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®-M3
Core Size	32-Bit Single-Core
Speed	96MHz
Connectivity	EBI/EMI, I²C, Memory Card, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	57
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	36K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 4x10b, 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°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/atsam3u2ca-cu

12. ARM Cortex-M3 Processor

12.1 About this section

This section provides the information required for application and system-level software development. It does not provide information on debug components, features, or operation.

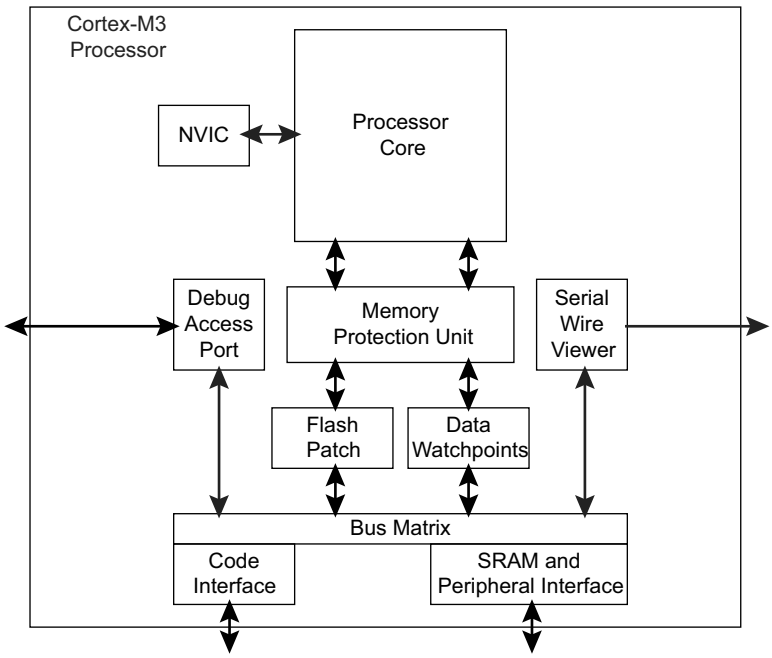
This material is for microcontroller software and hardware engineers, including those who have no experience of ARM products.

Note: The information in this section is reproduced from source material provided to Atmel by ARM Ltd. in terms of Atmel's license for the ARM Cortex-M3 processor core. This information is copyright ARM Ltd., 2008 - 2009.

12.2 About the Cortex-M3 processor and core peripherals

- The Cortex-M3 processor is a high performance 32-bit processor designed for the microcontroller market. It offers significant benefits to developers, including:
- outstanding processing performance combined with fast interrupt handling
- enhanced system debug with extensive breakpoint and trace capabilities
- efficient processor core, system and memories
- ultra-low power consumption with integrated sleep modes
- platform security, with integrated *memory protection unit* (MPU).

Figure 12-1. Typical Cortex-M3 implementation



The Cortex-M3 processor is built on a high-performance processor core, with a 3-stage pipeline Harvard architecture, making it ideal for demanding embedded applications. The processor delivers exceptional power efficiency through an efficient instruction set and extensively optimized design, providing high-end processing hardware including single-cycle 32x32 multiplication and dedicated hardware division.

To facilitate the design of cost-sensitive devices, the Cortex-M3 processor implements tightly-coupled system components that reduce processor area while significantly improving interrupt handling and system debug

12.13.3 SDIV and UDIV

Signed Divide and Unsigned Divide.

12.13.3.1 Syntax

```
SDIV{cond} {Rd,} Rn, Rm
UDIV{cond} {Rd,} Rn, Rm
```

where:

cond is an optional condition code, see “Conditional execution” on page 87.

Rd is the destination register. If *Rd* is omitted, the destination register is *Rn*.

Rn is the register holding the value to be divided.

Rm is a register holding the divisor.

12.13.3.2 Operation

SDIV performs a signed integer division of the value in *Rn* by the value in *Rm*.

UDIV performs an unsigned integer division of the value in *Rn* by the value in *Rm*.

For both instructions, if the value in *Rn* is not divisible by the value in *Rm*, the result is rounded towards zero.

12.13.3.3 Restrictions

Do not use SP and do not use PC.

12.13.3.4 Condition flags

These instructions do not change the flags.

12.13.3.5 Examples

```
SDIV  R0, R2, R4  ; Signed divide, R0 = R2/R4
UDIV  R8, R8, R1  ; Unsigned divide, R8 = R8/R1
```

12.17.9 SEV

Send Event.

12.17.9.1 Syntax

SEV{*cond*}

where:

cond is an optional condition code, see “Conditional execution” on page 87.

12.17.9.2 Operation

SEV is a hint instruction that causes an event to be signaled to all processors within a multiprocessor system. It also sets the local event register to 1, see “Power management” on page 76.

12.17.9.3 Condition flags

This instruction does not change the flags.

12.17.9.4 Examples

```
SEV ; Send Event
```

12.20.8 Configuration and Control Register

The CCR controls entry to Thread mode and enables:

- the handlers for hard fault and faults escalated by FAULTMASK to ignore bus faults
- trapping of divide by zero and unaligned accesses
- access to the STIR by unprivileged software, see “Software Trigger Interrupt Register” on page 158.

See the register summary in Table 12-30 on page 161 for the CCR attributes.

The bit assignments are:

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						STKALIGN	BFHFNMIGN

7	6	5	4	3	2	1	0
Reserved			DIV_0_TRP	UNALIGN_TRP	Reserved	USERSETMPEND	NONBASETHRDENA

- **STKALIGN**

Indicates stack alignment on exception entry:

0 = 4-byte aligned

1 = 8-byte aligned.

On exception entry, the processor uses bit[9] of the stacked PSR to indicate the stack alignment. On return from the exception it uses this stacked bit to restore the correct stack alignment.

- **BFHFNMIGN**

Enables handlers with priority -1 or -2 to ignore data bus faults caused by load and store instructions. This applies to the hard fault and FAULTMASK escalated handlers:

0 = data bus faults caused by load and store instructions cause a lock-up

1 = handlers running at priority -1 and -2 ignore data bus faults caused by load and store instructions.

Set this bit to 1 only when the handler and its data are in absolutely safe memory. The normal use of this bit is to probe system devices and bridges to detect control path problems and fix them.

- **DIV_0_TRP**

Enables faulting or halting when the processor executes an SDIV or UDIV instruction with a divisor of 0:

0 = do not trap divide by 0

1 = trap divide by 0.

When this bit is set to 0,a divide by zero returns a quotient of 0.

- **UNALIGN_TRP**

Enables unaligned access traps:

0 = do not trap unaligned halfword and word accesses

1 = trap unaligned halfword and word accesses.

If this bit is set to 1, an unaligned access generates a usage fault.

Unaligned LDM, STM, LDRD, and STRD instructions always fault irrespective of whether UNALIGN_TRP is set to 1.

17.5.1 RTC Control Register

Name: RTC_CR

Address: 0x400E1260

Access: Read-write

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

This register can only be written if the WPEN bit is cleared in “RTC Write Protect Mode Register” on page 263.

- **UPDTIM: Update Request Time Register**

0 = No effect.

1 = Stops the RTC time counting.

Time counting consists of second, minute and hour counters. Time counters can be programmed once this bit is set and acknowledged by the bit ACKUPD of the Status Register.

- **UPDCAL: Update Request Calendar Register**

0 = No effect.

1 = Stops the RTC calendar counting.

Calendar counting consists of day, date, month, year and century counters. Calendar counters can be programmed once this bit is set.

- **TIMEVSEL: Time Event Selection**

The event that generates the flag TIMEV in RTC_SR (Status Register) depends on the value of TIMEVSEL.

Value	Name	Description
0	MINUTE	Minute change
1	HOUR	Hour change
2	MIDNIGHT	Every day at midnight
3	NOON	Every day at noon

- **CALEVSEL: Calendar Event Selection**

The event that generates the flag CALEV in RTC_SR depends on the value of CALEVSEL

Value	Name	Description
0	WEEK	Week change (every Monday at time 00:00:00)
1	MONTH	Month change (every 01 of each month at time 00:00:00)
2	YEAR	Year change (every January 1 at time 00:00:00)
3	–	

20.4 Functional Description

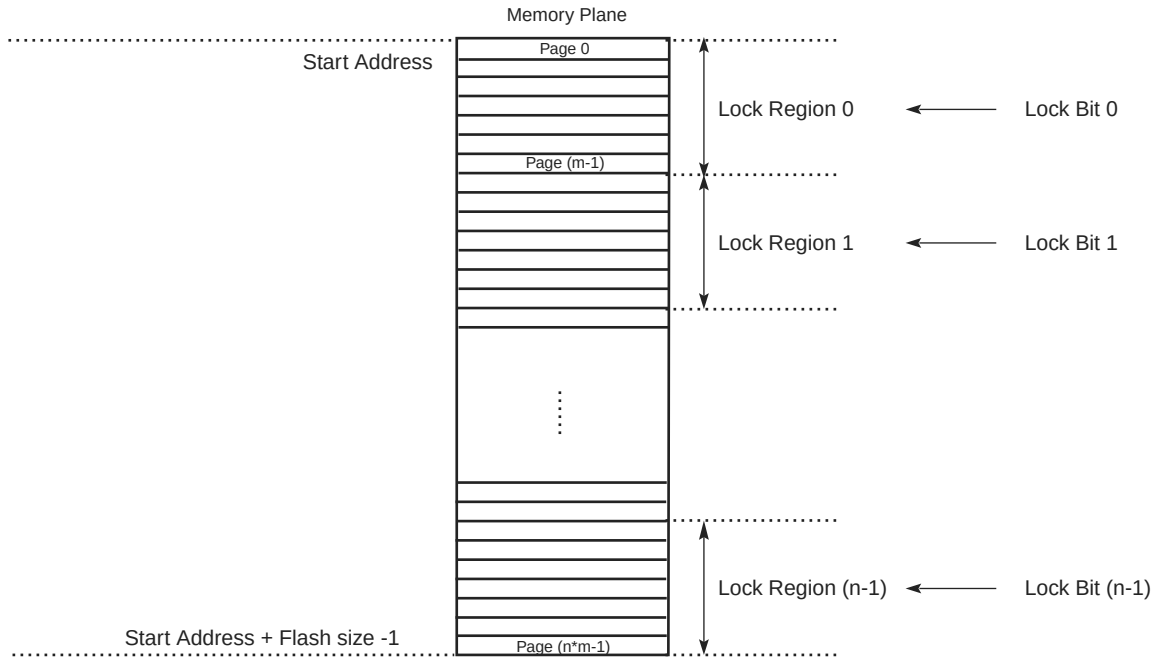
20.4.1 Embedded Flash Organization

The embedded Flash interfaces directly with the 32-bit internal bus. The embedded Flash is composed of:

- One memory plane organized in several pages of the same size.
- Two 128-bit or 64-bit read buffers used for code read optimization.
- One 128-bit or 64-bit read buffer used for data read optimization.
- One write buffer that manages page programming. The write buffer size is equal to the page size. This buffer is write-only and accessible all along the 1 MByte address space, so that each word can be written to its final address.
- Several lock bits used to protect write/erase operation on several pages (lock region). A lock bit is associated with a lock region composed of several pages in the memory plane.
- Several bits that may be set and cleared through the Enhanced Embedded Flash Controller (EEFC) interface, called General Purpose Non Volatile Memory bits (GPNVM bits).

The embedded Flash size, the page size, the lock regions organization and GPNVM bits definition are described in the product definition section. The Enhanced Embedded Flash Controller (EEFC) returns a descriptor of the Flash controlled after a get descriptor command issued by the application (see “Getting Embedded Flash Descriptor” on page 292).

Figure 20-1. Embedded Flash Organization



24.15 Slow Clock Mode

The SMC is able to automatically apply a set of “slow clock mode” read/write waveforms when an internal signal driven by the Power Management Controller is asserted because MCK has been turned to a very slow clock rate (typically 32 kHz clock rate). In this mode, the user-programmed waveforms are ignored and the slow clock mode waveforms are applied. This mode is provided so as to avoid reprogramming the User Interface with appropriate waveforms at very slow clock rate. When activated, the slow mode is active on all chip selects.

24.15.1 Slow Clock Mode Waveforms

Figure 24-28 illustrates the read and write operations in slow clock mode. They are valid on all chip selects. Table 24-8 indicates the value of read and write parameters in slow clock mode.

Figure 24-28. Read/Write Cycles in Slow Clock Mode

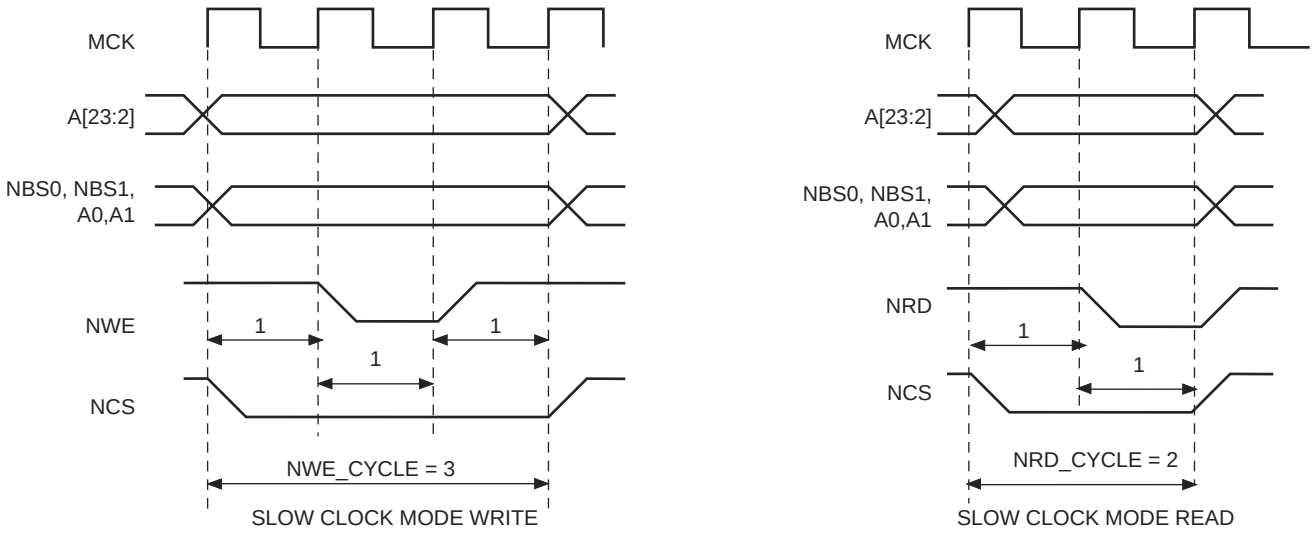


Table 24-8. Read and Write Timing Parameters in Slow Clock Mode

Read Parameters	Duration (cycles)	Write Parameters	Duration (cycles)
NRD_SETUP	1	NWE_SETUP	1
NRD_PULSE	1	NWE_PULSE	1
NCS_RD_SETUP	0	NCS_WR_SETUP	0
NCS_RD_PULSE	2	NCS_WR_PULSE	3
NRD_CYCLE	2	NWE_CYCLE	3

24.15.2 Switching from (to) Slow Clock Mode to (from) Normal Mode

When switching from slow clock mode to normal mode, the current slow clock mode transfer is completed at high clock rate, with the set of slow clock mode parameters. See Figure 24-29. The external device may not be fast enough to support such timings.

Figure 24-30 illustrates the recommended procedure to properly switch from one mode to the other.

24.18.6 SMC NFC Interrupt Mask Register

Name: SMC_IMR

Address: 0x400E0014

Access: Read-only

Reset: 0x00000000

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	RB_EDGE0
23	22	21	20	19	18	17	16
NFCASE	AWB	UNDEF	DTOE	–	–	CMDDONE	XFRDONE
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	RB_FALL	RB_RISE	–	–	–	–

- **RB_RISE: Ready Busy Rising Edge Detection Interrupt Mask**

0: Interrupt source is disabled.

1: Interrupt source is enabled.

- **RB_FALL: Ready Busy Falling Edge Detection Interrupt Mask**

0: Interrupt source is disabled.

1: Interrupt source is enabled.

- **XFRDONE: Transfer Done Interrupt Mask**

0: Interrupt source is disabled.

1: Interrupt source is enabled.

- **CMDDONE: Command Done Interrupt Mask**

0: Interrupt source is disabled.

1: Interrupt source is enabled.

- **DTOE: Data Timeout Error Interrupt Mask**

0: Interrupt source is disabled.

1: Interrupt source is enabled.

- **UNDEF: Undefined Area Access Interrupt Mask5**

0: Interrupt source is disabled.

1: Interrupt source is enabled.

- **AWB: Accessing While Busy Interrupt Mask**

0: Interrupt source is disabled.

1: Interrupt source is enabled.

31.8.4 SPI Transmit Data Register

Name: SPI_TDR

Address: 0x4000800C

Access: Write-only

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

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: Transmit Data**

Data to be transmitted by the SPI Interface is stored in this register. Information to be transmitted must be written to the transmit data register in a right-justified format.

- **PCS: Peripheral Chip Select**

This field is only used if Variable Peripheral Select is active (PS = 1).

If PCSDEC = 0:

PCS = xxx0 NPCS[3:0] = 1110

PCS = xx01 NPCS[3:0] = 1101

PCS = x011 NPCS[3:0] = 1011

PCS = 0111 NPCS[3:0] = 0111

PCS = 1111 forbidden (no peripheral is selected)

(x = don’t care)

If PCSDEC = 1:

NPCS[3:0] output signals = PCS

- **LASTXFER: Last Transfer**

0 = No effect.

1 = The current NPCS will be deasserted after the character written in TD has been transferred. When CSAAT is set, this allows to close the communication with the current serial peripheral by raising the corresponding NPCS line as soon as TD transfer has completed.

This field is only used if Variable Peripheral Select is active (PS = 1).

35.6.11.2 WAVSEL = 10

When WAVSEL = 10, the value of TC_CV is incremented from 0 to the value of RC, then automatically reset on a RC Compare. Once the value of TC_CV has been reset, it is then incremented and so on. See Figure 35-9.

It is important to note that TC_CV can be reset at any time by an external event or a software trigger if both are programmed correctly. See Figure 35-10.

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

Figure 35-9. WAVSEL = 10 without Trigger

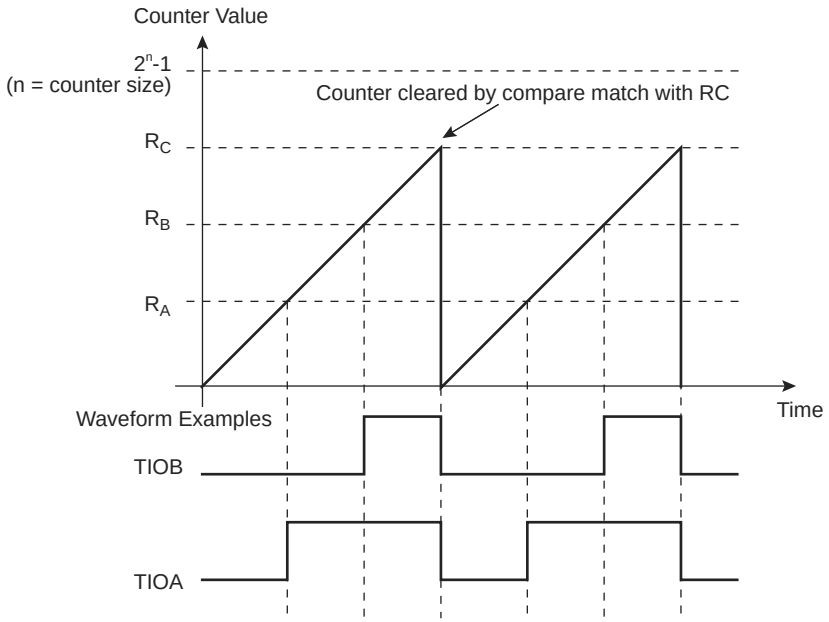
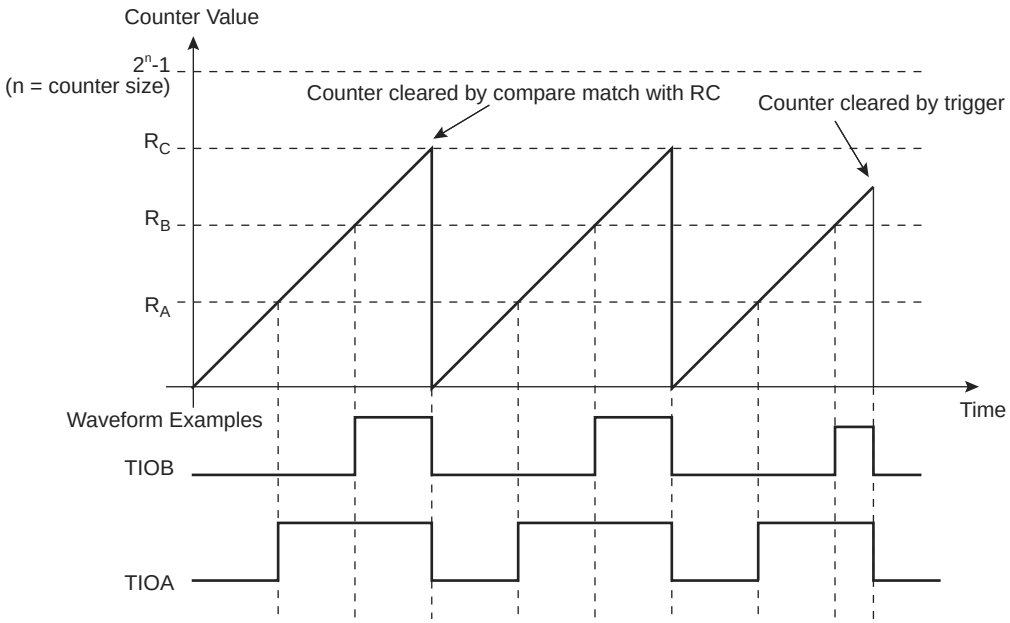


Figure 35-10. WAVSEL = 10 with Trigger



35.6.11.4 WAVSEL = 11

When WAVSEL = 11, the value of TC_CV is incremented from 0 to RC. Once RC is reached, the value of TC_CV is decremented to 0, then re-incremented to RC and so on. See Figure 35-13.

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 35-14.

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

Figure 35-13. WAVSEL = 11 without Trigger

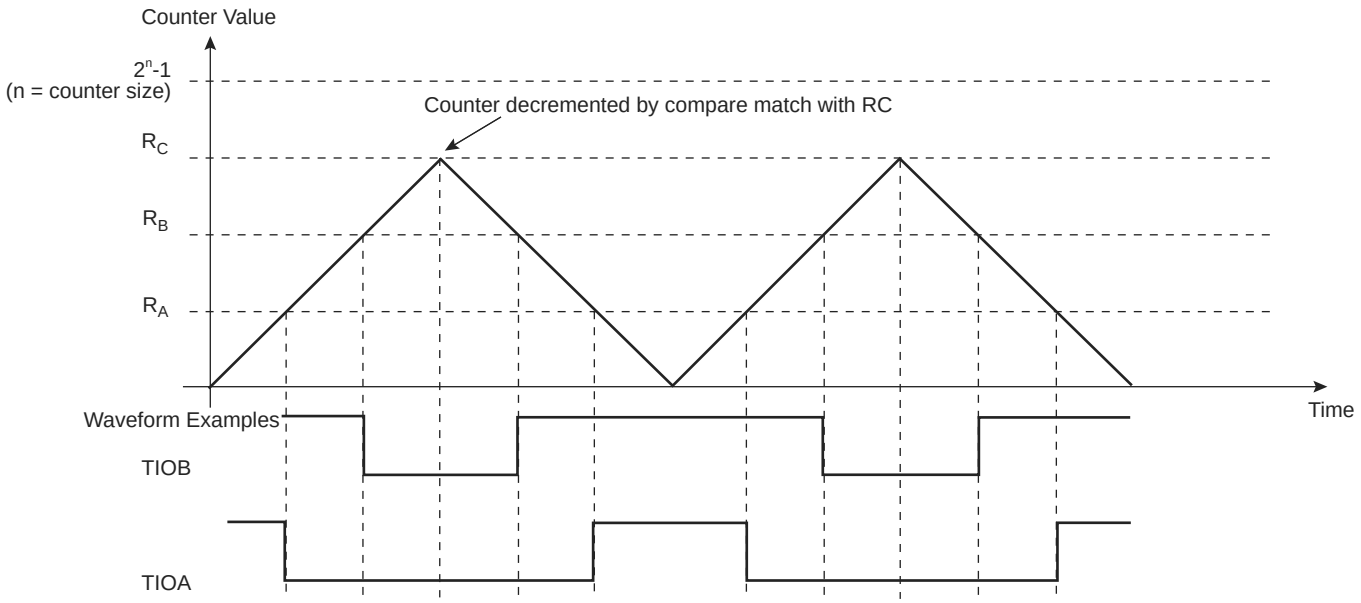
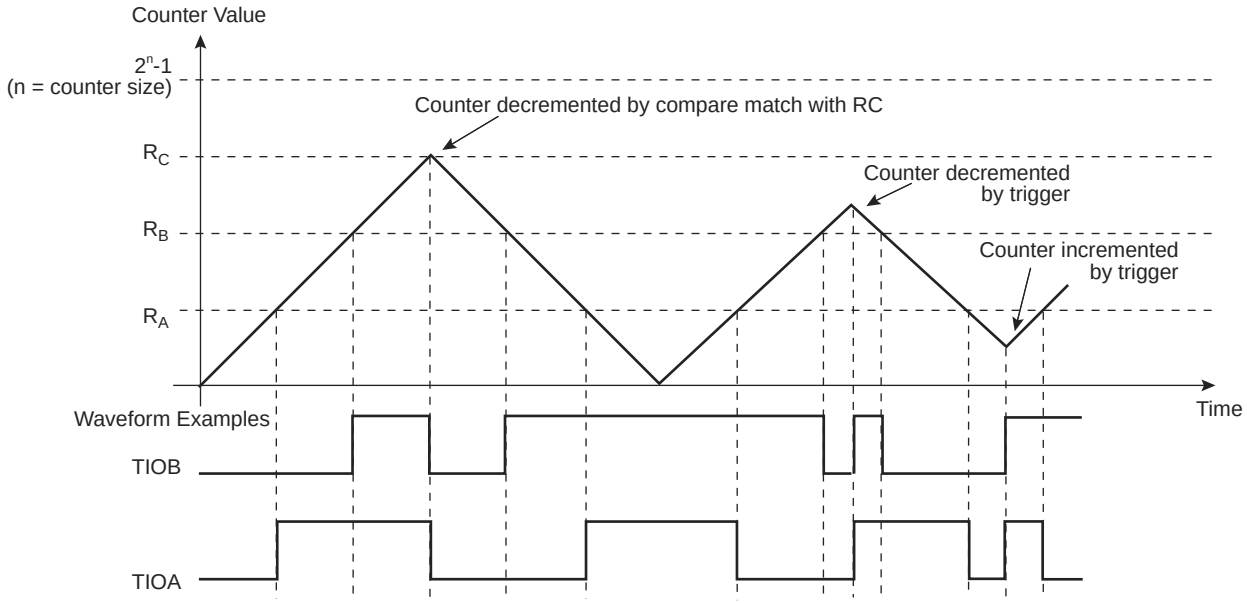


Figure 35-14. WAVSEL = 11 with Trigger



37.7.24 PWM Fault Status Register

Name: PWM_FSR

Address: 0x4008C060

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
FS							

7	6	5	4	3	2	1	0
FIV							

- **FIV: Fault Input Value (fault input bit varies from 0 to 3)**

For each field bit y (fault input number):

0 = The current sampled value of the fault input y is 0 (after filtering if enabled).

1 = The current sampled value of the fault input y is 1 (after filtering if enabled).

- **FS: Fault Status (fault input bit varies from 0 to 3)**

For each field bit y (fault input number):

0 = The fault y is not currently active.

1 = The fault y is currently active.

37.7.30 PWM Write Protect Status Register

Name: PWM_WPSR

Address: 0x4008C0E8

Access: Read-only

31	30	29	28	27	26	25	24
WPVSRC							

23	22	21	20	19	18	17	16
WPVSRC							

15	14	13	12	11	10	9	8
–	–	WPHWS5	WPHWS4	WPHWS3	WPHWS2	WPHWS1	WPHWS0

7	6	5	4	3	2	1	0
WPVS	–	WPSWS5	WPSWS4	WPSWS3	WPSWS2	WPSWS1	WPSWS0

- **WPSWSx: Write Protect SW Status**

0 = The Write Protect SW x of the register group x is disabled.

1 = The Write Protect SW x of the register group x is enabled.

- **WPHWSx: Write Protect HW Status**

0 = The Write Protect HW x of the register group x is disabled.

1 = The Write Protect HW x of the register group x is enabled.

- **WPVS: Write Protect Violation Status**

0 = No Write Protect violation has occurred since the last read of the PWM_WPSR register.

1 = At least one Write Protect violation has occurred since the last read of the PWM_WPSR register. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

- **WPVSRC: Write Protect Violation Source**

When WPVS is active, this field indicates the write-protected register (through address offset) in which a write access has been attempted.

Note: The two LSBs of the address offset of the write-protected register are not reported

Note: Reading PWM_WPSR automatically clears WPVS and WPVSRC fields.

38.6.9.7 Isochronous IN

Isochronous-IN is used to transmit a stream of data whose timing is implied by the delivery rate. Isochronous transfer provides periodic, continuous communication between host and device.

It guarantees bandwidth and low latencies appropriate for telephony, audio, video, etc.

If the endpoint is not available (TX_PK_RDY = 0), then the device does not answer to the host. An ERR_FL_ISO interrupt is generated in the UDPHS_EPTSTAx register and once enabled, then sent to the CPU.

The STALL_SNT command bit is not used for an ISO-IN endpoint.

38.6.9.8 High Bandwidth Isochronous Endpoint Handling: IN Example

For high bandwidth isochronous endpoints, the DMA can be programmed with the number of transactions (BUFF_LENGTH field in UDPHS_DMACONTROLx) and the system should provide the required number of packets per microframe, otherwise, the host will notice a sequencing problem.

A response should be made to the first token IN recognized inside a microframe under the following conditions:

- If at least one bank has been validated, the correct DATAx corresponding to the programmed Number Of Transactions per Microframe (NB_TRANS) should be answered. In case of a subsequent missed or corrupted token IN inside the microframe, the USB 2.0 Core available data bank(s) that should normally have been transmitted during that microframe shall be flushed at its end. If this flush occurs, an error condition is flagged (ERR_FLUSH is set in UDPHS_EPTSTAx).
- If no bank is validated yet, the default DATA0 ZLP is answered and underflow is flagged (ERR_FL_ISO is set in UDPHS_EPTSTAx). Then, no data bank is flushed at microframe end.
- If no data bank has been validated at the time when a response should be made for the second transaction of NB_TRANS = 3 transactions microframe, a DATA1 ZLP is answered and underflow is flagged (ERR_FL_ISO is set in UDPHS_EPTSTAx). If and only if remaining untransmitted banks for that microframe are available at its end, they are flushed and an error condition is flagged (ERR_FLUSH is set in UDPHS_EPTSTAx).
- If no data bank has been validated at the time when a response should be made for the last programmed transaction of a microframe, a DATA0 ZLP is answered and underflow is flagged (ERR_FL_ISO is set in UDPHS_EPTSTAx). If and only if the remaining untransmitted data bank for that microframe is available at its end, it is flushed and an error condition is flagged (ERR_FLUSH is set in UDPHS_EPTSTAx).
- If at the end of a microframe no valid token IN has been recognized, no data bank is flushed and no error condition is reported.

At the end of a microframe in which at least one data bank has been transmitted, if less than NB_TRANS banks have been validated for that microframe, an error condition is flagged (ERR_TRANS is set in UDPHS_EPTSTAx).

Cases of Error (in UDPHS_EPTSTAx)

- ERR_FL_ISO: There was no data to transmit inside a microframe, so a ZLP is answered by default.
- ERR_FLUSH: At least one packet has been sent inside the microframe, but the number of token IN received is lesser than the number of transactions actually validated (TX_BK_RDY) and likewise with the NB_TRANS programmed.
- ERR_TRANS: At least one packet has been sent inside the microframe, but the number of token IN received is lesser than the number of programmed NB_TRANS transactions and the packets not requested were not validated.
- ERR_FL_ISO + ERR_FLUSH: At least one packet has been sent inside the microframe, but the data has not been validated in time to answer one of the following token IN.
- ERR_FL_ISO + ERR_TRANS: At least one packet has been sent inside the microframe, but the data has not been validated in time to answer one of the following token IN and the data can be discarded at the microframe end.

38.7.4 UDPHS Interrupt Status Register

Name: UDPHS_INTSTA

Address: 0x400A4014

Access: Read-only

31	30	29	28	27	26	25	24
–	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
7	6	5	4	3	2	1	0
UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	SPEED

- **SPEED: Speed Status**

0 = reset by hardware when the hardware is in Full Speed mode.

1 = set by hardware when the hardware is in High Speed mode

- **DET_SUSPD: Suspend Interrupt**

0 = cleared by setting the DET_SUSPD bit in UDPHS_CLRINT register

1 = set by hardware when a UDPHS Suspend (Idle bus for three frame periods, a J state for 3 ms) is detected. This triggers a UDPHS interrupt when the DET_SUSPD bit is set in UDPHS_IEN register.

- **MICRO_SOF: Micro Start Of Frame Interrupt**

0 = cleared by setting the MICRO_SOF bit in UDPHS_CLRINT register.

1 = set by hardware when an UDPHS micro start of frame PID (SOF) has been detected (every 125 us) or synthesized by the macro. This triggers a UDPHS interrupt when the MICRO_SOF bit is set in UDPHS_IEN. In case of detected SOF, the MICRO_FRAME_NUM field in UDPHS_FNUM register is incremented and the FRAME_NUMBER field doesn't change.

Note: The Micro Start Of Frame Interrupt (MICRO_SOF), and the Start Of Frame Interrupt (INT_SOF) are not generated at the same time.

- **INT_SOF: Start Of Frame Interrupt**

0 = cleared by setting the INT_SOF bit in UDPHS_CLRINT.

1 = set by hardware when an UDPHS Start Of Frame PID (SOF) has been detected (every 1 ms) or synthesized by the macro. This triggers a UDPHS interrupt when the INT_SOF bit is set in UDPHS_IEN register. In case of detected SOF, in High Speed mode, the MICRO_FRAME_NUMBER field is cleared in UDPHS_FNUM register and the FRAME_NUMBER field is updated.

- **ENDRESET: End Of Reset Interrupt**

0 = cleared by setting the ENDRESET bit in UDPHS_CLRINT.

1 = set by hardware when an End Of Reset has been detected by the UDPHS controller. This triggers a UDPHS interrupt when the ENDRESET bit is set in UDPHS_IEN.

If one of the TIOA outputs is selected, the corresponding Timer Counter channel must be programmed in Waveform Mode.

Only one start command is necessary to initiate a conversion sequence on all the channels. The ADC hardware logic automatically performs the conversions on the active channels, then waits for a new request. The Channel Enable (ADC_CHER) and Channel Disable (ADC_CHDR) Registers enable the analog channels to be enabled or disabled independently.

If the ADC is used with a PDC, only the transfers of converted data from enabled channels are performed and the resulting data buffers should be interpreted accordingly.

Warning: Enabling hardware triggers does not disable the software trigger functionality. Thus, if a hardware trigger is selected, the start of a conversion can be initiated either by the hardware or the software trigger.

41.5.6 Sleep Mode and Conversion Sequencer

The ADC Sleep Mode maximizes power saving by automatically deactivating the ADC when it is not being used for conversions. Sleep Mode is selected by setting the bit SLEEP in the Mode Register ADC_MR.

The SLEEP mode is automatically managed by a conversion sequencer, which can automatically process the conversions of all channels at lowest power consumption.

When a start conversion request occurs, the ADC is automatically activated. As the analog cell requires a start-up time, the logic waits during this time and starts the conversion on the enabled channels. When all conversions are complete, the ADC is deactivated until the next trigger. Triggers occurring during the sequence are not taken into account.

The conversion sequencer allows automatic processing with minimum processor intervention and optimized power consumption. Conversion sequences can be performed periodically using a Timer/Counter output or a PWM Event line. The periodic acquisition of several samples can be processed automatically without any intervention of the processor thanks to the PDC.

Note: The reference voltage pins always remain connected in normal mode as in sleep mode.

41.5.7 ADC Timings

Each ADC has its own minimal Startup Time that is programmed through the field STARTUP in the Mode Register ADC_MR.

In the same way, a minimal Sample and Hold Time is necessary for the ADC to guarantee the best converted final value between two channels selection. This time has to be programmed through the bitfield SHTIM in the Mode Register ADC_MR.

Warning: No input buffer amplifier to isolate the source is included in the ADC. This must be taken into consideration to program a precise value in the SHTIM field. See the section, ADC Characteristics in the product datasheet.

42.9 AC Characteristics

42.9.1 Master Clock Characteristics

Table 42-45. Master Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Unit
$1/(t_{CPMCK})$	Master Clock Frequency	VDDCORE @ 1.62V VDDCORE @ 1.8V		84 96	MHz

42.9.2 I/O Characteristics

Criteria used to define the maximum frequency of the I/Os:

- Output duty cycle (40%–60%)
- Minimum output swing: 100 mV to VDDIO - 100 mV
- Minimum output swing: 100 mV to VDDIO - 100 mV
- Addition of rising and falling time inferior to 75% of the period

Table 42-46. I/O Characteristics

Symbol	Parameter	Conditions		Min	Max	Unit
FreqMax1	Pin Group 1 (1) Maximum output frequency	30 pF	V_{DDIO} = 1.62V V_{DDIO} = 3.0V		45 65	MHz
		45 pF	V_{DDIO} = 1.62V V_{DDIO} = 3.0V		34 45	
$PulseminH_1$	Pin Group 1 (1) High Level Pulse Width	30 pF	V_{DDIO} = 1.62V V_{DDIO} = 3.0V	11 7.7		ns
		45 pF	V_{DDIO} = 1.62V V_{DDIO} = 3.0V	14.7 11		
$PulseminL_1$	Pin Group 1 (1) Low Level Pulse Width	30 pF	V_{DDIO} = 1.62V V_{DDIO} = 3.0V	11 7.7		ns
		45 pF	V_{DDIO} = 1.62V V_{DDIO} = 3.0V	14.7 11		
FreqMax2	Pin Group 2 (2) Maximum output frequency	25 pF	1.62V < V_{DDIO} < 3.6V		35	MHz
$PulseminH_2$	Pin Group 2 (2) High Level Pulse Width	25 pF	1.62V < V_{DDIO} < 3.6V	14.5		ns
$PulseminL_2$	Pin Group 2 (2) Low Level Pulse Width	25 pF	1.62V < V_{DDIO} < 3.6V	14.5		ns

Notes: 1. Pin Group 1 = PA3, PA15

2. Pin Group 2 = PA[0–2], PA[4–14], PA[16–31], PB[0–31], PC[0–31]

Table 42-47. NRSTB Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{NRSTB(filtered)}$	Filtered Pulse Width				1	µs
$t_{NRSTB(unfiltered)}$	Unfiltered Pulse Width		100			µs

Note that in SPI master mode the SAM3U does not sample the data (MISO) on the opposite edge where data clocks out (MOSI) but the same edge is used as shown in Figure 42-17 and Figure 42-18.

42.9.4 MCI Timings

The High Speed MultiMedia Card Interface (HSMCI) supports the MultiMedia Card (MMC) Specification V4.3, the SD Memory Card Specification V2.0, the SDIO V2.0 specification and CE-ATA V1.1.

Figure 42-24. SSC Transmitter, TK and TF as Input

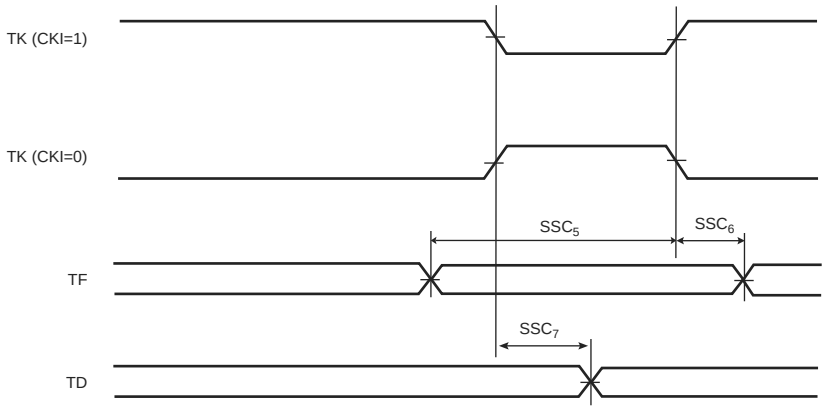


Figure 42-25. SSC Receiver RK and RF as Input

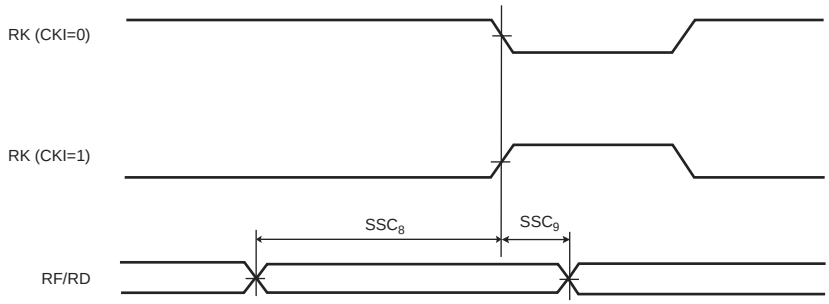
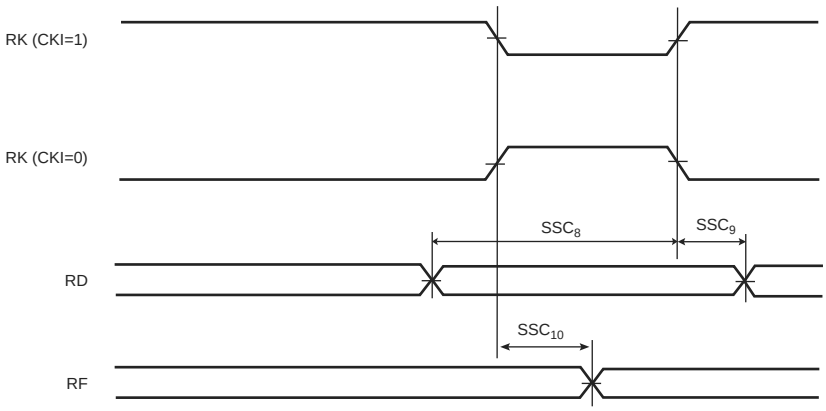


Figure 42-26. SSC Receiver, RK as Input and RF as Output



Doc. Rev. 6430C	Comments (Continued)	Change Request Ref.(1)
	PWM:	
	Section 38.6.23 ”PWM Fault Mode Register”, Section 38.6.24 ”PWM Fault Status Register”, Section 38.6.25 ”PWM Fault Clear Register”, Section 38.6.27 ”PWM Fault Protection Enable Register”, bitfield descriptions updated.	6824
	Table 38-2, “I/O Lines”, new to datasheet.	rfo
	RTC:	
	Section 17.3.2 ”Interrupt”, updated.	7071
	Section 17.5 ”Real Time Clock (RTC) User Interface”, the reset for RTC_CALR is 0x01210720.	7046/7087
	TIMEVSEL, CALEVSEL bitfield descriptions reorganized.	6796
	SSC:	
	Redundant letter C removed from title.	6949
	SUPC:	
	Section 19.4 ”Supply Controller (SUPC) User Interface”, offset updated for GPBR: 0x90-0xDC.	6950
	FWUPDBC, WUPDBC bitfield descriptions reorganized.	6796
	Backup supply is VDDBU	6714
	TC:	
	Figure 36-2 ”Clock Chaining Selection”, channel 1 updated.	6687
	updated bitfields: TC0XC0S, TC1XC1S, TC2XC2S, TCCLKS, BURST, ETRGEDG, LDRA, LDRB, TCCLKS, BURST, EEVTEDG, EEVT, WAVSEL, ACPA, ACPC, AEEVT, ASWTRG, BCPB, BCPC, BEEVT, BSWTRG	6796
	UDHP:	
	Figure 39-4 ”Logical Address Space for DPR Access”, EP0 has but 1 bank	6750
	Figure 39-1 ”Block Diagram”, 1 PMC to UTMI signal line. Notes removed.	6792
	Figure 39.4 ”Product Dependencies”, added to datasheet.	rfo
	Figure 39-6 ”Register Mapping”, DMA offset updated to 0x300 + channel *...	6822
	USART:	
	Section 35.6.7 ”Modem Mode”, is available.	6791
	Section 35.6 ”Functional Description”, ...SCK up to MCK/6	rfo→7097
	Section 35.6.8.2 ”Baud Rate”	
	SPI Master Mode: ...”the value programmed in CD must be superior or equal to 6.”	
	SPI Slave Mode:...”the external clock (SCK) frequency must be at least 6 times lower than the system clock.”	
	Section 35.6.1 ”Baud Rate Generator”,”...signal provided on SCK must be at least 3 times lower than MCK in USART mode, or 6 in SPI mode.”	
	Section 35.6.1.3 ”Baud Rate in Synchronous Mode or SPI Mode”, ...”limits the SCK maximum frequency to MCK/3 in USART mode, or MCK/6 in SPI mode.”	