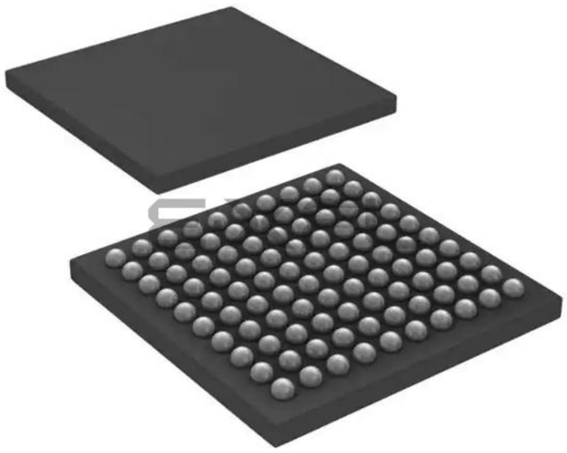
Microchip Technology - ATSAM4SA16CB-CFN Datasheet



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	EBI/EMI, I²C, IrDA, Memory Card, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	79
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	160K 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 ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-VFBGA
Supplier Device Package	100-VFBGA (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4sa16cb-cfn

Table 11-1. Peripheral Identifiers (Continued)

Instance ID	Instance Name	NVIC Interrupt	PMC Clock Control	Instance Description
29	ADC	X	X	Analog-to-Digital Converter
30	DACC	X	X	Digital-to-Analog Converter Controller
31	PWM	X	X	Pulse Width Modulation
32	CRCCU	X	X	CRC Calculation Unit
33	ACC	X	X	Analog Comparator Controller
34	UDP	X	X	USB Device Port

11.2 Peripheral Signal Multiplexing on I/O Lines

The SAM4S features two PIO controllers on 64-pin versions (PIOA and PIOB) or three PIO controllers on the 100-pin version (PIOA, PIOB and PIOC), that multiplex the I/O lines of the peripheral set.

The SAM4S 64-pin and 100-pin PIO controllers control up to 32 lines. Each line can be assigned to one of three peripheral functions: A, B or C. The multiplexing tables in the following tables define how the I/O lines of the peripherals A, B and C are multiplexed on the PIO Controllers. The column “Comments” has been inserted in this table for the user’s own comments; it may be used to track how pins are defined in an application.

Note that some peripheral functions which are output only, might be duplicated within the tables.

12.4.1.3 Core Registers

Figure 12-2. Processor Core Registers

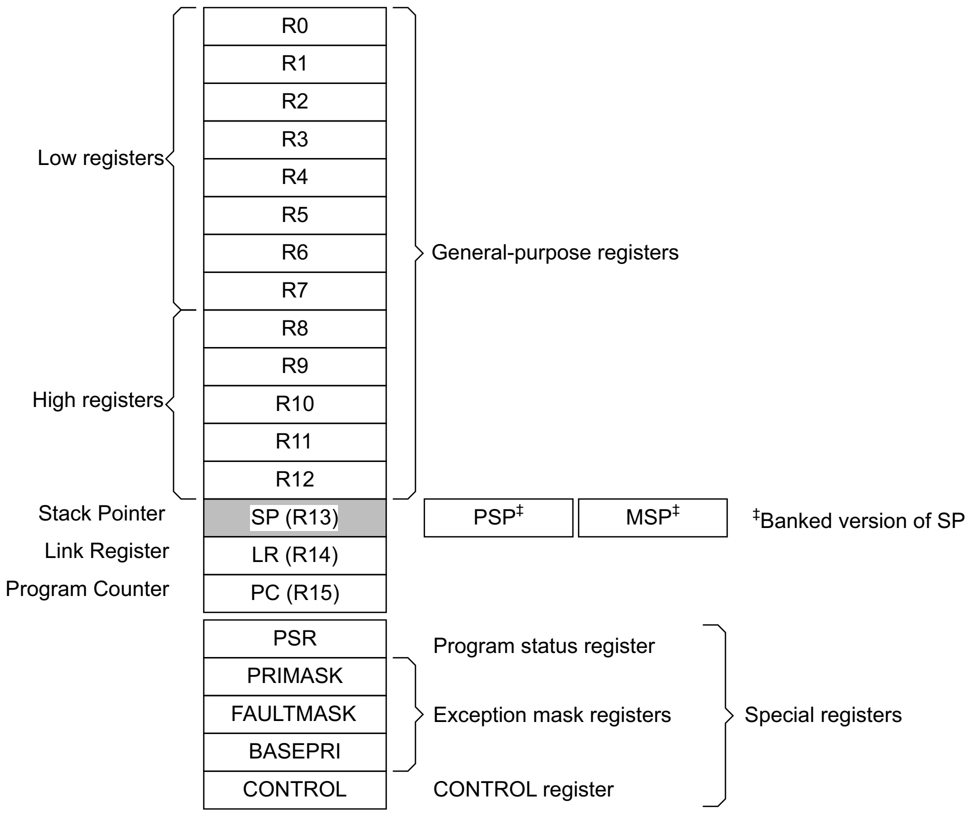


Table 12-2. Core Processor Registers

Register	Name	Access[1]	Required Privilege[2]	Reset
General-purpose registers	R0–R12	Read/Write	Either	Unknown
Stack Pointer	MSP	Read/Write	Privileged	See description
Stack Pointer	PSP	Read/Write	Either	Unknown
Link Register	LR	Read/Write	Either	0xFFFFFFFF
Program Counter	PC	Read/Write	Either	See description
Program Status Register	PSR	Read/Write	Privileged	0x01000000
Application Program Status Register	APSR	Read/Write	Either	0x00000000
Interrupt Program Status Register	IPSR	Read-only	Privileged	0x00000000
Execution Program Status Register	EPSR	Read-only	Privileged	0x01000000
Priority Mask Register	PRIMASK	Read/Write	Privileged	0x00000000
Fault Mask Register	FAULTMASK	Read/Write	Privileged	0x00000000
Base Priority Mask Register	BASEPRI	Read/Write	Privileged	0x00000000
Control Register	CONTROL	Read/Write	Privileged	0x00000000

Notes: 1. Describes access type during program execution in thread mode and Handler mode. Debug access can differ.
2. An entry of Either means privileged and unprivileged software can access the register.

Tail-chaining

This mechanism speeds up exception servicing. On completion of an exception handler, if there is a pending exception that meets the requirements for exception entry, the stack pop is skipped and control transfers to the new exception handler.

Late-arriving

This mechanism speeds up preemption. If a higher priority exception occurs during state saving for a previous exception, the processor switches to handle the higher priority exception and initiates the vector fetch for that exception. State saving is not affected by late arrival because the state saved is the same for both exceptions. Therefore the state saving continues uninterrupted. The processor can accept a late arriving exception until the first instruction of the exception handler of the original exception enters the execute stage of the processor. On return from the exception handler of the late-arriving exception, the normal tail-chaining rules apply.

Exception Entry

An Exception entry occurs when there is a pending exception with sufficient priority and either the processor is in Thread mode, or the new exception is of a higher priority than the exception being handled, in which case the new exception preempts the original exception.

When one exception preempts another, the exceptions are nested.

Sufficient priority means that the exception has more priority than any limits set by the mask registers, see “Exception Mask Registers” . An exception with less priority than this is pending but is not handled by the processor.

When the processor takes an exception, unless the exception is a tail-chained or a late-arriving exception, the processor pushes information onto the current stack. This operation is referred as *stacking* and the structure of eight data words is referred to as *stack frame*.

```
                                ; with top halfword of R5, subtracts second from
                                ; first, adds R6, writes to R0
        SMLSDX  R1, R3, R2, R0  ; Multiplies bottom halfword of R3 with top
                                ; halfword of R2, multiplies top halfword of R3
                                ; with bottom halfword of R2, subtracts second from
                                ; first, adds R0, writes to R1
        SMLSLD  R3, R6, R2, R7  ; Multiplies bottom halfword of R6 with bottom
                                ; halfword of R2, multiplies top halfword of R6
                                ; with top halfword of R2, subtracts second from
                                ; first, adds R6:R3, writes to R6:R3
        SMLSLDX R3, R6, R2, R7  ; Multiplies bottom halfword of R6 with top
                                ; halfword of R2, multiplies top halfword of R6
                                ; with bottom halfword of R2, subtracts second from
                                ; first, adds R6:R3, writes to R6:R3.
```

12.9.1.6 System Control Register

Name: SCB_SCR

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
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	–	–	SEVONPEND	–	SLEEPDEEP	SLEEPONEXIT	–

- **SEVONPEND: Send Event on Pending Bit**

0: Only enabled interrupts or events can wake up the processor; disabled interrupts are excluded.

1: Enabled events and all interrupts, including disabled interrupts, can wake up the processor.

When an event or an interrupt enters the pending state, the event signal wakes up the processor from WFE. If the processor is not waiting for an event, the event is registered and affects the next WFE.

The processor also wakes up on execution of an SEV instruction or an external event.

- **SLEEPDEEP: Sleep or Deep Sleep**

Controls whether the processor uses sleep or deep sleep as its low power mode:

0: Sleep.

1: Deep sleep.

- **SLEEPONEXIT: Sleep-on-exit**

Indicates sleep-on-exit when returning from the Handler mode to the Thread mode:

0: Do not sleep when returning to Thread mode.

1: Enter sleep, or deep sleep, on return from an ISR.

Setting this bit to 1 enables an interrupt-driven application to avoid returning to an empty main application.

17.5.2 Watchdog Timer Mode Register

Name: WDT_MR

Address: 0x400E1454

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 three slow clock periods following a restart of the watchdog performed by a write access in WDT_CR. Any modification will cause the watchdog to 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.

- **WDDIS: Watchdog Disable**

0: Enables the Watchdog Timer.

1: Disables the Watchdog Timer.

- **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, setting bit WDT_CR.WDRSTT restarts the timer.

If the Watchdog Timer value is greater than WDD, setting bit WDT_CR.WDRSTT causes a watchdog error.

- **OSCSEL: 32-kHz Oscillator Selection Status**

0 (RC): The slow clock, SLCK, is generated by the embedded 32 kHz RC oscillator.

1 (CRYST): The slow clock, SLCK, is generated by the 32 kHz crystal oscillator.

- **LPDBCS0: Low-power Debouncer Wake-up Status on WKUP0 (cleared on read)**

0 (NO): No wake-up due to the assertion of the WKUP0 pin has occurred since the last read of SUPC_SR.

1 (PRESENT): At least one wake-up due to the assertion of the WKUP0 pin has occurred since the last read of SUPC_SR.

- **LPDBCS1: Low-power Debouncer Wake-up Status on WKUP1 (cleared on read)**

0 (NO): No wake-up due to the assertion of the WKUP1 pin has occurred since the last read of SUPC_SR.

1 (PRESENT): At least one wake-up due to the assertion of the WKUP1 pin has occurred since the last read of SUPC_SR.

- **WKUPISx: WKUPx Input Status (cleared on read)**

0 (DIS): The corresponding wake-up input is disabled, or was inactive at the time the debouncer triggered a wake-up event.

1 (EN): The corresponding wake-up input was active at the time the debouncer triggered a wake-up event since the last read of SUPC_SR.

- **FARG: Flash Command Argument**

GETD, GLB, GGPB, STUI, SPUI, GCALB, WUS, EUS, STUS, SPUS, EA	Commands requiring no argument, including Erase all command	FARG is meaningless, must be written with 0
ES	Erase sector command	FARG must be written with any page number within the sector to be erased
EPA	Erase pages command	FARG[1:0] defines the number of pages to be erased The start page must be written in FARG[15:2]. FARG[1:0] = 0: Four pages to be erased. FARG[15:2] = Page_Number / 4 FARG[1:0] = 1: Eight pages to be erased. FARG[15:3] = Page_Number / 8, FARG[2]=0 FARG[1:0] = 2: Sixteen pages to be erased. FARG[15:4] = Page_Number / 16, FARG[3:2]=0 FARG[1:0] = 3: Thirty-two pages to be erased. FARG[15:5] = Page_Number / 32, FARG[4:2]=0 Refer to Table 20-4 “EEFC_FCR.FARG Field for EPA Command”.
WP, WPL, EWP, EWPL	Programming commands	FARG must be written with the page number to be programmed
SLB, CLB	Lock bit commands	FARG defines the page number to be locked or unlocked
SGPB, CGPB	GPNVM commands	FARG defines the GPNVM number to be programmed

- **FKEY: Flash Writing Protection Key**

Value	Name	Description
0x5A	PASSWD	The 0x5A value enables the command defined by the bits of the register. If the field is written with a different value, the write is not performed and no action is started.

21. Fast Flash Programming Interface (FFPI)

21.1 Description

The Fast Flash Programming Interface (FFPI) provides parallel high-volume programming using a standard gang programmer. The parallel interface is fully handshaked and the device is considered to be a standard EEPROM. Additionally, the parallel protocol offers an optimized access to all the embedded Flash functionalities.

Although the Fast Flash Programming mode is a dedicated mode for high volume programming, this mode is not designed for in-situ programming.

21.2 Embedded Characteristics

- Programming Mode for High-volume Flash Programming Using Gang Programmer
  - Offers Read and Write Access to the Flash Memory Plane
  - Enables Control of Lock Bits and General-purpose NVM Bits
  - Enables Security Bit Activation
  - Disabled Once Security Bit is Set
- Parallel Fast Flash Programming Interface
  - Provides an 16-bit Parallel Interface to Program the Embedded Flash
  - Full Handshake Protocol

22.5.5 Cache Controller Maintenance Register 0

Name:	CMCC_MAINT0
Address:	0x4007C020
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
–	–	–	–	–	–	–	INVALL

- **INVALL: Cache Controller Invalidate All**

0: No effect.

1: All cache entries are invalidated.

For more details about VID/PID for End Product/Systems, please refer to the Vendor ID form available from the USB Implementers Forum on www.usb.org.

Atmel provides an INF example to see the device as a new serial port and also provides another custom driver used by the SAM-BA application: atm6124.sys. Refer to the application note “USB Basic Application”, Atmel literature number 6123, for more details.

24.5.3.1 Enumeration Process

The USB protocol is a master/slave protocol. This is the host that starts the enumeration sending requests to the device through the control endpoint. The device handles standard requests as defined in the USB Specification.

Table 24-3. Handled Standard Requests

Request	Definition
GET_DESCRIPTOR	Returns the current device configuration value.
SET_ADDRESS	Sets the device address for all future device access.
SET_CONFIGURATION	Sets the device configuration.
GET_CONFIGURATION	Returns the current device configuration value.
GET_STATUS	Returns status for the specified recipient.
SET_FEATURE	Set or Enable a specific feature.
CLEAR_FEATURE	Clear or Disable a specific feature.

The device also handles some class requests defined in the CDC class.

Table 24-4. Handled Class Requests

Request	Definition
SET_LINE_CODING	Configures DTE rate, stop bits, parity and number of character bits.
GET_LINE_CODING	Requests current DTE rate, stop bits, parity and number of character bits.
SET_CONTROL_LINE_STATE	RS-232 signal used to tell the DCE device the DTE device is now present.

Unhandled requests are STALLed.

24.5.3.2 Communication Endpoints

There are two communication endpoints and endpoint 0 is used for the enumeration process. Endpoint 1 is a 64-byte Bulk OUT endpoint and endpoint 2 is a 64-byte Bulk IN endpoint. SAM-BA Boot commands are sent by the host through endpoint 1. If required, the message is split by the host into several data payloads by the host driver.

If the command requires a response, the host can send IN transactions to pick up the response.

31.6.16 PIO Interrupt Mask Register

Name: PIO_IMR

Address: 0x400E0E48 (PIOA), 0x400E1048 (PIOB), 0x400E1248 (PIOC)

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: Input Change Interrupt Mask**

0: Input change interrupt is disabled on the I/O line.

1: Input change interrupt is enabled on the I/O line.

32.9.15 SSC Interrupt Disable Register

Name: SSC_IDR

Address: 0x40004048

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**
–	–	–	–	RXSYN	TXSYN	CP1	CP0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXBUFF	ENDRX	OVRUN	RXRDY	TXBUFE	ENDTX	TXEMPTY	TXRDY

- **TXRDY: Transmit Ready Interrupt Disable**

0: No effect.

1: Disables the Transmit Ready Interrupt.

- **TXEMPTY: Transmit Empty Interrupt Disable**

0: No effect.

1: Disables the Transmit Empty Interrupt.

- **ENDTX: End of Transmission Interrupt Disable**

0: No effect.

1: Disables the End of Transmission Interrupt.

- **TXBUFE: Transmit Buffer Empty Interrupt Disable**

0: No effect.

1: Disables the Transmit Buffer Empty Interrupt.

- **RXRDY: Receive Ready Interrupt Disable**

0: No effect.

1: Disables the Receive Ready Interrupt.

- **OVRUN: Receive Overrun Interrupt Disable**

0: No effect.

1: Disables the Receive Overrun Interrupt.

- **ENDRX: End of Reception Interrupt Disable**

0: No effect.

1: Disables the End of Reception Interrupt.

36.6.7 Modem Mode

The USART features Modem mode, which enables control of the signals: DTR (Data Terminal Ready), DSR (Data Set Ready), RTS (Request to Send), CTS (Clear to Send), DCD (Data Carrier Detect) and RI (Ring Indicator). While operating in Modem mode, the USART behaves as a DTE (Data Terminal Equipment) as it drives DTR and RTS and can detect level change on DSR, DCD, CTS and RI.

Setting the USART in Modem mode is performed by writing the USART_MODE field in US_MR to the value 0x3. While operating in Modem mode, the USART behaves as though in Asynchronous mode and all the parameter configurations are available.

Table 36-13 gives the correspondence of the USART signals with modem connection standards.

Table 36-13. Circuit References

USART Pin	V24	CCITT	Direction
TXD	2	103	From terminal to modem
RTS	4	105	From terminal to modem
DTR	20	108.2	From terminal to modem
RXD	3	104	From modem to terminal
CTS	5	106	From terminal to modem
DSR	6	107	From terminal to modem
DCD	8	109	From terminal to modem
RI	22	125	From terminal to modem

The control of the DTR output pin is performed by writing a 1 to the DTRDIS and DTREN bits respectively in US_CR. The disable command forces the corresponding pin to its inactive level, i.e., high. The enable command forces the corresponding pin to its active level, i.e., low. The RTS output pin is automatically controlled in this mode.

The level changes are detected on the RI, DSR, DCD and CTS pins. If an input change is detected, the RIIC, DSRIC, DCDIC and CTSIC bits in US_CSR are set respectively and can trigger an interrupt. The status is automatically cleared when US_CSR is read. Furthermore, the CTS automatically disables the transmitter when it is detected at its inactive state. If a character is being transmitted when the CTS rises, the character transmission is completed before the transmitter is actually disabled.

36.7.10 USART Interrupt Mask Register (SPI_MODE)

Name: US_IMR (SPI_MODE)

Address: 0x40024010 (0), 0x40028010 (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
–	–	–	RXBUFF	TXBUFE	UNRE	TXEMPTY	–

7	6	5	4	3	2	1	0
–	–	OVRE	ENDTX	ENDRX	–	TXRDY	RXRDY

This configuration is relevant only if USART_MODE = 0xE or 0xF in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

- **RXRDY: RXRDY Interrupt Mask**

- **TXRDY: TXRDY Interrupt Mask**

- **ENDRX: End of Receive Buffer Interrupt Mask**

- **ENDTX: End of Transmit Buffer Interrupt Mask**

- **OVRE: Overrun Error Interrupt Mask**

- **TXEMPTY: TXEMPTY Interrupt Mask**

- **UNRE: SPI Underrun Error Interrupt Mask**

- **TXBUFE: Transmit Buffer Empty Interrupt Mask**

- **RXBUFF: Receive Buffer Full Interrupt Mask**

40.7.10 UDP Endpoint Control and Status Register (CONTROL_BULK)

Name:	UDP_CSRx [x = 0..7] (CONTROL_BULK)
Address:	0x40034030
Access:	Read/Write

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

23	22	21	20	19	18	17	16
RXBYTECNT							

15	14	13	12	11	10	9	8
EPEDS	–	–	–	DTGLE	EPTYPE		

7	6	5	4	3	2	1	0
DIR	RX_DATA_BK1	FORCESTALL	TXPKTRDY	STALLSENT	RXSETUP	RX_DATA_BK0	TXCOMP

WARNING: Due to synchronization between MCK and UDPCK, the software application must wait for the end of the write operation before executing another write by polling the bits which must be set/cleared.

As an example, to perform a control operation on the endpoint without modifying the status flags while accessing the control bits and fields of this register, the status flag bits must first be defined with the "No effect" value '1'. Once the overall UDP_CSR value is defined, the register can be written and then the synchronization wait procedure must be executed.

- **TXCOMP: Generates an IN Packet with Data Previously Written in the DPR**

This flag generates an interrupt while it is set to one.

Write (cleared by the firmware):

0: Clear the flag, clear the interrupt

1: No effect

Read (Set by the USB peripheral):

0: Data IN transaction has not been acknowledged by the Host

1: Data IN transaction is achieved, acknowledged by the Host

After having issued a Data IN transaction setting TXPKTRDY, the device firmware waits for TXCOMP to be sure that the host has acknowledged the transaction.

- **RX_DATA_BK0: Receive Data Bank 0**

This flag generates an interrupt while it is set to one.

Write (cleared by the firmware):

0: Notify USB peripheral device that data have been read in the FIFO's Bank 0.

1: To leave the read value unchanged.

Read (Set by the USB peripheral):

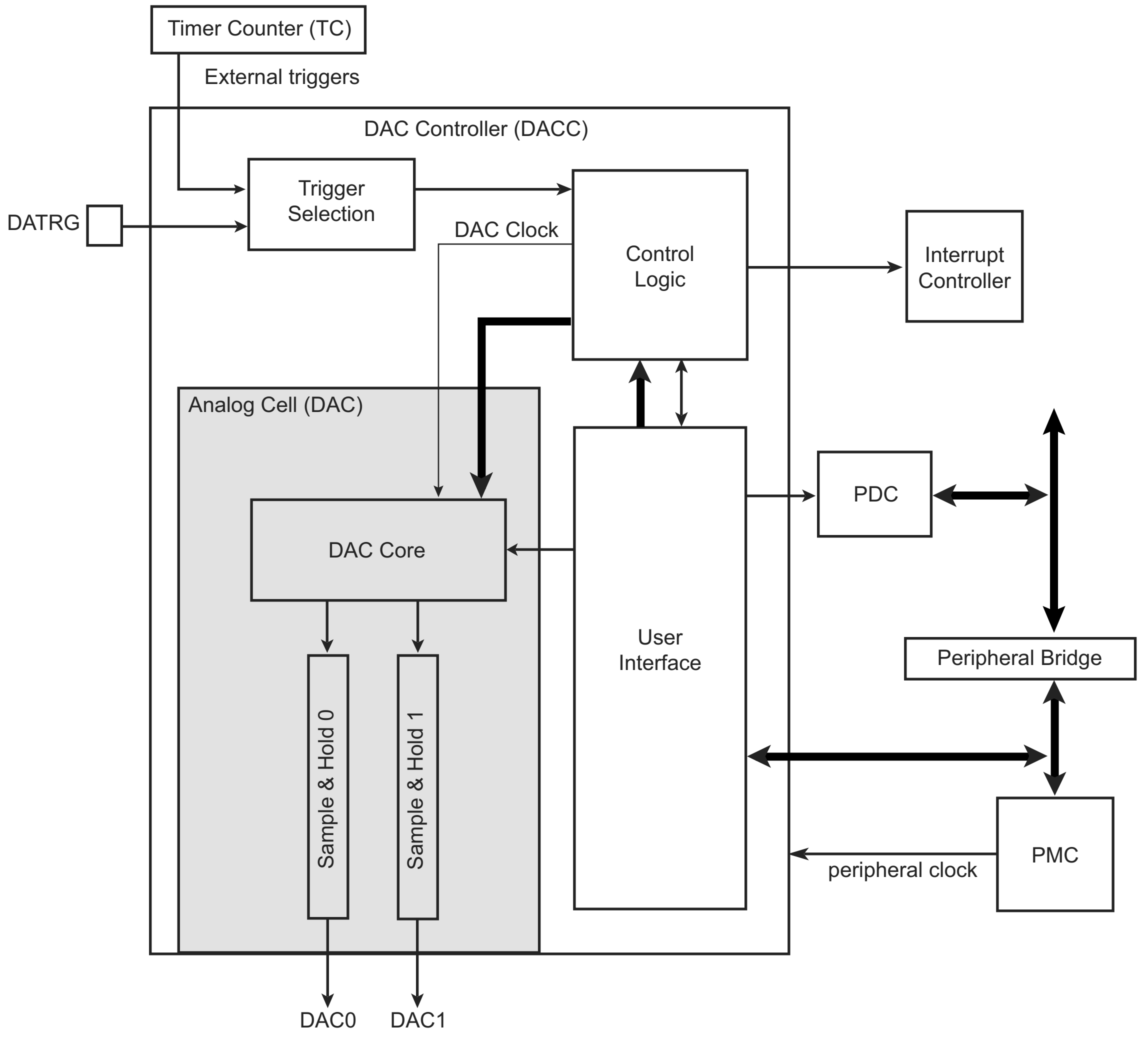
0: No data packet has been received in the FIFO's Bank 0.

1: A data packet has been received, it has been stored in the FIFO's Bank 0.

When the device firmware has polled this bit or has been interrupted by this signal, it must transfer data from the FIFO to the microcontroller memory. The number of bytes received is available in RXBYTCENT field. Bank 0 FIFO values are read

43.3 Block Diagram

Figure 43-1. DACC Block Diagram



43.4 Signal Description

Table 43-1. DACC Pin Description

Pin Name	Description
DAC0–DAC1	Analog output channels
DATRG	External triggers

43.5 Product Dependencies

43.5.1 Power Management

The user must first enable the DAC Controller Clock in the Power Management Controller (PMC) before using the DACC.

The DACC becomes active as soon as a conversion is requested and at least one channel is enabled. The DACC is automatically deactivated when no channels are enabled.

Table 44-4. 1.2V Voltage Regulator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDIN}	DC Input Voltage	(4)	1.6	3.3	3.6	V
V_{DDOUT}	DC Output Voltage	Normal Mode	–	1.2	–	V
		Standby Mode	–	0	–	
$V_{O(accuracy)}$	Output Voltage Accuracy	I_{LOAD} = 0.8mA to 80mA (after trimming)	-3		3	%
I_{LOAD}	Maximum DC Output Current	$V_{DDIN} > 1.8V$	–	–	80	mA
		$V_{DDIN} \leq 1.8V$	–	–	40	
$I_{LOAD-START}$	Maximum Peak Current during Startup	(3)	–	–	400	
$V_{DROPOUT}$	Dropout Voltage	V_{DDIN} = 1.6V, I_{LOAD} = Max	–	400	–	mV
V_{LINE}	Line Regulation	V_{DDIN} from 2.7V to 3.6V; I_{LOAD} MAX	–	10	30	mV
$V_{LINE-TR}$	Transient Line Regulation	V_{DDIN} from 2.7V to 3.6V; $t_r = t_f$ = 5µs; I_{LOAD} Max	–	50	150	
V_{LOAD}	Load Regulation	$V_{DDIN} \geq 1.8V$; I_{LOAD} = 10% to 90% MAX	–	20	40	mV
$V_{LOAD-TR}$	Transient Load Regulation	$V_{DDIN} \geq 1.8V$; I_{LOAD} = 10% to 90% MAX; $t_r = t_f$ = 5 µs	–	50	150	mV
I_Q	Quiescent Current	Normal Mode, I_{LOAD} = 0 mA	–	5	–	µA
		Normal Mode, I_{LOAD} = 80 mA	–	500	–	
		Standby Mode	–	500	1	
CD_{IN}	Input Decoupling Capacitor	(1)	–	4.7	–	µF
CD_{OUT}	Output Decoupling Capacitor	(2)	1.85	2.2	5.9	µF
		ESR	0.1		10	Ω
t_{on}	Turn-on Time	CD_{OUT} = 2.2µF, V_{DDOUT} reaches 1.2V (± 3%)	–	300	–	µs
t_{off}	Turn-off Time	CD_{OUT} = 2.2µF	–	–	40	ms

Notes:
1. A 4.7µF or higher ceramic capacitor must be connected between VDDIN and the closest GND pin of the device. This large decoupling capacitor is mandatory to reduce startup current, improving transient response and noise rejection.
2. To ensure stability, an external 2.2 µF output capacitor, CD_{OUT} must be connected between the VDDOUT and the closest GND pin of the device. The ESR (Equivalent Series Resistance) of the capacitor must be in the range 0.1 to 10 Ω.
Solid tantalum, and multilayer ceramic capacitors are all suitable as output capacitor.
A 100 nF bypass capacitor between VDDOUT and the closest GND pin of the device helps decrease output noise and improves the load transient response.
3. Defined as the current needed to charge external bypass/decoupling capacitor network.
4. See Section 5.2.2 “VDDIO Versus VDDIN”

48.1.2 Flash

Issue: Incorrect Flash Read May Occur Depending on VDDIO Voltage and Flash Wait State

Flash read issues leading to wrong instruction fetch or incorrect data read may occur under the following operating conditions:

VDDIO < 2.4V and Flash wait state(1) ≥ 1

If the core clock frequency does not require the use of the Flash wait state (2) (FWS = 0 in EEFC_FMR) or if only data reads are performed on the Flash (e.g., if the code is running out of SRAM), there are no constraints on VDDIO voltage. The usable voltage range for VDDIO is defined in Table 44-3 “DC Characteristics”.

Notes: 1. Defined by the FWS field in EEFC_FMR register.
2. See Section 44.12.9 “Embedded Flash Characteristics” for the maximum core clock frequency at zero (0) wait state.

Workaround: Two workarounds are available:

1. Reduce the device speed to decrease the number of wait states to 0.
2. Copy the code from Flash to SRAM at 0 wait states and then run the code out of SRAM.

The issue will be corrected in the next device revision, Marketing Revision Level B (MRL B). Please contact your local Sales Representative for further details.

Issue: Read Error after a GPNVM or Lock Bit Writing

The sequence below leads to a bad read value.

Fail sequence is:

Read Flash @ address XXX

Programming Flash: Write GPNVM or Lock Bit instructions

Read Flash @ address XXX

Workaround: A dummy read at another address needs to be included in the sequence.

Sequence is:

Read Flash @ address XXX

Programming Flash: Write GPNVM or Lock Bit instructions

Read Flash @ address YYY (dummy read)

Read Flash @ address XXX

48.1.3 Watchdog

Issue: Watchdog Not Stopped in Wait Mode

When the Watchdog is enabled and the bit WAITMODE = 1 is used to enter wait mode, the watchdog is not halted. If the time spent in Wait Mode is longer than the Watchdog time-out, the device will be reset if Watchdog reset is enabled.

Workaround: When entering wait mode, the Wait For Event (WFE) instruction of the processor Cortex-M4 must be used with the SLEEPDEEP of the System Control Register (SCB_SCR) of the Cortex-M = 0.

Table 49-6. SAM4S Datasheet Rev. 11100F 29-Jan-14 Revision History (Continued)

Doc. Date	Changes
	Section 38. “High Speed MultiMedia Card Interface (HSMCI)” Changed PDCFBYTE to FBYTE in Section 9.6 “WRITE_SINGLE_BLOCK/WRITE_MULTIPLE_BLOCK Operation using DMA Controller” and in Section 9.8 “READ_SINGLE_BLOCK/READ_MULTIPLE_BLOCK Operation using DMA Controller”. Section 38.13 “Register Write Protection”: changed title (was “Write Protection Registers”); revised content In Section 38.14.2 “HSMCI Mode Register”, PDCMODE bit description, corrected reference to MCI Mode Register to HSMCI Status Register. In Section 38.14.7 “HSMCI Block Register”, BLKLEN bit description, removed sentence on its accessibility in HSMCI Mode Register. Section 38.14.17 “HSMCI Write Protection Mode Register”: modified register name (was HSMCI Write Protect Mode Register); replaced list of protectable registers with cross-reference to section “Register Write Protection” Section 38.14.18 “HSMCI Write Protection Status Register”: modified register name (was HSMCI Write Protect Status Register) and updated description
	Section 40. “USB Device Port (UDP)” Section 40.2 “Embedded Characteristics”: replaced bullet “Integrated Pull-up on DP” with “Integrated Pull-up on DPP”, added bullet “Integrated Pull-down on DDM” Section 40.6.3.6 “Entering in Suspend State”: replaced “must drain less than 500uA” with “must drain no more than 2.5 mA” Section 40.6.3 “Controlling Device States”: replaced “may not consume more than 500 µA” with “must not consume more than 2.5 mA” Table 40-6 ”Register Mapping”: corrected reset values for for UDP-FDR0..Y. Updated note (1). Section 40.4.2 “Power Management”: added detail on fast RC. Changed register names: ● Section 40.7.10 old: UDP Endpoint Control and Status Register (Control, Bulk Interrupt Endpoints), new: “UDP Endpoint Control and Status Register (CONTROL_BULK)” ● Section 40.7.11 old: UDP Endpoint Control and Status Register (Isochronous Endpoints), new: “UDP Endpoint Control and Status Register (ISOCHRONOUS)”.
	Section 41. “Analog Comparator Controller (ACC)” Section 41.1 “Description” Updated section for clarity. Figure 41-1 “Analog Comparator Controller Block Diagram”: Updated for clarity. Section 41.6 “Functional Description”, Section 41.6.2 “Analog Settings” and Section 41.6.4 “Fault Mode”: Updated for clarity. Replaced section “Write Protection System” with Section 41.6.5 “Register Write Protection”. Updated Section 41.7.8 “ACC Write Protection Mode Register” and Section 41.7.9 “ACC Write Protection Status Register”. Bit 0 name in Section 41.7.9 “ACC Write Protection Status Register” changed from WPROTERR to WPVS.
	Section 42. “Analog-to-Digital Converter (ADC)” Section 42.1 “Description”: Added sentence: The last channel is internally connected by a temperature sensor. Section 42.2 “Embedded Characteristics”: updated section with new characteristics Section 42.6.3 “Conversion Resolution”: Modified content to limit information on 12-bit resolution. Section 42.6.14 “Register Write Protection”: Reworked content. Section 42.7.3 “ADC Channel Sequence 1 Register” and Section 42.7.4 “ADC Channel Sequence 2 Register”: modified max channel number to 15. Section 42.7.12 “ADC Interrupt Status Register”: updated ‘ENDRX’ and ‘RXBUFF’ bit descriptions. Section 42.7.15 “ADC Compare Window Register”: updated ‘LOWTHRES’ and ‘HIGHTHRES’ field descriptions. Section 42.7.20 “ADC Write Protection Mode Register” and Section 42.7.21 “ADC Write Protection Status Register”: Modified register names (from Write Protect to Write Protection). Reworked field descriptions.