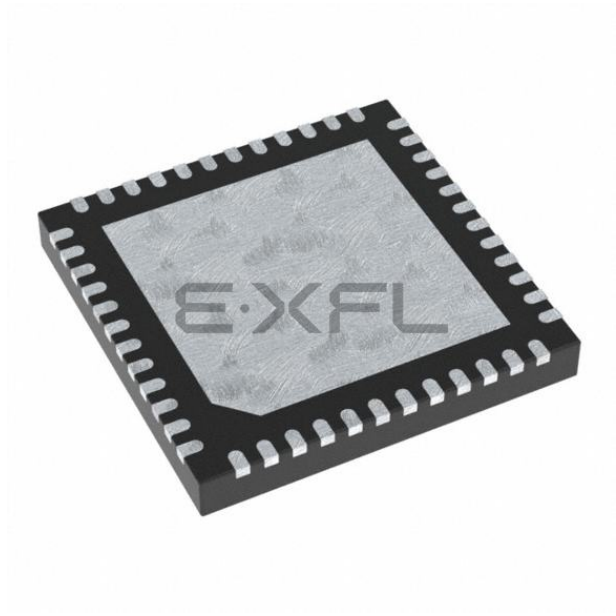


Welcome to E-XFL.COM

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	100MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	34
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 9x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4n8aa-mu

11.6.5.22 USAD8

Unsigned Sum of Absolute Differences

Syntax

```
USAD8{cond}{Rd,} Rn, Rm
```

where:

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

Rd is the destination register.

Rn is the first operand register.

Rm is the second operand register.

Operation

The USAD8 instruction:

1. Subtracts each byte of the second operand register from the corresponding byte of the first operand register.
2. Adds the absolute values of the differences together.
3. Writes the result to the destination register.

Restrictions

Do not use SP and do not use PC.

Condition Flags

These instructions do not change the flags.

Examples

```
USAD8 R1, R4, R0 ; Subtracts each byte in R0 from corresponding byte of R4
                 ; adds the differences and writes to R1
USAD8 R0, R5     ; Subtracts bytes of R5 from corresponding byte in R0,
                 ; adds the differences and writes to R0.
```

- **MEMFAULTPENDED: Memory Management Fault Exception Pending**

Read:

0: The exception is not pending.

1: The exception is pending.

Note: The user can write to these bits to change the pending status of the exceptions.

- **USGFAULTPENDED: Usage Fault Exception Pending**

Read:

0: The exception is not pending.

1: The exception is pending.

Note: The user can write to these bits to change the pending status of the exceptions.

- **SYSTICKACT: SysTick Exception Active**

Read:

0: The exception is not active.

1: The exception is active.

Note: The user can write to these bits to change the active status of the exceptions.
- Caution: A software that changes the value of an active bit in this register without a correct adjustment to the stacked content can cause the processor to generate a fault exception. Ensure that the software writing to this register retains and subsequently restores the current active status.
- Caution: After enabling the system handlers, to change the value of a bit in this register, the user must use a read-modify-write procedure to ensure that only the required bit is changed.

- **PENDSVACT: PendSV Exception Active**

0: The exception is not active.

1: The exception is active.

- **MONITORACT: Debug Monitor Active**

0: Debug monitor is not active.

1: Debug monitor is active.

- **SVCALLACT: SVC Call Active**

0: SVC call is not active.

1: SVC call is active.

- **USGFAULTACT: Usage Fault Exception Active**

0: Usage fault exception is not active.

1: Usage fault exception is active.

- **BUSFAULTACT: Bus Fault Exception Active**

0: Bus fault exception is not active.

1: Bus fault exception is active.

- **MEMFAULTACT: Memory Management Fault Exception Active**

0: Memory management fault exception is not active.

1: Memory management fault exception is active.

11.9.1.13 Configurable Fault Status Register

Name: SCB_CFSR

Access: Read-write

Reset: 0x000000000

31	30	29	28	27	26	25	24
–						DIVBYZERO	UNALIGNED
23	22	21	20	19	18	17	16
–				NOCP	INVPC	INVSTATE	UNDEFINSTR
15	14	13	12	11	10	9	8
BFRVALID	–		STKERR	UNSTKERR	IMPRECISERR	PRECISERR	IBUSERR
7	6	5	4	3	2	1	0
MMARVALID	–	MLSPERR	MSTKERR	MUNSTKERR	–	DACCVIOL	IACCVIOL

- **IACCVIOL: Instruction Access Violation Flag**

This is part of “MMFSR: Memory Management Fault Status Subregister” .

0: No instruction access violation fault.

1: The processor attempted an instruction fetch from a location that does not permit execution.

This fault occurs on any access to an XN region, even when the MPU is disabled or not present.

When this bit is 1, the PC value stacked for the exception return points to the faulting instruction. The processor has not written a fault address to the SCB_MMFAR register.

- **DACCVIOL: Data Access Violation Flag**

This is part of “MMFSR: Memory Management Fault Status Subregister” .

0: No data access violation fault.

1: The processor attempted a load or store at a location that does not permit the operation.

When this bit is 1, the PC value stacked for the exception return points to the faulting instruction. The processor has loaded the SCB_MMFAR register with the address of the attempted access.

- **MUNSTKERR: Memory Manager Fault on Unstacking for a Return From Exception**

This is part of “MMFSR: Memory Management Fault Status Subregister” .

0: No unstacking fault.

1: Unstack for an exception return has caused one or more access violations.

This fault is chained to the handler. This means that when this bit is 1, the original return stack is still present. The processor has not adjusted the SP from the failing return, and has not performed a new save. The processor has not written a fault address to the SCB_MMFAR register.

- **MSTKERR: Memory Manager Fault on Stacking for Exception Entry**

This is part of “MMFSR: Memory Management Fault Status Subregister” .

0: No stacking fault.

1: Stacking for an exception entry has caused one or more access violations.

When this bit is 1, the SP is still adjusted but the values in the context area on the stack might be incorrect. The processor has not written a fault address to SCB_MMFAR register.

Undefined	Indicates an instruction that generates an Undefined instruction exception.
Unpredictable	One cannot rely on the behavior. Unpredictable behavior must not represent security holes. Unpredictable behavior must not halt or hang the processor, or any parts of the system.
Warm reset	Also known as a core reset. Initializes the majority of the processor excluding the debug controller and debug logic. This type of reset is useful if debugging features of a processor.
Word	A 32-bit data item.
Write	Writes are defined as operations that have the semantics of a store. Writes include the Thumb instructions STM, STR, STRH, STRB, and PUSH.

16. Watchdog Timer (WDT)

16.1 Description

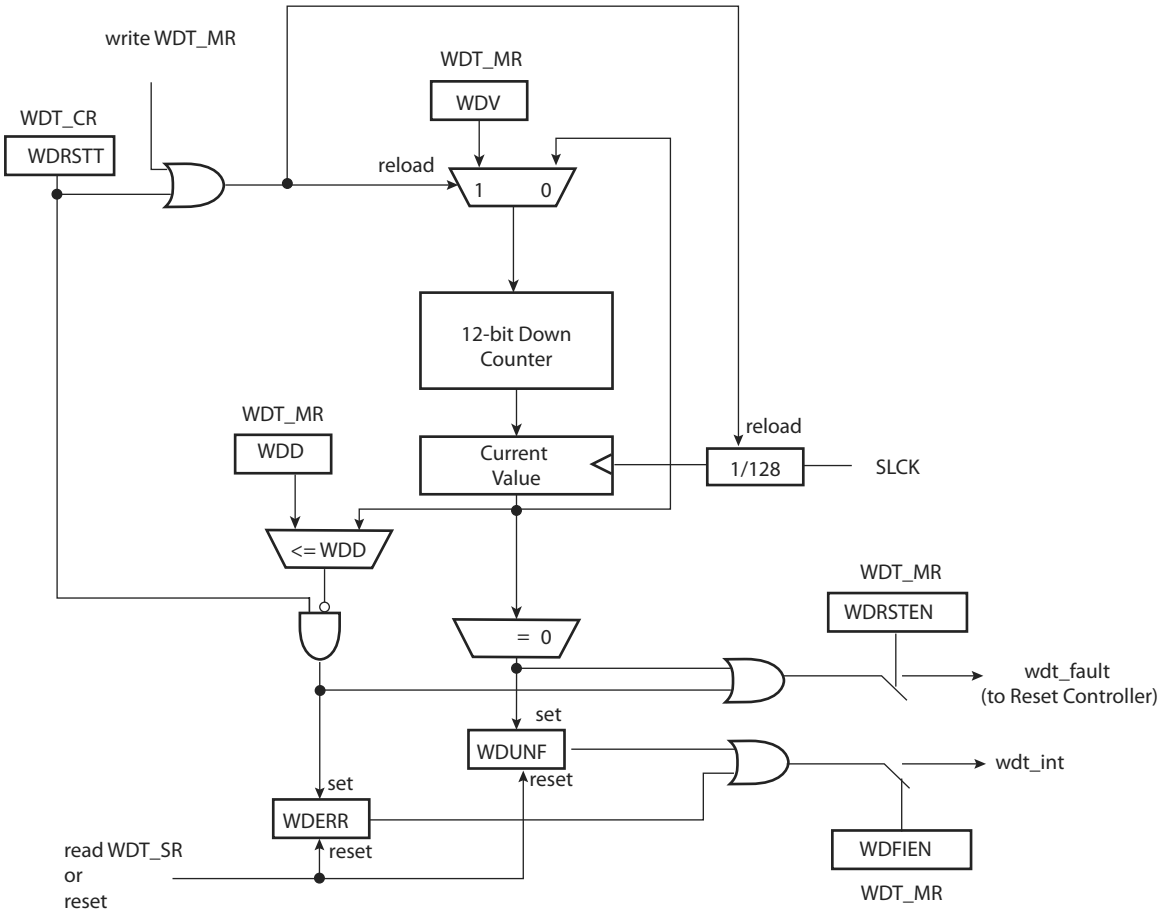
The Watchdog Timer (WDT) can be used to prevent system lock-up if the software becomes trapped in a deadlock. It features a 12-bit down counter that allows a watchdog period of up to 16 seconds (slow clock around 32 kHz). It can generate a general reset or a processor reset only. In addition, it can be stopped while the processor is in debug mode or idle mode.

16.2 Embedded Characteristics

- 12-bit key-protected programmable counter
- Watchdog Clock is independent from Processor Clock
- Provides reset or interrupt signals to the system
- Counter may be stopped while the processor is in debug state or in idle mode

16.3 Block Diagram

Figure 16-1. Watchdog Timer Block Diagram



The read operations can be performed with or without wait states. Wait states must be programmed in the field FWS (Flash Read Wait State) in the Flash Mode Register (EEFC_FMR). Defining FWS to be 0 enables the single-cycle access of the embedded Flash. Refer to the Electrical Characteristics for more details.

19.4.2.1 128-bit or 64-bit Access Mode

By default the read accesses of the Flash are performed through a 128-bit wide memory interface. It enables better system performance especially when 2 or 3 wait state needed.

For systems requiring only 1 wait state, or to privilege current consumption rather than performance, the user can select a 64-bit wide memory access via the FAM bit in the Flash Mode Register (EEFC_FMR)

Please refer to the electrical characteristics section of the product datasheet for more details.

19.4.2.2 Code Read Optimization

This feature is enabled if the EEFC_FMR register bit SCOD is cleared.

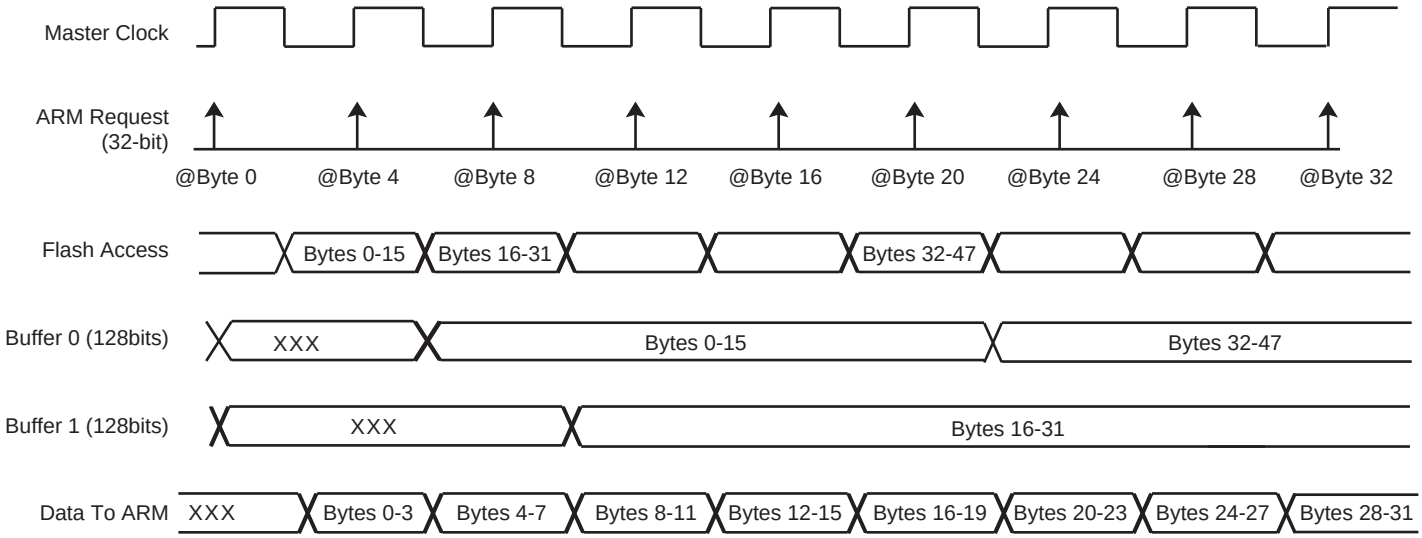
A system of 2 x 128-bit or 2 x 64-bit buffers is added in order to optimize sequential Code Fetch.

Note: Immediate consecutive code read accesses are not mandatory to benefit from this optimization.

The sequential code read optimization is enabled by default. If the bit SCOD in Flash Mode Register (EEFC_FMR) is set to 1, these buffers are disabled and the sequential code read is not optimized anymore.

Another system of 2 x 128-bit or 2 x 64-bit buffers is added in order to optimize loop code fetch (see “Code Loops Optimization” on page 331).

Figure 19-2. Code Read Optimization for FWS = 0



Note: When FWS is equal to 0, all the accesses are performed in a single-cycle access.

24.5.4 Main Clock Oscillator Selection

The user can select either the 12/8/4 MHz Fast RC Oscillator or the 3 to 20 MHz Crystal or Ceramic Resonator-based oscillator to be the source of Main Clock.

The advantage of the 12/8/4 MHz Fast RC Oscillator is that it provides fast start-up time, this is why it is selected by default (to start up the system) and when entering Wait Mode.

The advantage of the 3 to 20 MHz Crystal or Ceramic Resonator-based oscillator is that it is very accurate.

The selection is made by writing the MOSCSEL bit in the Main Oscillator Register (CKGR_MOR). The switch of the Main Clock source is glitch free, so there is no need to run out of SLCK, PLLACK in order to change the selection. The MOSCSELS bit of the Power Management Controller Status Register (PMC_SR) allows knowing when the switch sequence is done.

Setting the MOSCSELS bit in PMC_IMR can trigger an interrupt to the processor.

Enabling the Fast RC Oscillator (MOSCRCEN = 1) and changing the Fast RC Frequency (MOSCCRF) at the same time is not allowed.

The Fast RC must be enabled first and its frequency changed in a second step.

24.5.5 Software Sequence to Detect the Presence of Fast Crystal

The frequency meter carried on the CKGR_MCFR register is operating on the selected main clock and not on the fast crystal clock nor on the fast RC Oscillator clock.

Therefore, to check for the presence of the fast crystal clock, it is necessary to have the main clock (MAINCK) driven by the fast crystal clock (MOSCSEL=1).

The following software sequence order must be followed:

- MCK must select the slow clock (CSS=0 in the PLL_MCKR register).
- Wait for the MCKRDY flag in the PLL_SR register to be 1.
- The fast crystal must be enabled by programming 1 in the MOSCXTEN field in the CKGR_MOR register with the MOSCXTST field being programmed to the appropriate value (see the Electrical Characteristics chapter).
- Wait for the MOSCXTS flag to be 1 in the PLL_SR register to get the end of a start-up period of the fast crystal oscillator.
- Then, MOSCSEL must be programmed to 1 in the CKGR_MOR register to select fast main crystal oscillator for the main clock.
- MOSCSEL must be read until its value equals 1.
- Then the MOSCSELS status flag must be checked in the PLL_SR register.

At this point, 2 cases may occur (either MOSCSELS = 0 or MOSCSELS = 1).

- If MOSCSELS = 1, there is a valid crystal connected and its frequency can be determined by initiating a frequency measure by programming RCMEAS in the CKGR_MCFR register.
- If MOSCSELS = 0, there is no fast crystal clock (either no crystal connected or a crystal clock out of specification).
  A frequency measure can reinforce this status by initiating a frequency measure by programming RCMEAS in the CKGR_MCFR register.
- If MOSCSELS=0, the selection of the main clock must be programmed back to the main RC oscillator by writing MOSCSEL to 0 prior to disabling the fast crystal oscillator.
- If MOSCSELS=0, the crystal oscillator can be disabled (MOSCXTEN=0 in the CKGR_MOR register).

25.16.2 PMC System Clock Disable Register

Name: PMC_SCDR

Address: 0x400E0404

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
–	–	–	–	–	PCK2	PCK1	PCK0
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

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

- **PCKx: Programmable Clock x Output Disable**

0 = No effect.

1 = Disables the corresponding Programmable Clock output.

27.7.23 PIO Pull Up Status Register

Name: PIO_PUSR

Address: 0x400E0E68 (PIOA), 0x400E1068 (PIOB), 0x400E1268 (PIOC)

Access: Read-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: Pull Up Status.**

0: Pull Up resistor is enabled on the I/O line.

1: Pull Up resistor is disabled on the I/O line.

27.7.24 PIO Peripheral ABCD Select Register 1

Name: PIO_ABCDSR1

Access: Read-write

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

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

- **P0-P31: Peripheral Select.**
  If the same bit is set to 0 in PIO_ABCDSR2:

0: Assigns the I/O line to the Peripheral A function.

1: Assigns the I/O line to the Peripheral B function.
  If the same bit is set to 1 in PIO_ABCDSR2:

0: Assigns the I/O line to the Peripheral C function.

1: Assigns the I/O line to the Peripheral D function.

27.7.40 PIO Level Select Register

Name: PIO_LSR

Address: 0x400E0EC4 (PIOA), 0x400E10C4 (PIOB), 0x400E12C4 (PIOC)

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: Level Interrupt Selection**

0: No effect.

1: The interrupt source is a Level detection event.

28.7.3.10 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/NSS signal. In this case, multi-master configuration, NPCS0, MOSI, MISO and SPCK pins must be configured in open drain (through the PIO controller). 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 automatically disabled until re-enabled by writing the SPIEN bit in the SPI_CR (Control Register) at 1.

By default, the Mode Fault detection circuitry is enabled. The user can disable Mode Fault detection by setting the MODFDIS bit in the SPI Mode Register (SPI_MR).

28.7.4 SPI Slave Mode

When operating in Slave Mode, the SPI processes data bits on the clock provided on the SPI clock pin (SPCK).

The SPI waits for NSS to go active before receiving the serial clock from an external master. When NSS falls, the clock is validated on the serializer, which processes the number of bits defined by the BITS field of the Chip Select Register 0 (SPI_CSR0). These bits are processed following a phase and a polarity defined respectively by the NCPHA and CPOL bits of the SPI_CSR0. Note that BITS, CPOL and NCPHA of the other Chip Select Registers have no effect when the SPI is programmed in Slave Mode.

The bits are shifted out on the MISO line and sampled on the MOSI line.

(For more information on BITS field, see also, the (Note:) below the register table; Section 28.8.9 “SPI Chip Select Register” on page 530.)

When all the bits are processed, the received data is transferred in the Receive Data Register and the RDRF bit rises. If the SPI_RDR (Receive Data Register) has not been read before new data is received, the Overrun Error bit (OVRES) in SPI_SR is set. As long as this flag is set, data is loaded in SPI_RDR. The user has to read the status register to clear the OVRES bit.

When a transfer starts, the data shifted out is the data present in the Shift Register. If no data has been written in the Transmit Data Register (SPI_TDR), the last data received is transferred. If no data has been received since the last reset, all bits are transmitted low, as the Shift Register resets at 0.

When a first data is written in SPI_TDR, it is transferred immediately in the Shift Register and the TDRE bit rises. If new data is written, it remains in SPI_TDR until a transfer occurs, i.e. NSS falls and there is a valid clock on the SPCK pin. When the transfer occurs, the last data written in SPI_TDR is transferred in the Shift Register and the TDRE bit rises. This enables frequent updates of critical variables with single transfers.

Then, a new data is loaded in the Shift Register from the Transmit Data Register. In case no character is ready to be transmitted, i.e. no character has been written in SPI_TDR since the last load from SPI_TDR to the Shift Register, the SPI_TDR is retransmitted. In this case the Underrun Error Status Flag (UNDES) is set in the SPI_SR.

30.5.3 Transmitter

30.5.3.1 Transmitter Reset, Enable and Disable

After device reset, the UART transmitter is disabled and it must be enabled before being used. The transmitter is enabled by writing the control register UART_CR with the bit TXEN at 1. From this command, the transmitter waits for a character to be written in the Transmit Holding Register (UART_THR) before actually starting the transmission.

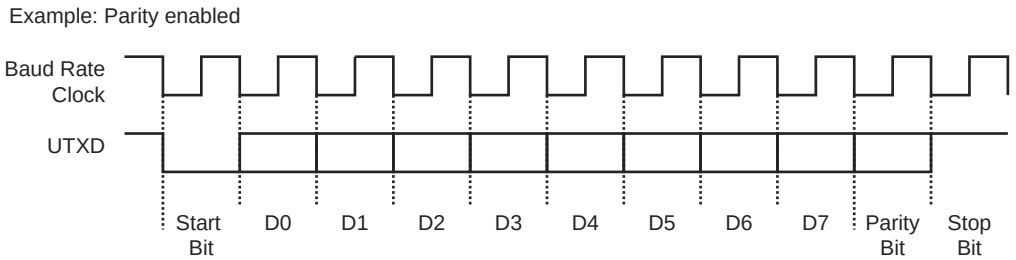
The programmer can disable the transmitter by writing UART_CR with the bit TXDIS at 1. If the transmitter is not operating, it is immediately stopped. However, if a character is being processed into the Shift Register and/or a character has been written in the Transmit Holding Register, the characters are completed before the transmitter is actually stopped.

The programmer can also put the transmitter in its reset state by writing the UART_CR with the bit RSTTX at 1. This immediately stops the transmitter, whether or not it is processing characters.

30.5.3.2 Transmit Format

The UART transmitter drives the pin UTXD at the baud rate clock speed. The line is driven depending on the format defined in the Mode Register and the data stored in the Shift Register. One start bit at level 0, then the 8 data bits, from the lowest to the highest bit, one optional parity bit and one stop bit at 1 are consecutively shifted out as shown in the following figure. The field PARE in the mode register UART_MR defines whether or not a parity bit is shifted out. When a parity bit is enabled, it can be selected between an odd parity, an even parity, or a fixed space or mark bit.

Figure 30-9. Character Transmission



30.5.3.3 Transmitter Control

When the transmitter is enabled, the bit TXRDY (Transmitter Ready) is set in the status register UART_SR. The transmission starts when the programmer writes in the Transmit Holding Register (UART_THR), and after the written character is transferred from UART_THR to the Shift Register. The TXRDY bit remains high until a second character is written in UART_THR. As soon as the first character is completed, the last character written in UART_THR is transferred into the shift register and TXRDY rises again, showing that the holding register is empty.

When both the Shift Register and UART_THR are empty, i.e., all the characters written in UART_THR have been processed, the TXEMPTY bit rises after the last stop bit has been completed.

31.7 Functional Description

The USART is capable of managing several types of serial synchronous or asynchronous communications.

It supports the following communication modes:

- 5- to 9-bit full-duplex asynchronous serial communication
  - MSB- or LSB-first
  - 1, 1.5 or 2 stop bits
  - Parity even, odd, marked, space or none
  - By 8 or by 16 over-sampling receiver frequency
  - Optional hardware handshaking
  - Optional break management
  - Optional multidrop serial communication
- High-speed 5- to 9-bit full-duplex synchronous serial communication
  - MSB- or LSB-first
  - 1 or 2 stop bits
  - Parity even, odd, marked, space or none
  - By 8 or by 16 over-sampling frequency
  - Optional hardware handshaking
  - Optional break management
  - Optional multidrop serial communication
- RS485 with driver control signal
- ISO7816, T0 or T1 protocols for interfacing with smart cards
  - NACK handling, error counter with repetition and iteration limit, inverted data.
- InfraRed IrDA Modulation and Demodulation
- SPI Mode
  - Master or Slave
  - Serial Clock Programmable Phase and Polarity
  - SPI Serial Clock (SCK) Frequency up to Internal Clock Frequency MCK/6
- Test modes
  - Remote loopback, local loopback, automatic echo

31.7.1 Baud Rate Generator

The Baud Rate Generator provides the bit period clock named the Baud Rate Clock to both the receiver and the transmitter.

The Baud Rate Generator clock source can be selected by setting the USCLKS field in the Mode Register (US_MR) between:

- The Master Clock MCK
- A division of the Master Clock, the divider being product dependent, but generally set to 8
- The external clock, available on the SCK pin

The Baud Rate Generator is based upon a 16-bit divider, which is programmed with the CD field of the Baud Rate Generator Register (US_BRGR). If CD is programmed to 0, the Baud Rate Generator does not generate any clock. If CD is programmed to 1, the divider is bypassed and becomes inactive.

If the external SCK clock is selected, the duration of the low and high levels of the signal provided on the SCK pin must be longer than a Master Clock (MCK) period. The frequency of the signal provided on SCK must be at least 3 times lower than MCK in USART mode, or 6 times lower in SPI mode.

32. Timer Counter (TC)

32.1 Description

A Timer Counter (TC) module includes three identical TC channels. The number of implemented TC modules is device-specific.

Each TC channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation.

Each channel has three external clock inputs, five internal clock inputs and two multi-purpose input/output signals which can be configured by the user. Each channel drives an internal interrupt signal which can be programmed to generate processor interrupts.

The TC embeds a quadrature decoder (QDEC) connected in front of the timers and driven by TIOA0, TIOB0 and TIOB1 inputs. When enabled, the QDEC performs the input lines filtering, decoding of quadrature signals and connects to the timers/counters in order to read the position and speed of the motor through the user interface.

The TC block has two global registers which act upon all TC channels:

- Block Control Register (TC_BCR)—allows channels to be started simultaneously with the same instruction
- Block Mode Register (TC_BMR)—defines the external clock inputs for each channel, allowing them to be chained

32.2 Embedded Characteristics

- Total number of TC channels: 6
- TC channel size: 16-bit
- Wide range of functions including:
  - Frequency measurement
  - Event counting
  - Interval measurement
  - Pulse generation
  - Delay timing
  - Pulse Width Modulation
  - Up/down capabilities
  - Quadrature decoder
  - 2-bit gray up/down count for stepper motor
- Each channel is user-configurable and contains:
  - Three external clock inputs
  - Five Internal clock inputs
  - Two multi-purpose input/output signals acting as trigger event
- Internal interrupt signal
- Read of the Capture registers by the PDC
- Compare event fault generation for PWM
- Register Write Protection

- **ETRGS: External Trigger**

0: The External Trigger Interrupt is disabled.

1: The External Trigger Interrupt is enabled.

- **ENDRX: End of Receiver Transfer**

0: The PDC Receive End of Transfer Interrupt is disabled.

1: The PDC Receive End of Transfer Interrupt is enabled.

- **RXBUFF: Reception Buffer Full**

0: The PDC Receive Buffer Full Interrupt is disabled.

1: The PDC Receive Buffer Full Interrupt is enabled.

32.7.13 TC Block Control Register

Name: TC_BCR

Address: 0x400100C0 (0), 0x400140C0 (1)

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
–	–	–	–	–	–	–	SYNC

- **SYNC: Synchro Command**

0: No effect.

1: Asserts the SYNC signal which generates a software trigger simultaneously for each of the channels.

33.7.10 PWM Channel Duty Cycle Register

Name: PWM_CDTY[0..3]

Address: 0x40020204 [0], 0x40020224 [1], 0x40020244 [2], 0x40020264 [3]

Access: Read/Write

31	30	29	28	27	26	25	24
CDTY							

23	22	21	20	19	18	17	16
CDTY							

15	14	13	12	11	10	9	8
CDTY							

7	6	5	4	3	2	1	0
CDTY							

Only the first 16 bits (internal channel counter size) are significant.

- **CDTY: Channel Duty Cycle**

Defines the waveform duty cycle. This value must be defined between 0 and CPRD (PWM_CPRx).

35.6.4 Conversion FIFO

To provide flexibility and high efficiency, a 4 half-word FIFO is used to handle the data to be converted.

As long as the TXRDY flag in the DACC Interrupt Status Register is active the DAC Controller is ready to accept conversion requests by writing data in the DACC Conversion Data Register (DACC_CDR). Data which cannot be converted immediately are stored in the DACC FIFO.

When the FIFO is full or the DACC is not ready to accept conversion requests, the TXRDY flag is inactive.

Warning: Writing in the DACC_CDR register while TXRDY flag is inactive will corrupt FIFO data.

35.6.5 Conversion Width

The WORD field of the DACC Mode Register allows the user to switch between half-word and word transfer.

In half-word transfer mode only one 10-bit data item is sampled (DACC_MR[9:0]) per DACC_CDR register write.

In word transfer mode each time the DACC_CDR register is written 2 data items are sampled. First data item sampled for conversion will be DACC_CDR[9:0] and the second DACC_CDR[25:16].

35.6.6 DAC Timings

The DAC startup time must be defined by the user in the STARTUP field of the DACC Mode Register.

The DAC maximum clock frequency is 500 KHz, therefore the internal trigger period can be configured through the CLKDIV field of the DACC Mode Register.

35.6.7 Write Protection Registers

In order to bring security to the DACC, a write protection system has been implemented.

The write protection mode prevents the write of the DACC Mode Register. When this mode is enabled and the protected register is written an error is generated in the DACC Write Protect Status Register and the register write request is canceled. When a write protection error occurs, the WPROTERR flag is set and the address of the corresponding canceled register write is available in the WPROTADRR field of the DACC Write Protect Status Register.

Due to the nature of the write protection feature, enabling and disabling the write protection mode requires the use of a security code. Thus when enabling or disabling the write protection mode, the WPKEY field of the DACC Write Protect Mode Register must be filled with the “DAC” ASCII code (corresponding to 0x444143) otherwise the register write will be canceled.

Table 37-20. 48-lead LQFP Package Dimensions (in mm)

Symbol	Millimeter			Inch		
	Min	Nom	Max	Min	Nom	Max
A	–	–	1.60	–	–	0.063
A1	0.05	–	0.15	0.002	–	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
D	9.00 BSC			0.354SC		
D1	7.00 BSC			0.276 BSC		
E	9.00 BSC			0.354 BSC		
E1	7.00 BSC			0.276 BSC		
R2	0.08	–	0.20	0.003	–	0.008
R1	0.08	–	–	0.003	–	–
q	0°	3.5°	7°	0°	3.5°	7°
θ_1	0°	–	–	0°	–	–
θ_2	11°	12°	13°	11°	12°	13°
θ_3	11°	12°	13°	11°	12°	13°
c	0.09	–	0.20	0.004	–	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 REF			0.039 REF		
S	0.20	–	–	0.008	–	–
b	0.17	0.20	0.27	0.007	0.008	0.011
e	0.50 BSC.			0.020 BSC.		
D2	5.50			0.217		
E2	5.50			0.217		
Tolerances of Form and Position						
aaa	0.20			0.008		
bbb	0.20			0.008		
ccc	0.08			0.003		
ddd	0.08			0.003		

Table 37-21. Device and 48-lead LQFP Package Maximum Weight

SAM4N	190	mg

Table 37-22. 48-lead LQFP Package Characteristics

Moisture Sensitivity Level	3

Table 37-23. 48-lead LQFP Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	e3

This package respects the recommendations of the NEMI User Group.