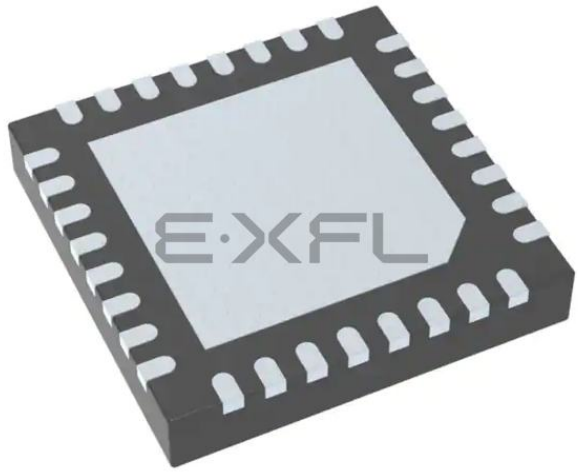


Welcome to E-XFL.COM

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	16MHz
Connectivity	I²C, LINbus, SPI, UART/USART, USI
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-VQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny167-mur

Table 4-10. Clock Prescaler Select

CLKPS3	CLKPS2	CLKPS1	CLKPS0	Clock Division Factor
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

4.5.3 CLKCSR – Clock Control & Status Register

Bit	7	6	5	4	3	2	1	0	
(0x62)	**CLKCCE**	–	–	**CLKRDY**	**CLKC3**	**CLKC2**	**CLKC1**	**CLKC0**	**CLKCSR**
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – CLKCCE: Clock Control Change Enable**

The CLKCCE bit must be written to logic one to enable change of the CLKCSR bits. The CLKCCE bit is only updated when the other bits in CLKCSR are simultaneously written to zero. CLKCCE is cleared by hardware four cycles after it is written or when the CLKCSR bits are written. Rewriting the CLKCCE bit within this time-out period does neither extend the time-out period, nor clear the CLKCCE bit.

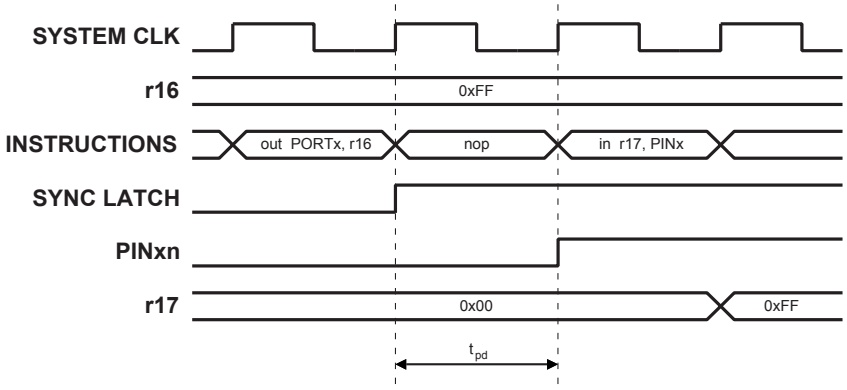
- **Bits 6:5 – Res: Reserved Bits**

These bits are reserved bits in the ATtiny87/167 and will always read as zero.

- **Bit 4 – CLKRDY: Clock Ready Flag**

This flag is the output of the '*Clock Availability* ' logic.
This flag is cleared by the '*Request for Clock Availability*' command or '*Enable Clock Source*' command being entered.
It is set when '*Clock Availability*' logic confirms that the (selected) clock is running and is stable. The delay from the request and the flag setting is not fixed, it depends on the clock start-up time,

Figure 9-5. Synchronization when Reading a Software Assigned Pin Value



The following code example shows how to set port B pins 0 and 1 high, 2 and 3 low, and define the port pins from 4 to 7 as input with pull-ups assigned to port pins 6 and 7. The resulting pin values are read back again, but as previously discussed, a nop instruction is included to be able to read back the value recently assigned to some of the pins.

Assembly Code Example[1]

```
...
; Define pull-ups and set outputs high
; Define directions for port pins
ldi   r16,(1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0)
ldi   r17,(1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)
out   PORTB,r16
out   DDRB,r17
; Insert nop for synchronization
nop
; Read port pins
in    r16,PINB
...
```

C Code Example

```
unsigned char i;
...
/* Define pull-ups and set outputs high */
/* Define directions for port pins */
PORTB = (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0);
DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
/* Insert nop for synchronization*/
__no_operation();
/* Read port pins */
i = PINB;
...
```

Note: 1. For the assembly program, two temporary registers are used to minimize the time from pull-ups are set on pins 0, 1, 6, and 7, until the direction bits are correctly set, defining bit 2 and 3 as low and redefining bits 0 and 1 as strong high drivers.

- MOSI: SPI Master Output / Slave Input. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDA4. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDA4. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTA4 bit.

- **Port A, Bit 3 – PCINT3/ADC3/ISRC/INT1**
  - PCINT3: Pin Change Interrupt, source 3.
  - ADC3: Analog to Digital Converter, channel 3.
  - ISCR: Current Source Output pin. While current is sourced by the Current Source module, the user can use the Analog to Digital Converter channel 3 (ADC3) to measure the pin voltage.
  - INT1: External Interrupt, source 1. The PA3 pin can serve as an external interrupt source.

- **Port A, Bit 2 – PCINT2/ADC2/OC0A/DO/MISO**
  - PCINT2: Pin Change Interrupt, source 2.
  - ADC2: Analog to Digital Converter, channel 2.
  - OC0A: Output Compare Match A or output PWM A for Timer/Counter0. The pin has to be configured as an output (DDA2 set (one)) to serve these functions.
  - DO: Three-wire Mode USI Data Output. Three-wire mode data output overrides PORTA2 and it is driven to the port when the data direction bit DDA2 is set. PORTA2 still enables the pull-up, if the direction is input and PORTA2 is set (one).
  - MISO: Master Data input, Slave Data output pin for SPI channel. When the SPI is enabled as a Master, this pin is configured as an input regardless of the setting of DDA2. When the SPI is enabled as a Salve, the data direction of this pin is controlled by DDA2. When the pin is forced to be an input, the pull-up can still be controlled by PORTA2.

- **Port A, Bit 1 – PCINT1/ADC1/TXD/TXLIN**
  - PCINT1: Pin Change Interrupt, source 1.
  - ADC1: Analog to Digital Converter, channel 1.
  - TXD: UART Transmit pin. When the UART transmitter is enabled, this pin is configured as an output regardless the value of DDA1. PORTA1 still enables the pull-up, if the direction is input and PORTA2 is set (one).
  - TXLIN: LIN Transmit pin. When the LIN is enabled, this pin is configured as an output regardless the value of DDA1. PORTA1 still enables the pull-up, if the direction is input and PORTA2 is set (one).

- **Port A, Bit 0 – PCINT0/ADC0/RXD/RXLIN**
  - PCINT0: Pin Change Interrupt, source 0.
  - ADC0: Analog to Digital Converter, channel 0.
  - RXD: UART Receive pin. When the UART receiver is enabled, this pin is configured as an input regardless of the value of DDA0. When the pin is forced to be an input, a logical one in PORTA0 will turn on the internal pull-up.
  - RXLIN: LIN Receive pin. When the LIN is enabled, this pin is configured as an input regardless of the value of DDA0. When the pin is forced to be an input, a logical one in PORTA0 will turn on the internal pull-up.

- PCINT10: Pin Change Interrupt, source 10.
- OC1AV: Output Compare and PWM Output A-V for Timer/Counter1. The PB2 pin has to be configured as an output (DDB2 set (one)) to serve this function. The OC1AV pin is also the output pin for the PWM mode timer function (c.f. OC1AV bit of TCCR1D register).
- USCK: Three-wire Mode USI Clock Input.
- SCL: Two-wire Mode USI Clock Input.

- **Port B, Bit 1 – PCINT9/OC1BU/DO**
  - PCINT9: Pin Change Interrupt, source 9.
  - OC1BU: Output Compare and PWM Output B-U for Timer/Counter1. The PB1 pin has to be configured as an output (DDB1 set (one)) to serve this function. The OC1BU pin is also the output pin for the PWM mode timer function (c.f. OC1BU bit of TCCR1D register).
  - DO: Three-wire Mode USI Data Output. Three-wire mode data output overrides PORTB1 and it is driven to the port when the data direction bit DDB1 is set. PORTB1 still enables the pull-up, if the direction is input and PORTB1 is set (one).

- **Port B, Bit 0 – PCINT8/OC1AU/DI/SDA**
  - IPCINT8: Pin Change Interrupt, source 8.
  - OC1AU: Output Compare and PWM Output A-U for Timer/Counter1. The PB0 pin has to be configured as an output (DDB0 set (one)) to serve this function. The OC1AU pin is also the output pin for the PWM mode timer function (c.f. OC1AU bit of TCCR1D register).
  - DI: Three-wire Mode USI Data Input. USI Three-wire mode does not override normal port functions, so pin must be configure as an input for DI function.
  - SDA: Two-wire Mode Serial Interface (USI) Data Input / Output.

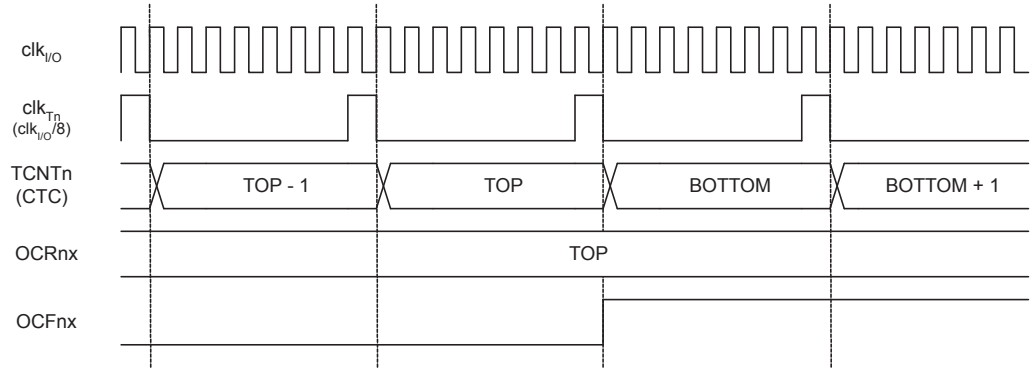
Table 9-7 and Table 9-8 relate the alternate functions of Port B to the overriding signals shown in Figure 9-6 on page 73.

Table 9-7. Overriding Signals for Alternate Functions in PB[7:4]

Signal Name	PB7/PCINT15/ ADC10/OC1BX/ $\overline{\text{RESET}}$/dW	PB6/PCINT14/ ADC9/OC1AX/INT0	PB5/PCINT13/ ADC8/OC1BW/ XTAL2/CLKO	PB4/PCINT12/ OC1AW/XTAL1/CLKI
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC1B_ENABLE & OC1BX	OC1A_ENABLE & OC1AX	OC1B_ENABLE & OC1BW	OC1A_ENABLE & OC1AW
PVOV	OC1B	OC1A	OC1B	OC1A
PTOE	0	0	0	0
DIEOE	ADC10D I (PCIE1 & PCMSK15)	ADC9D I INT0_ENABLE I (PCIE1 & PCMSK14)	ADC8D I (PCIE1 & PCMSK13)	(PCIE1 & PCMSK13)
DIEOV	PCIE1 & PCMSK15	INT0_ENABLE I (PCIE1 & PCMSK14)	PCIE1 & PCMSK13	1
DI	PCINT15	PCINT14 -/- INT1	PCINT13	PCINT12
AIO	RESET -/- ADC10 -/-	ADC9 -/- ISRC	ADC8 -/- XTAL2	XTAL1 -/- CLKI

Figure 10-11 shows the setting of OCF0A and the clearing of TCNT0 in CTC mode.

Figure 10-11. Timer/Counter Timing Diagram, Clear Timer on Compare Match mode, with Prescaler ($f_{clk\_I/O}/8$)



10.9 Asynchronous Operation of Timer/Counter0

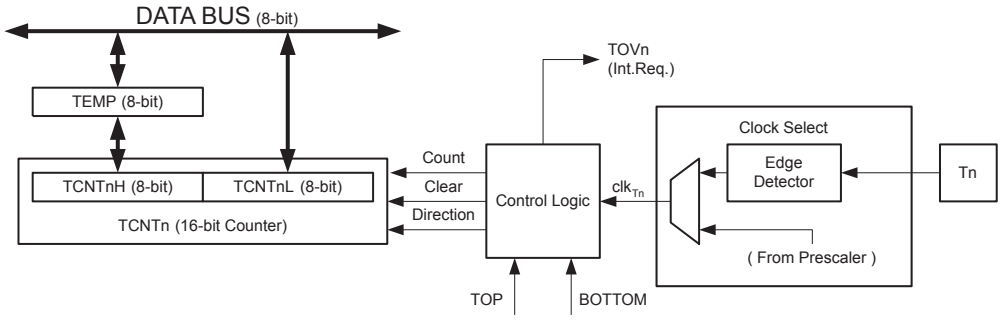
When Timer/Counter0 operates asynchronously, some considerations must be taken.

- Warning: When switching between asynchronous and synchronous clocking of Timer/Counter0, the timer registers TCNT0, OCR0A, and TCCR0A might be corrupted. A safe procedure for switching clock source is:
  a. Disable the Timer/Counter0 interrupts by clearing OCIE0A and TOIE0.
  b. Select clock source by setting AS0 and EXCLK as appropriate.
  c. Write new values to TCNT0, OCR0A, and TCCR0A.
  d. To switch to asynchronous operation: Wait for TCN0UB, OCR0UB, and TCR0UB.
  e. Clear the Timer/Counter0 interrupt flags.
  f. Enable interrupts, if needed.
- If an 32.768 kHz watch crystal is used, the CPU main clock frequency must be more than four times the Oscillator or external clock frequency.
- When writing to one of the registers TCNT0, OCR0A, or TCCR0A, the value is transferred to a temporary register, and latched after two positive edges on TOSC1. The user should not write a new value before the contents of the temporary register have been transferred to its destination. Each of the three mentioned registers have their individual temporary register, which means that e.g. writing to TCNT0 does not disturb an OCR0A write in progress. To detect that a transfer to the destination register has taken place, the Asynchronous Status Register – ASSR has been implemented.
- When entering Power-save mode after having written to TCNT0, OCR0A, or TCCR0A, the user must wait until the written register has been updated if Timer/Counter0 is used to wake up the device. Otherwise, the MCU will enter sleep mode before the changes are effective. This is particularly important if the Output Compare0 interrupt is used to wake up the device, since the Output Compare function is disabled during writing to OCR0A or TCNT0. If the write cycle is not finished, and the MCU enters sleep mode before the OCR0UB bit returns to zero, the device will never receive a compare match interrupt, and the MCU will not wake up.
- If Timer/Counter0 is used to wake the device up from Power-save mode, precautions must be taken if the user wants to re-enter one of these modes: The interrupt logic needs one TOSC1 cycle to be reset. If the time between wake-up and re-entering sleep mode is less than one

12.5 Counter Unit

The main part of the 16-bit Timer/Counter is the programmable 16-bit bi-directional counter unit. Figure 12-2 shows a block diagram of the counter and its surroundings.

Figure 12-2. Counter Unit Block Diagram



Signal description (internal signals):

Count	Increment or decrement TCNT1 by 1.
Direction	Select between increment and decrement.
Clear	Clear TCNT1 (set all bits to zero).
clk_T1	Timer/Counter clock.
TOP	Signalize that TCNT1 has reached maximum value.
BOTTOM	Signalize that TCNT1 has reached minimum value (zero).

The 16-bit counter is mapped into two 8-bit I/O memory locations: Counter High (TCNT1H) containing the upper eight bits of the counter, and Counter Low (TCNT1L) containing the lower eight bits. The TCNT1H Register can only be indirectly accessed by the CPU. When the CPU does an access to the TCNT1H I/O location, the CPU accesses the high byte temporary register (TEMP). The temporary register is updated with the TCNT1H value when the TCNT1L is read, and TCNT1H is updated with the temporary register value when TCNT1L is written. This allows the CPU to read or write the entire 16-bit counter value within one clock cycle via the 8-bit data bus. It is important to notice that there are special cases of writing to the TCNT1 Register when the counter is counting that will give unpredictable results. The special cases are described in the sections where they are of importance.

Depending on the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_T1). The clk_T1 can be generated from an external or internal clock source, selected by the Clock Select bits (CS1[2:0]). When no clock source is selected (CS1[2:0] = 0) the timer is stopped. However, the TCNT1 value can be accessed by the CPU, independent of whether clk_T1 is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the Waveform Generation mode bits (WGM1[3:0]) located in the Timer/Counter Control Registers A and B (TCCR1A and TCCR1B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC1A/B. For more details about advanced counting sequences and waveform generation, see “Modes of Operation” on page 121.

If external pin modes are used for the Timer/Counter1, transitions on the T1 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

12.11.3 TCCR1C – Timer/Counter1 Control Register C

Bit	7	6	5	4	3	2	1	0	
(0x82)	FOC1A	FOC1B	–	–	–	–	–	–	TCCR1C
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC1A: Force Output Compare for Channel A**

- **Bit 6 – FOC1B: Force Output Compare for Channel B**

The FOC1A/FOC1B bits are only active when the WGM1[3:0] bits specifies a non-PWM mode. However, for ensuring compatibility with future devices, these bits must be set to zero when TCCR1A is written when operating in a PWM mode. When writing a logical one to the FOC1A/FOC1B bit, an immediate compare match is forced on the Waveform Generation unit. The OC1nx output is changed according to its COM1A/B[1:0] and OC1nx bits setting. Note that the FOC1A/FOC1B bits are implemented as strobes. Therefore it is the value present in the COM1A/B[1:0] bits that determine the effect of the forced compare.

A FOC1A/FOC1B strobe will not generate any interrupt nor will it clear the timer in Clear Timer on Compare match (CTC) mode using OCR1A as TOP.

The FOC1A/FOC1B bits are always read as zero.

12.11.4 TCCR1D – Timer/Counter1 Control Register D

Bit	7	6	5	4	3	2	1	0	
(0x83)	OC1BX	OC1BW	OC1BV	OC1BU	OC1AX	OC1AW	OC1AV	OC1AU	TCCR1D
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:4 – OC1Bi: Output Compare Pin Enable for Channel B**

The OC1Bi bits enable the Output Compare pins of Channel B as shown in Figure 12-6 on page 121.

- **Bits 3:0 – OC1Ai: Output Compare Pin Enable for Channel A**

The OC1Ai bits enable the Output Compare pins of Channel A as shown in Figure 12-6 on page 121.

Table 14-1. Relations between USIWM[1:0] and the USI Operation

USIWM1	USIWM0	Description
0	0	Outputs, clock hold, and start detector disabled. Port pins operates as normal.
0	1	Three-wire mode. Uses DO, DI, and USCK pins. The *Data Output* (DO) pin overrides the corresponding bit in the PORT Register in this mode. However, the corresponding DDR bit still controls the data direction. When the port pin is set as input the pins pull-up is controlled by the PORT bit. The *Data Input* (DI) and *Serial Clock* (USCK) pins do not affect the normal port operation. When operating as master, clock pulses are software generated by toggling the PORT Register, while the data direction is set to output. The USITC bit in the USICR Register can be used for this purpose.
1	0	Two-wire mode. Uses SDA (DI) and SCL (USCK) pins(1). The *Serial Data* (SDA) and the *Serial Clock* (SCL) pins are bi-directional and uses open-collector output drives. The output drivers are enabled by setting the corresponding bit for SDA and SCL in the DDR Register. When the output driver is enabled for the SDA pin, the output driver will force the line SDA low if the output of the USI Data Register or the corresponding bit in the PORT Register is zero. Otherwise the SDA line will not be driven (i.e., it is released). When the SCL pin output driver is enabled the SCL line will be forced low if the corresponding bit in the PORT Register is zero, or by the start detector. Otherwise the SCL line will not be driven. The SCL line is held low when a start detector detects a start condition and the output is enabled. Clearing the Start Condition Flag (USISIF) releases the line. The SDA and SCL pin inputs is not affected by enabling this mode. Pull-ups on the SDA and SCL port pin are disabled in Two-wire mode.
1	1	Two-wire mode. Uses SDA and SCL pins. Same operation as for the Two-wire mode described above, except that the SCL line is also held low when a counter overflow occurs, and is held low until the Counter Overflow Flag (USIOIF) is cleared.

Note: 1. The DI and USCK pins are renamed to *Serial Data* (SDA) and *Serial Clock* (SCL) respectively to avoid confusion between the modes of operation.

- **Bits 3:2 – USICS[1:0]: Clock Source Select**

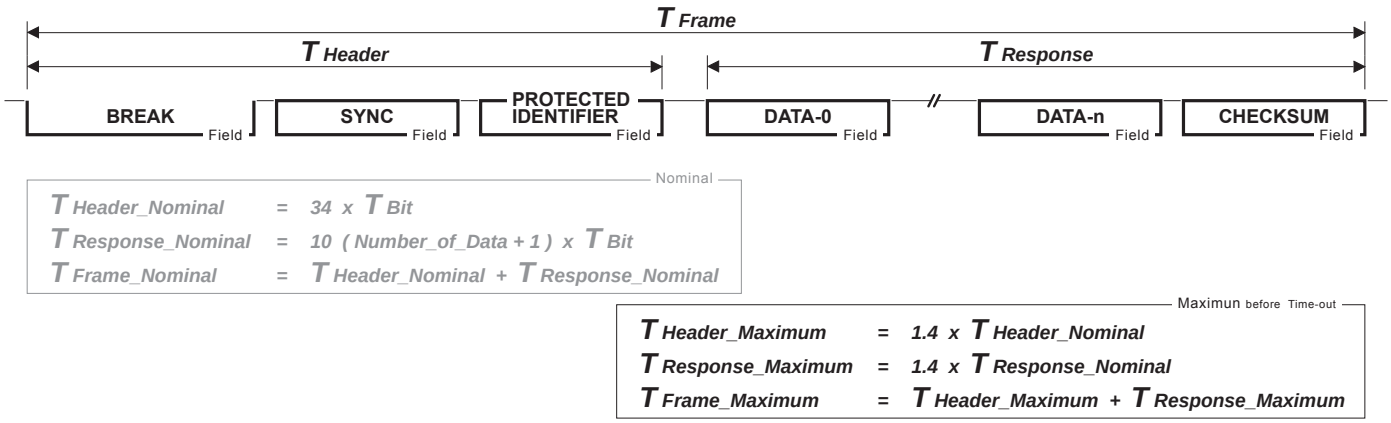
These bits set the clock source for the USI Data Register and counter. The data output latch ensures that the output is changed at the opposite edge of the sampling of the data input (DI/SDA) when using external clock source (USCK/SCL). When software strobe or Timer/Counter0 Compare Match clock option is selected, the output latch is transparent and therefore the output is changed immediately. Clearing the USICS[1:0] bits enables software strobe option. When using this option, writing a one to the USICLK bit clocks both the USI Data Register and the counter. For external clock source (USICS1 = 1), the USICLK bit is no longer used as a strobe, but selects between external clocking and software clocking by the USITC strobe bit.

Table 14-2 on page 159 shows the relationship between the USICS[1:0] and USICLK setting and clock source used for the USI Data Register and the 4-bit counter.

15.5.10 Frame Time Out

According to the LIN protocol, a frame time-out error is flagged if: $T_{Frame} > T_{Frame\_Maximum}$. This feature is implemented in the LIN/UART controller.

Figure 15-12. LIN timing and frame time-out



15.5.11 Break-in-data

According to the LIN protocol, the LIN/UART controller can detect the BREAK/SYNC field sequence even if the break is partially superimposed with a byte of the response. When a BREAK/SYNC field sequence happens, the transfer in progress is aborted and the processing of the new frame starts.

- On slave node(s), an error is generated (i.e. LBERR in case of *Tx Response* or LFERR in case of *Rx Response*). Information on data error is also available, refer to the Section 15.5.7.5.
- On master node, the user (code) is responsible for this aborting of frame. To do this, the master task has first to abort the on-going communication (clearing LCMD bits - *LIN Abort* command) and then to apply the *Tx Header* command. In this case, the abort error flag - LABORT - is set.

On the slave node, the BREAK detection is processed with the synchronization setting available when the LIN/UART controller processed the (aborted) response. But the re-synchronization restarts as usual. Due to a possible difference of timing reference between the BREAK field and the rest of the frame, the time-out values can be slightly inaccurate.

15.5.12 Checksum

The last field of a frame is the checksum.

In LIN 2.1, the checksum contains the inverted eight bit sum with carry over all data bytes and the protected identifier. This calculation is called enhanced checksum.

$$\text{CHECKSUM} = 255 - \left(\text{unsigned char} \left(\left(\sum_{0}^{n} \text{DATA}_n \right) + \text{PROTECTED ID.} \right) + \text{unsigned char} \left(\left(\left(\sum_{0}^{n} \text{DATA}_n \right) + \text{PROTECTED ID.} \right) \gg 8 \right) \right)$$

In LIN 1.3, the checksum contains the inverted eight bit sum with carry over all data bytes. This calculation is called classic checksum.

- **Bit 3 – $\overline{\text{LAINC}}$: $\overline{\text{Auto Increment}}$ of Data Buffer Index**

  In LIN mode:

  – 0 = Auto incrementation of FIFO data buffer index (default),

  – 1 = No auto incrementation.

  In UART mode this field is unused.

- **Bits 2:0 – LINDX[2:0]: FIFO LIN Data Buffer Index**

  In LIN mode: location (index) of the LIN response data byte into the FIFO data buffer. The FIFO data buffer is accessed through LINDAT.

  In UART mode this field is unused.

15.6.10 LINDAT – LIN Data Register

Bit	7	6	5	4	3	2	1	0	
(0xD2)	LDATA7	LDATA6	LDATA5	LDATA4	LDATA3	LDATA2	LDATA1	LDATA0	LINDAT
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:0 – LDATA[7:0]: LIN Data In / Data out**

  In LIN mode: FIFO data buffer port.

  In UART mode: data register (no data buffer - no FIFO).

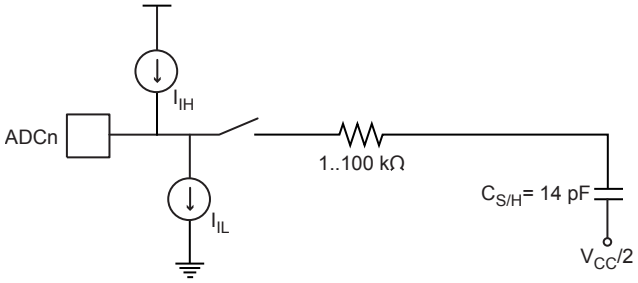
  – In Write access, data out.

  – In Read access, data in.

S/H capacitor, with can vary widely. The user is recommended to only use low impedant sources with slowly varying signals, since this minimizes the required charge transfer to the S/H capacitor.

Signal components higher than the Nyquist frequency ($f_{ADC}/2$) should not be present to avoid distortion from unpredictable signal convolution. The user is advised to remove high frequency components with a low-pass filter before applying the signals as inputs to the ADC.

Figure 17-8. Analog Input Circuitry



17.7.2 Analog Noise Canceling Techniques

Digital circuitry inside and outside the device generates EMI which might affect the accuracy of analog measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

a. Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane, and keep them well away from high-speed switching digital tracks.
b. Use the ADC noise canceler function to reduce induced noise from the CPU.
c. If any port pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress.

17.7.3 ADC Accuracy Definitions

An n-bit single-ended ADC converts a voltage linearly between GND and V_{REF} in 2^n steps (LSBs). The lowest code is read as 0, and the highest code is read as 2^n-1.

Several parameters describe the deviation from the ideal behavior:

- Offset: The deviation of the first transition (0x000 to 0x001) compared to the ideal transition (at 0.5 LSB). Ideal value: 0 LSB.

This bit enables the SPM instruction for the next four clock cycles. If written to one together with either SIGRD, CTPB, RFLB, PGWRT, or PGERS, the following SPM instruction will have a special meaning, see description above. If only SPMEN is written, the following SPM instruction will store the value in R1:R0 in the temporary page buffer addressed by the Z-pointer. The LSB of the Z-pointer is ignored. The SPMEN bit will auto-clear upon completion of an SPM instruction, or if no SPM instruction is executed within four clock cycles. During Page Erase and Page Write, the SPMEN bit remains high until the operation is completed.

Writing any other combination than “10 0001 $_b$”, “01 0001 $_b$”, “00 1001 $_b$”, “00 0101 $_b$”, “00 0011 $_b$” or “00 0001 $_b$” in the lower six bits will have no effect.

Note: Only one SPM instruction should be active at any time.

20.2.2 EEPROM Write Prevents Writing to SPMCSR

Note that an EEPROM write operation will block all software programming to Flash. Reading the Fuses and Lock bits from software will also be prevented during the EEPROM write operation. It is recommended that the user checks the status bit (EEPE) in the EECR Register and verifies that the bit is cleared before writing to the SPMCSR Register.

20.2.3 Reading the Fuse and Lock Bits from Software

It is possible to read both the Fuse and Lock bits from software. To read the Lock bits, load the Z-pointer with 0x0001 and set the RFLB and SPMEN bits in SPMCSR. When an LPM instruction is executed within three CPU cycles after the RFLB and SPMEN bits are set in SPMCSR, the value of the Lock bits will be loaded in the destination register. The RFLB and SPMEN bits will auto-clear upon completion of reading the Lock bits or if no LPM instruction is executed within three CPU cycles or no SPM instruction is executed within four CPU cycles. When RFLB and SPMEN are cleared, LPM will work as described in the Instruction set Manual.

Bit	7	6	5	4	3	2	1	0
Rd (Z=0x0001)	–	–	–	–	–	–	**LB2**	**LB1**

The algorithm for reading the Fuse Low byte is similar to the one described above for reading the Lock bits. To read the Fuse Low byte, load the Z-pointer with 0x0000 and set the RFLB and SPMEN bits in SPMCSR. When an LPM instruction is executed within three cycles after the RFLB and SPMEN bits are set in the SPMCSR, the value of the Fuse Low byte (FLB) will be loaded in the destination register as shown below. See Table 21-5 on page 226 for a detailed description and mapping of the Fuse Low byte.

Bit	7	6	5	4	3	2	1	0
Rd (Z=0x0000)	**FLB7**	**FLB6**	**FLB5**	**FLB4**	**FLB3**	**FLB2**	**FLB1**	**FLB0**

Similarly, when reading the Fuse High byte (FHB), load 0x0003 in the Z-pointer. When an LPM instruction is executed within three cycles after the RFLB and SPMEN bits are set in the SPMCSR, the value of the Fuse High byte will be loaded in the destination register as shown below. See Table 21-4 on page 225 for detailed description and mapping of the Fuse High byte.

Bit	7	6	5	4	3	2	1	0
Rd (Z=0x0003)	**FHB7**	**FHB6**	**FHB5**	**FHB4**	**FHB3**	**FHB2**	**FHB1**	**FHB0**

21.7 Parallel Programming

21.7.1 Enter Programming Mode

The following algorithm puts the device in parallel programming mode:

1. Apply 4.5 - 5.5V between V_{CC} and GND.
2. Set RESET to “0” and toggle XTAL1 at least six times.
3. Set the Prog_enable pins listed in Table 21-10 on page 229 to “0000 $_b$” and wait at least 100 ns.
4. Apply 11.5 - 12.5V to RESET. Any activity on Prog_enable pins within 100 ns after +12V has been applied to RESET, will cause the device to fail entering programming mode.
5. Wait at least 50 µs before sending a new command.

21.7.2 Considerations for Efficient Programming

The loaded command and address are retained in the device during programming. For efficient programming, the following should be considered.

- The command needs only be loaded once when writing or reading multiple memory locations.
- Skip writing the data value 0xFF, that is the contents of the entire EEPROM (unless the EESAVE Fuse is programmed) and Flash after a Chip Erase.
- Address high byte needs only be loaded before programming or reading a new 256 word window in Flash or 256 byte EEPROM. This consideration also applies to Signature bytes reading.

21.7.3 Chip Erase

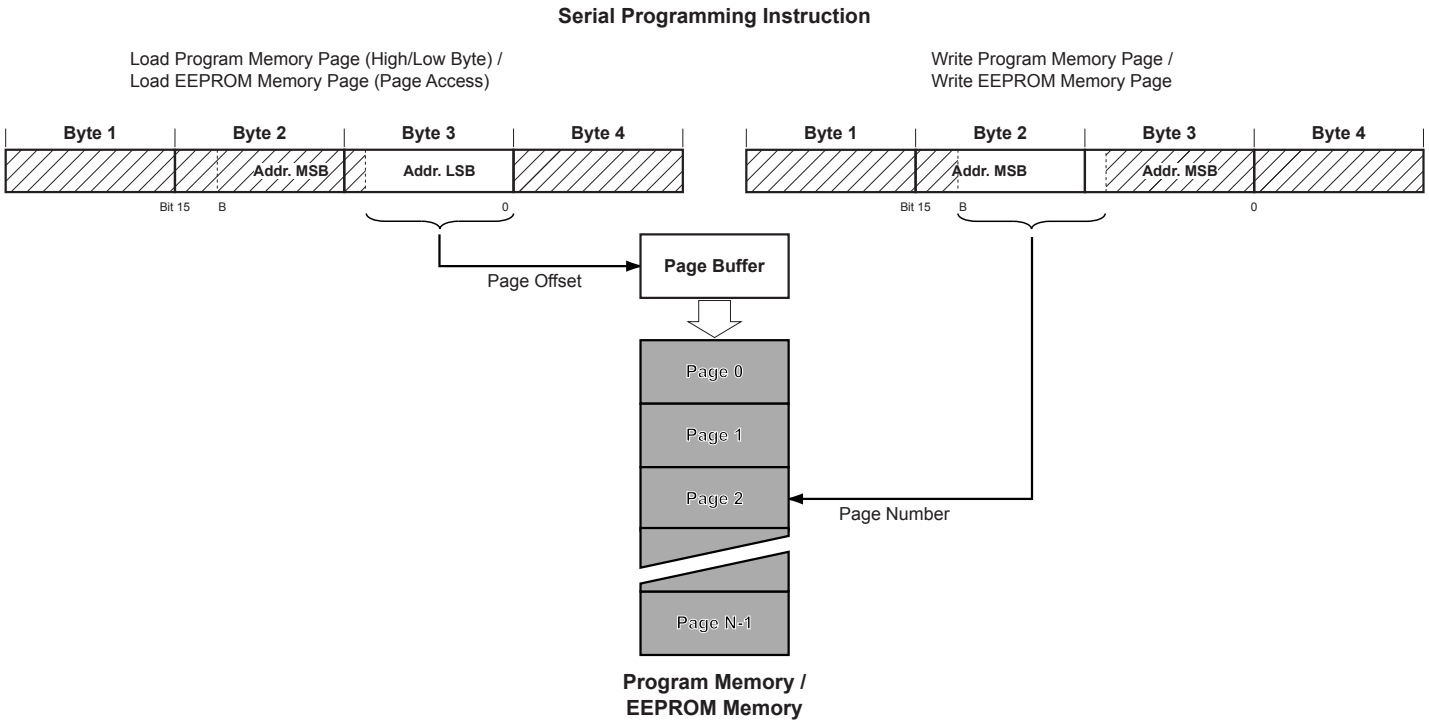
The Chip Erase will erase the Flash and EEPROM(Note:) memories plus Lock bits. The Lock bits are not reset until the program memory has been completely erased. The Fuse bits are not changed. A Chip Erase must be performed before the Flash and/or EEPROM are reprogrammed.

Note: The EEPRPOM memory is preserved during Chip Erase if the EESAVE Fuse is programmed.

Load Command “Chip Erase”

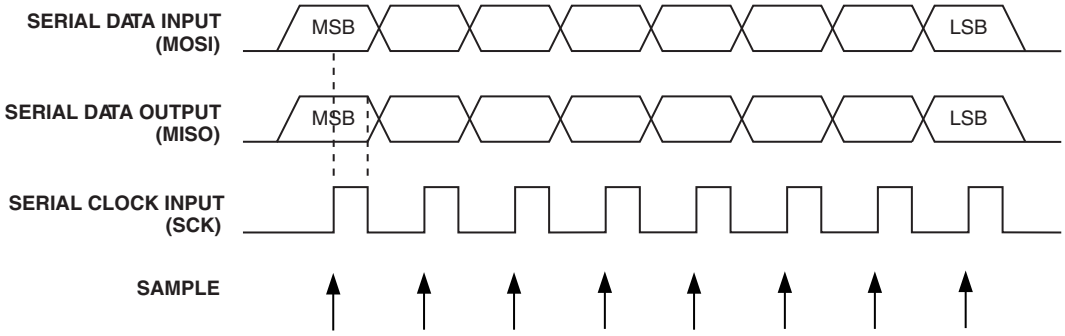
1. Set XA1, XA0 to “1,0 ”. This enables command loading.
2. Set BS1 to “0”.
3. Set DATA to “1000 0000 $_b$”. This is the command for Chip Erase.
4. Give XTAL1 a positive pulse. This loads the command.
5. Give $\overline{WR}$ a negative pulse. This starts the Chip Erase. RDY/$\overline{BSY}$ goes low.
6. Wait until RDY/$\overline{BSY}$ goes high before loading a new command.

Figure 21-8. Serial programming Instruction Example



21.9 Serial Programming Characteristics

Figure 21-9. Serial Programming Waveforms



For characteristics of the SPI module, See ”SPI Timing Characteristics” on page 252.

22. Electrical Characteristics

22.1 Absolute Maximum Ratings*

Operating Temperature.....– 40°C to +85°C Storage Temperature.....– 65°C to +150°C Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground .....– 0.5V to V_{CC}+0.5V Voltage on $\overline{\text{RESET}}$ with respect to Ground....– 0.5V to +13.0V Voltage on V_{CC} with respect to Ground.....– 0.5V to 6.0V DC Current per I/O Pin ..... 40.0 mA DC Current V_{CC} and GND Pins ..... 200.0 mA Injection Current at VCC = 0V to 5V(2)..... ±5.0mA(1)	*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Notes: 1. Maximum current per port = ±30mA
2. Functional corruption may occur .

22.2 DC Characteristics

T_A = -40°C to +85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.(1)	Max.	Units
V_{IL}	Input Low Voltage	Except XTAL1 and $\overline{\text{RESET}}$ pins	- 0.5		0.2 V_{CC} (2)	V
V_{IL1}		XTAL1 pin - External Clock Selected	- 0.5		0.1 V_{CC} (2)	V
V_{IL2}		$\overline{\text{RESET}}$ pin	- 0.5		0.2 V_{CC} (2)	V
V_{IL3}		$\overline{\text{RESET}}$ pin as I/O	- 0.5		0.2 V_{CC} (2)	V
V_{IH}	Input High Voltage	Except XTAL1 and $\overline{\text{RESET}}$ pins	0.7 V_{CC} (3)		V_{CC} + 0.5	V
V_{IH1}		XTAL1 pin - External Clock Selected	0.8 V_{CC} (3)		V_{CC} + 0.5	V
V_{IH2}		$\overline{\text{RESET}}$ pin	0.9 V_{CC} (3)		V_{CC} + 0.5	V
V_{IH3}		$\overline{\text{RESET}}$ pin as I/O	0.7 V_{CC} (3)		V_{CC} + 0.5	V
V_{OL}	Output Low Voltage (4) (Ports A, B,)	I_{OL} = 10 mA, V_{CC} = 5V I_{OL} = 5 mA, V_{CC} = 3V			0.6 0.5	V
V_{OH}	Output High Voltage (5) (Ports A, B)	I_{OH} = – 10 mA, V_{CC} = 5V I_{OH} = – 5 mA, V_{CC} = 3V	4.3 2.5			V
I_{IL}	Input Leakage Current I/O Pin	V_{CC} = 5.5V, pin low (absolute value)		< 0.05	1	µA
I_{IH}	Input Leakage Current I/O Pin	V_{CC} = 5.5V, pin high (absolute value)		< 0.05	1	µA

Figure 23-2. Active Supply Current vs. Frequency (1 - 16 MHz)

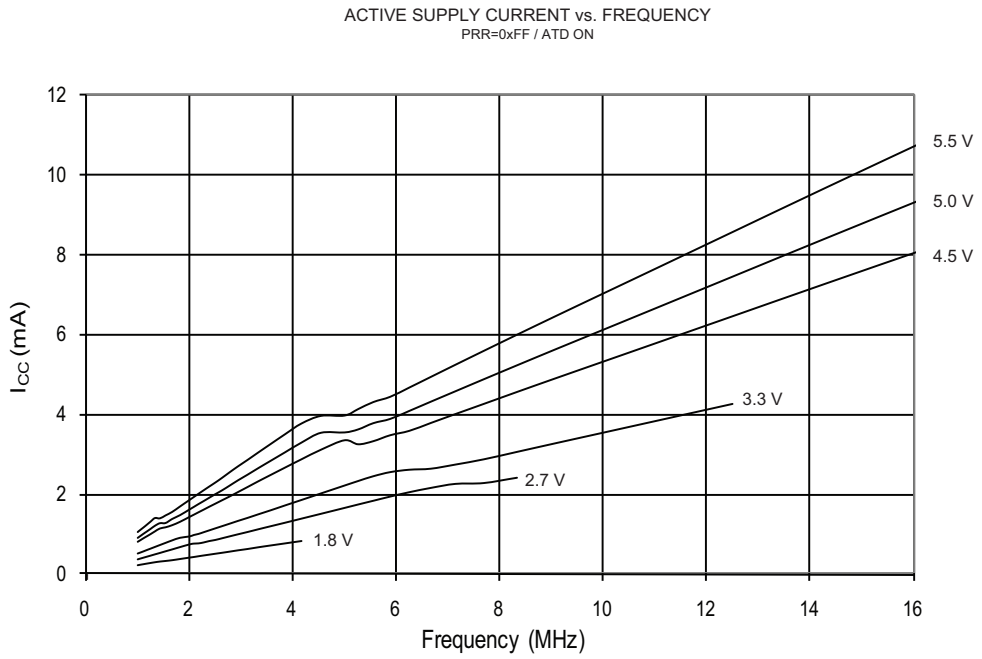
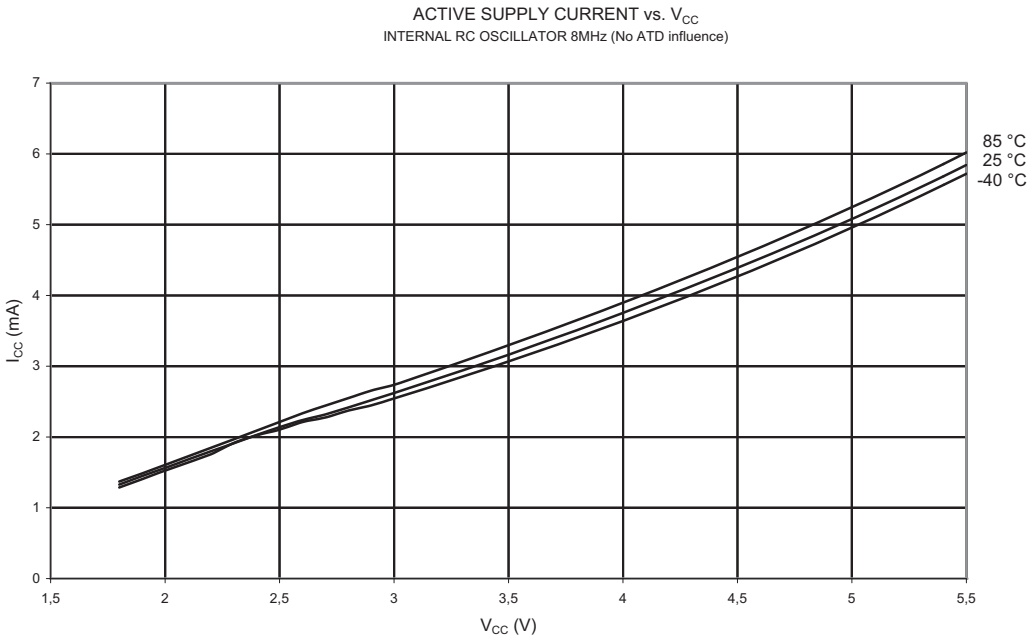


Figure 23-3. Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0x7C)	**ADMUX**	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	page 204
(0x7B)	**ADCSRB**	BIN	ACME	ACIR1	ACIR0	–	ADTS2	ADTS1	ADTS0	page 208, page 212
(0x7A)	**ADCSRA**	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	page 206
(0x79)	**ADCH**	- / ADC9	- / ADC8	- / ADC7	- / ADC6	- / ADC5	- / ADC4	ADC9 / ADC3	ADC8 / ADC2	page 207
(0x78)	**ADCL**	ADC7 / ADC1	ADC6 / ADC0	ADC5 / -	ADC4 / -	ADC3 / -	ADC2 / -	ADC1 / -	ADC0 /	page 207
(0x77)	**AMISCR**	–	–	–	–	–	AREFEN	XREFEN	ISRCEN	page 189, page 209
(0x76)	**Reserved**									
(0x75)	**Reserved**									
(0x74)	**Reserved**									
(0x73)	**Reserved**									
(0x72)	**Reserved**									
(0x71)	**Reserved**									
(0x70)	**Reserved**									
(0x6F)	**TIMSK1**	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	page 137
(0x6E)	**TIMSK0**	–	–	–	–	–	–	OCIE0A	TOIE0	page 104
(0x6D)	**Reserved**									
(0x6C)	**PCMSK1**	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	page 65
(0x6B)	**PCMSK0**	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	page 65
(0x6A)	**Reserved**									
(0x69)	**EICRA**	–	–	–	–	ISC11	ISC10	ISC01	ISC00	page 63
(0x68)	**PCICR**	–	–	–	–	–	–	PCIE1	PCIE0	page 64
(0x67)	**Reserved**									
(0x66)	**OSCCAL**	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	page 37
(0x65)	**Reserved**									
(0x64)	**PRR**	–	–	PRLIN	PRSPI	PRTIM1	PRTIM0	PRUSI	PRADC	page 47
(0x63)	**CLKSELR**	–	COUT	CSUT1	CSUT0	CSEL3	CSEL2	CSEL1	CSEL0	page 40
(0x62)	**CLKCSR**	CLKCCE	–	–	CLKRDY	CLKC3	CLKC2	CLKC1	CLKC0	page 38
(0x61)	**CLKPR**	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	page 38
(0x60)	**WDTCR**	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	page 57
0x3F (0x5F)	**SREG**	I	T	H	S	V	N	Z	C	page 9
0x3E (0x5E)	**SPH**	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	page 11
0x3D (0x5D)	**SPL**	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	page 11
0x3C (0x5C)	**Reserved**									
0x3B (0x5B)	**Reserved**									
0x3A (0x5A)	**Reserved**									
0x39 (0x59)	**Reserved**									
0x38 (0x58)	**Reserved**									
0x37 (0x57)	**SPMCSR**	–	RWWSB	SIGRD	CTPB	RFLB	PGWRT	PGERS	SPMEN	page 218
0x36 (0x56)	**Reserved**	–	–	–	–	–	–	–	–	
0x35 (0x55)	**MCUCR**	–	BODS	BODSE	PUD	–	–	–	–	page 47, page 75
0x34 (0x54)	**MCUSR**	–	–	–	–	WDRF	BORF	EXTRF	PORF	page 52
0x33 (0x53)	**SMCR**	–	–	–	–	–	SM1	SM0	SE	page 46
0x32 (0x52)	**Reserved**									
0x31 (0x51)	**DWDR**	DWDR7	DWDR6	DWDR5	DWDR4	DWDR3	DWDR2	DWDR1	DWDR0	page 215
0x30 (0x50)	**ACSR**	ACD	ACIRS	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	page 212
0x2F (0x4F)	**Reserved**									
0x2E (0x4E)	**SPDR**	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0	page 146
0x2D (0x4D)	**SPSR**	SPIF	WCOL	–	–	–	–	–	SPI2X	page 146
0x2C (0x4C)	**SPCR**	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	page 144
0x2B (0x4B)	**GPIOR2**	GPIOR27	GPIOR26	GPIOR25	GPIOR24	GPIOR23	GPIOR22	GPIOR21	GPIOR20	page 23
0x2A (0x4A)	**GPIOR1**	GPIOR17	GPIOR16	GPIOR15	GPIOR14	GPIOR13	GPIOR12	GPIOR11	GPIOR10	page 23
0x29 (0x49)	**Reserved**									
0x28 (0x48)	**OCR0A**	OCR0A7	OCR0A6	OCR0A5	OCR0A4	OCR0A3	OCR0A2	OCR0A1	OCR0A0	page 102
0x27 (0x47)	**TCNT0**	TCNT07	TCNT06	TCNT05	TCNT04	TCNT03	TCNT02	TCNT01	TCNT00	page 102
0x26 (0x46)	**TCCR0B**	FOC0A	–	–	–	–	CS02	CS01	CS00	page 101
0x25 (0x45)	**TCCR0A**	COM0A1	COM0A0	–	–	–	–	WGM01	WGM00	page 99
0x24 (0x44)	**Reserved**									
0x23 (0x43)	**GTCCR**	TSM	–	–	–	–	–	PSR0	PSR1	page 105, page 108
0x22 (0x42)	**EEARH**[1]	–	–	–	–	–	–	–	EEAR8	page 21
0x21 (0x41)	**EEARL**	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	page 21
0x20 (0x40)	**EEDR**	EEDR7	EEDR6	EEDR5	EEDR4	EEDR3	EEDR2	EEDR1	EEDR0	page 22
0x1F (0x3F)	**EECR**	–	–	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	page 22
0x1E (0x3E)	**GPIOR0**	GPIOR07	GPIOR06	GPIOR05	GPIOR04	GPIOR03	GPIOR02	GPIOR01	GPIOR00	page 23
0x1D (0x3D)	**EIMSK**	–	–	–	–	–	–	INT1	INT0	page 63
0x1C (0x3C)	**EIFR**	–	–	–	–	–	–	INTF1	INTF0	page 64
0x1B (0x3B)	**PCIFR**	–	–	–	–	–	–	PCIF1	PCIF0	page 65

26.2 ATtiny167

Speed (MHz)	Power Supply (V)	Ordering Code	Package[1]	Operational Range
16	1.8 – 5.5	ATtiny167-MU ATtiny167-MUR[2] ATtiny167-SU ATtiny167-SUR[2] ATtiny167-XU ATtiny167-XUR[2]	32PN 32PN 20S2 20S2 20X 20X	Industrial (-40°C to +85°C)[3]

Notes:
1. All packages are Pb-free, halide-free and fully green and they comply with the European directive for Restriction of Hazardous Substances (RoHS).
2. Tape and reel.
3. These devices can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.

Package Type	
32PN	32-lead, 0.5mm pitch, 5 x 5 mm Very Thin Quad Flat No Lead Package (VQFN) Sawn
20S2	20-lead, 0.300" Wide, Plastic Gull Wing Small Outline Package (SOIC)
20X	20-lead, 4.4 mm Wide, Plastic Thin Shrink Small Outline Package (TSSOP)

```
  ; Select watchdog clock ( 128KHz, fast rising power)
     ldi    temp3,((0x03<<CSEL0)|(0x02<<CSUT0))
     sts    CLKSELR, temp3        ; (*)
; (*) !!! Loose gain control of crystal oscillator !!!
; ==> WORKAROUND ...
     sts    CLKSELR, temp1
; ...
```

3. **'*Disable Clock Source*' command remains enabled.**
In the Dynamic Clock Switch module, the '*Disable Clock Source*' command remains running after disabling the targeted clock source (the clock source is set in the CLKSELR register).

Problem fix / workaround.
After a '*Disable Clock Source*' command, reset the CLKCSR register writing 0x80.

Code example:

```
  ; Select crystal oscillator
     ldi    temp1,(0x0F<<CSEL0)
     sts    CLKSELR, temp1
  ; Disable clock source (crystal oscillator)
     ldi    temp2,(1<<CLKCCE)
     ldi    temp3,(0x01<<CLKC0)   ; CSEL = "0001"
     sts    CLKCSR,temp2          ; Enable CLKCSR register access
     sts    CLKCSR,temp3          ; (*) Disable crystal oscillator clock
; (*) !!! At this moment, if any other clock source is selected by CLKSELR,
;         this clock source will also stop !!!
; ==> WORKAROUND ...
     sts    CLKCSR,temp2
```