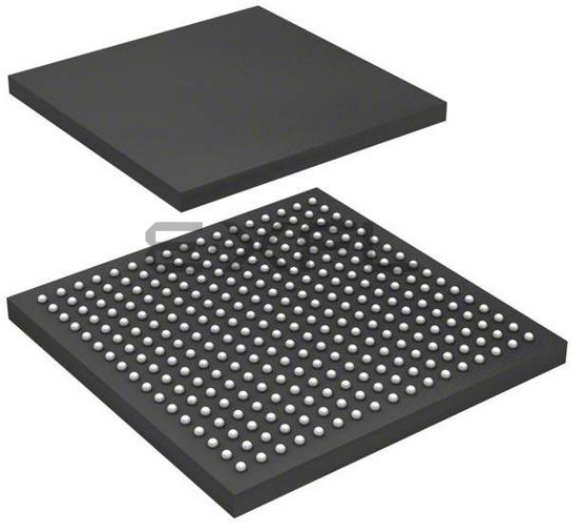


Welcome to E-XFL.COM

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 ~ 105°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/atsama5d27a-cnr

18.13 Matrix (H64MX/H32MX) User Interface

The user interface below is constructed with the maximum numbers of masters, slaves and regions by slave that are possible on the two product matrixes. The exact number of these elements must be used to deduce the exact register description of the Matrix user interface.

The exact numbers of these elements can be found in:

- Section 18.3 “64-bit Matrix (H64MX)”
- Section 18.4 “32-bit Matrix (H32MX)”

Table 18-10: Register Mapping

Offset	Register	Name	Access	Reset
0x0000	Master Configuration Register 0	MATRIX_MCFG0	Read/Write	0x00000004
0x0004	Master Configuration Register 1	MATRIX_MCFG1	Read/Write	0x00000004
0x0008	Master Configuration Register 2	MATRIX_MCFG2	Read/Write	0x00000004
0x000C	Master Configuration Register 3	MATRIX_MCFG3	Read/Write	0x00000004
0x0010	Master Configuration Register 4	MATRIX_MCFG4	Read/Write	0x00000004
0x0014	Master Configuration Register 5	MATRIX_MCFG5	Read/Write	0x00000004
0x0018	Master Configuration Register 6	MATRIX_MCFG6	Read/Write	0x00000004
0x001C	Master Configuration Register 7	MATRIX_MCFG7	Read/Write	0x00000004
0x0020	Master Configuration Register 8	MATRIX_MCFG8	Read/Write	0x00000004
0x0024	Master Configuration Register 9	MATRIX_MCFG9	Read/Write	0x00000004
0x0028	Master Configuration Register 10	MATRIX_MCFG10	Read/Write	0x00000004
0x002C	Master Configuration Register 11	MATRIX_MCFG11	Read/Write	0x00000004
0x0030–0x003C	Reserved	–	–	–
0x0040	Slave Configuration Register 0	MATRIX_SCFG0	Read/Write	0x000001FF
0x0044	Slave Configuration Register 1	MATRIX_SCFG1	Read/Write	0x000001FF
0x0048	Slave Configuration Register 2	MATRIX_SCFG2	Read/Write	0x000001FF
0x004C	Slave Configuration Register 3	MATRIX_SCFG3	Read/Write	0x000001FF
0x0050	Slave Configuration Register 4	MATRIX_SCFG4	Read/Write	0x000001FF
0x0054	Slave Configuration Register 5	MATRIX_SCFG5	Read/Write	0x000001FF
0x0058	Slave Configuration Register 6	MATRIX_SCFG6	Read/Write	0x000001FF
0x005C	Slave Configuration Register 7	MATRIX_SCFG7	Read/Write	0x000001FF
0x0060	Slave Configuration Register 8	MATRIX_SCFG8	Read/Write	0x000001FF
0x0064	Slave Configuration Register 9	MATRIX_SCFG9	Read/Write	0x000001FF
0x0068	Slave Configuration Register 10	MATRIX_SCFG10	Read/Write	0x000001FF
0x006C	Slave Configuration Register 11	MATRIX_SCFG11	Read/Write	0x000001FF
0x0070	Slave Configuration Register 12	MATRIX_SCFG12	Read/Write	0x000001FF
0x0074	Slave Configuration Register 13	MATRIX_SCFG13	Read/Write	0x000001FF
0x0078	Slave Configuration Register 14	MATRIX_SCFG14	Read/Write	0x000001FF
0x007C	Reserved	–	–	–
0x0080	Priority Register A for Slave 0	MATRIX_PRAS0	Read/Write	0x00000000
0x0084	Priority Register B for Slave 0	MATRIX_PRBS0	Read/Write	0x00000000

26.6.12 RTC Interrupt Enable Register

Name: RTC_IER

Address: 0xF80480D0

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
–	–	TDERREN	CALEN	TIMEN	SECEN	ALREN	ACKEN

ACKEN: Acknowledge Update Interrupt Enable

0: No effect.

1: The acknowledge for update interrupt is enabled.

ALREN: Alarm Interrupt Enable

0: No effect.

1: The alarm interrupt is enabled.

SECEN: Second Event Interrupt Enable

0: No effect.

1: The second periodic interrupt is enabled.

TIMEN: Time Event Interrupt Enable

0: No effect.

1: The selected time event interrupt is enabled.

If the RTC is configured in UTC mode, this bit has no effect.

CALEN: Calendar Event Interrupt Enable

0: No effect.

1: The selected calendar event interrupt is enabled.

If the RTC is configured in UTC mode, this bit has no effect.

TDERREN: Time and/or Date Error Interrupt Enable

0: No effect.

1: The time and date error interrupt is enabled.

If the RTC is configured in UTC mode, this bit has no effect.

37.20.22 PMECC Error Location Primitive Register

Name: HSMC_ELPRIM

Address: 0xF8014504

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
PRIMITIV							

7	6	5	4	3	2	1	0
PRIMITIV							

PRIMITIV: Primitive Polynomial

This field indicates the Primitive Polynomial used in the ECC computation.

38.5.3 Peripheral Synchronized Transfer

A peripheral hardware request interface is used to control the pace of the chunk transfer. When a peripheral is ready to transmit or receive a chunk of data, it asserts its request line and the DMA Controller transfers a data to or from the memory to the peripheral.

38.5.3.1 Software Triggered Synchronized Transfer

The Peripheral hardware request can be software controlled using the SWREQ field of the XDMAC Global Channel Software Request Register (XDMAC_GSWR). The peripheral synchronized transfer is paced using a processor write access in the XDMAC_GSWR. Each bit of that register triggers a transfer request. The XDMAC Global Channel Software Request Status Register (XDMAC_GSWS) indicates the status of the request; when set, the request is still pending.

38.5.4 XDMAC Transfer Software Operation

Note: When a memory-to-memory transfer is performed, configure the field XDMAC_CCx.PERID (where 'x' is the index of the channel used for transfer) to an unused peripheral ID (refer to Table 11-1 "Peripheral Identifiers").

38.5.4.1 Single Block With Single Microblock Transfer

1. Read the XDMAC Global Channel Status Register (XDMAC_GS) to select a free channel.
2. Clear the pending Interrupt Status bit(s) by reading the selected XDMAC Channel x Interrupt Status Register (XDMAC_CISx).
3. Write the XDMAC Channel x Source Address Register (XDMAC_CSAx) for channel x.
4. Write the XDMAC Channel x Destination Address Register (XDMAC_CDAx) for channel x.
5. Program field UBLEN in the XDMAC Channel x Microblock Control Register (XDMAC_CUBCx) with the number of data.
6. Program the XDMAC Channel x Configuration Register (XDMAC_CCx):
   a) Clear XDMAC_CCx.TYPE for a memory-to-memory transfer, otherwise set this bit.
   b) Configure XDMAC_CCx.MBSIZE to the memory burst size used.
   c) Configure XDMAC_CCx.SAM and DAM to Memory Addressing mode.
   d) Configure XDMAC_CCx.DSYNC to select the peripheral transfer direction.
   e) Set XDMAC_CCx.PROT to activate a secure channel.
   f) Configure XDMAC_CCx.CSIZE to configure the channel chunk size (only relevant for peripheral synchronized transfer).
   g) Configure XDMAC_CCx.DWIDTH to configure the transfer data width.
   h) Configure XDMAC_CCx.SIF, XDMAC_CCx.DIF to configure the master interface used to read data and write data, respectively.
   i) Configure XDMAC_CCx.PERID to select the active hardware request line (only relevant for a peripheral synchronized transfer).
   j) Set XDMAC_CCx.SWREQ to use a software request (only relevant for a peripheral synchronized transfer).
7. Clear the following five registers:
   - XDMAC Channel x Next Descriptor Control Register (XDMAC_CNDCx)
   - XDMAC Channel x Block Control Register (XDMAC_CBCx)
   - XDMAC Channel x Data Stride Memory Set Pattern Register (XDMAC_CDS_MSPx)
   - XDMAC Channel x Source Microblock Stride Register (XDMAC_CSUSx)
   - XDMAC Channel x Destination Microblock Stride Register (XDMAC_CDUSx)

   This indicates that the linked list is disabled, there is only one block and striding is disabled.
8. Enable the Microblock interrupt by writing a '1' to bit BIE in the XDMAC Channel x Interrupt Enable Register (XDMAC_CIEx). Enable the Channel x Interrupt Enable bit by writing a '1' to bit IEx in the XDMAC Global Interrupt Enable Register (XDMAC_GIE).
9. Enable channel x by writing a '1' to bit ENx in the XDMAC Global Channel Enable Register (XDMAC_GE). XDMAC_GS.STx (XDMAC Channel x Status bit) is set by hardware.
10. Once completed, the DMA channel sets XDMAC_CISx.BIS (End of Block Interrupt Status bit) and generates an interrupt. XDMAC_GS.STx is cleared by hardware. The software can either wait for an interrupt or poll the channel status bit.

38.5.4.2 Single Block Transfer With Multiple Microblock

1. Read the XDMAC_GS register to choose a free channel.
2. Clear the pending Interrupt Status bit by reading the chosen XDMAC_CISx register.
3. Write the XDMAC_CSAx register for channel x.
4. Write the XDMAC_CDAx register for channel x.
5. Program XDMAC_CUBCx.UBLEN with the number of data.

40.8.49 GMAC Frames Transmitted Register

Name: GMAC_FT

Address: 0xF8008108

Access: Read-only

31	30	29	28	27	26	25	24
FTX							

23	22	21	20	19	18	17	16
FTX							

15	14	13	12	11	10	9	8
FTX							

7	6	5	4	3	2	1	0
FTX							

FTX: Frames Transmitted without Error

Frames transmitted without error. This register counts the number of frames successfully transmitted, i.e., no underrun and not too many retries. Excludes pause frames.

40.8.56 GMAC 256 to 511 Byte Frames Transmitted Register

Name: GMAC_TBFT511

Address: 0xF8008124

Access: Read-only

31	30	29	28	27	26	25	24
NFTX							

23	22	21	20	19	18	17	16
NFTX							

15	14	13	12	11	10	9	8
NFTX							

7	6	5	4	3	2	1	0
NFTX							

NFTX: 256 to 511 Byte Frames Transmitted without Error

This register counts the number of 256 to 511 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

40.8.88 GMAC Receive Overruns Register

Name: GMAC_ROE

Address: 0xF80081A4

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
–	–	–	–	–	–	RXOVR	

7	6	5	4	3	2	1	0
RXOVR							

RXOVR: Receive Overruns

This register counts the number of frames that are address recognized but were not copied to memory due to a receive overrun.

40.8.89 GMAC IP Header Checksum Errors Register

Name: GMAC_IHCE

Address: 0xF80081A8

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
HCKER							

HCKER: IP Header Checksum Errors

This register counts the number of frames discarded due to an incorrect IP header checksum, but are between 64 and 1518 bytes (1536 bytes if bit 8 is set in the Network Configuration Register) and do not have a CRC error, an alignment error, nor a symbol error.

42.7.4 UHPHS USB Command Register

Name: UHPHS_USBCMD

Access: Read/Write

31	30	29	28	27	26	25	24
–							
23	22	21	20	19	18	17	16
ITC							
15	14	13	12	11	10	9	8
–				ASPME	–	ASPMC	
7	6	5	4	3	2	1	0
LHCR	IAAD	ASE	PSE	FLS		HCRESET	RS

The Command Register indicates the command to be executed by the serial bus host controller. Writing to the register causes a command to be executed.

RS: Run/Stop (read/write)

0: Stop (default value).

1: Run.

When set to 1, the Host Controller proceeds with execution of the schedule. The Host Controller continues execution as long as this bit is set to 1. When this bit is set to 0, the Host Controller completes the current and any actively pipelined transactions on the USB and then halts. The Host Controller must halt within 16 micro-frames after software clears the Run bit. The HC Halted bit in the status register indicates when the Host Controller has finished its pending pipelined transactions and has entered the stopped state. Software must not write 1 to this field unless the host controller is in the Halted state (i.e., HCHalted in the UHPHS_USBSTS register is 1). Doing so will yield undefined results.

HCRESET: Host Controller Reset (read/write)

This control bit is used by software to reset the host controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset.

When software writes a 1 to this bit, the Host Controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports.

PCI Configuration registers are not affected by this reset. All operational registers, including port registers and port state machines, are set to their initial values. Port ownership reverts to the companion host controller(s) with side effects. Software must reinitialize the host controller in order to return the host controller to an operational state.

This bit is set to 0 by the Host Controller when the reset process is complete. Software cannot terminate the reset process early by writing a 0 to this register.

Software should not set this bit to 1 when the HCHalted bit in the UHPHS_USBSTS register is 0. Attempting to reset an actively running host controller will result in undefined behavior.

FLS: Frame List Size (read/write or read-only)

This field is R/W only if Programmable Frame List Flag in the UHPHS_HCCPARAMS registers is set to 1. This field specifies the size of the frame list. The size of the frame list controls which bits in the Frame Index Register should be used for the Frame List Current index.

00b: 1024 elements (4096 bytes) (default value).

01b: 512 elements (2048 bytes).

10b: 256 elements (1024 bytes), for resource-constrained environments.

11b: Reserved.

PSE: Periodic Schedule Enable (read/write)

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

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

1: Use the UHPHS_PERIODICLISTBASE register to access the Periodic Schedule.

ASE: Asynchronous Schedule Enable (read/write)

46.7.2 TWIHS Control Register (FIFO_ENABLED)

Name: TWIHS_CR (FIFO_ENABLED)

Address: 0xF8028000 (0), 0xFC028000 (1)

Access: Write-only

31	30	29	28	27	26	25	24
–	–	FIFODIS	FIFOEN	–	TXFLCLR	RXFCLR	TXFCLR
23	22	21	20	19	18	17	16
–	–	–	–	–	–	ACMDIS	ACMEN
15	14	13	12	11	10	9	8
CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
7	6	5	4	3	2	1	0
SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START

This configuration is relevant only if TWIHS_CR.FIFOEN = ‘1’.

START: Send a START Condition

0: No effect.

1: A frame beginning with a START bit is transmitted according to the features defined in the TWIHS Master Mode Register (TWIHS_MMR).

This action is necessary when the TWIHS peripheral needs to read data from a slave. When configured in Master mode with a write operation, a frame is sent as soon as the user writes a character in the Transmit Holding Register (TWIHS_THR).

STOP: Send a STOP Condition

0: No effect.

1: STOP condition is sent just after completing the current byte transmission in Master Read mode.

- In single data byte master read, both START and STOP must be set.
- In multiple data bytes master read, the STOP must be set after the last data received but one.
- In Master Read mode, if a NACK bit is received, the STOP is automatically performed.
- In master data write operation, a STOP condition will be sent after the transmission of the current data is finished.

MSEN: TWIHS Master Mode Enabled

0: No effect.

1: Enables the Master mode (MSDIS must be written to 0).

Note: Switching from Slave to Master mode is only permitted when TXCOMP = 1.

MSDIS: TWIHS Master Mode Disabled

0: No effect.

1: The Master mode is disabled, all pending data is transmitted. The shifter and holding characters (if it contains data) are transmitted in case of write operation. In read operation, the character being transferred must be completely received before disabling.

SVEN: TWIHS Slave Mode Enabled

0: No effect.

1: Enables the Slave mode (SVDIS must be written to 0).

Note: Switching from Master to Slave mode is only permitted when TXCOMP = 1.

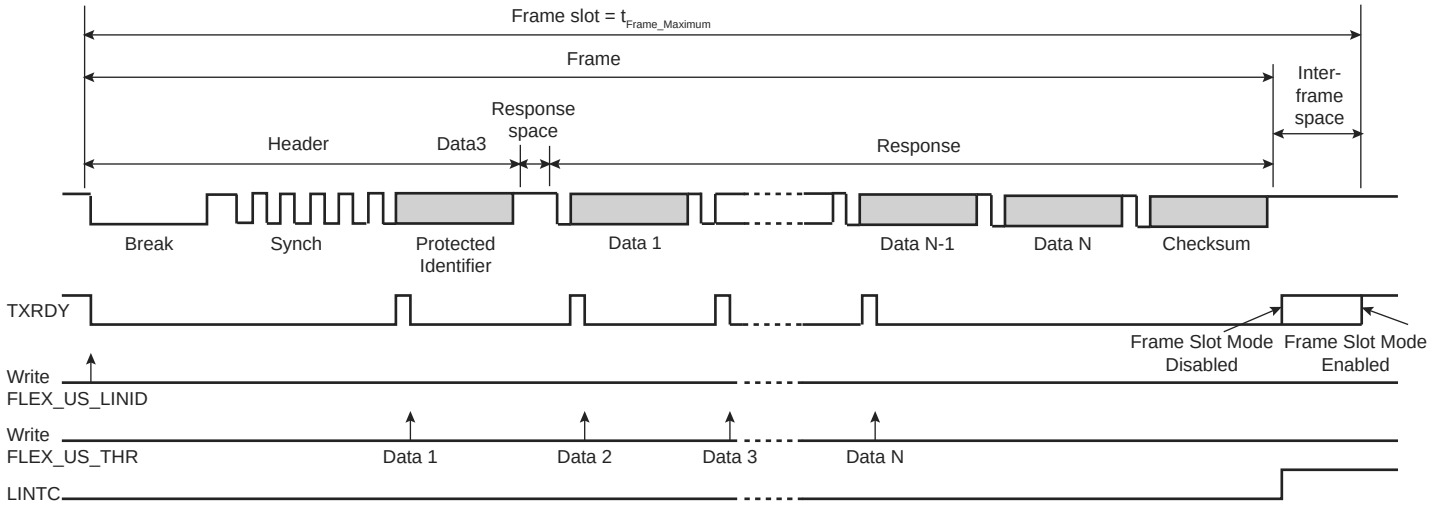
SVDIS: TWIHS Slave Mode Disabled

0: No effect.

- $t_{Frame\_Maximum} = (63 + 14 \times DLC) \times t_{bit}$

  Note 1: The term “+1” leads to an integer result for $t_{Frame\_Maximum}$ (LIN Specification 1.3).

Figure 47-46: Frame Slot Mode



47.7.9.14 LIN Errors

- Bit Error

This error is generated in master of slave node configuration, when the USART is transmitting and if the transmitted value on the Tx line is different from the value sampled on the Rx line. If a bit error is detected, the transmission is aborted at the next byte border.

This error is reported by the FLEX_US_CSR.LINBE flag.

- Inconsistent Synch Field Error

This error is generated in slave node configuration, if the Synch Field character received is other than 0x55.

This error is reported by the FLEX_US_CSR.LINISFE flag.

- Identifier Parity Error

This error is generated in slave node configuration, if the parity of the identifier is wrong. This error can be generated only if the parity feature is enabled (PARDIS = 0).

This error is reported by the FLEX_US_CSR.LINIPE flag.

- Checksum Error

This error is generated in master of slave node configuration, if the received checksum is wrong. This flag can be set to 1 only if the checksum feature is enabled (CHKDIS = 0).

This error is reported by the FLEX_US_CSR.LINCE flag.

- Slave Not Responding Error

This error is generated in master of slave node configuration, when the USART expects a response from another node (NACT = SUBSCRIBE) but no valid message appears on the bus within the time given by the maximum length of the message frame, $t_{Frame\_Maximum}$ (see **Section 47.7.9.13 “Frame Slot Mode”**). This error is disabled if the USART does not expect any message (NACT = PUBLISH or NACT = IGNORE).

This error is reported by the FLEX_US_CSR.LINSNRE.

- Synch Tolerance Error

This error is generated in slave node configuration if, after the clock synchronization procedure, it appears that the computed baud rate deviation compared to the initial baud rate is superior to the maximum tolerance FTol_Unsynch (±15%).

This error is reported by the FLEX_US_CSR.LINSTE flag.

- Header Timeout Error

This error is generated in slave node configuration, if the Header is not entirely received within the time given by the maximum length of the Header, $t_{Header\_Maximum}$.

50.7.5 QSPI Status Register

Name: QSPI_SR

Address: 0xF0020010 (0), 0xF0024010 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	QSPIENS
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	INSTRE	CSS	CSR
7	6	5	4	3	2	1	0
–	–	–	–	OVRES	TXEMPTY	TDRE	RDRF

RDRF: Receive Data Register Full (cleared by reading QSPI_RDR)

0: No data has been received since the last read of QSPI_RDR.

1: Data has been received and the received data has been transferred from the serializer to QSPI_RDR since the last read of QSPI_RDR.

TDRE: Transmit Data Register Empty (cleared by writing QSPI_TDR)

0: Data has been written to QSPI_TDR and not yet transferred to the serializer.

1: The last data written in the QSPI_TDR has been transferred to the serializer.

TDRE equals zero when the QSPI is disabled or at reset. The QSPI enable command sets this bit to one.

TXEMPTY: Transmission Registers Empty (cleared by writing QSPI_TDR)

0: As soon as data is written in QSPI_TDR.

1: QSPI_TDR and the internal shifter are empty. If a transfer delay has been defined, TXEMPTY is set after the completion of such delay.

OVRES: Overrun Error Status (cleared on read)

0: No overrun has been detected since the last read of QSPI_SR.

1: At least one overrun error has occurred since the last read of QSPI_SR.

An overrun occurs when QSPI_RDR is loaded at least twice from the serializer since the last read of the QSPI_RDR.

CSR: Chip Select Rise (cleared on read)

0: No chip select rise has been detected since the last read of QSPI_SR.

1: At least one chip select rise has been detected since the last read of QSPI_SR.

CSS: Chip Select Status

0: The chip select is asserted.

1: The chip select is not asserted.

INSTRE: Instruction End Status (cleared on read)

0: No instruction end has been detected since the last read of QSPI_SR.

1: At least one instruction end has been detected since the last read of QSPI_SR.

QSPIENS: QSPI Enable Status

0: QSPI is disabled.

1: QSPI is enabled.

50.7.11 QSPI Instruction Code Register

Name: QSPI_ICR

Address: 0xF0020034 (0), 0xF0024034 (1)

Access: Read/Write

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

23	22	21	20	19	18	17	16
OPT							

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
INST							

INST: Instruction Code

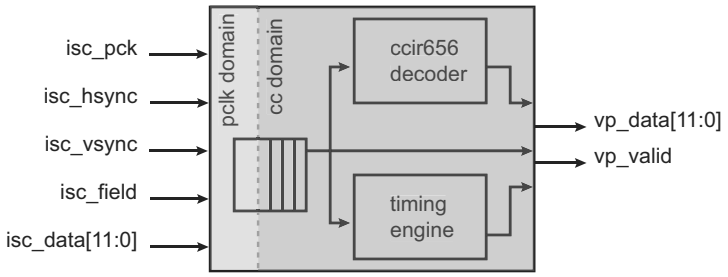
Instruction code to send to the serial Flash memory.

OPT: Option Code

Option code to send to the serial Flash memory.

52.5.5 Parallel Front End (PFE) Module

Figure 52-14: PFE Block Diagram



The Parallel Front End module performs data resampling across clock domain boundary. It includes a CCIR656 decoder used to convert a standard ITU-R BT.656 stream to 24-bit digital video. It also generates pixels, syncs flags and valid signals to the main video pipeline. It ouputs field, video and synchronization signals. The PFE can optionally crop and limit the incoming pixel stream to a predefined horizontal and vertical value. By default, the PFE only relies on the cmos sensor horizontal and vertical reference to sample the incoming pixel stream. A pixel is sampled if, and only if, the vertical and horizontal synchronizations are valid and a pixel clock edge is detected. ISC_PFE_CFG0.BPS shows the number of bits per sample. The PFE module outputs a 12-bit data on the vp_data[11:0] bus, and asserts the vp_valid signal when the data can be sampled.

PFE VP_DATA Mapping	Raw Bayer 12-bit	Raw Bayer 10-bit	YUV422 8-bit	YUV422 10-bit	Mono 12-bit
VP_DATA[11]	RGGB[11]	RGGB[9]	YC422[7]	YC422[9]	Y[11]
VP_DATA[10]	RGGB[10]	RGGB[8]	YC422[6]	YC422[8]	Y[10]
VP_DATA[9]	RGGB[9]	RGGB[7]	YC422[5]	YC422[7]	Y[9]
VP_DATA[8]	RGGB[8]	RGGB[6]	YC422[4]	YC422[6]	Y[8]
VP_DATA[7]	RGGB[7]	RGGB[5]	YC422[3]	YC422[5]	Y[7]
VP_DATA[6]	RGGB[6]	RGGB[4]	YC422[2]	YC422[4]	Y[6]
VP_DATA[5]	RGGB[5]	RGGB[3]	YC422[1]	YC422[3]	Y[5]
VP_DATA[4]	RGGB[4]	RGGB[2]	YC422[0]	YC422[2]	Y[4]
VP_DATA[3]	RGGB[3]	RGGB[1]	YC422[7] or 0	YC422[1]	Y[3]
VP_DATA[2]	RGGB[2]	RGGB[0]	YC422[6] or 0	YC422[0]	Y[2]
VP_DATA[1]	RGGB[1]	RGGB[9] or 0	YC422[5] or 0	YC422[9] or 0	Y[1]
VP_DATA[0]	RGGB[0]	RGGB[8] or 0	YC422[4] or 0	YC422[8] or 0	Y[0]

Note: When ISC_PFE_CFG0.REP is set, missing VP_DATA LSBs are replaced with replicated LSBs of the incoming stream, otherwise they are forced to zero.

The PFE module also includes logic to synchronize capture request with the incoming pixel stream. Two operating modes are available: Single Shot and Continuous Acquisition. When the ISC_PFE_CFG0.CONT field is cleared, the ISC transfers a single image to memory,

52.6.32 ISC Gamma Correction Green Entry Register

Name: ISC_GAM_GENTRYx[x=0..63]

Address: 0xF0008198

Access: Read/Write

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

23	22	21	20	19	18	17	16
GCONSTANT							

15	14	13	12	11	10	9	8
–	–	–	–	–	–	GSLOPE	

7	6	5	4	3	2	1	0
GSLOPE							

GSLOPE: Green Color Slope for Piecewise Interpolation (signed 10 bits 1:3:6)

GCONSTANT: Green Color Constant for Piecewise Interpolation (unsigned 10 bits 0:10:0)

56.7.38 PWM Comparison x Mode Register

Name: PWM_CMPMx

Address: 0xF802C138 [0], 0xF802C148 [1], 0xF802C158 [2], 0xF802C168 [3], 0xF802C178 [4], 0xF802C188 [5], 0xF802C198 [6], 0xF802C1A8 [7]

Access: Read/Write

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

23	22	21	20	19	18	17	16
CUPRCNT				CUPR			

15	14	13	12	11	10	9	8
CPRCNT				CPR			

7	6	5	4	3	2	1	0
CTR				–	–	–	CEN

CEN: Comparison x Enable

0: The comparison x is disabled and can not match.

1: The comparison x is enabled and can match.

CTR: Comparison x Trigger

The comparison x is performed when the value of the comparison x period counter (CPRCNT) reaches the value defined by CTR.

CPR: Comparison x Period

CPR defines the maximum value of the comparison x period counter (CPRCNT). The comparison x value is performed periodically once every CPR+1 periods of the channel 0 counter.

CPRCNT: Comparison x Period Counter

Reports the value of the comparison x period counter.

Note: The field CPRCNT is read-only

CUPR: Comparison x Update Period

Defines the time between each update of the comparison x mode and the comparison x value. This time is equal to CUPR+1 periods of the channel 0 counter.

CUPRCNT: Comparison x Update Period Counter

Reports the value of the comparison x update period counter.

Note: The field CUPRCNT is read-only

60.3 Product Dependencies

60.3.1 Power Management

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

60.3.2 Interrupt Sources

The AES interface has an interrupt line connected to the Interrupt Controller.

Handling the AES interrupt requires programming the Interrupt Controller before configuring the AES.

Table 60-1: Peripheral IDs

Instance	ID
AES	9

60.4 Functional Description

The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information.

Encryption converts data to an unintelligible form called ciphertext. Decrypting the ciphertext converts the data back into its original form, called plaintext. The CIPHER bit in the AES Mode register (AES_MR) allows selection between the encryption and the decryption processes.

The AES is capable of using cryptographic keys of 128/192/256 bits to encrypt and decrypt data in blocks of 128 bits. This 128-bit/192-bit/256-bit key is defined in the user interface AES_KEYWRx register.

The input to the encryption processes of the CBC, CFB, and OFB modes includes, in addition to the plaintext, a 128-bit data block called the initialization vector (IV), which must be set in AES_IVRx. The initialization vector is used in an initial step in the encryption of a message and in the corresponding decryption of the message. AES_IVRx are also used by the CTR mode to set the counter value.

60.4.1 AES Register Endianness

In ARM processor-based products, the system bus and processors manipulate data in little-endian form. The AES interface requires little-endian format words. However, in accordance with the protocol of the FIPS 197 specification, data is collected, processed and stored by the AES algorithm in big-endian form.

The following example illustrates how to configure the AES:

If the first 64 bits of a message (according to FIPS 197, i.e., big-endian format) to be processed is 0xcafedeca_01234567, then AES_IDATAR0 and AES_IDATAR1 registers must be written with the following pattern:

- AES_IDATAR0 = 0xcadefeca
- AES_IDATAR1 = 0x67452301

Table 66-5: DC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
R_{PULLUP}	Pull-up Resistor	GPIO_CLK, GPIO_IO, GPIO: 1.8V	80	143	310	kΩ
		GPIO_CLK, GPIO_IO, GPIO: 3.3V	40	66	130	
$R_{PULLDOWN}$	Pull-down Resistor	GPIO_CLK, GPIO_IO, GPIO: 1.8V	80	161	430	kΩ
		GPIO_CLK, GPIO_IO, GPIO: 3.3V	40	77	160	
R_{PULLUP}	Pull-up Resistor	GPIO_AD: 1.8V	280	380	480	kΩ
		GPIO_AD: 3.3V	280	380	480	
$R_{PULLDOWN}$	Pull-down Resistor	GPIO_AD: 1.8V	280	380	480	kΩ
		GPIO_AD: 3.3V	280	380	480	
R_{SERIAL}	Serial Resistor	GPIO	–	30	–	Ω
		GPIO_IO	–	13	–	Ω
		GPIO_CLK, GPIO_AD	–	0	–	Ω

Notes: 1. V_{DDIO} voltage must be equal to V_{DDIN} voltage.

2. Current injection may lead to performance degradation or functional failures.

Table 66-6: I/O Switching Frequency

GPIO Type	Drive	VDD			Unit
	C_{Load} = 30pF	1.8V	2.5V	3.3V	
GPIO_IO	Low	8	12	15	MHz
	Medium	60	80	90	
	High	80	110	110	
GPIO_CLK	Low	10	15	18	
	Medium	90	100	120	
	High	–	–	–	
GPIO_AD	Low	10	15	18	
	Medium	90	100	120	
	High	–	–	–	
GPIO	Low	7	10	12	
	Medium	40	50	60	
	High	50	60	70	

Table 66-7: QSPI I/O Switching Frequency

GPIO Type	Description	Name	Conditions	Min	Max	Unit
GPIO_QSPI	Maximum output frequency	f_{max}	Load = 30 pF	–	133	MHz
	Output duty cycle	–	Load = 30 pF	45	55	%

68. Schematic Checklist

The schematic checklist provides the user with the requirements regarding the different pin connections that must be considered before starting any new board design as well as information on the minimum hardware resources required to quickly develop an application with the SAMA5D2. It does not consider PCB layout constraints.

It also provides recommendations regarding low-power design constraints to minimize power consumption.

This information is not intended to be exhaustive. Its objective is to cover as many configurations of use as possible.

The checklist contains a column for use by designers, making it easy to track and verify each line item.

72. Revision History

Table 72-1: SAMA5D2 Datasheet DS60001476 Rev. B Revision History

Issue Date	Changes
Jul-2017	Section 6. “Package and Pinout” Table 6-2 Pin Description: added notes (2) and (3). Table 6-3 Pin Description (SAMA5D23 pins different from those in Table 6-2 “Pin Description”): added note (2). Table 6-4 Pin Description (SAMA5D28B/C pins different from those in Table 6-2 “Pin Description”): added note (2).
	Section 7. “Power Considerations” Updated Figure 7-1 “Recommended Powerup Sequence” and Table 7-2 “Powerup Timing Specification”.
	Section 16. “Standard Boot Strategies” Updated Section 16.4.7.3 “SDCard / e.MMC Boot”. Reworked Section 16.4.7.6 “QSPI NOR Flash Boot for MRL C”.
	Section 29. “Peripheral Touch Controller (PTC)” - Replaced "Peripheral Touch Controller" with "Peripheral Touch Controller Subsystem". - Changed "PTC_IRQ_EVT" to "PTC_IRQ". - Replaced "PIO" and "IO" with "GPIO". Updated: - Figure 29-1. "PTC Block Diagram" - Section 29.2 “Embedded Characteristics” - Section 29.5.2 “I/O Lines” - Section 29.5.3 “Interrupt Sources” - Section 29.6.2.3 “Firmware in SRAM Code Area” Section 29.5.1 “Power Management”: updated description of “SLCK”. Section 29.6.3.1 “PTC Digital Controller Operations”: modified Sensing mode description. Renamed and reworked Section 29.7.1 “PTC Command Register” and Section 29.7.2 “PTC Interrupt Status Register”. Section 29.7.3 “PTC Enable Register”: removed field CLR_IRQEN, and replaced field SET_IRQEN with bits IER0, IER1, IER2, IER3. Renamed the following registers: - “Mode 1: Write Access” to “PTC Command Register” - “Host Flags” to “PTC Interrupt Status Register” - “Host Flags Control” to “PTC Enable Register”
	Section 39. “LCD Controller (LCDC)” Corrected Figure 39-1. Block Diagram (added “PP Layer” block).
cont’d on next page	