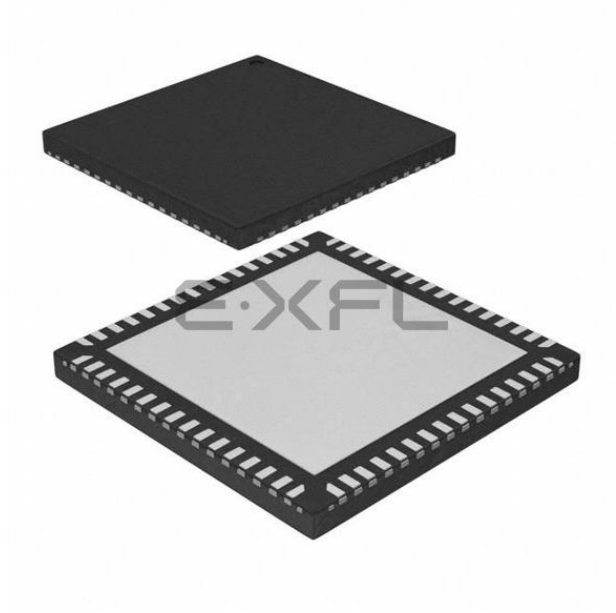


Welcome to E-XFL.COM

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®-M0+
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	I²C, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	52
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 20x12b; D/A 1x10b
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/microchip-technology/atsamd21j15b-mu

Value	Name	Description
10	MEDIUM	Sensitive Latency
11	HIGH	Critical Latency

If a master is configured with QoS level 0x00 or 0x01 there will be minimum one cycle latency for the RAM access.

The priority order for concurrent accesses are decided by two factors. First the QoS level for the master and then a static priority given by table nn-mm (table: SRAM port connection) where the lowest port ID has the highest static priority.

The MTB has fixed QoS level 3 and the DSU has fixed QoS level 1.

The CPU QoS level can be written/read at address 0x41007110, bits [1:0]. Its reset value is 0x0.

Refer to different master QOSCTRL registers for configuring QoS for the other masters (USB, DMAC).

11.5 AHB-APB Bridge

The AHB-APB bridge is an AHB slave, providing an interface between the high-speed AHB domain and the low-power APB domain. It is used to provide access to the programmable control registers of peripherals (see Product Mapping).

AHB-APB bridge is based on AMBA APB Protocol Specification V2.0 (ref. as APB4) including:

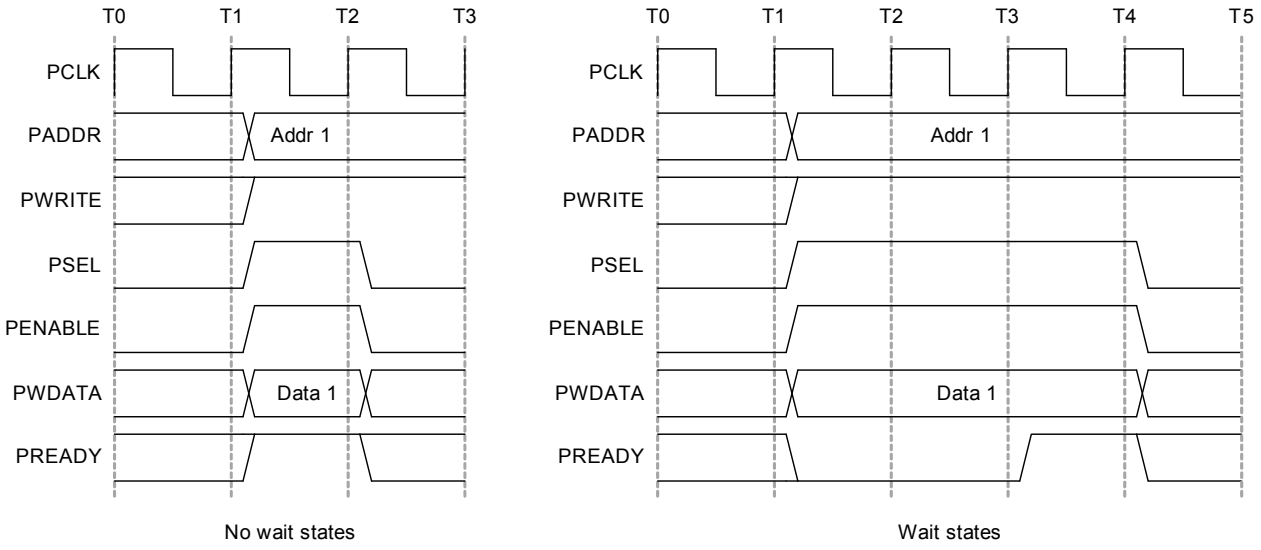
- Wait state support
- Error reporting
- Transaction protection
- Sparse data transfer (byte, half-word and word)

Additional enhancements:

- Address and data cycles merged into a single cycle
- Sparse data transfer also apply to read access

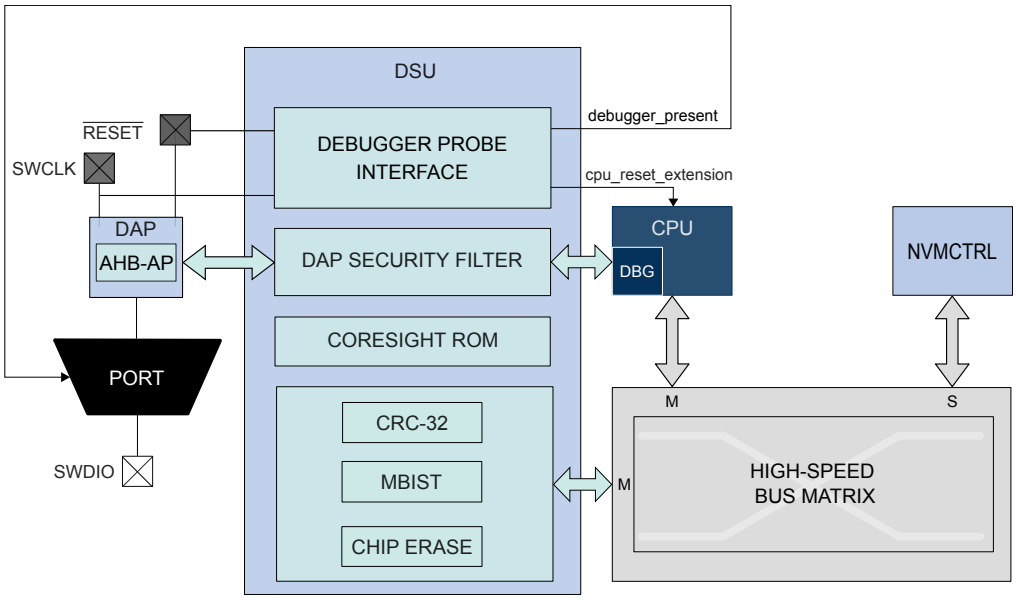
to operate the AHB-APB bridge, the clock (CLK_HPBx_AHB) must be enabled. See *PM – Power Manager* for details.

Figure 11-1. APB Write Access.



13.3 Block Diagram

Figure 13-1. DSU Block Diagram



13.4 Signal Description

The DSU uses three signals to function.

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital Input	External reset
SWCLK	Digital Input	SW clock
SWDIO	Digital I/O	SW bidirectional data pin

Related Links

I/O Multiplexing and Considerations

13.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.5.1 IO Lines

The SWCLK pin is by default assigned to the DSU module to allow debugger probe detection and to stretch the CPU reset phase. For more information, refer to Debugger Probe Detection. The Hot-Plugging feature depends on the PORT configuration. If the SWCLK pin function is changed in the PORT or if the PORT_MUX is disabled, the Hot-Plugging feature is disabled until a power-reset or an external reset.

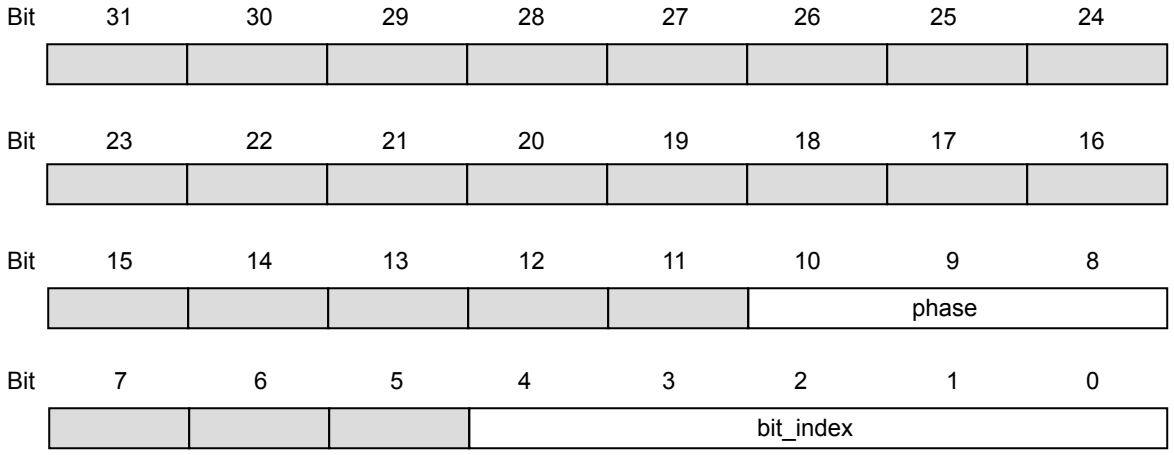
13.5.2 Power Management

The DSU will continue to operate in any sleep mode where the selected source clock is running.

Related Links

PM – Power Manager

Figure 13-6. DATA bits Description When MBIST Operation Returns an Error



- bit_index: contains the bit number of the failing bit
- phase: indicates which phase of the test failed and the cause of the error, as listed in the following table.

Table 13-4. MBIST Operation Phases

Phase	Test actions
0	Write all bits to zero. This phase cannot fail.
1	Read '0', write '1', increment address
2	Read '1', write '0'
3	Read '0', write '1', decrement address
4	Read '1', write '0', decrement address
5	Read '0', write '1'
6	Read '1', write '0', decrement address
7	Read all zeros. bit_index is not used

Table 13-5. AMOD Bit Descriptions for MBIST

AMOD[1:0]	Description
0x0	Exit on Error
0x1	Pause on Error
0x2, 0x3	Reserved

Related Links

NVMCTRL – Non-Volatile Memory Controller

Security Bit

Product Mapping

13.11.6 System Services Availability when Accessed Externally

External access: Access performed in the DSU address offset 0x200-0x1FFF range.

Internal access: Access performed in the DSU address offset 0x0-0x100 range.

Bit	7	6	5	4	3	2	1	0
	ADDR[5:0]						AMOD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:2 – ADDR[29:0]: Address
Initial word start address needed for memory operations.

Bits 1:0 – AMOD[1:0]: Access Mode
The functionality of these bits is dependent on the operation mode.

Bit description when operating CRC32: refer to 32-bit Cyclic Redundancy Check CRC32

Bit description when testing onboard memories (MBIST): refer to Testing of On-Board Memories MBIST

13.13.5 Length

Name: LENGTH
Offset: 0x0008
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	LENGTH[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LENGTH[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LENGTH[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LENGTH[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – LENGTH[29:0]: Length
Length in words needed for memory operations.

13.13.6 Data

Name: DATA
Offset: 0x000C
Reset: 0x00000000

The prescaler counter used to trigger one-shot brown-out detections also operates asynchronously from the peripheral bus. As a consequence, the prescaler registers require synchronization when written or read. The synchronization results in a delay from when the initialization of the write or read operation begins until the operation is complete.

The write-synchronization is triggered by a write to the BOD12 or BOD33 control register. The Synchronization Ready bit (PCLKSR.B12SRDY or PCLKSR.B33SRDY) in the PCLKSR register will be cleared when the write-synchronization starts and set when the write-synchronization is complete. When the write-synchronization is ongoing (PCLKSR.B33SRDY or PCLKSR.B12SRDY is zero), an attempt to do any of the following will cause the peripheral bus to stall until the synchronization is complete:

- Writing to the BOD33 or BOD12 control register
- Reading the BOD33 or BOD12 control register that was written

The user can either poll PCLKSR.B12SRDY or PCLKSR.B33SRDY or use the INTENSET.B12SRDY or INTENSET.B33SRDY interrupts to check when the synchronization is complete. It is also possible to perform the next read/write operation and wait, as this next operation will be completed after the ongoing read/write operation is synchronized.

Table 19-1. MODE0 - Mode Register Summary

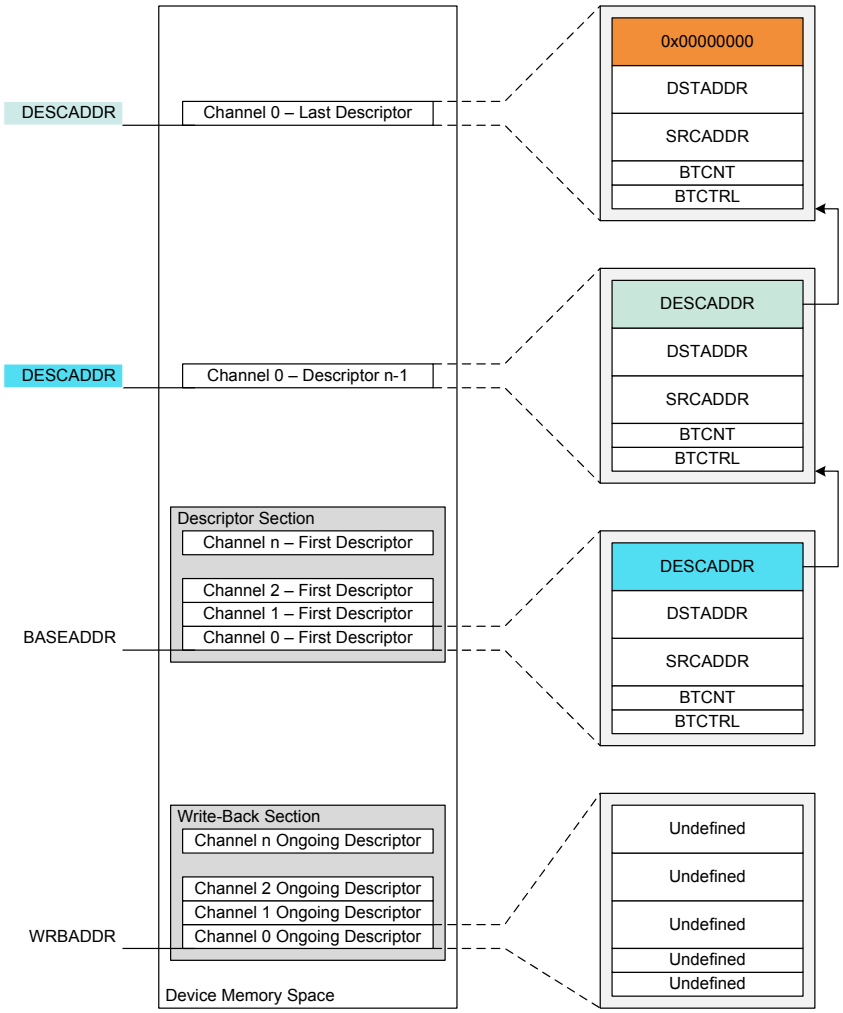
Offset	Name	Bit Pos.								
0x00	CTRL	7:0	MATCHCLR				MODE[1:0]		ENABLE	SWRST
0x01		15:8					PRESCALER[3:0]			
0x02	READREQ	7:0			ADDR[5:0]					
0x03		15:8	RREQ	RCONT						
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x05		15:8	OVFEO							CMPEO0
0x06	INTENCLR	7:0	OVF	SYNCRDY						CMP0
0x07	INTENSET	7:0	OVF	SYNCRDY						CMP0
0x08	INTFLAG	7:0	OVF	SYNCRDY						CMP0
0x09	Reserved									
0x0A	STATUS	7:0	SYNCBUSY							
0x0B	DBGCTRL	7:0								DBGRUN
0x0C	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x0D ... 0x0F	Reserved									
0x10	COUNT	7:0	COUNT[7:0]							
0x11		15:8	COUNT[15:8]							
0x12		23:16	COUNT[23:16]							
0x13		31:24	COUNT[31:24]							
0x14 ... 0x17	Reserved									
0x18	COMP0	7:0	COMP[7:0]							
0x19		15:8	COMP[15:8]							
0x1A		23:16	COMP[23:16]							
0x1B		31:24	COMP[31:24]							

Table 19-2. MODE1 - Mode Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0					MODE[1:0]		ENABLE	SWRST
0x01		15:8					PRESCALER[3:0]			
0x02	READREQ	7:0			ADDR[5:0]					
0x03		15:8	RREQ	RCONT						
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x05		15:8	OVFEO						CMPEO1	CMPEO0
0x06	INTENCLR	7:0	OVF	SYNCRDY					CMP1	CMP0
0x07	INTENSET	7:0	OVF	SYNCRDY					CMP1	CMP0
0x08	INTFLAG	7:0	OVF	SYNCRDY					CMP1	CMP0
0x09	Reserved									
0x0A	STATUS	7:0	SYNCBUSY							
0x0B	DBGCTRL	7:0								DBGRUN
0x0C	FREQCORR	7:0	SIGN	VALUE[6:0]						

ordered according to their channel number. The figure below shows an example of linked descriptors on DMA channel 0. For further details on linked descriptors, refer to Linked Descriptors.

Figure 20-3. Memory Sections



The size of the descriptor and write-back memory sections is dependent on the number of the most significant enabled DMA channel *m*, as shown below:

$$Size = 128\text{bits} \cdot (m + 1)$$

For memory optimization, it is recommended to always use the less significant DMA channels if not all channels are required.

The descriptor and write-back memory sections can either be two separate memory sections, or they can share memory section (BASEADDR=WRBADDR). The benefit of having them in two separate sections, is that the same transaction for a channel can be repeated without having to modify the first transfer descriptor. The benefit of having descriptor memory and write-back memory in the same section is that it requires less SRAM. In addition, the latency from fetching the first descriptor of a transaction to the first burst transfer is executed, is reduced.

20.6.2.4 Arbitration

If a DMA channel is enabled and not suspended when it receives a transfer trigger, it will send a transfer request to the arbiter. When the arbiter receives the transfer request it will include the DMA channel in the queue of channels having pending transfers, and the corresponding Pending Channel x bit in the Pending Channels registers (PENDCH.PENDCHx) will be set. Depending on the arbitration scheme, the arbiter

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

26.8.6 Interrupt Enable Set

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 5 – RXBRK: Receive Break Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Break Interrupt Enable bit, which enables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC: Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Clear To Send Input Change Interrupt Enable bit, which enables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 3 – RXS: Receive Start Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Start Interrupt Enable bit, which enables the Receive Start interrupt.

The missing ACK response can indicate that the I2C slave is busy with other tasks or sleeping. Therefore, it is not able to respond. In this event, the next step can be either issuing a stop condition (recommended) or resending the address packet by a repeated start condition. When using SMBus logic, the slave must ACK the address. If there is no response, it means that the slave is not available on the bus.

Case 3: Address packet transmit complete – Write packet, Master on Bus set

If the I2C master receives an acknowledge response from the I2C slave, INTFLAG.MB will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I2C operation to continue:

- Initiate a data transmit operation by writing the data byte to be transmitted into DATA.DATA.
- Transmit a new address packet by writing ADDR.ADDR. A repeated start condition will automatically be inserted before the address packet.
- Issue a stop condition, consequently terminating the transaction.

Case 4: Address packet transmit complete – Read packet, Slave on Bus set

If the I2C master receives an ACK from the I2C slave, the I2C master proceeds to receive the next byte of data from the I2C slave. When the first data byte is received, the Slave on Bus bit in the Interrupt Flag register (INTFLAG.SB) will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I2C operation to continue:

- Let the I2C master continue to read data by acknowledging the data received. ACK can be sent by software, or automatically in smart mode.
- Transmit a new address packet.
- Terminate the transaction by issuing a stop condition.

Note: An ACK or NACK will be automatically transmitted if smart mode is enabled. The Acknowledge Action bit in the Control B register (CTRLB.ACKACT) determines whether ACK or NACK should be sent.

Transmitting Data Packets

When an address packet with direction Master Write (see Figure 28-2) was transmitted successfully , INTFLAG.MB will be set. The I2C master will start transmitting data via the I2C bus by writing to DATA.DATA, and monitor continuously for packet collisions. I

If a collision is detected, the I2C master will lose arbitration and STATUS.ARBLOST will be set. If the transmit was successful, the I2C master will receive an ACK bit from the I2C slave, and STATUS.RXNACK will be cleared. INTFLAG.MB will be set in both cases, regardless of arbitration outcome.

It is recommended to read STATUS.ARBLOST and handle the arbitration lost condition in the beginning of the I2C Master on Bus interrupt. This can be done as there is no difference between handling address and data packet arbitration.

STATUS.RXNACK must be checked for each data packet transmitted before the next data packet transmission can commence. The I2C master is not allowed to continue transmitting data packets if a NACK is received from the I2C slave.

Receiving Data Packets (SCLSM=0)

When INTFLAG.SB is set, the I2C master will already have received one data packet. The I2C master must respond by sending either an ACK or NACK. Sending a NACK may be unsuccessful when

Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

28.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]					AACKEN	GCMD	SMEN
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NPWM	Single-slope PWM	PER	TOP/ ZERO	See section 'Output Polarity' below		TOP	ZERO
DSCRITICAL	Dual-slope PWM	PER	ZERO			-	ZERO
DSBOTTOM	Dual-slope PWM	PER	ZERO			-	ZERO
DSBOTH	Dual-slope PWM	PER	TOP(1) & ZERO			TOP	ZERO
DSTOP	Dual-slope PWM	PER	ZERO			TOP	–

1. The UPDATE condition on TOP only will occur when circular buffer is enabled for the channel.

Related Links

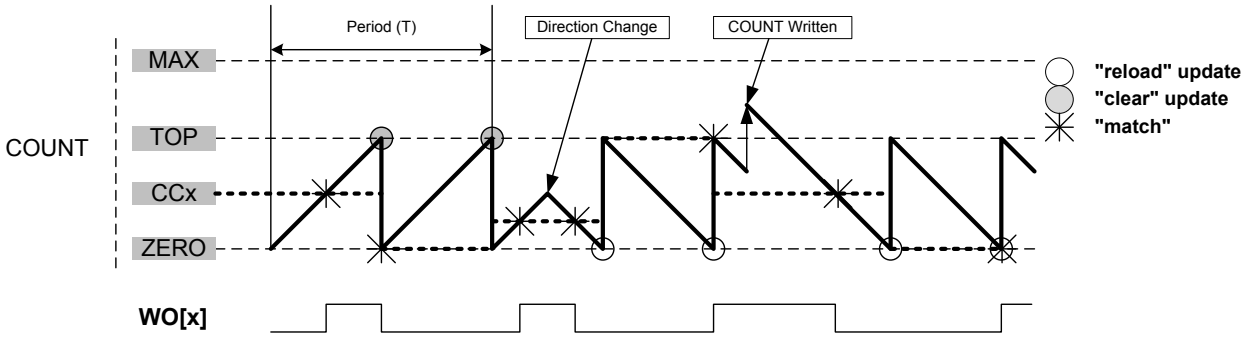
Circular Buffer

PORT: IO Pin Controller

Normal Frequency (NFRQ)

For Normal Frequency generation, the period time (T) is controlled by the period register (PER). The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (EVCTRL.MCEOx) will be set.

Figure 31-4. Normal Frequency Operation



Match Frequency (MFRQ)

For Match Frequency generation, the period time (T) is controlled by CC0 register instead of PER. WO[0] toggles on each update condition.

Table 32-4. Device Endpoint n Descriptor Bank 1

Offset 0x n0 + 0x10 + index	Name	Bit Pos.								
0x00	ADDR	7:0	ADD[7:0]							
0x01		15:8	ADD[15:8]							
0x02		23:16	ADD[23:16]							
0x03		31:24	ADD[31:24]							
0x04	PCKSIZE	7:0	BYTE_COUNT[7:0]							
0x05		15:8	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
0x06		23:16	MULTI_PACKET_SIZE[9:2]							
0x07		31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
0x08	Reserved	7:0								
0x09	Reserved	15:8								
0x0A	STATUS_BK	7:0							ERRORFLOW	CRCERR
0x0B	Reserved	7:0								
0x0C	Reserved	7:0								
0x0D	Reserved	7:0								
0x0E	Reserved	7:0								
0x0F	Reserved	7:0								

32.7.3 Host Summary

Table 32-5. General Host Registers

Offset	Name	Bit Pos.								
0x04	Reserved									
0x05	Reserved									
0x06	Reserved									
0x07	Reserved									
0x08	CTRLB	7:0		TSTK	TSTJ		SPDCONF[1:0]		RESUME	
0x09		15:8					L1RESUME	VBUSOK	BUSRESET	SOFE
0x0A	HSOFC	7:0	FLENCE				FLENC[3:0]			
0x0B	Reserved									
0x0C	STATUS	7:0	LINESTATE[1:0]				SPEED[1:0]			
0x0E	Reserved									
0x0F	Reserved									
0x10	FNUM	7:0	FNUM[4:0]							
0x11		15:8			FNUM[10:5]					
0x12	FLENHIGH	7:0	FLENHIGH[7:0]							
0x14	INTENCLR	7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x15		15:8							DDISC	DCONN
0x16	Reserved									
0x17	Reserved									
0x18	INTENSET	7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x19		15:8							DDISC	DCONN
0x1A	Reserved									

- Enable bit in control register (CTRLA.ENABLE)
- Enable bit in Comparator Control register (COMPCTRLn.ENABLE)

The following registers are synchronized when written:

- Window Control register (WINCTRL)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

Register Synchronization

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

34.8.2 Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							START1	START0
Access							R/W	R/W
Reset							0	0

Bits 1,0 – STARTx: Comparator x Start Comparison
Writing a '0' to this field has no effect.

Writing a '1' to STARTx starts a single-shot comparison on COMPx if both the Single-Shot and Enable bits in the Comparator x Control Register are '1' (COMPCTRLx.SINGLE and COMPCTRLx.ENABLE). If comparator x is not implemented, or if it is not enabled in single-shot mode, Writing a '1' has no effect.

This bit always reads as zero.

34.8.3 Event Control

Name: EVCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
							COMPEI1	COMPEI0
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
				WINEO0			COMPEO1	COMPEO0
Access				R/W			R/W	R/W
Reset				0			0	0

Bits 9,8 – COMPEIx: Comparator x Event Input
Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, the enabled action will be taken for any of the incoming events. There is no way to tell which of the incoming events caused the action.

These bits indicate whether a comparison will start or not on any incoming event.

When executing an operation that requires synchronization, the Synchronization Busy bit in the Status register (STATUS.SYNCBUSY) will be set immediately, and cleared when synchronization is complete.

If an operation that requires synchronization is executed while its busy bit is one, the operation is discarded and an error is generated.

The following bits need synchronization when written:

- Software Reset bit in the Control A register (CTRLA.SWRST)
- Enable bit in the Control A register (CTRLA.ENABLE)
- All bits in the Data register (DATA)
- All bits in the Data Buffer register (DATABUF)

Write-synchronization is denoted by the Write-Synchronized property in the register description.

The following bits need synchronization when read:

- All bits in the Data register (DATA)

35.6.8 Additional Features

35.6.8.1 DAC as an Internal Reference

The DAC output can be internally enabled as input to the analog comparator. This is enabled by writing a one to the Internal Output Enable bit in the Control B register (CTRLB.IOEN). It is possible to have the internal and external output enabled simultaneously.

The DAC output can also be enabled as input to the Analog-to-Digital Converter. In this case, the output buffer must be enabled.

35.6.8.2 Data Buffer

The Data Buffer register (DATABUF) and the Data register (DATA) are linked together to form a two-stage FIFO. The DAC uses the Start Conversion event to load data from DATABUF into DATA and start a new conversion. The Start Conversion event is enabled by writing a one to the Start Event Input bit in the Event Control register (EVCTRL.STARTEI). If a Start Conversion event occurs when DATABUF is empty, an Underrun interrupt request is generated if the Underrun interrupt is enabled.

The DAC can generate a Data Buffer Empty event when DATABUF becomes empty and new data can be loaded to the buffer. The Data Buffer Empty event is enabled by writing a one to the Empty Event Output bit in the Event Control register (EVCTRL.EMPTYEO). A Data Buffer Empty interrupt request is generated if the Data Buffer Empty interrupt is enabled.

35.6.8.3 Voltage Pump

When the DAC is used at operating voltages lower than 2.5V, the voltage pump must be enabled. This enabling is done automatically, depending on operating voltage.

The voltage pump can be disabled by writing a one to the Voltage Pump Disable bit in the Control B register (CTRLB.VPD). This can be used to reduce power consumption when the operating voltage is above 2.5V.

The voltage pump uses the asynchronous GCLK_DAC clock, and requires that the clock frequency be at least four times higher than the sampling period.

Table 37-7. Current Consumption (Device Variant A)

Mode	Conditions	T_A	Min.	Typ.	Max.	Units
ACTIVE	CPU running a While(1) algorithm	25°C	3.11	3.37	3.64	mA
		85°C	3.24	3.48	3.76	
	CPU running a While(1) algorithm V_{DDIN}=1.8V, CPU is running on Flash with 3 wait states	25°C	3.10	3.36	3.64	
		85°C	3.24	3.48	3.75	
	CPU running a While(1) algorithm, CPU is running on Flash with 3 wait states with GCLKIN as reference	25°C	60*freq + 74	60*freq + 136	62*freq + 196	µA (with freq in MHz)
		85°C	62*freq + 154	62*freq + 228	62*freq + 302	
	CPU running a Fibonacci algorithm	25°C	4.12	4.53	4.92	mA
		85°C	4.27	4.63	4.98	
	CPU running a Fibonacci algorithm V_{DDIN}=1.8V, CPU is running on flash with 3 wait states	25°C	4.12	4.53	4.92	
		85°C	4.27	4.63	4.98	
	CPU running a Fibonacci algorithm, CPU is running on Flash with 3 wait states with GCLKIN as reference	25°C	86*freq + 76	88*freq + 136	88*freq + 196	µA (with freq in MHz)
		85°C	88*freq + 156	88*freq + 230	88*freq + 302	
	CPU running a CoreMark algorithm	25°C	5.78	6.32	6.80	mA
		85°C	5.93	6.47	7.00	
	CPU running a CoreMark algorithm V_{DDIN}=1.8V, CPU is running on flash with 3 wait states	25°C	5.17	5.60	5.96	
		85°C	5.35	5.73	6.10	
	CPU running a CoreMark algorithm, CPU is running on Flash with 3 wait states with GCLKIN as reference	25°C	106*freq + 78	106*freq + 136	108*freq + 196	µA (with freq in MHz)
		85°C	106*freq + 154	108*freq + 232	108*freq + 310	

37.12.4 32.768kHz Internal oscillator (OSC32K) Characteristics

Table 37-53. 32kHz RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
f_{OUT}	Output frequency	Calibrated against a 32.768kHz reference at 25°C, over [-40, +85]C, over [1.62, 3.63]V	28.508	32.768	34.734	kHz
		Calibrated against a 32.768kHz reference at 25°C, at V_{DD}=3.3V	32.276	32.768	33.260	
		Calibrated against a 32.768kHz reference at 25°C, over [1.62, 3.63]V	31.457	32.768	34.079	
I_{OSC32K}	Current consumption		-	0.67	1.31	µA
$t_{STARTUP}$	Start-up time		-	1	2	cycle
Duty	Duty Cycle		-	50	-	%

37.12.5 Ultra Low Power Internal 32kHz RC Oscillator (OSCULP32K) Characteristics

Table 37-54. Ultra Low Power Internal 32kHz RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
f_{OUT}	Output frequency	Calibrated against a 32.768kHz reference at 25°C, over [-40, +85]C, over [1.62, 3.63]V	25.559	32.768	38.011	kHz
		Calibrated against a 32.768kHz reference at 25°C, at V_{DD}=3.3V	31.293	32.768	34.570	
		Calibrated against a 32.768kHz reference at 25°C, over [1.62, 3.63]V	31.293	32.768	34.570	
$i_{OSCULP32K}$(1)(2)			-	-	125	nA
$t_{STARTUP}$	Start-up time		-	10	-	cycles
Duty	Duty Cycle		-	50	-	%

Notes: 1. These values are based on simulation. These values are not covered by test limits in production or characterization.

2. This oscillator is always on.

Table 37-61. SPI Timing Characteristics and Requirements(1)

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
t_{SCK}	SCK period	Master			84		ns
t_{SCKW}	SCK high/low width	Master		-	$0.5*t_{SCK}$	-	
t_{SCKR}	SCK rise time(2)	Master		-	-	-	
t_{SCKF}	SCK fall time(2)	Master		-	-	-	
t_{MIS}	MISO setup to SCK	Master		-	21	-	
t_{MIH}	MISO hold after SCK	Master		-	13	-	
t_{MOS}	MOSI setup SCK	Master		-	$t_{SCK}/2 - 3$	-	
t_{MOH}	MOSI hold after SCK	Master		-	3	-	
t_{SSCK}	Slave SCK Period	Slave		$1*t_{CLK\_APB}$	-	-	
t_{SSCKW}	SCK high/low width	Slave		$0.5*t_{SSCK}$	-	-	
t_{SSCKR}	SCK rise time(2)	Slave		-	-	-	
t_{SSCKF}	SCK fall time(2)	Slave		-	-	-	
t_{SIS}	MOSI setup to SCK	Slave		$t_{SSCK}/2 - 9$	-	-	
t_{SIH}	MOSI hold after SCK	Slave		$t_{SSCK}/2 - 3$	-	-	
t_{SSS}	SS setup to SCK	Slave	PRELOADEN =1	$2*t_{CLK\_APB} + t_{SOS}$	-	-	
			PRELOADEN =0	$t_{SOS}+7$	-	-	
t_{SSH}	SS hold after SCK	Slave		$t_{SIH} - 4$	-	-	
t_{SOS}	MISO setup SCK	Slave		-	$t_{SSCK}/2 - 18$	-	
t_{SOH}	MISO hold after SCK	Slave		-	18	-	
t_{SOSS}	MISO setup after SS low	Slave		-	18	-	
t_{SOSH}	MISO hold after SS high	Slave		-	10	-	

Notes: 1. These values are based on simulation. These values are not covered by test limits in production.

2. See I/O Pin Characteristics

37.15.3 SERCOM in I²C Mode Timing

This section describes the requirements for devices connected to the I²C Interface Bus.

Do not use the TCC interrupts FAULT1, FAULT0, FAULTB, FAULTA, DFS, ERR, or CNT to wake up the chip from standby mode.

3 – If the OVF flag in the INTFLAG register is already set when enabling the DMA, this will trigger an immediate DMA transfer and overwrite the current buffered value in the TCC register.
Errata reference: 12127
Fix/Workaround:
None

4 – Advance capture mode (CAPTMIN CAPTMAX LOCMIN LOCMAX DERIV0) doesn't work if an upper channel is not in one of these mode. Example: when CC[0]=CAPTMIN, CC[1]=CAPTMAX, CC[2]=CAPTEN, and CC[3]=CAPTEN, CAPTMIN and CAPTMAX won't work.
Errata reference: 14817
Fix/Workaround:
Basic capture mode must be set in lower channel and advance capture mode in upper channel.
Example: CC[0]=CAPTEN , CC[1]=CAPTEN , CC[2]=CAPTMIN, CC[3]=CAPTMAX
All capture will be done as expected.

5 – In RAMP 2 mode with Fault keep, qualified and restart:
If a fault occurred at the end of the period during the qualified state, the switch to the next ramp can have two restarts.
Errata reference: 13262
Fix/Workaround:
Avoid faults few cycles before the end or the beginning of a ramp.

6 – With blanking enabled, a recoverable fault that occurs during the first increment of a rising TCC is not blanked.
Errata reference: 12519
Fix/Workaround:
None

7 – In Dual slope mode a Retrigger Event does not clear the TCC counter.
Errata reference: 12354
Fix/Workaround:
None

8 – In two ramp mode, two events will be generated per cycle, one on each ramp's end. EVCTRL.CNTSEL.END cannot be used to identify the end of a double ramp cycle.
Errata reference: 12224
Fix/Workaround:
None

9 – If an input event triggered STOP action is performed at the same time as the counter overflows, the first pulse width of the subsequent counter start can be altered with one prescaled clock cycle.
Errata reference: 12107
Fix/Workaround:
None

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.