Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

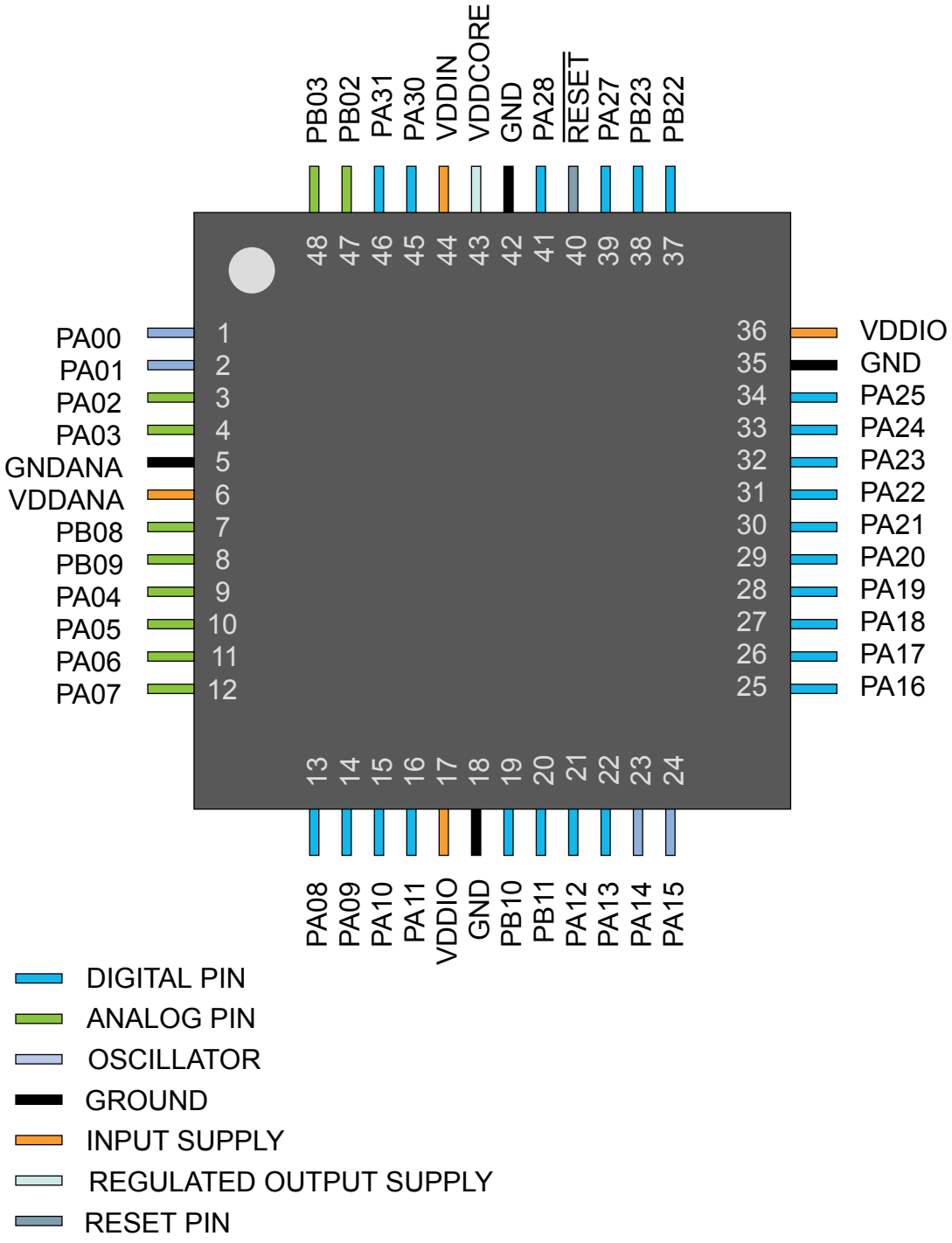
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	38
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 14x12b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamd21g15b-au

5.2 SAM D21G

5.2.1 QFN48 / TQFP48



6. Signal Descriptions List

The following table gives details on signal names classified by peripheral.

Signal Name	Function	Type	Active Level
Analog Comparators - AC			
AIN[3:0]	AC Analog Inputs	Analog	
CMP[:0]	AC Comparator Outputs	Digital	
Analog Digital Converter - ADC			
AIN[19:0]	ADC Analog Inputs	Analog	
VREFA	ADC Voltage External Reference A	Analog	
VREFB	ADC Voltage External Reference B	Analog	
Digital Analog Converter - DAC			
VOUT	DAC Voltage output	Analog	
VREFA	DAC Voltage External Reference	Analog	
External Interrupt Controller			
EXTINT[15:0]	External Interrupts	Input	
NMI	External Non-Maskable Interrupt	Input	
Generic Clock Generator - GCLK			
GCLK_IO[7:0]	Generic Clock (source clock or generic clock generator output)	I/O	
Inter-IC Sound Controller - I2S			
MCK[1:0]	Master Clock	I/O	
SCK[1:0]	Serial Clock	I/O	
FS[1:0]	I2S Word Select or TDM Frame Sync	I/O	
SD[1:0]	Serial Data Input or Output	I/O	
Power Manager - PM			
RESETN	Reset	Input	Low
Serial Communication Interface - SERCOMx			
PAD[3:0]	SERCOM I/O Pads	I/O	
System Control - SYSCTRL			
XIN	Crystal Input	Analog/ Digital	
XIN32	32kHz Crystal Input	Analog/ Digital	
XOUT	Crystal Output	Analog	
XOUT32	32kHz Crystal Output	Analog	

8.4.1 Power-On Reset on VDDANA

POR monitors VDDANA. It is always activated and monitors voltage at startup and also during all the sleep modes. If VDDANA goes below the threshold voltage, the entire chip is reset.

8.4.2 Brown-Out Detector on VDDANA

BOD33 monitors VDDANA. Refer to *SYSCTRL – System Controller* for details.

Related Links

SYSCTRL – System Controller

8.4.3 Brown-Out Detector on VDDCORE

Once the device has started up, BOD12 monitors the internal VDDCORE.

The initial value used for the CRC32 calculation must be written to the Data register (DATA). This value will usually be 0xFFFFFFFF, but can be, for example, the result of a previous CRC32 calculation if generating a common CRC32 of separate memory blocks.

Once completed, the calculated CRC32 value can be read out of the Data register. The read value must be complemented to match standard CRC32 implementations or kept non-inverted if used as starting point for subsequent CRC32 calculations.

If the device is in protected state by the NVMCTRL security bit, it is only possible to calculate the CRC32 of the whole flash array when operated from the external address space. In most cases, this area will be the entire onboard non-volatile memory. The Address, Length and Data registers will be forced to predefined values once the CRC32 operation is started, and values written by the user are ignored. This allows the user to verify the contents of a protected device.

The actual test is started by writing a '1' in the 32-bit Cyclic Redundancy Check bit of the Control register (CTRL.CRC). A running CRC32 operation can be canceled by resetting the module (writing '1' to CTRL.SWRST).

Related Links

NVMCTRL – Non-Volatile Memory Controller

Security Bit

13.11.3.2 Interpreting the Results

The user should monitor the Status A register. When the operation is completed, STATUSA.DONE is set. Then the Bus Error bit of the Status A register (STATUSA.BERR) must be read to ensure that no bus error occurred.

13.11.4 Debug Communication Channels

The Debug Communication Channels (DCCO and DCC1) consist of a pair of registers with associated handshake logic, accessible by both CPU and debugger even if the device is protected by the NVMCTRL security bit. The registers can be used to exchange data between the CPU and the debugger, during run time as well as in debug mode. This enables the user to build a custom debug protocol using only these registers.

The DCC0 and DCC1 registers are accessible when the protected state is active. When the device is protected, however, it is not possible to connect a debugger while the CPU is running (STATUSA.CRSTEXT is not writable and the CPU is held under Reset).

Two Debug Communication Channel status bits in the Status B registers (STATUS.DCCDx) indicate whether a new value has been written in DCC0 or DCC1. These bits, DCC0D and DCC1D, are located in the STATUSB registers. They are automatically set on write and cleared on read.

Note: The DCC0 and DCC1 registers are shared with the on-board memory testing logic (MBIST). Accordingly, DCC0 and DCC1 must not be used while performing MBIST operations.

Related Links

NVMCTRL – Non-Volatile Memory Controller

Security Bit

13.11.5 Testing of On-Board Memories MBIST

The DSU implements a feature for automatic testing of memory also known as MBIST (memory built-in self test). This is primarily intended for production test of on-board memories. MBIST cannot be operated from the external address range when the device is protected by the NVMCTRL security bit. If an MBIST command is issued when the device is protected, a protection error is reported in the Protection Error bit in the Status A register (STATUSA.PERR).

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	x	x	x	x				

Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							1	x

Bits 31:12 – ADDOFF[19:0]: Address Offset
The base address of the component, relative to the base address of this ROM table.

Bit 1 – FMT: Format
Always reads as '1', indicating a 32-bit ROM table.

Bit 0 – EPRES: Entry Present
This bit indicates whether an entry is present at this location in the ROM table.

This bit is set at power-up if the device is not protected indicating that the entry is not present.

This bit is cleared at power-up if the device is not protected indicating that the entry is present.

13.13.11 CoreSight ROM Table Entry 1

Name: ENTRY1
Offset: 0x1004
Reset: 0xXXXXX00X
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

20.6.2.5 Data Transmission

Before the DMAC can perform a data transmission, a DMA channel has to be configured and enabled, its corresponding transfer descriptor has to be initialized, and the arbiter has to grant the DMA channel access as the active channel.

Once the arbiter has granted a DMA channel access as the active channel (refer to DMA Block Diagram section) the transfer descriptor for the DMA channel will be fetched from SRAM using the fetch bus, and stored in the internal memory for the active channel. For a new block transfer, the transfer descriptor will be fetched from the descriptor memory section (BASEADDR); For an ongoing block transfer, the descriptor will be fetched from the write-back memory section (WRBADDR). By using the data transfer bus, the DMAC will read the data from the current source address and write it to the current destination address. For further details on how the current source and destination addresses are calculated, refer to the section on Addressing.

The arbitration procedure is performed after each burst transfer. If the current DMA channel is granted access again, the block transfer counter (BTCNT) of the internal transfer descriptor will be decremented by the number of beats in a burst transfer, the optional output event Beat will be generated if configured and enabled, and the active channel will perform a new burst transfer. If a different DMA channel than the current active channel is granted access, the block transfer counter value will be written to the write-back section before the transfer descriptor of the newly granted DMA channel is fetched into the internal memory of the active channel.

When a block transfer has come to its end (BTCNT is zero), the Valid bit in the Block Transfer Control register will be cleared (BTCTRL.VALID=0) before the entire transfer descriptor is written to the write-back memory. The optional interrupts, Channel Transfer Complete and Channel Suspend, and the optional output event Block, will be generated if configured and enabled. After the last block transfer in a transaction, the Next Descriptor Address register (DESCADDR) will hold the value 0x00000000, and the DMA channel will either be suspended or disabled, depending on the configuration in the Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT). If the transaction has further block transfers pending, DESCADDR will hold the SRAM address to the next transfer descriptor to be fetched. The DMAC will fetch the next descriptor into the internal memory of the active channel and write its content to the write-back section for the channel, before the arbiter gets to choose the next active channel.

20.6.2.6 Transfer Triggers and Actions

A DMA transfer through a DMA channel can be started only when a DMA transfer request is detected, and the DMA channel has been granted access to the DMA. A transfer request can be triggered from software, from a peripheral, or from an event. There are dedicated Trigger Source selections for each DMA Channel Control B (CHCTRLB.TRIGSRC).

The trigger actions are available in the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT). By default, a trigger generates a request for a block transfer operation. If a single descriptor is defined for a channel, the channel is automatically disabled when a block transfer has been completed. If a list of linked descriptors is defined for a channel, the channel is automatically disabled when the last descriptor in the list is executed. If the list still has descriptors to execute, the channel will be waiting for the next block transfer trigger. When enabled again, the channel will wait for the next block transfer trigger. The trigger actions can also be configured to generate a request for a beat transfer (CHCTRLB.TRIGACT=0x2) or transaction transfer (CHCTRLB.TRIGACT=0x3) instead of a block transfer (CHCTRLB.TRIGACT=0x0).

Figure 20-7 shows an example where triggers are used with two linked block descriptors.

20.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0						CRCENABLE	DMAENABLE	SWRST
0x01		15:8					LVLEN3	LVLEN2	LVLEN1	LVLEN0
0x02	CRCCTRL	7:0					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
0x03		15:8			CRCSRC[5:0]					
0x04	CRCDATAIN	7:0	CRCDATAIN[7:0]							
0x05		15:8	CRCDATAIN[15:8]							
0x06		23:16	CRCDATAIN[23:16]							
0x07		31:24	CRCDATAIN[31:24]							
0x08	CRCCHKSUM	7:0	CRCCHKSUM[7:0]							
0x09		15:8	CRCCHKSUM[15:8]							
0x0A		23:16	CRCCHKSUM[23:16]							
0x0B		31:24	CRCCHKSUM[31:24]							
0x0C	CRCSTATUS	7:0							CRCZERO	CRCBUSY
0x0D	DBGCTRL	7:0								DBGRUN
0x0E	QOSCTRL	7:0			DQOS[1:0]		FQOS[1:0]		WRBQOS[1:0]	
0x0F	Reserved									
0x10	SWTRIGCTRL	7:0	SWTRIG7	SWTRIG6	SWTRIG5	SWTRIG4	SWTRIG3	SWTRIG2	SWTRIG1	SWTRIG0
0x11		15:8					SWTRIG11	SWTRIG10	SWTRIG9	SWTRIG8
0x12		23:16								
0x13		31:24								
0x14	PRICTRL0	7:0	RRLVLEN0				LVLPRI0[3:0]			
0x15		15:8	RRLVLEN1				LVLPRI1[3:0]			
0x16		23:16	RRLVLEN2				LVLPRI2[3:0]			
0x17		31:24	RRLVLEN3				LVLPRI3[3:0]			
0x18 ... 0x1F	Reserved									
0x20	INTPEND	7:0					ID[3:0]			
0x21		15:8	PEND	BUSY	FERR			SUSP	TCMPL	TERR
0x22 ... 0x23	Reserved									
0x24	INTSTATUS	7:0	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
0x25		15:8					CHINT11	CHINT10	CHINT9	CHINT8
0x26		23:16								
0x27		31:24								
0x28	BUSYCH	7:0	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
0x29		15:8					BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
0x2A		23:16								
0x2B		31:24								
0x2C	PENDCH	7:0	PENDCH7	PENDCH6	PENDCH5	PENDCH4	PENDCH3	PENDCH2	PENDCH1	PENDCH0
0x2D		15:8					PENDCH11	PENDCH10	PENDCH9	PENDCH8
0x2E		23:16								
0x2F		31:24								

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

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

20.8.22 Channel Interrupt Flag Status and Clear

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHINTFLAG
Offset: 0x4E
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP: Channel Suspend
This flag is cleared by writing a '1' to it.

This flag is set when a block transfer with suspend block action is completed, when a software suspend command is executed, when a suspend event is received or when an invalid descriptor is fetched by the DMA.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Suspend interrupt flag for the corresponding channel.

For details on available software commands, refer to CHCTRLB.CMD.

For details on available event input actions, refer to CHCTRLB.EVACT.

For details on available block actions, refer to BTCTRL.BLOCKACT.

Bit 1 – TCMPL: Channel Transfer Complete
This flag is cleared by writing a '1' to it.

This flag is set when a block transfer is completed and the corresponding interrupt block action is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Transfer Complete interrupt flag for the corresponding channel.

Bit 0 – TERR: Channel Transfer Error
This flag is cleared by writing a '1' to it.

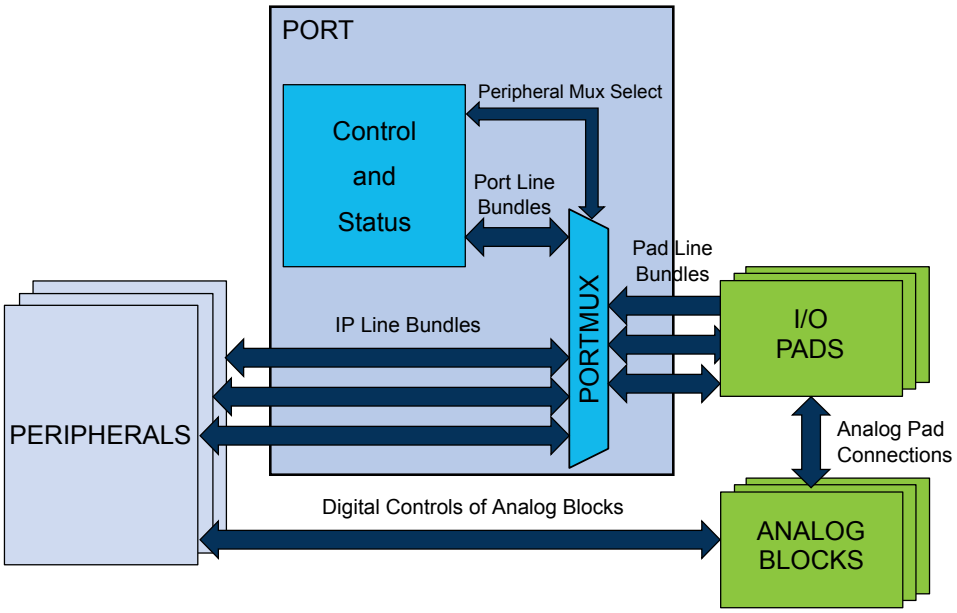
This flag is set when a bus error is detected during a beat transfer or when the DMAC fetches an invalid descriptor.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Transfer Error interrupt flag for the corresponding channel.

23.3 Block Diagram

Figure 23-1. PORT Block Diagram



23.4 Signal Description

Table 23-1. Signal description for PORT

Signal name	Type	Description
Pxy	Digital I/O	General-purpose I/O pin y in group x

Refer to the *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations

23.5 Product Dependencies

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

23.5.1 I/O Lines

The I/O lines of the PORT are mapped to pins of the physical device. The following naming scheme is used:

Each line bundle with up to 32 lines is assigned an identifier 'xy', with letter x=A, B, C… and two-digit number y=00, 01, …31. Examples: A24, C03.

PORT pins are labeled 'Pxy' accordingly, for example PA24, PC03. This identifies each pin in the device uniquely.

Each pin may be controlled by one or more peripheral multiplexer settings, which allow the pad to be routed internally to a dedicated peripheral function. When the setting is enabled, the selected peripheral

Bit	23	22	21	20	19	18	17	16
	DIRTGL[23:16]							
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
	DIRTGL[15:8]							
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
	DIRTGL[7:0]							
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 31:0 – DIRTGL[31:0]: Port Data Direction Toggle
Writing '0' to a bit has no effect.

Writing '1' to a bit will toggle the corresponding bit in the DIR register, which reverses the direction of the I/O pin.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The direction of the corresponding I/O pin is toggled.

23.8.5 Data Output Value

This register sets the data output drive value for the individual I/O pins in the PORT.
This register can be manipulated without doing a read-modify-write operation by using the Data Output Value Clear (OUTCLR), Data Output Value Set (OUTSET), and Data Output Value Toggle (OUTTGL) registers.

Name: OUT
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	OUT[31:24]							
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
	OUT[23:16]							
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
	OUT[15:8]							
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

24.4 Signal Description

Not applicable.

24.5 Product Dependencies

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

24.5.1 I/O Lines

Not applicable.

24.5.2 Power Management

The EVSYS can be used to wake up the CPU from all sleep modes, even if the clock used by the EVSYS channel and the EVSYS bus clock are disabled. Refer to the *PM – Power Manager* for details on the different sleep modes.

In all sleep modes, although the clock for the EVSYS is stopped, the device still can wake up the EVSYS clock. Some event generators can generate an event when their clocks are stopped.

Related Links

PM – Power Manager

24.5.3 Clocks

The EVSYS bus clock (CLK_EVSYS_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_EVSYS_APB can be found in *Peripheral Clock Masking*.

Each EVSYS channel has a dedicated generic clock (GCLK_EVSYS_CHANNEL_n). These are used for event detection and propagation for each channel. These clocks must be configured and enabled in the generic clock controller before using the EVSYS. Refer to *GCLK - Generic Clock Controller* for details.

Related Links

Peripheral Clock Masking

GCLK - Generic Clock Controller

24.5.4 DMA

Not applicable.

24.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the EVSYS interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

24.5.6 Events

Not applicable.

24.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging.

29.8 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0			SEREN1	SEREN0	CKEN1	CKEN0	ENABLE	SWRST
0x01 ... 0x03	Reserved									
0x04	CLKCTRLn0	7:0	BITDELAY	FSWIDTH[1:0]		NBSLOTS[2:0]			SLOTSIZE[1:0]	
0x05		15:8				SCKSEL	FSINV			FSSEL
0x06		23:16	MCKDIV[4:0]					MCKEN		MCKSEL
0x07		31:24	MCKOUTINV	SCKOUTINV	FSOUTINV	MCKOUTDIV[4:0]				
0x08	CLKCTRLn1	7:0	BITDELAY	FSWIDTH[1:0]		NBSLOTS[2:0]			SLOTSIZE[1:0]	
0x09		15:8				SCKSEL	FSINV			FSSEL
0x0A		23:16	MCKDIV[4:0]					MCKEN		MCKSEL
0x0B		31:24	MCKOUTINV	SCKOUTINV	FSOUTINV	MCKOUTDIV[4:0]				
0x0C	INTENCLR	7:0			RXOR1	RXOR0			RXRDY1	RXRDY0
0x0D		15:8			TXUR1	TXUR0			TXRDY1	TXRDY0
0x0E ... 0x0F	Reserved									
0x10	INTENSET	7:0			RXOR1	RXOR0			RXRDY1	RXRDY0
0x11		15:8			TXUR1	TXUR0			TXRDY1	TXRDY0
0x12 ... 0x13	Reserved									
0x14	INTFLAG	7:0			RXOR1	RXOR0			RXRDY1	RXRDY0
0x15		15:8			TXUR1	TXUR0			TXRDY1	TXRDY0
0x16 ... 0x17	Reserved									
0x18	SYNCBUSY	7:0			SEREN1	SEREN0	CKEN1	CKEN0	ENABLE	SWRST
0x19		15:8							DATA1	DATA0
0x1A ... 0x1F	Reserved									
0x20	SERCTRLn0	7:0	SLOTADJ		CLKSEL	TXSAME	TXDEFAULT[1:0]		SERMODE[1:0]	
0x21		15:8	BITREV	EXTEND[1:0]		WORDADJ		DATASIZE[2:0]		
0x22		23:16	SLOTDIS8	SLOTDIS7	SLOTDIS6	SLOTDIS5	SLOTDIS4	SLOTDIS3	SLOTDIS1	SLOTDIS0
0x23		31:24						RXLOOP	DMA	MONO
0x24	SERCTRLn1	7:0	SLOTADJ		CLKSEL	TXSAME	TXDEFAULT[1:0]		SERMODE[1:0]	
0x25		15:8	BITREV	EXTEND[1:0]		WORDADJ		DATASIZE[2:0]		
0x26		23:16	SLOTDIS8	SLOTDIS7	SLOTDIS6	SLOTDIS5	SLOTDIS4	SLOTDIS3	SLOTDIS1	SLOTDIS0
0x27		31:24						RXLOOP	DMA	MONO
0x28 ... 0x2F	Reserved									

When a start event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT0=0x3, START), enabling the counter will not start the counter. The counter will start on the next incoming event, but it will not restart on subsequent events.

Count Event Action

The TCC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR).

The count event action is selected by the Event Action 0 bit group in the Event Control register (EVCTRL.EVACT0=0x5, COUNT).

Direction Event Action

The direction event action can be selected in the Event Control register (EVCTRL.EVACT1=0x2, DIR). When this event is used, the asynchronous event path specified in the event system must be configured or selected. The direction event action can be used to control the direction of the counter operation, depending on external events level. When received, the event level overrides the Direction settings (CTRLBSET.DIR or CTRLBCLR.DIR) and the direction bit value is updated accordingly.

Increment Event Action

The increment event action can be selected in the Event Control register (EVCTRL.EVACT0=0x4, INC) and can change the counter state when an event is received. When the TCE0 event (TCCx_EV0) is received, the counter increments, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

Decrement Event Action

The decrement event action can be selected in the Event Control register (EVCTRL.EVACT1=0x4, DEC) and can change the counter state when an event is received. When the TCE1 (TCCx_EV1) event is received, the counter decrements, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

Non-recoverable Fault Event Action

Non-recoverable fault actions can be selected in the Event Control register (EVCTRL.EVACTn=0x7, FAULT). When received, the counter will be stopped and the output of the compare channels is overridden according to the Driver Control register settings (DRVCTRL.NREx and DRVCTRL.NRVx). TCE0 and TCE1 must be configured as asynchronous events.

Event Action Off

If the event action is disabled (EVCTRL.EVACTn=0x0, OFF), enabling the counter will also start the counter.

31.6.2.5 Compare Operations

By default, the Compare/Capture channel is configured for compare operations. To perform capture operations, it must be re-configured.

When using the TCC with the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The Channel x Compare/Capture Buffer Value (CCBx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a force update command (CTRLBSET.CMD=0x3, UPDATE). For further details, refer to Double Buffering. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

Bit 11 – DFS: Non-Recoverable Debug Fault Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Debug Fault State Interrupt Disable/Enable bit, which disables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 3 – ERR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Disable/Enable bit, which disables the Compare interrupt.

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

Bit 2 – CNT: Counter Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Counter Interrupt Disable/Enable bit, which disables the Counter interrupt.

Value	Description
0	The Counter interrupt is disabled.
1	The Counter interrupt is enabled.

Bit 1 – TRG: Retrigger Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Retrigger Interrupt Disable/Enable bit, which disables the Retrigger interrupt.

Value	Description
0	The Retrigger interrupt is disabled.
1	The Retrigger interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Disable/Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

31.8.11 Interrupt Enable Set

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

Name: INTENSET

Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
		TRIM[2:0]				TRANSN[4:2]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

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

Bits 14:12 – TRIM[2:0]: Trim bits for DP/DM
These bits calibrate the matching of rise/fall of DP/DM.

Bits 10:6 – TRANSN[4:0]: Trimmable Output Driver Impedance N
These bits calibrate the NMOS output impedance of DP/DM drivers.

Bits 4:0 – TRANSP[4:0]: Trimmable Output Driver Impedance P
These bits calibrate the PMOS output impedance of DP/DM drivers.

32.8.2 Device Registers - Common

32.8.2.1 Control B

Name: CTRLB
Offset: 0x08
Reset: 0x0001
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
					LPMHDSK[1:0]		GNAK	
Access					R/W	R/W	R/W	
Reset					0	0	0	

Bit	7	6	5	4	3	2	1	0
				NREPLY	SPDCONF[1:0]		UPRSM	DETACH
Access				R	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 11:10 – LPMHDSK[1:0]: Link Power Management Handshake
These bits select the Link Power Management Handshake configuration.

Value	Description
0x0	No handshake. LPM is not supported.
0x1	ACK
0x2	NYET
0x3	Reserved

PTOKEN[1:0](1)	Description
0x2	OUT
0x3	Reserved

1. PTOKEN field is ignored when PTYPE is configured as EXTENDED.
2. Available only when PTYPE is configured as CONTROL

Theses bits are cleared upon sending a USB reset.

32.8.6.2 Interval for the Bulk-Out/Ping Transaction

Name: BINTERVAL
Offset: 0x103 + (n x 0x20)
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BINTERVAL[7:0]							
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 7:0 – BINTERVAL[7:0]: BINTERVAL
These bits contains the Ping/Bulk-out period.

These bits are cleared when a USB reset is sent or when PEN[n] is zero.

BINTERVAL	Description
=0	Multiple consecutive OUT token is sent in the same frame until it is acked by the peripheral
>0	One OUT token is sent every BINTERVAL frame until it is acked by the peripheral

Depending from the type of pipe the desired period is defined as:

PTYPE	Description
Interrupt	1 ms to 255 ms
Isochronous	2^(Binterval) * 1 ms
Bulk or control	1 ms to 255 ms
EXT LPM	bInterval ignored. Always 1 ms when a NYET is received.

32.8.6.3 Pipe Status Clear n

Name: PSTATUSCLR
Offset: 0x104 + (n x 0x20)
Reset: 0x00
Property: PAC Write-Protection

Offset: 0x20
Reset: 0x0000
Property: Write-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	WINUT[15:8]							
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
	WINUT[7:0]							
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 15:0 – WINUT[15:0]: Window Upper Threshold
If the window monitor is enabled, these bits define the upper threshold value.

33.8.17 Gain Correction

Name: GAINCORR
Offset: 0x24
Reset: 0x0000
Property: Write-Protected

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

Bit	7	6	5	4	3	2	1	0
	GAINCORR[7:0]							
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 11:0 – GAINCORR[11:0]: Gain Correction Value
If the CTRLB.CORREN bit is one, these bits define how the ADC conversion result is compensated for gain error before being written to the result register. The gain-correction is a fractional value, a 1-bit integer plusan 11-bit fraction, and therefore 1/2 <= GAINCORR < 2. GAINCORR values range from 0.10000000000 to 1.11111111111.

33.8.18 Offset Correction

Name: OFFSETCORR
Offset: 0x26
Reset: 0x0000
Property: Write-Protected

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
	Variation over V_{DDANA} voltage	V_{DDANA}=1.62V to 3.6V	-1.7	1	3.7	mV/V
	Temperature Sensor accuracy	Using the method described in the Software-based Refinement of the Actual Temperature	-10	-	10	°C

Table 37-38. Temperature Sensor Characteristics(1) (Device Variant B and C)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
	Temperature sensor output voltage	T= 25°C, V_{DDANA} = 3.3V	-	0.688	-	V
	Temperature sensor slope		2.06	2.16	2.26	mV/°C
	Variation over V_{DDANA} voltage	V_{DDANA}=1.62V to 3.6V	-0.4	1.4	3	mV/V
	Temperature Sensor accuracy	Using the method described in the Software-based Refinement of the Actual Temperature	-10	-	10	°C

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

37.10.8.2 Software-based Refinement of the Actual Temperature

The temperature sensor behavior is linear but depends on several parameters such as the internal voltage reference, which itself depends on the temperature. To take this into account, each device contains a Temperature Log row with data measured and written during the production tests. These calibration values should be read by software to infer the most accurate temperature readings possible.

This Software Temperature Log row can be read at address 0x00806030

This section specifies the Temperature Log row content and explains how to refine the temperature sensor output using the values in the Temperature Log row.

Temperature Log Row

All values in this row were measured in the following conditions:

- V_{DDIN} = V_{DDIO} = V_{DDANA} = 3.3V
- ADC Clock speed = 1MHz
- ADC mode: Free running mode, ADC averaging mode with 4 averaged samples
- ADC voltage reference = 1.0V internal reference (INT1V)
- ADC input = Temperature sensor

Table 37-39. Temperature Log Row Content

Bit position	Name	Description
7:0	ROOM_TEMP_VAL_INT	Integer part of room temperature in °C
11:8	ROOM_TEMP_VAL_DEC	Decimal part of room temperature
19:12	HOT_TEMP_VAL_INT	Integer part of hot temperature in °C

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
I_{DFLL}	Power consumption on V_{DDIN}	f_{REF} = XTAL, 32 .768kHz, 100ppm DFLLMUL = 1464	-	425	482	µA
t_{LOCK}	Lock time	f_{REF} = XTAL, 32 .768kHz, 100ppm DFLLMUL = 1464 DFLLVAL.COARSE = DFLL48M COARSE CAL DFLLVAL.FINE = 512 DFLLCTRL.BPLCKC = 1 DFLLCTRL.QLDIS = 0 DFLLCTRL.CCDIS = 1 DFLLMUL.FSTEP = 10	100	200	500	µs

Table 37-52. DFLL48M Characteristics - Closed Loop Mode(1) (Device Variant B and C)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
f_{OUT}	Average Output frequency	f_{REF} = 32 .768kHz	47.963	47.972	47.981	MHz
f_{REF}	Reference frequency		0.732	32.768	33	kHz
Jitter	Cycle to Cycle jitter	f_{REF} = 32 .768kHz	-	-	0.42	ns
I_{DFLL}	Power consumption on V_{DDIN}	f_{REF} =32 .768kHz	-	403	453	µA
t_{LOCK}	Lock time	f_{REF} = 32 .768kHz DFLLVAL.COARSE = DFLL48M COARSE CAL DFLLVAL.FINE = 512 DFLLCTRL.BPLCKC = 1 DFLLCTRL.QLDIS = 0 DFLLCTRL.CCDIS = 1 DFLLMUL.FSTEP = 10	-	200	500	µs

1. To insure that the device stays within the maximum allowed clock frequency, any reference clock for DFLL in close loop must be within a 2% error accuracy.

Table 37-62. I²C Interface Timing (Device Variant A)

Symbol	Parameter		Conditions	Min.	Typ.	Max.	Units
t_R	Rise time for both SDA and SCL	Standard / Fast Mode	IC_b(2) = 400pF	-	215	300	ns
		Fast Mode +	IC_b(2) = 550pF		60	100	
		High Speed Mode	IC_b(2) = 100pF		20	40	
t_{OF}	Output fall time from V_{IHmin} to V_{ILmax}	Standard / Fast Mode	10pF < C_b(2) < 400pF		20.0	50.0	
		Fast Mode +	10pF < C_b(2) < 550pF		15.0	50.0	
		High Speed Mode	10pF < C_b(2) < 100pF		10.0	40.0	
$t_{HD;STA}$	Hold time (repeated) START condition		f_{SCL} > 100kHz, Master	t_{LOW}-9	-	-	
t_{LOW}	Low period of SCL Clock		f_{SCL} > 100kHz	113	-	-	
t_{BUF}	Bus free time between a STOP and a START condition		f_{SCL} > 100kHz	t_{LOW}	-	-	
$t_{SU;STA}$	Setup time for a repeated START condition		f_{SCL} > 100kHz, Master	t_{LOW}+7	-	-	
$t_{HD;DAT}$	Data hold time		f_{SCL} > 100kHz, Master	9	-	12	
$t_{SU;DAT}$	Data setup time		f_{SCL} > 100kHz, Master	104	-	-	
$t_{SU;STO}$	Setup time for STOP condition		f_{SCL} > 100kHz, Master	t_{LOW}+9	-	-	
$t_{SU;DAT;rx}$	Data setup time (receive mode)		f_{SCL} > 100kHz, Slave	51	-	56	
$t_{HD;DAT;tx}$	Data hold time (send mode)		f_{SCL} > 100kHz, Slave	71	90	138	