**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, Ethernet, I²C, IrDA, LINbus, SPI, SSC, UART/USART |
| Peripherals | Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT |
| Number of I/O | 44 |
| Program Memory Size | 1MB (1M x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 384K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.7V ~ 3.6V |
| Data Converters | A/D 5x12b; D/A 1x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°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/atsame70j20b-ant |

| Step | Handshake Sequence | MODE[3:0] | DATA[15:0] |
|---|---|---|---|
| n | Write handshaking | ADDR0 | Memory Address LSB |
| n+1 | Write handshaking | ADDR1 | Memory Address |
| n+2 | Write handshaking | DATA | *Memory Address++ |
| n+3 | Write handshaking | DATA | *Memory Address++ |
| ... | ... | ... | ... |

#### 18.3.5.8 Get Version Command

The Get Version (GVE) command retrieves the version of the FFPI interface.

**Table 18-15. Get Version Command**

| Step | Handshake Sequence | MODE[3:0] | DATA[15:0] |
|---|---|---|---|
| 1 | Write handshaking | CMDE | GVE |
| 2 | Read handshaking | DATA | Version |

### 22.4.2 Read Operations

An optimized controller manages embedded Flash reads, thus increasing performance when the processor is running in Thumb-2 mode by means of the 128-bit-wide memory interface.

The Flash memory is accessible through 8-, 16- and 32-bit reads.

As the Flash block size is smaller than the address space reserved for the internal memory area, the embedded Flash wraps around the address space and appears to be repeated within it.

The read operations can be performed with or without wait states. Wait states must be programmed in the field FWS in the Flash Mode register (EEFC_FMR). Defining FWS as 0 enables the single-cycle access of the embedded Flash. For more details, refer to the section “Electrical Characteristics” of this datasheet.

**Related Links**

59. Electrical Characteristics for SAM E70/S70
58. Electrical Characteristics for SAM V70/V71

#### 22.4.2.1 Code Read Optimization

Code read optimization is enabled if the bit EEFC_FMR.SCOD is cleared.

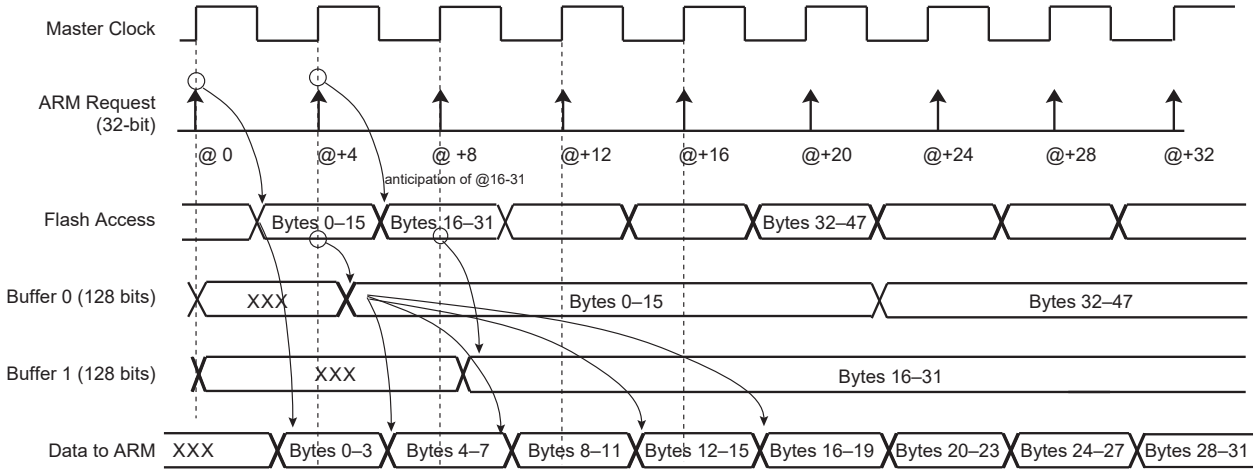
A system of 2 x 128-bit buffers is added in order to optimize sequential code fetch.

**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 EEFC_FMR.SCOD is set, these buffers are disabled and the sequential code read is no longer optimized.

Another system of 2 x 128-bit buffers is added in order to optimize loop code fetch. Refer to the “Code Loop Optimization” section for more details.

**Figure 22-3. Code Read Optimization for FWS = 0**




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

**Table 22-2. Flash Descriptor Definition**

| Symbol | Word Index | Description |
|---|---|---|
| FL_ID | 0 | Flash interface description |
| FL_SIZE | 1 | Flash size in bytes |
| FL_PAGE_SIZE | 2 | Page size in bytes |
| FL_NB_PLANE | 3 | Number of planes |
| FL_PLANE[0] | 4 | Number of bytes in the plane |
| FL_NB_LOCK | 4 + FL_NB_PLANE | Number of lock bits. A bit is associated with a lock region. A lock bit is used to prevent write or erase operations in the lock region. |
| FL_LOCK[0] | 4 + FL_NB_PLANE + 1 | Number of bytes in the first lock region |

#### 22.4.3.2 Write Commands

DMA write accesses must be 32-bit aligned. If a single byte is to be written in a 32-bit word, the rest of the word must be written with ones.

Several commands are used to program the Flash.

Only 0 values can be programmed using Flash technology; 1 is the erased value. In order to program words in a page, the page must first be erased. Commands are available to erase the full memory plane or a given number of pages. With the EWP and EWPL commands, a page erase is done automatically before a page programming.

After programming, the page (the entire lock region) can be locked to prevent miscellaneous write or erase sequences. The lock bit can be automatically set after page programming using WPL or EWPL commands.

Data to be programmed in the Flash must be written in an internal latch buffer before writing the programming command in EEFC_FCR. Data can be written at their final destination address, as the latch buffer is mapped into the Flash memory address space and wraps around within this Flash address space.

Byte and half-word AHB accesses to the latch buffer are not allowed. Only 32-bit word accesses are supported.

32-bit words must be written continuously, in either ascending or descending order. Writing the latch buffer in a random order is not permitted. This prevents mapping a C-code structure to the latch buffer and accessing the data of the structure in any order. It is instead recommended to fill in a C-code structure in SRAM and copy it in the latch buffer in a continuous order.

Write operations in the latch buffer are performed with the number of wait states programmed for reading the Flash.

The latch buffer is automatically re-initialized, i.e., written with logical ‘1’, after execution of each programming command.

The programming sequence is the following:

1. Write the data to be programmed in the latch buffer.
2. Write the programming command in EEFC_FCR. This automatically clears the bit EEFC_FSR.FRDY.

### 26.4.5.1 RSTC Control Register

**Name:** RSTC_CR
**Offset:** 0x00
**Property:** Write-only

| 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 | – |

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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | EXTRST | | | PROCRST |
| Access | | | | | W | | | W |
| Reset | | | | | – | | | – |

**Bits 31:24 – KEY[7:0]** System Reset Key

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

**Bit 3 – EXTRST** External Reset

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | If KEY = 0xA5, asserts the NRST pin. |

**Bit 0 – PROCRST** Processor Reset

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | If KEY = 0xA5, resets the processor and all the embedded peripherals. |

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x06E0 | GMAC_ST2ER0 | 7:0 | COMPVAL[7:0] | | | | | | | |
| | | 15:8 | COMPVAL[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x06E4 | GMAC_ST2ER1 | 7:0 | COMPVAL[7:0] | | | | | | | |
| | | 15:8 | COMPVAL[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x06E8 | GMAC_ST2ER2 | 7:0 | COMPVAL[7:0] | | | | | | | |
| | | 15:8 | COMPVAL[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x06EC | GMAC_ST2ER3 | 7:0 | COMPVAL[7:0] | | | | | | | |
| | | 15:8 | COMPVAL[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x06F0 ... 0x06FF | Reserved | | | | | | | | | |
| 0x0700 | GMAC_ST2CW00 | 7:0 | MASKVAL[7:0] | | | | | | | |
| | | 15:8 | MASKVAL[15:8] | | | | | | | |
| | | 23:16 | COMPVAL[7:0] | | | | | | | |
| | | 31:24 | COMPVAL[15:8] | | | | | | | |
| 0x0704 | GMAC_ST2CW10 | 7:0 | OFFSSTRT[0:0] | OFFSVAL[6:0] | | | | | | |
| | | 15:8 | | | | | | | | OFFSSTRT[1:1] |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0708 | GMAC_ST2CW01 | 7:0 | MASKVAL[7:0] | | | | | | | |
| | | 15:8 | MASKVAL[15:8] | | | | | | | |
| | | 23:16 | COMPVAL[7:0] | | | | | | | |
| | | 31:24 | COMPVAL[15:8] | | | | | | | |
| 0x070C | GMAC_ST2CW11 | 7:0 | OFFSSTRT[0:0] | OFFSVAL[6:0] | | | | | | |
| | | 15:8 | | | | | | | | OFFSSTRT[1:1] |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0710 | GMAC_ST2CW02 | 7:0 | MASKVAL[7:0] | | | | | | | |
| | | 15:8 | MASKVAL[15:8] | | | | | | | |
| | | 23:16 | COMPVAL[7:0] | | | | | | | |
| | | 31:24 | COMPVAL[15:8] | | | | | | | |
| 0x0714 | GMAC_ST2CW12 | 7:0 | OFFSSTRT[0:0] | OFFSVAL[6:0] | | | | | | |

### 38.8.24 GMAC Type ID Match n Register

**Name:** GMAC_TIDMx
**Offset:** 0xA8 + x*0x04 [x=0..3]
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ENIDn | | | | | | | |
| Access | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TID[15:8] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | TID[7:0] | | | | | | | |
| 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 31 – ENIDn** Enable Copying of TID Matched Frames

| Value | Description |
|---|---|
| 0 | TID n is not part of the comparison match. |
| 1 | TID n is processed for the comparison match. |

**Bits 15:0 – TID[15:0]** Type ID Match n
For use in comparisons with received frames type ID/length frames.

| Value | Description |
|---|---|
| 0 | Cleared when the HBISOFLUSHIC bit is written to one. This acknowledges the interrupt. |
| 1 | Set for High-bandwidth isochronous IN endpoint (with NBTRANS = 2 or 3) at the end of the microframe, if less than N transactions have been completed by the USBHS without underflow error. This may occur in case of a missing IN token. In this case, the banks are flushed out to ensure the data synchronization between the host and the device. This triggers a PEP_x interrupt if HBISOFLUSHE = 1. |

**Bit 3 – HBISOINERRI** High Bandwidth Isochronous IN Underflow Error Interrupt

| Value | Description |
|---|---|
| 0 | Cleared when the HBISOINERRIC bit is written to one. This acknowledges the interrupt. |
| 1 | Set for High-bandwidth isochronous IN endpoint (with NBTRANS = 2 or 3) at the end of the microframe, if less than N banks were written by the CPU within this microframe. This triggers a PEP_x interrupt if HBISOINERRE = 1. |

**Bit 2 – UNDERFI** Underflow Interrupt
This bit is set, for isochronous IN/OUT endpoints, when an underflow error occurs. This triggers a PEP_x interrupt if UNDERFE = 1.

An underflow can occur during IN stage if the host attempts to read from an empty bank. A zero-length packet is then automatically sent by the USBHS.

An underflow can also occur during OUT stage if the host sends a packet while the bank is already full. Typically, the CPU is not fast enough. The packet is lost.

It is cleared by writing a one to the UNDERFIC bit. This acknowledges the interrupt.

**Bit 1 – RXOUTI** Received OUT Data Interrupt
For control endpoints:

0: Cleared by writing a one to the RXOUTIC bit. This acknowledges the interrupt and frees the bank.

1: Set when the current bank contains a bulk OUT packet (data or status stage). This triggers a PEP_x interrupt if USBHS_DEVEPTIMRx.RXOUTE = 1.

For OUT endpoints:

0: Cleared by writing a one to the RXOUTIC bit. This acknowledges the interrupt, which has no effect on the endpoint FIFO. USBHS_DEVEPTISRx.RXOUTI shall always be cleared before clearing USBHS_DEVEPTIMRx.FIFOCON.

1: Set at the same time as USBHS_DEVEPTIMRx.FIFOCON when the current bank is full. This triggers a PEP_x interrupt if USBHS_DEVEPTIMRx.RXOUTE = 1.

The user reads from the FIFO and clears the USBHS_DEVEPTIMRx.FIFOCON bit to free the bank. If the OUT endpoint is composed of multiple banks, this also switches to the next bank. The USBHS_DEVEPTISRx.RXOUTI and USBHS_DEVEPTIMRx.FIFOCON bits are set/cleared in accordance with the status of the next bank.

This bit is inactive (cleared) for IN endpoints.

**Bit 0 – TXINI** Transmitted IN Data Interrupt
For control endpoints:

0: Cleared when TXINIC = 1. This acknowledges the interrupt and sends the packet.

### 39.6.22 Device Endpoint Interrupt Mask Register (Isochronous Endpoints)

**Name:** USBHS_DEVEPTIMRx (ISOENPT)
**Offset:** 0x01C0 + x*0x04 [x=0..9]
**Reset:** 0
**Property:** Read/Write

This register view is relevant only if EPTYPE = 0x1 in "Device Endpoint x Configuration Register".

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | RSTDT | | EPDISHDMA |
| Access | | | | | | | | |
| Reset | | | | | | 0 | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | FIFOCON | KILLBK | NBUSYBKE | | ERRORTRANSE | DATAXE | MDATAE |
| Access | | | | | | | | |
| Reset | | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SHORTPACKETE | CRCERRE | OVERFE | HBISOFLUSHE | HBISOINERRE | UNDERFE | RXOUTE | TXINE |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 18 – RSTDT** Reset Data Toggle
This bit is set when USBHS_DEVEPTIERx.RSTDTS = 1. This clears the data toggle sequence, i.e., sets to Data0 the data toggle sequence of the next sent (IN endpoints) or received (OUT endpoints) packet.

This bit is cleared instantaneously.

The user does not have to wait for this bit to be cleared.

**Bit 16 – EPDISHDMA** Endpoint Interrupts Disable HDMA Request
This bit is set when USBHS_DEVEPTIERx.EPDISHDMAS = 1. This pauses the on-going DMA channel x transfer on any Endpoint x interrupt (PEP_x), whatever the state of the Endpoint x Interrupt Enable bit (PEP_x).

The user then has to acknowledge or to disable the interrupt source (e.g. USBHS_DEVEPTISRx.RXOUTI) or to clear the EPDISHDMA bit (by writing a one to the USBHS_DEVEPTIDRx.EPDISHDMAC bit) in order to complete the DMA transfer.

In Ping-pong mode, if the interrupt is associated to a new system-bank packet (e.g. Bank1) and the current DMA transfer is running on the previous packet (Bank0), then the previous-packet DMA transfer completes normally, but the new-packet DMA transfer does not start (not requested).

| Value | Description |
|---|---|
| 0 | Cleared when USBHS_HSTPIPIDR.OVERFIEC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.OVERFIE). |
| 1 | Set when USBHS_HSTPIPIER.OVERFIES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.OVERFIE). |

**Bit 4 – NAKEDE** NAKed Interrupt Enable

| Value | Description |
|---|---|
| 0 | Cleared when USBHS_HSTPIPIDR.NAKEDEC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.NAKEDE). |
| 1 | Set when USBHS_HSTPIPIER.NAKEDES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.NAKEDE). |

**Bit 3 – PERRE** Pipe Error Interrupt Enable

| Value | Description |
|---|---|
| 0 | Cleared when USBHS_HSTPIPIDR.PERREC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.PERRE). |
| 1 | Set when USBHS_HSTPIPIER.PERRES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.PERRE). |

**Bit 2 – UNDERFIE** Underflow Interrupt Enable

| Value | Description |
|---|---|
| 0 | Cleared when USBHS_HSTPIPIDR.UNDERFIEC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.UNDERFIE). |
| 1 | Set when USBHS_HSTPIPIER.UNDERFIES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.UNDERFIE). |

**Bit 1 – TXOUTE** Transmitted OUT Data Interrupt Enable

| Value | Description |
|---|---|
| 0 | Cleared when USBHS_HSTPIPIDR.TXOUTEC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.TXOUTE). |
| 1 | Set when USBHS_HSTPIPIER.TXOUTES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.TXOUTE). |

**Bit 0 – RXINE** Received IN Data Interrupt Enable

| Value | Description |
|---|---|
| 0 | Cleared when USBHS_HSTPIPIDR.RXINEC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.RXINE). |
| 1 | Set when USBHS_HSTPIPIER.RXINES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.RXINE). |

### 42.7.8 QSPI Interrupt Mask Register

**Name:** QSPI_IMR
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** Read-only

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

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

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

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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | INSTRE | CSS | CSR |
| Access | | | | | | R | R | R |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | OVRES | TXEMPTY | TDRE | RDRF |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 10 – INSTRE** Instruction End Interrupt Mask

**Bit 9 – CSS** Chip Select Status Interrupt Mask

**Bit 8 – CSR** Chip Select Rise Interrupt Mask

**Bit 3 – OVRES** Overrun Error Interrupt Mask

**Bit 2 – TXEMPTY** Transmission Registers Empty Mask

**Bit 1 – TDRE** Transmit Data Register Empty Interrupt Mask

**Bit 0 – RDRF** Receive Data Register Full Interrupt Mask

### 45.8.7 I2SC Interrupt Disable Register

**Name:** I2SC_IDR
**Offset:** 0x18
**Property:** Write-only

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

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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | TXUR | TXRDY | | | RXOR | RXRDY | |
| Access | | W | W | | | W | W | |
| Reset | | – | – | | | – | – | |

**Bit 6 – TXUR** Transmit Underflow Interrupt Disable

| Value | Description |
|---|---|
| 0 | Writing a ’0’ to this bit has no effect. |
| 1 | Writing a ’1’ to this bit clears the corresponding bit in I2SC_IMR. |

**Bit 5 – TXRDY** Transmit Ready Interrupt Disable

| Value | Description |
|---|---|
| 0 | Writing a ’0’ to this bit has no effect. |
| 1 | Writing a ’1’ to this bit clears the corresponding bit in I2SC_IMR. |

**Bit 2 – RXOR** Receiver Overrun Interrupt Disable

| Value | Description |
|---|---|
| 0 | Writing a ’0’ to this bit has no effect. |
| 1 | Writing a ’1’ to this bit clears the corresponding bit in I2SC_IMR. |

**Bit 1 – RXRDY** Receiver Ready Interrupt Disable

| Value | Description |
|---|---|
| 0 | Writing a ’0’ to this bit has no effect. |
| 1 | Writing a ’1’ to this bit clears the corresponding bit in I2SC_IMR. |

### 46.7.19 USART Channel Status Register (LIN_MODE)

**Name:** US_CSR (LIN_MODE)
**Offset:** 0x0014
**Reset:** 0x0
**Property:** Read-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | LINHTE | LINSTE | LINSNRE | LINCE | LINIPE | LINISFE | LINBE | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | LINBLS | | | | | | | |
| Access | | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | LINTC | LINID | LINBK | | | | TXEMPTY | TIMEOUT |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PARE | FRAME | OVRE | | | | TXRDY | RXRDY |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | | | | 0 | 0 |

**Bit 31 – LINHTE** LIN Header Timeout Error (cleared by writing a one to bit US_CR.RSTSTA)

| Value | Description |
|---|---|
| 0 | No LIN header timeout error has been detected since the last RSTSTA. |
| 1 | A LIN header timeout error has been detected since the last RSTSTA. |

**Bit 30 – LINSTE** LIN Synch Tolerance Error (cleared by writing a one to bit US_CR.RSTSTA)

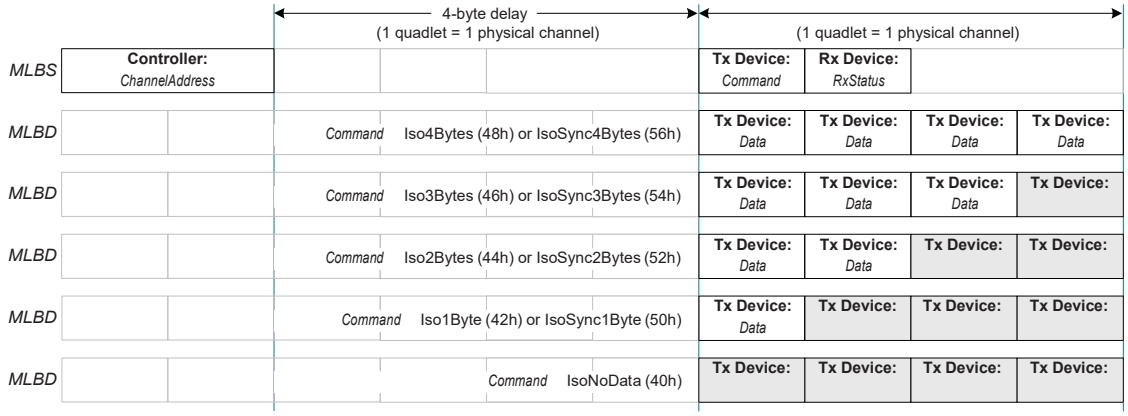
| Value | Description |
|---|---|
| 0 | No LIN synch tolerance error has been detected since the last RSTSTA. |
| 1 | A LIN synch tolerance error has been detected since the last RSTSTA. |

**Bit 29 – LINSNRE** LIN Slave Not Responding Error (cleared by writing a one to bit US_CR.RSTSTA)

| Value | Description |
|---|---|
| 0 | No LIN slave not responding error has been detected since the last RSTSTA. |
| 1 | A LIN slave not responding error has been detected since the last RSTSTA. |

**Bit 28 – LINCE** LIN Checksum Error (cleared by writing a one to bit US_CR.RSTSTA)

**Figure 48-13. MediaLB Isochronous Data Structure**




The isochronous flow for a Tx Device is illustrated in Figure 48-14. The data transfer blocks (slanted rectangle shapes) occur only during a physical channel (PCn) associated with the logical channel defined by a single ChannelAddress. Similar to the synchronous flow, isochronous data immediately starts transmitting. When data exists from the application, the IsoSync?Bytes commands are used to indicate the start of a block, which provides alignment information to the Rx Device. The Iso?Bytes commands indicate the middle of a block of data. The definition of block for isochronous data is outside the scope of this document. For physical channels that transfer less than four bytes, the Rx Device must only use/ store the number of valid bytes, and ignore the unused portion.

The isochronous flow for an Rx Device is illustrated in Figure 48-15. The NoData command indicates that the channel is not setup yet. Once an isochronous channel is setup, the Rx Device continually receives the channel data, similar to synchronous data. The only two valid responses for an isochronous channel are ReceiverBusy, and the default bus state of ReceiverReady. Although Rx Devices can respond with ReceiverBusy, its use should be minimized, since Tx Devices may not be able to store much isochronous data that gets backed up due to the ReceiverBusy responses. If any Rx Device uses ReceiverBusy, then only one Rx Device is allowed. If all targeted Rx Devices do not drive RxStatus (default ReceiverReady response), then the isochronous stream can support multiple Rx Devices (broadcast).

### 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

### 48.7.29 AHB Channel Mask 1 Register

**Name:** MLB_ACMR1
**Offset:** 0x3DC
**Reset:** 0x00000000
**Property:** Read/Write

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

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

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

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

**Bits 31:0 – CHM[31:0]** Bitwise Channel Mask Bits 63 to 32
CHM[n] = 1 indicates that channel n can generate an interrupt.

**Bit 0 – RF0N** Receive FIFO 0 New Message

| Value | Description |
|---|---|
| 0 | No new message written to Receive FIFO 0. |
| 1 | New message written to Receive FIFO 0. |

### 51.7.3 PWM Disable Register

**Name:** PWM_DIS
**Offset:** 0x08
**Reset:** –
**Property:** Write-only

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

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

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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CHID3 | CHID2 | CHID1 | CHID0 |
| Access | | | | | W | W | W | W |
| Reset | | | | | 0 | 0 | 0 | – |

**Bits 0, 1, 2, 3 – CHIDx** Channel ID

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | Disable PWM output for channel x. |

## 54.4 Signal Description

**Table 54-1. ACC Signal Description**

| Pin Name | Description | Type |
|---|---|---|
| AFE0_AD[5:0] | External analog data inputs | Input |
| AFE1_AD[1:0] | | |
| TS | On-chip temperature sensor | Input |
| VREFP | AFE and DAC voltage reference | Input |
| DAC0, DAC1 | On-chip DAC outputs | Input |

## 54.5 Product Dependencies

### 54.5.1 I/O Lines

The analog input pins are multiplexed with digital functions (PIO) on the IO line. By writing the SELMINUS and SELPLUS fields in the ACC Mode Register (ACC_MR), the associated IO lines are set to Analog mode.

### 54.5.2 Power Management

The ACC is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the ACC clock.

Note that the voltage regulator must be activated to use the analog comparator.

### 54.5.3 Interrupt Sources

The ACC has an interrupt line connected to the Interrupt Controller (IC). In order to handle interrupts, the Interrupt Controller must be programmed before configuring the ACC.

### 54.5.4 Fault Output

The ACC has the FAULT output connected to the FAULT input of PWM. See Fault Mode and the implementation of the PWM in the product.

## 54.6 Functional Description

### 54.6.1 Description

The Analog Comparator Controller (ACC) controls the analog comparator settings and performs postprocessing of the analog comparator output.

When the analog comparator settings are modified, the output of the analog cell may be invalid. The ACC masks the output for the invalid period.

A comparison flag is triggered by an event on the output of the analog comparator and an interrupt is generated. The event on the analog comparator output can be selected among falling edge, rising edge or any edge.

The ACC registers are listed in the Register Summary.

### 57.5.6 AES Interrupt Status Register

**Name:** AES_ISR
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** Read-only

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | R | R | R | R | | | | R |
| Reset | 0 | 0 | 0 | 0 | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

**Bit 16 – TAGRDY** GCM Tag Ready

| Value | Description |
|---|---|
| 0 | GCM Tag is not valid. |
| 1 | GCM Tag generation is complete (cleared by reading GCM Tag, starting another processing or when writing a new key). |

**Bits 15:12 – URAT[3:0]** Unspecified Register Access (cleared by writing SWRST in AES_CR)
Only the last Unspecified Register Access Type is available through the URAT field.

| Value | Name | Description |
|---|---|---|
| 0 | IDR_WR_PROCESSING | Input Data register written during the data processing when SMOD = 2 mode. |
| 1 | ODR_RD_PROCESSING | Output Data register read during the data processing. |
| 2 | MR_WR_PROCESSING | Mode register written during the data processing. |
| 3 | ODR_RD_SUBKGEN | Output Data register read during the sub-keys generation. |
| 4 | MR_WR_SUBKGEN | Mode register written during the sub-keys generation. |
| 5 | WOR_RD_ACCESS | Write-only register read access. |

**Bit 8 – URAD** Unspecified Register Access Detection Status (cleared by writing SWRST in AES_CR)

| Signal Name | Recommended Pin Connection | Description |
|---|---|---|
| | | Decoupling/filtering capacitors must be added to improve startup stability and reduce source voltage drop.<br>Supply ripple must not exceed 20 mVrms for 10 kHz to 20 MHz range.<br><br>**WARNING** Powerup and powerdown sequences given in the "Power Considerations" chapter must be respected. |
| VDDPLL | Decoupling/filtering capacitors ferrite beads (100 nF and 470 Ohm @ 100 MHz) (1) (2) | Powers the PLLA and the fast RC oscillator.<br>The VDDPLL power supply pin draws small current, but it is noise sensitive. Care must be taken in VDDPLL power supply routing, decoupling and also on bypass capacitors.<br>Supply ripple must not exceed 20 mVrms for 10 kHz to 10 MHz range and 10 mVrms for higher frequencies. |
| VDDUTMIC | Decoupling/filtering capacitors ferrite beads (100 nF and 470 Ohm @ 100 MHz) (1) (2) | Powers the USB transceiver core.<br>Must always be connected even if the USB is not used.<br>Decoupling/filtering capacitors/ferrite beads must be added to improve startup stability and reduce source voltage drop.<br>Supply ripple must not exceed 10 mVrms for 10 kHz to 10 MHz range. |
| GND | Voltage Regulator, Core Chip and Peripheral I/O lines ground | GND pins are common to VDDIN, VDDCORE and VDDIO pins.<br>GND pins should be connected as shortly as possible to the system ground plane. |
| GNDUTMI | UDPHS and UHPHS UTMI+ Core and interface ground | GNDUTMI pins are common to VDDUTMII and VDDUTMIC pins.<br>GNDUTMI pins should be connected as shortly as possible to the system ground plane. |
| GNDPLL | PLLA cell and Main Oscillator ground | GNDPLL pin is provided for VDDPLL pin. GNDPLL pin should be connected as shortly as possible to the system ground plane. |
| GNDANA | Analog ground | GNDANA pins are common to AFE, DAC and ACC supplied by VDDIN pin.<br>GNDANA pins should be connected as shortly as possible to the system ground plane. |
| GNDPLLUSB | USB PLL ground | GNDPLLUSB pin is provided for VDDPLLUSB pin.<br>GNDPLLUSB pin should be connected as shortly as possible to the system ground plane. |