



Welcome to [E-XFL.COM](#)

### What is "[Embedded - Microcontrollers](#)"?

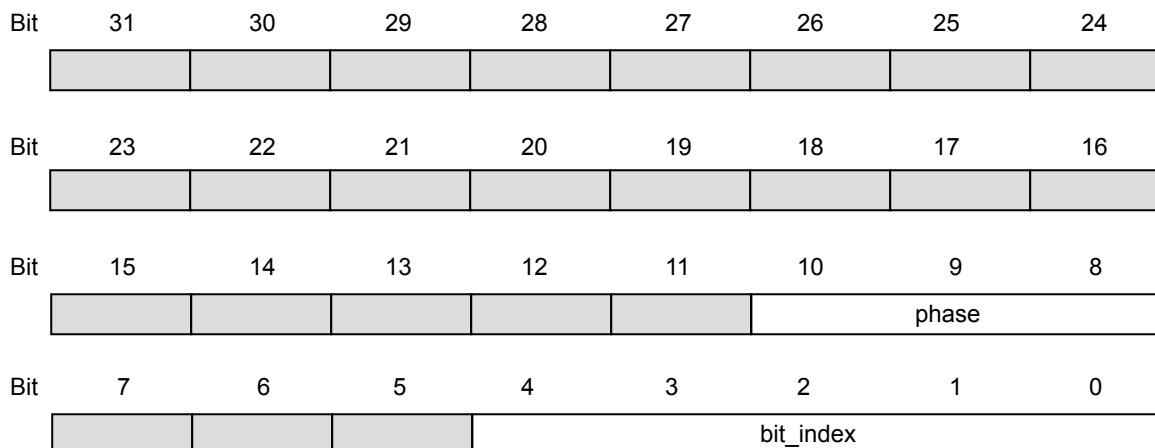
"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	I <sup>2</sup> C, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I <sup>2</sup> S, POR, PWM, WDT
Number of I/O	26
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 10x12b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-TQFP
Supplier Device Package	32-TQFP (7x7)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/atsamd21e15b-af">https://www.e-xfl.com/product-detail/microchip-technology/atsamd21e15b-af</a>

**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.

## 32-bit ARM-Based Microcontrollers

Value	Name	Description
0x00	XOSC	XOSC oscillator output
0x01	GCLKIN	Generator input pad
0x02	GCLKGEN1	Generic clock generator 1 output
0x03	OSCULP32K	OSCULP32K oscillator output
0x04	OSC32K	OSC32K oscillator output
0x05	XOSC32KReserved	XOSC32K oscillator output
0x06	OSC8M	OSC8M oscillator output
0x07	DFLL48M	DFLL48M output
0x08	FDPLL96M	FDPLL96M output
0x09-0x1F	Reserved	Reserved for future use

### Bits 3:0 – ID[3:0]: Generic Clock Generator Selection

These bits select the generic clock generator that will be configured or read. The value of the ID bit group versus which generic clock generator is configured is shown in the next table.

A power reset will reset the GENCTRL register for all IDs, including the generic clock generator used by the RTC. If a generic clock generator ID other than generic clock generator 0 is not a source of a “locked” generic clock or a source of the RTC generic clock, a user reset will reset the GENCTRL for this ID.

After a power reset, the reset value of the GENCTRL register is as shown in the next table.

GCLK Generator ID	Reset Value after a Power Reset
0x00	0x00010600
0x01	0x00000001
0x02	0x00010302
0x03	0x00000003
0x04	0x00000004
0x05	0x00000005
0x06	0x00000006
0x07	0x00000007
0x08	0x00000008

After a user reset, the reset value of the GENCTRL register is as shown in the table below.

GCLK Generator ID	Reset Value after a User Reset
0x00	0x00010600
0x01	0x00000001 if the generator is not used by the RTC and not a source of a 'locked' generic clock No change if the generator is used by the RTC or used by a GCLK with a WRTLOCK as one

## 16.6.2.5 Clock Ready Flag

There is a slight delay from when CPUSEL and APBxSEL are written until the new clock setting becomes effective. During this interval, the Clock Ready flag in the Interrupt Flag Status and Clear register (INTFLAG.CKRDY) will read as zero. If CKRDY in the INTENSET register is written to one, the Power Manager interrupt can be triggered when the new clock setting is effective. CPUSEL must not be re-written while CKRDY is zero, or the system may become unstable or hang.

## 16.6.2.6 Peripheral Clock Masking

It is possible to disable or enable the clock for a peripheral in the AHB or APBx clock domain by writing the corresponding bit in the Clock Mask register (APBxMASK) to zero or one. Refer to the table below for the default state of each of the peripheral clocks.

**Table 16-1. Peripheral Clock Default State**

Peripheral Clock	Default State
CLK_PAC0_APB	Enabled
CLK_PM_APB	Enabled
CLK_SYSCTRL_APB	Enabled
CLK_GCLK_APB	Enabled
CLK_WDT_APB	Enabled
CLK_RTC_APB	Enabled
CLK_EIC_APB	Enabled
CLK_PAC1_APB	Enabled
CLK_DSU_APB	Enabled
CLK_NVMCTRL_APB	Enabled
CLK_PORT_APB	Enabled
CLK_HMATRIX_APB	Enabled
CLK_PAC2_APB	Disabled
CLK_SERCOMx_APB	Disabled
CLK_TCx_APB	Disabled
CLK_ADC_APB	Enabled
CLK_AC_APB	Disabled
CLK_DAC_APB	Disabled
CLK_PTC_APB	Disabled
CLK_USB_APB	Enabled
CLK_DMAC_APB	Enabled
CLK_TCC_APB	Disabled
CLK_I2S_APB	Disabled

## 16.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0								
0x01	SLEEP	7:0							IDLE[1:0]	
0x02	Reserved									
...										
0x07										
0x08	CPUSEL	7:0							CPUDIV[2:0]	
0x09	APBASEL	7:0							APBADIV[2:0]	
0x0A	APBBSEL	7:0							APBBDIV[2:0]	
0x0B	APBCSEL	7:0							APBCDIV[2:0]	
0x0C	Reserved									
...										
0x13										
0x14	AHBMASK	7:0		USB	DMAC	NVMCTRL	DSU	HPB2	HPB1	HPB0
0x15		15:8								
0x16		23:16								
0x17		31:24								
0x18	APBAMASK	7:0		EIC	RTC	WDT	GCLK	SYSCTRL	PM	PAC0
0x19		15:8								
0x1A		23:16								
0x1B		31:24								
0x1C	APBBMASK	7:0			USB	DMAC	PORT	NVMCTRL	DSU	PAC1
0x1D		15:8								
0x1E		23:16								
0x1F		31:24								
0x20	APBCMASK	7:0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS	PAC2
0x21		15:8	TC7	TC6	TC5	TC4	TC3	TCC2	TCC1	TCC0
0x22		23:16				I2S	PTC	DAC	AC	ADC
0x23		31:24								
0x24	Reserved									
...										
0x33										
0x34	INTENCLR	7:0								CKRDY
0x35	INTENSET	7:0								CKRDY
0x36	INTFLAG	7:0								CKRDY
0x37	Reserved									
0x38	RCAUSE	7:0		SYST	WDT	EXT		BOD33	BOD12	POR

## 16.8 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.

Exception for APBASEL, APBBSEL and APBCSEL: These registers must only be accessed with 8-bit access.

## 17.6.9.3 Sampling Mode

The sampling mode is a low-power mode where the BOD33 or BOD12 is being repeatedly enabled on a sampling clock's ticks. The BOD33 or BOD12 will monitor the supply voltage for a short period of time and then go to a low-power disabled state until the next sampling clock tick.

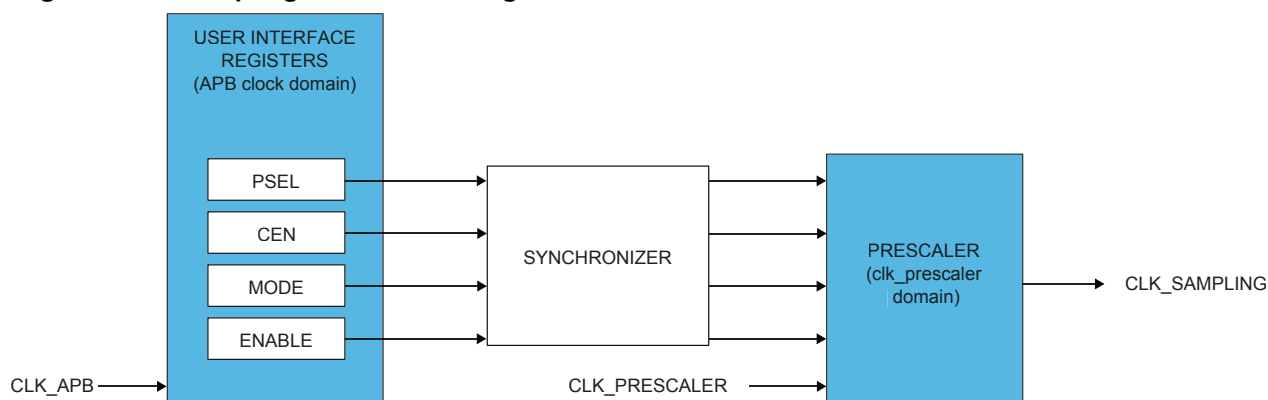
Sampling mode is enabled by writing one to BOD33.MODE for BOD33, and by writing one to BOD12.MODE for BOD12. The frequency of the clock ticks ( $F_{\text{clk\_sampling}}$ ) is controlled by the BOD33 Prescaler Select bit group (BOD33.PSEL) in the BOD33 register and Prescaler Select bit group (BOD12.PSEL) in the BOD12 register for BOD33 and BOD12, respectively.

$$F_{\text{clk\_sampling}} = \frac{F_{\text{clk\_prescaler}}}{2^{(\text{PSEL}+1)}}$$

The prescaler signal ( $F_{\text{clk\_prescaler}}$ ) is a 1kHz clock, output from the 32kHz Ultra Low Power Oscillator, OSCULP32K.

As the sampling mode clock is different from the APB clock domain, synchronization among the clocks is necessary. The next figure shows a block diagram of the sampling mode. The BOD33 and BOD12 Synchronization Ready bits (PCLKSR.B33SRDY and PCLKSR.B12SRDY, respectively) in the Power and Clocks Status register show the synchronization ready status of the synchronizer. Writing attempts to the BOD33 register are ignored while PCLKSR.B33SRDY is zero. Writing attempts to the BOD12 register are ignored while PCLKSR.B12SRDY is zero.

**Figure 17-7. Sampling Mode Block diagram**



The BOD33 Clock Enable bit (BOD33.CEN) in the BOD33 register and the BOD12 Clock Enable bit (BOD12.CEN) in the BOD12 register should always be disabled before changing the prescaler value. To change the prescaler value for the BOD33 or BOD12 during sampling mode, the following steps need to be taken:

1. Wait until the PCLKSR.B33SRDY bit or the PCLKSR.B12SRDY bit is set.
2. Write the selected value to the BOD33.PSEL or BOD12.PSEL bit group.

## 17.6.9.4 Hysteresis

The hysteresis functionality can be used in both continuous and sampling mode. Writing a one to the BOD33 Hysteresis bit (BOD33.HYST) in the BOD33 register will add hysteresis to the BOD33 threshold level. Writing a one to the BOD12 Hysteresis bit (BOD12.HYST) in the BOD12 register will add hysteresis to the BOD12 threshold level.

## 17.6.10 Voltage Reference System Operation

The Voltage Reference System (VREF) consists of a Bandgap Reference Voltage Generator and a temperature sensor.

Value	Description
0	Output clock before the DFLL is locked.
1	Output clock when DFLL is locked.

## Bit 10 – BPLCKC: Bypass Coarse Lock

This bit controls the coarse lock procedure:

Value	Description
0	Bypass coarse lock is disabled.
1	Bypass coarse lock is enabled.

## Bit 9 – QLDIS: Quick Lock Disable

Value	Description
0	Quick Lock is enabled.
1	Quick Lock is disabled.

## Bit 8 – CCDIS: Chill Cycle Disable

Value	Description
0	Chill Cycle is enabled.
1	Chill Cycle is disabled.

## Bit 7 – ONDEMAND: On Demand Control

The On Demand operation mode allows an oscillator to be enabled or disabled depending on peripheral clock requests.

In On Demand operation mode, i.e., if the ONDEMAND bit has been previously written to one, the oscillator will only be running when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.

If On Demand is disabled the oscillator will always be running when enabled.

In standby sleep mode, the On Demand operation is still active if the DFLLCTRL.RUNSTDBY bit is one. If DFLLCTRL.RUNSTDBY is zero, the oscillator is disabled.

Value	Description
0	The oscillator is always on, if enabled.
1	The oscillator is enabled when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is disabled if no peripheral is requesting the clock source.

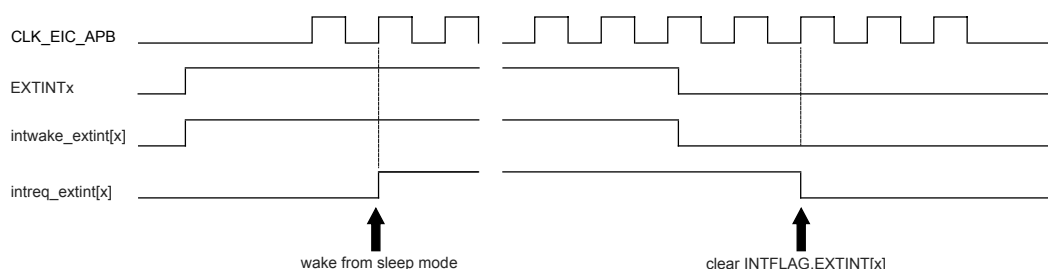
## Bit 6 – RUNSTDBY: Run in Standby

This bit controls how the DFLL behaves during standby sleep mode:

Value	Description
0	The oscillator is disabled in standby sleep mode.
1	The oscillator is not stopped in standby sleep mode. If DFLLCTRL.ONDEMAND is one, the clock source will be running when a peripheral is requesting the clock. If DFLLCTRL.ONDEMAND is zero, the clock source will always be running in standby sleep mode.

## Bit 5 – USBCRM: USB Clock Recovery Mode

**Figure 21-3. Wake-Up Operation Example (High-Level Detection, No Filter, WAKEUPEN[x]=1)**



## 21.6.9 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

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 STATUS.SYNCBUSY is one, the bus will be stalled. All operations will complete successfully, but the CPU will be stalled, and interrupts will be pending as long as the bus is stalled.

The following bits are synchronized when written:

- Software Reset bit in the Control register (CTRL.SWRST)
- Enable bit in the Control register (CTRL.ENABLE)

### Related Links

[Register Synchronization](#)

## 21.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0							ENABLE	SWRST
0x01	STATUS	7:0	SYNCBUSY							
0x02	NMICTRL	7:0					NMIFILTEN		NMISENSE[2:0]	
0x03	NMIFLAG	7:0								NMI
0x04	EVCTRL	7:0	EXTINTEO7	EXTINTEO6	EXTINTEO5	EXTINTEO4	EXTINTEO3	EXTINTEO2	EXTINTEO1	EXTINTEO0
0x05		15:8	EXTINTEO15	EXTINTEO14	EXTINTEO13	EXTINTEO12	EXTINTEO11	EXTINTEO10	EXTINTEO9	EXTINTEO8
0x06		23:16							EXTINTEO17	EXTINTEO16
0x07		31:24								
0x08	INTENCLR	7:0	EXTINT7	EXTINT6	EXTINT5	EXTINT4	EXTINT3	EXTINT2	EXTINT1	EXTINT0
0x09		15:8	EXTINT15	EXTINT14	EXTINT13	EXTINT12	EXTINT11	EXTINT10	EXTINT9	EXTINT8
0x0A		23:16							EXTINT17	EXTINT16
0x0B		31:24								
0x0C	INTENSET	7:0	EXTINT7	EXTINT6	EXTINT5	EXTINT4	EXTINT3	EXTINT2	EXTINT1	EXTINT0
0x0D		15:8	EXTINT15	EXTINT14	EXTINT13	EXTINT12	EXTINT11	EXTINT10	EXTINT9	EXTINT8
0x0E		23:16							EXTINT17	EXTINT16
0x0F		31:24								



**Note:** Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

## Related Links

[PAC - Peripheral Access Controller](#)

### 23.5.9 Analog Connections

Analog functions are connected directly between the analog blocks and the I/O pads using analog buses. However, selecting an analog peripheral function for a given pin will disable the corresponding digital features of the pad.

### 23.5.10 CPU Local Bus

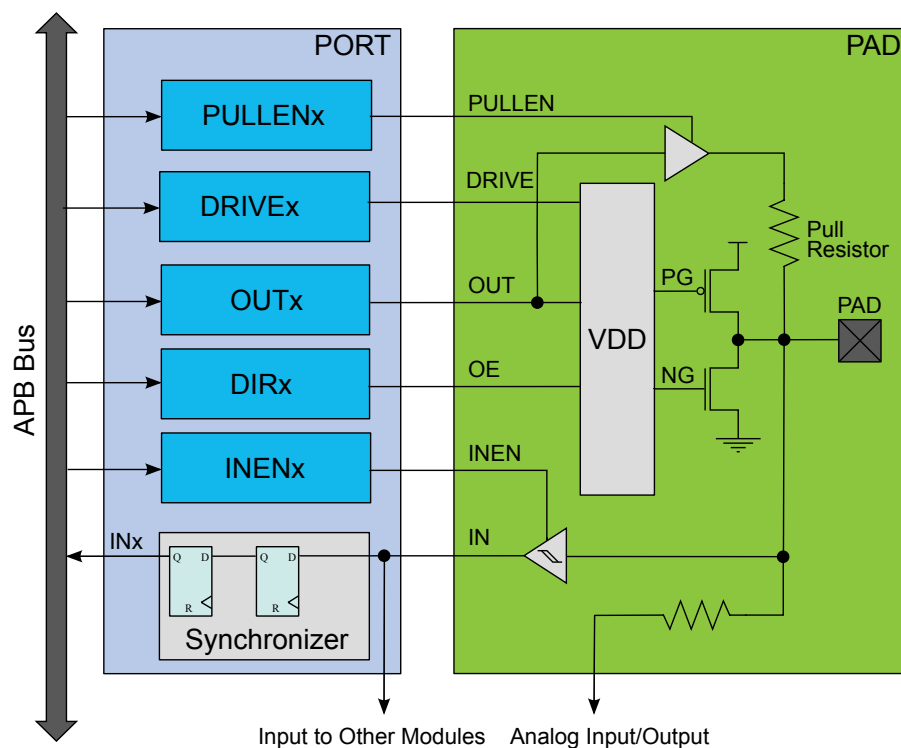
The CPU local bus (IOBUS) is an interface that connects the CPU directly to the PORT. It is a single-cycle bus interface, which does not support wait states. It supports 8-bit, 16-bit and 32-bit sizes.

This bus is generally used for low latency operation. The Data Direction (DIR) and Data Output Value (OUT) registers can be read, written, set, cleared or be toggled using this bus, and the Data Input Value (IN) registers can be read.

Since the IOBUS cannot wait for IN register resynchronization, the Control register (CTRL) must be configured to continuous sampling of all pins that need to be read via the IOBUS in order to prevent stale data from being read.

## 23.6 Functional Description

Figure 23-2. Overview of the PORT



## 32-bit ARM-Based Microcontrollers

Bit	23	22	21	20	19	18	17	16
		EVGEN[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

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

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

### Bits 27:26 – EDGSEL[1:0]: Edge Detection Selection

These bits set the type of edge detection to be used on the channel.

These bits must be written to zero when using the asynchronous path.

EDGSEL[1:0]	Name	Description
0x0	NO_EVT_OUTPUT	No event output when using the resynchronized or synchronous path
0x1	RISING_EDGE	Event detection only on the rising edge of the signal from the event generator when using the resynchronized or synchronous path
0x2	FALLING_EDGE	Event detection only on the falling edge of the signal from the event generator when using the resynchronized or synchronous path
0x3	BOTH_EDGES	Event detection on rising and falling edges of the signal from the event generator when using the resynchronized or synchronous path

### Bits 25:24 – PATH[1:0]: Path Selection

These bits are used to choose the path to be used by the selected channel.

The path choice can be limited by the channel source.

PATH[1:0]	Name	Description
0x0	SYNCHRONOUS	Synchronous path
0x1	RESYNCHRONIZED	Resynchronized path
0x2	ASYNCHRONOUS	Asynchronous path
0x3		Reserved

### Bits 22:16 – EVGEN[6:0]: Event Generator Selection

These bits are used to choose which event generator to connect to the selected channel.

Value	Description
0	The transmitter is disabled or being enabled.
1	The transmitter is enabled or will be enabled when the USART is enabled.

## Bit 13 – PMODE: Parity Mode

This bit selects the type of parity used when parity is enabled (CTRLA.FORM is '1'). The transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The receiver will generate a parity value for the incoming data and parity bit, compare it to the parity mode and, if a mismatch is detected, STATUS.PERR will be set.

This bit is not synchronized.

Value	Description
0	Even parity.
1	Odd parity.

## Bit 10 – ENC: Encoding Format

This bit selects the data encoding format.

This bit is not synchronized.

Value	Description
0	Data is not encoded.
1	Data is IrDA encoded.

## Bit 9 – SFDE: Start of Frame Detection Enable

This bit controls whether the start-of-frame detector will wake up the device when a start bit is detected on the RxD line.

This bit is not synchronized.

SFDE	INTENSET.RXS	INTENSET.RXC	Description
0	X	X	Start-of-frame detection disabled.
1	0	0	Reserved
1	0	1	Start-of-frame detection enabled. RXC wakes up the device from all sleep modes.
1	1	0	Start-of-frame detection enabled. RXS wakes up the device from all sleep modes.
1	1	1	Start-of-frame detection enabled. Both RXC and RXS wake up the device from all sleep modes.

## Bit 8 – COLDEN: Collision Detection Enable

This bit enables collision detection.

This bit is not synchronized.

Value	Description
0	Collision detection is not enabled.
1	Collision detection is enabled.

arbitration is lost during the transmission. In this case, a lost arbitration will prevent setting INTFLAG.SB. Instead, INTFLAG.MB will indicate a change in arbitration. Handling of lost arbitration is the same as for data bit transmission.

## Receiving Data Packets (SCLSM=1)

When INTFLAG.SB is set, the I<sup>2</sup>C master will already have received one data packet and transmitted an ACK or NACK, depending on CTRLB.ACKACT. At this point, CTRLB.ACKACT must be set to the correct value for the next ACK bit, and the transaction can continue by reading DATA and issuing a command if not in the smart mode.

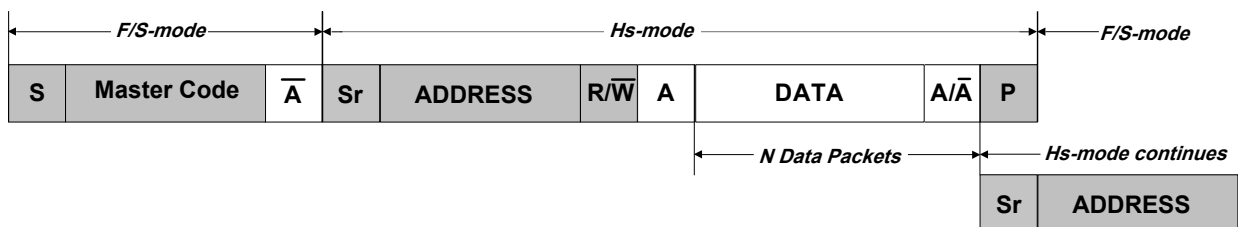
## High-Speed Mode

High-speed transfers are a multi-step process, see [High Speed Transfer](#).

First, a master code (0b00001nnn, where 'nnn' is a unique master code) is transmitted in Full-speed mode, followed by a NACK since no slave should acknowledge. Arbitration is performed only during the Full-speed Master Code phase. The master code is transmitted by writing the master code to the address register (ADDR.ADDR) and writing the high-speed bit (ADDR.HS) to '0'.

After the master code and NACK have been transmitted, the master write interrupt will be asserted. In the meanwhile, the slave address can be written to the ADDR.ADDR register together with ADDR.HS=1. Now in High-speed mode, the master will generate a repeated start, followed by the slave address with RW-direction. The bus will remain in High-speed mode until a stop is generated. If a repeated start is desired, the ADDR.HS bit must again be written to '1', along with the new address ADDR.ADDR to be transmitted.

**Figure 28-7. High Speed Transfer**



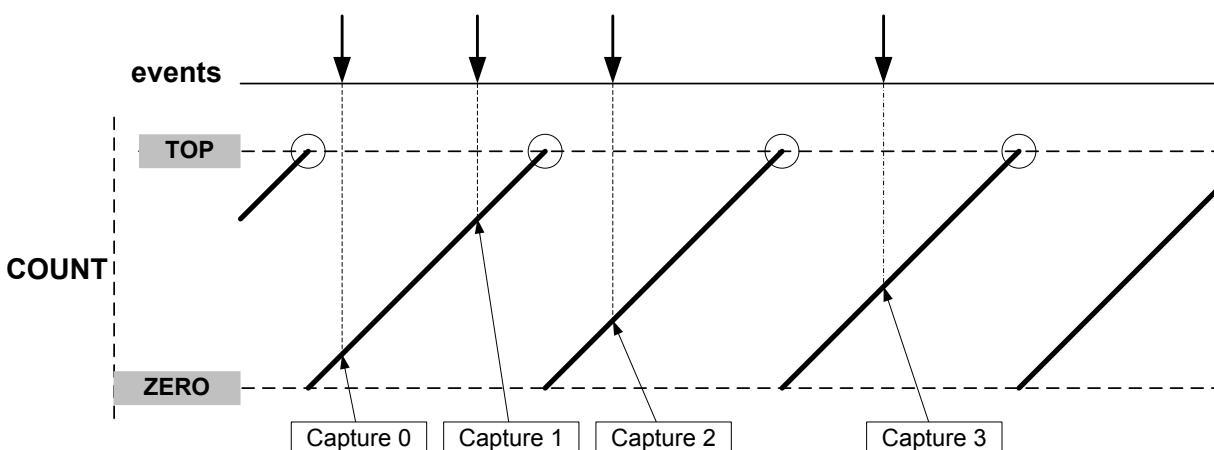
Transmitting in High-speed mode requires the I<sup>2</sup>C master to be configured in High-speed mode (CTRLA.SPEED=0x2) and the SCL clock stretch mode (CTRLA.SCLSM) bit set to '1'.

## 10-Bit Addressing

When 10-bit addressing is enabled by the Ten Bit Addressing Enable bit in the Address register (ADDR.TENBITEN=1) and the Address bit field ADDR.ADDR is written, the two address bytes will be transmitted, see [10-bit Address Transmission for a Read Transaction](#). The addressed slave acknowledges the two address bytes, and the transaction continues. Regardless of whether the transaction is a read or write, the master must start by sending the 10-bit address with the direction bit (ADDR.ADDR[0]) being zero.

If the master receives a NACK after the first byte, the write interrupt flag will be raised and the STATUS.RXNACK bit will be set. If the first byte is acknowledged by one or more slaves, then the master will proceed to transmit the second address byte and the master will first see the write interrupt flag after the second byte is transmitted. If the transaction direction is read-from-slave, the 10-bit address transmission must be followed by a repeated start and the first 7 bits of the address with the read/write bit equal to '1'.

Figure 30-9. Input Capture Timing



The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

## Period and Pulse-Width (PPW) Capture Action

The TC can perform two input captures and restart the counter on one of the edges. This enables the TC to measure the pulse width and period and to characterize the frequency  $f$  and duty cycle of an input signal:

$$f = \frac{1}{T}$$

$$\text{dutyCycle} = \frac{t_p}{T}$$

Selecting PWP (pulse-width, period) in the Event Action bit group in the Event Control register (EVCTRL.EVACT) enables the TC to perform one capture action on the rising edge and the other one on the falling edge. The period  $T$  will be captured into CC1 and the pulse width  $t_p$  in CC0.

EVCTRL.EVACT=PPW (period and pulse-width) offers identical functionality, but will capture  $T$  into CC0 and  $t_p$  into CC1.

The TC Event Input Invert Enable bit in the Event Control register (EVCTRL.TCINV) is used to select whether the wraparound should occur on the rising edge or the falling edge. If EVCTRL.TCINV=1, the wraparound will happen on the falling edge.

To fully characterize the frequency and duty cycle of the input signal, activate capture on CC0 and CC1 by writing 0x3 to the Capture Channel x Enable bit group in the Control C register (CTRLC.CPTEN). When only one of these measurements is required, the second channel can be used for other purposes.

The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the TC is reset. on how to clear interrupt flags.

The TC has one common interrupt request line for all the interrupt sources. 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. Refer to *Nested Vector Interrupt Controller* for details.

### Related Links

[Nested Vector Interrupt Controller](#)

### 30.6.4.3 Events

The TC can generate the following output events:

- Overflow/Underflow (OVF)
- Match or Capture (MC)

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.MCEOx) enables the corresponding output event. The output event is disabled by writing EVCTRL.MCEOx=0.

One of the following event actions can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT):

- Start TC (START)
- Re-trigger TC (RETRIGGER)
- Increment or decrement counter (depends on counter direction)
- Count on event (COUNT)
- Capture Period (PPW and PWP)
- Capture Pulse Width (PW)

Writing a '1' to the TC Event Input bit in the Event Control register (EVCTRL.TCEI) enables input events to the TC. Writing a '0' to this bit disables input events to the TC. The TC requires only asynchronous event inputs. For further details on how configuring the asynchronous events, refer to *EVSYS - Event System*.

### Related Links

[EVSYS – Event System](#)

### 30.6.5 Sleep Mode Operation

The TC can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be written to one. The TC can wake up the device using interrupts from any sleep mode or perform actions through the Event System.

### 30.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the Control A register (CTRLA.SWRST)

## 32-bit ARM-Based Microcontrollers

Bit	15	14	13	12	11	10	9	8
					DTIEN3	DTIEN2	DTIEN1	DTIEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
							OTMX[1:0]	
Access							R/W	R/W
Reset							0	0

### Bits 31:24 – DTHTS[7:0]: Dead-Time High Side Outputs Value

This register holds the number of GCLK\_TCC clock cycles for the dead-time high side.

### Bits 23:16 – DTLS[7:0]: Dead-time Low Side Outputs Value

This register holds the number of GCLK\_TCC clock cycles for the dead-time low side.

### Bits 11,10,9,8 – DTIENx : Dead-time Insertion Generator x Enable

Setting any of these bits enables the dead-time insertion generator for the corresponding output matrix. This will override the output matrix [x] and [x+WO\_NUM/2], with the low side and high side waveform respectively.

Value	Description
0	No dead-time insertion override.
1	Dead time insertion override on signal outputs[x] and [x+WO_NUM/2], from matrix outputs[x] signal.

### Bits 1:0 – OTMX[1:0]: Output Matrix

These bits define the matrix routing of the TCC waveform generation outputs to the port pins, according to [Table 31-4](#).

## 31.8.7 Driver Control

**Name:** DRVCTRL

**Offset:** 0x18

**Reset:** 0x00000000

**Property:** PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	FILTERVAL1[3:0]				FILTERVAL0[3:0]			
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
Bit	23	22	21	20	19	18	17	16
	INVEN7	INVEN6	INVEN5	INVEN4	INVEN3	INVEN2	INVEN1	INVEN0
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 6 – BK0RDY: Bank 0 is ready

Writing a one to the bit EPSTATUSCLR.BK0RDY will clear this bit.

Writing a one to the bit EPSTATUSSET.BK0RDY will set this bit.

Value	Description
0	The bank number 0 is not ready : For IN direction Endpoints, the bank is not yet filled in. For Control/OUT direction Endpoints, the bank is empty.
1	The bank number 0 is ready: For IN direction Endpoints, the bank is filled in. For Control/OUT direction Endpoints, the bank is full.

## Bit 4 – STALLRQ: STALL bank x request

Writing a zero to the bit EPSTATUSCLR.STALLRQ will clear this bit.

Writing a one to the bit EPSTATUSSET.STALLRQ will set this bit.

This bit is cleared by hardware when receiving a SETUP packet.

Value	Description
0	Disable STALLRQx feature.
1	Enable STALLRQx feature: a STALL handshake will be sent to the host in regards to bank x.

## Bit 2 – CURBK: Current Bank

Writing a zero to the bit EPSTATUSCLR.CURBK will clear this bit.

Writing a one to the bit EPSTATUSSET.CURBK will set this bit.

Value	Description
0	The bank0 is the bank that will be used in the next single/multi USB packet.
1	The bank1 is the bank that will be used in the next single/multi USB packet.

## Bit 1 – DTGLIN: Data Toggle IN Sequence

Writing a zero to the bit EPSTATUSCLR.DTGLINCLR will clear this bit.

Writing a one to the bit EPSTATUSSET.DTGLINSET will set this bit.

Value	Description
0	The PID of the next expected IN transaction will be zero: data 0.
1	The PID of the next expected IN transaction will be one: data 1.

## Bit 0 – DTGLOUT: Data Toggle OUT Sequence

Writing a zero to the bit EPSTATUSCLR.DTGLOUTCLR will clear this bit.

Writing a one to the bit EPSTATUSSET.DTGLOUTSET will set this bit.

Value	Description
0	The PID of the next expected OUT transaction will be zero: data 0.
1	The PID of the next expected OUR transaction will be one: data 1.

### 32.8.3.5 Device EndPoint Interrupt Flag n

**Name:** EPINTFLAGn

**Offset:** 0x107 + (n x 0x20)

**Reset:** 0x00

**Property:** -



This bit is also set by the hardware:

- When a STALL handshake has been received.
- After a PIPE has been enabled (rising of bit PEN.N).
- When an LPM transaction has completed whatever handshake is returned or the transaction was timed-out.
- When a pipe transfer was completed with a pipe error. See [PINTFLAG](#) register.

When PFREEZE bit is set while a transaction is in progress on the USB bus, this transaction will be properly completed. PFREEZE bit will be read as “1” only when the ongoing transaction will have been completed.

Value	Description
0	The Pipe operates in normal operation.
1	The Pipe is frozen and no additional requests will be sent to the device on this pipe address.

## Bit 2 – CURBK: Current Bank

Value	Description
0	The bank0 is the bank that will be used in the next single/multi USB packet.
1	The bank1 is the bank that will be used in the next single/multi USB packet.

## Bit 0 – DTGL: Data Toggle Sequence

Writing a one to the bit EPSTATUSCLR.DTGL will clear this bit.

Writing a one to the bit EPSTATUSSET.DTGL will set this bit.

This bit is toggled automatically by hardware after a data transaction.

This bit will reflect the data toggle in regards of the token type (IN/OUT/SETUP).

Value	Description
0	The PID of the next expected transaction will be zero: data 0.
1	The PID of the next expected transaction will be one: data 1.

## 32.8.6.6 Host Pipe Interrupt Flag Register

**Name:** PINTFLAG

**Offset:** 0x107 + (n x 0x20)

**Reset:** 0x00

**Property:** -

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL		TRCPT
Access			R/W	R/W	R/W	R/W		R/W
Reset			0	0	0	0		0

## Bit 5 – STALL: STALL Received Interrupt Flag

This flag is cleared by writing a one to the flag.

This flag is set when a stall occurs and will generate an interrupt if PINTENCLR/SET.STALL is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the STALL Interrupt Flag.

**Property:** NA

Bit	7	6	5	4	3	2	1	0
							ERRORFLOW	CRCERR
Access							R/W	R/W
Reset							x	x

## Bit 1 – ERRORFLOW: Error Flow Status

This bit defines the Error Flow Status.

This bit is set when a Error Flow has been detected during transfer from/towards this bank.

For IN transfer, a NAK handshake has been received. For OUT transfer, a NAK handshake has been received. For Isochronous IN transfer, an overrun condition has occurred. For Isochronous OUT transfer, an underflow condition has occurred.

Value	Description
0	No Error Flow detected.
1	A Error Flow has been detected.

## Bit 0 – CRCERR: CRC Error

This bit defines the CRC Error Status.

This bit is set when a CRC error has been detected in an isochronous IN endpoint bank.

Value	Description
0	No CRC Error.
1	CRC Error detected.

### 32.8.7.6 Host Control Pipe

**Name:** CTRL\_PIPE

**Offset:** 0x0C

**Reset:** 0xFFFF

**Property:** PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	PERMAX[3:0]				PEPNUM[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	x	0	0	0	x

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

## Bits 15:12 – PERMAX[3:0]: Pipe Error Max Number

These bits define the maximum number of error for this Pipe before freezing the pipe automatically.

## Bits 11:8 – PEPNUM[3:0]: Pipe EndPoint Number

These bits define the number of endpoint for this Pipe.

## 32-bit ARM-Based Microcontrollers

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$I_{DFLL}$	Power consumption on $V_{DDIN}$	$f_{REF} = XTAL, 32.768kHz, 100ppm$ $DFLLMUL = 1464$	-	425	482	$\mu A$
$t_{LOCK}$	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	$\mu s$

**Table 37-52. DFLL48M Characteristics - Closed Loop Mode<sup>(1)</sup> (Device Variant B and C)**

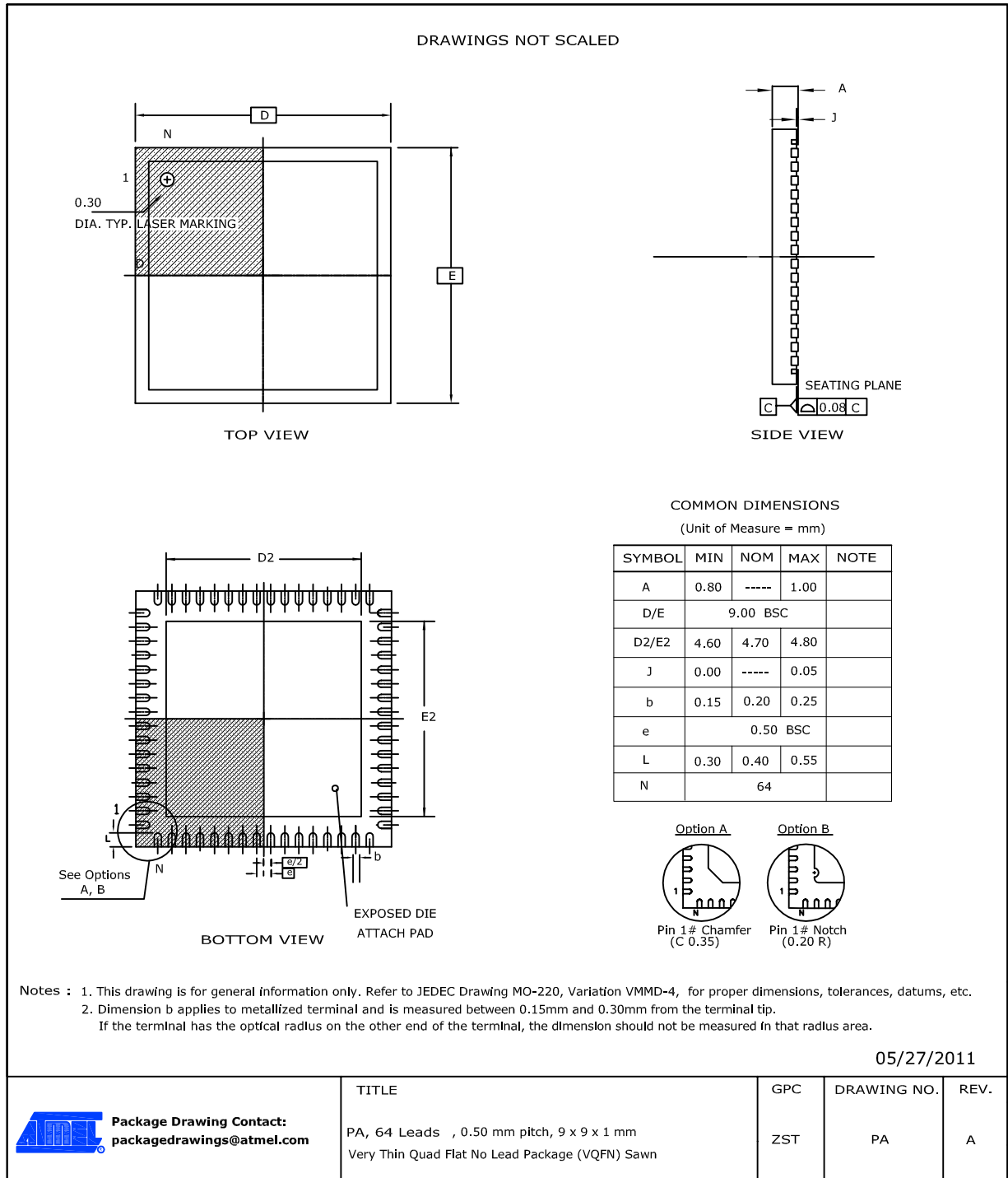
Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$f_{OUT}$	Average Output frequency	$f_{REF} = 32.768kHz$	47.963	47.972	47.981	MHz
$f_{REF}$	Reference frequency		0.732	32.768	33	kHz
Jitter	Cycle to Cycle jitter	$f_{REF} = 32.768kHz$	-	-	0.42	ns
$I_{DFLL}$	Power consumption on $V_{DDIN}$	$f_{REF} = 32.768kHz$	-	403	453	$\mu A$
$t_{LOCK}$	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	$\mu s$

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 38-4. Package Reference**

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

## 38.2.2 64 pin QFN



**Note:** The exposed die attach pad is not connected electrically inside the device.

Symbol	Description
GHz	$10^9 = 1,000,000,000\text{Hz}$
s	second
ms	millisecond
$\mu\text{s}$	microsecond
ns	nanosecond

## 41.4 Registers and Bits

**Table 41-3. Register and Bit Mnemonics**

Symbol	Description
R/W	Read/Write accessible register bit. The user can read from and write to this bit.
R	Read-only accessible register bit. The user can only read this bit. Writes will be ignored.
W	Write-only accessible register bit. The user can only write this bit. Reading this bit will return an undefined value.
BIT	Bit names are shown in uppercase. (Example ENABLE)
FIELD[n:m]	A set of bits from bit n down to m. (Example: PINA[3:0] = {PINA3, PINA2, PINA1, PINA0})
Reserved	Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to zero when the register is written. Reserved bits will always return zero when read.
PERIPHERAL <i>i</i>	If several instances of a peripheral exist, the peripheral name is followed by a number to indicate the number of the instance in the range 0-n. PERIPHERAL0 denotes one specific instance.
Reset	Value of a register after a power reset. This is also the value of registers in a peripheral after performing a software reset of the peripheral, except for the Debug Control registers.
SET/CLR	Registers with SET/CLR suffix allows the user to clear and set bits in a register without doing a read-modify-write operation. These registers always come in pairs. Writing a one to a bit in the CLR register will clear the corresponding bit in both registers, while writing a one to a bit in the SET register will set the corresponding bit in both registers. Both registers will return the same value when read. If both registers are written simultaneously, the write to the CLR register will take precedence.