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	I²C, SPI, UART/USART, USB
Peripherals	DMA, I²S, POR, PWM, WDT
Number of I/O	48
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	176K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamg55j19a-au

13.4.1.12 Exception Mask Registers

The exception mask registers disable the handling of exceptions by the processor. Disable exceptions where they might impact on timing critical tasks.

To access the exception mask registers use the MSR and MRS instructions, or the CPS instruction to change the value of PRIMASK or FAULTMASK. See “MRS”, “MSR”, and “CPS” for more information.

13.6.4.1 ADR

Load PC-relative address.

Syntax

```
ADR{cond} Rd, label
```

where:

cond is an optional condition code, see “Conditional Execution”.

Rd is the destination register.

label is a PC-relative expression. See “PC-relative Expressions”.

Operation

ADR determines the address by adding an immediate value to the PC, and writes the result to the destination register.

ADR produces position-independent code, because the address is PC-relative.

If ADR is used to generate a target address for a BX or BLX instruction, ensure that bit[0] of the address generated is set to 1 for correct execution.

Values of *label* must be within the range of –4095 to +4095 from the address in the PC.

Note: The user might have to use the .W suffix to get the maximum offset range or to generate addresses that are not word-aligned. See “Instruction Width Selection”.

Restrictions

Rd must not be SP and must not be PC.

Condition Flags

This instruction does not change the flags.

Examples

```
ADR     R1, TextMessage   ; Write address value of a location labelled as
                          ; TextMessage to R1
```

13.6.11.7 VDIV

Divides floating-point values.

Syntax

```
VDIV{cond}.F32 {Sd,} Sn, Sm
```

where:

cond is an optional condition code, see “Conditional Execution”.

Sd is the destination register.

Sn, Sm are the operand registers.

Operation

This instruction:

1. Divides one floating-point value by another floating-point value.
2. Writes the result to the floating-point destination register.

Restrictions

There are no restrictions.

Condition Flags

These instructions do not change the flags.

13.12.2.3 Floating-point Context Address Register

Name: FPCAR

Access: Read/Write

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

The FPCAR holds the location of the unpopulated floating-point register space allocated on an exception stack frame.

- **ADDRESS: Location of Unpopulated Floating-point Register Space Allocated on an Exception Stack Frame**

The location of the unpopulated floating-point register space allocated on an exception stack frame.

- **LCKDOWN: Lockdown Supported**

0: Lockdown is not supported.

1: Lockdown is supported.

- **CSIZE: Data Cache Size**

Value	Name	Description
0	CSIZE_1KB	Data cache size is 1 Kbyte
1	CSIZE_2KB	Data cache size is 2 Kbytes
2	CSIZE_4KB	Data cache size is 4 Kbytes
3	CSIZE_8KB	Data cache size is 8 Kbytes

- **CLSIZE: Cache LIne Size**

Value	Name	Description
0	CLSIZE_1KB	Cache line size is 4 bytes
1	CLSIZE_2KB	Cache line size is 8 bytes
2	CLSIZE_4KB	Cache line size is 16 bytes
3	CLSIZE_8KB	Cache line size is 32 bytes

18.20.10 PMC Clock Generator PLLB Register

Name: CKGR_PLLBR

Address: 0x400E042C

Access: Read/Write

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

23	22	21	20	19	18	17	16
MULB							

15	14	13	12	11	10	9	8
–	–	PLLBCOUNT					

7	6	5	4	3	2	1	0
PLLBEN							

Possible limitations on PLLB input frequencies and multiplier factors should be checked before using the PMC.

This register can only be written if the WPEN bit is cleared in the “PMC Write Protection Mode Register” .

- **PLLBEN: PLLB Control**

0: PLLB is disabled

1: PLLB is enabled

2–255: Forbidden

- **PLLBCOUNT: PLLB Counter**

Specifies the number of Slow Clock cycles before the LOCKB bit is set in PMC_SR after CKGR_PLLBR is written.

- **MULB: PLLB Multiplier**

0: The PLLB is deactivated (PLLB also disabled if DIVB = 0).

8 up to 2400: The PLLB Clock frequency is the PLLB input frequency multiplied by MULB + 1.

Unlisted values are forbidden.

- **ZERO: Must Be Written to 0**

19.4.4 Reset State Priorities

The reset state manager manages the priorities among the different reset sources. The resets are listed in order of priority as follows:

1. General reset
2. Backup reset
3. 32.768 kHz Crystal Failure Detection reset
4. Watchdog reset
5. Software reset
6. User reset

Specific cases are listed below:

- When in user reset:
  - A watchdog event is impossible because the Watchdog Timer is being reset by the proc_nreset signal.
  - A software reset is impossible, since the processor reset is being activated.
- When in software reset:
  - A watchdog event has priority over the current state.
  - The NRST has no effect.
- When in watchdog reset:
  - The processor reset is active and so a software reset cannot be programmed.
  - A user reset cannot be entered.

21.5.5.4 Transmit Buffer Empty

The transmit buffer empty flag is set when PERIPH_TCR reaches zero, with PERIPH_TNCR also set to zero and the last data written to peripheral THR.

This flag is reset by writing a non-zero value to PERIPH_TCR or PERIPH_TNCR.

24.6 Functional Description

24.6.1 Description

All channels of the Timer Counter are independent and identical in operation. The registers for channel programming are listed in Table 24-6 “Register Mapping”.

24.6.2 16-bit Counter

Each 16-bit channel is organized around a 16-bit counter. The value of the counter is incremented at each positive edge of the selected clock. When the counter has reached the value 2^{16}-1 and passes to zero, an overflow occurs and the COVFS bit in the TC Status Register (TC_SR) is set.

The current value of the counter is accessible in real time by reading the TC Counter Value Register (TC_CV). The counter can be reset by a trigger. In this case, the counter value passes to zero on the next valid edge of the selected clock.

24.6.3 Clock Selection

At block level, input clock signals of each channel can be connected either to the external inputs TCLKx, or to the internal I/O signals TIOAx for chaining(1) by programming the TC Block Mode Register (TC_BMR). See Figure 24-2.

Each channel can independently select an internal or external clock source for its counter(2):

- External clock signals: XC0, XC1 or XC2
- Internal clock signals: MCK/2, MCK/8, MCK/32, MCK/128, PCK3

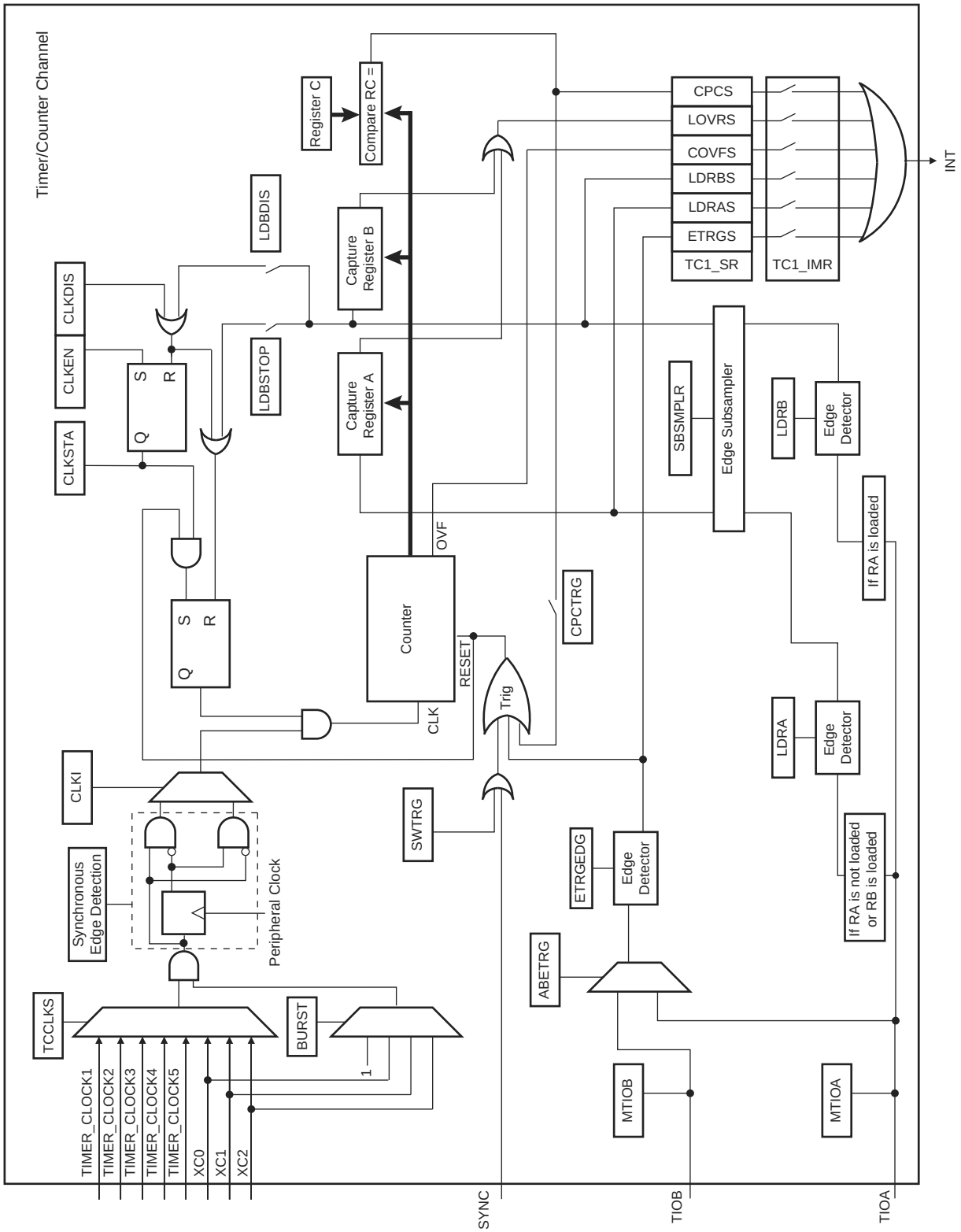
This selection is made by the TCCLKS bits in the TC Channel Mode Register (TC_CMR).

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

The burst function allows the clock to be validated when an external signal is high. The BURST parameter in the TC_CMR defines this signal (none, XC0, XC1, XC2). See Figure 24-3.

Notes:
1. In Waveform mode, to chain two timers, it is mandatory to initialize some parameters:
   - Configure TIOx outputs to 1 or 0 by writing the required value to TC_CMR.ASWTRG.
   - Bit TC_BCR.SYNC must be written to 1 to start the channels at the same time.
2. In all cases, if an external clock or asynchronous internal clock PCK3 is used, the duration of each of its levels must be longer than the peripheral clock period, so the clock frequency will be at least 2.5 times lower than the peripheral clock.

Figure 24-6. Capture Mode



24.7.1 TC Channel Control Register

Name: TC_CCRx [x=0..2]

Address: 0x40010000 (0)[0], 0x40010040 (0)[1], 0x40010080 (0)[2], 0x40014000 (1)[0], 0x40014040 (1)[1], 0x40014080 (1)[2]

Access: Write-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
–	–	–	–	–	SWTRG	CLKDIS	CLKEN

- **CLKEN: Counter Clock Enable Command**

0: No effect.

1: Enables the clock if CLKDIS is not 1.

- **CLKDIS: Counter Clock Disable Command**

0: No effect.

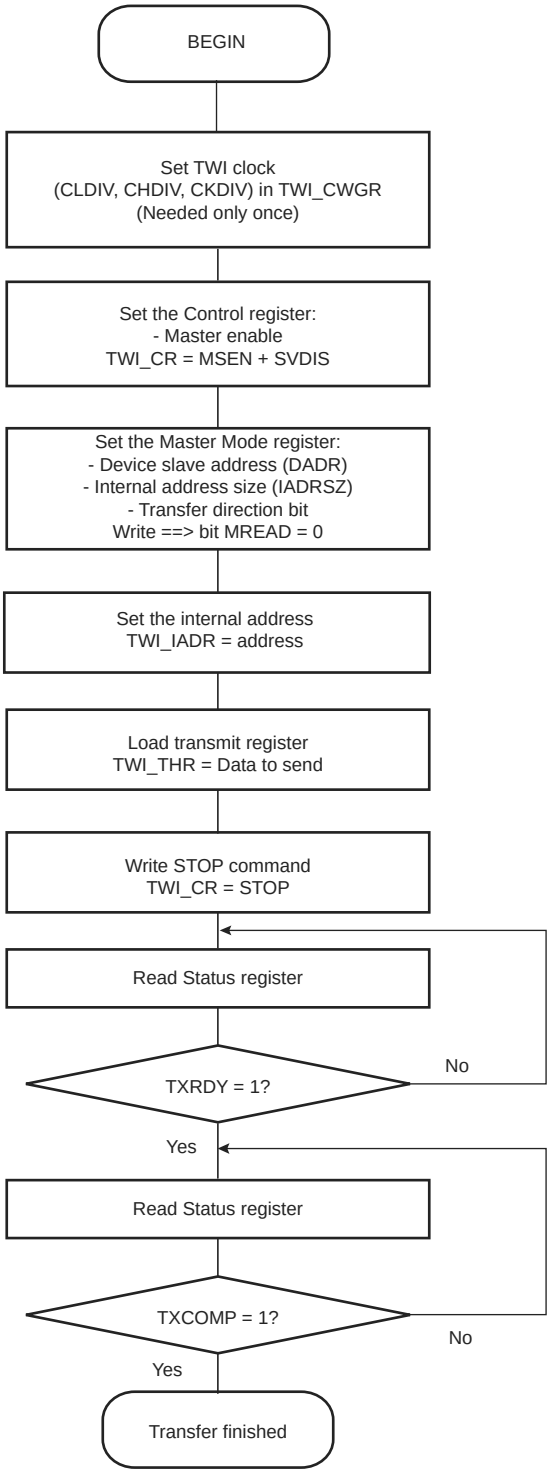
1: Disables the clock.

- **SWTRG: Software Trigger Command**

0: No effect.

1: A software trigger is performed: the counter is reset and the clock is started.

Figure 32-15. TWI Write Operation with Single Data Byte and Internal Address



32.7.9 TWI Filter Register

Name: TWI_FILTR

Address: 0x4000C644 (0), 0x40020644 (1), 0x40024644 (2), 0x40018644 (3), 0x4001C644 (4), 0x40008644 (5), 0x40040644 (6), 0x40034644 (7)

Access: Read/Write

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

- **FILT: RX Digital Filter**

0: No filtering applied on TWI inputs.

1: TWI input filtering is active. (Only in Standard and Fast modes)

Note: TWI digital input filtering follows a majority decision based on three samples from SDA/SCL lines at peripheral clock frequency.

- **PADFEN: PAD Filter Enable**

0: PAD analog filter is disabled.

1: PAD analog filter is enabled. (The analog filter must be enabled if High-speed mode is enabled.)

- **PADFCFG: PAD Filter Configuration**

See the electrical characteristics section for filter configuration details.

- **THRES: Digital Filter Threshold**

0: No filtering applied on TWI inputs.

1–7: Maximum pulse width of spikes which will be suppressed by the input filter, defined in peripheral clock cycles.

32.7.15 TWI Transmit Holding Register

Name: TWI_THR

Address: 0x4000C634 (0), 0x40020634 (1), 0x40024634 (2), 0x40018634 (3), 0x4001C634 (4), 0x40008634 (5), 0x40040634 (6), 0x40034634 (7)

Access: Write-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
TXDATA							

- **TXDATA: Master or Slave Transmit Holding Data**

35.7.1 CRCCU Descriptor Base Address Register

Name: CRCCU_DSCR

Address: 0x40048000

Access: Read/Write

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

- **DSCR: Descriptor Base Address**

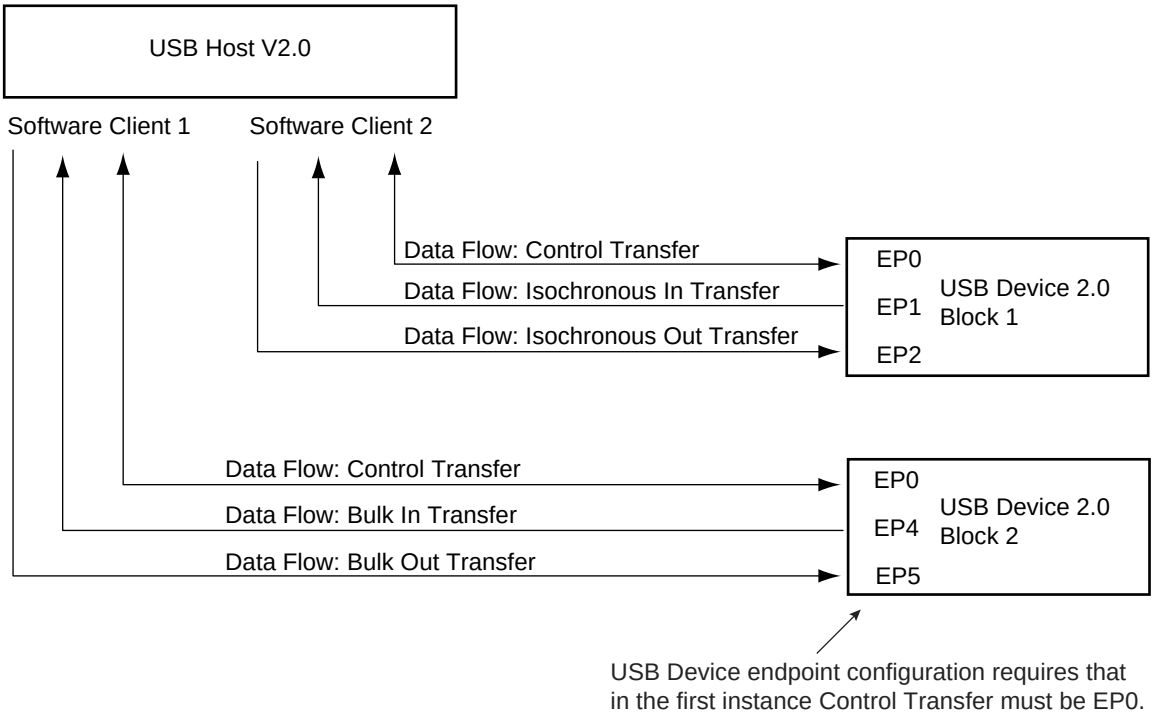
DSCR needs to be aligned with 512-byte boundaries.

37.6 Functional Description

37.6.1 USB 2.0 Full-speed Introduction

The USB 2.0 full-speed provides communication services between host and attached USB devices. Each device is offered with a collection of communication flows (pipes) associated with each endpoint. Software on the host communicates with a USB device through a set of communication flows.

Figure 37-3. Example of USB 2.0 Full-speed Communication Control



The Control Transfer endpoint EP0 is always used when a USB device is first configured (USB 2.0 specifications).

37.6.1.1 USB 2.0 Full-speed Transfer Types

A communication flow is carried over one of four transfer types defined by the USB device.

Table 37-4. USB Communication Flow

Transfer	Direction	Bandwidth	Supported Endpoint Size	Error Detection	Retrying
Control	Bidirectional	Not guaranteed	8, 16, 32, 64	Yes	Automatic
Isochronous	Unidirectional	Guaranteed	512	Yes	No
Interrupt	Unidirectional	Not guaranteed	≤ 64	Yes	Yes
Bulk	Unidirectional	Not guaranteed	8, 16, 32, 64	Yes	Yes

37.6.1.2 USB Bus Transactions

Each transfer results in one or more transactions over the USB bus. There are three kinds of transactions flowing across the bus in packets:

- Setup Transaction
- Data IN Transaction
- Data OUT Transaction

37.7.8 UDP Interrupt Clear Register

Name: UDP_ICR

Address: 0x40044020

Access: Write-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
–	–	WAKEUP	ENDBUSRES	SOFINT	EXTRSM	RXRSM	RXSUSP
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **RXSUSP: Clear UDP Suspend Interrupt**

0: No effect

1: Clears UDP Suspend Interrupt

- **RXRSM: Clear UDP Resume Interrupt**

0: No effect

1: Clears UDP Resume Interrupt

- **SOFINT: Clear Start Of Frame Interrupt**

0: No effect

1: Clears Start Of Frame Interrupt

- **ENDBUSRES: Clear End of Bus Reset Interrupt**

0: No effect

1: Clears End of Bus Reset Interrupt

- **WAKEUP: Clear Wakeup Interrupt**

0: No effect

1: Clears Wakeup Interrupt

38.6.6 Conversion Triggers

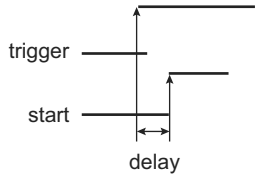
Conversions of the active analog channels are started with a software or hardware trigger. The software trigger is provided by writing the Control Register (ADC_CR) with the START bit at 1.

The list of external/internal events is provided in Section 38.7.2 “ADC Mode Register”. The hardware trigger is selected using the TRGSEL field in ADC_MR. When the TRGEN bit is set in ADC_MR, the selected hardware trigger is enabled and the software trigger is disabled.

The ADC also provides a dual trigger mode (ADC_LCTMR.DUALTRIG=1) in which the higher index channel can be sampled at a rhythm different from the other channels. The trigger of the last channel is generated by the RTC. Refer to Section 38.6.11 “Last Channel Specific Measurement Trigger”.

If a hardware trigger is selected, the start of a conversion is triggered after a delay starting at each rising edge of the selected signal. Due to asynchronous handling, the delay may vary in a range of two peripheral clock periods to one ADC clock period. This delay introduces sampling jitter in the A/D conversion process and may therefore degrade the conversion performance (e.g., SNR, THD).

Figure 38-6. Hardware Trigger Delay



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.

38.6.7 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 SLEEP bit in ADC_MR.

Sleep mode is managed by a conversion sequencer, which automatically processes the conversions of all channels at lowest power consumption.

This mode can be used when the minimum period of time between two successive trigger events is greater than the startup period of the ADC. See section “Electrical Characteristics”.

When a start conversion request occurs, the ADC is automatically activated. As the analog cell requires a startup 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 ignored.

A Fast wakeup mode is available in ADC_MR as a compromise between power-saving strategy and responsiveness. Setting the FWUP bit enables the Fast wakeup mode. In Fast wakeup mode, the ADC cell is not fully deactivated while no conversion is requested, thereby providing less power saving but faster wakeup.

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. The periodic

43.2 Revision B Parts

Errata in this section is applicable to the devices listed in the following table:

Ordering Code
ATSAMG55G19B-UUT
ATSAMG55J19B-MU
ATSAMG55J19B-MUT
ATSAMG55J19B-AU
ATSAMG55J19B-AUT

43.2.1 SUPC

Issue: No write protection on SUPC_WUIR

The System Controller Write Protection Mode register (SYSC_WPMR) does not work on the Supply Controller Wakeup Inputs Register (SUPC_WUIR). The SUPC_WUIR is not write-protected.

Workaround: None

43.2.2 TWI/TWIHS

Issue: TWI/TWIHS Clear command does not work

Bus reset using the “CLEAR” bit of the TWIHS control register does not work correctly during a bus busy state.

Workaround:

When the TWI master detects the SDA line stuck in low state the procedure to recover is:

1. Reconfigure the SDA/SCL lines as PIO.
2. Try to assert a Logic 1 on the SDA line (PIO output = 1).
3. Read the SDA line state. If the PIO state is a Logic 0 then generates a clock pulse on SCL (1-0-1 transition).
4. Read the SDA line state. If the SDA line = 0, go to Step 3; if SDA = 1, go to Step 5.
5. Generate a STOP condition.
6. Reconfigure SDA/SCL PIOs as peripheral.

43.2.3 CMCC

Issue: RAM CMCC usage limitation

The 16 Kbytes of cache RAM cannot be allocated simultaneously to the RAM Cache and to the SRAM on I/D bus. If the cache is used, up to 14 Kbytes can be unusable (depending on the selected cache size). If the cache function is required to optimize the device performances, the size must be set to 8 Kbytes.

Workaround: None