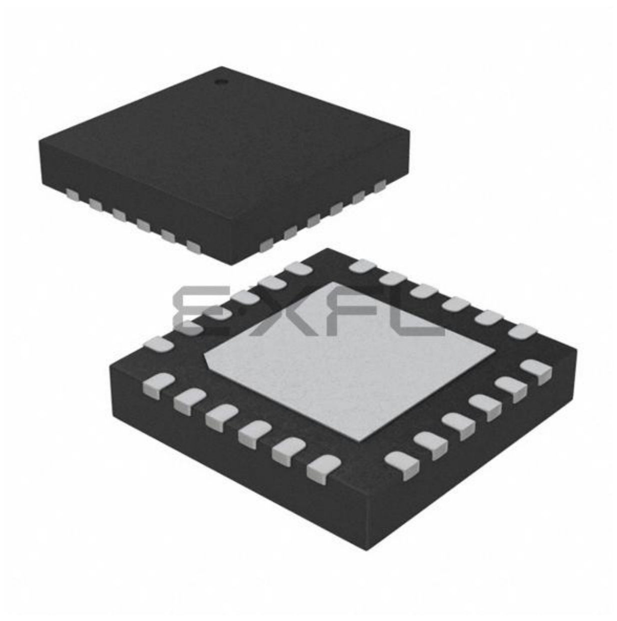


Welcome to E-XFL.COM

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	AVR
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 24x10b; D/A 3x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	24-VFQFN Exposed Pad
Supplier Device Package	24-VQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny1617-mnr

5. I/O Multiplexing and Considerations

5.1 Multiplexed Signals

Table 5-1. PORT Function Multiplexing

VQFN 24-pin	Pin Name [1,2]	Other/Special	ADC0	ADC1	PTC[3]	AC0	AC1	AC2	DAC0	USART0	SPI0	TWI0	TCA0	TCBn	TCD0	CCL
23	PA0	$\overline{\text{RESET}}$ UPDI	AIN0													LUT0-IN0
24	PA1		AIN1							`TXD`	MOSI	`SDA`				LUT0-IN1
1	PA2	EVOUT0	AIN2							`RxD`	MISO	`SCL`				LUT0-IN2
2	PA3	EXTCLK	AIN3							`XCK`	SCK		WO3	TCB1 WO		
3	GND															
4	VDD															
5	PA4		AIN4	AIN0	X0/Y0					`XDIR`	$\overline{\text{SS}}$		WO4		WOA	LUT0-OUT
6	PA5	VREFA	AIN5	AIN1	X1/Y1	OUT	AINN0						WO5	TCB0 WO	WOB	
7	PA6		AIN6	AIN2	X2/Y2	AINN0	AINP1	AINP0	OUT							
8	PA7		AIN7	AIN3	X3/Y3	AINP0	AINP0	AINN0								LUT1-OUT
9	PB7			AIN4			AINN1	AINP3								
10	PB6			AIN5		AINP3		AINN1								
11	PB5	CLKOUT	AIN8		X12/Y12	AINP1		AINP2					`WO2`			
12	PB4		AIN9		X13/Y13	AINN1	AINP3						`WO1`			`LUT0-OUT`
13	PB3	TOSC1					OUT			RxD			`WO0`			
14	PB2	TOSC2, EVOUT1						OUT		TxD			WO2			
15	PB1		AIN10		X4/Y4	AINP2				XCK		SDA	WO1			
16	PB0		AIN11		X5/Y5		AINP2	AINP1		XDIR		SCL	WO0			
17	PC0			AIN6	X6/Y6						`SCK`			`TCB0 WO`	WOC	
18	PC1			AIN7	X7/Y7						`MISO`				WOD	`LUT1-OUT`
19	PC2	EVOUT2		AIN8	X9/Y8						`MOSI`					
20	PC3			AIN9	X9/Y9						$\overline{\texttt{SS}}$		`WO3`			LUT1-IN0
21	PC4			AIN10	X10/Y10								`WO4`	`TCB1 WO`		LUT1-IN1
22	PC5			AIN11	X11/Y11								`WO5`			LUT1-IN2

Note:

1. Pin names are of type P*xn*, with *x* being the PORT instance (A, B) and *n* the pin number. Notation for signals is PORT*x*_PIN*n*. All pins can be used as event input.
2. All pins can be used for external interrupt, where pins P*x*2 and P*x*6 of each port have full asynchronous detection.
3. Every PTC line can be configured as X- or Y-line.

Tip: Signals on alternative pin locations are in `typewriter` font.

between registers or between a register and an immediate. Six of the 32 registers can be used as three 16-bit Address Pointers for program and data space addressing, enabling efficient address calculations.

The program memory bus is connected to Flash, and the first program memory Flash address is 0x0000.

The data memory space is divided into I/O registers, SRAM, EEPROM, and Flash.

All I/O Status and Control registers reside in the lowest 4 KB addresses of the data memory. This is referred to as the I/O memory space. The lowest 64 addresses are accessed directly with single-cycle `IN`/`OUT` instructions, or as the data space locations from 0x00 to 0x3F. These addresses can be accessed using load (`LD`/`LDS`/`LDD`) and store (`ST`/`STS`/`STD`) instructions. The lowest 32 addresses can even be accessed with single-cycle `SBI`/`CBI` instructions and `SBIS`/`SBIC` instructions. The rest is the extended I/O memory space, ranging from 0x0040 to 0x0FFF. The I/O registers here must be accessed as data space locations using load and store instructions.

Data addresses 0x1000 to 0x1800 are reserved for memory mapping of fuses, the NVM controller and EEPROM. The addresses from 0x1800 to 0x7FFF are reserved for other memories, such as SRAM.

The Flash is mapped in the data space from 0x8000 and above. The Flash can be accessed with all load and store instructions by using addresses above 0x8000. The `LPM` instruction accesses the Flash similar to the code space, where the Flash starts at address 0x0000.

For a summary of all AVR instructions, refer to the Instruction Set Summary section. For details of all AVR instructions, refer to http://www.microchip.com/design-centers/8-bit.

Related Links

9. NVMCTRL - Nonvolatile Memory Controller
6. Memories
34. Instruction Set Summary

8.4 Arithmetic Logic Unit (ALU)

The Arithmetic Logic Unit (ALU) supports arithmetic and logic operations between registers, or between a constant and a register. Also, single-register operations can be executed.

The ALU operates in direct connection with all 32 general purpose registers. Arithmetic operations between general purpose registers or between a register and an immediate are executed in a single clock cycle, and the result is stored in the register file. After an arithmetic or logic operation, the Status register (CPU.SREG) is updated to reflect information about the result of the operation.

ALU operations are divided into three main categories – arithmetic, logical, and bit functions. Both 8- and 16-bit arithmetic are supported, and the instruction set allows for efficient implementation of 32-bit arithmetic. The hardware multiplier supports signed and unsigned multiplication and fractional format.

8.4.1 Hardware Multiplier

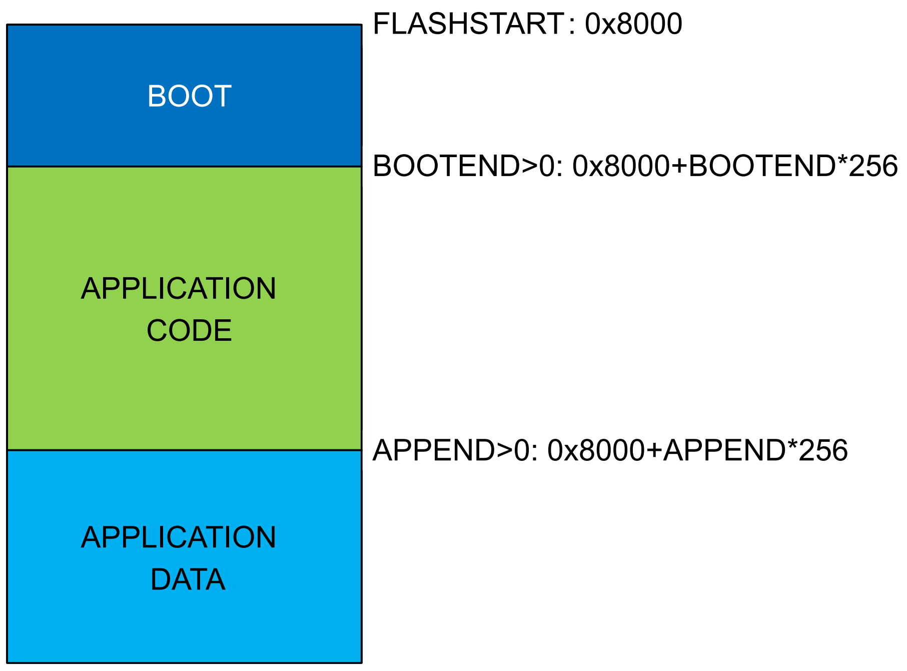
The multiplier is capable of multiplying two 8-bit numbers into a 16-bit result. The hardware multiplier supports different variations of signed and unsigned integer and fractional numbers:

- Multiplication of signed/unsigned integers
- Multiplication of signed/unsigned fractional numbers
- Multiplication of a signed integer with an unsigned integer
- Multiplication of a signed fractional number with an unsigned one

A multiplication takes two CPU clock cycles.

The Flash can be divided into three sections in blocks of 256 bytes for different security. The three different sections are BOOT, Application Code (APPCODE), and Application Data (APPDATA).

Figure 9-2. Flash Sections



Section Sizes
The sizes of these sections are set by the Boot Section End fuse (FUSE.BOOTEND) and Application Code Section End fuse (FUSE.APPEND).

The fuses select the section sizes in blocks of 256 bytes. The BOOT section stretches from the start of the Flash until BOOTEND. The APPCODE section runs from BOOTEND until APPEND. The remaining area is the APPDATA section. If APPEND is written to 0, the APPCODE section runs from BOOTEND to the end of Flash (removing the APPDATA section). If BOOTEND and APPEND are written to 0, the entire Flash is regarded as BOOT section. APPEND should either be set to 0 or a value greater or equal than BOOTEND.

Table 9-2. Setting Up Flash Sections

BOOTEND	APPEND	BOOT Section	APPCODE Section	APPDATA Section
0	0	0 to FLASHEND	-	-
> 0	0	0 to 256*BOOTEND	256*BOOTEND to FLASHEND	-
> 0	== BOOTEND	0 to 256*BOOTEND	-	256*BOOTEND to FLASHEND
> 0	> BOOTEND	0 to 256*BOOTEND	256*BOOTEND to 256*APPEND	256*APPEND to FLASHEND

Note:
- See also the BOOTEND and APPEND descriptions.

Bit 0 – ENABLE Enable
When this bit is written to '1', the configuration of the respective input pins is overridden to TOSC1 and TOSC2. Also, the Source Select bit (SEL) and Crystal Start-Up Time (CSUT) become read-only.

This bit is I/O protected to prevent unintentional enabling of the oscillator.

16. PORT - I/O Pin Configuration

16.1 Features

- General Purpose Input and Output Pins with Individual Configuration
- Output Driver with Configurable Inverted I/O and Pull-up
- Input with Interrupts and Events:
  - Sense both edges
  - Sense rising edges
  - Sense falling edges
  - Sense low level
- Asynchronous Pin Change Sensing That Can Wake the Device From all Sleep Modes
- Efficient and Safe Access to Port Pins
  - Hardware read-modify-write through dedicated toggle/clear/set registers
  - Mapping of often-used PORT registers into bit-accessible I/O memory space (virtual ports)

16.2 Overview

The I/O pins of the device are controlled by instances of the Port Peripheral registers. This device has the following instances of the I/O pin configuration (PORT): PORTA, PORTB, and PORTC.

Refer to the I/O multiplexing table to see which pins are controlled by what instance of port. The offsets of the port instances and of the corresponding virtual port instances are listed in the Peripherals and Architecture section.

Each of the port pins has a corresponding bit in the Data Direction (PORT.DIR) and Data Output Value (PORT.OUT) registers to enable that pin as an output and to define the output state. For example, pin PA3 is controlled by DIR[3] and OUT[3] of the PORTA instance.

The Data Input Value (PORT.IN) is set as the input value of a port pin with resynchronization to the main clock. To reduce power consumption, these input synchronizers are not clocked if the Input Sense Configuration bit field (ISC) in PORT.PINnCTRL is INPUT_DISABLE. The value of the pin can always be read, whether the pin is configured as input or output.

The port supports synchronous and asynchronous input sensing with interrupts for selectable pin change conditions. Asynchronous pin-change sensing means that a pin change can wake the device from all sleep modes, including the modes where no clocks are running.

All pin functions are configurable individually per pin. The pins have hardware read-modify-write (RMW) functionality for a safe and correct change of drive value and/or pull resistor configuration. The direction of one port pin can be changed without unintentionally changing the direction of any other pin.

The port pin configuration controls input and output selection of other device functions.

Related Links

5. I/O Multiplexing and Considerations
7. Peripherals and Architecture

16.5.7 Output Value Clear

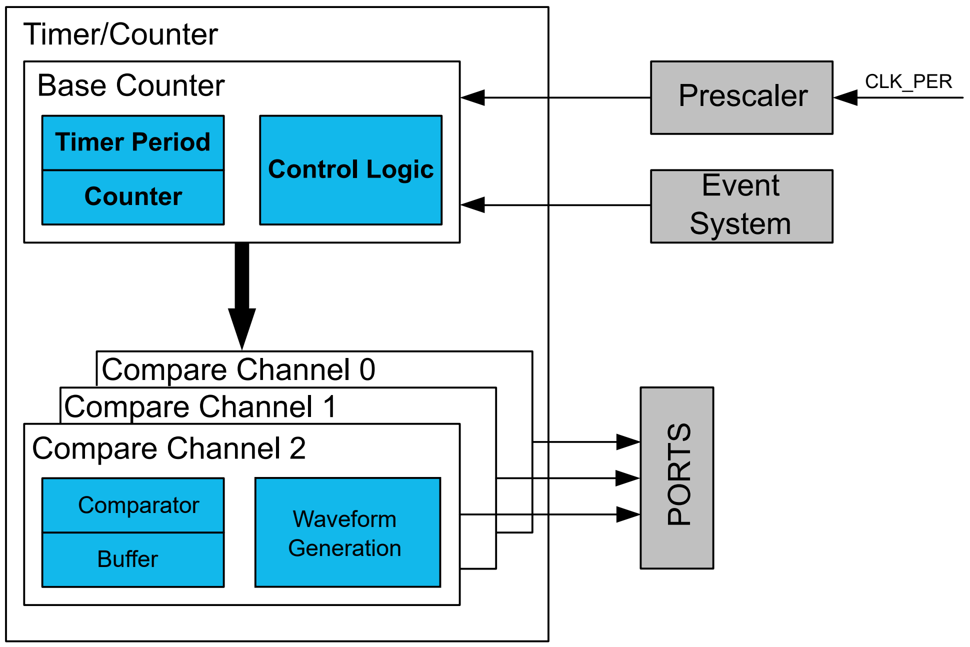
Name:	OUTCLR
Offset:	0x06
Reset:	0x00
Property:	-

Bit	7	6	5	4	3	2	1	0
	OUTCLR[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 – OUTCLR[7:0] Output Value Clear
This register can be used instead of a read-modify-write to clear the output value of individual pins to '0'. Writing a '1' to OUTCLR[n] will clear the corresponding bit in PORT.OUT.

Reading this bit field will always return the value of PORT.OUT.

Figure 20-1. 16-bit Timer/Counter and Closely Related Peripherals



This device provides one instance of the TCA peripheral, TCA0.

20.2.1 Block Diagram

The figure below shows a detailed block diagram of the timer/counter.

20.5.18 Compare n Buffer Register

Name:	CMPBUF
Offset:	0x38 + n*0x02 [n=0..2]
Reset:	0x00
Property:	-

This register serves as the buffer for the associated compare registers (TCAn.CMPn). Accessing any of these registers using the CPU or UPDI will affect the corresponding CMPnBV status bit.

The TCAn.CMPnBUFL and TCAn.CMPnBUFH register pair represents the 16-bit value, TCAn.CMPnBUF. The low byte [7:0] (suffix L) is accessible at the original offset. The high byte [15:8] (suffix H) can be accessed at offset + 0x01.

Bit	15	14	13	12	11	10	9	8
	CMPBUF[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
	CMPBUF[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 15:8 – CMPBUF[15:8] Compare High Byte
These bits hold the MSB of the 16-bit compare buffer register.

Bits 7:0 – CMPBUF[7:0] Compare Low Byte
These bits hold the LSB of the 16-bit compare buffer register.

23.3 RTC Functional Description

The RTC peripheral offers two timing functions: the Real-Time Counter (RTC) and a Periodic Interrupt Timer (PIT). This subsection describes the RTC.

Related Links

23.4 PIT Functional Description

23.3.1 Initialization

To operate the RTC, the source clock for the RTC counter must be configured before enabling the RTC peripheral, and the desired actions (interrupt requests, output Events).

Related Links

10. CLKCTRL - Clock Controller

23.4 PIT Functional Description

23.3.1.1 Configure the Clock CLK_RTC

To configure CLK_RTC, follow these steps:

1. Configure the desired oscillator to operate as required, in the Clock Controller peripheral (CLKCTRL).
2. Write the Clock Select bits (CLKSEL) in the Clock Selection register (RTC.CLKSEL) accordingly.

The CLK_RTC clock configuration is used by both RTC and PIT functionality.

23.3.1.2 Configure RTC

To operate the RTC, follow these steps:

1. Set the compare value in the Compare register (RTC.CMP), and/or the overflow value in the Top register (RTC.PER).
2. Enable the desired interrupts by writing to the respective Interrupt Enable bits (CMP, OVF) in the Interrupt Control register (RTC.INTCTRL).
3. Configure the RTC internal prescaler and enable the RTC by writing the desired value to the PRESCALER bit field and a '1' to the RTC Enable bit (RTCEN) in the Control A register (RTC.CTRLA).

Note: The RTC peripheral is used internally during device start-up. Always check the Busy bits in the RTC.STATUS and RTC.PITSTATUS registers, also on initial configuration.

23.3.2 Operation - RTC

23.3.2.1 Enabling, Disabling, and Resetting

The RTC is enabled by setting the Enable bit in the Control A register (ENABLE bit in RTC.CTRLA to 1). The RTC is disabled by writing ENABLE bit in RTC.CTRLA to 0.

23.4 PIT Functional Description

The RTC peripheral offers two timing functions: the Real-Time Counter (RTC) and a Periodic Interrupt Timer (PIT). This subsection describes the PIT.

Related Links

23.3 RTC Functional Description

Offset	Name	Vector Description	Conditions
			• RXC: Receive Complete Interrupt

When an interrupt condition occurs, the corresponding interrupt flag is set in the Interrupt Flags register of the peripheral (*peripheral*.INTFLAGS).

An interrupt source is enabled or disabled by writing to the corresponding enable bit in the peripheral's Interrupt Control register (*peripheral*.INTCTRL).

An interrupt request is generated when the corresponding interrupt source is enabled and the interrupt flag is set. The interrupt request remains active until the interrupt flag is cleared. See the peripheral's INTFLAGS register for details on how to clear interrupt flags.

Related Links

8.7.3 SREG

13. CPUINT - CPU Interrupt Controller

25.3.4 Sleep Mode Operation

The SPI will continue working in Idle Sleep mode. When entering any deeper sleep mode, an active transaction will be stopped.

Related Links

11. SLPCTRL - Sleep Controller

25.3.5 Configuration Change Protection

Not applicable.

26.5.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: -

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

Bit 4 – SDASETUP SDA Setup Time
By default, there are four clock cycles of setup time on SDA out signal while reading from the slave part of the TWI module. Writing this bit to '1' will change the setup time to eight clocks.

Value	Name	Description
0	4CYC	SDA setup time is four clock cycles
1	8CYC	SDA setup time is eight clock cycle

Bits 3:2 – SDAHOLD[1:0] SDA Hold Time
Writing these bits selects the SDA hold time.

Table 26-3. SDA Hold Time

SDAHOLD[1:0]	Nominal Hold Time	Hold Time Range Across All Corners in ns	Description
0x0	OFF	0	Hold time OFF.
0x1	50 ns	36 - 131	Backward compatible setting.
0x2	300 ns	180 - 630	Meets SMBus specification under typical conditions.
0x3	500 ns	300 - 1050	Meets SMBus specification across all corners.

Bit 1 – FMPEN FM Plus Enable
Writing these bits selects the 1 MHz bus speed (Fast mode plus, Fm+) for the TWI in default configuration.

Value	Description
0	Fm+ disabled
1	Fm+ enabled

3. Writing to the TWIn.MDATA register.
4. Reading the TWIn.DATA register while the ACKACT control bits in TWIn.MCTRLB are set to either send ACK or NACK.
5. Writing a valid command to the TWIn.MCTRLB register.

Bit 4 – RXACK Received Acknowledge
This bit is read-only and contains the most recently received Acknowledge bit from the slave. When read as '0', the most recent acknowledge bit from the slave was ACK. When read as '1', the most recent acknowledge bit was NACK.

Bit 3 – ARBLOST Arbitration Lost
If read as '1' this bit indicates that the master has lost arbitration while transmitting a high data or NACK bit, or while issuing a Start or repeated Start condition (S/Sr) on the bus.

Writing a '1' to it will clear the ARBLOST flag. However, normal use of the TWI does not require the flag to be cleared by this method. However, as for the CLKHOLD flag, clearing the ARBLOST flag is not required during normal use of the TWI.

Clearing the ARBLOST bit will follow the same software interaction as the CLKHOLD flag.

Given the condition where the bus ownership is lost to another master, the software must either abort operation or resend the data packet. Either way, the next required software interaction is in both cases to write to the TWIn.MADDR register. A write access to the TWIn.MADDR register will then clear the ARBLOST flag.

Bit 2 – BUSERR Bus Error
The BUSERR flag indicates that an illegal bus condition has occurred. An illegal bus condition is detected if a protocol violating Start (S), repeated Start (Sr), or Stop (P) is detected on the TWI bus lines. A Start condition directly followed by a Stop condition is one example of protocol violation.

Writing a '1' to this bit will clear the BUSERR. However, normal use of the TWI does not require the BUSERR to be cleared by this method.

A robust TWI driver software design will treat the bus error flag similarly to the ARBLOST flag, assuming the bus ownership is lost when the bus error flag is set. As for the ARBLOST flag, the next software operation of writing the TWIn.MADDR register will consequently clear the BUSERR flag. For bus error to be detected, the bus state logic must be enabled and the system frequency must be 4x the SCL frequency.

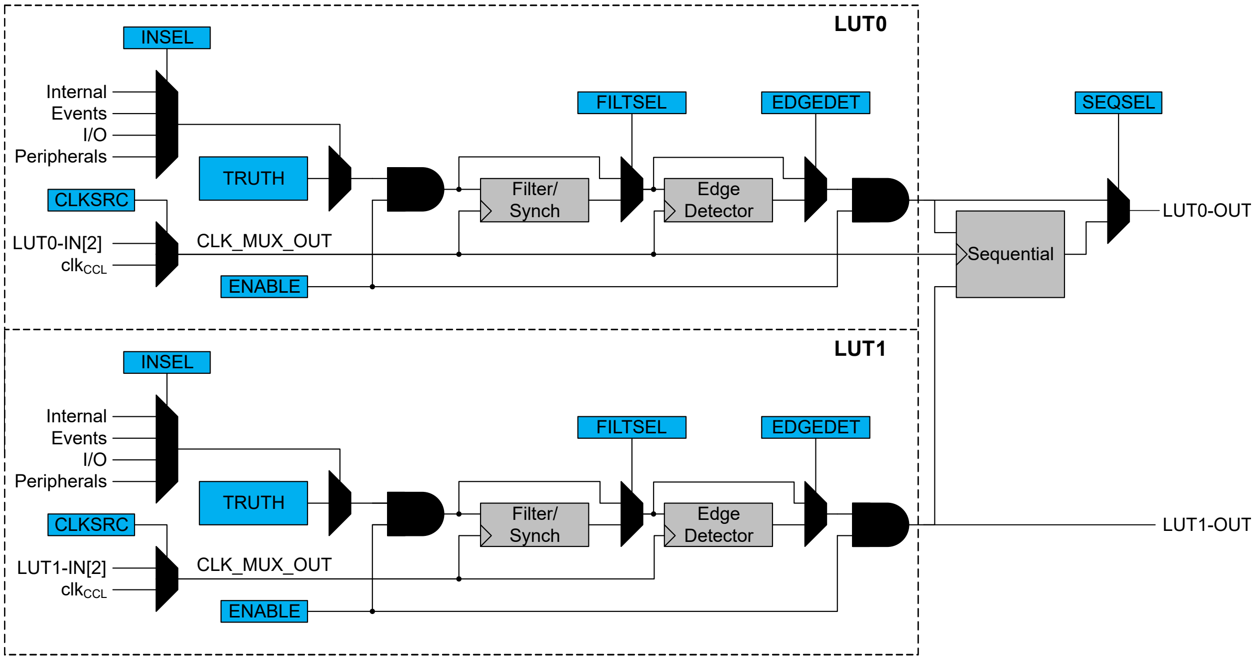
Bits 1:0 – BUSSTATE[1:0] Bus State
These bits indicate the current TWI bus state as defined in the table below. After a System Reset or re-enabling, the TWI master bus state will be unknown. The change of bus state is dependent on the bus activity.

Writing 0x1 to the BUSSTATE bits forces the bus state logic into its Idle state. However, the bus state logic cannot be forced into any other state. When the master is disabled, the bus state is 'unknown'.

Value	Name	Description
`0x0`	UNKNOWN	Unknown bus state
`0x1`	IDLE	Bus is idle
`0x2`	OWNER	This TWI controls the bus
`0x3`	BUSY	The bus is busy

28.2.1 Block Diagram

Figure 28-1. Configurable Custom Logic



28.2.2 Signal Description

Pin Name	Type	Description
LUTn-OUT	Digital output	Output from look-up table
LUTn-IN[2:0]	Digital input	Input to look-up table

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

Related Links

5. I/O Multiplexing and Considerations

28.2.3 System Dependencies

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

Table 28-1. CCL System Dependencies

Dependency	Applicable	Peripheral
Clocks	Yes	CLKCTRL
I/O Lines and Connections	Yes	PORT
Interrupts	Yes	CPUINT
Events	Yes	EVSYS
Debug	Yes	UPDI

When several interrupt request conditions are supported by an interrupt vector, the interrupt requests are ORed together into one combined interrupt request to the interrupt controller. The user must read the peripheral's INTFLAGS register to determine which of the interrupt conditions are present.

28.3.4 Events

The CCL can generate the following output events:

- LUTnOUT: Look-Up Table Output Value

The CCL can take the following actions on an input event:

- INx: The event is used as input for the TRUTH table

Related Links

14. EVSYS - Event System

28.3.5 Sleep Mode Operation

Writing the Run In Standby bit (RUNSTDBY) in the Control A register (CCL.CTRLA) to '1' will allow the system clock to be enabled in Standby Sleep mode.

If RUNSTDBY is '0' the system clock will be disabled in Standby Sleep mode. If the Filter, Edge Detector, or Sequential logic is enabled, the LUT output will be forced to '0' in Standby Sleep mode. In Idle sleep mode, the TRUTH table decoder will continue operation and the LUT output will be refreshed accordingly, regardless of the RUNSTDBY bit.

If the Clock Source bit (CLKSRC) in the LUT n Control A register (CCL.LUTnCTRLA) is written to '1', the LUT input 2 (IN[2]) will always clock the Filter, Edge Detector, and Sequential block. The availability of the IN[2] clock in sleep modes will depend on the sleep settings of the peripheral employed.

28.3.6 Configuration Change Protection

Not applicable.

30.5.7 MUXPOS

Name: MUXPOS
Offset: 0x06
Reset: 0x00
Property: -

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

Bits 4:0 – MUXPOS[4:0] MUXPOS
This bit field selects which single-ended analog input is connected to the ADC. If these bits are changed during a conversion, the change will not take effect until this conversion is complete.

Value	Name	Description
`0x00`	AIN0	ADC input pin 0
`0x01`	AIN1	ADC input pin 1
`0x02`	AIN2	ADC input pin 2
`0x03`	AIN3	ADC input pin 3
`0x04`	AIN4	ADC input pin 4
`0x05`	AIN5	ADC input pin 5
`0x06`	AIN6	ADC input pin 6
`0x07`	AIN7	ADC input pin 7
`0x08`	AIN8	ADC input pin 8
`0x09`	AIN9	ADC input pin 9
`0x0A`	AIN10	ADC input pin 10
`0x0B`	AIN11	ADC input pin 11
`0x1B`	PTC	ADC0: Reserved / ADC1: DAC2
`0x1C`	DAC0	DAC0
`0x1D`	INTREF	Internal reference (from VREF peripheral)
`0x1E`	TEMPSENSE	ADC0: Temperature sensor / ADC1: DAC1
`0x1F`	GND	0V (GND)
`Other`	-	Reserved

31.2.3 System Dependencies

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

Table 31-1. DAC System Dependencies

Dependency	Applicable	Peripheral
Clocks	Yes	CLKCTRL
I/O Lines and Connections	Yes	PORT
Interrupts	No	-
Events	No	-
Debug	Yes	UPDI

Related Links

11.2.2.1 Clocks
31.2.3.2 I/O Lines and Connections
31.2.3.5 Debug Operation

31.2.3.1 Clocks

This peripheral depends on the peripheral clock.

Related Links

10. CLKCTRL - Clock Controller

31.2.3.2 I/O Lines and Connections

Using the I/O lines of the peripheral requires configuration of the I/O pins.

Table 31-2. I/O Lines

Instance	Signal	I/O Line	Peripheral Function
DAC0	OUT	PA6	A

The DAC0 has one analog output pin (OUT) that must be configured before it can be used.

A DAC is also internally connected to the AC and to the ADC. To use this internal OUT as input, both output and input must be configured in their respective registers.

Note: Only DAC0 has an output driver for an external pin.

Related Links

16. PORT - I/O Pin Configuration
29. AC - Analog Comparator
30. ADC - Analog-to-Digital Converter

31.2.3.3 Events

Not applicable.

31.2.3.4 Interrupts

Not applicable.

31.2.3.5 Debug Operation

This peripheral is unaffected by entering Debug mode.

31.5.1 Control A

Name:	CTRLA
Offset:	0x00
Reset:	0x00
Property:	-

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY	OUTEN						ENABLE
Access	R/W	R/W						R/W
Reset	0	0						0

Bit 7 – RUNSTDBY Run in Standby Mode
If this bit is written to '1', the DAC or output buffer will not automatically be disabled when the device is entering Standby Sleep mode.

Note: Only DAC0 has an output driver for an external pin.

Bit 6 – OUTEN Output Buffer Enable
Writing a '1' to this bit enables the output buffer and sends the OUT signal to a pin.
Note: Only DAC0 has an output driver for an external pin.

Bit 0 – ENABLE DAC Enable
Writing a '1' to this bit enables the DAC.

33.4 Register Summary - UPDI

Offset	Name	Bit Pos.								
0x00	STATUSA	7:0	UPDIREV[3:0]							
0x01	STATUSB	7:0						PESIG[2:0]		
0x02	CTRLA	7:0	IBDLY		PARD	DTD	RSD	GTVAL[2:0]		
0x03	CTRLB	7:0				NACKDIS	CCDETDIS	UPDIDIS		
0x04 ... 0x06	Reserved									
0x07	ASI_KEY_STATUS	7:0			UROWWRITE	NVMPROG	CHIPERASE			
0x08	ASI_RESET_REQ	7:0	RSTREQ[7:0]							
0x09	ASI_CTRLA	7:0							UPDICLKSEL[1:0]	
0x0A	ASI_SYS_CTRLA	7:0							UROWWRITE_FINAL	CLKREQ
0x0B	ASI_SYS_STATUS	7:0			RSTSYS	INSLEEP	NVMPROG	UROWPROG		LOCKSTATUS
0x0C	ASI_CRC_STATUS	7:0						CRC_STATUS[2:0]		

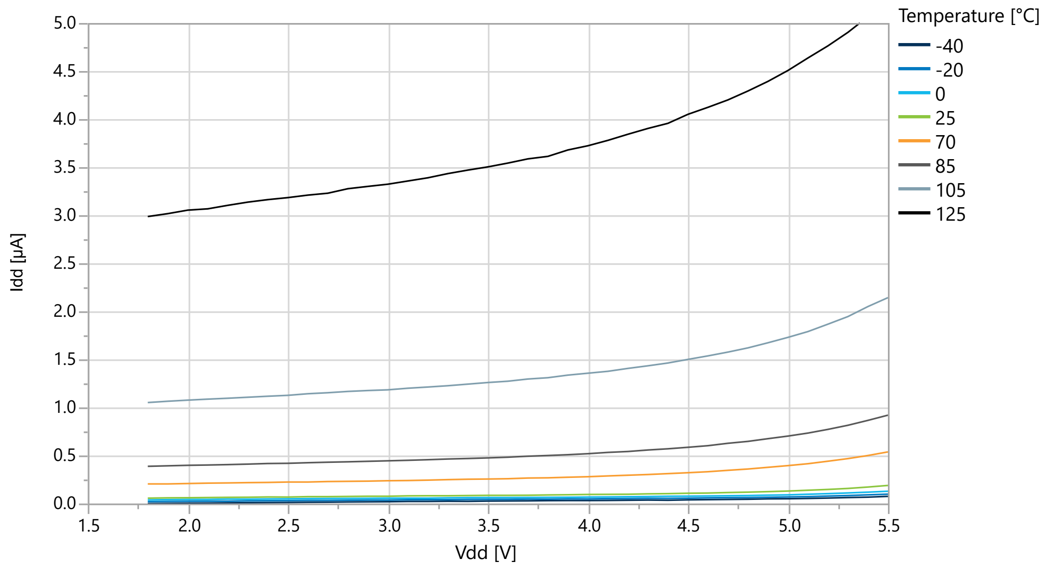
33.5 Register Description

These registers are readable only through the UPDI with special instructions and are NOT readable through the CPU.

Registers at offset addresses 0x0-0x3 are the UPDI Physical configuration registers.

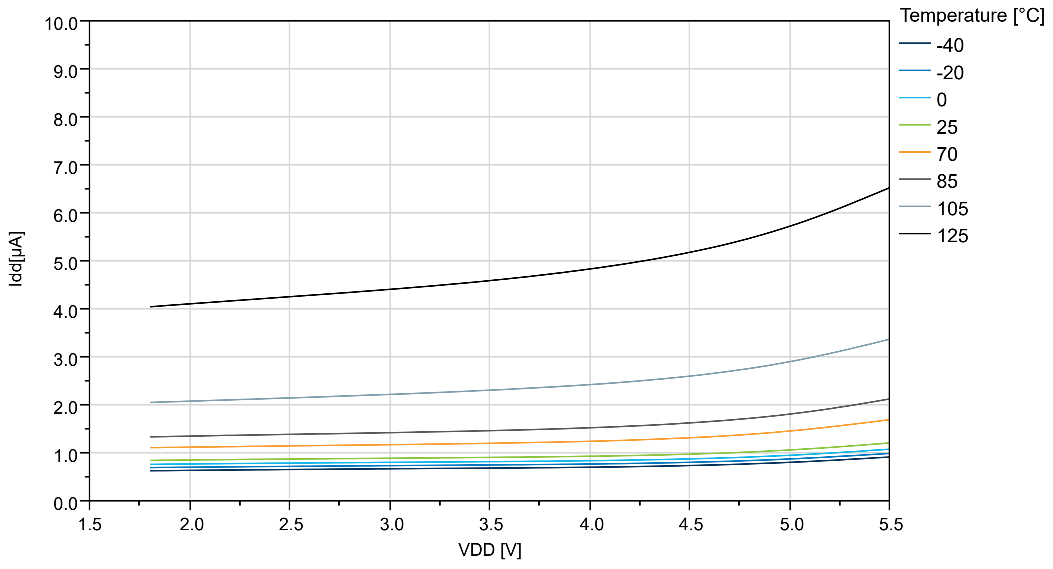
Registers at offset addresses 0x4-0xC are the ASI level registers.

Figure 38-11. ATtiny1617 Power-Down Mode Supply Current vs. V_{DD} (all functions disabled)



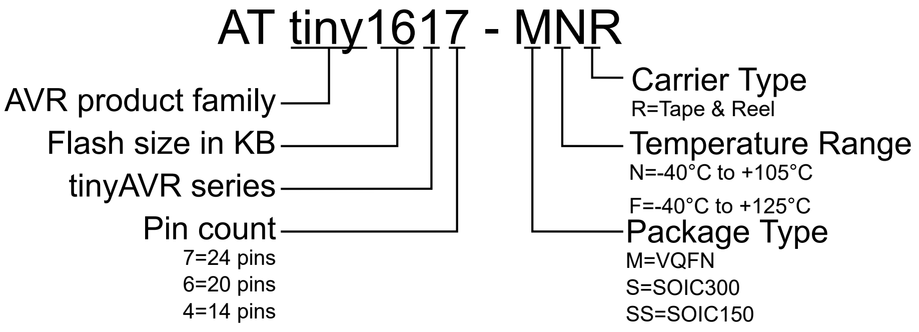
38.1.4 Supply Currents in Standby Mode for ATtiny1617

Figure 38-12. ATtiny1617 Standby Mode Supply Current vs. V_{DD} (RTC Running with External 32 KHz Osc.)



Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Microchip Devices Code Protection Feature

Legal Notice