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-mmu

2.3.1 SREG – AVR Status Register

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
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	

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry is useful in BCD arithmetic. See the “Instruction Set Description” for detailed information.

- **Bit 4 – S: Sign Bit, S = N ⊕ V**

The S-bit is always an exclusive or between the Negative Flag N and the Two’s Complement Overflow Flag V. See the “Instruction Set Description” for detailed information.

- **Bit 3 – V: Two’s Complement Overflow Flag**

The Two’s Complement Overflow Flag V supports two’s complement arithmetics. See the “Instruction Set Description” for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

Table 4-3. Internal Calibrated RC Oscillator Operating Modes[1]

Frequency Range[2] (MHz)	CKSEL[3:0][3][4] CSEL[3:0][5]
7.6 - 8.4	0010

Notes:
1. If 8 MHz frequency exceeds the specification of the device (depends on V_{CC}), 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 4-4.

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

SUT[1:0][1] CSUT[1:0][2]	Start-up Time from Power-down/save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00[3]	6 CK	14CK	BOD enabled
01	6 CK	14CK + 4.1 ms	Fast rising power
10[4]	6 CK	14CK + 65 ms	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.

4.2.3 128 KHz Internal Oscillator

The 128 KHz internal Oscillator is a low power Oscillator providing a clock of 128 KHz. The frequency is nominal at 3V and 25°C. This clock may be selected as the system clock by programming CKSEL Fuses or CSEL field as shown in Table 4-1 on page 25.

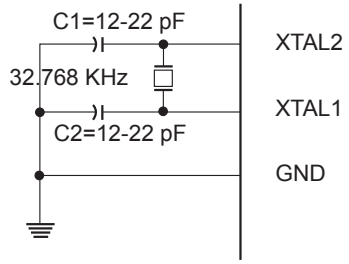
When this clock source is selected, start-up times are determined by the SUT Fuses or by CSUT field as shown in Table 4-5.

Table 4-5. Start-up Times for the 128 kHz Internal Oscillator

SUT[1:0][1] CSUT[1:0][2]	Start-up Time from Power-down/save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00[3]	6 CK	14CK	BOD enabled
01	6 CK	14CK + 4.1 ms	Fast rising power
10	6 CK	14CK + 65 ms	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

Figure 4-3. Low-frequency Crystal Oscillator Connections



12-22 pF capacitors may be necessary if parasitic impedance (pads, wires & PCB) is very low.

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 Low Frequency Crystal Oscillator Clock Selection

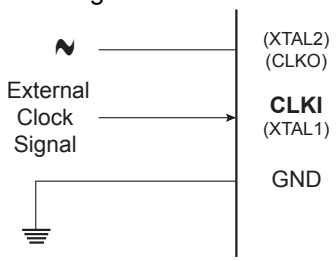
SUT[1:0](1) CSUT[1:0](2)	Start-up Time from Power-down/save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended usage
00	1K (1024) CK(3)	4.1 ms	Fast rising power or BOD enabled
01	1K (1024) CK(3)	65 ms	Slowly rising power
10	32K (32768) CK	65 ms	Stable frequency at start-up
11	Reserved		

Notes: 1. Flash Fuse bits.
2. CLKSELR register bits.
3. These options should only be used if frequency stability at start-up is not important for the application.

4.2.6 External Clock

To drive the device from this external clock source, CLKI should be driven as shown in Figure 4-4. To run the device on an external clock, the CKSEL Fuses or CSEL field must be programmed as shown in Table 4-1 on page 25.

Figure 4-4. External Clock Drive Configuration



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

This external clock can be used by the asynchronous timer if the high or low frequency Crystal Oscillator is not running (See "Enable/Disable Clock Source" on page 32.). The asynchronous timer is then able to enable this input.

– ‘*Enable Watchdog in Automatic Reload Mode*’.

- **Command status return.** The ‘*Request Clock Availability* ’ command returns status via the CLKRDY bit in the CLKCSR register. The ‘*Recover System Clock Source* ’ command returns a code of the current clock source in the CLKSELR register. This information is used in the supervisory software routines as shown in Section 4.3.7 on page 33.

4.3.2 CLKSELR Register

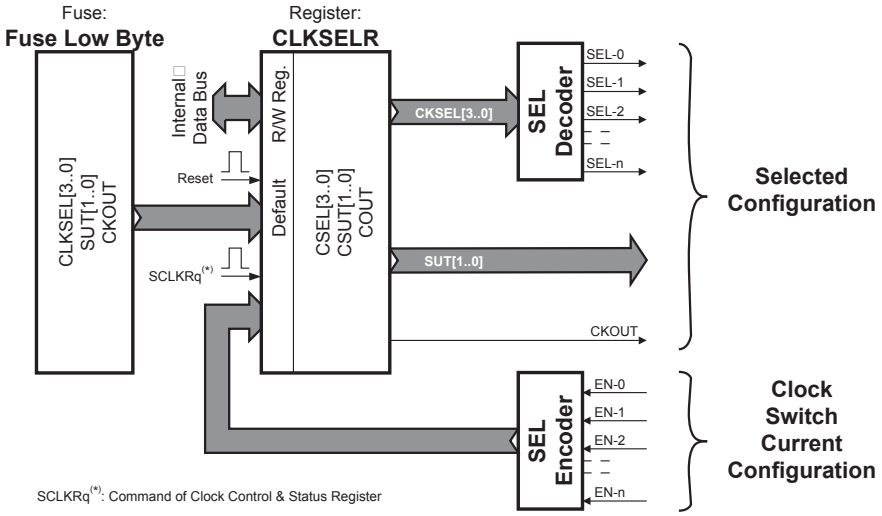
4.3.2.1 Fuses Substitution

At reset, bits of the Low Fuse Byte are copied into the CLKSELR register. The content of this register can subsequently be user modified to overwrite the default values from the Low Fuse Byte. CKSEL[3:0], SUT[1:0] and CKOUT fuses correspond respectively to CSEL[3:0], CSUT[1:0] and ~(COUT) bits of the CLKSELR register as shown in Figure 4-5 on page 32.

4.3.2.2 Source Selection

The available codes of clock source are given in Table 4-1 on page 25.

Figure 4-5. Fuses substitution and Clock Source Selection



The CLKSELR register contains the CSEL, CSUT and COUT values which will be used by the ‘Enable/Disable Clock Source’, ‘Request for Clock Availability’ or ‘Clock Source Switching’ commands.

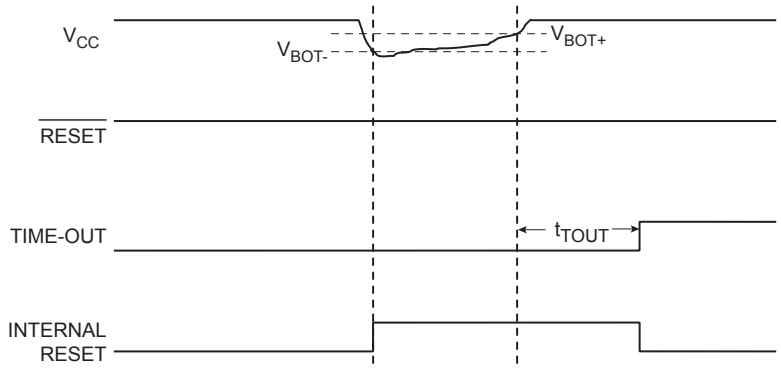
4.3.2.3 Source Recovering

The ‘*Recover System Clock Source*’ command updates the CKSEL field of CLKSELR register (See “System Clock Source Recovering” on page 33.).

4.3.3 Enable/Disable Clock Source

The ‘*Enable Clock Source*’ command selects and enables a clock source configured by the settings in the CLKSELR register. CSEL[3:0] will select the clock source and CSUT[1:0] will select the start-up time (just as CKSEL and SUT fuse bits do). To be sure that a clock source is operating, the ‘*Request for Clock Availability* ’ command must be executed after the ‘*Enable Clock Source*’ command. This will indicate via the CLKRDY bit in the CLKCSR register that a valid clock source is available and operational.

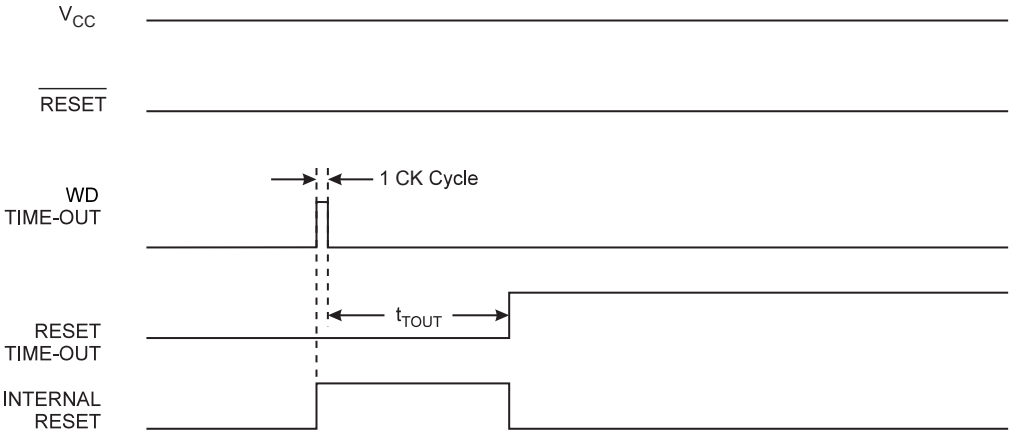
Figure 6-5. Brown-out Reset During Operation



6.1.6 Watchdog System Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT}. Refer to page 53 for details on operation of the Watchdog Timer.

Figure 6-6. Watchdog System Reset During Operation



6.1.7 MCUSR – MCU Status Register

The MCU Status Register provides information on which reset source caused an MCU reset.

Bit	7	6	5	4	3	2	1	0	
0x34 (0x54)	–	–	–	–	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bits 7:4 – Res: Reserved Bits**

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

- **Bit 3 – WDRF: Watchdog System Reset Flag**

This bit is set if a Watchdog System Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 2 – BORF: Brown-out Reset Flag**

This bit is set if a Brown-out Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

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**

7.2 Program Setup in ATtiny87

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATtiny87 is (2-byte step - using “rjmp” instruction):

```
Address(Note:) Label  Code                  Comments
0x0000         rjmp   RESET                 ; Reset Handler
0x0001         rjmp   INT0addr              ; IRQ0 Handler
0x0002         rjmp   INT1addr              ; IRQ1 Handler
0x0003         rjmp   PCINT0addr            ; PCINT0 Handler
0x0004         rjmp   PCINT1addr            ; PCINT1 Handler
0x0005         rjmp   WDTaddr               ; Watchdog Timer Handler
0x0006         rjmp   ICP1addr              ; Timer1 Capture Handler
0x0007         rjmp   OC1Aaddr              ; Timer1 Compare A Handler
0x0008         rjmp   OC1Baddr              ; Timer1 Compare B Handler
0x0009         rjmp   OVF1addr              ; Timer1 Overflow Handler
0x000A         rjmp   OC0Aaddr              ; Timer0 Compare A Handler
0x000B         rjmp   OVF0addr              ; Timer0 Overflow Handler
0x000C         rjmp   LINTCaddr             ; LIN Transfer Complete Handler
0x000D         rjmp   LINERRaddr            ; LIN Error Handler
0x000E         rjmp   SPIaddr               ; SPI Transfer Complete Handler
0x000F         rjmp   ADCCaddr              ; ADC Conversion Complete Handler
0x0010         rjmp   ERDYaddr              ; EEPROM Ready Handler
0x0011         rjmp   ACIaddr               ; Analog Comparator Handler
0x0012         rjmp   USISTARTaddr          ; USI Start Condition Handler
0x0013         rjmp   USIOVFaddr            ; USI Overflow Handler

0x0014 RESET:  ldi    r16, high(RAMEND); Main program start
0x0015         out    SPH,r16              ; Set Stack Pointer to top of RAM
0x0016         ldi    r16, low(RAMEND)
0x0017         out    SPL,r16
0x0018         sei                         ; Enable interrupts
0x0019         <instr>  xxx
  ...     ...    ...     ...
```

Note: 16-bit address

Table 9-1 summarizes the control signals for the pin value.

Table 9-1. Port Pin Configurations

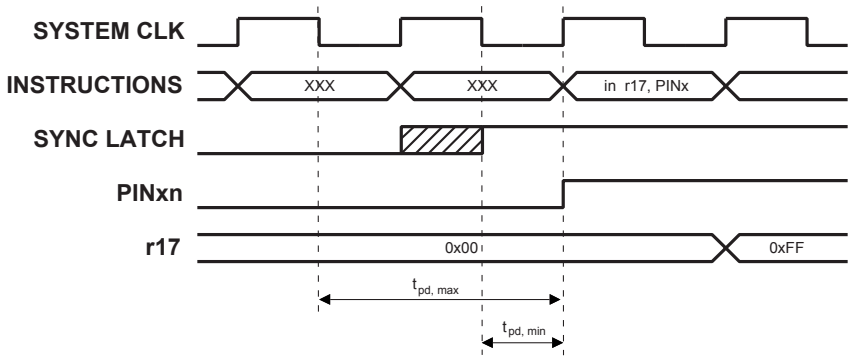
DDxn	PORTxn	PUD (in MCUCR) [1]	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

Note: 1. Or port-wise PUDx bit in PORTCR register.

9.2.5 Reading the Pin Value

Independent of the setting of Data Direction bit DDxn, the port pin can be read through the PINxn Register bit. As shown in Figure 9-2, the PINxn Register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay. Figure 9-4 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted $t_{pd,max}$ and $t_{pd,min}$ respectively.

Figure 9-4. Synchronization when Reading an Externally Applied Pin value



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 9-5. 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.

9.3.3 Alternate Functions of Port A

The Port A pins with alternate functions are shown in Table 9-3.

Table 9-3. Port A Pins Alternate Functions

Port Pin	Alternate Function
PA7	PCINT7 (Pin Change Interrupt 7) ADC7 (ADC Input Channel 7) AIN1 (Analog Comparator Positive Input) XREF (Internal Voltage Reference Output) AREF (External Voltage Reference Input)
PA6	PCINT6 (Pin Change Interrupt 6) ADC6 (ADC Input Channel 6) AIN0 (Analog Comparator Negative Input) $\overline{SS}$ (SPI Slave Select Input)
PA5	PCINT5 (Pin Change Interrupt 5) ADC5 (ADC Input Channel 5) T1 (Timer/Counter1 Clock Input) USCK (Three-wire Mode USI *Alternate* Clock Input) SCL (Two-wire Mode USI *Alternate* Clock Input) SCK (SPI Master Clock)
PA4	PCINT4 (Pin Change Interrupt 4) ADC4 (ADC Input Channel 4) ICP1 (Timer/Counter1 Input Capture Trigger) DI (Three-wire Mode USI *Alternate* Data Input) SDA (Two-wire Mode USI *Alternate* Data Input / Output) MOSI (SPI Master Output / Slave Input)
PA3	PCINT3 (Pin Change Interrupt 3) ADC3 (ADC Input Channel 3) ISRC (Current Source Pin) INT1 (External Interrupt1 Input)
PA2	PCINT2 (Pin Change Interrupt 2) ADC2 (ADC Input Channel 2) OC0A (Output Compare and PWM Output A for Timer/Counter0) DO (Three-wire Mode USI *Alternate* Data Output) MISO (SPI Master Input / Slave Output)
PA1	PCINT1 (Pin Change Interrupt 1) ADC1 (ADC Input Channel 1) TXD (UART Transmit Pin) TXLIN (LIN Transmit Pin)
PA0	PCINT0 (Pin Change Interrupt 0) ADC0 (ADC Input Channel 0) RXD (UART Receive Pin) RXLIN (LIN Receive Pin)

10.8 Timer/Counter Timing Diagrams

The following figures show the Timer/Counter in synchronous mode, and the timer clock (clk_T0) is therefore shown as a clock enable signal. In asynchronous mode, $clk_{I/O}$ should be replaced by the Timer/Counter Oscillator clock. The figures include information on when interrupt flags are set. Figure 10-8 contains timing data for basic Timer/Counter operation. The figure shows the count sequence close to the MAX value in all modes other than phase correct PWM mode.

Figure 10-8. Timer/Counter Timing Diagram, no Prescaling

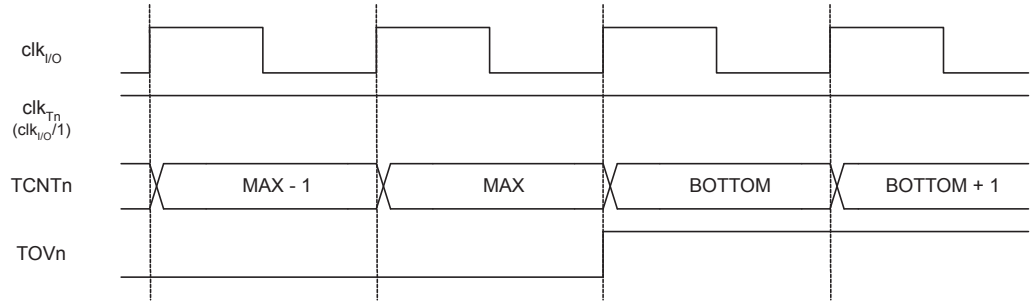


Figure 10-9 shows the same timing data, but with the prescaler enabled.

Figure 10-9. Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}/8$)

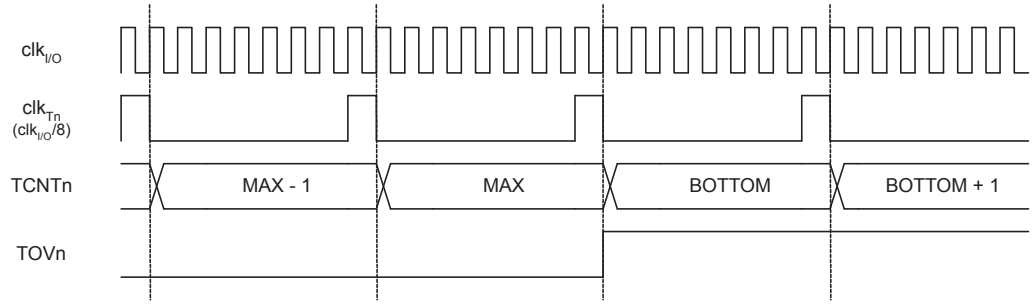
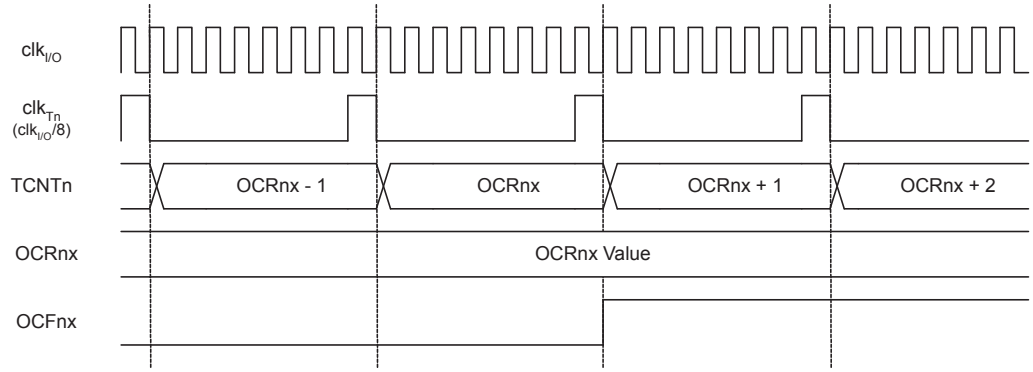


Figure 10-10 shows the setting of OCF0A in all modes except CTC mode.

Figure 10-10. Timer/Counter Timing Diagram, Setting of OCF0A, with Prescaler ($f_{clk\_I/O}/8$)



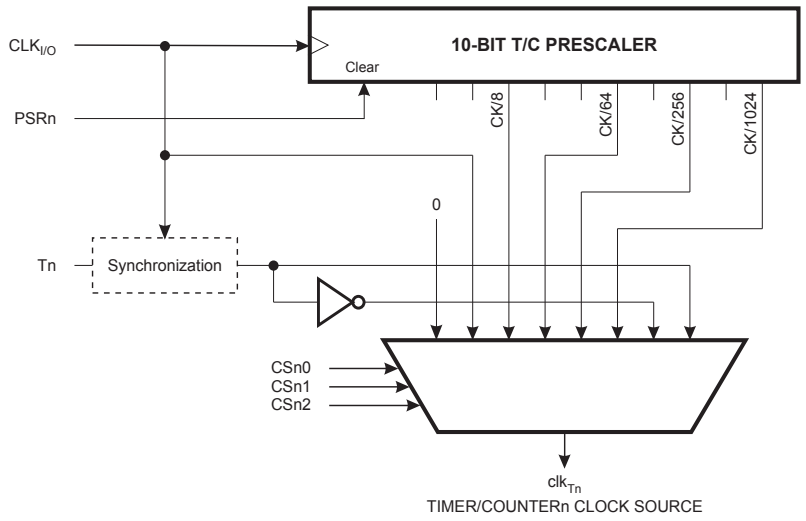
The synchronization and edge detector logic introduces a delay of 2.5 to 3.5 system clock cycles from an edge has been applied to the T1 pin to the counter is updated.

Enabling and disabling of the clock input must be done when T1 has been stable for at least one system clock cycle, otherwise it is a risk that a false Timer/Counter clock pulse is generated.

Each half period of the external clock applied must be longer than one system clock cycle to ensure correct sampling. The external clock must be guaranteed to have less than half the system clock frequency ($f_{ExtClk} < f_{clk\_I/O}/2$) given a 50/50 % duty cycle. Since the edge detector uses sampling, the maximum frequency of an external clock it can detect is half the sampling frequency (Nyquist sampling theorem). However, due to variation of the system clock frequency and duty cycle caused by Oscillator source (crystal, resonator, and capacitors) tolerances, it is recommended that maximum frequency of an external clock source is less than $f_{clk\_I/O}/2.5$.

An external clock source can not be prescaled.

Figure 11-2. Prescaler for Timer/Counter1 (1)



Note: 1. The synchronization logic on the input pin (T1) is shown in Figure 11-1.

12.11.2 TCCR1B – Timer/Counter1 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x81)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ICNC1: Input Capture Noise Canceler**

Setting this bit (to one) activates the Input Capture Noise Canceler. When the noise canceler is activated, the input from the Input Capture pin (ICP1) is filtered. The filter function requires four successive equal valued samples of the ICP1 pin for changing its output. The Input Capture is therefore delayed by four Oscillator cycles when the noise canceler is enabled.

- **Bit 6 – ICES1: Input Capture Edge Select**

This bit selects which edge on the Input Capture pin (ICP1) that is used to trigger a capture event. When the ICES1 bit is written to zero, a falling (negative) edge is used as trigger, and when the ICES1 bit is written to one, a rising (positive) edge will trigger the capture.

When a capture is triggered according to the ICES1 setting, the counter value is copied into the Input Capture Register (ICR1). The event will also set the Input Capture Flag (ICF1), and this can be used to cause an Input Capture Interrupt, if this interrupt is enabled.

When the ICR1 is used as TOP value (see description of the WGM1[3:0] bits located in the TCCR1A and the TCCR1B Register), the ICP1 is disconnected and consequently the Input Capture function is disabled.

- **Bit 5 – Res: Reserved Bit**

This bit is reserved for future use. For ensuring compatibility with future devices, this bit must be written to zero when TCCR1B is written.

- **Bits 4:3 – WGM1[3:2]: Waveform Generation Mode**

See TCCR1A Register description.

- **Bits 2:0 – CS1[2:0]: Clock Select**

The three Clock Select bits select the clock source to be used by the Timer/Counter, see Figure 12-11 and Figure 12-12.

Table 12-5. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/1 (No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

14.5.5 USIPP – USI Pin Position

Bit	7	6	5	4	3	2	1	0	
(0xBC)	–	–	–	–	–	–	–	USIPOS	USIPP
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:1 – Res: Reserved Bits**

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

- **Bit 0 – USIPOS: USI Pin Position**

Setting or clearing this bit changes the USI pin position.

Table 14-3. USI Pin Position

USIPOS		USI Pin Position	
0	PortB (Default)	DI, SDA	PB0 - (PCINT8/OC1AU)
		DO	PB1 - (PCINT9/OC1BU)
		USCK, SCL	PB2 - (PCINT10/OC1AV)
1	Port A (Alternate)	DI, SDA	PA4 - (PCINT4/ADC4/ICP1/MOSI)
		DO	PA2 - (PCINT2/ADC2/OC0A/MISO)
		USCK, SCL	PA5 - (PCINT5/ADC5/T1/SCK)

Gives the number of samples of a bit.

$$\text{sample-time} = (1 / f\text{clk}_{i/o}) \times (\text{LDIV}[11:0] + 1)$$

Default value: LBT[6:0]=32 — Min. value: LBT[6:0]=8 — Max. value: LBT[6:0]=63

15.6.6 LINBRR – LIN Baud Rate Register

Bit	7	6	5	4	3	2	1	0	
(0xCD)	LDIV7	LDIV6	LDIV5	LDIV4	LDIV3	LDIV2	LDIV1	LDIV0	LINBRRL
(0xCE)	–	–	–	–	LDIV11	LDIV10	LDIV9	LDIV8	LINBRRH
Bit	15	14	13	12	11	10	9	8	
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 15:12 – Res: Reserved Bits**

  These bits are reserved for future use. For compatibility with future devices, they must be written to zero when LINBRR is written.

- **Bits 11:0 – LDIV[11:0]: Scaling of $clk_{i/o}$ Frequency**

  The LDIV value is used to scale the entering $clk_{i/o}$ frequency to achieve appropriate LIN or UART baud rate.

15.6.7 LINDLR – LIN Data Length Register

Bit	7	6	5	4	3	2	1	0	
(0xCF)	LTXDL3	LTXDL2	LTXDL1	LTXDL0	LRXDL3	LRXDL2	LRXDL1	LRXDL0	LINDLR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:4 – LTXDL[3:0]: LIN Transmit Data Length**

  In LIN mode, this field gives the number of bytes to be transmitted (clamped to 8 Max).

  In UART mode this field is unused.

- **Bits 3:0 – LRXDL[3:0]: LIN Receive Data Length**

  In LIN mode, this field gives the number of bytes to be received (clamped to 8 Max).

  In UART mode this field is unused.

17.6.1 ADC Input Channels

When changing channel selections, the user should observe the following guidelines to ensure that the correct channel is selected:

In Single Conversion mode, always select the channel before starting the conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the conversion to complete before changing the channel selection.

In Free Running mode, always select the channel before starting the first conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the first conversion to complete, and then change the channel selection. Since the next conversion has already started automatically, the next result will reflect the previous channel selection. Subsequent conversions will reflect the new channel selection.

17.6.2 ADC Voltage Reference

The voltage reference for the ADC (V_{REF}) indicates the conversion range for the ADC. Single ended channels that exceed V_{REF} will result in codes close to 0x3FF. V_{REF} can be selected as either AV_{CC}, internal 1.1V / 2.56V voltage reference or external AREF pin. The first ADC conversion result after switching voltage reference source may be inaccurate, and the user is advised to discard this result.

17.7 ADC Noise Canceler

The ADC features a noise canceler that enables conversion during sleep mode to reduce noise induced from the CPU core and other I/O peripherals. The noise canceler can be used with ADC Noise Reduction and Idle mode. To make use of this feature, the following procedure should be used:

a. Make sure that the ADC is enabled and is not busy converting. Single Conversion mode must be selected and the ADC conversion complete interrupt must be enabled.
b. Enter ADC Noise Reduction mode (or Idle mode). The ADC will start a conversion once the CPU has been halted.
c. If no other interrupts occur before the ADC conversion completes, the ADC interrupt will wake up the CPU and execute the ADC Conversion Complete interrupt routine. If another interrupt wakes up the CPU before the ADC conversion is complete, that interrupt will be executed, and an ADC Conversion Complete interrupt request will be generated when the ADC conversion completes. The CPU will remain in active mode until a new sleep command is executed.

Note that the ADC will not be automatically turned off when entering other sleep modes than Idle mode and ADC Noise Reduction mode. The user is advised to write zero to ADEN before entering such sleep modes to avoid excessive power consumption.

17.7.1 Analog Input Circuitry

The analog input circuitry for single ended channels is illustrated in Figure 17-8. An analog source applied to ADCn is subjected to the pin capacitance and input leakage of that pin, regardless of whether that channel is selected as input for the ADC. When the channel is selected, the source must drive the S/H capacitor through the series resistance (combined resistance in the input path).

The ADC is optimized for analog signals with an output impedance of approximately 10 kΩ or less. If such a source is used, the sampling time will be negligible. If a source with higher impedance is used, the sampling time will depend on how long time the source needs to charge the

17.11 Register Description

17.11.1 ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
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 7:6 – REFS[1:0]: Voltage Reference Selection Bits**

These bits and AREFEN bit from the Analog Miscellaneous Control Register (AMISCR) select the voltage reference for the ADC, as shown in Table 17-4. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA register is set). Whenever these bits are changed, the next conversion will take 25 ADC clock cycles. If active channels are used, using AVCC or an external AREF higher than (AV_{CC} - 1V) is not recommended, as this will affect ADC accuracy. The internal voltage reference options may not be used if an external voltage is being applied to the AREF pin.

Table 17-4. Voltage Reference Selections for ADC

REFS1	REFS0	AREFEN	Voltage Reference (V_{REF}) Selection
X	0	0	AV_{CC} used as Voltage Reference, diconnected from AREF pin
X	0	1	External Voltage Reference at AREF pin (AREF ≥ 2.0V)
0	1	0	Internal 1.1V Voltage Reference
1	1	0	Internal 2.56V Voltage Reference

- **Bit 5 – ADLAR: ADC Left Adjust Result**

The ADLAR bit affects the presentation of the ADC conversion result in the ADC Data Register. Write one to ADLAR to left adjust the result. Otherwise, the result is right adjusted. Changing the ADLAR bit will affect the ADC Data Register immediately, regardless of any ongoing conversions. For a complete description of this bit, see “ADCL and ADCH – ADC Data Register” on page 207.

- **Bits 4:0 – MUX[4:0]: Analog Channel and Gain Selection Bits**

These bits select which combination of analog inputs are connected to the ADC. In case of differential input, gain selection is also made with these bits. Refer to Table 17-5 for details. If these bits are changed during a conversion, the change will not go into effect until this conversion is complete (ADIF in ADCSRA register is set).

Table 17-6. ADC Prescaler Selections (Continued)

ADPS2	ADPS1	ADPS0	Division Factor
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

17.11.3 ADCL and ADCH – ADC Data Register

17.11.3.1 ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
(0x79)	–	–	–	–	–	–	ADC9	ADC8	ADCH
(0x78)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

17.11.3.2 ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
(0x79)	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
(0x78)	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

When an ADC conversion is complete, the result is found in these two registers.

When ADCL is read, the ADC Data Register is not updated until ADCH is read. Consequently, if the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH.

The ADLAR bit in ADMUX, and the MUXn bits in ADMUX affect the way the result is read from the registers. If ADLAR is set, the result is left adjusted. If ADLAR is cleared (default), the result is right adjusted.

- **ADC[9:0]: ADC Conversion Result**

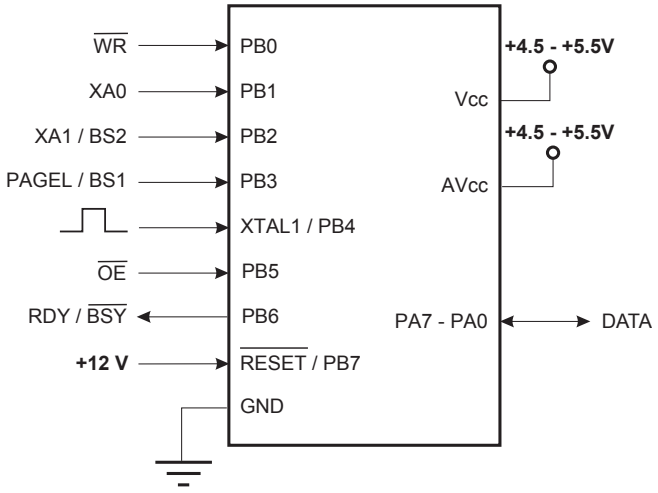
These bits represent the result from the conversion, as detailed in “ADC Conversion Result” on page 201.

20.2.7 Simple Assembly Code Example for a Boot Loader

Note that the RWWSB bit will always be read as zero in ATtiny87/167. Nevertheless, it is recommended to check this bit as shown in the code example, to ensure compatibility with devices supporting Read-While-Write.

```
  ;- The routine writes one page of data from RAM to Flash
  ;  the first data location in RAM is pointed to by the Y-pointer
  ;  the first data location in Flash is pointed to by the Z-pointer
  ;- Error handling is not included
  ;- Registers used: r0, r1, temp1 (r16), temp2 (r17), looplo (r24),
  ;                  loophi (r25), spmcsrval (r20)
  ; -Storing and restoring of registers is not included in the routine
  ;  register usage can be optimized at the expense of code size

.equ PAGESIZEB = PAGESIZE*2   ; AGESIZEB is page size in BYTES, not words
.org SMALLBOOTSTART

Write_page:
  ; Page Erase
  ldi    spmcsrval, (1<<PGERS) | (1<<SELFPGEN)
  rcall  Do_spm

  ; Clear temporary page buffer
  ldi    spmcsrval, (1<<CPTB) | (1<<SELFPGEN)
  rcall  Do_spm

  ; Transfer data from RAM to Flash temporary page buffer
  ldi    looplo, low(PAGESIZEB)    ; init loop variable
  ldi    loophi, high(PAGESIZEB)   ; not required for PAGESIZEB<=256

Wrloop:
  ld     r0, Y+
  ld     r1, Y+
  ldi    spmcsrval, (1<<SELFPGEN)
  rcall  Do_spm
  adiw   ZH:ZL, 2
  sbiw   loophi:looplo, 2          ; use subi for PAGESIZEB<=256
  brne   Wrloop

  ; Execute Page Write
  subi   ZL, low(PAGESIZEB)        ; restore pointer
  sbci   ZH, high(PAGESIZEB)       ; not required for PAGESIZEB<=256
  ldi    spmcsrval, (1<<PGWRT) | (1<<SELFPGEN)
  rcall  Do_spm

  ; Clear temporary page buffer
  ldi    spmcsrval, (1<<CPTB) | (1<<SELFPGEN)
  rcall  Do_spm

  ; Read back and check, optional
  ldi    looplo, low(PAGESIZEB)    ; init loop variable
  ldi    loophi, high(PAGESIZEB)   ; not required for PAGESIZEB<=256
  subi   YL, low(PAGESIZEB)        ; restore pointer
  sbci   YH, high(PAGESIZEB)

Rdloop:
  lpm    r0, Z+
```

Figure 21-1. Parallel programming



Note: V_{CC} - 0.3V < AV_{CC} < Vcc + 0.3V, however, AV_{CC} should always be within 4.5 - 5.5V

Table 21-9. Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
$\overline{WR}$	PB0	I	Write Pulse (Active low).
XA0	PB1	I	XTAL1 Action Bit 0
XA1 / BS2	PB2	I	- XTAL1 Action Bit 1 - Byte Select 2 ("0" selects low byte, "1" selects 2'nd high byte)
PAGEL / BS1	PB3	I	- Program Memory and EEPROM data Page Load - Byte Select 1 ("0" selects low byte, "1" selects high byte)
⎍	PB4	I	XTAL1 (Clock input)
$\overline{OE}$	PB5	I	Output Enable (Active low).
RDY / $\overline{BSY}$	PB6	O	0: Device is busy programming, 1: Device is ready for new command.
+12V	PB7	I	- Reset (Active low) - Parallel programming mode (+12V).
DATA	PA[7:0]	I/O	Bi-directional Data bus (Output when $\overline{OE}$ is low).

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

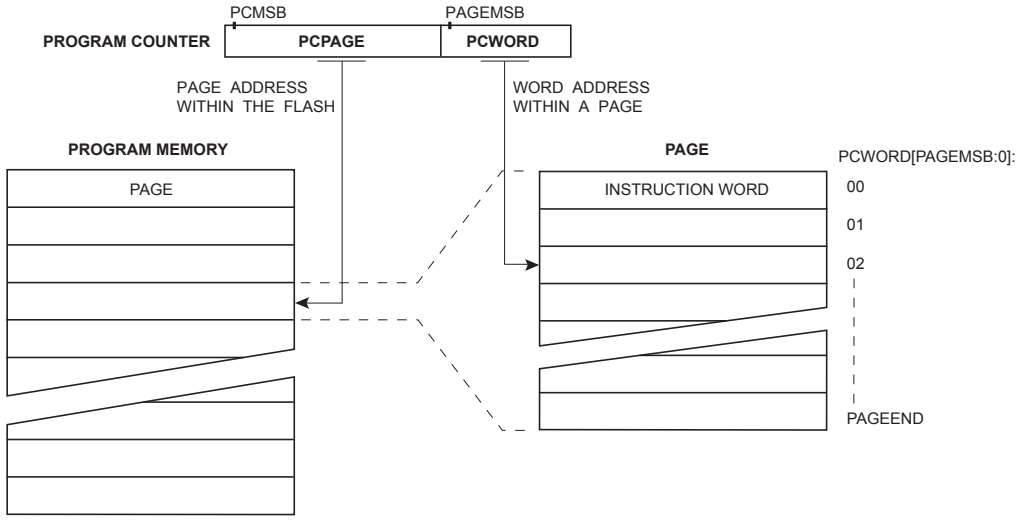
1. Give $\overline{WR}$ a negative pulse. This starts programming of the entire page of data. RDY/$\overline{BSY}$ goes low.
2. Wait until RDY/$\overline{BSY}$ goes high (See Figure 21-3 for signal waveforms).

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