Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M4
Core Size	32-Bit Single-Core
Speed	120MHz
Connectivity	CANbus, Ethernet, IrDA, MMC/SD, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	79
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 16x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TFBGA
Supplier Device Package	100-TFBGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4e16cb-cnr

15.6.6 RTC Calendar Alarm Register

Name: RTC_CALALR

Address: 0x400E1874

Access: Read/Write

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

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Note: To change one of the DATE, MONTH fields, it is recommended to disable the field before changing the value and then re-enable it after the change has been made. This requires up to three accesses to the RTC_CALALR. The first access clears the enable corresponding to the field to change (DATEEN, MTHEN). If the field is already cleared, this access is not required. The second access performs the change of the value (DATE, MONTH). The third access is required to re-enable the field by writing 1 in DATEEN, MTHEN fields.

- **MONTH: Month Alarm**

This field is the alarm field corresponding to the BCD-coded month counter.

- **MTHEN: Month Alarm Enable**

0: The month-matching alarm is disabled.

1: The month-matching alarm is enabled.

- **DATE: Date Alarm**

This field is the alarm field corresponding to the BCD-coded date counter.

- **DATEEN: Date Alarm Enable**

0: The date-matching alarm is disabled.

1: The date-matching alarm is enabled.

16.5.3 Watchdog Timer Status Register

Name: WDT_SR

Address: 0x400E1858

Access Read-only

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
–	–	–	–	–	–	WDERR	WDUNF

- **WDUNF: Watchdog Underflow (cleared on read)**

0: No watchdog underflow occurred since the last read of WDT_SR.

1: At least one watchdog underflow occurred since the last read of WDT_SR.

- **WDERR: Watchdog Error (cleared on read)**

0: No watchdog error occurred since the last read of WDT_SR.

1: At least one watchdog error occurred since the last read of WDT_SR.

19.3 General Purpose Backup Registers (GPBR) User Interface

Table 19-1. Register Mapping

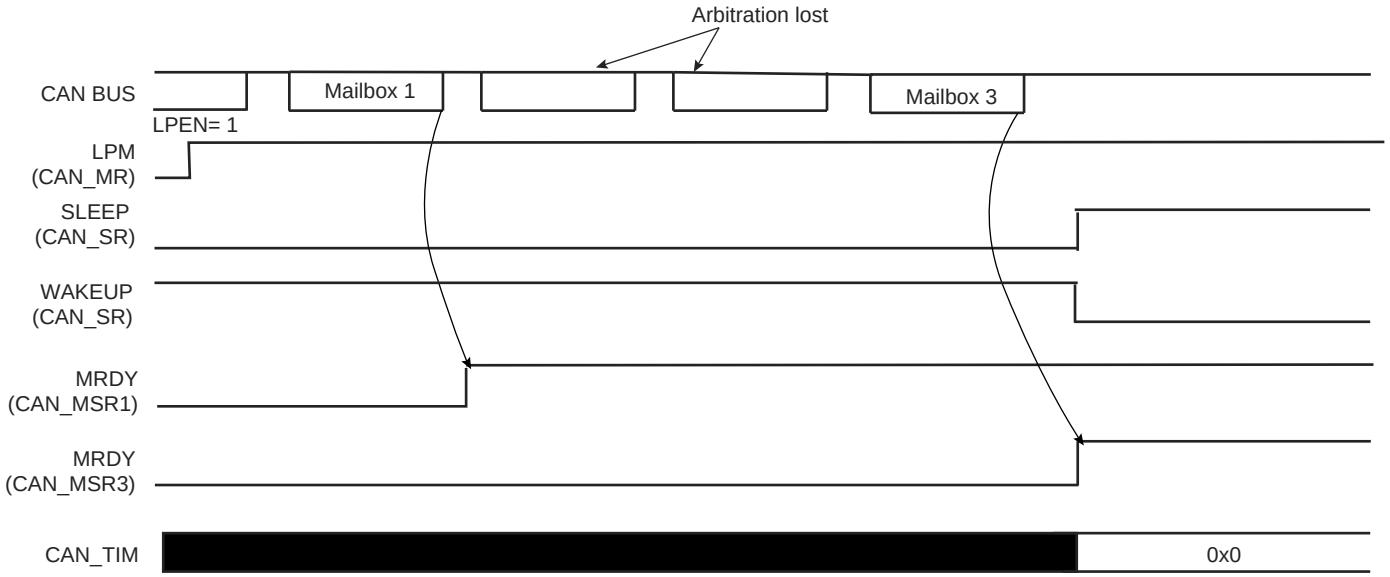
Offset	Register	Name	Access	Reset
0x0	General Purpose Backup Register 0	SYS_GPBR0	Read/Write	0x00000000
...	...	...	...	...
0x64	General Purpose Backup Register 19	SYS_GPBR19	Read/Write	0x00000000

25.8 DMA Controller (DMAC) User Interface

Table 25-4. Register Mapping

Offset	Register	Name	Access	Reset
0x000	DMAC Global Configuration Register	DMAC_GCFG	Read/Write	0x10
0x004	DMAC Enable Register	DMAC_EN	Read/Write	0x0
0x008	DMAC Software Single Request Register	DMAC_SREQ	Read/Write	0x0
0x00C	DMAC Software Chunk Transfer Request Register	DMAC_CREQ	Read/Write	0x0
0x010	DMAC Software Last Transfer Flag Register	DMAC_LAST	Read/Write	0x0
0x014	Reserved	–	–	–
0x018	DMAC Error, Chained Buffer Transfer Completed Interrupt and Buffer Transfer Completed Interrupt Enable Register	DMAC_EBCIER	Write-only	–
0x01C	DMAC Error, Chained Buffer Transfer Completed Interrupt and Buffer Transfer Completed Interrupt Disable Register	DMAC_EBCIDR	Write-only	–
0x020	DMAC Error, Chained Buffer Transfer Completed Interrupt and Buffer transfer completed Mask Register	DMAC_EBCIMR	Read-only	0x0
0x024	DMAC Error, Chained Buffer Transfer Completed Interrupt and Buffer transfer completed Status Register	DMAC_EBCISR	Read-only	0x0
0x028	DMAC Channel Handler Enable Register	DMAC_CHER	Write-only	–
0x02C	DMAC Channel Handler Disable Register	DMAC_CHDR	Write-only	–
0x030	DMAC Channel Handler Status Register	DMAC_CHSR	Read-only	0x00FF0000
0x034–0x038	Reserved	–	–	–
0x03C+ch_num*(0x28)+(0x0)	DMAC Channel Source Address Register	DMAC_SADDR	Read/Write	0x0
0x03C+ch_num*(0x28)+(0x4)	DMAC Channel Destination Address Register	DMAC_DADDR	Read/Write	0x0
0x03C+ch_num*(0x28)+(0x8)	DMAC Channel Descriptor Address Register	DMAC_DSCR	Read/Write	0x0
0x03C+ch_num*(0x28)+(0xC)	DMAC Channel Control A Register	DMAC_CTRLA	Read/Write	0x0
0x03C+ch_num*(0x28)+(0x10)	DMAC Channel Control B Register	DMAC_CTRLB	Read/Write	0x0
0x03C+ch_num*(0x28)+(0x14)	DMAC Channel Configuration Register	DMAC_CFG	Read/Write	0x01000000
0x03C+ch_num*(0x28)+(0x18)	Reserved	–	–	–
0x03C+ch_num*(0x28)+(0x1C)	Reserved	–	–	–
0x03C+ch_num*(0x28)+(0x20)	Reserved	–	–	–
0x03C+ch_num*(0x28)+(0x24)	Reserved	–	–	–
0x1E4	DMAC Write Protection Mode Register	DMAC_WPMR	Read/Write	0x0
0x1E8	DMAC Write Protection Status Register	DMAC_WPSR	Read-only	0x0
0x1EC–0x1FC	Reserved	–	–	–

Figure 31-8. Enabling Low-power Mode



31.7.5.2 Disabling Low-power Mode

The CAN controller can be awake after detecting a CAN bus activity. Bus activity detection is done by an external module that may be embedded in the chip. When it is notified of a CAN bus activity, the software application disables Low-power mode by programming the CAN controller.

To disable Low-power mode, the software application must:

- Enable the CAN Controller clock. This is done by programming the Power Management Controller (PMC).
- Clear the LPM field in the CAN_MR

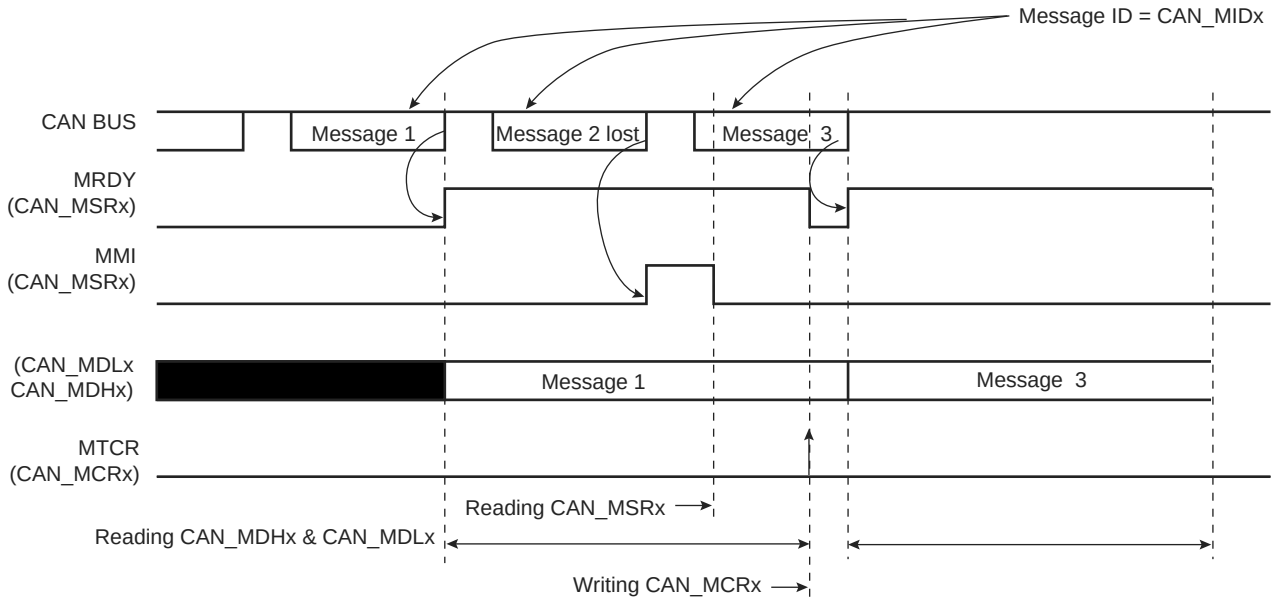
The CAN controller synchronizes itself with the bus activity by checking for eleven consecutive “recessive” bits. Once synchronized, the WAKEUP signal in the CAN_SR is set.

Depending on the corresponding mask in the CAN_IMR, an interrupt is generated while WAKEUP is set. The SLEEP signal in the CAN_SR is automatically cleared once WAKEUP is set. WAKEUP signal is automatically cleared once SLEEP is set.

If no message is being sent on the bus, then the CAN controller is able to send a message eleven bit times after disabling Low-power mode.

If there is bus activity when Low-power mode is disabled, the CAN controller is synchronized with the bus activity in the next interframe. The previous message is lost (see Figure 31-9).

Figure 31-11. Receive Mailbox



Note: In the case of ARM architecture, CAN_MSRx, CAN_MDLx, CAN_MDHx can be read using an optimized ldm assembler instruction.

Receive with Overwrite Mailbox

A mailbox is in Receive with Overwrite Mode once the MOT field in the CAN_MMRx has been configured. Message ID and Message Acceptance masks must be set before Receive Mode is enabled.

After Receive Mode is enabled, the MRDY flag in the CAN_MSR is automatically cleared until the first message is received. When the first message has been accepted by the mailbox, the MRDY flag is set. An interrupt is pending for the mailbox while the MRDY flag is set. This interrupt is masked depending on the mailbox flag in the CAN_IMR global register.

If a new message is received while the MRDY flag is set, this new message is stored in the mailbox data register, overwriting the previous message. The MMI flag in the CAN_MSRx notifies the software that a message has been dropped by the mailbox. This flag is cleared when reading the CAN_MSRx.

The CAN controller may store a new message in the CAN data registers while the application reads them. To check that CAN_MDHx and CAN_MDLx do not belong to different messages, the application must check the MMI bit in the CAN_MSRx before and after reading CAN_MDHx and CAN_MDLx. If the MMI flag is set again after the data registers have been read, the software application has to re-read CAN_MDHx and CAN_MDLx (see Figure 31-12).

33.3 Block Diagram

Figure 33-1. Block Diagram

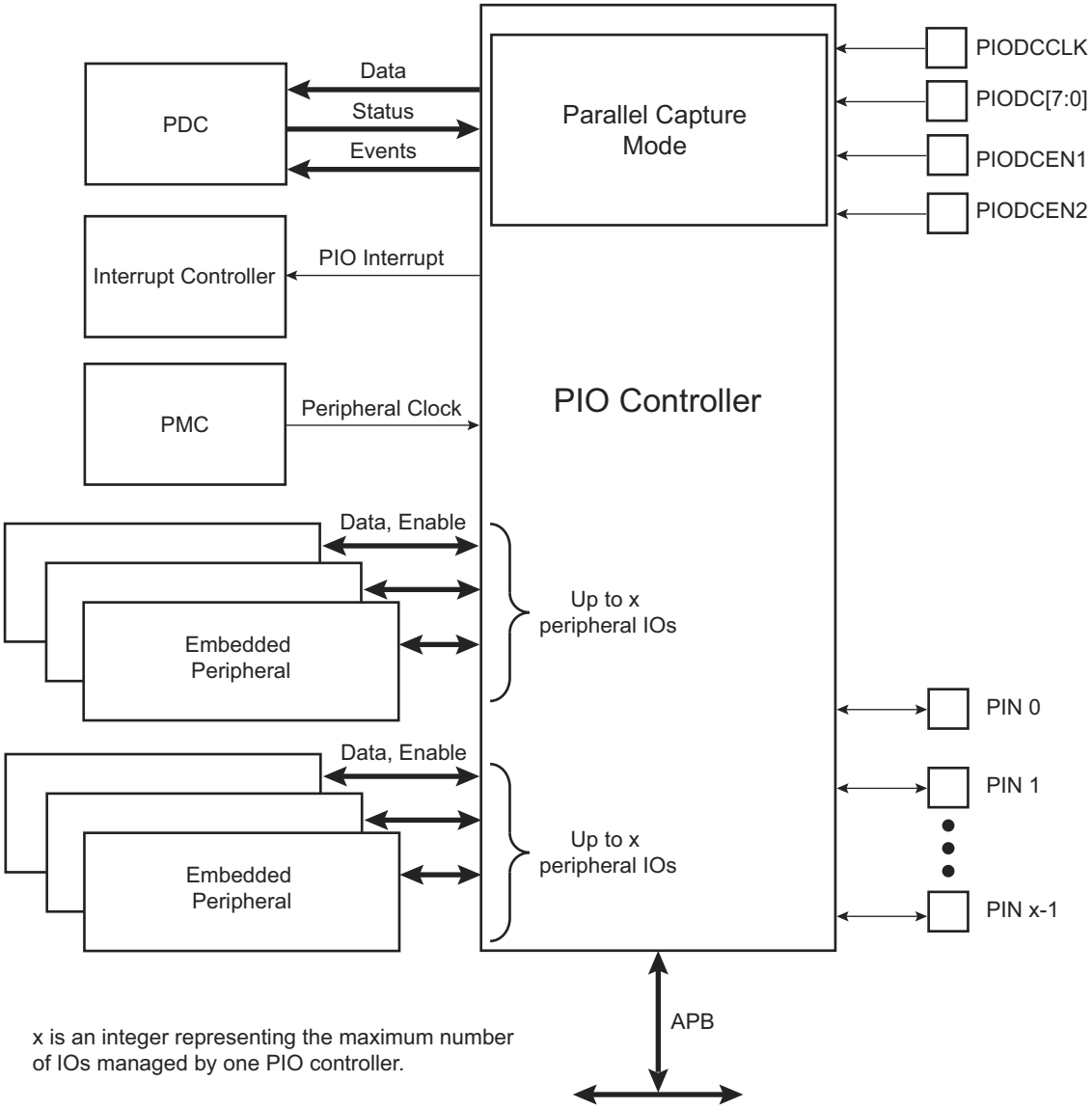


Table 33-1. Signal Description

Signal Name	Signal Description	Signal Type
PIODCCLK	Parallel Capture Mode Clock	Input
PIODC[7:0]	Parallel Capture Mode Data	Input
PIODCEN1	Parallel Capture Mode Data Enable 1	Input
PIODCEN2	Parallel Capture Mode Data Enable 2	Input

33.6.27 PIO Input Filter Slow Clock Enable Register

Name: PIO_IFSCER

Address: 0x400E0E84 (PIOA), 0x400E1084 (PIOB), 0x400E1284 (PIOC), 0x400E1484 (PIOD), 0x400E1684 (PIOE)

Access: Write-only

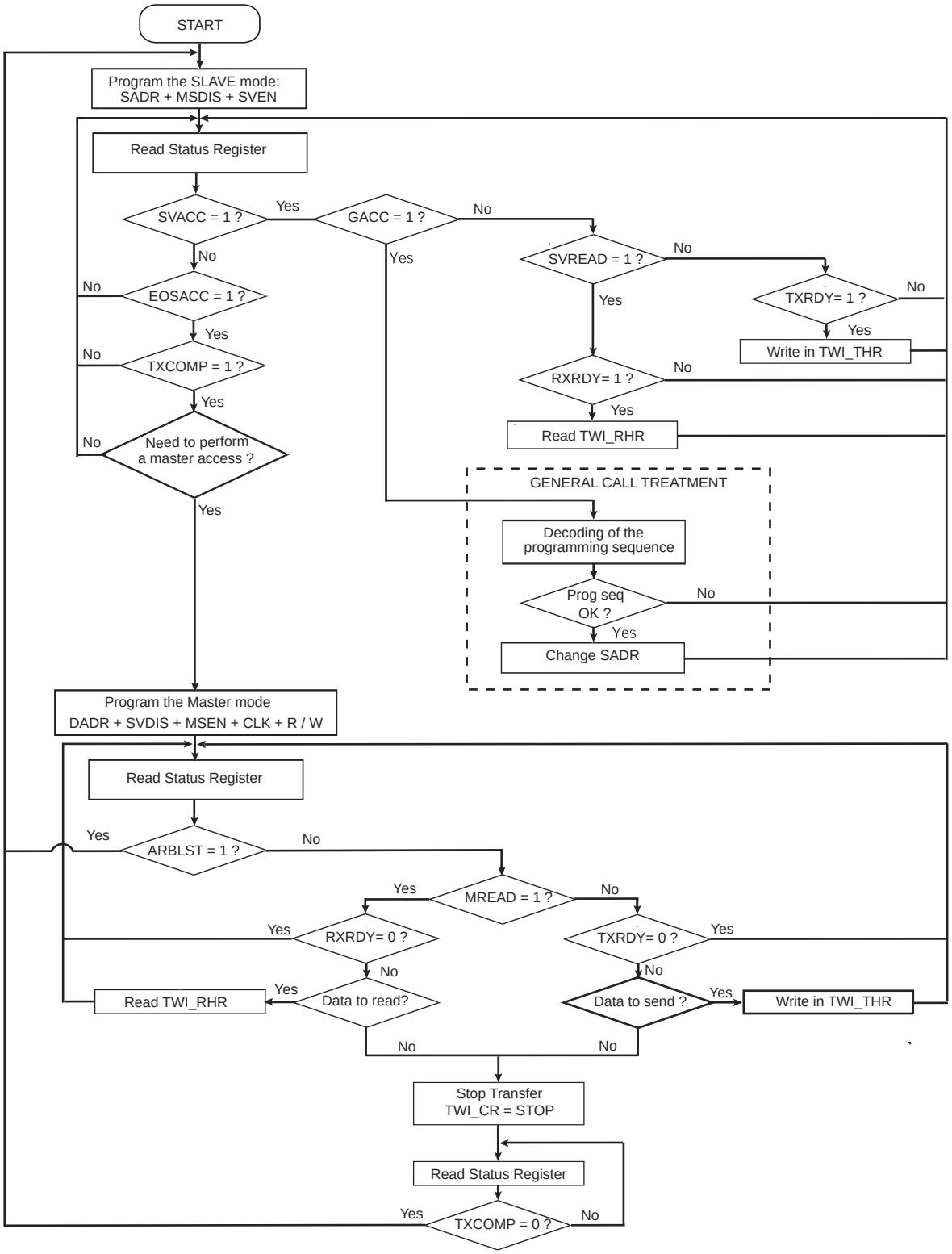
31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

- **P0–P31: Slow Clock Debouncing Filtering Select**

0: No effect.

1: The debouncing filter is able to filter pulses with a duration < $t_{div\_slck}/2$.

Figure 35-22. Multi-master Flowchart



37. Universal Synchronous Asynchronous Receiver Transmitter (USART)

37.1 Description

The Universal Synchronous Asynchronous Receiver Transceiver (USART) provides one full duplex universal synchronous asynchronous serial link. Data frame format is widely programmable (data length, parity, number of stop bits) to support a maximum of standards. The receiver implements parity error, framing error and overrun error detection. The receiver time-out enables handling variable-length frames and the transmitter timeguard facilitates communications with slow remote devices. Multidrop communications are also supported through address bit handling in reception and transmission.

The USART features three test modes: Remote Loopback, Local Loopback and Automatic Echo.

The USART supports specific operating modes providing interfaces on RS485, and SPI buses, with ISO7816 T = 0 or T = 1 smart card slots, infrared transceivers and connection to modem ports. The hardware handshaking feature enables an out-of-band flow control by automatic management of the pins RTS and CTS.

The USART supports the connection to the DMA Controller and the Peripheral DMA Controller, which enables data transfers to the transmitter and from the receiver. The PDC and DMAC provide chained buffer management without any intervention of the processor.

37.2 Embedded Characteristics

- Programmable Baud Rate Generator
- 5- to 9-bit Full-duplex Synchronous or Asynchronous Serial Communications
  - 1, 1.5 or 2 Stop Bits in Asynchronous Mode or 1 or 2 Stop Bits in Synchronous Mode
  - Parity Generation and Error Detection
  - Framing Error Detection, Overrun Error Detection
  - Digital Filter on Receive Line
  - MSB- or LSB-first
  - Optional Break Generation and Detection
  - By 8 or by 16 Oversampling Receiver Frequency
  - Optional Hardware Handshaking RTS-CTS
  - Optional Modem Signal Management DTR-DSR-DCD-RI
  - Receiver Time-out and Transmitter Timeguard
  - Optional Multidrop Mode with Address Generation and Detection
- RS485 with Driver Control Signal
- ISO7816, T = 0 or T = 1 Protocols for Interfacing with Smart Cards
  - NACK Handling, Error Counter with Repetition and Iteration Limit
- IrDA Modulation and Demodulation
  - Communication at up to 115.2 kbit/s
- SPI Mode
  - Master or Slave
  - Serial Clock Programmable Phase and Polarity
  - SPI Serial Clock (SCK) Frequency up to $f_{peripheral\ clock}/6$
- Test Modes
  - Remote Loopback, Local Loopback, Automatic Echo
- Supports Connection of:
  - Two DMA Controller Channels (DMAC) and Two Peripheral DMA Controller Channels (PDC)

When the start frame delimiter is a sync pattern (ONEBIT field to 0), both command and data delimiter are supported. If a valid sync is detected, the received character is written as RXCHR field in the US_RHR and the RXSYNH is updated. RXCHR is set to 1 when the received character is a command, and it is set to 0 if the received character is a data. This mechanism alleviates and simplifies the direct memory access as the character contains its own sync field in the same register.

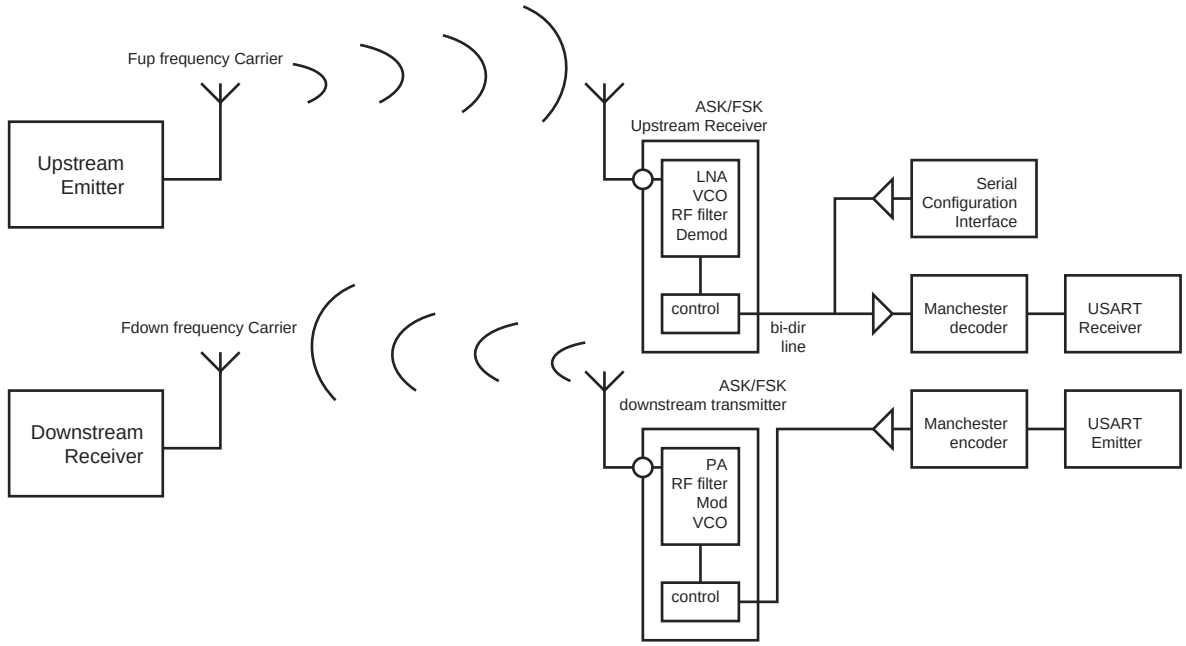
As the decoder is setup to be used in Unipolar mode, the first bit of the frame has to be a zero-to-one transition.

37.6.3.5 Radio Interface: Manchester Encoded USART Application

This section describes low data rate RF transmission systems and their integration with a Manchester encoded USART. These systems are based on transmitter and receiver ICs that support ASK and FSK modulation schemes.

The goal is to perform full duplex radio transmission of characters using two different frequency carriers. See the configuration in Figure 37-16.

Figure 37-16. Manchester Encoded Characters RF Transmission



The USART peripheral is configured as a Manchester encoder/decoder. Looking at the downstream communication channel, Manchester encoded characters are serially sent to the RF emitter. This may also include a user defined preamble and a start frame delimiter. Mostly, preamble is used in the RF receiver to distinguish between a valid data from a transmitter and signals due to noise. The Manchester stream is then modulated. See Figure 37-17 for an example of ASK modulation scheme. When a logic one is sent to the ASK modulator, the power amplifier, referred to as PA, is enabled and transmits an RF signal at downstream frequency. When a logic zero is transmitted, the RF signal is turned off. If the FSK modulator is activated, two different frequencies are used to transmit data. When a logic 1 is sent, the modulator outputs an RF signal at frequency F0 and switches to F1 if the data sent is a 0. See Figure 37-18.

From the receiver side, another carrier frequency is used. The RF receiver performs a bit check operation examining demodulated data stream. If a valid pattern is detected, the receiver switches to Receiving mode. The demodulated stream is sent to the Manchester decoder. Because of bit checking inside RF IC, the data transferred to the microcontroller is reduced by a user-defined number of bits. The Manchester preamble length is to be defined in accordance with the RF IC configuration.

38.6.12.3 WAVSEL = 01

When WAVSEL = 01, the value of TC_CV is incremented from 0 to 2^{32}-1 . Once 2^{32}-1 is reached, the value of TC_CV is decremented to 0, then re-incremented to 2^{32}-1 and so on. See Figure 38-12.

A trigger such as an external event or a software trigger can modify TC_CV at any time. If a trigger occurs while TC_CV is incrementing, TC_CV then decrements. If a trigger is received while TC_CV is decrementing, TC_CV then increments. See Figure 38-13.

RC Compare cannot be programmed to generate a trigger in this configuration.

At the same time, RC Compare can stop the counter clock (CPCSTOP = 1) and/or disable the counter clock (CPCDIS = 1).

Figure 38-12. WAVSEL = 01 without Trigger

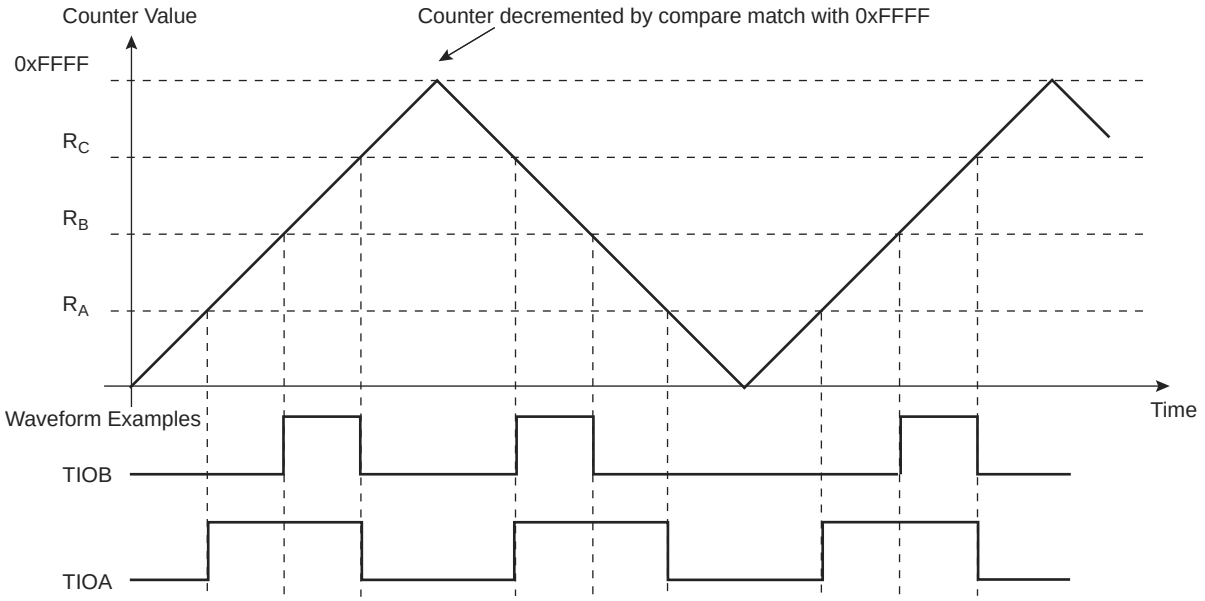
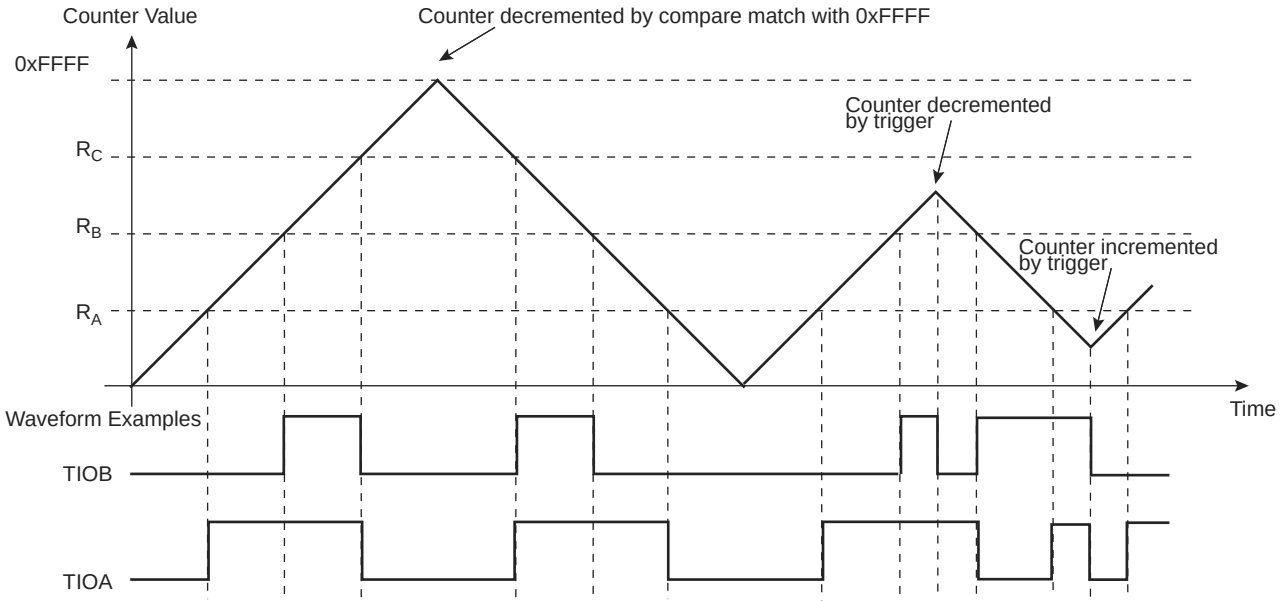


Figure 38-13. WAVSEL = 01 with Trigger



- **SFR: PTP Sync Frame Received**
- **DRQFT: PTP Delay Request Frame Transmitted**
- **SFT: PTP Sync Frame Transmitted**
- **PDRQFR: PDelay Request Frame Received**
- **PDRSFR: PDelay Response Frame Received**
- **PDRQFT: PDelay Request Frame Transmitted**
- **PDRSFT: PDelay Response Frame Transmitted**
- **SRI: TSU Seconds Register Increment**
- **WOL: Wake On LAN**

42.8.39 GMAC 1588 Timer Increment Register

Name: GMAC_TI

Address: 0x400341DC

Access: Read/Write

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

23	22	21	20	19	18	17	16
NIT							

15	14	13	12	11	10	9	8
ACNS							

7	6	5	4	3	2	1	0
CNS							

- **CNS: Count Nanoseconds**

A count of nanoseconds by which the 1588 Timer Nanoseconds Register will be incremented each clock cycle.

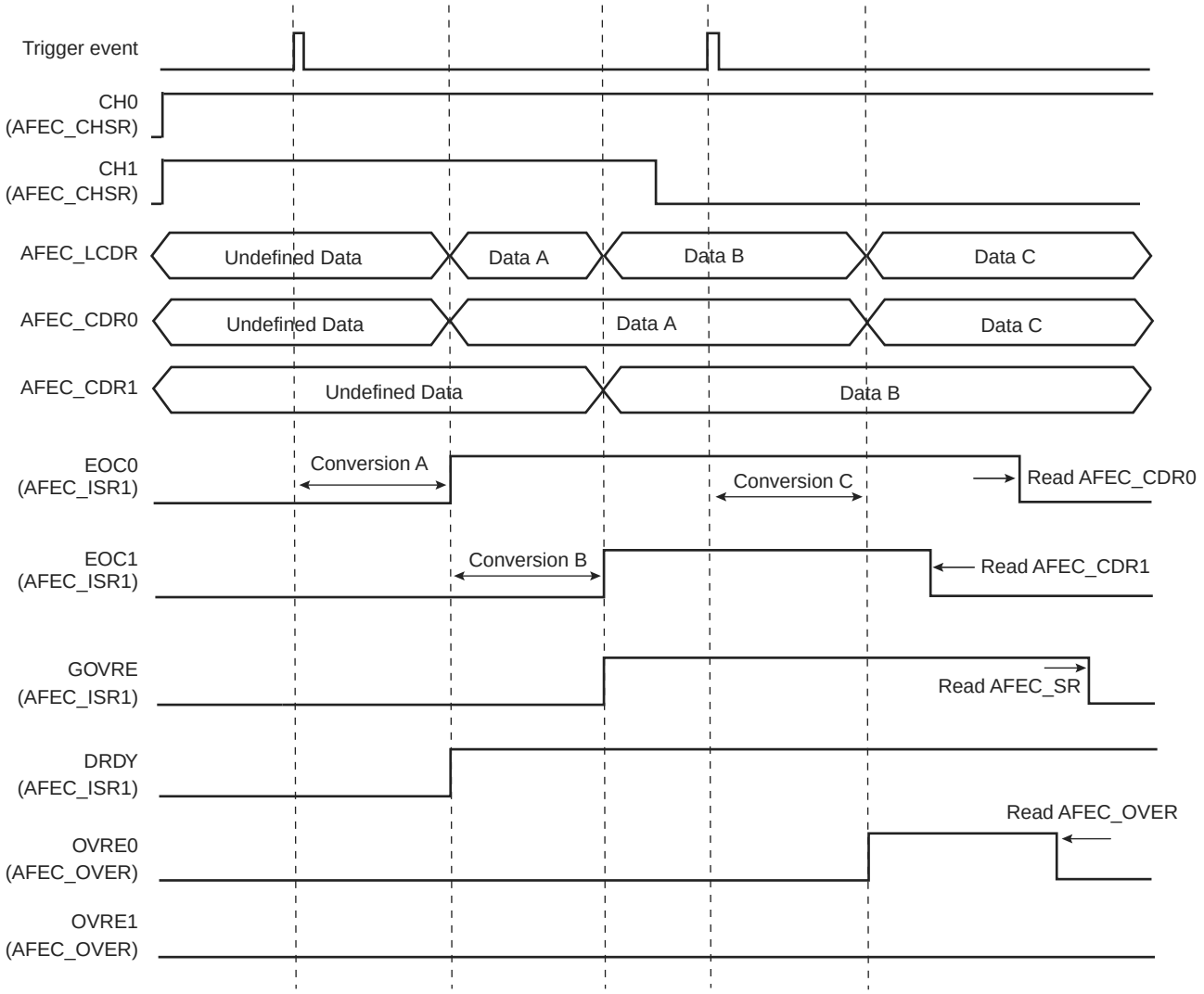
- **ACNS: Alternative Count Nanoseconds**

Alternative count of nanoseconds by which the 1588 Timer Nanoseconds Register will be incremented each clock cycle.

- **NIT: Number of Increments**

The number of increments after which the alternative increment is used.

Figure 43-5. EOCx, GOVRE and OVREx Flag Behavior



Warning: If the corresponding channel is disabled during a conversion, or if it is disabled and then reenabled during a conversion, its associated data and its corresponding EOCx and GOVRE flags in AFEC_ISR and OVREx flags in AFEC_OVER are unpredictable.

43.7.12 AFEC Interrupt Mask Register

Name: AFEC_IMR

Address: 0x400B002C (0), 0x400B402C (1)

Access: Read-only

31	30	29	28	27	26	25	24
EOCAL	TEMPCHG	–	RXBUFF	ENDRX	COMPE	GOVRE	DRDY
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
EOC15	EOC14	EOC13	EOC12	EOC11	EOC10	EOC9	EOC8
7	6	5	4	3	2	1	0
EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

- **EOCx: End of Conversion Interrupt Mask x**
- **DRDY: Data Ready Interrupt Mask**
- **GOVRE: General Overrun Error Interrupt Mask**
- **COMPE: Comparison Event Interrupt Mask**
- **ENDRX: End of Receive Buffer Interrupt Mask**
- **RXBUFF: Receive Buffer Full Interrupt Mask**
- **TEMPCHG: Temperature Change Interrupt Mask**
- **EOCAL: End of Calibration Sequence Interrupt Mask**

46.11.5 SMC Timings

Timings are given in the following domains:

- 1.8V domain: V_{DDIO} from 1.65V to 1.95V, maximum external capacitor = 30 pF
- 3.3V domain: V_{DDIO} from 2.85V to 3.6V, maximum external capacitor = 50 pF

Timings are given assuming a capacitance load on data, control and address pads.

In the tables that follow, t_{CPMCK} is MCK period.

46.11.5.1 Read Timings

Table 46-58. SMC Read Signals - NRD Controlled (READ_MODE = 1)

	VDDIO Supply	1.8V Domain	3.3V Domain	1.8V Domain	3.3V Domain	
Symbol	**Parameter**	**Min**		**Max**		**Unit**
NO HOLD Settings (NRD_HOLD = 0)						
SMC_1	Data Setup before NRD High	20.6	18.6	—	—	ns
SMC_2	Data Hold after NRD High	0	0	—	—	ns
HOLD Settings (NRD_HOLD ≠ 0)						
SMC_3	Data Setup before NRD High	15.9	14.1	—	—	ns
SMC_4	Data Hold after NRD High	0	0	—	—	ns
HOLD or NO HOLD Settings (NRD_HOLD ≠ 0, NRD_HOLD = 0)						
SMC_5	A0–A22 Valid before NRD High	(NRD_SETUP + NRD_PULSE) × t_{CPMCK} - 6.8	(NRD_SETUP + NRD_PULSE) × t_{CPMCK} - 6.5	—	—	ns
SMC_6	NCS low before NRD High	(NRD_SETUP + NRD_PULSE - NCS_RD_SETUP) × t_{CPMCK} - 4.6	(NRD_SETUP + NRD_PULSE - NCS_RD_SETUP) × t_{CPMCK} - 5.1	—	—	ns
SMC_7	NRD Pulse Width	NRD_PULSE × t_{CPMCK} - 7.5	NRD_PULSE × t_{CPMCK} - 6.6	—	—	ns

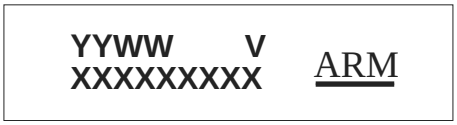
Table 46-59. SMC Read Signals - NCS Controlled (READ_MODE = 0)

	VDDIO Supply	1.8V Domain	3.3V Domain	1.8V Domain	3.3V Domain	
Symbol	**Parameter**	**Min**		**Max**		**Unit**
NO HOLD Settings (NCS_RD_HOLD = 0)						
SMC_8	Data Setup before NCS High	21.8	19.4	—	—	ns
SMC_9	Data Hold after NCS High	0	0	—	—	ns
HOLD Settings (NCS_RD_HOLD ≠ 0)						
SMC_{10}	Data Setup before NCS High	17.1	14.9	—	—	ns
SMC_{11}	Data Hold after NCS High	0	0	—	—	ns
HOLD or NO HOLD Settings (NCS_RD_HOLD ≠ 0, NCS_RD_HOLD = 0)						
SMC_{12}	A0–A22 valid before NCS High	(NCS_RD_SETUP + NCS_RD_PULSE) × t_{CPMCK} - 6.9	(NCS_RD_SETUP + NCS_RD_PULSE) × t_{CPMCK} - 6.6	—	—	ns
SMC_{13}	NRD low before NCS High	(NCS_RD_SETUP + NCS_RD_PULSE - NRD_SETUP) × t_{CPMCK} - 5.8	(NCS_RD_SETUP + NCS_RD_PULSE - NRD_SETUP) × t_{CPMCK} - 5.5	—	—	ns

48. Marking

All devices are marked with the Atmel logo and the ordering code.

Additional marking is as follows:



where

- “YY”: manufactory year
- “WW”: manufactory week
- “V”: revision
- “XXXXXXXXX”: lot number

Table 51-5. SAM4E Datasheet Rev. 11157D 12-Jun-14 Revision history (Continued)

Doc. Date	Changes
12-Jun-2014	Section 1. “Watchdog Timer (WDT)” Figure 1-2, “Watchdog Behavior”, “WDT_CR = WDRSTT” replaced with “WDT_CR.WDRSTT=1”
	Section 17. “Reinforced Safety Watchdog Timer (RSWDT)” General formatting and editorial changes throughout Section 17.2 “Embedded Characteristics”: added bullet “Windowed Watchdog” Figure 17-2 “Watchdog Behavior” replaced “RSWDT_CR = WDRSTT” with “RSWDT_CR.WDRSTT = 1” Added notes in Section 17.5.2 “Reinforced Safety Watchdog Timer Mode Register” and updated Section 17.4 “Functional Description”. KEY is now decribed with a table in Section 17.5.1 “Reinforced Safety Watchdog Timer Control Register”
	Section 18. “Supply Controller (SUPC)” Added Tamper detection and Anti-tampering (Section 18.2 “Embedded Characteristics”, Section 18.4.7.3 “Low-power Tamper Detection and Anti-Tampering”) “Low-power Debouncer Inputs” section restructured: content modified and included in Section 18.4.7.3 “Low-power Tamper Detection and Anti-Tampering” Updated Section 18.3 “Block Diagram” and Figure 18-4 “Wake-up Sources” Updated Section 18.4.2 “Slow Clock Generator”, Section 18.4.4 “Supply Monitor” Updated Section 18.4.4 “Supply Monitor” Section 18.4.6.2 “Brownout Detector Reset” : Reworked 1st paragraph Added Section 18.4.8 “Register Write Protection” and Section 18.4.9 “Register Bits in Backup Domain (VDDIO)” In Section 18.5.9 “System Controller Write Protection Mode Register”: updated register name and bit descriptions. Section 18-5 “Low-power Debouncer (Push-to-Make Switch, Pull-up Resistors)”, Section 18-6 “Low-power Debouncer (Push-to-Break Switch, Pull-down Resistors)” and Section 18-7 “Using WKUP Pins Without RTCOUTx Pins”: Modified pin names. Updated Section 18.5.3 “Supply Controller Control Register”, Section 18.5.4 “Supply Controller Supply Monitor Mode Register”, Section 18.5.6 “Supply Controller Wake-up Mode Register”, Section 18.5.7 “Supply Controller Wake-up Inputs Register”, Section 18.5.8 “Supply Controller Status Register” and Section 18.5.9 “System Controller Write Protection Mode Register” (added information on VDDIO domain and WPEN bit) Section 18.4.7.2 “Wake-up Inputs” corrected WKUPPLx pins to WKUPTx pins. WKUP0, WKUP15 references changed to WKUPx.
	Section 19. “General Purpose Backup Registers (GPBR)” Minor editorial changes Section 19-1 “Register Mapping”: added reset value 0x00000000 for all registers SYS_GPBRx Section 19.3.1 “General Purpose Backup Register x”: inserted sentence “These registers are reset at first power-up and on each loss of VDDBU” below bitmap
	Section 20. “Enhanced Embedded Flash Controller (EEFC)” Reworked section Section 20.4.3.2 “Write Commands” and all sub-sections with figures Figure 20-7 “Full Page Programming” to Figure 20-9 “Programming Bytes in the Flash” Modified Section 20.4.3.3 “Erase Commands” In Section 20.5.2 “EEFC Flash Command Register”, changed the description of FARG field Replaced NVIC by “interrupt controller” everywhere in the document. Revised all figures in the section.
	Section 21. “Fast Flash Programming Interface (FFPI)” Modified Table 21-1 “Signal Description List” (removed references to PGMEN2)

Table 51-5. SAM4E Datasheet Rev. 11157D 12-Jun-14 Revision history (Continued)

Doc. Date	Changes
12-Jun-2014	Section 1. “Timer Counter (TC)” Editorial and formatting changes throughout Master clock” or “MCK” replaced with “peripheral clock”. Removed references to FILTER bit (register bit 19 now reserved in Section 39.7.16 “TC Block Mode Register”) Figure 1-16 “Synchronization with PWM” added value ‘1’ to all multiplexers Figure 1-18 “Input Stage”: replaced “FILTER” with “MAXFILTER > 0” Updated Figure 1-1 “Timer Counter Block Diagram” Updated Figure 1-5 “Example of Transfer with PDC” Erroneous description of TCCLKS table, rows 0 to 4 reworked in Section 1.7.2 “TC Channel Mode Register: Capture Mode” and Section 1.7.3 “TC Channel Mode Register: Waveform Mode”" Updated Section 1.7.16 “TC Block Mode Register” Section 1.6.16.3 “Direction Status and Change Detection”: rewrote sixth paragraph for clarity Section 1.6.16.4 “Position and Rotation Measurement” rewrote first paragraph for clarity Section 1.6.16.3 “Direction Status and Change Detection” replaced sentence “The speed can be read on TC_RA0 register in TC_CMR0” with “The speed can be read on field RA in register TC_RA0” Added Section 1.6.16.6 “Missing Pulse Detection and Auto-correction” Added configuration bit AUTOC in Section 1.7.16 “TC Block Mode Register” Section 1.6.18 “Register Write Protection” changed title (was “Write Protection System”); revised content Section 1.7.22 “TC Write Protection Mode Register”: modified register name (was “TC Write Protect Mode Register”); updated WPEN field description (replaced list of protectable registers with link to Section 1.6.18 “Register Write Protection”) Replaced “0xFFFF” with “2^n-1” (with “n” representing counter size) in Section 1.6.12.1 “WAVSEL = 00”, Section 1.6.12.3 “WAVSEL = 01”, Figure 1-10 “WAVSEL = 10 without Trigger”, Section 1-14 “WAVSEL = 11 without Trigger”, Figure 1-11 “WAVSEL = 10 with Trigger” and Figure 1-15 “WAVSEL = 11 with Trigger”:
	Section 1. “Pulse Width Modulation Controller (PWM)” Editorial and formatting changes throughout. Updated Table 1-4 “Fault Inputs” Modified Section 1.6.2.2 “Comparator” Section 1.6.6 “Register Write Protection”: at end of section, replaced sentence “The WPVS and PWM_WPSR fields are automatically reset after reading the PWM_WPSR register” with “The WPVS and WPVSRC fields are automatically cleared after reading the PWM_WPSR” Section 1.7.9 “PWM Sync Channels Mode Register”: removed table row for value 3 “reserved” in UPDM field description WPKEY/WPCMD are now described with tables in Section 1.7.34 “PWM Write Protection Control Register” Corrected reset value of PWM_FPV2 in Table 1-7 “Register Mapping” (was 0x0000_0000; is 0x003F_003F) Deleted instances of “(fault input bit varies from 0 to Z-1)” from field descriptions in Section 1.7.24 “PWM Fault Mode Register”, Section 1.7.25 “PWM Fault Status Register” on page 1139, Section 1.7.26 “PWM Fault Clear Register” and Section 1.7.28 on page 1142 Updated Section 1.7.34 “PWM Write Protection Control Register” on page 1148