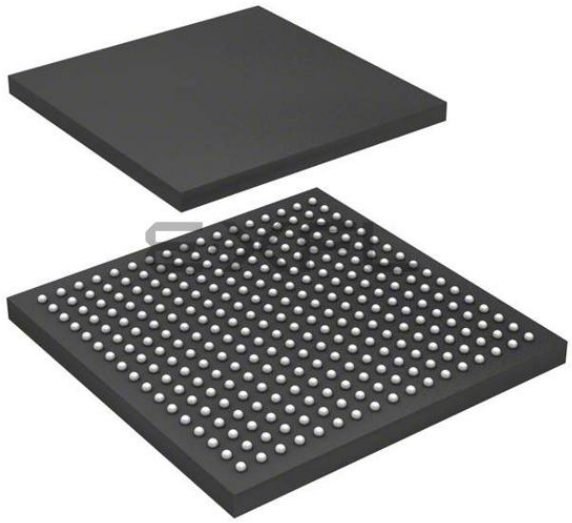


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Active
Core Processor	ARM® Cortex®-A5
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	500MHz
Co-Processors/DSP	Multimedia; NEON™ MPE
RAM Controllers	LPDDR1, LPDDR2, LPDDR3, DDR2, DDR3, DDR3L, QSPI
Graphics Acceleration	Yes
Display & Interface Controllers	Keyboard, LCD, Touchscreen
Ethernet	10/100Mbps (1)
SATA	-
USB	USB 2.0 + HSIC
Voltage - I/O	3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	ARM TZ, Boot Security, Cryptography, RTIC, Secure Fusebox, Secure JTAG, Secure Memory, Secure RTC
Package / Case	289-LFBGA
Supplier Device Package	289-LFBGA (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d28a-cur

18.13.10 Write Protection Mode Register

Name: MATRIX_WPMR

Address: 0xF00181E4 (0), 0xFC03C1E4 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
WPKEY							

23	22	21	20	19	18	17	16
WPKEY							

15	14	13	12	11	10	9	8
WPKEY							

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	WPEN

WPEN: Write Protection Enable

0: Disables the Write Protection if WPKEY corresponds to 0x4D4154 (“MAT” in ASCII).

1: Enables the Write Protection if WPKEY corresponds to 0x4D4154 (“MAT” in ASCII).

See Section 18.11 “Register Write Protection” for list of registers that can be write-protected.

WPKEY: Write Protection Key (Write-only)

Value	Name	Description
0x4D4154	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

21.9.21 AIC Write Protection Status Register

Name: AIC_WPSR

Address: 0xFC0200E8 (AIC), 0xF803C0E8 (SAIC)

Access: Read-only

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

23	22	21	20	19	18	17	16
WPVSRC							

15	14	13	12	11	10	9	8
WPVSRC							

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	WPVS

WPVS: Write Protection Violation Status

0: No write protection violation has occurred since the last read of AIC_WPSR.

1: A write protection violation has occurred since the last read of AIC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

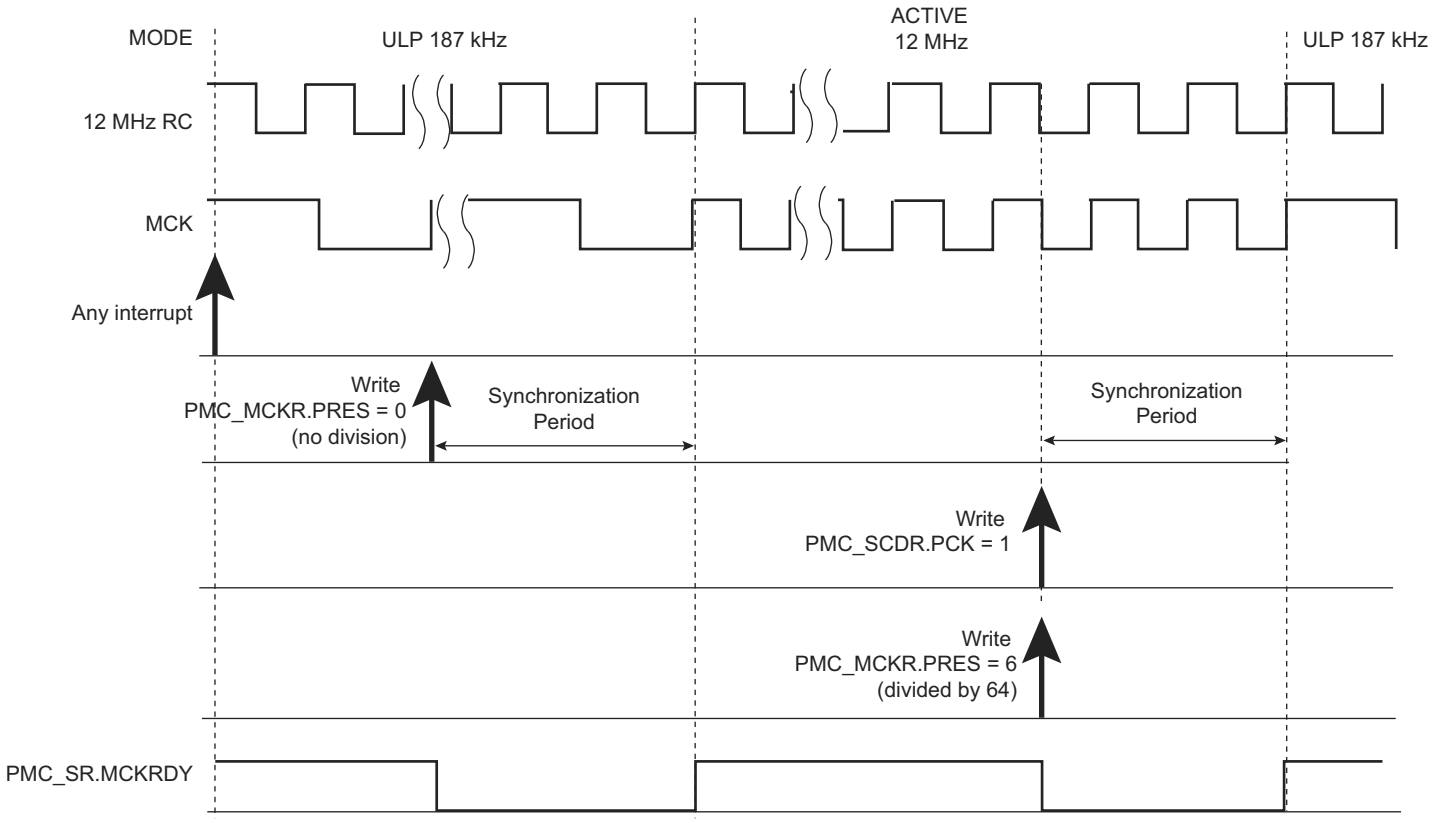
WPVSRC: Write Protection Violation Source

When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

33.14 Fast Startup from Ultra Low-power (ULP) Mode 0

In Ultra Low-power (ULP) mode 0, the Main clock (MAINCK) must be running, thus either the 12 MHz crystal oscillator or the Fast RC oscillator must be enabled. The lowest power consumption that can be achieved in ULP Mode 0, can be obtained when dividing the selected oscillator frequency by 64 by writing PMC_MCKR.PRES to 6. Any interrupt exits the system from ULP Mode. The software must write PMC_MCKR.PRES to 1 to provide MCK with the fastest clock. If the PLL is used, the startup procedure must be done prior to writing PMC_MCKR.PRES to 1. Figure 33-5 illustrates an example of startup phase from ULP Mode 0 without use of PLL.

Figure 33-5: Fast Startup from Ultra Low-Power Mode 0



Warning: The duration of the WKUPx pins active level must be greater than four MAINCK cycles.

36.6.3 DDR-SDRAM Address Mapping for Low-cost Memories

Table 36-27: Sequential Mapping for DDR-SDRAM Configuration, 2K Rows, 512 Columns, 2 banks, 16 bits

CPU Address Line																									
27	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
						Bk	Row[10:0]											Column[8:0]									M0

Table 36-28: Interleaved Mapping for DDR-SDRAM Configuration, 2K Rows, 512 Columns, 2 banks, 16 bits

CPU Address Line																									
27	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
						Row[10:0]											Bk	Column[8:0]									M0

Table 36-29: Sequential Mapping for DDR-SDRAM Configuration: 4K Rows, 256 Columns, 2 banks, 32 bits

CPU Address Line																									
27	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
						Bk	Row[11:0]											Column[7:0]								M[1:0]	

Table 36-30: Interleaved Mapping for DDR-SDRAM Configuration: 4K Rows, 256 Columns, 2 banks, 32 bits

CPU Address Line																									
27	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
						Row[11:0]											Bk	Column[7:0]								M[1:0]	

Note 1: M[1:0] is the byte address inside a 32-bit word.

2: Bk[2] = BA2, Bk[1] = BA1, Bk[0] = BA0

1: Number of requests or words is provided by software, see “NRQ_NWD_BDW_Px: Number of Requests, Number of Words or Bandwidth Allocation from Port 0-1-2-3” .

38.9.21 XDMAC Channel x [x = 0..15] Interrupt Status Register

Name: XDMAC_CISx [x = 0..15]

Address: 0xF000405C (1)[0], 0xF000409C (1)[1], 0xF00040DC (1)[2], 0xF000411C (1)[3], 0xF000415C (1)[4], 0xF000419C (1)[5], 0xF00041DC (1)[6], 0xF000421C (1)[7], 0xF000425C (1)[8], 0xF000429C (1)[9], 0xF00042DC (1)[10], 0xF000431C (1)[11], 0xF000435C (1)[12], 0xF000439C (1)[13], 0xF00043DC (1)[14], 0xF000441C (1)[15], 0xF001005C (0)[0], 0xF001009C (0)[1], 0xF00100DC (0)[2], 0xF001011C (0)[3], 0xF001015C (0)[4], 0xF001019C (0)[5], 0xF00101DC (0)[6], 0xF001021C (0)[7], 0xF001025C (0)[8], 0xF001029C (0)[9], 0xF00102DC (0)[10], 0xF001031C (0)[11], 0xF001035C (0)[12], 0xF001039C (0)[13], 0xF00103DC (0)[14], 0xF001041C (0)[15]

Access: Read-only

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

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS

BIS: End of Block Interrupt Status Bit

0: End of block interrupt has not occurred.

1: End of block interrupt has occurred since the last read of the Status register.

LIS: End of Linked List Interrupt Status Bit

0: End of linked list condition has not occurred.

1: End of linked list condition has occurred since the last read of the Status register.

DIS: End of Disable Interrupt Status Bit

0: End of disable condition has not occurred.

1: End of disable condition has occurred since the last read of the Status register.

FIS: End of Flush Interrupt Status Bit

0: End of flush condition has not occurred.

1: End of flush condition has occurred since the last read of the Status register.

RBEIS: Read Bus Error Interrupt Status Bit

0: Read bus error condition has not occurred.

1: At least one bus error has been detected in a read access since the last read of the Status register.

WBEIS: Write Bus Error Interrupt Status Bit

0: Write bus error condition has not occurred.

1: At least one bus error has been detected in a write access since the last read of the Status register.

ROIS: Request Overflow Error Interrupt Status Bit

0: Overflow condition has not occurred.

1: Overflow condition has occurred at least once. (This information is only relevant for peripheral synchronized transfers.)

39.7.39 Overlay 1 Interrupt Mask Register

Name: LCDC_OVR1IMR

Address: 0xF0000154

Access: Read-only

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

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	OVR	DONE	ADD	DSCR	DMA	–	–

DMA: End of DMA Transfer Interrupt Mask

0: Interrupt source is disabled

1: Interrupt source is enabled

DSCR: Descriptor Loaded Interrupt Mask

0: Interrupt source is disabled

1: Interrupt source is enabled

ADD: Head Descriptor Loaded Interrupt Mask

0: Interrupt source is disabled

1: Interrupt source is enabled

DONE: End of List Interrupt Mask

0: Interrupt source is disabled

1: Interrupt source is enabled

OVR: Overflow Interrupt Mask

0: Interrupt source is disabled

1: Interrupt source is enabled

39.7.105 High-End Overlay Configuration Register 10

Name: LCDC_HEOCFG10

Address: 0xF00003B4

Access: Read/Write

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

23	22	21	20	19	18	17	16
RKEY							

15	14	13	12	11	10	9	8
GKEY							

7	6	5	4	3	2	1	0
BKEY							

RKEY: Red Color Component Chroma Key

Reference Red chroma key used to match the Red color of the current overlay.

GKEY: Green Color Component Chroma Key

Reference Green chroma key used to match the Green color of the current overlay.

BKEY: Blue Color Component Chroma Key

Reference Blue chroma key used to match the Blue color of the current overlay.

The GMAC features 3 interrupt sources. Refer to Table 11-1 "Peripheral Identifiers" for the interrupt numbers for GMAC priority queues.

Table 40-3: Peripheral IDs

Instance	ID
GMAC	5

40.6 Functional Description

40.6.1 Media Access Controller

The Media Access Controller (MAC) transmit block takes data from FIFO, adds preamble and, if necessary, pad and frame check sequence (FCS). Both half duplex and full duplex Ethernet modes of operation are supported. When operating in half duplex mode, the MAC transmit block generates data according to the carrier sense multiple access with collision detect (CSMA/CD) protocol. The start of transmission is deferred if carrier sense (CRS) is active. If collision (COL) becomes active during transmission, a jam sequence is asserted and the transmission is retried after a random backoff. The CRS and COL signals have no effect in full duplex mode.

The MAC receive block checks for valid preamble, FCS, alignment and length, and presents received frames to the MAC address checking block and FIFO. Software can configure the GMAC to receive jumbo frames up to 10240bytes. It can optionally strip CRC from the received frame prior to transfer to FIFO.

The address checker recognizes four specific 48-bit addresses, can recognize four different type ID values, and contains a 64-bit Hash register for matching multicast and unicast addresses as required. It can recognize the broadcast address of all ones and copy all frames. The MAC can also reject all frames that are not VLAN tagged and recognize Wake on LAN events.

The MAC receive block supports offloading of IP, TCP and UDP checksum calculations (both IPv4 and IPv6 packet types supported), and can automatically discard bad checksum frames.

40.6.2 1588 Timestamp Unit

The 1588 timestamp unit (TSU) is implemented as a 94-bit timer.

The 48 upper bits [93:46] of the timer count seconds and are accessible in the “GMAC 1588 Timer Seconds High Register” (GMAC_TSH) and “GMAC 1588 Timer Seconds Low Register” (GMAC_TSL). The 30 lower bits [45:16] of the timer count nanoseconds and are accessible in the “GMAC 1588 Timer Nanoseconds Register” (GMAC_TN). The lowest 16 bits [15:0] of the timer count sub-nanoseconds.

The 46 lower bits roll over when they have counted to one second. The timer increments by a programmable period (to approximately 15.2 femtoseconds resolution) with each MCK period and can also be adjusted in 1ns resolution (incremented or decremented) through APB register accesses.

40.6.3 AHB Direct Memory Access Interface

The GMAC DMA controller is connected to the MAC FIFO interface and provides a scatter-gather type capability for packet data storage.

The DMA implements packet buffering where dual-port memories are used to buffer multiple frames.

This bit controls whether the host controller skips processing the Asynchronous Schedule.

0: Do not process the Asynchronous Schedule (default value).

1: Use the UHPHS_ASYNCLISTADDR register to access the Asynchronous Schedule.

IAAD: Interrupt on Async Advance Doorbell (read/write)

This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to ring the doorbell.

When the host controller has evicted all appropriate cached schedule state, it sets the Interrupt on Async Advance status bit in the UHPHS_USBSTS register. If the Interrupt on Async Advance Enable bit in the UHPHS_USBINTR register is set to 1, then the host controller will assert an interrupt at the next interrupt threshold.

The host controller sets this bit to 0 after it has set the Interrupt on Async Advance status bit in the UHPHS_USBSTS register to 1.

Software should not write a 1 to this bit when the asynchronous schedule is disabled. Doing so will yield undefined results.

LHCR: Light Host Controller Reset (optional) (read/write)

This control bit is not required. If implemented, it allows the driver to reset the EHCI controller without affecting the state of the ports or the relationship to the companion host controllers. For example, the UHPHS_PORTSC registers should not be reset to their default values and the CF bit setting should not go to 0 (retaining port ownership relationships).

A host software read of this bit as 0 indicates the Light Host Controller Reset has completed and it is safe for host software to reinitialize the host controller. A host software read of this bit as 1 indicates the Light Host Controller Reset has not yet completed.

If not implemented, a read of this field will always return a 0.

ASPMC: Asynchronous Schedule Park Mode Count (optional) (read/write or read-only)

If the Asynchronous Park Capability bit in the UHPHS_HCCPARAMS register is set to 1, then this field defaults to 3h and is R/W. Otherwise it defaults to 0 and is RO. It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. Valid values are 1h to 3h. Software must not write a 0 to this bit when Park Mode Enable is set to 1 as this will result in undefined behavior.

ASPME: Asynchronous Schedule Park Mode Enable (optional) (read/write or read-only)

If the Asynchronous Park Capability bit in the UHPHS_HCCPARAMS register is set to 1, then this bit defaults to a 1h and is R/W. Otherwise the bit must be a 0 and is RO. Software uses this bit to enable or disable Park mode. When this bit is set to 1, Park mode is enabled. When this bit is set to 0, Park mode is disabled.

ITC: Interrupt Threshold Control (read/write)

This field is used by system software to select the maximum rate at which the host controller will issue interrupts. The only valid values are defined below. If software writes an invalid value to this register, the results are undefined.

Value	Maximum Interrupt Interval
00h	Reserved
01h	1 micro-frame
02h	2 micro-frames
04h	4 micro-frames
08h	8 micro-frames (default, equates to 1 ms)
10h	16 micro-frames (2 ms)
20h	32 micro-frames (4 ms)
40h	64 micro-frames (8 ms)

Any other value in this register yields undefined results.

Software modifications to this bit while HCHalted bit is equal to 0 results in undefined behavior.

Similarly, for a Receive FIFO containing six data, the options are:

- Perform six TWIHS_RHR-byte read accesses.
- Perform three TWIHS_RHR-halfword read accesses.
- Perform one TWIHS_RHR-word read access and one TWIHS_RHR-halfword read access.

This mode can minimize the number of accesses by concatenating the data to send/read in one access.

- TXRDY and RXRDY Configuration

In Multiple Data mode, the TXRDYM and RXRDYM fields in the TWIHS_FMR become useful.

As in Multiple Data mode, it is possible to write several data in the same access it might be useful to configure TXRDY flag behavior to indicate if 1, 2 or 4 data can be written in the FIFO depending on the access to perform on TWIHS_THR.

If for instance four data are written each time in the TWIHS_THR it might be useful to configure TXRDYM field to 0x2 value so that TXRDY flag will be at ‘1’ only when at least four data can be written in the Transmit FIFO.

In the same way if four data are read each time in the TWIHS_RHR it might be useful to configure RXRDYM field to 0x2 value so that RXRDY flag will be at ‘1’ only when at least four unread data are in the Receive FIFO.

- DMAC

If DMAC transfer is used it is mandatory to configure TXRDYM/RXRDYM to the right value depending on the DMAC channel size (byte, halfword or word).

- Transmit FIFO Lock

If a frame is terminated early due to a not-acknowledge error (NACK flag), SMBus timeout error (TOUT flag) or master code acknowledge error (MACK flag), a lock is set on the Transmit FIFO preventing any new frame from being sent until it is cleared. This allows clearing the FIFO if needed, reset DMAC channels, etc., without any risk.

The LOCK bit in the TWIHS_SR is used to check the state of the Transmit FIFO lock.

The Transmit FIFO lock can be cleared setting the TXFLCLR bit to ‘1’ in the TWIHS_CR.

- FIFO Pointer Error

In some specific cases, it is possible to generate a FIFO pointer error.

- Transmit FIFO:

If the Transmit FIFO is full and a write access is performed on the TWIHS_THR, it will generate a Transmit FIFO pointer error and set the TXFPTEF flag in TWIHS_FSR.

In Multiple Data mode, if the number of data written in the TWIHS_THR (according to the register access size) is bigger than the Transmit FIFO free space, it will generate a Transmit FIFO pointer error and set the TXFPTEF flag in TWIHS_FSR.

- Receive FIFO:

In Multiple Data mode, if the number of data read in the TWIHS_RHR (according to the register access size) is bigger than the number of unread data in the Receive FIFO, it will generate a Receive FIFO pointer error and set the RXFPTEF flag in TWIHS_FSR.

Pointer error should not happen if FIFO state is checked before writing/reading in TWIHS_THR/TWIHS_RHR. FIFO state can be checked either with TXRDY, RXRDY, TXFL or RXFL. When a pointer error occurs, other FIFO flags might not behave as expected; their state should be ignored.

If a Transmit or Receive pointer error occurs, a software reset must be performed using the SWRST bit in the TWIHS_CR. Note that issuing a software while transmitting might leave a slave in an unknown state holding the TWD line. In such case, a Bus Clear Command will allow to make the slave release the TWD line (the first frame sent afterwards might not be received properly by the slave).

- FIFO Thresholds

Each Transmit and Receive FIFO includes a threshold feature used to set a flag and an interrupt when a FIFO threshold is crossed. Thresholds are defined as a number of data in the FIFO, and the FIFO state (TXFL or RXFL) represents the number of data currently in the FIFO.

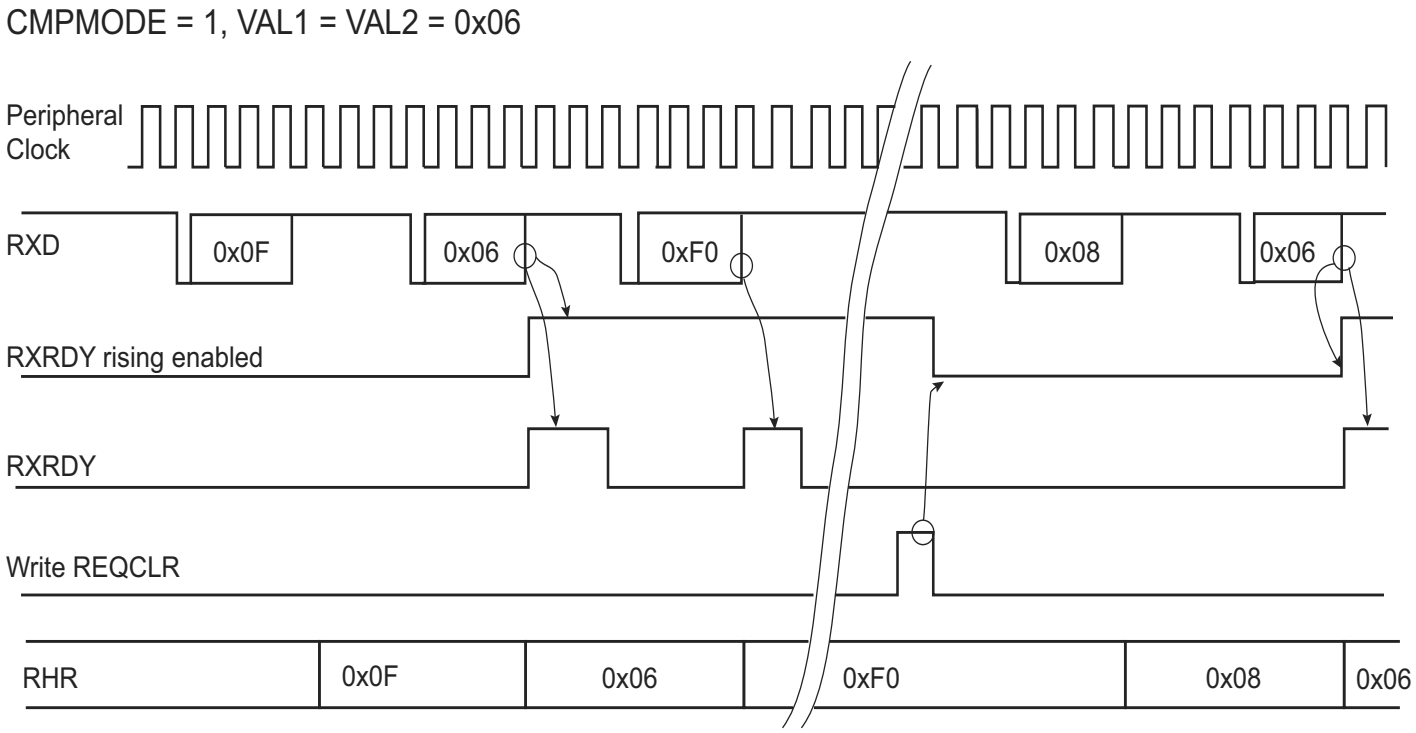
- Transmit FIFO:

The Transmit FIFO threshold can be set using the TXFTHRES field in TWIHS_FMR. Each time the Transmit FIFO goes from the ‘above threshold’ to the ‘equal or below threshold’ state, the TXFTHF flag in TWIHS_FSR is set. This enables the application to know that the Transmit FIFO reached the defined threshold and to refill it before it becomes empty.

- Receive FIFO:

The Receive FIFO threshold can be set using the RXFTHRES field in TWIHS_FMR. Each time the Receive FIFO goes from the ‘below threshold’ to the ‘equal or above threshold’ state, the RXFTHF flag in TWIHS_FSR is set. This enables the application to know that the Receive FIFO reached the defined threshold and to read some data before it becomes full.

Figure 47-38: Receive Holding Register Management



47.7.8 SPI Mode

The Serial Peripheral Interface (SPI) mode is a synchronous serial data link that provides communication with external devices in Master or Slave mode. It also enables communication between processors if an external processor is connected to the system.

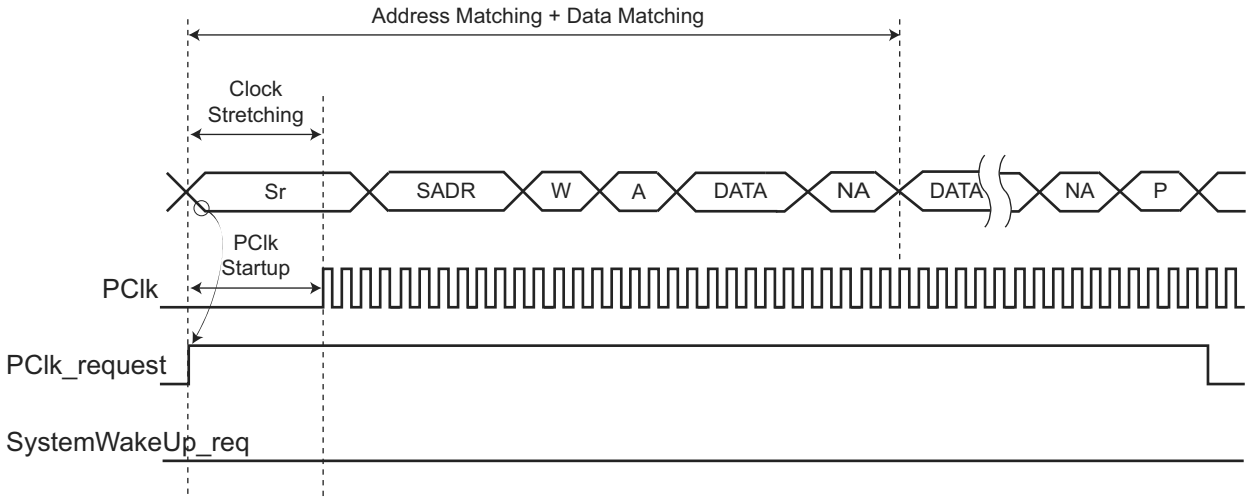
The Serial Peripheral Interface is essentially a shift register that serially transmits data bits to other SPIs. During a data transfer, one SPI system acts as the “master” which controls the data flow, while the other devices act as “slaves'' which have data shifted into and out by the master. Different CPUs can take turns being masters and one master may simultaneously shift data into multiple slaves. (Multiple master protocol is the opposite of single master protocol, where one CPU is always the master while all of the others are always slaves.) However, only one slave may drive its output to write data back to the master at any given time.

A slave device is selected when its NSS signal is asserted by the master. The USART in SPI Master mode can address only one SPI slave because it can generate only one NSS signal.

The SPI system consists of two data lines and two control lines:

- Master Out Slave In (MOSI): This data line supplies the output data from the master shifted into the input of the slave.
- Master In Slave Out (MISO): This data line supplies the output data from a slave to the input of the master.
- Serial Clock (SCK): This control line is driven by the master and regulates the flow of the data bits. The master may transmit data at a variety of bit rates. The SCK line cycles once for each bit that is transmitted.
- Slave Select (NSS): This control line allows the master to select or deselect the slave.

Figure 47-124: Address Matches and Data Do Not Match (Data Matching Enabled)



47.10.49 SPI Transmit Data Register (FIFO Multiple Data, 8- to 16-bit)

Name: FLEX_SPI_TDR (FIFO_MULTI_DATA)

Address: 0xF803440C (0), 0xF803840C (1), 0xFC01040C (2), 0xFC01440C (3), 0xFC01840C (4)

Access: Write-only

31	30	29	28	27	26	25	24
TD1							
23	22	21	20	19	18	17	16
TD1							
15	14	13	12	11	10	9	8
TD0							
7	6	5	4	3	2	1	0
TD0							

Note: If FIFO is enabled (FLEX_SPI_CR.FIFOEN) and FLEX_SPI_FMR.TXRDYM > 0, see Section 47.8.7.7 “SPI Multiple Data Mode” for details.

TDx: Transmit Data

Next data to write in the Transmit FIFO. Information to be transmitted must be written to this register in a right-justified format.

51.13.55 SDMMC Retuning Status Slots Register

Name: SDMMC_RTSSR

Access: Read-only

7	6	5	4	3	2	1	0
–	–	–	–	–	–	TEVTSLOT	

TEVTSLOT: Retuning Timer Event Slots

These status bits indicate the TEVT status for each SDMMC instance in the product (TEVTSLOT[x] corresponds to SDMMCx instance in the product).

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by MCAN_RXFnS.RFnL = ‘1’. In addition, the interrupt flag MCAN_IR.RFnL is set.

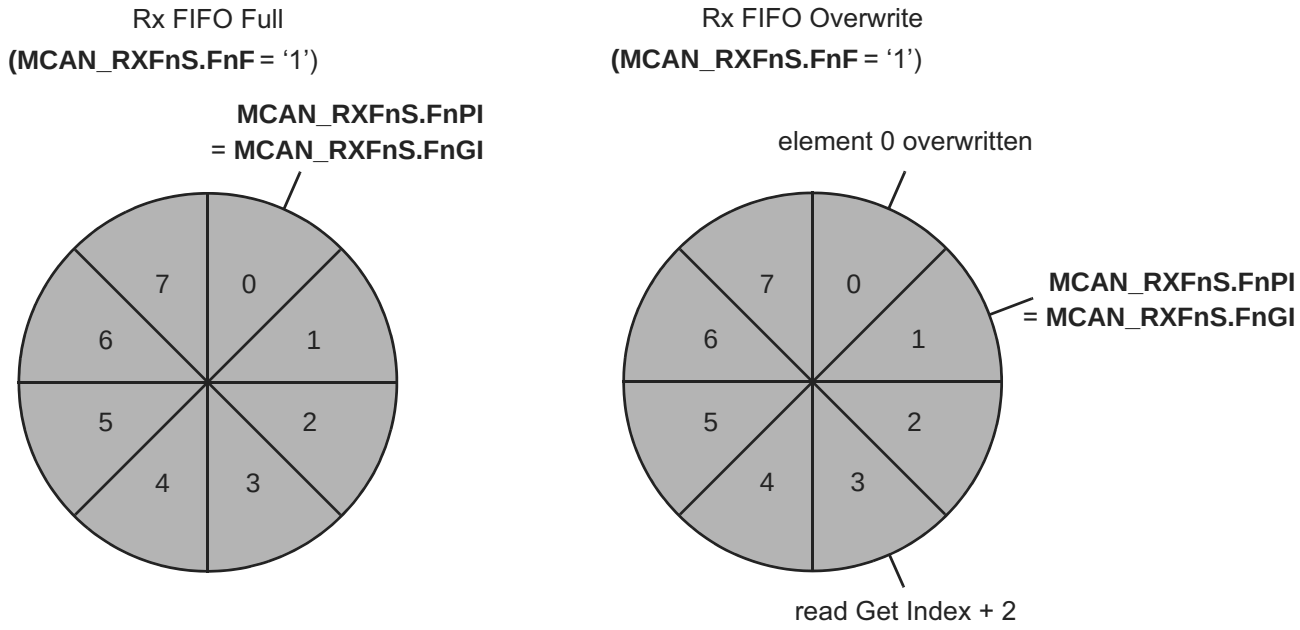
- Rx FIFO Overwrite Mode

The Rx FIFO Overwrite mode is configured by MCAN_RXFnC.FnOM = ‘1’.

When an Rx FIFO full condition (MCAN_RXFnS.FnPI = MCAN_RXFnS.FnGI) is signalled by MCAN_RXFnS.FnF = ‘1’, the next message accepted for the FIFO will overwrite the oldest FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in Overwrite mode and an Rx FIFO full condition is signalled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (put index) while the processor is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the processor accesses the Rx FIFO. Figure 53-8 shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 53-8: Rx FIFO Overflow Handling



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index MCAN_RXFnA.FnA. This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (MCAN_RXFnS.FnF = ‘0’).

53.5.4.3 Dedicated Rx Buffers

The MCAN supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via MCAN_RXBC.RBSA.

For each Rx Buffer, a Standard or Extended Message ID Filter Element with SFEC / EFEC = 7 and SFID2 / EFID2[10:9] = 0 has to be configured (see Section 53.5.7.5 and Section 53.5.7.6).

55.7.10 PDMIC Write Protection Mode Register

Name: PDMIC_WPMR

Address: 0xF80180E4

Access: Read/Write

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

WPEN: Write Protection Enable

0: Disables the write protection if WPKEY corresponds to 0x414443 (“ADC” in ASCII).

1: Enables the write protection if WPKEY corresponds to 0x414443 (“ADC” in ASCII).

See Section 55.6.4 “Register Write Protection” for the list of registers that can be write-protected.

WPKEY: Write Protection Key

Value	Name	Description
0x414443	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

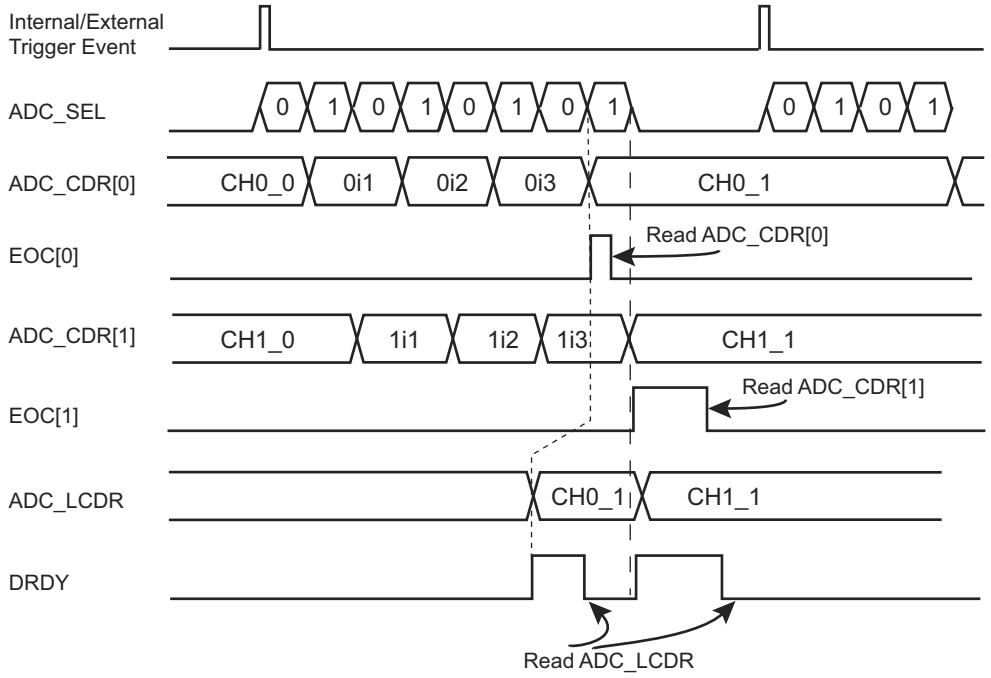
57.5 Secure Fuse Controller (SFC) User Interface

Table 57-1: Register Mapping

Offset	Register	Name	Access	Reset
0x00	SFC Key Register	SFC_KR	Write-only	–
0x04	SFC Mode Register	SFC_MR	Read/Write	0x0
0x08–0x0C	Reserved	–	–	–
0x10	SFC Interrupt Enable Register	SFC_IER	Write-only	–
0x14	SFC Interrupt Disable Register	SFC_IDR	Write-only	–
0x18	SFC Interrupt Mask Register	SFC_IMR	Read-only	0x0
0x1C	SFC Status Register	SFC_SR	Read-only	0x0
0x20	SFC Data Register 0	SFC_DR0	Read/Write	0x0
0x24	SFC Data Register 1	SFC_DR1	Read/Write	0x0
...	...	...	...	...
0x7C	SFC Data Register 23	SFC_DR23	Read/Write	0x0
0x80–0xFC	Reserved	–	–	–

Figure 65-12: Digital Averaging Function Waveforms on a Single Trigger Event

ADC_EMR.OSR = 1, ASTE = 1, ADC_CHSR[1:0] = 0x3 and ADC_MR.USEQ = 0



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

When USEQ = 1, the user can define the channel sequence to be converted by configuring ADC_SEQRx and ADC_CHER so that channels are not interleaved during the averaging period. Under these conditions, a sample is defined for each end of conversion as described in Figure 65-13.

When USEQ = 1 and ASTE = 1, OSR can be only configured to 1. Up to three channels can be converted in this mode. The averaging result will be placed in the corresponding ADC_CDRx and in ADC_LCDR for each trigger event. The ADC real sample rate remains the maximum ADC sample rate divided by 4.

It is important that the user sequence follows a specific pattern. The user sequence must be programmed in such a way that it generates a stream of conversion, where a same channel is successively converted.

Table 65-6: Example Sequence Configurations (USEQ = 1, ASTE = 1, OSR = 1)

	Number of Channels Non-interleaved Averaging - Register Value		
Register	**1 (e.g., CH0)**	**2 (e.g., CH0, CH1)**	**3 (e.g., CH0, CH1, CH2)**
ADC_CHSR	0x0000_000F	0x0000_00FF	0x0000_0FFF
ADC_SEQR1	0x0000_0000	0x1111_0000	0x1111_0000
ADC_SEQR2	0x0000_0000	0x0000_0000	0x0000_2222

where

- Tracking time expressed in ns and Z_{SOURCE} expressed in Ω
- n = 8 for 12-bit accuracy
- R_{ON} = 2 kΩ

Table 66-38: Number of Tau:n

Resolution (bits)	12
RES	0
n	8

The ADC already includes a tracking time of 15 $t_{ADC\ Clock}$.

66.12 Analog Comparator Characteristics

Table 66-39: Analog Comparator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDBU}	Power Supply Voltage Range (VDDBU)	Analog Comparator is supplied by VDDIN	1.62	3.3	3.6	V
V_{COMPx}	Input Voltage Range	On COMPP or COMPN input	0	–	VDDBU	V
V_{hys}	Hysteresis	–	35	–	70	mV
TPD	Propagation Delay	100mV Overdrive	–	–	350	µs
f_{in}	COMPx Input Signal Frequency	Common mode and differential	–	–	1	kHz
I_{VDDBU}	Current Consumption (VDDBU)	OFF Mode (ACC_MR.ACEN = 0)	–		50	nA
		ON Mode (ACC_MR.ACEN = 1)	–	100	200	
t_{START}	Startup Time	–	–	–	300	µs