



Sri Lanka Institute of Information Technology

B.Sc. Honours Degree in Information Technology  
Specialized in Computer Systems and Network Engineering

Final Examination  
Paper  
Year 2, Semester 2 (2022)

IE2070 – Embedded Systems

Duration: 2 Hours

2022

Instructions to Candidates:

- ◆ This paper has 4 questions.
- ◆ Answer all questions in the booklet given/paper itself.
- ◆ The total marks for the paper is 100.
- ◆ This paper contains 35 pages, including the cover page.
- ◆ Students are allowed to use calculators

**Question 1 [ 25 Marks]**

1. Mention 4 most important considerations when choosing a micro controller for a given application. [4 Marks]
2. State whether the following statements are true or false. [6 Marks]
  - a) No value can be directly added to the GPR
  - b) I/O registers are used for storing data
  - c) No value can be loaded directly into internal SRAM
  - d) Compiler converts C programs in to Assembly language first
  - e) The GPR together with I/O registers and SRAM are called Data memory
  - f) 256 is the largest value that can be moved into a location in the data memory
3. Show the status of the C, H and Z flags after the addition of 0x38 and 0X2F in the following instructions. [6 Marks]
 

```
LDI R15,0X38
      LDI R17,0X2F
      ADD R15,R17
```
4. Write an assembly program which subtracts 3 from the contents of location 0x300 and stores the result in location 0x320. [9 Marks]

**Question 2 [25 marks]**

You are asked to design an embedded system that limits the number of students inside a lecture theatre at the SLIIT Malabe Campus as a measure of preventing the spread of COVID-19. Hint: You may assume the student number to be less than 250

- a) Clearly explain with the aid of a diagram, how you plan to identify if a student is entering to the lecture theatre or leaving the lecture theatre. [4 marks]
- b) Using a flow chart explain the algorithm you plan to introduce if a buzzer has to sound when the number of students within the lecture theatre exceeds 50 and stop buzzer when the number falls below 50. [6 marks]
- c) What is the advantage of using interrupts in this solution? [3 marks]
- d) Write an ISR/ISRs in assembly for the foreground process of your design. [7 marks]
- e) List the interrupt source/sources used and justify your choice of interrupt source/sources for the given problem. [5 marks]

**Question 3****[25 Marks]**

1. Calculate the time taken for one machine cycle using the following oscillators: [6 Marks]
  - a) 4MHz
  - b) 2MHz
  - c) 16MHz
2. The round robin session in a Toastmasters club has a time limit of 30 seconds for each speaker. Use a Timer Module in ATMEGA328P to design an autonomous system that notifies the speaker when the allotted time is over. XTAL=16MHz. (Assume that PORTB0 pin gets an input of logic 1 when the speaker starts talking.)
  - a) Illustrate using a diagram the hardware circuit that you propose. [4 Marks]
  - b) Write an Assembly program **OR** a C program for your proposed hardware. State any assumptions that you may have made. [15 Marks]

**Question 4****[25 Marks]**

1. Name the conversion technique is used in the Analog to Digital converter in ATMega328A microcontroller? [1 Mark]
2. If an analog signal from a temperature sensor outputs 0V to 2V and if you are using an 8-bit ADC,
  - a. What is the smallest step size that can be achieved? Clearly show any calculation involved. [2 Marks]
  - b. How do you configure the ADC for this requirement? [2 Marks]
3. What is the advantage in using interrupts for a multi-channel ADC measurement application compared to polling? [2 Marks]
4. What is an advantage and a disadvantage in serial communications over parallel communication? [2 Marks]
5. With FOSC = 16 MHz, find the UBRR value in hexadecimal needed to have the following baud rates:
  - a. 9600
  - b. 4800
  - c. 2400
  - d. 1200
6. List down the 6 steps required to transfer character bytes serially using ATMega328A. [6 Marks]
7. Explain using a diagram the function of the shift register in the receive unit of the UART clearly indicating the LSB, MSB, start and stop bits. [6 Marks]

## 14.9 Register Description

### 14.9.1 TCCR0A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bits 7:6 – COM0A1:0: Compare Match Output A Mode

These bits control the output compare pin (OC0A) behavior. If one or both of the COM0A1:0 bits are set, the OC0A output overrides the normal port functionality of the I/O pin it is connected to. However, note that the data direction register (DDR) bit corresponding to the OC0A pin must be set in order to enable the output driver.

When OC0A is connected to the pin, the function of the COM0A1:0 bits depends on the WGM02:0 bit setting. Table 14-2 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 14-2. Compare Output Mode, non-PWM Mode

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected
0	1	Toggle OC0A on compare match
1	0	Clear OC0A on compare match
1	1	Set OC0A on compare match (non-inverting mode)

Table 14-3 shows the COM0A1:0 bit functionality when the WGM01:0 bits are set to fast PWM mode.

Table 14-3. Compare Output Mode, Fast PWM Mode<sup>(1)</sup>

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected
0	1	WGM02 = 0: Normal port operation, OC0A disconnected WGM02 = 1: Toggle OC0A on compare match
1	0	Clear OC0A on compare match, set OC0A at BOTTOM (non-inverting mode)
1	1	Set OC0A on compare match, clear OC0A at BOTTOM (non-inverting mode)

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 is set. In this case, the compare match is ignored, but the set or clear is done at BOTTOM. See Section 14.7.3 "Fast PWM Mode" on page 80 for more details.

Table 14-4 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 14-4. Compare Output Mode, Phase Correct PWM Mode<sup>(1)</sup>

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected
0	1	WGM02 = 0: Normal port operation, OC0A disconnected WGM02 = 1: Toggle OC0A on compare match
1	0	Clear OC0A on compare match when up-counting, Set OC0A on compare match when down-counting
1	1	Set OC0A on compare match when up-counting, Clear OC0A on compare match when down-counting

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 is set. In this case, the compare match is ignored, but the set or clear is done at TOP. See Section 15.9.4 "Phase Correct PWM Mode" on page 103 for more details.

- Bits 5:4 – COM0B1:0: Compare Match Output B Mode**

These bits control the output compare pin (OC0B) behavior. If one or both of the COM0B1:0 bits are set, the OC0B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the data direction register (DDR) bit corresponding to the OC0B pin must be set in order to enable the output driver.

When OC0B is connected to the pin, the function of the COM0B1:0 bits depends on the WGM02:0 bit setting. Table 14-5 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

**Table 14-5. Compare Output Mode, non-PWM Mode**

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on compare match.
1	1	Set OC0B on compare match.

Table 14-6 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to fast PWM mode.

**Table 14-6. Compare Output Mode, Fast PWM Mode<sup>(1)</sup>**

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on compare match, set OC0B at BOTTOM, (non-inverting mode).
1	1	Set OC0B on compare match when TOP or BOTTOM, inverting mode.

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 is set. In this case, the compare match is ignored, but the set or clear is done at TOP. See Section 14.7.3 "Fast PWM Mode" on page 80 for more details.

Table 14-7 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

**Table 14-7. Compare Output Mode, Phase Correct PWM Mode<sup>(1)</sup>**

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on compare match when up-counting. Set OC0B on compare match when down-counting.
1	1	Set OC0B on compare match when up-counting. Clear OC0B on compare match when down-counting.

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 is set. In this case, the compare match is ignored, but the set or clear is done at TOP. See Section 14.7.4 "Phase Correct PWM Mode" on page 81 for more details.

- Bits 3, 2 – Res: Reserved Bits**

These bits are reserved bits in the Atmel® ATmega328P and will always read as zero.

- Bits 1:0 – WGM01:0: Waveform Generation Mode**

Combined with the WGM02 bit found in the TCCR0B register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see Table 14-8. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), clear timer on compare match (CTC) mode, and two types of pulse width modulation (PWM) modes (see Section 14.7 “Modes of Operation” on page 78).

**Table 14-8. Waveform Generation Mode Bit Description**

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on <sup>(1)(2)</sup>
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	CTC	OCRA	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Reserved	OCRA	BOTTOM	MAX
4	1	0	0	Reserved	-	-	-
5	1	0	1	Reserved	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	OCRA	BOTTOM	TOP
7	1	1	1	Reserved	OCRA	TOP	BOTTOM

Notes:

1. MAX = 0xFF
2. BOTTOM = 0x00

#### 14.9.2 TCCR0B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00	TCCR0B
ReadWrite	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7 – FOC0A: Force Output Compare A**

The FOC0A bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0A bit, an immediate compare match is forced on the waveform generation unit. The OC0A output is changed according to its COM0A1:0 bits setting. Note that the FOC0A bit is implemented as a strobe. Therefore it is the value present in the COM0A1:0 bits that determines the effect of the forced compare.

A FOC0A strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0A as TOP.

The FOC0A bit is always read as zero.

- Bit 6 – FOC0B: Force Output Compare B**

The FOC0B bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0B bit, an immediate compare match is forced on the waveform generation unit. The OC0B output is changed according to its COM0B1:0 bits setting. Note that the FOC0B bit is implemented as a strobe. Therefore it is the value present in the COM0B1:0 bits that determines the effect of the forced compare.

A FOC0B strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0B as TOP.

The FOC0B bit is always read as zero.

- Bits 5:4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega328P and will always read as zero.

- Bit 3 – WGM02: Waveform Generation Mode

See the description in the Section 14.9.1 "TCCR0A – Timer/Counter Control Register A" on page 84.

- Bits 2:0 – CS02:0: Clock Select

The three clock select bits select the clock source to be used by the Timer/Counter.

**Table 14-9. Clock Select Bit Description**

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
1	0	1	clk <sub>IO</sub> (no prescaling)
0	1	0	clk <sub>IO</sub> /8 (from prescaler)
0	1	1	clk <sub>IO</sub> /64 (from prescaler)
1	0	0	clk <sub>IO</sub> /256 (from prescaler)
1	0	1	clk <sub>IO</sub> /1024 (from prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge
1	1	1	External clock source on T0 pin. Clock on rising edge

If external pin modes are used for the Timer/Counter0, transitions on the T0 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

#### 14.9.3 TCNT0 – Timer/Counter Register

Bit	7	6	5	4	3	2	1	0	
0x26 (0x46)	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Timer/Counter register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNT0 register blocks (removes) the compare match on the following timer clock. Modifying the counter (TCNT0) while the counter is running, introduces a risk of missing a compare match between TCNT0 and the OCR0x registers.

#### 14.9.4 OCR0A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
0x27 (0x47)	OCR0A[7:0]								OCR0A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The output compare register A contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC0A pin.

#### 14.9.5 OCR0B – Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
0x28 (0x48)	OCR0B[7:0]								OCR0B
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The output compare register B contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC0B pin.

#### 14.9.6 TIMSK0 – Timer/Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6E)	—	—	—	—	—	OCIE0B	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..3 – Res: Reserved Bits**

These bits are reserved bits in the Atmel® ATmega328P and will always read as zero.

- **Bit 2 – OCIE0B: Timer/Counter Output Compare Match B Interrupt Enable**

When the OCIE0B bit is written to one, and the I-bit in the status register is set, the Timer/Counter compare match B interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter occurs, i.e., when the OCF0B bit is set in the Timer/Counter interrupt flag register – TIFR0.

- **Bit 1 – OCIE0A: Timer/Counter0 Output Compare Match A Interrupt Enable**

When the OCIE0A bit is written to one, and the I-bit in the status register is set, the Timer/Counter0 compare match A interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter0 occurs, i.e., when the OCF0A bit is set in the Timer/Counter 0 interrupt flag register – TIFR0.

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

When the TOIE0 bit is written to one, and the I-bit in the status register is set, 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 0 interrupt flag register – TIFR0.

#### 14.9.7 TIFR0 – Timer/Counter 0 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x15 (0x35)	—	—	—	—	—	OCF0B	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..3 – Res: Reserved Bits**

These bits are reserved bits in the Atmel ATmega328P and will always read as zero.

- **Bit 2 – OCF0B: Timer/Counter 0 Output Compare B Match Flag**

The OCF0B bit is set when a compare match occurs between the Timer/Counter and the data in OCR0B – output compare register0 B. OCF0B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0B is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0B (Timer/Counter compare B match interrupt enable), and OCF0B are set, the Timer/Counter compare match interrupt is executed.

- **Bit 1 – OCF0A: Timer/Counter 0 Output Compare A Match Flag**

The OCF0A bit is set when a compare match occurs between the Timer/Counter0 and the data in OCR0A – output compare register0. OCF0A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0A is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0A (Timer/Counter0 compare match interrupt enable), and OCF0A are set, the Timer/Counter0 compare match interrupt is executed.

- **Bit 0 – TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE0 (Timer/Counter0 overflow interrupt enable), and TOV0 are set, the Timer/Counter0 overflow interrupt is executed.

The setting of this flag is dependent of the WGM02:0 bit setting. Refer to Table 14-8 on page 86, Section 14-8 "Waveform Generation Mode Bit Description" on page 86.

## 15.11 Register Description

### 15.11.1 TCCR1A – Timer/Counter1 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7:6 – COM1A1:0: Compare Output Mode for Channel A

- Bit 5:4 – COM1B1:0: Compare Output Mode for Channel B

The COM1A1:0 and COM1B1:0 control the output compare pins (OC1A and OC1B respectively) behavior. If one or both of the COM1A1:0 bits are written to one, the OC1A output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COM1B1:0 bit are written to one, the OC1B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the data direction register (DDR) bit corresponding to the OC1A or OC1B pin must be set in order to enable the output driver.

When the OC1A or OC1B is connected to the pin, the function of the COM1x1:0 bits is dependent of the WGM13:0 bits setting. Table 15-2 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to a normal or a CTC mode (non-PWM).

Table 15-2. Compare Output Mode, non-PWM

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	Toggle OC1A/OC1B on compare match.
1	0	Clear OC1A/OC1B on compare match (set output to low level).
1	1	Set OC1A/OC1B on compare match (set output to high level).

Table 15-3 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the fast PWM mode.

Table 15-3. Compare Output Mode, Fast PWM<sup>(1)</sup>

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 1101: Toggle OC1A on compare match, OC1B disconnected (normal port operation). For all other WGM13:0 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on compare match, set OC1A/OC1B at BOTTOM (non-inverting mode).
1	1	Set OC1A/OC1B on compare match, clear OC1A/OC1B at BOTTOM (inverting mode).

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. In this case the compare match is ignored, but the set or clear is done at BOTTOM. See Section 15.9.3 "Fast PWM Mode" on page 101 for more details.

Table 15-4 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the phase correct or the phase and frequency correct, PWM mode.

**Table 15-4. Compare Output Mode, Phase Correct and Phase and Frequency Correct PWM<sup>(1)</sup>**

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 0011 (Top=0): OC1A/OC1B compare match (OC1A=TOP). Clear timer on compare match (OC1A=0). Set OC1A/OC1B on compare match when up-counting. Set OC1A/OC1B on compare match when down-counting.
1	0	Clear OC1A/OC1B on compare match when up-counting. Set OC1A/OC1B on compare match when down-counting.
1	1	Set OC1A/OC1B on compare match when up-counting. Clear OC1A/OC1B on compare match when down-counting.

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. See Section 15.9.4 "Phase Correct PWM Mode" on page 103 for more details.

- Bit 1:0 – WGM11:0: Waveform Generation Mode**

Combined with the WGM13:2 bits found in the TCCR1B register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see Table 15-5. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), clear timer on compare match (CTC) mode, and three types of pulse width modulation (PWM) modes. See (Section 15.9 "Modes of Operation" on page 100).

**Table 15-5. Waveform Generation Mode Bit Description<sup>(1)</sup>**

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, phase correct, 9-bit	0xFFFF	TOP	BOTTOM
2	0	0	1	0	PWM, phase correct, 10-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, phase correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 9-bit	0xFFFF	Bottom	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	Bottom	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	Bottom	TOP
8	1	0	0	0	PWM, phase and frequency correct	ICR1	Bottom	BOTTOM
9	1	0	0	1	PWM, phase and frequency correct	OCR1A	Bottom	BOTTOM
10	1	0	1	0	PWM, phase correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, phase correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	(Reserved)			
14	1	1	1	0	Fast PWM	ICR1	Bottom	TOP
15	1	1	1	1	Fast PWM	OCR1A	Bottom	TOP

Note: 1. The CTC1 and PWM11:0 bit definition names are obsolete. Use the WGM12:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

### 15.11.2 TCCR1B – Timer/Counter1 Control Register B

Bit	7	6	5	4	3	2	1	0	TCCR1B
(0x81)	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	
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 WGM13:0 bits located in the TCCR1A and the TCCR1B register), the ICP1 is disconnected and consequently the input capture function is disabled.

- **Bit 5 – 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.

- **Bit 4:3 – WGM13:2: Waveform Generation Mode**

See TCCR1A register description.

- **Bit 2:0 – CS12:0: Clock Select**

The three clock select bits select the clock source to be used by the Timer/Counter, see Figure 15-10 on page 106 and Figure 15-11 on page 106.

**Table 15-6. Clock Select Bit Description**

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$\text{clk}_{\text{IO}}$ (no prescaling)
0	1	0	$\text{clk}_{\text{IO}}/8$ (from prescaler)
0	1	1	$\text{clk}_{\text{IO}}/64$ (from prescaler)
1	0	0	$\text{clk}_{\text{IO}}/256$ (from prescaler)
1	0	1	$\text{clk}_{\text{IO}}/1024$ (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.

If external pin modes are used for the Timer/Counter1, transitions on the T1 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

### 15.11.3 TCCR1C – Timer/Counter1 Control Register C

Bit	7	6	5	4	3	2	1	0	
(0x82)	FOC1A	FOC1B	-	-	-	-	-	-	TCCR1C
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7 – FOC1A: Force Output Compare for Channel A

- Bit 6 – FOC1B: Force Output Compare for Channel B

The FOC1A/FOC1B bits are only active when the WGM13:0 bits specifies a non-PWM mode. When writing a logical one to the FOC1A/FOC1B bit, an immediate compare match is forced on the waveform generation unit. The OC1A/OC1B output is changed according to its COM1x1:0 bits setting. Note that the FOC1A/FOC1B bits are implemented as strobes. Therefore it is the value present in the COM1x1:0 bits that determine the effect of the forced compare.

A FOC1A/FOC1B strobe will not generate any interrupt nor will it clear the timer in clear timer on compare match (CTC) mode using OCR1A as TOP. The FOC1A/FOC1B bits are always read as zero.

### 15.11.4 TCNT1H and TCNT1L – Timer/Counter1

Bit	7	6	5	4	3	2	1	0	
(0x85)	TCNT1[15:8]								TCNT1H
(0x84)	TCNT1[7:0]								TCNT1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The two Timer/Counter I/O locations (TCNT1H and TCNT1L, combined TCNT1) give direct access, both for read and for write operations, to the Timer/Counter unit 16-bit counter. To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers. See Section 15.3 "Accessing 16-bit Registers" on page 91.

Modifying the counter (TCNT1) while the counter is running introduces a risk of missing a compare match between TCNT1 and one of the OCR1x registers.

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

### 15.11.5 OCR1AH and OCR1AL – Output Compare Register 1 A

Bit	7	6	5	4	3	2	1	0	
(0x89)	OCR1A[15:8]								OCR1AH
(0x88)	OCR1A[7:0]								OCR1AL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

### 15.11.6 OCR1BH and OCR1BL – Output Compare Register 1 B

Bit	7	6	5	4	3	2	1	0	
(0x8B)	OCR1B[15:8]								OCR1BH
(0x8A)	OCR1B[7:0]								OCR1BL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The output compare registers contain a 16-bit value that is continuously compared with the counter value (TCNT1). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC1x pin.

The output compare registers are 16-bit in size. To ensure that both the high and low bytes are written simultaneously when the CPU writes to these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers. See Section 15.3 "Accessing 16-bit Registers" on page 91.

### 15.11.7 ICR1H and ICR1L – Input Capture Register 1

Bit	7	6	5	4	3	2	1	0	
(0x87)	ICR1[15:8]								ICR1H
(0x86)	ICR1[7:0]								ICR1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The input capture is updated with the counter (TCNT1) value each time an event occurs on the ICP1 pin (or optionally on the analog comparator output for Timer/Counter1). The input capture can be used for defining the counter TOP value.

The input capture register is 16-bit in size. To ensure that both the high and low bytes are read simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers. See Section 15.3 "Accessing 16-bit Registers" on page 91.

### 15.11.8 TIMSK1 – Timer/Counter1 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6F)	-	-	ICIE1	-	-	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 6 – Res: Reserved Bits**

These bits are unused bits in the Atmel® ATmega328P, and will always read as zero.

- **Bit 5 – ICIE1: Timer/Counter1, Input Capture Interrupt Enable**

When this bit is written to one, and the I-flag in the status register is set (interrupts globally enabled), the Timer/Counter1 input capture interrupt is enabled. The corresponding interrupt vector (see Section 11. "Interrupts" on page 49) is executed when the ICF1 flag, located in TIFR1, is set.

- **Bit 4, 3 – Res: Reserved Bits**

These bits are unused bits in the Atmel ATmega328P, and will always read as zero.

- **Bit 2 – OCIE1B: Timer/Counter1, Output Compare B Match Interrupt Enable**

When this bit is written to one, and the I-flag in the status register is set (interrupts globally enabled), the Timer/Counter1 output compare B match interrupt is enabled. The corresponding interrupt vector (see Section 11. "Interrupts" on page 49) is executed when the OCF1B flag, located in TIFR1, is set.

- **Bit 1 – OCIE1A: Timer/Counter1, Output Compare A Match Interrupt Enable**

When this bit is written to one, and the I-flag in the status register is set (interrupts globally enabled), the Timer/Counter1 output compare A match interrupt is enabled. The corresponding interrupt vector (see Section 11. "Interrupts" on page 49) is executed when the OCF1A flag, located in TIFR1, is set.

- **Bit 0 – TOIE1: Timer/Counter1, Overflow Interrupt Enable**

When this bit is written to one, and the I-flag in the status register is set (interrupts globally enabled), the Timer/Counter1 overflow interrupt is enabled. The corresponding interrupt vector (see Section 11. "Interrupts" on page 49) is executed when the TOV1 flag, located in TIFR1, is set.

### 15.11.9 TIFR1 – Timer/Counter1 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1	TIFR1
ReadWrite	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 6 – Res: Reserved Bits**

These bits are unused bits in the Atmel® ATmega328P, and will always read as zero.

- **Bit 5 – ICF1: Timer/Counter1, Input Capture Flag**

This flag is set when a capture event occurs on the ICP1 pin. When the input capture register (ICR1) is set by the WGM13:0 to be used as the TOP value, the ICF1 flag is set when the counter reaches the TOP value.

ICF1 is automatically cleared when the input capture interrupt vector is executed. Alternatively, ICF1 can be cleared by writing a logic one to its bit location.

- **Bit 4, 3 – Res: Reserved Bits**

These bits are unused bits in the Atmel ATmega328P, and will always read as zero.

- **Bit 2 – OCF1B: Timer/Counter1, Output Compare B Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the output compare register B (OCR1B).

Note that a forced output compare (FOC1B) strobe will not set the OCF1B flag.

OCF1B is automatically cleared when the output compare match B interrupt vector is executed. Alternatively, OCF1B can be cleared by writing a logic one to its bit location.

- **Bit 1 – OCF1A: Timer/Counter1, Output Compare A Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the output compare register A (OCR1A).

Note that a forced output compare (FOC1A) strobe will not set the OCF1A flag.

OCF1A is automatically cleared when the output compare match A interrupt vector is executed. Alternatively, OCF1A can be cleared by writing a logic one to its bit location.

- **Bit 0 – TOV1: Timer/Counter1, Overflow Flag**

The setting of this flag is dependent of the WGM13:0 bits setting. In normal and CTC modes, the TOV1 flag is set when the timer overflows. Refer to Table 15-5 on page 109 for the TOV1 flag behavior when using another WGM13:0 bit setting.

TOV1 is automatically cleared when the Timer/Counter1 overflow interrupt vector is executed. Alternatively, TOV1 can be cleared by writing a logic one to its bit location.

## 11. Interrupts

This section describes the specifics of the interrupt handling as performed in Atmel® ATmega328P. For a general explanation of the AVR® interrupt handling, refer to Section 6.7 "Reset and Interrupt Handling" on page 15.

- Each interrupt vector occupies two instruction words in Atmel ATmega328P.
- In Atmel ATmega328P, the reset vector is affected by the BOOTRST fuse, and the interrupt vector start address is affected by the IVSEL bit in MCUCR.

### 11.1 Interrupt Vectors in ATmega328P

Table 11-1. Reset and Interrupt Vectors in ATmega328P

Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x0021	INT0	External interrupt request 0
3	0x004	INT1	External interrupt request 1
4	0x006	PCINT0	Pin change interrupt request 0
5	0x008	PCINT1	Pin change interrupt request 1
6	0x00A	PCINT2	Pin change interrupt request 2
7	0x00C	WDT	Watchdog time-out interrupt
8	0x00E	TIMER2 COMP A	Timer/Counter2 compare match A
9	0x010	TIMER2 COMP B	Timer/Counter2 compare match B
10	0x012	TIMER2 OVF	Timer/Counter2 overflow
11	0x014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x016	TIMER1 COMP A	Timer/Counter1 compare match A
13	0x018	TIMER1 COMP B	Timer/Counter1 compare match B
14	0x01A	TIMER1 OVF	Timer/Counter1 overflow
15	0x01C	TIMER0 COMP A	Timer/Counter0 compare match A
16	0x01E	TIMER0 COMP B	Timer/Counter0 compare match B
17	0x020	TIMER0 OVF	Timer/Counter0 overflow
18	0x022	SPI STC	SPI serial transfer complete
19	0x024	USART, RX	USART Rx complete
20	0x026	USART UDRE	USART data register empty
21	0x028	USART, TX	USART, Tx complete
22	0x02A	ADC	ADC conversion complete
23	0x02C	EE READY	EEPROM ready
24	0x02E	ANALOG COMP	Analog comparator
25	0x030	TWI	2-wire serial interface
26	0x032	SPM READY	Store program memory ready

Table 11-2 shows reset and interrupt vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. This is also the case if the reset vector is in the application section while the interrupt vectors are in the boot section or vice versa.

**Table 11-2. Reset and Interrupt Vectors Placement in ATmega328P<sup>(1)</sup>**

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x002
1	1	0x000	Boot reset address + 0x012
0	0	Boot reset address	0x002
0	1	Boot reset address	Boot reset address + 0x002

Note: 1. For the BOOTRST fuse "1" means unprogrammed while "0" means programmed.

The most typical and general program setup for the reset and interrupt vector addresses in Atmel® ATmega328P is:

Address	Labels	Code	Comments
0x0000		jmp RESET	; Reset Handler
0x0002		jmp EXT_INT0	; IRQ0 Handler
0x0004		jmp EXT_INT1	; IRQ1 Handler
0x0006		jmp PCINT0	; PCINT0 Handler
0x0008		jmp PCINT1	; PCINT1 Handler
0x000A		jmp PCINT2	; PCINT2 Handler
0x000C		jmp WDT	; Watchdog Timer Handler
0x000E		jmp TIM2_COMPA	; Timer2 Compare A Handler
0x0010		jmp TIM2_COMPB	; Timer2 Compare B Handler
0x0012		jmp TIM2_OVF	; Timer2 Overflow Handler
0x0014		jmp TIM1_CAPT	; Timer1 Capture Handler
0x0016		jmp TIM1_COMPA	; Timer1 Compare A Handler
0x0018		jmp TIM1_COMPB	; Timer1 Compare B Handler
0x001A		jmp TIM1_OVF	; Timer1 Overflow Handler
0x001C		jmp TIM0_COMPA	; Timer0 Compare A Handler
0x001E		jmp TIM0_COMPB	; Timer0 Compare B Handler
0x0020		jmp TIM0_OVF	; Timer0 Overflow Handler
0x0022		jmp SPI_STC	; SPI Transfer Complete Handler
0x0024		jmp USART_RXC	; USART, RX Complete Handler
0x0026		jmp USART_UDRE	; USART, UDR Empty Handler
0x0028		jmp USART_TXC	; USART, TX Complete Handler
0x002A		jmp ADC	; ADC Conversion Complete Handler
0x002C		jmp EE_RDY	; EEPROM Ready Handler
0x002E		jmp ANA_COMP	; Analog Comparator Handler
0x0030		jmp TWI	; 2-wire Serial Interface Handler
0x0032		jmp SPM_RDY	; Store Program Memory Ready Handler
;			
0x0033	RESET:	ldi r16, high(RAMEND); Main program start	
0x0034		out SPH,r16 ; Set Stack Pointer to top of RAM	
0x0035		ldi r16, low(RAMEND)	
0x0036		out SPL,r16	
0x0037		sei ; Enable interrupts	
0x0038	<instr>	xxx	
...	...	...	

When the BOOTRST fuse is unprogrammed, the boot section size set to 2Kbytes and the IVSEL bit in the MCUCR register is set before any interrupts are enabled, the most typical and general program setup for the reset and interrupt vector addresses in Atmel® ATmega328P is:

Address	Labels	Code	Comments
0x0000	RESET:	ldi r16,high(RAMEND); Main program start	
0x0001		out SPH,r16 ; Set Stack Pointer to top of RAM	
0x0002		ldi r16,low(RAMEND)	
0x0003		out SPL,r16	
0x0004		sei ; Enable interrupts	
0x0005		<instr> xxx	
;			
.org 0x3C02			
0x3C02		jmp EXT_INT0 ; IRQ0 Handler	
0x3C04		jmp EXT_INT1 ; IRQ1 Handler	
...		...	;
0x3C32		jmp SPM_RDY ; Store Program Memory Ready Handler	

When the BOOTRST fuse is programmed and the boot section size set to 2Kbytes, the most typical and general program setup for the reset and interrupt vector addresses in Atmel ATmega328P is:

Address	Labels	Code	Comments
.org 0x0002			
0x0002		jmp EXT_INT0 ; IRQ0 Handler	
0x0004		jmp EXT_INT1 ; IRQ1 Handler	
...		...	;
0x0032		jmp SPM_RDY ; Store Program Memory Ready Handler	
;			
.org 0x3C00			
0x3C00	RESET:	ldi r16,high(RAMEND); Main program start	
0x3C01		out SPH,r16 ; Set Stack Pointer to top of RAM	
0x3C02		ldi r16,low(RAMEND)	
0x3C03		out SPL,r16	
0x3C04		sei ; Enable interrupts	
0x3C05		<instr> xxx	

When the BOOTRST fuse is programmed, the boot section size set to 2Kbytes and the IVSEL bit in the MCUCR register is set before any interrupts are enabled, the most typical and general program setup for the reset and interrupt vector addresses in Atmel ATmega328P is:

Address	Labels	Code	Comments
;			
.org 0x3C00			
0x3C00		jmp RESET ; Reset handler	
0x3C02		jmp EXT_INT0 ; IRQ0 Handler	
0x3C04		jmp EXT_INT1 ; IRQ1 Handler	
...		...	;
0x3C32		jmp SPM_RDY ; Store Program Memory Ready Handler	
;			
0x3C33	RESET:	ldi r16,high(RAMEND); Main program start	
0x3C34		out SPH,r16 ; Set Stack Pointer to top of RAM	
0x3C35		ldi r16,low(RAMEND)	
0x3C36		out SPL,r16	
0x3C37		sei ; Enable interrupts	
0x3C38		<instr> xxx	

## 11.2 Register Description

### 11.2.1 Moving Interrupts Between Application and Boot Space

The MCU control register controls the placement of the interrupt vector table.

### 11.2.2 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
	BOOTSZ	BOOSZ	PUB			I	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – IVSEL: Interrupt Vector Select**

When the IVSEL bit is cleared (zero), the interrupt vectors are placed at the start of the flash memory. When this bit is set (one), the interrupt vectors are moved to the beginning of the boot loader section of the flash. The actual address of the start of the boot flash section is determined by the BOOTSZ fuses. Refer to the Section 26. "Boot Loader Support – Read-While-Write Self-Programming" on page 229 for details. To avoid unintentional changes of interrupt vector tables, a special write procedure must be followed to change the IVSEL bit:

- a. Write the interrupt vector change enable (IVCE) bit to one.
- b. Within four cycles, write the desired value to IVSEL while writing a zero to IVCE.

Interrupts will automatically be disabled while this sequence is executed. Interrupts are disabled in the cycle IVCE is set, and they remain disabled until after the instruction following the write to IVSEL. If IVSEL is not written, interrupts remain disabled for four cycles. The I-bit in the status register is unaffected by the automatic disabling.

Note: If interrupt vectors are placed in the boot loader section and boot lock bit BLB02 is programmed, interrupts are disabled while executing from the application section. If interrupt vectors are placed in the application section and boot lock bit BLB12 is programmed, interrupts are disabled while executing from the boot loader section. Refer to the Section 26. "Boot Loader Support – Read-While-Write Self-Programming" on page 229 for details on boot lock bits.

This bit is not available in Atmel® ATmega328P.

- **Bit 0 – IVCE: Interrupt Vector Change Enable**

The IVCE bit must be written to logic one to enable change of the IVSEL bit. IVCE is cleared by hardware four cycles after it is written or when IVSEL is written. Setting the IVCE bit will disable interrupts, as explained in the IVSEL description above. See code example below.

#### Assembly Code Example

```
Move_interrupts:
    ; Enable change of Interrupt Vectors
    ldi    r16, (1<<IVCE)
    out   MCUCR, r16
    ; Move interrupts to Boot Flash section
    ldi    r16, (1<<IVSEL)
    out   MCUCR, r16
    ret
```

#### C Code Example

```
void Move_interrupts(void)
{
    /* Enable change of Interrupt Vectors */
    MCUCR = (1<<IVCE);
    /* Move interrupts to Boot Flash section */
    MCUCR = (1<<IVSEL);
}
```

This bit is not available in Atmel ATmega328P.

## 12. External Interrupts

The external interrupts are triggered by the INT0 and INT1 pins or any of the PCINT23..0 pins. Observe that, if enabled, the interrupts will trigger even if the INT0 and INT1 or PCINT23..0 pins are configured as outputs. This feature provides a way of generating a software interrupt. The pin change interrupt PCI2 will trigger if any enabled PCINT23..16 pin toggles. The pin change interrupt PCI1 will trigger if any enabled PCINT14..8 pin toggles. The pin change interrupt PCI0 will trigger if any enabled PCINT7..0 pin toggles. The PCMSK2, PCMSK1 and PCMSK0 registers control which pins contribute to the pin change interrupts. Pin change interrupts on PCINT23..0 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode.

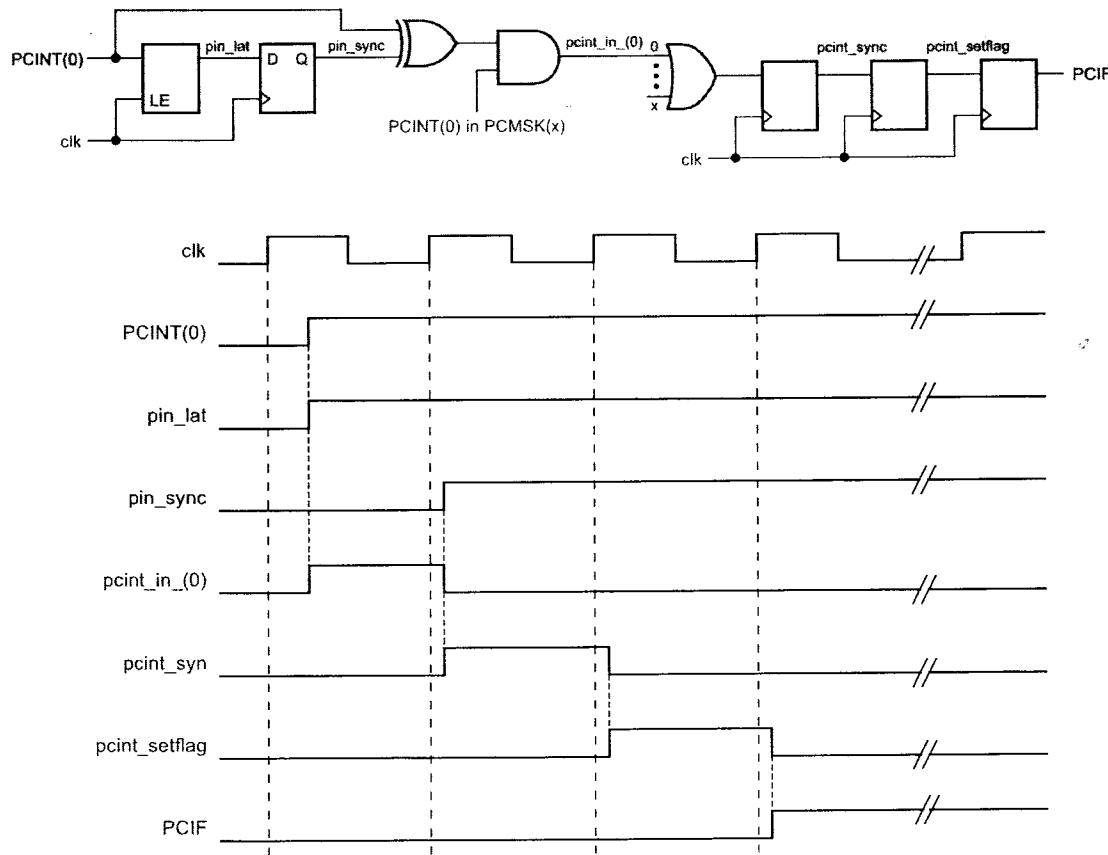
The INT0 and INT1 interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the external interrupt control register A – EICRA. When the INT0 or INT1 interrupts are enabled and are configured as level triggered, the interrupts will trigger as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT0 or INT1 requires the presence of an I/O clock, described in Section 8.1 “Clock Systems and their Distribution” on page 24. Low level interrupt on INT0 and INT1 is detected asynchronously. This implies that this interrupt can be used for waking the part also from sleep modes other than idle mode. The I/O clock is halted in all sleep modes except Idle mode.

Note that if a level triggered interrupt is used for wake-up from power-down, the required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the start-up time, the MCU will still wake up, but no interrupt will be generated. The start-up time is defined by the SUT and CKSEL fuses as described in Section 8. “System Clock and Clock Options” on page 24.

### 12.1 Pin Change Interrupt Timing

An example of timing of a pin change interrupt is shown in Figure 12-1.

**Figure 12-1. Timing of Pin Change Interrupts**



## 12.2 Register Description

### 12.2.1 EICRA – External Interrupt Control Register A

The external interrupt control register A contains control bits for interrupt sense control.

Bit	7	6	5	4	3	2	1	0	
(0x69)	-	-	-	-	ISC11	ISC10	ISC01	ISC00	EICRA
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..4 – Res: Reserved Bits**

These bits are unused bits in the Atmel® ATmega328P, and will always read as zero.

- **Bit 3, 2 – ISC11, ISC10: Interrupt Sense Control 1 Bit 1 and Bit 0**

The external interrupt 1 is activated by the external pin INT1 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT1 pin that activate the interrupt are defined in Table 12-1. The value on the INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 12-1. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logic change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

- **Bit 1, 0 – ISC01, ISC00: Interrupt Sense Control 0 Bit 1 and Bit 0**

The external interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT0 pin that activate the interrupt are defined in Table 12-2. The value on the INT0 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 12-2. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logic change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

## 12.2.2 EIMSK – External Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	EIMSK
0x1D (0x3D)	-	-	-	-	-	-	INT1	INT0	
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7..2 – Res: Reserved Bits

These bits are unused bits in the Atmel® ATmega328P, and will always read as zero.

- Bit 1 – INT1: External Interrupt Request 1 Enable

When the INT1 bit is set (one) and the I-bit in the status register (SREG) is set (one), the external pin interrupt is enabled. The interrupt sense control1 bits 1/0 (ISC11 and ISC10) in the external interrupt control register A (EICRA) define whether the external interrupt is activated on rising and/or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of external interrupt request 1 is executed from the INT1 interrupt vector.

- Bit 0 – INT0: External Interrupt Request 0 Enable

When the INT0 bit is set (one) and the I-bit in the status register (SREG) is set (one), the external pin interrupt is enabled. The interrupt sense control0 bits 1/0 (ISC01 and ISC00) in the external interrupt control register A (EICRA) define whether the external interrupt is activated on rising and/or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of external interrupt request 0 is executed from the INT0 interrupt vector.

## 12.2.3 EIFR – External Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	EIFR
0x1C (0x3C)	-	-	-	-	-	-	INTF1	INTF0	
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7..2 – Res: Reserved Bits

These bits are unused bits in the Atmel ATmega328P, and will always read as zero.

- Bit 1 – INTF1: External Interrupt Flag 1

When an edge or logic change on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in EIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT1 is configured as a level interrupt.

- Bit 0 – INTF0: External Interrupt Flag 0

When an edge or logic change on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in EIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT0 is configured as a level interrupt.

#### 12.2.4 PCICR – Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
(0x68)	–	–	–	–	–	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..3 - Res: Reserved Bits**

These bits are unused bits in the Atmel® ATmega328P, and will always read as zero.

- **Bit 2 - PCIE2: Pin Change Interrupt Enable 2**

When the PCIE2 bit is set (one) and the I-bit in the status register (SREG) is set (one), pin change interrupt 2 is enabled. Any change on any enabled PCINT23..16 pin will cause an interrupt. The corresponding interrupt of pin change interrupt request is executed from the PCI2 interrupt vector. PCINT23..16 pins are enabled individually by the PCMSK2 register.

- **Bit 1 - PCIE1: Pin Change Interrupt Enable 1**

When the PCIE1 bit is set (one) and the I-bit in the status register (SREG) is set (one), pin change interrupt 1 is enabled. Any change on any enabled PCINT14..8 pin will cause an interrupt. The corresponding interrupt of pin change interrupt request is executed from the PCI1 interrupt vector. PCINT14..8 pins are enabled individually by the PCMSK1 register.

- **Bit 0 - PCIE0: Pin Change Interrupt Enable 0**

When the PCIE0 bit is set (one) and the I-bit in the status register (SREG) is set (one), pin change interrupt 0 is enabled. Any change on any enabled PCINT7..0 pin will cause an interrupt. The corresponding interrupt of pin change interrupt request is executed from the PCI0 interrupt vector. PCINT7..0 pins are enabled individually by the PCMSK0 register.

#### 12.2.5 PCIFR – Pin Change Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1B (0x3B)	–	–	–	–	–	PCIF2	PCIF1	PCIF0	PCIFR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..3 - Res: Reserved Bits**

These bits are unused bits in the Atmel ATmega328P, and will always read as zero.

- **Bit 2 - PCIF2: Pin Change Interrupt Flag 2**

When a logic change on any PCINT23..16 pin triggers an interrupt request, PCIF2 becomes set (one). If the I-bit in SREG and the PCIE2 bit in PCICR are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

- **Bit 1 - PCIF1: Pin Change Interrupt Flag 1**

When a logic change on any PCINT14..8 pin triggers an interrupt request, PCIF1 becomes set (one). If the I-bit in SREG and the PCIE1 bit in PCICR are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

- **Bit 0 - PCIF0: Pin Change Interrupt Flag 0**

When a logic change on any PCINT7..0 pin triggers an interrupt request, PCIF0 becomes set (one). If the I-bit in SREG and the PCIE0 bit in PCICR are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

### 12.2.6 PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	3	2	1	0	
(0x6D)	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	PCMSK2
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7..0 – PCINT23..16: Pin Change Enable Mask 23..16

Each PCINT23..16-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT23..16 is set and the PCIE2 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT23..16 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

### 12.2.7 PCMSK1 – Pin Change Mask Register 1

Bit	7	6	5	4	3	2	1	0	
(0x6C)	-	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R	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 – Res: Reserved Bit

This bit is an unused bit in the Atmel® ATmega328P, and will always read as zero.

- Bit 6..0 – PCINT14..8: Pin Change Enable Mask 14..8

Each PCINT14..8-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT14..8 is set and the PCIE1 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT14..8 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

### 12.2.8 PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7..0 – PCINT7..0: Pin Change Enable Mask 7..0

Each PCINT7..0 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT7..0 is set and the PCIE0 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT7..0 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

## 19.10 Register Description

### 19.10.1 UDRn – USART I/O Data Register n

Bit	7	6	5	4	3	2	1	0	
Read/Write	R/W	UDRn (Read)							
Initial Value	0	0	0	0	0	0	0	0	UDRn (Write)
RXB[7:0]									
TXB[7:0]									

The USART transmit data buffer register and USART receive data buffer registers share the same I/O address referred to as USART data register or UDRn. The transmit data buffer register (TXB) will be the destination for data written to the UDRn register location. Reading the UDRn register location will return the contents of the receive data buffer register (RXB).

For 5-, 6-, or 7-bit characters the upper unused bits will be ignored by the transmitter and set to zero by the receiver.

The transmit buffer can only be written when the UDREn flag in the UCSRnA register is set. Data written to UDRn when the UDREn flag is not set, will be ignored by the USART transmitter. When data is written to the transmit buffer, and the transmitter is enabled, the transmitter will load the data into the transmit shift register when the shift register is empty. Then the data will be serially transmitted on the TxDn pin.

The receive buffer consists of a two level FIFO. The FIFO will change its state whenever the receive buffer is accessed. Due to this behavior of the receive buffer, do not use read-modify-write instructions (SBI and CBI) on this location. Be careful when using bit test instructions (SBIC and SBIS), since these also will change the state of the FIFO.

### 19.10.2 UCSRnA – USART Control and Status Register n A

Bit	7	6	5	4	3	2	1	0	
Read/Write	RXCn	TXCn	UDREn	FEn	DORn	UPEn	U2Xn	MPCMn	UCSRnA
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 7 – RXCn: USART Receive Complete**

This flag bit is set when there are unread data in the receive buffer and cleared when the receive buffer is empty (i.e., does not contain any unread data). If the receiver is disabled, the receive buffer will be flushed and consequently the RXCn bit will become zero. The RXCn flag can be used to generate a receive complete interrupt (see description of the RXCIEn bit).

- **Bit 6 – TXCn: USART Transmit Complete**

This flag bit is set when the entire frame in the transmit shift register has been shifted out and there are no new data currently present in the transmit buffer (UDRn). The TXCn flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXCn flag can generate a transmit complete interrupt (see description of the TXCIEn bit).

- **Bit 5 – UDREn: USART Data Register Empty**

The UDREn flag indicates if the transmit buffer (UDRn) is ready to receive new data. If UDREn is one, the buffer is empty, and therefore ready to be written. The UDREn flag can generate a data register empty interrupt (see description of the UDRIEn bit). UDREn is set after a reset to indicate that the transmitter is ready.

- **Bit 4 – FEn: Frame Error**

This bit is set if the next character in the receive buffer had a frame error when received. I.e., when the first stop bit of the next character in the receive buffer is zero. This bit is valid until the receive buffer (UDRn) is read. The FEn bit is zero when the stop bit of received data is one. Always set this bit to zero when writing to UCSRnA.

- **Bit 3 – DORn: Data OverRun**

This bit is set if a data overrun condition is detected. A data overrun occurs when the receive buffer is full (two characters), it is a new character waiting in the receive shift register, and a new start bit is detected. This bit is valid until the receive buffer (UDRn) is read. Always set this bit to zero when writing to UCSRnA.

- **Bit 2 – UPEn: USART Parity Error**

This bit is set if the next character in the receive buffer had a parity error when received and the parity checking was enabled at that point ( $UPMn1 = 1$ ). This bit is valid until the receive buffer (UDRn) is read. Always set this bit to zero when writing to UCSRnA.

- **Bit 1 – U2Xn: Double the USART Transmission Speed**

This bit only has effect for the asynchronous operation. Write this bit to zero when using synchronous operation.

Writing this bit to one will reduce the divisor of the baud rate divider from 16 to 8 effectively doubling the transfer rate for asynchronous communication.

- **Bit 0 – MPCMn: Multi-processor Communication Mode**

This bit enables the multi-processor communication mode. When the MPCMn bit is written to one, all the incoming frames received by the USART receiver that do not contain address information will be ignored. The transmitter is unaffected by the MPCMn setting. For more detailed information see Section 19.9 "Multi-processor Communication Mode" on page 158.

### 19.10.3 UCSRnB – USART Control and Status Register n B

Bit	7	6	5	4	3	2	1	0	UCSRnB
	RXCIEn	TXCIEn	UDRIE <sub>n</sub>	RXEN <sub>n</sub>	TXEN <sub>n</sub>	UCSZn2	RXB8n	TXB8n	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – RXCIEn: RX Complete Interrupt Enable n**

Writing this bit to one enables interrupt on the RXCn flag. A USART receive complete interrupt will be generated only if the RXCIEn bit is written to one, the global interrupt flag in SREG is written to one and the RXCn bit in UCSRnA is set.

- **Bit 6 – TXCIEn: TX Complete Interrupt Enable n**

Writing this bit to one enables interrupt on the TXCn flag. A USART transmit complete interrupt will be generated only if the TXCIEn bit is written to one, the global interrupt flag in SREG is written to one and the TXCn bit in UCSRnA is set.

- **Bit 5 – UDRIE<sub>n</sub>: USART Data Register Empty Interrupt Enable n**

Writing this bit to one enables interrupt on the UDREN flag. A data register empty interrupt will be generated only if the UDRIE<sub>n</sub> bit is written to one, the global interrupt flag in SREG is written to one and the UDREN bit in UCSRnA is set.

- **Bit 4 – RXEN<sub>n</sub>: Receiver Enable n**

Writing this bit to one enables the USART receiver. The receiver will override normal port operation for the RxD<sub>n</sub> pin when enabled. Disabling the receiver will flush the receive buffer invalidating the FEn, DORn, and UPEn flags.

- **Bit 3 – TXEN<sub>n</sub>: Transmitter Enable n**

Writing this bit to one enables the USART transmitter. The transmitter will override normal port operation for the TxD<sub>n</sub> pin when enabled. The disabling of the transmitter (writing TXEN<sub>n</sub> to zero) will not become effective until ongoing and pending transmissions are completed, i.e., when the transmit shift register and transmit buffer register do not contain data to be transmitted. When disabled, the transmitter will no longer override the TxD<sub>n</sub> port.

- **Bit 2 – UCSZn2: Character Size n**

The UCSZn2 bits combined with the UCSZn1:0 bit in UCSRnC sets the number of data bits (character size) in a frame the receiver and transmitter use.

- **Bit 1 – RXB8n: Receive Data Bit 8 n**

RXB8n is the ninth data bit of the received character when operating with serial frames with nine data bits. Must be read before reading the low bits from UDRn.

- **Bit 0 – TXB8n: Transmit Data Bit 8 n**

TXB8n is the ninth data bit in the character to be transmitted when operating with serial frames with nine data bits. Must be written before writing the low bits to UDRn.

#### 19.10.4 UCSRnC – USART Control and Status Register n C

Bit	7	6	5	4	3	2	1	0	UCSRnC
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	
ReadWrite	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

- Bits 7:6 – UMSELn1:0 USART Mode Select

These bits select the mode of operation of the USARTn as shown in Table 19-4.

Table 19-4. UMSELn Bits Settings

UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM)

Note: 1. See Section 20, "USART in SPI Mode" on page 166 for full description of the master SPI mode (MSPIM) operation

- Bits 5:4 – UPMn1:0: Parity Mode

These bits enable and set type of parity generation and check. If enabled, the transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The receiver will generate a parity value for the incoming data and compare it to the UPMn setting. If a mismatch is detected, the UPEn flag in UCSRnA will be set.

Table 19-5. UPMn Bits Settings

UPMn1	UPMn0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, even parity
1	1	Enabled, odd parity

- Bit 3 – USBSn: Stop Bit Select

This bit selects the number of stop bits to be inserted by the transmitter. The receiver ignores this setting.

Table 19-6. USBS Bit Settings

USBSn	Stop Bit(s)
0	1-bit
1	2-bit

- Bit 2:1 – UCSZn1:0: Character Size

The UCSZn1:0 bits combined with the UCSZn2 bit in UCSRnB sets the number of data bits (character size) in a frame the receiver and transmitter use.

**Table 19-7. UCSZn Bits Settings**

UCSZn2	UCSZn1	UCSZn0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
			9-bit

- Bit 0 – UCPOLn: Clock Polarity

This bit is used for synchronous mode only. Write this bit to zero when asynchronous mode is used. The UCPOLn bit sets the relationship between data output change and data input sample, and the synchronous clock (XCKn).

**Table 19-8. UCPOLn Bit Settings**

UCPOLn	Transmitted Data Changed (Output of TxDn Pin)	Received Data Sampled (Input on RxDn Pin)
0	Rising XCKn edge	Falling XCKn edge
1	Falling XCKn edge	Rising XCKn edge

### 19.10.5 UBRRnL and UBRRnH – USART Baud Rate Registers

Bit	15	14	13	12	11	10	9	8	UBRRnH	UBRRnL
	–	–	–	–	UBRRn[11:8]					
	UBRRn[7:0]									
	7	6	5	4	3	2	1	0		
Read/Write	R	R	R	R	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		

- Bit 15:12 – Reserved Bits

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

- Bit 11:0 – UBRR11:0: USART Baud Rate Register

This is a 12-bit register which contains the USART baud rate. The UBRRnH contains the four most significant bits, and the UBRRnL contains the eight least significant bits of the USART baud rate. Ongoing transmissions by the transmitter and receiver will be corrupted if the baud rate is changed. Writing UBRRnL will trigger an immediate update of the baud rate prescaler.

## 23.9 Register Description

### 23.9.1 ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7:6 – REFS1:0: Reference Selection Bits

These bits select the voltage reference for the ADC, as shown in Table 23-3. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set). The internal voltage reference options may not be used if an external reference voltage is being applied to the AREF pin.

Table 23-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, internal $V_{REF}$ turned off
0	1	AVDD with external capacitor to AREF pin
1	0	Reserved
		Internal 1.1V voltage reference with external capacitor to AREF pin

- 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 Section 23.9.3 "ADCL and ADCH – The ADC Data Register" on page 219.

- Bit 4 – Res: Reserved Bit

This bit is an unused bit in the Atmel® ATmega328P, and will always read as zero.

- Bits 3:0 – MUX3:0: Analog Channel Selection Bits

The value of these bits selects which analog inputs are connected to the ADC. See Table 23-4 on page 218 for details. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set).

Table 23-4. Input Channel Selections

MUX3..0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8 <sup>(1)</sup>
1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1V ( $V_{BG}$ )
1111	0V (GND)

Note: 1. For temperature sensor.

### 23.9.2 ADCSRA – ADC Control and Status Register A

Bit (0x7A)	7	6	5	4	3	2	1	0	ADCSRA
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ADEN: ADC Enable**

Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress, will terminate this conversion.

- **Bit 6 – ADSC: ADC Start Conversion**

In single conversion mode, write this bit to one to start each conversion. In free running mode, write this bit to one to start the first conversion. The first conversion after ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, will take 25 ADC clock cycles instead of the normal 13. This first conversion performs initialization of the ADC.

ADSC will read as one as long as a conversion is in progress. When the conversion is complete, it returns to zero. Writing zero to this bit has no effect.

- **Bit 5 – ADATE: ADC Auto Trigger Enable**

When this bit is written to one, auto triggering of the ADC is enabled. The ADC will start a conversion on a positive edge of the selected trigger signal. The trigger source is selected by setting the ADC trigger select bits, ADTS in ADCSRB.

- **Bit 4 – ADIF: ADC Interrupt Flag**

This bit is set when an ADC conversion completes and the data registers are updated. The ADC conversion complete interrupt is executed if the ADIE bit and the I-bit in SREG are set. ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical one to the flag. Beware that if doing a read-modify-write on ADCSRA, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

- Bit 3 – ADIE: ADC Interrupt Enable

When this bit is written to one and the I-bit in SREG is set, the ADC conversion complete interrupt is activated.

- Bits 2:0 – ADPS2:0: ADC Prescaler Select Bits

These bits determine the division factor between the system clock frequency and the input clock to the ADC.

**Table 23-5. ADC Prescaler Selections**

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	4
0	1	0	8
1	0	0	16
1	0	1	32
1	1	0	64
			128

### 23.9.3 ADCL and ADCH – The ADC Data Register

#### 23.9.3.1 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	
ReadWrite	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	

#### 23.9.3.2 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	
ReadWrite	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.

- ADC9:0: ADC Conversion Result

These bits represent the result from the conversion, as detailed in Section 23.7 "ADC Conversion Result" on page 215.

### 23.9.4 ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
(0x7B)	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	ADCSR <sub>B</sub>
ReadWrite	R	R/W	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 5:3 – Res: Reserved Bits**

These bits are reserved for future use. To ensure compatibility with future devices, these bits must be written to zero when ADCSRB is written.

- **Bit 2:0 – ADTS2:0: ADC Auto Trigger Source**

If ADATE in ADCSRA is written to one, the value of these bits selects which source will trigger an ADC conversion. If ADATE is cleared, the ADTS2:0 settings will have no effect. A conversion will be triggered by the rising edge of the selected interrupt flag. Note that switching from a trigger source that is cleared to a trigger source that is set, will generate a positive edge on the trigger signal. If ADEN in ADCSRA is set, this will start a conversion. Switching to free running mode (ADTS[2:0]=0) will not cause a trigger event, even if the ADC interrupt flag is set.

Table 23-6. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free running mode
0	0	1	Analog Comparator
0	1	0	External interrupt request 0
0	1	1	Timer/Counter0 compare match A
1	0	0	Timer/Counter0 overflow
1	0	1	Timer/Counter1 compare match B
1	1	0	Timer/Counter1 overflow
1	1	1	Timer/Counter1 capture event

### 23.9.5 DIDR0 – Digital Input Disable Register 0

Bit	7	6	5	4	3	2	1	0	
(0x7E)	–	–	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	DIDR <sub>0</sub>
ReadWrite	R	R	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 – Res: Reserved Bits**

These bits are reserved for future use. To ensure compatibility with future devices, these bits must be written to zero when DIDR0 is written.

- **Bit 5:0 – ADC5D..ADC0D: ADC5..0 Digital Input Disable**

When this bit is written logic one, the digital input buffer on the corresponding ADC pin is disabled. The corresponding PIN register bit will always read as zero when this bit is set. When an analog signal is applied to the ADC5..0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

Note that ADC pins ADC7 and ADC6 do not have digital input buffers, and therefore do not require digital input disable bits.

# Appendix A: Summary of Atmel AVR Instruction Set

Arithmetic and Logic Instructions					
ADD	Rd, Rr	$Rd \leftarrow Rd + Rr$	Z,C,N,V,S,H	1	
ADC	Rd, Rr	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,S,H	1	
ADIW <sup>(1)</sup>	Rd, K	$R_{d+1}:R_d \leftarrow R_{d+1}:R_d + k$	Z,C,N,V,S	2	
SUB	Rd, Rr	$Rd \leftarrow Rd - Rr$	Z,C,N,V,S,H	1	
SUBI	Rd, K	$Rd \leftarrow Rd - K$	Z,C,N,V,S,H	1	
SBC	Rd, Rr	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,S,H	1	
SBCI	Rd, K	$Rd \leftarrow Rd - K - C$	Z,C,N,V,S,H	1	
SBIW <sup>(1)</sup>	Rd, K	$R_{d+1}:R_d \leftarrow R_{d+1}:R_d - K$	Z,C,N,V,S	2	
AND	Rd, Rr	$Rd \leftarrow Rd \& Rr$	Z,N,V,S	1	
ANDI	Rd, K	$Rd \leftarrow Rd \& K$	Z,N,V,S	1	
OR	Rd, Rr	$Rd \leftarrow Rd   Rr$	Z,N,V,S	1	
ORI	Rd, K	$Rd \leftarrow Rd   K$	Z,N,V,S	1	
EOR	Rd, Rr	$Rd \leftarrow Rd \oplus Rr$	Z,N,V,S	1	
COM	Rd	$Rd \leftarrow 0 \times FF - Rd$	Z,C,N,V,S	1	
NEG	Rd	$Rd \leftarrow 0 \times 00 - Rd$	Z,C,N,V,S,H	1	
SBR	Rd, K	$Rd \leftarrow Rd   K$	Z,N,V,S	1	
CBR	Rd, K	$Rd \leftarrow Rd \& (0 \times FF - K)$	Z,N,V,S	1	
INC	Rd	$Rd \leftarrow Rd + 1$	Z,N,V,S	1	
DEC	Rd	$Rd \leftarrow Rd - 1$	Z,N,V,S	1	
TST	Rd	$Rd \leftarrow Rd \& Rd$	Z,N,V,S	1	
CLR	Rd	$Rd \leftarrow Rd \oplus Rd$	Z,N,V,S	1	
SER	Rd	$Rd \leftarrow 0 \times FF$	None	1	
MUL <sup>(1)</sup>	Rd, Rr	$R1:R0 \leftarrow Rd \times Rr (UU)$	Z,C	2	
MULS <sup>(1)</sup>	Rd, Rr	$R1:R0 \leftarrow Rd \times Rr (SS)$	Z,C	2	
MULSU <sup>(1)</sup>	Rd, Rr	$R1:R0 \leftarrow Rd \times Rr (SU)$	Z,C	2	
FMUL <sup>(1)</sup>	Rd, Rr	$R1:R0 \leftarrow Rd \times Rr << 1 (UU)$	Z,C	2	
FMULS <sup>(1)</sup>	Rd, Rr	$R1:R0 \leftarrow Rd \times Rr << 1 (SS)$	Z,C	2	
FMULSU <sup>(1)</sup>	Rd, Rr	$R1:R0 \leftarrow Rd \times Rr << 1 (SU)$	Z,C	2	
DES	K	if (H = 0) then R15:R0 $\leftarrow$ Encrypt (R15:R0, K) else R15:R0 Decrypt (R15:R0, K)			1/2
Branch Instructions					
RJMP	K	$PC \leftarrow PC + k + 1$	none	2	
IJMP <sup>(1)</sup>		$PC(15:0) \leftarrow Z, PC(21:16) \leftarrow 0$	none	2	
EIJMP <sup>(1)</sup>		$PC(15:0) \leftarrow Z, PC(21:16) \leftarrow EIND$	none	2	
JMP <sup>(1)</sup>	K	$PC \leftarrow k$	none	3	

Table A.1 ■ AVR Instruction Set Summary

(Continued)

**Table A.1** ■ (Continued)

RCALL	K	$PC \leftarrow PC + k + 1$	none	3/4 <sup>(3)(5)</sup>	2/3 <sup>(3)</sup>
ICALL <sup>(1)</sup>		$PC(15:0) \leftarrow Z, PC(21:16) \leftarrow 0$	none	3/4 <sup>(3)(5)</sup>	2/3 <sup>(3)</sup>
EICALL <sup>(1)</sup>		$PC(15:0) \leftarrow Z, PC(21:16) \leftarrow EIND$	none	4 <sup>(3)</sup>	3 <sup>(3)</sup>
CALL <sup>(1)</sup>	K	$PC \leftarrow k$	none	4/5 <sup>(3)</sup>	3/4 <sup>(3)</sup>
RET		$PC \leftarrow STACK$	none	4/5 <sup>(3)</sup>	
RETI		$PC \leftarrow STACK$	I	4/5 <sup>(3)</sup>	
CPSE	Rd, Rr	if ( $Rd = Rr$ ) $PC \leftarrow PC + 2$ or 3	none	1/2/3	
CP	Rd, Rr	$Rd = Rr$	Z,C,N,V,S,H	1	
CPC	Rd, Rr	$Rd = Rr - C$	Z,C,N,V,S,H	1	
CPI	Rd, K	$Rd = K$	Z,C,N,V,S,H	1	
SBRC	Rr, b	if ( $Rr(b) = 0$ ) then $PC \leftarrow PC + 2$ or 3	none	1/2/3	
SBRS	Rr, b	if ( $Rr(b) = 1$ ) then $PC \leftarrow PC + 2$ or 3	none	1/2/3	
SBIC	A, b	if ( $I/O(A, b) = 0$ ) then $PC \leftarrow PC + 2$ or 3	none	1/2/3	2/3/4
SBIS	A, b	if ( $I/O(A, b) = 1$ ) then $PC \leftarrow PC + 2$ or 3	none	1/2/3	2/3/4
BRBS	s, k	if ( $SREG(s) = 1$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRBC	s, k	if ( $SREG(s) = 0$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BREQ	K	if ( $Z = 1$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRNE	k	if ( $Z = 0$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRCS	k	if ( $C = 1$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRCC	k	if ( $C = 0$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRSH	k	if ( $C = 0$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRLO	k	if ( $C = 1$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRMI	k	if ( $N = 1$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRPL	k	if ( $N = 0$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRGE	k	if ( $N \oplus V = 0$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRLT	k	if ( $N \oplus V = 1$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRHS	k	if ( $H = 1$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRHC	k	if ( $H = 0$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRTS	k	if ( $T = 1$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRTC	k	if ( $T = 0$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRVS	k	if ( $V = 1$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRVC	k	if ( $V = 0$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRIE	k	if ( $I = 1$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
BRID	k	if ( $I = D$ ) then $PC \leftarrow PC + k + 1$	none	1/2	
Data Transfer Instructions					
MOV	Rd, Rr	$Rd \leftarrow Rd$	none	1	
MOVW	Rd, Rr	$R_{d+1}:R_d \leftarrow R_{r+1}:R_r$	none	1	
LDI	Rd, K	$Rd \leftarrow K$	none	1	
LDS <sup>(1)</sup>	Rd, k	$Rd \leftarrow (k)$	none	1 <sup>(5)</sup> 2 <sup>(3)</sup>	2 <sup>(3)(4)</sup>
LD <sup>(2)</sup>	Rd, X	$Rd \leftarrow (X)$	none	1 <sup>(5)</sup> 2 <sup>(3)</sup>	1 <sup>(3)(4)</sup>
LD <sup>(2)</sup>	Rd, X+	$Rd \leftarrow (X), X \leftarrow X + 1$	none	2 <sup>(3)</sup>	1 <sup>(3)(4)</sup>

**Table A.1** ■ (Continued)

LD <sup>(2)</sup>	Rd, -X	X ← X − 1, Rd ← (X)	none	2 <sup>(3)</sup> 3 <sup>(5)</sup>	2 <sup>(3)(4)</sup>
LD <sup>(2)</sup>	Rd, Y	Rd ← (Y)	none	1 <sup>(5)</sup> 2 <sup>(3)</sup>	1 <sup>(3)(4)</sup>
LD <sup>(2)</sup>	Rd, Y+	Rd ← (Y), Y ← Y + 1	none	2 <sup>(3)</sup>	1 <sup>(3)(4)</sup>
LD <sup>(2)</sup>	Rd, -Y	Y ← Y − 1, Rd ← (Y)	none	2 <sup>(3)</sup> 3 <sup>(5)</sup>	2 <sup>(3)(4)</sup>
LDD <sup>(1)</sup>	Rd, Y + q	Rd ← (Y + q)	none	2 <sup>(3)</sup>	2 <sup>(3)(4)</sup>
LD <sup>(2)</sup>	Rd, Z	Rd ← (Z)	none	1 <sup>(5)</sup> 2 <sup>(3)</sup>	1 <sup>(3)(4)</sup>
LD <sup>(2)</sup>	Rd, Z+	Rd ← (Z), Z ← Z + 1	none	2 <sup>(3)</sup>	1 <sup>(3)(4)</sup>
LD <sup>(2)</sup>	Rd, -Z	Z ← Z − 1, Rd ← (Z)	none	2 <sup>(3)</sup> 3 <sup>(5)</sup>	2 <sup>(3)(4)</sup>
LDD <sup>(1)</sup>	Rd, Z + q	Rd ← (Z + q)	none	2 <sup>(3)</sup>	2 <sup>(3)(4)</sup>
STS <sup>(1)</sup>	k, Rr	(k) ← Rr	none	1 <sup>(5)</sup> 2 <sup>(3)</sup>	2 <sup>(3)</sup>
ST <sup>(2)</sup>	X, Rr	(X) ← Rr	none	1 <sup>(5)</sup> 2 <sup>(3)</sup>	1 <sup>(3)</sup>
ST <sup>(2)</sup>	X+, Rr	(X) ← Rr, X ← X + 1	none	1 <sup>(5)</sup> 2 <sup>(3)</sup>	1 <sup>(3)</sup>
ST <sup>(2)</sup>	-X, Rr	X ← X − 1, (X) ← Rr	none	2 <sup>(3)</sup>	2 <sup>(3)</sup>
ST <sup>(2)</sup>	Y, Rr	(Y) ← Rr	none	1 <sup>(5)</sup> 2 <sup>(3)</sup>	1 <sup>(3)</sup>
ST <sup>(2)</sup>	Y+, Rr	(Y) ← Rr, (Y) ← Y + 1	none	1 <sup>(5)</sup> 2 <sup>(3)</sup>	1 <sup>(3)</sup>
ST <sup>(2)</sup>	-Y, Rr	Y ← Y − 1, (Y) ← Rr	none	2 <sup>(3)</sup>	2 <sup>(3)</sup>
STD <sup>(1)</sup>	Y + q, Rr	(Y + q) ← Rr	none	2 <sup>(3)</sup>	2 <sup>(3)</sup>
ST <sup>(2)</sup>	Z, Rr	(Z) ← Rr	none	1 <sup>(5)</sup> 2 <sup>(3)</sup>	1 <sup>(3)</sup>
ST <sup>(2)</sup>	Z+, Rr	(Z) ← Rr, Z ← Z + 1	none	1 <sup>(5)</sup> 2 <sup>(3)</sup>	1 <sup>(3)</sup>
ST <sup>(2)</sup>	-Z, Rr	Z ← Z − 1, (Z) ← Rr	none	2 <sup>(3)</sup>	2 <sup>(3)</sup>
STD <sup>(1)</sup>	Z + q, Rr	(Z + q) ← Rr	none	2 <sup>(3)</sup>	2 <sup>(3)</sup>
LPM <sup>(1)(2)</sup>		R0 ← (Z); load program memory	none	3	3
LPM <sup>(1)(2)</sup>	Rd, Z	Rd ← (Z); load program memory	none	3	3
LPM <sup>(1)(2)</sup>	Rd, Z+	Rd ← (Z), Z ← Z + 1; load program memory	none	3	3
ELPM <sup>(1)</sup>		R0 ← (RAMPZ:Z); load program memory	none	3	
ELPM <sup>(1)</sup>	Rd, Z	Rd ← (RAMPZ:Z); load program memory	none	3	
ELPM <sup>(1)</sup>	Rd, Z+	Rd ← (RAMPZ:Z), Z ← Z + 1	none	3	
SPM <sup>(1)</sup>		(RAMPZ:Z) ← R1:R0; store program mem.	none	—	—
SPM <sup>(1)</sup>	Z+	(RAMPZ:Z) ← R1:R0, Z ← Z + 2	none	—	—
IN	Rd, A	Rd ← I/O(A)	none	1	
OUT	A, Rr	I/O(A) ← Rr	none	1	
PUSH <sup>(1)</sup>	Rr	STACK ← Rr	none	2	1 <sup>(3)</sup>
POP <sup>(1)</sup>	Rd	Rd ← STACK	none	2	2 <sup>(3)</sup>
XCH	Z, Rd	(Z) ← Rd, Rd ← (Z)	none	1	
LAS	Z, Rd	(Z) ← Rd   (Z), Rd ← (Z)	none	1	
LAC	Z, Rd	(Z) ← (0 × FF − Rd) & (Z), Rd ← (Z)	none	1	
LAT	Z, Rd	(Z) ← Rd ⊕ (Z), Rd ← (Z)	none	1	
<b>Bit and Bit-test Instructions</b>					
LSL	Rd	Rd(n + 1) ← Rd(n), Rd(0) ← 0, C ← Rd(7)	Z,C,N,V,H	1	
LSR	Rd	Rd(n) ← Rd(n + 1), Rd(7) ← 0, C ← Rd(0)	Z,C,N,V	1	
ROL	Rd	Rd(0) ← C, Rd(n + 1) ← Rd(n), C ← Rd(7)	Z,C,N,V,H	1	

(Continued)

**Table A.1** ■ (Continued)

ROR	Rd	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z,C,N,V	1	
ASR	Rd	$Rd(n) \leftarrow Rd(n+1), n=0, \dots, 6, Rd(7) \leftarrow Rd(7)$	Z,C,N,V	1	
SWAP	Rd	$Rd(3..0) \leftrightarrow Rd(7..4)$	none	1	
BSET	s	$SREG(s) \leftarrow 1$	SREG(s)	1	
BCLR	s	$SREG(s) \leftarrow 0$	SREG(s)	1	
SBI	A, b	$I/O(A,b) \leftarrow 1$	none	<sup>1</sup> / <sub>5</sub> / <sup>2</sup>	1
CBI	A, b	$I/O(A,b) \leftarrow 0$	none	<sup>1</sup> / <sub>5</sub> / <sup>2</sup>	1
BST	Rr, b	$T \leftarrow Rr(b)$	T	1	
BLD	Rd, b	$Rd(b) \leftarrow T$	none	1	
SEC		$C \leftarrow 1$	C	1	
CLC		$C \leftarrow 0$	C	1	
SEN		$N \leftarrow 1$	N	1	
CLN		$N \leftarrow 0$	N	1	
SEZ		$Z \leftarrow 1$	Z	1	
CLZ		$Z \leftarrow 0$	Z	1	
SEI		$I \leftarrow 1$	I	1	
CLI		$I \leftarrow 0$	I	1	
SES		$S \leftarrow 1$	S	1	
CLS		$S \leftarrow 0$	S	1	
SEV		$V \leftarrow 1$	V	1	
CLV		$V \leftarrow 0$	V	1	
SET		$T \leftarrow 1$	T	1	
CLT		$T \leftarrow 0$	T	1	
SEH		$H \leftarrow 1$	H	1	
CLH		$H \leftarrow 0$	H	1	
<b>MCU Control Instructions</b>					
BREAK		See specific description for BREAK	none	1	
NOP			none	1	
SLEEP		See specific description about SLEEP	none	1	
WDR		See specific description about WDR	none	1	

- Note:
1. This instruction is not available in all devices. Refer to the device specific instruction set summary.
  2. Not all variants of this instruction are available in all devices. Refer to the device specific set summary.
  3. Cycle times for data memory accesses assume internal memory accesses, and are not valid for accesses via the external RAM interface.
  4. One extra cycle must be added when accessing internal SRAM.
  5. Number of clock cycles for reduced core tinyAVR.