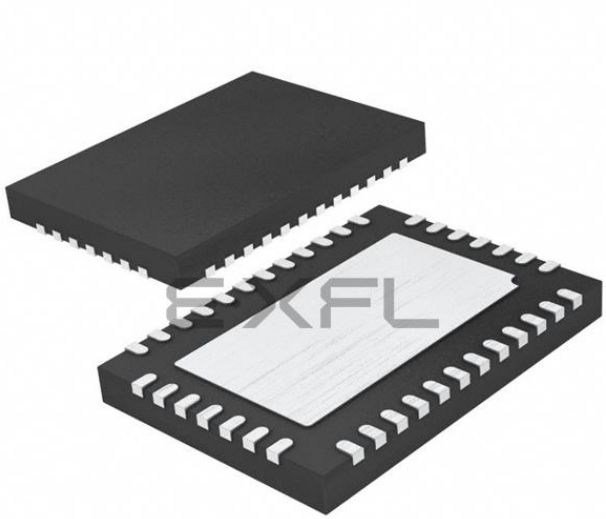


**Welcome to E-XFL.COM**

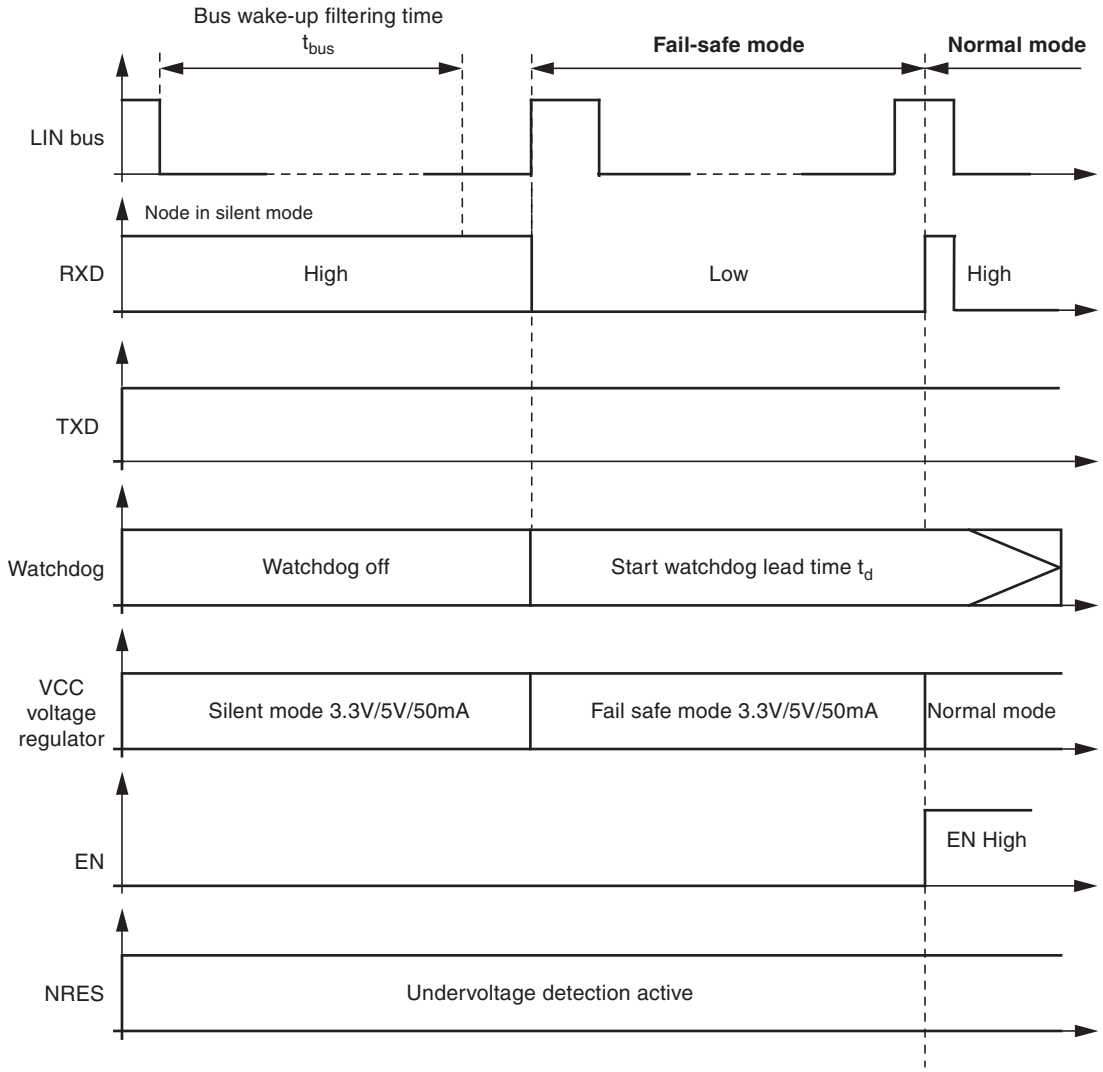
## What is "Embedded - Microcontrollers"?

"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 | Obsolete |
| 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 (16K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 512 x 8 |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.5V |
| Data Converters | A/D 11x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TC) |
| Mounting Type | Surface Mount |
| Package / Case | 38-VFQFN Exposed Pad |
| Supplier Device Package | 38-VQFN (5x7) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/ata6617-p3qw |

**Figure 3-4.** LIN Wake-up from Silent Mode




### 3.4.3 Sleep Mode

A falling edge at EN when TXD is low switches the IC into Sleep Mode. The TXD Signal has to be logic low during the Mode Select window (Figure 3-5 on page 12). In order to avoid any influence to the LIN-pin during switching into sleep mode it is possible to switch the EN up to 3.2µs earlier to LOW than the TXD. Therefore, the best and easiest way are two falling edges at TXD and EN at the same time.The transmission path is disabled in Sleep Mode. The supply current $I_{VSsleep}$ from $V_{Batt}$ is typically 10µA.

The VCC regulator is switched off. NRES and RXD are low. The internal slave termination between the LIN pin and VS pin is disabled, only a weak pull-up current (typically 10µA) between the LIN pin and the VS pin is present. Sleep Mode can be activated independently from the current level on the LIN, WAKE, or KL_15 pin.

A voltage less than the LIN Pre_Wake detection VLINL at the LIN pin activates the internal LIN receiver and switches on the internal slave termination between the LIN pin and the $V_S$ pin.

# 4. Atmel ATtiny87/ATtiny167 Microcontroller Block for Atmel ATA6616/ATA6617

## 4.1 Features

- High Performance, Low Power AVR 8-bit Microcontroller
- Advanced RISC Architecture
  - 123 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 × 8 General Purpose Working Registers
  - Fully Static Operation
- Non-volatile Program and Data Memories
  - 8 Kbytes/16 Kbytes of In-System Programmable (ISP) Program Memory Flash
    - Endurance: 10,000 Write/Erase Cycles
  - 512 Bytes In-System Programmable EEPROM
    - Endurance: 100,000 Write/Erase Cycles
  - 512 Bytes Internal SRAM
  - Programming Lock for Self-Programming Flash Program and EEPROM Data Security
  - Low Size LIN/UART Software In-System Programmable
- Peripheral Features
  - LIN 2.1 and 1.3 Controller or 8-Bit UART
  - 8-bit Asynchronous Timer/Counter0:
    - 10-bit Clock Prescaler
    - 1 Output Compare or 8-bit PWM Channel
  - 16-bit Synchronous Timer/Counter1:
    - 10-bit Clock Prescaler
    - External Event Counter
    - 2 Output Compares Units or 16-bit PWM Channels each Driving up to 4 Ouput Pins
  - Master/Slave SPI Serial Interface,
  - Universal Serial Interface (USI) with Start Condition Detector (Master/Slave SPI, TWI, ...)
  - 10-bit ADC:
    - 11 Single Ended Channels
    - 8 Differential ADC Channel Pairs with Programmable Gain (8x or 20x)
  - On-chip Analog Comparator with Selectable Voltage Reference
  - 100µA ±10% Current Source (LIN Node Identification)
  - On-chip Temperature Sensor
  - Programmable Watchdog Timer with Separate On-chip Oscillator
- Special Microcontroller Features
  - Dynamic Clock Switching (External/Internal RC/Watchdog Clock) for Power Control, EMC Reduction
  - DebugWIRE On-chip Debug (OCD) System
  - Hardware In-System Programmable (ISP) via SPI Port
  - External and Internal Interrupt Sources
  - Interrupt and Wake-up on Pin Change
  - Low Power Idle, ADC Noise Reduction, and Power-down Modes
  - Enhanced Power-on Reset Circuit
  - Programmable Brown-out Detection Circuit
  - Internal Calibrated RC Oscillator 8MHz
  - 4MHz to 16MHz and 32KHz Crystal/Ceramic Resonator Oscillators

**Table 4-7.** Internal Calibrated RC Oscillator Operating Modes[1]

| Frequency Range[2] (MHz) | CKSEL3..0[3][4] CSEL3..0[5] |
|---|---|
| 7.6 - 8.4 | 0010 |

Notes:
1. If 8MHz frequency exceeds the specification of the device (depends on Vcc), the CKDIV8 fuse can be programmed in order to divide the internal frequency by 8
2. The frequency ranges are guideline values
3. The device is shipped with this CKSEL = “0010”
4. Flash Fuse bits.
5. CLKSELR register bits

When this Oscillator is selected, start-up times are determined by the SUT Fuses or by CSUT field as shown in Table 4-8.

**Table 4-8.** Start-up Times for the Internal Calibrated RC Oscillator Clock Selection

| SUT1..0[1] CSUT1..0[2] | Start-up Time from Power-down/save | Additional Delay from Reset (Vcc = 5.0V) | Recommended Usage |
|---|---|---|---|
| 00[3] | 6 CK | 14CK | BOD enabled |
| 01 | 6 CK | 14CK + 4.1ms | Fast rising power |
| 10[4] | 6 CK | 14CK + 65ms | Slowly rising power |
| 11 | Reserved | | |

Notes:
1. Flash Fuse bits
2. CLKSELR register bits
3. This setting is only available if RSTDISBL fuse is not set
4. The device is shipped with this option selected

#### *4.5.2.3 128 KHz Internal Oscillator*

The 128KHz internal Oscillator is a low power Oscillator providing a clock of 128KHz. The frequency is nominal at 5V and 25°C. This clock may be selected as the system clock by programming CKSEL Fuses or CSEL field as shown in Table 4-5 on page 49.

When this clock source is selected, start-up times are determined by the SUT Fuses or by CSUT field as shown in Table 4-9.

**Table 4-9.** Start-up Times for the 128kHz Internal Oscillator

| SUT1..0[1] CSUT1..0[2] | Start-up Time from Power-down/save | Additional Delay from Reset (Vcc = 5.0V) | Recommended Usage |
|---|---|---|---|
| 00[3] | 6 CK | 14CK | BOD enabled |
| 01 | 6 CK | 14CK + 4.1ms | Fast rising power |
| 10 | 6 CK | 14CK + 65ms | Slowly rising power |
| 11 | Reserved | | |

Notes:
1. Flash Fuse bits
2. CLKSELR register bits
3. This setting is only available if RSTDISBL fuse is not set

### 4.6.5 Power-down Mode

When the SM1..0 bits are written to 10, the SLEEP instruction makes the MCU enter Power-down mode. In this mode, the external Oscillator is stopped, while the external interrupts, the USI start condition, and the Watchdog continue operating (if enabled). Only an External Reset, a Watchdog System Reset, a Watchdog Interrupt, a Brown-out Reset, the USI start condition interrupt, an external level interrupt on INT0 or INT1, or a pin change interrupt can wake up the MCU. This sleep mode basically halts all generated clocks, allowing operation of asynchronous modules only.

Note that if a level triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. Refer to Section 4.9 “External Interrupts” on page 85 for details.

When waking up from Power-down mode, there is a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined by the same CKSEL Fuses that define the Reset Time-out period, as described in Section 4.5.2 “Clock Sources” on page 49.

### 4.6.6 Power-save Mode

When the SM1..0 bits are written to 11, the SLEEP instruction makes the MCU enter Power-save mode. This mode is identical to Power-down, with one exception:

If Timer/Counter0 is clocked asynchronously, i.e., the AS0 bit in ASSR is set, Timer/Counter0 will run during sleep. The device can wake up from either Timer Overflow or Output Compare event from Timer/Counter0 if the corresponding Timer/Counter0 interrupt enable bits are set in TIMSK0, and the global interrupt enable bit in SREG is set.

If the Asynchronous Timer is **NOT** clocked asynchronously, Power-down mode is recommended instead of Power-save mode because the contents of the registers in the asynchronous timer should be considered undefined after wake-up in Power-save mode if AS0 is 0.

This sleep mode basically halts all clocks except $clk_{ASY}$, allowing operation only of asynchronous modules, including Timer/Counter0 if clocked asynchronously.

### 4.6.7 Power Reduction Register

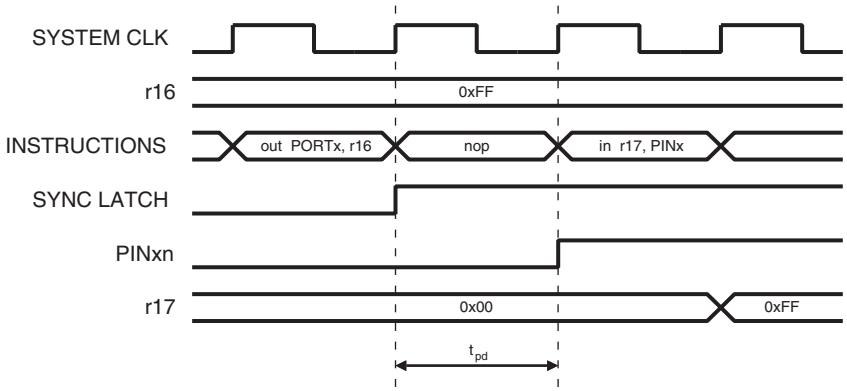
The Power Reduction Register (PRR), see “PRR – Power Reduction Register” on page 70, provides a method to stop the clock to individual peripherals to reduce power consumption. The current state of the peripheral is frozen and the I/O registers can not be read or written. Resources used by the peripheral when stopping the clock will remain occupied, hence the peripheral should in most cases be disabled before stopping the clock. Waking up a module, which is done by clearing the bit in PRR, puts the module in the same state as before shutdown.

Module shutdown can be used in Idle mode and Active mode to significantly reduce the overall power consumption. In all other sleep modes, the clock is already stopped.

Consider the clock period starting shortly after the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the “SYNC LATCH” signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn Register at the succeeding positive clock edge. As indicated by the two arrows tpd,max and tpd,min, a single signal transition on the pin will be delayed between ½ and 1½ system clock period depending upon the time of assertion.

When reading back a software assigned pin value, a nop instruction must be inserted as indicated in Figure 4-28. The out instruction sets the “SYNC LATCH” signal at the positive edge of the clock. In this case, the delay tpd through the synchronizer is 1 system clock period.

**Figure 4-28.** 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.

- **PCINT11/OC1BV – Port B, Bit 3**

PCINT11: Pin Change Interrupt, source 11.

OC1BV: Output Compare and PWM Output B-V for Timer/Counter1. The PB3 pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC1BV pin is also the output pin for the PWM mode timer function (c.f. OC1BV bit of TCCR1D register).

- **PCINT10/OC1AV/USCK/SCL – Port B, Bit 2**

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.

- **PCINT9/OC1BU/DO – Port B, Bit 1**

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).

- **PCINT8/OC1AU/DI/SDA – Port B, Bit 0**

IPCINT8: Pin Change Interrupt, source 8.

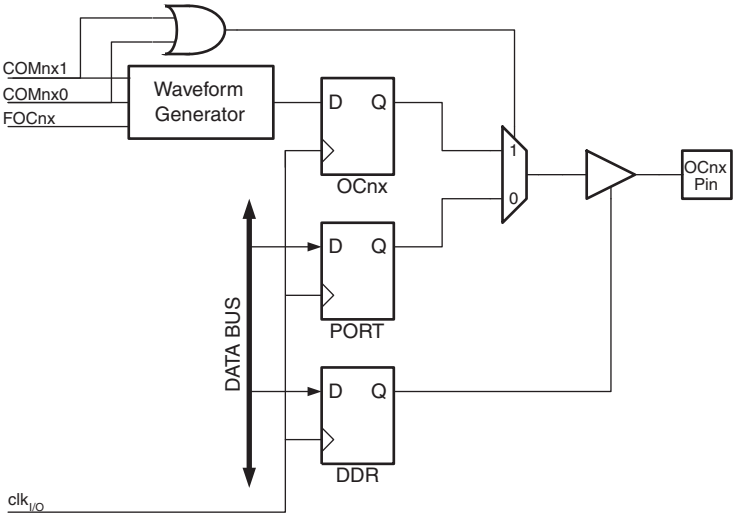
OC1AU: Output Compare and PWM Output A-U for Timer/Counter1. The PB0 pin has to be configured as an output (DDB0 set (one)) to serve this function. The OC1AU pin is also the output pin for the PWM mode timer function (c.f. OC1AU bit of TCCR1D register).

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 4-28 and Table 4-29 relate the alternate functions of Port B to the overriding signals shown in Figure 4-29 on page 97.

**Figure 4-33.** Compare Match Output Logic




### 4.11.6.1 *Compare Output Function*

The general I/O port function is overridden by the Output Compare (OC0A) from the Waveform Generator if either of the COM0A1:0 bits are set. However, the OC0A pin direction (input or output) is still controlled by the Data Direction Register (DDR) for the port pin. The Data Direction Register bit for the OC0A pin (DDR_OC0A) must be set as output before the OC0A value is visible on the pin. The port override function is independent of the Waveform Generation mode.

The design of the Output Compare pin logic allows initialization of the OC0A state before the output is enabled. Note that some COM0A1:0 bit settings are reserved for certain modes of operation. See ”8-bit Timer/Counter Register Description” on page 124.

### 4.11.6.2 *Compare Output Mode and Waveform Generation*

The Waveform Generator uses the COM0A1:0 bits differently in normal, CTC, and PWM modes. For all modes, setting the COM0A1:0 = 0 tells the Waveform Generator that no action on the OC0A Register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to Table 4-30 on page 124. For fast PWM mode, refer to Table 4-31 on page 124, and for phase correct PWM refer to Table 4-32 on page 125.

A change of the COM0A1:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC0A strobe bits.

An interrupt can be generated each time the counter value reaches the TOP value by using the OCF0A flag. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value. However, changing the TOP to a value close to BOTTOM when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to OCR0A is lower than the current value of TCNT0, the counter will miss the compare match. The counter will then have to count to its maximum value (0xFF) and wrap around starting at 0x00 before the compare match can occur.

For generating a waveform output in CTC mode, the OC0A output can be set to toggle its logical level on each compare match by setting the Compare Output mode bits to toggle mode (COM0A1:0 = 1). The OC0A value will not be visible on the port pin unless the data direction for the pin is set to output. The waveform generated will have a maximum frequency of $f_{OC0A} = f_{clk\_I/O}/2$ when OCR0A is set to zero (0x00). The waveform frequency is defined by the following equation:

$$f_{OCnx} = \frac{f_{clk\_I/O}}{2 \times N \times (1 + OCRnx)}$$

The N variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

As for the Normal mode of operation, the TOV0 flag is set in the same timer clock cycle that the counter counts from MAX to 0x00.

#### *4.11.7.3 Fast PWM Mode*

The fast Pulse Width Modulation or fast PWM mode (WGM01:0 = 3) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM option by its single-slope operation. The counter counts from BOTTOM to MAX then restarts from BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC0A) is cleared on the compare match between TCNT0 and OCR0A, and set at BOTTOM. In inverting Compare Output mode, the output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct PWM mode that uses dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (coils, capacitors), and therefore reduces total system cost.

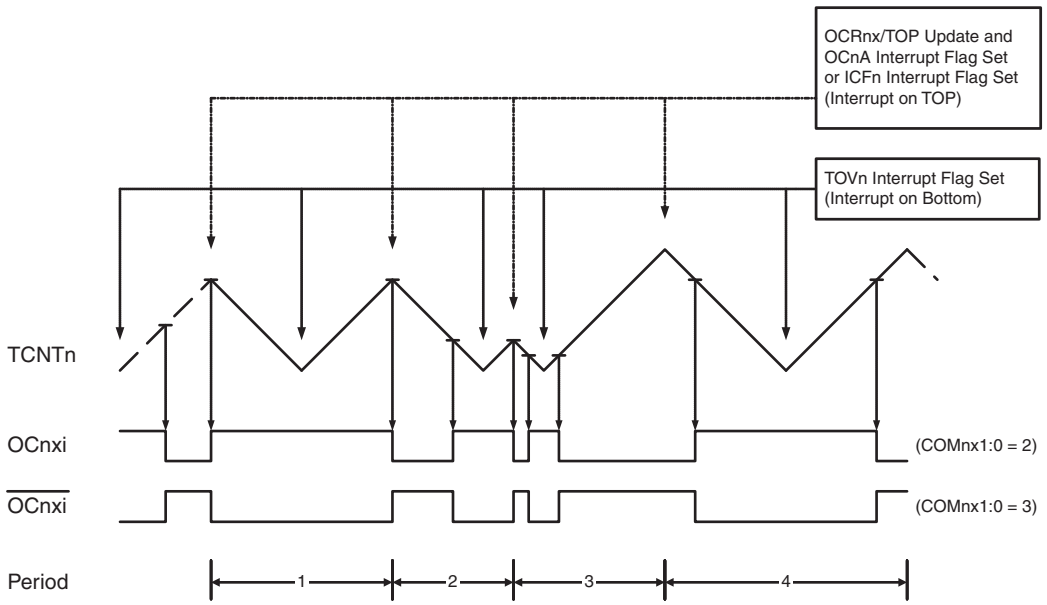
In fast PWM mode, the counter is incremented until the counter value matches the MAX value. The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 4-35. The TCNT0 value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent compare matches between OCR0A and TCNT0.

#### 4.11.7.4 *Phase Correct PWM Mode*

The phase correct PWM mode (WGM01:0 = 1) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to MAX and then from MAX to BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC0A) is cleared on the compare match between TCNT0 and OCR0A while upcounting, and set on the compare match while downcounting. In inverting Output Compare mode, the operation is inverted. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

The PWM resolution for the phase correct PWM mode is fixed to eight bits. In phase correct PWM mode the counter is incremented until the counter value matches MAX. When the counter reaches MAX, it changes the count direction. The TCNT0 value will be equal to MAX for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 4-36. The TCNT0 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent compare matches between OCR0A and TCNT0.

**Figure 4-36.** Phase Correct PWM Mode, Timing Diagram




The Timer/Counter Overflow Flag (TOV0) is set each time the counter reaches BOTTOM. The interrupt flag can be used to generate an interrupt each time the counter reaches the BOTTOM value.

In phase correct PWM mode, the compare unit allows generation of PWM waveforms on the OC0A pin. Setting the COM0A1:0 bits to two will produce a non-inverted PWM. An inverted PWM output can be generated by setting the COM0A1:0 to three (See Table 4-32 on page 125). The actual OC0A value will only be visible on the port pin if the data direction for the port pin is set as output.

### 4.11.11.7 *Timer/Counter0 Interrupt Flag Register – TIFR0*

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | – | – | – | – | – | – | OCF0A | TOV0 | TIFR0 |
| Read/Write | R | R | R | R | R | R | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 7:2 – Res: Reserved Bits**

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

- **Bit 1 – OCF0A: Output Compare Flag 0 A**

The OCF0A bit is set (one) when a compare match occurs between the Timer/Counter0 and the data in OCR0A – Output Compare Register0. OCF0A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0A is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0 (Timer/Counter0 Compare match Interrupt Enable), and OCF0A are set (one), the Timer/Counter0 Compare match Interrupt is executed.

- **Bit 0 – TOV0: Timer/Counter0 Overflow Flag**

The TOV0 bit is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE0A (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set (one), the Timer/Counter0 Overflow interrupt is executed. In PWM mode, this bit is set when Timer/Counter0 changes counting direction at 0x00.

### 4.11.11.8 *General Timer/Counter Control Register – GTCCR*

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | TSM | – | – | – | – | – | PSR0 | PSR1 | GTCCR |
| Read/Write | R/W | R | R | R | R | R | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 1 – PSR0: Prescaler Reset Timer/Counter0**

When this bit is one, the Timer/Counter0 prescaler will be reset. This bit is normally cleared immediately by hardware. If the bit is written when Timer/Counter0 is operating in asynchronous mode, the bit will remain one until the prescaler has been reset. The bit will not be cleared by hardware if the TSM bit is set. Refer to the description of the “Bit 7 – TSM: Timer/Counter Synchronization Mode” on page 132 for a description of the Timer/Counter Synchronization mode.

- **Bit 4 – MSTR: Master/Slave Select**

This bit selects Master SPI mode when written to one, and Slave SPI mode when written logic zero. If $\overline{SS}$ is configured as an input and is driven low while MSTR is set, MSTR will be cleared, and SPIF in SPSR will become set. The user will then have to set MSTR to re-enable SPI Master mode.

- **Bit 3 – CPOL: Clock Polarity**

When this bit is written to one, SCK is high when idle. When CPOL is written to zero, SCK is low when idle. Refer to Figure 4-60 and Figure 4-61 for an example. The CPOL functionality is summarized below:

**Table 4-41.** CPOL Functionality

| CPOL | Leading Edge | Trailing Edge |
|---|---|---|
| 0 | Rising | Falling |
| 1 | Falling | Rising |

- **Bit 2 – CPHA: Clock Phase**

The settings of the Clock Phase bit (CPHA) determine if data is sampled on the leading (first) or trailing (last) edge of SCK. Refer to Figure 4-60 and Figure 4-61 for an example. The CPOL functionality is summarized below:

**Table 4-42.** CPHA Functionality

| CPHA | Leading Edge | Trailing Edge |
|---|---|---|
| 0 | Sample | Setup |
| 1 | Setup | Sample |

- **Bits 1, 0 – SPR1, SPR0: SPI Clock Rate Select 1 and 0**

These two bits control the SCK rate of the device configured as a Master. SPR1 and SPR0 have no effect on the Slave. The relationship between SCK and the $clk_{IO}$ frequency $f_{clkio}$ is shown in the following table:

**Table 4-43.** Relationship Between SCK and the Oscillator Frequency

| SPI2X | SPR1 | SPR0 | SCK Frequency |
|---|---|---|---|
| 0 | 0 | 0 | $f_{clkio}/4$ |
| 0 | 0 | 1 | $f_{clkio}/16$ |
| 0 | 1 | 0 | $f_{clkio}/64$ |
| 0 | 1 | 1 | $f_{clkio}/128$ |
| 1 | 0 | 0 | $f_{clkio}/2$ |
| 1 | 0 | 1 | $f_{clkio}/8$ |
| 1 | 1 | 0 | $f_{clkio}/32$ |
| 1 | 1 | 1 | $f_{clkio}/64$ |

### 4.15.5 Register Descriptions

#### *4.15.5.1 USIDR – USI Data Register*

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | USID7 | USID6 | USID5 | USID4 | USID3 | USID2 | USID1 | USID0 | USIDR |
| 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 – USID7..0: USI Data**

When accessing the USI Data Register (USIDR) the Serial Register can be accessed directly. If a serial clock occurs at the same cycle the register is written, the register will contain the value written and no shift is performed. A (left) shift operation is performed depending of the USICS1..0 bits setting. The shift operation can be controlled by an external clock edge, by a Timer/Counter0 Compare Match, or directly by software using the USICLK strobe bit. Note that even when no wire mode is selected (USIWM1..0 = 0) both the external data input (DI/SDA) and the external clock input (USCK/SCL) can still be used by the USI Data Register.

The output pin in use, DO or SDA depending on the wire mode, is connected via the output latch to the most significant bit (bit 7) of the Data Register. The output latch is open (transparent) during the first half of a serial clock cycle when an external clock source is selected (USICS1 = 1), and constantly open when an internal clock source is used (USICS1 = 0). The output will be changed immediately when a new MSB written as long as the latch is open. The latch ensures that data input is sampled and data output is changed on opposite clock edges.

Note that the corresponding Data Direction Register to the pin must be set to one for enabling data output from the USI Data Register.

#### *4.15.5.2 USIBR – USI Buffer Register*

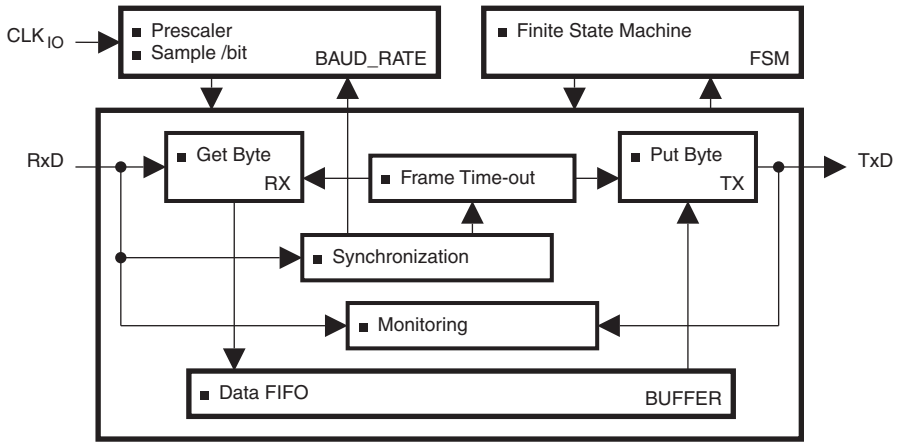
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | USIB7 | USIB6 | USIB5 | USIB4 | USIB3 | USIB2 | USIB1 | USIB0 | USIBR |
| Read/Write | R | R | R | R | R | R | R | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bits 7:0 – USID7..0: USI Buffer**

The content of the Serial Register is loaded to the USI Buffer Register when the transfer is completed, and instead of accessing the USI Data Register (the Serial Register) the USI Data Buffer can be accessed when the CPU reads the received data. This gives the CPU time to handle other program tasks too as the controlling of the USI is not so timing critical. The USI flags as set same as when reading the USIDR register.

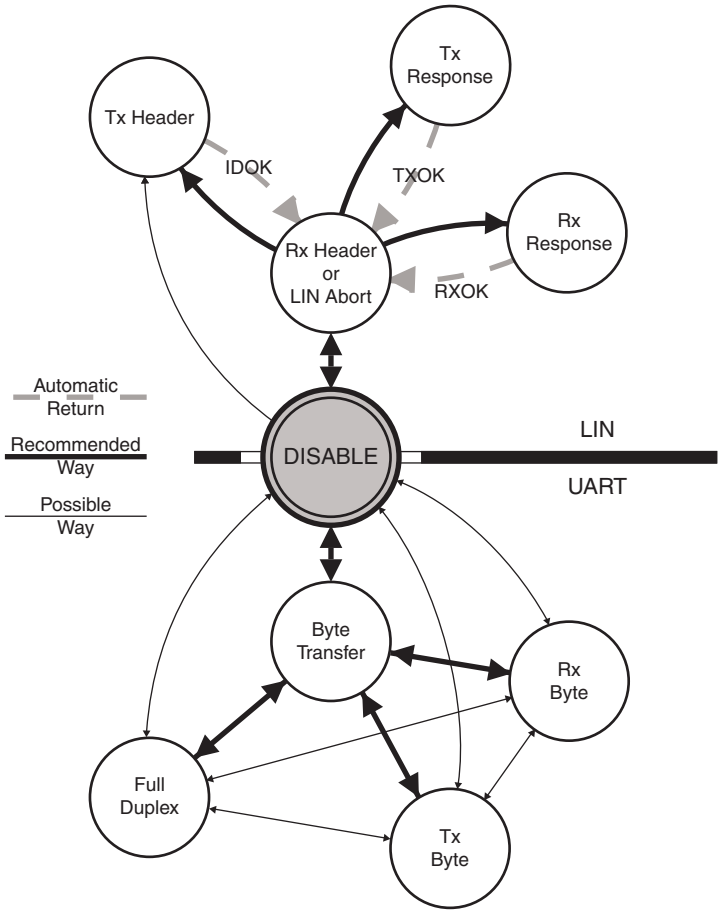
### 4.16.4.3 LIN/UART Controller Structure

**Figure 4-71.** LIN/UART Controller Block Diagram




### 4.16.4.4 LIN/UART Command Overview

**Figure 4-72.** LIN/UART Command Dependencies



### Tx Header Function

In accordance with the LIN protocol, only the master task must enable this function. The header is sent in the appropriate timed slots at the programmed baud rate (c.f. LINBRR & LINBTR registers).

The controller is responsible for:

- The transmission of the BREAK field - 13 dominant bits,
- The transmission of the SYNCH field - character 0x55,
- The transmission of the PROTECTED IDENTIFIER field. It is the full content of the LINIDR register (automatic check bits included).

At the end of this transmission, the controller automatically returns to *Rx Header / LIN Abort* state (i.e. LCMD[1..0] = 00) after setting the appropriate flags. This function leaves the controller in the same setting as after the *Rx Header* function. This means that, in LIN 1.3, the LINDLR register is set with the uncoded length value at the end of the *Tx Header* function.

During this function, the controller is also responsible for:

- The starting of the Frame_Time_Out,
- The checking of the LIN communication integrity.

### Rx and TX Response Functions

These functions are initiated by the slave task of a LIN node. They must be used after sending an header (master task) or after receiving an header (considered as belonging to the slave task). When the TX Response order is sent, the transmission begins. A Rx Response order can be sent up to the reception of the last serial bit of the first byte (before the stop-bit).

In LIN 1.3, the header slot configures the LINDLR register. In LIN 2.1, the user must configure the LINDLR register, either LRXDL[3..0] for *Rx Response* either LTXDL[3..0] for *Tx Response*.

When the command starts, the controller checks the LIN13 bit of the LINCR register to apply the right rule for computing the checksum. Checksum calculation over the DATA bytes and the PROTECTED IDENTIFIER byte is called enhanced checksum and it is used for communication with LIN 2.1 slaves. Checksum calculation over the DATA bytes only is called classic checksum and it is used for communication with LIN 1.3 slaves. Note that identifiers 60 (0x3C) to 63 (0x3F) shall always use classic checksum.

At the end of this reception or transmission, the controller automatically returns to *Rx Header / LIN Abort* state (i.e. LCMD[1..0] = 00) after setting the appropriate flags.

If an LIN error occurs, the reception or the transmission is stopped, the appropriate flags are set and the LIN bus is left to recessive state.

During these functions, the controller is responsible for:

- The initialization of the checksum operator,
- The transmission or the reception of ‘n’ data with the update of the checksum calculation,
- The transmission or the checking of the CHECKSUM field,
- The checking of the Frame_Time_Out,
- The checking of the LIN communication integrity.

#### 4.16.5.8 xxOK Flags

There are three xxOK flags in LINSIR register:

- LIDOK: LIN IDentifier OK
  It is set at the end of the header, either by the Tx Header function or by the Rx Header. In LIN 1.3, before generating LIDOK, the controller updates the LRXDL & LTXDL fields in LINDLR register.
  It is not driven in UART mode.
- LRXOK: LIN RX response complete
  It is set at the end of the response by the Rx Response function in LIN mode and once a character is received in UART mode.
- LTXOK: LIN TX response complete
  It is set at the end of the response by the Tx Response function in LIN mode and once a character has been sent in UART mode.

These flags can generate interrupts if the corresponding enable interrupt bit is set in the LINENIR register (see Section 4.16.5.13 “Interrupts” on page 202).

#### 4.16.5.9 xxERR Flags

LERR bit of the LINSIR register is an logical ‘OR’ of all the bits of LINERR register (see Section 4.16.5.13 “Interrupts” on page 202). There are eight flags:

- LBERR = LIN Bit ERRor.
  A unit that is sending a bit on the bus also monitors the bus. A LIN bit error will be flagged when the bit value that is monitored is different from the bit value that is sent. After detection of a LIN bit error the transmission is aborted.
- LCERR = LIN Checksum ERRor.
  A LIN checksum error will be flagged if the inverted modulo-256 sum of all received data bytes (and the protected identifier in LIN 2.1) added to the checksum does not result in 0xFF.
- LPERR = LIN Parity ERRor (identifier).
  A LIN parity error in the IDENTIFIER field will be flagged if the value of the parity bits does not match with the identifier value. (See LP[1:0] bits in Section 4.16.6.8 “LIN Identifier Register - LINIDR” on page 209). A LIN slave application does not distinguish between corrupted parity bits and a corrupted identifier. The hardware does not undertake any correction. However, the LIN slave application has to solve this as:
  - known identifier (parity bits corrupted),
  - or corrupted identifier to be ignored,
  - or new identifier.
- LSERR = LIN Synchronization ERRor.
  A LIN synchronization error will be flagged if a slave detects the edges of the SYNCH field outside the given tolerance.
- LFERR = LIN Framing ERRor.
  A framing error will be flagged if dominant STOP bit is sampled.
  Same function in UART mode.
- LTOERR = LIN Time Out ERRor.
  A time-out error will be flagged if the MESSAGE frame is not fully completed within the maximum length $T_{Frame\_Maximum}$ by any slave task upon transmission of the SYNCH and IDENTIFIER fields (see Section 4.16.5.10 “Frame Time Out” on page 201).

## 4.23 Electrical Characteristics

Note: All Characteristics contained in this data sheet are based on simulation and characterization of Atmel® ATtiny87/167 AVR® microcontrollers manufactured in a typical process technology. These values are preliminary values representing design targets, and will be updated after characterization of actual Automotive silicon.

### 4.23.1 Absolute Maximum Ratings

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 any 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.

| Parameters | Value | Unit |
|---|---|---|
| Operating Temperature | –40 to +125 | °C |
| Storage Temperature | –65 to +150 | °C |
| Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground | –0.5 to Vcc + 0.5 | V |
| Voltage on $\overline{\text{RESET}}$ with respect to Ground | –0.5 to +13.0 | V |
| Voltage on Vcc with respect to Ground | –0.5 to 6.0 | V |
| DC Current per I/O Pin | 40.0 | mA |
| DC Current Vcc and GND Pins | 200.0 | mA |
| Injection Current at VCC = 0V to 5V(2) | ±5(1) | mA |

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

### 4.23.2 DC Characteristics

$T_A$ = –40°C to +125°C, Vcc = 2.7V to 5.5V (unless otherwise noted)

| Parameters | Test Conditions | Symbol | Min. | Typ.(1) | Max. | Unit |
|---|---|---|---|---|---|---|
| Input Low Voltage | Except XTAL1 and $\overline{\text{RESET}}$ pins | $V_{IL}$ | –0.5 | | 0.2 Vcc(2) | V |
| | XTAL1 pin - External Clock Selected | $V_{IL1}$ | –0.5 | | 0.1 Vcc(2) | V |
| | $\overline{\text{RESET}}$ pin | $V_{IL2}$ | –0.5 | | 0.2 Vcc(2) | V |
| | $\overline{\text{RESET}}$ pin as I/O | $V_{IL3}$ | –0.5 | | 0.2 Vcc(2) | V |
| Input High Voltage | Except XTAL1 and $\overline{\text{RESET}}$ pins | $V_{IH}$ | 0.7 Vcc(3) | | Vcc + 0.5 | V |
| | XTAL1 pin - External Clock Selected | $V_{IH1}$ | 0.8 Vcc(3) | | Vcc + 0.5 | V |
| | $\overline{\text{RESET}}$ pin | $V_{IH2}$ | 0.9 Vcc(3) | | Vcc + 0.5 | V |
| | $\overline{\text{RESET}}$ pin as I/O | $V_{IH3}$ | 0.7 Vcc(3) | | Vcc + 0.5 | V |
| Output Low Voltage(4) (Ports A, B,) | $I_{OL}$ = 10mA, Vcc = 5V<br>$I_{OL}$ = 5mA, Vcc = 3V | $V_{OL}$ | | | 0.6<br>0.5 | V |
| Output High Voltage(5) (Ports A, B) | $I_{OH}$ = –10mA, Vcc = 5V<br>$I_{OH}$ = –5mA, Vcc = 3V | $V_{OH}$ | 4.3<br>2.5 | | | V |
| Input Leakage Current I/O Pin | Vcc = 5.5V, pin low (absolute value) | $I_{IL}$ | | < 0.05 | 1 | µA |
| Input Leakage Current I/O Pin | Vcc = 5.5V, pin high (absolute value) | $I_{IH}$ | | < 0.05 | 1 | µA |
| Reset Pull-up Resistor | | $R_{RST}$ | 30 | | 60 | kΩ |
| I/O Pin Pull-up Resistor | | $R_{pu}$ | 20 | | 50 | kΩ |

**Figure 4-116.** Active Supply Current versus Frequency (≥ 1MHz)

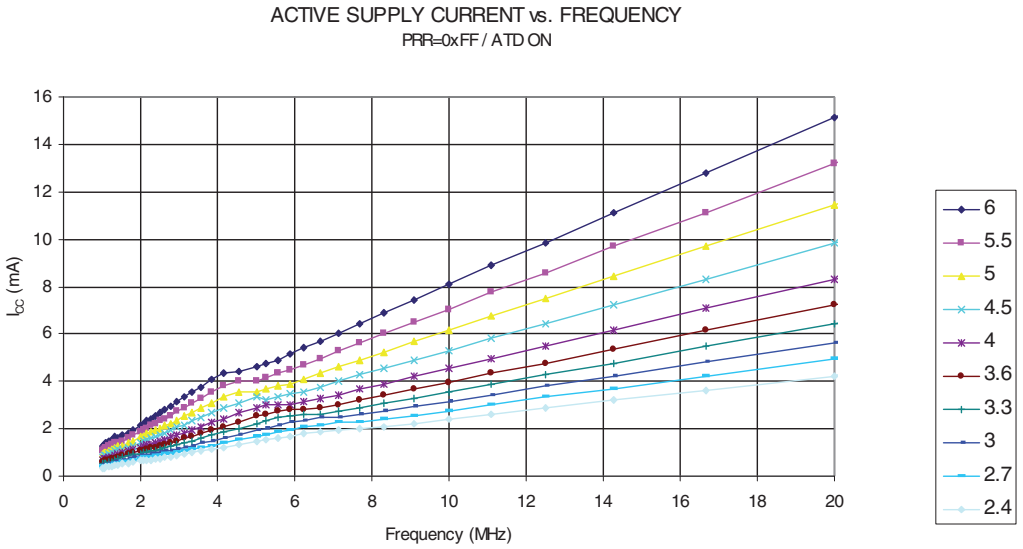
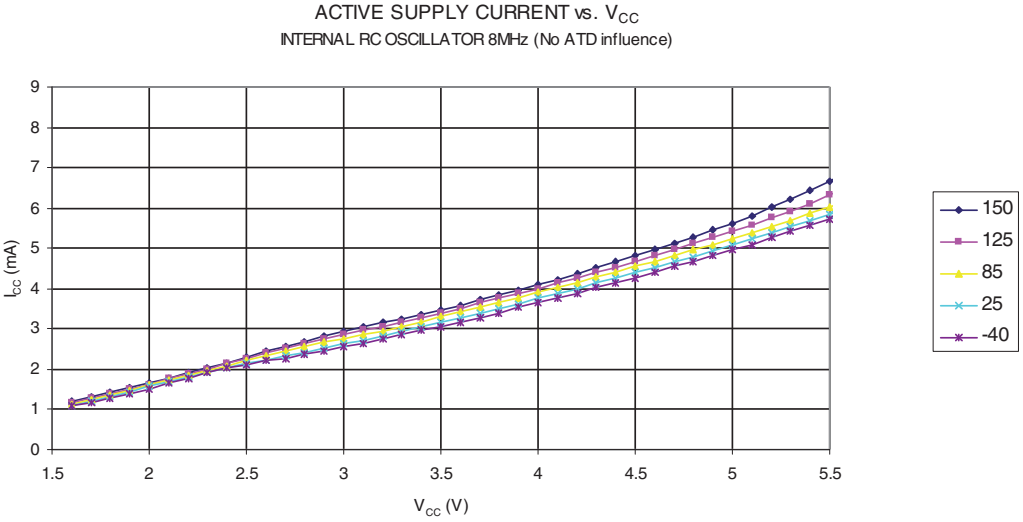



**Figure 4-117.** Active Supply Current versus $V_{CC}$ (Internal RC Oscillator, 8MHz)



## 4.26 Register Summary (Continued)

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x08 (0x28) | Reserved | | | | | | | | | |
| 0x07 (0x27) | Reserved | | | | | | | | | |
| 0x06 (0x26) | Reserved | | | | | | | | | |
| 0x05 (0x25) | PORTB | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | page 109 |
| 0x04 (0x24) | DDRB | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | page 109 |
| 0x03 (0x23) | PINB | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | page 109 |
| 0x02 (0x22) | PORTA | PORTA7 | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | page 109 |
| 0x01 (0x21) | DDRA | DDA7 | DDA6 | DDA5 | DDA4 | DDA3 | DDA2 | DDA1 | DDA0 | page 109 |
| 0x00 (0x20) | PINA | PINA7 | PINA6 | PINA5 | PINA4 | PINA3 | PINA2 | PINA1 | PINA0 | page 109 |

Notes:
1. Address bits exceeding EEAMSB (Table 4-74 on page 253) are don't care.
2. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
3. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
4. Some of the status flags are cleared by writing a logical one to them. Note that, unlike most other AVR®s, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
5. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The Atmel® ATtiny87/167 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

## 4.27 Instruction Set Summary (Continued)

| Mnemonics | Operands | Description | Operation | Flags | #Clocks |
|---|---|---|---|---|---|
| BRCS | k | Branch if Carry Set | if (C = 1) then PC ←PC + k + 1 | None | 1/2 |
| BRCC | k | Branch if Carry Cleared | if (C = 0) then PC ←PC + k + 1 | None | 1/2 |
| BRSH | k | Branch if Same or Higher | if (C = 0) then PC ←PC + k + 1 | None | 1/2 |
| BRLO | k | Branch if Lower | if (C = 1) then PC ←PC + k + 1 | None | 1/2 |
| BRMI | k | Branch if Minus | if (N = 1) then PC ←PC + k + 1 | None | 1/2 |
| BRPL | k | Branch if Plus | if (N = 0) then PC ←PC + k + 1 | None | 1/2 |
| BRGE | k | Branch if Greater or Equal, Signed | if (N ⊕ V= 0) then PC ←PC + k + 1 | None | 1/2 |
| BRLT | k | Branch if Less Than Zero, Signed | if (N ⊕ V= 1) then PC ←PC + k + 1 | None | 1/2 |
| BRHS | k | Branch if Half Carry Flag Set | if (H = 1) then PC ←PC + k + 1 | None | 1/2 |
| BRHC | k | Branch if Half Carry Flag Cleared | if (H = 0) then PC ←PC + k + 1 | None | 1/2 |
| BRTS | k | Branch if T Flag Set | if (T = 1) then PC ←PC + k + 1 | None | 1/2 |
| BRTC | k | Branch if T Flag Cleared | if (T = 0) then PC ←PC + k + 1 | None | 1/2 |
| BRVS | k | Branch if Overflow Flag is Set | if (V = 1) then PC ←PC + k + 1 | None | 1/2 |
| BRVC | k | Branch if Overflow Flag is Cleared | if (V = 0) then PC ←PC + k + 1 | None | 1/2 |
| BRIE | k | Branch if Interrupt Enabled | if (I = 1) then PC ←PC + k + 1 | None | 1/2 |
| BRID | k | Branch if Interrupt Disabled | if (I = 0) then PC ←PC + k + 1 | None | 1/2 |
| **BIT AND BIT-TEST INSTRUCTIONS** | | | | | |
| SBI | P,b | Set Bit in I/O Register | I/O(P,b) ←1 | None | 2 |
| CBI | P,b | Clear Bit in I/O Register | I/O(P,b) ←0 | None | 2 |
| LSL | Rd | Logical Shift Left | Rd(n+1) ←Rd(n), Rd(0) ←0 | Z,C,N,V | 1 |
| LSR | Rd | Logical Shift Right | Rd(n) ←Rd(n+1), Rd(7) ←0 | Z,C,N,V | 1 |
| ROL | Rd | Rotate Left Through Carry | Rd(0)←C,Rd(n+1)←Rd(n),C←Rd(7) | Z,C,N,V | 1 |
| ROR | Rd | Rotate Right Through Carry | Rd(7)←C,Rd(n)←Rd(n+1),C←Rd(0) | Z,C,N,V | 1 |
| ASR | Rd | Arithmetic Shift Right | Rd(n) ←Rd(n+1), n=0..6 | Z,C,N,V | 1 |
| SWAP | Rd | Swap Nibbles | Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0) | None | 1 |
| BSET | s | Flag Set | SREG(s) ←1 | SREG(s) | 1 |
| BCLR | s | Flag Clear | SREG(s) ←0 | SREG(s) | 1 |
| BST | Rr, b | Bit Store from Register to T | T ←Rr(b) | T | 1 |
| BLD | Rd, b | Bit load from T to Register | Rd(b) ←T | None | 1 |
| SEC | | Set Carry | C ←1 | C | 1 |
| CLC | | Clear Carry | C ←0 | C | 1 |
| SEN | | Set Negative Flag | N ←1 | N | 1 |
| CLN | | Clear Negative Flag | N ←0 | N | 1 |
| SEZ | | Set Zero Flag | Z ←1 | Z | 1 |
| CLZ | | Clear Zero Flag | Z ←0 | Z | 1 |
| SEI | | Global Interrupt Enable | I ←1 | I | 1 |
| CLI | | Global Interrupt Disable | I ←0 | I | 1 |
| SES | | Set Signed Test Flag | S ←1 | S | 1 |
| CLS | | Clear Signed Test Flag | S ←0 | S | 1 |
| SEV | | Set Twos Complement Overflow. | V ←1 | V | 1 |
| CLV | | Clear Twos Complement Overflow | V ←0 | V | 1 |
| SET | | Set T in SREG | T ←1 | T | 1 |
| CLT | | Clear T in SREG | T ←0 | T | 1 |
| SEH | | Set Half Carry Flag in SREG | H ←1 | H | 1 |
| CLH | | Clear Half Carry Flag in SREG | H ←0 | H | 1 |


***Atmel Corporation***
2325 Orchard Parkway
San Jose, CA 95131
USA
**Tel:** (+1)(408) 441-0311
**Fax:** (+1)(408) 487-2600

***Atmel Asia Limited***
Unit 01-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
**Tel:** (+852) 2245-6100
**Fax:** (+852) 2722-1369

***Atmel Munich GmbH***
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
**Tel:** (+49) 89-31970-0
**Fax:** (+49) 89-3194621

***Atmel Japan***
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
JAPAN
**Tel:** (+81) (3) 3523-3551
**Fax:** (+81) (3) 3523-7581