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	I²C, IrDA, Memory Card, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	47
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 11x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/atsam4sd16ba-mu

4. Refer to “Fault Mode” in “Analog Comparator Controller (ACC)” .
5. Refer to “Fault Output” in Section 42. “Analog-to-Digital Converter (ADC)”.
6. Refer to “Fault Mode” in Section 37. “Timer Counter (TC)”
7. Refer to “Parallel Capture Mode” in “Parallel Input/Output Controller (PIO)” .
8. Refer to “Conversion Triggers” and the ADC Mode Register (ADC_MR) in Section 42., “Analog-to-Digital Converter (ADC)”.
9. Refer to PWM Comparison Value Register (PWM_CMPV) in Section 39. “Pulse Width Modulation Controller (PWM)”.
10. Refer to “PWM Comparison Units” and “PWM Event Lines” in Section 39. “Pulse Width Modulation Controller (PWM)”.
11. Refer to Section 39.6.2.2 “Comparator” in Section 39. “Pulse Width Modulation Controller (PWM)”.
12. Refer to Section 37. “Timer Counter (TC)”.
13. Refer to DACC Trigger Register (DACC_TRIGR) in Section 43. “Digital-to-Analog Converter Controller (DACC)”.

12. ARM Cortex-M4 Processor

12.1 Description

The Cortex-M4 processor is a high performance 32-bit processor designed for the microcontroller market. It offers significant benefits to developers, including outstanding processing performance combined with fast interrupt handling, enhanced system debug with extensive breakpoint and trace capabilities, efficient processor core, system and memories, ultra-low power consumption with integrated sleep modes, and platform security robustness, with integrated memory protection unit (MPU).

The Cortex-M4 processor is built on a high-performance processor core, with a 3-stage pipeline Harvard architecture, making it ideal for demanding embedded applications. The processor delivers exceptional power efficiency through an efficient instruction set and extensively optimized design, providing high-end processing hardware including a range of single-cycle and SIMD multiplication and multiply-with-accumulate capabilities, saturating arithmetic and dedicated hardware division.

To facilitate the design of cost-sensitive devices, the Cortex-M4 processor implements tightly-coupled system components that reduce processor area while significantly improving interrupt handling and system debug capabilities. The Cortex-M4 processor implements a version of the Thumb® instruction set based on Thumb-2 technology, ensuring high code density and reduced program memory requirements. The Cortex-M4 instruction set provides the exceptional performance expected of a modern 32-bit architecture, with the high code density of 8-bit and 16-bit microcontrollers.

The Cortex-M4 processor closely integrates a configurable NVIC, to deliver industry-leading interrupt performance. The NVIC includes a non-maskable interrupt (NMI), and provides up to 256 interrupt priority levels. The tight integration of the processor core and NVIC provides fast execution of interrupt service routines (ISRs), dramatically reducing the interrupt latency. This is achieved through the hardware stacking of registers, and the ability to suspend load-multiple and store-multiple operations. Interrupt handlers do not require wrapping in assembler code, removing any code overhead from the ISRs. A tail-chain optimization also significantly reduces the overhead when switching from one ISR to another.

To optimize low-power designs, the NVIC integrates with the sleep modes, that include a deep sleep function that enables the entire device to be rapidly powered down while still retaining program state.

12.1.1 System Level Interface

The Cortex-M4 processor provides multiple interfaces using AMBA® technology to provide high speed, low latency memory accesses. It supports unaligned data accesses and implements atomic bit manipulation that enables faster peripheral controls, system spinlocks and thread-safe Boolean data handling.

The Cortex-M4 processor has a Memory Protection Unit (MPU) that provides fine grain memory control, enabling applications to utilize multiple privilege levels, separating and protecting code, data and stack on a task-by-task basis. Such requirements are becoming critical in many embedded applications such as automotive.

12.1.2 Integrated Configurable Debug

The Cortex-M4 processor implements a complete hardware debug solution. This provides high system visibility of the processor and memory through either a traditional JTAG port or a 2-pin Serial Wire Debug (SWD) port that is ideal for microcontrollers and other small package devices.

For system trace the processor integrates an Instrumentation Trace Macrocell (ITM) alongside data watchpoints and a profiling unit. To enable simple and cost-effective profiling of the system events these generate, a Serial Wire Viewer (SWV) can export a stream of software-generated messages, data trace, and profiling information through a single pin.

The Flash Patch and Breakpoint Unit (FPB) provides up to eight hardware breakpoint comparators that debuggers can use. The comparators in the FPB also provide remap functions of up to eight words in the program code in the

Examples

```
LDR     R8, [R10]              ; Loads R8 from the address in R10.
LDRNE   R2, [R5, #960]!        ; Loads (conditionally) R2 from a word
                               ; 960 bytes above the address in R5, and
                               ; increments R5 by 960.
STR     R2, [R9,#const-struc]  ; const-struc is an expression evaluating
                               ; to a constant in the range 0-4095.
STRH    R3, [R4], #4           ; Store R3 as halfword data into address in
                               ; R4, then increment R4 by 4
LDRD    R8, R9, [R3, #0x20]    ; Load R8 from a word 32 bytes above the
                               ; address in R3, and load R9 from a word 36
                               ; bytes above the address in R3
STRD    R0, R1, [R8], #-16     ; Store R0 to address in R8, and store R1 to
                               ; a word 4 bytes above the address in R8,
                               ; and then decrement R8 by 16.
```

12.6.4.3 LDR and STR, Register Offset

Load and Store with register offset.

Syntax

op{*type*}{*cond*} *Rt*, [*Rn*, *Rm* {, LSL #*n*}]

where:

op is one of:

- LDR Load Register.
- STR Store Register.

type is one of:

- B unsigned byte, zero extend to 32 bits on loads.
- SB signed byte, sign extend to 32 bits (LDR only).
- H unsigned halfword, zero extend to 32 bits on loads.
- SH signed halfword, sign extend to 32 bits (LDR only).
- \- omit, for word.

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

Rt is the register to load or store.

Rn is the register on which the memory address is based.

Rm is a register containing a value to be used as the offset.

LSL *#n* is an optional shift, with *n* in the range 0 to 3.

Operation

LDR instructions load a register with a value from memory.

STR instructions store a register value into memory.

The memory address to load from or store to is at an offset from the register *Rn*. The offset is specified by the register *Rm* and can be shifted left by up to 3 bits using LSL.

The value to load or store can be a byte, halfword, or word. For load instructions, bytes and halfwords can either be signed or unsigned. See “Address Alignment” .

Restrictions

In these instructions:

- *Rn* must not be PC

12.6.11.9 SEV

Send Event.

Syntax

`SEV{cond}`

where:

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

Operation

SEV is a hint instruction that causes an event to be signaled to all processors within a multiprocessor system. It also sets the local event register to 1, see “Power Management” .

Condition Flags

This instruction does not change the flags.

Examples

```
SEV ; Send Event
```

12.6.11.10 SVC

Supervisor Call.

Syntax

`SVC{cond} #imm`

where:

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

imm is an expression evaluating to an integer in the range 0-255 (8-bit value).

Operation

The SVC instruction causes the SVC exception.

imm is ignored by the processor. If required, it can be retrieved by the exception handler to determine what service is being requested.

Condition Flags

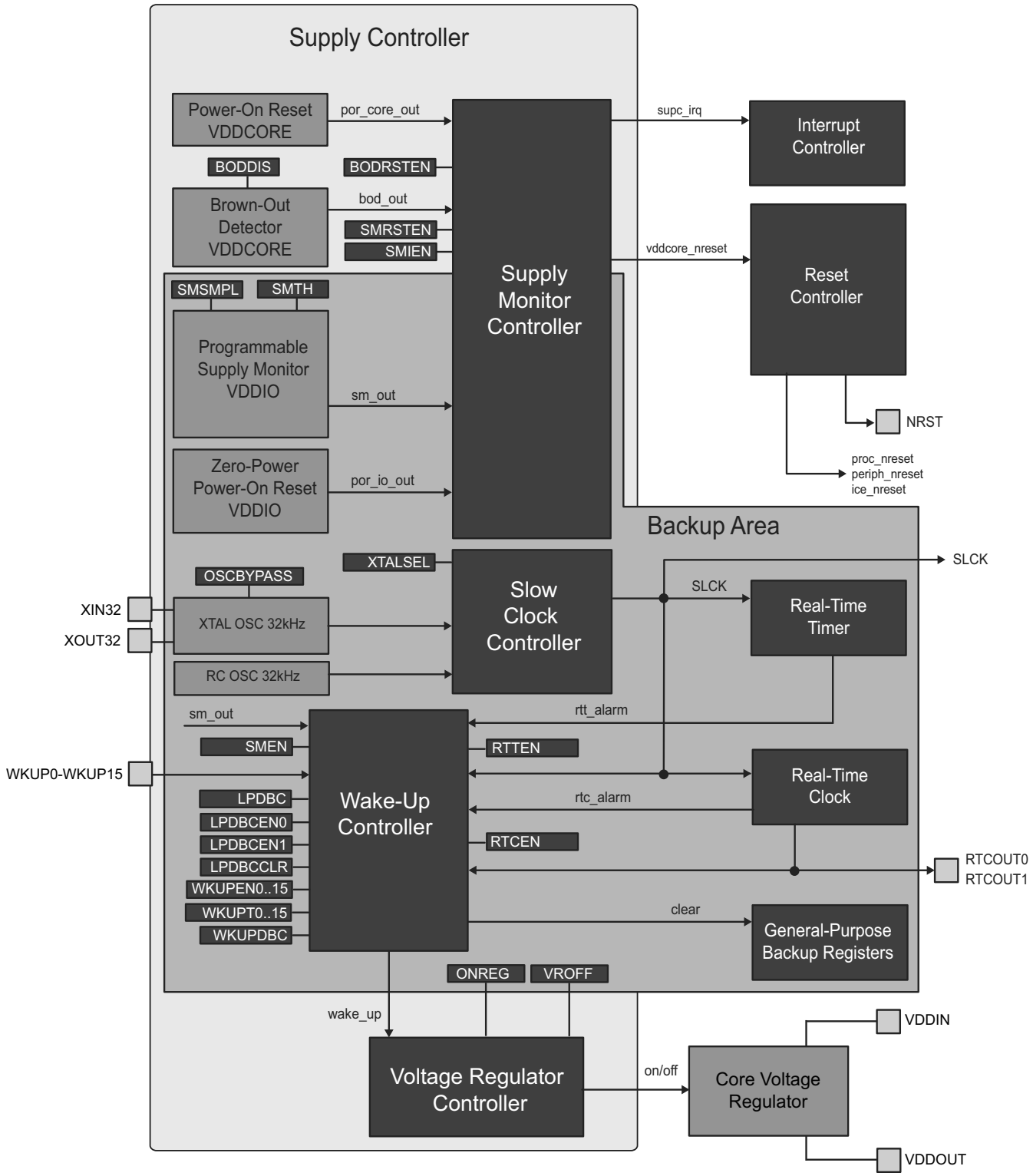
This instruction does not change the flags.

Examples

```
SVC  0x32  ; Supervisor Call (SVC handler can extract the immediate value
           ; by locating it via the stacked PC)
```

18.3 Block Diagram

Figure 18-1. Supply Controller Block Diagram



20.5 Enhanced Embedded Flash Controller (EEFC) User Interface

The User Interface of the Embedded Flash Controller (EEFC) is integrated within the System Controller with base address 0x400E0A00.

Table 20-6. Register Mapping

Offset	Register	Name	Access	Reset State
0x00	EEFC Flash Mode Register	EEFC_FMR	Read/Write	0x0400_0000
0x04	EEFC Flash Command Register	EEFC_FCR	Write-only	–
0x08	EEFC Flash Status Register	EEFC_FSR	Read-only	0x0000_0001
0x0C	EEFC Flash Result Register	EEFC_FRR	Read-only	0x0
0x10–0x14	Reserved	–	–	–
0x18–0xE4	Reserved	–	–	–

Table 21-3. Command Bit Coding (Continued)

DATA[15:0]	Symbol	Command Executed
0x0035	GSE	Get Security Bit
0x001F	WRAM	Write Memory
0x001E	GVE	Get Version

21.3.3 Entering Programming Mode

The following algorithm puts the device in Parallel Programming mode:

1. Apply the supplies as described in Table 21-1.
2. Apply XIN clock within t_{POR_RESET} if an external clock is available.
3. Wait for t_{POR_RESET}
4. Start a read or write handshaking.

Note: After reset, the device is clocked by the internal RC oscillator. Before clearing RDY signal, if an external clock (> 32 kHz) is connected to XIN, then the device switches on the external clock. Else, XIN input is not considered. A higher frequency on XIN speeds up the programmer handshake.

21.3.4 Programmer Handshaking

An handshake is defined for read and write operations. When the device is ready to start a new operation (RDY signal set), the programmer starts the handshake by clearing the NCMD signal. The handshaking is achieved once NCMD signal is high and RDY is high.

21.3.4.1 Write Handshaking

For details on the write handshaking sequence, refer to Figure 21-2 and Table 21-4.

Figure 21-2. Parallel Programming Timing, Write Sequence

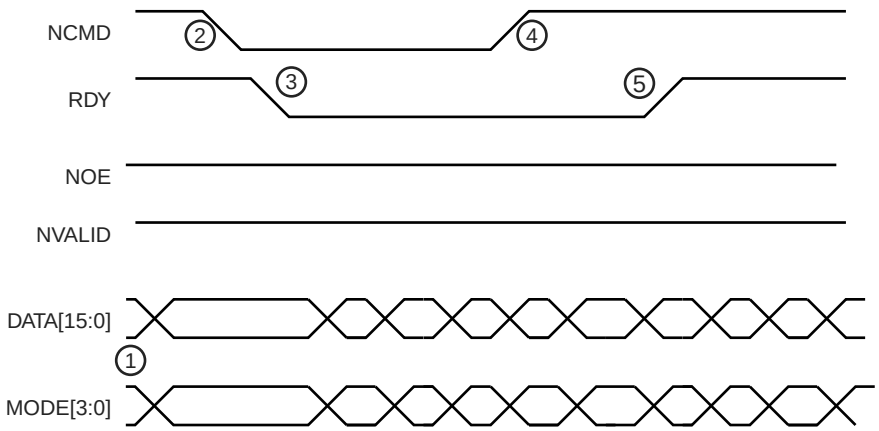


Table 21-4. Write Handshake

Step	Programmer Action	Device Action	Data I/O
1	Sets MODE and DATA signals	Waits for NCMD low	Input
2	Clears NCMD signal	Latches MODE and DATA	Input
3	Waits for RDY low	Clears RDY signal	Input

27.6.9 Transfer Control Register

Name: PERIPH_PTCR

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
–	–	–	–	–	–	TXTDIS	TXTEN
7	6	5	4	3	2	1	0
–	–	–	–	–	–	RXTDIS	RXTEN

- **RXTEN: Receiver Transfer Enable**

0: No effect.

1: Enables PDC receiver channel requests if RXTDIS is not set.

When a half-duplex peripheral is connected to the PDC, enabling the receiver channel requests automatically disables the transmitter channel requests. It is forbidden to set both TXTEN and RXTEN for a half-duplex peripheral.

- **RXTDIS: Receiver Transfer Disable**

0: No effect.

1: Disables the PDC receiver channel requests.

When a half-duplex peripheral is connected to the PDC, disabling the receiver channel requests also disables the transmitter channel requests.

- **TXTEN: Transmitter Transfer Enable**

0: No effect.

1: Enables the PDC transmitter channel requests.

When a half-duplex peripheral is connected to the PDC, it enables the transmitter channel requests only if RXTEN is not set. It is forbidden to set both TXTEN and RXTEN for a half-duplex peripheral.

- **TXTDIS: Transmitter Transfer Disable**

0: No effect.

1: Disables the PDC transmitter channel requests.

When a half-duplex peripheral is connected to the PDC, disabling the transmitter channel requests disables the receiver channel requests.

29.17 Power Management Controller (PMC) User Interface

Table 29-3. Register Mapping

Offset	Register	Name	Access	Reset
0x0000	System Clock Enable Register	PMC_SCER	Write-only	–
0x0004	System Clock Disable Register	PMC_SCDR	Write-only	–
0x0008	System Clock Status Register	PMC_SCSR	Read-only	0x0000_0001
0x000C	Reserved	–	–	–
0x0010	Peripheral Clock Enable Register 0	PMC_PCER0	Write-only	–
0x0014	Peripheral Clock Disable Register 0	PMC_PCDR0	Write-only	–
0x0018	Peripheral Clock Status Register 0	PMC_PCSR0	Read-only	0x0000_0000
0x001C	Reserved	–	–	–
0x0020	Main Oscillator Register	CKGR_MOR	Read/Write	0x0000_0008
0x0024	Main Clock Frequency Register	CKGR_MCFR	Read/Write	0x0000_0000
0x0028	PLLA Register	CKGR_PLLAR	Read/Write	0x0000_3F00
0x002C	PLLB Register	CKGR_PLLBR	Read/Write	0x0000_3F00
0x0030	Master Clock Register	PMC_MCKR	Read/Write	0x0000_0001
0x0034	Reserved	–	–	–
0x0038	USB Clock Register	PMC_USB	Read/Write	0x0000_0000
0x003C	Reserved	–	–	–
0x0040	Programmable Clock 0 Register	PMC_PCK0	Read/Write	0x0000_0000
0x0044	Programmable Clock 1 Register	PMC_PCK1	Read/Write	0x0000_0000
0x0048	Programmable Clock 2 Register	PMC_PCK2	Read/Write	0x0000_0000
0x004C– 0x005C	Reserved	–	–	–
0x0060	Interrupt Enable Register	PMC_IER	Write-only	–
0x0064	Interrupt Disable Register	PMC_IDR	Write-only	–
0x0068	Status Register	PMC_SR	Read-only	0x0003_0008
0x006C	Interrupt Mask Register	PMC_IMR	Read-only	0x0000_0000
0x0070	Fast Startup Mode Register	PMC_FSMR	Read/Write	0x0000_0000
0x0074	Fast Startup Polarity Register	PMC_FSPR	Read/Write	0x0000_0000
0x0078	Fault Output Clear Register	PMC_FOCR	Write-only	–
0x007C–0x00E0	Reserved	–	–	–
0x00E4	Write Protection Mode Register	PMC_WPMR	Read/Write	0x0000_0000
0x00E8	Write Protection Status Register	PMC_WPSR	Read-only	0x0000_0000
0x00EC–0x00FC	Reserved	–	–	–
0x0100	Peripheral Clock Enable Register 1	PMC_PCER1	Write-only	–
0x0104	Peripheral Clock Disable Register 1	PMC_PCDR1	Write-only	–
0x0108	Peripheral Clock Status Register 1	PMC_PCSR1	Read-only	0x0000_0000
0x010C	Reserved	–	–	–

29.17.1 PMC System Clock Enable Register

Name: PMC_SCER

Address: 0x400E0400

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
–	–	–	–	–	PCK2	PCK1	PCK0
7	6	5	4	3	2	1	0
UDP	–	–	–	–	–	–	–

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

- **UDP: USB Device Port Clock Enable**

0: No effect.

1: Enables the 48 MHz clock (UDPCK) of the USB Device Port.

- **PCKx: Programmable Clock x Output Enable**

0: No effect.

1: Enables the corresponding Programmable Clock output.

31.5.4 Output Control

When the I/O line is assigned to a peripheral function, i.e., the corresponding bit in PIO_PSR is at zero, the drive of the I/O line is controlled by the peripheral. Peripheral A or B or C or D depending on the value in PIO_ABCDSR1 and PIO_ABCDSR2 determines whether the pin is driven or not.

When the I/O line is controlled by the PIO Controller, the pin can be configured to be driven. This is done by writing the Output Enable Register (PIO_OER) and Output Disable Register (PIO_ODR). The results of these write operations are detected in the Output Status Register (PIO_OSR). When a bit in this register is at zero, the corresponding I/O line is used as an input only. When the bit is at one, the corresponding I/O line is driven by the PIO Controller.

The level driven on an I/O line can be determined by writing in the Set Output Data Register (PIO_SODR) and the Clear Output Data Register (PIO_CODR). These write operations, respectively, set and clear the Output Data Status Register (PIO_ODSR), which represents the data driven on the I/O lines. Writing in PIO_OER and PIO_ODR manages PIO_OSR whether the pin is configured to be controlled by the PIO Controller or assigned to a peripheral function. This enables configuration of the I/O line prior to setting it to be managed by the PIO Controller.

Similarly, writing in PIO_SODR and PIO_CODR affects PIO_ODSR. This is important as it defines the first level driven on the I/O line.

31.5.5 Synchronous Data Output

Clearing one or more PIO line(s) and setting another one or more PIO line(s) synchronously cannot be done by using PIO_SODR and PIO_CODR. It requires two successive write operations into two different registers. To overcome this, the PIO Controller offers a direct control of PIO outputs by single write access to PIO_ODSR. Only bits unmasked by the Output Write Status Register (PIO_OWSR) are written. The mask bits in PIO_OWSR are set by writing to the Output Write Enable Register (PIO_OWER) and cleared by writing to the Output Write Disable Register (PIO_OWDR).

After reset, the synchronous data output is disabled on all the I/O lines as PIO_OWSR resets at 0x0.

31.5.6 Multi-Drive Control (Open Drain)

Each I/O can be independently programmed in open drain by using the multi-drive feature. This feature permits several drivers to be connected on the I/O line which is driven low only by each device. An external pull-up resistor (or enabling of the internal one) is generally required to guarantee a high level on the line.

The multi-drive feature is controlled by the Multi-driver Enable Register (PIO_MDER) and the Multi-driver Disable Register (PIO_MDDR). The multi-drive can be selected whether the I/O line is controlled by the PIO Controller or assigned to a peripheral function. The Multi-driver Status Register (PIO_MDSR) indicates the pins that are configured to support external drivers.

After reset, the multi-drive feature is disabled on all pins, i.e., PIO_MDSR resets at value 0x0.

31.5.7 Output Line Timings

Figure 31-3 shows how the outputs are driven either by writing PIO_SODR or PIO_CODR, or by directly writing PIO_ODSR. This last case is valid only if the corresponding bit in PIO_OWSR is set. Figure 31-3 also shows when the feedback in the Pin Data Status Register (PIO_PDSR) is available.

37.7.15 TC QDEC Interrupt Enable Register

Name: TC_QIER

Address: 0x400100C8 (0), 0x400140C8 (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
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	QERR	DIRCHG	IDX

- **IDX: Index**

0: No effect.

1: Enables the interrupt when a rising edge occurs on IDX input.

- **DIRCHG: Direction Change**

0: No effect.

1: Enables the interrupt when a change on rotation direction is detected.

- **QERR: Quadrature Error**

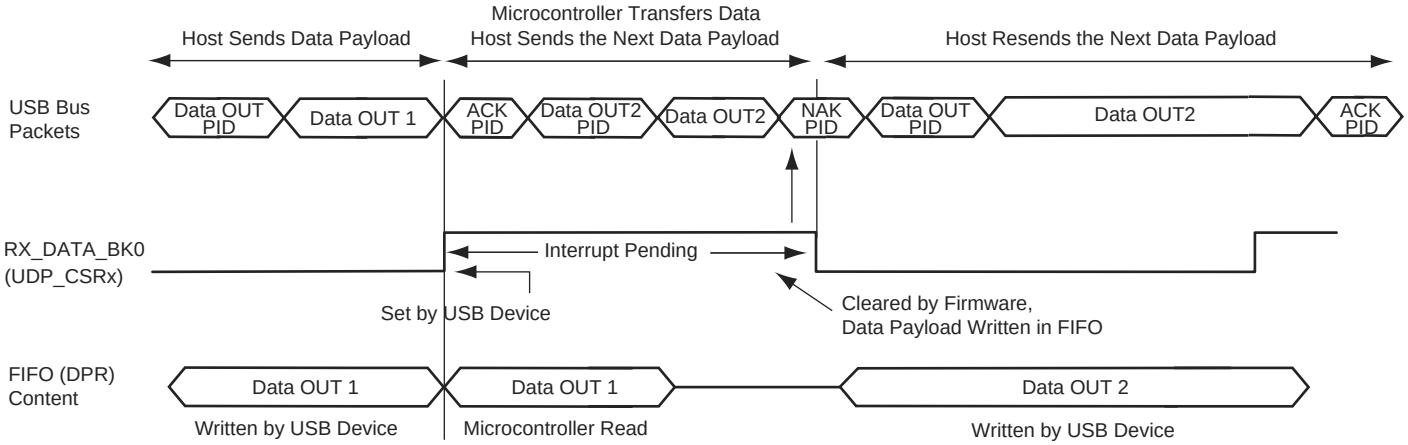
0: No effect.

1: Enables the interrupt when a quadrature error occurs on PHA, PHB.

39.2 Embedded Characteristics

- 4 Channels
- Common Clock Generator Providing Thirteen Different Clocks
  - A Modulo n Counter Providing Eleven Clocks
  - Two Independent Linear Dividers Working on Modulo n Counter Outputs
- Independent Channels
  - Independent 16-bit Counter for Each Channel
  - Independent Complementary Outputs with 12-bit Dead-Time Generator (Also Called Dead-Band or Non-Overlapping Time) for Each Channel
  - Independent Enable Disable Command for Each Channel
  - Independent Clock Selection for Each Channel
  - Independent Period, Duty-Cycle and Dead-Time for Each Channel
  - Independent Double Buffering of Period, Duty-Cycle and Dead-Times for Each Channel
  - Independent Programmable Selection of The Output Waveform Polarity for Each Channel
  - Independent Programmable Center- or Left-aligned Output Waveform for Each Channel
  - Independent Output Override for Each Channel
  - Independent Interrupt for Each Channel, at Each Period for Left-Aligned or Center-Aligned Configuration
- 2 2-bit Gray Up/Down Channels for Stepper Motor Control
- Synchronous Channel Mode
  - Synchronous Channels Share the Same Counter
  - Mode to Update the Synchronous Channels Registers after a Programmable Number of Periods
  - Synchronous Channels Supports Connection of one Peripheral DMA Controller Channel Which Offers Buffer Transfer Without Processor Intervention To Update Duty-Cycle Registers
- 2 Independent Events Lines Intended to Synchronize ADC Conversions
  - Programmable delay for Events Lines to delay ADC measurements
- 8 Comparison Units Intended to Generate Interrupts, Pulses on Event Lines and Peripheral DMA Controller Transfer Requests
- 8 Programmable Fault Inputs Providing an Asynchronous Protection of PWM Outputs
  - 3 User Driven through PIO Inputs
  - PMC Driven when Crystal Oscillator Clock Fails
  - ADC Controller Driven through Configurable Comparison Function
  - Analog Comparator Controller Driven
  - Timer/Counter Driven through Configurable Comparison Function
- Register Write Protection

Figure 40-9. Data OUT Transfer for Non Ping-pong Endpoints

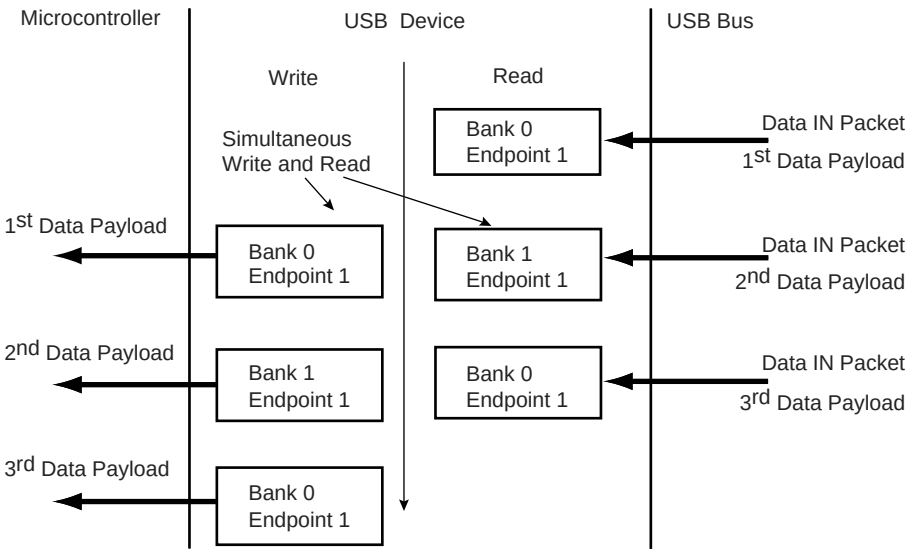


An interrupt is pending while the flag RX_DATA_BK0 is set. Memory transfer between the USB device, the FIFO and microcontroller memory is not possible after RX_DATA_BK0 has been cleared. Otherwise, the USB device would accept the next Data OUT transfer and overwrite the current Data OUT packet in the FIFO.

Using Endpoints With Ping-pong Attributes

During isochronous transfer, using an endpoint with ping-pong attributes is obligatory. To be able to guarantee a constant bandwidth, the microcontroller must read the previous data payload sent by the host, while the current data payload is received by the USB device. Thus two banks of memory are used. While one is available for the microcontroller, the other one is locked by the USB device.

Figure 40-10. Bank Swapping in Data OUT Transfers for Ping-pong Endpoints



When using a ping-pong endpoint, the following procedures are required to perform Data OUT transactions:

1. The host generates a Data OUT packet.
2. This packet is received by the USB device endpoint. It is written in the endpoint's FIFO Bank 0.
3. The USB device sends an ACK PID packet to the host. The host can immediately send a second Data OUT packet. It is accepted by the device and copied to FIFO Bank 1.
4. The microcontroller is notified that the USB device has received a data payload, polling RX_DATA_BK0 in the endpoint's UDP_CSRx. An interrupt is pending for this endpoint while RX_DATA_BK0 is set.

41.7.2 ACC Mode Register

Name: ACC_MR

Address: 0x40040004

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
–	FE	SELFS	INV	–	EDGETYP		ACEN
7	6	5	4	3	2	1	0
–	SELPLUS			–	SELMINUS		

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

- **SELMINUS: Selection for Minus Comparator Input**

0..7: Selects the input to apply on analog comparator SELMINUS comparison input.

Value	Name	Description
0	TS	Select TS
1	ADVREF	Select ADVREF
2	DAC0	Select DAC0
3	DAC1	Select DAC1
4	AD0	Select AD0
5	AD1	Select AD1
6	AD2	Select AD2
7	AD3	Select AD3

- **SELPLUS: Selection For Plus Comparator Input**

0..7: Selects the input to apply on analog comparator SELPLUS comparison input.

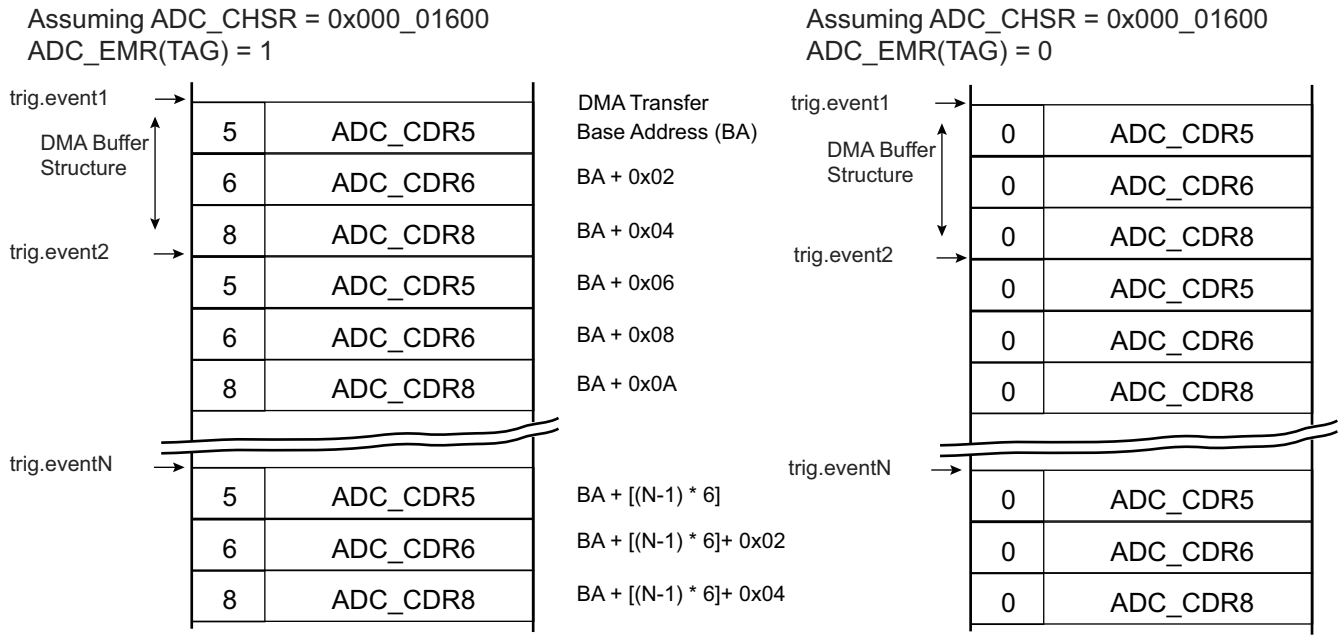
Value	Name	Description
0	AD0	Select AD0
1	AD1	Select AD1
2	AD2	Select AD2
3	AD3	Select AD3
4	AD4	Select AD4
5	AD5	Select AD5
6	AD6	Select AD6
7	AD7	Select AD7

- **ACEN: Analog Comparator Enable**

0 (DIS): Analog comparator disabled.

1 (EN): Analog comparator enabled.

Figure 42-8. Buffer Structure



42.6.14 Fault Output

The ADC Controller internal fault output is directly connected to PWM fault input. Fault output may be asserted depending on the configuration of ADC_EMR and ADC_CWR and converted values. When the compare occurs, the ADC fault output generates a pulse of one peripheral clock cycle to the PWM fault input. This fault line can be enabled or disabled within PWM. Should it be activated and asserted by the ADC Controller, the PWM outputs are immediately placed in a safe state (pure combinational path). Note that the ADC fault output connected to the PWM is not the COMPE bit. Thus the Fault mode (FMOD) within the PWM configuration must be FMOD = 1.

42.6.15 Register Write Protection

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

If a write access to the protected registers is detected, the WPVS flag in the “ADC Write Protection Status Register” (ADC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is automatically reset by reading the ADC_WPSR.

The following registers can be write-protected:

- ADC Mode Register
- ADC Channel Sequence 1 Register
- ADC Channel Sequence 2 Register
- ADC Channel Enable Register
- ADC Channel Disable Register
- ADC Extended Mode Register
- ADC Compare Window Register
- ADC Channel Gain Register
- ADC Channel Offset Register
- ADC Analog Control Register

43.7.10 DACC Interrupt Status Register

Name: DACC_ISR

Address: 0x4003C030

Access: Read-only

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

- **TXRDY: Transmit Ready Interrupt Flag**

0: DACC is not ready to accept new conversion requests.

1: DACC is ready to accept new conversion requests.

- **EOC: End of Conversion Interrupt Flag**

0: No conversion has been performed since the last DACC_ISR read.

1: At least one conversion has been performed since the last DACC_ISR read.

- **ENDTX: End of DMA Interrupt Flag**

0: The Transmit Counter register has not reached 0 since the last write in DACC_TCR or DACC_TNCR.

1: The Transmit Counter register has reached 0 since the last write in DACC _TCR or DACC_TNCR.

- **TXBUFE: Transmit Buffer Empty**

0: The Transmit Counter register has not reached 0 since the last write in DACC_TCR or DACC_TNCR.

1: The Transmit Counter register has reached 0 since the last write in DACC _TCR or DACC_TNCR.

44.6 PLLA, PLLB Characteristics

Table 44-34. Supply Voltage Phase Lock Loop Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDPLLR}	Supply Voltage Range		1.08	1.2	1.32	V

Table 44-35. PLLA and PLLB Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{IN}	Input Frequency		3	–	32	MHz
f_{OUT}	Output Frequency		80	–	240	MHz
I_{PLL}	Current Consumption	Active mode @ 80 MHz @ 1.2V Active mode @ 96 MHz @ 1.2V Active mode @ 160 MHz @ 1.2V Active mode @ 240 MHz @ 1.2V	–	0.94 1.2 2.1 3.34	1.2 1.5 2.5 4	mA
t_s	Settling Time		–	60	150	µs

Table 49-4. SAM4S Datasheet Rev. 11100H Revision History (Continued)

Doc. Date	Changes
08-Jan-15	Added “Symbol” column to Table 44-57 ”Static Performance Characteristics”, Table 44-58 ”Dynamic Performance Characteristics”, Table 44-59 ”Analog Outputs”, and Table 44-60 ”Analog Comparator Characteristics” Section 44.11 “Temperature Sensor”: specified instances of “27°C” as ambient temperature Table 44-63 ”I/O Characteristics”: added parameter “Maximum I/O skew” Section 44.12.3.1 “Maximum SPI Frequency”: - under “Master Write Mode”, replaced “the maximum SPI frequency is the one from the pad” with “the maximum SPI frequency is defined by the pin FreqMax value - updated content under “Master Read Mode” Table 44-65 ”SSC Timings”: in Min/Max values for SSC_4 and SSC_7, corrected links to footnote 2 Section 44.12.9 “Embedded Flash Characteristics”: in first paragraph, corrected “field FWS of the MC_FMR” to “field FWS of the EEFC_FMR”
	Section 45. “Mechanical Characteristics” Inserted heading Section 45.1 “100-lead LQFP Mechanical Characteristics” Inserted heading Section 45.2 “100-ball TFBGA Mechanical Characteristics” Inserted heading Section 45.3 “100-ball VFBGA Mechanical Characteristics” Inserted heading Section 45.4 “64-lead LQFP Mechanical Characteristics” Table 45-16 ”LQFP Package Characteristics”: corrected title (was “LQFP and QFN Package Characteristics”) Inserted heading Section 45.5 “64-lead QFN Mechanical Characteristics” Inserted heading Section 45.6 “64-ball WLCSP Mechanical Characteristics” Inserted heading Section 45.7 “48-lead LQFP Mechanical Characteristics” and added sentence “This package respects the recommendations of the NEMI User Group.” Inserted heading Section 45.8 “48-lead QFN Mechanical Characteristics” and added sentence “This package respects the recommendations of the NEMI User Group.” Table 45-29 ”48-lead QFN Package Characteristics”: corrected title (was “48-lead LQFP Package Characteristics”) and changed Moisture Sensitivity Level from 1 to 3 Table 45-30 ”48-lead QFN Package Reference”: corrected title (was “48-lead LQFP Package Reference”)
	Added Section 46. “Marking”
	Section 47. “Ordering Information”: Table 47-1 ”Ordering Codes for SAM4S Devices”: added ordering codes for MRL ‘B’
	Section 48. “Errata” Section 48.1 “Errata SAM4SD32/SD16/SA16/S16/S8 Rev. A Parts”: added Section 48.1.5 “Low-power Mode” Added Section 48.2 “Errata SAM4SD32/SD16/SA16/S16/S8 Rev. B Parts” Section 48.3 “Errata SAM4S4/S2 Rev. A Parts”: added Section 48.3.4 “Low-power Mode” Added Section 48.4 “Errata SAM4S4/S2 Rev. B Parts”

.

Table of Contents