**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.

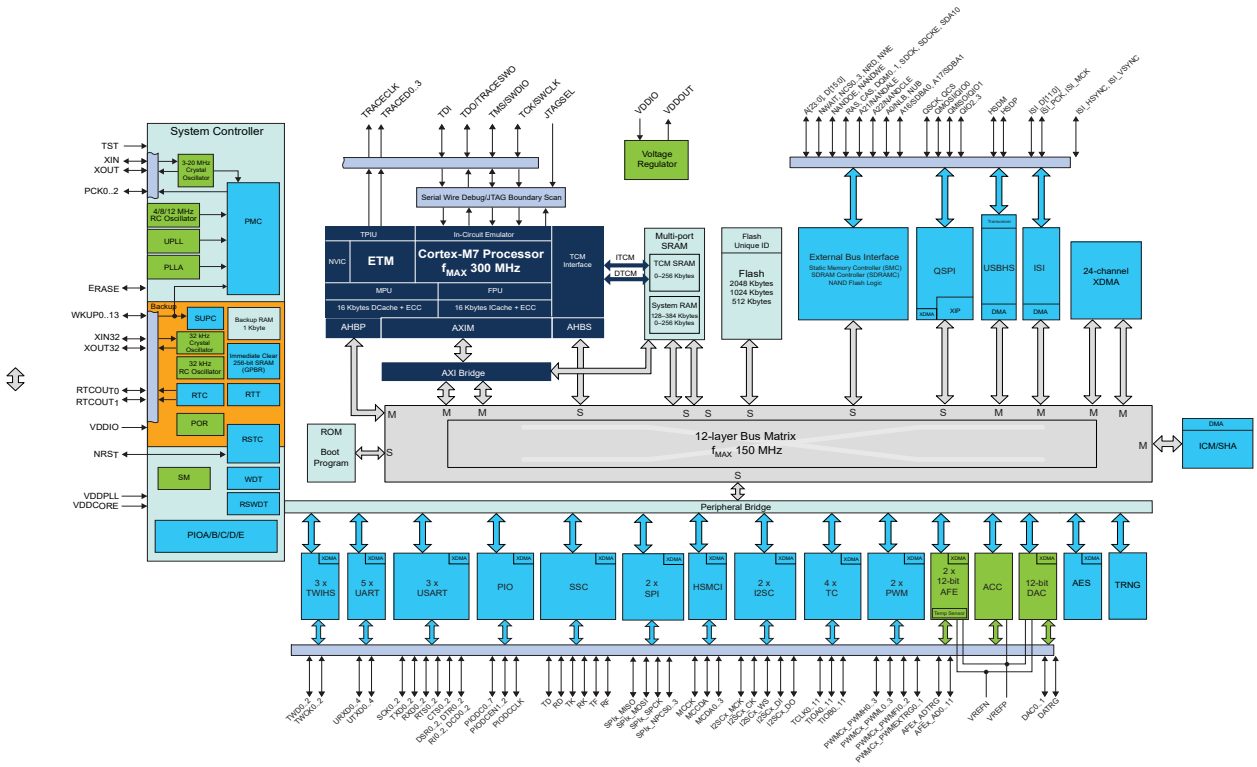
## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | ARM® Cortex®-M7 |
| Core Size | 32-Bit Single-Core |
| Speed | 300MHz |
| Connectivity | CANbus, EBI/EMI, I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, SSC, UART/USART, USB |
| Peripherals | Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT |
| Number of I/O | 114 |
| Program Memory Size | 512KB (512K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 256K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 3.6V |
| Data Converters | A/D 24x12b; D/A 2x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 144-TFBGA |
| Supplier Device Package | 144-TFBGA (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/atsamv70q19b-cbt |

# 3. Block Diagram

Refer to the table 1. Configuration Summary for detailed configurations of memory size, package and features of the SAM E70/S70/V70/V71 devices.

**Figure 3-1. SAM S70 144-pin Block Diagram**



## 19.4 Register Summary

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | MATRIX_MCFG0 | 7:0 | | | | | | ULBT[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | MATRIX_MCFG1 | 7:0 | | | | | | ULBT[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | MATRIX_MCFG2 | 7:0 | | | | | | ULBT[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | MATRIX_MCFG3 | 7:0 | | | | | | ULBT[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | MATRIX_MCFG4 | 7:0 | | | | | | ULBT[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | MATRIX_MCFG5 | 7:0 | | | | | | ULBT[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | MATRIX_MCFG6 | 7:0 | | | | | | ULBT[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | MATRIX_MCFG7 | 7:0 | | | | | | ULBT[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | MATRIX_MCFG8 | 7:0 | | | | | | ULBT[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | MATRIX_MCFG9 | 7:0 | | | | | | ULBT[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x28 | MATRIX_MCFG10 | 7:0 | | | | | | ULBT[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |

| Value | Description |
|---|---|
| 0 | MATRIX dynamic clock gating enabled. The MATRIX circuitry is driven by the clock only when a transfer to a peripheral is being performed. Power consumption is optimized. |
| 1 | MATRIX dynamic clock gating disabled. The MATRIX circuitry is always driven by the clock in Active mode. |

### 26.4.5.3 RSTC Mode Register

**Name:** RSTC_MR
**Offset:** 0x08
**Reset:** 0x00000001
**Property:** Read/Write

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

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | KEY[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | ERSTL[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | URSTIEN | | | | URSTEN |
| Access | | | | R/W | | | | R/W |
| Reset | | | | 0 | | | | 1 |

**Bits 31:24 – KEY[7:0]** Write Access Password

| Value | Name | Description |
|---|---|---|
| 0xA5 | PASSWD | Writing any other value in this field aborts the write operation. Always reads as 0. |

**Bits 11:8 – ERSTL[3:0]** External Reset Length
This field defines the external reset length. The external reset is asserted during a time of $2^{(ERSTL+1)}$ SLCK cycles. This allows assertion duration to be programmed between 60 µs and 2 seconds. Note that synchronization cycles must also be considered when calculating the actual reset length as previously described.

**Bit 4 – URSTIEN** User Reset Interrupt Enable

| Value | Description |
|---|---|
| 0 | RSTC_SR.USRTS at ‘1’ has no effect on the RSTC interrupt line. |
| 1 | RSTC_SR.USRTS at ‘1’ asserts the RSTC interrupt line if URSTEN = 0. |

**Bit 0 – URSTEN** User Reset Enable

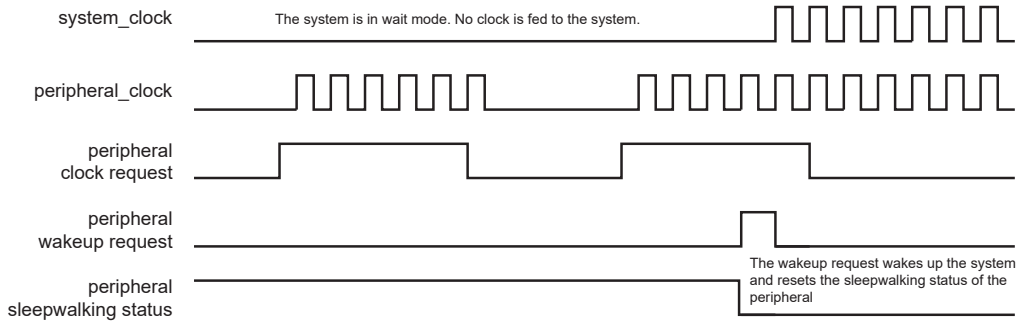
Only the following peripherals can be configured with asynchronous partial wakeup: UARTx and TWIHSx.

The peripheral selected for asynchronous partial wakeup must first be configured so that its clock is enabled. To do so, write a ‘1’ to the appropriate PIDx bit in PMC_PCER registers.

### 31.10.2 Asynchronous Partial Wakeup in Wait Mode (SleepWalking)

When the system is in Wait mode, all clocks of the system except SLCK are stopped. When an asynchronous clock request from a peripheral occurs, the PMC partially wakes up the system to feed the clock only to this peripheral. The rest of the system is not fed with the clock, thus optimizing power consumption. Finally, depending on user-configurable conditions, the peripheral either wakes up the whole system if these conditions are met or stops the peripheral clock until the next clock request. If a wakeup request occurs, SleepWalking is automatically disabled until the user instructs the PMC to enable SleepWalking. This is done by writing a ‘1’ to PIDx in the PMC SleepWalking Enable register (PMC_SLPWK_ER).

**Figure 31-2. SleepWalking Waveforms**




#### 31.10.2.1 Configuration Procedure

Before configuring SleepWalking for a peripheral, check that the PIDx bit in PMC_PCSR is set. This ensures that the peripheral clock is enabled.

The steps to enable SleepWalking for a peripheral are the following:

1. Check that the corresponding PIDx bit in the PMC SleepWalking Activity Status register (PMC_SLPWK_ASR) is set to ‘0’. This ensures that the peripheral has no activity in progress.
2. Enable SleepWalking for the peripheral by writing a ‘1’ to the corresponding PIDx bit in the PMC_SLPWK_ER.
3. Check that the corresponding PIDx bit in PMC_SLPWK_ASR is set to ‘0’. This ensures that no activity has started during the enable phase.
4. In the PMC_SLPWK_ASR, if the corresponding PIDx bit is set, SleepWalking must be immediately disabled by writing a ‘1’ to the PIDx bit in the PMC SleepWalking Disable register (PMC_SLPWK_DR). Wait for the end of peripheral activity before reinitializing the procedure.
   If the corresponding PIDx bit is set to ‘0’, then the peripheral clock is disabled and the system can then be placed in Wait mode.

Before entering Wait mode, check that the AIP bit in the PMC SleepWalking Activity In Progress Register (PMC_SLPWK_AIPR) is cleared. This ensures that none of the peripherals is currently active.

Note: When SleepWalking for a peripheral is enabled and the core is running (system not in Wait mode), the peripheral must not be accessed before a wakeup of the peripheral is performed.

### 38.8.13 GMAC Interrupt Mask Register

**Name:** GMAC_IMR
**Offset:** 0x030
**Reset:** 0x07FFFFFF
**Property:** Read/Write

This register is a read-only register indicating which interrupts are masked. All bits are set at Reset and can be reset individually by writing to the Interrupt Enable Register (GMAC_IER), or set individually by writing to the Interrupt Disable Register (GMAC_IDR).

For test purposes there is a write-only function to this register that allows the bits in the Interrupt Status Register to be set or cleared, regardless of the state of the mask register. A write to this register directly affects the state of the corresponding bit in the Interrupt Status Register, causing an interrupt to be generated if a 1 is written.

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

0: The corresponding interrupt is enabled.

1: The corresponding interrupt is not enabled.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | TSUTIMCMP | WOL | RXLPISBC | SRI | PDRSFT | PDRQFT |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 1 | 1 | 1 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PDRSFR | PDRQFR | SFT | DRQFT | SFR | DRQFR | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | EXINT | PFTR | PTZ | PFNZ | HRESP | ROVR | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TCOMP | TFC | RLEX | TUR | TXUBR | RXUBR | RCOMP | MFS |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bit 29 – TSUTIMCMP** TSU Timer Comparison

**Bit 28 – WOL** Wake On LAN

**Bit 27 – RXLPISBC** Receive LPI indication Status Bit Change
Receive LPI indication status bit change.

Cleared on read.

**Bit 26 – SRI** TSU Seconds Register Increment

**Bit 25 – PDRSFT** PDelay Response Frame Transmitted

**Bit 24 – PDRQFT** PDelay Request Frame Transmitted

**Bit 23 – PDRSFR** PDelay Response Frame Received

**Bit 22 – PDRQFR** PDelay Request Frame Received

**Bit 21 – SFT** PTP Sync Frame Transmitted

**Bit 20 – DRQFT** PTP Delay Request Frame Transmitted

**Bit 19 – SFR** PTP Sync Frame Received

**Bit 18 – DRQFR** PTP Delay Request Frame Received

**Bit 15 – EXINT** External Interrupt

**Bit 14 – PFTR** Pause Frame Transmitted

**Bit 13 – PTZ** Pause Time Zero

**Bit 12 – PFNZ** Pause Frame with Non-zero Pause Quantum Received

**Bit 11 – HRESP** HRESP Not OK

**Bit 10 – ROVR** Receive Overrun

**Bit 7 – TCOMP** Transmit Complete

**Bit 6 – TFC** Transmit Frame Corruption Due to AHB Error

**Bit 5 – RLEX** Retry Limit Exceeded

**Bit 4 – TUR** Transmit Underrun

**Bit 3 – TXUBR** TX Used Bit Read

**Bit 2 – RXUBR** RX Used Bit Read

**Bit 1 – RCOMP** Receive Complete

**Bit 0 – MFS** Management Frame Sent

The bank is really killed: USBHS_DEVEPTISRx.NBUSYBK is decremented.

The bank is not cleared but sent (IN transfer): USBHS_DEVEPTISRx.NBUSYBK is decremented.

The bank is not cleared because it was empty.

The user should wait for this bit to be cleared before trying to kill another packet.

This kill request is refused if at the same time an IN token is coming and the last bank is the current one being sent on the USB line. If at least two banks are ready to be sent, there is no problem to kill a packet even if an IN token is coming. Indeed, in this case, the current bank is sent (IN transfer) while the last bank is killed.

**Bit 12 – NBUSYBKE** Number of Busy Banks Interrupt

| Value | Description |
|---|---|
| 0 | Cleared when USBHS_DEVEPTIDRx.NBUSYBKEC = 0. This disables the Number of Busy Banks interrupt (USBHS_DEVEPTISRx.NBUSYBK). |
| 1 | Set when the USBHS_DEVEPTIERx.NBUSYBKES = 1. This enables the Number of Busy Banks interrupt (USBHS_DEVEPTISRx.NBUSYBK). |

**Bit 7 – SHORTPACKETE** Short Packet Interrupt
If this bit is set for non-control IN endpoints, a short packet transmission is guaranteed upon ending a DMA transfer, thus signaling an end of isochronous frame or a bulk or interrupt end of transfer, provided that the End of DMA Buffer Output Enable (END_B_EN) bit and the Automatic Switch (AUTOSW) = 1.

| Value | Description |
|---|---|
| 0 | Cleared when USBHS_DEVEPTIDRx.SHORTPACKETEC = 1. This disables the Short Packet interrupt (USBHS_DEVEPTISRx.SHORTPACKET). |
| 1 | Set when USBHS_DEVEPTIERx.SHORTPACKETES = 1. This enables the Short Packet interrupt (USBHS_DEVEPTISRx.SHORTPACKET). |

**Bit 6 – STALLEDE** STALLed Interrupt

| Value | Description |
|---|---|
| 0 | Cleared when USBHS_DEVEPTIDRx.STALLEDEC = 1. This disables the STALLed interrupt (USBHS_DEVEPTISRx.STALLEDI). |
| 1 | Set when USBHS_DEVEPTIERx.STALLEDES = 1. This enables the STALLed interrupt (USBHS_DEVEPTISRx.STALLEDI). |

**Bit 5 – OVERFE** Overflow Interrupt

| Value | Description |
|---|---|
| 0 | Cleared when USBHS_DEVEPTIDRx.OVERFEC = 1. This disables the Overflow interrupt (USBHS_DEVEPTISRx.OVERFI). |
| 1 | Set when USBHS_DEVEPTIERx.OVERFES = 1. This enables the Overflow interrupt (USBHS_DEVEPTISRx.OVERFI). |

**Bit 4 – NAKINE** NAKed IN Interrupt

| Value | Description |
|---|---|
| 0 | No PID error occurred since last clear of this bit. |
| 1 | This bit is automatically set when a PID error has been detected. |

**Bit 1 – DATAPID** Data PID Error

| Value | Description |
|---|---|
| 0 | No Data PID error occurred since last clear of this bit. |
| 1 | This bit is automatically set when a Data PID error has been detected. |

**Bit 0 – DATATGL** Data Toggle Error

| Value | Description |
|---|---|
| 0 | No Data Toggle error occurred since last clear of this bit. |
| 1 | This bit is automatically set when a Data Toggle error has been detected. |

**Bit 7 – UNRE** Underrun Error Interrupt Disable

**Bit 6 – OVRE** Overrun Error Interrupt Disable

**Bit 5 – GACC** General Call Access Interrupt Disable

**Bit 4 – SVACC** Slave Access Interrupt Disable

**Bit 2 – TXRDY** Transmit Holding Register Ready Interrupt Disable

**Bit 1 – RXRDY** Receive Holding Register Ready Interrupt Disable

**Bit 0 – TXCOMP** Transmission Completed Interrupt Disable

**Bit 7 – TXDIS** Transmitter Disable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | Disables the transmitter. |

**Bit 6 – TXEN** Transmitter Enable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | Enables the transmitter if TXDIS is 0. |

**Bit 5 – RXDIS** Receiver Disable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | Disables the receiver. |

**Bit 4 – RXEN** Receiver Enable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | Enables the receiver, if RXDIS is 0. |

**Bit 3 – RSTTX** Reset Transmitter

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | Resets the transmitter. |

**Bit 2 – RSTRX** Reset Receiver

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | Resets the receiver. |

**Bit 5 – OVRE** Overrun Error Interrupt Enable

**Bit 1 – TXRDY** TXRDY Interrupt Enable

**Bit 0 – RXRDY** RXRDY Interrupt Enable

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ...<br>0x7F | | | | | | | | | | |
| 0x80 | MLB_HCTL | 7:0 | | | | | | | RST1 | RST0 |
| | | 15:8 | EN | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x84<br>...<br>0x87 | Reserved | | | | | | | | | |
| 0x88 | MLB_HCMR0 | 7:0 | CHM: Bitwise Channel Mask Bit [31[7:0] | | | | | | | |
| | | 15:8 | CHM: Bitwise Channel Mask Bit [31[15:8] | | | | | | | |
| | | 23:16 | CHM: Bitwise Channel Mask Bit [31[23:16] | | | | | | | |
| | | 31:24 | CHM: Bitwise Channel Mask Bit [31[31:24] | | | | | | | |
| 0x8C | MLB_HCMR1 | 7:0 | CHM: Bitwise Channel Mask Bit [63[7:0] | | | | | | | |
| | | 15:8 | CHM: Bitwise Channel Mask Bit [63[15:8] | | | | | | | |
| | | 23:16 | CHM: Bitwise Channel Mask Bit [63[23:16] | | | | | | | |
| | | 31:24 | CHM: Bitwise Channel Mask Bit [63[31:24] | | | | | | | |
| 0x90 | MLB_HCER0 | 7:0 | CERR: Bitwise Channel Error Bit [31[7:0] | | | | | | | |
| | | 15:8 | CERR: Bitwise Channel Error Bit [31[15:8] | | | | | | | |
| | | 23:16 | CERR: Bitwise Channel Error Bit [31[23:16] | | | | | | | |
| | | 31:24 | CERR: Bitwise Channel Error Bit [31[31:24] | | | | | | | |
| 0x94 | MLB_HCER1 | 7:0 | CERR: Bitwise Channel Error Bit [63[7:0] | | | | | | | |
| | | 15:8 | CERR: Bitwise Channel Error Bit [63[15:8] | | | | | | | |
| | | 23:16 | CERR: Bitwise Channel Error Bit [63[23:16] | | | | | | | |
| | | 31:24 | CERR: Bitwise Channel Error Bit [63[31:24] | | | | | | | |
| 0x98 | MLB_HCBR0 | 7:0 | CHB: Bitwise Channel Busy Bit [31[7:0] | | | | | | | |
| | | 15:8 | CHB: Bitwise Channel Busy Bit [31[15:8] | | | | | | | |
| | | 23:16 | CHB: Bitwise Channel Busy Bit [31[23:16] | | | | | | | |
| | | 31:24 | CHB: Bitwise Channel Busy Bit [31[31:24] | | | | | | | |
| 0x9C | MLB_HCBR1 | 7:0 | CHB: Bitwise Channel Busy Bit [63[7:0] | | | | | | | |
| | | 15:8 | CHB: Bitwise Channel Busy Bit [63[15:8] | | | | | | | |
| | | 23:16 | CHB: Bitwise Channel Busy Bit [63[23:16] | | | | | | | |
| | | 31:24 | CHB: Bitwise Channel Busy Bit [63[31:24] | | | | | | | |
| 0xA0<br>...<br>0xBF | Reserved | | | | | | | | | |
| 0xC0 | MLB_MDAT0 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0xC4 | MLB_MDAT1 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0xC8 | MLB_MDAT2 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |

### 48.7.16 MIF Data 1 Register

**Name:** MLB_MDAT1
**Offset:** 0x0C4
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

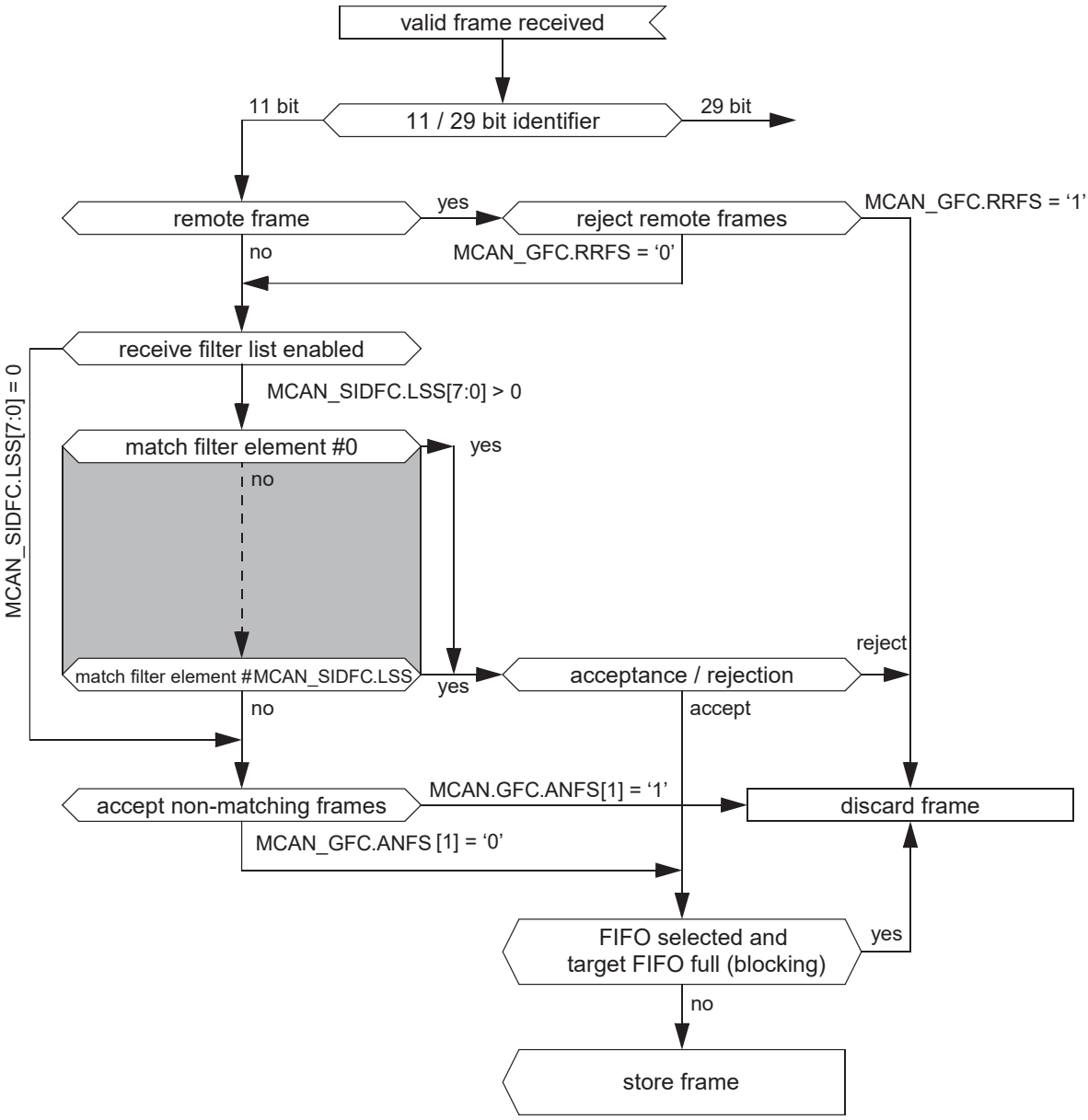
| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DATA[31:0]** CRT Data
CTR data - bits[63:32] of 128-bit entry

**Figure 49-5. Standard Message ID Filter Path**




***Extended Message ID Filtering***

The figure below shows the flow for extended Message ID (29-bit Identifier) filtering. The Extended Message ID Filter element is described in 49.5.7.6 Extended Message ID Filter Element.

Controlled by MCAN_GFC and MCAN_XIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

MCAN_XIDAM is ANDed with the received identifier before the filter list is executed.

Buffer element is described in Tx Buffer Element. The table below describes the possible configurations for frame transmission.

**Table 49-5. Possible Configurations for Frame Transmission**

| MCAN_CCCR | | Tx Buffer Element | | Frame Transmission |
|---|---|---|---|---|
| BRSE | FDOE | FDF | BRS | |
| ignored | 0 | ignored | ignored | Classic CAN |
| 0 | 1 | 0 | ignored | Classic CAN |
| 0 | 1 | 1 | ignored | FD without bit rate switching |
| 1 | 1 | 0 | ignored | Classic CAN |
| 1 | 1 | 1 | 0 | FD without bit rate switching |
| 1 | 1 | 1 | 1 | FD with bit rate switching |

**Note:** AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation.

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when MCAN_TXBRP is updated, or when a transmission has been started.

#### 49.5.5.1 Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two CAN bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by bit MCAN_CCCR.TXP. If the bit is set, the MCAN will, each time it has successfully transmitted a message, pause for two CAN bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (MCAN_CCCR.TXP = '0').

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

#### 49.5.5.2 Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the processor. Each dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via MCAN_TXBAR.ARn. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

### 50.7.3 TC Channel Mode Register: Waveform Mode

**Name:** TC_CMRx
**Offset:** 0x04 + x*0x40 [x=0..2]
**Reset:** 0x00000000
**Property:** Read/Write

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

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | BSWTRG[1:0] | | BEEVT[1:0] | | BCPC[1:0] | | BCPB[1:0] | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ASWTRG[1:0] | | AEEVT[1:0] | | ACPC[1:0] | | ACPA[1:0] | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | WAVE | WAVSEL[1:0] | | ENETRG | EEVT[1:0] | | EEVTEDG[1:0] | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CPCDIS | CPCSTOP | BURST[1:0] | | CLKI | TCCLKS[2:0] | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:30 – BSWTRG[1:0]** Software Trigger Effect on TIOBx

| Value | Name | Description |
|---|---|---|
| 0 | NONE | None |
| 1 | SET | Set |
| 2 | CLEAR | Clear |
| 3 | TOGGLE | Toggle |

**Bits 29:28 – BEEVT[1:0]** External Event Effect on TIOBx

| Value | Name | Description |
|---|---|---|
| 0 | NONE | None |
| 1 | SET | Set |
| 2 | CLEAR | Clear |
| 3 | TOGGLE | Toggle |

**Bits 27:26 – BCPC[1:0]** RC Compare Effect on TIOBx

| Value | Name | Description |
|---|---|---|
| 0 | NONE | None |
| 1 | SET | Set |

### 51.7.10 PWM DMA Register

**Name:** PWM_DMAR
**Offset:** 0x24
**Reset:** –
**Property:** Write-only

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

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DMADUTY[23:16] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DMADUTY[15:8] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DMADUTY[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | – |

**Bits 23:0 – DMADUTY[23:0]** Duty-Cycle Holding Register for DMA Access
Each write access to PWM_DMAR sequentially updates PWM_CDTYUPDx.CDTYUPD with DMADUTY (only for channel configured as synchronous). See “Method 3: Automatic write of duty-cycle values and automatic trigger of the update” .

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 31:24 | | | | | | | | |
| 0x74 | AFEC_TEMPCWR | 7:0 | TLOWTHRES[7:0] | | | | | | | |
| | | 15:8 | TLOWTHRES[15:8] | | | | | | | |
| | | 23:16 | THIGHTHRES[7:0] | | | | | | | |
| | | 31:24 | THIGHTHRES[15:8] | | | | | | | |
| 0x78 ... 0x93 | Reserved | | | | | | | | | |
| 0x94 | AFEC_ACR | 7:0 | | | | | PGA1EN | PGA0EN | | |
| | | 15:8 | | | | | | | IBCTL[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x98 ... 0x9F | Reserved | | | | | | | | | |
| 0xA0 | AFEC_SHMR | 7:0 | DUAL7 | DUAL6 | DUAL5 | DUAL4 | DUAL3 | DUAL2 | DUAL1 | DUAL0 |
| | | 15:8 | | | | | DUAL11 | DUAL10 | DUAL9 | DUAL8 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xA4 ... 0xCF | Reserved | | | | | | | | | |
| 0xD0 | AFEC_COSR | 7:0 | | | | | | | | CSEL |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xD4 | AFEC_CVR | 7:0 | OFFSETCORR[7:0] | | | | | | | |
| | | 15:8 | OFFSETCORR[15:8] | | | | | | | |
| | | 23:16 | GAINCORR[7:0] | | | | | | | |
| | | 31:24 | GAINCORR[15:8] | | | | | | | |
| 0xD8 | AFEC_CECR | 7:0 | ECORR7 | ECORR6 | ECORR5 | ECORR4 | ECORR3 | ECORR2 | ECORR1 | ECORR0 |
| | | 15:8 | | | | | ECORR11 | ECORR10 | ECORR9 | ECORR8 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xDC ... 0xE3 | Reserved | | | | | | | | | |
| 0xE4 | AFEC_WPMR | 7:0 | | | | | | | | WPEN |
| | | 15:8 | WPKEY[7:0] | | | | | | | |
| | | 23:16 | WPKEY[15:8] | | | | | | | |
| | | 31:24 | WPKEY[23:16] | | | | | | | |
| 0xE8 | AFEC_WPSR | 7:0 | | | | | | | | WPVS |
| | | 15:8 | WPVSRC[7:0] | | | | | | | |
| | | 23:16 | WPVSRC[15:8] | | | | | | | |
| | | 31:24 | | | | | | | | |

| Core Clock/MCK (MHz) | VDDCORE Consumption (AMP1) | Total Consumption (AMP2) | Unit | Wakeup Time | Unit |
|---|---|---|---|---|---|
| 96/96 | 12.5 | 15 | | 1.4 | |
| 96/48 | 7.5 | 10 | | 2.5 | |
| 48/48 | 7 | 9.5 | | 2.8 | |
| 24/24 | 3.5 | 5 | | 5.6 | |
| 24/12 | 2 | 3 | | 10 | |
| 12/12 | 2 | 3 | | 11.2 | |
| 8/8 | 1.5 | 2 | | 16.8 | |
| 4/4 | 1.0 | 1.5 | | 32.9 | |
| 4/2 | 0.9 | 1 | | 60 | |
| 4/1 | 0.8 | 1 | | 112.6 | |

**Table 59-14. Typical Sleep Mode Current Consumption vs. Master Clock (MCK) Variation with Fast RC**

| Core Clock/MCK (MHz) | VDDCORE Consumption (AMP1) | Total Consumption (AMP2) | Unit | Wakeup Time | Unit |
|---|---|---|---|---|---|
| 12 | 2.0 | 2.0 | | 12 | |
| 8 | 1.5 | 1.5 | | 18 | |
| 4 | 1.0 | 1.1 | | 31 | |
| 2 | 0.8 | 0.8 | mA | 62 | µs |
| 1 | 0.6 | 0.7 | | 123 | |
| 0.5 | 0.6 | 0.6 | | 247 | |
| 0.25 | 0.5 | 0.5 | | 494 | |

### 59.3.3 Wait Mode Current Consumption and Wakeup Time

The Wait mode configuration and measurements are defined as follows:

- Core clock and Master clock stopped
- Current measurement as shown below
- All peripheral clocks deactivated
- BOD disabled
- RTT enabled