Features

- ARM7TDMI® ARM® Thumb® Processor Core
  - High-performance 32-bit RISC
  - High-density 16-bit Thumb Instruction Set
  - Leader in MIPS/Watt
  - Embedded ICE (In Circuit Emulation)
- 4 Kbytes Internal RAM
- Clock Manager (CM) with Programmable PLL
  - PLL Multiplier from x2 to x20
  - 32.768 kHz Oscillator for Low-power Operation
  - Master Clock Divider/multiplier
- Fully Programmable External Bus Interface (EBI) through Advanced Memory Controller (AMC)
  - Maximum External Address Space of 16 Mbytes, Up to Six Chip Select Lines
- 8-level Priority, Vectored Interrupt Controller
  - Individually Maskable, Two External Interrupts including One Fast Interrupt Line
- 11-channel Peripheral Data Controller (PDC)
- 49 Programmable I/O Lines
- One 3-channel 16-bit General Purpose Timers (GPT)
  - Three Configurable Modes: Counter, PWM, Capture
  - Three Multi-purpose I/O Pins Per Channel
- Four 16-bit Simple Timers (ST)
- 4-channel 16-bit Pulse Width Modulation (PWM)
- Two 16-bit Capture Modules (CAPT)
- CAN Controller 2.0A and 2.0B Full CAN (16 Buffers)
- Three USARTs
  - Six Peripheral Data Controller (PDC) Channels
  - Support for Up to 9-bit Data Lengths
  - Support for LIN (Software) Protocol
- Master SPI Interface
  - Two Peripheral Data Controller (PDC) Channels
  - 8- to 16-bit Programmable Data Length
  - Four External Chip Select Lines
- One 8-channel 10-bit Analog-to-digital Converter (ADC)
  - One Peripheral Data Controller (PDC) Channel
- Programmable Watch Timer (WT)
- Programmable Watchdog (WD)
- Power Management Controller (PMC)
  - CPU and Peripherals Can Be Deactivated Individually
- Fully Static Operation Up to 40 MHz
  - 3.0V to 3.6V Core, Memory and Analog Voltage Range
  - 3.0 V to 5.5V Compliant I/Os
  - -40° to +85°C Operating Temperature Range
- Available in a 144-pin LQFP

AT91 ARM Thumb-based Microcontroller

AT91SAM7A1

6048B–ATARM–29-Jun-06

1. Description

The AT91SAM7A1 is a member of the Atmel Smart ARM Microcontrollers product family, based on the ARM7TDMI embedded processor. This processor has a high-performance 32-bit RISC architecture with a high-density 16-bit instruction set and very low power consumption.

In addition, a large number of internally banked registers result in very fast exception handling, making the device ideal for real-time control applications.

The AT91SAM7A1 has a direct connection to off-chip memory, including Flash, through the fully-programmable External Bus Interface.

An 8-level priority vectored Interrupt Controller in conjunction with the Peripheral Data Controller significantly improves the real-time performance of the device.

The device is manufactured using high-density CMOS technology.

By combining the ARM7TDMI processor with an on-chip RAM and a wide range of peripheral functions on a monolithic chip, the AT91SAM7A1 is a powerful device that provides a flexible, cost-effective solution to many compute-intensive embedded control applications in the industrial world.

2. Pin Configuration

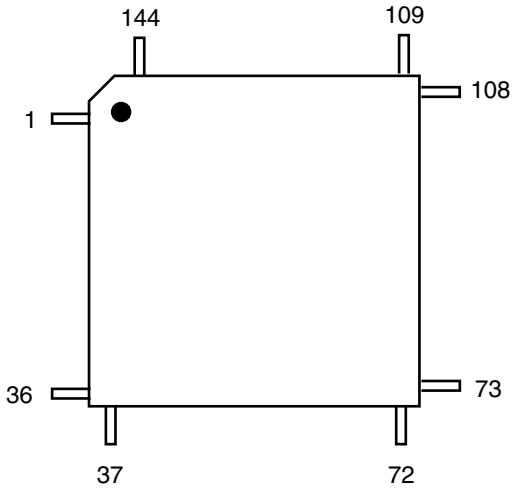
The AT91SAM7A1 is available in a 144-lead LQFP package.

2.1 144-lead LQFP Package

Figure 2-1 shows the orientation of the 144-lead LQFP package.

A detailed mechanical description is given in the section Mechanical Characteristics.

Figure 2-1. 144-lead LQFP Package



2.2 144-lead LQFP Package Pinout

Table 2-1. AT91SAM7A1 Pinout for 144-lead LQFP Package

Pin	Name	Pin	Name	Pin	Name	Pin	Name
1	D0	37	ADD11	73	GND5V (I/O)	109	ANA0IN1
2	D8	38	ADD12	74	PIOA2	110	ANA0IN2
3	D1	39	ADD13	75	PIOA3	111	ANA0IN3
4	D9	40	ADD14	76	VDD5V (I/O)	112	ANA0IN4
5	VDD3V (I/O)	41	ADD15	77	PIOA4	113	ANA0IN5
6	GND3V (I/O+CORE)	42	GND3V (IO)	78	PIOA5	114	ANA0IN6
7	VDD3V (I/O+CORE)	43	VDD3V (CORE)	79	PIOA6	115	ANA0IN7
8	D2	44	VDD5V (IO)	80	PIOA7	116	GND3V (ANA)
9	D10	45	IRQ0	81	PIOA8	117	VDD3V (PLL)
10	D3	46	FIQ	82	PIOA9	118	MCKI
11	D11	47	T0TIOA0/MPIO	83	GND5V (I/O)	119	MCKO
12	D4	48	T0TIOB0/MPIO	84	PIOA10	120	PLLRC
13	D12	49	T0TCLK0/MPIO	85	PIOA11	121	GND3V (PLL)
14	D5	50	T0TIOA1/MPIO	86	PIOA12	122	VDD3V (RTCK)
15	D13	51	T0TIOB1/MPIO	87	PIOA13	123	RTCKI
16	D6	52	T0TCLK1/MPIO	88	PIOA14	124	RTCKO
17	D14	53	T0TIOA2/MPIO	89	PIOA15	125	GND3V (RTCK)
18	D7	54	T0TIOB2/MPIO	90	PIOA16	126	VDD3V (I/O)
19	D15	55	GND5V (I/O)	91	PIOA17	127	GND3V (CORE)
20	ADD17	56	T0TCLK2/MPIO	92	PWM0/MPIO	128	GND3V (I/O)
21	ADD16	57	TXD0/MPIO	93	VDD5V (I/O)	129	SCANEN
22	NWR0/NWE	58	RXD0/MPIO	94	PWM1/MPIO	130	TEST
23	ADD19	59	SCK0/MPIO	95	PWM2/MPIO	131	TMS
24	ADD18	60	TXD1/MPIO	96	PWM3/MPIO	132	TDO
25	ADD7	61	RXD1/MPIO	97	CAPT0/MPIO	133	TDI
26	ADD6	62	SCK1/MPIO	98	CAPT1/MPIO	134	TCK
27	GND3V (I/O+CORE)	63	VDD5V (I/O)	99	NRESET	135	NWAIT
28	VDD3V (I/O+CORE)	64	SPCK/MPIO	100	CANRX0	136	ADD21/CS6
29	ADD2	65	MISO/MPIO	101	CANTX0	137	NCS3
30	ADD3	66	MOSI/MPIO	102	TXD2/MPIO	138	NCS2
31	ADD4	67	NPCS0/NSS/MPIO	103	RXD2/MPIO	139	NWR1/NUB
32	ADD5	68	NPCS1/MPIO	104	SCK2/MPIO	140	ADD0/NLB
33	ADD8	69	NPCS2/MPIO	105	GND5V (I/O)	141	NCS1
34	ADD20/CS7	70	NPCS3/MPIO	106	VDD3V (ANA)	142	NOE/NRD
35	ADD9	71	PIOA0	107	VREFP0	143	NCS0
36	ADD10	72	PIOA1	108	ANA0IN0	144	ADD1

3. Signal Description

Table 3-1. Pin Description

Signal Name	Function	Type[1]	Level[1]	Comments
EBI[2]				
ADD[19:1]	External address bus	O	(Z)	The EBI is tri-stated when NRESET is at a logical low level. Internal pull-downs on data bus bits. ADD20 and ADD21 are address lines at reset.
ADD0/NLB	External address line/Lower byte enable	O	L (Z)	
ADD20/CS7	External address line/Chip select	O	H (Z)	
ADD21/CS6	External address line/Chip select	O	H (Z)	
D[15:0][3]	External data bus	I/O	(Z)	
NOE/NRD	Output enable	O	L (Z)	
NWR0/NWE	Write enable	O	L (Z)	
NCS[3:0]	Chip select lines	O	L (Z)	
NWR1/NUB	Upper byte enable	O	L (Z)	
NWAIT	Wait input	I	L	Internal pull-up (must be connected to VCC or leave unconnected for normal operation if functionality not used)
GIC				
IRQ0	External interrupt line	I		
FIQ	Fast interrupt line	I		
Power-on Reset				
NRESET	Hardware reset input	I	L	Schmitt input with internal filter
Master Clock				
MCKI	Master clock input	I		Connected to external crystal (4 to 16 MHz)
MCKO	Master clock output	O		
PLLRC	PLL RC network input	I		
Real-time Clock				
RTCKI	32.768 kHz clock input	I		Connected to external 32.768 kHz crystal
RTCKO	32.768 kHz clock output	O		
UPIO				
UPIO[17:0]	Unified I/O	I/O (I)	(Z)	General purpose I/O
USART0				
SCK0/MPIO	USART0 clock line	I/O (I)	(Z)	Multiplexed with general purpose I/O
RXD0/MPIO	USART0 receive line	I/O (I)	(Z)	Multiplexed with general purpose I/O
TXD0/MPIO	USART0 transmit line	I/O (I)	(Z)	Multiplexed with general purpose I/O

Table 3-1. Pin Description (Continued)

Signal Name	Function	Type[1]	Level[1]	Comments
		USART1		
SCK1/MPIO	USART1 clock line	I/O (I)	(Z)	Multiplexed with general purpose I/O
RXD1/MPIO	USART1 receive line	I/O (I)	(Z)	Multiplexed with general purpose I/O
TXD1/MPIO	USART1 transmit line	I/O (I)	(Z)	Multiplexed with general purpose I/O
		USART2		
SCK2/MPIO	USART2 clock line	I/O (I)	(Z)	Multiplexed with general purpose I/O
RXD2/MPIO	USART2 receive line	I/O (I)	(Z)	Multiplexed with general purpose I/O
TXD2/MPIO	USART2 transmit line	I/O (I)	(Z)	Multiplexed with general purpose I/O
		Capture		
CAPT[1:0]/MPIO	Capture input	I/O (I)	(Z)	Multiplexed with general purpose I/O
		PWM		
PWM[3:0]/MPIO	Pulse Width Modulation output	I/O (I)	(Z)	Multiplexed with general purpose I/O
		Timer 0		
T0TIOA[2:0]/MPIO	Capture/waveform I/O	I/O (I)	(Z)	Multiplexed with a general purpose I/O
T0TIOB[2:0]/MPIO	Trigger/waveform I/O	I/O (I)	(Z)	Multiplexed with a general purpose I/O
T0TCLK[2:0]/MPIO	External clock/trigger/input	I/O (I)	(Z)	Multiplexed with a general purpose I/O
		ADC		
ANAIN[7:0]	Analog input	I		
VREFP	Positive voltage reference	I		
		SPI		
SPCK/MPIO	SPI clock line	I/O (I)	(Z)	Multiplexed with a general purpose I/O
MISO/MPIO	SPI master in slave out	I/O (I)	(Z)	Multiplexed with a general purpose I/O
MOSI/MPIO	SPI master out slave in	I/O (I)	(Z)	Multiplexed with a general purpose I/O
NPCS[3:1]/MPIO	SPI chip select	I/O (I)	(Z)	Multiplexed with a general purpose I/O
NPCS0/NSS/MPIO	SPI chip select (master and slave)	I/O (I)	(Z)	Multiplexed with a general purpose I/O
		CAN0		
CANRX0	CAN0 receive line	I	L	
CANTX0	CAN0 transmit line	O	L (H)	
		JTAG		
SCANEN	Scan enable (Factory test)	I	H	Internal pull-down (must be connected to GND or leave unconnected for normal operation)
TDI	Test Data In	I		Schmitt trigger, internal pull-up
TDO	Test Data Out	O		
TMS	Test Mode Select	I		Schmitt trigger, internal pull-up
TCK	Test Clock	I		Schmitt trigger, internal pull-up
TEST	Factory test	I	H	Internal pull-down (must be connected to GND or leave unconnected for normal operation)

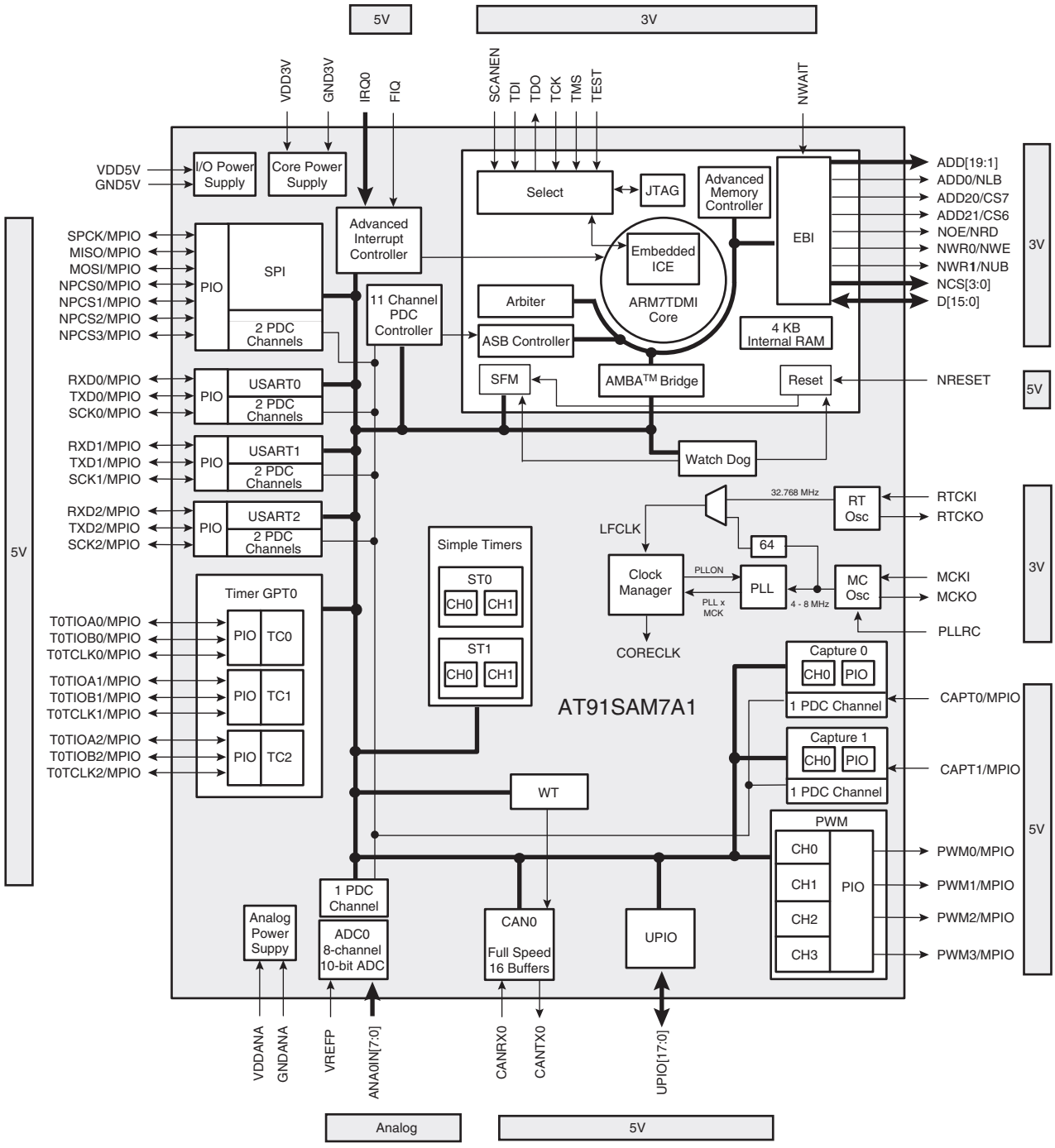
Table 3-1. Pin Description (Continued)

Signal Name	Function	Type[1]	Level[1]	Comments
Power Supply and Ground				
VDD3V (CORE)	3.3V for core	-		
GND3V (CORE)	Ground for core	-		
VDD3V (I/O+CORE)	3.3V for core and 3V I/O	-		
GND3V (I/O+CORE)	Ground for core and 3V I/O	-		
VDD3V (RTCK)	3.3V for RTCK oscillator	-		
GND3V (RTCK)	Ground for RTCK oscillator	-		
VDD3V (PLL)	3.3V for PLL and master oscillator	-		
GND3V (PLL)	Ground for PLL and master oscillator	-		
VDD3V (ANA)	3.3V for analog cells	-		
GND3V (ANA)	Analog Ground for analog cells	-		
VDD3V (I/O)	3.3V for functional I/O	-		
GND3V (I/O)	Ground for functional I/O	-		
VDD5V (I/O)	3.3V to 5V for functional I/O	-		
GND5V (I/O)	Ground for functional I/O	-		

Notes:
1. Values in brackets are values at reset: H (high level), L (low level), Z (tri-state), I (input), O (output).
2. The EBI bus (address bus A[21:0], data bus D[15:0] and control lines NOE/NRD, NWR0/NWE, NWR1/NUB and NCS[3:0]) is tri-stated when NRESET is at logical 0. This allows external equipment to access the external memory devices (e.g., for Flash programming). It is up to the application to add an external pull-up on the chip select lines in order to avoid EBI conflicts at reset.
3. The EBI data bus D[15:0] has an internal pull-down.

4. Block Diagram

Figure 4-1. AT91SAM7A1 Block Diagram



5. Product Overview

5.1 Register Considerations

5.1.1 Enable/Disable/Status Registers

In order to reduce code size and subsequently increase speed when accessing internal peripherals, most of the registers have been split into three address locations:

- The first address location (Enable or Set Register) is used to set a bit to a logical 1.
- The second address location (Disable or Clear Register) is used to set a bit to a logical 0.
- The third address location (Status register or Mask Register) gives the current state of the bit.

To set a bit to a logical 1 in the Status or Mask Register, a write command in the Enable or Set Register must be performed with the corresponding bit at a logical 1.

To set a bit to a logical 0 in the Status or Mask Register, a write command in the Disable or Clear Register must be performed with the corresponding bit at a logical 1.

5.1.2 Example

Supposing that the US0_PSR register value is 0x00000000. To enable the RXD and SCK pins as PIOs in the USART0 block, 0x00050000 must be written in the US0_PER register. The value read in the US0_PSR register will be 0x00050000.

Now if the software wants to disable the RXD pin as a PIO (i.e. enable it for USART0 use), a write access to the US0_PDR register with the value 0x00040000 must be performed. The new value read in the US0_PSR register will be 0x00010000.

5.1.3 Key Access to Registers

Some bits in registers can be set to a value (0 or 1) only if the right key is written at the same time.

5.1.3.1 Example 1

The TESTEN bit in the SFM_TM register can be set to a logical 0 or 1 only if the KEY[15:0] bits are equal to 0xD64A.

To enable test mode, 0xD64A0002 must be written in the SFM_TM register.

To disable test mode, 0xD64A0000 must be written in the SFM_TM register.

5.1.3.2 Example 2

To set the RTCKEN bit in the CM_CS register to logical 1, a write access to the CM_CE register must be done with a value of 0x23050080.

To set the RTCKEN bit in the CM_CS register to logical 0, a write access to the CM_CD register must be done with a value of 0x18070080.

5.2 Power Consumption

5.2.1 Working Modes

The AT91SAM7A1 microcontroller provides different working modes.

Table 5-1. Working Modes

Mode	Note
Low-power Mode (LPM)	The master clock oscillator, the PLL and the internal divider are switched off. The real time oscillator is enabled. The low frequency clock is selected from the real time oscillator and used as a system clock (i.e., 32.768 kHz used for GIC, WD, WT, ST and any peripheral needed for interrupt generation). CORECLK = RTCK, LFCLK = RTCK
Slow Mode (SLM)	The PLL is switched off. The system clock is the master clock (CORECLK = MCK) or the master clock divided by β (CORECLK = MCK/β, β in the range [2:256]).
Operational (OPE)	Master oscillator and PLL are enabled. The system clock is the clock from the PLL, CORECLK = α x MCK (α in the range [x2:x20])

5.2.2 Low-power Mode

Low-power mode is defined as the state in which:

- Master clock oscillator and PLL are stopped
- Low frequency oscillator (32.768 kHz) is used as an internal system clock for core and all peripherals (CORECLK = RTCK, LFCLK = RTCK)

The total power dissipation of the AT91SAM7A1 embedded system, when in low power mode, is estimated to be 170 µW maximum, at an operating voltage of 3.3V, over the operating temperature range. Additional conditions are: ARM core stopped, PDC stopped, all modules disabled except ST0, ST1, WT, WD and PMC working at 32.768 kHz.

5.2.3 Slow Mode

Slow mode is defined as the state in which:

- Master clock oscillator is enabled, divided by β (β in the range [2:256]) and used as the system clock (CORECLK = MCK or MCK/β)
- The low frequency clock can still be used as low frequency clock for peripherals (LFCLK = RTCK or MCK/β)

The total power dissipation of the AT91SAM7A1 embedded system, when in halt mode, is estimated to be 78 mW with CORECLK = MCK, at an operating voltage of 3.3V, over the operating temperature range and with ARM core and modules working at CORECLK frequency = 4 MHz. With CORECLK = MCK/64, total power dissipation is estimated at 4 mW, at an operating voltage of 3.3V, over the operating temperature range and with ARM core and modules working at CORECLK frequency = 62.5 kHz (i.e., 4 MHz/64).

5.2.4 Operational Mode

Operational mode is defined as the state in which:

- Master clock oscillator and PLL are enabled, system clock is taken from the PLL output (CORECLK = α x MCK, where α is in the range [2:20])
- The Low frequency clock can still be used as low frequency clock for peripherals (LFCLK = RTCK or MCK/β, β in the range [2:256])

The total power dissipation of the AT91SAM7A1 embedded system, when in operational mode, is estimated to be 605 mW maximum, at an operating voltage of 3.3V, over the operat-

ing temperature range and with ARM core and modules working at CORECLK frequency = 32 MHz (i.e., MCK = 4 MHz, PLL multiplier = 8).

5.3 Reset

The application must ensure a reset of at least 5 ms to allow time for the system clock to stabilize (CORECLK).

The 32.768 kHz clock (RTCK) will be stabilized 300 ms after the reset is asserted. Software should not use or program the peripherals (WD, WT, ST) which are using this clock until it is stabilized.

5.4 Electrical Characteristics

Table 5-2. AT91SAM7A1 Pin Connections for 144-lead LQFP Package

Pin	Name	Pad	Pin	Name	Pad	Pin	Name	Pad	Pin	Name	Pad
1	D0	PC3B01D	37	ADD11	PC3T02	73	GND5V (I/O)		109	ANA0IN1	AIMUX1
2	D8	PC3B01D	38	ADD12	PC3T02	74	PIOA2	MC5B04	110	ANA0IN2	AIMUX1
3	D1	PC3B01D	39	ADD13	PC3T02	75	PIOA3	MC5B04	111	ANA0IN3	AIMUX1
4	D9	PC3B01D	40	ADD14	PC3T02	76	VDD5V (I/O)		112	ANA0IN4	AIMUX1
5	VDD3V (I/O)		41	ADD15	PC3T02	77	PIOA4	MC5B03	113	ANA0IN5	AIMUX1
6	GND3V (I/O+CORE)		42	GND3V (IO)		78	PIOA5	MC5B03	114	ANA0IN6	AIMUX1
7	VDD3V (I/O+CORE)		43	VDD3V (CORE)		79	PIOA6	MC5B03	115	ANA0IN7	AIMUX1
8	D2	PC3B01D	44	VDD5V (IO)		80	PIOA7	MC5B03	116	GND3V (ANA)	
9	D10	PC3B01D	45	IRQ0	MC5D00	81	PIOA8	MC5B03	117	VDD3V (PLL)	
10	D3	PC3B01D	46	FIQ	MC5D00	82	PIOA9	MC5B03	118	MCKI	OSC16M
11	D11	PC3B01D	47	T0TIOA0/MPIO	MC5B01	83	GND5V (I/O)		119	MCKO	OSC16M
12	D4	PC3B01D	48	T0TIOB0/MPIO	MC5B01	84	PIOA10	MC5B02	120	PLLRC	PLL080M1
13	D12	PC3B01D	49	T0TCLK0/MPIO	MC5B01	85	PIOA11	MC5B02	121	GND3V (PLL)	
14	D5	PC3B01D	50	T0TIOA1/MPIO	MC5B01	86	PIOA12	MC5B01	122	VDD3V (RTCK)	
15	D13	PC3B01D	51	T0TIOB1/MPIO	MC5B01	87	PIOA13	MC5B01	123	RTCKI	OSC33K
16	D6	PC3B01D	52	T0TCLK1/MPIO	MC5B01	88	PIOA14	MC5B01	124	RTCKO	OSC33K
17	D14	PC3B01D	53	T0TIOA2/MPIO	MC5B01	89	PIOA15	MC5B01	125	GND3V (RTCK)	
18	D7	PC3B01D	54	T0TIOB2/MPIO	MC5B01	90	PIOA16	MC5B01	126	VDD3V (I/O)	
19	D15	PC3B01D	55	GND5V (I/O)		91	PIOA17	MC5B01	127	GND3V (CORE)	
20	ADD17	PC3T02	56	T0TCLK2/MPIO	MC5B01	92	PWM0/MPIO	MC5B01	128	GND3V (I/O)	
21	ADD16	PC3T02	57	TXD0/MPIO	MC5B01	93	VDD5V (I/O)		129	SCANEN	PC3D01D
22	NWR0/NWE	PC3B02	58	RXD0/MPIO	MC5B01	94	PWM1/MPIO	MC5B01	130	TEST	PC3D01D
23	ADD19	PC3T02	59	SCK0/MPIO	MC5B01	95	PWM2/MPIO	MC5B01	131	TMS	PC3D21U
24	ADD18	PC3T02	60	TXD1/MPIO	MC5B01	96	PWM3/MPIO	MC5B01	132	TDO	PC3T03
25	ADD7	PC3T02	61	RXD1/MPIO	MC5B01	97	CAPT0/MPIO	MC5B01	133	TDI	PC3D21U
26	ADD6	PC3T02	62	SCK1/MPIO	MC5B01	98	CAPT1/MPIO	MC5B01	134	TCK	PC3D21U
27	GND3V (I/O+CORE)		63	VDD5V (I/O)		99	NRESET	MC5D20	135	NWAIT	PC3D01U
28	VDD3V (I/O+CORE)		64	SPCK/MPIO	MC5B01	100	CANRX0	MC5D00	136	ADD21/CS6	PC3T02
29	ADD2	PC3T02	65	MISO/MPIO	MC5B01	101	CANTX0	MC5O01	137	NCS3	PC3T02
30	ADD3	PC3T02	66	MOSI/MPIO	MC5B01	102	TXD2/MPIO	MC5B01	138	NCS2	PC3T02
31	ADD4	PC3T02	67	NPCS0/NSS/MPIO	MC5B01	103	RXD2/MPIO	MC5B01	139	NWR1/NUB	PC3B02
32	ADD5	PC3T02	68	NPCS1/MPIO	MC5B01	104	SCK2/MPIO	MC5B01	140	ADD0/NLB	PC3T02
33	ADD8	PC3T02	69	NPCS2/MPIO	MC5B01	105	GND5V (I/O)		141	NCS1	PC3T02
34	ADD20/CS7	PC3T02	70	NPCS3/MPIO	MC5B01	106	VDD3V (ANA)		142	NOE/NRD	PC3B02
35	ADD9	PC3T02	71	PIOA0	MC5B04	107	VREFP0	ANAIN	143	NCS0	PC3T02
36	ADD10	PC3T02	72	PIOA1	MC5B04	108	ANA0IN0	AIMUX1	144	ADD1	PC3T02

Note: Pins 7, 28 and 43, i.e. VDD3V (CORE) and VDD3V (I/O + CORE), are internally connected together.

Note: Pins 6, 27 and 127, i.e. GND3V (CORE) and GND3V (I/O + CORE), are internally connected together.

Note: Pins K6, K4 and H4, i.e. VDD3V (CORE) and VDD3V (I/O + CORE), are internally connected together.

Note: Pins G4, J4 and D6, i.e. GND3V (CORE) and GND3V (I/O + CORE), are internally connected together.

Pad types are given in Table 5-3 below.

Table 5-3. Pad Types

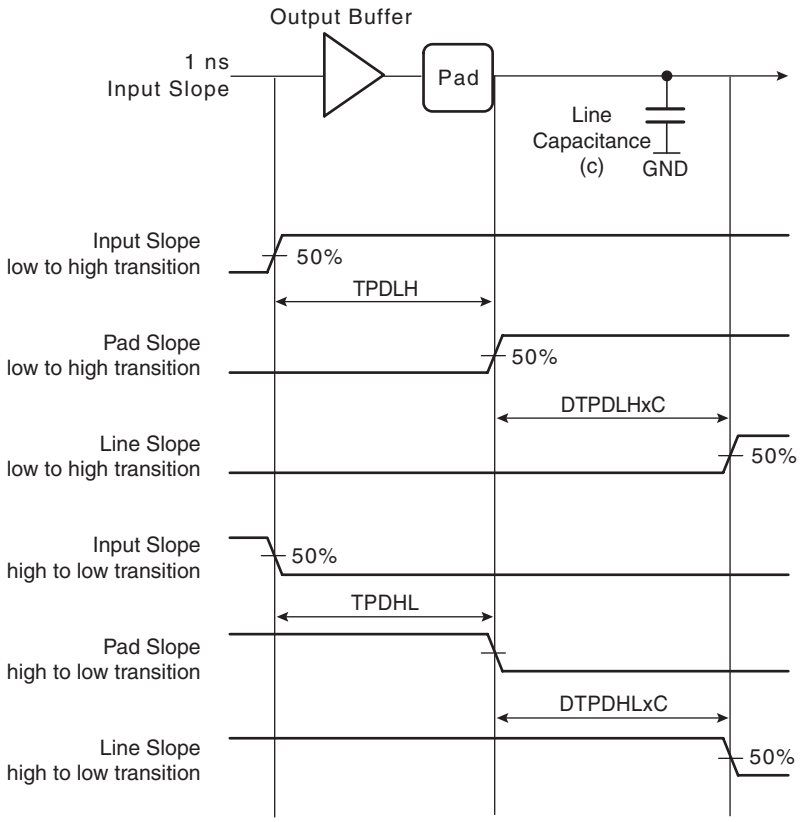
Pad	Type	DTPDHL[1]	DTPDLH[2]	TPDHL[3]	TPDLH[4]	Output Current
MC5B01	5 V CMOS bidirectional pad	0.144 ns/pF	0.131 ns/pF	2.327 ns	2.192 ns	2 mA AC 2 mA DC
MC5B02	5 V CMOS bidirectional pad	0.072 ns/pF	0.066 ns/pF	2.298 ns	2.179 ns	4 mA AC 4 mA DC
MC5B03	5 V CMOS bidirectional pad	0.036 ns/pF	0.033 ns/pF	2.727 ns	2.034 ns	8 mA AC 8 mA DC
MC5B04	5 V CMOS bidirectional pad	0.018 ns/pF	0.017 ns/pF	3.265 ns	2.449 ns	16 mA AC 16 mA DC
MC5O01	5 V CMOS output pad	0.144 ns/pF	0.131 ns/pF	2.310 ns	2.174 ns	2 mA AC 2 mA DC
MC5D00	5 V CMOS non-inverting input pad					
MC5D20	5 V CMOS schmitt non-inverting input pad					
PC3D01D	3 V CMOS non-inverting input pad with pull-down resistor					
PC3D01U	3 V CMOS non-inverting input pad with pull-up resistor					
PC3D21	3 V CMOS schmitt non-inverting input pad					
PC3D21U	3V CMOS schmitt non-inverting input pad with pull-up resistor					
PC3T01	3 V CMOS three state output pad	0.120 ns/pF	0.116 ns/pF	1.357 ns	1.011 ns	2 mA AC 0.3 mA DC
PC3T02	3 V CMOS three state output pad	0.060 ns/pF	0.058 ns/pF	1.002 ns	0.781 ns	4 mA AC 0.3 mA DC
PC3T03	3 V CMOS three state output pad	0.040 ns/pF	0.039 ns/pF	0.943 ns	0.800 ns	6 mA AC 0.3 mA DC
PC3B01D	3 V CMOS bidirectional pad with pull-down resistor	0.118 ns/pF	0.116 ns/pF	1.357 ns	1.040 ns	2 mA AC 0.3 mA DC
PC3B01	3 V CMOS non-inverting bidirectional pad	0.120 ns/pF	0.116 ns/pF	1.372 ns	1.033 ns	2 mA AC 0.3 mA DC
PC3B02	3 V CMOS non-inverting bidirectional pad	0.060 ns/pF	0.058 ns/pF	1.010 ns	0.789 ns	6 mA AC 0.3 mA DC
PC3B03	3 V CMOS non-inverting bidirectional pad	0.040 ns/pF	0.039 ns/pF	0.948 ns	0.808 ns	6 mA AC 0.3 mA DC
OSCK33	32.768 kHz crystal oscillator pad					
OSC16M	2-6 MHz crystal oscillator pad					
PLL080M1	20 MHz to 80 MHz single pad Phase-Locked Loop					
AIMUX1	Analog input pad					

Notes:
1. Differential (load-dependent) propagation delay, high-to-low or high impedance-to-low (V_{DD} = 3.3 V, Temp. = 25°C, Input Slope = 1 ns)
2. Differential (load-dependent) propagation delay, low-to-high or high impedance-to-high (V_{DD} = 3.3 V, Temp. = 25°C, Input Slope = 1 ns)
3. Propagation delay, high-to-low (V_{DD} = 3.3 V, Temp. = 25°C, Input Slope = 1 ns)
4. Propagation delay, low-to-high (V_{DD} = 3.3 V, Temp. = 25°C, Input Slope = 1 ns)

5.4.1 Propagation Delay

The propagation delay time shown in Table 5-3, “Pad Types,” on page 12, is the time in nanoseconds from the 50% point of the input to the 50% point of the output.

Figure 5-1. Propagation Delay

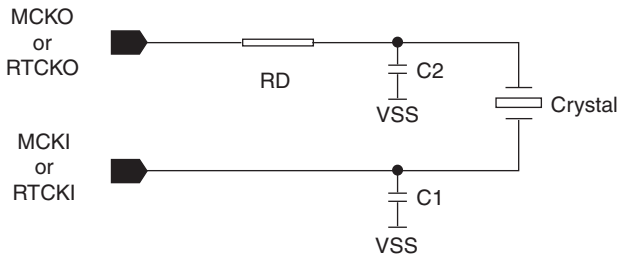


6. Clocks

6.1 Crystals

Crystals with 10 pF load capacitance can be directly connected to the oscillator pads. Nevertheless, it is recommended to implement the circuitry as described hereafter and in Figure 6-1 below.

Figure 6-1. Circuitry for 10 pF Load Capacitance



If the crystal recommended capacitor Cx is greater than 10 pF, then C1 and C2 must be added. Cx can be approximated to: Cx = (C1 x C2)/(C1 + C2).

Value of resistor RD depends on crystal frequency and manufacturer. Typical values of RD are given in Table 6-1 (values should be adjusted in the application environment).

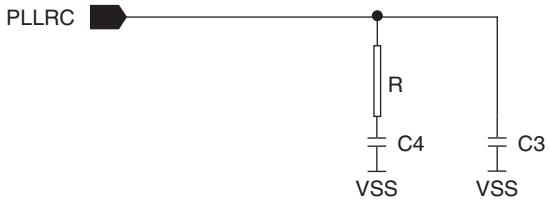
Table 6-1. Typical Crystal Series Resistor

Signal	RD	Conditions
MCKO	0 Ω	Crystal: CP12A-4MHz-S1-4085-1050 (NDK®)
RTCKO	10 kΩ	Crystal: MC-306 32.768K-A (EPSON®)

6.2 Phase Locked Loop

The AT91SAM7A1 microcontroller integrates a programmable PLL. The PLL requires an external RC network as described hereafter and in Figure 6-2 below.

Figure 6-2. External RC Network



The optimum response with a simple RC filter is obtained when:

Equation1:

$$0.4 < \sqrt{\left(\frac{K_0 \times I_P}{n \times (C_3 + C_4)}\right)} \times \frac{R \times C_4}{2} < 1 \text{ with an optimum value of } 0.707$$

Where:

- K_0 is the PLL V_{CO} gain (typ 105.10^6 Hz/V, min 65.10^6 Hz/V, max 172.10^6 Hz/V)
- I_P is the peak current delivered by the charge pump into the filter (typ. 350 µA, min. 50 µA, max. 800 µA)
- n is the division ratio of the divider (i.e., PLL multiplication factor)

Stability can be improved with an additional capacitor C_3. The value of C_3 must be chosen so that:

Equation 2:
$$4 < \frac{C_4}{C_3} < 15$$

Equation 3:
$$\sqrt{\left(\frac{K_0 \times I_p}{n \times (C_3 + C_4)}\right)} \leq \frac{\Pi \times f_{CKR}}{5}$$

Where:

- f_{CKR} is the PLL input frequency (i.e., MCK).

Phase jitter for the PLL is 200 ps typical.

6.2.1 PLL Characteristics

Table 6-2. PLL Characteristics

Code	Parameter	Conditions	Min	Typ	Max	Unit
f_{CKR}	Input frequency		0.02		30	MHz
f_{CK}	Output frequency		20		30	MHz
Wlow	Duty cycle			50		%
j_{CK}	Jitter	With ratio 1:1		200		ps
n	Division ratio		1:1		1:1024	
K_0	V_{CO} gain		65	105	172	MHz/V
I_P	CHP current		50	350	800	mA

6.3 Clock Timings

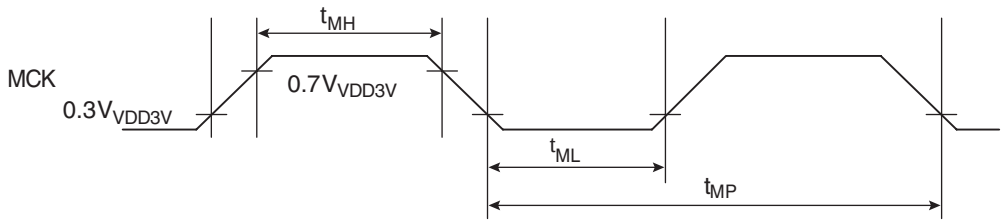
6.3.1 Master Clock

The master clock is the clock generated by the master clock oscillator. The master clock (MCK) characteristics are given in Table 6-3.

Table 6-3. Master Clock Timings

Symbol	Parameter	Minimum	Typical	Maximum	Unit
$1/t_{MP}$	Master oscillator frequency	4000		16000	kHz
t_{MP}	Master clock period	62.5		250	ns
t_{MH}	Master clock high time	0.40 x t_{MP}	0.50 x t_{MP}	0.60 x t_{MP}	ns
t_{ML}	Master clock low time	0.40 x t_{MP}	0.50 x t_{MP}	0.60 x t_{MP}	ns

Figure 6-3. Master Clock Waveform



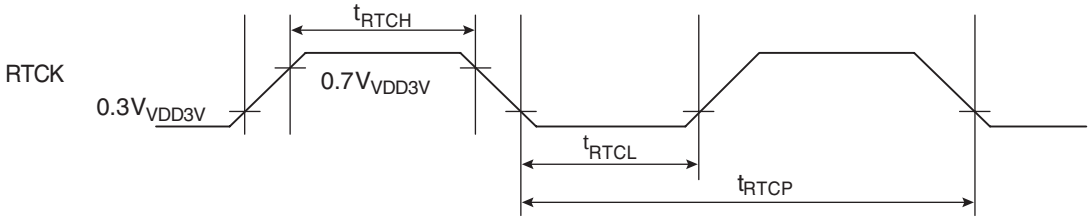
6.3.2 32.768 kHz Frequency Clock

The 32.768 kHz clock is the clock generated by the real time clock oscillator. The real time clock (RTCK) characteristics are given below in Table 6-4.

Table 6-4. Low Frequency Clock Timings

Symbol	Parameter	Minimum	Typical	Maximum	Unit
$1/t_{RTCP}$	32.768kHz oscillator frequency		32.768		kHz
t_{RTCP}	32.768kHz clock period		30517.58		ns
t_{RTCH}	32.768kHz clock high time	0.40 x t_{RTCP}	0.50 x t_{RTCP}	0.60 x t_{RTCP}	ns
t_{RTCL}	32.768kHz clock low time	0.40 x t_{RTCP}	0.50 x t_{RTCP}	0.60 x t_{RTCP}	ns
Dt_{RTCP}	Duty cycle (t_{RTCH}/t_{RTCP})	40	50	60	%

Figure 6-4. 32.768 kHz Clock Waveform



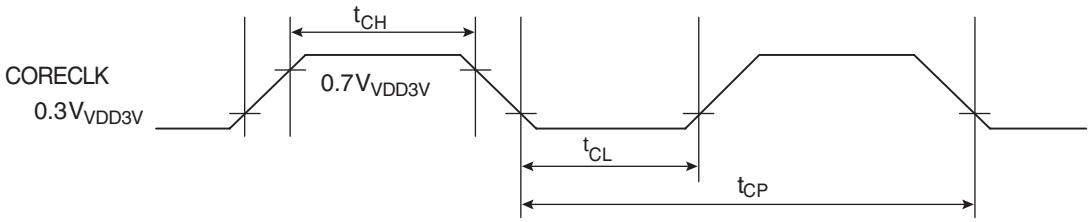
6.3.3 Core Clock

The core clock is the clock used in the system for the core and peripheral. The core clock (CORECLK) characteristics are given in Table 6-5.

Table 6-5. Core Clock Timings

Symbol	Parameter	Minimum		Maximum	Unit
$1/t_{CP}$	Core clock frequency	32.768		40000	kHz
t_{CP}	Core clock period	25		30517.58	ns
t_{CH}	Core clock high time	0.40 x t_{CP}	0.50 x t_{CP}	0.60 x t_{CP}	ns
t_{CL}	Core clock low time	0.40 x t_{CP}	0.50 x t_{CP}	0.60 x t_{CP}	ns
Dt_{CP}	Duty cycle (t_{CH}/t_{CP})	40	50	60	%

Figure 6-5. Core Clock Waveform



6.4 Internal Oscillator Characteristics

6.4.1 Core Clock Oscillator

Table 6-6. Core Clock Oscillator

Code	Parameter	Conditions	Min	Typ	Max	Unit
Du	Duty cycle	Crystal @ 4 MHz	40	50	60	%
Opf	Operating frequency		4		16	MHz
t_{SU}	Startup time	Crystal @ 4 MHz			10	ms
t_{SU}	Startup time	Crystal @ 8 MHz			5(1)	ms
C1	Internal capacitance (MCKI/GND)			10		pF
C2	Internal capacitance (MCKO/GND)			10		pF
CL	Equivalent load capacitance (MCKI/MCKO)			5		pF
DL	Drive level				50(1)	W
Rs	Equivalent Series Resistance	Fundamental @ 8 Mhz			100(1)	
Rs	Equivalent Series Resistance	Fundamental @ 4 Mhz			50(1)	
Cs	Shunt capacitance	Crystal			6	pF
CL	Load capacitance	Crystal @ 4 MHz		10(1)		pF
Cm	Motional capacitance	Crystal @ 4 MHz			3(1)	fF

Note: 1. These values are not characterized.

6.4.2 Real Time Clock Oscillator

Table 6-7. Real Time Clock Oscillator

Code	Parameter	Conditions	Min	Typ	Max	Unit
Du	Duty cycle	@ 32.768 kHz	40	50	60	%
tsu	Startup time				1.5	s
C1	Internal capacitance (RTCKI/GND)			20		pF
C2	Internal capacitance (RTCKO/GND)			20		pF
CL	Equivalent load capacitance (RTCKI/RTCKO)			10		pF
DL	Drive level				1	µW
Rs	Series resistance	Crystal			60	kOhm
Cs	Shunt capacitance	Crystal	0.8		1.7	pF
	Load capacitance	Crystal @ 32.768 kHz		10		pF
Cm	Motional capacitance	Crystal @ 32.768 kHz	1		4	fF

7. Memory Map

The AT91SAM7A1 microcontroller memory space is 4 Gbytes.

When the AT91SAM7A1D microcontroller is reset, the ARM core is in reboot mode to access the external memory (usually a ROM) on NCS0 at address 0x00000000. The internal RAM is located at address 0x00300000.

When the software execute the remap command (write 1 in RCB bit in AMC_RCR register), the internal RAM is automatically located at address 0x00000000 and the external memory accessed on the NCS0 is located in the memory space from 0x40000000 to 0x7FFFFFFF depending on the AMC_CSR0 register in the Advanced Memory Controller, then the chip is in remap mode.

7.1 Reboot Mode

Table 7-1. Internal Memory (Reboot Mode)

Memory Space	Size	Application	Abort Generation
0xFFFFFFFF 0xFFE00000	2 Mbytes	Peripheral devices	No
0xFFDFFFFF 0x00400000	4090 Mbytes	Reserved	Yes
0x003FFFFF 0x00300000	1 Mbytes (4 Kbytes repeated 256 times)	4 Kbytes internal RAM	No
0x002FFFFF 0x00200000	1 Mbytes	Reserved	No
0x001FFFFF 0x00100000	1 Mbytes	Reserved	Yes
0x000FFFFF 0x00000000	1 Mbytes	External memory on NCS0	No

7.2 Remap Mode

Table 7-2. Internal Memory (Remap Mode)

Memory Space	Size	Application	Abort Generation
0xFFFFFFFF 0xFFE00000	2 Mbytes	Peripheral devices	No
0xFFDFFFFF 0x80000000	2046 Mbytes	Reserved	Yes
0x7FFFFFFF 0x40000000	1024 Mbytes	Up to 6 external memories repeated within the page size programmed in the AMC_CSRx register	Yes, outside of defined page size in the AMC_CSRx
0x3FFFFFFF 0x00300000	1021 Mbytes	Reserved	Yes
0x002FFFFF 0x00100000	2 Mbytes	Reserved	No
0x000FFFFF 0x00000000	1 Mbytes (4 Kbytes repeated 256 times)	4 Kbytes internal RAM	No

7.3 External Memories

The AT91SAM7A1 external memories can be relocated in the address space from 0x40000000 to 0x7FFFFFFF. The configuration of the base address and the page size of each EBI chip select line (NCS[3:0], CS[7:6]) is done through the Advanced Memory Controller (AMC) registers.

It is to be noted that the two most significant bits of the base address are fixed to 01b allocating these memories in the second of the four Gbytes memory spaces.

The maximum external memory space is 16 Mbytes (i.e. CS[7:6] used as address lines).

Table 7-3. External Memory

Memory Space	Size	Application
0x($01XX_b$)XXFFFFF 0x($01XX_b$)XX00000	Up to 1 Mbytes	External memory on CS7
0x($01XX_b$)X1FFFFF 0x($01XX_b$)XX00000	Up to 2 Mbytes	External memory on CS6
0x($01XX_b$)X3FFFFF 0x($01XX_b$)XX00000	Up to 4 Mbytes	External memory on NCS3
0x($01XX_b$)X3FFFFF 0x($01XX_b$)XX00000	Up to 4 Mbytes	External memory on NCS2
0x($01XX_b$)X3FFFFF 0x($01XX_b$)XX00000	Up to 4 Mbytes	External memory on NCS1
0x($01XX_b$)X3FFFFF 0x($01XX_b$)XX00000	Up to 4 Mbytes	External memory on NCS0

7.4 Peripheral Resources

The peripheral modules of the AT91SAM7A1 embedded system are listed in Table 7-4.

Table 7-4. Peripheral Resources

Peripheral	Address	IRQ source	PDC Channel	PIO
AMC	0xFFE00000	-	-	
SFM	0xFFF00000	-	-	
Watchdog	0xFFFA0000	2	-	
Watch Timer	0xFFFA4000	3	-	
USART0	0xFFFA8000	4	RX: Ch0	3
			TX: Ch1	
USART1	0xFFFAC000	5	RX: Ch2	3
			TX: Ch3	
USART2	0xFFFB0000	6	RX: Ch4	3
			TX: Ch5	
SPI	0xFFFB4000	7	RX: Ch6	7
			TX: Ch7	
ADC0 (8-channel 10-bit)	0xFFFC0000	10	Ch10	
GPT0 (3 Channels)	0xFFFC8000	12	-	9
		13	-	
		14	-	
PWM (4 Channels)	0xFFFD0000	16	-	4
CAN (16 Channels)	0xFFFD4000	20	-	
UPIO	0xFFFD8000	21	-	19
Capture CAPT0	0xFFFDC000	22	Ch8	1
Capture CAPT1	0xFFFE0000	23	Ch9	1
Simple Timer ST0	0xFFFE4000	24	-	
Simple Timer ST1	0xFFFE8000	25	-	
CM	0xFFFEC000	-	-	
PMC	0xFFFF4000	-	-	
PDC	0xFFFF8000	-	-	
GIC	0xFFFFF000	-	-	

8. Power Management Block

In order to reduce power consumption, the AT91SAM7A1 microcontroller provides a power management block in some peripherals used to switch on/off the peripheral clocks (peripheral and PIO block).

This function is independent of the Power Management Controller (peripheral) used to switch on/off the ARM7TDMI core and the PDC clocks.

Three registers are provided:

- PERIPHERAL_ECR (at peripheral offset 0x0050) enables the clock
- PERIPHERAL_DCR (at peripheral offset 0x0054) disables the clock
- PERIPHERAL_PMSR (at peripheral offset 0x0058) gives the status of the clock

Two bits are provided in these registers:

- Bit 0 controls the PIO block of the peripheral
- Bit 1 controls the peripheral function

When the peripheral clock (and/or the PIO clock) is disabled, the clock is immediately stopped. When the clock is re-enabled, the peripheral controller (and/or the PIO controller) resumes action where it left off.

The interrupt registers are common to the peripheral controller and its PIO controller. The clock on the interrupt registers and its associated logic are stopped only if both the peripheral controller clock and the PIO controller clock are stopped.

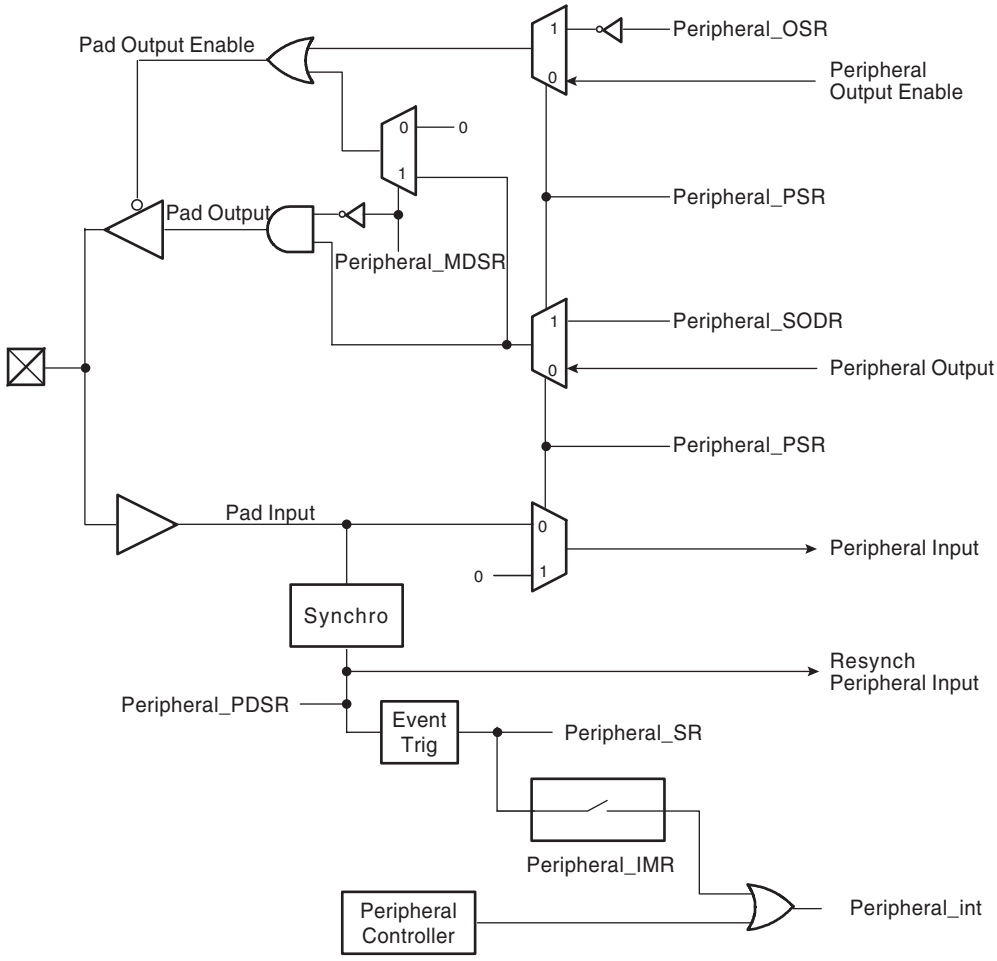
Table 8-1 lists the different power management blocks.

Table 8-1. AT91SAM7A1 Power Management Blocks

Module	Power Management Block Present	Module	Power Management Block Present
AMC	No	PWM	Yes
SFM	No	CAN	Yes
Watchdog	No	UPIO	Yes
Watch Timer	No	CAPT0	Yes
USART0	Yes	CAPT1	Yes
USART1	Yes	Simple Timer ST0	Yes
USART2	Yes	Simple Timer ST1	Yes
SPI	Yes	CM	No
ADC0	Yes	PMC	Yes
GPT0 TC0	Yes	PDC	No
GPT0 TC1	Yes	GIC	No
GPT0 TC2	Yes		

9. PIO Controller Block

Figure 9-1. PIO Controller Block Diagram



To match different applications, the AT91SAM7A1 peripherals have their dedicated pins multiplexed with general-purpose I/O pins (MPIO).

Table 9-1 lists the modules sharing the dedicated pins with MPIOs.

Table 9-1. PIO Block Multiplexing

Module	PIO Block Present	Number of MPIO	Name of PIO Lines
AMC	No	-	-
SFM	No	-	-
Watchdog	No	-	-
Watch Timer	No	-	-
USART0	Yes	3	TXD0, RXD0, SCK0
USART1	Yes	3	TXD1, RXD1, SCK1
USART2	Yes	3	TXD2, RXD2, SCK2

Table 9-1. PIO Block Multiplexing (Continued)

Module	PIO Block Present	Number of MPIO	Name of PIO Lines
SPI	Yes	7	MISO, MOSI, SPCK, NPCS[3:0]
ADC0	No	-	-
GPT0 TC0	Yes	3	TIOA0, TIOB0, TCLK0
GPT0 TC1	Yes	3	TIOA1, TIOB1, TCLK1
GPT0 TC2	Yes	3	TIOA2, TIOB2, TCLK2
PWM	Yes	4	PWM[3:0]
CAN	No	-	-
UPIO	Yes	18	UPIO[17:0]
CAPT0	Yes	1	CAPT0
CAPT1	Yes	1	CAPT1
Simple Timer ST0	No	-	-
Simple Timer ST1	No	-	-
CM	No	-	-
PMC	No	-	-
PDC	No	-	-
GIC	No	-	-

Each PIO block in the peripheral is controlled through the peripheral interface. The PIO block clock is enabled/disabled by the peripheral Power Management Controller (see Table 7-4 on page 22).

9.1 Multiplexed I/O Lines

All I/O lines are multiplexed with an I/O signal of the peripheral. After reset, the pin is controlled by the peripheral PIO controller. 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 9-1 on page 24 shows the multiplexing of the peripheral signals with the PIO controller signal.

Each pin of the peripheral can be independently controlled using the Peripheral_PER (PIO Enable) and Peripheral_PDR (PIO Disable) registers.

The Peripheral_PSR (PIO Status) indicates whether the pin is controlled by the peripheral or by the PIO controller block.

9.1.1 Output Selection

The user can select the direction of each individual I/O signal (input or output) using the Peripheral_OER (Output Enable) and Peripheral_ODR (Output Disable) registers. The output status of the I/O signal can be read in the Peripheral_OSR (Output Status) register. The direction defined has effect only if the pin is configured to be controlled by the PIO controller block.

9.1.2 I/O Levels

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

If a pin is controlled by the PIO controller block and is defined as an output (see Output Selection above), the level is programmed using the Peripheral_SODR (Set Output Data) and Peripheral_CODR (Clear Output Data) registers. In this case, the programmed value can be read in the Peripheral_ODSR (Output Data Status) register.

If a pin is controlled by the PIO controller block 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 block, the state of the pin is defined by the Peripheral controller. In all cases, the level on the pin can be read in the Peripheral_PDSR (Pin Data Status) register.

9.1.3 Interrupts

Each PIO controller block also provides an internal interrupt signal shared with the peripheral interrupt.

Each PIO can be programmed to generate an interrupt when a level change occurs. This is controlled by the Peripheral_IER (Interrupt Enable) and Peripheral_IDR (Interrupt Disable) registers which enable/disable the I/O interrupt (and the peripheral interrupts) by setting/clearing the corresponding bit in the Peripheral_IMR.

When a change in level occurs, the corresponding bit in the Peripheral_SR (Interrupt Status) register is set whether the pin is used as a PIO or a peripheral signal and whether it is defined as input or output.

If the corresponding interrupt in Peripheral_IMR (Interrupt Mask) register is enabled, the PIO interrupt is asserted.

The PIO interrupt is cleared when:

- a write access is performed on the Peripheral_CISR register (with the corresponding bit set at a logical 1) or
- a read access is performed in the Peripheral_SR register (if no Peripheral_CISR register is present in the peripheral)

9.1.4 User Interface

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

9.1.5 Open Drain/Push-pull Output

The PIO can either be configured as an open drain output (only drives a low level) or as a push pull output (drives high and low levels).

When the PIO is configured as open drain (multidriver), an external pull-up is necessary to guarantee a logic level (logical one) when the pin is not being driven.

The PERIPHERAL_MDER (Multidriver Enable) and PERIPHERAL_MDDR (Multidriver Disable) registers control this option and respectively configure the I/O as open drain or push pull. The multidriver option can be selected whether the I/O pin is controlled by the PIO controller or the peripheral controller. Bits at logical one in the PERIPHERAL_MDSR (Multidriver Status) indicate pins configured as open drain.

10. Advanced Memory Controller (AMC)

The AT91SAM7A1 microcontroller is provided with an Advanced Memory Controller allowing the software to configure external and internal memory mapping (at boot level).

The external 16-bit data bus interface is called the External Bus Interface (EBI) and is the physical layer used to connect external devices to the AT91SAM7A1 microcontroller. Subsequently, the EBI generates the signals which control the access to the external memory or peripheral devices.

The EBI is fully programmable through the Advanced Memory Controller (AMC) and can address up to 16 Mbytes. It has up to six chip selects and a 22-bit address bus.

The AT91SAM7A1 can only boot on a 16-bit external memory device connected to the NCS0 signal. All the other chip select lines (NCS[3:1] and CS[7:6]) can be configured to access 8- or 16-bit memory devices.

10.1 Boot on NCS0

By default, the AT91SAM7A1 boots on a 16-bit external memory device connected on NCS0.

At reset, access through NCS0 is configured as follows (in the AMC_CSR0 register):

- 8 wait states (WSE = 1, NWS = 7 in AMC_CSR0)
- 16-bit data bus width (DBW[1:0] = 01_b)
- Base address is at 0x00000000
- Byte access type is configured as Byte Write Access, BAT = 0
- The number of data float time is 0 (TDF[2:0] = 000_b)
- The EBI is configured in normal read protocol (DRP = 0 in AMC_MCR register)

The user can modify the chip select 0 configuration, programming the AMC_CSR0 with exact boot memory characteristics. The base address becomes effective after the remap command (set to a logical 1 the RCB in AMC_RCR), but the other parameters are changed immediately after the write access in the AMC_CSR0 register.

10.2 External Memory Mapping

The memory map associates the internal 32-bit address space with the external 22-bit address bus.

The memory map is defined by programming the base address and page size of the external memories.

If the physical memory device is smaller than the programmed page size, it wraps around and appears to be repeated within the page. The AMC correctly handles any valid access to the memory device within the page.

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).

The AT91SAM7A1 microcontroller must be wired so the NCS0 accesses a non volatile 16-bit memory as shown in Figure 10-6 on page 32 or Figure 10-7 on page 32.

10.3 External Memory Device Connection

10.3.1 Data Bus Width

Each chip select can access 8- or 16-bit data bus devices. This option is selected by the DBW[1:0] bits in the corresponding AMC_CSRx register.

NCS0 is used at reset to access a 16-bit memory device (DBW[1:0] = 01_b).

10.3.2 Byte Select or Byte Write Access

Each chip select can operate with one of two different types of write access by setting the Byte Access Type (BAT) bit in the corresponding AMC_CSRx register.

- Byte select access (BAT = 1): Uses one write signal, one read signal, and two signals to select upper and/or lower memory bank in a 16-bit memory.

  Typically used with 16-bit memories, except when the user wants to connect 2 x 8-bit memories in parallel. In this case, this is considered a 16-bit memory by the AMC.

- Byte write access (BAT = 0): Uses two byte write signals to select two different 8-bit devices and a single read signal. This mode is used at reset to boot on the memory connected on NCS0 (Chip Select 0).

  Typically used with 2 x 8-bit memories.

10.3.3 Byte Select Access (BAT = 1)

This mode is selected by setting the bit BAT to 1 in AMC_CSRx registers and is typically used to connect 16-bit devices in a memory page, except when user wants to connect 2 x 8-bit devices in parallel, in that case seen by the AMC this is a 16-bit memory page.

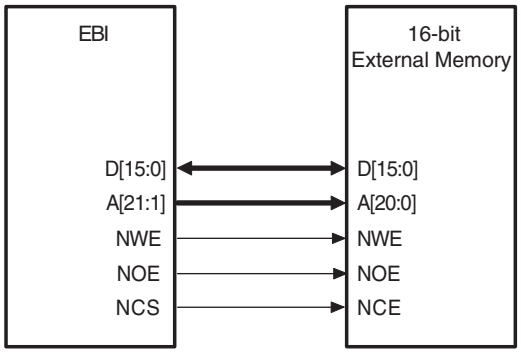
Users can use the upper/lower bank selection signals NUB and NLB to have either an 8-bit or a 16-bit access.

10.3.3.1 16-bit Access Device Connection

A typical 16-bit memory (e.g., Flash memory) device connection with 16-bit access is shown in Figure 10-1.

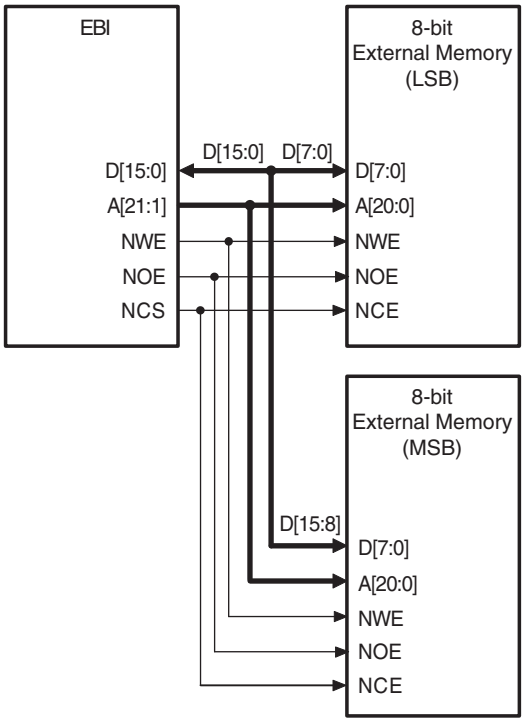
- The signal A0/NLB is not used
- The signal NWR1/NUB is not used
- The signal NWR0/NWE is used as NWE and enables half-word writes.
- The signal NRD/NOE is used as NOE and enables half-word reads.

Figure 10-1. EBI Connection for External 16-bit Memory Device, 16-bit Access Only



In the same configuration as above, Figure 10-2 shows how to connect 2 x 8-bit memory devices with 16-bit access.

Figure 10-2. EBI Connection for 2 x 8-bit Memory Devices, 16-bit Access Only



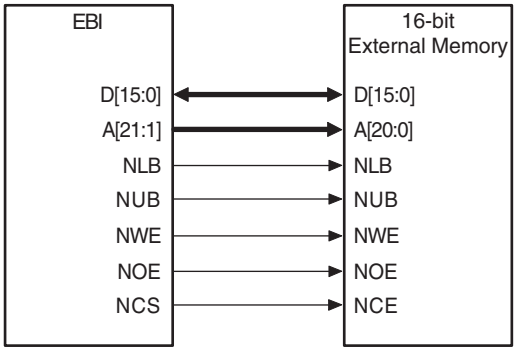
10.3.3.2 8-bit or 16-bit Access Device Connection

A typical 16-bit memory (e.g., SRAM) device connection with 8-bit or 16-bit access is shown in Figure 10-3.

The 16-bit memory allows upper/lower bank selection and NUB, NLB are used to have an 8-bit access.

- 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 half-word or byte writes.
- The signal NRD/NOE is used as NOE and enables half-word and byte reads.

Figure 10-3. EBI Connection for External 16-bit Memory Devices, 8-bit or 16-bit Access



10.3.4 Byte Write Access (BAT = 0)

This mode is selected by setting the bit BAT to 0 in AMC_CSRx registers and is typically used to connect 2 x 8-bit devices as a 16-bit memory page. This is the mode selected at reset on NCS0.

In this mode, users can interface the EBI with one or two 8-bit memories.

If the EBI is interfaced with two 8-bit memories, the users have the choice of either an 8-bit or a 16-bit access.

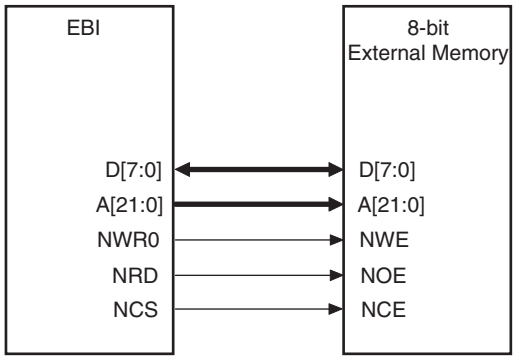
10.3.4.1 8-bit Access Device Connection

A typical 8-bit memory device connection with 8-bit access is shown in Figure 10-4.

DBW[1:0] should be for a 8-bit-data bus width and only NWR0 is used

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

Figure 10-4. EBI Connection for External 8-bit Memory Device, 8-bit Access Only



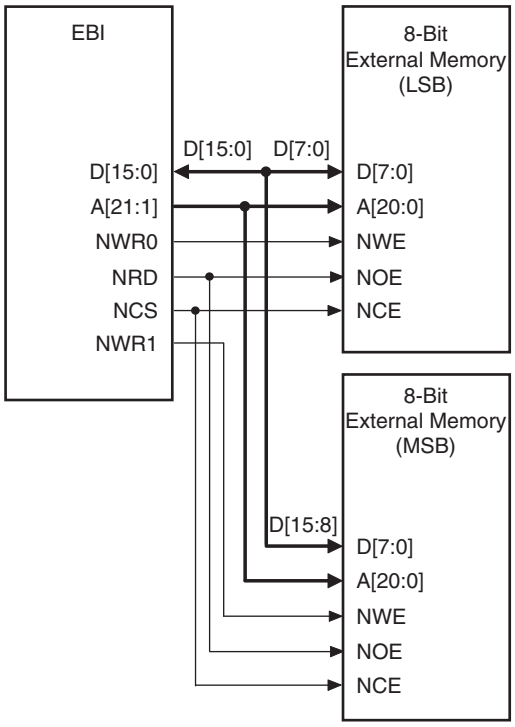
10.3.4.2 8-bit or 16-bit Access Device Connection

A typical 2 x 8-bit memory device connection with 8-bit or 16-bit access is shown in Figure 10-5.

- 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 10-5. EBI Connection for External 2x8-bit Memory Devices, 8-bit or 16-bit Access



10.3.4.3 16-bit Access Device Connection

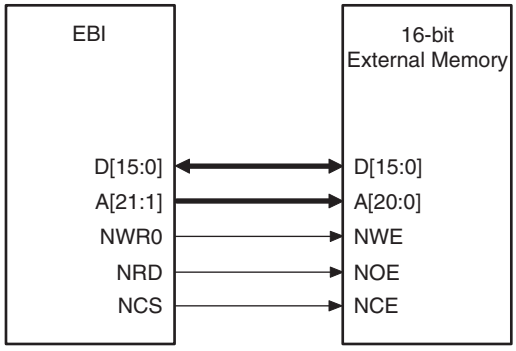
A typical 16-bit memory device connection with 16-bit access only is shown in Figure 10-6.

In this case, the AT91SAM7A1 is in byte write access mode and boots on the 16-bit memory. NWR1 and NWR0 are used by the EBI but only NWR0 is used by the memory, enabling a 16-bit access.

The correct mode for this configuration is byte select access and should be set in the boot.

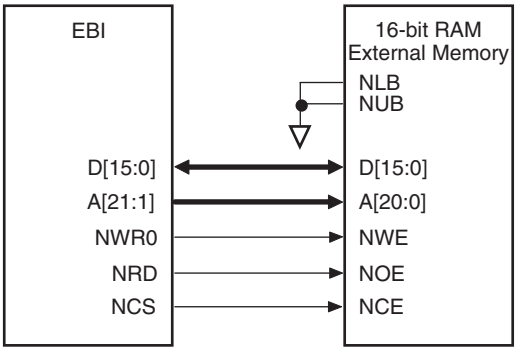
- The signal A0/NLB is not used.
- The signal NWR1/NUB is not used.
- The signal NWR0/NWE is used as NWR0 and enables half-word writes.
- The signal NRD/NOE is used as NRD and enables half-word and byte reads.

Figure 10-6. EBI Connection for External 16-bit Memory Devices, 16-bit Access Only



If users want to boot on a RAM memory for debug purposes, the RAM memory should be connected the same way as a Flash memory (NUB and NLB of the RAM memory connected to the ground) to emulate a pure 16-bit Flash memory as shown in Figure 10-7.

Figure 10-7. EBI Connected to an External 16-bit RAM Memory Device, 16-bit Access Only Used as a Boot Memory for Debug Purpose



10.4 External Bus Interface Timings

Simple read and write access cycles are explained in detail where read access can be done through two modes:

- Standard read protocol
- Early read protocol which increases the EBI performance for read access.

The EBI can automatically insert wait states during the external access cycles. These wait states are applied within the actual access cycle.

Data float wait states can also be inserted and applied between cycles. Data float wait states depend on the previous access.

10.4.1 Read Access

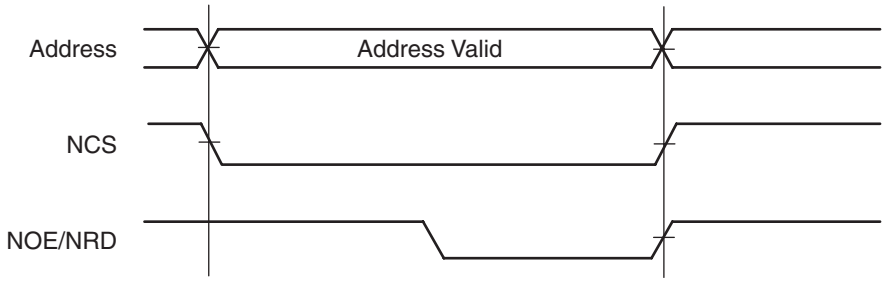
10.4.1.1 Standard Read Protocol

Standard read protocol (default read mode) implements a read cycle in which NRD/NOE is active during the second half of the read cycle.

The first half of the read 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 address is valid at the beginning of the access while NRD/NOE goes low only in the second half the read cycle to avoid bus conflict.

Figure 10-8. Standard Read Address

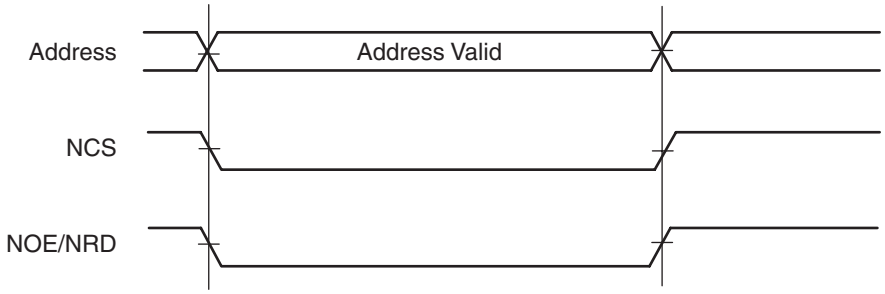


10.4.1.2 Early Read Protocol

Early read protocol provides more memory access time for a read access by asserting NRD at the beginning of the read cycle. This mode is selected by setting the bit DRP in AMC_MCR register.

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 timing of the EBI to be used. However, an extra data float wait state is required in some cases to avoid contentions on the external bus.

Figure 10-9. Early Read Address



10.4.2 Write Access

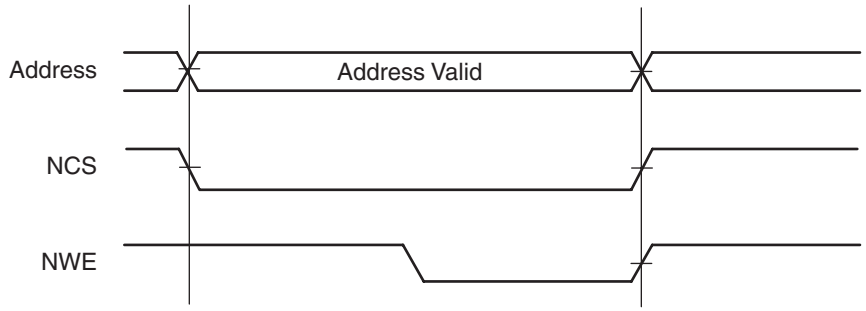
In a write access cycle, NWE (or NWR0, NWR1) is active during the second half of the write cycle.

The first half of the write cycle allows time to ensure completion of the previous access as well as the address and NCS set up time before NWE (or NWR0, NWR1) is asserted.

During an external write memory access, NCS is set low and address is valid at the beginning of the access while NWE (or NWR0, NWR1) goes low only in the second half of the write cycle to avoid bus conflict.

NWE (or NWR0, NWR1) goes high at the end of the write cycle unless wait states are asserted.

Figure 10-10. Write Access



10.4.3 Wait States

The EBI can automatically insert wait states during the external access cycles. These wait states are applied within the actual access cycle.

Different types of wait states are possible:

- Standard wait states
- Data float wait states
- External wait states

10.4.3.1 Standard Wait State

Each chip select line can be programmed to insert one or more wait states during an external access. This is done by setting the WSE bit in the corresponding AMC_CSRx register. The number of cycles to insert is programmed in the NWS[2:0] field in the same register.

Wait states are inserted within cycles and delay the read and write access cycles.

- Wait state with read cycle

The read cycle is delayed one cycle for each wait state programmed. In early mode, NOE/NRD goes low at the start of the read cycle. In standard mode, this signal goes low at the half of the first cycle.

Figure 10-11. Read Cycle with One Wait State

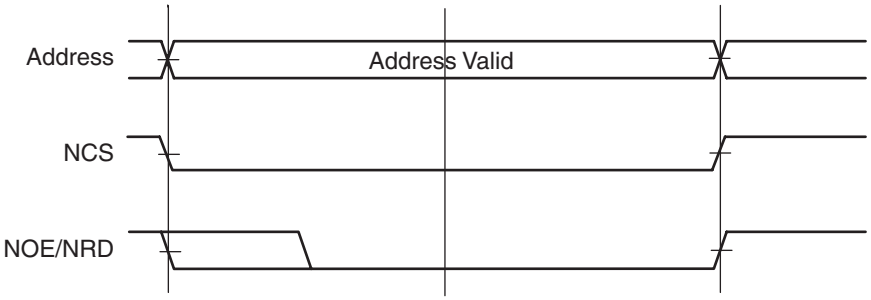
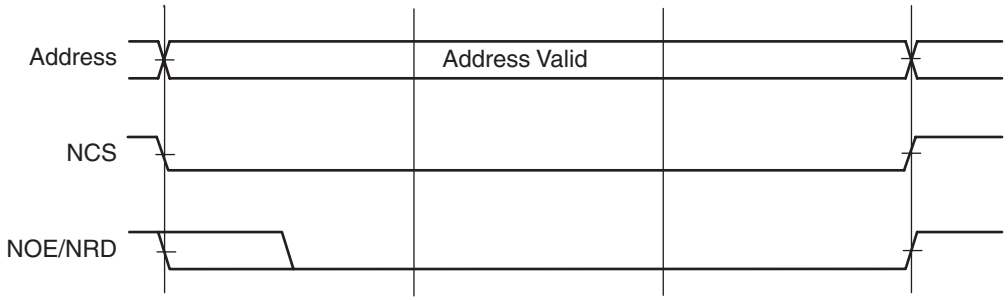


Figure 10-12. Read Cycle with Two Wait States



- Wait state with write cycle

The write cycle is delayed one cycle for each wait state programmed. NWE (or NWR0, NWR1) goes high one half cycle before the end of the write cycle.

Figure 10-13. Write Cycle with One Wait State

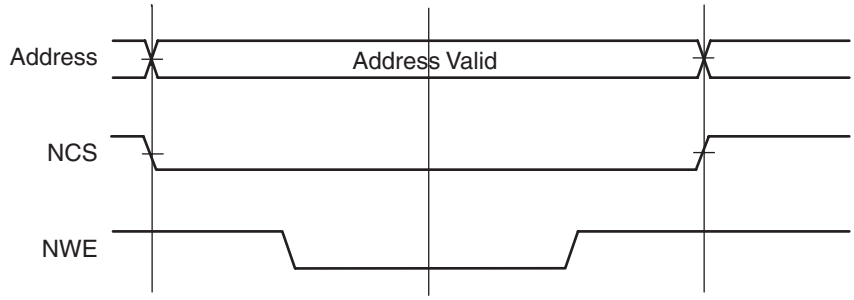
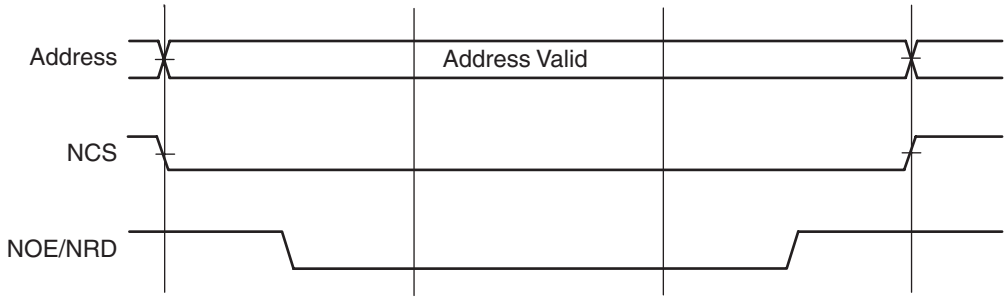


Figure 10-14. Write Cycle with Two Wait States



10.4.3.2 Data Float Wait State

Data float wait states are added to avoid data bus conflict.

After a read access, data float wait states allow more time for the external memory to release the data bus.

After a write access, data float wait states allow more time for the EBI to release the data bus.

The Data Float Output time (t_{DF}) for each external memory device is programmed in the TDF field of the AMC_CSRx register for the corresponding chip select. The value (0 - 7 clock cycles) indicates the number of data float wait states to be inserted.

Data float wait states are asserted between accesses.

Data float wait state insertion depends on the previous access.

Table 10-1 describes the data float wait states applied between external access cycles.

Table 10-1. Data Float States Applied

Previous Access	Next Access	Number of Data Float Wait States Applied	
		Early Read Mode	Standard Read Mode
NCSx Read	NCSx Read	0	0
NCSx Read	NCSx Write	nTDF	nTDF
NCSx Write	NCSx Read	1	0
NCSx Write	NCSx Write	0	0
NCSx Read	NCSy Read	Max(1, nTDFx)	Max(1, nTDFx)
NCSx Read	NCSy Write	Max(1, nTDFx)	Max(1, nTDFx)
NCSx Write	NCSy Read	1	1
NCSx Write	NCSy Write	1	1

Table 10-2. Examples

Previous Access	Next Access	Early Read Mode				Standard Read Mode			
		TDFx = 0	TDFx = 1	TDFx = 2	TDFx = 3	TDFx = 0	TDFx = 1	TDFx = 2	TDFx = 3
NCSx Read	NCSx Read	0	0	0	0	0	0	0	0
NCSx Read	NCSx Write	0	1	2	3	0	1	2	3
NCSx Write	NCSx Read	1	1	1	1	0	0	0	0
NCSx Write	NCSx Write	0	0	0	0	0	0	0	0
NCSx Read	NCSy Read	1	1	2	3	1	1	2	3
NCSx Read	NCSy Write	1	1	2	3	1	1	2	3
NCSx Write	NCSy Read	1	1	1	1	1	1	1	1
NCSx Write	NCSy Write	1	1	1	1	1	1	1	1

Illustrations from Figure 10-15 on page 37 to Figure 10-29 on page 41 give a complete description of how data float wait states apply.

Figure 10-15. Read and Write Access on Different Chip Select with NTDF = 0 or 1

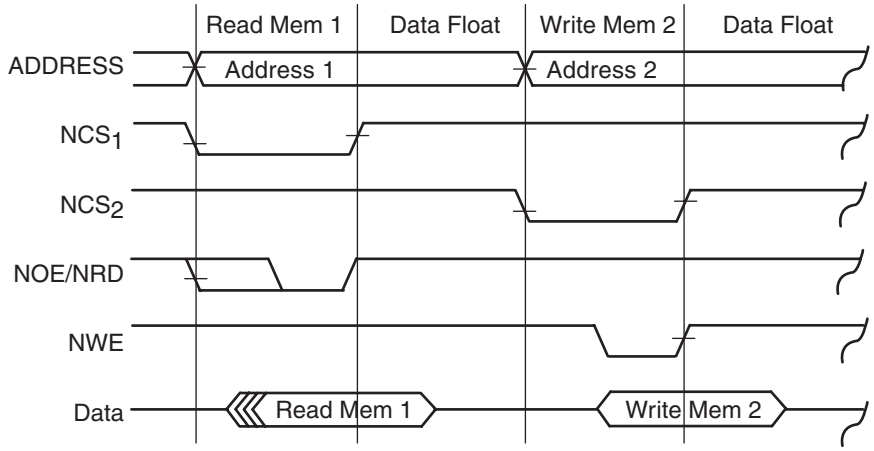


Figure 10-16. Read and Write Access on Different Chip Select with NTDF = 2

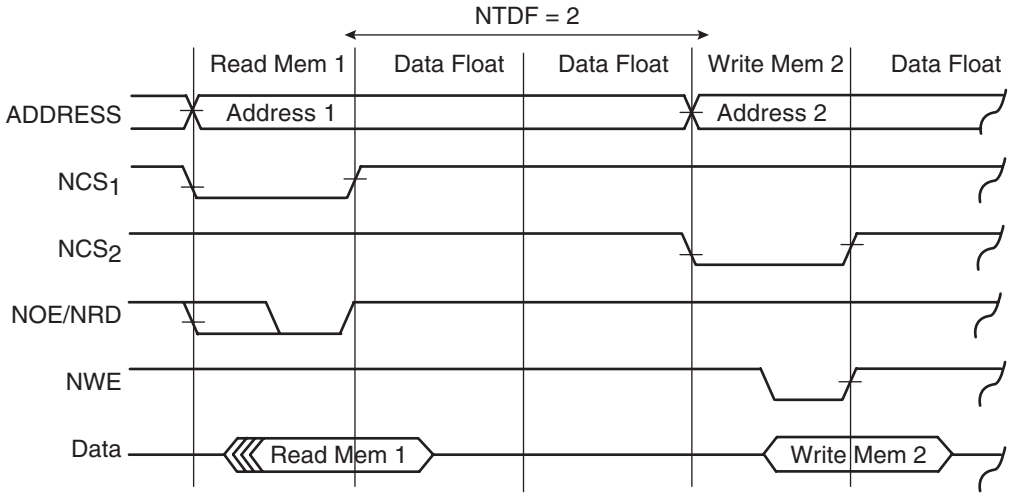


Figure 10-17. Standard Read and Write Access on the Same Chip Select with NTDF = 2

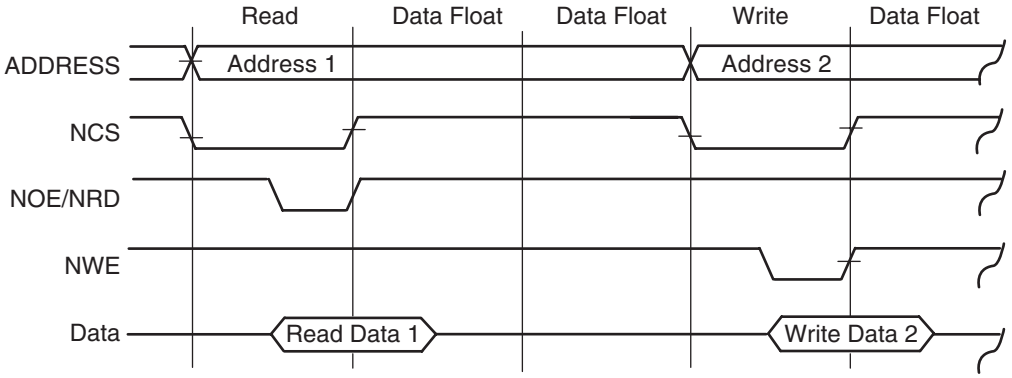


Figure 10-18. Sequential Early Read Access on the Same Chip Select with One Wait State

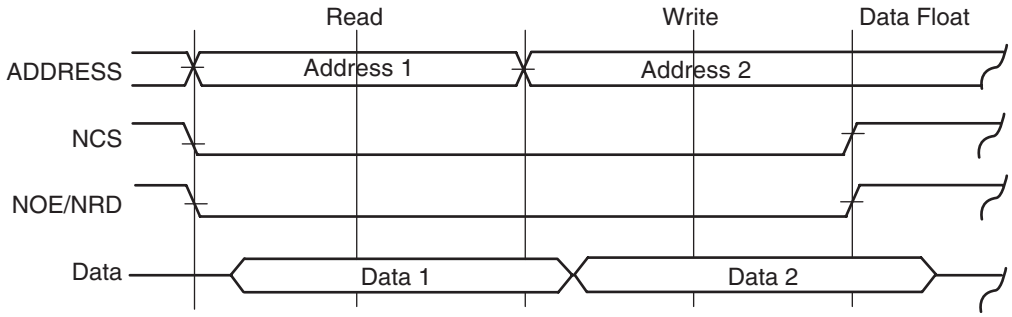


Figure 10-19. Sequential Early Read Access on the Same Chip Select with No Wait State

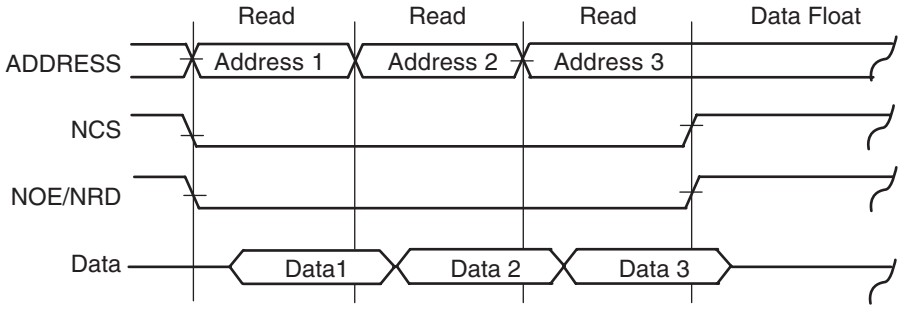


Figure 10-20. Sequential Read Access on Different Chip Select with NTDF = 2

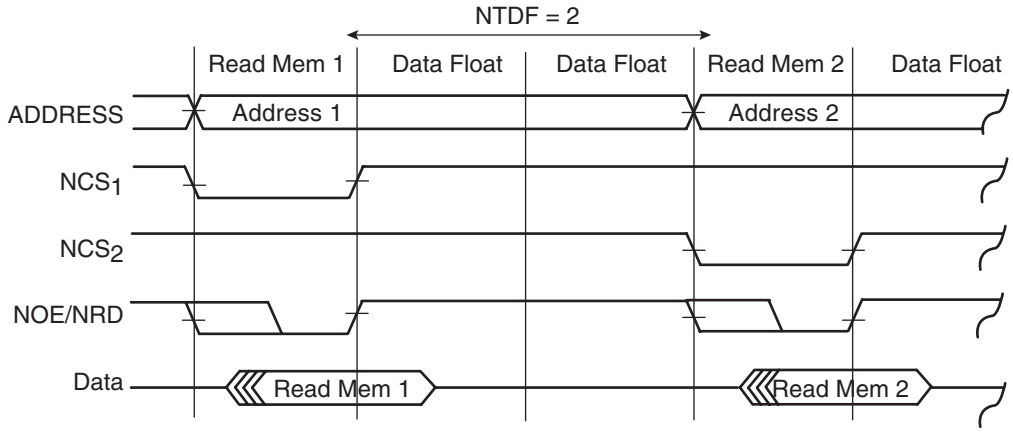


Figure 10-21. Sequential Read Access on Different Chip Select with NTDF = 0 or 1

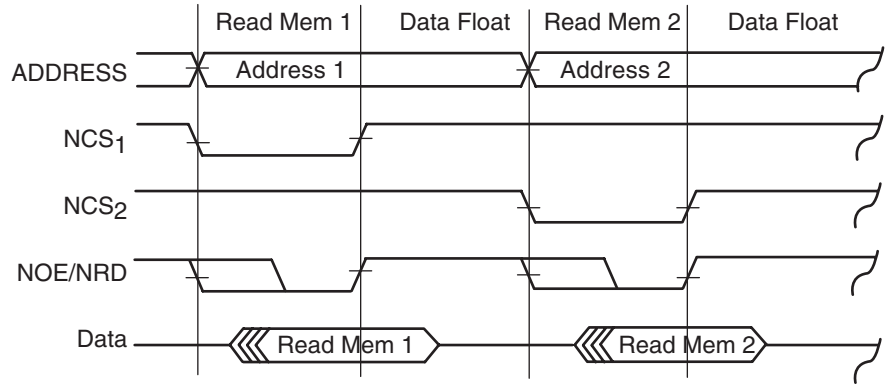


Figure 10-22. Sequential Standard Read Access on the Same Chip Select with One Wait State

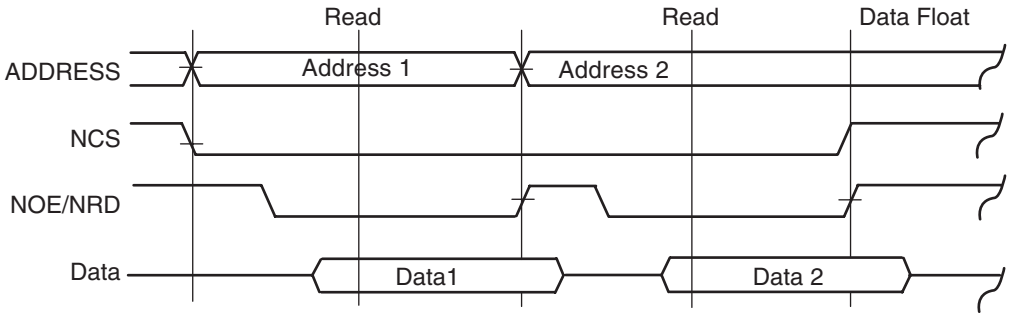


Figure 10-23. Sequential Standard Read Access on the Same Chip Select with No Wait State

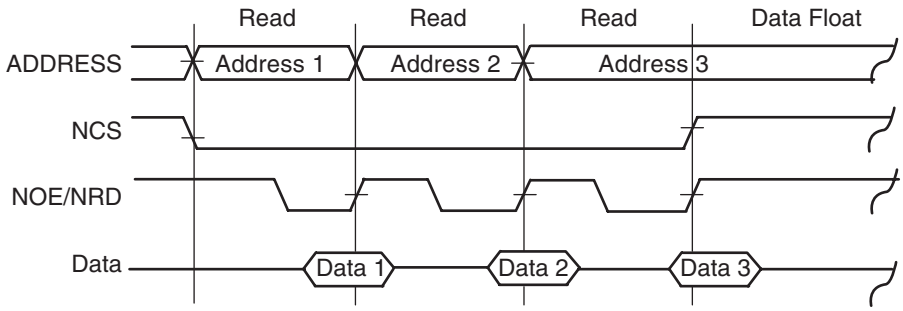


Figure 10-24. Sequential Write Access on the Same Chip Select with One Wait State

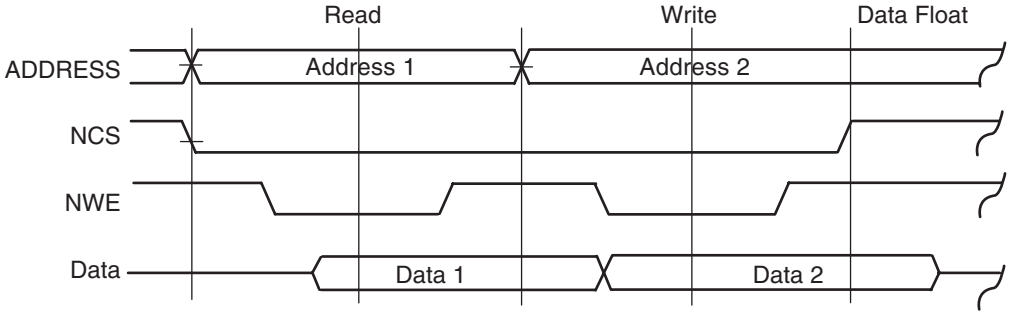


Figure 10-25. Sequential Write Access on the Same Chip Select with No Wait State

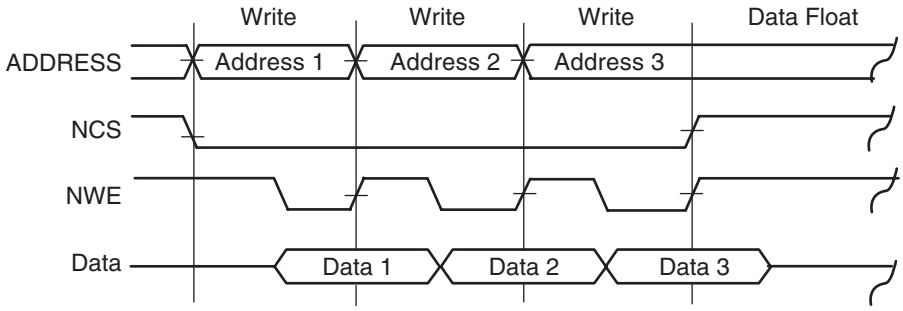


Figure 10-26. Sequential Write Access on Different Chip Select

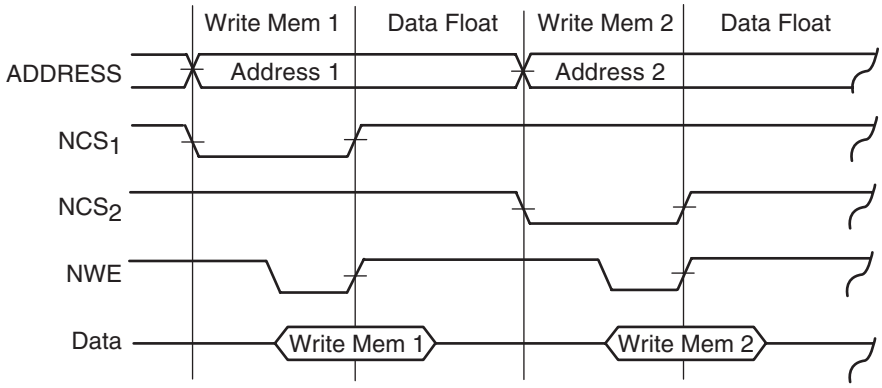


Figure 10-27. Write and Early Read on the Same Chip Select

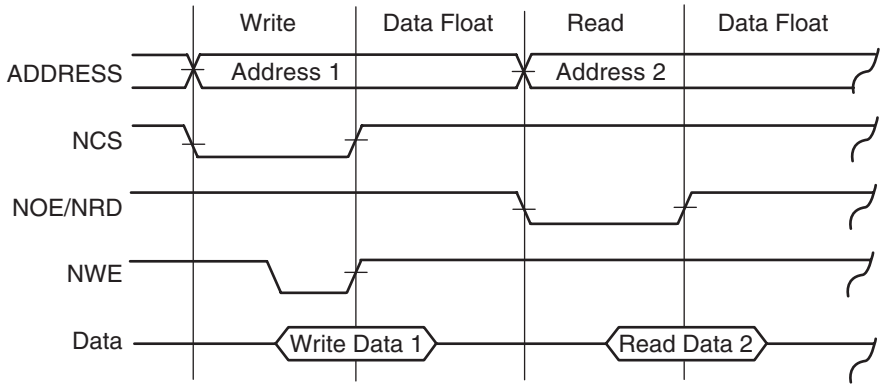


Figure 10-28. Write and Read on Different Chip Select

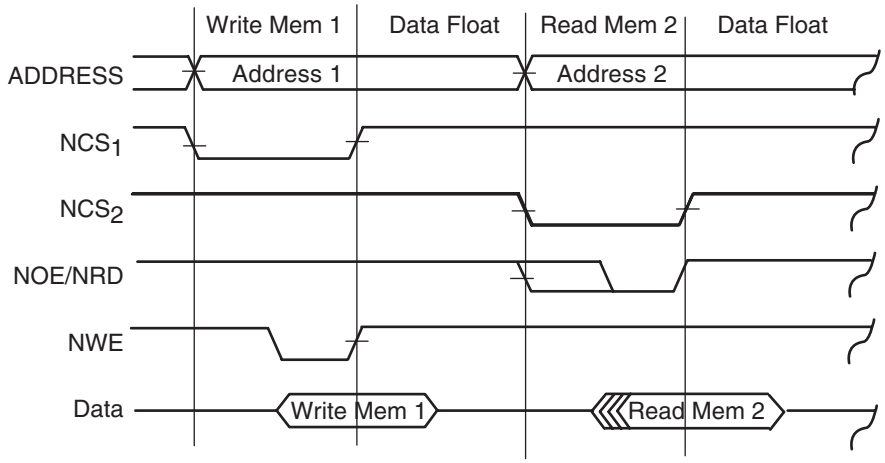
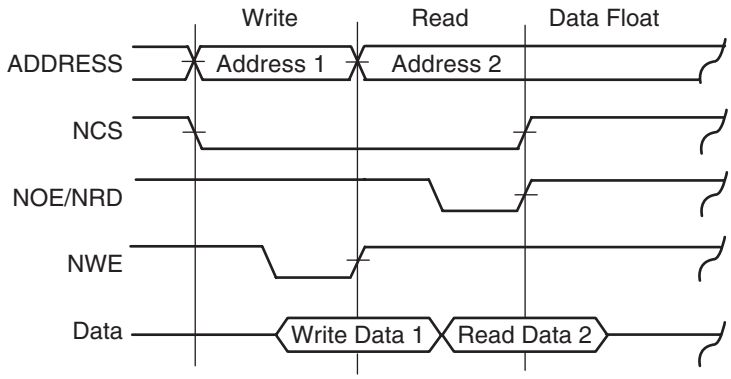


Figure 10-29. Write and Standard Read on the Same Chip Select



10.4.3.3 External Wait States

The NWAIT input can be used to add wait states at any time. The NWAIT signal is active low and is detected on the rising edge of the CORECLK signal. If the NWAIT signal is active on the rising edge of the CORECLK signal, the EBI adds a wait state and does not change either the output signal or its internal counters and state. When the NWAIT signal is released, the EBI finishes the access sequence after a minimum of two CORECLK periods and a maximum of three CORECLK periods (the positive edge of the NWAIT signal is internally resynchronized with the falling edge of the CORECLK signal and the ARM core finishes the access cycle two CORECLK periods after it has been sampled high with the falling edge of CORECLK).

In case of an external NWAIT, the number of internal wait states NWS must be calculated as in the equation below to verify the conditions in Table 10-5, “Timings for External NWAIT,” on page 47.

$$t_{WSCSAWSL} < (0.5 + NWS) \times (t_{CYCLE} - \text{load\_delay})$$

Case 1: No internal wait states (NWS = 0)

External NWAIT must be activated before the first rising edge of CORECLK (Figure 10-31) else it is not detected (Figure 10-32).

Figure 10-30. External Accesses with 0 Internal Wait States and no NWAIT

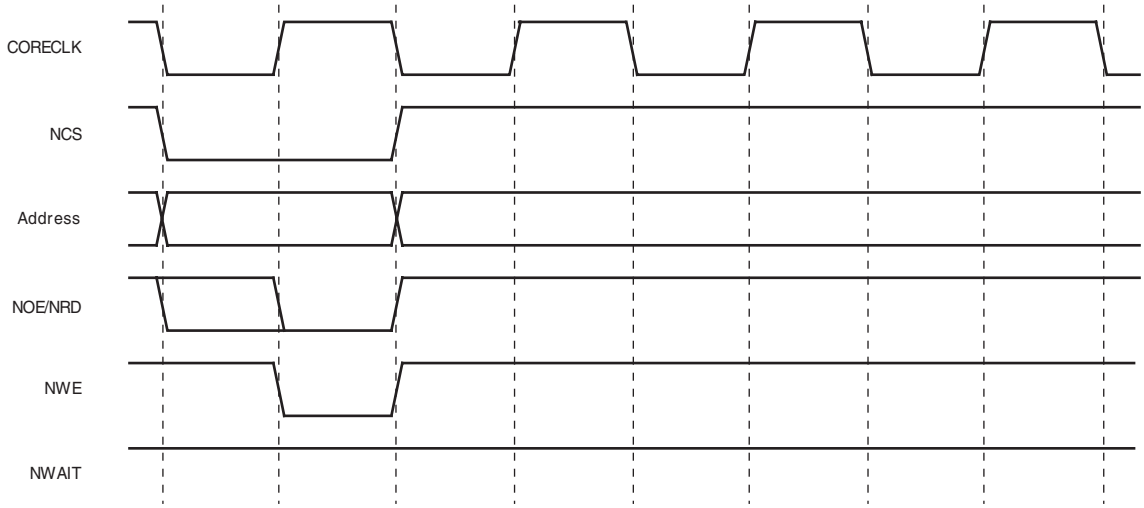


Figure 10-31. External Accesses with 0 Internal Wait States and NWAIT Detected

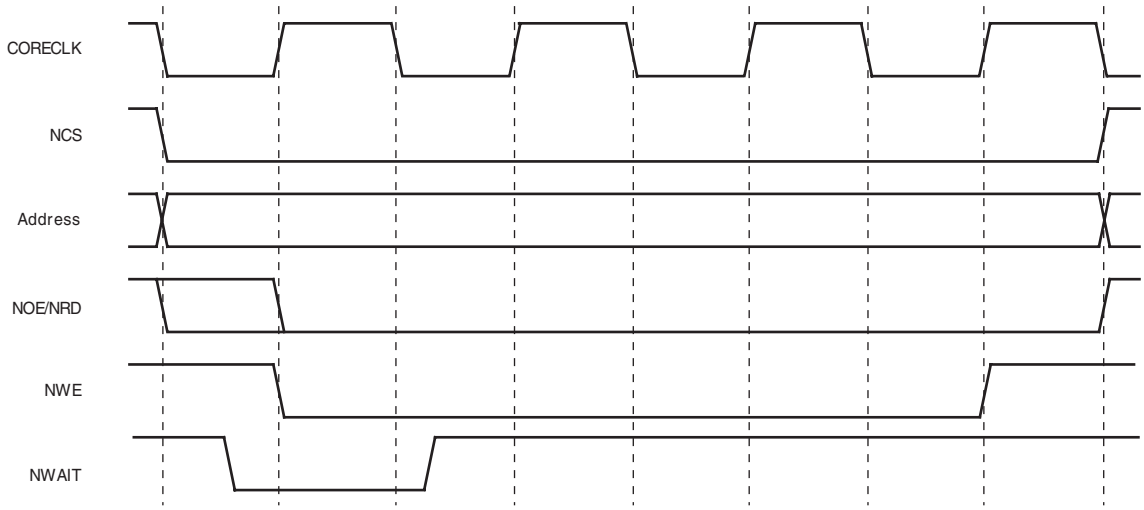
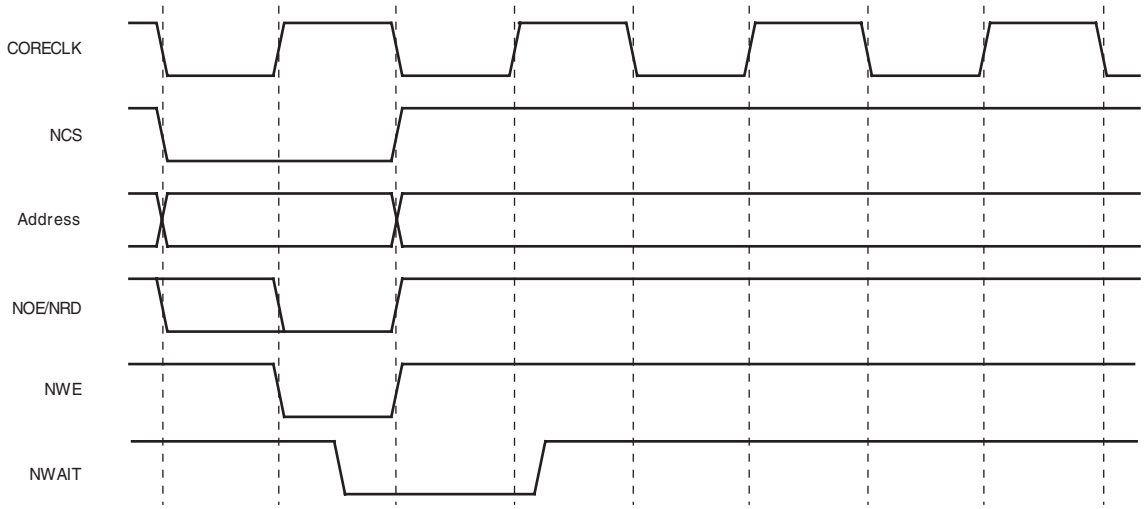


Figure 10-32. External Accesses with 0 Internal Wait States and NWAIT Not Detected



Case 2: With internal wait states (NWS ≥ 1)

As no CORECLK output signal is provided on the device, NWAIT timings are given relative to the address change and chip select activation. At high frequencies, it may be necessary to add internal wait states in order to allow an external device to decode the chip select and address lines and to drive the NWAIT input correctly ($t_{WSADCWSL}$, $t_{WSCSAWSL}$ and t_{WSPL} must be respected).

Figure 10-33. External Accesses with One Internal Wait State and One NWAIT

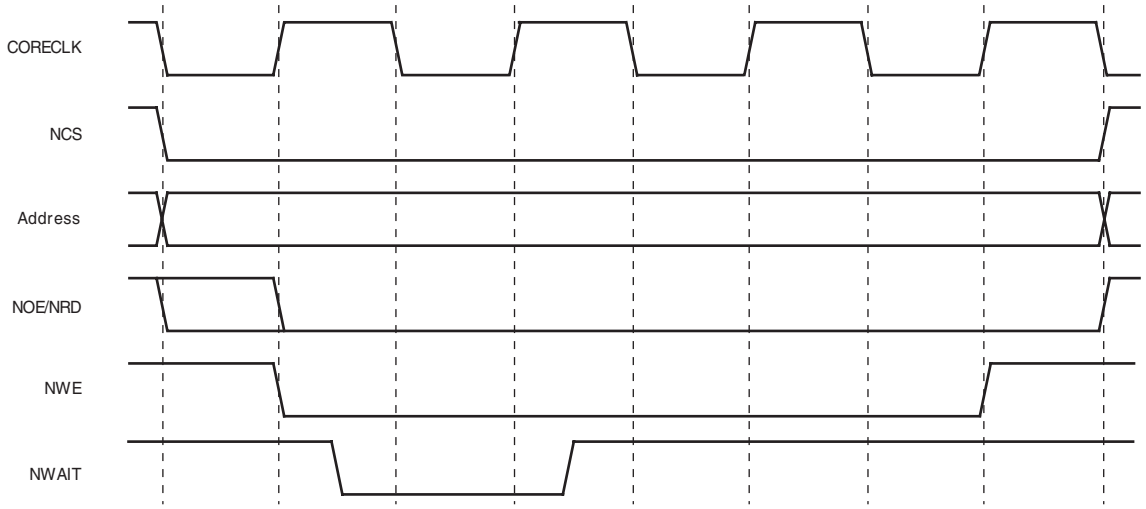
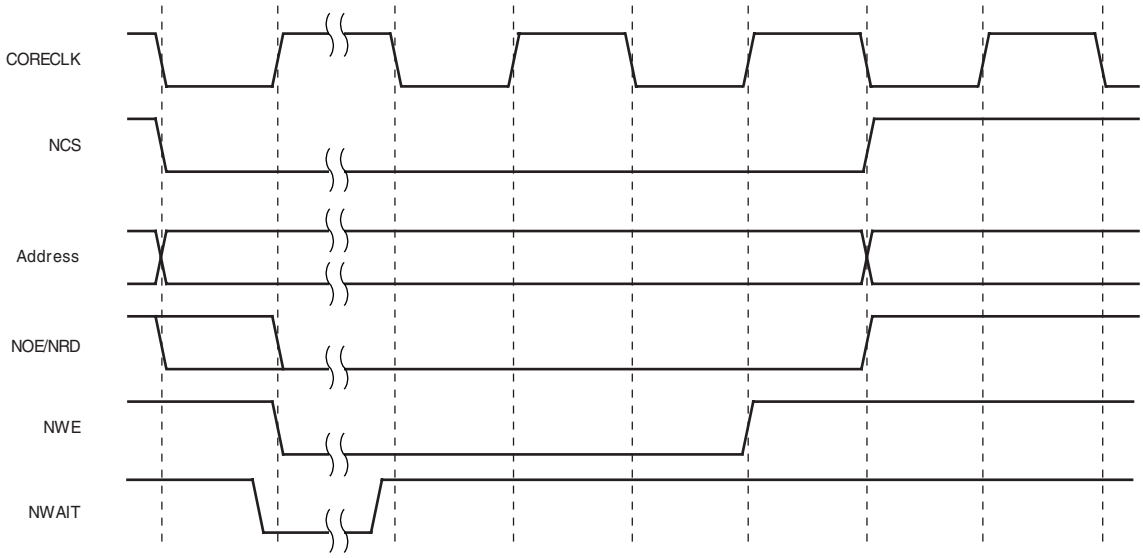


Figure 10-34. External Accesses with x Internal Wait States and y NWAIT



10.4.4 Timings

The following tables show the minimum and maximum timings for external memory read/write cycles (valid for the recommended operating conditions) for a capacitive load of 15 pF, 40 pF and 60 pF.

Table 10-3. Timings for Read Access

Read Access	Description	Load = 15pf		Load = 40pf		Load = 60pf	
		Min	Max	Min	Max	Min	Max
$tr_{ADCRDV2}$	Address change to read data valid (read/write memory, 0 wait state, standard read)		t_{CYCLE} - 19.2ns		t_{CYCLE} - 22.9ns		t_{CYCLE} - 25.8ns
$tr_{ADCRDV1}$	Address change to read data valid (all other cases)		(1 + NWS) * t_{CYCLE} - 11.0ns		(1 + NWS) * t_{CYCLE} - 12.9ns		(1 + NWS) * t_{CYCLE} - 14.4ns
tr_{CSLRDV}	Chip select low to read data valid		(1 + NWS) * t_{CYCLE} - 10.5ns		(1 + NWS) * t_{CYCLE} - 12.4ns		(1 + NWS) * t_{CYCLE} - 13.9ns
$tr_{OELRDV1}$	Output enable low to read data valid (standard read)		(0.5 + NWS) * t_{CYCLE} - 11.5ns		(0.5 + NWS) * t_{CYCLE} - 13.4ns		(0.5 + NWS) * t_{CYCLE} - 14.9ns
$tr_{OELRDV2}$	Output enable low to read data valid (early read)		(1 + NWS) * t_{CYCLE} - 11.6ns		(1 + NWS) * t_{CYCLE} - 13.5ns		(1 + NWS) * t_{CYCLE} - 15.0ns
$tr_{BSLRDV2}$	Byte select low to read data valid (read/write memory, 0 wait state, standard read)		(1 + NWS) * t_{CYCLE} - 18.5ns		(1 + NWS) * t_{CYCLE} - 22.2ns		(1 + NWS) * t_{CYCLE} - 25.1ns
$tr_{BSLRDV1}$	Byte select low to read data valid (all other cases)		(1 + NWS) * t_{CYCLE} - 12.0ns		(1 + NWS) * t_{CYCLE} - 13.8ns		(1 + NWS) * t_{CYCLE} - 15.3ns

Table 10-3. Timings for Read Access (Continued)

Read Access	Description	Load = 15pf		Load = 40pf		Load = 60pf	
		Min	Max	Min	Max	Min	Max
tr_{DH}	Data hold time from address change/NCS high/NOE high	0		0		0	
$tr_{DHZOEL1}$	Data Hi-Z to output enable low (previous is a write cycle, standard read)	-1.5ns		-3.4ns		-4.8ns	
$trr_{DHZOEL2}$	Data Hi-Z to output enable low (previous is a write cycle, early read)	0.5 * t_{CYCLE} - 1.5ns		0.5 * t_{CYCLE} - 3.4ns		0.5 * t_{CYCLE} - 4.8ns	
tr_{DHZCSL}	Data Hi-Z to chip select low (previous is a write cycle, standard read)	0.5 * t_{CYCLE} - 2.8ns		0.5 * t_{CYCLE} - 4.7ns		0.5 * t_{CYCLE} - 6.1ns	
tr_{ADHRH}	Address/CS/NUB/NLB hold time from read high	4.8ns		7.0ns		8.8ns	

Figure 10-35. Read Access Waveform

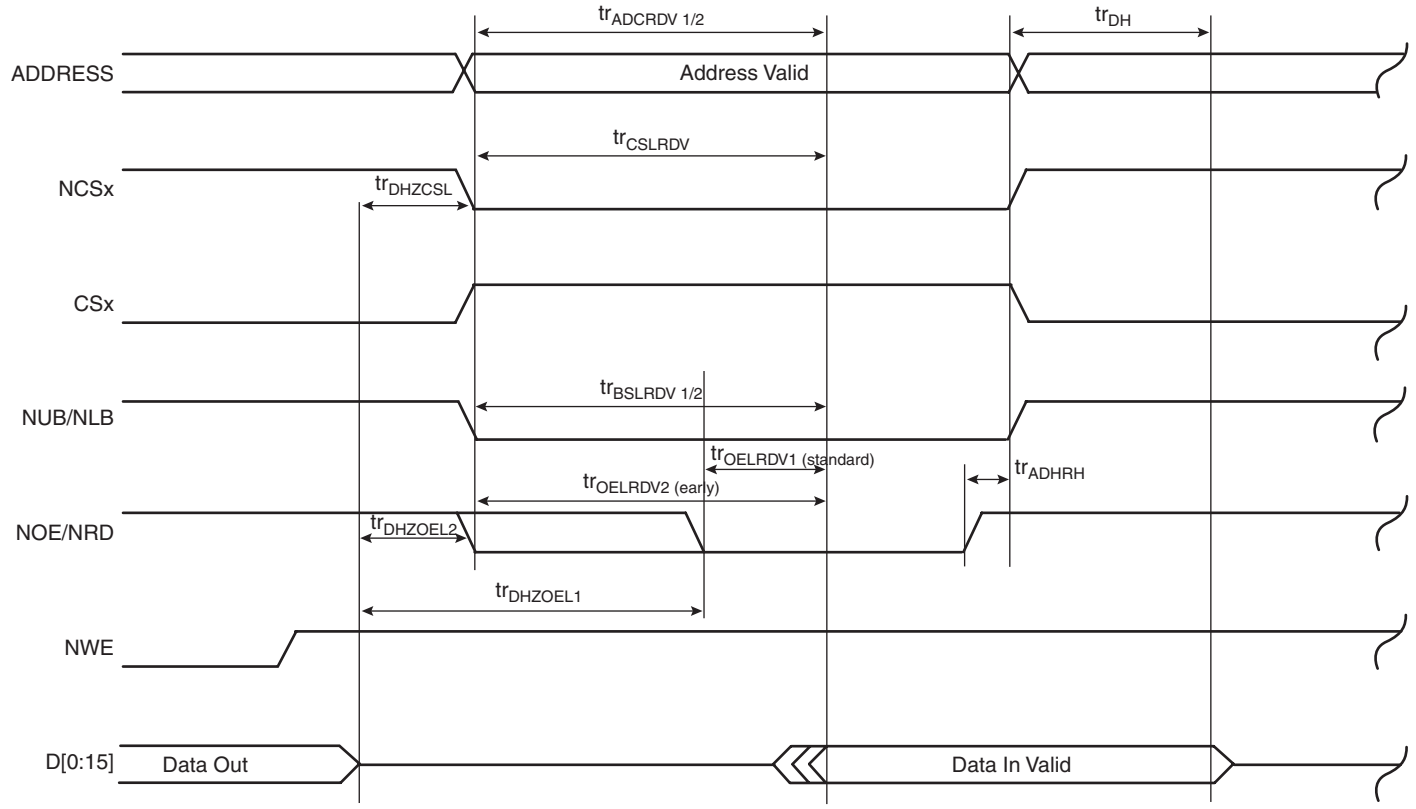


Table 10-4. Timings for Write Access

Write Access	Description	Load = 15pf		Load = 40pf		Load = 60pf	
		Min	Max	Min	Max	Min	Max
tw_{ADSWL}	Address/NCS/NUB/NLB setup time to write low	0.5 * t_{CYCLE} - 2.0ns		0.5 * t_{CYCLE} - 2.0ns		0.5 * t_{CYCLE} - 2.0ns	
tw_{WPL1}	Write pulse low (one or more wait states)	(1 + NWS) * t_{CYCLE} - 1.6ns		(1 + NWS) * t_{CYCLE} - 1.6ns		(1 + NWS) * t_{CYCLE} - 1.7ns	
tw_{WPL2}	Write pulse low (0 wait state)	0.5 * t_{CYCLE} - 2.1ns		0.5 * t_{CYCLE} - 2.1ns		0.5 * t_{CYCLE} - 2.2ns	
tw_{DSWH1}	Data setup time to write high (one or more wait states)	(1 + NWS) * t_{CYCLE} - 3.6ns		(1 + NWS) * t_{CYCLE} - 5.5ns		(1 + NWS) * t_{CYCLE} - 7.0ns	
tw_{DSWH2}	Data setup time to write high (0 Wait State)	0.5 * t_{CYCLE} - 4.1ns		0.5 * t_{CYCLE} - 6.0ns		0.5 * t_{CYCLE} - 7.5ns	
tw_{ADHWH}	Address/CS/NUB/NLB hold time from write high	2.6ns		3.3ns		3.9ns	
tw_{OEHDD}	Output enable high (previous is a read cycle) to data drive	0.5 * t_{CYCLE} - 5.7ns		0.5 * t_{CYCLE} - 7.5ns		0.5 * t_{CYCLE} - 8.9ns	
tw_{CSHDD}	Chip select high (previous is a read cycle) to data drive	1.5 * t_{CYCLE} - 4.3ns		1.5 * t_{CYCLE} - 6.2ns		1.5 * t_{CYCLE} - 7.6ns	
tw_{DHWH1}	Data hold time from write high (one or more wait states)	t_{CYCLE} - 3.8ns		t_{CYCLE} - 5.6ns		t_{CYCLE} - 7.0ns	
tw_{DHWH2}	Data hold time from write high (0 wait state)	0.5 * t_{CYCLE} - 4.1ns		0.5 * t_{CYCLE} - 5.9ns		0.5 * t_{CYCLE} - 7.4ns	

Figure 10-36. Write Access Waveform

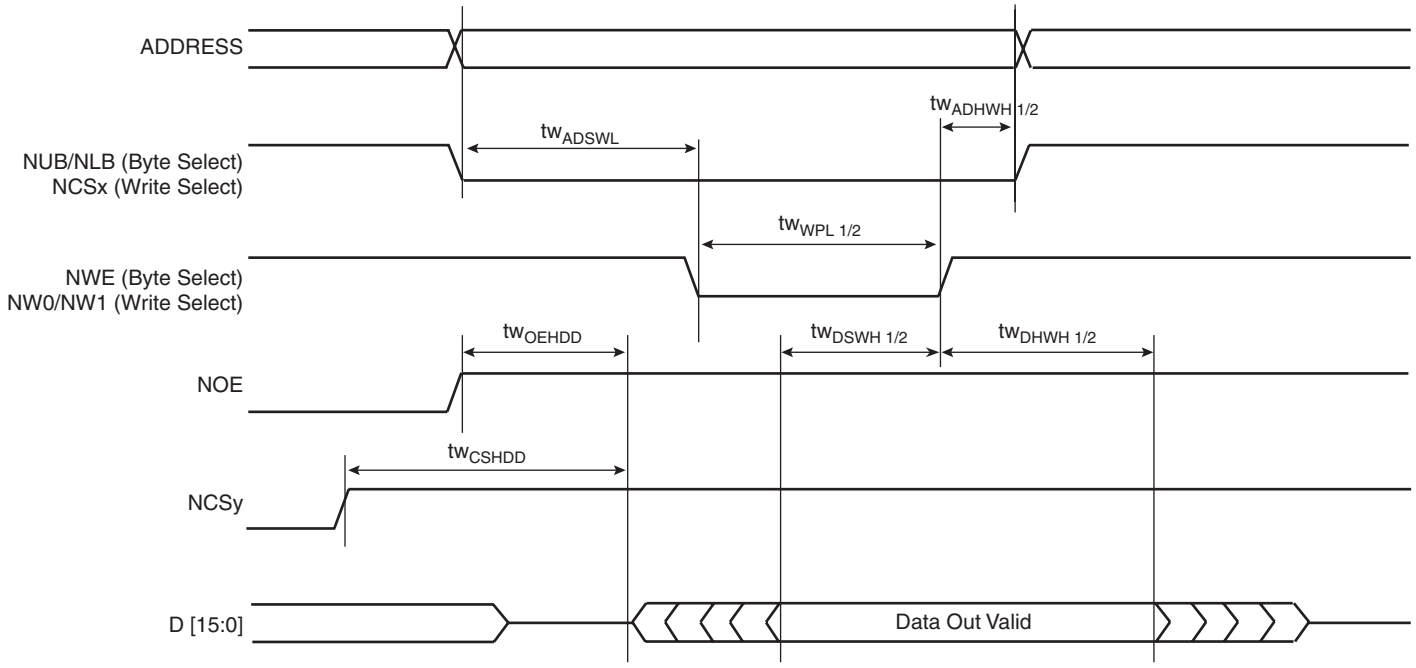
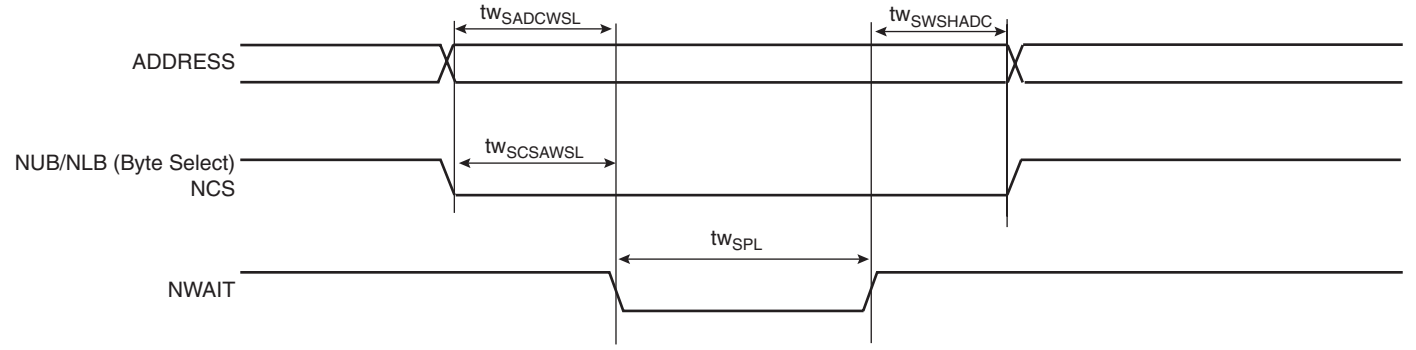


Table 10-5. Timings for External NWAIT

Write Access	Description	Load = 15pf		Load = 40pf		Load = 60pf	
		Min	Max	Min	Max	Min	Max
$tw_{SADCWSL}$	Address/NUB/NLB change to NWAIT active		(0.5 + NWS) * t_{CYCLE} - 16.8ns		(0.5 + NWS) * t_{CYCLE} - 23.5ns		(0.5 + NWS) * t_{CYCLE} - 26.9ns
$tw_{SCSAWSL}$	Chip select active to NWAIT active		(0.5 + NWS) * t_{CYCLE} - 15.8ns		(0.5 + NWS) * t_{CYCLE} - 22.5ns		(0.5 + NWS) * t_{CYCLE} - 25.8ns
tw_{SPL}	NWAIT pulse	t_{CYCLE}		t_{CYCLE}		t_{CYCLE}	
$tw_{SWSHADC}$	NWAIT inactive to address/NUB/NLB/chip select change	2 * t_{CYCLE}	3 * t_{CYCLE}	2 * t_{CYCLE}	3 * t_{CYCLE}	2 * t_{CYCLE}	3 * t_{CYCLE}

Figure 10-37. External NWAIT Waveform



10.4.4.1 EBI Timings Calculation

Table 10-6 and Table 10-7 show estimated timings relative to operating condition limits (worst case: VDD3V = 3.0V, Temp = +85°C) with a 40 pF load on address and control lines except for chip select lines with a 15 pF load.

Table 10-6. General-purpose EBI Signals

Symbol	Parameter	Min	Max	Units
EBI5	CORECLK falling to NCS, CS active	4.57	11.44	ns
EBI24	CORECLK falling to NCS, CS inactive	4.52	11.45	ns
EBI1	CORECLK falling to A[21:0] active	5.62	14.28	ns
EBI2	CORECLK falling to A[21:0] inactive	5.87	13.33	ns
EBI6	NWAIT setup before CORECLK rising	4.32		ns
EBI7	NWAIT hold after CORECLK rising	0		ns

Table 10-7. EBI Write Signals

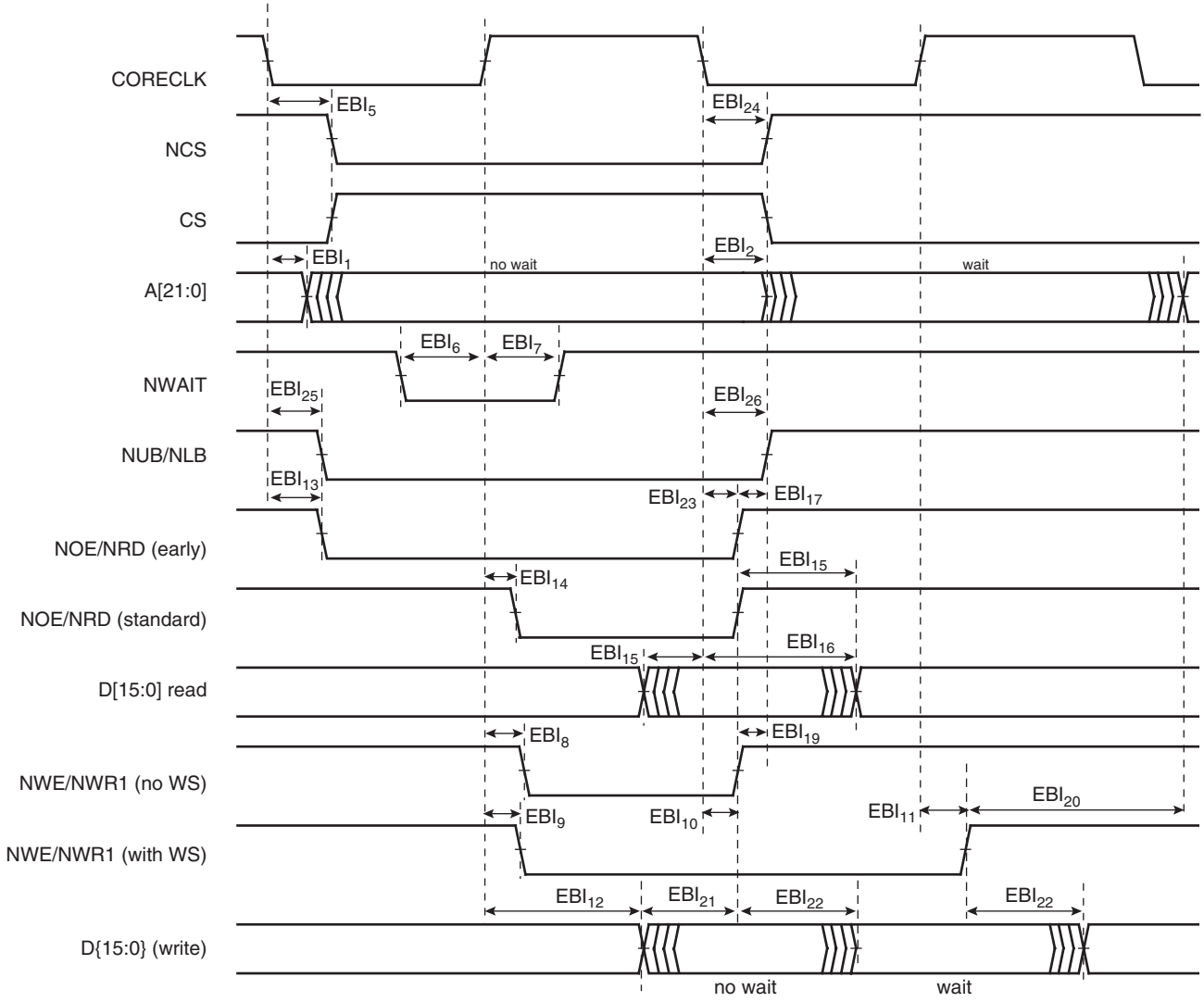
Symbol	Parameter	Min	Max	Units
EBI8	CORECLK rising to NWR active (no wait)	5.25	13.29	ns
EBI9	CORECLK rising to NWR active (wait)	5.25	13.29	ns
EBI12	CORECLK rising to D[15:0] out valid	5.90	17.76	ns
EBI10	CORECLK falling to NWR inactive (no wait)	5.15	13.08	ns
EBI11	CORECLK rising to NWR inactive (wait)	5.22	12.75	ns
EBI20	NWR high to A[21:1], NCS, CS changes (wait states)	3.38	9.53	ns
EBI21	Data out valid before NWR high	0.5 x t_{CP} - 6.00		ns
EBI22	Data out valid after NWR high	0.5 x t_{CP} - 5.90		ns
EBI19	NWR high to A[21:1], NCS, CS changes (no wait states)	3.30	9.53	ns

Table 10-8. EBI Read Signals

Symbol	Parameter	Min	Max	Units
EBI25	CORECLK falling to NUB/NLB active	5.05	14.28	ns
EBI26	CORECLK falling to NUB/NLB inactive	5.38	13.52	ns
EBI13	CORECLK falling to NRD active (early)	5.82	13.93	ns
EBI23	CORECLK falling to NRD inactive	5.67	13.92	ns
EBI14	CORECLK rising to NRD active (standard)	5.67	13.85	ns
EBI15	D[15:0] in setup before CORECLK falling	0.00	2.60	ns
EBI16	D[15:0] in hold after CORECLK falling	0.00		ns
EBI17	NRD high to A[21:1], NCS, CS changes	7.02	9.21	ns
EBI18	D[15:0] hold after NRD high	0.00		ns

Note: For t_{CP} t_{CH} and t_{CL} (see "Core Clock" on page 17).

Figure 10-38. External Bus Interface Signals Relative to CORECLK



If a different load is applied, the new timing can be calculated as follows:

For a high-to-low transition on the signal:

$$EBI_{Xnew} = EBI_X + \text{Derating Factor} \times \{[TPDHL_{new} + (DTPDHL_{new} \times C_{new})] - [TPDHL + (DTPDHL \times C)]\}$$

For a low-to-high transition on the signal:

$$EBI_{Xnew} = EBI_X + \text{Derating Factor} \times \{[TPDLH_{new} + (DTPDLH_{new} \times C_{new})] - [TPDLH + (DTPDLH \times C)]\}$$

where:

- Derating factor equals 1.65V in worst conditions (i.e., 3.0V @ +85°C) and 0.64 in best conditions (i.e., 3.6V @ -40°C)
- TPDHL is propagation delay, high-to-low
- TPDLH is propagation delay, low-to-high
- DTPDHL is differential (load-dependent) propagation delay, high-to-low or high impedance-to-low

- DTPDLH is differential (load-dependent) propagation delay, low-to-high or high impedance-to-high
- C_{NEW} is the new capacitive charge on the affected signal
- C is the current capacitive charge on the signal (40 pF load for all signals except for chip select lines - 15pF -)

10.5 Advanced Memory Controller (AMC) Memory Map

Table 10-9. AMC Memory Map[1]

Address	Register	Name	Access	Reset State
0xFFE0 0000	AMC Chip Select Register 0	AMC_CSR0	Read/Write	0x4000203D
0xFFE0 0004	AMC Chip Select Register 1	AMC_CSR1	Read/Write	0x48000000
0xFFE0 0008	AMC Chip Select Register 2	AMC_CSR2	Read/Write	0x50000000
0xFFE0 000C	AMC Chip Select Register 3	AMC_CSR3	Read/Write	0x58000000
0xFFE0 0010 – 0xFFE0 0014	Reserved	---	---	---
0xFFE0 0018	AMC Chip Select Register 6	AMC_CSR6	Read/Write	0x70000000
0xFFE0 001C	AMC Chip Select Register 7	AMC_CSR7	Read/Write	0x78000000
0xFFE0 0020	AMC Remap Control Register	AMC_RCR	Read/Write	0x00000000
0xFFE0 0024	AMC Memory Control Register	AMC_MCR	Read/Write	0x00000000

Note: 1. The software must set the AMC Registers for correct operation.

10.5.1 AMC Chip Select Register 0

Name: AMC_CSR0
Access: Read/Write
Address: 0xFFE0 0000

31	30	29	28	27	26	25	24
–	–	BA[9:4]					

23	22	21	20	19	18	17	16
BA[3:0]				–	–	–	–

15	14	13	12	11	10	9	8
–	–	CSEN	BAT	TDF[2:0]			PAGES1

7	6	5	4	3	2	1	0
PAGES0	–	WSE	NWS[2:0]			DBW[1:0]	

- **BA[9:0]: Base Address**

Bits [31:30] are set by hardware, so the base address can only be in the memory space 0x40000000-0x7FFFFFFF. The other bits contain the highest bits of the base address. If the page size is larger than 1 Mbyte, the unused bits of the base address are ignored by the AMC decoder.

- **CSEN: Chip Select Enable**

0: Chip select is disabled.

1: Chip select is enabled.

- **BAT: Byte Access Type**

0: Byte write access type.

1: Byte select access type.

- **TDF[2:0]: Data Float Output Time**

These bits select the number of cycles added after a memory transfer.

TDF[2:0]			Cycles Added
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

- **PAGES[1:0]: Page Size**

These bits select the memory page size.

PAGES[1:0]		Page Size	Active Bits in Base Address
0	0	1 Mbytes	12 (31-20)
0	1	4 Mbytes	10 (31-22)
1	0	16 Mbytes	8 (31-24)
1	1	64 Mbytes	6 (31-26)

- **WSE: Wait State Enable**

0: Wait state generation is disabled. No wait state is inserted.

1: Wait state generation is enabled.

- **NWS[2:0]: Number of Wait States**

These bits select the number of wait states added. This field is only valid if the **WSE** bit is set.

NWS[2:0]			WS Added
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

- **DBW[1:0]: Data Bus Width**

Type of data bus selected.

DBW[1:0]		Data Bus Width
0	0	Reserved
0	1	16-bit Data Bus
1	0	8-bit Data Bus
1	1	Reserved

10.5.2 AMC Chip Select Register x (x = 1..3)

Name: AMC_CSRx
Access: Read/Write
Address: 0xFFE0 0004, 0xFFE0 0008, 0xFFE0 000C

31	30	29	28	27	26	25	24
–	–	BA[9:4]					

23	22	21	20	19	18	17	16
BA[3:0]				–	–	–	–

15	14	13	12	11	10	9	8
–	–	CSEN	BAT	TDF[2:0]			PAGES1

7	6	5	4	3	2	1	0
PAGES0	–	WSE	NWS[2:0]			DBW[1:0]	

- **BA[9:0]: Base Address**

Bits [31:30] are set by hardware, so the base address can only be in the memory space 0x40000000-0x7FFFFFFF. The other bits contain the highest bits of the base address. If the page size is larger than 1Mbyte, the unused bits of the base address are ignored by the AMC decoder.

For AMC_CSR1 default value after reset is BA[9:0] = 0x080 (i.e., default base address = 0x48000000)

For AMC_CSR2 default value after reset is BA[9:0] = 0x100 (i.e., default base address = 0x50000000)

For AMC_CSR3 default value after reset is BA[9:0] = 0x180 (i.e., default base address = 0x58000000)

- **CSEN: Chip Select Enable**

0: Chip select is disabled.

1: Chip select is enabled.

- **BAT: Byte Access Type**

0: Byte write access type.

1: Byte select access type.

- **TDF[2:0]: Data Float Output Time**

These bits select the number of cycles added after a memory transfer.

TDF[2:0]			Cycles Added
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

- **PAGES[1:0]: Page Size**

These bits select the memory page size.

PAGES[1:0]		Page Size	Active Bits in Base Address
0	0	1 Mbytes	12 (31-20)
0	1	4 Mbytes	10 (31-22)
1	0	16 Mbytes	8 (31-24)
1	1	64 Mbytes	6 (31-26)

- **WSE: Wait State Enable**

0: Wait state generation is disabled. No wait state is inserted.

1: Wait state generation is enabled.

- **NWS[2:0]: Number of Wait States**

These bits select the number of wait states added. This field is only valid if the **WSE** bit is set.

NWS[2:0]			WS Added
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

- **DBW[1:0]: Data Bus Width**

Type of data bus selected.

DBW[1:0]		Data Bus Width
0	0	Reserved
0	1	16-bit Data Bus
1	0	8-bit Data Bus
1	1	Reserved

10.5.3 AMC Chip Select Register x (x = 6..7)

Name: AMC_CSRx
Access: Read/Write
Address: 0xFFE0 0018, 0xFFE0 001C

31	30	29	28	27	26	25	24
–	–	BA[9:4]					

23	22	21	20	19	18	17	16
BA[3:0]				–	–	–	–

15	14	13	12	11	10	9	8
–	–	CSEN	BAT	TDF[2:0]			PAGES1

7	6	5	4	3	2	1	0
PAGES0	–	WSE	NWS[2:0]			DBW[1:0]	

- **BA[9:0]: Base Address**

Bits [31:30] are set by hardware, so the base address can only be in the memory space 0x40000000-0x7FFFFFFF. The other bits contain the highest bits of the base address. If the page size is larger than 1Mbyte, the unused bits of the base address are ignored by the AMC decoder.

For AMC_CSR6 default value after reset is BA[9:0] = 0x300 (i.e., default base address = 0x70000000)

For AMC_CSR7 default value after reset is BA[9:0] = 0x380 (i.e., default base address = 0x78000000)

- **CSEN: Chip Select Enable**

0: Chip select is disabled.

1: Chip select is enabled.

Reset value for CSEN is 0, A21/CS6 and A20/CS7 are configured as address lines after reset.

- **BAT: Byte Access Type**

0: Byte write access type.

1: Byte select access type.

- **TDF[2:0]: Data Float Output Time**

These bits select the number of cycles added after a memory transfer.

TDF[2:0]			Cycles Added
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

- **PAGES[1:0]: Page Size**

These bits select the memory page size.

PAGES[1:0]		Page Size	Active Bits in Base Address
0	0	1 Mbytes	12 (31-20)
0	1	4 Mbytes	10 (31-22)
1	0	16 Mbytes	8 (31-24)
1	1	64 Mbytes	6 (31-26)

- **WSE: Wait State Enable**

0: Wait state generation is disabled. No wait state is inserted.

1: Wait state generation is enabled.

- **NWS[2:0]: Number of Wait States**

These bits select the number of wait states added. This field is only valid if the **WSE** bit is set.

NWS[2:0]			WS Added
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

- **DBW[1:0]: Data Bus Width**

Type of data bus selected.

DBW[1:0]		Data Bus Width
0	0	Reserved
0	1	16-bit Data Bus
1	0	8-bit Data Bus
1	1	Reserved

10.5.4 AMC Remap Control Register

Name: AMC_RCR
Access: Read/Write
Address: 0xFFE0 0020

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**

0: No effect

1: Performs the remapping of the two memory devices (internal RAM and external memory on NCS0). Memory map switches from 'reboot' mode to 'remap mode'.

This bit is read at logical 0 in reboot mode and at logical 1 in remap mode.

10.5.5 AMC Memory Control Register

Name: AMC_MCR
Access: Read/Write
Address: 0xFFE0 0024

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[2:0]		

- **DRP: Data Read Protocol**

0: Standard read protocol for all external memory devices enabled.

1: Early read protocol for all external memory devices enabled.

- **ALE[2:0]: Address Line Enable**

These bits indicate the number of valid chip select lines.

ALE2	ALE1	ALE0	Valid Address Bits	Maximum Addressable Space per Chip Select Line	Valid Chip Select
0	x	x	ADD[21:0]	4 Mbytes	NCS[3:0]
1	0	x	ADD[21:0]	4 Mbytes	NCS[3:0]
1	1	0	ADD[20:0]	2 Mbytes	NCS[3:0], CS6
1	1	1	ADD[19:0]	1 Mbytes	NCS[3:0], CS6, CS7

11. Clock Manager (CM)

The AT91SAM7A1 microcontroller provides:

- 32.768 kHz oscillator (real-time clock oscillator)
- 4 MHz to 16 MHz oscillator
- Programmable PLL (x2 to x20)
- Programmable master clock divider

Clock management is done through the Clock Manager (CM). This allows the user to select between the different working modes LPM, SLM and OPE.

At power-up, the master clock oscillator and the real-time clock oscillator are enabled. As the application may or may not use the real-time clock oscillator, the DIVCLK clock is used as the low-frequency clock (LFCLK). This ensures that both the CORECLK and the LFCLK clock can be used at power-up.

At power-up, the PLL is disabled (PLLSCLT = 0 in the CM_CS register) to allow the user to select the application CORECLK frequency.

Figure 11-1. Clock Management

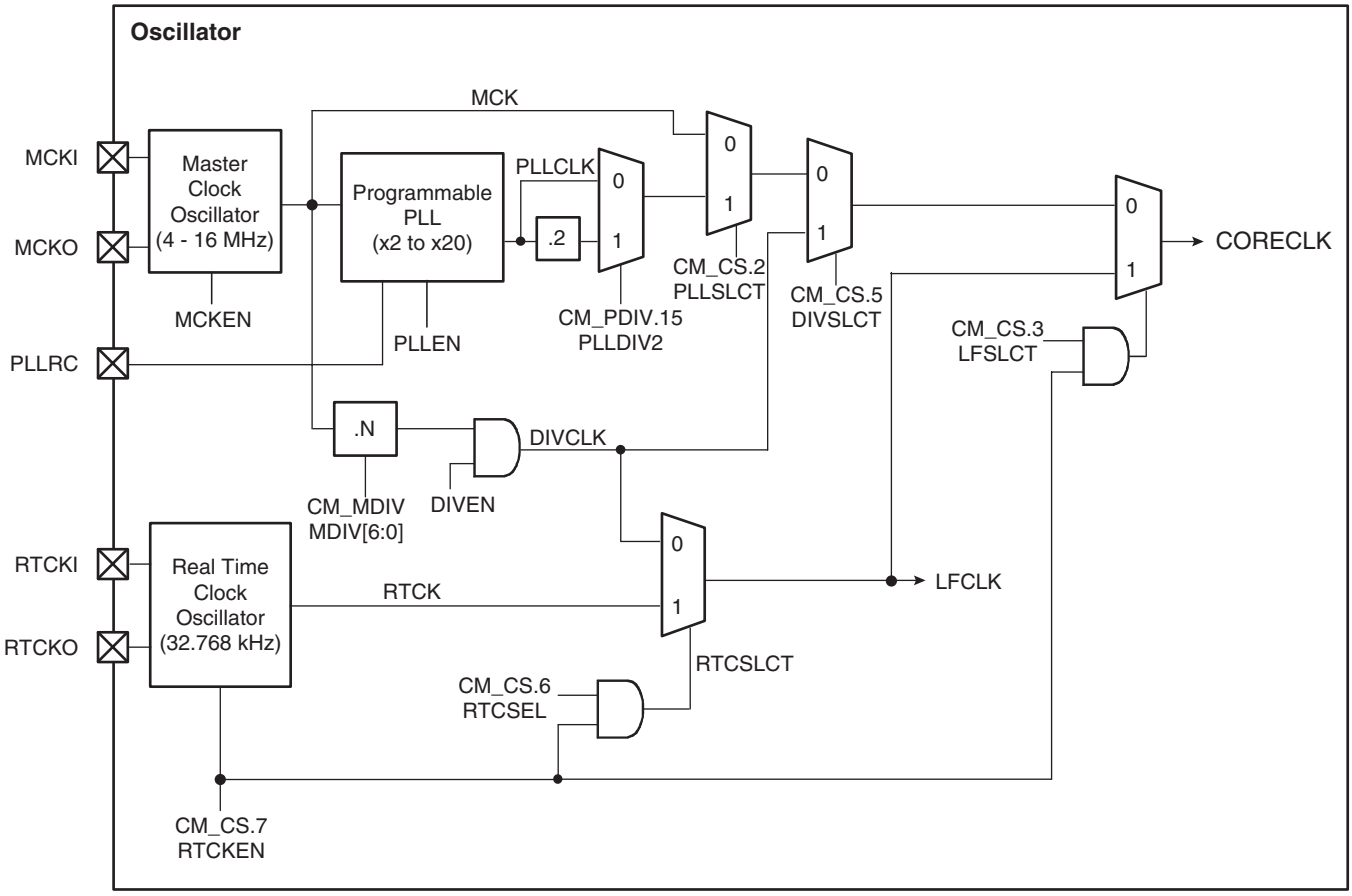


Table 11-1 lists the different working modes depending on the value set in the CM_CS register. The grayed row shows values at reset.

Table 11-1. Clock Selection

CM_CS7 RTCKEN	CM_CS6 RTCSEL	CM_CS3 LFSLCT	CM_CS5 DIVSLCT	CM_CS2 PLLSLCT	CM_CS1 PLLDIV2	CM_CS8 MCKEN	CM_CS9 PLLEN	CM_CS13 DIVEN	CM_CS14 RTCSLCT	CORECLK	LFCLK	Mode
0	X	X	1	X	X	1	0	1	0	MCK/(2x(MDIV+1))	MCK/(2x(MDIV+1))	SLM
0	X	X	0	0	X	1	0	1	0	MCK	MCK/(2x(MDIV+1))	SLM
0	X	0	0	1	0	1	1	1	0	MCKxPMUL	MCK/(2x(MDIV+1))	OPE
0	X	0	0	1	1	1	1	1	0	(MCKxPMUL)/2	MCK/(2x(MDIV+1))	OPE
1	1	1	X	X	X	0	0	0	1	RTCK	RTCK	LPM
1	1	0	1	X	X	1	0	1	1	MCK/(2x(MDIV+1))	RTCK	SLM
1	1	0	0	0	X	1	0	0	1	MCK	RTCK	SLM
1	1	0	0	1	0	1	1	0	1	MCKxPMUL	RTCK	OPE
1	1	0	0	1	1	1	1	0	1	(MCKxPMUL)/2	RTCK	OPE
1	0	1	X	X	X	1	0	1	0	MCK/(2x(MDIV+1))	MCK/(2x(MDIV+1))	SLM
1	0	0	1	X	X	1	0	1	0	MCK/(2x(MDIV+1))	MCK/(2x(MDIV+1))	SLM
1	0	0	0	0	X	1	0	1	0	MCK	MCK/(2x(MDIV+1))	SLM
1	0	0	0	1	0	1	1	1	0	MCKxPMUL	MCK/(2x(MDIV+1))	OPE
1	0	0	0	1	1	1	1	1	0	(MCKxPMUL)/2	MCK/(2x(MDIV+1))	OPE

11.1 Clock Manager (CM) Memory Map

Table 11-2. CM Memory Map

Address	Register	Name	Access	Reset State
0xFFFEC000	CM Clock Enable	CM_CE	Write-only	---
0xFFFEC004	CM Clock Disable	CM_CD	Write-only	---
0xFFFEC008	CM Clock Status	CM_CS	Read-only	0x00002180
0xFFFEC00C	CM PLL Stabilization Time	CM_PST	Read/Write	0x000000B0
0xFFFEC010	CM PLL Divider	CM_PDIV	Read/Write	0x0000800A
0xFFFEC014	CM Oscillator Stabilization Time	CM_OST	Read/Write	0x000000B0
0xFFFEC018	CM Master Clock Divider	CM_MDIV	Read/Write	0x0000001F

11.1.1 CM Clock Enable Register

Name: CM_CE
Access: Write-only
Address: 0xFFFEC000

31	30	29	28	27	26	25	24
CLKEKEY[15:8]							
23	22	21	20	19	18	17	16
CLKEKEY[7:0]							
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
RTCKEN	RTCSEL	DIVSLCT	–	LFSLCT	PLLSLCT	–	–

- **CLKEKEY[15:0]: Key for Write Access into the CM_CE Register**

Any write in the CM_CD register bits will be effective only if CLKEKEY[15:0] is equal to 0x2305.

11.1.2 CM Clock Disable Register

Name: CM_CD
Access: Write-only
Address: 0xFFFEC004

31	30	29	28	27	26	25	24
CLKDKEY[15:8]							
23	22	21	20	19	18	17	16
CLKDKEY[7:0]							
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
RTCKEN	RTCSEL	DIVSLCT	–	LFSLCT	PLLSLCT	–	–

- **CLKDKEY[15:0]: Key for Write Access into the CM_CD Register**

Any write in the CM_CD register bits will be effective only if CLKDKEY[15:0] is equal to 0x1807.

11.1.3 CM Clock Status Register

Name: CM_CS
Access: Read-only
Address: 0xFFFEC008

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

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

15	14	13	12	11	10	9	8
–	RTCSLCT	DIVEN	–	–	–	PLLEN	MCKEN

7	6	5	4	3	2	1	0
RTCKEN	RTCSEL	DIVSLCT	–	LFSLCT	PLLSLCT	–	–

- **PLLSLCT: PLL/Master Clock Selection**

0: Selects MCK clock (deselects PLLCLK or PLLCLK/2 clock).

1: Selects PLLCLK or PLLCLK/2 clock (deselects MCK clock).

- **LFSLCT: Low Frequency Clock Selection**

0: Allows selection of MCK, PLLCLK, PLLCK/2 or DIVCLK.

1: Selects low frequency clock LFCLK (also disables master clock oscillator and PLL).

- **DIVSLCT: Programmable Clock Selection**

0: Allows selection of MCK, PLLCK or PLLCLK/2 (also deselects the DIVCLK clock).

1: Selects DIVCLK, i.e. MCK divided by MDIV[6:0] (also deselects the master clock or PLL clock).

- **RTCSEL: RTC frequency clock selection**

0: Selects the DIVCLK clock for low power clock (deselects the RTCK clock).

1: Selects the RTCK clock for low power clock (deselects the DIVCLK clock).

- **RTCKEN: Low Frequency Clock Oscillator**

0: The low frequency clock oscillator is disabled.

1: The low frequency clock oscillator is enabled.

- **MCKEN: Master Clock Oscillator Enable**

0: MCKEN signal is at a logical 0. The master clock oscillator is disabled and bypassed.

1: MCKEN signal is at a logical 1. The master clock oscillator is activated.

- **PLLEN: PLL Enable**

0: PLLEN signal is at a logical 0. PLL is deactivated.

1: PLLEN signal is at a logical 1. PLL is enabled.

- **DIVEN: Programmable Divider Enable**

0: DIVEN signal is at a logical 0. The programmable divider is disabled.

1: DIVEN signal is at a logical 1. The programmable divider is enabled.

- **RTCSLCT: Low Frequency Clock Selection**

0: RTCSLCT signal is at a logical 0. The DIVCLK is selected for LFCLK.

1: RTCSLCT signal is at a logical 1. The RTCK is selected for LFCLK.

11.1.4 CM PLL Stabilization Timer Register

Name: CM_PST
Access: Read/Write
Address: 0xFFFEC00C

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

- **PSTB[9:0]: PLL Stabilization Time**

Number of clock cycles needed before PLL stabilization. This register must be configured by the software after a hardware reset and before using the PLL.

The required time for PLL stabilization is T_{SETUP} minimum and is based on the MCK/256 clock (MCKEN = 1).

The default value is 0x000000B0 guaranteeing 176 x 256 MCK clock cycles (i.e., 11.264 ms with MCK = 4.0 MHz).

When the clock manager is configured in LPM mode and the program enables both the PLL and the master oscillator, PSTB should include the oscillator stabilization time and the PLL stabilization time. This is due to the fact that PSTB counter and OSTB counter decrement in parallel.

The PLL transient behavior before mathematical locking (phase error between the reference signal and derived signal less than ± 2π) is complex and difficult to describe using simple mathematical expression. Thus, there is no general formula giving the set-up time for any step-response transient behavior that unlocks the loop. Nevertheless, this set-up time can be approximated by a simple loop filter capacitor charging time T_{SETUP} in the worst case:

$$T_{SETUP} \leq \alpha \cdot \left[\frac{C3 + C4}{I_P}\right] \cdot \frac{VDDPLLV3}{2}$$

where:

– C3 and C4 are the loop filter capacitors,
– I_p the charge pump current (see ”PLL Characteristics” on page 15),
– α is a margin factor, set to 3 or 4 as a minimum

This formula overestimates the required time, but gives an easy way to approximate this setup time.

- **PSTKEY[15:0]: Key for Write Access into the CM_PST Register**

Any write in PSTB[9:0] are effective only if PSTKEY[15:0] is equal to 0x59C1. These bits are always read at 0.

Note: Write accesses to this register are only valid if PLLEN is at logical 0 (i.e., PLL not enabled).

11.1.5 CM PLL Divider Register

Name: CM_PDIV
Access: Read/Write
Address: 0xFFFEC010

31	30	29	28	27	26	25	24
PDIVKEY[15:8]							

23	22	21	20	19	18	17	16
PDIVKEY[7:0]							

15	14	13	12	11	10	9	8
PLLDIV2	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	–	–	PMUL[4:0]				

- **PMUL[4:0]: PLL Multiplier**

These bits select the PLL multiplier.

PMUL[4:0]	PLL Multiplier
0	Remains in previous state
1	Remains in previous state(1)
2	2
3	3
…	…
19	19
20	20
21 to 31	Remains in previous state

Note: Multiplying the MCK clock by 1 is equivalent to bypassing the PLL (i.e., CM_CS.2 = 0, CM_CS.3 = 0, CM_CS.5 = 0)

- **PLLDIV2: PLL Divider**

0: Selects PLLCLK clock (deselects PLLCLK/2)

1: Selects PLLCLK/2 clock (deselects PLLCLK)

- **PDIVKEY[15:0]: Key for Write Access into the CM_PDIV Register**

Any write in the PMUL[4:0] bits will only be effective if the PDIVKEY[15:0] is equal to 0x762D. These bits are always read at 0.

The output frequency of the PLL is equal to: MCK x PMUL[4:0], where MCK is the PLL input clock.

Note: Write accesses to this register are only valid if PLLEN is at logical 0 (i.e., PLL disabled).

11.1.6 CM Oscillator Stabilization Timer Register

Name: CM_OST
Access: Read/Write
Address: 0xFFFEC014

31	30	29	28	27	26	25	24
OSTKEY[15:8]							

23	22	21	20	19	18	17	16
OSTKEY[7:0]							

15	14	13	12	11	10	9	8
–	–	–	–	–	–	OSTB[9:8]	

7	6	5	4	3	2	1	0
OSTB[7:0]							

- **OSTB[9:0]: Oscillator Stabilization Time**

Number of clock cycles needed before master oscillator stabilization. This register must be configured by the software after a hardware reset and before using the master oscillator.

The required time for master oscillator stabilization is 4 ms maximum and is based on the MCK/256 clock.

The default value is 0x000000B0 guaranteeing 176 x 256 MCK clock cycles (i.e., 11.264 ms with MCK = 4.0 MHz)

- **OSTKEY[15:0]: Key for Write Access into the CM_OST Register**

Any write in the OSTB[9:0] bits will only be effective if the OSTKEY[15:0] bits are equal to 0xDB5A. These bits are always read at 0.

11.1.7 CM Master Clock Divider Register

Name: CM_MDIV
Access: Read/Write
Address: 0xFFFEC018

31	30	29	28	27	26	25	24
MDIVKEY[15:8]							

23	22	21	20	19	18	17	16
MDIVKEY[7:0]							

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	MDIV[6:0]						

- **MDIV[6:0]: Master Clock Divider**

MDIV[6:0] is used to divide the MCK clock and generate the DIVCLK. Default value for MDIV[6:0] is 0x1F.

$$DIVCLK = \frac{MCK}{2 \times (MDIV[6:0] + 1)}$$

- **MDIVKEY[15:0]: Key for Write Access into the CM_MDIV Register**

Any write in the MDIV[6:0] bits is effective only if the MDIVKEY[15:0] bits are equal to 0xACDC. These bits are always read at 0.

12. Special Function Mode (SFM)

The AT91SAM7A1 provides registers which implement the following special functions:

- Chip identification
- Reset status
- Test mode

12.1 Chip Identification

Chip identification is done via three registers: the Chip ID (SFM_CIDR), the Extended Chip ID (SFM_EXID) and the manufacturer ID.

These registers give information on:

- Internal memories used (type and size)
- Europe Technologies® project code

12.2 Reset Status

The AT91SAM7A1 includes the Reset Status (SFM_RSR) register to give the last cause of reset (i.e., hardware reset or internal watchdog reset).

12.3 Test Mode

The AT91SAM7A1 provides functional test mode in order to check or test some registers in internal peripherals.

The test mode is entered using the SFM_TM register.

It must be noted that this test mode is different from the factory test mode entered through the external pins TEST and SCANEN.

12.4 Special Function Mode (SFM) Memory Map

Table 12-1. SFM Memory Map

Address	Register	Name	Access	Reset State
0xFFF00000	SFM Chip ID	SFM_CIDR	Read-only	0x80000300
0xFFF00004	SFM Extended Chip ID	SFM_EXID	Read-only	0x02001102
0xFFF00008	SFM Reset Status	SFM_RSR	Read-only	0x0000006C or 0x00000053
0xFFF0000C	Reserved	---	---	---
0xFFF00014	SFM Test Mode	SFM_TM	Read/Write	0x00000000

12.4.1 SFM Chip ID Register

Name: SFM_CIDR
Access: Read-only
Address: 0xFFF00000

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

23	22	21	20	19	18	17	16
				ARCH[3:0]			

15	14	13	12	11	10	9	8
NVPMT[3:0]				IRS[3:0]			

7	6	5	4	3	2	1	0
NVDMS[3:0]				NVPMS[3:0]			

- **NVPMS[3:0]: Non Volatile Program Memory Size**

0000_b: None.

Other: Memory size = $2^{(14+NVPMS[3:0])}$ bytes.

- **NVDMS[3:0]: Non Volatile Data Memory Size**

0000_b: None.

Other: Reserved.

- **IRS[3:0]: Internal RAM Size**

Internal RAM size = $2^{(9+IRS[3:0])}$ bytes.

- **NVPMT[3:0]: Non Volatile Program Memory Type**

0000_b: ROMless.

0001_b: Mask ROM.

Other: Reserved.

- **ARCH[3:0]: Core Architecture**

0000_b: ARM7TDMI.

Other: Reserved.

- **EXT: Extension Flag**

0: No extended chip ID.

1: Extended chip ID existing.

12.4.2 SFM Extended Chip ID Register

Name: SFM_EXID
Access: Read-only
Address: 0xFFF00004

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

12.4.3 SFM Reset Status Register

Name: SFM_RSR
Access: Read-only
Address: 0xFFF00008

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[7:0]							

This register gives the last cause of reset.

- **RESET[7:0]: Cause of Reset**

0x6C: External reset on NRESET pin.

0x53: Internal watchdog reset.

12.4.4 SFM Test Mode Register

Name: SFM_TM
Access: Read/Write
Address: 0xFFF00014

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

This register must only be used for debug purposes.

- **TESTEN: Test Enable**

0: Disable test mode (the internal TEST_ENABLE signal is driven low at peripherals input).

1: Enable test mode (the internal TEST_ENABLE signal is driven high at peripherals input).

- **KEY[15:0]: Test Key for Test Mode Entry**

TESTEN bit can only be set/reset if the correct KEY word is entered: KEY[15:0] = 0xD64A.

13. Watchdog Timer (WD)

The watchdog timer (WD) is used to prevent the system from locking-up, for example in infinite software loops.

If the software does not clear the watchdog between watchdog pending interrupt and a programmed timeout delay, then it can generate an interrupt or internal reset.

The watchdog timer has a 16-bit down counter. Bits 15 to 10 of the value loaded when the watchdog is restarted are programmable using the HPVC[4:0] parameter in WD_MR register.

The software can control the action to perform when the WD counter overflows (i.e., reaches 0):

- If the RSTEN bit is set in the WD_OMR register, an internal reset is generated.
- If the WDOVF bit is set in the WD_IMR register, an interrupt is generated on the Generic Interrupt Controller (GIC).

Multiple clock sources are available to the watchdog counter (see Figure 31: Watchdog block diagram).

The SYSCLK bit in the WD_MR register is used to select the source input:

- WDSCLK = CORECLK if SYSCLK bit is at a logical 1
- WDSCLK = LFCLK if SYSCLK bit is at a logical 0

If the SYSCLK bit is at a logical 1 (i.e., the CORECLK clock input is selected), the WDSCLK is then divided by the SYSCAL[10:0] divider generating the WDPDIVCLK clock.

If the SYSCLK bit is at a logical 0 (i.e., the LFCLK clock input is selected), the WDPDIVCLK = WDSCLK (i.e., WDPDIVCLK = LFCLK).

The WDPDIVCLK is then divided by the WDPDIV[2:0] divider and provided to the down-counter input WDCLK.

Table 13-1. WD Counter Clock Dividers

WD_MR.3			
SYSCLK	**WDSCLK**	**WDPIDCLK**	**WDCLK**
0	LFCLK	LFCLK	$\frac{\text{LFCLK}}{\text{WDPDIV[2:0]divider}}$
1	CORECLK	$\frac{\text{CORECLK}}{\text{SYSCAL[10:0]divider}}$	$\frac{\text{CORECLK}}{\text{SYSCAL[10:0]divider x WDPDIV[2:0]divider}}$

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).

13.1 Architecture

The WD contains a programmable length down-counter. The count length determines the time-out period, and is controlled by loading the upper bits (15 to 11) of the counter with a programmed value HPCV[4:0]. The lower bits (10 to 0) of the counter are loaded with 0x7FF.

The time-out period (in seconds) is:

$$\frac{\left(\text{HPCV}[4:0]\times 2^{11}\right)+0\text{xFFF}+1}{\text{WDCLK}_{\text{freq}}} = \frac{\left(\text{HPCV}[4:0]\times 2^{11}\right)+2048}{\text{WDCLK}_{\text{freq}}}$$

When the counter reaches 0x00FF, the watchdog can generate a watchdog pending interrupt.

The pending interrupt occurs after:

$$\frac{\left(\text{HPCV}[4:0]\times 2^{11}\right)+1792}{\text{WDCLK}_{\text{freq}}} \text{ seconds}$$

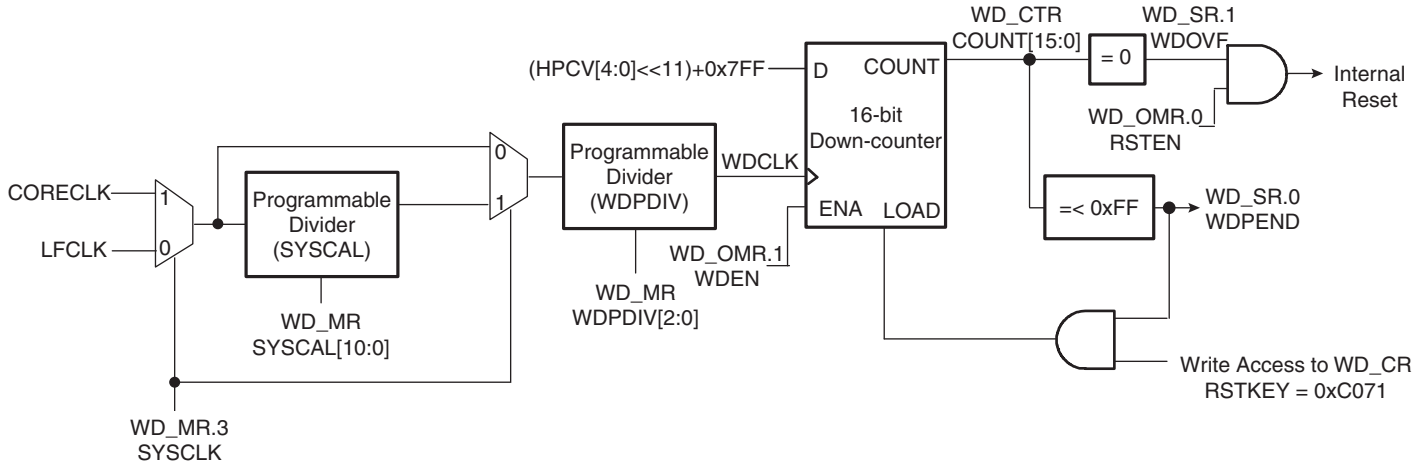
In order to prevent an internal reset (if RSTEN bit is set in the WD_OMR) or interrupt (if bit WDOVF is set in the WD_IMR), the software must reset the counter before it reaches 0 by writing the correct key in the WD_CR register (0xC071). The time (in seconds) between the WD pending interrupt and the WD overflow is:

$$\frac{256}{\text{WDCLK}_{\text{freq}}}$$

When the counter reaches 0, it triggers the programmed action (internal reset or interrupt).

If no WD reset is programmed (i.e., RSTEN is at a logical 0) when the WD reaches 0, it is reset to the programmed value and continues to count, unless it is disabled. This enables it to generate periodic interrupts.

Figure 13-1. Watchdog Block Diagram



13.1.1 Internal Reset Pulse Generation

If the RSTEN bit is set in the WD_OMR register, an internal reset pulse is generated (when the overflow occurs) which is 8 clock cycles long (CORECLK) and which is not affected by hardware reset.

After a reset (hardware or WD) the clock selected by the watchdog is LFCLK/2.

13.1.2 Internal Interrupt Request

The watchdog can generate an internal interrupt request (when the overflow occurs). The software can enable or disable this interrupt either in the WD module (WD_IMR register) or in the GIC registers.

13.2 WD Examples

Conditions: CORECLK = 30 MHz, LFCLK = 32.768 kHz.

13.2.1 SYSCLK = 0

The LFCLK clock source is selected for WDSCLK (i.e., WDSCLK = 32.768 kHz).

Table 13-2. SYSCLK = 0

HPCV[4:0]	WD Counter Start Value	WDPDIV[2:0]	WDCLK	Time to WD Pending	Time to WD Overflow
00000b	0x07FF	000b	LFCLK/2	0.109 s	0.125 s
00001b	0x0FFF	001b	LFCLK/4	0.468 s	0.500 s
00010b	0x17FF	010b	LFCLK/8	1.437 s	1.500 s
10101b	0xAFFF	101b	LFCLK/128	175 s	176 s
11111b	0xFFFF	111b	LFCLK/1024	1040 s	2048 s

13.2.2 SYSCLK = 1

The CORECLK clock source is selected for WDSCLK (i.e., WDSCLK = 30 MHz).

Table 13-3. SYSCLK = 1

HPCV[4:0]	WD Counter Start Value	SYSCAL[10:0]	WDPDIVCLK	WDPDIV[2:0]	WDCLK	Time to WD Pending	Time to WD Overflow
00000b	0x07FF	0x000	CORECLK	000b	WDPDIVCLK/2	119.4 µs	136.5 µs
00001b	0x0FFF	0x001	CORECLK/2	001b	WDPDIVCLK/4	1.024 ms	1.092 ms
00010b	0x17FF	0x100	CORECLK/512	010b	WDPDIVCLK/8	803.9 ms	838.8 ms
10101b	0xAFFF	0x325	CORECLK/1610	101b	WDPDIVCLK/128	307.7 s	309.5 s
11111b	0xFFFF	0x7FF	CORECLK/4094	111b	WDPDIVCLK/1024	9126.8 s	9162.5 s

13.3 Watchdog Timer (WD) Memory Map

Table 13-4. WD Memory Map

Address	Register	Name	Access	Reset State
0xFFFA0000 – 0xFFFA005C	Reserved	---	---	---
0xFFFA0060	Control Register	WD_CR	Write-only	---
0xFFFA0064	Mode Register	WD_MR	Read/Write	0x00000000
0xFFFA0068	Overflow Mode Register	WD_OMR	Read/Write	0x00000000
0xFFFA006C	Clear Status Register	WD_CSR	Write-only	---
0xFFFA0070	Status Register	WD_SR	Read-only	0x00000000
0xFFFA0074	Interrupt Enable Register	WD_IER	Write-only	---
0xFFFA0078	Interrupt Disable Register	WD_IDR	Write-only	---
0xFFFA007C	Interrupt Mask Register	WD_IMR	Read-only	0x00000000
0xFFFA0080	Test Register	WD_TR	Read-only	(see Note 1)
0xFFFA0084	Counter Test Register	WD_CTR	Read-only	0x00000FFF
0xFFFA0088	Preload Test Register	WD_PTR	Write-only	---

Note: 1. The reset value of this register depends on the level of the external pin at reset.

13.3.1 WD Control Register

Name: WD_CR
Access: Write-only
Address: 0xFFFA0060

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[15:8]							

7	6	5	4	3	2	1	0
RSTKEY[7:0]							

- **RSTKEY[15:0]: Restart Key**

0xC071: Watchdog counter is restarted if its value is equal or less than 0x00FF (PENDING = 1 in WD_SR).

Other value: No effect.

13.3.2 WD Mode Register

Name: WD_MR
Access: Read/Write
Address: 0xFFFA0064

31	30	29	28	27	26	25	24
CKEY[7:0]							

23	22	21	20	19	18	17	16
–	–	–	HPCV[4:0]				

15	14	13	12	11	10	9	8
–	SYSCAL[10:4]						

7	6	5	4	3	2	1	0
SYSCAL[3:0]				SYSCLK	WDPDIV[2:0]		

- **WDPDIV[2:0]: WD Clock Divider**

WDPDIV[2:0]			WDCLK
0	0	0	WDPDIVCLK/2
0	0	1	WDPDIVCLK/4
0	1	0	WDPDIVCLK/8
0	1	1	WDPDIVCLK/16
1	0	0	WDPDIVCLK/32
1	0	1	WDPDIVCLK/128
1	1	0	WDPDIVCLK/256
1	1	1	WDPDIVCLK/1024

- **PCV[15:0]: Preload Counter Value**

Counter is preloaded when watchdog counter is restarted.

- **SYSCLK: System Clock Selection**

0: WDSCLK = LFCLK

1: WDSCLK = CORECLK

- **SYSCAL[10:0]: System Clock Prescalar Value**

This prescalar is used to divide the WDSCLK clock when system clock is selected (SYSCLK = 1).

0x000:	WDPDIVCLK = CORECLK
Other value:	$\text{WDPDIVCLK} = \frac{\text{CORECLK}}{(2 \times \text{SYSCAL}[10:0])}$

A write in this field can only be done when system clock is not selected (SYSCLK = 0).

- **HPCV[4:0]: High Preload Counter Value**

Counter is preloaded when watchdog counter is restarted with bits 10 to 0 set to 0x7FF and bits 15 to 11 equal to HPCV[4:0] (i.e., the counter value is (HPCV[4:0] x 2^{11}) + 0x7FF).

- **CKEY[7:0]: Clock Access Key**

Used only when writing in WD_MR. CKEY is read as 0.

Write access in WD_MR is allowed only if CKEY[7:0] = 0x37.

13.3.3 WD Overflow Mode Register

Name: WD_OMR
Access: Read/Write
Address: 0xFFFA0068

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[11:4]							

7	6	5	4	3	2	1	0
OKEY[3:0]				–	–	RSTEN	WDEN

- **WDEN: Watchdog Enable**

0: Watchdog is disabled and does not generate any signals.

1: Watchdog is enabled and does not generate any signals.

- **RSTEN: Reset Enable**

0: Generation of an internal reset by the Watchdog is disabled.

1: When overflow occurs, the Watchdog generates an internal reset.

- **OKEY[11:0]: Overflow Access Key**

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.

13.3.4 WD Clear Status Register

Name: WD_CSR
Access: Write-only
Address: 0xFFFA006C

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	WDPEND

- **WDPEND: Watchdog Pending Clear**

0: No effect.

1: Clear Watchdog pending interrupt.

- **WDOVF: Watchdog Overflow Clear**

0: No effect.

1: Clear Watchdog overflow interrupt.

13.3.5 WD Status Register

Name: WD_SR
Access: Read-only
Address: 0xFFFA0070

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

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

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

7	6	5	4	3	2	1	0
–	–	–	–	–	–	WDOVF	WDPEND

- **WDPEND: Watchdog Pending**

0: No Watchdog pending.

1: A Watchdog pending has occurred.

- **WDOVF: Watchdog Overflow**

0: No Watchdog overflow.

1: A Watchdog overflow has occurred.

- **PENDING: Watchdog Pending Status**

0: Watchdog counter is over 0x00FF (not pending).

1: Watchdog is pending window (between 0x00FF and 0x0000).

13.3.6 WD Interrupt Enable Register

Name: WD_IER
Access: Write-only
Address: 0xFFFA0074

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	WDPEND

13.3.7 WD Interrupt Disable Register

Name: WD_IDR
Access: Write-only
Address: 0xFFFA0078

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	WDPEND

13.3.8 WD Interrupt Mask Register

Name: WD_IMR
Access: Read-only
Address: 0xFFFA007C

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	WDPEND

- **WDPEND: Watchdog Pending Interrupt Mask**

0: The WDPEND interrupt is disabled.

1: The WDPEND interrupt is enabled.

- **WDOVF: Watchdog Overflow Interrupt Mask**

0: The WDOVF interrupt is disabled.

1: The WDOVF interrupt is enabled.

13.3.9 WD Test Register

Name: WD_TR
Access: Read-only
Address: 0xFFFA0080

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
–	–	–	–	–	WDIV	WDRV	TMEN

This register can only be accessed in test mode.

- **TMEN: Test Mode Enabled**

0: Test Mode Enabled.

1: Normal operation of Watchdog.

- **WDRV: Watchdog Reset Value**

When read, this bit contains the value of the watchdog reset output.

Writing to this register has no effect; it does not change the value which is read.

This bit is reset when a reset timeout occurs.

- **WDIV: Watchdog Interrupt Value**

When read, this bit contains the value of the interrupt output.

Writing to this register has no effect, it does not change the value which is read.

This bit is reset when a reset timeout occurs.

13.3.10 WD Counter Test Register

Name: WD_CTR
Access: Read-only
Address: 0xFFFA0084

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

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

15	14	13	12	11	10	9	8
COUNT[15:8]							

7	6	5	4	3	2	1	0
COUNT[7:0]							

This register can only be accessed in test mode.

- **COUNT[15:0]: Counter Value**

Value of Watchdog Counter.

- **RESET: Watchdog Counter Reset**

0: No watchdog counter reset occurred or watchdog counter reset is achieved.

1: A watchdog counter reset is pending.

As the watchdog counter is asynchronous, the write of the right key in WD_CR does not immediately set the watchdog value to initial value. During this delay, this bit is set to one (between reset command and real reset of the counter).

For the same reason, this register must be read twice to be sure of the value.

13.3.11 WD Preload Test Register

Name: WD_PTR
Access: Write-only
Address: 0xFFFA0088

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

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

15	14	13	12	11	10	9	8
CTPR[15:8]							

7	6	5	4	3	2	1	0
CTPR[7:0]							

This register can only be accessed in test mode.

- **CTPR[15:0]: Counter Test Preload Register**

This register is used to preload the initial value of the counter in test mode. It is not readable (if a read is attempted the value is undefined) and it can only be accessed in test mode. After reset, CTPR[15:0] is initialized to 0x0FFF.

14. Watch Timer (WT)

The watch timer provides a second counter and an alarm function.

14.1 Seconds Counter

The seconds counter is a 32-bit counter that indicates the number of low-frequency clock (LFCLK) pulses elapsed since the last time it was reset to zero.

The seconds counter is incremented every low frequency clock (LFCLK) period (one period of 32.768 kHz clock or on period of the MCK/64 clock).

The counter is reset to 0x00000000 when it reaches 0xA8C00000 (86400 seconds or 24 hours when the low frequency clock is the 32.768 kHz) or 0xFFFFFFFF (configurable in the Mode Register).

A write access can only be performed when the seconds counter is disabled because of an asynchronous interface (see ”Asynchronous Interface” on page 90).

14.2 Alarm

The alarm register has the same resolution as the seconds counter. This allows a 32-bit register to have sufficient range to cater for a 24 hour period (when the 32.768 kHz clock is selected).

An interrupt is generated at the end of the period at which the value in the seconds register equals the value in the alarm register.

A write access can only be performed when the alarm counter is disabled because of an asynchronous interface. An invalid data (i.e., value greater or equal to 0xA8C00000) is not written into the alarm register in 24-hour mode.

14.3 Asynchronous Interface

As the seconds counter is an asynchronous counter (can use the RTCK clock), some precautions must be taken with it.

When enabling or disabling the alarm or seconds counter, the software must wait for an enabled or disabled interrupt to be sure that the alarm or the counter is really enabled or disabled.

14.4 CAN Time Stamp

The 32-bit register that forms the seconds counter is provided to the CAN module. After each transmission or reception of a CAN frame, the value of the current seconds counter is automatically written in the corresponding CAN channel CAN_STPx register.

14.5 Watch Timer (WT) Memory Map

Table 14-1. WT Memory Map

Address	Register	Name	Access	Reset State
0xFFFA4000 – 0xFFFA405C	Reserved	---	---	---
0xFFFA4060	Control Register	WT_CR	Write-only	---
0xFFFA4064	Mode Register	WT_MR	Read/Write	0x00000000
0xFFFA4068	Reserved	---	---	---
0xFFFA406C	Clear Status Register	WT_CSR	Write-only	---
0xFFFA4070	Status Register	WT_SR	Read-only	0x00000000
0xFFFA4074	Interrupt Enable Register	WT_IER	Write-only	---
0xFFFA4078	Interrupt Disable Register	WT_IDR	Write-only	---
0xFFFA407C	Interrupt Mask Register	WT_IMR	Read-only	0x00000000
0xFFFA4080	Seconds Register	WT_SECS	Read/Write	0x00000000
0xFFFA4084	Alarm Register	WT_ALARM	Read/Write	0x00000000

14.5.1 WT Control Register

Name: WT_CR
Access: Write-only
Address: 0xFFFA4060

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
–	–	–	ALARMDIS	ALARMEN	SECSDIS	SECSEN	SWRST

- **SWRST: WT Software Reset**

0: No effect.

1: Reset the WT.

A software triggered hardware reset of the WT is performed. It resets all the registers.

- **SECSEN: WT Seconds Counter Enable**

0: No effect.

1: Enables the WT seconds counter.

- **SECSDIS: WT Seconds Counter Disable**

0: No effect.

1: Disables the WT seconds counter.

In case both SECSEN and SECSDIS are equal to one when the control register is written, the WT seconds counter is disabled.

- **ALARMEN: WT Alarm Enable**

0: No effect.

1: Enables the WT alarm.

- **ALARMDIS: WT Alarm Disable**

0: No effect.

1: Disables the WT alarm.

In case both ALARMEN and ALARMDIS are equal to one when the control register is written, the WT alarm is disabled.

14.5.2 WT Mode Register

Name: WT_MR
Access: Read/Write
Address: 0xFFFA4064

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
–	–	–	–	–	–	–	SECRST

- **SECRST: Second Reset**

0: The seconds counter is reset to 0x00000000 at the end of the period when it reaches 0xA8BFFFFF.

1: The seconds counter is reset to 0x00000000 at the end of the period when it reaches 0xFFFFFFFF.

14.5.3 WT Clear Status Register

Name: WT_CSR
Access: Write-only
Address: 0xFFFA406C

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
–	–	–	ALARMDIS	ALARMEN	SECSDIS	SECSEN	ALARM

- **ALARM: Clear Alarm Interrupt**

0: No effect.

1: Clear ALARM interrupt.

- **SECSEN: Clear Seconds Counter Enabled**

0: No effect.

1: Clear the seconds counter enabled interrupt.

- **SECSDIS: Clear Seconds Counter Disabled**

0: No effect.

1: Clear the seconds counter disabled interrupt.

- **ALARMEN: Clear Alarm Enabled**

0: No effect.

1: Clear the alarm enabled interrupt.

- **ALARMDIS: Clear Alarm Disabled**

0: No effect.

1: Clear the alarm disabled interrupt.

14.5.4 WT Status Register

Name: WT_SR
Access: Read-only
Address: 0xFFFA4070

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

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

15	14	13	12	11	10	9	8
–	–	–	–	–	–	ALARMENS	SECENS

7	6	5	4	3	2	1	0
–	–	WSEC	ALARMDIS	ALARMEN	SECSDIS	SECSEN	ALARM

- **ALARM: Alarm Interrupt**

0: No alarm occurred.

1: An alarm occurred since last clear of the status register.

- **SECSEN: Seconds Counter Enabled Interrupt**

0: No seconds counter enabled interrupt.

1: A seconds counter enabled interrupt occurred since last clear of the status register.

- **SECSDIS: Seconds Counter Disabled Interrupt**

0: No seconds counter disabled interrupt.

1: A seconds counter disabled interrupt occurred since last clear of the status register.

- **ALARMEN: Alarm Enabled Interrupt**

0: No alarm enabled interrupt.

1: An alarm enabled interrupt occurred since last clear of the status register.

- **ALARMDIS: Alarm Disabled Interrupt**

0: No alarm disabled interrupt.

1: An alarm disabled interrupt occurred since last clear of the status register.

- **WSEC: Write Second**

0: No effect.

1: A write is occurring on the seconds counter register.

- **SECSENS: Seconds Counter Enable Status**

0: Seconds counter is disabled.

1: Seconds counter is enabled.

- **ALARMENS: Alarm Enable Status**

0: Alarm is disabled.

1: Alarm is enabled.

14.5.5 WT Interrupt Enable Register

Name: WT_IER
Access: Write-only
Address: 0xFFFA4074

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
–	–	–	ALARMDIS	ALARMEN	SECSDIS	SECSEN	ALARM

14.5.6 WT Interrupt Disable Register

Name: WT_IMR
Access: Write-only
Address: 0xFFFA4078

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
–	–	–	ALARMDIS	ALARMEN	SECSDIS	SECSEN	ALARM

14.5.7 WT Interrupt Mask Register

Name: WT_IMR
Access: Read-only
Address: 0xFFFA407C

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
–	–	–	ALARMDIS	ALARMEN	SECSDIS	SECSEN	ALARM

- **ALARM: Alarm Interrupt Mask**

0: ALARM interrupt is disabled.

1: ALARM interrupt is enabled.

- **SECSEN: Seconds Counter Enabled Interrupt Mask**

0: SECSEN interrupt is disabled.

1: Seconds counter enabled interrupt is enabled.

- **SECSDIS: Seconds Counter Disabled Interrupt Mask**

0: SECSDIS interrupt is disabled.

1: SECSDIS interrupt is enabled.

- **ALARMEN: Alarm Enabled Interrupt Mask**

0: ALARMEN interrupt is disabled.

1: ALARMEN interrupt is enabled.

- **ALARMDIS: Alarm Disabled Interrupt Mask**

0: ALARMDIS interrupt is disabled.

1: ALARMDIS interrupt is enabled.

14.5.8 WT Seconds Register

Name: WT_SECS
Access: Read/Write
Address: 0xFFFA4080

31	30	29	28	27	26	25	24
SECONDS[31:24]							

23	22	21	20	19	18	17	16
SECONDS[23:16]							

15	14	13	12	11	10	9	8
SECONDS[15:8]							

7	6	5	4	3	2	1	0
SECONDS[7:0]							

- **SECONDS[31:0]: Seconds Register**

Number of LFCLK clock cycle periods elapsed since last reset to zero.

This register can only be written when SECSENS = 0.

An invalid data (i.e., value greater than or equal to 0xA8C00000) is not written into the seconds register if in 24-hour mode.

14.5.9 WT Alarm Register

Name: WT_ALARM
Access: Read/Write
Address: 0xFFFA0084

31	30	29	28	27	26	25	24
ALARMREG[31:24]							

23	22	21	20	19	18	17	16
ALARMREG[23:16]							

15	14	13	12	11	10	9	8
ALARMREG[15:8]							

7	6	5	4	3	2	1	0
ALARMREG[7:0]							

- **ALARMREG[31:0]: Alarm Register**

An interrupt can be generated when the seconds register reaches this value.

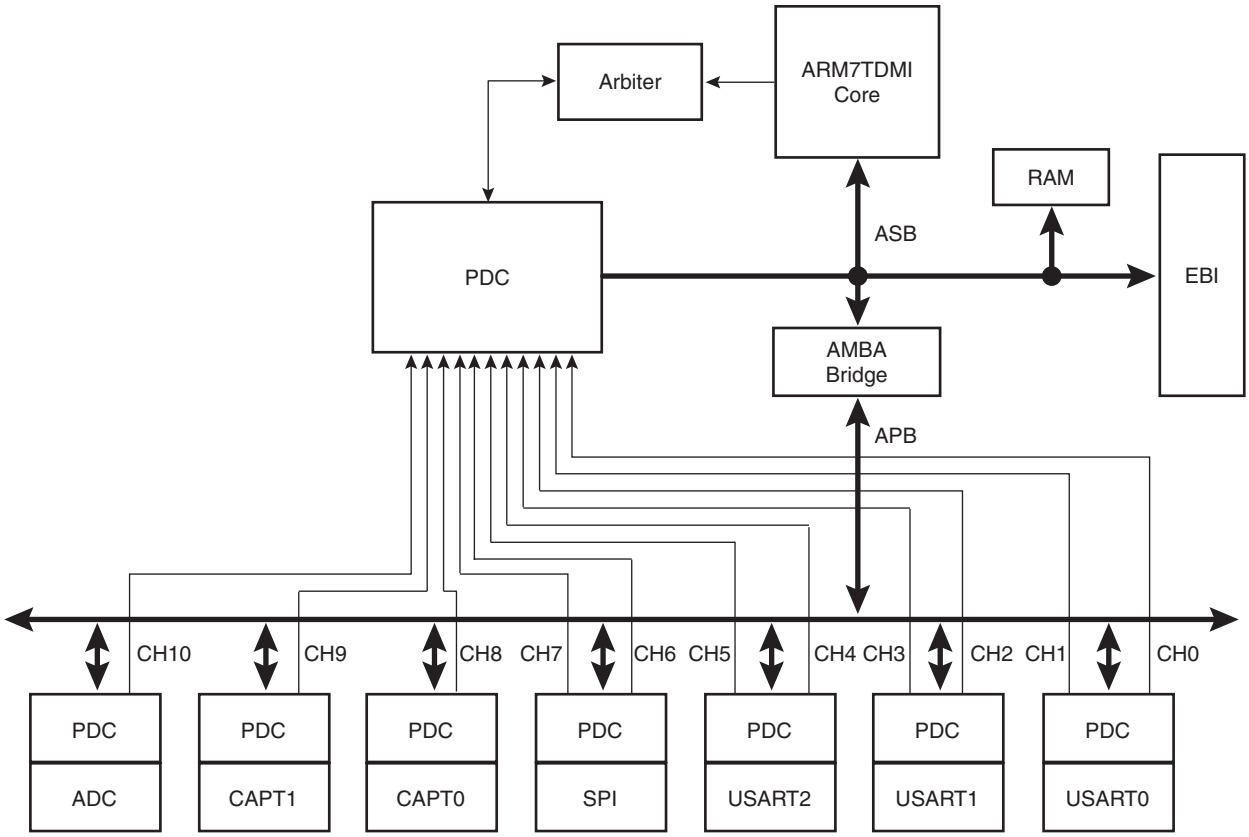
This register can only be written when ALARMENS = 0.

An invalid data (i.e., value greater than or equal to 0xA8C00000) is not written into the alarm register if in 24-hour mode.

15. Peripheral Data Controller (PDC)

The Peripheral Data Controller (PDC) permits easy and quick transfers of large blocks of words between memory to peripheral or between peripheral to memory.

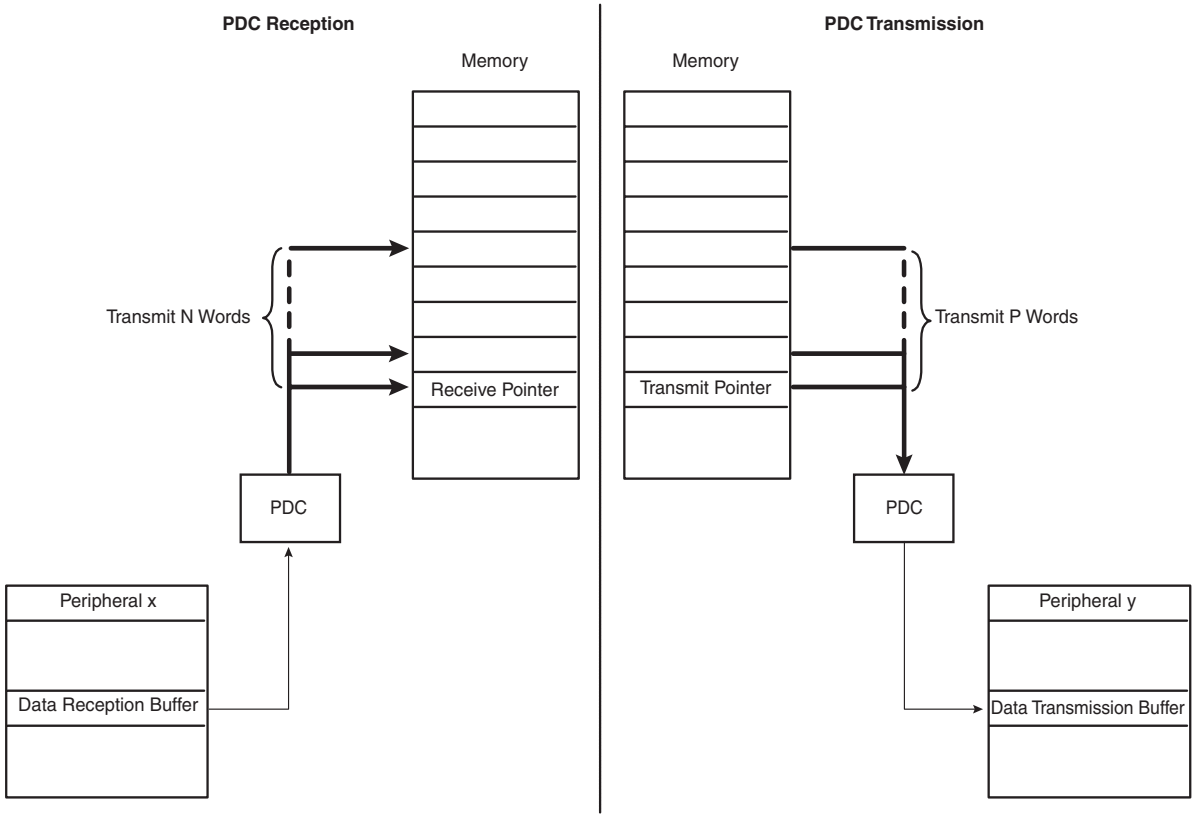
Figure 15-1. PDC Block Diagram



Each PDC channel, with a start address and a counter (number of words), can automatically put/get a block of transmitted words in/from a specific memory area.

The peripherals that are associated to PDC channel are listed in Table 15-1 on page 102.

Figure 15-2. PDC Function



The PDC has data transfer ports which connect to the ASB and the AMBA™ Bridge. It is programmed via the APB. The ASB bus request and grant signals are used to request bus access, and to detect when that access has been granted according to the standard AMBA bus arbitration scheme.

Data transfer to a peripheral is made directly via the APB (AMBA Bridge to avoid tying up the ASB). A simple arbitration scheme is implemented between the ASB and PDC to control APB access.

Before programming any PDC transfer, the PDC module must be enabled. This is done by setting to a logical 1 the PDC bit in the PMC_ECR register (see Power Management Controller (PMC) on page 265).

The PDC channel is programmed using PDC_CRx (Control Register x, x between 0 to 10), PDC_MPRx (Memory Pointer x) and PDC_TCRx (Transfer Counter x).

The status of the PDC transfer is given in the Status Register of the associated peripheral.

The pointer registers (PDC_MPRx) are used to store the address of the buffer.

The counter registers (PDC_TCRx) are used to store the size of these buffers (i.e., the number of data to be transferred).

When a transfer is performed, the counter is decremented and the pointer is incremented. When the counter reaches 0 (i.e., all the data have been sent/receive to/from the module), the end status bit is set in peripheral status register and can be programmed to generate an interrupt.

15.1 PDC Transfers

Each PDC transfer is a byte (8 bits), half-word (16 bits) or word (32 bits) data from peripheral to memory or from memory to peripheral.

Transfers are triggered by the peripheral.

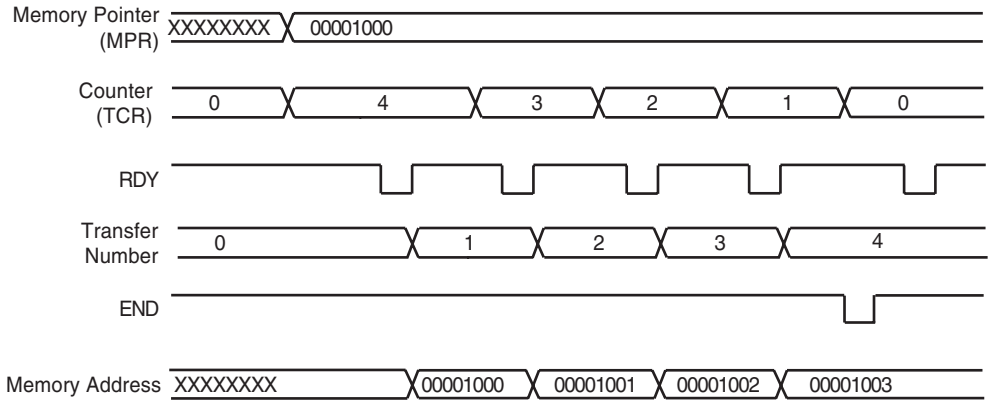
The PDC makes block transfers one byte, one half-word or one word at a time (programmed in the PDC_CRx register, with direction). Each transfer is triggered by a PDC request from the peripheral. The PDC releases the AMBA bus after each transfer. A new trigger is needed for each transfer.

Between transfers, the memory pointer is incremented and the transfer counter is decremented.

Each block of data can be programmed to be up to 64 kbytes.

Transfers stop when the transfer counter reaches zero, and the trigger is disabled.

Figure 15-3. Transfer Example (byte)



15.2 Memory Pointer

There is one 32-bit memory address pointer for each channel (PDC_MPRx). Each memory pointer points to a location in the AT91SAM7A1 memory space (on chip RAM or external memory on the EBI).

The PDC_MPRx is automatically incremented by 1, 2 or 4 after each transfer, for byte, half-word or word transfers. The PDC_MPRx must be initialized before any transfers are started.

If PDC_MPRx is re-programmed while the PDC is operating then the address of the transfers will be changed. The PDC will continue to perform transfers when triggered, from the newly programmed address.

15.3 Transfer Counter

There is one 16-bit Transfer Counter (PDC_TCRx) for each channel, which is used to count the size of the block of transfers. The PDC_TCRx is decremented after every data transfer. When the PDC_TCRx reaches zero the block transfer is complete, the PDC stops transferring data and disables the trigger.

It is not possible to trigger a block of transfers if the PDC_TCRx value is zero. When the PDC_TCRx is programmed to a non-zero value, transfers can be triggered by the peripheral for that particular channel.

The number of transfers required is programmed in the PDC_TCRx which is memory mapped as a 16-bit read/write register. The number of transfers remaining can be read in the PDC_TCRx register.

If the PDC_TCRx is re-programmed while the PDC is operating, the number of transfers will be changed. The PDC will continue to count transfers when triggered, from the newly programmed value.

The end of transfer is signaled to the peripheral via the PDC_END signal. The PDC does not have any dedicated status registers.

While the PDC is operating, in order to stop PDC transfers correctly and to get a fixed value in the PDC_MPRx register, user should write two consecutive '0' values in the PDC_TCRx register. In case the second write is not done and if a PDC transfer has started before the first write in the PDC_TCRx register then PDC_MPRx register value will change during the next core clock periods.

15.4 PDC Configuration

For emulation purposes, each PDC channel can be software configured to be attached to a different peripheral.

In the AT91SAM7A1 microcontroller, each PDC channel is attached to a dedicated peripheral (with a fixed direction and fixed address). Software must configure each PDC channel so the accesses are correctly done by the PDC module:

Table 15-1. PDC Connection

Peripheral	PDC Channel	Transfer Direction	DIR Bit in PDC_CRx	Associated Peripheral Register	Associated Peripheral Address
USART0	RX: Ch0	Reception	0	USART0_RHR	0xFFFA8080
	TX: Ch1	Transmission	1	USART0_THR	0xFFFA8084
USART1	RX: Ch2	Reception	0	USART1_RHR	0xFFFAC080
	TX: Ch3	Transmission	1	USART1_THR	0xFFFAC084
USART2	RX: Ch4	Reception	0	USART2_RHR	0xFFFB0080
	TX: Ch5	Transmission	1	USART2_THR	0xFFFB0084
SPI	RX: Ch6	Reception	0	SPI_RDR	0xFFFB4080
	TX: Ch7	Transmission	1	SPI_TDR	0xFFFB4084
Capture CAPT0	Ch8	Reception	0	CAPT0_DR	0xFFFDC080
Capture CAPT1	Ch9	Reception	0	CAPT1_DR	0xFFFE0080
ADC (8-channel 10-bit)	Ch10	Reception	0	ADC_DR	0xFFFC0080

The end of transmission or reception for each PDC channel transfer is indicated in the status register of the attached peripheral. The PDC_TCRx is decremented with the peripheral trigger

when word has been transferred either from memory to peripheral or from peripheral to memory.

Table 15-2. PDC Transfer Status

Peripheral	PDC Channel	Transfer Direction	Associated Peripheral Status Register	End of Transfer Bit in Status Register	Status Bit for PDC_TCRx Decrement (Trigger)
USART0	RX: Ch0	Reception	USART0_SR	ENDRX	RXRDY
	TX: Ch1	Transmission	USART0_SR	ENDTX	TXRDY
USART1	RX: Ch2	Reception	USART1_SR	ENDRX	RXRDY
	TX: Ch3	Transmission	USART1_SR	ENDTX	TXRDY
USART2	RX: Ch4	Reception	USART2_SR	ENDRX	RXRDY
	TX: Ch5	Transmission	USART2_SR	ENDTX	TXRDY
SPI	RX: Ch6	Reception	SPI_SR	REND	RDRF
	TX: Ch7	Transmission	SPI_SR	TEND	TDRE
Capture CAPT0	Ch8	Reception	CAPT0_SR	PDCEND	DATACAPT
Capture CAPT1	Ch9	Reception	CAPT1_SR	PDCEND	DATACAPT
ADC (8-channel 10-bit)	Ch10	Reception	ADC_SR	TEND	EOC

15.5 PDC Transfer Example

15.5.1 Transmission on SPI

Assuming the following:

- SPI bits per transfer = 10 on NPCS0 (i.e., BITS[3:0] = 0010b in SPI_CRS0)
- Number of 10-bit words to transfer: 15
- Address of buffer in internal RAM for 10-bit words to be transmitted 0x00000100 (first 10-bit word is at address 0x00000100, second 10-bit word is at address 0x00000102, etc.)
- SPI clock is enabled (SPI = 1 in SPI_PMSR)
- SPI is enabled (SPIENS = 1 in SPI_SR)

PDC channel 7 must be configured as follows:

- PDC_PRA7 = 0xFFFB4084 (i.e., address of the SPI_TDR register)
- PDC_CR7 = 0x00000003 (i.e., 10-bit words cater in 16-bit words so PDC transfer size is an half-word incrementing the address pointer by 2 after each transfer, transfers are done from memory to peripheral so DIR = 1)
- PDC_MPR7 = 0x00000100 (address of buffer)
- PDC_TCR7 = 0x0000000F (number of 10-bit words to transfer)

As soon as the software writes the number of bytes to transfer in the PDC_TCR7 register, the PDC starts transmitting the 15 10-bit words.

When all the 10-bit words have been transferred to the SPI_TDR register, the TEND bit in the SPI_SR register will be set to a logical 1 informing the software that the transfer is completed. The TEND bit in the SPI_SR register can also generate an interrupt if the corresponding bit is set in the SPI_IMR register.

15.5.2 Reception on SPI

Assuming the following:

- SPI bits per transfer = 8 on NPCS0 (i.e., BITS[3:0] = 0000b in SPI_CRS0)
- Number of 8-bit words to transfer: 69
- Address of buffer in external RAM for 8-bit words to be transmitted 0x48000000 (first 8-bit word is at address 0x48000000, second 8-bit word is at address 0x48000001, etc.)
- SPI clock is enabled (SPI = 1 in SPI_PMSR)
- SPI is enabled (SPIENS = 1 in SPI_SR)

PDC channel 6 must be configured as follows:

- PDC_PRA6 = 0xFFFB4080 (i.e., address of the SPI_RDR register)
- PDC_CR6 = 0x00000000 (i.e., 8-bit words cater in 8-bit words so PDC transfer size is a byte incrementing the address pointer by 1 after each transfer, transfers are done from peripheral to memory so DIR = 0)
- PDC_MPR6 = 0x48000000 (address of buffer in external RAM)
- PDC_TCR6 = 0x00000045 (number of 8-bit words to transfer)

As soon as the software writes the number of bytes to transfer in the PDC_TCR6 register, the PDC starts receiving the 69 8-bit words.

When all the 8-bit words have been received (i.e., all bytes have been written in external RAM), the REND bit in the SPI_SR register will be set to a logical 1 informing the software that the transfer is completed. The REND bit in the SPI_SR register can also generate an interrupt if the corresponding bit is set in the SPI_IMR register.

15.6 Peripheral Data Controller (PDC) Memory Map

Table 15-3. PDC Memory Map

Address	Register	Name	Access	Reset State
0xFFFF8000 – 0xFFFF807C	Reserved	---	---	---
0xFFFF8080	CH0 Peripheral Register Address	PDC_PRA0	Read/Write	0xFFE00000
0xFFFF8084	CH0 Control Register	PDC_CR0	Read/Write	0x00000000
0xFFFF8088	CH0 Memory Pointer	PDC_MPR0	Read/Write	0x00000000
0xFFFF808C	CH0 Transfer Counter	PDC_TCR0	Read/Write	0x00000000
0xFFFF8090	CH1 Peripheral Register Address	PDC_PRA1	Read/Write	0xFFE00000
0xFFFF8094	CH1 Control Register	PDC_CR1	Read/Write	0x00000000
0xFFFF8098	CH1 Memory Pointer	PDC_MPR1	Read/Write	0x00000000
0xFFFF809C	CH1 Transfer Counter	PDC_TCR1	Read/Write	0x00000000
0xFFFF80A0	CH2 Peripheral Register Address	PDC_PRA2	Read/Write	0xFFE00000
0xFFFF80A4	CH2 Control Register	PDC_CR2	Read/Write	0x00000000
0xFFFF80A8	CH2 Memory Pointer	PDC_MPR2	Read/Write	0x00000000
0xFFFF80AC	CH2 Transfer Counter	PDC_TCR2	Read/Write	0x00000000
0xFFFF80B0	CH3 Peripheral Register Address	PDC_PRA3	Read/Write	0xFFE00000
0xFFFF80B4	CH3 Control Register	PDC_CR3	Read/Write	0x00000000
0xFFFF80B8	CH3 Memory Pointer	PDC_MPR3	Read/Write	0x00000000
0xFFFF80BC	CH3 Transfer Counter	PDC_TCR3	Read/Write	0x00000000
0xFFFF80C0	CH4 Peripheral Register Address	PDC_PRA4	Read/Write	0xFFE00000
0xFFFF80C4	CH4 Control Register	PDC_CR4	Read/Write	0x00000000
0xFFFF80C8	CH4 Memory Pointer	PDC_MPR4	Read/Write	0x00000000
0xFFFF80CC	CH4 Transfer Counter	PDC_TCR4	Read/Write	0x00000000
0xFFFF80D0	CH5 Peripheral Register Address	PDC_PRA5	Read/Write	0xFFE00000
0xFFFF80D4	CH5 Control Register	PDC_CR5	Read/Write	0x00000000
0xFFFF80D8	CH5 Memory Pointer	PDC_MPR5	Read/Write	0x00000000
0xFFFF80DC	CH5 Transfer Counter	PDC_TCR5	Read/Write	0x00000000
0xFFFF80E0	CH6 Peripheral Register Address	PDC_PRA6	Read/Write	0xFFE00000
0xFFFF80E4	CH6 Control Register	PDC_CR6	Read/Write	0x00000000
0xFFFF80E8	CH6 Memory Pointer	PDC_MPR6	Read/Write	0x00000000
0xFFFF80EC	CH6 Transfer Counter	PDC_TCR6	Read/Write	0x00000000
0xFFFF80F0	CH7 Peripheral Register Address	PDC_PRA7	Read/Write	0xFFE00000
0xFFFF80F4	CH7 Control Register	PDC_CR7	Read/Write	0x00000000
0xFFFF80F8	CH7 Memory Pointer	PDC_MPR7	Read/Write	0x00000000
0xFFFF80FC	CH7 Transfer Counter	PDC_TCR7	Read/Write	0x00000000

Table 15-3. PDC Memory Map (Continued)

Address	Register	Name	Access	Reset State
0xFFFF8100	CH8 Peripheral Register Address	PDC_PRA8	Read/Write	0xFFE00000
0xFFFF8104	CH8 Control Register	PDC_CR8	Read/Write	0x00000000
0xFFFF8108	CH8 Memory Pointer	PDC_MPR8	Read/Write	0x00000000
0xFFFF810C	CH8 Transfer Counter	PDC_TCR8	Read/Write	0x00000000
0xFFFF8110	CH9 Peripheral Register Address	PDC_PRA9	Read/Write	0xFFE00000
0xFFFF8114	CH9 Control Register	PDC_CR9	Read/Write	0x00000000
0xFFFF8118	CH9 Memory Pointer	PDC_MPR9	Read/Write	0x00000000
0xFFFF811C	CH9 Transfer Counter	PDC_TCR9	Read/Write	0x00000000
0xFFFF8120	CH10 Peripheral Register Address	PDC_PRA10	Read/Write	0xFFE00000
0xFFFF8124	CH10 Control Register	PDC_CR10	Read/Write	0x00000000
0xFFFF8128	CH10 Memory Pointer	PDC_MPR10	Read/Write	0x00000000
0xFFFF812C	CH10 Transfer Counter	PDC_TCR10	Read/Write	0x00000000
0xFFFF8130 – 0xFFFF8EEC	Reserved	---	---	---
0xFFFF8F00	Test Register	PDC_TR	Write only	---

15.6.1 PDC CHx Peripheral Register Address

Name: PDC_PRAx
Access: Read/Write
Address: 0xFFFF8XX0

31	30	29	28	27	26	25	24
CHPRA[31:24]							

23	22	21	20	19	18	17	16
CHPRA[23:16]							

15	14	13	12	11	10	9	8
CHPRA[15:8]							

7	6	5	4	3	2	1	0
CHPRA[7:0]							

- **CHPRA[31:0] Peripheral Register Address**

CHPRA[31:0] must be loaded with the address of the target register (peripheral receive or transmit register).

15.6.2 PDC CHx Control Register

Name: PDC_CRx
Access: Read/Write
Address: 0xFFFF8XX4

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
–	–	–	–	–	SIZE[1:0]		DIR

- **DIR: Transfer Direction**

0: Peripheral to memory.

1: Memory to peripheral.

- **SIZE[1:0]: Transfer Size**

Defines the size of the transfer.

SIZE[1:0]		Transfer Size
0	0	Byte (8-bit)
0	1	Half-word (16-bit)
1	0	Word (32-bit)
1	1	Reserved

15.6.3 PDC CHx Memory Pointer Register

Name: PDC_MPRx
Access: Read/Write
Address: 0xFFFF8XX8

31	30	29	28	27	26	25	24
CHPTR[31:24]							

23	22	21	20	19	18	17	16
CHPTR[23:16]							

15	14	13	12	11	10	9	8
CHPTR[15:8]							

7	6	5	4	3	2	1	0
CHPTR[7:0]							

- **CHPTR[31:0]: Channel Pointer**

CHPTR[31:0] must be loaded with the address of the target buffer (memory address).

15.6.4 PDC CHx Transfer Counter Register

Name: PDC_TCRx
Access: Read/Write
Address: 0xFFFF8XXC

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

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

15	14	13	12	11	10	9	8
CHCTR[15:8]							

7	6	5	4	3	2	1	0
CHCTR[7:0]							

- **CHCTR[15:0]: Channel Counter**

CHCTR[15:0] must be loaded with the size of the receive buffer.

0: Stop Peripheral Data Transfer dedicated to the peripheral X.

1 to 65535: Start immediately Peripheral Data Transfer.

15.6.5 PDC Test Register

Name: PDC_TR
Access: Write-only
Address: 0xFFFF8F00

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

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

15	14	13	12	11	10	9	8
–	–	–	–	–	TCH10	TCH9	TCH8

7	6	5	4	3	2	1	0
TCH7	TCH6	TCH5	TCH4	TCH3	TCH2	TCH1	TCH0

- **TCHx: Trigger Channel x**

This register is only valid in test mode (see SFM register) and used to emulate a transfer trigger.

0: No effect.

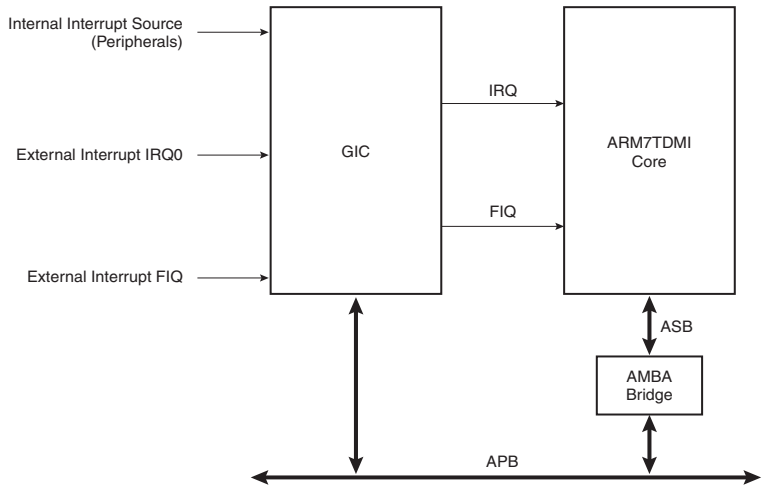
1: Triggers a transfer on channel x.

16. Generic Interrupt Controller (GIC)

The GIC is an 8-level priority, individually maskable, vectored interrupt controller. It can substantially reduce 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 an ARM7TDMI processor. See Figure 16-1. The processor's nFIQ line can only be asserted by the external fast interrupt request input, FIQ. The nIRQ line can be asserted by all others internal and external interrupt sources.

Figure 16-1. GIC Connection to Core



An 8-level priority encoder allows the customer to define the priority between the different nIRQ interrupt sources. The internal interrupt sources are programmed to be level sensitive or edge triggered. The external interrupt sources can be programmed to be positive or negative edge triggered or high or low level sensitive.

Table 16-1. Interrupt Sources

Interrupt Source	Description	GIC Bit
0	Fast interrupt	FIQ
1	Software interrupt 0	SWIRQ0
2	Watchdog	WD
3	Watch Timer	WT
4	USART0	USART0
5	USART1	USART1
6	USART2	USART2
7	SPI	SPI
8	Software interrupt 1	SWIRQ1
9	Software interrupt 2	SWIRQ2
10	ADC0	ADC0
11	Software interrupt 3	SWIRQ3

Table 16-1. Interrupt Sources (Continued)

Interrupt Source	Description	GIC Bit
12	General Purpose Timer 0 channel 0	GPT0CH0
13	General Purpose Timer 0 channel 1	GPT0CH1
14	General Purpose Timer 0 channel 2	GPT0CH2
15	Software interrupt 4	SWIRQ4
16	Pulse Width Modulation	PWM
17	Software interrupt 5	SWIRQ5
18	Software interrupt 6	SWIRQ6
19	Software interrupt 7	SWIRQ7
20	CAN	CAN
21	UPIO	UPIO
22	Capture 0	CAPT0
23	Capture 1	CAPT1
24	Simple Timer 0	ST0
25	Simple Timer 1	ST1
26	Software interrupt 8	SWIRQ8
27	Software interrupt 9	SWIRQ9
28	External interrupt IRQ0	IRQ0
29	Software interrupt 10	SWIRQ10
30	Software interrupt 11	SWIRQ11
31	Software interrupt 12	SWIRQ12

16.1 Interrupt Handling

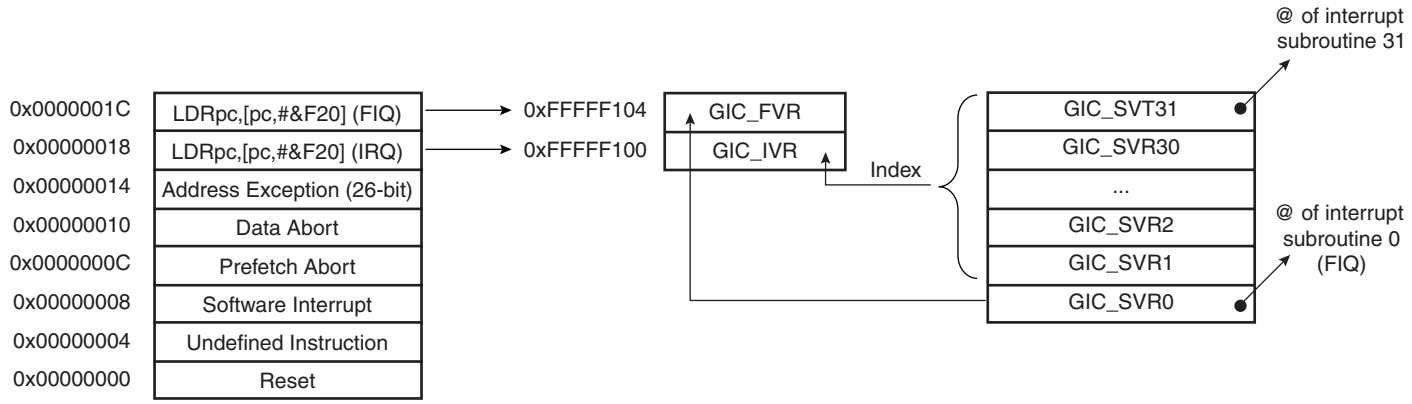
16.1.1 Hardware Interrupt Vectoring

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 GIC_IVR register. Execution is then vectored to the interrupt handler corresponding to the current interrupt. See Figure 16-2.

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

The current interrupt is the interrupt with the highest priority when the Interrupt Vector Register (GIC_IVR) is read. The value read in the GIC_IVR corresponds to the address stored in the Source Vector Register (GIC_SVR) of the current interrupt. Each interrupt source has its corresponding GIC_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 GIC_SVR at system initialization.

Figure 16-2. GIC Automatic Vectoring



16.1.2 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 GIC 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 16-1 on page 111).

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

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

- If the nIRQ line has been asserted but the GIC_IVR has not been read, then the processor reads the new higher priority interrupt handler address in the GIC_IVR register and the current interrupt level is updated.
- If the processor has already read the GIC_IVR then the nIRQ line is reasserted. When the processor has authorized nested interrupts to occur and reads the GIC_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 (GIC_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 GIC returns to the previous state corresponding to the preceding lower priority interrupt which had been interrupted.

16.1.3 Software Interrupt Handling

The interrupt handler must read the GIC_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 GIC 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 (GIC_EOICR) must be written. This allows pending interrupts to be serviced.

16.1.4 Interrupt Masking

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

16.1.5 Interrupt Clearing and Setting

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

16.1.6 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. It can be programmed to be positive or negative edge triggered or high or low level sensitive in the GIC_SMR0 register.

The fast interrupt handler address can be stored in the GIC_SVR0 register. The value written into this register is available by reading the GIC_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 GIC_FVR register.

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

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

16.1.7 Software Interrupt

Any interrupt source of the GIC can be a software interrupt. It must be programmed to be edge triggered in order to set or clear it by writing to the GIC_ISCR and GIC_ICCR. This is totally independent of the SWI instruction of the ARM7TDMI processor.

16.1.8 Spurious Interrupt

A spurious interrupt is a signal of very short duration on one of the interrupt input lines.

16.2 Standard Interrupt Sequence

For details on the registers mentioned in the steps below, refer to the ARM7TDMI Embedded Core Datasheet.

It is assumed that:

- The Generic Interrupt Controller has been programmed, GIC_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, incrementing 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 GIC_IVR. Reading the GIC_IVR has the following effects:

- Sets the current interrupt to be the pending one with the highest priority. The current level is the priority level of the current interrupt.

- De-asserts the nIRQ line on the processor (Even if vectoring is not used, GIC_IVR must be read in order to de-assert nIRQ).

- Automatically clears the interrupt if it has been programmed to be edge triggered, pushes the current level on to the stack, returns the value written in the GIC_SVR corresponding to the current interrupt.

4. The previous step branches 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. The instruction `sub pc, lr, #4` may be used, for example.
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 (GIC_EOICR) must be written in order to indicate to the GIC 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 the 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).

16.3 Fast Interrupt Sequence

For details on the registers mentioned in the steps below, refer to the ARM7TDMI Embedded Core Datasheet.

It is assumed that:

- The Generic Interrupt Controller has been programmed, GIC_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, incrementing 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 GIC_FVR. Reading the GIC_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 the effect of returning from the interrupt 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.

Note: 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).

16.4 Spurious Interrupt Sequence

A spurious interrupt is a signal of very short duration on one of the interrupt input lines. It is handled by the following sequence of actions.

1. When an interrupt is active, the GIC asserts then IRQ (or nFIQ) line and the ARM7TDMI enters IRQ (or FIQ) mode. At this moment, if the interrupt source disappears, the nIRQ (or nFIQ) line is de-asserted but the ARM7TDMI continues with the interrupt handler.
2. If the IRQ Vector Register (GIC_IVR) is read when the nIRQ is not asserted, the GIC_IVR is read with the contents of the Spurious Interrupt Vector Register.
3. If the register FIQ Vector Register (GIC_FVR) is read when the nFIQ is not asserted, the GIC_FVR is read with the contents of the Spurious Interrupt Vector Register.
4. The Spurious Interrupt Routine must at least write into the GIC_EOICR to perform an end of interrupt command. Until the GIC_EOICR write is received by the interrupt controller, the nIRQ (or nFIQ) line is not re-asserted.
5. This causes the ARM7TDMI to jump into the Spurious Interrupt Routine.
6. During a Spurious Interrupt Routine, the Interrupt Status Register GIC_ISR reads 0.

16.5 Generic Interrupt Controller (GIC) Memory Map

Table 16-2. GIC Memory Map

Address	Register	Name	Access	Reset State
0xFFFFF000 – 0xFFFFF07C	GIC Source Mode Register 0 – GIC Source Mode Register 31	GIC_SMR0 – GIC_SMR31	Read/Write	0x00000000
0xFFFFF080 – 0xFFFFF0FC	GIC Source Vector Register 0 – GIC Source Vector Register 31	GIC_SVR0 – GIC_SVR31	Read/Write	0x00000000
0xFFFFF100	GIC IRQ Vector	GIC_IVR	Read-only	0x00000000
0xFFFFF104	GIC FIQ Vector	GIC_FVR	Read-only	0x00000000
0xFFFFF108	GIC Interrupt Status	GIC_ISR	Read-only	0x00000000
0xFFFFF10C	GIC Interrupt Pending	GIC_IPR	Read-only	---
0xFFFFF110	GIC Interrupt Mask	GIC_IMR	Read-only	0x00000000
0xFFFFF114	GIC Core Interrupt Status	GIC_CISR	Read-only	0x00000000
0xFFFFF118 – 0xFFFFF11C	Reserved	---	---	---
0xFFFFF120	GIC Interrupt Enable Command	GIC_IECR	Write-only	---
0xFFFFF124	GIC Interrupt Disable Command	GIC_IDCR	Write-only	---
0xFFFFF128	GIC Interrupt Clear Command	GIC_ICCR	Write-only	---
0xFFFFF12C	GIC Interrupt Set Command	GIC_ISCR	Write-only	---
0xFFFFF130	GIC End of Interrupt Command	GIC_EOICR	Write-only	---
0xFFFFF134	GIC Spurious Vector	GIC_SPU	Read/Write	0x00000000

16.5.1 GIC Source Mode Register

Name: GIC_SMR0 ... GIC_SMR31
Access: Read/Write
Address: 0xFFFFF000 ... 0xFFFFF07C

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
–	SRCTYP[1:0]		–	–	PRIOR[2:0]		

- **PRIOR[2:0]: Priority Level**

These bits program the priority level (from 0-lowest to 7-highest) of all the interrupt sources. The priority level is not used for the FIQ in the SMR0.

- **SRCTYP[1:0]: Interrupt Source Type**

SRCTYP[1:0](1)		Internal Sources	External Sources
0	0	High level sensitive	Low level sensitive
0	1	Positive edge triggered	Negative edge triggered
1	0	High level sensitive	High level sensitive
1	1	Positive edge triggered	Positive edge triggered

Note: 1. All the interrupts used by internal peripherals are considered as internal interrupts and subsequently the SRCTYP1 bit is always read at 0.

16.5.2 GIC Source Vector Register

Name: GIC_SVR0 ... GIC_SVR31
Access: Read/Write
Address: 0xFFFFF080 ... 0xFFFFF0FC

31	30	29	28	27	26	25	24
VECT[31:24]							

23	22	21	20	19	18	17	16
VECT[23:16]							

15	14	13	12	11	10	9	8
VECT[15:8]							

7	6	5	4	3	2	1	0
VECT[7:0]							

- **VECT[31:0]: Interrupt Handler Address**

Address of the corresponding handler for each interrupt source.

16.5.3 GIC Interrupt Vector Register

Name: GIC_IVR
Access: Read-only
Address: 0xFFFFF100

31	30	29	28	27	26	25	24
IRQV[31:24]							

23	22	21	20	19	18	17	16
IRQV[23:16]							

15	14	13	12	11	10	9	8
IRQV[15:8]							

7	6	5	4	3	2	1	0
IRQV[7:0]							

- **IRQV[31:0]: Interrupt Vector Address**

Address of the currently serviced interrupt vector (user programmed in the GIC_SVR register).

Note: GIC_IVR = 0x00000000 when there is no current interrupt.

16.5.4 GIC FIQ Vector Register

Name: GIC_FVR
Access: Read-only
Address: 0xFFFFF104

31	30	29	28	27	26	25	24
FIQV[31:24]							

23	22	21	20	19	18	17	16
FIQV[23:16]							

15	14	13	12	11	10	9	8
FIQV[15:8]							

7	6	5	4	3	2	1	0
FIQV[7:0]							

- **FIQV[31:0]: FIQ Vector Address**

Address of the FIQ serviced interrupt (user programmed in the GIC_SVR0 register).

16.5.5 GIC Interrupt Status Register

Name: GIC_ISR
Access: Read-only
Address: 0xFFFFF108

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[4:0]				

- **IRQID[4:0]: Current IRQ Identifier**

Current interrupt source number.

16.5.6 GIC Interrupt Pending Register

Name: GIC_IPR
Access: Read-only
Address: 0xFFFFF10C

31	30	29	28	27	26	25	24
SWIRQ12	SWIRQ11	SWIRQ10	IRQ0	SWIRQ9	SWIRQ8	ST1	ST0
23	22	21	20	19	18	17	16
CAPT1	CAPT0	UPIO	CAN	SWIRQ7	SWIRQ6	SWIRQ5	PWM
15	14	13	12	11	10	9	8
SWIRQ4	GPT0CH2	GPT0CH1	GPT0CH0	SWIRQ3	ADC0	SWIRQ2	SWIRQ1
7	6	5	4	3	2	1	0
SPI	USART2	USART1	USART0	WT	WD	SWIRQ0	FIQ

- **Interrupt Pending**

0: Corresponding interrupt is inactive.

1: Corresponding interrupt is pending.

16.5.7 GIC Core Interrupt Status Register

Name: GIC_CISR
Access: Read-only
Address: 0xFFFFF114

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

- **NIRQ: NIRQ Status**

0: nIRQ line is inactive.

1: nIRQ line is active.

- **NFIQ: NFIQ Status**

0: nFIQ line is inactive.

1: nFIQ line is active.

16.5.8 GIC Interrupt Enable Command Register

Name: GIC_IECR
Access: Write-only
Address: 0xFFFFF120

31	30	29	28	27	26	25	24
SWIRQ7	SWIRQ6	IRQ1	IRQ0	SWIRQ5	SWIRQ4	ST1	ST0

23	22	21	20	19	18	17	16
CAPT1	CAPT0	UPIO	CAN0	PWM	GPT1CH0	SWIRQ3	SWIRQ2

15	14	13	12	11	10	9	8
SWIRQ1	GPT0CH2	GPT0CH1	GPT0CH0	ADC1	ADC0	CAN2	CAN1

7	6	5	4	3	2	1	0
SPI	CAN3	USART1	USART0	WT	WD	SWIRQ0	FIQ

16.5.9 GIC Interrupt Disable Command Register

Name: GIC_IDCR
Access: Write-only
Address: 0xFFFFF124

31	30	29	28	27	26	25	24
SWIRQ7	SWIRQ6	IRQ1	IRQ0	SWIRQ5	SWIRQ4	ST1	ST0
23	22	21	20	19	18	17	16
CAPT1	CAPT0	UPIO	CAN0	PWM	GPT1CH0	SWIRQ3	SWIRQ2
15	14	13	12	11	10	9	8
SWIRQ1	GPT0CH2	GPT0CH1	GPT0CH0	ADC1	ADC0	CAN2	CAN1
7	6	5	4	3	2	1	0
SPI	CAN3	USART1	USART0	WT	WD	SWIRQ0	FIQ

16.5.10 GIC Interrupt Mask Register

Name: GIC_IMR
Access: Read-only
Address: 0xFFFFF110

31	30	29	28	27	26	25	24
SWIRQ7	SWIRQ6	IRQ1	IRQ0	SWIRQ5	SWIRQ4	ST1	ST0
23	22	21	20	19	18	17	16
CAPT1	CAPT0	UPIO	CAN0	PWM	GPT1CH0	SWIRQ3	SWIRQ2
15	14	13	12	11	10	9	8
SWIRQ1	GPT0CH2	GPT0CH1	GPT0CH0	ADC1	ADC0	CAN2	CAN1
7	6	5	4	3	2	1	0
SPI	CAN3	USART1	USART0	WT	WD	SWIRQ0	FIQ

- **Interrupt Mask**

0: Corresponding interrupt is disabled.

1: Corresponding interrupt is enabled.

16.5.11 GIC Interrupt Clear Command Register

Name: GIC_ICCR
Access: Write-only
Address: 0xFFFFF128

31	30	29	28	27	26	25	24
SWIRQ7	SWIRQ6	IRQ1	IRQ0	SWIRQ5	SWIRQ4	ST1	ST0
23	22	21	20	19	18	17	16
CAPT1	CAPT0	UPIO	CAN0	PWM	GPT1CH0	SWIRQ3	SWIRQ2
15	14	13	12	11	10	9	8
SWIRQ1	GPT0CH2	GPT0CH1	GPT0CH0	ADC1	ADC0	CAN2	CAN1
7	6	5	4	3	2	1	0
SPI	CAN3	USART1	USART0	WT	WD	SWIRQ0	FIQ

- **Software Interrupt Clear**

0: No effect.

1: Clears corresponding software interrupt.

16.5.12 GIC Interrupt Set Command Register

Name: GIC_ICSR
Access: Write-only
Address: 0xFFFFF12C

31	30	29	28	27	26	25	24
SWIRQ7	SWIRQ6	IRQ1	IRQ0	SWIRQ5	SWIRQ4	ST1	ST0
23	22	21	20	19	18	17	16
CAPT1	CAPT0	UPIO	CAN0	PWM	GPT1CH0	SWIRQ3	SWIRQ2
15	14	13	12	11	10	9	8
SWIRQ1	GPT0CH2	GPT0CH1	GPT0CH0	ADC1	ADC0	CAN2	CAN1
7	6	5	4	3	2	1	0
SPI	CAN3	USART1	USART0	WT	WD	SWIRQ0	FIQ

- **Software Interrupt Set**

0: No effect.

1: Sets corresponding software interrupt.

16.5.13 GIC End of Interrupt Command Register

Name: GIC_EOICR
Access: Write-only
Address: 0xFFFFF130

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
–	–	–	–	–	–	–	–

A write access to this register (with any value) indicates that the interrupt treatment is completed.

16.5.14 GIC Spurious Vector Register

Name: GIC_SPU
Access: Read/Write
Address: 0xFFFFF134

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

- **SPUVECT[31:0]: Spurious Interrupt Vector Handler Address**

Address of the spurious interrupt handler.

17. 10-bit Analog-to-digital Converter (ADC)

The 10-bit Analog-to-digital Converter (ADC) can be configured in different modes as follows.

- Single input/simple conversion mode: One input selected and a single conversion.
- Single input/continuous conversion mode: One input selected, the microprocessor or an external command gives the first start to the peripheral which is then completely independent. The PDC can be used to save the resulting data in memory.
  The conversion is stopped in two ways:
  a. By the microprocessor, by setting the STOP bit of the active control register.
  b. By the PDC: TEND can stop the conversion if the STOPEN bit of the mode register is active. This allows the user to order a conversion of a certain number of data without any software intervention.
- Multiple input/simple conversion mode: The user selects which analog inputs are converted and specifies the names of the inputs that are considered by the ADC and in which order they are to be converted. This allows the user to make a single conversion of some of the four inputs in the order of preference. The PDC can be used to save each result.
- Multiple input/continuous conversion mode: Several analog inputs are converted, the microprocessor first starts up the peripheral which then becomes completely independent. The conversion is stopped in two ways:
  – By the microprocessor, by setting the STOP bit of the active control register.
  – By the PDC: TEND can stop the conversion if the STOPEN bit of the mode register is active. This allows the user to order a conversion of a certain number of data without any software intervention.

If the PDC is active, a flag is set when the transfer of all the data is finished.

The converter is composed of a 10-bit cascaded potentiometric digital-to-analog converter connected to the negative input of a Sample and Hold comparator. It is based on a string of 64 polysilicon resistors connected between reference inputs VREF. Thus, the analog input to be converted needs to be in the interval [GNDANA:VREFP].

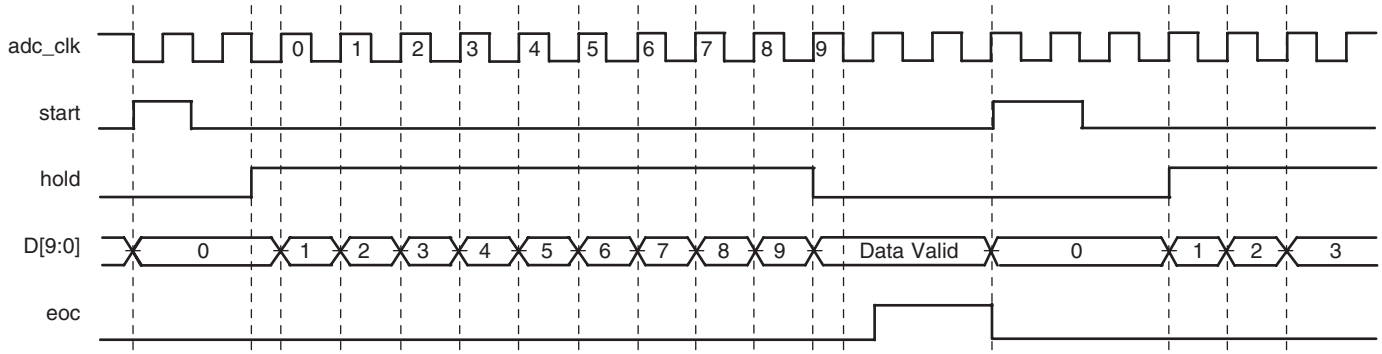
In typical mode, the precision of the ADC10 is about 1 LSB of Integral Non-Linearity (INL) and 0.5 LSB of Differential Non-Linearity (DNL).

In the terminology, the Integral Non-Linearity (INL) is a measure of the maximum deviation from a straight line passing through the end-points of the transfer function. It does not include the full scale error and the zero error. It is calculated from the real value of the LSB which is calculated from the output range (output variation between the minimal value 0 and the maximal one).

Differential Non-linearity (DNL) is the difference between the measured change and the ideal change between any two adjacent codes. A specified DNL of ±1 LSB max over the operating point temperature range ensures monotony. The DNL is relative to the real value of the LSB.

The advantage of such a structure is to offer higher accuracy during the conversion because the resistor bridge can be divided in parts adapted to the input signal. However, both voltage references must be separated by a minimum of 2.4V (i.e., VREFP must be greater than 2.4V according to GNDANA).

Figure 17-1. Signal Description



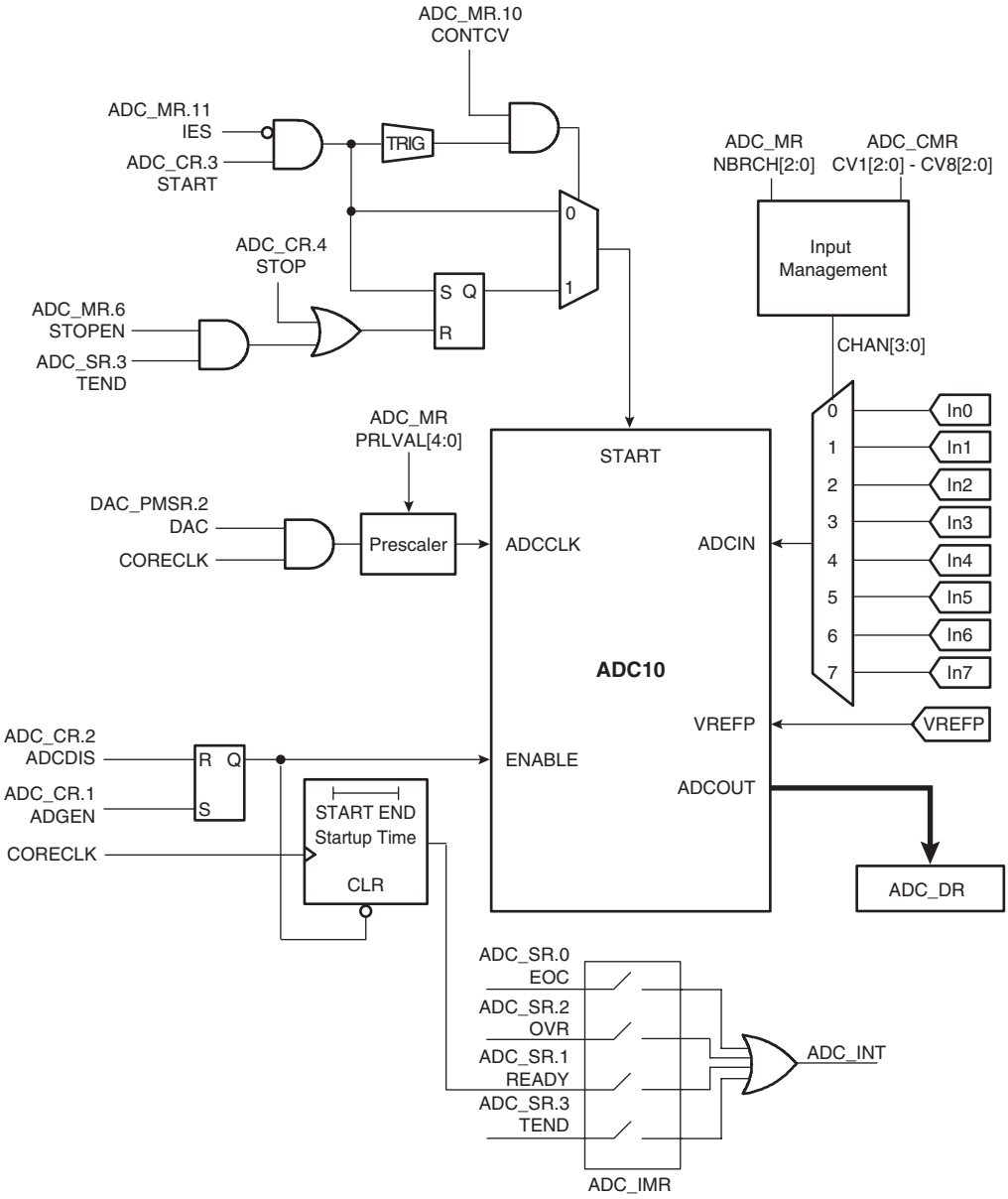
When start is high, the cell is reset and the internal clock is inhibited. The D[9:0] signal is at 512, so the internal ADC output is at (VREFP - GNDana)/2. The hold output is low, so the input voltage at the Sample and Hold stage of the comparator is sampled.

When the start goes low, the hold signal goes high with the next falling clock edge, and the input voltage is stored on the Sample and Hold capacitor. The comparator performs a comparison between the stored input voltage and the ADC output.

With the next rising clock signal, the comparator output becomes valid, this value being stored as D[9] in the internal register. The internal shift register now sets D[8] high, and a new comparison is performed. The next rising edge of clock stores the result in D[8]. After another 8 low-pulse of clock, all 10 bits are valid at output D[9:0]. The End of Conversion signal (EOC) is set high. The input is sampled again as the hold signal goes low and a new conversion can be started with a high-pulse of the start signal.

17.1 Synoptic

Figure 17-2. ADC Block Diagram



17.1.1 Conversion Details

The conversion of a single analog value to 10-bit digital data requires 11 ADC clock cycles comprised of the following:

- One ADC clock cycle to sample the analog input signal.
- Ten ADC clock cycles to fix the ten resulting bits.

The clock input to the ADC has to be lower than 500 kHz. The user can select a frequency by writing the preload value of the counter (PRLVAL[4:0]) to the ADC Mode Register (ADC_MR). The master clock is divided by this value, and the result is the ADC clock. The preload value is coded on 7 bits with the 2 LSBs always low to guarantee a duty cycle of ½. The user divides

the master clock by 0 (ADC clock = CORECLK), 4, 8, …, 48, ..., 64, ..., 124. Thus, the ADC clock is adapted as much as possible to a master clock comprised between 500 kHz and 40 MHz. For example, if CORECLK = 30 MHz with a preload value of 60, the ADC clock is 500 kHz.

A single conversion at the maximum clock rate permitted (i.e., 500 kHz) occurs in 22.0 µs.

The IES bit in the ADC mode register is used to select whether the conversion starts with a software start or with an external trigger start.

The signal that starts the conversion is selected by the IES bit in the ADC Mode Register (ADC_MR).

- If IES = 0, the ADC starts the conversion by writing 1 in the START bit of the control register (ADC_CR).
- If IES = 1, the bit START of the control register is ignored and the conversions starts as soon as a rising edge occurs on the external pin; the signal has to be high during at least one period of the CORECLK, else it is impossible to detect a rising edge.

Writing 1 to the START bit starts the conversion even if the analog structure begins the conversion when start goes low. In this case, the interface transmits the opposite of the start command to the analog part.

For a conversion, different input combinations can be selected. NBRCH[2:0] indicates how many inputs will be converted (the real number is the value of NBRCH incremented by one).

The result of the conversion is stored in the Convert Data Register (ADC_DR). When the conversion is complete, the analog part activates the EOC bit in the ADC Status Register (ADC_SR) and sends an EOC signal to the PDC which then takes the result and writes to a memory location. The EOC bit in ADC_SR (Status Register) is cleared when the ADC_DR (Convert Data Register) is read. If a new result arrives before the PDC or the CPU read the old data, the Overrun bit (OVR) is set active to specify to the microprocessor that data is lost. If the PDC is used to save the results and if the transfer of all the data is finished, the PDC sets the TEND bit to a logical 1. The TEND bit and the OVR bit are reset when the status register (ADC_SR) is read.

The READY bit is set after an absolute time of 4 µs after an enable command, which corresponds to the initialization time of the analog part. This time is necessary to stabilize the analog structure and does not depend on the choice of the ADC clock or the names of analog inputs considered. The number of master clock periods necessary to wait during 4 µs is memorized in the STARTUPTIME bits of the mode register.

The user can make continuous conversions. This status is indicated by the CONTCV bit of the ADC Mode Register (ADC_MR). In this case, the microprocessor or an external command gives the first start to the ADC and the peripheral does not stop the conversion until the STOP bit of the ADC Control Register is set, or when the TEND bit of the status register is set if STOPEN is active. The digital interface between the analog part and the APB bus is in standalone mode; this permits conversion without any help. This mode can be associated with multiple conversion as well as single conversion. The different steps of the conversion are equivalent to those of a single conversion.

17.1.2 Modes of Operation

The ADC can be active or shutdown; in the latter case, it is in a power-saving mode.

At any time, the software can program the ADC to be disabled to save power. Setting the ADCDIS bit of the ADC Control Register puts the ADC analog circuitry into standby mode. To reduce power consumption to nearly 0, the user can switch off the ADC peripheral clock in the Disable Clock Register (ADC_DCR), thus disabling the digital part of the ADC.

When the ADC is re-enabled, a minimum of 4 µs is required before the analog circuitry is stabilized and ready for reliable use. The user has to initialize STARTUPTIME in the ADC_MR register value by indicating how many clock periods of the master clock are necessary to achieve 4 µs.

When the ADC is enabled for the first time (after standby mode but not after wait mode), the interface starts counting and then sets the READY bit in ADC_SR (Status Register). This allows a conversion. The ready flag also indicates if the ADC is converting data (flag is low) or waiting for a start (flag is high).

17.1.2.1 Wait Mode

When the analog part of the ADC peripheral is in standby mode, but the digital part of the peripheral is active, the circuitry is in wait mode. To leave this status, the user can do one of the following:

- Disable the CPU clock. The peripheral is then in standby mode.
- Enable the analog circuitry by setting the ADCEN of the Control Register. The peripheral is active.

17.1.2.2 Standby Mode

When the analog part as well as the digital part are in standby mode, the ADC peripheral is in full standby mode. The digital part is in standby mode when the clock is disabled. The microcontroller disables the peripheral clock by writing to the Disable Clock Register (ADC_DCR) The microcontroller disables the analog part by setting the ADCDIS bit of the control register.

As the ADC structure has a standby mode, the consumption of the analog part is reduced to 1 µW, whereas the DC power dissipation is about 316 µA in typical mode. When standby mode is exited, the ADC peripheral is in wait mode until the ADCEN bit of the control register is set.

17.1.2.3 Warnings

As the standby mode can be effective only after having been in wait mode, a specific order in the different actions must be respected. If the user disables the clock of the interface before disabling the analog part of the peripheral, the peripheral is not in standby mode.

START Mode

The IES bit of the ADC Mode Register indicates if the peripheral is commanded by an internal (given by the microprocessor) start. In this mode, the analog part of the ADC peripheral is waiting for a start.

NBRCH[2:0]

These bits indicate how many inputs are considered by the peripheral for conversion.

CONT Mode

The CONTCV bit of the ADC Mode Register indicates if the peripheral is converting in a continuous mode (in this case the bit is high) or not. This bit is initialized to 0.

17.1.3 Conversion Sequence

The basic sequence of operation after a reset is detailed below.

1. Program the ADC Mode Register: The interface has to know the STARTUPTIME and the PRLVAL before receiving a start command. Determine the names of analog inputs. Indicate if conversion must be continuous (CONTCV = 1).
2. Program the ADC Control Register (ADC_CR) to enable the ADC.
3. Wait for READY bit in ADC_SR. When the flag is set, the ADC is ready to start conversion. An interrupt can be generated to detect when the ADC is ready to start a conversion if the corresponding bit is enabled in the ADC_IMR (Interrupt Mask Register). This flag can be high only after an enable command has been performed because it is this operation which starts the startup time of 4 µs. All commands (even start) before the ready flag are ignored.
4. Initiate conversion start by writing a start command in the ADC Control Register.

If a single conversion is being done, the following applies:

5. The analog input voltage is sampled during 1 ADC clock cycle. The digital result is transferred after 10 ADC clock cycles.
6. Conversion completes after 11 ADC clock cycles from the start command. The digital 10-bit data is latched into ADC_DR (Convert Data Register), the EOC bit in ADC_SR (Status Register) is set.
7. The PDC or the CPU can then read the digital value in ADC_DR, which automatically clears the EOC bit in ADC_SR. Another result can be saved before the data has been read. In this case, the OVR bit is set. When the PDC has stored all the data required, the TEND bit is set.

When multiple conversions or continuous conversions are programmed, steps 5, 6 and 7 above are repeated.

17.2 Power Management

The ADC is provided with a power management block allowing optimization of power consumption. See “Power Management Block” on page 23.

17.3 10-bit Analog to Digital Converter (ADC) Memory Map

Table 17-1. ADC Memory Map

Address	Register	Name	Access	Reset State
0xFFFC0000 – 0xFFFC004C	Reserved	–	–	–
0xFFFC0050	Enable Clock Register	ADC_ECR	Write-only	–
0xFFFC0054	Disable Clock Register	ADC_DCR	Write-only	–
0xFFFC0058	Power Management Status Register	ADC_PMSR	Read-only	0x00000000
0xFFFC005C	Reserved	–	–	–
0xFFFC0060	Control Register	ADC_CR	Write-only	–
0xFFFC0064	Mode Register	ADC_MR	Read/Write	0x00000000
0xFFFC0068	Conversion Mode Register	ADC_CMR	Read/Write	0x00000000
0xFFFC006C	Clear Status Register	ADC_CSR	Write-only	–
0xFFFC0070	Status Register	ADC_SR	Read-only	0x00000000
0xFFFC0074	Interrupt Enable Register	ADC_IER	Write-only	–
0xFFFC0078	Interrupt Disable Register	ADC_IDR	Write-only	–
0xFFFC007C	Interrupt Mask Register	ADC_IMR	Read-only	0x00000000
0xFFFC0080	Convert Data Register	ADC_DR	Read-only	0x00000000
0xFFFC0084 – 0xFFFC008C	Reserved	–	–	–
0xFFFC0090	Test Mode Register	ADC_TSTR	Read/write	0x00000000

17.3.1 ADC Enable Clock Register

Name: ADC_ECR
Access: Write-only
Address: 0xFFFC0050

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
–	–	–	–	–	–	ADC	–

17.3.2 ADC Disable Clock Register

Name: ADC_DCR
Access: Write-only
Address: 0xFFFC0054

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
–	–	–	–	–	–	ADC	–

- **ADC: ADC Clock Status**

0: ADC clock disabled.

1: ADC clock enabled.

17.3.3 ADC Power Management Status Register

Name: ADC_PMSR
Access: Read-only
Address: 0xFFFC0058

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
–	–	–	–	–	–	ADC	–

- **ADC: ADC Clock Status**

0: ADC clock disabled.

1: ADC clock enabled.

17.3.4 ADC Control Register

Name: ADC_CR
Access: Write-only
Address: 0xFFFC0060

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
–	–	–	STOP	START	ADCDIS	ADCEN	SWRST

- **SWRST: ADC Software Reset**

0: No effect.

1: Reset of the ADC peripheral.

When a software reset is performed, all the registers of the peripheral are reset.

- **ADCEN: ADC Enable**

0: No effect.

1: ADC is enabled for conversion.

- **ADCDIS: ADC Disable**

0: No effect.

1: ADC is disabled (Standby Mode).

- **START: Start Conversion**

0: No analog-to-digital conversion to be started.

1: Begin analog-to-digital conversion, clears EOC bit.

This bit is ignored if IES bit of the Mode Register is high.

- **STOP: Stop Conversion in Continuous Conversion**

0: No effect.

1: Stop the continuous conversion.

17.3.5 ADC Mode Register

Name: ADC_MR
Access: Read/Write
Address: 0xFFFC0064

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

23	22	21	20	19	18	17	16
–	–	–	–	CONTCV	NBRCH[2:0]		

15	14	13	12	11	10	9	8
STARTUPTIME[7:0]							

7	6	5	4	3	2	1	0
–	STOPEN	–	PRLVAL[4:0]				

- **PRLVAL[4:0]: Preload Value**

A division of the CORECLK determines ADC_clk. The preload value is chosen by the user to adapt the Master clock to the ADC peripheral as closely as possible. It is the start value of the down counter. The LSB is fixed to 0 because this value has to be even parity to guarantee a duty cycle of ½, so the user has only to initialize the 5 MSB.

ADC_clk = CORECLK/(4xPRLVAL[4:0]) if PRLVAL[4:0] ≠ 0.

ADC_clk = CORECLK/2 if PRLVAL[4:0] = 0.

Note: The clock rate to the ADC must not exceed 500 kHz.

- **STOPEN: Stop Enable**

0: TEND cannot stop conversion when the device is in continuous mode.

1: TEND stops conversion when the device is in continuous conversion mode.

This bit is initialized to 0.

- **STARTUPTIME[7:0]: Startup Time**

This value indicates the number of master clock periods necessary to make 4 µs.

This time is the stabilization time of the analog circuitry.

For example, if CORECLK = 30 MHz, 120 periods of CORECLK are needed to obtain the absolute time of 4 µs. Thus, the STARTUPTIME value of the mode register is 120 (0x78 in hexadecimal code).

- **NBRCH[2:0]: Number of Conversions**

NBRCH[2:0]	Number of Conversions
000	1
001	2
010	3
011	4
100	5
101	6
110	7
111	8

- **CONTCV: Continuous Conversion**

0: Single conversion mode.

1: Continuous conversion mode.

This bit is initialized to 0.

17.3.6 ADC Conversion Mode Register

Name: ADC_CMR
Access: Read/Write
Address: 0xFFFC0068

31	30	29	28	27	26	25	24
–	CV8[2:0]			–	CV7[2:0]		

23	22	21	20	19	18	17	16
–	CV6[2:0]			–	CV5[2:0]		

15	14	13	12	11	10	9	8
–	CV4[2:0]			–	CV3[2:0]		

7	6	5	4	3	2	1	0
–	CV2[2:0]			–	CV1[2:0]		

- **CVx[2:0]: Input Selection**

x = {1, 2, 3, 4, 5, 6, 7, 8} is the conversion number.

CVx[2:0]	Input Selected
000	In0
001	In1
010	In2
011	In3
100	In4
101	In5
110	In6
111	In7

17.3.7 ADC Clear Status Register

Name: ADC_CSR
Access: Write-only
Address: 0xFFFC006C

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
–	–	–	–	TEND	OVR	–	–

- **OVR: Overrun Interrupt**

0: No effect.

1: Clear OVR interrupt.

- **TEND: End of PDC Transfer Interrupt**

0: No effect.

1: Clear TEND interrupt.

17.3.8 ADC Status Register

Name: ADC_CSR
Access: Read-only
Address: 0xFFFC0070

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

- **EOC: End of Conversion**

0: Conversion not complete or inactive.

1: Conversion complete, data in ADC_DR is valid.

This bit is cleared when the ADC_DR is read.

- **READY: ADC Ready for Conversion**

0: ADC ignores start or stop command. It is not ready for conversion or is converting data.

1: ADC is ready to start a conversion.

To explain “ready_flag” more clearly, define “working” as the event high when ADC is converting data and “analog_ready” as the event low when the analog part is disabled or in the initialization phase.

analog_ready	working	ready_flag
0	0	0
0	1	0
1	0	1
1	1	0

- **OVR: Overrun**

0: No data has been transferred by ADC from the last ADC_DR read.

1: At least one data has been transferred by ADC since the last ADC_DR read.

- **TEND: End of Total Transfer of PDC**

0: Transfer of all data not complete.

1: PDC transfer complete.

This bit is set when the transfer of all the data by the PDC is complete. When we are in continuous mode, it stops the conversion only if the STOPEN bit of the mode register is active.

- **ADCENS: ADC Enable Status**

0: ADC is disabled.

1: ADC is enabled.

- **CTCVS: Continuous Mode Status**

0: Single conversion with help of microprocessor.

1: Continuous conversion, the peripheral is stand-alone.

This bit is initialized to 0 and changes when there is a change of mode. This bit never generates an interrupt.

17.3.9 ADC Interrupt Enable Register

Name: ADC_IER
Access: Write-only
Address: 0xFFFC0074

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
–	–	–	–	TEND	OVR	READY	EOC

17.3.10 ADC Interrupt Disable Register

Name: ADC_IDR
Access: Write-only
Address: 0xFFFC0078

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
–	–	–	–	TEND	OVR	READY	EOC

17.3.11 ADC Interrupt Mask Register

Name: ADC_IMR
Access: Read-only
Address: 0xFFFC007C

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
–	–	–	–	TEND	OVR	READY	EOC

- **EOC: End of Conversion**

0: EOC interrupt is disabled.

1: EOC interrupt is enabled.

- **READY: ADC Ready for Conversion**

0: READY interrupt is disabled.

1: READY interrupt is enabled.

- **OVR: Overrun**

0: OVR interrupt is disabled.

1: OVR interrupt is enabled.

- **TEND: End of PDC Transfer**

0: TEND interrupt is disabled.

1: TEND interrupt is enabled.

17.3.12 ADC Convert Data Register

Name: ADC_CDR
Access: Read-only
Address: 0xFFFC0080

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

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

15	14	13	12	11	10	9	8
–	–	–	–	–	–	DATA[9:8]	

7	6	5	4	3	2	1	0
DATA[7:0]							

- **DATA[9:0]: Converted Data**

The resulting data from an analog to digital conversion is latched into this register at the end of a conversion and remains valid until a new conversion is completed.

When this register is read, the EOC bit in the ADC_SR register is cleared.

17.3.13 ADC Test Mode Register

Name: ADC_TSTR
Access: Read/write
Address: 0xFFFC0090

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
–	–	–	–	–	–	–	TEST

- **TEST: Test Mode**

0: Normal mode

1: Test mode

TEST mode must be set at to logical 1 in the SFM.

18. Universal Synchronous/Asynchronous Receiver/Transmitter (USART)

The AT91SAM7A1 includes three USARTs. Each transmitter and receiver module is connected to the Peripheral Data Controller.

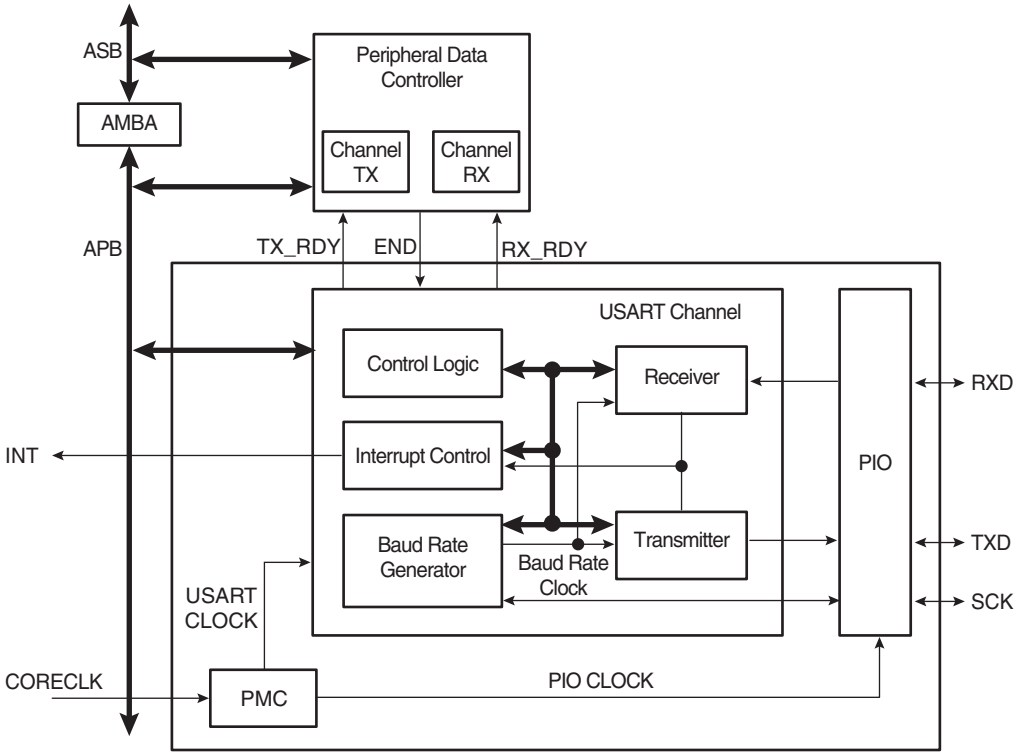
18.1 Description

The main features are:

- Programmable baud rate generator
- Parity, framing and overrun error detection
- Supports Hardware LIN protocol (specification 1.2)
- Idle flag for J1587 protocol
- Line break generation and detection
- Automatic echo, local loopback and remote loopback channel modes
- Multidrop mode: address detection and generation
- Interrupt generation
- Two dedicated Peripheral Data Controller channels per USART
- 5- to 9-bit character length

18.2 Synoptic

Figure 18-1. USART Synoptic



18.3 Baud Rate Generator

The baud rate generator provides the bit period clock, also called 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 CORECLK or CORECLK divided by 8 (CORECLK/8).

In all cases, if an external clock is used, the duration of each of its levels must be longer than the CORECLK period. The external clock frequency must be less than 40% of CORECLK frequency.

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.

$$\text{Baud rate} = \text{Selected Clock}/16 \times \text{CD}$$

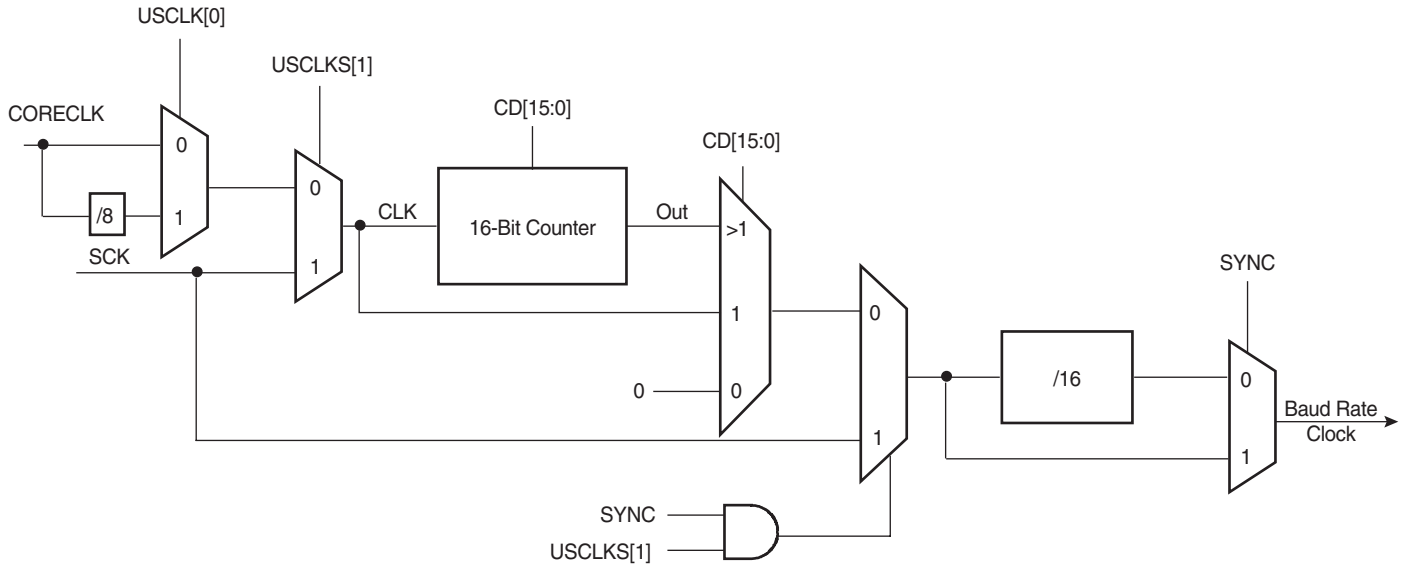
when the selected clock is either CORECLK, CORECLK/8 or SCK.

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.

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

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 Receivers

The USART is configured with two receiver operating modes, one for asynchronous operations and the other for synchronous operations.

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 longer than 7/16 of the bit period is detected as a valid start bit. A space that 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 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. Each subsequent bit is sampled 16 cycles (1 bit period) after the previous one.

Figure 18-3. Asynchronous Mode Start Bit Detection

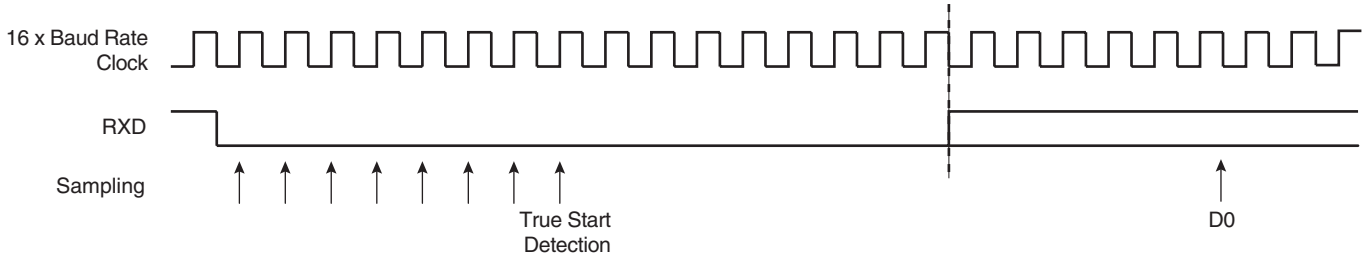
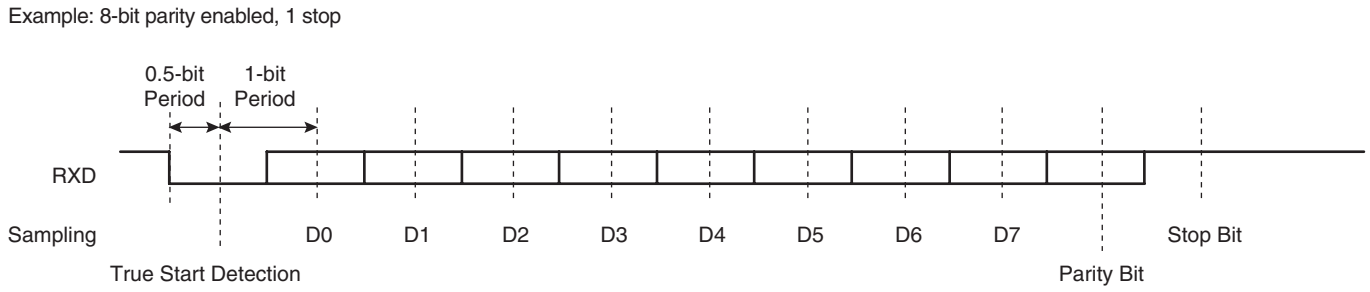


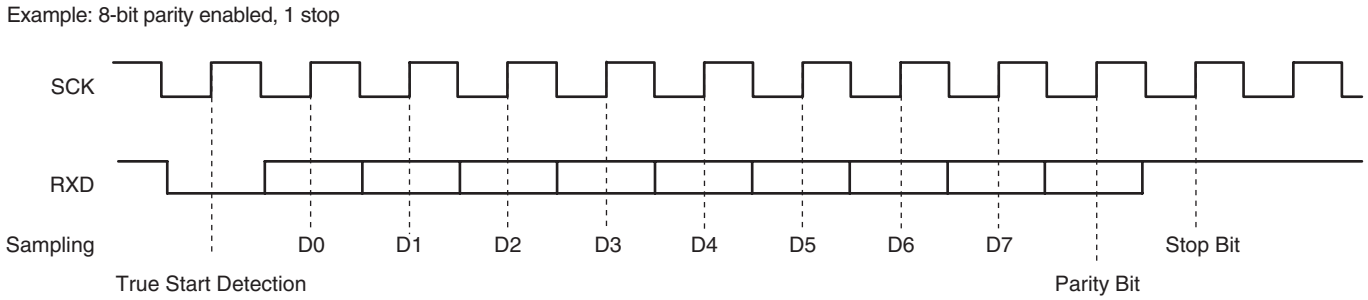
Figure 18-4. Asynchronous Mode Character Reception



18.4.2 Synchronous Receiver

When configured for synchronous operation (SYNC = 1), the receiver samples the RXD signal on each rising edge of SCK. 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 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_SR is set. The RXRDY is set after the last stop bit.

If US_RHR has not been read since the last transfer, the USOVRE status bit in US_SR 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 PAR field in US_MR. It then compares the result with the received parity bit. If different, the parity error bit PARE in US_SR is set.

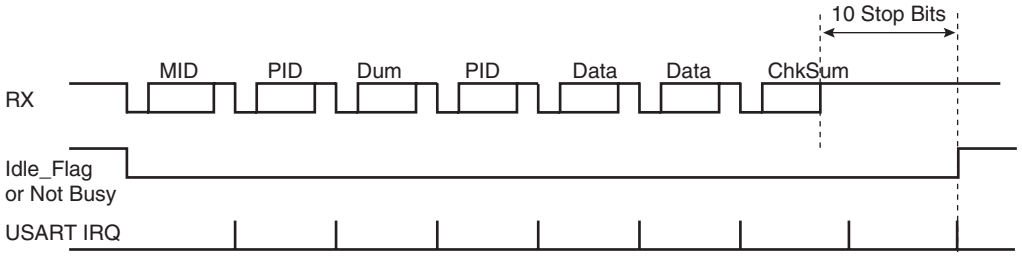
18.4.5 Framing Error

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_SR.

18.4.6 Idle Flag for J1587 Protocol Frame

The idle flag turns low when the USART receives a start bit and turns high at the end of a J1587 protocol frame (after 10 stop bits). An interrupt can be generated on the rising edge of the idle flag.

Figure 18-6. Idle Flag



18.4.7 Time-out

The time-out 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 Register). 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_SR is set. The user starts (or restarts) the wait for a first character by setting the STTTO (start time-out) bit in US_CR.

To start a time-out, the following conditions must be met:

- US_RTOR must not be equal to 0.
- The time-out must be started by setting STTTO to a logical 1 in the US_CR register.
- One character must be received.

Moreover, it should be noted that writing STTTO in US_CR is taken into account under the two following conditions:

- Baud rate is set to a value different from '0' (in order to clock the receiver state machine)
- Receiver state machine is reset

Calculation of time-out duration in asynchronous mode:

Duration = Value x 4 x 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, least significant bit first, on the falling edge of the serial clock.

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 Register), it is transferred to the Shift Register as soon as it is empty.

When the transfer occurs, the TXRDY bit in US_SR is set until a new character is written to US_THR. If the Transmit Shift Register and US_THR are both empty, the TXEMPTY bit in US_SR is set (after the last stop bit of the last transfer).

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 Register). 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.

18.5.2 Multidrop Mode

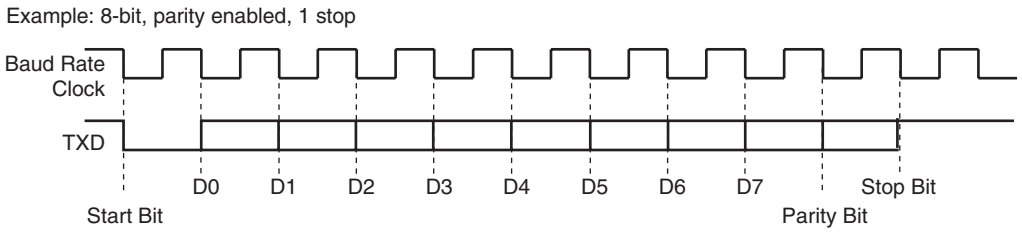
When the PAR field in US_MR equals $11X_b$, the USART is configured to run in multidrop mode. In this case, the parity error bit (PARE in US_SR) is set when data is detected with a parity bit set to identify an address byte. PARE is cleared with the Reset Status Bits Command (RSTSTA) in US_CR. If the 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 has the parity bit cleared.

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

Figure 18-7. Synchronous and Asynchronous Modes, Character Transmission



18.6 Break Condition

18.6.1 Transmit Break

The transmitter can generate a break condition on the TXD line when the STTBRK command is set in US_CR (Control Register). In this case, the characters present in US_THR and in the Transmit Shift Register are completed before the line is held low.

To remove this break condition on the TXD line, the STPBRK command in US_CR must be set. The USART generates 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. The transmitter then resumes normal operation.

18.6.2 Receive Break

The break condition is detected by the receiver when all data, parity and stop bits are low. At the moment of the low stop bit detection, the receiver asserts the RXBRK bit in US_SR.

The end of receive break is detected by a high level for at least 2/16 of the bit period in asynchronous operating mode or at least one sample in synchronous operating mode. RXBRK is also set after end of break has been detected.

18.6.3 Interrupt Generation

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

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

18.6.4 Channel Modes

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

Automatic Echo Mode provides bit-by-bit retransmission. 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 enables 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 provides bit by bit retransmission.

18.7 PIO Controller

Each USART has three programmable I/O lines. These I/O lines are multiplexed with signals (TXD, RXD, SCK) of the USART to optimize the use of available package pins. These lines are controlled by the USART PIO controller.

18.8 Power Management

The USART is provided with a power management block that optimizes power consumption. See ”Power Consumption” on page 9.

18.9 USART Memory Map

Base Address USART0: 0xFFFA8000

Base Address USART1: 0xFFFAC000

Base Address USART2: 0xFFFB0000

Table 18-1. USART Memory Map

Offset	Register	Name	Access	Reset State
0x000	PIO Enable Register	US_PER	Write-only	–
0x004	PIO Disable Register	US_PDR	Write-only	–
0x008	PIO Status Register	US_PSR	Read-only	0x00070000
0x00C	Reserved	–	–	–
0x010	Output Enable Register	US_OER	Write-only	–
0x014	Output Disable Register	US_ODR	Write-only	–
0x018	Output Status Register	US_OSR	Read-only	0x00000000
0x01C – 0x02C	Reserved	–	–	–
0x030	Set Output Data Register	US_SODR	Write-only	–
0x034	Clear Output Data Register	US_CODR	Write-only	–
0x038	Output Data Status Register	US_ODSR	Read-only	0x00000000
0x03C	Pin Data Status Register	US_PDSR	Read-only	0x000X0000
0x040	Multidriver Enable Register	US_MDER	Write-only	–
0x044	Multidriver Disable Register	US_MDDR	Write-only	–
0x048	Multidriver Status Register	US_MDSR	Read-only	0x00000000
0x04C	Reserved	–	–	–
0x050	Enable Clock Register	US_ECR	Write-only	–
0x054	Disable Clock Register	US_DCR	Write-only	–
0x058	Power Management Status Register	US_PMSR	Read-only	0x00000000
0x05C	Reserved	–	–	–
0x060	Control Register	US_CR	Write-only	0x00000000
0x064	Mode Register	US_MR	Read/Write	0x00000000
0x068	Reserved	–	–	–
0x06C	Reserved	–	–	–
0x070	Status Register	US_SR	Read-only	0x00000800
0x074	Interrupt Enable Register	US_IER	Write-only	–
0x078	Interrupt Disable Register	US_IDR	Write-only	–
0x07C	Interrupt Mask Register	US_IMR	Read-only	0x00000000
0x080	Receiver Holding Register	US_RHR	Read-only	0x00000000
0x084	Transmitter Holding Register	US_THR	Write-only	–

Table 18-1. USART Memory Map

Offset	Register	Name	Access	Reset State
0x088	Baud Rate Generator Register	US_BRGR	Read/Write	0x00000000
0x08C	Receiver Time-out Register	US_RTOR	Read/Write	0x00000000
0x090	Transmitter Time-guard Register	US_TTGR	Read/Write	0x00000000

18.9.1 USART PIO Enable Register

Name: US_PER
Access: Write-only
Offset: 0x000

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

18.9.2 USART PIO Disable Register

Name: US_PDR
Access: Write-only
Offset: 0x004

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

18.9.3 USART PIO Status Register

Name: US_PSR
Access: Read-only
Offset: 0x008

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

- **SCK: SCK Pin**

0: PIO is inactive on the SCK pin.

1: PIO is active on the SCK pin.

- **TXD: TXD Pin**

0: PIO is inactive on the TXD pin.

1: PIO is active on the TXD pin.

- **RXD: RXD Pin**

0: PIO is inactive on the RXD pin.

1: PIO is active on the RXD pin.

18.9.4 USART PIO Output Enable Register

Name: US_OER
Access: Write-only
Offset: 0x010

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

18.9.5 USART PIO Output Disable Register

Name: US_ODR
Access: Write-only
Offset: 0x014

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

18.9.6 USART PIO Output Status Register

Name: US_OSR
Access: Read-only
Offset: 0x018

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

23	22	21	20	19	18	17	16
–	–	–	–	–	RXD	TXD	SCK

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SCK: SCK Pin**

0: PIO is an input on the SCK pin.

1: PIO is an output on the SCK pin.

- **TXD: TXD Pin**

0: PIO is an input on the TXD pin.

1: PIO is an output on the TXD pin.

- **RXD: RXD Pin**

0: PIO is an input on the RXD pin.

1: PIO is an output on the RXD pin.

18.9.7 USART PIO Set Output Data Register

Name: US_SODR
Access: Write-only
Offset: 0x030

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

18.9.8 USART PIO Clear Output Data Register

Name: US_CODR
Access: Write-only
Offset: 0x034

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

18.9.9 USART PIO Output Data Status Register

Name: US_ODSR
Access: Read-only
Offset: 0x038

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

- **SCK: SCK Pin**

0: The output data for the SCK pin is programmed to 0.

1: The output data for the SCK pin is programmed to 1.

- **TXD: TXD Pin**

0: The output data for the TXD pin is programmed to 0.

1: The output data for the TXD pin is programmed to 1.

- **RXD: RXD Pin**

0: The output data for the RXD pin is programmed to 0.

1: The output data for the RXD pin is programmed to 1.

18.9.10 USART PIO Pin Data Status Register

Name: US_PDSR
Access: Read-only
Offset: 0x03C

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

23	22	21	20	19	18	17	16
–	–	–	–	–	RXD	TXD	SCK

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SCK: SCK Pin**

0: The SCK pin is at logic 0.

1: The SCK pin is at logic 1.

- **TXD: TXD Pin**

0: The TXD pin is at logic 0.

1: The TXD pin is at logic 1.

- **RXD: RXD Pin**

0: The RXD pin is at logic 0.

1: The RXD pin is at logic 1.

18.9.11 USART PIO Multi Drive Enable Register

Name: US_MDER
Access: Write-only
Offset: 0x040

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

18.9.12 USART PIO Multi Drive Disable Register

Name: US_MDDR
Access: Write-only
Offset: 0x044

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

18.9.13 USART PIO Multi Drive Status Register

Name: US_MDSR
Access: Read-only
Offset: 0x048

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

- **SCK: SCK Pin**

0: The SCK pin is not configured as an open drain.

1: The SCK pin is configured as an open drain.

- **TXD: TXD Pin**

0: The TXD pin is not configured as an open drain.

1: The TXD pin is configured as an open drain.

- **RXD: RXD Pin**

0: The RXD pin is not configured as an open drain.

1: The RXD pin is configured as an open drain.

18.9.14 USART Enable Clock Register

Name: US_ECR
Access: Write-only
Offset: 0x050

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
–	–	–	–	–	–	USART	PIO

18.9.15 USART Disable Clock Register

Name: US_DCR
Access: Write-only
Offset: 0x054

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
–	–	–	–	–	–	USART	PIO

18.9.16 USART Power Management Status Register

Name: US_PMSR
Access: Read-only
Offset: 0x058

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
–	–	–	–	–	–	USART	PIO

- **PIO: PIO Clock Status**

0: PIO clock disabled.

1: PIO clock enabled.

- **USART: USART Clock Status**

0: USART clock disabled.

1: USART clock enabled.

Note: The US_PMSR register is not reset by software reset.

18.9.17 USART Control Register

Name: US_CR
Access: Write-only
Offset: 0x060

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	–	SWRST

- **SWRST: Software Reset**

0: No effect.

1: Reset the USART.

A software triggered hardware reset of the USART is performed. It resets all the registers, including PIO registers (except US_PMSR).

- **RSTRX: Reset Receiver**

0: No effect.

1: The receiver logic is reset.

The internal reset is synchronized with the baud rate clock and is maintained internally for one baud rate clock period. Thus no byte can be received in the two baud rate clock periods following the receiver reset (this can be avoided by setting the baud rate clock to 0 and then to the desired value after a receiver reset).

- **RSTTX: Reset Transmitter**

0: No effect.

1: The transmitter logic is reset.

The internal reset is synchronized with the baud rate clock and is maintained internally for one baud rate clock period. Thus no byte can be transmitted in the two baud rate clock periods following the transmitter reset (this can be avoided by setting the baud rate clock to 0 and then to the desired value after a receiver reset).

- **RXEN: Receiver Enable**

0: No effect.

1: The receiver is enabled if RXDIS is 0.

- **RXDIS: Receiver Disable**

0: No effect.

1: The receiver is disabled.

- **TXEN: Transmitter Enable**

0: No effect.

1: The transmitter is enabled if TXDIS is 0.

- **TXDIS: Transmitter Disable**

0: No effect.

1: The transmitter is disabled.

- **RSTSTA: Reset Status Bit**

0: No effect.

1: Resets the status bits PARE, FRAME, USOVRE and RXBRK in US_SR.

- **STTBRK: Start Break**

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**

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**

0: No effect.

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

- **SENDA: Send Address**

0: No effect.

1: In Multidrop Mode only, the next character written to the US_THR is sent with the address bit set.

18.9.18 USART Mode Register

Name: US_MR
Access: Read/Write
Offset: 0x064

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[1:0]		NBSTOP[1:0]		PAR[2:0]			SYNC

7	6	5	4	3	2	1	0
CHRL[1:0]		USCLKS[1:0]		–	–	–	–

- **USCLKS[1:0]: Clock Selection (Baud Rate Generator Input Clock)**

USCLKS[1:0]		Selected Clock
0	0	CORECLK
0	1	CORECLK/8
1	X	External (SCK)

- **CHRL[1:0]: Character Length**

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

CHRL[1:0]		Character Length
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits

- **SYNC: Synchronous Mode Select**

0: USART operates in Asynchronous Mode.

1: USART operates in Synchronous Mode.

- **PAR[2:0]: Parity Type**

PAR[2:0]			Parity Type
0	0	0	Even Parity
0	0	1	Odd Parity
0	1	0	Parity forced to 0 (Space)
0	1	1	Parity forced to 1 (Mark)
1	0	x	No parity
1	1	x	Multidrop mode

- **NBSTOP[1:0]: Number of Stop Bits**

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

NBSTOP[1:0]		Asynchronous (SYNC = 0)	Synchronous (SYNC = 1)
0	0	1 stop bit	1 stop bit
0	1	1.5 stop bits	Reserved
1	0	2 stop bits	2 stop bits
1	1	Reserved	Reserved

- **CHMODE[1:0]: Channel Mode**

CHMODE[1:0]		Mode Description
0	0	Normal Mode: The USART Channel operates as a Rx/Tx USART
0	1	Automatic Echo: Receiver Data Input is connected to TXD pin
1	0	Local Loopback: Transmitter Output Signal is connected to Receiver Input Signal
1	1	Remote Loopback: RXD pin is internally connected to TXD pin

- **MODE9: 9-bit Character Length**

0: CHRL defines character length.

1: 9-bit character length.

- **CLKO: Clock Output Select**

0: The USART does not drive the SCK pin.

1: The USART drives the SCK pin if CLKS[1] is 0.

18.9.19 USART Status Register

Name: US_SR
Access: Read-only
Offset: 0x070

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

23	22	21	20	19	18	17	16
–	–	–	–	–	RXD	TXD	SCK

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

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

Note: This register is a “read-active” register, which means that reading it can affect the state of some bits. When reading US_SR register, the following bits are cleared if set: IDLE, ENDRX, ENDTX, SCK, TXD and RXD.

- **RXRDY: Receiver Ready**

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**

0: A character is in US_THR waiting to be transferred to the Transmit Shift Register or the transmitter is disabled.

1: There is no character in US_THR.

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**

0: No Break Received and no 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 PDC Receiver Transfer**

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 PDC Transmitter Transfer**

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.

- **USOVRE: Overrun Error**

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**

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**

0: No parity bit has been detected false (or a parity bit high in multidrop mode) since the last “Reset Status Bits” command.

1: At least one parity bit has been detected false (or a parity bit high in multidrop mode) since the last “Reset Status Bits” command.

- **TIMEOUT: Receiver Time-out**

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**

0: There are characters in either US_THR or the Transmit Shift Register.

1: There are no characters in either US_THR or the Transmit Shift Register.

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

- **IDLE: Idle Interrupt**

0: No end of J1587 protocol frame since last read of the US_SR register.

1: An end of J1587 protocol frame occurred since last read of the US_SR register.

- **IDLEFLAG: Idle Flag**

0: A frame is being received by the USART.

1: No frame is being received by the USART.

This bit indicates a frame transmission in J1587 protocol. It is turned low when a reception starts and turned high when a reception is followed by at least 10 stop bits (10 bits at high level).

- **SCK, TXD, RXD: PIO Interrupt Status**

These bits indicate for each pin when an input logic value change has been detected (rising or falling edge). This is valid whether the PIO is selected for the pin or not.

These bits are reset to zero following a read and at reset.

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

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

18.9.20 USART Interrupt Enable Register

Name: US_IER
Access: Write-only
Offset: 0x074

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

18.9.21 USART Interrupt Disable Register

Name: US_IDR
Access: Write-only
Offset: 0x078

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

18.9.22 USART Interrupt Mask Register

Name: US_IMR
Access: Read-only
Offset: 0x07C

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

- **RXRDY: Mask RXRDY Interrupt**

0: RXRDY Interrupt is disabled.

1: RXRDY Interrupt is enabled.

- **TXRDY: Mask TXRDY Interrupt**

0: TXRDY Interrupt is disabled.

1: TXRDY Interrupt is enabled.

- **RXBRK: Mask Receiver Break Interrupt**

0: RXBRK Interrupt is disabled.

1: RXBRK Interrupt is enabled.

- **ENDRX: Mask End of PDC Receive Transfer Interrupt**

0: ENDRX Interrupt is disabled.

1: ENDRX Interrupt is enabled.

- **ENDTX: Mask End of PDC Transmit Transfer Interrupt**

0: ENDTX Interrupt is disabled.

1: ENDTX Interrupt is enabled.

- **USOVRE: Mask Overrun Error Interrupt**

0: USOVRE Interrupt is disabled.

1: USOVRE Interrupt is enabled.

- **FRAME: Mask Framing Error Interrupt**

0: FRAME Interrupt is disabled.

1: FRAME Interrupt is enabled.

- **PARE: Mask Parity Error Interrupt**

0: PARE Interrupt is disabled.

1: PARE Interrupt is enabled.

- **TIMEOUT: Mask Time-out Interrupt**

0: TIMEOUT Interrupt is disabled.

1: TIMEOUT Interrupt is enabled.

- **TXEMPTY: Mask TXEMPTY Interrupt**

0: TXEMPTY Interrupt is disabled.

1: TXEMPTY Interrupt is enabled.

- **IDLE: Mask IDLE Interrupt**

0: IDLE Interrupt is disabled.

1: IDLE Interrupt is enabled.

- **SCK, TXD, RXD: PIO Interrupt Mask**

These bits show which pins have interrupts enabled. They are updated by writing to US_IER or US_IDR.

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

1: Interrupt is enabled on the corresponding input pin.

18.9.23 USART Receiver Holding Register

Name: US_RHR
Access: Read-only
Offset: 0x080

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

- **RXCHR[8:0]: Received Character**

Last character received if RXRDY is set. When the number of data bits is less than 9, the bits are right aligned.

18.9.24 USART Transmit Holding Register

Name: US_THR
Access: Write-only
Offset: 0x084

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

- **TXCHR[8:0]: Character to be Transmitted**

Next character to be transmitted after the current character if TXRDY is not set. When the number of data bits is less than 9, the bits are right aligned.

18.9.25 USART Baud Rate Generator Register

Name: US_BRGR
Access: Read/Write
Offset: 0x088

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[15:8]							

7	6	5	4	3	2	1	0
CD[7:0]							

- **CD[15:0]: Clock divisor**

This register has no effect if synchronous mode is selected with an external clock.

CD[15:0]	Action
0	Disables clock
1	Clock divider bypassed
2 to 65535	Baud Rate (Asynchronous Mode) = Selected clock/(16 x CD) Baud Rate (Synchronous Mode) = Selected clock/CD

Notes: 1. In synchronous mode, the value programmed must be even to ensure a 50:50 mark/space ratio.
2. CD = 1 must not be used when internal clock (CORECLK) is selected (i.e., USCLKS[1:0] = 00_b).

18.9.26 USART Receiver Time-out Register

Name: US_RTOR
Access: Read/Write
Offset: 0x08C

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[7:0]							

- **TO[7:0]: Time-out Value**

TO[7:0]	Action
0	Disables the RX Time-out function.
1 - 255	The Time-out counter is loaded with TO[7:0] when the Start Time-out Command is given or when each new data character is received (after reception has started).

Time-out duration = TO[7:0] x 4 x Bit period.

When the receiver is disabled by setting the RXDIS bit in the US_CR register, the time-out is stopped. If the receiver is re-enabled by setting the RXEN bit in the US_CR register, the time-out restarts where it left off (i.e., it is not reset).

18.9.27 USART Transmit Time-guard Register

Name: US_TTGR
Access: Read/Write
Offset: 0x090

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[7:0]							

- **TG[7:0]: Time-guard Value**

TG[7:0]	Action
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.

19. Capture (CAPT)

The AT91SAM7A1 provides a Capture module that serves as a frame analyzer. It stores the duration period (high and low level) in a register; these durations are described as a number of counter cycles (CAPTCLK). The Capture peripheral provides data transfer with the PDC.

Capture can detect all frame variations (with an error of one CAPTCLK period maximum) if the input signal has a frequency less than CAPTCLK/2.

It is possible to choose among three modes of measurement:

- Duration between both edges (positive and negative)
- Duration between positive edges
- Duration between negative edges

It is important to check that the frame to be watched by the capture module is not too fast, otherwise the PDC can monopolize the ASB bus, as the PDC, in this case, often has something to read on the Capture Data Register (CAPT_DR).

If an overrun occurs, it is possible to overwrite the data stored in the Data Register (CAPT_DR) or to stop the data acquisition through the mode register (CAPT_MR).

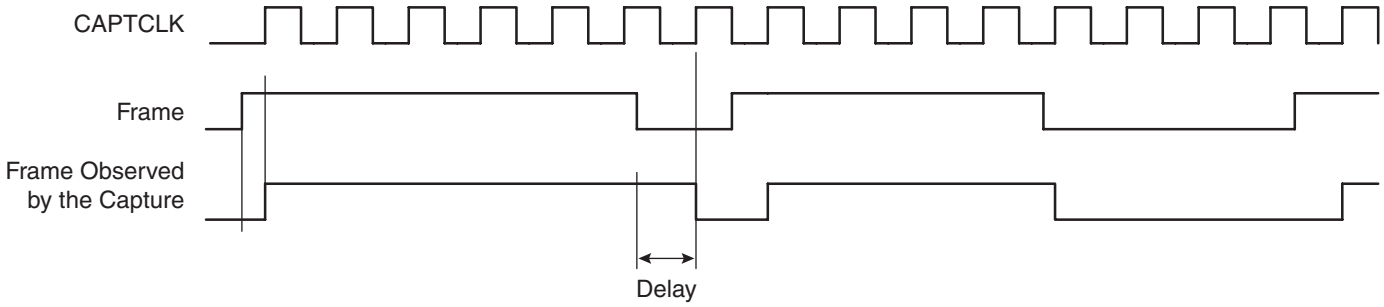
After a read of the data register, the DATACAPT bit (in the CAPT_SR register) is automatically cleared (i.e., set to logical 0).

It is recommended to disable the Capture module after every modification of the CAPT_MR register, otherwise the first measurement can be false.

When Capture is disabled, the capture counter is reset.

19.1 Example of Use

Figure 19-1. Capture Pin Resynchronization



19.2 Capture Limits

To prevent the capture from missing frame edges, it is important to check that:

$$\text{Frame frequency} \leq \text{CAPTCLK}/2$$

Moreover, the capture detects each frame edge with a delay equal to the CAPTCLK period.

The CAPTCLK frequency can range from 30 MHz and 916 Hz (if 30 MHz is the CORECLK frequency).

Table 19-1. Capture Delay and Error (Example)

CAPTCLK Frequency	Fastest Observable Frame	Delay Max for Edge Detection	Error Max in Level Duration Value
F	F/2 if F< CORECLK frequency F/4 if F= CORECLK frequency	1/F	2/F
30 MHz	7.5 MHz	33 ns	66 ns
916 Hz	458 Hz	1.1 ms	2.2 ms

19.3 PIO Controller

Each capture has one programmable I/O line. The I/O line is multiplexed with the capture input signal to optimize the use of available package pins. The line is controlled by the capture PIO controller.

19.4 Power Management

Each capture module (CAPT0 and CAPT1) is provided with a power management block allowing optimization of power consumption. See “Power Management Block” on page 23.

19.5 Capture (CAPT) Memory Map

Base Address CAPT0: 0xFFFDC000

Base Address CAPT1: 0xFFFE0000

Table 19-2. CAPT Memory Map

Offset	Register	Name	Access	Reset State
0x000	PIO Enable Register	CAPT_PER	Write-only	---
0x004	PIO Disable Register	CAPT_PDR	Write-only	---
0x008	PIO Status Register	CAPT_PSR	Read-only	0x00010000
0x00C	Reserved	---	---	---
0x010	Output Enable Register	CAPT_OER	Write-only	---
0x014	Output Disable Register	CAPT_ODR	Write-only	---
0x018	Output Status Register	CAPT_OSR	Read-only	0x00000000
0x01C – 0x02C	Reserved	---	---	---
0x030	Set Output Data Register	CAPT_SODR	Write-only	---
0x034	Clear Output Data Register	CAPT_CODR	Write-only	---
0x038	Output Data Status Register	CAPT_ODSR	Read-only	0x00000000
0x03C	Pin Data Status Register	CAPT_PDSR	Read-only	0x000X0000
0x040	Multi-Driver Enable Register	CAPT_MDER	Write-only	---
0x044	Multi-Driver Disable Register	CAPT_MDDR	Write-only	---
0x048	Multi-Driver Status Register	CAPT_MDSR	Read-only	0x00000000
0x04C	Reserved	---	---	---
0x000 – 0x04C	Reserved	---	---	---
0x050	Enable Clock Register	CAP_ECR	Write-only	–
0x054	Disable Clock Register	CAP_DCR	Write-only	–
0x058	Power Management Status Register	CAP_PMSR	Read-only	0x00000000
0x05C	Reserved	–	–	–
0x060	Control Register	CAP_CR	Write-only	–
0x064	Mode Register	CAP_MR	Read/Write	0x00000000
0x068	Reserved	---	---	---
0x06C	Clear Status Register	CAP_CSR	Write-only	–
0x070	Status Register	CAP_SR	Read-only	0x00000000
0x074	Interrupt Enable Register	CAP_IER	Write-only	–
0x078	Interrupt Disable Register	CAP_IDR	Write-only	–
0x07C	Interrupt Mask Register	CAP_IMR	Read-only	0x00000000
0x080	Data Register	CAP_DR	Read-only	0x00000000

19.5.1 CAPTURE PIO Enable Register

Name: CAPT_PER
Access: Write-only
Offset: 0x000

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

19.5.2 CAPTURE PIO Disable Register

Name: CAPT_PDR
Access: Write-only
Offset: 0x004

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

19.5.3 CAPTURE PIO Status Register

Name: CAPT_PSR
Access: Read-only
Offset: 0x008

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

- **CAPTPIN: Capture Pin**

0: The PIO control of the capture pin is disabled (the pin works in the capture mode).

1: The PIO control of the capture pin is enabled (the pin works as a PIO).

19.5.4 CAPTURE PIO Output Enable Register

Name: CAPT_OER
Access: Write-only
Offset: 0x010

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

19.5.5 CAPTURE PIO Output Disable Register

Name: CAPT_ODR
Access: Write-only
Offset: 0x014

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

19.5.6 CAPTURE PIO Output Status Register

Name: CAPT_OSR
Access: Read-only
Offset: 0x018

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

- **CAPTPIN: Capture Pin**

0: The PIO output is disabled on the capture pin.

1: The PIO output is enabled on the capture pin.

19.5.7 CAPTURE PIO Set Output Data Register

Name: CAPT_SODR
Access: Write-only
Offset: 0x030

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

19.5.8 CAPTURE PIO Clear Output Data Register

Name: CAPT_CODR
Access: Write-only
Offset: 0x034

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

19.5.9 CAPTURE PIO Output Data Status Register

Name: CAPT_ODSR
Access: Read-only
Offset: 0x038

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

- **CAPTPIN: Capture Pin**

0: The PIO output is set to logical 0 on the capture pin.

1: The PIO output is set to logical 1 on the capture pin.

19.5.10 CAPTURE PIO Pin Data Status Register

Name: CAPT_PDSR
Access: Read-only
Offset: 0x03C

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

- **CAPTPIN: Capture Pin**

0: The capture pin is at a logical 0.

1: The capture pin is at a logical 1.

19.5.11 CAPTURE PIO Multi-drive Enable Register

Name: CAPT_MDER
Access: Write-only
Offset: 0x040

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

19.5.12 CAPTURE PIO Multi-drive Disable Register

Name: CAPT_MDDR
Access: Write-only
Offset: 0x044

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

19.5.13 CAPTURE PIO Multi-drive Status Register

Name: CAPT_MDSR
Access: Read-only
Offset: 0x048

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

- **CAPTPIN: Capture Pin**

0: The capture pin is not configured as an open drain.

1: The capture pin is configured as an open drain.

19.5.14 CAPTURE Enable Clock Register

Name: CAPT_ECR
Access: Write-only
Offset: 0x050

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
–	–	–	–	–	–	CAP	PIO

19.5.15 CAPTURE Disable Clock Register

Name: CAPT_DCR
Access: Write-only
Offset: 0x054

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
–	–	–	–	–	–	CAP	PIO

19.5.16 CAPTURE Power Management Status Register

Name: CAPT_PMSR
Access: Read-only
Offset: 0x058

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
–	–	–	–	–	–	CAP	PIO

- **PIO: PIO Clock**

0: PIO clock disabled.

1: PIO clock enabled.

- **CAP: CAPTURE Clock**

0: Capture clock disabled.

1: Capture clock enabled.

Note: The CAPT_PMSR register is not reset by a software reset.

19.5.17 CAPTURE Control Register

Name: CAPT_CR
Access: Write-only
Offset: 0x060

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
–	–	–	–	STARTCAPT	CAPDIS	CAPEN	SWRST

- **SWRST: Capture Software Reset**

0: No effect.

1: Resets the capture.

A software reset of the capture is performed. It resets all the registers.

- **CAPEN: Capture Enable**

0: No effect.

1: Enables the capture.

- **CAPDIS: Capture Disable**

0: No effect.

1: Disables the Capture.

If both CAPEN and CAPDIS are equal to one when the control register is written, the CAPTURE is disabled.

- **STARTCAPT: Start Capture**

0: No effect.

1: The capture starts a new capture.

19.5.18 CAPTURE Mode Register

Name: CAPT_MR
Access: Read/Write
Offset: 0x064

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
ONESHOT	OVERMODE	MEASMODE[1:0]		PRESCALAR[3:0]			

- **PRESCALAR[3:0]: Counter Clock Prescalar**

$$\text{CAPTCLK Frequency} = \text{CORECLK} / 2^{PRESCALAR-1}$$

- **MEASMODE[1:0]: Measurement Mode**

MEASMODE[1:0]		Measure
0	X	Measure between each edge (positive and negative)
1	0	Measure between positive edges
1	1	Measure between negative edges

- **OVERMODE: Overrun Mode**

0: In case of an overrun, the capture stops writing on the Data Register.

1: In case of an overrun, the capture does not stop writing on the Data Register.

- **ONESHOT: One Shot**

0: The capture still captures a frame variation.

1: The module captures a frame variation and stops. To ask for another capture, the STARTCAPT bit has to be set at 1 in CAPT_CR.

19.5.19 CAPTURE Clear Status Register

Name: CAPT_CSR
Access: Write-only
Offset: 0x06C

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

- **PDCEND: Clear PDCEND Interrupt**

0: No effect.

1: Clears the PDCEND interrupt.

- **OVERRUN: Clear Overrun Interrupt**

0: No effect.

1: Clears the OVERRUN interrupt.

- **OVERFLOW: Clear Overflow Interrupt**

0: No effect.

1: Clears the OVERFLOW interrupt.

- **CAPTPIN: Capture Pin**

0: No effect.

1: Clears the CAPTPIN interrupt.

19.5.20 CAPTURE Status Register

Name: CAPT_SR
Access: Read-only
Offset: 0x070

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

- **PDCEND: PDC End**

0: No effect.

1: The PDC has finished the data transfer.

- **OVERRUN: Overrun**

0: No effect.

1: An overrun has occurred.

Overrun indicates a valid data was not read when an overwrite occurred.

- **OVERFLOW: Overflow**

0: No effect.

1: An overflow has occurred.

Overflow indicates that the counter of the duration is saturated.

- **DATACAPT: Data Captured**

0: No effect.

1: Data in CAP_DR has to be read.

This bit is cleared by reading the CAP_DR register.

- **CAPENS: Capture Enable Status**

0: Capture is disabled.

1: Capture is enabled.

- **CAPTPIN: Capture Pin**

0: No effect.

1: The capture input pin has changed state.

19.5.21 CAPTURE Interrupt Enable Register

Name: CAPT_IER
Access: Write-only
Offset: 0x074

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

19.5.22 CAPTURE Interrupt Disable Register

Name: CAPT_IDR
Access: Write-only
Offset: 0x078

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

19.5.23 CAPTURE Interrupt Mask Register

Name: CAPT_IMR
Access: Read-only
Offset: 0x07C

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

- **PDCEND: PDC End Interrupt Mask**

0: PDCEND interrupt is disabled.

1: PDCEND interrupt is enabled.

- **OVERRUN: Overrun Interrupt Mask**

0: OVERRUN interrupt is disabled.

1: OVERRUN interrupt is enabled.

- **OVERFLOW: Overflow Interrupt Mask**

0: OVERFLOW interrupt is disabled.

1: OVERFLOW interrupt is enabled.

- **DATACAPT: Data Capture Interrupt Mask**

0: DATACAPT interrupt is disabled.

1: DATACAPT interrupt is enabled.

- **CAPTPIN: Capture Pin**

0: CAPTPIN interrupt is disabled.

1: CAPTPIN interrupt is enabled.

19.5.24 CAPTURE Data Register

Name: CAPT_DR
Access: Read/Write
Offset: 0x080

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

- **DURATION[14:0]: Capture Duration**

Number of CAPTCLK cycles during which the output is at the level indicated by the LEVEL bit, or during a frame period, depending on MEASMODE in CAP_MR.

- **LEVEL: Level measured**

0: The duration concerns a low level.

1: The duration concerns a high level.

Note that if the MEASMODE[1:0] (CAPT_MR) equals 1X, the LEVEL bit is used as a duration bit.

20. Simple Timer (ST)

The AT91SAM7A1 microcontroller includes two 16-bit Simple Timers (ST0 and ST1) with 2 channels per simple timer.

Each simple timer channel provides basic functions for timing calculation including two cascaded dividers and a 16 bit-counter.

The prescalar defines the clock frequency of the channel counter.

For each channel it is possible to select the divider clock between the core clock (CORECLK) and the low frequency clock (LFCLK).

The 16-bit counter starts down-counting when a value different from zero is loaded. An interrupt is generated when the counter reaches 0x0000.

When a value is loaded in the LOAD[15:0] bits of the STx_CTz register and the channel is started, the counter starts down-counting at channel clock frequency until the counter reaches zero. The delay between the load and the interrupt is:

Counter Value x Clock Period

The counter value is the value loaded in the counter. The clock period is the period of the channel clock (divided by the prescalar).

The precision is one clock period (from 0 to 1 channel clock period lost).

When enabling or disabling the Simple Timer, software must wait for enabled or disabled interrupt (or status) to be sure that the counter is really enabled or disabled (it is asynchronously clocked).

It is not possible to change contents of the Channel X Counter Register when the Simple Timer is enabled.

If a channel is disabled before it finishes down-counting, the counter value is held. If no write has occurred on the Counter Register before it is re-enabled, the down-counting restarts from the latest counter value.

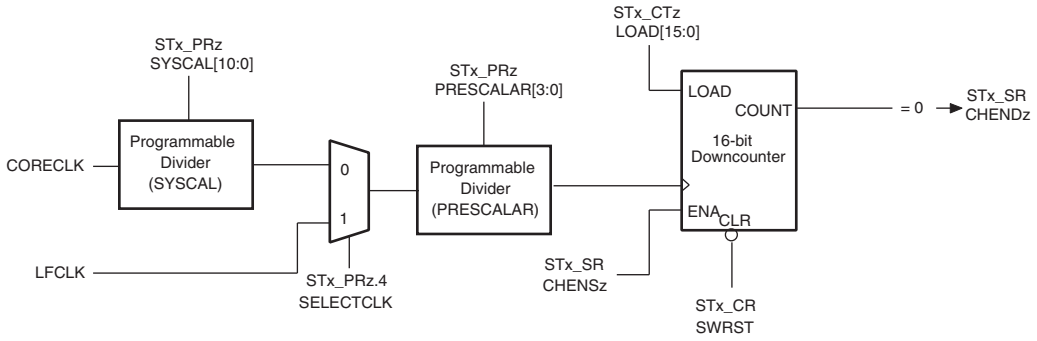
When the core clock (CORECLK) is selected on the Prescalar Register, the clock is divided twice to obtain the counter clock. First, it is divided by a divider driven by the SYSCAL bits of the Prescalar Register, and then by a divider driven by the prescalar bits of the Prescalar Register.

On the other hand, if the low frequency clock (LFCLK) is selected, the clock is divided only once to obtain the counter clock. In this case, it is divided by a divider driven by the prescalar bits of the Prescalar Register.

The Simple Timer also integrates an automatic reload function if the AUTOREL bit is set on the corresponding STx_PRz register. When the counter reaches 0x0000, it is automatically reloaded with the value written in LOAD[15:0] bits of the STx_CTz registers (x for ST0 or ST1 and z for channel 0 or channel 1).

The current value of the counter can be read in the corresponding ST_CCVx register.

Figure 20-1. Simple Timer Block Diagram



20.1 Interrupts

For each channel there are three types of interrupts:

- A CHDISz interrupt indicating that the channel has been disabled
- A CHLOADz interrupt indicating that the channel has started to down-count and load the data
- A CHENDz interrupt indicating that the channel has finished down-counting. This interrupt rises three CORECLK cycles after the down counter reaches the value 0x0000

where z is the channel number.

20.2 Power Management

Each Simple Timer (ST0 and ST1) is provided with a power management block allowing optimization of power consumption. See ”Power Management Block” on page 23.

20.3 Simple Timer (ST) Memory Map

Base Address ST0: 0xFFFE4000

Base Address ST1: 0xFFFE8000

Table 20-1. ST Memory Map

Offset	Register	Name	Access	Reset State
0x000 – 0x04C	Reserved	–	–	–
0x050	Enable Clock Register	ST_ECR	Write-only	–
0x054	Disable Clock Register	ST_DCR	Write-only	–
0x058	Power Management Status Register	ST_PMSR	Read-only	0x00000000
0x05C	Reserved	–	–	–
0x060	Control Register	ST_CR	Write-only	–
0x064	Reserved	–	–	–
0x068	Reserved	–	–	–
0x06C	Clear Status Register	ST_CSR	Write-only	–
0x070	Status Register	ST_SR	Read-only	0x00000000
0x074	Interrupt Enable Register	ST_IER	Write-only	–
0x078	Interrupt Disable Register	ST_IDR	Write-only	–
0x07C	Interrupt Mask Register	ST_IMR	Read-only	0x00000000
0x080	Channel 0 Prescalar	ST_PR0	Read/Write	0x00000000
0x084	Channel 0 Counter	ST_CT0	Read/Write	0x00000000
0x088	Channel 1 Prescalar	ST_PR1	Read/Write	0x00000000
0x08C	Channel 1 Counter	ST_CT1	Read/Write	0x00000000
0x200	Current Counter Value 0	ST_CCV0	Read	0x00000000
0x204	Current Counter Value 1	ST_CCV1	Read	0x00000000

20.3.1 ST Enable Clock Register

Name: ST_ECR
Access: Write-only
Offset: 0x050

20.3.2 ST Disable Clock Register

Name: ST_DCR
Access: Write-only
Offset: 0x054

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
–	–	–	–	–	–	ST	–

- **ST: Simple Timer Clock Status**

0: Simple Timer clock disabled.

1: Simple Timer clock enabled.

20.3.3 ST Power Management Status Register

Name: ST_PMSR
Access: Read-only
Offset: 0x058

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
–	–	–	–	–	–	ST	–

- **ST: Simple Timer Clock Status**

0: Simple Timer clock disabled.

1: Simple Timer clock enabled.

Note: The ST_PMSR register is not reset by a software reset.

20.3.4 ST Control Register

Name: ST_CR

Access: Write-only

Offset: 0x060

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
–	–	–	CHDIS1	CHEN1	CHDIS0	CHEN0	SWRST

- **SWRST: ST Software Reset**

0: No effect.

1: Resets the Simple Timer.

A software reset of the ST is performed. It resets all the registers.

- **CHENX: Simple Timer Channel Enable**

0: No effect.

1: Enables the Simple Timer Channel X.

- **CHDISX: Simple Timer Channel Disable**

0: No effect.

1: Disables the Simple Timer Channel X.

If both CHENX and CHDISX are equal to one when the control register is written, the Simple Timer Channel X is disabled.

20.3.5 ST Clear Status Register

Name: ST_CSR
Access: Write-only
Offset: 0x06C

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
–	–	CHLD1	CHDIS1	CHEND1	CHLD0	CHDIS0	CHEND0

- **CHENDx: Clear Channel x End Interrupt**

0: No effect.

1: Clears CHENDx interrupt.

- **CHDISx: Clear Channel x Disable Interrupt**

0: No effect.

1: Clears CHDISx interrupt.

- **CHLDX: Clear Channel x Load Interrupt**

0: No effect.

1: Clears CHLDx interrupt.

20.3.6 ST Status Register

Name: ST_SR
Access: Read-only
Offset: 0x070

31	30	29	28	27	26	25	24
–	–	–	–	–	–	CHENS1	CHENS0
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	CHLD1	CHDIS1	CHEND1	CHLD0	CHDIS0	CHEND0

- **CHENDx: Channel x End Status**

0: No end of counting on Channel x.

1: End of counting on Channel x.

- **CHENSx: Channel x Enable Status**

0: Channel x is disabled.

1: Channel x is enabled.

- **CHDISx: Channel x Disable Status**

0: No effect.

1: Channel x divider has been reset.

CHDISx indicates that the divider module has been reset (a delay due to asynchronous clock is introduced).

- **CHLDx: Channel x Load Status**

0: No effect.

1: Channel x down-counter is loaded.

Note: CHLDx also indicates that the counter (divider) has been enabled (a delay due to asynchronous clock is introduced).

20.3.7 ST Interrupt Enable Register

Name: ST_IER
Access: Write-only
Offset: 0x074

20.3.8 ST Interrupt Disable Register

Name: ST_IDR
Access: Write-only
Offset: 0x078

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
–	–	CHLD1	CHDIS1	CHEND1	CHLD0	CHDIS0	CHEND0

20.3.9 ST Interrupt Mask Register

Name: ST_IMR
Access: Read-only
Offset: 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
–	–	CHLD1	CHDIS1	CHEND1	CHLD0	CHDIS0	CHEND0

- **CHENDx: Channel x End Interrupt Mask**

0: Channel x end interrupt is disabled.

1: Channel x end interrupt is enabled.

- **CHDISx: Channel x Disable Interrupt Mask**

0: Channel x disable interrupt is disabled.

1: Channel x disable interrupt is enabled.

- **CHLDx: Channel x Load Interrupt Mask**

0: Channel x load interrupt is disabled.

1: Channel x load interrupt is enabled.

20.3.10 ST Channel 0 Prescalar Register

Name: ST_PR0
Access: Read/Write
Offset: 0x080

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

- **PRESCALAR: Channel 0 Prescalar**

$$\text{Channel 0 counter\_clock\_frequency} = \text{divider\_clock\_1} / 2^{PRESCALAR}$$

- **SELECTCLK: Select Clock**

0: The divider_clock_0 is generated by divider driven by the SYSCAL bits.

1: The divider_clock_0 is connected to the low frequency clock.

- **AUTOREL: Auto Reload**

0: The counter is not automatically reloaded with the COUNTER[15:0] value when it reaches 0x0000.

1: The counter is automatically reloaded with the COUNTER[15:0] value when it reaches 0x0000.

- **SYSCAL: System Clock Prescalar Value**

This prescalar is used to divide the CORECLK when the system clock is selected (SELECTCLK = 0).

$$\text{divider\_clock\_0} = \text{CORECLK} / (2 \times (\text{SYSCAL} + 1))$$

A write in this register can only be done if CHENS0 = 0 (i.e., channel 0 is disabled).

20.3.11 ST Channel 0 Counter Register

Name: ST_CT0
Access: Read/Write
Offset: 0x084

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

- **LOAD: Channel 0 Preload Value**

Gives the instruction to the timer; i.e., from which counter value the Simple Timer has to decrement.

20.3.12 ST Channel 1 Prescalar Register

Name: ST_PR1
Access: Read/Write
Offset: 0x088

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

- **PRESCALAR: Channel 1 Prescalar**

$$\text{Channel 1 counter\_clock\_frequency} = \text{divider\_clock\_2} / 2^{PRESCALAR}$$

- **SELECTCLK: Select Clock**

0: The divider_clock_1 is generated by the divider driven by the SYSCAL bits.

1: The divider_clock_1 is connected to the low frequency clock.

- **AUTOREL: Auto Reload**

0: The counter is not automatically reloaded with the COUNTER[15:0] value when it reaches 0x0000.

1: The counter is automatically reloaded with the COUNTER[15:0] value when it reaches 0x0000.

- **SYSCAL: System Clock Prescalar Value**

This prescalar is used to divide the CORECLK when the system clock is selected (SELECTCLK = 0).

$$\text{divider\_clock\_1} = \text{CORECLK} / (2 \times (\text{SYSCAL} + 1))$$

A write in this register can only be done if CHENS1 = 0 (i.e., channel 1 is disabled).

20.3.13 ST Channel 1 Counter Register

Name: ST_CT1
Access: Read/Write
Offset: 0x08C

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

- **LOAD: Channel 1 Preload Value**

Gives the instruction to the timer; i.e., from which counter value the Simple Timer has to decrement.

20.3.14 ST Current Counter Value 0 Register

Name: ST_CCV0
Access: Read-only
Offset: 0x0200

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

- **COUNT[15:0]: Current Counter Value 0 Register**

This register gives the current value of the Channel 0 down counter.

20.3.15 ST Current Counter Value 1 Register

Name: ST_CCV1
Access: Read-only
Offset: 0x0204

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

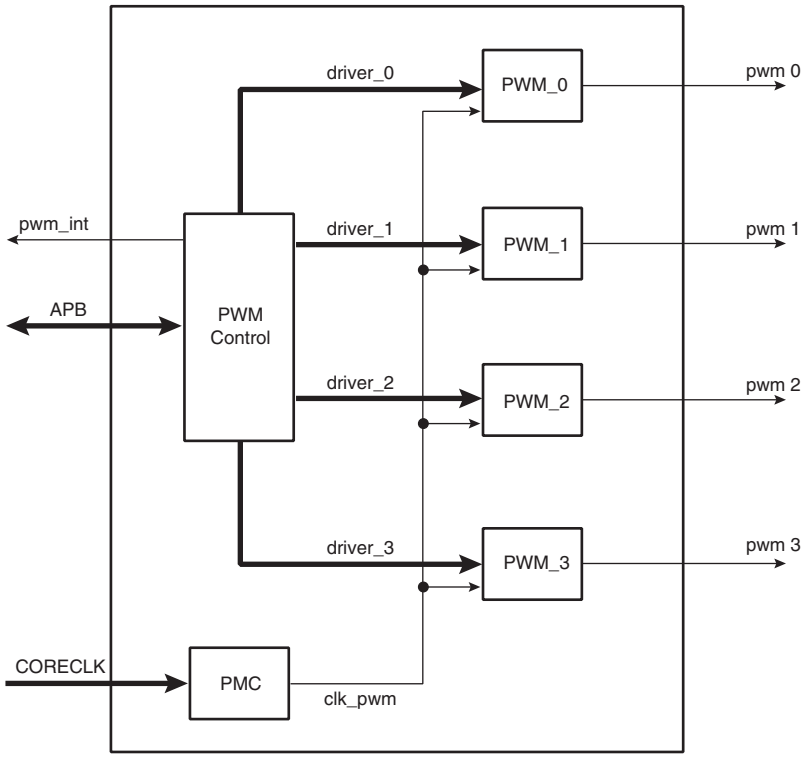
- **COUNT[15:0]: Current Counter Value 1 Register**

This register gives the current value of the Channel 1 down counter.

21. Pulse Width Modulation (PWM)

The AT91SAM7A1 includes a four-channel PWM that generates pulses. The width and the frequency of the pulses can be controlled independently on each channel.

Figure 21-1. PWM Block Diagram



21.1 Pin Description

There are four output pins: PWM0, PWM1, PWM2 and PWM3. The pins are multiplexed with a PIO block so they can be used as general-purpose I/O pins.

After a hardware reset, the pins are configured as PIO (inputs).

21.2 PWM Parameters

There are three parameters for each PWM: counter frequency, delay and pulse width.

21.2.1 Counter Frequency

The PWM module is a counter. It counts delay width cycles, setting the PWMX output inactive, then it counts pulse width cycles, setting the PWMX output active. This operation is repeated until the PWM channel is stopped.

The user can control the frequency of this counter with a prescalar.

21.2.2 Delay Width

Delay width is the number of counter cycles during which the output is inactive.

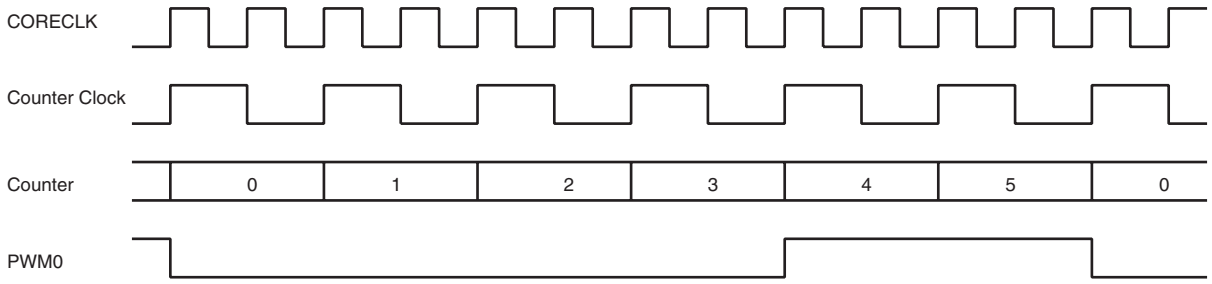
21.2.3 Pulse Width

Pulse width is the number of counter cycles during which the output is active.

21.3 Example of Use

If the counter frequency is set to CORECLK/2, delay width is equal to four cycles and pulse width to two cycles. See Figure 21-2.

Figure 21-2. Waveform Diagram



21.4 PIO Controller

The PWM has four programmable I/O lines. These I/O lines are multiplexed with signals (PWM0, PWM1, PWM2, PWM3) of the PWM to optimize the use of available package pins. These lines are controlled by the PWM PIO controller.

21.5 Power Management

The PWM is provided with a power management block allowing optimization of power consumption (see ”Power Management Block” on page 23).

21.6 Pulse Width Modulator (PWM) Memory Map

Table 21-1. PWM Memory Map

Address	Register	Name	Access	Reset State
0xFFFD0000	PIO Enable Register	PWM_PER	Write-only	---
0xFFFD0004	PIO Disable Register	PWM_PDR	Write-only	---
0xFFFD0008	PIO Status Register	PWM_PSR	Read-only	0x000F0000
0xFFFD000C	Reserved	---	---	---
0xFFFD0010	Output Enable Register	PWM_OER	Write-only	---
0xFFFD0014	Output Disable Register	PWM_ODR	Write-only	---
0xFFFD0018	Output Status Register	PWM_OSR	Read-only	0x00000000
0xFFFD001C	Reserved	---	---	---
0xFFFD0020	Reserved	---	---	---
0xFFFD0024	Reserved	---	---	---
0xFFFD0028	Reserved	---	---	---
0xFFFD002C	Reserved	---	---	---
0xFFFD0030	Set Output Data Register	PWM_SODR	Write-only	---
0xFFFD0034	Clear Output Data Register	PWM_CODR	Write-only	---
0xFFFD0038	Output Data Status Register	PWM_ODSR	Read-only	0x00000000
0xFFFD003C	Pin Data Status Register	PWM_PDSR	Read-only	0x000X0000
0xFFFD0040	Multi-Drive Enable Register	PWM_MDER	Write-only	---
0xFFFD0044	Multi-Drive Disable Register	PWM_MDDR	Write-only	---
0xFFFD0048	Multi-Drive Status Register	PWM_MDSR	Read-only	0x00000000
0x000 - 0x04C	Reserved	---	---	---
0xFFFD0050	Enable Clock Register	PWM_ECR	Write-only	–
0xFFFD0054	Disable Clock Register	PWM_DCR	Write-only	–
0xFFFD0058	Power Management Status Register	PWM_PMSR	Read-only	0x00000000
0xFFFD005C	Reserved	---	---	---
0xFFFD0060	Control Register	PWM_CR	Write-only	–
0xFFFD0064	Mode Register	PWM_MR	Read/Write	0x10101010
0xFFFD0068	Reserved	---	---	---
0xFFFD006C	Clear Status Register	PWM_CSR	Write-only	---
0xFFFD0070	Status Register	PWM_SR	Read-only	0x00000000
0xFFFD0074	Interrupt Enable Register	PWM_IER	Write-only	---
0xFFFD0078	Interrupt Disable Register	PWM_IDR	Write-only	---
0xFFFD007C	Interrupt Mask Register	PWM_IMR	Read-only	0x00000000
0xFFFD0080	Delay Register 0	PWM_DLY_0	Read/Write	0x00000000
0xFFFD0084	Pulse Register 0	PWM_PUL_0	Read/Write	0x00000000
0xFFFD0088	Delay Register 1	PWM_DLY_1	Read/Write	0x00000000

Table 21-1. PWM Memory Map (Continued)

Address	Register	Name	Access	Reset State
0xFFFD008C	Pulse Register 1	PWM_PUL_1	Read/Write	0x00000000
0xFFFD0090	Delay Register 2	PWM_DLY_2	Read/Write	0x00000000
0xFFFD0094	Pulse Register 2	PWM_PUL_2	Read/Write	0x00000000
0xFFFD0098	Delay Register 3	PWM_DLY_3	Read/Write	0x00000000
0xFFFD009C	Pulse Register 3	PWM_PUL_3	Read/Write	0x00000000

21.6.1 PWM PIO Enable Register

Name: PWM_PER
Access: Write-only
Address 0xFFFD0000

21.6.2 PWM PIO Disable Register

Name: PWM_PDR
Access: Write-only
Address: 0xFFFD0004

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	PWM3	PWM2	PWM1	PWM0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **PWM[3:0]: PWM Pin**

0: The PIO control of the corresponding PWM pin is disabled (the pin works in the PWM mode).

1: The PIO control of the corresponding PWM pin is enabled (the pin works as PIO).

21.6.3 PWM PIO Status Register

Name: PWM_PSR
Access: Read-only
Address: 0xFFFD0008

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	PWM3	PWM2	PWM1	PWM0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **PWM[3:0]: PWM Pin**

0: The PIO control of the corresponding PWM pin is disabled (the pin works in the PWM mode).

1: The PIO control of the corresponding PWM pin is enabled (the pin works as PIO).

21.6.4 PWM PIO Output Enable Register

Name: PWM_OER
Access: Write-only
Address: 0xFFFD0010

21.6.5 PWM PIO Output Disable Register

Name: PWM_ODR
Access: Write-only
Address: 0xFFFD0014

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	PWM3	PWM2	PWM1	PWM0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **PWM[3:0]: PWM Pin**

0: The PIO output is disabled on the corresponding PWM pin.

1: The PIO output is enabled on the corresponding PWM pin.

21.6.6 PWM PIO Output Status Register

Name: PWM_OSR
Access: Read-only
Address: 0xFFFD0018

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	PWM3	PWM2	PWM1	PWM0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **PWM[3:0]: PWM Pin**

0: The PIO output is disabled on the corresponding PWM pin.

1: The PIO output is enabled on the corresponding PWM pin.

21.6.7 PWM PIO Set Output Data Register

Name: PWM_SODR
Access: Write-only
Address: 0xFFFD0030

21.6.8 PWM PIO Clear Output Data Register

Name: PWM_CODR
Access: Write-only
Address: 0xFFFD0034

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	PWM3	PWM2	PWM1	PWM0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **PWM[3:0]: PWM Pin**

0: The PIO output is set to logical 0 on the corresponding PWM pin.

1: The PIO output is set to logical 1 on the corresponding PWM pin.

21.6.9 PWM PIO Output Data Status Register

Name: PWM_ODSR
Access: Read-only
Address: 0xFFFD0038

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	PWM3	PWM2	PWM1	PWM0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **PWM[3:0]: PWM Pin**

0: The PIO output is set to logical 0 on the corresponding PWM pin.

1: The PIO output is set to logical 1 on the corresponding PWM pin.

21.6.10 PWM PIO Pin Data Status Register

Name: PWM_PDSR
Access: Read-only
Address: 0xFFFD003C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	PWM3	PWM2	PWM1	PWM0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **PWM[3:0]: PWM Pin**

0: The corresponding PWM pin is at logical 0.

1: The corresponding PWM pin is at logical 1.

21.6.11 PWM PIO Multi-drive Enable Register

Name: PWM_MDER
Access: Write-only
Address: 0xFFFD0040

21.6.12 PWM PIO Multi-drive Disable Register

Name: PWM_MDDR
Access: Write-only
Address: 0xFFFD0044

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	PWM3	PWM2	PWM1	PWM0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **PWM[3:0]: PWM Pin**

0: The corresponding PWM pin is not configured as an open drain.

1: The corresponding PWM pin is configured as an open drain.

21.6.13 PWM PIO Multi-drive Status Register

Name: PWM_MDSR
Access: Read-only
Address: 0xFFFD0048

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	PWM3	PWM2	PWM1	PWM0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **PWM[3:0]: PWM Pin**

0: The corresponding PWM pin is not configured as an open drain.

1: The corresponding PWM pin is configured as an open drain.

21.6.14 PWM Enable Clock Register

Name: PWM_ECR
Access: Write-only
Address: 0xFFFD0050

21.6.15 PWM Disable Clock Register

Name: PWM_DCR
Access: Write-only
Address: 0xFFFD0054

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
–	–	–	–	–	–	–	–

- **PWM: PWM Clock**

0: PWM clock disabled.

1: PWM clock enabled.

Note: The PWM_PMSR register is not reset by software reset.

21.6.16 PWM Power Management Status Register

Name: PWM_PMSR
Access: Read-only
Address: 0xFFFD0058

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
–	–	–	–	–	–	–	–

- **PWM: PWM Clock**

0: PWM clock disabled.

1: PWM clock enabled.

Note: The PWM_PMSR register is not reset by software reset.

21.6.17 PWM Control Register

Name: PWM_CR
Access: Write-only
Address: 0FFFD0x060

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	PWMDIS3
7	6	5	4	3	2	1	0
PWMEN3	PWMDIS2	PWMEN2	PWMDIS1	PWMEN1	PWMDIS0	PWMEN0	SWRST

- **SWRST: PWM Software Reset**

0: No effect

1: Resets the PWM.

A software-triggered hardware reset of the PWM is performed. It resets all the registers except the PWM_PMSR register.

- **PWMENX: PWM Enable Channel Number X**

0: No effect.

1: Enables the PWM.

- **PWMDISX: PWM Disable Channel Number X**

0: No effect.

1: Disables the PWM.

If both PWMENX and PWMDISX are equal to one, the corresponding PWM channel is disabled.

21.6.18 PWM Mode Register

Name: PWM_MR
Access: Read/Write
Address: 0xFFFD0064

31	30	29	28	27	26	25	24
–	–	–	PL3	PRESCAL3			
23	22	21	20	19	18	17	16
–	–	–	PL2	PRESCAL2			
15	14	13	12	11	10	9	8
–	–	–	PL1	PRESCAL1			
7	6	5	4	3	2	1	0
			PL0	PRESCAL0			

- **PRESCALX[3:0]: Counter Clock Prescalar for PWM Channel X**

$$\text{Counter Clock Frequency} = \text{CORECLK} / 2^{PRESCALAR1}$$

Note: "X" indicates that the bit concerns the PWMX.

- **PLX: Pulse Level for PWM Channel X**

0: The pulses are at logic level 0. During the delay, the output is at logic level 1.

1: The pulses are at logic level 1. During the delay, the output is at logic level 0.

Note: When pulse level changes, a pulse start or a pulse end is detected, and the corresponding bit is set in the status register (generating an interrupt if enabled).

21.6.19 PWM Clear Status Register

Name: PWM_CSR
Access: Write-only
Address: 0xFFFD006C

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
PEND3	PSTA3	PEND2	PSTA2	PEND1	PSTA1	PEND0	PSTA0

- **PSTAX: Pulse Start Interrupt**

0: No effect.

1: Clears the PSTA interrupt.

- **PENDX: Pulse End Interrupt**

0: No effect.

1: Clears the PEND interrupt.

Note: "X" indicates that the bit concerns the PWMX.

21.6.20 PWM Status Register

Name: PWM_SR

Access: Read-only

Address: 0xFFFD0070

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	PWMENS3	PWMENS2	PWMENS1	PWMENS0
7	6	5	4	3	2	1	0
PEND3	PSTA3	PEND2	PSTA2	PEND1	PSTA1	PEND0	PSTA0

- **PSTAX: Pulse Start**

0: No pulse started since the last read of PWM_SR.

1: A pulse started since the last read of PWM_SR.

- **PENDX: Pulse End**

0: No pulse ended since the last read of PWM_SR.

1: A pulse ended since the last read of PWM_SR.

- **PWMENSX: PWM Enable Status of Channel X**

0: PWM is disabled.

1: PWM is enabled.

Note: “X” indicates that the bit concerns the PWMX.

21.6.21 PWM Interrupt Enable Register

Name: PWM_IER
Access: Write-only
Address: 0xFFFD0074

21.6.22 PWM Interrupt Disable Register

Name: PWM_IDR
Access: Write-only
Address: 0xFFFD0078

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
PEND3	PSTA3	PEND2	PSTA2	PEND1	PSTA1	PEND0	PSTA0

- **PSTAX: Pulse Start Interrupt**

0: Pulse Start Interrupt is disabled.

1: Pulse Start Interrupt is enabled.

- **PENDX: Pulse End Interrupt**

0: Pulse End Interrupt is disabled.

1: Pulse End Interrupt is enabled.

Note: “X” indicates that the bit concerns the PWMX.

21.6.23 PWM Interrupt Mask Register

Name: PWM_IMR
Access: Read-only
Address: 0xFFFD007C

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
PEND3	PSTA3	PEND2	PSTA2	PEND1	PSTA1	PEND0	PSTA0

- **PSTAX: Pulse Start Interrupt**

0: Pulse Start Interrupt is disabled.

1: Pulse Start Interrupt is enabled.

- **PENDX: Pulse End Interrupt**

0: Pulse End Interrupt is disabled.

1: Pulse End Interrupt is enabled.

Note: "X" indicates that the bit concerns the PWMX.

21.6.24 PWM Delay Register x [x = 0..3]

Name: PWM_DLY_x
Access: Read/Write
Address: 0xFFFD0080...0xFFFD0098

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

- **DELAY[15:0]: PWM Delay on Channel X**

Number of counter cycles during which the output is inactive.

Note: “X” indicates that the bit concerns the PWMX.

Change of this value is immediately taken into account. This may cause a bad delay on the current pulse.

21.7 PWM Pulse Register x [x = 0..3]

Name: PWM_PUL_x
Access: Read/Write
Address: 0xFFFD0084...0xFFFD009C

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

- **PULSE[15:0]: Pulse Width on Channel X**

Number of counter cycles during which the output is active.

Note: “X” indicates that the bit concerns the PWMX.

Change of this value is immediately taken into account. This may cause a bad delay on the current pulse.

22. Serial Peripheral Interface (SPI)

22.1 Description

The AT91SAM7A1 includes a Serial Peripheral Interface (SPI) that provides communication with external devices in Master or Slave Mode. It also allows communication between processors if an external processor is connected to the system.

Seven pins are associated with the SPI interface. When not needed for the SPI function, each of these pins can be configured as a PIO, using PIO Controller functions.

After a hardware reset, the SPI pins are not enabled by default. The user must configure the PIO Controller to enable the corresponding pins to their PIO function. NPCS0 Parallel I/O Controller for multi-master operation: support for an external master is provided by the PIO Controller.

To configure an SPI pin as open-drain to support external drivers, the software can set the corresponding bits in the SPI_MDSR registers. The NPCS0 pin can function as a peripheral chip select output or slave select input.

22.2 Master Mode

In Master Mode, the SPI controls data transfers to and from the slave(s) connected to the SPI bus. The SPI drives the chip select(s) to the slave(s) and the serial clock (SCK). After enabling the SPI, a data transfer begins when the ARM core writes to the Transmit Data Register (SPI_TDR).

Transmit and Receive buffers maintain the data flow at a constant rate with a reduced requirement for high-priority interrupt servicing. As long as new data is available in the Transmit Data Register (SPI_TDR), the SPI continues to transfer data. If the Receive Data Register (SPI_RDR) has not been read before new data is received, the overrun error (SPIOVRE) flag is set.

The delay between the activation of the chip select and the start of the data transfer (DLYBS) as well as the delay between each data transfer (DLYBCT) can be programmed for each of the four external chip selects. All data transfer characteristics including the two timing values are programmed in registers SPI_CSR0 to SPI_CSR3 (Chip Select Registers).

In Master Mode, the peripheral selection can be defined in two different ways:

- Fixed peripheral select: SPI exchanges data with only one peripheral
- Variable peripheral select: Data can be exchanged with more than one peripheral

22.2.1 Fixed Peripheral Select

This mode is used for transferring memory blocks without the extra overhead in the Transmit Data Register to determine the peripheral.

Fixed peripheral select is activated by setting bit PS to a logical 0 in the SPI Mode Register (SPI_MR).

The peripheral is defined by the PCS field in SPI_MR.

This option is only available when the SPI is programmed in Master Mode.

22.2.2 Variable Peripheral Select

Variable peripheral select is activated by setting bit PS to a logical 1 in the SPI Mode Register (SPI_MR). The PCS field in Transmit Data Register (SPI_TDR) is used to select the destination peripheral. The data transfer characteristics are changed when the selected peripheral changes, according to the associated chip select register.

The PCS field in the SPI_MR has no effect.

This option is only available when the SPI is programmed in Master Mode.

22.2.3 Chip Selects

The Chip Select lines are driven by the SPI only if it is programmed in Master Mode. These lines are used to select the destination peripheral.

The PCSDEC field in SPI_MR (Mode Register) is used to select one to four peripherals or up to 16 peripherals:

- PCSDEC = 0, each NPCSx acts as a chip select line so up to four devices can be selected.
- PCSDEC = 1, the NPCS[3:0] signals act as an address bus selecting up to 16 peripherals.

If variable peripheral select is active (PS = 1 in SPI_MR), the chip select signals are defined for each transfer in the PCS field in SPI_TDR. Chip select signals can thus be defined independently for each transfer.

If fixed peripheral select is active (PS = 0 in SPI_MR), chip select signals are defined for all transfers by the field PCS in SPI_MR. If a transfer with a new peripheral is necessary, the software must wait until the current transfer is completed, then change the value of PCS in SPI_MR before writing new data in SPI_TDR.

The value on the NPCS pins at the end of each transfer can be read in the Receive Data Register (SPI_RDR). By default, all NPCS signals are high (equal to one) before and after each transfer.

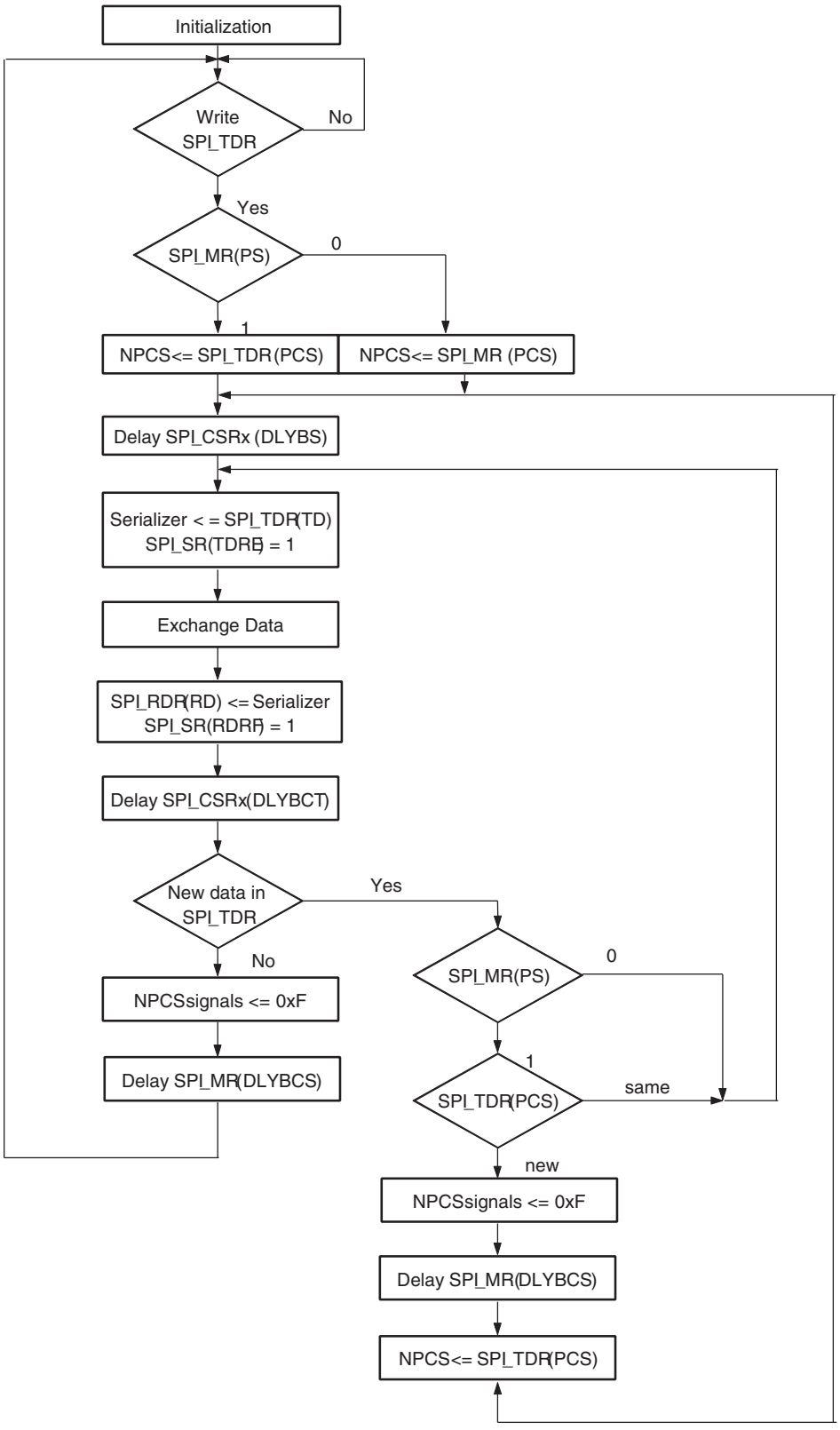
22.2.4 Mode Fault Detection

A mode fault is detected when the SPI is programmed in Master Mode and a low level is driven by an external master on the NPCS0 signal.

When a mode fault is detected, the MODF bit in the SPI_SR is set until the SPI_SR is read and the SPI is disabled until re-enabled by bit SPIEN in the SPI_CR (Control Register).

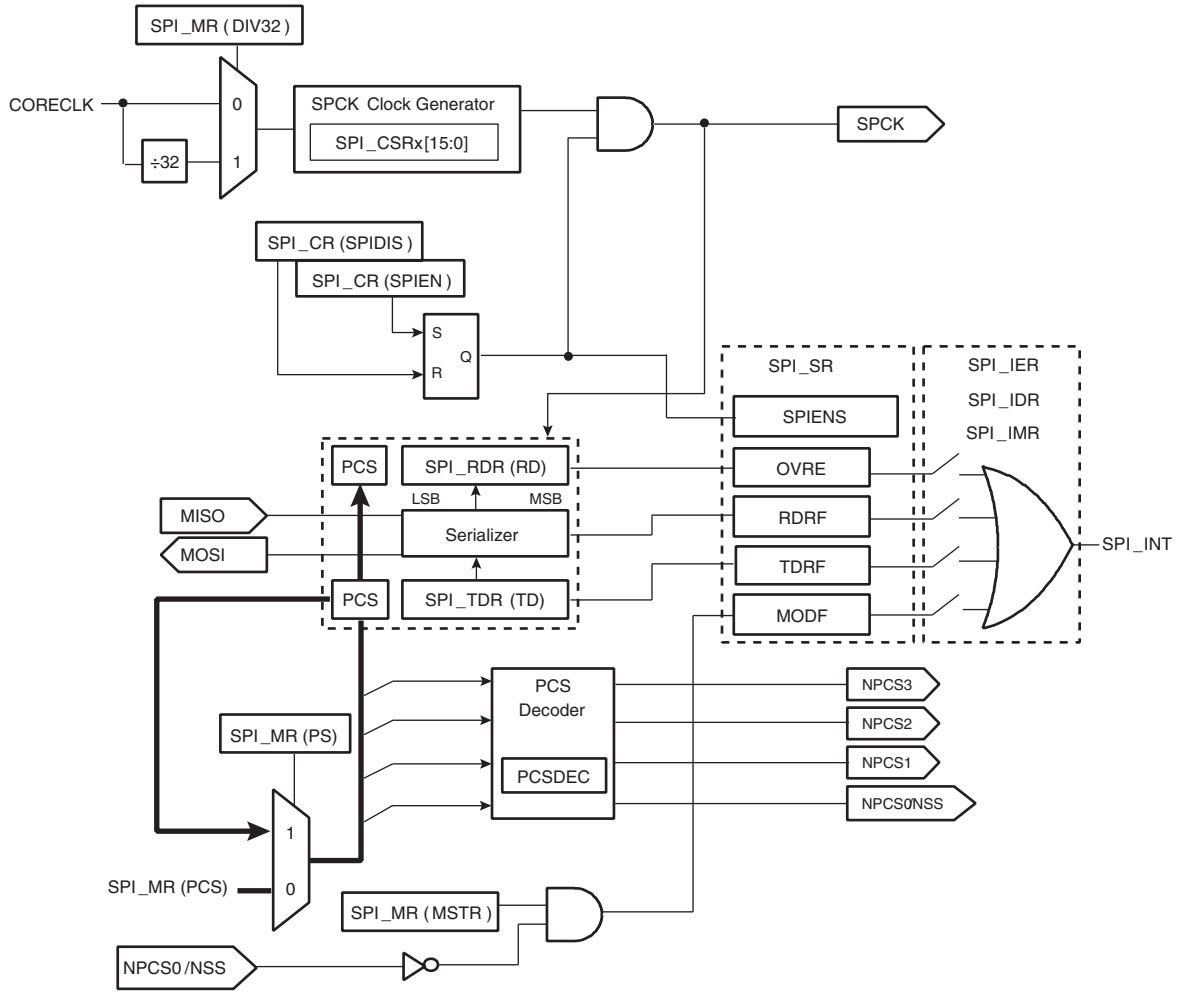
22.2.5 Functional Flow Diagram in Master Mode

Figure 22-1. SPI Flow Diagram in Master Mode



22.2.6 SPI in Master Mode

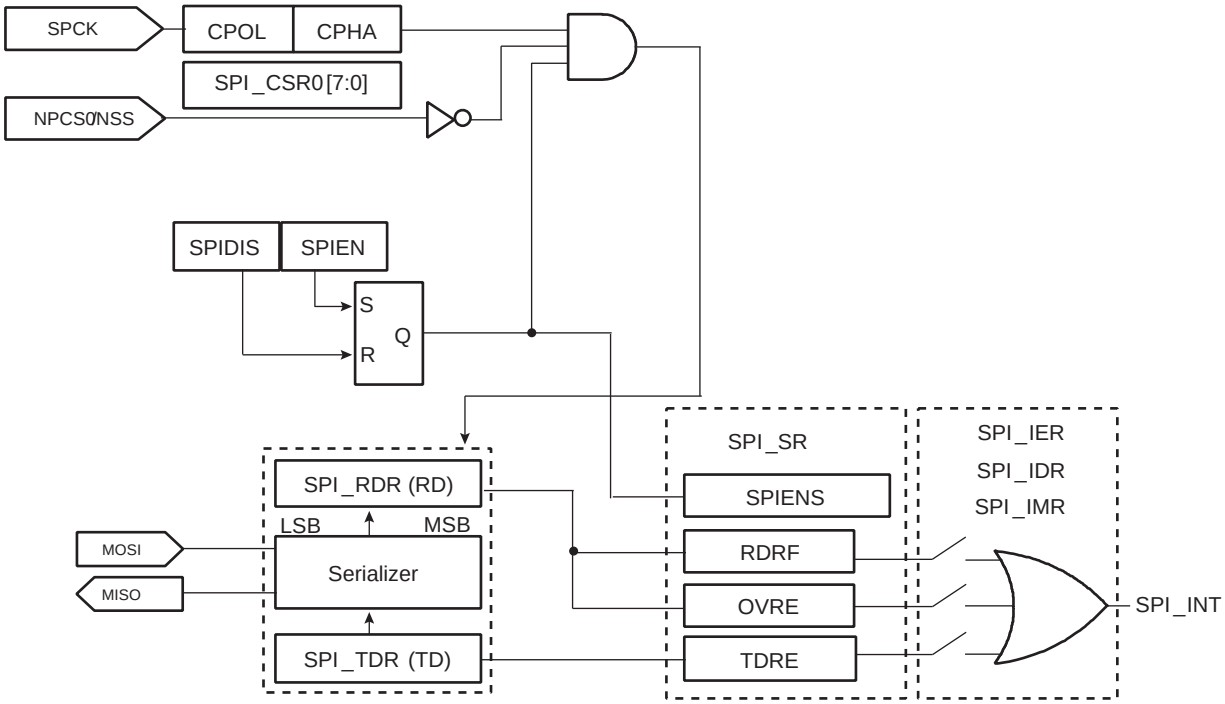
Figure 22-2. SPI in Master Mode



22.3 Slave Mode

In Slave Mode, the SPI waits for NSS to go active low before receiving the serial clock from an external master. In Slave Mode, CPOL, NCPHA and BITS fields of SPI_CSR0 are used to define the transfer characteristics. The other fields in SPI_CSR0 and the other Chip Select Registers are not used in Slave Mode.

Figure 22-3. SPI in Slave Mode



22.4 PIO Controller

The SPI has 7 programmable I/O lines. These I/O lines are multiplexed with signals (MISO, MOSI, SPCK, NPCS[3:0]) of the SPI to optimize the use of available package pins. These lines are controlled by the SPI PIO controller.

22.5 Power Management

The SPI is provided with a power management block allowing optimization of power consumption (see ”Power Management Block” on page 23).

22.6 Serial Peripheral Interface (SPI) Memory Map

Table 22-1. SPI Memory Map

Address	Register	Name	Access	Reset State
0xFFFB4000	PIO Enable Register	SPI_PER	Write-only	–
0xFFFB4004	PIO Disable Register	SPI_PDR	Write-only	–
0xFFFB4008	PIO Status Register	SPI_PSR	Read-only	0x007F0000
0xFFFB400C	Reserved	–	–	–
0xFFFB4010	Output Enable Register	SPI_OER	Write-only	–
0xFFFB4014	Output Disable Register	SPI_ODR	Write-only	–
0xFFFB4018	Output Status Register	SPI_OSR	Read-only	0x00000000
0xFFFB401C – 0xFFFB402C	Reserved	–	–	–
0xFFFB4030	Set Output Data Register	SPI_SODR	Write-only	–
0xFFFB4034	Clear Output Data Register	SPI_CODR	Write-only	–
0xFFFB4038	Output Data Status Register	SPI_ODSR	Read-only	0x00000000
0xFFFB403C	Pin Data Status Register	SPI_PDSR	Read-only	0x00XX0000
0xFFFB4040	Multidriver Enable Register	SPI_MDER	Write-only	–
0xFFFB4044	Multidriver Disable Register	SPI_MDDR	Write-only	–
0xFFFB4048	Multidriver Status Register	SPI_MDSR	Read-only	0x00000000
0xFFFB404C	Reserved	–	–	–
0xFFFB4050	Enable Clock Register	SPI_ECR	Write-only	–
0xFFFB4054	Disable Clock Register	SPI_DCR	Write-only	–
0xFFFB4058	Power Management Status Register	SPI_PMSR	Read-only	0x00000000
0xFFFB405C	Reserved	–	–	–
0xFFFB4060	Control Register	SPI_CR	Write-only	0x00000000
0xFFFB4064	Mode Register	SPI_MR	Read/Write	0x00000000
0xFFFB4068 – 0xFFFB406C	Reserved	–	–	–
0xFFFB4070	Status Register	SPI_SR	Read-only	0x00000000
0xFFFB4074	Interrupt Enable Register	SPI_IER	Write-only	–
0xFFFB4078	Interrupt Disable Register	SPI_IDR	Write-only	–
0xFFFB407C	Interrupt Mask Register	SPI_IMR	Read-only	0x00000000
0xFFFB4080	Receive Data Register	SPI_RDR	Read-only	–
0xFFFB4084	Transmit Data Register	SPI_TDR	Write-only	0x00000000
0xFFFB4088 – 0xFFFB408C	Reserved	–	–	–

Table 22-1. SPI Memory Map (Continued)

Address	Register	Name	Access	Reset State
0xFFFB4090	Chip Select Register 0	SPI_CSR0	Read/Write	0x00000000
0xFFFB4094	Chip Select Register 1	SPI_CSR1	Read/Write	0x00000000
0xFFFB4098	Chip Select Register 2	SPI_CSR2	Read/Write	0x00000000
0xFFFB409C	Chip Select Register 3	SPI_CSR3	Read/Write	0x00000000

22.6.1 SPI PIO Enable Register

Name: SPI_PER
Access: Write-only
Address: 0xFFFB4000

22.6.2 SPI PIO Disable Register

Name: SPI_PDR
Access: Write-only
Address: 0xFFFB4004

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	NPCS3	NPCS2	NPCS1	NPCS0	MOSI	MISO	SPCK
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SPCK: SPCK Pin**

0: PIO is inactive on the pin SCK.

1: PIO is active on the pin SCK.

- **MISO: MISO Pin**

0: PIO is inactive on the pin MISO.

1: PIO is active on the pin MISO.

- **MOSI: MOSI Pin**

0: PIO is inactive on the pin MOSI.

1: PIO is active on the pin MOSI.

- **NPCS0: NPCS0 Pin**

0: PIO is inactive on the pin NPCS0/NSS.

1: PIO is active on the pin NPCS0/NSS.

- **NPCSx[x = 1..3]: NPCSx Pin**

0: PIO is inactive on the pin NPCSx.

1: PIO is active on the pin NPCSx.

22.6.3 SPI PIO Status Register

Name: SPI_PSR
Access: Read-only
Address: 0xFFFB4008

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	NPCS3	NPCS2	NPCS1	NPCS0	MOSI	MISO	SPCK
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SPCK: SPCK Pin**

0: PIO is inactive on the pin SCK.

1: PIO is active on the pin SCK.

- **MISO: MISO Pin**

0: PIO is inactive on the pin MISO.

1: PIO is active on the pin MISO.

- **MOSI: MOSI Pin**

0: PIO is inactive on the pin MOSI.

1: PIO is active on the pin MOSI.

- **NPCS0: NPCS0 Pin**

0: PIO is inactive on the pin NPCS0/NSS.

1: PIO is active on the pin NPCS0/NSS.

- **NPCSx[x = 1..3]: NPCSx Pin**

0: PIO is inactive on the pin NPCSx.

1: PIO is active on the pin NPCSx.

22.6.4 SPI PIO Output Enable Register

Name: SPI_OER
Access: Write-only
Address: 0xFFFB4010

22.6.5 SPI PIO Output Disable Register

Name: SPI_ODR
Access: Write-only
Address: 0xFFFB4014

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	NPCS3	NPCS2	NPCS1	NPCS0	MOSI	MISO	SPCK
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SPCK: SPCK Pin**

0: PIO is input on the pin SCK.

1: PIO is output on the pin SCK.

- **MISO: MISO Pin**

0: PIO is input on the pin MISO.

1: PIO is output on the pin MISO.

- **MOSI: MOSI Pin**

0: PIO is input on the pin MOSI.

1: PIO is output on the pin MOSI.

- **NPCS0: NPCS0 Pin**

0: PIO is input on the pin NPCS0/NSS.

1: PIO is output on the pin NPCS0/NSS.

- **NPCSx[x = 1..3]: NPCSx Pin**

0: PIO is input on the pin NPCSx.

1: PIO is output on the pin NPCSx.

22.6.6 SPI PIO Output Status Register

Name: SPI_OSR
Access: Read-only
Address: 0x0FFFB418

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	NPCS3	NPCS2	NPCS1	NPCS0	MOSI	MISO	SPCK
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SPCK: SPCK Pin**

0: PIO is input on the pin SCK.

1: PIO is output on the pin SCK.

- **MISO: MISO Pin**

0: PIO is input on the pin MISO.

1: PIO is output on the pin MISO.

- **MOSI: MOSI Pin**

0: PIO is input on the pin MOSI.

1: PIO is output on the pin MOSI.

- **NPCS0: NPCS0 Pin**

0: PIO is input on the pin NPCS0/NSS.

1: PIO is output on the pin NPCS0/NSS.

- **NPCSx [x = 1..3]: NPCSx Pin**

0: PIO is input on the pin NPCSx.

1: PIO is output on the pin NPCSx.

22.6.7 SPI PIO Set Output Data Register

Name: SPI_SODR
Access: Write-only
Address: 0xFFFB4030

22.6.8 SPI PIO Clear Output Data Register

Name: SPI_CODR
Access: Write-only
Address: 0xFFFB4034

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	NPCS3	NPCS2	NPCS1	NPCS0	MOSI	MISO	SPCK
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SPCK: SPCK Pin**

0: The output data for the pin SCK is programmed to 0.

1: The output data for the pin SCK is programmed to 1.

- **MISO: MISO Pin**

0: The output data for the pin MISO is programmed to 0.

1: The output data for the pin MISO is programmed to 1.

- **MOSI: MOSI Pin**

0: The output data for the pin MOSI is programmed to 0.

1: The output data for the pin MOSI is programmed to 1.

- **NPCS0: NPCS0 Pin**

0: The output data for the pin NPCS0/NSS is programmed to 0.

1: The output data for the pin NPCS0/NSS is programmed to 1.

- **NPCSx [x = 1..3]: NPCSx Pin**

0: The output data for the pin NPCSx is programmed to 0.

1: The output data for the pin NPCSx is programmed to 1.

22.6.9 SPI PIO Output Data Status Register

Name: SPI_ODSR
Access: Read-only
Address: 0xFFFB4038

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	NPCS3	NPCS2	NPCS1	NPCS0	MOSI	MISO	SPCK
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SPCK: SPCK Pin**

0: The output data for the pin SCK is programmed to 0.

1: The output data for the pin SCK is programmed to 1.

- **MISO: MISO Pin**

0: The output data for the pin MISO is programmed to 0.

1: The output data for the pin MISO is programmed to 1.

- **MOSI: MOSI Pin**

0: The output data for the pin MOSI is programmed to 0.

1: The output data for the pin MOSI is programmed to 1.

- **NPCS0: NPCS0 Pin**

0: The output data for the pin NPCS0/NSS is programmed to 0.

1: The output data for the pin NPCS0/NSS is programmed to 1.

- **NPCSx [x = 1..3]: NPCSx Pin**

0: The output data for the pin NPCSx is programmed to 0.

1: The output data for the pin NPCSx is programmed to 1.

22.6.10 SPI PIO Pin Data Status Register

Name: SPI_PDSR
Access: Read-only
Address: 0xFFFB403C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	NPCS3	NPCS2	NPCS1	NPCS0	MOSI	MISO	SPCK
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SPCK: SPCK Pin**

0: The pin SCK is at logic 0.

1: The pin SCK is at logic 1.

- **MISO: MISO Pin**

0: The pin MISO is at logic 0.

1: The pin MISO is at logic 1.

- **MOSI: MOSI Pin**

0: The pin MOSI is at logic 0.

1: The pin MOSI is at logic 1.

- **NPCS0: NPCS0 Pin**

0: The pin NPCS0/NSS is at logic 0.

1: The pin NPCS0/NSS is at logic 1.

- **NPCSx [x = 1..3]: NPCSx Pin**

0: The pin NPCSx is at logic 0.

1: The pin NPCSx is at logic 1.

22.6.11 SPI PIO Multidriver Enable Register

Name: SPI_MDER
Access: Write-only
Address: 0xFFFB4040

22.6.12 SPI PIO Multidriver Disable Register

Name: SPI_MDDR
Access: Write-only
Address: 0xFFFB4044

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	NPCS3	NPCS2	NPCS1	NPCS0	MOSI	MISO	SPCK
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SPCK: SPCK Pin**

0: The pin SCK is not configured as an open drain.

1: The pin SCK is configured as an open drain.

- **MISO: MISO Pin**

0: The pin MISO is not configured as an open drain.

1: The pin MISO is configured as an open drain.

- **MOSI: MOSI Pin**

0: The pin MOSI is not configured as an open drain.

1: The pin MOSI is configured as an open drain.

- **NPCS0: NPCS0 Pin**

0: The pin NPCS0/NSS is not configured as an open drain.

1: The pin NPCS0/NSS is configured as an open drain.

- **NPCSx [x = 1..3]: NPCSx Pin**

0: The pin NPCSx is not configured as an open drain.

1: The pin NPCSx is configured as an open drain.

22.6.13 SPI PIO Multidriver Status Register

Name: SPI_MDSR
Access: Read-only
Address: 0xFFFB4048

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	NPCS3	NPCS2	NPCS1	NPCS0	MOSI	MISO	SPCK
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SPCK: SPCK Pin**

0: The pin SCK is not configured as an open drain.

1: The pin SCK is configured as an open drain.

- **MISO: MISO Pin**

0: The pin MISO is not configured as an open drain.

1: The pin MISO is configured as an open drain.

- **MOSI: MOSI Pin**

0: The pin MOSI is not configured as an open drain.

1: The pin MOSI is configured as an open drain.

- **NPCS0: NPCS0 Pin**

0: The pin NPCS0/NSS is not configured as an open drain.

1: The pin NPCS0/NSS is configured as an open drain.

- **NPCSx [x = 1..3]: NPCSx Pin**

0: The pin NPCSx is not configured as an open drain.

1: The pin NPCSx is configured as an open drain.

22.6.14 SPI Enable Clock Register

Name: SPI_ECR
Access: Write-only
Address: 0xFFFB4050

22.6.15 SPI Disable Clock Register

Name: SPI_DCR
Access: Write-only
Address: 0xFFFB4054

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
–	–	–	–	–	–	SPI	PIO

- **PIO: PIO Clock Status**

0: PIO clock disabled.

1: PIO clock enabled.

- **SPI: SPI Clock Status**

0: SPI clock disabled.

1: SPI clock enabled.

22.6.16 SPI Power Management Status Register

Name: SPI_PMSR
Access: Read-only
Address: 0xFFFB4058

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
–	–	–	–	–	–	SPI	PIO

- **PIO: PIO Clock Status**

0: PIO clock disabled.

1: PIO clock enabled.

- **SPI: SPI Clock Status**

0: SPI clock disabled.

1: SPI clock enabled.

Note: The SPI_PMSR register is not reset by software reset.

22.6.17 SPI Control Register

Name: SPI_CR
Access: Write-only
Address: 0xFFFB4060

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
–	–	–	–	–	SPIDIS	SPIEN	SWRST

- **SWRST: SPI Software Reset**

0: No effect.

1: Resets the SPI.

A software-triggered hardware reset of the SPI is performed. It resets all the registers, including PIO and PMC registers (except SPI_PMSR).

- **SPIEN: SPI Enable**

0: No effect.

1: Enables the SPI.

This enables the SPI to transfer and receive data.

- **SPIDIS: SPI Disable**

0: No effect.

1: Disables the SPI.

All pins will be set to input mode and no data is received or transmitted.

In case a transfer is in progress, the transfer is finished before the SPI is disabled.

In case both SPIEN and SPIDIS are equal to one when the control register is written, the SPI is disabled.

22.6.18 SPI Mode Register

Name: SPI_MR
Access: Read/Write
Address: 0xFFFB4064

31	30	29	28	27	26	25	24
DLYBCS							
23	22	21	20	19	18	17	16
–	–	–	–	PCS			
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
LLB	–	–	–	DIV32	PCSDEC	PS	MSTR

- **MSTR: Master/Slave Mode**

0: SPI is in Slave Mode. The SPI in Slave Mode can not be used with the PDC for data transmission or reception.

1: SPI is in Master Mode.

MSTR configures the SPI for either master or Slave Mode operation.

- **PS: Peripheral Select**

0: Fix peripheral select.

1: Variable peripheral select.

The peripheral mode is only used in Master Mode. In case of fix peripheral select, the selected peripheral is defined in the mode register. For variable peripheral select, it is defined in the transmit data register.

- **PCSDEC: Chip Select Decode**

0: The chip selects are directly connected to a peripheral device.

1: The four chip select lines are connected to a 4-to-16 decoder.

In case PCSDEC is one, up to 16 Chip Select signals can be generated with the four lines using an external 4 to 16 decoder.

The Chip Select Registers define the characteristics of the 16 chip selects according to the following rules:

– SPI_CSR0 defines peripheral chip select signals 0 to 3.
– SPI_CSR1 defines peripheral chip select signals 4 to 7.
– SPI_CSR2 defines peripheral chip select signals 8 to 11.
– SPI_CSR3 defines peripheral chip select signals 12 to 15.

- **DIV32: Clock Selection**

0: SPI Master Clock equals CORECLK.

1: SPI Master Clock equals CORECLK/32.

- **LLB: Local Loopback**

0: Local loopback path disabled.

1: Local loopback path enabled.

LLB controls the local loopback on the data serializer for testing.

- **PCS[3:0]: Peripheral Chip Select**

This field is only used if single peripheral select is active (PS = 0).

If PCSDEC = 0:

PCS[3:0] = xxx0 NPCS[3:0] = 1110
PCS[3:0] = xx01 NPCS[3:0] = 1101
PCS[3:0] = x011 NPCS[3:0] = 1011
PCS[3:0] = 0111 NPCS[3:0] = 0111
PCS[3:0] = 1111 forbidden (no peripheral is selected)

where x = don't care.

If PCSDEC = 1:

NPCS[3:0] output signals = PCS[3:0]

- **DLYBCS[7:0]: Delay Between Chip Selects**

This field defines the delay from one NPCS inactive to the activation of another NPCS. The DLYBCS[7:0] time will guarantee non-overlapping chip selects and resolves bus contentions in case of peripherals with long data float times.

When DLYBCS[7:0] is less than six, six SPI Master Clock periods are inserted by default.

Else, the following equation determines the delay:

$$NPCS\_to\_SCK\_Delay = DLYBCS[7:0] \times SPI\_Master\_Clock\_Period$$

22.6.19 SPI Status Register

Name: SPI_SR
Access: Read-only
Address: 0xFFFB4070

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	NPCS3	NPCS2	NPCS1	NPCS0	MOSI	MISO	SPCK
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	SPIENS
7	6	5	4	3	2	1	0
–	–	TEND	REND	SPIOVRE	MODF	TDRE	RDRF

Note: This register is a “read-active” register, which means that reading it can affect the state of some bits. When reading SPI_SR register, following bits are cleared if set: MODF, SPIOVRE, REND, TEND, SPCK, MISO, MOSI, NPCS0, NPCS1, NPCS2 and NPCS3.

- **RDRF: Receive Data Register Full**

0: No data has been received since the last read of SPI_RDR.

1: A data has been received and the receive data has been transferred from the serializer in SPI_RDR since the last read of SPI_RDR.

- **TDRE: Transmit Data Register Empty**

0: Data has been written in SPI_TDR and not yet transferred to the serializer.

1: The last data written in the Transmit Data Register has been transferred to the serializer.

TDRE equals zero when the SPI is disabled or at reset. The SPI enable command sets this bit to one.

- **MODF: Mode Fault Error**

0: No Mode Fault has been detected since the last read of SPI_SR.

1: A Mode Fault occurred since the last read of SPI_SR.

- **SPIOVRE: Overrun Error**

0: No overrun has been detected since the last read of SPI_SR.

1: An overrun has occurred since the last read of SPI_SR.

An overrun occurs when SPI_RDR is loaded at least twice from the serializer since the last read of the SPI_RDR.

- **REND: Reception End**

0: No reception or reception processing.

1: End of reception.

- **TEND: Transfer End**

0: No transfer or transfer processing.

1: End of transfer.

- **SPIENS: SPI Enable**

0: SPI is disabled.

1: SPI is enabled.

- **SPCK, MISO, MOSI, NPCS0, NPCS1, NPCS2, NPCS3: PIO Interrupt**

These bits indicate for each pin when an input logic value change has been detected (rising or falling edge). This is valid whether the PIO is selected for the pin or not.

These bits are reset to zero following a read and at reset.

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

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

22.6.20 SPI Interrupt Enable Register

Name: SPI_IER
Access: Write-only
Address: 0xFFFB4074

22.6.21 SPI Interrupt Disable Register

Name: SPI_IDR
Access: Write-only
Address: 0xFFFB4078

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	NPCS3	NPCS2	NPCS1	NPCS0	MOSI	MISO	SPCK
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	TEND	REND	SPIOVRE	MODF	TDRE	RDRF

- **RDRF: Receive Data Register Full Interrupt Mask**

0: Receive Data Register Full Interrupt is disabled.

1: Receive Data Register Full Interrupt is enabled.

- **TDRE: Transmit Data Register Empty Interrupt Mask**

0: Transmit Data Register Empty Interrupt is disabled.

1: Transmit Data Register Empty Interrupt is enabled.

- **MODF: Mode Fault Error Interrupt Mask**

0: Mode Fault Interrupt is disabled.

1: Mode Fault Interrupt is enabled.

Only used in Master Mode.

- **SPIOVRE: Overrun Error Interrupt Mask**

0: Overrun Error Interrupt is disabled.

1: Overrun Error Interrupt is enabled.

- **REND: PDC Reception End Interrupt Mask**

0: Reception End Interrupt is disabled.

1: Reception End Interrupt is enabled.

- **TEND: PDC Transfer End Interrupt Mask**

0: Transfer End Interrupt is disabled.

1: Transfer End Interrupt is enabled.

- **SPCK, MISO, MOSI, NPCS0, NPCS1, NPCS2, NPCS3: PIO Interrupt Mask**

These bits show which pins have interrupts enabled. They are updated when interrupts are enabled or disabled by writing to SPI_IER or SPI_IDR.

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

1: Interrupt is enabled on the corresponding input pin.

22.6.22 SPI Interrupt Mask Register

Name: SPI_IMR
Access: Read-only
Address: 0xFFFB407C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	NPCS3	NPCS2	NPCS1	NPCS0	MOSI	MISO	SPCK
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	TEND	REND	SPIOVRE	MODF	TDRE	RDRF

- **RDRF: Receive Data Register Full Interrupt Mask**

0: Receive Data Register Full Interrupt is disabled.

1: Receive Data Register Full Interrupt is enabled.

- **TDRE: Transmit Data Register Empty Interrupt Mask**

0: Transmit Data Register Empty Interrupt is disabled.

1: Transmit Data Register Empty Interrupt is enabled.

- **MODF: Mode Fault Error Interrupt Mask**

0: Mode Fault Interrupt is disabled.

1: Mode Fault Interrupt is enabled.

Only used in Master Mode.

- **SPIOVRE: Overrun Error Interrupt Mask**

0: Overrun Error Interrupt is disabled.

1: Overrun Error Interrupt is enabled.

- **REND: PDC Reception End Interrupt Mask**

0: Reception End Interrupt is disabled.

1: Reception End Interrupt is enabled.

- **TEND: PDC Transfer End Interrupt Mask**

0: Transfer End Interrupt is disabled.

1: Transfer End Interrupt is enabled.SPCK, MISO, MOSI, NPCS0, NPCS1, NPCS2, NPCS3: PIO Interrupt Mask

These bits show which pins have interrupts enabled. They are updated when interrupts are enabled or disabled by writing to SPI_IER or SPI_IDR.

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

1: Interrupts is enabled on the corresponding input pin.

22.6.23 SPI Receive Data Register

Name: SPI_RDR
Access: Read-only
Address: 0xFFFB4080

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

Note: When reading this register, RDRF bit is cleared in the SPI_RDR.

- **RD[15:0]: Receive Data**

Data received by the SPI interface is stored in this register. Data stored is right-justified. Unused bits are read at zero.

- **PCS[3:0]: Peripheral Chip Select Status**

In Master Mode only, these bits indicate the value on the NPCS pins at the end of a transfer.

22.6.24 SPI Transmit Data Register

Name: SPI_TDR
Access: Write-only
Address: 0xFFFB4084

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

- **TD[15:0]: Transmit Data**

Data that is to be transmitted by the SPI is stored in this register. Information to be transmitted must be written to the transmit data register in a right-justified format.

- **PCS[3:0]: Peripheral Chip Select**

This field is only used if multiple peripheral select is active (PS = 1).

If PCSDEC = 0:

PCS[3:0] = xxx0 NPCS[3:0] = 1110
PCS[3:0] = xx01 NPCS[3:0] = 1101
PCS[3:0] = x011 NPCS[3:0] = 1011
PCS[3:0] = 0111 NPCS[3:0] = 0111
PCS[3:0] = 1111 forbidden (no peripheral is selected)

where x = don't care.

If PCSDEC = 1:

NPCS[3:0] output signals = PCS[3:0].

22.6.25 SPI Chip Select Register x [x = 0..3]

Name: SPI_CSRx
Access: Read/Write
Address: 0xFFFB4090...0xFFFB409C

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

- **CPOL: Clock Polarity**

0: The inactive state value of SCK is logic level zero.

1: The inactive state value of SCK is logic level one.

CPOL is used to determine the inactive state value of the serial clock (SCK). It is used with NCPHA to produce a desired clock/data relationship between master and the slave devices.

- **NCPHA: Clock Phase**

0: Data is changed on the leading edge of SCK and captured on the following edge of SCK.

1: Data is captured on the leading edge of SCK and changed on the following edge of SCK.

NCPHA determines which edge of SCK causes data to change and which edge causes data to be captured.

NCPHA is used with CPOL to produce a desired clock/data relationship between master and slave devices.

- **BITS[3:0]: Bits Per Transfer**

The BITS field determines the number of data bits transferred. Reserved values default to 8 bits.

BITS[3:0]	Bits Per Transfer	BITS[3:0]	Bits Per Transfer
0000	8	0111	15
0001	9	1000	16
0010	10	1001	Reserved
0011	11	1010	Reserved
0100	12	1011	Reserved
0101	13	11XX	Reserved
0110	14		

- **SCBR[7:0]: Serial Clock Baud Rate**

The SPI interface uses a modulus counter to derive SCK baud rate from the SPI Clock, selected between CORECLK and CORECLK/32. Baud rate is selected by writing a value from 2 to 255 into SCBR[7:0]. The following equation determines the SCK baud rate:

$$SCK\_Baud\_Rate = SPI\_Master\_Clock\_Frequency / (2 \times SCBR[7:0])$$

Note: Giving SCBR[7:0] a value of zero or one disables the baud rate generator. SCK is disabled and assumes its inactive state value. No serial transfers occur. At reset, baud rate is disabled.

- **DLYBS[7:0]: Delay Before SCK**

This field defines the length of delay from NPCS valid to the first valid SCK transition.

When DLYBS[7:0] equals zero, the NPCS valid to SCK transition is one-half SCK clock period.

Else, the following equation determines the delay:

$$NPCS\_to\_SCK\_Delay = DLYBS[7:0] \times SPI\_Master\_Clock\_Period$$

- **DLYBCT[7:0]: Delay Between Consecutive Transfers**

This field determines the delay between two consecutive transfers with the same peripheral without removing the chip select or the length of delay after transfer before chip select is deselected.

When DLYBCT[7:0] equals zero, the delay is equal to four SPI Clock periods.

Else, the following equation determines the delay:

$$Delay\_After\_Transfer = 32 \times DLYBCT[7:0] \times SPI\_Master\_Clock\_Period$$

22.7 Timing Diagrams

Figure 22-4. SPI in Master Mode, Phase = 0

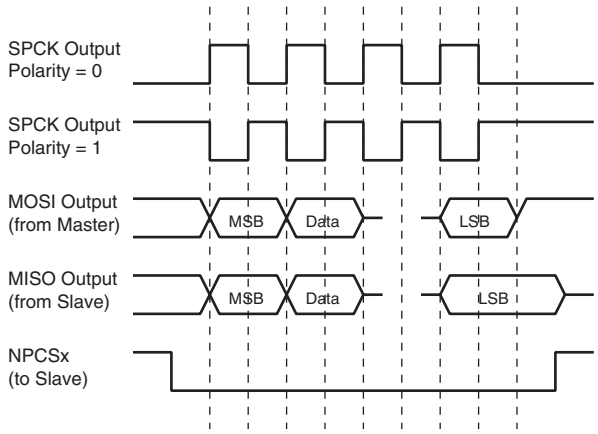


Figure 22-5. SPI in Master Mode, Phase = 1

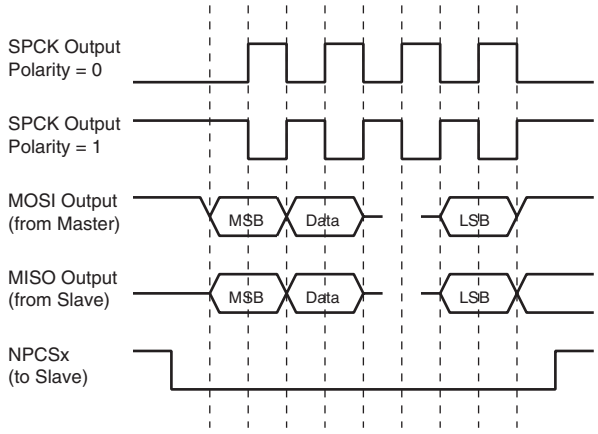
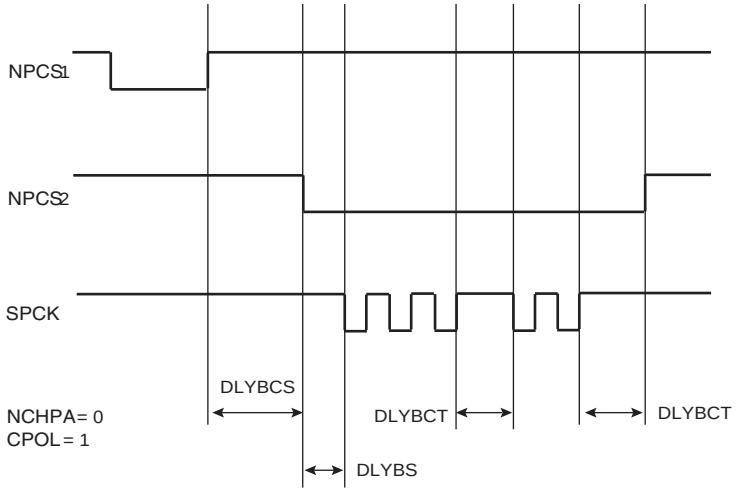


Figure 22-6. DLYBCS, DLYBS and DLYBCT



23. Unified Parallel I/O Controller (UPIO)

23.1 Description

The AT91SAM7A1 microcontroller has 18 unified programmable I/O lines. These lines are controlled by the UPIO module and are not multiplexed with other module pins. The UPIO controller also provides an internal interrupt signal to the Generic Interrupt Controller.

Each PIO block in the peripheral is controlled through the peripheral interface. The PIO block clock is enabled/disabled by the peripheral power management controller.

23.2 Output Selection

The user can select the direction of each individual I/O signal (input or output) with the UPIO_OER (Output Enable) register or the UPIO_ODR (Output Disable) register. The output status of the I/O signals can be read in the register UPIO_OSR (Output Status).

23.3 I/O Levels

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

If a pin is defined as output, the level is programmed using the UPIO_SODR (Set Output Data) and UPIO_CODR (Clear Output Data) registers. In this case, the programmed value can be read in UPIO_ODSR (Output Data Status).

If a pin is not defined as output, the level is determined by the external circuit.

The level on the pin can be read in the register UPIO_PDSR (Pin Data Status).

23.4 Interrupts

The UPIO controller block also provides an internal interrupt to the GIC.

Each parallel I/O can be programmed to generate an interrupt when a level change occurs. This is controlled by the UPIO_IER (Interrupt Enable) and UPIO_IDR (Interrupt Disable) registers which enable/disable the I/O interrupt by setting/clearing the corresponding bit in the UPIO_IMR.

When a change in level occurs, the corresponding bit in the UPIO_SR (Interrupt Status) is set whether the pin is defined as input or output. If the corresponding interrupt in UPIO_IMR (Interrupt Mask) is enabled, the UPIO interrupt is asserted.

The UPIO interrupt is cleared when a read access is performed in the UPIO_SR register.

23.5 User Interface

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

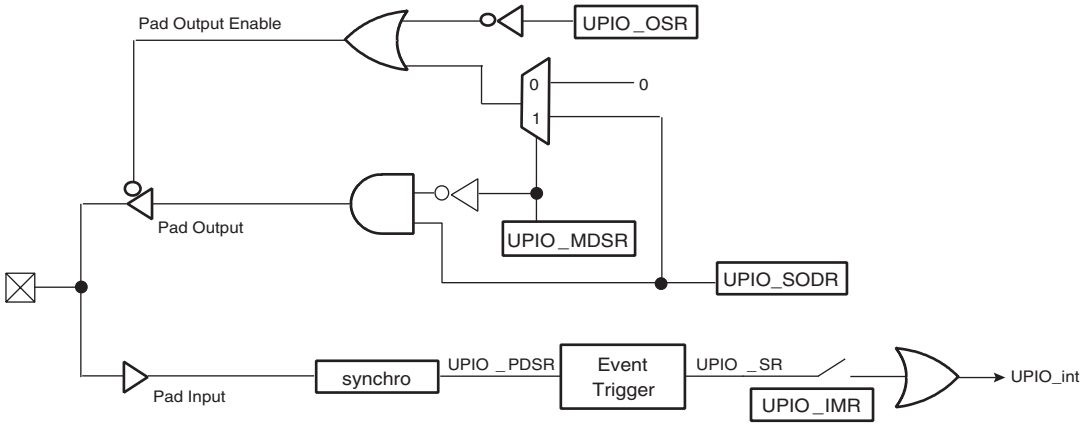
23.6 Open Drain/Push-pull Output

The UPIO can be configured either as an open drain output (only drives a low level) or as a push/pull output (drives high and low levels).

When the UPIO is configured as open drain (multidriver), an external pull-up is necessary to guarantee a logic level (logical one) when the pin is not being driven.

Registers PERIPHERAL_MDER (Multidriver Enable) and PERIPHERAL_MDDR (Multidriver Disable) control this option and configure the I/O as open drain or push/pull, respectively. The multidriver option can be selected whether the I/O pin is controlled by the UPIO Controller or the peripheral controller. Bits at logical one in the register PERIPHERAL_MDSR (Multidriver Status) indicate pins configured as open drain.

Figure 23-1. UPIO Block Diagram



23.7 Unified Parallel I/O Controller (UPIO) Memory Map

Table 23-1. UPIO Memory Map

Address	Register	Name	Access	Reset State
0xFFFD8000 – 0xFFFD800C	Reserved	–	–	–
0xFFFD8010	Output Enable Register	UPIO_OER	Write-only	–
0xFFFD8014	Output Disable Register	UPIO_ODR	Write-only	–
0xFFFD8018	Output Status Register	UPIO_OSR	Read-only	0x00000000
0xFFFD801C – 0xFFFD802C	Reserved	–	–	–
0xFFFD8030	Set Output Data Register	UPIO_SODR	Write-only	–
0xFFFD8034	Clear Output Data Register	UPIO_CODR	Write-only	–
0xFFFD8038	Output Data Status Register	UPIO_ODSR	Read-only	0x00000000
0xFFFD803C	Pin Data Status Register	UPIO_PDSR	Read-only	0xXXXXXXXX
0xFFFD8040	Multidriver Enable Register	UPIO_MDER	Write-only	–
0xFFFD8044	Multidriver Disable Register	UPIO_MDDR	Write-only	–
0xFFFD8048	Multidriver Status Register	UPIO_MDSR	Read-only	0x00000000
0xFFFD804C	Reserved	–	–	–
0xFFFD8050	Enable Clock Register	UPIO_ECR	Write-only	–
0xFFFD8054	Disable Clock Register	UPIO_DCR	Write-only	–
0xFFFD8058	Power Management Status Register	UPIO_PMSR	Read-only	0x00000000
0xFFFD805C	Reserved	–	–	–
0xFFFD8060	Control Register	UPIO_CR	Write-only	–
0xFFFD8064 – 0xFFFD806C	Reserved	–	–	–
0xFFFD8070	Status Register	UPIO_SR	Read-only	0x00000000
0xFFFD8074	Interrupt Enable Register	UPIO_IER	Write-only	–
0xFFFD8078	Interrupt Disable Register	UPIO_IDR	Write-only	–
0xFFFD807C	Interrupt Mask Register	UPIO_IMR	Read-only	0x00000000

23.7.1 UPIO Output Enable Register

Name: UPIO_OER
Access: Write-only
Address: 0xFFFD8010

23.7.2 UPIO Output Disable Register

Name: UPIO_ODR
Access: Write-only
Address: 0xFFFD8014

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	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

- **Px: Px Pin**

0: The corresponding PIO is input on this line.

1: The corresponding PIO is output on this line.

23.7.3 UPIO Output Status Register

Name: UPIO_OSR
Access: Read-only
Address: 0xFFFD8018

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	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

- **Px: Px Pin**

0: The corresponding PIO is input on this line.

1: The corresponding PIO is output on this line.

23.7.4 UPIO Set Output Data Register

Name: UPIO_SODR
Access: Write-only
Address: 0xFFFD8030

23.7.5 UPIO Clear Output Data Register

Name: UPIO_CODR
Access: Write-only
Address: 0xFFFD8034

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	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

- **Px: Px Pin**

0: The output data for the corresponding pin is programmed to 0.

1: The output data for the corresponding pin is programmed to 1.

23.7.6 UPIO Output Data Status Register

Name: UPIO_ODSR
Access: Read-only
Address: 0xFFFD8038

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	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

- **Px: Px Pin**

0: The output data for the corresponding pin is programmed to 0.

1: The output data for the corresponding pin is programmed to 1.

23.7.7 UPIO Pin Data Status Register

Name: UPIO_PDSR
Access: Read-only
Address: 0xFFFD8030

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	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

- **Px: Px Pin**

0: The corresponding pin is at logic 0.

1: The corresponding pin is at logic 1.

23.7.8 UPIO Multidriver Enable Register

Name: UPIO_MDER
Access: Write-only
Address: 0xFFFD8040

23.7.9 UPIO Multidriver Disable Register

Name: UPIO_MDDR
Access: Write-only
Address: 0xFFFD8044

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	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

- **Px: Px Pin**

0: PIO is not configured as an open drain.

1: PIO is configured as an open drain.

23.7.10 UPIO Multidriver Status Register

Name: UPIO_MDSR
Access: Read-only
Address: 0xFFFD8048

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	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

- **Px: Px Pin**

0: PIO is not configured as an open drain.

1: PIO is configured as an open drain.

23.7.11 UPIO Enable Clock Register

Name: UPIO_ECR
Access: Write-only
Address: 0xFFFD8050

23.7.12 UPIO Disable Clock Register

Name: UPIO_DCR
Access: Write-only
Address: 0xFFFD8054

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
–	–	–	–	–	–	–	PIO

- **PIO: PIO Clock Status**

0: PIO clock disabled.

1: PIO clock enabled.

23.7.13 UPIO Power Management Status Register

Name: UPIO_PMSR
Access: Read-only
Address: 0xFFFD8058

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
–	–	–	–	–	–	–	PIO

- **PIO: PIO Clock Status**

0: PIO clock disabled.

1: PIO clock enabled.

23.7.14 UPIO Control Register

Name: UPIO_CR
Access: Write-only
Address: 0xFFFD8060

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
–	–	–	–	–	–	–	SWRST

- **SWRST: PIO Software Reset**

0: No effect.

1: Resets the PIO.

A software-triggered hardware reset of the PIO is performed. It resets all the registers except the UPIO_PMSR.

23.7.15 UPIO Status Register

Name: UPIO_SR
Access: Read-only
Address: 0xFFFD8070

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	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

Note: This register is a “read-active” register, thus reading it can affect the state of some bits. When read UPIO_SR register, all bits set to logical 1 are cleared.

- **Px: PIO Interrupt Status**

These bits indicate for each pin when an input logic value change has been detected (rising or falling edge).

These bits are reset to zero following a read and at reset.

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

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

23.7.16 UPIO Interrupt Enable Register

Name: UPIO_IER
Access: Write-only
Address: 0xFFFD8074

23.7.17 UPIO Interrupt Disable Register

Name: UPIO_IDR
Access: Write-only
Address: 0xFFFD8078

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	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

- **Px: PIO Interrupt Mask**

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

1: Interrupt is enabled on the corresponding input pin.

23.7.18 UPIO Interrupt Mask Register

Name: UPIO_IMR
Access: Read-only
Address: 0x07C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	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

- **Px: PIO Interrupt Mask**

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

1: Interrupt is enabled on the corresponding input pin.

24. Power Management Controller (PMC)

24.1 Description

The AT91SAM7A1 Power Management Controller allows optimization of power consumption. The PMC controls the clock inputs of the ARM core and of the PDC module.

When the ARM core clock is disabled, the current instruction is finished before the clock is stopped. The clock can be re-enabled by any interrupt or by any hardware reset.

The PDC clock cannot be stopped during PDC transfers (if any TCR register is different from 0). The request is stored but the Power Management Controller will wait until the end of the transfers to stop the PDC clock (the Power Management Controller needs an authorization from the PDC before it stops the PDC clock).

24.2 Power Management Controller (PMC) Memory Map

Table 24-1. PMC Memory Map

Offset	Register	Name	Access	Reset State
0xFFFF4000 – 0xFFFF404C	Reserved	–	–	–
0xFFFF4050	Enable Clock Register	PMC_ECR	Write-only	–
0xFFFF4054	Disable Clock Register	PMC_DCR	Write-only	–
0xFFFF4058	Power Management Status Register	PMC_PMSR	Read-only	0x00000001

24.2.1 PMC Enable Clock Register

Name: PMC_ECR
Access: Write-only
Address: 0xFFFF4050

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
–	–	–	–	–	–	PDC	–

- **PDC: PDC Clock**

0: PDC clock is disabled, or user instructed PDC clock to be disabled during PDC transfers. The PDC clock is disabled when all TCR registers reach 0 (all transfers are finished).

1: PDC clock is enabled.

24.2.2 PMC Disable Clock Register

Name: PMC_DCR
Access: Write-only
Address: 0xFFFF4054

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
–	–	–	–	–	–	PDC	ARM

- **ARM: ARM Clock**

0: ARM core clock is disabled.

1: ARM core clock is enabled.

- **PDC: PDC Clock**

0: PDC clock is disabled, or user instructed PDC clock to be disabled during PDC transfers. The PDC clock is disabled when all TCR registers reach 0 (all transfers are finished).

1: PDC clock is enabled.

24.2.3 PMC Power Management Status Register

Name: PMC_PMSR
Access: Read-only
Address: 0xFFFF4058

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
–	–	–	–	–	–	PDC	ARM

- **ARM: ARM Clock**

0: ARM core clock is disabled.

1: ARM core clock is enabled.

- **PDC: PDC Clock**

0: PDC clock is disabled, or user instructed PDC clock to be disabled during PDC transfers. The PDC clock is disabled when all TCR registers reach 0 (all transfers are finished).

1: PDC clock is enabled.

25. Controller Area Network (CAN)

25.1 Description

The Controller Area Network (CAN) is a serial communications protocol that supports distributed real-time control with a very high level of security.

Possible applications range from high-speed networks to low-cost multiplex wiring.

This section describes how to achieve compatibility between any two CAN implementations. Compatibility, however, has different aspects regarding, e.g., electrical features and the interpretation of data to be transferred.

To achieve design transparency and implementation flexibility, the CAN has been subdivided into different layers according to the ISO/OSI Reference Model:

- Data Link Layer
  - Logical Link Control (LLC) sublayer
  - Medium Access Control (MAC) sublayer
- Physical Layer

Note that in previous versions of the CAN specification, services and functions of the LLC and MAC sublayers of the Data Link Layer were described in layers denoted as 'object layer' and 'transfer layer'.

The tasks of the LLC sublayer include

- determining which messages received by the LLC sublayer are to be accepted
- providing services for data transfer and for remote data request
- providing the means for recovery management and overload notifications

There is much flexibility in defining object handling.

The tasks of the MAC sublayer concern primarily the transfer protocol, i.e., controlling framing, performing arbitration, error checking, error and fault confinement. Within the MAC sublayer, it is determined whether the bus is free to start signaling a new transmission or whether a reception is just starting. Also, some general features of bit timing are regarded as a task of the MAC sublayer. One of the constraints of the MAC sublayer is that it is not possible to make modifications.

The task of the physical layer is the actual transfer of bits between the different nodes with respect to electrical properties. Within a network, the physical layer has to be the same for all nodes. There are, however, many possible implementations of a physical layer.

In the following, the MAC sublayer and a small part of the LLC sublayer of the Data Link Layer are defined and the consequences of the CAN protocol on the surrounding layers are described.

25.2 Basic Concepts

25.2.1 CAN Properties

The CAN has the following properties:

- Prioritization of messages
- Guarantee of latency times

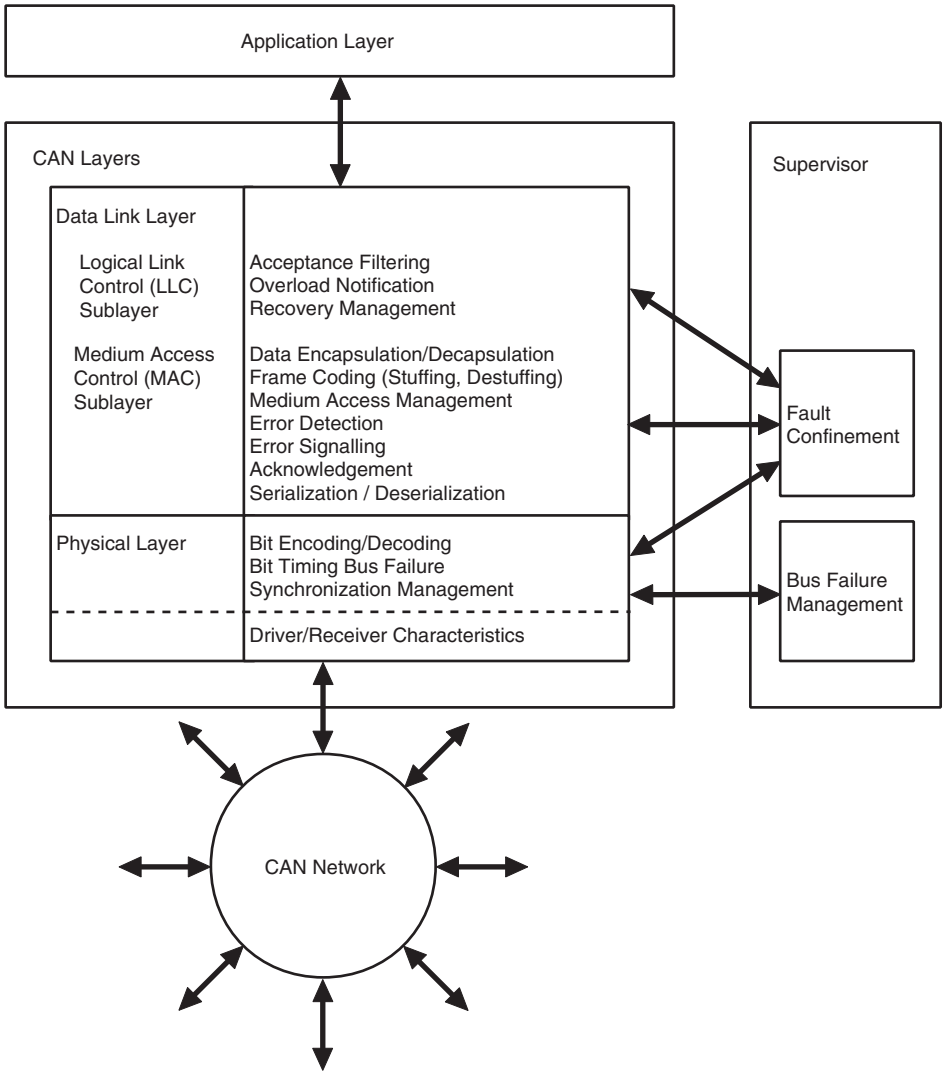
- Configuration flexibility
- Multicast reception with time synchronization
- System wide data consistency
- Multi-master functions
- Error detection and error signaling
- Automatic retransmission of corrupted messages as soon as the bus is idle again
- Distinction between temporary errors and permanent failures of nodes and autonomous switching off of defect nodes.

25.2.2 Layered Structure of a CAN Node

The layered architecture of the CAN is compliant with the ISO/OSI reference model:

- The LLC sublayer is concerned with message filtering, overload notification and recovery management.
- The MAC sublayer represents the kernel of the CAN protocol. It presents messages received from the LLC sublayer and accepts messages to be transmitted to the LLC sublayer. The MAC sublayer is responsible for Message Framing, Arbitration, Acknowledgement, Error Detection and Signalling. The MAC sublayer is supervised by a management entity called Fault Confinement which is a self-checking mechanism for distinguishing short disturbances from permanent failures.
- The physical layer defines how signals are actually transmitted and deals with the description of bit timing, bit encoding, and synchronization. Within this description, the physical layer is not defined, as it will vary according to the requirements of individual applications (for example, transmission medium and signal level implementations).

Figure 25-1. Layered Structure of a CAN Node



25.2.3 Messages

Information on the bus is sent in fixed format messages of different but limited length (see ”Message Transfer” on page 274). When the bus is free, any connected unit may start to transmit a new message.

25.2.4 Information Routing

In CAN systems, a CAN node does not make use of any information about the system configuration (e.g., node addresses). This has several important consequences:

- System flexibility: Nodes can be added to the CAN network without requiring any change in the software or hardware of any node and application layer.
- Message routing: The content of a message is named by an identifier. The identifier does not indicate the destination of the message, but describes the meaning of the data, so that all nodes in the network are able to decide, by message filtering, whether the data is to be acted upon by them or not.

- Multicast: As a consequence of the message filtering, any number of nodes can receive and simultaneously act upon the same message.
- Data consistency: Within a CAN network, it is guaranteed that a message is simultaneously accepted either by all nodes or by no node. Thus, data consistency of a system is achieved by the concepts of multicast and by error handling.

25.2.5 Bit Rate

The speed of the CAN may be different in different systems. However, in a given system the bit-rate is uniform and fixed.

25.2.6 Priorities

The identifier defines a static message priority during bus access.

25.2.7 Remote Data Request

By sending a remote frame, a node requiring data may request another node to send the corresponding data frame. The data frame and the corresponding remote frame are named by the same identifier.

25.2.8 Multi-master Function

When the bus is free, any unit may start transmitting a message. The unit with the message of highest priority to be transmitted gains bus access.

25.2.9 Arbitration

Whenever the bus is free, any unit may start transmitting a message. If two or more units start transmitting messages at the same time, the bus access conflict is resolved by bit-wise arbitration using the identifier. The mechanism of arbitration guarantees that neither information nor time is lost. If a data frame and a remote frame with the same identifier are initiated at the same time, the data frame prevails over the remote frame. During arbitration every transmitter compares the level of the bit transmitted with the level that is monitored on the bus. If these levels are equal the unit may continue to send. When a recessive level is sent and a dominant level is monitored (see ”Bus Values” on page 273), the unit has lost arbitration and must withdraw without sending one more bit.

25.2.10 Data Transfer Security

In order to achieve the utmost security of data transfer, powerful measures for error detection, signalling and self-checking are implemented in every CAN node.

25.2.10.1 Error Detection

For detecting errors, the following methods are used:

- Monitoring (transmitters compare the bit levels to be transmitted with the bit levels detected on bus)
- Cyclic Redundancy Check (CRC)
- Bit stuffing
- Message frame check

25.2.10.2 Performance of Error Detection

The error detection mechanisms have the following properties:

- All global errors are detected by means of monitoring.

- All local errors at transmitters are detected by means of monitoring.
- Up to 5 randomly distributed errors in a message are detected by means of CRC.
- Burst errors of length less than 15 in a message are detected by means of CRC.
- Errors of any odd number in a message are detected by means of CRC.

Total residual error probability for undetected corrupted messages is less than

$$\text{Message Error Rate} \times 4.7 \times 10^{11}$$

25.2.11 Error Signaling and Recovery Time

Corrupted messages are flagged by any node detecting an error. Such messages are aborted and will be retransmitted automatically. The recovery time from detecting an error until the start of the next message is at most 31 bit times if there is no further error.

25.2.12 Fault Confinement

CAN nodes are able to distinguish short disturbances from permanent failures. Defective nodes are switched off.

25.2.13 Connections

The CAN serial communication link is a bus to which a number of units may be connected. This number has no theoretical limit. In practice, the total number of units is limited by delay times and/or electrical loads on the bus line.

25.2.14 Single Channel

The bus consists of a single bidirectional channel that carries bits. From this data, resynchronization information can be derived. The way in which this channel is implemented is not fixed in this document, e.g., single wire (plus ground), two differential wires, optical fibers, etc.

25.2.15 Bus Values

The bus can have one of two complementary logical values: dominant or recessive. During simultaneous transmission of dominant and recessive bits, the resulting bus value will be dominant. For example, in case of a wired-AND implementation of the bus, the dominant level is represented by a logical 0 and the recessive level by a logical 1. Physical states (e.g., electrical voltage, light) that represent the logical levels are not given in this document.

25.2.16 Acknowledgement

All receivers check the consistency of the message being received and acknowledge a consistent message and flag an inconsistent message.

25.3 Channel Overview

The CAN module has a set of buffers, also called channels or mailboxes. Each mailbox is assigned an identifier and can be set to transmit or receive.

It is possible to reconfigure mailboxes dynamically.

When the CAN module receives a message, it checks the mailboxes in order to see if there is a receive mailbox with the same identifier as the message. If such a mailbox is found, the message is stored in it. If several mailboxes are configured with the same identifier, only the smaller channel store the message. If no mailbox is found, the message is discarded.

When the CAN module has to transmit a message, the message length, data and identifier are written to a mailbox set in transmission. If several messages in different mailboxes are waiting

to be transmitted, the mailbox sending the highest priority message (smaller identifier) sends its message first.

If a remote frame is received, the CAN module checks the remote identifier against the mailbox identifier. If a match is found, and if the mailbox is set to automatically reply to the remote frame, the CAN module automatically sends a message with the identifier and data contained in that mailbox.

Transmit buffers must always be configured using channel numbers inferior to all the receive buffer channels.

25.4 Message Transfer

25.4.1 Definition of Transmitter/Receiver

A node originating a message is called the transmitter of that message. The node stays transmitter until the bus is idle or the node loses arbitration.

A node is called receiver of a message if it is not the transmitter of that message and the bus is not idle.

25.4.2 Frame Formats

There are two different formats that differ in the length of the identifier field:

Table 25-1. Identifier Length within Standard and Extended Frames

Frame Format	Identifier Length
Standard frame	11 bits
Extended frame	29 bits

25.4.3 Frame Types

Message transfer is manifested and controlled by four different frame types:

- A data frame carries data from a transmitter to the receivers.
- A remote frame is transmitted by a bus unit to request the transmission of the data frame with the same identifier.
- An error frame is transmitted by any unit on detecting a bus error.
- An overload frame is used to provide for an extra delay between the preceding and the succeeding data or remote frame.

Data frames and remote frames can be used both in standard frame format and extended frame format. They are separated from preceding frames by an interframe space.

25.4.3.1 Data Frame

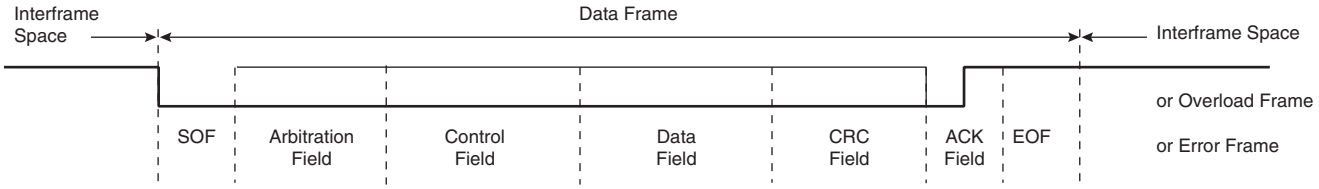
A data frame is composed of seven different bit fields:

- Start Of Frame (SOF)
- Arbitration field
- Control field
- Data field
- CRC field
- ACK field

• End Of Frame (EOF)

The data field can be of length zero.

Figure 25-2. Data Frame



Start Of Frame (Standard and Extended Format)

The Start of Frame (SOF) marks the beginning of data frames and remote frames. It consists of a single dominant bit.

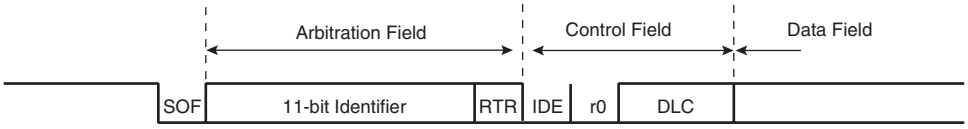
A node is only allowed to start transmission when the bus is idle. All nodes have to synchronize to the leading edge caused by start of frame of the node starting transmission first.

Arbitration Field

The format of the arbitration field is different for standard format and extended format frames.

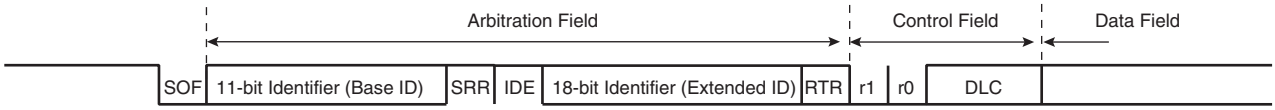
In standard format, the arbitration field consists of the 11-bit identifier and the RTR bit. The identifier bits are denoted ID[10:0].

Figure 25-3. Standard Format



In extended format, the arbitration field consists of the 29-bit identifier, the SRR bit, the IDE bit, and the RTR bit. The identifier bits are denoted ID[28:0]. The base identifier ID[10:0] are sent first, and extended identifier ID[28:11] later.

Figure 25-4. Extended Format



In order to distinguish between standard format and extended format, the reserved bit r1 in previous CAN specifications version 1.0-1.2 now is denoted as IDE bit.

• Identifier - Standard Format

The identifier's length is 11 bits and corresponds to the base ID in extended format. These bits are transmitted in the order from ID10 to ID0. The least significant bit is ID0. The 7 most significant bits ID[10:4] must not be all recessive.

• Identifier - Extended Format

In contrast to the standard format, the extended format consists of 29 bits. The format comprises two sections:

– Base ID: the base ID consists of 11 bits. It is transmitted in the order from ID10 to ID0. It is equivalent to format of the Standard Identifier. The base ID defines the Extended Frame's base priority.

– Extended ID: the Extended ID consists of 18 bits. It is transmitted in the order of ID28 to ID11.

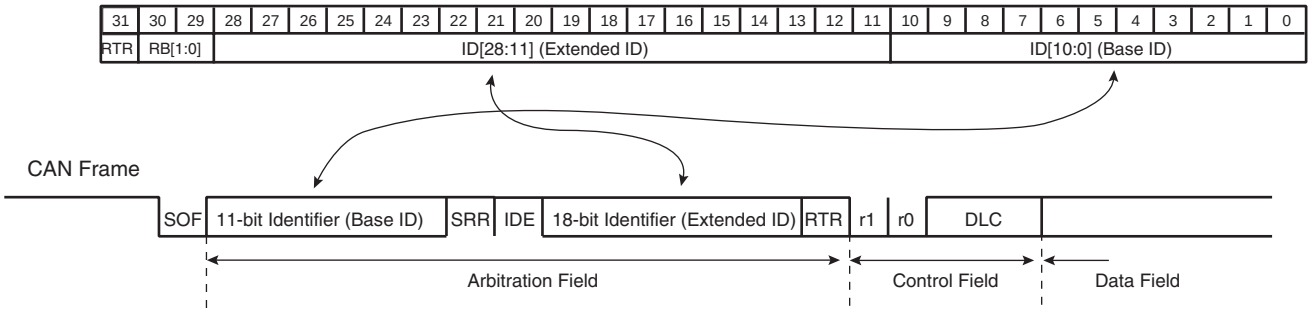
In extended format, the identifier has to be written to the CAN_IRx register in a specific format:

CAN_IRx = ((id & 0x1FFC0000) >> 18) | ((id & 0x3FFFF) << 11)

where (id & 0x1FFC0000) is the 11-bit base ID and (id & 0x3FFFF) is the 18-bit extended ID.

Figure 25-5 shows how the identifier written in the CAN_IRx register is transmitted to the CAN network.

Figure 25-5. Identifier Format from CAN_IRx Register to CAN Frame



- RTR Bit (Standard and Extended Formats)

Table 25-2 shows the RTR bit logical value depending on frame type.

Table 25-2. RTR Bit Logical Value Depending on Frame Type

Frame Type	RTR Bit Logical Value
Data frame	Dominant
Remote frame	Recessive

In an extended frame, the base ID is transmitted first, followed by the IDE bit and the SRR bit. The extended ID is transmitted after the SRR bit.

- SRR Bit (Extended Format)

The SRR bit (Substitute Remote Request) is a recessive bit. It is transmitted in Extended Frames at the position of the RTR bit in standard frames and so substitutes the RTR bit in the standard frame.

Therefore, collisions of a standard frame and an extended frame, the base ID (see Extended Identifier below) of which is the same as the standard frame's identifier, are resolved in such a way that the standard frame prevails the extended frame.

- IDE Bit (Extended Format)

Table 25-3 shows the IDE bit logical value and membership according to frame format type.

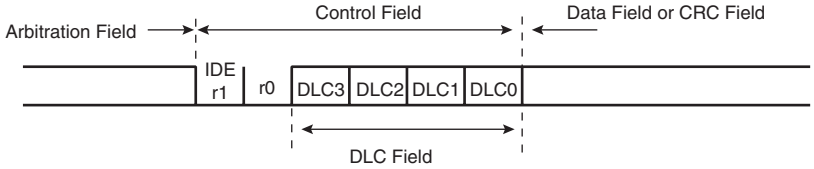
Table 25-3. IDE Bit Logical Value and Membership Depending on Format Type

Frame Format Type	IDE Bit Belongs to	IDE Bit Logical Value
Standard frame format	Control field	Dominant
Extended frame format	Arbitration field	Recessive

Control Field (Standard and Extended Format)

The control field consists of six bits. The format of the control field is different for standard format and extended format. Frames in standard format include the data length code, the IDE bit, which is transmitted dominant, and the reserved bit r0. Frames in extended format include the data length code and two reserved bits, r1 and r0. The reserved bits have to be sent dominant, but receivers accept dominant and recessive bits in all combinations.

Figure 25-6. Control Field



- Data Length Code (Standard and Extended Format)

The number of bytes in the data field is indicated by the data length code. The DLC field is four bits wide and is transmitted within the control field. Coding of the number of data bytes by the DLC field is described in Table 25-4.

Table 25-4. Data Length Code (D = dominant, R = recessive)

Number of Data (Bytes)	Data Length Code			
	DLC3	DLC2	DLC1	DLC0
0	D	D	D	D
1	D	D	D	R
2	D	D	R	D
3	D	D	R	R
4	D	R	D	D
5	D	R	D	R
6	D	R	R	D
7	D	R	R	R
8	R	D	D	D

Data Frame (Standard and Extended Format)

The admissible numbers of data bytes is 0 to 8. Other values may not be used.

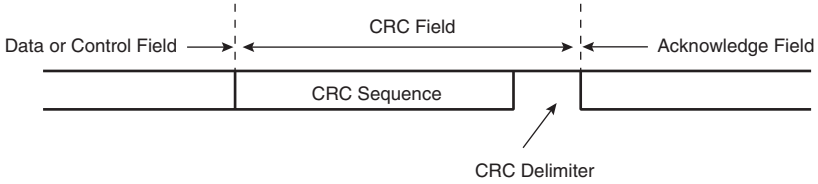
Data Field (Standard and Extended Format)

The data field consists of the data to be transferred within a data frame. It can contain from 0 to 8 bytes, which each contain 8 bits which are transferred MSB first.

CRC Field (Standard and Extended Format)

The CRC (Cyclic Redundancy Check) field contains the CRC sequence followed by a CRC delimiter.

Figure 25-7. CRC Field



- CRC Sequence (Standard and Extended Format)

The frame check sequence is derived from a cyclic redundancy code best suited for frames with bit counts less than 127 bits (BCH code).

In order to carry out the CRC calculation, the polynomial to be divided is defined as the polynomial, the coefficients of which are given by the de-stuffed bit stream consisting of start of frame, arbitration field, control field, data field (if present) and, for the 15 lowest coefficients, by 0. This polynomial is divided (the coefficients are calculated modulo 2) by the generator-polynomial:

$$X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$$

The remainder of this polynomial division is the CRC sequence transmitted over the bus. In order to implement this function, a 15-bit shift register CRC_RG(14:0) can be used. If NXTBIT denotes the next bit of the bit stream, given by the de-stuffed bit sequence from start of frame until the end of the data field, the CRC sequence is calculated as follows:

```
CRC_RG = 0                          // initialize shift register
REPEAT
    CRCNXT = NXTBIT EXOR CRC_RG[14]
    CRC_RG[14:1] = CRC_RG[13:0]     // shift left by
    CRC_RG[0] = 0                   // 1 position
    IF CRCNXT THEN
         CRC_RG[14:0] = CRC_RG[14:0] EXOR 0x4599
    ENDIF
UNTIL (CRC sequence starts or there is an error condition)
```

After the transmission/reception of the last bit of the data field, CRC_RG contains the CRC sequence.

- CRC Delimiter (Standard and Extended Format)

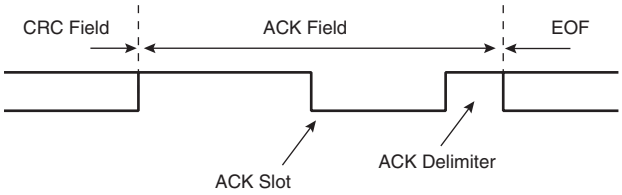
The CRC sequence is followed by the CRC delimiter which consists of a single recessive bit.

ACK Field (Standard and Extended Format)

The ACK (acknowledgement) field is two bits long and contains the ACK slot and the ACK delimiter. In the ACK field, the transmitting node sends two recessive bits.

A receiver that has received a valid message correctly reports this to the transmitter by sending a dominant bit during the ACK slot (it sends ACK).

Figure 25-8. ACK Field



- ACK Slot

All nodes having received the matching CRC sequence report this within the ACK slot by overwriting the recessive bit of the transmitter by a dominant bit.

- ACK Delimiter

The ACK delimiter is the second bit of the ACK field and has to be a recessive bit. As a consequence, the ACK slot is surrounded by two recessive bits (CRC delimiter, ACK delimiter).

End Of Frame (Standard and Extended Format)

Each data frame and remote frame is delimited by seven recessive bits. These bits form the end of frame sequence (EOF).

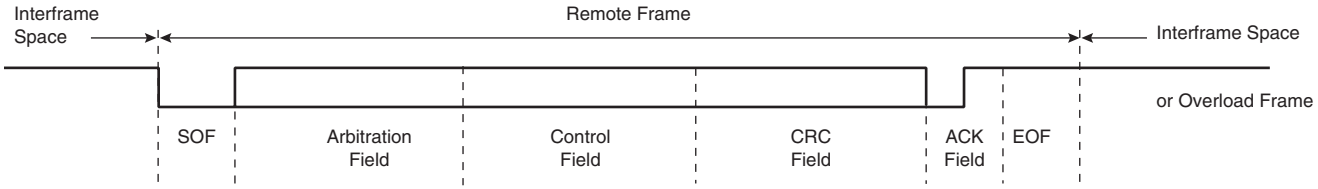
25.4.3.2 Remote Frame

A node acting as a receiver for certain data can initiate the transmission of the respective data by its source node by sending a remote frame. A remote frame exists both in standard format and in extended format. In both cases, it is composed of six different bit fields:

- Start Of Frame (SOF)
- Arbitration field
- Control field
- CRC field
- ACK field
- End Of Frame

Contrary to data frames, the RTR bit of remote frames is recessive. There is no data field, independent of the values of the data length code which may be signed any value within the admissible range from 0 to 8. The value is the data length code of the corresponding data frame.

Figure 25-9. Remote Frame



The polarity of the RTR bit (CAN_IRX register) indicates whether a transmitted frame is a data frame (RTR bit dominant) or a remote frame (RTR bit recessive).

If a mailbox is set to reply automatically to the remote frame (RPLYV bit set to logical 1 in CAN_CRx register), the CAN module automatically sends a message with the identifier and data contained in that mailbox after receiving the remote frame.

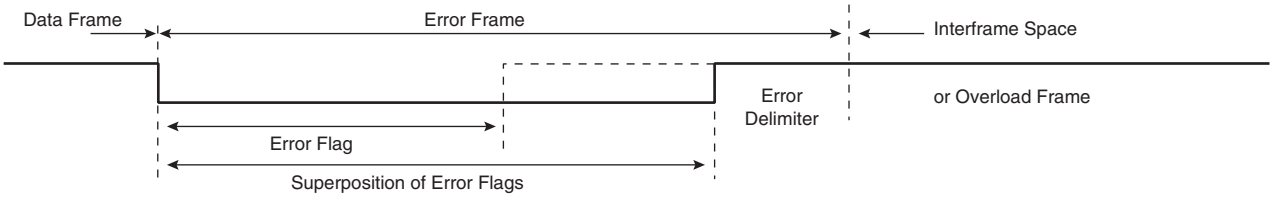
To allow a channel to receive a remote frame, the RTR bit in CAN_IRx register has to be recessive (i.e., at logical 1), or masked (MRTR bit of CAN_MSKx register set to a logical 1).

25.4.3.3 *Error Frame*

The error frame consists of two different fields. The first field is given by the superposition of error flags contributed from different nodes. The second field is the error delimiter.

In order to terminate an error frame correctly, an error passive node may need the bus to be bus idle for at least three bit times if there is a local error at an error passive receiver. Therefore, the bus should not be loaded to 100%.

Figure 25-10. Error Frame



Error Flag

There are two forms of error flag: an active error flag and a passive error flag.

- The active error flag consists of six consecutive dominant bits.
- The passive error flag consists of six consecutive recessive bits unless it is overwritten by dominant bits from other nodes.

An error active node detecting an error condition signals this by transmission of an active error flag. The error flag's form violates the law of bit stuffing (see ”Coding” on page 283) applied to all fields from start of frame to CRC delimiter or destroys the fixed form ACK field or end of frame field. As a consequence, all other nodes detect an error condition and start transmission of an error flag. Thus the sequence of dominant bits that can actually be monitored on the bus results from a superposition of different error flags transmitted by individual nodes. The total length of this sequence varies between a minimum of six and a maximum of twelve bits.

An error passive node detecting an error condition tries to signal this by transmission of a passive error flag. The error passive node waits for six consecutive bits of equal polarity, beginning at the start of the passive error flag. The passive error flag is complete when these six equal bits have been detected.

Error Delimiter

The error delimiter consists of eight recessive bits.

After transmission of an error flag, each node sends recessive bits and monitors the bus until it detects a recessive bit. Afterwards, it starts transmitting seven more recessive bits.

25.4.3.4 *Overload Frame*

The overload frame (this emission can be enabled/disabled in CAN_CR register) contains the two bit fields, overload flag and overload delimiter. There are two kinds of overload conditions that lead to the transmission of an overload flag:

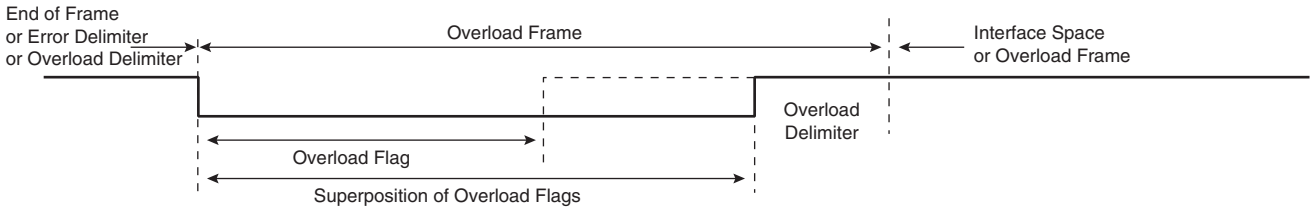
1. The internal conditions of a receiver, requiring a delay of the next data frame or remote frame.
2. Detection of a dominant bit at the first and second bit of intermission.

If a CAN node samples a dominant bit at the eighth bit (the last one) of an error delimiter or overload delimiter, it starts transmitting an overload frame (not an error frame). The error counters are incremented.

The start of an overload frame due to the first overload condition is only allowed to be started at the first bit time of an expected intermission, whereas overload frames due to the second overload condition and condition 3 start one bit after detecting the dominant bit.

At most two overload frames may be generated to delay the next data or remote frame.

Figure 25-11. Overload Frame



Overload Flag

The overload flag consists of six dominant bits. The overall form corresponds to that of the active error flag.

The overload flag's form destroys the fixed form of the intermission field. As a consequence, all other nodes also detect an overload condition and start transmission of an overload flag. If there is a dominant bit detected during the 3rd bit of intermission, then it interpret this bit as start of frame.

Note: Controllers based on the CAN Specification version 1.0 and 1.1 have another interpretation of the 3rd bit of intermission: if a dominant bit was detected locally at some node, the other nodes do not interpret the overload flag correctly, but interpret the first of these six dominant bits as start of frame; the sixth dominant bit violates the rule of bit stuffing causing an error condition.

Overload Delimiter

The overload delimiter consists of eight recessive bits.

The overload delimiter is of the same form as the error delimiter. After transmission of an overload flag, the node monitors the bus until it detects a transition from a dominant to a recessive bit. At this time, every bus node has finished sending its overload flag and all nodes start transmission of seven more recessive bits simultaneously.

25.4.3.5 Interframe Spacing

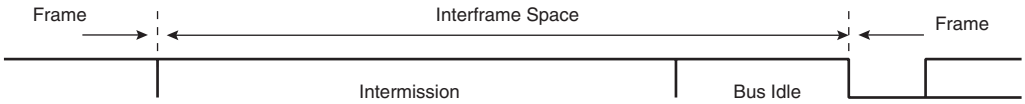
Data frames and remote frames are separated from preceding frames whatever type they are (data frame, remote frame, error frame, overload frame) by a bit field called interframe space. In contrast, overload frames and error frames are not preceded by an interframe space and multiple overload frames are not separated by an interframe space.

Interframe Space

The interframe space contains the bit field intermission and bus idle and, for error passive nodes that have been transmitter of the previous message, suspend transmission.

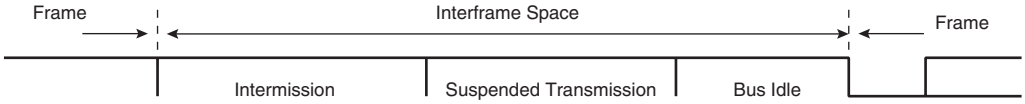
For nodes that are not error passive or have been receiver of the previous message, see Figure 25-12.

Figure 25-12. Interframe Space for Receiver



For error passive nodes which have been transmitter of the previous message, see Figure 25-13.

Figure 25-13. Interframe Space for Transmitter



Intermission

Intermission consists of three recessive bits.

During intermission, the only action to be taken is signalling an overload condition. No node is allowed to actively start a transmission of a data frame or remote frame.

Note: If a CAN node has a message waiting for transmission and it samples a dominant bit at the third bit of intermission, it will interpret this as a start of frame bit, and, with the next bit, start transmitting its message with the first bit of its identifier without first transmitting a start of frame bit and without becoming receiver.

Bus Idle

The period of bus idle may be of arbitrary length. The bus is recognized to be free and any node having something to transmit can access the bus. A message, which is pending for transmission during the transmission of another message, is started in the first bit following intermission.

The detection of a dominant bit on the bus is interpreted as start of frame.

Suspend Transmission

After an error passive node has transmitted a message, it sends eight recessive bits following intermission, before starting to transmit a further message or recognizing the bus to be idle. If a transmission (caused by another node) starts in the meantime, the node becomes receiver of this message.

25.4.4 Conformance with Frame Formats

The Standard Format is equivalent to the Data/Remote Frame Format as described in the CAN Specification 1.2. In contrast, the Extended Format is a new feature of the CAN protocol. In order to allow the design of relatively simple controllers, the implementation of the Extended Format to its full extent is not required (e.g., send messages or accept data from messages in Extended Format), whereas the Standard Format must be supported without restriction.

New controllers are considered to be in conformance with this CAN Specification if they have at least the following properties with respect to the frame formats defined in ”Definition of Transmitter/Receiver” on page 274 and ”Frame Types” on page 274.

- Every new controller supports the standard format.
- Every new controller can receive messages of the extended format. This requires that extended frames are not destroyed just because of their format. However, it is not required that the extended format must be supported by new controllers.

25.5 Message Filtering

Message filtering is based upon the whole identifier. Mask registers that allow any identifier bit to be set “don't care” for message filtering may be used to select groups of identifiers to be mapped into the attached received buffers.

If mask registers are implemented, every bit of the mask registers must be programmable, i.e., they can be enabled or disabled for message filtering. The length of the mask register can comprise the whole identifier or only part of it.

25.6 Message Validation

The point in time at which a message is taken to be valid is different for the transmitter and the receiver of the message.

- Transmitter

The message is valid for the transmitter if there is no error until the end of end of frame. If a message is corrupted, retransmission will follow automatically and according to prioritization. In order to be able to compete for bus access with other messages, retransmission has to start as soon as the bus is idle.

- Receivers

The message is valid for the receivers if there is no error until the first-to-last bit of end of frame. The value of the last bit of EOF is treated as “don't care”, a dominant value does not lead to a FORM error (see ”Error Detection” on page 283).

25.7 Coding

In bit stream coding, the frame segments start of frame, arbitration field, control field, data field and CRC sequence are coded by bit stuffing. Whenever a transmitter detects five consecutive bits of identical value in the bit stream to be transmitted, it automatically inserts a complementary bit in the currently transmitted bit stream.

The remaining bit fields of the data frame or remote frame (CRC delimiter, ACK field and end of frame) are of fixed form and not stuffed. The error frame and the overload frame are of fixed form as well and not coded via bit stuffing.

The bit stream in a message is coded according to the Non-Return-to-Zero (NRZ) method. This means that during the total bit time the generated bit level is either dominant or recessive.

25.8 Error Handling

25.8.1 Error Detection

There are 5 different error types (not mutually exclusive):

- Bit error (BUS bit in CAN_SRX): A unit that is sending a bit on the bus also monitors the bus. A bit error has to be detected when the bit value that is monitored is different from the bit value that is sent. An exception is the sending of a recessive bit during the stuffed bit stream of the arbitration field or during the ACK slot. In this case, no bit error occurs when

a dominant bit is monitored. A transmitter sending a passive error flag and detecting a dominant bit does not interpret this as a bit error.

- Stuff error (STUFF bit in CAN_SRX): A stuff error is detected at the bit time of the 6th consecutive equal bit level in a message field that should be coded by the method of bit stuffing.
- CRC error (CRC bit in CAN_SRX): The CRC sequence consists of the result of the CRC calculation by the transmitter. The receivers calculate the CRC in the same way as the transmitter. A CRC error has to be detected if the calculated result is not the same as that received in the CRC sequence.
- Form error (FRAME bit in CAN_SRX): A form error has to be detected when a fixed-form bit field contains one or more illegal bits. (Note that for a receiver a dominant bit during the last bit of EOF is not treated as a form error).
- Acknowledgement error (ACK bit in CAN_SRX): An acknowledgement error is detected by a transmitter whenever it does not monitor a dominant bit during ACK slot.

25.8.2 Error Signalling

A node detecting an error condition signals this by transmitting an error flag. For an error active node, it is an active error flag; for an error passive node, it is a passive error flag.

Whenever a bit error, a stuff error, a form error or an acknowledgement error is detected by any node, transmission of an error flag is started at the respective node at the next bit.

Whenever a CRC error is detected, transmission of an error flag starts at the bit following the ACK delimiter, unless an error flag for another error condition has already been started.

25.9 Fault Confinement

Regarding fault confinement, a unit may be in one of three states:

- error active
- error passive
- bus off

An error active unit can normally take part in bus communication and sends an active error flag when an error has been detected.

An error passive unit must not send an active error flag. It takes part in bus communication, but when an error has been detected, only a passive error flag is sent. After a transmission, an error passive unit will wait before initiating a further transmission (see ”Suspend Transmission” on page 282).

A bus off unit is not allowed to have any influence on the bus. (e.g., output drivers switched off).

For fault confinement, two counts are implemented in every bus unit:

- transmit error count
- receive error count

These counts are modified according to the following rules (note that more than one rule may apply during a given message transfer):

- When a receiver detects an error, the receive error count will be increased by 1, except when the detected error was a bit error during the sending of an active error flag or an overload flag.

- When a receiver detects a dominant bit as the first bit after sending an error flag, the receive error count will be increased by 8.
- When a transmitter sends an error flag, the transmit error count is increased by 8 except:
  - if the transmitter is error passive and detects an acknowledgement error because of not detecting a dominant ACK. It does not detect a dominant bit while sending its passive error flag.
  - if the transmitter sends an error flag because a stuff error occurred during arbitration, and should never been recessive, and has been sent as recessive but monitored as dominant

  In case of one of these exceptions, the transmit error count is not changed.
- If a transmitter detects a bit error while sending an active error flag or an overload flag, the transmit error count is increased by 8.
- If a receiver detects a bit error while sending an active error flag or an overload flag, the receive error count is increased by 8.
- Any node tolerates up to 7 consecutive dominant bits after sending an active error flag, passive error flag or overload flag. After detecting the 14th consecutive dominant bit (in case of an active error flag or an overload flag) or after detecting the 8th consecutive dominant bit following a passive error flag, and after each sequence of additional eight consecutive dominant bits, every transmitter increases its transmit error count by 8 and every receiver increases its receive error count by 8.
- After the successful transmission of a message (getting ACK and no error until end of frame is finished), the transmit error count is decreased by 1 unless it was already 0.
- After the successful reception of a message (reception without error up to the ACK slot and the successful sending of the ACK bit), the receive error count is decreased by 1, if it was between 1 and 127. If the receive error count was 0, it stays 0, and if it was greater than 127, then it will be set to a value between 119 and 127.
- A node is error passive when the transmit error count equals or exceeds 128, or when the receive error count equals or exceeds 128. An error condition letting a node become error passive causes the node to send an active error flag.
- A node is bus off when the transmit error count is greater than or equal to 256.
- An error passive node becomes error active again when both the transmit error count and the receive error count are less than or equal to 127.
- An node that is bus off is permitted to become error active (no longer bus off) with its error counters both set to 0 after 128 occurrences of 11 consecutive recessive bits have been monitored on the bus.

Note: An error count value greater than 96 indicates a heavily disturbed bus. It may be useful to provide means to test for this condition.

Note: Start-up/Wake-up: If during system start-up only one node is online, and if this node transmits a message, it gets no acknowledgement, detects an error and repeats the message. It can become error passive but not bus off due to this reason.

25.10 Oscillator Tolerance

A maximum oscillator tolerance of 1.58% is given and therefore a ceramic resonator can be used at a bus speed of up to 125 Kbits/s as a rule of thumb.

For the full bus speed range of the CAN protocol, a quartz oscillator is required.

The chip of the CAN network with the highest requirement for its oscillator accuracy determines the oscillator accuracy which is required from all the other nodes.

Note: CAN controllers compliant with this CAN Specification and controllers compliant with the previous versions 1.0 and 1.1, used in one and the same network, must all be equipped with a quartz oscillator. Thus ceramic resonators can only be used in a network with all the nodes of the network according to the CAN Protocol Specification versions 1.2 or later.

25.11 Bit Timing Requirements

25.11.1 Nominal Bit Rate

The nominal bit rate is the number of bits per second transmitted in the absence of resynchronization by an ideal transmitter.

25.11.2 Nominal Bit Time

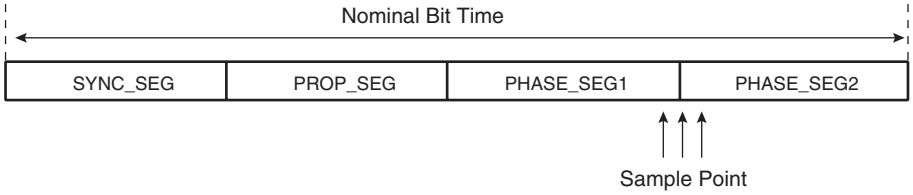
The nominal bit time of the network is uniform throughout the network and is given by:

$$\text{Nominal Bit Time} = 1 / \text{Nominal Bit Rate}$$

The nominal bit time can be thought of as being divided into separate non-overlapping time segments. These segments are:

- synchronization segment (SYNC_SEG)
- propagation time segment (PROP_SEG)
- phase buffer segment 1 (PHASE_SEG1)
- phase buffer segment 2 (PHASE_SEG2)

Figure 25-14. Partition of the Bit Time



The duration of each segment is set in the mode register and is expressed as a multiple of time quantum (t_{CAN}).

25.11.3 Time Quantum

Each segment (SYNC_SEG, PROP_SEG, PHASE_SEG1 and PHASE_SEG2) is an integer multiple of a unit of time called a Time Quantum (t_{CAN}). The duration of a Time Quantum is equal to the period of the CAN system clock (t_{CAN}), which is derived from the microcontroller system clock (CORECLK) by way of a programmable prescalar, called the Baud Rate Prescalar (in the CAN_MR register).

The formula relating t_{CAN}, CORECLK and BD[5:0] is the following:

$$t_{CAN} = \frac{(BD[5:0] + 1)}{CORECLK}$$

Setting BD[5:0] to ‘0’ is forbidden in order to ensure a proper resynchronization of the CAN_RX input.

25.11.4 Synchronization Segment

This part of the bit time is used to synchronize the various nodes on the bus. An edge is expected to lie within this segment.

The duration of the synchronization segment, SYNC_SEG, is not programmable and is fixed at one time quantum.

25.11.5 Propagation Segment

This part of the bit time is used to compensate for the physical delay times within the network. It is twice the sum of the signal's propagation time on the bus line, the input comparator delay, and the output driver delay.

The duration of the propagation segment, PROP_SEG, may be between 1 and 8 time quanta. It is programmable using the PROP bits in the CAN_MR, and is equal to:

$$t_{PRS} = t_{CAN} \times (PROP[2:0] + 1)$$

25.11.6 Phase Buffer Segments

The phase buffer segments are used to compensate for edge phase errors. The segments can be lengthened or shortened by resynchronization.

The duration of the segment PHASE_SEG1 may be between 1 and 8 time quanta.

The duration of segment PHASE_SEG2 is the maximum of PHASE_SEG1 and the information processing time (See ”Information Processing Time (IPT)” on page 287).

PHASE_SEG1 and PHASE_SEG2 are programmable using the PHSEG1[2:0] and PHSEG1[2:0] bits in the MR, respectively. They respect the following equations:

$$t_{PHS1} = t_{CAN} \times (PHSEG1[2:0] + 1)$$

$$t_{PHS2} = t_{CAN} \times (PHSEG2[2:0] + 1)$$

Setting the duration of segment PHASE_SEG2 to 1 time quanta is forbidden (PHSEG2[2:0] field in CAN_MR must be set to 1 at minimum).

25.11.7 Sample Point

The sample point is the point of time at which the bus level is read and interpreted as the value of that respective bit. Its location is at the end of PHASE_SEG1. Two methods can be used: the incoming stream is sampled once at a sample point, or sampled 3 times with a period of half a CAN clock period, centered at one sample point.

Configuring the CAN controller to sample once is forbidden in order to ensure a proper resynchronization of the CAN_RX input.

25.11.8 Information Processing Time (IPT)

The information processing time is the time segment starting with the sample point reserved for calculation of the subsequent bit level.

In other words, the time after the sample point that is needed to calculate the next bit to be sent (e.g., data bit, CRC bit, stuff bit, error flag, or idle) is called the Information Processing Time (IPT).

The CAN module has zero delay IPT.

25.11.9 Length of Time Segments

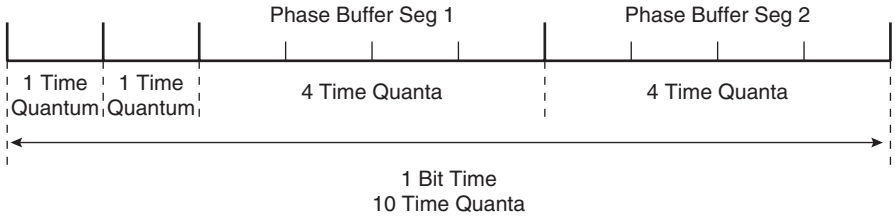
- SYNC_SEG is 1 time quantum long.
- PROP_SEG is programmable to be between 1 and 8 time quanta long.
- PHASE_SEG1 is programmable to be between 1 and 8 time quanta long.
- PHASE_SEG2 is the maximum of PHASE_SEG1 and the information processing time.
- The information processing time is 0 time quanta long.

The total number of time quanta in a bit time is programmable at least from 8 to 25.

Note: It is often intended that control units do not make use of different oscillators for the local CPU and its communication device. Therefore, the oscillator frequency of a CAN device tends to be that of the local CPU and is determined by the requirements of the control unit. In order to derive the desired bit rate, programmability of the bit timing is necessary. In case of CAN implementations that are designed for use without a local CPU, the bit timing cannot be programmable. On the other hand, these devices allow selection of an external oscillator in such a way that the device is adjusted to the appropriate bit rate so that the programmability is dispensable for such components.

The position of the sample point, however, should be selected in common for all nodes. Therefore, the bit timing of CAN devices without local CPU should be compatible with the definition of the bit time in Figure 25-15.

Figure 25-15. Example of Nominal Bit Time



25.11.10 Hard Synchronization

After a hard synchronization, the internal bit time is restarted with SYNC_SEG. Thus hard synchronization forces the edge that caused the hard synchronization to lie within the synchronization segment of the restarted bit time.

25.11.11 Resynchronization Jump Width

As a result of resynchronization, PHASE_SEG1 may be lengthened or PHASE_SEG2 may be shortened. The amount of lengthening or shortening of the phase buffer segments has an upper bound given by the resynchronization jump width. The resynchronization jump width is programmable between 1 and min(4, PHASE_SEG1).

Clocking information may be derived from transitions from one bit value to the other. The property that only a fixed maximum number of successive bits have the same value provides the possibility of resynchronizing a bus unit to the bit stream during a frame. The maximum length between two transitions that can be used for resynchronization is 29 bit times.

Setting the resynchronization jump width to 1 time quantum (SJW[1:0] field in CAN_MR register set to '0') is allowed only if the duration of segment PHASE_SEG2 is lower or equal to 4 time quanta (PHSEG2[2:0] field in CAN_MR lower or equal to 3). If the duration of segment PHASE_SEG2 is greater or equal to 5 time quanta (PHSEG2[2:0] field in CAN_MR greater or equal to 4), the resynchronization jump width must be configured at minimum to 2 time quanta (SJW[1:0] field in CAN_MR register set to 1 at minimum).

25.11.12 Phase Error of an Edge

The phase error of an edge is given by the position of the edge relative to SYNC_SEG, measured in time quanta. The sign of phase error is defined as follows:

- e = 0 if the edge lies within SYNC_SEG.
- e > 0 if the edge lies before the sample point.
- e < 0 if the edge lies after the sample point of the previous bit.

25.11.13 Resynchronization

The effect of a resynchronization is the same as that of a hard synchronization when the magnitude of the phase error of the edge causing the resynchronization is less than or equal to the programmed value of the resynchronization jump width. When the magnitude of the phase error is larger than the resynchronization jump width,

- and if the phase error is positive, then PHASE_SEG1 is lengthened by an amount equal to the resynchronization jump width.
- and if the phase error is negative, then PHASE_SEG2 is shortened by an amount equal to the resynchronization jump width.

25.11.14 Synchronization Rules

Hard synchronization and resynchronization are the two forms of synchronization. They obey the following rules:

1. Only one synchronization within one bit time is allowed.
2. An edge is used for synchronization only if the value detected at the previous sample point (previous read bus value) differs from the bus value immediately after the edge.
3. Hard synchronization is performed whenever there is a recessive-to-dominant edge during bus idle.
4. All other recessive-to-dominant edges fulfilling rules 1 and 2 are used for resynchronization with the exception that a node transmitting a dominant bit does not perform a resynchronization as a result of a recessive-to-dominant edge with a positive phase error if only recessive-to-dominant edges are used for resynchronization.

25.12 Reception Mode

In reception, channels can be configured in two different modes:

- Normal mode (OVERWRITE bit reset to 0 in CAN_CRx): the channel is disabled after a successful reception.
- Overwrite mode (OVERWRITE bit set to 1 in CAN_CRx): the channel is still enabled after a successful reception.

25.13 Time Stamp

A 32-bit stamp dates all messages sent/received (depending on the producer/consumer bit).

The 32-bit register forming the second counter in the WT module is provided to the CAN module. After each transmission or reception of a CAN frame, the value of the current second counter is automatically written in the corresponding CAN channel CAN_STPx register.

25.14 Power Management

The CAN is provided with a power management block allowing optimization of power consumption (see ”Power Management Block” on page 23).

25.15 Example of Use

25.15.1 Message Transmission

25.15.1.1 Processing Transmission Request Delay

When a CAN channel configured in transmission is enabled, the transmission of the frame does not start immediately but after a short delay. This delay is due to channel scanning. When the bus is idle, the CAN state machine continually scans all channels, looking for transmit channels. During a scanning cycle - starting from channel 0 and ending at channel 15 or 31 - if several channels are configured in transmission, the CAN state machine memorizes the channel with the lowest identifier, and at the end of the scanning cycle it starts the transmission of the highest priority frame.

Scanning delay of CAN channels depends on the channel state: a channel enabled and configured in transmission is scanned in three system clock periods, otherwise it takes 2 system clock periods. Then, scanning all the channels takes at maximum 49 system clock periods for a 16-channel CAN and 97 system clock periods for a 32-channel CAN.

As the CPU may enable a transmit channel at any moment during the scanning cycle, the delay from the moment the channel is enabled by the CPU to the moment the frame starts to be transmitted may be unsettled and can take at maximum two times the delay for scanning all channels. Moreover, as the scan of the channels is also asynchronous with the bit time, between 0 and 1 bit times should be added to this delay.

25.15.1.2 Warning on Frame Transmission Modification

Once a transmission has been requested, users should not modify the frame to transmit (CAN_IRX,CAN_DRAX, CAN_DRBX and CAN_CRX registers) until the transmission is completed (flag TXOK set to ‘1’ in the CAN_SRx register). In order to modify a frame to transmit, users should cancel the transmission, change the frame and then request a new transmission. Find below how to cancel a transmission.

25.15.1.3 Cancelling a Pending Transmission

A pending transmission may be cancelled but users must respect the following steps:

- Clear the CHANEN flag in the CAN_CRx register. Note that other bits of this register should not be modified (a read-modify is mandatory).
- Wait for a minimum delay before setting a new reception or transmission in this channel. This delay depends on the identifier length configured in the pending transmission to cancel. The delay is:
  - for a standard frame (11-bit length identifier):
    97 system clock periods + 106 CAN bit time
  - for an extended frame (29-bit length identifier):
    97 system clock periods + 131 CAN bit time
- Afterwards, the frame related registers (CAN_IRX, CAN_DRAX, CAN_DRBX and CAN_CRX registers) can be modified.

This assures that if the frame transmission process has started before the cancel has been requested, the frame transmitted has the correct identifier, data and control.

25.15.2 Frequency Limitation

In order to ensure proper CAN operation, the CAN module should be clocked with a minimum frequency. This minimum frequency can be computed with the following formula:

$$\text{Minimum Frequency} = \frac{1 + (3 \times \text{Number of channel enable in transmission}) + (2 \times \text{Reminder channels})}{3 \times \text{CAN Bit Time}}$$

For example, for a application using only one channel in transmission, a 32-channel CAN and a CAN baud rate of 1 MHz, the minimum frequency is 22 MHz
(22 MHz = (1 + 3*1 + 2*31)/3µs).

25.16 Controller Area Network (CAN) Memory Map

Base Address: 0xFFFD4000

Table 25-5. CAN Memory Map

Address	Register	Name	Access	Reset State
0x000 – 0x04C	Reserved	–	–	–
0x050	Enable Clock Register	CAN_ECR	Write -only	–
0x054	Disable Clock Register	CAN_DCR	Write-only	–
0x058	Power Management Status Register	CAN_PMSR	Read-only	0x00000000
0x05C	Reserved	–	–	–
0x060	Control Register	CAN_CR	Write-only	–
0x064	Mode Register	CAN_MR	Read/Write	0x00000000
0x068	Reserved	–	–	–
0x06C	Clear Status Register	CAN_CSR	Write-only	–
0x070	Status Register	CAN_SR	Read-only	0x00000002
0x074	Interrupt Enable Register	CAN_IER	Write-only	–
0x078	Interrupt Disable Register	CAN_IDR	Write-only	–
0x07C	Interrupt Mask Register	CAN_IMR	Read-only	0x00000000
0x080	Clear Interrupt Source Status Register	CAN_CISR	Write-only	–
0x084	Interrupt Source Status Register	CAN_ISSR	Read-only	0x00000000
0x088	Source Interrupt Enable Register	CAN_SIER	Write-only	–
0x08C	Source Interrupt Disable Register	CAN_SIDR	Write-only	–
0x090	Source Interrupt Mask Register	CAN_SIMR	Read-only	0x00000000
0x094 – 0x0FC	Reserved	–	–	–
0x100	Channel 0 Data Register A	CAN_DRA0	Read/Write	Note (1)
0x104	Channel 0 Data Register B	CAN_DRB0	Read/Write	Note (1)
0x108	Channel 0 Mask Register	CAN_MSK0	Read/Write	Note (1)
0x10C	Channel 0 Identifier Register	CAN_IR0	Read/Write	Note (1)
0x110	Channel 0 Control Register	CAN_CR0	Read/Write	Note (1)
0x114	Channel 0 Stamp Register	CAN_STP0	Read-only	Note (1)
0x118	Channel 0 Clear Status Register	CAN_CSR0	Write-only	–
0x11C	Channel 0 Status Register	CAN_SR0	Read-only	Note (1)
0x120	Channel 0 Interrupt Enable Register	CAN_IER0	Write-only	–
0x124	Channel 0 Interrupt Disable Register	CAN_IDR0	Write-only	–
0x128	Channel 0 Interrupt Mask Register	CAN_IMR0	Read-only	0x00000000
0x12C – 0x13C	Reserved	–	–	–
0x140	Channel 1 Data Register A	CAN_DRA1	Read/Write	Note (1)
0x144	Channel 1 Data Register B	CAN_DRB1	Read/Write	Note (1)

Table 25-5. CAN Memory Map (Continued)

Address	Register	Name	Access	Reset State
0x148	Channel 1 Mask Register	CAN_MSK1	Read/Write	Note (1)
–	–	–	–	–
0x3E0	Channel 15 Interrupt Enable Register	CAN_IER15	Write-only	–
0x3E4	Channel 15 Interrupt Disable Register	CAN_IDR15	Write-only	–
0x3E8	Channel 15 Interrupt Mask Register	CAN_IMR15	Read-only	Note (1)

Note: 1. The value of this register is undefined during CAN initialization phase. After the initialization phase, the value is 0x00000000.

25.16.1 CAN Enable Clock Register

Name: CAN_ECR
Access: Write-only
Offset: 0x050

25.16.2 CAN Disable Clock Register

Name: CAN_DCR
Access: Write-only
Offset: 0x054

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
–	–	–	–	–	–	CAN	–

- **CAN: CAN Clock Status**

0: CAN clock disabled.

1: CAN clock enabled.

Note: The CAN_PMSR register is not reset by software reset.

25.16.3 CAN Power Management Status Register

Name: CAN_PMSR
Access: Read-only
Offset: 0x058

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
–	–	–	–	–	–	CAN	–

- **CAN: CAN Clock Status**

0: CAN clock disabled.

1: CAN clock enabled.

Note: The CAN_PMSR register is not reset by software reset.

25.16.4 CAN Control Register

Name: CAN_CR
Access: Write-only
Offset: 0x060

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
–	–	–	ABDIS	ABEN	CANDIS	CANEN	SWRST

- **SWRST: CAN Software Reset**

0: No effect.

1: Resets the CAN.

A software reset triggered hardware reset of the CAN is performed. It resets all the registers (except the CAN_PMSR).

- **CANEN: CAN Enable**

0: No effect.

1: Enables the CAN.

This enables the CAN to transfer and receive data.

- **CANDIS: CAN Disable**

0: No effect.

1: Disables the CAN.

No data is received or transmitted.

In case a transfer is in progress, the transfer is finished before the CAN is disabled.

In case both CANEN and CANDIS are equal to one when the control register is written, the CAN is disabled.

- **ABEN: Abort Request Activate**

0: No effect.

1: Activates the CAN abort request.

- **ABDIS: Abort Request Deactivate**

0: No effect.

1: Deactivates the CAN abort request.

In case both ABEN and ABDIS are equal to one when the control register is written, the CAN abort request is deactivated.

25.16.5 CAN Mode Register

Name: CAN_MR
Access: Read/Write
Offset: 0x064

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	PHSEG2			–	PHSEG1		
15	14	13	12	11	10	9	8
–	SMP	SJW		–	PROP		
7	6	5	4	3	2	1	0
–	–	BD					

- **BD[5:0]: Time Quantum Period**

These bits are used to determine the time quantum t_{CAN}:

$$t_{CAN} = \frac{(BD[5:0] + 1)}{CORECLK}$$

Note: Setting BD to 0 is forbidden.

- **PROP[2:0]: Propagation Segment Value**

This data reflects the physical delay within the network, including the signal propagation time and the chip internal delay.

$$t_{PRS} = t_{CAN} \times (PROP[2:0] + 1)$$

- **SJW[1:0]: Synchronization Jump Width**

The CAN controller resynchronizes on each edge of the transmission. The SJW value defines the maximum number of CAN clock cycles a bit period may be shortened or lengthened.

$$t_{SWJ} = t_{CAN} \times (SWJ[1:0] + 1)$$

Note: If the duration of segment PHASE_SEG2 is greater or equal to 5 time quanta (PHSEG2[2:0] field in CAN_MR greater or equal to 4), the resynchronization jump width must be configured at minimum to 2 time quanta (SJW[1:0] field in CAN_MR register set to 1 at minimum).

- **SMP: Sampling Mode**

0: The incoming stream is sampled once at sample point.

1: The incoming stream is sampled 3 times with a period of half a CAN clock period, centered at sample point.

Note: Setting SMP to ‘1’ is recommended. Setting SMP to ‘0’ can cause generation of error frames occasionally while receiving a non-corrupted frame.

- **PHSEG1[2:0]: Phase Segment 1 Value**

This data is used to compensate edge phase errors. This segment can be shortened or lengthened by SJW.

$$t_{PHS1} = t_{CAN} \times (PHSEG1[2:0] + 1)$$

- **PHSEG2[2:0]: Phase Segment 2 Value**

This data is used to compensate edge phase errors. This segment can be shortened or lengthened by SJW.

$$t_{PHS2} = t_{CAN} \times (PHSEG2[2:0] + 1)$$

Note: Setting the duration of segment PHASE_SEG2 to 1 time quanta is forbidden (PHSEG2[2:0] field in CAN_MR must be set to 1 at minimum).

25.16.6 CAN Clear Status Register

Name: CAN_CSR
Access: Write-only
Offset: 0x06C

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
–	–	–	–	–	ENDINIT	–	–

- **ENDINIT: Clear End of CAN Initialization**

0: No effect.

1: Clears end of CAN initialization interrupt.

25.16.7 CAN Status Register

Name: CAN_SR
Access: Read-only
Offset: 0x070

31	30	29	28	27	26	25	24
TEC							
23	22	21	20	19	18	17	16
REC							
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ISS	–	ABRQ	BUSOFF	ERPAS	ENDINIT	CANINIT	CANENA

- **CANENA: CAN Enabled**

0: CAN is disabled.

1: CAN is enabled.

No interrupt is generated on CAN enable or disable.

- **CANINIT: CAN Initialized**

0: CAN is initialized.

1: CAN is in initialization phase.

During initialization phase, channel registers cannot be written. This bit does not generate an interrupt. After the initialization phase, this bit returns to 0 and the ENDINIT bit is set to a logical 1.

- **ENDINIT: End of CAN Initialization**

0: No end of CAN initialization.

1: End of CAN initialization phase.

- **ERPAS: Error Passive**

0: No transition in error passive mode.

1: CAN enters in error passive mode.

- **BUSOFF: Bus Off**

0: No transition in bus off mode.

1: CAN enters in bus off mode.

- **ABRQ: CAN Abort Request**

0: No CAN abort requested.

1: All the enabled channels are disabled. If this bit is set during communication, the transmission will end.

No interrupt is generated on CAN abort request activation or deactivation.

- **ISS: Interrupt Source Status**

0: No interrupt in any channel.

1: At least one interrupt occurred in a channel (read CAN_ISSR for more information).

- **REC[7:0]: Reception Error Counter**

Value of the reception error counter.

- **TEC[7:0]: Transmit Error Counter**

Value of the transmit error counter.

25.16.8 CAN Interrupt Enable Register

Name: CAN_IER
Access: Write-only
Offset: 0x074

25.16.9 CAN Interrupt Disable Register

Name: CAN_IDR
Access: Write-only
Offset: 0x078

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
–	–	–	BUSOFF	ERPAS	ENDINIT	–	–

- **ENDINIT: End of CAN Initialization Mask**

0: End of CAN initialization interrupt is disabled.

1: End of CAN initialization interrupt is enabled.

- **ERPAS: Error Passive Mask**

0: Error passive interrupt is disabled.

1: Error passive interrupt is enabled.

- **BUSOFF: Bus Off Mask**

0: Bus off interrupt is disabled.

1: Bus off interrupt is enabled.

25.16.10 CAN Interrupt Mask Register

Name: CAN_IMR
Access: Read-only
Offset: 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
–	–	–	BUSOFF	ERPAS	ENDINIT	–	–

- **ENDINIT: End of CAN Initialization Mask**

0: End of CAN initialization interrupt is disabled.

1: End of CAN initialization interrupt is enabled.

- **ERPAS: Error Passive Mask**

0: Error passive interrupt is disabled.

1: Error passive interrupt is enabled.

- **BUSOFF: Bus Off Mask**

0: Bus off interrupt is disabled.

1: Bus off interrupt is enabled.

25.16.11 CAN Clear Interrupt Source Status Register

Name: CAN_CISR
Access: Write-only
Offset: 0x080

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

- **CHX: Channel X Interrupt Clear**

0: No effect.

1: Clears interrupt for Channel X.

25.16.12 CAN Interrupt Source Status Register

Name: CAN_ISSR
Access: Write-only
Offset: 0x084

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

- **CHX: Channel X Interrupt**

0: No interrupt occurred on Channel X.

1: An interrupt occurred on Channel X.

25.16.13 CAN Source Interrupt Enable Register

Name: CAN_SIER
Access: Write-only
Offset: 0x088

25.16.14 CAN Source Interrupt Disable Register

Name: CAN_SIDR
Access: Write-only
Offset: 0x08C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

- **CHX: Channel X Interrupt Mask**

0: Channel X interrupt is disabled.

1: Channel X interrupt is enabled.

25.16.15 CAN Source Interrupt Mask Register

Name: CAN_SIMR
Access: Read-only
Offset: 0x090

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

- **CHX: Channel X Interrupt Mask**

0: Channel X interrupt is disabled.

1: Channel X interrupt is enabled.

25.16.16 CAN Data A Register Channel x

Name: CAN_DRAx
Access: Read/Write
Offset: 0xXX0

31	30	29	28	27	26	25	24
DATA3							
23	22	21	20	19	18	17	16
DATA2							
15	14	13	12	11	10	9	8
DATA1							
7	6	5	4	3	2	1	0
DATA0							

- **DATAy [y = 3..0]: Data y of Channel x**

Data number y of Channel x.

25.16.17 CAN Data B Register Channel x

Name: CAN_DRBx
Access: Read/Write
Offset: 0xXX4

31	30	29	28	27	26	25	24
DATA7							
23	22	21	20	19	18	17	16
DATA6							
15	14	13	12	11	10	9	8
DATA5							
7	6	5	4	3	2	1	0
DATA4							

- **DATAy [y = 7..4]: Data y of Channel x**

Data number y of Channel x.

25.16.18 CAN Channel x Mask Register

Name: CAN_MSKx
Access: Read/Write
Offset: 0xXX8

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

- **MASK[28:0]: Identifier Mask of Channel X**

29-bit mask for identifier.

Setting a MASK bit to ‘1’ sets the corresponding identifier bit for message filtering as 'do not care'.

If CAN operates with 11-bit identifier (standard frame), the upper bits MASK[28:11] are not used.

If CAN operates with 29-bit identifier (extended frame), the lower bits MASK[10:0] are used against the first received identifier bits and the upper bits MASK[28:11] are used with the last received identifier bits.

- **MRB[1:0]: Reserved Mask Bits**

Masks the reserved bits. MRB[1] is only used in extended format.

- **MRTR: Remote Transmission Request Mask**

Masks the RTR bit.

25.16.19 CAN Channel x Identifier Register

Name: CAN_IRx
Access: Read/Write
Offset: 0xXXC

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

- **ID[28:0]: Identifier of Channel x**

29-bit value for identifier.

For channels configured in standard frame mode (IDE bit reset to '0' in CAN_CRx), upper bits ID[28:11] are not used. In case of a transmission, only the base identifier bits ID[10:0] are sent from ID10 to ID0. In case of a reception, for acceptance filtering, only the base identifier ID[10:0] (together with MASK[10:0] bits from CAN_MSKx register) are checked against the received identifier.

For channels configured in extended frame mode (IDE bit set to '1' in CAN_CRx), upper bits ID[28:11] are used for the extended identifier and lower bits ID[10:0] are used for the base identifier. In case of a transmission, the base identifier ID[10:0] are sent first from ID10 to ID0, and extended identifier ID[28:11] are sent later from ID28 to ID11. And in case of a reception, for acceptance filtering, base identifier ID[10:0] (together with MASK[10:0] bits from CAN_MSKx register) are checked against first received identifier bits, and the extended identifier ID[28:11] (together with MASK[28:11] bits from CAN_MSKx register) are checked against last received identifier bits.

Frame Format	Identifier Length	Base ID	Extended ID
Standard frame	11 bits	ID[10:0]	N/A
Extended frame	29 bits	ID[10:0]	ID[28:11]

- **RB[1:0]: Reserved Bits**

These bits are sent to the control field. RB[1] generates the IDE bit in the standard frame format and the r1 bit in the extended frame format, respectively. RB[0] generates the r0 bit in the standard and extended frame formats. In the CAN 2.0A specification, RB[1] and RB[0] must be set dominant (i.e., to logical 0).

- **RTR: Remote Transmission Request**

In data frames, the RTR bit has to be dominant. Within a remote frame, the RTR bit has to be recessive.

25.16.20 CAN Channel x Control Register

Name: CAN_CRx
Access: Read/Write
Offset: 0xXX0

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	OVERWRITE
7	6	5	4	3	2	1	0
CHANEN	PCB	RPLYV	IDE	DLC[3:0]			

- **DLC[3:0]: Data Length Code**

This is the number of bytes in the data field of a message (from 0 to 8). This value is updated whenever a frame is received (data or remote). If the incoming DLC differs from the expected one, a warning is issued in the status register of the channel (CAN_SRX).

- **IDE: Extended Identifier Flag**

0: Identifier is 11 bits long (CAN rev 2.0A).

1: Identifier is 29 bits long (CAN rev 2.0B).

- **RPLYV: Automatic Reply**

0: No effect.

1: Channel x makes an automatic reply after receiving a remote frame.

- **PCB: Channel Producer**

0: Channel x is consumer.

1: Channel x is producer.

Bit PCB resets to 0 after a transmission.

- **CHANEN: Channel Enable**

0: Channel x disabled.

1: Channel x enabled.

Note: For detailed information, see ”Example of Use” on page 290.

- **OVERWRITE: Channel Overwrite Mode**

0: Channel x in normal mode.

1: Channel x in overwrite mode.

In overwrite mode, the channel is not disabled after each reception. New frames overwrite the previous frame.

In normal mode, the channel is automatically disabled after a frame reception.

25.16.21 CAN Channel x Stamp Register

Name: CAN_STPx
Access: Read-only
Offset: 0xXX4

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

- **STAMP[31:0]: Stamp Value**

These 32 bits stamp the date at which the message linked with Channel x has been emitted/received (depending on the producer/consumer bit). The value is copied from the WT second counter register.

25.16.22 CAN Channel x Clear Status Register

Name: CAN_CSRx
Access: Write-only
Offset: 0xXX8

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	OVRUN	FILLED	DLCW	–
7	6	5	4	3	2	1	0
RFRAME	TXOK	RXOK	BUS	STUFF	CRC	FRAME	ACK

- **ACK: Acknowledge Error Clear**

0: No effect.

1: Clears acknowledge error interrupt.

- **FRAME: Frame Error Clear**

0: No effect.

1: Clears frame error interrupt.

- **CRC: CRC Error Clear**

0: No effect.

1: Clears CRC error interrupt.

- **STUFF: Stuffing Error Clear**

0: No effect.

1: Clears stuffing error interrupt.

- **BUS: Bus Error Clear**

0: No effect.

1: Clear bus error interrupt.

- **RXOK: Reception Completed Clear**

0: No effect.

1: Clears reception completed interrupt.

- **TXOK: Transmission Completed Clear**

0: No effect.

1: Clears transmission completed interrupt.

- **RFRAME: Remote Frame Clear**

0: No effect.

1: Clears remote frame interrupt.

- **DLCW: DLC Warning Clear**

0: No effect.

1: Clears DLC warning.

- **FILLED: Filled Flag Clear**

0: No effect.

1: Clears the FILLED flag.

- **OVRUN: Overrun Flag Clear**

0: No effect.

1: Clears the OVERRUN flag.

25.16.23 CAN Channel x Status Register

Name: CAN_SRx
Access: Read-only
Offset: 0xXXC

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	OVRUN	FILLED	DLCW	–
7	6	5	4	3	2	1	0
RFRAME	TXOK	RXOK	BUS	STUFF	CRC	FRAME	ACK

- **ACK: Acknowledge Error**

0: No acknowledge error during last transmission.

1: An acknowledge error occurred during the last transmission. There was not a dominant bit during the ACK slot.

- **FRAME: Frame Error**

0: No frame error during last communication.

1: A frame error occurred during last communication. A fixed form bit contained one or more illegal bits.

- **CRC: CRC Error**

0: No CRC error during last reception.

1: A CRC error has been detected during the last reception. Received CRC sequence is not equal to the calculated one.

- **STUFF: Stuffing Error**

0: No stuffing error during the last communication.

1: A stuffing error occurred during the last communication. At least 6 consecutive equal bits have been detected.

- **BUS: Bus Error**

0: No bus error during last transmission.

1: A bus error occurred during last transmission. The transmitter sent a dominant bit but a recessive bit was detected on the network.

- **RXOK: Reception Completed**

0: No new reception completed.

1: A reception was completed without any error.

- **TXOK: Transmission Completed**

0: No new transmission completed.

1: A transmission was completed without any error.

- **RFRAME: Remote Frame**

0: No remote frame received.

1: A remote frame has been received.

Note: The RFRAME flag rises during the reception of remote frames, whereas the reception is not completed. On reception of a remote frame, software should wait for the RFRAME and RXOK flags in order to be sure that the received frame is not corrupted and that the identifier and DLC fields have been updated according to the received remote frame.

- **DLCW: DLC Warning**

0: No DLC warning.

1: DLC warning. Last message accepted with a different DLC than programmed in the CAN channel control register (CAN_CRx).

This bit does not generate an interrupt.

- **FILLED: Reception Buffer Filled**

0: No frame has been received since last clear of FILLED bit.

1: A frame has been received while the corresponding CAN channel is configured in overwrite mode and as a consumer (PCB = 0 and OVERWRITE = 1 in CAN_CRx). This flag can be enabled only when OVERWRITE mode is selected.

- **OVRUN: Overrun**

0: No frame has been received while FILLED flag is set to logical 1.

1: A frame has been received while FILLED flag is set to logical 1.

This flag can be raised only when OVERWRITE mode is selected.

This bit does not generate an interrupt.

25.16.24 CAN Channel x Interrupt Enable Register

Name: CAN_IERx
Access: Write-only
Offset: 0xXX0

25.16.25 CAN Channel x Interrupt Disable Register

Name: CAN_IDRx
Access: Write-only
Offset: 0xXX4

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
RFRAME	TXOK	RXOK	BUS	STUFF	CRC	FRAME	ACK

- **ACK: Acknowledge Error Mask**

0: Acknowledge error interrupt is disabled.

1: Acknowledge error interrupt is enabled.

- **FRAME: Frame Error Mask**

0: Frame error interrupt is disabled.

1: Frame error interrupt is enabled.

- **CRC: CRC Error Mask**

0: CRC error interrupt is disabled.

1: CRC error interrupt is enabled.

- **STUFF: Stuffing Error Mask**

0: Stuffing error interrupt is disabled.

1: Stuffing error interrupt is enabled.

- **BUS: Bus Error Mask**

0: Bus error interrupt is disabled.

1: Bus error interrupt is enabled.

- **RXOK: Reception Completed Mask**

0: Completed reception interrupt is disabled.

1: Completed reception interrupt is enabled.

- **TXOK: Transmission Completed Mask**

0: Completed transmission interrupt is disabled.

1: Completed transmission interrupt is enabled.

- **RFRAME: Remote Frame Mask**

0: Remote frame interrupt is disabled.

1: Remote frame interrupt is enabled.

25.16.26 CAN Channel x Interrupt Mask Register

Name: CAN_IMRx
Access: Read-only
Offset: 0xXX8

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
RFRAME	TXOK	RXOK	BUS	STUFF	CRC	FRAME	ACK

- **ACK: Acknowledge Error Mask**

0: Acknowledge error interrupt is disabled.

1: Acknowledge error interrupt is enabled.

- **FRAME: Frame Error Mask**

0: Frame error interrupt is disabled.

1: Frame error interrupt is enabled.

- **CRC: CRC Error Mask**

0: CRC error interrupt is disabled.

1: CRC error interrupt is enabled.

- **STUFF: Stuffing Error Mask**

0: Stuffing error interrupt is disabled.

1: Stuffing error interrupt is enabled.

- **BUS: Bus Error Mask**

0: Bus error interrupt is disabled.

1: Bus error interrupt is enabled.

- **RXOK: Reception Completed Mask**

0: Completed reception interrupt is disabled.

1: Completed reception interrupt is enabled.

- **TXOK: Transmission Completed Mask**

0: Completed transmission interrupt is disabled.

1: Completed transmission interrupt is enabled.

- **RFRAME: Remote Frame Mask**

0: Remote frame interrupt is disabled.

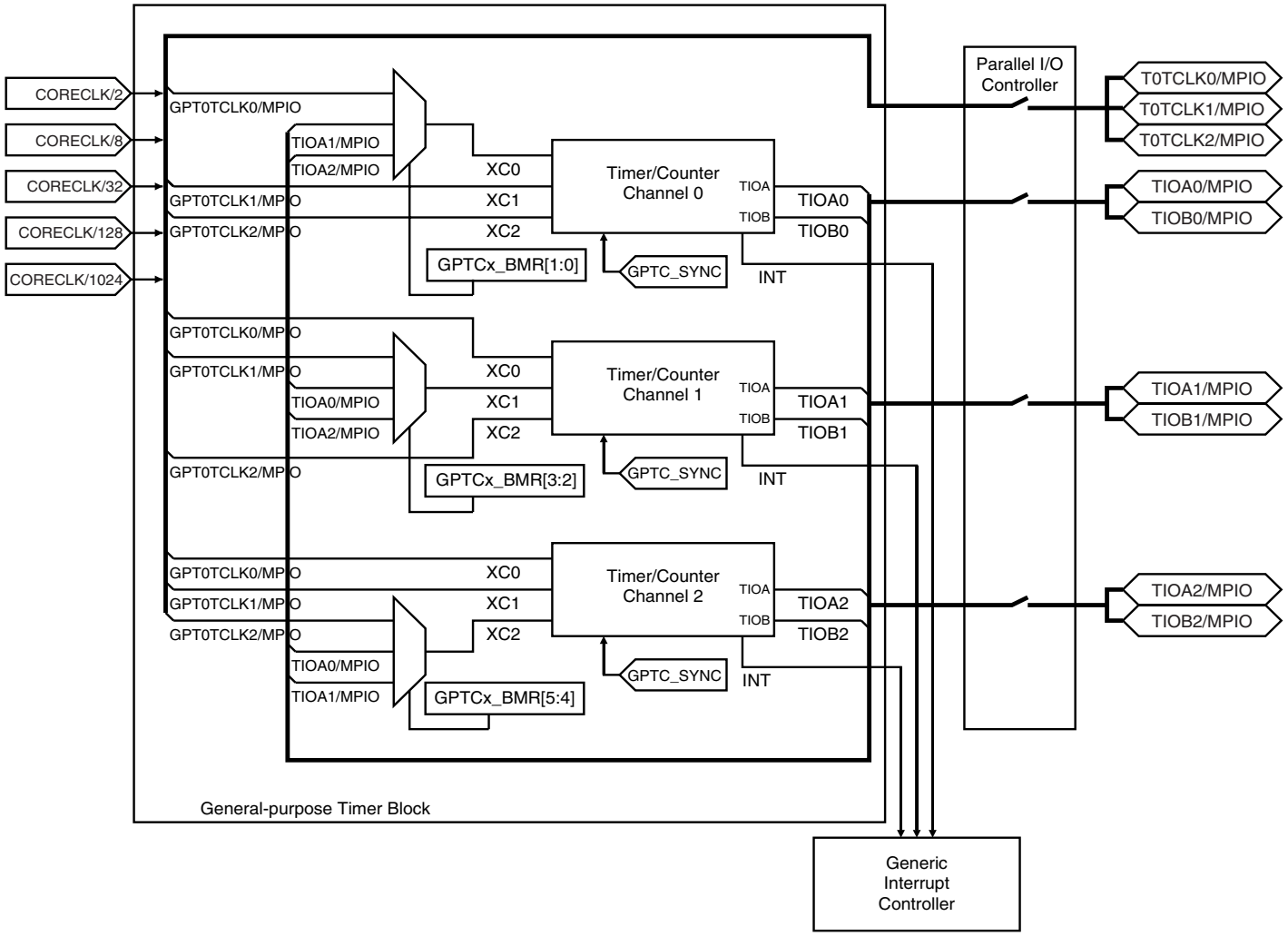
1: Remote frame interrupt is enabled.

26. General-purpose Timer (GPT)

26.1 Description

The AT91SAM7A1 has three independent general-purpose timer blocks. They are grouped in the same block and can be cascaded. Each timer has a 16-bit Timer/counter channel, a Power Management Controller and a Parallel I/O Controller. Each channel can be independently programmed using the two operating modes (capture mode or waveform mode) to perform a range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing, pulse width modulation and interrupt generation.

Figure 26-1. General-purpose Timer - Three-channel Block Diagram



Each General Purpose Timer may operate in a different mode (capture mode or waveform mode). Their counters are fully independent and each channel is associated with its own parallel I/O block and Power Management Controller.

- After a hardware reset, the timer pins are set as general purpose I/O (configured as input), the interrupts are disabled in the interrupt controller and the Timer Controller Clock and the PIO Controller Clock are disabled.
- Parallel I/O has priority over any Timer I/O configuration.

Each channel, shown in its configuration in Figure 26-1, has the following components:

- one 16-bit counter
- one 16-bit compare register (RC)
- two 16-bit capture/compare registers, RA and RB
- one multiplexer allowing the selection of eight clocks: five internal and three external (common to all channels). It is possible to combine two clocks to generate a burst clock. Each external clock can be used as external trigger source.
- an internal interrupt signal that can be programmed to generate processor interrupts via the Generic Interrupt Controller (GIC module). They are generated when one of the following events occurs:
  - Counter overflow
  - Load Register (A or B)
  - Equal Compare Register (A or B or C)
  - External edge detection
  - Overrun
- one software trigger
- one software reset that resets the channel and its associated registers (except Power Management registers)
- three parallel I/O pins that can be dedicated for multiple counter functions (Operation mode dependent) or in PIO mode.
  - TIOA, in capture mode, is an event input to load register A or register B or an input trigger. In waveform mode, it is an output to generate a waveform.
  - TIOB, in capture mode, is an input trigger. In waveform mode, it is either an output to generate a waveform (dual waveform mode) or an input trigger (single waveform mode).
  - TCLK, in capture mode and in waveform mode, is an external clock input.
- the synchronization bit that allows the synchronization of the three channels by generating a software trigger simultaneously on the three channels.
- the reset bit of the Block Control Register that resets the three channels simultaneously.

It is possible to chain two or three channels to increase the counter capacity (see ”Timer Controller Block Programming” on page 368).

26.2 Pin Description

Table 26-1 shows the internal pin configurations in the different operating modes.

Table 26-1. Pin Definition

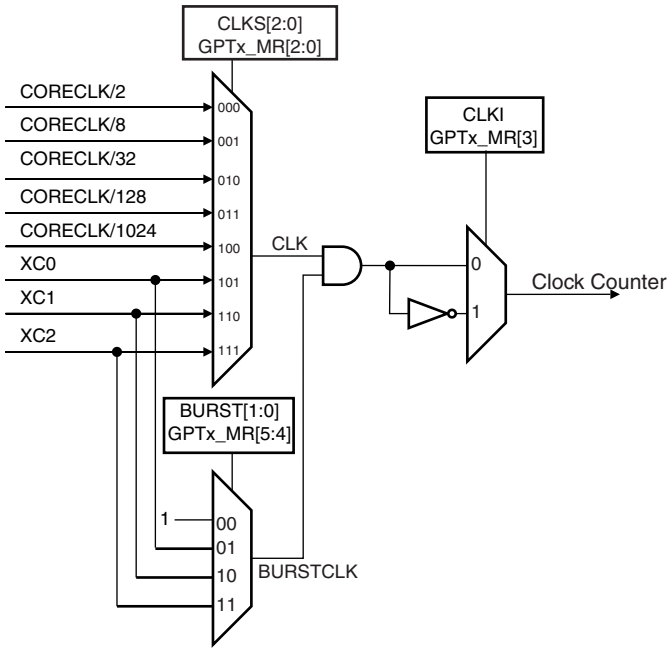
Pin (x = channel)	Capture Mode	Single Waveform Mode	Dual Waveform Mode
TIOAx	Input Capture or Trigger	Output Waveform	Output Waveform
TIOBx	Input Trigger	Input Trigger	Output Waveform
TCLKx	Input External Clock	Input External Clock	Input External Clock External Trigger

26.3 Clock Sources

The timer controller can use one of eight different clocks. The CLKS[2:0] bits of the mode registers GPTx_MR determine whether the counter is clocked by one of the five internal clock sources generated in the prescalar block and derived from CORECLK or one of the three external clock sources (TCLKx).

Figure 26-2 shows the clock selection block.

Figure 26-2. Clock Selection Block



The counter clock sources can be any of the following:

- External event input XCx
- One of 5 internal clocks (CORECLK/2, CORECLK/8, CORECLK/32, CORECLK/128, CORECLK/1024)

- A burst clock (see ”Burst Clock” on page 318).

The maximal count duration when an internal clock is used is determined by the internal clock MCK and the prescale number.

Maximal Count Duration (seconds) = 2^{16}/CLK where CLK is in Hz.

Counter Resolution = 1/CLK

26.3.1 External Clock

When an external clock source is used, the 16-bit counter can be programmed as a 16-bit event counter. An external transition (rising or falling following the state of the bit CLKI of the Mode register) increments the counter.

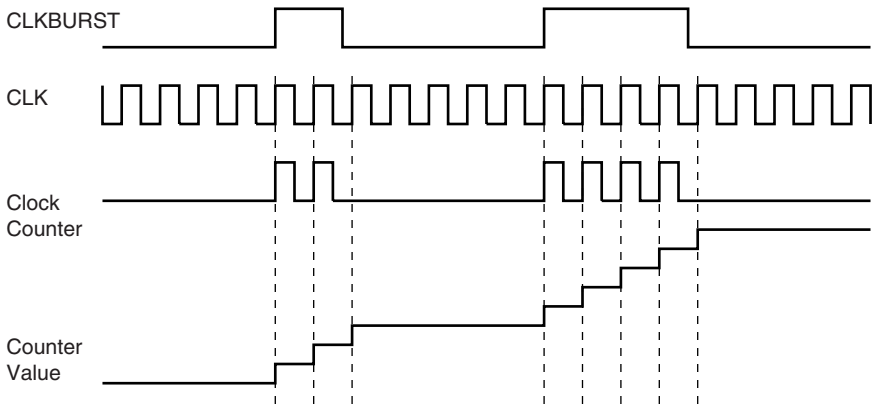
If an external clock is used, make sure that each of its pulse has a duration strictly higher than the CORECLK period.

26.3.2 Burst Clock

If the field BURST of the mode register selects an external clock, this clock is combined with CLK through a logical AND.

Thus the considered timer channel is clocked by CLK only when CLKBURST is high as shown in Figure 26-3.

Figure 26-3. Burst Clock



26.4 16-bit Counter

The 16-bit counter is a free-running counter clocked by eight sources: 5 internal and 3 external.

The program can access the counter value in real-time in read-only access with the counter value register GPT_CV.

When counter reset occurs, the counter is loaded with 0x0000 and begins its count if its clock is enabled (the clock is disabled or enabled by CLKDIS and CLKEN of the control register GPT_CR).

When the maximal value is reached (0xFFFF), the counter rolls over to a count of 0x0000, sets an overflow flag (the bit COVFS of the status register GPT_SR), may generate an interrupt (enabled by the bit COVFS of the interrupt enable register GPT_IER and disabled by the bit COVFS of the interrupt disable register GPT_IDR), and continues to count up.

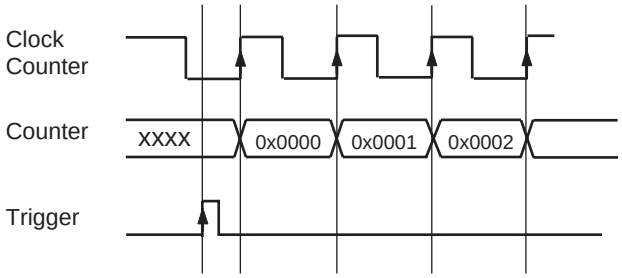
26.4.1 Counter Reset

During counting, the counter can be reset to 0x0000 following:

- a software Trigger
- an external Trigger
- an equality on Compare C
- the synchronous bit TCSYNC of the block control register GPT_BCR.

Each time it is reset, the counter passes to 0x0000 at the next valid counter clock edge. See Figure 26-4.

Figure 26-4. Counter Reset Diagram



26.5 16-bit Registers

Each channel contains three 16-bit registers.

The mode determines whether the capture/compare registers are used as capture registers or compare registers.

In capture mode, registers A and B are capture registers and can be loaded by TIOAx edges.

In waveform mode, registers A and B are compare registers.

Register C is always a compare register. The compare registers can generate a counter reset (RC) or a waveform modification (RA, RB and RC) when the counter reaches the value programmed in them.

In an application, the required compare register values must be calculated using the following equation:

$$\text{CompareValue} = (t \times \text{CLK}) - 1$$

where:

t = desired timer compare period (in seconds)

CLK = counter clock (in Hertz)

26.5.1 Example

To determine the value needed in a compare register to obtain an equality with the counter after 0.1 second with CORECLK = 30 MHz:

1. Determine the minimal prescale value by dividing CORECLK by the maximal counter value 0xFFFF (65,535) to know the divisor factor:
$$\frac{30,\ 000,\ 000}{65,\ 535} = 457.77$$

The value of the divider greater than or equal to 457.77 is DIV_{min} = 1024.

CLK must therefore be at least CORECLK/1024 (29.3 kHz) in order to obtain an equal condition after 0.1 seconds.

2. Calculate the register value with this clock using the following equation:

$$\text{Compare Value} = \left(0.1 \times \frac{\text{CORECLK}}{\text{DIV}}\right) - 1 = \left(0.1 \times \frac{30,\ 000,\ 000}{1024}\right) - 1 = 2928.69$$

Rounding value to 2929 (0x0B71) generates a 0.01% error.

26.6 External Edge Detection

The timer contains many external edge detection options.

The operating mode determines their number:

- Capture mode
  - TIOAx as load register and external trigger
  - TIOBx as external trigger
- Waveform mode: Dual waveform mode
  - XC0, XC1 or XC2 as external trigger
- Waveform mode: Single waveform mode
  - TIOBx as external trigger

For each edge detection, it is possible to choose a rising edge, a falling edge or both.

Whereas when a reset is caused, it occurs at the next valid counter clock edge.

If an external trigger is used, each of its pulses must have a duration strictly greater than the CORECLK period.

26.7 Interrupts

Each timer contains a total of eight timer interrupts and three PIO interrupts. They can be enabled or disabled from the GPT_IER and GPT_IDR. The programming mode determines which interrupts are available in Table 26-2.

Table 26-2. Available Interrupts

Interrupt Name	Capture Mode	Waveform Mode
Counter Overflow Interrupt COVFS	X	X
Load Overrun Interrupt LOVRS	X	
Compare Register A Interrupt CPAS		X
Compare Register B Interrupt CPBS		X
Compare Register C Interrupt CPCS	X	X
Load Capture Register A Interrupt LDRAS	X	
Load Capture Register B Interrupt LDRBS	X	
External Trigger Interrupt ETRGS	X	X
TCLK/MPIO Interrupt TCLKS	X	X
TIOA/MPIO Interrupt TIOAS	X	X
TIOB/MPIO Interrupt TIOBS	X	X

26.8 PIO Controller

Each timer channel has three programmable I/O lines. These I/O lines are multiplexed with signals (TIOA, TIOB, TCLK) of the timer channel to optimize the use of available package pins. These lines are controlled by the timer channel PIO controller.

26.9 Power Management

Each timer channel (TC0, TC1 and TC2) is provided with a power management block allowing optimization of power consumption (see ”Power Management Block” on page 23).

26.10 General-purpose Timer (GPT) User Interface

Base Address GPT Channel 0: 0xFFFC8000

Base Address GPT Channel 1: 0xFFFC8100

Base Address GPT Channel 2: 0xFFFC8200

Table 26-3. GPT Memory Map and Control Registers

Offset	Register	Name[1]	Access	Reset State
0x00	PIO Enable Register	GPTx_PER	Write-only	–
0x04	PIO Disable Register	GPTx_PDR	Write-only	–
0x08	PIO Status Register	GPTx_PSR	Read-only	0x00070000
0x0C	Reserved	–	–	–
0x10	PIO Output Enable Register	GPTx_OER	Write-only	–
0x14	PIO Output Disable Register	GPTx_ODR	Write-only	–
0x18	PIO Output Status Register	GPTx_OSR	Read-only	0x00000000
0x1C – 0x2C	Reserved	–	–	–
0x30	PIO Set Output Data Register	GPTx_SODR	Write-only	–
0x34	PIO Clear Output Data Register	GPTx_CODR	Write-only	–
0x38	PIO Output Data Status Register	GPTx_ODSR	Read-only	0x00000000
0x3C	PIO Pin Data Status Register	GPTx_PDSR	Read-only	0x000X0000
0x40	PIO Multi-Driver Enable Register	GPTx_MDER	Write-only	–
0x44	PIO Multi-Driver Disable Register	GPTx_MDDR	Write-only	–
0x48	PIO Multi-Driver Status Register	GPTx_MDSR	Read-only	0x00000000
0x4C	Reserved	–	–	–
0x50	Enable Clock Register	GPTx_ECR	Write-only	–
0x54	Disable Clock Register	GPTx_DCR	Write-only	–
0x58	Power Management Status Register	GPTx_PMSR	Read-only	0x00000000
0x5C	Reserved	–	–	–
0x60	Control Register	GPTx_CR	Write-only	–
0x64	Mode Register	GPTx_MR	Read/Write	0x00000000
0x68	Reserved	–	–	–
0x6C	Reserved	–	–	–
0x70	Status Register	GPTx_SR	Read-only	0x00000X00
0x74	Interrupt Enable Register	GPTx_IER	Write-only	–
0x78	Interrupt Disable Register	GPTx_IDR	Write-only	–
0x7C	Interrupt Mask Register	GPTx_IMR	Read-only	0x00000000
0x80	Counter Value	GPTx_CV	Read-only	0x00000000
0x84	Capture - Compare Register A	GPTx_RA	Read/Write	0x00000000
0x88	Capture - Compare Register B	GPTx_RB	Read/Write	0x00000000
0x8C	Compare Register C	GPTx_RC	Read/Write	0x00000000

Table 26-3. GPT Memory Map and Control Registers (Continued)

Offset	Register	Name[1]	Access	Reset State
Control and Test Registers				
0x0300	Block Control Register	GPT_BCR	Write-only	–
0x0304	Block Mode Register	GPT_BMR	Read/Write	0x00000000
0x308 – 0x3FC	Reserved	–	–	–
0x400	Test Control Register	GPT_TSTC	Write-only	–
0x404	Test Mode Register	GPT_TSTM	Read/Write	0x00000000

Note: 1. x denotes the GPT channel number (0, 1, 2).

26.10.1 GPT PIO Enable Register

Name: GPT_PER
Access: Write-only
Offset: 0x00

26.10.2 GPT PIO Disable Register

Name: GPT_PDR
Access: Write-only
Offset: 0x04

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

- **TIOB: TIOB Pin**

0: PIO is inactive on the TIOBx pin (Timer Controller is active).

1: PIO is active on the TIOBx pin (Timer Controller is inactive).

- **TIOA: TIOA Pin**

0: PIO is inactive on the TIOAx pin (Timer Controller is active).

1: PIO is active on the TIOAx pin (Timer Controller is inactive).

- **TCLK: TCLK Pin**

0: PIO is inactive on the TCLKx pin (Timer Controller is active).

1: PIO is active on the TCLKx pin (Timer Controller is inactive).

Note: x: channel number

26.10.3 GPT PIO Status Register

Name: GPT_PSR
Access: Read-only
Offset: 0x08

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

- **TIOB: TIOB Pin**

0: PIO is inactive on the TIOBx pin (Timer Controller is active).

1: PIO is active on the TIOBx pin (Timer Controller is inactive).

- **TIOA: TIOA Pin**

0: PIO is inactive on the TIOAx pin (Timer Controller is active).

1: PIO is active on the TIOAx pin (Timer Controller is inactive).

- **TCLK: TCLK Pin**

0: PIO is inactive on the TCLKx pin (Timer Controller is active).

1: PIO is active on the TCLKx pin (Timer Controller is inactive).

Note: x: channel number

26.10.4 GPT PIO Output Enable Register

Name: GPT_OER
Access: Write-only
Offset: 0x10

26.10.5 GPT PIO Output Disable Register

Name: GPT_ODR
Access: Write-only
Offset: 0x14

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

- **TIOB: TIOB Pin**

0: The TIOBx PIO pin is input.

1: The TIOBx PIO pin is output

- **TIOA: TIOA Pin**

0: The TIOAx PIO pin is input.

1: The TIOAx PIO pin is output

- **TCLK: TCLK Pin**

0: The TCLKx PIO pin is input.

1: The TCLKx PIO pin is output.

Note: x: channel number

26.10.6 GPT PIO Output Status Register

Name: GPT_OSR
Access: Read-only
Offset: 0x18

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

- **TIOB: TIOB Pin**

0: The TIOBx PIO pin is input.

1: The TIOBx PIO pin is output

- **TIOA: TIOA Pin**

0: The TIOAx PIO pin is input.

1: The TIOAx PIO pin is output

- **TCLK: TCLK Pin**

0: The TCLKx PIO pin is input.

1: The TCLKx PIO pin is output.

Note: x: channel number

26.10.7 GPT PIO Set Output Data Register

Name: GPT_SODR
Access: Write-only
Offset: 0x30

26.10.8 GPT PIO Clear Output Data Register

Name: GPT_CODR
Access: Write-only
Offset: 0x34

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

- **TIOB: TIOB Pin**

0: The output data for the TIOBx is programmed to 0.

1: The output data for the TIOBx is programmed to 1.

- **TIOA: TIOA Pin**

0: The output data for the TIOAx is programmed to 0.

1: The output data for the TIOAx is programmed to 1.

- **TCLK: TCLK Pin**

0: The output data for the TCLKx is programmed to 0.

1: The output data for the TCLKx is programmed to 1.

Note: x: channel number

26.10.9 GPT PIO Output Data Status Register

Name: GPT_ODSR
Access: Read-only
Offset: 0x38

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

- **TIOB: TIOB Pin**

0: The output data for the TIOBx is programmed to 0.

1: The output data for the TIOBx is programmed to 1.

- **TIOA: TIOA Pin**

0: The output data for the TIOAx is programmed to 0.

1: The output data for the TIOAx is programmed to 1.

- **TCLK: TCLK Pin**

0: The output data for the TCLKx is programmed to 0.

1: The output data for the TCLKx is programmed to 1.

Note: x: channel number

26.10.10 GPT PIO Pin Data Status Register

Name: GPT_PDSR
Access: Read-only
Offset: 0x3C

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

- **TIOB: TIOB Pin**

0: The pin TIOBx is at logic 0.

1: The pin TIOBx is at logic 1.

- **TIOA: TIOA Pin**

0: The pin TIOAx is at logic 0.

1: The pin TIOAx is at logic 1.

- **TCLK: TCLK Pin**

0: The pin TCLKx is at logic 0.

1: The pin TCLKx is at logic 1.

Note: x: channel number

26.10.11 GPT PIO Multi-driver Enable Register

Name: GPT_MDER
Access: Write-only
Offset: 0x40

26.10.12 GPT PIO Multi-driver Disable Register

Name: GPT_MDDR
Access: Write-only
Offset: 0x44

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

- **TIOB: TIOB Pin**

0: TIOBx pin is not configured as an open drain.

1: TIOBx pin is configured as an open drain.

- **TIOA: TIOA Pin**

0: TIOAx pin is not configured as an open drain.

1: TIOAx pin is configured as an open drain.

- **TCLK: TCLK Pin**

0: TCLKx pin is not configured as an open drain.

1: TCLKx pin is configured as an open drain.

Note: x: channel number

26.10.13 GPT PIO Multi-driver Status Register

Name: GPT_MDSR
Access: Read-only
Offset: 0x48

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

- **TIOB: TIOB Pin**

0: TIOBx pin is not configured as an open drain.

1: TIOBx pin is configured as an open drain.

- **TIOA: TIOA Pin**

0: TIOAx pin is not configured as an open drain.

1: TIOAx pin is configured as an open drain.

- **TCLK: TCLK Pin**

0: TCLKx pin is not configured as an open drain.

1: TCLKx pin is configured as an open drain.

Note: x: channel number

26.10.14 GPT Enable Clock Register

Name: GPT_ECR
Access: Write-only
Offset: 0x50

26.10.15 GPT Disable Clock Register

Name: GPT_DCR
Access: Write-only
Offset: 0x54

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
–	–	–	–	–	–	GPT	PIO

- **PIO: PIO Clock**

1: PIO controller clock is enabled.

0: PIO controller clock is disabled.

- **GPT: General-purpose Timer Clock**

1: General-purpose Timer channel and clock divider clock enabled.

0: General-purpose Timer channel and clock divider clock disabled.

26.10.16 GPT Power Management Status Register

Name: GPT_PMSR
Access: Read-only
Offset: 0x58

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
–	–	–	–	–	–	GPT	PIO

- **PIO: PIO Clock**

1: PIO controller clock is enabled.

0: PIO controller clock is disabled.

- **GPT: General-purpose Timer Clock**

1: General-purpose Timer channel and clock divider clock enabled.

0: General-purpose Timer channel and clock divider clock disabled.

26.11 General-purpose Timer in Capture Mode

26.11.1 Description

The capture (wave measurement) mode is entered by setting WAVE (bit [15] in the Mode Register) to 0.

It is the default operating mode after a hardware reset. It forces TIOAx and TIOBx pins as input pins.

The capture mode provides the possibility to determine the duration between two events. An event may either be an external input signal on TIOAx or TIOBx or an internal event (software trigger or equality between the counter and a predefined compare value). An external event (rising or falling edge) on TIOAx can result in capture register A being loaded, capture register B being loaded or a trigger effect (reset and start the counter).

A predefined compare value (16-bit) can be set in the compare register C.

When the capture register B is loaded, it can disable the counter clock and/or stop the counter.

The user may choose an internal clock source (CORECLK/2, CORECLK/8, CORECLK/32, CORECLK/128 or CORECLK/1024) or an external clock (TCLK0, TCLK1 or TCLK2).

A burst mode is available. It generates a burst clock. For more details, refer to ”Clock Sources” on page 317.

Six interrupts can be produced:

- External trigger detected
- RA loaded
- RB loaded
- Counter overflow (when the counter passes from 0xFFFF to 0x0000)
- Overrun (when RA or RB is reloaded before the old value is read)
- Compare RC (the counter reaches the value stored in register C)

Finally, the synchronize register can be used to cause a software trigger for reset and start the counter at the next valid counter clock edge on all channels at the same time.

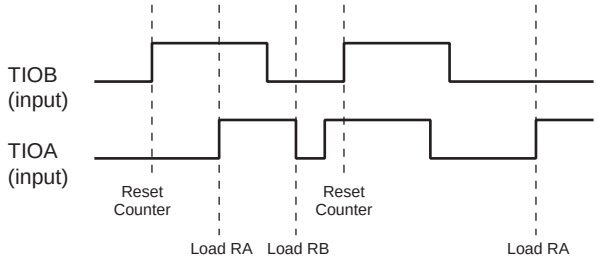
Figure 26-5 to Figure 26-8 show different applications using the capture mode.

For more details, refer to application notes.

26.11.1.1 Measure TIOA Pulse and Phase between TIOB and TIOA

A TIOBx rising edge resets and starts the counter. A rising TIOAx edge loads RA and a falling TIOAx edge loads RB. Once RB is loaded, a trigger restarts a capture cycle. RA contains the phase between TIOBx and TIOAx. (RB - RA) is the duration of the TIOAx pulse.

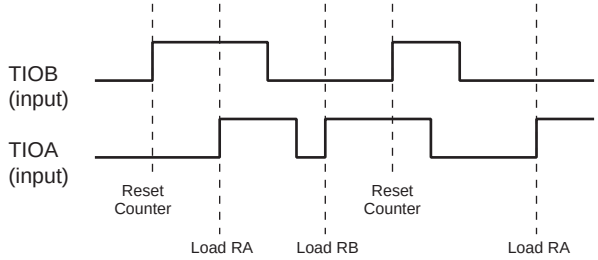
Figure 26-5. TIOA Pulse



26.11.1.2 Measure Duration between Two Successive Rising TIOA Edges

A TIOBx rising edge resets and starts the counter. The first rising TIOAx edge after the reset loads RA and a second loads RB. RA contains the phase between TIOBx and TIOAx. (RB-RA) is the period of the TIOAx pulse.

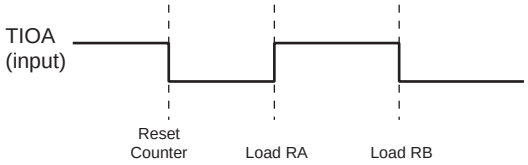
Figure 26-6. TIOA Edges



26.11.1.3 Measure the Duration of a TIOA Pulse or Period (TIOB Not Used)

A TIOAx falling edge resets, starts the counter and loads RB if RA is already loaded. A TIOAx rising edge loads RA. RA contains the duration of a TIOAx pulse (low level). RB contains the duration of the TIOAx period.

Figure 26-7. TIOA Pulse or Period



26.11.1.4 Event Counter on an External Clock TCLK

The counter is incremented with each TCLKx rising edge. This application can be generated in waveform mode. The counter value contains the number of detected TCLKx rising edges.

Figure 26-8. Event Counter on an External Clock TCLK

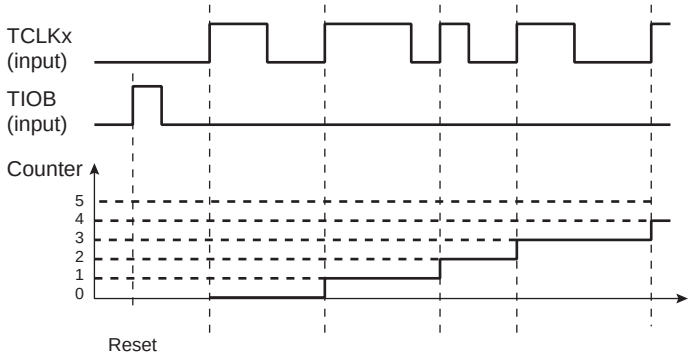
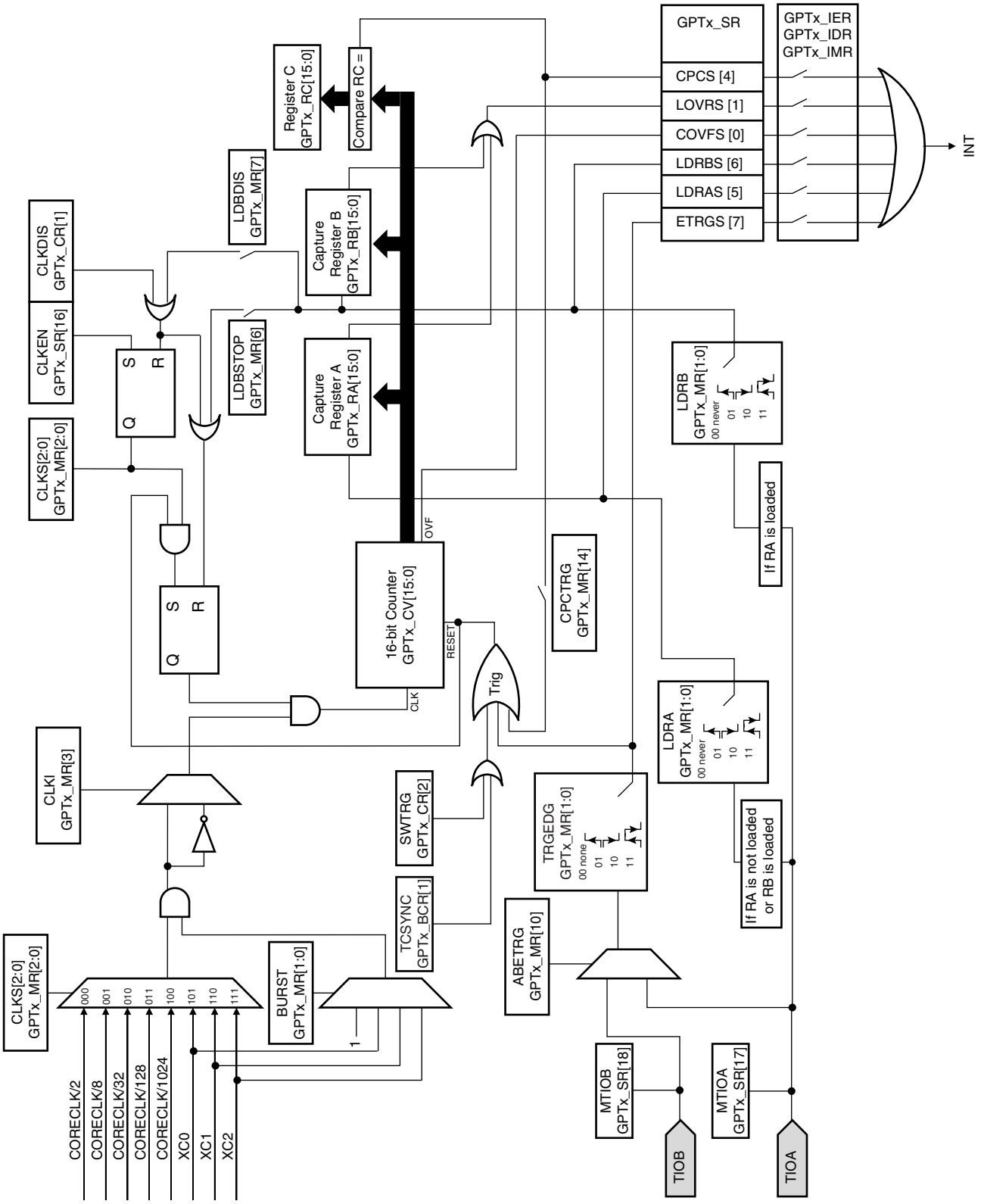


Figure 26-9. General-purpose Timer in Capture Mode



26.11.2 GPT Control Register in Capture Mode

Name: GPT_CR
Access: Write-only
Offset: 0x60

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	SWRST

- **SWRST: Software Reset**

0: No effect.

1: Generates a software reset.

A software triggered hardware reset of the channel is performed. It resets all the registers, including PIO and PMC registers (except GPTX_PMSR).

- **CLKEN: Counter Clock Enable**

0: No effect.

1: Enables counter clock if CLKDIS = 0.

- **CLKDIS: Counter Clock Disable**

0: No effect.

1: Disables counter clock.

- **SWTRG: Software Trigger**

0: No effect.

1: Generates a software trigger.

This bit generates a software trigger for resetting and starting the counter at the next valid counter clock edge when the counter clock is enabled.

Note: x: channel number

26.11.3 GPT Mode Register in Capture Mode

Name: GPT_MR
Access: Read/Write
Offset: 0x64

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	LDRB[1:0]		LDRA[1:0]	
15	14	13	12	11	10	9	8
WAVE = 0	CPCTRG	–	–	–	ABETRG	ETRGEDG[1:0]	
7	6	5	4	3	2	1	0
LDBIS	LDBSTOP	BURST[1:0]		CLKI	CLKS[2:0]		

- **CLKS[2:0]: Clock Select**

CLKS[2:0]			Counter Clock Source
0	0	0	CORECLK/2
0	0	1	CORECLK/8
0	1	0	CORECLK/32
0	1	1	CORECLK/128
1	0	0	CORECLK/1024
1	0	1	XC0
1	1	0	XC1
1	1	1	XC2

XCx consists of three external clocks.

For more details, see ”Clock Sources” on page 317.

- **CLKI: Clock Inverter**

0: Normal clock (The counter is incremented on a rising edge)

1: Inverted clock (The counter is incremented on a falling edge)

- **BURST[1:0]: Burst**

This signal is combined with the selected clock through a logical AND.

BURST[1:0]		Burst Signal Selected
0	0	None
0	1	XC0
1	0	XC1
1	1	XC2

For more details, see ”Clock Sources” on page 317.

- **LDBSTOP Load RB Stops Counter**

0: The counter is not stopped when RB is loaded.

1: The counter is stopped when RB is loaded.

If the counter is stopped, it can restart (to 0x0000) just with a trigger condition.

If a TIOAx edge both induces a trigger condition and loads capture register B which in turn stops the counter, the trigger has no effect.

- **LDBDIS: Load RB Disables Clock**

0: The counter clock is not disabled when RB is loaded.

1: The counter clock is disabled and the counter stopped when RB is loaded.

If the counter clock is disabled, it can be enabled only by asserting CLKEN, bit [0] of the control register.

- **ETRGEDG[1:0]: External Trigger Edge**

The external trigger source is either TIOAx or TIOBx following ABETRG, bit [10] of the mode register.

ETRGEDG[1:0]		Edge
0	0	None
0	1	Rising edge
1	0	Falling edge
1	1	Each edge

When an external trigger is generated, three events occur:

– It resets and starts the counter.

– The ETRGS flag is set in the status register.

– If enabled, ETRGS interrupt is generated.

- **ABETRG: TIOA or TIOB as External Trigger**

0: Select TIOBx as external trigger

1: Select TIOAx as external trigger

Note: The counter can start only if the clock is enabled.

- **CPCTRG: Compare RC Trigger**

0: An equal condition on RC does not cause a trigger.

1: An equal condition on RC causes a trigger.

Note: The counter can start only if the clock is enabled.

- **WAVE: Waveform**

0: Capture mode.

1: Waveform mode.

Note: The capture mode is the default mode after hardware reset.

- **LDRA[1:0]: Load RA**

These two bits activate one of four possible TIOAx edge conditions to load RA.

Note: The application must ensure that the event that loads RA occurs after the next counter clock edge following the configuration of LDRA (the counter is reset on the next counter clock edge following the configuration of LDRA).

- **LDRB[1:0]: Load RB**

These two bits activate one of four possible TIOAx edge conditions to load RB.

LDRx		Edge
0	0	None
0	1	Rising edge
1	0	Falling edge
1	1	Each edge

26.11.4 GPT Status Register in Capture Mode

Name: GPT_SR
Access: Read-only
Offset: 0x70

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	TCLKS	TIOAS	TIOBS
15	14	13	12	11	10	9	8
–	–	–	–	–	MTIOB	MTIOA	CLKSTA
7	6	5	4	3	2	1	0
ETRGS	LDRBS	LDRAS	CPCS	–	–	LOVRS	COVFS

Note: This register is a “read-active” register, which means that reading it can affect the state of some bits. When reading GPT_SR register, following bits are cleared if set: COVFS, LOVRS, CPCS, LDRAS, LDRBS, ETRGS, TIOBS, TIOAS and TCLKS.

- **COVFS: Counter Overflow Status**

This bit is set when a counter overflow is detected. An overflow occurs when the counter reaches its maximal value 0xFFFF (216 - 1) and passes to 0x0000.

0: No overflow detected.

1: Overflow detected since last read of GPTX_SR.

- **LOVRS: Load Overrun Status**

This bit is set when an overrun is detected. An overrun occurs when the capture registers A or B are reloaded before being read.

0: No overrun detected.

1: An overrun detected since last read of GPTX_SR.

- **CPCS: Compare Register C Status**

This bit is set when the counter reaches the register C value.

0: Compare C condition has not occurred since last read of GPTX_SR.

1: Compare C condition has occurred since last read of GPTX_SR.

- **LDRAS: Load Register A Status**

0: Register A not loaded.

1: Register A loaded since last read of GPTX_SR.

- **LDRBS: Load Register B Status**

0: Register B not loaded.

1: Register B loaded since last read of GPTX_SR.

- **ETRGS: External Trigger Status**

This bit is set when an external trigger is detected. An external trigger occurs with a valid edge (the edge polarity is set by ETRGEDG[1:0] of the mode register) on the valid trigger pin (set by ABETRG of the mode register).

0: External trigger not detected.

1: External trigger detected since last read of GPTX_SR.

- **CLKSTA: Clock Status**

0: Clock disabled.

1: Clock enabled.

- **MTIOA: TIOA Mirror**

This bit reflects the TIOAx pin value.

As TIOAx is an input after a hardware reset, its reset value is undefined.

- **MTIOB: TIOB Mirror**

This bit reflects the TIOBx pin value.

As TIOBx is an input after a hardware reset, its reset value is undefined.

- **TIOBS: TIOB Status**

0: At least one input change has been detected on the pin TIOBx since the register was last read.

1: No input change has been detected on the TIOBx pin since the register was last read.

- **TIOAS: TIOA Status**

0: At least one input change has been detected on the TIOAx pin since the register was last read.

1: No input change has been detected on the TIOAx pin since the register was last read.

- **TCLKS: TCLK Status**

0: At least one input change has been detected on the TCLKx pin since the register was last read.

1: No input change has been detected on the TCLKx pin since the register was last read.

Note: X: channel number

26.11.5 GPT Interrupt Enable Register in Capture Mode

Name: GPT_IER
Access: Write-only
Offset: 0x74

26.11.6 GPT Interrupt Disable Register in Capture Mode

Name: GPT_IDR
Access: Write-only
Offset: 0x78

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	TCLKS	TIOAS	TIOBS
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ETRGS	LDRBS	LDRAS	CPCS	–	–	LOVRS	COVFS

- **COVFS: Counter Overflow Status**

0: COVFS interrupt is disabled.

1: COVFS interrupt is enabled.

- **LOVRS: Load Overrun Status**

0: LOVRS interrupt is disabled.

1: LOVRS interrupt is enabled.

- **CPCS: Compare Register C Status**

0: CPCS interrupt is disabled.

1: CPCS interrupt is enabled.

- **LDRAS: Load Register A Status**

0: LDRAS interrupt is disabled.

1: LDRAS interrupt is enabled.

- **LDRBS: Load Register B Status**

0: LDRBS interrupt is disabled.

1: LDRBS interrupt is enabled.

- **ETRGS: External Trigger Status**

0: ETRGS interrupt is disabled.

1: ETRGS interrupt is enabled.

- **TIOBS: TIOB Status**

0: TIOBS interrupt is disabled.

1: TIOBS interrupt is enabled.

- **TIOAS: TIOA Status**

0: TIOAS interrupt is disabled.

1: TIOAS interrupt is enabled.

- **TCLKS: TCLK Status**

0: TCLKS interrupt is disabled.

1: TCLKS interrupt is enabled.

26.11.7 GPT Interrupt Mask Register in Capture Mode

Name: GPT_IMR
Access: Read-only
Offset: 0x7C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	TCLKS	TIOAS	TIOBS
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ETRGS	LDRBS	LDRAS	CPCS	–	–	LOVRS	COVFS

- **COVFS: Counter Overflow Status**

0: COVFS interrupt is disabled.

1: COVFS interrupt is enabled.

- **LOVRS: Load Overrun Status**

0: LOVRS interrupt is disabled.

1: LOVRS interrupt is enabled.

- **CPCS: Compare Register C Status**

0: CPCS interrupt is disabled.

1: CPCS interrupt is enabled.

- **LDRAS: Load Register A Status**

0: LDRAS interrupt is disabled.

1: LDRAS interrupt is enabled.

- **LDRBS: Load Register B Status**

0: LDRBS interrupt is disabled.

1: LDRBS interrupt is enabled.

- **ETRGS: External Trigger Status**

0: ETRGS interrupt is disabled.

1: ETRGS interrupt is enabled.

- **TIOBS: TIOB Status**

0: TIOBS interrupt is disabled.

1: TIOBS interrupt is enabled.

- **TIOAS: TIOA Status**

0: TIOAS interrupt is disabled.

1: TIOAS interrupt is enabled.

- **TCLKS: TCLK Status**

0: TCLKS interrupt is disabled.

1: TCLKS interrupt is enabled.

26.11.8 GPT Counter Value in Capture Mode

Name: GPT_CV
Access: Read-only
Offset: 0x80

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[15:8]							
7	6	5	4	3	2	1	0
CV[7:0]							

- **CV[15:0]: Counter Value**

These 16 bits contain the counter value in real time.

The maximal counter value is 0xFFFF = 65535.

When a trigger occurs, the counter will be reset to 0x0000 at the next valid counter clock edge.

26.11.9 GPT Register A in Capture Mode

Name: GPT_RA
Access: Read-only
Offset: 0x84

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[15:8]							
7	6	5	4	3	2	1	0
RA[7:0]							

- **RA[15:0]: Register A Value**

This register is loaded with the current counter value when a valid edge occurs on TIOAx pin.

This valid edge is defined by LDRA[1:0] of the mode register.

When this register is loaded, two events occur:

– The LDRAS flag is set in the status register.

– If enabled, LDRAS interrupt is generated.

This register can not be loaded if the counter is stopped or the counter clock disabled.

26.11.10 GPT Register B in Capture Mode

Name: GPT_RA
Access: Read-only
Offset: 0x88

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[15:8]							
7	6	5	4	3	2	1	0
RB[7:0]							

- **RB[15:0]: Register B Value**

This register is loaded with the current counter value when a valid edge occurs on TIOBx pin.

This valid edge is defined by LDRB[1:0] of the mode register.

When this register is loaded, four events occur:

– The LDRBS flag is set in the status register.
– If enabled, LDRBS interrupt is generated.
– The counter clock can be disabled according to LDBDIS (bit [7] of the mode register).
– The counter can be stopped according to LDBSTOP (bit [6] of the mode register).

This register cannot be loaded if the counter is stopped or the counter clock disabled.

26.11.11 GPT Register C in Capture Mode

Name: GPT_RC
Access: Read/Write
Offset: 0x8C

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[15:8]							
7	6	5	4	3	2	1	0
RC[7:0]							

- **RC[15:0]: Register C Value**

When the counter reaches this value, three events occur:

– The CPCS flag is set in the status register.
– If enabled, CPCS interrupt is generated.
– If CPCTRG (bit [14] of the mode register) is high, the counter is reset and restarts at 0x0000.

26.12 General-purpose Timer in Waveform Mode

26.12.1 Description

The waveform mode is entered by setting the bit WAVE in the GPTX_MR to 1. It forces TIOAx as an output pin. TIOBx can be used either as an output (dual waveform mode) or as an input (single waveform mode).

The waveform mode provides the possibility to generate either symmetrical or variable duty-cycle waveforms.

The TIOA pin is controlled (set, cleared or toggled) by four events:

- a software trigger
- an external event edge (rising, falling edge or both)
- an equality between the counter and compare register A value
- an equality between the counter and compare register C value

As an output pin, the TIOB pin is controlled (set, cleared or toggled) by four events:

- a software trigger
- an external event edge (rising, falling edge or both)
- an equality between the counter and compare register B value
- an equality between the counter and compare register C value

When TIOB is used as an external trigger source, the compare register B is not used.

When an equal condition on compare register C is detected, one of three events can occur:

- The counter can reset and start at the next valid counter clock edge.
- The counter can be stopped.
- The counter can be stopped and the counter clock disabled.

If this condition restarts the counter, the user can generate a continuous wave with a period proportional to compare register C +1 value. If it does not restart the counter, the user can generate a continuous waveform with a period proportional to 0xFFFF (maximal counter value: 2^{16} - 1).

The user may choose an internal clock source (CORECLK/2, CORECLK/8, CORECLK/32, CORECLK/128 or CORECLK/1024) or external clock (TCLK0, TCLK1 or TCLK2). A burst mode is available. It generates a burst clock. For more details, refer to ”Clock Sources” on page 317.

Five interrupts can be produced:

- External trigger detected
- Counter overflow (when the counter passes from 0xFFFF to 0x0000)
- Compare RA (the counter reaches the value stored in register A)
- Compare RB (the counter reaches the value stored in register B)
- Compare RC (the counter reaches the value stored in register C)

Finally, the synchronize register can be used to cause a software trigger for reset and start the counter at the next valid counter clock edge on all channels at the same time.

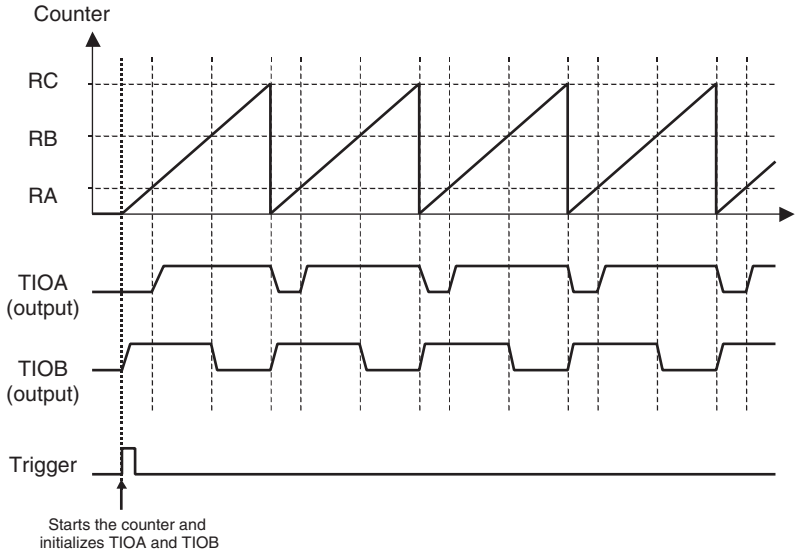
Figure 26-10 on page 351 to Figure 26-14 on page 353 show different applications possible with the waveform mode.

26.12.1.1 Dual Pulse Width Modulation (PWM) Generation

TIOAx is toggled by RA and RC, TIOBx by RB and RC.

RC contains frequency of both signals. RA determines the TIOAx duty cycle and RB the TIOBx duty cycle.

Figure 26-10. Dual Pulse Width Modulation



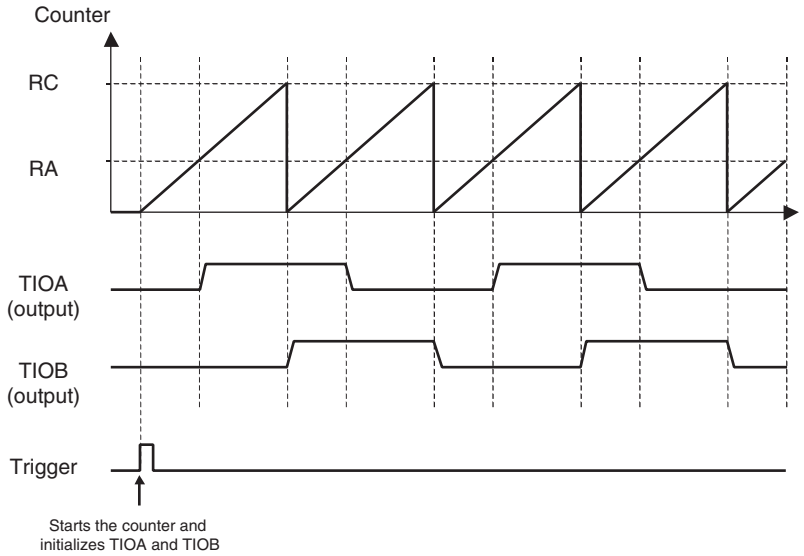
26.12.1.2 Generation of Two Identical Out-of-phase Square-wave Signals

TIOAx is toggled each time the counter reaches RA value, TIOBx each time the counter reaches RC value.

A trigger (external or software) starts the counter and initializes TIOAx and TIOBx.

RC contains the frequency of both signals. RA contains the delay between the signals.

Figure 26-11. Two Square Signals



26.12.1.3 Pulse Generation

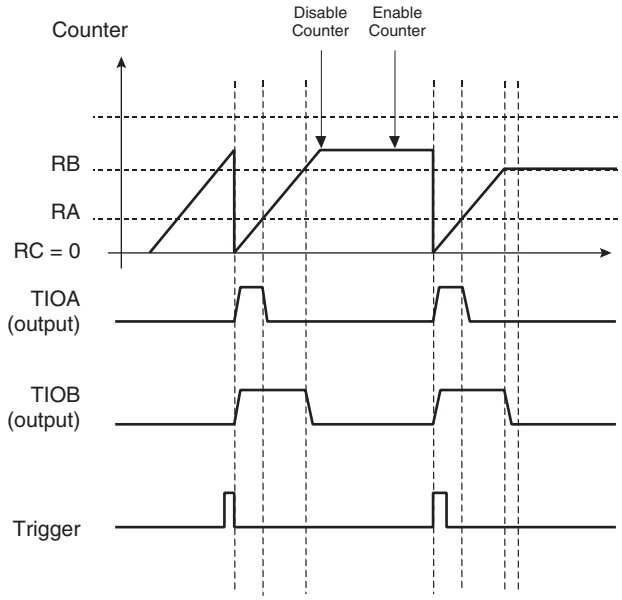
This application generates only one pulse on TIOAx and on TIOBx between two triggers. The pulses start with a trigger. The duration is given by the value of RA for TIOAx pulse and the value of RB for TIOBx pulse. After each new trigger, another pulse is generated. When a trigger occurs, the counter is reset at the next valid counter clock edge when the counter clock is enabled.

If we want to start the pulse exactly when the counter is reset, RC must equal 0. Thus, it is the comparison with RC = 0 that starts the pulse, and not the trigger.

In this case, the user must disable the counter clock when a compare RB is detected to stop the counter.

To accept a new trigger, the user must then re-enable the counter clock.

Figure 26-12. Pulse Generation

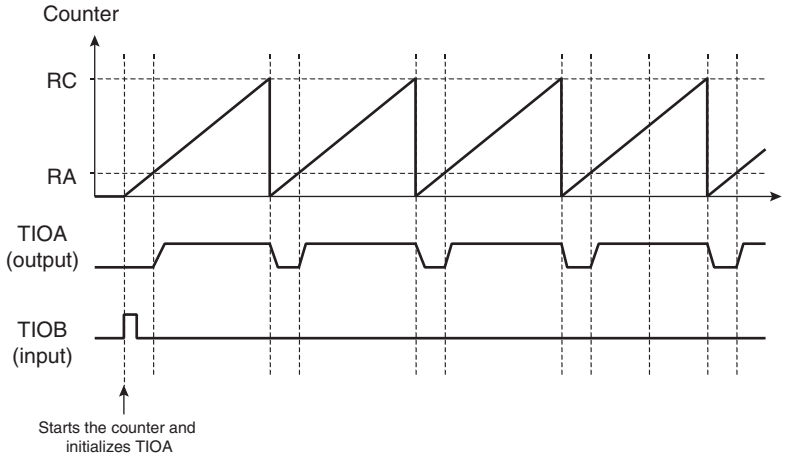


26.12.1.4 Trigger on TIOB Input Pin

In each of the previous examples, TIOBx is used as an output. It is possible to use it as a trigger input and generate only TIOAx output signal.

The following application is the same as ”Dual Pulse Width Modulation (PWM) Generation” on page 351, where TIOBx is not used as an output but as an input.

Figure 26-13. Single Waveform with Trigger on TIOB

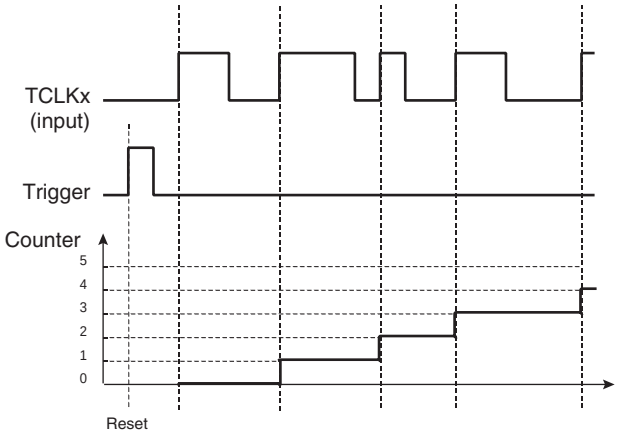


26.12.1.5 Event Counter on an External Clock (TCLK)

The counter is incremented with each TCLKx rising edge.

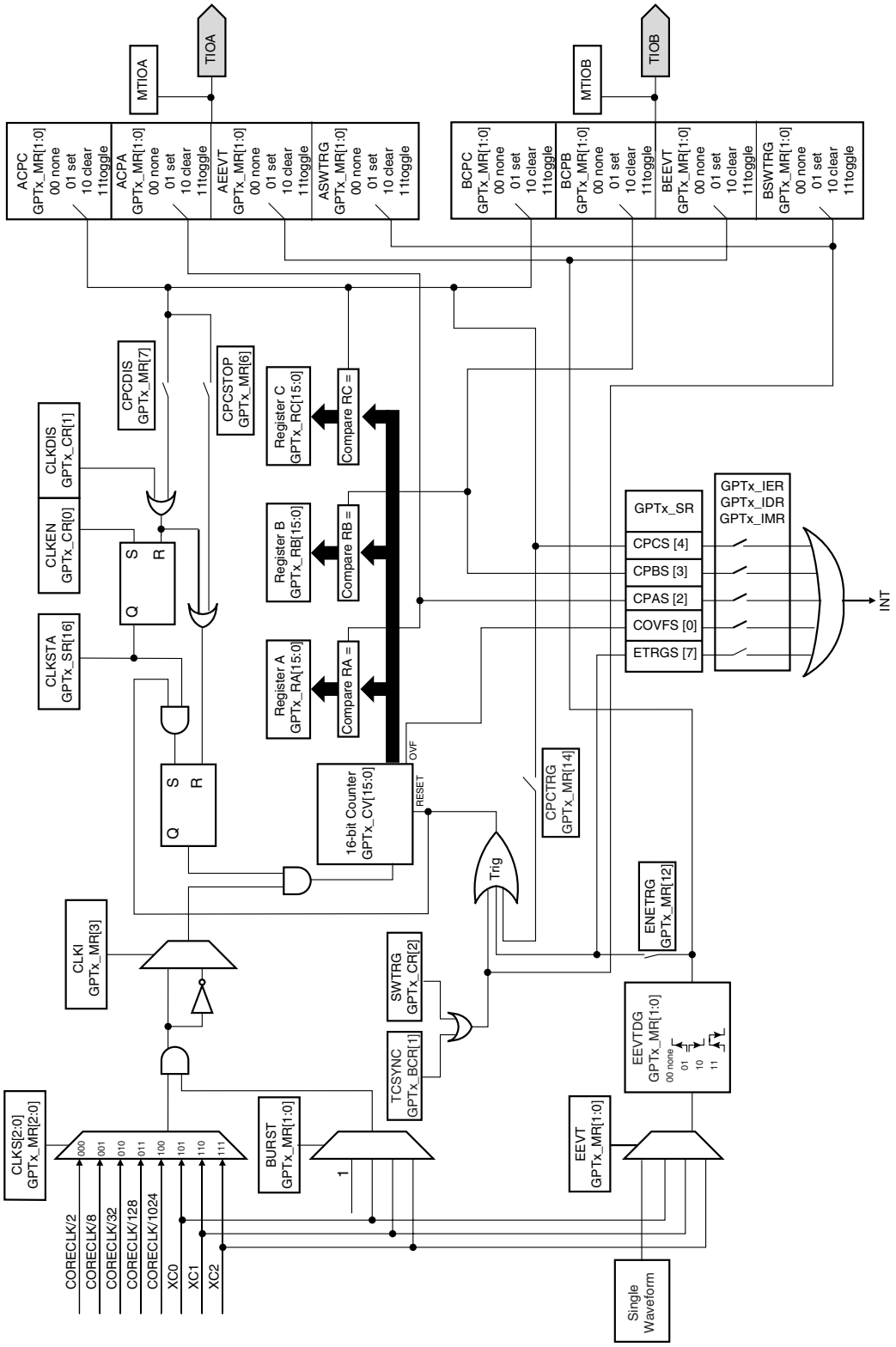
This application can be generated in capture mode.

Figure 26-14. Event Counter



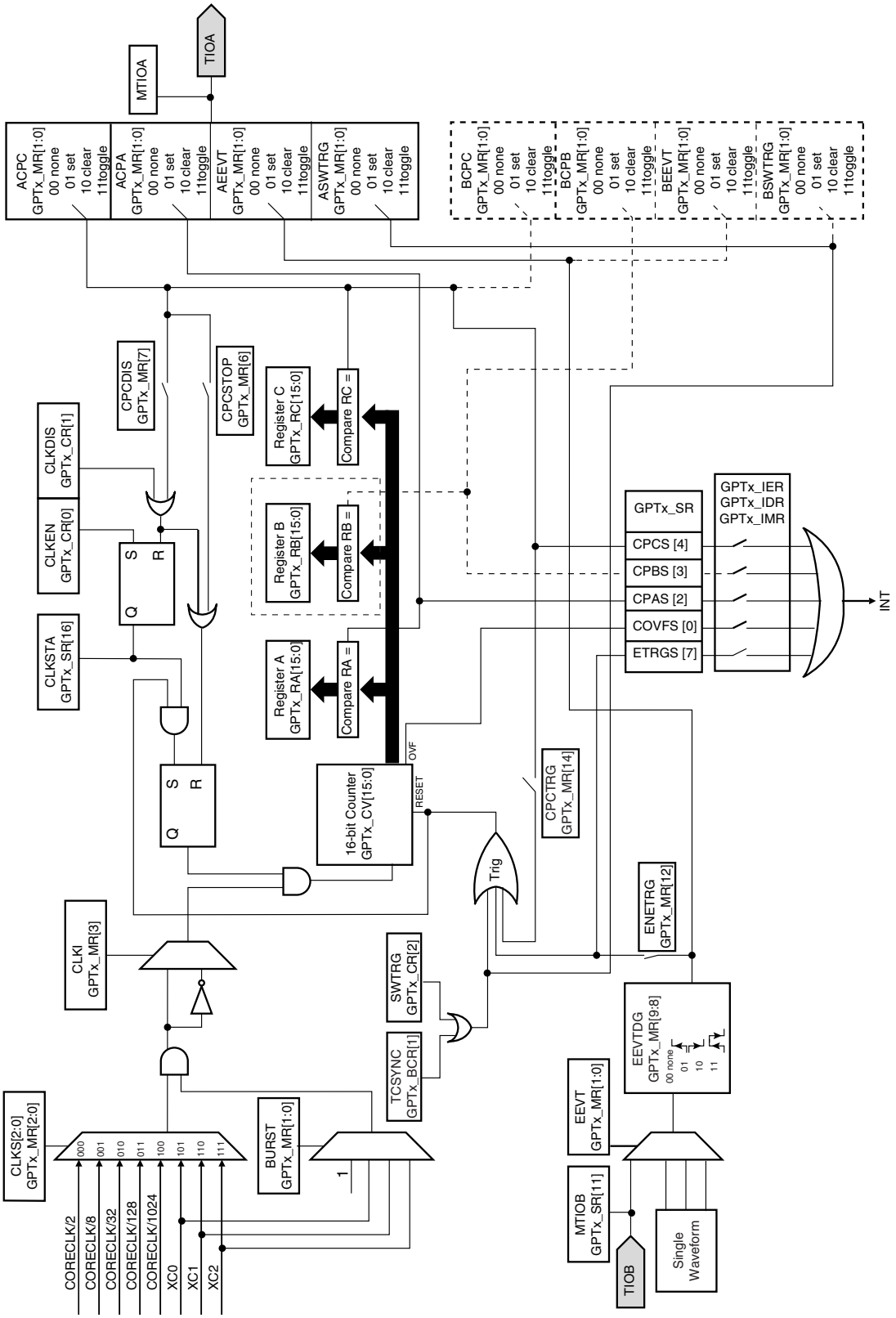
26.12.2 Dual Waveform Mode

Figure 26-15. Dual Waveform Mode



26.12.3 Single Waveform Mode

Figure 26-16. Single Waveform Mode (The dotted blocks are not used in this case.)



26.12.4 GPT Control Register in Waveform Mode

Name: GPT_CR
Access: Write-only
Offset: 0x60

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	SWRST

- **SWRST: Software Reset**

0: No effect.

1: Generates a software reset.

A software triggered hardware reset of the channel is performed. It reset all the registers, including PIO and PMC registers (except GPTX_PMSR).

- **CLKEN: Counter Clock Enable**

0: No effect.

1: Enables counter clock if CLKDIS = 0.

- **CLKDIS: Counter Clock Disable**

0: No effect.

1: Disables counter clock.

- **SWTRG: Software Trigger**

0: No effect.

1: Generates a software trigger.

This bit generates a software trigger for resetting and starting the counter at the next valid counter clock edge when the counter clock is enabled.

26.12.5 GPT Mode Register in Waveform Mode

Name: GPT_MR
Access: Read/Write
Offset: 0x64

31	30	29	28	27	26	25	24
BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
23	22	21	20	19	18	17	16
ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
15	14	13	12	11	10	9	8
WAVE = 1	CPCTRG	–	ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
7	6	5	4	3	2	1	0
CPCDIS	CPCSTOP	BURST[1:0]		CLKI	CLKS[2:0]		

- **CLKS[2:0]: Clock Select**

CLKS[2:0]			Counter Clock Source
0	0	0	CORECLK/2
0	0	1	CORECLK/8
0	1	0	CORECLK/32
0	1	1	CORECLK/128
1	0	0	CORECLK/1024
1	0	1	XC0
1	1	0	XC1
1	1	1	XC2

For more details, see ”Clock Sources” on page 317.

- **CLKI: Clock Inverter**

0: Normal clock (The counter is incremented on a rising edge)

1: Inverted clock (The counter is incremented on a falling edge)

- **BURST[1:0]: Burst**

This signal is combined with the selected clock through a logical AND.

BURST[1:0]		Burst signal selected
0	0	None
0	1	XC0
1	0	XC1
1	1	XC2

For more details, see ”Clock Sources” on page 317.

- **CPCSTOP: Compare RC Stops the Counter**

0: The counter is not stopped when an equal condition on RC is detected.

1: The counter is stopped when an equal condition on RC is detected.

If the counter is stopped, it can restart (to 0x0000) just with a trigger condition.

If an equal condition on RC induces both trigger condition and stop counter, the trigger will have no effect.

- **CPCDIS: Compare RC Disables Clock**

0: The counter clock is not disabled when an equal condition on RC is detected.

1: The counter clock is disabled and the counter stopped when an equal condition on RC is detected.

If the counter clock is disabled, it can be enabled only by asserting CLKEN, bit [1] of the control register.

- **EEVTEDG[1:0]: External Event Edge**

These two bits activate one of four possible external event modes. The external event source is selected by EEVT[1:0] of the mode register.

EEVTEDG[1:0]		Edge
0	0	None
0	1	Rising edge
1	0	Falling edge
1	1	Each edge

When an external event is generated, five events occur:

– The ETRGS flag is set in the status register.

– If enabled, ETRGS interrupt is generated.

– It can reset and start the counter at the next valid counter clock edge if ENETRG (bit [12] of the mode register) is high.

– TIOAx pin can be set, clear, toggle or unchanged following AEEVT[1:0] of the mode register.

– TIOBx pin can be set, clear, toggle or unchanged following BEEVT[1:0] of the mode register.

- **EEVT[1:0]: External Event**

These bits select an external event source among four pins:

EEVT[1:0]		External Trigger
0	0	TIOBx
0	1	XC0
1	0	XC1
1	1	XC2

If TIOBx is selected, the mode is in single waveform mode (see Figure 26-16 on page 355). TIOAx is used as an output and TIOBx as an input. The following bits are disabled:

– BSWTRG[1:0] of the mode register

– BEEVT[1:0] of the mode register

– BCPC[1:0] of the mode register

– BCPB[1:0] of the mode register

– Compare register B

If an external clock is selected, the mode is in dual waveform mode. TIOAx and TIOBx are used as outputs.

- **ENETRG: Enable External Trigger**

This bit determines whether an external event can be used as a trigger to reset and start the counter at the next valid counter clock edge. The external event source is selected by EEVT[1:0] of the mode register.

0: External event does not reset and start the counter. Selected external trigger can only be used to control TIOAx and TIOBx.

1: External trigger resets and starts the counter.

Note: The counter can start only if the clock is enabled.

- **CPCTRG: Compare RC Trigger**

This bit determines whether an equal condition on the Compare C register can cause a trigger (reset and start the counter at the next valid counter clock edge).

0: An equal condition on RC does not cause a trigger.

1: An equal condition on RC causes a trigger.

Note: The counter can start only if the clock is enabled.

- **WAVE: Waveform**

0: Capture mode.

1: Waveform mode.

- **ACPA: TIOA Compare A**

These two bits determine the effect on the TIOAx output pin caused by an equal comparison between the counter and the Compare Register A value. See the bit description table in “ASWTRG: TIOA Software Trigger” where xxx = CPA.

Note: If several events that control TIOAx output (set, clear or toggle) arrive at the same time, only one has an action according to the following priority order:
1. ASWTRG (highest priority)
2. AEEVT
3. ACPC
4. ACPA

- **ACPC: TIOA Compare C**

These two bits determine the effect on the TIOAx output pin caused by an equal comparison between the counter and the Compare register C value. See the bit description table in “ASWTRG: TIOA Software Trigger” where xxx = CPC.

- **AEEVT: TIOA External Event**

These two bits determine the effect on the TIOAx output pin caused by an external event. The external event source is selected by EEVT[1:0] of the mode register. See the bit description table in “ASWTRG: TIOA Software Trigger” where xxx = EEVT.

- **ASWTRG: TIOA Software Trigger**

These two bits determine the effect on the TIOAx output pin caused by a software trigger.

See the following table where xxx = SWTRG.

Axxx		Waveform Pin
0	0	None
0	1	Set
1	0	Clear
1	1	Toggle

- **BCPB: TIOB Compare B**

These two bits determine the effect on the TIOBx output pin caused by an equal comparison between the counter and the Compare Register B value. These bits are active only if TIOBx is not an input (see “EEVT[1:0]: External Event” of the ”GPT Mode Register in Waveform Mode” on page 357). See the bit description table in “BSWTRG: TIOB Software Trigger” where xxx = CPB.

Note: If several events that control TIOBx output (set, clear or toggle) arrive at the same time, only one has an action according to the following priority:
1. BSWTRG (highest priority)
2. BEEVT
3. BCPC
4. BCPB

- **BCPC: TIOB Compare C**

These two bits determine the effect on the TIOBx output pin caused by an equal comparison between the counter and the Compare register C value. These bits are active only if TIOBx is not an input (see “EEVT[1:0]: External Event” of the ”GPT Mode Register in Waveform Mode” on page 357). See the bit description table in “BSWTRG: TIOB Software Trigger” where xxx = CPC.

- **BEEVT: TIOB External Event**

These two bits determine the effect on the TIOBx output pin caused by an external event. The external event source is selected by EEVT[1:0] of the mode register. These bits are active only if TIOBx is not an input (see “EEVT[1:0]: External Event” of the ”GPT Mode Register in Waveform Mode” on page 357). See the bit description table in “BSWTRG: TIOB Software Trigger” where xxx = EEVT.

- **BSWTRG: TIOB Software Trigger**

These two bits determine the effect on the TIOBx output pin caused by a software trigger. These bits are active only if TIOBx is not an input (see “EEVT[1:0]: External Event” of the ”GPT Mode Register in Waveform Mode” on page 357).

See following table where xxx = SWTRG:

Bxxx		Waveform Pin
0	0	None
0	1	Set
1	0	Clear
1	1	Toggle

26.12.6 GPT Status Register in Waveform Mode

Name: GPT_SR
Access: Read-only
Offset: 0x70

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	TCLKS	TIOAS	TIOBS
15	14	13	12	11	10	9	8
–	–	–	–	–	MTIOB	MTIOA	CLKSTA
7	6	5	4	3	2	1	0
ETRGS	–	–	CPCS	CPBS	CPAS	–	COVFS

Note: This register is a “read-active” register; thus, reading it can affect the state of some bits. When reading GPT_SR register, following bits are cleared if set: COVFS, CPAS, CPBS, CPCS, ETRGS, TIOBS, TIOAS and TCLKS.

- **COVFS: Counter Overflow Status**

This bit is set when a counter overflow is detected. An overflow occurs when the counter reaches its maximal value 0xFFFF (2^{16}-1) and passes to 0x0000.

0: No overflow detected

1: Overflow detected since last read of GPTX_SR

- **CPAS: Compare Register A Status**

This bit is set when the counter reaches the register A value.

0: Compare A condition has not occurred since last read of GPTX_SR

1: Compare A condition has occurred since last read of GPTX_SR

- **CPBS: Compare Register B Status**

This bit is set when the counter reaches the register B value.

0: Compare B condition has not occurred since last read of GPTX_SR

1: Compare B condition has occurred since last read of GPTX_SR

- **CPCS: Compare Register C Status**

This bit is set when the counter reaches the register C value.

0: Compare C condition has not occurred since last read of GPTX_SR

1: Compare C condition has occurred since last read of GPTX_SR

- **ETRGS: External Trigger Status**

This bit is set when an external trigger is detected. An external trigger occurs with a valid edge (the edge polarity is set by EEVTEDG[1:0] of the mode register) on the valid trigger pin (set by EEVT[1:0] of the mode register if ENETRG, bit 12 of the Mode Register, is high).

0: External trigger not detected

1: External trigger detected since last read of GPTX_SR

- **CLKSTA: Clock Status**

0: Clock disabled

1: Clock enabled

- **MTIOA: TIOA Mirror**

This bit reflects the TIOAx pin value.

Its reset value is undefined because the operating mode after hardware reset is capture mode.

- **MTIOB: TIOB Mirror**

This bit reflects the TIOBx pin value.

Its reset value is undefined because the operating mode after hardware reset is capture mode.

- **TIOBS: TIOB Status**

0: At least one input change has been detected on the pin TIOBx since the register was last read.

1: No input change has been detected on the TIOBx pin since the register was last read.

- **TIOAS: TIOA Status**

0: At least one input change has been detected on the TIOAx pin since the register was last read.

1: No input change has been detected on the TIOAx pin since the register was last read.

- **TCLKS: TCLK Status**

0: At least one input change has been detected on the TCLKx pin since the register was last read.

1: No input change has been detected on the TCLKx pin since the register was last read.

Note: x: Channel number

26.12.7 GPT Interrupt Enable Register in Waveform Mode

Name: GPT_IER
Access: Write-only
Offset: 0x74

26.12.8 GPT Interrupt Disable Register in Waveform Mode

Name: GPT_IDR
Access: Write-only
Offset: 0x78

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	TCLKS	TIOAS	TIOBS
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ETRGS	–	–	CPCS	CPBS	CPAS	–	COVFS

- **COVFS: Counter Overflow Status**

0: COVFS interrupt is disabled.

1: COVFS interrupt is enabled.

- **CPAS: Compare Register A Status**

0: CPAS interrupt is disabled.

1: CPAS interrupt is enabled.

- **CPBS: Compare Register B Status**

0: CPBS interrupt is disabled.

1: CPBS interrupt is enabled.

- **CPCS: Compare Register C Status**

0: CPCS interrupt is disabled.

1: CPCS interrupt is enabled.

- **ETRGS: External Trigger Status**

0: ETRGS interrupt is disabled.

1: ETRGS interrupt is enabled.

- **TIOBS: TIOB Status**

0: TIOBS interrupt is disabled.

1: TIOBS interrupt is enabled.

- **TIOAS: TIOA Status**

0: TIOAS interrupt is disabled.

1: TIOAS interrupt is enabled.

- **TCLKS: TCLK Status**

0: TCLKS interrupt is disabled.

1: TCLKS interrupt is enabled.

26.12.9 GPT Interrupt Mask Register in Waveform Mode

Name: GPT_IMR
Access: Read-only
Offset: 0x7C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	TCLKS	TIOAS	TIOBS
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ETRGS	–	–	CPCS	CPBS	CPAS	–	COVFS

- **COVFS: Counter Overflow Status**

0: COVFS interrupt is disabled.

1: COVFS interrupt is enabled.

- **CPAS: Compare Register A Status**

0: CPAS interrupt is disabled.

1: CPAS interrupt is enabled.

- **CPBS: Compare Register B Status**

0: CPBS interrupt is disabled.

1: CPBS interrupt is enabled.

- **CPCS: Compare Register C Status**

0: CPCS interrupt is disabled.

1: CPCS interrupt is enabled.

- **ETRGS: External Trigger Status**

0: ETRGS interrupt is disabled.

1: ETRGS interrupt is enabled.

- **TIOBS: TIOB Status**

0: TIOBS interrupt is disabled.

1: TIOBS interrupt is enabled.

- **TIOAS: TIOA Status**

0: TIOAS interrupt is disabled.

1: TIOAS interrupt is enabled.

- **TCLKS: TCLK Status**

0: TCLKS interrupt is disabled.

1: TCLKS interrupt is enabled.

26.12.10 GPT Counter Value in Waveform Mode

Name: GPT_CV
Access: Read-only
Offset: 0x80

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[15:8]							
7	6	5	4	3	2	1	0
CV[7:0]							

- **CV[15:0]: Counter Value**

These 16 bits contain the counter value in real time.

The maximal counter value is 0xFFFF = 2^{16} - 1 = 65535.

When a trigger occurs, the counter will be reset to 0x0000 at the next valid counter clock edge.

26.12.11 GPT Register A in Waveform Mode

Name: GPT_RA
Access: Read/Write
Offset: 0x84

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[15:8]							
7	6	5	4	3	2	1	0
RA[7:0]							

- **RA[15:0]: Register A Value**

When the counter reaches this value, three events occur:

– The CPAS flag is set in the status register.

– If enabled, CPAS interrupt is generated.

– TIOAx pin can be set, clear, toggle or unchanged following bits ACPA[1:0] of the mode register.

26.12.12 GPT Register B in Waveform Mode

Name: GPT_RB
Access: Read/Write
Offset: 0x88

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[15:8]							
7	6	5	4	3	2	1	0
RB[7:0]							

- **RB[15:0]: Register B Value**

When the counter reaches this value, three events occur:

– The CPBS flag is set in the status register.

– If enabled, CPBS interrupt is generated.

– TIOBx pin can be set, clear, toggle or unchanged following bits BCPB[1:0] of the mode register.

These bits are active only if TIOB is not an input (see “EEVT[1:0]: External Event” of the ”GPT Mode Register in Waveform Mode” on page 357).

26.12.13 GPT Register C in Waveform Mode

Name: GPT_RC
Access: Read/Write
Offset: 0x8C

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[15:8]							
7	6	5	4	3	2	1	0
RC[7:0]							

- **RC[15:0]: Register C Value**

When the counter reaches this value, seven events can occur:

– The CPCS flag is set in the status register.
– If enabled, CPCS interrupt is generated.
– If bit CPCTRG (bit [14] of the GPTX_MR) is high, the counter is reset and restarts at 0x0000 at the next valid counter clock edge.
– The counter clock can be disabled according to CPCDIS (bit [7] of the mode register).
– The counter can be stopped according to CPCSTOP (bit [6] of the mode register).
– TIOAx pin can be set, clear, toggle or unchanged following bits ACPC[1:0] of the mode register.
– TIOBx pin can be set, clear, toggle or unchanged following bits BCPC[1:0] of the mode register.

26.13 Timer Controller Block Programming

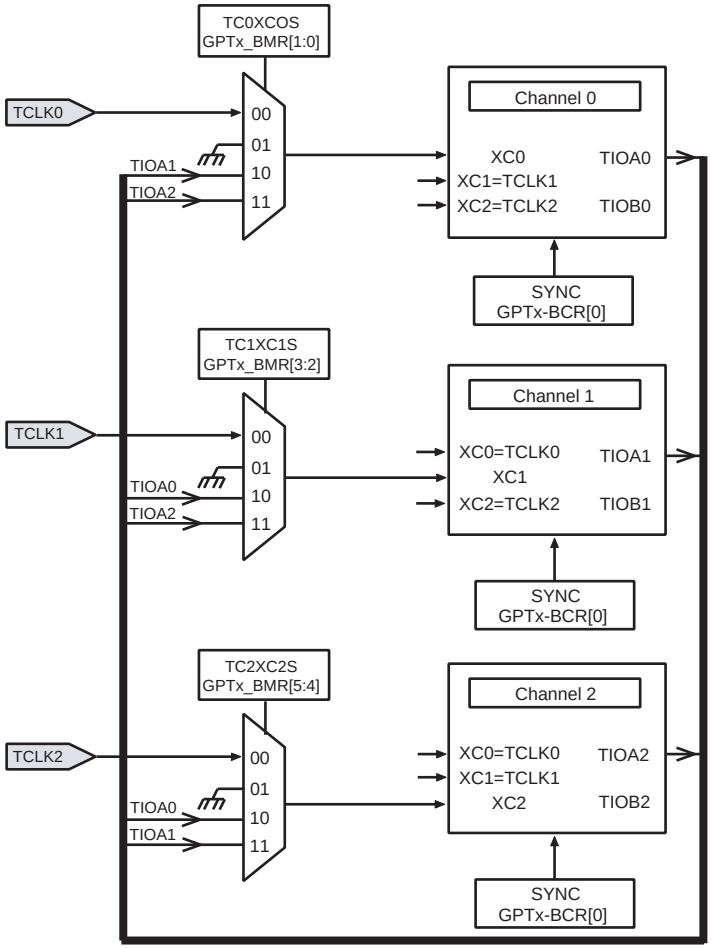
This module controls the entire timer controller with its three channels.

It has two functions:

- The block control register provides the means to synchronize the three timer channels. It can generate a software trigger on all three channels at exactly the same time.
- The block mode register provides the means to daisy chain two or three channels. Thus the user can improve counter capacity.

Figure 26-17 shows the block diagram.

Figure 26-17. Timer Controller Block Programming Diagram



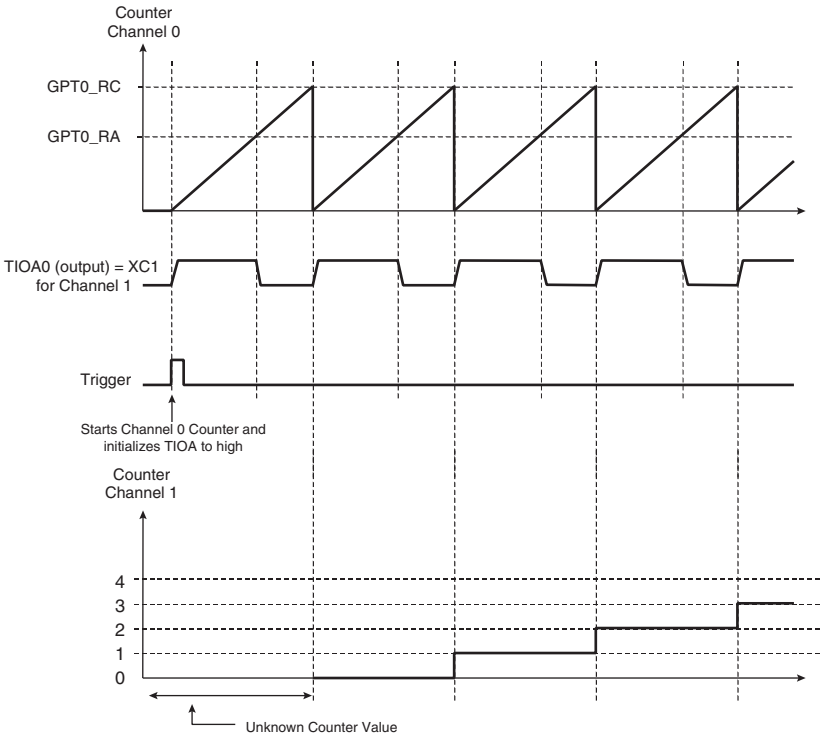
26.13.1 External Clock Generation for Channel 1 Using Channel 0

With GPT0_RA and GPT0_RC, Channel 0 generates a pulse width modulation output (PWM) in waveform mode that is used as an external clock by Channel 1.

Note that if RC is not used, this generates a 32-bit counter.

Each time the counter 0 passes 0xFFFF (overflow condition), this increments the counter by 1.

Figure 26-18. Timer Controller Block Programming Application



26.13.2 GPT Block Control Register

Name: GPT_BCR
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
–	–	–	–	–	–	TCSYNC	SWRST

- **SWRST: Software Reset**

This bit generates a software reset on the three timer channels simultaneously.

0: No effect.

1: Generates a software reset.

- **TCSYNC: Synchronization Bit**

This bit generates a software trigger on the three channels of the general-purpose timer simultaneously.

A software trigger resets and starts the counter at the next valid counter clock edge.

0: No effect.

1: Resets and starts all three timer channel counters simultaneously.

26.13.3 GPT Block Mode Register

Name: GPT_BMR
Access: Read/Write
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
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	TC2XC2S		TC1XC1S		TC0XC0S	

- **TC0XC0S[1:0]: TCLK0 XC0 Selection**

These bits select the external clock XC0 source for the channel 0.

TC0XC0S[1:0]		Selected Signal
0	0	TCLK0
0	1	none
1	0	TIOA1
1	1	TIOA2

- **TC1XC1S[1:0]: TCLK1 XC1 Selection**

These bits select the external clock XC1 source for the channel 1.

TC1XC1S[1:0]		Selected Signal
0	0	TCLK1
0	1	none
1	0	TIOA0
1	1	TIOA2

- **TC2XC2S[1:0]: TCLK2 XC2 Selection**

These bits select the external clock XC2 source for the channel 2.

TC2XC2S[1:0]		Selected Signal
0	0	TCLK2
0	1	none
1	0	TIOA0
1	1	TIOA1

26.14 Timer Test Controller Block

The test mode is used to increase the fault coverage of the timer. This mode is entered by setting SFM registers (see ”Special Function Mode (SFM)” on page 70). When the test mode is entered, two registers are available, GPT_TSTC and GPT_TSTM.

The register GPT_TSTC is used to load the counter of each timer with the value 0xFFF0.

The register GPT_TSTM outputs the counter clock enable on the TIOB output for each channel. For this, it is necessary to put the timer in dual waveform mode and set the bit corresponding to the channel in the GPT_TSTM (for example, 0x00000004 for the channel 2).

26.14.1 GPT Test Control Register in Test Mode

Name: GPT_TSTC
Access: Write-only
Offset: 0x40

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
–	–	–	–	–	LDCT2	LDCT1	LDCT0

- **LDCT0: Load Timer 0 Counter**

This bit generates a load counter with the value 0xFFFE.

- **LDCT1: Load Timer 1 Counter**

This bit generates a load counter with the value 0xFFFE.

- **LDCT2: Load Timer 2 Counter**

This bit generates a load counter with the value 0xFFFE.

26.14.2 GPT Test Control Register in Test Mode

Name: GPT_TSTC
Access: Write-only
Offset: 0x40

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
–	–	–	–	–	OCLKEN2	OCLKEN1	OCLKEN0

- **OCLKEN0: Out Clock Counter Enable for the Timer 0**

This bit enables to output the clock counter on TIOB0.

- **OCLKEN1: Out Clock Counter Enable for the Timer 1**

This bit enables to output the clock counter on TIOB1.

- **OCLKEN2: Out Clock Counter Enable for the Timer 2**

This bit enables to output the clock counter on TIOB2.

27. AT91SAM7A1 Electrical Characteristics

27.1 Core and 3V3 I/O Power Supply

Applicable over recommended operating temperature and voltage range.

Table 27-1. Core and 3V3 I/O Power Supply

Symbol	Parameter	Min	Typ	Max	Unit
VDD3V(CORE)	Core supply voltage	3.0	3.3	3.6	V
VDD3V(PLL)	Supply voltage for PLL and master oscillator	3.0	3.3	3.6	V
VDD3V(I/O)	3V I/O supply voltage	3.0	3.3	3.6	V
VDD3V (I/O+CORE)	3V I/O and core supply voltage	3.0	3.3	3.6	V
VDD3V(RTCK)	Supply voltage for RTCK oscillator	3.0	3.3	3.6	V
VIH	High-level input voltage	0.7 x VDD3V		VDD3V + 0.3	V
VIL	Low-level input voltage	-0.3		0.3 x VDD3V	V
VIHST	High-level input voltage (Schmitt trigger)	1.8		2.265	V
VILST	Low-level input voltage (Schmitt trigger)	1.0		1.145	V
VOH	High-level output voltage (drive = 0.3 mA)	VDD3V - 0.1			V
VOL	Low-level output voltage (drive = 0.3 mA)			VSS + 0.1	V
ILEAK	Input leakage current		90		nA
OLEAK	Output leakage current		100		nA
FANIN	Pad capacitance		5.75		pF
IPD	Internal pull-down current	99		429	µA
IPU	Internal pull-up current	130		352	µA

Note: In CMOS, the behavior of a cell is independent of its load as loads are purely capacitive.

27.2 5V I/O Power Supply

Table 27-2. 5V I/O Power Supply

Symbol	Parameter	Min	Typ	Max	Unit
VDD5V(I/O)	Supply voltage for 3V or 5V I/Os	VDD3V		5.5	V
VIH	High-level input voltage	2.0		VDD5V + 0.3	V
VIL	Low-level input voltage	-0.3		0.8	V
VOH	High-level output voltage (drive = pin output current)	VDD5V - 0.4			V
VOL	Low-level output voltage (drive = pin output current)			0.4	V
ILEAK	Input leakage current		90		nA
OLEAK	Output leakage current		100		nA
FANIN	Pad capacitance		5.75		pF

Note: In CMOS, the behavior of a cell is independent of its load as loads are purely capacitive.

27.3 Analog Power Supply

Table 27-3. Analog Power Supply

Symbol	Parameter	Min	Typ	Max	Unit
VDD3V(ANA)	Supply voltage for ADC	3.0		3.6	V
Vana	High-level input voltage	VSS		VDDANA	V
VREFP	Low-level input voltage	2.4		VDDANA	V
ILEAK	Input leakage current	-100		+100	nA
IMAXP	Maximum current per pad (peak @100°C)		150		mA
IMAXR	Maximum current per pad (RMS @100°C)		50		mA
FANIN	Pad capacitance		5.73		pF

27.4 Analog-to-digital Converter Characteristics

Table 27-4. Analog-to-digital Converter Characteristics (Temperature Range: [-40ºC to +85ºC], worst cases of VDDANA & process, unless otherwise noted)

Code	Parameter	Conditions/Details	Min	Typ	Max	Unit
	Resolution			10		bit
VDDANA	Supply voltage		3.0	3.3	3.6	V
ΔVDDANA	Supply ripple	rms value, 10 kHz to 10 MHz			30	mV
IDDANA on	Current consumption (not taking into account Vref current)	onad = 1	40	150	400	µA
IDDANA standby	Standby current consumption	onad = 0			10	µA
VREFP	Positive reference voltage range		2.4	3.0	VDD	V
	Vref input resistance	@ 25° C	12	18	24	kOhm
	Conversion time	1 for sample, 10 for conversion	11			clk
	Clock frequency (1/Tclk)	For specified duty cycle range			600	kHz
	Clock duty cycle	Up to max. clock frequency (20% duty cycle = 20% high - 80%)	20	50	60	%
	Input capacitance [1]	Switched during sampling process	0	80	90	pF
	Input capacitance (including sample and hold)	Pad selected		106		pF
	Integral non-linearity	3.0 ≤VDDANA ≤3.6V, 3.0 ≤VREFP ≤VDDANA, Clk ≤250kHz, -40° ≤Temp. ≤85° C, Best Fit Straight Line	-1	-0.5/+0.7	1	lsb
	Differential non-linearity		-0.9	-0.3/+0.6	1	lsb
	Offset		+3	+9	+15	mV
	Gain error		-6	-3	0	mV
	Integral non-linearity	3.0 ≤VDDANA ≤3.6V, 2.4 ≤VREFP ≤VDDANA, Clk ≤600kHz, -40° ≤Temp. ≤85° C, Best Fit Straight Line	-1.5		1.5	lsb
	Differential non-linearity		0.999		1.5	lsb
	Offset (intrinsic)		+3	+9	+15	mV
	Gain error		-6	-3	0	mV
	Startup time				4	µs

Note: 1. Capacitor on pin does not take into account any pad and/or connection to pad capacitance.

28. Packaging Information

28.1 Thermal Considerations

The heat transfer between the top surface of the die and the surrounding ambient air can be characterized by the following equation:

$$T_J - T_A = P \times \theta_{JA}$$

where:

P = Device operating power (in W)

T_J = Temperature of a junction on the device (in °C)

T_A = Temperature of the surrounding ambient air (in °C)

θ_{JA} = Package thermal resistance i.e. between junction and ambient air (in °C/W)

Table 28-1. Package Thermal Resistance Data

Package	**Θ_{JA}(in °C/W)**
LQFP	37

28.2 Package Drawings

28.2.1 LQFP Package Drawing

Figure 28-1. 144-pin LQFP Version A (Foundry Reference)

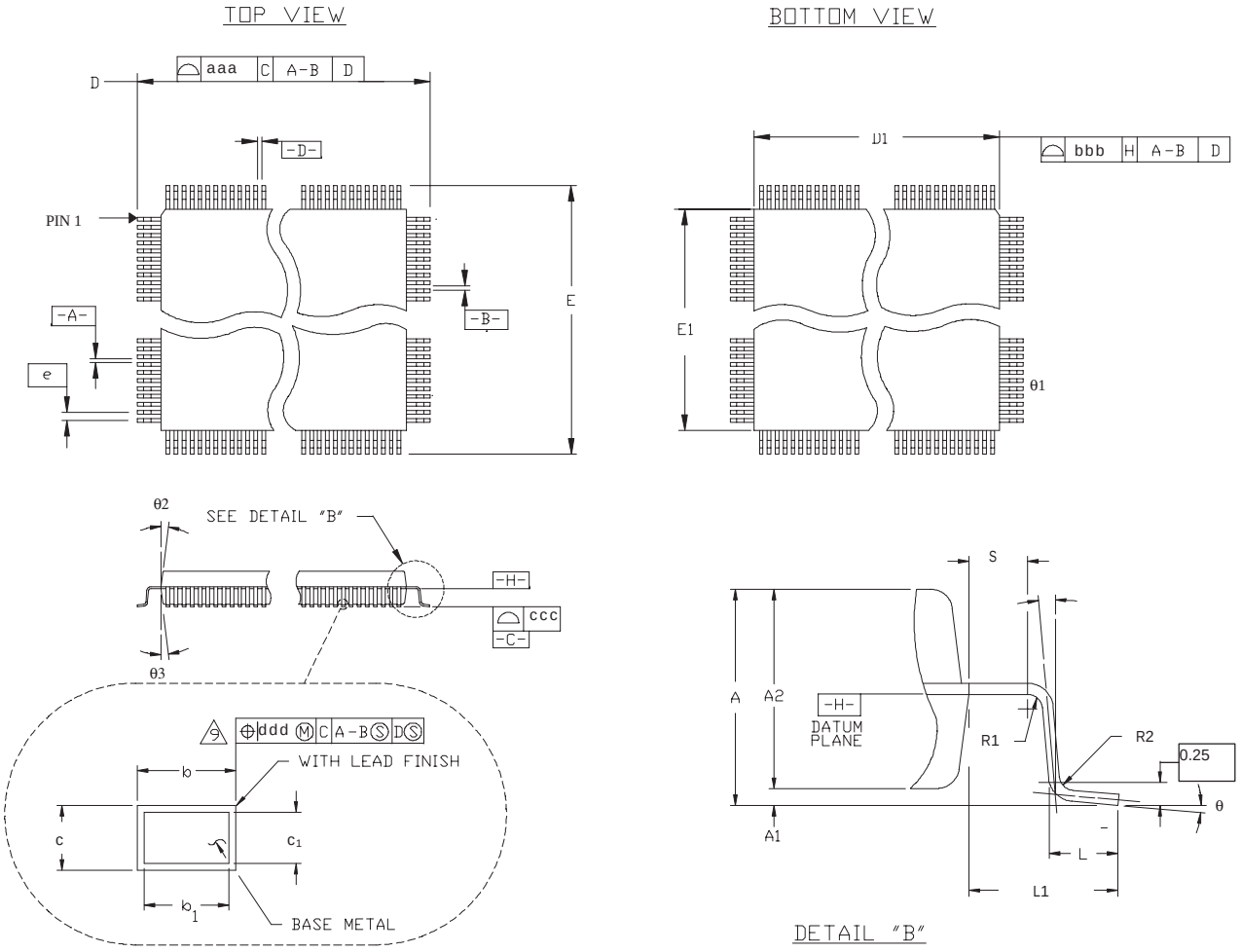


Table 28-2. Package Dimensions in mm

Symbol	Min	Nom	Max
c	0.09		0.2
c1	0.09		0.16
L	0.45	0.6	0.75
L1		1.00 REF	
R2	0.08		0.2
R1	0.08		
S	0.2		
q	0°	3.5°	7°
θ1	V		

Table 28-2. Package Dimensions in mm (Continued)

Symbol	Min	Nom	Max
θ2	11°	12°	13°
θ3	11°	12°	13°
A			1.6
A1	0.05		0.15
A2	1.35	1.4	1.45
D/E		22.0	
D1/E1		20.0	
b	0.17	0.22	0.27
b1	0.17	0.2	0.23
e		0.50	
ccc		0.10	
ddd		0.08	
Tolerance of form and position			
aaa		0.2	
bbb		0.2	

29. Soldering Profile

Table 29-1 below gives the recommended soldering profile from J-STD-20.

Table 29-1. Soldering Profile

	Convection or IR/Convection	VPR
Average Ramp-up Rate (183° C to Peak)	3° C/sec. max	10° C/sec.
Preheat Temperature 125° C ±25° C	120 sec. max	
Temperature Maintained Above 183° C	60 sec. to 150 sec.	
Time within 5° C of Actual Peak Temperature	10 sec. to 20 sec.	60 sec.
Peak Temperature Range	220 +5/-0° C or 235 +5/-0° C	215 to 219° C or 235 +5/-0° C
Ramp-down Rate	6° C/sec.	10° C/sec.
Time 25° C to Peak Temperature	6 min. max	

Small packages may be subject to higher temperatures if they are reflowed in boards with larger components. In this case, small packages may have to withstand temperatures of up to 235° C, not 220° C (IR reflow).

Recommended package reflow conditions depend on package thickness and volume. See Table 29-2 below.

Table 29-2. Recommended Package Reflow Conditions [(1)(2)(3)]

Parameter	Measure
Convection	220 + 5/-0° C
VPR	215 to 219° C
IR/Convection	220 +5/-0° C

Notes:
1. The packages are qualified by Atmel by using IR reflow conditions, not convection or VPR.
2. By default, the package level 1 is qualified at 220°C (unless 235°C is stipulated).
3. The body temperature is the most important parameter but other profile parameters such as total exposure time to hot temperature or heating rate may also influence component reliability.
4. A maximum of three reflow passes is allowed per component.

30. Ordering Information

Table 30-1. AT91SAM7A1 Ordering Information

Ordering Code	Package	Package Type	Temperature Operating Range
AT91SAM7A1-AU	LQFP144	Green	Industrial (-40° C to +85° C)

Revision History

Doc. Rev.	Date	Comments	Change Request Ref.
6048A	03-Mar-05	First issue.	
6048B	15-May-06/ 27-Jun-06	Changed “Measure between negative edges” in ”MEASMODE[1:0]: Measurement Mode” on page 190	2553
		Removed references to automotive application. Global ”Features” on page 1, ”Description” on page 2, ”Controller Area Network (CAN)” on page 269 and ”Oscillator Tolerance” on page 286	2739
		Corrected base addresses to reflect 32 bits ”CM Master Clock Divider Register” on page 69, ”WT Status Register” on page 95, ”Special Function Mode (SFM) Memory Map” on page 71 and in the SFM registers that follow.	551
		Offset address changed to base address in ”SFM Extended Chip ID Register” on page 73 and GIC registers page 122 to page 125	552
		Corrected base address ”PWM Status Register” on page 220,	553
		GIC_IPR Reset value changed to undetermined in Table 16-2, “GIC Memory Map,” on page 117. Added Section 30. ”Ordering Information” on page 381. In the Memory Map tables and Register descriptions: “Base Address”, “Address” and “Offset” rubrics conform to IP user interface.	Marcom review

Table of Contents

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

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

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

6048B–ATARM–29-Jun-06