**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, MMC/SD/SDIO, QSPI, SPI, UART/USART, USB |
| Peripherals | Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT |
| Number of I/O | 75 |
| Program Memory Size | 512KB (512K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 256K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.62V ~ 3.6V |
| Data Converters | A/D 10x12b; D/A 2x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 100-LQFP |
| Supplier Device Package | 100-LQFP (14x14) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/atsame70n19a-an |

# 16. Debug and Test Features

## 16.1 Description

The device features a number of complementary debug and test capabilities. The Serial Wire Debug Port (SW-DP) is used for standard debugging functions, such as downloading code and single-stepping through programs. It also embeds a serial wire trace.

## 16.2 Embedded Characteristics

- Debug access to all memory and registers in the system, including Cortex-M register bank, when the core is running, halted, or held in reset.
- Serial Wire Debug Port (SW-DP) debug access
- Flash Patch and Breakpoint (FPB) unit for implementing breakpoints and code patches
- Data Watchpoint and Trace (DWT) unit for implementing watchpoints, data tracing, and system profiling
- Instrumentation Trace Macrocell (ITM) for support of printf style debugging
- 6-pin Embedded Trace Macrocell (ETM) for instruction trace stream, including CoreSight™ Trace Port Interface Unit (TPIU)
- IEEE1149.1 JTAG Boundary scan on All Digital Pins

## 16.3 Associated Documents

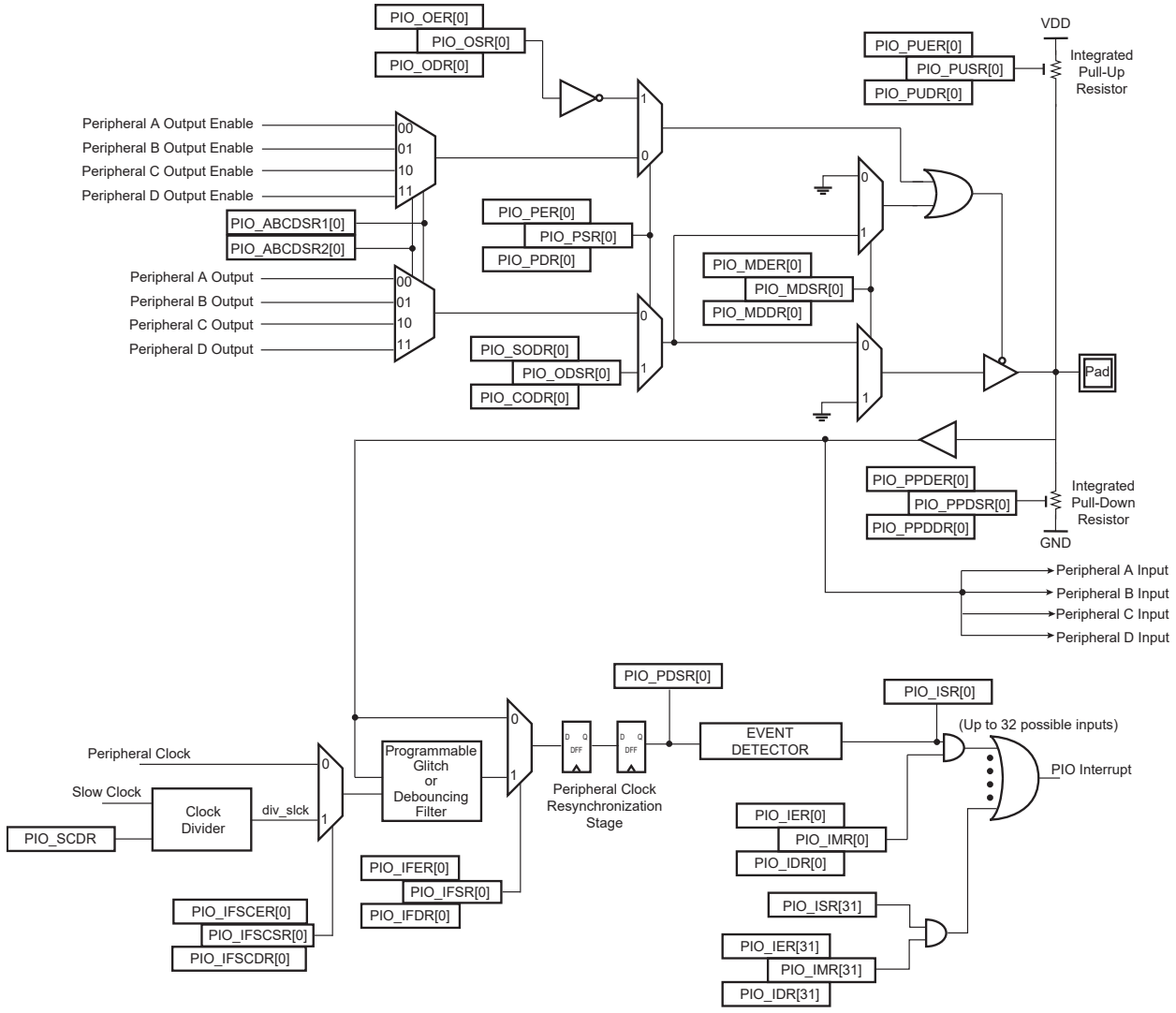
The SAM E70/S70/V70/V71 implements the standard ARM CoreSight macrocell. For information on CoreSight, the following reference documents are available from the ARM web site (www.arm.com):

- Cortex-M7 User Guide Reference Manual (ARM DUI 0644)
- Cortex-M7 Technical Reference Manual (ARM DDI 0489)
- CoreSight Technology System Design Guide (ARM DGI 0012)
- CoreSight Components Technical Reference Manual (ARM DDI 0314)
- ARM Debug Interface v5 Architecture Specification (Doc. ARM IHI 0031)
- ARMv7-M Architecture Reference Manual (ARM DDI 0403)

## 24.5 Register Summary

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | WDT_CR | 7:0 | | | | | | | | WDRSTT |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | KEY[7:0] | | | | | | | |
| 0x04 | WDT_MR | 7:0 | WDV[7:0] | | | | | | | |
| | | 15:8 | WDDIS | | WDRSTEN | WDFIEN | WDV[11:8] | | | |
| | | 23:16 | WDD[7:0] | | | | | | | |
| | | 31:24 | | | WDIDLEHLT | WDDBGHLT | WDD[11:8] | | | |
| 0x08 | WDT_SR | 7:0 | | | | | | | WDERR | WDUNF |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |

**Figure 32-2. I/O Line Control Logic**




### 32.5.1 Pullup and Pulldown Resistor Control

Each I/O line is designed with an embedded pullup resistor and an embedded pulldown resistor. The pullup resistor can be enabled or disabled by writing to the Pull-Up Enable Register (PIO_PUER) or Pull-Up Disable Register (PIO_PUDR), respectively. Writing to these registers results in setting or clearing the corresponding bit in the Pull-Up Status Register (PIO_PUSR). Reading a one in PIO_PUSR means the pullup is disabled and reading a zero means the pullup is enabled. The pulldown resistor can be enabled or disabled by writing the Pull-Down Enable Register (PIO_PPDER) or the Pull-Down Disable Register (PIO_PPDDR), respectively. Writing in these registers results in setting or clearing the corresponding bit in the Pull-Down Status Register (PIO_PPDSR). Reading a one in PIO_PPDSR means the pullup is disabled and reading a zero means the pulldown is enabled.

Enabling the pulldown resistor while the pullup resistor is still enabled is not possible. In this case, the write of PIO_PPDER for the relevant I/O line is discarded. Likewise, enabling the pullup resistor while the pulldown resistor is still enabled is not possible. In this case, the write of PIO_PUER for the relevant I/O line is discarded.

Control of the pullup resistor is possible regardless of the configuration of the I/O line.

### 34.7.11 SDRAMC OCMS Register

**Name:** SDRAMC_OCMS
**Offset:** 0x2C
**Reset:** 0x00000000
**Property:** Read/Write

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SDR_SE |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – SDR_SE** SDRAM Memory Controller Scrambling Enable

| Value | Description |
|---|---|
| 0 | Disables off-chip scrambling for SDR-SDRAM access. |
| 1 | Enables off-chip scrambling for SDR-SDRAM access. |

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0544 | USBHS_HSTPIPISR5 (INTPIPES) | 7:0 | SHORTPACKETI | RXSTALLDI | OVERFI | NAKEDI | PERRI | UNDERFI | TXOUTI | RXINI |
| | | 15:8 | CURRBK[1:0] | | NBUSYBK[1:0] | | | | DTSEQ[1:0] | |
| | | 23:16 | PBYCT[3:0] | | | | | CFGOK | | RWALL |
| | | 31:24 | | PBYCT[10:4] | | | | | | |
| 0x0544 | USBHS_HSTPIPISR5 (ISOPIPES) | 7:0 | SHORTPACKETI | CRCERRI | OVERFI | NAKEDI | PERRI | UNDERFI | TXOUTI | RXINI |
| | | 15:8 | CURRBK[1:0] | | NBUSYBK[1:0] | | | | DTSEQ[1:0] | |
| | | 23:16 | PBYCT[3:0] | | | | | CFGOK | | RWALL |
| | | 31:24 | | PBYCT[10:4] | | | | | | |
| 0x0548 | USBHS_HSTPIPISR6 | 7:0 | SHORTPACKETI | RXSTALLDI | OVERFI | NAKEDI | PERRI | TXSTPI | TXOUTI | RXINI |
| | | 15:8 | CURRBK[1:0] | | NBUSYBK[1:0] | | | | DTSEQ[1:0] | |
| | | 23:16 | PBYCT[3:0] | | | | | CFGOK | | RWALL |
| | | 31:24 | | PBYCT[10:4] | | | | | | |
| 0x0548 | USBHS_HSTPIPISR6 (INTPIPES) | 7:0 | SHORTPACKETI | RXSTALLDI | OVERFI | NAKEDI | PERRI | UNDERFI | TXOUTI | RXINI |
| | | 15:8 | CURRBK[1:0] | | NBUSYBK[1:0] | | | | DTSEQ[1:0] | |
| | | 23:16 | PBYCT[3:0] | | | | | CFGOK | | RWALL |
| | | 31:24 | | PBYCT[10:4] | | | | | | |
| 0x0548 | USBHS_HSTPIPISR6 (ISOPIPES) | 7:0 | SHORTPACKETI | CRCERRI | OVERFI | NAKEDI | PERRI | UNDERFI | TXOUTI | RXINI |
| | | 15:8 | CURRBK[1:0] | | NBUSYBK[1:0] | | | | DTSEQ[1:0] | |
| | | 23:16 | PBYCT[3:0] | | | | | CFGOK | | RWALL |
| | | 31:24 | | PBYCT[10:4] | | | | | | |
| 0x054C | USBHS_HSTPIPISR7 | 7:0 | SHORTPACKETI | RXSTALLDI | OVERFI | NAKEDI | PERRI | TXSTPI | TXOUTI | RXINI |
| | | 15:8 | CURRBK[1:0] | | NBUSYBK[1:0] | | | | DTSEQ[1:0] | |
| | | 23:16 | PBYCT[3:0] | | | | | CFGOK | | RWALL |
| | | 31:24 | | PBYCT[10:4] | | | | | | |
| 0x054C | USBHS_HSTPIPISR7 (INTPIPES) | 7:0 | SHORTPACKETI | RXSTALLDI | OVERFI | NAKEDI | PERRI | UNDERFI | TXOUTI | RXINI |
| | | 15:8 | CURRBK[1:0] | | NBUSYBK[1:0] | | | | DTSEQ[1:0] | |
| | | 23:16 | PBYCT[3:0] | | | | | CFGOK | | RWALL |
| | | 31:24 | | PBYCT[10:4] | | | | | | |
| 0x054C | USBHS_HSTPIPISR7 (ISOPIPES) | 7:0 | SHORTPACKETI | CRCERRI | OVERFI | NAKEDI | PERRI | UNDERFI | TXOUTI | RXINI |
| | | 15:8 | CURRBK[1:0] | | NBUSYBK[1:0] | | | | DTSEQ[1:0] | |
| | | 23:16 | PBYCT[3:0] | | | | | CFGOK | | RWALL |
| | | 31:24 | | PBYCT[10:4] | | | | | | |
| 0x0550 | USBHS_HSTPIPISR8 | 7:0 | SHORTPACKETI | RXSTALLDI | OVERFI | NAKEDI | PERRI | TXSTPI | TXOUTI | RXINI |
| | | 15:8 | CURRBK[1:0] | | NBUSYBK[1:0] | | | | DTSEQ[1:0] | |
| | | 23:16 | PBYCT[3:0] | | | | | CFGOK | | RWALL |
| | | 31:24 | | PBYCT[10:4] | | | | | | |

details). To send a suspend or a resume command, the host must set the SDIO Special Command field (IOSPCMD) in the HSMCI Command Register.

### 40.9.2 SDIO Interrupts

Each function within an SDIO or Combo card may implement interrupts (Refer to the SDIO Specification for more details). In order to allow the SDIO card to interrupt the host, an interrupt function is added to a pin on the DAT[1] line to signal the card's interrupt to the host. An SDIO interrupt on each slot can be enabled through the HSMCI Interrupt Enable Register. The SDIO interrupt is sampled regardless of the currently selected slot.

## 40.10 CE-ATA Operation

CE-ATA maps the streamlined ATA command set onto the MMC interface. The ATA task file is mapped onto MMC register space.

CE-ATA utilizes five MMC commands:

- GO_IDLE_STATE (CMD0): used for hard reset.
- STOP_TRANSMISSION (CMD12): causes the ATA command currently executing to be aborted.
- FAST_IO (CMD39): Used for single register access to the ATA taskfile registers, 8-bit access only.
- RW_MULTIPLE_REGISTERS (CMD60): used to issue an ATA command or to access the control/status registers.
- RW_MULTIPLE_BLOCK (CMD61): used to transfer data for an ATA command.

CE-ATA utilizes the same MMC command sequences for initialization as traditional MMC devices.

### 40.10.1 Executing an ATA Polling Command

1. Issue READ_DMA_EXT with RW_MULTIPLE_REGISTER (CMD60) for 8 KB of DATA.
2. Read the ATA status register until DRQ is set.
3. Issue RW_MULTIPLE_BLOCK (CMD61) to transfer DATA.
4. Read the ATA status register until DRQ && BSY are configured to 0.

### 40.10.2 Executing an ATA Interrupt Command

1. Issue READ_DMA_EXT with RW_MULTIPLE_REGISTER (CMD60) for 8 KB of DATA with nIEN field set to zero to enable the command completion signal in the device.
2. Issue RW_MULTIPLE_BLOCK (CMD61) to transfer DATA.
3. Wait for Completion Signal Received Interrupt.

### 40.10.3 Aborting an ATA Command

If the host needs to abort an ATA command prior to the completion signal it must send a special command to avoid potential collision on the command line. The SPCMD field of the HSMCI_CMDR must be set to 3 to issue the CE-ATA completion Signal Disable Command.

### 40.10.4 CE-ATA Error Recovery

Several methods of ATA command failure may occur, including:

- No response to an MMC command, such as RW_MULTIPLE_REGISTER (CMD60).
- CRC is invalid for an MMC command or response.
- CRC16 is invalid for an MMC data packet.

## 40.14 Register Summary

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | HSMCI_CR | 7:0 | SWRST | | | | PWSDIS | PWSEN | MCIDIS | MCIEN |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x00 | HSMCI_FIFOx [x=0..255] | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x04 | HSMCI_MR | 7:0 | CLKDIV[7:0] | | | | | | | |
| | | 15:8 | | PADV | FBYTE | WRPROOF | RDPROOF | PWSDIV[2:0] | | |
| | | 23:16 | | | | | | | | CLKODD |
| | | 31:24 | | | | | | | | |
| 0x08 | HSMCI_DTOR | 7:0 | | DTOMUL[2:0] | | | DTOCYC[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | HSMCI_SDCR | 7:0 | SDCBUS[1:0] | | | | | | SDCSEL[1:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | HSMCI_ARGR | 7:0 | ARG[7:0] | | | | | | | |
| | | 15:8 | ARG[15:8] | | | | | | | |
| | | 23:16 | ARG[23:16] | | | | | | | |
| | | 31:24 | ARG[31:24] | | | | | | | |
| 0x14 | HSMCI_CMDR | 7:0 | RSPTYP[1:0] | | CMDNB[5:0] | | | | | |
| | | 15:8 | | | | MAXLAT | OPDCMD | SPCMD[2:0] | | |
| | | 23:16 | | | TRTYP[2:0] | | | TRDIR | TRCMD[1:0] | |
| | | 31:24 | | | | | BOOT_ACK | ATACS | IOSPCMD[1:0] | |
| 0x18 | HSMCI_BLKR | 7:0 | BCNT[7:0] | | | | | | | |
| | | 15:8 | BCNT[15:8] | | | | | | | |
| | | 23:16 | BLKLEN[7:0] | | | | | | | |
| | | 31:24 | BLKLEN[15:8] | | | | | | | |
| 0x1C | HSMCI_CSTOR | 7:0 | | CSTOMUL[2:0] | | | CSTOCYC[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | HSMCI_RSPR | 7:0 | RSP[7:0] | | | | | | | |
| | | 15:8 | RSP[15:8] | | | | | | | |
| | | 23:16 | RSP[23:16] | | | | | | | |
| | | 31:24 | RSP[31:24] | | | | | | | |
| 0x24 ... 0x2F | Reserved | | | | | | | | | |

### 41.8.10 SPI Write Protection Mode Register

**Name:** SPI_WPMR
**Offset:** 0xE4
**Reset:** 0x0
**Property:** Read/Write

See section Register Write Protection for the list of registers that can be write-protected.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | WPKEY[23:16] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | WPKEY[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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | WPKEY[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 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | WPCREN | WPITEN | WPEN |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bits 31:8 – WPKEY[23:0]** Write Protection Key

| Value | Name | Description |
|---|---|---|
| 0x535049 | PASSWD | Writing any other value in this field aborts the write operation of the WPEN bit.<br>Always reads as 0. |

**Bit 2 – WPCREN** Write Protection Control Register Enable

| Value | Description |
|---|---|
| 0 | Disables the write protection on the Control register if WPKEY corresponds to 0x535049. |
| 1 | Enables the write protection on the Control register if WPKEY corresponds to 0x535049. |

**Bit 1 – WPITEN** Write Protection Interrupt Enable

| Value | Description |
|---|---|
| 0 | Disables the write protection on Interrupt registers if WPKEY corresponds to 0x535049. |
| 1 | Enables the write protection on Interrupt registers if WPKEY corresponds to 0x535049. |

**Bit 0 – WPEN** Write Protection Enable

### 43.7.4 TWIHS Internal Address Register

**Name:** TWIHS_IADR
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** Read/Write

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | IADR[23:16] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | IADR[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 |
|---|---|---|---|---|---|---|---|---|
| | IADR[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 |

**Bits 23:0 – IADR[23:0]** Internal Address
0, 1, 2 or 3 bytes depending on IADRSZ.

### 43.7.6 TWIHS Status Register

**Name:** TWIHS_SR
**Offset:** 0x20
**Reset:** 0x03000009
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | SDA | SCL |
| Access | | | | | | | R | R |
| Reset | | | | | | | 1 | 1 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | SMBHHM | SMBDAM | PECERR | TOUT | | MCACK |
| Access | | | R | R | R | R | | R |
| Reset | | | 0 | 0 | 0 | 0 | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | EOSACC | SCLWS | ARBLST | NACK |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | UNRE | OVRE | GACC | SVACC | SVREAD | TXRDY | RXRDY | TXCOMP |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

**Bit 25 – SDA** SDA Line Value

| Value | Description |
|---|---|
| 0 | SDA line sampled value is ‘0’. |
| 1 | SDA line sampled value is ‘1’. |

**Bit 24 – SCL** SCL Line Value

| Value | Description |
|---|---|
| 0 | SCL line sampled value is ‘0’. |
| 1 | SCL line sampled value is ‘1.’ |

**Bit 21 – SMBHHM** SMBus Host Header Address Match (cleared on read)

| Value | Description |
|---|---|
| 0 | No SMBus Host Header Address received since the last read of TWIHS_SR. |
| 1 | An SMBus Host Header Address was received since the last read of TWIHS_SR. |

**Bit 20 – SMBDAM** SMBus Default Address Match (cleared on read)

| Value | Description |
|---|---|
| 0 | No SMBus Default Address received since the last read of TWIHS_SR. |
| 1 | An SMBus Default Address was received since the last read of TWIHS_SR. |

1: When both holding register and internal shifter are empty and STOP condition has been sent.

TXCOMP behavior in Master mode can be seen in Master Write with One-Byte Internal Address and Multiple Data Bytes and in Master Read with Multiple Data Bytes.

- TXCOMP used in Slave mode:

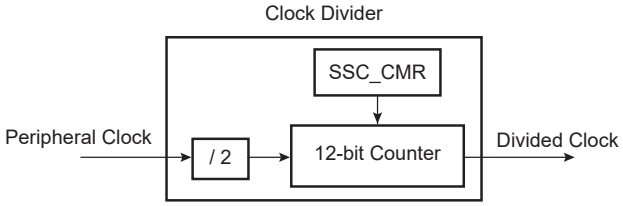
0: As soon as a START is detected.

1: After a STOP or a REPEATED START + an address different from SADR is detected.

TXCOMP behavior in Slave mode can be seen in Clock Stretching in Read Mode, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

#### 44.8.1.1 Clock Divider

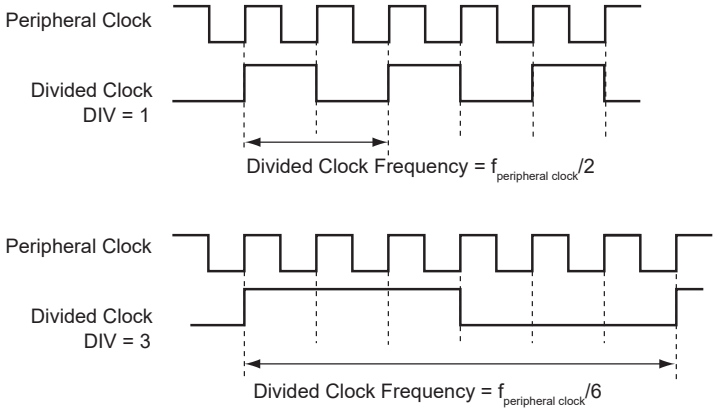
**Figure 44-7. Divided Clock Block Diagram**




The peripheral clock divider is determined by the 12-bit field DIV counter and comparator (so its maximal value is 4095) in the Clock Mode Register (SSC_CMR), allowing a peripheral clock division by up to 8190. The Divided Clock is provided to both the receiver and the transmitter. When this field is programmed to 0, the Clock Divider is not used and remains inactive.

When DIV is set to a value equal to or greater than 1, the Divided Clock has a frequency of peripheral clock divided by 2 times DIV. Each level of the Divided Clock has a duration of the peripheral clock multiplied by DIV. This ensures a 50% duty cycle for the Divided Clock regardless of whether the DIV value is even or odd.

**Figure 44-8. Divided Clock Generation**




#### 44.8.1.2 Transmit Clock Management

The transmit clock is generated from the receive clock or the divider clock or an external clock scanned on the TK I/O pad. The transmit clock is selected by the CKS field in the Transmit Clock Mode Register (SSC_TCMR). Transmit Clock can be inverted independently by the CKI bits in the SSC_TCMR.

The transmitter can also drive the TK I/O pad continuously or be limited to the current data transfer. The clock output is configured by the SSC_TCMR. The Transmit Clock Inversion (CKI) bits have no effect on the clock outputs. Programming the SSC_TCMR to select TK pin (CKS field) and at the same time Continuous Transmit Clock (CKO field) can lead to unpredictable results.

### 44.9.13 SSC Status Register

**Name:** SSC_SR
**Offset:** 0x40
**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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | RXEN | TXEN |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | RXSYN | TXSYN | CP1 | CP0 |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | OVRUN | RXRDY | | | TXEMPTY | TXRDY |
| Access | | | R | R | | | R | R |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bit 17 – RXEN** Receive Enable

| Value | Description |
|---|---|
| 0 | Receive is disabled. |
| 1 | Receive is enabled. |

**Bit 16 – TXEN** Transmit Enable

| Value | Description |
|---|---|
| 0 | Transmit is disabled. |
| 1 | Transmit is enabled. |

**Bit 11 – RXSYN** Receive Sync

| Value | Description |
|---|---|
| 0 | An Rx Sync has not occurred since the last read of the Status Register. |
| 1 | An Rx Sync has occurred since the last read of the Status Register. |

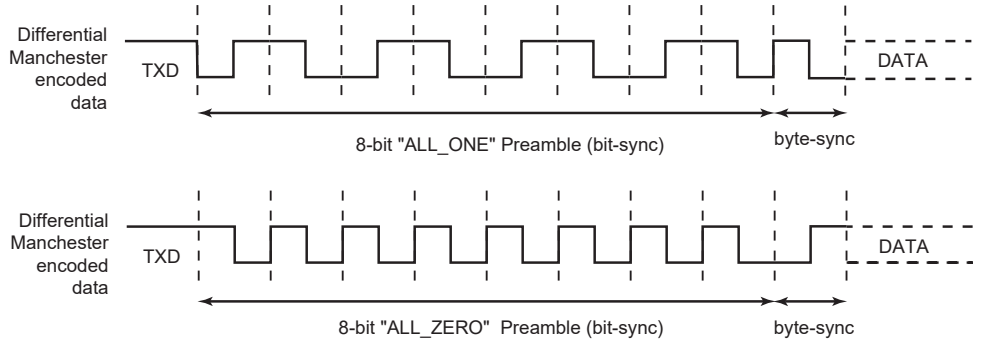
**Bit 10 – TXSYN** Transmit Sync

| Value | Description |
|---|---|
| 0 | A Tx Sync has not occurred since the last read of the Status Register. |
| 1 | A Tx Sync has occurred since the last read of the Status Register. |

The LON implementation allows two different preamble patterns ALL_ONE and ALL_ZERO which can be configured via US_MAN.TX_PL. The following figure illustrates and defines the valid patterns.

Other preamble patterns are not supported.

**Figure 46-57. Preamble Patterns**




#### 46.6.10.5.3 Preamble Reception

LON received frames begin with a preamble of variable length. The receiving algorithm does not check the preamble length, although a minimum of length of 4 bits is required for the receiving algorithm to consider the received preamble as valid.

As is the case with LON preamble transmission, two preamble patterns (ALL_ONE and ALL_ZERO) are allowed and can be configured through US_MAN.RX_PL. Figure 46-57 illustrates and defines the valid patterns.

Other preamble patterns are not supported.

#### 46.6.10.5.4 Header Transmission

Each LON frame, after sending the preamble, starts with the frame header also called L2HDR according to the CEA-709 specification. This header consist of the priority bit, the alternative path bit and the backlog increment. It is the first data to be sent.

In LON mode the transmitting algorithm starts when the US_LONL2HDR register is written (it is the first data to send).

#### 46.6.10.5.5 Header Reception

Each LON frame, after receiving the preamble, receives the frame header also called L2HDR according to the CEA-709 specification. This header consists of the priority bit, the alternative path bit, and the backlog increment.

The frame header is the first received data and the RXRDY bit rises as soon as the frame header as been received and stored in the Receive Holding register (US_RHR).

#### 46.6.10.5.6 Data

Data are sent/received serially after the preamble transmission/reception. Data can be either sent/ received MSB first or LSB first depending on US_MR.MSBF.

#### 46.6.10.5.7 CRC

The two last bytes of LON frames are dedicated to CRC.

When transmitting, the CRC of the frame is automatically generated and sent when expected.

When receiving frames the CRC is automatically checked and a LCRCE flag is set in US_CSR if the calculated CRC do not match the received one. Note that the two received CRC bytes are seen as two additional data from the user point of view.

### 48.7.11 HBI Channel Error 0 Register

**Name:** MLB_HCER0
**Offset:** 0x090
**Reset:** 0x00000000
**Property:** Read-only

The HBI Channel Error Registers (HCERn) indicate which channel(s) have encountered fatal errors.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CERR: Bitwise Channel Error Bit [31[31:24] | | | | | | | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CERR: Bitwise Channel Error Bit [31[23:16] | | | | | | | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CERR: Bitwise Channel Error Bit [31[15:8] | | | | | | | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CERR: Bitwise Channel Error Bit [31[7:0] | | | | | | | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – CERR: Bitwise Channel Error Bit [31[31:0]** 0]
CERR[n] = 1 indicates that a fatal error occurred on channel n.

### 49.6.18 MCAN Interrupt Line Select Register

**Name:** MCAN_ILS
**Offset:** 0x58
**Reset:** 0x00000000
**Property:** Read/Write

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register to one of the two module interrupt lines.

0: Interrupt assigned to interrupt line MCAN_INT0.

1: Interrupt assigned to interrupt line MCAN_INT1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | ARAL | PEDL | PEAL | WDIL | BOL | EWL |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | EPL | ELOL | | | DRXL | TOOL | MRAFL | TSWL |
| Access | R/W | R/W | | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TEFLL | TEFFL | TEFWL | TEFNL | TFEL | TCFL | TCL | HPML |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | RF1LL | RF1FL | RF1WL | RF1NL | RF0LL | RF0FL | RF0WL | RF0NL |
| 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 29 – ARAL** Access to Reserved Address Line

**Bit 28 – PEDL** Protocol Error in Data Phase Line

**Bit 27 – PEAL** Protocol Error in Arbitration Phase Line

**Bit 26 – WDIL** Watchdog Interrupt Line

**Bit 25 – BOL** Bus_Off Status Interrupt Line

**Bit 24 – EWL** Warning Status Interrupt Line

**Bit 23 – EPL** Error Passive Interrupt Line

**Bit 22 – ELOL** Error Logging Overflow Interrupt Line

**Bit 19 – DRXL** Message stored to Dedicated Receive Buffer Interrupt Line

As in Capture mode, the SYNC signal and the software trigger are also available as triggers. RC Compare can also be used as a trigger depending on the parameter WAVSEL.

### 50.6.14 Synchronization with PWM

The inputs TIOAx/TIOBx can be bypassed, and thus channel trigger/capture events can be directly driven by the independent PWM module.

PWM comparator outputs (internal signals without dead-time insertion - OCx), respectively source of the PWMH/L[2:0] outputs, are routed to the internal TC inputs. These specific TC inputs are multiplexed with TIOA/B input signal to drive the internal trigger/capture events.

The selection is made in the Extended Mode register (TC_EMR) fields TRIGSRCA and TRIGSRCB (see “TC Extended Mode Register”).

Each channel of the TC module can be synchronized by a different PWM channel as described in the following figure.

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x8C | AES_TAGR1 | 7:0 | TAG[7:0] | | | | | | | |
| | | 15:8 | TAG[15:8] | | | | | | | |
| | | 23:16 | TAG[23:16] | | | | | | | |
| | | 31:24 | TAG[31:24] | | | | | | | |
| 0x90 | AES_TAGR2 | 7:0 | TAG[7:0] | | | | | | | |
| | | 15:8 | TAG[15:8] | | | | | | | |
| | | 23:16 | TAG[23:16] | | | | | | | |
| | | 31:24 | TAG[31:24] | | | | | | | |
| 0x94 | AES_TAGR3 | 7:0 | TAG[7:0] | | | | | | | |
| | | 15:8 | TAG[15:8] | | | | | | | |
| | | 23:16 | TAG[23:16] | | | | | | | |
| | | 31:24 | TAG[31:24] | | | | | | | |
| 0x98 | AES_CTRR | 7:0 | CTR[7:0] | | | | | | | |
| | | 15:8 | CTR[15:8] | | | | | | | |
| | | 23:16 | CTR[23:16] | | | | | | | |
| | | 31:24 | CTR[31:24] | | | | | | | |
| 0x9C | AES_GCMHR0 | 7:0 | H[7:0] | | | | | | | |
| | | 15:8 | H[15:8] | | | | | | | |
| | | 23:16 | H[23:16] | | | | | | | |
| | | 31:24 | H[31:24] | | | | | | | |
| 0xA0 | AES_GCMHR1 | 7:0 | H[7:0] | | | | | | | |
| | | 15:8 | H[15:8] | | | | | | | |
| | | 23:16 | H[23:16] | | | | | | | |
| | | 31:24 | H[31:24] | | | | | | | |
| 0xA4 | AES_GCMHR2 | 7:0 | H[7:0] | | | | | | | |
| | | 15:8 | H[15:8] | | | | | | | |
| | | 23:16 | H[23:16] | | | | | | | |
| | | 31:24 | H[31:24] | | | | | | | |
| 0xA8 | AES_GCMHR3 | 7:0 | H[7:0] | | | | | | | |
| | | 15:8 | H[15:8] | | | | | | | |
| | | 23:16 | H[23:16] | | | | | | | |
| | | 31:24 | H[31:24] | | | | | | | |

### 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. |