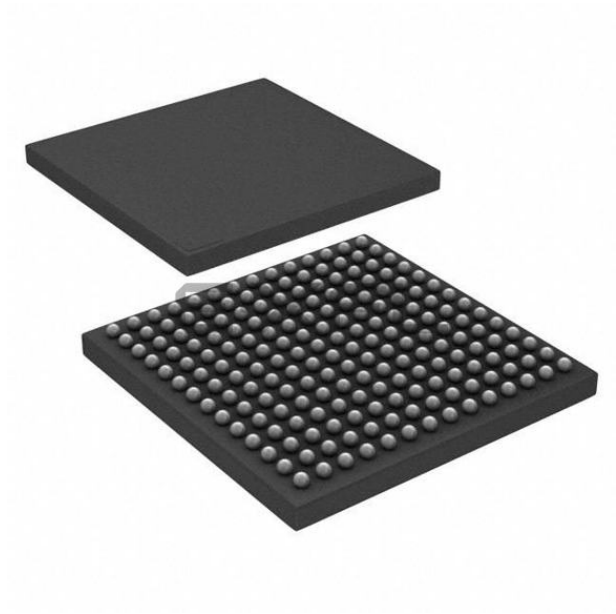


Welcome to E-XFL.COM

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-cu

Table 5-2: Security Features (Continued)

Peripheral	Function	Description	Comments
SECUMOD	JTAG	JTAG entry monitor	These tamper pins (JTAG, test, PIOBUs, monitors, etc.) can be configured to immediately erase Backup memories (BUSRAM4KB and BUREG256b), or generate an interrupt or a wakeup signal.
	Test	Test entry monitor	
	Active Shield(2)	Die Active Shield	
	Voltage Monitoring(2)	VDDBU monitoring	
		VDDCORE monitoring	
	Temperature Monitoring(2)	Temperature monitoring	
	Frequency Monitoring(2)	32.768 kHz crystal oscillator monitoring	
		CPU clock monitoring	
	IO Tamper Pin	8 tamper detection pins. Active and Dynamic modes supported.	
	Secure Backup SRAM	5 Kbytes scrambled and non-imprinting avoiding data persistance	4 Kbytes erasable on tamper detection
	Secure Backup Registers	256-bit register bank, scrambled	Erasable on tamper detection
RTC	RTC	Timestamping of tamper events. Protection against bad configuration (invalid entry for date and time are impossible)	All events are logged in the RTC. Timestamping gives the source of the reset/erase memory/interruption
		RTC robustness against glitch attack on 32 kHz crystal oscillator	–
Secure Fuse	JTAG Access Control	Disable JTAG access by fuse bit	–
	Secure Debug Disable	JTAG debug allowed in Normal mode only, not in Secure mode	TrustZone

Note 1: A PCI-certified Atmel Software Crypto Library (ASCL) is available under NDA.

2: Available on SAMA5D23 and SAMA5D28 only. For environmental monitors, refer to the document "*SAMA5D23 and SAMA5D28 Environmental Monitors*" (document no. 44036), available under Non-Disclosure Agreement (NDA). Contact a Microchip sales representative for details.

3: Refer to the sections on each peripheral for details on FIPS compliancy.

15.6.4 Chip Access Using JTAG Connection

The JTAG connection is not enabled by default on this chip at delivery due to the secure ROM code implementation.

By default, the SAMA5D2 devices boot in Standard mode and not in Secure mode. When the secure ROM code starts, it disables the JTAG access for the entire boot sequence.

If the secure ROM code does not find any program in the external memory, it enables the USB connection and the serial port and waits for a dedicated command to switch the chip into Secure mode.

If any other character is received, the secure ROM code starts the standard SAM-BA® monitor, locks access to the ROM memory, and enables the JTAG.

The chip can then be accessed using the JTAG connection.

If the secure ROM code finds a bootable program, it automatically disables ROM access and enables the JTAG connection just before launching the program.

The procedure to enable JTAG access is as follows:

- Connect your computer to the board with JTAG and USB (J20 USB-A)
- Power on the chip
- Open a terminal console (TeraTerm or HyperTerminal, etc.) on your computer and connect to the USB CDC Serial COM port related to the J14 connector on the board
- Send the '#' character. You will see then the prompt '>' character sent by the device (indicating that the Standard SAM-BA Monitor is running)
- Use the Standard SAM-BA Monitor to connect to the chip with JTAG

Note that you don't need to follow this sequence in order to connect the Standard SAM-BA Monitor with USB.

15.6.5 IEEE 1149.1 JTAG Boundary Scan

IEEE 1149.1 JTAG Boundary Scan allows pin-level access independent of the device packaging technology.

IEEE 1149.1 JTAG Boundary Scan is enabled when JTAGSEL is high. The SAMPLE, EXTEST and BYPASS functions are implemented. In ICE debug mode, the ARM processor responds with a non-JTAG chip ID that identifies the processor to the ICE system. This is not IEEE 1149.1 JTAG-compliant.

It is not possible to switch directly between JTAG and ICE operations. A chip reset must be performed after JTAGSEL is changed.

A Boundary-scan Descriptor Language (BSDL) file is provided to set up test.

21.8.5 Protect Mode

The Protect mode is used to read the Interrupt Vector register without performing the associated automatic operations. This is necessary when working with a debug system. When a debugger, working either with a Debug Monitor or the ARM processor's ICE, stops the applications and updates the opened windows, it might read the AIC User Interface and thus the IVR. This has adverse consequences:

- If an enabled interrupt with a higher priority than the current one is pending, it is stacked.
- If there is no enabled pending interrupt, the spurious vector is returned.

In either case, an End of Interrupt command is necessary to acknowledge and restore the context of the AIC. This operation is generally not performed by the debug system, as the debug system would become strongly intrusive and cause the application to enter an undesired state.

This is avoided by using the Protect mode. Writing PROT in the Debug Control register (AIC_DCR) at 0x1 enables the Protect mode.

When the Protect mode is enabled, the AIC performs interrupt stacking only when a write access is performed on AIC_IVR. Therefore, the Interrupt Service Routines must write (arbitrary data) to AIC_IVR just after reading it. The new context of the AIC, including the value of AIC_ISR, is updated with the current interrupt only when AIC_IVR is written.

An AIC_IVR read on its own (e.g., by a debugger) modifies neither the AIC context nor AIC_ISR. Extra AIC_IVR reads perform the same operations. However, it is recommended to not stop the processor between the read and the write of AIC_IVR of the interrupt service routine to make sure the debugger does not modify the AIC context.

To summarize, in normal operating mode, the read of AIC_IVR performs the following operations within the AIC:

1. Calculates active interrupt (higher than current or spurious).
2. Determines and returns the vector of the active interrupt.
3. Memorizes the interrupt.
4. Pushes the current priority level onto the internal stack.
5. Acknowledges the interrupt.

However, while the Protect mode is activated, only operations 1 to 3 are performed when AIC_IVR is read. Operations 4 and 5 are only performed by the AIC when AIC_IVR is written.

Software that has been written and debugged using the Protect mode runs correctly in normal mode without modification. However, in normal mode, the AIC_IVR write has no effect and can be removed to optimize the code.

21.8.6 Spurious Interrupt

The Advanced Interrupt Controller features a protection against spurious interrupts. A spurious interrupt is defined as being the assertion of an interrupt source long enough for the AIC to assert the nIRQ, but no longer present when AIC_IVR is read. This is most prone to occur when:

- An external interrupt source is programmed in Level-Sensitive mode and an active level occurs for only a short time.
- An internal interrupt source is programmed in level-sensitive and the output signal of the corresponding embedded peripheral is activated for a short time (as is the case for the watchdog).
- An interrupt occurs just a few cycles before the software begins to mask it, thus resulting in a pulse on the interrupt source.

The AIC detects a spurious interrupt at the time AIC_IVR is read while no enabled interrupt source is pending. When this happens, the AIC returns the value stored by the programmer in the Spurious Vector register (AIC_SPU). The programmer must store the address of a spurious interrupt handler in AIC_SPU as part of the application, to enable an as fast as possible return to the normal execution flow. This handler writes in AIC_EOICR and performs a return from interrupt.

21.8.7 General Interrupt Mask

The AIC features a General Interrupt Mask bit (AIC_DCR.GMSK) to prevent interrupts from reaching the processor. Both the nIRQ and the nFIQ lines are driven to their inactive state if AIC_DCR.GMSK is set. However, this mask does not prevent waking up the processor if it has entered Idle mode. This function facilitates synchronizing the processor on a next event and, as soon as the event occurs, performs subsequent operations without having to handle an interrupt. It is strongly recommended to use this mask with caution.

33.22.40 PMC Audio PLL Control Register 1

Name: PMC_AUDIO_PLL1

Address: 0xF0014150

Access: Read/Write

31	30	29	28	27	26	25	24
–	QDAUDIO					DIV	

23	22	21	20	19	18	17	16
–	–	FRACR					

15	14	13	12	11	10	9	8
FRACR							

7	6	5	4	3	2	1	0
FRACR							

FRACR: Fractional Loop Divider Setting

DIV: Divider Value

Value	Name	Description
0	FORBIDDEN	Reserved
1	FORBIDDEN	Reserved
2	DIV2	Divide by 2
3	DIV3	Divide by 3

QDAUDIO: Output Divider Ratio for Pad Clock

$f_{audio} = f_{ref} \times ((ND + 1) + FRACR \div 2^{22}) / (DIV \times QDAUDIO)$

34.7.4 PIO Lock Status Register

Name: PIO_LOCKSRx [x=0..3]

Address: 0xFC03800C [0], 0xFC03804C [1], 0xFC03808C [2], 0xFC0380CC [3]

Access: Read-only

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

P0–P31: Lock Status

0: The I/O line of the I/O group x is not locked.

1: The I/O line of the I/O group x is locked.

34.7.19 Secure PIO Set Output Data Register

Name: S_PIO_SODRx [x=0..3]

Address: 0xFC039010 [0], 0xFC039050 [1], 0xFC039090 [2], 0xFC0390D0 [3]

Access: Write-only

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

P0–P31: Set Output Data

0: No effect.

1: Sets the data to be driven on the I/O line of I/O group x.

38.9.20 XDMAC Channel x [x = 0..15] Interrupt Mask Register

Name: XDMAC_CIMx [x = 0..15]

Address: 0xF0004058 (1)[0], 0xF0004098 (1)[1], 0xF00040D8 (1)[2], 0xF0004118 (1)[3], 0xF0004158 (1)[4], 0xF0004198 (1)[5], 0xF00041D8 (1)[6], 0xF0004218 (1)[7], 0xF0004258 (1)[8], 0xF0004298 (1)[9], 0xF00042D8 (1)[10], 0xF0004318 (1)[11], 0xF0004358 (1)[12], 0xF0004398 (1)[13], 0xF00043D8 (1)[14], 0xF0004418 (1)[15], 0xF0010058 (0)[0], 0xF0010098 (0)[1], 0xF00100D8 (0)[2], 0xF0010118 (0)[3], 0xF0010158 (0)[4], 0xF0010198 (0)[5], 0xF00101D8 (0)[6], 0xF0010218 (0)[7], 0xF0010258 (0)[8], 0xF0010298 (0)[9], 0xF00102D8 (0)[10], 0xF0010318 (0)[11], 0xF0010358 (0)[12], 0xF0010398 (0)[13], 0xF00103D8 (0)[14], 0xF0010418 (0)[15]

Access: Read-only

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

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

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM

BIM: End of Block Interrupt Mask Bit

0: Block interrupt is masked.

1: Block interrupt is activated.

LIM: End of Linked List Interrupt Mask Bit

0: End of linked list interrupt is masked.

1: End of linked list interrupt is activated.

DIM: End of Disable Interrupt Mask Bit

0: End of disable interrupt is masked.

1: End of disable interrupt is activated.

FIM: End of Flush Interrupt Mask Bit

0: End of flush interrupt is masked.

1: End of flush interrupt is activated.

RBEIM: Read Bus Error Interrupt Mask Bit

0: Bus error interrupt is masked.

1: Bus error interrupt is activated.

WBEIM: Write Bus Error Interrupt Mask Bit

0: Bus error interrupt is masked.

1: Bus error interrupt is activated.

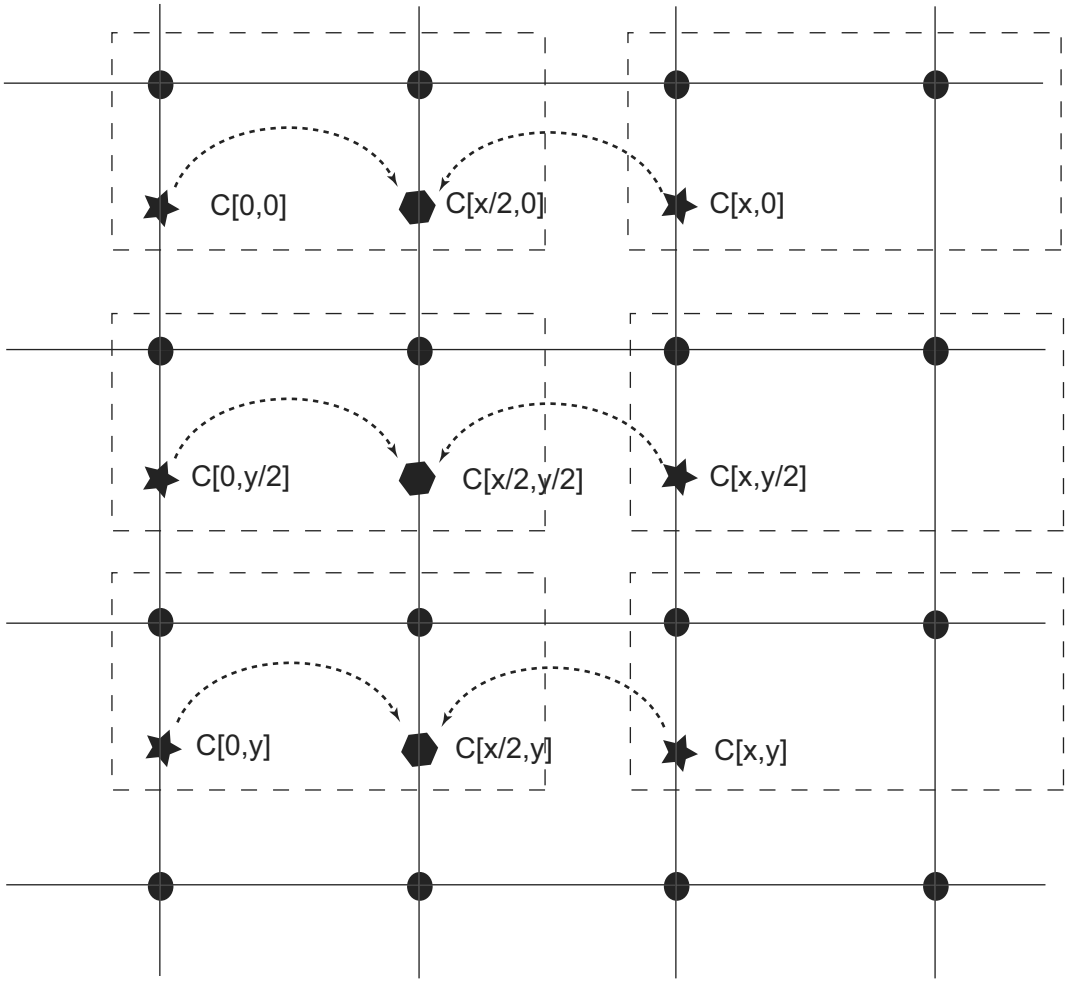
ROIM: Request Overflow Error Interrupt Mask Bit

0: Request overflow interrupt is masked.

1: Request overflow interrupt is activated.

Figure 39-2: 4:2:2 Upsampling Algorithm

Vertical and Horizontal upsampling 4:2:2 to 4:4:4 conversion 0 or 180 degrees



SFT: PTP Sync Frame Transmitted

PDRQFR: PDelay Request Frame Received

PDRSFR: PDelay Response Frame Received

PDRQFT: PDelay Request Frame Transmitted

PDRSFT: PDelay Response Frame Transmitted

SRI: TSU Seconds Register Increment

RXLPISBC: Enable RX LPI Indication

WOL: Wake On LAN

TSUTIMCOMP: TSU Timer Comparison

40.8.50 GMAC Broadcast Frames Transmitted Register

Name: GMAC_BCFT

Address: 0xF800810C

Access: Read-only

31	30	29	28	27	26	25	24
BFTX							

23	22	21	20	19	18	17	16
BFTX							

15	14	13	12	11	10	9	8
BFTX							

7	6	5	4	3	2	1	0
BFTX							

BFTX: Broadcast Frames Transmitted without Error

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

40.8.85 GMAC Receive Symbol Errors Register

Name: GMAC_RSE

Address: 0xF8008198

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
–	–	–	–	–	–	RXSE	

7	6	5	4	3	2	1	0
RXSE							

RXSE: Receive Symbol Errors

This register counts the number of frames that had GRXER asserted during reception. For 10/100 mode symbol errors are counted regardless of frame length checks. Receive symbol errors will also be counted as an FCS or alignment error if the frame is between 64 and 1518 bytes (1536 bytes if bit 8 is set in the Network Configuration Register). If the frame is larger it will be recorded as a jabber error. See Section 40.8.2 ”GMAC Network Configuration Register”.

47.10.28 USART Number of Errors Register

Name: FLEX_US_NER

Address: 0xF8034244 (0), 0xF8038244 (1), 0xFC010244 (2), 0xFC014244 (3), 0xFC018244 (4)

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
NB_ERRORS							

This register is relevant only if USART_MODE = 0x4 or 0x6 in the USART Mode Register.

NB_ERRORS: Number of Errors

Total number of errors that occurred during an ISO7816 transfer. This register automatically clears when read.

47.10.34 USART Comparison Register

Name: FLEX_US_CMPR

Address: 0xF8034290 (0), 0xF8038290 (1), 0xFC010290 (2), 0xFC014290 (3), 0xFC018290 (4)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	VAL2
23	22	21	20	19	18	17	16
VAL2							
15	14	13	12	11	10	9	8
–	CMPPAR	–	CMPMODE	–	–	–	VAL1
7	6	5	4	3	2	1	0
VAL1							

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

VAL1: First Comparison Value for Received Character

0–511: The received character must be higher or equal to the value of VAL1 and lower or equal to VAL2 to set the FLEX_US_CSR.CMP flag.

CMPMODE: Comparison Mode

Value	Name	Description
0	FLAG_ONLY	Any character is received and comparison function drives CMP flag.
1	START_CONDITION	Comparison condition must be met to start reception.

CMPPAR: Compare Parity

0: The parity is not checked and a bad parity cannot prevent from waking up the system.

1: The parity is checked and a matching condition on data can be cancelled by an error on parity bit, so no wakeup is performed.

VAL2: Second Comparison Value for Received Character

0–511: The received character must be lower or equal to the value of VAL2 and higher or equal to VAL1 to set the FLEX_US_CSR.CMP flag.

48.6.9 UART Baud Rate Generator Register

Name: UART_BRGR

Address: 0xF801C020 (0), 0xF8020020 (1), 0xF8024020 (2), 0xFC008020 (3), 0xFC00C020 (4)

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
CD							
7	6	5	4	3	2	1	0
CD							

CD: Clock Divisor

0: Baud rate clock is disabled

1 to 65,535:

If BRSRCCK = 0:

$$CD = \frac{f_{peripheral\ clock}}{16 \times Baud\ Rate}$$

If BRSRCCK = 1:

$$CD = \frac{f_{GCLKx}}{16 \times Baud\ Rate}$$

CMDIDX: Command Index Error

This bit is set to 1 if a Command Index error occurs in the command response.

This bit can only be set to 1 if SDMMC_EISTER.CMDIDX is set to 1. An interrupt can only be generated if SDMMC_EISIER.CMDIDX is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

DATTEO: Data Timeout error

This bit is set to 1 when detecting one of following timeout conditions.

- Busy timeout for R1b, R5b response type (see “Physical Layer Simplified Specification V3.01” and “SDIO Simplified Specification V3.00”).
- Busy timeout after Write CRC Status.
- Write CRC Status timeout.
- Read data timeout

This bit can only be set to 1 if SDMMC_EISTER.DATTEO is set to 1. An interrupt can only be generated if SDMMC_EISIER.DATTEO is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

DATCRC: Data CRC Error

This bit is set to 1 when detecting a CRC error during a transfer of read data which uses the DAT line or when detecting that the Write CRC Status has a value other than “010”.

This bit can only be set to 1 if SDMMC_EISTER.DATCRC is set to 1. An interrupt can only be generated if SDMMC_EISIER. DATCRC is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

DATEND: Data End Bit Error

This bit is set to 1 either when detecting 0 at the end bit position of read data which uses the DAT line or at the end bit position of the CRC Status.

This bit can only be set to 1 if SDMMC_EISTER.DATEND is set to 1. An interrupt can only be generated if SDMMC_EISIER.DATEND is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

CURLIM: Current Limit Error

By setting SD Bus Power (SDBPWR) in SDMMC_PSR, the SDMMC is requested to supply power for the SD Bus. The SDMMC is protected from an illegal card by stopping power supply to the card, in which case this bit indicates a failure status. Reading 1 means the SDMMC is not supplying power to the card due to some failure. Reading 0 means that the SDMMC is supplying power and no error has occurred. The SDMMC may require some sampling time to detect the current limit.

This bit can only be set to 1 if SDMMC_EISTER.CURLIM is set to 1. An interrupt can only be generated if SDMMC_EISIER.CURLIM is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

52.6.10 ISC Clock Configuration Register

Name: ISC_CLKCFG

Address: 0xF0008024

Access: Read-only

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

23	22	21	20	19	18	17	16
MCDIV							

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

7	6	5	4	3	2	1	0
ICDIV							

ICDIV: ISP Clock Divider

$$f_{cc} = \frac{f_{ccref}}{ICDIV + 1}$$

ICSEL: ISP Clock Selection

0: HCLOCK is selected.

1: ISCCLK is selected.

MCDIV: Master Clock Divider

$$f_{mc} = \frac{f_{mcref}}{MCDIV + 1}$$

MCSEL: Master Clock Reference Clock Selection

0: HCLOCK is selected.

1: ISCCLK is selected.

2: GCK is selected.

53.5.7.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address MCAN_XIDFC.FLESA plus two times the index of the filter element (0…63).

Table 53-13: Extended Message ID Filter Element

	31	24	23	16	15	8	7	0
F0	EFEC [2:0]		EFID1[28:0]					
F1	EFT[1:0]	–	EFID2[28:0]					

- **F0 Bit 31:29 EFEC[2:0]: Extended Filter Element Configuration**

All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = “100”, “101”, or “110”, a match sets the interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case, register MCAN_HPMS is updated with the status of the priority match.

Value	Description
0	Disable filter element
1	Store in Rx FIFO 0 if filter matches
2	Store in Rx FIFO 1 if filter matches
3	Reject ID if filter matches
4	Set priority if filter matches
5	Set priority and store in FIFO 0 if filter matches
6	Set priority and store in FIFO 1 if filter matches
7	Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored

- **F0 Bits 28:0 EFID1[28:0]: Extended Filter ID 1**

First ID of extended ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only MCAN_XIDAM masking mechanism (see Extended Message ID Filtering) is used.

- **F1 Bits 31:30 EFT[1:0]: Extended Filter Type**

Value	Description
0	Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID)
1	Dual ID filter for EF1ID or EF2ID
2	Classic filter: EF1ID = filter, EF2ID = mask
3	Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID), MCAN_XIDAM mask not applied

62.5.1 TDES Control Register

Name: TDES_CR

Address: 0xFC044000

Access: Write-only

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

- **START: Start Processing**

0: No effect

1: Starts Manual encryption/decryption process.

SWRST: Software Reset

0: No effect

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

65.7.22 ADC Touchscreen X Position Register

Name: ADC_XPOSR

Address: 0xFC0300B4

Access: Read-only

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

XPOS: X Position

The position measured is stored here. If XPOS = 0 or XPOS = XSIZE, the pen is on the border.

When pen detection is enabled (PENDET set to ‘1’ in ADC_TSMR), XPOS is tied to 0 while there is no detection of contact on the touchscreen (i.e., when PENS bit is cleared in ADC_ISR).

XSCALE: Scale of XPOS

Indicates the max value that XPOS can reach. This value should be close to 2^{12}.

Table 66-91: I2SC0 IOSET2 Timings (Continued)

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
$I2SC_8$	SCK falling to SDO valid	3.7	12	3	10	ns

Table 66-92: I2SC1 IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master					
$I2SC_0$	SDI Input setup time before SCK rises	13.1	–	11.4	–	ns
$I2SC_1$	SDI Input hold time after SCK rises	0	–	0	–	ns
$I2SC_2$	SCK falling to SDO valid	0	3.5	0	3.6	ns
$I2SC_3$	SCK falling to WS valid	0	2.9	0	3	ns
	Slave					
$I2SC_4$	SDI Input setup time before SCK rises	1.4	–	1.3	–	ns
$I2SC_5$	SDI Input hold time after SCK rises	1	–	0.8	–	ns
$I2SC_6$	WS Input setup time before SCK rises	2.4	–	2.1	–	ns
$I2SC_7$	WS Input hold time after SCK rises	0.8	–	0.7	–	ns
$I2SC_8$	SCK falling to SDO valid	4.4	13.8	3.7	11.9	ns

Table 66-93: I2SC1 IOSET2 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master					
$I2SC_0$	SDI Input setup time before SCK rises	12.9	–	11.2	–	ns
$I2SC_1$	SDI Input hold time after SCK rises	0	–	0	–	ns
$I2SC_2$	SCK falling to SDO valid	0	3.6	0	3.7	ns
$I2SC_3$	SCK falling to WS valid	0	2.9	0	3	ns
	Slave					
$I2SC_4$	SDI Input setup time before SCK rises	1.1	–	1	–	ns
$I2SC_5$	SDI Input hold time after SCK rises	1.2	–	1	–	ns
$I2SC_6$	WS Input setup time before SCK rises	2.2	–	2	–	ns
$I2SC_7$	WS Input hold time after SCK rises	0.9	–	0.7	–	ns
$I2SC_8$	SCK falling to SDO valid	4.3	14	3.7	12	ns