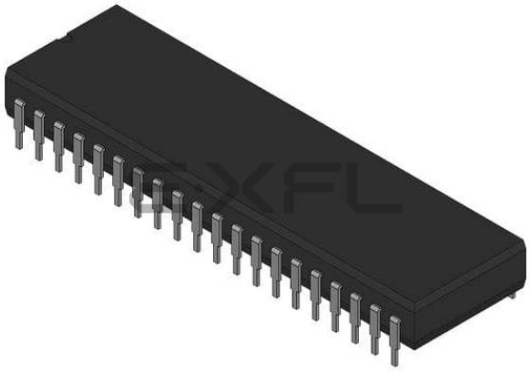


Welcome to E-XFL.COM

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	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/atmel/atmega323l-4pc

Clock Options

The device has the following clock source options, selectable by Flash Fuse bits as shown:

Table 1. Device Clocking Options Select(1)

Device Clocking Option	CKSEL3..0
External Crystal/Ceramic Resonator	1111 - 1010
External Low-frequency Crystal	1001 - 1000
External RC Oscillator	0111 - 0101
Internal RC Oscillator	0100 - 0010
External Clock	0001 - 0000

Note: 1. “1” means unprogrammed, “0” means programmed.

The various choices for each clocking option give different start-up times as shown in Table 6 on page 27.

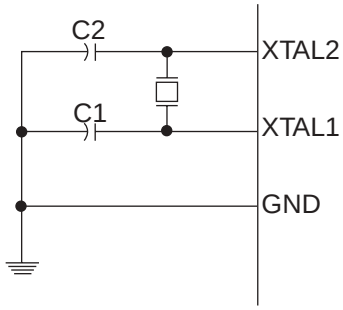
Internal RC Oscillator

The internal RC Oscillator option is an On-chip Oscillator running at a fixed frequency of nominally 1 MHz. If selected, the device can operate with no external components. See “Calibrated Internal RC Oscillator” on page 41 for information on calibrating this Oscillator.

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 2. Either a quartz crystal or a ceramic resonator may be used.

Figure 2. Oscillator Connections



- **Bit 6 – TOIE2: Timer/Counter2 Overflow Interrupt Enable**

When the TOIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter2 occurs, i.e., when the TOV2 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 5 – TICIE1: Timer/Counter1 Input Capture Interrupt Enable**

When the TICIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Input Capture Event Interrupt is enabled. The corresponding interrupt is executed if a capture triggering event occurs on PD6 (ICP), i.e., when the ICF1 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 4 – OCIE1A: Timer/Counter1 Output Compare A Match Interrupt Enable**

When the OCIE1A bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Compare A Match interrupt is enabled. The corresponding interrupt is executed if a Compare A Match in Timer/Counter1 occurs, i.e., when the OCF1A bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 3 – OCIE1B: Timer/Counter1 Output Compare B Match Interrupt Enable**

When the OCIE1B bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Compare B Match interrupt is enabled. The corresponding interrupt is executed if a Compare B Match in Timer/Counter1 occurs, i.e., when the OCF1B bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 2 – TOIE1: Timer/Counter1 Overflow Interrupt Enable**

When the TOIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter1 occurs, i.e., when the TOV1 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 1 – OCIE0: Timer/Counter0 Output Compare Match Interrupt Enable**

When the OCIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match interrupt is enabled. The corresponding interrupt is executed if a Compare0 Match in Timer/Counter0 occurs, i.e., when the OCF0 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, i.e. when the TOV0 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

Power-down Mode

When the SM2..0 bits are 010, the SLEEP instruction makes the MCU enter Power-down mode. In this mode, the external Oscillator is stopped, while the external interrupts, the Two-wire Serial Interface address watch, and the Watchdog continue operating (if enabled). Only an External Reset, a Watchdog Reset, an Two-wire Serial Interface address match interrupt, an external level interrupt on INT0 or INT1, or an external edge interrupt on INT2 can wake up the MCU.

Note that if a level triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. This makes the MCU less sensitive to noise. The changed level is sampled twice by the Watchdog Oscillator clock, and if the input has the required level during this time, the MCU will wake up. The period of the Watchdog Oscillator is 1 µs (nominal) at 5.0V and 25°C. The frequency of the Watchdog Oscillator is voltage dependent as shown in the Electrical Characteristics section.

When waking up from Power-down mode, there is a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined by the same CKSEL Fuses that define the Reset Time-out Period, as seen in Table 6 on page 27.

Power-save Mode

When the SM2..0 bits are 011, the SLEEP instruction forces the MCU into the Power-save mode. This mode is identical to Power-down, with one exception:

If Timer/Counter2 is clocked asynchronously, i.e., the AS2 bit in ASSR is set, Timer/Counter2 will run during sleep. The device can wake up from either Timer Overflow or Output Compare event from Timer/Counter2 if the corresponding Timer/Counter2 interrupt enable bits are set in TIMSK, and the Global Interrupt Enable bit in SREG is set.

If the asynchronous timer is NOT clocked asynchronously, Power-down mode is recommended instead of Power-save mode because the contents of the registers in the asynchronous timer should be considered undefined after wake-up in Power-save mode if AS2 is 0.

Standby Mode

When the SM2..0 bits are 110 and an external crystal/resonator clock option is selected, the SLEEP instruction forces the MCU into the Standby mode. This mode is identical to Power-down with the exception that the Oscillator is kept running. From Standby mode, the device wakes up in only six clock cycles.

Extended Standby Mode

When the SM2..0 bits are 111 and an external crystal/resonator clock option is selected, the SLEEP instruction forces the MCU into the Extended Standby mode. This mode is identical to Power-save mode with the exception that the Oscillator is kept running. From Extended Standby mode, the device wakes up in only six clock cycles.

Timer/Counter0 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 49 for a detailed description on this function.

Timer/Counter0 Control Register – TCCR0

Bit	7	6	5	4	3	2	1	0	
$33 ($53)	FOC0	PWM0	COM01	COM00	CTC0	CS02	CS01	CS00	TCCR0
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	

Timer/Counter2 Control Register – TCCR2

Bit	7	6	5	4	3	2	1	0	
$25 ($45)	FOC2	PWM2	COM21	COM20	CTC2	CS22	CS21	CS20	TCCR2
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 – FOC0/FOC2: Force Output Compare**

Writing a logical one to this bit, forces a change in the Compare Match output pin PB3 (Timer/Counter0) and PD7 (Timer/Counter2) according to the values already set in COMn1 and COMn0. If the COMn1 and COMn0 bits are written in the same cycle as FOC0/FOC2, the new settings will not take effect until next Compare Match or Forced Output Compare Match occurs. The Force Output Compare bit can be used to change the output pin without waiting for a Compare Match in the timer. The automatic action programmed in COMn1 and COMn0 happens as if a Compare Match had occurred, but no interrupt is generated and the Timer/Counters will not be cleared even if CTC0/CTC2 is set. The corresponding I/O pin must be set as an output pin for the FOC0/FOC2 bit to have effect on the pin. The FOC0/FOC2 bits will always be read as zero. Setting the FOC0/FOC2 bits has no effect in PWM mode.

- **Bit 6 – PWM0/PWM2: Pulse Width Modulator Enable**

When set (one) this bit enables PWM mode for Timer/Counter0 or Timer/Counter2. This mode is described on page 49.

- **Bits 5, 4 – COM01, COM00/COM21, COM20: Compare Output Mode, Bits 1 and 0**

The COMn1 and COMn0 control bits determine any output pin action following a compare match in Timer/Counter0 or Timer/Counter2. Output pin actions affect pins PB3(OC0) or PD7(OC2). This is an alternative function to an I/O port, and the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 12.

Table 12. Compare Mode Select[1]

COMn1[2]	COMn0	Description
0	0	Timer/Counter Disconnected from Output Pin OCn
0	1	Toggle the OCn Output Line.
1	0	Clear the OCn Output Line (to Zero).
1	1	Set the OCn Output Line (to One).

Notes: 1. In PWM mode, these bits have a different function. Refer to Table 15 for a description.
2. n = 0 or 2

Timer/Counter1 Control Register A – TCCR1A

Bit	7	6	5	4	3	2	1	0	
$2F ($4F)	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	PWM11	PWM10	TCCR1A
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 – COM1A1, COM1A0: Compare Output Mode1A, Bits 1 and 0**

The COM1A1 and COM1A0 control bits determine any output pin action following a Compare Match in Timer/Counter1. Any output pin actions affect pin OC1A – Output Compare A. This is an alternative function to an I/O port, and the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 10.

- **Bits 5, 4 – COM1B1, COM1B0: Compare Output Mode1B, Bits 1 and 0**

The COM1B1 and COM1B0 control bits determine any output pin action following a Compare Match in Timer/Counter1. Any output pin actions affect pin OC1B – Output Compare B. This is an alternative function to an I/O port, and the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 10.

Table 17. Compare 1 Mode Select[1]

COM1X1	COM1X0	Description
0	0	Timer/Counter1 Disconnected from Output Pin OC1X
0	1	Toggle the OC1X Output Line.
1	0	Clear the OC1X Output Line (to Zero).
1	1	Set the OC1X Output Line (to One).

Note: 1. X = A or B.

In PWM mode, these bits have a different function. Refer to Table 22 for a description.

- **Bit 3 – FOC1A: Force Output Compare 1A**

Writing a logical one to this bit, forces a change in the Compare Match output pin PD5 according to the values already set in COM1A1 and COM1A0. If the COM1A1 and COM1A0 bits are written in the same cycle as FOC1A, the new settings will not take effect until next Compare Match or Forced Compare Match occurs. The Force Output Compare bit can be used to change the output pin without waiting for a Compare Match in the timer. The automatic action programmed in COM1A1 and COM1A0 happens as if a Compare Match had occurred, but no interrupt is generated and it will not clear the timer even if CTC1 in TCCR1B is set. The corresponding I/O pin must be set as an output pin for the FOC1A bit to have effect on the pin. The FOC1A bit will always be read as zero. The setting of the FOC1A bit has no effect in PWM mode.

- **Bit 2 – FOC1B: Force Output Compare 1B**

Writing a logical one to this bit, forces a change in the Compare Match output pin PD4 according to the values already set in COM1B1 and COM1B0. If the COM1B1 and COM1B0 bits are written in the same cycle as FOC1B, the new settings will not take effect until next Compare Match or Forced Compare Match occurs. The Force Output Compare bit can be used to change the output pin without waiting for a Compare Match in the timer. The automatic action programmed in COM1B1 and COM1B0 happens as if

The TEMP Register is also used when accessing TCNT1 and ICR1. If the main program and also interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program and interrupt routines.

Timer/Counter1 Input Capture Register – ICR1H and ICR1L

Bit	15	14	13	12	11	10	9	8	
$27 ($47)	MSB								ICR1H
$26 ($46)								LSB	ICR1L
	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	

The Input Capture Register is a 16-bit read-only register.

When the rising or falling edge (according to the Input Capture Edge setting – ICES1) of the signal at the Input Capture Pin – ICP – is detected, the current value of the Timer/Counter1 Register – TCNT1 – is transferred to the Input Capture Register – ICR1. At the same time, the Input Capture Flag – ICF1 – is set (one).

Since the Input Capture Register – ICR1 – is a 16-bit register, a temporary register TEMP is used when ICR1 is read to ensure that both bytes are read simultaneously. When the CPU reads the Low Byte ICR1L, the data is sent to the CPU and the data of the High Byte ICR1H is placed in the TEMP Register. When the CPU reads the data in the High Byte ICR1H, the CPU receives the data in the TEMP Register. Consequently, the Low Byte ICR1L must be accessed first for a full 16-bit register read operation.

The TEMP Register is also used when accessing TCNT1, OCR1A, and OCR1B. If the main program and also interrupt routines accesses registers using TEMP, interrupts must be disabled during access from the main program and interrupt routines.

Timer/Counter1 In PWM Mode

When the PWM mode is selected, Timer/Counter1 and the Output Compare Register1A – OCR1A and the Output Compare Register1B – OCR1B, form a dual 8-, 9-, or 10-bit, Free Running, Glitch-free, and phase correct PWM with outputs on the PD5(OC1A) and PD4(OC1B) pins. In this mode, the Timer/Counter1 acts as an up/down counter, counting up from $0000 to TOP (see Table 21), where it turns and counts down again to zero before the cycle is repeated. When the counter value matches the contents of the 8, 9, or 10 least significant bits (depending on resolution) of OCR1A or OCR1B, the PD5(OC1A)/PD4(OC1B) pins are set or cleared according to the settings of the COM1A1/COM1A0 or COM1B1/COM1B0 bits in the Timer/Counter1 Control Register TCCR1A. Refer to Table 17 on page 56 for details.

Alternatively, the Timer/Counter1 can be configured to a PWM that operates at twice the speed as in the mode described above. Then the Timer/Counter1 and the Output Compare Register1A – OCR1A and the Output Compare Register1B – OCR1B, form a dual 8-, 9-, or 10-bit, free running and glitch-free PWM with outputs on the PD5(OC1A) and PD4(OC1B) pins..

Table 36. Examples of UBRR Settings for Commonly Used Oscillator Frequencies – **UBRR = 0, Error = 0.0%** (Continued)

14.4K	8	-3.5%	16	2.1%	15	0.0%	31	0.0%
19.2K	6	-7.0%	12	0.2%	11	0.0%	23	0.0%
28.8K	3	8.5%	8	-3.5%	7	0.0%	15	0.0%
38.4K	2	8.5%	6	-7.0%	5	0.0%	11	0.0%
57.6K	1	8.5%	3	8.5%	3	0.0%	7	0.0%
76.8K	1	-18.6%	2	8.5%	2	0.0%	5	0.0%
115.2K	0	8.5%	1	8.5%	1	0.0%	3	0.0%
230.4K	–	–	–	–	0	0.0%	1	0.0%
250K	–	–	0	0.0%	0	-7.8%	1	-7.8%
1M					–	–	0	-7.8%
Max	**125 Kbps**		**250 Kbps**		**230.4 Kbps**		**460.8 Kbps**	

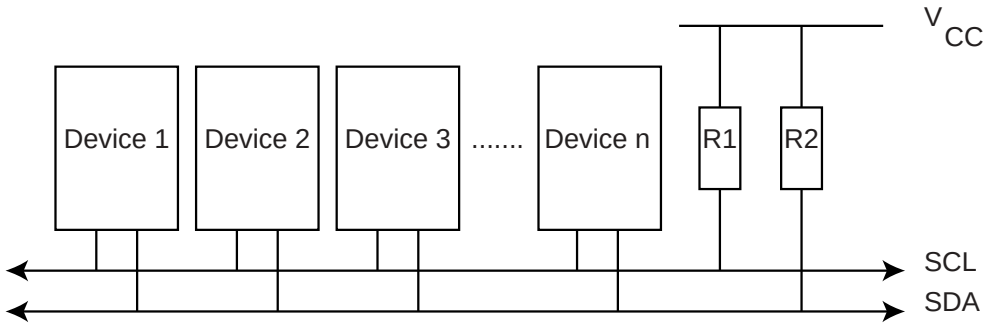
Baud Rate (bps)	**f_{osc} = 4.0000 MHz**				**f_{osc} = 7.3728 MHz**			
	U2X = 0		**U2X = 1**		**U2X = 0**		**U2X = 1**	
	UBRR	**Error**	**UBRR**	**Error**	**UBRR**	**Error**	**UBRR**	**Error**
2400	103	0.2%	207	0.2%	191	0.0%	383	0.0%
4800	51	0.2%	103	0.2%	95	0.0%	191	0.0%
9600	25	0.2%	51	0.2%	47	0.0%	95	0.0%
14.4K	16	2.1%	34	-0.8%	31	0.0%	63	0.0%
19.2K	12	0.2%	25	0.2%	23	0.0%	47	0.0%
28.8K	8	-3.5%	16	2.1%	15	0.0%	31	0.0%
38.4K	6	-7.0%	12	0.2%	11	0.0%	23	0.0%
57.6K	3	8.5%	8	-3.5%	7	0.0%	15	0.0%
76.8K	2	8.5%	6	-7.0%	5	0.0%	11	0.0%
115.2K	1	8.5%	3	8.5%	3	0.0%	7	0.0%
230.4K	0	8.5%	1	8.5%	1	0.0%	3	0.0%
250K	0	0.0%	1	0.0%	1	-7.8%	3	-7.8%
0.5M	–	–	0	0.0%	0	-7.8%	1	-7.8%
1M	–	–	–	–	–	–	0	-7.8%
Max	**250 Kbps**		**0.5 Mbps**		**460.8 Kpbs**		**921.6 Kbps**	

Baud Rate (bps)	**f_{osc} = 8.0000 MHz**			
	U2X = 0		**U2X = 1**	
	UBRR	**Error**	**UBRR**	**Error**
2400	207	0.2%	416	-0.1%
4800	103	0.2%	207	0.2%
9600	51	0.2%	103	0.2%
14.4K	34	-0.8%	68	0.6%

Two-wire Serial Interface (Byte Oriented)

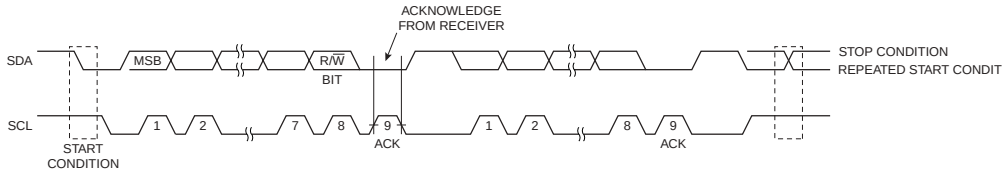
The Two-wire Serial Interface supports bi-directional serial communication. It is designed primarily for simple but efficient integrated circuit (IC) control. The system is comprised of two lines, SCL (Serial Clock) and SDA (Serial Data) that carry information between the ICs connected to them. Various communication configurations can be designed using this bus. Figure 52 shows a typical Two-wire Serial Bus configuration. Any device connected to the bus can be Master or Slave. Note that all AVR devices connected to the bus must be powered to allow any bus operation.

Figure 52. Two-wire Serial Bus Configuration



The Two-wire Serial Interface supports Master/Slave and Transmitter/Receiver operation at up to 400 kHz bus clock rate. The Two-wire Serial Interface has hardware support for 7-bit addressing. When the Two-wire Serial Interface is enabled (TWEN in TWCR is set), a glitch filter is enabled for the input signals from the pins PC0 (SCL) and PC1 (SDA), and the output from these pins is slew-rate controlled. The Two-wire Serial Interface is byte oriented. The operation of the Two-wire Serial Bus is shown as a pulse diagram in Figure 53, including the START and STOP conditions and generation of ACK signal by the bus Receiver.

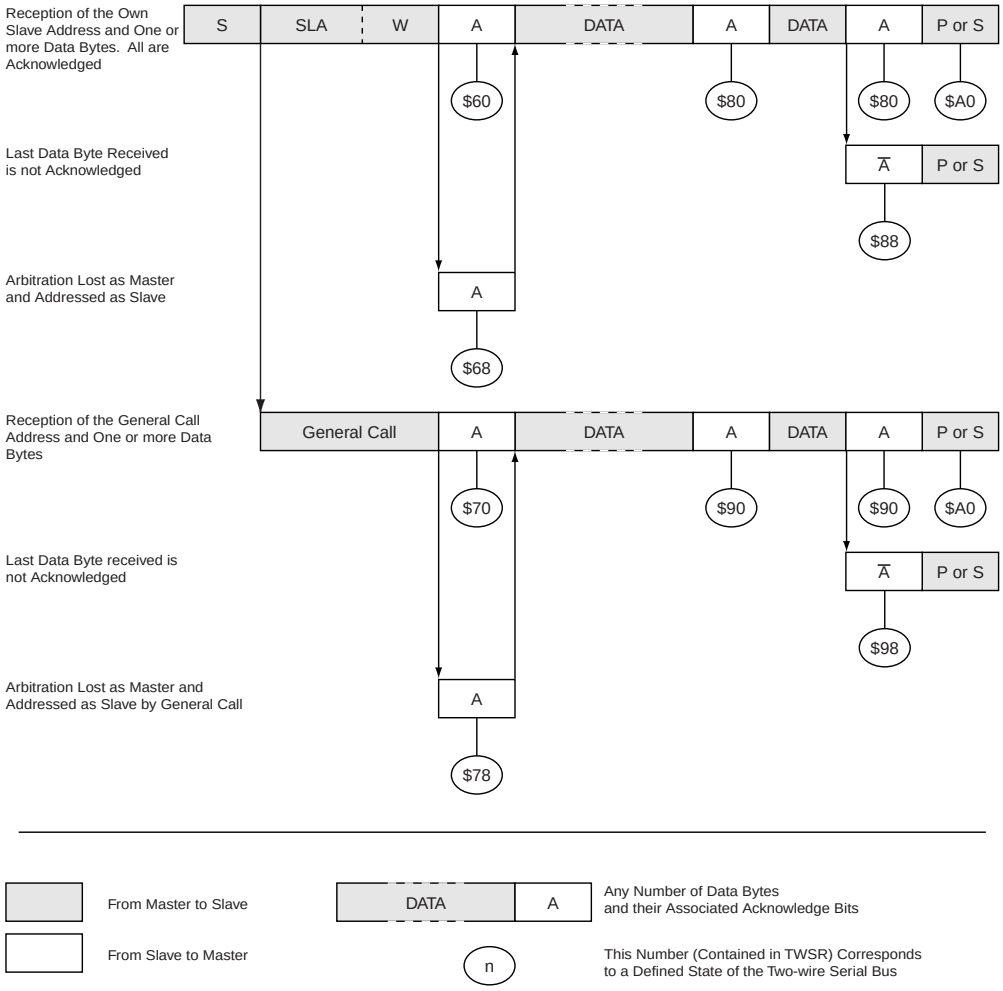
Figure 53. Two-wire Serial Bus Timing Diagram



The block diagram of the Two-wire Serial Interface is shown in Figure 54.

```
      in    r16, TWSR          ; Check value of TWI Status Register. If status
      cpi   r16, MR_SLA_ACK    ; different from MR_SLA_ACK, go to ERROR
      brne  ERROR

      ldi   r16, (1<<TWINT) | (1<<TWEA) | (1<<TWEN)
      out   TWCR, r16          ; Clear TWINT bit in TWCR to start reception of
                               ; data. Setting TWEA causes ACK to be returned
                               ; after reception of data byte
wait7:in    r16,TWCR           ; Wait for TWINT Flag set. This indicates that
      sbrs  r16, TWINT         ; data has been received and ACK returned
      rjmp  wait7

      in    r16, TWSR          ; Check value of TWI Status Register. If status
      cpi   r16, MR_DATA_ACK   ; different from MR_DATA_ACK, go to ERROR
      brne  ERROR

      in    r16, TWDR          ; Input received data from TWDR.
      nop                      ;<do something with received data>
      ldi   r16, (1<<TWINT) | (1<<TWEA) | (1<<TWEN)
      out   TWCR, r16          ; Clear TWINT bit in TWCR to start reception of
                               ; data. Setting TWEA causes ACK to be returned
                               ;after reception of data byte
   ;<Receive more data bytes if needed>



   ;receive next to last data byte.
wait8:in    r16,TWCR           ; Wait for TWINT Flag set. This indicates that
      sbrs  r16, TWINT         ; data has been received and ACK returned
      rjmp  wait8

      in    r16, TWSR          ; Check value of TWI Status Register. If status
      cpi   r16, MR_DATA_ACK   ; different from MR_DATA_ACK, go to ERROR
      brne  ERROR

      in    r16, TWDR          ; Input received data from TWDR.
      nop                      ;<do something with received data>
      ldi   r16, (1<<TWINT) | (1<<TWEN)
      out   TWCR, r16          ; Clear TWINT bit in TWCR to start reception of
                               ; data. Not setting TWEA causes NACK to be
                               ; returned after reception of next data byte
                               ; receive last data byte. Signal this to Slave
                               ; by returning NACK
wait9:in    r16,TWCR           ; Wait for TWINT Flag set. This indicates that
      sbrs  r16, TWINT         ; data has been received and NACK returned
      rjmp  wait9

      in    r16, TWSR          ; Check value of TWI Status Register. If status
      cpi   r16, MR_DATA_NACK  ; different from MR_DATA_NACK, go to ERROR
      brne  ERROR
```

Figure 57. Formats and States in the Slave Receiver Mode



Assembly Code Example – Slave Receiver Mode

```
;Part specific include file and TWI include file must be included.
; <Initialize registers TWAR and TWBR>

      ldi   r16, (1<<TWINT) | (1<<TWEA) | (1<<TWEN)
      out   TWCR, r16          ; Enable TWI in Slave Receiver Mode

; <Receive START condition and SLA+W>

wait10:in   r16,TWCR           ; Wait for TWINT Flag set. This indicates that
      sbrs  r16, TWINT         ; START followed by SLA+W has been received
      rjmp  wait10

      in    r16, TWSR          ; Check value of TWI Status Register. If status
      cpi   r16, SR_SLA_ACK    ; different from SR_SLA_ACK, go to ERROR
      brne  ERROR

      ldi   r16, (1<<TWINT) | (1<<TWEA) | (1<<TWEN)
      out   TWCR, r16          ; Clear TWINT bit in TWCR to start reception of
                               ; first data byte. Setting TWEA indicates that
```

Figure 69. Port B Schematic Diagram (Pin PB2)

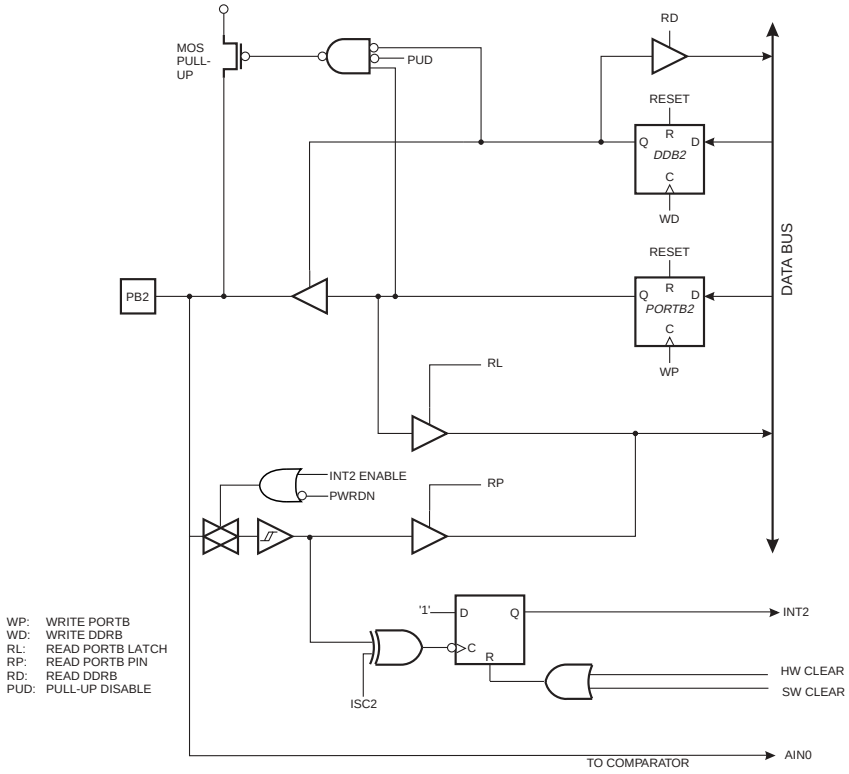
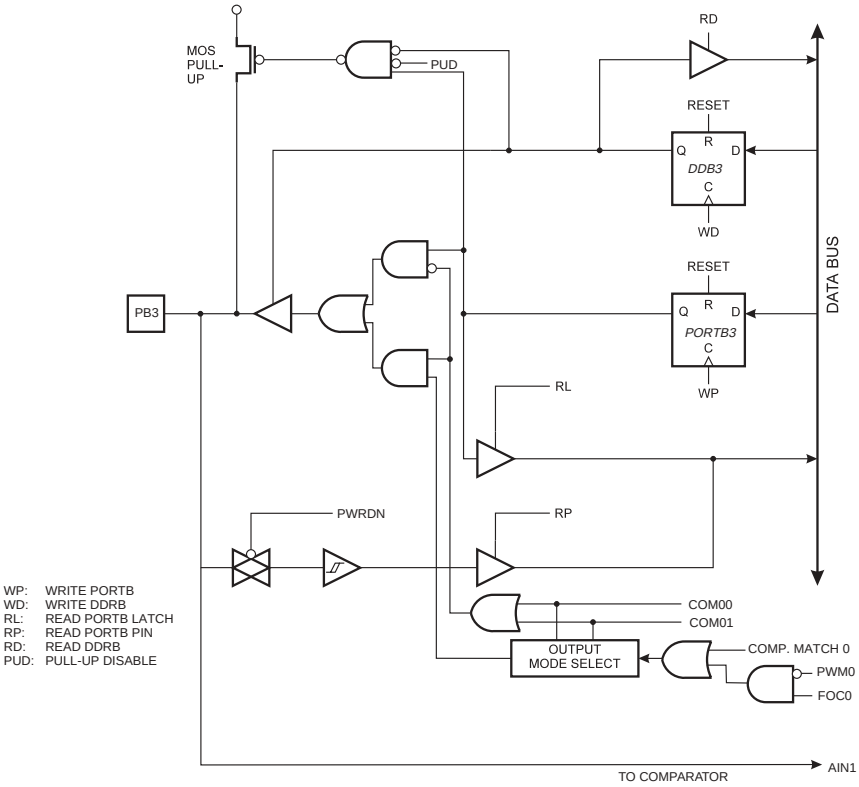


Figure 70. Port B Schematic Diagram (Pin PB3)



- **TCK – Port C, Bit 2**

TCK, JTAG Test Clock: JTAG operation is synchronous to TCK. When the JTAG interface is enabled, this pin can not be used as an I/O pin. Refer to the section “JTAG Interface and the On-chip Debug System” on page 157 for details on operation of the JTAG interface.

- **SDA – Port C, Bit 1**

SDA, Two-wire Serial Interface Data: When the TWEN bit in TWCR is set (one) to enable the Two-wire Serial Interface, pin PC1 is disconnected from the port and becomes the Serial Data I/O pin for the Two-wire Serial Interface. In this mode, there is a spike filter on the pin to capture spikes shorter than 50 ns on the input signal, and the pin is driven by an open collector driver with slew rate limitation.

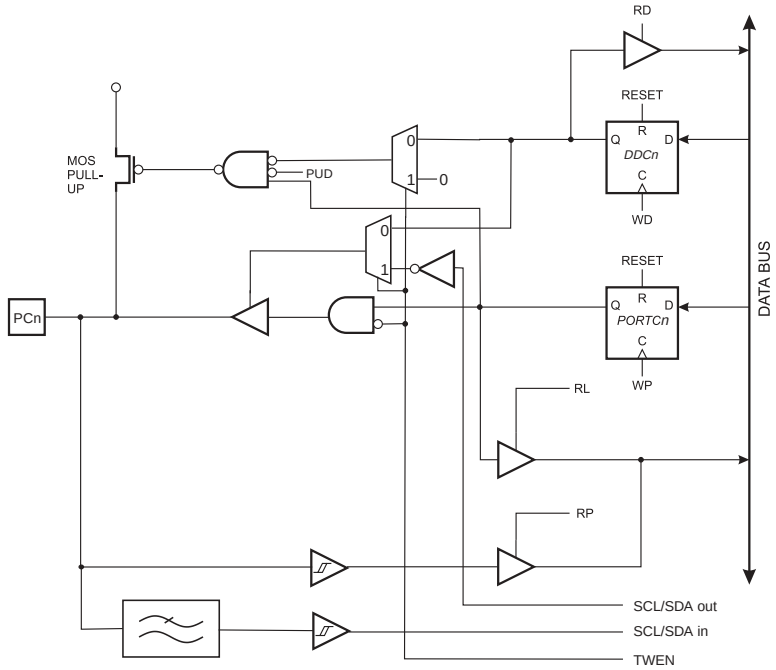
- **SCL – Port C, Bit 0**

SCL, Two-wire Serial Interface Clock: When the TWEN bit in TWCR is set (one) to enable the Two-wire Serial Interface, pin PC1 is disconnected from the port and becomes the Serial Clock I/O pin for the Two-wire Serial Interface. In this mode, there is a spike filter on the pin to capture spikes shorter than 50 ns on the input signal.

Port C Schematics

Note that all port pins are synchronized. The synchronization latches are not shown in the figure.

Figure 75. Port C Schematic Diagram (Pins PC0 - PC1)



WP: WRITE PORTC
WD: WRITE DDRC
RL: READ PORTC LATCH
RP: READ PORTC PIN
RD: READ DDRC
PUD: PULL-UP DISABLE
n = 0, 1

Table 58. ATmega323 Boundary-Scan Order (Continued)

Bit Number	Signal Name	Module
44	PD5.Data	Port D
43	PD5.Control	
42	PD5.PuLLup_Disable	
41	PD6.Data	
40	PD6.Control	
39	PD6.PuLLup_Disable	
38	PD7.Data	
37	PD7.Control	
36	PD7.PuLLup_Disable	
35	PC0.Data	Port C
34	PC0.Control	
33	PC0.PuLLup_Disable	
32	PC1.Data	
31	PC1.Control	
30	PC1.PuLLup_Disable	
29	PC6.Data	
28	PC6.Control	
27	PC6.PuLLup_Disable	
26	PC7.Data	
25	PC7.Control	
24	PC7.PuLLup_Disable	
23	PA7.Data	Port A
22	PA7.Control	
21	PA7.PuLLup_Disable	
20	PA6.Data	
19	PA6.Control	
18	PA6.PuLLup_Disable	
17	PA5.Data	
16	PA5.Control	
15	PA5.PuLLup_Disable	
14	PA4.Data	
13	PA4.Control	
12	PA4.PuLLup_Disable	

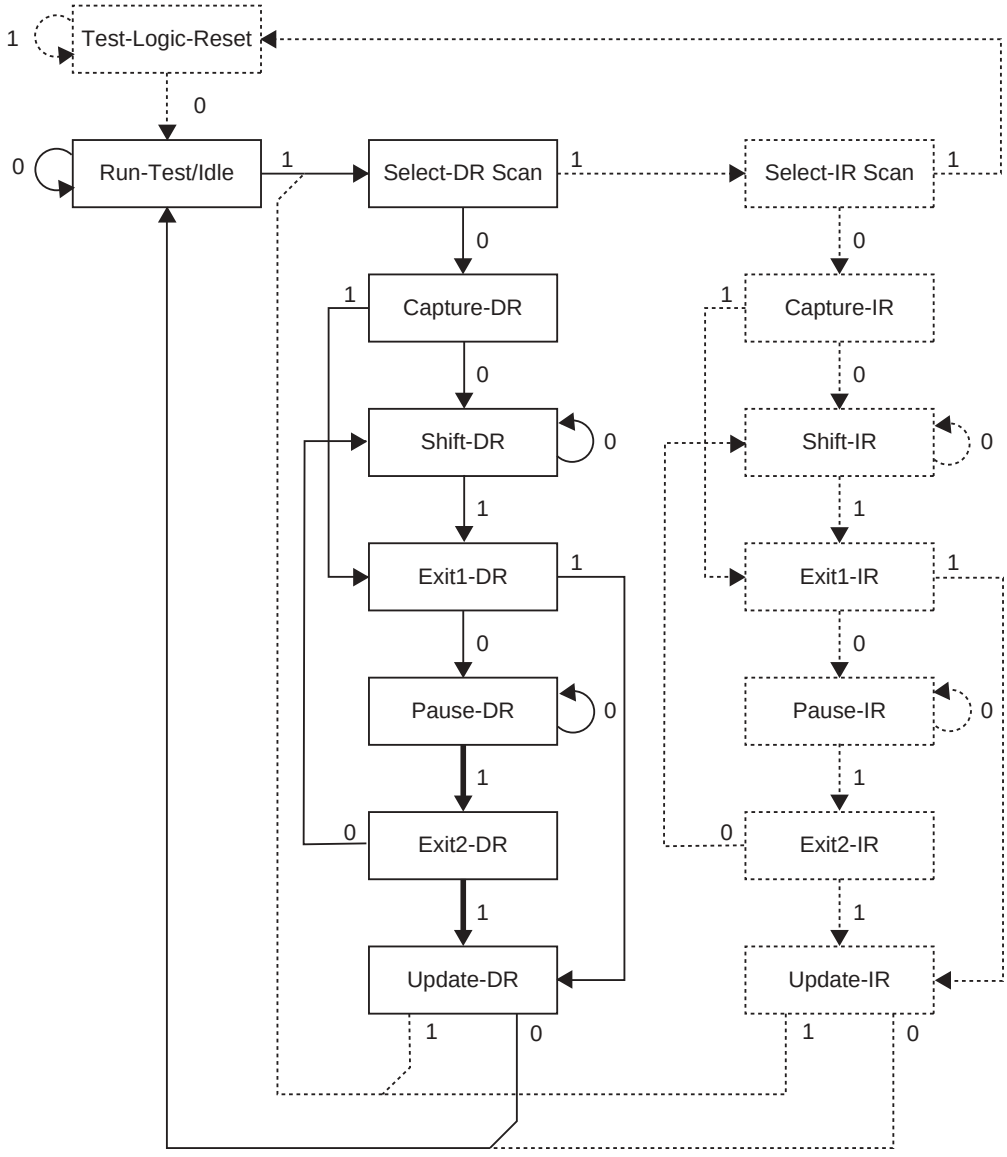
Table 69. Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Enable Serial Programming after $\overline{\text{RESET}}$ goes low.
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase EEPROM and Flash.
Read Program Memory	0010 **H**000	xx**aa aaaa**	**bbbb bbbb**	**oooo oooo**	Read **H** (high or low) data **o** from Program memory at word address **a**:**b**.
Load Program Memory Page	0100 **H**000	xxxx xxxx	xx**bb bbbb**	**iiii iiii**	Write **H** (high or low) data **i** to Program memory page at word address **b**.
Write Program Memory Page	0100 1100	xx**aa aaaa**	**bb**xx xxxx	xxxx xxxx	Write Program memory Page at address **a**:**b**.
Read EEPROM Memory	1010 0000	xxxx xx**aa**	**bbbb bbbb**	**oooo oooo**	Read data **o** from EEPROM Memory at address **a**:**b**.
Write EEPROM Memory	1100 0000	xxxx xx**aa**	**bbbb bbbb**	**iiii iiii**	Write data **i** to EEPROM Memory at address **a**:**b**.
Read Lock Bits	0101 1000	0000 0000	xxxx xxxx	xx***65 4321***	Read Lock bits. “0” = programmed, “1” = unprogrammed.
Write Lock Bits	1010 1100	111x xxxx	xxxx xxxx	11***65 4321***	Write Lock bits. Set bits ***6*** - ***1*** = “0” to program Lock bits.
Read Signature Byte	0011 0000	xxxx xxxx	xxxx xx**bb**	**oooo oooo**	Read Signature Byte **o** at address **b**.
Write Fuse Bits	1010 1100	1010 0000	xxxx xxxx	**CB**11 **A987**	Set bits **C** - **A**, **9** - **7** = “0” to program, “1” to unprogram
Write Fuse High Bits	1010 1100	1010 1000	xxxx xxxx	**IH**11 **GFED**	Set bits **F** - **D** = “0” to program, “1” to unprogram
Read Fuse Bits	0101 0000	0000 0000	xxxx xxxx	**CB**xx **A987**	Read Fuse bits. “0” = programmed, “1” = unprogrammed
Read Fuse High Bits	0101 1000	0000 1000	xxxx xxxx	**IH**xx **GFED**	Read Fuse high bits. “0” = programmed, “1” = unprogrammed
Read Calibration Byte	0011 1000	xxxx xxxx	0000 0000	**oooo oooo**	Read Signature Byte **o** at address **b**.

Note: **a** = address high bits; **b** = address low bits; **H** = 0 - Low Byte, 1 - High Byte; **o** = data out; **i** = data in; x = don’t care;
1 = lock bit 1; ***2*** = lock bit 2; ***3*** = Boot Lock Bit01; ***4*** = Boot Lock Bit02; ***5*** = Boot Lock Bit11; ***6*** = Boot Lock Bit12;
7 = CKSEL0 Fuse; **8** = CKSEL1 Fuse; **9** = CKSEL2 Fuse; **A** = CKSEL3 Fuse; **B** = BODEN Fuse; **C** = BODLEVEL Fuse;
D = BOOTRST Fuse; **E** = BOOTSZ0 Fuse; **F** = BOOTSZ1 Fuse; **G** = EESAVE Fuse; **H** = JTAGEN Fuse; and **I** = OCDEN Fuse

4. Set bits to “0” to program the corresponding lock bit, “1” to leave the lock bit unchanged.
5. “0” = programmed, “1” = unprogrammed.
6. **a** = address High Byte; **b** = address Low Byte; **i** = data in; **o** = data out; ***1*** = lock bit 1; ***2*** = lock bit 2; ***3*** = Boot Lock Bit01;
   4 = Boot Lock Bit02; ***5*** = Boot Lock Bit11; ***6*** = Boot Lock Bit12; **7** = CKSEL0 Fuse; **8** = CKSEL1 Fuse; **9** = CKSEL2 Fuse;
   A = CKSEL3 Fuse; **B** = BODEN Fuse; **C** = BODLEVEL Fuse; **D** = BOOTRST Fuse; **E** = BOOTSZ0 Fuse; **F** = BOOTSZ1 Fuse; **G** = EESAVE Fuse; **H** = JTAGEN Fuse; **I** = OCDEN Fuse

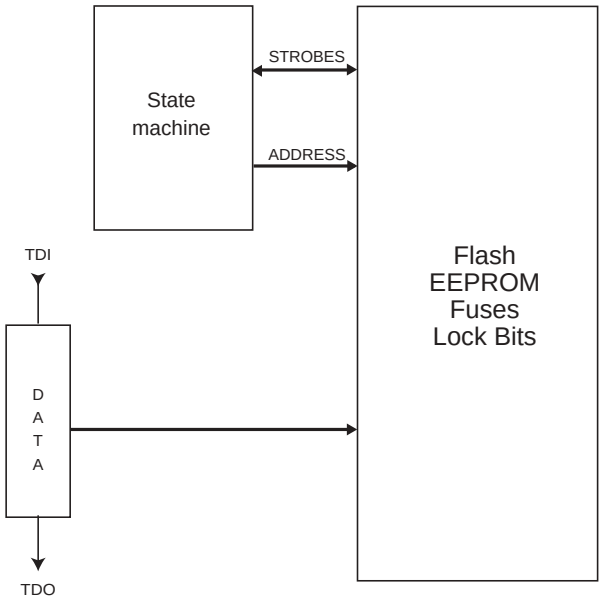
Figure 103. State Machine Sequence for Changing / Reading the Data Word



Virtual Flash Page Load Register

The Virtual Flash Page Load Register is a virtual scan chain with length equal to the number of bits in one Flash page, 1,024. Internally the Shift Register is 8-bit, and the data are automatically transferred to the Flash page buffer byte-by-byte. Shift in all instruction words in the page, starting with the LSB of the instruction with page address 0 and ending with the MSB of the instruction with page address 3F. This provides an efficient way to load the entire Flash page buffer before executing Page Write.

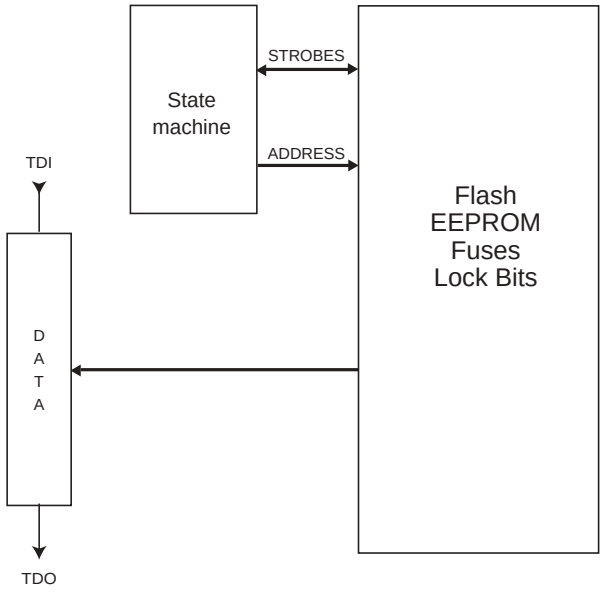
Figure 104. Virtual Flash Page Load Register



Virtual Flash Page Read Register

The Virtual Flash Page Read Register is a virtual scan chain with length equal to the number of bits in one Flash page plus eight, 1,032 in total. Internally the Shift Register is 8-bit, and the data are automatically transferred from the Flash data page byte-by-byte. The first eight cycles are used to transfer the first byte to the internal Shift Register, and the bits that are shifted out during these eight cycles should be ignored. Following this initialization, data are shifted out starting with the LSB of the instruction with page address 0 and ending with the MSB of the instruction with page address 3F. This provides an efficient way to read one full Flash page to verify programming.

Figure 105. Virtual Flash Page Read Register



Programming algorithm

All references below of type “1a”, “1b”, and so on, refer to Table 71.

Entering programming mode

1. Enter JTAG instruction AVR_RESET and shift 1 in the Reset Register.
2. Enter instruction PROG_ENABLE and shift 1010_0011_0111_0000 in the Programming Enable Register.

Leaving Programming Mode

1. Enter JTAG instruction PROG_COMMANDS.
2. Disable all programming instructions by using no operation instruction 11a.
3. Enter instruction PROG_ENABLE and shift 0000_0000_0000_0000 in the programming Enable Register.
4. Enter JTAG instruction AVR_RESET and shift 0 in the Reset Register.

If PROG_ENABLE instruction is not followed by the AVR_RESET instruction, the following algorithm should be used:

1. Enter JTAG instruction PROG_COMMANDS.
2. Disable all programming instructions by using no operation instruction 11a.
3. Enter instruction PROG_ENABLE and shift 0000_0000_0000_0000 in the Programming Enable Register.
4. Enter instruction PROG_ENABLE and shift 0000_0000_0000_0000 in the Programming Enable Register.
5. Wait until the selected Oscillator has started before applying more commands.

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
4	2.7 - 5.5V	ATmega323L-4AC ATmega323L-4PC	44A 40P6	Commercial (0°C to 70°C)
		ATmega323L-4AI ATmega323L-4PI	44A 40P6	Industrial (-40°C to 85°C)
8	4.0 - 5.5V	ATmega323-8AC ATmega323-8PC	44A 40P6	Commercial (0°C to 70°C)
		ATmega323-8AI ATmega323-8PI	44A 40P6	Industrial (-40°C to 85°C)

Package Type	
44A	44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
40P6	40-pin, 0.600” Wide, Plastic Dual Inline Package (PDIP)

About Code Examples 75
AVR USART vs. AVR UART – Compatibility 75

Clock Generation 76
Frame Formats 78
Parity Bit Calculation 79
USART Initialization 79
Data Transmission – The USART Transmitter 81
Data Reception – The USART Receiver 84
Asynchronous Data Reception 88
Multi-processor Communication Mode 91
Accessing UBRRH/UCSRC Registers 92
USART Register Description 94
Examples of Baud Rate Setting 99

Two-wire Serial Interface (Byte Oriented) 102
Two-wire Serial Interface Modes 107
Master Transmitter Mode 108
Master Receiver Mode 108
Slave Receiver Mode 109
Slave Transmitter Mode 110
Miscellaneous States 110

The Analog Comparator 124
Analog Comparator Multiplexed Input 126

Analog to Digital Converter 127
Features 127
Operation 128
Prescaling and Conversion Timing 129
ADC Noise Canceler Function 131
Scanning Multiple Channels 135
ADC Noise Canceling Techniques 135
ADC Characteristics – Preliminary Data 136

I/O Ports 137
Port A 137
Port B 139
Port C 146
Port D 151

JTAG Interface and the On-chip Debug System 157
Features 157
Overview 157
The Test Access Port – TAP 158
Using the Boundary-Scan Chain 161

Register Summary *231*

Instruction Set Summary *233*

Ordering Information *236*

Packaging Information *237*

44A 237

40P6 238

Errata for ATmega323 Rev. B *239*

Datasheet Change Log for ATmega323 ..... *242*

Changes from Rev. 1457F – 09/02 to Rev. 1457G – 09/03 242

Changes from Rev. 1457E – 11/01 to Rev. 1457F – 09/02 242

Table of Contents ..... *i*