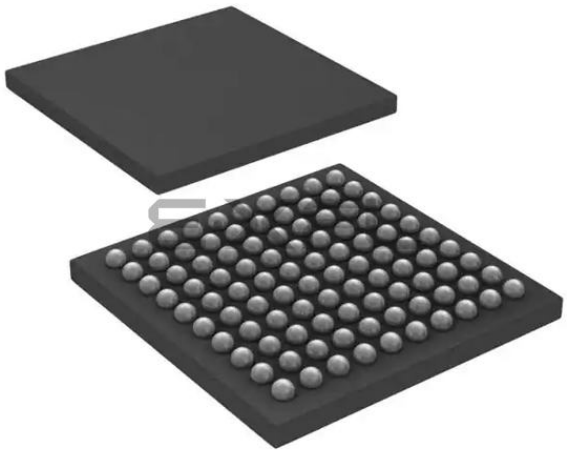


Welcome to E-XFL.COM

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	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-VFBGA
Supplier Device Package	100-VFBGA (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4s16ca-cfu

4.1.4 100-lead LQFP Pinout

Table 4-1. SAM4SD32/SD16/SA16/S16/S8/S4/S2 100-lead LQFP Pinout

1	ADVREF	26	GND	51	TDI/PB4	76	TDO/TRACESWO/PB5
2	GND	27	VDDIO	52	PA6/PGMNOE	77	JTAGSEL
3	PB0/AD4	28	PA16/PGMD4	53	PA5/PGMRDY	78	PC18
4	PC29/AD13	29	PC7	54	PC28	79	TMS/SWDIO/PB6
5	PB1/AD5	30	PA15/PGMD3	55	PA4/PGMNCMD	80	PC19
6	PC30/AD14	31	PA14/PGMD2	56	VDDCORE	81	PA31
7	PB2/AD6	32	PC6	57	PA27/PGMD15	82	PC20
8	PC31	33	PA13/PGMD1	58	PC8	83	TCK/SWCLK/PB7
9	PB3/AD7	34	PA24/PGMD12	59	PA28	84	PC21
10	VDDIN	35	PC5	60	NRST	85	VDDCORE
11	VDDOUT	36	VDDCORE	61	TST	86	PC22
12	PA17/PGMD5/AD0	37	PC4	62	PC9	87	ERASE/PB12
13	PC26	38	PA25/PGMD13	63	PA29	88	DDM/PB10
14	PA18/PGMD6/AD1	39	PA26/PGMD14	64	PA30	89	DDP/PB11
15	PA21/PGMD9/AD8	40	PC3	65	PC10	90	PC23
16	VDDCORE	41	PA12/PGMD0	66	PA3	91	VDDIO
17	PC27	42	PA11/PGMM3	67	PA2/PGMEN2	92	PC24
18	PA19/PGMD7/AD2	43	PC2	68	PC11	93	PB13/DAC0
19	PC15/AD11	44	PA10/PGMM2	69	VDDIO	94	PC25
20	PA22/PGMD10/AD9	45	GND	70	GND	95	GND
21	PC13/AD10	46	PA9/PGMM1	71	PC14	96	PB8/XOUT
22	PA23/PGMD11	47	PC1	72	PA1/PGMEN1	97	PB9/PGMCK/XIN
23	PC12/AD12	48	PA8/XOUT32/PGMM0	73	PC16	98	VDDIO
24	PA20/PGMD8/AD3	49	PA7/XIN32/ PGMNVALID	74	PA0/PGMEN0	99	PB14/DAC1
25	PC0	50	VDDIO	75	PC17	100	VDDPLL

5. Power Considerations

5.1 Power Supplies

The SAM4S has several types of power supply pins:

- VDDCORE pins: Power the core, the first flash rail and the embedded memories and peripherals. Voltage ranges from 1.08 to 1.32 V.
- VDDIO pins: Power the peripheral I/O lines (input/output buffers), the second Flash rail, USB transceiver, backup part, 32 kHz crystal oscillator and oscillator pads. Voltage ranges from 1.62 to 3.6 V.
- VDDIN pin: Voltage regulator input, ADC, DAC and analog comparator power supply. Voltage ranges from 1.62 to 3.6 V.
- VDDPLL pin: Powers the PLLA, PLLB, the fast RC and the 3 to 20 MHz oscillator. Voltage ranges from 1.08 to 1.32 V.

5.2 Power-up Considerations

5.2.1 VDDIO Versus VDDCORE

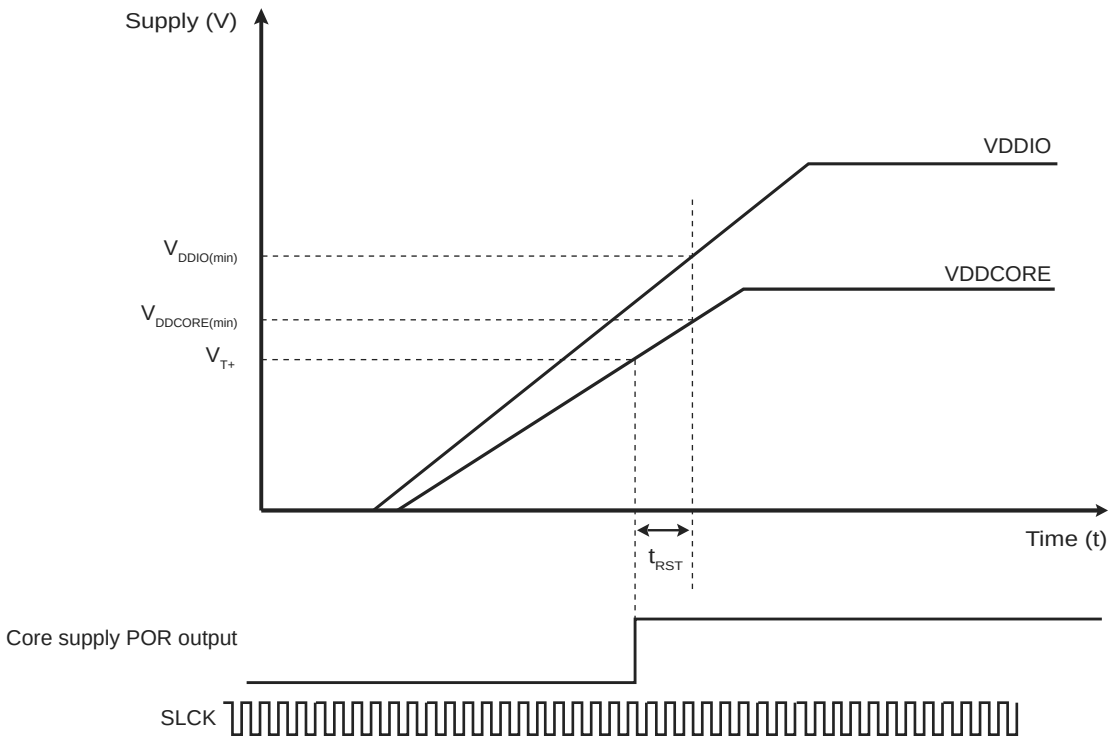
V_{DDIO} must always be higher than or equal to V_{DDCORE}.

V_{DDIO} must reach its minimum operating voltage (1.62 V) before V_{DDCORE} has reached $V_{DDCORE(min)}$. The minimum slope for V_{DDCORE} is defined by ($V_{DDCORE(min)}$ - V_{T+}) / t_{RST}.

If V_{DDCORE} rises at the same time as V_{DDIO}, the V_{DDIO} rising slope must be higher than or equal to 8.8 V/ms.

If VDDCORE is powered by the internal regulator, all power-up considerations are met.

Figure 5-1. VDDCORE and VDDIO Constraints at Startup



12.4.1.9 Application Program Status Register

Name: APSR

Access: Read/Write

Reset: 0x000000000

31	30	29	28	27	26	25	24
N	Z	C	V	Q	–		

23	22	21	20	19	18	17	16
–				GE[3:0]			

15	14	13	12	11	10	9	8
–							

7	6	5	4	3	2	1	0
–							

The APSR contains the current state of the condition flags from previous instruction executions.

- **N: Negative Flag**

0: Operation result was positive, zero, greater than, or equal

1: Operation result was negative or less than.

- **Z: Zero Flag**

0: Operation result was not zero

1: Operation result was zero.

- **C: Carry or Borrow Flag**

Carry or borrow flag:

0: Add operation did not result in a carry bit or subtract operation resulted in a borrow bit

1: Add operation resulted in a carry bit or subtract operation did not result in a borrow bit.

- **V: Overflow Flag**

0: Operation did not result in an overflow

1: Operation resulted in an overflow.

- **Q: DSP Overflow and Saturation Flag**

Sticky saturation flag:

0: Indicates that saturation has not occurred since reset or since the bit was last cleared to zero

1: Indicates when an SSAT or USAT instruction results in saturation.

This bit is cleared to zero by software using an MRS instruction.

- **GE[19:16]: Greater Than or Equal Flags**

See “SEL” for more information.

12.6.5.13 SSUB16 and SSUB8

Signed Subtract 16 and Signed Subtract 8

Syntax

```
op{cond}{Rd,} Rn, Rm
```

where:

op is any of:

SSUB16 Performs two 16-bit signed integer subtractions.

SSUB8 Performs four 8-bit signed integer subtractions.

cond is an optional condition code, see “Conditional Execution” .

Rd is the destination register.

Rn is the first operand register.

Rm is the second operand register.

Operation

Use these instructions to change endianness of data:

The SSUB16 instruction:

1. Subtracts each halfword from the second operand from the corresponding halfword of the first operand
2. Writes the difference result of two signed halfwords in the corresponding halfword of the destination register.

The SSUB8 instruction:

1. Subtracts each byte of the second operand from the corresponding byte of the first operand
2. Writes the difference result of four signed bytes in the corresponding byte of the destination register.

Restrictions

Do not use SP and do not use PC.

Condition Flags

These instructions do not change the flags.

Examples

```
SSUB16 R1, R0     ; Subtracts halfwords in R0 from corresponding halfword
                  ; of R1 and writes to corresponding halfword of R1
SSUB8  R4, R0, R5 ; Subtracts bytes of R5 from corresponding byte in
                  ; R0, and writes to corresponding byte of R4.
```

```
                        ; 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.
```

12.10.1 System Timer (SysTick) User Interface

Table 12-34. System Timer (SYST) Register Mapping

Offset	Register	Name	Access	Reset
0xE000E010	SysTick Control and Status Register	SYST_CSR	Read/Write	0x00000000
0xE000E014	SysTick Reload Value Register	SYST_RVR	Read/Write	Unknown
0xE000E018	SysTick Current Value Register	SYST_CVR	Read/Write	Unknown
0xE000E01C	SysTick Calibration Value Register	SYST_CALIB	Read-only	0x000030D4

The table below shows the encodings for the TEX, C, B, and S access permission bits.

Table 12-36. TEX, C, B, and S Encoding

TEX	C	B	S	Memory Type	Shareability	Other Attributes
b000	0	0	x(1)	Strongly-ordered	Shareable	–
		1	x(1)	Device	Shareable	–
	1	0	0	Normal	Not shareable	Outer and inner write-through. No write allocate.
			1		Shareable	
		1	0	Normal	Not shareable	Outer and inner write-back. No write allocate.
			1		Shareable	
b001	0	0	0	Normal	Not shareable	Outer and inner noncacheable.
			1		Shareable	
		1	x(1)	Reserved encoding		–
	1	0	x(1)	Implementation defined attributes.		–
		1	0	Normal	Not shareable	Outer and inner write-back. Write and read allocate.
			1		Shareable	
b010	0	0	x(1)	Device	Not shareable	Nonshared Device.
		1	x(1)	Reserved encoding		–
	1	x(1)	x(1)	Reserved encoding		–
b1BB	A	A	0	Normal	Not shareable	Cached memory BB = outer policy, AA = inner policy.
			1		Shareable	

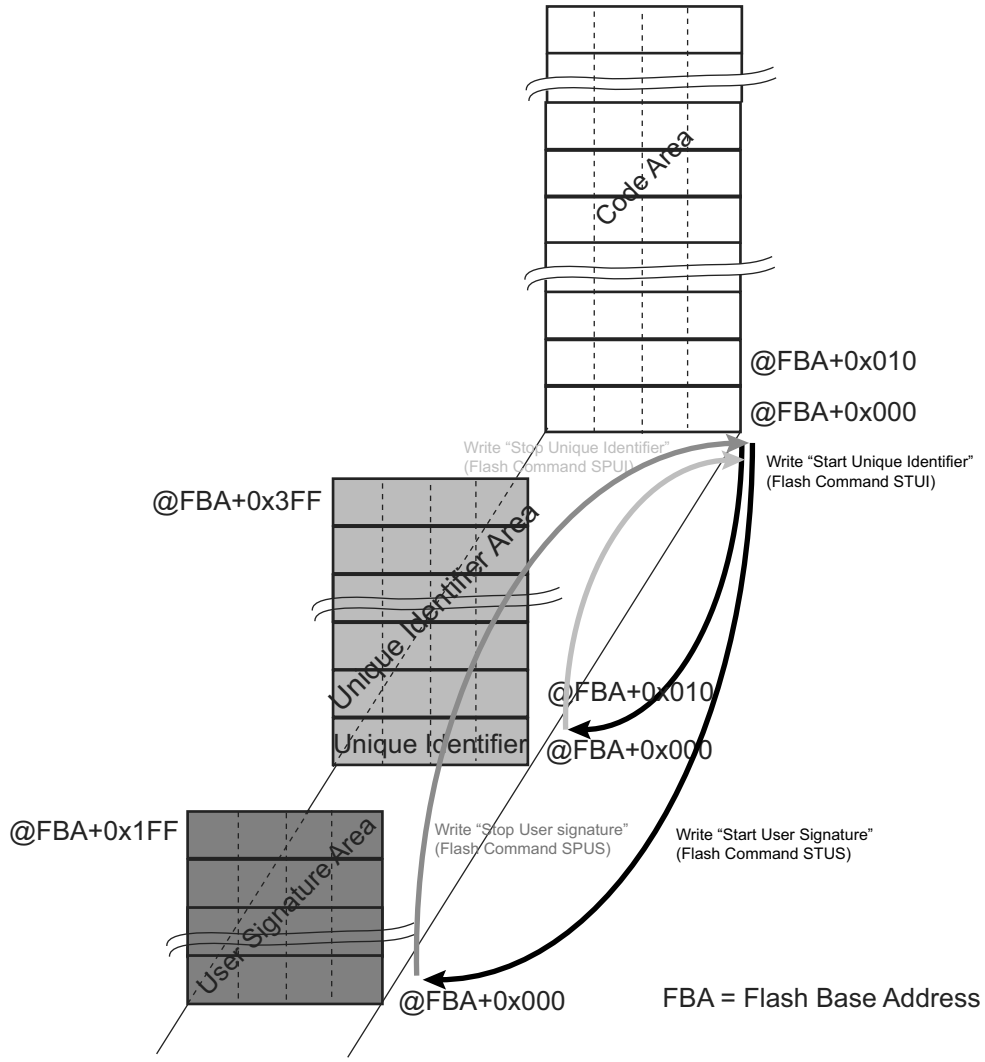
Note: 1. The MPU ignores the value of this bit.

Table 12-37 shows the cache policy for memory attribute encodings with a TEX value is in the range 4–7.

Table 12-37. Cache Policy for Memory Attribute Encoding

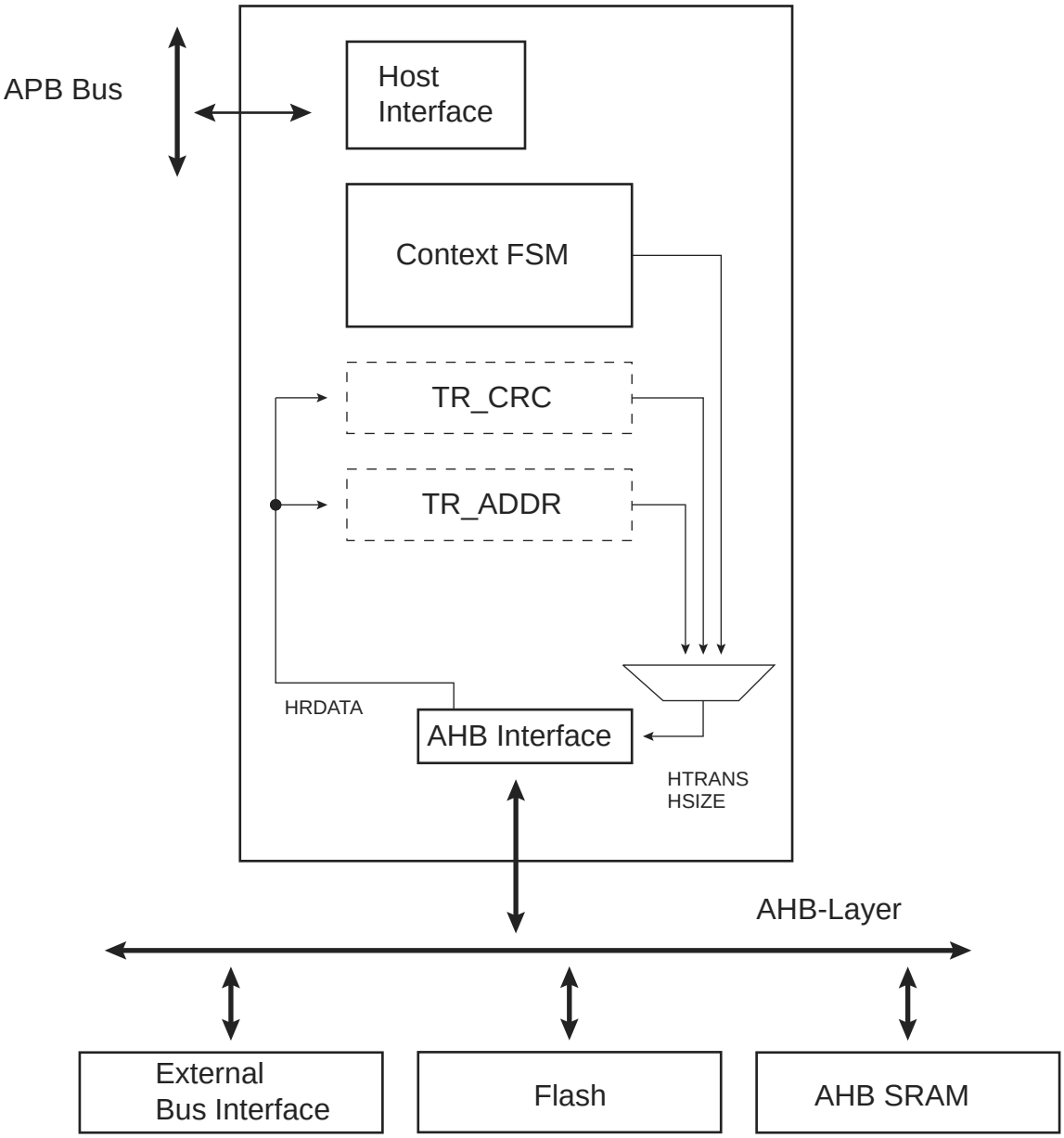
Encoding, AA or BB	Corresponding Cache Policy
00	Non-cacheable
01	Write back, write and read allocate
10	Write through, no write allocate
11	Write back, no write allocate

Figure 20-1. Flash Memory Areas



23.3 CRCCU Block Diagram

Figure 23-1. Block Diagram



29. Power Management Controller (PMC)

29.1 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 Cortex-M4 processor.

The Supply Controller selects between the 32 kHz RC oscillator or the slow crystal oscillator. The unused oscillator is disabled automatically so that power consumption is optimized.

By default, at startup, the chip runs out of the master clock using the fast RC oscillator running at 4 MHz.

The user can trim the 8 and 12 MHz RC oscillator frequencies by software.

29.2 Embedded Characteristics

The PMC provides the following clocks:

- MCK, the Master Clock, programmable from a few hundred Hz to the maximum operating frequency of the device. It is available to the modules running permanently, such as the Enhanced Embedded Flash Controller.
- Processor Clock (HCLK) , automatically switched off when entering the processor in Sleep Mode
- Free-running processor Clock (FCLK)
- The Cortex-M4 SysTick external clock
- UDP Clock (UDPCK), required by USB Device Port operations
- Peripheral Clocks, provided to the embedded peripherals (USART, SPI, TWI, TC, etc.) and independently controllable.
- Programmable Clock Outputs (PCKx), selected from the clock generator outputs to drive the device PCK pins

The PMC also provides the following operations on clocks:

- A main crystal oscillator clock failure detector
- A frequency counter on main clock and an on-the-fly adjustable main RC oscillator frequency

29.17.12PMC USB Clock Register

Name: PMC_USB

Address: 0x400E0438

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
–	–	–	–	USBDIV			

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	USBS

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

- **USBS: USB Input Clock Selection**

0: USB Clock Input is PLLA.

1: USB Clock Input is PLLB

- **USBDIV: Divider for USB Clock**

USB Clock is Input clock divided by USBDIV + 1.

34.8.7 TWI Interrupt Enable Register

Name: TWI_IER

Address: 0x40018024 (0), 0x4001C024 (1)

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
TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK

7	6	5	4	3	2	1	0
–	OVRE	GACC	SVACC	–	TXRDY	RXRDY	TXCOMP

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

0: No effect.

1: Enables the corresponding interrupt.

- **TXCOMP: Transmission Completed Interrupt Enable**
- **RXRDY: Receive Holding Register Ready Interrupt Enable**
- **TXRDY: Transmit Holding Register Ready Interrupt Enable**
- **SVACC: Slave Access Interrupt Enable**
- **GACC: General Call Access Interrupt Enable**
- **OVRE: Overrun Error Interrupt Enable**
- **NACK: Not Acknowledge Interrupt Enable**
- **ARBLST: Arbitration Lost Interrupt Enable**
- **SCL_WS: Clock Wait State Interrupt Enable**
- **EOSACC: End Of Slave Access Interrupt Enable**
- **ENDRX: End of Receive Buffer Interrupt Enable**
- **ENDTX: End of Transmit Buffer Interrupt Enable**
- **RXBUFF: Receive Buffer Full Interrupt Enable**
- **TXBUFE: Transmit Buffer Empty Interrupt Enable**

36.7.2 USART Control Register (SPI_MODE)

Name: US_CR (SPI_MODE)

Address: 0x40024000 (0), 0x40028000 (1)

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	RCS	FCS	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	RSTSTA
7	6	5	4	3	2	1	0
TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX	–	–

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

- **RSTRX: Reset Receiver**

0: No effect.

1: Resets the receiver.

- **RSTTX: Reset Transmitter**

0: No effect.

1: Resets the transmitter.

- **RXEN: Receiver Enable**

0: No effect.

1: Enables the receiver, if RXDIS is 0.

- **RXDIS: Receiver Disable**

0: No effect.

1: Disables the receiver.

- **TXEN: Transmitter Enable**

0: No effect.

1: Enables the transmitter if TXDIS is 0.

- **TXDIS: Transmitter Disable**

0: No effect.

1: Disables the transmitter.

- **RSTSTA: Reset Status Bits**

0: No effect.

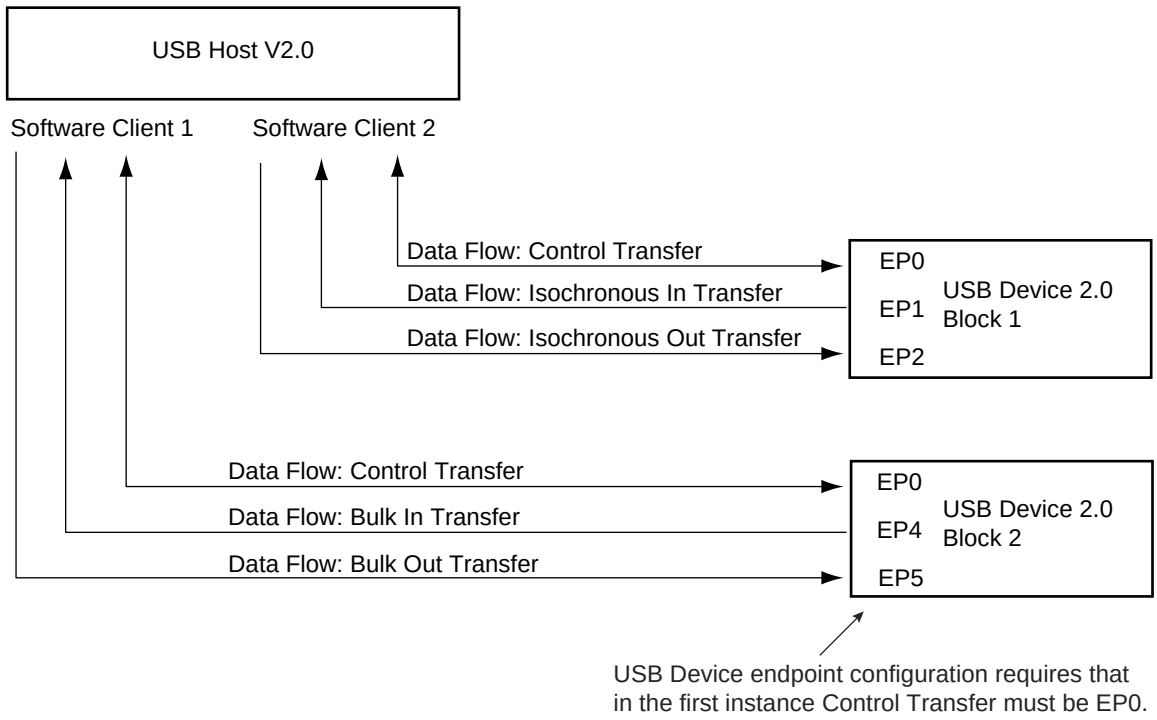
1: Resets the status bits OVRE, UNRE in US_CSR.

40.6 Functional Description

40.6.1 USB 2.0 Full-speed Introduction

The USB 2.0 full-speed provides communication services between host and attached USB devices. Each device is offered with a collection of communication flows (pipes) associated with each endpoint. Software on the host communicates with a USB device through a set of communication flows.

Figure 40-3. Example of USB 2.0 Full-speed Communication Control



The Control Transfer endpoint EP0 is always used when a USB device is first configured (USB 2.0 specifications).

40.6.1.1 USB 2.0 Full-speed Transfer Types

A communication flow is carried over one of four transfer types defined by the USB device.

Table 40-4. USB Communication Flow

Transfer	Direction	Bandwidth	Supported Endpoint Size	Error Detection	Retrying
Control	Bidirectional	Not guaranteed	8, 16, 32, 64	Yes	Automatic
Isochronous	Unidirectional	Guaranteed	512	Yes	No
Interrupt	Unidirectional	Not guaranteed	≤ 64	Yes	Yes
Bulk	Unidirectional	Not guaranteed	8, 16, 32, 64	Yes	Yes

40.6.1.2 USB Bus Transactions

Each transfer results in one or more transactions over the USB bus. There are three kinds of transactions flowing across the bus in packets:

- Setup Transaction
- Data IN Transaction
- Data OUT Transaction

40.6.2.2 Data IN Transaction

Data IN transactions are used in control, isochronous, bulk and interrupt transfers and conduct the transfer of data from the device to the host. Data IN transactions in isochronous transfer must be done using endpoints with ping-pong attributes.

Using Endpoints Without Ping-pong Attributes

To perform a Data IN transaction using a non ping-pong endpoint:

1. The application checks if it is possible to write in the FIFO by polling TXPKTRDY in the endpoint's UDP_CSRx (TXPKTRDY must be cleared).
2. The application writes the first packet of data to be sent in the endpoint's FIFO, writing zero or more byte values in the endpoint's UDP_FDRx.
3. The application notifies the USB peripheral it has finished by setting the TXPKTRDY in the endpoint's UDP_CSRx.
4. The application is notified that the endpoint's FIFO has been released by the USB device when TXCOMP in the endpoint's UDP_CSRx has been set. Then an interrupt for the corresponding endpoint is pending while TXCOMP is set.
5. The microcontroller writes the second packet of data to be sent in the endpoint's FIFO, writing zero or more byte values in the endpoint's UDP_FDRx.
6. The microcontroller notifies the USB peripheral it has finished by setting the TXPKTRDY in the endpoint's UDP_CSRx.
7. The application clears the TXCOMP in the endpoint's UDP_CSRx.

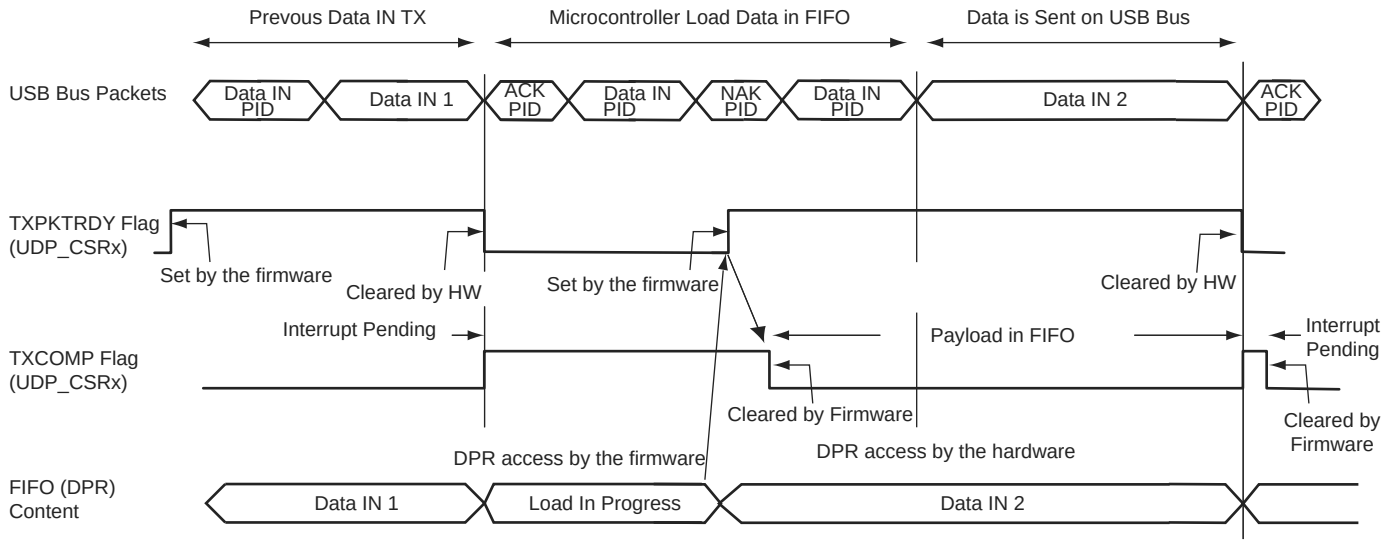
After the last packet has been sent, the application must clear TXCOMP once this has been set.

TXCOMP is set by the USB device when it has received an ACK PID signal for the Data IN packet. An interrupt is pending while TXCOMP is set.

Warning: TX_COMP must be cleared after TX_PKTRDY has been set.

Note: Refer to Chapter 8 of the *Universal Serial Bus Specification, Rev 2.0,* for more information on the Data IN protocol layer.

Figure 40-6. Data IN Transfer for Non Ping-pong Endpoint



40.6.2.4 Stall Handshake

A stall handshake can be used in one of two distinct occasions. (For more information on the stall handshake, refer to Chapter 8 of the *Universal Serial Bus Specification, Rev 2.0.*)

- A functional stall is used when the halt feature associated with the endpoint is set. (Refer to Chapter 9 of the *Universal Serial Bus Specification, Rev 2.0,* for more information on the halt feature.)
- To abort the current request, a protocol stall is used, but uniquely with control transfer.

The following procedure generates a stall packet:

1. The microcontroller sets the FORCESTALL flag in the UDP_CSRx endpoint's register.
2. The host receives the stall packet.
3. The microcontroller is notified that the device has sent the stall by polling the STALLSENT to be set. An endpoint interrupt is pending while STALLSENT is set. The microcontroller must clear STALLSENT to clear the interrupt.

When a setup transaction is received after a stall handshake, STALLSENT must be cleared in order to prevent interrupts due to STALLSENT being set.

Figure 40-12. Stall Handshake (Data IN Transfer)

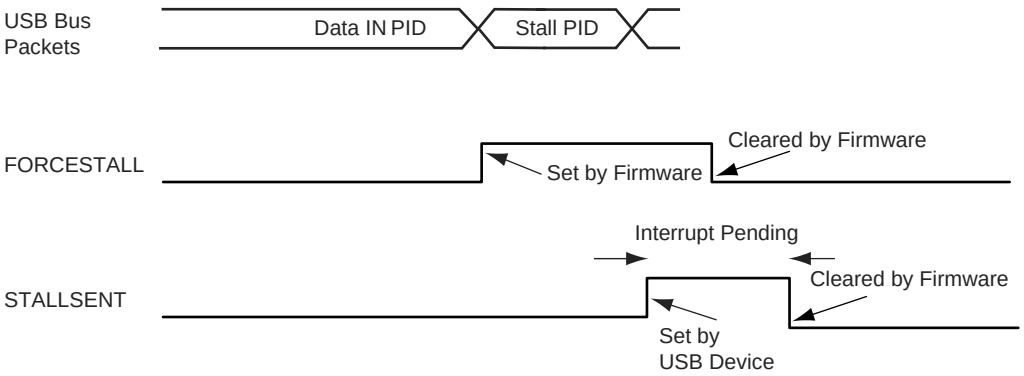
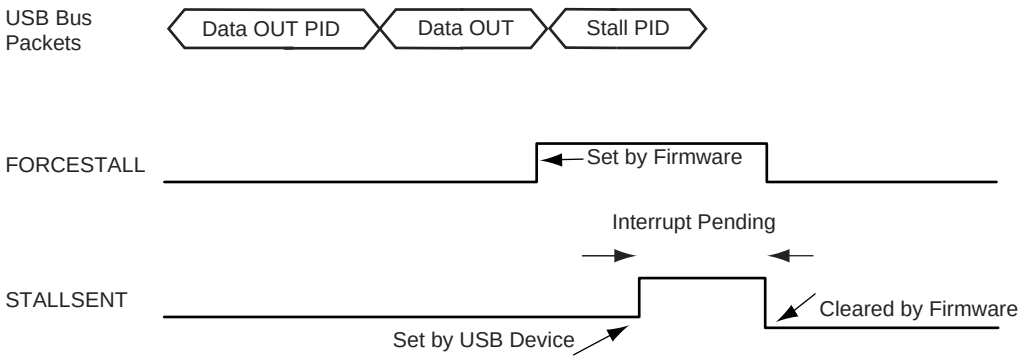


Figure 40-13. Stall Handshake (Data OUT Transfer)



40.6.2.5 Transmit Data Cancellation

Some endpoints have dual-banks whereas some endpoints have only one bank. The procedure to cancel transmission data held in these banks is described below.

To see the organization of dual-bank availability refer to Table 40-1 ”USB Endpoint Description”.

Endpoints Without Dual-Banks

The cancellation procedure depends on the TXPKTRDY flag value in the UDP_CSR:

- TXPKTRDY is not set:
  - Reset the endpoint to clear the FIFO (pointers). (See Section 40.7.9 ”UDP Reset Endpoint Register”.)
- TXPKTRDY has already been set:
  - Clear TXPKTRDY so that no packet is ready to be sent
  - Reset the endpoint to clear the FIFO (pointers). (See Section 40.7.9 ”UDP Reset Endpoint Register”.)

Endpoints With Dual-Banks

The cancellation procedure depends on the TXPKTRDY flag value in the UDP_CSR:

- TXPKTRDY is not set:
  - Reset the endpoint to clear the FIFO (pointers). (See Section 40.7.9 ”UDP Reset Endpoint Register”.)
- TXPKTRDY has already been set:
  - Clear TXPKTRDY and read it back until actually read at 0.
  - Set TXPKTRDY and read it back until actually read at 1.
  - Clear TXPKTRDY so that no packet is ready to be sent.
  - Reset the endpoint to clear the FIFO (pointers). (See Section 40.7.9 ”UDP Reset Endpoint Register”.)

42.7.10 ADC Interrupt Disable Register

Name: ADC_IDR

Address: 0x40038028

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	RXBUFF	ENDRX	COMPE	GOVRE	DRDY
23	22	21	20	19	18	17	16
EOCAL	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
EOC15	EOC14	EOC13	EOC12	EOC11	EOC10	EOC9	EOC8
7	6	5	4	3	2	1	0
EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0

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

0: No effect.

1: Disables the corresponding interrupt.

- **EOCx: End of Conversion Interrupt Disable x**
- **EOCAL: End of Calibration Sequence**
- **DRDY: Data Ready Interrupt Disable**
- **GOVRE: General Overrun Error Interrupt Disable**
- **COMPE: Comparison Event Interrupt Disable**
- **ENDRX: End of Receive Buffer Interrupt Disable**
- **RXBUFF: Receive Buffer Full Interrupt Disable**

42.7.17 ADC Channel Offset Register

Name: ADC_COR

Address: 0x4003804C

Access: Read/Write

31	30	29	28	27	26	25	24
DIFF15	DIFF14	DIFF13	DIFF12	DIFF11	DIFF10	DIFF9	DIFF8
23	22	21	20	19	18	17	16
DIFF7	DIFF6	DIFF5	DIFF4	DIFF3	DIFF2	DIFF1	DIFF0
15	14	13	12	11	10	9	8
OFF15	OFF14	OFF13	OFF12	OFF11	OFF10	OFF9	OFF8
7	6	5	4	3	2	1	0
OFF7	OFF6	OFF5	OFF4	OFF3	OFF2	OFF1	OFF0

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

- **OFFx: Offset for Channel x**

0: No offset.

1: Centers the analog signal on $V_{ADVREF}/2$ before the gain scaling. The offset applied is

$(G-1)V_{ADVREF}/2$

where G is the gain applied (see Section 42.7.16 “ADC Channel Gain Register”).

- **DIFFx: Differential Inputs for Channel x**

0: Single-ended mode.

1: Fully differential mode.

Table 47-1. Ordering Codes for SAM4S Devices (Continued)

Ordering Code	MRL	Flash (Kbytes)	SRAM (Kbytes)	Package	Carrier Type	Operating Temperature Range
ATSAM4S8CA-CFU	A	512	128	VFBGA100	Tray	Industrial (-40°C to +85°C)
ATSAM4S8CB-CFU	B					
ATSAM4S8CA-AU	A	512	128	LQFP100	Tray	Industrial (-40°C to +85°C)
ATSAM4S8CB-AU	B					
ATSAM4S8CA-CFN	A	512	128	VFBGA100	Tray	Industrial (-40°C to +105°C)
ATSAM4S8CB-CFN	B					
ATSAM4S8CA-AN	A	512	128	LQFP100	Tray	Industrial (-40°C to +105°C)
ATSAM4S8CB-AN	B					
ATSAM4S8BA-MU	A	512	128	QFN64	Tray	Industrial (-40°C to +85°C)
ATSAM4S8BB-MU	B					
ATSAM4S8BA-AU	A	512	128	LQFP64	Tray	Industrial (-40°C to +85°C)
ATSAM4S8BB-AU	B					
ATSAM4S8BA-UUR	A	512	128	WLCSP64	Reel	Industrial (-40°C to +85°C)
ATSAM4S8BB-UUR	B					
ATSAM4S8BA-AN	A	512	128	LQFP64	Tray	Industrial (-40°C to +105°C)
ATSAM4S8BB-AN	B					
ATSAM4S4CA-CU	A	256	64	TFBGA100	Tray	Industrial (-40°C to +85°C)
ATSAM4S4CB-CU	B					
ATSAM4S4CA-CFU	A	256	64	VFBGA100	Tray	Industrial (-40°C to +85°C)
ATSAM4S4CB-CFU	B					
ATSAM4S4CA-AU	A	256	64	LQFP100	Tray	Industrial (-40°C to +85°C)
ATSAM4S4CB-AU	B					
ATSAM4S4CA-AN	A	256	64	LQFP100	Tray	Industrial (-40°C to +105°C)
ATSAM4S4CB-AN	B					
ATSAM4S4BA-MU	A	256	64	QFN64	Tray	Industrial (-40°C to +85°C)
ATSAM4S4BB-MU	B					
ATSAM4S4BA-AU	A	256	64	LQFP64	Tray	Industrial (-40°C to +85°C)
ATSAM4S4BB-AU	B					
ATSAM4S4BA-UUR	A	256	64	WLCSP64	Reel	Industrial (-40°C to +85°C)
ATSAM4S4BB-UUR	B					
ATSAM4S4BA-AN	A	256	64	LQFP64	Tray	Industrial (-40°C to +105°C)
ATSAM4S4BB-AN	B					
ATSAM4S4AA-MU	A	256	64	QFN48	Tray	Industrial (-40°C to +85°C)
ATSAM4S4AB-MU	B					
ATSAM4S4AA-AU	A	256	64	LQFP48	Tray	Industrial (-40°C to +85°C)
ATSAM4S4AB-AU	B					