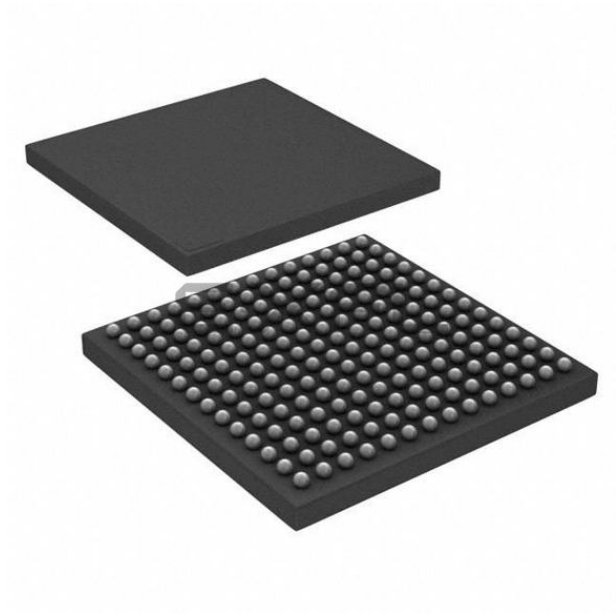


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Active
Core Processor	ARM® Cortex®-A5
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	500MHz
Co-Processors/DSP	Multimedia; NEON™ MPE
RAM Controllers	LPDDR1, LPDDR2, LPDDR3, DDR2, DDR3, DDR3L, QSPI
Graphics Acceleration	Yes
Display & Interface Controllers	Keyboard, LCD, Touchscreen
Ethernet	10/100Mbps (1)
SATA	-
USB	USB 2.0 + HSIC
Voltage - I/O	3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	ARM TZ, Boot Security, Cryptography, RTIC, Secure Fusebox, Secure JTAG, Secure Memory, Secure RTC
Package / Case	196-TFBGA, CSBGA
Supplier Device Package	196-TFBGA (11x11)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d22c-cur

14.5.27 L2CC Debug Control Register

Name: L2CC_DCR

Address: 0x00A00F40

Access: Read/Write in Secure mode

Read-only in Non-secure mode

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
–	–	–	–	–	SPNIDEN	DWB	DCL

DCL: Disable Cache Linefill

0: Enables cache linefills. This is the default value.

1: Disables cache linefills.

DWB: Disable Write-back, Force Write-through

0: Enables write-back behavior. This is the default value.

1: Forces write-through behavior.

SPNIDEN: SPNIDEN Value

Reads value of the SPNIDEN input.

16.4.8 Hardware and Software Constraints

The table below provides clock frequencies configured by the ROM code during boot.

Table 16-6: Clock Frequencies during External Memory Boot Sequence

Clock	MRL A	MRL B	MRL C
PLLA	792 MHz	792 MHz	756 MHz
PCK	396 MHz	396 MHz	378 MHz
MCK	132 MHz	132 MHz	126 MHz
SDMMC (init/operational)	400 kHz / 25 MHz	400 kHz / 25 MHz	400 kHz / 25 MHz
SPI	6 MHz	12 MHz	12 MHz
QSPI	25 MHz	50 MHz	50 MHz

The NVM drivers use several PIOs in Peripheral mode to communicate with external memory devices. Care must be taken when these PIOs are used by the application. The connected devices could be unintentionally driven at boot time, and thus electrical conflicts between the output pins used by the NVM drivers and the connected devices could occur.

To ensure the correct functionality, it is recommended to plug in critical devices to other pins not used by the NVM.

Table 16-7 contains a list of pins that are driven during the boot program execution. These pins are driven during the boot sequence for a period of less than 1 second if no correct boot program is found. For MRL C parts only, the drive strength of some I/O pins is set to 'medium' while the pins are used in peripheral mode by the ROM code. For MRL A and B, drive strength is always low.

Before performing the jump to the application in the internal SRAM, all the PIOs and peripherals used in the boot program are set to their reset state.

When the glitch and/or debouncing filter is enabled, it does not modify the behavior of the inputs on the peripherals. It acts only on the value read in PIO_PDSRx and on the input change interrupt detection. The glitch and debouncing filters require that the PIO Controller clock is enabled.

Figure 34-4: Input Glitch Filter Timing

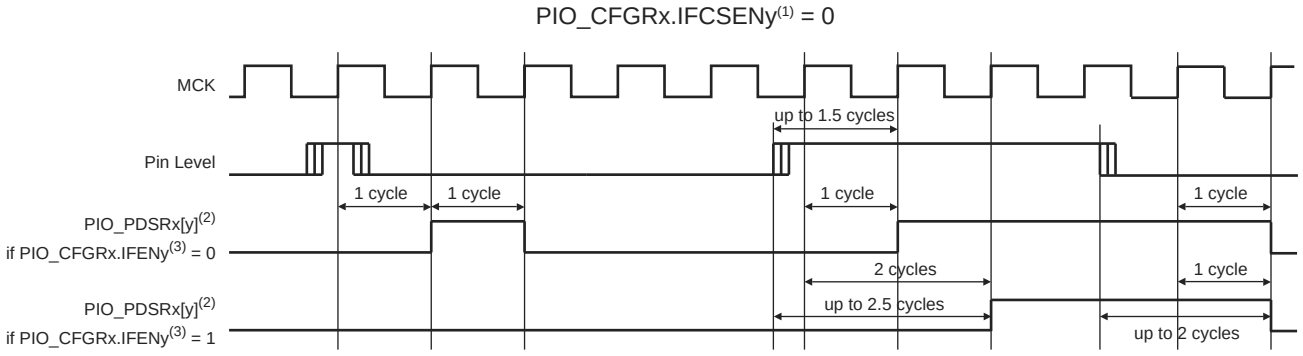
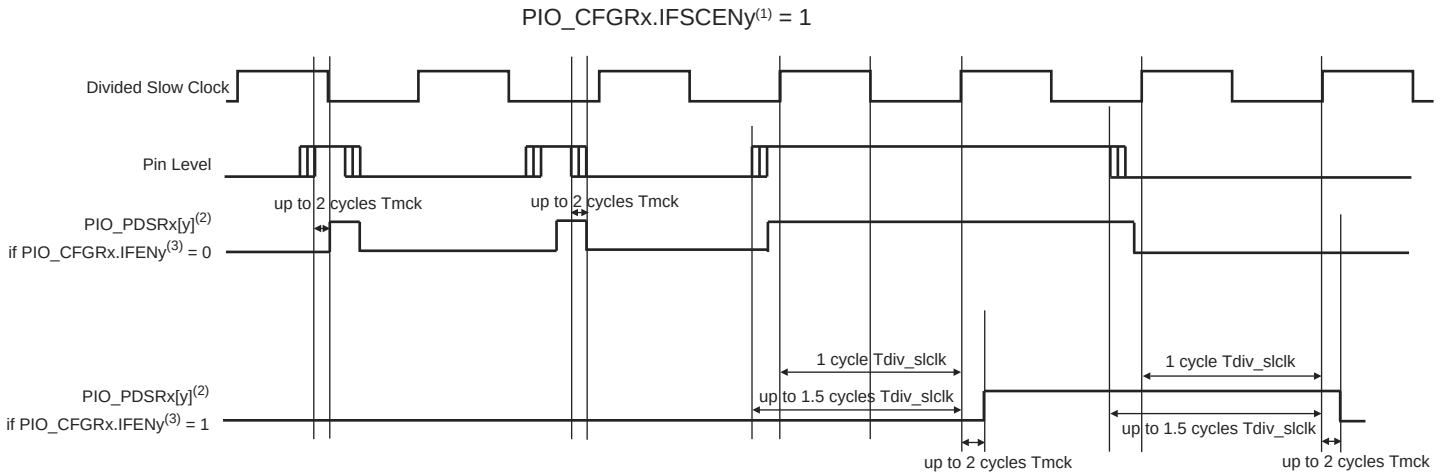


Figure 34-5: Input Debouncing Filter Timing



Note 1: Means IFCSEN bit of the I/O line y of the I/O group x

2: Means PIO Data Status value of the I/O line y of the I/O group x

3: Means IFEN bit of the I/O line y of the I/O group x

34.5.10 Input Edge/Level Interrupt

Each I/O group can be programmed to generate an interrupt when it detects an edge or a level on an I/O line. The Input Edge/Level interrupts are controlled by writing the PIO Interrupt Enable Register (PIO_IERx) and the PIO Interrupt Disable Register (PIO_IDRx), which enable and disable the input change interrupt respectively by setting and clearing the corresponding bit in the PIO Interrupt Mask Register (PIO_IMRx). For the Secure I/O lines, the Input Edge/Level interrupts are controlled by writing S_PIO_IERx and S_PIO_IDRx, which enable and disable input change interrupts respectively by setting and clearing the corresponding bit in the S_PIO_IMRx. As input change detection is possible only by comparing two successive samplings of the input of the I/O line, the PIO Controller clock must be enabled. The Input Change interrupt is available regardless of the configuration of the I/O line, i.e., configured as an input only, controlled by the PIO Controller or assigned to a peripheral function.

Each I/O group can generate a Non-Secure interrupt and a Secure interrupt according to the security level of the I/O line which triggers the interrupt.

According to the EVTSEL field value in the PIO Configuration Register (PIO_CFGRx) or the Secure PIO Configuration Register (S_PIO_CFGRx) in case of a Secure I/O line, the interrupt signal of the I/O group x can be generated on the following occurrence:

- (S_)PIO_CFGRx.EVTSELy = 0: The interrupt signal of the I/O group x is generated on the I/O line y falling edge detection (assuming that (S_)PIO_IMRx[y] = 1).

36.7.14 MPDDRC OCMS KEY1 Register

Name: MPDDRC_OCMS_KEY1

Address: 0xF000C03C

Access: Write once

31	30	29	28	27	26	25	24
KEY1							

23	22	21	20	19	18	17	16
KEY1							

15	14	13	12	11	10	9	8
KEY1							

7	6	5	4	3	2	1	0
KEY1							

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

KEY1: Off-chip Memory Scrambling (OCMS) Key Part 1

When Off-chip Memory Scrambling is enabled, the data scrambling depends on KEY1 and KEY2 values.

39. LCD Controller (LCDC)

39.1 Description

The LCD Controller (LCDC) consists of logic for transferring LCD image data from an external display buffer to an LCD module. The LCD has one display input buffer per overlay that fetches pixels through the dual AHB master interface and a lookup table to allow palletized display configurations. The LCD controller is programmable on a per overlay basis, and supports different LCD resolutions, window sizes, image formats and pixel depths.

The LCD is connected to the ARM Advanced High Performance Bus (AHB) as a master for reading pixel data. It also integrates an APB interface to configure its registers.

39.2 Embedded Characteristics

- Dual AHB Master Interface
- Supports Single Scan Active TFT Display
- Supports 12-, 16-, 18- and 24-bit Output Mode through the Spatial Dithering Unit
- Asynchronous Output Mode Supported
- 1, 2, 4, 8 bits per Pixel (Palletized)
- 12, 16, 18, 19, 24, 25 and 32 bits per Pixel (Non-palletized)
- Supports One Base Layer (Background)
- Supports One Overlay 1 Layer Window
- Supports One Overlay 2 Layer Window
- Supports One High-End Overlay (HEO) Window
- Little Endian Memory Organization
- Programmable Timing Engine, with Integer Clock Divider
- Programmable Polarity for Data, Line Synchro and Frame Synchro
- Up to 1024x768 (XGA) with Overlay (Application-Dependent). Still Image up to WXGA.
- Color Lookup Table with up to 256 Entries and Predefined 8-bit Alpha
- Programmable Negative and Positive Row Striding for all Layers
- Programmable Negative and Positive Pixel Striding for Layers
- High-End Overlay supports 4:2:0 Planar Mode and Semiplanar Mode
- High-End Overlay supports 4:2:2 Planar Mode, Semiplanar Mode and Packed
- High-End Overlay includes Chroma Upsampling Unit
- Horizontal and Vertical Rescaling Unit with Edge Interpolation and Independent Non-Integer Ratio, up to 1024x768
- Hidden Layer Removal supported
- Integrates Fully Programmable Color Space Conversion
- Blender Function Supports Arbitrary 8-bit Alpha Value and Chroma Keying
- DMA User Interface uses Linked List Structure and Add-to-queue Structure

39.7.122 High-End Overlay Configuration Register 27

Name: LCDC_HEOCFG27

Address: 0xF00003F8

Access: Read/Write

31	30	29	28	27	26	25	24
XPHI5COEFF3							

23	22	21	20	19	18	17	16
XPHI5COEFF2							

15	14	13	12	11	10	9	8
XPHI5COEFF1							

7	6	5	4	3	2	1	0
XPHI5COEFF0							

XPHI5COEFF0: Horizontal Coefficient for phase 5 tap 0

Coefficient format is 1 sign bit and 7 fractional bits.

XPHI5COEFF1: Horizontal Coefficient for phase 5 tap 1

Coefficient format is 1 sign bit and 7 fractional bits.

XPHI5COEFF2: Horizontal Coefficient for phase 5 tap 2

Coefficient format is 1 magnitude bit and 7 fractional bits.

XPHI5COEFF3: Horizontal Coefficient for phase 5 tap 3

Coefficient format is 1 sign bit and 7 fractional bits.

39.7.141 Post Processing Interrupt Disable Register

Name: LCDC_PPIDR

Address: 0xF0000550

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
–	–	DONE	ADD	DSCR	DMA	–	–

DMA: End of DMA Transfer Interrupt Disable

0: No effect

1: Interrupt source is disabled

DSCR: Descriptor Loaded Interrupt Disable

0: No effect

1: Interrupt source is disabled

ADD: Head Descriptor Loaded Interrupt Disable

0: No effect

1: Interrupt source is disabled

DONE: End of List Interrupt Disable

0: No effect

1: Interrupt source is disabled

COMPB: Index of Screening Type 2 Compare Word 0/Word 1 register x

COMPB is a pointer to the compare registers GMAC_ST2CW0x and GMAC_ST2CW1x. When COMPBE is set, the compare is true if the data at the frame offset ANDed with the value MASKVAL is equal to the value of COMPVAL ANDed with the value of MASKVAL.

COMPBE: Compare B Enable

0: Comparison via the register designated by index COMPB is disabled.

1: Comparison via the register designated by index COMPB is enabled.

COMPC: Index of Screening Type 2 Compare Word 0/Word 1 register x

COMPC is a pointer to the compare registers GMAC_ST2CW0x and GMAC_ST2CW1x. When COMPCE is set, the compare is true if the data at the frame offset ANDed with the value MASKVAL is equal to the value of COMPVAL ANDed with the value of MASKVAL.

COMPCE: Compare C Enable

0: Comparison via the register designated by index COMPC is disabled.

1: Comparison via the register designated by index COMPC is enabled.

Figure 47-22: Timeguard Operations

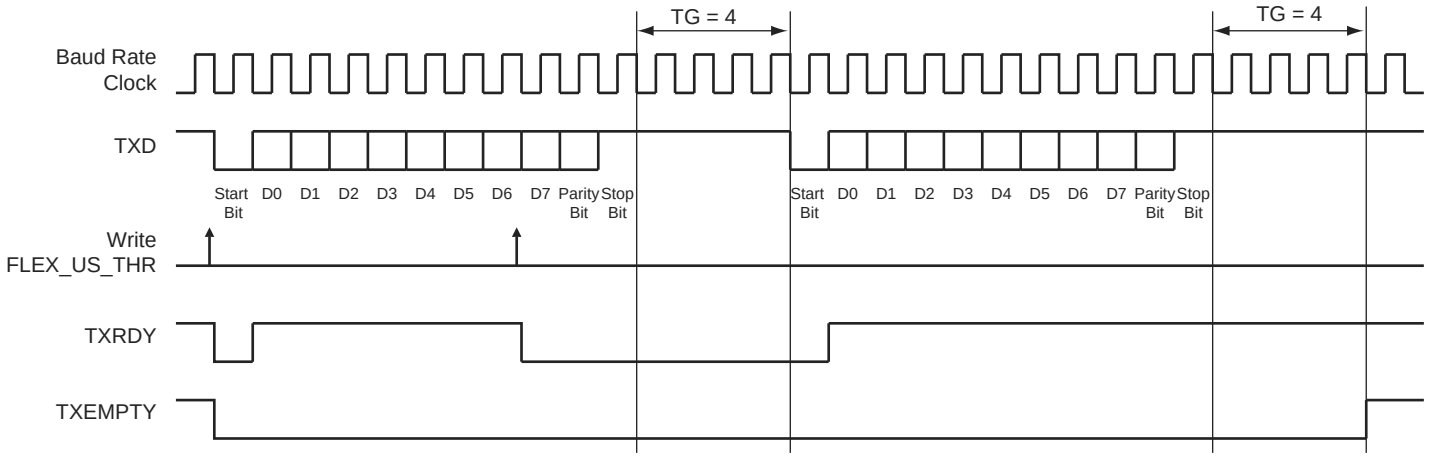


Table 47-10 indicates the maximum length of a timeguard period that the transmitter can handle in relation to the function of the baud rate.

Table 47-10: Maximum Timeguard Length Depending on Baud Rate

Baud Rate (bit/s)	Bit Time (µs)	Timeguard (ms)
1,200	833	212.50
9,600	104	26.56
14,400	69.4	17.71
19,200	52.1	13.28
28,800	34.7	8.85
38,400	26	6.63
56,000	17.9	4.55
57,600	17.4	4.43
115,200	8.7	2.21

47.7.3.11 Receiver Timeout

The Receiver Timeout provides support in handling variable-length frames. This feature detects an idle condition on the RXD line. When a timeout is detected, the FLEX_US_CSR.TIMEOUT bit rises and can generate an interrupt, thus indicating to the driver an end of frame.

The timeout delay period (during which the receiver waits for a new character) is programmed in the TO field of the Receiver Timeout Register (FLEX_US_RTOR). If the TO field is written to 0, the Receiver Timeout is disabled and no timeout is detected. The FLEX_US_CSR.TIMEOUT bit remains at 0. Otherwise, the receiver loads a 16-bit counter with the value programmed in TO. This counter is decremented at each bit period and reloaded each time a new character is received. If the counter reaches 0, the FLEX_US_CSR.TIMEOUT bit rises. Then, the user can either:

- Stop the counter clock until a new character is received. This is performed by writing a ‘1’ to FLEX_US_CR.STTTO. In this case, the idle state on RXD before a new character is received does not provide a timeout. This prevents having to handle an interrupt before a character is received and enables waiting for the next idle state on RXD after a frame is received.
- Obtain an interrupt while no character is received. This is performed by writing a ‘1’ to FLEX_US_CR.RETTO. In this case, the counter starts counting down immediately from the value TO. This generates a periodic interrupt so that a user timeout can be handled, for example when no key is pressed on a keyboard.

Figure 47-23 shows the block diagram of the Receiver Timeout feature.

Figure 47-83: START and STOP Conditions

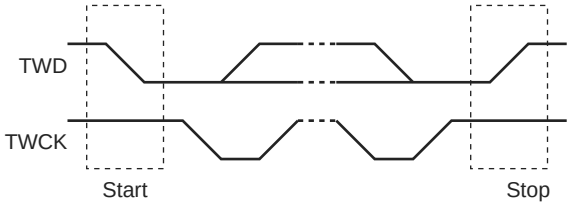
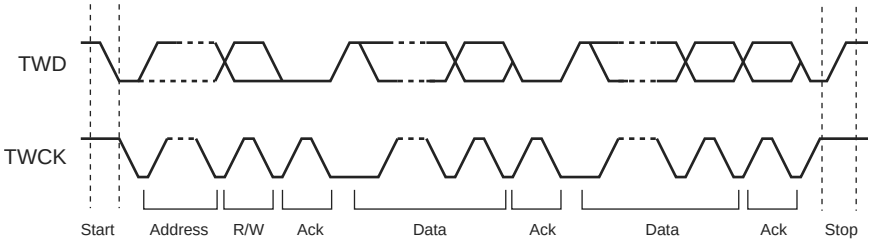


Figure 47-84: Transfer Format



47.9.2 Modes of Operation

The TWI has different modes of operation:

- Master Transmitter mode (Standard and Fast modes only)
- Master Receiver mode (Standard and Fast modes only)
- Multi-master Transmitter mode (Standard and Fast modes only)
- Multi-master Receiver mode (Standard and Fast modes only)
- Slave Transmitter mode (Standard, Fast and High-speed modes)
- Slave Receiver mode (Standard, Fast and High-speed modes)

These modes are described in the following sections.

47.9.3 Master Mode

47.9.3.1 Definition

The master is the device that starts a transfer, generates a clock and stops it. This operating mode is not available if High-speed mode is selected.

47.9.3.2 Programming Master Mode

The following fields must be programmed before entering Master mode:

1. DADR (+ IADRSZ + IADR if a 10-bit device is addressed): The device address is used to access slave devices in Read or Write mode.
2. CWGR + CKDIV + CHDIV + CLDIV: Clock waveform.
3. SVDIS: Disables Slave mode.
4. MSEN: Enables Master mode.

Note: If the TWI is already in Master mode, the device address (DADR) can be configured without disabling the Master mode.

47.9.3.3 Transfer Speed/Bit Rate

The TWI speed is defined in FLEX_TWI_CWGR. The TWI bit rate can be based either on the peripheral clock if the BRSRCCLK bit value is 0 or on a programmable clock source provided by the GCLK if the BRSRCCLK bit value is 1.

If BRSRCCLK = 1, the bit rate is independent of the processor/peripheral clock and thus processor/peripheral clock frequency can be changed without affecting the TWI transfer rate.

The GCLK frequency must be at least three times lower than the peripheral clock frequency.

47.9.3.4 Master Transmitter Mode

This operating mode is not available if High-speed mode is selected.

47.10.85 TWI Write Protection Mode Register

Name: FLEX_TWI_WPMR

Address: 0xF80346E4 (0), 0xF80386E4 (1), 0xFC0106E4 (2), 0xFC0146E4 (3), 0xFC0186E4 (4)

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 0x545749 (“TWI” in ASCII).

1: Enables the write protection if WPKEY corresponds to 0x545749 (“TWI” in ASCII).

See Section 47.9.8 ”TWI Register Write Protection” for the list of registers that can be write-protected.

WPKEY: Write Protection Key

Value	Name	Description
0x545749	PASSWD	Writing any other value in this field aborts the write operation of bit WPEN. Always reads as 0

49.8.2 SPI Mode Register

Name: SPI_MR

Address: 0xF8000004 (0), 0xFC000004 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
DLYBCS							
23	22	21	20	19	18	17	16
–	–	–	–	PCS			
15	14	13	12	11	10	9	8
–	–	–	CMPMODE	–	–	–	LSBHALF
7	6	5	4	3	2	1	0
LLB	–	WDRBT	MODFDIS	BRSRCCLK	PCSDEC	PS	MSTR

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

MSTR: Master/Slave Mode

0: SPI is in Slave mode

1: SPI is in Master mode

PS: Peripheral Select

0: Fixed Peripheral Select

1: Variable Peripheral Select

PCSDEC: Chip Select Decode

0: The chip select lines are directly connected to a peripheral device.

1: The four NPCS chip select lines are connected to a 4-bit to 16-bit decoder.

When PCSDEC = 1, up to 15 chip select signals can be generated with the four NPCS lines using an external 4-bit to 16-bit decoder. The chip select registers define the characteristics of the 15 chip selects, with the following rules:

SPI_CSR0 defines peripheral chip select signals 0 to 3.

SPI_CSR1 defines peripheral chip select signals 4 to 7.

SPI_CSR2 defines peripheral chip select signals 8 to 11.

SPI_CSR3 defines peripheral chip select signals 12 to 14.

BRSRCCLK: Bit Rate Source Clock

0 (PERIPH_CLK): The peripheral clock is the source clock for the bit rate generation.

1 (GCLK): PMC GCLK is the source clock for the bit rate generation, thus the bit rate can be independent of the core/peripheral clock.

Note: If bit BRSRCCLK = 1, the SCBR field in SPI_CSRx must be programmed with a value greater than 1.

MODFDIS: Mode Fault Detection

0: Mode fault detection enabled

1: Mode fault detection disabled

WDRBT: Wait Data Read Before Transfer

0: No Effect. In Master mode, a transfer can be initiated regardless of SPI_RDR state.

1: In Master mode, a transfer can start only if SPI_RDR is empty, i.e., does not contain any unread data. This mode prevents overrun error in reception.

LLB: Local Loopback Enable

0: Local loopback path disabled.

1: Local loopback path enabled.

50.5 Product Dependencies

50.5.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the QSPI pins to their peripheral functions.

Table 50-2: I/O Lines

Instance	Signal	I/O Line	Peripheral
QSPI0	QSPI0_CS	PA1	B
QSPI0	QSPI0_CS	PA15	C
QSPI0	QSPI0_CS	PA23	F
QSPI0	QSPI0_IO0	PA2	B
QSPI0	QSPI0_IO0	PA16	C
QSPI0	QSPI0_IO0	PA24	F
QSPI0	QSPI0_IO1	PA3	B
QSPI0	QSPI0_IO1	PA17	C
QSPI0	QSPI0_IO1	PA25	F
QSPI0	QSPI0_IO2	PA4	B
QSPI0	QSPI0_IO2	PA18	C
QSPI0	QSPI0_IO2	PA26	F
QSPI0	QSPI0_IO3	PA5	B
QSPI0	QSPI0_IO3	PA19	C
QSPI0	QSPI0_IO3	PA27	F
QSPI0	QSPI0_SCK	PA0	B
QSPI0	QSPI0_SCK	PA14	C
QSPI0	QSPI0_SCK	PA22	F
QSPI1	QSPI1_CS	PA11	B
QSPI1	QSPI1_CS	PB6	D
QSPI1	QSPI1_CS	PB15	E
QSPI1	QSPI1_IO0	PA7	B
QSPI1	QSPI1_IO0	PB7	D
QSPI1	QSPI1_IO0	PB16	E
QSPI1	QSPI1_IO1	PA8	B
QSPI1	QSPI1_IO1	PB8	D
QSPI1	QSPI1_IO1	PB17	E
QSPI1	QSPI1_IO2	PA9	B
QSPI1	QSPI1_IO2	PB9	D
QSPI1	QSPI1_IO2	PB18	E
QSPI1	QSPI1_IO3	PA10	B
QSPI1	QSPI1_IO3	PB10	D

- When a read transfer is stopped at the block gap initiated by a Stop At Block Gap Request (STPBGR).

The SDMMC stops a read operation at the start of the interrupt cycle by driving the Read Wait (DAT[2] line) or by stopping the SD Clock. If the Read Wait signal is already driven (due to the fact that the data buffer cannot receive data), the SDMMC can continue to stop the read operation by driving the Read Wait signal. It is necessary to support the Read Wait in order to use the Suspend/ Resume operation.

In the case of write transactions:

This status indicates that a write transfer is executing on the bus. A change from 1 to 0 raises the Transfer Complete (TRFC) status flag in SDMMC_NISTR if SDMMC_NISTER.TRFC is set to 1. An interrupt is generated if SDMMC_NISIER.TRFC is set to 1. Refer to section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for details on timing.

This bit is set in either of the following cases:

- After the end bit of the write command.

- When writing 1 to SDMMC_BGCR.CONTR (Continue Request) to continue a write transfer.

This bit is cleared in either of the following cases:

- When the card releases Write Busy of the last data block. If the card does not drive a Busy signal for 8 SDCLK, the SDMMC considers the card drive “Not Busy”. In the case of ADMA2, the last block is designated by the last transfer of the Descriptor Table.

- When the card releases Write Busy prior to wait for write transfer as a result of a Stop At Block Gap Request (STPBGR).

Command with Busy:

This status indicates whether a command that indicates Busy (ex. erase command for memory) is executing on the bus. This bit is set to 1 after the end bit of the command with Busy and cleared when Busy is deasserted. A change from 1 to 0 raises the Transfer Complete (TRFC) status flag in SDMMC_NISTR if SDMMC_NISTER.TRFC is set to 1. An interrupt is generated if SDMMC_NISIER.TRFC is set to 1. Refer to Figures 2.11 to 2.13 in the “SD Host Controller Simplified Specification V3.00” .

0: DAT line inactive.

1: DAT line active.

WTACT: Write Transfer Active

This bit indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the SDMMC. Refer to section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.

This bit is set to 1 in either of the following conditions:

- After the end bit of the write command.

- When a write operation is restarted by writing a 1 to SDMMC_BGCR.CONTR (Continue Request).

This bit is cleared to 0 in either of the following conditions:

- After getting the CRC status of the last data block as specified by the transfer count (single and multiple). In case of ADMA2, transfer count is designated by the descriptor table.

- After getting the CRC status of any block where a data transmission is about to be stopped by a Stop At Block Gap Request (STPBGR) of SDMMC_BGCR.

During a write transaction and as the result of the Stop At Block Gap Request (STPBGR) being set, a change from 1 to 0 raises the Block Gap Event (BLKGE) status flag in SDMMC_NISTR if SDMMC_NISTER.BLKGE is set to 1. An interrupt is generated if BLKGE is set to 1 in SDMMC_NISIER. This status is useful to determine whether nonDAT line commands can be issued during Write Busy.

RTACT: Read Transfer Active

This bit is used to detect completion of a read transfer. Refer to section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.

This bit is set to 1 in either of the following conditions:

- After the end bit of the read command.

- When a read operation is restarted by writing a 1 to SDMMC_BGCR.CONTR (Continue Request).

This bit is cleared to 0 in either of the following conditions:

- When the last data block as specified by Transfer Block Size (BLKSIZE) is transferred to the system.

- In case of ADMA2, end of read is designated by the descriptor table.

51.13.52 SDMMC Retuning Interrupt Status Enable Register

Name: SDMMC_RTISTER

Access: Read/Write

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	TEVT

TEVT: Retuning Timer Event

0 (MASKED): The TEVT status flag in SDMMC_RTISTR is masked.

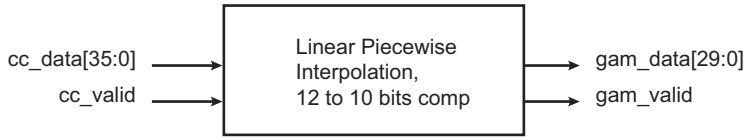
1 (ENABLED): The TEVT status flag in SDMMC_RTISTR is enabled.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} RRGAIN & RGGAIN & RBGAIN \\ GRGAIN & GGGAIN & GBGAIN \\ BRGAIN & BGGAIN & BBGAIN \end{bmatrix} \times \begin{bmatrix} cfa\_data[35:24] \\ cfa\_data[23:12] \\ cfa\_data[11:0] \end{bmatrix} + \begin{bmatrix} ROFST \\ GOFST \\ BOFST \end{bmatrix}$$

52.5.9 Gamma Curve (GAM) Module

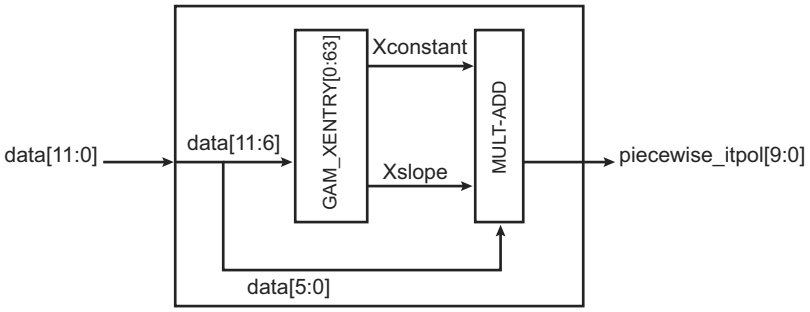
The GAM module samples the cc_data[35:0] bus when cc_valid is asserted, and generates gam_data[29:0] 30-bit width data along with the validity signal gam_valid. Imaging devices have non-linear characteristics, but the transfer function is approximated by a power function. The intensity of each of the linear RGB components is transformed to a non-linear signal through the use of the gamma correction submodule. The power function is linearly interpolated using 64 breakpoints. This also performs a 12-bit to 10-bit compression. The polynomial for the linear interpolation between breakpoints is ***i*** and ***i*** +1. Consequently, for each breakpoint, two values are required: constant and slope. The table values are programmable through the user interface when the gamma correction module is disabled (ISC_GAM_CTRL.ENABLE is cleared). ISC_GAM_RENTRY is used for Red gamma correction. ISC_GAM_GENTRY is used for Green gamma correction. ISC_GAM_BENTRY is used for Blue gamma correction. Each table entry is composed of a 10-bit (signed) slope and a 10-bit constant.

Figure 52-23: GAM Block Diagram



ISC_GAM_CTRL.ENABLE	ISC_GAM_CTRL.XLUT	GAM_DATA Slice	Value
0	0	gam_data[29:0]	cc_data[29:0]
1	0	gam_data[29:20]	cc_data[35:26]
		gam_data[19:10]	cc_data[23:14]
		gam_data[9:0]	cc_data[11:2]
1	1	gam_data[29:20]	R=piecewise_itpol(cc_data_r[35:24])
		gam_data[19:10]	G=piecewise_itpol(cc_data_r[23:12])
		gam_data[9:0]	B=piecewise_itpol(cc_data_r[11:0])

Figure 52-24: Piecewise Linear Interpolation Block Diagram



The interpolation consists of three tables that store the function values GAM_XENTRY[0:63] where X stands for R, G and B. The input of the table has six bits. It outputs a slope and a constant. The slope is later multiplied by the data lsb (6-bit) and added to a constant. The final value is the gamma-corrected value of the input. This module performs a 12-to-10 compression.

53.6 Controller Area Network (MCAN) User Interface

Table 53-14: Register Mapping

Offset	Register	Name	Access	Reset
0x00	Core Release Register	MCAN_CREL	Read-only	0xrrrddddd[1]
0x04	Endian Register	MCAN_ENDN	Read-only	0x87654321
0x08	Customer Register	MCAN_CUST	Read/Write	0
0x0C	Data Bit Timing and Prescaler Register	MCAN_DBTP	Read/Write	0x00000A33
0x10	Test Register	MCAN_TEST	Read/Write	0x000000x0[2]
0x14	RAM Watchdog Register	MCAN_RWD	Read/Write	0x00000000
0x18	CC Control Register	MCAN_CCCR	Read/Write	0x00000001
0x1C	Nominal Bit Timing and Prescaler Register	MCAN_NBTP	Read/Write	0x06000A03
0x20	Timestamp Counter Configuration Register	MCAN_TSCC	Read/Write	0x00000000
0x24	Timestamp Counter Value Register	MCAN_TSCV	Read/Write	0x00000000
0x28	Timeout Counter Configuration Register	MCAN_TOCC	Read/Write	0xFFFF0000
0x2C	Timeout Counter Value Register	MCAN_TOCV	Read/Write	0x0000FFFF
0x30–0x3C	Reserved	–	–	–
0x40	Error Counter Register	MCAN_ECR	Read-only	0x00000000
0x44	Protocol Status Register	MCAN_PSR	Read-only	0x00000707
0x48	Transmit Delay Compensation Register	MCAN_TDCR	Read/Write	0x00000000
0x4C	Reserved	–	–	–
0x50	Interrupt Register	MCAN_IR	Read/Write	0x00000000
0x54	Interrupt Enable Register	MCAN_IE	Read/Write	0x00000000
0x58	Interrupt Line Select Register	MCAN_ILS	Read/Write	0x00000000
0x5C	Interrupt Line Enable Register	MCAN_ILE	Read/Write	0x00000000
0x60–0x7C	Reserved	–	–	–
0x80	Global Filter Configuration Register	MCAN_GFC	Read/Write	0x00000000
0x84	Standard ID Filter Configuration Register	MCAN_SIDFC	Read/Write	0x00000000
0x88	Extended ID Filter Configuration Register	MCAN_XIDFC	Read/Write	0x00000000
0x8C	Reserved	–	–	–
0x90	Extended ID AND Mask Register	MCAN_XIDAM	Read/Write	0x1FFFFFFF
0x94	High Priority Message Status Register	MCAN_HPMS	Read-only	0x00000000
0x98	New Data 1 Register	MCAN_NDAT1	Read/Write	0x00000000
0x9C	New Data 2 Register	MCAN_NDAT2	Read/Write	0x00000000
0xA0	Receive FIFO 0 Configuration Register	MCAN_RXF0C	Read/Write	0x00000000
0xA4	Receive FIFO 0 Status Register	MCAN_RXF0S	Read-only	0x00000000
0xA8	Receive FIFO 0 Acknowledge Register	MCAN_RXF0A	Read/Write	0x00000000
0xAC	Receive Rx Buffer Configuration Register	MCAN_RXBC	Read/Write	0x00000000

57.5.6 SFC Status Register

Name: SFC_SR

Address: 0xF804C01C

Access: Read-only

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

PGMC: Programming Sequence Completed (cleared on read)

0: No programming sequence completion since the last read of SFC_SR.

1: At least one programming sequence completion since the last read of SFC_SR.

PGMF: Programming Sequence Failed (cleared on read)

0: No programming failure occurred during last programming sequence since the last read of SFC_SR.

1: A programming failure occurred since the last read of SFC_SR.

LCHECK: Live Integrity Checking Error (cleared on read)

0: No live integrity check error since the last read of SFC_SR.

1: At least one live integrity check error since the last read of SFC_SR.

APLE: Area Programming Lock Error (cleared on read)

0: No programming attempt has been made in the locked area since the last read of SFC_SR.

1: A programming attempt has been made in the locked area since the last read of SFC_SR.

ACE: Area Check Error (cleared on read)

0: No check error in the reserved area since the last read of SFC_SR.

1: At least one check error in the reserved area since the last read of SFC_SR.

65.6.15 Touchscreen

65.6.15.1 Touchscreen Mode

The TSMODE parameter of the ADC Touchscreen Mode register (ADC_TSMR) is used to enable/disable the touchscreen functionality, to select the type of screen (4-wire or 5-wire) and, in the case of a 4-wire screen and to activate (or not) the pressure measurement.

In 4-wire mode, channel 0, 1, 2 and 3 must not be used for classic ADC conversions. Likewise, in 5-wire mode, channel 0, 1, 2, 3, and 4 must not be used for classic ADC conversions.

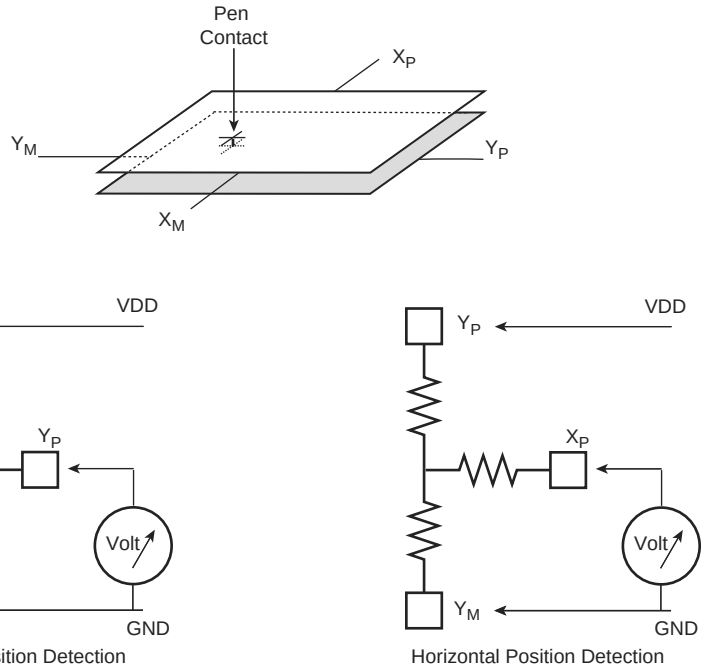
65.6.15.2 4-wire Resistive Touchscreen Principles

A resistive touchscreen is based on two resistive films, each one being fitted with a pair of electrodes, placed at the top and bottom on one film, and on the right and left on the other. In between, there is a layer acting as an insulator, but also enables contact when you press the screen. This is illustrated in Figure 65-14.

The ADC controller can perform the following tasks without external components:

- position measurement
- pressure measurement
- pen detection

Figure 65-14: Touchscreen Position Measurement



65.6.15.3 4-wire Position Measurement Method

As shown in Figure 65-14, to detect the position of a contact, a supply is first applied from top to bottom. Due to the linear resistance of the film, there is a voltage gradient from top to bottom. When a contact is performed on the screen, the voltage propagates at the point the two surfaces come into contact with the second film. If the input impedance on the right and left electrodes sense is high enough, the film does not affect this voltage, despite its resistive nature.

For the horizontal direction, the same method is used, but by applying supply from left to right. The range depends on the supply voltage and on the loss in the switches that connect to the top and bottom electrodes.

In an ideal world (linear, with no loss through switches), the horizontal position is equal to:

VY_M / VDD or VY_P / VDD.

The implementation with on-chip power switches is shown in Figure 65-15. The voltage measurement at the output of the switch compensates for the switches loss.

It is possible to correct for switch loss by performing the operation:

When TSMODE = 1 or 3, each trigger event adds two half-words in the buffer (assuming TSAV = 0), first half-word being XPOS of ADC_XPOSR then YPOS of ADC_YPOSR. If TSAV/TSFREQ ≠ 0, the data structure remains unchanged. Not all trigger events add data to the buffer.

When TSMODE = 2, each trigger event adds four half-words to the buffer (assuming TSAV = 0), first half-word being XPOS of ADC_XPOSR followed by YPOS of ADC_YPOSR and finally Z1 followed by Z2, both located in ADC_PRESSR.

When TAG is set (ADC_EMR), the CHNB field (four most significant bits of ADC_LCDR) is cleared when XPOS is transmitted and set when YPOS is transmitted, allowing an easier post-processing of the buffer or a better checking of the buffer integrity. In case 4-wire with Pressure mode is selected, Z1 value is transmitted to the buffer along with tag set to 2 and Z2 is tagged with value 3.

XSCALE and YSCALE (calibration values) are not transmitted to the buffer because they are supposed to be constant and moreover only measured at the very first startup of the controller or upon user request.

There is no change in buffer structure whatever the value of PENDET bit configuration in ADC_TSMR but it is recommended to use the pen detection function for buffer post-processing (see Section 65.6.17.4 “Pen Detection Status”).