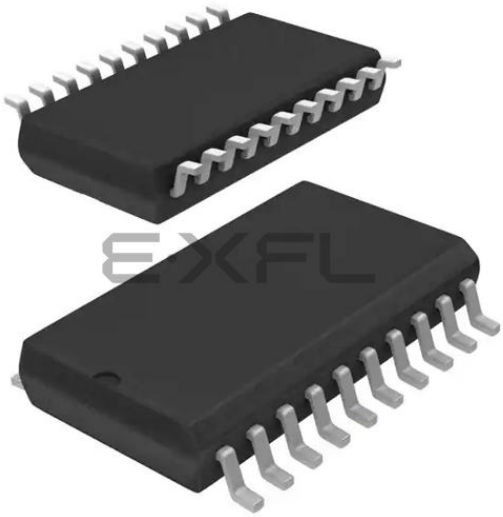


**Welcome to E-XFL.COM**

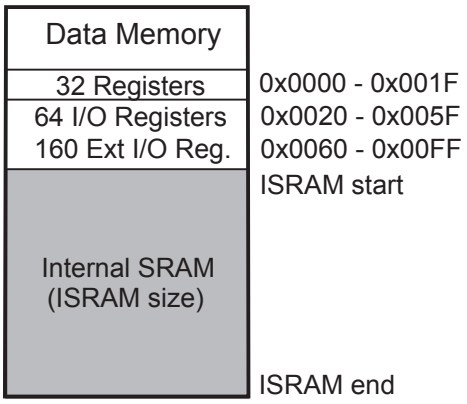
## What is "Embedded - Microcontrollers"?

"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 | 20-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | 20-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/attiny167-su |

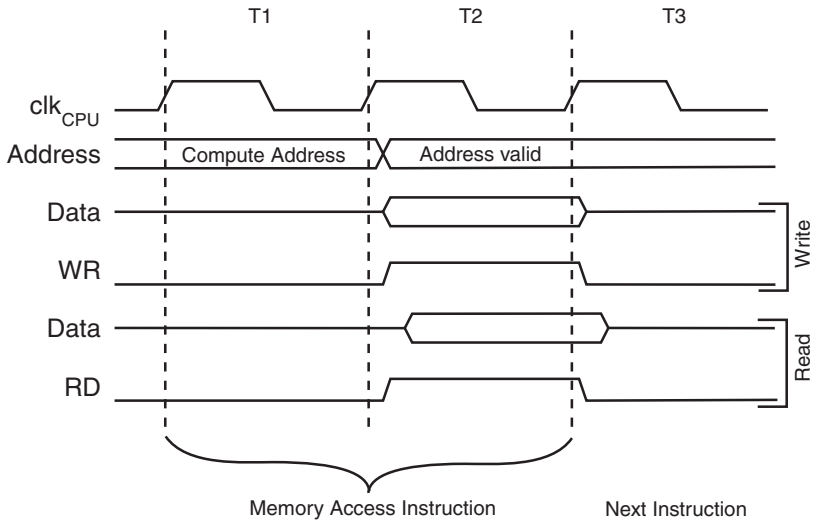
**Figure 3-2.** Data Memory Map




### 3.2.1 Data Memory Access Times

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two $clk_{CPU}$ cycles as described in Figure 3-3.

**Figure 3-3.** On-chip Data SRAM Access Cycles




## 3.3 EEPROM Data Memory

The ATtiny87/167 contains EEPROM memory (see “E2 size” in Table 3-1 on page 15). It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described in the following, specifying the EEPROM Address Registers, the EEPROM Data Register and the EEPROM Control Register.

Section 21. “Memory Programming” on page 224 contains a detailed description on EEPROM programming in SPI or Parallel Programming mode.

### 6.3.3 WDTCR – Watchdog Timer Control Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| (0x60) | WDIF | WDIE | WDP3 | WDCE | WDE | WDP2 | WDP1 | WDP0 | WDTCR |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | X | 0 | 0 | 0 | |

- **Bit 7 – WDIF: Watchdog Interrupt Flag**

This bit is set when a time-out occurs in the Watchdog Timer and the Watchdog Timer is configured for interrupt. WDIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, WDIF is cleared by writing a logic one to the flag. When the I-bit in SREG and WDIE are set, the Watchdog Time-out Interrupt is executed.

- **Bit 6 – WDIE: Watchdog Interrupt Enable**

When this bit is written to one and the I-bit in the Status Register is set, the Watchdog Interrupt is enabled. If WDE is cleared in combination with this setting, the Watchdog Timer is in Interrupt Mode, and the corresponding interrupt is executed if time-out in the Watchdog Timer occurs.

If WDE is set, the Watchdog Timer is in Interrupt and System Reset Mode. The first time-out in the Watchdog Timer will set WDIF. Executing the corresponding interrupt vector will clear WDIE and WDIF automatically by hardware (the Watchdog goes to System Reset Mode). This is useful for keeping the Watchdog Timer security while using the interrupt. To stay in Interrupt and System Reset Mode, WDIE must be set after each interrupt. This should however not be done within the interrupt service routine itself, as this might compromise the safety-function of the Watchdog System Reset mode. If the interrupt is not executed before the next time-out, a System Reset will be applied.

If the Watchdog Timer is used as clock monitor (c.f. Section • “Bits 3:0 – CLKC[3:0]: Clock Control Bits 3 - 0” on page 40), the System Reset Mode is enabled and the Interrupt Mode is automatically disabled.

**Table 6-1.** Watchdog Timer Configuration

| Clock Monitor | WDTON | WDE | WDIE | Mode | Action on Time-out |
|---|---|---|---|---|---|
| x | 0 | 0 | 0 | Stopped | None |
| On | y(1) | y(1) | y(1) | System Reset Mode | Reset |
| Off | 0 | 0 | 1 | Interrupt Mode | Interrupt |
| | 0 | 1 | 0 | System Reset Mode | Reset |
| | 0 | 1 | 1 | Interrupt and System Reset Mode | Interrupt, then go to System Reset Mode |
| | 1 | x | x | System Reset Mode | Reset |

Note: 1. At least one of these three enables (WDTON, WDE & WDIE) equal to 1.

- **Bit 4 – WDCE: Watchdog Change Enable**

This bit is used in timed sequences for changing WDE and prescaler bits. To clear the WDE bit, and/or change the prescaler bits, WDCE must be set.

Once written to one, hardware will clear WDCE after four clock cycles.

- **Bit 3 – WDE: Watchdog System Reset Enable**

### 9.3.1 MCUCR – MCU Control Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| 0x35 (0x55) | – | BODS | BODSE | PUD | – | – | – | – | MCUCR |
| Read/Write | R | R/W | R/W | R/W | R | R | R | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 4 – PUD: Pull-up Disable**

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0, 1). See “Configuring the Pin” on page 68 for more details about this feature.

### 9.3.2 PORTCR – Port Control Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| 0x12 (0x32) | – | – | BBMB | BBMA | – | – | PUDB | PUDA | PORTCR |
| Read/Write | R | R | R/W | R/W | R | R | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bits 5:4 – BBMx: Break-Before-Make Mode Enable**

When these bits are written to one, the port-wise Break-Before-Make mode is activated. The intermediate tri-state cycle is then inserted when writing DDRxn to make an output. For further information, see “Break-Before-Make Switching” on page 69.

- **Bits 1:0 – PUDx: Port-Wise Pull-up Disable**

When these bits are written to one, the port-wise pull-ups in the defined I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0, 1). The Port-Wise Pull-up Disable bits are ORed with the global Pull-up Disable bit (PUD) from the MCUCR register. See “Configuring the Pin” on page 68 for more details about this feature.

- 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.

- $\overline{\text{RESET}}$: Reset input pin. When the RSTDISBL Fuse is programmed, this pin functions as a normal I/O pin, and the part will have to rely on Power-on Reset and Brown-out Reset as its reset sources. When the RSTDISBL Fuse is unprogrammed, the reset circuitry is connected to the pin, and the pin can not be used as an I/O pin.
  If PB7 is used as a reset pin, DDB7, PORTB7 and PINB7 will all read 0.
- dW: When the debugWIRE Enable (DWEN) Fuse is programmed and Lock bits are unprogrammed, the RESET port pin is configured as a wire-AND (open-drain) bi-directional I/O pin with pull-up enabled and becomes the communication gateway between target and emulator.

- **Port B, Bit 6 – PCINT14/ADC9/OC1AX/INT0**
  - PCINT14: Pin Change Interrupt, source 14.
  - ADC9: Analog to Digital Converter, channel 9.
  - OC1AX: Output Compare and PWM Output A-X for Timer/Counter1. The PB6 pin has to be configured as an output (DDB6 set (one)) to serve this function. The OC1AX pin is also the output pin for the PWM mode timer function (c.f. OC1AX bit of TCCR1D register).
  - INT0: External Interrupt0 Input. The PB6 pin can serve as an external interrupt source.

- **Port B, Bit 5 – PCINT13/ADC8/OC1BW/XTAL2/CLKO**
  - PCINT13: Pin Change Interrupt, source 13.
  - ADC8: Analog to Digital Converter, channel 8.
  - OC1BW: Output Compare and PWM Output B-W for Timer/Counter1. The PB5 pin has to be configured as an output (DDB5 set (one)) to serve this function. The OC1BW pin is also the output pin for the PWM mode timer function (c.f. OC1BW bit of TCCR1D register).
  - XTAL2: Chip clock Oscillator pin 2. Used as clock pin for crystal Oscillator or Low-frequency crystal Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.
  - CLKO: Divided system clock output. The divided system clock can be output on the PB5 pin. The divided system clock will be output if the CKOUT Fuse is programmed, regardless of the PORTB5 and DDB5 settings. It will also be output during reset.

- **Port B, Bit 4 – PCINT12/OC1AW/XTAL1/CLKI**
  - PCINT12: Pin Change Interrupt, source 12.
  - OC1AW: Output Compare and PWM Output A-W for Timer/Counter1. The PB4 pin has to be configured as an output (DDB4 set (one)) to serve this function. The OC1AW pin is also the output pin for the PWM mode timer function (c.f. OC1AW bit of TCCR1D register).
  - XTAL1: Chip clock Oscillator pin 1. Used for all chip clock sources except internal calibrated RC Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.
  - CLKI: External clock input. When used as a clock pin, the pin can not be used as an I/O pin.

  Note: If PB4 is used as a clock pin (XTAL1 or CLKI), DDB4, PORTB4 and PINB4 will all read 0.

- **Port B, Bit 3 – PCINT11/OC1BV**
  - 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).

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

The following code examples show how to do an atomic write of the TCNT1 Register contents. Writing any of the OCR1A/B or ICR1 Registers can be done by using the same principle.

Assembly Code Example[1]

```
TIM16_WriteTCNT1:
  ; Save global interrupt flag
  in     r18,SREG
  ; Disable interrupts
  cli
  ; Set TCNT1 to r17:r16
  sts    TCNT1H,r17
  sts    TCNT1L,r16
  ; Restore global interrupt flag
  out    SREG,r18
  ret
```

C Code Example[1]

```
void TIM16_WriteTCNT1(unsigned int i)
{
  unsigned char sreg;
  unsigned int i;
  /* Save global interrupt flag */
  sreg = SREG;
  /* Disable interrupts */
  _CLI();
  /* Set TCNT1 to i */
  TCNT1 = i;
  /* Restore global interrupt flag */
  SREG = sreg;
}
```

Note: 1. The example code assumes that the part specific header file is included.

The assembly code example requires that the r17:r16 register pair contains the value to be written to TCNT1.

### 12.3.2 Reusing the Temporary High Byte Register

If writing to more than one 16-bit register where the high byte is the same for all registers written, then the high byte only needs to be written once. However, note that the same rule of atomic operation described previously also applies in this case.

## 12.4 Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the Clock Select logic which is controlled by the Clock Select (CS1[2:0]) bits located in the Timer/Counter control Register B (TCCR1B). For details on clock sources and prescaler, see “Timer/Counter1 Prescaler” on page 106.

Table 12-2 shows the COM1A/B[1:0] bit functionality when the WGM1[3:0] bits are set to the fast PWM mode.

**Table 12-2.** Compare Output Mode, Fast PWM (1)

| OC1Ai OC1Bi | COM1A1 COM1B1 | COM1A0 COM1B0 | Description |
|---|---|---|---|
| 0 | x | x | Normal port operation, OC1A/OC1B disconnected. |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | WGM13=0: Normal port operation, OC1A/OC1B disconnected.<br>WGM13=1: Toggle OC1A on Compare Match, OC1B reserved. |
| 1 | 1 | 0 | Clear OC1A/OC1B on Compare Match<br>Set OC1A/OC1B at TOP |
| 1 | 1 | 1 | Set OC1A/OC1B on Compare Match<br>Clear OC1A/OC1B at TOP |

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. In this case the compare match is ignored, but the set or clear is done at TOP. See “Fast PWM Mode” on page 123. for more details.

Table 12-3 shows the COM1A/B[1:0] bit functionality when the WGM1[3:0] bits are set to the phase correct or the phase and frequency correct, PWM mode.

**Table 12-3.** Compare Output Mode, Phase Correct and Phase and Frequency Correct PWM(1)

| OC1Ai OC1Bi | COM1A1 COM1B1 | COM1A0 COM1B0 | Description |
|---|---|---|---|
| 0 | x | x | Normal port operation, OC1A/OC1B disconnected. |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | WGM13=0: Normal port operation, OC1A/OC1B disconnected.<br>WGM13=1: Toggle OC1A on Compare Match, OC1B reserved. |
| 1 | 1 | 0 | Clear OC1A/OC1B on Compare Match when up-counting.<br>Set OC1A/OC1B on Compare Match when downcounting. |
| 1 | 1 | 1 | Set OC1A/OC1B on Compare Match when up-counting.<br>Clear OC1A/OC1B on Compare Match when downcounting. |

Note: 1. A special case occurs when OC1A/OC1B equals TOP and COM1A1/COM1B1 is set. See “Phase Correct PWM Mode” on page 125. for more details.

- **Bits 3:2 – Res: Reserved Bits**

These bits are reserved for future use.

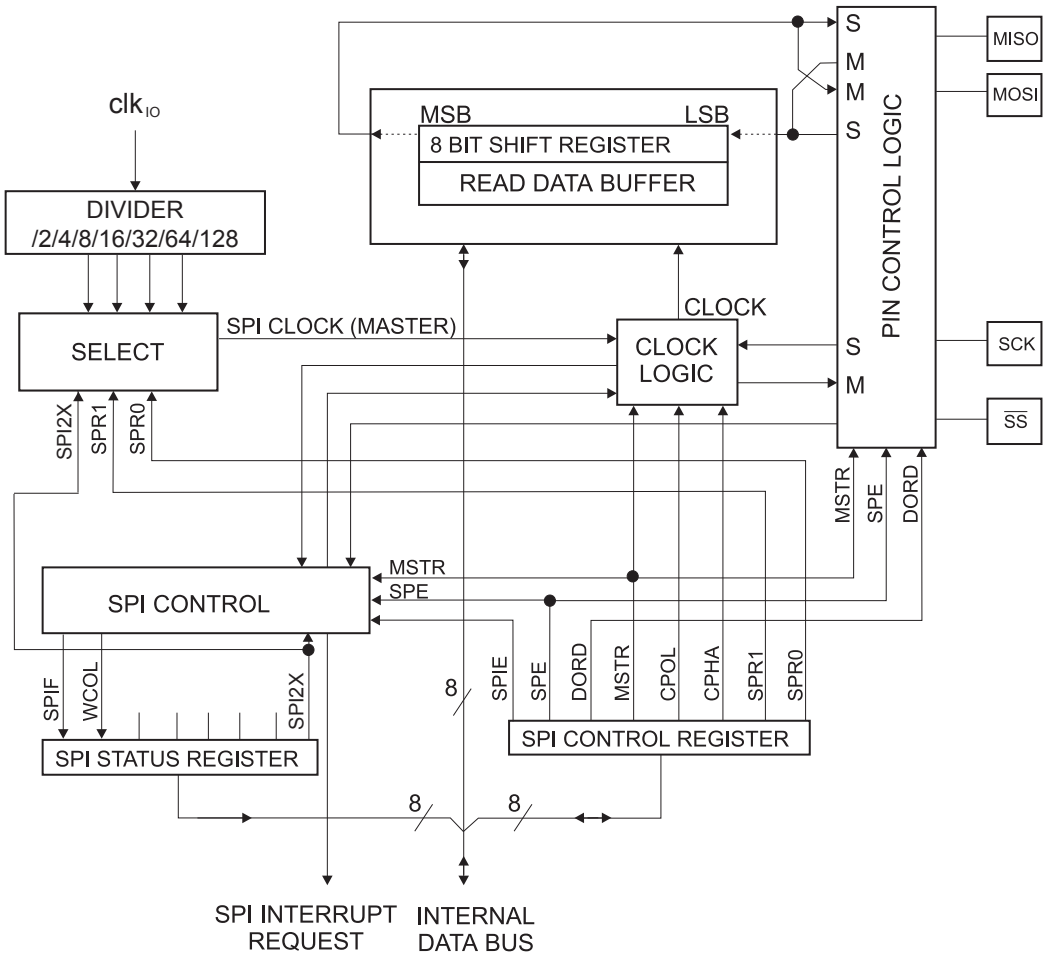
# 13. SPI - Serial Peripheral Interface

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the ATtiny87/167 and peripheral devices or between several AVR devices. The ATtiny87/167 SPI includes the following features:

## 13.1 Features

- **Full-duplex, Three-wire Synchronous Data Transfer**
- **Master or Slave Operation**
- **LSB First or MSB First Data Transfer**
- **Seven Programmable Bit Rates**
- **End of Transmission Interrupt Flag**
- **Write Collision Flag Protection**
- **Wake-up from Idle Mode**
- **Double Speed (CK/2) Master SPI Mode**

**Figure 13-1.** SPI Block Diagram[1]




Note: 1. Refer to Figure 1.4 on page 4, and Table 9-3 on page 76 for SPI pin placement.

### 15.3.3 Data Transport

Two types of data may be transported in a frame; signals or diagnostic messages.

- Signals
  Signals are scalar values or byte arrays that are packed into the data field of a frame. A signal is always present at the same position in the data field for all frames with the same identifier.
- Diagnostic messages
  Diagnostic messages are transported in frames with two reserved identifiers. The interpretation of the data field depends on the data field itself as well as the state of the communicating nodes.

### 15.3.4 Schedule Table

The master task (in the master node) transmits frame headers based on a schedule table. The schedule table specifies the identifiers for each header and the interval between the start of a frame and the start of the following frame. The master application may use different schedule tables and select among them.

### 15.3.5 Compatibility with LIN 1.3

LIN 2.1 is a super-set of LIN 1.3.

A LIN 2.1 master node can handle clusters consisting of both LIN 1.3 slaves and/or LIN 2.1 slaves. The master will then avoid requesting the new LIN 2.1 features from a LIN 1.3 slave:

- Enhanced checksum,
- Re-configuration and diagnostics,
- Automatic baud rate detection,
- "Response error" status monitoring.

LIN 2.1 slave nodes can not operate with a LIN 1.3 master node (e.g. the LIN1.3 master does not support the enhanced checksum).

The LIN 2.1 physical layer is backwards compatible with the LIN1.3 physical layer. But not the other way around. The LIN 2.1 physical layer sets greater requirements, i.e. a master node using the LIN 2.1 physical layer can operate in a LIN 1.3 cluster.

## 15.4 LIN / UART Controller

The LIN/UART controller is divided in three main functions:

- Tx LIN Header function,
- Rx LIN Header function,
- LIN Response function.

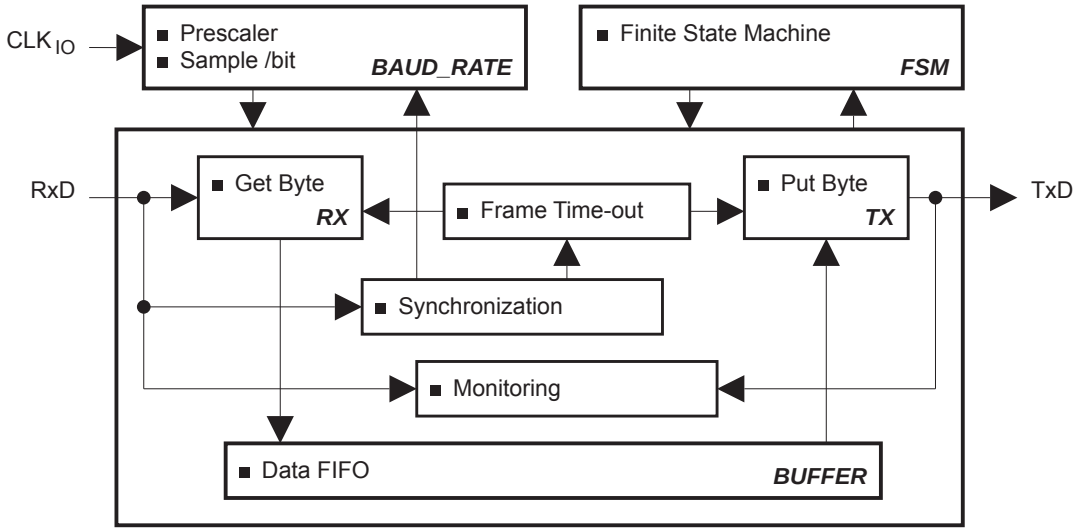
These functions mainly use two services:

- Rx service,
- Tx service.

Because these two services are basically UART services, the controller is also able to switch into an UART function.

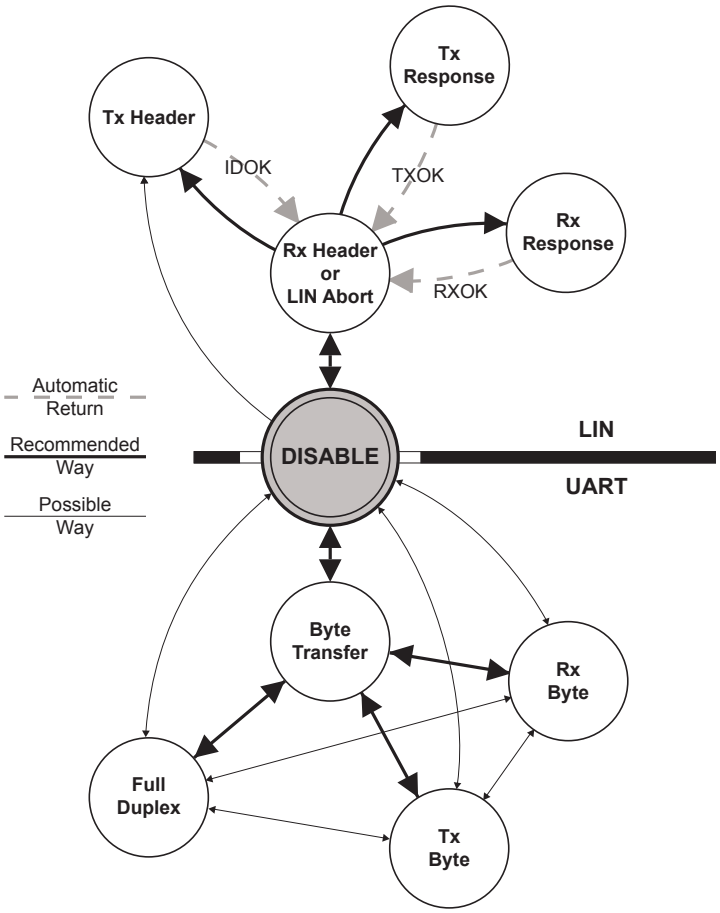
### 15.4.3 LIN/UART Controller Structure

**Figure 15-4.** LIN/UART Controller Block Diagram




### 15.4.4 LIN/UART Command Overview

**Figure 15-5.** LIN/UART Command Dependencies



When the busy signal is set, some registers are locked, user writing is not allowed:

- “LIN Control Register” - LINCR - except LCMD[2:0], LENA & LSWRES,
- “LIN Baud Rate Registers” - LINBRRL & LINBRRH,
- “LIN Data Length Register” - LINDLR,
- “LIN Identifier Register” - LINIDR,
- “LIN Data Register” - LINDAT.

If the busy signal is set, the only available commands are:

- LCMD[1:0] = 00 $_b$, the abort command is taken into account at the end of the byte,
- LENA = 0 and/or LCMD[2] = 0, the kill command is taken into account immediately,
- LSWRES = 1, the reset command is taken into account immediately.

Note that, if another command is entered during busy signal, the new command is not validated and the LOVRERR bit flag of the LINERR register is set. The on-going transfer is not interrupted.

#### *15.5.5.2 Busy Signal in UART Mode*

During the byte transmission, the busy signal is set. This locks some registers from being written:

- “LIN Control Register” - LINCR - except LCMD[2:0], LENA & LSWRES,
- “LIN Data Register” - LINDAT.

The busy signal is not generated during a byte reception.

### 15.5.6 Bit Timing

#### *15.5.6.1 Baud rate Generator*

The baud rate is defined to be the transfer rate in bits per second (bps):

- BAUD: Baud rate (in bps),
- $fclk_{i/o}$: System I/O clock frequency,
- LDIV[11:0]: Contents of LINBRRH & LINBRRL registers - (0-4095), the pre-scaler receives $clk_{i/o}$ as input clock.
- LBT[5:0]: Least significant bits of - LINBTR register- (0-63) is the number of samplings in a LIN or UART bit (default value 32).

Equation for calculating baud rate:

$$BAUD = fclk_{i/o} / LBT[5:0] \times (LDIV[11:0] + 1)$$

Equation for setting LINDIV value:

$$LDIV[11:0] = ( fclk_{i/o} / LBT[5:0] \times BAUD ) - 1$$

Note that in reception a majority vote on three samplings is made.

#### *15.5.6.2 Re-synchronization in LIN Mode*

When waiting for Rx Header, LBT[5:0] = 32 in LINBTR register. The re-synchronization begins when the BREAK is detected. If the BREAK size is not in the range (10.5 bits min., 28 bits max. — 13 bits nominal), the BREAK is refused. The re-synchronization is done by adjusting LBT[5:0] value to the SYNCH field of the received header (0x55). Then the PROTECTED IDENTIFIER is sampled using the new value of LBT[5:0].

# 17. ADC – Analog to Digital Converter

## 17.1 Features

- 10-bit Resolution
- 1.0 LSB Integral Non-linearity
- ± 2 LSB Absolute Accuracy
- 13 - 260 µs Conversion Time (Low - High Resolution)
- Up to 15 kSPS at Maximum Resolution
- 11 Multiplexed Single Ended Input Channels
- 8 Differential input pairs with selectable gain
- Temperature sensor input channel
- Voltage from Internal Current Source Driving (ISRC)
- Optional Left Adjustment for ADC Result Readout
- 0 - $AV_{CC}$ ADC Input Voltage Range
- Selectable 1.1V / 2.56V ADC Voltage Reference
- Free Running or Single Conversion Mode
- ADC Start Conversion by Auto Triggering on Interrupt Sources
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler
- Unipolar / Bipolar Input Mode
- Input Polarity Reversal Mode

## 17.2 Overview

The ATtiny87/167 features a 10-bit successive approximation ADC. The ADC is connected to a 11-channel Analog Multiplexer which allows 16 differential voltage input combinations and 11 single-ended voltage inputs constructed from the pins PA[7:0] or PB[7:4]. The differential input is equipped with a programmable gain stage, providing amplification steps of 8x or 20x on the differential input voltage before the A/D conversion. The single-ended voltage inputs refer to 0V (AGND).

The ADC contains a Sample and Hold circuit which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown in Figure 17-1.

Internal reference voltages of nominally 1.1V or 2.56V are provided On-chip. Alternatively, $AV_{CC}$ can be used as reference voltage for single ended channels. There are also options to output the internal 1.1V or 2.56V reference voltages or to input an external voltage reference and turn-off the internal voltage reference. These options are selected using the REFS[1:0] bits of the ADMUX control register and using AREFEN and XREFEN bits of the AMISCR control register.

### 20.1.3 Performing a Page Write

To execute Page Write, set up the address in the Z-pointer, write “00000101 $_b$” to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE. Other bits in the Z-pointer must be written to zero during this operation.

- The CPU is halted during the Page Write operation.

## 20.2 Addressing the Flash During Self-Programming

The Z-pointer is used to address the SPM commands. The Z pointer consists of the Z-registers ZL and ZH in the register file. The number of bits actually used is implementation dependent.

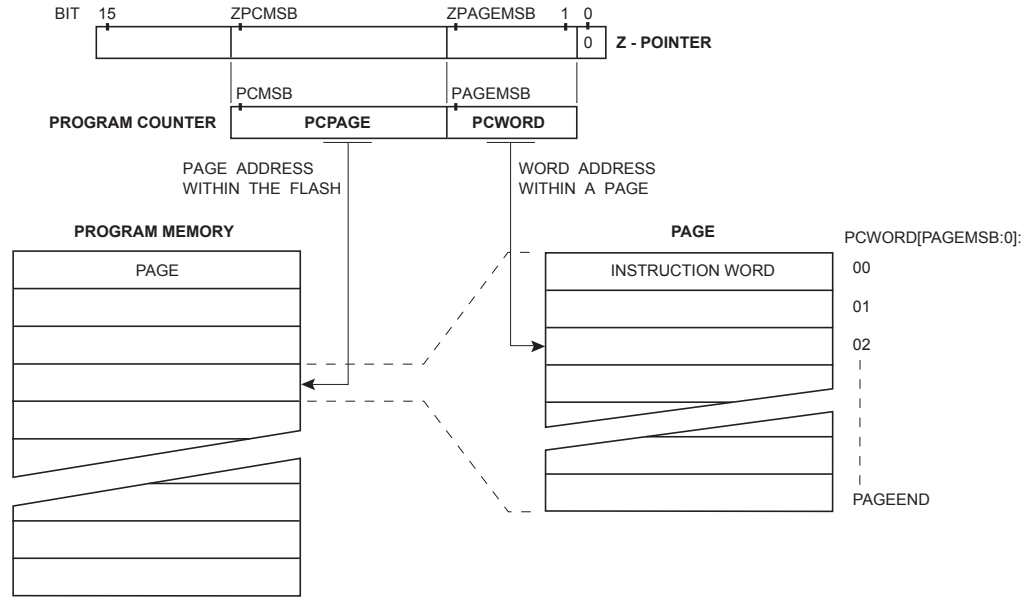
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| | Z15 | Z14 | Z13 | Z12 | Z11 | Z10 | Z9 | Z8 | ZH (R31) |
| | Z7 | Z6 | Z5 | Z4 | Z3 | Z2 | Z1 | Z0 | ZL (R30) |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |

Since the Flash is organized in pages (see Table 21-7 on page 227), the Program Counter can be treated as having two different sections. One section, consisting of the least significant bits, is addressing the words within a page, while the most significant bits are addressing the pages. This is shown in Figure 20-1.

Note that the Page Erase and Page Write operations are addressed independently. Therefore it is of major importance that the software addresses the same page in both the Page Erase and Page Write operation.

The LPM instruction uses the Z-pointer to store the address. Since this instruction addresses the Flash byte-by-byte, also the LSB (bit Z0) of the Z-pointer is used.

**Figure 20-1.** Addressing the Flash During SPM (1)




Note: 1. The different variables used in Table 20-2 are listed in Table 21-7 on page 227.

## 21.4 Calibration Byte

The ATtiny87/167 has a byte calibration value for the internal RC Oscillator. This byte resides in the high byte of address 0x000 in the signature address space. During reset, this byte is automatically written into the OSCCAL Register to ensure correct frequency of the calibrated RC Oscillator.

## 21.5 Page Size

**Table 21-7.** Number of Words in a Page and No. of Pages in the Flash

| Device | Flash Size | Page Size | PCWORD | No. of Pages | PCPAGE | PCMSB |
|---|---|---|---|---|---|---|
| ATtiny87 | 4K words | 64 words | PC[5:0] | 64 | PC[11:6] | 11 |
| ATtiny167 | 8K words | 64 words | PC[5:0] | 128 | PC[12:6] | 12 |

**Table 21-8.** Number of Words in a Page and No. of Pages in the EEPROM

| Device | EEPROM Size | Page Size | PCWORD | No. of Pages | PCPAGE | EEAMSB |
|---|---|---|---|---|---|---|
| ATtiny87<br>ATtiny167 | 512 bytes | 4 bytes | EEA[1:0] | 128 | EEA[8:2] | 8 |

## 21.6 Parallel Programming Parameters, Pin Mapping, and Commands

This section describes how to parallel program and verify Flash Program memory, EEPROM Data memory, Memory Lock bits, and Fuse bits in the ATtiny87/167. Pulses are assumed to be at least 250 ns unless otherwise noted.

### 21.6.1 Signal Names

In this section, some pins of the ATtiny87/167 are referenced by signal names describing their functionality during parallel programming, see Figure 21-1 and Figure 21-9. Pins not described in the following table are referenced by pin names.

The XA1/XA0 pins determine the action executed when the XTAL1 pin is given a positive pulse. The bit coding is shown in Figure 21-11.

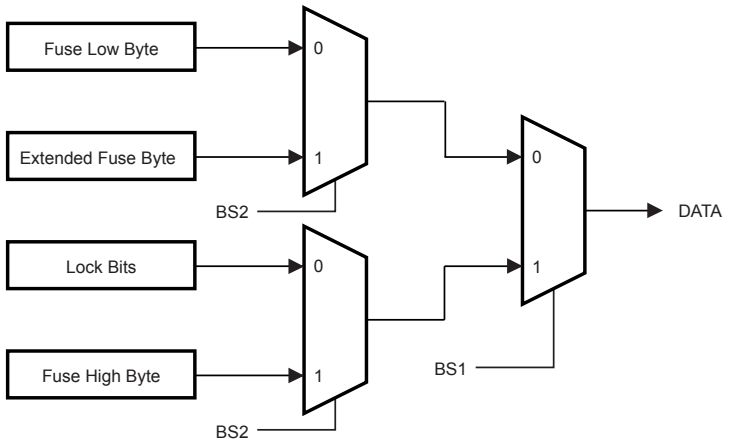
When pulsing $\overline{WR}$ or $\overline{OE}$, the command loaded determines the action executed. The different commands are shown in Figure 21-12.

### 21.7.12 Reading the Fuse and Lock Bits

The algorithm for reading the Fuse and Lock bits is as follows (refer to "Programming the Flash" on page 231 for details on Command loading):

1. **A:** Load Command "0000 0100 $_b$".
2. Set $\overline{OE}$ to "0", BS2 to "0" and BS1 to "0". The status of the Fuse Low bits can now be read at DATA ("0" means programmed).
3. Set $\overline{OE}$ to "0", BS2 to "1" and BS1 to "1". The status of the Fuse High bits can now be read at DATA ("0" means programmed).
4. Set OE to "0", BS2 to "1", and BS1 to "0". The status of the Extended Fuse bits can now be read at DATA ("0" means programmed).
5. Set $\overline{OE}$ to "0", BS2 to "0" and BS1 to "1". The status of the Lock bits can now be read at DATA ("0" means programmed).
6. Set $\overline{OE}$ to "1".

**Figure 21-6.** Mapping Between BS1, BS2 and the Fuse and Lock Bits During Read




### 21.7.13 Reading the Signature Bytes

The algorithm for reading the Signature bytes is as follows (refer to "Programming the Flash" on page 231 for details on Command and Address loading):

1. **A:** Load Command "0000 1000 $_b$".
2. **B:** Load Address Low Byte (0x00 - 0x02).
3. Set $\overline{OE}$ to "0", and BS to "0". The selected Signature byte can now be read at DATA.
4. Set $\overline{OE}$ to "1".

### 21.7.14 Reading the 8 MHz RC Oscillator Calibration Byte

The algorithm for reading the 8 MHz RC Oscillator Calibration byte is as follows (refer to "Programming the Flash" on page 231 for details on Command and Address loading):

1. **A:** Load Command "0000 1000 $_b$".
2. **B:** Load Address Low Byte, 0x00.
3. Set $\overline{OE}$ to "0", and BS1 to "1". The 8 MHz RC Oscillator Calibration byte can now be read at DATA.
4. Set $\overline{OE}$ to "1".

**Table 22-5.** BODLEVEL Fuse Coding

| BODLEVEL[2:0] Fuses | Min. $V_{BOT}$ (1) | Typ. $V_{BOT}$ | Max. $V_{BOT}$ | Units |
|---|---|---|---|---|
| $1\,1\,1_b$ | BOD Disabled | | | |
| $1\,1\,0_b$ | 1.7 | 1.8 | 2.0 | V |
| $1\,0\,1_b$ | 2.5 | 2.7 | 2.9 | |
| $1\,0\,0_b$ | 4.1 | 4.3 | 4.5 | |
| $0\,1\,1_b$ | Reserved | | | |
| $0\,1\,0_b$ | | | | |
| $0\,0\,1_b$ | | | | |
| $0\,0\,0_b$ | | | | |

Notes: 1. $V_{BOT}$ may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to $V_{CC} = V_{BOT}$ during the production test. This guarantees that a Brown-Out Reset will occur before $V_{CC}$ drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 101 for Low Operating Voltage and BODLEVEL = 100 for High operating Voltage.

**Table 22-6.** Brown-out Characteristics

| Symbol | Parameter | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| $V_{HYST}$ | Brown-out Detector Hysteresis | | 80 | | mV |
| $t_{BOD}$ | Min Pulse Width on Brown-out Reset | | 2 | | µs |

## 22.6 Internal Voltage Characteristics

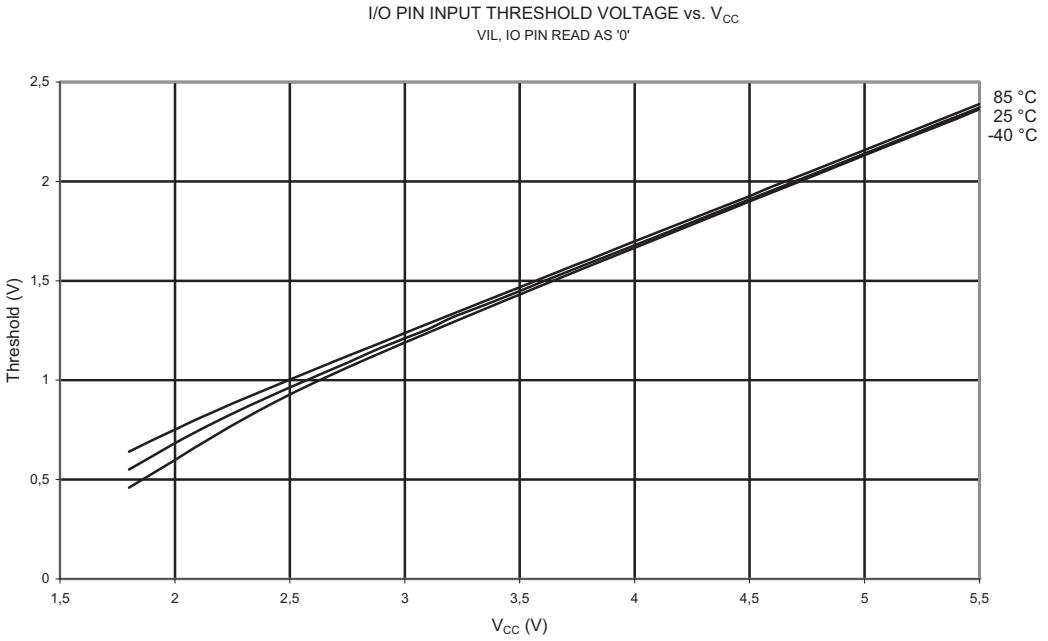
**Table 22-7.** Internal Voltage Reference Characteristics

| Symbol | Parameter | Condition | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $V_{BG}$ | Bandgap reference voltage | $V_{CC}$ = 4.5<br>$T_A$ = 25°C | 1.0 | 1.1 | 1.2 | V |
| $t_{BG}$ | Bandgap reference start-up time | $V_{CC}$ = 4.5<br>$T_A$ = 25°C | | 40 | 70 | µs |
| $I_{BG}$ | Bandgap reference current consumption | $V_{CC}$ = 4.5<br>$T_A$ = 25°C | | 10 | | µA |

**Table 22-10.** ADC Characteristics, Differential Channels (-40°C/+85°C)

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| | Resolution | Differential conversion | | 8 | | |
| TUE | Absolute accuracy | Gain = 8x, BIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | | 1.0 | 3.0 | LSB |
| | | Gain = 20x, BIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | | 1.5 | 3.5 | |
| | | Gain = 8x, UNIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | | 2.0 | 4.5 | |
| | | Gain = 20x, UNIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | | 2.0 | 6.0 | |
| INL | Integral Non Linearity | Gain = 8x, BIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | | 0.2 | 1.0 | LSB |
| | | Gain = 20x, BIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | | 0.4 | 1.5 | |
| | | Gain = 8x, UNIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | | 0.5 | 2.0 | |
| | | Gain = 20x, UNIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | | 1.6 | 5.0 | |
| DNL | Differential Non Linearity | Gain = 8x, BIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | | 0.3 | 0.8 | LSB |
| | | Gain = 20x, BIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | | 0.3 | 0.8 | |
| | | Gain = 8x, UNIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | | 0.4 | 0.8 | |
| | | Gain = 20x, UNIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | | 0.6 | 1.6 | |
| | Gain error | Gain = 8x, BIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | -3.0 | 1.0 | 3.0 | LSB |
| | | Gain = 20x, BIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | -4.0 | 1.5 | 4.0 | |
| | | Gain = 8x, UNIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | -5.0 | -2.5 | 0.0 | |
| | | Gain = 20x, UNIPOLAR<br>$V_{REF}$ = 4V, $V_{CC}$ = 5V<br>ADC clock = 200 kHz | -4.0 | -0.5 | 4.0 | |

**Figure 23-21.** $V_{IL}$: Input Threshold Voltage vs. $V_{CC}$ (I/O Pin, Read as '0')




## 23.9 BOD, Bandgap and Reset

**Figure 23-22.** BOD Thresholds vs. Temperature (BOD Level is 4.3V)

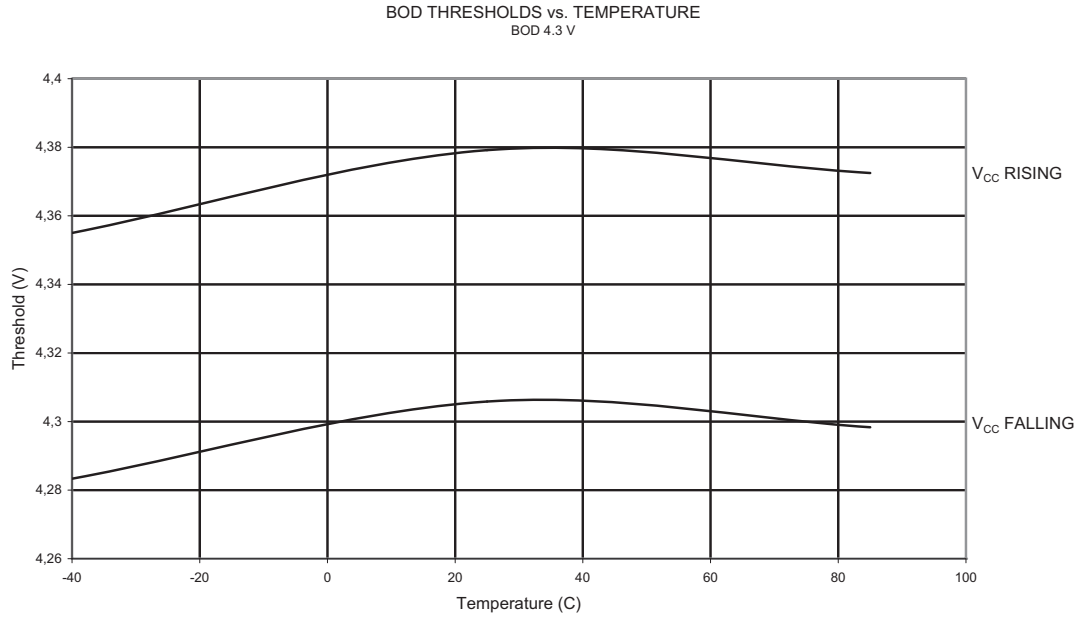


**Figure 23-23.** BOD Thresholds vs. Temperature (BOD Level is 2.7V)

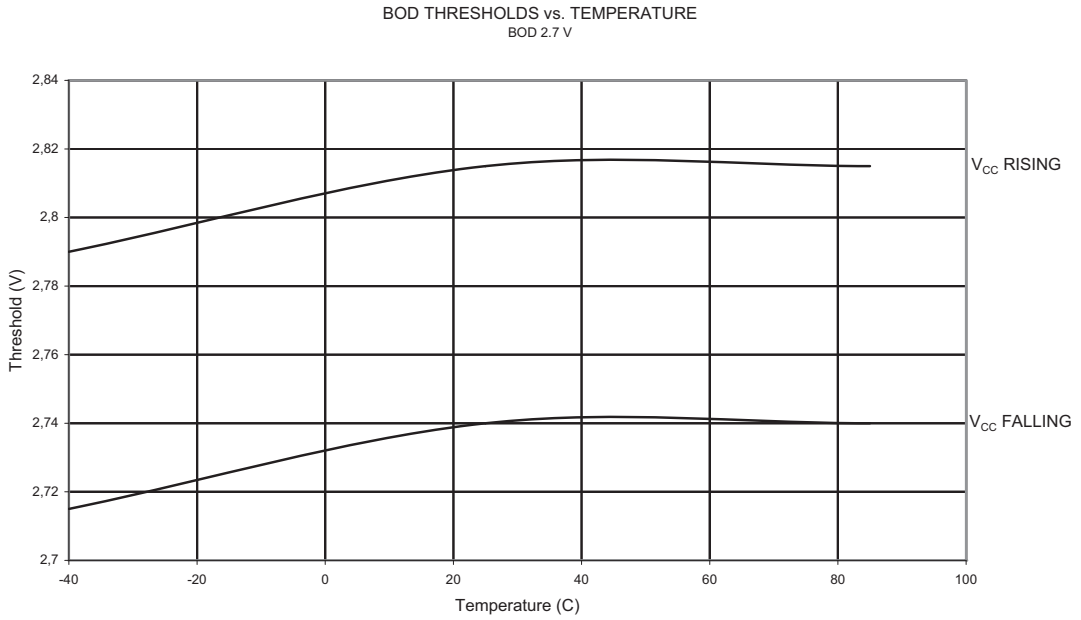
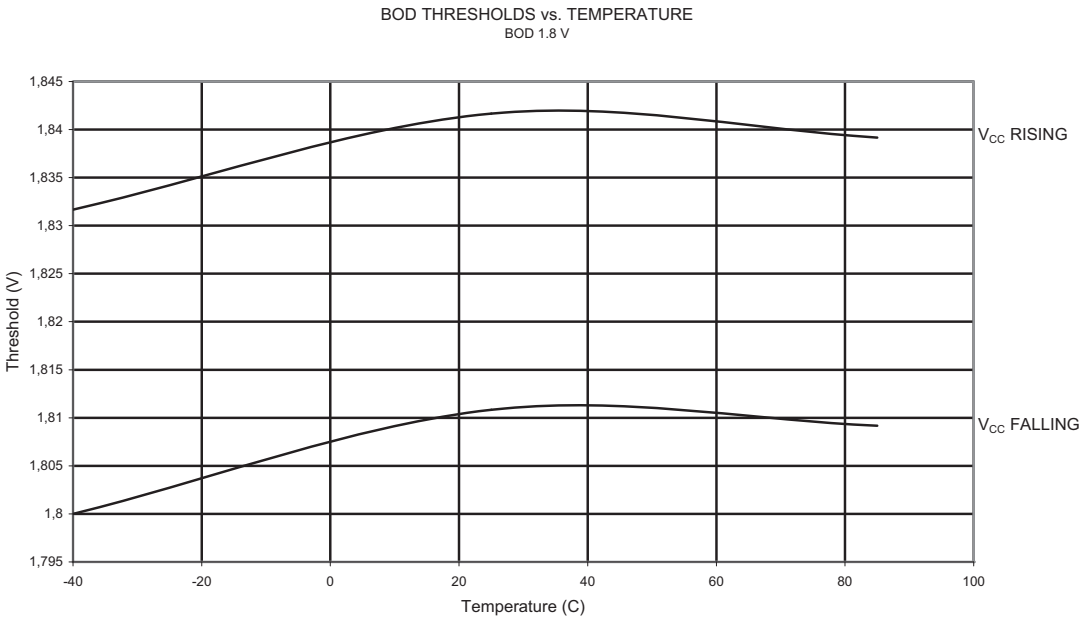



**Figure 23-24.** BOD Thresholds vs. Temperature (BOD Level is 1.8V)



# Table of Contents