Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M4
Core Size	32-Bit Single-Core
Speed	120MHz
Connectivity	CANbus, Ethernet, IrDA, MMC/SD, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	79
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 16x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4e8ca-au

11. Peripherals

11.1 Peripheral Identifiers

Table 11-1 defines the Peripheral Identifiers of the SAM4E device. A peripheral identifier is required for the control of the peripheral interrupt with the Nested Vectored Interrupt Controller and control of the peripheral clock with the Power Management Controller.

Table 11-1. Peripheral Identifiers

Instance ID	Instance Name	NVIC Interrupt	PMC Clock Control	Instance Description
0	SUPC	X		Supply Controller
1	RSTC	X		Reset Controller
2	RTC	X		Real-time Clock
3	RTT	X		Real-time Timer
4	WDT	X		Watchdog/Dual Watchdog Timer
5	PMC	X		Power Management Controller
6	EFC	X		Enhanced Embedded Flash Controller
7	UART0	X	X	UART 0
8	SMC		X	Static Memory Controller
9	PIOA	X	X	Parallel I/O Controller A
10	PIOB	X	X	Parallel I/O Controller B
11	PIOC	X	X	Parallel I/O Controller C
12	PIOD	X	X	Parallel I/O Controller D
13	PIOE	X	X	Parallel I/O Controller E
14	USART0	X	X	USART 0
15	USART1	X	X	USART 1
16	HSMCI	X	X	Multimedia Card Interface
17	TWI0	X	X	Two-wire Interface 0
18	TWI1	X	X	Two-wire Interface 1
19	SPI	X	X	Serial Peripheral Interface
20	DMAC	X	X	DMAC
21	TC0	X	X	Timer/Counter 0
22	TC1	X	X	Timer/Counter 1
23	TC2	X	X	Timer/Counter 2
24	TC3	X	X	Timer/Counter 3
25	TC4	X	X	Timer/Counter 4
26	TC5	X	X	Timer/Counter 5
27	TC6	X	X	Timer/Counter 6
28	TC7	X	X	Timer/Counter 7
29	TC8	X	X	Timer/Counter 8

12.4.1.12 Exception Mask Registers

The exception mask registers disable the handling of exceptions by the processor. Disable exceptions where they might impact on timing critical tasks.

To access the exception mask registers use the MSR and MRS instructions, or the CPS instruction to change the value of PRIMASK or FAULTMASK. See “MRS” , “MSR” , and “CPS” for more information.

12.4.1.13 Priority Mask Register

Name: PRIMASK

Access: Read-write

Reset: 0x000000000

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
–							PRIMASK

The PRIMASK register prevents the activation of all exceptions with a configurable priority.

- **PRIMASK**

0: No effect

1: Prevents the activation of all exceptions with a configurable priority.

19.5.2 Watchdog Timer Mode Register

Name: WDT_MR

Address: 0x400E1854

Access: Read-write Once

31	30	29	28	27	26	25	24
		WDIDLEHLT	WDDBGHLT	WDD			
23	22	21	20	19	18	17	16
WDD							
15	14	13	12	11	10	9	8
WDDIS	WDRPROC	WDRSTEN	WDFIEN	WDV			
7	6	5	4	3	2	1	0
WDV							

Note: The first write access prevents any further modification of the value of this register, read accesses remain possible.

Note: The WDD and WDV values must not be modified within a period of time of 3 slow clock periods following a restart of the watchdog performed by means of a write access in the WDT_CR register, else the watchdog may trigger an end of period earlier than expected.

- **WDV: Watchdog Counter Value**

Defines the value loaded in the 12-bit Watchdog Counter.

- **WDFIEN: Watchdog Fault Interrupt Enable**

0: A Watchdog fault (underflow or error) has no effect on interrupt.

1: A Watchdog fault (underflow or error) asserts interrupt.

- **WDRSTEN: Watchdog Reset Enable**

0: A Watchdog fault (underflow or error) has no effect on the resets.

1: A Watchdog fault (underflow or error) triggers a Watchdog reset.

- **WDRPROC: Watchdog Reset Processor**

0: If WDRSTEN is 1, a Watchdog fault (underflow or error) activates all resets.

1: If WDRSTEN is 1, a Watchdog fault (underflow or error) activates the processor reset.

- **WDD: Watchdog Delta Value**

Defines the permitted range for reloading the Watchdog Timer.

If the Watchdog Timer value is less than or equal to WDD, writing WDT_CR with WDRSTT = 1 restarts the timer.

If the Watchdog Timer value is greater than WDD, writing WDT_CR with WDRSTT = 1 causes a Watchdog error.

- **WDDBGHLT: Watchdog Debug Halt**

0: The Watchdog runs when the processor is in debug state.

1: The Watchdog stops when the processor is in debug state.

20.5.7 System Controller Wake-up Inputs Register

Name: SUPC_WUIR

Address: 0x400E1820

Access: Read-write

31	30	29	28	27	26	25	24
WKUPT15	WKUPT14	WKUPT13	WKUPT12	WKUPT11	WKUPT10	WKUPT9	WKUPT8
23	22	21	20	19	18	17	16
WKUPT7	WKUPT6	WKUPT5	WKUPT4	WKUPT3	WKUPT2	WKUPT1	WKUPT0
15	14	13	12	11	10	9	8
WKUPEN15	WKUPEN14	WKUPEN13	WKUPEN12	WKUPEN11	WKUPEN10	WKUPEN9	WKUPEN8
7	6	5	4	3	2	1	0
WKUPEN7	WKUPEN6	WKUPEN5	WKUPEN4	WKUPEN3	WKUPEN2	WKUPEN1	WKUPEN0

- **WKUPEN0 - WKUPEN15: Wake-up Input Enable 0 to 15**

0 (DISABLE) = the corresponding wake-up input has no wake-up effect.

1 (ENABLE) = the corresponding wake-up input forces the wake-up of the core power supply.

- **WKUPT0 - WKUPT15: Wake-up Input Type 0 to 15**

0 (LOW) = a low level for a period defined by WKUPDBC on the corresponding wake-up input forces the wake-up of the core power supply.

1 (HIGH) = a high level for a period defined by WKUPDBC on the corresponding wake-up input forces the wake-up of the core power supply.

21. General Purpose Backup Registers (GPBR)

21.1 Description

The System Controller embeds 20 General Purpose Backup Registers (GPBR).

It is possible to generate an immediate clear of the content of general purpose backup registers 0 to 9 (first half), if a low power debounce event is detected on a wakeup pin, WKUP0 or WKUP1. The content of the other general-purpose backup registers (second half) remains unchanged.

To enter this mode of operation, the supply controller module must be programmed accordingly. In supply controller SUPC_WUMR register, LPDBCCLR, LPDBCEN0 and/or LPDBCEN1 bit must be configured to 1 and LPDBC must be other than 0.

If a tamper event has been detected, it is not possible to write into general purpose backup registers while the LPDBCS0 or LPDBCS1 flags are not cleared in supply controller status register SUPC_SR.

21.2 Embedded Characteristics

- Twenty 32-bit General Purpose Backup Registers

26.12.6 SMC NAND Flash Chip Select Configuration Register

Name: CCFG_SMCNFCS

Address: 0x400E0324

Type: Read-write

Reset: 0x0000_0000

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
–	–	–	–	SMC_NFCS3	SMC_NFCS2	SMC_NFCS1	SMC_NFCS0

- **SMC_NFCS0: SMC NAND Flash Chip Select 0 Assignment**

0 = NCS0 is not assigned to a NAND Flash (NANDOE and NANWE not used for NCS0)

1 = NCS0 is assigned to a NAND Flash (NANDOE and NANWE used for NCS0)

- **SMC_NFCS1: SMC NAND Flash Chip Select 1 Assignment**

0 = NCS1 is not assigned to a NAND Flash (NANDOE and NANWE not used for NCS1)

1 = NCS1 is assigned to a NAND Flash (NANDOE and NANWE used for NCS1)

- **SMC_NFCS2: SMC NAND Flash Chip Select 2 Assignment**

0 = NCS2 is not assigned to a NAND Flash (NANDOE and NANWE not used for NCS2)

1 = NCS2 is assigned to a NAND Flash (NANDOE and NANWE used for NCS2)

- **SMC_NFCS3: SMC NAND Flash Chip Select 3 Assignment**

0 = NCS3 is not assigned to a NAND Flash (NANDOE and NANWE not used for NCS3)

1 = NCS3 is assigned to a NAND Flash (NANDOE and NANWE used for NCS3)

27. DMA Controller (DMAC)

27.1 Description

The DMA Controller (DMAC) is an AHB-central DMA controller core that transfers data from a source peripheral to a destination peripheral over one or more AMBA buses. One channel is required for each source/destination pair. In the most basic configuration, the DMAC has one master interface and one channel. The master interface reads the data from a source and writes it to a destination. Two AMBA transfers are required for each DMAC data transfer. This is also known as a dual-access transfer.

The DMAC is programmed via the APB interface.

27.2 Embedded Characteristics

- 1 AHB-Lite Master Interfaces
- DMA Module Supports the Following Transfer Schemes: Peripheral-to-Memory, Memory-to-Peripheral, Peripheral-to-Peripheral and Memory-to-Memory
- Source and Destination Operate independently on BYTE (8-bit), HALF-WORD (16-bit) and WORD (32-bit)
- Supports Hardware and Software Initiated Transfers
- Supports Multiple Buffer Chaining Operations
- Supports Incrementing/decrementing/fixed Addressing Mode Independently for Source and Destination
- Programmable Arbitration Policy, Modified Round Robin and Fixed Priority are Available
- Supports Specified Length and Unspecified Length AMBA AHB Burst Access to Maximize Data Bandwidth
- AMBA APB Interface Used to Program the DMA Controller
- 4 DMA Channels
- 16 External Request Lines
- Embedded FIFO
- Channel Locking and Bus Locking Capability

The DMA Controller can handle the transfer between peripherals and memory and so receives the triggers from the peripherals listed below.

The hardware interface numbers are given in Table 27-1.

Table 27-1. DMA Controller

Instance Name	Channel T/R	DMA Channel HW Interface Number
HSMCI	Transmit/Receive	0
SPI	Transmit	1
SPI	Receive	2
USART0	Transmit	3
USART0	Receive	4
USART1	Transmit	5
USART1	Receive	6
AES	Transmit	11
AES	Receive	12
PWM	Transmit	13

27.7.20 DMAC Write Protect Status Register

Name: DMAC_WPSR

Address: 0x400C01E8

Access: Read-only

Reset: See Table 27-4

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

23	22	21	20	19	18	17	16
WPVSRC							

15	14	13	12	11	10	9	8
WPVSRC							

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	WPVS

- **WPVS: Write Protect Violation Status**

0 = No Write Protect Violation has occurred since the last read of the DMAC_WPSR register.

1 = A Write Protect Violation has occurred since the last read of the DMAC_WPSR register. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

- **WPVSRC: Write Protect Violation Source**

When WPVS is active, this field indicates the write-protected register (through address offset or code) in which a write access has been attempted.

Note: Reading DMAC_WPSR automatically clears all fields.

28.15 Static Memory Controller (SMC) User Interface

The SMC is programmed using the registers listed in Table 28-7. For each chip select, a set of 4 registers is used to program the parameters of the external device connected on it. In Table 28-7, “CS_number” denotes the chip select number. 16 bytes (0x10) are required per chip select.

The user must complete writing the configuration by writing any one of the SMC_MODE registers.

Table 28-7. Register Mapping

Offset	Register	Name	Access	Reset
0x10 x CS_number + 0x00	SMC Setup Register	SMC_SETUP	Read-write	0x01010101
0x10 x CS_number + 0x04	SMC Pulse Register	SMC_PULSE	Read-write	0x01010101
0x10 x CS_number + 0x08	SMC Cycle Register	SMC_CYCLE	Read-write	0x00030003
0x10 x CS_number + 0x0C	SMC Mode Register	SMC_MODE	Read-write	0x10000003
0x80	SMC OCMS MODE Register	SMC_OCMS	Read-write	0x00000000
0x84	SMC OCMS KEY1 Register	SMC_KEY1	Write once	0x00000000
0x88	SMC OCMS KEY2 Register	SMC_KEY2	Write once	0x00000000
0xE4	SMC Write Protect Mode Register	SMC_WPMR	Read-write	0x00000000
0xE8	SMC Write Protect Status Register	SMC_WPSR	Read-only	0x00000000
0xEC-0xFC	Reserved	-	-	-

30.2.16.10 PMC Master Clock Register

Name: PMC_MCKR

Address: 0x400E0430

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
–	–	–	PLLADIV2	–	–	–	–
7	6	5	4	3	2	1	0
–	PRES			–	–	CSS	

This register can only be written if the WPEN bit is cleared in “PMC Write Protect Mode Register” .

- **CSS: Master Clock Source Selection**

Value	Name	Description
0	SLOW_CLK	Slow Clock is selected
1	MAIN_CLK	Main Clock is selected
2	PLLA_CLK	PLLA Clock is selected

- **PRES: Processor Clock Prescaler**

Value	Name	Description
0	CLK_1	Selected clock
1	CLK_2	Selected clock divided by 2
2	CLK_4	Selected clock divided by 4
3	CLK_8	Selected clock divided by 8
4	CLK_16	Selected clock divided by 16
5	CLK_32	Selected clock divided by 32
6	CLK_64	Selected clock divided by 64
7	CLK_3	Selected clock divided by 3

- **PLLADIV2: PLLA Divisor by 2**

PLLADIV2	PLLA Clock Division
0	PLLA clock frequency is divided by 1.
1	PLLA clock frequency is divided by 2.

30.2.16.15 PMC Status Register

Name: PMC_SR

Address: 0x400E0468

Access: Read-only

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

23	22	21	20	19	18	17	16
–	–	–	FOS	CFDS	CFDEV	MOSCRCS	MOSCSELS

15	14	13	12	11	10	9	8
–	–	–	–	–	PCKRDY2	PCKRDY1	PCKRDY0

7	6	5	4	3	2	1	0
OSCSELS	–	–	–	MCKRDY	–	LOCKA	MOSCXTS

- **MOSCXTS: Main XTAL Oscillator Status**

0 = Main XTAL oscillator is not stabilized.

1 = Main XTAL oscillator is stabilized.

- **LOCKA: PLLA Lock Status**

0 = PLLA is not locked

1 = PLLA is locked.

- **MCKRDY: Master Clock Status**

0 = Master Clock is not ready.

1 = Master Clock is ready.

- **OSCSELS: Slow Clock Oscillator Selection**

0 = Internal slow clock RC oscillator is selected.

1 = External slow clock 32 kHz oscillator is selected.

- **PCKRDYx: Programmable Clock Ready Status**

0 = Programmable Clock x is not ready.

1 = Programmable Clock x is ready.

- **MOSCSELS: Main Oscillator Selection Status**

0 = Selection is in progress.

1 = Selection is done.

- **MOSCRCS: Main On-Chip RC Oscillator Status**

0 = Main on-chip RC oscillator is not stabilized.

1 = Main on-chip RC oscillator is stabilized.

- **CFDEV: Clock Failure Detector Event**

30.2.16.24 PMC Peripheral Clock Status Register 1

Name: PMC_PCSR1

Address: 0x400E0508

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
PID47	PID46	PID45	PID44	PID43	PID42	PID41	PID40
7	6	5	4	3	2	1	0
PID39	PID38	PID37	PID36	PID35	PID34	PID33	PID32

- **PIDx: Peripheral Clock x Status**

0 = The corresponding peripheral clock is disabled.

1 = The corresponding peripheral clock is enabled.

Note: To get PIDx, refer to identifiers as defined in the section “Peripheral Identifiers” in the product datasheet.

32. Controller Area Network (CAN) Programmer Datasheet

32.1 Description

The CAN controller provides all the features required to implement the serial communication protocol CAN defined by Robert Bosch GmbH, the CAN specification as referred to by ISO/11898A (2.0 Part A and 2.0 Part B) for high speeds and ISO/11519-2 for low speeds. The CAN Controller is able to handle all types of frames (Data, Remote, Error and Overload) and achieves a bitrate of 1 Mbps.

CAN controller accesses are made through configuration registers. 8 independent message objects (mailboxes) are implemented.

Any mailbox can be programmed as a reception buffer block (even non-consecutive buffers). For the reception of defined messages, one or several message objects can be masked without participating in the buffer feature. An interrupt is generated when the buffer is full. According to the mailbox configuration, the first message received can be locked in the CAN controller registers until the application acknowledges it, or this message can be discarded by new received messages.

Any mailbox can be programmed for transmission. Several transmission mailboxes can be enabled in the same time. A priority can be defined for each mailbox independently.

An internal 16-bit timer is used to stamp each received and sent message. This timer starts counting as soon as the CAN controller is enabled. This counter can be reset by the application or automatically after a reception in the last mailbox in Time Triggered Mode.

The CAN controller offers optimized features to support the Time Triggered Communication (TTC) protocol.

32.2 Embedded Characteristics

- Fully Compliant with CAN 2.0 Part A and 2.0 Part B
- Bit Rates up to 1Mbit/s
- 8 Object Oriented Mailboxes with the Following Properties:
  - CAN Specification 2.0 Part A or 2.0 Part B Programmable for Each Message
  - Object Configurable in Receive (with Overwrite or Not) or Transmit Modes
  - Independent 29-bit Identifier and Mask Defined for Each Mailbox
  - 32-bit Access to Data Registers for Each Mailbox Data Object
  - Uses a 16-bit Timestamp on Receive and Transmit Messages
  - Hardware Concatenation of ID Masked Bitfields To Speed Up Family ID Processing
- 16-bit Internal Timer for Timestamping and Network Synchronization
- Programmable Reception Buffer Length up to 8 Mailbox Objects
- Priority Management between Transmission Mailboxes
- Autobaud and Listening Mode
- Low Power Mode and Programmable Wake-up on Bus Activity or by the Application
- Data, Remote, Error and Overload Frame Handling
- Write Protected Registers

32.9.8 CAN Timestamp Register

Name: CAN_TIMESTP

Address: 0x4001001C (0), 0x4001401C (1)

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
MTIMESTAMP							
7	6	5	4	3	2	1	0
MTIMESTAMP							

- **MTIMESTAMP: Timestamp**

This field carries the value of the internal CAN controller 16-bit timer value at the start or end of frame.

If the TEOF bit is cleared in the CAN_MR register, the internal Timer Counter value is captured in the MTIMESTAMP field at each start of frame else the value is captured at each end of frame. When the value is captured, the TSTP flag is set in the CAN_SR register. If the TSTP mask in the CAN_IMR register is set, an interrupt is generated while TSTP flag is set in the CAN_SR register. This flag is cleared by reading the CAN_SR register.

Note: The CAN_TIMESTP register is reset when the CAN is disabled then enabled thanks to the CANEN bit in the CAN_MR.

Figure 39-5. SD Memory Card Bus Topology

The SD Memory Card bus includes the signals listed in Table 39-5.

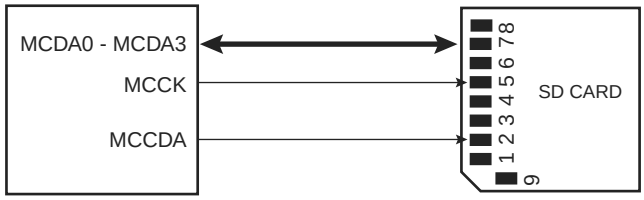
Table 39-5. SD Memory Card Bus Signals

Pin Number	Name	Type[1]	Description	HSMCI Pin Name[2] (Slot z)
1	CD/DAT[3]	I/O/PP	Card detect/ Data line Bit 3	MCDz3
2	CMD	PP	Command/response	MCCDz
3	VSS1	S	Supply voltage ground	VSS
4	VDD	S	Supply voltage	VDD
5	CLK	I/O	Clock	MCCK
6	VSS2	S	Supply voltage ground	VSS
7	DAT[0]	I/O/PP	Data line Bit 0	MCDz0
8	DAT[1]	I/O/PP	Data line Bit 1 or Interrupt	MCDz1
9	DAT[2]	I/O/PP	Data line Bit 2	MCDz2

Notes: 1. I: input, O: output, PP: Push Pull, OD: Open Drain.

2. When several HSMCI (x HSMCI) are embedded in a product, MCCK refers to HSMCIx_CK, MCCDA to HSMCIx_CDA, MCDAy to HSMCIx_DAy.

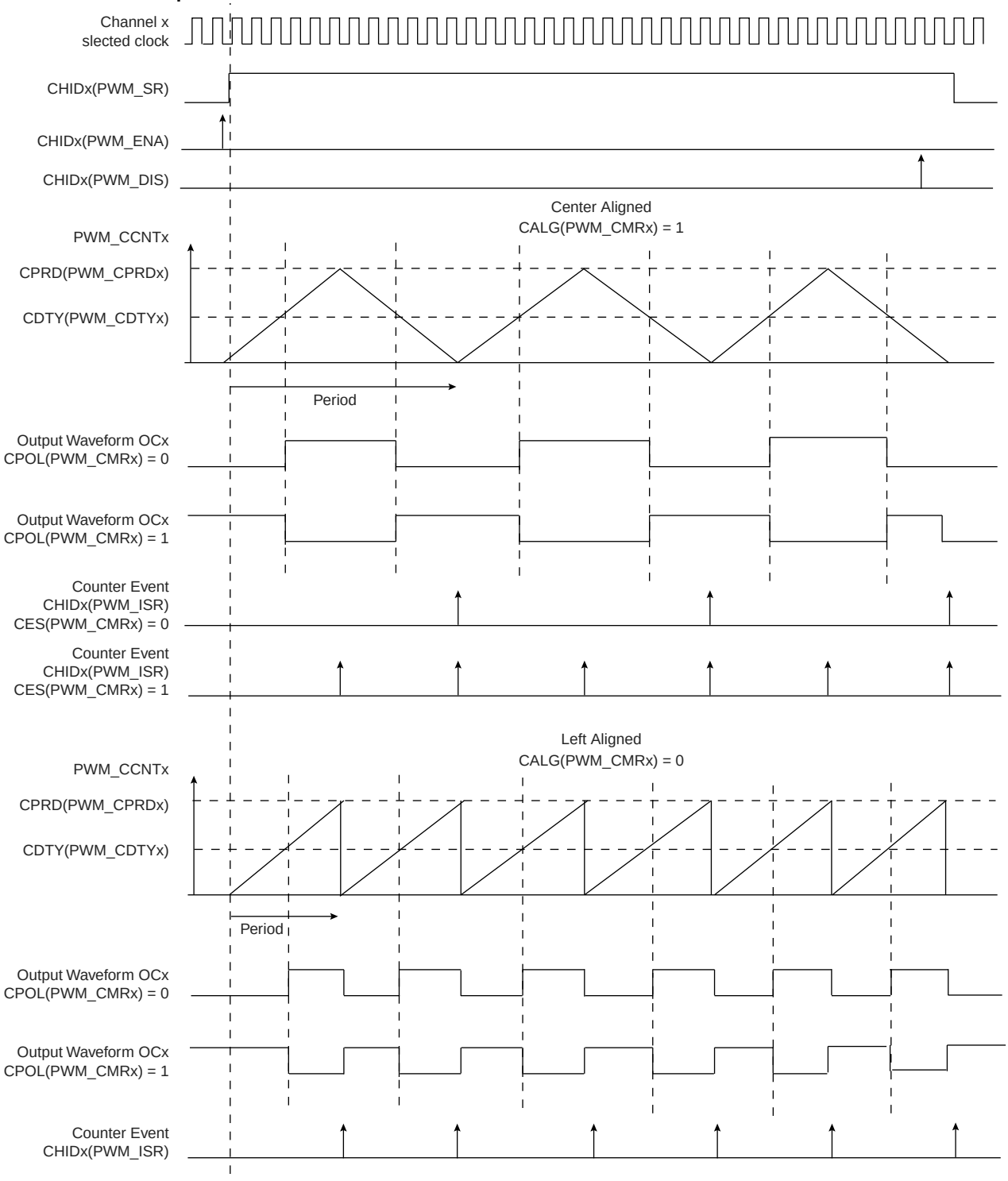
Figure 39-6. SD Card Bus Connections with One Slot



Note: When several HSMCI (x HSMCI) are embedded in a product, MCCK refers to HSMCIx_CK, MCCDA to HSMCIx_CDA MCDAy to HSMCIx_DAy.

When the HSMCI is configured to operate with SD memory cards, the width of the data bus can be selected in the HSMCI_SDCR register. Clearing the SDCBUS bit in this register means that the width is one bit; setting it means that the width is four bits. In the case of High Speed MultiMedia cards, only the data line 0 is used. The other data lines can be used as independent PIOs.

Figure 40-5. Waveform Properties



44.4 Signal Interface

The GMAC includes the following signal interfaces

- MII to an external PHY
- MDIO interface for external PHY management
- Slave APB interface for accessing GMAC registers
- Master AHB interface for memory access

Table 44-1. GMAC connections in the different modes

Signal Name	Function	MII
GTXCK	Transmit Clock or Reference Clock	TXCK
GTXEN	Transmit Enable	TXEN
GTX[3..0]	Transmit Data	TXD[3:0]
GTXER	Transmit Coding Error	TXER
GRXCK	Receive Clock	RXCK
GRXDV	Receive Data Valid	RXDV
GRX[3..0]	Receive Data	RXD[3:0]
GRXER	Receive Error	RXER
GCRS	Carrier Sense and Data Valid	CRS
GCOL	Collision Detect	COL
GMDC	Management Data Clock	MDC
GMDIO	Management Data Input/Output	MDIO

44.7.68 1519 to Maximum Byte Frames Received Register

Name: GMAC_TMXBFR

Address: 0x40034180

Access: Read-only

31	30	29	28	27	26	25	24
NFRX							

23	22	21	20	19	18	17	16
NFRX							

15	14	13	12	11	10	9	8
NFRX							

7	6	5	4	3	2	1	0
NFRX							

- **NFRX: 1519 to Maximum Byte Frames Received without Error**

This register counts the number of 1519 byte or above frames successfully received without error. Maximum frame size is determined by the Network Configuration Register bit 8 (1536 maximum frame size) or bit 3 (jumbo frame size). Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory. See: Section 44.7.2 ”Network Configuration Register”.

45.7 Digital-to-Analog Converter Controller (DACC) User Interface

Table 45-3. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Control Register	DACC_CR	Write-only	–
0x04	Mode Register	DACC_MR	Read-write	0x00000000
0x08	Reserved	–	–	–
0x0C	Reserved	–	–	–
0x10	Channel Enable Register	DACC_CHER	Write-only	–
0x14	Channel Disable Register	DACC_CHDR	Write-only	–
0x18	Channel Status Register	DACC_CHSR	Read-only	0x00000000
0x1C	Reserved	–	–	–
0x20	Conversion Data Register	DACC_CDR	Write-only	0x00000000
0x24	Interrupt Enable Register	DACC_IER	Write-only	–
0x28	Interrupt Disable Register	DACC_IDR	Write-only	–
0x2C	Interrupt Mask Register	DACC_IMR	Read-only	0x00000000
0x30	Interrupt Status Register	DACC_ISR	Read-only	0x00000000
0x94	Analog Current Register	DACC_ACR	Read-write	0x00000000
0xE4	Write Protect Mode register	DACC_WPMR	Read-write	0x00000000
0xE8	Write Protect Status register	DACC_WPSR	Read-only	0x00000000
...	...	...	...	...
0xEC - 0xFC	Reserved	–	–	–

46.7.2 Static Performance Characteristics

In the tables that follow, the LSB is relative to analog scale:

- Single Ended (ex: ADVREF=3.0V),
  - Gain = 1, LSB = (3.0V / 4096) = 732 uV
  - Gain = 2, LSB = (1.5V / 4096) = 366 uV
  - Gain = 4, LSB = (750 mV / 4096) = 183 uV
- Differential (ex: ADVREF=3.0V),
  - Gain = 0.5, LSB = (6.0V / 4096) = 1465 uV
  - Gain = 1, LSB = (3.0V / 4096) = 732 uV
  - Gain = 2, LSB = (1.5V / 4096) = 366 uV

Table 46-32. INL, DNL, 12-bit mode, VDDIN 2.4V to 3.6V Supply Voltage Conditions for All Gains

Parameter	Conditions	Min	Typ	Max	Units
Resolution			12		bit
Integral Non-linearity (INL)	f_{ADC} = 2 MHz; differential mode or single mode, Gain = xx	-2	+/-1	2	LSB
Differential Non-linearity (DNL)	f_{ADC} = 2 MHz; differential mode or single mode, Gain = xx	-1	+/-0.5	1	LSB

Table 46-33. Gain and Error Offset, 12-bit Mode, VDDIN 2.4V to 3.6V Supply Voltage Conditions

Parameter	Conditions	Min	Typ	Max	Units
Offset Error (Without DAC Compensation)	Differential mode, Gain = 0.5	-18	—	18	LSB
	Differential mode, Gain = 1	-35	—	35	
	Differential mode, Gain = 2	-60	—	60	
	Single Ended Gain = 1	-18	—	18	
	Single Ended Gain = 2	-35	—	35	
	Single Ended Gain = 4	-60	—	60	
Gain Error, after calibration	Differential mode, Gain = xx	-50	—	0	
	Single Ended Gain = xx	-25	—	0	