |
| PyDIN | Port Py Data Input | RO |
| PwDIN | Port Pw Data Input | RO |
| PxWPU | Px Weak Pull-Up | R/W |
| PyWPU | Py Weak Pull-Up | R/W |

### Port Alternate Function Registers (PxALT, PyALT and PzALT)

These registers control the use of each of the Px, Py and Pz pins, respectively, as GPIO ports or as alternate functions.

- When cleared (0), each bit in PxALT, PyALT or PzALT enables the corresponding pin as a GPIO signal.
- When set (1), each bit enables the corresponding pin as an alternate function.

These registers are cleared (0) on reset, except when otherwise noted in Appendix A on page 367.

Location: See Appendix A

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Px Pins Alternate Function Enable** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Py Pins Alternate Function Enable** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Pz Pins Alternate Function Enable** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

# 4.0 Embedded Controller Modules (Continued)

## Port Direction Registers (PxDIR and PwDIR)

These registers configure the direction of the Px and Pw pins.

- When cleared (0), each bit in PxDIR or PwDIR defines the corresponding pin as input.
- When set (1), each pin is defined as output.

PxDIR and PwDIR are cleared (0) on reset except when otherwise noted in Appendix A. Clearing these registers configures all the pins in port Px and Pw as input. Some specific reset values differ, as described in Appendix A on page 367.

Location: See Appendix A

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Px Port Direction** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Pw Port Direction** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## Port Data Output Register (PxDOUT, PzDOUT and PwDOUT)

Writing to PxDOUT, PzDOUT or PwDOUT registers sets the values of the output pins in ports Px, Pz and Pw, respectively. Reading from one of these registers returns the last value written to the register.

Location: See Appendix A

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Px Port Output Data** | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Pz Port Output Data** | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Pw Port Output Data** | | | | | | | |

## Port Data Input Registers (PxDIN, PyDIN and PwDIN)

Reading from PxDIN, PyDIN or PwDIN returns the current value of the pins in port Px, Py and Pw, respectively.

Location: See Appendix A

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Px Port Input Data** | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Py Port Input Data** | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Pw Port Input Data** | | | | | | | |
| Reset | | | | | | | | |

# 4.0 Embedded Controller Modules (Continued)

## Port Weak Pull-Up Registers (PxWPU, PyWPU)

These registers control the pull-up for the related pin, when it used either as GPIO or in its alternate function. The pull-up is enabled when the corresponding bit of PxWPU or PyWPU is set and the port buffer is in TRI-STATE. Otherwise, the pull-up is disabled (i.e., high impedance).

On reset, PxWPU or PyWPU is cleared (0), disabling all pull-ups.

Location: See Appendix A

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Px Port Weak Pull-Up Enable** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Py Port Weak Pull-Up Enable** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

# 4.0 Embedded Controller Modules (Continued)

## 4.6 PS/2 INTERFACE

The PS/2 protocol is an industry-standard, PC-AT-compatible interface for keyboards. It uses a two-wire bidirectional TTL interface for data transmission. Several vendors also supply PS/2 mouse products and other pointing devices that employ the same type of interface.

The PC87591L-N05 provides four PS/2 data transfer channels. Each channel has two quasi-bidirectional signals that serve as direct interfaces to an external keyboard, mouse or any other PS/2-compatible pointing device. Since the four channels are identical, the connector ports are interchangeable.

### 4.6.1 Features

- Four PS/2 channels
- Enable/Disable for each of the four channels
- Automatic hardware shift mechanism
- Hardware support for PS/2 auxiliary device protocol
- Processor interrupts at the beginning and end of data transfer
- Optional software-based PS/2 implementation

### 4.6.2 General Description

In the previous generation of keyboard controllers, firmware executed the PS/2 device interface by toggling the interface signals. The PC87591L-N05 supports this bit toggling mode via either polling or interrupt-driven clock edge detection.

PS/2 devices' firmware is significantly simplified through the use of a hardware accelerator mechanism. The accelerator includes an 8-bit shift register, a state-machine and control logic that handle both the incoming and outgoing data. It reduces code overhead, performance requirements and reduces the overall interrupt latency from the core firmware. The hardware is designed to meet the PS/2 device interface as defined in *Keyboard and Auxiliary Device Controller (Types 1 and 2), August 1988.*

**Section Naming conventions**

- In this section, the term "channel" describes the interface to one of the PS/2 devices and its two associated signals (clock and data).
- The term "shift mechanism" refers to the hardware accelerator.
- The term "PS/2 interface" refers to the entire mechanism.

**Interface Signals**

The PS/2 interface includes eight external signals (PSCLK4-1 and PSDAT4-1) and six registers.

**Module Block Diagram**

A schematic description of the PS/2 interface appears in Figure 37. The interface to the three channels is symmetric and only channel 1 is detailed in the figure.

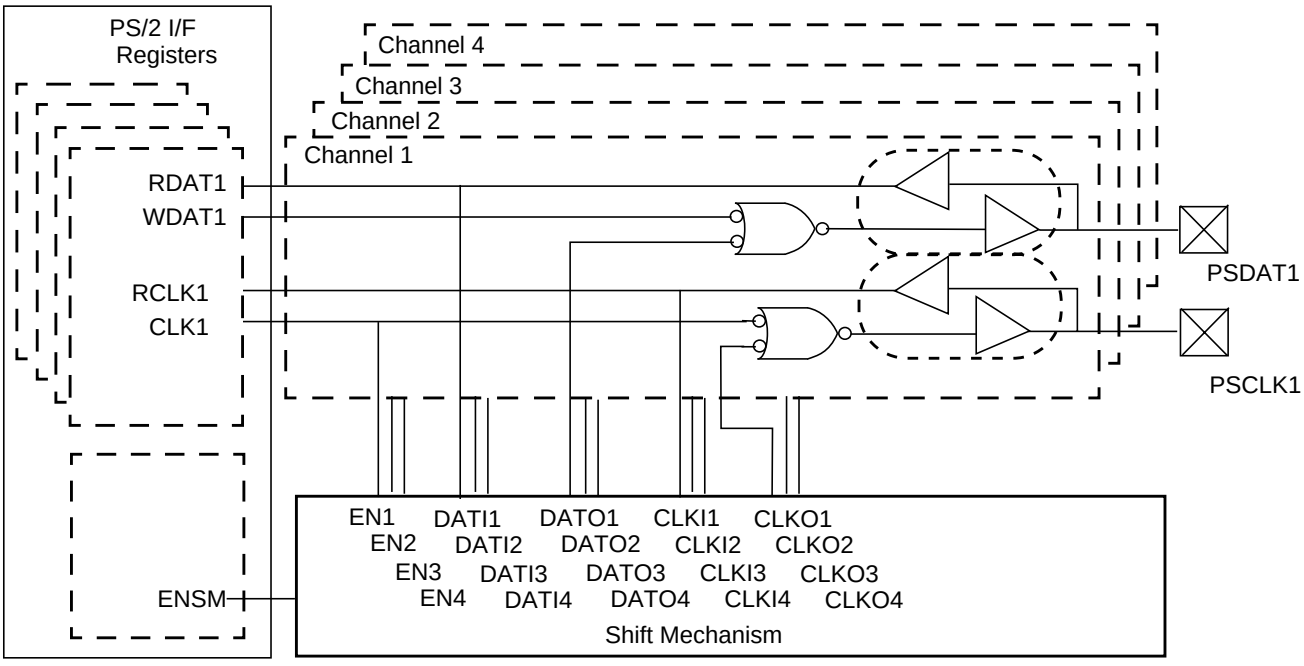



**Figure 37. PS/2 Interface Functional Diagram**

## 4.0 Embedded Controller Modules (Continued)

### Quasi-Bidirectional Drivers

The quasi-bidirectional drivers have an open-drain output (Q2), an internal pull-up (Q3) and a low-impedance pull-up(Q1). Q2 pulls the signal low whenever the output buffer data is '0'. The weak pull-up (Q3) is active whenever the output buffer data is '1' and WPUEN in PSCON register is set (1). The low impedance pull-up is active whenever the PC87591L-N05 changes the output data buffer from '0' to '1', thereby reducing the low-to-high transition time. The length of time that the low-impedance pull-up is active is determined by HDRV field in PSCON register. A schematic description of this output driver appears in Figure 38.

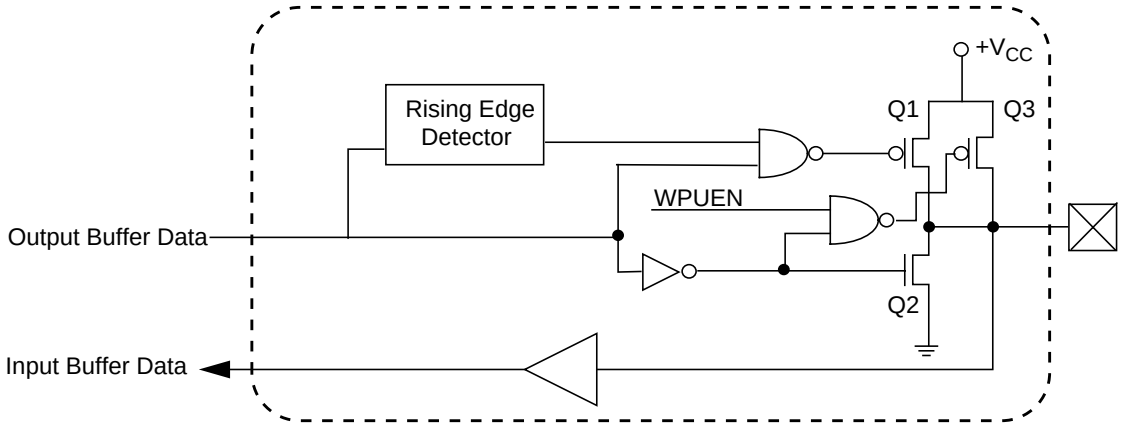



**Figure 38. Quasi-Bidirectional Buffer**

### Interrupt Signals

The firmware can use an interrupt-driven scheme to implement the PS/2 device interface. When the shift mechanism is not in use, four interrupts are available (PSINT4-1), one for each channel. (PSINT4 is not connected to the ICU as a separate input, the MIWU PSCLK4 input should be used instead). When the shift mechanism is in use, only one interrupt signal is used (PSINT1). More details on the use of the interrupts are provided in Section 4.6.4 on page 121. Figure 39 shows the interrupt scheme with the associated enable bits.

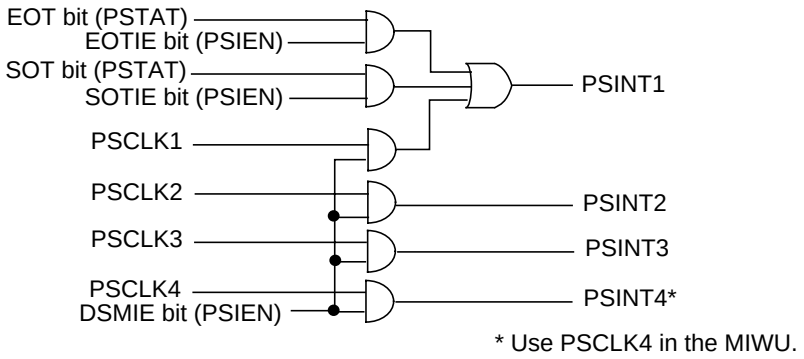



**Figure 39. PS/2 Interface Interrupt Signals**

### Power Modes

The PS/2 interface is active only when the PC87591L-N05 is in Active mode. The shift mechanism should be disabled before entering Idle mode. In Idle mode, the state of output signals cannot be changed (i.e., the firmware cannot write to PSOSIG register, and the shift mechanism does not function).

When the PC87591L-N05 needs to wake up on a Start bit detection by the MIWU, the PS/2 channels that may serve as wake-up event sources must be enabled before entering Idle mode. To enable them, set to 1 their corresponding CLK bits in PSOSIG register.

The MIWU module can be used to identify a start bit in Idle mode and to return the PC87591L-N05 to Active mode. The MIWU receives PSCLK4-1 and PSDAT4-1 signals as inputs (see Table 16 on page 103). The MIWU should be programed to identify a falling edge on the clock or data lines of the enabled channels. In this configuration, a start bit causes the PC87591L-N05 to switch from Idle mode to Active mode. Once Active mode is reached, the firmware should cancel the transaction just started and then enable re-transmission of the information by the device.

## 4.0 Embedded Controller Modules (Continued)

**PS/2 Interface Operation**

The PS/2 interface has two basic operating methods: with the shift mechanism disabled and with the shift mechanism enabled. The following sections describe how to use the PS/2 interface with each of these operating methods.

### 4.6.3 Operating With the Shift Mechanism Disabled

The shift mechanism is disabled when EN bit in PSCON register is cleared (0). In this state, the PS/2 clock and data signals are controlled by the firmware, which performs the PS/2 protocol by manipulating the PS/2 clock and data signals.

**Clock Signal Control**

CLK4-1 bits in PSOSIG register control the value of the respective clock signals (PSCLK4-1). When one of these bits is cleared (0), the relevant pin is held low. When set (1), the open-drain output is open and the respective clock signal is either floating or held high by the pull-up. In this case, an external device can force the respective clock signal low.

When reading PSISIG register, bits RCLK4-1 indicate the current state of the corresponding clock signal.

**Data Signal Control**

WDAT4-1 bits in PSOSIG register control the value of the respective data signals (PSDAT4-1). When one of these bits is cleared (0), the relevant data signal is held low. When set (1), the open-drain output is open and the respective data signal is held high by the pull-up. In this case, an external device can force the respective data signal low.

When reading PSISIG register, bits RDAT4-1 indicate the current state of the corresponding data signal.

**Interrupt Generation**

When DSMIE bit in PSIEN register is set (1), the clock input signals are connected to the Interrupt Control Unit (ICU) for an interrupt driven PS/2 protocol. The four interrupts that are generated are PSINT4-1 for channels 4-1, respectively.

The ICU should be programed to detect a falling edge on each of the clock signals. Disabling a channel by writing 0 to the clock control signals (CLK4-1) may cause a falling edge on a clock signal. When such an interrupt is not desired, clear the clock control bit (0); then clear the respective pending bit in the ICU (or in the MIWU, for PSINT4). This should be done while interrupts are disabled. For more details about the ICU, see Section 4.3 on page 96.

### 4.6.4 Operating With the Shift Mechanism Enabled

The shift mechanism is designed to off load the bit level handling of the data transfer from the firmware to a hardware scheme; this improves system tolerance to interrupt latency. The mechanism includes a shift register and a state machine that controls the PS/2 protocol.

Figure 40 shows the shift mechanism PS/2 data transfer sequence. There are three basic modes: Disabled, Receive and Transmit. Different states in each mode define the progress of the data transfer. The rest of this section details the use of the shift mechanism for implementing a PS/2 data transfer.

## 4.0 Embedded Controller Modules (Continued)

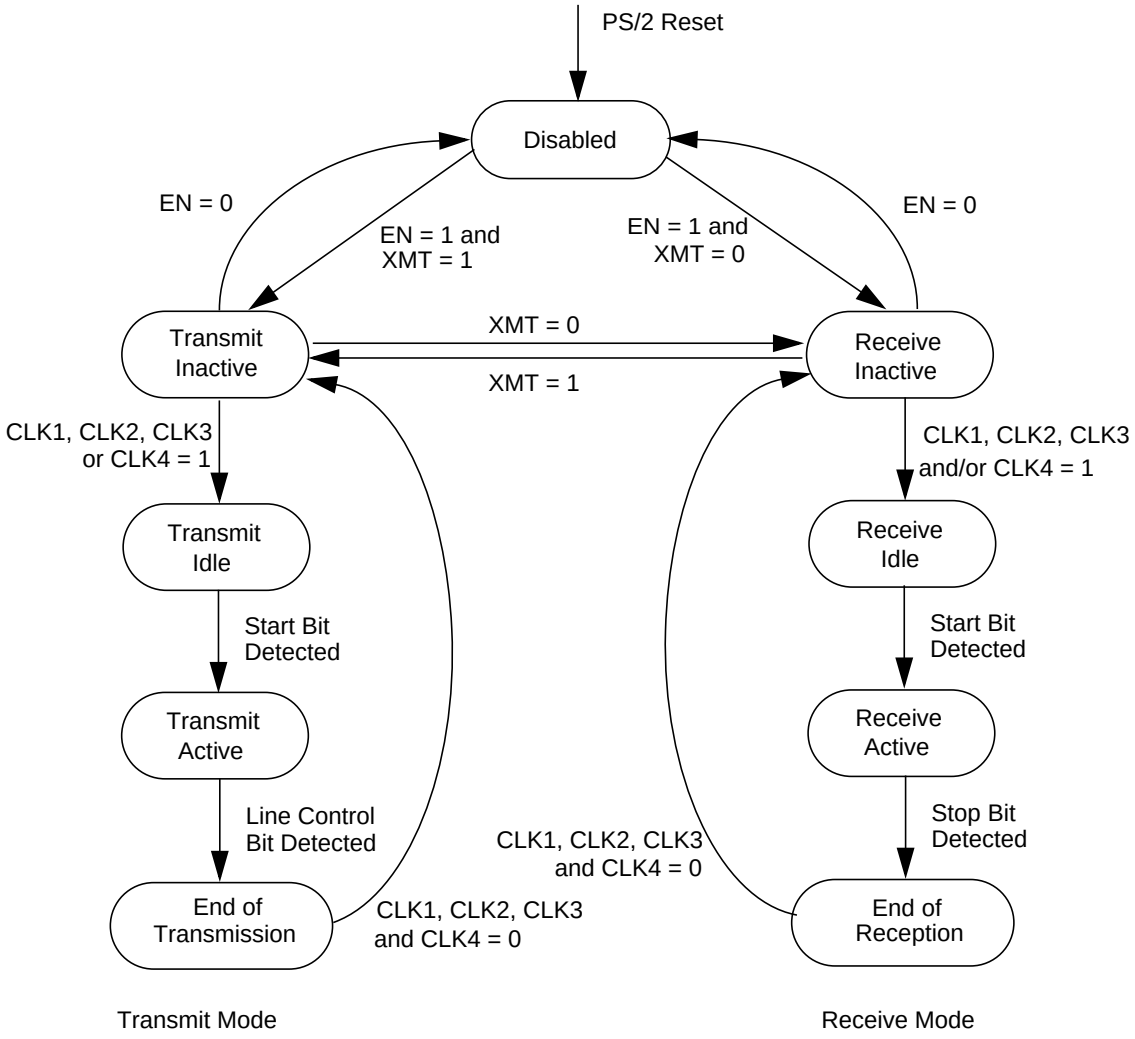



**Figure 40. Shift Mechanism State Diagram**

### Reset the Shift Mechanism

Clearing either the shift mechanism enable bit (EN = 0 in PSCON register) or all the channels' clock bits (CLK4-1 = 0) resets the shift mechanism. In this state, PSTAT register is cleared ($00_{16}$), and the state of the PS/2 clock and data signals (PSCLK4-1 and PSDAT4-1) is set according to the value of their control bits (CLK4-1 and WDAT4-1, respectively).

When the shift mechanism is reset while in an unknown state or while in Transmit Idle state, the firmware should set (1) WDAT4-1 before the shift mechanism is reset.

Before disabling the shift mechanism, the software should clear (0) CLK4-1 to prevent glitches on the clock signals.

### Enable the Shift Mechanism

To enable the shift mechanism, verify that PSOSIG register is set to $47_{16}$ and then set (1) EN bit in PSCON register. This puts the shift register state machine in Receive Inactive or Transmit Inactive state (XMT is 0 or 1, respectively, in PSCON register). In either of these states, the clock signals (PSCLK4-1) are low and the data signals PSDAT4-1 are either floating or pulled high.

## 4.0 Embedded Controller Modules (Continued)

### Shift Status

The PSTAT register indicates the current status of the shift mechanism. The data transfer process may be in one of the following three states:

- Shifter Empty:
  The shift mechanism is in Receive Inactive, Receive Idle, Transmit Inactive or Transmit Idle state. The PSTAT is cleared because none of the enabled devices has sent a start bit.
- Start Bit Detected:
  The shift mechanism is in Receive Active or Transmit Active state. This indicates that a start bit was identified for at least one of the channels and the shift process has begun. SOT bit in PSTAT register indicates the detection of the start bit and ACH field in PSTAT register indicates the active channel (the channel on which the start bit was detected).
- End of Transaction:
  The shift mechanism is in End-of-Reception or End-of-Transmission state. This indicates that the last bit of the transfer sequence was detected (and the data can therefore be read from PSDAT register) or that the data transmission was completed (for receive and transmit, respectively). EOT bit in PSTAT register indicates transfer completion. If a parity error was detected in the received data, PERR bit in PSTAT register is set. If a stop bit was detected low instead of high, RFERR bit in PSTAT register is set.

### Input Signal Debounce

The PC87591L-N05 performs a debounce operation on the clock input signal before determining its logical value. IDB field in PSCON register determines for how many clock cycles the input signal must be stable to define a change in its value.

### Interrupt Generation

The PSINT1 is an interrupt signal generated by the shift mechanism to allow an interrupt driven interface with the firmware.

The ICU should be programed to detect high-level interrupts on the PSINT1 interrupt. See Section 4.3 on page 96 for details on the ICU. SOTIE and EOTIE bits in PSIEN register mask the interrupt signaling for SOT and EOT bits, respectively, in PSTAT register.

## Receive Mode

### Receive Inactive

When the shift mechanism is enabled and bit XMT=0 in PSCON register, the shift mechanism enters Receive mode in the Receive Inactive state. Receive Idle state is entered when one (or more) of the channels is enabled, by setting the channel enable bit (CLK4-1 for channels 4-1, respectively). In this state, the shift-mechanism sets the clock and data lines of the enabled channels high (1) and waits for a start bit.

### Receive Idle

In the Receive Idle state, the PS/2 interface waits for input from any one of the enabled channels. The first of the enabled channels to send a start bit is selected for handling by the shift mechanism. The other two channels are disabled by forcing ‘0’ on their clock lines.

### Start Bit Detection

The start bit is identified by a falling edge on the clock signal while the data signal is low (0).

If the start bit is identified simultaneously in more than one channel, one channel is selected for receive, while the other channel’s transfer is aborted. The channel with the lower number is selected (i.e., channel 1 has priority over channels 2, 3 and 4, channel 2 has priority over channel 3 and 4 and channel 3 has priority over channel 4). The data transfer in the other channels is aborted before 10 data bits have been sent (by forcing the clock signal to 0), and the transmitting PS/2 device resends its data when its interface is enabled again by the firmware. This mechanism ensures that no incoming data is lost.

When the hardware sets (1) SOT bit and designates the selected channel in ACH field, this indicates receipt of the start bit in PSTAT register. In addition, if SOTIE is set in PSIEN register, an interrupt signal to the ICU is set high. The firmware may use this interrupt to start a time-out timer for the data transfer.

### Receive Active

After identifying the start bit, the shift mechanism enters the “Receive-Active” state. In this state the clock signal of the selected device (PSCLK1, PSCLK2, PSCLK3 or PSCLK4) sets the data bit rate. On each falling edge of the clock, new data is sampled on the data signal of the active channel (i.e., PSDAT1 PSDAT2, PSDAT3 or PSDAT4).

Following the start bit, eight bits of data are received (clocks 2 through 9); a parity bit follows (10th clock) and then a stop bit (11th clock). The stop bit is indicated by a falling edge of the clock with the data signal high (1). If the 11th clock is identified with data low, the receive frame error bit (RFERR in PSTAT register) is set but the clock is treated as the stop bit.

After the parity is received, the shift mechanism checks the incoming data for parity errors. If there are eight data bits with a value of 1 and the parity bit is even, PERR bit in PSCON register is set, indicating a parity error.

## 4.0 Embedded Controller Modules (Continued)

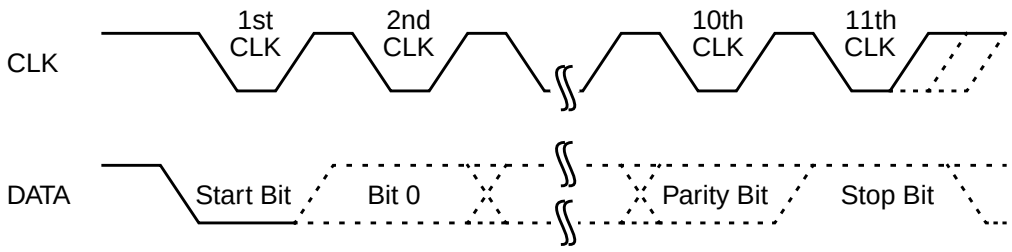



**Figure 41. PS/2 Receive Data Byte Timing**

#### End of Receive

When the stop-bit is detected, the shift mechanism enters the “End-Of-Reception” state. In this state, the shift mechanism:

- Disables all the clock signals by forcing them low
- Sets End-Of-Transaction status bit (EOT = 1 in PSTAT register)
- If EOTIE bit in PSTAT register is set, it asserts (1) the interrupt signal to the ICU.

The shift mechanism stays in this state until it is reset.

Figure 41 shows the receive byte sequence, as defined by the PS/2 standard.

### Transmit Mode

#### Transmit Inactive

When the shift mechanism is enabled and XMT bit in PSCON register is set (1):

- The shift mechanism enters Transmit mode in Transmit Inactive state with all clock signals low and data signals high (PSOSIG = $47_{16}$).
- The firmware writes the data to be transmitted to the PS/2 data register (PSDAT).
- The data line of the channel to be transmitted is forced low by the firmware clearing its data bit (WDAT4-1 for channels 4-1, respectively).

#### Transmit Idle

The Transmit Idle state can be entered by setting the channel enable bit (CLK4-1 for channel 4-1, respectively). This enables the channel to be used for transmission. In this state, the shift-mechanism sets the clock of the enabled channel high (1) while the data line of that channel is held low and waits for a start bit. When a PS/2 device senses the clock signal high with the data signal low, it identifies a transmit request from the PC87591L-N05.

The three channels not in use are disabled by forcing ‘0’ on their clock lines.

#### Start Bit Detection

The start bit is identified by a falling edge on the clock signal while the data signal is low (0).

When a start bit is detected, data transmission begins by outputting bit 0 (LSB) of the transmitted data and setting data bits WDAT4-1 in PSOSIG register. This allows bit 0 of the transmitted data to be output on the PS/2 data signal (PSDAT1, PSDAT2, PSDAT3 or PSDAT4, according to the active channel).

In addition, the hardware sets the SOT bit (to 1) and stores the active channel number in ACH field, indicating transmission of the start bit in PSTAT register. Note that if SOTIE bit in PSIEN register is set, an interrupt signal to the ICU is set high. The firmware can use this interrupt to start a time-out timer for the data transfer.

#### Transmit Active

After identifying the start bit, the shift mechanism enters the Transmit Active state. The clock signal of the selected device (PSCLK1, PSCLK2, PSCLK3 or PSCLK4) sets the data bit rate.

After each of the next seven falling edges of the clock line, one more data bit (bits 1 through 7) is driven on the data line of the active channel (either PSDAT1, PSDAT2, PSDAT3 or PSDAT4).

On the ninth falling edge of the clock, the parity bit is output. The parity bit is high (1) if the number of bits with a value of 1 in the transmitted data was even (i.e., odd parity).

The tenth falling edge causes a 1 to be output as a stop bit. The data signal remains high to allow the PS/2 device to send the line control bit.

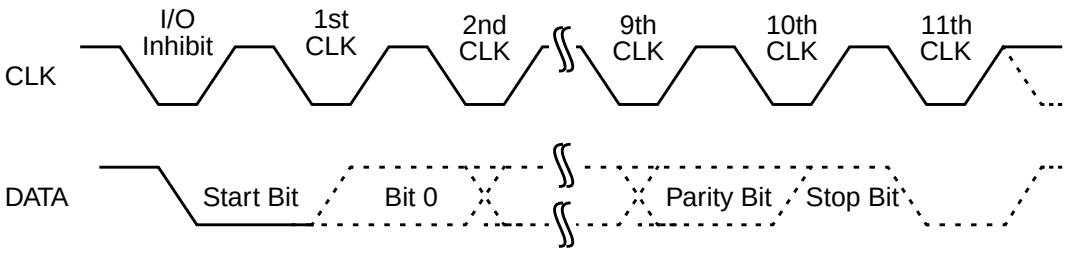



**Figure 42. PS/2 Transmit Data Byte Timing**

The auxiliary device then completes the transfer by sending the line-control bit. The line-control bit, is identified by the data signal being low after the 11th falling edge of the clock.

**End of Transmission**

The End-Of-Transmission state is entered when the line-control bit is detected. In response, the shift mechanism holds all clock signals low, and if the internal pull-up is enabled, all data signals are pulled high by the internal pull-up.

The End-Of-Transaction flag (EOT in PSTAT register) is set to indicate that the transmit operation was completed; in addition, if EOTIE bit in PSIEN register is set, the interrupt signal to the ICU is set high.

The shift mechanism stays at this state until being reset.

Figure 42 shows the transmit byte sequence, as defined by the PS/2 standard.

**Transfer Abort**

At each stage of a receive or transmit operation, the transaction can be aborted by clearing all four channel enable bits (CLK4-1) in PS/2 Output Signal register (PSOSIG) to 0. This resets the shifter state machine and puts it in the Enabled Inactive state. If the shift mechanism is in Transmit Inactive or Transmit Idle state, WDAT4-1 bits should also be set.

### 4.6.5 PS/2 Interface Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### PS/2 Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| PSDAT | PS/2 Data Register | R/W |
| PSTAT | PS/2 Status Register | RO |
| PSCON | PS/2 Control Register | R/W |
| PSOSIG | PS/2 Output Signal Register | R/W |
| PSISIG | PS/2 Input Signal Register | RO |
| PSIEN | PS/2 Interrupt Enable Register | R/W |

# 4.0 Embedded Controller Modules (Continued)

## PS/2 Data Register (PSDAT)

The PSDAT register is a byte-wide read/write register. In Receive mode, PSDAT holds the data received in the last message from the PS/2 device. In Transmit mode, the data to be shifted out is written to this register. When the PS/2 i/f is reset, the contents of this register become invalid.

On reset, the PS/2 interface is set to Receive mode. In this mode, PSDAT should be read only when EOT bit in PSTAT register is set to 1.

Setting the transmit enable bit in PSCON register to 1 (XMT = 1 in PSCON register) puts the PS/2 interface in Transmit mode. PSDAT should be written only when in Transmit mode and when all four channel enable bits CLK4-1 in PSOSIG register are cleared (0).

Location: 00 $FE80_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Data.** Contains the data received in the last message (or that is transmitted in the following transmission). Bit 0 is the first bit to be shifted (LSB). |

## PS/2 Status Register (PSTAT)

The PSTAT register is a byte-wide read-only register. It contains the status information on the data transfer on the PS/2 ports. All non-reserved bits of PSTAT are cleared (0) on reset when CLK1, CLK2 and CLK3 in PSOSIG are cleared and when EN bit in PSCON register is cleared. Reading PSTAT does not clear any of its bits.

Location: 00 $FE82_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **RFERR** | **ACH** | | | **PERR** | **EOT** | **SOT** |
| Reset | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **SOT (Start of Transaction).** When set to 1, indicates that a start bit was detected. The ACH field (bits 5-3 of this register) indicates which of the channels it was detected on. |
| 1 | **EOT (End of Transaction).** When set to 1, Indicates that a PS/2 data transfer was completed, i.e., a stop bit was detected at Receive mode or a line control bit was detected at Transmit mode. |
| 2 | **PERR (Parity Error).**<br>When set to 1, indicates that a parity error was detected in the last data transfer. |
| 5-3 | **ACH (Active Channel).** Defines which of the PS/2 channels is currently active (i.e., a start bit was detected). In case more than one channel become active simultaneously, only the one with the highest priority (lowest number) is flagged.<br>**Bits**<br>**5 4 3 Description**<br>0 0 0: None of the channels is active (default)<br>0 0 1: Channel 1<br>0 1 0: Channel 2<br>1 0 0: Channel 3<br>1 0 1: Channel 4 |
| 6 | **RFERR (Receive Frame Error).**<br>When set to 1, indicates that the stop bit in a received frame was detected low instead of high. |
| 7 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## PS/2 Control Register (PSCON)

The PSCON register is an 8-bit read/write register. It controls the operation of the PS/2 interface by enabling it and controlling the data transfer direction. On reset, PSCON is set to $00_{16}$.

Location: 00 $FE84_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **WPUEN** | **IDB** | | | **HDRV** | | **XMT** | **EN** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **EN (Shift Mechanism Enable).**<br>0: The hardware shift mechanism is disabled and the software controls and monitors the PS/2 signals using PSOSIG and PSISIG registers (default).<br>1: The hardware shift mechanism is enabled. The enabled channels are controlled by PSOSIG, and Transmit/Receive mode is controlled by the XMT bit. |
| 1 | **XMT (Transmit Enable).**<br>0: Receive mode.<br>1: Causes the PS/2 interface to enter Transmit mode. |
| 3-2 | **HDRV (High Drive).** Defines the quasi-bidirectional buffers' behavior on transition from low to high. HDRV defines the period of time for which the output is pulled high with a low-impedance drive (when the PC87591L-N05 changes the output level from low to high). This period is a function of the PC87591L-N05 clock as follows:<br>**Bits 3 2 Description**<br>0 0: Disabled (default)<br>0 1: Low-impedance drive for one clock cycle<br>1 0: Low-impedance drive for two clock cycles<br>1 1: Low-impedance drive for three clock cycles |
| 6-4 | **IDB (Input Debounce).** Defines the number of PC87591L-N05 clock cycles during which the clock input is expected to be stable before the shift mechanism identifies its new value. This protects the shift mechanism from false edge detections. The number of PC87591L-N05 clock cycles for which the input should be stable before an edge is detected is as follows:<br>**Bits 6 5 4 Description**<br>0 0 0: One cycle (default)<br>0 0 1: Two cycles<br>0 1 0: Four cycles<br>0 1 1: Eight cycles<br>1 0 0: 16 cycles<br>1 0 1: 32 cycles |
| 7 | **WPUEN (Weak Pull-Up Enable).**<br>0: The pull-up is disabled. In this state, the system must ensure that PS/2 interface signals are not floating, to enable proper PS/2 operation (default).<br>1: Enables the internal pull-up of the output buffer. The pull-up remains active as long as the buffer does not drive the signal to low level. |

# 4.0 Embedded Controller Modules (Continued)

## PS/2 Output Signal Register (PSOSIG)

The PSOSIG register is a byte-wide, read/write register. It allows setting the value of the PS/2 port signals. When the shift mechanism is enabled, the clock control bits in this register define the active channel(s). On reset, this register is set to $47_{16}$.

Location: 00 $FE86_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **CLK4** | **WDAT4** | **CLK3** | **CLK2** | **CLK1** | **WDAT3** | **WDAT2** | **WDAT1** |
| Reset | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 0 | **WDAT1 (Write Data Signal Channel 1).** Controls the data output to channel 1 data signal (PSDAT1). Use of this bit depends on whether or not the shift mechanism is enabled.<br>• When the shift mechanism is disabled (EN bit in PSCON register is set to 0), the data in WDAT1 is output to PSDAT1 signal.<br>– If WDAT1 is cleared (0), the output buffer data is 0 (i.e., PSDAT1 is forced low).<br>– If WDAT1 is set (1), the output buffer data is 1 (i.e., PSDAT1 is pulled high by the internal pull-up and may be pulled low by an external device).<br>• When the shift mechanism is enabled (EN=1), WDAT1 should be set to 1, except when the shift mechanism is in Transmit mode. In this case, when in transmit-inactive and it is intended to transmit data to channel 1, the firmware should clear WDAT1 bit to force the transmit signaling (low) to the PS/2 device.<br>**Note:** WDAT1 is set by the hardware after the PC87591L-N05 detected a start bit (i.e., on entering Transmit Active state). If a transmission is aborted before Transmit Active state, WDAT1 should be set (1) prior to disabling the channel. |
| 1 | **WDAT2 (Write Data Signal Channel 2).** Controls the data output to channel 2 data signal (PSDAT2). For more information, see the description of bit 1 (above). |
| 2 | **WDAT3 (Write Data Signal Channel 3).** Controls the data output to channel 3 data signal (PSDAT3). For more information, see the description of bit 1 (above). |
| 3 | **CLK1 (Enable Channel 1)**<br>0: Forces the PSCLK1 pin low (0) and disables channel 0 of the shift mechanism.<br>1: Depends on whether or not the shift mechanism is enabled.<br>• When the shift mechanism is enabled (EN bit in PSCON register is set to 1), channel 1 of the PS/2 ports is enabled.<br>• When the shift mechanism is disabled (EN bit in PSCON register is set to 0), the clock line output buffer data is 1 (i.e., the signal is pulled high by the pull-up, if enabled, and may be pulled low by an external device). |
| 4 | **CLK2 (Enable Channel 2).** Same as bit 3 of this register (described above) but for channel 2. |
| 5 | **CLK3 (Enable Channel 3).** Same as bit 3 of this register (described above) but for channel 3. |
| 6 | **WDAT4 (Write Data Signal Channel 4).** Controls the data output to channel 4 data signal (PSDAT4). For more information, see the description of bit 1 (above). |
| 7 | **CLK4 (Enable Channel 4).** Same as bit 3 of this register (described above) but for channel 4. |

**Note:** When CLK1, CLK2, CLK3 and CLK4 are all 0, this is interpreted as a shift mechanism reset. In this case, the PSTAT register and the shift state machine are reset to their initial state.

## PS/2 Input Signal Register (PSISIG)

The PSISIG register is an 8-bit read-only register. It provides the current value of the PS/2 port signals.

Location: 00 $FE88_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **RCLK4** | **RDAT4** | **RCLK3** | **RCLK2** | **RCLK1** | **RDAT3** | **RDAT2** | **RDAT1** |

| Bit | Description |
|---|---|
| 0 | **RDAT1 (Read Data Signal Channel 1).** The current value of the channel 1 data signal (PSDAT1). |
| 1 | **RDAT2 (Read Data Signal Channel 2).** The current value of the channel 2 data signal (PSDAT2). |
| 2 | **RDAT3 (Read Data Signal Channel 3).** The current value of the channel 3 data signal (PSDAT3). |
| 3 | **RCLK1 (Read Clock Signal Channel 1).** When read, returns the current value of the channel 1 clock signal (PSCLK1). |
| 4 | **RCLK2 (Read Clock Signal Channel 2).** When read, returns the current value of the channel 2 clock signal (PSCLK2). |
| 5 | **RCLK3 (Read Clock Signal Channel 3).** When read, returns the current value of the channel 3 clock signal (PSCLK3). |
| 6 | **RDAT4 (Read Data Signal Channel 4).** The current value of the channel 4 data signal (PSDAT4). |
| 7 | **RCLK4 (Read Clock Signal Channel 4).** When read, returns the current value of the channel 4 clock signal (PSCLK4). |

### PS/2 Interrupt Enable Register (PSIEN)

The PSIEN register is an 8-bit read/write register. It enables/disables the various interrupts generated by the PS/2 module. Bits in PSIEN register may be cleared to 0 only when interrupts are disabled (i.e., in the core, I or E bits in PSR register are 0) or when the corresponding interrupts in the ICU are masked. Bits in PSIEN register may be set to 1 at any time. On reset, non-reserved bits of PSIEN are cleared.

Location: 00 $FE8A_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | **DSMIE** | **EOTIE** | **SOTIE** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **SOTIE (Start of Transaction Interrupt Enable).** Used for enabling the interrupt generation on a transaction start detection.<br>0: SOT bit in PSTAT register does not affect the interrupt signal (default).<br>1: The interrupt signal (PSINT1) to the ICU is active (1) whenever SOT bit in PSTAT register is set.<br>**Note:** Once set, SOT is not cleared until the shift mechanism is reset. Therefore SOTIE should be cleared on the first occurrence of an SOT interrupt. SOTIE should be set (1) when the PS/2 module is programed to handle the impending transfer. |
| 1 | **EOTIE (End of Transaction Interrupt Enable).** Used for enabling the interrupt generation on an End of Transaction detection.<br>0: EOT bit in PSTAT register does not affect the interrupt signal (default).<br>1: The interrupt signal (PSINT1) to the ICU is active (1) whenever EOT bit in PSTAT register is set.<br>**Note:** Once set, EOT is not cleared until the shift mechanism is reset. Therefore EOTIE should be cleared on the first occurrence of an EOT interrupt. EOTIE should be set (1) when the PS/2 module is programed to handle the impending transfer. |
| 2 | **Disabled Shift Mechanism Interrupt Enable (DSMIE).** Used for enabling the interrupt generation when the shift mechanism is disabled.<br>0: The four interrupt signals are low. Note that PSINT1 may be activated (1) by other interrupt sources of the module (default).<br>1: The clock input signals are connected to the Interrupt Control Unit (ICU), to allow implementing an interrupt driven PS/2 protocol. The four interrupts generated are PSINT1, PSINT2, PSINT3 and PSINT4, for channels 1, 2, 3 and 4, respectively. Note that PSINT4 is connected to the MIWU and not directly to the ICU.<br>**Note:** When the shift mechanism is disabled, no debounce is applied to the PSCLK inputs before producing the interrupt signals, except for local synchronization. |
| 7-3 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## 4.7 MULTI-FUNCTION 16-BIT TIMER (MFT16)

The PC87591L-N05 includes two Multi-Function Timer (MFT) modules. The registers of each module are prefixed with Tn, and the signals are suffixed with an n (where n is the module's number, i.e., 1 or 2). Each MFT16 module consists of two functional units designed to satisfy a wide range of application requirements. The units include:

- Clock Source Unit that contains a pre-scaler with one clock source selector for each counter.
- Main timer/counter and action unit that contains two counters, two reload registers for PWM, Input Capture or Counter modes.
- Mode selector/control unit that defines the function of the I/O pins and the interrupts.

Figure 43 shows the contents of an MFT16 and the top-level interaction. The rest of the section describes an MFT16 module.

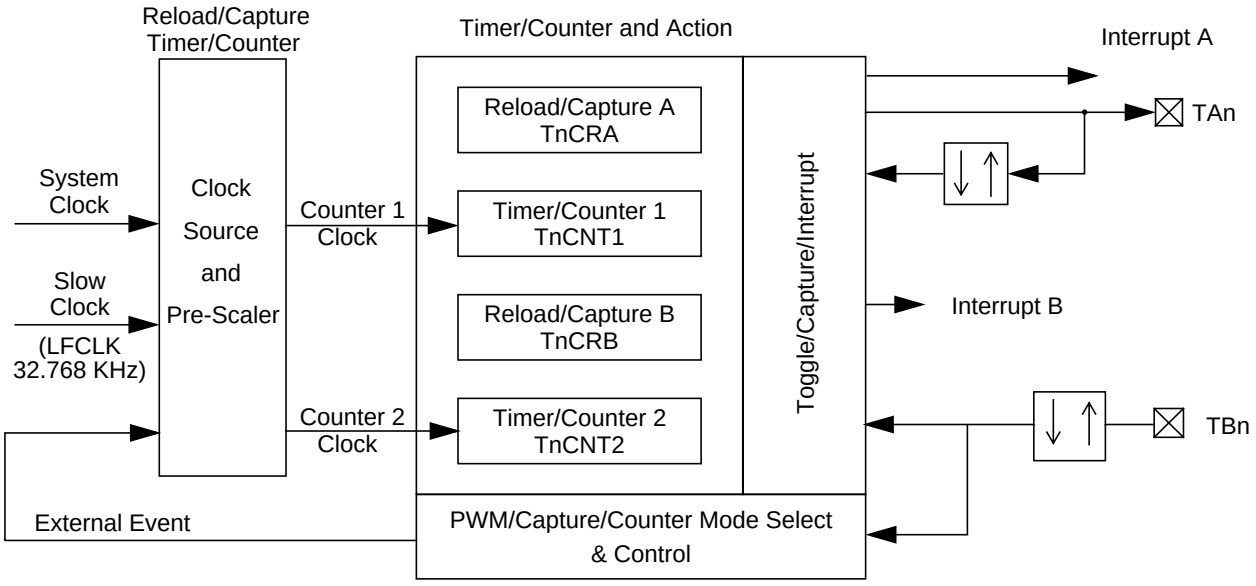



**Figure 43. MFT16 Functional Diagram**

### 4.7.1 Features

- Two 16-bit programmable timers/counters
- Two 16-bit reload/capture registers. These registers are used either as reload registers or capture registers, depending on the mode of operation.
- A 5-bit fully programmable clock pre-scaler
- Clock source selectors for each counter. These enable each counter to operate in:
  - Pulse accumulate mode
  - External event mode
  - Prescaled system clock mode
  - Slow speed clock (LFCLK) input mode
- Two I/O pins (TAn and TBn), with programmable edge detection; these operate as:
  - Capture inputs
  - Capture and preset inputs
  - External event (clock) inputs
  - PWM outputs
- Two interrupts, one for each counter, that can be generated/ triggered by:
  - Timer underflow
  - Timer reload
  - Input capture
- Four pending bits, which can be polled by software, are associated with the two interrupts.

# 4.0 Embedded Controller Modules (Continued)

## 4.7.2 Clock Source Unit

The clock source unit, as shown in Figure 44, contains two clock selectors for each counter and a 5-bit clock pre-scaler.

### Pre-Scaler

The 5-bit clock pre-scaler consists of a pre-scaler register and a 5-bit counter, allowing the timer to run with a prescaled clock. The system clock is divided by the value contained in TnPRSC+1. The minimum counter clock frequency is thus the system clock divided by 32, and the maximum counter clock frequency is equal to the system clock. The pre-scaler register, TnPRSC, can be read or written by software at any time. The pre-scaler counter is a 5-bit down counter which can not be read or written by software. The 5-bit counter and the pre-scaler register TnPRSC are cleared on reset

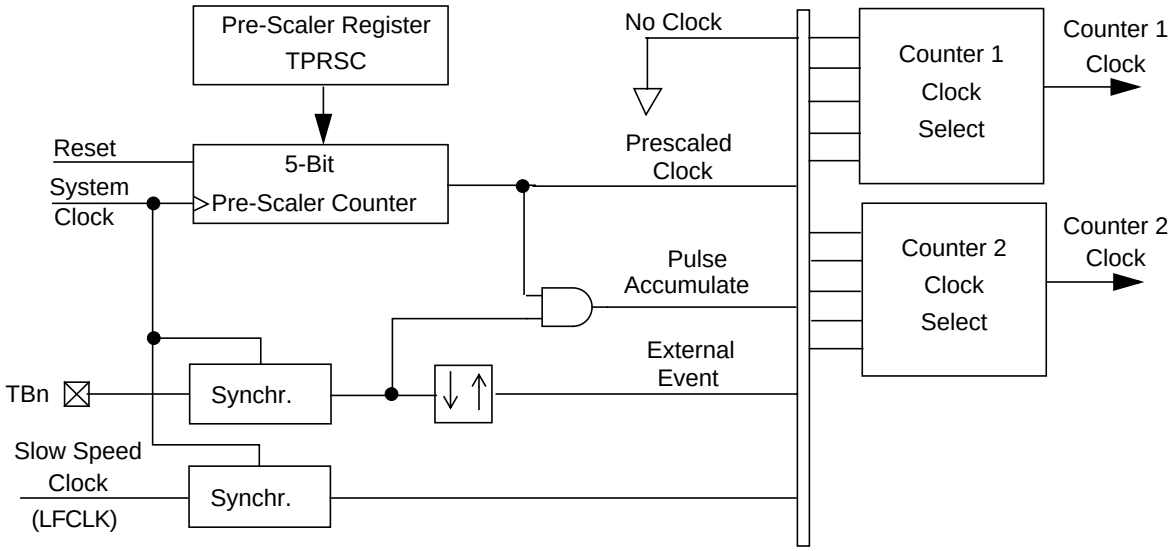



**Figure 44. Clock Pre-Scaler and Selector**

### External Event Clock

The TBn I/O pin can be selected as an external event input clock source for either of the two 16-bit counters. The polarity of the input signal is programmable to trigger a count if either a rising or a falling edge is detected on TBn. The minimum pulse width of the external signal is one system clock cycle; thus the maximum frequency with which the counter can run in this mode is limited to half the system clock frequency.

**Note:** An External Event clock is not available in Dual Channel Capture mode because this mode requires TBn as an input.

### Pulse Accumulate Mode

In Pulse Accumulate mode, the counter can also be clocked while an external signal on TBn is either high or low. In this configuration, the output of the pre-scaler is gated with an external signal applied on TBn input. This mode can be used to obtain a cumulative count of pre-scaler output clock pulses, as shown in Figure 45.

**Note:** Pulse Accumulate mode is not available in Dual Channel Capture mode because this mode requires TBn as an input.

### Slow-Speed Clock (LFCLK)

A slow-speed clock of 32.768 KHz (LFCLK) can be used as a clock source for the two 16-bit counters. The MFT16 synchronizes the slow-speed clock with the system clock.

Some power save modes stop the system clock completely. When this occurs, the timer stops counting the slow-speed clock until the system clock resumes.

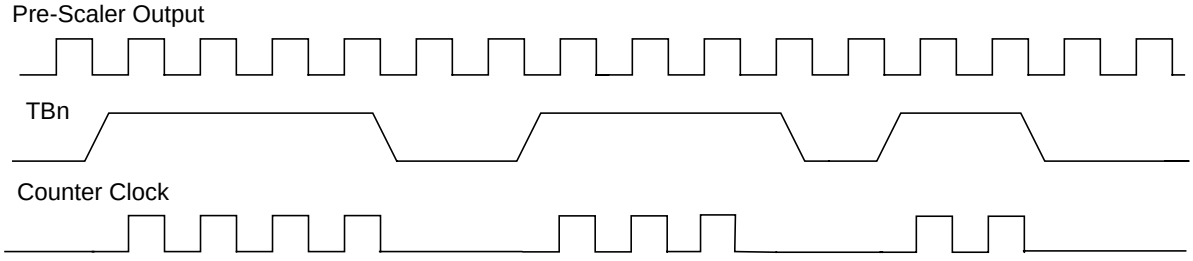



**Figure 45. Pulse Accumulate Mode**

# 4.0 Embedded Controller Modules (Continued)

## Counter Clock Source Select

The clock source unit contains two clock source selectors that allow the clock source to be selected independently for each of the two 16-bit counters from one of the following sources:

- No clock, in which case the counter is stopped
- Prescaled system clock
- External Event count based on TBn
- Pulse Accumulate mode based on TBn
- Slow Speed Clock (LFCLK) i.e., 32.768 KHz

### 4.7.3 Timer/Counter and Action Unit

The timer/counter and action unit consists of two 16-bit counters, TnCNT1 and TnCNT2, in addition to two 16-bit reload/capture registers, TnCRA and TnCRB. The timers are down counters capable of triggering events on underflow detection (count roll-over from $0000_{16}$ to $FFFF_{16}$). In addition, it contains the mode control logic which allows the timer to operate in any of four operation modes described below.

Different interrupts can be triggered on certain conditions, and the functionality of the I/O pins changes depending on the mode of operation. Therefore the interrupt control and the I/O control are an integral part of the timer/counter unit.

## Operation Modes

The MFT16 can be configured to operate in any one of four modes, as summarized in Table 17 and described in this section.

**Table 17. Operation Modes**

| Mode | Description | Timer/Counter 1 (TnCNT1) | Reload/Capture A (TnCRA) | Reload/Capture B (TnCRB) | Timer/Counter 2 (TnCNT2) |
|---|---|---|---|---|---|
| 1 | PWM and system timer or external event counter | Counter for PWM | Auto Reload A = PWM time 1 | Auto reload = PWM time 2 | System Timer or external event counter |
| 2 | Dual input capture and system timer | Capture A and B time base | Capture counter 1 value on TAn event | Capture counter 1 value on TBn event | System Timer |
| 3 | Dual independent timer | Time base for first timer | Reload register for timer/counter 1 | Reload register for timer/counter 2 | Time base for second timer |
| 4 | Input capture and timer | Time base for first timer | Reload register for timer/counter 1 | Capture counter 1 value on TBn event | Capture B time base |

## Mode 1, PWM and Counter

PWM can be used to generate precise pulses of known width and duty cycle on the TAn pin. The timer is clocked by the selected clock. An underflow causes the timer register to be reloaded alternately from the TnCRA and TnCRB registers and optionally causes TAn output to toggle. Thus, the values stored in TnCRA and TnCRB registers control the high and low time of the signal produced on TAn. In PWM mode, timer/counter 2 can be used either as a simple system timer or as an external event counter. The counter can be loaded by software with a specific value; the counter can then generate an interrupt after the pre-programed number of external events have been received on TBn input.

Figure 46 shows a block diagram of the timer operating in mode 1. In PWM mode, counter 1, TnCNT1, functions as the time base for the PWM timer. Counter 1 counts down at the clock rate selected via the counter 1 clock selector. When an underflow occurs, the timer register is reloaded alternately from the TnCRA and TnCRB registers, and counting proceeds downward from the loaded value. On reset, and every time this mode is entered, the first reload in this mode is from the TnCRA register. Once enabled, the counter starts counting down from the value currently in TnCNT1. At the first underflow, the timer is loaded from TnCRA; on the second underflow, it is loaded from TnCRB; on the third underflow, it is loaded from TnCRA, and so on. Note that every time the counter is stopped through the selection of “No-Clock” in the counter 1 clock selector (TnCKC), it obtains its first reload value after it has been re-started from TnCRA register.

The timer can be configured to toggle TAn output bit on underflow. This results in the generation of a clock signal on TAn, with the width and duty cycle controlled by the values stored in TnCRA and TnCRB registers. This PWM clock is processor-independent because once the timer is set up, no more interaction is required by software (and therefore the CPU) to generate a continuous PWM signal. Software can select the initial value of the PWM output signal as either high or low. See “Timer I/O Functions” on page 137 for additional details. The timer can be configured to generate separate interrupts on reload from TnCRA and TnCRB. The interrupts can be enabled or disabled under software control. The TAnPND or TnBPND flags, respectively, which are set by the hardware on occurrence of a timer reload, indicate which interrupt occurred. See Section 4.7.4 on page 136 for detailed information.

## 4.0 Embedded Controller Modules (Continued)

In this mode of operation, timer/counter 2 can be used as a simple system timer, an external event counter or a pulse accumulate counter. Counter TnCNT2 counts down with the clock selected via the counter 2 clock selector, and TnCNT2 can be configured to generate an interrupt on underflow if the interrupt is enabled by TnDIEN bit. See Section 4.7.4 on page 136 for detailed information.

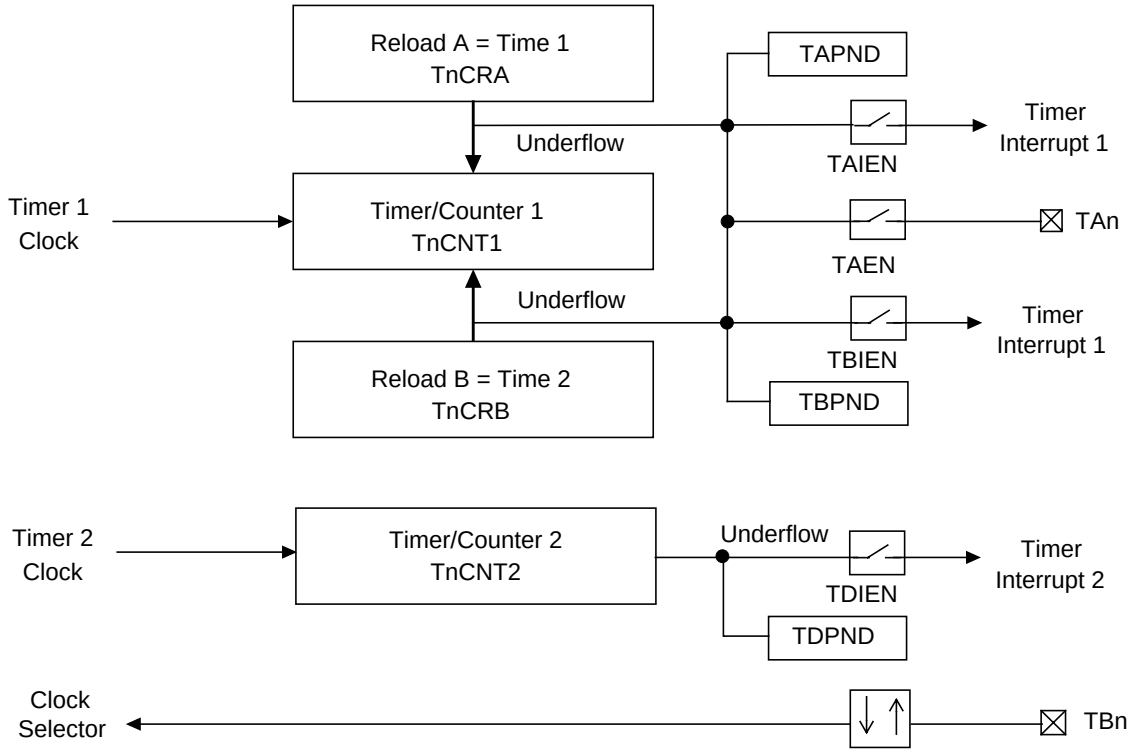



**Figure 46. Mode 1, PWM and Counter**

### Mode 2, Dual Input Capture

Dual Input Capture mode can be used to precisely measure the frequency of an external clock that is slower than the selected clock source frequency or to measure the elapsed time between external events. A transition received on the TAn or TBn pin causes a transfer of timer/counter 1 contents to TnCRA or TnCRB register, respectively. In this mode, timer/counter 2 can be utilized as a system timer that is pre-loaded by software and generates an interrupt on underflow.

Figure 47 shows a block diagram of the timer operating in mode 2. In this mode of operation, the timebase of the capture timer is formed by counter 1, which counts down with the clock selected via the counter 1 clock selector. In Dual Input Capture mode, TAn and TBn pins function as capture inputs. A transition received on TAn pin causes a transfer of the timer contents to TnCRA register. Similarly, a transition received on the TBn pin causes a transfer of the timer contents to TnCRB register. TAn and TBn inputs can be configured to perform a counter preset to $FFFF_{16}$ on reception of a valid capture event. In this case, the current value of the counter is transferred to the corresponding capture register; following this, the counter is preset to $FFFF_{16}$. Using this approach enables an external signal's on-time, off-time or period to be directly determined, while reducing CPU overhead.

The pulse width of the input signal on TAn and TBn must be equal to or greater than one system clock cycle (see Section 7.6.10 on page 358 for additional details). The values captured in TnCRA register at different times reflect the elapsed time between transitions on the TAn pin. The same is true for TnCRB register and the TBn pin. Each input pin can be configured to sense either rising or falling edge transitions.

The timer can be configured to generate interrupts on reception of a transition on either TAn or TBn. The interrupts can be enabled or disabled separately for TAn or TBn by TAnIEN and TnBIEN bits. An underflow of TnCNT1 can also generate an interrupt if the interrupt was enabled by TnCIEN bit. All three interrupts have individual pending flags. See Section 4.7.4 on page 136 for detailed information.

Timer/counter 2 can be used as a "simple" system timer in this mode of operation. The TnCNT2 counter counts down with the clock selected via the counter 2 clock selector, and TnCNT2 can be configured to generate an interrupt on underflow if the interrupt was enabled by TnDIEN bit. See Section 4.7.4 on page 136 for detailed information.

Note that TnCNT1 cannot operate in the "Pulse Accumulate" or "External Event Counter" modes of operation since TBn input is used as a capture input. Selecting either "Pulse Accumulate" mode or "External Event Counter" mode for TnCNT1 causes TnCNT1 to stop. However, all available clock source modes may be selected for TnCNT2. Thus it is possible to use TnCNT2 to determine the number of capture events on TBn or the elapsed time between capture events on TBn.

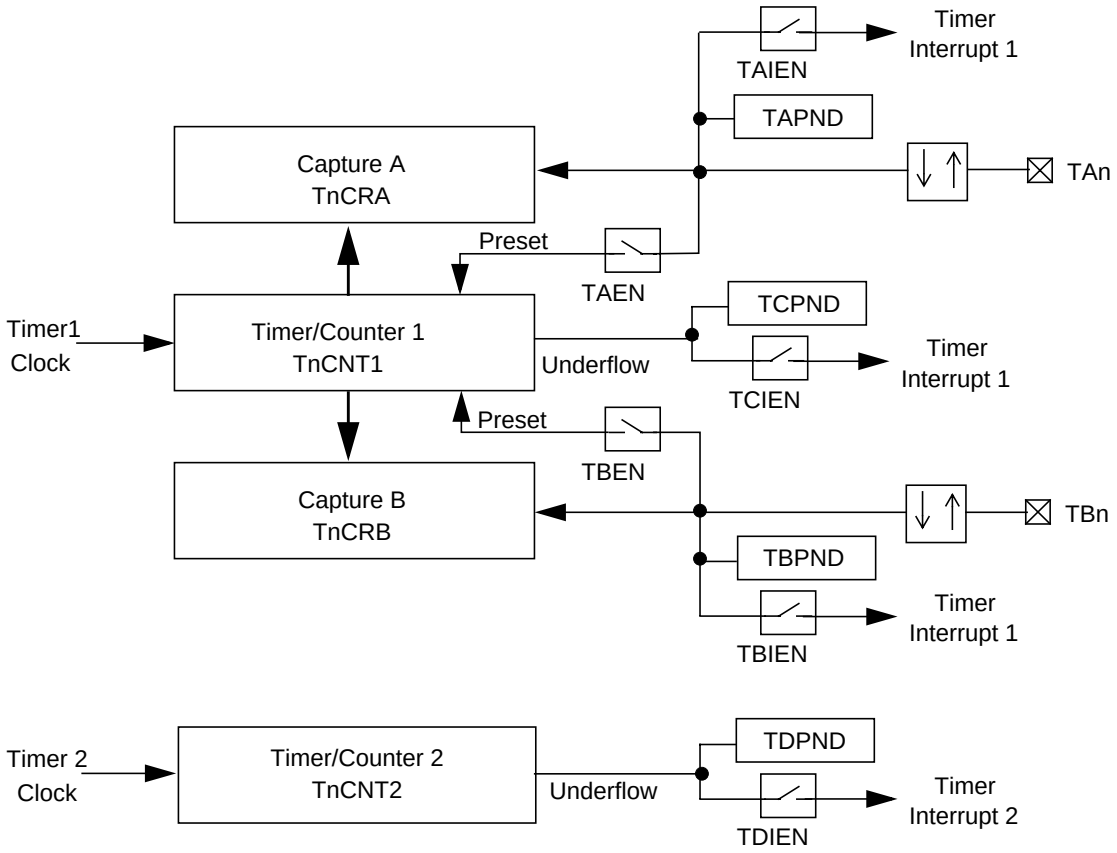



**Figure 47. Mode 2, Dual Input Capture**

### Mode 3, Dual Independent Timer

Dual Independent Timer mode can be used for a wide variety of system tasks such as the generation of periodic system interrupts, based either on the prescaled clock or external events on TBn. The timer can also toggle TAn pin on underflow, allowing the simple generation of a processor-independent 50% duty cycle signal on TAn. In this mode, TnCNT1 counts down and reloads from TnCRA on underflow while TnCNT2 is reloaded from TnCRB on underflow.

In this mode, the timer is configured to operate as a dual independent system timer or dual external event counter. In addition, timer/counter 1 can generate a 50% duty cycle signal on the TAn pin. The TBn pin can be used as an external event input or pulse accumulate input and forms the clock source to either counter 1 or counter 2, as described above. Both counters can also be operated using the prescaled system clock. Figure 48 shows a block diagram of the timer in mode 3.

Timer/counter 1 (TnCNT1) counts down at the rate of the selected clock (see “Counter Clock Source Select” on page 132 for additional details). On underflow, TnCNT1 is reloaded from TnCRA register and counting proceeds. If enabled, the TAn pin toggles on underflow of TnCNT1. Software can select the initial value of the TAn output signal as either high or low (see “Timer I/O Functions” on page 137 for additional details). In addition, the TnAPND interrupt pending flag is set, and a timer interrupt 1 is generated if TnAIEN bit is set to 1 (see Section 4.7.4 on page 136 for detailed information). Since TAn toggles on every underflow, a 50% duty cycle PWM signal can be generated on TAn without requiring interaction by the core.

Timer/counter 2 (TnCNT2) counts down at the rate of the selected clock (see “Counter Clock Source Select” on page 132 additional details). On every underflow of TnCNT2, the value contained in TnCRB register is loaded into TnCNT2, and counting proceeds downwards from that value. In addition, the TnDPND interrupt pending flag is set, and a timer interrupt 2 is generated if TnDIEN bit is set to 1. See Section 4.7.4 on page 136 for detailed information.

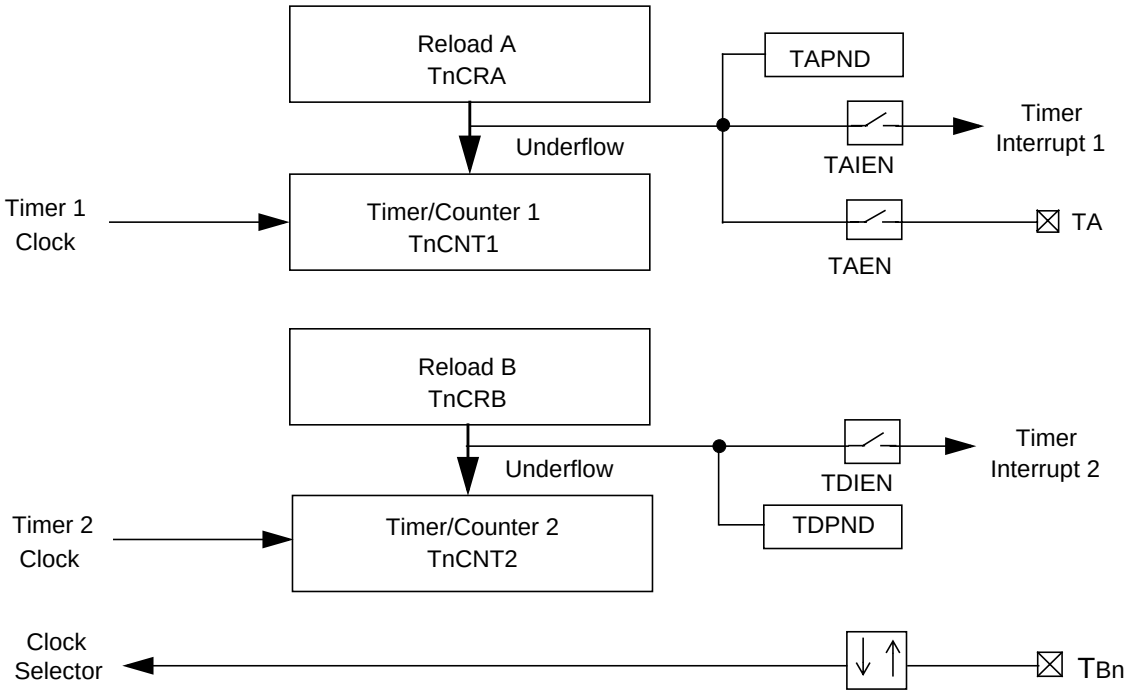



**Figure 48. Mode 3, Dual Independent Timer**

### Mode 4, Input Capture and Timer

It is also possible to operate in a mode that offers a combination of a single timer, with automatic reload, and a single capture timer. In this mode, TnCNT1 operates as a timer that is reloaded from TnCRA on underflow while TnCNT2 forms the time base of the capture timer. The value on TnCNT2 is transferred to TnCRB when a valid event on TBn is detected. It is possible to toggle TA on every underflow of TnCNT1 and thus generate a 50% duty cycle signal on TAn.

This mode is a combination of modes 3 and 2. It allows timer/counter 2 to operate as a single-input capture timer concurrently with timer/counter 1. (Timer/counter 2 can be used as a system timer, as described in mode 3.) Figure 49 shows a block diagram of the timer in mode 4.

TnCNT1 starts counting down once a clock has been enabled. On underflow, TnCNT1 is reloaded from TnCRA register, and counting proceeds downwards from that value. If enabled, the TAn pin toggles on every underflow of TnCNT1. Software can select the initial value of the TAn output signal as either high or low (see Section 4.7.5 on page 137 for additional details). In addition, the TnAPND interrupt pending flag is set, and a timer interrupt 1 is generated if TnAIEN bit is set to 1 (see Section 4.7.4 on page 136 for detailed information). Since TAn toggles on every underflow, a 50% duty cycle signal can be generated on TAn without requiring any interaction of software (and therefore the core).

TnCNT2 starts counting down once a clock has been enabled. When a transition is received on TBn, the value contained in TnCNT2 is transferred to TnCRB, and the interrupt pending flag, TnBPND, is set. A timer interrupt 2 is generated if it is enabled. A preset of the counter to $FFFF_{16}$ on detection of a transition on TBn can be enabled. In this case, the current value of TnCNT2 is transferred to TnCRB, followed by a preset of the counter to $FFFF_{16}$. TnCNT2 starts counting downwards from $FFFF_{16}$ until the next transition is received on TBn, which causes the procedure of capture and preset to be repeated. Underflow of TnCNT2 sets the TnDPND interrupt pending flag and can also generate a timer interrupt 2 if the interrupt was enabled (see Section 4.7.4 on page 136 for detailed information.). The input signal on TBn must have a pulse width equal to or greater than one system clock cycle (see Section 7.6.10 on page 358 for additional details). TBn can be configured to sense either rising or negative edge transitions.

Note that TnCNT2 can not operate in the Pulse Accumulate or External Event Counter modes since TBn input is used as a capture input. Selecting either Pulse Accumulate mode or External Event Counter mode for TnCNT2 causes TnCNT2 to stop.

Note, however, that all available clock source modes may be selected for TnCNT1. Thus it is possible to use TnCNT1 to determine the number of capture events on TBn or the elapsed time between capture events on TBn.

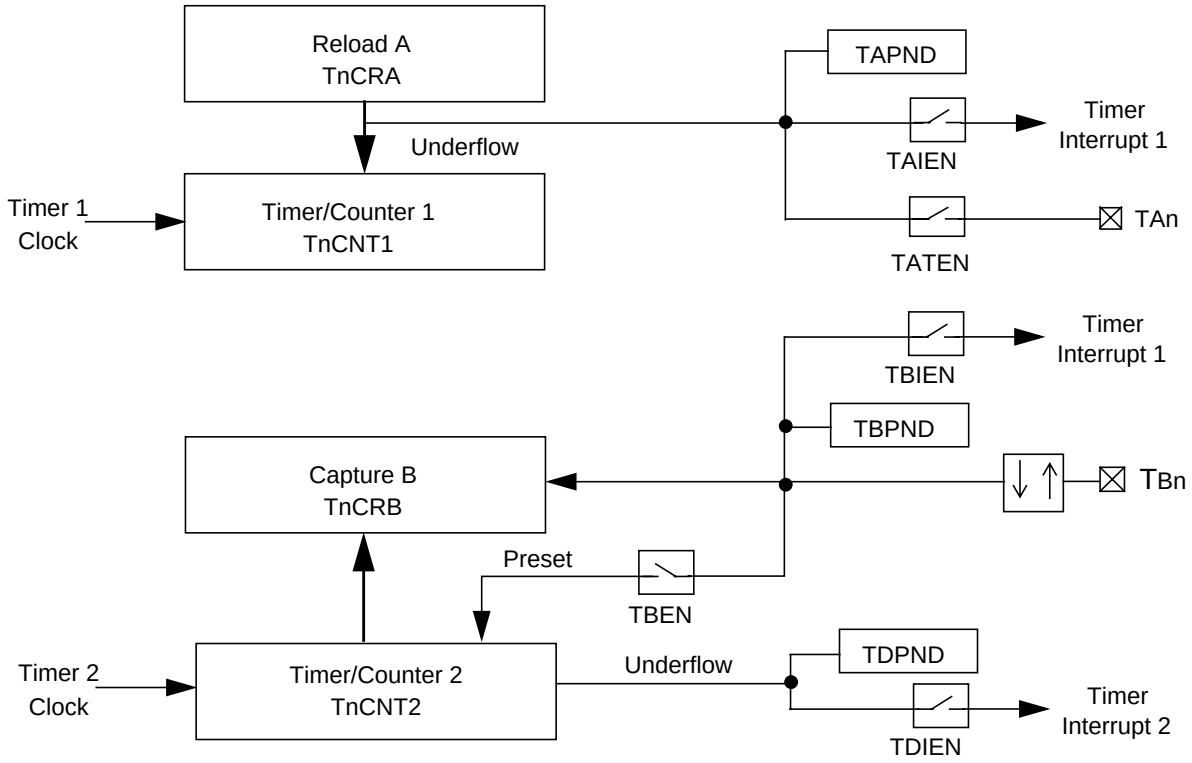



**Figure 49. Mode 4, Input Capture and Timer**

### 4.7.4 Timer Interrupts

The MFT16 contains a total of four interrupt sources that are mapped to two different system interrupts. All sources have a pending flag associated with them and can be enabled or disabled under software control. The pending flags are TnXPND, where n is the module and X is a letter from A to D. An interrupt enable flag, TnXIEN, is associated with each interrupt pending flag. Interrupt sources A, B or C can generate a timer interrupt 1; interrupt source D can generate a timer interrupt 2. Note that not all interrupt sources are available in all modes. Table 18 shows which events can trigger an interrupt in each mode of operation:

**Table 18. MFT16 Interrupts**

| System Interrupt | Interrupt Pending Flag | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
|---|---|---|---|---|---|
| | | PWM and Counter | Dual Input Capture | Dual Independent Timer | Input Capture and Timer |
| Timer Int. 1 | TnAPND | TnCNT1 reload from TnCRA | Input capture on TAn transition | TnCNT1 reload from TnCRA | TnCNT1 reload from TnCRA |
| | TnBPND | TnCNT1 reload from TnCRB | Input Capture on TBn transition | N/A | Input Capture on TBn transition |
| | TnCPND | N/A | TnCNT1 underflow | N/A | N/A |
| Timer Int. 2 | TnDPND | TnCNT2 underflow | TnCNT2 underflow | TnCNT2 reload from TnCRB | TnCNT2 underflow |

# 4.0 Embedded Controller Modules (Continued)

## 4.7.5 Timer I/O Functions

There are two I/O pins associated with each of the MFT16 modules: TAn and TBn, where n denotes the module on a given device. The functionality of TA and TB depends on the mode of operation and the value of TAEN and TBEN bits. Table 19 shows the function of TA and TB versus the selected mode of operation. Note that if TA functions as a PWM output, the initial and present value of TA is defined by TAOUT. For example, to start with TA high, TAOUT must be set (1) prior to enabling the timer clock.

**Table 19. MFT16 I/O Functions**

| I/O | TnAEN TnBEN | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
|---|---|---|---|---|---|
| | | PWM and Counter | Dual Input Capture | Dual Independent Timer | Input Capture and Timer |
| TAn | TnAEN=0 TnBEN=X | No Output | Capture TnCNT1 into TnCRA | No Output toggle | No Output toggle |
| | TnAEN=1 TnBEN=X | Toggle Output on underflow of TnCNT1 | Capture TnCNT1 into TnCRA and preset TnCNT1 | Toggle Output on underflow of TnCNT1 | Toggle Output on underflow of TnCNT1 |
| TBn | TnAEN=X TnBEN=0 | Ext. Event or Pulse Accumulate Input | Capture TnCNT1 into TnCRB | Ext. Event or Pulse Accumulate Input | Capture TnCNT2 into TnCRB |
| | TnAEN=X TnBEN=1 | Ext. Event or Pulse Accumulate Input | Capture TnCNT1 into TnCRB and preset TnCNT1 | Ext. Event or Pulse Accumulate Input | Capture TnCNT2 into TnCRB and preset TnCNT2 |

## 4.7.6 Operation in Development System

The MFT16 supports freezing the counters during breakpoints while operating in development systems. If FREEZE bit is asserted, all timer counter clocks are inhibited, and the current values of timer/counter registers TnCNT1 and TnCNT2 are frozen. Once FREEZE becomes inactive, counting resumes from the previous value. See 4.20.5 on page 237 for more details.

## 4.7.7 MFT16 Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### MFT16 Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| TnPRSC | Clock Pre-Scaler Register | R/W |
| TnCKC | Clock Unit Control Register | R/W |
| TnCNT1 | Timer/Counter Register 1 | R/W |
| TnCNT2 | Timer/Counter Register 2 | R/W |
| TnCRA | Reload/Capture Register A | R/W |
| TnCRB | Reload/Capture Register B | R/W |
| TnCTRL | Timer Mode Control Register | R/W |
| TnICTL | Timer Interrupt Control Register | R/W |
| TnICLR | Timer Interrupt Clear Register | WO |

# 4.0 Embedded Controller Modules (Continued)

## Timer/Counter Register 1 (TnCNT1)

The TnCNT1 register is a word-wide read/write register that is not altered by reset. The value on power-on is unknown.

Location: MFT16 1: 00 $FD80_{16}$
MFT16 2: 00 $FDA0_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **TCNT1** | | | | | | | | | | | | | | | |

## Reload/Capture Register A (TnCRA)

The TnCRA register is a word-wide read/write register that is not affected by reset and thus contains random data on power-up.

Location: MFT16 1: 00 $FD82_{16}$
MFT16 2: 00 $FDA2_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **TCRA** | | | | | | | | | | | | | | | |

## Reload/Capture Register B (TnCRB)

The TnCRB register is a word-wide read/write register that is not affected by reset and thus contains random data on power-up.

Location: MFT16 1: 00 $FD84_{16}$
MFT16 2: 00 $FDA4_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **TCRB** | | | | | | | | | | | | | | | |

## Timer/Counter Register 2 (TnCNT2)

The TnCNT2 register is a word-wide read/write register that is not altered by reset. The power-up value is unknown.

Location: MFT16 1: 00 $FD86_{16}$
MFT16 2: 00 $FDA6_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **TCNT2** | | | | | | | | | | | | | | | |

# 4.0 Embedded Controller Modules (Continued)

## Clock Pre-Scaler Register (TnPRSC)

The TnPRSC register is a byte-wide read/write register. It contains the current value of the clock pre-scaler, CLKPS. The register is cleared on reset. It defines the timer clock pre-scaler ratio.

Location: MFT16 1: 00 $FD88_{16}$
MFT16 2: 00 $FDA8_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | **CLKPS** | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 4-0 | **CLKPS (Clock Pre-Scaler).** The timer clock is generated by dividing the system clock by CLKPS+1. Therefore the maximum timer clock frequency is equal to the system clock (CLKPS=00000) and the minimum timer clock is the system clock divided by 32 (CLKPS=11111). |
| 7-5 | **Reserved.** |

## Clock Unit Control Register (TnCKC)

The TnCKC register is a byte-wide read/write register. It defines the clock source selection for each timer counter. The register is cleared on reset, thus disabling timer/counter 1 and timer/counter 2 clocks.

Location: MFT16 1: 00 $FD8A_{16}$
MFT16 2: 00 $FDAA_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **C2CSEL** | | | **C1CSEL** | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 2-0 | **C1CSEL (Counter 1 Clock Select).** Defines the clock mode for timer/counter 1 where:<br>**Bits 2 1 0 Description**<br>0 0 0: No Clock (Counter 1 stopped) (default)<br>0 0 1: Prescaled system clock<br>0 1 0: External Event on TBn<br>0 1 1: Pulse Accumulate<br>1 0 0: Slow-speed Clock<br>Other: Reserved |
| 5-3 | **C2CSEL (Counter 2 Clock Select).** Defines the clock mode for timer/counter 2.<br>**Bits 5 4 3 Description**<br>0 0 0: No Clock (Counter 2 stopped) (default)<br>0 0 1: Prescaled system clock<br>0 1 0: External Event on TBn<br>0 1 1: Pulse Accumulate<br>1 0 0: Slow-speed Clock<br>Other: Reserved |
| 7-6 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## Timer Mode Control Register (TnCTRL)

The TnCTRL register is a byte-wide read/write register. It defines the mode of operation of timer/counter and TAn and TBn I/O pins. The register is cleared on reset.

Location: MFT16 1: 00 $FD8C_{16}$

MFT16 2: 00 $FDAC_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved (must be 1)** | **TAOUT** | **TBEN** | **TAEN** | **TBEDG** | **TAEDG** | **MDSEL** | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 1-0 | **MDSEL (Mode Select).** Defines the MFT16 mode of operation.<br>**Bits 1 0 Description**<br>0 0: Mode 1 (default)<br>0 1: Mode 2<br>1 0: Mode 3<br>1 1: Mode 4 |
| 2 | **TAEDG (TAn Edge Polarity).**<br>0: A high-to-low transition on TAn causes the action defined by the mode of operation, e.g., input capture (default)<br>1: A low-to-high transition on TAn results in the defined action. |
| 3 | **TBEDG (TBn Edge Polarity).**<br>0: A high-to-low transition on TBn causes the action defined by the mode of operation, e.g., input capture or external event count (default)<br>1: A low-to-high transition on TBn results in the defined action<br>In Pulse Accumulate mode, when this bit is set to 1, the count is enabled if TBn is high. When cleared (0) and while operating in Pulse Accumulate mode, the counter is enabled if TBn is low. |
| 4 | **TAEN (TAn Enable).** Enables TAn to function either as a preset input or as a PWM output, depending on the mode of operation.<br>If this bit is set (1), while operating in Dual Input Capture mode (mode 2), a transition on TAn causes TnCNT1 to be preset to $FFFF_{16}$. In the remaining modes of operation, setting TnAEN enables TAn to function as a PWM output. See Table 19 on page 137 for additional information. |
| 5 | **TBEN (TBn Enable).** When set (1), and while operating in either Dual Input Capture mode (mode 2) or Input Capture and Timer mode (mode 4), a transition on TBn causes the corresponding timer/counter to be preset to $FFFF_{16}$. In mode 2, TnCNT1 is preset to $FFFF_{16}$; in mode 4, TnCNT2 is preset to $FFFF_{16}$. The bit has no effect while operating in any mode other than modes 2 or 4. See Table 19 on page 137 for additional information. |
| 6 | **TAOUT (TAn Output Data).** Contains the value of TAn output when TAn is used as a PWM output.<br>0: TAn is low (default)<br>1: TAn is high<br>This bit is set and cleared by hardware and thus reflects the status of TAn. This bit can be read at any time. It may be used to set the initial value of TAn output in PWM mode. Note that if the hardware attempts to toggle this bit at the same time as software is writing to the bit, the software write takes precedence over the hardware update. This bit has no effect when TAn is used as input. |
| 7 | **Reserved (must be set to 1).** |

# 4.0 Embedded Controller Modules (Continued)

## Timer Interrupt Control Register (TnICTL)

The TnICTL register is a byte-wide read/write register. It contains the interrupt enable bit and associated interrupt pending bits for the four timer interrupt sources. The TnICTL register format is shown below. The register is cleared on reset.

Location: MFT16 1: 00 $FD8E_{16}$
MFT16 2: 00 $FDAE_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **TDIEN** | **TCIEN** | **TBIEN** | **TAIEN** | **TDPND** | **TCPND** | **TBPND** | **TAPND** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **TAPND (Timer Interrupt Source A Pending).** When asserted, indicates that an interrupt condition (as shown in Table 18 on page 136) has occurred. This bit can be set by either hardware or software.<br>This bit can not be cleared (set to 0) directly. TAPND can be cleared via the Timer Interrupt Clear register. A write of 0 to TAPND is ignored. The bit is cleared on reset. |
| 1 | **Timer Interrupt Source B Pending (TBPND).** Same as TAPND but for a different condition, as shown in Table 18. |
| 2 | **Timer Interrupt Source C Pending (TCPND).** Same as TAPND but for a different condition, as shown in Table 18. |
| 3 | **Timer Interrupt Source D Pending (TDPND).** Same as TAPND but for a different condition, as shown in Table 18. |
| 4 | **TAIEN (Timer Interrupt A Enable)**.<br>0: No system interrupt occurs, but the associated pending flag TAPND is set<br>1: Enables a system interrupt based on the occurrence of a condition, as listed in Table 18<br>**Note:** The bit can be set or cleared by software at any time. |
| 5 | **TBIEN (Timer Interrupt B Enable).**<br>0: No system interrupt occurs, but the associated pending flag TBPND is set (default)<br>1: Enables a system interrupt based on the occurrence of a condition, as listed in Table 18<br>**Note:** The bit can be set or cleared by software at any time. |
| 6 | **TCIEN (Timer Interrupt C Enable).**<br>0: No system interrupt occurs, but the associated pending flag TCPND is set (default)<br>1: Enables a system interrupt based on the occurrence of a condition, as listed in Table 18<br>**Note:** The bit can be set or cleared by software at any time. |
| 7 | **TDIEN (Timer Interrupt D Enable).**<br>0: No system interrupt occurs, but the associated pending flag TDPND is set (default)<br>1: Enables a system interrupt based on the occurrence of a condition, as listed in Table 18<br>**Note:** The bit can be set or cleared by software at any time. |

# 4.0 Embedded Controller Modules (Continued)

## Timer Interrupt Clear Register (TnICLR)

The TnICLR register is a byte-wide write-only register. It controls the clear of pending flags TAPND, TBPND, TCPND and TDPND, which are located in TnICTRL register.

Location: MFT16 1: 00 $FD90_{16}$
MFT16 2: 00 $FDB0_{16}$

Type: WO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | **TDCLR** | **TCCLR** | **TBCLR** | **TACLR** |

| Bit | Description |
|---|---|
| 0 | **TACLR (Timer Pending A Clear).** Writing a 1 to this bit clears the TAPND flag in TnICTL register.<br>0: Has no effect on TAPND. The previous value of TAPND is maintained<br>1: Causes the TAPND flag to be cleared (0) |
| 1 | **TBCLR (Timer Pending B Clear).** Writing a 1 to this bit clears the TBPND flag in TnICTL register.<br>0: Has no effect on TBPND. The previous value of TBPND is maintained<br>1: Causes the TBPND flag to be cleared (0) |
| 2 | **TCCLR (Timer Pending C Clear).** Writing a 1 to this bit clears the TCPND flag in TnICTL register.<br>0: Has no effect on TCPND. The previous value of TCPND is maintained<br>1: Causes the TCPND flag to be cleared (0) |
| 3 | **TDCLR (Timer Pending D Clear).** Writing a 1 to this bit clears the TDPND flag in TnICTL register.<br>0: Has no effect on TDPND. The previous value of TDPND is maintained<br>1: Causes the TDPND flag to be cleared (0) |
| 7-4 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## 4.8 PULSE WIDTH MODULATOR (PWM)

The PWM module generates eight 8/16-bit PWM outputs; each may have a different duty cycle. A common 8/16-bit clock pre-scaler and an 8/16-bit down-counter determine the cycle time, the minimal possible pulse width and the duty cycle steps.

The PWM is configurable to provide 8-bit or 16-bit resolution PWM outputs, as defined by PWMRES bit of PWM Control Register (PWMCNT).

### 4.8.1 Features

- Eight PWM outputs
- Common 8/16-bit fully programmable pre-scaler
- Common 8/16-bit fully programmable down-counter
- 8/16-bit duty cycle control per output
- Programmable polarity per output
- Low Power mode

### 4.8.2 Functional Description

The PWM generates eight 8/16-bit PWM outputs, PWM0 to PWM7. The Duty Cycle registers 0 to 7 (DCRi, i = 0 to 7) control the duty cycle of PWM0 to PWM7 output signals, respectively. The Cycle Time register (CTR) controls the cycle time and duty cycle steps of all outputs.

The PWM use the core domain clock as a reference for its activities. The pre-scaler divider divides the PWM input clock by a pre-programmable ratio of 1:1 through 1:65536, as defined by the Clock Pre-Scaler register (PRSC).

The PWM cycle is defined in terms of cycles of the pre-scaler output clock. It has a period of (CTR +1) cycles, out of which the signal will be high for DCRi cycles. INVPi, when set, may be used to inverse this behavior.

The cycle period is defined by the 8/16-bit down-counter. It counts using the output clock from the pre-scaler divider, starting from the value held in CTR register down to 0. When it reaches 0, the down-counter restarts from the CTR value.

The DCRi register defines the number of clock cycles during which PWMi signal is high during the complete down-counter cycle.

If Inverse PWMi bit (INVPi) in PWM Polarity register (PWMPOL) is active (1), the DCRi register defines the number of clock cycles during which PWMi signal is low in each of the down-counter complete cycles.

Figure 50 is a functional block diagram of the PWM module.

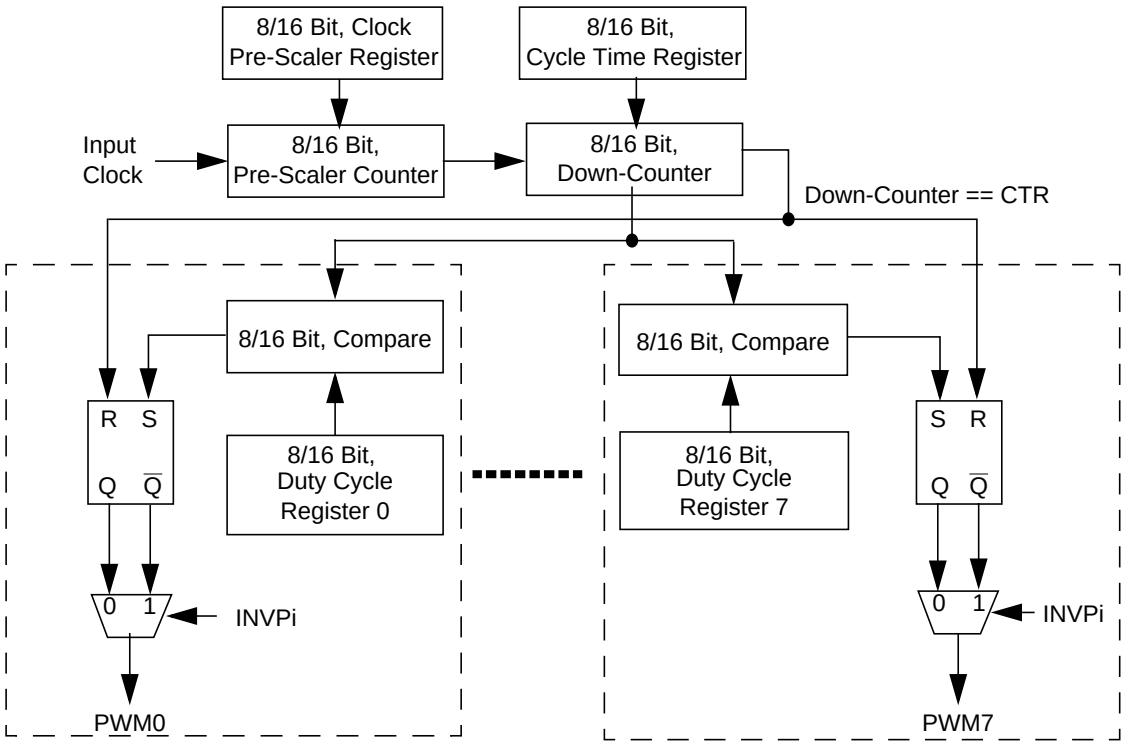



**Figure 50. PWM Block Diagram**

# 4.0 Embedded Controller Modules (Continued)

## 4.8.3 Cycle Time and Duty Cycle Calculation

The PWM module supports duty cycles in the range of 0% to 100%.

The PWMi output signal cycle time is: (PRSC + 1) x (CTR + 1) x $T_{CLK}$

where:

- $T_{CLK}$ is the core domain clock cycle time (i.e., the PWM input clock).
- The cycle time may range from 2 x $T_{CLK}$ to 65536 x $T_{CLK}$.

The PWMi output signal duty cycle (in %, when INVPi is 0) is: (DCRi + 1) / (CTR + 1) x 100.

Special cases:

- If the DCRi value is greater than the CTR value, PWMi signal is always low.
- If DCRi value is equal to the CTR value, PWMi signal is always high.

When Inverse PWMi bit is 1, the value of PWMi output is inverted. i.e., in the period described as 1 is 0 and vice versa.

## 4.8.4 Power Modes

The PWM is in Low Power mode when Power Mode bit (PWR) in PWM Control Register (PWMCNT) is 0. In this mode, the PWM input clock is disabled (stopped), but the registers are accessible and maintained. The PWMi signal is 0 when INVPi bit is 0; it is 1 when INVPi bit is 1.

The PWM is in normal power mode when PWR bit in PWMCNT register is 1. In this mode, the PWM module is enabled, its registers are accessible and its clock is functional.

The PRSC and CTR registers should be updated during Low Power mode. Otherwise, there may be unpredictable transient behavior.

## 4.8.5 PWM Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### PWM Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| PRSC | Clock Pre-Scaler | R/W |
| CTR | Cycle Time | R/W |
| DCRi | Duty Cycle 0 to 7 | R/W |
| PWMPOL | PWM Polarity | R/W |
| PWMCNT | PWM Control | R/W |

### Clock Pre-Scaler Register (PRSC)

The PRSC register controls the cycle time and the minimal pulse width. PRSC is cleared ($0000_{16}$) on reset.

Location: 00 $FD00_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **PRSC15-0** | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 15-0 | **Pre-Scaler Divider Value.** The divider of the Input Clock is the number defined either by PRSC15-0 + 1 when PWMRES bit (in PWMCNT register) is set to 1, or by PRSC7-0 + 1 when PWMRES bit is set to 0. For example, a value of $0000_{16}$ results in a divide by 1, a value of $FFFF_{16}$ results in a divide by 65536.<br>When PWMRES bit is set to 0, only the low byte (PRSC7-0) of this register must be written (PRSC15-8 are ignored).<br>The contents of this register may be changed only when the PWM module is in Low Power mode. Otherwise, there may be unpredictable results. |

# 4.0 Embedded Controller Modules (Continued)

## Cycle Time Register (CTR)

The CTR register controls the cycle time and duty cycle steps, CTR is set ($FFFF_{16}$) on reset.

Location: 00 $FD02_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **CTR15-0** | | | | | | | | | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 15-0 | **Cycle Time Value.** The 8/16-bit down-counter divides the pre-scaler output clock either by CTR15-0 + 1 when PWMRES bit (in PWMCNT register) is set to 1, or by CTR7-0 + 1 when PWMRES bit is set to 0. For example, a value of $0000_{16}$ results in a divide by 1, a value of $FFFF_{16}$ results in a divide by 65536.<br>When PWMRES bit is set to 0, only the low byte (CTR7-0) of this register must be written (CTR15-8 are ignored).<br>The contents of this register may be changed only when the PWM module is in Low Power mode. Otherwise, there may be unpredictable results. |

## Duty Cycle Registers 0 to 7 (DCRi)

The DCRi (i = 0 to 7) registers control the duty cycle of PWMi output signal. DCRi is cleared ($0000_{16}$) on reset.

Location: Channel 0 - 00 $FD08_{16}$
Channel 1 - 00 $FD0A_{16}$
Channel 2 - 00 $FD0C_{16}$
Channel 3 - 00 $FD0E_{16}$
Channel 4 - 00 $FD10_{16}$
Channel 5 - 00 $FD12_{16}$
Channel 6 - 00 $FD14_{16}$
Channel 7 - 00 $FD16_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **DCRi15-0** | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 15-0 | **Duty Cycle Value.** DCRi register defines the number of clocks for which PWMi is high (from the full cycle of the PWMi cycle), when Inverse PWMi bit in PWM Polarity register is 0.<br>If the DCRi value > CTR value, PWMi signal is always low.<br>If DCRi value == CTR value, PWMi signal is always high.<br>When Inverse PWMi bit is 1, the value of PWMi is inverse.<br>When PWMRES bit is set to 0, only the low byte (DCRi7-0) of this register must be written (DCRi15-8 are ignored). |

# 4.0 Embedded Controller Modules (Continued)

## PWM Polarity Register (PWMPOL)

This register controls the polarity of PWM0 to PWM7. The register is cleared ($00_{16}$) on reset.

Location: 00 $FD04_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **INVP7-0** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Inverse PWM Outputs.** Each bit controls the corresponding polarity of PWM0 to PWM7; e.g., INVP0 controls PWM0.<br>0: The number in DCRi indicates for how many clocks (of down-counter decrements) the PWMi signal is high (default)<br>1: The number in DCRi indicates for how many clocks (of down-counter decrements) the PWMi signal is low |

## PWM Control Register (PWMCNT)

This register controls PWM operation. The register is cleared ($00_{16}$) on reset.

Location: 00 $FD06_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | **PWMRES** | **Reserved** | | | **PWR** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **PWR (Power Mode).** This bit controls the operating mode of the PWM, as follows:<br>0: Low Power mode. PWM input clock is disabled (stopped). Its registers are accessible and maintained. PWMi signal is 0 when INVPi bit is 0. The PWMi signal is 1 when INVPi bit is 1 (default)<br>1: Normal Power mode. PWM module is enabled. Its registers are accessible and its clock is functional. |
| 3-1 | **Reserved.** |
| 4 | **PWMRES (PWM Resolution).** This bit selects PWM 8-bit or 16-bit resolution.<br>0: 8-bit resolution. The clock pre-scaler divider is defined by bits 7-0 of PRSC register. The down-counter restarts from bits 7-0 of CTR register. The duty cycle is defined by bits 7-0 of the DCRi registers (default).<br>1: 16-bit resolution. The clock pre-scaler divider is defined by bits 15-0 of PRSC register. The down-counter restarts from bits 15-0 of CTR register. The duty cycle is defined by bits 15-0 of the DCRi registers. |
| 7-5 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## 4.9 UNIVERSAL SYNCHRONOUS/ASYNCHRONOUS RECEIVER-TRANSMITTER (USART)

The PC87591L-N05 includes two USART interface modules. The registers of each module contain ‘n’, and the signals are suffixed with ‘n’, where ‘n’ is module number 0 or 1.

The USART is a full-duplex synchronous/asynchronous receiver-transmitter that supports a wide range of software programmable baud rates and data formats. It handles automatic parity generation and several error detection schemes. USART1 supports DMA transfers to enable fast processor-independent receive and transmit.

### 4.9.1 Features

- Full duplex double-buffered receiver-transmitter
- Synchronous operation
- Asynchronous operation
- Programmable baud rate between CLK/2 and CLK/32768 baud
- Numerous framing formats
  - — seven, eight or nine data bits
  - — one or two stop bits
  - — odd, even, mark, space or no parity
- Hardware support of parity-bit generation during transmission and parity-bit check during reception
- Interrupt on transmit buffer empty, receive buffer full and receive error conditions with separate enable
- Software-controlled break transmission and detection
- Internal diagnostic capability
- Automatic error detection
  - — Parity Error
  - — Framing Error
  - — Data Overrun Error
- 9-bit Attention mode
- DMA support for transmit and receive with separate enable

### 4.9.2 Functional Overview

The USART is composed of the following functional units:

- Transmitter
- Receiver
- Baud rate generator
- Control and error detection

## 4.0 Embedded Controller Modules (Continued)

Figure 51 shows the USART block diagram.

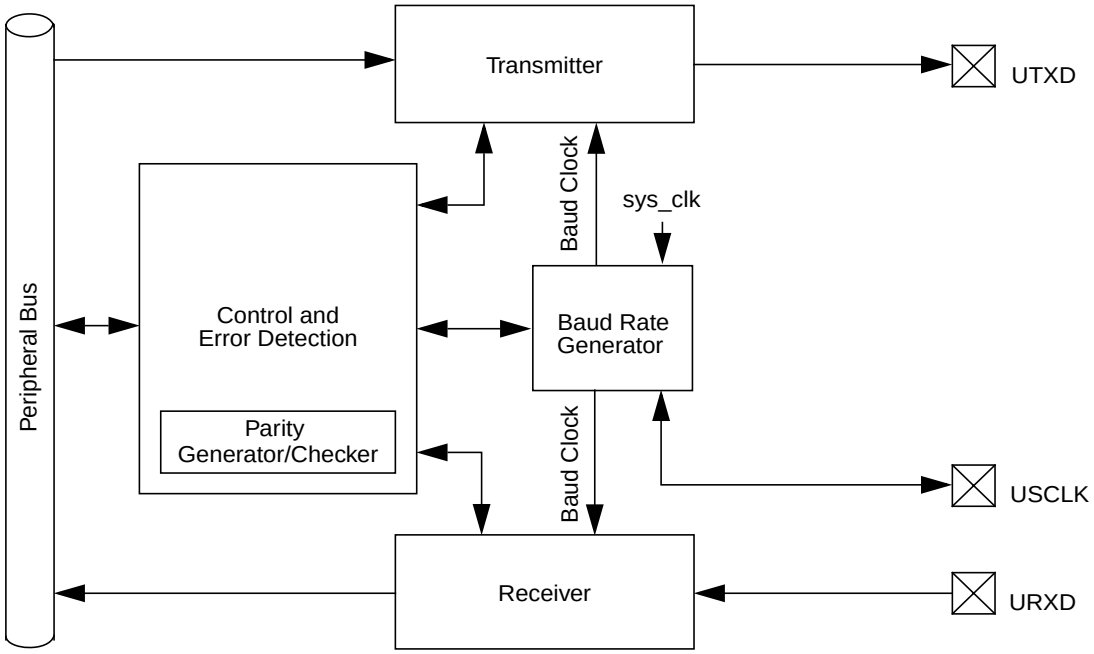



**Figure 51. USART Block Diagram**

Each functional unit is described briefly in this section.

**Transmitter**

The Transmitter consists of an 8-bit transmit shift register and an 8-bit transmit buffer. Data is loaded in parallel from the buffer into the shift register and then transmitted serially on the UTXDn pin.

**Receiver**

The Receiver consists of an 8-bit receive shift register and an 8-bit receive buffer. Data is received serially into the shift register from the URXDn pin. After eight bits have been received, the contents of the shift register are transferred in parallel to the receive buffer.

**Baud Rate Generator**

The Baud Rate Generator generates the clock for the Asynchronous and Synchronous modes of operation. It consists of two registers and a two-stage counter. The registers are used to specify a pre-scaler value and baud rate divisor. The first stage of the counter divides the CLK clock in 0.5 increments based on the value of the pre-scaler. The second stage of the counter divides the output of the first stage in integer increments based on the value of the baud rate divisor.

**Control and Error Detection**

This unit contains the control registers (see Section 4.9.4 on page 154), control logic, error detection circuitry and parity generator/checker. It supports:

- Selection of the data format, mode of operation, clock source and parity type
- Generation and detection of parity
- Reporting of parity errors
- Detection and reporting of data overrun and frame errors
- Interrupts on transmit buffer empty, receive buffer full, receive error and delta clear to send conditions
- Generation and detection of line breaks

### 4.9.3 Operation

The USART has two basic modes of operation; Synchronous and Asynchronous. In addition, two special Synchronous and Asynchronous modes, attention and diagnostic, are available. This section describes the operating modes of the USART.

## 4.0 Embedded Controller Modules (Continued)

### Asynchronous Mode

USART Asynchronous mode enables the device to communicate with other devices using two communication signals: transmit (UTXDn) and receive (URXDn).

In Asynchronous mode, Transmit Shift register (TSFT) and Transmit Buffer register (UnTBUF) double buffer data for transmission. To transmit a character, a data byte is loaded into UnTBUF register. The data is then transferred to TSFT register. While TSFT register is shifting out the current character (LSB first) on the UTXDn pin, UnTBUF register is loaded by software with the next byte to be transmitted. When TSFT completes transmitting the last stop bit of the current frame, the contents of UnTBUF are transferred to TSFT register and the Transmit Buffer Empty flag (TBE) is set. The TBE flag is automatically reset by the USART when the software loads a new character into UnTBUF register. During transmission, the XMIP bit is set high by the USART. This bit is reset only after the USART has sent the last stop bit of the current character and UnTBUF register is empty. UnTBUF register is a read/write register. TSFT register is not user accessible.

In Asynchronous mode, the input frequency to the USART is 16 times the baud rate, i.e., there are 16 clock cycles per bit time. In Asynchronous mode, the baud rate generator is always used as the UART clock.

Receive Shift register (RSFT) and Receive Buffer register (UnRBUF) double buffer the data being received. The USART receiver continually monitors the signal on the URXDn pin for a low level to detect the beginning of a start bit. On sensing this low level, the USART waits for seven input clock cycles and samples again three times. If all three samples still indicate a valid low, the receiver considers this to be a valid start bit, and the remaining bits in the character frame are each sampled three times around the mid-bit position. For any bit following the start bit, the logic value is found by majority voting, i.e., the two samples with the same value define the value of the data bit. Figure 52 shows the process of start-bit detection and bit sampling. Serial data input on the URXDn pin is shifted into RSFT register. On receiving the complete character, the contents of RSFT register are copied into UnRBUF register and Receive Buffer Full flag (RBF) is set. The RBF flag is automatically reset when software reads the character from UnRBUF register. The RSFT register is not user accessible.

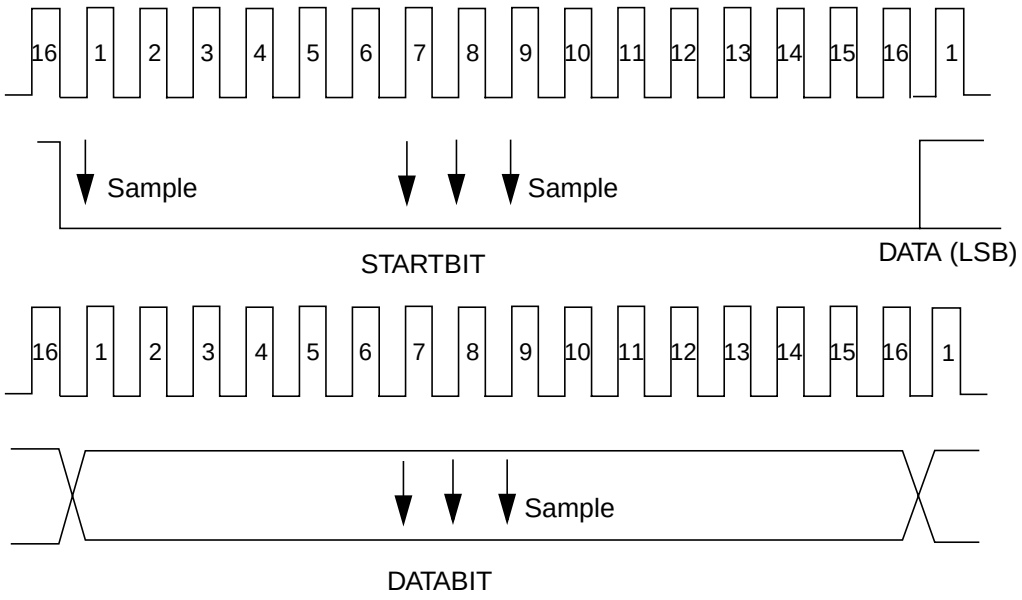



**Figure 52. USART Asynchronous Communication**

### Synchronous Mode

The USART Synchronous mode enables the device to communicate with other devices using three communication signals: transmit (UTXDn), receive (URXDn) and clock (USCLKn). In this mode, data bits are transferred synchronously using the USART clock signal. As shown in Figure 53, the data is transmitted on the rising edge and received on the falling edge of the clock. Data is transmitted and received with the LSB first.

## 4.0 Embedded Controller Modules (Continued)

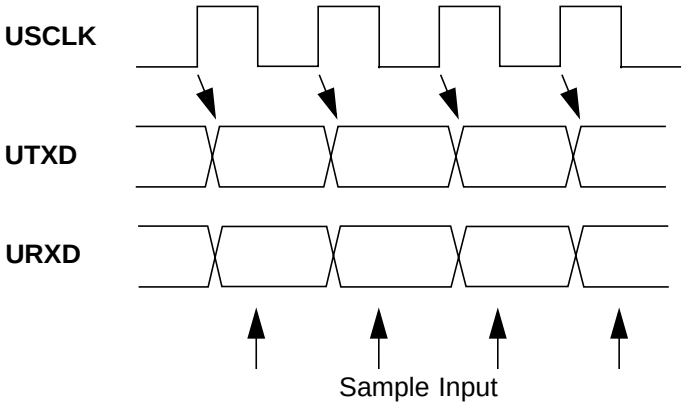



**Figure 53. USART Synchronous Communication**

In Synchronous mode, Transmit Shift register (TSFT) and Transmit Buffer register (UnTBUF) double buffer data for transmission. To transmit a character, a data byte is loaded into TBUF register. The data is then transferred to TSFT register. The TSFT shifts out one bit of the current character, LSB first, on each rising edge of the clock. While the TSFT is shifting out the current character on the UTXDn pin, UnTBUF register can be loaded by the software with the next byte to be transmitted. When TSFT completes transmitting the last stop bit within the current frame, the contents of UnTBUF are transferred to TSFT register, and the Transmit Buffer Empty flag (TBE) is set. The TBE flag is automatically reset by the USART when the software loads a new character into UnTBUF register. During transmission, XMIP bit is set high by the USART. This bit is reset only after the USART has sent the last frame bit of the current character and UnTBUF register is empty.

Receive Shift register (RSFT) and Receive Buffer register (UnRBUF) double buffer the data being received. Serial data input on the URXDn pin is shifted into RSFT register at the first falling edge of the clock. Each subsequent falling edge of the clock causes an additional bit to be shifted into RSFT register. The USART assumes a complete character has been received after the correct number of rising edges on USCLKn (based on the selected frame format) has been detected. On receiving a complete character, the contents of RSFT register are copied into UnRBUF register, and Receive Buffer Full flag (RBF) is set. The RBF flag is automatically reset when software reads the character from UnRBUF register.

The transmitter and receiver may be clocked from either an external source available on the USCLKn pin or the internal baud rate generator. If the internal baud rate generator is used, the baud clock is output on the USCLKn pin.

### Attention Mode

Attention mode is available for networking this device with other processors. This mode requires the 9-bit data format with no parity. The number of start/stop bits are user selectable. In this mode, two types of 9-bit characters are sent on the network: address characters consisting of eight address bits and a '1' in the ninth bit position and data characters consisting of eight data bits and a '0' in the ninth bit position.

While in Attention mode, the USART receiver monitors the communication flow but ignores all characters until an address character is received. On the receipt of an address character, the contents of Receive Shift register are copied to the receive buffer. The RBF flag is set and an interrupt (if enabled) is generated. The ATN bit is automatically reset to zero, and the USART begins receiving all subsequent characters. The software must examine the contents of UnRBUF register and respond by accepting the subsequent characters (by leaving the ATN bit reset) or waiting for the next address character (by setting the ATN bit again).

The operation of the USART transmitter is not affected by the selection of this mode. The value of the ninth bit to be transmitted is programed by setting the STPXB9 bit appropriately. The value of the ninth bit received is read from the RB9 bit.

### Diagnostic Mode

This mode is available for diagnostic tests of the USART. In this mode, the UTXDn and URXDn pins are internally connected, and data that is shifted out of Transmit Shift register is immediately transferred to Receive Shift register. This mode supports only the 9-bit data format with no parity. The number of start and stop bits is user selectable.

### Frame Format Selection

The format shown in Figure 54 consists of a start bit, seven data bits (excluding parity) and one or two stop bits. If parity bit generation is enabled by setting the PEN bit, a parity bit is generated and transmitted following the seven data bits.

## 4.0 Embedded Controller Modules (Continued)

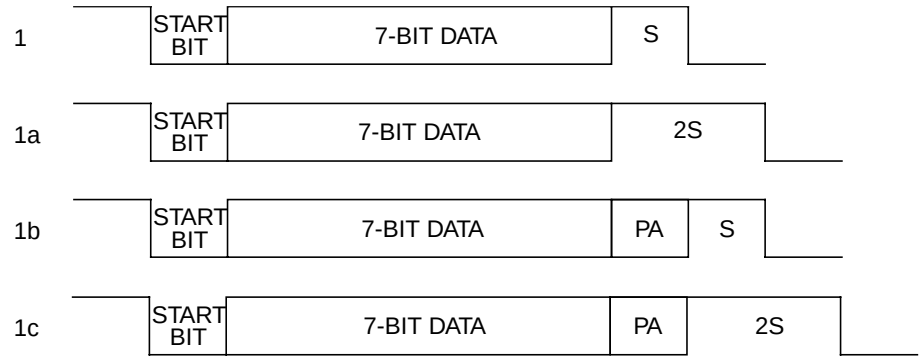



**Figure 54. Seven Data Bit Frame Options**

The format shown in Figure 55 consists of one start bit, eight data bits (excluding parity) and one or two stop bits. If parity bit generation is enabled by setting the PEN bit, a parity bit is generated and transmitted following the eight data bits.

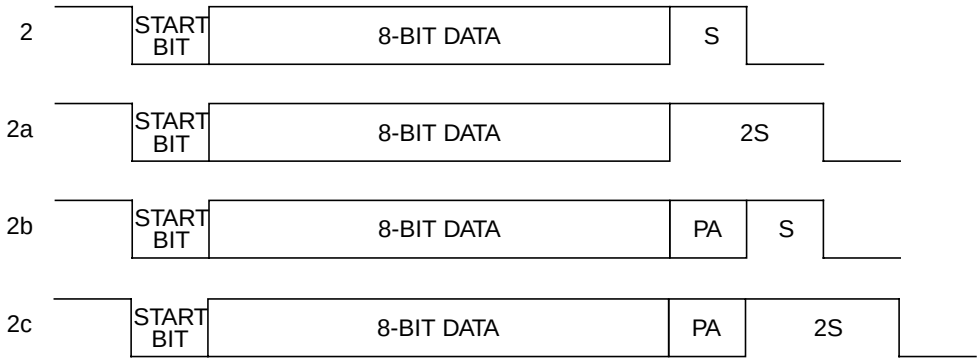



**Figure 55. Eight Data Bit Frame Options**

The format shown in Figure 56 consists of one start bit, nine data bits and one or two stop bits. This format also supports the USART attention feature. When operating in this format, all eight bits of UnTBUF and UnRBUF are used for data. The ninth data bit is transmitted and received using two bits in the control registers, called STPXB9 and RB9. Parity is not generated or verified in this mode.

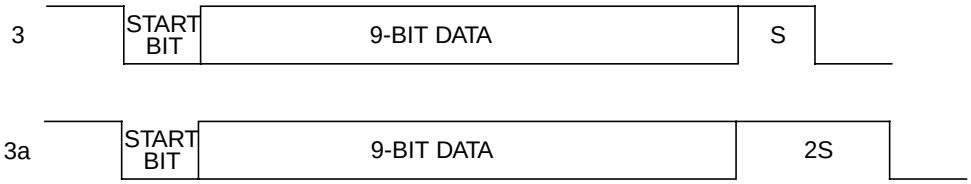



**Figure 56. Nine Data Bit Frame Options**

### Baud Rate Generator

The Baud Rate Generator provides the basic baud clock from the system clock. The system clock is passed through a two-stage divider chain consisting of a 5-bit baud rate pre-scaler (PSC) and an 11-bit baud rate divisor (DIV).

The correspondences between the 5-bit pre-scaler select (PSC) and pre-scaler factors are shown in Table 20.

A pre-scaler factor of zero corresponds to NO CLOCK. The NO CLOCK condition is the USART Power-Down mode. In this mode, the USART clock is turned off in order to reduce power consumption. The user should set the pre-scaler factor to zero (NO CLOCK) before selecting a new baud rate. Altering the baud rate while the USART is in operation could cause incorrect data to be received or transmitted. UnPSR register must contain a value other than zero when an external clock is used at USCLKn.

In Asynchronous mode, the baud rate is calculated by:

$$BR = SYS\_CLK/(16 \times N \times P)$$

## 4.0 Embedded Controller Modules (Continued)

**Table 20. Pre-Scaler Factors**

| Pre-Scaler Select | Pre-Scaler Factor | Pre-Scaler Select | Pre-Scaler Factor |
|---|---|---|---|
| 00000 | NO CLOCK | 10000 | 8.5 |
| 00001 | 1 | 10001 | 9 |
| 00010 | 1.5 | 10010 | 9.5 |
| 00011 | 2 | 10011 | 10 |
| 00100 | 2.5 | 10100 | 10.5 |
| 00101 | 3 | 10101 | 11 |
| 00110 | 3.5 | 10110 | 11.5 |
| 00111 | 4 | 10111 | 12 |
| 01000 | 4.5 | 11000 | 12.5 |
| 01001 | 5 | 11001 | 13 |
| 01010 | 5.5 | 11010 | 13.5 |
| 01011 | 6 | 11011 | 14 |
| 01100 | 6.5 | 11100 | 14.5 |
| 01101 | 7 | 11101 | 15 |
| 01110 | 7.5 | 11110 | 15.5 |
| 01111 | 8 | 11111 | 16 |

where:

- BR is the baud rate
- SYS_CLK is the system clock
- N is the value of the baud rate divisor + 1
- P is the pre-scaler divide factor selected by the value in the PSR register

The divide by 16 operation is performed because in Asynchronous mode, the input frequency to the USART is 16 times the baud rate. In Synchronous mode, the input clock to the USART is equal to the baud rate.

**Interrupts**

The USART is capable of generating interrupts on one of the following conditions:

- Receive Buffer Full
- Receive Error
- Transmit Buffer Empty

## 4.0 Embedded Controller Modules (Continued)

Figure 57 shows a diagram of the interrupt sources and associated enable bits.

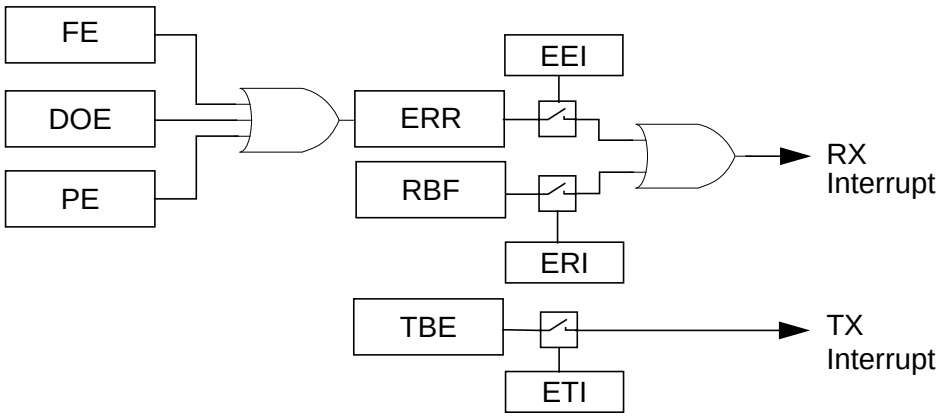



**Figure 57. USART Interrupt Sources**

The interrupts can be individually enabled or disabled using Enable Transmit Interrupt (ETI), Enable Receive Interrupt (ERI) and Enable Receive Error Interrupt (EEI) bits in UnICTRL register.

A transmit interrupt is generated when both the TBE and ETI bits are set. To remove this interrupt, the software must either disable the interrupt by clearing the ETI bit or write to UnTBUF register (thus clearing the TBE bit).

A receive interrupt is generated on two conditions:

- If both the RBF and ERI bits are set. To remove this interrupt, the software must either disable the interrupt, by clearing the ERI bit, or read from UnRBUF register (thus clearing the RBF bit).
- If both the ERR and the EEI bits are set. To remove this interrupt, the software must either disable it by clearing the EEI bit, or read UnSTAT register, which causes ERR flag to be cleared.

### DMA Support

The USART can operate with either one or two DMA channels. Two DMA channels are required for processor-independent full-duplex operation. Both receive and transmit DMA can be enabled individually.

If the transmit DMA is enabled (ETD=1), the USART issues a DMA request every time the TBE flag is set. Enabling the transmit DMA automatically disables the TX interrupt independent of the value of the ETI bit.

Enabling the receive DMA (ERD=1) causes a DMA request to be asserted every time the Receive Buffer Full flag (RBF) is set. Once the receive DMA is enabled the RX interrupt is automatically disabled independent of the value of the ERI bit. However, to detect errors during reception the receive error interrupt should be enabled (EEI=1) while using the DMA.

In the PC87591L-N05 only USART1 supports DMA.

### Break Generation and Detection

A line break is generated when BRK bit is set in MDSL register. The UTXDn line remains low until the user resets the BRK bit.

A line break is detected if URXDn remains low for a time equivalent to 10 bit times or longer, after a missing stop bit has been detected.

### Parity Generation and Detection

Parity is only generated or checked with 7- and 8-bit data formats. It is not generated or checked in Diagnostic Loopback mode, Attention mode or in Normal mode with 9-bit data format. Parity generation and checking is enabled and disabled via PEN bit in UnFRS register. PSEL bits in UnFRS register are used to select odd, even, mark or space parity.

### ISE Mode Operation

The USART module supports breakpoint operation by preserving some of the status bits of the UnSTAT and UnICTRL registers. While the FREEZE bit is asserted, the PE, FE, DOE, BKD and DCTS bits are not cleared on a read of UnSTAT or UnICTRL register.

# 4.0 Embedded Controller Modules (Continued)

## 4.9.4 USART Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### USART Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| UnRBUF | Receive Data Buffer Register | RO |
| UnTBUF | Transmit Data Buffer Register | R/W |
| UnPSR | Baud Rate Pre-Scaler Register | R/W |
| UnBAUD | Baud Rate Divisor Register | R/W |
| UnFRS | Frame Select Register | R/W |
| UnMDSL | Mode Select Register | R/W |
| UnSTAT | Status Register | RO |
| UnICTRL | Interrupt Control Register | Varies per bit |

### Receive Data Buffer Register (UnRBUF)

Location: USART1 - 00 $FD22_{16}$
USART2 - 00 $FC22_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **URBUF** | | | | | | | |

### Transmit Data Buffer Register (UnTBUF)

Location: USART1 - 00 $FD20_{16}$
USART2 - 00 $FC20_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **UTBUF** | | | | | | | |

### Baud Rate Pre-Scaler Register (UnPSR)

Byte-wide read/write register containing the 5-bit pre-scaler value in bit 7 (MSB) through bit 3 (LSB) and the upper three bits of the baud divisor in bit 2 (MSB) through bit 0 (LSB). The register is cleared ($00_{16}$) on reset.

Location: USART1 - 00 $FD2E_{16}$
USART2 - 00 $FC2E_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **UPSC** | | | | | **UDIV[10]: UDIV[8]** | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

# 4.0 Embedded Controller Modules (Continued)

## Baud Rate Divisor Register (UnBAUD)

This byte-wide read/write register contains the lower eight bits of the baud rate divisor. The UnPSR register contains the upper three bits. The register is cleared ($00_{16}$) on reset.

Location: USART1 - 00 $FD2C_{16}$
USART2 - 00 $FC2C_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **UDIV[7]: UDIV[0]** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## Frame Select Register (UnFRS)

This byte-wide read/write register controls the selection of the frame format, including number of data bits, number of stop bits and parity. The register is cleared ($00_{16}$) on reset.

Location: USART1 - 00 $FD28_{16}$
USART2 - 00 $FC28_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **PEN** | **PSEL** | | **XB9** | **STP** | **CHAR** | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 1-0 | **CHAR.** Selects the number of data bits per frame. Note that the parity bit is not included in the number of data bits.<br>**Bits**<br>**1 0 Description**<br>0 0: Frame contains eight data bits (default)<br>0 1: Frame contains seven data bits<br>1 0: Frame contains nine data bits<br>1 1: Loopback mode selected; frame contains nine data bits. |
| 2 | **STP.** Programs the number of stop bits to be transmitted.<br>0: One stop bit transmitted (default)<br>1: Two stop bits transmitted |
| 3 | **XB9.** Contains the value of the 9th data bit for transmission only. The bit has no effect while operating with seven or eight data bits per frame.<br>0: Transmit 0 as 9th data bit (default)<br>1: Transmit 1 as 9th data bit |
| 5-4 | **PSEL.** Controls the mode of parity bit generation and checking. Note that while operating with nine data bits-per-frame, the parity bit is omitted. In this case, the value of PSEL has no effect.<br>**Bits**<br>**5 4 Description**<br>0 0: Odd parity (default)<br>0 1: Even parity<br>1 0: Mark (1)<br>1 1: Space (0) |
| 6 | **PEN.** Enables or disables the generation of a parity bit generation and parity check. Note that there is no parity bit while operating in the nine data bits-per-frame mode. In this case, this bit has no effect.<br>0: Parity disabled (default)<br>1: Parity enabled |
| 7 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## Mode Select Register (UnMDSL)

This byte-wide read/write register controls the selection of the clock source, Synchronous mode, Attention mode and line break generation. It contains the enable bits for the DMA channels. The register is cleared ($00_{16}$) on reset.

Location: USART1 - 00 $FD2A_{16}$

USART2 - 00 $FC2A_{16}$

Type: R/W

USART1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **ERD** | **ETD** | **CKS** | **BRK** | **ATN** | **MOD** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

USART2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | **CKS** | **BRK** | **ATN** | **MOD** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **MOD.** Selects the Synchronous or Asynchronous mode of operation:<br>0: Asynchronous mode (default)<br>1: Synchronous mode |
| 1 | **ATN.** Selects the Attention mode of operation. Cleared by hardware after reception of an address frame that is a 9-bit character with a '1' in the ninth bit position.<br>0: Disable Attention mode (default)<br>1: Enable Attention mode |
| 2 | **BRK.** Setting the bit (1) causes UTXDn to go low. UTXDn remains low until the bit is cleared (0) by the user. |
| 3 | **CKS.** Controls the source of the clock while operating in Synchronous mode (MOD=1).<br>0: USART operates from the baud rate generator and outputs the baud rate clock on USCLKn (default)<br>1: USART operates from an external clock provided on USCLKn<br>While the USART is operated in Asynchronous mode (MOD=0), the bit has no effect. |
| 4 | **ETD.**<br>0: No DMA request is asserted for transmit operations (default)<br>1: DMA request is asserted when the Transmit Buffer Empty (TBE) flag is set (1) |
| 5 | **ERD.**<br>0: No DMA request is asserted for receive operations (default)<br>1: DMA request is asserted when the Receive Buffer Full (RBF) flag is set (1) |
| 7-6 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## Status Register (UnSTAT)

This byte-wide, read-only register contains the receive and transmit status bits. The register is cleared ($00_{16}$) on reset.

Location: USART1 - 00 $FD26_{16}$
USART2 - 00 $FC26_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **XMIP** | **RB9** | **BKD** | **ERR** | **DOE** | **FE** | **PE** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **PE.** The bit is set when a parity error is detected within a received character. The bit is cleared by the hardware when UnSTAT register is read.<br>0: No parity error detected (default)<br>1: Parity error detected in a received byte since the last time UnSTAT was read |
| 1 | **FE.** The bit is set when the USART fails to receive a valid stop bit at the end of a frame. Automatically cleared on read of UnSTAT.<br>0: No framing error detected (default)<br>1: Framing error detected on a received byte since the last time UnSTAT was read |
| 2 | **DOE.** The bit is set when a new character is received and transferred to RBUF before the software has read the previous character. Automatically cleared on read of the UnSTAT.<br>0: No data overrun error detected (default)<br>1: Data overrun error detected since the last time UnSTAT was read |
| 3 | **ERR.** The bit is set any time DOE, FE or PE is set. Automatically cleared if DOE, FE and PE are all zero. This bit is read only. Any attempt to write to the bit by software does not alter its present value.<br>0: No DOE, FE or PE has occurred since the last time UnSTAT register was read (default)<br>1: A DOE, FE or PE error has occurred since the last time UnSTAT register was read |
| 4 | **BKD.** If set, indicates that a line break condition has occurred. A break condition is detected if RXDn remains low for a least ten bit times after a missing stop bit has been detected at the end of a frame. The bit is cleared under the following conditions:<br>– On a read of UnSTAT register, if the break condition on RXDn is no longer present. If RXDn is still low when UnSTAT register is read, the bit is not cleared.<br>– If the read of UnSTAT register did not cause the bit to be cleared because the break condition on RXDn was still in effect, the hardware clears the bit as soon as the break condition no longer exists, i.e., RXDn returns to a high level. |
| 5 | **RB9.** Contains the ninth data bit of the last frame received when operating with the 9-bit data format.<br>0: '0' received in ninth bit position (default)<br>1: '1' received in ninth bit position |
| 6 | **XMIP.** Indicates that the USART is transmitting data. It is reset by hardware at the end of the last frame bit.<br>0: USART is not transmitting (default)<br>1: USART is transmitting |
| 7 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## Interrupt Control Register (UnICTRL)

This byte-wide register contains the interrupt enable bits and the interrupt status flags. The register is set to $01_{16}$ on reset.

Location: USART1 - 00 $FD24_{16}$
USART2 - 00 $FC24_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **EEI** | **ERI** | **ETI** | **Reserved** | | | **RBF** | **TBE** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Type | Description |
|---|---|---|
| 0 | RO | **TBE.** The bit is set by the hardware when the USART transfers data from UnTBUF register to TSFT register for transmission. It is automatically cleared on the next write to UnTBUF register. The bit is set on reset.<br>0: Transmit buffer not empty<br>1: Transmit buffer empty (default) |
| 1 | RO | **RBF.** The bit is set by the hardware when the USART has received a complete data frame and transferred the data from RSFT register to UnRBUF register. The bit is automatically cleared when RBUF register is read.<br>0: Receive buffer not full. New data has not been transferred to RBUF since the last time it was read (default)<br>1: Receive buffer full. RBUF contains new data since the last time it was read |
| 4-2 | | **Reserved.** |
| 5 | R/W | **ETI.** A TX interrupt is generated when the TBE flag is set.<br>0: Disable transmitter interrupt (default)<br>1: Enable transmitter interrupt |
| 6 | R/W | **ERI.** An RX interrupt is generated when the RBF flag is set.<br>0: Disable receiver interrupt (default)<br>1: Enable receiver interrupt |
| 7 | R/W | **EEI.** An RX interrupt is generated when the ERR flag is set, indicating that a receive error has occurred.<br>0: Disable receive error interrupt (default)<br>1: Enable receive error interrupt |

# 4.0 Embedded Controller Modules (Continued)

## 4.9.5 Usage Hints

**Calculating the Baud Rate in Asynchronous Mode**

The equation for calculating the baud rate is:

$$BR = SYS\_CLK/(16xNxP)$$

where:

- BR is the baud rate
- SYS_CLK is the system clock
- N is the value of the baud rate divisor + 1
- P is the pre-scaler divide factor selected by the value in the PSR register

Assuming a system clock of 5 MHz and a desired baud rate of 9600, the NxP term, according to the equation above, is:

$$NxP = (5x10^6)/(16x9600) = 32.552$$

The NxP term is then divided by each pre-scaler factor in Table 20 on page 152 to obtain a value closest to an integer. The factor for this example is 6.5:

$$N = 32.552/6.5 = 5.008\ (N = 5)$$

The baud rate register is programed with a baud rate divisor of 4 (N = baud rate divisor +1). This produces a baud clock of:

$$BR = (5x10^6)/(16x5x6.5) = 9615.385$$

$$\%\ error = (9615.385\text{-}9600)/9600 = 0.16$$

Note that the percent error is much lower than would be possible without the non-integer pre-scaler factor. Refer to the table below for more examples.

| System Clock | Desired Baud Rate | N | P | Actual Baud Rate | Percent Error |
|---|---|---|---|---|---|
| 4 MHz | 9600 | 2 | 13 | 9615.385 | 0.16 |
| 5 MHz | 9600 | 5 | 6.5 | 9615.385 | 0.16 |
| 10 MHz | 19200 | 5 | 6.5 | 19230.769 | 0.16 |
| 20 MHz | 19200 | 5 | 13 | 19230.769 | 0.16 |

**Calculating the Baud Rate in Synchronous Mode**

The equation for calculating the baud rate is:

$$BR = SYS\_CLK/(2xNxP)$$

where:

- BR is the baud rate
- SYS_CLK is the system clock
- N is the value of the baud rate divisor + 1
- P is the pre-scaler divide factor selected by the value in the PSR register

The same procedure is used for determining the values of N and P, as in Asynchronous mode. However, non-integer pre-scale values are not allowed.

# 4.0 Embedded Controller Modules (Continued)

## 4.10 TIMER AND WATCHDOG (TWD)

The Timer and Watchdog module (TWD) generates the clocks and interrupts used for timing periodic functions in the system. It also provides watchdog protection over software execution.

The TWD provides flexibility in system configuration by enabling the configuration of various clock ratios. After setting the TWD configuration, the software can lock the it to provide a higher level of protection against subsequent erroneous software action. Once a section of the TWD is locked, only reset releases it. Figure 58 shows the TWD block diagram.

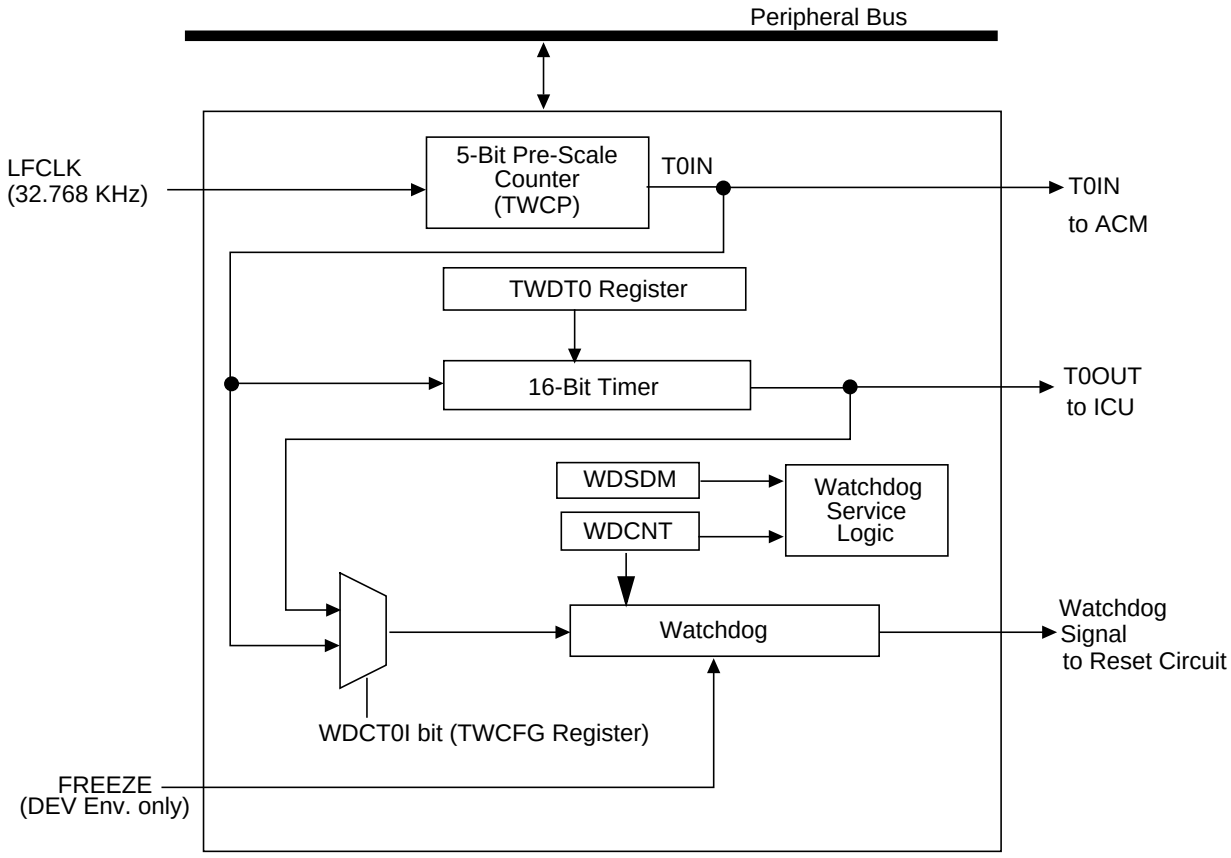



**Figure 58. Timer and Watchdog Block Diagram**

### 4.10.1 Features

- 32.768 KHz input clock
- Programmable pre-scale counter
- 16-bit programmable periodic interrupt timer
- 8-bit watchdog timer
- Watchdog signal generation in response to failure detection, such as:
  - — Watchdog service performed too early
  - — Watchdog service performed too late
  - — Wrong DATA used in a service by data match
- Watchdog input clock selector
- Watchdog Freeze input
- Configuration lock option for fully protected watchdog
- Data match mechanism for watchdog touch

# 4.0 Embedded Controller Modules (Continued)

## 4.10.2 Functional Description

### Input Clock

The TWD bases all its counting activities on a 32.768 KHz clock (LFCLK). The watchdog can count using a division of the 32 KHz clock (either T0OUT or T0IN).

### Pre-Scale

A pre-scale counter divides the LFCLK input clock (32.768 KHz) by a factor of $2^{MDIV}$. MDIV in TWCP register is in the range of 0 through 5 (i.e., divide ratio of 1:1 through 1:32). The pre-scaled output is used as an input clock for a 16-bit timer (TWDT0) and is referred to as T0IN.

### TWD Timer 0

TWD Timer 0 is a 16-bit, programmable, automatically re-triggered down-counter. It counts on the rising edge of T0IN. It starts from the value loaded to TWDT0 register down to zero and then restarts counting from TWDT0 at the next T0IN cycle.

When the counter reaches 0, T0OUT is set (1) for one T0IN cycle. This makes the Timer 0 cycle:

(TWDT0 + 1) x T0IN-cycle.

T0OUT is input to the ICU and can be used as the time base for activities such as system tick.

When TWDT0 is loaded with a new value, the counter uses it the next time it restarts counting (i.e., after reaching zero). If RST in Timer Control register (T0CSR) is written 1, the timer is restarted on the next rising edge of T0IN.

**Notes:**

- RST bit in T0CSR register is cleared after completing this load.
- When MDIV in TWCP register is 0, the timer counter may skip one count when loaded with a new value.

### Watchdog Operation

The watchdog is an 8-bit down counter, operating on the rising edge of its currently selected clock source. On reset, it is disabled (i.e., it does not count and no watchdog signal is generated). A write to the Watchdog Count register (WDCNT) or the Watchdog Service Data Match (WDSDM) register either starts the counter or, if watchdog is already running, performs a restart (“touch”) operation. Once the watchdog is counting down, only a reset can stop it.

Writing to WDCNT register is enabled while LWDCNT in TWCFG register is 0. A write to WDCNT starts the watchdog, and it begins counting down from the written value. If the service on data match is enabled (WDSDME in TWCFG register is 1), writing to WDSDM register with $5C_{16}$ restarts the watchdog timer from the value stored in WDCNT.

A watchdog signal is triggered if one of the following occurs:

- The counter reaches zero (too late service).
- The watchdog is written to more than once per watchdog clock cycle for the currently selected clock (too early service). Writing to the watchdog more than once per three watchdog clock cycles (for the currently selected clock) may cause the watchdog signal to trigger.
- Data other than $5C_{16}$ is written to WDSDM when WDSDME in TWCFG register is 1.

### Watchdog Clock Source Selection

Select the clock source as follows:

- WDCT01 bit in TWCFG register is 0: T0OUT
- WDCT01 bit in TWCFG register is 1: T0IN

Changing the watchdog clock source may cause it to gain or lose one clock cycle.

Notes:

- When MDIV in TWCP register is 0, the watchdog timer may skip one count when loaded with a new value.
- After activating watchdog, avoid entering Idle mode in the first four low-frequency clock cycles.

### TWD Control and Configuration

The TWD Configuration register (TWCFG) allows you to:

- Set the watchdog clock source: T0IN or T0OUT
- Enable watchdog service on write to WDSDM register
- Define which of TWCFG, TWCPR, TWDT0, T0CSR and WDCNT is locked.

Once LTWCFG, LTWCP, LTWDT0 or LWDCNT, in TWCFG register, is set its respective resources are locked and can be cleared only by reset. Setting any of these registers prevents runaway software from tampering with the respective watchdog function.

**Operation in Idle Mode**

The TWD is active in Idle mode: the counters continue to function, and interrupts and error signals are issued.

Write operations to TWCP, TWDT0 and WDCNT may be delayed by up to three 32.768 KHz clock cycles. The software should avoid entering Idle mode during this period. WDTLD bit in T0CSR register indicates when it is safe to switch power modes.

### 4.10.3 TWD Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

**TWD Register Map**

| Mnemonic | Register Name | Type |
|---|---|---|
| TWCFG | Timer and Watchdog Configuration Register | R/W |
| TWCP | Timer and Watchdog Clock Pre-Scaler Register | R/W |
| TWDT0 | TWD Timer 0 Register | R/W |
| T0CSR | TWDT0 Control and Status Register | R/W |
| WDCNT | Watchdog Count Register | WO |
| WDSDM | Watchdog Service Data Match Register | WO |

**Timer and Watchdog Configuration Register (TWCFG)**

The TWCFG register is a byte-wide, read/write register. It defines the watchdog clock input and service method and enables TWD control register locking. Setting the required configuration and locking the TWCFG stops the software from interfering with the watchdog operation. On reset, non-reserved bits of TWCFG are initialized to 0.

Location: 00 $FEE0_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **WDSDME** | **WDCT0I** | **LWDCNT** | **LTWDT0** | **LTWCP** | **LTWCFG** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **LTWCFG.**<br>0: Enables read/write from/to TWCFG register (default)<br>1: Any data written to it is ignored and reading from it returns unpredictable values<br>Once LTWCFG is set, it can only be cleared by reset. |
| 1 | **LTWCP.**<br>0: Enables read/write from/to TWCP register (default)<br>1: Any data written to it is ignored and reading from it returns unpredictable values<br>Once LTWCP is set, it can only be cleared by reset. |
| 2 | **LTWDT0.**<br>0: Enables read/write from/to TWDT0 and T0CSR registers (default)<br>1: Registers cannot be written to and TWDT0 cannot be read. Any data written to TWDT0 or T0CSR is ignored. Reading from TWDT0 returns unpredictable values.<br>Once LTWDT0 is set, it can only be cleared by reset. |

| Bit | Description |
|---|---|
| 3 | **LWDCNT.**<br>0: Enables write to WDCNT register (default)<br>1: Any data written to it is ignored and reading from it returns unpredictable values<br>Once LWDCNT is set, it can only be cleared by reset. When WDSDME bit is cleared, touch operations (i.e., writing to WDCNT register) may be performed when LWDCNT bit is either 0 or 1. |
| 4 | **WDCT0I.**<br>0: Selects T0OUT clock as the watchdog clock (default)<br>1: Selects T0IN as the input clock<br>The hardware clock source selection overrides this clock selection. |
| 5 | **WDSDME.** This bit selects the watchdog touch mechanism<br>0: Disables the watchdog service using WDSDM register. In this case, the watchdog should be serviced by writing a value to WDCNT register. When this bit is cleared, write operations to WDSDM are ignored (default).<br>1: Selects the use of data match using the WDSDM mechanism. |
| 7-6 | **Reserved.** |

### Timer and Watchdog Clock Pre-Scaler Register (TWCP)

The TWCP register is a byte-wide, read/write register. It defines the pre-scale ratio of the input clock and generates the T0IN clock. On reset, the non-reserved bits of TWCP are initialized to 0.

Location: 00 $FEE2_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | **MDIV** | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 2-0 | **MDIV.** Defines the pre-scale ratio of the input clock. The pre-scale ratio is $2^{MDIV}$. The value of MDIV must be in the range of 0-5, providing a pre-scale ratio of 1 to 32.<br>MDIV allowed values:<br>**Bits**<br>**2 1 0 Clock Ratio**<br>0 0 0: 1:1 (default)<br>0 0 1: 1:2<br>0 1 0: 1:4<br>0 1 1: 1:8<br>1 0 0: 1:16<br>1 0 1 1:32<br>Other Reserved |
| 7-3 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## TWD Timer 0 Register (TWDT0)

The TWDT0 register is a read/write register. It defines the T0OUT interrupt rate. On reset, this register is initialized to $FFFF_{16}$.

Location: 00 $FEE4_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **PRESET** | | | | | | | | | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 15-0 | **PRESET.** Defines the counter preset value. Whenever the counter reaches zero, it starts counting down from this value. The T0OUT frequency is the T0IN frequency divided by (PRESET+1). The allowed values of the PRESET field are $0001_{16}$ through $FFFF_{16}$. |

## TWDT0 Control and Status Register (T0CSR)

The T0CSR register is a read/write register. It controls the operation and provides the status of the T0 timer. The non-reserved bits of T0CSR are cleared (0) on reset.

Location: 00 $FEE6_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | **WDLTD** | **Reserved** | **TC** | **RST** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **RST (Reset).** When set (1), forces the timer to restart counting in the next input clock rising edge. The bit is cleared by the input clock rising edge, indicating that the counter resumed its automatic re-triggerable operation. Writing 0 to this bit is ignored. |
| 1 | **TC (Terminal Count).** Indicates that the counter has reached zero (terminal count). This bit is cleared each time the register is read. It is a read-only bit, and data written to it is ignored. |
| 2 | **Reserved.** |
| 3 | **WDLTD (Watchdog Last Touch Delay).** The bit is set when the WDCNT is written. It is cleared after watchdog is updated. (After watchdog is updated, it is safe to switch to Idle mode.) |
| 7-4 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## Watchdog Count Register (WDCNT)

The WDCNT register is a byte-wide, write-only register. It holds the value loaded into the watchdog timer when it is touched and counts down from it. The watchdog is started by the first write to the register. Each successive write restarts the watchdog timer. A write to WDCNT functions as a touch operation when WDSDME bit TWCFG register is cleared, even if WDCNT is locked; in this case, the watchdog timer is restarted using the value loaded in PRESET field before WDCNT was locked (i.e., the new PRESET value is ignored). On reset this register is initialized to $0F_{16}$.

Location: 00 $FEE8_{16}$

Type: WO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **PRESET** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 7-0 | **PRESET.** Defines the counter preset value. |

## Watchdog Service Data Match Register (WDSDM)

The WDSDM register is a byte-wide, write-only register. When WDSDME in TWCFG register is set, the watchdog counting restarts from the value in WDCNT, when WDSDM is written with $5C_{16}$. If any other data is written to this register, it triggers a watchdog signal. If RSDATA is written more than once per three watchdog clock cycles, a watchdog signal is also triggered. When the WDSDME bit is cleared, a write to this register is ignored.

Location: 00 $FEEA_{16}$

Type: WO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **RSDATA** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **RSDATA.** |

### 4.10.4 Usage Hints

The TWD protects watchdog operation from software tampering. To achieve the highest level of protection, proceed as follows:

1. Program the TWDT0 pre-scale and TMWT0 timers to the desired values.
2. Configure the watchdog clock to use T0IN or T0OUT using WDCT0I bit in TWCFG register.
3. Program the WDCTL to the maximum period between watchdog touch operations. Note that from this point, the watchdog starts operating and must be touched periodically to prevent a watchdog error signal.
4. Configure the watchdog to use data match, and lock all the TWD configuration and setting registers by setting bits 0 through 4 and bit 6 of the TWCFG.
5. Touch the watchdog by writing $5C_{16}$ to WDSDM at the appropriate rate (i.e., no more than once every watchdog clock cycle and no less than the period programed to WDCTL).

# 4.0 Embedded Controller Modules (Continued)

## 4.11 ANALOG TO DIGITAL CONVERTER (ADC)

The Analog to Digital Converter (ADC) monitors various voltages in the system and reports their values to the core.

The ADC can measure, with 8-bit resolution, up to 10 external voltage inputs and four internal voltage sources. The internal voltage sources measure the $V_{DD}$, $V_{CC}$, $AV_{CC}$ and $V_{BAT}$ supplies. $V_{DD}$, $V_{CC}$ and $AV_{CC}$ are divided by 2 to allow both overvoltage and undervoltage detection. The external voltage inputs support thermistor-based temperature measurement.

The ADC executes cycles of three voltage measurements, each assigned to a separate output channel. Each of the three voltage channels measures one selected voltage input during the cycle.

### 4.11.1 Features

- Voltage measurement
  - — Ten external voltage inputs and four internal power supply inputs
  - — 8-bit resolution
  - — 0V to 2.97V input voltage range
  - — High-impedance, ground-referenced inputs
  - — Enables thermistor-based temperature measurement
- Internal high-precision reference
- Digital reading output channels
  - — Three voltage buffers
  - — Input selection for each voltage channel
- Sampling sequence and timing
  - — Three voltage measurement within 100 ms
  - — Cyclic measurement of the three output channels
  - — Separate enable for each channel
  - — Programmable conversion-start delay to guarantee input settling time
- Polling- or interrupt-driven interface
- Power consumption
  - — Zero current when disabled
  - — Low operating current

### 4.11.2 Functional Description

**Inputs.** The ADC has 16 inputs (AI0-13, AI15, AI16) divided in three groups:

- External Voltage (AI0-9). These are either temperature-dependent voltage generated by using a Negative/Positive Temperature Coefficient (NTC/PTC) thermistor in a resistive divider, or general-purpose, positive DC voltage sources.
- Internal Voltage (AI10-13). These are internally connected to the supply voltages of the device ($V_{DD}$, $V_{CC}$, $AV_{CC}$ and $V_{BAT}$). Voltages higher than the full-scale voltage ($V_{FS}$) are divided by 2, except for $V_{BAT}$ which, in order to minimize the current drain, is not divided.

**Analog Multiplexer.** A 16 to 1 analog multiplexer selects one of the inputs for measurement by connecting it to the A/D converter. Switching between inputs at different voltage levels requires delaying the conversion start until the input voltage to the A/D converter has settled to the desired accuracy.

**A/D Converter.** The $\Sigma$–$\Delta$, high-resolution A/D converter receives the selected input and converts it. The result of the conversion is either an 8-bit, unsigned integer digital value (0 to 255) for voltage inputs.

A high-precision internal reference generator sets the full-scale voltage value ($V_{FS}$) of the A/D converter.

**ADC Cycle.** The ADC has three output “channels” for voltage measurement. The voltage measurement channels are not related to specific inputs. They hold the input select control data for the next measurement and contain a buffer in which the conversion result is stored.

An ADC cycle includes measurements of all three channels and a calibration operation. The first measurement is of the calibration measurement and the three voltage channels in ascending order, each for the specific input number contained in its control register. The ADC waits for a programmable delay between the selection of an input and the A/D conversion start that is necessary for A/D input settling to the required accuracy.

At the end of each A/D conversion, the result is stored in the corresponding buffer and a Data Valid (DATVAL) flag is set. At the end of the ADC cycle, a flag is set indicating that all the channels contain new data. If enabled, an interrupt request is sent to the core.

## 4.0 Embedded Controller Modules (Continued)

After reading the data at the end of the cycle, the software may choose to set up new values for the input select control data in order to measure different inputs during the following cycle. Another option is to use the old values for one or more voltage channels and repeat the measurement on the previous inputs for a higher sampling rate. Any voltage output channel may be disabled in order to skip its related measurement during the ADC cycle (shorter cycle).

**ADC Control/Status Registers and Data Out Buffers.** These may be read/written by the core through the Peripheral bus.

**Timing Control.** This block reduces the frequency of the system clock to the lower value required by the ADC.

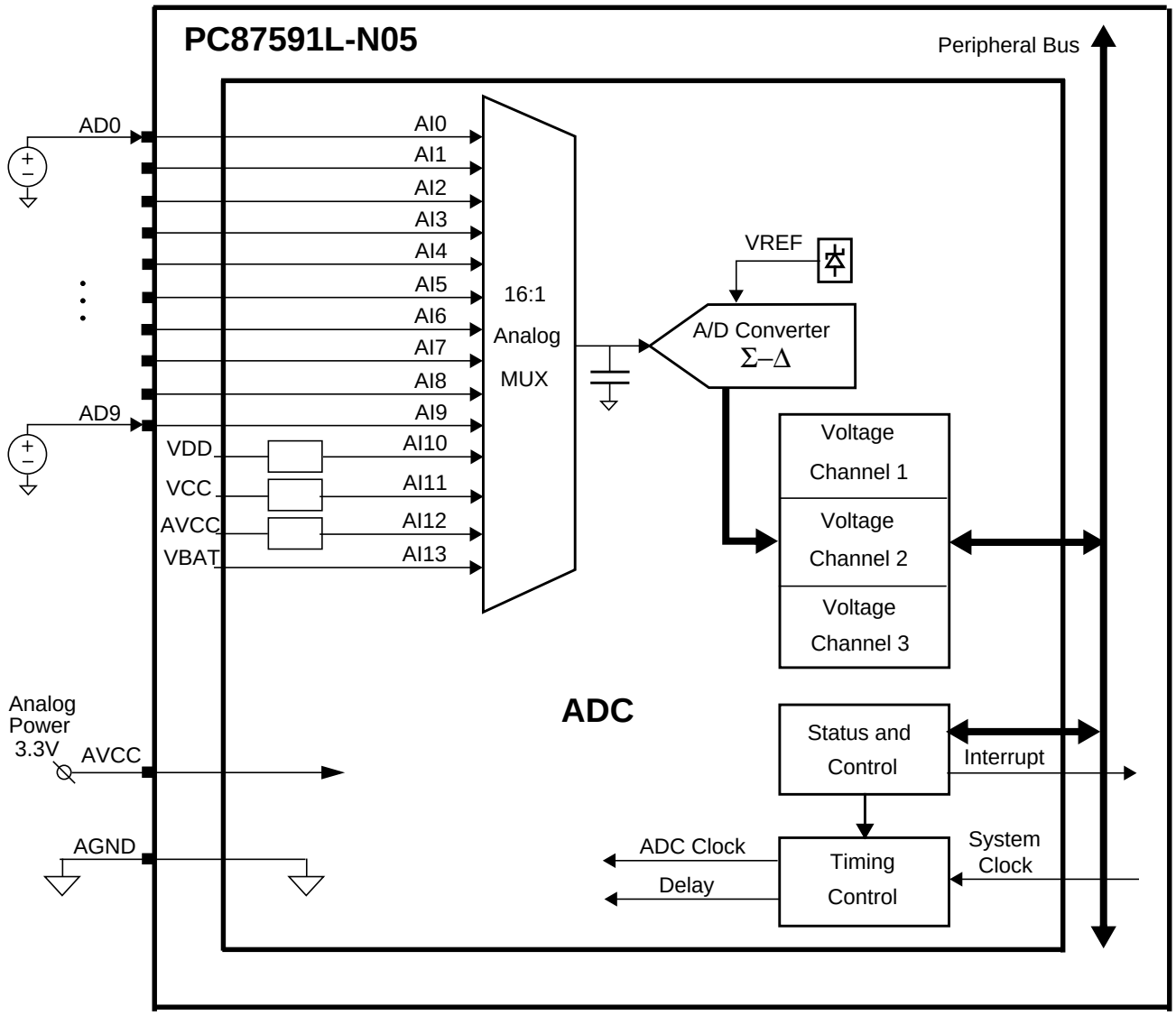



**Figure 59. ADC Functional Diagram**

### 4.11.3 Voltage Measurement

The ADC performs a linear conversion of the input voltage signal to an 8-bit, unsigned digital representation. The input signal should be applied relative to the AGND pin and should range from 0V to $V_{FS}$ (Full Scale).

## 4.0 Embedded Controller Modules (Continued)

Use the equations in the following table to calculate the input voltage based on the reading from the Voltage Channel Data result (VCHDAT field in VCHNxDAT register).

| Input Channel | Scale | Calculation[1] |
|---|---|---|
| AD0 to AD9 | Low | Vi = VCHDAT(9-2) * (1 / 256) * $V_{FSL}$ |
| AD0 to AD9 | High | Vi = VCHDAT(9-2) * (1 / 256) * $V_{FSH}$ |
| AD10 to AD13 | -[2,3] | Vi = VCHDAT(9-2) * (1 / 256) * $V_{FSV}$ |

1. See Section 7.4.1 on page 340 for the dynamic range relevant for each input.
2. No Scale (High or Low) is defined for these inputs.
3. These inputs are scaled down by 4 at the input and compensated back at the result read phase.

The input voltage is converted to Voltage Channel Data result (VCHDAT field in VCHNxDAT register) according to the following table:

| Input Voltage | Result |
|---|---|
| 0V (ground) | VCHDAT(9-2) = $00_{16}$ |
| (255/256) * $V_{FS}$ | VCHDAT(9-2) = $FF_{16}$ |

Changing the input selection for a new measurement requires switching between inputs at different voltage levels. The input interface circuits of the ADC, together with the externally added noise-rejection filters (if applicable), requires a settling time to reach the new voltage value with 8-bit accuracy (less than 1/2 LSB error). Therefore, the ADC waits for a programmable delay time between the selection of the input to be measured and the beginning of the A/D conversion. This Voltage Channel Delay is expressed in ADC clock cycles in ADC Delay Control register (ADLYCTL). The number of ADC clock cycles should be converted to time using the following formula:

$$t_{VD} = Number\_of\_ADC\_clocks * (System\_clock\_cycle) * SCLKDIV(5\text{-}0)$$

To calculate the required delay value according to externally added components, see Section 4.11.6 on page 176.

### 4.11.4 ADC Operation

#### Reset

Section 3.2 on page 61 describes the types of PC87591L-N05 resets. The ADC is affected by the core domain reset events, as described below:

All control, configuration and status registers are reset to their default values, as indicated in Section 4.11.5 on page 171. The Voltage (1, 2 and 3) Channel Data Buffer registers are not reset, since their value is undefined until the first measurement occurs (on each of them).

The ADC is disabled, with all interrupt sources masked and all event status bits reset. The clock division factor, as well as the voltage channel delay, are all set to their maximum value (for the slowest ADC operation speed). Each of the three channels is individually disabled, along with its interrupt source. The Selected Input for all three voltage channels is set to $1F_{16}$ (disabled).

#### ADC Clock

The ADC clock is generated by dividing the system clock by a factor in the range of 4 to 63, as defined in SCLKDIV field in ACLKCTL register (see Section 4.11.5 on page 171). The system clock's source is the on-chip clock multiplier (see Section 4.18 on page 212). The ADC clock needs to be at a frequency of 0.5 MHz. SCLKDIV must be programed prior to enabling the ADC (i.e., while ADCEN of the ADCCNF register is 0).

#### Initializing the ADC

The ADC must be initialized before it is enabled. The following steps need to be taken to initialize it before enabling the ADC (i.e., ADCEN bit in ADCCNF register is cleared):

- System Clock Division Factor - SCLKDIV field in ACLKCTL register.
- Voltage Channel Delay - VOLDLY field in ADLYCTL register.

## 4.0 Embedded Controller Modules (Continued)

### Enabling and Disabling the ADC

**Enabling the ADC.** The ADC is enabled by setting ADCEN in ADCCNF register to 1.

After the ADC is enabled, its internal circuits need an activation delay of 100 µs. This activation delay should be added to the ADC cycle until the first batch of measurements (after enabling the ADC) is available. Note that after activation, the first set of results using the large scale mode (CSCALE bit in VCHiCTL register is clear) may be wrong.

Other measurement operations may be enabled or disabled individually. When measurement conversions are enabled while the ADC is enabled, the measurement operations start on the following conversion cycle.

**Disabling the ADC.** The ADC is disabled by resetting ADCEN in ADCCNF register when one of these conditions applies:

- $V_{CC}$ Power-Up reset
- Warm reset
- Core enters Idle mode
- The software resets the ADCEN bit.

In this state, all ADC activities are halted and ADC current consumption from the $AV_{CC}$ is reduced. Note that re-enabling the ADC causes an activation delay.

It is recommended to disable the ADC only after the buffer registers of all four channels have been read.

### Interrupt Structure

The ADC Interrupt is a level high interrupt, generated if one (or more) of the events in Table 21 becomes active. The ADC interrupt is connected to the ICU.

**Table 21. ADC Interrupt Structure**

| Event Flag | Register Mnemonic | Mask Bit | Register Mnemonic | Description |
|---|---|---|---|---|
| EOCEV | ADCSTS | INTECEN | ADCCNF | End-of-Cycle Event/Enable |
| DATVAL | VCHN1CTL | INTDVEN | VCHN1CTL | Data Valid Event/Enable (Voltage Channel 1) |
| DATVAL | VCHN2CTL | INTDVEN | VCHN2CTL | Data Valid Event/Enable (Voltage Channel 2) |
| DATVAL | VCHN3CTL | INTDVEN | VCHN3CTL | Data Valid Event/Enable (Voltage Channel 3) |

When an event flag and its related mask bit are set (enabled), the ADC Interrupt request is asserted. This is indicated by a high level of the ADC Interrupt signal.

The software must reset the event flag (or reset its mask bit) in order to deassert the ADC Interrupt request.

All the interrupt mask bits (interrupt disabled), the data-related event flags (EOCEV and the four DATVAL bits) are cleared by both reset conditions.

The ADC Interrupt is routed to the ICU as an ADCI signal (see Section 4.3 on page 96).

### ADC Operating Principles

**Measurement Sequence.** The following measurements are executed during one ADC cycle, for all enabled channels (in the following order):

1. Calibration measurement.
2. Voltage measurement for Voltage Channel 1, from the input selected by SELIN field in VCHN1CTL register. The A/D conversion starts by selecting the input and waiting for the time period (VOLDLY delay) set in ADLYCTL register. The resulting 8-bit digital value is stored in VCHN1DAT register, and DATVAL bit in VCHN1CTL register is set. **Note:** This measurement is skipped if Voltage Channel 1 is disabled by setting the SELIN field in VCHN1CTL register to $1F_{16}$.
3. Voltage measurement for Voltage Channel 2, as above, using the VCHN2CTL and VCHN2DAT registers.
4. Voltage measurement for Voltage Channel 3, as above, using the VCHN3CTL and VCHN3DAT registers.
5. End of the ADC cycle. EOCEV bit in ADCSTS register is set (in addition to all the relevant DATVAL bits).

The software may read the measurement result for each channel immediately after its DATVAL bit is set. Alternatively, the results may be read at the end of the cycle when EOCEV bit is set. After the data in VCHNiDAT register is read, the software should clear the relevant DATVAL bit to indicate that the data in the buffers has been read.

The ADC cycle duration may be calculated using the formula below (N is the number of enabled voltage channels):

$$T_{ADC\ cycle} = 42.2\ ms + (N+1) * (t_{VD} + t_{VC})$$

## 4.0 Embedded Controller Modules (Continued)

Where:

$t_{VD}$ - Voltage Conversion Delay Time
$t_{VC}$ - Voltage Conversion time

See Section 7.4 on page 340 for the value of $t_{VC}$ and Section Section 4.11.3 on page 167 for $t_{VC}$ calculation.

**Input Selection Field.** Each Voltage Channel has its own programmable, input selection field (SELIN in VCHNxCTL register). This field determines which input is measured by the channel during the current ADC cycle. The field also indicates to which input the data in the channel buffer belongs. This field may be modified after the channel buffer has been read and the DATVAL bit has been reset.

If the input selection field is not changed, the same input is measured during the next ADC cycle. This gives a sampling rate of one $T_{ADC\ cycle}$ for the specific input. If this field is changed and a different input is sampled, the sampling rate is lower for each input, but the period for all scanned inputs is shorter.

**Operation Sequence.** After the ADC is properly initialized and enabled, one of the following example sequences can be used:

- EOCEV-driven ADC operation sequence for all voltage channels
- DATVAL-driven ADC operation sequence for one voltage channel

**EOCEV-Driven ADC Operation Sequence for All Channels**

1. When End-of-Cycle is reached (i.e., after all enabled channel conversions are completed), software can detect the event by waiting for EOCEV bit in ADCSTS register to be set to 1.
2. Read the number of input measured in Voltage Channel 1 by reading SELIN in VCHN1CTL register.
3. Read the input voltage value measured in Voltage Channel 1 by reading VCHDAT in VCHN1DAT register.
4. In preparation for the next measurement (i.e., to define which input will be measured by Voltage Channel 1 during the next ADC cycle), clear DATVAL bit in VCHN1CTL register by writing 1 to it (it may be the same input or, optionally, a different one).
5. For Voltage Channel 2, repeat steps 5 through 7 for the VCHN2CTL and VCHN2DAT registers.
6. For Voltage Channel 3, repeat steps 5 through 7 for the VCHN3CTL and VCHN3DAT registers.

**DATVAL-Driven ADC Operation Sequence for One Channel**

1. Wait for the end of channel by waiting for DATVAL in VCHNxCTL register to be set to 1.
2. Read the input number by reading SELIN in VCHNxCTL register.
3. Read the measured data by reading VCHDAT in VCHNxDAT register.
4. Optional: Change the input to be measured during the next ADC cycle: in VCHNxCTL register, write a new SELIN value.
5. Prepare the voltage channel to receive new data: in VCHNxCTL register, write 1 to DATVAL to clear it.
6. In preparation for the next measurement (i.e., to define which input will be measured by the voltage channel during the next ADC cycle), clear the DATVAL bit by writing 1 to it (it may be the same input or, optionally, a different one).

### Reading Measurement Results

**Polling-Driven Operation.** Measurement results may be read by polling either EOCEV in ADCSTS register or each of the three DATVAL bits in VCHNxCTL registers.

Polling EOCEV uses the sequence listed in EOCEV-Driven ADC Operation Sequence for All Channels, above. When EOCEV is set, all three channels contain valid data and may be read.

Polling DATVAL uses the sequence listed in DATVAL-Driven ADC Operation Sequence for One Channel, above. When a DATVAL bit is set, only its channel contains valid data that may be read. In this case, the EOCEV bit is redundant.

**Interrupt-Driven Operation.** The ADC may generate an interrupt to the core when any of the valid bits is set (EOCEV in ADCSTS register or DATVAL in any VCHNxCTL register). The interrupt is generated when the interrupt enable bit for the respective status bit is set. The software in the interrupt routine should check the status bits as described above for polling-driven operation to verify which of the DATVAL bits is set.

An interrupt is expected from EOCEV when using the sequence listed in the EOCEV-driven ADC operation sequence (see above). The EOCEV interrupt indicates that all three channels contain valid data and may be read. Interrupts from the DATVAL bits should be disabled.

An interrupt is expected from one of the DATVAL bits when using the sequence listed in the DATVAL-driven ADC operation sequence (see above). The DATVAL interrupt indicates that only its channel contains valid data that may be read. Interrupts from the EOCEV bit should be disabled.

### Failure Detection

**Overflow.** An overflow occurs when DATVAL bit is set at the end of a measurement, indicating that the result of the previous measurement was not read. If an overflow occurs in at least one channel, the new measurement result overrides the old data in the buffer, and OVFEV in ADCSTS register is set. This indicates that the result of the previous measurement was lost.

# 4.0 Embedded Controller Modules (Continued)

## 4.11.5 ADC Registers

The ADC control/status and data out registers set interfaces with the core through the Peripheral bus. These registers are mapped in the address space of the core, starting at the base address defined in Appendix 61 on page 393.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### ADC Register Map

The ADC register set contains six common, control and status registers and six channel-specific registers.

**Table 22. ADC Register Map**

| Mnemonic | Register Name | Type |
|---|---|---|
| ADCSTS | ADC Status | Varies per bit |
| ADCCNF | ADC Configuration | R/W |
| ACLKCTL | ADC Clock Control | R/W |
| ADLYCTL | ADC Delay Control | R/W |
| ADCPINX | ADC Parameters Index | R/W |
| ADCPD | ADC Parameters Data | R/W |
| VCHN1CTL | Voltage Channel 1 Control | Varies per bit |
| VCHN1DAT | Voltage Channel 1 Data Buffer | RO |
| VCHN2CTL | Voltage Channel 2 Control | Varies per bit |
| VCHN2DAT | Voltage Channel 2 Data Buffer | RO |
| VCHN3CTL | Voltage Channel 3 Control | Varies per bit |
| VCHN3DAT | Voltage Channel 3 Data Buffer | RO |

### ADC Status Register (ADCSTS)

This register indicates the global status of the ADC module. ADCSTS is cleared ($00_{16}$) on $V_{CC}$ Power-Up reset; on other resets, bit 2 is unchanged and other bits are cleared.

Location: 00 $FF20_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | **OVFEV** | **EOCEV** |
| Warm Reset | 0 | 0 | 0 | 0 | 0 | - | 0 | 0 |
| Power-Up Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Type | Description |
|---|---|---|
| 0 | RO | **EOCEV (End-of-Cycle Event).** End of ADC cycle; all enabled measurements (up to four) are completed. For each of the enabled channels, the DATVAL bit in the respective VCHNxCTL register is set, and the data for the channel is stored in the respective Channel Data Buffer register.<br>0: Cycle in progress (default)<br>1: End of ADC cycle (the bit remains set until all DATVAL bits in the Channel Control registers are reset) |
| 1 | R/W1C | **OVFEV (Data Overflow Event).** Measurement data from the previous cycle was overwritten with data from the current cycle before being read. In the event of a data overflow, the DATVAL bit remains set and new data is placed in Channel Data Buffer register.<br>0: No overflow (default)<br>1: Overflow |
| 7-2 | | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## ADC Configuration Register (ADCCNF)

This register controls the operation and global configuration of the ADC module. ADCCNF is cleared ($00_{16}$) on reset.

Location: 00 $FF22_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | **INTECEN** | **Reserved** | **ADCEN** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **ADCEN (ADC Module Enable).** Controls the operation of the ADC to minimize power consumption and prevent glitch effects during configuration changes; see “Enabling and Disabling the ADC” on page 169.<br>0: ADC disabled (default)<br>1: ADC enabled |
| 1 | **Reserved.** |
| 2 | **INTECEN (Interrupt from End-of-Cycle Event Enable).** Enables generation of an ADC interrupt on an End-of ADC-cycle event (EOCEV in ADCSTS register).<br>0: Disabled (default)<br>1: Enabled - ADC Interrupt from EOCEV |
| 7-3 | **Reserved.** |

## ADC Clock Control Register (ACLKCTL)

This register controls the system clock to ADC clock division. ACLKCTL is set to $3F_{16}$ on reset.

Location: 00 $FF24_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **SCLKDIV** | | | | | |
| Reset | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 5-0 | **SCLKDIV (System Clock Division Factor).** Used to divide the system clock in order to obtain the ADC clock. The system clock frequency is set separately (see Figure 74 on page 212). The resulting ADC clock frequency should be equal to 0.5 MHz.<br>Range: 4 to 63 (default is 63, decimal values); values 0 to 3 are illegal and may result in undetermined ADC behavior. |
| 7-6 | **Reserved.** |

## ADC Delay Control Register (ADLYCTL)

This register controls the delay between “input switching” and “conversion start” for the voltage and temperature channels. ADLYCTL is set to $A7_{16}$ on reset.

Location: 00 $FF26_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | **VOLDLY** | | |
| Reset | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| Must be | 0 | 0 | 0 | 1 | 0 | X | X | X |

| Bit | Description |
|---|---|
| 2-0 | **VOLDLY (Voltage Channel Delay).** Compensates for the settling time of the input interface and externally added filter (if used). The conversion of the VOLDLY value to delay, in terms of ADC clock cycles, is shown below. To calculate the required value and the resulting delay time, see Section 4.11.3 on page 167 and "Calculating the Voltage Channel Delay" on page 177.<br>**VOLDLY Value Bits 2 1 0: — Voltage Channel Delay (ADC Clock Cycles)**<br>0 0 1: — 4<br>0 1 0: — 8<br>0 1 1: — 16<br>1 0 0: — 32<br>1 0 1: — 64<br>1 1 0: — 128<br>1 1 1: — 256 (default)<br>Other: — Reserved |
| 7-3 | **Reserved (must be set to '00010').** |

### ADC Parameters Index Register (ADCPINX)

This register holds an index to the ADC parameters registers. Use it while initializing the ADC (see "Initializing the ADC" on page 168 for more details). ADCPINX is cleared ($00_{16}$) on reset.

Location: 00 $FF2A_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Index** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Index.** Defines which parameter register is being accessed by the ADCPD register. Use only the Index values recommended by National Semiconductor. Accessing other Index values may cause the ADC module to behave in an undefined manner. |

### ADC Parameters Data Register (ADCPD)

This register enables access to the ADC parameters registers. Use it while initializing the ADC (see "Initializing the ADC" on page 168 for more details).

Location: 00 $FF2C_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Parameter Data** | | | | | | | | | | | | | | | |

| Bit | Description |
|---|---|
| 15-0 | **Parameter Data.** The register is used to access the ADC parameters register that the Index register (ADCPINX) points to.<br>Writing $01_{16}$ to index $00_{16}$ locks the parameter data registers from any further writes until the next reset. |

# 4.0 Embedded Controller Modules (Continued)

## Voltage Channel 1 Control Register (VCHN1CTL)

This register both controls the operation and indicates the status of Voltage Channel 1. VCHN1CTL is set to $1F_{16}$ on reset.

Location: 00 $FF34_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **DATVAL** | **CSCALE** | **INTDVEN** | **SELIN** | | | | |
| Reset | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

| Bit | Type | Description |
|---|---|---|
| 4-0 | R/W | **SELIN (Selected Input).** Selects a voltage input to be measured during the next ADC cycle. The new value should be set before the beginning of the channel measurement. When read, this field indicates to which input the contents of the channel data buffer belongs. When written, it selects a new input for the next channel measurement. See Figure 59 on page 167 for details on the channel input signals connection to pins.<br>The channel is disabled by setting this field to $11111_2$. When disabled, the channel is "skipped", enabling a higher measurement rate for the remaining channels (shorter ADC cycle). In addition, the DATVAL bit is cleared.<br>**Bits**<br>**4 3 2 1 0 Description**<br>0 0 0 0 0: Channel 0<br>0 0 0 0 1: Channel 1<br>....<br>0 1 0 0 0: Channel 8<br>0 1 0 0 1: Channel 9<br>0 1 0 1 0: Channel 10<br>....<br>0 1 1 0 1: Channel 13<br>1 1 1 1 1: Channel Disabled (default)<br>Other: Reserved |
| 5 | R/W | **INTDVEN (Interrupt from Data Valid Enable).** Enables generation of an ADC interrupt on a Voltage Channel 1, Data Valid event (End-of-conversion).<br>0: Disabled (default)<br>1: Enabled - ADC Interrupt from local DATVAL |
| 6 | R/W | **CSCALE (Channel Scale).** Controls the input scale of the input for the channel to be converted (as selected by SELIN field).<br>0: Channel uses the high range scale (default)<br>1: Channel uses the low range scale |
| 7 | R/W1C | **DATVAL (Data Valid).** Voltage Channel 1 Data Buffer contains new data. The data may be read immediately or at the end-of-cycle. This flag is cleared when the channel is disabled, when the ADC module is disabled (ADCEN in ADCCNF register is cleared) or by a write of 1 to it.<br>0: No new valid data in VCHN1DAT register (default)<br>1: End of conversion - new data is available in the buffer |

## Voltage Channel 1 Data Buffer (VCHN1DAT)

This register (buffer) holds the data measured by Voltage Channel 1.

Location: 00 $FF36_{16}$

Type: RO

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | **VCHDAT** | | | | | | | | **Reserved** | |

| Bit | Description |
|---|---|
| 1-0 | **Reserved.** |
| 9-2 | **VCHDAT (Voltage Channel 1 Data).** Selected input voltage data, measured by Voltage Channel 1. To calculate the voltage, see Section 4.11.3 on page 167. VCHN1DAT holds valid result when DATVAL bit in VCHN1CTL register is set. To prepare for consecutive result, clear DAVAL after reading the data from VCHDAT.<br><br>Range: 0 to 255 (0 to $V_{FS}$); 8-bit, unsigned value with 1 LSB = $V_{FS}$/256. |
| 15-10 | **Reserved.** |

### Voltage Channel 2 Control Register (VCHN2CTL)

This register controls the operation and indicates the status of Voltage Channel 2. VCHN2CTL is set to $1F_{16}$ on reset.

Location: 00 $FF38_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **DATVAL** | **CSCALE** | **INTDVEN** | **SELIN** | | | | |
| Reset | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

| Bit | Type | Description |
|---|---|---|
| 4-0 | R/W | **SELIN (Selected Input).** Same as Voltage Channel 1. |
| 5 | R/W | **INTDVEN (Interrupt from Data Valid Enable).** Same as Voltage Channel 1. |
| 6 | R/W | **CSCALE (Channel Scale).** Same as Voltage Channel 1. |
| 7 | R/W1C | **DATVAL (Data Valid).** Same as Voltage Channel 1. |

### Voltage Channel 2 Data Buffer (VCHN2DAT)

This register (buffer) holds the data measured by the Voltage Channel 2.

Location: 00 $FF3A_{16}$

Type: RO

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | **VCHDAT** | | | | | | | | **Reserved** | |

| Bit | Description |
|---|---|
| 1-0 | **Reserved.** |
| 9-2 | **VCHDAT (Voltage Channel 2 Data).** Same as Voltage Channel 1. |
| 15-10 | **Reserved.** |

### Voltage Channel 3 Control Register (VCHN3CTL)

This register controls the operation and indicates the status of Voltage Channel 3. VCHN3CTL is set to $1F_{16}$ on reset.

Location: 00 $FF3C_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **DATVAL** | **CSCALE** | **INTDVEN** | **SELIN** | | | | |
| Reset | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

| Bit | Type | Description |
|---|---|---|
| 4-0 | R/W | **SELIN (Selected Input).** Same as Voltage Channel 1. |
| 5 | R/W | **INTDVEN (Interrupt from Data Valid Enable).** Same as Voltage Channel 1. |
| 6 | R/W | **CSCALE (Channel Scale).** Same as Voltage Channel 1. |
| 7 | R/W1C | **DATVAL (Data Valid).** Same as Voltage Channel 1. |

### Voltage Channel 3 Data Buffer (VCHN3DAT)

This register (buffer) holds the data measured by Voltage Channel 3.

Location: 00 $FF3E_{16}$

Type: RO

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | **VCHDAT** | | | | | | | | **Reserved** | |

| Bit | Description |
|---|---|
| 1-0 | **Reserved.** |
| 9-2 | **VCHDAT (Voltage Channel 3 Data).** Same as Voltage Channel 1. |
| 15-10 | **Reserved.** |

## 4.11.6 Usage Hints

### Power Supply and Layout Guidelines

For more information, see Section 3.1.3 on page 59.

### Power Consumption

ADC power consumption from the analog supply ($AV_{CC}$) is practically zero if the ADC is disabled by setting ADCEN in ADCCNF register to 0.

When the ADC is enabled, the current consumption depends on the channel type measured and the ADC clock frequency. To minimize current consumption, disable the ADC when not in use. See details in “Enabling and Disabling the ADC” on page 169.

### Back-Drive Protection

To maintain the high performance of the analog circuits, AD0-7, AD8 and AD9 pins are not back-drive protected. Therefore, the voltage applied to these pins must be within the AGND to $AV_{CC}$ range; otherwise, the device may be damaged. External circuits should not drive currents into these pins when the PC87591L-N05 is not powered up.

### Measuring Out of Range Voltages

The ADC is capable of measuring positive input voltages from 0V to $V_{FS}$. Input voltages outside this range should either be divided or level-shifted, as required.

## 4.0 Embedded Controller Modules (Continued)

For positive input voltages higher than $V_{FS}$, place a resistor divider in front of the PC87591L-N05 input pin (see Figure 60a). The divider should be calculated so that its output is lower than the full-scale value ($V_{FS}$) for the maximum input signal voltage.

For negative input voltages, place a resistive level-shifter in front of the PC87591L-N05 input pin (see Figure 60b). The level-shifter should be calculated so that its output is higher than 0V for the minimum input signal voltage.

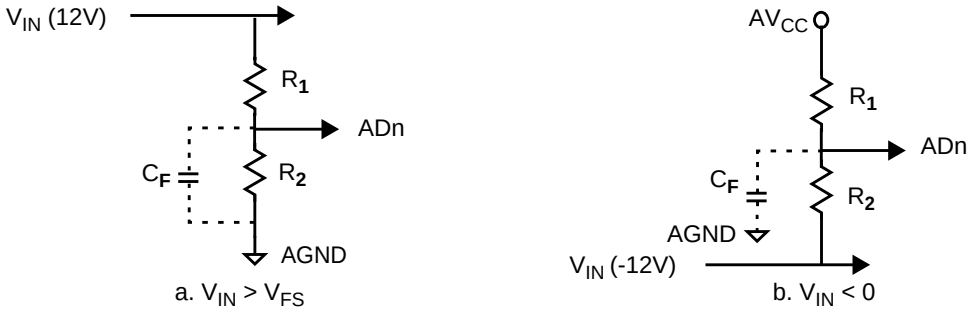



**Figure 60. Measurement of Positive and Negative Voltages**

### Filtering the Noise on Voltage Input Signals

Noise may be coupled to the input signal for various reasons, including close proximity to digital circuits. The slow change rate of the input signals, makes it possible to use an external Low Pass Filter (LPF). This can be implemented by placing a capacitor ($C_F$) between the divider output and AGND, as shown in Figure 60. The cutoff frequency of this LPF should be at least 22 times the maximum signal frequency required to be measured with 8-bit accuracy (a smaller capacitor may be used when a lower accuracy is acceptable). The following formula demonstrates the calculation of the components.

$$f_{(-3dB)} = 1 / (2 * \pi * R_{eq} * C_F)$$

where:$R_{eq} = (R_1 * R_2) / (R_1 + R_2)$

### Calculating the Voltage Channel Delay

The Voltage measurement delay time ($t_{VD}$) is the period between input selection and the A/D conversion start. This delay should be long enough to guarantee the voltage settling at the input of the A/D converter to within 1/2 LSB. This includes the input interface circuits of the ADC and the externally added noise-rejection filters (if applicable). Figure 61 shows the equivalent R-C circuit of a filtered input. $R_{eq}$ represents the equivalent resistance of the input divider or level-shifter (see the preceding section, Filtering the Noise on Voltage Input Signals). $C_F$ is the filter capacitor. The Input Interface Circuit includes the input the multiplexer.

Since the R-C of the input circuit is short, typically, all that is required is a very short delay, which shortens the overall measurement time. The delay length may be sized during development by increasing the delay time from the minimum value to the point were the ADC readout is not affected by the delay value (i.e., use the minimal delay to which the ADC readout is invariant).

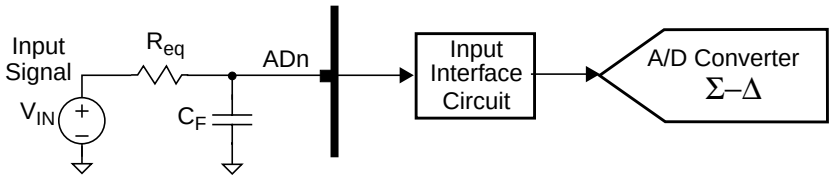



**Figure 61. Filtered Voltage Input Equivalent Circuit**

### Thermistor-Based Temperature Measurement

The ADC is capable of measuring temperature by means of NTC or PTC thermistor devices. These elements change their resistance according to their temperature. The resistance change is converted to voltage by connecting the element in a voltage divider between $AV_{CC}$ and AGND, as shown in Figure 62. The translation of voltage to temperature is determined by the parameters of the NTC/PTC thermistor in use.

## 4.0 Embedded Controller Modules (Continued)

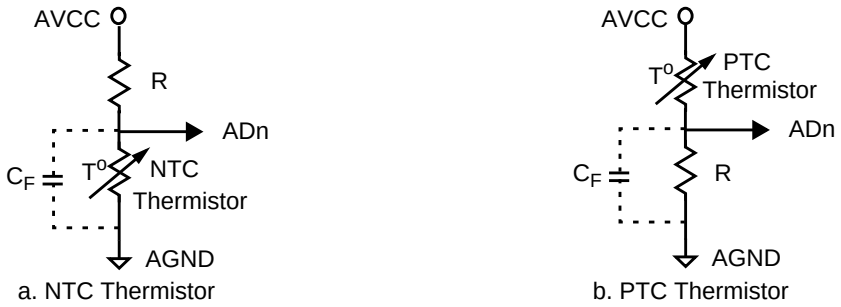



**Figure 62. Measuring Temperature Using Thermistors**

# 4.0 Embedded Controller Modules (Continued)

## 4.12 DIGITAL TO ANALOG CONVERTER (DAC)

The DAC converts digital input values to analog signals. The DAC support four channels for handling up to four independent conversions in parallel.

### 4.12.1 Features

- 8-bit resolution
- Independent 4-channel D/A converter
- Fast settling time, 1 µs typical, on 50 pF capacitive load
- Output range from AGND to $AV_{CC}$
- Independent enable/disable for each channel
- All converters can be automatically disabled in Idle mode
- Low power consumption when enabled; zero power consumption when disabled
- Outputs drive 0V when disabled

### 4.12.2 Functional Description

The DAC has four independent digital-to-analog converters. Each converter generates an output in the range of 0V to $AV_{CC}$, with 8-bit resolution. The converters drive the four output pins DA0-3, as shown in Figure 64. An output impedance of 3 KΩ allows a settling time of about 1 µs on a 50 pF load.

When a DAC channel is enabled, its output is defined by the value written to its DACDATn register. DACDAT0-3 control DA0-3, respectively. The maximum output voltage is (255 ÷ 256) * $AV_{CC}$ and is obtained for a value of $FF_{16}$. The minimum output, 0V, is obtained for a value of $00_{16}$.

The reference voltage of the converters is the $AV_{CC}$ analog power supply voltage. This allows full swing of the outputs from 0V to nearly $AV_{CC}$.

After reset, all four channels are disabled and the voltage on the DA0-3 outputs is 0V.

In Idle mode, the DAC channels may be enabled or disabled (to reduce power consumption). Two control modes are provided:

- Automatic disable of all channels on entering Idle mode
- Selective disable of channels by software, before entering Idle mode

### 4.12.3 D/A Conversion

#### Output Signal

The DAC performs a linear conversion of the input digital value DACDATA7-0 registers to a unipolar analog output signal, relative to the analog ground pin (AGND).

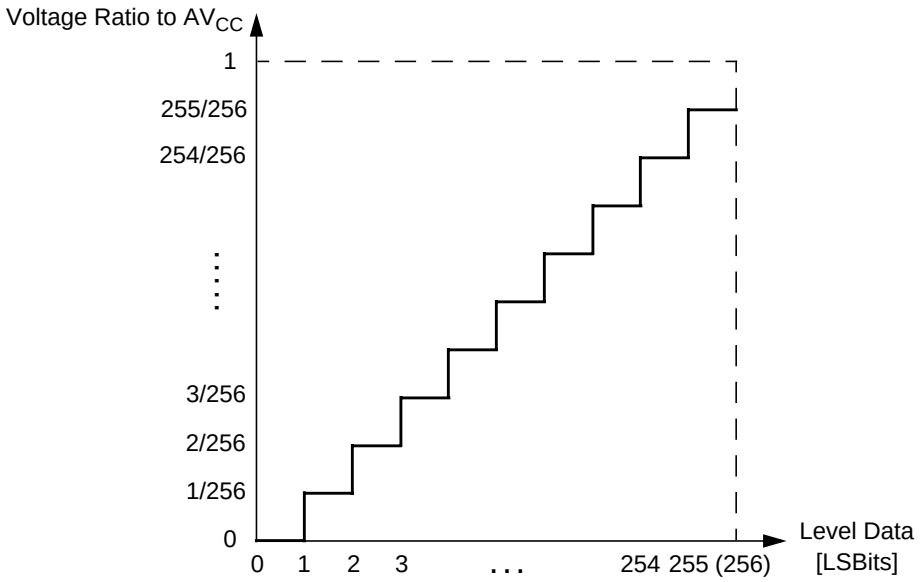



**Figure 63. Channel Data to Output Voltage Ratio Conversion**

## 4.0 Embedded Controller Modules (Continued)

When the value of DACDATA7-0 is $00_{16}$, the respective output has an output signal of 0V (AGND). When the value of DACDATn register is $FF_{16}$, the respective output has an output signal of (255/256) * $AV_{CC}$. For other values, as shown in Figure 63:

$$V_{OUT} = (DACDATAn) * (AV_{CC} / 256)$$

### Reference Voltage

The analog output voltages are converted relative to a reference voltage. The reference voltage of the converters is the analog power supply. To assure good signal quality at the PC87591L-N05 output, use a low-noise analog power supply.

### Conversion Time

When a DAC channel is enabled, the conversion is started by writing to DACDATn registers. The output settling time is defined as the time the DAC requires to get to within 1/2 LSB of the final value; see “Output Settling Time” on page 183.

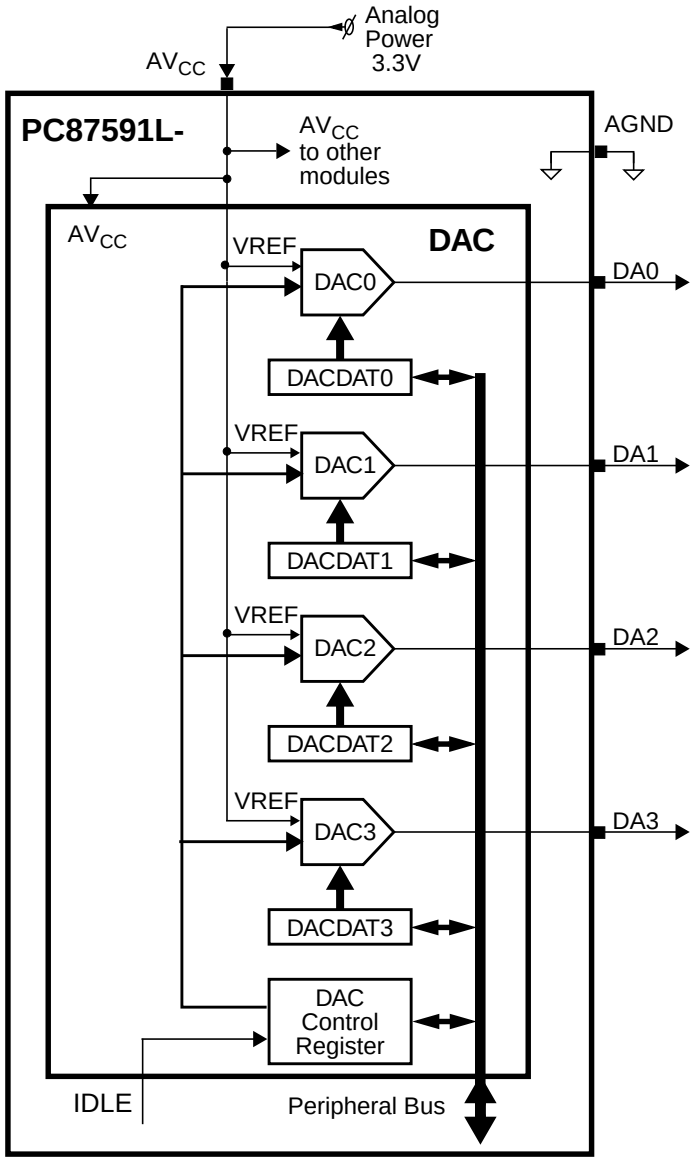



**Figure 64. DAC Functional Diagram**

# 4.0 Embedded Controller Modules (Continued)

## 4.12.4 Operation

### Initializing the DAC

The PC87591L-N05 wakes up after power-up with all the D/A channels disabled (DACEN0-3 bits in DACCTRL register are cleared to 0). In this state, all DAC activities are halted, and its current consumption is reduced to zero.

DACDATn registers (n=0 to 3) must be initialized to $00_{16}$, or according to the required output level, before setting DACEN0-3 in DACCTRL register to 1.

### Enabling and Disabling the DAC

**Enabling the DAC.** Each channel of the DAC is enabled independently by setting its DACEN bit. After enabling, it settles to the value stored in DACDATn register after the specified settling time.

**Disabling the DAC.** The DAC channels may be independently disabled in order to reduce current consumption by clearing the corresponding DACENn (n=0 to 3) bit in DACCTRL register. In this case, the output pin drives 0V, even if the respective DACDATn register does not contain $00_{16}$.

All DAC channels are automatically disabled when entering Idle mode if ENIDLE bit in DACCTRL register is cleared to 0. This happens regardless of the state of DACENn (n=0 to 3) bit in DACCTRL register. In this case, the DA0-3 outputs drive 0V.

If the ENIDLE bit is set to 1, entering the Idle mode does not affect DAC operation, and DA0-3 outputs drive the voltage level set by DACDATn (n=0 to 3) registers.

## 4.12.5 DAC Registers

The DAC interfaces with the core using one control and four data registers. These registers are mapped to the core address space, as defined in Appendix A on page 367.

### DAC Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| DACCTRL | DAC Control | R/W |
| DACDAT0 | DAC Data Channel 0 | R/W |
| DACDAT1 | DAC Data Channel 1 | R/W |
| DACDAT2 | DAC Data Channel 2 | R/W |
| DACDAT3 | DAC Data Channel 3 | R/W |

# 4.0 Embedded Controller Modules (Continued)

## DAC Control Register (DACCTRL)

This register controls the operation of the DAC module. DACCTRL is cleared ($00_{16}$) on reset.

Location: 00 $FF40_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | **ENIDLE** | **DACEN3** | **DACEN2** | **DACEN1** | **DACEN0** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **DACEN0 (DAC Channel 0 Enable).** Enables the DAC channel. The DA0 output pin drives a voltage level, according to the value written into the corresponding DACDAT0 register.<br>When cleared, the DA0 output pin drives 0V.<br>0: Disabled (default)<br>1: Enabled |
| 1 | **DACEN1 (DAC Channel 1 Enable).** Same as DACEN0 bit description, using DA1 output and DACDAT1 register. |
| 2 | **DACEN2 (DAC Channel 2 Enable).** Same as DACEN0 bit description, using DA2 output and DACDAT2 register. |
| 3 | **DACEN3 (DAC Channel 3 Enable).** Same as DACEN0 bit description, using DA3 output and DACDAT3 register. |
| 4 | **ENIDLE (Enable in Idle).** Controls the DAn (n=0 to 3) outputs in Idle mode.<br>0: Disabled - DAn outputs drive 0V (default)<br>1: Enabled - DAn outputs according to DACENn bits and DACDATn registers |
| 7-5 | **Reserved.** |

## DAC Data Channel 0-3 Registers (DACDAT0-3)

These registers hold the data to be loaded into Channels 0-3 of the DAC. These registers are not affected by reset or disable of the respective channel.

Location: Channel 0 - 00 $FF42_{16}$
Channel 1 - 00 $FF44_{16}$
Channel 2 - 00 $FF46_{16}$
Channel 3 - 00 $FF48_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **DAC DATAi** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **DAC Data.** 8-bit unsigned binary value used for the D/A operation. |

# 4.0 Embedded Controller Modules (Continued)

## 4.12.6 Usage Hints

### Power Consumption

When a channel is enabled and no load is connected, the DAC current consumption depends on the value set in DACDAT register. Minimal current is consumed when the data is $00_{16}$; maximum current is consumed when the data is $55_{16}$.

### DAC Output Protection

To maintain the high performance of the analog circuits, the DAn (n=0 to 3) pins are not back-drive protected. Therefore, the voltage applied to these pins must be within the AGND to $AV_{CC}$ range; otherwise the device may be damaged.

External circuits should not drive currents into these pins when the PC87591L-N05 power is off because this may cause the internal Power-Up reset circuit to fail.

### Output Voltage Accuracy

Besides the intrinsic accuracy of the D/A channels, the output voltage accuracy directly depends on the accuracy of the $AV_{CC}$ power supply, which serves as reference voltage. In order to improve the accuracy of the output voltage, the actual $AV_{CC}$ value, measured by the ADC module (see Section 4.11 on page 166), should be used when computing the value of DACDATA7-0 in DACDATn registers (see Section 4.12.3 on page 179).

The external load on DA0-3 pins may also affect the final output voltage of the DAC. Since the output resistance of these pins is typically 3 KΩ, use high-impedance loads; if high accuracy or high output currents are required, use external analog drivers.

For the worst case calculation, if the output resistance is 4 KΩ (maximum limit), the external load ($R_L$) must not be lower than 2 MΩ (see Figure 65). In this case, the error caused by the load is lower than 1/2 LSB.

To work with loads of 3 KΩ (1 mA at 3V) with an error lower than 1/2 LSB, the output resistance of the external driver should be lower than:

$$R_{0EXT} < 3\ K\Omega\ /\ (2*256) = 5.8\Omega$$

### Output Settling Time

The DAC output settling time depends on the external load characteristics and the required accuracy. Figure 65 shows the equivalent circuit used for evaluating DAC behavior. Each DAC output has a typical output impedance of 3 KΩ. For example, if the total load is a 50 pF capacitor only, the output settles to 1/2 LSB within 1 µs. The total load capacitance is comprises the analog output capacitance ($C_{AO}$) and the external load capacitance ($C_L$).

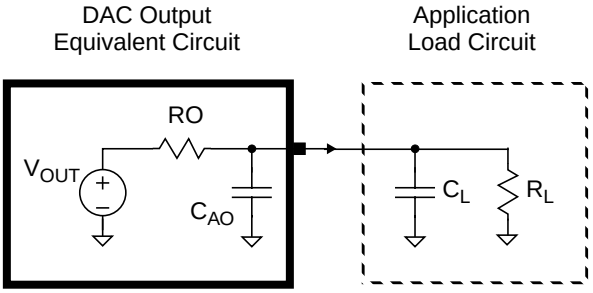



**Figure 65. DAC Output Equivalent Circuit**

### Filtering Noise on Output Signals

Output signals may have unwanted noise caused by nearby digital circuits. When using slow changing signals in a noisy environment, a low-pass filter (LPF) may be added externally. This may also be required in applications where the DAC outputs control sensitive circuits like audio amplifiers. This can be implemented as a simple RC circuit. The cutoff frequency of this LPF should be above the required signal frequency.

# 4.0 Embedded Controller Modules (Continued)

## 4.13 ACCESS.BUS (ACB) INTERFACE

The PC87591L-N05 includes four SMBus/ACCESS.bus Interface (ACB) modules. The registers of each module are prefixed with ACBn, and the signal names are suffixed with 'n', where 'n' is module number 1, 2, 3 or 4.

Each ACCESS.bus interface is a two-wire serial interface that is compatible with the ACCESS.bus physical layer. It is also compatible with Intel's SMBus and Philips' $I^2C$ bus. The module can be configured as either a bus master or slave and can maintain bidirectional communication with multiple master and slave devices. As a slave device, the ACB module may issue a request to become the bus master.

The ACB interface provides full support for a two-wire ACCESS.bus synchronous serial interface. It permits easy interfacing to a wide range of low-cost memories and I/O devices, including EEPROMs, SRAMs, timers, A/D converters, D/A converters, clock chips and peripheral drivers.

### 4.13.1 Features

- ACCESS.bus, SMBus and $I^2C$ compliant
- ACCESS.bus master
- ACCESS.bus slave
  - — One or two user-defined addresses
  - — Global (broadcast) address
  - — ARP address
- Supports polling- interrupt- and DMA-controlled (n=3 and 4 only) operation
- Generates a wake-up signal on detection of a Start Condition in Power-Down mode
- Optional internal pull-up on SDAn and SCLn pins

### 4.13.2 Functional Description

The ACCESS.bus protocol uses a two-wire interface for bidirectional communication between the ICs connected to the bus. The two interface lines are the Serial Data Line (SDLn) and the Serial Clock Line (SCLn). These lines should be connected to a positive supply via a pull-up resistor and remain high even when the bus is idle.

The ACCESS.bus protocol supports multiple master and slave transmitters and receivers. Each IC has a unique address and can operate as a transmitter or a receiver. Some peripherals are receivers only.

During data transactions, the master device initiates the transaction, generates the clock signal and terminates the transaction. For example, when the ACB initiates a data transaction with an attached ACCESS.bus-compliant peripheral, the ACB becomes the master. When the peripheral responds and transmits data to the ACB, their master/slave (data transaction initiator and clock generator) relationship is unchanged even though their transmitter/receiver functions are reversed.

#### Data Transactions

One data bit is transferred during each clock pulse. Data is sampled during the high state of the serial clock (SCLn). Consequently, throughout the clock's high period, the data should remain stable (see Figure 66). Any change on the SDAn line while SCLn is in high state during a transaction causes the current transaction to abort. New data should be sent during the low SCLn state. This protocol permits a single data line to transfer both command/control information and data, using the synchronous serial clock.

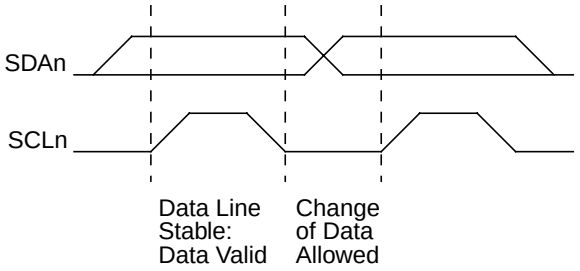



**Figure 66. Bit Transfer**

Each data transaction is composed of a Start Condition, a number of byte transfers (set by the software) and a Stop Condition to terminate the transaction. Each byte is transferred with the most significant bit first. An Acknowledge signal must follow each byte (8 bits). The following sections provide further details of this process.

At each clock cycle, the slave can stall the master while it handles the previous data or prepares new data. The slave does this, for each bit transferred or on a byte boundary, by holding SCLn low to extend the clock low period. Typically, slaves extend the first clock cycle of a transfer if a byte read has not yet been stored or if the next byte to be transmitted is not yet ready. Some microcontrollers with limited hardware support for the ACCESS.bus extend the access after each bit, thus allowing the software time to handle this bit.

## 4.0 Embedded Controller Modules (Continued)

The ACCESS.bus master generates Start and Stop Conditions (control codes). After a Start Condition is generated, the bus is considered busy. It retains this status for a given amount of time after a Stop Condition is generated. A high-to-low transition of the data line (SDAn) while the clock (SCLn) is high indicates a Start Condition. A low-to-high transition of the SDAn line while the SCLn is high indicates a Stop Condition (Figure 67).

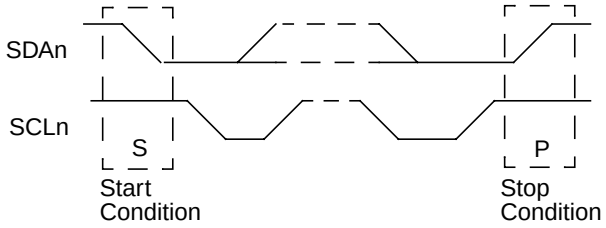



**Figure 67. Start and Stop Conditions**

In addition to the first Start Condition, a Repeated Start Condition can be generated in the middle of a transaction. This allows either another device to be accessed or a change in the direction of the data transfer.

### Acknowledge Cycle

The Acknowledge cycle consists of two signals:

- Acknowledge Clock pulse is sent by the master with each byte transferred
- Acknowledge signal is sent by the receiving device (see Figure 68)

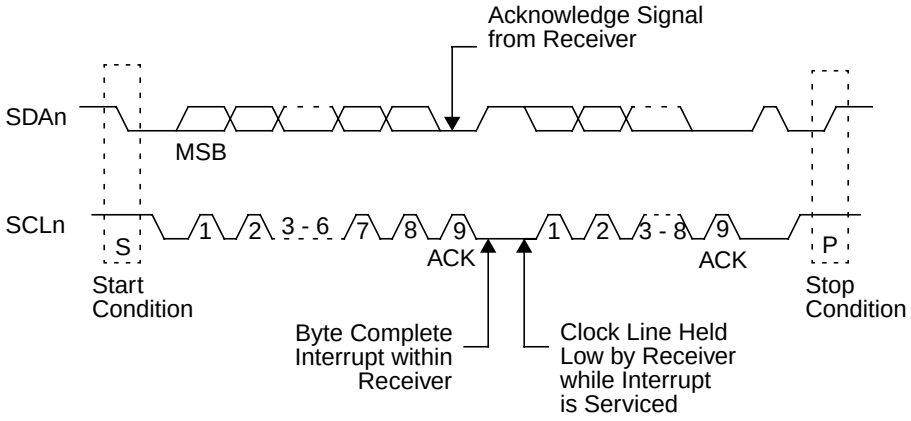



**Figure 68. ACCESS.bus Data Transaction**

The master generates the Acknowledge clock pulse on the ninth clock pulse of the byte transfer. The transmitter releases the SDAn line (permitting it to go high) to allow the receiver to send the Acknowledge signal.The receiver pulls down the SDAn line during the Acknowledge clock pulse, thus signalling that it has correctly received the last data byte and is ready to receive the next byte. Figure 69 shows the Acknowledge cycle.

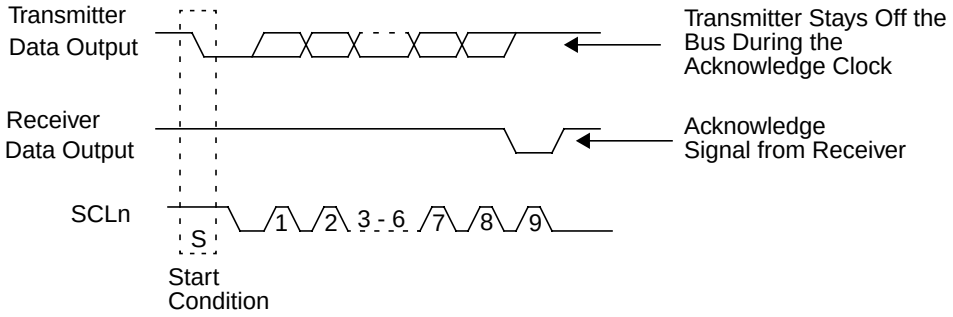



**Figure 69. ACCESS.bus Acknowledge Cycle**

### "Acknowledge After Every Byte" Rule

The master generates an Acknowledge clock pulse after each byte transfer. The receiver sends an Acknowledge signal after every byte is received.

## 4.0 Embedded Controller Modules (Continued)

There are two exceptions to the “acknowledge after every byte” rule:

- When the master is the receiver, it must indicate to the transmitter an end of data by not acknowledging (negative acknowledge) the last byte clocked out of the slave. This negative acknowledge still includes the Acknowledge clock pulse (generated by the master), but the SDAn line is not pulled down.
- When the receiver is full or otherwise occupied, or if a problem occurs, it sends a negative acknowledge to indicate that it cannot accept additional data bytes.

**Addressing Transfer Formats**

Each device on the bus has a unique address. Before any data is transmitted, the master transmits the address of the slave being addressed. The slave device should send an Acknowledge signal on the SDAn line once it recognizes its address.

The address consists of the first seven bits after a Start Condition. The eighth bit contains the direction of the data transfer (R/$\overline{W}$). A low-to-high transition during a SCLn high period indicates the Stop Condition and ends the transaction of SDAn (Figure 70).

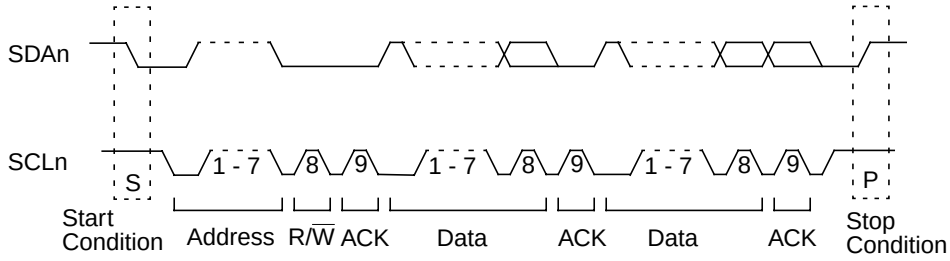



**Figure 70. A Complete ACCESS.bus Data Transaction**

When the address is sent, each device in the system compares this address with its own. If there is a match, the device considers itself addressed and sends an Acknowledge signal. Depending on the state of the R/$\overline{W}$ bit (1=read, 0=write), the device acts as a transmitter or a receiver.

The $I^2C$ bus protocol allows a general call address to be sent to all slaves connected to the bus. The first byte sent specifies the general call address ($00_{16}$); the second byte specifies the general call meaning (for example, “Write slave address by software only”). Slaves that require data acknowledge the call and become slave receivers; other slaves ignore the call.

**Arbitration on the Bus**

Multiple master devices on the bus require arbitration between their conflicting bus access demands. Control of the bus is initially determined according to address bits and clock cycle. If more than one master tries to address the same slave, data comparisons determine the outcome of this arbitration. In Master mode, the device immediately aborts a transaction if the value sampled on the SDAn line differs from the value driven by the device. (An exception to this rule is SDAn while receiving data; in this case, the lines may be driven low by the slave without causing an abort.)

The SCLn signal is monitored for clock synchronization to allow the slave to stall the bus. The actual clock period is the longest one set by the master or the slave stall period. The clock high period is determined by the master with the shortest clock high period.

When an abort occurs during address transmission, a master that identifies the conflict should give up the bus and switch to Slave mode. It should then continue to sample SDAn to see if it is being addressed by the winning master on the bus.

### 4.13.3 Master Mode

**Requesting Bus Mastership**

An ACCESS.bus transaction starts with a master device requesting bus mastership. It asserts a Start Condition, followed by the address of the device it wants to access. If this transaction is successfully completed, the software may assume that the device has become the bus master.

For the device to become the bus master, the software should perform the following steps:

1. Configure INTEN in ACBnCTL1 register to the desired operation mode (Polling or Interrupt) and set START in the same register. This causes the ACB to issue a Start Condition on the ACCESS.bus as soon as the ACCESS.bus is free (some conditions, such as when BB in ACBnCST register is set to 0, can delay start). It then stalls the bus by holding SCLn low.
2. If a bus conflict is detected (i.e., some other device pulls down the SCLn signal before the PC87591L-N05 does), BER in ACBnST register is set.
3. If there is no bus conflict, MASTER and SDAST in ACBnST register are set.
4. If INTEN in ACBnCTL1 register is set and either BER or SDAST in ACBnST register is set, an interrupt is sent to the core.

## 4.0 Embedded Controller Modules (Continued)

### Sending the Address Byte

Once the PC87591L-N05 is the active master of the ACCESS.bus (MASTER in ACBnST register is set), it can send the address on the bus. The address sent should **not** be any of the following:

- The PC87591L-N05's own address, as defined by ADDR in ACBnADDR register, if SAEN in ACBnADDR is set.
- The PC87591L-N05's own address, as defined by ADDR in ACBnADDR2, if SAEN in ACBnADDR2 is set.
- The global call address, if GCMATCH in ACBnCST register is set.
- The ARP address, if ARPMATCH in ACBnST register is set.

To send the address byte, use the following sequence:

1. For a receive transaction where the software requires only one byte of data, the software should set ACK in ACBnCTL1 register. If only an address needs to be sent (e.g., for quick read/write protocols) or if the device requires stall for some other reason, set STASTRE in ACBnCTL1 register to 1.
2. Write the address byte (7-bit target device address) and the direction bit to ACBnSDA register. This causes the module to generate a transaction. At the end of this transaction, the acknowledge bit received is copied to NEGACK in ACBnST register. During the transaction, the SDAn and SCLn lines are continuously checked for conflict with other devices. If a conflict is detected, the transaction is aborted, BER in ACBnST register is set and MASTER in ACBnST register is cleared.
3. If STASTRE in ACBnCTL1 register is set and the transaction was successfully completed (i.e., both BER and NEGACK in ACBnST register are cleared), STASTR in ACBnST register is set. In this case, the ACB stalls any further ACCESS.bus operations (i.e., holds SCLn low). If INTEN in ACBnCTL1 register is set, it also sends an interrupt to the core.
4. If the requested direction is transmit and the start transaction was completed successfully (i.e., neither NEGACK nor BER in ACBnST register is set and no other master has accessed the device), SDAST in ACBnST register is set to indicate that the module awaits attention.
5. If the requested direction is receive, the start transaction was completed successfully and STASTRE in ACBnCTL1 register is cleared, the module starts receiving the first byte automatically.
6. Check that both BER and NEGACK in ACBnST register are cleared. If either INTEN in ACBnCTL1 register or DMAEN in the ACBnCTL1 register is set, an interrupt is generated when either BER or NEGACK is set.

### Master Transmit

After becoming the bus master, the device can start transmitting data on the ACCESS.bus.

In interrupt or polling operation, to transmit a byte, the software should:

1. Check that BER and NEGACK bits in ACBnST register are cleared and SDAST bit is set. In addition, if STASTRE bit in ACBnCTL1 register is set, make sure that STASTR bit in ACBnST register is cleared.
2. Write the data byte to be transmitted to ACBnSDA register.

In DMA operation:

- If DMAEN in the ACBnCTL1 register was set before the start transaction, a DMA request is generated automatically at the end of the address transaction and after each following transaction, unless for some reason (e.g., ACBnCST, MATCH or BER were set) an interrupt was generated.

When NEGACK or BER in the ACBnST register is set, an interrupt is generated and the ACB stops sending DMA requests.When the slave responds with a negative acknowledge, NEGACK in ACBnST register is set and SDAST in ACBnST register remains cleared. In this case, if INTEN bit in ACBnCTL1 register or DMAEN bit in ACBCTL1 register is set, an interrupt is sent to the core.

### Master Receive

After becoming the bus master, the device can start receiving data on the ACCESS.bus.

In interrupt or polling operation, to receive a byte, the software should:

1. Check that SDAST bit in ACBnST register is set and BER bit is cleared. In addition, if STASTRE bit in ACBnCTL1 register is set, make sure that STASTR in ACBnST register is cleared.
2. If the next byte is the last byte that should be read, set ACK bit in ACBnCTL1 register to 1. This causes a negative acknowledge to be sent.
3. Read the data byte from ACBnSDA register.

In DMA operation:

- The DMA request becomes active after the module receives a byte of data. If an error occurs during the transaction (e.g., NMATCH in the ACBnCST register or BER in the ACBnST register is set), an interrupt is generated and DMA operation is stalled.

Before receiving the last byte of data, set ACK in the ACBnCTL1 register. This should be done by programing the DMA to interrupt the CPU one byte before the end of the transmission, and letting the software set ACK.

## 4.0 Embedded Controller Modules (Continued)

### Master Stop

To end a transaction, set STOP in ACBnCTL1 register before clearing the current stall flag (i.e., SDAST, NEGACK or STASTR in ACBnST register). This causes the module to send a Stop Condition immediately and to clear STOP in ACBnCTL1 register. A Stop Condition may be issued only when the PC87591L-N05 is the active bus master (MASTER in ACBnST register is set to 1).

### Master Bus Stall

The ACB module can stall the ACCESS.bus between transfers while waiting for the core's or DMA's response. The ACCESS.bus is stalled by holding the SCLn signal low after the acknowledge cycle. Note that this is interpreted as the start of the following bus operation. The user must make sure that the next operation is prepared before the flag that causes the bus stall is cleared.

The flags that can cause a bus stall in Master mode are:

- Negative acknowledge after sending a byte (NEGACK in ACBnST register is set to 1).
- SDAST in ACBnST register is set to 1.
- STASTRE in ACBnCTL1 register is set to 1 after a successful start (STASTR in ACBnST is set to 1).

### Repeated Start

A repeated start is performed when the PC87591L-N05 is already the bus master (MASTER in ACBnST register is set). In this case, the ACCESS.bus is stalled and the ACB module awaits core handling due to a negative acknowledge (NEGACK in ACBnST register is set to 1), an empty buffer (SDAST in ACBnST is set to 1) and/or a stall after start (STASTR in ACBnST is set to 1).

For a repeated start:

1. Set START in ACBnCTL1 register to 1.
2. In Master Receive mode, read the last data item from ACBnSDA.
3. Follow the address send sequence, as described in "Sending the Address Byte" on page 187.
4. If the ACB is awaiting handling because STASTR in ACBnST is set to 1, clear it only after writing the requested address and direction to ACBnSDA.

### Master Error Detection

The ACB detects an illegal Start or Stop Condition (i.e., a Start or Stop Condition within the data transfer or the acknowledge cycle) and a conflict on the data lines of the ACCESS.bus. If an illegal condition is detected, BER is set and Master mode is exited (MASTER in ACBnST register is cleared).

### Bus Idle Error Recovery

When a request to become the active bus master or a restart operation fails, BER in ACBnST register is set to indicate the error. In some cases, both the PC87591L-N05 and the other device may identify the failure and leave the bus idle. In this case, the start sequence may not finish and the ACCESS.bus may remain deadlocked.

To recover from deadlock, use the following sequence:

1. Clear BER in ACBnST register and BB in ACBnCST register.
2. Wait for a time-out period to check that there is no other active master on the bus (i.e., BB in ACBnCST remains cleared).
3. Disable and then re-enable the ACB to put it in non-addressed Slave mode. (This completely resets the module.)

At this point, some of the slaves may not identify the bus error. To recover, the ACB module becomes the bus master. It asserts a Start Condition, sends an address byte and then asserts a Stop Condition that synchronizes all the slaves.

### 4.13.4 Slave Mode

A slave device waits in Idle mode for a master to initiate a bus transaction. Whenever the ACB module is enabled is not acting as a master (i.e., MASTER in ACBnST register is cleared), it acts as a slave device.

Once a Start Condition on the bus is detected, the PC87591L-N05 checks whether the address sent by the current master matches any of the following possibilities:

- The ADDR value in ACBnADDR register, if SAEN in this register is set to 1
- The ADDR value in ACBnADDR2 register, if SAEN in this register is set to 1
- The global call address ($00_{16}$), if GCMEM in ACBnCTL1 register is set to 1
- The global ARP address ($110\ 0001_2$), if ARPMEN in ACBnCTL3 register is set to 1.

The address match is checked even when MASTER in ACBnST register is set. If a bus conflict (on SDAn or SCLn) is detected, BER is set, MASTER is cleared and the PC87591L-N05 continues to search the received message for a match.

If an address ARP or global match is detected:

1. The PC87591L-N05 asserts its SDAn pin during the acknowledge cycle.
2. MATCH in ACBnCST register, MATCHAF in ACBnST register (or GCMATCH if it is a global call address match, or ARP-MATCH if it is an ARP address) and NMATCH in ACBnST register are set. If XMIT in ACBnST register is set (i.e., Slave Transmit mode), SDAST in the same register is also set to indicate that the buffer is empty.
3. If INTEN in ACBnCTL1 register is set, an interrupt is generated if both INTEN and NMINTE in ACBnCTL1 register are set.
4. The software then reads XMIT in ACBnST register to identify the direction requested by the master device; it then clears NMATCH in the same register so that future byte transfers are identified as data bytes.

**Slave Receive and Transmit**

Slave Receive and Transmit are performed after a match is detected and the data transfer direction is identified. After a byte transfer, the ACB module extends the acknowledge clock until the software reads or writes ACBnSDA register. The receive and transmit sequences are identical to those used in the master routine.

**Slave Bus Stall**

When operating as a slave, the PC87591L-N05 stalls the ACCESS.bus by extending the first clock cycle of a transaction in the following cases:

- SDAST in ACBnST register is set.
- NMATCH in ACBnST register and NMINTE in ACBnCTL1 register are set.

**Slave Error Detection**

The ACB detects illegal Start and Stop Conditions (occurring within the data transfer or the acknowledge cycle) on the ACCESS.bus. When an illegal Start or Stop Condition is detected, BER is set and MATCH and GMATCH are cleared, setting the module as an unaddressed slave.

### 4.13.5 Power-Down

When the PC87591L-N05 is in Idle mode, the ACB module is not active, but retains its registers. An exception is the ACBnCTL1 register, which is reset in Idle mode. If the ACB is enabled (ENABLE in ACBnCTL2 register is set) on detection of a Start Condition, a wake-up signal is issued to the MIWU. This signal may be used to switch the PC87591L-N05 to Active mode.

Following the Start Condition that woke up the PC87591L-N05, the ACB module can not check the address byte for a match. The ACB responds with a negative acknowledge. The device should resend both the Start Condition and the address after the PC87591L-N05 has had time to wake up.

Before entering Idle mode, make sure that BUSY in ACBnCST register is inactive. This guarantees that the PC87591L-N05 will not stop responding after it acknowledges an address that was sent.

### 4.13.6 SDA and SCL Pin Configuration

The SDAn and SCLn are open collector signals that the user can choose to enable or disable. SDAn and SCLn also have internal pull-up resistors that the user may enable. For more information about configuring these pins, see Table 6 on page 49 and Section 4.5.2 on page 111.

### 4.13.7 ACB Clock Frequency Configuration

The ACB module enables the user to set the ACCESS.bus clock frequency. The SCLn clock period is set by SCLFRQ in ACBnCTL2 and ACBnCTL3 registers. This clock low period may be extended by stall periods initiated by the ACB module or by another ACCESS.bus device. In case of a conflict with another bus master, a shorter clock high period may be forced by the other bus master until the conflict is resolved.

# 4.0 Embedded Controller Modules (Continued)

## 4.13.8 ACB Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### ACB Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| ACBnSDA | ACB Serial Data | R/W |
| ACBnST | ACB Status | Varies per bit |
| ACBnCST | ACB Control Status | Varies per bit |
| ACBnCTL1 | ACB Control 1 | R/W |
| ACBnADDR | ACB Own Address | R/W |
| ACBnADDR2 | ACB Own Address 2 | R/W |
| ACBnCTL2 | ACB Control 2 | R/W |
| ACBnCTL3 | ACB Control 3 | R/W |

### ACB Serial Data Register (ACBnSDA)

The ACBnSDA register is a shift register used to transmit and receive data. The most significant bit is transmitted (received) first and the least significant bit is transmitted (received) last. Reading or writing to the ACBnSDA register is allowed only when SDAST in ACBnST register is set or for repeated starts after setting the START bit. An attempt to access the register in other cases may produce unpredictable results.

Locations: Channel 1 - 00 $FF60_{16}$
Channel 2 - 00 $FFE0_{16}$
Channel 3 - 00 $FC40_{16}$
Channel 4 - 00 $FC60_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Data.** |

# 4.0 Embedded Controller Modules (Continued)

## ACB Status Register (ACBnST)

The ACBnST register maintains current ACB status. Some of its bits may be cleared by software, as described below. On reset, and when the module is disabled, ACBnST is cleared ($00_{16}$).

Location: Channel 1 - 00 $FF62_{16}$
Channel 2 - 00 $FFE2_{16}$
Channel 3 - 00 $FC42_{16}$
Channel 4 - 00 $FC62_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **SLVSTP** | **SDAST** | **BER** | **NEGACK** | **STASTR** | **NMATCH** | **MASTER** | **XMIT** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Type | Description |
|---|---|---|
| 0 | RO | **XMIT (Transmit Mode).**<br>0: ACB not in master/slave Transmit mode (default)<br>1: ACB in master/slave Transmit mode |
| 1 | RO | **MASTER (Master Mode).**<br>0: Arbitration loss (BER is set) or Stop Condition occurred (default)<br>1: ACB in Master mode (successful request for bus mastership) |
| 2 | R/W1C | **NMATCH (New Match).** This bit is set when the address byte following a Start Condition or a repeated start causes an address match, ARP address match or a global call match. NMATCH is cleared by writing 1 to it. Writing 0 to NMATCH is ignored. If INTEN in ACBnCTL1 register is set, an interrupt is sent when this bit is set. |
| 3 | R/W1C | **STASTR (Stall After Start).** This bit is set by the successful completion of sending an address (i.e., a Start Condition sent without a bus error or negative acknowledge), if STASTRE in ACBnCTL1 register is set. This bit is ignored in Slave mode. When STASTR is set, it stalls the ACCESS.bus (by pulling down the SCL line) and suspends any further action on the bus (e.g., receiving the first byte in Master Receive mode). In addition, if INTEN in ACBnCTL1 register is set, it also causes the ACB module to send an interrupt to the core. Writing 1 to STASTR clears it. It is also cleared when the module is disabled and is always cleared when STASTRE is cleared. Writing 0 to STASTR has no effect. |
| 4 | R/W1C | **NEGACK (Negative Acknowledge).** This bit is set by hardware when a transmission is not acknowledged on the ninth clock (in this case, SDAST is not set). Writing 1 to NEGACK clears it. It is also cleared when the module is disabled. Writing 0 to NEGACK is ignored. |
| 5 | R/W1C | **BER (Bus Error).** This bit is set by the hardware when a Start or Stop Condition is detected during data transfer (i.e., Start or Stop Condition during the transfer of bits 2 through 8 and acknowledge cycle) or when an arbitration problem is detected. Writing 1 to BER clears it. It is also cleared when the module is disabled. Writing 0 to BER is ignored. |
| 6 | RO | **SDAST (SDA Status).** When set, this bit indicates that the SDA data register is waiting for data (Transmit mode - master or slave) or holds data that should be read (Receive mode - master or slave). This bit is cleared when reading from ACBnSDA register during a receive or when written to during a transmit. When START in the ACBnCTL1 is set, reading ACBnSDA register does not clear SDAST. This enables the ACB to send a repeated start in Master Receive mode. |
| 7 | R/W1C | **SLVSTP (Slave Stop).** When set, this bit indicates that a Stop Condition was detected after a slave transfer (i.e., after a slave transfer in which MATCH, ARPMATCH or GCMATCH was set). Writing 1 to SLVSTP clears it. It is also cleared when the module is disabled. Writing 0 to SLVSTP is ignored. |

# 4.0 Embedded Controller Modules (Continued)

## ACB Control Status Register (ACBnCST)

The ACBnCST register maintains current ACB status and controls several ACB module functions, as described below. On reset and when the module is disabled, the non-reserved bits of ACBnCST are cleared ($00_{16}$). An exception is the TSDA bit, which reflects the current value of the SDA pin.

Location: Channel 1 - 00 $FF64_{16}$
Channel 2 - 00 $FFE4_{16}$
Channel 3 - 00 $FC44_{16}$
Channel 4 - 00 $FC64_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **ARPMATCH** | **MATCHAF** | **TGSCL** | **TSDA** | **GCMATCH** | **MATCH** | **BB** | **BUSY** |
| Reset | 0 | 0 | 0 | X[1] | 0 | 0 | 0 | 0 |

1. According to the current value of the SDA pin.

| Bit | Type | Description |
|---|---|---|
| 0 | RO | **BUSY.** When set (1), indicates that the ACB module is in one of the following states:<br>– Generating a Start Condition<br>– In Master mode (MASTER in ACBnST register is set)<br>– In Slave mode (MATCH or GMATCH in ACBnCST register is set)<br>– In the period between detecting a Start Condition and completing the reception of the address byte; after this, the ACB either becomes not busy or enters Slave mode.<br>The BUSY bit is cleared by the completion of any of the above states or by disabling the module. It should always be written 0. |
| 1 | R/W1C | **BB (Bus Busy).** When set (1), indicates the bus is busy. It is set either when the bus is active (i.e., a low level on either SDA or SCL) or by a Start Condition. It is cleared when the module is disabled, on detection of a Stop Condition or by writing 1 to this bit. See Section 4.13.9 on page 196 for a description of the use of this bit. |
| 2 | RO | **MATCH (Address Match).** In Slave mode, MATCH is set (1) when SAEN in ACBnADDR register is set and the first seven bits of the address byte (the first byte transferred after a Start Condition) match the 7-bit address in ACBnADDR register. It is cleared by Start Condition, a Repeated Start or a Stop Condition (including illegal Start or Stop Condition). |
| 3 | RO | **GCMATCH (Global Call Match).** In Slave mode, GCMTCH is set (1) when GCMEN in ACBnCTL1 register is set and the address byte (the first byte transferred after a Start Condition) is $00_{16}$. It is cleared by a Start Condition, a Repeated Start or a Stop Condition (including illegal Start or Stop Condition). |
| 4 | RO | **TSDA (Test SDA Line).** Reads the current value of the SDA line. This bit can be used while recovering from an error condition in which the SDA line is constantly pulled low by a slave that went out of synch. Data written to this bit is ignored. |
| 5 | R/W | **TGSCL (Toggle SCL Line).** Enables toggling the SCL line during the process of error recovery. When the SDA line is low, writing 1 to this bit toggles the SCL line for one cycle. Writing 1 to TGSCL is ignored if any of the following conditions is true:<br>• The SDA line is high<br>• The ACB module is in Slave mode and a transaction is performed on the bus.<br>TGSCL bit is cleared when the SCL line toggle is completed. |
| 6 | RO | **MATCHAF (Match Address Field).** When the MATCH bit is set, MATCHAF indicates with which of the two possible slave addresses (ADDR cleared in ACBnADDR register or set in ACBnADDR2 register) the match has occurred. If both addresses match, the bit is cleared. |
| 7 | RO | **ARPMATCH (ARP address Match).** In Slave mode, ARPMTCH is set (1) when ARPMEN in ACBnCTL3 register is set and the address byte (the first byte transferred after a Start Condition) is $110\ 0001_2$. It is cleared by Start Condition, a Repeated Start or a Stop Condition (including illegal Start or Stop Condition). |

# 4.0 Embedded Controller Modules (Continued)

## ACB Control Register 1 (ACBnCTL1)

The ACBnCTL1 register is a byte-wide, read/write register that configures and controls the ACB module. On reset, the ACBnCTL1 is cleared ($00_{16}$).

Location: Channel 1 - 00 $FF66_{16}$
Channel 2 - 00 $FFE6_{16}$
Channel 3 - 00 $FC46_{16}$
Channel 4 - 00 $FC66_{16}$

Type: R/W

Channel 1 and Channel 2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **STASTRE** | **NMINTE** | **GCMEN** | **ACK** | **Reserved** | **INTEN** | **STOP** | **START** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Channel 3 and Channel 4

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **STASTRE** | **NMINTE** | **GCMEN** | **ACK** | **DMAEN** | **INTEN** | **STOP** | **START** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **START.** Should be set when a Start Condition must be generated on the ACCESS.bus.<br>– If the PC87591L-N05 is not the active bus master (MASTER in ACBnST register is set to 0), setting START generates a Start Condition as soon as the ACCESS.bus is free (BB in ACBnCST register is set to 0). An address transmission sequence should then be performed.<br>– If the PC87591L-N05 is the active master of the bus (MASTER in ACBnST register is set to 1), when START is set, a write to ACBnSDA register generates a Start Condition. ACBnSDA data is then transmitted as the slave's address and the requested transfer direction.<br>In case of a Repeated Start Condition, the set bit may be used to switch the direction of the data flow between the master and the slave or to choose another slave device without using a Stop Condition in between.<br>The START bit is cleared either when the Start Condition is sent or on detection of a Bus Error (BER in ACBnST register is set to 1).<br>This bit should be set only when in Master mode or when requesting Master mode. |
| 1 | **STOP.** In Master mode, setting this bit generates a Stop Condition, which completes or aborts the current message transfer. This bit clears itself after STOP is issued. |
| 2 | **INTEN (Interrupt Enable).** When INTEN is cleared (0), the ACB interrupt is disabled. When INTEN is set, interrupts are enabled. An interrupt is generated (the interrupt signals to the ICU are high) on one of the following events:<br>– An address match is detected (NMATCH in ACBnST register is set to1 and NMINTE in ACBnCTL1 register is set to 1).<br>– A Bus Error occurs (BER in ACBnST register is set to 1).<br>– A negative acknowledge is received after sending a byte (NEGACK in ACBnST register is set to 1).<br>– If DMA is not enabled, acknowledgment of each transaction (same as the hardware set of SDAST in ACBnST).<br>– In Master mode, if STASTRE in ACBnCTL1 register is set to 1 after a successful start (STASTR in ACBnST register is set to 1).<br>– Detection of a Stop Condition while in Slave mode (SLVSTP in ACBnST register is set to 1). |
| 3 | **DMAEN (DMA Enable - for Channel 3 and Channel 4).** When this bit is set, the DMA interface is enabled. A DMA request is generated at the end of any data transaction (set of SDAST in ACBnST). If INTEN is set, interrupts are generated on the occurrence of any error or a new match). |
| 4 | **ACK (Acknowledge).** When acting as a receiver, this bit holds the value of the next acknowledge cycle. It should be set when a negative acknowledge must be issued on the next byte. This bit is cleared (0) after the first acknowledge cycle.<br>This bit is ignored when in Transmit mode. It cannot be reset by software. |

## 4.0 Embedded Controller Modules (Continued)

| Bit | Description |
|---|---|
| 5 | **GCMEN (Global Call Match Enable).** When set, enables the matching of an incoming address byte to the general call address (Start Condition followed by address byte of $00_{16}$) while the ACB is in Slave mode. When cleared, the ACB does not respond to a global call. |
| 6 | **NMINTE (New Match Interrupt Enable).** When set, enables the interrupt on a new match (i.e., when NMATCH in ACBnST register is set). The interrupt is issued only if INTEN in ACBnCTL1 register is set. This bit must be set when using DMA for data transfer when n=3 and 4. |
| 7 | **STASTRE (Stall After Start Enable).** When set (1), enables the Stall After Start mechanism. In such a case, the ACB stalls the bus after the address byte. When STASTRE is cleared, STASTR in ACBnST is always cleared. |

### ACB Own Address Register (ACBnADDR and ACBnADDR2)

The ACBnADDR and ACBnADDR2 registers hold the module's ACCESS.bus addresses. The reset value of these registers are undefined.

ACBnADDR:

Location: Channel 1 - 00 $FF68_{16}$
Channel 2 - 00 $FFE8_{16}$
Channel 3 - 00 $FC48_{16}$
Channel 4 - 00 $FC68_{16}$

ACBnADDR2:

Location: Channel 1- 00 $FE6C_{16}$
Channel 2- 00 $FFEC_{16}$
Channel 3 - 00 $FC4C_{16}$
Channel 4 - 00 $FC6C_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **SAEN** | **ADDR** | | | | | | |

| Bit | Description |
|---|---|
| 6-0 | **ADDR (Address).** Holds the 7-bit ACCESS.bus address of the PC87591L-N05. When in Slave mode, the first seven bits received after a Start Condition are compared to this field (the first bit received is compared to bit 6, the next bit to bit 5 and so on until the last bit, which is compared to bit 0). If the address field matches the received data and SAEN in ACBnADDR register is set to 1, a match is declared. |
| 7 | **SAEN (Slave Address Enable).** When set (1), indicates that the ADDR field holds a valid address and enables the match of ADDR to an incoming address byte. When cleared, the ACB does not check for an address match. |

# 4.0 Embedded Controller Modules (Continued)

## ACB Control Register 2 (ACBnCTL2)

The ACBnCTL2 register enables/disables the module and determines the ACB clock rate. On reset and while the module is disabled (ENABLE in ACBnCTL2 register is set to 0), ACBnCTL2 is cleared ($00_{16}$).

Location: Channel 1 - 00 $FF6A_{16}$
Channel 2 - 00 $FFEA_{16}$
Channel 3 - 00 $FC4A_{16}$
Channel 4 - 00 $FC6A_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **SCLFRQ6-0** | | | | | | | **ENABLE** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **ENABLE.** When set, the ACB module is enabled. When the Enable bit is cleared, the ACB module is disabled, ACBnCTL1, ACBnST and ACBnCST are cleared and the clocks are halted. |
| 7-1 | **SCLFRQ6-0 (SCL Frequency bits 6 through 0).** This field, together with SCLFRQ8-7 in ACBCTL3 register, defines the SCL's period (low time and high time) when the PC87591L-N05 serves as a bus master. The clock low time and high time are defined as follows:<br>$t_{SCL}$ =4*SCLFRQ*$t_{CLK}$<br>$t_{SCLl}$ = $t_{SCLh}$<br>where $t_{CLK}$ is the PC87591L-N05 clock cycle when in Active mode (see Section 7.6.3 on page 346).<br><br>SCLFRQ may be programed to values in the range of 00 $0001000_2$ ($8_{10}$) through 11 $1111111_2$ ($511_{10}$). Values outside this range gives unpredictable results. |

## ACB Control Register 3 (ACBnCTL3)

The ACBnCTL3 register expands the clock pre-scaler field and enables the match to ARP addresses. ACBnCTL2 is cleared on reset ($00_{16}$).

Location: Channel 1 -00 $FF6E_{16}$
Channel 2- 00 $FFEE_{16}$
Channel 3 - 00 $FC4E_{16}$
Channel 4 - 00 $FC6E_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | **ARPMEN** | **SCLFRQ8-7** | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 1-0 | **SCLFRQ8-7 (SCL Frequency bits 8 and 7).** Extends SCLFRQ field and is concatenated with bits 0-6, which are part of ACBnCTL2 register. Detailed use of SCLFRQ is provided in the SCLFRQ6-0 description in ACBnCTL2. |
| 2 | **ARPMEN (ARP Match Enable).** When set, enables the matching of an incoming address byte to the SMBus ARP address ($110\ 0001_2$) while the ACB is in Slave mode. When cleared, the ACB does not respond to an ARP address. |
| 7-3 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## 4.13.9 Usage Hints

1. When the ACB is disabled, BB in ACBnCST register is cleared. After the ACB is enabled (by setting ENABLE in ACBnCTL2 register), the bus may be in the middle of a transaction with another device in systems with more than one master. This status is not reflected by BB.

   To prevent bus errors, the ACB must synchronize with the bus activity status before issuing a request to become the bus master for the first time. The software should check that there is no activity on the bus by checking the BB bit after the time-out period allowed by the bus.

2. When waking up from power-down before checking MATCH in ACBnCST register, check BUSY in the same register to make sure that the address transaction is completed.

3. The BB bit can help solve a deadlock in which two or more devices detect a usage conflict on the bus and both cease being bus masters at the same time. In this situation, the BB bits of both devices are active (because each “detects” another master currently performing a transaction, while in fact there is no transaction). This potentially causes the bus to stay locked until a device on the bus sends a Stop Condition (through STOP in ACBnCTL1 register).

   The BB bit allows the software to monitor bus usage so that it can detect whether the bus remains unused over a certain period of time while BB is set. It also avoids sending a STOP signal in the middle of the transaction of another device on the bus.

4. In some cases, the bus may get stuck with the SCL and/or SDA lines active, such as when an erroneous Start or Stop Condition occurs in the middle of a slave receive session.

   If the SCL line is stuck active, the module that holds the bus must release it.

   If the SDA line is stuck active, the sequence below releases the bus (Note**:** In normal cases, SCL may be toggled only by the bus master; this sequence is a recovery scheme which is an exception and should be only used if there is no other master on the bus):

   a. Disable and re-enable the module to set it for the Slave mode not addressed.
   b. Set START in ACBnCTL1 register to attempt to issue a Start Condition.
   c. Check if the SDA line is active (low) by reading TSDA in ACBnCST register. If it is active, issue a single SCL cycle by writing 1 to TGSCL in the same register. If it is not active, skip to step e.
   d. Check if MASTER in ACBnST register is set, which indicates that the Start Condition was sent. If it is not set, repeat step c and this step until the SDA is released.
   e. Clear BB. This enables START to be executed. Continue according to “Bus Idle Error Recovery” on page 188.

# 4.0 Embedded Controller Modules (Continued)

## 4.14 ANALOG COMPARATORS MONITOR (ACM)

The Analog Comparators Monitor (ACM) checks the voltage level of eight analog inputs and reports their values to the core.

The ACM can either measure the voltage of each input with 6-bit resolution or compare the level of all eight inputs with a programmable threshold.

An operation burst includes either the voltage measurement of all inputs or one comparison check of all inputs to a fixed threshold. The bursts may be triggered by a low-frequency clock for periodic operation.

### 4.14.1 Features

- Voltage measurement
  - Eight analog inputs
  - 6-bit resolution
  - 0V to $V_{CC}$ Full Scale (FS) input voltage range
  - High-impedance inputs
- Comparison check
  - 6-bit threshold resolution
  - Simultaneous on all inputs
- Ratiometric measurement using $V_{CC}$
- Digital reading output
  - Eight buffers for voltage value
  - One buffer for comparison check
- 170 µs voltage measurement of all the inputs
- Polling- or interrupt-driven interface
- Power consumption
  - Zero current when disabled
  - Low average current in Idle mode

### 4.14.2 Functional Description

**Inputs.** The ACM has eight analog inputs with a voltage ranging from 0V to $V_{CC}$ (FS), as shown in Figure 71.

The voltage level at each input is compared with a reference level, generated by a 6-bit D/A converter.

**Operation.** The ACM can be operated in one of three modes:

- Voltage Level Burst mode - The reference level generated by the D/A converter is ramped from 0V to FS by incrementing the value loaded into the converter. Whenever the ramp crosses the level of an input, its comparator toggles state, and the D/A loaded value is latched into the corresponding Voltage Level Buffer. At the end of the ramp, execution stops and a set of eight voltage level values (one per input) is available.
- Threshold Comparison Burst mode - A programmable value is loaded into the D/A converter, generating a constant threshold level. A bit is set for each input if the input's voltage level is higher than the threshold. At the end of the single comparison, the Comparison Result register holds the status of the eight inputs.
- Low Power Threshold Comparison mode - The inputs are periodically compared with a fixed threshold. The comparison is triggered by a low rate signal from the TWD module (T0IN). In addition, an interrupt is generated if at least one input is below (or above) the threshold.

**Timing.** When the PC87591L-N05 is in Active or Active Executing Wait mode, ACM module timing is based on the core domain clock (CLK). When the PC87591L-N05 is in Idle mode, timing is based on the low frequency clock (LFCLK) to minimize power consumption.

## 4.0 Embedded Controller Modules (Continued)

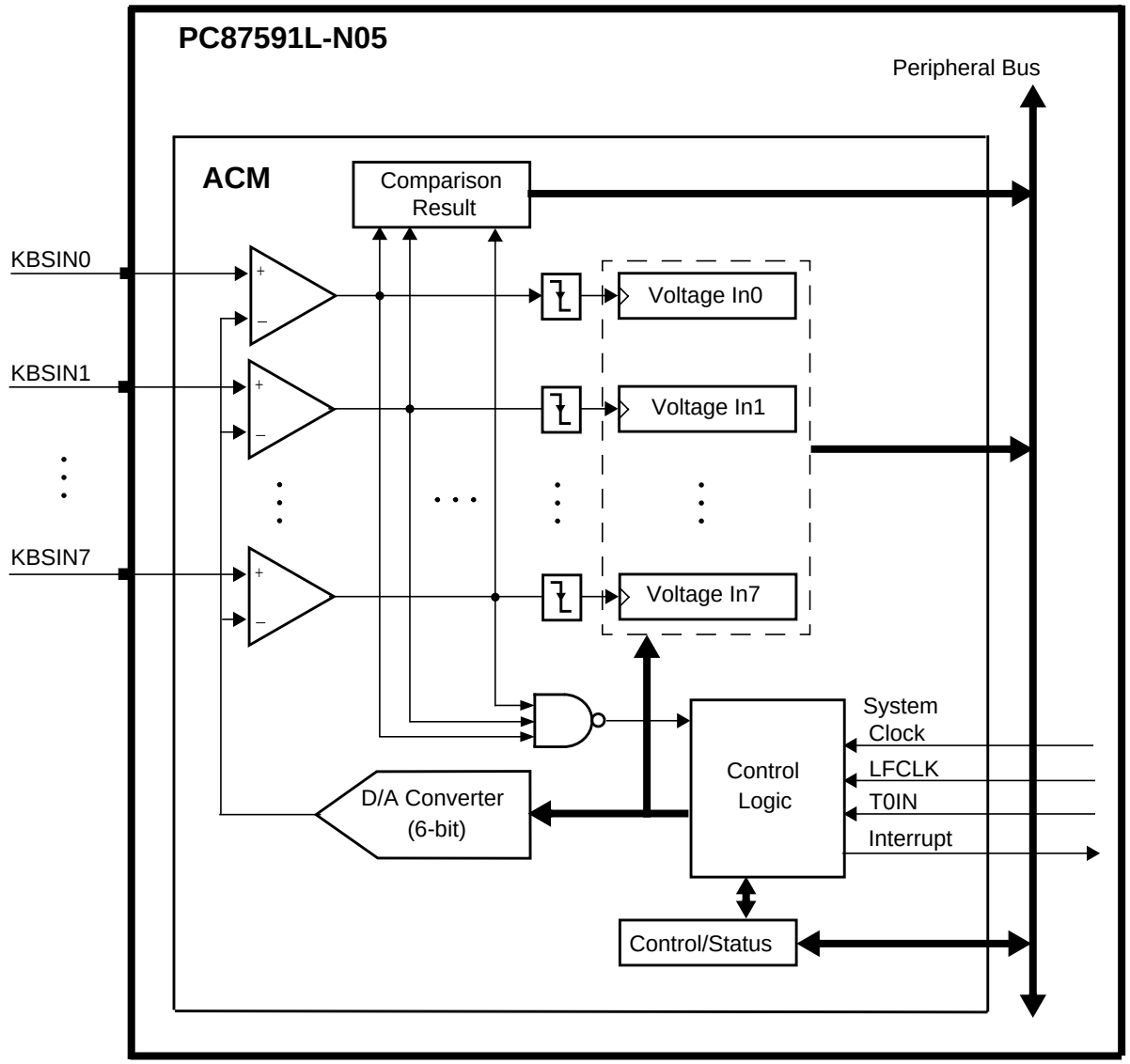



**Figure 71. ACM Functional Diagram**

### 4.14.3 Voltage Level

$V_{CC}$ supply is the reference and full scale (FS) voltage for the D/A converter. This enables ratiometric voltage measurement (parts of FS), were $V_{CC}$ is used to power an external voltage divider connected to the KBSINn inputs.

The 6-bit Voltage Level Data field (VOLTLVL) in VOLDATn (n = 0 to 7) registers and the Comparison Threshold Data field (THRSHD5-THRSHD0) in THRDAT register are converted to input voltage, according to input voltage ratio (to FS), as shown in Figure 72.

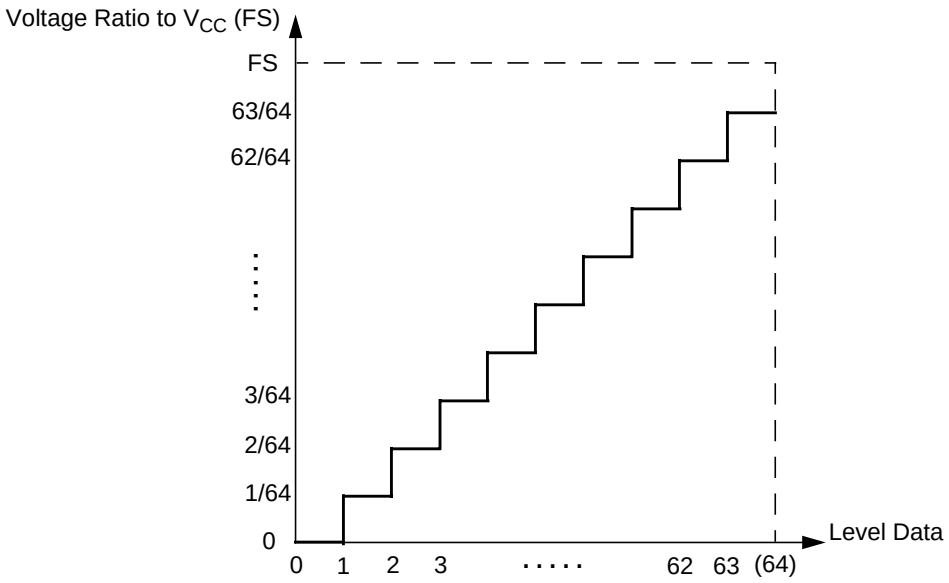



**Figure 72. Level Data to Input Voltage Ratio Conversion**

### 4.14.4 ACM Operation

#### Reset

The ACM is initialized by core domain reset (Section 3.2 on page 61 describes the PC87591L-N05 reset events).

Upon reset, the ACM is disabled, with all interrupt sources masked and the over/under threshold select set to “below”. All Event status bits are reset. The trigger division factor, as well as the data sampling delay, are set to their maximum value (for the slowest ACM operation speed). The comparison threshold is set to zero.

All control, configuration and status registers are reset to their default values, as indicated in Section 4.14.5 on page 201. Voltage Level Data Buffer registers and the Comparison Result register’s values are invalid until the first measurement occurs (on each of them).

#### Sampling Delay

This delay separates between the new value being loaded into the D/A converter and the data being sampled into the Voltage Level Data Buffer registers and the Comparison Result register. Its value should be longer than the total time required by the D/A output to settle (within 1/2 LSB of 6 bits) and the comparators to toggle their state.

The value set by SMPDLY field in ACMTIM register is expressed in terms of core CLK clock cycles. The maximum frequency at which each delay may be used is specified in the register’s description.

#### Initializing the ACM

The ACM must be initialized before it is enabled and used. The following operations should be done:

- Select the ACM Mode Control by setting ACMMOD field in ACMCNF register.
- Enable interrupts are required using the following bits of ACMCNF register:
  - — Interrupt from End-of-Measurement Event Enable by setting INTEMEN bit.
  - — Interrupt from Over/Under Threshold Event Enable by setting INTOUEN bit.
- Select Over or Under Threshold mode, using OVUNSEL bit in ACMCNF register.
- Set the data sampling delay using SMPDLY field in ACMTIM register.
- Select the low power trigger rate by setting T0DIV field in ACMTIM register.
- Select the Comparison Threshold Data for wake-up by setting THRSHD field in THRDAT register.

After initializations are done, the ACM may be enabled by writing 1 to START bit in ACMCTS register.

**Note:** Setting any of the above bits/fields during ACM operation may cause unpredictable results.

# 4.0 Embedded Controller Modules (Continued)

## Interrupt Structure

The ACM Interrupt is generated to the core if an event from Table 23 becomes active. Since the events are enabled in different ACM operation modes, only one event may occur at a time.

**Table 23. ACM Interrupt Structure**

| Event Flag | Register Mnemonic | Mask Bit | Register Mnemonic | Description |
|---|---|---|---|---|
| EOMEV | ACMCTS | INTEMEN | ACMCNF | End-of-Measurement event and associated interrupt enable |
| OVUNTHEV | ACMCTS | INTOUEN | ACMCNF | Over/Under Threshold event and associated interrupt enable |

When an event flag and its related mask bit are set (enabled), the ACM Interrupt request is asserted. This is indicated by a high level of the ACM Interrupt signal.

The software must reset the event flag (or its mask bit) to de-assert the ACM Interrupt request.

All the event flags (EOMEV, EOCEV and OVUNTHEV) are cleared by writing 1 to START in ACMCTS register.

The ACM Interrupt is routed both to the ICU and to the MIWU as an ACMI signal (see Section 4.3 on page 96 and Section 4.4 on page 103).

## ACM Operating Sequences

The ACM has three operating modes:

- Voltage Level Burst mode
- Threshold Comparison Burst mode
- Low-Power Threshold Comparison mode

After the ACM is properly initialized, use one of these three modes. The operation sequences are as follows:

**Voltage Level Burst Mode (ACMMOD = $01_2$)**

1. Start a new burst of eight voltage measurements by setting START bit in ACMCTS register to 1.
2. When End-of-Voltage level measurement burst is reached, software can detect the event by waiting for EOMEV in ACMCTS register to be set to 1.
3. Read measured voltage level - Input 0, for channel 0, by reading VOLTLVL in VOLDAT0 register.
4. Read measured voltage level - Input 1, for channel 1, by reading VOLTLVL in VOLDAT1 register.
5. Read measured voltage level - Input 2, for channel 2, by reading VOLTLVL in VOLDAT2 register.
6. Read measured voltage level - Input 3, for channel 3, by reading VOLTLVL in VOLDAT3 register.
7. Read measured voltage level - Input 4, for channel 4, by reading VOLTLVL in VOLDAT4 register.
8. Read measured voltage level - Input 5, for channel 5, by reading VOLTLVL in VOLDAT5 register.
9. Read measured voltage level - Input 6, for channel 6, by reading VOLTLVL in VOLDAT6 register.
10. Read measured voltage level - Input 7, for channel 7, by reading VOLTLVL in VOLDAT7 register.
11. Clear the event flag (release the ACM interrupt if enabled) by writing 1 to EOMEV in ACMCTS register.

**Threshold Comparison Burst Mode -(ACMMOD = $10_2$)**

1. Set the desired threshold value in bits THRSHD(5-0) in THRDAT register.
2. Start a new burst of threshold comparison by setting START in ACMCTS register to 1.
3. When End-of-Threshold comparison burst is reached, software can detect the event by waiting for EOCEV in ACMCTS register to be set to 1.
4. Read comparison result for inputs 0 to 7 by reading bits CMPIN(0-7) in CMPRES register.
5. Clear the event flag by writing 1 to EOCEV in ACMCTS register.

**Low-Power Threshold Comparison Mode (ACMMOD = $11_2$)**

1. Set the desired threshold value in bits THRSHD(5-0) in THRDAT register.
2. Start a new periodic, low-power threshold comparison by setting START in ACMCTS register to 1.
3. When there is an input under (or over) threshold event, software can detect the event by waiting for OVUNTHEV in ACMCTS register to be set to 1.
4. Read comparison result for inputs 0 to 7 by reading bits CMPIN(0-7) in CMPRES register.
5. Clear the event flag (release the ACM interrupt if enabled) by writing 1 to OVUNTHEV in ACMCTS register.

**Polling Driven Operation.** Valid measurement results may be retrieved by polling the event flags in ACMCTS register, EOMEV, EOCEV or OVUNTHEV, according to the selected operation mode. After reading the relevant data, clear the polled event flag by writing 1.

**Interrupt Driven Operation.** When receiving an active ACM interrupt request, the software should check the two event flags, EOMEV or OVUNTHEV, to identify the source of the interrupt. After reading the relevant data, clear (by setting to 1) the event flag that caused the interrupt. Clearing the flag releases the ACM interrupt request.

### 4.14.5 ACM Registers

The ACM control/status and data out register set interfaces with the core through the Peripheral bus.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### ACM Register Map

The ACM register set contains four control and status registers and nine data out registers.

**Table 24. ACM Register Map**

| Mnemonic | Register Name | Type |
|---|---|---|
| ACMCTS | ACM Control and Status | Varies per bit |
| ACMCNF | ACM Configuration | R/W |
| ACMTIM | ACM Timing Control | R/W |
| THRDAT | Comparison Threshold Data | R/W |
| CMPRES | Comparison Result | RO |
| | Reserved | |
| VOLDAT0 | Voltage Level Data - Input 0 | RO |
| VOLDAT1 | Voltage Level Data - Input 1 | RO |
| VOLDAT2 | Voltage Level Data - Input 2 | RO |
| VOLDAT3 | Voltage Level Data - Input 3 | RO |
| VOLDAT4 | Voltage Level Data - Input 4 | RO |
| VOLDAT5 | Voltage Level Data - Input 5 | RO |
| VOLDAT6 | Voltage Level Data - Input 6 | RO |
| VOLDAT7 | Voltage Level Data - Input 7 | RO |

# 4.0 Embedded Controller Modules (Continued)

## ACM Control and Status Register (ACMCTS)

This register controls the operation and indicates the status of the ACM module. ACMCTS is cleared ($00_{16}$) on reset.

Location: 00 $FD40_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | **OVUNTHEV** | **EOCEV** | **EOMEV** | **START** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Type | Description |
|---|---|---|
| 0 | W | **START (ACM Operation Start).** Start trigger for the ACM operation in the mode selected by ACMMOD field in ACMCNF register. If the new ACMMOD value differs from the value currently in use, all event flags (EOMEV, EOCEV, OVUNTHEV) are cleared and operation in the new mode begins. Writing a 1 to the START bit without modifying the ACMMODE value is ignored by the module.<br>0: No effect - (default)<br>1: Operation start trigger - automatically returns to 0 after start |
| 1 | R/W1C | **EOMEV (End-of-Measurement Event).** End of voltage level measurement burst (for all eight inputs). The bit is set only when the ACM is in Voltage Level Burst mode (ACMMOD = $01_2$ in ACMCNF register).<br>0: Measurement in progress, or **not** in Voltage Level Burst mode (default)<br>1: End of voltage level measurement burst |
| 2 | R/W1C | **EOCEV (End-of-Comparison Event).** End of threshold comparison in Threshold Comparison Burst mode (ACMMOD = $10_2$ in ACMCNF register).<br>0: Comparison in progress, or **not** in Threshold Comparison Burst mode (default)<br>1: End of threshold comparison |
| 3 | R/W1C | **OVUNTHEV (Over/Under Threshold Event).** At least one input is above (or below) the voltage threshold value. The bit is set only when the ACM is in Low Power Threshold Comparison mode (ACMMOD = $11_2$ in ACMCNF register).<br>0: No input above (or below) the threshold, or **not** in Low Power Threshold Comparison mode (default)<br>1: One or more inputs above (or below) the threshold |
| 7-4 | | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## ACM Configuration Register (ACMCNF)

This register controls the configuration of the ACM module. ACMCNF is cleared ($00_{16}$) on reset.

Location: 00 $FD42_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **OVUNSEL** | **INTOUEN** | **INTEMEN** | **Reserved** | **ACMMOD** | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 1-0 | **ACMMOD (ACM Mode Control).** Configures the operation mode of the ACM module. See “ACM Operating Sequences” on page 200. This value becomes the active mode after writing a 1 to START bit in ACMCTS register.<br>**Bits 1 0 Description**<br>0 0: ACM disabled (default)<br>0 1: Voltage Level Burst mode - single measurement of the voltage level for all the eight inputs<br>1 0: Threshold Comparison Burst mode - single comparison of all the eight inputs to the set threshold<br>1 1: Low-Power Threshold Comparison mode - periodic comparison of the inputs to a constant threshold (recommended for Idle PC87591L-N05 operation mode). Its operation is stopped by the selection of a different mode (ACM disabled = 002 is the recommended value). |
| 2 | **Reserved.** |
| 3 | **INTEMEN (Interrupt from End-of-Measurement Event Enable).** Enables generation of an ACM interrupt on an End of Voltage Measurement Burst event (EOMEV in ACMCTS register).<br>0: Disabled (default)<br>1: Enabled - ACM Interrupt from EOMEV |
| 4 | **INTOUEN (Interrupt from Over/Under Threshold Event Enable).** Enables generation of an ACM interrupt on at least one input above (or below) the threshold event (OVUNTHEV ACMCTS register).<br>0: Disabled (default)<br>1: Enabled - ACM Interrupt from OVUNTHEV |
| 5 | **OVUNSEL (Over or Under Threshold Select).** “Any Over” or “Any Under” logic selection for OVUNTHEV bit in ACMCTS register.<br>0: At least one input below threshold sets OVUNTHEV=1 (default)<br>1: At least one input above threshold sets OVUNTHEV=1 |
| 7-6 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## ACM Timing Control Register (ACMTIM)

This register controls the sampling delay for the voltage level and compare out data; it also controls the division of the trigger signal. ACMTIM is set to $37_{16}$ on reset.

Location: 00 $FD44_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **T0DIV** | | **Reserved** | **SMPDLY** | | |
| Reset | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 2-0 | **SMPDLY (Data Sampling Delay).** Compensates for the settling time of the D/A converter and the input comparators. To calculate the required delay value, see “Sampling Delay” on page 199.<br>**Bits 2 1 0** / **Settling Period[1]** / **Compare Period[1]** / **Max Core Freq[2]**<br>0 0 0: 1 10 5.5 MHz<br>0 0 1: 1 15 8.5 MHz<br>0 1 0: 2 20 11.5 MHz<br>0 1 1: 2 25 14.2 MHz<br>1 0 0: 2 30 16.5 MHz<br>1 0 1: 2 35 20.0 MHz<br>1 1 0: 3 45 20.0 MHz<br>1 1 1: 3 55 20.0 MHz (default)<br>1. The Settling and Compare periods are in clock cycles.<br>2. Conversion step time is equal to Settling Period + Compare Period |
| 3 | **Reserved.** |
| 5-4 | **T0DIV (Trigger Signal Division Factor).** Controls the division of the T0IN trigger signal in Low Power Threshold Comparison mode. A higher division factor gives a lower power consumption in Idle operation mode (see Section 4.17 on page 208). T0IN frequency is set separately (see Section 4.10 on page 160).<br>**Bits 5 4** / **T0In Division Factor**<br>0 0: 16<br>0 1: 32<br>1 0: 64<br>1 1: 128 (default) |
| 7-6 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## Comparison Threshold Data Register (THRDAT)

This register holds the data used by the D/A converter in the threshold comparison modes. ACMCNF is cleared ($00_{16}$) on reset.

Location: 00 $FD46_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **THRSHD5-THRSHD0** | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 5-0 | **THRSHD5-THRSHD0 (Comparison Threshold Data).** Data to be used by the D/A converter in Threshold Comparison Burst and Low Power Threshold Comparison modes. All eight inputs are compared with the resulting voltage value.<br>Range: 0 to 63 (0 to 63/64 * $V_{CC}$); 6-bit, unsigned value with 1 LSB = $V_{CC}$/64 (the value at reset is 0). |
| 7-6 | **Reserved.** |

## Comparison Result Register (CMPRES)

This register contains the result of the input comparison with the threshold.

Location: 00 $FD48_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **CMPIN7** | **CMPIN6** | **CMPIN5** | **CMPIN4** | **CMPIN3** | **CMPIN2** | **CMPIN1** | **CMPIN0** |

| Bit | Description |
|---|---|
| 0 | **CMPIN0 (Comparison Result for Input 0).** Result of KBSIN0 comparison with the voltage threshold value. Relevant only in Threshold Comparison Burst mode (ACMMOD = $10_2$ in ACMCNF register) and Low Power Threshold Comparison mode (ACMMOD = $11_2$). The value at reset is undefined.<br>0: Input below threshold<br>1: Input above threshold |
| 1 | **CMPIN1 (Comparison Result for Input 1).** Same as CMPIN0 for KBSIN1. |
| 2 | **CMPIN2 (Comparison Result for Input 2).** Same as CMPIN0 for KBSIN2. |
| 3 | **CMPIN3 (Comparison Result for Input 3).** Same as CMPIN0 for KBSIN3. |
| 4 | **CMPIN4 (Comparison Result for Input 4).** Same as CMPIN0 for KBSIN4. |
| 5 | **CMPIN5 (Comparison Result for Input 5).** Same as CMPIN0 for KBSIN5. |
| 6 | **CMPIN6 (Comparison Result for Input 6).** Same as CMPIN0 for KBSIN6. |
| 7 | **CMPIN7 (Comparison Result for Input 7).** Same as CMPIN0 for KBSIN7. |

### Voltage Level Data Buffer - Input 0 through 7 (VOLDAT0-7)

This register holds the voltage level measurement result for inputs 0 through 7.

Location: 00 $FD50_{16}$ - Input 0
00 $FD52_{16}$ - Input 1
00 $FD54_{16}$ - Input 2
00 $FD56_{16}$ - Input 3
00 $FD58_{16}$ - Input 4
00 $FD5A_{16}$ - Input 5
00 $FD5C_{16}$ - Input 6
00 $FD5E_{16}$ - Input 7

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **VOLTLVL** | | | | | |

| Bit | Description |
|---|---|
| 5-0 | **VOLTLVL (Voltage Level Data).** Voltage level of the respective input i (KBSINi for i 0 through 7), while operating in Voltage Level Burst mode (ACMMOD = $01_2$ in ACMCNF register). The readout is an unsigned value with 1 LSB = $V_{CC}/64$; the input voltage is calculated as: $V_{IN}$ = VOLTVL * ($V_{CC}/64$). |
| 7-6 | **Reserved.** |

### 4.14.6 Usage Hints

### Voltage Level Burst

The total duration of the Voltage Level Burst measurement is a function of the system clock frequency (see Section 4.18 on page 212) and the data sampling delay, selected by SMPDLY field in ACMTIM register. The values in Table 25 should only be used to evaluate the maximum duration of the measurement. Actual duration may be shorter, depending on the input level value. The software should use either EOMEV flag polling or the ACM interrupt, as described in Section 4.14.4 on page 199.

**Table 25. Voltage Level Burst Duration**

| System Clock Frequency (MHz) | SMPDLY Setting | Maximum Burst Duration (µs) |
|---|---|---|
| 4 | 000 | 176 |
| 10 | 010 | 140 |
| 20 | 101 | 118 |

# 4.0 Embedded Controller Modules (Continued)

## 4.15 ON-CHIP RAM

The PC87591L-N05 contains 4096-bytes of on-chip RAM.

A 16-bit wide data bus links the core and the system RAM array. The system RAM can be byte or word accessed. Each system RAM read or write operation is one cycle long and does not include any wait states. See Section 1.5.1 on page 29 for system RAM memory map.

## 4.16 ON-CHIP ROM

The PC87591L-N05 contains 4096-bytes of on-chip ROM.

The on-chip ROM contains the core boot code (the Booter program; see Appendix C on page 394). ROM contents can not be modified.

# 4.0 Embedded Controller Modules (Continued)

## 4.17 POWER MANAGEMENT CONTROLLER (PMC)

The Power Management Controller (PMC) module improves the efficiency of PC87591L-N05 operation by adjusting the chip's power consumption to the level of activity required by the application. This module works together with the High-Frequency Clock Generator (HCFG) and the core to control the activity of the PC87591L-N05. It also interacts with the Multi-Input Wake-Up (MIWU), Interrupt Control Unit (ICU) and Debugger interface for wake-up events.

### 4.17.1 Features

- Three core domain power modes:
  - Active
  - Idle
  - Power Off
- Two clock inputs:
  - High-frequency clock (HFCLK)
  - Low-frequency clock (LFCLK)
- Power mode switching by software and/or hardware control
- High-frequency clock source Enable/Disable control
- Other core domain modules are controlled with power mode indications

### 4.17.2 The Core Domain Power Modes

Table 26 summarizes the main properties of the three modes and shows the activity levels of clocks while in the various power states.

**Table 26. Core Domain Power Mode Summary**

| Mode | HFCG | LFCG | CLK | LFCLK[1] | $V_{CC}$ Supply |
|---|---|---|---|---|---|
| Active[2] | On | On | HF | LF | On |
| Idle | On or Off[3] | On | Off | LF | On |
| Power Off | Off | On | Off | Off | Off |

1. The RTC and TWM modules always work from the LF oscillator.
2. The core may execute the WAIT instruction while in Active mode to reduce power consumption while no core activity is required. This state is referred to as "Active Executing WAIT" in some places in the specification, but it is not a separate power state as far as clocks are concerned.
3. Can be turned off by software but depends also on SuperI/O clock domain activation.

#### Active Mode

In Active mode, the core domain operates at the frequency generated by the HFCG. This frequency may be changed dynamically using the Load, Fast, Load96 or Fast96 operations in the HFCG module. The module's respective enable/disable bits control module activity.

In this mode, power consumption can be reduced by selectively disabling modules and/or by the core executing the WAIT instruction. When WAIT is executed, the core stops executing new instructions until it receives an interrupt signal.

After reset, the PC87591L-N05 is in Active mode.

#### Idle Mode

In Idle mode, the clock is stopped for most of the core domain. Only the PMC and a limited number of core domain modules (such as the TWD) continue to operate at the low-frequency oscillator rate; they can wake up the core domain and resume instruction execution when required.

For modules that are active in Idle mode, details of their activity are included in the module's description.

Wake-up events are generated by the MIWU module according to the enabled internal and external events.

#### Power Off Mode

When $V_{CC}$ power is turned off, the core domain reaches its lowest activity level. The contents of the memories (except for the on-chip ROM) and registers are not preserved in this mode. Applying power to the core domain should be done using the $V_{CC}$ Power-Up reset sequence.

## 4.0 Embedded Controller Modules (Continued)

A battery supply pin ($V_{BAT}$) provides power to the RTC and low-frequency clock oscillator (LFCO), allowing these parts to continue functioning even in Power Off mode. Some registers are reserved and events recorded. These registers and events are maintained by $V_{PP}$, as noted.

Details of the activity of each battery-operable module are provided in the module's specification.

### 4.17.3 Switching Between Power Modes

The switching from one power mode to another is done using the protocols described below.

Figure 73 shows the three power modes of the core domain and the transitions between them.

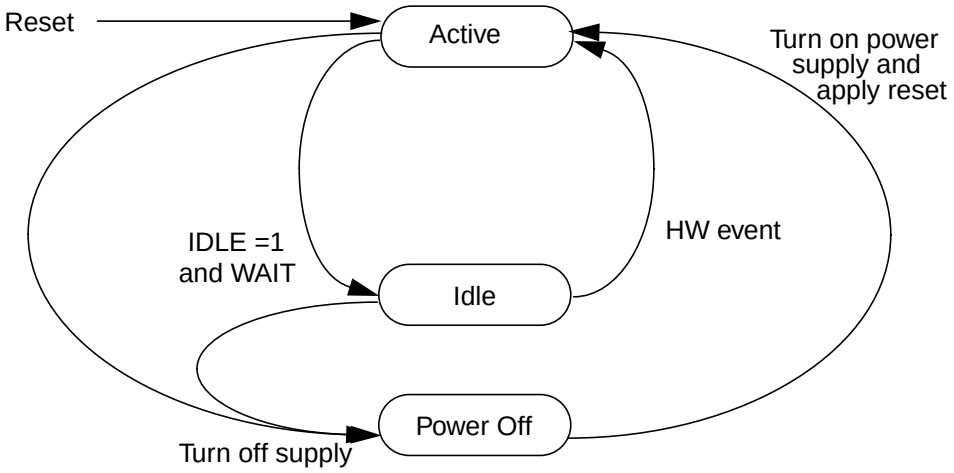



**Figure 73. Power Modes and Transitions**

**Wake-Up Event**

Some of the power-up switchings are based on receiving a wake-up event. This event has three possible sources:

- A maskable event (from MIWU)
- A Non-Maskable Interrupt (NMI)
- An ISE interrupt (from the Debugger interface)

The wake-up is identified by a high level on the maskable event and/or a low-to-high transition on NMI or ISE interrupts. Once a wake-up event is detected, it is latched until an interrupt acknowledge bus cycle is detected or a reset is applied.

### Decreasing Power Consumption

**Entering Idle Mode**

Enter Idle mode by setting (1) IDLE bit in PMCSR register and then executing the WAIT instruction. WBPSM must be set before executing WAIT.The HFCG may be disabled to further reduce the power consumption. This is done by writing 1 to DHF in PMCSR register before executing WAIT.

**Entering Power Off Mode**

Switch to Power Off mode by turning off the supply to the $V_{CC}$ pins of the PC87591L-N05. Note that $V_{DD}$ must be turned off as well.

The $\overline{\text{PFAIL}}$ input may be used to interrupt the PC87591L-N05 so that context saving to a non-volatile memory can be completed and write operations to the RTC can be stopped before the power to the PC87591L-N05 is disconnected.

### Increasing Activity

**Fast Wake-Up from Idle Mode to Active**

A hardware wake-up event causes the core domain to switch directly from Idle mode to Active mode. The following sequence is performed:

1. DHF in PMCSR register is cleared, thus enabling the high-frequency clock (if it was disabled).
2. After waiting for the high-frequency clock to become active (OHFC in PMCSR register is set), the core domain switches to Active mode.

When in Active mode, Idle bit in PMCSR register is cleared.

If the core was executing a WAIT instruction, it resumes operation by entering an interrupt routine (an enabled interrupt in the ICU, NMI or ISE).

## 4.0 Embedded Controller Modules (Continued)

### Exit from Power Off

When in Power Off, activity can be resumed only by switching to Active mode. This is done by applying $V_{CC}$ power to the PC87591L-N05. The Power-Up reset sequence described in “VCC Power-Up Reset” on page 62 should be applied.

### Power Mode Switch Protection

The PMC module includes a mechanism that protects the PC87591L-N05 from malfunctions caused by missing or unstable clock signals.

### Clock Toggling Indication

OHFC and OLFC bits in PMCSR register indicate the current status of the high- and low-frequency clock inputs, respectively. The current status is based on indications from the HFCG and LFCG modules.

The PMC does not use the high-frequency clock when the OHFC bit is 0; it does not use the low-frequency clock when OLFC is 0.

During reset, the PC87591L-N05 clock does not toggle until OHFC is 1. During power mode change, if there is a request to switch to a non-stable or non-toggling clock, the power mode change stalls.

### 4.17.4 The Power Management Controller Status Register (PMCSR)

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

PMCSR is a byte-wide, read/write register that selects the Active or Idle modes. In addition, it controls the operation of the HFCG by enabling or disabling the high-frequency core clock domain. On reset, all non-reserved bits are cleared. PMCSR format is shown below.

Location: 00 $FF80_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **OLFC** | **OHFC** | **WBPSM** | **Reserved** | | **IDLE** | **DHF** | **Reserved** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **Reserved.** |
| 1 | **DHF (Disable High-Frequency Oscillator).** When cleared (0), the HFCG is enabled. In Active mode, the HFCG is enabled regardless of the DHF value.<br>If in Idle mode, DHF can be used to reduce power consumption. When DHF=1, the HFCG is disabled and the high-frequency clock is not generated. In Power off mode, the HFCG is disabled regardless of the DHF value.<br>DHF is cleared by the hardware when a hardware wake-up event is detected. |
| 2 | **IDLE.** When set, the core domain enters Idle mode on the execution of a WAIT instruction. WBPSM must be set before executing the WAIT instruction.<br>This bit can be set and cleared by software; it is cleared by hardware when a hardware wake-up event is detected. |
| 4-3 | **Reserved.** |
| 5 | **WBPSM (Wait Before Entering Power Save Mode).** When set, the switch from Active to Idle mode is done by setting the IDLE bit and executing a WAIT instruction. In addition, if DHF is set, the high-frequency oscillator is disabled only after the WAIT instruction is executed and Idle mode is entered. |
| 6 | **OHFC (Oscillating High-Frequency Clock).**<br>0: Indicates that the high-frequency clock received by the PMC is either disabled, not available or not producing a stable clock. When OHFC is cleared, the PMC does not switch to Active mode (default).<br>1: Indicates that the high-frequency clock received by the PMC is available and producing a stable clock. |
| 7 | **OLFC (Oscillating Low-Frequency Clock).**<br>0: Indicates that the low-frequency clock received by the PMC is either disabled, not available or not producing a stable clock. When the OLFC is cleared, the PMC does not switch from Active mode to Power Save or Idle modes (default).<br>1: Indicates that the low-frequency clock received by the PMC is available and producing a stable clock. |

# 4.0 Embedded Controller Modules (Continued)

## 4.17.5 Usage Hints

The hints below apply when Idle mode is used with a disabled HFCG.

1. When disabling HFCG in Idle mode, on wake-up, a frequency clock may be generated that differs from the selected setting due to temperature variations in the working environment during the idle period. For details on the resulting deviation from the nominal frequency, refer to “tCLKINTwk” on page 346. To avoid any failures that may result from waking up to a higher frequency in case the access time to the internal or external memory is marginal using the current BIU configuration, follow the procedures exactly.

   Before entering Idle mode, configure SZCFGi (where i=0-2) for an additional Wait clock cycle (see Section 4.1.10 on page 81 for details on SZCFGi configuration).
2. After waking up from Idle mode, wait 0.5 seconds before returning to the optimal SZCFG0 configuration.

# 4.0 Embedded Controller Modules (Continued)

## 4.18 HIGH-FREQUENCY CLOCK GENERATOR (HFCG)

The HFCG generates the high-frequency clock (OSCCLK) based on the system's 32.768 KHz clock signal. The HFCG output is derived from OSCCLK and generates the host domain clock and the core domain clock. Clock generation is controlled by the core domain's PMC module (see Section 4.17 on page 208) and the host domain's SuperI/O configuration (see Section 6 on page 297).

To generate OSCCLK, the HFCG includes a programmable frequency multiplier. The core domain clock is derived from OSCCLK via a programmable pre-scaler; its frequency is in the range of 4 MHz to 20 MHz. The host domain clock is derived from OSCCLK via a pre-scaler that divides by 2, generating an output of 48 MHz (see Figure 74).

### 4.18.1 Features

- Programmable frequency multiplier for a wide range of output frequencies
- Core domain clock and host domain clock generation
- Programmable pre-scaler to derive the core domain clock from OSCCLK
- Separate enable/disable for core domain clock and host domain clock
- On $V_{CC}$ power-up:
  - 4 MHz default core domain clock frequency is set
  - Host domain clock is disabled
- On Watchdog reset and Debugger Interface reset:
  - If host domain clock is enabled, the 48 MHz clock monitor is initiated
  - If host domain clock is disabled, the 4 MHz default core domain clock frequency is set

### 4.18.2 Functional Description

Figure 74 shows the HFCG blocks.

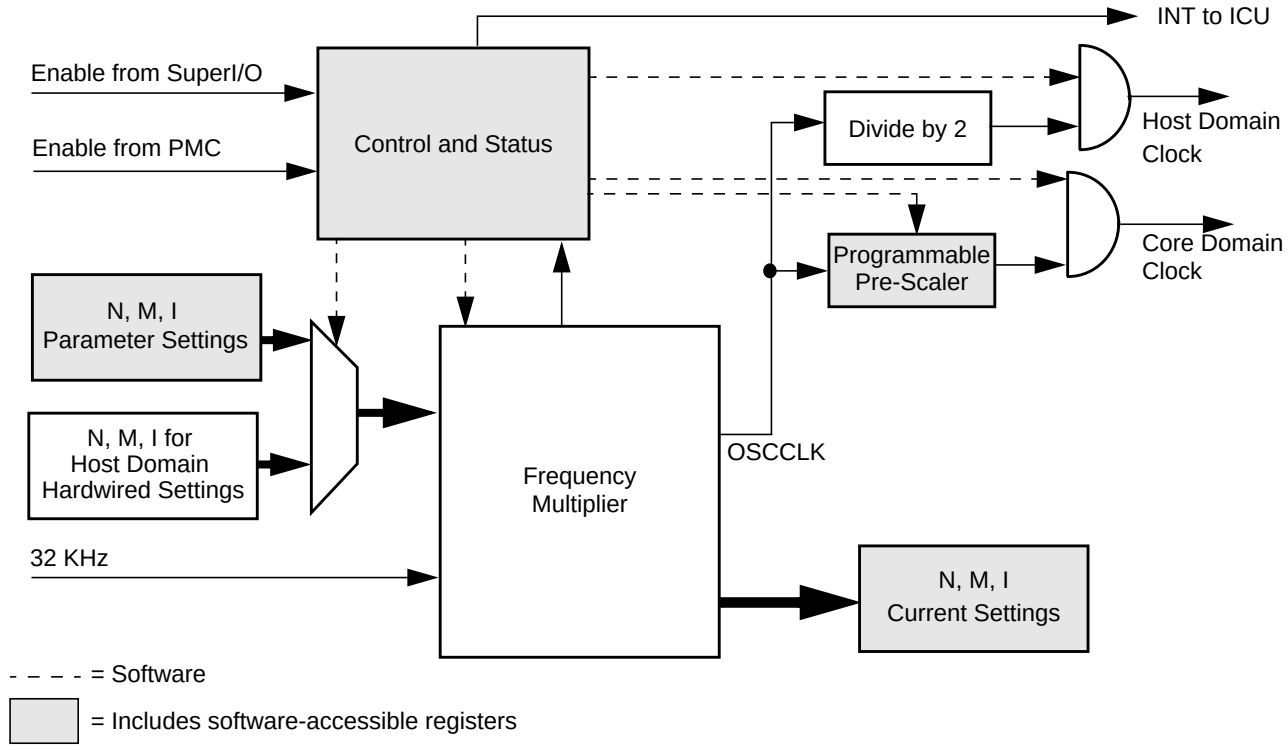



**Figure 74. HFCG Schematic Diagram**

The frequency multiplier includes a 5-bit variable (N) and a 14-bit variable (M); these variables define the OSCCLK frequency. There are two software methods to set the frequency of OSCCLK (by changing N and M); the method used depends on the HFCG state, as described in Section 4.18.3. The methods are:

**Software Method 1.** Write new values (HFCGML, HFCGMH and HFCGN registers) to a buffer and enable the programmable frequency setting. Either a normal or fast clock setting may be used. The programmable pre-scaler of the core domain clock is automatically set to a divide by 1; see "PMC Enabled SuperI/O Disabled State" on page 213.

**Software Method 2.** Enable the hardwired frequency setting (M and N are set to generate a 96 MHz OSCCLK). Either a normal or fast clock setting may be used. The programmable pre-scaler of the core domain clock is set according to the HFCGP register; see “PMC Enabled SuperI/O Disabled State” on page 213.

During a frequency change, the OSCCLK output is low to prevent the system from using an unstable clock.

The HFCG is designed to be tightly coupled with the PMC. The HFCG receives two enable signals: one from the PMC and the other from the host domain. The PMC can enable or disable core domain clock generation while in Idle mode; it enables the core domain clock in Active mode. The SuperI/O configuration enables or disables clock generation for the host domain according to the host processor requests.

### 4.18.3 HFCG States

There are three groups of HFCG states:

- PMC Enabled SuperI/O Disabled: OSCCLK is programmable (set by hardware or by software method 1). The core domain clock is enabled; the host domain clock is disabled.
- SuperI/O Enabled PMC Enabled/Disabled: OSCCLK is fixed at 96 MHz (set by hardware or by software method 2). The host domain clock is enabled; the core domain clock is either enabled or disabled depending on PMC.
- Disabled: OSCCLK is disabled. Both core domain clock and host domain clock are disabled.

Transitions between the states are controlled by either hardware or firmware. Figure 75 shows the states and the hardware or software transitions between them. Some of the software settings and transitions are protected to improve the system’s durability with regard to software errors; see details in the following sections.

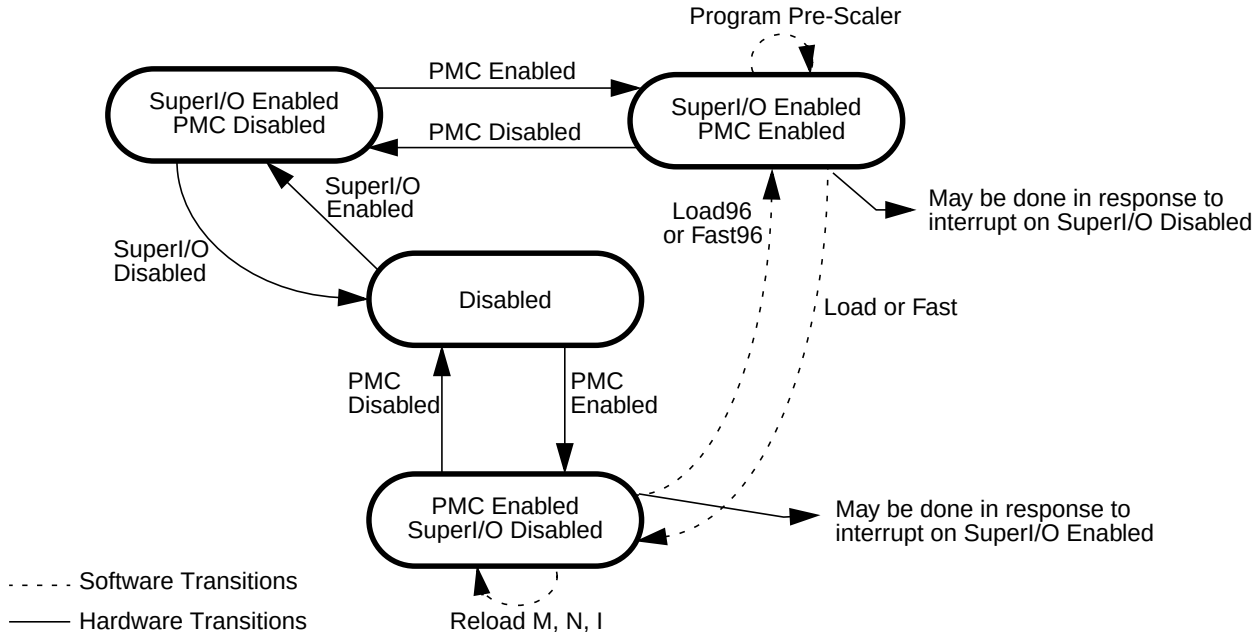



**Figure 75. HFCG States and Transitions**

### PMC Enabled SuperI/O Disabled State

**Normal Clock Setting.** This operation enables changing the clock frequency while in PMC Enabled SuperI/O Disabled state or switching from SuperI/O Enabled PMC Enabled state to PMC Enabled SuperI/O Disabled state.

To change the OSCCLK frequency, load the N and M variables with new values. M is loaded in two parts by writing to HFCGML and HFCGMH registers.

Load the new setting (N and M values, simultaneously) into the frequency multiplier. The core writes the new variables into a data input buffer. Then a command loads the new values into the frequency multiplier. The command also loads simultaneously the programmable pre-scaler with 0 (set to a divide by 1). Note that HFCGP register does not change its contents.

To set a new clock frequency:

1. Write the N value to HFCGN register.
2. Write the low byte of the M value to HFCGML register.
3. Write the upper bits of the M value to HFCGMH register.
4. Set LOAD in the HFCGCTRL1 register to 1.

## 4.0 Embedded Controller Modules (Continued)

When the new HFCGML, HFCGMH and HFCGN register values are loaded, the HFCG holds OSCCLK low until the frequency multiplier locks onto the target frequency and the new frequency stabilizes. The core domain clock is also low during this time. This automatic locking process can take several milliseconds to complete.

Frequencies within the range of 4 MHz to 20 MHz are valid. See Table 1 for a sampling of selected frequencies and their corresponding M and N values.

**Table 1. Frequencies of Selected Settings**

| Frequency[1] (MHz) | HFCGMH | HFCGML | HFCGN |
|---|---|---|---|
| 4.00 (default) | $03_{16}$ | $CF_{16}$ | $08_{16}$ |
| 5.00 | $03_{16}$ | $92_{16}$ | $06_{16}$ |
| 6.00 | $03_{16}$ | $92_{16}$ | $05_{16}$ |
| 7.00 | $07_{16}$ | $81_{16}$ | $09_{16}$ |
| 8.00 | $0A_{16}$ | $7C_{16}$ | $0B_{16}$ |
| 9.00 | $0A_{16}$ | $B9_{16}$ | $0A_{16}$ |
| 10.00 | $0A_{16}$ | $B9_{16}$ | $09_{16}$ |
| 12.00 | $11_{16}$ | $2A_{16}$ | $0C_{16}$ |
| 14.00 | $19_{16}$ | $07_{16}$ | $0F_{16}$ |
| 16.00 | $11_{16}$ | $29_{16}$ | $09_{16}$ |
| 18.00 | $11_{16}$ | $29_{16}$ | $08_{16}$ |
| 20.00 | $17_{16}$ | $D6_{16}$ | $0A_{16}$ |

1. This value is referred to as $t_{CLKINTnom}$ in the AC specifications.

**Fast Clock Setting.** This operation enables changing the clock frequency while in PMC Enabled SuperI/O Disabled state or switching from SuperI/O Enabled PMC Enabled state to PMC Enabled SuperI/O Disabled state.

The HFCG maintains an internal I variable. The I variable is defined by two byte-wide registers: HFCGIL and HFCGIH. If new M and N values are loaded, the frequency multiplier automatically searches for the I value needed to lock onto the target frequency. The locking process can take several milliseconds to complete. The I variable can be recorded for a given M and N set of values and used later to reduce the time needed for frequency locking.

To record the I value:

1. Read the low byte of the I value via HFCGIL register.
2. Check if IVLID bit in HFCGCTRL1 register is set to 0. If yes, repeat steps 1 and 2.
3. Read the upper six bits of the I value via HFCGIH register.

To fast load a new setting, load the M and N values and the corresponding I value. Then set FAST bit in HFCGCTRL1 register to 1. The FAST command loads (simultaneously) the new values into the frequency multiplier and loads the programmable pre-scaler with 0 (set to a divide by 1). Note that HFCGP register does not change its contents. The frequency multiplier quickly locks onto the target frequency without searching for a new I value.

To fast set a new clock frequency:

1. Write the N value to HFCGN register.
2. Write the low byte of the M value to HFCGML register.
3. Write the upper bits of the M value to HFCGMH register.
4. Write the low byte of the I value to HFCGIL register.
5. Write the upper six bits of the I value to HFCGIH register.
6. Set FAST bit in HFCGCTRL1 register to 1.

Changes in temperature or voltage may cause variations in the value of I for a given output frequency. If these changes occur in the interval between recording I and its use, the output frequency generated following a fast frequency setting may differ from the target frequency. However, after some time, the output frequency converges to the desired frequency.

### SuperI/O Enabled State

**Normal Clock Setting.** This operation enables switching from PMC Enabled SuperI/O Disabled state to SuperI/O Enabled PMC Enabled state.

To generate a 96 MHz OSCCLK, a command loads the hardwired M and N values (simultaneously) into the frequency multiplier. At the same time, the command loads the programmable pre-scaler with the value held in HFCGP register. To set a 96 MHz clock frequency and the required core clock.

1. Write the HFCGP value.
2. Set LOAD96 bit in HFCGCTRL1 register.

The HFCGN, HFCGML and HFCGMH registers are ignored and left unchanged when switching to SuperI/O Enabled states.

**Fast Clock Setting.** This operation enables fast switching from PMC Enabled SuperI/O Disabled state to SuperI/O Enabled PMC Enabled state.

The HFCG maintains an internal I variable for the 96 MHz clock. The I variable is defined by two byte-wide registers: HFCGIL and HFCGIH. In a LOAD96 operation, the frequency multiplier automatically searches for the I value needed to lock onto the target frequency. The locking process can take several milliseconds to complete. The I variable can be recorded for the 96 MHz setting and used later to reduce the time needed for frequency locking.

To record the 96 MHz I value:

1. Write the required HFCGP value.
2. Enter any of the SuperI/O Enabled states using the Normal clock setting or Hardware clock setting process.

To fast set a 96 MHz clock frequency:

1. Write the required HFCGP value.
2. Set FAST96 bit in HFCGCTRL1 register to 1.

The HFCGN, HFCGML and HFCGMH registers are ignored and left unchanged when switching to SuperI/O Enabled states.

### 4.18.4 The Programmable Pre-Scaler: Core Domain Clock Generation

The core domain clock (CLK) is driven from OSCCLK via a 5-bit pre-scaler. When in PMC Enabled SuperI/O Disabled state, the pre-scaler divides by 1. In SuperI/O Enabled PMC Disabled state, the pre-scaler is programmable, as defined in HFCGP register. In other states, the core-domain clock is disabled.

The core domain clock may be set in the range of 4 MHz to 20 MHz.

When LOAD96 or FAST96 bit in HFCGCTRL1 register is set (1), the pre-scaler is set to the value held in HFCGP register (core frequency = 96 MHz / (HFCGP +1)).

When in SuperI/O Enabled PMC Enabled state, a write to HFCGP register changes the core's frequency at the next cycle of the pre-scaler.

When LOAD or FAST bit in HFCGCTRL1 register is set (1), the pre-scaler is automatically set to a divide by 1. The contents of HFCGP are unchanged.

When in SuperI/O Enabled PMC Enabled state or SuperI/O Enabled PMC Disabled state, Watchdog reset or Debugger Interface reset sets HFCGP to its reset value and initializes the pre-scaler using this value; see Section 4.18.6 on page 216 for details about setting the pre-scaler during state transitions.

When in Disabled state or PMC Enabled and SuperI/O Disabled state, Watchdog reset or Debugger Interface reset keeps the pre-scaler in the divide by 1 operation and loads the HFCGP to its reset value.

In all states, on $V_{CC}$ power-up, Watchdog reset or Debugger Interface reset, HFCGP is set to its reset value and the pre-scaler is set to a divide by 1.

### 4.18.5 State Transitions

The following section describes the actions taken by the HFCG during state transitions. Unless explicitly specified, the actions are initiated by hardware.

#### Transition to PMC Enabled SuperI/O Disabled State

- When transitioning from Disabled state:
  - OSCCLK defaults to a frequency set by the input data buffer (according to the most recent LOAD, FAST or Reset command).
  - The pre-scaler defaults to a divide by 1.
- When transitioning from SuperI/O Enabled PMC Enabled state:
  - Software sets OSCCLK and the pre-scaler by the LOAD or FAST command (software method 1).
- The host domain clock is disabled (low).
- The core domain clock is enabled and starts toggling as soon as the frequency multiplier has stabilized.

When transitioning from SuperI/O Enabled PMC Enabled state, there may be a transition period in which the core domain clock is still toggling according to the setting in SuperI/O Enabled PMC Enabled state. This occurs until software method 1 is performed by the interrupt handler of the SuperI/O Disabled event.

**Transition to SuperI/O Enabled PMC Disabled State**

- OSCCLK defaults to 96 MHz
- The host domain clock is enabled and starts toggling as soon as the frequency multiplier has stabilized
- The core domain clock is disabled (low)

**Transition to SuperI/O Enabled PMC Enabled State**

- When transitioning from SuperI/O Enabled PMC Disabled state:
  - OSCCLK defaults to 96 MHz
  - The pre-scaler is set according to HFCGP
- When transitioning from PMC Enabled SuperI/O Disabled state:
  - Software sets OSCCLK and the pre-scaler by the LOAD96 or FAST96 command (software method 2)
- The host domain clock is enabled and starts toggling as soon as the frequency multiplier has stabilized
- The core domain clock is enabled and starts toggling as soon as the frequency multiplier has stabilized

When transitioning from PMC Enabled SuperI/O Disabled state, there may be a transition period in which the core domain clock is still toggling according to the setting in PMC Enabled SuperI/O Disabled state. This occurs until software method 2 is performed by the interrupt handler of the SuperI/O Enabled event. During this transition period, the host domain clock is still disabled.

### 4.18.6 48 MHz Clock Monitor

While in one of the SuperI/O Enabled states, a Watchdog reset or Debugger Interface reset does not interfere with the operation of the SuperI/O. Thus, if the 48 MHz clock is correctly toggling, it is not interfered with. Toggling is monitored by the 48 MHz clock monitor. If a violation is detected, the 48 MHz monitor error flag is set and the HFCG goes through a complete reset, after which it is put into PMC Enabled SuperI/O Disabled state, ignoring the host domain enable signal.

The 48 MHz clock monitor is initiated when a Watchdog reset or Debugger Interface reset occurs while the HFCG is in a SuperI/O Enabled state. The monitor checks that the host domain clock frequency is in the range of 48 MHz ±1%.

If the host domain clock frequency is in the correct range, the HFCG continues generating the 96 MHz OSCCLK, but the programmable pre-scaler is set to a default of divide by 24. The HFCG is in SuperI/O Enabled PMC Enabled state.

The core reset routine is responsible for switching to SuperI/O Enabled PMC Enabled state and/or communicating the failure to the host software.

### 4.18.7 HFCG Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

**HFCG Register Map**

| Mnemonic | Register Name | Type |
|---|---|---|
| HFCGCTRL1 | HFCG Control 1 | Varies per bit |
| HFCGML | M Low Byte Value | R/W |
| HFCGMH | M High Byte Value | R/W |
| HFCGN | N Value | R/W |
| HFCGIL | I Low Byte Value | R/W |
| HFCGIH | I High Byte Value | R/W |
| HFCGP | HFCG Pre-Scaler | R/W |
| HFCGCTRL2 | HFCG Control 2 | Varies per bit |

# 4.0 Embedded Controller Modules (Continued)

## HFCG Control Register 1 (HFCGCTRL1)

The HFCGCTRL1 register sets the frequency multiplier's operating parameters. On $V_{CC}$ power-up, Watchdog reset and Debugger Interface reset, it is initialized to $0C_{16}$.

Location: 00 $FFA0_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **FAST96** | **LOAD96** | **IVLID** | **OHFC** | **PENABLE** | **FAST** | **LOAD** |
| Reset | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

| Bit | Type | Description |
|---|---|---|
| 0 | WO | **LOAD (Load M and N Values).** Write 1 to this bit to perform a normal frequency change by loading the HFCGML, HFCGMH and HFCGN buffer data to the frequency multiplier. The pre-scaler is automatically set to a divide by 1. The bit always reads back 0. Results are undefined when more than one of the following bits is written with 1 at the same time: LOAD, FAST, LOAD96, FAST96. |
| 1 | WO | **FAST (Load M, N and I Values).** Write 1 to this bit to perform a fast frequency change by loading the HFCGML, HFCGMH, HFCGN, HFCGIH and HFCGIL input buffer data in the frequency multiplier. The pre-scaler is automatically set to a divide by 1. The bit always reads back 0. Results are undefined when more than one of the following bits is written with 1 at the same time: LOAD, FAST, LOAD96, FAST96. |
| 2 | RO | **PMC ENABLE (Enable Core Domain Clock).** Provides the status of the PMC enable/disable command. Any data written to this bit is ignored.<br>0: Disabled<br>1: Enabled (default) |
| 3 | RO | **OHFC (Output Clock Status).** Indicates when the HFCG is oscillating and produces a stable clock. Any data written to this bit is ignored.<br>0: Not oscillating<br>1: Oscillating with stable output (default) |
| 4 | RO | **IVLID (I Value Valid).**<br>0: Data read is invalid; repeat HFCGIL register read operation (default)<br>1: Data read is valid; HFCGIH can be read |
| 5 | WO | **LOAD96 (Load M and N Values for 96 MHz).** Write 1 to this bit to perform normal locking on 96 MHz frequency by loading the hardwired variables M and N to the frequency multiplier. The pre-scaler is set according to the value held in HFCGP register. The bit always reads back 0. Results are undefined when more than one of the following bits is written with 1 at the same time: LOAD, FAST, LOAD96, FAST96. |
| 6 | WO | **FAST96 (Load M, N and I Values** for 96 MHz**).** Write 1 to this bit to perform fast locking on 96 MHz frequency by loading the hardwired variables M and N and registers HFCGIH and HFCGIL to the frequency multiplier. The pre-scaler is set according to the value held in HFCGP register. The bit always reads back 0. Results are undefined results when more than one of the following bits is written with 1 at the same time: LOAD, FAST, LOAD96, FAST96. |
| 7 | | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## HFCGM Low Value Register (HFCGML)

The HFCGML register contains the lower eight bits of the frequency multiplier M value. Data written to the register is stored in the setup buffer. Reading the register returns the HFCGML value of the currently set frequency. On $V_{CC}$ power-up, Watchdog reset and Debugger Interface reset, it is loaded with $CF_{16}$.

Location: 00 $FFA2_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **HFCGM7-0** | | | | | | | |
| Reset | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 7-0 | **HFCGM7-0.** Contains the lower eight bits of the M value. |

## HFCGM High Value Register (HFCGMH)

The HFCGMH register contains the upper bits of the frequency multiplier M value. Data written to the register is stored in the setup buffer. Reading the register returns the HFCGMH value of the currently set frequency. On $V_{CC}$ power-up, Watchdog reset and Debugger Interface reset, it is loaded with $03_{16}$.

Location: 00 $FFA4_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **HFCGM** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

| Bit | Description |
|---|---|
| 7-0 | **HFCGM.** Contains the upper bits of the M value. |

## HFCGN Value Register (HFCGN)

The HFCGN register is a byte-wide, read/write register containing five bits of the frequency multiplier N value. Data written to the register is stored in the setup buffer. Reading the register returns the HFCGN value of the currently set frequency. On $V_{CC}$ power-up, Watchdog reset and Debugger Interface reset, it is loaded with $08_{16}$.

Location: 00 $FFA6_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | **HFCGN4-0** | | | | |
| Reset | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 4-0 | **HFCGN4-0.** Contains the five bits of the N value. |
| 7-5 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## HFCGI Low Value Register (HFCGIL)

The HFCGIL register contains the lower eight bits of the frequency multiplier I value. Data written to the register is stored in the setup buffer. Reading the register returns the value of its first 8 bits. (IVLID bit in HFCGCTRL1 register indicates if the data is valid.)

Location: 00 $FFA8_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **HFCGI7-0** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **HFCGI7-0.** Contains the lower eight bits of the I value. |

## HFCGI High Value Register (HFCGIH)

The HFCGIH register is a byte-wide, read/write register containing the upper six bits of the frequency multiplier I value. Data written to the register is stored in the setup buffer. Reading the register returns the HFCGIH value of the currently set frequency.

Location: 00 $FFAA_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **HFCGI13-8** | | | | | |

| Bit | Description |
|---|---|
| 5-0 | **HFCGI13-8.** Contains the upper six bits of the I value. |
| 7-6 | **Reserved.** |

## HFCG Pre-Scaler Register (HFCGP)

The HFCGP register is a byte-wide, read/write register. It allows the core clock to be derived from the 96 MHz clock.

On reset HFCGP is initialized to $17_{16}$.

Location: 00 $FFAC_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | **HFCGP4-0** | | | | |
| Reset | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 4-0 | **HFCGP4-0 (Pre-Scaler Divider Value).** The divider of OSCCLK is the number defined by HFCGP(4-0) + 1, e.g., a value of $00_{16}$ results in a divide by 1; a value of $1F_{16}$ results in a divide by 32.<br>Results are undefined when the pre-scaler holds a value that yields a core domain clock frequency higher than 20 MHz or lower than 3 MHz.<br>The pre-scaler is set according to the value held in HFCGP register when LOAD96 bit or FAST96 bit is set (1). Reading HFCGP register returns the last value written to it. |
| 7-5 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## HFCG Control Register 2 (HFCGCTRL2)

The HFCGCTRL2 register sets the frequency multiplier's operating parameters. On $V_{CC}$ power-up, Watchdog reset and Debugger Interface reset, it is initialized to $00_{16}$.

Location: 00 $FFAE_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | **96MON** | **MONERR** | **SCESTP** | **SCESTR** | **SENABLE** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Type | Description |
|---|---|---|
| 0 | RO | **SIO ENABLE (Enable Host Domain Clock).** Provides the status of the SuperI/O enable/disable command. Any data written to this bit is ignored.<br>0: Disabled (default)<br>1: Enabled |
| 1 | R/W1C | **SCESTR (SIO Clock Enable Start).** Indicates that a rising edge was detected on the SENABLE signal. When set, an interrupt is sent to the ICU (level high). Cleared by writing 1 to it. This interrupt may be masked only in the ICU.<br>0: No rising edge on SENABLE was detected (default)<br>1: A rising edge on SENABLE was detected and an interrupt is sent to the ICU |
| 2 | R/W1C | **SCESTP (SIO Clock Enable Stop).** Indicates that a falling edge was detected on the SENABEL signal. When set, an interrupt is sent to the ICU (level high). Cleared by writing 1 to it. This interrupt may be masked only in the ICU.<br>0: No falling edge on SENABLE was detected (default)<br>1: A falling edge on SENABLE was detected and an interrupt is sent to the ICU |
| 3 | R/W1C | **MONERR (Monitor Error).** Indicates that a 48 MHz monitor error was detected during the last Watchdog or Debugger reset process. Note that this bit is cleared by all resets except Watchdog and Debugger Interface resets. Once set, this bit is cleared by writing 1 to it.<br>0: No error detected (default)<br>1: 48 MHz monitor detected an out of range frequency |
| 4 | RO | **96MON (96 MHz Oscillations On).** Indicates the oscillation frequency at which the HFCG is operating.<br>0: HFCG is oscillating as defined by the HFCGM and HFCGN. The core clock is identical to HFCG frequency (default).<br>1: HFCG is oscillating at 96 MHz. The core clock is pre-scaled as defined by HFCGP. |
| 7-5 | | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## 4.19 THE DEBUGGER INTERFACE

The Debugger Interface module links between a debugger running on a host machine and the PC87591L-N05.

The Debugger interface associates a processor number with the processor core. The number associated with the core is 0.

### 4.19.1 Features

- Debugger communication via an IEEE 1149.1b 1994 JTAG serial bus
- Test Access Port (TAP) for JTAG serial bus
- Hardware reset signal assertion by debugger command
- Configurable ABORT event
- Interrupt signal ($\overline{\text{TINT}}$) to the debugger indicates a waiting message
- Waiting message to on-chip processor by non-maskable ISE interrupt
- 8-word (16-byte) Rx (downstream) data link
- 8-word (16-byte) Tx (upstream) data link

### 4.19.2 Structure

The Debugger interface consists of a TAP, ISE Interrupt Control, Rx and Tx data links and a PC87591L-N05 reset-by-command source circuit. Figure 76 shows these functional blocks.

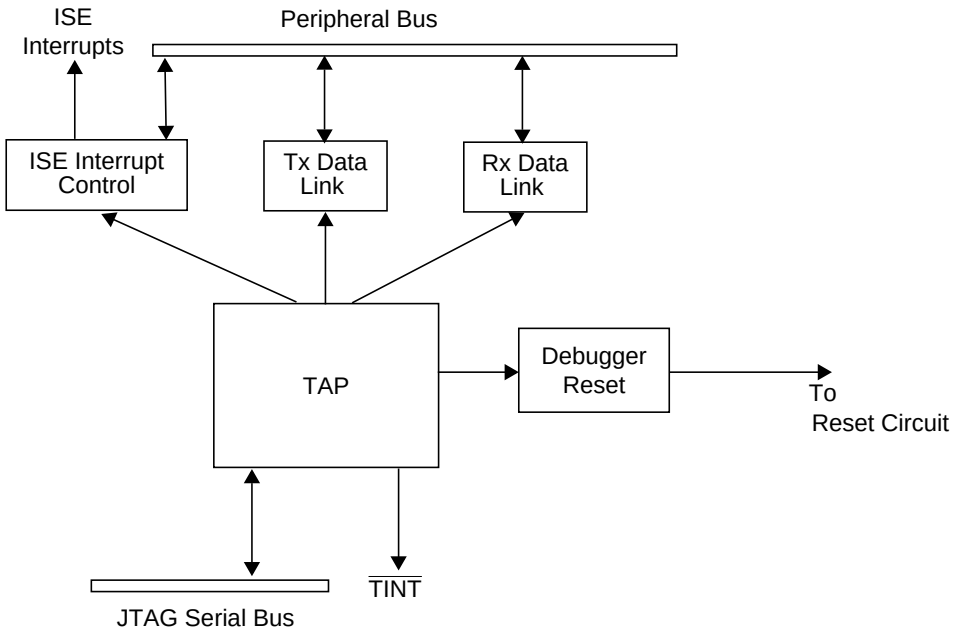



**Figure 76. Debugger Interface**

The TAP block has an IEEE 1149.1b 1994 standard interface to a JTAG serial bus. In addition, it uses $\overline{\text{TINT}}$ to notify the host/debugger that a message is waiting upstream. The TAP and the JTAG form the debugging communication channel and port for the PC87591L-N05. They operate as follows:

- The TAP copies downstream messages (from the debugger to one of the on-chip processors) to the Rx data link, to be read by the relevant processor via the peripheral bus.
- A processor writes upstream messages (from an on-chip processor to its debugger) to the Tx data link for upstream transmission via the JTAG serial bus.
- The ISE interrupt control generates ISE interrupts to the processor in cases of ABORT or a waiting message in the Rx data link.
- The debugger reset circuit generates a PC87591L-N05 reset in response to a debugger reset command. This is in addition to the Power-Up and Hardware reset sources.

# 4.0 Embedded Controller Modules (Continued)

## 4.19.3 Debugger Interface Functional Description

The Debugger interface supports four operating modes: Rx session, Tx session, chip RESET and ABORT. Some of these modes can be active simultaneously for the same or different processors. The ISE interrupt control block includes hooks to control these conditions.

**Rx Session (Sending Data Downstream)**

A message sent downstream by the debugger to a processor is called an Rx session. In an Rx session, a debugger uses both the TAP and the JTAG to monitor the “busy” indication for the Rx data link. Following a “not busy” indication, a message is sent to a processor via the JTAG, TAP and Rx data link.

One of the internal ISE interrupts is asserted (if not active) according to the PID field of the instruction currently loaded into the TAP IR register. The signaled processor accesses the data link via the peripheral bus, reads the message length (optional) and fetches it from the Rx data buffer.

At the end of the data transfer, the processor turns off the “busy” indication of the Rx data link.

**Tx Session (Sending Data Upstream)**

A message sent upstream by a processor to the debugger is called a Tx session. In a Tx session, one of the processors tries to own the Tx data link by accessing the Tx semaphore DBGTXLOC register. If successful, it writes a message body to the Tx data buffer and a message length, in words, to DBGTXST register.

After completion of the data buffer update, the processor sets ASSERT bit in DBGTINT register to 1. This signals the debugger with an active-low pulse on $\overline{\text{TINT}}$. The debugger reads the data using both the TAP and the JTAG.

At the end of the data transfer, the semaphore circuit is set to “not busy” and $\overline{\text{TINT}}$ is released.

**Chip RESET**

The PC87591L-N05 is reset by a dedicated TAP instruction.

**ABORT**

Either a TAP instruction or a bit-set operation in DBGABORT register generates an ABORT. Asserting an ISE interrupt together with a non-zero ABORT_i bit in DBGISESRC register signals an ABORT operation. The ISE interrupt control circuit asserts the ISE interrupt according to the pre-programed mask bits in ABORT_MASK register. A dedicated circuit, together with a set of registers in the ISE Interrupt Control, clears the ISE requests after they have been served.

### Rx Data Link

The Rx data link consists of an 8-word read/write data buffer, DBGRXD0 to DBGRXD7 registers and the Status (DBGRXST) register.

On PC87591L-N05 reset, DBGRXST register is set to its reset value. On TAP reset, the data link maintains its values.

When the TAP controller is in Update-DR state and the current IR is SCAN_RX, DBGRXDX registers are updated from the TAP Data Shift (DBGDATA) register. Data is valid to the processor only while BUSY bit in DBGRXST register is set. In this case (i.e., Update-DR of SCAN_RX), the TAP controller copies the PID and length fields from its IR to the Status register and sets BUSY bit to 1 in this state.

A processor may turn off the BUSY bit by writing 1 to it. BUSY can be cleared even when TCK is not toggling. The Rx data link functions in Active or Idle modes.

DBGDATA length is set to the length field of the SCAN_RX instruction before Capture_DR state (see Section 4.19.6 on page 229). No parallel load is executed in Capture-DR state.

### Tx Data Link

The Tx Data link consists of an 8-word, read/write data buffer, DBGTXD0 to DBGTXD7 registers, a read/write Status register (DBGTXST), a read/write semaphore lock register (DBGTXLOC) and a write-only $\overline{\text{TINT}}$ control register (DBGTINT).

On Power-Up reset, the Tx Data link is reset (negating any pending message), DBGTXLOC and DBGTXST are set to their reset values and $\overline{\text{TINT}}$ is released (1). On Warm and Internal reset, any partial message (i.e., $\overline{\text{TINT}}$=1) is negated by setting DBGTXLOC and DBGTXST to their reset values. Messages that were completed (i.e., $\overline{\text{TINT}}$=0) are maintained for transmission to the host by maintaining DBGTXLOC and DBGTXST values.

On TAP reset, the data link maintain its values.

DBGTXD registers are captured by the TAP data shift register (DBGDATA) in Capture-DR state of the TAP controller when the current Information Register (IR) is SCAN_TX. The DBGDATA length is set dynamically, according to the length field of DBGTXST register before the Capture_DR state of the SCAN_TX operation (see Section 4.19.6 on page 229). The TAP IR register captures the values of PID and MSG_LEN fields of DBGTXST register when the TAP controller is in Capture-IR state. No parallel load is executed in Update-DR state.

## 4.0 Embedded Controller Modules (Continued)

The semaphore is implemented by DBGTXLOC register. A write operation, to PID field of this register, of a value other than '1111' changes the contents of this field only when the PID field equals '1111'. This field returns to '1111' in one of the following ways:

- Write operation of '1111' to PID field of DBGTXLOC register from the peripheral bus while $\overline{TINT}$ is not asserted
- Update-DR state of the TAP controller when the current instruction loaded into TAP IR register is SCAN_TX (this action take place at the rising edge of TCK)
- On Warm or Internal reset if $\overline{TINT}$=1 and on Power-Up reset

A processor can access DBGTXDi registers in Active mode only. For a processor to gain ownership of the Tx link, it must capture the DBGTXLOC semaphore using the following sequence:

1. Verify that the value of PID in DBGTXLOC register is '1111'.
2. Write the processor PID code to DBGTXLOC register.
3. Read DBGTXLOC. If the PID field is equal to the value that was written, the data link is granted. If not, repeat step 1.

A processor should access DBGTXDi and DBGTXST registers only after successfully gaining ownership over the Tx link by using the above sequence.

The $\overline{TINT}$ signal is an active-low pulse asserted by the Tx data link when ASSERT bit in DBGTINT register is written with 1. It is de-asserted, together with the semaphore indication in Update-DR state of the TAP controller, when the current instruction loaded into TAP IR register is SCAN_TX.

Access to DBGTXLOC, DBGTXST and DBGTXDi registers should be done only while $\overline{TINT}$ is not asserted. $\overline{TINT}$ negation can be identified by the release of PID ('1111').

### Debugger Reset Circuit

Chip reset is asserted in the Update-DR state of the TAP controller when the current instruction is ASSERT_DBG_RST. This triggers a Debugger reset, as described in "ASSERT_DBG_RST" on page 229. This circuit is functional in Active or Idle modes. It is functional, while TCK is not toggling, one cycle after exit from Update-DR state.

### ISE Interrupt Control

The ISE interrupt control module sends ISE interrupt requests to the processor core. It consists of the write-only DBGABORT register, the read/write DBGISESRC register, and the DBGMASKS shift register.

The DBGISESRC register is cleared on PC87591L-N05 reset. During TAP reset, the values of these registers are maintained. The DBGMASKS value is modified only in Update-DR state of the TAP controller when the current instruction is SCAN_ABORT_MASK.

The ISE interrupt control module issues an ISE interrupt request to a specific processor or multiple processors, together with the matched bit in DBGISESRC register, according to the MESSAGE or ABORT event.

In a MESSAGE event, an ISE interrupt is requested for a specific processor according to the PID field of the SCAN_RX instruction. The request is issued (together with DBGISESRC bit assertion) if the current instruction loaded in TAP IR is SCAN_RX and the TAP controller is in Update-DR state (rising edge of TCK).

An ABORT event occurs when SCAN_RX is executed with a PID of all 1s (ISE and ABORT_i bits in DBGISESRC register are asserted) or when processor bits P_i are set in DBGABORT register by one of the processors. That is, ISE and DBGISESRC bits are asserted with the write operation itself; if DBGABORT is written with some 0 bits, the PIDs related to these bits do not get the ISE. In an ABORT event, the assertion of each ISE interrupt and DBGISESRC bit depends on its masking bit in DBGMASKS register.

Each ISE interrupt is cleared when ABORT_i and RX_i in DBGISESRC register are both 0. Any bit in DBGISESRC register is cleared by writing 1 to it (writing 0 is ignored). If there is a new source activity in the same write cycle during which DBGISESRC register of a specific processor is cleared, the ISE interrupt control asserts ISE again, together with its source bit.

The ISE interrupt is an active-high pulse. For nested ISE sources, the ISE remains asserted, and the ISE interrupt control module sets the new source bit to 1.

The DBGMASKS register is not available to the peripheral bus; it is accessed only from the JTAG serial bus when the SCAN_ABORT_MASK instruction is loaded into the TAP IR.

The ISE signal, together with its source bit, should be asserted in Active or Idle modes for wake-up purposes when the source is a debugger message or debugger abort. Other functionality should be consistent while changing modes (i.e., during wake-up). Full functionality of this module is maintained even when TCK is not toggling.

# 4.0 Embedded Controller Modules (Continued)

## Clock Synchronization

Some operations are related to TCK, others to the PC87591L-N05 main clock and yet others are asynchronous. The PC87591L-N05 logic guarantees correct operation and a meta-stable protected interface in all its operating modes (i.e., Active and Idle).

The core may access the Debugger interface registers only in Active mode.

### 4.19.4 Test Access Port (TAP)

This section defines the top-level design of the TAP, as shown in Figure 77. Subsequent sections provide detailed TAP design requirements.

The TAP includes the following elements:

- TAP signals
- TAP controller
- Instruction Register (IR)
- Data registers.

The Instruction and data registers have separate shift register-based paths connected in parallel. These registers have a common serial data input and a common serial data output connected to the TAP TDI and TDO signals, respectively.

The TAP controller selects between TDI and TDO as the alternative instruction and data register paths.

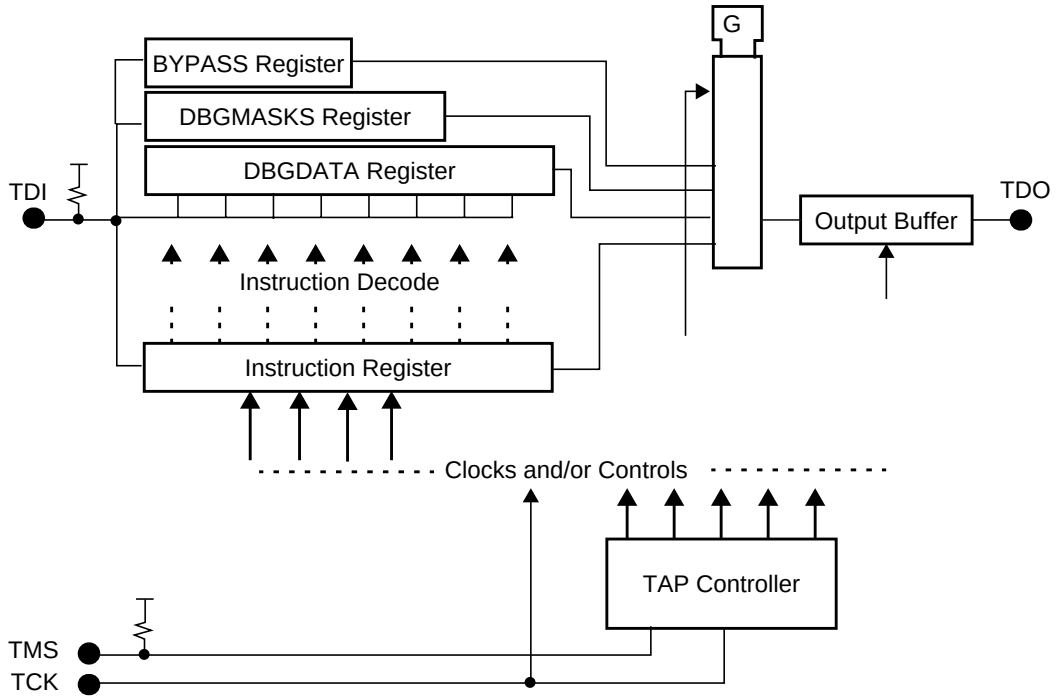



**Figure 77. TAP Block Diagram**

### Further Information

The TAP is based on the test logic in the IEEE 1149.1b-1994 specification. However, since its purpose is to facilitate communication, it does *not* support IEEE 1149.1b-1194 testing facilities. It is used here to benefit from off-the-shelf bus controller cards and software and potential future enhancements to the test scheme.

This document includes the relevant rules of this specification. See the following documents for further information.

- For further details and examples of the standard, see *IEEE Standard Test Access Port and Boundary-Scan Architecture, May 21, 1990*.
- For further details of test bus chips and equipment, see the relevant manufacturer datasheets and application notes, e.g., *SCANTM Data book, National Semiconductor 400102.*
- For technical background, refer to text books on the subject, for example, *Colin M. Maunder and Rodham E. Tulloss, "The Test Access Port and Boundary-Scan Architecture", IEEE Computer Society Press Tutorial*.

## 4.0 Embedded Controller Modules (Continued)

### TAP Signals

The TAP interface signals to the JTAG serial bus are: Test Clock Input (TCK), Test Mode Select Input (TMS), Test Data Input (TDI) and Test Data Output (TDO). The TAP controller is reset at Power-Up reset only; see Section 3.2 on page 61.

**TCK.** TCK provides the clock for the JTAG serial bus and the TAP controller. Stored-state devices in the TAP maintain their state indefinitely after the signal applied to TCK is stopped.

**TMS.** The TAP controller next state is set by the TMS value and its current state. The TAP samples the signal presented at TMS on the rising edge of TCK.

Circuitry, fed from TMS, produces the same response to the application of a logic 1 as for a non-driven input.

**TDI.** This is the data and instructions serial input. The TAP samples the signal presented at TDI on the rising edge of TCK.

Circuitry fed from TDI produces the same response to the application of a logic 1 as for a non-driven input.

When data is shifted from TDI towards TDO, test data received at TDI appears without inversion at TDO following a number of rising and falling edges of TCK. This is determined by the length of the instruction or Test Data register selected.

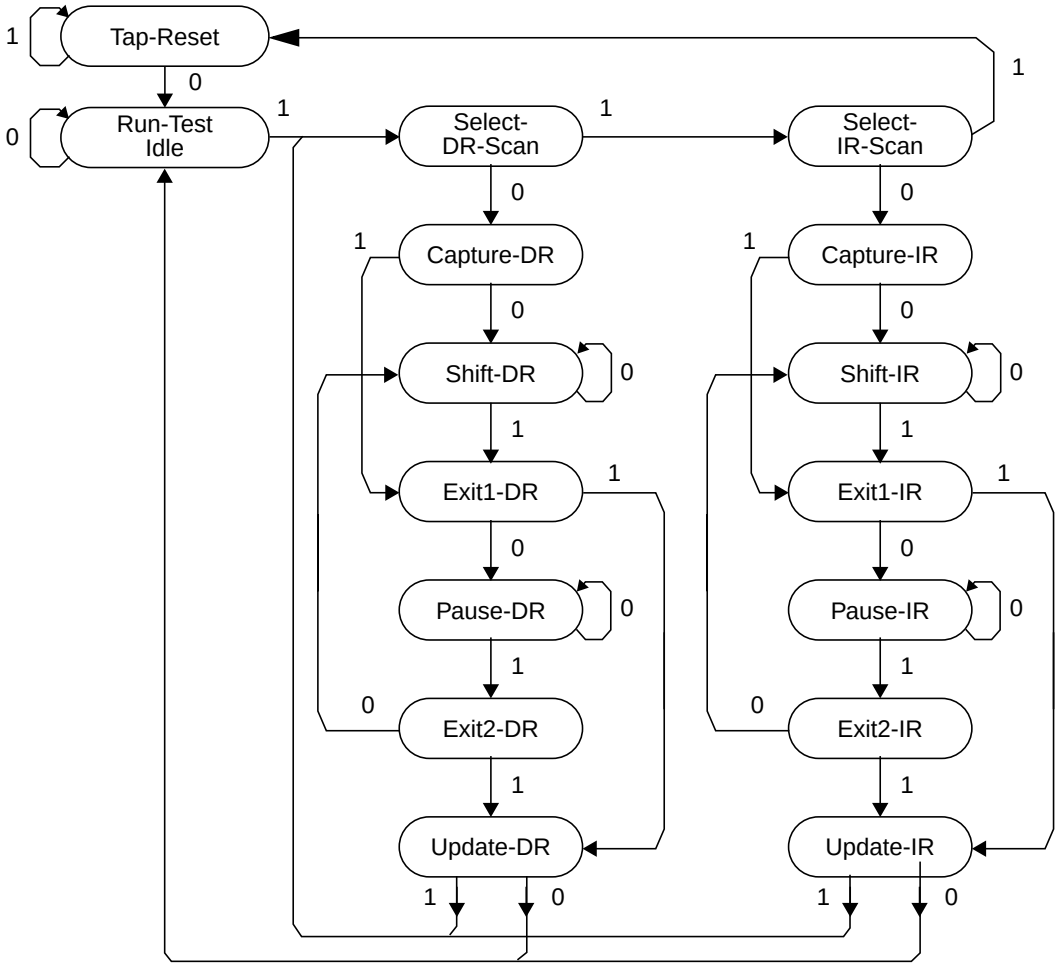
**TDO.** This is the data and instructions serial output. The signal driven through TDO changes its state only after the falling edge of TCK.

The TDO driver is set to its inactive drive state, except while data or an instruction is being scanned.

### TAP Controller

The TAP controller is a synchronous, finite state machine that responds to changes in the TMS and TCK signals and controls the sequence of operations of the PC87591L-N05 reset, ISE interrupt control and data link circuitry.

Figure 78 shows the TAP controller state diagram.




**Note:** The value for each state transition represents the signal on TMS for a rising edge at TCK.

**Figure 78. TAP Controller State Diagram**

## 4.0 Embedded Controller Modules (Continued)

For a detailed description of the controller states, see the IEEE 1149.1b-1994 specifications.

All state transitions of the TAP controller occur based on the value of TMS on the rising edge of TCK. TAP actions (described in Sections 2.1.3 to 2.1.7) occur either on the rising or falling edge of TCK in each controller state, as shown in Figure 79.

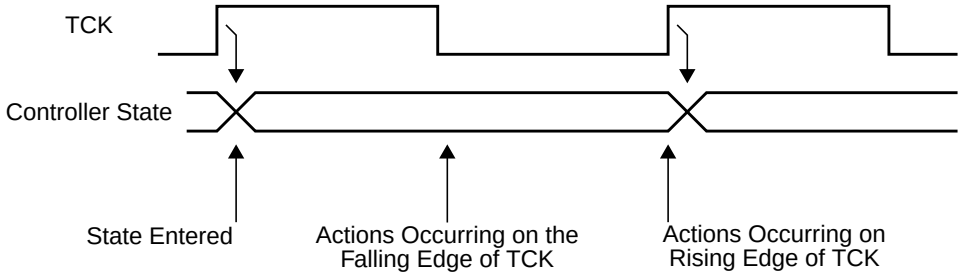



**Figure 79. Timing of Actions in a Controller State**

### TAP Controller Operation

The TAP controller changes state only in response to the following events:

- A rising edge of TCK
- Power-up

The TAP controller generates signals to control the operation of the TAP registers and associated PC87591L-N05 reset, ISE interrupt control and data link circuitry.

The TDO output buffer and the circuitry that selects the register output fed to TDO are controlled as shown in Table 27. Changes at TDO, as defined in Table 27, occur on the falling edge of TCK after entry into the state.

**Table 27. TAP Operation in Each Controller State**

| Controller State | Register Selected to Drive TDO | TDO Driver Level |
|---|---|---|
| Tap-Reset | Undefined | Inactive |
| Run-Test-Idle | Undefined | Inactive |
| Select-DR-Scan | Undefined | Inactive |
| Select-IR-Scan | Undefined | Inactive |
| Capture-IR | Undefined | Inactive |
| Shift-IR | Instruction | Active |
| Exit1-IR | Undefined | Inactive |
| Pause-IR | Undefined | Inactive |
| Exit2-IR | Undefined | Inactive |
| Update-IR | Undefined | Inactive |
| Capture-DR | Undefined | Inactive |
| Shift-DR | Test Data | Active |
| Exit-DR | Undefined | Inactive |
| Pause-DR | Undefined | Inactive |
| Exit2-DR | Undefined | Inactive |
| Update-DR | Undefined | Inactive |

### TAP Controller Initialization

Circuitry built into the PC87591L-N05 forces the TAP controller into TAP-Reset state at power-up in an asynchronous manner.

The TAP controller must not be initialized by operation of any system input, such as a system reset.

# 4.0 Embedded Controller Modules (Continued)

When TMS is equal to 1 for five consecutive TCK cycles, this forces the controller into TAP-Reset state from any state.

### 4.19.5 TAP Instruction Register

The Instruction Register (IR) allows an instruction to be shifted into the Debugger interface. The instruction selects the mode of operation, the data register to be addressed or both.

In any serial scan operation, the JTAG serial bus controller transmits and receive vectors of the same length. This serial scan chain property is used to transmit status bits from the device to the bus controller in any instruction scan operation. The Instruction register in the PC87591L-N05 allows status information generated within the data links to be examined as shown in Table 28 on page 227.

### Design and Construction

The IR uses a shift register-based design with a parallel input for register cells other than the two nearest to the serial output. An instruction, shifted into the register, is latched at the completion of the shifting process.

The IR includes 12 shift register-based cells capable of holding instruction data.

An instruction that is shifted into the register is latched so that the related changes occur only in the Update IR and TAP-Reset states.

Data is not inverted between the serial input and serial output of IR.

The IR parallel input status bits, loaded at the Capture IR state of any instruction scan operation, are shown below.

| Bit | 11 7 | 6 | 5 | 5 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **MSG_LEN** | **PID** | | | | **RX_BUSY** | **0** | **1** |
| Reset | | | | | | | | |

| Bit | Description |
|---|---|
| 0 | Fixed value (1). |
| 1 | Fixed value (0). |
| 2 | **RX_BUSY (Receive Busy).** Busy indication from the Rx data link. |
| 6-3 | **PID (Processor ID).** Contains the processor ID from the Tx data link.<br>When $\overline{\text{TINT}}$ is set to 1 (not active), PID field is '1111', indicating that the Tx link has no valid data.<br>When $\overline{\text{TINT}}$ is set to 0 (active), PID field indicates the value of PID field in DBGTXLOC register. Note that this value is latched when $\overline{\text{TINT}}$ becomes active and is held until the Rx data is read by the host. |
| 11-7 | **MSG_LEN (Message Length).** Contains the Tx message length in words. When the Tx status word PID field is not '1111', the MSG_LEN holds a copy of the MSG_LEN field in DBGTXST register. When the size of this field is less than five bits, the MSG_LEN MSBs in the status word are forced to 0. |

#### Instruction Register Operation

Table 28 shows the behavior of IR in each TAP controller state.

All actions resulting from an instruction terminate when a new instruction is transferred to the parallel output of the IR (i.e., in Update-IR or TAP-Reset states).

All operations of the shift register stages occur on the rising edge of TCK after entry into a TAP controller state.

**Table 28. Instruction Register Operation in Each Controller State**

| Controller State | Shift Register Stage | Parallel Output |
|---|---|---|
| TAP-Reset | Undefined | Set to give BYPASS instruction |
| Capture-IR | Load status data (table 2-2) | Retain last state |
| Shift-IR | Shift towards serial output | Retain last state |
| Exit1-IR<br>Exit2-IR<br>Pause-IR | Retain last state | Retain last state |
| Update-IR | Retain last state | Load from shift register stage |

## 4.0 Embedded Controller Modules (Continued)

| Controller State | Shift Register Stage | Parallel Output |
|---|---|---|
| All other states | Undefined | Retain last state |

The data present at the parallel output of the IR is latched from the shift register stage on the falling edge of TCK in Update-IR state.

After entry into TAP-Reset state as a result of TAP controller clocked operation, the BYPASS instruction is latched onto the IR output on the falling edge of TCK.

### Instructions

Instructions are entered serially into the Debugger interface logic during an IR scan by using the IR. See Table 29 for the required binary codes.

**Table 29. IR Instruction Binary Codes**

| D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Instruction |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x | x | x | x | x | x | x | x | x | 1 | 1 | 1 | BYPASS |
| x | x | x | x | x | x | x | x | x | 0 | 0 | 0 | BYPASS (Reserved for Scan) |
| x | x | x | x | x | x | x | x | x | 1 | 1 | 0 | BYPASS |
| other | | | | | | | | | 0 | 0 | 1 | BYPASS |
| L4 | L3 | L2 | L1 | L0 | PID3 | PID2 | PID1 | PID0 | 0 | 1 | 0 | SCAN_RX |
| x | x | x | x | x | x | x | x | x | 0 | 1 | 1 | SCAN_TX |
| x | x | x | x | x | x | x | x | x | 1 | 0 | 1 | SCAN_ABORT_MASK |
| x | x | x | x | x | x | x | x | x | 1 | 0 | 0 | ASSERT_DBG_RST |

**Notes:**
- D0 is the nearest to the serial output.
- "X" means ignore.
- Debugger abort is generated by SCAN_RX when PID='1111'.

Data registers not selected by the current instruction do not interfere with the operation of the on-chip system logic or with the selected data registers.

Each instruction enables a single serial data register path to shift data between TDI and TDO in Shift-DR state, as shown in Table 30.

Instruction codes that are not required to control test logic are equivalent to the BYPASS instruction.

#### BYPASS

The BYPASS instruction operates the BYPASS register. This register contains a single shift register stage and provides a minimum-length serial path between the TDI and TDO pins of a component when no test operation is needed for that component. This allows more rapid movement of test data to and from other board components that are required to perform test operations.

The BYPASS instruction selects BYPASS register to be connected for serial access between TDI and TDO in the Shift_DR state.

If the BYPASS instruction is selected, all other data registers continue their normal functionality.

### Debugger Interface Instructions

#### SCAN_RX

The SCAN_RX instruction switches the data scan path to DBGDATA register. The DBGDATA length is set to L0 to L4; see "Debug Data Register (DBGDATA)" on page 230. The result of the Capture-DR state of the TAP controller is unpredictable. A parallel load of data from DBGDATA register to the DBGRXD Rx data buffer is done in Update-DR state.

The controller sets the RX_BUSY indication in DBGRXST register to 1 in Update-DR state. The PID and message fields of the SCAN_RX instruction are available for read access, through the peripheral bus, from DBGRXST register. The ISE interrupt control block asserts the ISE interrupt and RX_i bit in DBGISESRC register, according to the PID index.

#### DEBUGGER ABORT

This operation has no dedicated operation code. It is performed using the SCAN_RX instruction with the PID field is '1111'. Following SCAN_RX mode, the ISE interrupt control block asserts ISE interrupts, together with ABORT_i bits in DBGISESRC register, according to the MASKS values in DBGMASKS register. The assertion is triggered during the TCK rising edge during Update-IR state. In this case, there is no RX_BUSY indication and no change in the contents of DBGRXST register (i.e., this format of SCAN_RX may be issued with a busy Rx data link).

**SCAN_TX**

The SCAN_TX instruction switches the data scan path to DBGDATA register. In Capture-DR state, DBGDATA register captures the values from the Tx data buffer DBGTXD. No parallel load is performed in Update-DR state. The length of DBGDATA register is set to the value of MSG_LEN in DBGTXST register (see "Debug Data Register (DBGDATA)" on page 230). No parallel load is performed in Update-DR state.

The Controller sets DBGTXLOC to all 1s and releases $\overline{\text{TINT}}$ on Update-DR state. This operation clears the semaphore and enables the data link for a new transaction. $\overline{\text{TINT}}$ is asserted, as described in "Tx Data Link" on page 222.

**SCAN_ABORT_MASK**

The SCAN_ABORT_MASK instruction switches the data scan path to ABORT_MASK register. In Capture-DR state, a parallel update of the shift register occurs with the DBGMASKS values. Parallel load is performed in Update-DR state to update the DBGMASKS values.

**ASSERT_DBG_RST**

A Debugger reset is asserted in Update-DR state. Serial data is switched to the BYPASS register.

### 4.19.6 TAP Data Registers, Debugger Interface

### Bit Arrangement and Mapping

Figure 80 shows the bit allocation for a parallel load operation between any pair of serial shift registers and a peripheral bus addressable register.

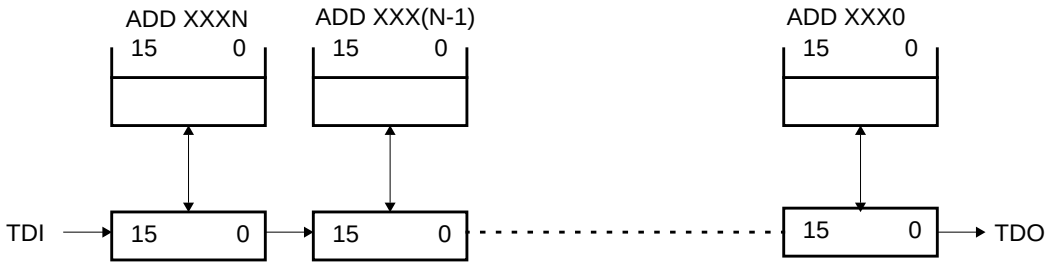



**Figure 80. Bit Allocation Arrangement**

### Functionality in Various TAP Controller States

When data is shifted through a data register, data applied to TDI appears without inversion at TDO following an appropriate number of TCK transitions, when the TAP controller is in Shift-DR state.

The data register connected between TDI and TDO shifts data one stage towards TDO after each rising edge of TCK in Shift-DR state.

In TAP-Reset state, all data registers either perform their system function (if one exists) or do not interfere with the operation of the on-chip system logic.

If, in response to the current instruction, a data register loads data from a parallel input, the data is loaded on the rising edge of TCK following entry into Capture-DR state.

If the data register connected between TDI and TDO in response to the current instruction is provided with latched parallel data outputs, the data is latched into the parallel output buffers on the falling edge of TCK, during Update-DR or Run-Test/Idle states, as appropriate.

If, in response to a current instruction, no operation of a selected data register is required in a given controller state, the register retains its last state unchanged.

When the TAP controller state machine is in TAP-Reset state during Power-Up reset, IR register is reset.

Data registers that are not selected by the current instruction are set to perform their functions, as described in Section 4.19; specifically, see "Tx Data Link" on page 222, "Debugger Reset Circuit" on page 223 and "ISE Interrupt Control" on page 223.

**Table 30. Data Register Operation in Each Controller State**

| Controller State | Action |
|---|---|
| Capture-DR | Load data at parallel input into shift register stage. Parallel output registers, or latch, retains last state. |
| Shift-DR | Shift data towards serial output. Parallel output register, or latch where provided, retains state. |

## 4.0 Embedded Controller Modules (Continued)

| Controller State | Action |
|---|---|
| Exit1-DR<br>Exit2-DR<br>Pause-DR | Retain last state. |
| Update-DR | Load parallel output register or latch from the shift register stage. Shift register stage retains state. |
| All other controller states | Registers that have a parallel output maintain the last state of this output; otherwise undefined. |

### Debug Bypass Register (BYPASS)

The BYPASS register provides a minimum length serial path for data movement between TDI and TDO. This path can be selected when no other data register needs to be accessed.

The BYPASS register consists of a single shift register stage.

When the current instruction selects the BYPASS register for inclusion in the serial path between TDI and TDO, the shift register stage is set to 0 on the rising edge of TCK following entry into Capture-DR state. The BYPASS register is accessed from the JTAG serial bus only.

### Debug Data Register (DBGDATA)

This register is the shift register element of DBGRXD and DBGTXD. It is accessed from the JTAG serial bus only. Figure 81 shows the parallel load data scheme.

The DBGDATA register consists of shift register stages according to the data buffer length. The actual length of the data value, in a scan operation, is set before Capture-DR state when the current instruction is SCAN_RX or SCAN_TX. In the first case, it is set according to SCAN_RX L0 to L4; in the latter case, it is set according to the value of MSG_LEN in DBGTXST register.

The actual length of the register is 16*(length+1), where “length” is the binary positive number created by the length field (with L4 as MSB).

A length longer than the maximum data buffer is mapped to maximum length. A length shorter than the maximum data buffer results in loading the data to/from the smallest addresses in the data buffer. Maximum length for this design is 128 shift register stages. Note that TDO is always fixed; the TDI “insertion-point” changes according to the actual length.

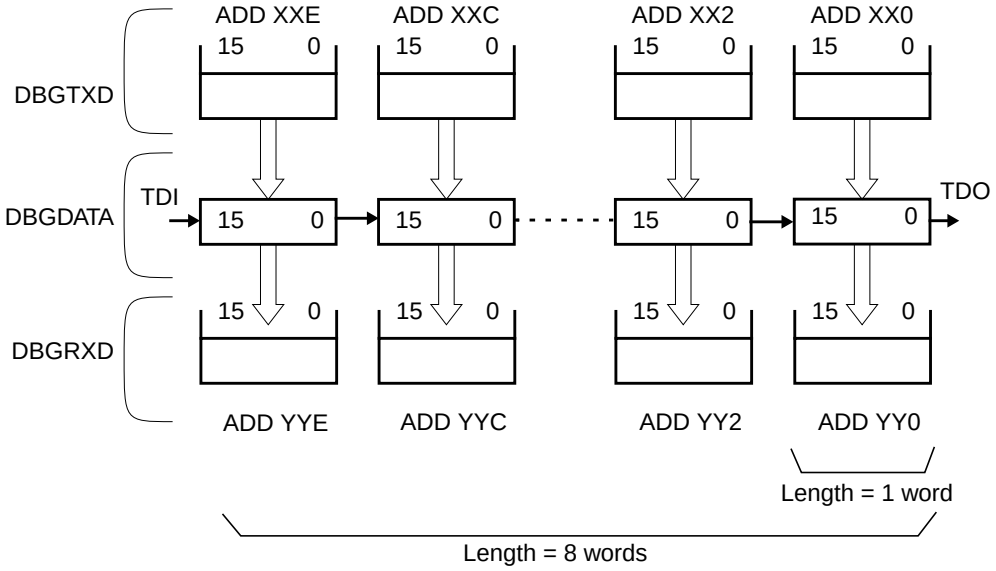



**Figure 81. DBGDATA Connection to the Data Links**

### Debug Abort Mask Register (DBGMASKS)

This is a serial shift register, with parallel output and parallel input (for read and write of ABORT_MASK register), accessed by the JTAG serial bus while the SCAN_ABORT_MASK instruction is set to TAP IR. The register is not addressable from the peripheral bus. The non-reserved bits of ABORT_MASK register are preset to 1 on Power-Up reset.

## 4.0 Embedded Controller Modules (Continued)

The ABORT_MASK register consists of shift register stages with 16 bits. Bits 1 to 15 are reserved and should be written with 0. A bit value of 1 enables the processor to abort; a bit value of 0 disables it.

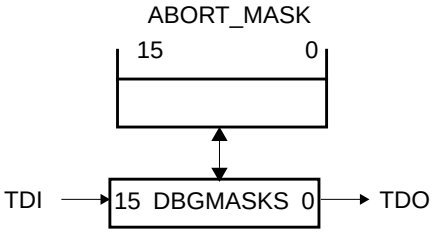



**Figure 82. ABORT_MASK and DBGMASKS Register Interaction**

### 4.19.7 Core Registers, Debugger Interface

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### Core Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| DBGRXD0-14 | Debug Receive Data Registers 0, 2, 4, 6, 8, 10, 12 and 14 | RO |
| DBGRXST | Debug Receive Status Register | Varies per bit |
| DBGTXD0-14 | Debug Transmit Data Registers 0, 2, 4, 6, 8, 10, 12 and 14 | R/W |
| DBGTXLOC | Debug Transmit Lock Register | R/W |
| DBGTXST | Debug Transmit Status Register | R/W |
| DBGTINT | Debug TINT Assert Register | WO |
| DBGABORT | Debug Abort Generate Register | WO |
| DBGISESRCA | Debug ISE Source Register A | R/W |

### Debug Receive Data Registers 0, 2, 4, 6, 8, 10, 12 and 14 (DBGRXD0-14)

DBGRXD0 to DBGRXD14 is a group of eight word-wide read-only registers. The DBGRXD0-14 registers are written from the JTAG serial bus. They can only be read from the peripheral bus. A representative DBGRXDi register is shown below.

Location: Channel 0 - 00 $FDC0_{16}$
Channel 2 - 00 $FDC2_{16}$
Channel 4 - 00 $FDC4_{16}$
Channel 6 - 00 $FDC6_{16}$
Channel 8 - 00 $FDC8_{16}$
Channel 10 - 00 $FDCA_{16}$
Channel 12 - 00 $FDCC_{16}$
Channel 14 - 00 $FDCE_{16}$

Type: RO

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **RX_DATAi** | | | | | | | | | | | | | | | |

| Bit | Description |
|---|---|
| 15-0 | **Receive Data (RX_DATA).** Data bits 16*i through 16*i+15 of the Rx data link data buffer. |

# 4.0 Embedded Controller Modules (Continued)

## Debug Receive Status Register (DBGRXST)

DBGRXST is a byte-wide register updated by the TAP controller to reflect the PID and MSG_LEN fields of the SCAN_RX instruction. Bits 1 to 7 of this register are read-only bits; data written to them is ignored. The register format is shown below.

Location: 00 $FDE0_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **MSG_LEN** | | | **PID** | | | | **RX_BUSY** |

| Bit | Type | Description |
|---|---|---|
| 0 | R/W | **RX_BUSY (Receive Busy).** Data link busy indication. This bit can be cleared by writing 1 to it. Writing 0 is ignored.<br>0: Not Busy<br>1: Busy |
| 4-1 | RO | **PID (Processor Index).** A write operation to this field is ignored.<br>**Bit Value (Decimal) Description**<br>0: Processor ID index<br>1-14: Invalid<br>15: All processors abort |
| 7-5 | RO | **MSG_LEN.** The message length equals (MSG_LEN+1). Write operations to this field are ignored. |

## Debug Transmit Data Registers 0, 2, 4, 6, 8, 10, 12 and 14 (DBGTXD0-14)

DBGTXD0 to DBGTXD14 is a group of eight word-wide read/write registers. The DBGTXD0-14 registers are written by a processor from the peripheral bus. A representative DBGTXDi register is shown below.

Location: Channel 0 - 00 $FDD0_{16}$
Channel 2 - 00 $FDD2_{16}$
Channel 4 - 00 $FDD4_{16}$
Channel 6 - 00 $FDD6_{16}$
Channel 8 - 00 $FDD8_{16}$
Channel 10 - 00 $FDDA_{16}$
Channel 12 - 00 $FDDC_{16}$
Channel 14 - 00 $FDDE_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **TX_DATAi** | | | | | | | | | | | | | | | |

| Bit | Description |
|---|---|
| 15-0 | **TX_DATAi (Transmit Data i).** Data bits i*16 through i*16+15 of the transmit buffer. |

# 4.0 Embedded Controller Modules (Continued)

## Debug Transmit Lock Register (DBGTXLOC)

DBGTXLOC is a byte-wide read/write register. The PID field of this register is used as a semaphore for locking the Tx channel for a specific processor. If $\overline{TINT}$ is not active (1), this register is loaded with $F_{16}$ on Warm or Internal reset. If $\overline{TINT}$ is active (0), it is not affected by a Warm or Internal reset. At Power-Up reset, the register is always loaded with $0F_{16}$. The register format is shown below.

Location: 00 $FDE4_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | **PID** | | | |
| Reset | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 3-0 | **PID (Processor Index).** Indicates which processor currently has control over the Tx channel. When $\overline{TINT}$ becomes active, the value of PID is locked. On the Update_DR state of SCAN_TX instruction, PID is reset to $F_{16}$ after the host reads the data.<br>$0_{16}$-$E_{16}$: Processor ID index. When the PID field holds any of these values, write operations of values other than $F_{16}$ are ignored.<br>$F_{16}$: Semaphore free indication. When the PID field holds this value, write operations may capture the Tx for processor use. |
| 7-4 | **Reserved.** |

## Debug Transmit Status Register (DBGTXST)

DBGTXST is a byte-wide read/write register. This register is written by a processor to indicate the message length, which is used by the Tx data link to set the length of the serial shift register and by the processor software to define the message length parameter. While $\overline{TINT}$ is inactive, this register may be written to at any time. While $\overline{TINT}$ is active, the contents of this register are locked and can be read by the host via the TAP controller status word MSG_LEN field. While $\overline{TINT}$ is inactive, DBGTXST is cleared on reset. While $\overline{TINT}$ is active, only Power-Up reset clears DBGTXST. The register format is shown below.

Location: 00 $FDE2_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | **MSG_LEN** | | |
| Bit | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## Debug TINT Assert Register (DBGTINT)

This is a byte-wide write-only register used to assert $\overline{TINT}$. The register format is shown below.

Location: 00 $FDE6_{16}$

Type: WO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | | **ASSERT** |

| Bit | Description |
|---|---|
| 0 | **ASSERT (Assert $\overline{TINT}$ Control).** Writing 1 to this bit asserts the $\overline{TINT}$ output. $\overline{TINT}$ is de-asserted during Update_DR state of SCAN_TX or during Power-Up reset. |
| 7-1 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## Debug Abort Generate Register (DBGABORT)

DBGABORT is a word-wide write-only register, used to generate an ABORT. A processor may generate an ABORT to a processor with a PID index of i by modifying its P_i bit to 1. Writing 0 to P_i does not result in ISE assertion. The register format is shown below.

Location: 00 $FDE8_{16}$

Type: WO

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | | | **Res** | **Res** | **Res** | **Res** | **Res** | **Res** | **Res** | **P_0** |

| Bit | Type | Description |
|---|---|---|
| 0 | WO | **P_0.** The ABORT source activation.<br>0: Processor ID i does not get an ABORT<br>1: Processor ID i gets an ABORT |
| 15-1 | | **Reserved.** Reserved for future expansion. |

## Debug ISE Source Registers A (DBGISESRCA)

DBGISESRCA is a word-wide read/write register that indicates the ISE sources. The register is cleared on reset. Writing 1 to a bit in this register clears it. Writing 0 to a bit does not change its value. The DBGISESRCA register format is shown below.

Location: 00 $FDEA_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | | | | | | | | | **ABORT_0** | **RX_0** |
| Reset | | | 0 | | 0 | | 0 | | 0 | | 0 | | 0 | | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **RX_i.** The ISE source is the Rx data link. Turn on with ISE assertion. Turn off by writing a byte to DBGISESRC containing 1 in the RX_i bit location.<br>0: Not an ISE source (default)<br>1: The ISE source is the Rx data link |
| 1 | **ABORT_i.** The ISE source is an ABORT request by the debugger or one of the processors. This bit is set by either an abort command from the debugger (SCAN_RX with PID='1111') or by a write of a 1 to the processor's respective bit in DBGISECA register.<br>0: Not an ISE source (default)<br>1: The ISE source is an ABORT event |
| 15-2 | **Reserved.** These bits are reserved for future expansion. |

# 4.0 Embedded Controller Modules (Continued)

## 4.19.8 Usage Hints

### JTAG Performance

For best performance, the PC87591L-N05 should be the only device in the scan path. Since the Debugger interface is used in the development phase of the product, this should not be a problem. Keep the scan path as short as possible.

### JTAG Clock Rate

For reliable communication over the JTAG serial bus, the error probability should be extremely low. This can be achieved by connecting the TAP to the JTAG bus controller card, paying attention to the TCK frequency, signal timing, this cable type and cable length (for long cables, transceivers may be required).

### Communication Lockout

Some erroneous operations by the host may cause a lockout of the communication hardware. Examples of such cases are:

- Performing a data scan when the JTAG instruction is SCAN_RX with a PID value of ‘1111’
- Performing a data scan with a JTAG instruction with a PID value that is not in the system (i.e., PID > 0)
- Performing a SCAN_TX when the PID value is ‘1111’

A reset instruction is guaranteed to recover from all of these locked cases, but it resets the entire PC87591L-N05.

# 4.0 Embedded Controller Modules (Continued)

## 4.20 DEVELOPMENT SYSTEM SUPPORT

The PC87591L-N05 supports code development and debug in the On Board Development (OBD) and Development (DEV) environments. OBD environment is used for debug of the code in the final production board. DEV environment is used in Application Development Boards (ADBs) and ISE systems.

### 4.20.1 Features

- ISE “clipping-on” support via a TRI-STATE strap input (TRIS)

Features available in OBD and DEV environments:

- Debugger interface via the JTAG based Debugger Interface module
- Internal ISE interrupt generation by the Debugger Interface module
- Internal Reset generation by the Debugger Interface module
- Ability to prevent real-time events from interfering with operation of ADB monitor
- Core-integrated hardware breakpoint

Features available in DEV environment:

- Optional use of break line input signal
- Status Information to trace internal activities and implement debug features such as hardware breakpoint and traces
- Use of SRAM in the ADB for fast download of code during development

### 4.20.2 The ISE Interrupt

The core ISE interrupt is an edge-triggered, non-maskable interrupt that is triggered on the rising edge of the Debugger interface output. The ISE interrupt is asserted by the Debugger Interface module for RX or ABORT events to the core.

The ISE interrupt is enabled in DEV and OBD environments when ON bit in DBGCFG register is set to 1; otherwise, it is held inactive.

### 4.20.3 Break Line and Reset Output Interrupt

The $\overline{\text{BRKL}}$_$\overline{\text{RSTO}}$ signal is available in DEV environment. It has two functions: $\overline{\text{BRKL}}$ interrupt input and $\overline{\text{RSTO}}$ reset indication. Multiplexing between the two functions is done based on bus activity as defined below.

#### $\overline{\text{BRKL}}$ Function

The $\overline{\text{BRKL}}$ interrupt input is enabled in DEV environment when ON and BRKLE in DBGCFG register are both set (1); otherwise, it is held inactive.

When $\overline{\text{BRKL}}$ is active during an instruction fetch, it indicates to the core a request to break on the execution of the instruction (if the instruction is to be executed). This enables implementing multiple hardware breakpoints using external hardware.

#### $\overline{\text{RSTO}}$ Function

$\overline{\text{RSTO}}$ is a pulse output that is driven low when the PC87591L-N05 is in reset due to any of its sources (i.e., any reset to the core). This output may be used to set any required defaults in the development system.

#### $\overline{\text{RSTO}}$ and $\overline{\text{BRKL}}$ Selection

The $\overline{\text{BRKL}}$_$\overline{\text{RSTO}}$ signal serves as input to the PC87591L-N05 whenever there is activity on the bus, i.e., while either $\overline{\text{SELIO}}$, $\overline{\text{SEL0}}$, or $\overline{\text{SEL12}}$ are active. If $\overline{\text{SELIO}}$, $\overline{\text{SEL0}}$, or $\overline{\text{SEL12}}$ are inactive, the system must stop driving the $\overline{\text{BRKL}}$_$\overline{\text{RSTO}}$ signal and hold it high using a pull-up resistor.

When there is no activity on the bus, $\overline{\text{BRKL}}$_$\overline{\text{RSTO}}$ may serve as output. It is driven low during an internal reset. There may be a delay from reset start to the signal being driven low, but it is guaranteed to be low for at least three CLK cycles.

### 4.20.4 TRIS Strap Input Pin

The TRIS strap input signal is used by ISEs to allow “clipping-on” a PC87591L-N05 while it is mounted in the production system.

The TRIS input is sampled at $V_{CC}$ Power-Up reset while the device is in IRE or OBD environments. When TRIS is low (0), the PC87591L-N05 acts normally. When TRIS is high (1), all PC87591L-N05 outputs, except for DAC outputs and 32KX2, are put in TRI-STATE. In this case the ISE monitors and controls the system signals connected to the PC87591L-N05, instead of the PC87591L-N05 itself.

Section 2.3 on page 48 describes the strap input handling.

### 4.20.5 Freezing Events

The PC87591L-N05 can prevent real-time events from interfering with the operation of the ADB monitor and changing the status of the PC87591L-N05. This is done by disabling maskable interrupts and setting FREEZE bit; the FREEZE bit freezes the watchdog timer and disables destructive read operations.

#### Disabling Maskable Interrupts

Clearing the core I or E bits in PSR register disables the maskable interrupts. The I bit is cleared automatically whenever a trap or interrupt occurs and after reset.

#### Freezing the Watchdog Timer

To freeze the watchdog timer, FREEZE bit in DBGCFG register must be set to 1 on entering a TMON routine (the bit must be cleared before returning to the application). Setting FREEZE bit prevents the watchdog from generating the reset that occurs if watchdog is not cleared in time (see Section 4.10 on page 160). The watchdog timer keeps its value while it is frozen and resumes counting after FREEZE is cleared.

If an application fails to touch the watchdog in time and a reset event is generated before or while FREEZE bit is set, the PC87591L-N05 receives the reset.

#### Disabling Additional Modules

The two MFT16, the two USARTs and the four ACB modules may be frozen by the FREEZE bit. Freezing is enabled when the respective bit in DBGFRZEN register is set; the bits can be set to meet specific needs of different applications.

#### Disabling Destructive Reads

When FREEZE in DBGCFG register is set (1), destructive reads do not change the system state (i.e., they return the read data but do not clear or set bits or send signals). This allows the debugger to present the values of these bits. NMISTAT is an exception to this rule and is not affected by FREEZE. Core accesses to Host domain registers (using the “Core Access to Host Controlled Modules”) may also be destructive but are not affected by FREEZE. Note that host operations continue without any FREEZE bit impact.

### 4.20.6 Monitoring Activity During Development

In DEV environment, information is available for monitoring on-chip activities and implementing debug features in the development system.

### Bus Status Signals

The Bus Status signals (BST2-0) indicate if a transaction on the core bus was issued and, if so, the type of transaction.

The BST2-0 signals reflect activity on the core bus. For word accesses involving 8-bit expansion memory, the core bus cycle triggers two external bus cycles. The first external bus cycle is flagged as a T1 cycle of the core bus. The second is not flagged as a T1 cycle, i.e., BST2-0 is 000. See Table 31.

**Table 31. Core Bus Transaction Encoding**

| BST | Core Bus Transaction Type |
|---|---|
| 000 | Not a T1 cycle, except when the core waits for an interrupt following WAIT instruction execution |
| 001 | Core waits for an interrupt following WAIT instruction execution |
| 010 | T1 of an interrupt acknowledge bus cycle |
| 011 | T1 of a data transfer of a non-core bus master |
| 100 | T1 of a sequential instruction fetch |
| 101 | T1 of a non-sequential instruction fetch |
| 110 | T1 of a core data transfer |
| 111 | T1 of an exception data transfer |

### Transaction Effect on the External Bus

The following core bus transactions are reflected on the external bus:

- Accesses to external zones of expansion memory, off-chip base memory and accesses that use the I/O Expansion protocol are indicated by the active state of the $\overline{SEL0}$, $\overline{SEL12}$ and $\overline{SELIO}$ signals, respectively, and are described by the address and data buses and the status signals, BST2-0.
- Accesses to on-chip memories and peripheral modules can be observed using the "Core Bus Monitoring Bus Cycles" (see "Core Bus Monitoring" on page 80). They accesses are indicated by an inactive state for the $\overline{SEL0}$, $\overline{SEL12}$ and/or $\overline{SELIO}$ signals. They are described by addresses A0-20, the byte-enable BE0-1 signals, the CBRD signal and the BST2-0 signals.
- BE0 is high when a lower memory byte (a byte in an even address) is accessed. BE1 is high when a higher memory byte (a byte in an odd address) is accessed.
- CBRD is high when the transaction is a read operation and low when it is a write operation.

### Pipe Status Signals (PFS and PLI)

The PFS indicates the completion of an instruction in the core. The Pipe Long Instruction (PLI) signal indicates the size of the completed instruction, where 0 = word instruction and 1 = double-word instruction (see Figure 83). If an instruction flushes the pipeline, the fetch for the next instruction (BST=101) is issued during the cycle following the instruction's PFS, or later.

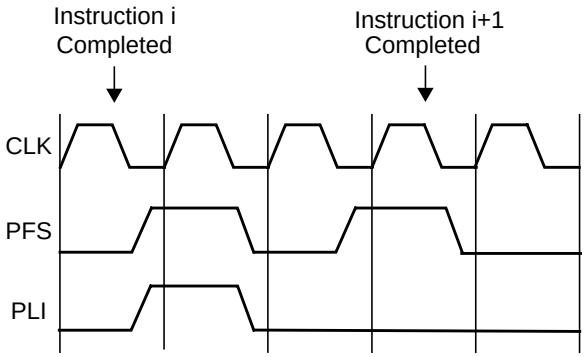



**Figure 83. Pipe Status Signal (PFS and PLI)**

### 4.20.7 On-Chip Hardware Breakpoint

The core provides two types of hardware breakpoints:

- Address Match - Detection of a matched address for the current executed instruction (PC value)
- Data Match - Detection of a read or write transaction for a matched memory location

For a detailed description of the core breakpoint mechanism, refer to the CR16B User Manual.

# 4.0 Embedded Controller Modules (Continued)

## 4.20.8 CR16B Development Support Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

**CR16B Development Support Register Map**

| Mnemonic | Register Name | Type |
|---|---|---|
| DBGCFG | Debug Configuration | R/W |
| DBGFRZEN | Debug Freeze Enable | R/W |
| DBGFRZEN2 | Debug Freeze Enable 2 | R/W |

### Debug Configuration Register (DBGCFG)

The DBGCFG register controls the configuration of debug support features. On reset, DBGCFG is cleared (0).

Only the software development tools may access DBGCFG. This enables application software to be binary compatible in all environments.

Location: 00 $FF16_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | **BRKLE** | **FREEZE** | **ON** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **ON.**<br>0: In IRE environment, this bit is always cleared to 0; any data written to it is ignored. The ON bit becomes read only when DBGL bit in DCR register in the core is set (default).<br>1: In OBD and DEV environments, enables the following debug support features:<br>- ISE interrupt signal<br>- Use of other bits in DBGCFG register |
| 1 | **FREEZE.**<br>0: No effect (default).<br>1: When ON is 1, the watchdog timer is stopped from counting. All bits that use destructive reads (i.e., bits set or cleared by read operations and other events triggered by reads) become indifferent to reads. An exception is NMISTAT register, which is always affected by reads. The DBGFRZEN and DBGFRZEN2 registers control the impact of FREEZE bit on a group of modules, enabling each module in the group to be indifferent to the FREEZE being set. FREEZE has no effect when ON is 0. |
| 2 | **BRKLE (Break Line Enable).**<br>0: Break Line input is ignored by core (default)<br>1: In DEV environment, when BRKLE is set, the $\overline{BRKL}$ input signal is passed to the core |
| 7-3 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## Debug Freeze Enable Register (DBGFRZEN)

The DBGFRZEN register enables the freeze operation to be performed on specific modules during debug. When the relevant bit is set, it enables the freeze of activities in the respective module when both FREEZE and ON are set in DBGCFG register. On reset, DBGFRZEN is loaded with $FF_{16}$.

Location: 00 $FF18_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **HIFEN** | **USARTFEN** | **ACB2FEN** | **ACB1FEN** | **MFT2FEN** | **MFT1FEN** |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 0 | **MFT1FEN (MFT16 1 Freeze Enable).**<br>0: FREEZE in DBGCFG register has no effect on the MFT16 1<br>1: Freezes the MFT16 1 when FREEZE is set (default) |
| 1 | **MFT2FEN (MFT16 2 Freeze Enable).**<br>0: FREEZE in DBGCFG register has no effect on the MFT16 2<br>1: Freezes the MFT16 2 when FREEZE is set (default) |
| 2 | **ACB1FEN (ACB1 Freeze Enable).**<br>0: FREEZE in DBGCFG register has no effect on the ACB 1 interface<br>1: Freezes the ACB 1 interface when FREEZE is set (default) |
| 3 | **ACB2FEN (ACB2 Freeze Enable).**<br>0: FREEZE in DBGCFG register has no effect on the ACB 2 interface<br>1: Freezes the ACB 2 interface when FREEZE is set (default) |
| 4 | **USARTFEN (USART Freeze Enable).**<br>0: FREEZE in DBGCFG register has no effect on the USART1 interface<br>1: Freezes the USART1 interface when FREEZE is set (default) |
| 5 | **HIFEN (Host Interface Freeze Enable).**<br>0: FREEZE in DBGCFG register has no effect on the Host Interface module<br>1: Freezes the Host interface when FREEZE is set (default) |
| 7-6 | **Reserved.** |

# 4.0 Embedded Controller Modules (Continued)

## Debug Freeze Enable Register2 (DBGFRZEN2)

The DBGFRZEN2 register enables the freeze operation to be performed on specific modules during debug. When the relevant bit is set, it enables the freeze of activities in the respective module when both FREEZE and ON are set in DBGCFG register. On reset, DBGFRZEN is loaded with $FF_{16}$.

Location: $00\ FF14_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | **USART2FEN** | **ACB4FEN** | **ACB3FEN** |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 0 | **ACB3FEN (ACB3 Freeze Enable).**<br>0: FREEZE in DBGCFG register has no effect on the ACB 3 interface<br>1: Freezes the ACB 3 interface when FREEZE is set |
| 1 | **ACB4FEN (ACB4 Freeze Enable).**<br>0: FREEZE in DBGCFG register has no effect on the ACB 4 interface<br>1: Freezes the ACB 4 interface when FREEZE is set |
| 2 | **USART2FEN (USART2 Freeze Enable).**<br>0: FREEZE in DBGCFG register has no effect on the USART2 interface<br>1: Freezes the USART2 interface when FREEZE is set |
| 7-3 | **Reserved.** |

# 5.0 Host Controller Interface Modules

This chapter describes functions that serve as an interface between the host and core domains. The functions are:

- Keyboard and Mouse Controller Interface (legacy $60_{16}$, $64_{16}$); see Section 5.1
- Two Power Management (PM) channels compliant with ACPI EC specifications; see Section 5.2 on page 251
- Shared Memory mechanism; see Section 5.3 on page 262
- Core Access to SuperI/O modules; see Section 5.4 on page 275
- Mobile System Wake-Up functions; see Section 5.5 on page 280

## 5.1 KEYBOARD AND MOUSE CONTROLLER INTERFACE

The PC87591L-N05 supports a standard Keyboard and Mouse Controller interface. This interface implements legacy ports $60_{16}$ and $64_{16}$.

### 5.1.1 Features

- Intel 8051SL-compatible Host interface
  - 8042 KBD standard interface (ports $60_{16}$, $64_{16}$)
  - Legacy IRQ: IRQ1 (KBD) and IRQ12 (mouse) support
  - Fast Gate A20 and Fast Reset via firmware
- Configured using two logical devices: Keyboard and Mouse

### 5.1.2 General Description

The PC87591L-N05 supports a keyboard/mouse communication channel that uses the standard command/status register and data registers. It uses either polling- or interrupt-driven communication schemes with the host and/or core. The hardware is designed to allow a race-free interface between the host and the PC87591L-N05.

The keyboard and mouse channel consists of three registers:

- DBBOUT - can be written by the core and read by the host processor.
- DBBIN - can be written by the host processor and read by the core.
- STATUS - can be read by both core and host processors. It has five bits (2, 4-7) that are written by the core. Three other bits are controlled by the hardware to indicate the status of DBBIN and DBBOUT registers.

#### Host Addresses

The host processor accesses the PC87591L-N05 Keyboard/Mouse Host Interface registers at two addresses in the host address space. These addresses are defined by two internal chip-select signals specified in the PC87591L-N05 host configuration registers; see Section 6.1.10 on page 310). Legacy settings of these addresses are $60_{16}$ and $64_{16}$ for the status/command and data registers, respectively.

Table 32 describes the register mapping to the host processor I/O space. For simplicity, the Host Interface module specification refers to the legacy addresses.

**Table 32. Mapping of the Host Interface Registers to the Host Processor**

| Port | Legacy Address | Internal Chip Select | Type | Register Name | Mnemonic |
|---|---|---|---|---|---|
| Keyboard and Mouse | $60_{16}$ | Keyboard/Mouse Data | Write | Data | DBBIN (A2=0) |
| | $64_{16}$ | Keyboard/Mouse Command | Write | Command | DBBIN (A2=1) |
| | $60_{16}$ | Keyboard/Mouse Data | Read | Data | DBBOUT |
| | $64_{16}$ | Keyboard/Mouse Command | Read | Status | STATUS |

#### Core Interrupts

The Host Interface module generates four level (high) interrupts to the core ICU. These can be used by the firmware for interrupt-driven control of the keyboard/mouse and/or PM channels.

#### Host Interrupts

The PC87591L-N05 Host Interface supports two interrupts to the host processor: Keyboard interrupt (legacy IRQ1) and Mouse interrupt (legacy IRQ12). These interrupts may be controlled either by hardware, according to the host interface buffer status, or by the PC87591L-N05 firmware toggling the bit value.

## 5.0 Host Controller Interface Modules (Continued)

The Host configuration module assigns host interrupts to IRQ numbers (see Section 6.1.10 on page 310). These interrupts are IRQ1 and IRQ12 for keyboard and mouse IRQs, respectively.

When IRQ1 and/or IRQ12 are disabled (OBFKIE and/or OBFMIE bits in HICTRL register are cleared), the firmware can control the IRQ1 and/or IRQ12 signals by writing to the signal's respective bit in HIIRQC register.

When IRQ1 and/or IRQ12 are controlled by hardware (OBFKIE and/or OBFMIE bits in HICTRL register are set to 1), interrupts to the host are generated according to the status of Output Buffer Full (OBF) flag.

In normal polarity mode (IRQNPOL in HIIRQC register is set to 0), the PC87591L-N05 supports two types of interrupts: Legacy Edge and Level. When an Edge interrupt is selected (IRQM in HIIRQC register is set), the interrupt signal default value is high (1). When an interrupt signal must be sent (i.e., the corresponding OBF flag is set), a negative pulse is generated. The pulse width is determined by IRQM field in HIIRQC register.

When the IRQ signals are set as level interrupts (IRQM in HIIRQC register is set to 0), the interrupt signal is usually low (0) and is asserted (1) to indicate that the respective OBF flag is set. The signal is de-asserted (0) when the output buffer is read (i.e., OBF flag is cleared).

Note that IRQ1 and IRQ12 have the same OBF flag but are not asserted together. Either IRQ1 or IRQ12 is set, depending on the internal register written (HIKDO or HIMDO, respectively).

In negative polarity mode (IRQNPOL in HIIRQC register is set to1), the IRQ signal behavior is inverted from the behavior described above.

The PC87591L-N05 firmware can read the values of the IRQ1 and IRQ12 signals by performing a read operation from IRQ1B and IRQ12B bits in HIIRQC register.

Figure 84 shows the effect of the different control bits on the IRQ signals.

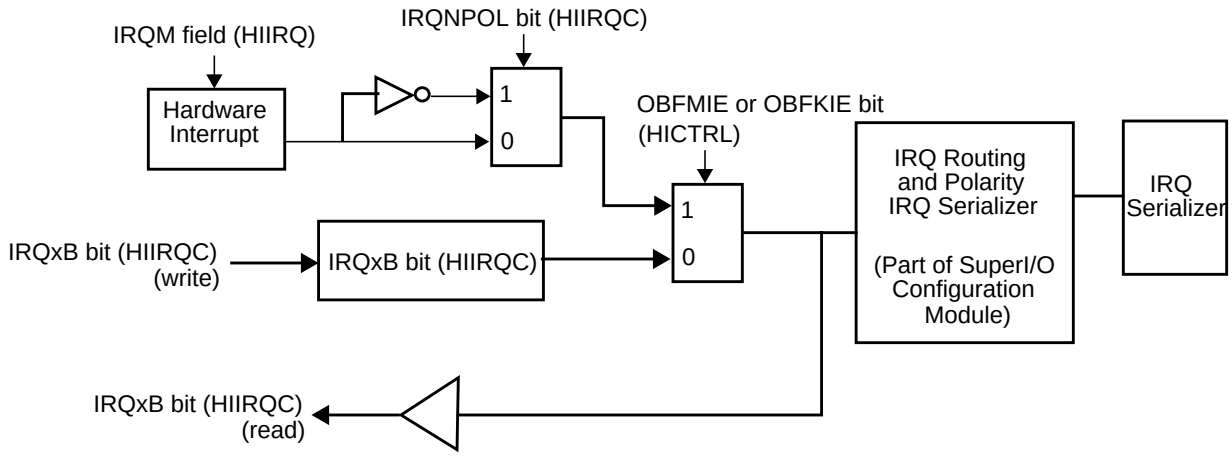



**Figure 84. IRQx (IRQ1, IRQ11 or IRQ12) Control Diagram**

### Keyboard/Mouse Channel ($60_{16}$, $64_{16}$)

The Host Interface of the PC87591L-N05 is compatible with the legacy 8042 host interface. It is based on two registers: Command/Data and Status. The Host Interface logic generates interrupts to the host processor and core according to the status of the input and output data buffers. Figure 85 provides a schematic description of the Host Interface Keyboard/Mouse channel.

## 5.0 Host Controller Interface Modules (Continued)

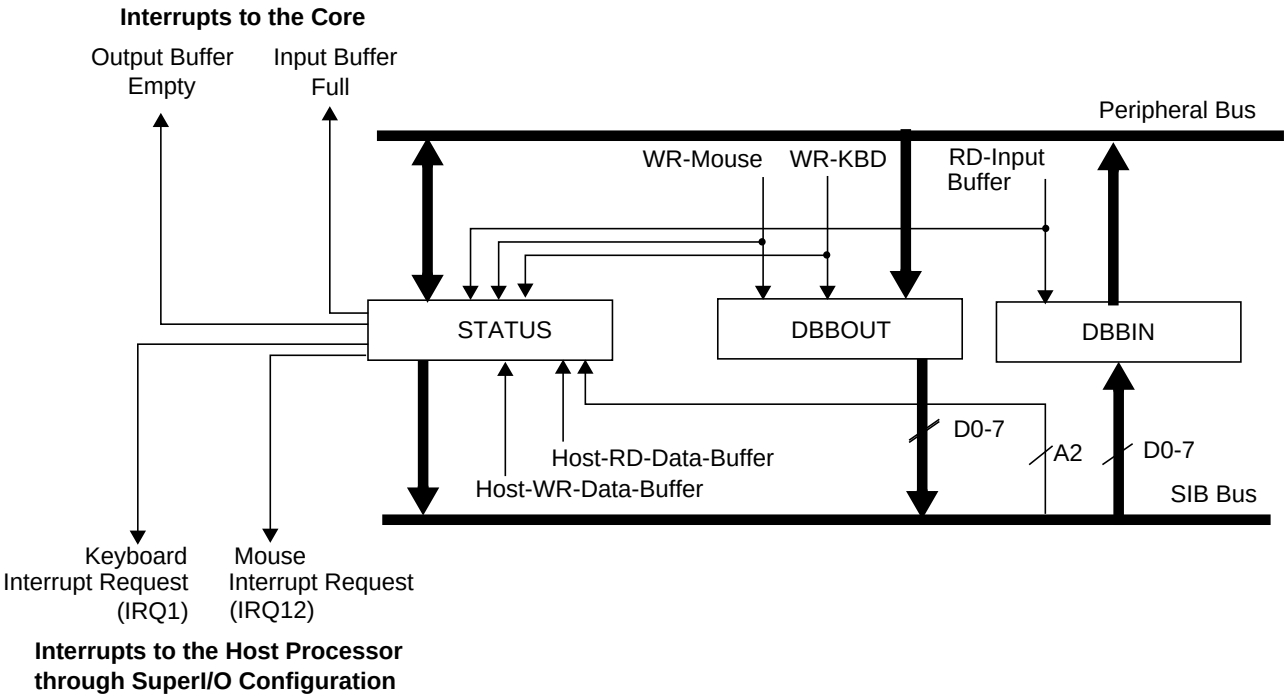



**Figure 85. Host Interface Keyboard/Mouse Channel (Ports 60,64) Block Diagram**

### Status Read

Both the host and the core can read the status of the KBC data buffers. Bits 2 and 4 to 7 can be written by the core. The host processor should read address $64_{16}$ to obtain the contents of the Status register. The core software can obtain this information by reading/writing the HIKMST register. The format of the Status register is identical for both the host and the core (see “Host Interface Keyboard/Mouse Status Register (HIKMST)” on page 249).

#### Host Data Write to Host Interface Keyboard/Mouse Channel

The data buffer has two latches; one serves as an input buffer and the other as an output buffer. When writing to address $60_{16}$ or $64_{16}$, the following sequence of events occurs: the data is written to the Data In latch (DBBIN), IBF bit in the Status register is set and bit 3 (A2) in the Status register indicates to the core which address (command or data) was written to. When writing to address $60_{16}$ (legacy A2=0), bit 3 of the Status register is cleared. When writing to address $64_{16}$ (legacy A2=1), bit 3 of the Status register is set.

The core identifies that data is present in the input buffer by either polling IBF bit of the Status register or acknowledging an interrupt when the input buffer interrupt is enabled (IBFCIE in HICTRL register is set to 1).

When the input buffer is full, reading the Status register identifies which address was written to (i.e., check A2 of HIKMST register). The core can then read the data from the input buffer (HIKMDI). The IBF status bit is cleared when the data input buffer is read by the core.

#### Host Data Read from Host Interface Keyboard/Mouse Channel

The output data latch (DBBOUT) is written by the core when it needs to send data to the host. The OBF flag in the Status register (OBF in HIKMST register) is set to indicate that data is available in DBBOUT. DBBOUT should be written only when this bit is cleared.

The PC87591L-N05 supports polling and interrupt communication schemes with the host. Both Keyboard interrupt (IRQ1) and Mouse interrupt (IRQ12) are supported.

The core firmware writes to HIKDO register data addressed to the keyboard driver (i.e., generate IRQ1). A write to HIKDO stores the data to DBBOUT and sets OBF bit. If the IRQ1 interrupt is enabled (OBFKIE in HICTRL register is set to 1), it is also sent according to the interrupt mode (IRQM field and IRQNPOL bit in HIIRQC register).

The core firmware writes data addressed to the mouse driver (IRQ12) to HIMDO register. A write to HIMDO stores the data in DBBOUT and sets OBF bit. If the IRQ12 interrupt is enabled (OBFMIE in HICTRL is set to 1); the IRQ12 interrupt is also sent according to the interrupt mode (IRQM field and IRQNPOL bit in HIIRQC register).

## 5.0 Host Controller Interface Modules (Continued)

The host processor identifies that data is present in the output buffer by either polling the Status register (reading address $64_{16}$) or responding to IRQ1 or IRQ12. When this data is available, the host can read it using a read operation from address $60_{16}$. Reading from address $60_{16}$ clears the OBF flag. In addition, when the host interrupt is in level mode (IRQM in HIIRQC register is set to $000_2$) and the hardware interrupt is enabled, IRQ1 or IRQ12 are de-asserted (low if IRQNPOL in HIIRQC register is set to 0).

The core can read OBF bit to identify when the output buffer is empty and ready for a new data transfer. When the Output Buffer Empty interrupt to the core is enabled (OBECIE in HICTRL register is set to 1), the interrupt signal to the ICU is set high if the output buffer is empty (OBF=0).

### 5.1.3 Host Interface Registers

The module has four registers, described below. The base address for each may be configured individually. For legacy operation, they should be configured to $60_{16}$ and $64_{16}$ (see Section 6.1.10 on page 310).

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

#### Host Interface Register Map

| Offset | Mnemonic | Register Name | Type |
|---|---|---|---|
| $60_{16}$ | DBBOUT | Data Out Buffer | R |
| $64_{16}$ | STATUS | Status | R |
| $60_{16}$ | DBBIN | Data In Buffer | W |
| $64_{16}$ | COMAND | Command In Buffer | W |

#### Data Out Buffer Register (DBBOUT, Legacy $60_{16}$)

This register allows the host to read DBBOUT register while clearing OBF bit in the Status register. If the host interrupts are level (IRQM in HIIRQC register is $000_2$), the interrupt is de-asserted. If the core interrupt on output buffer empty is enabled (OBECIE in HICTRL register is set to 1), reading this register asserts it (high).

Location: As defined in LDN $06_{16}$ registers, index $60_{16}$ and $61_{16}$

Type: R

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Keyboard/Mouse DBBOUT Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Keyboard/Mouse DBBOUT Data.** |

## 5.0 Host Controller Interface Modules (Continued)

### Status Register (STATUS, Legacy $64_{16}$)

This register provides the status of the host interface keyboard channel buffers (DBBIN and DBBOUT) and the value of messages sent by the core using the Status bits to the host. The Status register can also be read by the core as HIKMST. The Status register is cleared ($00_{16}$) on reset.

Location: As defined in LDN $06_{16}$ registers, index $62_{16}$ and $63_{16}$

Type: R

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **ST3-ST0** | | | | **A2** | **F0** | **IBF** | **OBF** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **OBF (Output Buffer Full).** This bit is set when the keyboard/mouse channel's DBBOUT is written by the core (i.e., writing to HIKDO or HIMDO registers). The bit is cleared by a host processor read from the keyboard/mouse channel output buffer ($60_{16}$). |
| 1 | **IBF (Input Buffer Full).** This bit is set when the keyboard/mouse channel's DBBIN is written by the host processor (i.e., writing to either address $60_{16}$, data or address $64_{16}$, control). The bit is cleared when the core reads the input buffer (HIKMDI). |
| 2 | **F0 (Flag 0).** A general-purpose flag that can be set or cleared by the core firmware. |
| 3 | **A2 (A2 Address).** Holds the value of the A2 signal in the last write operation of the host to the keyboard/mouse channel's input buffer (i.e., A2=0 for Data In Buffer write and A2=1 for Command In Buffer write). |
| 7-4 | **ST3-ST0 (Status).** Four general-purpose flags that can be set or cleared by the core firmware. |

### Data In Buffer Register (DBBIN, Legacy $60_{16}$)

This register allows the host to write to DBBIN register while setting Status register bit IBF and clearing Status register bit A2 bit. If the core interrupt on IBF is enabled (IBFCIE in HICTRL register is set to 1), writing to this register asserts it (high).

Location: As defined in LDN $06_{16}$ registers, index $60_{16}$ and $61_{16}$

Type: W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Keyboard/Mouse DBBIN Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Keyboard/Mouse DBBIN Data.** |

### Command In Buffer Register (COMAND, Legacy $64_{16}$)

This register allows the host to write to DBBIN register while setting IBF and A2 bits in the Status register. If the core interrupt on IBF is enabled (IBFCIE in HICTRL register is set to 1), writing to Data In Buffer asserts it (high).

Location: As defined in LDN $06_{16}$ registers, index $62_{16}$ and $63_{16}$

Type: W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Keyboard/Mouse DBBIN Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Keyboard/Mouse DBBIN Data.** |

# 5.0 Host Controller Interface Modules (Continued)

## 5.1.4 Core Interface Registers

Some register bits affect PM channel 1 to achieve firmware compatibility with the PC87570.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### Core Interface Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| HICTRL | Host Interface Control | R/W |
| HIIRQC | Host Interface IRQ Control | R/W |
| HIKMST | Host Interface Keyboard/Mouse Status | R/W |
| HIKDO | Host Interface Keyboard Data Out Buffer | WO |
| HIMDO | Host Interface Mouse Data Out Buffer | WO |
| HIKMDI | Host Interface Keyboard/Mouse Data In Buffer | RO |

### Host Interface Control Register (HICTRL)

The HICTRL is used in setting host interface mechanism options. On reset, non-reserved bits of HICTRL are cleared.

Location: 00 $FEA0_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **PMICIE** | **PMIOCIE** | **PMIHIE** | **IBFCIE** | **OBECIE** | **OBFMIE** | **OBFKIE** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **OBFKIE (Output Buffer Full Keyboard Interrupt Enable).**<br>0: IRQ1 is controlled by IRQ1B bit in HIIRQC register (default)<br>1: Enables the Output Buffer Full interrupt to the keyboard driver of the host processor (IRQ1). The interrupt is triggered by a core write to HIKDO register and sent according to IRQM field and IRQNPOL bit in HIIRQC register. |
| 1 | **OBFMIE (Output Buffer Full Mouse Interrupt Enable).**<br>0: IRQ12 is controlled by IRQ12B bit in HIIRQC register (default)<br>1: Enables the Output Buffer Full interrupt to the mouse driver in the host processor (IRQ12). The interrupt is triggered by the core write to HIMDO register and sent according to IRQM field and IRQNPOL bit in HIIRQC register. |
| 2 | **OBECIE (Output Buffer Empty Core Interrupt Enable).**<br>0: Interrupt signal is low (default).<br>1: Enables the Output Buffer Empty interrupt to the core ICU for the keyboard/mouse channel. The interrupt signal is active when the output buffer is empty (i.e., the interrupt signal is set (1) when OBF bit is cleared). |
| 3 | **IBFCIE (Input Buffer Full Core Interrupt Enable).**<br>0: Interrupt signal is low (default).<br>1: Enables the Input Buffer Full interrupt to the core ICU for the keyboard/mouse channel. The interrupt signal is active when the input buffer is full (i.e., the interrupt signal is set (1) when IBF bit is set). |
| 4 | **PMIHIE** (**PM Channel 1 Host Interrupt Enable),**<br>0: IRQ11 is controlled by IRQ11B bit in HIIRQC register (default).<br>1: Enables the Output Buffer Full interrupt of PM channel 1 in PC87570 Compatible mode, to drive the host processor interrupt. The interrupt is noted as IRQ11 and may be routed to any of the IRQs, to SMI or to the SCI events. The interrupt is triggered by a core write to HIPM0DO register and sent according to IRQM field and IRQNPOL bit in HIIRQC register. |

## 5.0 Host Controller Interface Modules (Continued)

| Bit | Description |
|---|---|
| 5 | **PMOCIE (PM Channel 1 Output Buffer Empty Core Interrupt Enable).**<br>0: Interrupt signal is low (default)<br>1: Enables the PM Output Buffer Empty interrupt to the core ICU for PM channel 1 in PC87570 Compatible mode. The interrupt signal is active when the output buffer is empty (OBF bit is cleared in the PM channel status register). |
| 6 | **PMICIE (PM Channel 1 Input Buffer Full Core Interrupt Enable).**<br>0: Interrupt signal is low (default)<br>1: Enables the PM input buffer full interrupt to the core ICU for PM channel 1 in PC87570 Compatible mode. The interrupt signal is active when the input buffer is full (IBF bit is set in the PM channel status register). |
| 7 | **Reserved.** |

### Host Interface IRQ Control Register (HIIRQC)

The HIIRQC register controls the IRQ signals mode of operation. On reset, HIIRQC is set to $07_{16}$.

Location: 00 $FEA2_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **IRQNPOL** | **IRQM** | | | **IRQ11B** | **IRQ12B** | **IRQ1B** |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| Bit | Description |
|---|---|
| 0 | **IRQ1B (Host Interrupt Request 1 Control Bit).** When the IRQ1 signal is configured for direct control by the firmware (OBFKIE in HICTRL register is 0), IRQ1B bit is output to the IRQ1 signal. When read, IRQ1B bit returns the current value of the IRQ1 pin. The IRQ1 signal value can be read regardless of the state of OBFKIE bit. |
| 1 | **IRQ12B (Host Interrupt Request 12 Control Bit).** When the IRQ12 signal is configured for direct control by the firmware (OBFMIE in HICTRL register is 0), IRQ12B bit is output to the IRQ12 signal. When read, IRQ12B bit returns the current value of the IRQ12 pin. The IRQ12 signal value can be read regardless of the state of the OBFMIE bit. |
| 2 | **IRQ11B (Host Interrupt Request 11 Control Bit).** When PM channel 1 is in PC87570 Compatible mode and its host interrupt is configured for direct control by the firmware (PMHIE in HICTRL register is 0), IRQ11B bit is output to the IRQ11 signal. When read, IRQ11B bit returns the current value of the IRQ11 signal. The IRQ11 signal value can be read regardless of the state of PMHIE bit; see Section 5.2 on page 251 for details about the PM channel 1 interrupt scheme. |
| 5-3 | **IRQM (IRQ Mode).** Sets the hardware-controlled IRQ signals to work in Level or Pulse mode and defines the pulse width in Pulse mode.<br>When IRQM = $000_2$, the IRQ signals function in Level mode. In this mode, when IRQNPOL = 0, the signal's default value is low, and a high level is set to issue an interrupt (the respective OBF is set).<br>When IRQM ≠ 0, the host interrupts are in Pulse mode. When IRQNPOL = 0, the signal's default value is high and toggles low to issue an interrupt (i.e., when the respective output buffer register is written). The pulse width is as follows:<br>**Bits 5 4 3 — Pulse Width**<br>0 0 0: Level Interrupt (default)<br>0 0 1: 1-Cycle Pulse<br>0 1 0: 2-Cycle Pulse<br>0 1 1: 4-Cycle Pulse<br>1 0 0: 8-Cycle Pulse<br>1 0 1: 16-Cycle Pulse<br>Other: Reserved |

## 5.0 Host Controller Interface Modules (Continued)

| Bit | Description |
|---|---|
| 6 | **IRQNPOL (Negative Polarity).** When IRQNPOL is cleared, the IRQ (IRQ1, IRQ11, IRQ12) signal polarity is compatible with the standard ISA bus interface (as specified in the IRQM field). When hardware IRQ generation is enabled (HICTRL register bits OBFKIE for IRQ1 and IRQ12; PMHIE for IRQ11), the interrupt output is inverted if IRQNPOL is set. |
| 7 | **Reserved.** |

### Host Interface Keyboard/Mouse Status Register (HIKMST)

The HIKMST register provides the status of the Host Interface keyboard channel buffers (DBBIN and DBBOUT) and a way for the PC87591L-N05 to send status bits to the host. This register can also be read by a host processor read operation from address $64_{16}$. On reset, the register is cleared.

Location: 00 $FEA4_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **ST3-ST0** | | | | **A2** | **F0** | **IBF** | **OBF** |
| Reset | 0 | | | | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **OBF (Output Buffer Full).** The bit is set when the keyboard/mouse channel's DBBOUT is written by the core (i.e., writing to HIKDO or HIMDO register). The bit is cleared by a host processor read from the keyboard/mouse channel output buffer ($60_{16}$). This read-only bit is ignored when writing to this register. |
| 1 | **IBF (Input Buffer Full).** The bit is set when the keyboard/mouse channel's DBBIN is written by the host processor, i.e., writing to either address $60_{16}$ (data) or address $64_{16}$ (control). The bit is cleared by a core read of the input buffer (HIKMDI). This read-only bit is ignored when writing to this register. |
| 2 | **F0 (Flag 0).** A general-purpose flag that can be set or cleared by the core firmware. |
| 3 | **A2 (A2 Address).** Holds the value of the A2 signal in the last write operation of the host to the keyboard/mouse channel's input buffer (i.e., indicates A2 value during write to address $60_{16}$ or $64_{16}$). This read-only bit is ignored when writing to this register. |
| 7-4 | **ST3-ST0 (Status Bits).** Four general-purpose flags that can be set or cleared by the core firmware. |

### Host Interface Keyboard Data Out Buffer Register (HIKDO)

The HIKDO register allows the core firmware to write to DBBOUT register while setting OBF bit in the Status register. If IRQ1 interrupt is enabled, it is sent. If the core interrupt on output buffer empty is enabled (OBECIE in HICTRL register is 1), writing to HIKDO de-asserts it (low).

Location: 00 $FEA6_{16}$

Type: WO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Keyboard DBBOUT Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Keyboard DBBOUT Data.** |

## 5.0 Host Controller Interface Modules (Continued)

### Host Interface Mouse Data Out Buffer Register (HIMDO)

The HIMDO register allows the core firmware to write to the DBBOUT register while setting OBF bit in the Status register. If an IRQ12 interrupt is enabled, it is sent. If the core interrupt on output buffer empty is enabled (OBECIE in HICTRL register is 1), writing to HIMDO de-asserts it (low).

Location: 00 $FEA8_{16}$

Type: WO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Mouse DBBOUT Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Mouse DBBOUT Data.** |

### Host Interface Keyboard/Mouse Data In Buffer Register (HIKMDI)

The HIKMDI register allows the core firmware to read to the DBBIN register while clearing IBF bit in the Status register. If the core interrupt on IBF is enabled (IBFCIE in HICTRL register is 1). Reading from HIKMDI de-asserts it (low).

Location: 00 $FEAA_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Keyboard/Mouse DBBIN Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Keyboard/Mouse DBBIN Data.** |

# 5.0 Host Controller Interface Modules (Continued)

## 5.2 POWER MANAGEMENT (PM) CHANNELS

The PC87591L-N05 implements two PM communication channels, both of which are compliant with ACPI specifications for an embedded controller interface. The PC87591L-N05 may be configured to work in either Private Interface or Shared Interface mode. The PM interface has two identical channels, which are individually configured and used. If using only one channel, Shared Interface mode should be used; if using both channels, Private Interface mode may be used.

The description below refers to a single channel. The signals may have the suffix 'n' to indicate the channel number, either 1 or 2, where applicable. Registers in the core domain have the prefix HIPMn to indicate the channel number. Host domain registers are identified by the logical device to which they belong.

**Note:** When working in PC87570 Compatible mode, only channel 1 may be used.

### 5.2.1 Features

- Two operation modes
  - PC87570 Compatible
  - Enhanced PM
- ACPI embedded controller interface compliant support
  - Shared interface
  - Private interface
- PM channel registers (channel 1: legacy $62_{16}$, $66_{16}$; channel 2: legacy $68_{16}$, $6C_{16}$)
  - Command/Status
  - Data
- PM interrupt using
  - IRQ
  - SMI
  - SCI

### 5.2.2 General Description

The PM channel has two modes of operation:

- PC87570 Compatible (available for channel 1 only) supports software previously written for the PC87570.
- Enhanced PM includes a mechanism that facilitates easier generation of SCI and SMI interrupts to the host.

Figure 86 is a schematic diagram of the PM channel.

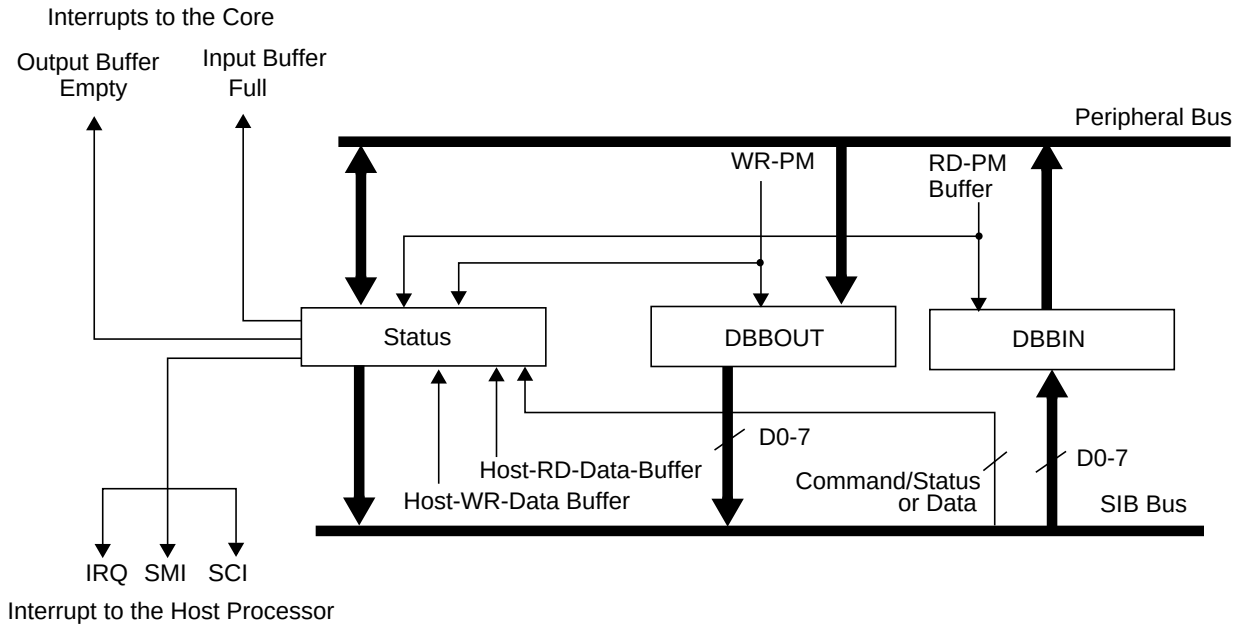



**Figure 86. Host Interface PM Channel n Block Diagram**

## 5.0 Host Controller Interface Modules (Continued)

### Data Registers

The PM channel has three registers.

- DBBOUT - can be written to by the core and read by the host processor. Multiple addresses in the core address space enable generating an IRQ, SMI or SCI interrupt on Output Buffer Full (OBF).
- DBBIN - can be written to by the host processor and read by the core.
- STATUS - can be read by both the core and the host processor. It has five bits (bits 2 and 4-7) that are written to by the core directly or, in Enhanced PM mode, via the control and configuration register. Three other bits are controlled by hardware to indicate the status of DBBIN and DBBOUT registers.

### Host Addresses

The host processor accesses the PC87591L-N05 PM channel interface registers at two addresses in the host address space. These addresses are defined by two internal chip select signals specified in the PC87591L-N05 host configuration registers; see Section 6.1.13 on page 317 and Section 6.1.14 for channels 1 and 2, respectively. The Legacy setting of these addresses is $62_{16}$ and $66_{16}$ for channel 1 Status/Command and Data registers, respectively.

Table 33 shows the register mapping to the host processor I/O space. For simplicity, the Host Interface module specification refers to the legacy addresses.

**Table 33. Host Interface Registers to Host Processor Mapping**

| Port | Legacy Address[1] | Configuration Register Index | PM Internal Chip Select | Type | Register Name | Mnemonic |
|---|---|---|---|---|---|---|
| PM Channel n | $62_{16}$ | Index $60_{16}$, $61_{16}$ | Data | Write | Data | DBBIN |
| | $66_{16}$ | Index $62_{16}$, $63_{16}$ | Command/Status | Write | Command | DBBIN |
| | $62_{16}$ | Index $60_{16}$, $61_{16}$ | Data | Read | Data | DBBOUT |
| | $66_{16}$ | Index $62_{16}$, $63_{16}$ | Command/Status | Read | Status | STATUS |

1. The legacy address serves as an example only. Do not assign the same address for both channels.

### Core Interrupts

For each channel, the Host Interface module generates two level (high) interrupts to the core ICU (see Figure 87). The firmware can use these for interrupt-driven control of the PM channels.

In PC87570 Compatible mode (EME in HIPMnCTL register is set to 0), interrupts are enabled using HICTRL register bits PMOCIE and PMICIE for output buffer empty and input buffer full interrupts, respectively.

In Enhanced PM mode (EME in HIPMnCTL register is set to1), interrupts are enabled using HIPMnCTL register bits OBEIE and IBFIE for output buffer empty and input buffer full interrupts, respectively.

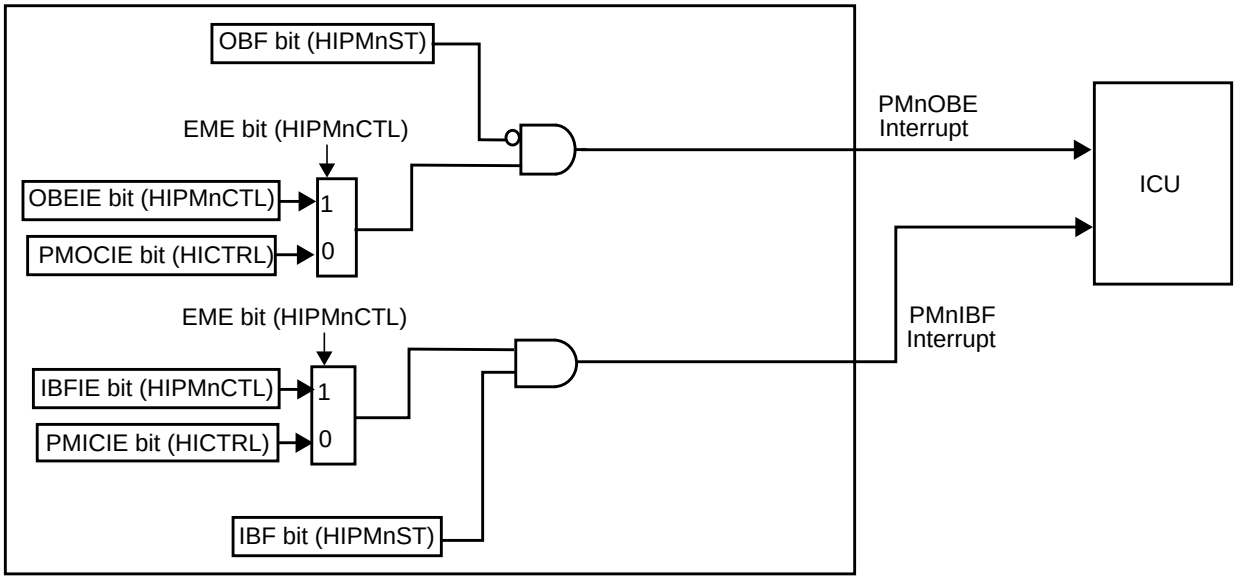



**Figure 87. Core Interrupt Request for PM Channel n**

# 5.0 Host Controller Interface Modules (Continued)

## Host Interrupt Generation Modes

The Host Interface module generates three types of interrupts to the host: regular IRQ, SMI and SCI. The interrupt schemes are designed to meet ACPI requirements for host interrupts. Two interrupt modes are supported: PC87570 Compatible and Enhanced PM.

### PC87570 Compatible Mode

PC87570 Compatible mode uses the same method for IRQ generation as the PC87570. It is available only for PM channel 1 and is enabled when EME in HIPMnCTL register is set to 0. Figure 89 shows this scheme.

The host configuration module assigns host interrupts to IRQ numbers (see Section 6.1.13 on page 317). IRQ11 is used as an example interrupt and for the signal naming (the actual interrupt number used is determined by the IRQ routing logic). When hardware-driven IRQ11 is disabled (PMHIE in HICTRL register is cleared), the firmware can control the IRQ11 signal by writing to the signal's respective bit in HIIRQC register. When hardware-driven IRQ11 is enabled (PMHIE is set to 1), interrupts to the host are generated according to the status of the OBF flag.

In Normal Polarity mode (IRQNPOL in HIIRQC register is cleared), the PC87591L-N05 supports two types of interrupts: legacy edge or legacy level. When an edge interrupt is selected (IRQM field in HIIRQC register is set to a value other than 0), the interrupt signal default value is high (1). When an interrupt signal must be sent (i.e., OBF flag is set), a negative pulse is generated. The pulse width is determined by the same field, IRQM, that selects the edge interrupt.

When a level interrupt is selected (IRQM in HIIRQC register is cleared), the interrupt signal is usually low (0). It is asserted (1) to indicate that the respective OBF flag has been set. The signal is de-asserted (0) when the output buffer is read (i.e., OBF flag is cleared).

In Negative Polarity mode (IRQNPOL in HIIRQC register is set to 1), IRQ signal behavior is inverted from the behavior described for normal polarity.

The PC87591L-N05 firmware can read the value of the IRQ11 signal by performing a read operation of IRQ11B bit in HIIRQC register.

The core can also control the routing of interrupts generated by the PM channel to one of the following:

- IRQ signal, when IRQE bit in HIPMnIE register is set
- SMI output, when SMIE bit in HIPMnIE register is set
- SCI event, using the $\overline{\text{ECSCI}}$ output, when SCIE bit in HIPMnIE register is set.

The core firmware should not enable more than one of these interrupts simultaneously. It should also update ST0 and ST1 bits to indicate the type of host interrupt used.

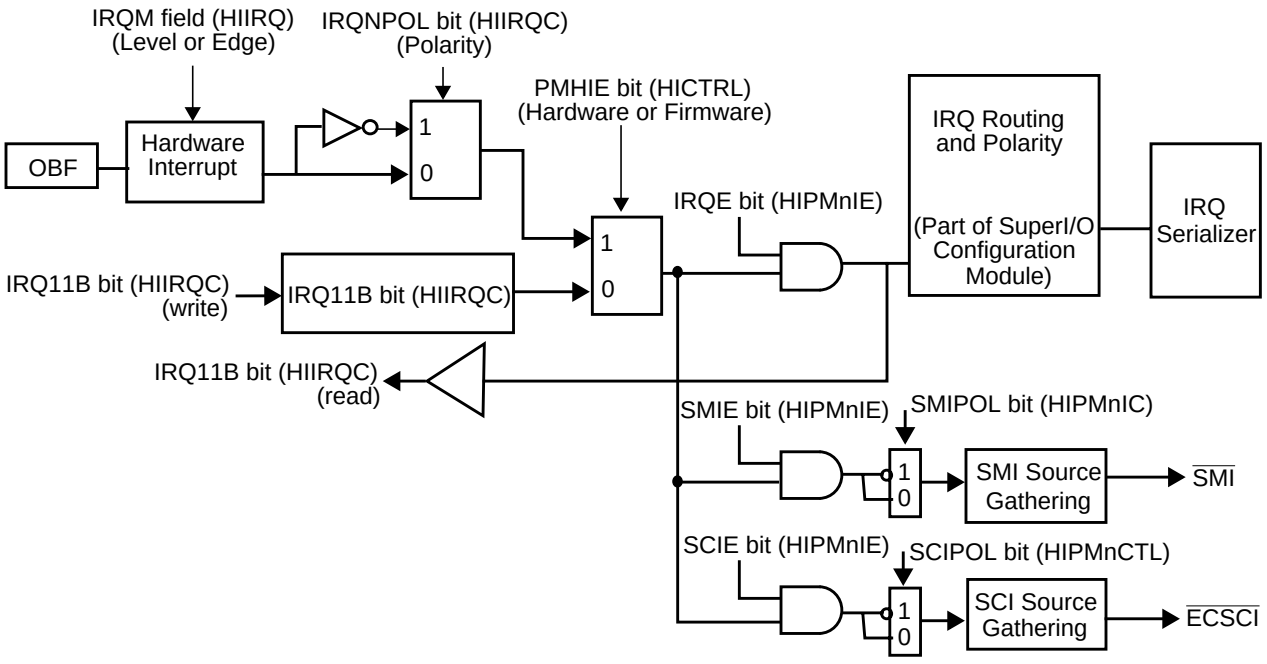



**Figure 88. IRQ, SCI and SMI Control in PC87570 Compatible Mode (PM Channel 1 Only)**

## 5.0 Host Controller Interface Modules (Continued)

### Enhanced PM Mode

Enhanced PM mode is available for both PM channels. It is enabled when EME in HIPMnCTL register is set to 1. Figure 89 shows interrupt generation in this mode.

Either IRQ, SMI or SCI interrupts may be generated under software control or by using hardware. Using hardware reduces software overhead and simplifies procedures.

The mechanism that generates the IRQ is identical to that used in PC87570 Compatible mode. To enable identical control of both channels, the bits that are used for channel 1 are separated from the keyboard/mouse channel's registers; see Figure 89 for bit usage.

IRQE in HIPMnIE register determines if an IRQ is sent from PM Channel n.

Enhanced PM mode supports direct generation of SCI and SMI on core writes to the Data output buffer and generation of SCI on core reading of the Data Input buffer. The core decides whether to generate an interrupt and which type of interrupt to generate by selecting the data register address in use.

When data is written to HIPMnDO register, the OBF flag in HIPMnST register is set and neither SMI nor SCI is generated.

When data is written to HIPMnDOM register, the OBF flag and the internal OBF_SMI flag are set and an SMI interrupt is generated. The OBF_SMI flag is cleared when OBF flag is cleared. The SMI is generated as a pulse whose width is defined by PLMM in HPIMnIC register.

The SMI interrupt is routed to the SMI pin only if both HSMIE and SMIE bits in HIPMnIE register are set. When SMIE is set and HSMIE is cleared, SMIB in HIPMnIC register is used as the PMnSMI signal value. When SMIE is cleared, the PMnSMI signal is inactive (high).

When data is written to HIPMnDOC register, the OBF flag and OBF_SCI internal flag are set and an SCI interrupt is generated. OBF_SCI is cleared when OBF is cleared. The SCI is generated as a pulse whose width is defined by PLMS in HPIMnCTL register.

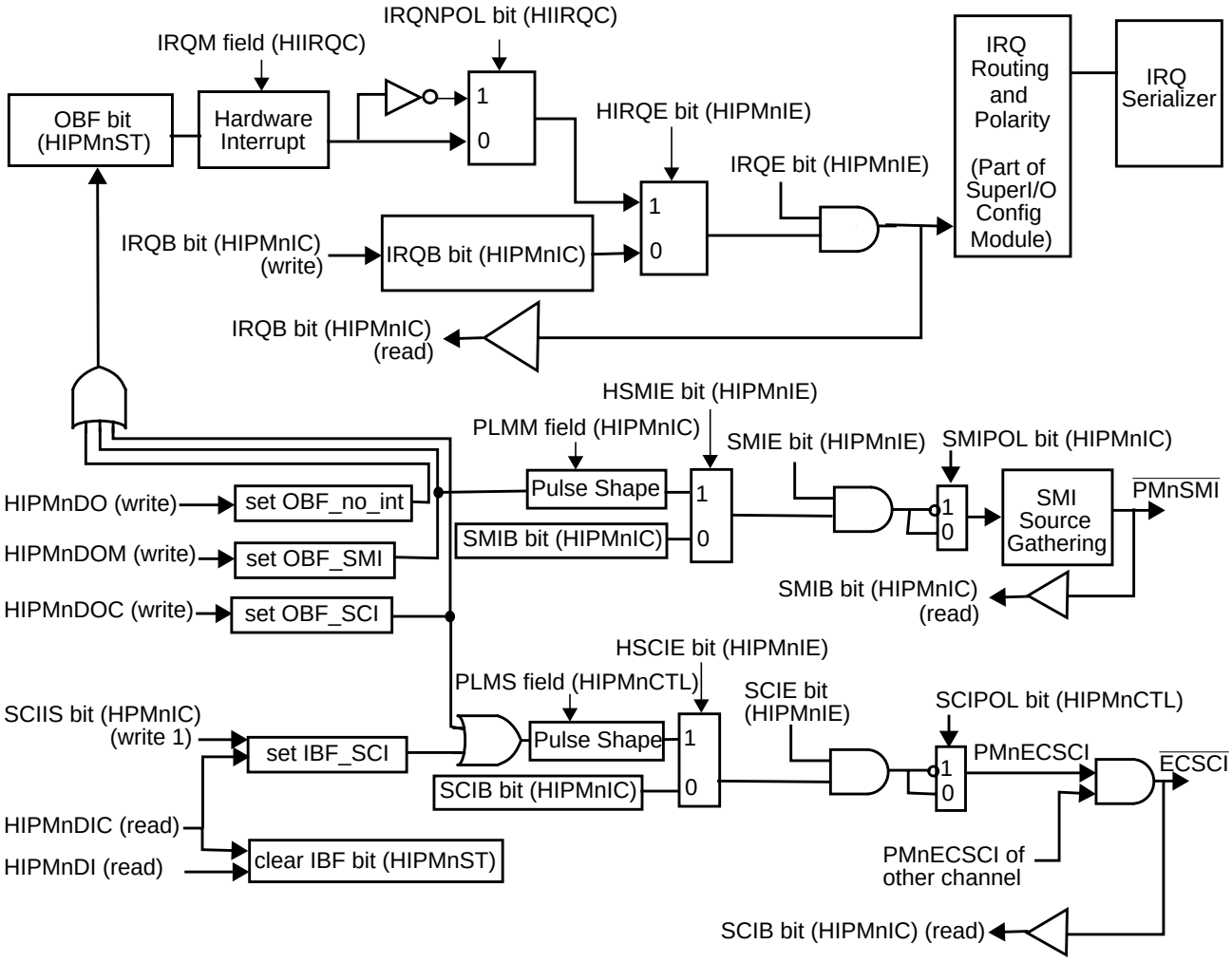



**Figure 89. IRQ, SCI and SMI Control, Enhanced PM Mode**

## 5.0 Host Controller Interface Modules (Continued)

When data is read from HIPMnDIC register, the IBF flag is cleared, the IBF_SCI Internal flag is set and an SCI interrupt is generated. The IBF_SCI flag is cleared when the IBF is set again. The SCI is generated as a pulse whose width is defined by PLMS in HPIMnCTL register. Reading from HIPnDI register clears the IBF flag but does not generate an SCI interrupt.

Note that IBF_SCI flag may also be set by writing a 1 to SCIIS bit in HIPMnIC register. This is done to start an SCI interrupt on input buffer empty without a read operation from the input buffer.

The SCI interrupt is routed to the SCI pin only if HSCIE and SCIE in HIPMnIE register are set. When SCIE is set and HSCIE is cleared, the value of SCIB in HIPMnIC register is used as the PMnSCI signal value. When SCIE is cleared, PMnSCI is inactive (high).

### Status Read

The status of the PM channel data buffers can be read by both the host and the core. Bits 2 and 4-7 can be written by the core.

The host processor should read the Status register I/O address (legacy $66_{16}$, for channel 1) to obtain the contents of the Status register. The core software should read/write the HIPMnST register to access the same information. The format of the Status register is identical for both the host and the core; see “Host Interface PM n Status Register (HIPMnST)” on page 256.

#### Host Data Write to Host PM Channel

The data buffer has two latches: one serves as an input buffer and the other serves as an output buffer. When writing to the Command (legacy address $66_{16}$) or Data (legacy address $62_{16}$) registers, the following sequence occurs: data is written to the Data In latch (DBBIN), IBF bit in the Status register is set and bit 3 (A2) in the Status register indicates to the core which of the two addresses was written to. When writing to the data register address, A2 bit of the Status register is cleared (0). When writing to the Command register address, A2 bit in the Status register is set (1).

The core identifies that data is present in the input buffer by either polling IBF bit in the Status register or acknowledging an interrupt when the input buffer interrupt is enabled (IBFCIE bit in HICTRL register is set to 1).

When the input buffer is full, the Status register should be read to identify which addresses were written to by checking A2 bit in HIPMnST register. The core can then read the data from the input buffer (HIPMnDI or HIPMnDIC registers). The IBF status bit is cleared when the data input buffer is read by the core.

### Host Data Read from Host Interface Power Management Channel

The core writes to the Output Data latch (DBBOUT) when it needs to send data to the host. The OBF flag in the Status register (HIPMnST) is set to indicate that data is available in DBBOUT. DBBOUT should be written to only when OBF in HIPMnST register is cleared.

The PC87591L-N05 supports polling and interrupt communication schemes with the host. IRQ, SMI or SCI interrupts may be used. The core firmware writes data addressed to the PM drivers to the HIPMnDO register. When working in Enhanced PM mode, writes to HIPMnDOC and HIPMnDOM may be used to automatically generate SCI and SMI, respectively. Refer to “Host Interrupt Generation Modes” on page 253 for details of the interrupt generation scheme.

The host processor identifies the presence of data in the output buffer by either polling the Status register or by responding to IRQ, SMI or SCI events. When such data is available, the host can read it using a read operation from the address of the data register (legacy $62_{16}$ for channel 1). Reading from the data register clears the OBF flag (HIPMnST). In addition, when the host interrupt is in level mode (IRQM in HIIRQC register is set to $000_2$) and the hardware interrupt is enabled, the IRQ signal is de-asserted (low).

The core can read OBF in HIPMnST register to identify when the output buffer is empty and ready for a new data transfer. When the output buffer empty interrupt to the core is enabled (PMOCIE bit in HICTRL register is 1 when EME bit in HIPMnCTL register is 0), the interrupt signal to the ICU is set high if the output buffer is empty (OBF is set to 0).

# 5.0 Host Controller Interface Modules (Continued)

## 5.2.3 Core PM Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### Core PM Register Map

| Mnemonic | Register Name | Type |
|---|---|---|
| HIPMnST[1] | Host Interface PM n Status | Varies per bit |
| HIPMnDO[1] | Host Interface PM n Data Out Buffer | WO |
| HIPMnDOC[1] | Host Interface PM n Data Out Buffer with SCI | WO |
| HIPMnDOM[1] | Host Interface PM n Data Out Buffer with SMI | WO |
| HIPMnDI[1] | Host Interface PM n Data In Buffer | RO |
| HIPMnDIC[1] | Host Interface PM n Data In Buffer with SCI | RO |
| HIPMnCTL[1] | Host Interface PM n Control | R/W |
| HIPMnIC[1] | Host Interface PM n Interrupt Control | R/W |
| HIPMnIE[1] | Host Interface PM n Interrupt Enable | R/W |

1. Where n stands for register 1 or 2.

### Host Interface PM n Status Register (HIPMnST)

The HIPMnST register contains the status of the host interface PM channel buffers (DBBIN and DBBOUT). It also provides a means for the PC87591L-N05 to send status bits to the host. This register is read by a host processor read operation from address $66_{16}$. HIPMnST is cleared on reset.

Location: Channel 1 - 00 $FEAC_{16}$

Channel 2 - 00 $FEBE_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **ST3-ST0** | | | | **A2** | **F0** | **IBF** | **OBF** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Type | Description |
|---|---|---|
| 0 | RO | **OBF (Output Buffer Full).** The bit is set when the PM channel’s DBBOUT is written to by the core (writing to HIPMnDO, HIPMnDOM or HIPMnDOC register). The bit is cleared by a host processor read of the output buffer ($62_{16}$). Writing to this bit is ignored. |
| 1 | RO | **IBF (Input Buffer Full).** The bit is set when the PM channel’s DBBIN is written to by the host processor (writing to either address $62_{16}$ or address $66_{16}$). The bit is cleared by a core read of the PM input buffer (HIPMnDI or HIPMnDIC). |
| 2 | R/W | **F0 (Flag 0).** General-purpose flag that can be set or cleared by the core firmware. |
| 3 | RO | **A2 (A2 Address).** Indicates whether the last write operation of the host to the PMn channel was to the data register or the Command register. Writing to this bit is ignored.<br>0: Last write was to the data register (pointed to by configuration register index $60_{16}$ and $61_{16}$) (default)<br>1: Last write was to the command register (pointed to by configuration register index $62_{16}$ and $63_{16}$) |
| 7-4 | R/W | **ST3-ST0 (Status).** Four general-purpose flags that can be used for signaling between the host and core. When used as an embedded controller interface channel for ACPI, a predefined meaning is assigned to ST0, ST1 and ST2. The standard meaning is BURST, SCI event and SMI event, respectively. |

# 5.0 Host Controller Interface Modules (Continued)

## Host Interface PM n Data Out Buffer (HIPMnDO)

The HIPMnDO register allows the core firmware to write to the PM port DBBOUT register while setting the PM port OBF bit in the Status register. If enabled, an IRQ11 (and/or SCI and/or SMI, in PC87570 Compatible mode) interrupt is sent at that time. If the core interrupt on PM port output buffer empty is enabled, writing to HIPMnDO de-asserts it (low).

Location: Channel 1 - 00 $FEAE_{16}$
Channel 2 - 00 $FEC0_{16}$

Type: WO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **PM Channel DBBOUT Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **PM Channel DBBOUT Data.** |

## Host Interface PM n Data Out Buffer with SCI (HIPMnDOC)

The HIPMnDOC register has the same function as the HIPMnDO register. In addition, it generates an SCI interrupt when OBF is set and hardware SCI generation is enabled.

Location: Channel 1 - 00 $FEB2_{16}$
Channel 2 - 00 $FEC4_{16}$

Type: WO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **PM Channel DBBOUT Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **PM Channel DBBOUT Data.** |

## Host Interface PM n Data Out Buffer with SMI (HIPMnDOM)

The HIPMnDOM register has the same function as the HIPMnDO register. In addition, it generates an SMI interrupt when OBF is set and hardware SMI generation is enabled.

Location: Channel 1 - 00 $FEB4_{16}$
Channel 2 - 00 $FEC6_{16}$

Type: WO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **PM Channel DBBOUT Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **PM Channel DBBOUT Data.** |

# 5.0 Host Controller Interface Modules (Continued)

## Host Interface PM n Data In Buffer (HIPMnDI)

The HIPMnDI register allows the core firmware to read the PM port DBBIN register while clearing the PM port IBF bit in the Status register. If the core interrupt on IBF for the PM channel is enabled, reading from HIPMnDI de-asserts it (low).

Location: Channel 1 - 00 $FEB0_{16}$
Channel 2 - 00 $FED2_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **PM Channel DBBIN Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **PM Channel DBBIN Data.** |

## Host Interface PM n Data In Buffer with SCI (HIPMnDIC)

The HIPMnDIC has the same function as the HIPMnDI register. In addition, it generates an SCI interrupt when IBF is cleared and when hardware SCI generation is enabled.

Location: Channel 1 - 00 $FEB6_{16}$
Channel 2 - 00 $FEC8_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **PM Channel DBBIN Data** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **PM Channel DBBIN Data.** |

# 5.0 Host Controller Interface Modules (Continued)

## Host Interface PM n Control Register (HIPMnCTL)

The HIPMnCTL register controls the operation mode and configuration of the PM channel. It includes the Enhanced mode enable bit and control bits for Enhanced mode operation. HIPMnCTL is $40_{16}$ on reset.

Location: Channel 1 - 00 $FEB8_{16}$

Channel 2 - 00 $FECA_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **EME** | **SCIPOL** | **PLMS** | | | **Reserved** | **OBEIE** | **IBFIE** |
| Reset | see text | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **IBFIE (Input Buffer Full Interrupt Enabler).**<br>0: IBF interrupt to the core is disabled (default)<br>1: Enables an interrupt to the core when IBF in HIPMnST register is set |
| 1 | **OBEIE (Output Buffer Empty Interrupt Enable).**<br>0: OBF interrupt to the core is disabled (default)<br>1: Enables an interrupt to the core when OBF in HIPMnST register is set |
| 2 | **Reserved.** |
| 5-3 | **PLMS (Pulse Level Mode SCI).** Sets the hardware-controlled SCI signal mode to be Level or Pulse and sets the pulse width.<br>When PLMS = $000_2$, the SCI signal functions in Level mode. In this mode, the SCI pulse shaper output value is low, and a high level is set to issue an interrupt (the respective OBF is set).<br>When PLMS ≠ 0, the host interrupts are in Pulse mode. In this mode, the SCI pulse shaper output value is low, and it toggles high to issue an interrupt (i.e., when the respective output buffer register is written).<br>The pulse widths are:<br>**Bits 5 4 3 Pulse Width**<br>0 0 0: Level interrupt (default)<br>0 0 1: 1-Cycle Pulse<br>0 1 0: 2-Cycle Pulse<br>0 1 1: 4-Cycle Pulse<br>1 0 0: 8-Cycle Pulse<br>1 0 1: 16-Cycle Pulse<br>Other: Reserved |
| 6 | **SCIPOL (SCI Negative Polarity).**<br>0: SCI output inactive value is low and its active (asserted) value is high<br>1: Inverted polarity is used. When SCIPOL is set, the SCI signal is the inverse of either what is stored in SCIB or the output of the SCI pulse shaper (default)<br>This bit affects the SCI signal polarity in both PC87570 Legacy and Enhanced modes. |
| 7 | **EME (Enhanced Mode Enable).**<br>0: PM channel is used in Legacy mode. HIPMnST status bits are controlled by writes to the bit value, and interrupts are controlled by HICTRL and HIIRQC register bits (default for HIPM1CTL).<br>1: Enables enhanced control of the PM channel. The bits in HICTRL and HIIRQC registers are ignored in this case (default for HIPM2CTL).<br>In HIPM2CTL (i.e., for PM channel 2), EME is a read-only bit that holds the value 1. Writes to this bit are ignored. |

# 5.0 Host Controller Interface Modules (Continued)

## Host Interface PM n Interrupt Control Register (HIPMnIC)

The HIPMnIC register and its bits affect operation in Enhanced mode only (i.e., when EME bit in HIPMnCTL register is set). In PC87570 Legacy mode, the bits in this registers are ignored. HIPMnIC controls the PM n interrupt signals mode of operation. HIPMnCTL is $41_{16}$ on reset.

Location: Channel 1 - 00 $FEBA_{16}$

Channel 2 - 00 $FECC_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **SCIIS** | **SMIPOL** | **PLMM** | | | **SCIB** | **SMIB** | **IRQB** |
| Reset | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Description |
|---|---|
| 0 | **IRQB (Host Interrupt Request Control Bit).** When the IRQ signal is configured for direct control by the firmware (HIRQE in HIPMnIE register is 0), IRQB bit is output to the PMnIRQ signal. When read, IRQB bit returns the current value of the PMnIRQ signal. IRQn signal's value can be read regardless of the state of HIRQE in HIPMnIE register. |
| 1 | **SMIB (Host SMI Request Control Bit)**. When the SMI signal is configured for direct control by the firmware (HSMIE in HIPMnIE register is 0), SMIB bit is output to the PMnSMI signal (if SMIPOL=0, SMIB is output; if SMIPOL=1, SMIB is inverted before output). When read, SMIB bit returns the current value of the SMI pin. The SMI signal's value can be read regardless of the state of HSMIE in HIPMnIE register. |
| 2 | **SCIB (Host SCI Request Control Bit).** When the SCI signal is configured for direct control by the firmware (HSCIE in HIPMnIE register is 0), SMIB bit is output to the PMnSCI signal (if SCIPOL=0, SCIB is output; if SCIPOL=1, SCIB is inverted before output). When read, SCIB bit returns the current value of the SCI pin. The ECSCI signal value can be read regardless of the state of HSCIE bit in HIPMnIE register. |
| 5-3 | **PLMM (Pulse Level Mode SMI).** Sets the hardware-controlled SMI signal mode to Level or Pulse and sets the pulse width.<br>When PLMM = $000_2$, the SCI signal functions in Level mode. In this mode, the SMI pulse shaper output value is low, and a high level is set to issue an interrupt (i.e., the respective OBF is set).<br>When PLMM $\neq$ 0, the host interrupts are in Pulse mode. In this mode, the SMI pulse shaper output value is low, and it toggles high to issue an interrupt (i.e., when the respective output buffer register is written).<br>The pulse widths are:<br>**Bits 5 4 3 Pulse Width**<br>0 0 0: Level interrupt (default)<br>0 0 1: 1-Cycle Pulse<br>0 1 0: 2-Cycle Pulse<br>0 1 1: 4-Cycle Pulse<br>1 0 0: 8-Cycle Pulse<br>1 0 1: 16-Cycle Pulse<br>Other: Reserved |
| 6 | **SMIPOL (SMI Negative Polarity).**<br>0: SMI output inactive value is low and its active (asserted) value is high<br>1: Inverted polarity is used. When SMIPOL is set, the SMI signal is either the inverse of what is stored in SMIB or the output of the SMI pulse shaper (default)<br>This bit affects the SMI signal polarity in both PC87570 Legacy and Enhanced modes |
| 7 | **SCIIS (SIC on IBF Start).** A write of 1 to this bit starts an SCI interrupt on IBF cleared. A write of 0 to SCIIS is ignored. When read, this bit always return 0. |

# 5.0 Host Controller Interface Modules (Continued)

## Host Interface PM n Interrupt Enable Register (HIPMnIE)

The HIPMnIE register controls the PM n interrupt signals that enable SMI, SCI and IRQ interrupts. HIPMnIE is cleared on reset.

Location: Channel 1 - 00 $FEBC_{16}$
Channel 2 - 00 $FECE_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **HSMIE** | **HSCIE** | **HIRQE** | **SMIE** | **SCIE** | **IRQE** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **IRQE (IRQ Enable).**<br>0: PMnIRQ signal assumes its default value (low) and no interrupts are issued (default)<br>1: Enables PM generation of IRQ events |
| 1 | **SCIE (SCI Enable).**<br>0: PMnSCI signal assumes its default value (high) and no interrupts are issued (default)<br>1: Enables PM generation of SCI events |
| 2 | **SMIE (SMI Enable).**<br>0: PMnSMI signal assumes its default value (high) and no interrupts are issued (default)<br>1: Enables the generation of SMI events by this module |
| 3 | **HIRQE (Hardware IRQ Enable).** Works only in Enhanced PM mode.<br>0: IRQB bit of HIPMnIC register controls the value of the IRQ (default)<br>1: Enables the generation of IRQ events by hardware control based on the status of the OBF flag |
| 4 | **HSCIE (Hardware SCI Enable).** Works only in Enhanced PM mode.<br>0: SCIB bit in HIPMnIC register controls the value of the SCI (default)<br>1: Enables the generation of SCI events by hardware control based on the status of the OBF and IBF flags |
| 5 | **HSMIE (Hardware SMI Enable).** Works only in Enhanced PM mode.<br>0: SMIB bit in HIPMnIC register controls the value of the SMI (default)<br>1: Enables the generation of SMI events by hardware control based on the status of the OBF flag |
| 7-6 | **Reserved.** |

# 5.0 Host Controller Interface Module (Continued)

## 5.3 SHARED MEMORY AND PROTECTION

The PC87591L-N05 off-chip expansion memory can be shared by the host and the core. It may also be used by the host for BIOS code storage or other purposes. The off-chip expansion memory resides in the core domain. In IRE and OBD environments, it is accessible via the core bus. For host accesses, the expansion memory is mapped to the host memory address space via the host interface, and a bridge is provided between the host bus and the core bus. The bridge functionality includes:

- Memory mapping between host domain address space and core domain address space
- Host bus to core bus transaction bridging
- Locking mechanism between host and core domains to maintain coherence of off-chip expansion memory contents during updates
- Read/write protection on host accesses to the off-chip expansion memory
- Host-accessible control and status registers of off-chip expansion memory
- Signaling interface for host-core communication associated with memory updates

### 5.3.1 Host Bus to Core Bus Access Translation

A core bus transaction is generated for each of the following types of host bus transactions:

- 8-bit memory read/write
- 8-bit FWH memory read/write
- 8-bit indirect read/write transactions, using I/O read/write to access the shared expansion memory (see Section 5.3.3 on page 265)

Memory and FWH memory read/write transactions drive Long Wait on the Sync field until the transaction is completed on the core bus. Section 5.3.3 on page 265 describes the Sync field for indirect memory read/write transactions. Section 5.3.5 on page 266 describes the behavior for restricted accesses.

The host bus transaction is forwarded to the core bus after the following is done:

- Address is translated.
- The translated address and the access type are verified to be both:
  - In core domain's expansion memory space
  - Unprotected
- For writes, the HLOCK bit in SMCCST register must be set.

Note that host bus read transactions are translated to read transactions on the core bus, and host bus write transactions are translated to write transactions on the core bus. Translated reads and writes behave the same as reads and writes by the core.

### 5.3.2 Memory Mapping and Host Address Translation

Section 6.1.11 on page 311 describes in detail the host domain addresses for which the core bus generates transactions. In general, the BIOS memory on the host bus can occupy one of three regions in the memory space (see Table 50 on page 311).

Address translation between the host and the core domains is performed for host memory and FWH memory transactions. The 32-bit address received from the host bus is used to decode the different zones, as described in Section 6.1.11 on page 311. The address is then translated to the core bus address using the following rules:

- Legacy and Extended Legacy BIOS Range
  Handle only when enabled (see Section 6.1.11 on page 311 for the enabling alternatives); otherwise, transactions to this zone are ignored. The address is converted to a shared memory internal address as follows:

  SM_Host_Address[31-0] = {1111 1111 1111 111, Host_Memory_Address[16-0]}
- User Defined Shared Memory Space
  This address range is handled only when enabled (see Section 6.1.11 on page 311 for the enabling alternatives); otherwise, transactions to this zone are ignored. The address translation depends on the window size defined. When the window size is $2^n$ bytes, the lower 'n' bits are taken from the memory address, and the upper 32 – n bits of the LPC address are replaced with 1. The address is converted to an internal address as follows:

  SM_Host_Address[31-0] = {1111 ... ... 1, Host_Memory_Address[(n-1)-0]}
- 386 Mode-Compatible BIOS Range
  This address range is handled only when enabled (see Section 6.1.11 on page 311 for the enabling alternatives); otherwise, transactions to this zone are ignored. The address is converted to an internal address as follows:

  SM_Host_Address[31-0] = Host_Memory_Address[31-0]
- Indirect Memory Address
  This address specified in IMA3-0 is used as follows:

  SM_Host_Address[31-0] = {IMA3[7-0], IMA2[7-0], IMA1[7-0], IMA0[7-0]}

## 5.0 Host Controller Interface Modules (Continued)

- For allowed addresses, the SM_Host_Address is translated to a core address using the following equation (a 21-bit address in the core address space is generated by adding the host memory address to the MBTA):

  CR_Space_Address[20-0] = (21 least significant bits)(SM_Host_Address + MBTA)

  MBTA is the size of the PC87591L-N05 on-chip ROM memory (4 Kbytes only), as defined in the Shared Memory Main Block Top Address register (SMCTA). This value is defined on reset to indicate the available memory size. In DEV environment, this value may be changed to allow code development for other memory sizes.
- The CR_Space_Address[20-0] is checked against the Host-Controlled Access Protection registers, the Core-Controlled Override Protection registers and general address space access limitations (i.e., space not mapped to the expansion memory).

Figure 90 shows the address translation scheme for shared memory transactions.

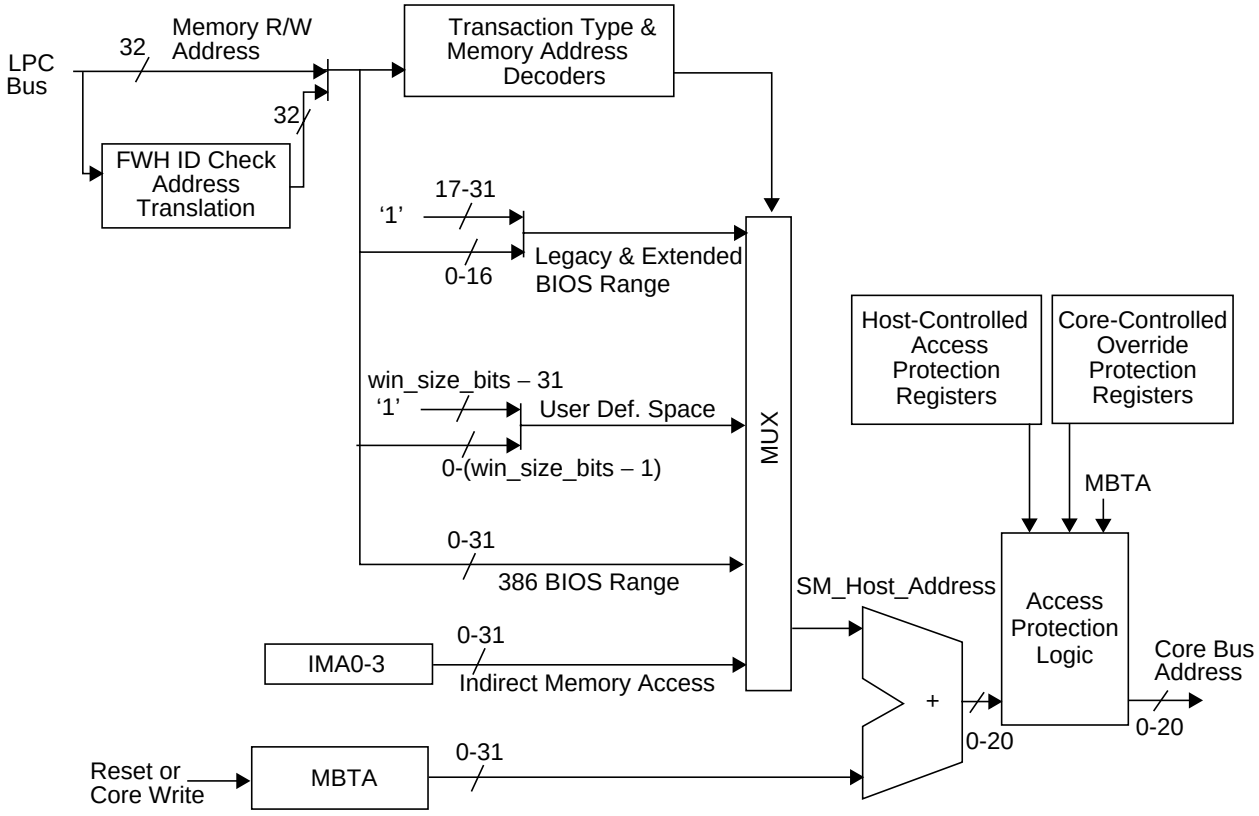



**Figure 90. Address Translation Mechanism for Extended Memory**

Access restrictions are based on the contents of the host-controlled and core-controlled access protection registers. The access protection logic may prevent read and/or write access to addresses in the core address space. The core-controlled register setting should always prevent host access to addresses that are not in the core domain's expansion memory space. Note that the resulting memory space is not continuous.

In DEV environment, the value of MBTA may be changed for ease of software development. The memory space between 00 $E000_{16}$ and 00 $FFFF_{16}$ is not accessible by the host.

## 5.0 Host Controller Interface Module (Continued)

The following figures illustrate the mapping function that results from the address translation function for a Shared BIOS scheme (Figure 91) and when the expansion memory is mapped as a non-BIOS block of memory (Figure 92).

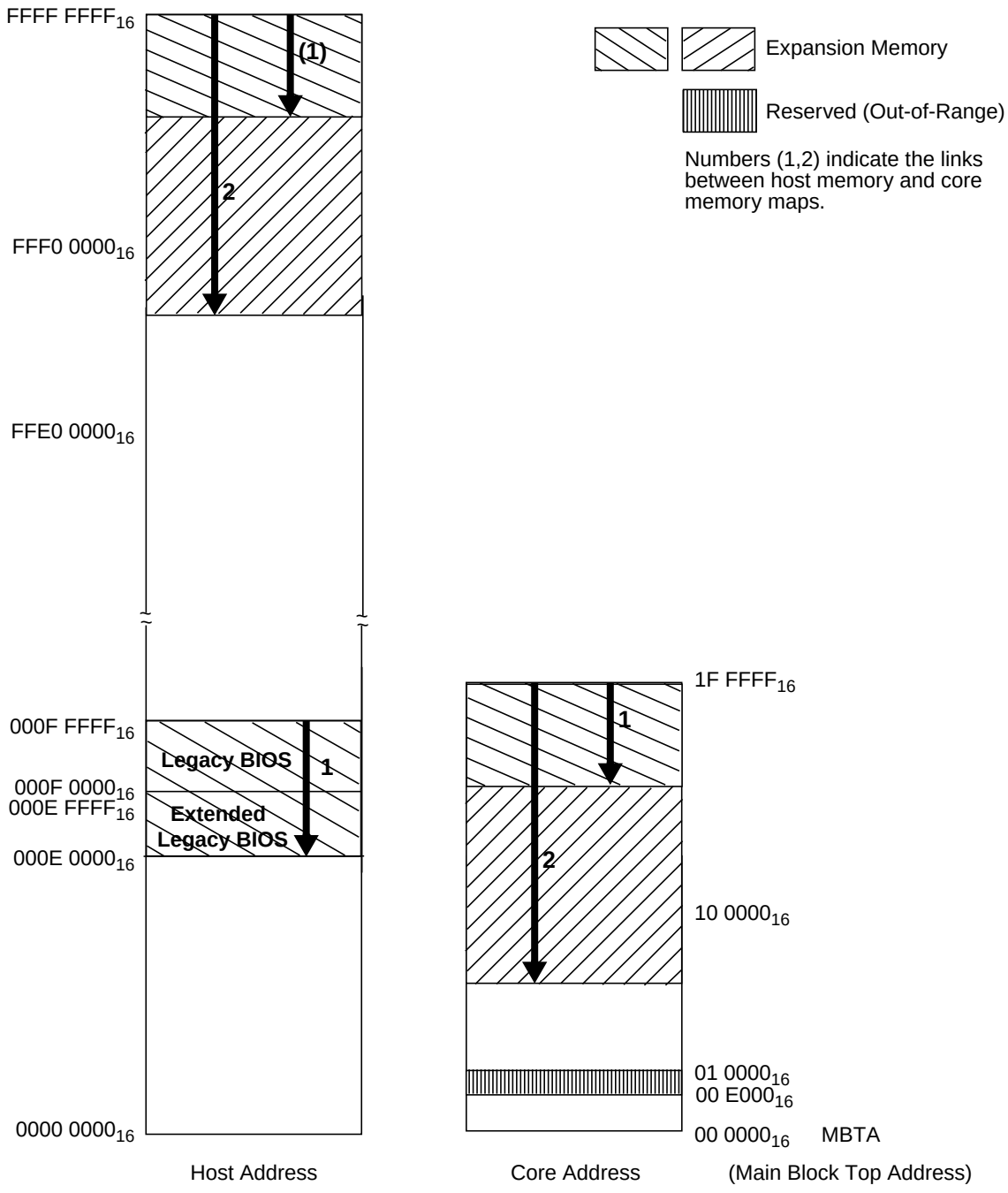



**Figure 91. Host to Core Address Translation: 386 Mode-Compatible BIOS Range**

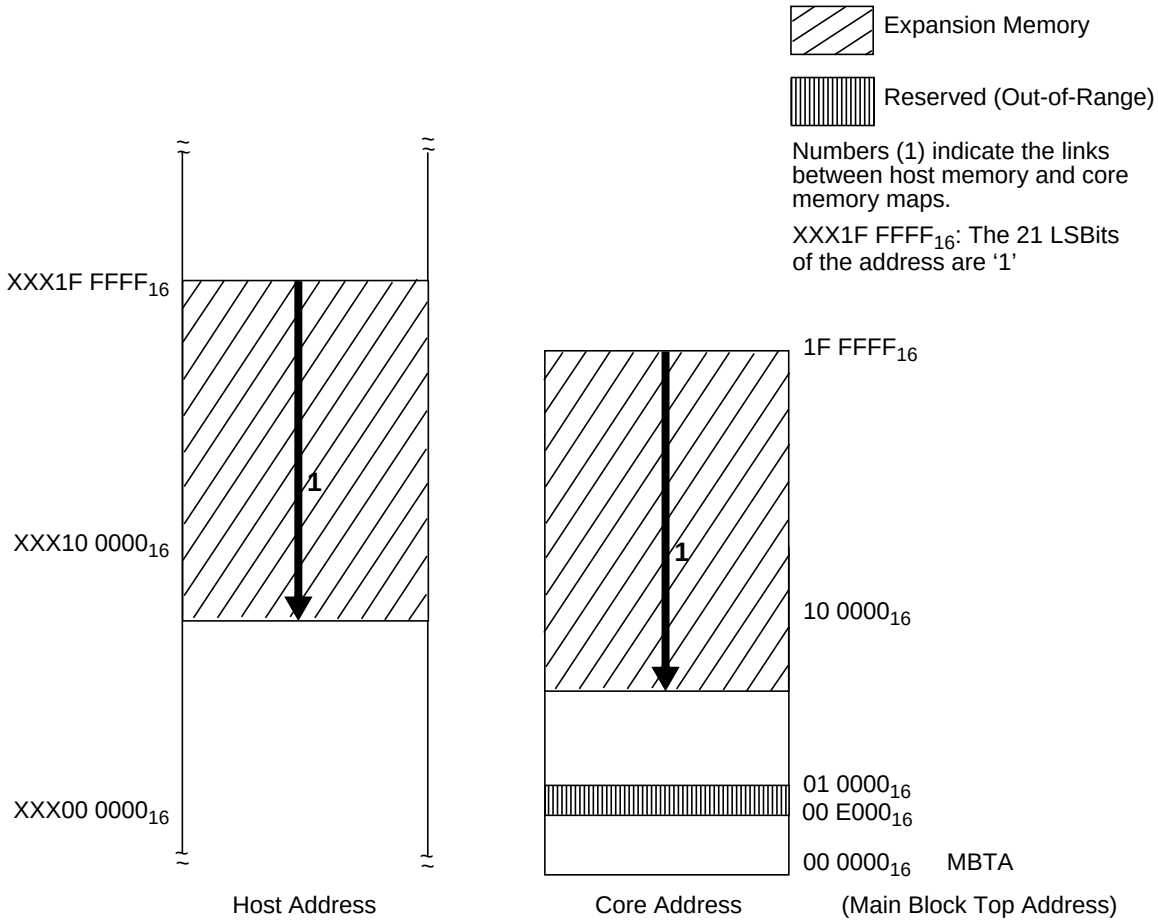



**Figure 92. Host to Core Address Translation: Non-BIOS Mode**

### 5.3.3 Indirect Memory Read and Write Transaction

The following I/O mapped registers can be utilized instead of memory mapping to perform a core bus transaction using an LPC I/O transaction:

- Four Indirect Memory Address registers (IMA3-IMA0), representing host address bits 31 to 0
- One Indirect Memory Data register (IMD), representing data bits 7 to 0

An LPC I/O write to the IMD register triggers a core bus memory write cycle using the addresses and data from IMA3-IMA0 and IMD registers, respectively. The LPC I/O write is completed when the core bus transaction is completed.

An LPC I/O read cycle from IMD register triggers a core bus memory read cycle using the addresses from IMA3-IMA0. The data returned from the core bus cycle is used to complete the LPC I/O read cycle from IMD register.

Read/write cycles from/to IMA3-IMA0 registers drive Short Wait on the Sync field. Read/write cycles from/to IMD register drive Long Wait on the Sync field until a transaction is actually performed and completed on the core bus.

Indirect memory read/write transactions are subject to the same memory mapping, locking mechanism and host access protection as memory and FWH memory read/write transactions. For more details, see Sections 5.3.2, 5.3.4 and 5.3.5.

### 5.3.4 Locking Between Domains

For read operations, hardware handles arbitration between the host and core. For expansion memory program and erase operations, the PC87591L-N05 provides the means for enabling exclusive use for any particular access path over a sequence of operations.

# 5.0 Host Controller Interface Module (Continued)

## 5.3.5 Host Access Protection

The host read/write protection is software controlled via a set of registers accessible to the host. The protection granularity is per block. Each of the 32 software-controlled protection blocks is 64 Kbytes; the block's read protection and write protection flags may be set independently. A Lock Protect flag may be set to prevent future changes to the read and write protection bits. Once locked, the lock bit and the read/write enable bits may be changed only after Host Domain Hardware reset.

The core can override the host settings and prevent host access to certain areas of the shared memory. The override may be set independently for read and write. In the first 128 Kbytes of address space, each core-controlled block is 8 Kbytes. For the rest of the memory space the blocks are 64 Kbytes each.

The default value of the protection registers is set according to the properties of the block. There are three types of blocks:

- Core Boot Block: Read, Erase and Program protected from the host (in PC87591L-N05 the core boot block is implemented in ROM).
- Host Boot Block: Open for Read by the host; Erase and Program protected from the host.
- Other Blocks: Open for Read, Erase and Program by the host.

Core on-chip peripherals and RAM are never accessible to the host (for both read and write). The range from MBTA to 0 $FFFF_{16}$ should be protected from host access using the core-controlled protection registers.

### Core Boot Block

This block is not accessible by the host and (for either read or write). The core boot block starts at address 0 $0000_{16}$ and ends as defined in the Core Boot Block field of the PTWRL register (4 Kbytes). The core boot block access protection settings (in SMCO*x*P0-2 registers) may not be changed, and the respective protection bits are read only.

### Host Boot Block

By default, this block may be read by the host. Host writes to this block are always disabled. The host boot block size is 64 Kbytes and is available when the Host Boot Block bit in PTWRL register is 0. The Host boot block is located at the upper 64 Kbytes of the core memory space (1F $0000_{16}$ to 1F $FFFF_{16}$), based on both the core address folding at 2 Mbyte boundaries and on the MBTA value forced to 0 $0000_{16}$ (the latter by Force MBTA Zero bit in PTWRL register). In case of an overlap between the host boot block and the core boot block, access protection settings of the core boot block are used. The host boot block access protection settings may not be changed, and the respective protection bits are read only.

### Other Blocks

By default, all other blocks are read and write protected. The core may enable host read and/or write access to these blocks.

### Setting the Host Access Protection Flags

There are two sets of host access protection flags, as shown in Figure 93:

- Host-controlled host access protection flags
- Core-controlled host access protection flags

**Host-Controlled Host Access Protection Flags.** For each of the 32 protection blocks there is a set of three bits (flags): Read Protect, Write Protect and Lock Protect. The 32 sets of flags are accessible via two registers, Shared Memory Host Access Protect Register 1 and 2 (SMHAP1-2), using an indexing scheme.

The Host Block index may be calculated using the following equation:

Host_Block_Index = CR_Space_Address[20-0] / 64K or
Host_Block_Index = (21 least significant bits)(SM_Host_Address + MBTA) / 64K

See Section 5.3.2 on page 262 for the definitions of the host address translation.

- To change a flag setting, write the new flag setting, together with the required index field (i.e., Host Access Protection Index) and a cleared Index Write bit, to the appropriate register (SMHAP1 or SMHAP2).
- To read the values of the flags:
  1. Read the value of the register and save the index field.
  2. Write the index of the register's flag (i.e., write the index with a 1 in the Index Write bit).
  3. Read the settings of the register's flag.
  4. Restore the index field by writing back the value of the index field stored in step 1.

**Core-Controlled Host Access Protection Flags.** For core-controlled host access protection there is a read protect and write protect bit for each block The core block number are parallel to the host blocks for blocks 2-31. The core-controlled host access protection has a finer granularity for the first two host blocks, which are split into 16 core blocks, indicated as LA0 - LA15.

The block number may be calculated using the following equation:

Core_Block_Number = (CR_Space_Address < 128K) ?
CR_Space_Address[20-0] / 64K : CR_Space_Address[20-0] / 8K

## 5.0 Host Controller Interface Modules (Continued)

The Shared Memory Core Override Read Protect 0-2 registers and Shared Memory Core Override Write Protect 0-2 registers provide core access to the read and write protect bits, respectively. Bits in these registers may be set and cleared by writing to the registers.

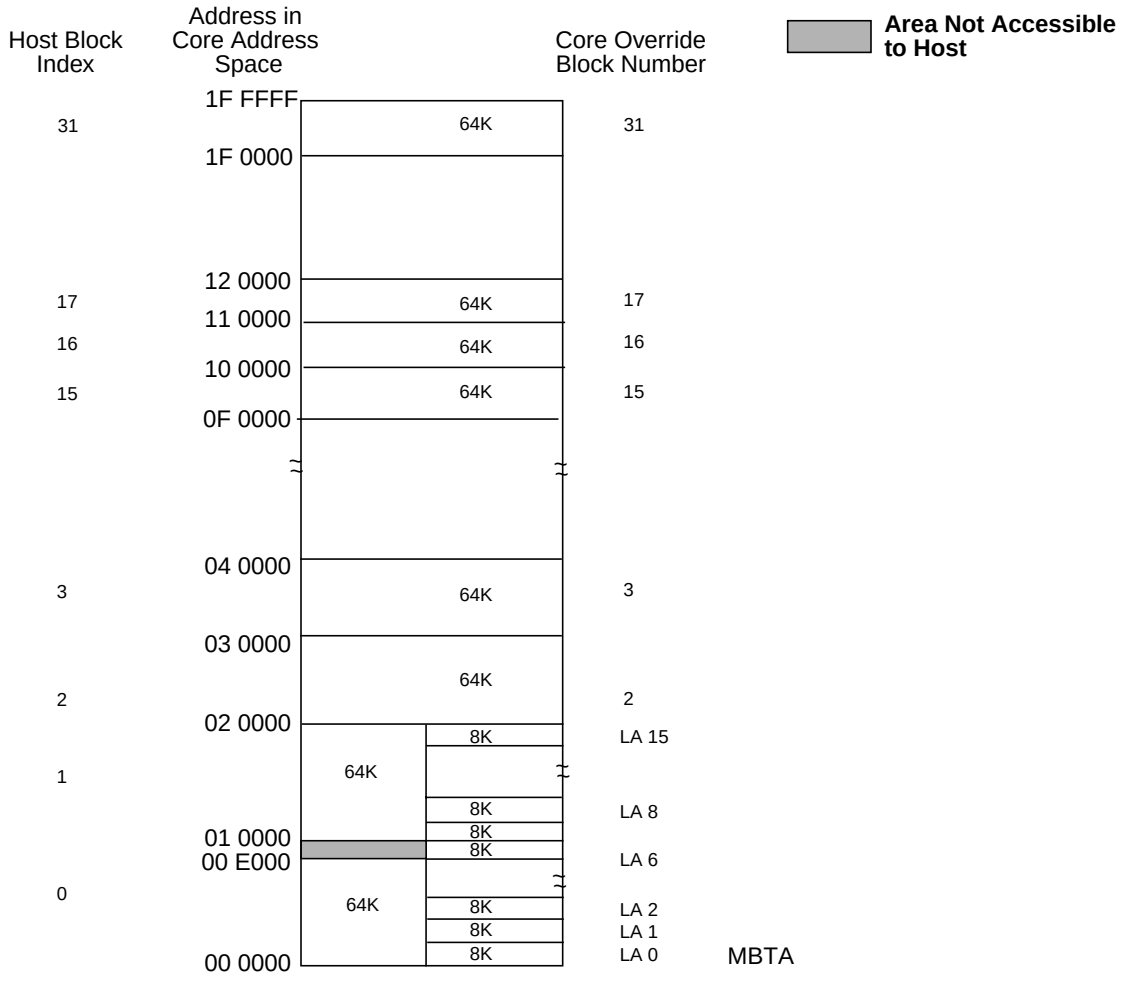



**Figure 93. Protection Blocks for Protection Bits and Override Protection Bits**

### Response to a Restricted Access

A restricted access is an access that complies with at least one of the following conditions:

- The access type (i.e., Read or Write) of the translated host address (i.e., core address) is protected.
- A reset to the core domain due to a Warm reset event is set (Core Boot Block field of PTWRL register is $1111_2$).

The PC87591L-N05 responds to a restricted access by generating an interrupt (if enabled by HERRIEN bit in SMCCST register) to the core. Two status bits (HWERR and HRERR) indicate whether the restricted access is a read or write access. The response on the host bus is according to the HERES field.

For restricted write accesses: Data written is ignored; when the HERES field is $10_2$, the read or write transaction is completed with an error SYNC; otherwise, it is completed with a ready SYNC.

For restricted read accesses: When the HERES field is $00_2$, the read or write transaction drives Long Wait (endlessly, unless the access becomes unrestricted or the HERES field is changed). When the field is $01_2$, the PC87591L-N05 completes the transaction with a ready SYNC and data of $00_{16}$; when the field is $10_2$, it completes the transaction with an error SYNC and data of $00_{16}$.

# 5.0 Host Controller Interface Module (Continued)

## 5.3.6 Signaling Interface

For memory update operations, the host may need to exchange signals (indicating the state) with the core. The Signaling interface supports this scheme. The scheme is designed mainly for polling-based operations, but it allows the host to interrupt the core.

The signaling hardware includes an 8-bit register that may be read by both host and core. The host may modify bits 0 through 3 of this register; the core may modify bits 4 through 7. A host write to the register sets the Host Semaphore Write (HSEMW) bit in the Shared Memory Core Control and Status register (SMCCST); if the interrupt enable bit (HSEMIE in SMCCST register) is set, the shared memory interrupt to the core is set.

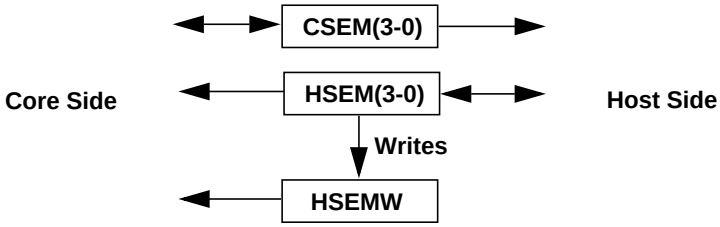



**Figure 94. Signaling Interface**

## 5.3.7 Shared Memory Host Registers

The following set of registers is accessible only by the host. The registers are maintained by $V_{DD}$.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

**Shared Memory Host Register Map**

| Offset | Mnemonic | Register Name | Type |
|---|---|---|---|
| $00_{16}$ | SMIMA0 | Shared Memory Indirect Memory Address 0 | R/W |
| $01_{16}$ | SMIMA1 | Shared Memory Indirect Memory Address 1 | R/W |
| $02_{16}$ | SMIMA2 | Shared Memory Indirect Memory Address 2 | R/W |
| $03_{16}$ | SMIMA3 | Shared Memory Indirect Memory Address 3 | R/W |
| $04_{16}$ | SMIMD | Shared Memory Indirect Memory Data | R/W |
| $07_{16}$ | SMHAP1 | Shared Memory Host Access Protect 1 | Varies per bit |
| $08_{16}$ | SMHAP2 | Shared Memory Host Access Protect 2 | Varies per bit |
| $0C_{16}$ | SMHSEM | Shared Memory Host Semaphore | Varies per bit |

### Shared Memory Indirect Memory Address Register 0 (SMIMA0)

This register defines the addresses 7-0 for a read or write transaction to the memory.

Location: Offset $00_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Indirect Memory Address 7-0** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Indirect Memory Address 7-0.** |

## 5.0 Host Controller Interface Modules (Continued)

### Shared Memory Indirect Memory Address Register 1 (SMIMA1)

This register defines addresses 15-8 for a read or write transaction to the memory.

Location: Offset $01_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Indirect Memory Address 15-8** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Indirect Memory Address 15-8.** |

### Shared Memory Indirect Memory Address Register 2 (SMIMA2)

This register defines addresses 23-16 for a read or write transaction to the memory.

Location: Offset $02_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Indirect Memory Address 23-16** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Indirect Memory Address 23-16.** |

### Shared Memory Indirect Memory Address Register 3 (SMIMA3)

This register defines addresses 31-24 for a read or write transaction to the memory.

Location: Offset $03_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Indirect Memory Address 31-24** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Indirect Memory Address 31-24.** |

### Shared Memory Indirect Memory Data Register (SMIMD)

This register defines data bits 7-0 for a read or write transaction to the memory.

Location: Offset $04_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Indirect Memory Data 7-0** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Indirect Memory Data 7-0.** |

# 5.0 Host Controller Interface Module (Continued)

## Shared Memory Host Access Protect Register 1 and 2 (SMHAP1-2)

This register holds the read/write protection and lock control from the host side to the shared memory. The memory is partitioned into 64 Kbyte blocks. SMHAP1 controls the first 16 blocks (addresses 0-1 Mbyte). SMHAP2 controls the second group of 16 blocks (addresses 1-2 Mbyte). The block mapping is in the core address space. See “Setting the Host Access Protection Flags” on page 266 for the calculation method of the block address in the host address space. On Host Domain Hardware reset, all write-protect flags are set and all lock-protect and read-protect flags are cleared.

Location: Offset $07_{16}$ and $08_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Host Access Protection Index** | | | | **Index Write** | **Host Lock Protection** | **Host Write Protection** | **Host Read Protection** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Bit | Type | Description |
|---|---|---|
| 0 | R/W | **Host Read Protection.** The block number is as held in the index field (bits 7-4). Note that the Core Override protection may disable reads even when reads are allowed by this register.<br>0: Host Reads are allowed for this block (default)<br>1: Host Reads are inhibited for this block |
| 1 | R/W | **Host Write Protection.** The block number is as held in the index field (bits 7-4). Note that the Core Override protection may disable writes even when writes are allowed by this register.<br>0: Program and erase are allowed for this block<br>1: Program and erase of the expansion memory are inhibited for this block (default) |
| 2 | R/W | **Host Lock Protection.** The block number is as held in the index field (bits 7-4). When set, the bit prevents changing the values of the Host Read Protect, Host Write Protect and Host Lock Protection bits for this block. Once set, this bit is cleared by Host Domain Hardware reset only.<br>0: Changes to protection bits (0-2) for this block are enabled (default)<br>1: Protection bits (0-2) for this block are locked, and the bits’ values may not be changed |
| 3 | WO | **Index Write.** Indicates that this is an index write transaction; therefore, bits 0-2 of this register are ignored. When read, always returns 0.<br>0: Write transaction affects all fields of this register (writes to bits 0-2 use the newly written index) (default)<br>1: Write transaction for purpose of index update; bits 0-2 should not be updated by this write. |
| 7-4 | R/W | **Host Access Protection Index.** Holds the index number of the host block accessed by the other fields in this register. All blocks are 64 Kbytes. The block index is calculated in the core address space. For details of the address conversion, see Section 5.3.2 on page 262.<br>Index = Block_First_Address / 64K<br>In SMHAP1: $0000\text{-}1111_{16}$ for indexes 0-15, respectively<br>In SMHAP2: $0000\text{-}1111_{16}$ for indexes 16-31, respectively |

## Shared Memory Host Semaphore Register (SMHSEM)

This register provides eight semaphore bits between the core and host. Four of the bits may be set by the host and four may be set by the core. The register is cleared ($00_{16}$) on Host Domain Hardware reset.

Location: Offset $0C_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **CSEM3** | **CSEM2** | **CSEM1** | **CSEM0** | **HSEM3** | **HSEM2** | **HSEM1** | **HSEM0** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Type | Description |
|---|---|---|
| 3-0 | R/W | **HSEM3-0.** Four bits that may be updated by the host and read by both the host and the core. |
| 7-4 | RO | **CSEM3-0.** Four bits that may be updated by the core and read by both the host and the core. |

# 5.0 Host Controller Interface Modules (Continued)

## 5.3.8 Shared Memory Core Registers

The following set of registers is accessible only by the core. These registers are maintained by $V_{CC}$.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

**Shared Memory Core Register Map**

| Mnemonic | Register Name | Type |
|---|---|---|
| SMCCST | Shared Memory Core Control and Status | R/W |
| SMCTA | Shared Memory Core Top Address | RO in IRE and OBD environments; R/W in DEV environment |
| SMHSEM | Shared Memory Host Semaphores | Varies per bit |
| SMCORP0-2 | Shared Memory Core Override Read Protect 0-2 | R/W or RO |
| SMCOWP0-2 | Shared Memory Core Override Write Protect 0-2 | R/W or RO |

### Shared Memory Core Control and Status Register (SMCCST)

This register provides control and status of read/write from/to a restricted address. The register is cleared ($00_{16}$) on reset.

Location: 00 $F900_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **HSEMIE** | **HSEMW** | **HLOCK** | **HERES** | | **HERRIEN** | **HWERR** | **HRERR** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **HRERR (Host Read Error).** The bit is set (1) when the host attempts to read from a read-protected block or out-of-range address. An out-of-range address is an address that the LPC configuration module defines as mapped to the PC87591L-N05, but it is actually translated to a reserved address in the core address space. Writing 1 to this bit clears it to 0. Writing 0 has no effect. |
| 1 | **HWERR (Host Write Error).** The bit is set (1) when the host attempts to write to a read-protected block or out-of-range address. An out-of-range address is an address that the LPC configuration module defines as mapped to the PC87591L-N05, but it is actually translated to a reserved address in the core address space. Writing 1 to this bit clears it to 0. Writing 0 has no effect. |
| 2 | **HERRIEN (Host Error Interrupt Enable).** When set (1) and either the HRERR or HWERR bit is set (1), a core interrupt is generated; otherwise, the core interrupt is inactive. |
| 4-3 | **HERES (Host Error Response).** Controls response type on read/write from/to a protected block or out-of-range address. An out-of-range address is an address that the LPC configuration module defines as mapped to the PC87591L-N05, but it is actually translated to a reserved address in the core address space.<br>**Bits**<br>**4 3 Description**<br>0 0: Drive Long Wait for read; ignore write (default)<br>0 1: Read back $00_{16}$; ignore write<br>1 0: Drive error SYNC for both read and write<br>1 1: Reserved |
| 5 | **HLOCK (Host Lock).**<br>0: The bridge does not generate write transactions on the core bus (default)<br>1: The bridge can generate write transactions on the core bus |
| 6 | **HSEMW (Host Semaphore Write).** The bit is set (1) when the host writes to HSEM register. Writing 1 to this bit position clears it to 0. Writing 0 has no effect. |
| 7 | **HSEMIE (Host Semaphore Interrupt Enable).** When the bit is set (1), the interrupt to the core is set (level high) if HSEMW is set. |

# 5.0 Host Controller Interface Module (Continued)

## Shared Memory Core Top Address Register (SMCTA)

This register provides information about the size of the on-chip main block. The register is loaded with its default value on $V_{CC}$ Power-Up reset only.

Location: 00 $F902_{16}$

Type: RO in IRE and OBD environments
R/W in DEV environment

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | **MBSD** | | | | |
| Reset | 0 | 0 | 0 | 2 (see note in field description) | | | | |

| Bit | Description |
|---|---|
| 4-0 | **MBSD (Main Block Size Definition).** Defines the size of the main block in 64 Kbyte units. Thus the MBTA value is MBSD * 1 $0000_{16}$. Note that MBTA is actually the first address beyond the main block.<br>The reset value of this field is affected by the Force MBTA Zero bit in PTWRL register (see Page 54). When the Force MBTA Zero bit is set, the reset value of this field is $0_{16}$; when the bit is cleared, the reset value is as shown in the bit table, above.<br>This field is loaded on $V_{CC}$ Power-Up reset with the on-chip ROM size. In DEV environment, the MBSD may be loaded with a new value. |
| 7-5 | **Reserved.** |

## Shared Memory Host Semaphore Register (SMHSEM)

This register provides eight semaphore bits between the core and the host. Four of the bits may be set by the host; four may be set by the core. The register is cleared ($00_{16}$) on reset.

Location: 00 $F904_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **CSEM3** | **CSEM2** | **CSEM1** | **CSEM0** | **HSEM3** | **HSEM2** | **HSEM1** | **HSEM1** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Type | Description |
|---|---|---|
| 3-0 | RO | **HSEM3-0.** Four bits that may be updated by the host and read by both the host and the core. |
| 7-4 | R/W | **CSEM3-0.** Four bits that may be updated by the core and read by both the host and the core. |

## Shared Memory Core Override Read Protect Registers 0-2 (SMCORP0-2)

SMCORP0-2 are 16-bit registers that provide core override on the host read protection bits. For the host to be able to read a memory location, both the Host Read Protection bit (controlled through the Shared Memory Host Access Protect Register 1 or 2) and the associated bit in SMCORP0-2 should be cleared. Each bit in this register is associated with a memory block, as described in the bits description. Bits in these registers may be RO or RW depending on their position and the size of the core and host boot blocks. SMCORP0-2 registers are loaded with reset values either on reset or when the value of SMCTA register is changed; the reset values depend on the size of the core and host boot blocks, as defined in PTWRL register.

Location: 00 $F910_{16}$, 00 $F912_{16}$, 00 $F914_{16}$

Type: R/W or RO as described in the description below

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **ORPLA15-0** | | | | | | | | | | | | | | | |
| Reset | See bit description below | | | | | | | | 1 | See bit description below | | | | | | |

## 5.0 Host Controller Interface Modules (Continued)

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **ORP15-2** | | | | | | | | | | | | | | **Reserved** | |
| Reset | See bit description below | | | | | | | | | | | | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **ORP 31-16** | | | | | | | | | | | | | | | |
| Reset | See bit description below | | | | | | | | | | | | | | | |

| Bit | Description |
|---|---|
| 15-0 | **ORPLA15-0 (Override Read Protect Low Addresses 15 through 0).**<br>**ORP15-2 (Override Read Protect 15 through 2).**<br>**ORP31-16 (Override Read Protect 15 through 0).**<br>Each bit affects the host's ability to read from one block. On the low addresses (covered by ORPLAi), the block size is 8 Kbytes. For the other blocks it is 64 Kbytes. The block address is calculated as follows:<br>Low address blocks, ORPLAi: from i*8K to (i+1)*8K–1<br>Other blocks, ORPLj: from j*64K to (j+1)*64K<br>See Figure 93 on page 267 for a description of the block mapping. Bit 7 (ORPLA7) in SMCORP0 register is read only.<br>0: Do not override the host read protect setting for the block<br>1: Host read for this block is disabled regardless of the setting of the respective bit in the host register<br>The reset values of these registers are as follows:<br>Core boot blocks that cover address 00 $0000_{16}$ to CR_Boot_Block_Size: 1<br>Host boot blocks not in above group and from (MBTA – Host_Boot_Block_Size) to MBTA: 0<br>All other blocks: 1<br>The following access limitations apply to the register bits:<br>Core boot blocks that cover address 00 $0000_{16}$ to CR_Boot_Block_Size: RO<br>Host boot blocks not in above group and ranging from (MBTA – Host_Boot_Block_Size) to MBTA: RO<br>All other blocks: RW |

### Shared Memory Core Override Write Protect Registers 0-2 (SMCOWP0-2)

SMCOWP0-2 are 16-bit registers that provide override on the host write protection bits. For the host to be able to write a memory location, both the Host Write Protection bit (controlled through the Shared Memory Host Access Protect Register 1 or 2) and the associated bit in SMCOWP0-2 should be cleared. Each bit in this register is associated with a memory block, as described in the bit description. Bits in these registers may be RO or RW depending on their position and the size of the core and host boot blocks. These registers are cleared on reset and change in value of SMCTA.

Location: 00 $F920_{16}$, 00 $F922_{16}$, 00 $F924_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **OWPLA15-0** | | | | | | | | | | | | | | | |
| Reset | $FFFF_{16}$ | | | | | | | | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **OWP15-2** | | | | | | | | | | | | | | **Reserved** | |
| Reset | $FFFF_{16}$ | | | | | | | | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **OWP 31-16** | | | | | | | | | | | | | | | |
| Reset | $FFFF_{16}$ | | | | | | | | | | | | | | | |

| Bit | Description |
|---|---|
| 15-0 | **OWPLA15-0 (Override Write Protect Low Addresses 15 through 0).**<br>**OWP15-2 (Override Write Protect 15 through 2).**<br>**OWP31-16 (Override Write Protect 15 through 0).**<br>Each bit affects the host's ability to write to one block. On the low addresses (covered by OWPLAi), the block size is 8 Kbytes. For the other blocks it is 64 Kbytes. The block address is calculated as follows:<br>Low Address Blocks, OWPLAi: from i*8K to (i+1)*8K–1<br>Other Blocks, OWPLj: from j*64K to (j+1)*64K<br>See Figure 93 on page 267 for a description of the block mapping. Bit 7 (OWPLA7) in SMCOWP0 register is read only.<br>0: Do not override the host Write Protect setting for the block<br>1: Host writes for this block are disabled regardless of the setting of the respective bit in the host register<br>The following access limitations apply to the register's bits:<br>Core boot blocks that cover address $00\ 0000_{16}$ to CR_Boot_Block_Size: RO<br>Host boot blocks not in above group and ranging from (MBTA – Host_Boot_Block_Size) to MBTA: RO<br>All other blocks: RW |

### 5.3.9 Usage Hints

- Enable Access to the shared memory: Before any access may occur, the shared memory access must be enabled using the SIO Configuration registers (see Section 6.1.11 on page 311). To enable shared memory as a boot device, the SHBM strap should be set appropriately (see Section 2.2.11 on page 45).
- Access to the Host boot block must be enabled by the core after the memory access configuration is completed.
- At different stages of the expansion memory programing by the host, communication between the core and host is required. Various mechanisms may be used for this, one of which is the Shared Memory Semaphore mechanism. This mechanism is tuned for host-initiated operations that use polling on the registers. The core may receive an interrupt or use polling to identify a semaphore change. An example of bit allocation is:

  Bit 0 - Host requests control of expansion memory
  Bit 4 - Core grants control to host

  The sequence is:
  1. Host sets bit 0 to request control of bus.
  2. Core identifies that bit 0 is set and does the required operations, including setting HLOCK bit in SMCCST register to enable host access.
  3. Core sets bit 4, indicating to the host that memory access is granted.
  4. Host performs write/erase to the memory, as required.
  5. Host clears bit 0, indicating completion of the process.
  6. Core clears HLOCK and protects the memory.
  7. Core indicates completion of process by clearing bit 4.

# 5.0 Host Controller Interface Modules (Continued)

## 5.4 CORE ACCESS TO HOST-CONTROLLED MODULES

The PC87591L-N05 enables the core to access the Host-Controlled module registers (e.g., host configuration module, RTC and MSWC), using the SuperI/O Internal Bus (SIB) controller.

**Host-Controlled Module Register Arbitration.** Since the host processor software and the PC87591L-N05 firmware cannot access a Host-Controlled module simultaneously, they must communicate to prevent conflicts in Host-Controlled module register usage.

Access to the Host-Controlled modules is controlled via a lock bit for each module. When the relevant lock bit is cleared, access to the Host-Controlled modules registers by the host processor is enabled. When the relevant lock bit is set, access to the Host-Controlled module registers by the host processor is blocked (i.e., write operations are ignored and read operations return $00_{16}$). Any attempt by the host to access the locked register is flagged by setting the respective bit in SIOLV register.

**SIB Arbitration.** The host and core should access the Host-Controlled modules only after preventing host access to the module (using lock bits, as explained in the previous paragraph). The SIB controller arbitrates SIB usage between the host and core. If a core transaction starts after an LPC transaction (to a different, unlocked module) has started, it waits for the completion of the LPC transaction. If a core transaction starts before an LPC transaction starts, the core transaction finishes before handling the LPC transaction.

The PC87591L-N05 firmware may access the Host-Controlled modules only while the core domain is in Active mode, the Host Domain power plane is on.

**Core Read Operation.** To perform a read operation by the core from a Host-Controlled module register:

1. Set CSAE bit in SIBCTRL register, if not already set.
2. Verify that both CSRD and CSWR bits in SIBCTRL register are cleared.
3. Select the device to be accessed by setting its respective bit in CRSMAE register, if not already set. All other bits in the register must be cleared.
4. Specify the offset of the register in the device in IHIOA register, if not already specified.
5. Write 1 to CSRD bit in SIBCTRL register.
6. Read the CSRD bit in SIBCTRL until it returns 0.
7. Read the data from IHD register.

**Core Write Operation.** To perform a write operation by the core from a Host-Controlled module register:

1. Set CSAE bit in SIBCTRL register, if not already set.
2. Verify that both CSRD and CSWR bits in SIBCTRL register are cleared.
3. Select the device to be accessed by setting its respective bit in CRSMAE register, if not already set. All other bits in the register must be cleared.
4. Specify the offset of the register in the device in IHIOA register, if not already specified.
5. Write the data to IHD register; this starts the write operation to the device.
6. Read the CSWR bit in SIBCTRL until it returns 0; this indicates the completion of the write transaction.

The following sequence is provided for minimal conflict between host and core in the use of Host-Controlled peripherals.

1. After arbitrating the use of the specific Host-Controlled modules with the host, set the corresponding lock bit (see LKSIOHA register).
2. Read and save all Host-Controlled module registers required for proper operation of the host. Beware of destructive reads.
3. After the Host-Controlled module access is complete, restore the Host-Controlled module registers saved in step 2.
4. Clear the corresponding lock bit to allow the host to access the Host-Controlled module.

When accessing the RTC, also:

1. To access locked memory locations in the RTC, set (1) RTCMR bit in SIBCTRL register to clear the RTC lock bits.
2. Access the RTC's CMOS-RAM and its registers. To prevent conflicts with the host software, the firmware should not read any of the RTC read-volatile registers.
3. After the RTC access is complete, restore the RTC address pointer. If the RTC locking was removed, re-lock the RTC memory.

# 5.0 Host Controller Interface Module (Continued)

## 5.4.1 Core Access to Host-Controlled Module Registers

The following set of registers is accessible only by the core. The registers are powered by $V_{CC}$.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

**Core Access to Host-Controlled Modules Register Map**

| Mnemonic | Register Name | Type |
|---|---|---|
| IHIOA | Indirect Host I/O Address | R/W |
| IHD | Indirect Host Data | R/W |
| LKSIOHA | Lock SuperI/O Host Access | R/W |
| SIOLV | SuperI/O Access Lock Violation | R/W1C |
| CRSMAE | Core to SIB Modules Access Enable | R/W |
| SIBCTRL | SIB Control | Varies per bit |

### Indirect Host I/O Address Register (IHIOA)

This register defines the host I/O address for read or write transactions from/to the Host-Controlled modules. The I/O address is an offset from the least significant bits of the logical device address. The accessed device is selected using the Core to SIB Modules Access Enable Register (CRSMAE); see page 278.

Location: 00 $FCE0_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | | | **Indirect Host I/O Offset** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Indirect Host I/O Offset.** Only offsets within the logical device range are allowed. Other offsets may have unpredictable results. |
| 15-8 | **Reserved.** |

### Indirect Host Data Register (IHD)

This register holds host data for read or write transactions from/to the Host-Controlled modules.

Location: 00 $FCE2_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Indirect Host Data** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Indirect Host Data.** |

# 5.0 Host Controller Interface Modules (Continued)

## Lock SuperI/O Host Access Register (LKSIOHA)

This register controls locking of host access to the Host-Controlled modules. All bits of this register, except bit 1, are cleared on reset. The bit 1 reset value is defined by RTC Lock Default bit in “Protection Word Low Register (PTWRL)” on page 54.

Location: 00 $FCE4_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | **LKRTCHA** | **LKCFG** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **LKCFG (Lock Configuration Registers Host Access).**<br>0: Host processor access to the Configuration registers is enabled (default)<br>1: Host processor access to the Configuration registers is blocked |
| 1 | **LKRTCHA (Lock Real-Time Clock (RTC) Host Access).**<br>0: Host processor access to the RTC registers is enabled (default)<br>1: Host processor access to the RTC registers is blocked |
| 15-2 | **Reserved.** |

## SuperI/O Access Lock Violation Register (SIOLV)

This register provides an error indication when a host lock violation occurs on Host-Controlled modules access.

Location: 00 $FCE6_{16}$

Type: R/W1C

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | **RTCLV** | **CFGLV** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **CFGLV (Configuration Register Lock Violation).** The bit is set (1) when the host processor attempts to access the configuration registers while LKCFG bit in LKSIOHA register is set. |
| 1 | **RTCLV (Real-Time Clock (RTC) Lock Violation).** The bit is set (1) when the host processor attempts to access the RTC while LKRTCHA bit in LKSIOHA register is set. |
| 15-2 | **Reserved.** |

# 5.0 Host Controller Interface Module (Continued)

## Core to SIB Modules Access Enable Register (CRSMAE)

This register enables core access to the Host-Controlled modules. Only one of the bits in this register may be set at a time.

Location: 00 $FCE8_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | **RTCAE** | **CFGAE** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | | **MSWCAE** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **CFGAE (Configuration Register Core Access Enable).** This bit enables access to the PnP Configuration Index/Data registers, with A0 of the offset used to differentiate between them. When A0 is 0, the Index register is accessed; when A0 is 1, the Data register is accessed.<br>0: Core access to the Configuration registers is disabled (default)<br>1: Core access to the Configuration registers is enabled |
| 1 | **RTCAE (Real-Time Clock (RTC) Core Access Enable).** The RTC has two chip-select signals defined in its configuration space, each with two registers. A1 of the offset is used to differentiate between the two (when A1 is 0, the pair pointed to by index 60, 61 is accessed; when A1 is 1, the pair pointed to by index 62, 63 is accessed).<br>0: Core access to the RTC registers is disabled (default)<br>1: Core access to the RTC registers is enabled |
| 7-2 | **Reserved.** |
| 8 | **MSWCAE (Mobile System Wake-Up Control (MSWC) Access Enable).**<br>0: Core access to the MSWC registers is disabled (default)<br>1: Core access to the MSWC registers is enabled |
| 15-9 | **Reserved.** |

# 5.0 Host Controller Interface Modules (Continued)

## SIB Control Register (SIBCTRL)

This register allows the core to control the SIB controller operation.

Location: 00 $FCEA_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | **RTCMR** | **CSWR** | **CSRD** | **CSAE** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Type | Description |
|---|---|---|
| 0 | R/W | **CSAE (Core to SIB Access Enabled).**<br>0: Core access to the SIB bus is disabled (default)<br>1: Core access to the SIB bus is enabled. The logical device is selected by the CRSMAE register. |
| 1 | R/W1S | **CSRD (Core Read from SIB).** Writing 1 to this bit starts a read from the SIB; the read is based on the address and enabled device specified in CRSMAE register. A write of 0 to this bit is ignored. This bit is cleared when the read operation is completed, indicating that the data is ready in IHD register. |
| 2 | RO | **CSWR (Core Write to SIB).** The bit is set by a write operation to IHD register. It is cleared when the write to the SIB is completed. |
| 3 | R/W1S | **RTCMR (Real-Time Clock (RTC) Master Reset).** Writing 1 to this bit generates a reset pulse to the RTC module. This bit is cleared by the hardware once the reset pulse is completed. Writing 0 to this bit is ignored. |
| 7-4 | | **Reserved.** |

# 5.0 Host Controller Interface Module (Continued)

## 5.5 MOBILE SYSTEM WAKE-UP CONTROL (MSWC)

The MSWC detects and handles wake-up events from various sources in the host-controlled modules. The MSWC generates interrupts to the host via $\overline{SMI}$ or $\overline{PWUREQ}$, and/or alerts the core, which enables the core to control the wake-up sequence. Since the MSWC is powered by $V_{CC}$, it can operate in low power consumption states. Some MSWC operations depend on the presence of a clock; these functions are not available when the core clock is turned off.

Figure 95 shows the block diagram of the MSWC.

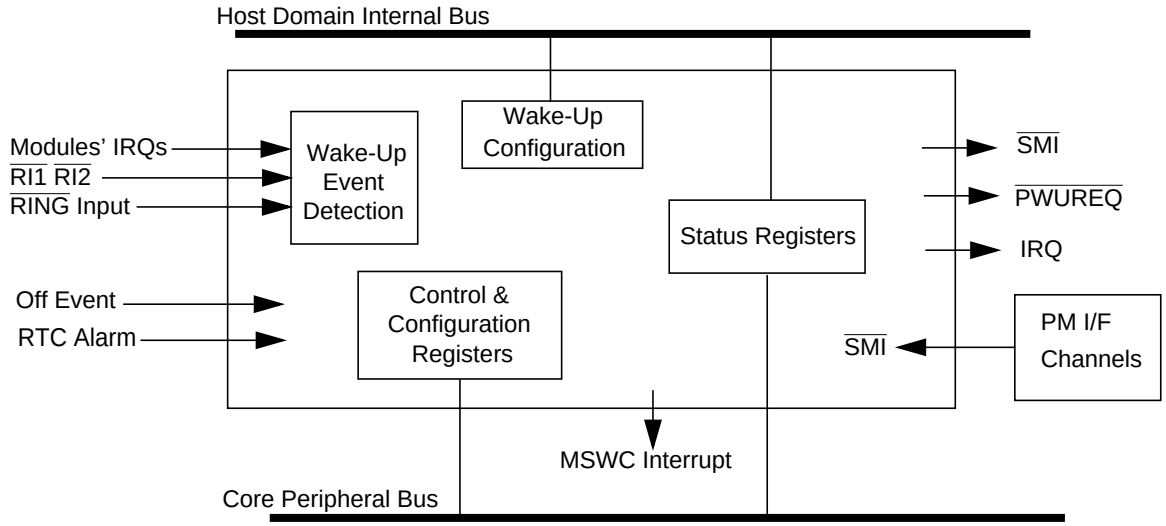



**Figure 95. MSWC Block Diagram**

### 5.5.1 Features

The MSWC recognizes the following maskable system events:

- Modem ring ($\overline{RI1}$ and $\overline{RI2}$ pins)
- Telephone ring ($\overline{RING}$ input pin)
- Wake-up on module IRQs for RTC, KBD and Mouse
- Software events
  - Software triggered wake-up event
  - ACPI power state change indications
  - Software off command

The MSWC notifies the host and/or core when any of the above events occur by asserting one or more of the following output pins:

- Power-Up Request ($\overline{PWUREQ}$)
- System Management Interrupt ($\overline{SMI}$)
- Interrupt to the host (IRQ)
- MSWC interrupt to the core

### 5.5.2 Wake-Up Event Detection and Status Bits

The MSWC monitors various system signals for a wake-up event. When an event is detected, a status bit is set to record it. Each event goes to the Wake-Up Mode Control Logic, which determines its effect (see Figure 97 for an illustration of this mechanism). A set of dedicated registers is used to determine the wake-up criteria, including the RING detection mode.

The following wake-up input events are detected by the MSWC:

- Software events
- IRQ from SuperI/O modules
- Modem Ring ($\overline{RI1}$ and $\overline{RI2}$)
- Telephone Ring ($\overline{RING}$ input)
- ACPI state change
- Legacy off event
- RTC Alarm

## 5.0 Host Controller Interface Module (Continued)

When an input event is detected, the corresponding status bit in both host and core status registers is set to 1, regardless of any Routing Enable bit setting. If both the status bit and a Routing Enable bit corresponding to a specific event are set to 1 (no matter in what order), the output pin corresponding to that Routing Enable bit is asserted.

A status bit is cleared by writing 1 to it. Writing 0 to a status bit does not change its value. Clearing the routing enable bit of an event prevents it from issuing the corresponding system notification (output event) but does not affect the status bit. Figure 97 shows the routing scheme of detected wake-up events to the various means of system notification (i.e., output events).

Both the core and the host have status registers; thus both core and host software can monitor the various event status bits. This enables handling of events via wake-up logic that is implemented as part of the Embedded Controller firmware, and passing the wake-up notifications through the Power Management host-interface protocol. The core uses a mask register (WK_SMIENn) to define which of the status bits it should respond to.

It is recommended that each of the wake-up sources be handled by one handler routine on the host (i.e., SMI, SCI or IRQ triggered) or the core.

**Software Event**

A software event may be used to trigger an interrupt to the host and/or core via software control, as shown in Figure 96.

A software event to the host is active when Software Event Status bit in WK_STS0 register is set. When that status bit is set, WK_EN0 bit 6 enables the generation of an interrupt to the host.

A software event to the core is active when the Software Event Status bit in MSHES0 register is set. When that status bit is set, Bit 6 in MSHEIE0 register enables generation of an interrupt to the core.

The software events are activated (i.e., the status bits in WK_STS0 and MSHES0 registers are set) by writing 1 to the Software Event Status bit in WK_STS0 register when that bit is cleared (the Software Event Status bit in MSHESO is set by a change of the respective bit in WK_STS0, from 0 to 1). The host can activate the software event when $V_{DD}$ is present. The core can activate the software event by accessing the MSWC host registers through the Core Access to Host-Controlled Modules bridge (even when $V_{DD}$ is off).

The software event clearing scheme is defined by HSECM bit in MSWCTL1 register, as follows:
When HSECM bit is cleared, the host Software Event Status bit in WK_STS0 register is cleared by writing 1 to it when it is set (i.e., writing 1 to Host Software Status bit in WK_STS0 register functions as a toggle operation). The core Software Event Status bit in MSHES0 is cleared by writing 1 to it (write 1 to clear). This mode is useful when the software event interrupts the host and is handled by it (Figure 96A).

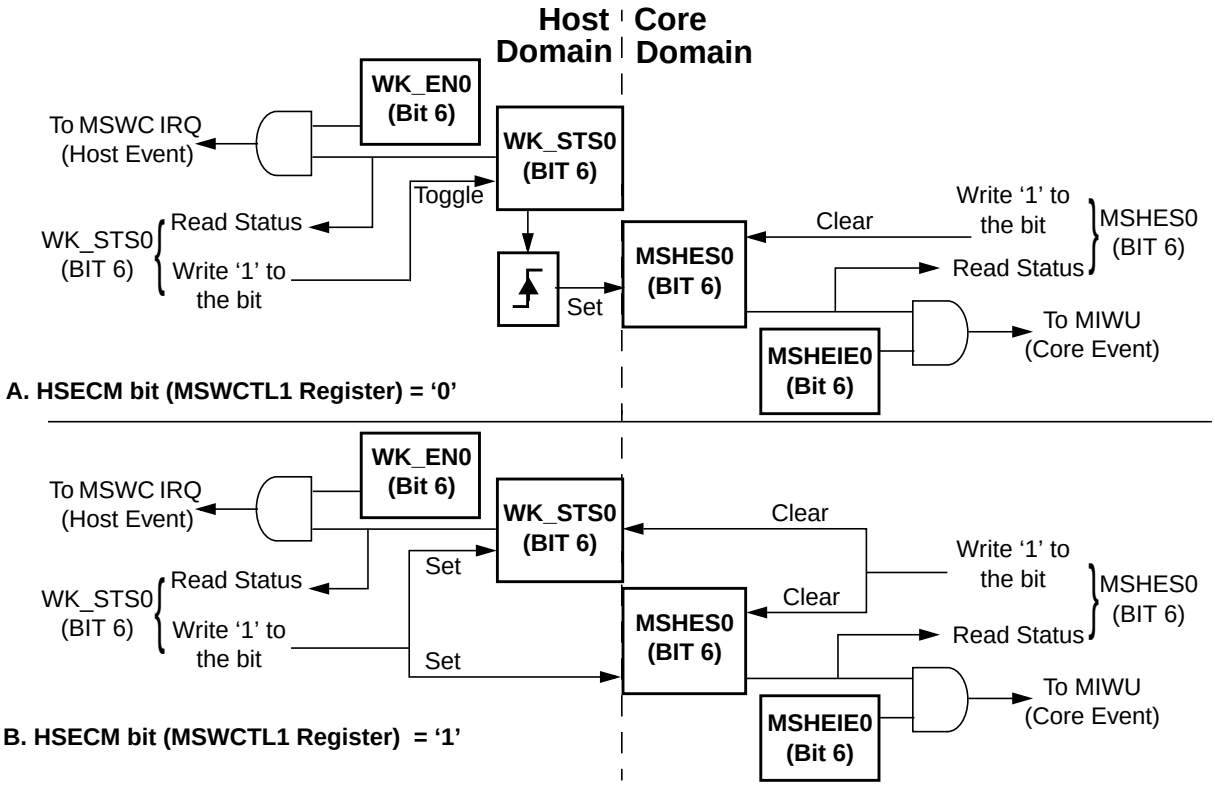



**Figure 96. MSWC Software Event Generation Scheme**

## 5.0 Host Controller Interface Module (Continued)

When HSECM bit is set, host Software Event Status bit in WK_STS0 register and the core Software Event Status bit in MSHES0 are both cleared by writing 1 to the core Software Event Status bit (MSHES0 register). This is useful when the software event is used to interrupt the core and is handled by it (Figure 96B).

### Module IRQ Wake-Up Event

A module IRQ wake-up event is defined as the leading edge of the IRQ assertion of the RTC.

To enable the IRQ of a specific logical device to trigger a wake-up event, the associated enable bit must be set to 1. This is bit 4 of the Interrupt Number and Wake-Up on IRQ Enable register, located at index $70_{16}$ in the configuration space of the logical device (see Table 42 on page 301). When this bit is set, any IRQ assertion of the corresponding logical device activates the module IRQ wake-up event. Therefore, the module IRQ wake-up event is a combination of all IRQ signals of the logical devices for which wake-up on IRQ is enabled.

When the event is detected as active, its associated status bit (bit 7 of WK0_STS register) is set to 1. If the associated enable bit (bit 7 of WK_EN0 register) is also set to 1, the $\overline{\text{PWUREQ}}$ output is asserted and remains asserted until the status bit is cleared.

Since $V_{DD}$ powers IRQ generation of the logical devices, a module IRQ event can be activated only when $V_{DD}$ is present (see Section 6.1 on page 297 for a list of logical devices).

### Modem Ring

High to low transitions on $\overline{\text{RI1}}$ (or $\overline{\text{RI2}}$) indicate the detection of a ring in an external modem and can be used as wake-up events.

### Telephone Ring

A telephone ring is detected by the MSWC by processing the raw signal coming directly from the telephone line into the $\overline{\text{RING}}$ input pin. Detection of a pulse train, with a frequency higher than 16 Hz lasting at least 0.19 sec, is used as a wake-up event.

The $\overline{\text{RING}}$ pulse-train detection is achieved by monitoring the falling edges on $\overline{\text{RING}}$ in time slots of 62.5 msec (a 16 Hz cycle). A positive detection occurs if falling edges of $\overline{\text{RING}}$ are detected in three consecutive time slots, following a time slot in which no $\overline{\text{RING}}$ falling edge is detected. This detection method guarantees the detection of a $\overline{\text{RING}}$ pulse train with frequencies higher than 16 Hz. It filters out (does not detect) pulses of less than 10 Hz and may detect pulses between 10 Hz and 16 Hz.

### ACPI State Change and Legacy Off Events

The host may operate in either Legacy or ACPI mode. The operation mode is specified by the Power Button Mode bit in SuperI/O Configuration D register (SIOCFD). When EICFGPBM bit in MSIEN2 register is set, a change to the Power Button Mode bit generates an interrupt to the core. The core may read the value of the Power Button Mode bit, using CFGPBM bit in MSWCTL2 register, to determine how to interpret the other power state request bits.

The Power Supply Off bit in SIOCFD register may be used in Legacy mode to indicate a request to turn power off. A write of 1 to this bit sets CFGPSO bit in MSWCTL2 register; then, if EICFGPSO bit in MSIEN2 register is set, an interrupt to the core is generated, indicating the event.

A set of System State Change Request bits (S1-S5) are provided in WK_STATE register. The host uses these bits for ACPI-compliant state change requests. A write of 1 to any of these bits indicates a state change request to the core through the respective bit in MSWCTL2 register. When all bits in WK_STATE are written with 0, a request of S0 is indicated, and ACPIS0 bit in MSWCTL2 register is set. When any S0-S5 bit in MSWCTL2 is set and the respective mask bit in MSIEN2 register is set, an interrupt to the core is generated whenever a change to any of the state bits is detected.

All interrupt requests may be cleared by writing 1 to the corresponding status bit or by masking the event (by clearing the corresponding Interrupt Enable bit).

### RTC Alarm

The RTC module may generate an ALARM signal (see Section 6.2.8 on page 321). The RTC alarm can serve as a wake-up request to wake up the system; the request is routed to the core, which then wakes up the system. To enable an alarm wake-up, the following settings should be made:

- Set the Alarm conditions in the RTC module. By masking the various interrupts, software may select a wake-up either to the host directly, using the RTC's IRQ, or to the core through the Alarm signal.
- Enable the Wake-Up on Alarm status interrupt masking (optional, for Interrupt Enabled mode) by setting EIRTCAL bit in MSIEN2 register.
- Verify that the RTCAL bit in MSWCTL3 bit is cleared (no pending Alarm request).
- Enable the Wake-Up on MSWC event in the MIWU and ICU modules.
- Verify that the ALARM bit in the RTC is cleared.

After an ALARM event is detected in the RTC, the RTC ALARM status bit is set (bit 5 in CRC register, page 331); in response, RTCAL bit in MSWCTL3 register is set.

## 5.0 Host Controller Interface Module (Continued)

When handling an ALARM event, make sure that no events are lost by clearing RTCAL bit before clearing the ALARM status in the RTC.

### 5.5.3 Wake-Up Output Events

The MSWC generates four types of output events:

- IRQ - an interrupt routed as configured in the MSWC PnP configuration registers.
- $\overline{\text{PWUREQ}}$ - an event that is typically connected to an input in the chipset that triggers an SCI event.
- $\overline{\text{SMI}}$ - an event typically connected to an input in the chipset that triggers an SMI event.
- MSWCI - an interrupt to the MIWU module in the core domain. This enables the core firmware to handle the wake-up events.

Figure 97 shows the enabling mechanism and the event generation scheme for the various output events. Output events to the host are generated for input events that have their status bit set (WK_STSn.i is 1). Output events to the core, through the MIWU, are generated for input events that have their core status bit set (MSHESn.i is 1).

Each of the three Host Wake-Up Event Routing Control registers (WK_ENn, WK_SMIENn and WK_IRQENn) holds a Routing Enable bit for each event; this allows selective routing of these events to $\overline{\text{PWUREQ}}$, $\overline{\text{SMI}}$ and/or the assigned MSWC interrupt request (IRQ) channel, respectively.

After an output event is asserted, it is active until all set status bits are cleared or masked. The current status of the event may be read at the ACPI status registers in the chipset's ACPI controller or by reading "Wake-Up Event Status Register 0 (WK_STS0)" on page 286 and "Wake-Up Signals Value Register (WK_SIGV)" on page 288.

As shown in Figure 98, for $\overline{\text{SMI}}$ output events, the MSWC combines the event request coming from the Host Interface's Power Management channels 1 and 2 with MSWC internal $\overline{\text{SMI}}$ events.

The $\overline{\text{SMI}}$ may be output from the PC87591L-N05 using the dedicated $\overline{\text{SMI}}$ signal or by routing $\overline{\text{SMI}}$ to an interrupt request channel via the device's configuration registers.

The Wake-Up Event Routing Control register, MSHEIEn, which is controlled by the core, holds an enable bit for each of the events, which allows selective routing of these events to the core wake-up interrupt (MSWCI) to the MIWU. The core event is controlled using a separate set of status signals to prevent race conditions when clearing events.

The MSWCI interrupt is a level high interrupt that gathers requests from MSHESn, MSWCTL2 and MSWCTL3 registers. Once an output event is asserted, it keeps its active state until all set status bits are cleared or masked. This interrupt signal is connected to the MSWC wake-up input of the MIWU. This enables handling state change requests even in Idle mode. The MSWC output for this input is connected to the core through the MIWU module, enabling a power state change on interrupt.

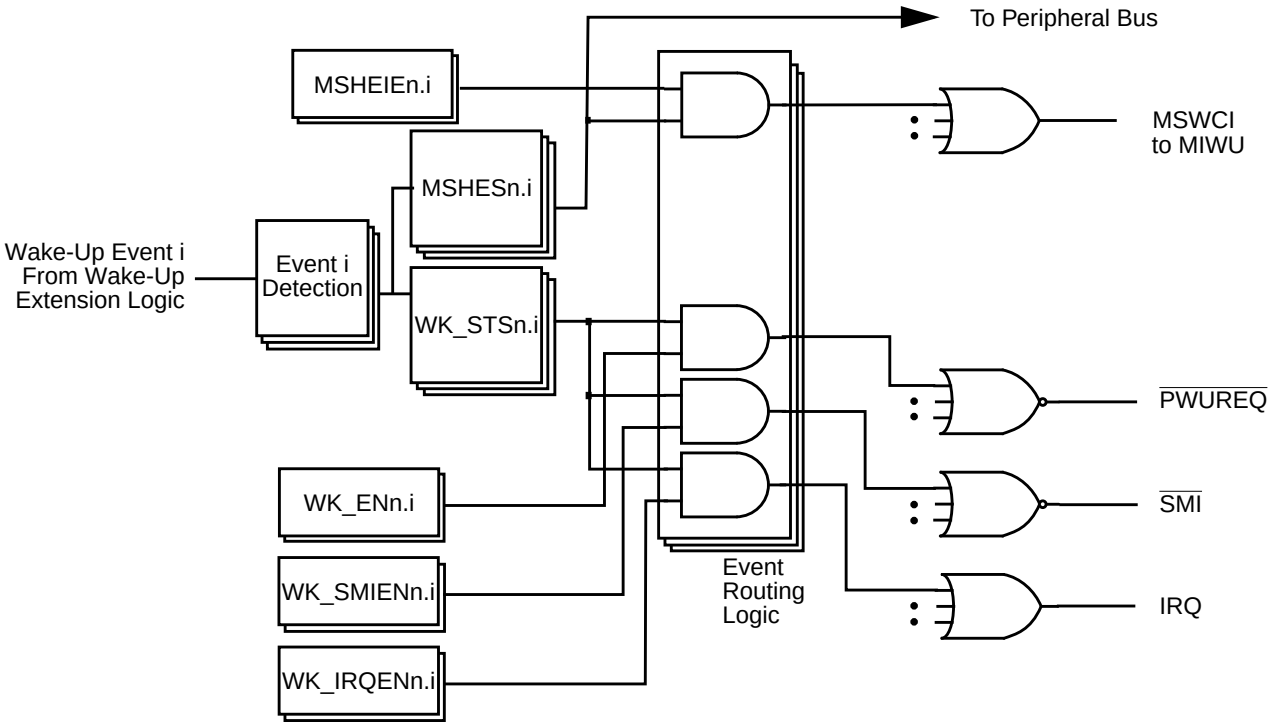



**Figure 97. Wake-Up Event Routing Scheme**

## 5.0 Host Controller Interface Module (Continued)

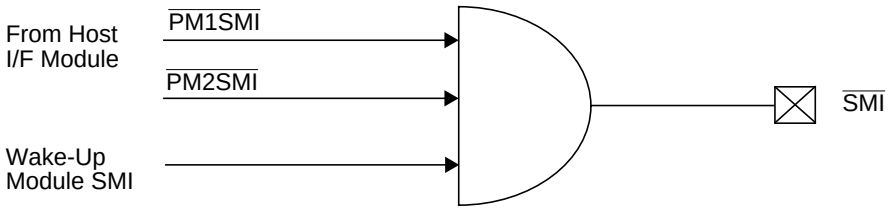



**Figure 98. $\overline{\text{SMI}}$ Source Gathering Scheme**

### 5.5.4 Other MSWC Controlled Elements

In addition to its Power Management functions, the MSWC controls the handling of the following system control elements:

- Host Configuration Address Selection
- Host Keyboard Reset Fast Reset Output ($\overline{\text{KBRST}}$)
- GA20 Pin Functionality
- Host Power on indication

#### Host Configuration Address Selection

The standard strap configuration enables the selection of one of two SuperI/O configuration register addresses. When the PC87591L-N05 is enabled in Programmable Configuration Address mode (see Table 37 on page 297), the core may set the address of the SuperI/O configuration index/data registers.

HCFGBAL and HCFGBAH are byte-wide read/write registers. HCFGBAL holds the least significant byte of a host motherboard PnP initial configuration address; HCFGBAH holds the most significant byte. The contents of HCFGBAH and HCFGBAL change only during $V_{CC}$ Power-Up reset.

To update the base address of the SuperI/O configuration index/data registers, do the following:

1. Clear VHCFGA bit in MSWCTL1 register by writing 1 to it.
2. Write the lower byte of the address to HCFGBAL (LSB must be written 0).
3. Write the higher byte of the address to HCFGBAH.
4. Set HCFGLK bit to prevent an accidental change of the address written to HCFGBAL and HCFGBAH.

The base address is preserved by $V_{CC}$, and VHCFGA is set as long as a valid address is maintained. If there is no valid configuration base address, the LPC interface does not respond to configuration requests.

#### Host Keyboard Fast Reset

The Host Keyboard Reset output ($\overline{\text{KBRST}}$) is an output of the PC87591L-N05 that serves as one of the sources for Host Soft reset commands (i.e., $\overline{\text{INIT}}$ input in the x86 processors). Figure 99 shows the $\overline{\text{KBRST}}$ generation scheme. The host is reset when the $\overline{\text{KBRST}}$ output is low. A reset command is issued by the PC87591L-N05 by software or hardware, as follows:

- Software: The core firmware can issue a reset command to the host by writing 1 to HRSTOB in MSWCTL1 register. The reset to the host ends by writing 0 to this bit.
- Hardware: The host is reset during $V_{CC}$ Power-Up reset if HRAPU bit in MSWCTL3 register is set and an LPC transaction is started. This is used to prevent accesses to the PC87591L-N05 from being ignored due to the duration of the Power-Up reset.

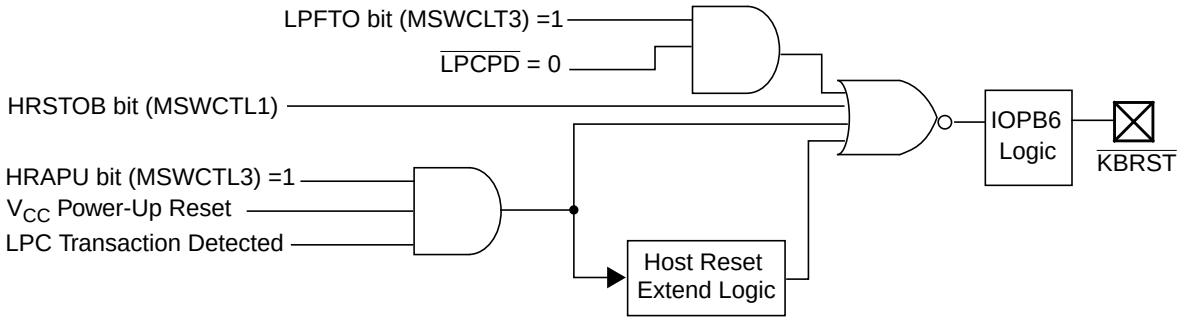



**Figure 99. $\overline{\text{KBRST}}$ Generation Scheme**

### GA20 Pin Functionality

The GA20 (Gate Address A20) function is part of the PC architecture. In PC87591L-N05, the GA20 function is implemented by a GPIO signal that is configured as output. Port PB5 is recommended to be used as GA20 since its default state after reset is output driving high. The firmware running on the core may change the GA20 signal state by modifying bit 5 in PBD-OUT register. There is no special hardware or multiplexing on PB5; since there is no multiplexing, bit 5 of PBALT register is always 0 and any writes to it are disregarded. PB5 may be used as a GPIO; however, note that wake-up for PB5 differs from the other signals in port B.

### 5.5.5 MSWC Host Registers

The MSWC registers are organized in four banks, all of which are battery-backed. The offsets are related to a base address that is determined by the MSWC Base Address register in the device configuration registers. The lower 19 offsets ($00_{16}$-$12_{16}$) are common to the four banks; the upper offsets ($13_{16}$-$1F_{16}$) are divided as follows:

- Bank 0 is reserved.
- Bank 1 is reserved.
- Bank 2 holds the Event Routing Configuration and Wake-Up Extension Control registers.
- Bank 3 is reserved.

The active bank is selected through the Configuration Bank Select field (bits 1-0) in the Wake-Up Configuration register (WK_CFG).

As a programing aid, the registers are described in this chapter according to the following functional groupings:

- General status
- Enable
- Configuration
- Routing

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

### MSWC Host Register Map

The following tables list the MSWC host registers. For the MSWC core register map, see Section 5.5.6 on page 291.

**Table 34. Banks 0, 1, 2 and 3 - The Common Control and Status Register Map**

| Offset | Mnemonic | Register Name | Type |
|---|---|---|---|
| $00_{16}$ | WK_STS0 | Wake-Up Event Status 0 | R/W1C |
| $02_{16}$ | WK_EN0 | Wake-Up Enable 0 | R/W |
| $04_{16}$ | WK_CFG | Wake-Up Configuration | R/W |
| $06_{16}$ | WK_SIGV | Wake-Up Signal Value | RO |
| $07_{16}$ | WK_STATE | Wake-Up ACPI State | WO |
| Other | Reserved | | |

**Table 35. Bank 2 - Event Routing Configuration Register Map**

| Offset | Mnemonic | Register Name | Type |
|---|---|---|---|
| $13_{16}$ | WK_SMIEN0 | Wake-Up SMI Enable 0 | R/W |
| $15_{16}$ | WK_IRQEN0 | Wake-Up Interrupt Request Enable 0 | R/W |
| Other | Reserved | | |

## 5.0 Host Controller Interface Module (Continued)

### Wake-Up Event Status Register 0 (WK_STS0)

This register is set to $00_{16}$ on $V_{PP}$ power-up, $V_{CC}$ power-up or Host Domain Software reset. It indicates which wake-up events, associated with the register, have occurred. Writing 1 to a bit clears it to 0. Writing 0 has no effect. Bit 6 behaves in a special way, as described in the table below.

Location: Offset $00_{16}$

Type: R/W1C

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Module IRQ Event Status** | **Software Event Status** | **Reserved** | | **$\overline{\text{RING}}$ Event Status** | **Reserved** | **$\overline{\text{RI2}}$ Event Status** | **$\overline{\text{RI1}}$ Event Status** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **$\overline{\text{RI1}}$ Event Status.**<br>0: Event not detected (default)<br>1: Event detected |
| 1 | **$\overline{\text{RI2}}$ Event Status.**<br>0: Event not detected (default)<br>1: Event detected |
| 2 | **Reserved.** |
| 3 | **$\overline{\text{RING}}$ Event Status.** $\overline{\text{RING}}$ event detection, according to the RING detection mode enabled.<br>0: Event not detected (default)<br>1: Event detected |
| 5-4 | **Reserved.** |
| 6 | **Software Event Status.** This bit may work in two modes, as defined by HSECM bit in MSWCTL1 register (see “MSWC Control Status Register 1 (MSWCTL1)” on page 291).<br>When HSECM is 0, writing 1 to Software Event Status bit inverts its value.<br>When HSECM is 1, writing 1 to Software Event Status bit sets it; the bit is cleared by a write of 1 to bit 6 in MSHES0 register.<br>0: Event not active (default)<br>1: Event active |
| 7 | **Module IRQ Event Status.** This sticky bit shows the status of the module IRQ event detection.<br>0: Event not active (default)<br>1: Event active |

### Wake-Up Events Enable Register (WK_EN0)

This register is set to $00_{16}$ on $V_{PP}$ power-up or Host Domain Software reset. When enabled wake-up events are detected, the $\overline{\text{PWUREQ}}$ signal is activated.

Location: Offset $02_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Module IRQ Event Enable** | **Software Event Enable** | **Reserved** | | **$\overline{\text{RING}}$ Event Enable** | **Reserved** | **$\overline{\text{RI2}}$ Event Enable** | **$\overline{\text{RI1}}$ Event Enable** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **$\overline{\text{RI1}}$ Event Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 1 | **$\overline{\text{RI2}}$ Event Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 2 | **Reserved.** |
| 3 | **$\overline{\text{RING}}$ Event Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 5-4 | **Reserved.** |
| 6 | **Software Event Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 7 | **Module IRQ Event Enable.**<br>0: Disabled (default)<br>1: Enabled |

### Wake-Up Configuration Register (WK_CFG)

This register is set to $00_{16}$ on $V_{PP}$ power-up or Host Domain Software reset. It enables access to Event Routing Control registers (bank selected).

Location: Offset $04_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | **Configuration Bank Select** | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Required | 0 | 0 | | | | | | |

| Bit | Description |
|---|---|
| 1-0 | **Configuration Bank Select.**<br>**Bits 1 0 : Bank — Register**<br>0 0 : 0 — Reserved (default)<br>0 1 : 1 — Reserved<br>1 0 : 2 — Event Routing, Wake-Up Extension<br>1 1 : 3 — Reserved |
| 7-2 | **Reserved.** |

# 5.0 Host Controller Interface Module (Continued)

## Wake-Up Signals Value Register (WK_SIGV)

This is a read-only register that returns the value of $\overline{SMI}$ and $\overline{PWUREQ}$ signal output and input to this module. This register helps to identify the source of the wake-up request when multiple sources are enabled.

Location: Offset $06_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **PWUREQ of Wake-Up Value** | **PWUREQ Output Value** | **PM2 SMI Output Value** | **PM1 SMI Output Value** | **SMI Wake-Up Output Value** | **SMI Output Value** |

| Bit | Description |
|---|---|
| 0 | **SMI Output Value.**<br>0: $\overline{SMI}$ output is low (asserted)<br>1: $\overline{SMI}$ output is high (de-asserted) |
| 1 | **SMI Wake-Up Output Value.**<br>0: $\overline{SMI}$ output of the wake-up module is low (asserted)<br>1: $\overline{SMI}$ output of the wake-up module is high (de-asserted) |
| 2 | **PM1 SMI Output Value.**<br>0: $\overline{SMI}$ output of the Power Management channel 1 is low (asserted)<br>1: $\overline{SMI}$ output of the Power Management channel 1 is high (de-asserted) |
| 3 | **PM2 SMI Output Value.**<br>0: $\overline{SMI}$ output of the Power Management channel 2 is low (asserted)<br>1: $\overline{SMI}$ output of the Power Management channel 2 is high (de-asserted) |
| 4 | **PWUREQ Wake-Up Output Value.**<br>0: $\overline{PWUREQ}$ output is low (asserted)<br>1: $\overline{PWUREQ}$ output is high (de-asserted) |
| 5 | **PWUREQ Wake-Up Value.**<br>0: $\overline{PWUREQ}$ output of the wake-up module is low (asserted)<br>1: $\overline{PWUREQ}$ output of the wake-up module is high (de-asserted) |
| 7-6 | **Reserved.** |

## Wake-Up ACPI State Register (WK_STATE)

This is a write-only register. It always returns $00_{16}$ when read.

Location: Offset $07_{16}$

Type: WO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **S5** | **S4** | **S3** | **S2** | **S1** | **Reserved** |

| Bit | Description |
|---|---|
| 0 | **Reserved.** |
| 1 | **S1 (Request to Change to S1 State).** A write of 1 to this bit indicates to the core that the host requests to change to S1 state. The S state and transition are interpreted, as specified in the ACPI standard for S state change requests. This bit always reads back 0.<br>0: Not an S1 state request<br>1: S1 state setting request |

## 5.0 Host Controller Interface Module (Continued)

| | |
|---|---|
| 2 | **S2 (Request to Change to S2 State).** A write of 1 to this bit indicates to the core that the host requests to change to S2 state. The S state and transition are interpreted, as specified in the ACPI standard for S state change requests This bit always reads back 0.<br>0: Not an S2 state request<br>1: S2 state setting request |
| 3 | **S3 (Request to Change to S3 State).** A write of 1 to this bit indicates to the core that the host requests to change to S3 state. The S state and transition are interpreted, as specified in the ACPI standard for S state change requests. This bit always reads back 0.<br>0: Not an S3 state request<br>1: S3 state setting request |
| 4 | **S4 (Request to Change to S4 State).** A write of 1 to this bit indicates to the core that the host requests to change to S4 state. The S state and transition are interpreted, as specified in the ACPI standard for S state change requests. This bit always reads back 0.<br>0: Not an S4 state request<br>1: S4 state setting request |
| 5 | **S5 (Request to Change to S5 State).** A write of 1 to this bit indicates to the core that the host requests to change to S5 state. The S state and transition are interpreted, as specified in the ACPI standard for S state change requests. This bit always reads back 0.<br>0: Not an S5 state request<br>1: S5 state setting request |
| 7-6 | **Reserved.** |

### Wake-Up Event Routing to $\overline{\text{SMI}}$ Enable Register 0 (WK_SMIEN0)

This register is set to $00_{16}$ on $V_{PP}$ power-up or Host Domain software reset. It controls the routing of detected wake-up events to the $\overline{\text{SMI}}$ signal. Detected wake-up events that are enabled activate the $\overline{\text{SMI}}$ signal regardless of the value of WK_EN0 register.

Location: Bank 2, Offset $13_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **Software Event to $\overline{\text{SMI}}$ Enable** | **Reserved** | | **$\overline{\text{RING}}$ Event to $\overline{\text{SMI}}$ Enable** | **Reserved** | **$\overline{\text{RI2}}$ Event to $\overline{\text{SMI}}$ Enable** | **$\overline{\text{RI1}}$ Event to $\overline{\text{SMI}}$ Enable** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **$\overline{\text{RI1}}$ Event to $\overline{\text{SMI}}$ Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 1 | **$\overline{\text{RI2}}$ Event to $\overline{\text{SMI}}$ Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 2 | **Reserved.** |
| 3 | **$\overline{\text{RING}}$ Event to $\overline{\text{SMI}}$ Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 5-4 | **Reserved.** |
| 6 | **Software Event to SMI Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 7 | **Reserved.** |

# 5.0 Host Controller Interface Module (Continued)

## Wake-Up Event Routing to IRQ Enable Register 0 (WK_IRQEN0)

This register is set to $00_{16}$ on $V_{PP}$ power-up or Host Domain Software reset. It controls the routing of detected wake-up events to the assigned MSWC interrupt request (IRQ) channel. Detected wake-up events that are enabled activate the assigned IRQ channel regardless of the value of WK_EN0 register.

Location: Bank 2, Offset $15_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **Software Event to IRQ Enable** | **Reserved** | | **$\overline{\text{RING}}$ Event to IRQ Enable** | **Reserved** | **$\overline{\text{RI2}}$ Event to IRQ Enable** | **$\overline{\text{RI1}}$ Event to IRQ Enable** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **$\overline{\text{RI1}}$ Event to IRQ Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 1 | **$\overline{\text{RI2}}$ Event to IRQ Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 2 | **Reserved.** |
| 3 | **$\overline{\text{RING}}$ Event to IRQ Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 5-4 | **Reserved.** |
| 6 | **Software Event to IRQ Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 7 | **Reserved.** |

# 5.0 Host Controller Interface Module (Continued)

## 5.5.6 MSWC Core Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

**Table 36. MSWC Core Register Map**

| Mnemonic | Register Name | Type |
|---|---|---|
| MSWCTL1 | MSWC Control Status Register 1 | Varies per bit |
| MSWCTL2 | MSWC Control Status Register 2 | Varies per bit |
| MSWCTL3 | MSWC Control Status Register 3 | R/W |
| HCFGBAL | Host Configuration Base Address Low | R/W |
| HCFGBAH | Host Configuration Base Address High | R/W |
| MSIEN2 | MSWC Interrupt Enable Register 2 | R/W |
| MSHES0 | MSWC Host Event Status Register 0 | R/W1C |
| MSHEIE0 | MSWC Host Event Interrupt Enable Register | R/W |

### MSWC Control Status Register 1 (MSWCTL1)

This is a byte-wide read/write register that controls the settings associated with host wake-up and activity. The contents of this register are preserved by $V_{CC}$. Bit 0 is cleared by Warm reset; other bits are reset only on $V_{CC}$ Power-Up reset.

Location: 00 $FCC0_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **HSECM** | **HCFGLK** | **VHCFGA** | **LPCRSTA** | **HPWRON** | **HRSTOB** |
| Reset | 0 | 0 | 0 | 0 | 0 | - | - | 0 |

| Bit | Type | Description |
|---|---|---|
| 0 | R/W | **HRSTOB (Host Reset Out Bit).** Enables the PC87591L-N05 to generate a Host Soft reset via firmware, using the $\overline{\text{KBRST}}$ pin. The pin is held low (reset is active) for as long this bit 1.<br>0: $\overline{\text{KBRST}}$ is not forced active (default)<br>1: Force $\overline{\text{KBRST}}$ active |
| 1 | RO | **HPWRON (Host Power On).** The $V_{DD}$ power detection logic indicates that $V_{DD}$ is on.<br>0: $V_{DD}$ is off (below $V_{DDON}$)<br>1: $V_{DD}$ is on (above $V_{DDON}$) |
| 2 | RO | **LPCRSTA (LPC Reset Active).** The $\overline{\text{RESET1}}$ input is active (low).<br>0: $\overline{\text{RESET1}}$ is not active (high)<br>1: $\overline{\text{RESET1}}$ is active (low) |
| 3 | R/W1C | **VHCFGA (Valid Host Configuration Address).** This bit is set by a write to HCFGBAH register, as detailed in the update sequence in “Host Configuration Address Selection” on page 284. The firmware can clear the bit by writing 1 to it. This register is used as the address of the Configuration registers when the PC87591L-N05 is set to operate with the internal base address. Writing 0 to this bit is ignored. This bit can be locked and made read only by setting HCFGLK (bit 4).<br>0: Host Configuration Registers base address is not valid and access to this registers by the host is not enabled (default)<br>1: Host Configuration Registers base address is specified in HCFGBAH and HCFGBAL registers |

## 5.0 Host Controller Interface Module (Continued)

| Bit | Type | Description |
|---|---|---|
| 4 | R/W1S | **HCFGLK (Host Configuration Address Lock).** This bit is cleared during $V_{CC}$ power-up, Watchdog reset or Debugger Interface reset, but is unchanged during other reset events.<br>When 1 is written to this bit, it becomes read only (i.e., it cannot be cleared by the firmware) and locks VHCFGA bit, HCFGBAH register and HCFGBAL register, preventing accidental alteration to them.<br>0: Allows update of the Host Configuration Registers base address (default)<br>1: Locks the Host Configuration Registers base address |
| 5 | R/W1C | **HSECM (Host Software Event Clear Mode).** Controls the clear mode of Host Software Event Status bit in WK_STS0. This bit is cleared at $V_{CC}$ power-up and $\overline{\text{RESET1}}$ events.<br>0: Host Software Event Status bit in WK_STS0 (bit 6) toggles on host writes of 1. MSHES0 bit 6 is set when WK_STS0 bit 6 changes from 0 to 1 (default).<br>1: Host Software Event Status bit in WK_STS0 (bit 6) and MSHES0 bit 6 are both cleared by writes of 1 to MSHES0 register |
| 7-6 | | **Reserved.** |

### MSWC Control Status Register 2 (MSWCTL2)

This is a byte-wide read/write register that controls the settings associated with host wake-up and activity. Bits in this register are cleared by $V_{CC}$ Power-Up and $\overline{\text{RESET1}}$ resets.

Location: 00 $FCC2_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **CFGPSO** | **CFGPBM** | **ACPIS5** | **ACPIS4** | **ACPIS3** | **ACPIS2** | **ACPIS1** | **ACPIS0** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Type | Description |
|---|---|---|
| 0 | R/W1C | **ACPIS0 (ACPI request for S0).** This bit may be used by ACPI software to directly request a change of power state. This bit is set when the host software writes a value of 0 to bits S1 through S5 in WK_STATE register. This bit is cleared by writing 1 to it. A write of 0 is ignored.<br>When ACPIS0 is set, an MSWC wake-up interrupt to the core, is asserted (via a MIWU input).<br>0: No pending request for S0 change (default)<br>1: A request for S0 change was detected |
| 5-1 | R/W1C | **ACPIS1-5 (ACPI request for S1 through S5).** These bits may be used by ACPI software to directly request a change of power state. These bits are set by a host software write of 1 to the respective bit in WK_STATE register. The bit is cleared by writing 1 to it. A write of 0 is ignored.<br>When any ACPIS1-5 bit is set, an MSWC wake-up interrupt to the core is asserted (via a MIWU input). |
| 6 | RO | **CFGPBM (SuperI/O Configuration Register D Power Button Mode).** This bit reflects the current status of the Power Button Mode bit in SIOCFD register. This bit may be used by the host software to specify to the core the method used for power off signaling. See “SuperI/O Configuration D Register (SIOCFD)” on page 309<br>A write of 1 clears the interrupt signal caused by a change in this bit value. A write of 0 to this bit is ignored. |
| 7 | R/W1C | **CFGPSO (SuperI/O Configuration Register D Power Supply Off).** This bit is set whenever a 1 is written to the Power Supply Off bit in SIOCFD register. This bit may be used by the host software to specify to the core that the power supply should be turned off in a non-ACPI system. See “SuperI/O Configuration D Register (SIOCFD)” on page 309.<br>A write of 1 clears this bit and the interrupt signal generated when this bit is set. A write of 0 to this bit is ignored. |

## 5.0 Host Controller Interface Module (Continued)

### MSWC Control Status Register 3 (MSWCTL3)

This is a byte-wide read/write register that controls the settings associated with host wake-up and activity. The contents of this register is preserved by $V_{PP}$ and it is reset only on $V_{PP}$ Power-Up reset.

Location: 00 $FCC4_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | **RTCAL** | **LPFTO** | **HRAPU** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Description |
|---|---|
| 0 | **HRAPU (Host Reset when Accessed During $V_{CC}$ Power-Up Reset).** Indicates that a reset should be sent to the host if LPC activity was detected while the $V_{CC}$ Power-Up reset was not completed. This intends to re-start any LPC transaction that may have addressed the PC87591L-N05 but could not be handled correctly. When HCFGLK bit is set, writes to this bits are ignored.<br>0: Do not generate a reset on LPC transactions while the PC87591L-N05 is in Power-Up reset<br>1: Assert $\overline{KBRST}$ output on LPC transactions while the PC87591L-N05 is executing the $V_{CC}$ Power-Up reset sequence (default) |
| 1 | **LPFTO (LPC Power Fail Turn Off $\overline{KBRST}$ and GA20).** Indicates the handling of $\overline{KBRST}$ and GA20 outputs when $\overline{LPCPD}$ is active.<br>0: Ignore $\overline{LPCPD}$ in handling these signals (default)<br>1: Force the two signals low while $\overline{LPCPD}$ is active or $V_{DD}$ is low |
| 2 | **RTCAL (RTC Alarm).** Indicates that an RTC Alarm event occurred. This bit is set on the rising edge of the RTC Alarm output. It is cleared by writing 1 to it. Note that the ALARM event detection is edge triggered by the RTCAL bit; thus for a new event to be detected, first RTCAL bit and then Alarm Status bit in the RTC must be cleared.<br>0: No RTC Alarm is flagged (default)<br>1: RTC Alarm rising edge was detected |
| 7-3 | **Reserved.** |

### Host Configuration Base Address Low (HCFGBAL)

This is a byte-wide read/write register that holds the lower byte of the Host Configuration Registers base address. Bit 0 of this register is always forced to 0 to guarantee address alignment. This register is cleared on $V_{CC}$ Power-Up reset.

Location: 00 $FCC8_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Host Configuration Registers Base Address Low** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### Host Configuration Base Address High (HCFGBAH)

This is a byte-wide read/write register that holds the higher byte of the Host Configuration Registers base address. This register is cleared on $V_{CC}$ Power-Up reset.

Location: 00 $FCCA_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Host Configuration Registers Base Address High** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 5.0 Host Controller Interface Module (Continued)

### MSWC Interrupt Enable Register 2 (MSIEN2)

This is a byte-wide read/write register that holds enable bits for interrupt generation to the core through the MIWU (level high) for the respective bits in MSWCTL2 and MSWCTL3 registers. The interrupt may be cleared by clearing the status bit or masking the interrupt. On Warm reset, this register is cleared ($00_{16}$).

Location: 00 $FCCC_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **EICFGPSO** | **EICFGPBM** | **EIACPIS5** | **EIACPIS4** | **EIACPIS3** | **EIACPIS2** | **EIACPIS1** | **EIRTCAL** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **EIRTCAL (Enable Interrupt on RTC Alarm).** Mask generation of interrupt to the core on setting of the RTC Alarm bit in MSWCTL1 register.<br>0: Interrupt disabled (default)<br>1: Generate a level high interrupt when the RTC Alarm bit is set |
| 5-1 | **EIACPIS5-1 (Enable Interrupt ACPI request for S5 through S1).** Mask generation of interrupt to the core on changes to ACPISi (i=5-1) bit in MSWCTL2 register. An interrupt enable for ACPIS0 is enabled when any of these bits is set.<br>0: Interrupt disabled (default)<br>1: Generate a level high interrupt on any change to the ACPISi bit |
| 6 | **EICFGPBM (Enable Interrupt SuperI/O Configuration Register D Power Button Mode).** Mask generation of interrupt to the core on changes to CFGPBM bit in MSWCTL2 register.<br>0: Interrupt disabled (default)<br>1: Generate a level high interrupt on any change to the CFGPBM bit |
| 7 | **EICFGPSO (Enable Interrupt SuperI/O Configuration Register D Power Supply Off).** Mask generation of interrupt to the core on set CFGPSO bit in MSWCTL2 register.<br>0: Interrupt disabled (default)<br>1: Generate a level high interrupt when the CFGPSO bit is set to 1 |

### MSWC Host Event Status Register 0 (MSHES0)

This register holds information similar to that in WK_STS0 register. The same event that causes a WK_STS0 bit to be set sets the respective bit in MSHES0 register. Clearing bits is done for each of the status registers separately. This register is reset to $00_{16}$ on $V_{CC}$ power-up or Host Domain Software reset. Writing 1 to a bit clears it to 0. Writing 0 has no effect. Bit 6 of this register behaves in a special way on set and clear, as described below.

Location: Offset 00 $FCCE_{16}$

Type: R/W1C

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Module IRQ Event Status** | **Software Event Status** | **Reserved** | | **$\overline{\text{RING}}$ Event Status** | **Reserved** | **$\overline{\text{RI2}}$ Event Status** | **$\overline{\text{RI1}}$ Event Status** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **$\overline{\text{RI1}}$ Event Status.**<br>0: Event not detected (default)<br>1: Event detected |
| 1 | **$\overline{\text{RI2}}$ Event Status.**<br>0: Event not detected (default)<br>1: Event detected |
| 2 | **Reserved.** |

## 5.0 Host Controller Interface Module (Continued)

| Bit | Description |
|---|---|
| 3 | **$\overline{\text{RING}}$ Event Status.** $\overline{\text{RING}}$ event detection, according to the RING detection mode enabled.<br>0: Event not detected (default)<br>1: Event detected |
| 5-4 | **Reserved.** |
| 6 | **Software Event Status.** This bit indicates a host software event. It may operate in two modes depending on HSECM bit in MSWCTL1 register.<br>When HSECM is cleared, this bit is set when bit 6 of WK_STS0 changes from 0 to 1.<br>When HSECM is set, this bit is set on a write of 1 to WK_STS0 bit 1. This bit is cleared by writing 1 to it.<br>0: Event not active (default)<br>1: Event active |
| 7 | **Module IRQ Event Status.** This sticky bit shows the status of the module IRQ event detection.<br>0: Event not active (default)<br>1: Event active |

### MSWC Host Event Interrupt Enable Register (MSHEIE0)

This register is cleared to $00_{16}$ on Warm reset. It enables a core interrupt through the MIWU (level high) for the respective bit in the MSHES0 register. The interrupt may be cleared by clearing the status bit or masking the interrupt.

Location: Offset 00 $FCD0_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Module IRQ Event Enable** | **Software Event Enable** | **Reserved** | | **RING Event Enable** | **Reserved** | **$\overline{\text{RI2}}$ Event Enable** | **$\overline{\text{RI1}}$ Event Enable** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **$\overline{\text{RI1}}$ Event Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 1 | **$\overline{\text{RI2}}$ Event Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 2 | **Reserved.** |
| 3 | **RING Event Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 5-4 | **Reserved.** |
| 6 | **Software Event Enable.**<br>0: Disabled (default)<br>1: Enabled |
| 7 | **Module IRQ Event Enable.**<br>0: Disabled (default)<br>1: Enabled |

# 5.0 Host Controller Interface Module (Continued)

## 5.5.7 Usage Hints

### $\overline{\text{PWUREQ}}$ Output Connection

The $\overline{\text{PWUREQ}}$ concentrates a set of wake-up and other power management events in the PC87591L-N05. In typical use, this signal is connected to one of the chipset inputs that drive SCI event to the host.

### $\overline{\text{RESET2}}$ Events

When $\overline{\text{RESET2}}$ is used to reset the host domain, some of the MSWC functions may need to be set to their default values when a falling edge of the $\overline{\text{RESET2}}$ is detected. An interrupt routine triggered by this event may be used for this task. The following functions should be reset:

- GA20
- $\overline{\text{KBRST}}$
- Host Registers: WK_EN0, WK_SMIEN0 and WK_IRWEN0;
  Use the “Core Access to Host-Controlled Modules” for this operation (see Section 5.4 on page 275).

# 6.0 Host-Controlled Modules and Host Interface

## 6.1 DEVICE ARCHITECTURE AND CONFIGURATION

The PC87591L-N05 Host-Controlled Functions comprises a collection of generic and proprietary functional blocks. Each functional block is described in a separate section in this document. However, some parameters in the implementation of the functional blocks may vary per function and/or device. This chapter describes the PC87591L-N05 structure and provides all logical device-specific information, including special implementation of generic blocks, system interface and device configuration.

The PC87591L-N05 Host-Controlled Functions consist of seven logical devices (involving six modules), the host interface and a central set of configuration registers, all built around a central internal bus. The internal bus is similar to an 8-bit ISA bus protocol. Figure 100 shows the blocks and their interconnection.

The LPC Bus Interface serves as a bridge between the external LPC interface and the internal bus. It supports the following operations, as defined in Intel's *LPC Interface Specification, Revision 1.0*:

- 8-bit I/O read
- 8-bit I/O write
- 8-bit Memory read
- 8-bit Memory write
- 8-bit FWH read
- 8-bit FWH write

The Configuration and Control register set supports ACPI-compliant PnP configuration. The configuration registers are structured as a subset of the Plug and Play Standard registers, defined in Appendix A of the *Plug and Play ISA Specification, Revision 1.0a* by Intel and Microsoft, and are similar to those used in National SuperI/O devices. All system resources assigned to the functional blocks (I/O address space, and IRQ lines) are configured in and managed by this register set. In addition, some function-specific parameters are configurable through the configuration registers and distributed to the functional blocks through special control signals.

### 6.1.1 Configuration Structure and Access

The configuration structure comprises a set of banked registers that are accessed via a pair of specialized registers.

#### The Index-Data Register Pair

Access to the Host-Controlled Functions configuration registers is via an Index-Data register pair, using two system I/O byte locations. The base address of this register pair is determined during $V_{CC}$ Power-Up reset, according to the state of the hardware strapping option on the BADDR1-0 pins. Table 37 shows the selected base addresses as a function of BADDR1-0 (see Section 2.2.11 on page 45).

**Table 37. BADDR1-0 Strapping Options**

| BADDR1-0 | I/O Address | |
|---|---|---|
| | Index Register | Data Register |
| 0 0 | $2E_{16}$ | $2F_{16}$ |
| 0 1 | $4E_{16}$ | $4F_{16}$ |
| 1 0[1] | (HCFGBAH,HCFGBAL) | (HCFGBAH,HCFGBAL)+1 |
| 1 1 | XOR-Tree Test Mode | |

1. See "Host Configuration Address Selection" on page 284 for more details about this option.

The Index register is an 8-bit read/write register located at the selected base address (Base+0). It is used as a pointer to the configuration register file and holds the index of the configuration register that is currently accessible via the Data register. Reading the Index register returns the last value written to it (or a default of $00_{16}$ after Host Domain reset).

The Data register is an 8-bit register located at the selected base address (Base+1) used as a data path to any configuration register. Accessing the Data register actually accesses the configuration register that the Index register is currently pointed to.

## 6.0 Host-Controlled Modules and Host Interface (Continued)

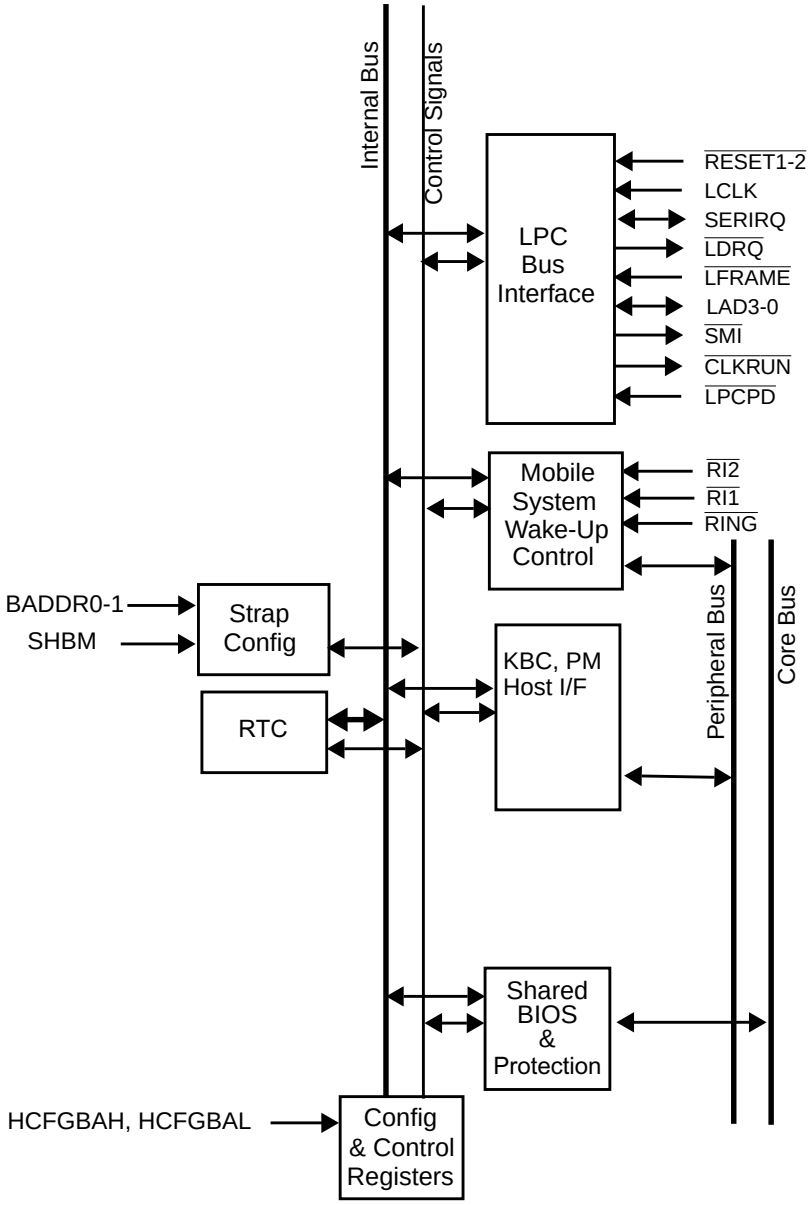



**Figure 100. Host-Controlled Domain Detailed Block Diagram**

### Banked Logical Device Registers Structure

Each functional block is associated with a Logical Device Number (LDN). The configuration registers are grouped into banks, where each bank holds the standard configuration registers of the corresponding logical device. Table 38 shows the LDN values of the PC87591L-N05 functional blocks. Any value not listed is reserved.

Figure 101 shows the structure of the standard configuration register file. The Host-Controlled Functions control and configuration registers are not banked and are accessed by the Index-Data register pair only, as described above. However, the device control and device configuration registers are replicated over seven banks for the seven logical devices. Therefore, two-dimensional indexing is used to access a specific register in a specific bank: the LDN register selects the bank (or logical device) and the Index register selects the register within the bank. Accessing the Data register while the Index register holds a value of $30_{16}$ or higher physically accesses the logical device configuration registers currently pointed to by the Index register, within the logical device currently selected by the LDN register.

## 6.0 Host-Controlled Modules and Host Interface (Continued)

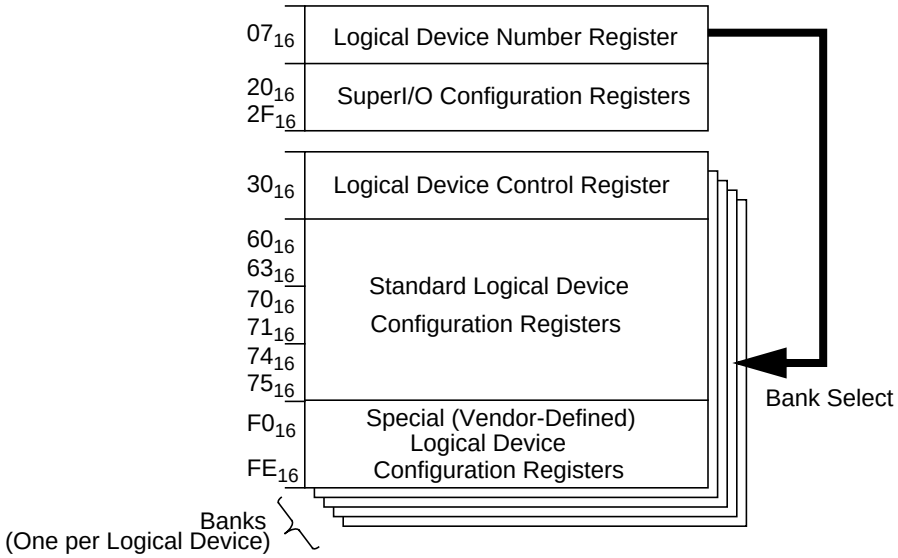



**Figure 101. Structure of Standard Configuration Register File**

**Table 38. Logical Device Number (LDN) Assignments**

| LDN | Functional Block |
|---|---|
| $04_{16}$ | Mobile System Wake-Up Control (MSWC) |
| $05_{16}$ | Keyboard and Mouse Controller (KBC) - Mouse Interface |
| $06_{16}$ | Keyboard and Mouse Controller (KBC) - Keyboard Interface |
| $0F_{16}$ | Shared BIOS and Protection |
| $10_{16}$ | Real Time Clock |
| $11_{16}$ | Power Management I/F Channel 1 |
| $12_{16}$ | Power Management I/F Channel 2 |

Write accesses to unimplemented registers (i.e., accessing the Data register while the Index register points to a non-existing register) are ignored; reads return $00_{16}$ for all addresses except $74_{16}$ and $75_{16}$ (DMA configuration registers), which return $04_{16}$ (indicating that no DMA channel is active). The configuration registers are accessible immediately after Host Domain reset.

### Standard Logical Device Configuration Register Definitions

In the registers below, any undefined bit is reserved. Unless otherwise noted, the following definitions also hold true:

- All registers are read/write.
- All reserved bits return 0 on reads except where noted. To prevent unpredictable results, do not modify these bits. Use read-modify-write to prevent the values of reserved bits from being changed during write.
- Write-only registers should not use read-modify-write during updates.

## 6.0 Host-Controlled Modules and Host Interface (Continued)

**Table 39. Standard Control Registers**

| Index | Register Name | Description |
|---|---|---|
| $07_{16}$ | Logical Device Number | This register selects the current logical device. See Table 38 for valid numbers. All other values are reserved. |
| $20_{16}$ - $2F_{16}$ | SuperI/O Configuration | SuperI/O configuration registers and ID registers |

**Table 40. Logical Device Activate Register**

| Index | Register Name | Description |
|---|---|---|
| $30_{16}$ | Activate | Bits 7-1: Reserved<br>Bit 0: Logical device activation control<br>0: Disabled<br>1: Enabled |

**Table 41. I/O Space Configuration Registers**

| Index | Register Name | Description |
|---|---|---|
| $60_{16}$ | I/O Port Base Address Bits (15-8) Descriptor 0 | Indicates selected I/O lower limit address bits 15-8 for I/O Descriptor 0. |
| $61_{16}$ | I/O Port Base Address Bits (7-0) Descriptor 0 | Indicates selected I/O lower limit address bits 7-0 for I/O Descriptor 0. |
| $62_{16}$ | I/O Port Base Address Bits (15-8) Descriptor 1 | Indicates selected I/O group 2 lower limit address bits 15-8 for I/O Descriptor 1. |
| $63_{16}$ | I/O Port Base Address Bits (7-0) Descriptor 1 | Indicates selected I/O group 2 lower limit address bits 7-0 for I/O Descriptor 1. |

## 6.0 Host-Controlled Modules and Host Interface (Continued)

**Table 42. Interrupt Configuration Registers**

| Index | Register Name | Description |
|---|---|---|
| $70_{16}$ | Interrupt Number and Wake-Up on IRQ Enable | Indicates selected interrupt number.<br>Bits 7-5: Reserved.<br>Bit 4: Enables wake-up on the IRQ of the logical device. When enabled, IRQ assertion triggers a wake-up event.<br>0: Disabled (default)<br>1: Enabled<br>Bits 3-0: select the interrupt number. A value of 1 selects IRQ1, a value of 2 selects IRQ2, etc. (up to IRQ15). A value of 0 disables this interrupt. |
| $71_{16}$ | Interrupt Request Type Select | Indicates the type and level of the interrupt request number selected in the previous register.<br>Bits 7-2: Reserved.<br>Bit 1: Polarity of interrupt request selected in previous register<br>0: Low polarity<br>1: High polarity<br>Bit 0: Type of interrupt request selected in previous register<br>0: Edge<br>1: Level |

**Table 43. DMA Configuration Registers**

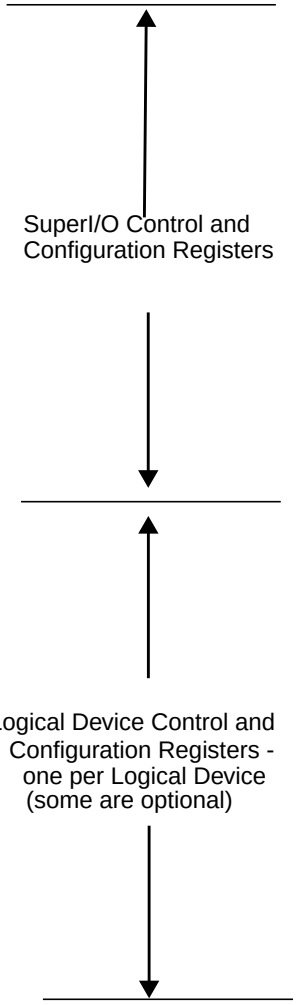
| Index | Register Name | Description |
|---|---|---|
| $74_{16}$ | DMA Channel Select 0 | Indicates selected DMA channel for DMA 0 of the logical device (0: The first DMA channel if more than one DMA channel is used).<br>Bits 7-3: Reserved.<br>Bits 2-0: Select the DMA channel for DMA 0. The valid choices are 0-3, where:<br>• A value of 0 selects DMA channel 0, 1 selects channel 1, etc.<br>• A value of 4 indicates that no DMA channel is active.<br>• The values 5-7 are reserved. |
| $75_{16}$ | DMA Channel Select 1 | Indicates selected DMA channel for DMA 1 of the logical device (1: The second DMA channel if more than one DMA channel is used).<br>Bits 7-3: Reserved.<br>Bits 2-0: Select the DMA channel for DMA 1. The valid choices are 0-3, where:<br>• A value of 0 selects DMA channel 0, 1 selects channel 1, etc.<br>• A value of 4 indicates that no DMA channel is active.<br>• The values 5-7 are reserved. |

**Table 44. Special Logical Device Configuration Registers**

| Index | Register Name | Description |
|---|---|---|
| $F0_{16}$-$FE_{16}$ | Logical Device Configuration | Special (vendor-defined) configuration options |

# 6.0 Host-Controlled Modules and Host Interface (Continued)

## 6.1.2 Standard Configuration Registers




| Index | Register Name |
|---|---|
| $07_{16}$ | Logical Device Number |
| $20_{16}$ | SuperI/O ID |
| $21_{16}$ | SuperI/O Configuration 1 |
| $25_{16}$ | SuperI/O Configuration 5 |
| $26_{16}$ | SuperI/O Configuration 6 |
| $27_{16}$ | SuperI/O Revision ID |
| $28_{16}$ | SuperI/O Configuration 8 |
| $29_{16}$ | SuperI/O Configuration 9 |
| $2D_{16}$ | SuperI/O Configuration D |
| $2B_{16}$ - $2E_{16}$ | Reserved exclusively for National use |
| $30_{16}$ | Logical Device Control (Activate) |
| $60_{16}$ | I/O Base Address Descriptor 0 Bits 15-8 |
| $61_{16}$ | I/O Base Address Descriptor 0 Bits 7-0 |
| $62_{16}$ | I/O Base Address Descriptor 1 Bits 15-8 |
| $63_{16}$ | I/O Base Address Descriptor 1 Bits 7-0 |
| $70_{16}$ | Interrupt Number and Wake-Up on IRQ Enable |
| $71_{16}$ | IRQ Type Select |
| $74_{16}$ | DMA Channel Select 0 |
| $75_{16}$ | DMA Channel Select 1 |
| $F0_{16}$ - $F9_{16}$ | Device Specific Logical Device Configuration 1 to 10 |

**Figure 102. Configuration Register Map**

### SuperI/O Control and Configuration Registers

The SuperI/O configuration registers at indexes $20_{16}$ and $27_{16}$ are mainly used for part identification, global power management and the selection of pin multiplexing options. For details, see Section 6.1.8 on page 306.

### Logical Device Control and Configuration Registers

A subset of these registers is implemented for each logical device. See the functional block descriptions in Chapter 5 on page 242 and Chapter 6 on page 297.

### Control

The only implemented control register for each logical device is the Activate register at index $30_{16}$. Bit 0 of Activate register controls the activation of the associated functional block. Activation enables access to the functional block's registers and attaches the functional block's system resources (e.g., address space and interrupts), which are unused as long as the block is not activated. Other effects may apply on a function-specific basis (such as clock enable and active pinout signaling).

## 6.0 Host-Controlled Modules and Host Interface (Continued)

### Standard Configuration

The standard configuration registers manage the PnP resource allocation to the functional blocks. The I/O port base address (descriptor 0) is a pair of registers at index $60\text{-}61_{16}$, holding the first 16-bit base address for the register set of the functional block. An optional 16-bit second base-address (descriptor 1) at index $62\text{-}63_{16}$ is used for logical devices with more than one continuous register set. Interrupt Number and Wake-Up on IRQ Enable (index $70_{16}$) and IRQ Type Select (index $71_{16}$) registers allocate an IRQ number to the module's interrupt and control the interrupt type and polarity. DMA Channel Select 0 (index $74_{16}$) allocates a DMA channel to the block, where applicable. DMA Channel Select 1 (index $75_{16}$) allocates a second DMA channel, where applicable.

### Special Configuration

The vendor-defined registers, starting at index $F0_{16}$ - $F9_{16}$, control function-specific parameters such as operation modes, power saving modes, clock rate selection and non-standard extensions to generic functions.

### 6.1.3 Default Configuration Setup

The default configuration setup of the PC87591L-N05 Host-Controlled functions is set according to the following reset types (see Section 3.2 on page 61):

- $V_{PP}$ Power-Up Reset
  Resets $V_{PP}$-retained SuperI/O functions, such as the RTC and the MSWC registers, whose values are retained by $V_{PP}$.
- $V_{CC}$ Power-Up Reset
  Resets the MSWC registers whose values are retained by $V_{CC}$ only.
- Host Domain Hardware Reset
  - Resets all SuperI/O logical devices, with the exception of the Mobile System Wake-Up Control (MSWC) and the RTC registers retained by $V_{PP}$ or $V_{CC}$.
  - Resets all SuperI/O configuration registers.
- Host Domain Software Reset
  - Resets all SuperI/O logical devices, except the MSWC and the RTC registers retained by $V_{PP}$ or $V_{CC}$.
  - Resets most bits in the SuperI/O configuration registers. This reset does not affect register bits that are locked for write access (see "SuperI/O Configuration 6 Register (SIOCF6)" on page 308 and Table 37 on page 297).

If a Host Domain Hardware reset occurs, the PC87591L-N05 wakes up with the following default SuperI/O configuration setup:

- The configuration base address is according to the BADDR strap pin value, as shown in Table 37 on page 297.
- All logical devices are disabled, with the exception of the MSWC and shared memory, which remain functional but whose registers cannot be accessed.

If a Host Domain reset occurs (either Software or Hardware), the PC87591L-N05 wakes up with the following default SuperI/O configuration setup:

- The legacy devices are assigned with their legacy system resource allocation.
- The National proprietary functions are not assigned with any default resources, and the default values of their base addresses are all $00_{16}$.

### 6.1.4 Address Decoding

A full 16-bit address decoding is applied when accessing the configuration I/O space as well as the registers of the functional blocks. However, the number of configurable bits in the base address registers varies for each logical device.

The lower 0, 1, 2, 3, 4 or 5 address bits are decoded within the functional block to determine the offset of the accessed register within the logical device's I/O range of 1, 2, 4, 8, 16 or 32 bytes, respectively. The rest of the bits are matched with the base address register to decode the entire I/O range allocated to the logical device. Therefore, the lower bits of the base address register are forced to 0 (read only), and the base address is forced to be 1, 2, 4, 8, 16 or 32 byte-aligned, according to the size of the I/O range.

The base address of the KBC, PM channel 1 and PM channel 2 are limited to the I/O address range of $0000_{16}$ to $07FX_{16}$ only (bits 11-15 are forced to 0). The addresses of other devices are configurable within the full 16-bit address range (up to $FFFF_{16}$).

In some special cases, other address bits are used for internal decoding (such as bit 2 in the KBC). The KBC has two I/O descriptors with some implied dependency between them. For more details, see the description of the base address register for each logical device.

The Shared Memory and Protection module serves as a bridge from the LPC to the on-chip ROM and off-chip expansion memory. For module control and protection function registers, the 16-bit base address is applied through the configuration address space. To access the registers, the lower four address bits are decoded within the Shared Memory module. The address ranges in the LPC memory space and the FWH memory space, which are bridged to the shared memory, are de-

## 6.0 Host-Controlled Modules and Host Interface (Continued)

fined in the SuperI/O configuration section for the shared memory bridge. The number of address bits used for this decoding varies according to the specified zones and their sizes. See “Memory Range Programing” on page 311 and “Shared Memory Configuration Register” on page 312 for details about the address range specifications.

### 6.1.5 Interrupt Serializer

The Interrupt Serializer translates internal IRQ sources into serial interrupt request data transmitted over the SERIRQ bus. Figure 103 shows the interrupt serialization mechanism.


**Figure 103. Interrupt Serialization Mechanism**

The internal IRQ signals are fed into an IRQ Mapping and Polarity Control block. This block maps them to their associated IRQ slots. The IRQs are then fed into the Interrupt Serializer, where they are translated into serial data and transmitted over the SERIRQ bus.

### 6.1.6 Protection

The PC87591L-N05 provides features to protect the Personal Computer (PC) at software levels. The PC can be locked to protect configuration bits and to prevent alteration of the device hardware configuration and several types of configuration settings.

The use of all protection mechanisms is optional.

### 6.1.7 LPC Interface

#### LPC Transactions Supported

The PC87591L-N05 LPC interface responds to the following LPC transactions as part of the standard Host Bus interface:

- I/O read cycles
- I/O write cycles

In addition, the Shared Memory module uses the following transactions:

- 8-bit memory read and write
- 8-bit FWH read and write

LPC transactions conform with Intel’s *LPC Interface Specification, Revision 1.0.*

The LPC- FWH read and write protocols are similar to memory read and write cycles. The specifications of these cycles are listed below. The Address, Data, TAR and SYNC cycles are as specified for LPC memory read and write cycles. The START and ID fields are similar to the equivalent cycle in LPC memory read and write transactions but differ in the data placed on the LAD signals (see details in the cycle description).

**Note:** The PC87591L-N05 supports FWH transactions from LPC controllers that accept wait-sync and long wait-sync cycles. With other LPC controllers, use the indirect write mechanism in the Shared Memory module to perform write operations.

**FWH Read Cycle**

1. START: $1101_{16}$ (0xD).
2. ID field: FWH ID nibble (compared with bits 7-4 of shared memory; see “Shared Memory Configuration Register” on page 312).
3. Address: Eight address nibbles, MS nibble first; see usage below).
4. TAR (two cycles).
5. SYNC.
6. DATA: Two data nibbles, LS nibble first (D3-D0, D7-D4).
7. TAR (two cycles).

## 6.0 Host-Controlled Modules and Host Interface (Continued)

**FWH Write Cycle:**

1. START: $1110_{16}$ (0xE).
2. ID field: FWH ID nibble (compared with bits 7-4 of shared memory; see “Shared Memory Configuration Register” on page 312).
3. Address: Eight address nibbles MS nibble first (see usage below).
4. DATA: Two data nibbles, LS nibble first (D3-D0, D7-D4).
5. TAR (two cycles).
6. SYNC.
7. TAR (two cycles).

The ID field is compared with bits 7-4 of shared memory; see “Shared Memory Configuration Register” on page 312. If the two match, the PC87591L-N05 continues handling the transaction; if they do not match, the current LPC-FWH transaction is ignored.

LPC-FWH Address translation: The address field in the LPC-FWH transaction is constructed of eight nibbles. The first seven correspond to the first LS seven address nibbles (A27-A0) as follows: The first nibble that appears corresponds to addresses A27-A24, the second to A23-A20, until the seventh incoming nibble, which corresponds to addresses A3-A0. Incoming nibble number eight is ignored. The MS bits of the 32-bit addresses are ‘1111’ (A31 - A28).

### Core Interrupt

Whenever there is an LPC or FWH transaction that is responded to by any of the PC87591L-N05 logical devices, a positive pulse is generated on the Host Access Wake-Up input of the MIWU module. This interrupt may be used to wake up the core for handling any host activity.

### $\overline{\text{CLKRUN}}$ Functionality

The PC87591L-N05 supports the $\overline{\text{CLKRUN}}$ I/O signal, the use of which is highly recommended in portable systems. This signal is implemented according to the specification in *PCI Mobile Design Guide, Revision 1.1*, December 18, 1998. The PC87591L-N05 supports operation with both a slow and stopped clock in ACPI state S0 (the system is active but is not being accessed). The PC87591L-N05 drives the $\overline{\text{CLKRUN}}$ low to force the LPC bus clock into full speed operation when an IRQ is pending internally and waiting to be sent through the serial IRQ.

### $\overline{\text{LPCPD}}$ Functionality

The PC87591L-N05 supports the $\overline{\text{LPCPD}}$ input. This signal is used when the $V_{DD}$ chip supply is not shared by all residents of the LPC bus. The $\overline{\text{LPCPD}}$ signal conforms with Intel’s *LPC Interface Specification, Revision 1.0*. Note that if the PC87591L-N05 power supply exists while $\overline{\text{LPCPD}}$ is active, it is not mandatory to reset the PC87591L-N05 when $\overline{\text{LPCPD}}$ is de-asserted.

# 6.0 Host-Controlled Modules and Host Interface (Continued)

## 6.1.8 SuperI/O Configuration Registers

This section describes the SuperI/O configuration and ID registers (those registers with first level indexes in the range of $20_{16}$ - $2E_{16}$). See Table 45 for a summary and directory of these registers.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

**Table 45. SuperI/O Configuration Registers**

| Index | Mnemonic | Register Name | Power Well | Type |
|---|---|---|---|---|
| $20_{16}$ | SID | SuperI/O ID | $V_{DD}$ | RO |
| $21_{16}$ | SIOCF1 | SuperI/O Configuration 1 | $V_{DD}$ | Varies per bit |
| $22_{16}$ - $24_{16}$ | Reserved exclusively for National use | | | |
| $25_{16}$ | SIOCF5 | SuperI/O Configuration 5 | $V_{DD}$ | R/W |
| $26_{16}$ | SIOCF6 | SuperI/O Configuration 6 | $V_{DD}$ | R/W |
| $27_{16}$ | SRID | SuperI/O Revision ID | $V_{DD}$ | RO |
| $28_{16}$ | SIOCF8 | SuperI/O Configuration 8 | $V_{DD}$ | R/W |
| $29_{16}$ | SIOCF9 | SuperI/O Configuration 9 | $V_{DD}$ | R/W |
| $2A_{16}$ - $2C_{16}$ | Reserved exclusively for National use | | | |
| $2D_{16}$ | SIOCFD | SuperI/O Configuration D | $V_{PP}$ | R/W |
| $2E_{16}$ | Reserved exclusively for National use | | | |

### SuperI/O ID Register (SID)

This register contains the identity number of the chip. The PC87591L-N05 is identified by the value $EC_{16}$.

Location: Index $20_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Family ID** | | | | | | | |
| Reset | $EC_{16}$ | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Family ID.** These bits identify a family of devices with similar functionality but with different options implemented. |

### SuperI/O Configuration 1 Register (SIOCF1)

Location: Index $21_{16}$

Type: Varies per bit

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | **Number of DMA Wait States** | | **Number of I/O Wait States** | | **Software Reset** | **SuperI/O Devices Enable** |
| Reset | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

## 6.0 Host-Controlled Modules and Host Interface (Continued)

| Bit | Type | Description |
|---|---|---|
| 0 | R/W | **SuperI/O Devices Enable.** Controls the function enable of all PC87591L-N05 SuperI/O logical devices, except shared memory and Mobile System Wake-Up Control (MSWC). This bit enables the simultaneous disabling of these modules using a write to a single bit.<br>0: All SuperI/O logical devices in the PC87591L-N05 are disabled, except MSWC and shared memory<br>1: Each SuperI/O logical device is enabled according to its Activate register (Index $30_{16}$) (default) |
| 1 | WO | **Software Reset.** Read always returns 0.<br>0: Ignored (default)<br>1: Triggers the Host Domain Software Reset event, which resets the logical devices (see Section 6.1.3 on page 303) |
| 3-2 | R/W | **Number of I/O Wait States.**<br>**Bits 3 2 Number**<br>0 0: 0 (default)<br>0 1: 2<br>1 0: 6<br>1 1: 12 |
| 5-4 | R/W | **Number of DMA Wait States.**<br>**Bits 5 4 Number**<br>0 0: Reserved<br>0 1: 2 (default)<br>1 0: 6<br>1 1: 12 |
| 7-6 | | **Reserved.** |

### SuperI/O Configuration 5 Register (SIOCF5)

Location: Index $25_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | **$\overline{\text{SMI}}$ to IRQ2 Enable** | **Reserved** | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Type | Description |
|---|---|---|
| 3-0 | | **Reserved.** |
| 4 | R/W | **$\overline{\text{SMI}}$ to IRQ2 Enable.** Enables using slot number 2 in the serial IRQ protocol as an SMI interrupt in parallel to or instead of using the dedicated pin.<br>0: Disabled (default)<br>1: Enabled |
| 7-5 | | **Reserved.** |

# 6.0 Host-Controlled Modules and Host Interface (Continued)

## SuperI/O Configuration 6 Register (SIOCF6)

Write access to this register can be inhibited by setting bit 7 of this register. Activation of each logical device (bits 0-4) is also affected by bit 0 of the logical device Activate register, index $30_{16}$, and bit 0 of SIOCF1 register.

Location: Index $26_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **SIOCF6 SW Lock** | **General-Purpose Scratch** | | **RTC Disabled** | **Reserved** | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 3-0 | **Reserved.** |
| 4 | **RTC Disabled.**<br>0: Enabled (default)<br>1: Disabled |
| 6-5 | **General-Purpose Scratch.** |
| 7 | **SIOCF6 Software Lock.** When set to 1 by software, it and other bits in this register can be cleared only by Host Domain Hardware reset.<br>0: Write access to bits 0-6 of this register enabled (default)<br>1: Bits 6-0 of this register are read only. |

## SuperI/O Revision ID Register (SRID)

This register contains the ID number of the specific family member (Chip ID) and the chip revision number (Chip Rev). The Chip Rev is incremented on each revision.

Location: Index $27_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Chip ID** | | | **Chip Rev** | | | | |
| Reset | See values in field description | | | X | X | X | X | X |

| Bit | Description |
|---|---|
| 4-0 | **Chip Rev.** Identifies the device revision. |
| 7-5 | **Chip ID.** Identifies a specific device.<br>**Bits**<br>**7 6 5 Device**<br>1 1 1: , PC87591L-N05<br>Other: Reserved |

# 6.0 Host-Controlled Modules and Host Interface (Continued)

## SuperI/O Configuration 8 Register (SIOCF8)

Location: Index $28_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Reserved.** |

## SuperI/O Configuration 9 Register (SIOCF9)

Location: Index $29_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | **Reserved** | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Description |
|---|---|
| 7-0 | **Reserved.** |

## SuperI/O Configuration D Register (SIOCFD)

This is a battery-backed register.

Location: Index $2D_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | **Power Supply Off** | **Power Button Mode** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 0 | **Power Button Mode.** This is a R/W bit. The value of this bit is available to the core through a register in the MSWC; see Section 5.5 on page 280.<br>0: Legacy (default at $V_{CC}$ Power-Up reset)<br>1: ACPI |
| 1 | **Power Supply Off.** This is a R/W bit. It always returns 0 when read. When using Legacy mode (bit 0 is set to 0) and setting this bit to 1, this bit indicates to the core that the host requests shutting down the power. The value of this bit is available to the core by reading a register in the MSWC; see Section 5.5 on page 280.<br>0: No action (default at $V_{PP}$ Power-Up reset)<br>1: Indicates power shut down to the core in Legacy mode |
| 7-2 | **Reserved.** |

# 6.0 Host-Controlled Modules and Host Interface (Continued)

## 6.1.9 Mobile System Wake-Up Control (MSWC) Configuration

### Logical Device 4 (MSWC) Configuration

Table 46 lists the configuration registers that affect the MSWC. See Section 6.1.2 on page 302 for a detailed description of these registers.

**Table 46. Mobile System Wake-Up Control (MSWC) Configuration Registers**

| Index | Configuration Register or Action | Type | Reset |
|---|---|---|---|
| $30_{16}$ | Activate. When bit 0 is cleared, the registers of this logical device are not accessible.[1] | R/W | $00_{16}$ |
| $60_{16}$ | Base Address MSB register. | R/W | $00_{16}$ |
| $61_{16}$ | Base Address LSB register. Bits 4-0 (for A4-0) are read only, '00000'. | R/W | $00_{16}$ |
| $70_{16}$ | Interrupt Number. | R/W | $00_{16}$ |
| $71_{16}$ | Interrupt Type. Bit 1 is read/write. Other bits are read only. | R/W | $03_{16}$ |
| $74_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |
| $75_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |

1. The logical device registers are maintained, and all wake-up detection mechanisms are functional.

## 6.1.10 Keyboard and Mouse Controller (KBC) Configuration

### Logical Devices 5 and 6 (Mouse and Keyboard) Configuration

Tables 47 and 48 list the configuration registers that affect the Mouse and the Keyboard respectively. Only the last register ($F0_{16}$) is described here. See "Standard Logical Device Configuration Register Definitions" on page 299 and Section 6.1.2 on page 302 for descriptions of the other configuration registers.

Usage Hints: It is recommended to set the type of interrupt request as level and its level of interrupt as high.

**Table 47. Mouse Configuration Registers**

| Index | Mouse Configuration Register or Action | Type | Reset |
|---|---|---|---|
| $30_{16}$ | Activate. See also bit 0 of the SIOCF1. When the Mouse of the KBC is inactive, the IRQ selected by the Mouse Interrupt Number and Wake-Up on IRQ Enable register (index $70_{16}$) is not asserted. This register has no effect on host KBC commands handling the PS/2 Mouse. | R/W | $00_{16}$ |
| $70_{16}$ | Mouse Interrupt Number and Wake-Up on IRQ Enable register. | R/W | $0C_{16}$ |
| $71_{16}$ | Mouse Interrupt Type. Bit 1 is read/write; other bits are read only. | R/W | $03_{16}$ |
| $74_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |
| $75_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |

**Table 48. Keyboard Configuration Registers**

| Index | Keyboard Configuration Register or Action | Type | Reset |
|---|---|---|---|
| $30_{16}$ | Activate. See also bit 0 of the SIOCF1. | R/W | $00_{16}$ |
| $60_{16}$ | Base Address MSB register. Bits 7-3 (for A15-11) are read only, '00000'. | R/W | $00_{16}$ |
| $61_{16}$ | Base Address LSB register. Bits 2-0 are read only, '000'. | R/W | $60_{16}$ |
| $62_{16}$ | Command Base Address MSB register. Bits 7-3 (for A15-11) are read only, '00000'. | R/W | $00_{16}$ |
| $63_{16}$ | Command Base Address LSB. Bits 2-0 are read only, '100'. | R/W | $64_{16}$ |
| $70_{16}$ | KBD Interrupt Number and Wake-Up on IRQ Enable register. | R/W | $01_{16}$ |
| $71_{16}$ | KBD Interrupt Type. Bit 1 is read/write; other bits are read only. | R/W | $03_{16}$ |
| $74_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |
| $75_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |

# 6.0 Host-Controlled Modules and Host Interface (Continued)

## 6.1.11 Shared Memory Configuration

### Logical Device 15 ($0F_{16}$) (Shared Memory) Configuration

Table 49 lists the configuration registers that affect the shared memory functional block. The shared memory base address registers point to the shared memory registers described in Section 5.3 on page 262. The memory space to which the shared memory responds is defined by the configuration registers in the following sections. See "Standard Logical Device Configuration Register Definitions" on page 299 and Section 6.1.2 on page 302 for a detailed description of the other configuration registers.

**Table 49. Shared Memory Configuration Registers**

| Index | Configuration Register or Action | Type | Reset |
|---|---|---|---|
| $30_{16}$ | Activate. When bit 0 is cleared, the registers of this logical device are not accessible. | R/W | $00_{16}$ |
| $60_{16}$ | Base Address MSB register. | R/W | $00_{16}$ |
| $61_{16}$ | Base Address LSB register. Bits 3-0 (for A3-A0) are read only, '0000'. | R/W | $00_{16}$ |
| $70_{16}$ | No interrupt assignment. | RO | $00_{16}$ |
| $71_{16}$ | No interrupt assignment. | RO | $00_{16}$ |
| $74_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |
| $75_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |
| $F4_{16}$ | Shared Memory Configuration register. | R/W | $00_{16}$ or $09_{16}$, depending on the value of the SHBM strap input |
| $F5_{16}$ | Shared Memory Base Address High Byte register. | R/W | $00_{16}$ |
| $F6_{16}$ | Shared Memory Base Address Low Byte register. | R/W | $00_{16}$ |
| $F7_{16}$ | Shared Memory Size Configuration register. | R/W | $00_{16}$ |

### Memory Range Programing

LPC memory transactions and/or LPC-FWH transactions can be forwarded to the PC87591L-N05 shared memory. The Shared Memory Configuration register defines the transaction type and address range to which the PC87591L-N05 responds. The SHBM strap inputs affect the default settings of the Shared Memory Configuration register to enable boot process from shared memories. Two memory areas may be individually enabled: a user-defined zone and BIOS memory (either BIOS-LPC and/or BIOS-FWH spaces).

To enable BIOS support, set the SHBM strap inputs to select any of the BIOS modes (see Section 2.3 on page 48 for details). The PC87591L-N05 responds to LPC memory read and write transactions from/to the BIOS address spaces, shown in Table 50, as long as BIOS LPC Enable (bit 0) of the Shared Memory Configuration register is set.

**Table 50. BIOS-LPC Memory Space Definition**

| Memory Address Range | Description |
|---|---|
| 000E $0000_{16}$ - 000E $FFFF_{16}$ | Extended BIOS range (Legacy); only when Extended BIOS Enable bit in Shared Memory Range Configuration register is set. |
| 000F $0000_{16}$ - 000F $FFFF_{16}$ | BIOS Range (Legacy) |
| FFE0 $0000_{16}$ - FFFF $FFFF_{16}$ | 386 mode BIOS range; this is the upper 2 Mbytes of the memory space. |

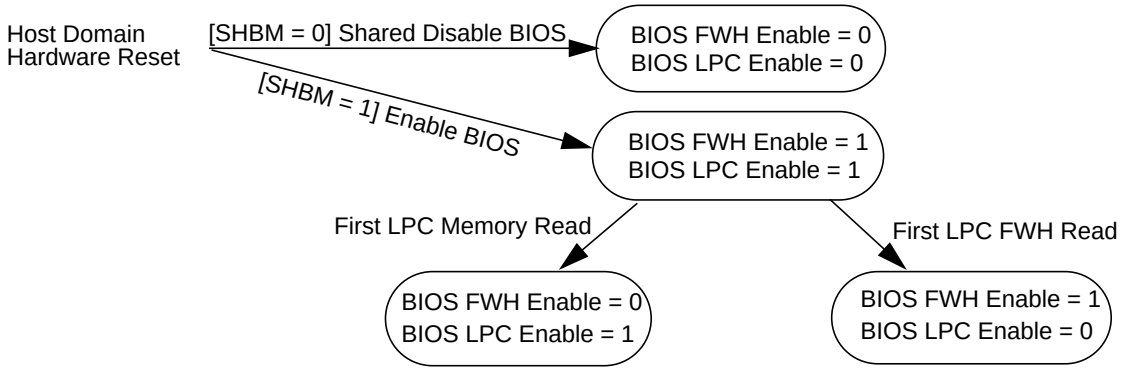
The PC87591L-N05 responds to LPC-FWH read and write transactions from/to the high memory address range ('386' mode BIOS range), shown in Table 50, as long as BIOS FWH Enable (bit 3) of the Shared Memory Configuration register is set.

## 6.0 Host-Controlled Modules and Host Interface (Continued)

**Table 51. BIOS-FWH Memory Space Definition**

| Memory Address Range | Description |
|---|---|
| FFE0 $0000_{16}$ - FFFF $FFFF_{16}$ | 386 mode BIOS range; this is the upper 2 Mbytes of the memory space. The PC87591L-N05 uses the first 21 address lines and ID field to identify FWH access to the shared memory. |

On host domain hardware reset in BIOS mode, the BIOS LPC Enable bit is set and the BIOS FWH Enable bit is set. The PC87591L-N05 automatically detects the type of host boot protocol in use, via the first completed BIOS read operation after host domain hardware reset. If the first read is an LPC memory read, the BIOS FWH Enable bit is cleared. If the first read is an LPC-FWH read, the BIOS LPC Enable bit is cleared. Any other LPC or LPC-FWH transactions are ignored. The bits are cleared only by the first read operation, allowing software to enable responding to these address ranges by setting the bit. Figure 104 illustrates this behavior.




**Note:** Only hardware-controlled transitions are shown;
other transitions are possible via software writes to the bits.

**Figure 104. BIOS Mapping Enable Scheme**

The user-defined shared memory enables sharing the memory without using it as a shared BIOS memory. The memory base address in the host address space and the size of the shared memory are defined in the Shared Memory Base and Shared Memory Size registers. Address bits above the block size are ignored and are internally replaced with 1s for the purpose of address translation (e.g., for a 2 Mbyte block size, A21 through A31 are replaced with 1s).

### Shared Memory Configuration Register

This register is reset on host domain hardware reset to $00_{16}$ or $09_{16}$, depending on the value of the SHBM strap input.

Location: Index $F4_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **BIOS FWH ID** | | | | **BIOS FWH Enable** | **User-Defined Memory Space Enable** | **BIOS Extended Space Enable** | **BIOS LPC Enable** |
| Reset | 0 | 0 | 0 | 0 | Strap | 0 | 0 | Strap |

| Bit | Description |
|---|---|
| 0 | **BIOS LPC Enable.** Enables the PC87591L-N05 to respond to LPC memory accesses to the BIOS-LPC space. The reset value of this register is defined by the SHBM configuration input. The value of this bit is updated later, based on the detected host BIOS scheme; see “Memory Range Programing” on page 311 for details.<br>0: Disabled (default when SHBM disable BIOS configuration)<br>1: Enabled (default when SHBM enable BIOS configuration) |

## 6.0 Host-Controlled Modules and Host Interface (Continued)

| Bit | Description |
|---|---|
| 1 | **BIOS Extended Space Enable.** Expands the BIOS address space to which the PC87591L-N05 responds to include the Extended BIOS address range.<br>0: Disabled (default)<br>1: Enabled |
| 2 | **User-Defined Memory Space Enable.** When set, enables the PC87591L-N05 to respond to LPC memory read and write accesses in the user-defined memory area range. The base address and size of the user-defined range are specified by the Shared Memory Base Address High and Low Byte registers and the Shared Memory Size Configuration register.<br>0: Disabled (default)<br>1: Enabled |
| 3 | **BIOS FWH Enable.** When set, enables PC87591L-N05 response to LPC-FWH transactions to the BIOS-FWH space. The reset value of this register is defined by the SHBM configuration input. The value of this bit is later updated based on the detected host BIOS scheme; see “Memory Range Programing” on page 311 for details.<br>0: Disabled (default when SHBM disable BIOS configuration)<br>1: Enabled (default when SHBM enable BIOS configuration) |
| 7-4 | **BIOS FWH ID.** These four bits correspond to the identification nibble, which is part of a FWH transaction (see Section 6.1.7 for details). |

### Shared Memory Base Address High Byte Register

This register describes the high byte for the user-defined memory zone mapped to the shared memory (decoded as bits 31 to 24 of the 32-bit address range, bits 15-0 are 0). This register is reset to $00_{16}$ on Host Domain Hardware reset.

Location: Index $F5_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **User-Defined Memory Zone Address High** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **User-Defined Memory Zone Address High.** Defines the higher eight bits of the user-defined memory block base address. The base address should be aligned on the selected block size. |

### Shared Memory Base Address Low Byte Register

This register describes the low byte for the user-defined memory zone mapped to the shared memory (decoded as bits 23 to 16 of the 32-bit address range, bits 15-0 are 0). This register is reset to $00_{16}$ on Host Domain Hardware reset.

Location: Index $F6_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **User-Defined Memory Zone Address Low** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **User-Defined Memory Zone Address Low.** Defines the lower eight bits of the user-defined memory block base address. The base address should be aligned on the selected block size. |

# 6.0 Host-Controlled Modules and Host Interface (Continued)

## Shared Memory Size Configuration Register

This register defines the size of the user-defined memory zone mapped to the shared memory. This register is reset to $00_{16}$ on host domain hardware reset.

Location: Index $F7_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | **User-Defined Memory Zone Size** | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 3-0 | **User-Defined Memory Zone Size.** Defines the size, in bytes, of the zone window. The size is defined as an exponent of two, using the equation: NumOfBytes = $2^n$ (where, n = User-Defined Memory Zone size+16). The zone must always be aligned to the window size (i.e., for a 128 Kbyte window, the 17 LSBs of the address should be zero).<br>**Bits 3 2 1 0 — Size (Bytes)**<br>0 0 0 0 — 64K (default)<br>. . .<br>0 1 0 1 — 2M<br>Other — Reserved |
| 7-4 | **Reserved.** |

# 6.0 Host-Controlled Modules and Host Interface (Continued)

## 6.1.12 Real Time Clock (RTC) Configuration

### Logical Device 16 ($10_{16}$) RTC Configuration

Table 52 lists the configuration registers that affect the RTC. See Section 6.1.1 on page 297 and Section 6.1.2 on page 302 for descriptions of the other configuration registers.

**Table 52. RTC Configuration Registers**

| Index | RTC Configuration Register or Action | Type | Reset |
|---|---|---|---|
| $30_{16}$ | Activate. When bit 0 is cleared, the registers of this logical device are not accessible. | R/W | $00_{16}$ |
| $60_{16}$ | Base Address MSB register. | R/W | $00_{16}$ |
| $61_{16}$ | Base Address LSB register. Bit 0 (for A0) is read only, '0'. | R/W | $70_{16}$ |
| $62_{16}$ | Base Address MSB register. | R/W | $00_{16}$ |
| $63_{16}$ | Base Address LSB register. Bit 0 (for A0) is read only, '0'. | R/W | $72_{16}$ |
| $70_{16}$ | RTC Interrupt Number and Wake-Up on IRQ Enable register. | R/W | $08_{16}$ |
| $71_{16}$ | RTC Interrupt Type. Bit 1 is read/write; other bits are read only. | R/W | $00_{16}$ |
| $74_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |
| $75_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |
| $F0_{16}$ | RAM Lock Register (RLR) | R/W | $00_{16}$ |
| $F1_{16}$ | Date of Month Alarm Register Offset (DOMAO) | R/W | $00_{16}$ |
| $F2_{16}$ | Month Alarm Register Offset (MONAO) | R/W | $00_{16}$ |
| $F3_{16}$ | Century Register Offset (CENO) | R/W | $00_{16}$ |

### RAM Lock Register (RLR)

When a non-reserved bit is set to 1, it can be cleared only by host domain hardware reset.

Location: Index $F0_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | **Block Standard RAM** | **Block RAM Write** | **Block Extended RAM Write** | **Block Extended RAM Read** | **Block Extended RAM** | **Reserved** | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 2-0 | **Reserved.** |
| 3 | **Block Extended RAM.** Controls access to the Extended RAM 128 bytes.<br>0: No effect on Extended RAM access (default)<br>1: Reads and writes to the Extended RAM are blocked, writes are ignored and reads return $FF_{16}$ |
| 4 | **Block Extended RAM Read.** Controls read from bytes $00_{16}$-$1F_{16}$ of the Extended RAM.<br>0: No effect on Extended RAM access (default)<br>1: Reads to bytes $00_{16}$-$1F_{16}$ of the Extended RAM are ignored; reads return $FF_{16}$ |
| 5 | **Block Extended RAM Write.** Controls writes to bytes $00_{16}$-$1F_{16}$ of the Extended RAM.<br>0: No effect on the Extended RAM access (default)<br>1: Writes to bytes $00_{16}$-$1F_{16}$ of the Extended RAM are ignored |

## 6.0 Host-Controlled Modules and Host Interface (Continued)

| Bit | Description |
|---|---|
| 6 | **Block RAM Write.**<br>0: No effect on RAM access (default)<br>1: Writes to RAM (Standard and Extended) are ignored |
| 7 | **Block Standard RAM.**<br>0: No effect on Standard RAM access (default)<br>1: Reads and writes to locations $38_{16}$-$3F_{16}$ of the Standard RAM are blocked, writes are ignored and reads return $FF_{16}$ |

### Date Of Month Alarm Register Offset (DOMAO)

Location: Index $F1_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **Date of Month Alarm Register Offset Value** | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 6-0 | **Date of Month Alarm Register Offset Value.** |
| 7 | **Reserved.** |

### Month Alarm Register Offset (MONAO)

Location: Index $F2_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **Month Alarm Register Offset Value** | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 6-0 | **Month Alarm Register Offset Value.** |
| 7 | **Reserved.** |

### Century Register Offset (CENO)

Location: Index $F3_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | **Century Register Offset Value** | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 6-0 | **Century Register Offset Value.** |
| 7 | **Reserved.** |

# 6.0 Host-Controlled Modules and Host Interface (Continued)

## 6.1.13 Power Management Interface Channel 1 Configuration

### Logical Device 17 ($11_{16}$) Power Management Channel 1

Table 53 lists the configuration registers that affect Power Management I/F Channel 1.

See “Standard Logical Device Configuration Register Definitions” on page 299 and Section 6.1.2 on page 302 for descriptions of the other configuration registers.

**Table 53. Power Management Configuration Registers**

| Index | Power Management Channel 1 Configuration Register or Action | Type | Reset |
|---|---|---|---|
| $30_{16}$ | Activate. See also bit 0 of the SIOCF1. | R/W | $00_{16}$ |
| $60_{16}$ | Data Register Base Address MSB register. Bits 7-3 (for A15-11) are read only, ‘00000’. | R/W | $00_{16}$ |
| $61_{16}$ | Data Register Base Address LSB register. | R/W | $62_{16}$ |
| $62_{16}$ | Command/Status Base Address MSB register. Bits 7-3 (for A15-11) are read only, ‘00000’. | R/W | $00_{16}$ |
| $63_{16}$ | Command/Status Base Address LSB. | R/W | $66_{16}$ |
| $70_{16}$ | PM Interrupt Number and Wake-Up on IRQ Enable register. | R/W | $01_{16}$ |
| $71_{16}$ | PM Interrupt Type. Bit 1 is read/write; other bits are read only. | R/W | $03_{16}$ |
| $74_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |
| $75_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |

## 6.1.14 Power Management Interface Channel 2 Configuration

### Logical Device 18 ($12_{16}$) Power Management Channel 2

Table 54 lists the configuration registers that affect Power Management I/F Channel 2. See Section 6.1.2 on page 302 for descriptions of the other configuration registers.

**Table 54. Power Management Configuration Registers**

| Index | Power Management 2 Configuration Register or Action | Type | Reset |
|---|---|---|---|
| $30_{16}$ | Activate. See also bit 0 of the SIOCF1. | R/W | $00_{16}$ |
| $60_{16}$ | Data Register Base Address MSB register. Bits 7-3 (for A15-11) are read only, ‘00000‘. | R/W | $00_{16}$ |
| $61_{16}$ | Data Register Base Address LSB register. | R/W | $68_{16}$ |
| $62_{16}$ | Command/Status Base Address MSB register. Bits 7-3 (for A15-11) are read only, ‘00000’. | R/W | $00_{16}$ |
| $63_{16}$ | Command/Status Base Address LSB. | R/W | $6C_{16}$ |
| $70_{16}$ | PM Interrupt Number and Wake-Up on IRQ Enable register. | R/W | $01_{16}$ |
| $71_{16}$ | PM Interrupt Type. Bit 1 is read/write; other bits are read only. | R/W | $03_{16}$ |
| $74_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |
| $75_{16}$ | Report no DMA assignment. | RO | $04_{16}$ |

## 6.2 REAL-TIME CLOCK (RTC)

The RTC provides timekeeping and calendar management capabilities. It uses a 32.768 KHz signal as the basic clock for timekeeping. The RTC also includes 242 bytes of battery-backed RAM for general-purpose use.

The RTC provides the following functions:

- Accurate timekeeping and calendar management
- Alarm at a predetermined time and/or date
- Three programmable interrupt sources
- Valid timekeeping during power-down, by utilizing external battery backup
- 242 bytes of battery-backed RAM
- RAM lock schemes to protect its content
- Internal oscillator circuit (the crystal itself is off-chip) or external clock supply for the 32.768 KHz clock
- A century counter
- PnP support:
  - Relocatable index and data registers
  - Module access enable/disable option
  - Host interrupt enable/disable option
- Additional low-power features such as:
  - Automatic switching from battery to $V_{CC}$
  - Internal power monitoring on the VRT bit
- Software compatible with the DS1287 and MC146818

### 6.2.1 Bus Interface

The RTC function is initially mapped to the default SuperI/O locations at indexes $70_{16}$ to $73_{16}$ (two Index/Data pairs). These locations may be reassigned in compliance with Plug and Play requirements.

### 6.2.2 RTC Clock Generation

The RTC uses a 32.768 KHz clock signal as the basic clock for timekeeping. The 32.768 KHz clock is generated by the internal oscillator circuit or by an external oscillator (see Sections 6.2.3 and 6.2.4).

### 6.2.3 Internal Oscillator

The internal oscillator employs an external crystal connected to the on-chip amplifier. The on-chip amplifier is accessible on the 32KX1 input pin and the 32KX2 output pin. See Figure 105 for the recommended external circuit and Table 55 for a listing of the circuit components. The oscillator may be disabled in certain conditions. See Section 6.2.12 on page 324 for more details.

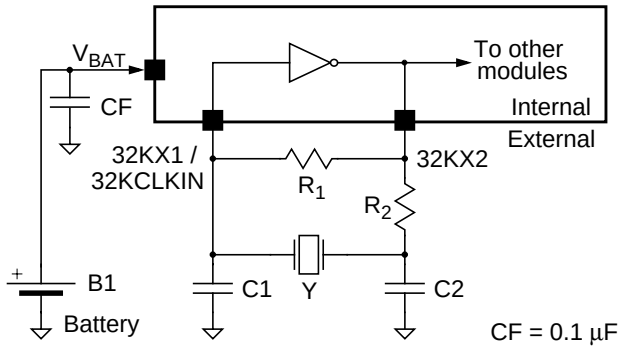



**Figure 105. Recommended Oscillator External Circuitry**

**Table 55. Crystal Oscillator Circuit Components**

| Component | Parameters | Values | Tolerance |
|---|---|---|---|
| Crystal | Resonance Frequency | 32.768 KHz Parallel Mode | User defined |
| | Type | N-Cut or XY-bar | |
| | Serial Resistance | 40 KΩ | Max |
| | Quality Factor, Q | 35000 | Min |
| | Shunt Capacitance | 2 pF | Max |
| | Load Capacitance, $C_L$ | 9-13 pF | |
| | Temperature Coefficient | User defined | |
| Resistor $R_1$ | Resistance | 20 MΩ | 5% |
| Resistor $R_2$ | Resistance | 121 KΩ | 5% |
| Capacitor $C_1$ | Capacitance | 5-20 pF | 10% |
| Capacitor $C_2$ | Capacitance | 2-15 pF | 10% |

**External Elements**

Choose C1 and C2 capacitors (see Figure 105) to match the crystal's load capacitance. The load capacitance $C_L$ "seen" by crystal Y is comprised of $C_1$ in series with $C_2$ and in parallel with the parasitic capacitance of the circuit. The parasitic capacitance is caused by the chip package, board layout and socket (if any), and can vary from 0 to 8 pF. The rule of thumb for choosing these capacitors is:

$$C_L = (C_1 * C_2) / (C_1 + C_2) + C_{PARASITIC}$$

To provide accurate 32.768 KHz frequency, the value of $C_1$ and $C_2$ capacitors may vary (within the range defined in Table 55), according to the specific board implementation. See the *PC87591x RTC Oscillator Circuit Design and Layout Guidelines* Application Note for details on how to select the capacitors value and how to lay out the PCB.

**Oscillator Start-Up**

The oscillator starts to generate 32.768 KHz pulses to the RTC after approximately $t_{32KW}$ from when $V_{BAT}$ is higher than $V_{BATMIN}$ (2.4 V) or from when $V_{CC}$ is higher than $V_{CCMIN}$ (3.0V).

$C_1$ can be trimmed to achieve precisely 32.768 KHz. To achieve a high time accuracy, use crystal and capacitors with low tolerance and temperature coefficients.

### 6.2.4 External Oscillator

32.768 KHz can be applied from an external clock source, as shown in Figure 106.

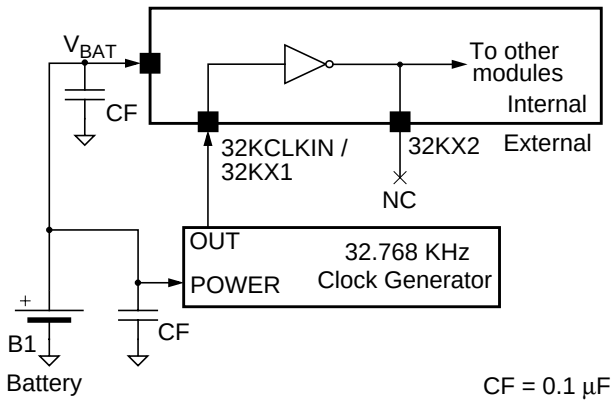



**Figure 106. External Oscillator Connections**

# Host-Controlled Modules and Host Interface (Continued)

## Connections

Connect the clock to the 32KCLKIN pin, leaving the oscillator output, 32KX2, unconnected.

## Signal Parameters

The signal levels should conform to the voltage level requirements for 32KCLKIN/32KX1, stated in Chapter 7 on page 334. The signal should have a duty cycle of approximately 50%. To oscillate during power-down, it should be sourced from a battery-backed source. This guarantees that the RTC delivers updated time/calendar information.

### 6.2.5 Timing Generation

The timing generation function divides the 32.768 KHz clock by $2^{15}$ to derive a 1 Hz signal, which serves as the input for the seconds counter. This is performed by a divider chain composed of 15 divide-by-two latches, as shown in Figure 107.

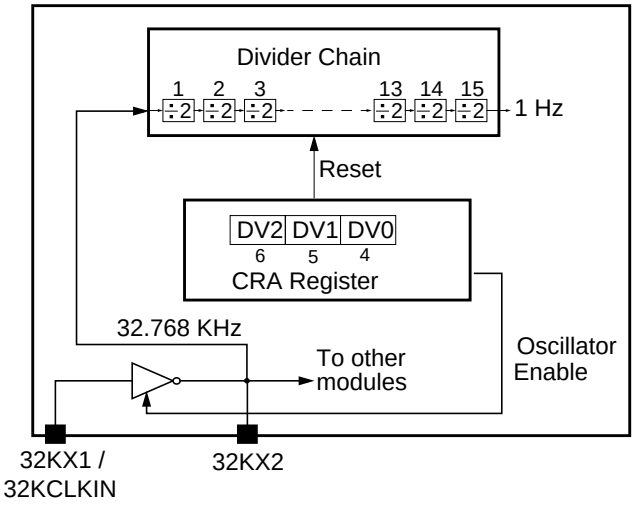



**Figure 107. Divider Chain Control**

Bits 6-4 (DV2-0) of CRA register control the following functions:

- Normal operation of the divider chain (counting)
- Divider chain reset to 0
- Oscillator activity when only $V_{BAT}$ power is present (backup state).

The divider chain can be activated by setting normal operational mode (bits 6-4 of CRA = $010_2$). The first update occurs 500 ms after divider chain activation.

Bits 3-0 of CRA register select one the of 15 taps from the divider chain to be used as a periodic interrupt. The periodic flag becomes active after half of the programed period has elapsed, following divider chain activation.

See “RTC Control Register A (CRA)” on page 329 for more details.

### 6.2.6 Timekeeping

#### Data Format

Time is kept in BCD or binary format, as determined by bit 2 (DM) of Control Register B (CRB), and in either 12 or 24-hour format, as determined by bit 1 of this register.

**Note:** When changing the above formats, re-initialize all the time registers.

#### Daylight Saving

Daylight saving time exceptions are handled automatically, as described in “RTC Control Register B (CRB)” in Section 6.2.15 on page 325.

#### Leap Years

Leap year exceptions are handled automatically by the internal calendar function. Every four years, February is extended to 29 days. Year 2000 is a leap year.

## Host-Controlled Modules and Host Interface (Continued)

### 6.2.7 Updating

The time and calendar registers are updated once per second regardless of bit 7 (SET) of CRB register. Since the time and calendar registers are updated serially, unpredictable results may occur if they are accessed during the update. Therefore, it is essential to ensure that reading or writing to the time storage locations does not coincide with a system update of these locations. There are several methods to avoid this contention.

**Method 1**

1. Set bit 7 of CRB register to 1. This takes a “snapshot” of the internal time registers and loads them into the user copy registers. The user copy registers are seen when accessing the RTC from outside and are part of the double buffering mechanism. This bit may be kept set for up to 1 second, since the time/calendar chain continues to be updated once per second.
2. Read or write the required registers (since bit 1 is set, the access is to the user copy registers). If a read operation is performed, the information read is correct from the time bit 1 was set. If a write operation is performed, the write is only to the user copy registers.
3. Reset bit 1 to 0. During the transition, the user copy registers update the internal registers, using the double buffering mechanism to ensure that the update is performed between two time updates. This mechanism enables new time parameters to be loaded in the RTC.

**Method 2**

1. Access the RTC registers after detection of an Update Ended interrupt. This implies that an update has just been completed, and 999 ms remain until the next update.
2. To detect an Update Ended interrupt, do one of the following:
   - Poll bit 4 of CRC register.
   - Use the following interrupt routine:
     a. Set bit 4 of CRB register.
     b. Wait for an interrupt from interrupt pin.
     c. Clear the IRQF flag of CRC register before exiting the interrupt routine.

**Method 3**

Poll bit 7 of CRA register. The update occurs 244 µs after this bit goes high. Therefore, if a 0 is read, the time registers remain stable for at least 244 µs.

**Method 4**

Use a periodic interrupt routine to determine if an update cycle is in progress, as follows:

1. Set the periodic interrupt to the desired period.
2. Set bit 6 of CRB register to enable the interrupt from periodic interrupt.
3. Wait for the periodic interrupt to occur. This indicates that the period represented by the following expression remains until another update occurs:

   [(Period of periodic interrupt / 2) + 244 µs]

### 6.2.8 Alarms

The timekeeping function can be set to generate an alarm when the current time reaches a stored alarm time. After each RTC time update (every 1 second), the seconds, minutes, hours, date of month and month counters are compared with their corresponding registers in the alarm settings. If equal, bit 5 of CRC register is set. Bit 5 of CRC is sent to the MSWC as an alarm signal. If the Alarm Interrupt Enable bit was previously set (bit 5 of CRB register), interrupt request pin is also active.

Any alarm register may be set to “Unconditional Match” by setting bits 7-6 to ‘11’. This combination, not used by any BCD or binary time codes, results in a periodic alarm. The rate of this periodic alarm is determined by the registers that were set to “Unconditional Match”.

For example, if all but the seconds and minutes alarm registers are set to “Unconditional Match”, an interrupt is generated every hour at the specified minute and second. If all but the seconds, minutes and hours alarm registers are set to “Unconditional Match”, an interrupt is generated every day at the specified hour, minute and second.

## Host-Controlled Modules and Host Interface (Continued)

### 6.2.9 Power Supply

The PC87591L-N05 is supplied from four supply voltages, as shown in Figure 108 (see Section 3.1 on page 58 for more detailed description):

- System power supply voltage, $V_{DD}$
- System analog power supply voltage, $AV_{CC}$
- System standby power supply voltage, $V_{CC}$
- Backup voltage, from low-capacity Lithium battery

A standby voltage ($V_{CC}$) from the external AC/DC power supply powers the RTC under normal conditions.

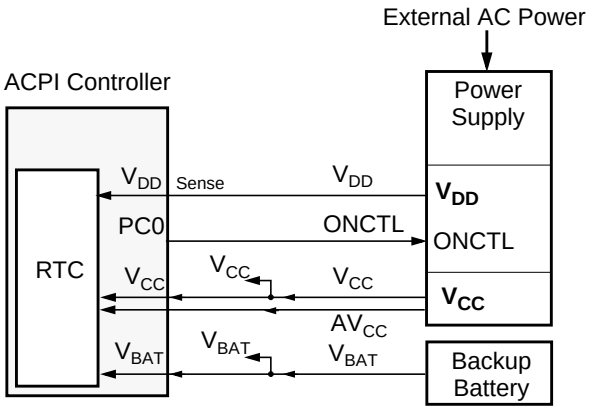
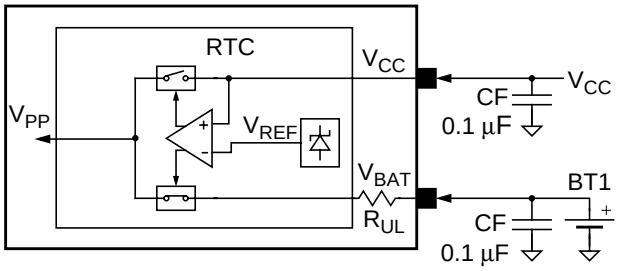



**Figure 108. Power Supply Connections**

Figure 109 shows a typical battery configuration. No external diode is required to meet the UL standard, because of the internal switch and internal serial resistor $R_{UL}$.




1. Place a 0.1 µF capacitor on each $V_{CC}$ power supply pin as close as possible to the pin, and also on $V_{BAT}$.
2. Place a 10-47 µF capacitor on the common power supply net as close as possible to the device.

**Figure 109. Typical Battery Configuration**

RTC power is supplied from either $V_{CC}$ or $V_{BAT}$ voltage source, according to the sources' levels. An internal voltage comparator delivers the control signals to a pair of switches. When $V_{CC}$ is active, the RTC power is drawn from it. Battery backup voltage ($V_{BAT}$) maintains the correct time and saves the CMOS memory when the $V_{CC}$ voltage is absent due to power failure, disconnection of the external AC/DC input power supply or disconnection of the main battery. Figure 109 illustrates the mechanism whereby $V_{CC}$ or $V_{BAT}$ is selected.

To assure that the module uses power from $V_{CC}$ and not from $V_{BAT}$, the $V_{CC}$ voltage should be maintained above its minimum ($V_{CC2PP}$), as detailed in Chapter 7 on page 334. Figure 110 illustrates the switching between $V_{CC}$ (thick gray line) and $V_{BAT}$ (thin black line) and vice-versa for generating $V_{PP}$ (thick black line).

The actual voltage point where the PC87591L-N05 switches from $V_{BAT}$ to $V_{CC}$ is lower than the minimum workable battery voltage but high enough to guarantee the correct functionality of the oscillator and the CMOS RAM.

## Host-Controlled Modules and Host Interface (Continued)

The Valid RAM and Timer bit (VRT) in CRD register indicates the state of non-interrupted power supplied to the PC87591L-N05 RTC module. VRT is cleared once $V_{PP}$ power is lost, i.e., when $V_{CC}$ and $V_{BAT}$ are both below $V_{LOWBAT.}$ The host code should read this bit as part of the startup routine, before using the timer or the CMOS-RAM contents. In case a value of 0 is read, the RAM and timer must be initialized before use. The VRT bit is set once read.

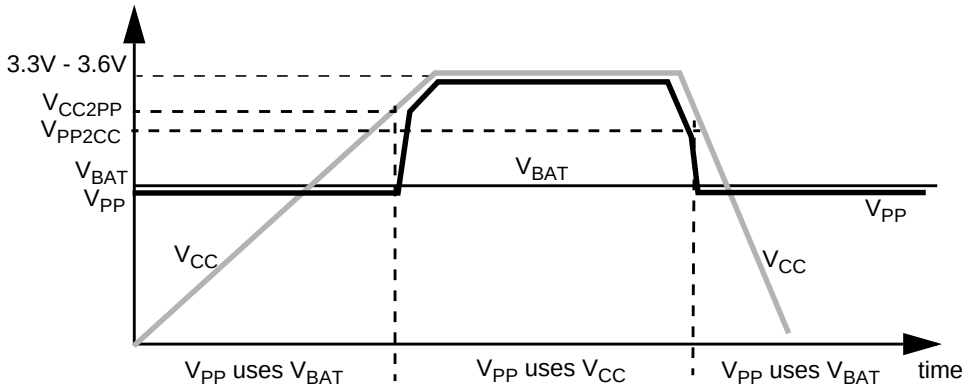



**Figure 110. $V_{PP}$ Generation Using $V_{CC}$ or $V_{BAT}$**

Figure 111 shows typical battery current consumption during battery-backed operation; Figure 112 shows typical battery current consumption during normal operation.

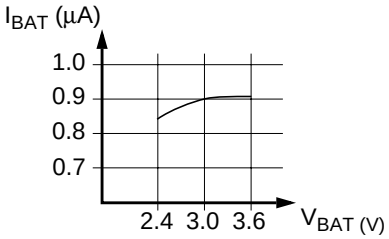
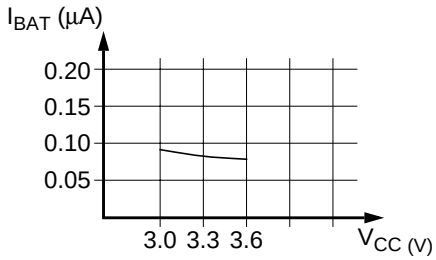



**Figure 111. Typical Battery Current During Battery-Backed Power Mode**




**Note:** Battery voltage in this test is 3.0V.

**Figure 112. Typical Battery Current During Normal Operation Mode**

### 6.2.10 System Bus Lockout

During power on or power off, spurious bus transactions from the host may occur. To protect the RTC internal registers from corruption, all inputs are automatically locked out. The lockout condition is asserted when $V_{CC}$ is lower than $V_{CCON}$.

### 6.2.11 Power-Up Detection

When system power is restored after a power failure or power off state ($V_{CC}$=0), the lockout condition continues for a delay of 62 ms (minimum) to 125 ms (maximum) after the RTC switches from battery to system power.

## Host-Controlled Modules and Host Interface (Continued)

The lockout condition is switched off immediately in the following situations:

- If the Divider Chain Control bits, DV0-2, (bits 6-4 in CRA register) specify a normal operation mode ($010_2$), all input signals are enabled immediately on detection of system voltage above $V_{CCON}$.
- When battery voltage is below $V_{BATDCT}$ and host domain hardware reset is active, all input signals are enabled immediately on detection of system voltage above $V_{CCON}$. This also initializes registers at offsets $00_{16}$ through $0D_{16}$.
- If bit 7 (VRT) of CRD register is 0, all input signals are enabled immediately on detection of system voltage above $V_{CCON}$.

### 6.2.12 Oscillator Activity

The RTC oscillator is active if:

- $V_{CC}$ power supply is higher than $V_{CCON}$, independent of the battery voltage, $V_{BAT}$.
- $V_{BAT}$ power supply is higher than $V_{BATMIN}$ and $V_{CC}$ is not present.

The RTC oscillator is disabled in the following cases:

- During power-down ($V_{BAT}$ only), if the battery voltage drops below $V_{BATMIN}$, the PC87591L-N05 may enter Battery Fail state. In this case, the oscillator may stop oscillating and memory contents may be corrupted or lost.
- The software wrote '00X' to DV2-0 bits of CRA register. This disables the oscillator and, when $V_{CC}$ is not present, it decreases the power consumption from the battery connected to the $V_{BAT}$ pin. When disabling the oscillator, the CMOS RAM is not affected as long as the battery is present at a correct voltage level. Oscillation is resumed, either by changing the DV2-0 bits, or after a $V_{PP}$ Power-Up reset.

If the RTC oscillator becomes inactive, the following features are non-functional/disabled:

- Timekeeping
- Periodic interrupt
- Alarm

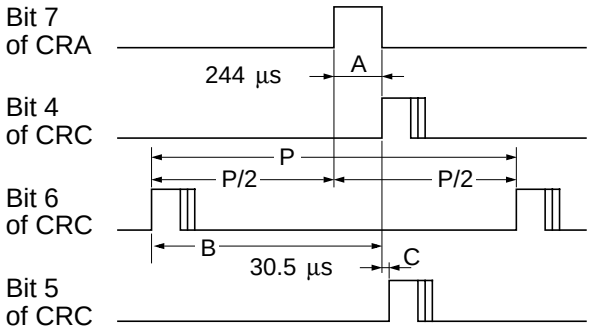
### 6.2.13 Interrupt Handling

The RTC has a single Interrupt Request line, which handles the following three interrupt conditions:

- Periodic interrupt
- Alarm interrupt
- Update end interrupt

The interrupts are generated if the respective enable bits in CRB register are set prior to an interrupt event occurrence. Reading the CRC register clears all interrupt flags. Therefore, when multiple interrupts are enabled, the interrupt service routine should first read and store the CRC register and then handle all pending interrupts by referring to this stored status.

If an interrupt is not serviced before a second occurrence of the same interrupt condition, the second interrupt event is lost. Figure 113 shows the interrupt timing in the RTC.




Flags (and IRQ) are reset at the conclusion of CRC read or by reset.

A = Update In Progress bit high before update occurs = 244 µs
B = Periodic interrupt to update
= Period (periodic int) / 2 + 244 µs
C = Update to Alarm Interrupt = 30.5 µs
P = Period is programed by RS3-0 of CRA

**Figure 113. Interrupt/Status Timing**

### 6.2.14 Battery-Backed RAMs and Registers

The RTC has two battery-backed RAMs and 17 registers, used by the logical units themselves. Battery-backup power enables information retention during system power down.

The RAMs are:

- Standard RAM
- Extended RAM

The memory maps and register content of the RAMs are shown in Section 6.2.18 on page 333 and shown in Figure 114.

The first 14 bytes and three programmable bytes of the Standard RAM are overlaid by time, alarm data and control registers. The remaining 111 bytes are general-purpose memory.

Registers with reserved bits should be written using the “Read-Modify-Write” method.

All register locations within the device are accessed by the RTC Index and Data registers (at base address and base address+1, as defined by RTC configuration registers at index $60_{16}$ and $61_{16}$). The Index register points to the register location being accessed, and the Data register contains the data to be transferred to or from the location. An additional 128 bytes of battery-backed RAM (also called Extended RAM) may be accessed via a second pair of Index and Data registers (at base address and base address+1, as defined by RTC configuration registers in index $62_{16}$ and $63_{16}$).

Access to the two RAMs may be locked. For details, see “RAM Lock Register (RLR)” in Section 6.1.12 on page 315.

The index of three of the RTC registers is programmable using registers in the RTC logical device bank (part of the SuperI/O configuration registers). If enabled, these registers override three of the Standard RAM locations; see Section 6.1.12 on page 315.

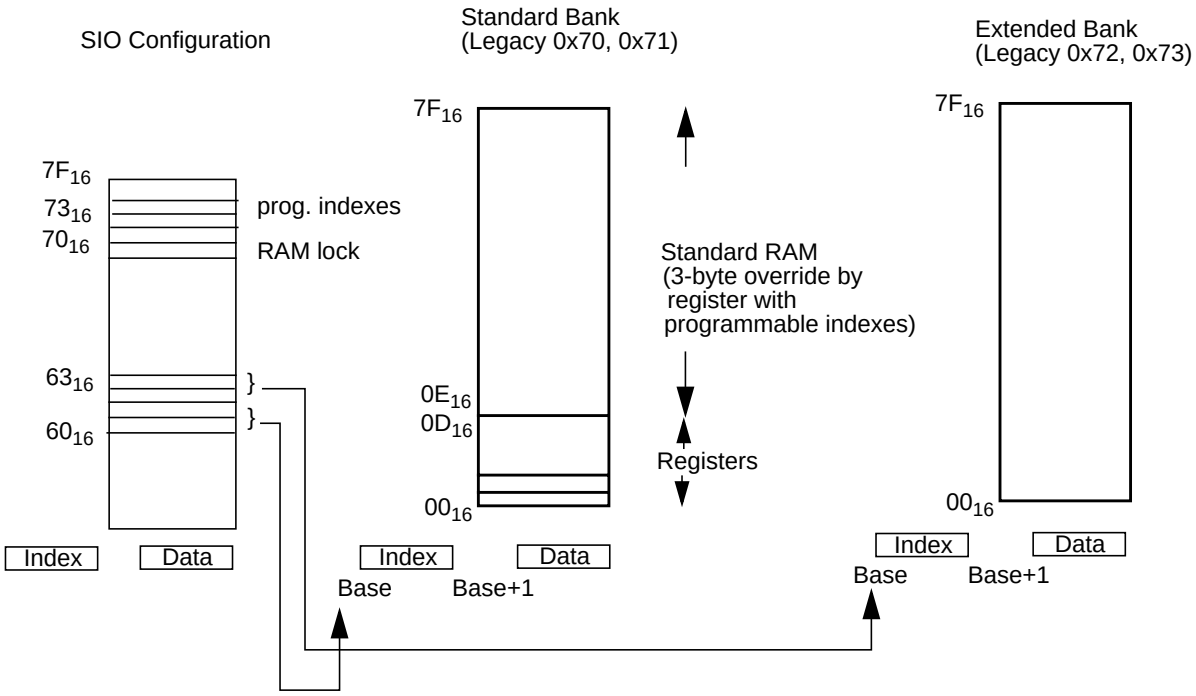



**Figure 114. RTC Module Registers Mapping**

### 6.2.15 RTC Registers

The RTC registers can be accessed at any time during normal operation mode; i.e., when $V_{CC}$ is within the recommended operation range. This access is disabled during battery-backed operation. Write operation to these registers is also disabled if bit 7 of CRD register is 0.

**Note:** Before attempting to perform any start-up procedures, read the explanation of bit 7 (VRT) of CRD register.

See Section 6.2.18 on page 333 for a detailed description of the memory map for the RTC registers.

This section describes the RTC Timing and Control registers, which control basic RTC functionality.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 32.

## Host-Controlled Modules and Host Interface (Continued)

### RTC Register Map

| Index | Mnemonic | Name | Type | Reset |
|---|---|---|---|---|
| $00_{16}$ | SEC | Seconds Register | R/W | $V_{PP}$ PUR |
| $01_{16}$ | SECA | Seconds Alarm Register | R/W | $V_{PP}$ PUR |
| $02_{16}$ | MIN | Minutes Register | R/W | $V_{PP}$ PUR |
| $03_{16}$ | MINA | Minutes Alarm Register | R/W | $V_{PP}$ PUR |
| $04_{16}$ | HOR | Hours Register | R/W | $V_{PP}$ PUR |
| $05_{16}$ | HORA | Hours Alarm Register | R/W | $V_{PP}$ PUR |
| $06_{16}$ | DOW | Day Of Week Register | R/W | $V_{PP}$ PUR |
| $07_{16}$ | DOM | Date Of Month Register | R/W | $V_{PP}$ PUR |
| $08_{16}$ | MON | Month Register | R/W | $V_{PP}$ PUR |
| $09_{16}$ | YER | Year Register | R/W | $V_{PP}$ PUR |
| $0A_{16}$ | CRA | RTC Control Register A | R/W | Bit specific |
| $0B_{16}$ | CRB | RTC Control Register B | R/W | Bit specific |
| $0C_{16}$ | CRC | RTC Control Register C | R/O | Bit specific |
| $0D_{16}$ | CRD | RTC Control Register D | R/O | $V_{PP}$ PUR |
| Programmable[1] | DOMA | Date of Month Alarm Register | R/W | $V_{PP}$ PUR |
| Programmable[1] | MONA | Month Alarm Register | R/W | $V_{PP}$ PUR |
| Programmable[1] | CEN | Century Register | R/W | $V_{PP}$ PUR |

1. Overlaid on RAM bytes in range $0E_{16}$-$7F_{16}$.

### Seconds Register (SEC)

Location: Index $00_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Seconds Data** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Seconds Data.** Values may be 00 to 59 in BCD format or 00 to 3B in binary format. |

## Host-Controlled Modules and Host Interface (Continued)

### Seconds Alarm Register (SECA)

Location: Index $01_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Seconds Alarm Data** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Seconds Alarm Data.** Values may be 00 to 59 in BCD format or 00 to 3B in binary format. When bits 7 and 6 are both set to one ('11'), unconditional match is selected. |

### Minutes Register (MIN)

Location: Index $02_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Minutes Data** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Minutes Data.** Values may be 00 to 59 in BCD format or 00 to 3B in binary format. |

### Minutes Alarm Register (MINA)

Location: Index $03_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Minutes Alarm Data** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Minutes Alarm Data.** Values may be 00 to 59 in BCD format or 00 to 3B in binary format. When bits 7 and 6 are both set to one ('11'), unconditional match is selected. |

### Hours Register (HOR)

Location: Index $04_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Hours Data** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Hours Data.** For 12-hour mode, values may be 01 to 12 (a.m.) and 81 to 92 (p.m.) in BCD format or 01 to 0C (a.m.) and 81 to 8C (p.m.) in binary format. For 24-hour mode, values may be 00 to 23 in BCD format or 00 to 17 in binary format. |

## Host-Controlled Modules and Host Interface (Continued)

### Hours Alarm Register (HORA)

Location: Index $05_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Hours Alarm Data** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Hours Alarm Data.** For 12-hour mode, values may be 01 to 12 (a.m.) and 81 to 92 (p.m.) in BCD format or 01 to 0C (a.m.) and 81 to 8C (p.m.) in binary format. For 24-hour mode, values may be 00 to 23 in BCD format or 00 to 17 in binary format.<br>When bits 7 and 6 are both set to one ('11'), unconditional match is selected. |

### Day Of Week Register (DOW)

Location: Index $06_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Day Of Week Data** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Day Of Week Data.** Values may be 01 to 07 in BCD format or 01 to 07 in binary format. |

### Date Of Month Register (DOM)

Location: Index $07_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Date Of Month Data** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Date Of Month Data.** Values may be 01 to 31 in BCD format or 01 to 1F in binary format. |

### Month Register (MON)

Location: Index $08_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Month Data** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Month Data.** Values may be 01 to 12 in BCD format or 01 to 0C in binary format. |

## Host-Controlled Modules and Host Interface (Continued)

### Year Register (YER)

Location: Index $09_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Year Data** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Year Data.** Values may be 00 to 99 in BCD format or 00 to 63 in binary format. |

### RTC Control Register A (CRA)

This register controls test selection, among other functions and cannot be written before reading bit 7 of the CRD register.

Location: Index $0A_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Update in Progress** | **Divider Chain Control 2-0** | | | **Periodic Interrupt Rate Select 3-0** | | | |
| Reset | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 3-0 | **Periodic Interrupt Rate Select.** These R/W bits select one of 15 output taps from the clock divider chain to control the rate of the periodic interrupt (see Table 57 on page 330 and Figure 107 on page 320). They are cleared to $000_2$ as long as bit 7 of CRD register is 0. |
| 6-4 | **Divider Chain Control.** These R/W bits control the configuration of the divider chain for timing generation (see Table 56). They are cleared to $010_2$ as long as bit 7 of CRD register reads 0. |
| 7 | **Update in Progress.** This RO bit is not affected by reset; it is 0 when bit 7 of CRB register is 1.<br>0: Timing registers not updated within 244 µs<br>1: Timing registers updated within 244 µs |

**Table 56. Divider Chain Control and Test Selection**

| DV2 | DV1 | DV0 | Configuration |
|---|---|---|---|
| CRA6 | CRA5 | CRA4 | |
| 0 | 0 | X | Oscillator Disabled |
| 0 | 1 | 0 | Normal Operation |
| 0 | 1 | 1 | Test |
| 1 | 0 | X | |
| 1 | 1 | X | Divider Chain Reset |

## Host-Controlled Modules and Host Interface (Continued)

**Table 57. Periodic Interrupt Rate Encoding**

| Rate Select 3 2 1 0 | Periodic Interrupt Rate (ms) | Divider Chain Output |
|---|---|---|
| 0 0 0 0 | No interrupts | - |
| 0 0 0 1 | 3.906250 | 7 |
| 0 0 1 0 | 7.812500 | 8 |
| 0 0 1 1 | 0.122070 | 2 |
| 0 1 0 0 | 0.244141 | 3 |
| 0 1 0 1 | 0.488281 | 4 |
| 0 1 1 0 | 0.976562 | 5 |
| 0 1 1 1 | 1.953125 | 6 |
| 1 0 0 0 | 3.906250 | 7 |
| 1 0 0 1 | 7.812500 | 8 |
| 1 0 1 0 | 15.625000 | 9 |
| 1 0 1 1 | 31.250000 | 10 |
| 1 1 0 0 | 62.500000 | 11 |
| 1 1 0 1 | 125.000000 | 12 |
| 1 1 1 0 | 250.000000 | 13 |
| 1 1 1 1 | 500.000000 | 14 |

### RTC Control Register B (CRB)

Location: Index $0B_{16}$

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Set Mode** | **Periodic Interrupt Enable** | **Alarm Interrupt Enable** | **Update Ended Interrupt Enable** | **Reserved** | **Data Mode** | **Hour Mode** | **Daylight Savings** |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

.

| Bit | Description |
|---|---|
| 0 | **Daylight Savings.** This bit is reset at $V_{PP}$ Power-Up reset only.<br>0: Disabled<br>1: Enabled<br>In the spring, time advances from 1:59:59 AM to 3:00:00 AM on the first Sunday in April.<br>In the fall, time returns from 1:59:59 AM to 1:00:00 AM on the last Sunday in October. |
| 1 | **Hour Mode.** This bit is reset at $V_{PP}$ Power-Up reset only.<br>0: 12-hour format enabled<br>1: 24-hour format enabled |
| 2 | **Data Mode.** This bit is reset at $V_{PP}$ Power-Up reset only.<br>0: BCD format enabled<br>1: Binary format enabled |

## Host-Controlled Modules and Host Interface (Continued)

| Bit | Description |
|---|---|
| 3 | **Reserved.** This bit is defined as “Square Wave Enable” by MC146818 and is not supported by the RTC. It is always read as 0. |
| 4 | **Update Ended Interrupt Enable.** This interrupt is generated when an update occurs. It is cleared to 0 on RTC reset (i.e., host domain reset).<br>0: Disabled<br>1: Enabled |
| 5 | **Alarm Interrupt Enable.** This interrupt is generated immediately after a time update in which the seconds, minutes, hours, date and month time equal their respective alarm counterparts. It is cleared to 0 as long as bit 7 of CRD register is 0.<br>0: Disabled<br>1: Enabled |
| 6 | **Periodic Interrupt Enable.** Bits 3-0 of CRA register determine the rate at which this interrupt is generated. It is cleared to 0 on RTC reset (i.e., host domain reset).<br>0: Disabled<br>1: Enabled |
| 7 | **Set Mode.** This bit is reset at $V_{PP}$ Power-Up reset only.<br>0: Timing updates occur normally<br>1: User copy of time is “frozen”, allowing the time registers to be accessed whether or not an update occurs |

### RTC Control Register C (CRC)

Location: Index $0C_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **IRQF** | **Periodic Interrupt Flag** | **Alarm Interrupt Flag** | **Update Ended Interrupt Flag** | **Reserved** | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 3-0 | **Reserved.** |
| 4 | **Update Ended Interrupt Flag.** This RO bit is cleared to 0 on RTC reset (i.e., host domain reset). In addition, this bit is cleared to 0 when this register is read.<br>0: No update occurred since the last read<br>1: Time registers update |
| 5 | **Alarm Interrupt Flag.** This RO bit is cleared to 0 as long as bit 7 of CRD register is 0. In addition, this bit is cleared to 0 when this register is read.<br>0: No alarm detected since the last read<br>1: Alarm condition detected |
| 6 | **Periodic Interrupt Flag.** This RO bit is cleared to 0 on RTC reset (i.e., host domain reset). In addition, this bit is cleared to 0 when this register is read.<br>0: No transition occurred on the selected tap since the last read<br>1: Transition occurred on the selected tap of the divider chain |
| 7 | **IRQF (IRQ Flag).** This RO bit mirrors the value of the interrupt output signal. When interrupt is active, IRQF is 1. To clear this bit (and deactivate the interrupt), read CRC register; this clears flags UF, AF and PF, which results in IRQF being cleared, as well.<br>0: IRQ inactive<br>1: Logic equation is true: ((UIE and UF) or (AIE and AF) or (PIE and PF)) |

# Host-Controlled Modules and Host Interface (Continued)

## RTC Control Register D (CRD)

Location: Index $0D_{16}$

Type: RO

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Valid RAM and Time** | **Reserved** | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 6-0 | **Reserved.** |
| 7 | **Valid RAM and Time.** This bit senses the voltage that feeds the RTC ($V_{CC}$ or $V_{BAT}$) and indicates whether or not it was too low since the last time this bit was read. If it was too low (i.e., < $V_{LOWBAT}$), the RTC contents (time/calendar registers and CMOS RAM) are not valid. This bit is set after the CRD is read.<br>0: The voltage that feeds the RTC was too low<br>1: RTC contents (time/calendar registers and CMOS RAM) are valid |

## Date of Month Alarm Register (DOMA)

Location: Programmable Index

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Date of Month Alarm Data** | | | | | | | |
| Reset | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Date of Month Alarm Data.** Values may be 01 to 31 in BCD format or 01 to 1F in binary format.<br>When bits 7 and 6 are both set to one ('11'), unconditional match is selected (default). |

## Month Alarm Register (MONA)

Location: Programmable Index

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Month Alarm Data** | | | | | | | |
| Reset | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Month Alarm Data.** Values may be 01 to 12 in BCD format or 01 to 0C in binary format.<br>When bits 7 and 6 are both set to one ('11'), unconditional match is selected (default). |

### Century Register (CEN)

Location: Programmable Index

Type: R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | **Century Data** | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| 7-0 | **Century Data.** Values may be 00 to 99 in BCD format or 00 to 63 in binary format. |

## 6.2.16 BCD and Binary Formats

| Parameter | BCD Format | Binary Format |
|---|---|---|
| Seconds | 00 to 59 | 00 to 3B |
| Minutes | 00 to 59 | 00 to 3B |
| Hours | 12-hour mode:01 to 12 (a.m.); 81 to 92 (p.m.)<br>24-hour mode:00 to 23 | 12-hour mode:01 to 0C (a.m.); 81 to 8C (p.m.)<br>24-hour mode:00 to 17 |
| Day | 01 to 07 (Sunday = 01) | 01 to 07 |
| Date | 01 to 31 | 01 to 1F |
| Month | 01 to 12 (January = 01) | 01 to 0C |
| Year | 00 to 99 | 00 to 63 |
| Century | 00 to 99 | 00 to 63 |

## 6.2.17 Usage Hints

1. Read bit 7 of CRD register at each system power-up to validate the contents of the RTC registers and the CMOS RAM. When this bit is 0, the contents of these registers and the CMOS RAM are questionable. This bit is reset when the backup battery voltage is too low. The voltage level at which this bit is reset is below the minimum recommended battery voltage, 2.4V. Although the RTC oscillator may function properly and the register contents may be correct at lower than 2.4V, this bit is reset since correct functionality cannot be guaranteed. System BIOS may use a checksum method to revalidate the contents of the CMOS-RAM. The checksum byte should be stored in the same CMOS RAM.
2. Change the backup battery while normal operating power is present, and not in backup mode, to maintain valid time and register information. If a low leakage capacitor is connected to $V_{BAT}$, the battery may be changed in backup mode.
3. A rechargeable NiCd battery may be used instead of a non-rechargeable Lithium battery. This is the preferred solution for portable systems, where small components are essential.
4. A supercap capacitor may be used instead of the normal Lithium battery. In a portable system, typically, the $V_{SB}$ voltage is always present because the power management stops the system before its voltage falls to low. A supercap capacitor in the range of 0.047F-0.47F should be able to supply the power during a battery replacement.

## 6.2.18 RTC General-Purpose RAM Map

**Table 58. Standard RAM Map**

| Index | Description |
|---|---|
| $0E_{16}$ - $7F_{16}$[1] | Battery-backed general-purpose 111-byte RAM. |

1. Battery-backed 111-byte RAM (114 – 3 overlaid registers).

**Table 59. Extended RAM Map**

| Index | Description |
|---|---|
| $00_{16}$ - $7F_{16}$ | Battery-backed general-purpose 128-byte RAM. |

# 7.0 Device Specifications

This section provides the power and grounding guidelines for the PC87591L-N05, specifies the device's maximum ratings and electrical characteristics and describes its timing.

## 7.1 GENERAL DC ELECTRICAL CHARACTERISTICS

### 7.1.1 Recommended Operating Conditions

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $V_{DD}$ | Host Domain Supply Voltage | 3.0 | 3.3 | 3.6 | V |
| $V_{CC}$ | Core Domain Supply Voltage | 3.0 | 3.3 | 3.6 | V |
| $AV_{CC}$ | Analog Supply Voltage | 3.15 | 3.3 | 3.45 | V |
| $V_{OFF}$ | $V_{DD}$, $V_{CC}$ and $AV_{CC}$ Power Off Voltage | –0.3 | 0 | +0.5 | V |
| $V_{BAT}$ | Battery Backup Supply Voltage | 2.4 | | 3.6 | V |
| $T_A$ | Operating Temperature | 0 | | +70 | °C |

### 7.1.2 Absolute Maximum Ratings

If military- or aerospace-specified devices are required, contact a National Semiconductor sales office or distributor for availability and specifications.

Storage Temperature: –65°C to +150°C

Temperature Under Bias: 0°C to +70°C

Absolute maximum ratings are values beyond which damage to the device may occur. Unless otherwise specified, all voltages are relative to ground.

| Symbol | Parameter | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{DD}$ | Host Domain Supply Voltage | | –0.5 | +4.2 | V |
| $V_{CC}$ | Core Domain Supply Voltage | | –0.5 | +4.2 | V |
| $AV_{CC}$ | Analog Supply Voltage | | –0.5 | +4.2 | V |
| $V_{BAT}$ | Battery Backup Supply Voltage | | –0.5 | +4.2 | V |
| $V_I$ | Input Voltage | All other buffer types | –0.5 | 5.5 | V |
| | | Buffer types: $IN_{AC}$[1], $IN_{AD}$, $IN_{OSC}$, $IN_{PCI}$, $IN_{TS}$ (IOPE0-3, IOPE6-7) | –0.5 | $V_{SUP}$[2] + 0.5 | V |
| $V_O$ | Output Voltage | All other buffer types | –0.5 | 5.5 | V |
| | | Buffer types: $O_{DA}$, $O_{OSC}$, $O_{PCI}$ | –0.5 | $V_{SUP}$[3] + 0.5 | V |
| $T_{STG}$ | Storage Temperature | | –65 | +165 | °C |
| $P_D$ | Power Dissipation | | | 1 | W |
| $T_L$ | Lead Temperature Soldering (10 s) | | | +260 | °C |
| | ESD Tolerance | $C_{ZAP}$ = 100 pF<br>$R_{ZAP}$ = 1.5 KΩ[4] | 2000 | | V |

1. When ACM is enabled.
2. $V_{SUP}$ is $V_{DD}$, $V_{CC}$, $AV_{CC}$ or $V_{BAT}$, according to the power well of the input.
3. $V_{SUP}$ is $V_{DD}$, $V_{CC}$, $AV_{CC}$ or $V_{BAT}$, according to the power well of the output.
4. Value based on test complying with RAI-5-048-RA human body model ESD testing.

## 7.0 Device Specifications (Continued)

### 7.1.3 Capacitance

| Symbol | Parameter | Min[2] | Typ[1] | Max[2] | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Pin Capacitance | | 5 | 7 | pF |
| $C_{IN1}$ | Clock Input Capacitance | 5 | 8 | 12 | pF |
| $C_{IO}$ | I/O Pin Capacitance | | 10 | 12 | pF |
| $C_O$ | Output Pin Capacitance | | 6 | 8 | pF |

1. $T_A$ = 25°C; f = 1 MHz.
2. Not tested; guaranteed by characterization.

### 7.1.4 Power Supply Current Consumption under Recommended Operating Conditions

| Symbol | Parameter | Conditions[1] | Typ | Max | Unit |
|---|---|---|---|---|---|
| $I_{DD}$ | $V_{DD}$ Average Main Supply Current | $V_{IL}$ = 0.5V, $V_{IH}$ = 2.4V<br>No Load | 3 | 4.5 | mA |
| $I_{DDLP}$ | $V_{DD}$ Quiescent Main Supply Current in Low Power Mode | $V_{IL}$ = GND, $V_{IH}$ = $V_{DD}$<br>No Load | 30 | 50 | µA |
| $I_{CC}$ | $V_{CC}$ Active Supply Current | $t_{CLK}$ = 250 ns | 15 | | mA |
| | | $t_{CLK}$ = 50 ns | 23 | 37 | mA |
| $I_{CCW}$ | $V_{CC}$ Active Executing WAIT Supply Current | $t_{CLK}$ = 250 ns | 6.6 | | mA |
| | | $t_{CLK}$ = 50 ns | 10 | | mA |
| $I_{CCI}$ | $V_{CC}$ Idle Mode Supply Current | Idle Mode<br>$V_{IL}$ = GND, $V_{IH}$ = $V_{CC}$<br>No Load | | 15 | µA |
| $I_{BAT}$ | $V_{BAT}$ Battery Supply Current | Power Off Mode | 0.9 | 1.5 | µA |

1. All parameters specified for 0°C ≤ $T_A$ ≤ 70°C; $V_{DD}$ and $V_{CC}$ = 3.3V ±10% unless otherwise specified.

### 7.1.5 Voltage Thresholds

| Symbol | Parameter | Conditions[1] | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{CCON}$ | $V_{CC}$ Detected as Power-On | | 2.5 | | 2.95 | V |
| $V_{CC2PP}$ | $V_{CC}$ detected for use as source for $V_{PP}$ | | 2.3 | | 2.4 | V |
| $V_{PP2CC}$ | $V_{CC}$ detected inactive for use of $V_{BAT}$ as source for $V_{PP}$ | | 1.9 | | 2.2 | V |
| $V_{DDON}$ | $V_{DD}$ Detected as on | | 2.5 | | 2.95 | V |
| $V_{BATDTC}$ | Battery Detected[2] | | 1.0 | | 1.2 | V |
| $V_{LOWBAT}$ | Low Battery Voltage | | 1.3 | | 1.9 | V |
| $V_{BATMIN}$ | Workable Battery Voltage | | 2.4 | | | V |
| $V_{BATMAX}$ | Battery Input Voltage | | | | 3.6 | V |

1. All parameters specified for 0°C ≤ $T_A$ ≤ 70°C.
2. Not tested; guaranteed by characterization.

# 7.0 Device Specifications (Continued)

## 7.2 DC CHARACTERISTICS OF PINS BY I/O BUFFER TYPES

The following tables summarize the DC characteristics of all device pins described in Chapter 2 on page 36. The characteristics describe the general I/O buffer types defined in Table 2 on page 38. For exceptions, see Section 7.2.9 on page 338. For the DC characteristics of the analog pins, see Section 7.4 on page 340. The DC characteristics of the system interface meet the PCI 2.1 3.3V DC signaling.

### 7.2.1 Input, CMOS Compatible with Schmitt Trigger

**Symbol:** $IN_{CS}$

| Symbol | Parameter | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{IH}$ | Input High Voltage | | 0.75 $V_{SUP}$[1] | 5.5[2] | V |
| $V_{IL}$ | Input Low Voltage | | –0.5[2] | 1.1 | V |
| $V_H$ | Input Hysteresis | | 500[3] | | mV |
| $I_{IL}$ | Input Leakage Current | $0 < V_{IN} < V_{SUP}$ | | ±1[4] | µA |

1. $V_{SUP}$ is $V_{DD}$ or $V_{CC}$, according to the power well of the input.
2. Not tested; guaranteed by design.
3. Not tested; guaranteed by characterization.
4. Maximum 10 µA for all pins together.

### 7.2.2 Input, PCI 3.3V

**Symbol:** $IN_{PCI}$

| Symbol | Parameter | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{IH}$ | Input High Voltage | | 0.5 $V_{DD}$ | $V_{DD}$ + 0.5[1] | V |
| $V_{IL}$ | Input Low Voltage | | –0.5[1] | 0.3 $V_{DD}$ | V |
| $I_{IL}$[2] | Input Leakage Current | $0 < V_{IN} < V_{DD}$ | | ±1[3] | µA |

1. Not tested; guaranteed by design.
2. Input leakage current includes the output leakage of the bidirectional buffers with TRI-STATE outputs.
3. Maximum 10 µA for all pins together.

### 7.2.3 Input, SMBus Compatible

**Symbol:** $IN_{SM}$

| Symbol | Parameter | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{IH}$ | Input High Voltage | | 1.4 | 5.5[1] | V |
| $V_{IL}$ | Input Low Voltage | | –0.5 [1] | 0.8 | V |
| $I_{IL}$[2] | Input Leakage Current | $0 < V_{IN} < V_{DD}$ | | ±1[3] | µA |

1. Not tested; guaranteed by design.
2. Input leakage current includes the output leakage of the bidirectional buffers with TRI-STATE outputs.
3. Maximum 10 µA for all pins together.

## 7.0 Device Specifications (Continued)

### 7.2.4 Input, TTL Compatible

**Symbol:** $IN_T$

| Symbol | Parameter | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{IH}$ | Input High Voltage | | 2.0 | 5.5[1] | V |
| $V_{IL}$ | Input Low Voltage | | –0.5[1] | 0.8 | V |
| $I_{IL}$[2] | Input Leakage Current | $0 < V_{IN} < V_{CC}$ | | ±1[3] | µA |

1. Not tested; guaranteed by design.
2. Input leakage current includes the output leakage of the bidirectional buffers with TRI-STATE outputs.
3. Maximum 10 µA for all pins together.

### 7.2.5 Input, TTL Compatible with Schmitt Trigger

**Symbol:** $IN_{TS}$

| Symbol | Parameter | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{IH}$ | Input High Voltage, 5V tolerant pins[1] | | 2.0 | 5.5[2] | V |
| | Input High Voltage, pins without 5V tolerance[1] | | 2.0 | $V_{SUP}$[3]+0.5[2] | V |
| $V_{IL}$ | Input Low Voltage | | –0.5[2] | 0.8 | V |
| $V_H$ | Input Hysteresis | | 250[4] | | mV |
| $I_{IL}$ | Input Leakage Current | $0 < V_{IN} < V_{SUP}$ | | ±1[5] | µA |

1. See Section 7.1.2 on page 334.
2. Not tested; guaranteed by design.
3. $V_{SUP}$ is $V_{DD}$ or $V_{CC}$, according to the power well of the input.
4. Not tested; guaranteed by characterization.
5. Maximum 10 µA for all pins together.

### 7.2.6 Output, TTL Compatible Push-Pull Buffer

**Symbol:** $O_{p/n}$

Output, TTL-compatible, rail-to-rail push-pull buffer that is capable of sourcing *p* mA and sinking *n* mA

| Symbol | Parameter | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{OH}$ | Output High Voltage | $I_{OH} = -p$ mA | 2.4 | | V |
| | | $I_{OH}$ = –50 µA | $V_{SUP}$[1] – 0.2 | | V |
| $V_{OL}$ | Output Low Voltage | $I_{OL} = n$ mA | | 0.4 | V |
| | | $I_{OL}$ = 50 µA | | 0.2 | V |

1. $V_{SUP}$ is $V_{DD}$ or $V_{CC}$, according to the power well of the input.

# 7.0 Device Specifications (Continued)

## 7.2.7 Output, Open-Drain Buffer

**Symbol:** $OD_n$

Output, TTL-compatible open-drain output buffer, capable of sinking *n* mA. Output from these signals is open-drain and is never forced high.

| Symbol | Parameter | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{OL}$ | Output Low Voltage | $I_{OL}$ = *n* mA | | 0.4 | V |
| | | $I_{OL}$ = 50 µA | | 0.2 | V |

## 7.2.8 Output, PCI 3.3V

**Symbol:** $O_{PCI\ I}$

| Symbol | Parameter | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{OH}$ | Output High Voltage | $I_{out}$ = –500 µA | 0.9 $V_{DD}$ | | V |
| $V_{OL}$ | Output Low Voltage | $I_{out}$ = 1500 µA | | 0.1 $V_{DD}$ | V |

## 7.2.9 Exceptions

1. All pins are back-drive protected, except for output pins with PCI buffer type ($IN_{PCI}$ or $O_{PCI}$), oscillator ($O_{OSC}$) and all analog type pins ($O_{DA}$ and $O_{DI}$).
2. The following pins have an internal static pull-up resistor and therefore may have leakage current to $V_{CC}$ (when $V_{IN}$ = 0): SCL1-4, SDA1-4, IOPA7-0, IOPB7-0, IOPC7-0, IOPD7-0, IOPE7,6,4, IOPF7-0, IOPQ2-0, KBSIN0-7, PSCLK1-4, PSDAT1-4.
3. The following strap pins have an internal static pull-down resistor enabled during Power-Up reset and therefore may have leakage current to $V_{SS}$ (when $V_{IN} = V_{SUP}$): BADDR1-0, ENV1-0, SHBM, TRIS.
4. $I_{OH}$ is valid for a GPIO pin only when it is not configured as open-drain.

## 7.2.10 Terminology

**Back-Drive Protection.** A pin that is back-drive protected does not sink current into the supply when an input voltage higher than the supply, but below the pin's maximum input voltage, is applied to the pin. This is true even when the supply is inactive. Note that active pull-up resistors and active output buffers are typically not back-drive protected.

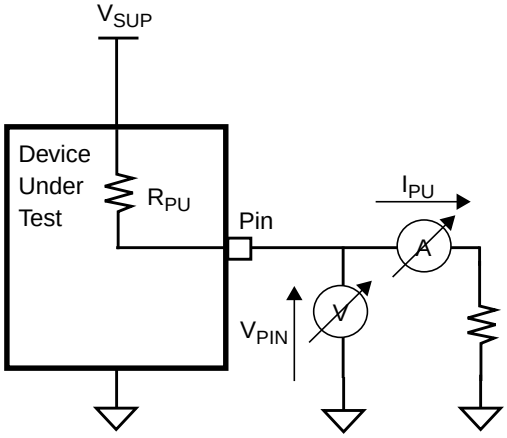
**5-Volt Tolerance.** An input signal that is 5V tolerant can operate with input voltage of up to 5V even though the supply to the device is only 3.3V. The actual maximum input voltage allowed to be supplied to the pin is indicated by the maximum high voltage allowed for the input buffer. Note that some pins have multiple buffers, not all of which are 5V tolerant. In such cases, there is a note that indicates at what conditions a 5V input may be applied to the pin; if there is no note, the low maximum voltage among the buffers is the maximum voltage allowed for the pin.

## 7.0 Device Specifications (Continued)

### 7.3 INTERNAL RESISTORS

**DC Test Conditions**

**Pull-Up Resistor Test Circuit**




**Pull-Down Resistor Test Circuit**

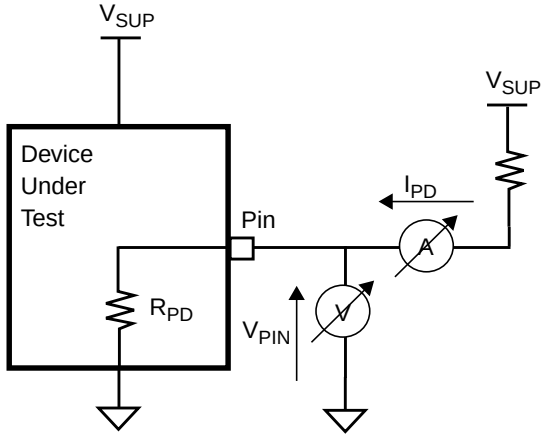



**Figure 115. Internal Resistor Test Conditions, $T_A$ = 0°C to 70°C, $V_{SUP}$ = 3.3V**

**Notes**:

1. The equivalent resistance of the pull-up resistor is calculated by $R_{PU} = (V_{SUP} - V_{PIN}) / I_{PU}$.
2. The equivalent resistance of the pull-down resistor is calculated by $R_{PD} = V_{PIN} / I_{PD}$.

#### 7.3.1 Pull-Up Resistor

**Symbol:** $PU_{nn}$

| Symbol | Parameter | Conditions | Typical | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $R_{PU}$[1] | Pull-up equivalent resistance | $V_{PIN}$ = 0V | *nn* | *nn* – 34% | *nn* + 47% | KΩ |
| | | $V_{PIN}$ = 0.17 $V_{DD}$ | | *nn* – 44% | | KΩ |
| | | $V_{PIN}$ = 0.8 $V_{DD}$ | | | *nn* – 41% | KΩ |

1. Not tested; guaranteed by characterization.

#### 7.3.2 Pull-Down Resistor

**Symbol:** $PD_{nn}$

| Symbol | Parameter | Conditions | Typical | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $R_{PD}$[1] | Pull-down equivalent resistance | $V_{PIN}$ = $V_{DD}$ | *nn* | *nn* – 35% | *nn* + 60% | KΩ |
| | | $V_{PIN}$ = 0.17 $V_{DD}$ | | | *nn* – 37% | KΩ |
| | | $V_{PIN}$ = 0.8 $V_{DD}$ | | *nn* – 48% | | KΩ |

1. Not tested; guaranteed by characterization.

# 7.0 Device Specifications (Continued)

## 7.4 ANALOG CHARACTERISTICS

### 7.4.1 ADC Characteristics

**Voltage Measurement**

| Parameter | Symbol | Conditions[1] | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| Resolution | RES | | | 8 | | Bit |
| Offset Error[2] | $OER_L$[3] | AD0-9: 12 mV ≤ $V_{IN}$ ≤ 0.9 V | | ±0.75 | | LSB |
| | $OER_H$ | AD0-9: 40 mV ≤ $V_{IN}$ ≤ $AV_{CC}$ –0.8 | | ±0.75 | | LSB |
| | $OER_B$ | AD13: 0.9V ≤ $V_{IN}$ ≤ $AV_{CC}$ –0.2 | | ±1 | | LSB |
| | $OER_V$ | AD10-12: 0.1V ≤ $V_{IN}$ ≤ $V_{FS}$ | | ±1 | | LSB |
| Gain Error[4] | $GER_L$ | AD0-9: 12 mV ≤ $V_{IN}$ ≤ 0.9 V | | ±0.75 | | LSB |
| | $GER_H$ | AD0-9: 40 mV ≤ $V_{IN}$ ≤ $AV_{CC}$ –0.8 | | ±0.75 | | LSB |
| | $GER_B$ | AD13: 0.9V ≤ $V_{IN}$ ≤ $AV_{CC}$ –0.2 | | ±1 | | LSB |
| | $GER_V$ | AD10-12: 0.1V ≤ $V_{IN}$ ≤ $V_{FS}$ | | ±1 | | LSB |
| Integral Non-linearity Error[5] | $INL_L$ | AD0-9: 12 mV ≤ $V_{IN}$ ≤ 0.9 V | | ±0.75 | | LSB |
| | $INL_H$ | AD0-9: 40 mV ≤ $V_{IN}$ ≤ $AV_{CC}$ –0.8 | | ±0.75 | | LSB |
| | $INL_B$ | AD13: 0.9V ≤ $V_{IN}$ ≤ $AV_{CC}$ –0.2 | | ±1 | | LSB |
| | $INL_V$ | AD10-12: 0.1V ≤ $V_{IN}$ ≤ $V_{FS}$ | | ±1 | | LSB |
| Differential Non-linearity Error[6] | $DNL_L$ | AD0-9: 12 mV ≤ $V_{IN}$ ≤ 0.9 V | | ±0.35[7] | | LSB |
| | $DNL_H$ | AD0-9: 40 mV ≤ $V_{IN}$ ≤ $AV_{CC}$ –0.8 | | ±0.35[6] | | LSB |
| | $DNL_B$ | AD13: 0.9V ≤ $V_{IN}$ ≤ $AV_{CC}$ –0.2 | | ±0.35[6] | | LSB |
| | $DNL_V$ | AD10-12: 0.1V ≤ $V_{IN}$ ≤ $V_{FS}$ | | ±0.35[6] | | LSB |
| External Inputs Accuracy[8] | $EACU_L$ | AD0-9: 12 mV ≤ $V_{IN}$ ≤ 0.9 V | | | ±1.5 | LSB |
| | $EACU_H$ | AD0-9: 40 mV ≤ $V_{IN}$ ≤ $AV_{CC}$ –0.8 | | | ±1.5 | LSB |
| Internal Inputs Accuracy[7, 9] | $IACU_B$ | AD13: 0.9V ≤ $V_{IN}$ ≤ $AV_{CC}$ –0.2 | | | ±2 | LSB |
| | $IACU_V$ | AD10-12: 0.1V ≤ $V_{IN}$ ≤ $V_{FS}$ | | | ±2 | LSB |

## 7.0 Device Specifications (Continued)

| Parameter | Symbol | Conditions[1] | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| Full Scale Voltage | $V_{FSL}$ | AD0-9 Low Scale | | 1.000 | | V |
| | $V_{FSH}$ | AD0-9 High Scale | | 3.000 | | V |
| | $V_{FSB}$ | AD13 | | 4.000 | | V |
| | $V_{FSV}$ | AD10-12 | | 4.000 | | V |
| Input Voltage Range | $V_{IN}$ | Note[10] | 0 | | $AV_{CC}$ | V |
| Analog Input Leakage Current | $I_{AL}$ | AD0-9: $0 \le V_{IN} \le AV_{CC}$ | | | ±1 | µA |
| Analog Input Resistance | $R_{AIN}$ | AD0-9, AD13 | 4 | | | MΩ |
| Analog Input Capacitance | $C_{AIN}$ | | | 10 | 15 | pF |
| ADC Clock Frequency | $F_{CLK}$ | | | 0.5 | | MHz |
| ADC Enable Delay[11] | $t_{END}$ | | | | 100 | µs |
| Voltage Conversion Duration | $t_{VC}$ | | | | 8.2 | ms |

1. All parameters specified for $0°C \le T_A \le 70°C$ and $AV_{CC}$= 3.3V ± 5% unless otherwise specified.
2. The difference between 0V and the actual voltage value for code $000_{16}$.
3. $XXX_L$ = Inputs AD0-9, Low Scale; $XXX_H$ = Inputs AD0-9, High Scale; $XXX_V$ = Inputs AD10-12; $XXX_B$ = Input AD13.
4. The difference between: $^{255}/_{256} * V_{FS}$ and the actual voltage for code $FF_{16}$.
5. The maximum difference between the ideal (straight) conversion line and the actual conversion curve, not including the offset, gain and quantization (±0.5 LSB) errors.
6. The maximum difference between an ideal step size (1 LSB) and any actual step size.
7. No missing codes.
8. Total unadjusted error (includes the offset, gain, integral non-linearity and quantization (±0.5 LSB) errors).
9. The internal power supply inputs: $V_{DD}$, $V_{CC}$, $AV_{CC}$.
10. Input Voltage allowed in normal operation. Linear range is as defined for the different measurement modes.
11. Time from the moment when ADCEN=1 in ADCCNF register until the beginning of the “ADC cycle”.

### 7.4.2 ACM Characteristics

| Parameter | Symbol | Conditions[1] | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| Resolution | RES | | | 6 | | Bit |
| Differential Non-linearity Error[2] | DNL | $0 \le V_{IN} \le V_{CC}$ | | | ±0.5[3] | LSB |
| Accuracy (Total unadjusted error) | ACU | $0 \le V_{IN} \le V_{CC}$ | | | ±1.5 | LSB |
| Input Voltage Range | $V_{IN}$ | | 0 | | $V_{CC}$ | V |
| Analog Input Leakage Current | $I_{AL}$ | $0 \le V_{IN} \le V_{CC}$ | | | ±1 | µA |
| Analog Input Capacitance | $C_{AIN}$ | | | 10 | 15 | pF |

1. All parameters specified for $0°C \le T_A \le 70°C$ and $V_{CC}$= 3.3V ± 10% unless otherwise specified.
2. The maximum difference between an ideal step size (1 LSB) and any actual step size.
3. No missing codes.

## 7.0 Device Specifications (Continued)

### 7.4.3 DAC Characteristics

| Parameter | Symbol | Conditions[1] | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| Resolution | RES | | | 8 | | Bit |
| Offset Error[2] | OER | $AV_{CC}$ = 3.3V | | | ±1 | LSB |
| Gain Error[3] | GER | $AV_{CC}$ = 3.3V | | | ±1 | LSB |
| Integral Non-linearity Error[4] | INL | $AV_{CC}$ = 3.3V, $0 \le V_{OUT} \le AV_{CC}$ | | | ±0.5 | LSB |
| Differential Non-linearity Error[5] | DNL | $AV_{CC}$ = 3.3V, $0 \le V_{OUT} \le AV_{CC}$ | | | ±0.5[6] | LSB |
| Output Voltage Range | $V_{OUT}$ | | 0 | | $AV_{CC}$ | V |
| Analog Output Resistance | $R_S$ | $0 \le V_{OUT} \le AV_{CC}$ | 2 | 3 | 4 | KΩ |
| Analog Output Capacitance | $C_{AO}$ | | | 10 | 15 | pF |
| DAC Settling Time[7] | $T_{SET}$ | $C_L$ = 50 pF | | | 1 | µs |
| DAC Enable Delay[8] | $T_{END}$ | $C_L$ = 50 pF | | | 10 | µs |

1. All parameters specified for $0°C \le T_A \le 70°C$. $AV_{CC}$ = 3.3V ± 5%, unless otherwise specified.
2. The difference between 0V and the actual voltage value for code $00_{16}$.
3. The difference between $^{255}/_{256} * AV_{CC}$ and the actual voltage for code $FF_{16}$.
4. The maximum difference between the ideal (straight) conversion line and the actual conversion curve, not including the offset, gain and quantization (±0.5 LSB) errors.
5. The maximum difference between an ideal step size (1 LSB) and any actual step size.
6. Monotonic.
7. Time from the converter loading with data, to output voltage settling within an error of ±0.5 LSB.
8. Time from the moment when DACEN*n*=1 in DACCTRL register until the settling of the output voltage.

## 7.5 PACKAGE THERMAL INFORMATION

Thermal resistance (degrees C/W) $Theta_{JC}$ and $Theta_{JA}$ values for the PC87591L-N05 packages are as follows:

**Table 60. Theta (Θ) J Values**

| Package Type | $Theta_{JA}$@0 lfpm | $Theta_{JA}$@225 lfpm | $Theta_{JA}$@500 lfpm | $Theta_{JA}$@900 lfpm | $Theta_{JC}$ |
|---|---|---|---|---|---|
| 176-Pin LQFP | 39.8 | 32.3 | 29.5 | 26.2 | 5.1 |
| 176-Ball FBGA | 30.0 | 25.7 | 23.3 | 19.9 | 8.5 |

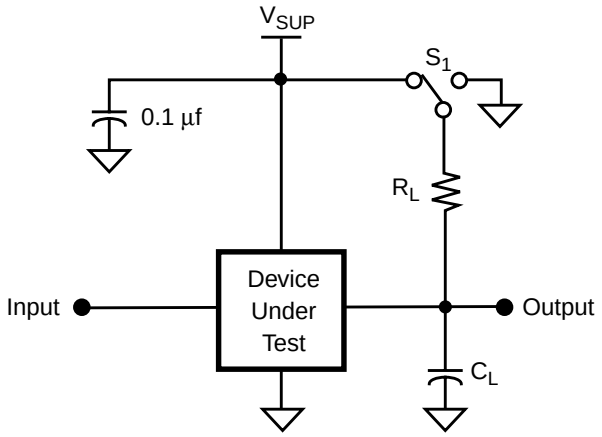
**Note:** Airflow for $Theta_{JA}$ values is measured in linear feet per minute (lfpm).

# 7.0 Device Specifications (Continued)

## 7.6 AC ELECTRICAL CHARACTERISTICS

### 7.6.1 AC Test Conditions

**Load Circuit** (Notes 1, 2, 3)




**AC Testing Input, Output Waveform**

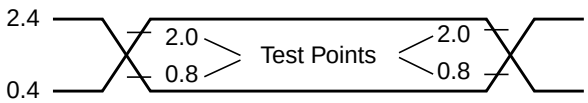



**Figure 116. AC Test Conditions, $T_A$ = 0°C to 70°C, $V_{SUP}$ = 3.3V ±10%**

**Notes**:

1. $V_{SUP}$ is $V_{DD}$, $V_{CC}$, $AV_{CC}$ or $V_{BAT}$, according to the power well of the pin.
2. $C_L$ = 50 pF for all output pins except the following pin groups (these values include both jig and oscilloscope capacitance):
   $C_L$ = 400 pF for ACCESS.bus pins.
3. $S_1$ = Open for push-pull output pins.
   $S_1$ = $V_{SUP}$ for high impedance to active low and active low to high impedance measurements.
   $S_1$ = $V_{SS}$ for high impedance to active high and active high to high impedance measurements.
   $R_L$ = 1.0 KΩ

The following abbreviations are used in Section 7.6:

RE = Rising Edge
FE = Falling Edge

### Definitions

The timing specifications in this section refer to low- or high-level voltage, according to the specific buffer type (TTL or CMOS) on the rising or falling edges of all the signals, as shown in the following figures, unless specifically stated otherwise.

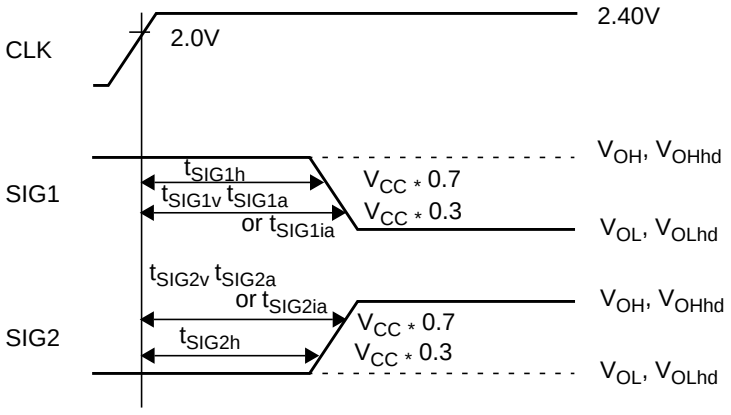



**Figure 117. CMOS Output Signals Specification Conventions**

## 7.0 Device Specifications (Continued)

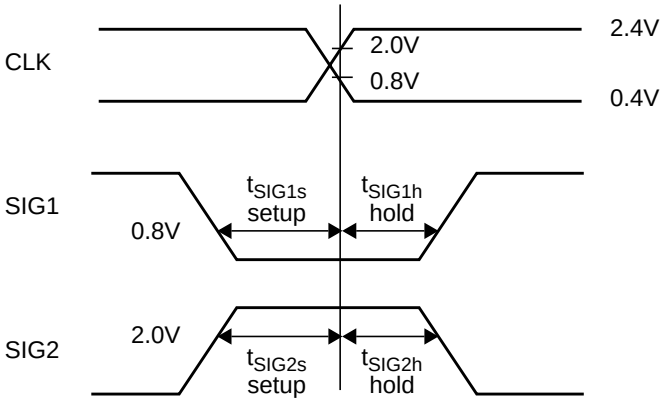



**Figure 118. TTL: Input Signal Specification Standard**

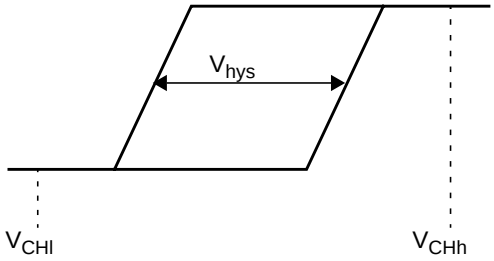



**Figure 119. CMOS with Hysteresis Inputs**

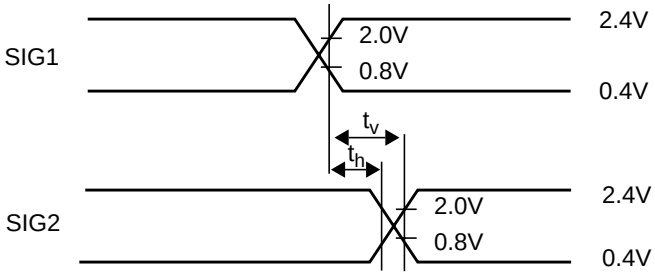



**Figure 120. Signal-to-Signal Delay**

## 7.0 Device Specifications (Continued)

### 7.6.2 Reset Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{SUPUP}$ | 121 | Supply wake-up time to 95% $V_{CC}$ | After $V_{CC} > V_{CCON}$ | | 1 | ms |
| $t_{IRST}$ | 121 | Internal or power-on reset time | Power stable to end of 16th clock cycle of CLK | $16 * t_{CLK} + t_{32KD}$ | $16 * t_{CLK} + t_{32KD} + t_{32KW}$ | - |
| $t_{IPLv}$ | 121 | Valid time: Internal strap pull-up resistors | Before end of 16th clock cycle of CLK | $16 * t_{CLK}$ | | - |
| $t_{EPLv}$ | 121 | Valid time: External strap pull-up resistors | Before end of 16th clock cycle of CLK | $6 * t_{CLK}$ | | - |
| $t_{WRST}$ | 122 | $\overline{\text{RESET1-2}}$ width | $\overline{\text{RESET1-2}}$ FE to $\overline{\text{RESET1-2}}$ RE | $3 * t_{CLK}$ | | - |

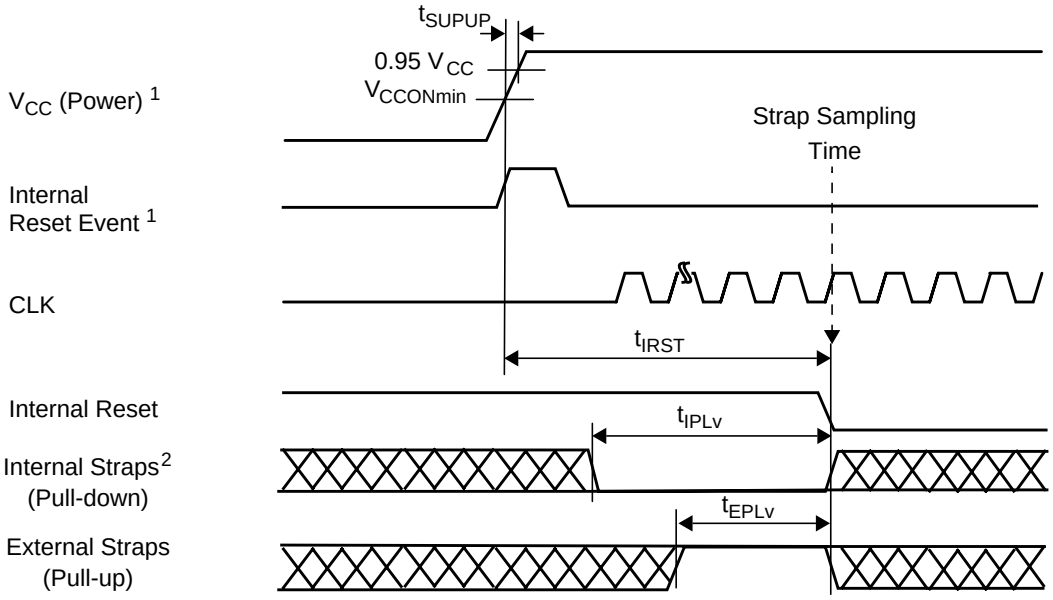



**Figure 121. Internal Power-Up Reset**

**Notes:**

1. Either Watchdog, Debugger I/F or Power-Up reset.
2. Valid on Power-Up reset only.

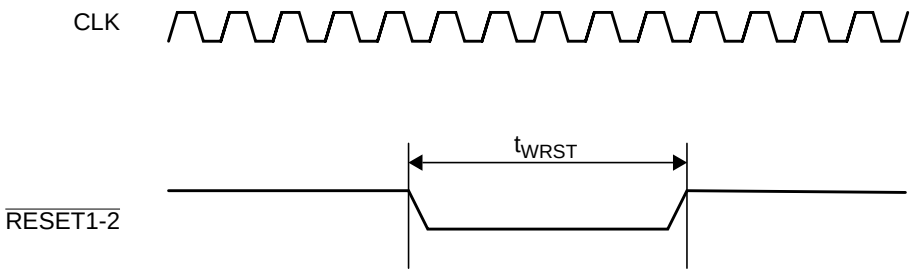



**Figure 122. Warm Hardware Reset**

## 7.0 Device Specifications (Continued)

### 7.6.3 Clock Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| Clock Input Timing | | | | | | |
| $t_{32KCLKIN}$ | 123 | Required clock period for 32KCLKIN | From RE to RE of 32KCLKIN $t_{32NOM}$ = 30.517578 | 30.5145 ($t_{32NOM}$ − 100ppm) | 30.5206 ($t_{32NOM}$ + 100ppm) | µs |
| Clock Output Timing | | | | | | |
| $t_{CLK}$ | 124 | CLK period | At 2.0V (both edges) | 50 | 250 | ns |
| $t_{CLKh}$ | 124 | CLK high time | At 2.0V (both edges) | 0.5 * $t_{CLK}$ − 5 ns | | - |
| $t_{CLKl}$ | 124 | CLK low time | At 0.8V (both edges) | 0.5 * $t_{CLK}$ − 5 ns | | - |
| $t_{CLKr}$ | 124 | CLK rise time | 0.8V to 2.0V | | 6 | ns |
| $t_{CLKf}$ | 124 | CLK fall time | 2.0V to 0.8V | | 6 | ns |
| $t_{CLKw}$ | 125 | CLK wake-up time | From wake-up event until CLK starts toggling | | 100 | µs |
| $t_{CLKINTst}$ | 125 | $t_{CLK}$ period | Active mode in steady state | 0.99 * $t_{CLKINTnom}$ [1] | 1.01 * $t_{CLKINTnom}$ | - |
| $t_{CLKINTwk}$ | 125 | $t_{CLK}$ period | After wake-up from Idle | 0.9 * $t_{CLKINTnom}$ | 1.1 * $t_{CLKINTnom}$ | - |
| $t_{CLKstab}$ | 125 | $t_{CLK}$ stabilization time | After wake-up from Idle | | 0.5 | s |
| $t_{32KW}$ | 123 | 32K oscillator wake-up time | After $V_{BAT} > V_{LOWBAT}$ | | 1 | s |
| $t_{32KD}$ | 124 | CLK delay time | After $V_{CC} > V_{CCON}$ | | 40 | ms |

1. $t_{CLKINTnom}$ is defined in Table 1 on page 214.

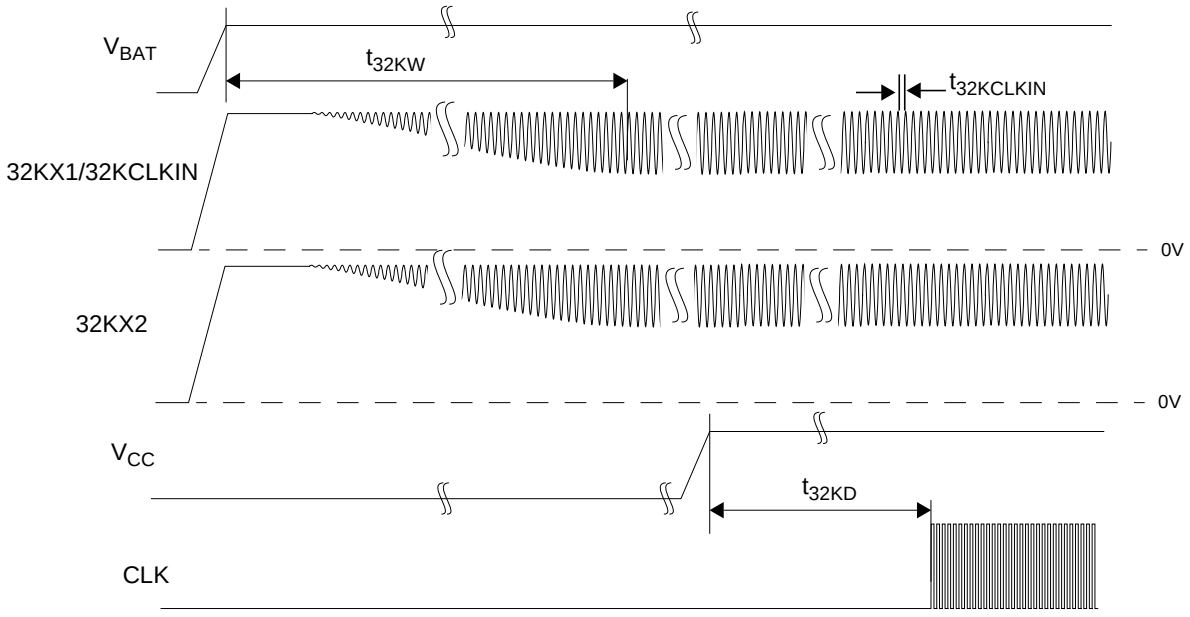



**Figure 123. 32K Waveforms**

## 7.0 Device Specifications (Continued)

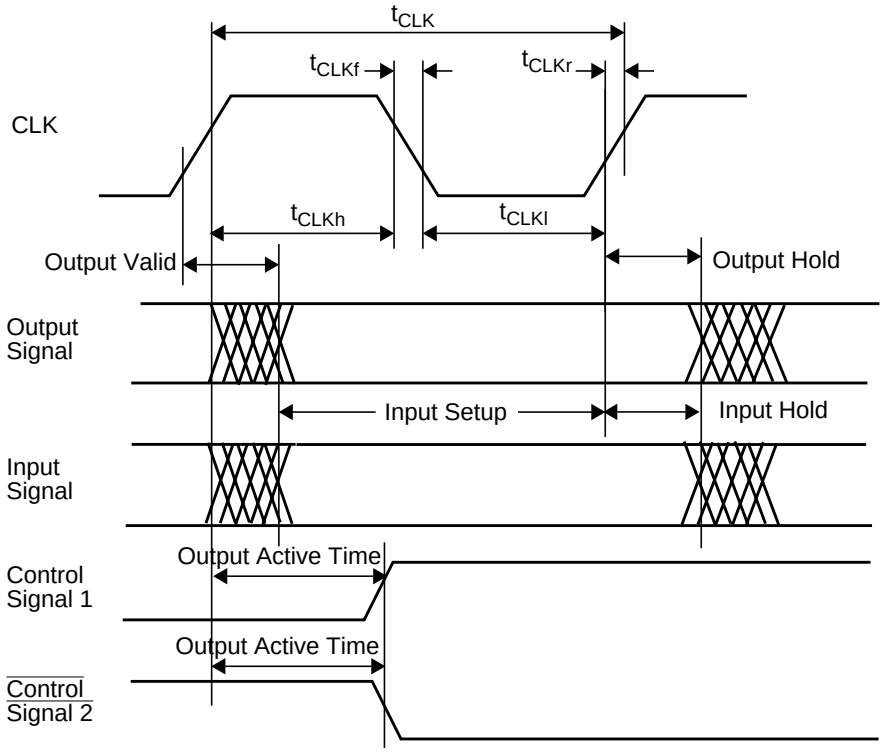



**Figure 124. Clock Waveforms**

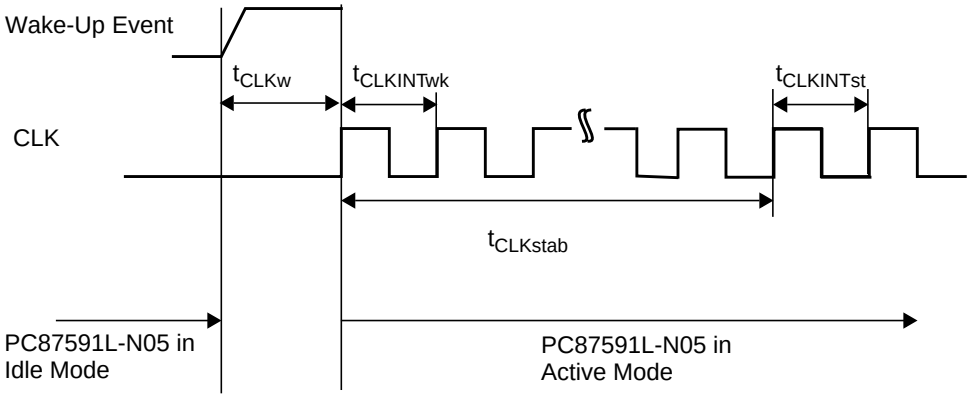



**Figure 125. Internal Clock Generator**

## 7.0 Device Specifications (Continued)

### 7.6.4 BIU Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| BIU Input Timing | | | | | | |
| $t_1$ | 126, 128 to 130 | Input setup time D0-15 | Before RE CLK | 15 | | ns |
| $t_2$ | 126, 128 to 130 | Input hold time D0-15 | After RE CLK | 0 | | ns |
| BIU Output Timing | | | | | | |
| $t_3$ | 126 to 131 | Output valid time A0-20, BE0,1,CBRD, D0-15 | After RE CLK | | 14 | ns |
| $t_4$ | 126 to 131 | Output valid time BST0-2 | After RE CLK | | $0.5 * t_{CLK}$ + 14 ns | - |
| $t_5$ | 126 to 130 | Output active/inactive time $\overline{RD}$, $\overline{SEL0\text{-}2}$, $\overline{SELIO}$ | After RE CLK | | 14 | ns |
| $t_6$ | 126, 127 | Output active/inactive time $\overline{WR0\text{-}1}$ | After RE CLK | | $0.5 * t_{CLK}$ + 14 ns | - |
| $t_7$ | 128 | Minimum inactive time $\overline{RD}$ | After RE $\overline{RD}$ | $t_{CLK}$ – 5 ns | | - |
| $t_8$ | 126, 127 | Output float time A0-20, D0-15, $\overline{RD}$, $\overline{SEL0\text{-}2}$, $\overline{SELIO}$,$\overline{WR0\text{-}1}$ | After RE CLK | | 14 | ns |
| $t_9$ | 126, 127 | Minimum delay time | From RE $\overline{RD}$ to D0-15 drive | $t_{CLK}$ – 8 ns | | - |
| $t_{10}$ | 127 | Minimum delay time | From RE $\overline{RD}$ to RE $\overline{SELn}$ | 0 | | ns |
| $t_{11}$ | 127 | Minimum delay time | From RE $\overline{SELx}$ to FE $\overline{SELy}$ | 0 | | ns |
| $t_{12}$ | 126 to 131 | Output hold time A0-20, BE0-1, CBRD, D0-15, $\overline{RD}$, $\overline{SEL0\text{-}2}$, $\overline{SELIO}$ | After RE CLK | 0 | | ns |
| $t_{13}$ | 126 to 131 | Output hold time BST0-2, $\overline{WR0\text{-}1}$ | After RE CLK | $0.5 * t_{CLK}$ – 4 ns | | - |
| $t_{14}$ | 127 | D0-15 valid in late write bus cycles | Before RE $\overline{WR0\text{-}1}$ | $(K + 0.5) * t_{CLK}$ – 8 ns | | - |

## 7.0 Device Specifications (Continued)

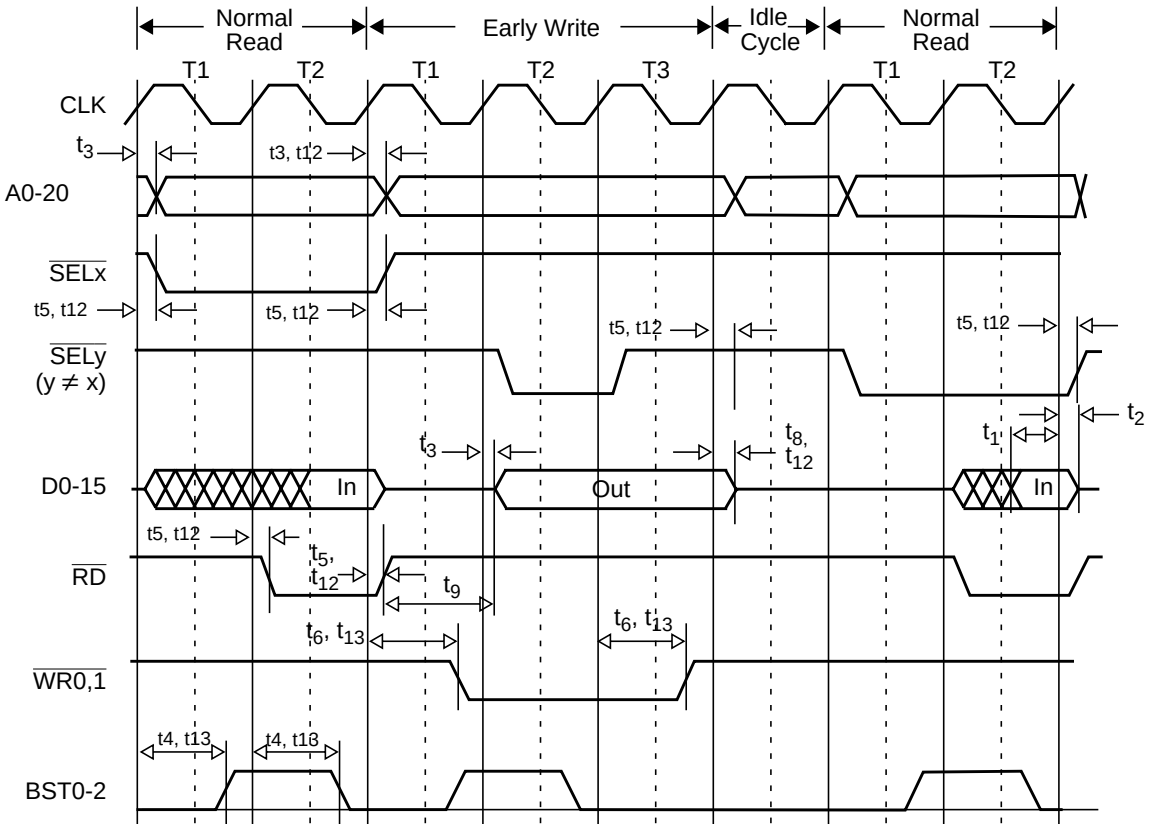



**Figure 126. Early Write Between Normal Read Bus Cycles**

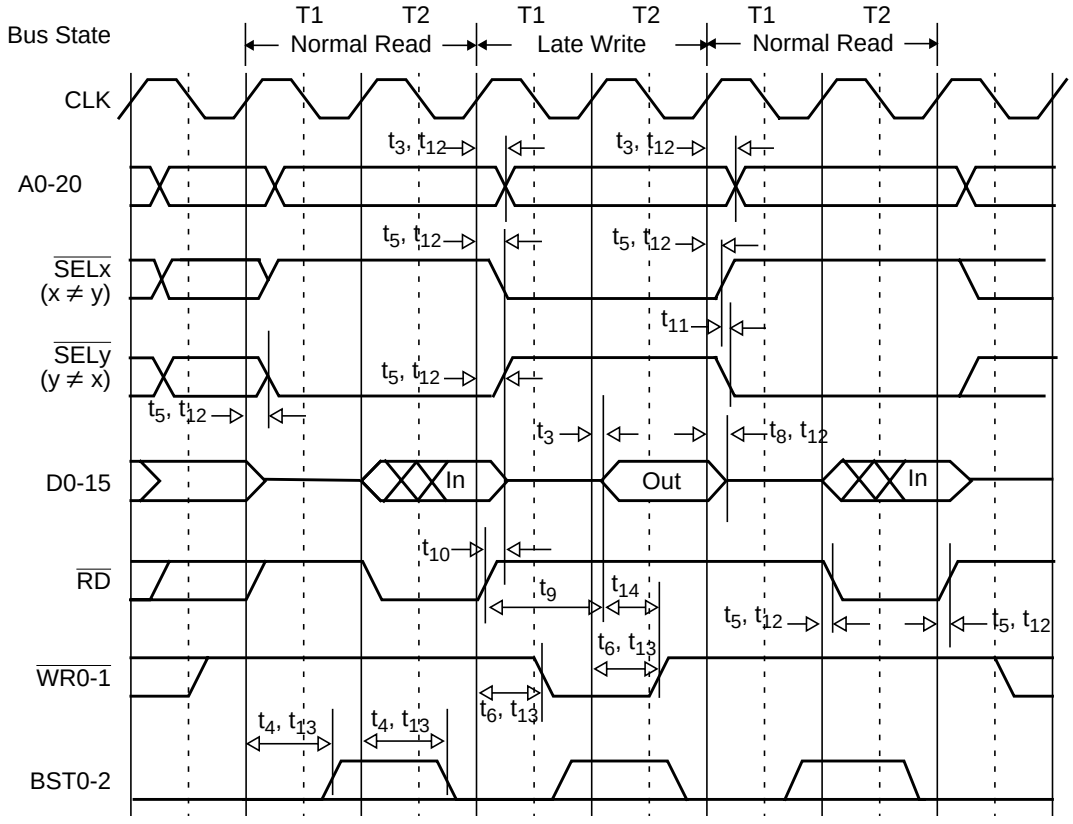



**Figure 127. Late Write between Two Normal Read Bus Cycles, 0 Wait, AC Timing**

## 7.0 Device Specifications (Continued)

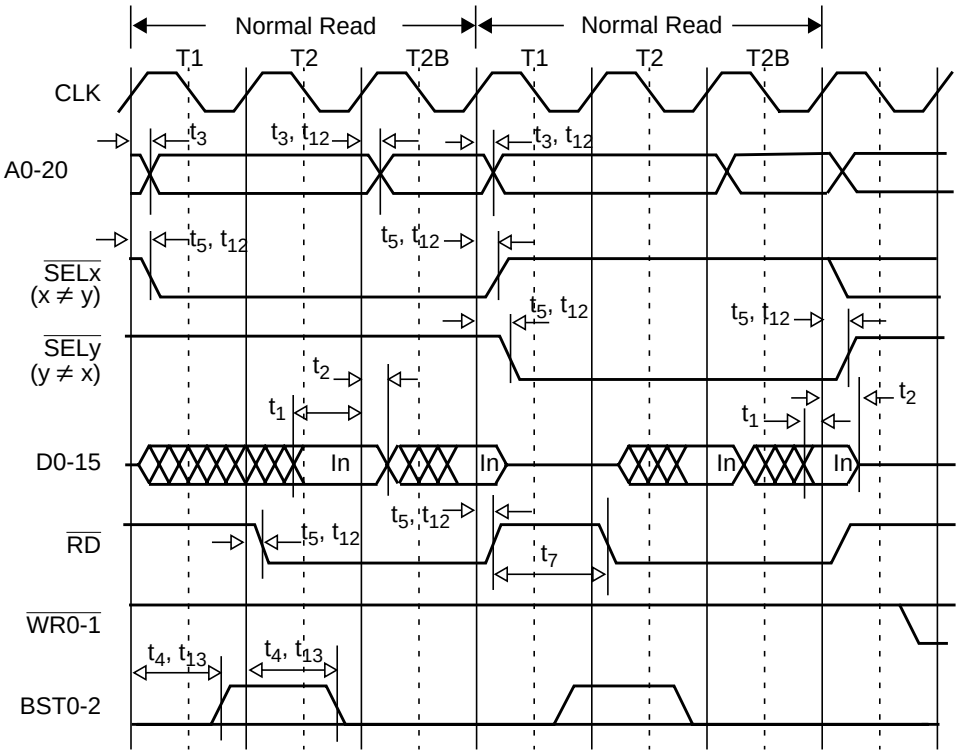



**Figure 128. Two Consecutive Normal Read Bus Cycles with Burst, 0 Wait, AC Timing**

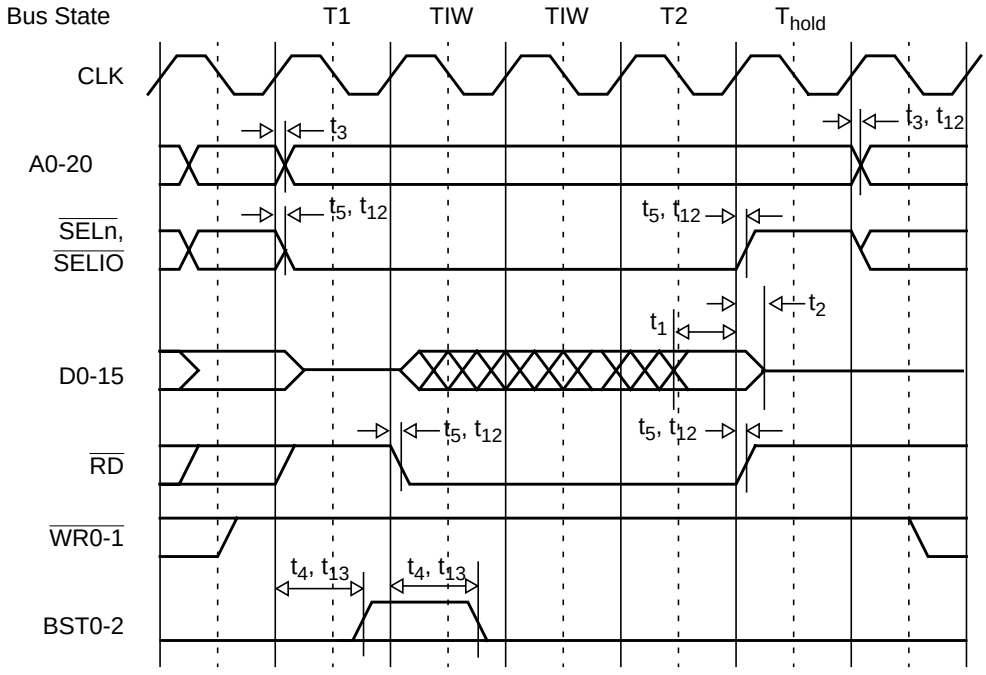



**Figure 129. Normal Read Bus Cycle (2 Internal Waits, and 1 Hold), AC Timing**

## 7.0 Device Specifications (Continued)

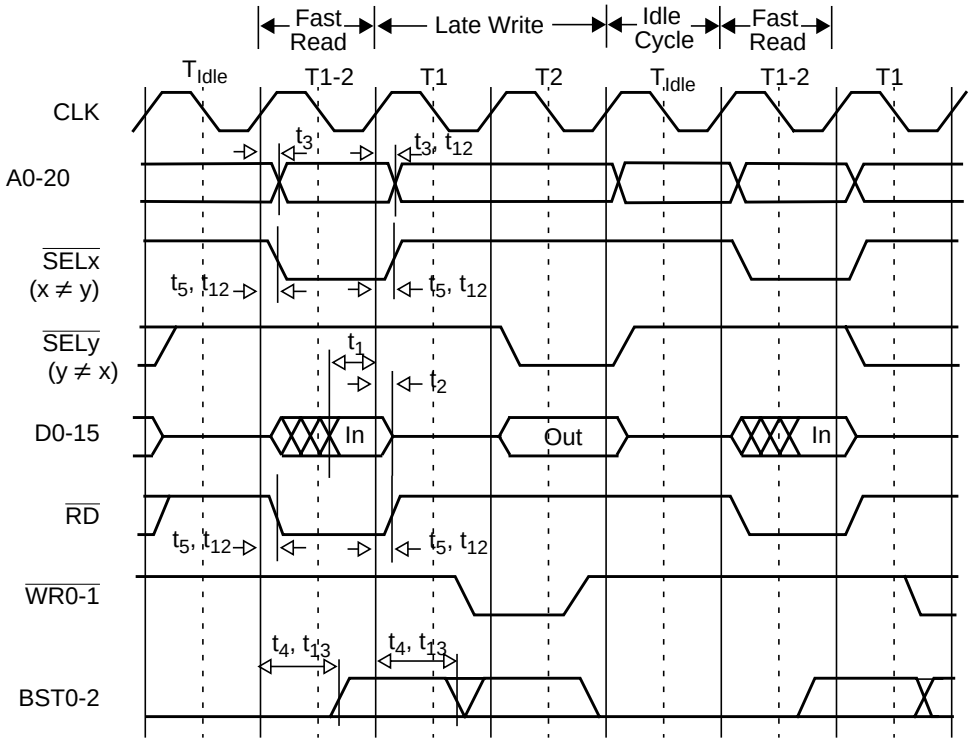



**Figure 130. Fast Read Bus Cycle, AC Timing**

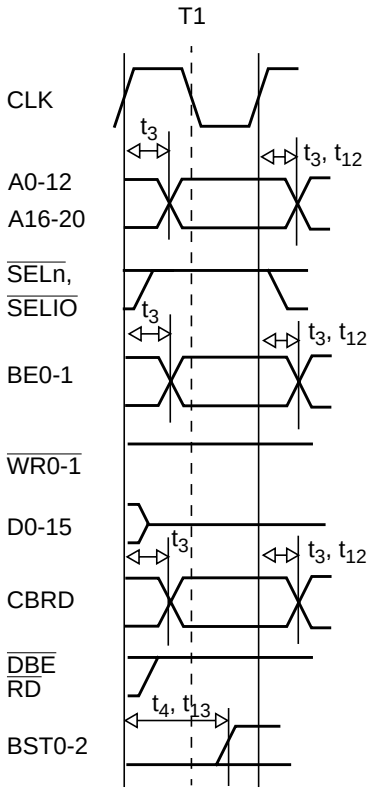



**Figure 131. Core Bus Monitoring Bus Cycle, AC Timing**

# 7.0 Device Specifications (Continued)

## 7.6.5 GPIO Port Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| GPIO Input Timing | | | | | | |
| $t_{INPs}$ | 132 | Input setup time IOPA0-7, IOPB0-7, IOPC0-7, IOPD0-7, IOPE0-7, IOPF0-7, IOPJ2-7, IOPL3-4, IOPM0-7, IOPQ0-2, KBSIN0-7[1] | Before RE CLK | $0.5 * t_{CLK}$ | | - |
| $t_{INPh}$ | 132 | Input hold time IOPA0-7, IOPB0-7, IOPC0-7, IOPD0-7, IOPE0-7, IOPF0-7, IOPJ2-7, IOPL3-4, IOPM0-7, IOPQ0-2, KBSIN0-7[1] | After RE CLK | 0 | | ns |
| GPIO Output Timing | | | | | | |
| $t_{OUTv}$ | 133 | Output valid time KBSOUT0-15, IOPA0-7, IOPB0-7, IOPC0-7, IOPD0-7, IOPF0-7, IOPJ2-7, IOPL3-4, IOPM0-7, IOPQ0-3 | After RE CLK | | $0.5 * t_{CLK}$ | - |
| $t_{OUTh}$ | 133 | Output hold time KBSOUT0-15, IOPA0-7, IOPB0-7, IOPC0-7, IOPD0-7, IOPF0-7, IOPJ2-7, IOPL3-4, IOPM0-7, IOPQ0-3 | After RE CLK | 0 | | ns |

1. When using the Schmitt input.

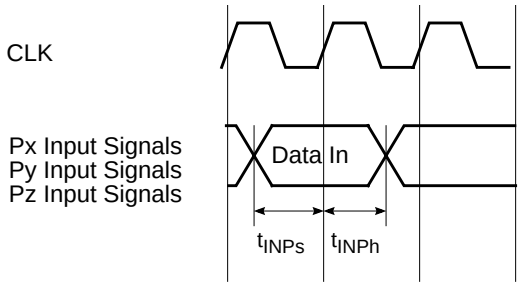



**Figure 132. Input Signal Timing for Input and I/O Port Signals**

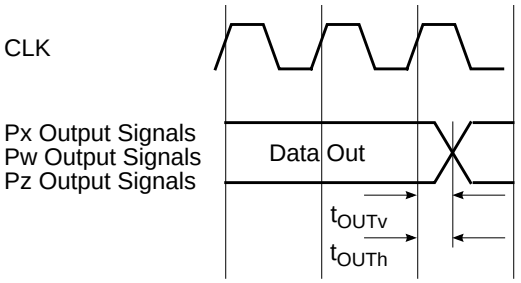



**Figure 133. Output Signal Timing for Output and I/O Port Signals**

# 7.0 Device Specifications (Continued)

## 7.6.6 PWM Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{OUTv}$ | 134 | Output valid time | After RE CLK | | 0.5 * $t_{CLK}$ | - |
| $t_{OUTh}$ | 134 | Output hold time | After RE CLK | 0 | | ns |

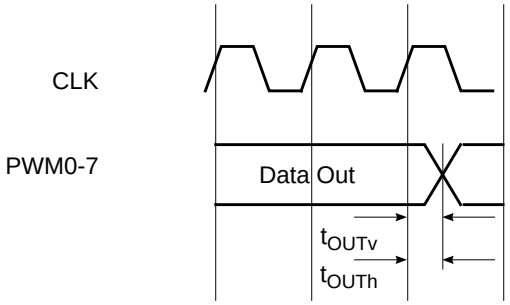



**Figure 134. Output Signal Timing for PWM Signals**

## 7.6.7 MSWC Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{wupd}$ | 135 | Wake-up propagation delay | After FE $\overline{RI2,1}$ | | 30 | ns |
| $t_L$ | 135 | $\overline{RI2,1}$ Low Time | | 10 | | ns |
| $t_H$ | 135 | $\overline{RI2,1}$ High Time | | 10 | | ns |
| $t_{LR}$ | 135 | $\overline{RING}$ Low Time | | 50 | | µs |
| $t_{HR}$ | 135 | $\overline{RING}$ High Time | | 50 | | µs |

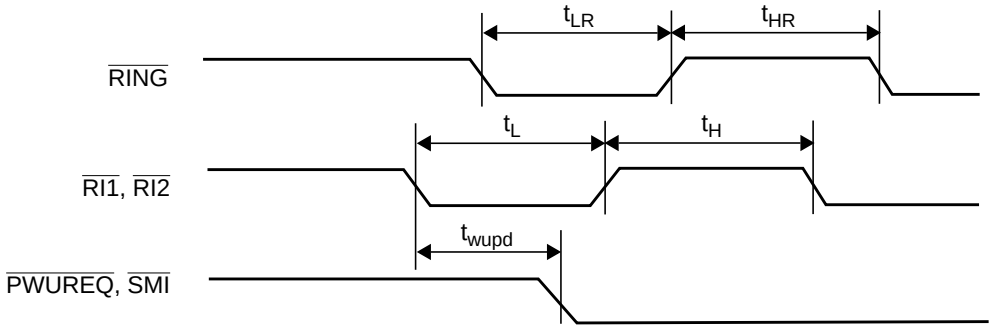



**Figure 135. Wake-Up Timing**

# 7.0 Device Specifications (Continued)

## 7.6.8 PS/2 Interface Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| PS/2 Input Timing | | | | | | |
| $t_{PSDIs}$ | 136 | Input setup time PSDAT1-3 | Before FE PSCLK1-3 | 0 | | ns |
| $t_{PSDIh}$ | 136 | Input hold time PSDAT1-3 | After RE PSCLK1-3 | 0 | | ns |
| $t_{PSCLKl}$ | 136 | PSCLK1-3 low time | At 0.8V (Both Edges) | $(n+1)t_{CLK}$[1] ns | | - |
| $t_{PSCLKh}$ | 136 | PSCLK1-3 high time | At 2.0V (Both Edges) | $(n+1)t_{CLK}$[1] ns | | - |
| PS/2 Output Timing | | | | | | |
| $t_{PSDOv}$ | 137 | Output valid time PSDAT1-4 | After FE PSCLK1-4 | | $(n+6)*t_{CLK}$ + 14[2] ns | - |
| $t_{PSDOh}$ | 137 | Output hold time PSDAT1-4 | After FE PSCLK1-4 | 0 | | ns |
| $t_{PSCLKa}$ | 138 | Output active time PSCLK1-4 | After RE CLK | | 17 | ns |
| $t_{PSCLKia}$ | 138 | Output inactive time PSCLK1-4 | After RE CLK | | 17 | ns |

1. 'n' is the number of clock cycles, as programed in the IDB field. See "PS/2 Control Register (PSCON)" on page 127.
2. 'n' is defined in "PS/2 Control Register (PSCON)" on page 127.

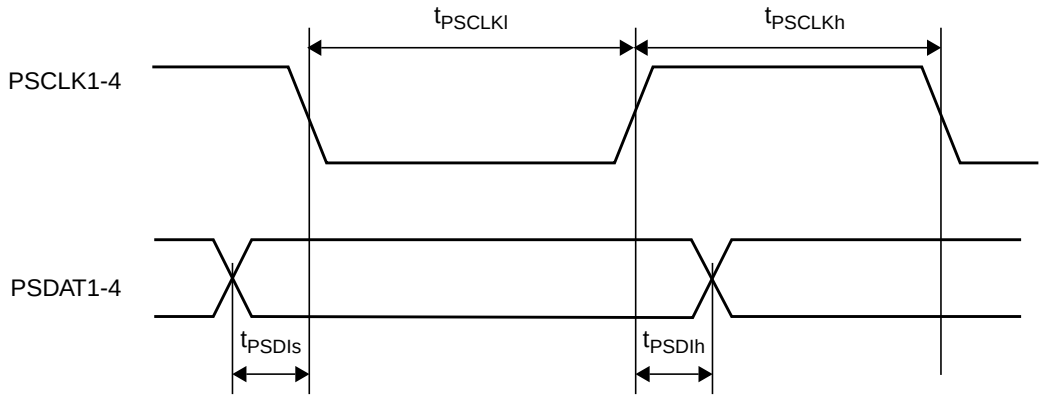



**Figure 136. PS/2 Receive Timing**

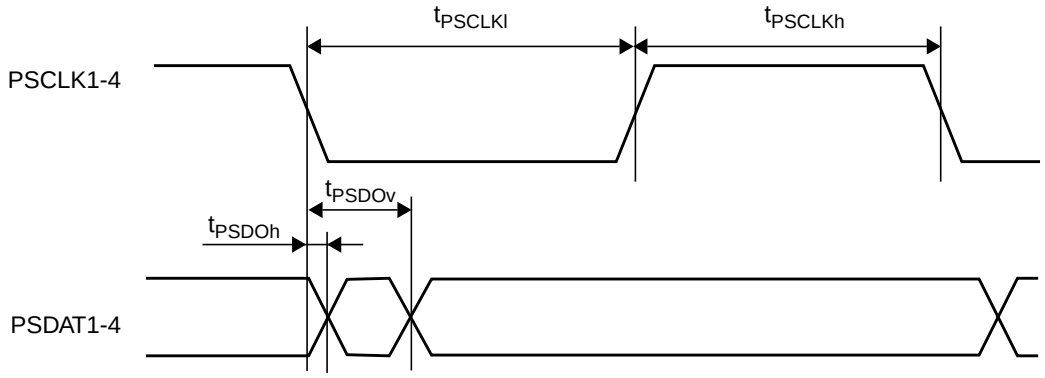



**Figure 137. PS/2 Transmit Timing**

## 7.0 Device Specifications (Continued)

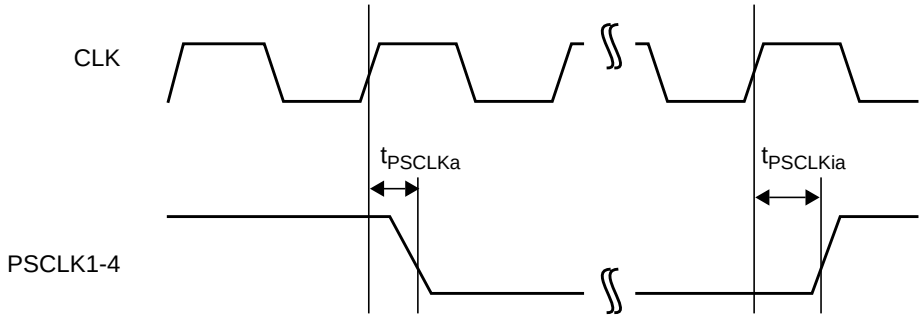



**Figure 138. PS/2 Clock Signal Pulled Low by PC87591L-N05**

### 7.6.9 ACCESS.bus Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| ACB Input Timing | | | | | | |
| $t_{BUFi}$ | 140 | Bus free time between Stop and Start conditions | | $t_{SCLhigho}$ | | - |
| $t_{CSTOsi}$ | 140 | SCL setup time | Before Stop condition | $8 * t_{CLK} - t_{SCLri}$ | | - |
| $t_{CSTRhi}$ | 140, 141 | SCL hold time | After Start condition | $8 * t_{CLK} - t_{SCLri}$ | | - |
| $t_{CSTRsi}$ | 141 | SCL setup time | Before Start condition | $8 * t_{CLK} - t_{SCLri}$ | | - |
| $t_{DHCsi}$ | 141 | Data high setup time | Before SCL RE | $2 * t_{CLK}$ | | - |
| $t_{DLCsi}$ | 140 | Data low setup time | Before SCL RE | $2 * t_{CLK}$ | | - |
| $t_{SCLfi}$ | 139 | SCL signal fall time | | | $300^{1,2}$ | ns |
| $t_{SCLri}$ | 139 | SCL signal rise time | | | $1^{1,2}$ | µs |
| $t_{SCLlowi}$ | 142 | SCL low time | After SCL FE | $16 * t_{CLK}$ | | - |
| $t_{SCLhighi}$ | 142 | SCL high time | After SCL RE | $16 * t_{CLK}$ | | - |
| $t_{SDAfi}$ | 139 | SDA signal fall time | | | $300^{1,2}$ | ns |
| $t_{SDAri}$ | 139 | SDA signal rise time | | | $1^{1,2}$ | µs |
| $t_{SDAhi}$ | 142 | SDA hold time | After SCL FE | 0 | | ns |
| $t_{SDAsi}$ | 142 | SDA setup time | Before SCL RE | $2 * t_{CLK}$ | | - |
| ACB Output Timing | | | | | | |
| $t_{SCLhigho}$ | 142 | SCL high time | After SCL RE | $K * t_{CLK} - 1$ µs | | - |
| $t_{SCLlowo}$ | 142 | SCL low time | After SCL FE | $K * t_{CLK} - 1$ µs | | - |

## 7.0 Device Specifications (Continued)

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{BUFo}$ | 140 | Bus free time between Stop and Start conditions | | $t_{SCLhigho}$ | | - |
| $t_{CSTOso}$ | 140 | SCL setup time | Before Stop condition | $t_{SCLhigho}$ | | - |
| $t_{CSTRho}$ | 140, 141 | SCL hold time | After Start condition | $t_{SCLhigho}$ | | - |
| $t_{CSTRso}$ | 141 | SCL setup time | Before Start condition | $t_{SCLhigho}$ | | - |
| $t_{DHCso}$ | 141 | Data high setup time | Before SCL RE | $t_{SCLhigho} - t_{SDAro}$ | | - |
| $t_{DLCso}$ | 140 | Data low setup time | Before SCL RE | $t_{SCLhigho} - t_{SDAfo}$ | | - |
| $t_{SCLfo}$ | 139 | SCL signal fall time | | | 300[1,3] | ns |
| $t_{SCLro}$ | 139 | SCL signal rise time | | | See note[4] | |
| $t_{SDAfo}$ | 139 | SDA signal fall time | | | 300 | ns |
| $t_{SDAro}$ | 139 | SDA signal rise time | | | See note[1,3,4] | - |
| $t_{SDAho}$ | 142 | SDA hold time | After SCL FE | $7 * t_{CLK} - t_{SCLfo}$ | | - |
| $t_{SDAvo}$ | 142 | SDA valid time | After SCL FE | | $7 * t_{CLK} + t_{SDAro}$ | - |

1. Test conditions: $R_L$ = 2.2 KΩ to $V_{CC}$ = 3.3V, $C_L$ = 400 pF to GND.
2. Not tested; guaranteed by design.
3. Not tested; guaranteed by characterization.
4. Depends on the signal's capacitance and the pull-up value. Must be less than 1 µs.

In Figure 139 through Figure 142, an "o" is added to parameter names in the timing tables for output signals and an "i" for input signals, as displayed in the preceding table:

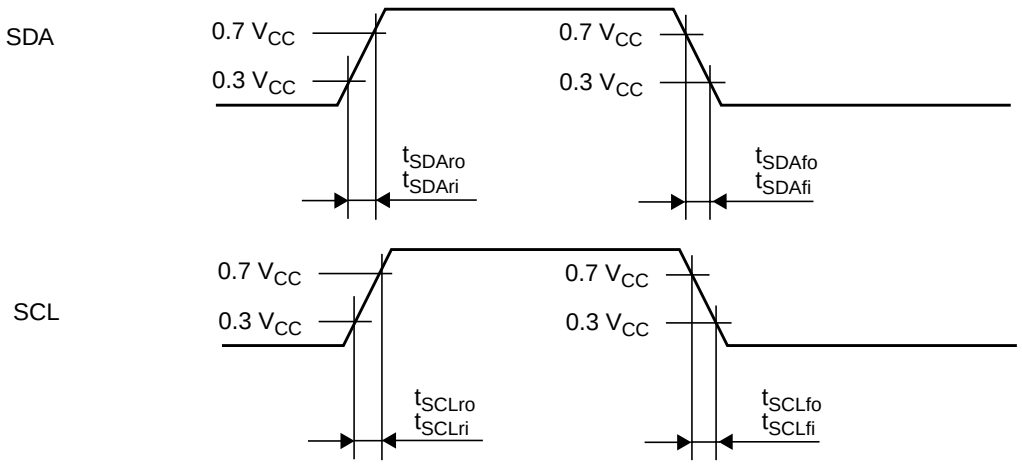



**Figure 139. ACB Signals (SDA and SCL) Rising Time and Falling Time**

## 7.0 Device Specifications (Continued)

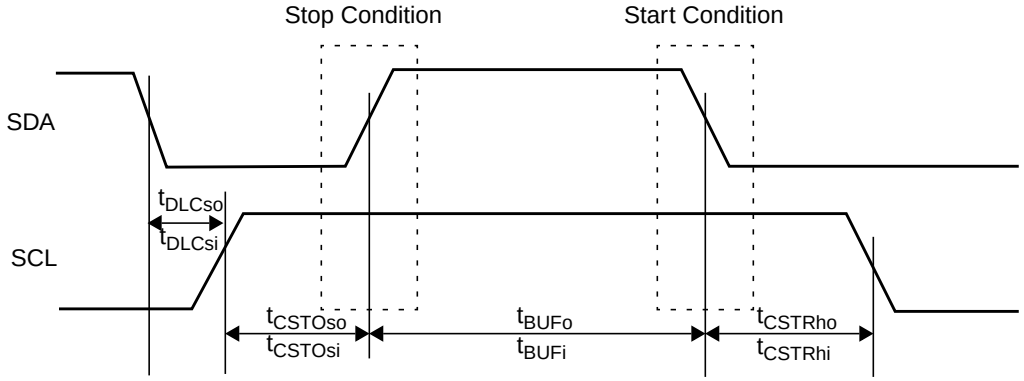



**Figure 140. ACB Start and Stop Condition Timing**

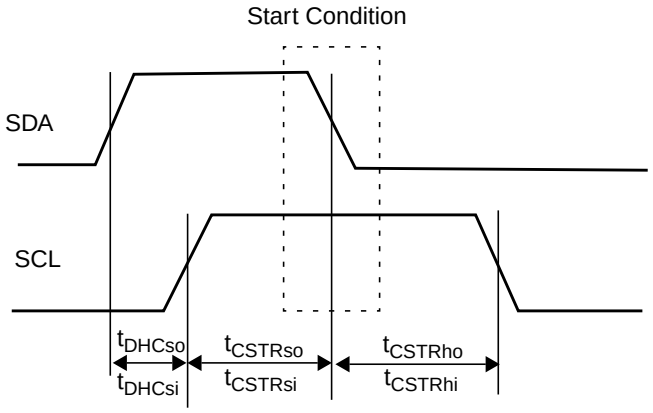



**Figure 141. ACB Start Condition TIming**

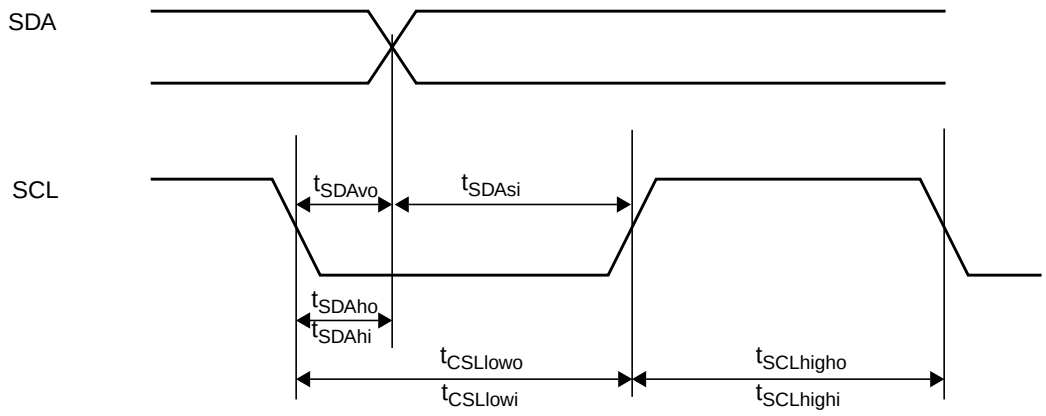



**Figure 142. ACB Data Bit Timing**

# 7.0 Device Specifications (Continued)

## 7.6.10 MFT16 Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| MFT16 Input Timing | | | | | | |
| $t_{TABH}$ | 143 | TA1-2/TB1-2 high time | | $t_{CLK}$ + 5 ns | | - |
| $t_{TABL}$ | 143 | TA1-2/TB1-2 low time | | $t_{CLK}$ + 5 ns | | - |

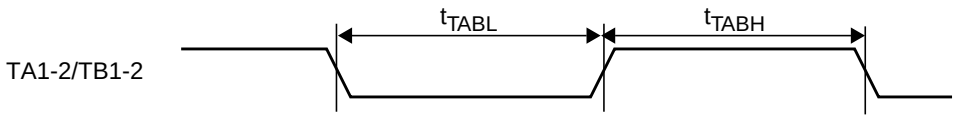



**Figure 143. Multi-Function Timer (MFT16) Input Timing**

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{OUTv}$ | 144 | Output valid time | After RE CLK | | 0.5 * $t_{CLK}$ | - |
| $t_{OUTh}$ | 144 | Output hold time | After RE CLK | 0 | | ns |

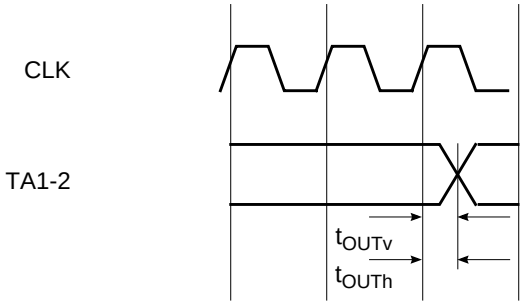



**Figure 144. Multi-Function Timer (MFT16) Output Timing**

## 7.0 Device Specifications (Continued)

### 7.6.11 ICU/Development Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| ICU/Development Input Timing | | | | | | |
| $t_{\overline{BRKLs}}$ | 145 | Input setup time for $\overline{BRKL}$ | Before RE CLK | 10 | | ns |
| $t_{\overline{BRKLh}}$ | 145 | Input hold time for $\overline{BRKL}$ | After RE CLK | 0 | | ns |
| ICU/Development Output Timing | | | | | | |
| $t_{PFSh}$ | 145 | Output hold time PFS, PLI | After RE CLK | $0.5 * t_{CLK} - 6$ ns | | - |
| $t_{PFSv}$ | 145 | Output active/inactive time PFS, PLI | After RE CLK | | $0.5 * t_{CLK} + 12$ ns | - |
| $t_{RSTOd}$[1] | 146 | Output delay time $\overline{RSTO}$ | After RE Internal Reset | $5 * t_{CLK}$ | | - |
| $t_{RSTOw}$[1] | 146 | Output pulse width $\overline{RSTO}$ | FE to RE $\overline{RSTO}$ | $3 * t_{CLK}$ | | - |
| $t_{RSTOs}$[1] | 146 | Output inactive time $\overline{RSTO}$ | Before FE Internal Reset | 0 | | ns |

1. Not tested; guaranteed by design.

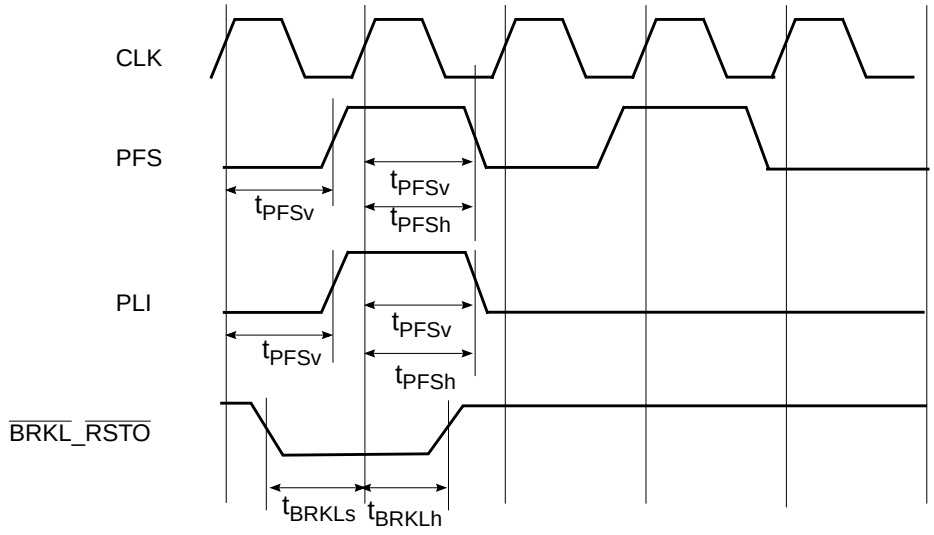



**Figure 145. Pipe Status Signal (PFS and PLI) Timing**

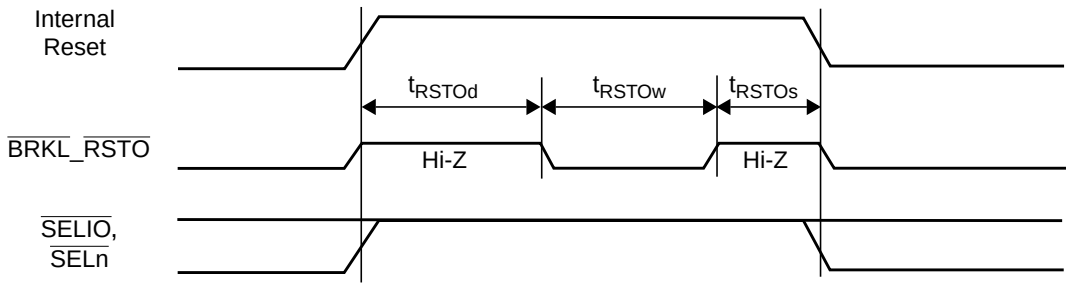



**Figure 146. Reset Out Signal Timing**

# 7.0 Device Specifications (Continued)

## 7.6.12 Asynchronous Edge Detected Signals Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{asw}$ | 147 | Wake-up Input Width:<br>EXWINT20-24, EXWINT(40,45-46) KBSIN0-7, $\overline{PFAIL}$ PSCLK1-4, PSDAT1-4, SWIN | Pulse width that guarantees detection on edge | 15 | | ns |
| $t_{is}$ | 147 | Input setup time:<br>EXWINT20-24, EXWINT(40,45-46, $\overline{PFAIL}$, PSCLK1-4, PSDAT1-4, SWIN | See note[1] | 10[2] | | ns |
| $t_{ih}$ | 147 | Input hold time:<br>EXWINT20-24, EXWINT(40,45-46), $\overline{PFAIL}$, PSCLK1-4, PSDAT1-4, SWIN | See note[1] | 0[2] | | ns |

1. All wake-ups are asynchronous. Meeting the setup and hold are required for repeatability of the wake cycle only.
2. Not tested; guaranteed by characterization.

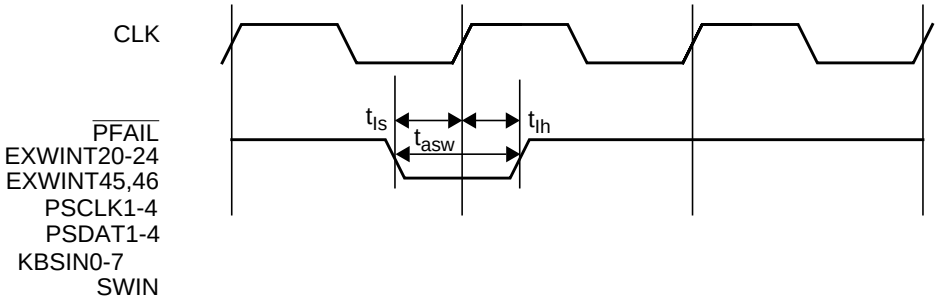



**Figure 147. $\overline{PFAIL}$, EXINTn and MIWU Input Signal Timing**

# 7.0 Device Specifications (Continued)

## 7.6.13 Debugger Interface Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| Debugger Interface Input Signals | | | | | | |
| $t_S$ | 148 | TMS Setup Time | Before RE TCK | 7 | | ns |
| $t_H$ | 148 | TMS Hold Time | After RE TCK | 2 | | ns |
| $t_S$ | 148 | TDI Setup Time | Before RE TCK | 1 | | ns |
| $t_H$ | 148 | TDI Hold Time | After RE TCK | 3.5 | | ns |
| $t_{HW}$ | 149 | TCK High Pulse Width | | 15 | | ns |
| $t_{LW}$ | 149 | TCK Low Pulse Width | | 15 | | ns |
| $f_{MAX}$ | | Maximum TCK Clock Frequency | | | 20 | MHz |
| $T_{pu}$ | 121 | Wait Time Power-Up to TCK | | $t_{IRST}$[1] | | - |
| Debugger Interface Output Signals | | | | | | |
| $t_{PLH}$<br>$t_{PLL}$ | 150 | Propagation Delay TCK to TDO | After FE TCK | 3<br>3 | 22<br>22 | ns |
| $t_{PLZ}$<br>$t_{PHZ}$ | 152<br>151 | Disable Time TCK to TDO | | 2<br>2 | 22<br>22 | ns |
| $t_{PZL}$<br>$t_{PZH}$ | 152<br>151 | Enable Time TCK to TDO | | 3<br>3 | 22<br>22 | ns |
| $t_{TINTh}$ | 153 | $\overline{TINT}$ hld time | After FE TCK | 10 | | ns |
| $t_{TINTv}$ | 153 | $\overline{TINT}$ valid time | After RE CLK | | 25 | ns |

1. See Section 7.6.2 on page 345.

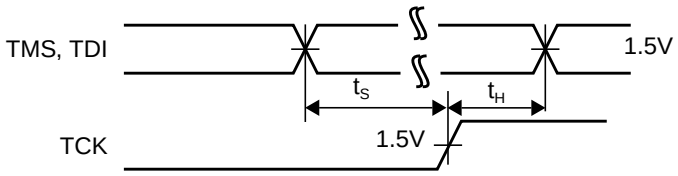



**Figure 148. Debugger I/F Setup Time, Hold Time and Recovery Time**

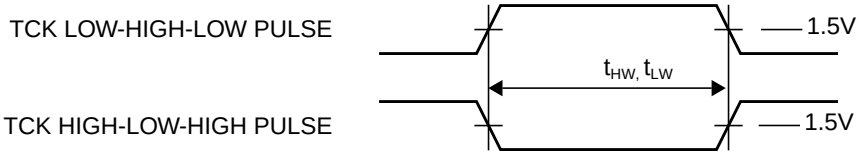



**Figure 149. TCK Pulse Width**

## 7.0 Device Specifications (Continued)

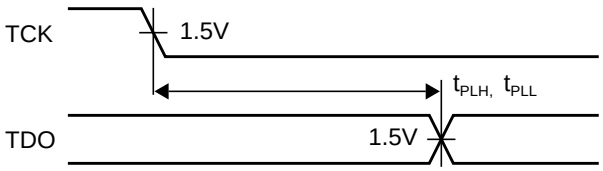



**Figure 150. Debugger Interface Propagation Delay**

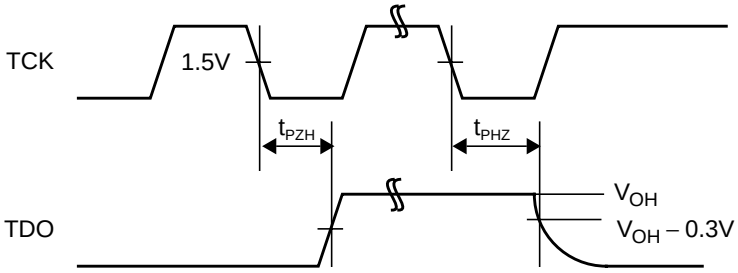



**Figure 151. TDO TRI-STATE Output High Enable and Disable Times**

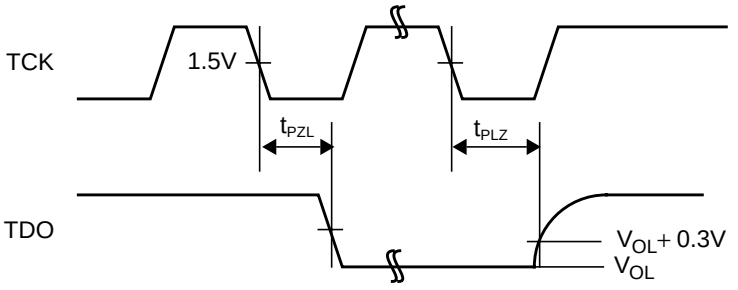



**Figure 152. TDO TRI-STATE Output Low Enable and Disable Times**

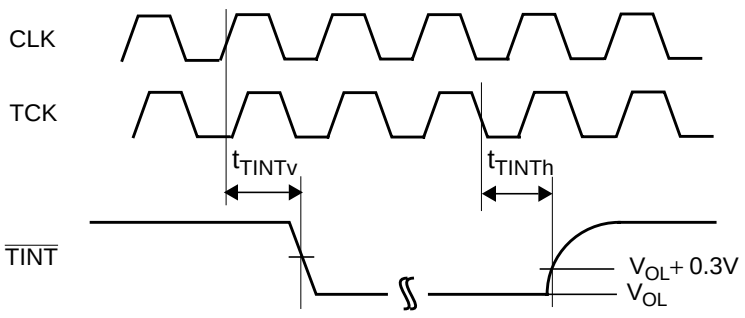



**Figure 153. $\overline{TINT}$ Output Timing**

# 7.0 Device Specifications (Continued)

## 7.6.14 USART Timing

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $t_r$ | 154 | CMOS output rise time. | Note | | 6[1] | ns |
| $t_f$ | 154 | CMOS output fall time. | Note | | 6[1] | ns |
| Asynchronous Mode | | | | | | |
| $t_{ACOv}$ | 154 | Output valid time | After RE CLK | 0 | 20 | ns |
| $t_{AIs}$ | 154 | Input setup time | Before RE CLK | 15 | | ns |
| $t_{AIh}$ | 154 | Input hold time | After RE CLK | 0 | | ns |
| Synchronous Mode, USCLK Input | | | | | | |
| $t_{CLKX}$ | 155 | USCLK input period | | 250 | | ns |
| $t_{CLKXh}$ | 155 | USCLK input low time. | | $t_{CLKX}$ / 2 | | - |
| $t_{CLKXl}$ | 155 | USCLK input high time. | | $t_{CLKX}$ / 2 | | - |
| $t_{SCOv1}$ | 155 | Output hold valid time | After RE USCLK | 0 | 25 | ns |
| $t_{SIs1}$ | 155 | Input setup time | Before FE USCLK | 20 | | ns |
| $t_{SIh1}$ | 155 | Input hold time | After FE USCLK | 20 | | ns |
| Synchronous Mode, USCLK Output | | | | | | |
| $t_{SCOv2}$ | 156 | Output hold valid time | After RE USCLK | 0 | 20 | ns |
| $t_{SIs2}$ | 156 | Input setup time | Before FE USCLK | 20 | | ns |
| $t_{SIh2}$ | 156 | Input hold time | After FE USCLK | 20 | | ns |

1. Not tested; guaranteed by characterization.

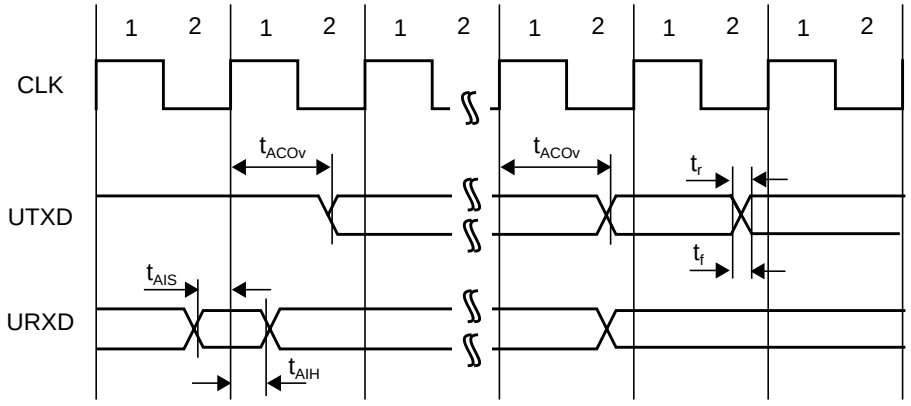



**Figure 154. USART Asynchronous Mode Timing**

## 7.0 Device Specifications (Continued)

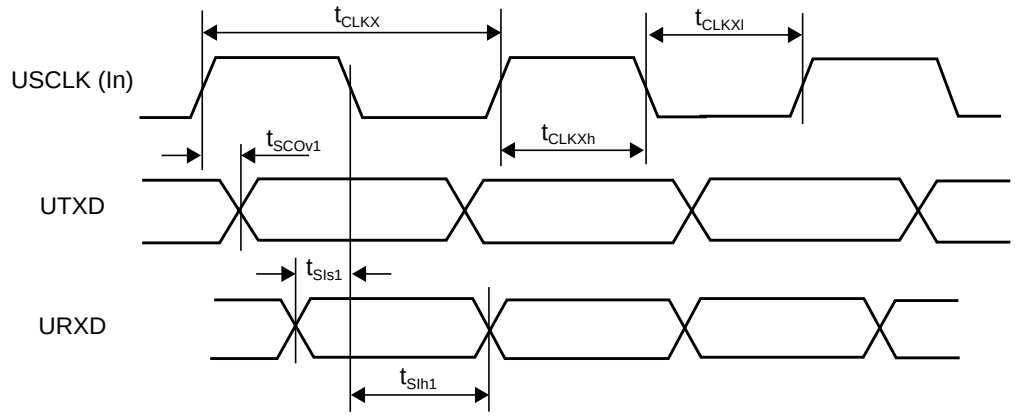



**Figure 155. USART Synchronous Mode Timing, USCLK Input**

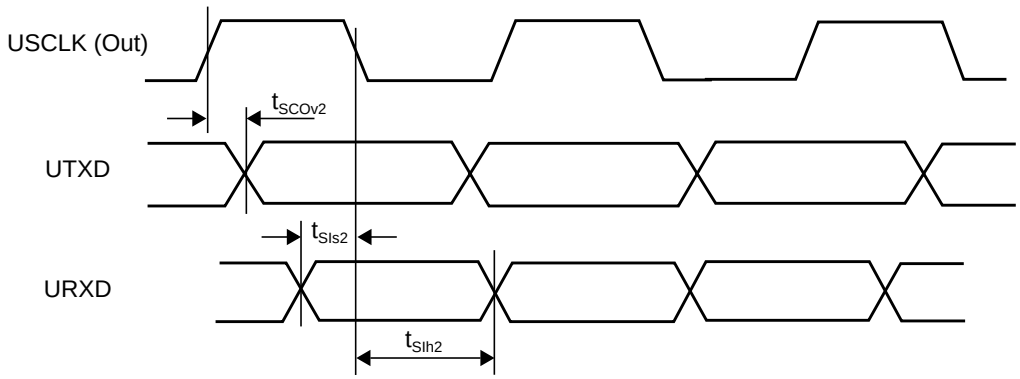



**Figure 156. USART Synchronous Mode Timing, USCLK Output**

## 7.0 Device Specifications (Continued)

### 7.6.15 LCLK and $\overline{\text{RESET1-2}}$

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_{CYC}$[1] | LCLK Cycle Time | 30 | | ns |
| $t_{HIGH}$ | LCLK High Time | 11 | | ns |
| $t_{LOW}$ | LCLK Low Time | 11 | | ns |
| - | LCLK Slew Rate[2] | 1 | 4 | V/ns |
| - | $\overline{\text{RESET1-2}}$ Slew Rate[3] | 50 | | mV/ns |

1. The PCI may have any clock frequency between nominal DC and 33 MHz. Device operational parameters at frequencies under 16 MHz may be guaranteed by design rather than by testing. The clock frequency may be changed at any time during the operation of the system as long as the clock edges remain “clean” (monotonic) and the minimum cycle and high and low times are not violated. The clock may only be stopped in a low state.
2. Rise and fall times are specified in terms of the edge rate measured in V/ns. This slew rate must be met across the minimum peak-to-peak portion of the clock wavering as shown below.
3. The minimum $\overline{\text{RESET1-2}}$ slew rate applies only to the rising (de-assertion) edge of the reset signal, and ensures that system noise cannot render an otherwise monotonic signal to appear to bounce in the switching range.

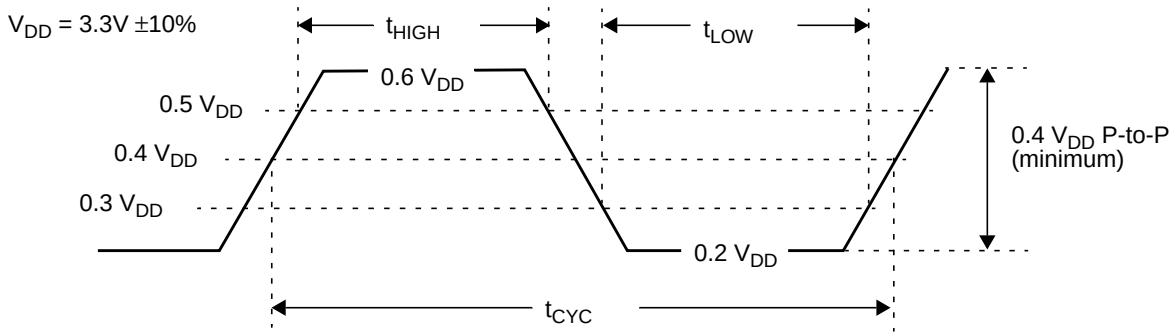



**Figure 157. LCLK Waveform**

## 7.0 Device Specifications (Continued)

### 7.6.16 LPC and SERIRQ Signals

| Symbol | Figure | Description | Reference Conditions | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{VAL}$ | 158 | Output Valid Delay | After RE CLK | 2[1] | 11 | ns |
| $t_{ON}$ | 158 | Float to Active Delay | After RE CLK | 2 | | ns |
| $t_{OFF}$ | 158 | Active to Float Delay | After RE CLK | | 28 | ns |
| $t_{SU}$ | 159 | Input Setup Time | Before RE CLK | 7 | | ns |
| $t_{HI}$ | 159 | Input Hold Time | After RE CLK | 0 | | ns |

1. Not tested; guaranteed by characterization.

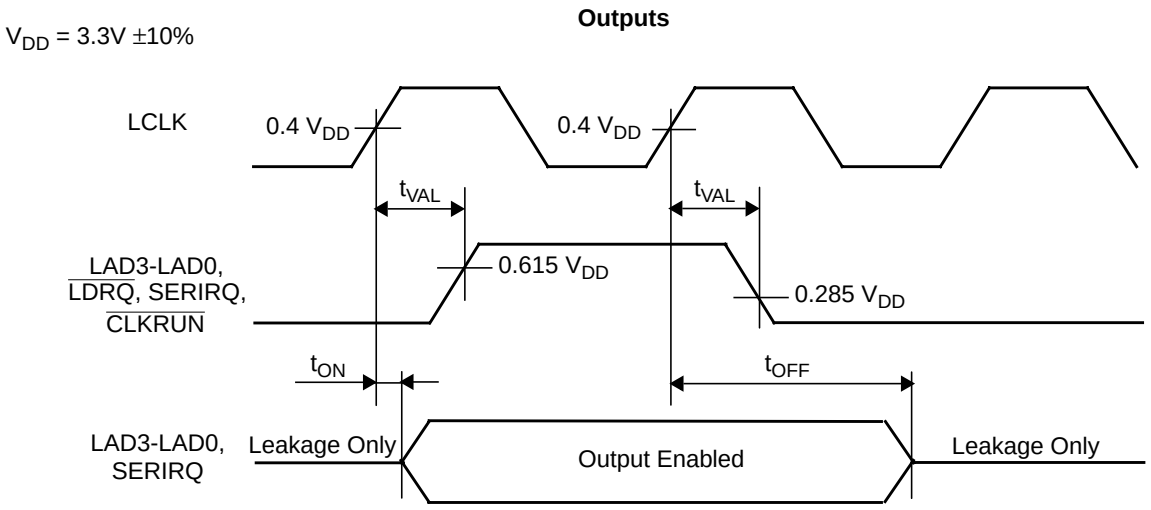



**Figure 158. LPC/SERIRQ Interface Output Timing**

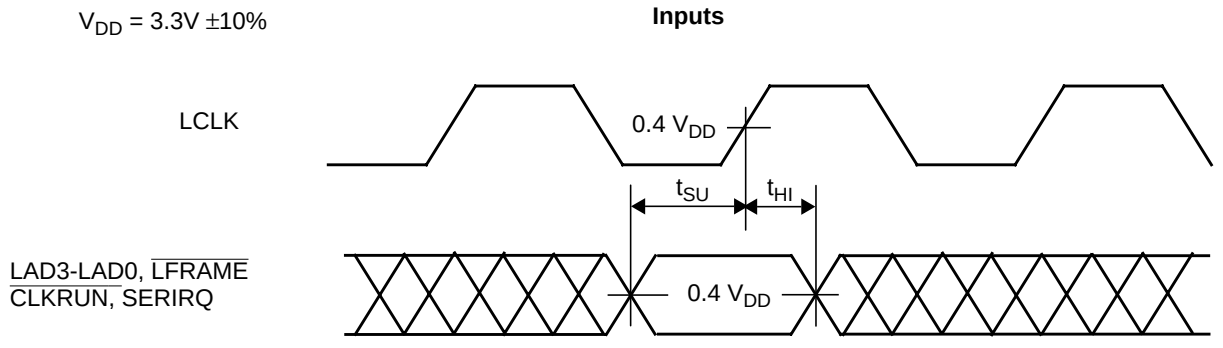



**Figure 159. LPC/SERIRQ Interface Input Timing**

# A. Register List

## A.1 CORE DOMAIN REGISTERS

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|

### A.1.1 Module Configuration

(Section 2.3 on page 48 and Section 2.4.2 on page 52)

| | | | | | |
|---|---|---|---|---|---|
| EICFG | Byte | 00 FF00$_{16}$ | Read/Write | 00$_{16}$ | |
| IOEE1 | Byte | 00 FF02$_{16}$ | Read/Write | 00$_{16}$ | |
| IOEE2 | Byte | 00 FF04$_{16}$ | Read/Write | 00$_{16}$ | |
| MCFG | Byte | 00 FF10$_{16}$ | Read/Write | 00$_{16}$ or 80$_{16}$ | |
| MCFGSH | Byte | 00 FBFE$_{16}$ | Write Only | MCFG shadow | |
| STRPST | Byte | 00 FF12$_{16}$ | Read Only | According to external straps | |
| PTWRL | Byte | 00 FF06$_{16}$ | Read/Write | FE$_{16}$ | |
| PTWRH | Byte | 00 FF08$_{16}$ | Read/Write | FF$_{16}$ | |
| PNMR | Byte | 00 FF0A$_{16}$ | Read/Write | 02$_{16}$ | |

### A.1.2 Bus Interface Unit (BIU)

(Section 4.1.10 on page 81)

| | | | | | |
|---|---|---|---|---|---|
| BCFG | Byte | 00 F980$_{16}$ | Read/Write | 07$_{16}$ | |
| IOCFG | Word | 00 F982$_{16}$ | Read/Write | 069F$_{16}$ | |
| SZCFG0 | Word | 00 F984$_{16}$ | Read/Write | 069F$_{16}$ | |
| SZCFG1 | Word | 00 F986$_{16}$ | Read/Write | 069F$_{16}$ | |
| SZCFG2 | Word | 00 F988$_{16}$ | Read/Write | 069F$_{16}$ | |

### A.1.3 DMA Controller

(Section 4.2.8 on page 90)

| | | | | | |
|---|---|---|---|---|---|
| ADCA0 | Double W. | 00 FA00$_{16}$ | Read/Write | | |
| ADRA0 | Double W. | 00 FA04$_{16}$ | Read/Write | | |
| ADCB0 | Double W. | 00 FA08$_{16}$ | Read/Write | | |
| ADRB0 | Double W. | 00 FA0C$_{16}$ | Read/Write | | |
| BLTC0 | Double W. | 00 FA10$_{16}$ | Read/Write | | |
| BLTR0 | Double W. | 00 FA14$_{16}$ | Read/Write | | |
| DMACNTL0 | Word | 00 FA1C$_{16}$ | Read/Write | 0000$_{16}$ | |

# A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|
| DMASTAT0 | Byte | 00 FA1E$_{16}$ | Read/Write | 00$_{16}$ | |
| ADCA1 | Double W. | 00 FA20$_{16}$ | Read/Write | | |
| ADRA1 | Double W. | 00 FA24$_{16}$ | Read/Write | | |
| ADCB1 | Double W. | 00 FA28$_{16}$ | Read/Write | | |
| ADRB1 | Double W. | 00 FA2C$_{16}$ | Read/Write | | |
| BLTC1 | Double W. | 00 FA30$_{16}$ | Read/Write | | |
| BLTR1 | Double W. | 00 FA34$_{16}$ | Read/Write | | |
| DMACNTL1 | Word | 00 FA3C$_{16}$ | Read/Write | 0000$_{16}$ | |
| DMASTAT1 | Byte | 00 FA3E$_{16}$ | Read Only | 00$_{16}$ | |
| ADCA2 | Double W. | 00 FA40$_{16}$ | Read/Write | | |
| ADRA2 | Double W. | 00 FA44$_{16}$ | Read/Write | | |
| ADCB2 | Double W. | 00 FA48$_{16}$ | Read/Write | | |
| ADRB2 | Double W. | 00 FA4C$_{16}$ | Read/Write | | |
| BLTC2 | Double W. | 00 FA50$_{16}$ | Read/Write | | |
| BLTR2 | Double W. | 00 FA54$_{16}$ | Read/Write | | |
| DMACNTL2 | Word | 00 FA5C$_{16}$ | Read/Write | 0000$_{16}$ | |
| DMASTAT2 | Byte | 00 FA5E$_{16}$ | Read Only | 00$_{16}$ | |
| ADCA3 | Double W. | 00 FA60$_{16}$ | Read/Write | | |
| ADRA3 | Double W. | 00 FA64$_{16}$ | Read/Write | | |
| ADCB3 | Double W. | 00 FA68$_{16}$ | Read/Write | | |
| ADRB3 | Double W. | 00 FA6C$_{16}$ | Read/Write | | |
| BLTC3 | Double W. | 00 FA70$_{16}$ | Read/Write | | |
| BLTR3 | Double W. | 00 FA74$_{16}$ | Read/Write | | |
| DMACNTL3 | Word | 00 FA7C$_{16}$ | Read/Write | 0000$_{16}$ | |
| DMASTAT3 | Byte | 00 FA7E$_{16}$ | Read Only | 00$_{16}$ | |

### A.1.4 General-Purpose I/O (GPIO) Ports

(Section 4.5.6 on page 116)

| | | | | | |
|---|---|---|---|---|---|
| PADIR | Byte | 00 FE20$_{16}$ | Read/Write | 00$_{16}$ | |
| PADIN | Byte | 00 FE22$_{16}$ | Read Only | | |
| PADOUT | Byte | 00 FE24$_{16}$ | Read/Write | | |
| PAWPU | Byte | 00 FE26$_{16}$ | Read/Write | 00$_{16}$ | |
| PAALT | Byte | 00 FE28$_{16}$ | Read/Write | 00$_{16}$ | |

## A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|
| PBDIR | Byte | 00 FE2A$_{16}$ | Read/Write | 20$_{16}$ | |
| PBDIN | Byte | 00 FE2C$_{16}$ | Read Only | | |
| PBDOUT | Byte | 00 FE2E$_{16}$ | Read/Write | Bit 5 is 1; the others are undefined | |
| PBWPU | Byte | 00 FE30$_{16}$ | Read/Write | 00$_{16}$ | |
| PBALT | Byte | 00 FE32$_{16}$ | Read/Write | 40$_{16}$ | |
| PCDIR | Byte | 00 FE34$_{16}$ | Read/Write | 00$_{16}$ | PC0 is reset on $V_{CC}$ Power-Up and Watchdog reset only. |
| PCDIN | Byte | 00 FE36$_{16}$ | Read Only | | |
| PCDOUT | Byte | 00 FE38$_{16}$ | Read/Write | | |
| PCWPU | Byte | 00 FE3A$_{16}$ | Read/Write | 00$_{16}$ | |
| PCALT | Byte | 00 FE3C$_{16}$ | Read/Write | 00$_{16}$ | |
| PDDIR | Byte | 00 FE3E$_{16}$ | Read/Write | 00$_{16}$ | |
| PDDIN | Byte | 00 FE40$_{16}$ | Read Only | | |
| PDDOUT | Byte | 00 FE42$_{16}$ | Read/Write | | |
| PDWPU | Byte | 00 FE44$_{16}$ | Read/Write | 00$_{16}$ | |
| PDALT | Byte | 00 FE46$_{16}$ | Read/Write | 00$_{16}$ | |
| PEDIN | Byte | 00 FE48$_{16}$ | Read Only | | |
| PEWPU | Byte | 00 FE4A$_{16}$ | Read/Write | 00$_{16}$ | |
| PEALT | Byte | 00 FE4C$_{16}$ | Read/Write | 00$_{16}$ | |
| KBSIN | Byte | 00 FE4E$_{16}$ | Read Only | | |
| KBSINPU | Byte | 00 FE50$_{16}$ | Read/Write | | |
| KBSOUT | Word | 00 FE52$_{16}$ | Read/Write | FFFF$_{16}$ | |
| PFDIR | Byte | 00 FE54$_{16}$ | Read/Write | 00$_{16}$ | |
| PFDIN | Byte | 00 FE56$_{16}$ | Read Only | | |
| PFDOUT | Byte | 00 FE58$_{16}$ | Read/Write | | |
| PFWPU | Byte | 00 FE5A$_{16}$ | Read/Write | 00$_{16}$ | |
| PFALT | Byte | 00 FE5C$_{16}$ | Read/Write | 00$_{16}$ | |
| PJDIR | Byte | 00 FB0C$_{16}$ | Read/Write | 00$_{16}$ | Bits 7-2 only |
| PJDIN | Byte | 00 FB0E$_{16}$ | Read Only | | Bits 7-2 only |
| PJDOUT | Byte | 00 FB10$_{16}$ | Read/Write | | Bits 7-2 only |
| PLDIR | Byte | 00 FB18$_{16}$ | Read/Write | 00$_{16}$ | Bits 4-3 only |
| PLDIN | Byte | 00 FB1A$_{16}$ | Read Only | | Bits 4-3 only |
| PLDOUT | Byte | 00 FB1C$_{16}$ | Read/Write | | Bits 4-3 only |

# A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|
| PMDIRX | Byte | 00 FB1E$_{16}$ | Read/Write | 00$_{16}$ | |
| PMDIN | Byte | 00 FB20$_{16}$ | Read Only | | |
| PMDOUT | Byte | 00 FB22$_{16}$ | Read/Write | | |
| PQDIR | Byte | 00 FE5E$_{16}$ | Read/Write | 0D$_{16}$ | Bits 3-0 only |
| PQDIN | Byte | 00 FE60$_{16}$ | Read Only | | Bits 3-0 only |
| PQDOUT | Byte | 00 FE62$_{16}$ | Read/Write | 00$_{16}$ | Bits 3-0 only |
| PQWPU | Byte | 00 FE64$_{16}$ | Read/Write | 00$_{16}$ | Bits 3-0 only; bit 3 is 0 |
| PQALT | Byte | 00 FE66$_{16}$ | Read/Write | 0F$_{16}$ | Bits 3-0 only; bit 3 is 1 |

### A.1.5 PS/2 Ports

(Section 4.6.5 on page 125)

| | | | | | |
|---|---|---|---|---|---|
| PSDAT | Byte | 00 FE80$_{16}$ | Read/Write | | |
| PSTAT | Byte | 00 FE82$_{16}$ | Read Only | 00$_{16}$ | |
| PSCON | Byte | 00 FE84$_{16}$ | Read/Write | 00$_{16}$ | |
| PSOSIG | Byte | 00 FE86$_{16}$ | Read/Write | 47$_{16}$ | |
| PSISIG | Byte | 00 FE88$_{16}$ | Read Only | | |
| PSIEN | Byte | 00 FE8A$_{16}$ | Read/Write | 00$_{16}$ | |

### A.1.6 Host Interface (KBC, PM1 and PM2 Channels)

(Section 5.1.4 on page 247 and Section 5.2.3 on page 256)

| | | | | | |
|---|---|---|---|---|---|
| HICTRL | Byte | 00 FEA0$_{16}$ | Read/Write | 00$_{16}$ | |
| HIIRQC | Byte | 00 FEA2$_{16}$ | Read/Write | 07$_{16}$ | |
| HIKMST | Byte | 00 FEA4$_{16}$ | Read/Write | 00$_{16}$ | |
| HIKDO | Byte | 00 FEA6$_{16}$ | Write Only | | |
| HIMDO | Byte | 00 FEA8$_{16}$ | Write Only | | |
| HIKMDI | Byte | 00 FEAA$_{16}$ | Read Only | | |
| HIPM1ST | Byte | 00 FEAC$_{16}$ | Varies per bit | 00$_{16}$ | |
| HIPM1DO | Byte | 00 FEAE$_{16}$ | Write Only | | |
| HIPM1DI | Byte | 00 FEB0$_{16}$ | Read Only | | |
| HIPM1DOC | Byte | 00 FEB2$_{16}$ | Write Only | | |
| HIPM1DOM | Byte | 00 FEB4$_{16}$ | Write Only | | |
| HIPM1DIC | Byte | 00 FEB6$_{16}$ | Read Only | | |

# A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|
| HIPM1CTL | Byte | 00 FEB8$_{16}$ | Read/Write | 40$_{16}$ | |
| HIPM1IC | Byte | 00 FEBA$_{16}$ | Read/Write | 41$_{16}$ | |
| HIPM1IE | Byte | 00 FEBC$_{16}$ | Read/Write | 00$_{16}$ | |
| HIPM2ST | Byte | 00 FEBE$_{16}$ | Varies per bit | 00$_{16}$ | |
| HIPM2DO | Byte | 00 FEC0$_{16}$ | Write Only | | |
| HIPM2DI | Byte | 00 FEC2$_{16}$ | Read Only | | |
| HIPM2DOC | Byte | 00 FEC4$_{16}$ | Write Only | | |
| HIPM2DOM | Byte | 00 FEC6$_{16}$ | Write Only | | |
| HIPM2DIC | Byte | 00 FEC8$_{16}$ | Read Only | | |
| HIPM2CTL | Byte | 00 FECA$_{16}$ | Read/Write | C0$_{16}$ | |
| HIPM2IC | Byte | 00 FECC$_{16}$ | Read/Write | 41$_{16}$ | |
| HIPM2IE | Byte | 00 FECE$_{16}$ | Read/Write | 00$_{16}$ | |

### A.1.7 Multi-Function Timer (MTF16) 1

(Section 4.7.7 on page 137)

| | | | | | |
|---|---|---|---|---|---|
| T1CNT1 | Word | 00 FD80$_{16}$ | Read/Write | | |
| T1CRA | Word | 00 FD82$_{16}$ | Read/Write | | |
| T1CRB | Word | 00 FD84$_{16}$ | Read/Write | | |
| T1CNT2 | Word | 00 FD86$_{16}$ | Read/Write | | |
| T1PRSC | Byte | 00 FD88$_{16}$ | Read/Write | 00$_{16}$ | |
| T1CKC | Byte | 00 FD8A$_{16}$ | Read/Write | 00$_{16}$ | |
| T1CTRL | Byte | 00 FD8C$_{16}$ | Read/Write | 00$_{16}$ | |
| T1ICTL | Byte | 00 FD8E$_{16}$ | Read/Write | 00$_{16}$ | |
| T1ICLR | Byte | 00 FD90$_{16}$ | Write Only | | |

### A.1.8 Multi-Function Timer (MFT16) 2

(Section 4.7.7 on page 137)

| | | | | | |
|---|---|---|---|---|---|
| T2CNT1 | Word | 00 FDA0$_{16}$ | Read/Write | | |
| T2CRA | Word | 00 FDA2$_{16}$ | Read/Write | | |
| T2CRB | Word | 00 FDA4$_{16}$ | Read/Write | | |
| T2CNT2 | Word | 00 FDA6$_{16}$ | Read/Write | | |
| T2PRSC | Byte | 00 FDA8$_{16}$ | Read/Write | 00$_{16}$ | |
| T2CKC | Byte | 00 FDAA$_{16}$ | Read/Write | 00$_{16}$ | |

# A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|
| T2CTRL | Byte | 00 $FDAC_{16}$ | Read/Write | $00_{16}$ | |
| T2ICTL | Byte | 00 $FDAE_{16}$ | Read/Write | $00_{16}$ | |
| T2ICLR | Byte | 00 $FDB0_{16}$ | Write Only | | |

### A.1.9 Timing and Watchdog (TWD)

(Section 4.10.3 on page 162)

| | | | | | |
|---|---|---|---|---|---|
| TWCFG | Byte | 00 $FEE0_{16}$ | Read/Write | $00_{16}$ | |
| TWCP | Byte | 00 $FEE2_{16}$ | Read/Write | $00_{16}$ | |
| TWDT0 | Word | 00 $FEE4_{16}$ | Read/Write | $FFFF_{16}$ | |
| T0CSR | Byte | 00 $FEE6_{16}$ | Read/Write | $00_{16}$ | |
| WDCNT | Byte | 00 $FEE8_{16}$ | Write Only | $0F_{16}$ | |
| WDSDM | Byte | 00 $FEEA_{16}$ | Write Only | | Write $5C_{16}$ |

### A.1.10 Analog to Digital Converter (ADC)

(Section 4.11.5 on page 171)

| | | | | | |
|---|---|---|---|---|---|
| ADCSTS | Byte | 00 $FF20_{16}$ | Varies per bit | $00_{16}$ | Bit 2 is reset on $V_{CC}$ Power-Up reset only |
| ADCCNF | Byte | 00 $FF22_{16}$ | Read/Write | $00_{16}$ | |
| ACLKCTL | Byte | 00 $FF24_{16}$ | Read/Write | $3F_{16}$ | |
| ADLYCTL | Byte | 00 $FF26_{16}$ | Read/Write | $A7_{16}$ | |
| ADCPINX | Byte | 00 $FF2A_{16}$ | Read/Write | $00_{16}$ | |
| ADCPD | Word | 00 $FF2C_{16}$ | Read/Write | | |
| VCHN1CTL | Byte | 00 $FF34_{16}$ | Varies per bit | $1F_{16}$ | |
| VCHN1DAT | Word | 00 $FF36_{16}$ | Read Only | | |
| VCHN2CTL | Byte | 00 $FF38_{16}$ | Varies per bit | $1F_{16}$ | |
| VCHN2DAT | Word | 00 $FF3A_{16}$ | Read Only | | |
| VCHN3CTL | Byte | 00 $FF3C_{16}$ | Varies per bit | $1F_{16}$ | |
| VCHN3DAT | Word | 00 $FF3E_{16}$ | Read Only | | |

### A.1.11 Digital to Analog Converter (DAC)

(Section 4.12.5 on page 181)

| | | | | | |
|---|---|---|---|---|---|
| DACCTRL | Byte | 00 $FF40_{16}$ | Read/Write | $00_{16}$ | |
| DACDAT0 | Byte | 00 $FF42_{16}$ | Read/Write | | |

# A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|
| DACDAT1 | Byte | 00 $FF44_{16}$ | Read/Write | | |
| DACDAT2 | Byte | 00 $FF46_{16}$ | Read/Write | | |
| DACDAT3 | Byte | 00 $FF48_{16}$ | Read/Write | | |

### A.1.12 ACCESS.bus Interface (ACB) 1

(Section 4.13.8 on page 190)

| | | | | | |
|---|---|---|---|---|---|
| ACB1SDA | Byte | 00 $FF60_{16}$ | Read/Write | | |
| ACB1ST | Byte | 00 $FF62_{16}$ | Varies per bit | $00_{16}$ | |
| ACB1CST | Byte | 00 $FF64_{16}$ | Varies per bit | $00_{16}$ | |
| ACB1CTL1 | Byte | 00 $FF66_{16}$ | Read/Write | $00_{16}$ | |
| ACB1ADDR | Byte | 00 $FF68_{16}$ | Read/Write | | |
| ACB1CTL2 | Byte | 00 $FF6A_{16}$ | Read/Write | $00_{16}$ | |
| ACB1ADDR2 | Byte | 00 $FE6C_{16}$ | Read/Write | | |
| ACB1CTL3 | Byte | 00 $FF6E_{16}$ | Read/Write | $00_{16}$ | |

### A.1.13 ACCESS.bus Interface (ACB) 2

(Section 4.13.8 on page 190)

| | | | | | |
|---|---|---|---|---|---|
| ACB2SDA | Byte | 00 $FFE0_{16}$ | Read/Write | | |
| ACB2ST | Byte | 00 $FFE2_{16}$ | Varies per bit | $00_{16}$ | |
| ACB2CST | Byte | 00 $FFE4_{16}$ | Varies per bit | $00_{16}$ | |
| ACB2CTL1 | Byte | 00 $FFE6_{16}$ | Read/Write | $00_{16}$ | |
| ACB2ADDR | Byte | 00 $FFE8_{16}$ | Read/Write | | |
| ACB2CTL2 | Byte | 00 $FFEA_{16}$ | Read/Write | $00_{16}$ | |
| ACB2ADDR2 | Byte | 00 $FFEC_{16}$ | Read/Write | | |
| ACB2CTL3 | Byte | 00 $FFEE_{16}$ | Read/Write | $00_{16}$ | |

### A.1.14 ACCESS.bus Interface (ACB) 3

(Section 4.13.8 on page 190)

| | | | | | |
|---|---|---|---|---|---|
| ACB3SDA | Byte | 00 $FC40_{16}$ | Read/Write | | |
| ACB3ST | Byte | 00 $FC42_{16}$ | Varies per bit | $00_{16}$ | |
| ACB3CST | Byte | 00 $FC44_{16}$ | Varies per bit | $00_{16}$ | |
| ACB3CTL1 | Byte | 00 $FC46_{16}$ | Read/Write | $00_{16}$ | |
| ACB3ADDR | Byte | 00 $FC48_{16}$ | Read/Write | | |

# A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|
| ACB3CTL2 | Byte | 00 FC4A$_{16}$ | Read/Write | 00$_{16}$ | |
| ACB3ADDR2 | Byte | 00 FC4C$_{16}$ | Read/Write | | |
| ACB3CTL3 | Byte | 00 FC4E$_{16}$ | Read/Write | 00$_{16}$ | |

### A.1.15 ACCESS.bus Interface (ACB) 4

(Section 4.13.8 on page 190)

| | | | | | |
|---|---|---|---|---|---|
| ACB4SDA | Byte | 00 FC60$_{16}$ | Read/Write | | |
| ACB4ST | Byte | 00 FC62$_{16}$ | Varies per bit | 00$_{16}$ | |
| ACB4CST | Byte | 00 FC64$_{16}$ | Varies per bit | 00$_{16}$ | |
| ACB4CTL1 | Byte | 00 FC66$_{16}$ | Read/Write | 00$_{16}$ | |
| ACB4ADDR | Byte | 00 FC68$_{16}$ | Read/Write | | |
| ACB4CTL2 | Byte | 00 FC6A$_{16}$ | Read/Write | 00$_{16}$ | |
| ACB4ADDR2 | Byte | 00 FC6C$_{16}$ | Read/Write | | |
| ACB4CTL3 | Byte | 00 FC6E$_{16}$ | Read/Write | 00$_{16}$ | |

### A.1.16 Analog Comparators Monitor (ACM)

(Section 4.14.5 on page 201)

| | | | | | |
|---|---|---|---|---|---|
| ACMCTS | Byte | 00 FD40$_{16}$ | Varies per bit | 00$_{16}$ | |
| ACMCNF | Byte | 00 FD42$_{16}$ | Read/Write | 00$_{16}$ | |
| ACMTIM | Byte | 00 FD44$_{16}$ | Read/Write | 37$_{16}$ | |
| THRDAT | Byte | 00 FD46$_{16}$ | Read/Write | 00$_{16}$ | |
| CMPRES | Byte | 00 FD48$_{16}$ | Read Only | | |
| VOLDAT0 | Byte | 00 FD50$_{16}$ | Read Only | | |
| VOLDAT1 | Byte | 00 FD52$_{16}$ | Read Only | | |
| VOLDAT2 | Byte | 00 FD54$_{16}$ | Read Only | | |
| VOLDAT3 | Byte | 00 FD56$_{16}$ | Read Only | | |
| VOLDAT4 | Byte | 00 FD58$_{16}$ | Read Only | | |
| VOLDAT5 | Byte | 00 FD5A$_{16}$ | Read Only | | |
| VOLDAT6 | Byte | 00 FD5C$_{16}$ | Read Only | | |
| VOLDAT7 | Byte | 00 FD5E$_{16}$ | Read Only | | |

### A.1.17 Power Management (PM)

(Section 4.17.4 on page 210)

# A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|
| PMCSR | Byte | 00 $FF80_{16}$ | Read/Write | $00_{16}$ | |

### A.1.18 High Frequency Clock Generator (HFCG)

(Section 4.18.7 on page 216)

| | | | | | |
|---|---|---|---|---|---|
| HFCGCTRL1 | Byte | 00 $FFA0_{16}$ | Varies per bit | $0C_{16}$ | |
| HFCGML | Byte | 00 $FFA2_{16}$ | Read/Write | $CF_{16}$ | |
| HFCGMH | Byte | 00 $FFA4_{16}$ | Read/Write | $03_{16}$ | |
| HFCGN | Byte | 00 $FFA6_{16}$ | Read/Write | $08_{16}$ | |
| HFCGIL | Byte | 00 $FFA8_{16}$ | Read/Write | | |
| HFCGIH | Byte | 00 $FFAA_{16}$ | Read/Write | | |
| HFCGP | Byte | 00 $FFAC_{16}$ | Read/Write | $17_{16}$ | |
| HFCGCTRL2 | Byte | 00 $FFAE_{16}$ | Varies per bit | $00_{16}$ | |

### A.1.19 Development System Support

(Section 4.20.8 on page 239)

| | | | | | |
|---|---|---|---|---|---|
| DBGFRZEN2 | Byte | 00 $FF14_{16}$ | Read/Write | $FF_{16}$ | |
| DBGCFG | Byte | 00 $FF16_{16}$ | Read/Write | $00_{16}$ | |
| DBGFRZEN | Byte | 00 $FF18_{16}$ | Read/Write | $FF_{16}$ | |

### A.1.20 Multi-Input Wake-Up (MIWU)

(Section 4.4.3 on page 106)

| | | | | | |
|---|---|---|---|---|---|
| WKEDG1 | Byte | 00 $FFC0_{16}$ | Read/Write | $00_{16}$ | |
| WKEDG2 | Byte | 00 $FFC2_{16}$ | Read/Write | $00_{16}$ | |
| WKEDG3 | Byte | 00 $FFC4_{16}$ | Read/Write | $00_{16}$ | |
| WKEDG4 | Byte | 00 $FFC6_{16}$ | Read/Write | $00_{16}$ | |
| WKPND1 | Byte | 00 $FFC8_{16}$ | Read/Write | $00_{16}$ | |
| WKPCL1 | Byte | 00 $FFCA_{16}$ | Write Only | | |
| WKPND2 | Byte | 00 $FFCC_{16}$ | Read/Write | $00_{16}$ | |
| WKPCL2 | Byte | 00 $FFCE_{16}$ | Write Only | | |
| WKPND3 | Byte | 00 $FFD0_{16}$ | Read/Write | $00_{16}$ | |
| WKPCL3 | Byte | 00 $FFD2_{16}$ | Write Only | | |
| WKPND4 | Byte | 00 $FFD4_{16}$ | Read/Write | $00_{16}$ | |
| WKPCL4 | Byte | 00 $FFD6_{16}$ | Write Only | | |

# A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|
| WKEN1 | Byte | 00 FFD8$_{16}$ | Read/Write | 00$_{16}$ | |
| WKEN2 | Byte | 00 FFDA$_{16}$ | Read/Write | 00$_{16}$ | |
| WKEN3 | Byte | 00 FFDC$_{16}$ | Read/Write | 00$_{16}$ | |
| WKEN4 | Byte | 00 FFDE$_{16}$ | Read/Write | 00$_{16}$ | |

### A.1.21 Interrupt Control Unit (ICU)

(Section 4.3.4 on page 99)

| | | | | | |
|---|---|---|---|---|---|
| IVCT | Byte | 00 FE00$_{16}$ | Read Only | 10$_{16}$ | |
| NMISTAT | Byte | 00 FE02$_{16}$ | Read Only | 00$_{16}$ | |
| PFAIL | Byte | 00 FE04$_{16}$ | Read/Write | 00$_{16}$ | |
| ISTAT0 | Word | 00 FE0A$_{16}$ | Read Only | 0000$_{16}$ | |
| ISTAT1 | Word | 00 FE0C$_{16}$ | Read Only | 0000$_{16}$ | |
| IENAM0 | Word | 00 FE0E$_{16}$ | Read/Write | 0000$_{16}$ | |
| IENAM1 | Word | 00 FE10$_{16}$ | Read/Write | 0000$_{16}$ | |
| IECLR0 | Word | 00 FE12$_{16}$ | Write Only | | |
| IECLR1 | Word | 00 FE14$_{16}$ | Write Only | | |

### A.1.22 Debugger Interface

(Section 4.19.7 on page 231)

| | | | | | |
|---|---|---|---|---|---|
| DBGRXD0 | Word | 00 FDC0$_{16}$ | Read Only | | |
| DBGRXD2 | Word | 00 FDC2$_{16}$ | Read Only | | |
| DBGRXD4 | Word | 00 FDC4$_{16}$ | Read Only | | |
| DBGRXD6 | Word | 00 FDC6$_{16}$ | Read Only | | |
| DBGRXD8 | Word | 00 FDC8$_{16}$ | Read Only | | |
| DBGRXD10 | Word | 00 FDCA$_{16}$ | Read Only | | |
| DBGRXD12 | Word | 00 FDCC$_{16}$ | Read Only | | |
| DBGRXD14 | Word | 00 FDCE$_{16}$ | Read Only | | |
| DBGTXD0 | Word | 00 FDD0$_{16}$ | Read/Write | | |
| DBGTXD2 | Word | 00 FDD2$_{16}$ | Read/Write | | |
| DBGTXD4 | Word | 00 FDD4$_{16}$ | Read/Write | | |
| DBGTXD6 | Word | 00 FDD6$_{16}$ | Read/Write | | |
| DBGTXD8 | Word | 00 FDD8$_{16}$ | Read/Write | | |
| DBGTXD10 | Word | 00 FDDA$_{16}$ | Read/Write | | |

# A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|
| DBGTXD12 | Word | 00 $FDDC_{16}$ | Read/Write | | |
| DBGTXD14 | Word | 00 $FDDE_{16}$ | Read/Write | | |
| DBGRXST | Byte | 00 $FDE0_{16}$ | Varies per bit | | |
| DBGTXST | Byte | 00 $FDE2_{16}$ | Read/Write | $00_{16}$ | |
| DBGTXLOC | Byte | 00 $FDE4_{16}$ | Read/Write | $0F_{16}$ | |
| DBGTINT | Byte | 00 $FDE6_{16}$ | Write Only | | |
| DBGABORT | Word | 00 $FDE8_{16}$ | Write Only | | |
| DBGISESRCA | Word | 00 $FDEA_{16}$ | Read/Write | $0000_{16}$ | |

### A.1.23 Pulse Width Modulator (PWM)

(Section 4.8.5 on page 144)

| | | | | | |
|---|---|---|---|---|---|
| PRSC | Word/Byte | 00 $FD00_{16}$ | Read/Write | $0000_{16}$ | |
| CTR | Word/Byte | 00 $FD02_{16}$ | Read/Write | $FFFF_{16}$ | |
| PWMPOL | Byte | 00 $FD04_{16}$ | Read/Write | $00_{16}$ | |
| PWMCNT | Byte | 00 $FD06_{16}$ | Read/Write | $00_{16}$ | |
| DCR0 | Word/Byte | 00 $FD08_{16}$ | Read/Write | $0000_{16}$ | |
| DCR1 | Word/Byte | 00 $FD0A_{16}$ | Read/Write | $0000_{16}$ | |
| DCR2 | Word/Byte | 00 $FD0C_{16}$ | Read/Write | $0000_{16}$ | |
| DCR3 | Word/Byte | 00 $FD0E_{16}$ | Read/Write | $0000_{16}$ | |
| DCR4 | Word/Byte | 00 $FD10_{16}$ | Read/Write | $0000_{16}$ | |
| DCR5 | Word/Byte | 00 $FD12_{16}$ | Read/Write | $0000_{16}$ | |
| DCR6 | Word/Byte | 00 $FD14_{16}$ | Read/Write | $0000_{16}$ | |
| DCR7 | Word/Byte | 00 $FD16_{16}$ | Read/Write | $0000_{16}$ | |

### A.1.24 Universal Synchronous/Asynchronous Receiver Transmitter (USART) 1

(Section 4.9.4 on page 154)

| | | | | | |
|---|---|---|---|---|---|
| U1TBUF | Byte | 00 $FD20_{16}$ | Read/Write | | |
| U1RBUF | Byte | 00 $FD22_{16}$ | Read Only | | |
| U1ICTRL | Byte | 00 $FD24_{16}$ | Varies per bit | $01_{16}$ | |
| U1STAT | Byte | 00 $FD26_{16}$ | Read Only | $00_{16}$ | |
| U1FRS | Byte | 00 $FD28_{16}$ | Read/Write | $00_{16}$ | |
| U1MDSL | Byte | 00 $FD2A_{16}$ | Read/Write | $00_{16}$ | |
| U1BAUD | Byte | 00 $FD2C_{16}$ | Read/Write | $00_{16}$ | |

# A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|
| U1PSR | Byte | 00 FD2E$_{16}$ | Read/Write | 00$_{16}$ | |

### A.1.25 Universal Synchronous/Asynchronous Receiver Transmitter (USART) 2

(Section 4.9.4 on page 154)

| | | | | | |
|---|---|---|---|---|---|
| U2TBUF | Byte | 00 FC20$_{16}$ | Read/Write | | |
| U2RBUF | Byte | 00 FC22$_{16}$ | Read Only | | |
| U2ICTRL | Byte | 00 FC24$_{16}$ | Varies per bit | 01$_{16}$ | |
| U2STAT | Byte | 00 FC26$_{16}$ | Read Only | 00$_{16}$ | |
| U2FRS | Byte | 00 FC28$_{16}$ | Read/Write | 00$_{16}$ | |
| U2MDSL | Byte | 00 FC2A$_{16}$ | Read/Write | 00$_{16}$ | |
| U2BAUD | Byte | 00 FC2C$_{16}$ | Read/Write | 00$_{16}$ | |
| U2PSR | Byte | 00 FC2E$_{16}$ | Read/Write | 00$_{16}$ | |

### A.1.26 Shared Memory Core

(Section 5.3.8 on page 271)

| | | | | | |
|---|---|---|---|---|---|
| SMCCST | Byte | 00 F900$_{16}$ | Read/Write | 00$_{16}$ | |
| SMCTA | Byte | 00 F902$_{16}$ | Read/Write and Read Only | | See description |
| SMHSEM | Byte | 00 F904$_{16}$ | Varies per bit | 00$_{16}$ | |
| SMCORP0 | Word | 00 F910$_{16}$ | Varies per bit | | See description |
| SMCORP1 | Word | 00 F912$_{16}$ | Varies per bit | | See description |
| SMCORP2 | Word | 00 F914$_{16}$ | Varies per bit | | See description |
| SMCOWP0 | Word | 00 F920$_{16}$ | Read/Write | FFFF$_{16}$ | |
| SMCOWP1 | Word | 00 F922$_{16}$ | Read/Write | FFFF$_{16}$ | |
| SMCOWP2 | Word | 00 F924$_{16}$ | Read/Write | FFFF$_{16}$ | |

### A.1.27 Core Access to SuperI/O

(Section 5.4.1 on page 276

| | | | | | |
|---|---|---|---|---|---|
| IHIOA | Word | 00 FCE0$_{16}$ | Read/Write | 00$_{16}$ | |
| IHD | Byte | 00 FCE2$_{16}$ | Read/Write | 00$_{16}$ | |
| LKSIOHA | Word | 00 FCE4$_{16}$ | Read/Write | 0002$_{16}$/0000$_{16}$ | |
| SIOLV | Word | 00 FCE6$_{16}$ | R/W1C | 0000$_{16}$ | |
| CRSMAE | Word | 00 FCE8$_{16}$ | Read/Write | 0000$_{16}$ | |
| SIBCTRL | Byte | 00 FCEA$_{16}$ | Varies per bit | 00$_{16}$ | |

# A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|

### A.1.28 Mobile System Wake-Up Control (MSWC)

(Section 5.5.6 on page 291)

| | | | | | |
|---|---|---|---|---|---|
| MSWCTL1 | Byte | 00 FCC0$_{16}$ | Varies per bit | 00$_{16}$ | Vcc power-up only |
| MSWCTL2 | Byte | 00 FCC2$_{16}$ | Varies per bit | 00$_{16}$ | |
| MSWCTL3 | Byte | 00 FCC4$_{16}$ | Varies per bit | 01$_{16}$ | $V_{PP}$ power-up only |
| HCFGBAL | Byte | 00 FCC8$_{16}$ | Read/Write | 00$_{16}$ | |
| HCFGBAH | Byte | 00 FCCA$_{16}$ | Read/Write | 00$_{16}$ | |
| MSIEN2 | Byte | 00 FCCC$_{16}$ | Read/Write | 00$_{16}$ | |
| MSHES0 | Byte | 00 FCCE$_{16}$ | R/W1C | 00$_{16}$ | |
| MSHEIE0 | Byte | 00 FCD0$_{16}$ | Read/Write | 00$_{16}$ | |

# A. Register List (Continued)

## A.2 HOST DOMAIN REGISTERS

### A.2.1 Configuration Registers

Access to all host configuration registers is via an index/data scheme that uses the host configuration index/data pair.

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|

**Common SuperI/O Configuration**

(Section 6.1.8 on page 306)

| | | | | | |
|---|---|---|---|---|---|
| SID | Byte | Index $20_{16}$ | Read Only | $EC_{16}$ | |
| SIOCF1 | Byte | Index $21_{16}$ | Varies per bit | $11_{16}$ | |
| SIOCF5 | Byte | Index $25_{16}$ | Read/Write | $00_{16}$ | |
| SIOCF6 | Byte | Index $26_{16}$ | Read/Write | $00_{16}$ | |
| SRID | Byte | Index $27_{16}$ | Read Only | | |
| SIOCF8 | Byte | Index $28_{16}$ | Read/Write | $00_{16}$ | |
| SIOCF9 | Byte | Index $29_{16}$ | Read/Write | $01_{16}$ | |
| SIOCFD | Byte | Index $2D_{16}$ | Read/Write | $00_{16}$ | |

**Shared Memory**

(Section 6.1.11 on page 311)

| | | | | | |
|---|---|---|---|---|---|
| Shared Memory Configuration | Byte | Index $F4_{16}$ | Read/Write | $00_{16}$ or $09_{16}$ depending on $\overline{SHBM}$ strap | |
| Shared Memory Base Address High Byte | Byte | Index $F5_{16}$ | Read/Write | $00_{16}$ | |
| Shared Memory Base Address Low Byte | Byte | Index $F6_{16}$ | Read/Write | $00_{16}$ | |
| Shared Memory Size Config | Byte | Index $F7_{16}$ | Read/Write | $00_{16}$ | |

**RTC Configuration**

When LDN is set to $10_{16}$. (Section 6.1.12 on page 315)

| | | | | | |
|---|---|---|---|---|---|
| RLR | Byte | Device Specific | Read/Write | $00_{16}$ | Cleared by H/W reset only |
| DOMAO | Byte | Index $F0_{16}$ | Read/Write | $00_{16}$ | |
| MONAO | Byte | Index $F1_{16}$ | Read/Write | $00_{16}$ | |
| CENO | Byte | Index $F3_{16}$ | Read/Write | $00_{16}$ | |

# A. Register List (Continued)

### A.2.2 Host Runtime Registers

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|

**Shared Memory Host**

The base address is defined by LDN $0F_{16}$ (Section 5.3.7 on page 268).

| | | | | | |
|---|---|---|---|---|---|
| SMIMA0 | Byte | Offset $00_{16}$ | Read/Write | | |
| SMIMA1 | Byte | Offset $01_{16}$ | Read/Write | | |
| SMIMA2 | Byte | Offset $02_{16}$ | Read/Write | | |
| SMIMA3 | Byte | Offset $03_{16}$ | Read/Write | | |
| SMIMD | Byte | Offset $04_{16}$ | Read/Write | | |
| SMHAP1 | Byte | Offset $07_{16}$ | Varies per bit | $02_{16}$ | |
| SMHAP2 | Byte | Offset $08_{16}$ | Varies per bit | $02_{16}$ | |
| SMHSEM | Byte | Offset $0C_{16}$ | Varies per bit | $00_{16}$ | |

**MSWC Host Registers**

The base address is defined by LDN $04_{16}$ (Section 5.5.5 on page 285).

| | | | | | |
|---|---|---|---|---|---|
| WK_STS0 | Byte | Offset $00_{16}$ | R/W1C | $00_{16}$ | |
| WK_EN0 | Byte | Offset $02_{16}$ | Read/Write | $00_{16}$ | |
| WK_CFG | Byte | Offset $04_{16}$ | Read/Write | $00_{16}$ | |
| WK_SIGV | Byte | Offset $06_{16}$ | Read Only | | |
| WK_STATE | Byte | Offset $07_{16}$ | Read/Write | | |
| WK_SMIEN0 | Byte | Bank 2 Offset $13_{16}$ | Read/Write | $00_{16}$ | |
| WK_IRQEN0 | Byte | Bank 2 Offset $15_{16}$ | Read/Write | $00_{16}$ | |

**Host Interface (HI)**

The base address is defined by LDN $06_{16}$ (“Host Addresses” on page 242).

| | | | | | |
|---|---|---|---|---|---|
| DBBOUT | Byte | Defined in LDN $06_{16}$ index $60_{16}$, $61_{16}$ | R | | |
| STATUS | Byte | Defined in LDN $06_{16}$ index $62_{16}$, $63_{16}$ | R | $00_{16}$ | |
| DBBIN | Byte | Defined in LDN $06_{16}$ index $60_{16}$, $61_{16}$ | W | | |

## A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|
| COMAND | Byte | Defined in LDN $06_{16}$ index $62_{16}$, $63_{16}$ | W | | |

### Power Management Channel 1

The base address is defined by LDN $11_{16}$ (“Host Addresses” on page 242).

| | | | | | |
|---|---|---|---|---|---|
| DBBOUT | Byte | Defined in LDN $11_{16}$ index $60_{16}$, $61_{16}$ | R | | |
| STATUS | Byte | Defined in LDN $11_{16}$ index $62_{16}$, $63_{16}$ | R | $00_{16}$ | |
| DBBIN | Byte | Defined in LDN $11_{16}$ index $60_{16}$, $61_{16}$ | W | | |
| COMAND | Byte | Defined in LDN $11_{16}$h index $62_{16}$, $63_{16}$ | W | | |

### Power Management Channel 2

The base address is defined by LDN $12_{16}$ (“Host Addresses” on page 242).

| | | | | | |
|---|---|---|---|---|---|
| DBBOUT | Byte | Defined in LDN $12_{16}$ index $60_{16}$, $61_{16}$ | R | | |
| STATUS | Byte | Defined in LDN $12_{16}$ index $62_{16}$, $63_{16}$ | R | $00_{16}$ | |
| DBBIN | Byte | Defined in LDN $12_{16}$ index $60_{16}$, $61_{16}$ | W | | |
| COMAND | Byte | Defined in LDN $12_{16}$ index $62_{16}$, $63_{16}$ | W | | |

# A. Register List (Continued)

| Register Name | Size | Register Address | Access Type | Value After Reset | Comments |
|---|---|---|---|---|---|

**RTC**

Access is via a index/data scheme that uses an index/data pair pointed to by LDN $10_{16}$ (Section 6.2.15 on page 325).

| | | | | | |
|---|---|---|---|---|---|
| SEC | Byte | Index $00_{16}$ | Read/Write | $00_{16}$ | |
| SECA | Byte | Index $01_{16}$ | Read/Write | $00_{16}$ | |
| MIN | Byte | Index $02_{16}$ | Read/Write | $00_{16}$ | |
| MINA | Byte | Index $03_{16}$ | Read Only | $00_{16}$ | |
| HOR | Byte | Index $04_{16}$ | Read/Write | $00_{16}$ | |
| HORA | Byte | Index $05_{16}$ | Read/Write | $00_{16}$ | |
| DOW | Byte | Index $06_{16}$ | Read/Write | $00_{16}$ | |
| DOM | Byte | Index $07_{16}$ | Read/Write | $00_{16}$ | |
| MON | Byte | Index $08_{16}$ | Read/Write | $00_{16}$ | |
| YER | Byte | Index $09_{16}$ | Read/Write | $00_{16}$ | |
| CRA | Byte | Index $0A_{16}$ | Read/Write | $20_{16}$ | |
| CRB | Byte | Index $0B_{16}$ | Read/Write | $00_{16}$ | |
| CRC | Byte | Index $0C_{16}$ | Read Only | $00_{16}$ | |
| CRD | Byte | Index $0D_{16}$ | Read Only | $00_{16}$ | |
| DOMA | Byte | Prog. Index | Read/Write | $C0_{16}$ | |
| MONA | Byte | Prog. Index | Read/Write | $C0_{16}$ | |
| CEN | Byte | Prog. Index | Read/Write | $00_{16}$ | |

## A.3 CORE DOMAIN REGISTER LAYOUT

### A.3.1 Module Configuration

(Section 2.3 on page 48 and Section 2.4.2 on page 52)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| MCFG/MCFGSH | GTMON | HOSTWAIT | ENZONE2 | CLKOM | | EXMEM16 | ENEMEM | ENEIO |
| EICFG | Reserved | | | | | | EXWINT46 | EXWINT45 |
| IOEE1 | Reserved | | | EEPA4 | EEPA3 | EEPA2 | EEPA1 | EEPA0 |
| IOEE2 | Reserved | | | EEPC0 | EEPD3 | EEPB2 | EEPB1 | EEPB0 |
| PTWRL | Reserved | Force MBTA Zero | RTC Lock Default | Host Boot Block | Core Boot Block | | | |
| PTWRH | RAM Size | RST2EN | | Reserved | | | Zone 2 Memory Range | |
| PNMR | Reserved | | | | Reserved | A20 | A19 | ENUSART2 |
| STRPST (p. 48) | Reserved | Reserved | Reserved | Reserved | Reserved | BADDR1 | BADDR0 | SHBM |

# A. Register List (Continued)

### A.3.2 Bus Interface Unit (BIU)

(Section 4.1.10 on page 81)

| | 15 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BCFG | N/A | | | | | | Reserved | | | | | ISTL | OBR | EWR |
| IOCFG | Reserved | | | | IPST | Res | BW | Reserved | | HOLD | | WAIT | | |
| SZCFGn | Reserved | | FRE | IPRE | IPST | Res | BW | WBR | BRE | HOLD | | WAIT | | |

### A.3.3 DMA Controller

(Section 4.2.8 on page 90)

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADCAn | Device A Address Counter | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ADRAn | Device A Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ADCBn | Device B Address Counter | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ADRBn | Device B Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BLTCn | Reserved | | | | | | | | | | | | | | | | Block Length Counter | | | | | | | | | | | | | | | |
| BLTRn | Reserved | | | | | | | | | | | | | | | | Block Length | | | | | | | | | | | | | | | |
| DMACNTLn | N/A | | | | | | | | | | | | | | | | Res | INCB | ADB | INCA | ADA | SWRQ | BPC | OT | DIR | IND | TCS | EOVR | ETC | CHEN | | |
| DMASTATn | N/A | | | | | | | | | | | | | | | | | | | | | | Reserved | | | | VLD | CHAC | OVR | TC | | |

### A.3.4 General-Purpose I/O (GPIO) Port

(Section 4.5.6 on page 116)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PADIR | PA Port Direction | | | | | | | |
| PBDIR | PB Port Direction | | | | | | | |
| PCDIR | PC Port Direction | | | | | | | |
| PDDIR | PD Port Direction | | | | | | | |
| PFDIR | PF Port Direction | | | | | | | |
| PJDIR | PJ Port Direction | | | | | | Reserved | |
| PLDIR | Reserved | | | PL Port Direction | | Reserved | | |
| PMDIR | PM Port Direction | | | | | | | |
| PQDIR | Reserved | | | | PQ Port Direction | | | |
| PADIN | PA Port Input Data | | | | | | | |
| PBDIN | PB Port Input Data | | | | | | | |
| PCDIN | PC Port Input Data | | | | | | | |
| PDDIN | PD Port Input Data | | | | | | | |
| PEDIN | PE Port Input Data | | | | | | | |
| PFDIN | PF Port Input Data | | | | | | | |
| PJDIN | PJ Port Input Data | | | | | | Reserved | |
| PLDIN | Reserved | | | PL Port Input Data | | Reserved | | |
| KBSIN | KBS Port Input Data | | | | | | | |
| PMDIN | PM Port Input Data | | | | | | | |
| PQDIN | Reserved | | | | | PQ Port Input Data | | |
| PADOUT | PA Port Output Data | | | | | | | |
| PBDOUT | PB Port Output Data | | | | | | | |
| PCDOUT | PC Port Output Data | | | | | | | |
| PDDOUT | PD Port Output Data | | | | | | | |
| PFDOUT | PF Port Output Data | | | | | | | |
| PJDOUT | PJ Port Output Data | | | | | | Reserved | |
| PLDOUT | Reserved | | | PL Port Output Data | | Reserved | | |
| KBSOUT | KBS Port Output Data | | | | | | | |

# A. Register List (Continued)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PMDOUT | PM Port Output Data | | | | | | | |
| PQDOUT | Reserved | | | | PQ Port Output Data | | | |
| PAWPU | PA Port Weak Pull-up Enable | | | | | | | |
| PBWPU | PB Port Weak Pull-up Enable | | | | | | | |
| PCWPU | PC Port Weak Pull-up Enable | | | | | | | |
| PDWPU | PD Port Weak Pull-up Enable | | | | | | | |
| PEWPU | PE Port Weak Pull-Up Enable | | Reserved | PE Port Weak Pull-Up Enable | Reserved | | | |
| PFWPU | PF Port Weak Pull-up Enable | | | | | | | |
| KBSINPU | KBS Weak Pull-up Enable | | | | | | | |
| PQWPU | Reserved | | | | | PQ Port Weak Pull-up Enable | | |
| PAALT | PA Pins Alternate Function Enable | | | | | | | |
| PBALT | PB Pins Alternate Function Enable | | | | | | | |
| PCALT | PC Pins Alternate Function Enable | | | | | | | |
| PDALT | PD Pins Alternate Function Enable | | | | | | | |
| PEALT | PE Pins Alternate Function Enable | | | | | | | |
| PFALT | PF Pins Alternate Function Enable | | | | | | | |
| PQALT | Reserved | | | | PQ Pins Alt. Function Enable | | | |

### A.3.5 PS/2 Interface

(Section 4.6.5 on page 125)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PSDAT | Data | | | | | | | |
| PSTAT | Reserved | RFERR | ACH | | | PERR | EOT | SOT |
| PSCON | WPUEN | IDB | | | HDRV | | XMT | EN |
| PSOSIG | CLK4 | WDAT4 | CLK3 | CLK2 | CLK1 | WDAT3 | WDAT2 | WDAT1 |
| PSISIG | RCLK4 | RDAT4 | RCLK3 | RCLK2 | RCLK1 | RDAT3 | RDAT2 | RDAT1 |
| PSIEN | Reserved | | | | | DSMIE | EOTIE | SOTIE |

### A.3.6 Core Interface

(Section 5.1.4 on page 247 and Section 5.2.3 on page 256)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| HICTRL | Reserved | PMICIE | PMOCIE | PMHIE | IBFCIE | OBECIE | OBFMIE | OBFKIE |
| HIIRQC | Reserved | IRQNPOL | IRQM | | | IRQ11B | IRQ12BOBFMIE | IRQ1B |
| HIKMST | ST3 | ST2 | ST1 | ST0 | A2 | F0 | IBF | OBF |
| HIKDO | Keyboard DBBOUT Data | | | | | | | |
| HIMDO | Mouse DBBOUT Data | | | | | | | |
| HIKMDI | Keyboard/Mouse DBBIN Data | | | | | | | |
| HIPMnST | ST3 | ST2 | ST1 | ST0 | A2 | F0 | IBF | OBF |
| HIPMnDO | PM Channel DBBOUT Data | | | | | | | |
| HIPMnDOC | PM Channel DBBOUT Data | | | | | | | |
| HIPMnDOM | PM Channel DBBOUT Data | | | | | | | |
| HIPMnDI | PM Channel DBBIN Data | | | | | | | |
| HIPMnDIC | PM Channel DBBIN Data | | | | | | | |
| HIPMnCTRL | EME | SCIPOL | PLMS | | | Reserved | OBEIE | IBFIE |
| HIPMnIC | SCIIS | SMIPOL | PLMM | | | SCIB | SMIB | IRQB |
| HIPMnIE | Reserved | | HSMIE | HSCIE | HIRQE | SMIE | SCIE | IRQE |

# A. Register List (Continued)

### A.3.7 Multi-Function Timer (MFT16)

(Section 4.7.7 on page 137)

| | 15 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| TnCNT1 | TCNT1 | | | | | | | | |
| TnCRA | TCRA | | | | | | | | |
| TnCRB | TCRB | | | | | | | | |
| TnCNT2 | TCNT2 | | | | | | | | |
| TnPRSC | N/A | Reserved | | | CLKPS | | | | |
| TnCKC | N/A | Reserved | | C2CSEL | | | C1CSEL | | |
| TnCTRL | N/A | Reserved | TAOUT | TBEN | TAEN | TBEDG | TAEDG | MDSEL | |
| TnICTL | N/A | TDIEN | TCIEN | TBIEN | TAIEN | TDPND | TCPND | TBPND | TAPND |
| TnICLR | N/A | Reserved | | | | TDCLR | TCCLR | TBCLR | TACLR |

### A.3.8 Timing and Watchdog (TWD)

(Section 4.10.3 on page 162)

| | 15 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| TWCFG | N/A | Reserved | | WDSDME | WDCT0I | LWDCNT | LTWDT0 | LTWCP | LTWCFG |
| TWCP | N/A | Reserved | | | | | MDIV | | |
| TWDT0 | Preset | | | | | | | | |
| T0CSR | N/A | Reserved | | | | WDLTD | Reserved | TC | RST |
| WDCNT | N/A | PRESET | | | | | | | |
| WDSDM | N/A | RSDATA | | | | | | | |

### A.3.9 Analog to Digital Converter (ADC)

(Section 4.11.5 on page 171)

| | 15 10 | 9 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| ADCSTS | N/A | | Reserved | | | | | | OVFEV | EOCEV |
| ADCCNF | N/A | | Reserved | | | | | INTE-CEN | Res | ADCEN |
| ACLKCTL | N/A | | Reserved | | SCLKDIV | | | | | |
| ADLYCTL | N/A | | Reserved | | | | | VOLDLY | | |
| ADCPINX | N/A | | Index | | | | | | | |
| ADCPD | Parameter Data | | | | | | | | | |
| VCHN1CTL | N/A | | DATVAL | CSCALE | INTDVEN | SELIN | | | | |
| VCHN1DAT | Reserved | VCHDAT | | | | | | | Reserved | |
| VCHN2CTL | N/A | | DATVAL | CSCALE | INTDVEN | SELIN | | | | |
| VCHN2DAT | Reserved | VCHDAT | | | | | | | Reserved | |
| VCHN3CTL | N/A | | DATVAL | CSCALE | INTDVEN | SELIN | | | | |
| VCHN3DAT | Reserved | VCHDAT | | | | | | | Reserved | |

### A.3.10 Digital to Analog (DAC)

(Section 4.12.5 on page 181)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DACCTRL | Reserved | | | ENIDLE | DACEN3 | DACEN2 | DACEN1 | DACEN0 |
| DACDATi | DAC DATAi | | | | | | | |

### A.3.11 ACCESS.bus Interface (ACB)

(Section 4.13.8 on page 190)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ACBnSDA | DATA | | | | | | | |
| ACBnST | SLVSTP | SDAST | BER | NEGACK | STASTR | NMATCH | MASTER | XMIT |
| ACBnCST | ARP-MATCH | MATCHAF | TGSCL | TSDA | GMATCH | MATCH | BB | BUSY |
| ACB(1,2)CTL1 | STASTRE | NMINTE | GCMEN | ACK | Reserved | INTEN | STOP | START |
| ACB(3,4)CTL1 | STASTRE | NMINTE | GCMEN | ACK | DMAEN | INTEN | STOP | START |

# A. Register List (Continued)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ACBnADDR | SAEN | ADDR | | | | | | |
| ACBnCTL2 | SCLFRQ6 | SCLFRQ5 | SCLFRQ4 | SCLFRQ3 | SCLFRQ2 | SCLFRQ1 | SCLFRQ0 | ENABLE |
| ACBnCTL3 | Reserved | | | | | ARPMEN | SCLFRQ8-7 | |
| ACBnADDR2 | SAEN | ADDR | | | | | | |

### A.3.12 Analog Comparators Monitor (ACM)

(Section 4.14.5 on page 201)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ACMCTS | Reserved | | | | OVUNTHEV | EOCEV | EOMEV | START |
| ACMCNF | Reserved | | OVUNSEL | INTOUEN | INTEMEN | Reserved | ACMMOD | |
| ACMTIM | Reserved | | T0DIV | | Reserved | SMPDLY | | |
| THRDAT | Reserved | | THRSHD5 | THRSHD4 | THRSHD3 | THRSHD2 | THRSHD1 | THRSHD0 |
| CMPRES | CMPIN7 | CMPIN6 | CMPIN5 | CMPIN4 | CMPIN3 | CMPIN2 | CMPIN1 | CMPIN0 |
| VOLDAT0-7 | Reserved | | Voltage Level Data 5-0 | | | | | |

### A.3.13 Power Management (PM)

(Section 4.17.4 on page 210)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PMCSR | OLFC | OHFC | WBPSM | Reserved | | IDLE | DHF | Reserved |

### A.3.14 High-Frequency Clock Generator (HFCG)

(Section 4.18.7 on page 216)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| HFCGCTRL1 | Reserved | FAST96 | LOAD96 | IVLID | OHFC | PENABLE | FAST | LOAD |
| HFCGML | HFCGM7-0 | | | | | | | |
| HFCGMH | HFCGM | | | | | | | |
| HFCGN | Reserved | | | HFCGN4-0 | | | | |
| HFCGIL | HFCGI7-0 | | | | | | | |
| HFCGIH | Reserved | | HFCGI13-8 | | | | | |
| HFCGP | Reserved | | | HFCGP4-0 | | | | |
| HFCGCTRL2 | Reserved | | | 96MON | MONERR | SCESTP | SCESTR | SENABLE |

### A.3.15 Development System Support

(Section 4.20.8 on page 239)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DBGFRZEN2 | Reserved | | | | | USART2FEN | ACB4FEN | ACB3FEN |
| DBGCFG | Reserved | | | | | BRKLE | FREEZE | ON |
| DBGFRZEN | Reserved | | HIFEN | USARTFEN | ACB2FEN | ACB1FEN | MFT2FEN | MFT1FEN |

### A.3.16 Multi-Input Wake-Up (MIWU)

(Section 4.4.3 on page 106)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| WKEDG1 | WKED17 | WKED16 | WKED15 | WKED14 | WKED03 | WKED12 | WKED11 | WKED10 |
| WKEDG2 | WKED27 | WKED26 | WKED25 | WKED24 | WKED23 | WKED22 | WKED21 | WKED20 |
| WKEDG3 | WKED37 | WKED36 | WKED35 | WKED34 | WKED33 | WKED32 | WKED31 | WKED30 |
| WKEDG4 | WKED47 | WKED46 | WKED45 | WKED44 | WKED43 | WKED42 | WKED41 | WKED40 |
| WKPND1 | WKPD17 | WKPD16 | WKPD15 | WKPD14 | WKPD03 | WKPD12 | WKPD11 | WKPD10 |
| WKCL1 | WKCL17 | WKCL16 | WKCL15 | WKCL14 | WKCL03 | WKCL12 | WKCL11 | WKCL10 |
| WKPND2 | WKPD27 | WKPD26 | WKPD25 | WKPD24 | WKPD23 | WKPD22 | WKPD21 | WKPD20 |
| WKCL2 | WKCL27 | WKCL26 | WKCL25 | WKCL24 | WKCL23 | WKCL22 | WKCL21 | WKCL20 |
| WKPND3 | WKPD37 | WKPD36 | WKPD35 | WKPD34 | WKPD33 | WKPD32 | WKPD31 | WKPD30 |
| WKCL3 | WKCL37 | WKCL36 | WKCL35 | WKCL34 | WKCL33 | WKCL32 | WKCL31 | WKCL30 |

# A. Register List (Continued)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| WKPND4 | WKPD47 | WKPD46 | WKPD45 | WKPD44 | WKPD43 | WKPD42 | WKPD41 | WKPD40 |
| WKCL4 | WKCL47 | WKCL46 | WKCL45 | WKCL44 | WKCL43 | WKCL42 | WKCL41 | WKCL40 |
| WKEN1 | WKEN17 | WKEN16 | WKEN15 | WKEN14 | WKEN13 | WKEN12 | WKEN11 | WKEN10 |
| WKEN2 | WKEN27 | WKEN26 | WKEN25 | WKEN24 | WKEN23 | WKEN22 | WKEN21 | WKEN20 |
| WKEN3 | WKEN37 | WKEN36 | WKEN35 | WKEN34 | WKEN33 | WKEN32 | WKEN31 | WKEN30 |
| WKEN4 | WKEN47 | WKEN46 | WKEN45 | WKEN44 | WKEN43 | WKEN42 | WKEN41 | WKEN40 |

### A.3.17 Interrupt Control Unit (ICU)

(Section 4.3.4 on page 99)

| | 15 12 | 11 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| IVCT | N/A | | 0 | 0 | INTVECT | | | | | |
| NMISTAT | N/A | | Reserved | | | | | | | EXT |
| PFAIL | N/A | | Reserved | | | | | ENLCK | PIN | EN |
| ISTAT0 | IST15-0 | | | | | | | | | |
| ISTAT1 | IST31-16 | | | | | | | | | |
| IENAM0 | IENA15-0 | | | | | | | | | |
| IENAM1 | IENA31-16 | | | | | | | | | |
| IECLR0 | IEC15-1 | | | | | | | | | Res |
| IECLR1 | IEC31-16 | | | | | | | | | |

### A.3.18 Debugger Interface

(Section 4.19.7 on page 231)

| | 15 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| DBGRXD0 | RX_Data0 | | | | | | | | |
| DBGRXD2 | RX_Data2 | | | | | | | | |
| DBGRXD4 | RX_Data4 | | | | | | | | |
| DBGRXD6 | RX_Data6 | | | | | | | | |
| DBGRXD8 | RX_Data8 | | | | | | | | |
| DBGRXD10 | RX_Data10 | | | | | | | | |
| DBGRXD12 | RX_Data12 | | | | | | | | |
| DBGRXD14 | RX_Data14 | | | | | | | | |
| DBGTXD0 | TX_Data0 | | | | | | | | |
| DBGTXD2 | TX_Data2 | | | | | | | | |
| DBGTXD4 | TX_Data4 | | | | | | | | |
| DBGTXD6 | TX_Data6 | | | | | | | | |
| DBGTXD8 | TX_Data8 | | | | | | | | |
| DBGTXD10 | TX_Data10 | | | | | | | | |
| DBGTXD12 | TX_Data12 | | | | | | | | |
| DBGTXD14 | TX_Data14 | | | | | | | | |
| DBGRXST | N/A | MSG_LEN | | | PID | | | | RX_BUSY |
| DBGTXST | N/A | Reserved | | | | | MSG_LEN | | |
| DBGTXLOC | N/A | Reserved | | | | PID | | | |
| DBGTINT | N/A | Reserved | | | | | | | ASSERT |
| DBGABORT | Reserved | | | | | | | | P_0 |
| DBGISESRCA | Reserved | | | | | | | ABORT_0 | RX_0 |

### A.3.19 Pulse with Modulator (PWM)

(Section 4.8.5 on page 144)

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PRSC | PRSC15-0 | | | | | | | | | | | | | | | |
| CTR | CTR15-0 | | | | | | | | | | | | | | | |
| PWMPOL | INVP7-0 | | | | | | | | | | | | | | | |

# A. Register List (Continued)

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PWMCNT | Reserved | | | | | | PWMRES | | Reserved | | | | | | PWR | |
| DCRi | DCRi15-0 | | | | | | | | | | | | | | | |

### A.3.20 Universal Synchronous/Asynchronous Receiver Transmitter (USART1 and USART2)

(Section 4.9.4 on page 154)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| UnTBUF | UTBUF | | | | | | | |
| UnRBUF | URBUF | | | | | | | |
| UnICTRL | EEI | ERI | ETI | Reserved | Reserved | Reserved | RBF | TBE |
| UnSTAT | Reserved | XMIP | RB9 | BKD | ERR | DOE | FE | PE |
| UnFRS | Reserved | PEN | PSEL | | XB9 | STP | CHAR | |
| U1MDSL | Reserved | Reserved | ERD | ETD | CKS | BRK | UnATN | MOD |
| U2MDSL | Reserved | Reserved | Reserved | | CKS | BRK | UnATN | MOD |
| UnBAUD | UDIV7-0 | | | | | | | |
| UnPSR | UPSC | | | | | UDIV(10-8) | | |

### A.3.21 Shared Memory Core

(Section 5.3.8 on page 271)

| | 15 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| SMCCST | N/A | HSEMIE | HSEMW | HLOCK | HERES | | HERRIEN | HWERR | HRERR |
| SMCTA | N/A | Reserved | | | | MBSD | | | |
| SMHSEM | N/A | CSEM3 | CSEM2 | CSEM1 | CSEM0 | HSEM3 | HSEM2 | HSEM1 | HSEM0 |
| SMCOHRP0 | ORPLA15-0 | | | | | | | | |
| SMCOHRP1 | ORP(15-2) | | | | | | | Reserved | |
| SMCOHRP2 | ORP31-16 | | | | | | | | |
| SMCOHWP0 | OWPLA15-0 | | | | | | | | |
| SMCOHWP1 | OWP15-2 | | | | | | | Reserved | |
| SMCOHWP2 | OWP31-16 | | | | | | | | |

### A.3.22 Core Access to SuperI/O Modules

(Section 5.4.1 on page 276)

| | 15 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| IHIOA | Reserved | | Indirect Host I/O Offset | | | | | | | |
| IHD | N/A | | Indirect Host Data | | | | | | | |
| LKSIOHA | Reserved | | | | | | | | LKRTCHA | LKCFG |
| SIOLV | Reserved | | | | | | | | RTCLV | CFGLV |
| CRSMAE | Reserved | MSWCAE | Reserved | | | | | | RTCAE | CFGAE |
| SIBCTRL | N/A | | Reserved | | | | RTCMR | CSWR | CSRD | CSAE |

### A.3.23 MSWC

(Section 5.5.6 on page 291)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| MSWCTL1 | Reserved | | HSECM | VHCFGLK | VHCFGA | LPCRSTA | HPWRON | HRSTOB |
| MSWCTL2 | CFGPSO | CFGPBM | ACPIS5 | ACPIS4 | ACPIS3 | ACPIS2 | ACPIS1 | ACPIS0 |
| MSWCTL3 | Reserved | | | | | RTCAL | LPFTO | HRAPU |
| HCFGBAL | Host Configuration Registers Base Address Low | | | | | | | |
| HCFGBAH | Host Configuration Registers Base Address High | | | | | | | |
| MSIEN2 | EICFGPSO | EICFGPBM | EIACPIS5 | EIACPIS4 | EIACPIS3 | EIACPIS2 | EIACPIS1 | EIRTCAL |
| MSHES0 | Module IRQ Event Status | Software Event Status | Reserved | | RING Event Status | Reserved | RI2 Event Status | RI1 Event Status |

# A. Register List (Continued)

## A.4 HOST DOMAIN REGISTER LAYOUT

### Host Configuration Registers

#### A.4.1 SuperI/O Configuration

(Section 6.1.8 on page 306)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| SID | Family ID | | | | | | | |
| SIOCF1 | Reserved | | Number of DMA Wait States | | Number of I/O Wait States | | Software Reset | SuperI/O Devices Enable |
| SIOCF5 | Reserved | | | SMI to IRQ2 Enable | Reserved | | | |
| SIOCF6 | SCIOF6 Software Lock | General-Purpose Scratch | | RTC Disabled | Reserved | | | |
| SRID | Chip Revision ID | | | | | | | |
| SIOCF8 | Reserved | | | | Reserved | | Reserved | |
| SIOCF9 | Reserved | | | Module Enable Status | Valid Multiplier Clock Status | Clock Enable | SuperI/O Clock Domain Source | |
| SIOCFD | Reserved | | | | | | Power Supply Off | Power Button Mode |

#### A.4.2 Shared Memory Configuration

(Section 6.1.11 on page 311)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Shared Memory Configuration | BIOS FWH ID | | | | BIOS FWH Enable | User-Defined Memory Space Enable | BIOS Extended Space Enable | BIOS LPC Enable |
| Shared Memory Base Address High Byte | User-Defined Memory Zone Address High | | | | | | | |
| Shared Memory Base Address Low Byte | User-Defined Memory Zone Address Low | | | | | | | |
| Shared Memory Size Configuration | Reserved | | | | User-Defined Memory Zone Size | | | |

#### A.4.3 RTC Configuration

(Section 6.1.12 on page 315)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| RLR | Block Standard RAM | Block RAM Write | Block Extended RAM Write | Block Extended RAM Read | Block Extended RAM | Reserved | | |
| DOMAO | Reserved | Date of Month Alarm Register Offset Value | | | | | | |
| MONAO | Reserved | Month Alarm Register Offset Value | | | | | | |
| CENO | Reserved | Century Register Offset Value | | | | | | |

# A. Register List (Continued)

## Host Runtime Registers

### A.4.4 Shared Memory Host

(Section 5.3.7 on page 268)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| SMIMA0 | Indirect Memory Address (7-0) | | | | | | | |
| SMIMA1 | Indirect Memory Address (15-8) | | | | | | | |
| SMIMA2 | Indirect Memory Address (23-16) | | | | | | | |
| SMIMA3 | Indirect Memory Address (31-24) | | | | | | | |
| SMIMD | Indirect Memory Data (7-0) | | | | | | | |
| SMHAP1,2 | Host Access Protection Index | | | | Index Write | Host Lock Protection | Host Write Protection | Host Read Protection |
| SMHSEM | CSEM3 | CSEM2 | CSEM1 | CSEM0 | HSEM3 | HSEM2 | HSEM1 | HSEM0 |

### A.4.5 MSWC Host

(Section 5.5.5 on page 285)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| WK_STS0 | Module IRQ Event Status | Software Event Status | Reserved | | RING Event Status | Reserved | RI2 Event Status | RI1 Event Status |
| WK_EN0 | Module IRQ Event Enable | Software Event Enable | Reserved | | RING Event Enable | Reserved | RI2 Event Enable | RI1 Event Enable |
| WK_CFG | Reserved | | | | | | Configuration Bank Select | |
| WK_SIGV | Reserved | | $\overline{\text{PWUREQ}}$ Wakeup or IRQ | $\overline{\text{PWUREQ}}$ Output Value | PM2 $\overline{\text{SMI}}$ Output | PM1 $\overline{\text{SMI}}$ Output | $\overline{\text{SMI}}$ Wake-up Output | $\overline{\text{SMI}}$ Output Value |
| WK_STATE | Reserved | | S5 | S4 | S3 | S2 | S1 | Reserved |
| WK_SMIEN0 | Reserved | SW Event to $\overline{\text{SMI}}$ Enable | Reserved | | Ring Event to $\overline{\text{SMI}}$ Enable | Reserved | RI2 Event to $\overline{\text{SMI}}$ Enable | RI1 Event to $\overline{\text{SMI}}$ Enable |
| WK_IRQENO | Reserved | SW Event to IRQ Enable | Reserved | | Ring Event to IRQ Enable | Reserved | RI2 Event to IRQ Enable | RI1 Event to IRQ Enable |

### A.4.6 Host Interface (HI) Registers

(“Host Addresses” on page 242)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DBBOUT | Keyboard/Mouse DBBOUT Data | | | | | | | |
| STATUS | ST3 | ST2 | ST1 | ST0 | A2 | F0 | IBF | OBF |
| DBBIN | Keyboard/Mouse DBBIN Data | | | | | | | |
| COMAND | Keyboard/Mouse DBBIN Data | | | | | | | |

# A. Register List (Continued)

### A.4.7 RTC Registers

(Section 6.2.15 on page 325)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| SEC | Seconds Data | | | | | | | |
| SECA | Seconds Alarm Data | | | | | | | |
| MIN | Minutes Data | | | | | | | |
| MINA | Minutes Alarm Data | | | | | | | |
| HOR | Hours Data | | | | | | | |
| HORA | Hours Alarm Data | | | | | | | |
| DOW | Day of Week Data | | | | | | | |
| DOM | Date of Month Data | | | | | | | |
| MON | Month Data | | | | | | | |
| YER | Year Data | | | | | | | |
| CRA | Update in Progress | Divider Chain Control 2-0 | | | Periodic Interrupt Rate Select 3-0 | | | |
| CRB | Set Mode | Periodic Interrupt Enable | Alarm Interrupt Enable | Update Ended Interrupt Enable | Reserved | Data Mode | Hour Mode | Daylight Savings |
| CRC | IRQ Flag | Periodic Interrupt Flag | Alarm Interrupt Flag | Update Ended Interrupt Flag | Reserved | | | |
| CRD | Valid RAM and Time | Reserved | | | | | | |
| DOMA | Date of Month Alarm Data | | | | | | | |
| MONA | Month Alarm Data | | | | | | | |
| CEN | Century Data | | | | | | | |

## B. Software for Hardware Interface

The following table shows the factory parameters of the Information Block.

**Table 61. Factory Parameters**

| Information Block Address | Data [Word] |
|---|---|
| $0000_{16}$ | RevisionCode (unsigned int) |
| $0002_{16}$ | Count (unsigned int) |
| $007C_{16}$ | Copy of SRID register indication for the device revision number.<br>For PC87591L-N05, PC97551: $FFEn_{16}$ |

RevisionCode Interpretation

- $FFxx_{16}$ - Production version of the PC87591L-N05.
- Other - Reserved for future use.

The information block includes factory parameters that are saved during device production and are used for various calibrations.

This information may be read by the core. The core accesses the Information Block using indirect byte/word read access. To read from the Information Block, the byte or word address must be stored in IBAI register, and data bytes/words must be read using a byte/word read operation from IBD register.

### Information Block Access Index Register (IBAI)

This register defines address bits 7-0 for the read transaction from the Information Block.

Location: $00\ F880_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Reserved** | | | | | | | | **Indirect Memory Address 7-0** | | | | | | | |

| Bit | Description |
|---|---|
| 7-0 | **Index Address 7-0.** The address of a byte address in the Information Block. Access to words is allowed only via word-aligned addresses. |
| 15-8 | **Reserved.** |

### Information Block Data Register (IBD)

This register holds the data for the read transaction from the Information Block. Byte or word reads from this register are allowed to access a byte or word pointed to by IBAI register.

Location: $00\ F882_{16}$

Type: R/W

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | **Information Block Data 15-0** | | | | | | | | | | | | | | | |

| Bit | Description |
|---|---|
| 15-0 | **Information Block Data 15-0.** The byte or word data read from the Information Block. |

# C. Booter Program

The PC87591L-N05 Booter program resides in the 4K on-chip ROM.

The Booter has two main functions:

- On power-up, it performs all boot procedures and then passes control to the firmware (EC BIOS).
- If there is a problem with the firmware (EC BIOS), or if the user forces Recovery mode, the Booter enters Recovery mode and allows debugging via JTAG or RS-232 debugging channels.

## C.1 BOOT DATA

**Header 1** resides at the address $1000_{16}$ in the external flash and is defined as follows:

| Offset | Length | |
|---|---|---|
| 00-01 | 2 | Signature |
| 02-03 | 2 | Bus Width |
| 04-05 | 2 | Address of Header 2 |
| 06 | 1 | Action Flag |
| 07 | 1 | Config |
| | | ..... |
| | | Header 2 |

**Note:** The first eight bytes are not included in the checksum count.

- *Signature* (two bytes at offset 0):
  The signature can be one of the following:
  - $49_{16}$ in the lower byte and $4A_{16}$ in the higher byte or $4A49_{16}$ in little endian convention (Light signature),
  - $49_{16}$ in the lower byte and $4E_{16}$ in the higher byte or $4E49_{16}$ in little endian convention (Normal signature).
- *Bus Width* (one word at offset 2):
  This byte instructs the Booter as follows:

| Bit | Description |
|---|---|
| 0-6 | **Reserved.** |
| 7 | **Bus width.**<br>0: External flash device is in 8-bit data mode (byte wide)<br>1: External flash device is in 16-bit data mode (word wide) |
| 8-15 | **Reserved.** |

- *Address of Header 2*: (1 word at offset 4)
  The starting address of Header 2. This field stores the PC value of Header 2. Thus, its value is the address divided by 2. Checksum counting begins at this address.
- *Action Flag*: (1 byte at offset 6)
  Reserved.
- *Config*: (1 byte at offset 7)
  This byte instructs the Booter as follows:

| Bit | Description |
|---|---|
| 0 | **XOR Checksum.**<br>0: Do not perform the XOR checksum<br>1: Perform the XOR checksum |

## C. Booter Program (Continued)

| Bit | Description |
|---|---|
| 1 | **USART Configuration.**<br>0: Do not configure USART module for debug before booting<br>1: Configure USART module for debug before booting |
| 2 | **Force Recovery Mode.**<br>0: If the header is valid, perform boot normally<br>1: Enter recovery mode in all cases |
| 5-3 | **HFCG Clock Frequency.** Determines the clock frequency that is set before performing the checksum.<br>**Bits**<br>**5 4 3 Frequency**<br>0 0 0: 4 MHz<br>0 0 1: 8 MHz<br>0 1 0: 16 MHz<br>0 1 1: 20 MHz<br>Other: Reserved |
| 7-6 | **Reserved** |

**Header 2** includes the following:

| Offset | Length | |
|---|---|---|
| 00-01 | 2 | ROM Size |
| 02-03 | 2 | Start 1 |
| 04 | 1 | Checksum |
| 05 | 1 | Forced Update 1 |
| 06 | 1 | Forced Update 2 |
| 07 | 1 | Forced Update 3 |
| 08 | 1 | Flash Size |
| 09 | 1 | Reserved |
| 10 | 1 | MCFG_DAT |
| 11 | 2 | ZONE0CFG |
| 13 | 2 | ZONE1CFG |
| 15 | 1 | XOR Checksum Res. |
| 16 | 2 | Protection Word |
| 18 | 2 | ZONE2CFG |
| 20 | 1 | PNMR |
| 21 | 3 | Reserved |

- *ROM Size*: (1 word at offset 0)
  This is the length, in bytes, of the area for which checksum is performed.
- *Start 1*: (1 word at offset 2)
  The firmware (EC BIOS) entry point address. This is the PC value; thus its value is the address divided by 2.
- *Checksum*: (1 byte at offset 4)
  The checksum result of Header 2.

  To generate the value of this field, first calculate the checksum starting at offset 00 of Header 2, up to the offset of the "last_ROM_byte", not including offset 04 (Checksum), byte per byte; then calculate the 1-byte 2's complement of this number and store it in offset 04 (Checksum). The offset of the "last_ROM_byte" is the value of the ROM Size minus 1.

## C. Booter Program (Continued)

To verify the Header 2 checksum is correct, calculate the checksum starting at offset 00 of Header 2, up to the offset of the "last_ROM_byte", byte per byte; the resulting value must be $00_{16}$.

- *Forced Update 1*: (1 byte at offset 5) Reserved.
- *Forced Update 2*: (1 byte at offset 6) Reserved.
- *Forced Update 3*: (1 byte at offset 7) Reserved.
- *FlashSize*: (1 byte at offset 8) Reserved.
- *Reserved*: (1 byte at offset 9) Reserved. This byte should be programed to $00_{16}$.
- *MCFG_DAT*: (1 byte at offset 10) Bits 3,4 and 5 are copied into MCFG register (address $FF10_{16}$) and MCFGSH register (address $FBFE_{16}$).
- *ZONE0CFG*: (1 word at offset 11) The Zone 0 configuration register value.
- *ZONE1CFG*: (1 word at offset 13) The Zone 1 configuration register value.

  **Note:** The Booter sets the BIU configuration registers before performing the checksum.
- *XOR Checksum Result:* (1 byte at offset 15)

  The XOR checksum result of Header 2, not including the checksum field at offset 4.

  To generate the value of this field, first calculate the XOR checksum starting at offset 00 of Header 2, up to the offset of the "last_ROM_byte", not including offset 04 (Checksum) and offset 15 (XOR Checksum Result), byte per byte; then store this number in offset 15 (XOR Checksum Result). The offset of the "last_ROM_byte" is the value of the ROM Size minus 1.

  To verify the Header 2 XOR checksum is correct, calculate the XOR checksum starting at offset 00 of Header 2, up to the offset of the "last_ROM_byte", not including offset 04 (Checksum), byte per byte; the resulting value must be $00_{16}$.

  **Note:** The XOR Checksum Result value is counted in the normal Checksum operation; therefore, calculate the XOR checksum before starting to calculate the normal Checksum (at offset 04).
- *Protection Word* (1 word at offset 16): Protection Word value.

  If this word is $00_{16}$, the value of PTWRL and PTWRH registers is not changed.

  If this word is different from $00_{16}$, its value is copied into PTWRL and PTWRH register as follows:

  The lower byte is copied to PTWRL (address $FF06_{16}$) and the higher byte to PTWRH (address $FF08_{16}$).
- *ZONE2CFG* (1 word at offset 18): The Zone 2 configuration register value.
- PNMR (1 byte at offset 20): If this byte is different from $FF_{16}$ and bit 7 is set the value is copied into PNMR register (address $FF0A_{16}$), bit 7 is not copied.
- *Reserved*: (3 bytes at offset 21
  This field is reserved. Set all 3 bytes to $00_{16}$.

### C.2 BOOT SEQUENCE

The boot sequence is as follows:

1. If the chip has valid ADC calibration values in the information block, the Booter uses them to calibrate the ADC module. Otherwise, it sets the ADC Calibration registers to the default values as follows:

| Calibration Register Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value (Hex) | $AC_{16}$ | $30_{16}$ | $30_{16}$ | $F5_{16}$ | $F5_{16}$ | $00_{16}$ | $00_{16}$ | $87_{16}$ | $87_{16}$ | $88_{16}$ | $85_{16}$ | $8C_{16}$ | $86_{16}$ |

2. The Booter checks the firmware (EC BIOS) header signature. If the signature is valid ($4A49_{16}$ or $4E49_{16}$), the Booter configures the BIU and other system settings, as specified by the header, as follows:
   a. It sets MCFG register (Offset $FF10_{16}$) bits 0 and 1 to enable expansion I/O and memory. Bit 2 is also set if the *Bus Width* field in the Header is set to 16-bit mode. Bits 3,4 and 5 from MCFG_DAT field in Header 2 are copied to the MCFG register.
   b. It sets MCFGSH register (Offset $FBFE_{16}$) to the same value as MCFG.
   c. In OBD mode only, It sets DBGCFG register (Offset $FF16_{16}$) bit 0, to enable an ISE interrupt.
   d. It sets SZCFG0 register (Offset $F984_{16}$) to the value specified in ZONE0CFG field in Header 2.
   e. It sets SZCFG1 register (Offset $F986_{16}$) to the value specified in ZONE1CFG field in Header 2.
   f. It sets SZCFG2 register (Offset $F988_{16}$) to the value specified in ZONE2CFG field in Header 2.
   g. It sets the High Frequency clock to the frequency specified in the Config field in Header 1. The accelerator clock is disabled.

## C. Booter Program (Continued)

h. It sets the PTWRL and PTWRH registers (Offset $FF06_{16}$ and $FF08_{16}$) to the value specified in the low byte and high byte of the Protection Word field (in Header 2) respectively, if the field value is other than $00_{16}$.

i. $_{16}$It clears HOSTWAIT bit (sticky bit) in MCFG register (Offset $FBFE_{16}$) to release the host from LPC wait state.

j. It sets PNMR register (Offset $FF0A_{16}$) to the value specified in PNMR field in Header 2 if the value is different from $FF_{16}$ and bit 7 is not set. Bit 7 is not copied.

3. If the USART configuration bit (Bit 1 in Config field in Header 1) is set, the Booter configures the USART1 module as follows:
   a. It sets PBALT register (Offset $FE32_{16}$) bit 1, to enable RX alternate function. (TX remain disabled until the first USART transaction is received in Recovery mode).
   b. It sets PBWPU register (Offset $FE30_{16}$) bit s1 and 2, to enable weak pull-up and to avoid noise interference.
   c. It sets UPSR register (Offset $FE2E_{16}$) to $C8_{16}$, to set up the baud rate.
   d. It sets UICTRL register (Offset $FD24_{16}$) bit 6, to enable the RX interrupt.
   e. It sets IENAM1 register (Offset $FE10_{16}$) bit 0, to enable the USART1 interrupt in ICU module.
4. The Booter checks the validity of the firmware (EC BIOS) using code checksum.The sum operation begins at the starting address of Header 2 (as specified in Header 1), and includes the area of the firmware (EC BIOS) memory, as specified in Header 2 (*ROM-size* field).
5. If the firmware (EC BIOS) is valid, the Booter jumps to the firmware (EC BIOS) entry point (also specified in the header). At this point, all resources (RAM) used by the Booter are free and available.

   If the firmware (EC BIOS) is invalid, the Booter enters Recovery mode and acts as a Target Monitor (TMON), implementing debugging functionality via JTAG or RS-232 debugging channels, as described in the *TMON Communication Protocol*.

**Note:** You can define memory writes for the Booter, so that it can perform writes to any type of flash module mapped in the external memory.

### C.3 RECOVERY MODE

The PC87591L-N05 Booter enters Recovery mode if any of the following EC Firmware problems occurs:

- The firmware (EC BIOS) signature is invalid.
- The firmware (EC BIOS) code checksum is wrong.
- Force recovery mode bit (bit 2) in *Config* field in Header 1 is set.
- The firmware (EC BIOS) is valid, an abort signal is sent via debugging channel at run-time and the EC-BIOS dispatch table is initialized as described in Section C.6.

In Recovery mode, you can connect to the PC87591L-N05 and communicate with the internal monitor via JTAG or RS-232 debugging channels.

While the Booter is in Recovery mode, RAM resources from address $F6A0_{16}$ to $F7FF_{16}$ are used. Do not change data in this memory section while debugging.

#### C.3.1 RS-232 Connection

If the RS-232 channel is used:

- Connect IOPB0/URXD1 to the RS-232 RX pin.
- Connect IOPB1/UTXD1 to the RS-232 TX pin.

This connection needs a driver/receiver to transform the voltage between the host RS-232 (±12V) and the PC87591L-N05 USART (3V).

The RS-232 channel settings must be:

- 38400 bps baud rate.
- Software flow control.
- Software reset.

Resources needed:

- Host system must have a serial port available.
- CR16B debugging tools must be installed.

## C. Booter Program (Continued)

Debugging limitations:

- No hardware reset.
- Debugging transactions are triggered by a maskable interrupt (instead of the ISE trap used in JTAG).
- Low data transfer speed.

### C.3.2 JTAG Connection

If the JTAG connection is used, connect these five pins:

- TDI
- TDO
- TMS
- TCK
- TINT

To connect through JTAG, the PC87591L-N05 must boot in OBD environment. This means that the ENV1 strap pin must be pulled up.

Resources needed:

- The host system must be equipped with a JTAG device.
- CR16B debugging tools must be installed.

Debugging limitations:

- The chip must be in OBD environment.
- Five pins must be connected (compared to two pins for the RS-232 connection).

### C.4 MONITOR MEMORY WRITES

Monitor memory writes are divided into two groups:

- Internal memory writes from addresses $0000_{16}$ - $0FFF_{16}$ (Internal ROM - however the ROM cannot be written) and $E000_{16}$ - $FFFF_{16}$ (Internal RAM and I/O-mapped registers); these are normal memory writes.
- External memory writes (from addresses $1000_{16}$ - $DFFF_{16}$ and $10000_{16}$ and higher); there are three configurable working modes that affect the memory writes to these addresses

**Working Modes**

The working mode is selected by the byte-wide RAM register placed at address $F7FE_{16}$.

There are three working modes:

- **Eprom** mode, selected by writing $01_{16}$ to the RAM register (default).

  The monitor performs the following JEDEC-compatible algorithm for each byte; when writing value XX at address YY the monitor:

  — When the external flash bus width is eight bits:
    1. Writes $AA_{16}$ to address $5555_{16}$.
    2. Writes $55_{16}$ to address $2AAA_{16}$.
    3. Writes $A0_{16}$ to address $5555_{16}$.
    4. Writes XX to address YY.

  — When the external flash bus width is 16 bits:
    1. Writes $AA_{16}$ to address $AAAA_{16}$.
    2. Writes $55_{16}$ to address $5554_{16}$.
    3. Writes $A0_{16}$ to address $AAAA_{16}$.
    4. Writes XX to address YY.

  **Note:** The Booter auto-detects the flash bus width if the flash device is JEDEC compatible.
- **Normal** mode, selected by writing $02_{16}$ to the RAM register:

  The monitor performs a normal write (i.e. writes XX to address YY).

  This mode is recommended when external memory is mapped to a RAM device.
- **Generic** mode, selected by writing $03_{16}$ to the RAM register:

  The monitor calls a function placed in the RAM at address $F400_{16}$ with all parameters. This function should perform the memory write. The function prototype is:

## C. Booter Program (Continued)

```
void Rom_Write(
                __far unsigned      char* Source,
                __far unsigned      char* Destination,
                unsigned            short Size
    );
```

Where:

— **`Source`** is the source pointer address (four bytes sent via registers r2 and r3)

— **`Destination`** is the destination pointer address (four bytes sent via registers r4 and r5)

— **`Size`** is the size in bytes of the memory block to copy (two bytes sent via the stack at the address: 0(sp))

Before attempting to perform external memory writes, load the function into RAM via either the JTAG or RS-232 debugging channel.

### C.5 EXTERNAL FLASH ERASE

A special erase (external flash erase) can be performed by writing a special erase function, loading it to the RAM and executing it via the JTAG or RS-232 debugging channels.

**Note:** The same procedure can also be used for other operations on the external flash, such as reading the flash device and manufacturer ID etc.

### C.6 DEBUGGING CAPABILITIES OF THE BOOTER

The Booter contains TMON libraries and can be used for debugging. If the header is valid, the user can force Recovery mode at run time.

The Booter contains all the entry addresses required for the dispatch table, which must be implemented in the EC firmware to allow debugging. The entry address list is:

- NmiHandler (entry number 1) Address value: $0006_{16}$.
- SvcHandler (entry number 5) Address value: $000A_{16}$.
- DvzHandler (entry number 6) Address value: $000E_{16}$.
- FlgHandler (entry number 7) Address value: $0012_{16}$.
- BptHandler (entry number 8) Address value: $0016_{16}$
- TrcHandler (entry number 9) Address value: $001A_{16}$.
- UndHandler (entry number 10) Address value: $001E_{16}$.
- DbgHandler (entry number 14) Address value: $0022_{16}$.
- IseHandler (entry number 15) Address value: $0026_{16}$.
- USART1 interrupt handler (entry number 34) Address value: $002A_{16}$.

All these values must be present in the dispatch table at the specific entry places to permit debugging. The USART1 interrupt handler is not needed when debugging via the JTAG channel (JTAG channel is enabled only in OBD mode).

**Note:** For Large model, same values are used in 32-bit format.

To allow debugging via the RS-232 channel, the USART1 interrupt handler must be present with all the other entries in the dispatch table, and the USART1 Enable bit (bit 1 in Config field in the main header) must be set (USART1 channel is enabled in both OBD and IRE modes). The RS-232 debugging channel must be configured to work at 38400 BPS baud rate. (See the *CompactRISC Debugger Communication Interface (DbgCom) User Guide*).

To force Recovery mode at run time, an ABORT signal must be sent via the debugging channel. The Booter enters Recovery mode, while keeping the core status, and waits for debug commands.

**Note:** The user should not clear GTMON bit (bit 7 in MCFG register) nor modify byte at address F7FF16 in the RAM in order to allow forcing recovery mode at run time.

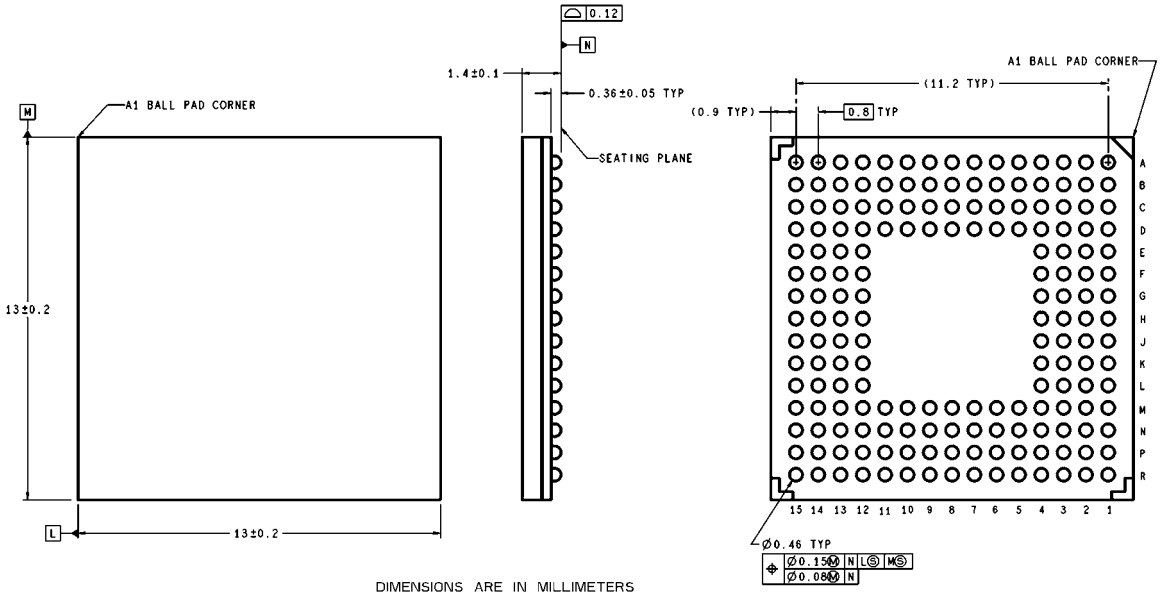
#### Debugging Limitations

When debugging with the Booter, the following limitations must be taken into account:

- Booter memory resources must not be overwritten (Offset $F6A0_{16}$ to $F7FF_{16}$ in all modes, except Generic mode: offset $F400_{16}$ to $F7FF_{16}$).
- Software breakpoints are not allowed on code placed in the external flash.
- To use the JTAG interface, the chip must be in OBD mode.
- When working with the RS-232 channel, the HFCG frequency must not be changed and core interrupts must always be enabled (PSR I and E bits).

## Physical Dimensions

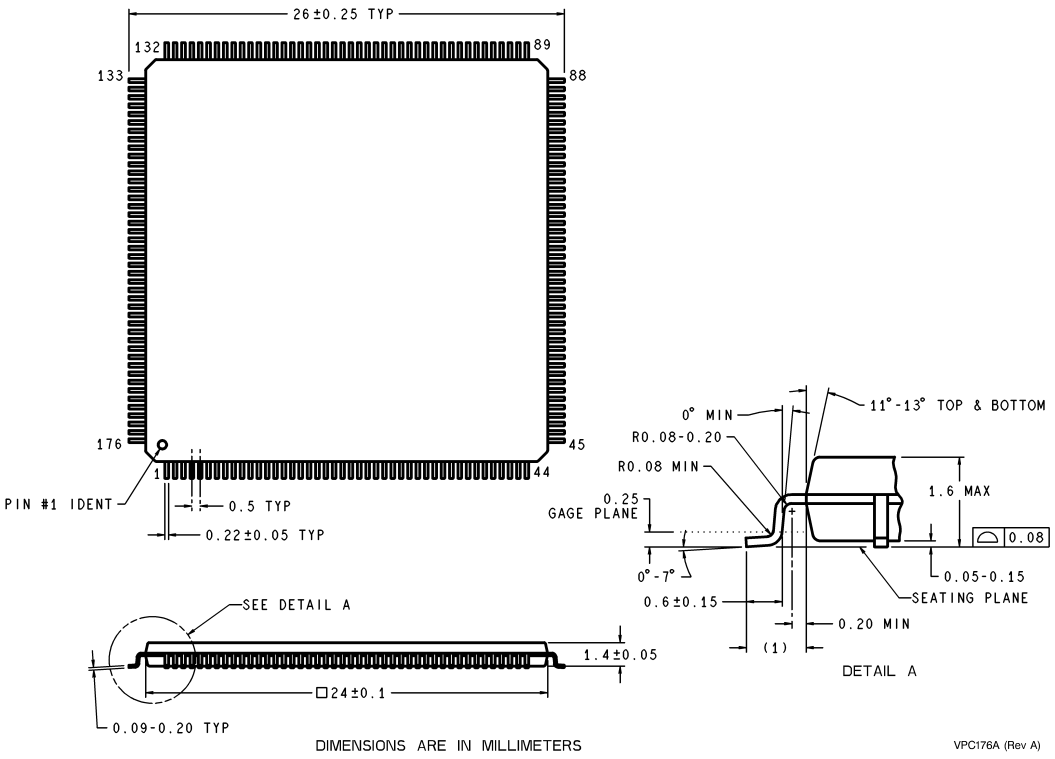
All dimensions are in millimeters




**176-Pin Fine Pitch Ball Grid Array (FBGA)**
**Order Number P87591L-SLCN05**
**NS Package Number SLC176A**

## Physical Dimensions (Continued)

All dimensions are in millimeters




**176-Low Profile Plastic Quad Flatpack (LQFP)**
**Order Number PC87591L-VPCN05**
**NS Package Number VPC176**

### LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

BANNED SUBSTANCE COMPLIANCE

National Semiconductor certifies that the products and packing materials meet the provisions of the Customer Products Stewardship Specification (CSP-9-111C2) and the Banned Substances and Materials of Interest Specification (CSP-9-111S2) and contain no "Banned Substances" as defined in CSP-9-111S2.

**National Semiconductor Corporation**
**Americas**
Email:
new.feedback@nsc.com

**National Semiconductor**
**Europe**
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel:+33 (0) 1 41 91 87 90

**National Semiconductor**
**Asia Pacific Customer**
**Response Group**
Tel: 65-2544466
Fax: 65-2504466
Email: ap.support@nsc.com

**National Semiconductor**
**Japan Ltd.**
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507
Email: nsj.crc@jksmtp.nsc.com

www.national.com