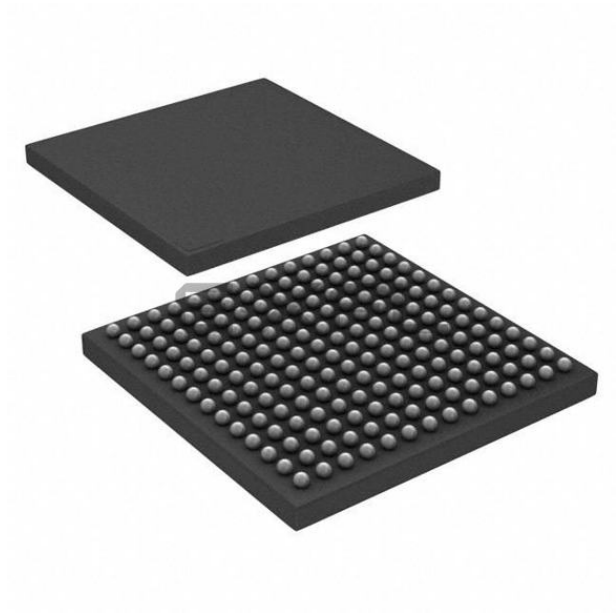


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Obsolete
Core Processor	ARM® Cortex®-A5
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	500MHz
Co-Processors/DSP	Multimedia; NEON™ MPE
RAM Controllers	LPDDR1, LPDDR2, LPDDR3, DDR2, DDR3, DDR3L, QSPI
Graphics Acceleration	Yes
Display & Interface Controllers	Keyboard, LCD, Touchscreen
Ethernet	10/100Mbps (1)
SATA	-
USB	USB 2.0 + HSIC
Voltage - I/O	3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	ARM TZ, Boot Security, Cryptography, RTIC, Secure Fusebox, Secure JTAG, Secure Memory, Secure RTC
Package / Case	196-TFBGA, CSBGA
Supplier Device Package	196-TFBGA (11x11)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d22a-cu

1. Description

The SAMA5D2 Series is a high-performance, power-efficient embedded MPU based on the ARM Cortex-A5 processor. It integrates the ARM NEON SIMD engine for accelerated multimedia and signal processing, a configurable 128-Kbyte L2 cache, a floating point unit for high-precision computing and reliable performance, as well as high data bandwidth architecture. The device features an advanced user interface and connectivity peripherals. Advanced security is provided by powerful cryptographic accelerators, by the ARM TrustZone technology securing access to memories and sensitive peripherals, and by several hardware features that safeguard memory content, authenticate software reliability, detect physical attacks and prevent information leakage during code execution.

The SAMA5D2 features an internal multilayer bus architecture associated with 2 x 16 DMA channels and dedicated DMAs for the communication and interface peripherals required to ensure uninterrupted data transfers with minimal processor overhead. The device supports DDR2, DDR3, DDR3L, LPDDR1, LPDDR2, LPDDR3, and SLC/MLC NAND Flash memory up to 32-bit ECC.

The comprehensive peripheral set includes an LCD TFT controller with overlays for hardware-accelerated image composition, an image sensor controller, audio support through I^2S, SSC, a stereo Class D amplifier and a digital microphone. Connectivity peripherals include a 10/100 EMAC, USBs, CAN-FDs, FLEXCOMs, UARTs, SPIs and two QSPIs, SDIO/SD/e.MMCs, and TWIs/I^2C.

Protection of code and data is provided by automatic scrambling of memories and an Integrity Check Monitor (ICM) to detect any modification of the memory contents. The SAMA5D2 also supports execution of encrypted code (QSPI or one portion of the DDR) with an “on-the-fly” encryption-decryption process.

With its secure design architecture, cryptographic acceleration engines, and secure boot loader, the SAMA5D2 is the ideal solution for point-of-sale (POS), IoT and industrial applications requiring anti-cloning, data protection and secure communication transfer.

SAMA5D2 devices feature three software-selectable low-power modes: Idle, Ultra-low-power and Backup.

In Idle mode, the processor is stopped while all other functions can be kept running.

In Ultra-low-power-mode 0, the processor is stopped while all other functions are clocked at 512 Hz and interrupts or peripherals can be configured to wake up the system based on events, including partial asynchronous wakeup (SleepWalking).

In Ultra-low-power mode 1, all clocks and functions are stopped but some peripherals can be configured to wake up the system based on events, including partial asynchronous wakeup (SleepWalking).

In Backup mode, RTC and wakeup logic are active. The Backup mode can be extended to feature DDR in Self-refresh mode.

SAMA5D2 devices also include an Event System that allows peripherals to receive, react to and send events in Active and Idle modes without processor intervention.

Table 6-2: Pin Description (Continued)

289-pin BGA	256-pin BGA	196-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO peripheral				Reset State (Signal, Dir, PU, PD, HiZ, ST)(1)(2)
					Signal	Dir	Signal	Dir	Func	Signal	Dir	IO Set	
R12	U12	–	VDDSDMMC	GPIO_EMMC	PA6	I/O	–	–	A	SDMMC0_DAT4	I/O	1	PIO, I, PU, ST
									B	QSPI1_SCK	O	1	
									D	TIOA5	I/O	1	
									E	FLEXCOM2_IO0	I/O	1	
									F	D6	I/O	2	
T13	V12	–	VDDSDMMC	GPIO_EMMC	PA7	I/O	–	–	A	SDMMC0_DAT5	I/O	1	PIO, I, PU, ST
									B	QSPI1_IO0	I/O	1	
									D	TIOB5	I/O	1	
									E	FLEXCOM2_IO1	I/O	1	
									F	D7	I/O	2	
N10	N11	–	VDDSDMMC	GPIO_EMMC	PA8	I/O	–	–	A	SDMMC0_DAT6	I/O	1	PIO, I, PU, ST
									B	QSPI1_IO1	I/O	1	
									D	TCLK5	I	1	
									E	FLEXCOM2_IO2	I/O	1	
									F	NWE/NANDWE	O	2	
N11	P12	–	VDDSDMMC	GPIO_EMMC	PA9	I/O	–	–	A	SDMMC0_DAT7	I/O	1	PIO, I, PU, ST
									B	QSPI1_IO2	I/O	1	
									D	TIOA4	I/O	1	
									E	FLEXCOM2_IO3	O	1	
									F	NCS3	O	2	
U13	U13	–	VDDSDMMC	GPIO_EMMC	PA10	I/O	–	–	A	SDMMC0_RSTN	O	1	PIO, I, PU, ST
									B	QSPI1_IO3	I/O	1	
									D	TIOB4	I/O	1	
									E	FLEXCOM2_IO4	O	1	
									F	A21/NANDALE	O	2	
P15	R14	–	VDDIOP1	GPIO	PA11	I/O	–	–	A	SDMMC0_1V8SEL	O	1	PIO, I, PU, ST
									B	QSPI1_CS	O	1	
									D	TCLK4	I	1	
									F	A22/NANDCLE	O	2	
N15	N13	–	VDDIOP1	GPIO	PA12	I/O	–	–	A	SDMMC0_WP	I	1	PIO, I, PU, ST
									B	IRQ	I	1	
									F	NRD/NANDOE	O	2	
P16	P14	–	VDDIOP1	GPIO	PA13	I/O	–	–	A	SDMMC0_CD	I	1	PIO, I, PU, ST
									E	FLEXCOM3_IO1	I/O	1	
									F	D8	I/O	2	

Table 6-2: Pin Description (Continued)

289-pin BGA	256-pin BGA	196-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO peripheral				Reset State (Signal, Dir, PU, PD, HiZ, ST)[1][2]
					Signal	Dir	Signal	Dir	Func	Signal	Dir	IO Set	
M14	P17	–	VDDIOP1	GPIO_QSPI	PA14	I/O	–	–	A	SPI0_SPCK	I/O	1	PIO, I, PU, ST
									B	TK1	I/O	1	
									C	QSPI0_SCK	O	2	
									D	I2SC1_MCK	O	2	
									E	FLEXCOM3_IO2	I/O	1	
									F	D9	I/O	2	
N16	R18	–	VDDIOP1	GPIO	PA15	I/O	–	–	A	SPI0_MOSI	I/O	1	PIO, I, PU, ST
									B	TF1	I/O	1	
									C	QSPI0_CS	O	2	
									D	I2SC1_CK	I/O	2	
									E	FLEXCOM3_IO0	I/O	1	
									F	D10	I/O	2	
M10	N15	–	VDDIOP1	GPIO_IO	PA16	I/O	–	–	A	SPI0_MISO	I/O	1	PIO, I, PU, ST
									B	TD1	O	1	
									C	QSPI0_IO0	I/O	2	
									D	I2SC1_WS	I/O	2	
									E	FLEXCOM3_IO3	O	1	
									F	D11	I/O	2	
N17	P18	–	VDDIOP1	GPIO_IO	PA17	I/O	–	–	A	SPI0_NPCS0	I/O	1	PIO, I, PU, ST
									B	RD1	I	1	
									C	QSPI0_IO1	I/O	2	
									D	I2SC1_DI0	I	2	
									E	FLEXCOM3_IO4	O	1	
									F	D12	I/O	2	
U14	M9	L9	VDDIOP1	GPIO_IO	PA18	I/O	–	–	A	SPI0_NPCS1	O	1	PIO, I, PU, ST
									B	RK1	I/O	1	
									C	QSPI0_IO2	I/O	2	
									D	I2SC1_DO0	O	2	
									E	SDMMC1_DAT0	I/O	1	
									F	D13	I/O	2	
T14	V13	N9	VDDIOP1	GPIO_IO	PA19	I/O	–	–	A	SPI0_NPCS2	O	1	PIO, I, PU, ST
									B	RF1	I/O	1	
									C	QSPI0_IO3	I/O	2	
									D	TIOA0	I/O	1	
									E	SDMMC1_DAT1	I/O	1	
									F	D14	I/O	2	

7.3 Powerdown Considerations

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

Figure 7-2: Recommended Powerdown Sequence

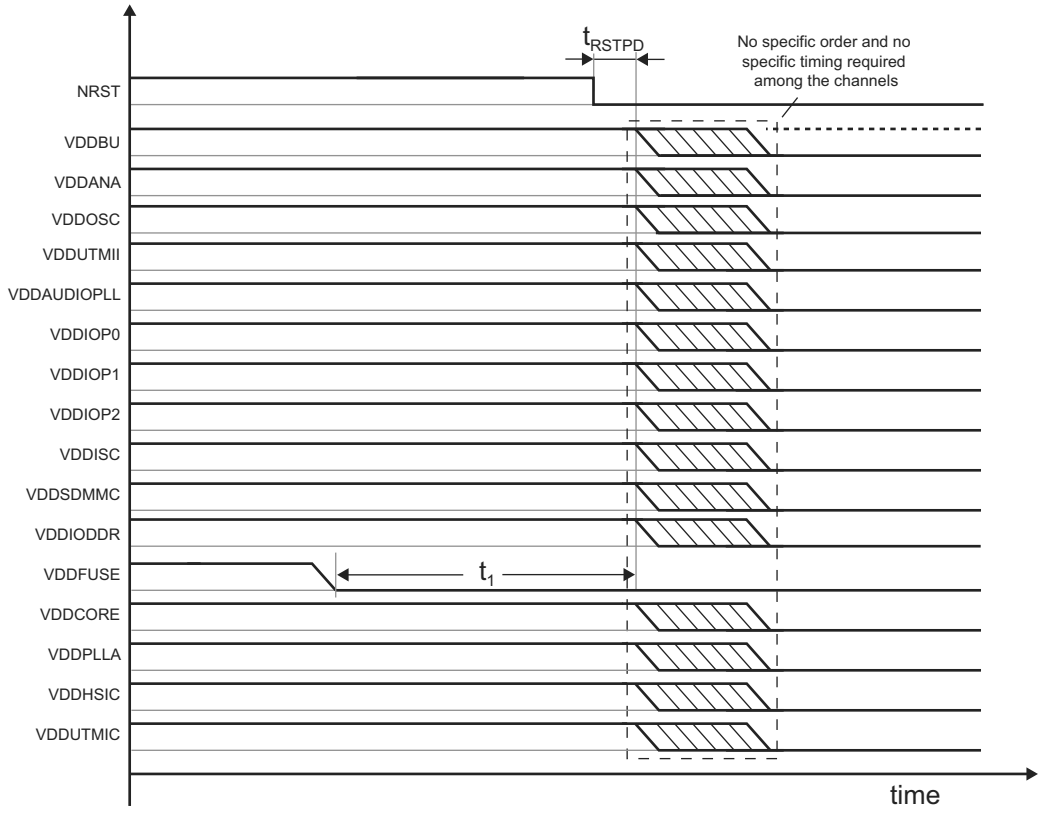


Table 7-3: Powerdown Timing Specification

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

7.4 Power Supply Sequencing at Backup Mode Entry and Exit

7.4.1 VDDBU Power Architecture

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

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

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

2. QSPI NOR memories without SFDP

This section only applies when the ROM code fails to read the SFDP tables from the QSPI NOR memory.

The ROM code reads the JEDEC ID of the QSPI NOR memory, and then selects the read settings based on the manufacturer ID (first byte of the JEDEC ID) from the following hard-coded values:

	Cypress (01h)	**Micron (20h)**	**Macronix (C2h)**	**Winbond (EFh)**	**Others**
Fast Read protocol	SPI 1-4-4	SPI 1-4-4	SPI 1-4-4	SPI 1-4-4	SPI 1-1-1
Fast Read op code	EBh	EBh	EBh	EBh	0Bh
Address width	24 bits	24 bits	24 bits	24 bits	24 bits
Number of mode clock cycles	2	1	2	2	0
Number of wait states	4	9	4	4	8
Value of mode cycles to enter the 0-4-4 mode (XIP)	A0h	0h The ROM code first sets XIP bit[3] in the Volatile Configuration Register (VCR)	0Fh	A5h	N/A
Value of mode cycles to exit the 0-4-4 mode (normal read)	00h	1h	00h	FFh	N/A
XIP supported	yes	yes	yes	yes	no

Those hard-coded parameters give a last chance to the ROM code to boot from a QSPI NOR memory in either normal mode or XIP (continuous read) mode.

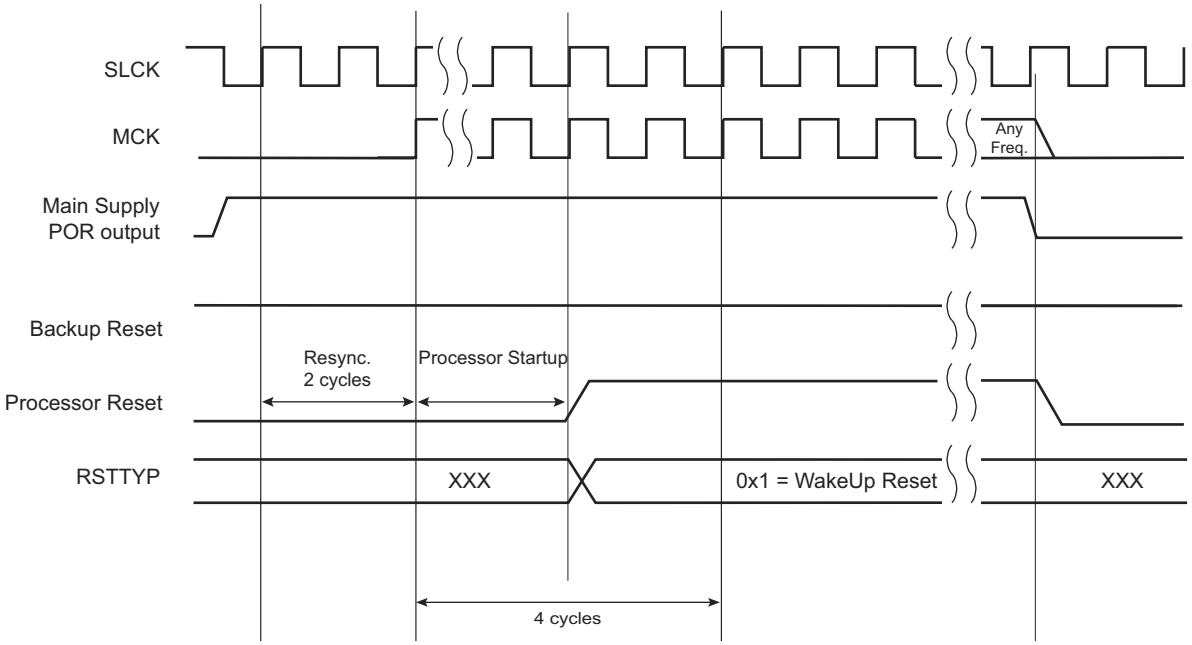
23.4.3.2 Wakeup Reset

The wakeup reset occurs when the main supply is down. When the main supply POR output is active, all the reset signals are asserted except Backup Reset. When the main supply powers up, the POR output is resynchronized on Slow Clock. The processor clock is then re-enabled during 2 Slow Clock cycles, depending on the requirements of the ARM processor.

At the end of this delay, the processor and other reset signals rise. The field RSTTYP in the RSTC_SR is updated to report a wakeup reset.

When the main supply is detected falling, the reset signals are immediately asserted. This transition is synchronous with the output of the main supply POR.

Figure 23-4: Wakeup Reset



33. Power Management Controller (PMC)

33.1 Description

The Power Management Controller (PMC) optimizes power consumption by controlling all system and user peripheral clocks. The PMC enables/disables the clock inputs to many of the peripherals and the Core.

33.2 Embedded Characteristics

The Power Management Controller provides the following clocks:

- Master Clock (MCK)—programmable from a few hundred Hz to the maximum operating frequency of the device. It is available to the modules running permanently.
- Processor Clock (PCK)—must be switched off when processor is entering Idle mode
- HS USB Device Clock (UDPCK)
- H64MX Matrix Clock (MCK) and H32MX Matrix Clock (MCK or MCK/2)
- Peripheral Clocks—provided to the embedded peripherals and independently controllable
- Programmable Clock outputs can be selected from the clocks provided by the clock generator and driven on the PCKx pins.
- Generic Clock (GCLK) for peripherals that can accept a second clock source
- Asynchronous partial wakeup (SleepWalking) for FLEXCOMx, SPIx, TWIx, UARTx and ADC

39.7.69 Overlay 2 Configuration Register 3

Name: LCDC_OVR2CFG3

Address: 0xF0000278

Access: Read/Write

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

23	22	21	20	19	18	17	16
YSIZE							

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

7	6	5	4	3	2	1	0
XSIZE							

XSIZE: Horizontal Window Size

Overlay 2 window width in pixels. The window width is set to (XSIZE + 1).

The following constraint must be met: XPOS + XSIZE ≤ PPL

YSIZE: Vertical Window Size

Overlay 2 window height in pixels. The window height is set to (YSIZE + 1).

The following constraint must be met: YPOS + YSIZE ≤ RPF

If, during the frame reception, the frame is found to be too long, a bad frame indication is sent to the FIFO interface. The receiver logic ceases to send data to memory as soon as this condition occurs.

At end of frame reception the receive block indicates to the DMA block whether the frame is good or bad. The DMA block will recover the current receive buffer if the frame was bad.

Ethernet frames are normally stored in DMA memory complete with the FCS. Setting the FCS remove bit in the network configuration (bit 17) causes frames to be stored without their corresponding FCS. The reported frame length field is reduced by four bytes to reflect this operation.

The receive block signals to the register block to increment the alignment, CRC (FCS), short frame, long frame, jabber or receive symbol errors when any of these exception conditions occur.

If bit 26 is set in the network configuration, CRC errors will be ignored and CRC errored frames will not be discarded, though the Frame Check Sequence Errors statistic register will still be incremented. Additionally, if not enabled for jumbo frames mode, then bit[13] of the receiver descriptor word 1 will be updated to indicate the FCS validity for the particular frame. This is useful for applications such as EtherCAT whereby individual frames with FCS errors must be identified.

Received frames can be checked for length field error by setting the length field error frame discard bit of the Network Configuration register (bit-16). When this bit is set, the receiver compares a frame's measured length with the length field (bytes 13 and 14) extracted from the frame. The frame is discarded if the measured length is shorter. This checking procedure is for received frames between 64 bytes and 1518 bytes in length.

Each discarded frame is counted in the 10-bit Length Field Frame Error statistics register. Frames where the length field is greater than or equal to 0x0600 hex will not be checked.

40.6.6 Checksum Offload for IP, TCP and UDP

The GMAC can be programmed to perform IP, TCP and UDP checksum offloading in both receive and transmit directions, which is enabled by setting bit 24 in the Network Configuration register for receive and bit 11 in the DMA Configuration register for transmit.

IPv4 packets contain a 16-bit checksum field, which is the 16-bit 1’s complement of the 1’s complement sum of all 16-bit words in the header. TCP and UDP packets contain a 16-bit checksum field, which is the 16-bit 1’s complement of the 1’s complement sum of all 16-bit words in the header, the data and a conceptual IP pseudo header.

To calculate these checksums in software requires each byte of the packet to be processed. For TCP and UDP this can use a large amount of processing power. Offloading the checksum calculation to hardware can result in significant performance improvements.

For IP, TCP or UDP checksum offload to be useful, the operating system containing the protocol stack must be aware that this offload is available so that it can make use of the fact that the hardware can either generate or verify the checksum.

40.6.6.1 Receiver Checksum Offload

When receive checksum offloading is enabled in the GMAC, the IPv4 header checksum is checked as per RFC 791, where the packet meets the following criteria:

- If present, the VLAN header must be four octets long and the CFI bit must not be set.
- Encapsulation must be RFC 894 Ethernet Type Encoding or RFC 1042 SNAP Encoding.
- IPv4 packet
- IP header is of a valid length

The GMAC also checks the TCP checksum as per RFC 793, or the UDP checksum as per RFC 768, if the following criteria are met:

- IPv4 or IPv6 packet
- Good IP header checksum (if IPv4)
- No IP fragmentation
- TCP or UDP packet

When an IP, TCP or UDP frame is received, the receive buffer descriptor gives an indication if the GMAC was able to verify the checksums. There is also an indication if the frame had SNAP encapsulation. These indication bits will replace the type ID match indication bits when the receive checksum offload is enabled. For details of these indication bits refer to Table 40-4 “Receive Buffer Descriptor Entry”.

If any of the checksums are verified as incorrect by the GMAC, the packet is discarded and the appropriate statistics counter incremented.

40.6.6.2 Transmitter Checksum Offload

The transmitter checksum offload is only available if the full store and forward mode is enabled. This is because the complete frame to be transmitted must be read into the packet buffer memory before the checksum can be calculated and written back into the headers at the beginning of the frame.

43.7.10 CLASSD Write Protection Mode Register

Name: CLASSD_WPMR

Address: 0xFC0480E4

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 0x434C44 (“CLD” in ASCII).

1: Enables the write protection if WPKEY corresponds to 0x434C44 (“CLD” in ASCII).

See Section 43.6.7 “Register Write Protection” for the list of registers that can be write-protected.

WPKEY: Write Protection Key

Value	Name	Description
0x434C44	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

SDA: SDA Line Value

0: SDA line sampled value is ‘0’.

1: SDA line sampled value is ‘1’.

47.7.12 USART Register Write Protection

The FLEXCOM operating mode (FLEX_MR.OPMODE) must be set to FLEX_MR_OPMODE_USART to enable access to the write protection registers.

To prevent any single software error from corrupting USART behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable) bit in the USART Write Protection Mode Register (FLEX_US_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the USART Write Protection Status Register (FLEX_US_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading FLEX_US_WPSR.

The following registers can be write-protected when WPEN is set:

- USART Mode Register
- USART Baud Rate Generator Register
- USART Receiver Timeout Register
- USART Transmitter Timeguard Register
- USART FI DI RATIO Register
- USART IrDA FILTER Register
- USART Manchester Configuration Register
- USART Comparison Register

47.9.4 Multi-Master Mode

47.9.4.1 Definition

In Multi-Master mode, more than one master may handle the bus at the same time without data corruption by using arbitration.

Arbitration starts as soon as two or more masters place information on the bus at the same time, and stops (arbitration is lost) for the master that intends to send a logical one while the other master sends a logical zero.

As soon as arbitration is lost by a master, it stops sending data and listens to the bus in order to detect a STOP. When the STOP is detected, the master that has lost arbitration may put its data on the bus by respecting arbitration.

Arbitration is illustrated in Figure 47-111.

47.9.4.2 Different Multi-Master Modes

Two Multi-Master modes may be distinguished:

- TWI as Master Only—TWI is considered as a master only and will never be addressed.
- TWI as Master or Slave—TWI may be either a master or a slave and may be addressed.

  Note: Arbitration in supported in both Multi-Master modes.

- TWI as Master Only

In this mode, the TWI is considered as a master only (MSEN is always at one) and must be driven like a master with the ARBLST (ARBitration Lost) flag in addition.

If arbitration is lost (ARBLST = 1), the user must reinitiate the data transfer.

If the user starts a transfer (ex.: DADR + START + W + Write in THR) and if the bus is busy, the TWI automatically waits for a STOP condition on the bus to initiate the transfer (see Figure 47-110).

Note: The state of the bus (busy or free) is not indicated in the user interface.

- TWI as Master or Slave

The automatic reversal from master to slave is not supported in case of a lost arbitration.

Then, in the case where TWI may be either a master or a slave, the user must manage the pseudo Multi-Master mode described in the steps below:

1. Program the TWI in Slave mode (SADR + MSDIS + SVEN) and perform a slave access (if TWI is addressed).
2. If the TWI has to be set in Master mode, wait until TXCOMP flag is at 1.
3. Program the Master mode (DADR + SVDIS + MSEN) and start the transfer (ex: START + Write in THR).
4. As soon as the Master mode is enabled, the TWI scans the bus in order to detect if it is busy or free. When the bus is considered as free, the TWI initiates the transfer.
5. As soon as the transfer is initiated and until a STOP condition is sent, the arbitration becomes relevant and the user must monitor the ARBLST flag.
6. If the arbitration is lost (ARBLST is = 1), the user must program the TWI in Slave mode in case the master that won the arbitration needs to access the TWI.
7. If the TWI has to be set in Slave mode, wait until TXCOMP flag is at 1 and then program the Slave mode.

Note: In case the arbitration is lost and the TWI is addressed, the TWI will not acknowledge even if it is programmed in Slave mode as soon as ARBLST = 1. Then the master must repeat SADR.

47.10.24 USART Baud Rate Generator Register

Name: FLEX_US_BRGR

Address: 0xF8034220 (0), 0xF8038220 (1), 0xFC010220 (2), 0xFC014220 (3), 0xFC018220 (4)

Access: Read/Write

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

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

15	14	13	12	11	10	9	8
CD							

7	6	5	4	3	2	1	0
CD							

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

CD: Clock Divider

	USART_MODE ≠ ISO7816			
	SYNC = 0		**SYNC = 1 or USART_MODE = SPI (master or Slave)**	**USART_MODE = ISO7816**
CD	**OVER = 0**	**OVER = 1**		
0	Baud Rate Clock Disabled			
1 to 65535	CD = Selected Clock / (16 × Baud Rate)	CD = Selected Clock / (8 × Baud Rate)	CD = Selected Clock / Baud Rate	CD = Selected Clock / (FI_DI_RATIO × Baud Rate)

FP: Fractional Part

0: Fractional divider is disabled.

1–7: Baud rate resolution, defined by FP × 1/8.

Warning: When the value of field FP is greater than 0, the SCK (oversampling clock) generates nonconstant duty cycles. The SCK high duration is increased by “selected clock” period from time to time. The duty cycle depends on the value of the CD field.

47.10.25 USART Receiver Timeout Register

Name: FLEX_US_RTOR

Address: 0xF8034224 (0), 0xF8038224 (1), 0xFC010224 (2), 0xFC014224 (3), 0xFC018224 (4)

Access: Read/Write

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

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

TO: Timeout Value

0: The receiver timeout is disabled.

1–131071: The receiver timeout is enabled and the timeout delay is TO × bit period.

52.6 Image Sensor Controller (ISC) User Interface

Table 52-18: Register Mapping

Offset	Register	Name	Access	Reset
0x00	Control Enable Register	ISC_CTRLEN	Write-only	–
0x04	Control Disable Register	ISC_CTRLDIS	Write-only	–
0x08	Control Status Register	ISC_CTRLSR	Read-only	0x00000000
0x0C	Parallel Front End Configuration 0 Register	ISC_PFE_CFG0	Read/Write	0x00000000
0x10	Parallel Front End Configuration 1 Register	ISC_PFE_CFG1	Read/Write	0x00000000
0x14	Parallel Front End Configuration 2 Register	ISC_PFE_CFG2	Read/Write	0x00000000
0x18	Clock Enable Register	ISC_CLKEN	Write-only	–
0x1C	Clock Disable Register	ISC_CLKDIS	Write-only	–
0x20	Clock Status Register	ISC_CLKSR	Read-only	0x00000000
0x24	Clock Configuration Register	ISC_CLKCFG	Read/Write	0x00000000
0x28	Interrupt Enable Register	ISC_INTEN	Write-only	–
0x2C	Interrupt Disable Register	ISC_INTDIS	Write-only	–
0x30	Interrupt Mask Register	ISC_INTMASK	Read-only	0x00000000
0x34	Interrupt Status Register	ISC_INTSR	Read-only	0x00000000
0x38–0x3C	Reserved	–	–	0x00000000
0x40–0x54	Reserved	–	–	0x00000000
0x58	White Balance Control Register	ISC_WB_CTRL	Read/Write	0x00000000
0x5C	White Balance Configuration Register	ISC_WB_CFG	Read/Write	0x00000000
0x60	White Balance Offset for R, GR Register	ISC_WB_O_RGR	Read/Write	0x00000000
0x64	White Balance Offset for B, GB Register	ISC_WB_O_BGB	Read/Write	0x00000000
0x68	White Balance Gain for R, GR Register	ISC_WB_G_RGR	Read/Write	0x00000000
0x6C	White Balance Gain for B, GB Register	ISC_WB_G_BGB	Read/Write	0x00000000
0x70	Color Filter Array Control Register	ISC_CFA_CTRL	Read/Write	0x00000000
0x74	Color Filter Array Configuration Register	ISC_CFA_CFG	Read/Write	0x00000000
0x78	Color Correction Control Register	ISC_CC_CTRL	Read/Write	0x00000000
0x7C	Color Correction RR RG Register	ISC_CC_RR_RG	Read/Write	0x00000000
0x80	Color Correction RB OR Register	ISC_CC_RB_OR	Read/Write	0x00000000
0x84	Color Correction GR GG Register	ISC_CC_GR_GG	Read/Write	0x00000000
0x88	Color Correction GB OG Register	ISC_CC_GB_OG	Read/Write	0x00000000
0x8C	Color Correction BR BG Register	ISC_CC_BR_BG	Read/Write	0x00000000
0x90	Color Correction BB OB Register	ISC_CC_BB_OB	Read/Write	0x00000000
0x94	Gamma Correction Control Register	ISC_GAM_CTRL	Read/Write	0x00000000
0x98	Gamma Correction Blue Entry 0	ISC_GAM_BENTRY0	Read/Write	0x00000000
...	...	...	...	...
0x194	Gamma Correction Blue Entry 63	ISC_GAM_BENTRY63	Read/Write	0x00000000

53.5 Functional Description

53.5.1 Operating Modes

53.5.1.1 Software Initialization

Software initialization is started by setting bit MCAN_CCCR.INIT, either by software or by a hardware reset, when an uncorrected bit error was detected in the Message RAM, or by going Bus_Off. While MCAN_CCCR.INIT is set, message transfer from and to the CAN bus is stopped and the status of the CAN bus output CANTX is recessive (HIGH). The counters of the Error Management Logic EML are unchanged. Setting MCAN_CCCR.INIT does not change any configuration register. Resetting MCAN_CCCR.INIT finishes the software initialization. Afterwards the Bit Stream Processor BSP synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (≡ Bus_Idle) before it can take part in bus activities and start the message transfer.

Access to the MCAN configuration registers is only enabled when both bits MCAN_CCCR.INIT and MCAN_CCCR.CCE are set (protected write).

MCAN_CCCR.CCE can only be configured when MCAN_CCCR.INIT = ‘1’. MCAN_CCCR.CCE is automatically cleared when MCAN_CCCR.INIT = ‘0’.

The following registers are cleared when MCAN_CCCR.CCE = ‘1’:

- High Priority Message Status (MCAN_HPMS)
- Receive FIFO 0 Status (MCAN_RXF0S)
- Receive FIFO 1 Status (MCAN_RXF1S)
- Transmit FIFO/Queue Status (MCAN_TXFQS)
- Transmit Buffer Request Pending (MCAN_TXBRP)
- Transmit Buffer Transmission Occurred (MCAN_TXBTO)
- Transmit Buffer Cancellation Finished (MCAN_TXBCF)
- Transmit Event FIFO Status (MCAN_TXEFS)

The Timeout Counter value MCAN_TOCV.TOC is loaded with the value configured by MCAN_TOCC.TOP when MCAN_CCCR.CCE = ‘1’.

In addition, the state machines of the Tx Handler and Rx Handler are held in idle state while MCAN_CCCR.CCE = ‘1’.

The following registers are only writeable while MCAN_CCCR.CCE = ‘0’

- Transmit Buffer Add Request (MCAN_TXBAR)
- Transmit Buffer Cancellation Request (MCAN_TXBCR)

MCAN_CCCR.TEST and MCAN_CCCR.MON can only be set when MCAN_CCCR.INIT = ‘1’ and MCAN_CCCR.CCE = ‘1’. Both bits may be cleared at any time. MCAN_CCCR.DAR can only be configured when MCAN_CCCR.INIT = ‘1’ and MCAN_CCCR.CCE = ‘1’.

53.5.1.2 Normal Operation

Once the MCAN is initialized and MCAN_CCCR.INIT is cleared, the MCAN synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC are stored into a dedicated Rx Buffer or into Rx FIFO 0 or Rx FIFO 1.

For messages to be transmitted, dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

53.5.1.3 CAN FD Operation

There are two variants in the CAN FD frame format, first the CAN FD frame without bit rate switching where the data field of a CAN frame may be longer than 8 bytes. The second variant is the CAN FD frame where control field, data field, and CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame.

The previously reserved bit in CAN frames with 11-bit identifiers and the first previously reserved bit in CAN frames with 29-bit identifiers will now be decoded as FDF bit. FDF = recessive signifies a CAN FD frame, FDF = dominant signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF, res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. The coding of res = recessive is reserved for future expansion of the protocol. In case the MCAN receives a frame with FDF = recessive and res = recessive, it will signal a Protocol Exception Event by setting bit MCAN_PSR.PXE. When Protocol Exception Handling is enabled (MCAN_CCCR.PXHD = 0), this causes the operation state to change from Receiver (MCAN_PSR.ACT = 2) to Integrating (MCAN_PSR.ACT = 00) at the next sample point. In case Protocol Exception Handling is disabled (MCAN_CCCR.PXHD = 1), the MCAN will treat a recessive res bit as an form error and will respond with an error frame.

PENDET: Pen Contact Detection Enable

0: Pen contact detection disabled.

1: Pen contact detection enabled.

When PENDET = 1, XPOS, YPOS, Z1, Z2 values of ADC_XPOSR, ADC_YPOSR, ADC_PRESSR are automatically cleared when PENS = 0 in ADC_ISR.

NOTSDMA: No TouchScreen DMA

0: XPOS, YPOS, Z1, Z2 are transmitted in ADC_LCDR.

1: XPOS, YPOS, Z1, Z2 are never transmitted in ADC_LCDR, therefore the buffer does not contains touchscreen values.

PENDBC: Pen Detect Debouncing Period

Debouncing period = 2^{PENDBC} ADCCLK periods.

71.1.18 MCAN High Priority Message (HPM)

Issue: Unexpected High Priority Message (HPM) interrupt

There are two configurations where the issue occurs:

Configuration A:

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

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

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

Configuration B:

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

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

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

Workaround:

Configuration A:

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

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

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

Configuration B:

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

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

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

Table 72-4: SAMA5D2 Datasheet Rev. 11267D Revision History (Continued)

Issue Date	Changes
12-May-16	Section 29. “Clock Generator” Section 29.2 “Embedded Characteristics”: AUDIOPLLCK changed to AUDIOPLLCLK Figure 29-1 “Clock Generator Block Diagram”: lines changed to arrows for OSCSEL to multiplexer, for MOSCSEL to multiplexer, and for PLLADIV2 to “PLLA and Divider” block Figure 29-5 “Divider and PLLA Block Diagram”: added PLLADIV2 divider Updated Section 29.8 “Audio PLL”
	Section 30. “Power Management Controller (PMC)” AUDIOPLLCK changed to AUDIOPLLCLK in Section 30.15 “Programmable Clock Controller” and Section 30.16 “Generic Clock Controller” Figure 30-1 “General Clock Block Diagram”: added PLLA block; repositioned PLLACK signal; at bottom of diagram “PCKx” changed to “PCKx (to pads)” Table 30-3 “Register Mapping”: PMC_AUDIO_PLL0 reset value ‘0x0000_0000’ changed to ‘0x0000_00D0’
	Section 30. “Power Management Controller (PMC)” (cont”d) Section 30.22.11 “PMC Master Clock Register”: updated CSS field description Section 30.22.13 “PMC Programmable Clock Register”: added addresses 0xF0014044 and 0xF0014048; updated CSS field description Section 30.22.39 “PMC Audio PLL Control Register 0”: added fields DCO_FILTER (bits 29:28), DCO_GAIN (bits 27:24) and PLLFLT (bits 7:4) Section 30.22.40 “PMC Audio PLL Control Register 1”: updated DIV field description (cont’don next page)