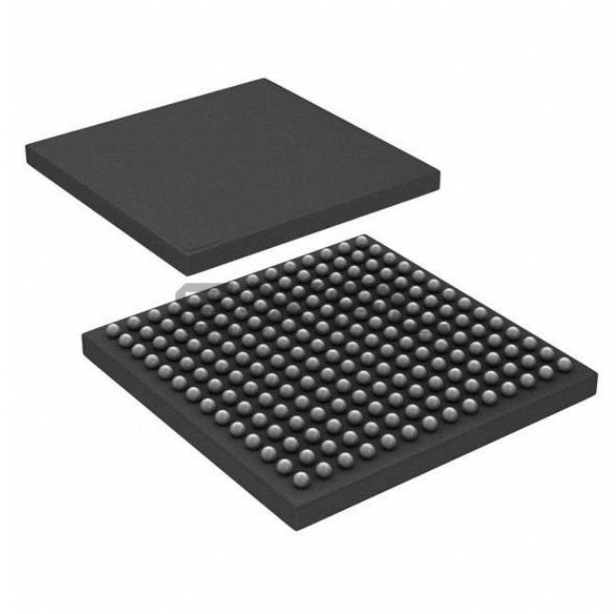


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

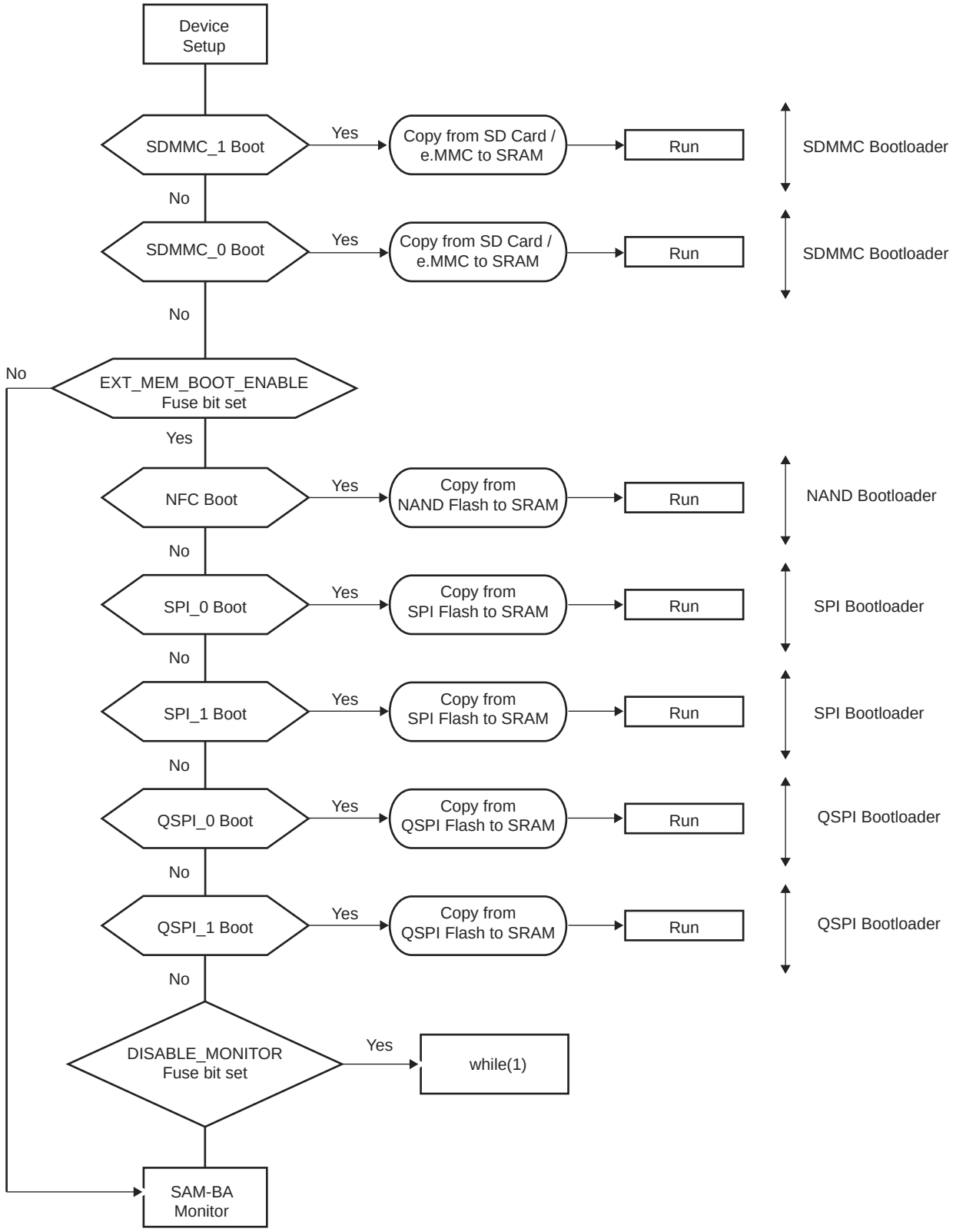
Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, 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

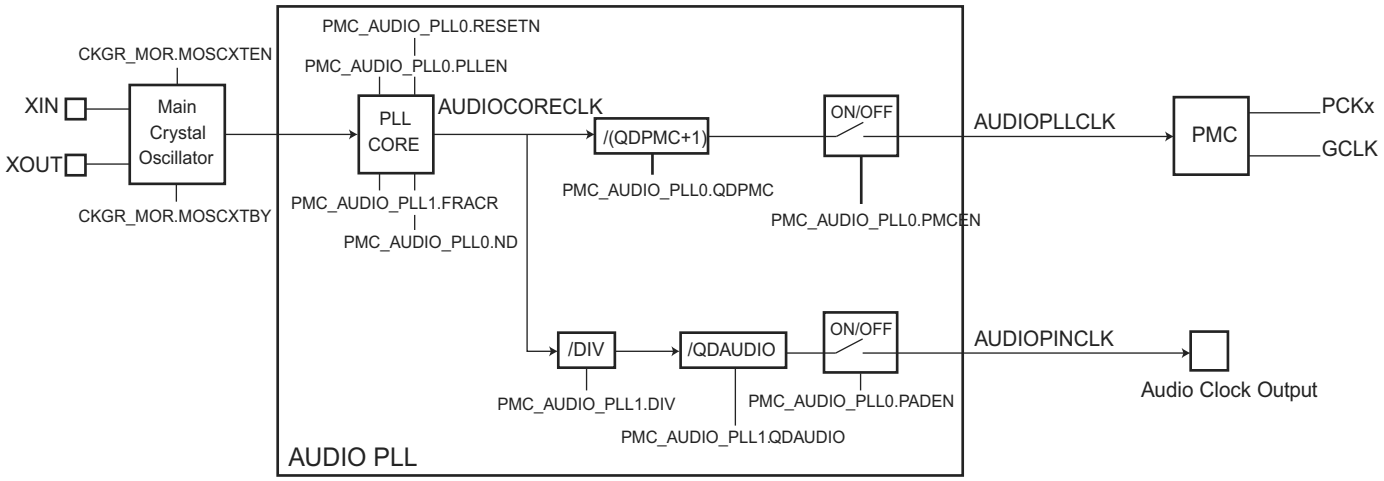
Details	
Product Status	Obsolete
Core Processor	ARM® Cortex®-A5
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	500MHz
Co-Processors/DSP	Multimedia; NEON™ MPE
RAM Controllers	LPDDR1, LPDDR2, LPDDR3, DDR2, DDR3, DDR3L, QSPI
Graphics Acceleration	Yes
Display & Interface Controllers	Keyboard, LCD, Touchscreen
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/atsama5d22b-cur

Figure 16-4: NVM Bootloader Program Description for MRL C Parts



6. If needed, ND or FRACR can be adjusted at any time. The typical frequency settling time of this PLL is indicated in Section 66. “Electrical Characteristics”.

Figure 32-7: Audio PLL



34.5.12 I/O Lines Lock

When an I/O line is controlled by a peripheral (particularly the Pulse Width Modulation Controller PWM), it can become locked by the action of this peripheral via an input of the PIO Controller. When an I/O line is locked, the following fields in PIO_CFGRx are locked and cannot be modified:

- FUNC: Peripheral selection cannot be changed when the corresponding I/O line is locked.
- PUEN: Pull-Up configuration cannot be changed when the corresponding I/O line is locked.
- PDEN: Pull-Down configuration cannot be changed when the corresponding I/O line is locked.
- OPD: Open Drain configuration cannot be changed when the corresponding I/O line is locked.

Writing to one of these fields while the corresponding I/O line is locked will have no effect.

The user can know at anytime which I/O line is locked by reading the PIO Lock Status Register (PIO_LOCKSR) or Secure PIO Lock Status Register (S_PIO_LOCKSR) for locked Secure I/O lines. Once an I/O line is locked, the only way to unlock it is to apply a hardware reset to the PIO Controller.

34.5.13 Programmable I/O Drive

It is possible to configure the I/O drive for pads PA0 to PD31. The I/O drive of the pad can be programmed by writing the DRVSTR field in the PIO Configuration Register (PIO_CFGRx) if the corresponding line is Non-Secure or the Secure PIO Configuration Register (S_PIO_CFGRx) if the I/O line is Secure. For details, refer to Section 66. “Electrical Characteristics”.

34.5.14 Programmable Schmitt Trigger

It is possible to configure each input for the Schmitt trigger. The Schmitt trigger can be enabled by setting the SCHMITT bit of the PIO Configuration Register (PIO_CFGRx) if the corresponding line is Non-Secure or the Secure PIO Configuration Register (S_PIO_CFGRx) if the I/O line is Secure. By default the Schmitt trigger is active. Disabling the Schmitt trigger is requested when using the QTouch® Library.

34.5.15 I/O Line Configuration Freeze

34.5.15.1 Software Freeze

Once the I/O line configuration is done, it can be frozen by using the PIO I/O Freeze Configuration Register (PIO_IOFRx) of the corresponding group or the Secure PIO I/O Freeze Configuration Register (S_PIO_IOFRx) if the I/O line is Secure.

- Physical Freeze

Setting the FPHY bit of PIO_IOFRx freezes the following fields of the Non-Secure I/O lines defined in PIO_MSKRx:

- FUNC: I/O Line Function
- DIR: Direction
- PUEN: Pull-Up Enable
- PDEN: Pull-Down Enable
- OPD: Open-Drain
- SCHMITT: Schmitt Trigger
- DRVSTR: Drive Strength

For Secure I/O lines, use the FPHY bit of the S_PIO_IOFRx and the S_PIO_MSKRx to freeze the fields above.

When the physical freeze is currently active on an I/O line, the PCFS flag is set when reading the PIO_CFGRx of the I/O line (or the S_PIO_CFGRx if the I/O line is Secure).

Only a hardware reset can release fields listed above.

- Interrupt Freeze

Setting the FINT bit of the PIO_IOFRx will freeze the following fields of the Non-Secure I/O lines defined in the PIO_MSKRx:

- IFEN: Input Filter Enable
- IFSCEN: Input Filter Slow Clock Enable
- EVTSEL: Event Selection

For Secure I/O lines, use the FINT bit of the S_PIO_IOFRx and the S_PIO_MSKRx to freeze the fields above.

When the “Interrupt Freeze” is currently active on an I/O line, the ICFS flag is set when reading the PIO_CFGRx of the I/O line (or the S_PIO_CFGRx if the I/O line is Secure).

Only a hardware reset can release fields listed above.

Table 36-20: Interleaved Mapping DDR-SDRAM Configuration Mapping: 8K Rows /512/1024/2048 Columns, 4 banks

CPU Address Line																												
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**
			Row[13:0]													Bk[1:0]	Column[8:0]										M[1:0]	
			Row[13:0]												Bk[1:0]	Column[9:0]										M[1:0]		
		Row[13:0]												Bk[1:0]	Column[10:0]										M[1:0]			

Table 36-21: Sequential Mapping DDR-SDRAM Configuration Mapping: 8K Rows /1024 Columns, 8 banks

CPU Address Line																												
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**
	Bk[2:0]			Row[12:0]													Column[9:0]										M[1:0]	

Table 36-22: Interleaved Mapping DDR-SDRAM Configuration Mapping: 8K Rows /1024 Columns, 8 banks

CPU Address Line																												
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**
	Row[12:0]													Bk[2:0]			Column[9:0]										M[1:0]	

Table 36-23: Sequential Mapping DDR-SDRAM Configuration Mapping: 16K Rows /1024 Columns, 4 banks

CPU Address Line																												
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**
	Bk[1:0]		Row[13:0]														Column[9:0]										M[1:0]	

Table 36-24: Interleaved Mapping DDR-SDRAM Configuration Mapping: 16K Rows /1024 Columns, 4 banks

CPU Address Line																												
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**
	Row[13:0]														Bk[1:0]		Column[9:0]										M[1:0]	

Table 36-25: Sequential Mapping DDR-SDRAM Configuration Mapping: 16K Rows /1024 Columns, 8 banks

CPU Address Line																												
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**
Bk[2:0]			Row[13:0]														Column[9:0]										M[1:0]	

Table 36-26: Interleaved Mapping DDR-SDRAM Configuration Mapping: 16K Rows /1024 Columns, 8 banks

CPU Address Line																												
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**
Row[13:0]														Bk[2:0]			Column[9:0]										M[1:0]	

Data Timeout Multiplier is defined by DTOMUL as shown in the following table:

Value	Name	Description
0	X1	DTOCYC
1	X16	DTOCYC x 16
2	X128	DTOCYC x 128
3	X256	DTOCYC x 256
4	X1024	DTOCYC x 1024
5	X4096	DTOCYC x 4096
6	X65536	DTOCYC x 65536
7	X1048576	DTOCYC x 1048576

If the data timeout set by DTOCYC and DTOMUL has been exceeded, the Data Timeout Error flag (DTOE) in the NFC Status Register (NFC_SR) raises.

NFCSPARESIZE: NAND Flash Spare Area Size Retrieved by the Host Controller

The spare size is set to (NFCSPARESIZE + 1) * 4 bytes. The spare area is only retrieved when RSPARE or WSPARE is activated.

39.7.91 High-End Overlay V DMA Head Register

Name: LCDC_HEOVHEAD

Address: 0xF000037C

Access: Read/Write

31	30	29	28	27	26	25	24
VHEAD							

23	22	21	20	19	18	17	16
VHEAD							

15	14	13	12	11	10	9	8
VHEAD							

7	6	5	4	3	2	1	0
VHEAD							

VHEAD: DMA Head Pointer

The Head Pointer points to a new descriptor.

39.7.127 High-End Overlay Configuration Register 32

Name: LCDC_HEOCFG32

Address: 0xF000040C

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
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
XPHI7COEFF4							

XPHI7COEFF4: Horizontal Coefficient for phase 7 tap 4

Coefficient format is 1 sign bit and 7 fractional bits.

39.7.131 High-End Overlay Configuration Register 36

Name: LCDC_HEOCFG36

Address: 0xF000041C

Access: Read/Write

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

23	22	21	20	19	18	17	16
YPHI3COEFF2							

15	14	13	12	11	10	9	8
YPHI3COEFF1							

7	6	5	4	3	2	1	0
YPHI3COEFF0							

YPHI3COEFF0: Vertical Coefficient for phase 3 tap 0

Coefficient format is 1 sign bit and 7 fractional bits.

YPHI3COEFF1: Vertical Coefficient for phase 3 tap 1

Coefficient format is 1 magnitude bit and 7 fractional bits.

YPHI3COEFF2: Vertical Coefficient for phase 3 tap 2

Coefficient format is 1 sign bit and 7 fractional bits.

40.6.3.1 Packet Buffer DMA

- Easier to guarantee maximum line rate due to the ability to store multiple frames in the packet buffer, where the number of frames is limited by the amount of packet buffer memory and Ethernet frame size
- Full store and forward, or partial store and forward programmable options (partial store will cater for shorter latency requirements)
- Support for Transmit TCP/IP checksum offload
- Support for priority queueing
- When a collision on the line occurs during transmission, the packet will be automatically replayed directly from the packet buffer memory rather than having to re-fetch through the AHB (full store and forward ONLY)
- Received error packets are automatically dropped before any of the packet is presented to the AHB (full store and forward ONLY), thus reducing AHB activity
- Supports manual RX packet flush capabilities
- Optional RX packet flush when there is lack of AHB resource

40.6.3.2 Partial Store and Forward Using Packet Buffer DMA

The DMA uses SRAM-based packet buffers, and can be programmed into a low latency mode, known as Partial Store and Forward. This allows for a reduced latency as the full packet is not buffered before forwarding. Note that this option is only available when the device is configured for full duplex operation.

This feature is enabled via the programmable TX and RX Partial Store and Forward registers. When the transmit Partial Store and Forward mode is activated, the transmitter will only begin to forward the packet to the MAC when there is enough packet data stored in the packet buffer. Likewise, when the receive Partial Store and Forward mode is activated, the receiver will only begin to forward the packet to the AHB when enough packet data is stored in the packet buffer. The amount of packet data required to activate the forwarding process is programmable via watermark registers which are located at the same address as the partial store and forward enable bits.

Note that the minimum operational value for the TX partial store and forward watermark is 20. There is no operational limit for the RX partial store and forward watermark. Enabling partial store and forward is a useful means to reduce latency, but there are performance implications.

The GMAC DMA uses separate transmit and receive lists of buffer descriptors, with each descriptor describing a buffer area in memory. This allows Ethernet packets to be broken up and scattered around the AHB memory space.

40.6.3.3 Receive AHB Buffers

Received frames, optionally including FCS, are written to receive AHB buffers stored in memory. The receive buffer depth is programmable in the range of 64 bytes to 16 Kbytes through the DMA Configuration register, with the default being 128 bytes.

The start location for each receive AHB buffer is stored in memory in a list of receive buffer descriptors at an address location pointed to by the receive buffer queue pointer. The base address for the receive buffer queue pointer is configured in software using the Receive Buffer Queue Base Address register.

Each list entry consists of two words. The first is the address of the receive AHB buffer and the second the receive status. If the length of a receive frame exceeds the AHB buffer length, the status word for the used buffer is written with zeroes except for the “start of frame” bit, which is always set for the first buffer in a frame. Bit zero of the address field is written to 1 to show the buffer has been used. The receive buffer manager then reads the location of the next receive AHB buffer and fills that with the next part of the received frame data. AHB buffers are filled until the frame is complete and the final buffer descriptor status word contains the complete frame status. Refer to Table 40-4 for details of the receive buffer descriptor list.

Each receive AHB buffer start location is a word address. The start of the first AHB buffer in a frame can be offset by up to three bytes, depending on the value written to bits 14 and 15 of the Network Configuration register. If the start location of the AHB buffer is offset, the available length of the first AHB buffer is reduced by the corresponding number of bytes.

Table 40-4: Receive Buffer Descriptor Entry

Bit	Function
Word 0	
31:2	Address of beginning of buffer
1	Wrap—marks last descriptor in receive buffer descriptor list.
0	Ownership—needs to be zero for the GMAC to write data to the receive buffer. The GMAC sets this to one once it has successfully written a frame to memory. Software has to clear this bit before the buffer can be used again.
Word 1	

RXRDY_TXKL: Received OUT Data Interrupt Disable

0: No effect.

1: Disable Received OUT Data Interrupt.

TX_COMPLT: Transmitted IN Data Complete Interrupt Disable

0: No effect.

1: Disable Transmitted IN Data Complete Interrupt.

TXRDY_TRER: TX Packet Ready/Transaction Error Interrupt Disable

0: No effect.

1: Disable TX Packet Ready/Transaction Error Interrupt.

ERR_FL_ISO: Error Flow Interrupt Disable

0: No effect.

1: Disable Error Flow ISO Interrupt.

ERR_CRC_NTR: ISO CRC Error/Number of Transaction Error Interrupt Disable

0: No effect.

1: Disable Error CRC ISO/Error Number of Transaction Interrupt.

ERR_FLUSH: bank flush error Interrupt Disable

0: No effect.

1: Disable Bank Flush Error Interrupt.

BUSY_BANK: Busy Bank Interrupt Disable

0: No effect.

1: Disable Busy Bank Interrupt.

SHRT_PCKT: Short Packet Interrupt Disable

For OUT endpoints:

0: No effect.

1: Disable Short Packet Interrupt.

For IN endpoints: Never automatically add a zero length packet at end of DMA transfer.

ERR_OVFLW: Overflow Error (cleared upon USB reset)

This bit is set by hardware when a new too-long packet is received.

Example: If the user programs an endpoint 64 bytes wide and the host sends 128 bytes in an OUT transfer, then the Overflow Error bit is set.

This bit is updated at the same time as the BYTE_COUNT field.

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

RXRDY_TXKL: Received OUT Data/KILL Bank (cleared upon USB reset)

– **Received OUT Data** (for OUT endpoint or Control endpoint):

This bit is set by hardware after a new packet has been stored in the endpoint FIFO.

This bit is cleared by the device firmware after reading the OUT data from the endpoint.

For multi-bank endpoints, this bit may remain active even when cleared by the device firmware, this if an other packet has been received meanwhile.

Hardware assertion of this bit may generate an interrupt if enabled by the UDPHS_EPTCTLx register RXRDY_TXKL bit.

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

– **KILL Bank** (for IN endpoint):

– The bank is really cleared or the bank is sent, BUSY_BANK_STA is decremented.

– The bank is not cleared but sent on the IN transfer, TX_COMPLT

– The bank is not cleared because it was empty. The user should wait that this bit is cleared before trying to clear another packet.

Note: “Kill a packet” may be refused if at the same time, an IN token is coming and the current packet is sent on the UDPHS line. In this case, the TX_COMPLT bit is set. Take notice however, that if at least two banks are ready to be sent, there is no problem to kill a packet even if an IN token is coming. In fact, in that case, the current bank is sent (IN transfer) and the last bank is killed.

TX_COMPLT: Transmitted IN Data Complete (cleared upon USB reset)

This bit is set by hardware after an IN packet has been accepted (ACK’ed) by the host.

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

TXRDY: TX Packet Ready (cleared upon USB reset)

This bit is cleared by hardware after the host has acknowledged the packet.

For Multi-bank endpoints, this bit may remain clear even after software is set if another bank is available to transmit.

Hardware clear of this bit may generate an interrupt if enabled by the UDPHS_EPTCTLx register TXRDY bit.

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

RX_SETUP: Received SETUP (cleared upon USB reset)

– (for Control endpoint only)

This bit is set by hardware when a valid SETUP packet has been received from the host.

It is cleared by the device firmware after reading the SETUP data from the endpoint FIFO.

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

47.10.82 TWI FIFO Interrupt Enable Register

Name: FLEX_TWI_FIER

Address: 0xF8034664 (0), 0xF8038664 (1), 0xFC010664 (2), 0xFC014664 (3), 0xFC018664 (4)

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
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF

TXFEF: TXFEF Interrupt Enable

TXFFF: TXFFF Interrupt Enable

TXFTHF: TXFTHF Interrupt Enable

RXFEF: RXFEF Interrupt Enable

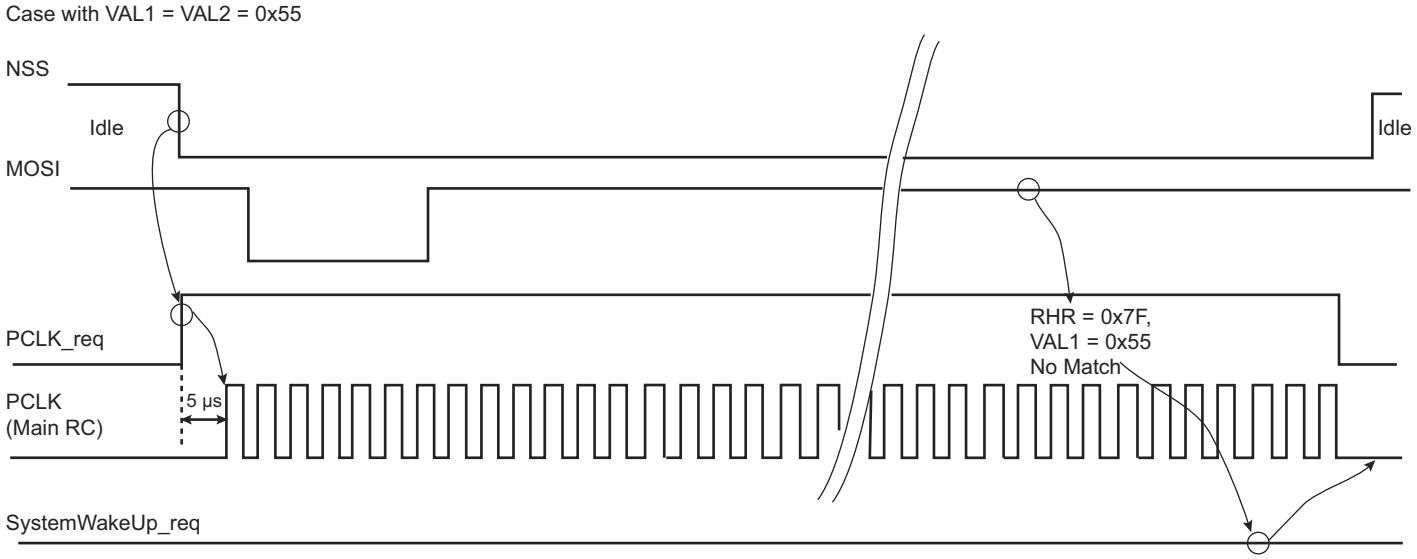
RXFFF: RXFFF Interrupt Enable

RXFTHF: RXFTHF Interrupt Enable

TXFPTEF: TXFPTEF Interrupt Enable

RXFPTEF: RXFPTEF Interrupt Enable

Figure 49-15: Asynchronous Event Generating Only Partial Wakeup



49.7.7 FIFOs

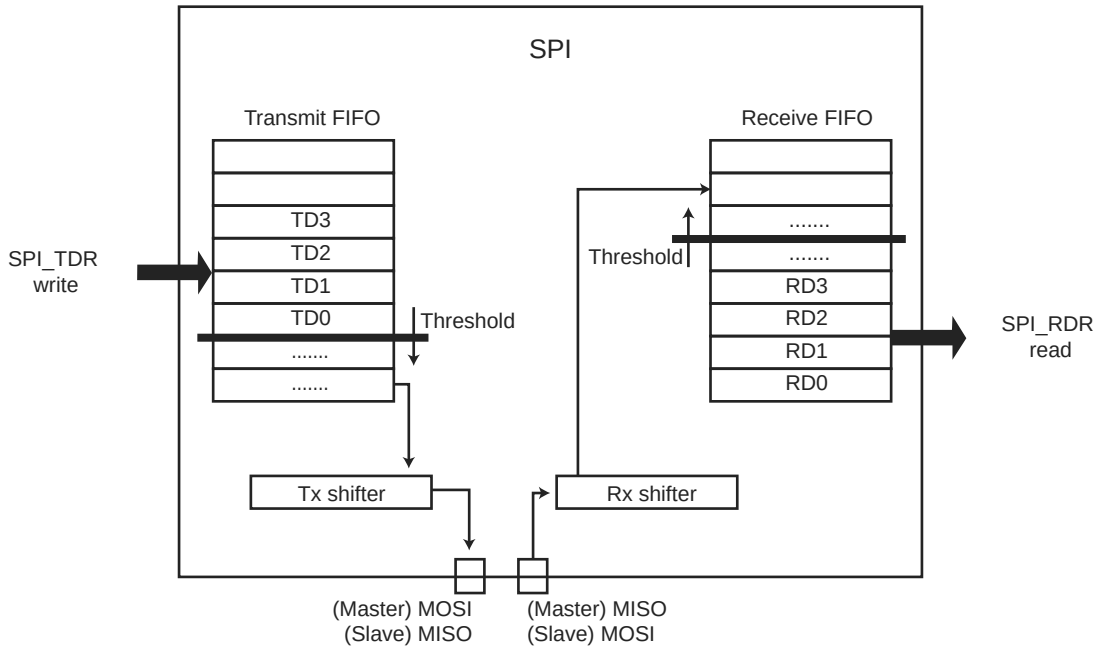
49.7.7.1 Overview

The SPI includes two FIFOs which can be enabled/disabled using SPI_CR.FIFOEN/FIFODIS. It is recommended to disable the SPI module before enabling or disabling the SPI FIFOs (SPI_CR.SPIDIS).

Writing SPI_CR.FIFOEN to ‘1’ enables a 16-data Transmit FIFO and a 16-data Receive FIFO.

It is possible to write or to read single or multiple data in the same access to SPI_TDR/RDR. Refer to Section 49.7.7.6 “Single Data Mode” and to Section 49.7.7.7 “Multiple Data Mode”.

Figure 49-16: FIFOs Block Diagram



49.7.7.2 Sending Data with FIFO Enabled

When the Transmit FIFO is enabled, write access to SPI_TDR loads the Transmit FIFO.

Figure 52-28: Cosited Filter Configuration

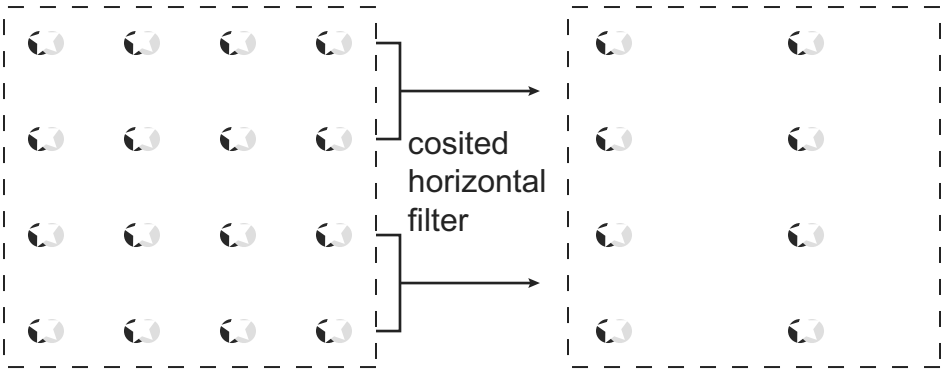
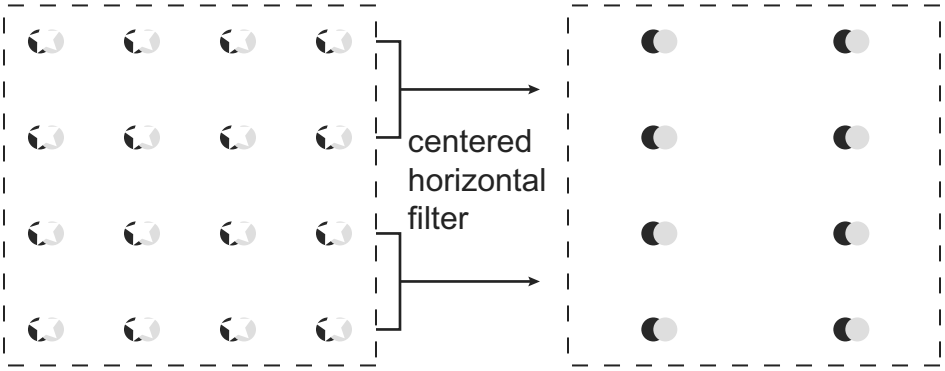


Figure 52-29: Centered Filter Configuration



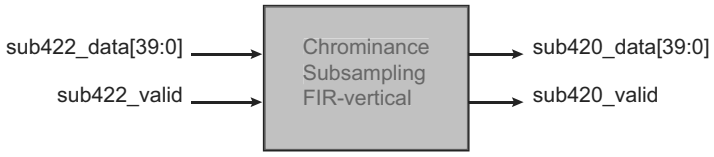
The SUB422 module performs luminance and chrominance packing. When the line length is odd, the missing luminance is a copy of the last but one luminance. It also means that the final dma stream written to memory is equal to the original horizontal size plus one when the line length is odd.

SUB422_DATA Slice	Line Length Even	Line Length Odd
sub422_data[39:30]	Y(n)	Y(n-1)
sub422_data[29:20]	Y(n-1)	Y(n-1)
sub422_data[19:10]	Cb (filtered)	Cb (filtered)
sub422_data[9:0]	Cr (filtered)	Cr (filtered)

52.5.13 4:2:2 To 4:2:0 Chrominance Vertical Subsampler (SUB420) Module

The chrominance subsampling divides the vertical chrominance sampling rate by two. A vertical low pass filter is applied to avoid aliasing effect. Two different filters are used when the source frame is interlaced, and the filter configuration depends on the field value (the field is propagated in the video pipeline).

Figure 52-30: SUB420 Block Diagram



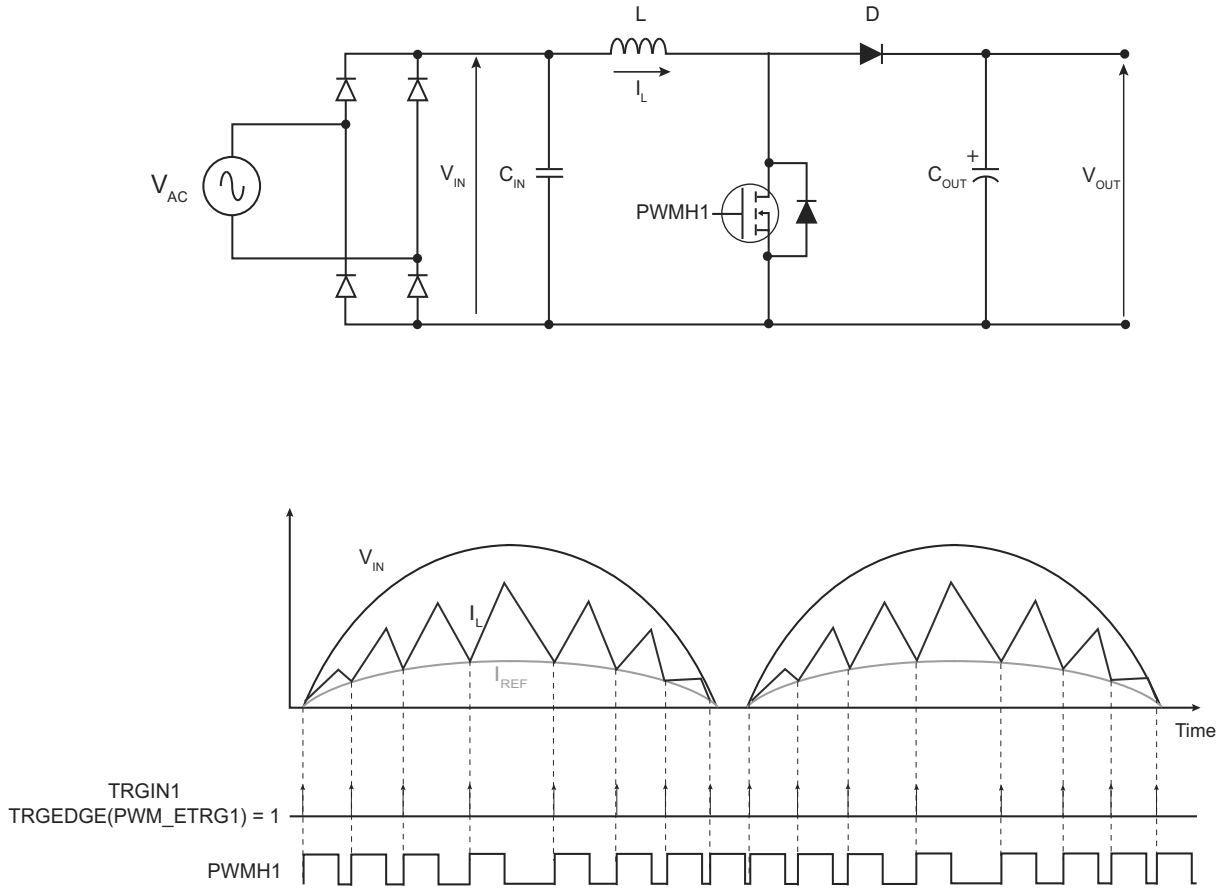
- Application Example

The external PWM Reset mode can be used in power factor correction applications.

In the example below, the external trigger input is the PWMEXTRG1 (therefore the PWM channel used for regulation is the channel 1). The PWM channel 1 period (CPRD in the PWM Channel Period Register of the channel 1) must be programmed so that the TRGIN1 event always triggers before the PWM channel 1 period elapses.

In Figure 56-28, an external circuit (not shown) is required to sense the inductor current I_L. The internal PWM counter of the channel 1 is cleared when the inductor current falls below a specific threshold (I_{REF}). This starts a new PWM period and increases the inductor current.

Figure 56-28: External PWM Reset Mode: Power Factor Correction Application



56.7.32 PWM Stepper Motor Mode Register

Name: PWM_SMMR

Address: 0xF802C0B0

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	DOWN1	DOWN0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	GCEN1	GCEN0

GCENx: Gray Count Enable

0: Disable Gray count generation on PWML[2*x], PWMH[2*x], PWML[2*x +1], PWMH[2*x +1]

1: Enable Gray count generation on PWML[2*x], PWMH[2*x], PWML[2*x +1], PWMH[2*x +1].

DOWNx: Down Count

0: Up counter.

1: Down counter.

59.4.8 AESB Input Data Register x

Name: AESB_IDATARx

Address: 0xF001C040

Access: Write-only

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

IDATA: Input Data Word

The four 32-bit Input Data registers set the 128-bit data block used for encryption/decryption.

AESB_IDATAR0 corresponds to the first word of the data to be encrypted/decrypted, AESB_IDATAR3 to the last one.

These registers are write-only to prevent the input data from being read by another application.

60.5.3 AES Interrupt Enable Register

Name: AES_IER

Address: 0xF002C010

Access: Write-only

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

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

0: No effect.

1: Enables the corresponding interrupt.

DATRDY: Data Ready Interrupt Enable

URAD: Unspecified Register Access Detection Interrupt Enable

TAGRDY: GCM Tag Ready Interrupt Enable

EOPAD: End of Padding Interrupt Enable

PLENERR: Padding Length Error Interrupt Enable

61.4.7 Automatic Check

The SHA module features an automatic check of the hash result with the expected hash. A check failure can generate an interrupt if configured in the SHA Interrupt Enable register (SHA_IER).

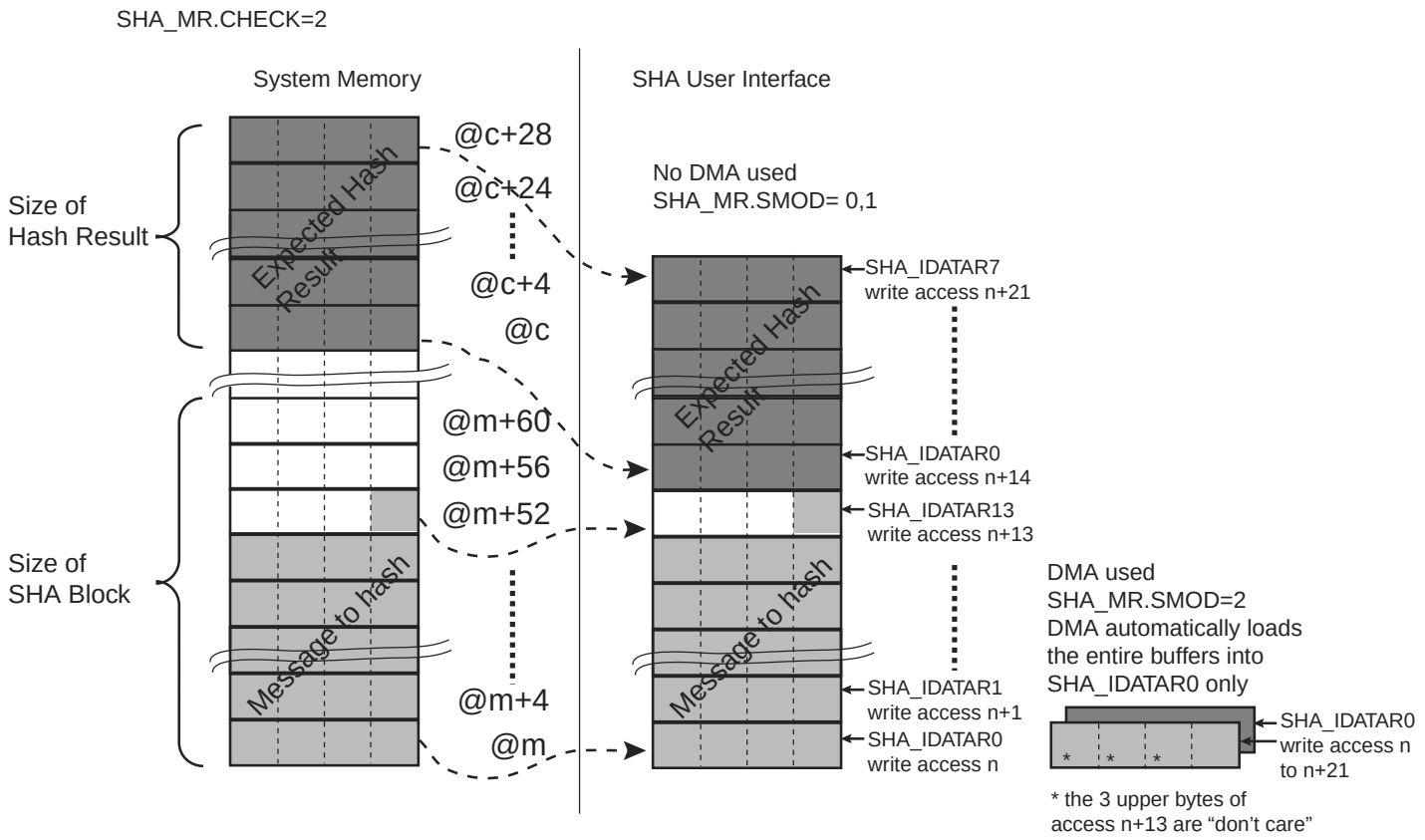
Automatic check requires the automatic padding feature to be enabled (MSGSIZE and BYTCNT fields must be greater than 0).

There are two methods to configure the expected hash result:

- if SHA_MR.CHECK = 1, the expected hash result is read from the internal register (IR1). This method cannot be used when HMAC algorithms is selected because this register is already used to store user initial hash values for the second hash processing. IR1 cannot be read by software.
- If SHA_MR.CHECK = 2, the expected hash result is written in the SHA_IDATARx after the message.

When SHA_MR.CHECK = 2, the method can provide more flexibility of use if a message is stored in system memory together with its expected hash result. A DMA with linked list can be used to ease the transfer of the message and its expected hash result.

Figure 61-2: Message and Expected Hash Result Memory Mapping



The number of 32-bit words of the hash result to check with the expected hash can be selected with SHA_MR.CHKCNT. The status of the check is available in the CHKST field in the SHA Interrupt Status register (SHA_ISR).

An interrupt can be generated (if enabled) when the check is completed. The check occurs several clock cycles after the computation of the requested hash, so the interrupt and the CHECKF bit are set several clock cycles after the DATRDY flag of the SHA_ISR.

61.4.8 Protocol Layers Improved Performances

The SHA can be tightly coupled to the AES module to improve performances when processing protocol layers such as IPsec or OpenSSL.

When the AES is configured to be tightly coupled to SHA (AES_MR), SHA must be always configured in Double Buffer mode (SHA_MR.DUALBUFF = 1).

Refer to the section “Advanced Encryption Standard (AES)” for details.

Table 66-4: DC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDISC}	DC Supply Peripheral I/Os	ISC I/Os Lines	1.65	–	3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	20	mV
V_{DDOSC}	DC Supply Oscillator	–	1.65	–	3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	15	mV
$V_{DDIODDR}$	DC Supply SDRAM I/Os	- LPDDR1-DDR2 Interface I/O lines - LPDDR2-LPDDR3 Interface I/O lines - DDR3L Interface I/O lines - DDR3 Interface I/O lines	1.7 1.14 1.283 1.425	1.8 1.2 1.35 1.5	1.95 1.30 1.45 1.575	V
V_{IL}	Low-level Input Voltage(2)	VDDIO in 3.3V range	-0.3	–	0.8	V
		VDDIO in 1.8V range	-0.3	–	0.3 x V_{DDIO}	
V_{IH}	High-level Input Voltage(2)	VDDIO in 3.3V range	2	–	V_{DDIO} +0.3	V
		VDDIO in 1.8V range	0.7 x V_{DDIO}	–	V_{DDIO} +0.3	
V_{OL}	Low-level Output Voltage	I_O MAX	–	–	0.41	V
V_{OH}	High-level Output Voltage	I_O MAX	V_{DDIO} - 0.4	–	–	V
V_{hys}	Schmitt Trigger Hysteresis	All PIO lines, VDDIOx in 3.3V range	0.34	–	–	V
		All PIO lines, VDDIOx in 1.8V range	0.2	–	–	
I_{OL}	I_{OIL}(or I_{SINK}) (VOL = 0.4V)	All GPIO_x, 1.8V: Low	-1	–	–	mA
		All GPIO_x, 1.8V: Medium	-10	–	–	
		All GPIO_x, 1.8V: High	-18	–	–	
I_{OH}	I_{OH} (or I_{SOURCE}) (VOH = VDDIO - 0.4V)	All GPIO_x, 1.8V: Low	–	–	1	mA
		All GPIO_x, 1.8V: Medium	–	–	10	
		All GPIO_x, 1.8V: High	–	–	18	
I_{OL}	I_{OIL}(or I_{SINK}) (VOL = 0.4V)	All GPIO_x, 3.3V: Low	-2	–	–	mA
		All GPIO_x, 3.3V: Medium	-20	–	–	
		All GPIO_x, 3.3V: High	-32	–	–	
I_{OH}	I_{OH} (or I_{SOURCE}) (VOH = VDDIO - 0.4V)	All GPIO_x, 3.3V: Low	–	–	2	mA
		All GPIO_x, 3.3V: Medium	–	–	20	
		All GPIO_x, 3.3V: High	–	–	32	
I_{IL}	Low-level Input Current	LCDPCK, ISC_MCK, GPIO, QSPI_SCK	-1	–	1	µA
I_{IH}	High-level Input Current	LCDPCK, ISC_MCK, GPIO, QSPI_SCK	-1	–	1	µA
I_{IL}	Low-level Input Current	GPIO_AD	-1	–	1	µA
I_{IH}	High-level Input Current	GPIO_AD	-1	–	1	µA