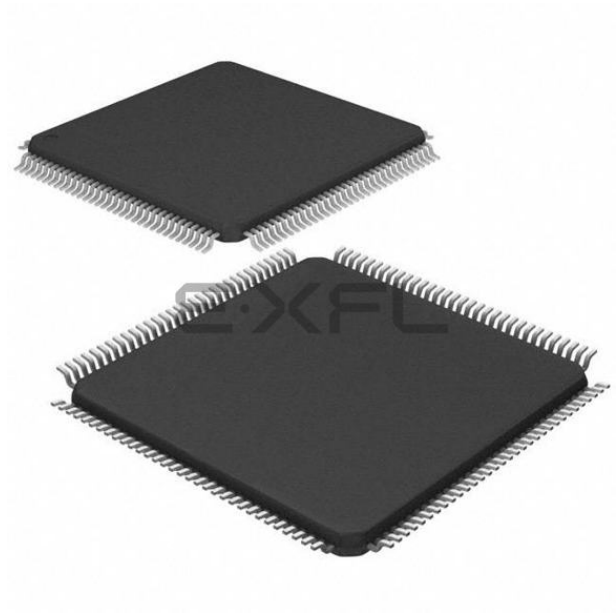


Welcome to E-XFL.COM

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®-M4F
Core Size	32-Bit Single-Core
Speed	120MHz
Connectivity	EBI/EMI, I²C, IrDA, LINbus, MMC/SD, QSPI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM
Number of I/O	99
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	192K x 8
Voltage - Supply (Vcc/Vdd)	1.71V ~ 3.63V
Data Converters	A/D 32x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	128-TQFP
Supplier Device Package	128-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamd51p19a-au

Table 6-29. TCC1 IO SET Configuration

TCC Signal	IOSET 1 PINs	IOSET 2 PINs	IOSET 3 PINs	IOSET 4 PINs	IOSET 5 PINs
WO0	PA16	PD20	PB18	PB10	PC14
WO1	PA17	PD21	PB19	PB11	PC15
WO2	PA18	PB20	PB26	PA12	PA14
WO3	PA19	PB21	PB27	PA13	PA15
WO4	PA20	PB28	PA08	PC10	N/A(1)
WO5	PA21	PB29	PA09	PC11	N/A(1)
WO6	PA22	PA10	PC12	N/A(1)	N/A(1)
WO7	PA23	PA11	PC13	N/A(1)	N/A(1)

Note: 1. The signal is available, but the edges are not aligned wrt. the other signals as specified.

Table 6-30. TCC2 IO SET Configuration

TCC Signal	IOSET 1 PINs	IOSET 2 PINs
WO0	PA14	PA30
WO1	PA15	PA31
WO2	PA24	PB02

Table 6-31. TCC3 IO SET Configuration

TCC Signal	IOSET 1 PINs	IOSET 2 PINs
WO0	PB12	PB16
WO1	PB13	PB17

Table 6-32. TCC4 IO SET Configuration

TCC Signal	IOSET 1 PINs	IOSET 2 PINs
WO0	PB14	PB30
WO1	PB15	PB31

6.2.8.6 PDEC IOSET Configurations

The following tables lists each IOSET Pins for PDEC instance.

Table 6-33. PDEC IO SET Configuration

PDEC Signal	IOSET 1 PINs	IOSET 2 PINs	IOSET 3 PINs	IOSET 4 PINs
QDI[0]	PC16	PB18	PA24	PB23
QDI[1]	PC17	PB19	PA25	PB24
QDI[2]	PC18	PB20	PB22	PB25

14.8.4 Peripheral Channel Control

Name: PCHCTRLm
Offset: 0x80 + m*0x04 [m=0..47]
Reset: 0x00000000
Property: PAC Write-Protection

PCHTRLm controls the settings of Peripheral Channel number m (m=[47:0]).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bit 7 – WRTLOCK Write Lock
After this bit is set to '1', further writes to the PCHCTRLm register will be discarded. The control register of the corresponding Generator n (GENCTRLn), as assigned in PCHCTRLm.GEN, will also be locked. It can only be unlocked by a Power Reset.

Note that Generator 0 cannot be locked.

Value	Description
0	The Peripheral Channel register and the associated Generator register are not locked
1	The Peripheral Channel register and the associated Generator register are locked

Bit 6 – CHEN Channel Enable
This bit is used to enable and disable a Peripheral Channel.

Value	Description
0	The Peripheral Channel is disabled
1	The Peripheral Channel is enabled

Bits 3:0 – GEN[3:0] Generator Selection
This bit field selects the Generator to be used as the source of a peripheral clock, as shown in the table below:

24.9.5 GMAC DMA Configuration Register

Name: DCFGR
Offset: 0x010
Reset: 0x00020004
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
								DDRP
Access								
Reset								0

Bit	23	22	21	20	19	18	17	16
	DRBS[7:0]							
Access								
Reset	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8
					TXCOEN	TXPBMS	RXBMS[1:0]	
Access								
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ESPA	ESMA		FBLDO[4:0]				
Access								
Reset	0	0		0	0	1	0	0

Bit 24 – DDRP DMA Discard Receive Packets
A write to this bit is ignored if the DMA is not configured in the packet buffer full store and forward mode.

Value	Description
0	Received packets are stored in the SRAM based packet buffer until next AHB buffer resource becomes available.
1	Receive packets from the receiver packet buffer memory are automatically discarded when no AHB resource is available.

Bits 23:16 – DRBS[7:0] DMA Receive Buffer Size
These bits defined by these bits determines the size of buffer to use in main AHB system memory when writing received data.

The value is defined in multiples of 64 bytes. For example:

- 0x02: 128 bytes
- 0x18: 1536 bytes (1 × max length frame/buffer)
- 0xA0: 10240 bytes (1 × 10K jumbo frame/buffer)

WARNING Do not write 0x00 to this bit field.

24.9.78 GMAC Receive Resource Errors Register

Name: RRE
Offset: 0x1A0
Reset: 0x00000000
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXRER[17:16]	
Access							R	R
Reset							0	0

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

Bit	7	6	5	4	3	2	1	0
	RXRER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 17:0 – RXRER[17:0] Receive Resource Errors
This bit field counts frames that are not an integral number of bytes long and have bad CRC when their length is truncated to an integral number of Bytes and are between 64 and 1518 Bytes in length (1536 if NCFGR.MAXFS=1). This bit field is also incremented if a symbol error is detected and the frame is of valid length and does not have an integral number of Bytes.

24.9.91 GMAC PTP Event Frame Transmitted Seconds Low Register

Name: EFTSL
Offset: 0x1E0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RUD[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RUD[31:0] Register Update
The register is updated with the value that the IEEE 1588 Timer Seconds Register holds when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the register is updated.

24.9.93 GMAC PTP Event Frame Received Seconds Low Register

Name: EFRSL
Offset: 0x1E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RUD[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RUD[31:0] Register Update
The register is updated with the value that the IEEE 1588 Timer Seconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.

24.9.98 GMAC PTP Peer Event Frame Received Nanoseconds Register

Name: PEFRN
Offset: 0x1FC
Reset: 0x00000000
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
			RUD[29:24]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 29:0 – RUD[29:0] Register Update
The register is updated with the value that the IEEE 1588 Timer Nanoseconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.

28.6.9 Synchronization

Due to the multiple clock domains, some registers in the DFLL48M must be synchronized when accessed. A register can require:

- Synchronization when written
- Synchronization when read
- No synchronization

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

The following registers need synchronization:

- ENABLE bit in DFLLCTRLA register - write-synchronized
- DFLLCTRLB register - read-synchronized
- DFLLVAL register - read- and write-synchronized
- DFLLMUL register - write-synchronized

Due to the multiple clock domains (XOSC32K, XOSC, GCLK and CK), some registers in the DPLL must be synchronized when accessed. A register can require:

- Synchronization when written
- No synchronization

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

The following bits need synchronization when written:

- Enable bit in control register A (DPLLnCTRLA.ENABLE)
- DPLLn Ratio register (DPLLnRATIO)

32. PORT - I/O Pin Controller

32.1 Overview

The IO Pin Controller (PORT) controls the I/O pins of the device. The I/O pins are organized in a series of groups, collectively referred to as a PORT group. Each PORT group can have up to 32 pins that can be configured and controlled individually or as a group. The number of PORT groups on a device may depend on the package/number of pins. Each pin may either be used for general-purpose I/O under direct application control or be assigned to an embedded device peripheral. When used for general-purpose I/O, each pin can be configured as input or output, with highly configurable driver and pull settings.

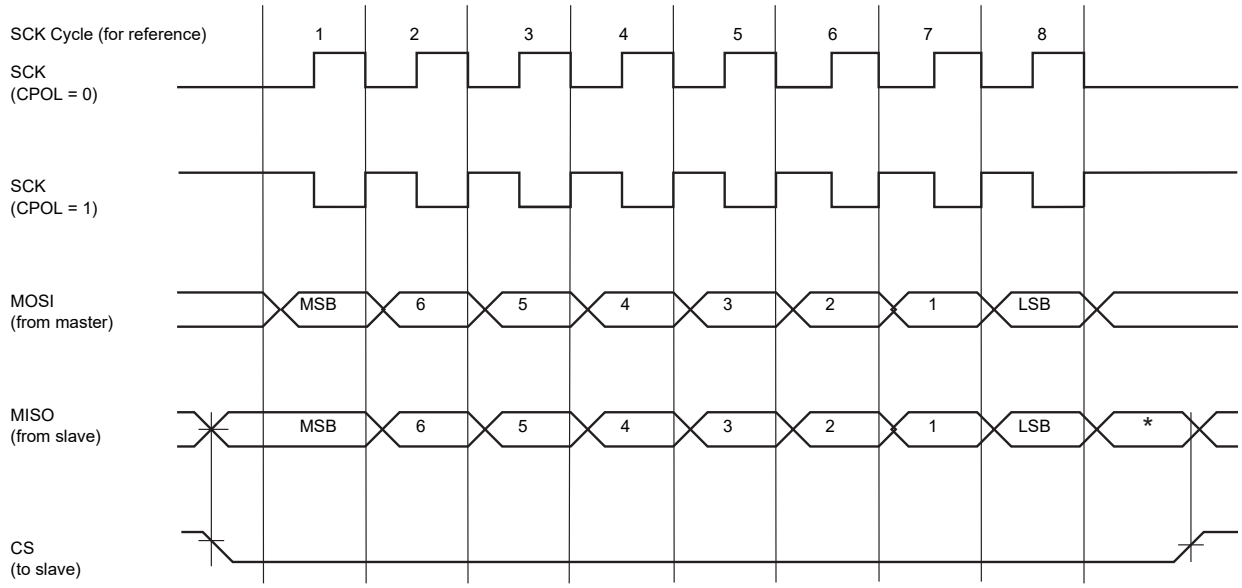
All I/O pins have true read-modify-write functionality when used for general-purpose I/O; the direction or the output value of one or more pins may be changed (set, reset or toggled) explicitly without unintentionally changing the state of any other pins in the same port group by a single, atomic 8-, 16- or 32-bit write.

The PORT is connected to the high-speed bus matrix through an AHB/APB bridge.

32.2 Features

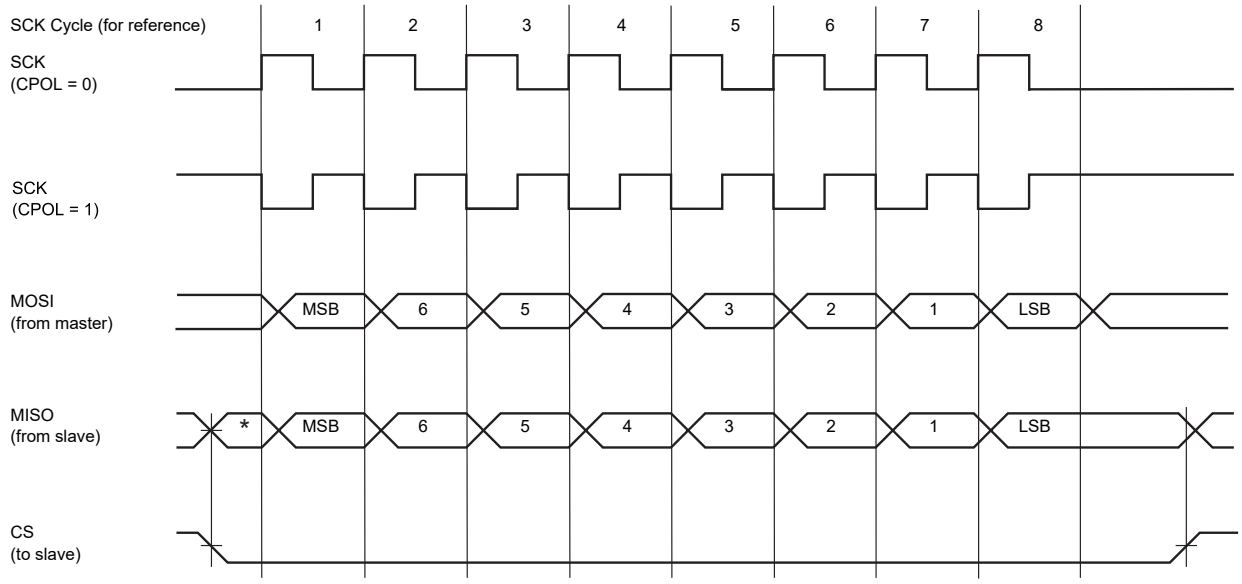
- Selectable input and output configuration for each individual pin
- Software-controlled multiplexing of peripheral functions on I/O pins
- Flexible pin configuration through a dedicated Pin Configuration register
- Configurable output driver and pull settings:
  - Totem-pole (push-pull)
  - Pull configuration
  - Driver strength
- Configurable input buffer and pull settings:
  - Internal pull-up or pull-down
  - Input sampling criteria
  - Input buffer can be disabled if not needed for lower power consumption
- Input event:
  - Up to four input event pins for each PORT group
  - SET/CLEAR/TOGGLE event actions for each event input on output value of a pin
  - Can be output to pin

Figure 37-3. QSPI Transfer Modes (BAUD.CPHA = 0, 8-bit transfer)



* Not defined, but normally MSB of previous character received

Figure 37-4. QSPI Transfer Modes (BAUD.CPHA = 1, 8-bit transfer)



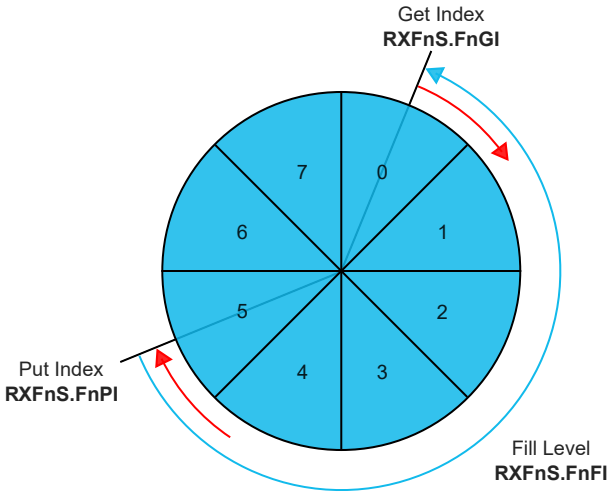
* Not defined, but normally LSB of previous character received

37.6.6 Transfer Delays

The QSPI supports several consecutive transfers while the chip select is active. Three delays can be programmed to modify the transfer waveforms:

- The delay between the inactivation and the activation of CS is programmed by writing the Minimum Inactive CS Delay bit field in the Control B register (CTRLB.DLYCS), allowing to tune the minimum time of CS at high level.
- The delay between consecutive transfers is programmed by writing the Delay Between Consecutive Transfers bit field in the Control B register (CTRLB.DLYBCT), allowing to insert a

Figure 39-7. Rx FIFO Status



When reading from an Rx FIFO, Rx FIFO Get Index RXFnS.FnGI • FIFO Element Size has to be added to the corresponding Rx FIFO start address RXFnC.FnSA.

Table 39-3. Rx Buffer / FIFO Element Size

RXESC.RBDS[2:0] RXESC.FnDS[2:0]	Data Field [bytes]	FIFO Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

Rx FIFO Blocking Mode

The Rx FIFO blocking mode is configured by RXFnC.FnOM = ‘0’. This is the default operation mode for the Rx FIFOs.

When an Rx FIFO full condition is reached (RXFnS.FnPI = RXFnS.FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signaled by RXFnS.FnF = ‘1’. In addition interrupt flag IR.RFnF is set.

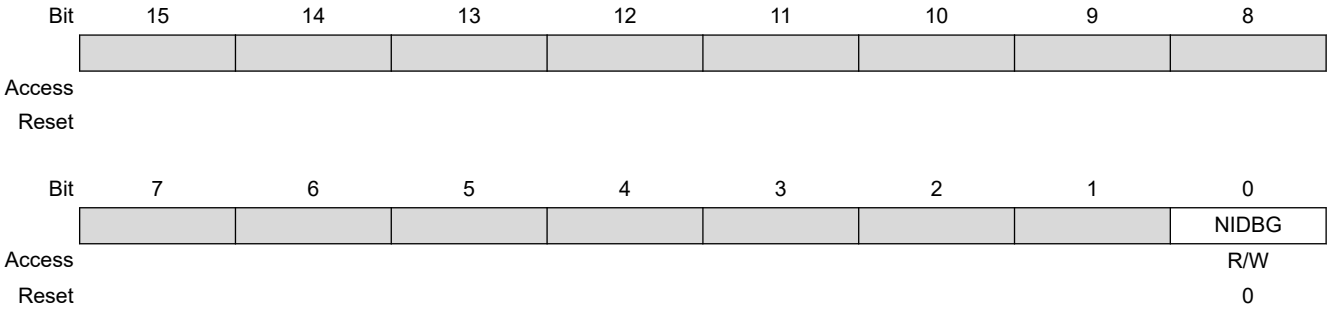
In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by RXFnS.RFnL = ‘1’. In addition interrupt flag IR.RFnL is set.

Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by RXFnC.FnOM = ‘1’.

40.8.42 Debug Register

Name: DBGR
Offset: 0x234
Reset: 0x00
Property: -



Bit 0 – NIDBG Non-Intrusive Debug

Value	Name	Description
0	DISABLED	Reading the BDPR via debugger increments the dual port RAM read pointer.
1	ENABLED	Reading the BDPR via debugger does not increment the dual port RAM read pointer.

Offset	Name	Bit Pos.								
		15:8								OFFSETCOR R
		23:16								
		31:24	BUSY							
0x40	RESULT	7:0	RESULT[7:0]							
		15:8	RESULT[15:8]							
0x42 ... 0x43	Reserved									
0x44	RESS	7:0	RESS[7:0]							
		15:8	RESS[15:8]							
0x46 ... 0x47	Reserved									
0x48	CALIB	7:0		BIASR2R[2:0]				BIASCOMP[2:0]		
		15:8						BIASREFBUF[2:0]		

45.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to the section on Synchronization.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization section.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Related Links

45.6.8 Synchronization

register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the DAC Controller is reset. See 47.8.6 INTFLAG for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

47.6.6 Events

The DAC Controller can generate the following output events:

- Data Buffer 0 Empty (EMPTY0): Generated when the internal data buffer of DAC0 is empty. Refer to 47.6.4 DMA Operation for details.
- Data Buffer 1 Empty (EMPTY1): Generated when the internal data buffer of DAC1 is empty. Refer to 47.6.4 DMA Operation for details.
- Filter 0 Result Ready (RESRDY0): Generated when standalone filter 0 result is ready.
- Filter 1 Result Ready (RESRDY1): Generated when standalone filter 1 result is ready.

Writing a '1' to an Event Output bit in the Event Control Register (EVCTRL.EMPTYEOx) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The DAC Controller can take the following actions on an input event:

- DAC0 Start Conversion (START0): DATABUF0 value is transferred into DATA0 as soon as DAC0 is ready for the next conversion, and then conversion is started. START0 is considered as asynchronous to GCLK_DAC, thus it is resynchronized in the DAC Controller. Refer to 47.6.2.4 Digital to Analog Conversion for details.
- DAC1 Start Conversion (START1): DATABUF1 value is transferred into DATA1 as soon as DAC1 is ready for the next conversion, and then conversion is started. START1 is considered as asynchronous to GCLK_DAC, thus it is resynchronized in the DAC Controller. Refer to 47.6.2.4 Digital to Analog Conversion for details.

Writing a '1' to an Event Input bit in the Event Control register (EVCTRL.STARTEIx) enables the corresponding action on input event. Writing a '0' to this bit will disable the corresponding action on input event.
Note: When several events are connected to the DAC Controller, the enabled action will be taken on any of the incoming events.

By default, DAC Controller detects rising edge events. Falling edge detection can be enabled by writing '1' to EVCTRL.INVEIx.

Note that if an event occurs before startup time is completed, DATAx is loaded but start of conversion is ignored.

47.6.7 Sleep Mode Operation

If the Run In Standby bit in the DAC Control x register DACCCTRLx.RUNSTDBY=1, the DACx will continue the conversions in standby sleep mode.

If DACCCTRLx.RUNSTDBY=0, the DACx will stop conversions in standby sleep mode.

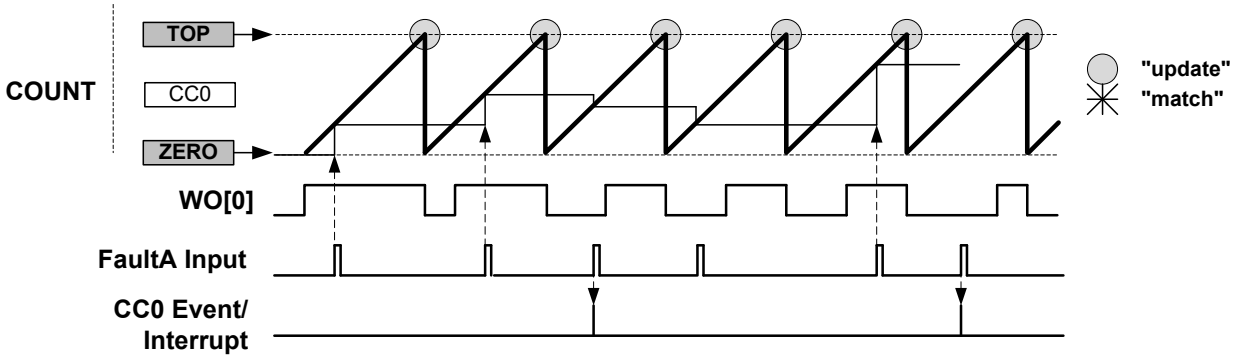
48.7.1.16 Period Buffer Value, 8-bit Mode

Name: PERBUF
Offset: 0x2F
Reset: 0xFF
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	PERBUF[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	1

Bits 7:0 – PERBUF[7:0] Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

Figure 49-29. Capture Action “DERIV0”



Hardware Halt Action

This is configured by writing 0x1 to the Fault n Halt mode bits in the Recoverable Fault n Configuration register (FCTRLn.HALT). When enabled, the timer/counter is halted and the cycle is extended as long as the corresponding fault is present.

The next figure ('Waveform Generation with Halt and Restart Actions') shows an example where both restart action and hardware halt action are enabled for Fault A. The compare channel 0 output is clamped to inactive level as long as the timer/counter is halted. The timer/counter resumes the counting operation as soon as the fault condition is no longer present. As the restart action is enabled in this example, the timer/counter is restarted after the fault condition is no longer present.

The figure after that ('Waveform Generation with Fault Qualification, Halt, and Restart Actions') shows a similar example, but with additionally enabled fault qualification. Here, counting is resumed after the fault condition is no longer present.

Note that in RAMP2 and RAMP2A operations, when a new timer/counter cycle starts, the cycle index will automatically change.

Figure 49-30. Waveform Generation with Halt and Restart Actions

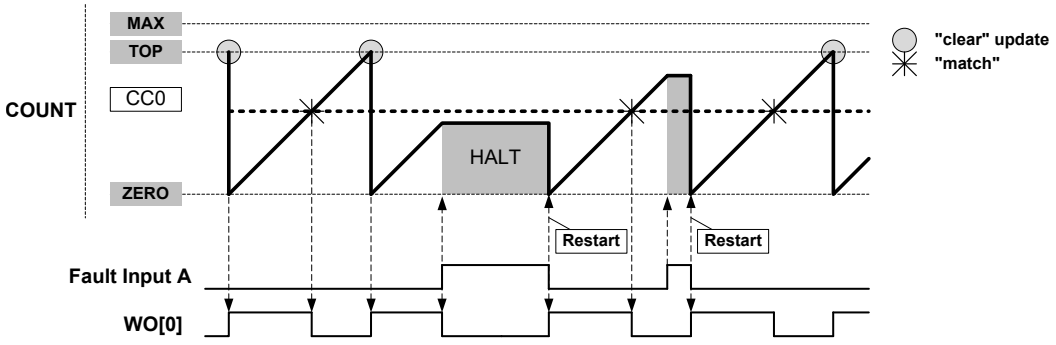
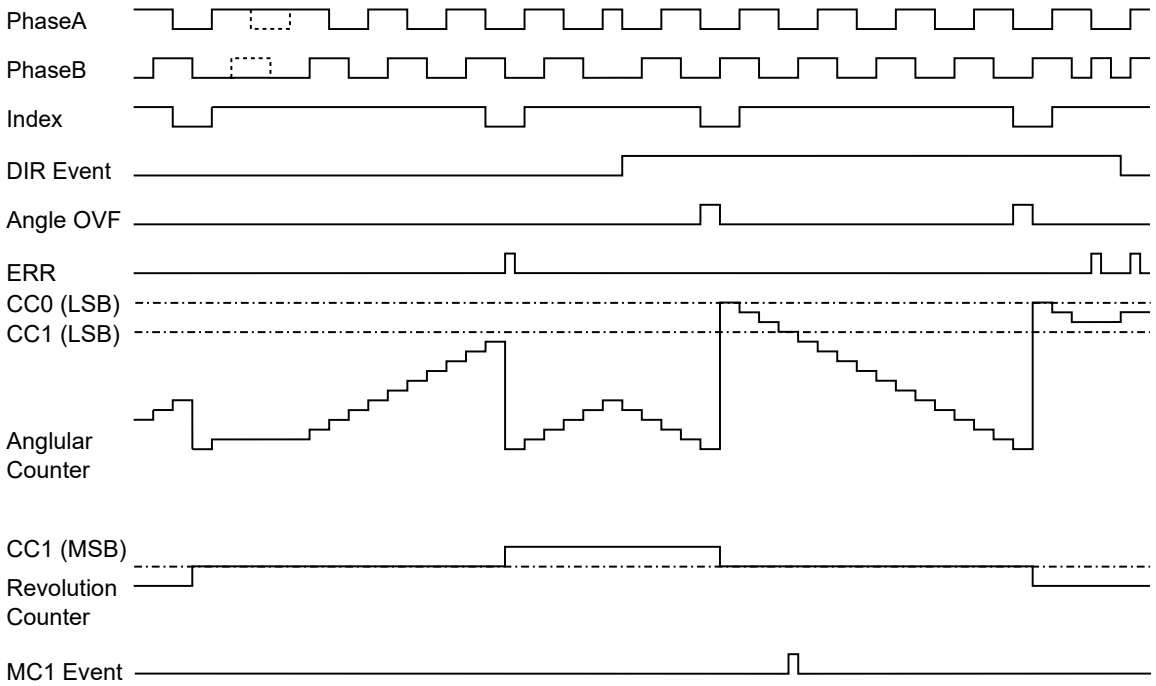


Figure 53-5. Position and Rotation Measurement



in Q4 and Q4S configuration, a valid index is detected when the three inputs (PhaseA, PhaseB and Index) are at low level.

In Q2 and Q2S configuration, a valid index is detected when the two inputs (Count and Index) are at low level.

in Q2 and Q4 configuration, depending on current detected direction, Index will reset or reload the Angular counter and increment or decrement the Revolution counter.

In Q2S and Q4S configuration, the Angular counter is reset on the first Index occurrence after the PDEC decoding is enabled. When any next Index occurrence does not match an Angular counter overflow or underflow, the Index Error flag in Status register is set (STATUS.IDXERR). The Error Interrupt Flag is set (INTFLAG.ERR) and an optional interrupt can be generated.

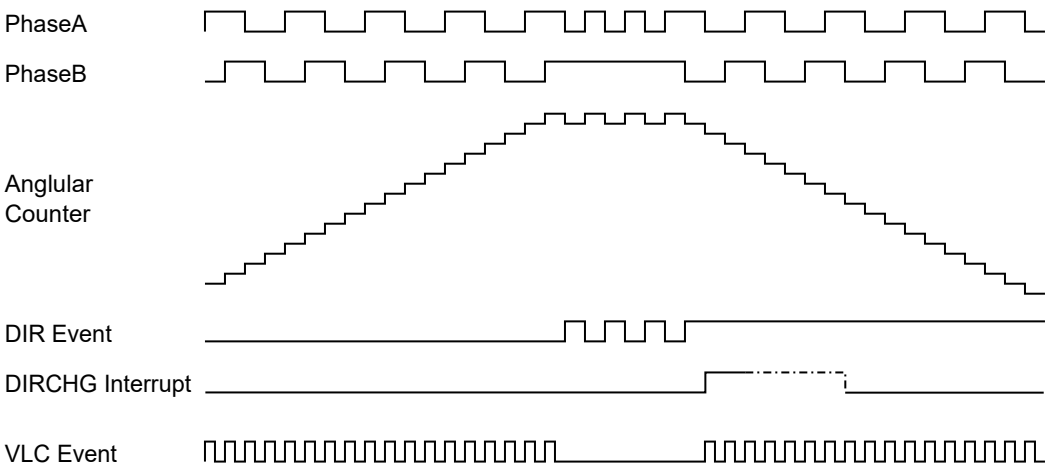
An Index Error is also generated after the PDEC decoding is enabled and no Index has been detected after one Angular counter revolution.

53.6.2.6.2 Direction Status and Change Detection

The direction (DIR) status can be directly read anytime in the STATUS register (STATUS.DIR). The polarity of the direction flag status depends of the input signal swap and active level configuration.

Each time a rotation direction change is detected, the Direction Change Interrupt Flag is set (INTFLAG.DIR) and an optional interrupt can be generated. The same interrupt condition is source of Direction event output.

Figure 53-6. Rotation Direction Change



To avoid spurious interrupts when coding wheel is stopped, the direction change condition is reported as an interrupt, only on the second edge confirming the direction change.

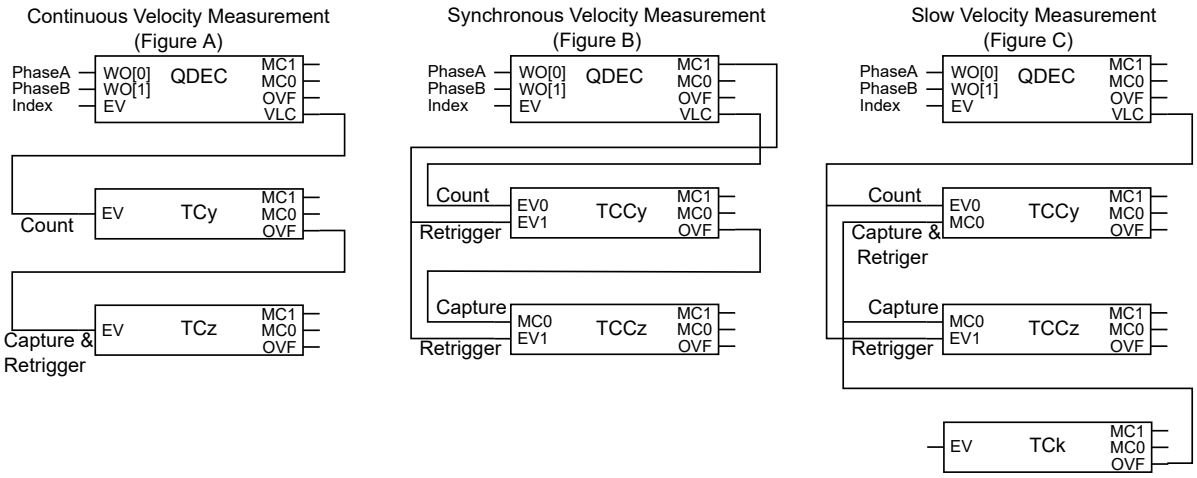
Velocity output event is generated on each QDEC transition except when the direction changes.

53.6.2.6.3 Speed Measurement

Three types of speed measurement can be done using velocity event output (VLC) and Timer/Counter (TC/TCC) device resources.

- Continuous velocity measurement: TCz measures the time on which n VLC (TCy) output events occur
- Synchronous Velocity measurement: On a specific motor position TCCz, the time is measured on which n VLC (TCCy) output events occur.
- Slow Velocity measurement: measure the number of VLC output events (TCCy) plus the delay since the last VLC output event (TCCz) within a given time slot (TCk).

Figure 53-7. Speed Measurement



54.12.4 Digital Frequency Locked Loop (DFLL48M) Characteristics

Table 54-46. DFLL48M Characteristics - Open Loop Mode

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$F_{OpenOUT}$	Output frequency	DFLLVAL after Reset LDO Regulator mode, [-40, 85]°C	45.8	48	49.3	MHz
		DFLLVAL after Reset LDO Regulator mode, [0, 60]°C	47.2	48	48.81	
$T_{OpenSTARTUP}$	Startup time	DFLLVAL after Reset F_{OUT} within 90% of final value	-	4.3	7	µs

Note: DFLL48 in open loop can be used only with LDO regulator.

Table 54-47. DFLL48M Characteristics - Closed Loop Mode

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$F_{CloseOUT}$	Average Output frequency	f_{REF} = XTAL, 32.768 kHz, 100 ppm DFLLMUL = 1464	-	47.972	-	MHz
F_{REF}(1,2)	Input reference frequency	-	732	32768	33000	Hz
$F_{CloseJitter}$	Period Jitter	f_{REF} = XTAL, 32.768 kHz, 100 ppm DFLLMUL = 1464	-	-	0.42	ns
T_{Lock}	Lock time	F_{REF} = XTAL, 32.768 kHz, 100 ppm DFLLMUL = 1464 DFLLVAL after Reset DFLLCTRL.BPLCKC = 1 DFLLCTRL.QLDIS = 0 DFLLCTRL.CCDIS = 1 DFLLMUL.FSTEP = 10	-	429	1145	µs

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. To ensure that the device stays within the maximum allowed clock frequency, any reference clock for the DFLL in close loop must be within 2% error accuracy.

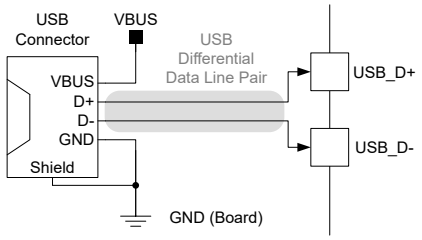
Table 54-48. DFLL48M Power Consumption

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Current Consumption	Open Loop mode - DFLLVAL after reset VCC = 3.3V	Max. 85°C Typ. 25°C	-	400	854	µA
		Closed Loop mode - f_{REF} = 32 .768 kHz VCC = 3.3V		-	404	851	µA

Table 56-11. USB Interface Checklist

Signal Name	Recommended Pin Connection	Description
D+	• The impedance of the pair should be matched on the PCB to minimize reflections. • USB differential tracks should be routed with the same characteristics (length, width, number of vias, etc.) • For a tightly coupled differential pair,the signal routing should be as parallel as possible, with a minimum number of angles and vias.	USB full speed / low speed positive data upstream pin
D-		USB full speed / low speed negative data upstream pin

Figure 56-17. Low Cost USB Interface Example Schematic



It is recommended to increase ESD protection on the USB D+, D-, and VBUS lines using dedicated transient suppressors. These protections should be located as close as possible to the USB connector to reduce the potential discharge path and reduce discharge propagation within the entire system.

The USB FS cable includes a dedicated shield wire that should be connected to the board with caution. Special attention should be paid to the connection between the board ground plane and the shield from the USB connector and the cable.

Tying the shield directly to ground would create a direct path from the ground plane to the shield, turning the USB cable into an antenna. To limit the USB cable antenna effect, it is recommended to connect the shield and ground through an RC filter.

Figure 56-18. Protected USB Interface Example Schematic