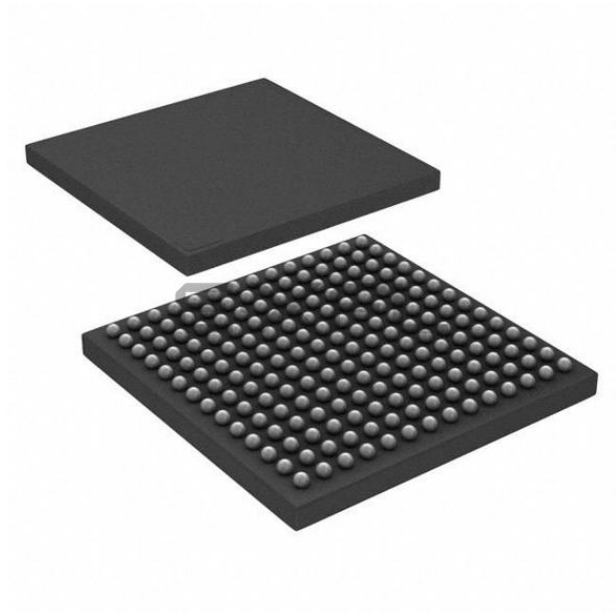


Welcome to E-XFL.COM

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	Obsolete
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	196-TFBGA, CSBGA
Supplier Device Package	196-TFBGA (11x11)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d22a-cur

7.3 Powerdown Considerations

Figure 7-2 shows the SAMA5D2 powerdown sequence that starts by asserting the NRST line to 0. Once NRST is asserted, the supply inputs can be immediately shutdown without any specific timing or order. VDDBU may not be shutdown if the application uses a backup battery on this supply input. In applications where VDDFUSE is powered, it is mandatory to shutdown VDDFUSE prior to removing any other supply. VDDFUSE can be removed before or after asserting the NRST signal.

Figure 7-2: Recommended Powerdown Sequence

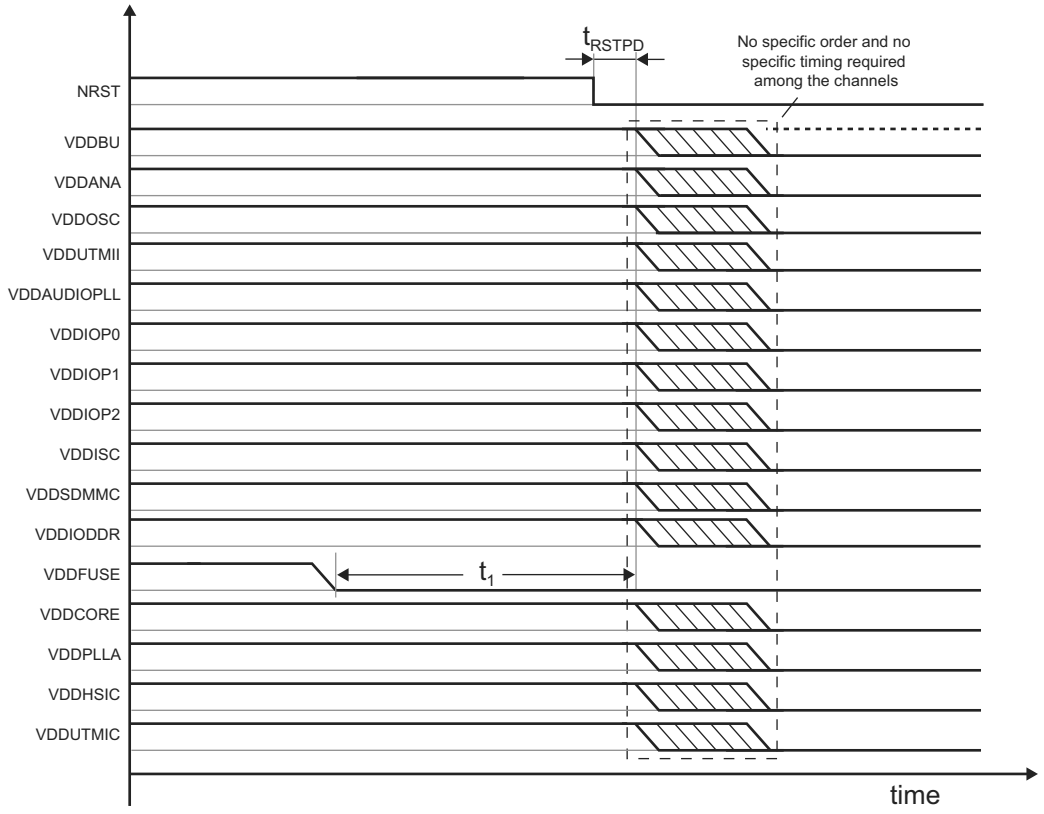


Table 7-3: Powerdown Timing Specification

Symbol	Parameter	Conditions	Min	Max	Unit
t_{RSTPD}	Reset delay at powerdown	From NRST low to the first supply turn-off	0	–	ms
t_1	VDDFUSE delay at shutdown	From VDDFUSE < 1V to the first supply turn-off	0	–	

7.4 Power Supply Sequencing at Backup Mode Entry and Exit

7.4.1 VDDBU Power Architecture

The backup power switch aims at optimizing the power consumption on VDDBU source by switching the supply of the backup digital part (BUREG memories + 64-kHz RC oscillator) to VDDANA.

When enabled, the backup power source can be automatically switched to VDDANA, which reduces power consumption on VDDBU. Then, VDDBU powers the pads, VDDBU POR, 32-kHz crystal and, on secure products SAMA5D23 and SAMA5D28, the temperature sensor and the backup supply monitor.

The power source (VDDANA or VDDBU) can be selected manually or can be set to work automatically by programming an SFRBU register (refer to SFRBU_PSWBUCTRL in Section 20. “Special Function Registers Backup (SFRBU)”).

34.7.13 PIO Write Protection Mode Register

Name: PIO_WPMR

Address: 0xFC0385E0

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 0x50494F (“PIO” in ASCII).

1: Enables the write protection if WPKEY corresponds to 0x50494F (“PIO” in ASCII).

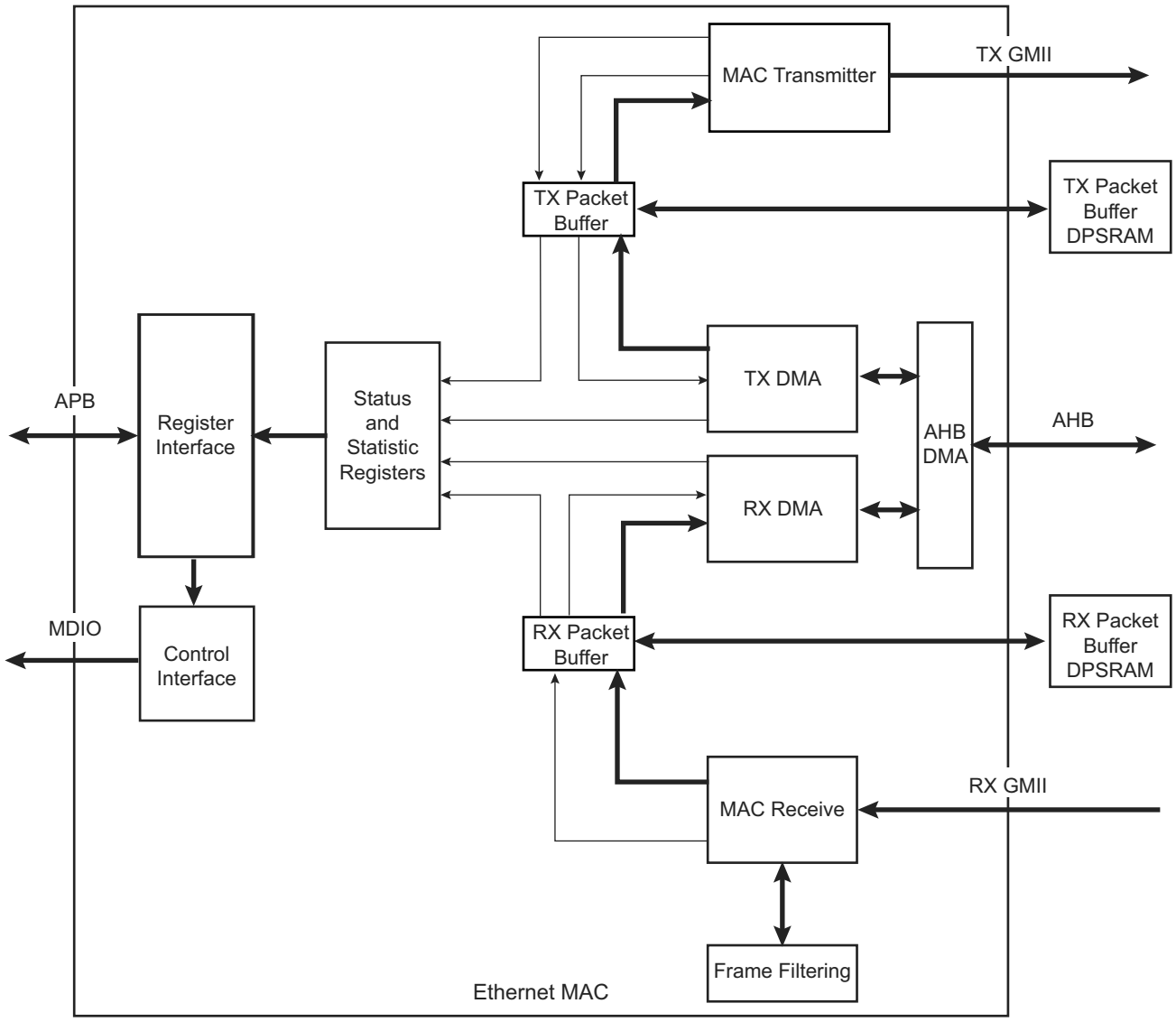
See Section 34.5.16 “Register Write Protection” for the list of registers that can be protected.

WPKEY: Write Protection Key

Value	Name	Description
0x50494F	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

When the PMECC engine is searching for roots, the BUSY field of the ELSR register remains asserted. An interrupt is asserted at the end of the computation, and the DONE bit of the PMECC Error Location Interrupt Status Register (HSMC_ELSIR) is set. The ERR_CNT field of the HSMC_ELISR indicates the number of errors. The error position can be read in the PMERRLOCX registers.

Figure 40-2: Data Paths with Packet Buffers Included



40.6.3.7 Transmit Packet Buffer

The transmitter packet buffer will continue attempting to fetch frame data from the AHB system memory until the packet buffer itself is full, at which point it will attempt to maintain its full level.

To accommodate the status and statistics associated with each frame, three words per packet (or two if the GMAC is configured in 64-bit datapath mode) are reserved at the end of the packet data. If the packet is bad and requires to be dropped, the status and statistics are the only information held on that packet. Storing the status in the DPRAM is required in order to decouple the DMA interface of the buffer from the MAC interface, to update the MAC status/statistics and to generate interrupts in the order in which the packets that they represent were fetched from the AHB memory.

If any errors occur on the AHB while reading the transmit frame, the fetching of packet data from AHB memory is halted. The MAC transmitter will continue to fetch packet data, thereby emptying the packet buffer and allowing any good non-errored frames to be transmitted successfully. Once these have been fully transmitted, the status/statistics for the errored frame will be updated and software will be informed via an interrupt that an AHB error occurred. This way, the error is reported in the correct packet order.

The transmit packet buffer will only attempt to read more frame data from the AHB when space is available in the packet buffer memory. If space is not available it must wait until the a packet fetched by the MAC completes transmission and is subsequently removed from the packet buffer memory. Note that if full store and forward mode is active and if a single frame is fetched that is too large for the packet buffer

- A priority enable vector taken from Transmit PFC Pause register
- 8 Pause Quantum registers
- Fill of 00 to take the frame to minimum frame length
- Valid FCS

The Pause Quantum registers used in the generated frame will depend on the trigger source for the frame as follows:

- If bit 17 of the Network Control register is written with a one, then the priority enable vector of the priority-based pause frame will be set equal to the value stored in the Transmit PFC Pause register [7:0]. For each entry equal to zero in the Transmit PFC Pause register [15:8], the pause quantum field of the pause frame associated with that entry will be taken from the Transmit Pause Quantum register. For each entry equal to one in the Transmit PFC Pause register [15:8], the pause quantum associated with that entry will be zero.
- The Transmit Pause Quantum register resets to a value of 0xFFFF giving maximum pause quantum as default.

After transmission, a pause frame transmitted interrupt will be generated (bit 14 of the Interrupt Status register) and the only statistics register that will be incremented will be the Pause Frames Transmitted register.

PFC Pause frames can also be transmitted by the MAC using normal frame transmission methods.

40.8.121 GMAC Interrupt Mask Register Priority Queue x

Name: GMAC_IMRPQx[x=1..2]

Address: 0xF8008640

Access: Read/Write

31	30	29	28	27	26	25	24
–							

23	22	21	20	19	18	17	16
–							

15	14	13	12	11	10	9	8
–				HRESP	ROVR	–	–

7	6	5	4	3	2	1	0
TCOMP	AHB	RLEX	–	–	RXUBR	RCOMP	–

A read of this register returns the value of the receive complete interrupt mask.

A write to this register directly affects the state of the corresponding bit in the Interrupt Status Register, causing an interrupt to be generated if a 1 is written.

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

0: Corresponding interrupt is enabled.

1: Corresponding interrupt is disabled.

RCOMP: Receive Complete

RXUBR: RX Used Bit Read

RLEX: Retry Limit Exceeded or Late Collision

AHB: AHB Error

TCOMP: Transmit Complete

ROVR: Receive Overrun

HRESP: HRESP Not OK

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.

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.

The TXFTHF and RXFTHF flags can be both configured to generate an interrupt using TWIHS_FIER and TWIHS_FIDR.

- FIFO Flags

FIFOs come with a set of flags which can be configured each to generate an interrupt through TWIHS_FIER and TWIHS_FIDR.

FIFO flags state can be read in TWIHS_FSR. They are cleared on TWIHS_FSR read.

46.6.6 TWIHS Comparison Function on Received Character

The comparison function differs if asynchronous partial wakeup (SleepWalking) is enabled or not.

If asynchronous partial wakeup is disabled (see the section “Power Management Controller (PMC)”), the TWIHS can extend the address matching on up to three slave addresses. The SADR1EN, SADR2EN and SADR3EN bits in TWIHS_SMR enable address matching on additional addresses which can be configured through SADR1, SADR2 and SADR3 fields in the TWIHS_SWMR. The DATAMEN bit in the TWIHS_SMR has no effect.

The SVACC bit is set when there is a comparison match with the received slave address.

In Wait mode, if asynchronous partial wakeup is enabled, a system wakeup is performed (see Section 47.8.6 ”SPI Asynchronous and Partial Wakeup (SleepWalking)”).

In Active mode, the CMP flag in FLEX_SPI_SR is raised. It is set when the received character matches the conditions programmed in the SPI Comparison Register (FLEX_SPI_CMPR). The CMP flag is set as soon as FLEX_SPI_RDR is loaded with the new received character. The CMP flag is cleared by reading FLEX_SPI_SR.

FLEX_SPI_CMPR (see **Section 47.10.57 “SPI Comparison Register”**) can be programmed to provide different comparison methods. These are listed below:

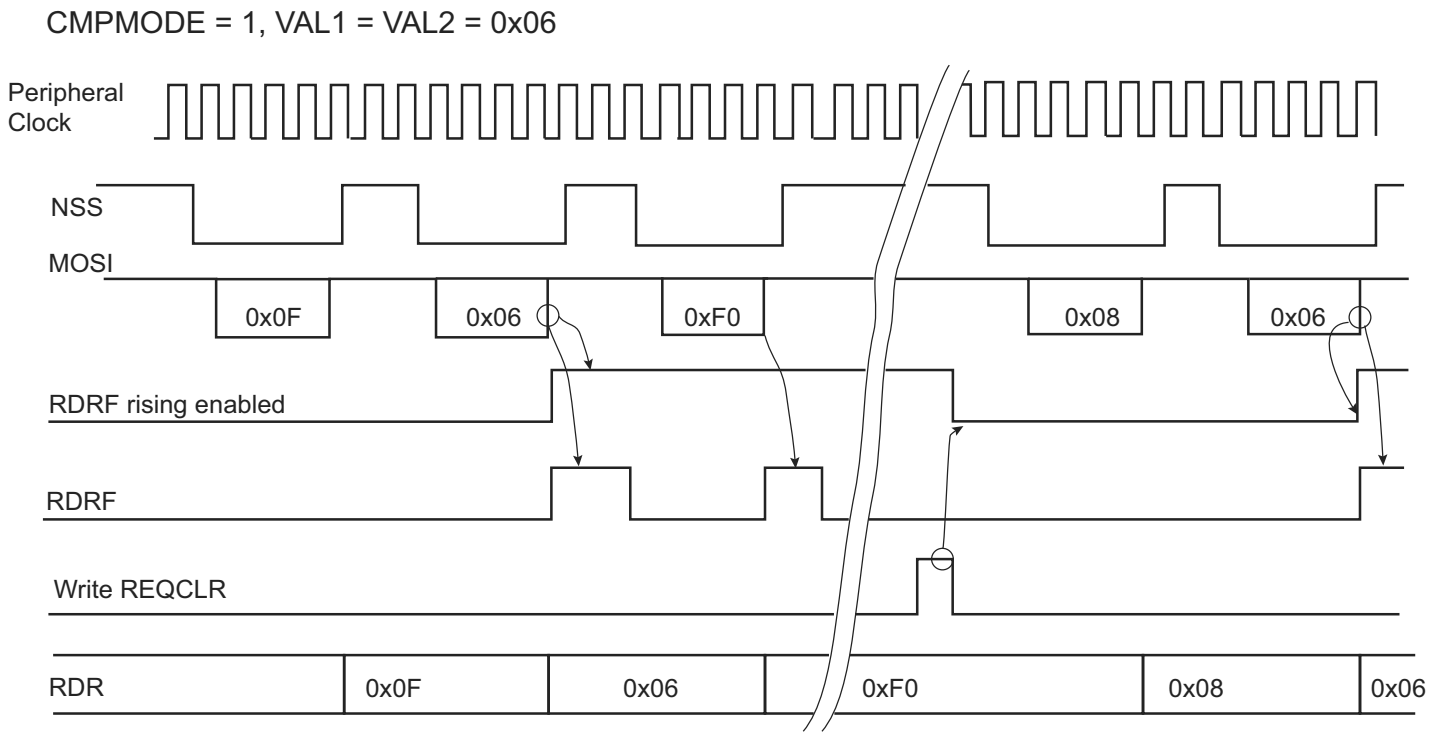
- If VAL1 equals VAL2, then the comparison is performed on a single value and the flag is set to 1 if the received character equals VAL1.
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 sets the CMP flag.
- If VAL1 is strictly higher than VAL2, then the flag CMP is set to 1 if any received character equals VAL1 or VAL2.

When FLEX_SPI_MR.CMPMODE is cleared, all received data is loaded in FLEX_SPI_RDR and the CMP flag provides the status of the comparison result.

By setting the CMPMODE bit, the comparison result triggers the start of FLEX_SPI_RDR loading (see Figure 47-75). The trigger condition exists as soon as the received character value matches the conditions defined by VAL1 and VAL2 in FLEX_SPI_CMPR. The comparison trigger event is restarted by writing a 1 to the FLEX_SPI_CR.REQCLR bit.

The value programmed in VAL1 and VAL2 fields must not exceed the maximum value of the received character (see BITS field in SPI Chip Select Register (FLEX_SPI_CSR)).

Figure 47-75: Receive Data Register Management



47.8.6 SPI Asynchronous and Partial Wakeup (SleepWalking)

This operating mode is a means of data preprocessing that qualifies an incoming event, thus allowing the SPI to decide whether or not to wake up the system. Asynchronous and partial wakeup is mainly used when the system is in Wait mode (refer to Section 33. “Power Management Controller (PMC)” for further details). It can also be enabled when the system is fully running.

Asynchronous and partial wakeup can be used only when SPI is configured in Slave mode (FLEX_SPI_MR.MSTR is cleared).

The maximum SPI clock (SPCK) frequency that can be provided by the SPI master is bounded by the peripheral clock frequency. The SPCK frequency must be lower than or equal to the peripheral clock. The NSS line must be deasserted by the SPI master between two characters. The NSS deassertion duration time must be greater than or equal to six peripheral clock periods. The time between the assertion of NSS line (falling edge) and the first edge of the SPI clock must be higher than 15 µs.

49.8.3 SPI Receive Data Register

Name: SPI_RDR

Address: 0xF8000008 (0), 0xFC000008 (1)

Access: Read-only

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

RD: Receive Data

Data received by the SPI Interface is stored in this register in a right-justified format. Unused bits are read as zero.

PCS: Peripheral Chip Select

In Master mode only, these bits indicate the value on the NPCS pins at the end of a transfer. Otherwise, these bits are read as zero.

Note: When using Variable Peripheral Select mode (PS = 1 in SPI_MR), it is mandatory to set SPI_MR.WDRBT bit if the PCS field must be processed in SPI_RDR.

49.8.16 SPI Write Protection Mode Register

Name: SPI_WPMR

Address: 0xF80000E4 (0), 0xFC0000E4 (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 0x535049 (“SPI” in ASCII)

1: Enables the write protection if WPKEY corresponds to 0x535049 (“SPI” in ASCII)

WPKEY: Write Protection Key

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

See Section 49.7.8 “Register Write Protection” for the list of registers that can be write-protected.

51.13.11 SDMMC Host Control 1 Register (e.MMC)

Name: SDMMC_HC1R (e.MMC)

Access: Read/Write

7	6	5	4	3	2	1	0
–	–	EXTDW	DMASEL		HSEN	DW	–

DW: Data Width

This bit selects the data width of the SDMMC. It must be set to match the data width of the card.

0 (1_BIT): 1-bit mode.

1 (4_BIT): 4-bit mode.

Note: If the Extended Data Transfer Width is 1, this bit has no effect and the data width is 8-bit mode.

HSEN: High Speed Enable

Before setting this bit, the user must check High Speed Support (HSSUP) in SDMMC_CA0R.

If this bit is set to 0 (default), the SDMMC outputs CMD line and DAT lines at the falling edge of the SD clock (up to 25 MHz). If this bit is set to 1, the SDMMC outputs the CMD line and the DAT lines at the rising edge of the SD clock (up to 50 MHz).

If Preset Value Enable (PVALEN) in SDMMC_HC2R is set to 1, the user needs to reset the SD Clock Enable (SDCLKEN) before changing this bit to avoid generating clock glitches. After setting this bit to 1, the user sets SDCLEN to 1 again.

0: Normal Speed mode.

1: High Speed mode.

Note 1: This bit is effective only if SDMMC_MC1R.DDR is set to 0.

2: The clock divider (DIV) in SDMMC_CCR must be set to a value different from 0 when HSEN is 1.

DMASEL: DMA Select

One of the supported DAM modes can be selected. The user must check support of DMA modes by referring the SDMMC_CA0R. Use of selected DMA is determined by DMA Enable (DMAEN) in SDMMC_TMR.

Value	Name	Description
0	SDMA	SDMA is selected
1	–	Reserved
2	ADMA32	32-bit Address ADMA2 is selected
3	–	Reserved

EXTDW: Extended Data Width

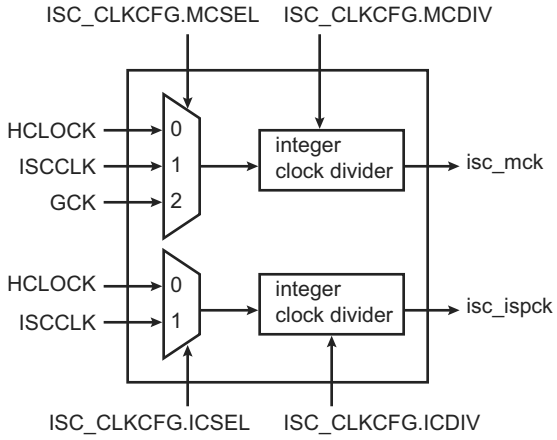
This bit controls the 8-bit Bus Width mode for embedded devices. Support of this function is indicated in 8-bit Support for Embedded Device in SDMMC_CA0R. If a device supports the 8-bit mode, this may be set to 1. If this bit is 0, the bus width is controlled by Data Width (DW).

52.5 Functional Description

52.5.1 ISC Clock Management

The ISC module provides the isc_mck output clock to the image sensor. The isc_mck clock has three selectable clock sources (ISC_CLKCFG.MCSEL field) and one programmable clock divider (ISC_CLKCFG.MCDIV). The clock is enabled using the ISC_CLKEN.MCEN field. The isc_mck is driven by the ISC and is the external reference clock of the CMOS sensor.

Figure 52-9: Clock Divider Block Diagram



The ISC digital pipeline requires internally a functional clock named isc_ispck. This clock is also fully programmable. This isc_ispck is enabled using the ISC_CLKEN.ICEN field. This clock is mandatory for ISC operation. The ISC module is designed to accept input signals that are asynchronous to the isc_ispck. Synchronization is done internally as long as the following relationship holds:

isc_pck frequency is less than or equal to isc_ispck, and isc_ispck is greater than or equal to HCLOCK.

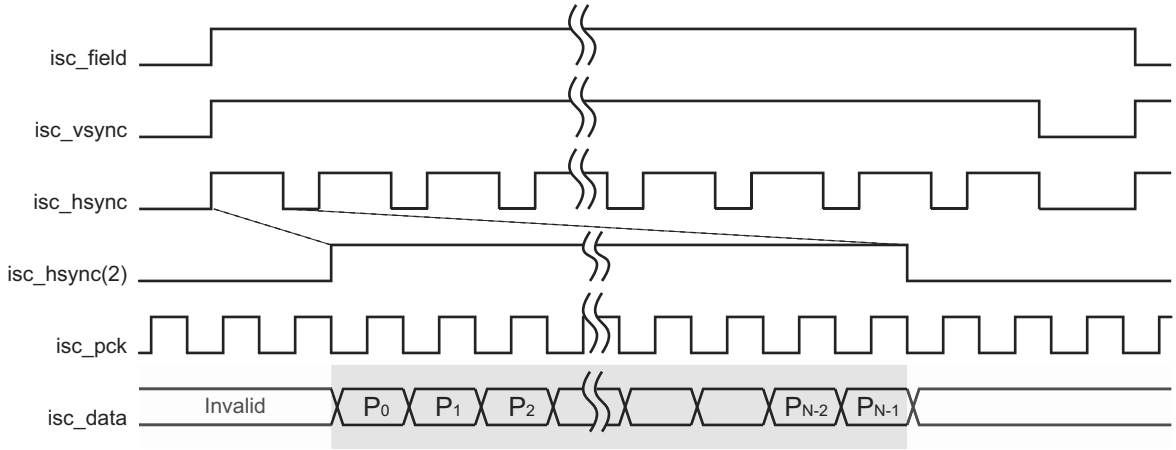
52.5.1.1 Software Requirement

A software write operation to the ISC_CLKEN or ISC_CLKDIS register requires double clock domain synchronization and is not permitted when the ISC_CLKSR.SIP is asserted.

52.5.2 Parallel Interface Timing Description

The parallel interface protocol supports two operating modes.

Figure 52-10: Free-Running Pixel Clock



TC: Transmission Completed

0: No transmission completed.

1: Transmission completed.

TCF: Transmission Cancellation Finished

0: No transmission cancellation finished.

1: Transmission cancellation finished.

TFE: Tx FIFO Empty

0: Tx FIFO non-empty.

1: Tx FIFO empty.

TEFN: Tx Event FIFO New Entry

0: Tx Event FIFO unchanged.

1: Tx Handler wrote Tx Event FIFO element.

TEFW: Tx Event FIFO Watermark Reached

0: Tx Event FIFO fill level below watermark.

1: Tx Event FIFO fill level reached watermark.

TEFF: Tx Event FIFO Full

0: Tx Event FIFO not full.

1: Tx Event FIFO full.

TEFL: Tx Event FIFO Element Lost

0: No Tx Event FIFO element lost.

1: Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

TSW: Timestamp Wraparound

0: No timestamp counter wrap-around.

1: Timestamp counter wrapped around.

MRAF: Message RAM Access Failure

The flag is set, when the Rx Handler

- has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message.
- was not able to write a message to the Message RAM. In this case message storage is aborted.

In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Receive Buffer is not set, a partly stored message is overwritten when the next message is stored to this location.

The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the MCAN is switched into Restricted Operation mode (see Section 53.5.1.5). To leave Restricted Operation mode, the processor has to reset MCAN_CCCR.ASM.

0: No Message RAM access failure occurred.

1: Message RAM access failure occurred.

TOO: Timeout Occurred

0: No timeout.

1: Timeout reached.

For ICM_UIHVAL4 field:

Example	Comment
0xC3D2E1F0	SHA1 algorithm
0xFFC00B31	SHA224 algorithm
0x510E527F	SHA256 algorithm

For ICM_UIHVAL5 field:

Example	Comment
0x68581511	SHA224 algorithm
0x9B05688C	SHA256 algorithm

For ICM_UIHVAL6 field:

Example	Comment
0x64F98FA7	SHA224 algorithm
0x1F83D9AB	SHA256 algorithm

For ICM_UIHVAL7 field:

Example	Comment
0xBEFA4FA4	SHA224 algorithm
0x5BE0CD19	SHA256 algorithm

Example of Initial Value for SHA-1 Algorithm

	Address Offset / Byte Lane			
Register Address	**0x3 / 31:24**	**0x2 / 23:16**	**0x1 / 15:8**	**0x0 / 7:0**
0x000 ICM_UIHVAL0	01	23	45	67
0x004 ICM_UIHVAL1	89	ab	cd	ef
0x008 ICM_UIHVAL2	fe	dc	ba	98
0x00C ICM_UIHVAL3	76	54	32	10
0x010 ICM_UIHVAL4	f0	e1	d2	c3

62.5.1 TDES Control Register

Name: TDES_CR

Address: 0xFC044000

Access: Write-only

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

- **START: Start Processing**

0: No effect

1: Starts Manual encryption/decryption process.

SWRST: Software Reset

0: No effect

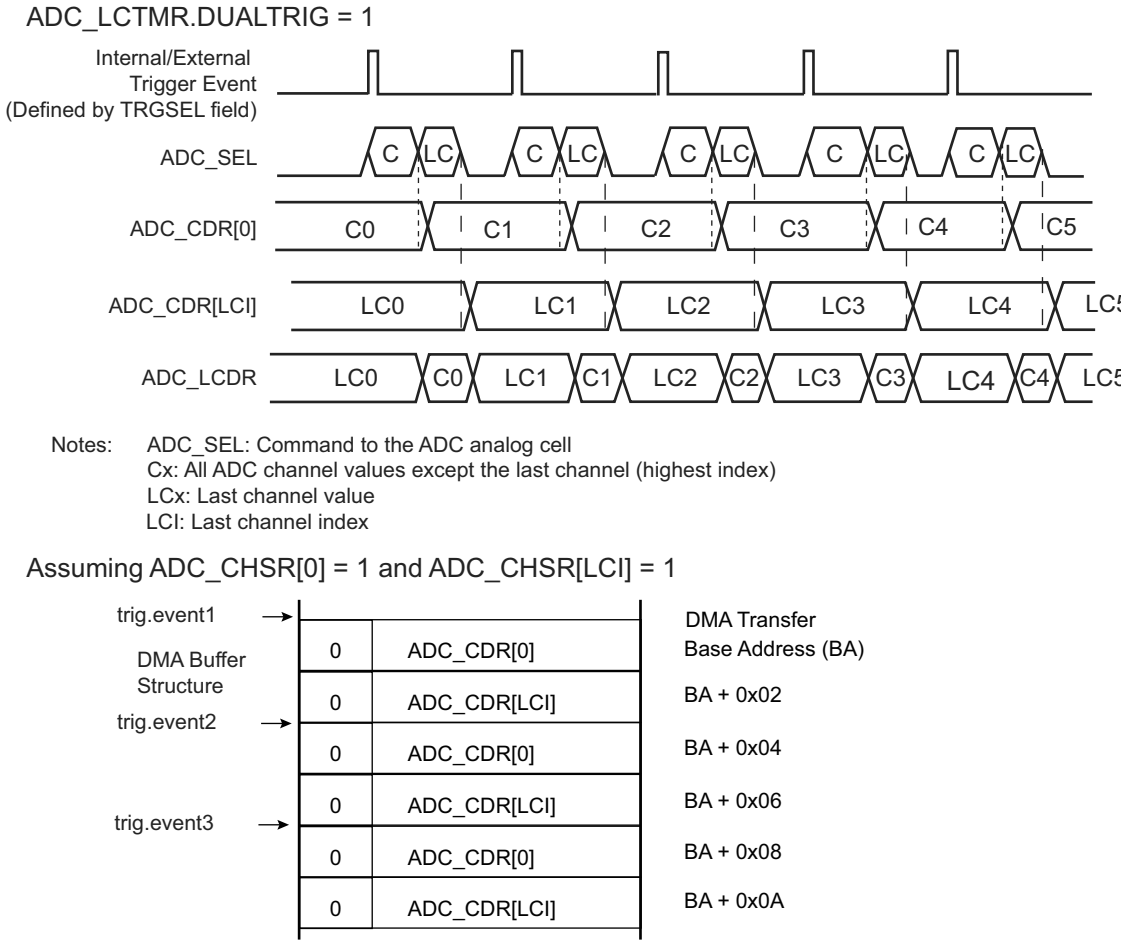
1: Resets the TDES. A software triggered hardware reset of the TDES interface is performed.

The manual start can only be performed if field TRGMOD = 0. When the START bit in ADC_CR is set, the last channel conversion is scheduled together with the other enabled channels (if any). The result of the conversion is placed in ADC_CDR11 register and the associated flag EOC11 is set in ADC_ISR.

If the last channel is enabled in ADC_CHSR, DUALTRIG is cleared and field TRGMOD = 1, 2, 3, 5, the last channel is periodically converted together with the other enabled channels and the result is placed in the ADC_LCDR and ADC_CDR11 registers. Thus the last channel conversion result is part of the DMA Controller buffer (see Figure 65-8).

When the conversion result matches the conditions defined in the ADC_LCTMR and ADC_LCCWR, the LCCHG flag is set in ADC_ISR.

Figure 65-8: Same Trigger for All Channels (ADC_CHSR[LCI] = 1 and ADC_TRGR.TRGMOD = 1, 2, 3, 5)



Notes: ADC_SEL: Command to the ADC analog cell
Cx: All ADC channel values except the last channel (highest index)
LCx: Last channel value
LCI: Last channel index

If the last channel is driven by a device with a slower variation compared to other channels (temperature sensor for example), the channel can be enabled/disabled at any time. However, this may not be optimal for downstream processing.

The ADC controller allows a different way of triggering the measurement when DUALTRIG is set in the Last Channel Trigger Mode Register (ADC_LCTMR) but CH11 is not set in ADC_CHSR.

Under these conditions, the last channel conversion is triggered with a period defined by the OUT1 field in the RTC_MR (Real-time Clock Mode Register) while other channels are still active. OUT1 configures an internal trigger generated by the RTC, totally independent of the internal/external triggers. The RTC event will be processed on the next internal/external trigger event as described in Figure 65-9. The internal/external trigger for other channels is selected through the TRGSEL field of ADC_MR.

When DUALTRIG = 1, the result of each conversion of channel 11 is only uploaded in the ADC_CDR11 register and not in ADC_LCDR (see Figure 65-9). Therefore, there is no change in the structure of the peripheral DMA controller buffer due to the conversion of the last channel: only the enabled channels are kept in the buffer. The end of conversion of the last channel is reported by the EOC11 flag in ADC_ISR.

65.7.16 ADC Extended Mode Register

Name: ADC_EMR

Address: 0xFC030040

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	ADCMODE		–	SIGNMODE		TAG
23	22	21	20	19	18	17	16
–	–	SRCCLK	ASTE	–	–	OSR	
15	14	13	12	11	10	9	8
–	–	CMPFILTER		–	–	CMPALL	–
7	6	5	4	3	2	1	0
CMPSEL				–	CMPTYPE	CMPMODE	

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

CMPMODE: Comparison Mode

Value	Name	Description
0	LOW	When the converted data is lower than the low threshold of the window, generates the COMPE flag in ADC_ISR or, in Partial Wakeup mode, defines the conditions to exit system from Wait mode.
1	HIGH	When the converted data is higher than the high threshold of the window, generates the COMPE flag in ADC_ISR or, in Partial Wakeup mode, defines the conditions to exit system from Wait mode.
2	IN	When the converted data is in the comparison window, generates the COMPE flag in ADC_ISR or, in Partial Wakeup mode, defines the conditions to exit system from Wait mode.
3	OUT	When the converted data is out of the comparison window, generates the COMPE flag in ADC_ISR or, in Partial Wakeup mode, defines the conditions to exit system from Wait mode.

CMPTYPE: Comparison Type

Value	Name	Description
0	FLAG_ONLY	Any conversion is performed and comparison function drives the COMPE flag.
1	START_CONDITIO N	Comparison conditions must be met to start the storage of all conversions until the CMPRST bit is set.

CMPSEL: Comparison Selected Channel

If CMPALL = 0: CMPSEL indicates which channel has to be compared.

If CMPALL = 1: No effect.

CMPALL: Compare All Channels

0: Only channel indicated in CMPSEL field is compared.

1: All channels are compared.

CMPFILTER: Compare Event Filtering

Number of consecutive compare events necessary to raise the flag = CMPFILTER+1

When programmed to 0, the flag rises as soon as an event occurs.

See Section 65.6.9 “Comparison Window” when using filtering option (CMPFILTER > 0).

Figure 66-36: SSC Receiver, RK and RF in Output

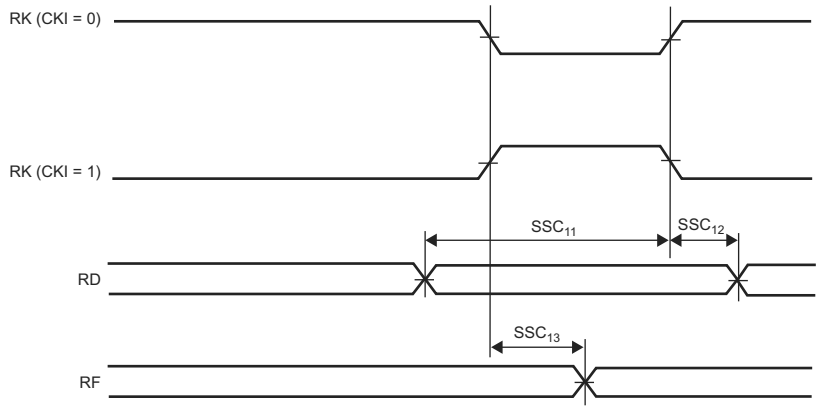


Figure 66-37: SSC Receiver, RK in Output and RF in Input

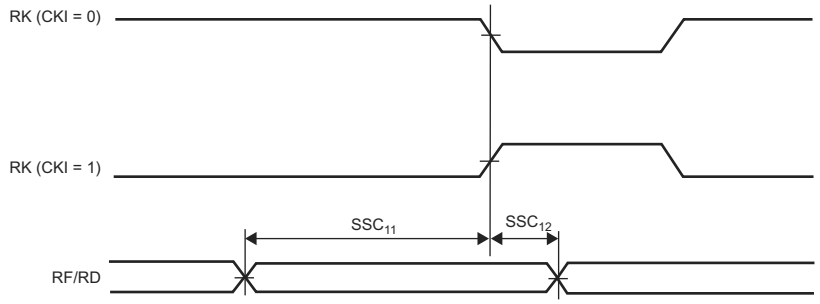


Table 66-82: SSC0 IOSET1 Timings

Symbol	Power supply Parameter	Conditions	1.8V Min	 Max	3.3V Min	 Max	Unit
Transmitter							
SSC_0	TK edge to TF/TD (TK output, TF output) (1)	–	0	3	0	3.3	ns
SSC_1	TK edge to TF/TD (TK input, TF output) (1)	–	3.7	13	3	11.3	ns
SSC_2	TF setup time before TK edge (TK output)	–	12.8	–	11.2	–	ns
SSC_3	TF hold time after TK edge (TK output)	–	0	–	0	–	ns
SSC_4	TK edge to TF/TD (TK output, TF input)(1)	–	0	3	0	3.3	ns
		STTDLY = 0 START = 4, 5 or 7	$2 \times t_{CPMCK}$	3 + ($2 \times t_{CPMCK}$)	$2 \times t_{CPMCK}$	3.3 + ($2 \times t_{CPMCK}$)	ns
SSC_5	TF setup time before TK edge (TK input)	–	0	–	0	–	
SSC_6	TF hold time after TK edge (TK input)	–	t_{CPMCK}	–	t_{CPMCK}	–	

71.2.19 MCAN High Priority Message (HPM)

Issue: Unexpected High Priority Message (HPM) interrupt

There are two configurations where the issue occurs:

Configuration A:

- At least one Standard Message ID Filter Element is configured with priority flag set (S0.SFEC = "100"/"101"/"110")
- No Extended Message ID Filter Element configured
- Non-matching extended frames are accepted (MCAN_GFC.ANFE = "00"/"01")

The HPM interrupt flag MCAN_IR.HPM is set erroneously on reception of a non-high-priority extended message under the following conditions:

1. A standard HPM frame is received, and accepted by a filter with priority flag set. Then, interrupt flag MCAN_IR.HPM is set as expected.
2. Next an extended frame is received and accepted because of MCAN_GFC.ANFE configuration. Then, interrupt flag MCAN_IR.HPM is set erroneously.

Configuration B:

- At least one Extended Message ID Filter Element is configured with priority flag set (F0.EFEC = "100"/"101"/"110")
- No Standard Message ID Filter Element configured
- Non-matching standard frames are accepted (MCAN_GFC.ANFS = "00"/"01")

The HPM interrupt flag MCAN_IR.HPM is set erroneously on reception of a non-high-priority standard message under the following conditions:

1. An extended HPM frame is received, and accepted by a filter with priority flag set. Then, interrupt flag MCAN_IR.HPM is set as expected.
2. Next a standard frame is received and accepted because of MCAN_GFC.ANFS configuration. Then, interrupt flag MCAN_IR.HPM is set erroneously.

Workaround:

Configuration A:

Setup an Extended Message ID Filter Element with the following configuration:

- F0.EFEC = "001"/"010" - select Rx FIFO for storage of extended frames
- F0.EFID1 = any value - value not relevant as all ID bits are masked out by F1.EFID2
- F1.EFT = "10" - classic filter, F0.EFID1 = filter, F1.EFID2 = mask
- F1.EFID2 = zero - all bits of the received extended ID are masked out

Now all extended frames are stored in Rx FIFO 0 respectively Rx FIFO 1 depending on the configuration of F0.EFEC.

Configuration B:

Setup a Standard Message ID Filter Element with the following configuration:

- S0.SFEC = "001"/"010" - select Rx FIFO for storage of standard frames
- S0.SFID1 = any value - value not relevant as all ID bits are masked out by S0.SFID2
- S0.SFT = "10" - classic filter, S0.SFID1 = filter, S0.SFID2 = mask
- S0.SFID2 = zero - all bits of the received standard ID are masked out

Now all standard frames are stored in Rx FIFO 0 respectively Rx FIFO 1 depending on the configuration of S0.SFEC.

71.2.20 ROM Code: Using JTAG IOSET 4

Issue: JTAG_TCK on IOSET 4 pin has a wrong configuration after boot

The JTAG_TCK signal on IOSET 4 shares its pin (PA22) with the clock signal of the following boot memory interfaces: SDMMC1, SPI1 IOSET 2, QSPI 0 IOSET 3.