Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

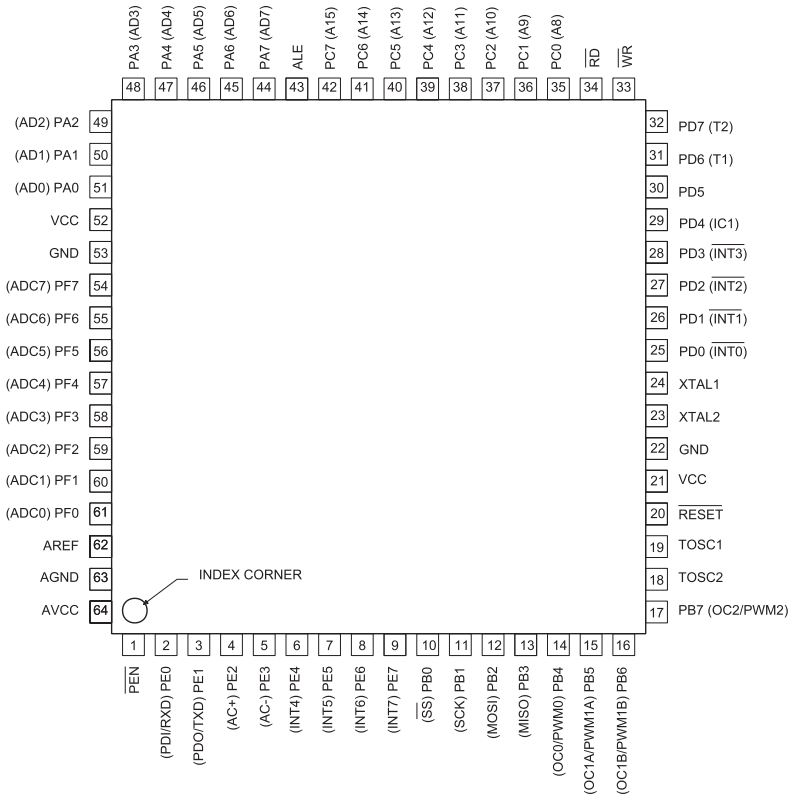
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded"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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	4MHz
Connectivity	SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	32
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega103l-4ac

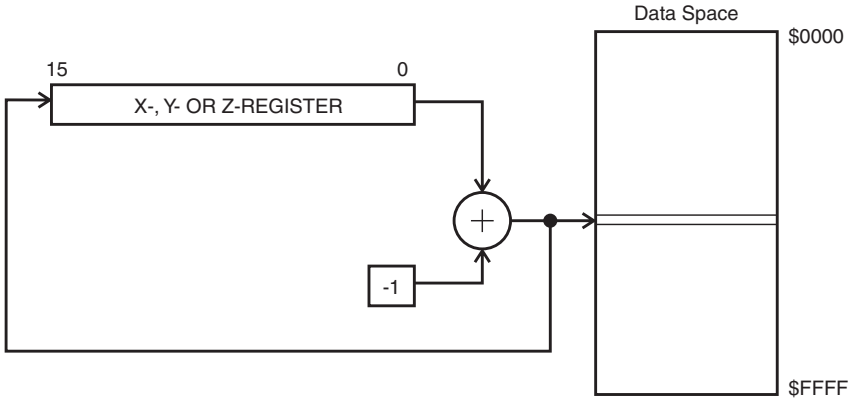
Pin Configuration

TQFP



Data Indirect with Pre-decrement

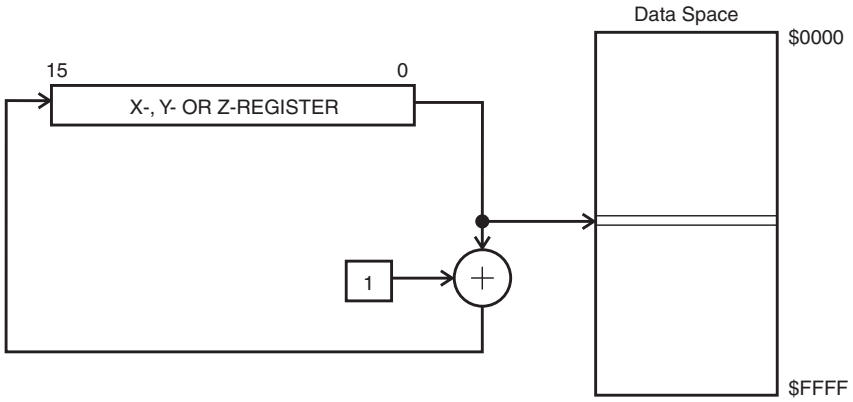
Figure 14. Data Indirect Addressing with Pre-decrement



The X-, Y-, or the Z-register is decremented before the operation. Operand address is the decremented contents of the X-, Y-, or the Z-register.

Data Indirect with Post-increment

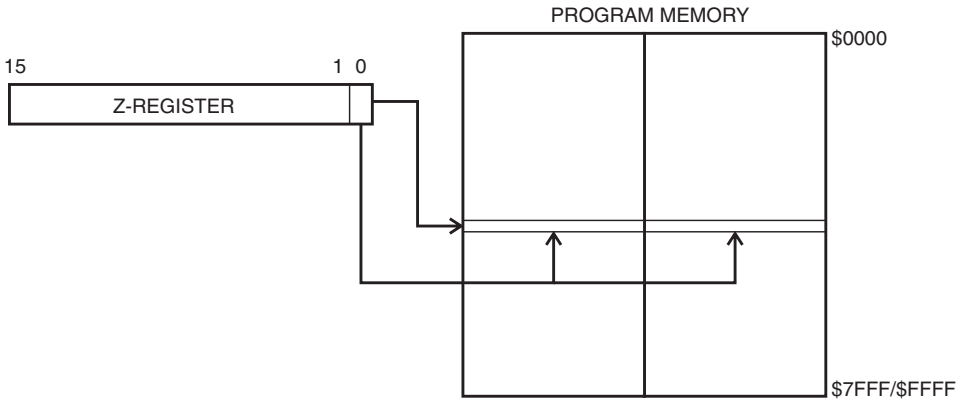
Figure 15. Data Indirect Addressing with Post-increment



The X-, Y-, or the Z-register is incremented after the operation. Operand address is the contents of the X-, Y-, or the Z-register prior to incrementing.

Constant Addressing Using the LPM and ELPM Instructions

Figure 16. Code Memory Constant Addressing

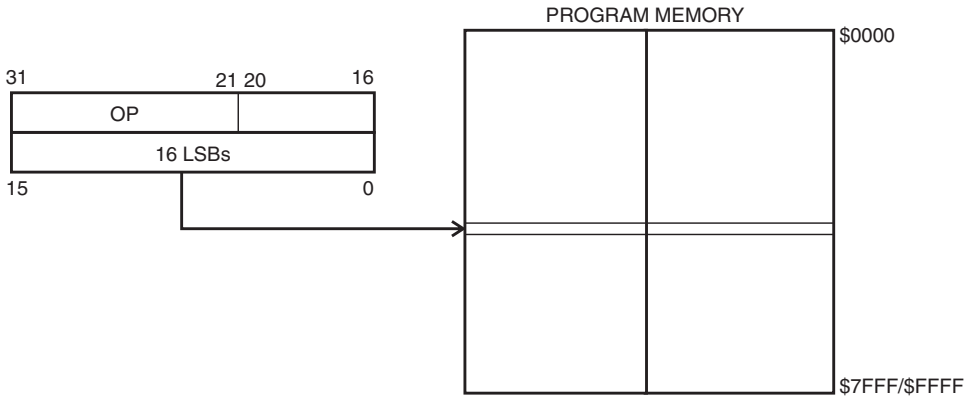


Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 32K), LSB selects Low Byte if cleared (LSB = 0) or High Byte if set (LSB =

1). If ELPM is used, LSB of the RAM Page Z register (RAMPZ) is used to select low or high memory page (RAMPZ0 = 0: Low Page, RAMPZ0 = 1: High Page).

Direct Program Address, JMP and CALL

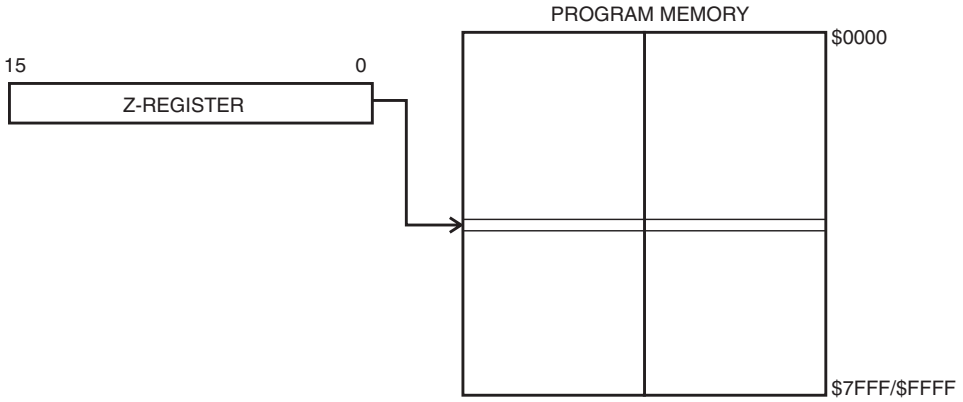
Figure 17. Direct Program Memory Addressing



Program execution continues at the address immediate in the instruction words.

Indirect Program Addressing, IJMP and ICALL

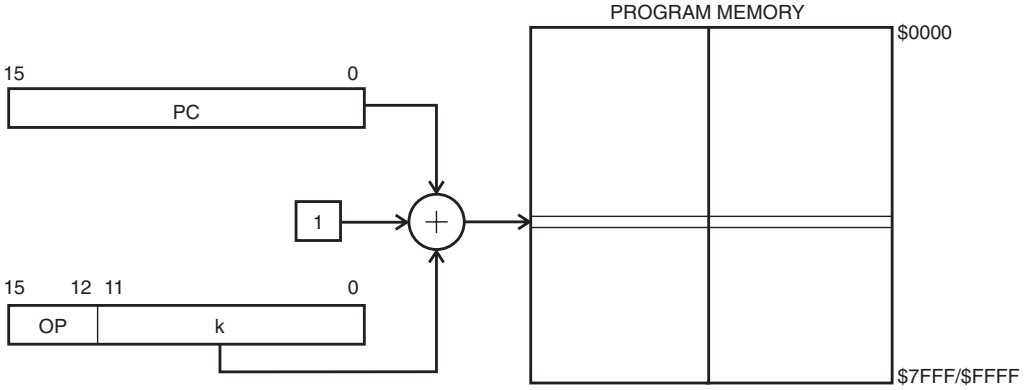
Figure 18. Indirect Program Memory Addressing



Program execution continues at address contained by the Z-register (i.e., the PC is loaded with the contents of the Z-register).

Relative Program Addressing, RJMP and RCALL

Figure 19. Relative Program Memory Addressing

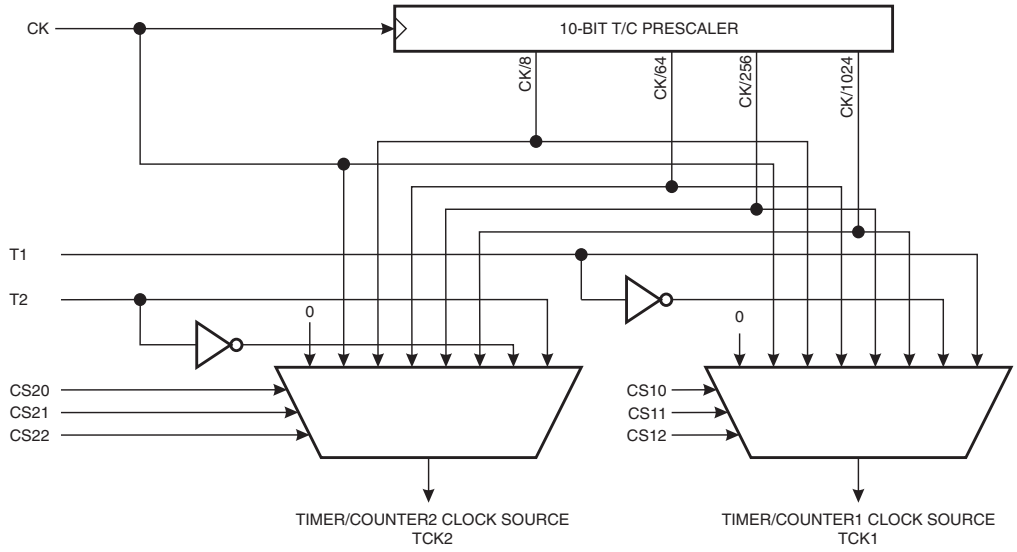


Timer/Counters

The ATmega103(L) provides three general purpose Timer/Counters – two 8-bit T/Cs and one 16-bit T/C. Timer/Counter0 optionally can be asynchronously clocked from an external Oscillator. This Oscillator is optimized for use with a 32.768 kHz crystal, enabling use of Timer/Counter0 as a Real Time Clock (RTC). Timer/Counter0 has its own prescaler. Timer/Counters 1 and 2 have individual prescaling selection from the same 10-bit prescaling timer. These Timer/Counters can either be used as a Timer with an internal clock time base or as a counter with an external pin connection that triggers the counting.

Timer/Counter Prescalers

Figure 28. Prescaler for Timer/Counter1 and Timer/Counter2



For Timer/Counters 1 and 2, the four different prescaled selections are: CK/8, CK/64, CK/256 and CK/1024, where CK is the CPU clock. Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled. For Timer/Counters 1 and 2, added selections as CK, external source and stop can be selected as clock sources.

Figure 29. The Timer/Counter0 Prescaler

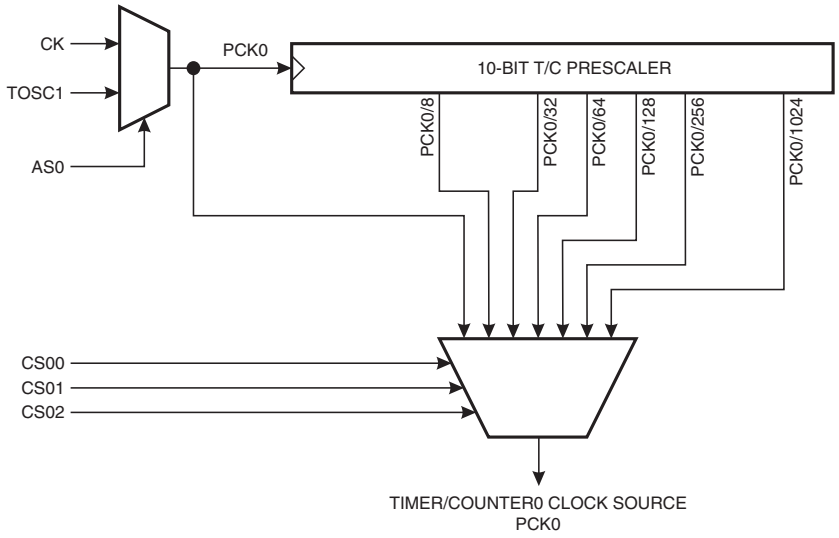
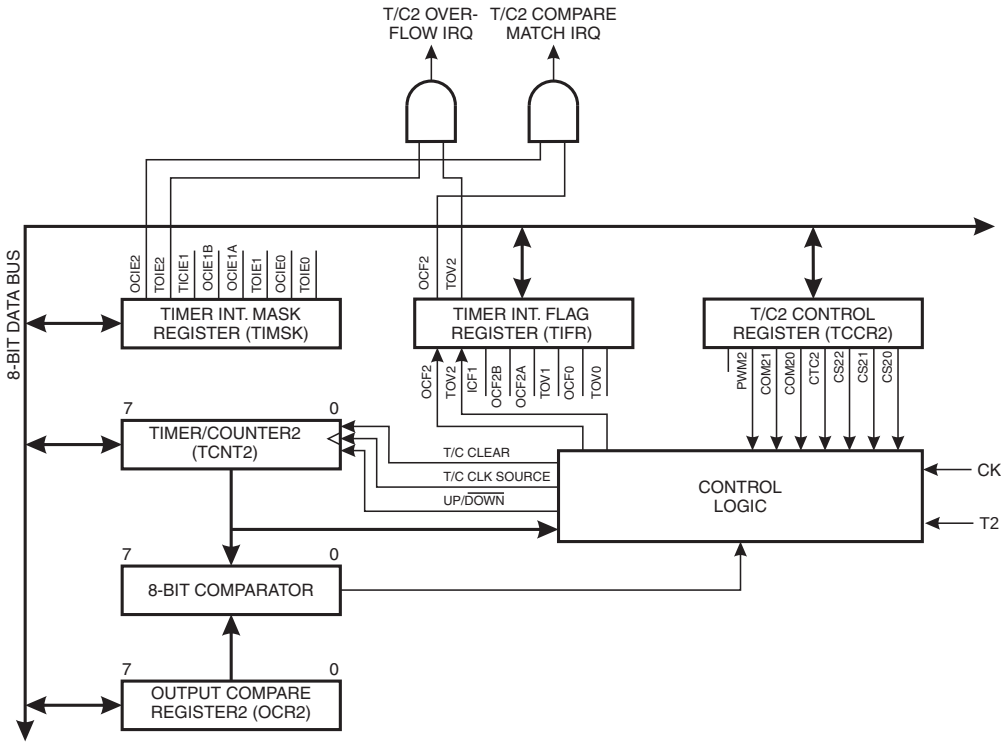


Figure 31. Timer/Counter2 Block Diagram



The 8-bit Timer/Counter0 can select clock source from PCK0 or prescaled PCK0. The 8-bit Timer/Counter2 can select clock source from CK, prescaled CK or an external pin. Both Timer/Counters can be stopped as described in the specification for the Timer/Counter Control Registers – TCCR0 and TCCR2.

The different Status Flags (Overflow, Compare Match and Capture Event) are found in the Timer/Counter Interrupt Flag Register (TIFR). Control signals are found in the Timer/Counter Control Registers – TCCR0 and TCCR2. The interrupt enable/disable settings are found in the Timer/Counter Interrupt Mask Register (TIMSK).

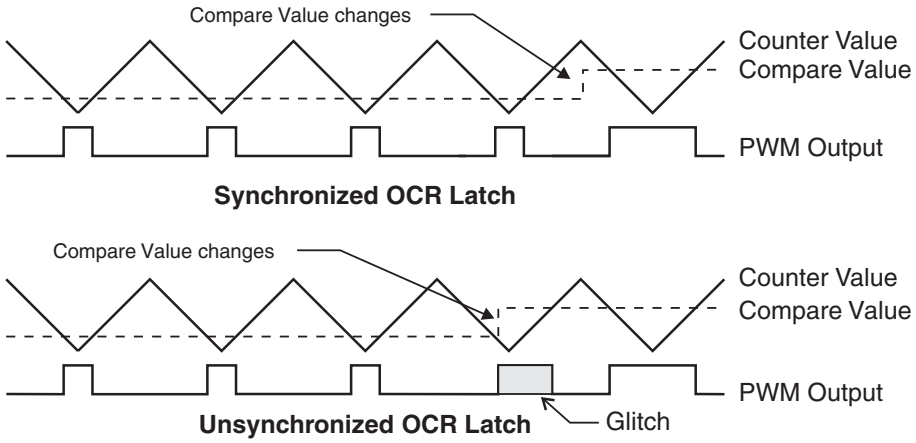
When Timer/Counter2 is externally clocked, the external signal is synchronized with the Oscillator frequency of the CPU. To assure proper sampling of the external clock, the minimum time between two external clock transitions must be at least one internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 8-bit Timer/Counters feature a high-resolution and a high-accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities make these units useful for lower speed functions or exact timing functions with infrequent actions.

Both Timer/Counters support two Output Compare functions using the Output Compare Registers (OCR0 and OCR2) as the data source to be compared to the Timer/Counter contents. The Output Compare functions include optional clearing of the counter on compare match and action on the Output Compare pins – PB4(OC0/PWM0) and PB7(OC2/PWM2) – on compare match.

Timer/Counters 0 and 2 can also be used as 8-bit Pulse Width Modulators. In this mode the Timer/Counter and the Output Compare Register serve as a glitch-free, stand-alone PWM with centered pulses. Refer to page 43 for a detailed description of this function.

Figure 32. Effects on Unsynchronized OCR Latching



During the time between the write and the latch operation, a read from OCR0 or OCR2 will read the contents of the temporary location. This means that the most recently written value always will read out of OCR0/2.

When the OCR Register (not the temporary register) is updated to $00 or $FF, the PWM output changes to low or high immediately according to the settings of COM21/COM20 or COM11/COM10. This is shown in Table 14.

Table 14. PWM Outputs OCRn = $00 or $FF

COMn1	COMn0	OCRn	Output PWMn
1	0	$00	L
1	0	$FF	H
1	1	$00	H
1	1	$FF	L

Note: n = 0 or 2

In PWM mode, the Timer Overflow Flag, TOV0 or TOV2, is set when the counter advances from $00. Timer Overflow Interrupts 0 and 2 operate exactly as in normal Timer/Counter mode, i.e., it is executed when TOV0 or TOV2 is set, provided that Timer Overflow interrupt and Global Interrupts are enabled. This also applies to the Timer Output Compare Flags and interrupts.

The frequency of the PWM will be Timer Clock Frequency divided by 510.

Asynchronous Status Register – ASSR

Bit	7	6	5	4	3	2	1	0	
$30 ($50)	–	–	–	–	AS0	TCN0UB	OCR0UB	TCR0UB	ASSR
Read/Write	R	R	R	R	R/W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega103(L) and always read as zero.

- **Bits 2, 1, 0 – CS12, CS11, CS10: Clock Select1, Bits 2, 1 and 0**

The lock Select1 bits 2, 1 and 0 define the prescaling source of Timer/Counter1.

Table 17. Clock1 Prescale Select

CS12	CS11	CS10	Description
0	0	0	Stop, the Timer/Counter1 is stopped.
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T1, falling edge
1	1	1	External Pin T1, rising edge

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK CPU clock. If the external pin modes are used for Timer/Counter1, transitions on PD6/(T1) will clock the counter even if the pin is configured as an output. This feature can give the user software control of the counting.

Timer/Counter1 – TCNT1H and TCNT1L

Bit	15	14	13	12	11	10	9	8	
$2D ($4D)	MSB								TCNT1H
$2C ($4C)								LSB	TCNT1L
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	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	
	0	0	0	0	0	0	0	0	

This 16-bit register contains the prescaled value of the 16-bit Timer/Counter1. 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 register (TEMP). This temporary register is also used when accessing OCR1A, OCR1B and ICR1. If the main program and interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program (and from interrupt routines if interrupts are allowed from within interrupt routines).

- TCNT1 Timer/Counter1 Write:

  When the CPU writes to the High Byte TCNT1H, the written data is placed in the TEMP Register. Next, when the CPU writes the Low Byte TCNT1L, this byte of data is combined with the byte data in the TEMP Register, and all 16 bits are written to the TCNT1 Timer/Counter1 Register simultaneously. Consequently, the High Byte TCNT1H must be accessed first for a full 16-bit register write operation. When using Timer/Counter1 as an 8-bit Timer, it is sufficient to write the Low Byte only.

- TCNT1 Timer/Counter1 Read:

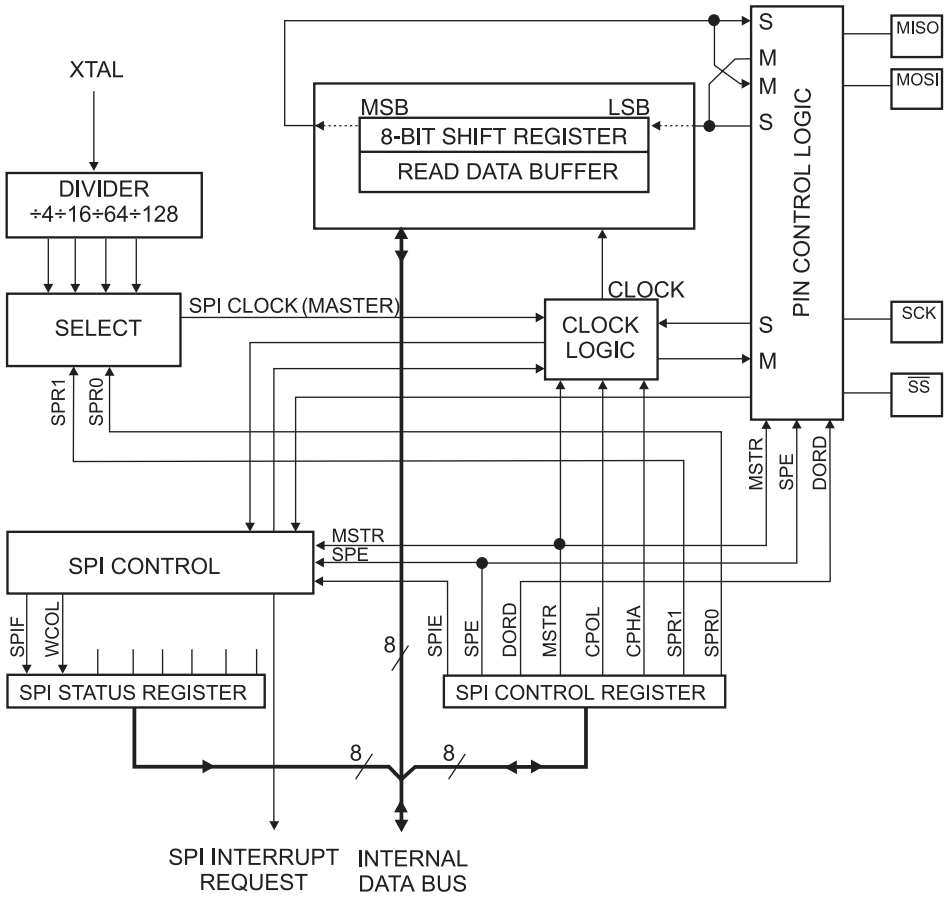
  When the CPU reads the Low Byte TCNT1L, the data of TCNT1L is sent to the CPU and the data of the High Byte TCNT1H is placed in the TEMP Register. When the CPU reads the data in the High Byte TCNT1H, the CPU receives the data in the TEMP Register. Consequently, the Low Byte TCNT1L must be accessed first for a

Serial Peripheral Interface – SPI

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the ATmega103(L) and peripheral devices or between several AVR devices. The ATmega103(L) SPI features include the following:

- **Full-duplex, Three-wire Synchronous Data Transfer**
- **Master or Slave Operation**
- **LSB First or MSB First Data Transfer**
- **Four Programmable Bit Rates**
- **End-of-Transmission Interrupt Flag**
- **Write Collision Flag Protection**
- **Wake-up from Idle Mode (Slave Mode only)**

Figure 37. SPI Block Diagram



The interconnection between Master and Slave CPUs with SPI is shown in Figure 38. The PB1 (SCK) pin is the clock output in the Master mode and is the clock input in the Slave mode. Writing to the SPI Data Register of the Master CPU starts the SPI clock generator, and the data written shifts out of the PB2 (MOSI) pin and into the PB2 (MOSI) pin of the Slave CPU. After shifting one byte, the SPI clock generator stops, setting the End-of-Transmission Flag (SPIF). If the SPI interrupt enable bit (SPIE) in the SPCR Register is set, an interrupt is requested. The Slave Select input, PB0($\overline{SS}$), is set low to select an individual Slave SPI device. The two Shift Registers in the Master and the Slave can be considered as one distributed 16-bit circular Shift Register. This is shown in Figure 38. When data is shifted from the Master to the Slave, data is also shifted in the opposite direction, simultaneously. This means that during one shift cycle, data in the Master and the Slave are interchanged.

SPI Data Register – SPDR

Bit	7	6	5	4	3	2	1	0	
$0F ($2F)	MSB							LSB	SPDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	X	X	X	X	X	X	X	X	Undefined

The SPI Data Register is a read/write register used for data transfer between the Register File and the SPI Shift Register. Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be read.

Analog-to-Digital Converter

Feature list:

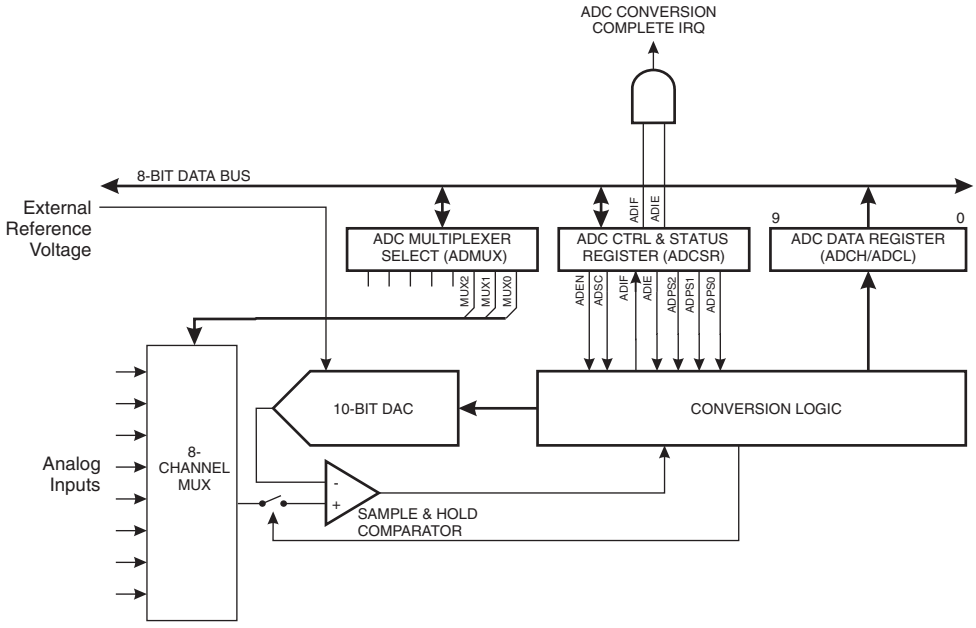
- **10-bit Resolution**
- **±2 LSB Absolute Accuracy**
- **0.5 LSB Integral Non-linearity**
- **70 - 280 µs Conversion Time**
- **Up to 14 kSPS**
- **8 Multiplexed Input Channels**
- **Interrupt on ADC Conversion Complete**
- **Sleep Mode Noise Canceler**

The ATmega103(L) features a 10-bit successive approximation ADC. The ADC is connected to an 8-channel Analog Multiplexer, which allows each pin of Port F to be used as an input for the ADC. The ADC contains a Sample and Hold Amplifier, 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 45.

The ADC has two separate analog supply voltage pins, AV_{CC} and AGND. AGND must be connected to GND, and the voltage on AV_{CC} must not differ more than ± 0.3V from V_{CC}. See the section “ADC Noise Canceling Techniques” on page 82 on how to connect these pins.

An external reference voltage must be applied to the AREF pin. This voltage must be in the range AGND - AV_{CC}.

Figure 45. Analog-to-Digital Converter Block Schematic



ADC DC Characteristics

TA = -40°C to 85°C

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Resolution			10		Bits
	Absolute accuracy	VREF = 4V, V_{CC} = 4V ADC clock = 200 kHz		1	2	LSB
	Absolute accuracy	VREF = 4V, V_{CC} = 4V ADC clock = 1 MHz		4		LSB
	Absolute accuracy	VREF = 4V, V_{CC} = 4V ADC clock = 2 MHz		16		LSB
	Integral Non-linearity	VREF > 2V		0.5		LSB
	Differential Non-linearity	VREF > 2V		0.5		LSB
	Zero Error (Offset)			1		LSB
	Conversion Time		70		280	µs
	Clock Frequency		50		200	kHz
AV_{CC}	Analog Supply Voltage		V_{CC} - 0.3(1)		V_{CC} + 0.3(2)	V
V_{REF}	Reference Voltage		2		AV_{CC}	V
R_{REF}	Reference Input Resistance		6	10	13	kΩ
R_{AIN}	Analog Input Resistance			100		MΩ

Notes: 1. Minimum for AV_{CC} is 2.7V.
2. Maximum for AV_{CC} is 6.0V.

Interface to External SRAM

The interface to the SRAM consists of:

Port A: multiplexed low-order address bus and data bus

Port C: high-order address bus

The ALE pin: address latch enable

The $\overline{RD}$ and $\overline{WR}$ pin: read and write strobes

The external data SRAM is enabled by setting the external SRAM enable bit (SRE) of the MCU Control Register (MCUCR) and will override the setting of the Data Direction Register (DDRA). When the SRE bit is cleared (zero), the external data SRAM is disabled and the normal pin and data direction settings are used. When SRE is cleared (zero), the address space above the internal SRAM boundary is not mapped into the internal SRAM as AVR parts do not have an interface to the external SRAM.

When ALE goes from high to low, there is a valid address on Port A. ALE is low during a data transfer. $\overline{RD}$ and $\overline{WR}$ are active when accessing the external SRAM only.

When the external SRAM is enabled, the ALE signal may have short pulses when accessing the internal RAM, but the ALE signal is stable when accessing the external SRAM.

Figure 50 shows how to connect an external SRAM to the AVR using eight latches that are transparent when G is high.

By default, the external SRAM access is a three-cycle scheme as depicted in Figure 51. When one extra wait state is needed in the access cycle, set the SRW bit (one) in the MCUCR Register. The resulting access scheme is shown in Figure 52. In both cases, note that Port A is data bus in one cycle only. As soon as the data access finishes, Port A becomes a low-order address bus again.

Note: If a read is followed by a write, or vice versa, there is no extra insertion of wait states in between. The user may insert a NOP between consecutive read and write operations to the external RAM, because such short time for releasing the bus is difficult to obtain without making bus contention.

For details on the timing for the SRAM interface, please refer to Figure 79, Table 45, Table 46, Table 47, and Table 48 in the section “DC Characteristics” on page 118 and refer to “Architectural Overview” on page 8 for a description of the memory map, including address space for SRAM.

Figure 50. External SRAM Connected to the AVR

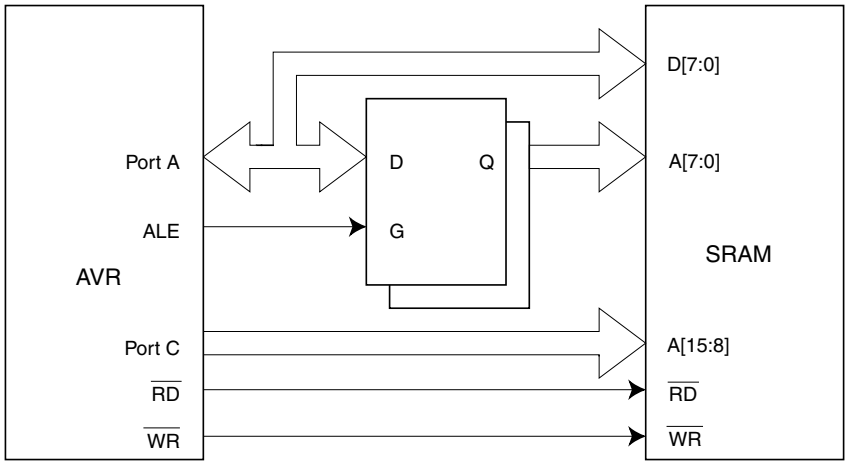


Figure 59. Port B Schematic Diagram (Pins PB5 and PB6)

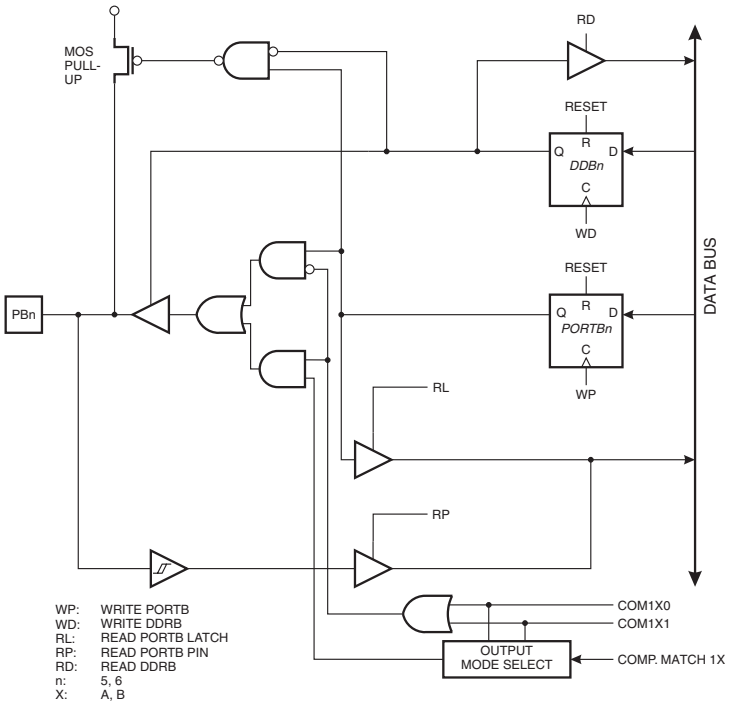
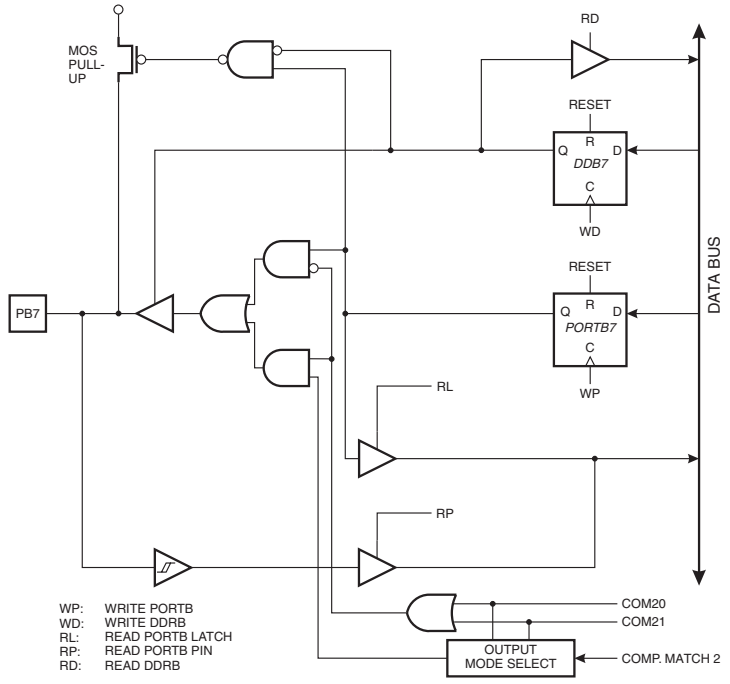


Figure 60. Port B Schematic Diagram (Pin PB7)



- **IC1 – Port D, Bit 4**

IC1, Input Capture pin for Timer/Counter1. When a positive or negative (selectable) edge is applied to this pin, the contents of Timer/Counter1 is transferred to the Timer/Counter1 Input Capture Register. The pin has to be configured as an input to serve this function. See the Timer/Counter1 description on how to operate this function. The internal pull-up MOS resistor can be activated as described above.

- **T1 – Port D, Bit 6**

T1, Timer/Counter1 counter source. See the Timer description for further details.

- **T2 – Port D, Bit 7**

T2, Timer/Counter2 counter source. See the Timer description for further details.

Port D Schematics

Note that all port pins are synchronized. The synchronization latches are, however, not shown in the figures.

Figure 62. Port D Schematic Diagram (Pins PD0, PD1, PD2 and PD3)

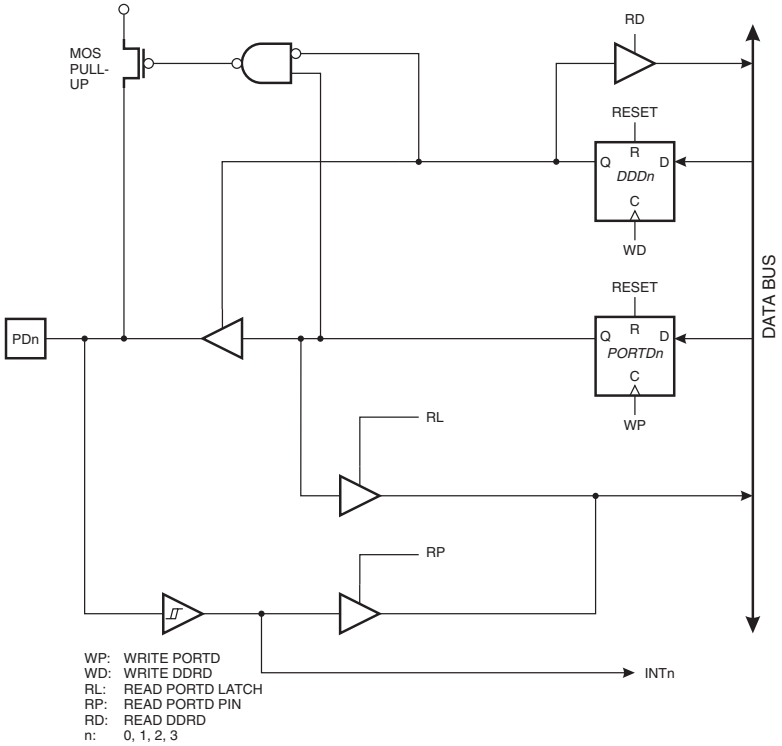
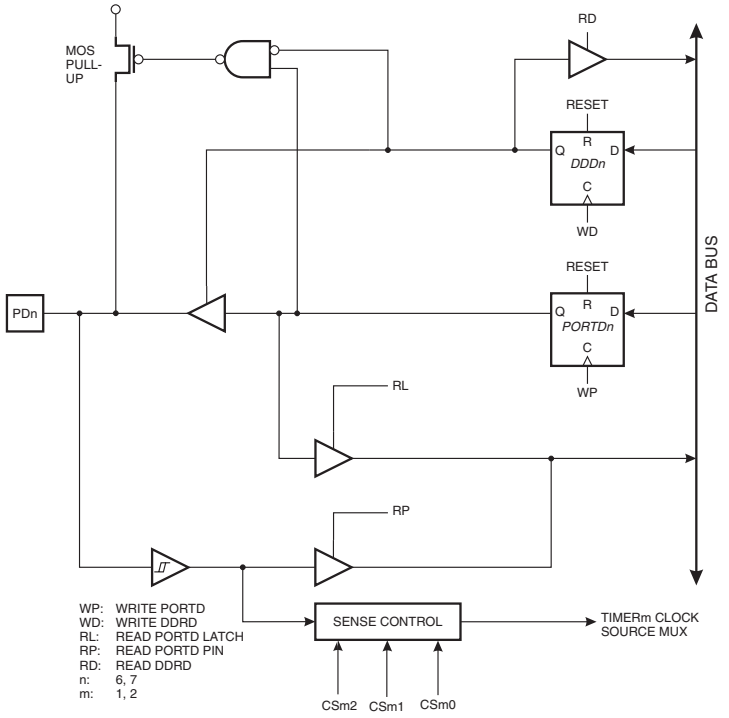


Figure 65. Port D Schematic Diagram (Pins PD6 and PD7)



Port E

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors.

Three I/O memory address locations are allocated for the Port E, one each for the Data Register – PORTE, $03($23), Data Direction Register – DDRE, $02($22) and the Port E Input Pins – PINE, $01($21). The Port E Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

The Port E output buffers can sink 20 mA. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated.

All Port E pins have alternate functions as shown in Table 33.

Table 33. Port E Pin Alternate Functions

Port Pin	Alternate Function
PE0	PDI/RXD (Programming Data Input or UART Receive Pin)
PE1	PDO/TXD (Programming Data Output or UART Transmit Pin)
PE2	AC+ (Analog Comparator Positive Input)
PE3	AC- (Analog Comparator Negative Input)
PE4	INT4 (External Interrupt4 Input)
PE5	INT5 (External Interrupt5 Input)
PE6	INT6 (External Interrupt6 Input)
PE7	INT7 (External Interrupt7 Input)

When the pins are used for the alternate function, the DDRE and PORTE Registers have to be set according to the alternate function description.

Memory Programming

Program and Data Memory Lock Bits

The ATmega103(L) MCU provides two Lock bits that can be left unprogrammed (“1”) or can be programmed (“0”) to obtain the additional features listed in Table 35. The Lock bits can only be erased to “1” with the Chip Erase command.

Table 35. Lock Bit Protection Modes

Memory Lock Bits			Protection Type
Mode	LB1	LB2	
1	1	1	No memory lock features enabled.
2	0	1	Further programming of the Flash and EEPROM is disabled.(1)
3	0	0	Same as mode 2, and verify is also disabled.

Note: 1. In Parallel mode, programming of the Fuse bits are also disabled. Program the Fuse bits before programming the Lock bits.

Fuse Bits

The ATmega103(L) has four Fuse bits, SPIEN, SUT1..0 and EESAVE.

- When the SPIEN Fuse is programmed (“0”), Serial Program and Data Downloading is enabled. Default value is programmed (“0”). The SPIEN Fuse is not accessible in Serial Programming mode.
- When EESAVE is programmed, the EEPROM memory is preserved through the Chip Erase cycle. Default value is unprogrammed (“1”). The EESAVE Fuse bit cannot be programmed if any of the Lock bits are programmed.
- SUT1..0 Fuses: Determine the MCU start-up time. See Table 5 on page 27 for further details. Default value is unprogrammed (“11”), which gives a nominal start-up time of 16 ms.

The status of the Fuse bits is not affected by Chip Erase.

Signature Bytes

All Atmel microcontrollers have a 3-byte signature code that identifies the device. This code can be read in both Serial and Parallel mode. The three bytes reside in a separate address space.

For the ATmega103 they are:

1. $000: $1E (indicates manufactured by Atmel)
2. $001: $97 (indicates 128K bytes Flash memory)
3. $002: $01 (indicates ATmega103 when signature byte $001 is $97)

Programming the Flash and EEPROM

Atmel’s ATmega103(L) offers 128K bytes of In-System Reprogrammable Flash memory and 4K bytes of EEPROM Data memory.

The ATmega103(L) is shipped with the On-chip Flash Program and EEPROM Data memory arrays in the erased state (i.e., contents = $FF) and ready to be programmed. This device supports a Parallel Programming mode and a Serial Programming mode. The +12V supplied to the $\overline{\text{RESET}}$ pin in Parallel Programming mode is used for programming enable only, and no current of significance is drawn by this pin. The Serial Programming mode provides a convenient way to download program and data into the ATmega103(L) inside the user’s system.

E: Load Data High Byte.

1. Set BS1 to “1”. This selects high data.
2. Set XA1, XA0 to “01”. This enables data loading.
3. Set DATA = Data High Byte ($00 - $FF).
4. Give XTAL1 a positive pulse. This loads the data High Byte.

F: Latch Data High Byte.

1. Give PAGEL a positive pulse. This latches the data High Byte.

G: Repeat B through F 128 times to fill the page buffer.

H: Load Address High Byte.

1. Set XA1, XA0 to “00”. This enables address loading.
2. Set BS1 to “1”. This selects high address.
3. Set DATA = Address High Byte ($00 - $FF).
4. Give XTAL1 a positive pulse. This loads the address High Byte.

I: 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.

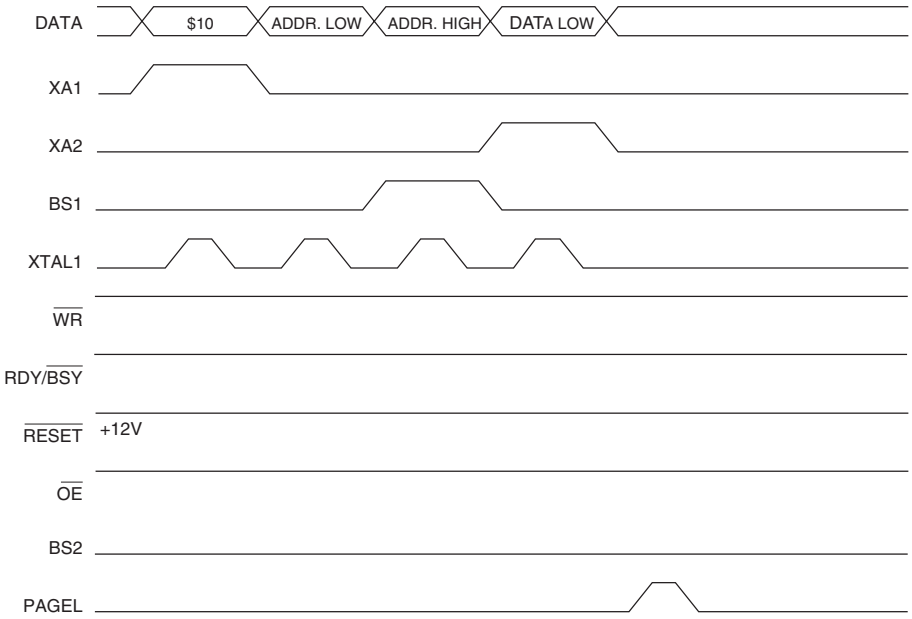
(See Figure 74 for signal waveforms.)

J: End Page Programming.

1. Set XA1, XA0 to “10”. This enables command loading.
2. Set DATA = “0000 0000”. This is the command for No Operation.
3. Give XTAL1 a positive pulse. This loads the command and the internal write signals are reset.

K: Repeat A through J 512 times or until all data has been programmed.

Figure 73. Programming the Flash Waveforms



Serial Downloading

Both the Flash and EEPROM memory arrays can be programmed using the serial interface while $\overline{\text{RESET}}$ is pulled to GND, or when $\overline{\text{PEN}}$ is low during Power-on Reset. The serial interface consists of pins SCK, RXD/PDI (input) and TXD/PDO (output). After $\overline{\text{RESET}}$ is set low, the Programming Enable instruction needs to be executed first before program/erase instructions can be executed.

For the EEPROM, an auto-erase cycle is provided within the self-timed Write instruction and there is no need to first execute the Chip Erase instruction. The Chip Erase instruction turns the content of every memory location in both the Program and EEPROM arrays into $FF.

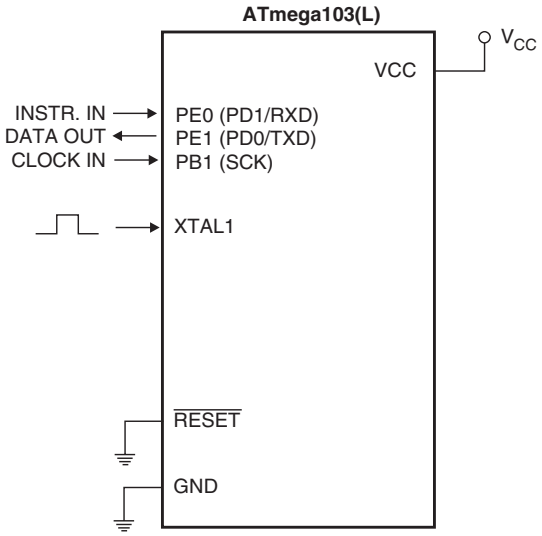
The Program and EEPROM memory arrays have separate address spaces: $0000 to $FFFF for Program memory and $0000 to $0FFF for EEPROM memory.

Either an external clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The minimum low and high periods for the serial clock (SCK) input are defined as follows:

Low: > 2 XTAL1 clock cycles

High: > 2 XTAL1 clock cycles

Figure 77. Serial Programming



Note: Instruction in and data out is not using the SPI pins as on other AVR devices. SCK uses the SPI pin as usual.

Serial Programming Algorithm

When writing serial data to the ATmega103(L), data is sampled by the ATmega 103/103L on the rising edge of SCK. When reading data from the ATmega103(L), data is clocked on the falling edge of SCK. See Figure 78 for an explanation. To program and verify the ATmega103(L) in the Serial Programming mode, the following sequence is recommended (See 4-byte instruction formats in Table 44.):

1. Power-up sequence: Apply power between V_{CC} and GND while $\overline{\text{RESET}}$ and SCK are set to “0”. The $\overline{\text{RESET}}$ signal must be kept low during the complete serial programming session. If a crystal is not connected across pins XTAL1 and XTAL2, apply a clock signal to the XTAL1 pin. In some systems, the programmer cannot guarantee that SCK is held low during Power-up. In this case, $\overline{\text{RESET}}$ must be given a positive pulse of at least two XTAL1 cycles duration after SCK has been set to “0”.

As an alternative to using the $\overline{\text{RESET}}$ signal, $\overline{\text{PEN}}$ can be held low during Power-on Reset while SCK is set to “0”. In this case, only the $\overline{\text{PEN}}$ value at Power-on Reset is important. If a crystal is not connected across pins XTAL1 and XTAL2, apply a clock signal to the XTAL1 pin. If the programmer cannot guarantee that SCK is held low during power-up, the $\overline{\text{PEN}}$ method cannot be used. The device must be powered down in order to commence normal operation when using this method.

2. Wait for at least 20 ms and enable serial programming by sending the Programming Enable serial instruction to pin PE0(PDI/RXD).
3. The serial programming instructions will not work if the communication is out of synchronization. When in sync, the second byte ($53) will echo back when issuing the third byte of the Programming Enable instruction. Whether the echo is correct or not, all four bytes of the instruction must be transmitted. If the $53 did not echo back, give SCK a positive pulse and issue a new Programming Enable instruction. If the $53 is not seen within 32 attempts, there is no functional device connected.
4. If a chip erase is performed (must be done to erase the Flash), wait at least (2 x t_{WD_FLASH}), give $\overline{\text{RESET}}$ a positive pulse of at least two XTAL1 cycles duration after SCK has been set to “0”, and start over from step 2.
5. The Flash is programmed one page at a time. The memory page is loaded one byte at a time by supplying the 7 LSB of the address and data together with the Load Program Memory Page instruction. The Program Memory Page is stored by loading the Write Program Memory Page instruction with the 9 MSB of the address. The next page can be written after t_{WD_FLASH}, i.e., writing 256 bytes takes t_{WD_FLASH}. Accessing the serial programming interface before the Flash write operation completes can result in incorrect programming.
6. The EEPROM array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. An EEPROM memory location is first automatically erased before new data is written. If polling is not used, the user must wait at least t_{WD_EEPROM} before issuing the next byte. (Please refer to Table 42.)
7. Any memory location can be verified by using the Read instruction, which returns the content at the selected address at serial output PE1(PDO/TXD).
8. At the end of the programming session, $\overline{\text{RESET}}$ can be set high to commence normal operation.
9. Power-off sequence (if needed): Set XTAL1 to “0” (if a crystal is not used). Set $\overline{\text{RESET}}$ to “1”. Turn V_{CC} power off.

Table 42 shows the actual delays used in this section.

Please note: The MISO pin is not high-Z during serial programming.

Data Polling for the EEPROM

When a new EEPROM byte has been written and is being programmed into the EEPROM, reading the address location being programmed will first give the value P1 (please refer to Table 43) until the auto-erase is finished, and then the value P2.

At the time the device is ready for a new EEPROM byte, the programmed value will read correctly. This is used to determine when the next byte can be written. This will not work for the values P1 and P2, so when programming these values, the user will have to wait for at least the prescribed time t_{WD_EEPROM} (please refer to Table 42) before programming the next byte. As a chip-erased device contains $FF in all locations, programming of addresses that are meant to contain $FF can be skipped. This does not apply if the EEPROM is reprogrammed without chip erasing the device.

Figure 90. Analog Comparator Offset Voltage vs. Common Mode Voltage

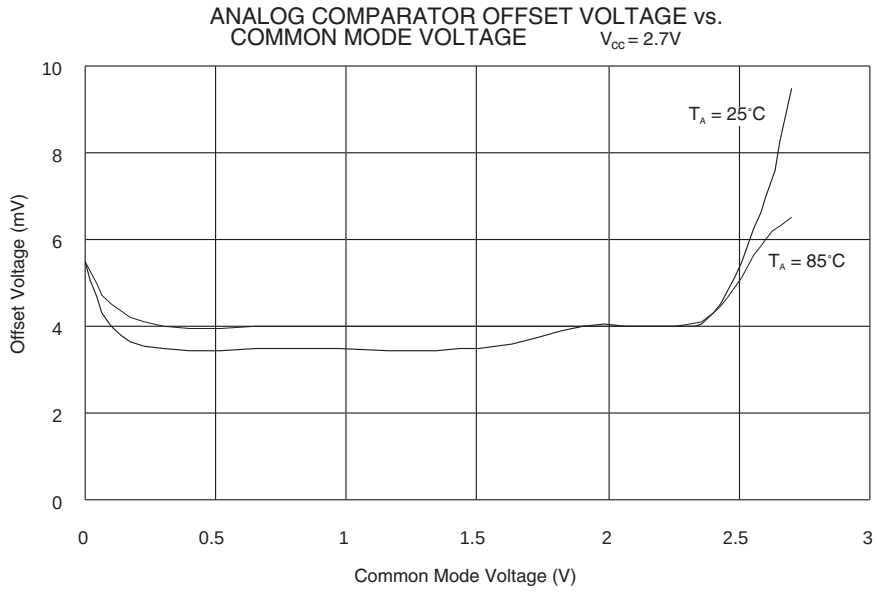


Figure 91. Analog Comparator Input Leakage Current