AT91 ARM Thumb-based Microcontrollers

AT91FR40162SB

Preliminary

Features

- Incorporates the ARM7TDMI® ARM® Thumb® Processor Core
  - High-performance 32-bit RISC Architecture
  - High-density 16-bit Instruction Set
  - Leader in MIPS/Watt
  - EmbeddedICE™ (In-circuit Emulation)
- 256K Bytes of On-chip SRAM
  - 32-bit Data Bus, Single-clock Cycle Access
- 1024K Words 16-bit Flash Memory (2M bytes)
  - Single Voltage Read/Write,
  - Sector Erase Architecture
  - Erase Suspend Capability
  - Low-power Operation
  - Data Polling, Toggle Bit and Ready/Busy End of Program Cycle Detection
  - Reset Input for Device Initialization
  - Sector Program Unlock Command
  - 128-bit Protection Register
  - Factory-programmed AT91 Flash Memory Uploader Software
- Fully Programmable External Bus Interface (EBI)
  - Up to 8 Chip Selects, Maximum External Address Space of 64M Bytes
  - Software Programmable 8/16-bit External Data Bus
- 8-level Priority, Individually Maskable, Vectored Interrupt Controller
  - 4 External Interrupts, Including a High-priority Low-latency Interrupt Request
- 32 Programmable I/O Lines
- 3-channel 16-bit Timer/Counter
  - 3 External Clock Inputs, 2 Multi-purpose I/O Pins per Channel
- 2 USARTs
  - Two Dedicated Peripheral Data Controller (PDC) Channels per USART
- Programmable Watchdog Timer
- Advanced Power-saving Features
  - CPU and Peripherals Can be De-activated Individually
- Fully Static Operation:
  - 0 Hz to 75 MHz Internal Frequency Range at VDDCORE = 1.8V, 85· C
- 2.7V to 3.6V I/O Operating Range, 1.65V to 1.95V Core Operating Range
- -40· C to 85· C Temperature Range
- Available in a 121-ball 10 x 10 x 1.26 mm BGA Package with 0.8 mm Ball Pitch

6410B–ATARM–12-Jan-10

1. Description

The AT91FR40162SB is a member of the Atmel AT91 16/32-bit Microcontroller family, which is based on the ARM7TDMI processor core. The processor has a high-performance 32-bit RISC architecture with a high-density 16-bit instruction set and very low power consumption.

The AT91FR40162SB ARM microcontroller features 2 Mbits of on-chip SRAM and 2 Mbytes of Flash memory in a single compact 121-ball BGA package. Its high level of integration and very small footprint make the device ideal for space-constrained applications. The high-speed on-chip SRAM enables a performance of up to 74 MIPs in typical conditions with significant power reduction and EMC improvement over an external SRAM implementation.

The Flash memory may be programmed via the JTAG/ICE interface or the factory-programmed Flash Memory Uploader (FMU) using a single device supply, making the AT91FR40162SB suitable for in-system programmable applications.

2. Migrating from the AT91FR40162S to the AT91FR40162SB

2.1 Hardware Requirements

The AT91FR40162SB is pin-to-pin compatible to the AT91FR40162S, so the AT91FR40162SB can be soldered in place of the AT91FR40162S without any other hardware changes.

The AT91FR40162SB does not feature a VPP pin, thus ball D5 of the 121-ball BGA package of the AT91FR40162SB is NC (Not connected). This ball can either be connected to a supply up to 13V (as could be the VPP ball of the AT91FR40162S), grounded or left unconnected.

2.2 Software Requirements

Except for the Flash memory, the processor, the architecture and the peripherals of both the AT91FR40162S and the AT91FR40162SB are identical, any program written for an AT91FR40162S-based system can run as is on the same system built with an AT91FR40162SB, with the exception of aspects related to the Flash memory.

2.3 Flash Memory Difference

Some features of the embedded Flash memories in the AT91FR40162S and the AT91FR40162SB are not fully identical.

2.3.1 Device ID

The Device Code of the Flash Memory of the AT91FR40162SB is 01C0H instead of 00C0H for the AT91FR40162S. Users who use this Device Code must modify the software.

2.3.2 VPP Features

As the AT91FR40162SB does not feature a VPP pin, neither the write protection feature nor the double-word fast write feature are available on this device.

If the hardware write protection feature is used on the AT91FR40162S, it should be replaced by a software-controlled write protection method with the Sector Lockdown command, or removed from the application.

If the Double Byte/Word Program command was used on the AT91FR40162S, the user needs to change the flash programming sequence and to use only the standard Byte/Word Program command.

The VPP Status I/O3 does not exist anymore in the Status word returned by the Flash Memory.

2.3.3 Erase Cycle Timings

The 32K Word sector erase cycle time maximum value has been increased from 5 seconds to 6 seconds. In case the end of erase cycle is not used, but a fixed timeout is used instead, the value of the timeout must be checked against the new value.

2.3.4 CFI Common Flash Interface

The Common Flash Interface table (Table 12-5, “Common Flash Interface Definition,” on page 68) Erase block information of the 64-KByte and the 8-KByte sectors addresses was not fully CFI-compliant on the AT91FR40162S. The AT91FR40162SB is fully CFI-compliant, and thus the Erase block information of the 64-KByte and the 8-KByte sector addresses in the Common Flash Interface table have changed.

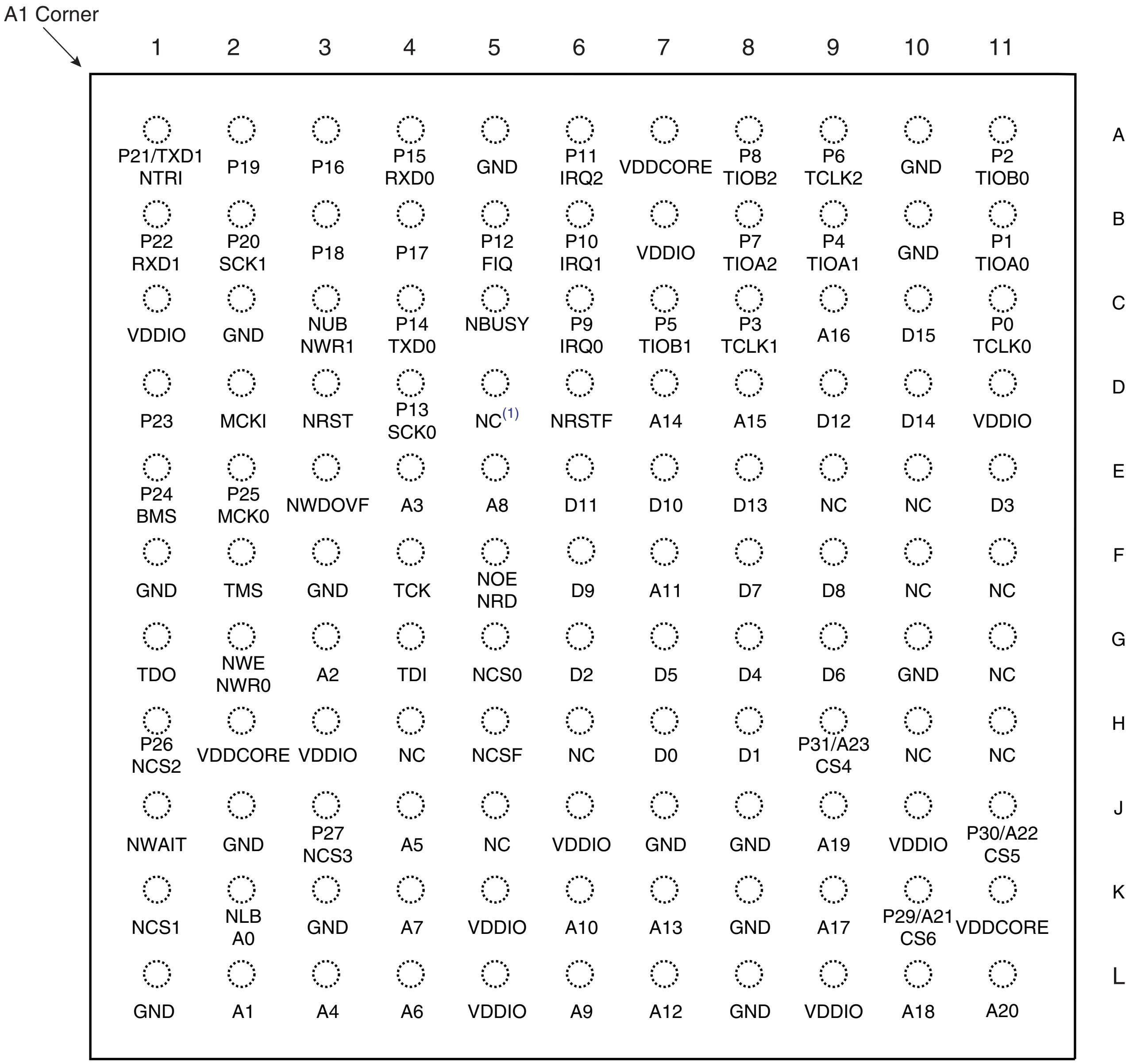
Users who managed the programming of the flash with the CFI algorithm on the AT91FR40162S should adapt their programming for the AT91FR40162SB.

2.3.5 Fully Green Package

The AT91FR40162S is RoHS compliant, whereas the AT91FR40162SB is fully Green qualified. This has no impact on the soldering profile to be used, but only improves environmental considerations.

3. Pin Configuration

Figure 3-1. AT91FR40162SB Pinout for 121-ball BGA Package (Top View)



	1	2	3	4	5	6	7	8	9	10	11
A	P21/TXD1 NTRI	P19	P16	P15 RXD0	GND	P11 IRQ2	VDDCORE	P8 TIOB2	P6 TCLK2	GND	P2 TIOB0
B	P22 RXD1	P20 SCK1	P18	P17	P12 FIQ	P10 IRQ1	VDDIO	P7 TIOA2	P4 TIOA1	GND	P1 TIOA0
C	VDDIO	GND	NUB NWR1	P14 TXD0	NBUSY	P9 IRQ0	P5 TIOB1	P3 TCLK1	A16	D15	P0 TCLK0
D	P23	MCKI	NRST	P13 SCK0	NC[1]	NRSTF	A14	A15	D12	D14	VDDIO
E	P24 BMS	P25 MCK0	NWDOVF	A3	A8	D11	D10	D13	NC	NC	D3
F	GND	TMS	GND	TCK	NOE NRD	D9	A11	D7	D8	NC	NC
G	TDO	NWE NWR0	A2	TDI	NCS0	D2	D5	D4	D6	GND	NC
H	P26 NCS2	VDDCORE	VDDIO	NC	NCSF	NC	D0	D1	P31/A23 CS4	NC	NC
J	NWAIT	GND	P27 NCS3	A5	NC	VDDIO	GND	GND	A19	VDDIO	P30/A22 CS5
K	NCS1	NLB A0	GND	A7	VDDIO	A10	A13	GND	A17	P29/A21 CS6	VDDCORE
L	GND	A1	A4	A6	VDDIO	A9	A12	GND	VDDIO	A18	A20

Note: 1. Not connected, can either be connected to GND, VCC or left unconnected.

4. Signal Description

Table 4-1. AT91FR40162SB Signal Description

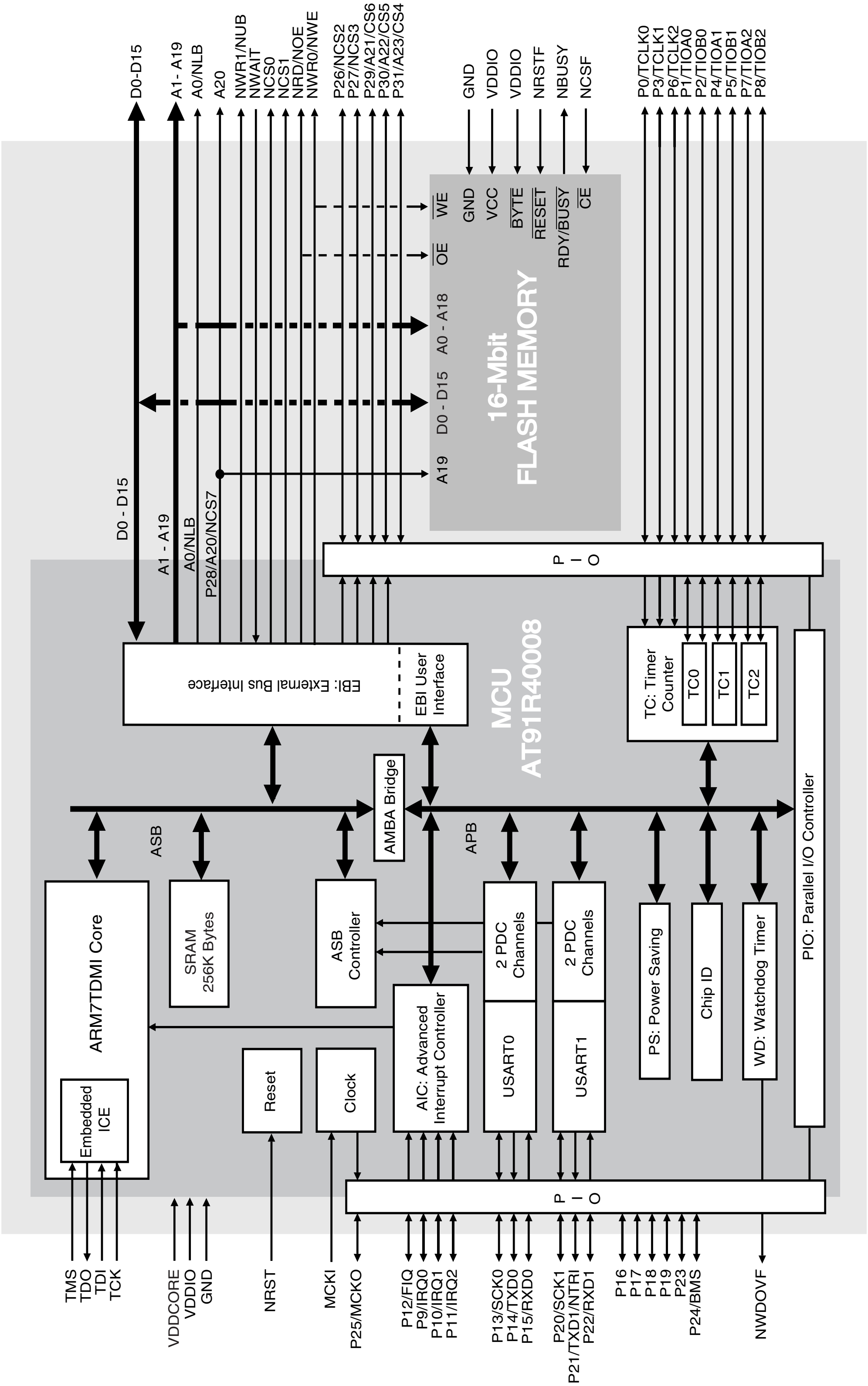
Module	Name	Function	Type	Active Level	Comments
EBI	A0 - A23	Address Bus	Output	–	Valid after reset; do not reprogram A20 to I/O, as it is MSB of Flash address
	D0 - D15	Data Bus	I/O	–	
	NCS0 - NCS3	External Chip Select	Output	Low	Used to select external devices
	CS4 - CS7	External Chip Select	Output	High	A23 - A20 after reset
	NWR0	Lower Byte 0 Write Signal	Output	Low	Used in Byte Write option
	NWR1	Upper Byte 1 Write Signal	Output	Low	Used in Byte Write option
	NRD	Read Signal	Output	Low	Used in Byte Write option
	NWE	Write Enable	Output	Low	Used in Byte Select option
	NOE	Output Enable	Output	Low	Used in Byte Select option
	NUB	Upper Byte Select	Output	Low	Used in Byte Select option
	NLB	Lower Byte Select	Output	Low	Used in Byte Select option
	NWAIT	Wait Input	Input	Low	
	BMS	Boot Mode Select	Input	–	Sampled during reset; must be driven low during reset for Flash to be used as boot memory
AIC	FIQ	Fast Interrupt Request	Input	–	PIO-controlled after reset
	IRQ0 - IRQ2	External Interrupt Request	Input	–	PIO-controlled after reset
Timer	TCLK0 - TCLK2	Timer External Clock	Input	–	PIO-controlled after reset
	TIOA0 - TIOA2	Multi-purpose Timer I/O Pin A	I/O	–	PIO-controlled after reset
	TIOB0 - TIOB2	Multi-purpose Timer I/O Pin B	I/O	–	PIO-controlled after reset
USART	SCK0 - SCK1	External Serial Clock	I/O	–	PIO-controlled after reset
	TXD0 - TXD1	Transmit Data Output	Output	–	PIO-controlled after reset
	RXD0 - RXD1	Receive Data Input	Input	–	PIO-controlled after reset
PIO	P0 - P31	Parallel IO Line	I/O	–	
WD	NWDOVF	Watchdog Overflow	Output	Low	Open drain
Clock	MCKI	Master Clock Input	Input	–	Schmidt trigger
	MCKO	Master Clock Output	Output	–	
Reset	NRST	Hardware Reset Input	Input	Low	Schmidt trigger
	NTRI	Tri-state Mode Select	Input	Low	Sampled during reset
ICE	TMS	Test Mode Select	Input	–	Schmidt trigger, internal pull-up
	TDI	Test Data Input	Input	–	Schmidt trigger, internal pull-up
	TDO	Test Data Output	Output	–	
	TCK	Test Clock	Input	–	Schmidt trigger, internal pull-up

Table 4-1. AT91FR40162SB Signal Description (Continued)

Module	Name	Function	Type	Active Level	Comments
Flash Memory	NCSF	Flash Memory Select	Input	Low	$\overline{\text{Enables}}$ Flash Memory when pulled low
	NBUSY	Flash Memory Busy Output	Output	Low	Flash RDY/$\overline{\text{BUSY}}$ signal; open-drain
	NRSTF	Flash Memory Reset Input	Input	Low	Resets Flash to standard operating mode
Power	VDDIO	Power	Power	–	All $V_{DDIO,}$ V_{DDCORE} and all GND pins MUST be connected to their respective supplies by the shortest route
	VDDCORE	Power	Power	–	
	GND	Ground	Ground	–	

5. Block Diagram

Figure 5-1. AT91FR40162SB Block Diagram



6. Architectural Overview

The AT91FR40162SB integrates Atmel's AT91R40008 ARM Thumb processor and a 2-Mbyte (16-Mbit) Flash memory die in a single compact 121-ball BGA package. The address, data and control signals, except the Flash memory enable, are internally interconnected.

The AT91R40008 architecture consists of two main buses, the Advanced System Bus (ASB) and the Advanced Peripheral Bus (APB). Designed for maximum performance and controlled by the memory controller, the ASB interfaces the ARM7TDMI processor with the on-chip 32-bit SRAM memory, the External Bus Interface (EBI) connected to the encapsulated Flash and the AMBA™ Bridge. The AMBA Bridge drives the APB, which is designed for accesses to on-chip peripherals and optimized for low power consumption.

The AT91FR40162SB implements the ICE port of the ARM7TDMI processor on dedicated pins, offering a complete, low-cost and easy-to-use debug solution for target debugging.

6.1 Memories

The AT91FR40162SB embeds 256K bytes of internal SRAM. The internal memory is directly connected to the 32-bit data bus and is single-cycle accessible. This provides maximum performance of 67 MIPS at 75 MHz by using the ARM instruction set of the processor, minimizing system power consumption and improving on the performance of separate memory solutions.

The AT91FR40162SB features an External Bus Interface (EBI), which enables connection of external memories and application-specific peripherals. The EBI supports 8- or 16-bit devices and can use two 8-bit devices to emulate a single 16-bit device. The EBI implements the early read protocol, enabling faster memory accesses than standard memory interfaces.

The AT91FR40162SB encapsulates a Flash memory organized as 1024K 16-bit words, accessed via the EBI. A 16-bit Thumb instruction can be loaded from Flash memory in a single access. Separate MCU and Flash memory reset inputs (NRST and NRSTF) are provided for maximum flexibility. The user is thus free to tailor the reset operation to the application.

The AT91FR40162SB integrates resident boot software called AT91 Flash Memory Uploader software in the encapsulated Flash. The AT91 Flash Memory Uploader software is able to upload program application software into its Flash memory.

6.2 Peripherals

The AT91FR40162SB integrates several peripherals, which are classified as system or user peripherals.

All on-chip peripherals are 32-bit accessible by the AMBA Bridge, and can be programmed with a minimum number of instructions. The peripheral register set is composed of control, mode, data, status and enable/disable/status registers.

An on-chip Peripheral Data Controller (PDC) transfers data between the on-chip USARTs and on- and off-chip memory address space without processor intervention. Most importantly, the PDC removes the processor interrupt handling overhead, making it possible to transfer up to 64K contiguous bytes without reprogramming the start address, thus increasing the performance of the microcontroller, and reducing the power consumption.

6.2.1 System Peripherals

The External Bus Interface (EBI) controls the external memory or peripheral devices via an 8- or 16-bit data bus and is programmed through the APB. Each chip select line has its own programming register.

The Power-saving (PS) module implements the Idle Mode (ARM7TDMI core clock stopped until the next interrupt) and enables the user to adapt the power consumption of the microcontroller to application requirements (independent peripheral clock control).

The Advanced Interrupt Controller (AIC) controls the internal interrupt sources from the internal peripherals and the four external interrupt lines (including the FIQ) to provide an interrupt and/or fast interrupt request to the ARM7TDMI. It integrates an 8-level priority controller, and, using the Auto-vectoring feature, reduces the interrupt latency time.

The Parallel Input/Output Controller (PIO) controls up to 32 I/O lines. It enables the user to select specific pins for on-chip peripheral input/output functions, and general-purpose input/output signal pins. The PIO controller can be programmed to detect an interrupt on a signal change from each line.

The Watchdog (WD) can be used to prevent system lock-up if the software becomes trapped in a deadlock.

The Special Function (SF) module integrates the Chip ID, the Reset Status and the Protect registers.

6.2.2 User Peripherals

Two USARTs, independently configurable, enable communication at a high baud rate in synchronous or asynchronous mode. The format includes start, stop and parity bits and up to 8 data bits. Each USART also features a Timeout and a Time Guard register, facilitating the use of the two dedicated Peripheral Data Controller (PDC) channels.

The 3-channel, 16-bit Timer Counter (TC) is highly programmable and supports capture or waveform modes. Each TC channel can be programmed to measure or generate different kinds of waves, and can detect and control two input/output signals. The TC has also 3 external clock signals.

7. Product Overview

7.1 Power Supply

The AT91FR40162SB device has two types of power supply pins:

- VDDCORE pins that power the chip core (i.e., the AT91R40008 with its embedded SRAM and peripherals)
- VDDIO pins that power the AT91R40008 I/O lines and the Flash memory

An independent I/O supply allows a flexible adaptation to external component signal levels.

7.2 Input/Output Considerations

The AT91FR40162SB I/O pads accept voltage levels up to the VDDIO power supply limit. After the reset, the microcontroller peripheral I/Os are initialized as inputs to provide the user with maximum flexibility. It is recommended that in any application phase, the inputs to the microcontroller be held at valid logic levels to minimize the power consumption.

7.3 Master Clock

The AT91FR40162SB has a fully static design and works on the Master Clock (MCK), provided on the MCKI pin from an external source.

The Master Clock is also provided as an output of the device on the pin MCKO, which is multiplexed with a general purpose I/O line. While NRST is active, and after the reset, the MCKO is valid and outputs an image of the MCK signal. The PIO Controller must be programmed to use this pin as standard I/O line.

7.4 Reset

Reset restores the default states of the user interface registers (defined in the user interface of each peripheral), and forces the ARM7TDMI to perform the next instruction fetch from address zero. Except for the program counter the ARM7TDMI registers do not have defined reset states.

7.4.1 NRST Pin

NRST is an active low-level input. It is asserted asynchronously, but exit from reset is synchronized internally to the MCK. The signal presented on MCKI must be active within the specification for a minimum of 10 clock cycles up to the rising edge of NRST to ensure correct operation. The first processor fetch occurs 80 clock cycles after the rising edge of NRST.

7.4.2 Watchdog Reset

The watchdog can be programmed to generate an internal reset. In this case, the reset has the same effect as the NRST pin assertion, but the pins BMS and NTRI are not sampled. Boot Mode and Tri-state Mode are not updated. If the NRST pin is asserted and the watchdog triggers the internal reset, the NRST pin has priority.

7.5 Emulation Functions

7.5.1 Tri-state Mode

The AT91FR40162SB microcontroller provides a tri-state mode, which is used for debug purposes. This enables the connection of an emulator probe to an application board without having to desolder the device from the target board. In tri-state mode, all the output pin drivers of the AT91R40008 microcontroller are disabled.

In tri-state mode, direct access to the Flash via external pins is provided. This enables production Flash programming using classical Flash programmers prior to board mounting.

To enter tri-state mode, the NTRI pin must be held low during the last 10 clock cycles before the rising edge of NRST. For normal operation, the NTRI pin must be held high during reset by a resistor of up to 400 kΩ

NTRI is multiplexed with I/O line P21 and USART1 serial data transmit line TXD1.

7.5.2 JTAG/ICE Debug

ARM-standard embedded In-circuit Emulation is supported via the JTAG/ICE port. The pins TDI, TDO, TCK and TMS are dedicated to this debug function and can be connected to a host computer via the external ICE interface. In ICE Debug Mode, the ARM7TDMI core responds with a non-JTAG chip ID that identifies the microcontroller. This is not fully IEEE1149.1 compliant.

7.6 Memory Controller

The ARM7TDMI processor address space is 4G bytes. The memory controller decodes the internal 32-bit address bus and defines three address spaces:

- Internal memories in the four lowest megabytes
- Middle space reserved for the external devices (memory or peripherals) controlled by the EBI
- Internal peripherals in the four highest megabytes

In any of these address spaces, the ARM7TDMI operates in little-endian mode only.

7.6.1 Internal Memories

The AT91FR40162SB microcontroller integrates 256K bytes of internal SRAM. It is 32 bits wide and single-clock cycle accessible. Byte (8-bit), half-word (16-bit) and word (32-bit) accesses are supported and are executed within one cycle. Fetching either Thumb or ARM instructions is supported, and internal memory can store two times as many Thumb instructions as ARM instructions.

The SRAM is mapped at address 0x0 (after the Remap command), allowing ARM7TDMI exception vectors between 0x0 and 0x20 to be modified by the software.

Placing the SRAM on-chip and using the 32-bit data bus bandwidth maximizes the microcontroller performance and minimizes system power consumption. The 32-bit bus increases the effectiveness of the use of the ARM instruction set and the processing of data that is wider than 16 bits, thus making optimal use of the ARM7TDMI advanced performance.

Being able to dynamically update application software in the 256-Kbyte SRAM adds an extra dimension to the AT91FR40162SB.

The AT91FR40162SB also integrates a 2-Mbyte Flash memory that is accessed via the External Bus Interface. All data, address and control lines, except for the Chip Select signal, are connected within the device.

7.6.2 Boot Mode Select

The ARM reset vector is at address 0x0. After the NRST line is released, the ARM7TDMI executes the instruction stored at this address. This means that this address must be mapped in nonvolatile memory after the reset. The input level on the BMS pin during the last 10 clock cycles before the rising edge of the NRST selects the type of boot memory (see Table 4-1 on page 5).

If the embedded Flash memory is to be used as boot memory, the BMS input must be pulled down externally and NCS0 must be connected to NCSF externally.

The pin BMS is multiplexed with the I/O line P24 that can be programmed after reset like any standard PIO line.

Table 7-1. Boot Mode Select

BMS	Boot Memory
1	External 8-bit memory on NCS0
0	Internal or External 16-bit memory on NCS0

7.6.3 Remap Command

The ARM vectors (Reset, Abort, Data Abort, Prefetch Abort, Undefined Instruction, Interrupt, Fast Interrupt) are mapped from address 0x0 to address 0x20. In order to allow these vectors to be redefined dynamically by the software, the AT91FR40162SB uses a remap command that enables switching between the boot memory and the internal primary SRAM bank addresses. The remap command is accessible through the EBI User Interface by writing one in RCB of EBI_RCR (Remap Control Register). Performing a remap command is mandatory if access to the other external devices (connected to chip selects 1 to 7) is required. The remap operation can only be changed back by an internal reset or an NRST assertion.

7.6.4 Abort Control

The abort signal providing a Data Abort or a Prefetch Abort exception to the ARM7TDMI is asserted when accessing an undefined address in the EBI address space.

No abort is generated when reading the internal memory or by accessing the internal peripherals, whether the address is defined or not.

7.6.5 External Bus Interface

The External Bus Interface handles the accesses between addresses 0x0040 0000 and 0xFFC0 0000. It generates the signals that control access to the external devices, and can be configured from eight 1-Mbyte banks up to four 16-Mbyte banks. It supports byte, half-word and word aligned accesses.

For each of these banks, the user can program:

- Number of wait states
- Number of data float times (wait time after the access is finished to prevent any bus contention in case the device is too long in releasing the bus)
- Data bus width (8-bit or 16-bit)
- With a 16-bit wide data bus, the user can program the EBI to control one 16-bit device (Byte Access Select Mode) or two 8-bit devices in parallel that emulate a 16-bit memory (Byte Write Access Mode).

The External Bus Interface features also the Early Read Protocol, configurable for all the devices, that significantly reduces access time requirements on an external device in the case of single-clock cycle access.

In the AT91FR40162SB, the External Bus Interface connects internally to the Flash memory.

7.6.6 Flash Memory

The 2-Mbyte Flash memory is organized as 1, 048, 576 words of 16 bits each. The Flash memory is addressed as 16-bit words via the EBI. It uses address lines A1 - A20 of the processor.

The address, data and control signals, except the Flash memory enable, are internally interconnected. The user should connect the Flash memory enable (NCSF) to one of the active-low chip selects on the EBI; NCS0 must be used if the Flash memory is to be the boot memory. In addition, if the Flash memory is to be used as boot memory, the BMS input must be pulled down externally in order for the processor to perform correct 16-bit fetches after reset.

During boot, the EBI must be configured with correct number of standard wait states. As an example, five standard wait states are required when the microcontroller is running at 66 MHz.

The user must ensure that all VDDIO, VDDCORE and all GND pins are connected to their respective supplies by the shortest route. The Flash memory powers-on in read mode. Command sequences are used to place the device in other operating modes, such as program and erase.

A separate Flash memory reset input pin (NRSTF) is provided for maximum flexibility, enabling the reset operation to adapt to the application. When this input is at a logic high level, the memory is in its standard operating mode; a low level on this input halts the current memory operation and puts its outputs in a high impedance state.

The Flash memory features data polling to detect the end of a program cycle. While a program cycle is in progress, an attempted read of the last word written will return the complement of the written data on I/O7. An open-drain NBUSY output pin provides another method of detecting the end of a program or erase cycle. This pin is pulled low while program and erase cycles are in progress and is released at the completion of the cycle. A toggle bit feature provides a third means of detecting the end of a program or erase cycle.

The Flash memory is divided into 39 sectors for erase operations. To further enhance device flexibility, an Erase Suspend feature is offered. This feature puts the erase cycle on hold for an indefinite period and allows the user to read data from, or to write data to, any other sector within the same memory plane. There is no need to suspend an erase cycle if the data to be read is in the other memory plane.

The device has the capability to protect data stored in any sector. Once the data protection for a sector is enabled, the data in that sector cannot be changed while input levels lie between ground and VDDIO.

Note: This data protection does not prevent read accesses of the Flash.

A 6-byte command sequence (Enter Single Pulse Program Mode) allows the device to be written to directly, using single pulses on the write control lines. This mode (Single-pulse Programming) is exited by powering down the device or by pulsing the NRSTF pin low for a defined duration and then bringing it back to VDDIO.

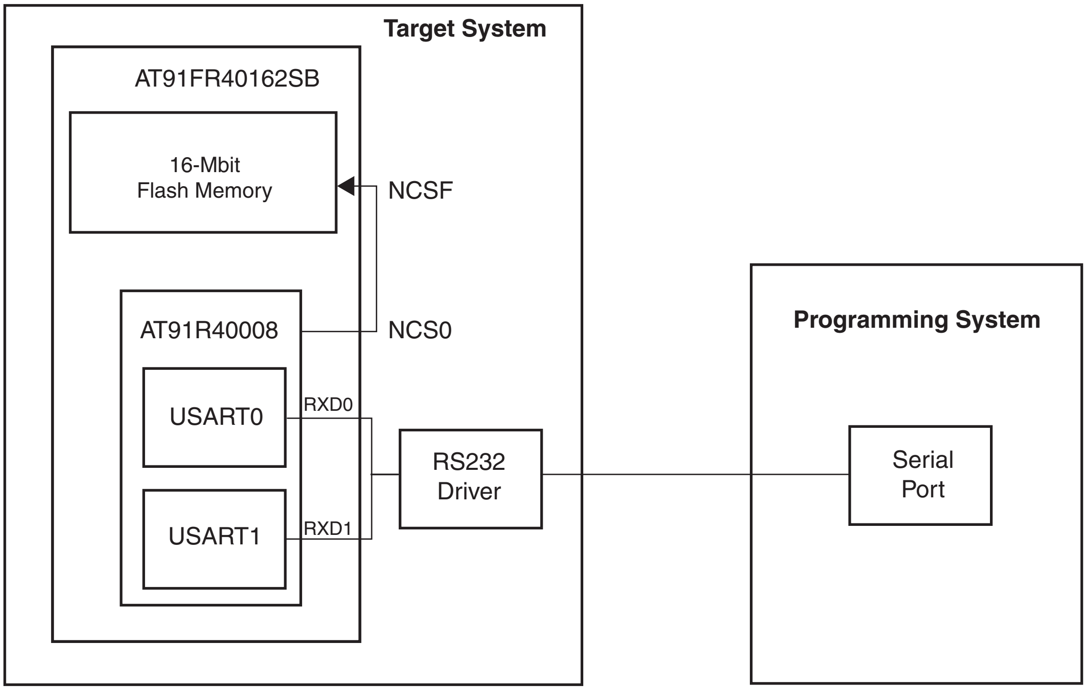
The following hardware features protect against inadvertent programming of the Flash memory:

- VDDIO Sense – if VDDIO is below a certain level, the program function is inhibited.
- VDDIO Power-on Delay – once VDDIO has reached the VDDIO sense level, the device will automatically time out a certain duration before programming.
- Program Inhibit – holding any one of OE low, CE high or WE high inhibits program cycles.
- Noise Filter – pulses of less than a certain duration on the WE or CE inputs will not initiate a program cycle.

7.7 AT91 Flash Memory Uploader (FMU) Software

All Flash-based AT91 devices are delivered with pre-programmed software called the AT91 Flash Memory Uploader, which resides in the first sector of the embedded Flash. The Flash Memory Uploader allows programming to the embedded flash through a serial port. Either of the on-chip USARTs can be used by the Flash Memory Uploader. The purpose of the AT91 Flash Memory Uploader is to provide a Flash programming solution during small and medium productiion. The FMU is “one-time usable”. This means that once the customer’s code is written in sector 0 of the Flash, the FMU is overwritten. If IAP functionality is needed, customers need to use the JTAG port or implement their own boot loader with IAP capability.

Figure 7-1. Flash Memory Uploader



7.7.1 Flash Memory Uploader Operations

The Flash Memory Uploader requires the encapsulated Flash to be used as the AT91FR40162SB boot memory and a valid clock to be applied to MCKI. After reset, the Flash Memory Uploader immediately recopies itself into the internal SRAM and jumps to it. The following operation requires this memory resource only. External accesses are performed only to program the encapsulated Flash.

When starting, PIO input change interrupts are initialized on the RXD lines of both USARTs. When an interrupt occurs, a Timer Counter channel is started. When the next input change is detected on the RXD line, the Timer Counter channel is stopped. This is how the first character length is measured and the USART can be initiated by taking into account the ratio between the device master clock speed and the actual communication baud rate speed.

The Programming System, then, can send commands and data following a proprietary protocol for the Flash device to be programmed. It is up to the Programming System to erase and program the first sector of the Flash as the last step of the operation, in order to reduce, to a minimum, the risk that the Flash Memory Uploader is erased and the power supply shuts down.

Note that in the event that the Flash Memory Uploader is erased from the first sector while the new final application is not yet programmed, and while the target system power supply is switched off, it leads to a non-recoverable error and the AT91FR40162SB cannot be re-programmed by using the Flash Memory Uploader.

7.7.2 Programming System

Atmel provides a free Host Loader that runs on an IBM® compatible PC under Windows95, Windows98 or Windows2000 operating system. It can be downloaded from the Atmel Web site and requires only a serial cable to connect the Host to the Target.

Communications can be selected on either COM1 or COM2 and the serial link speed is limited to 115200 bauds. Because the serial link is the bottleneck in this configuration, the Flash programming lasts 110 seconds per Mbyte.

Reduced programming time can be achieved by using a faster programming system. An AT91 Evaluation Board is capable of running a serial link at up to 500 Kbits/sec and can match the fastest programming allowed by the Flash, for example, about 40 seconds per Mbyte when the word programming becomes the bottleneck.

For more details about the Flash Memory Uploader protocol and the Host Loader Programming System, see the application note page of the AT91 Products at www.atmel.com.

8. Peripherals

The AT91FR40162SB peripherals are connected to the 32-bit wide Advanced Peripheral Bus.

Peripheral registers are only word accessible. Byte and half-word accesses are not supported. If a byte or a half-word access is attempted, the memory controller automatically masks the lowest address bits and generates a word access.

Each peripheral has a 16-Kbyte address space allocated (the AIC only has a 4-Kbyte address space).

8.0.1 Peripheral Registers

The following registers are common to all peripherals:

- Control Register – write only register that triggers a command when a one is written to the corresponding position at the appropriate address. Writing a zero has no effect.
- Mode Register – read/write register that defines the configuration of the peripheral. Usually has a value of 0x0 after a reset.
- Data Registers – read and/or write register that enables the exchange of data between the processor and the peripheral.
- Status Register – read only register that returns the status of the peripheral.
- Enable/Disable/Status Registers are shadow command registers. Writing a one in the Enable Register sets the corresponding bit in the Status Register. Writing a one in the Disable Register resets the corresponding bit and the result can be read in the Status Register. Writing a bit to zero has no effect. This register access method maximizes the efficiency of bit manipulation, and enables modification of a register with a single non-interruptible instruction, replacing the costly read-modify-write operation.

Unused bits in the peripheral registers must be written at 0 for upward compatibility. These bits read 0.

8.0.2 Peripheral Interrupt Control

The Interrupt Control of each peripheral is controlled from the status register using the interrupt mask. The status register bits are ANDed to their corresponding interrupt mask bits and the result is then ORed to generate the Interrupt Source signal to the Advanced Interrupt Controller.

The interrupt mask is read in the Interrupt Mask Register and is modified with the Interrupt Enable Register and the Interrupt Disable Register. The enable/disable/status (or mask) makes it possible to enable or disable peripheral interrupt sources with a non-interruptible single instruction. This eliminates the need for interrupt masking at the AIC or Core level in real-time and multi-tasking systems.

8.0.3 Peripheral Data Controller

The AT91FR40162SB has a 4-channel PDC dedicated to the two on-chip USARTs. One PDC channel is dedicated to the receiver and one to the transmitter of each USART.

The user interface of a PDC channel is integrated in the memory space of each USART. It contains a 32-bit Address Pointer Register (RPR or TPR) and a 16-bit Transfer Counter Register (RCR or TCR). When the programmed number of transfers are performed, a status bit indicating the end of transfer is set in the USART Status Register and an interrupt can be generated.

8.1 System Peripherals

8.1.1 PS: Power-saving

The power-saving feature optimizes power consumption, enabling the software to stop the ARM7TDMI clock (idle mode), restarting it when the module receives an interrupt (or reset). It also enables on-chip peripheral clocks to be enabled and disabled individually, matching power consumption and application needs.

8.1.2 AIC: Advanced Interrupt Controller

The Advanced Interrupt Controller has an 8-level priority, individually maskable, vectored interrupt controller, and drives the NIRQ and NFIQ pins of the ARM7TDMI from:

- The external fast interrupt line (FIQ)
- The three external interrupt request lines (IRQ0 - IRQ2)
- The interrupt signals from the on-chip peripherals

The AIC is extensively programmable offering maximum flexibility, and its vectoring features reduce the real-time overhead in handling interrupts.

The AIC also features a spurious vector detection feature, which reduces spurious interrupt handling to a minimum, and a protect mode that facilitates the debug capabilities.

8.1.3 PIO: Parallel I/O Controller

The AT91FR40162SB has 32 programmable I/O lines. Six pins are dedicated as general-purpose I/O pins. Other I/O lines are multiplexed with an external signal of a peripheral to optimize the use of available package pins. The PIO controller enables generation of an interrupt on input change and insertion of a simple input glitch filter on any of the PIO pins.

8.1.4 WD: Watchdog

The Watchdog is built around a 16-bit counter and is used to prevent system lock-up if the software becomes trapped in a deadlock. It can generate an internal reset or interrupt, or assert an active level on the dedicated pin NWDOVF. All programming registers are password-protected to prevent unintentional programming.

8.1.5 SF: Special Function

The AT91FR40162SB provides registers that implement the following special functions.

- Chip Identification
- RESET Status
- Protect Mode

8.2 User Peripherals

8.2.1 USART: Universal Synchronous/ Asynchronous Receiver Transmitter

The AT91FR40162SB provides two identical, full-duplex, universal synchronous/asynchronous receiver/transmitters.

Each USART has its own baud rate generator, and two dedicated Peripheral Data Controller channels. The data format includes a start bit, up to 8 data bits, an optional programmable parity bit and up to 2 stop bits.

The USART also features a Receiver Timeout register, facilitating variable length frame support when it is working with the PDC, and a Time-guard register, used when interfacing with slow remote equipment.

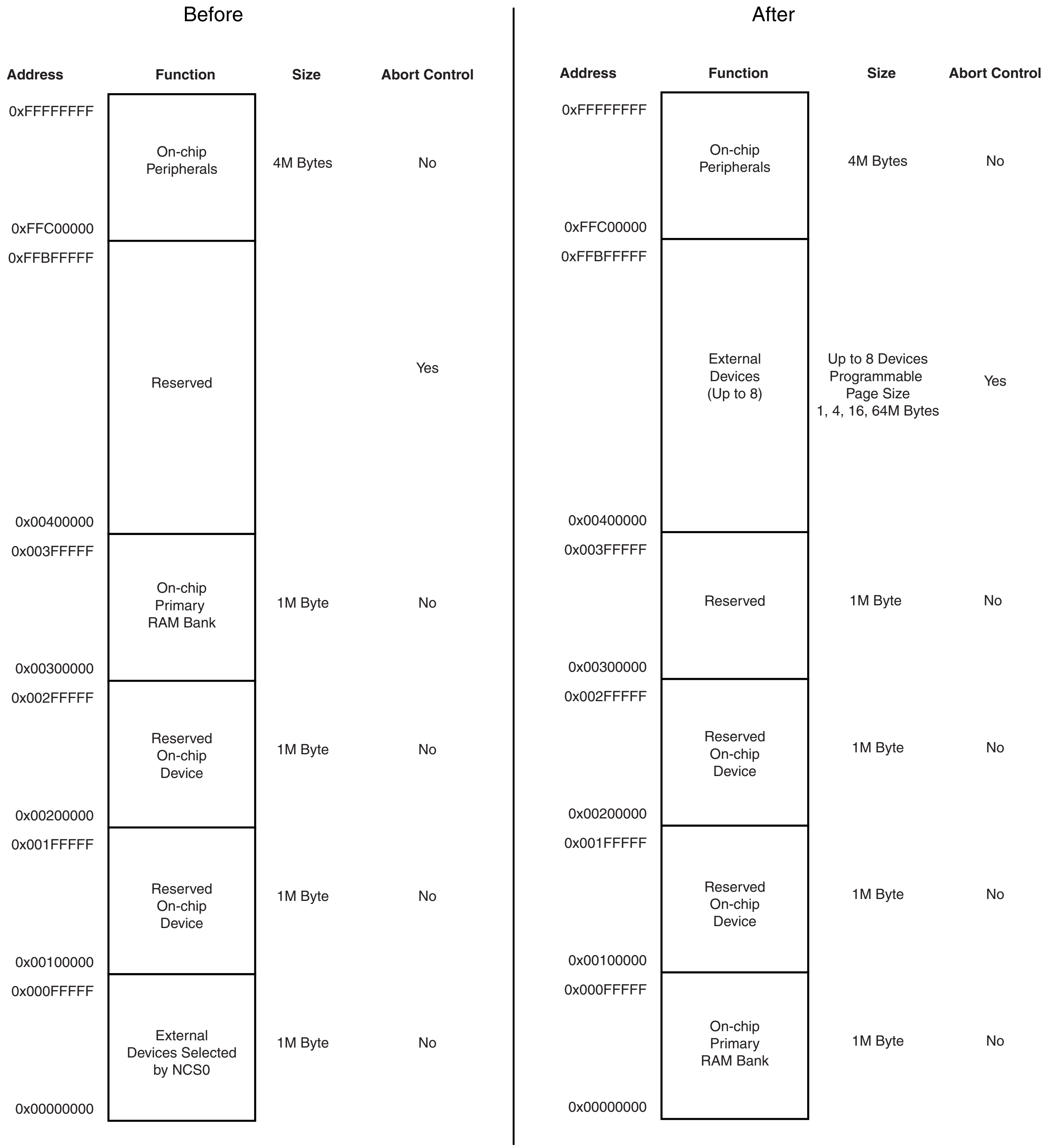
8.2.2 TC: Timer Counter

The AT91FR40162SB features a Timer Counter block that includes three identical 16-bit timer counter channels. Each channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation.

The Timer Counter can be used in Capture or Waveform mode, and all three counter channels can be started simultaneously and chained together.

9. Memory Map

Figure 9-1. AT91FR40162SB Memory Map Before and After the Remap Command



Before

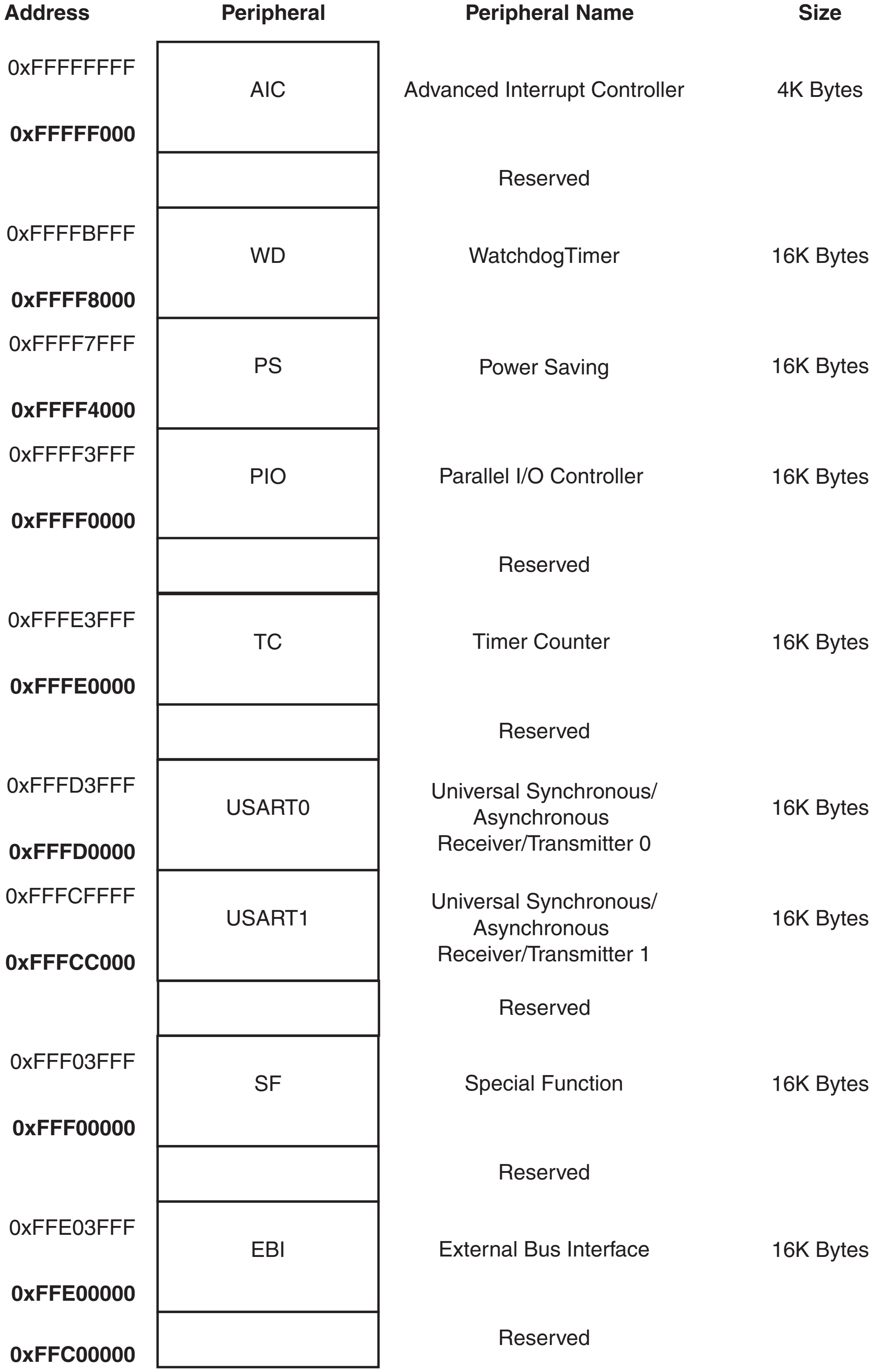
Address	Function	Size	Abort Control
0xFFFFFFFF 0xFFC00000	On-chip Peripherals	4M Bytes	No
0xFFBFFFFF 0x00400000	Reserved		Yes
0x003FFFFF 0x00300000	On-chip Primary RAM Bank	1M Byte	No
0x002FFFFF 0x00200000	Reserved On-chip Device	1M Byte	No
0x001FFFFF 0x00100000	Reserved On-chip Device	1M Byte	No
0x000FFFFF 0x00000000	External Devices Selected by NCS0	1M Byte	No

After

Address	Function	Size	Abort Control
0xFFFFFFFF 0xFFC00000	On-chip Peripherals	4M Bytes	No
0xFFBFFFFF 0x00400000	External Devices (Up to 8)	Up to 8 Devices Programmable Page Size 1, 4, 16, 64M Bytes	Yes
0x003FFFFF 0x00300000	Reserved	1M Byte	No
0x002FFFFF 0x00200000	Reserved On-chip Device	1M Byte	No
0x001FFFFF 0x00100000	Reserved On-chip Device	1M Byte	No
0x000FFFFF 0x00000000	On-chip Primary RAM Bank	1M Byte	No

10. Peripheral Memory Map

Figure 10-1. Peripheral Memory Map



Address	Peripheral	Peripheral Name	Size
0xFFFFFFFF **0xFFFFF000**	AIC	Advanced Interrupt Controller	4K Bytes
		Reserved	
0xFFFFBFFF **0xFFFF8000**	WD	WatchdogTimer	16K Bytes
0xFFFF7FFF **0xFFFF4000**	PS	Power Saving	16K Bytes
0xFFFF3FFF **0xFFFF0000**	PIO	Parallel I/O Controller	16K Bytes
		Reserved	
0xFFFE3FFF **0xFFFE0000**	TC	Timer Counter	16K Bytes
		Reserved	
0xFFFD3FFF **0xFFFD0000**	USART0	Universal Synchronous/ Asynchronous Receiver/Transmitter 0	16K Bytes
0xFFFCFFFF **0xFFFCC000**	USART1	Universal Synchronous/ Asynchronous Receiver/Transmitter 1	16K Bytes
		Reserved	
0xFFF03FFF **0xFFF00000**	SF	Special Function	16K Bytes
		Reserved	
0xFFE03FFF **0xFFE00000**	EBI	External Bus Interface	16K Bytes
0xFFC00000		Reserved	

11. EBI: External Bus Interface

The EBI generates the signals that control the access to the external memory or peripheral devices. The EBI is fully-programmable and can address up to 64M bytes. It has eight chip selects and a 24-bit address bus, the upper four bits of which are multiplexed with a chip select.

The 16-bit data bus can be configured to interface with 8- or 16-bit external devices. Separate read and write control signals allow for direct memory and peripheral interfacing.

The EBI supports different access protocols allowing single-clock cycle memory accesses.

The main features are:

- External memory mapping
- Up to 8 chip select lines
- 8- or 16-bit data bus
- Byte write or byte select lines
- Remap of boot memory
- Two different read protocols
- Programmable wait state generation
- External wait request
- Programmable data float time

Section 11.11 “EBI User Interface” on page 46 describes the EBI User Interface.

11.1 External Memory Mapping

The memory map associates the internal 32-bit address space with the external 24-bit address bus.

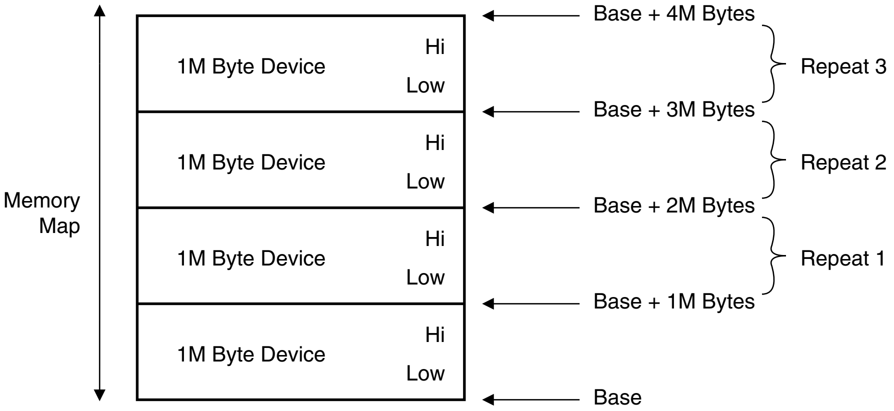
The memory map is defined by programming the base address and page size of the external memories (see “EBI User Interface” and the “EBI Chip Select Register” describing EBI_CSR0 to EBI_CSR7). Note that A0 - A23 is only significant for 8-bit memory; A1 - A23 is used for 16-bit memory.

If the physical memory device is smaller than the programmed page size, it wraps around and appears to be repeated within the page. The EBI correctly handles any valid access to the memory device within the page (see Figure 11-1 on page 23).

In the event of an access request to an address outside any programmed page, an Abort signal is generated. Two types of Abort are possible: instruction prefetch abort and data abort. The corresponding exception vector addresses are respectively 0x0000000C and 0x00000010. It is up to the system programmer to program the error handling routine to use in case of an Abort (see the ARM7TDMI datasheet for further information).

If two chip selects are defined as having the same base address, an access to the overlapping address space asserts both NCS lines. The Chip Select Register with the smaller number defines the characteristics of the external access and the behavior of the control signals.

Figure 11-1. External Memory Smaller than Page Size



11.2 External Bus Interface Pin Description

Table 11-1. EBI Pin Description

Name	Description	Type
A0 - A23	Address bus (output)	Output
D0 - D15	Data bus (input/output)	I/O
NCS0 - NCS3	Active low chip selects (output)	Output
CS4 - CS7	Active high chip selects (output)	Output
NRD	Read enable (output)	Output
NWR0 - NWR1	Lower and upper write enable (output)	Output
NOE	Output enable (output)	Output
NWE	Write enable (output)	Output
NUB, NLB	Upper and lower byte select (output)	Output
NWAIT	Wait request (input)	Input

The following table shows how certain EBI signals are multiplexed:

Table 11-2. EBI Signals

Multiplexed Signals		Functions
A23 - A20	CS4 - CS7	Allows from 4 to 8 chip select lines to be used
A0	NLB	8- or 16-bit data bus
NRD	NOE	Byte write or byte select access
NWR0	NWE	Byte write or byte select access
NWR1	NUB	Byte write or byte select access

11.3 Chip Select Lines

The EBI provides up to eight chip select lines:

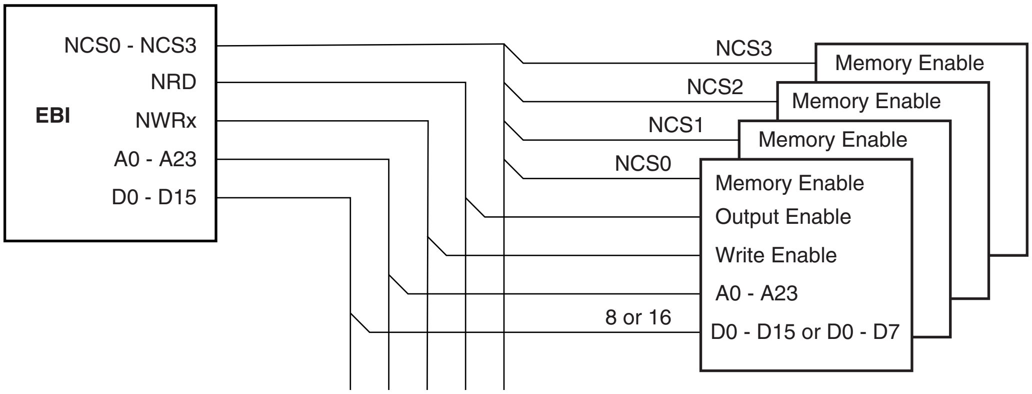
- Chip select lines NCS0 - NCS3 are dedicated to the EBI (not multiplexed).
- Chip select lines CS4 - CS7 are multiplexed with the top four address lines A23 - A20.

By exchanging address lines for chip select lines, the user can optimize the EBI to suit the external memory requirements: more external devices or larger address range for each device.

The selection is controlled by the ALE field in EBI_MCR (Memory Control Register). The following combinations are possible:

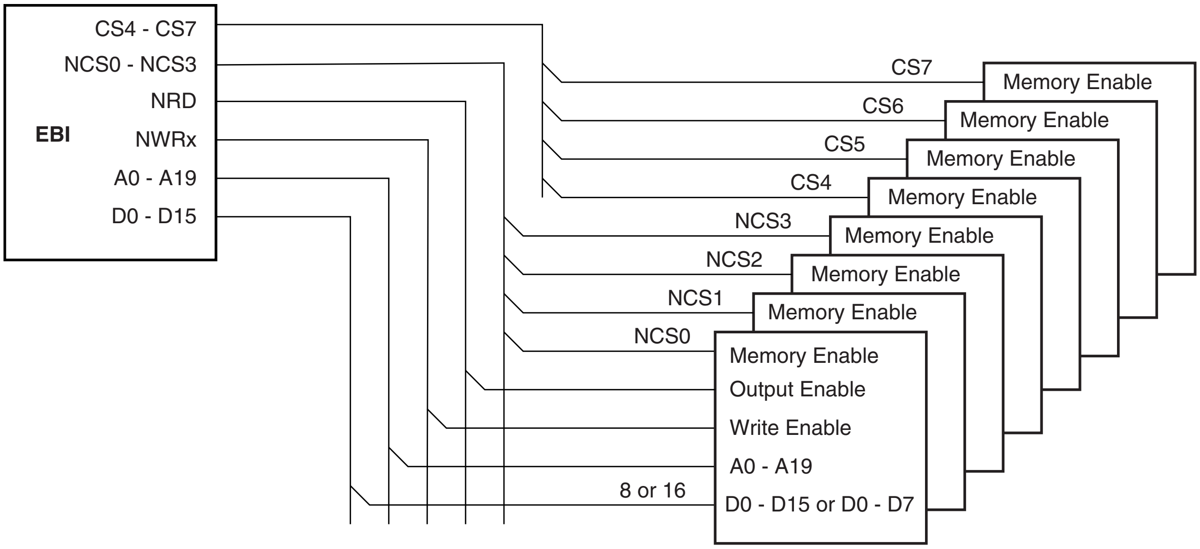
A20, A21, A22, A23 (configuration by default)
A20, A21, A22, CS4
A20, A21, CS5, CS4
A20, CS6, CS5, CS4
CS7, CS6, CS5, CS4

Figure 11-2. Memory Connections for Four External Devices



Note: For four external devices, the maximum address space per device is 16M bytes.

Figure 11-3. Memory Connections for Eight External Devices



Note: For eight external devices, the maximum address space per device is 1M byte.

11.4 Data Bus Width

A data bus width of 8 or 16 bits can be selected for each chip select. This option is controlled by the DBW field in the EBI_CSR (Chip Select Register) for the corresponding chip select.

Figure 11-4 shows how to connect a 512K x 8-bit memory on NCS2.

Figure 11-4. Memory Connection for an 8-bit Data Bus

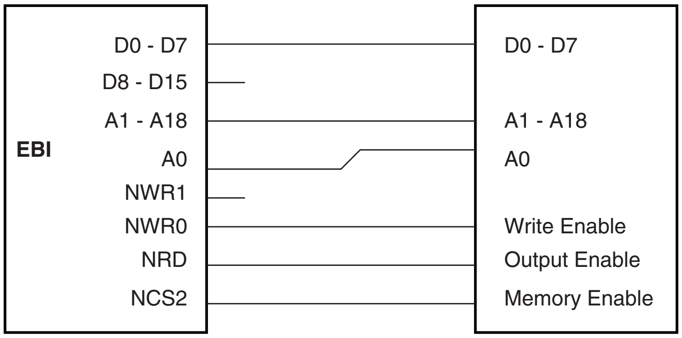
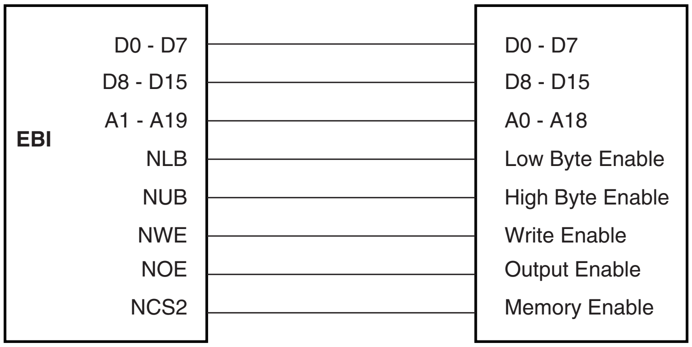


Figure 11-5 shows how to connect a 512K x 16-bit memory on NCS2.

Figure 11-5. Memory Connection for a 16-bit Data Bus



11.5 Byte Write or Byte Select Access

Each chip select with a 16-bit data bus can operate with one of two different types of write access:

- Byte Write Access supports two byte write and a single read signal.
- Byte Select Access selects upper and/or lower byte with two byte select lines, and separate read and write signals.

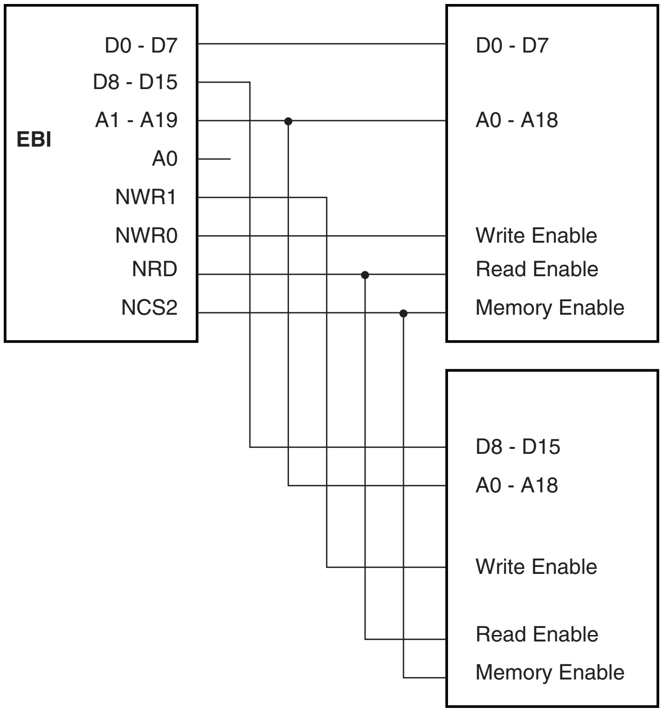
This option is controlled by the BAT field in the EBI_CSR (Chip Select Register) for the corresponding chip select.

Byte Write Access is used to connect 2 x 8-bit devices as a 16-bit memory page.

- The signal A0/NLB is not used.
- The signal NWR1/NUB is used as NWR1 and enables upper byte writes.
- The signal NWR0/NWE is used as NWR0 and enables lower byte writes.
- The signal NRD/NOE is used as NRD and enables half-word and byte reads.

Figure 11-6 shows how to connect two 512K x 8-bit devices in parallel on NCS2.

Figure 11-6. Memory Connection for 2 x 8-bit Data Busses



Byte Select Access is used to connect 16-bit devices in a memory page.

- The signal A0/NLB is used as NLB and enables the lower byte for both read and write operations.
- The signal NWR1/NUB is used as NUB and enables the upper byte for both read and write operations.
- The signal NWR0/NWE is used as NWE and enables writing for byte or half word.
- The signal NRD/NOE is used as NOE and enables reading for byte or half word.

Figure 11-7 shows how to connect a 16-bit device with byte and half-word access (e.g. 16-bit SRAM) on NCS2.

Figure 11-7. Connection for a 16-bit Data Bus with Byte and Half-word Access

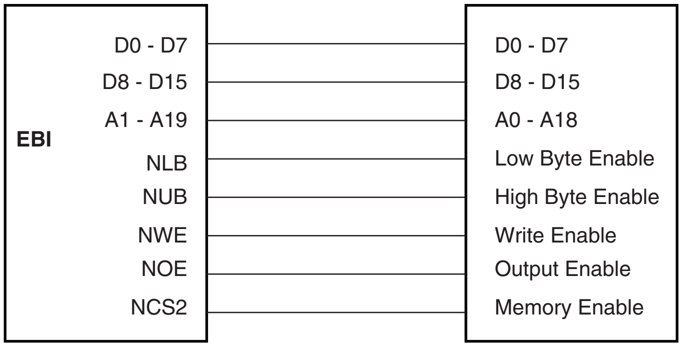
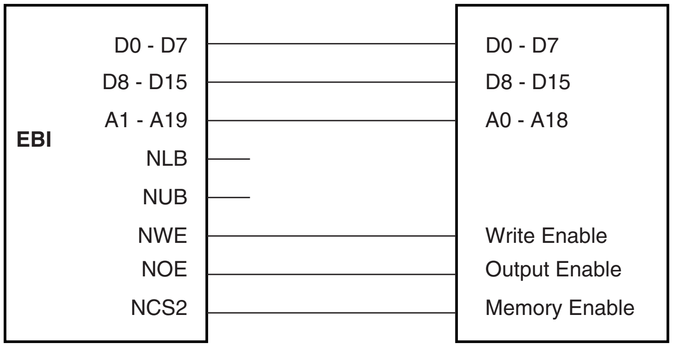


Figure 11-8 shows how to connect a 16-bit device without byte access (e.g. Flash) on NCS2.

Figure 11-8. Connection for a 16-bit Data Bus without Byte Write Capability.



11.6 Boot on NCS0

Depending on the device and the BMS pin level during the reset, the user can select either an 8-bit or 16-bit external memory device connected on NCS0 as the Boot Memory. In this case, EBI_CSR0 (Chip Select Register 0) is reset at the following configuration for chip select 0:

- 8 wait states (WSE = 1, NWS = 7)
- 8-bit or 16-bit data bus width, depending on BMS

Byte access type and number of data float time are respectively set to Byte Write Access and 0. With a non-volatile memory interface, any values can be programmed for these parameters.

Before the remap command, the user can modify the chip select 0 configuration, programming the EBI_CSR0 with exact boot memory characteristics. the base address becomes effective after the remap command, but the new number of wait states can be changed immediately. This is useful if a boot sequence needs to be faster.

11.7 Read Protocols

The EBI provides two alternative protocols for external memory read access: standard and early read. The difference between the two protocols lies in the timing of the NRD (read cycle) waveform.

The protocol is selected by the DRP field in EBI_MCR (Memory Control Register) and is valid for all memory devices. Standard read protocol is the default protocol after reset.

Note: In the following waveforms and descriptions, **NRD** represents NRD and NOE since the two signals have the same waveform. Likewise, **NWE** represents NWE, NWR0 and NWR1 unless NWR0 and NWR1 are otherwise represented. **ADDR** represents A0 - A23 and/or A1 - A23.

11.7.1 Standard Read Protocol

Standard read protocol implements a read cycle in which NRD and NWE are similar. Both are active during the second half of the clock cycle. The first half of the clock cycle allows time to ensure completion of the previous access as well as the output of address and NCS before the read cycle begins.

During a standard read protocol, external memory access, NCS is set low and ADDR is valid at the beginning of the access while NRD goes low only in the second half of the master clock cycle to avoid bus conflict (see Figure 11-9). NWE is the same in both protocols. NWE always goes low in the second half of the master clock cycle (see Figure 11-10).

11.7.2 Early Read Protocol

Early read protocol provides more time for a read access from the memory by asserting NRD at the beginning of the clock cycle. In the case of successive read cycles in the same memory, NRD remains active continuously. Since a read cycle normally limits the speed of operation of the external memory system, early read protocol can allow a faster clock frequency to be used. However, an extra wait state is required in some cases to avoid contentions on the external bus.

11.7.3 Early Read Wait State

In early read protocol, an early read wait state is automatically inserted when an external write cycle is followed by a read cycle to allow time for the write cycle to end before the subsequent read cycle begins (see Figure 11-11). This wait state is generated in addition to any other programmed wait states (i.e. data float wait).

No wait state is added when a read cycle is followed by a write cycle, between consecutive accesses of the same type or between external and internal memory accesses.

Early read wait states affect the external bus only. They do not affect internal bus timing.

Figure 11-9. Standard Read Protocol

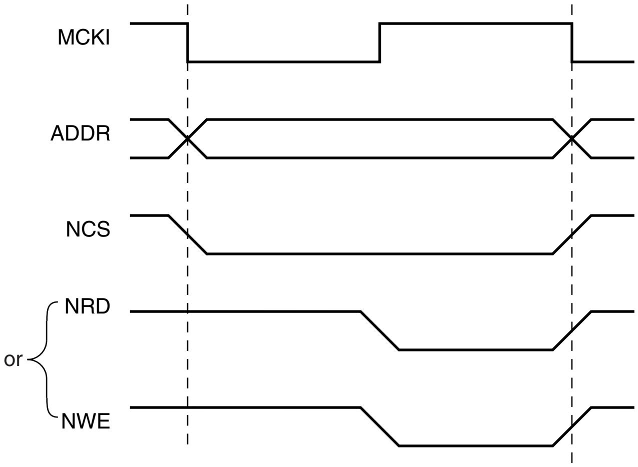


Figure 11-10. Early Read Protocol

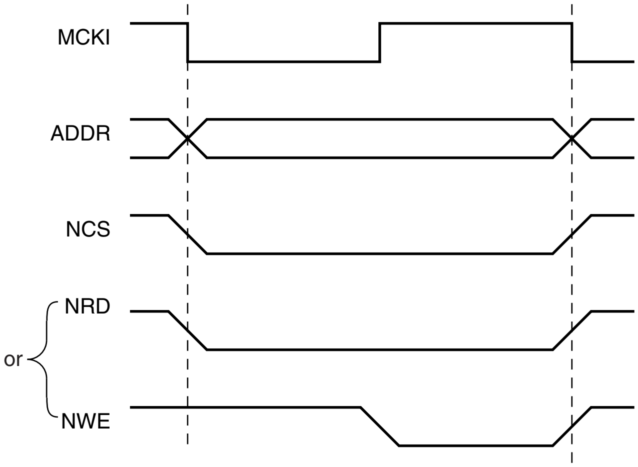
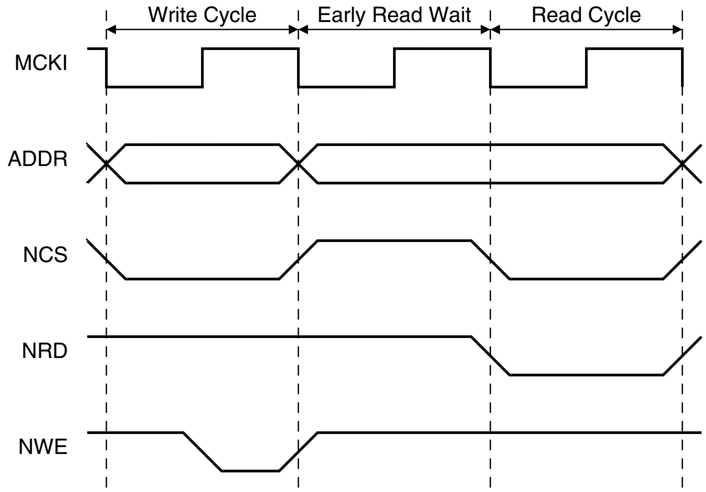


Figure 11-11. Early Read Wait State

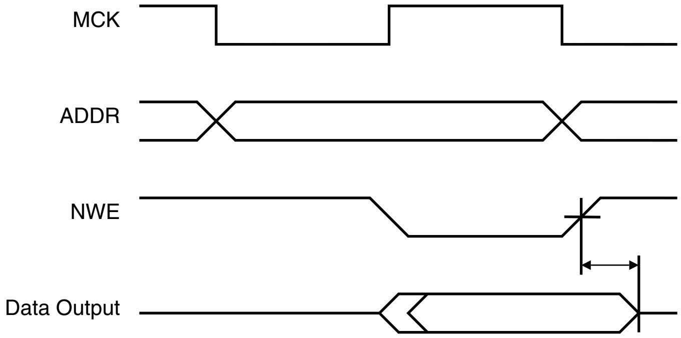


11.8 Write Data Hold Time

During write cycles in both protocols, output data becomes valid after the falling edge of the NWE signal and remains valid after the rising edge of NWE, as illustrated in Figure 11-12. The external NWE waveform (on the NWE pin) is used to control the output data timing to guarantee this operation.

It is therefore necessary to avoid excessive loading of the NWE pins, which could delay the write signal too long and cause a contention with a subsequent read cycle in standard protocol.

Figure 11-12. Data Hold Time



In early read protocol the data can remain valid longer than in standard read protocol due to the additional wait cycle which follows a write access.

11.9 Wait States

The EBI can automatically insert wait states. The different types of wait states are listed below:

- Standard wait states
- Data float wait states
- External wait states
- Chip select change wait states
- Early read wait states (see Section 11.7 "Read Protocols" on page 28)

11.9.1 Standard Wait States

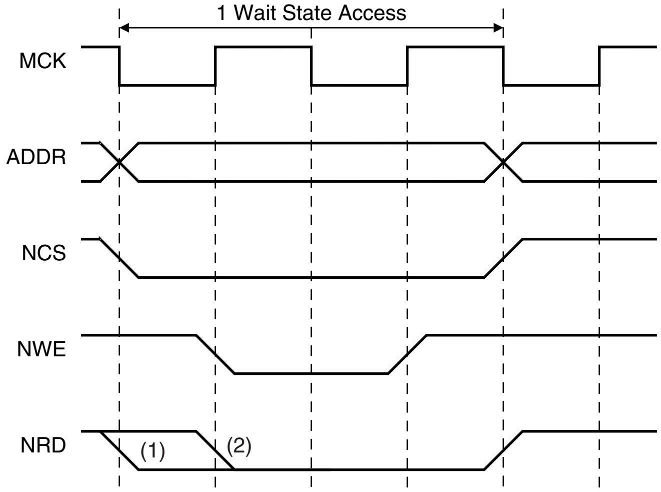
Each chip select can be programmed to insert one or more wait states during an access on the corresponding device. This is done by setting the WSE field in the corresponding EBI_CSR. The number of cycles to insert is programmed in the NWS field in the same register.

Below is the correspondence between the number of standard wait states programmed and the number of cycles during which the NWE pulse is held low:

0 wait states	1/2 cycle
1 wait state	1 cycle

For each additional wait state programmed, an additional cycle is added.

Figure 11-13. One Wait State Access



Notes: 1. Early Read Protocol
2. Standard Read Protocol

11.9.2 Data Float Wait State

Some memory devices are slow to release the external bus. For such devices it is necessary to add wait states (data float waits) after a read access before starting a write access or a read access to a different external memory.

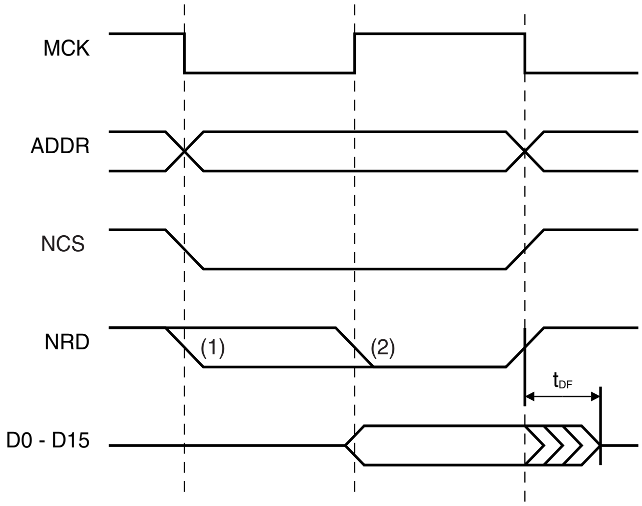
The Data Float Output Time (t_{DF}) for each external memory device is programmed in the TDF field of the EBI_CSR register for the corresponding chip select. The value (0 - 7 clock cycles) indicates the number of data float waits to be inserted and represents the time allowed for the data output to go high impedance after the memory is disabled.

Data float wait states do not delay internal memory accesses. Hence, a single access to an external memory with long t_{DF} will not slow down the execution of a program from internal memory.

The EBI keeps track of the programmed external data float time during internal accesses, to ensure that the external memory system is not accessed while it is still busy.

Internal memory accesses and consecutive accesses to the same external memory do not have added Data Float wait states.

Figure 11-14. Data Float Output Time



Notes: 1. Early Read Protocol

2. Standard Read Protocol

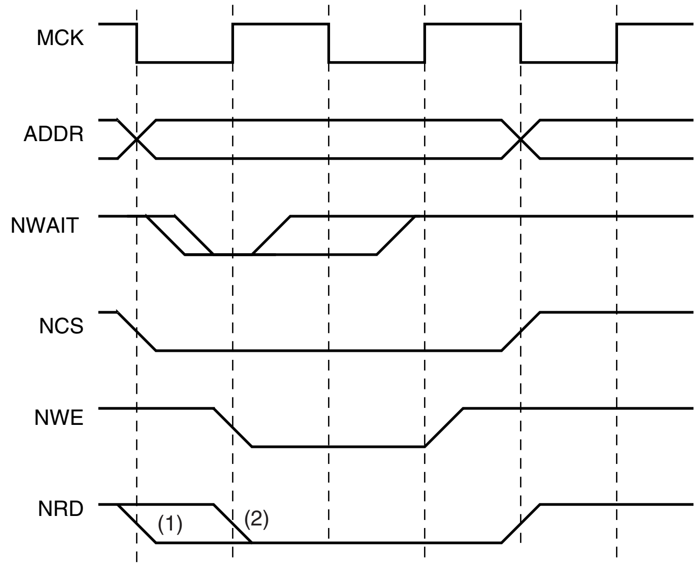
11.9.3 External Wait

The NWAIT input can be used to add wait states at any time. NWAIT is active low and is detected on the rising edge of the clock.

If NWAIT is low at the rising edge of the clock, the EBI adds a wait state and changes neither the output signals nor its internal counters and state. When NWAIT is de-asserted, the EBI finishes the access sequence.

The NWAIT signal must meet setup and hold requirements on the rising edge of the clock.

Figure 11-15. External Wait

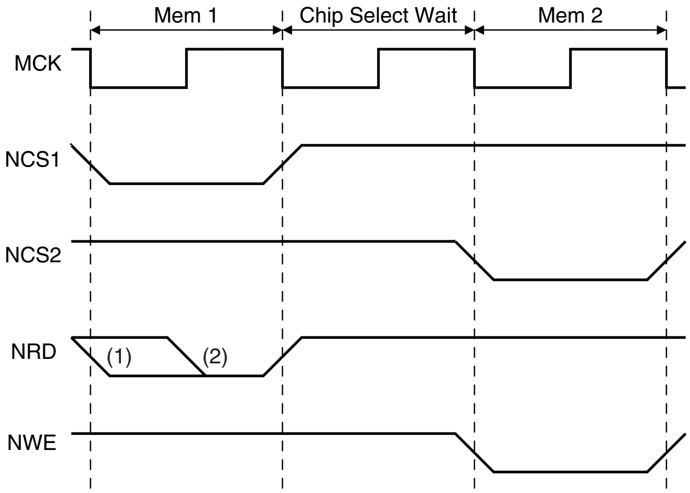


Notes: 1. Early Read Protocol
2. Standard Read Protocol

11.9.4 Chip Select Change Wait States

A chip select wait state is automatically inserted when consecutive accesses are made to two different external memories (if no wait states have already been inserted). If any wait states have already been inserted, (e.g., data float wait) then none are added.

Figure 11-16. Chip Select Wait



Notes: 1. Early Read Protocol
2. Standard Read Protocol

11.10 Memory Access Waveforms

Figure 11-17 through Figure 11-20 show examples of the two alternative protocols for external memory read access.

Figure 11-17. Standard Read Protocol without t_{DF}

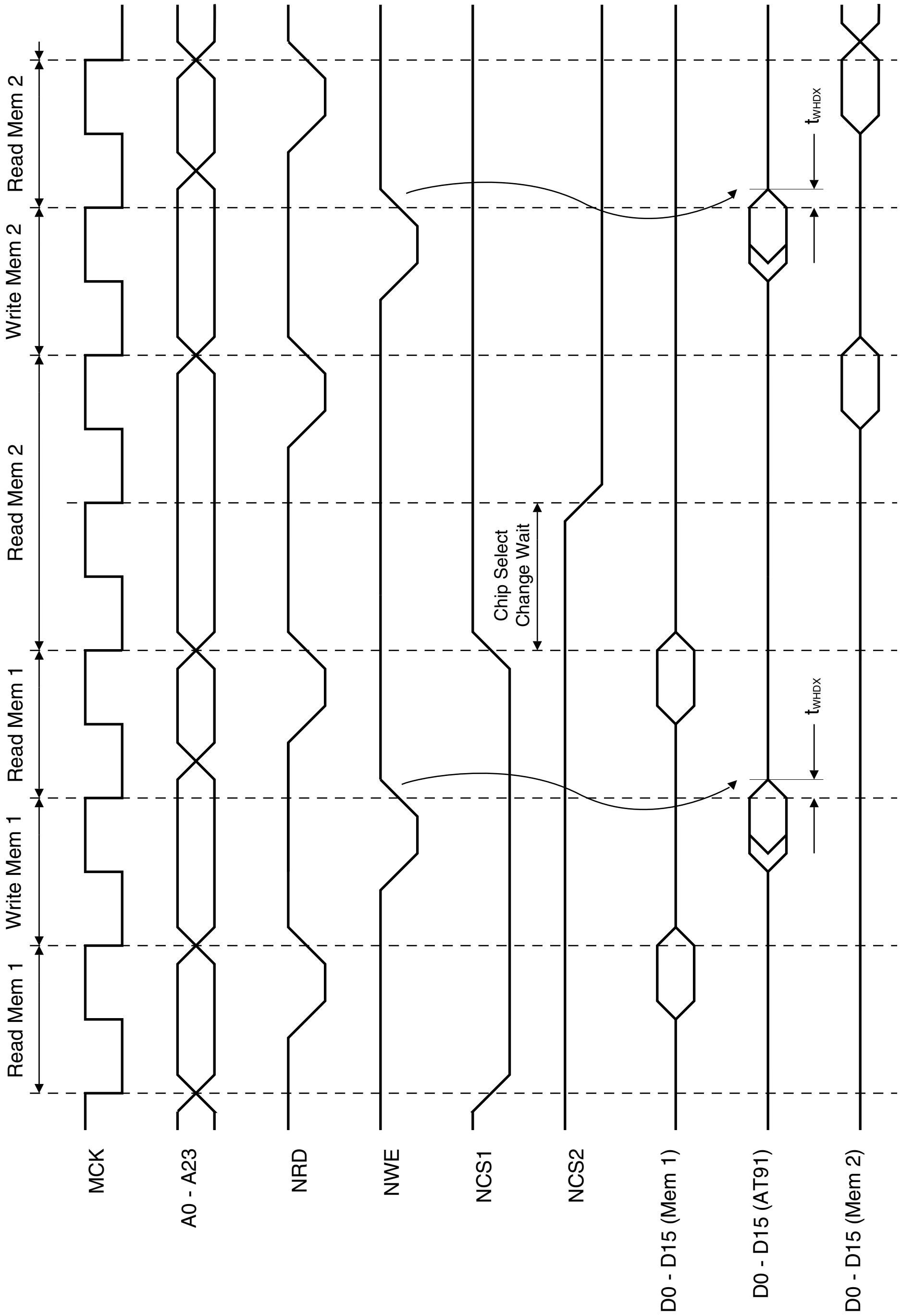


Figure 11-18. Early Read Protocol Without t_{DF}

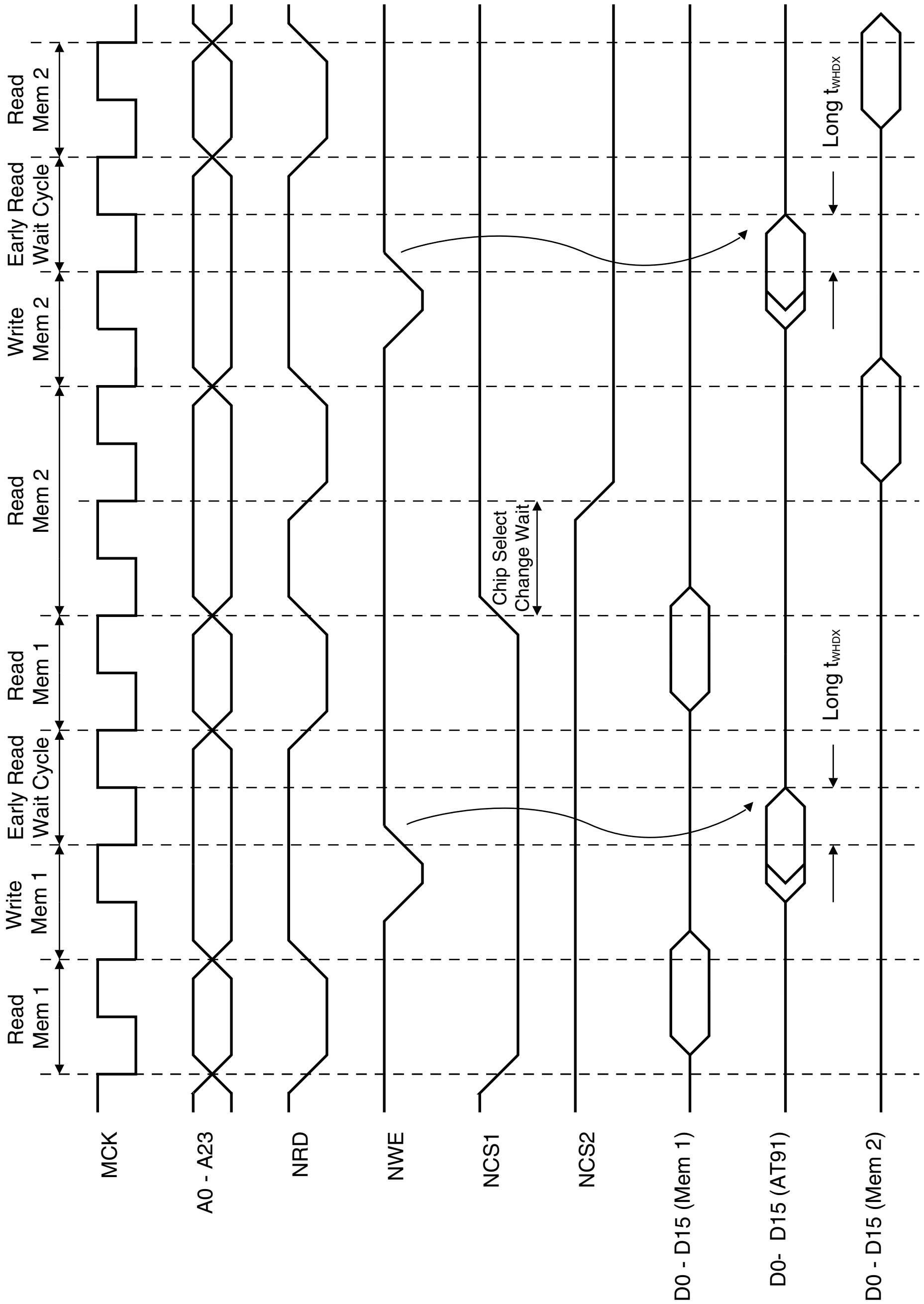


Figure 11-19. Standard Read Protocol with t_{DF}

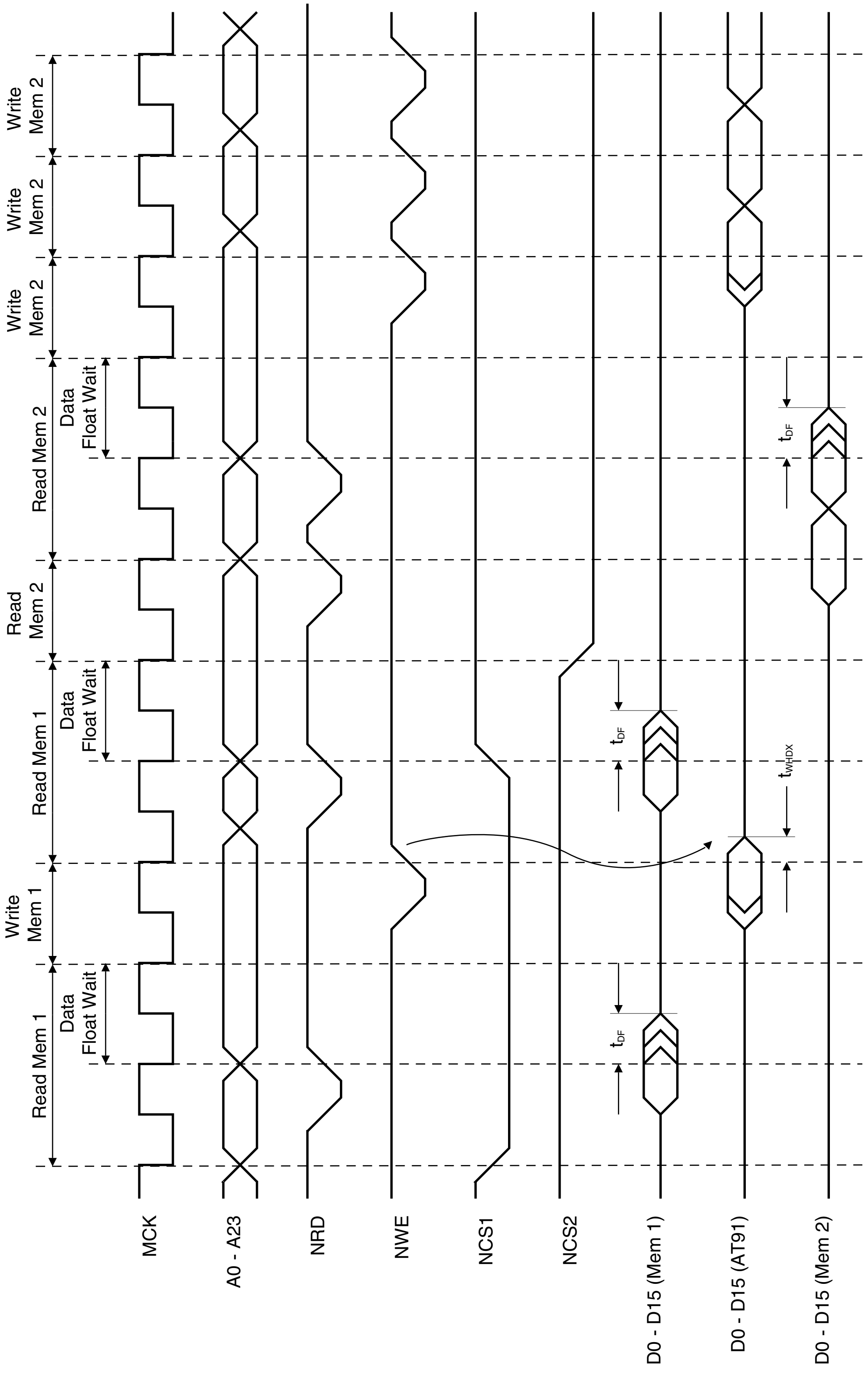


Figure 11-20. Early Read Protocol With t_{DF}

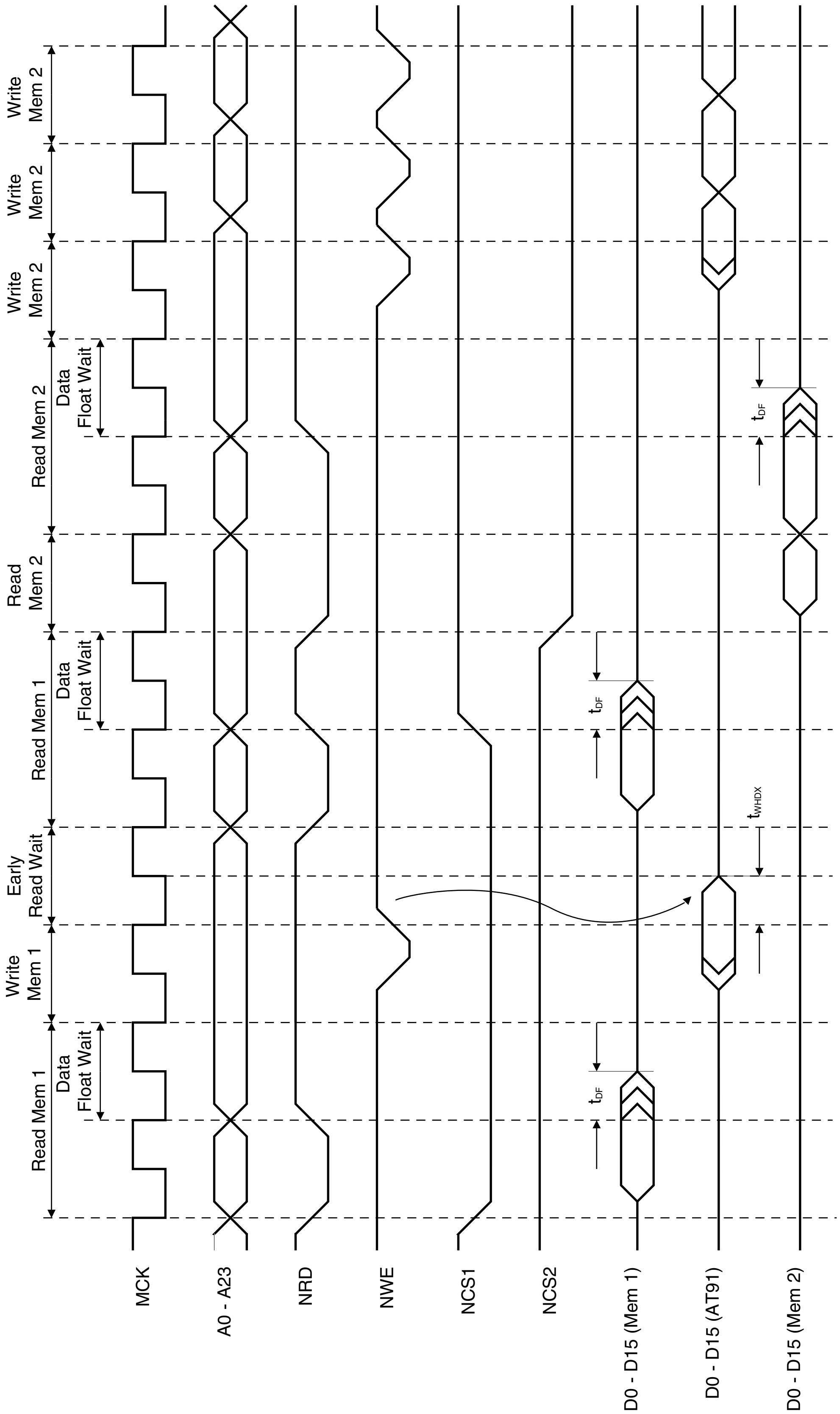


Figure 11-21 through Figure 11-27 show the timing cycles and wait states for read and write access to the various AT91FR40162SB external memory devices. The configurations described are shown in the following table:

Table 11-3. Memory Access Waveforms

Figure Number	Number of Wait States	Bus Width	Size of Data Transfer
Figure 11-21	0	16	Word
Figure 11-22	1	16	Word
Figure 11-23	1	16	Half-word
Figure 11-24	0	8	Word
Figure 11-25	1	8	Half-word
Figure 11-26	1	8	Byte
Figure 11-27	0	16	Byte

Figure 11-21. 0 Wait States, 16-bit Bus Width, Word Transfer

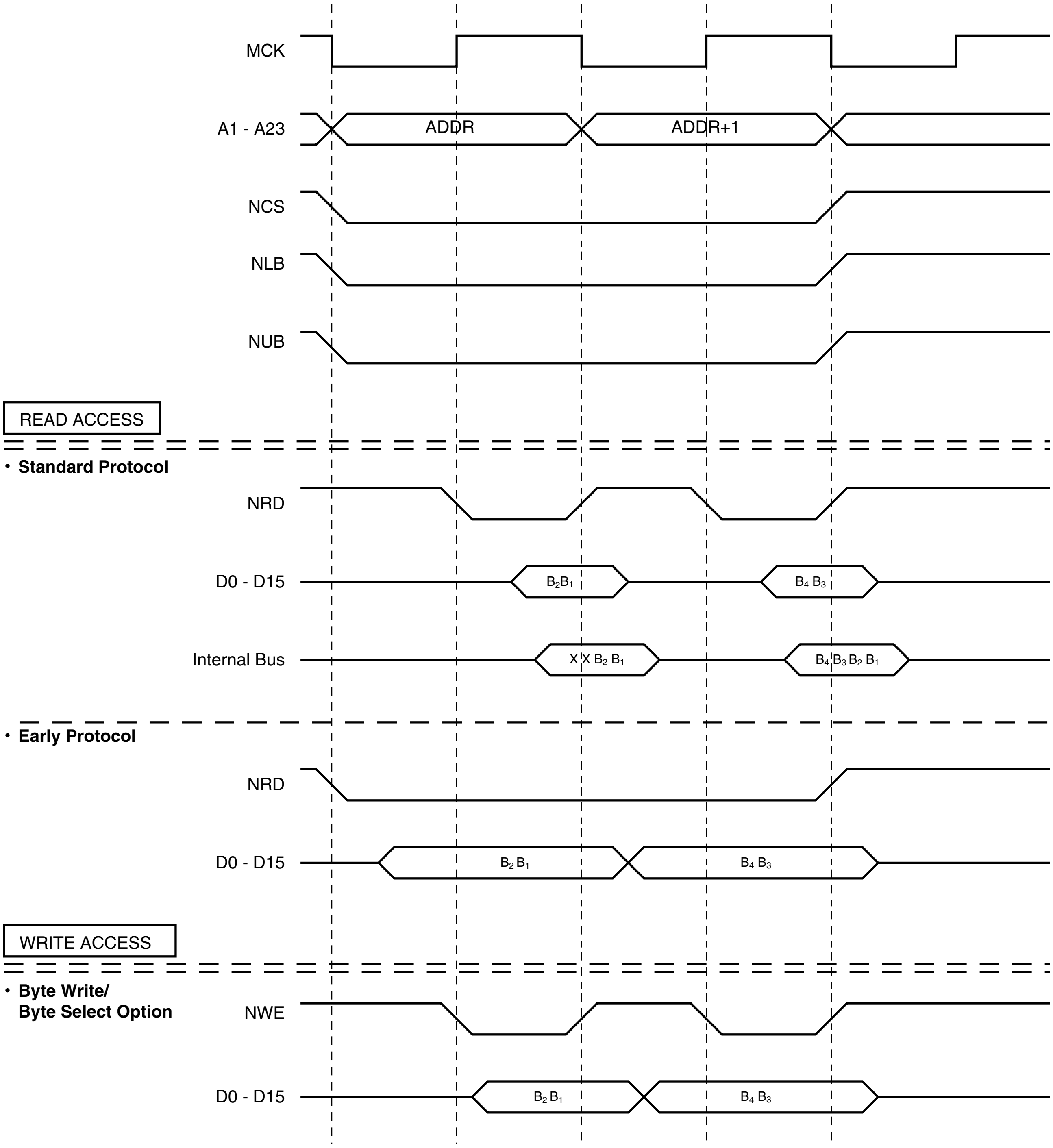


Figure 11-22. 1 Wait, 16-bit Bus Width, Word Transfer

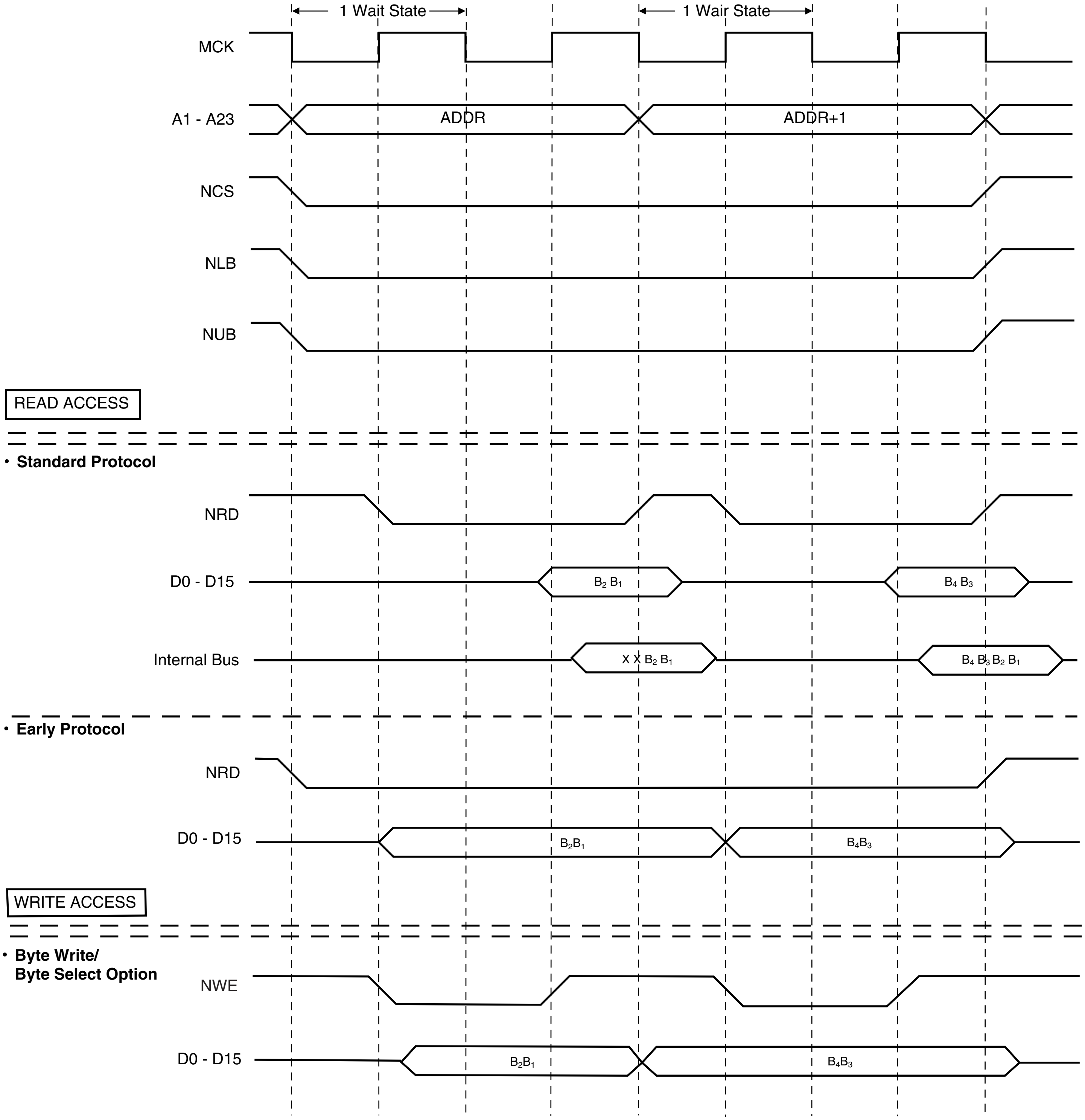


Figure 11-23. 1 Wait State, 16-bit Bus Width, Half-word Transfer

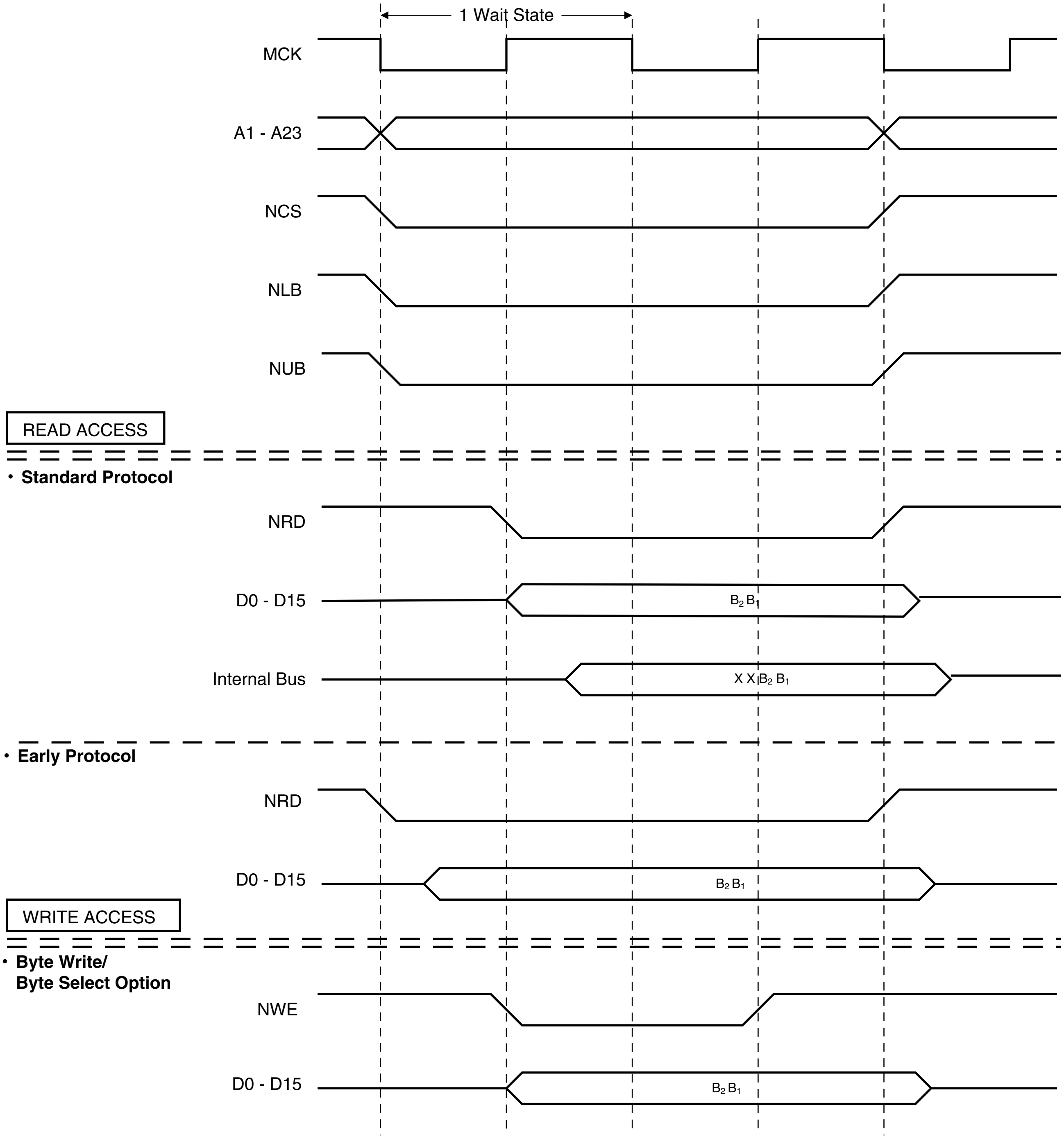


Figure 11-24. 0 Wait States, 8-bit Bus Width, Word Transfer

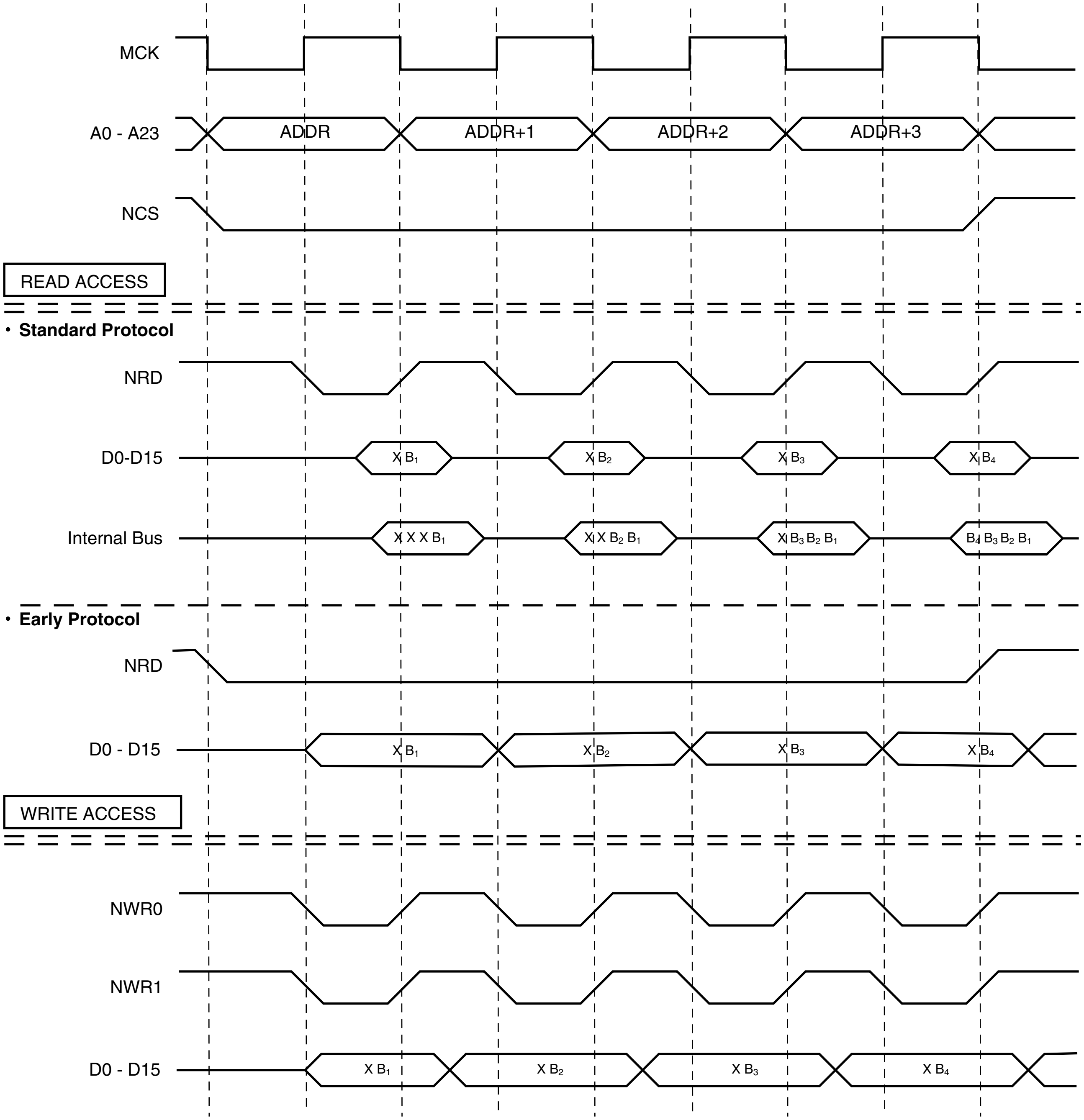


Figure 11-25. 1 Wait State, 8-bit Bus Width, Half-word Transfer

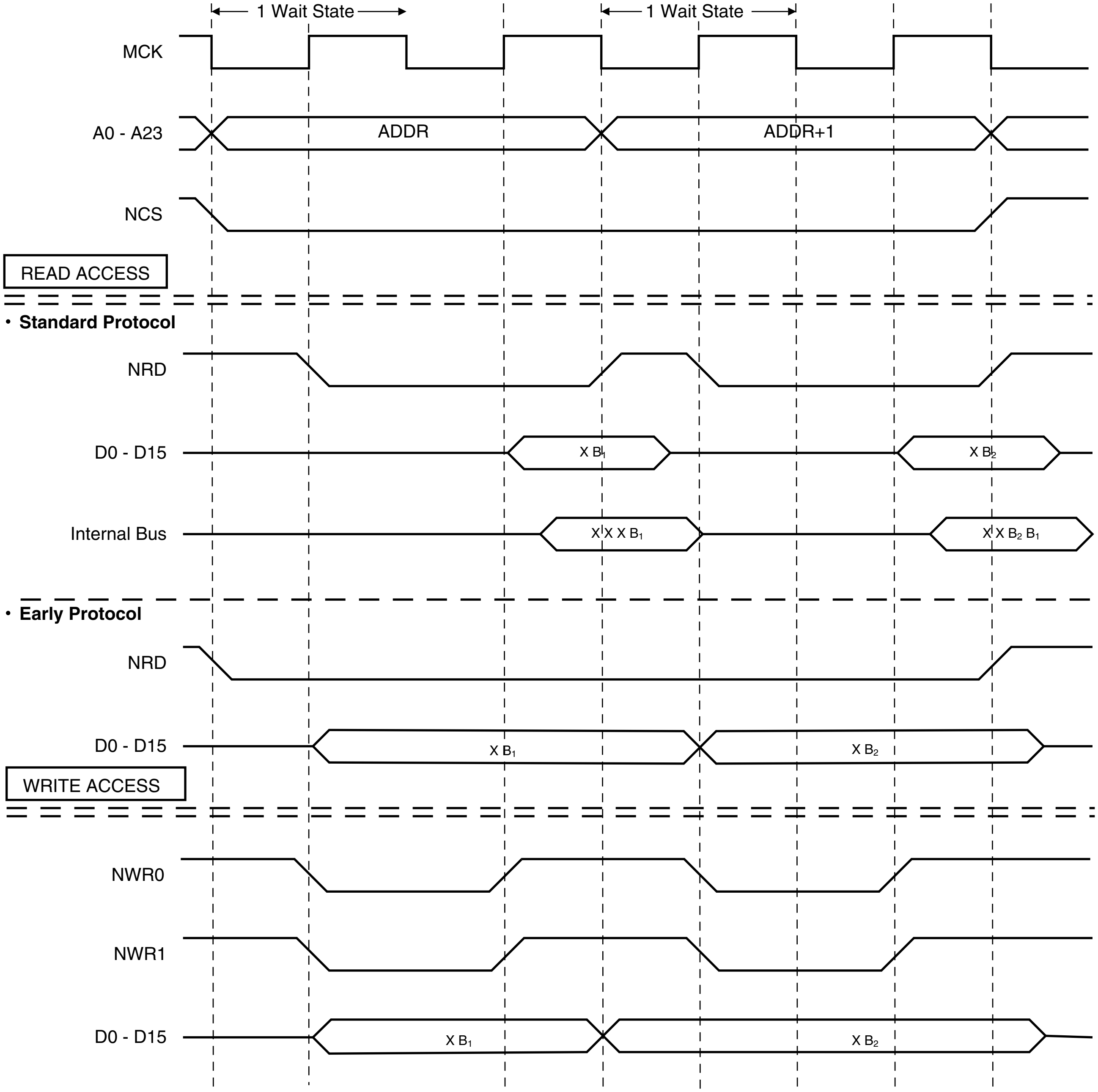


Figure 11-26. 1 Wait State, 8-bit Bus Width, Byte Transfer

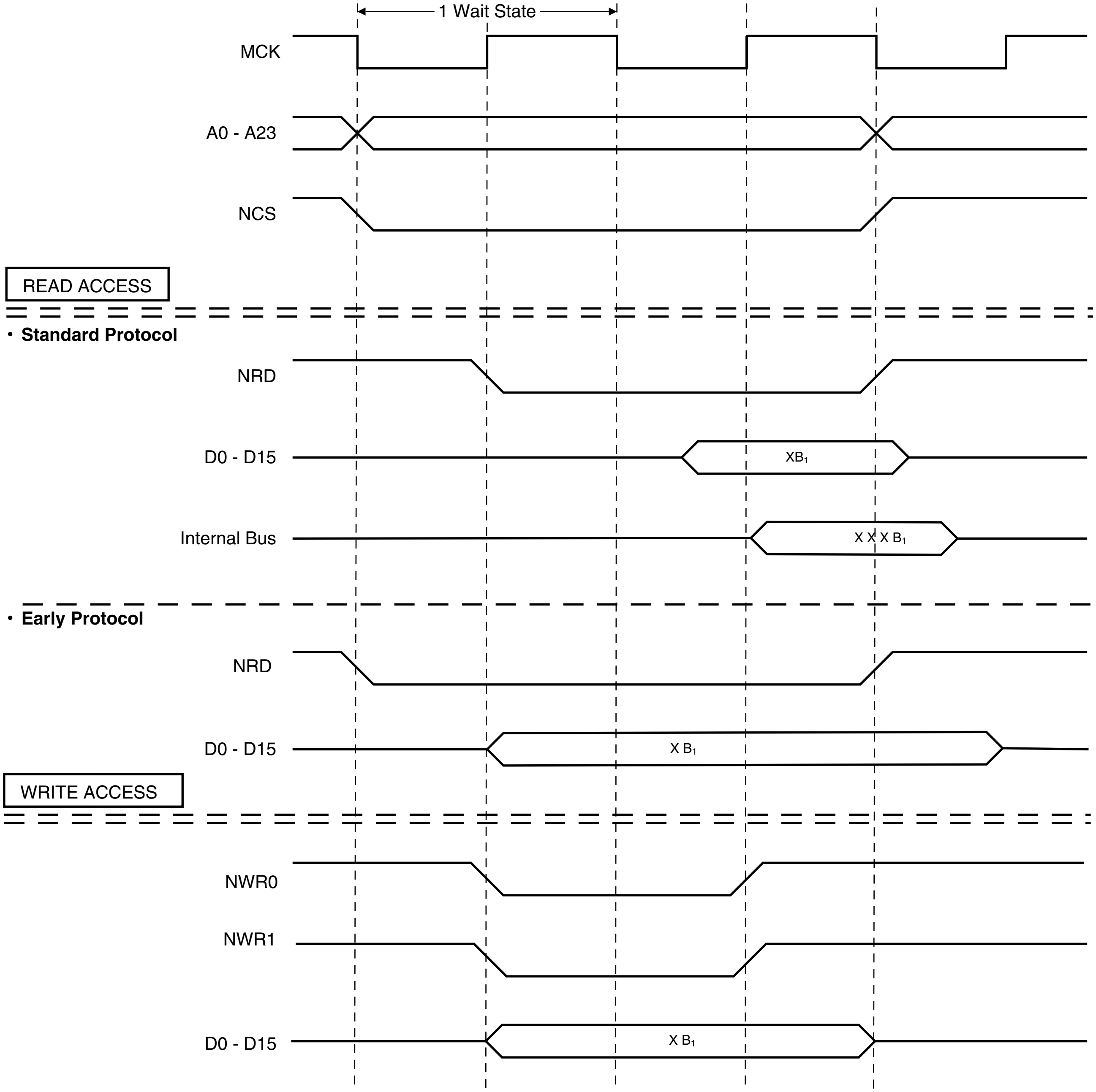
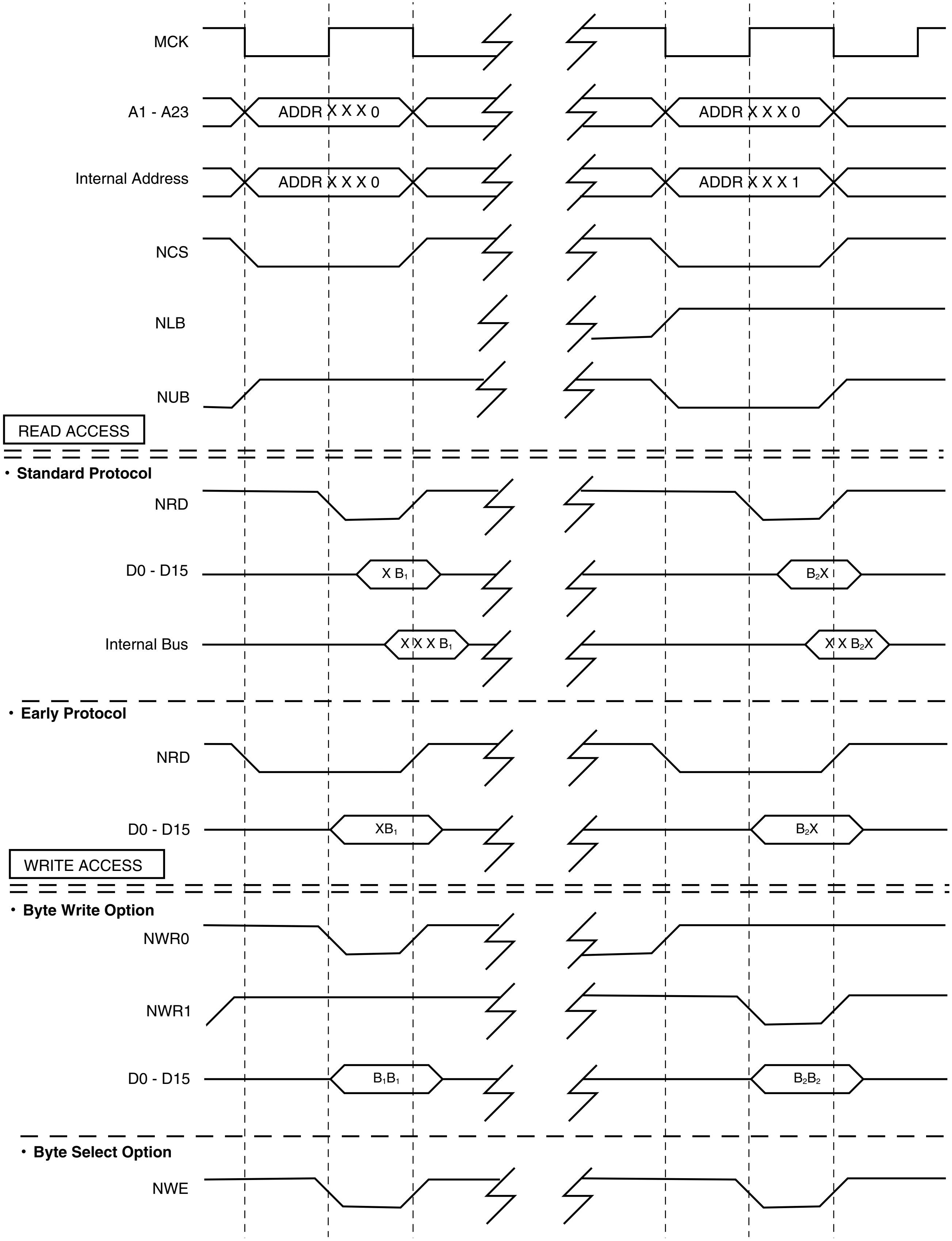


Figure 11-27. 0 Wait States, 16-bit Bus Width, Byte Transfer



11.11 EBI User Interface

The EBI is programmed using the registers listed in the table below. The Remap Control Register (EBI_RCR) controls exit from Boot Mode (See "Boot on NCS0" on page 28.) The Memory Control Register (EBI_MCR) is used to program the number of active chip selects and data read protocol. Eight Chip Select Registers (EBI_CSR0 to EBI_CSR7) are used to program the parameters for the individual external memories. Each EBI_CSR must be programmed with a different base address, even for unused chip selects.

Base Address: 0xFFE00000 (Code Label EBI_BASE)

Table 11-4. EBI Memory Map

Offset	Register	Name	Access	Reset State

0x00	Chip Select Register 0	EBI_CSR0	Read/Write	0x0000203E(1)
				0x0000203D(2)

0x04	Chip Select Register 1	EBI_CSR1	Read/Write	0x10000000
0x08	Chip Select Register 2	EBI_CSR2	Read/Write	0x20000000
0x0C	Chip Select Register 3	EBI_CSR3	Read/Write	0x30000000
0x10	Chip Select Register 4	EBI_CSR4	Read/Write	0x40000000
0x14	Chip Select Register 5	EBI_CSR5	Read/Write	0x50000000
0x18	Chip Select Register 6	EBI_CSR6	Read/Write	0x60000000
0x1C	Chip Select Register 7	EBI_CSR7	Read/Write	0x70000000
0x20	Remap Control Register	EBI_RCR	Write-only	–
0x24	Memory Control Register	EBI_MCR	Read/Write	0

Notes: 1. 8-bit boot (if BMS is detected high)
2. 16-bit boot (if BMS is detected low)

11.11.1 EBI Chip Select Register

Register Name:EBI_CSR0 - EBI_CSR7

Access Type: Read/Write

Reset Value: See Table 11-4 on page 46

Absolute Address:0xFFE00000 - 0xFFE0001C

Offset: 0x00 - 0x1C

31	30	29	28	27	26	25	24
BA							

23	22	21	20	19	18	17	16
BA				–	–	–	–

15	14	13	12	11	10	9	8
–	–	CSEN	BAT	TDF			PAGES

7	6	5	4	3	2	1	0
PAGES	–	WSE	NWS			DBW	

- **DBW: Data Bus Width**

DBW		Data Bus Width	Code Label
			EBI_DBW
0	0	Reserved	–
0	1	16-bit data bus width	EBI_DBW_16
1	0	8-bit data bus width	EBI_DBW_8
1	1	Reserved	–

- **NWS: Number of Wait States**

This field is valid only if WSE is set.

NWS			Number of Standard Wait States	Code Label
				EBI_NWS
0	0	0	1	EBI_NWS_1
0	0	1	2	EBI_NWS_2
0	1	0	3	EBI_NWS_3
0	1	1	4	EBI_NWS_4
1	0	0	5	EBI_NWS_5
1	0	1	6	EBI_NWS_6
1	1	0	7	EBI_NWS_7
1	1	1	8	EBI_NWS_8

- **WSE: Wait State Enable (Code Label `EBI_WSE`)**

0 = Wait state generation is disabled. No wait states are inserted.

1 = Wait state generation is enabled.

- **PAGES: Page Size**

PAGES		Page Size	Active Bits in Base Address	Code Label EBI_PAGES
0	0	1M Byte	12 Bits (31 - 20)	EBI_PAGES_1M
0	1	4M Bytes	10 Bits (31 - 22)	EBI_PAGES_4M
1	0	16M Bytes	8 Bits (31 - 24)	EBI_PAGES_16M
1	1	64M Bytes	6 Bits (31 - 26)	EBI_PAGES_64M

- **TDF: Data Float Output Time**

TDF			Number of Cycles Added after the Transfer	Code Label EBI_TDF
0	0	0	0	EBI_TDF_0
0	0	1	1	EBI_TDF_1
0	1	0	2	EBI_TDF_2
0	1	1	3	EBI_TDF_3
1	0	0	4	EBI_TDF_4
1	0	1	5	EBI_TDF_5
1	1	0	6	EBI_TDF_6
1	1	1	7	EBI_TDF_7

- **BAT: Byte Access Type**

BAT	Selected BAT	Code Label EBI_BAT
0	Byte-write access type.	EBI_BAT_BYTE_WRITE
1	Byte-select access type.	EBI_BAT_BYTE_SELECT

- **CSEN: Chip Select Enable (Code Label `EBI_CSEN`)**

0 = Chip select is disabled.

1 = Chip select is enabled.

- **BA: Base Address (Code Label `EBI_BA`)**

These bits contain the highest bits of the base address. If the page size is larger than 1M byte, the unused bits of the base address are ignored by the EBI decoder.

11.11.2 EBI Remap Control Register

Register Name:EBI_RCR

Access Type: Write-only

Absolute Address:0xFFE00020

Offset: 0x20

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**
–	–	–	–	–	–	–	RCB

- **RCB: Remap Command Bit (Code Label `EBI_RCB`)**

0 = No effect.

1 = Cancels the remapping (performed at reset) of the page zero memory devices.

11.11.3 EBI Memory Control Register

Register Name:EBI_MCR

Access Type: Read/Write

Reset Value: 0

Absolute Address:0xFFE00024

Offset: 0x24

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
–	–	–	DRP	–	ALE		

- **ALE: Address Line Enable**

This field determines the number of valid address lines and the number of valid chip select lines.

ALE			Valid Address Bits	Maximum Addressable Space	Valid Chip Select	Code Label
						EBI_ALE
0	X	X	A20, A21, A22, A23	16M Bytes	None	EBI_ALE_16M
1	0	0	A20, A21, A22	8M Bytes	CS4	EBI_ALE_8M
1	0	1	A20, A21	4M Bytes	CS4, CS5	EBI_ALE_4M
1	1	0	A20	2M Bytes	CS4, CS5, CS6	EBI_ALE_2M
1	1	1	None	1M Byte	CS4, CS5, CS6, CS7	EBI_ALE_1M

- **DRP: Data Read Protocol**

DRP	Selected DRP	Code Label
		EBI_DRP
0	Standard read protocol for all external memory devices enabled	EBI_DRP_STANDARD
1	Early read protocol for all external memory devices enabled	EBI_DRP_EARLY

12. Flash Memory

The device powers on in the read mode. Command sequences are used to place the device in other operation modes such as program and erase. The device has the capability to protect the data in any sector (see "Sector Lockdown" on page 55).

To increase the flexibility of the device, it contains an Erase Suspend and Program Suspend feature. This feature will put the erase or program on hold for any amount of time and let the user read data from or program data to any of the remaining sectors within the memory. The end of a program or an erase cycle is detected by the READY/$\overline{\text{BUSY}}$ pin, $\overline{\text{Data}}$ Polling or by the toggle bit.

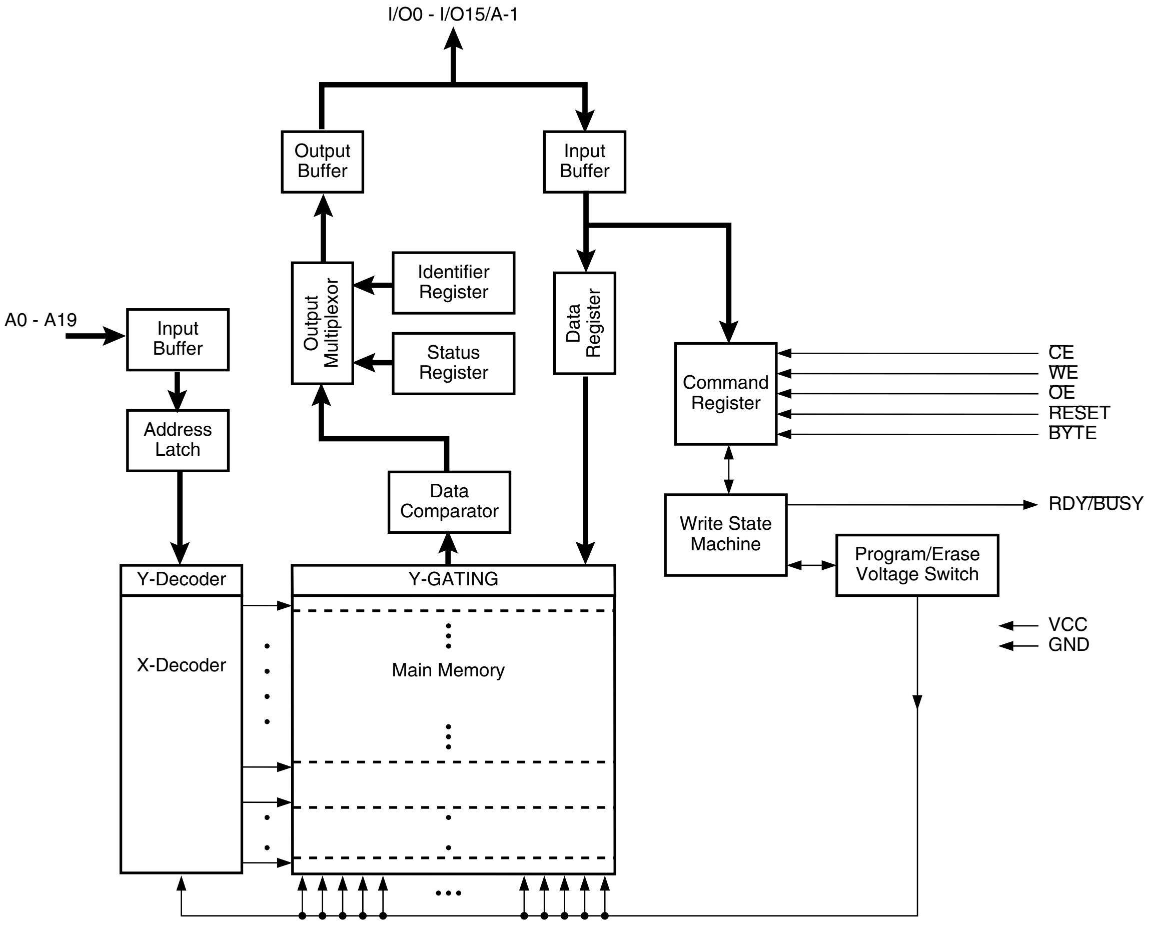
A six-byte command (Enter Single Pulse Program Mode) sequence to remove the requirement of entering the three-byte program sequence is offered to further improve programming time. After entering the six-byte code, only single pulses on the write control lines are required for writing into the device. This mode (Single Pulse Byte/Word Program) is exited by powering down the device, or by pulsing the $\overline{\text{RESET}}$ pin low for a minimum of 500 ns and then bringing it back to V_{CC}. Erase, Erase Suspend/Resume and Program Suspend/Resume commands will not work while in this mode; if entered they will result in data being programmed into the device. It is not recommended that the six-byte code reside in the software of the final product but only exist in external programming code.

The $\overline{\text{BYTE}}$ pin controls whether the device data I/O pins operate in the byte or word configuration. If the $\overline{\text{BYTE}}$ pin is set at logic "1", the device is in word configuration, I/O0 - I/O15 are active and controlled by $\overline{\text{CE}}$ and $\overline{\text{OE}}$.

If the $\overline{\text{BYTE}}$ pin is set at logic "0", the device is in byte configuration, and only data I/O pins I/O0 - I/O7 are active and controlled by $\overline{\text{CE}}$ and $\overline{\text{OE}}$. The data I/O pins I/O8 - I/O14 are tri-stated, and the I/O15 pin is used as an input for the LSB (A-1) address function.

12.1 Block Diagram

Figure 12-1. Flash Memory Block Diagram



12.2 Device Operation

12.2.1 Read

The Flash Memory is accessed like an EPROM. When CE and OE are low and WE is high, the data stored at the memory location determined by the address pins are asserted on the outputs. The outputs are put in the high impedance state whenever CE or OE is high. This dual-line control gives designers flexibility in preventing bus contention.

12.2.2 Command Sequences

When the device is first powered on, it will be reset to the read or standby mode, depending upon the state of the control line inputs. In order to perform other device functions, a series of command sequences are entered into the device. The command sequences are shown in the "Command Definition Table" on page 63 (I/O8 - I/O15 are don't care inputs for the command codes). The command sequences are written by applying a low pulse on the WE or CE input with CE or WE low (respectively) and OE high. The address is latched on the falling edge of CE or WE, whichever occurs last. The data is latched by the first rising edge of CE or WE. Standard microprocessor write timings are used. The address locations used in the command sequences are not affected by entering the command sequences.

12.2.3 Reset

A $\overline{\text{RESET}}$ input pin is provided to ease some system applications. When $\overline{\text{RESET}}$ is at a logic high level, the device is in its standard operating mode. A low level on the $\overline{\text{RESET}}$ input halts the present device operation and puts the outputs of the device in a high impedance state. When a high level is reasserted on the $\overline{\text{RESET}}$ pin, the device returns to the read or standby mode, depending upon the state of the control inputs.

12.2.4 Erasure

Before a byte/word can be reprogrammed, it must be erased. The erased state of memory bits is a logical "1". The entire device can be erased by using the Chip Erase command or individual sectors can be erased by using the Sector Erase command.

12.2.4.1 Chip Erase

The entire device can be erased at one time by using the six-byte chip erase software code. After the chip erase has been initiated, the device will internally time the erase operation so that no external clocks are required. The maximum time to erase the chip is t_{EC}.

If the sector lockdown has been enabled, the chip erase will not erase the data in the sector that has been locked out; it will erase only the unprotected sectors. After the chip erase, the device will return to the read or standby mode.

12.2.4.2 Sector Erase

As an alternative to a full chip erase, the device is organized into 39 sectors (SA0 - SA38) that can be individually erased. The Sector Erase command is a six-bus cycle operation. The sector address is latched on the falling $\overline{\text{WE}}$ edge of the sixth cycle while the 30H data input command is latched on the rising edge of $\overline{\text{WE}}$. The sector erase starts after the rising edge of $\overline{\text{WE}}$ of the sixth cycle. The erase operation is internally controlled; it will automatically time to completion. The maximum time to erase a sector is t_{SEC}. When the sector programming lockdown feature is not enabled, the sector will erase (from the same Sector Erase command). An attempt to erase a sector that has been protected will result in the operation terminating immediately.

12.2.5 Byte/Word Programming

Once a memory block is erased, it is programmed (to a logical "0") on a byte-by-byte or on a word-by-word basis. Programming is accomplished via the internal device command register and is a four-bus cycle operation. The device will automatically generate the required internal program pulses.

Any commands written to the chip during the embedded programming cycle will be ignored. If a hardware reset happens during programming, the data at the location being programmed will be corrupted. Please note that a data "0" cannot be programmed back to a "1"; only erase operations can convert "0"s to "1"s. Programming is completed after the specified t_{BP} cycle time. The $\overline{\text{Data}}$ Polling feature or the Toggle Bit feature may be used to indicate the end of a program cycle. If the erase/program status bit is a "1", the device was not able to verify that the erase or program operation was performed successfully.

12.2.6 Program/Erase Status

The device provides several bits to determine the status of a program or erase operation: I/O2, I/O5, I/O6 and I/O7. The "Status Bit Table" on page 62 and the following four sections describe the function of these bits. To provide greater flexibility for system designers, the Flash Memory contains a programmable configuration register. The configuration register allows the user to specify the status bit operation. The configuration register can be set to one of two different values, “00” or “01”. If the configuration register is set to “00”, the part will automatically return to the read mode after a successful program or erase operation. If the configuration register is set to a “01”, a Product ID Exit command must be given after a successful program or erase operation before the part will return to the read mode. It is important to note that whether the configuration register is set to a “00” or to a “01”, any unsuccessful program or erase operation requires using the Product ID Exit command to return the device to read mode. The default value (after power-up) for the configuration register is “00”. Using the four-bus cycle Set Configuration Register command as shown in Table 12-2, “Command Definition Table,” on page 63, the value of the configuration register can be changed. Voltages applied to the $\overline{\text{RESET}}$ pin will not alter the value of the configuration register. The value of the configuration register will affect the operation of the I/O7 status bit as described below.

12.2.7 $\overline{\text{DATA}}$ Polling

The Flash Memory features $\overline{\text{Data}}$ Polling to indicate the end of a program cycle. If the status configuration register is set to a “00”, during a program cycle an attempted read of the last byte/word loaded will result in the complement of the loaded data on I/O7. Once the program cycle has been completed, true data is valid on all outputs and the next cycle may begin. During a chip or sector erase operation, an attempt to read the device will give a “0” on I/O7. Once the program or erase cycle has completed, true data will be read from the device. $\overline{\text{Data}}$ Polling may begin at any time during the program cycle. Please see Table 12-1 on page 62 for more details.

If the status bit configuration register is set to a “01”, the I/O7 status bit will be low while the device is actively programming or erasing data. I/O7 will go high when the device has completed a program or erase operation. Once I/O7 has gone high, status information on the other pins can be checked.

The $\overline{\text{Data}}$ Polling status bit must be used in conjunction with the erase/program status bit as shown in the algorithm in Figure 12-2 on page 58 and Figure 12-3 on page 59.

12.2.8 Toggle Bit

In addition to $\overline{\text{Data}}$ Polling the Flash Memory provides another method for determining the end of a program or erase cycle. During a program or erase operation, successive attempts to read data from the memory will result in I/O6 toggling between one and zero. Once the program cycle has completed, I/O6 will stop toggling and valid data will be read. Examining the toggle bit may begin at any time during a program cycle. Please see “Status Bit Table” on page 62 for more details.

The toggle bit status bit should be used in conjunction with the erase/program status bit as shown in the algorithm in Figures 12-4 and and 12-5 on page 60.

12.2.9 Erase/Program Status Bit

The device offers a status bit on I/O5, which indicates whether the program or erase operation has exceeded a specified internal pulse count limit. If the status bit is a “1”, the device is unable to verify that an erase or a byte/word program operation has been successfully performed. If a program (Sector Erase) command is issued to a protected sector, the protected sector will not be programmed (erased). The device will go to a status read mode and the I/O5 status bit will be set high, indicating the program (erase) operation did not complete as requested. Once the erase/program status bit has been set to a “1”, the system must write the Product ID Exit command to return to the read mode. The erase/program status bit is a “0” while the erase or program operation is still in progress. Please see “Status Bit Table” on page 62 for more details.

12.3 Sector Lockdown

Each sector has a programming lockdown feature. This feature prevents programming of data in the designated sectors once the feature has been enabled. These sectors can contain secure code that is used to bring up the system. Enabling the lockdown feature will allow the boot code to stay in the device while data in the rest of the device is updated. This feature does not have to be activated; any sector's usage as a write-protected region is optional to the user.

At power-up or reset, all sectors are unlocked. To activate the lockdown for a specific sector, the six-bus cycle Sector Lockdown command must be issued. Once a sector has been locked down, the contents of the sector is read-only and cannot be erased or programmed.

12.3.1 Sector Lockdown Detection

A software method is available to determine if programming of a sector is locked down. When the device is in the software product identification mode (see Section 12.8 ”Software Product Identification Entry” and Section 12.9 “Software Product Identification Exit” on page 66), a read from address location 00002H within a sector will show if programming the sector is locked down. If the data on I/O0 is low, the sector can be programmed; if the data on I/O0 is high, the program lockdown feature has been enabled and the sector cannot be programmed. The software product identification exit code should be used to return to standard operation.

12.3.2 Sector Lockdown Override

The only way to unlock a sector that is locked down is through reset or power-up cycles. After power-up or reset, the content of a sector that is locked down can be erased and reprogrammed.

12.3.3 Erase Suspend/Erase Resume

The Erase Suspend command allows the system to interrupt a sector or chip erase operation and then program or read data from a different sector within the memory. After the Erase Suspend command is given, the device requires a maximum time of 15 µs to suspend the erase operation. After the erase operation has been suspended, the system can then read data or program data to any other sector within the device. An address is not required during the Erase Suspend command. During a sector erase suspend, another sector cannot be erased. To resume the sector erase operation, the system must write the Erase Resume command. The Erase Resume command is a one-bus cycle command. The device also supports an erase suspend during a complete chip erase. While the chip erase is suspended, the user can read from any sector within the memory that is protected. The command sequence for a chip erase suspend and a sector erase suspend are the same.

12.3.4 Program Suspend/Program Resume

The Program Suspend command allows the system to interrupt a programming operation and then read data from a different byte/word within the memory. After the Program Suspend command is given, the device requires a maximum of 10 µs to suspend the programming operation. After the programming operation has been suspended, the system can then read data from any other byte/word that is not contained in the sector in which the programming operation was suspended. An address is not required during the program suspend operation. To resume the programming operation, the system must write the Program Resume command. The program suspend and resume are one-bus cycle commands. The command sequence for the erase suspend and program suspend are the same, and the command sequence for the erase resume and program resume are the same.

12.3.5 Product Identification

The product identification mode identifies the device and manufacturer as Atmel. It is accessed using a software operation.

For details, see “Software Product Identification Entry” and “Software Product Identification Exit” on page 66.

12.3.6 128-bit Protection Register

The Flash Memory contains a 128-bit register that can be used for security purposes in system design. The protection register is divided into two 64-bit blocks. The two blocks are designated as block A and block B. The data in block A is non-changeable and is programmed at the factory with a unique number. The data in block B is programmed by the user and can be locked out such that data in the block cannot be reprogrammed. To program block B in the protection register, the four-bus cycle Program Protection Register command must be used as shown in the Table 12-2, “Command Definition Table,” on page 63. To lock out block B, the four-bus cycle Lock Protection Register command must be used as shown in the “Command Definition Table” . Data bit D1 must be zero during the fourth bus cycle. All other data bits during the fourth bus cycle are don’t cares. To determine whether block B is locked out, the Product ID Entry command is given followed by a read operation from address 80H. If data bit D1 is zero, block B is locked. If data bit D1 is one, block B can be reprogrammed. See Table 12-3 on page 64 for the address locations in the protection register. To read the protection register, the Product ID Entry command is given followed by a normal read operation from an address within the protection register. After determining whether block B is protected or not, or reading the protection register, the Product ID Exit command must be given prior to performing any other operation.

12.3.7 RDY/$\overline{\text{BUSY}}$

An open-drain READY/$\overline{\text{BUSY}}$ output pin provides another method of detecting the end of a program or erase operation. RDY/$\overline{\text{BUSY}}$ is actively pulled low during the internal program and erase cycles and is released at the completion of the cycle. The open-drain connection allows for OR-tying of several devices to the same RDY/$\overline{\text{BUSY}}$ line. See Table 12-1, “Status Bit Table,” on page 62 for more details.

12.3.8 Common Flash Interface (CFI)

CFI is a published, standardized data structure that may be read from a flash device. CFI allows system software to query the installed device to determine the configurations, various electrical and timing parameters, and functions supported by the device. CFI is used to allow the system to learn how to interface to the flash device most optimally. The two primary benefits of using CFI are ease of upgrading and second source availability. The command to enter the CFI Query mode is a one-bus cycle command which requires writing data 98h to address 55h. The CFI Query command can be written when the device is ready to read data or can also be written when the part is in the product ID mode. Once in the CFI Query mode, the system can read CFI data at the addresses given in Table 12-5, “Common Flash Interface Definition,” on page 68. To exit the CFI Query mode, the product ID exit command must be given.

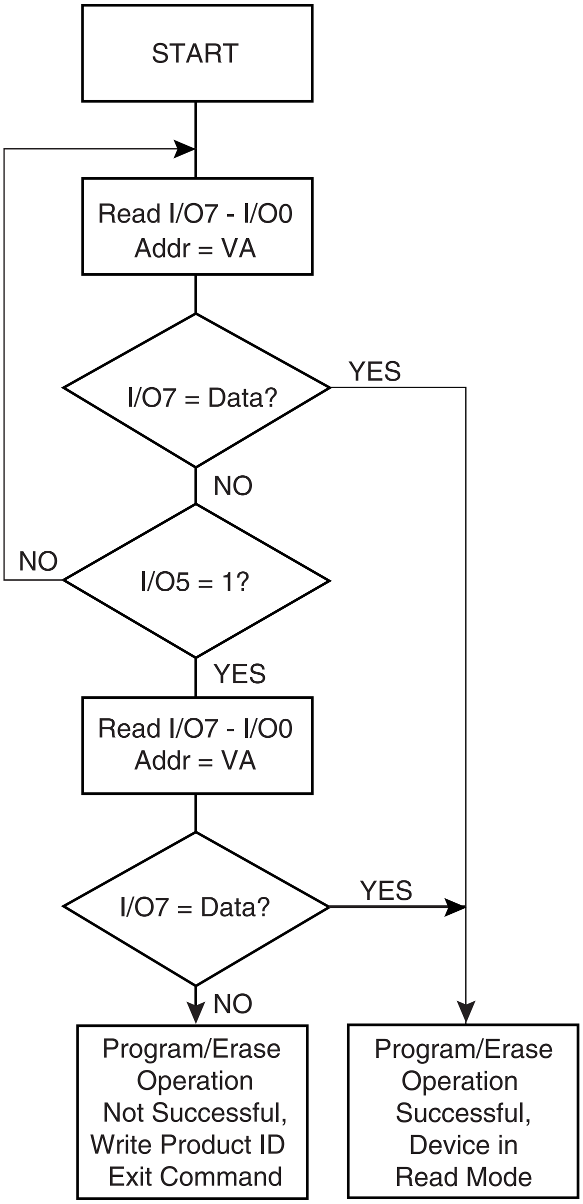
12.3.9 Hardware Data Protection

The Hardware Data Protection feature protects against inadvertent programs to the Flash Memory in the following ways: (a) V_{CC} sense: if V_{CC} is below 1.8V (typical), the program function is inhibited. (b) Program inhibit: holding any one of $\overline{\text{OE}}$ low, $\overline{\text{CE}}$ high or $\overline{\text{WE}}$ high inhibits program cycles.

12.3.10 Input Levels

While operating with a 2.65V to 3.6V power supply, the address inputs and control inputs ($\overline{OE}$, $\overline{CE}$ and $\overline{WE}$) may be driven from 0 to 5.5V without adversely affecting the operation of the device. The I/O lines can only be driven from 0 to V_{CC} + 0.6V.

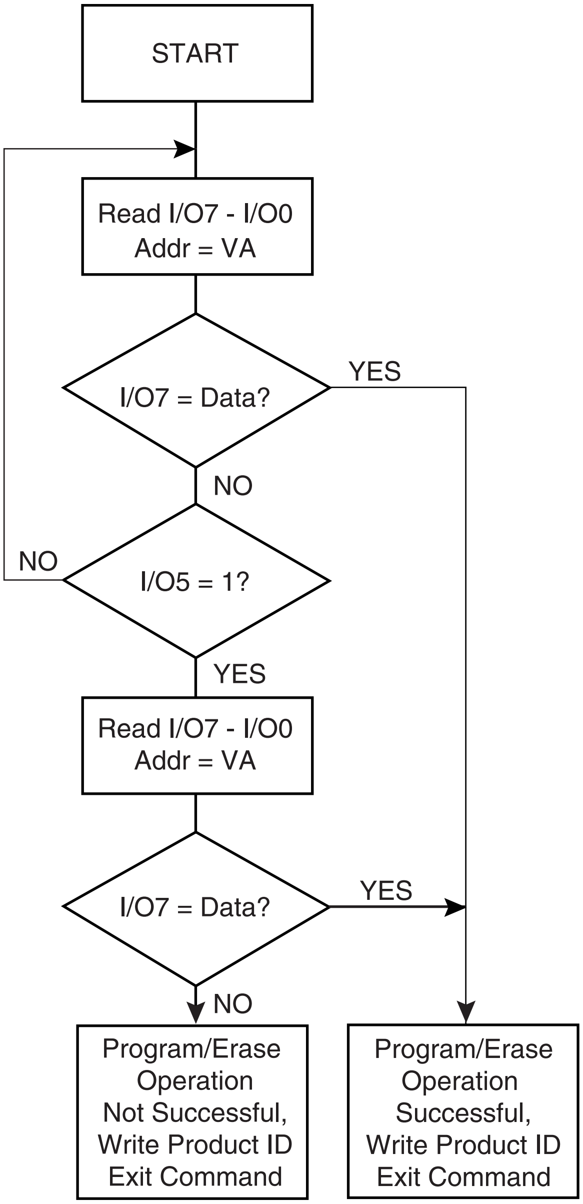
Figure 12-2. $\overline{\text{Data}}$ Polling Algorithm (Configuration Register = 00)


START
Read I/O7 - I/O0
Addr = VA
I/O7 = Data?
YES
NO
NO
I/O5 = 1?
YES
Read I/O7 - I/O0
Addr = VA
I/O7 = Data?
YES
NO
Program/Erase Operation Not Successful, Write Product ID Exit Command
Program/Erase Operation Successful, Device in Read Mode

Notes: 1. VA = Valid address for programming. During a sector erase operation, a valid address is any sector address within the sector being erased. During chip erase, a valid address is any non-protected sector address.

2. I/O7 should be rechecked even if I/O5 = “1” because I/O7 may change simultaneously with I/O5.

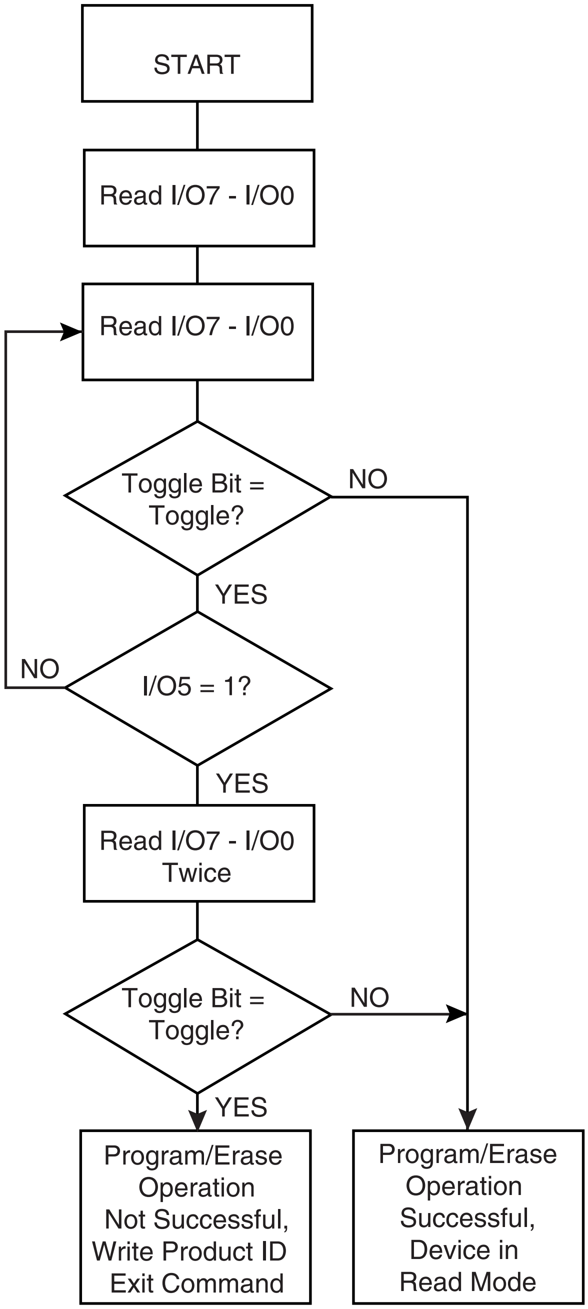
Figure 12-3. $\overline{\text{Data}}$ Polling Algorithm (Configuration Register = 01)



Notes: 1. VA = Valid address for programming. During a sector erase operation, a valid address is any sector address within the sector being erased. During chip erase, a valid address is any non-protected sector address.

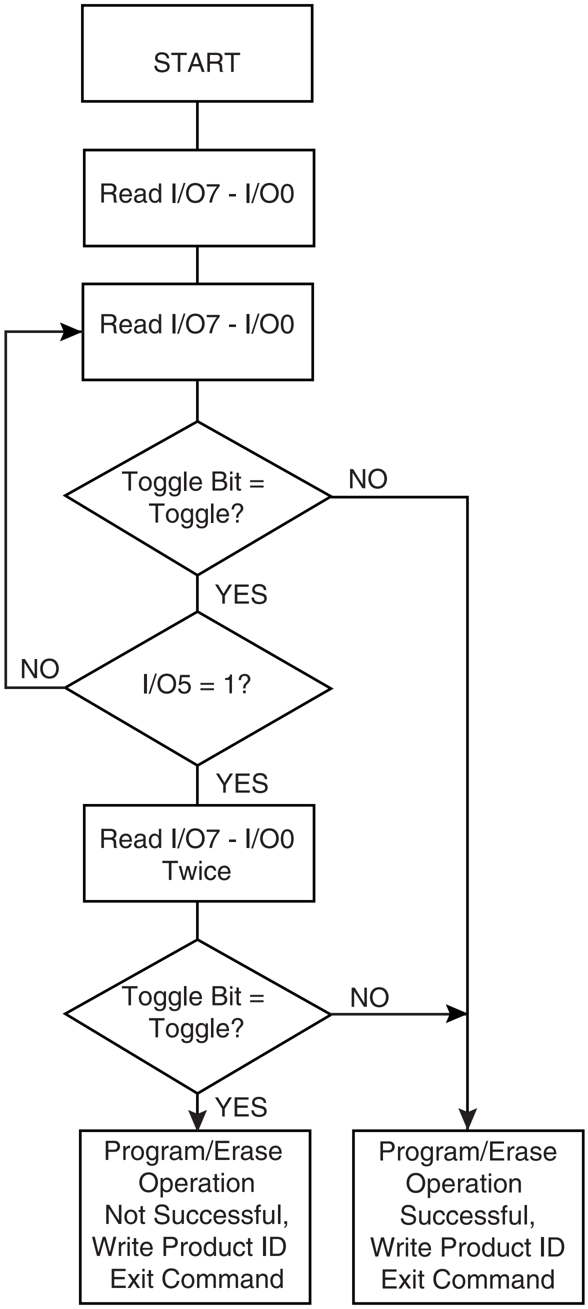
2. I/O7 should be rechecked even if I/O5 = “1” because I/O7 may change simultaneously with I/O5.

Figure 12-4. Toggle Bit Algorithm (Configuration Register = 00)


START
Read I/O7 - I/O0
Read I/O7 - I/O0
Toggle Bit = Toggle?
NO
YES
NO
I/O5 = 1?
YES
Read I/O7 - I/O0 Twice
Toggle Bit = Toggle?
NO
YES
Program/Erase Operation Not Successful, Write Product ID Exit Command
Program/Erase Operation Successful, Device in Read Mode

Note: The system should recheck the toggle bit even if I/O5 = “1” because the toggle bit may stop toggling as I/O5 changes to “1”.

Figure 12-5. Toggle Bit Algorithm (Configuration Register = 01)


START
Read I/O7 - I/O0
Read I/O7 - I/O0
Toggle Bit = Toggle?
NO
YES
I/O5 = 1?
NO
YES
Read I/O7 - I/O0 Twice
Toggle Bit = Toggle?
NO
YES
Program/Erase Operation Not Successful, Write Product ID Exit Command
Program/Erase Operation Successful, Write Product ID Exit Command

Note: The system should recheck the toggle bit even if I/O5 = "1" because the toggle bit may stop toggling as I/O5 changes to "1".

12.4 Status Bit Table

Table 12-1. Status Bit Table

	Status Bit					
	I/O7	**I/O7**	**I/O6**	**I/O5**[1]	**I/O2**	**RDY/$\overline{\text{BUSY}}$**
Configuration Register	**00**	**01**	**00/01**	**00/01**	**00/01**	**00/01**
Programming	$\overline{\text{I/O7}}$	0	TOGGLE	0	1	0
Erasing	0	0	TOGGLE	0	TOGGLE	0
Erase Suspended & Read Erasing Sector	1	1	1	0	TOGGLE	1
Erase Suspended & Read Non-erasing Sector	DATA	DATA	DATA	DATA	DATA	1
Erase Suspended & Program Non-erasing Sector	$\overline{\text{I/O7}}$	0	TOGGLE	0	TOGGLE	0
Erase Suspended & Program Suspended and Reading from Non-suspended Sectors	DATA	DATA	DATA	DATA	DATA	1
Program Suspended & Read Programming Sector	I/O7	1	1	0	TOGGLE	1
Program Suspended & Read Non-programming Sector	DATA	DATA	DATA	DATA	DATA	1

Note: 1. I/O5 switches to a "1" when a program or an erase operation has exceeded the maximum time limits or when a program or sector erase operation is performed on a protected sector.

12.5 Flash Memory Command Definition

Table 12-2. Command Definition Table

Command Sequence	Bus Cycles	1st Bus Cycle		2nd Bus Cycle		3rd Bus Cycle		4th Bus Cycle		5th Bus Cycle		6th Bus Cycle	
		Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
Read	1	Addr	D_{OUT}										
Chip Erase	6	555	AA	AAA[2]	55	555	80	555	AA	AAA	55	555	10
Sector Erase	6	555	AA	AAA	55	555	80	555	AA	AAA	55	SA[3][4]	30
Byte/Word Program	4	555	AA	AAA	55	555	A0	Addr	D_{IN}				
Enter Single Pulse Program Mode	6	555	AA	AAA	55	555	80	555	AA	AAA	55	555	A0
Single Pulse Byte/Word Program	1	Addr	D_{IN}										
Sector Lockdown	6	555	AA	AAA[2]	55	555	80	555	AA	AAA	55	SA[3][4]	60
Erase/Program Suspend	1	XXX	B0										
Erase/Program Resume	1	XXX	30										
Product ID Entry	3	555	AA	AAA	55	555	90						
Product ID Exit [6]	3	555	AA	AAA	55	555	F0[8]						
Product ID Exit [6]	1	XXX	F0[8]										
Program Protection Register	4	555	AA	AAA	55	555	C0	Addr[10]	D_{IN}				
Lock Protection Register - Block B	4	555	AA	AAA	55	555	C0	080	X0				
Status of Block B Protection	4	555	AA	AAA	55	555	90	80	D_{OUT}[6]				
Set Configuration Register	4	555	AA	AAA	55	555	D0	XXX	00/01[7]				
CFI Query[9]	1	X55	98										

Notes:

1. The DATA FORMAT shown for each bus cycle is as follows; I/O7 - I/O0 (Hex). In word operation I/O15 - I/O8 are don't care. The ADDRESS FORMAT shown for each bus cycle is as follows: A11 - A0 (Hex). Address A19 through A11 are don't care in the word mode. Address A19 through A11 and A-1 are don't care in the byte mode.
2. Since A11 is a Don't Care, AAA can be replaced with 2AA.
3. SA = sector address. Any byte/word address within a sector can be used to designate the sector address (see Section 12.7 "Sector Address" on page 65 for details).
4. Once a sector is in the lockdown mode, data in the protected sector cannot be changed unless the chip is reset or power cycled.
5. Either one of the Product ID Exit commands can be used.
6. If data bit D1 is "0", block B is locked. If data bit D1 is "1", block B can be reprogrammed.
7. The default state (after power-up) of the configuration register is "00".
8. Bytes of data other than F0 may be used to exit the Product ID mode. However, it is recommended that F0 be used.
9. When accessing data in the CFI table, the address format is A15- A0 (Hex) in word mode, A14-A0 (Hex) and A-1 = 0 in byte mode.
10. Any address within the user programmable register region. Address locations are shown in Table 12-3 on page 64.

12.6 Protection Register Addressing

Table 12-3. Protection Register Addressing Table [(1)]

Word	Use	Block	A7	A6	A5	A4	A3	A2	A1	A0
0	Factory	A	1	0	0	0	0	0	0	1
1	Factory	A	1	0	0	0	0	0	1	0
2	Factory	A	1	0	0	0	0	0	1	1
3	Factory	A	1	0	0	0	0	1	0	0
4	User	B	1	0	0	0	0	1	0	1
5	User	B	1	0	0	0	0	1	1	0
6	User	B	1	0	0	0	0	1	1	1
7	User	B	1	0	0	0	1	0	0	0

Note: 1. All address lines not specified in the above table must be “0” when accessing the protection register, i.e., A19 - A8 = 0.

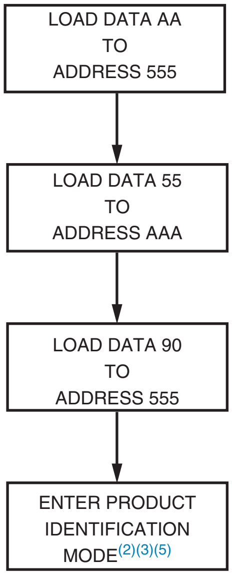
12.7 Sector Address

Table 12-4. Sector Address Table

Sector	Size (Bytes/Words)	x8 Address Range (A19 - A-1)	x16 Address Range (A19 - A0)
SA0	8K/4K	000000 - 001FFF	00000 - 00FFF
SA1	8K/4K	002000 - 003FFF	01000 - 01FFF
SA2	8K/4K	004000 - 005FFF	02000 - 02FFF
SA3	8K/4K	006000 - 007FFF	03000 - 03FFF
SA4	8K/4K	008000 - 009FFF	04000 - 04FFF
SA5	8K/4K	00A000 - 00BFFF	05000 - 05FFF
SA6	8K/4K	00C000 - 00DFFF	06000 - 06FFF
SA7	8K/4K	00E000 - 00FFFF	07000 - 07FFF
SA8	64K/32K	010000 - 01FFFF	08000 - 0FFFF
SA9	64K/32K	020000 - 02FFFF	10000 - 17FFF
SA10	64K/32K	030000 - 03FFFF	18000 - 1FFFF
SA11	64K/32K	040000 - 04FFFF	20000 - 27FFF
SA12	64K/32K	050000 - 05FFFF	28000 - 2FFFF
SA13	64K/32K	060000 - 06FFFF	30000 - 37FFF
SA14	64K/32K	070000 - 07FFFF	38000 - 3FFFF
SA15	64K/32K	080000 - 08FFFF	40000 - 47FFF
SA16	64K/32K	090000 - 09FFFF	48000 - 4FFFF
SA17	64K/32K	0A0000 - 0AFFFF	50000 - 57FFF
SA18	64K/32K	0B0000 - 0BFFFF	58000 - 5FFFF
SA19	64K/32K	0C0000 - 0CFFFF	60000 - 67FFF
SA20	64K/32K	0D0000 - 0DFFFF	68000 - 6FFFF
SA21	64K/32K	0E0000 - 0EFFFF	70000 - 77FFF
SA22	64K/32K	0F0000 - 0FFFFF	78000 - 7FFFF
SA23	64K/32K	100000 - 10FFFF	80000 - 87FFF
SA24	64K/32K	110000 - 11FFFF	88000 - 8FFFF
SA25	64K/32K	120000 - 12FFFF	90000 - 97FFF
SA26	64K/32K	130000 - 13FFFF	98000 - 9FFFF
SA27	64K/32K	140000 - 14FFFF	A0000 - A7FFF
SA28	64K/32K	150000 - 15FFFF	A8000 - AFFFF
SA29	64K/32K	160000 - 16FFFF	B0000 - B7FFF
SA30	64K/32K	170000 - 17FFFF	B8000 - BFFFF
SA31	64K/32K	180000 - 18FFFF	C0000 - C7FFF
SA32	64K/32K	190000 - 19FFFF	C8000 - CFFFF
SA33	64K/32K	1A0000 - 1AFFFF	D0000 - D7FFF
SA34	64K/32K	1B0000 - 1BFFFF	D8000 - DFFFF
SA35	64K/32K	1C0000 - 1CFFFF	E0000 - E7FFF
SA36	64K/32K	1D0000 - 1DFFFF	E8000 - EFFFF
SA37	64K/32K	1E0000 - 1EFFFF	F0000 - F7FFF
SA38	64K/32K	1F0000 - 1FFFFF	F8000 - FFFFF

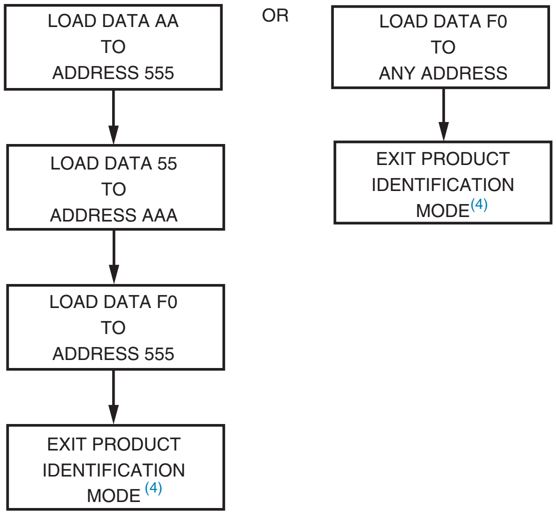
12.8 Software Product Identification Entry

Figure 12-6. Software Product Identification Entry [1]


LOAD DATA AA TO ADDRESS 555
LOAD DATA 55 TO ADDRESS AAA
LOAD DATA 90 TO ADDRESS 555
ENTER PRODUCT IDENTIFICATION MODE[2][3][5]

12.9 Software Product Identification Exit

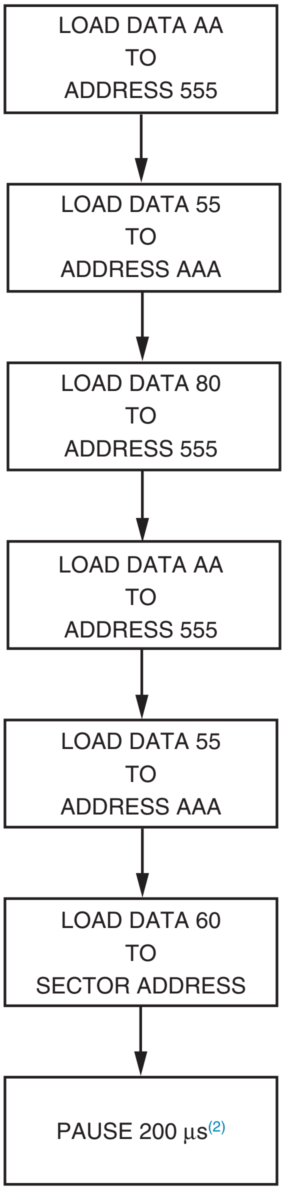
Figure 12-7. Software Product Identification Exit [1] [6]


LOAD DATA AA TO ADDRESS 555
OR
LOAD DATA F0 TO ANY ADDRESS
LOAD DATA 55 TO ADDRESS AAA
EXIT PRODUCT IDENTIFICATION MODE[4]
LOAD DATA F0 TO ADDRESS 555
EXIT PRODUCT IDENTIFICATION MODE [4]

Notes:
1. Data Format: I/O15 - I/O8 (Don't Care); I/O7 - I/O0 (Hex) Address Format: A11 - A0 (Hex), A-1, and A11 - A19 (Don't Care).
2. A1 - A19 = V_{IL}. Manufacturer Code is read for A0 = V_{IL}; Device Code is read for A0 = V_{IH}. Additional Device Code is read from address 0003H.
3. The device does not remain in identification mode if powered down.
4. The device returns to standard operation mode.
5. Manufacturer Code: 001FH
   Device Code: 01C0H
6. Either one of the Product ID Exit commands can be used.

12.10 Sector Lockdown Enable Algorithm

Figure 12-8. Sector Lockdown Enable Algorithm [(1)]


LOAD DATA AA TO ADDRESS 555
LOAD DATA 55 TO ADDRESS AAA
LOAD DATA 80 TO ADDRESS 555
LOAD DATA AA TO ADDRESS 555
LOAD DATA 55 TO ADDRESS AAA
LOAD DATA 60 TO SECTOR ADDRESS
PAUSE 200 µs[(2)]

Notes:
1. Data Format: I/O15 - I/O8 (Don't Care); I/O7 - I/O0 (Hex) Address Format: A11 - A0 (Hex), A-1, and A11 - A19 (Don't Care).
2. Sector Lockdown feature enabled.

12.11 Common Flash Interface Definition

Table 12-5. Common Flash Interface Definition

Address [x16 Mode]	Address [x8 Mode]	Data	Comments
10h	20h	0051h	“Q”
11h	22h	0052h	“R”
12h	24h	0059h	“Y”
13h	26h	0002h	
14h	28h	0000h	
15h	2Ah	0041h	
16h	2Ch	0000h	
17h	2Eh	0000h	
18h	30h	0000h	
19h	32h	0000h	
1Ah	34h	0000h	
1Bh	36h	0027h	V_{CC} min write/erase
1Ch	38h	0036h	V_{CC} max write/erase
1Dh	3Ah	0000h	V_{PP} min voltage
1Eh	3Ch	0000h	V_{PP} max voltage
1Fh	3Eh	0004h	Typ word write – 10 µs
20h	40h	0000h	
21h	42h	0009h	Typ sector erase: 500 ms
22h	44h	000Eh	Typ chip erase: 16,000 ms
23h	46h	0004h	Max word write/typ time
24h	48h	0000h	N/A
25h	4Ah	0004h	Max sector erase/typ sector erase
26h	4Ch	0004h	Max chip erase/typ chip erase
27h	4Eh	0015h	Device size
28h	50h	0002h	x8/x16 device
29h	52h	0000h	x8/x16 device
2Ah	54h	0000h	Multiple byte write not supported
2Bh	56h	0000h	Multiple byte write not supported
2Ch	58h	0002h	2 regions, X = 2
2Dh	5Ah	0007h	8K bytes, Y = 7
2Eh	5Ch	0000h	8K bytes, Y = 7
2Fh	5Eh	0020h	8K bytes, Z = 32
30h	60h	0000h	8K bytes, Z = 32
31h	62h	001Eh	64K bytes, Y = 30
32h	64h	0000h	64K bytes, Y = 30

Table 12-5. Common Flash Interface Definition (Continued)

Address [x16 Mode]	Address [x8 Mode]	Data	Comments
33h	66h	0000h	64K bytes, Z = 256
34h	68h	0001h	64K bytes, Z = 256
Vendor Specific Extended Query			
41h	82h	0050h	“P”
42h	84h	0052h	“R”
43h	86h	0049h	“I”
44h	88h	0031h	Major version number, ASCII
45h	8Ah	0030h	Minor version number, ASCII
46h	8Ch	0087h	Bit 0 – chip erase supported, 0 – no, 1 – yes Bit 1 – erase suspend supported, 0 – no, 1 – yes Bit 2 – program suspend supported, 0 – no, 1 – yes Bit 3 – simultaneous operations supported, 0 – no, 1 – yes Bit 4 – burst mode read supported, 0 – no, 1 – yes Bit 5 – page mode read supported, 0 – no, 1 – yes Bit 6 – queued erase supported, 0 – no, 1 – yes Bit 7 – protection bits supported, 0 – no, 1 – yes
47h	8Eh	0000h (top) or 0001h (bottom)	Bit 8 – top (“0”) or bottom (“1”) boot block device undefined bits are “0”
48h	90h	0000h	Bit 0 – 4-word linear burst with wrap around, 0 – no, 1 – yes Bit 1 – 8-word linear burst with wrap around, 0 – no, 1 – yes Bit 2 – continuos burst, 0 – no, 1 – yes Undefined bits are “0”
49h	92h	0000h	Bit 0 – 4-word page, 0 – no, 1 – yes Bit 1 – 8-word page, 0 – no, 1 – yes Undefined bits are “0”
4Ah	94h	0080h	Location of protection register lock byte, the section’s first byte
4Bh	96h	0003h	# of bytes in the factory prog section of prot register – 2*n
4Ch	98h	0003h	# of bytes in the user prog section of prot register – 2*n

13. PS: Power-saving

The AT91X40 Series' Power-saving feature enables optimization of power consumption. The PS controls the CPU and Peripheral Clocks. One control register (PS_CR) enables the user to stop the ARM7TDMI Clock and enter Idle Mode. One set of registers with a set/clear mechanism enables and disables the peripheral clocks individually.

The ARM7TDMI clock is enabled after a reset and is automatically re-enabled by any enabled interrupt in the Idle Mode.

13.1 Peripheral Clocks

The clock of each peripheral integrated in the AT91FR40162SB can be individually enabled and disabled by writing to the Peripheral Clock Enable (PS_PCER) and Peripheral Clock Disable Registers (PS_PCDR). The status of the peripheral clocks can be read in the Peripheral Clock Status Register (PS_PCSR).

When a peripheral clock is disabled, the clock is immediately stopped. When the clock is re-enabled, the peripheral resumes action where it left off.

To avoid data corruption or erroneous behavior of the system, the system software only disables the clock after all programmed peripheral operations have finished.

The peripheral clocks are automatically enabled after a reset.

The bits that control the peripheral clocks are the same as those that control the Interrupt Sources in the AIC.

13.2 Power Saving (PS) User Interface

Base Address: 0xFFFF4000 (Code Label `PS_BASE`)

Table 13-1. PS Memory Map

Offset	Register	Name	Access	Reset State
0x00	Control Register	PS_CR	Write-only	–
0x04	Peripheral Clock Enable Register	PS_PCER	Write-only	–
0x08	Peripheral Clock Disable Register	PS_PCDR	Write-only	–
0x0C	Peripheral Clock Status Register	PS_PCSR	Read-only	0x17C

13.2.1 PS Control Register

Name: PS_CR

Access: Write-only

Offset: 0x00

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
–	–	–	–	–	–	–	CPU

- **CPU: CPU Clock Disable**

0 = No effect.

1 = Disables the CPU clock.

The CPU clock is re-enabled by any enabled interrupt or by hardware reset.

13.2.2 PS Peripheral Clock Enable Register

Name: PS_PCER

Access: Write-only

Offset: 0x04

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	–	–	–	–	PIO
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	TC2	TC1	TC0	US1	US0	–	–

- **US0: USART 0 Clock Enable**

0 = No effect.

1 = Enables the USART 0 clock.

- **US1: USART 1 Clock Enable**

0 = No effect.

1 = Enables the USART 1 clock.

- **TC0: Timer Counter 0 Clock Enable**

0 = No effect.

1 = Enables the Timer Counter 0 clock.

- **TC1: Timer Counter 1 Clock Enable**

0 = No effect.

1 = Enables the Timer Counter 1 clock.

- **TC2: Timer Counter 2 Clock Enable**

0 = No effect.

1 = Enables the Timer Counter 2 clock.

- **PIO: Parallel IO Clock Enable**

0 = No effect.

1 = Enables the Parallel IO clock.

13.2.3 PS Peripheral Clock Disable Register

Name: PS_PCDR

Access: Write-only

Offset: 0x08

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	PIO

7	6	5	4	3	2	1	0
–	TC2	TC1	TC0	US1	US0	–	–

- **US0: USART 0 Clock Disable**

0 = No effect.

1 = Disables the USART 0 clock.

- **US1: USART 1 Clock Disable**

0 = No effect.

1 = Disables the USART 1 clock.

- **TC0: Timer Counter 0 Clock Disable**

0 = No effect.

1 = Disables the Timer Counter 0 clock.

- **TC1: Timer Counter 1 Clock Disable**

0 = No effect.

1 = Disables the Timer Counter 1 clock.

- **TC2: Timer Counter 2 Clock Disable**

0 = No effect.

1 = Disables the Timer Counter 2 clock.

- **PIO: Parallel IO Clock Disable**

0 = No effect.

1 = Disables the Parallel IO clock.

13.2.4 PS Peripheral Clock Status Register

Name: PS_PCSR

Access: Read-only

Reset Value: 0x17C

Offset: 0x0C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	–	–	–	–	PIO
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	TC2	TC1	TC0	US1	US0	–	–

- **US0: USART 0 Clock Status**

0 = USART 0 clock is disabled.

1 = USART 0 clock is enabled.

- **US1: USART 1 Clock Status**

0 = USART 1 clock is disabled.

1 = USART 1 clock is enabled.

- **TC0: Timer Counter 0 Clock Status**

0 = Timer Counter 0 clock is disabled.

1 = Timer Counter 0 clock is enabled.

- **TC1: Timer Counter 1 Clock Status**

0 = Timer Counter 1 clock is disabled.

1 = Timer Counter 1 clock is enabled.

- **TC2: Timer Counter 2 Clock Status**

0 = Timer Counter 2 clock is disabled.

1 = Timer Counter 2 clock is enabled.

- **PIO: Parallel IO Clock Status**

0 = Parallel IO clock is disabled.

1 = Parallel IO clock is enabled.

14. AIC: Advanced Interrupt Controller

The AT91FR40162SB has an 8-level priority, individually maskable, vectored interrupt controller. This feature substantially reduces the software and real-time overhead in handling internal and external interrupts.

The interrupt controller is connected to the NFIQ (fast interrupt request) and the NIRQ (standard interrupt request) inputs of the ARM7TDMI processor. The processor's NFIQ line can only be asserted by the external fast interrupt request input: FIQ. The NIRQ line can be asserted by the interrupts generated by the on-chip peripherals and the external interrupt request lines: IRQ0 to IRQ2.

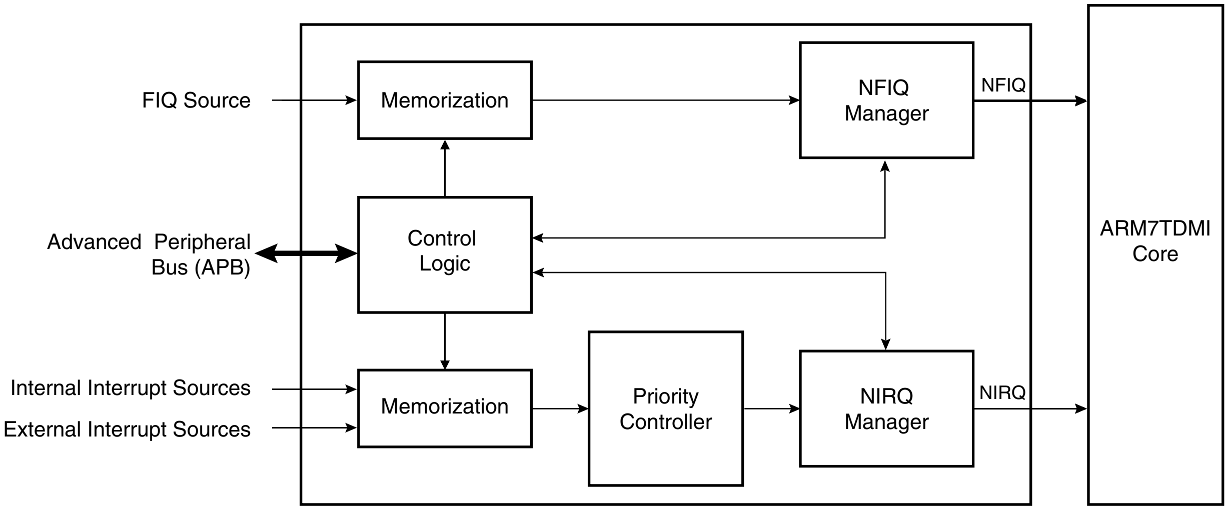
The 8-level priority encoder allows the customer to define the priority between the different NIRQ interrupt sources.

Internal sources are programmed to be level sensitive or edge triggered. External sources can be programmed to be positive or negative edge triggered or high- or low-level sensitive.

The interrupt sources are listed in Table 14-1 on page 77 and the AIC programmable registers in Table 14-3 on page 84.

14.1 Block Diagram

Figure 14-1. Advanced Interrupt Controller Block Diagram



Note: After a hardware reset, the AIC pins are controlled by the PIO Controller. They must be configured to be controlled by the peripheral before being used.

Table 14-1. AIC Interrupt Sources

Interrupt Source [(1)]	Interrupt Name	Interrupt Description
0	FIQ	Fast Interrupt
1	SWIRQ	Software Interrupt
2	US0IRQ	USART Channel 0 interrupt
3	US1IRQ	USART Channel 1 interrupt
4	TC0IRQ	Timer Channel 0 interrupt
5	TC1IRQ	Timer Channel 1 interrupt
6	TC2IRQ	Timer Channel 2 interrupt
7	WDIRQ	Watchdog interrupt
8	PIOIRQ	Parallel I/O Controller interrupt
9	–	Reserved
10	–	Reserved
11	–	Reserved
12	–	Reserved
13	–	Reserved
14	–	Reserved
15	–	Reserved
16	IRQ0	External interrupt 0
17	IRQ1	External interrupt 1
18	IRQ2	External interrupt 2
19	–	Reserved
20	–	Reserved
21	–	Reserved
22	–	Reserved
23	–	Reserved
24	–	Reserved
25	–	Reserved
26	–	Reserved
27	–	Reserved
28	–	Reserved
29	–	Reserved
30	–	Reserved
31	–	Reserved

Note: Reserved interrupt sources are not available. Corresponding registers must not be used and read 0.

14.2 Hardware Interrupt Vectoring

The hardware interrupt vectoring reduces the number of instructions to reach the interrupt handler to only one. By storing the following instruction at address 0x00000018, the processor loads the program counter with the interrupt handler address stored in the AIC_IVR register. Execution is then vectored to the interrupt handler corresponding to the current interrupt.

```
ldr     PC,[PC,# - &F20]
```

The current interrupt is the interrupt with the highest priority when the Interrupt Vector Register (AIC_IVR) is read. The value read in the AIC_IVR corresponds to the address stored in the Source Vector Register (AIC_SVR) of the current interrupt. Each interrupt source has its corresponding AIC_SVR. In order to take advantage of the hardware interrupt vectoring it is necessary to store the address of each interrupt handler in the corresponding AIC_SVR, at system initialization.

14.3 Priority Controller

The NIRQ line is controlled by an 8-level priority encoder. Each source has a programmable priority level of 7 to 0. Level 7 is the highest priority and level 0 the lowest.

When the AIC receives more than one unmasked interrupt at a time, the interrupt with the highest priority is serviced first. If both interrupts have equal priority, the interrupt with the lowest interrupt source number is serviced first (see Table 14-1 on page 77).

The current priority level is defined as the priority level of the current interrupt at the time the register AIC_IVR is read (the interrupt which will be serviced).

In the case when a higher priority unmasked interrupt occurs while an interrupt already exists, there are two possible outcomes depending on whether the AIC_IVR has been read.

- If the NIRQ line has been asserted but the AIC_IVR has not been read, then the processor will read the new higher priority interrupt handler address in the AIC_IVR register and the current interrupt level is updated.
- If the processor has already read the AIC_IVR then the NIRQ line is reasserted. When the processor has authorized nested interrupts to occur and reads the AIC_IVR again, it reads the new, higher priority interrupt handler address. At the same time the current priority value is pushed onto a first-in last-out stack and the current priority is updated to the higher priority.

When the end of interrupt command register (AIC_EOICR) is written the current interrupt level is updated with the last stored interrupt level from the stack (if any). Hence at the end of a higher priority interrupt, the AIC returns to the previous state corresponding to the preceding lower priority interrupt which had been interrupted.

14.4 Interrupt Handling

The interrupt handler must read the AIC_IVR as soon as possible. This de-asserts the NIRQ request to the processor and clears the interrupt in case it is programmed to be edge triggered. This permits the AIC to assert the NIRQ line again when a higher priority unmasked interrupt occurs.

At the end of the interrupt service routine, the end of interrupt command register (AIC_EOICR) must be written. This allows pending interrupts to be serviced.

14.5 Interrupt Masking

Each interrupt source, including FIQ, can be enabled or disabled using the command registers AIC_IECR and AIC_IDCR. The interrupt mask can be read in the read-only register AIC_IMR. A disabled interrupt does not affect the servicing of other interrupts.

14.6 Interrupt Clearing and Setting

All interrupt sources which are programmed to be edge triggered (including FIQ) can be individually set or cleared by respectively writing to the registers AIC_ISCR and AIC_ICCR. This function of the interrupt controller is available for auto-test or software debug purposes.

14.7 Fast Interrupt Request

The external FIQ line is the only source which can raise a fast interrupt request to the processor. Therefore, it has no priority controller.

The external FIQ line can be programmed to be positive or negative edge triggered or high- or low-level sensitive in the AIC_SMR0 register.

The fast interrupt handler address can be stored in the AIC_SVR0 register. The value written into this register is available by reading the AIC_FVR register when an FIQ interrupt is raised. By storing the following instruction at address 0x0000001C, the processor will load the program counter with the interrupt handler address stored in the AIC_FVR register.

```
ldr      PC,[PC,# -&F20]
```

Alternatively the interrupt handler can be stored starting from address 0x0000001C as described in the ARM7TDMI datasheet.

14.8 Software Interrupt

Interrupt source 1 of the advanced interrupt controller is a software interrupt. It must be programmed to be edge triggered in order to set or clear it by writing to the AIC_ISCR and AIC_ICCR.

This is totally independent of the SWI instruction of the ARM7TDMI processor.

14.9 Spurious Interrupt

When the AIC asserts the NIRQ line, the ARM7TDMI enters IRQ Mode and the interrupt handler reads the IVR. It may happen that the AIC de-asserts the NIRQ line after the core has taken into account the NIRQ assertion and before the read of the IVR.

This behavior is called a Spurious Interrupt.

The AIC is able to detect these Spurious Interrupts and returns the Spurious Vector when the IVR is read. The Spurious Vector can be programmed by the user when the vector table is initialized.

A spurious interrupt may occur in the following cases:

- With any sources programmed to be level sensitive, if the interrupt signal of the AIC input is de-asserted at the same time as it is taken into account by the ARM7TDMI.
- If an interrupt is asserted at the same time as the software is disabling the corresponding source through AIC_IDCR (this can happen due to the pipelining of the ARM core).

The same mechanism of spurious interrupt occurs if the ARM7TDMI reads the IVR (application software or ICE) when there is no interrupt pending. This mechanism is also valid for the FIQ interrupts.

Once the AIC enters the spurious interrupt management, it asserts neither the NIRQ nor the NFIQ lines to the ARM7TDMI as long as the spurious interrupt is not acknowledged. Therefore, it is mandatory for the Spurious Interrupt Service Routine to acknowledge the "spurious" behavior by writing to the AIC_EOICR (End of Interrupt) before returning to the interrupted software. It also can perform other operation(s), e.g., trace possible undesirable behavior.

14.10 Protect Mode

The Protect Mode permits reading of the Interrupt Vector Register without performing the associated automatic operations. This is necessary when working with a debug system.

When a Debug Monitor or an ICE reads the AIC User Interface, the IVR could be read. This would have the following consequences in Normal Mode.

- If an enabled interrupt with a higher priority than the current one is pending, it would be stacked
- If there is no enabled pending interrupt, the spurious vector would be returned.

In either case, an End of Interrupt command would be necessary to acknowledge and to restore the context of the AIC. This operation is generally not performed by the debug system. Hence the debug system would become strongly intrusive, and could cause the application to enter an undesired state.

This is avoided by using Protect Mode.

The Protect Mode is enabled by setting the AIC bit in the SF Protect Mode Register (see Section 17. "SF: Special Function Registers" on page 113).

When Protect Mode is enabled, the AIC performs interrupt stacking only when a write access is performed on the AIC_IVR. Therefore, the Interrupt Service Routines must write (arbitrary data) to the AIC_IVR just after reading it.

The new context of the AIC, including the value of the Interrupt Status Register (AIC_ISR), is updated with the current interrupt only when IVR is written.

An AIC_IVR read on its own (e.g. by a debugger), modifies neither the AIC context nor the AIC_ISR.

Extra AIC_IVR reads performed in between the read and the write can cause unpredictable results. Therefore, it is strongly recommended not to set a breakpoint between these two actions, nor to stop the software.

The debug system must not write to the AIC_IVR as this would cause undesirable effects.

The following table shows the main steps of an interrupt and the order in which they are performed according to the mode:

Table 14-2. Order of Interrupt Steps According to Mode

Action	Normal Mode	Protect Mode
Calculate active interrupt (higher than current or spurious)	Read AIC_IVR	Read AIC_IVR
Determine and return the vector of the active interrupt	Read AIC_IVR	Read AIC_IVR
Memorize interrupt	Read AIC_IVR	Read AIC_IVR
Push on internal stack the current priority level	Read AIC_IVR	Write AIC_IVR
Acknowledge the interrupt [1]	Read AIC_IVR	Write AIC_IVR
No effect[2]	Write AIC_IVR	–

Notes: 1. NIRQ de-assertion and automatic interrupt clearing if the source is programmed as level sensitive.

2. Software that has been written and debugged using Protect Mode will run correctly in Normal Mode without modification. However, in Normal Mode the AIC_IVR write has no effect and can be removed to optimize the code.

14.11 Standard Interrupt Sequence

It is assumed that:

- The Advanced Interrupt Controller has been programmed, AIC_SVR are loaded with corresponding interrupt service routine addresses and interrupts are enabled.
- The Instruction at address 0x18(IRQ exception vector address) is

  ldr pc, [pc, # - &F20]

When NIRQ is asserted, if the bit I of CPSR is 0, the sequence is:

1. The CPSR is stored in SPSR_irq, the current value of the Program Counter is loaded in the IRQ link register (r14_irq) and the Program Counter (r15) is loaded with 0x18. In the following cycle during fetch at address 0x1C, the ARM core adjusts r14_irq, decrementing it by 4.
2. The ARM core enters IRQ Mode, if it is not already.
3. When the instruction loaded at address 0x18 is executed, the Program Counter is loaded with the value read in AIC_IVR. Reading the AIC_IVR has the following effects:
   – Set the current interrupt to be the pending one with the highest priority. The current level is the priority level of the current interrupt.
   – De-assert the NIRQ line on the processor. (Even if vectoring is not used, AIC_IVR must be read in order to de-assert NIRQ)
   – Automatically clear the interrupt, if it has been programmed to be edge triggered
   – Push the current level on to the stack
   – Return the value written in the AIC_SVR corresponding to the current interrupt
4. The previous step has effect to branch to the corresponding interrupt service routine. This should start by saving the Link Register(r14_irq) and the SPSR (SPSR_irq). Note that the Link Register must be decremented by 4 when it is saved, if it is to be restored directly into the Program Counter at the end of the interrupt.
5. Further interrupts can then be unmasked by clearing the I bit in the CPSR, allowing re-assertion of the NIRQ to be taken into account by the core. This can occur if an interrupt with a higher priority than the current one occurs.
6. The Interrupt Handler can then proceed as required, saving the registers which will be used and restoring them at the end. During this phase, an interrupt of priority higher than the current level will restart the sequence from step 1. Note that if the interrupt is

programmed to be level sensitive, the source of the interrupt must be cleared during this phase.

7. The I bit in the CPSR must be set in order to mask interrupts before exiting, to ensure that the interrupt is completed in an orderly manner.
8. The End Of Interrupt Command Register (AIC_EOICR) must be written in order to indicate to the AIC that the current interrupt is finished. This causes the current level to be popped from the stack, restoring the previous current level if one exists on the stack. If another interrupt is pending, with lower or equal priority than old current level but with higher priority than the new current level, the NIRQ line is re-asserted, but the interrupt sequence does not immediately start because the I bit is set in the core.
9. The SPSR (SPSR_irq) is restored. Finally, the saved value of the Link Register is restored directly into the PC. This has effect of returning from the interrupt to whatever was being executed before, and of loading the CPSR with the stored SPSR, masking or unmasking the interrupts depending on the state saved in the SPSR (the previous state of the ARM core).

Note: The I bit in the SPSR is significant. If it is set, it indicates that the ARM core was just about to mask IRQ interrupts when the mask instruction was interrupted. Hence, when the SPSR is restored, the mask instruction is completed (IRQ is masked).

14.12 Fast Interrupt Sequence

It is assumed that:

- The Advanced Interrupt Controller has been programmed, AIC_SVR[0] is loaded with fast interrupt service routine address and the fast interrupt is enabled.
- The Instruction at address 0x1C(FIQ exception vector address) is:
- ldr pc, [pc, # - &F20].
- Nested Fast Interrupts are not needed by the user.

When NFIQ is asserted, if the bit F of CPSR is 0, the sequence is:

1. The CPSR is stored in SPSR_fiq, the current value of the Program Counter is loaded in the FIQ link register (r14_fiq) and the Program Counter (r15) is loaded with 0x1C. In the following cycle, during fetch at address 0x20, the ARM core adjusts r14_fiq, decrementing it by 4.
2. The ARM core enters FIQ Mode.
3. When the instruction loaded at address 0x1C is executed, the Program Counter is loaded with the value read in AIC_FVR. Reading the AIC_FVR has effect of automatically clearing the fast interrupt (source 0 connected to the FIQ line), if it has been programmed to be edge triggered. In this case only, it de-asserts the NFIQ line on the processor.
4. The previous step has effect to branch to the corresponding interrupt service routine. It is not necessary to save the Link Register(r14_fiq) and the SPSR (SPSR_fiq) if nested fast interrupts are not needed.
5. The Interrupt Handler can then proceed as required. It is not necessary to save registers r8 to r13 because FIQ Mode has its own dedicated registers and the user r8 to r13 are banked. The other registers, r0 to r7, must be saved before being used, and restored at the end (before the next step). Note that if the fast interrupt is programmed to be level sensitive, the source of the interrupt must be cleared during this phase in order to de-assert the NFIQ line.
6. Finally, the Link Register (r14_fiq) is restored into the PC after decrementing it by 4 (with instruction sub pc, lr, #4 for example). This has effect of returning from the inter-

rupt to whatever was being executed before, and of loading the CPSR with the SPSR, masking or unmasking the fast interrupt depending on the state saved in the SPSR.

The F bit in the SPSR is significant. If it is set, it indicates that the ARM core was just about to mask FIQ interrupts when the mask instruction was interrupted. Hence when the SPSR is restored, the interrupted instruction is completed (FIQ is masked).

14.13 AIC User Interface

Base Address: 0xFFFFF000 (Code Label AIC_BASE)

Table 14-3. AIC Memory Map

Offset	Register	Name	Access	Reset State
0x000	Source Mode Register 0	AIC_SMR0	Read/Write	0
0x004	Source Mode Register 1	AIC_SMR1	Read/Write	0
–	–	–	Read/Write	0
0x07C	Source Mode Register 31	AIC_SMR31	Read/Write	0
0x080	Source Vector Register 0	AIC_SVR0	Read/Write	0
0x084	Source Vector Register 1	AIC_SVR1	Read/Write	0
–	–	–	Read/Write	0
0x0FC	Source Vector Register 31	AIC_SVR31	Read/Write	0
0x100	IRQ Vector Register	AIC_IVR	Read-only	0
0x104	FIQ Vector Register	AIC_FVR	Read-only	0
0x108	Interrupt Status Register	AIC_ISR	Read-only	0
0x10C	Interrupt Pending Register [(1)]	AIC_IPR	Read-only	[(1)]
0x110	Interrupt Mask Register	AIC_IMR	Read-only	0
0x114	Core Interrupt Status Register	AIC_CISR	Read-only	0
0x118	Reserved	–	–	–
0x11C	Reserved	–	–	–
0x120	Interrupt Enable Command Register	AIC_IECR	Write-only	–
0x124	Interrupt Disable Command Register	AIC_IDCR	Write-only	–
0x128	Interrupt Clear Command Register	AIC_ICCR	Write-only	–
0x12C	Interrupt Set Command Register	AIC_ISCR	Write-only	–
0x130	End of Interrupt Command Register	AIC_EOICR	Write-only	–
0x134	Spurious Vector Register	AIC_SPU	Read/Write	0

Note: 1. The reset value of this register depends on the level of the External IRQ lines. All other sources are cleared at reset.

14.13.1 AIC Source Mode Register

Register Name: AIC_SMR0 - AIC_SMR31

Access Type:Read/Write

Reset Value: 0

Offset: 0x000 - 0x07C

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
–	SRCTYPE		–	–	PRIOR		

- **PRIOR: Priority Level (Code Label `AIC_PRIOR`)**

Program the priority level for all sources except source 0 (FIQ).

The priority level can be between 0 (lowest) and 7 (highest).

The priority level is not used for the FIQ, in the SMR0.

- **SRCTYPE: Interrupt Source Type**

Program the input to be positive or negative level sensitive or positive or negative edge triggered.

The active level or edge is not programmable for the internal sources.

SRCTYPE		External Sources	Code Label
			AIC_SRCTYPE
0	0	Low Level Sensitive	AIC_SRCTYPE_EXT_LOW_LEVEL
0	1	Negative Edge Triggered	AIC_SRCTYPE_EXT_NEGATIVE_EDGE
1	0	High Level Sensitive	AIC_SRCTYPE_EXT_HIGH_LEVEL
1	1	Positive Edge Triggered	AIC_SRCTYPE_EXT_POSITIVE_EDGE

SRCTYPE		Internal Sources	Code Label
			AIC_SRCTYPE
x	0	Level Sensitive	AIC_SRCTYPE_INT_LEVEL
x	1	Edge Triggered	AIC_SRCTYPE_INT_EDGE

14.13.2 AIC Source Vector Register

Register Name: AIC_SVR0 - AIC_SVR31

Access Type:Read/Write

Reset Value: 0

Offset: 0x080 - 0x0FC

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

- **VECTOR: Interrupt Handler Address**

The user may store in these registers the addresses of the corresponding handler for each interrupt source.

14.13.3 AIC Interrupt Vector Register

Register Name: AIC_IVR

Access Type:Read-only

Reset Value: 0

Offset: 0x100

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

- **IRQV: Interrupt Vector Register**

The IRQ Vector Register contains the vector programmed by the user in the Source Vector Register corresponding to the current interrupt.

The Source Vector Register (1 to 31) is indexed using the current interrupt number when the Interrupt Vector Register is read.

When there is no current interrupt, the IRQ Vector Register reads 0.

14.13.4 AIC_FIQ Vector Register

Register Name: AIC_FVR

Access Type:Read-only

Reset Value: 0

Offset: 0x104

31	30	29	28	27	26	25	24
FIQV							
23	22	21	20	19	18	17	16
FIQV							
15	14	13	12	11	10	9	8
FIQV							
7	6	5	4	3	2	1	0
FIQV							

- **FIQV: FIQ Vector Register**

The FIQ Vector Register contains the vector programmed by the user in the Source Vector Register 0 which corresponds to FIQ.

14.13.5 AIC Interrupt Status Register

Register Name: AIC_ISR

Access Type:Read-only

Reset Value: 0

Offset: 0x108

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
–	–	–	IRQID				

- **IRQID: Current IRQ Identifier (Code Label `AIC_IRQID`)**

The Interrupt Status Register returns the current interrupt source number.

14.13.6 AIC Interrupt Pending Register

Register Name: AIC_IPR

Access Type:Read-only

Reset Value: 0

Offset: 0x10C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	IRQ2	IRQ1	IRQ0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	PIOIRQ
7	6	5	4	3	2	1	0
WDIRQ	TC2IRQ	TC1IRQ	TC0IRQ	US1IRQ	US0IRQ	SWIRQ	FIQ

- **Interrupt Pending**

0 = Corresponding interrupt is inactive.

1 = Corresponding interrupt is pending.

14.13.7 AIC Interrupt Mask Register

Register Name: AIC_IMR

Access Type:Read-only

Reset Value: 0

Offset: 0x110

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	IRQ2	IRQ1	IRQ0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	PIOIRQ
7	6	5	4	3	2	1	0
WDIRQ	TC2IRQ	TC1IRQ	TC0IRQ	US1IRQ	US0IRQ	SWIRQ	FIQ

- **Interrupt Mask**

0 = Corresponding interrupt is disabled.

1 = Corresponding interrupt is enabled.

14.13.8 AIC Core Interrupt Status Register

Register Name: AIC_CISR

Access Type:Read-only

Reset Value: 0

Offset: 0x114

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
–	–	–	–	–	–	NIRQ	NFIQ

- **NFIQ: NFIQ Status (Code Label `AIC_NFIQ`)**

0 = NFIQ line inactive.

1 = NFIQ line active.

- **NIRQ: NIRQ Status (Code Label `AIC_NIRQ`)**

0 = NIRQ line inactive.

1 = NIRQ line active.

14.13.9 AIC Interrupt Enable Command Register

Register Name: AIC_IECR

Access Type: Write-only

Offset: 0x120

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	IRQ2	IRQ1	IRQ0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	PIOIRQ
7	6	5	4	3	2	1	0
WDIRQ	TC2IRQ	TC1IRQ	TC0IRQ	US1IRQ	US0IRQ	SWIRQ	FIQ

- **Interrupt Enable**

0 = No effect.

1 = Enables corresponding interrupt.

14.13.10 AIC Interrupt Disable Command Register

Register Name: AIC_IDCR

Access Type: Write-only

Offset: 0x124

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	IRQ2	IRQ1	IRQ0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	PIOIRQ
7	6	5	4	3	2	1	0
WDIRQ	TC2IRQ	TC1IRQ	TC0IRQ	US1IRQ	US0IRQ	SWIRQ	FIQ

- **Interrupt Disable**

0 = No effect.

1 = Disables corresponding interrupt.

14.13.11 AIC Interrupt Clear Command Register

Register Name: AIC_ICCR

Access Type: Write-only

Offset: 0x128

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	IRQ2	IRQ1	IRQ0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	PIOIRQ
7	6	5	4	3	2	1	0
WDIRQ	TC2IRQ	TC1IRQ	TC0IRQ	US1IRQ	US0IRQ	SWIRQ	FIQ

- **Interrupt Clear**

0 = No effect.

1 = Clears corresponding interrupt.

14.13.12 AIC Interrupt Set Command Register

Register Name: AIC_ISCR

Access Type: Write-only

Offset: 0x12C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	IRQ2	IRQ1	IRQ0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	PIOIRQ
7	6	5	4	3	2	1	0
WDIRQ	TC2IRQ	TC1IRQ	TC0IRQ	US1IRQ	US0IRQ	SWIRQ	FIQ

- **Interrupt Set**

0 = No effect.

1 = Sets corresponding interrupt.

14.13.13 AIC End of Interrupt Command Register

Register Name: AIC_EOICR

Access Type: Write-only

Offset: 0x130

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
–	–	–	–	–	–	–	–

The End of Interrupt Command Register is used by the interrupt routine to indicate that the interrupt treatment is complete. Any value can be written because it is only necessary to make a write to this register location to signal the end of interrupt treatment.

14.13.14 AIC Spurious Vector Register

Register Name:AIC_SPU

Access Type:Read/Write

Reset Value: 0

Offset: 0x134

31	30	29	28	27	26	25	24
SPUVEC							
23	22	21	20	19	18	17	16
SPUVEC							
15	14	13	12	11	10	9	8
SPUVEC							
7	6	5	4	3	2	1	0
SPUVEC							

- **SPUVEC: Spurious Interrupt Vector Handler Address**

The user may store the address of the spurious interrupt handler in this register.

15. PIO: Parallel I/O Controller

The AT91FR40162SB has 32 programmable I/O lines. Six pins are dedicated as general purpose I/O pins (P16, P17, P18, P19, P23 and P24). Other I/O lines are multiplexed with an external signal of a peripheral to optimize the use of available package pins (see Table 15-1 on page 96). The PIO controller also provides an internal interrupt signal to the Advanced Interrupt Controller.

15.1 Multiplexed I/O Lines

Some I/O lines are multiplexed with an I/O signal of a peripheral. After reset, the pin is generally controlled by the PIO Controller and is in Input Mode. Table 15-1 on page 96 indicates which of these pins are not controlled by the PIO Controller after reset.

When a peripheral signal is not used in an application, the corresponding pin can be used as a parallel I/O. Each parallel I/O line is bi-directional, whether the peripheral defines the signal as input or output. Figure 15-1 on page 95 shows the multiplexing of the peripheral signals with Parallel I/O signals.

If a pin is multiplexed between the PIO Controller and a peripheral, the pin is controlled by the registers PIO_PER (PIO Enable) and PIO_PDR (PIO Disable). The register PIO_PSR (PIO Status) indicates whether the pin is controlled by the corresponding peripheral or by the PIO Controller.

If a pin is a general-purpose parallel I/O pin (not multiplexed with a peripheral), PIO_PER and PIO_PDR have no effect and PIO_PSR returns 1 for the bits corresponding to these pins.

When the PIO is selected, the peripheral input line is connected to zero.

15.2 Output Selection

The user can enable each individual I/O signal as an output with the registers PIO_OER (Output Enable) and PIO_ODR (Output Disable). The output status of the I/O signals can be read in the register PIO_OSR (Output Status). The direction defined has effect only if the pin is configured to be controlled by the PIO Controller.

15.3 I/O Levels

Each pin can be configured to be driven high or low. The level is defined in four different ways, according to the following conditions.

If a pin is controlled by the PIO Controller and is defined as an output (see "Output Selection" above), the level is programmed using the registers PIO_SODR (Set Output Data) and PIO_CODR (Clear Output Data). In this case, the programmed value can be read in PIO_ODSR (Output Data Status).

If a pin is controlled by the PIO Controller and is not defined as an output, the level is determined by the external circuit.

If a pin is not controlled by the PIO Controller, the state of the pin is defined by the peripheral (see peripheral datasheets).

In all cases, the level on the pin can be read in the register PIO_PDSR (Pin Data Status).

15.4 Interrupts

Each parallel I/O can be programmed to generate an interrupt when a level change occurs. This is controlled by the PIO_IER (Interrupt Enable) and PIO_IDR (Interrupt Disable) registers which enable/disable the I/O interrupt by setting/clearing the corresponding bit in the PIO_IMR. When a change in level occurs, the corresponding bit in the PIO_ISR (Interrupt Status) is set whether the pin is used as a PIO or a peripheral and whether it is defined as input or output. If the corresponding interrupt in PIO_IMR (Interrupt Mask) is enabled, the PIO interrupt is asserted.

When PIO_ISR is read, the register is automatically cleared.

15.5 User Interface

Each individual I/O is associated with a bit position in the Parallel I/O user interface registers. Each of these registers are 32 bits wide. If a parallel I/O line is not defined, writing to the corresponding bits has no effect. Undefined bits read zero.

Figure 15-1. Parallel I/O Multiplexed with a Bi-directional Signal

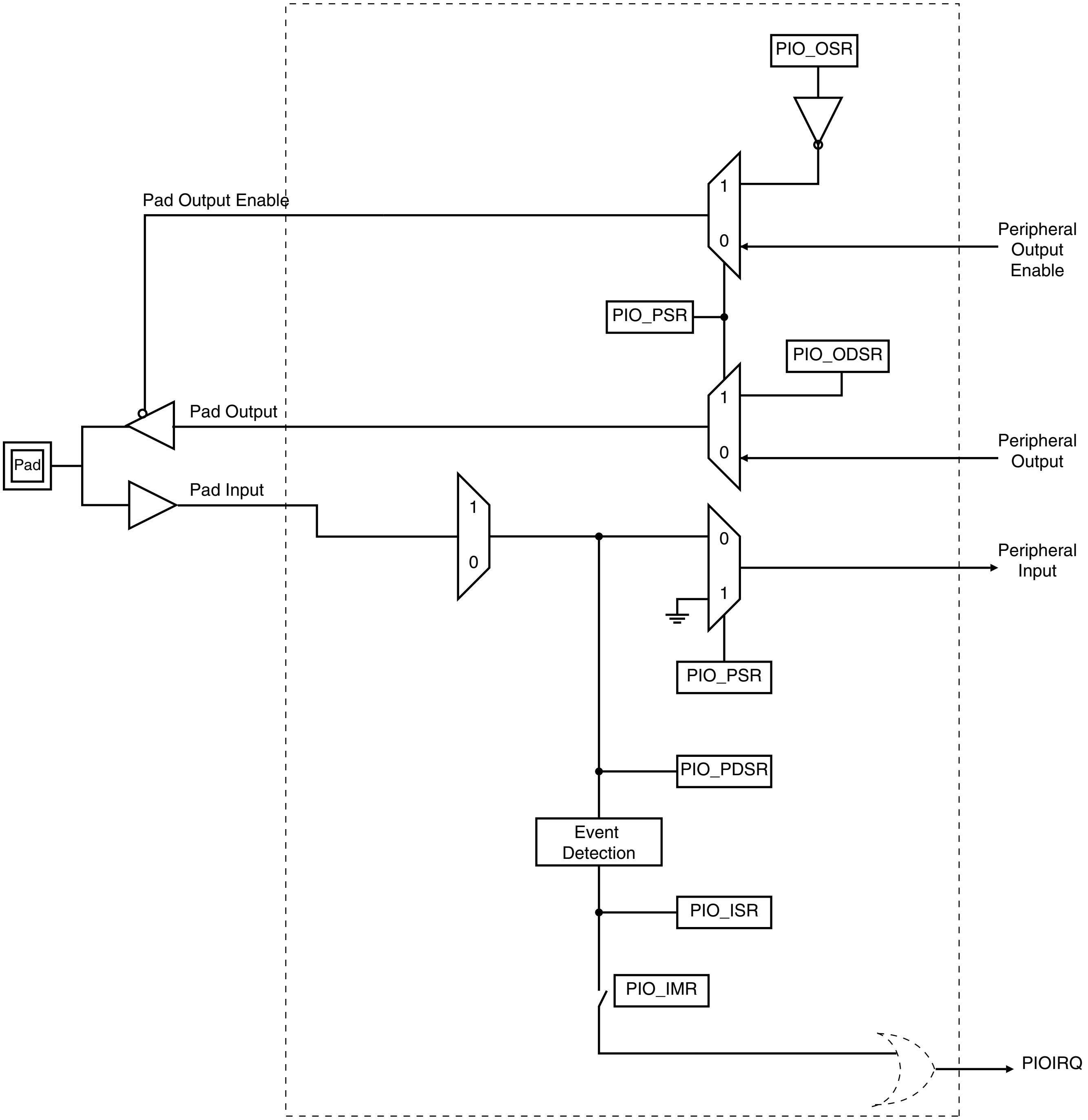


Table 15-1. Multiplexed Parallel I/Os

PIO Controller		Peripheral				
Bit Number[1]	**Port Name**	**Port Name**	**Signal Description**	**Signal Direction**	**Reset State**	**Pin Number**
0	P0	TCLK0	Timer 0 Clock signal	Input	PIO Input	49
1	P1	TIOA0	Timer 0 Signal A	Bi-directional	PIO Input	50
2	P2	TIOB0	Timer 0 Signal B	Bi-directional	PIO Input	51
3	P3	TCLK1	Timer 1 Clock signal	Input	PIO Input	54
4	P4	TIOA1	Timer 1 Signal A	Bi-directional	PIO Input	55
5	P5	TIOB1	Timer 1 Signal B	Bi-directional	PIO Input	56
6	P6	TCLK2	Timer 2 Clock signal	Input	PIO Input	57
7	P7	TIOA2	Timer 2 Signal A	Bi-directional	PIO Input	58
8	P8	TIOB2	Timer 2 Signal B	Bi-directional	PIO Input	59
9	P9	IRQ0	External Interrupt 0	Input	PIO Input	60
10	P10	IRQ1	External Interrupt 1	Input	PIO Input	63
11	P11	IRQ2	External Interrupt 2	Input	PIO Input	64
12	P12	FIQ	Fast Interrupt	Input	PIO Input	66
13	P13	SCK0	USART 0 clock signal	Bi-directional	PIO Input	67
14	P14	TXD0	USART 0 transmit data signal	Output	PIO Input	68
15	P15	RXD0	USART 0 receive data signal	Input	PIO Input	69
16	P16	–	–	–	PIO Input	70
17	P17	–	–	–	PIO Input	71
18	P18	–	–	–	PIO Input	72
19	P19	–	–	–	PIO Input	73
20	P20	SCK1	USART 1 clock signal	Bi-directional	PIO Input	74
21	P21	TXD1	USART 1 transmit data signal	Output	PIO Input	75
22	P22	RXD1	USART 1 receive data signal	Input	PIO Input	76
23	P23	–	–	–	PIO Input	83
24	P24	–	–	–	PIO Input	84
25	P25	MCKO	Master Clock Output	Output	MCKO	85
26	P26	NCS2	Chip Select 2	Output	NCS2	99
27	P27	NCS3	Chip Select 3	Output	NCS3	100
28	P28	A20/CS7	Address 20/Chip Select 7	Output	A20	25
29	P29	A21/CS6	Address 21/Chip Select 6	Output	A21	26
30	P30	A22/CS5	Address 22/Chip Select 5	Output	A22	29
31	P31	A23/CS4	Address 23/Chip Select 4	Output	A23	30

Note: 1. Bit Number refers to the data bit that corresponds to this signal in each of the User Interface registers.

15.6 PIO User Interface

PIO Base Address:0xFFFF0000 (Code Label `PIO_BASE`)

Table 15-2. PIO Controller Memory Map

Offset	Register	Name	Access	Reset State
0x00	PIO Enable Register	PIO_PER	Write-only	–
0x04	PIO Disable Register	PIO_PDR	Write-only	–
0x08	PIO Status Register	PIO_PSR	Read-only	0x01FFFFFF (see Table 15-1)
0x0C	Reserved	–	–	–
0x10	Output Enable Register	PIO_OER	Write-only	–
0x14	Output Disable Register	PIO_ODR	Write-only	–
0x18	Output Status Register	PIO_OSR	Read-only	0
0x1C	Reserved	–	–	–
0x20	Input Filter Enable Register	PIO_IFER	Write-only	–
0x24	Input Filter Disable Register	PIO_IFDR	Write-only	–
0x28	Input Filter Status Register	PIO_IFSR	Read-only	0
0x2C	Reserved	–	–	–
0x30	Set Output Data Register	PIO_SODR	Write-only	–
0x34	Clear Output Data Register	PIO_CODR	Write-only	–
0x38	Output Data Status Register	PIO_ODSR	Read-only	0
0x3C	Pin Data Status Register [(1)]	PIO_PDSR	Read-only	[(1)]
0x40	Interrupt Enable Register	PIO_IER	Write-only	–
0x44	Interrupt Disable Register	PIO_IDR	Write-only	–
0x48	Interrupt Mask Register	PIO_IMR	Read-only	0
0x4C	Interrupt Status Register [(2)]	PIO_ISR	Read-only	[(2)]

Notes: 1. The reset value of this register depends on the level of the external pins at reset.

2. This register is cleared at reset. However, the first read of the register can give a value not equal to zero if any changes have occurred on any pins between the reset and the read.

15.6.1 PIO Enable Register

Register Name:PIO_PER

Access Type:Write-only

Offset: 0x00

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24

23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16

15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register is used to enable individual pins to be controlled by the PIO Controller instead of the associated peripheral. When the PIO is enabled, the associated peripheral input (if any) is held at logic zero.

1 = Enables the PIO to control the corresponding pin (disables peripheral control of the pin).

0 = No effect.

15.6.2 PIO Disable Register

Register Name: PIO_PDR

Access Type:Write-only

Offset: 0x04

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24

23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16

15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register is used to disable PIO control of individual pins. When the PIO control is disabled, the normal peripheral function is enabled on the corresponding pin.

1 = Disables PIO control (enables peripheral control) on the corresponding pin.

0 = No effect.

15.6.3 PIO Status Register

Register Name:PIO_PSR

Access Type:Read-only

Reset Value: 0x01FFFFFF

Offset: 0x08

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register indicates which pins are enabled for PIO control. This register is updated when PIO lines are enabled or disabled.

1 = PIO is active on the corresponding line (peripheral is inactive).

0 = PIO is inactive on the corresponding line (peripheral is active).

15.6.4 PIO Output Enable Register

Register Name:PIO_OER

Access Type:Write-only

Offset: 0x10

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register is used to enable PIO output drivers. If the pin is driven by a peripheral, this has no effect on the pin, but the information is stored. The register is programmed as follows:

1 = Enables the PIO output on the corresponding pin.

0 = No effect.

15.6.5 PIO Output Disable Register

Register Name:PIO_ODR

Access Type:Write-only

Offset: 0x14

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24

23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16

15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register is used to disable PIO output drivers. If the pin is driven by the peripheral, this has no effect on the pin, but the information is stored. The register is programmed as follows:

1 = Disables the PIO output on the corresponding pin.

0 = No effect.

15.6.6 PIO Output Status Register

Register Name:PIO_OSR

Access Type:Read-only

Reset Value: 0

Offset: 0x18

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24

23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16

15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register shows the PIO pin control (output enable) status which is programmed in PIO_OER and PIO ODR. The defined value is effective only if the pin is controlled by the PIO. The register reads as follows:

1 = The corresponding PIO is output on this line.

0 = The corresponding PIO is input on this line.

15.6.7 PIO Input Filter Enable Register

Register Name:PIO_IFER

Access Type:Write-only

Offset: 0x20

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24

23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16

15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register is used to enable input glitch filters. It affects the pin whether or not the PIO is enabled. The register is programmed as follows:

1 = Enables the glitch filter on the corresponding pin.

0 = No effect.

15.6.8 PIO Input Filter Disable Register

Register Name:PIO_IFDR

Access Type:Write-only

Offset: 0x24

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24

23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16

15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register is used to disable input glitch filters. It affects the pin whether or not the PIO is enabled. The register is programmed as follows:

1 = Disables the glitch filter on the corresponding pin.

0 = No effect.

15.6.9 PIO Input Filter Status Register

Register Name:PIO_IFSR

Access Type:Read-only

Reset Value :0

Offset: 0x28

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24

23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16

15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register indicates which pins have glitch filters selected. It is updated when PIO outputs are enabled or disabled by writing to PIO_IFER or PIO_IFDR.

1 = Filter is selected on the corresponding input (peripheral and PIO).

0 = Filter is not selected on the corresponding input.

15.6.10 PIO Set Output Data Register

Register Name:PIO_SODR

Access Type:Write-only

Offset: 0x30

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24

23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16

15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register is used to set PIO output data. It affects the pin only if the corresponding PIO output line is enabled and if the pin is controlled by the PIO. Otherwise, the information is stored.

1 = PIO output data on the corresponding pin is set.

0 = No effect.

15.6.11 PIO Clear Output Data Register

Register Name:PIO_CODR

Access Type:Write-only

Offset: 0x34

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register is used to clear PIO output data. It affects the pin only if the corresponding PIO output line is enabled and if the pin is controlled by the PIO. Otherwise, the information is stored.

1 = PIO output data on the corresponding pin is cleared.

0 = No effect.

15.6.12 PIO Output Data Status Register

Register Name:PIO_ODSR

Access Type:Read-only

Reset Value: 0

Offset: 0x38

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register shows the output data status which is programmed in PIO_SODR or PIO_CODR. The defined value is effective only if the pin is controlled by the PIO Controller and only if the pin is defined as an output.

1 = The output data for the corresponding line is programmed to 1.

0 = The output data for the corresponding line is programmed to 0.

15.6.13 PIO Pin Data Status Register

Register Name:PIO_PDSR

Access Type:Read-only

Reset Value: see Table 15-2

Offset: 0x3C

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24

23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16

15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register shows the state of the physical pin of the chip. The pin values are always valid regardless of whether the pins are enabled as PIO, peripheral, input or output. The register reads as follows:

1 = The corresponding pin is at logic 1.

0 = The corresponding pin is at logic 0.

15.6.14 PIO Interrupt Enable Register

Register Name:PIO_IER

Access Type:Write-only

Offset: 0x40

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24

23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16

15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register is used to enable PIO interrupts on the corresponding pin. It has effect whether PIO is enabled or not.

1 = Enables an interrupt when a change of logic level is detected on the corresponding pin.

0 = No effect.

15.6.15 PIO Interrupt Disable Register

Register Name:PIO_IDR

Access Type:Write-only

Offset: 0x44

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register is used to disable PIO interrupts on the corresponding pin. It has effect whether the PIO is enabled or not.

1 = Disables the interrupt on the corresponding pin. Logic level changes are still detected.

0 = No effect.

15.6.16 PIO Interrupt Mask Register

Register Name:PIO_IMR

Access Type:Read-only

Reset Value: 0

Offset: 0x48

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register shows which pins have interrupts enabled. It is updated when interrupts are enabled or disabled by writing to PIO_IER or PIO_IDR.

1 = Interrupt is enabled on the corresponding input pin.

0 = Interrupt is not enabled on the corresponding input pin.

15.6.17 PIO Interrupt Status Register

Register Name:PIO_ISR

Access Type:Read-only

Reset Value: 0

Offset: 0x4C

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24

23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16

15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register indicates for each pin when a logic value change has been detected (rising or falling edge). This is valid whether the PIO is selected for the pin or not and whether the pin is an input or output.

The register is reset to zero following a read, and at reset.

1 = At least one change has been detected on the corresponding pin since the register was last read.

0 = No change has been detected on the corresponding pin since the register was last read.

16. WD: Watchdog Timer

The AT91FR40162SB has an internal watchdog timer which can be used to prevent system lock-up if the software becomes trapped in a deadlock. In normal operation the user reloads the watchdog at regular intervals before the timer overflow occurs. If an overflow does occur, the watchdog timer generates one or a combination of the following signals, depending on the parameters in WD_OMR (Overflow Mode Register):

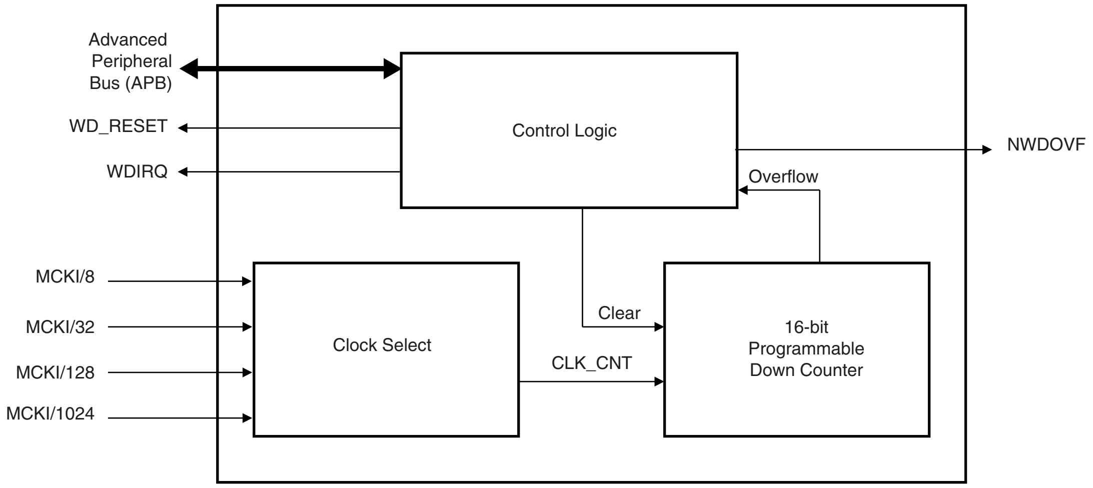
- If RSTEN is set, an internal reset is generated (WD_RESET as shown in Figure 16-1).
- If IRQEN is set, a pulse is generated on the signal WDIRQ which is connected to the Advanced Interrupt Controller
- If EXTEN is set, a low level is driven on the NWDOVF signal for a duration of 8 MCK cycles.

The watchdog timer has a 16-bit down counter. Bits 12-15 of the value loaded when the watchdog is restarted are programmable using the HPVC parameter in WD_CMR (Clock Mode). Four clock sources are available to the watchdog counter: MCK/8, MCK/32, MCK/128 or MCK/1024. The selection is made using the WDCLKS parameter in WD_CMR. This provides a programmable time-out period of 1 ms to 2 sec. with a 33 MHz system clock.

All write accesses are protected by control access keys to help prevent corruption of the watchdog should an error condition occur. To update the contents of the mode and control registers it is necessary to write the correct bit pattern to the control access key bits at the same time as the control bits are written (the same write access).

16.1 Block Diagram

Figure 16-1. Watchdog Timer Block Diagram



16.2 WD Enabling Sequence

To enable the Watchdog Timer the sequence is as follows:

1. Disable the Watchdog by clearing the bit WDEN:
   Write 0x2340 to WD_OMR

   This step is unnecessary if the WD is already disabled (reset state).

2. Initialize the WD Clock Mode Register:
   Write 0x373C to WD_CMR
   (HPCV = 15 and WDCLKS = MCK/8)

3. Restart the timer:
   Write 0xC071 to WD_CR

4. Enable the watchdog:
   Write 0x2345 to WD_OMR (interrupt enabled)

16.3 WD User Interface

WD Base Address: 0xFFFF8000 (Code Label `WD_BASE`)

Table 16-1. WD Memory Map

Offset	Register	Name	Access	Reset State
0x00	Overflow Mode Register	WD_OMR	Read/Write	0
0x04	Clock Mode Register	WD_CMR	Read/Write	0
0x08	Control Register	WD_CR	Write-only	–
0x0C	Status Register	WD_SR	Read-only	0

16.3.1 WD Overflow Mode Register

Name: WD_OMR

Access: Read/Write

Reset Value: 0

Offset: 0x00

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
OKEY							
7	6	5	4	3	2	1	0
OKEY				EXTEN	IRQEN	RSTEN	WDEN

- **WDEN: Watch Dog Enable (Code Label `WD_WDEN`)**

0 = Watch Dog is disabled and does not generate any signals.

1 = Watch Dog is enabled and generates enabled signals.

- **RSTEN: Reset Enable (Code Label `WD_RSTEN`)**

0 = Generation of an internal reset by the Watch Dog is disabled.

1 = When overflow occurs, the Watch Dog generates an internal reset.

- **IRQEN: Interrupt Enable (Code Label `WD_IRQEN`)**

0 = Generation of an interrupt by the Watch Dog is disabled.

1 = When overflow occurs, the Watch Dog generates an interrupt.

- **EXTEN: External Signal Enable (Code Label `WD_EXTEN`)**

0 = Generation of a pulse on the pin NWDOVF by the Watch Dog is disabled.

1 = When an overflow occurs, a pulse on the pin NWDOVF is generated.

- **OKEY: Overflow Access Key (Code Label `WD_OKEY`)**

Used only when writing WD_OMR. OKEY is read as 0.

0x234 = Write access in WD_OMR is allowed.

Other value = Write access in WD_OMR is prohibited.

16.3.2 WD Clock Mode Register

Name: WD_CMR

Access: Read/Write

Reset Value: 0

Offset: 0x04

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
CKEY							

7	6	5	4	3	2	1	0
CKEY	–	HPCV				WDCLKS	

- **WDCLKS: Clock Selection**

WDCLKS		Clock Selected	Code Label
			WD_WDCLKS
0	0	MCK/8	WD_WDCLKS_MCK8
0	1	MCK/32	WD_WDCLKS_MCK32
1	0	MCK/128	WD_WDCLKS_MCK128
1	1	MCK/1024	WD_WDCLKS_MCK1024

- **HPCV: High Preload Counter Value (Code Label `WD_HPCV`)**

Counter is preloaded when watchdog counter is restarted with bits 0 to 11 set (FFF) and bits 12 to 15 equaling HPCV.

- **CKEY: Clock Access Key (Code Label `WD_CKEY`)**

Used only when writing WD_CMR. CKEY is read as 0.

0x06E: Write access in WD_CMR is allowed.

Other value: Write access in WD_CMR is prohibited.

16.3.3 WD Control Register

Name: WD_CR

Access: Write-only

Offset: 0x08

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
RSTKEY							
7	6	5	4	3	2	1	0
RSTKEY							

- **RSTKEY: Restart Key (Code Label `WD_RSTKEY`)**

0xC071 = Watch Dog counter is restarted.

Other value = No effect.

16.3.4 WD Status Register

Name: WD_SR

Access: Read-only

Reset Value: 0

Offset: 0x0C

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
–	–	–	–	–	–	–	WDOVF

- **WDOVF: Watchdog Overflow (Code Label `WD_WDOVF`)**

0 = No watchdog overflow.

1 = A watchdog overflow has occurred since the last restart of the watchdog counter or since internal or external reset.

17. SF: Special Function Registers

The AT91FR40162SB provides registers to implement the following special functions.

- Chip identification
- RESET status
- Protect Mode (see Section 14.10 “Protect Mode” on page 80)

17.1 Chip Identification

Table 17-1 provides the Chip ID values for the products as listed.

Table 17-1. Chip ID Values

Product	Chip
AT91M40800	0x14080044
AT91M40800A	0x14080045
AT91R40807	0x44080746
AT91M40807	0x14080745
AT91R40008	0x44000840

17.2 SF User Interface

Chip ID Base Address = 0xFFF00000 (Code Label `SF_BASE`)

Table 17-2. SF Memory Map

Offset	Register	Name	Access	Reset State
0x00	Chip ID Register	SF_CIDR	Read-only	Hardwired
0x04	Chip ID Extension Register	SF_EXID	Read-only	Hardwired
0x08	Reset Status Register	SF_RSR	Read-only	See register description
0x10	Reserved	–	–	–
0x14	Reserved	–	–	–
0x18	Protect Mode Register	SF_PMR	Read/Write	0x0

17.2.1 Chip ID Register

Register Name:SF_CIDR

Access Type:Read-only

Reset Value: Hardwired

Offset: 0x00

31	30	29	28	27	26	25	24
EXT	NVPTYP			ARCH			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ARCH				VDSIZ			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
NVDSIZ				NVPSIZ			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	1	0	VERSION				

- **VERSION: Version of the chip (Code Label `SF_VERSION`)**

This value is incremented by one with each new version of the chip (from zero to a maximum value of 31).

- **NVPSIZ: Non Volatile Program Memory Size**

					Code Label
NVPSIZ				**Size**	**SF_NVPSIZ**
0	0	0	0	None	SF_NVPSIZ_NONE
0	0	1	1	32K bytes	SF_NVPSIZ_32K
0	1	0	1	64K bytes	SF_NVPSIZ_64K
0	1	1	1	128K bytes	SF_NVPSIZ_128K
1	0	0	0	256K bytes	SF_NVPSIZ_256K
Others				Reserved	–

- **NVDSIZ: Non Volatile Data Memory Size**

					Code Label
NVDSIZ				**Size**	**SF_NVDSIZ**
0	0	0	0	None	SF_NVDSIZ_NONE
Others				Reserved	–

- **VDSIZ: Volatile Data Memory Size**

VDSIZ				Size	Code Label SF_VDSIZ
0	0	0	0	None	SF_VDSIZ_NONE
0	0	0	1	1K bytes	SF_VDSIZ_1K
0	0	1	0	2K bytes	SF_VDSIZ_2K
0	1	0	0	4K bytes	SF_VDSIZ_4K
1	0	0	0	8K bytes	SF_VDSIZ_8K
Others				Reserved	–

- **ARCH: Chip Architecture (Code Label `SF_ARCH`)**

Code of Architecture: Two BCD digits.

		Code Label
0100 0000	AT91x40yyy	SF_ARCH_AT91x40

- **NVPTYP: Non Volatile Program Memory Type**

NVPTYP			Type	Code Label SF_NVPTYP
0	0	0	Reserved	–
0	0	1	“F” Series	SF_NVPTYP_M
1	x	x	Reserved	–
1	0	0	“R” Series	SF_NVPTYP_R

- **EXT: Extension Flag (Code Label `SF_EXT`)**

0 = Chip ID has a single register definition without extensions

1 = An extended Chip ID exists (to be defined in the future).

17.2.2 Chip ID Extension Register

Register Name:SF_EXID

Access Type:Read-only

Reset Value: Hardwired

Offset: 0x04

This register is reserved for future use. It will be defined when needed.

17.2.3 Reset Status Register

Register Name:SF_RSR

Access Type:Read-only

Reset Value: See Below

Offset: 0x08

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**
RESET							

- **RESET: Reset Status Information**

This field indicates whether the reset was demanded by the external system (via NRST) or by the Watchdog internal reset request.

		Code Label
Reset	**Cause of Reset**	**SF_RESET**
0x6C	External Pin	SF_EXT_RESET
0x53	Internal Watchdog	SF_WD_RESET

17.2.4 SF Protect Mode Register

Register Name:SF_PMR

Access Type:Read/Write

Reset Value: 0

Offset: 0x18

31	30	29	28	27	26	25	24
PMRKEY							
23	22	21	20	19	18	17	16
PMRKEY							
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	AIC	–	–	–	–	–

- **PMRKEY: Protect Mode Register Key (Code Label `SF_PMRKEY`)**

Used only when writing SF_PMR. PMRKEY is reads 0.

0x27A8: Write access in SF_PMR is allowed.

Other value: Write access in SF_PMR is prohibited.

- **AIC: AIC Protect Mode Enable (Code Label `SF_AIC`)**

0 = The Advanced Interrupt Controller runs in Normal Mode.

1 = The Advanced Interrupt Controller runs in Protect Mode.

See Section 14.10 “Protect Mode” on page 80.

18. USART: Universal Synchronous Asynchronous Receiver Transmitter

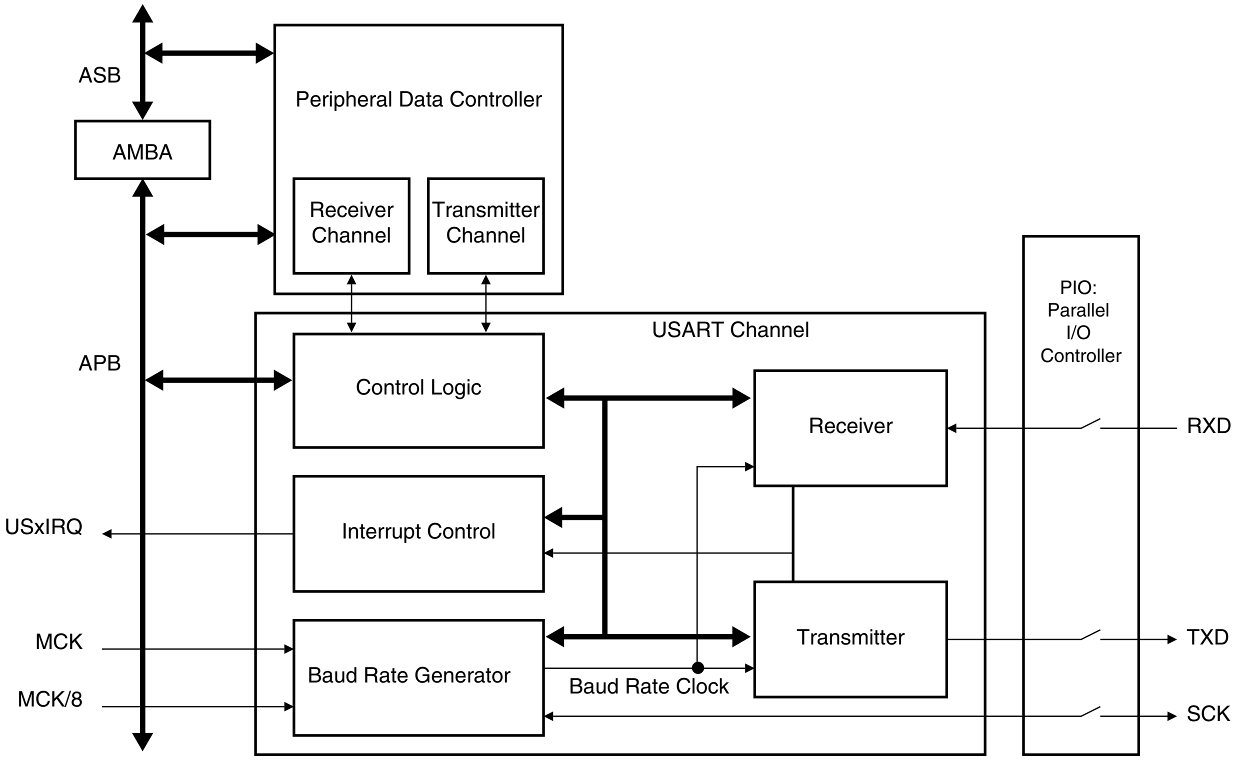
The AT91FR40162SB provides two identical, full-duplex, universal synchronous/asynchronous receiver/transmitters that interface to the APB and are connected to the Peripheral Data Controller.

The main features are:

- Programmable Baud Rate Generator
- Parity, Framing and Overrun Error Detection
- Line Break Generation and Detection
- Automatic Echo, Local Loopback and Remote Loopback channel modes
- Multi-drop Mode: Address Detection and Generation
- Interrupt Generation
- Two Dedicated Peripheral Data Controller channels
- 5-, 6-, 7-, 8- and 9-bit character length

18.1 Block Diagram

Figure 18-1. USART Block Diagram



18.2 Pin Description

Each USART channel has the following external signals:

Table 18-1.

Name	Description
SCK	USART Serial clock can be configured as input or output: SCK is configured as input if an External clock is selected (USCLKS[1] = 1) SCK is driven as output if the External Clock is disabled (USCLKS[1] = 0) and Clock output is enabled (CLKO = 1)
TXD	Transmit Serial Data is an output
RXD	Receive Serial Data is an input

Notes:
1. After a hardware reset, the USART pins are not enabled by default (see "PIO: Parallel I/O Controller" on page 93). The user must configure the PIO Controller before enabling the transmitter or receiver.
2. If the user selects one of the internal clocks, SCK can be configured as a PIO.

18.3 Baud Rate Generator

The Baud Rate Generator provides the bit period clock (the Baud Rate clock) to both the Receiver and the Transmitter.

The Baud Rate Generator can select between external and internal clock sources. The external clock source is SCK. The internal clock sources can be either the master clock (MCK) or the master clock divided by 8 (MCK/8).

Note: In all cases, if an external clock is used, the duration of each of its levels must be longer than the system clock (MCK) period. The external clock frequency must be at least 2.5 times lower than the system clock.

When the USART is programmed to operate in Asynchronous Mode (SYNC = 0 in the Mode Register US_MR), the selected clock is divided by 16 times the value (CD) written in US_BRGR (Baud Rate Generator Register). If US_BRGR is set to 0, the Baud Rate Clock is disabled.

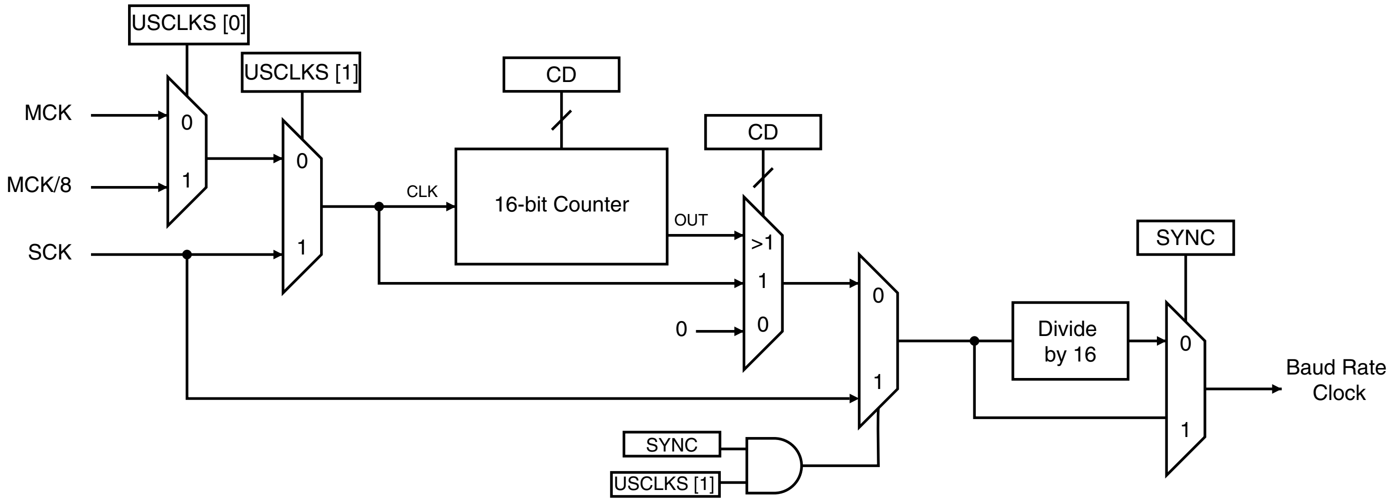
$$\text{Baud Rate} = \frac{\text{Selected Clock}}{16 \times \text{CD}}$$

When the USART is programmed to operate in Synchronous Mode (SYNC = 1) and the selected clock is internal (USCLKS[1] = 0 in the Mode Register US_MR), the Baud Rate Clock is the internal selected clock divided by the value written in US_BRGR. If US_BRGR is set to 0, the Baud Rate Clock is disabled.

$$\text{Baud Rate} = \frac{\text{Selected Clock}}{\text{CD}}$$

In Synchronous Mode with external clock selected (USCLKS[1] = 1), the clock is provided directly by the signal on the SCK pin. No division is active. The value written in US_BRGR has no effect.

Figure 18-2. Baud Rate Generator



18.4 Receiver

18.4.1 Asynchronous Receiver

The USART is configured for asynchronous operation when SYNC = 0 (bit 7 of US_MR). In Asynchronous Mode, the USART detects the start of a received character by sampling the RXD signal until it detects a valid start bit. A low level (space) on RXD is interpreted as a valid start bit if it is detected for more than 7 cycles of the sampling clock, which is 16 times the baud rate. Hence a space which is longer than 7/16 of the bit period is detected as a valid start bit. A space which is 7/16 of a bit period or shorter is ignored and the receiver continues to wait for a valid start bit.

When a valid start bit has been detected, the receiver samples the RXD at the theoretical midpoint of each bit. It is assumed that each bit lasts 16 cycles of the sampling clock (one bit period) so the sampling point is 8 cycles (0.5 bit periods) after the start of the bit. The first sampling point is therefore 24 cycles (1.5 bit periods) after the falling edge of the start bit was detected. Each subsequent bit is sampled 16 cycles (1 bit period) after the previous one.

Figure 18-3. Asynchronous Mode: Start Bit Detection

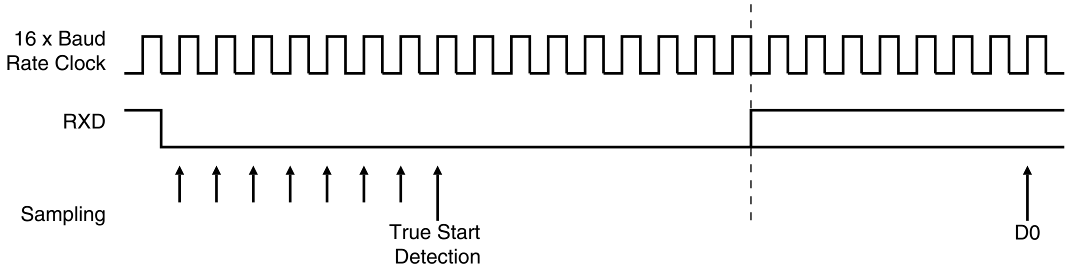
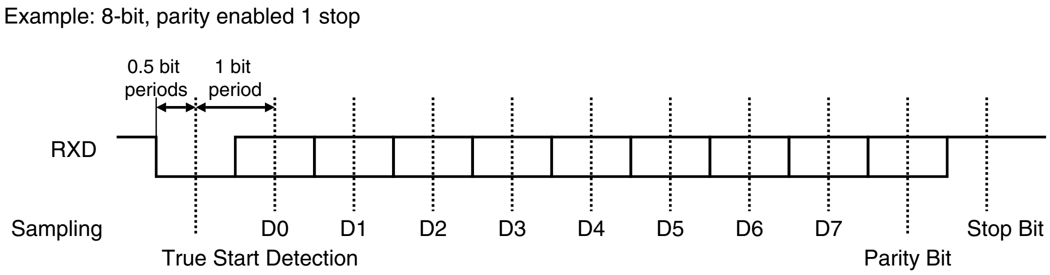


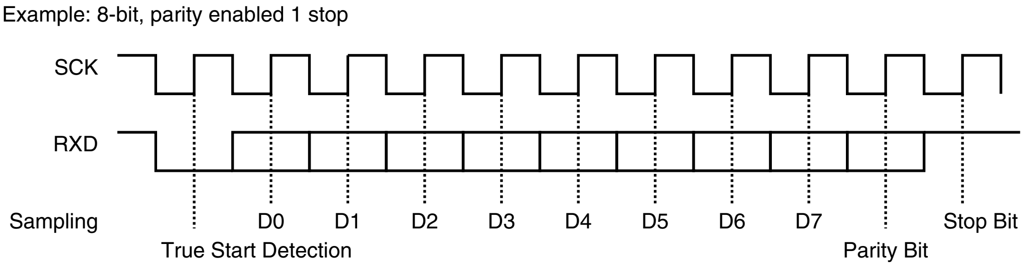
Figure 18-4. Asynchronous Mode: Character Reception



18.4.2 Synchronous Receiver

When configured for synchronous operation (SYNC = 1), the receiver samples the RXD signal on each rising edge of the Baud Rate clock. If a low level is detected, it is considered as a start. Data bits, parity bit and stop bit are sampled and the receiver waits for the next start bit. See example in Figure 18-5.

Figure 18-5. Synchronous Mode: Character Reception



18.4.3 Receiver Ready

When a complete character is received, it is transferred to the US_RHR and the RXRDY status bit in US_CSR is set. If US_RHR has not been read since the last transfer, the OVRE status bit in US_CSR is set.

18.4.4 Parity Error

Each time a character is received, the receiver calculates the parity of the received data bits, in accordance with the field PAR in US_MR. It then compares the result with the received parity bit. If different, the parity error bit PARE in US_CSR is set.

18.4.5 Framing Error

If a character is received with a stop bit at low level and with at least one data bit at high level, a framing error is generated. This sets FRAME in US_CSR.

18.4.6 Time-out

This function allows an idle condition on the RXD line to be detected. The maximum delay for which the USART should wait for a new character to arrive while the RXD line is inactive (high level) is programmed in US_RTOR (Receiver Time-out). When this register is set to 0, no time-out is detected. Otherwise, the receiver waits for a first character and then initializes a counter which is decremented at each bit period and reloaded at each byte reception. When the counter reaches 0, the TIMEOUT bit in US_CSR is set. The user can restart the wait for a first character with the STTTO (Start Time-out) bit in US_CR.

Calculation of time-out duration:

$$\text{Duration} = \text{Value} \times 4 \times \text{Bit period}$$

18.5 Transmitter

The transmitter has the same behavior in both synchronous and asynchronous operating modes. Start bit, data bits, parity bit and stop bits are serially shifted, lowest significant bit first, on the falling edge of the serial clock. See example in Figure 18-6.

The number of data bits is selected in the CHRL field in US_MR.

The parity bit is set according to the PAR field in US_MR.

The number of stop bits is selected in the NBSTOP field in US_MR.

When a character is written to US_THR (Transmit Holding), it is transferred to the Shift Register as soon as it is empty. When the transfer occurs, the TXRDY bit in US_CSR is set until a new character is written to US_THR. If Transmit Shift Register and US_THR are both empty, the TXEMPTY bit in US_CSR is set.

18.5.1 Time-guard

The Time-guard function allows the transmitter to insert an idle state on the TXD line between two characters. The duration of the idle state is programmed in US_TTGR (Transmitter Time-guard). When this register is set to zero, no time-guard is generated. Otherwise, the transmitter holds a high level on TXD after each transmitted byte during the number of bit periods programmed in US_TTGR

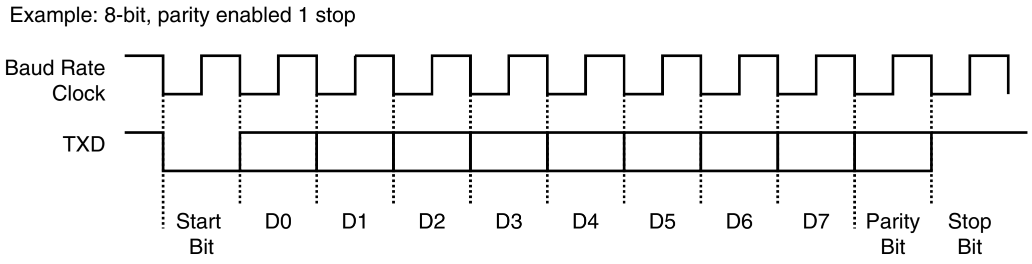
Idle state duration between two characters = Time-guard Value x Bit Period

18.5.2 Multi-drop Mode

When the field PAR in US_MR equals 11X (binary value), the USART is configured to run in Multi-drop Mode. In this case, the parity error bit PARE in US_CSR is set when data is detected with a parity bit set to identify an address byte. PARE is cleared with the Reset Status Bits Command (RSTSTA) in US_CR. If the parity bit is detected low, identifying a data byte, PARE is not set.

The transmitter sends an address byte (parity bit set) when a Send Address Command (SENDA) is written to US_CR. In this case, the next byte written to US_THR will be transmitted as an address. After this any byte transmitted will have the parity bit cleared.

Figure 18-6. Synchronous and Asynchronous Modes: Character Transmission



18.6 Break

A break condition is a low signal level which has a duration of at least one character (including start/stop bits and parity).

18.6.1 Transmit Break

The transmitter generates a break condition on the TXD line when STTBRK is set in US_CR (Control Register). In this case, the character present in the Transmit Shift Register is completed before the line is held low.

To cancel a break condition on the TXD line, the STPBRK command in US_CR must be set. The USART completes a minimum break duration of one character length. The TXD line then returns to high level (idle state) for at least 12 bit periods to ensure that the end of break is correctly detected. Then the transmitter resumes normal operation.

The BREAK is managed like a character:

- The STTBRK and the STPBRK commands are performed only if the transmitter is ready (bit TXRDY = 1 in US_CSR)
- The STTBRK command blocks the transmitter holding register (bit TXRDY is cleared in US_CSR) until the break has started
- A break is started when the Shift Register is empty (any previous character is fully transmitted). TXEMPTY is cleared in US_CSR. The break blocks the transmitter shift register until it is completed (high level for at least 12-bit periods after the STPBRK command is requested)

In order to avoid unpredictable states:

- STTBRK and STPBRK commands must not be requested at the same time
- Once an STTBRK command is requested, further STTBRK commands are ignored until the BREAK is ended (high level for at least 12-bit periods)
- All STPBRK commands requested without a previous STTBRK command are ignored
- A byte written into the Transmit Holding Register while a break is pending but not started (US_CSR.TXRDY = 0) is ignored
- It is *not permitted* to write new data in the Transmit Holding Register while a break is in progress (STPBRK has not been requested), even though TXRDY = 1 in US_CSR.
- A new STTBRK command *must not* be issued until an existing break has ended (TXEMPTY = 1 in US_CSR)

The standard break transmission sequence is:

1. Wait for the transmitter ready
   (US_CSR.TXRDY = 1)
2. Send the STTBRK command
   (write 0x0200 to US_CR)
3. Wait for the transmitter ready
   (TXRDY = 1 in US_CSR)
4. Send the STPBRK command
   (write 0x0400 to US_CR)

The next byte can then be sent:

5. Wait for the transmitter ready
   (TXRDY = 1 in US_CSR)

6. Send the next byte
   (write byte to US_THR)

Each of these steps can be scheduled by using the interrupt if the bit TXRDY in US_IMR is set. For character transmission, the USART channel must be enabled before sending a break.

18.6.2 Receive Break

The receiver detects a break condition when all data, parity and stop bits are low. When the low stop bit is detected, the receiver asserts the RXBRK bit in US_CSR. An end of receive break is detected by a high level for at least 2/16 of a bit period in Asynchronous Mode or at least one sample in Synchronous Mode. RXBRK is also asserted when an end of break is detected.

Both the beginning and the end of a break can be detected by interrupt if the bit US_IMR.RXBRK is set.

18.7 Peripheral Data Controller

Each USART channel is closely connected to a corresponding Peripheral Data Controller channel. One is dedicated to the receiver. The other is dedicated to the transmitter.

Note: The PDC is disabled if 9-bit character length is selected (MODE9 = 1) in US_MR.

The PDC channel is programmed using US_TPR (Transmit Pointer) and US_TCR (Transmit Counter) for the transmitter and US_RPR (Receive Pointer) and US_RCR (Receive Counter) for the receiver. The status of the PDC is given in US_CSR by the ENDTX bit for the transmitter and by the ENDRX bit for the receiver.

The pointer registers (US_TPR and US_RPR) are used to store the address of the transmit or receive buffers. The counter registers (US_TCR and US_RCR) are used to store the size of these buffers.

The receiver data transfer is triggered by the RXRDY bit and the transmitter data transfer is triggered by TXRDY. When a transfer is performed, the counter is decremented and the pointer is incremented. When the counter reaches 0, the status bit is set (ENDRX for the receiver, ENDTX for the transmitter in US_CSR) which can be programmed to generate an interrupt. Transfers are then disabled until a new non-zero counter value is programmed.

18.8 Interrupt Generation

Each status bit in US_CSR has a corresponding bit in US_IER (Interrupt Enable) and US_IDR (Interrupt Disable) which controls the generation of interrupts by asserting the USART interrupt line connected to the Advanced Interrupt Controller. US_IMR (Interrupt Mask Register) indicates the status of the corresponding bits.

When a bit is set in US_CSR and the same bit is set in US_IMR, the interrupt line is asserted.

18.9 Channel Modes

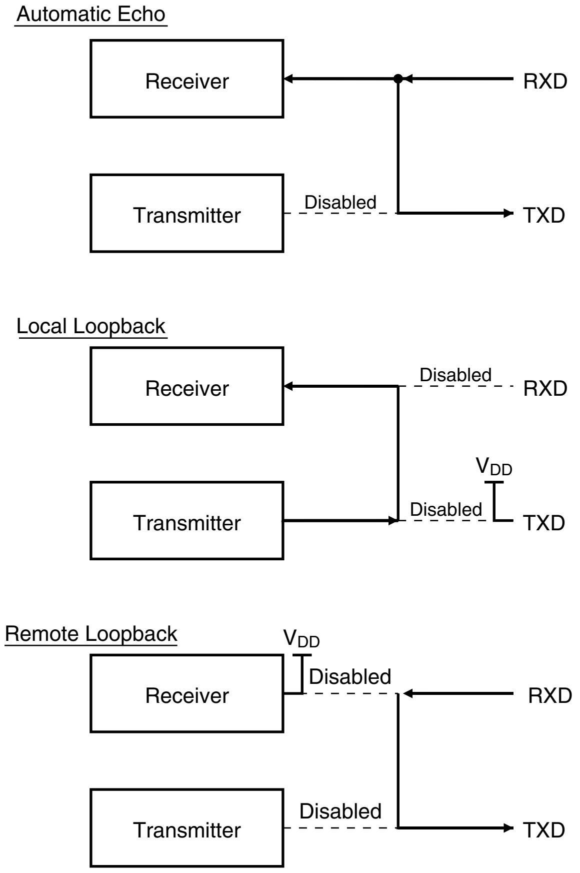
The USART can be programmed to operate in three different test modes, using the field CHMODE in US_MR.

Automatic Echo Mode allows bit by bit re-transmission. When a bit is received on the RXD line, it is sent to the TXD line. Programming the transmitter has no effect.

Local Loopback Mode allows the transmitted characters to be received. TXD and RXD pins are not used and the output of the transmitter is internally connected to the input of the receiver. The RXD pin level has no effect and the TXD pin is held high, as in idle state.

Remote Loopback Mode directly connects the RXD pin to the TXD pin. The Transmitter and the Receiver are disabled and have no effect. This mode allows bit by bit re-transmission.

Figure 18-7. Channel Modes



18.10 USART User Interface

Base Address USART0: 0xFFFD0000 (Code Label `USART0_BASE`)

Base Address USART1: 0xFFFCC000 (Code Label `USART1_BASE`)

Table 18-2. USART Memory Map

Offset	Register	Name	Access	Reset State
0x00	Control Register	US_CR	Write-only	–
0x04	Mode Register	US_MR	Read/Write	0
0x08	Interrupt Enable Register	US_IER	Write-only	–
0x0C	Interrupt Disable Register	US_IDR	Write-only	–
0x10	Interrupt Mask Register	US_IMR	Read-only	0
0x14	Channel Status Register	US_CSR	Read-only	0x18
0x18	Receiver Holding Register	US_RHR	Read-only	0
0x1C	Transmitter Holding Register	US_THR	Write-only	–
0x20	Baud Rate Generator Register	US_BRGR	Read/Write	0
0x24	Receiver Time-out Register	US_RTOR	Read/Write	0
0x28	Transmitter Time-guard Register	US_TTGR	Read/Write	0
0x2C	Reserved	–	–	–
0x30	Receive Pointer Register	US_RPR	Read/Write	0
0x34	Receive Counter Register	US_RCR	Read/Write	0
0x38	Transmit Pointer Register	US_TPR	Read/Write	0
0x3C	Transmit Counter Register	US_TCR	Read/Write	0

18.10.1 USART Control Register

Name: US_CR

Access Type:Write-only

Offset: 0x00

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	SENDA	STTTO	STPBRK	STTBRK	RSTSTA
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX	–	–

- **RSTRX: Reset Receiver (Code Label `US_RSTRX`)**

0 = No effect.

1 = The receiver logic is reset.

- **RSTTX: Reset Transmitter (Code Label `US_RSTTX`)**

0 = No effect.

1 = The transmitter logic is reset.

- **RXEN: Receiver Enable (Code Label `US_RXEN`)**

0 = No effect.

1 = The receiver is enabled if RXDIS is 0.

- **RXDIS: Receiver Disable (Code Label `US_RXDIS`)**

0 = No effect.

1 = The receiver is disabled.

- **TXEN: Transmitter Enable (Code Label `US_TXEN`)**

0 = No effect.

1 = The transmitter is enabled if TXDIS is 0.

- **TXDIS: Transmitter Disable (Code Label `US_TXDIS`)**

0 = No effect.

1 = The transmitter is disabled.

- **RSTSTA: Reset Status Bits (Code Label `US_RSTSTA`)**

0 = No effect.

1 = Resets the status bits PARE, FRAME, OVRE and RXBRK in the US_CSR.

- **STTBRK: Start Break (Code Label `US_STTBRK`)**

0 = No effect.

1 = If break is not being transmitted, start transmission of a break after the characters present in US_THR and the Transmit Shift Register have been transmitted.

- **STPBRK: Stop Break (Code Label `US_STPBRK`)**

0 = No effect.

1 = If a break is being transmitted, stop transmission of the break after a minimum of one character length and transmit a high level during 12 bit periods.

- **STTTO: Start Time-out (Code Label `US_STTTO`)**

0 = No effect.

1 = Start waiting for a character before clocking the time-out counter.

- **SENDA: Send Address (Code Label `US_SENDA`)**

0 = No effect.

1 = In Multi-drop Mode only, the next character written to the US_THR is sent with the address bit set.

18.10.2 USART Mode Register

Name: US_MR

Access Type:Read/Write

Reset Value: 0

Offset: 0x04

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	CLKO	MODE9	–

15	14	13	12	11	10	9	8
CHMODE		NBSTOP		PAR			SYNC

7	6	5	4	3	2	1	0
CHRL		USCLKS		–	–	–	–

- **USCLKS: Clock Selection (Baud Rate Generator Input Clock)**

			Code Label
USCLKS		Selected Clock	US_CLKS
0	0	MCK	US_CLKS_MCK
0	1	MCK/8	US_CLKS_MCK8
1	X	External (SCK)	US_CLKS_SCK

- **CHRL: Character Length**

			Code Label
CHRL		Character Length	US_CHRL
0	0	Five bits	US_CHRL_5
0	1	Six bits	US_CHRL_6
1	0	Seven bits	US_CHRL_7
1	1	Eight bits	US_CHRL_8

Start, stop and parity bits are added to the character length.

- **SYNC: Synchronous Mode Select (Code Label US_SYNC)**

0 = USART operates in Asynchronous Mode.

1 = USART operates in Synchronous Mode.

- **PAR: Parity Type**

PAR			Parity Type	Code Label
				US_PAR
0	0	0	Even Parity	US_PAR_EVEN
0	0	1	Odd Parity	US_PAR_ODD
0	1	0	Parity forced to 0 (Space)	US_PAR_SPACE
0	1	1	Parity forced to 1 (Mark)	US_PAR_MARK
1	0	x	No parity	US_PAR_NO
1	1	x	Multi-drop mode	US_PAR_MULTIDROP

- **NBSTOP: Number of Stop Bits**

The interpretation of the number of stop bits depends on SYNC.

NBSTOP		Asynchronous (SYNC = 0)	Synchronous (SYNC = 1)	Code Label
				US_NBSTOP
0	0	1 stop bit	1 stop bit	US_NBSTOP_1
0	1	1.5 stop bits	Reserved	US_NBSTOP_1_5
1	0	2 stop bits	2 stop bits	US_NBSTOP_2
1	1	Reserved	Reserved	–

- **CHMODE: Channel Mode**

CHMODE		Mode Description	Code Label
			US_CHMODE
0	0	Normal Mode The USART Channel operates as an Rx/Tx USART.	US_CHMODE_NORMAL
0	1	Automatic Echo Receiver Data Input is connected to TXD pin.	US_CHMODE_AUTOMATIC_ECHO
1	0	Local Loopback Transmitter Output Signal is connected to Receiver Input Signal.	US_CHMODE_LOCAL_LOOPBACK
1	1	Remote Loopback RXD pin is internally connected to TXD pin.	US_CHMODE_REMODE_LOOPBACK

- **MODE9: 9-bit Character Length (Code Label `US_MODE9`)**

0 = CHRL defines character length.

1 = 9-bit character length.

- **CKLO: Clock Output Select (Code Label `US_CLKO`)**

0 = The USART does not drive the SCK pin.

1 = The USART drives the SCK pin if USCLKS[1] is 0.

18.10.3 USART Interrupt Enable Register

Name: US_IER

Access Type:Write-only

Offset: 0x08

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	–	–	–	TXEMPTY	TIMEOUT
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY

- **RXRDY: Enable RXRDY Interrupt (Code Label `US_RXRDY`)**

0 = No effect.

1 = Enables RXRDY Interrupt.

- **TXRDY: Enable TXRDY Interrupt (Code Label `US_TXRDY`)**

0 = No effect.

1 = Enables TXRDY Interrupt.

- **RXBRK: Enable Receiver Break Interrupt (Code Label `US_RXBRK`)**

0 = No effect.

1 = Enables Receiver Break Interrupt.

- **ENDRX: Enable End of Receive Transfer Interrupt (Code Label `US_ENDRX`)**

0 = No effect.

1 = Enables End of Receive Transfer Interrupt.

- **ENDTX: Enable End of Transmit Interrupt (Code Label `US_ENDTX`)**

0 = No effect.

1 = Enables End of Transmit Interrupt.

- **OVRE: Enable Overrun Error Interrupt (Code Label `US_OVRE`)**

0 = No effect.

1 = Enables Overrun Error Interrupt.

- **FRAME: Enable Framing Error Interrupt (Code Label `US_FRAME`)**

0 = No effect.

1 = Enables Framing Error Interrupt.

- **PARE: Enable Parity Error Interrupt (Code Label `US_PARE`)**

0 = No effect.

1 = Enables Parity Error Interrupt.

- **TIMEOUT: Enable Time-out Interrupt (Code Label `US_TIMEOUT`)**

0 = No effect.

1 = Enables Reception Time-out Interrupt.

- **TXEMPTY: Enable TXEMPTY Interrupt (Code Label `US_TXEMPTY`)**

0 = No effect.

1 = Enables TXEMPTY Interrupt.

18.10.4 USART Interrupt Disable Register

Name: US_IDR

Access Type:Write-only

Offset: 0x0C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	–	–	–	TXEMPTY	TIMEOUT
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY

- **RXRDY: Disable RXRDY Interrupt (Code Label `US_RXRDY`)**

0 = No effect.

1 = Disables RXRDY Interrupt.

- **TXRDY: Disable TXRDY Interrupt (Code Label `US_TXRDY`)**

0 = No effect.

1 = Disables TXRDY Interrupt.

- **RXBRK: Disable Receiver Break Interrupt (Code Label `US_RXBRK`)**

0 = No effect.

1 = Disables Receiver Break Interrupt.

- **ENDRX: Disable End of Receive Transfer Interrupt (Code Label `US_ENDRX`)**

0 = No effect.

1 = Disables End of Receive Transfer Interrupt.

- **ENDTX: Disable End of Transmit Interrupt (Code Label `US_ENDTX`)**

0 = No effect.

1 = Disables End of Transmit Interrupt.

- **OVRE: Disable Overrun Error Interrupt (Code Label `US_OVRE`)**

0 = No effect.

1 = Disables Overrun Error Interrupt.

- **FRAME: Disable Framing Error Interrupt (Code Label `US_FRAME`)**

0 = No effect.

1 = Disables Framing Error Interrupt.

- **PARE: Disable Parity Error Interrupt (Code Label `US_PARE`)**

0 = No effect.

1 = Disables Parity Error Interrupt.

- **TIMEOUT: Disable Time-out Interrupt (Code Label `US_TIMEOUT`)**

0 = No effect.

1 = Disables Receiver Time-out Interrupt.

- **TXEMPTY: Disable TXEMPTY Interrupt (Code Label `US_TXEMPTY`)**

0 = No effect.

1 = Disables TXEMPTY Interrupt.

18.10.5 USART Interrupt Mask Register

Name: US_IMR

Access Type:Read-only

Reset Value: 0

Offset: 0x10

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	TXEMPTY	TIMEOUT
7	6	5	4	3	2	1	0
PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY

- **RXRDY: Mask RXRDY Interrupt (Code Label `US_RXRDY`)**

0 = RXRDY Interrupt is Disabled

1 = RXRDY Interrupt is Enabled

- **TXRDY: Mask TXRDY Interrupt (Code Label `US_TXRDY`)**

0 = TXRDY Interrupt is Disabled

1 = TXRDY Interrupt is Enabled

- **RXBRK: Mask Receiver Break Interrupt (Code Label `US_RXBRK`)**

0 = Receiver Break Interrupt is Disabled

1 = Receiver Break Interrupt is Enabled

- **ENDRX: Mask End of Receive Transfer Interrupt (Code Label `US_ENDRX`)**

0 = End of Receive Transfer Interrupt is Disabled

1 = End of Receive Transfer Interrupt is Enabled

- **ENDTX: Mask End of Transmit Interrupt (Code Label `US_ENDTX`)**

0 = End of Transmit Interrupt is Disabled

1 = End of Transmit Interrupt is Enabled

- **OVRE: Mask Overrun Error Interrupt (Code Label `US_OVRE`)**

0 = Overrun Error Interrupt is Disabled

1 = Overrun Error Interrupt is Enabled

- **FRAME: Mask Framing Error Interrupt (Code Label `US_FRAME`)**

0 = Framing Error Interrupt is Disabled

1 = Framing Error Interrupt is Enabled

- **PARE: Mask Parity Error Interrupt (Code Label `US_PARE`)**

0 = Parity Error Interrupt is Disabled

1 = Parity Error Interrupt is Enabled

- **TIMEOUT: Mask Time-out Interrupt (Code Label `US_TIMEOUT`)**

0 = Receive Time-out Interrupt is Disabled

1 = Receive Time-out Interrupt is Enabled

- **TXEMPTY: Mask TXEMPTY Interrupt (Code Label `US_TXEMPTY`)**

0 = TXEMPTY Interrupt is Disabled.

1 = TXEMPTY Interrupt is Enabled.

18.10.6 USART Channel Status Register

Name: US_CSR

Access Type:Read-only

Reset Value: 0x18

Offset: 0x14

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	TXEMPTY	TIMEOUT

7	6	5	4	3	2	1	0
PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY

- **RXRDY: Receiver Ready (Code Label `US_RXRDY`)**

0 = No complete character has been received since the last read of the US_RHR or the receiver is disabled.

1 = At least one complete character has been received and the US_RHR has not yet been read.

- **TXRDY: Transmitter Ready (Code Label `US_TXRDY`)**

0 = US_THR contains a character waiting to be transferred to the Transmit Shift Register, or an STTBRK command has been requested.

1 = US_THR is empty and there is no Break request pending TSR availability.

Equal to zero when the USART is disabled or at reset. Transmitter Enable command (in US_CR) sets this bit to one.

- **RXBRK: Break Received/End of Break (Code Label `US_RXBRK`)**

0 = No Break Received nor End of Break has been detected since the last “Reset Status Bits” command in the Control Register.

1 = Break Received or End of Break has been detected since the last “Reset Status Bits” command in the Control Register.

- **ENDRX: End of Receiver Transfer (Code Label `US_ENDRX`)**

0 = The End of Transfer signal from the Peripheral Data Controller channel dedicated to the receiver is inactive.

1 = The End of Transfer signal from the Peripheral Data Controller channel dedicated to the receiver is active.

- **ENDTX: End of Transmitter Transfer (Code Label `US_ENDTX`)**

0 = The End of Transfer signal from the Peripheral Data Controller channel dedicated to the transmitter is inactive.

1 = The End of Transfer signal from the Peripheral Data Controller channel dedicated to the transmitter is active.

- **OVRE: Overrun Error (Code Label `US_OVRE`)**

0 = No byte has been transferred from the Receive Shift Register to the US_RHR when RxRDY was asserted since the last “Reset Status Bits” command.

1 = At least one byte has been transferred from the Receive Shift Register to the US_RHR when RxRDY was asserted since the last “Reset Status Bits” command.

- **FRAME: Framing Error (Code Label `US_FRAME`)**

0 = No stop bit has been detected low since the last “Reset Status Bits” command.

1 = At least one stop bit has been detected low since the last “Reset Status Bits” command.

- **PARE: Parity Error (Code Label `US_PARE`)**

1 = At least one parity bit has been detected false (or a parity bit high in Multi-drop Mode) since the last “Reset Status Bits” command.

0 = No parity bit has been detected false (or a parity bit high in Multi-drop Mode) since the last “Reset Status Bits” command.

- **TIMEOUT: Receiver Time-out (Code Label `US_TIMEOUT`)**

0 = There has not been a time-out since the last “Start Time-out” command or the Time-out Register is 0.

1 = There has been a time-out since the last “Start Time-out” command.

- **TXEMPTY: Transmitter Empty (Code Label `US_TXEMPTY`)**

0 = There are characters in either US_THR or the Transmit Shift Register or a Break is being transmitted.

1 = There are no characters in US_THR and the Transmit Shift Register and Break is not active.

Equal to zero when the USART is disabled or at reset. Transmitter Enable command (in US_CR) sets this bit to one.

18.10.7 USART Receiver Holding Register

Name: US_RHR

Access Type:Read-only

Reset Value: 0

Offset: 0x18

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
RXCHR							

- **RXCHR: Received Character**

Last character received if RXRDY is set. When number of data bits is less than 8 bits, the bits are right-aligned.

All non-significant bits read zero.

18.10.8 USART Transmitter Holding Register

Name: US_THR

Access Type:Write-only

Offset: 0x1C

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
TXCHR							

- **TXCHR: Character to be Transmitted**

Next character to be transmitted after the current character if TXRDY is not set. When number of data bits is less than 8 bits, the bits are right-aligned.

18.10.9 USART Baud Rate Generator Register

Name: US_BRGR

Access Type:Read/Write

Reset Value: 0

Offset: 0x20

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
CD							

7	6	5	4	3	2	1	0
CD							

- **CD: Clock Divisor**

This register has no effect if Synchronous Mode is selected with an external clock.

CD	Effect
0	Disables Clock
1	Clock Divisor Bypass [1]
2 to 65535	Baud Rate (Asynchronous Mode) = Selected Clock / (16 x CD) Baud Rate (Synchronous Mode) = Selected Clock / CD [2]

Notes: 1. Clock divisor bypass (CD = 1) must not be used when internal clock MCK is selected (USCLKS = 0).

2. In Synchronous Mode, the value programmed must be even to ensure a 50:50 mark:space ratio.

18.10.10 USART Receiver Time-out Register

Name: US_RTOR

Access Type:Read/Write

Reset Value: 0

Offset: 0x24

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**
TO							

- **TO: Time-out Value**

When a value is written to this register, a Start Time-out Command is automatically performed.

TO	
0	Disables the RX Time-out function.
1 - 255	The Time-out counter is loaded with TO when the Start Time-out Command is given or when each new data character is received (after reception has started).

Time-out duration = TO x 4 x Bit period

18.10.11 USART Transmitter Time-guard Register

Name: US_TTGR

Access Type:Read/Write

Reset Value: 0

Offset: 0x28

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**
TG							

- **TG: Time-guard Value**

TG	
0	Disables the TX Time-guard function.
1 - 255	TXD is inactive high after the transmission of each character for the time-guard duration.

Time-guard duration = TG x Bit period

18.10.12 USART Receive Pointer Register

Name: US_RPR

Access Type:Read/Write

Reset Value: 0

Offset: 0x30

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

- **RXPTR: Receive Pointer**

RXPTR must be loaded with the address of the receive buffer.

18.10.13 USART Receive Counter Register

Name: US_RCR

Access Type:Read/Write

Reset Value: 0

Offset: 0x34

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	4920	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
RXCTR							
7	6	5	4	3	2	1	0
RXCTR							

- **RXCTR: Receive Counter**

RXCTR must be loaded with the size of the receive buffer.

0: Stop Peripheral Data Transfer dedicated to the receiver.

1 - 65535: Start Peripheral Data transfer if RXRDY is active.

18.10.14 USART Transmit Pointer Register

Name: US_TPR

Access Type:Read/Write

Reset Value: 0

Offset: 0x38

31	30	29	28	27	26	25	24
TXPTR							
23	22	21	20	19	18	17	16
TXPTR							
15	14	13	12	11	10	9	8
TXPTR							
7	6	5	4	3	2	1	0
TXPTR							

- **TXPTR: Transmit Pointer**

TXPTR must be loaded with the address of the transmit buffer.

18.10.15 USART Transmit Counter Register

Name: US_TCR

Access Type:Read/Write

Reset Value: 0

Offset: 0x3C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
TXCTR							

7	6	5	4	3	2	1	0
TXCTR							

- **TXCTR: Transmit Counter**

TXCTR must be loaded with the size of the transmit buffer.

0: Stop Peripheral Data Transfer dedicated to the transmitter.

1 - 65535: Start Peripheral Data transfer if TXRDY is active.

19. TC: Timer Counter

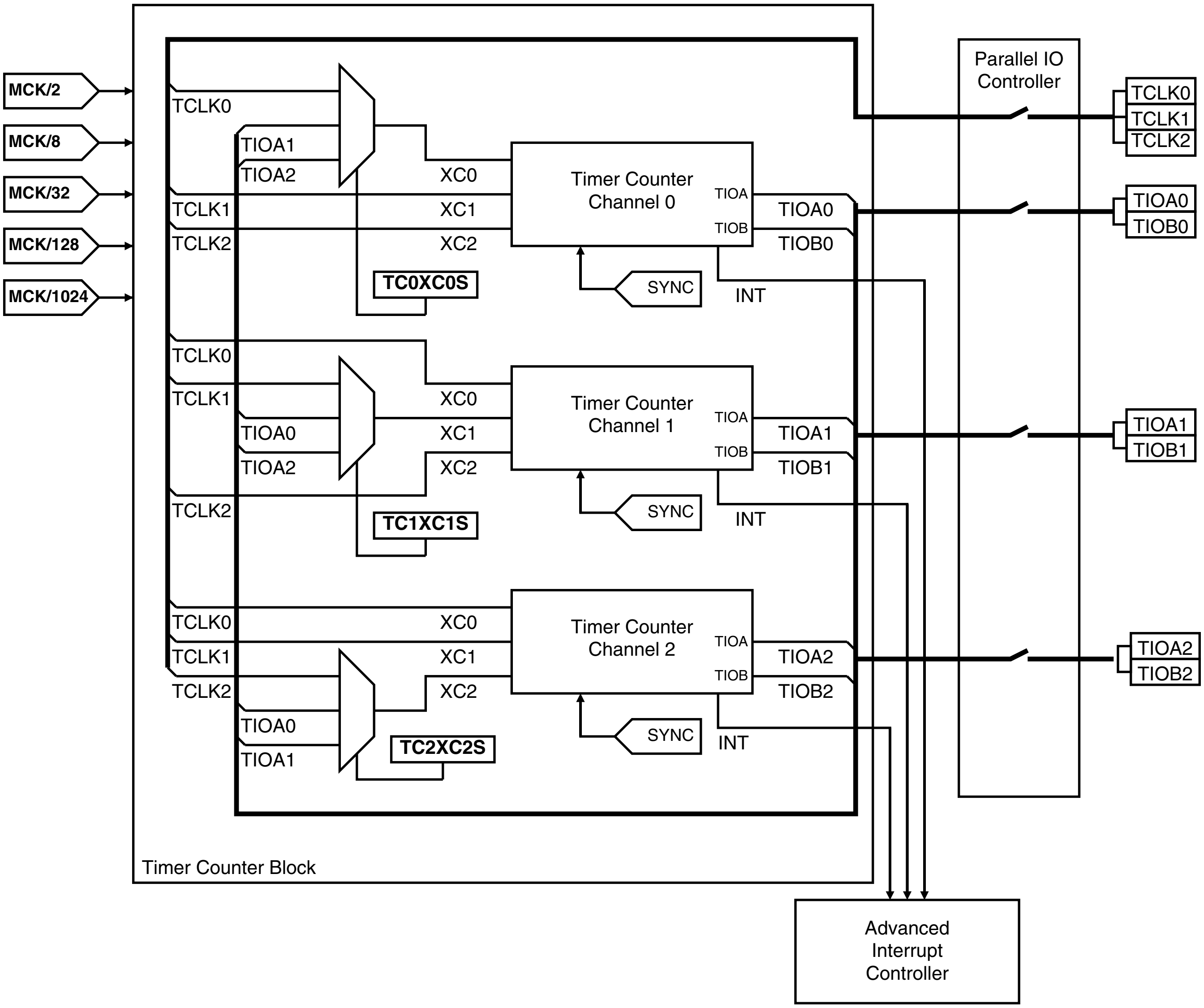
The AT91FR40162SB features a Timer Counter block which includes three identical 16-bit timer counter channels. Each channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation.

Each Timer Counter channel has 3 external clock inputs, 5 internal clock inputs, and 2 multi-purpose input/output signals which can be configured by the user. Each channel drives an internal interrupt signal which can be programmed to generate processor interrupts via the AIC (Advanced Interrupt Controller).

The Timer Counter block has two global registers which act upon all three TC channels. The Block Control Register allows the three channels to be started simultaneously with the same instruction. The Block Mode Register defines the external clock inputs for each Timer Counter channel, allowing them to be chained.

19.1 Block Diagram

Figure 19-1. TC Block Diagram



19.2 Signal Description

Table 19-1. Signal Description

Channel Signal	Description
XC0, XC1, XC2	External Clock Inputs
TIOA	Capture Mode: General Purpose Input Waveform Mode: General Purpose Output
TIOB	Capture Mode: General Purpose Input Waveform Mode: General Purpose Input/Output
INT	Interrupt Signal Output
SYNC	Synchronization Input Signal
Block Signals	**Description**
TCLK0, TCLK1, TCLK2	External Clock Inputs
TIOA0	TIOA Signal for Channel 0
TIOB0	TIOB Signal for Channel 0
TIOA1	TIOA Signal for Channel 1
TIOB1	TIOB Signal for Channel 1
TIOA2	TIOA Signal for Channel 2
TIOB2	TIOB Signal for Channel 2

Note: After a hardware reset, the Timer Counter block pins are controlled by the PIO Controller. They must be configured to be controlled by the peripheral before being used.

19.3 Timer Counter Description

The three Timer Counter channels are independent and identical in operation. The registers for channel programming are listed in Table 19-3 on page 159.

19.3.1 Counter

Each Timer Counter channel is organized around a 16-bit counter. The value of the counter is incremented at each positive edge of the selected clock. When the counter has reached the value 0xFFFF and passes to 0x0000, an overflow occurs and the bit COVFS in TC_SR (Status Register) is set.

The current value of the counter is accessible in real-time by reading TC_CV. The counter can be reset by a trigger. In this case, the counter value passes to 0x0000 on the next valid edge of the selected clock.

19.3.2 Clock Selection

At block level, input clock signals of each channel can either be connected to the external inputs TCLK0, TCLK1 or TCLK2, or be connected to the configurable I/O signals TIOA0, TIOA1 or TIOA2 for chaining by programming the TC_BMR (Block Mode).

Each channel can independently select an internal or external clock source for its counter:

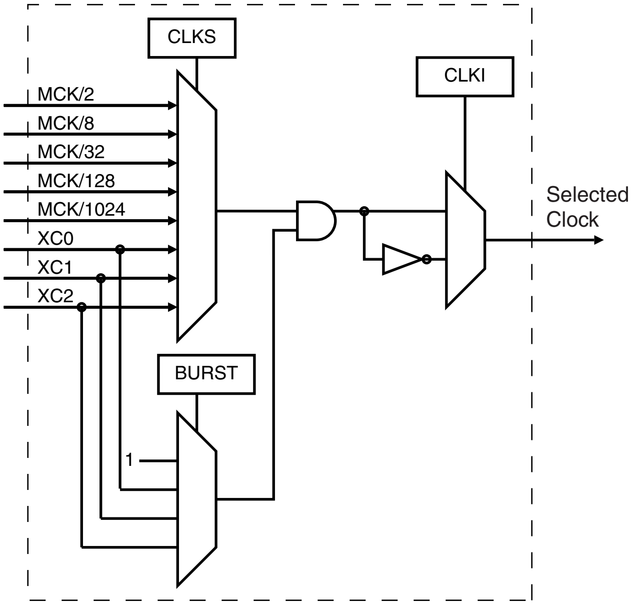
- Internal clock signals: MCK/2, MCK/8, MCK/32, MCK/128, MCK/1024
- External clock signals: XC0, XC1 or XC2

The selected clock can be inverted with the CLKI bit in TC_CMR (Channel Mode). This allows counting on the opposite edges of the clock.

The burst function allows the clock to be validated when an external signal is high. The BURST parameter in the Mode Register defines this signal (none, XC0, XC1, XC2).

Note: In all cases, if an external clock is used, the duration of each of its levels must be longer than the system clock (MCK) period. The external clock frequency must be at least 2.5 times lower than the system clock (MCK).

Figure 19-2. Clock Selection

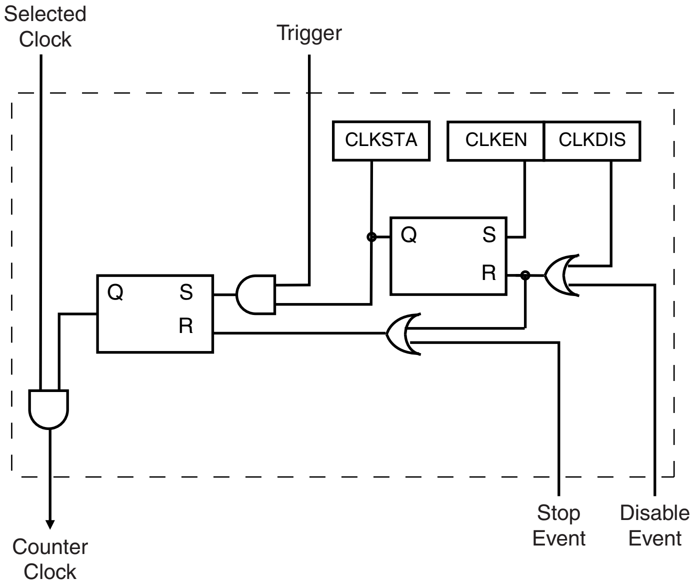


19.3.3 Clock Control

The clock of each counter can be controlled in two different ways: it can be enabled/disabled and started/stopped.

- The clock can be **enabled** or **disabled** by the user with the CLKEN and the CLKDIS commands in the Control Register. In Capture Mode it can be disabled by an RB load event if LDBDIS is set to 1 in TC_CMR. In Waveform Mode, it can be disabled by an RC Compare event if CPCDIS is set to 1 in TC_CMR. When disabled, the start or the stop actions have no effect: only a CLKEN command in the Control Register can re-enable the clock. When the clock is enabled, the CLKSTA bit is set in the Status Register.
- The clock can also be **started** or **stopped**: a trigger (software, synchro, external or compare) always starts the clock. The clock can be stopped by an RB load event in Capture Mode (LDBSTOP = 1 in TC_CMR) or a RC compare event in Waveform Mode (CPCSTOP = 1 in TC_CMR). The start and the stop commands have effect only if the clock is enabled.

Figure 19-3. Clock Control



19.3.4 Timer Counter Operating Modes

Each Timer Counter channel can independently operate in two different modes:

- Capture Mode allows measurement on signals
- Waveform Mode allows wave generation

The Timer Counter Operating Mode is programmed with the WAVE bit in the TC Mode Register. In Capture Mode, TIOA and TIOB are configured as inputs. In Waveform Mode, TIOA is always configured to be an output and TIOB is an output if it is not selected to be the external trigger.

19.3.5 Trigger

A trigger resets the counter and starts the counter clock. Three types of triggers are common to both modes, and a fourth external trigger is available to each mode.

The following triggers are common to both modes:

- Software Trigger: Each channel has a software trigger, available by setting SWTRG in TC_CCR.
- SYNC: Each channel has a synchronization signal SYNC. When asserted, this signal has the same effect as a software trigger. The SYNC signals of all channels are asserted simultaneously by writing TC_BCR (Block Control) with SYNC set.
- Compare RC Trigger: RC is implemented in each channel and can provide a trigger when the counter value matches the RC value if CPCTRG is set in TC_CMR.

The Timer Counter channel can also be configured to have an external trigger. In Capture Mode, the external trigger signal can be selected between TIOA and TIOB. In Waveform Mode, an external event can be programmed on one of the following signals: TIOB, XC0, XC1 or XC2. This external event can then be programmed to perform a trigger by setting ENETRG in TC_CMR.

If an external trigger is used, the duration of the pulses must be longer than the system clock (MCK) period in order to be detected.

Whatever the trigger used, it will be taken into account at the following active edge of the selected clock. This means that the counter value may not read zero just after a trigger, especially when a low frequency signal is selected as the clock.

19.4 Capture Operating Mode

This mode is entered by clearing the WAVE parameter in TC_CMR (Channel Mode Register). Capture Mode allows the TC Channel to perform measurements such as pulse timing, frequency, period, duty cycle and phase on TIOA and TIOB signals which are inputs.

Figure shows the configuration of the TC Channel when programmed in Capture Mode.

19.4.1 Capture Registers A and B (RA and RB)

Registers A and B are used as capture registers. This means that they can be loaded with the counter value when a programmable event occurs on the signal TIOA.

The parameter LDRA in TC_CMR defines the TIOA edge for the loading of register A, and the parameter LDRB defines the TIOA edge for the loading of Register B.

RA is loaded only if it has not been loaded since the last trigger or if RB has been loaded since the last loading of RA.

RB is loaded only if RA has been loaded since the last trigger or the last loading of RB.

Loading RA or RB before the read of the last value loaded sets the Overrun Error Flag (LOVRS) in TC_SR (Status Register). In this case, the old value is overwritten.

19.4.2 Trigger Conditions

In addition to the SYNC signal, the software trigger and the RC compare trigger, an external trigger can be defined.

Bit ABETRG in TC_CMR selects input signal TIOA or TIOB as an external trigger. Parameter ETRGEDG defines the edge (rising, falling or both) detected to generate an external trigger. If ETRGEDG = 0 (none), the external trigger is disabled.

19.4.3 Status Register

The following bits in the status register are significant in Capture Operating Mode.

- CPCS: RC Compare Status

  There has been an RC Compare match at least once since the last read of the status

- COVFS: Counter Overflow Status

  The counter has attempted to count past $FFFF since the last read of the status

- LOVRS: Load Overrun Status

  RA or RB has been loaded at least twice without any read of the corresponding register, since the last read of the status

- LDRAS: Load RA Status

  RA has been loaded at least once without any read, since the last read of the status

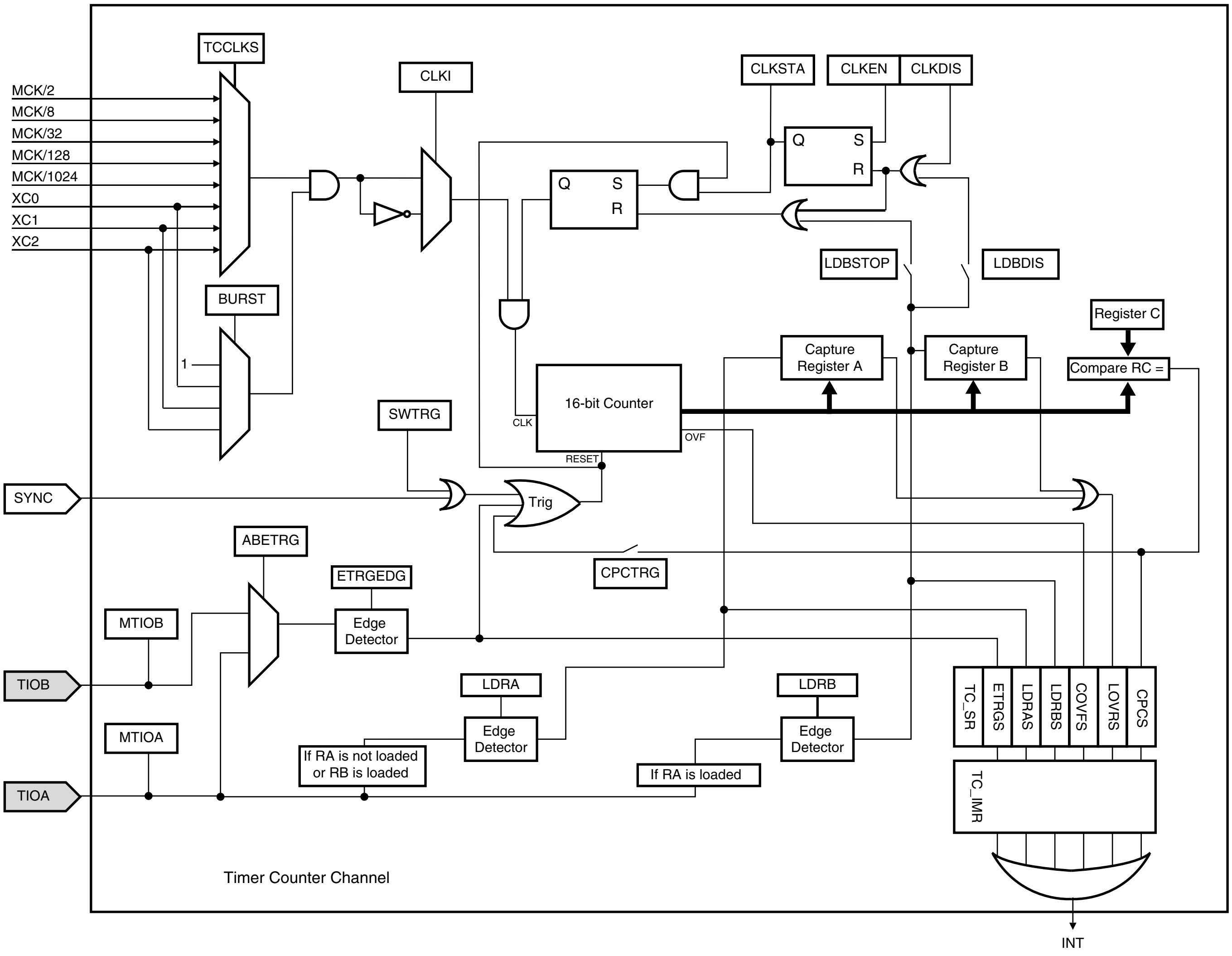
- LDRBS: Load RB Status

  RB has been loaded at least once without any read, since the last read of the status

- ETRGS: External Trigger Status An external trigger on TIOA or TIOB has been detected since the last read of the status

Note: All the status bits are set when the corresponding event occurs and they are automatically cleared when the Status Register is read.

Figure 19-4. Capture Mode



19.5 Waveform Operating Mode

This mode is entered by setting the WAVE parameter in TC_CMR (Channel Mode Register).

Waveform Operating Mode allows the TC Channel to generate 1 or 2 PWM signals with the same frequency and independently programmable duty cycles, or to generate different types of one-shot or repetitive pulses.

In this mode, TIOA is configured as output and TIOB is defined as output if it is not used as an external event (EEVT parameter in TC_CMR).

Figure 19-5 shows the configuration of the TC Channel when programmed in Waveform Operating Mode.

19.5.1 Compare Register A, B and C (RA, RB, and RC)

In Waveform Operating Mode, RA, RB and RC are all used as compare registers.

RA Compare is used to control the TIOA output. RB Compare is used to control the TIOB (if configured as output). RC Compare can be programmed to control TIOA and/or TIOB outputs.

RC Compare can also stop the counter clock (CPCSTOP = 1 in TC_CMR) and/or disable the counter clock (CPCDIS = 1 in TC_CMR).

As in Capture Mode, RC Compare can also generate a trigger if CPCTRG = 1. A trigger resets the counter so RC can control the period of PWM waveforms.

External Event/Trigger Conditions

An external event can be programmed to be detected on one of the clock sources (XC0, XC1, XC2) or TIOB. The external event selected can then be used as a trigger.

The parameter EEVT in TC_CMR selects the external trigger. The parameter EEVTEDG defines the trigger edge for each of the possible external triggers (rising, falling or both). If EEVTEDG is cleared (none), no external event is defined.

If TIOB is defined as an external event signal (EEVT = 0), TIOB is no longer used as output and the TC channel can only generate a waveform on TIOA.

When an external event is defined, it can be used as a trigger by setting bit ENETRG in TC_CMR.

As in Capture Mode, the SYNC signal, the software trigger and the RC compare trigger are also available as triggers.

19.5.2 Output Controller

The output controller defines the output level changes on TIOA and TIOB following an event. TIOB control is used only if TIOB is defined as output (not as an external event).

The following events control TIOA and TIOB: software trigger, external event and RC compare. RA compare controls TIOA and RB compare controls TIOB. Each of these events can be programmed to set, clear or toggle the output as defined in the corresponding parameter in TC_CMR.

The tables below show which parameter in TC_CMR is used to define the effect of each event.

Parameter	TIOA Event
ASWTRG	Software Trigger
AEEVT	External Event
ACPC	RC Compare
ACPA	RA Compare

Parameter	TIOB Event
BSWTRG	Software Trigger
BEEVT	External Event
BCPC	RC Compare
BCPB	RB Compare

If two or more events occur at the same time, the priority level is defined as follows:

1. Software Trigger
2. External Event
3. RC Compare
4. RA or RB Compare

19.5.3 Status

The following bits in the status register are significant in Waveform Mode:

- CPAS: RA Compare Status

  There has been a RA Compare match at least once since the last read of the status

- CPBS: RB Compare Status

  There has been a RB Compare match at least once since the last read of the status

- CPCS: RC Compare Status

  There has been a RC Compare match at least once since the last read of the status

- COVFS: Counter Overflow

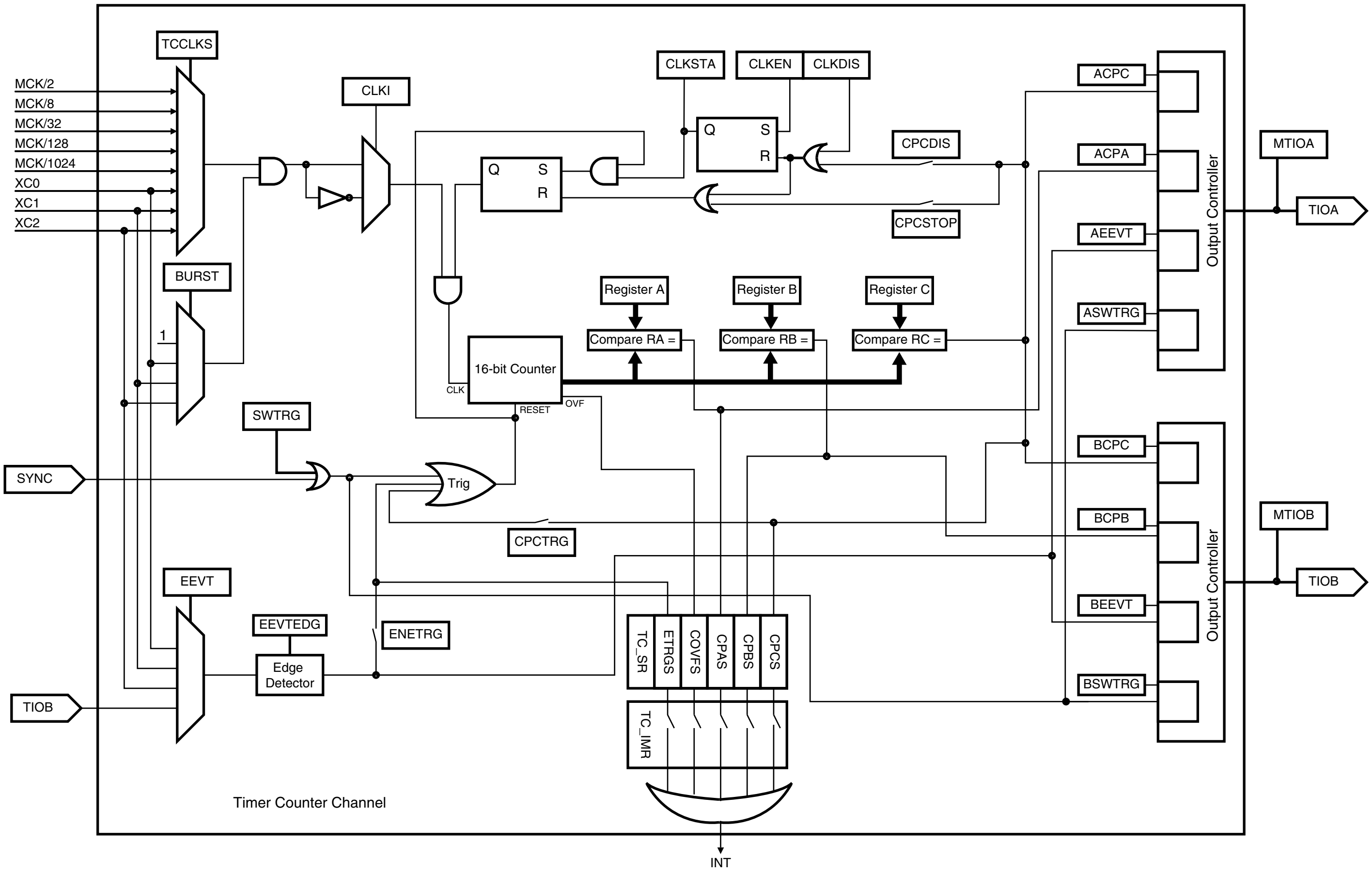
  Counter has attempted to count past $FFFF since the last read of the status

- ETRGS: External Trigger

  External trigger has been detected since the last read of the status

Note: All the status bits are set when the corresponding event occurs and they are automatically cleared when the Status Register is read.

Figure 19-5. Waveform Mode



19.6 TC User Interface

TC Base Address: 0xFFFE0000 (Code Label `TC_BASE`)

Table 19-2. TC Global Memory Map

Offset	Channel/Register	Name	Access	Reset State
0x00	TC Channel 0	See Table 19-3		
0x40	TC Channel 1	See Table 19-3		
0x80	TC Channel 2	See Table 19-3		
0xC0	TC Block Control Register	TC_BCR	Write-only	–
0xC4	TC Block Mode Register	TC_BMR	Read/Write	0

TC_BCR (Block Control Register) and TC_BMR (Block Mode Register) control the TC block. TC Channels are controlled by the registers listed in Table 19-3. The offset of each of the Channel registers in Table 19-3 is in relation to the offset of the corresponding channel as mentioned in Table 19-2.

Table 19-3. TC Channel Memory Map

Offset	Register	Name	Access	Reset State
0x00	Channel Control Register	TC_CCR	Write-only	–
0x04	Channel Mode Register	TC_CMR	Read/Write	0
0x08	Reserved			–
0x0C	Reserved			–
0x10	Counter Value	TC_CV	Read/Write	0
0x14	Register A	TC_RA	Read/Write[1]	0
0x18	Register B	TC_RB	Read/Write[1]	0
0x1C	Register C	TC_RC	Read/Write	0
0x20	Status Register	TC_SR	Read-only	0
0x24	Interrupt Enable Register	TC_IER	Write-only	–
0x28	Interrupt Disable Register	TC_IDR	Write-only	–
0x2C	Interrupt Mask Register	TC_IMR	Read-only	0

Note: 1. Read-only if WAVE = 0

19.6.1 TC Block Control Register

Register Name:TC_BCR

Access Type:Write-only

Offset: 0xC0

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**
–	–	–	–	–	–	–	SYNC

- **SYNC: Synchro Command**

0 = No effect.

1 = Asserts the SYNC signal which generates a software trigger simultaneously for each of the channels.

19.6.2 TC Block Mode Register

Register Name:TC_BMR

Access Type:Read/Write

Reset Value: 0

Offset: 0xC4

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
–	–	TC2XC2S		TC1XC1S		TC0XC0S	

- **TC0XC0S: External Clock Signal 0 Selection**

TC0XC0S		Signal Connected to XC0
0	0	TCLK0
0	1	None
1	0	TIOA1
1	1	TIOA2

- **TC1XC1S: External Clock Signal 1 Selection**

TC1XC1S		Signal Connected to XC1
0	0	TCLK1
0	1	None
1	0	TIOA0
1	1	TIOA2

- **TC2XC2S: External Clock Signal 2 Selection**

TC2XC2S		Signal Connected to XC2
0	0	TCLK2
0	1	None
1	0	TIOA0
1	1	TIOA1

19.6.3 TC Channel Control Register

Register Name:TC_CCR

Access Type:Write-only

Offset: 0x00

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**
–	–	–	–	–	SWTRG	CLKDIS	CLKEN

- **CLKEN: Counter Clock Enable Command (Code Label `TC_CLKEN`)**

0 = No effect.

1 = Enables the clock if CLKDIS is not 1.

- **CLKDIS: Counter Clock Disable Command (Code Label `TC_CLKDIS`)**

0 = No effect.

1 = Disables the clock.

- **SWTRG: Software Trigger Command (Code Label `TC_SWTRG`)**

0 = No effect.

1 = A software trigger is performed: the counter is reset and clock is started.

19.6.4 TC Channel Mode Register: Capture Mode

Register Name:TC_CMR

Access Type:Read/Write

Reset Value: 0

Offset: 0x04

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	LDRB		LDRA	

15	14	13	12	11	10	9	8
WAVE = 0	CPCTRG	–	–	–	ABETRG	ETRGEDG	

7	6	5	4	3	2	1	0
LDBDIS	LDBSTOP	BURST		CLKI	TCCLKS		

- **TCCLKS: Clock Selection**

				Code Label
TCCLKS			**Clock Selected**	**TC_CLKS**
0	0	0	MCK/2	TC_CLKS_MCK2
0	0	1	MCK/8	TC_CLKS_MCK8
0	1	0	MCK/32	TC_CLKS_MCK32
0	1	1	MCK/128	TC_CLKS_MCK128
1	0	0	MCK/1024	TC_CLKS_MCK1024
1	0	1	XC0	TC_CLKS_XC0
1	1	0	XC1	TC_CLKS_XC1
1	1	1	XC2	TC_CLKS_XC2

- **CLKI: Clock Invert (Code Label `TC_CLKI`)**

0 = Counter is incremented on rising edge of the clock.

1 = Counter is incremented on falling edge of the clock.

- **BURST: Burst Signal Selection**

			Code Label
BURST		**Selected BURST**	**TC_BURST**
0	0	The clock is not gated by an external signal	TC_BURST_NONE
0	1	XC0 is ANDed with the selected clock	TC_BURST_XC0
1	0	XC1 is ANDed with the selected clock	TC_BURST_XC1
1	1	XC2 is ANDed with the selected clock	TC_BURST_XC2

- **LDBSTOP: Counter Clock Stopped with RB Loading (Code Label `TC_LDBSTOP`)**

0 = Counter clock is not stopped when RB loading occurs.

1 = Counter clock is stopped when RB loading occurs.

- **LDBDIS: Counter Clock Disable with RB Loading (Code Label `TC_LDBDIS`)**

0 = Counter clock is not disabled when RB loading occurs.

1 = Counter clock is disabled when RB loading occurs.

- **ETRGEDG: External Trigger Edge Selection**

ETRGEDG		Edge	Code Label TC_ETRGEDG
0	0	None	TC_ETRGEDG_EDGE_NONE
0	1	Rising Edge	TC_ETRGEDG_RISING_EDGE
1	0	Falling Edge	TC_ETRGEDG_FALLING_EDGE
1	1	Each Edge	TC_ETRGEDG_BOTH_EDGE

- **ABETRG: TIOA or TIOB External Trigger Selection**

ABETRG	Selected ABETRG	Code Label TC_ABETRG
0	TIOB is used as an external trigger.	TC_ABETRG_TIOB
1	TIOA is used as an external trigger.	TC_ABETRG_TIOA

- **CPCTRG: RC Compare Trigger Enable (Code Label `TC_CPCTRG`)**

0 = RC Compare has no effect on the counter and its clock.

1 = RC Compare resets the counter and starts the counter clock.

- **WAVE = 0 (Code Label `TC_WAVE`)**

0 = Capture Mode is enabled.

1 = Capture Mode is disabled (Waveform Mode is enabled).

- **LDRA: RA Loading Selection**

LDRA		Edge	Code Label TC_LDRA
0	0	None	TC_LDRA_EDGE_NONE
0	1	Rising edge of TIOA	TC_LDRA_RISING_EDGE
1	0	Falling edge of TIOA	TC_LDRA_FALLING_EDGE
1	1	Each edge of TIOA	TC_LDRA_BOTH_EDGE

- **LDRB: RB Loading Selection**

LDRB		Edge	Code Label TC_LDRB
0	0	None	TC_LDRB_EDGE_NONE
0	1	Rising edge of TIOA	TC_LDRB_RISING_EDGE
1	0	Falling edge of TIOA	TC_LDRB_FALLING_EDGE
1	1	Each edge of TIOA	TC_LDRB_BOTH_EDGE

19.6.5 TC Channel Mode Register: Waveform Mode

Register Name:TC_CMR

Access Type:Read/Write

Reset Value: 0

Offset: 0x04

31	30	29	28	27	26	25	24
BSWTRG		BEEVT		BCPC		BCPB	

23	22	21	20	19	18	17	16
ASWTRG		AEEVT		ACPC		ACPA	

15	14	13	12	11	10	9	8
WAVE = 1	CPCTRG	–	ENETRG	EEVT		EEVTEDG	

7	6	5	4	3	2	1	0
CPCDIS	CPCSTOP	BURST		CLKI	TCCLKS		

- **TCCLKS: Clock Selection**

TCCLKS			Clock Selected	Code Label
				TC_CLKS
0	0	0	MCK/2	TC_CLKS_MCK2
0	0	1	MCK/8	TC_CLKS_MCK8
0	1	0	MCK/32	TC_CLKS_MCK32
0	1	1	MCK/128	TC_CLKS_MCK128
1	0	0	MCK/1024	TC_CLKS_MCK1024
1	0	1	XC0	TC_CLKS_XC0
1	1	0	XC1	TC_CLKS_XC1
1	1	1	XC2	TC_CLKS_XC2

- **CLKI: Clock Invert (Code Label `TC_CLKI`)**

0 = Counter is incremented on rising edge of the clock.

1 = Counter is incremented on falling edge of the clock.

- **BURST: Burst Signal Selection**

BURST		Selected BURST	Code Label
			TC_BURST
0	0	The clock is not gated by an external signal.	TC_BURST_NONE
0	1	XC0 is ANDed with the selected clock.	TC_BURST_XC0
1	0	XC1 is ANDed with the selected clock.	TC_BURST_XC1
1	1	XC2 is ANDed with the selected clock.	TC_BURST_XC2

- **CPCSTOP: Counter Clock Stopped with RC Compare (Code Label `TC_CPCSTOP`)**

0 = Counter clock is not stopped when counter reaches RC.

1 = Counter clock is stopped when counter reaches RC.

- **CPCDIS: Counter Clock Disable with RC Compare (Code Label `TC_CPCDIS`)**

0 = Counter clock is not disabled when counter reaches RC.

1 = Counter clock is disabled when counter reaches RC.

- **EEVTEDG: External Event Edge Selection**

EEVTEDG		Edge	Code Label TC_EEVTEDG
0	0	None	TC_EEVTEDG_EDGE_NONE
0	1	Rising edge	TC_EEVTEDG_RISING_EDGE
1	0	Falling edge	TC_EEVTEDG_FALLING_EDGE
1	1	Each edge	TC_EEVTEDG_BOTH_EDGE

- **EEVT: External Event Selection**

EEVT		Signal Selected as External Event	TIOB Direction	Code Label TC_EEVT
0	0	TIOB	Input[1]	TC_EEVT_TIOB
0	1	XC0	Output	TC_EEVT_XC0
1	0	XC1	Output	TC_EEVT_XC1
1	1	XC2	Output	TC_EEVT_XC2

Note: If TIOB is chosen as the external event signal, it is configured as an input and no longer generates waveforms.

- **ENETRG: External Event Trigger Enable (Code Label `TC_ENETRG`)**

0 = The external event has no effect on the counter and its clock. In this case, the selected external event only controls the TIOA output.

1 = The external event resets the counter and starts the counter clock.

- **CPCTRG: RC Compare Trigger Enable (Code Label `TC_CPCTRG`)**

0 = RC Compare has no effect on the counter and its clock.

1 = RC Compare resets the counter and starts the counter clock.

- **WAVE = 1 (Code Label `TC_WAVE`)**

0 = Waveform Mode is disabled (Capture Mode is enabled).

1 = Waveform Mode is enabled.

- **ACPA: RA Compare Effect on TIOA**

			Code Label
ACPA		Effect	TC_ACPA
0	0	None	TC_ACPA_OUTPUT_NONE
0	1	Set	TC_ACPA_SET_OUTPUT
1	0	Clear	TC_ACPA_CLEAR_OUTPUT
1	1	Toggle	TC_ACPA_TOGGLE_OUTPUT

- **ACPC: RC Compare Effect on TIOA**

			Code Label
ACPC		Effect	TC_ACPC
0	0	None	TC_ACPC_OUTPUT_NONE
0	1	Set	TC_ACPC_SET_OUTPUT
1	0	Clear	TC_ACPC_CLEAR_OUTPUT
1	1	Toggle	TC_ACPC_TOGGLE_OUTPUT

- **AEEVT: External Event Effect on TIOA**

			Code Label
AEEVT		Effect	TC_AEEVT
0	0	None	TC_AEEVT_OUTPUT_NONE
0	1	Set	TC_AEEVT_SET_OUTPUT
1	0	Clear	TC_AEEVT_CLEAR_OUTPUT
1	1	Toggle	TC_AEEVT_TOGGLE_OUTPUT

- **ASWTRG: Software Trigger Effect on TIOA**

			Code Label
ASWTRG		Effect	TC_ASWTRG
0	0	None	TC_ASWTRG_OUTPUT_NONE
0	1	Set	TC_ASWTRG_SET_OUTPUT
1	0	Clear	TC_ASWTRG_CLEAR_OUTPUT
1	1	Toggle	TC_ASWTRG_TOGGLE_OUTPUT

- **BCPB: RB Compare Effect on TIOB**

			Code Label
BCPB		**Effect**	**TC_BCPB**
0	0	None	TC_BCPB_OUTPUT_NONE
0	1	Set	TC_BCPB_SET_OUTPUT
1	0	Clear	TC_BCPB_CLEAR_OUTPUT
1	1	Toggle	TC_BCPB_TOGGLE_OUTPUT

- **BCPC: RC Compare Effect on TIOB**

			Code Label
BCPC		**Effect**	**TC_BCPC**
0	0	None	TC_BCPC_OUTPUT_NONE
0	1	Set	TC_BCPC_SET_OUTPUT
1	0	Clear	TC_BCPC_CLEAR_OUTPUT
1	1	Toggle	TC_BCPC_TOGGLE_OUTPUT

- **BEEVT: External Event Effect on TIOB**

			Code Label
BEEVT		**Effect**	**TC_BEEVT**
0	0	None	TC_BEEVT_OUTPUT_NONE
0	1	Set	TC_BEEVT_SET_OUTPUT
1	0	Clear	TC_BEEVT_CLEAR_OUTPUT
1	1	Toggle	TC_BEEVT_TOGGLE_OUTPUT

- **BSWTRG: Software Trigger Effect on TIOB**

			Code Label
BSWTRG		**Effect**	**TC_BSWTRG**
0	0	None	TC_BSWTRG_OUTPUT_NONE
0	1	Set	TC_BSWTRG_SET_OUTPUT
1	0	Clear	TC_BSWTRG_CLEAR_OUTPUT
1	1	Toggle	TC_BSWTRG_TOGGLE_OUTPUT

19.6.6 TC Counter Value Register

Register Name:TC_CVR

Access Type:Read-only

Reset Value: 0

Offset: 0x10

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
CV							
7	6	5	4	3	2	1	0
CV							

- **CV: Counter Value (Code Label `TC_CV`)**

CV contains the counter value in real-time.

19.6.7 TC Register A

Register Name:TC_RA

Access Type:Read-only if WAVE = 0, Read/Write if WAVE = 1

Reset Value: 0

Offset: 0x14

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
RA							
7	6	5	4	3	2	1	0
RA							

- **RA: Register A (Code Label `TC_RA`)**

RA contains the Register A value in real-time.

19.6.8 TC Register B

Register Name:TC_RB

Access Type:Read-only if WAVE = 0, Read/Write if WAVE = 1

Reset Value: 0

Offset: 0x18

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RB							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RB							

- **RB: Register B (Code Label `TC_RB`)**

RB contains the Register B value in real-time.

19.6.9 TC Register C

Register Name:TC_RC

Access Type:Read/Write

Reset Value: 0

Offset: 0x1C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RC							

- **RC: Register C (Code Label `TC_RC`)**

RC contains the Register C value in real-time.

19.6.10 TC Status Register

Register Name:TC_SR

Access Type:Read-only

Reset Value: 0

Offset: 0x20

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	MTIOB	MTIOA	CLKSTA
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

- **COVFS: Counter Overflow Status (Code Label TC_COVFS)**

0 = No counter overflow has occurred since the last read of the Status Register.

1 = A counter overflow has occurred since the last read of the Status Register.

- **LOVRS: Load Overrun Status (Code Label TC_LOVRS)**

0 = Load overrun has not occurred since the last read of the Status Register or WAVE = 1.

1 = RA or RB have been loaded at least twice without any read of the corresponding register since the last read of the Status Register, if WAVE = 0.

- **CPAS: RA Compare Status (Code Label TC_CPAS)**

0 = RA Compare has not occurred since the last read of the Status Register or WAVE = 0.

1 = RA Compare has occurred since the last read of the Status Register, if WAVE = 1.

- **CPBS: RB Compare Status (Code Label TC_CPBS)**

0 = RB Compare has not occurred since the last read of the Status Register or WAVE = 0.

1 = RB Compare has occurred since the last read of the Status Register, if WAVE = 1.

- **CPCS: RC Compare Status (Code Label TC_CPCS)**

0 = RC Compare has not occurred since the last read of the Status Register.

1 = RC Compare has occurred since the last read of the Status Register.

- **LDRAS: RA Loading Status (Code Label TC_LDRAS)**

0 = RA Load has not occurred since the last read of the Status Register or WAVE = 1.

1 = RA Load has occurred since the last read of the Status Register, if WAVE = 0.

- **LDRBS: RB Loading Status (Code Label TC_LDRBS)**

0 = RB Load has not occurred since the last read of the Status Register or WAVE = 1.

1 = RB Load has occurred since the last read of the Status Register, if WAVE = 0.

- **ETRGS: External Trigger Status (Code Label TC_ETRGS)**

0 = External trigger has not occurred since the last read of the Status Register.

1 = External trigger has occurred since the last read of the Status Register.

- **CLKSTA: Clock Enabling Status (Code Label `TC_CLKSTA`)**

0 = Clock is disabled.

1 = Clock is enabled.

- **MTIOA: TIOA Mirror (Code Label `TC_MTIOA`)**

0 = TIOA is low. If WAVE = 0, this means that TIOA pin is low. If WAVE = 1, this means that TIOA is driven low.

1 = TIOA is high. If WAVE = 0, this means that TIOA pin is high. If WAVE = 1, this means that TIOA is driven high.

- **MTIOB: TIOB Mirror (Code Label `TC_MTIOB`)**

0 = TIOB is low. If WAVE = 0, this means that TIOB pin is low. If WAVE = 1, this means that TIOB is driven low.

1 = TIOB is high. If WAVE = 0, this means that TIOB pin is high. If WAVE = 1, this means that TIOB is driven high.

19.6.11 TC Interrupt Enable Register

Register Name:TC_IER

Access Type:Write-only

Offset: 0x24

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
ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

- **COVFS: Counter Overflow (Code Label `TC_COVFS`)**

0 = No effect.

1 = Enables the Counter Overflow Interrupt.

- **LOVRS: Load Overrun (Code Label `TC_LOVRS`)**

0 = No effect.

1: Enables the Load Overrun Interrupt.

- **CPAS: RA Compare (Code Label `TC_CPAS`)**

0 = No effect.

1 = Enables the RA Compare Interrupt.

- **CPBS: RB Compare (Code Label `TC_CPBS`)**

0 = No effect.

1 = Enables the RB Compare Interrupt.

- **CPCS: RC Compare (Code Label `TC_CPCS`)**

0 = No effect.

1 = Enables the RC Compare Interrupt.

- **LDRAS: RA Loading (Code Label `TC_LDRAS`)**

0 = No effect.

1 = Enables the RA Load Interrupt.

- **LDRBS: RB Loading (Code Label `TC_LDRBS`)**

0 = No effect.

1 = Enables the RB Load Interrupt.

- **ETRGS: External Trigger (Code Label `TC_ETRGS`)**

0 = No effect.

1 = Enables the External Trigger Interrupt.

19.6.12 TC Interrupt Disable Register

Register Name:TC_IDR

Access Type:Write-only

Offset: 0x28

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**
ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

- **COVFS: Counter Overflow (Code Label `TC_COVFS`)**

0 = No effect.

1 = Disables the Counter Overflow Interrupt.

- **LOVRS: Load Overrun (Code Label `TC_LOVRS`)**

0 = No effect.

1 = Disables the Load Overrun Interrupt (if WAVE = 0).

- **CPAS: RA Compare (Code Label `TC_CPAS`)**

0 = No effect.

1 = Disables the RA Compare Interrupt (if WAVE = 1).

- **CPBS: RB Compare (Code Label `TC_CPBS`)**

0 = No effect.

1 = Disables the RB Compare Interrupt (if WAVE = 1).

- **CPCS: RC Compare (Code Label `TC_CPCS`)**

0 = No effect.

1 = Disables the RC Compare Interrupt.

- **LDRAS: RA Loading (Code Label `TC_LDRAS`)**

0 = No effect.

1 = Disables the RA Load Interrupt (if WAVE = 0).

- **LDRBS: RB Loading (Code Label `TC_LDRBS`)**

0 = No effect.

1 = Disables the RB Load Interrupt (if WAVE = 0).

- **ETRGS: External Trigger (Code Label `TC_ETRGS`)**

0 = No effect.

1 = Disables the External Trigger Interrupt.

19.6.13 TC Interrupt Mask Register

Register Name:TC_IMR

Access Type:Read-only

Reset Value: 0

Offset: 0x2C

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**
ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

- **COVFS: Counter Overflow (Code Label `TC_COVFS`)**

0 = The Counter Overflow Interrupt is disabled.

1 = The Counter Overflow Interrupt is enabled.

- **LOVRS: Load Overrun (Code Label `TC_LOVRS`)**

0 = The Load Overrun Interrupt is disabled.

1 = The Load Overrun Interrupt is enabled.

- **CPAS: RA Compare (Code Label `TC_CPAS`)**

0 = The RA Compare Interrupt is disabled.

1 = The RA Compare Interrupt is enabled.

- **CPBS: RB Compare (Code Label `TC_CPBS`)**

0 = The RB Compare Interrupt is disabled.

1 = The RB Compare Interrupt is enabled.

- **CPCS: RC Compare (Code Label `TC_CPCS`)**

0 = The RC Compare Interrupt is disabled.

1 = The RC Compare Interrupt is enabled.

- **LDRAS: RA Loading (Code Label `TC_LDRAS`)**

0 = The Load RA Interrupt is disabled.

1 = The Load RA Interrupt is enabled.

- **LDRBS: RB Loading (Code Label `TC_LDRBS`)**

0 = The Load RB Interrupt is disabled.

1 = The Load RB Interrupt is enabled.

- **ETRGS: External Trigger (Code Label `TC_ETRGS`)**

0 = The External Trigger Interrupt is disabled.

1 = The External Trigger Interrupt is enabled.

20. AT91FR40162SB Electrical Characteristics

20.1 Absolute Maximum Ratings

Table 20-1. Absolute Maximum Ratings*

Operating Temperature (Industrial)	-40· C to + 85· C
Storage Temperature	-60· C to + 150· C
Voltage on Any Input Pin with Respect to Ground	-0.3V to max of V_{DDIO} + 0.3V and 3.6V
Maximum Operating Voltage (V_{DDIO})	3.6V
Maximum Operating Voltage (V_{DDCORE})	1.95V

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

20.2 AT91FR40162SB DC Characteristics

The following characteristics are applicable to the Operating Temperature range: T_A = -40· C to +85· C, unless otherwise specified and are certified for a Junction Temperature up to 100· C.

Table 20-2. DC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{DDIO}	DC Supply I/Os		2.7		3.6	V
V_{DDCORE}	DC Supply Core		1.65		1.95	V
V_{IL}	Input Low Voltage		-0.3		0.8	V
V_{IH}	Input High Voltage		2.0		V_{DDIO} + 0.3	V
V_{OL}	Output Low Voltage	Pin Group 1[2]: I_{OL} = 16 mA[1]			0.4	V
		Pin Group 2[3]: I_{OL} = 8 mA[1]			0.4	V
		Pin Group 3[4]: I_{OL} = 2 mA[1]			0.4	V
		All Output Pins: I_{OL} = 0 mA[1]			0.2	V
V_{OH}	Output High Voltage	Pin Group 1[2]: I_{OH} = 16 mA[1]	V_{DDIO} - 0.4			V
		Pin Group 2[3]: I_{OH} = 8 mA[1]	V_{DDIO} - 0.4			
		Pin Group 3[4]: I_{OH} = 2 mA[1]	V_{DDIO} - 0.4			
		All Output Pins: I_{OH} = 0 mA[1]	V_{DDIO} - 0.2			
I_{LEAK}	Input Leakage Current				10	µA
I_{PULL}	Input Pull-up Current	V_{DDIO} = 3.6V, V_{IN} = 0V			400	µA
I_{OUT}	Output Current	Pin Group 1[2]			16	mA
		Pin Group 2[3]:			8	mA
		Pin Group 3[4]:			2	mA
C_{IN}	Input Capacitance	121-BGA Package			5.3	pF
I_{SC}	Static Current	VDDIO= 3.6V, V_{DDCORE} = 1.95V, MCKI = 0Hz All Inputs Driven TMS, TCK, TDI, NRST = 1 T_A = 25· C			400	µA
		T_A = 85· C			2.3	mA

Notes:
1. I_{OL} = Output Current at low level. I_{OH}= Output Current at high level.
2. Pin Group 1 = NUB/NWR1, NWE/NWR0, NOE/NRD1
3. Pin Group 2 = D0-D15, A0/NLB, A1-A19, P28/A20/CS7, P29/A21/CS6, P30/A22/CS5, P31/A23/CS4, NCS0, NCS1, P26/NCS2, P27/NCS3
4. Pin Group 3 = All Others

20.3 Flash DC Characteristics

Table 20-3. Flash DC Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{LI}	Input Load Current	V_{IN} = 0V to V_{CC}			2	µA
I_{LO}	Output Leakage Current	$V_{I/O}$ = 0V to V_{CC}			2	µA
I_{SB}	V_{CC} Standby Current CMOS	$\overline{CE}$ = V_{CC} - 0.3V to V_{CC}		15	25	µA
I_{CC} [1]	V_{CC} Active Read Current	f = 5 MHz; I_{OUT} = 0 mA		10	15	mA
I_{CC1}	V_{CC} Programming Current				25	mA
V_{IL}	Input Low Voltage				0.6	V
V_{IH}	Input High Voltage		2.0			V

Note: 1. In the erase mode, I_{CC} is 45 mA.

20.4 Flash Operating Modes

Table 20-4. Flash Operating Modes

Mode	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	$\overline{RESET}$	Ai	I/O
Read	V_{IL}	V_{IL}	V_{IH}	V_{IH}	Ai	D_{OUT}
Program/Erase	V_{IL}	V_{IH}	V_{IL}	V_{IH}	Ai	D_{IN}
Standby/Program Inhibit	V_{IH}	X[1]	X	V_{IH}	X	High-Z
Program Inhibit	X	X	V_{IH}	V_{IH}		
	X	V_{IL}	X	V_{IH}		
	X	X	X	V_{IH}		
Output Disable	X	V_{IH}	X	V_{IH}		High-Z
Reset	X	X	X	V_{IL}	X	High-Z
Product Identification Software				V_{IH}	A0 = V_{IL}, A1 - A19 = V_{IL}	Manufacturer Code[2]
					A0 = V_{IH}, A1 - A19 = V_{IL}	Device Code[2]

Notes: 1. X can be V_{IL} or V_{IH}.

2. Manufacturer Code: 001FH, Device Code: 01C0H

20.5 Power Consumption

The values in the following tables are values measured in the typical operating conditions (i.e., V_{DDIO} = 3.3V, V_{DDCORE} = 1.8V, T_A = 25· C) on the AT91EB40A Evaluation Board and are given as demonstrative values.

Table 20-5. Power Consumption on VDDCORE

Mode	Conditions	Consumption	Unit
Reset		0.02	mW/MHz
Normal	Fetch in ARM mode from internal SRAM All peripheral clocks activated	0.83	
	Fetch in ARM mode from internal SRAM All peripheral clocks deactivated	0.73	
	Fetch in ARM mode from external SRAM[1] All peripheral clocks deactivated	0.20	
	Fetch in Thumb mode from external SRAM[1] All peripheral clocks deactivated	0.24	
Idle	All peripheral clocks activated	0.16	
	All peripheral clocks deactivated	0.06	

Note: 1. With two Wait States.

Table 20-6. Power Consumption per Peripheral on VDDCORE

Peripheral	Consumption	Unit
PIO Controller	15.3	µW/MHz
Timer/Counter Channel	15.0	
Timer/Counter Block (3 Channels)	36.3	
USART	27.8	

20.6 Clock Waveforms

Table 20-7. Master Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Units
$1/(t_{CP})$	Oscillator Frequency			82.1	MHz
t_{CP}	Oscillator Period		12.2		ns
t_{CH}	High Half-period		5.0		ns
t_{CL}	Low Half-period		5.5		ns

Table 20-8. Clock Propagation Times

Symbol	Parameter	Conditions	Min	Max	Units
t_{CDLH}	Rising Edge Propagation Time	C_{MCKO} = 0 pF	4.4	6.6	ns
		C_{MCKO} derating	0.199	0.295	ns/pF
t_{CDHL}	Falling Edge Propagation Time	C_{MCKO} = 0 pF	4.5	6.7	ns
		C_{MCKO} derating	0.153	0.228	ns/pF

Figure 20-1. Clock Waveform

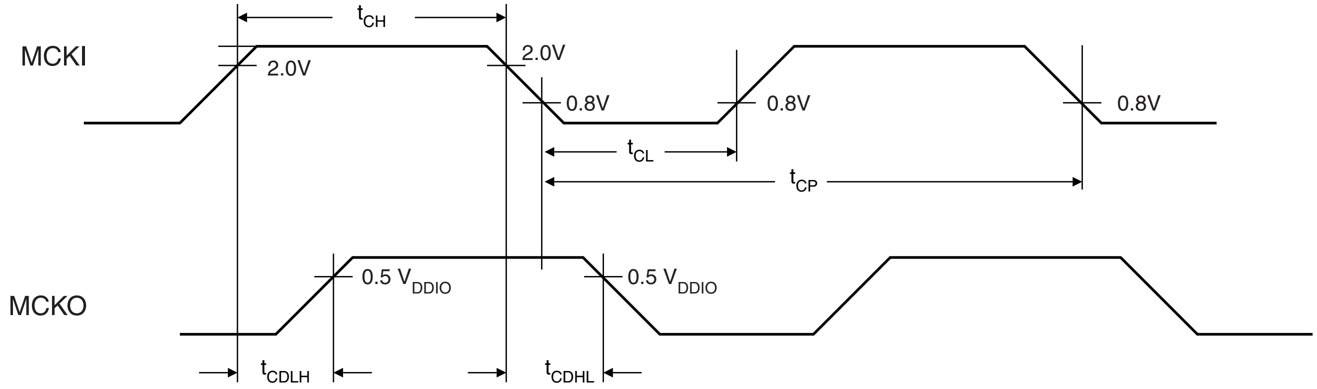
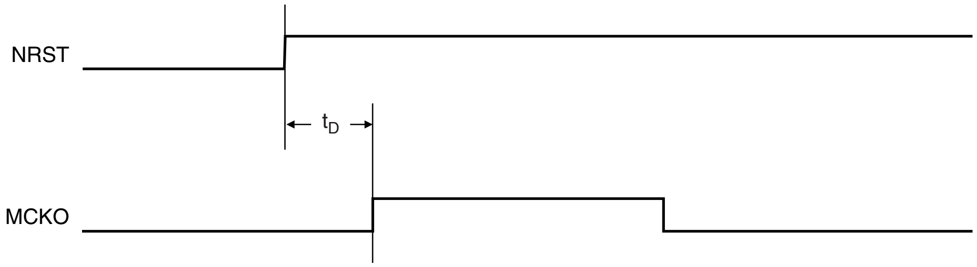


Table 20-9. NRST to MCKO

Symbol	Parameter	Min	Max	Units
t_D	NRST Rising Edge to MCKO Valid Time	$3(t_{CP}/2)$	$7(t_{CP}/2)$	ns

Figure 20-2. MCKO Relative to NRST



21. AC Characteristics

21.1 Applicable Conditions and Derating Data

21.1.1 Conditions and Timing Results

The delays are given as typical values in the following conditions:

- $V_{DDIO} = 3.0V$
- $V_{DDCORE} = 1.8V$
- Ambient Temperature = 25· C
- Load Capacitance = 0 pF
- The output level change detection is *0.5 x* V_{DDIO}
- The input level is *0.8V* for a low-level detection and is *2.0V* for a high level detection.

The minimum and maximum values given in the AC characteristic tables of this datasheet take into account the process variation and the design.

In order to obtain the timing for other conditions, the following equation should be used:

$$t = \delta_{T^\circ} \times \left((\delta_{VDDCORE} \times t_{DATASHEET}) + \left(\delta_{VDDIO} \times \sum (C_{SIGNAL} \times \delta_{CSIGNAL}) \right) \right)$$

Where:

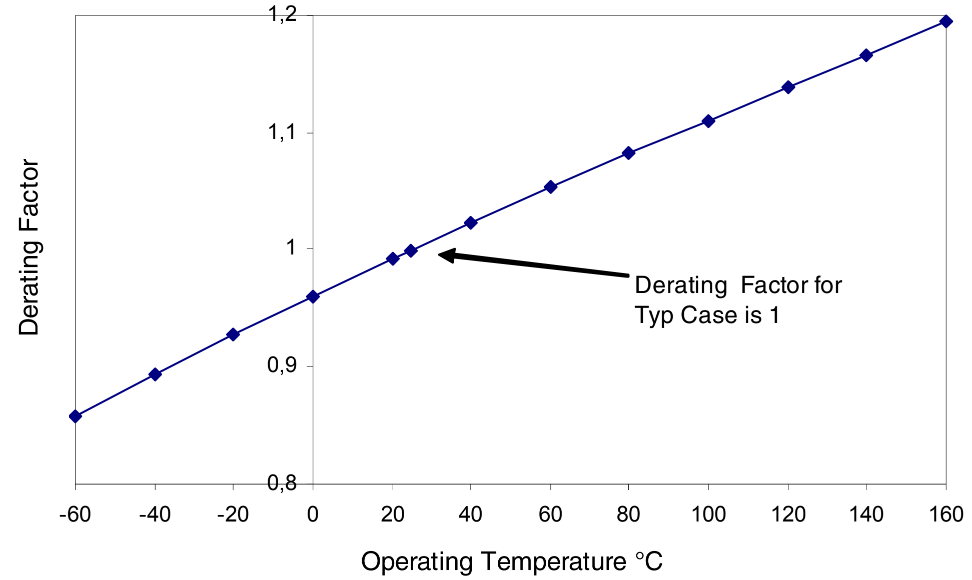
- $\delta_{T^\cdot}$ is the derating factor in temperature given in Figure 21-1 on page 184.
- $\delta_{VDDCORE}$ is the derating factor for the Core Power Supply given in Figure 21-2 on page 184.
- $t_{DATASHEET}$ is the minimum or maximum timing value given in this datasheet for a load capacitance of 0 pF.
- δ_{VDDIO} is the derating factor for the I/O Power Supply given in Figure 21-3 on page 185.
- C_{SIGNAL} is the capacitance load on the considered output pin.[1]
- $\delta_{CSIGNAL}$ is the load derating factor depending on the capacitance load on the related output pins given in Min and Max values in this datasheet.

The input delays are given as typical values.

Note: 1. The user must take into account the package capacitance load contribution (C_{IN}) described in Table 20-2 on page 178.

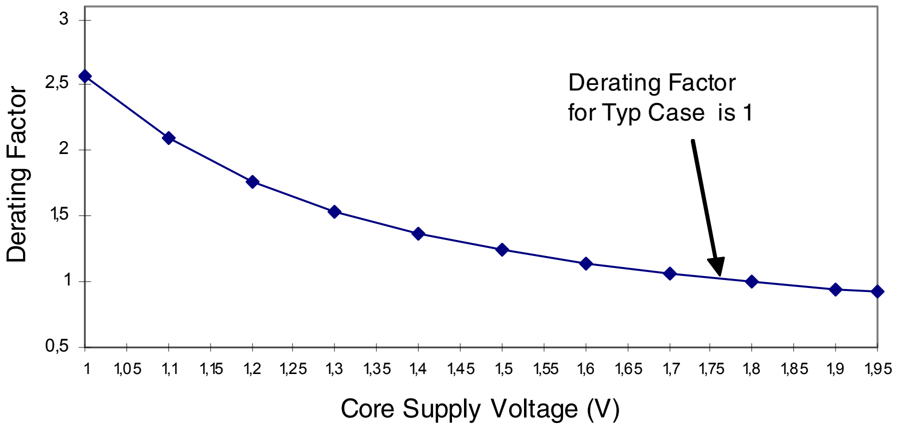
21.1.2 Temperature Derating Factor

Figure 21-1. Derating Curve for Different Operating Temperatures



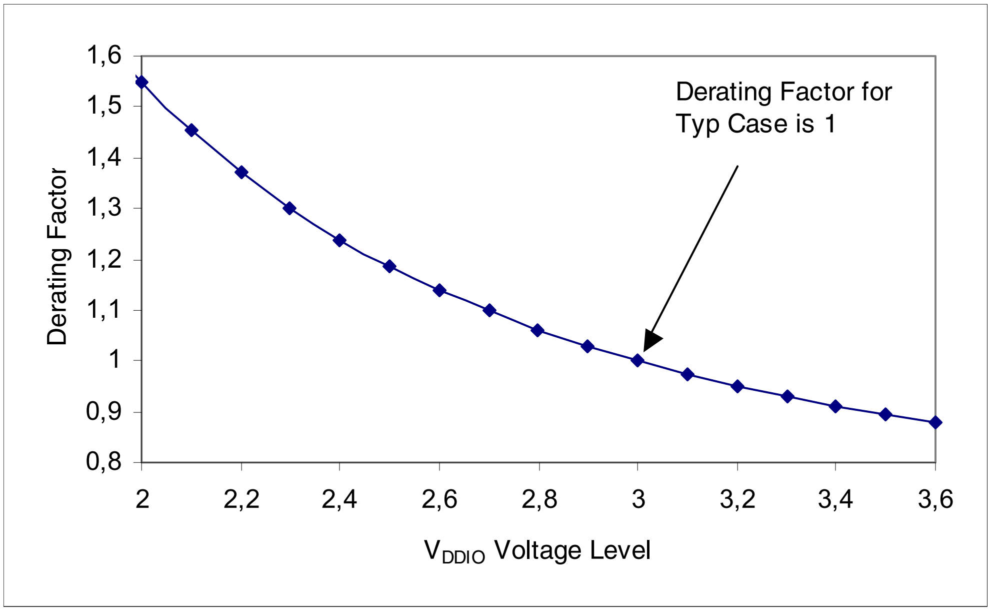
21.1.3 Core Voltage Derating Factor

Figure 21-2. Core Voltage Derating Factor



21.1.4 IO Voltage Derating Factor

Figure 21-3. Derating Factor for Different V_{DDIO} Power Supply Levels



21.2 Peripheral Signals

21.2.1 USART Signals

The inputs have to meet the minimum pulse width and period constraints shown in Table 21-1 and Table 21-2, and represented in Figure 21-4.

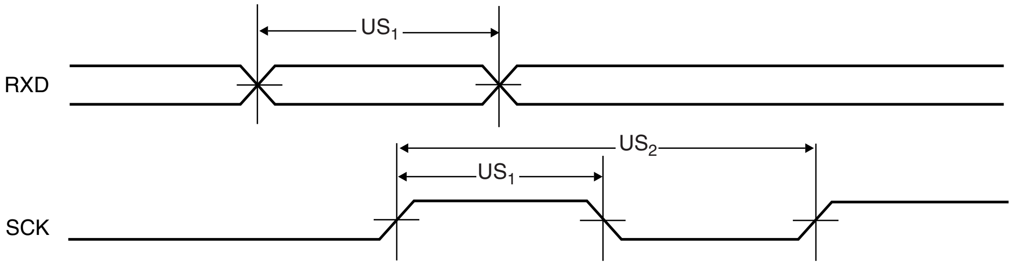
Table 21-1. USART Input Minimum Pulse Width

Symbol	Parameter	Min Pulse Width	Units
US_1	SCK/RXD Minimum Pulse Width	$5(t_{CP}/2)$	ns

Table 21-2. USART Minimum Input Period

Symbol	Parameter	Min Input Period	Units
US_2	SCK Minimum Input Period	$9(t_{CP}/2)$	ns

Figure 21-4. USART Signals



21.2.2 Timer/Counter Signals

Due to internal synchronization of input signals, there is a delay between an input event and a corresponding output event. This delay is $3(t_{CP})$ in Waveform Event Detection mode and $4(t_{CP})$ in Waveform Total-count Detection mode. The inputs have to meet the minimum pulse width and minimum input period shown in Table 21-3 and Table 21-4, and as represented in Figure 21-5.

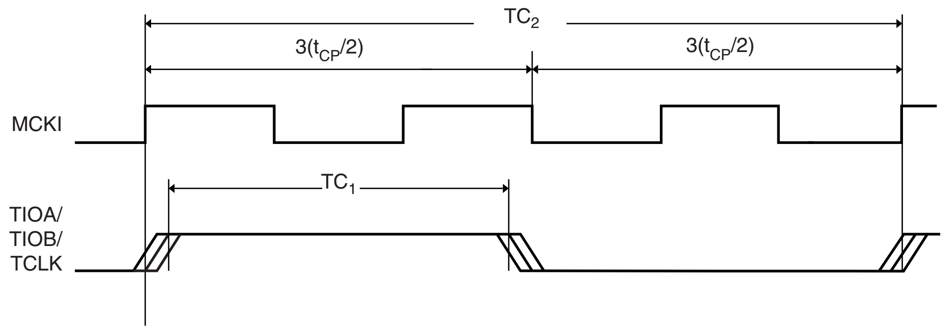
Table 21-3. Timer Input Minimum Pulse Width

Symbol	Parameter	Min Pulse Width	Units
TC_1	TCLK/TIOA/TIOB Minimum Pulse Width	$3(t_{CP}/2)$	ns

Table 21-4. Timer Input Minimum Period

Symbol	Parameter	Min Input Period	Units
TC_2	TCLK/TIOA/TIOB Minimum Input Period	$5(t_{CP}/2)$	ns

Figure 21-5. Timer Input



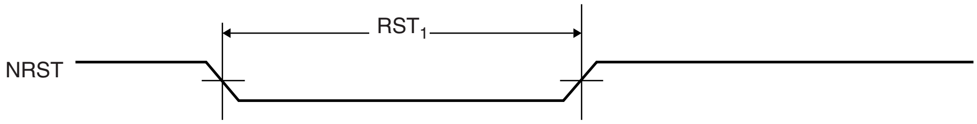
21.2.3 Reset Signals

A minimum pulse width is necessary, as shown in Table 21-5 and as represented in Figure 21-6.

Table 21-5. Reset Minimum Pulse Width

Symbol	Parameter	Min Pulse-width	Units
RST$_1$	NRST Minimum Pulse Width	10(t$_{CP}$)	ns

Figure 21-6. Reset Signal



Only the NRST rising edge is synchronized with MCKI. The falling edge is asynchronous.

21.2.4 Advanced Interrupt Controller Signals

Inputs have to meet the minimum pulse width and minimum input period shown in Table 21-6 and Table 21-7 and represented in Figure 21-7.

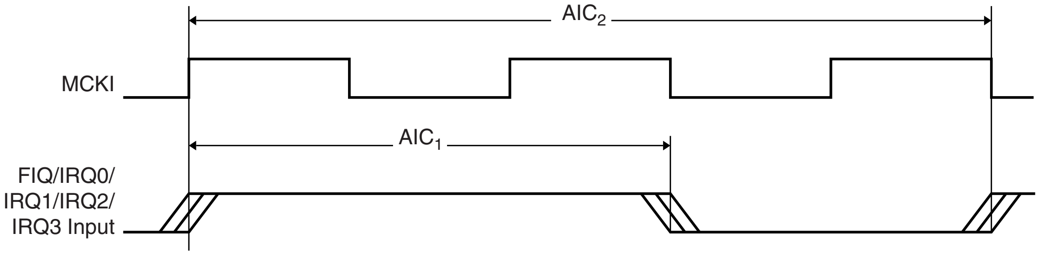
Table 21-6. AIC Input Minimum Pulse Width

Symbol	Parameter	Min Pulse Width	Units
AIC$_1$	FIQ/IRQ0/IRQ1/IRQ2/IRQ3 Minimum Pulse Width	3(t$_{CP}$/2)	ns

Table 21-7. AIC Input Minimum Period

Symbol	Parameter	Min Input Period	Units
AIC$_2$	AIC Minimum Input Period	5(t$_{CP}$/2)	ns

Figure 21-7. AIC Signals



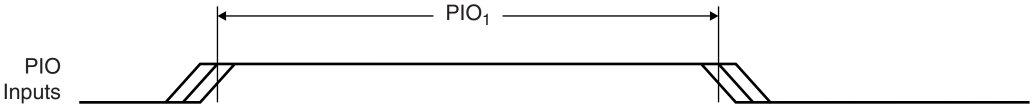
21.2.5 Parallel I/O Signals

The inputs have to meet the minimum pulse width shown in Table 21-8 and represented in Figure 21-8.

Table 21-8. PIO Input Minimum Pulse Width

Symbol	Parameter	Min Pulse Width	Units
PIO_1	PIO Input Minimum Pulse Width	$3(t_{CP}/2)$	ns

Figure 21-8. PIO Signal

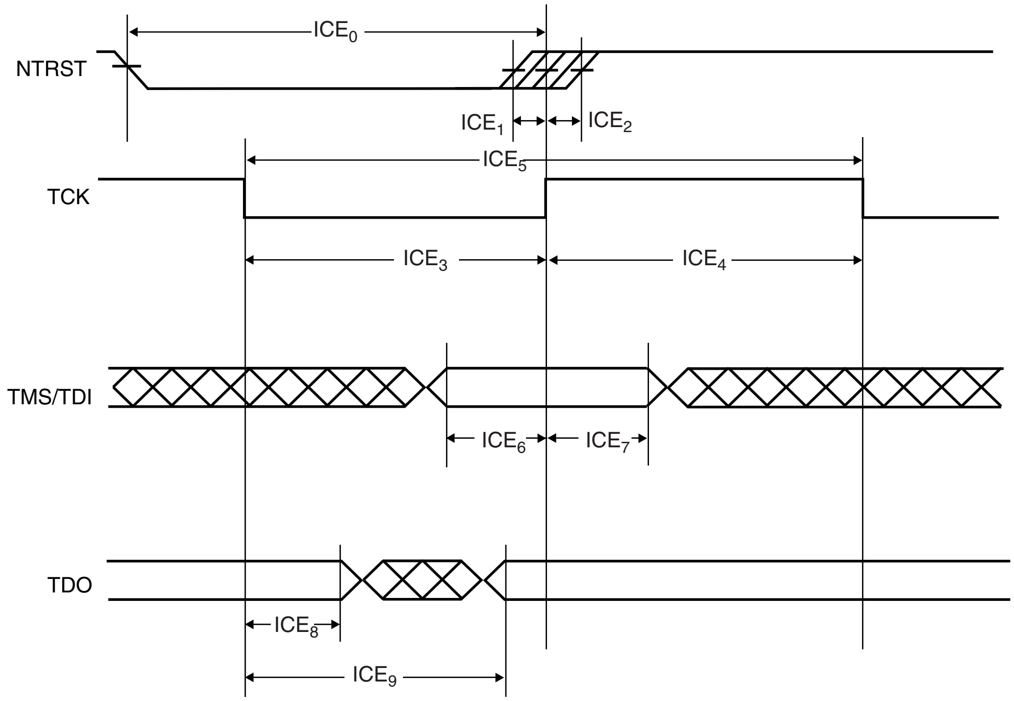


21.2.6 ICE Interface Signals

Table 21-9. ICE Interface Timing Specifications

Symbol	Parameter	Conditions	Min	Max	Units
ICE_0	NTRST Minimum Pulse Width		10.9		ns
ICE_1	NTRST High Recovery to TCK High		0.9		ns
ICE_2	NTRST High Removal from TCK High		-0.3		ns
ICE_3	TCK Low Half-period		23.5		ns
ICE_4	TCK High Half-period		22.7		ns
ICE_5	TCK Period		46.1		ns
ICE_6	TDI, TMS Setup before TCK High		0.4		ns
ICE_7	TDI, TMS Hold after TCK High		0.4		ns
ICE_8	TDO Hold Time	C_{TDO} = 0 pF	3.3		ns
		C_{TDO} derating	0.001		ns/pF
ICE_9	TCK Low to TDO Valid	C_{TDO} = 0 pF		7.4	ns
		C_{TDO} derating		0.28	ns/pF

Figure 21-9. ICE Interface Signal



21.3 EBI Signals Relative to MCKI

The following tables show timings relative to operating condition limits defined in the section "Conditions and Timing Results" on page 183.

Table 21-10. General-purpose EBI Signals

Symbol	Parameter	Conditions	Min	Max	Units
EBI_1	MCKI Falling to NUB Valid	C_{NUB} = 0 pF	4.4	8.9	ns
		C_{NUB} derating	0.030	0.043	ns/pF
EBI_2	MCKI Falling to NLB/A0 Valid	C_{NLB} = 0 pF	3.7	6.7	ns
		C_{NLB} derating	0.045	0.069	ns/pF
EBI_3	MCKI Falling to A1 - A23 Valid	C_{ADD} = 0 pF	3.4	7.8	ns
		C_{ADD} derating	0.045	0.076	ns/pF
EBI_4	MCKI Falling to Chip Select Change	C_{NCS} = 0 pF	3.7	8.6	ns
		C_{NCS} derating	0.045	0.078	ns/pF
EBI_5	NWAIT Setup before MCKI Rising		1.7		ns
EBI_6	NWAIT Hold after MCKI Rising		1.7		ns

Table 21-11. EBI Write Signals

Symbol	Parameter	Conditions	Min	Max	Units
EBI_7	MCKI Rising to NWR Active (No Wait States)	C_{NWR} = 0 pF	3.9	6.3	ns
		C_{NWR} derating	0.029	0.043	ns/pF
EBI_8	MCKI Rising to NWR Active (Wait States)	C_{NWR} = 0 pF	4.4	7.0	ns
		C_{NWR} derating	0.029	0.043	ns/pF
EBI_9	MCKI Falling to NWR Inactive (No Wait States)	C_{NWR} = 0 pF	3.8	6.3	ns
		C_{NWR} derating	0.029	0.044	ns/pF
EBI_{10}	MCKI Rising to NWR Inactive (Wait States)	C_{NWR} = 0 pF	4.2	6.7	ns
		C_{NWR} derating	0.029	0.044	ns/pF
EBI_{11}	MCKI Rising to D0 - D15 Out Valid	C_{DATA} = 0 pF	4.2	7.5	ns
		C_{DATA} derating	0.045	0.080	ns/pF
EBI_{12}	NWR High to NUB Change	C_{NUB} = 0 pF	3.1	7.0	ns
		C_{NUB} derating	0.030	0.043	ns/pF
EBI_{13}	NWR High to NLB/A0 Change	C_{NLB} = 0 pF	3.1	5.4	ns
		C_{NLB} derating	0.043	0.073	ns/pF
EBI_{14}	NWR High to A1 - A23 Change	C_{ADD} = 0 pF	2.9	7.0	ns
		C_{ADD} derating	0.043	0.076	ns/pF
EBI_{15}	NWR High to Chip Select Inactive	C_{NCS} = 0 pF	2.9	6.8	ns
		C_{NCS} derating	0.052	0.067	ns/pF

Table 21-11. EBI Write Signals (Continued)

Symbol	Parameter	Conditions	Min	Max	Units
EBI_{16}	Data Out Valid before NWR High (No Wait States)[1]	C = 0 pF	t_{CH} - 1.8		ns
		C_{DATA} derating	-0.080		ns/pF
		C_{NWR} derating	0.044		ns/pF
EBI_{17}	Data Out Valid before NWR High (Wait States)[1]	C = 0 pF	n x t_{CP} - 1.3[2]		ns
		C_{DATA} derating	-0.080		ns/pF
		C_{NWR} derating	0.044		ns/pF
EBI_{18}	Data Out Valid after NWR High (No Wait States)		2.2		ns
EBI_{18}bis	Data Out Valid after NWR High (Wait States)		$t_{CP}/2 + EBI_{18}$		ns
EBI_{19}	NWR Minimum Pulse Width (No Wait States)[1]	C_{NWR} = 0 pF	t_{CH} - 0.6		ns
		C_{NWR} derating	0		ns/pF
EBI_{20}	NWR Minimum Pulse Width (Wait States)[1]	C_{NWR} = 0 pF	n x t_{CP} - 0.9[2]		ns
		C_{NWR} derating	0		ns/pF

Notes: 1. The derating factor should not be applied to t_{CH} or t_{CP}

2. n = number of standard wait states inserted.

Table 21-12. EBI Read Signals

Symbol	Parameter	Conditions	Min	Max	Units
EBI_{21}	MCKI Falling to NRD Active[1]	C_{NRD} = 0 pF	4.5	7.9	ns
		C_{NRD} derating	0.029	0.043	ns/pF
EBI_{22}	MCKI Rising to NRD Active[2]	C_{NRD} = 0 pF	3.8	7.3	ns
		C_{NRD} derating	0.029	0.043	ns/pF
EBI_{23}	MCKI Falling to NRD Inactive[1]	C_{NRD} = 0 pF	4.1	6.5	ns
		C_{NRD} derating	0.030	0.044	ns/pF
EBI_{24}	MCKI Falling to NRD Inactive[2]	C_{NRD} = 0 pF	3.9	5.8	ns
		C_{NRD} derating	0.030	0.044	ns/pF
EBI_{25}	D0 - D15 In Setup before MCKI Falling Edge[5]		1.5		ns
EBI_{26}	D0 - D15 In Hold after MCKI Falling Edge[6]		1.2		ns
EBI_{27}	NRD High to NUB Change	C_{NUB} = 0 pF	3.2	7.1	ns
		C_{NUB} derating	0.030	0.043	ns/pF
EBI_{28}	NRD High to NLB/A0 Change	C_{NLB} = 0 pF	3.2	4.6	ns
		C_{NLB} derating	0.043	0.073	ns/pF
EBI_{29}	NRD High to A1 - A23 Change	C_{ADD} = 0 pF	2.8	6.1	ns
		C_{ADD} derating	0.043	0.076	ns/pF
EBI_{30}	NRD High to Chip Select Inactive	C_{NCS} = 0 pF	2.9	6.2	ns
		C_{NCS} derating	0.052	0.067	ns/pF

Table 21-12. EBI Read Signals (Continued)

Symbol	Parameter	Conditions	Min	Max	Units
EBI_{31}	Data Setup before NRD High [5]	C_{NRD} = 0 pF	8.0		ns
		C_{NRD} derating	0.044		ns/pF
EBI_{32}	Data Hold after NRD High[6]	C_{NRD} = 0 pF	-3.1		ns
		C_{NRD} derating	-0.030		ns/pF
EBI_{33}	NRD Minimum Pulse Width[1][3]	C_{NRD} = 0 pF	(n +1) t_{CP} - 1.9[4]		ns
		C_{NRD} derating	0.001		ns/pF
EBI_{34}	NRD Minimum Pulse Width[2][3]	C_{NRD} = 0 pF	n x t_{CP} + (t_{CH} - 1.5)[4]		ns
		C_{NRD} derating	0.001		ns/pF

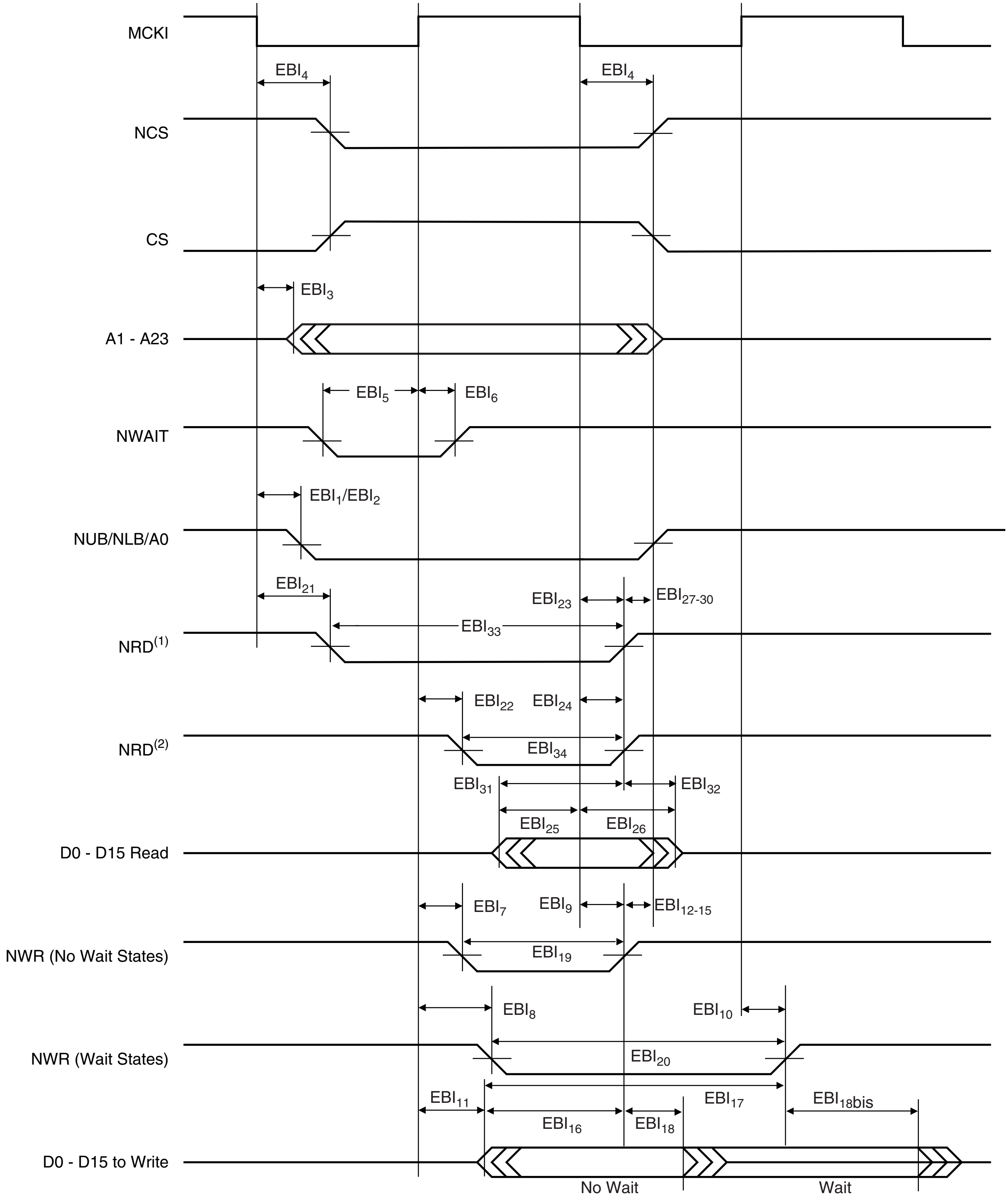
Notes:
1. Early Read Protocol.
2. Standard Read Protocol.
3. The derating factor should not be applied to t_{CH} or t_{CP}
4. n = number of standard wait states inserted.
5. Only one of these two timings, EB_{25} or EBI_{31}, needs to be met.
6. Only one of these two timings, EB_{26} or EBI_{32}, needs to be met.

Table 21-13. EBI Read and Write Control Signals. Capacitance Limitation

Symbol	Parameter	Conditions	Min	Max	Units
T_{CPLNRD}[1]	Master Clock Low Due to NRD Capacitance	C_{NRD} = 0 pF	7.3		ns
		C_{NRD} derating	0.044		ns/pF
T_{CPLNWR}[2]	Master CLock Low Due to NWR Capacitance	C_{NWR} = 0 pF	7.6		ns
		C_{NWR} derating	0.044		ns/pF

Notes:
1. If this condition is not met, the action depends on the read protocol intended for use.
   - Early Read Protocol: Programing an additional t_{DF} (Data Float Output Time) cycle.
   - Standard Read Protocol: Programming an additional t_{DF} Cycle and an additional wait state.
2. Applicable only for chip select programmed with 0 wait state. If this condition is not met, at least one wait state must be programmed.

Figure 21-10. EBI Signals Relative to MCKI



Notes: 1. Early Read Protocol.

2. Standard Read Protocol.

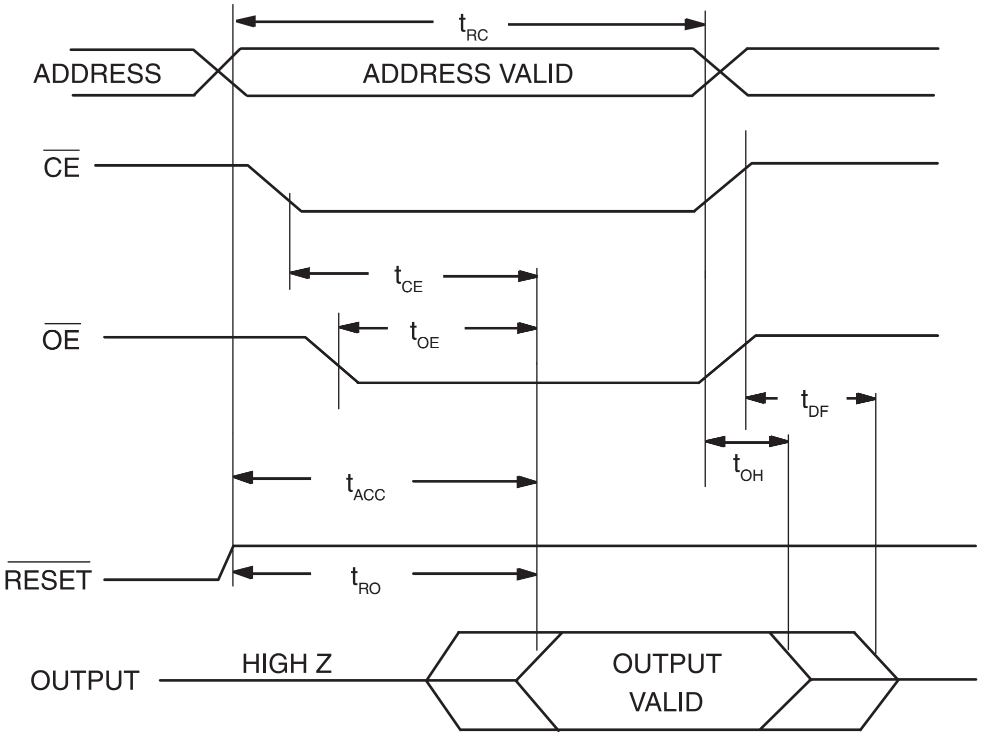
21.4 AC Flash Read Characteristics

Table 21-14. AC Flash Read Characteristics

Symbol	Parameter	Min	Max	Units
t_{RC}	Read Cycle Time	70		ns
t_{ACC}	Address to Output Delay		70	ns
t_{CE}(1)	$\overline{\text{CE}}$ to Output Delay		70	ns
t_{OE}(2)	$\overline{\text{OE}}$ to Output Delay	0	20	ns
t_{DF}(3)(4)	$\overline{\text{CE}}$ or $\overline{\text{OE}}$ to Output Float	0	25	ns
t_{OH}	Output Hold from $\overline{\text{OE}}$, $\overline{\text{CE}}$ or Address, whichever occurred first	0		ns
t_{RO}	$\overline{\text{RESET}}$ to Output Delay		100	ns

21.4.1 AC Read Waveforms

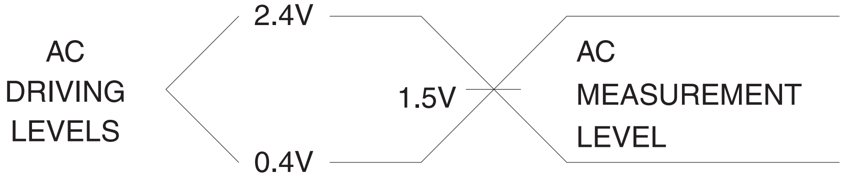
Figure 21-11. AC Read Waveforms(1) (2) (3) (4)



Notes:
1. $\overline{\text{CE}}$ may be delayed up to t_{ACC} - t_{CE} after the address transition without impact on t_{ACC}.
2. $\overline{\text{OE}}$ may be delayed up to t_{CE} - t_{OE} after the falling edge of $\overline{\text{CE}}$ without impact on t_{CE} or by t_{ACC} - t_{OE} after an address change without impact on t_{ACC}.
3. t_{DF} is specified from $\overline{\text{OE}}$ or $\overline{\text{CE}}$, whichever occurs first (CL = 5 pF).
4. This parameter is characterized and is not 100% tested.

21.4.2 Input Test Waveforms and Measurement Level

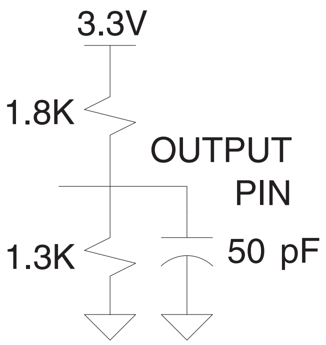
Figure 21-12. Input Test Waveforms and Measurement Level



t_R, t_F < 5 ns

21.4.3 Output Test Load

Figure 21-13. Output Test Load



21.4.4 Pin Capacitance

Table 21-15. Pin Capacitance f = 1 MHz, T = 25°C[1]

Symbol	Conditions	Typ	Max	Units
C_{IN}	V_{IN} = 0V	4	6	pF
C_{OUT}	V_{OUT} = 0V	8	12	pF

Note: 1. This parameter is characterized and is not 100% tested.

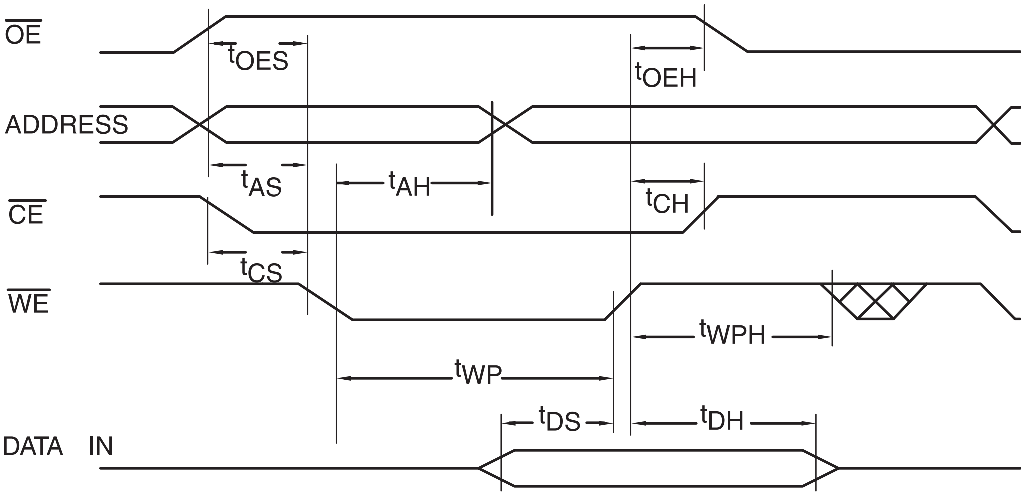
21.5 AC Flash Byte/Word Load Waveforms

Table 21-16. AC Byte/Word Load Waveforms

Symbol	Parameter	Min	Max	Units
t_{AS}, t_{OES}	Address, $\overline{OE}$ Setup Time	0		ns
t_{AH}	Address Hold Time	35		ns
t_{CS}	Chip Select Setup Time	0		ns
t_{CH}	Chip Select Hold Time	0		ns
t_{WP}	Write Pulse Width ($\overline{WE}$ or $\overline{CE}$)	35		ns
t_{WPH}	Write Pulse Width High	35		ns
t_{DS}	Data Setup Time	35		ns
t_{DH}, t_{OEH}	Data, $\overline{OE}$ Hold Time	0		ns

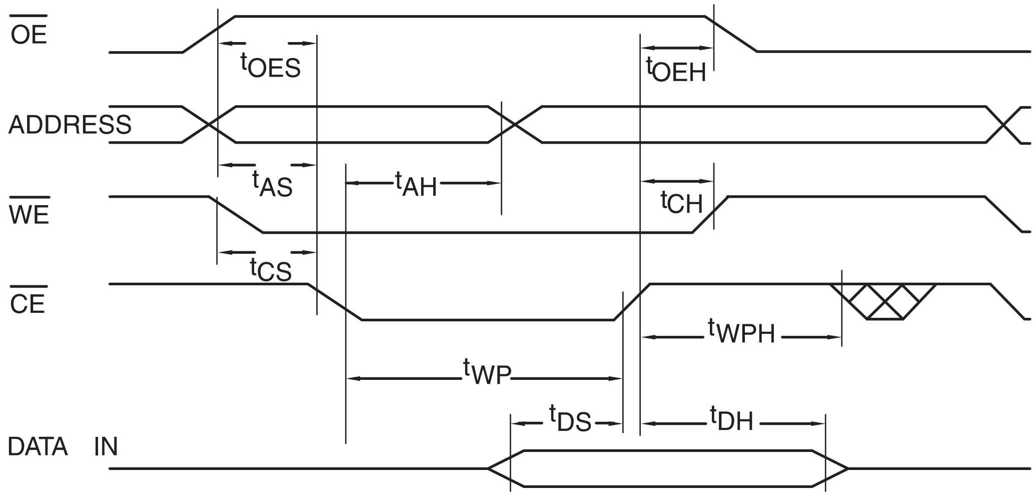
21.5.1 $\overline{WE}$ Controlled

Figure 21-14. $\overline{WE}$ Controlled



21.5.2 $\overline{\text{CE}}$ Controlled

Figure 21-15. CE Controlled



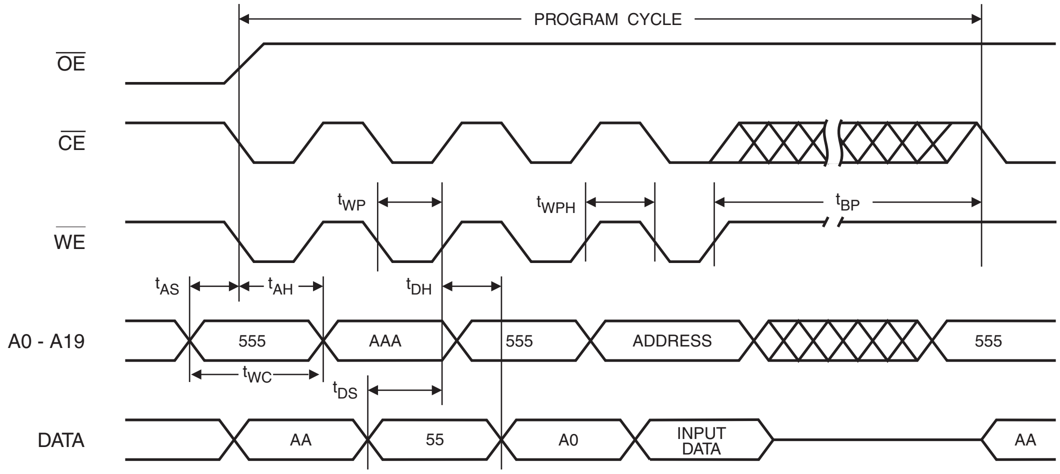
21.6 Flash Program Cycle Characteristics

Table 21-17. Program Cycle Characteristics

Symbol	Parameter	Min	Typ	Max	Units
t_{BP}	Byte/Word Programming Time		12	200	µs
t_{AS}	Address Setup Time	0			ns
t_{AH}	Address Hold Time	35			ns
t_{DS}	Data Setup Time	35			ns
t_{DH}	Data Hold Time	0			ns
t_{WP}	Write Pulse Width	35			ns
t_{WPH}	Write Pulse Width High	35			ns
t_{WC}	Write Cycle Time	70			ns
t_{RP}	$\overline{\text{Reset}}$ Pulse Width	500			ns
t_{EC}	Chip Erase Cycle Time		25		seconds
t_{SEC1}	Sector Erase Cycle Time (4K Word Sectors)		0.3	3.0	seconds
t_{SEC2}	Sector Erase Cycle Time (32K Word Sectors)		1.0	6.0	seconds
t_{ES}	Erase Suspend Time			15	µs
t_{PS}	Program Suspend Time			10	µs
t_{ERES}	Delay between Erase Resume and Erase Suspend	500			ns

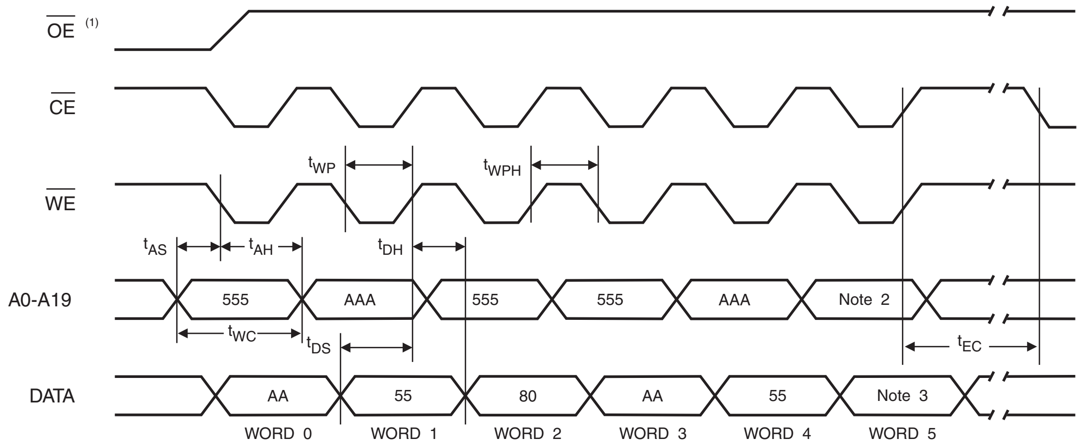
21.6.1 Program Cycle Waveforms

Figure 21-16. Program Cycle Waveforms



21.6.2 Sector or Chip Erase Cycle Waveforms

Figure 21-17. Sector or Chip Erase Cycle Waveforms



Notes:

1. $\overline{OE}$ must be high only when $\overline{WE}$ and $\overline{CE}$ are both low.
2. For chip erase, the address should be 555. For sector erase, the address depends on what sector is to be erased. (See footnote [3] of Table 12-2, "Command Definition Table," on page 63.)
3. For chip erase, the data should be 10H, and for sector erase, the data should be 30H.

21.7 Flash $\overline{\text{Data}}$ Polling Characteristics

Table 21-18. $\overline{\text{Data}}$ Polling Characteristics[1]

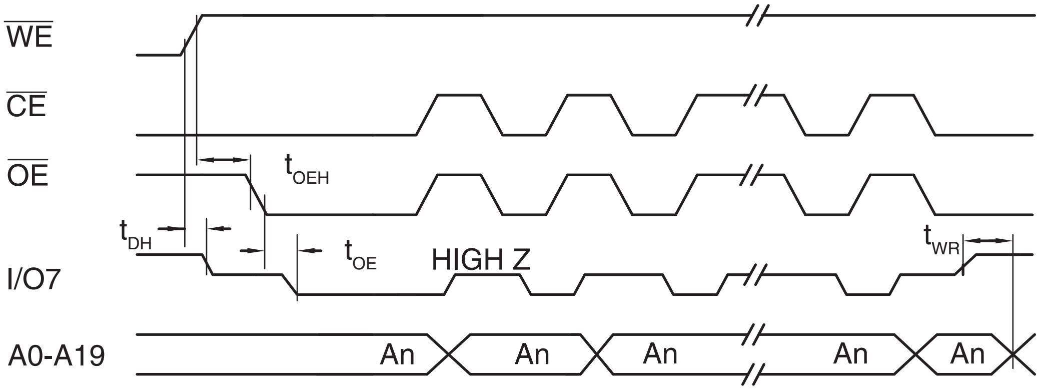
Symbol	Parameter	Min	Typ	Max	Units
t_{DH}	Data Hold Time	10			ns
t_{OEH}	$\overline{\text{OE}}$ Hold Time	10			ns
t_{OE}	$\overline{\text{OE}}$ to Output Delay[2]				ns
t_{WR}	Write Recovery Time	0			ns

Notes: 1. These parameters are characterized and not 100% tested.

2. See t_{OE} spec in "AC Flash Read Characteristics" on page 194.

21.7.1 $\overline{\text{Data}}$ Polling Waveforms

Figure 21-18. $\overline{\text{Data}}$ Polling Waveforms



21.8 Flash Toggle Bit Characteristics

Table 21-19. Toggle Bit Characteristics[1]

Symbol	Parameter	Min	Typ	Max	Units
t_{DH}	Data Hold Time	10			ns
t_{OEH}	$\overline{OE}$ Hold Time	10			ns
t_{OE}	$\overline{OE}$ to Output Delay[2]				ns
t_{OEHP}	$\overline{OE}$ High Pulse	50			ns
t_{WR}	Write Recovery Time	0			ns

Notes: 1. These parameters are characterized and not 100% tested.

2. See t_{OE} spec in "AC Flash Read Characteristics" on page 194.

21.8.1 Toggle Bit Waveforms

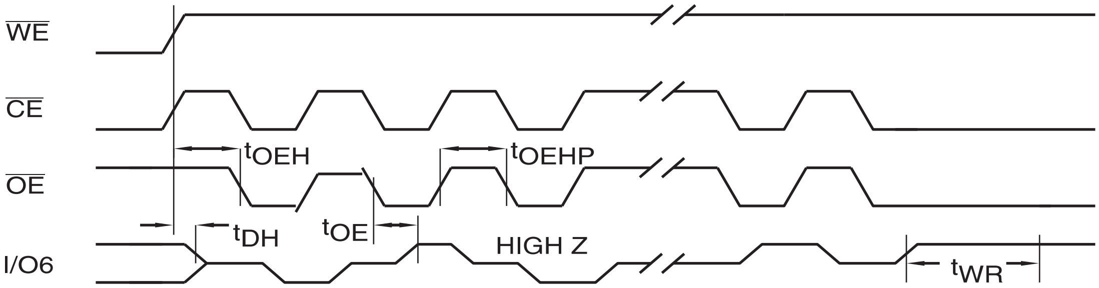


Figure 21-19. Toggle Bit Waveforms [1][2][3]

Notes: 1. Toggling either $\overline{OE}$ or $\overline{CE}$ or both $\overline{OE}$ and $\overline{CE}$ will operate toggle bit. The t_{OEHP} specification must be met by the toggling input(s).

2. Beginning and ending state of I/O6 will vary.

3. Any address location may be used but the address should not vary.

22. Mechanical Characteristics

22.1 Package Drawing

Figure 22-1. AT91FR40162SBPackage

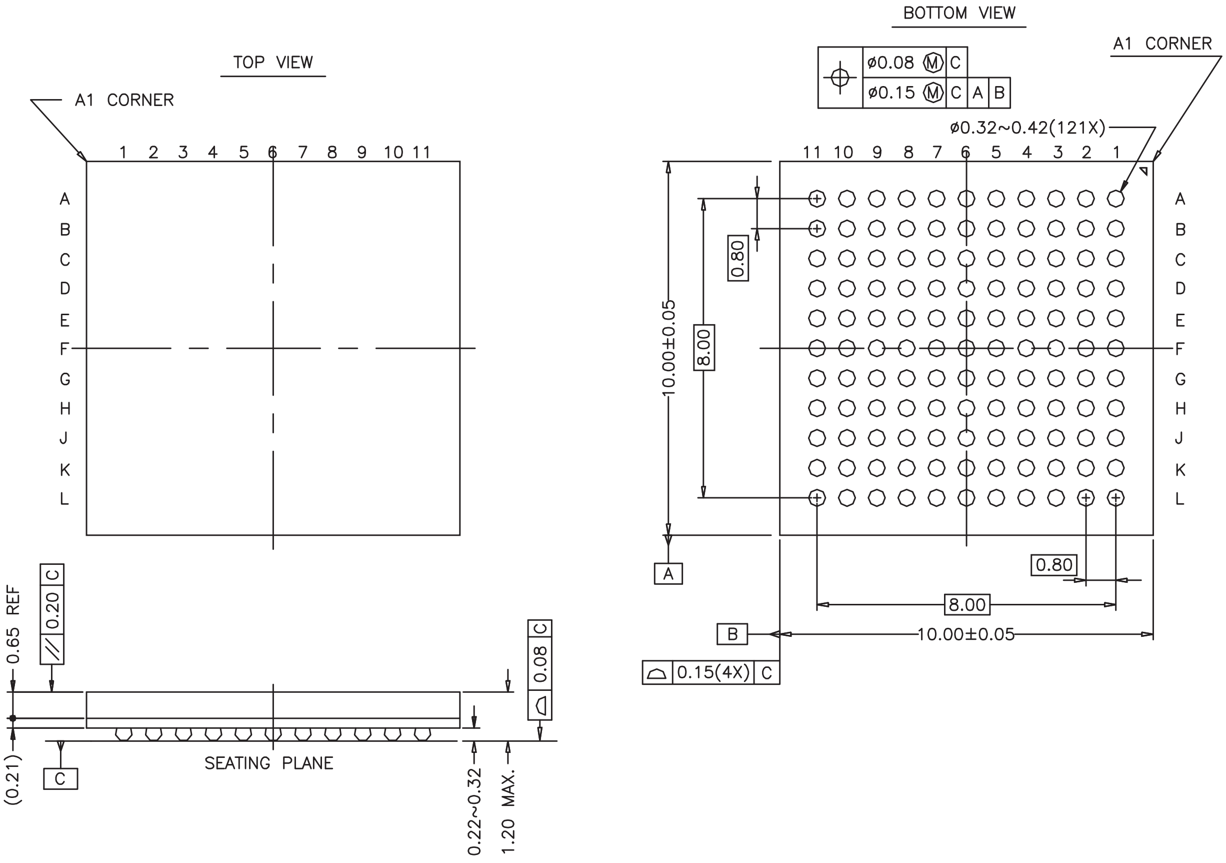


Table 22-1. Device and 121-ball BGA Package Maximum Weight

194	mg

Table 22-2. 121-ball BGA Package Characteristics

Ball diameter	0.35 mm
Ball land	0.4 ± 0.05 mm
Solder mask opening	0.3 ± 0.05 mm
Plating material	Copper
Solder ball material	Sn/Ag/Cu
Moisture Sensitivity Level	3

Table 22-3. Package Reference

JESD97 Classification	e1

This package respects the recommendations of the NEMI User Group

22.2 Soldering Profile

Table 22-4 gives the recommended soldering profile from J-STD-20C.

Table 22-4. Soldering Profile

Profile Feature	Convection or IR/Convection
Average Ramp-up Rate (183· C to Peak)	3· C/sec. max.
Preheat Temperature 125· C ±25· C	180 sec. max
Temperature Maintained Above 183· C	60 sec. to 150 sec.
Time within 5· C of Actual Peak Temperature	20 sec. to 40 sec.
Peak Temperature Range	260 · C
Ramp-down Rate	6· C/sec.
Time 25· C to Peak Temperature	8 min. max

Note: It is recommended to apply a soldering temperature higher than 250°C.

A maximum of three reflow passes is allowed per component.

23. Ordering Information

Table 23-1. Ordering Information

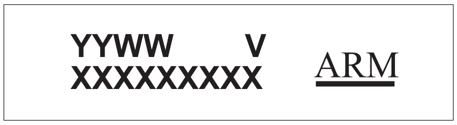
Ordering Code	Package	Package Type	Temperature Operating Range
AT91FR40162SB-CU	BGA 121	Green	Industrial (-40· C to 85· C)

24. AT91FR40162SB Errata

24.1 Marking

All devices are marked with the Atmel logo and the ordering code.

Additional marking has the following format:



where

- "YY": manufactory year
- "WW": manufactory week
- "V": revision

"XXXXXXXXX": lot number

24.2 Flash

24.2.1 Flash: Erroneous Read of the first instruction out of the integrated Flash memory

At power-up when fetching the first instruction out of the integrated Flash memory, the ARM processor may read the first two 16-bit words at 0xFFFF instead of the prgrammed value, preventing the device to start-up correctly.

Problem Fix/Workaround

Software workaround is to add 0xFFFFFFFF, 32-bit instruction at the beginning of the program code. When reading instruction 0xFFFFFFFF, equivalent to a NOP instruction, the ARM7TDMI processor will execute the next instruction corresponding to the true first instruction of the program you want the processor to run.

25. Revision History

Doc. Rev.		Change Request Ref.
6410A	First Issue	
6410B	Section 2.3.3 “Erase Cycle Timings”, updated section.	5617
	Section 24. “AT91FR40162SB Errata”, Section 24.2.1 “Flash: Wrong Read of the first instruction out of the integrated Flash memory”, added to errata.	6942

Table of Contents

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com
www.atmel.com/AT91SAM

Technical Support
AT91SAM Support
Atmel techincal support

Sales Contacts
www.atmel.com/contacts/

Literature Requests
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.