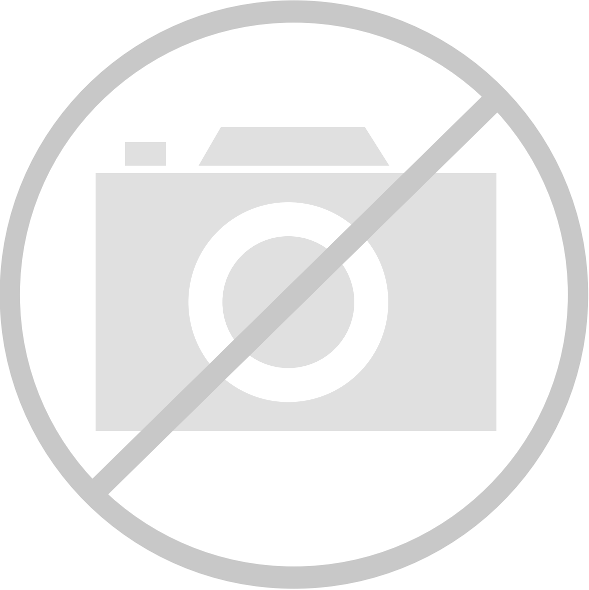
Microchip Technology - ATTINY167-MMUR Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, 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-WFQFN Exposed Pad
Supplier Device Package	20-WQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny167-mmur

3.4 I/O Memory

The I/O space definition of the ATtiny87/167 is shown in Section 24. “Register Summary” on page 270.

All ATtiny87/167 I/Os and peripherals are placed in the I/O space. All I/O locations may be accessed by the LD/LDS/LDD and ST/STS/STD instructions, transferring data between the 32 general purpose working registers and the I/O space. 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. Refer to the instruction set section for more details. 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 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.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, 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.

The I/O and Peripherals Control Registers are explained in later sections.

3.4.1 General Purpose I/O Registers

The ATtiny87/167 contains three General Purpose I/O Registers. These registers can be used for storing any information, and they are particularly useful for storing global variables and Status Flags.

The General Purpose I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI, CBI, SBIS, and SBIC instructions.

3.5 Register Description

3.5.1 EEARH and EEARL – EEPROM Address Register

Bit	7	6	5	4	3	2	1	0	
0x22 (0x42)	–	–	–	–	–	–	–	**EEAR8**	**EEARH**
0x21 (0x41)	**EEAR7**	**EEAR6**	**EEAR5**	**EEAR4**	**EEAR3**	**EEAR2**	**EEAR1**	**EEAR0**	**EEARL**
Bit	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
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	X	
Initial Value	X	X	X	X	X	X	X	X	

- **Bits 7:1 – Res: Reserved Bits**

These bits are reserved for future use and will always read as 0 in ATtiny87/167.

- **Bits 8:0 – EEAR[8:0]: EEPROM Address**

The EEPROM Address Registers – EEARH and EEARL – specifies the high EEPROM address in the EEPROM space (see “E2 size” in Table 3-1 on page 15). The EEPROM data bytes are addressed linearly between 0 and “E2 size”. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

3.5.2 EEDR – EEPROM Data Register

Bit	7	6	5	4	3	2	1	0	
0x20 (0x40)	EEDR7	EEDR6	EEDR5	EEDR4	EEDR3	EEDR2	EEDR1	EEDR0	EEDR
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 – EEDR[7:0]: EEPROM Data**

For the EEPROM write operation the EEDR Register contains the data to be written to the EEPROM in the address given by the EEAR Register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

3.5.3 EECR – EEPROM Control Register

Bit	7	6	5	4	3	2	1	0	
0x1F (0x3F)	–	–	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	EECR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	X	X	0	0	X	0	

- **Bits 7:6 – Res: Reserved Bits**

These bits are reserved for future use and will always read as 0 in ATtiny87/167. After reading, mask out these bits. For compatibility with future AVR devices, always write these bits to zero.

- **Bits 5:4 – EEPM[1:0]: EEPROM Programming Mode Bits**

The EEPROM Programming mode bits setting defines which programming action that will be triggered when writing EEPE. It is possible to program data in one atomic operation (erase the old value and program the new value) or to split the Erase and Write operations in two different operations. The Programming times for the different modes are shown in Table 3-2. While EEPE is set, any write to EEPMn will be ignored. During reset, the EEPMn bits will be reset to 0b00 unless the EEPROM is busy programming.

Table 3-2. EEPROM Mode Bits

EEPM1	EEPM0	Typical Programming Time	Operation
0	0	3.4 ms	Erase and Write in one operation (Atomic Operation)
0	1	1.8 ms	Erase Only
1	0	1.8 ms	Write Only
1	1	–	Reserved for future use

- **Bit 3 – EERIE: EEPROM Ready Interrupt Enable**

Writing EERIE to one enables the EEPROM Ready Interrupt if the I-bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM Ready Interrupt generates a constant interrupt when Non-volatile memory is ready for programming.

- **Bit 2 – EEMPE: EEPROM Master Program Enable**

The EEMPE bit determines whether writing EEPE to one will have effect or not.

4. System Clock and Clock Options

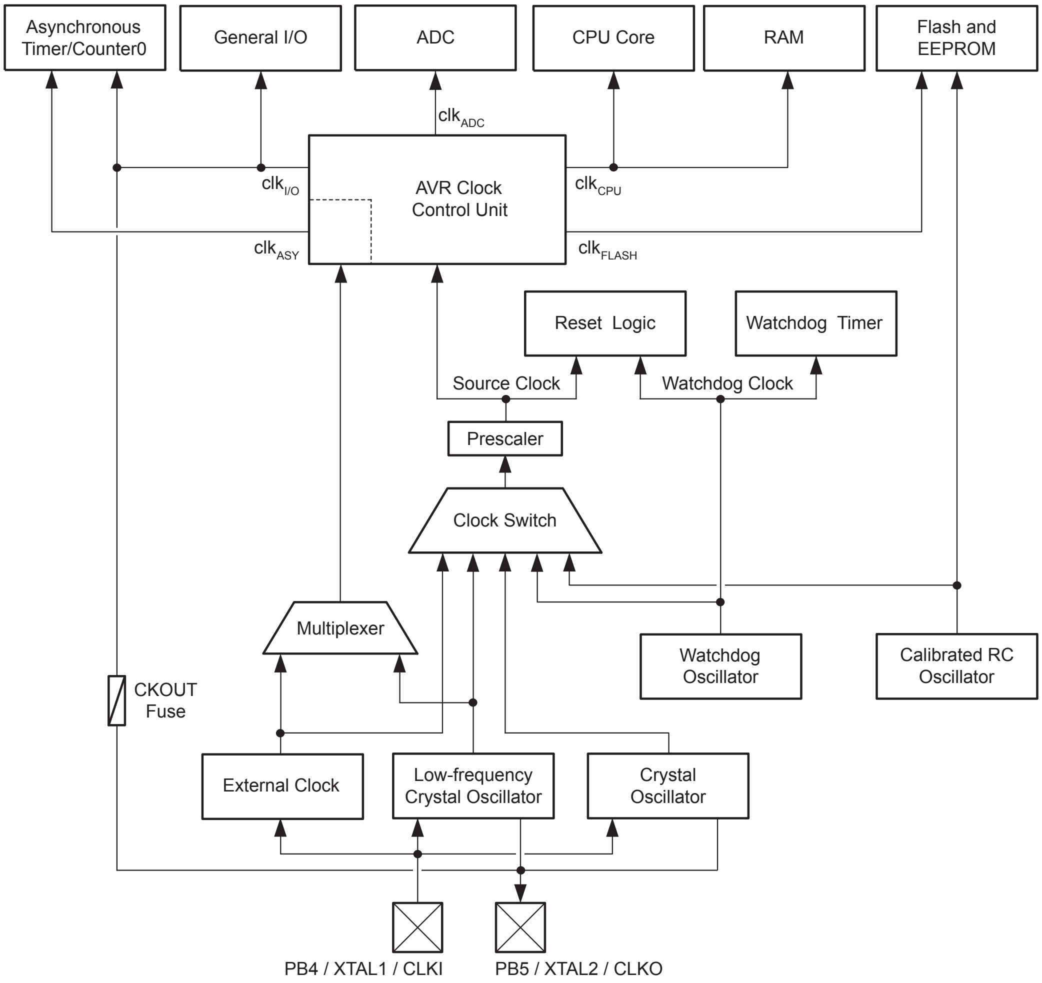
The ATtiny87/167 provides a large number of clock sources. They can be divided into two categories: internal and external. Some external clock sources can be shared with the asynchronous timer. After reset, the clock source is determined by the CKSEL Fuses. Once the device is running, software clock switching is possible to any other clock sources.

Hardware controls are provided for clock switching management but some specific procedures must be observed. Clock switching should be performed with caution as some settings could result in the device having an incorrect configuration.

4.1 Clock Systems and their Distribution

Figure 4-1 presents the principal clock systems in the AVR and their distribution. All of the clocks may not need to be active at any given time. In order to reduce power consumption, the clocks to modules not being used can be halted by using different sleep modes or by using features of the dynamic clock switch circuit (See "Power Management and Sleep Modes" on page 42 and "Dynamic Clock Switch" on page 31). The clock systems are detailed below.

Figure 4-1. Clock Distribution

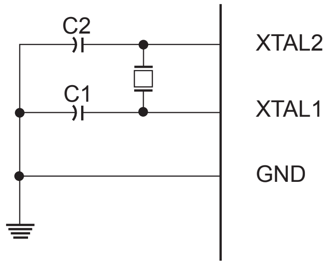


4.2.4 Crystal Oscillator

XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an On-chip Oscillator, as shown in Figure 4-2. Either a quartz crystal or a ceramic resonator may be used.

C1 and C2 should always be equal for both crystals and resonators. The optimal value of the capacitors depends on the crystal or resonator in use, the amount of stray capacitance, and the electromagnetic noise of the environment. Some initial guidelines for choosing capacitors for use with crystals are given in Table 4-6. For ceramic resonators, the capacitor values given by the manufacturer should be used.

Figure 4-2. Crystal Oscillator Connections



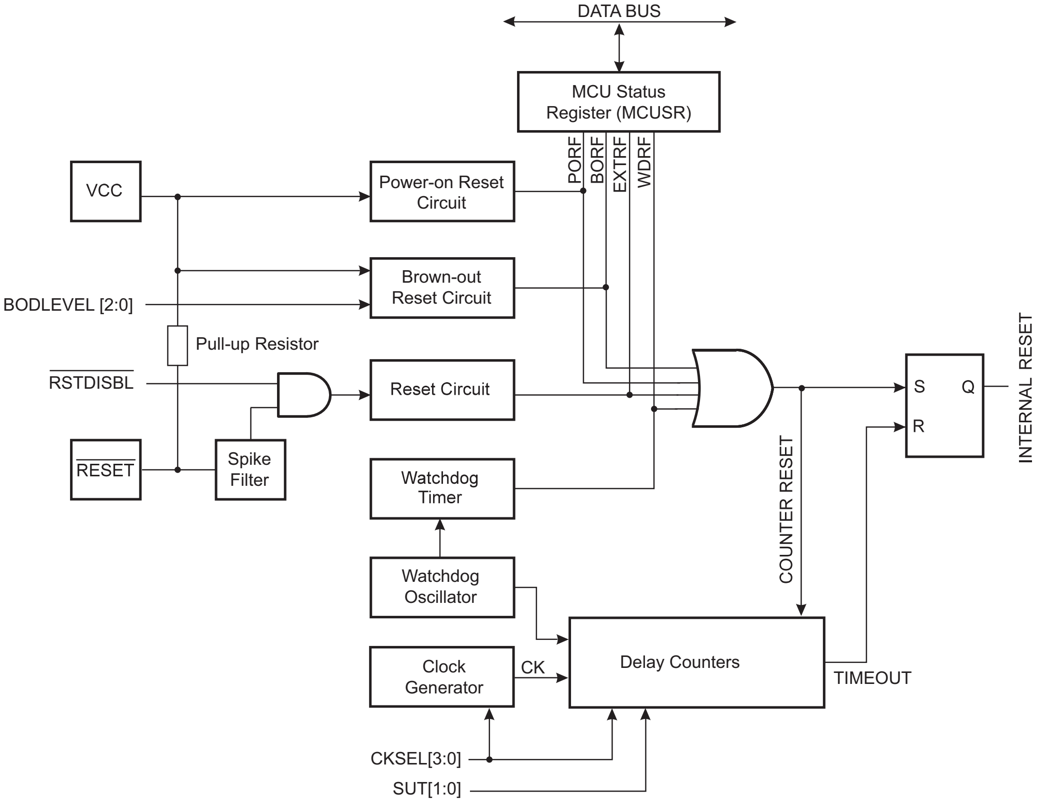
The Oscillator can operate in three different modes, each optimized for a specific frequency range. The operating mode is selected by CKSEL[3:1] fuses or by CSEL[3:1] field as shown in Table 4-6.

Table 4-6. Crystal Oscillator Operating Modes

CKSEL[3:1][(1)] CSEL[3:1][(2)]	Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals (pF)
100[(3)]	0.4 - 0.9	–
101	0.9 - 3.0	12 - 22
110	3.0 - 8.0	12 - 22
111	8.0 - 16.0	12 - 22

Notes:
1. Flash Fuse bits.
2. CLKSELR register bits.
3. This option should not be used with crystals, only with ceramic resonators.

Figure 6-1. Reset Circuit



6.1.3 Power-on Reset

A Power-on Reset (POR) pulse is generated by an On-chip detection circuit. The detection level is defined in Table 22-4 on page 245. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the start-up Reset, as well as to detect a failure in supply voltage.

A Power-on Reset (POR) circuit ensures that the device is reset from Power-on. Reaching the Power-on Reset threshold voltage invokes the delay counter, which determines how long the device is kept in RESET after V_{CC} rise. The RESET signal is activated again, without any delay, when V_{CC} decreases below the detection level.

Figure 6-2. MCU Start-up, $\overline{\text{RESET}}$ Tied to V_{CC}

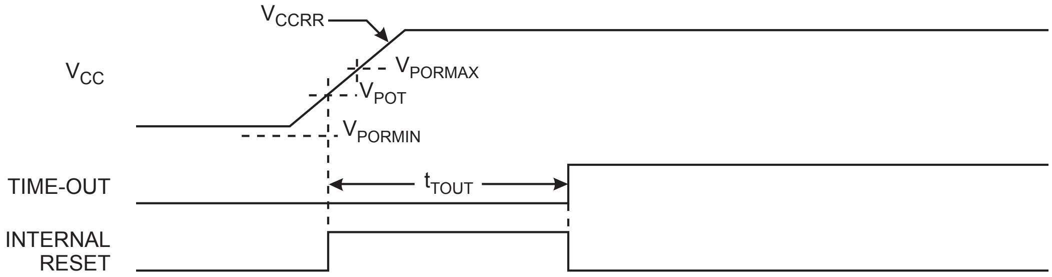
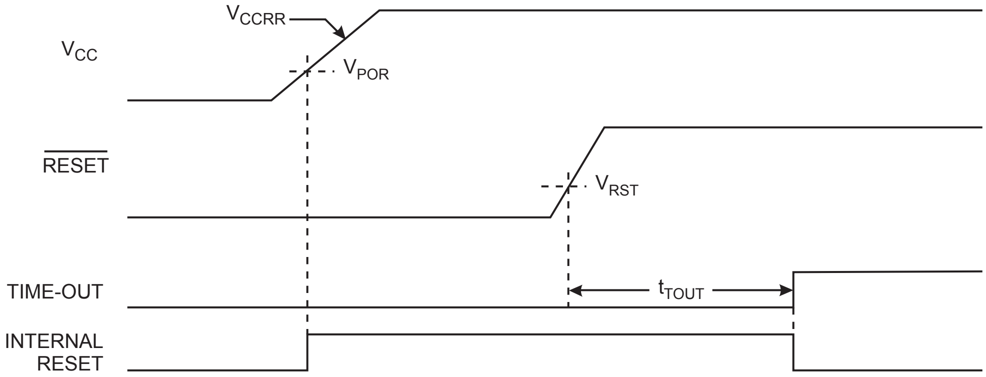


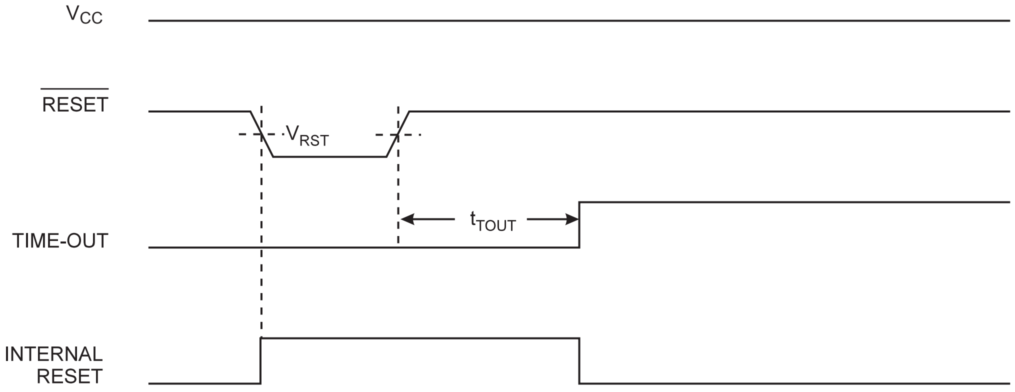
Figure 6-3. MCU Start-up, $\overline{\text{RESET}}$ Extended Externally



6.1.4 External Reset

An External Reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than the minimum pulse width (see Table 22-3 on page 245) will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage – V_{RST} – on its positive edge, the delay counter starts the MCU after the Time-out period – t_{TOUT} – has expired. The External Reset can be disabled by the RSTDISBL fuse, see Table 21-4 on page 225.

Figure 6-4. External Reset During Operation



6.1.5 Brown-out Detection

ATtiny87/167 has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the BODLEVEL Fuses (See "BODLEVEL Fuse Coding" on page 246.). The trigger level has a hysteresis to ensure spike free Brown-out Detection. The hysteresis on the detection level should be interpreted as $V_{BOT+} = V_{BOT} + V_{HYST} / 2$ and $V_{BOT-} = V_{BOT} - V_{HYST} / 2$.

When the BOD is enabled, and V_{CC} decreases to a value below the trigger level (V_{BOT-} in Figure 6-5), the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level (V_{BOT+} in Figure 6-5), the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than t_{BOD} given in Table 22-6 on page 246.

- 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.

9.4 Register Description

9.4.1 PORTA – Port A Data Register

Bit	7	6	5	4	3	2	1	0	
0x02 (0x22)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
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	

9.4.2 DDRA – Port A Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x01 (0x21)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
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	

9.4.3 PINA – Port A Input Pins Register

Bit	7	6	5	4	3	2	1	0	
0x00 (0x20)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

9.4.4 PORTB – Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
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	

9.4.5 DDRB – Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
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	

9.4.6 PINB – Port B Input Pins Register

Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

The OCR0A Register is double buffered when using any of the Pulse Width Modulation (PWM) modes. For the Normal and Clear Timer on Compare (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR0A Compare Register to either top or bottom of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The OCR0A Register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR0A Buffer Register, and if double buffering is disabled the CPU will access the OCR0A directly.

10.5.1 Force Output Compare

In non-PWM waveform generation modes, the match output of the comparator can be forced by writing a one to the Force Output Compare (FOC0A) bit. Forcing compare match will not set the OCF0A flag or reload/clear the timer, but the OC0A pin will be updated as if a real compare match had occurred (the COM0A[1:0] bits settings define whether the OC0A pin is set, cleared or toggled).

10.5.2 Compare Match Blocking by TCNT0 Write

All CPU write operations to the TCNT0 Register will block any compare match that occurs in the next timer clock cycle, even when the timer is stopped. This feature allows OCR0A to be initialized to the same value as TCNT0 without triggering an interrupt when the Timer/Counter clock is enabled.

10.5.3 Using the Output Compare Unit

Since writing TCNT0 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT0 when using the Output Compare channel, independently of whether the Timer/Counter is running or not. If the value written to TCNT0 equals the OCR0A value, the compare match will be missed, resulting in incorrect waveform generation. Similarly, do not write the TCNT0 value equal to BOTTOM when the counter is downcounting.

The setup of the OC0A should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC0A value is to use the Force Output Compare (FOC0A) strobe bit in Normal mode. The OC0A Register keeps its value even when changing between Waveform Generation modes.

Be aware that the COM0A[1:0] bits are not double buffered together with the compare value. Changing the COM0A[1:0] bits will take effect immediately.

10.6 Compare Match Output Unit

The Compare Output mode (COM0A[1:0]) bits have two functions. The Waveform Generator uses the COM0A[1:0] bits for defining the Output Compare (OC0A) state at the next compare match. Also, the COM0A[1:0] bits control the OC0A pin output source. Figure 10-4 shows a simplified schematic of the logic affected by the COM0A[1:0] bit setting. The I/O Registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O port control registers (DDR and PORT) that are affected by the COM0A[1:0] bits are shown. When referring to the OC0A state, the reference is for the internal OC0A Register, not the OC0A pin.

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the Tn pin. The Clock Select logic block controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_{Tn}).

The double buffered Output Compare Registers (OCR1A/B) are compared with the Timer/Counter value at all time. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pins See ”Output Compare Units” on page 118.. The compare match event will also set the Compare Match Flag (OCF1A/B) which can be used to generate an Output Compare interrupt request.

The Input Capture Register can capture the Timer/Counter value at a given external (edge triggered) event on either the Input Capture pin (ICP1) or on the Analog Comparator pins (See ”AnaComp - Analog Comparator” on page 210.). The Input Capture unit includes a digital filtering unit (Noise Canceler) for reducing the chance of capturing noise spikes.

The TOP value, or maximum Timer/Counter value, can in some modes of operation be defined by either the OCR1A Register, the ICR1 Register, or by a set of fixed values. When using OCR1A as TOP value in a PWM mode, the OCR1A Register can not be used for generating a PWM output. However, the TOP value will in this case be double buffered allowing the TOP value to be changed in run time. If a fixed TOP value is required, the ICR1 Register can be used as an alternative, freeing the OCR1A to be used as PWM output.

12.2.2 Definitions

The following definitions are used extensively throughout the section:

BOTTOM	The counter reaches the BOTTOM when it becomes 0x0000.
MAX	The counter reaches its MAXimum when it becomes 0xFFFF (decimal 65,535).
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be one of the fixed values: 0x00FF, 0x01FF, or 0x03FF, or to the value stored in the OCR1A or ICR1 Register. The assignment is dependent of the mode of operation.

12.3 Accessing 16-bit Registers

The TCNT1, OCR1A/B, and ICR1 are 16-bit registers that can be accessed by the AVR CPU via the 8-bit data bus. The 16-bit register must be byte accessed using two read or write operations. Each 16-bit timer has a single 8-bit register for temporary storing of the high byte of the 16-bit access. The same temporary register is shared between all 16-bit registers within each 16-bit timer. Accessing the low byte triggers the 16-bit read or write operation. When the low byte of a 16-bit register is written by the CPU, the high byte stored in the temporary register, and the low byte written are both copied into the 16-bit register in the same clock cycle. When the low byte of a 16-bit register is read by the CPU, the high byte of the 16-bit register is copied into the temporary register in the same clock cycle as the low byte is read.

Not all 16-bit accesses uses the temporary register for the high byte. Reading the OCR1A/B 16-bit registers does not involve using the temporary register.

To do a 16-bit write, the high byte must be written before the low byte. For a 16-bit read, the low byte must be read before the high byte.

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.

or toggle at a compare match (See "Compare Match Output Unit" on page 119.). The OCnxi bits over control the setting of the COM1A/B[1:0] bits as shown in Figure 12-6 on page 121.

For detailed timing information refer to "Timer/Counter Timing Diagrams" on page 129.

12.9.1 Normal Mode

The simplest mode of operation is the Normal mode (WGM1[3:0] = 0). In this mode the counting direction is always up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its maximum 16-bit value (MAX = 0xFFFF) and then restarts from the BOTTOM (0x0000). In normal operation the Timer/Counter Overflow Flag (TOV1) will be set in the same timer clock cycle as the TCNT1 becomes zero. The TOV1 flag in this case behaves like a 17th bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the TOV1 flag, the timer resolution can be increased by software. There are no special cases to consider in the Normal mode, a new counter value can be written anytime.

The Input Capture unit is easy to use in Normal mode. However, observe that the maximum interval between the external events must not exceed the resolution of the counter. If the interval between events are too long, the timer overflow interrupt or the prescaler must be used to extend the resolution for the capture unit.

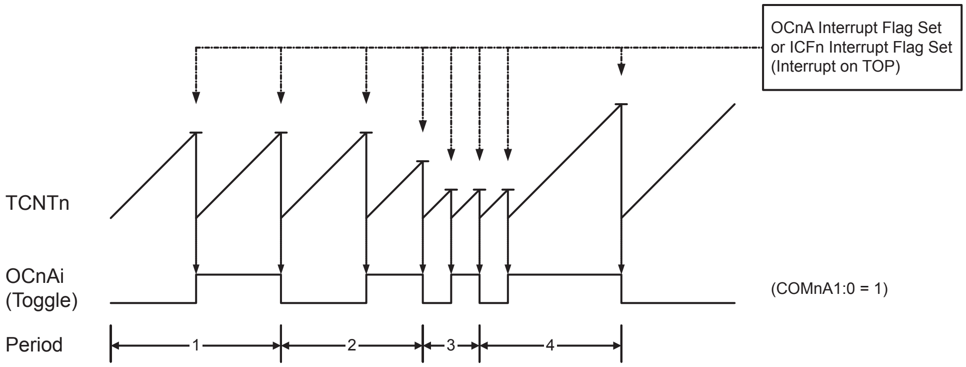
The Output Compare units can be used to generate interrupts at some given time. Using the Output Compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

12.9.2 Clear Timer on Compare Match (CTC) Mode

In Clear Timer on Compare or CTC mode (WGM1[3:0] = 4 or 12), the OCR1A or ICR1 Register are used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT1) matches either the OCR1A (WGM1[3:0] = 4) or the ICR1 (WGM1[3:0] = 12). The OCR1A or ICR1 define the top value for the counter, hence also its resolution. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events.

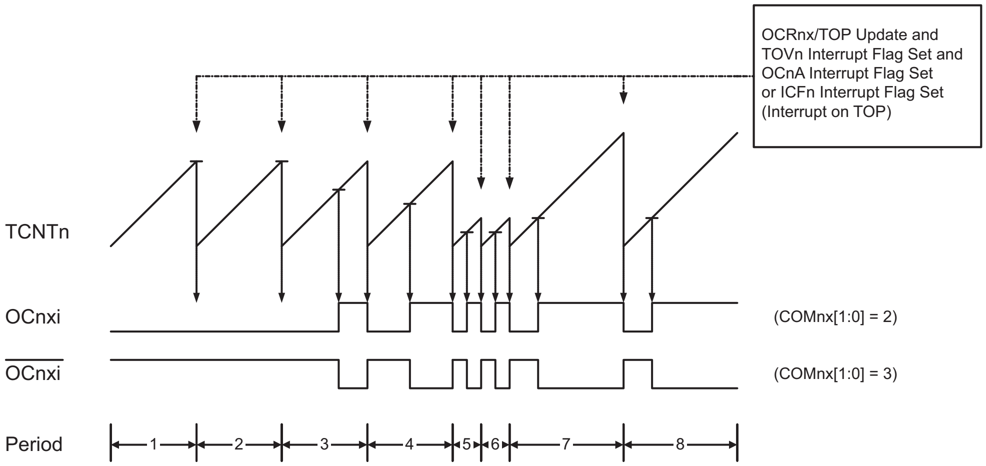
The timing diagram for the CTC mode is shown in Figure 12-7. The counter value (TCNT1) increases until a compare match occurs with either OCR1A or ICR1, and then counter (TCNT1) is cleared.

Figure 12-7. CTC Mode, Timing Diagram



In fast PWM mode the counter is incremented until the counter value matches either one of the fixed values 0x00FF, 0x01FF, or 0x03FF (WGM1[3:0] = 5, 6, or 7), the value in ICR1 (WGM1[3:0] = 14), or the value in OCR1A (WGM1[3:0] = 15). The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 12-8. The figure shows fast PWM mode when OCR1A or ICR1 is used to define TOP. The TCNT1 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 TCNT1 slopes represent compare matches between OCR1A/B and TCNT1. The OC1A/B interrupt flag will be set when a compare match occurs.

Figure 12-8. Fast PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV1) is set each time the counter reaches TOP. In addition the OC1A or ICF1 flag is set at the same timer clock cycle as TOV1 is set when either OCR1A or ICR1 is used for defining the TOP value. If one of the interrupts are enabled, the interrupt handler routine can be used for updating the TOP and compare values.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the Compare Registers. If the TOP value is lower than any of the Compare Registers, a compare match will never occur between the TCNT1 and the OCR1A/B. Note that when using fixed TOP values the unused bits are masked to zero when any of the OCR1A/B Registers are written.

The procedure for updating ICR1 differs from updating OCR1A when used for defining the TOP value. The ICR1 Register is not double buffered. This means that if ICR1 is changed to a low value when the counter is running with none or a low prescaler value, there is a risk that the new ICR1 value written is lower than the current value of TCNT1. The result will then be that the counter will miss the compare match at the TOP value. The counter will then have to count to the MAX value (0xFFFF) and wrap around starting at 0x0000 before the compare match can occur. The OCR1A Register however, is double buffered. This feature allows the OCR1A I/O location to be written anytime. When the OCR1A I/O location is written the value written will be put into the OCR1A Buffer Register. The OCR1A Compare Register will then be updated with the value in the Buffer Register at the next timer clock cycle the TCNT1 matches TOP. The update is done at the same timer clock cycle as the TCNT1 is cleared and the TOV1 flag is set.

12.11.5 TCNT1H and TCNT1L – Timer/Counter1

Bit	7	6	5	4	3	2	1	0	
(0x85)	TCNT1[15:8]								TCNT1H
(0x84)	TCNT1[7:0]								TCNT1L
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	

The two Timer/Counter I/O locations (TCNT1H and TCNT1L, combined TCNT1) give direct access, both for read and for write operations, to the Timer/Counter unit 16-bit counter. To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers. See “Accessing 16-bit Registers” on page 111.

Modifying the counter (TCNT1) while the counter is running introduces a risk of missing a compare match between TCNT1 and one of the OCR1A/B Registers.

Writing to the TCNT1 Register blocks (removes) the compare match on the following timer clock for all compare units.

12.11.6 OCR1AH and OCR1AL – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
(0x89)	OCR1A[15:8]								OCR1AH
(0x88)	OCR1A[7:0]								OCR1AL
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	

12.11.7 OCR1BH and OCR1BL – Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
(0x8B)	OCR1B[15:8]								OCR1BH
(0x8A)	OCR1B[7:0]								OCR1BL
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	

The Output Compare Registers contain a 16-bit value that is continuously compared with the counter value (TCNT1). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC1A/B pin.

The Output Compare Registers are 16-bit in size. To ensure that both the high and low bytes are written simultaneously when the CPU writes to these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers. See “Accessing 16-bit Registers” on page 111.

14. USI – Universal Serial Interface

14.1 Features

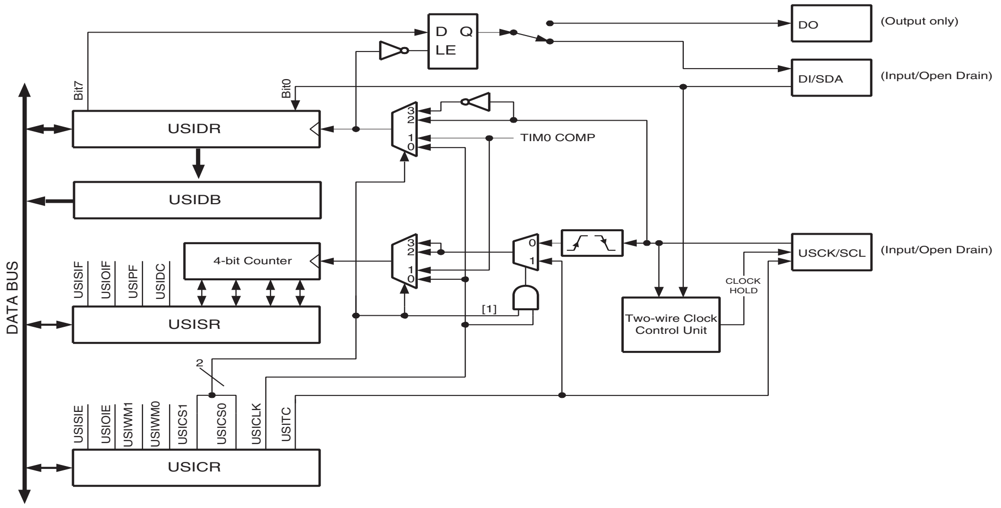
- **Two-wire Synchronous Data Transfer (Master or Slave)**
- **Three-wire Synchronous Data Transfer (Master or Slave)**
- **Data Received Interrupt**
- **Wakeup from Idle Mode**
- **In Two-wire Mode: Wake-up from All Sleep Modes, Including Power-down Mode**
- **Two-wire Start Condition Detector with Interrupt Capability**

14.2 Overview

The Universal Serial Interface, or USI, provides the basic hardware resources needed for serial communication. Combined with a minimum of control software, the USI allows significantly higher transfer rates and uses less code space than solutions based on software only. Interrupts are included to minimize the processor load.

A simplified block diagram of the USI is shown on Figure 14-1 For the actual placement of I/O pins, refer to "Pin Configuration" on page 4. CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the "Register Description" on page 155.

Figure 14-1. Universal Serial Interface, Block Diagram



The 8-bit USI Data Register is directly accessible via the data bus and contains the incoming and outgoing data. The register has no buffering so the data must be read as quickly as possible to ensure that no data is lost. The USI Data Register is a serial shift register and the most significant bit that is the output of the serial shift register is connected to one of two output pins depending of the wire mode configuration.

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.

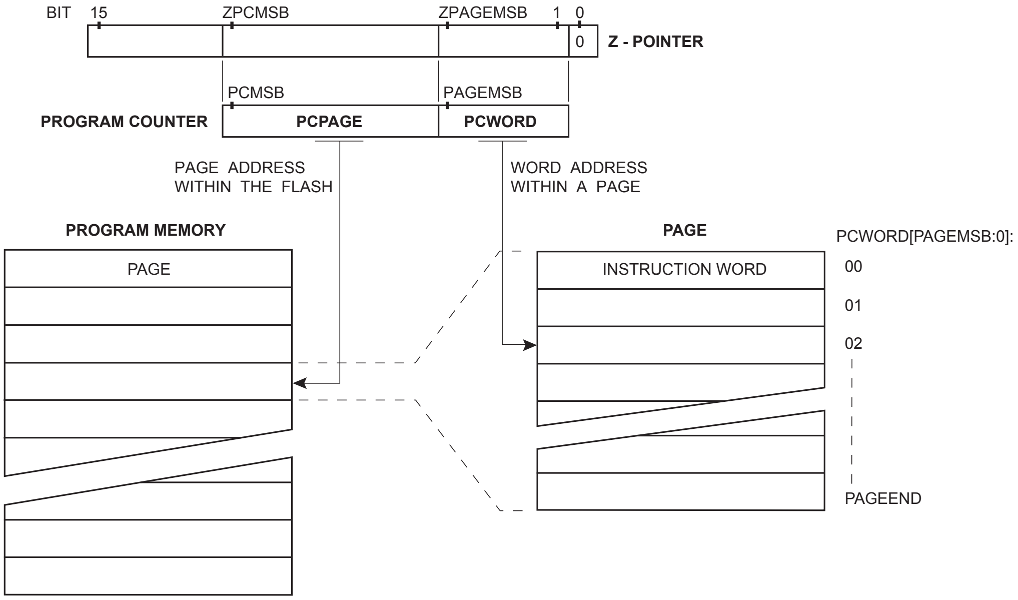
Bit	15	14	13	12	11	10	9	8	
	Z15	**Z14**	**Z13**	**Z12**	**Z11**	**Z10**	**Z9**	**Z8**	**ZH (R31)**
	Z7	**Z6**	**Z5**	**Z4**	**Z3**	**Z2**	**Z1**	**Z0**	**ZL (R30)**
Bit	7	6	5	4	3	2	1	0	

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.

Note that the fuses are read as logical zero, “0”, if they are programmed.

Table 21-3. Extended Fuse Byte

Fuse Extended Byte	Bit No	Description	Default Value
–	7	–	1 (unprogrammed)
–	6	–	1 (unprogrammed)
–	5	–	1 (unprogrammed)
–	4	–	1 (unprogrammed)
–	3	–	1 (unprogrammed)
–	2	–	1 (unprogrammed)
–	1	–	1 (unprogrammed)
SELFPRGEN	0	Self Programming Enable	1 (unprogrammed)

Table 21-4. Fuse High Byte

Fuse High Byte	Bit No	Description	Default Value
RSTDISBL(1)	7	External Reset Disable	1 (unprogrammed)
DWEN	6	DebugWIRE Enable	1 (unprogrammed)
SPIEN(2)	5	Enable Serial Program and Data Downloading	0 (programmed, SPI programming enabled)
WDTON(3)	4	Watchdog Timer always on	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed, EEPROM not preserved)
BODLEVEL2(4)	2	Brown-out Detector trigger level	1 (unprogrammed)
BODLEVEL1(4)	1	Brown-out Detector trigger level	1 (unprogrammed)
BODLEVEL0(4)	0	Brown-out Detector trigger level	1 (unprogrammed)

Notes:
1. See ”Alternate Functions of Port B” on page 81. for description of RSTDISBL Fuse.
2. The SPIEN Fuse is not accessible in serial programming mode.
3. See ”WDTCR – Watchdog Timer Control Register” on page 57. for details.
4. See Table 22-5 on page 246 for BODLEVEL Fuse coding.

Table 21-10. Pin Values Used to Enter Programming Mode

Pin	Symbol	Value
PAGEL / BS1	Prog_enable[3]	0
XA1 / BS2	Prog_enable[2]	0
XA0	Prog_enable[1]	0
$\overline{\text{WR}}$	Prog_enable[0]	0

Table 21-11. XA1 and XA0 Coding

XA1	XA0	Action when XTAL1 is Pulsed
0	0	Load Flash or EEPROM Address (High or low address byte determined by BS1).
0	1	Load Data (High or Low data byte for Flash determined by BS1).
1	0	Load Command
1	1	No Action, Idle

Table 21-12. Command Byte Bit Coding

Command Byte	Command Executed
1000 0000 $_b$	Chip Erase
0100 0000 $_b$	Write Fuse bits
0010 0000 $_b$	Write Lock bits
0001 0000 $_b$	Write Flash
0001 0001 $_b$	Write EEPROM
0000 1000 $_b$	Read Signature bytes and Calibration byte
0000 0100 $_b$	Read Fuse and Lock bits
0000 0010 $_b$	Read Flash
0000 0011 $_b$	Read EEPROM

1. Set XA1, XA0 to “0,0”. This enables address loading.
2. Set BS1 to “1”. This selects high address.
3. Set DATA = Address high byte (0x00 - 0xFF).
4. Give XTAL1 a positive pulse. This loads the address high byte.

H. Program Page

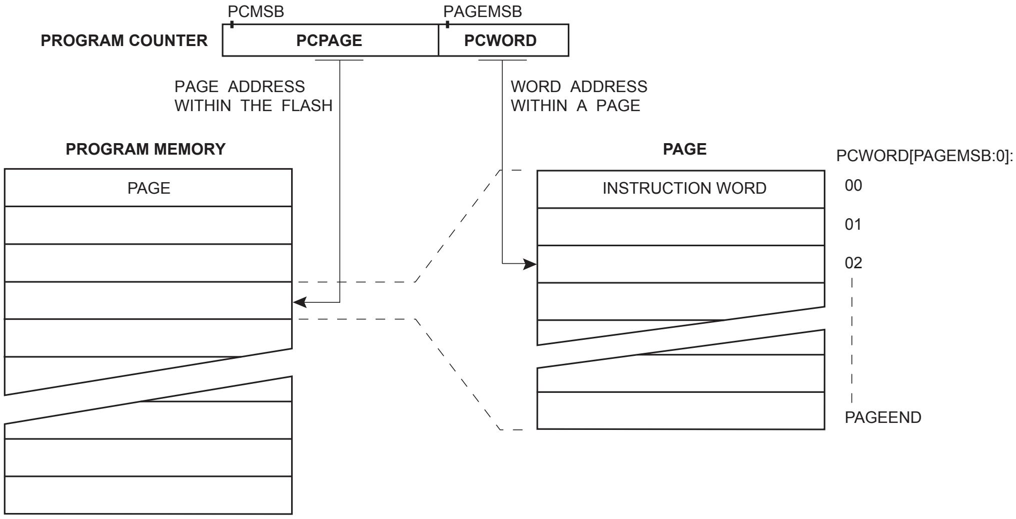
1. Give $\overline{\text{WR}}$ a negative pulse. This starts programming of the entire page of data. RDY/$\overline{\text{BSY}}$ goes low.
2. Wait until RDY/$\overline{\text{BSY}}$ goes high (See Figure 21-3 for signal waveforms).

I. Repeat B through H until the entire Flash is programmed or until all data has been programmed.

J. End Page Programming

1. 1. Set XA1, XA0 to “1,0”. This enables command loading.
2. Set DATA to “0000 0000 $_b$”. This is the command for No Operation.
3. Give XTAL1 a positive pulse. This loads the command, and the internal write signals are reset.

Figure 21-2. Addressing the Flash Which is Organized in Pages



Note: 1. PCPAGE and PCWORD are listed in

Figure 23-6. Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)

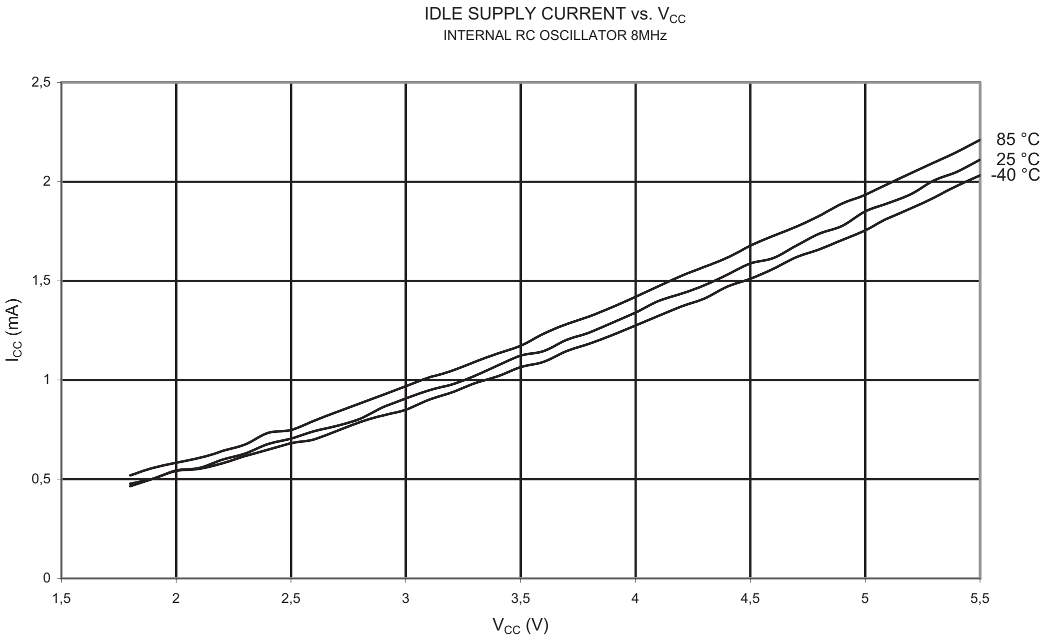
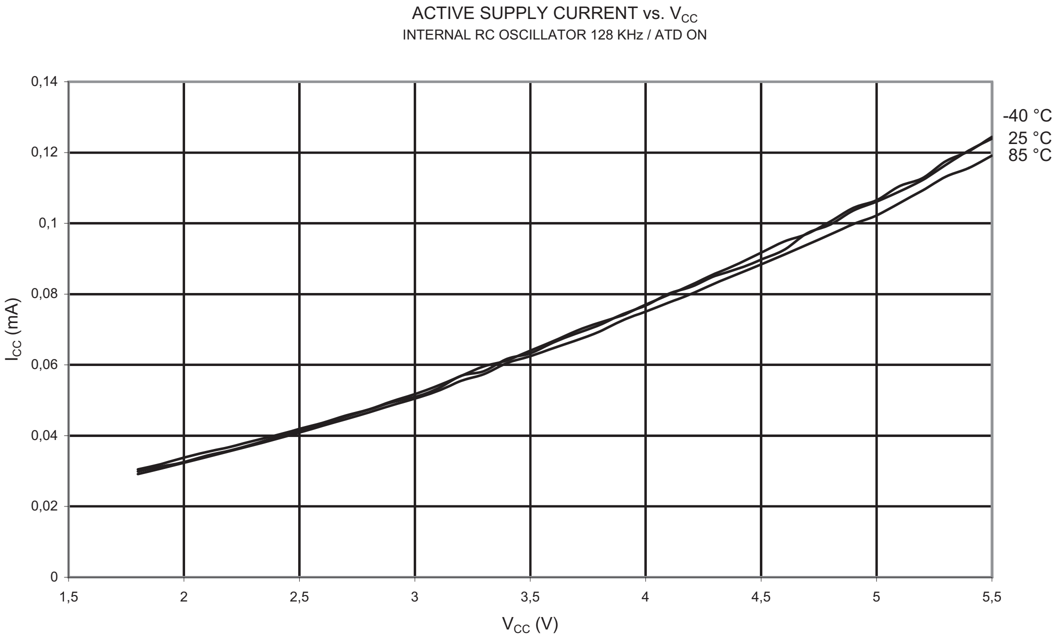


Figure 23-7. Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)


ACTIVE SUPPLY CURRENT vs. V_{CC}
INTERNAL RC OSCILLATOR 128 KHz / ATD ON
I_{CC} (mA)
V_{CC} (V)
-40 °C
25 °C
85 °C