

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	10MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	23
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-VQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega48v-10mu

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

4.5.1 The X-register, Y-register, and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the data space. The three indirect address registers X, Y, and Z are defined as described in Figure 4-3.



Figure 4-3. The X-, Y-, and Z-registers

In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the instruction set reference for details).

4.6 Stack Pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The Stack Pointer Register always points to the top of the Stack. Note that the Stack is implemented as growing from higher memory locations to lower memory locations. This implies that a Stack PUSH command decreases the Stack Pointer.

The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above 0x0100, preferably RAMEND. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the Stack with subroutine call or interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
l	7	6	5	4	3	2	1	0	
Read/Write	R/W								
	R/W								
Initial Value	RAMEND								
	RAMEND								





4.7 Instruction Execution Timing

This section describes the general access timing concepts for instruction execution. The AVR CPU is driven by the CPU clock clk_{CPU} , directly generated from the selected clock source for the chip. No internal clock division is used.

Figure 4-4 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.



Figure 4-4. The Parallel Instruction Fetches and Instruction Executions

Figure 4-5 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.





4.8 Reset and Interrupt Handling

The AVR provides several different interrupt sources. These interrupts and the separate Reset Vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section "Memory Programming" on page 280 for details.

The lowest addresses in the program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in "Interrupts" on page 54. The list also determines the priority levels of the different interrupts. The lower the address the higher is the

12 ATmega48/88/168



5.4 I/O Memory

The I/O space definition of the ATmega48/88/168 is shown in "Register Summary" on page 334.

All ATmega48/88/168 I/Os and peripherals are placed in the I/O space. All I/O locations may be accessed by the LD/LDS/LDD and ST/STS/STD instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set section for more details. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega48/88/168 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

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

Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

The I/O and peripherals control registers are explained in later sections.

5.4.1 General Purpose I/O Registers

The ATmega48/88/168 contains three General Purpose I/O Registers. These registers can be used for storing any information, and they are particularly useful for storing global variables and Status Flags. General Purpose I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI, CBI, SBIS, and SBIC instructions.

5.4.2 General Purpose I/O Register 2 – GPIOR2



5.4.3 General Purpose I/O Register 1 – GPIOR1



5.4.4 General Purpose I/O Register 0 – GPIOR0

Bit	7	6	5	4	3	2	1	0	_
	MSB							LSB	GPIOR0
Read/Write	R/W	-							
Initial Value	0	0	0	0	0	0	0	0	



CLKPS3	CLKPS2	CLKPS1	CLKPS0	Clock Division Factor
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved





8.0.6 Watchdog System Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT} . Refer to page 49 for details on operation of the Watchdog Timer.





8.0.7 MCU Status Register – MCUSR

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



• Bit 7..4: Res: Reserved Bits

These bits are unused bits in the ATmega48/88/168, and will always read as zero.

Bit 3 – WDRF: Watchdog System Reset Flag

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

Bit 2 – BORF: Brown-out Reset Flag

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





8.2.1 Watchdog Timer Control Register - WDTCSR

Bit	7	6	5	4	3	2	1	0	_
	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	WDTCSR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
Initial Value	0	0	0	0	Х	0	0	0	

• Bit 7 - WDIF: Watchdog Interrupt Flag

This bit is set when a time-out occurs in the Watchdog Timer and the Watchdog Timer is configured for interrupt. WDIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, WDIF is cleared by writing a logic one to the flag. When the I-bit in SREG and WDIE are set, the Watchdog Time-out Interrupt is executed.

Bit 6 - WDIE: Watchdog Interrupt Enable

When this bit is written to one and the I-bit in the Status Register is set, the Watchdog Interrupt is enabled. If WDE is cleared in combination with this setting, the Watchdog Timer is in Interrupt Mode, and the corresponding interrupt is executed if time-out in the Watchdog Timer occurs.

If WDE is set, the Watchdog Timer is in Interrupt and System Reset Mode. The first time-out in the Watchdog Timer will set WDIF. Executing the corresponding interrupt vector will clear WDIE and WDIF automatically by hardware (the Watchdog goes to System Reset Mode). This is useful for keeping the Watchdog Timer security while using the interrupt. To stay in Interrupt and System Reset Mode, WDIE must be set after each interrupt. This should however not be done within the interrupt service routine itself, as this might compromise the safety-function of the Watchdog System Reset mode. If the interrupt is not executed before the next time-out, a System Reset will be applied.

WDTON	WDE	WDIE Mode		Action on Time-out
0	0	0 Stopped		None
0	0	1	Interrupt Mode	Interrupt
0	1	0	System Reset Mode	Reset
0	1	1	Interrupt and System Reset Mode	Interrupt, then go to System Reset Mode
1	х	х	System Reset Mode	Reset

Table 8-5.Watchdog Timer Configuration

• Bit 4 - WDCE: Watchdog Change Enable

This bit is used in timed sequences for changing WDE and prescaler bits. To clear the WDE bit, and/or change the prescaler bits, WDCE must be set.

Once written to one, hardware will clear WDCE after four clock cycles.

Bit 3 - WDE: Watchdog System Reset Enable

WDE is overridden by WDRF in MCUSR. This means that WDE is always set when WDRF is set. To clear WDE, WDRF must be cleared first. This feature ensures multiple resets during conditions causing failure, and a safe start-up after the failure.

 Table 9-1.
 Reset and Interrupt Vectors in ATmega48 (Continued)

Vector No.	Program Address	Source	Interrupt Definition
24	0x017	ANALOG COMP	Analog Comparator
25	0x018	TWI	2-wire Serial Interface
26	0x019	SPM READY	Store Program Memory Ready

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega48 is:

Address	Labels Code		С	omments
0x000	rjmp	RESET	;	Reset Handler
0x001	rjmp	EXT_INT0	;	IRQ0 Handler
0x002	rjmp	EXT_INT1	;	IRQ1 Handler
0x003	rjmp	PCINT0	;	PCINTO Handler
0x004	rjmp	PCINT1	;	PCINT1 Handler
0x005	rjmp	PCINT2	;	PCINT2 Handler
0x006	rjmp	WDT	;	Watchdog Timer Handler
0x007	rjmp	TIM2_COMPA	;	Timer2 Compare A Handler
0x008	rjmp	TIM2_COMPB	;	Timer2 Compare B Handler
0x009	rjmp	TIM2_OVF	;	Timer2 Overflow Handler
0x00A	rjmp	TIM1_CAPT	;	Timer1 Capture Handler
0x00B	rjmp	TIM1_COMPA	;	Timer1 Compare A Handler
0x00C	rjmp	TIM1_COMPB	;	Timer1 Compare B Handler
0x00D	rjmp	TIM1_OVF	;	Timer1 Overflow Handler
0x00E	rjmp	TIM0_COMPA	;	Timer0 Compare A Handler
0x00F	rjmp	TIM0_COMPB	;	Timer0 Compare B Handler
0x010	rjmp	TIM0_OVF	;	Timer0 Overflow Handler
0x011	rjmp	SPI_STC	;	SPI Transfer Complete Handler
0x012	rjmp	USART_RXC	;	USART, RX Complete Handler
0x013	rjmp	USART_UDRE	;	USART, UDR Empty Handler
0x014	rjmp	USART_TXC	;	USART, TX Complete Handler
0x015	rjmp	ADC	;	ADC Conversion Complete Handler
0x016	rjmp	EE_RDY	;	EEPROM Ready Handler
0x017	rjmp	ANA_COMP	;	Analog Comparator Handler
0x018	rjmp	TWI	;	2-wire Serial Interface Handler
0x019	rjmp	SPM_RDY	;	Store Program Memory Ready Handler
;				
0x01ARES	ET: ldi	r16, high(RAME	ND)); Main program start
0x01B	out	SPH,r16	;	Set Stack Pointer to top of RAM
0x01C	ldi	r16, low(RAMENI	D)	
0x01D	out	SPL,r16		
0x01E	sei		;	Enable interrupts
0x01F	<instr< td=""><td>> xxx</td><td></td><td></td></instr<>	> xxx		



		•	-
BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x001
1	1	0x000	Boot Reset Address + 0x001
0	0	Boot Reset Address	0x001
0	1	Boot Reset Address	Boot Reset Address + 0x001

 Table 9-3.
 Reset and Interrupt Vectors Placement in ATmega88⁽¹⁾

Note: 1. The Boot Reset Address is shown in Table 24-6 on page 276. 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 ATmega88 is:

Address	Labels Code		Co	omments
0x000	rjmp	RESET	;	Reset Handler
0x001	rjmp	EXT_INT0	;	IRQ0 Handler
0x002	rjmp	EXT_INT1	;	IRQ1 Handler
0x003	rjmp	PCINT0	;	PCINTO Handler
0x004	rjmp	PCINT1	;	PCINT1 Handler
0x005	rjmp	PCINT2	;	PCINT2 Handler
0x006	rjmp	WDT	;	Watchdog Timer Handler
0x007	rjmp	TIM2_COMPA	;	Timer2 Compare A Handler
0X008	rjmp	TIM2_COMPB	;	Timer2 Compare B Handler
0x009	rjmp	TIM2_OVF	;	Timer2 Overflow Handler
0x00A	rjmp	TIM1_CAPT	;	Timer1 Capture Handler
0x00B	rjmp	TIM1_COMPA	;	Timer1 Compare A Handler
0x00C	rjmp	TIM1_COMPB	;	Timer1 Compare B Handler
0x00D	rjmp	TIM1_OVF	;	Timer1 Overflow Handler
0x00E	rjmp	TIM0_COMPA	;	Timer0 Compare A Handler
0x00F	rjmp	TIM0_COMPB	;	Timer0 Compare B Handler
0x010	rjmp	TIM0_OVF	;	Timer0 Overflow Handler
0x011	rjmp	SPI_STC	;	SPI Transfer Complete Handler
0x012	rjmp	USART_RXC	;	USART, RX Complete Handler
0x013	rjmp	USART_UDRE	;	USART, UDR Empty Handler
0x014	rjmp	USART_TXC	;	USART, TX Complete Handler
0x015	rjmp	ADC	;	ADC Conversion Complete Handler
0x016	rjmp	EE_RDY	;	EEPROM Ready Handler
0x017	rjmp	ANA_COMP	;	Analog Comparator Handler
0x018	rjmp	TWI	;	2-wire Serial Interface Handler
0x019	rjmp	SPM_RDY	;	Store Program Memory Ready Handler
;				
0x01ARES	ET: ldi	r16, high(RAME	1D); Main program start
0x01B	out	SPH,r16	;	Set Stack Pointer to top of RAM
0x01C	ldi	r16, low(RAMENI	D)	
0x01D	out	SPL,r16		
0x01E	sei		;	Enable interrupts
0x01F	<instr< td=""><td>> xxx</td><td></td><td></td></instr<>	> xxx		



11. 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 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 "Clock Systems and their Distribution" on page 25. 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 "System Clock and Clock Options" on page 25.

11.1 Pin Change Interrupt Timing

An example of timing of a pin change interrupt is shown in Figure 11-1.





Table 12-7 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

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.			

 Table 12-7.
 Compare Output Mode, Phase Correct PWM Mode⁽¹⁾

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 "Phase Correct PWM Mode" on page 96 for more details.

• Bits 3, 2 - Res: Reserved Bits

These bits are reserved bits in the ATmega48/88/168 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 12-8. 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 "Modes of Operation" on page 93).

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	ТОР	Update of OCRx at	TOV Flag Set on ⁽¹⁾⁽²⁾	
0	0	0	0	Normal	0xFF	Immediate	MAX	
1	0	0	1	PWM, Phase Correct	0xFF	TOP	воттом	
2	0	1	0	СТС	OCRA	Immediate	MAX	
3	0	1	1	Fast PWM	0xFF	TOP	MAX	
4	1	0	0	Reserved	_	_	_	
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM	
6	1	1	0	Reserved	_	_	_	
7	1	1	1	Fast PWM	OCRA	TOP	TOP	

 Table 12-8.
 Waveform Generation Mode Bit Description

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



ATmega48/88/168



Figure 13-13. Timer/Counter Timing Diagram, with Prescaler (${\rm f}_{\rm clk_I/O}/8)$



13.10.7 Input Capture Register 1 – ICR1H and ICR1L



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 "13.2" on page 108.

13.10.8 Timer/Counter1 Interrupt Mask Register – TIMSK1



• Bit 7, 6 - Res: Reserved Bits

These bits are unused bits in the ATmega48/88/168, 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 "Interrupts" on page 54) is executed when the ICF1 Flag, located in TIFR1, is set.

• Bit 4, 3 - Res: Reserved Bits

These bits are unused bits in the ATmega48/88/168, 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 "Interrupts" on page 54) 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 "Interrupts" on page 54) 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 "8.2" on page 49.) is executed when the TOV1 Flag, located in TIFR1, is set.





Enabling and disabling of the clock input must be done when T1/T0 has been stable for at least one system clock cycle, otherwise it is a risk that a false Timer/Counter clock pulse is generated.

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

An external clock source can not be prescaled.



Figure 14-2. Prescaler for Timer/Counter0 and Timer/Counter1⁽¹⁾

Note: 1. The synchronization logic on the input pins (T1/T0) is shown in Figure 14-1.

	f _{osc} = 16.0000 MHz				f _{osc} = 18.4320 MHz				f _{osc} = 20.0000 MHz			
Baud	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
Rate (bps)	UBRR n	Error	UBRR n	Error	UBRR n	Error	UBRR n	Error	UBRR n	Error	UBRR n	Error
2400	416	-0.1%	832	0.0%	479	0.0%	959	0.0%	520	0.0%	1041	0.0%
4800	207	0.2%	416	-0.1%	239	0.0%	479	0.0%	259	0.2%	520	0.0%
9600	103	0.2%	207	0.2%	119	0.0%	239	0.0%	129	0.2%	259	0.2%
14.4k	68	0.6%	138	-0.1%	79	0.0%	159	0.0%	86	-0.2%	173	-0.2%
19.2k	51	0.2%	103	0.2%	59	0.0%	119	0.0%	64	0.2%	129	0.2%
28.8k	34	-0.8%	68	0.6%	39	0.0%	79	0.0%	42	0.9%	86	-0.2%
38.4k	25	0.2%	51	0.2%	29	0.0%	59	0.0%	32	-1.4%	64	0.2%
57.6k	16	2.1%	34	-0.8%	19	0.0%	39	0.0%	21	-1.4%	42	0.9%
76.8k	12	0.2%	25	0.2%	14	0.0%	29	0.0%	15	1.7%	32	-1.4%
115.2k	8	-3.5%	16	2.1%	9	0.0%	19	0.0%	10	-1.4%	21	-1.4%
230.4k	3	8.5%	8	-3.5%	4	0.0%	9	0.0%	4	8.5%	10	-1.4%
250k	3	0.0%	7	0.0%	4	-7.8%	8	2.4%	4	0.0%	9	0.0%
0.5M	1	0.0%	3	0.0%	-	_	4	-7.8%	-	-	4	0.0%
1M	0	0.0%	1	0.0%	-	-	-	-	-	-	-	-
Max. (1)	1 N	1bps	2 N	lbps	1.152	Mbps	2.304	Mbps	1.25	Mbps	2.5	Vbps

Table 17-12.	Examples of UBRBn	Settings for Common	v Used Oscillator Frequencies	(Continued)
				(Continucu)

1. UBRRn = 0, Error = 0.0%



depicted below, START and STOP conditions are signalled by changing the level of the SDA line when the SCL line is high.





19.3.3 Address Packet Format

All address packets transmitted on the TWI bus are 9 bits long, consisting of 7 address bits, one READ/WRITE control bit and an acknowledge bit. If the READ/WRITE bit is set, a read operation is to be performed, otherwise a write operation should be performed. When a Slave recognizes that it is being addressed, it should acknowledge by pulling SDA low in the ninth SCL (ACK) cycle. If the addressed Slave is busy, or for some other reason can not service the Master's request, the SDA line should be left high in the ACK clock cycle. The Master can then transmit a STOP condition, or a REPEATED START condition to initiate a new transmission. An address packet consisting of a slave address and a READ or a WRITE bit is called SLA+R or SLA+W, respectively.

The MSB of the address byte is transmitted first. Slave addresses can freely be allocated by the designer, but the address 0000 000 is reserved for a general call.

When a general call is issued, all slaves should respond by pulling the SDA line low in the ACK cycle. A general call is used when a Master wishes to transmit the same message to several slaves in the system. When the general call address followed by a Write bit is transmitted on the bus, all slaves set up to acknowledge the general call will pull the SDA line low in the ack cycle. The following data packets will then be received by all the slaves that acknowledged the general call. Note that transmitting the general call address followed by a Read bit is meaningless, as this would cause contention if several slaves started transmitting different data.

All addresses of the format 1111 xxx should be reserved for future purposes.



Figure 19-4. Address Packet Format





• Bits 7..0 – TWD: TWI Data Register

These eight bits constitute the next data byte to be transmitted, or the latest data byte received on the 2-wire Serial Bus.

19.6.5 TWI (Slave) Address Register – TWAR

Bit	7	6	5	4	3	2	1	0	_
	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	TWAR
Read/Write	R/W	•							
Initial Value	1	1	1	1	1	1	1	0	

The TWAR should be loaded with the 7-bit Slave address (in the seven most significant bits of TWAR) to which the TWI will respond when programmed as a Slave Transmitter or Receiver, and not needed in the Master modes. In multi master systems, TWAR must be set in masters which can be addressed as Slaves by other Masters.

The LSB of TWAR is used to enable recognition of the general call address (0x00). There is an associated address comparator that looks for the slave address (or general call address if enabled) in the received serial address. If a match is found, an interrupt request is generated.

Bits 7..1 – TWA: TWI (Slave) Address Register

These seven bits constitute the slave address of the TWI unit.

• Bit 0 – TWGCE: TWI General Call Recognition Enable Bit

If set, this bit enables the recognition of a General Call given over the 2-wire Serial Bus.

19.6.6 TWI (Slave) Address Mask Register – TWAMR



Bits 7..1 – TWAM: TWI Address Mask

The TWAMR can be loaded with a 7-bit Salve Address mask. Each of the bits in TWAMR can mask (disable) the corresponding address bits in the TWI Address Register (TWAR). If the mask bit is set to one then the address match logic ignores the compare between the incoming address bit and the corresponding bit in TWAR. Figure 19-10 shown the address match logic in detail.



If Auto Triggering is enabled, single conversions can be started by writing ADSC in ADCSRA to one. ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as one during a conversion, independently of how the conversion was started.

21.3 Prescaling and Conversion Timing





ADC CLOCK SOURCE

By default, the successive approximation circuitry requires an input clock frequency between 50 kHz and 200 kHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 200 kHz to get a higher sample rate.

The ADC module contains a prescaler, which generates an acceptable ADC clock frequency from any CPU frequency above 100 kHz. The prescaling is set by the ADPS bits in ADCSRA. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSRA. The prescaler keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.

When initiating a single ended conversion by setting the ADSC bit in ADCSRA, the conversion starts at the following rising edge of the ADC clock cycle.

A normal conversion takes 13 ADC clock cycles. The first conversion after the ADC is switched on (ADEN in ADCSRA is set) takes 25 ADC clock cycles in order to initialize the analog circuitry.

The actual sample-and-hold takes place 1.5 ADC clock cycles after the start of a normal conversion and 13.5 ADC clock cycles after the start of an first conversion. When a conversion is complete, the result is written to the ADC Data Registers, and ADIF is set. In Single Conversion mode, ADSC is cleared simultaneously. The software may then set ADSC again, and a new conversion will be initiated on the first rising ADC clock edge.

When Auto Triggering is used, the prescaler is reset when the trigger event occurs. This assures a fixed delay from the trigger event to the start of conversion. In this mode, the sample-and-hold takes place two ADC clock cycles after the rising edge on the trigger source signal. Three additional CPU clock cycles are used for synchronization logic.

In Free Running mode, a new conversion will be started immediately after the conversion completes, while ADSC remains high. For a summary of conversion times, see Table 21-1 on page 244.



- 4. Give XTAL1 a positive pulse. This loads the address low byte.
- C. Load Data Low Byte
- 1. Set XA1, XA0 to "01". This enables data loading.
- 2. Set DATA = Data low byte (0x00 0xFF).
- 3. Give XTAL1 a positive pulse. This loads the data byte.
- D. Load Data High Byte
- 1. Set BS1 to "1". This selects high data byte.
- 2. Set XA1, XA0 to "01". This enables data loading.
- 3. Set DATA = Data high byte (0x00 0xFF).
- 4. Give XTAL1 a positive pulse. This loads the data byte.
- E. Latch Data
- 1. Set BS1 to "1". This selects high data byte.
- 2. Give PAGEL a positive pulse. This latches the data bytes. (See Figure 25-3 for signal waveforms)

F. Repeat B through E until the entire buffer is filled or until all data within the page is loaded.

While the lower bits in the address are mapped to words within the page, the higher bits address the pages within the FLASH. This is illustrated in Figure 25-2 on page 289. Note that if less than eight bits are required to address words in the page (pagesize < 256), the most significant bit(s) in the address low byte are used to address the page when performing a Page Write.

- G. Load Address High byte
- 1. Set XA1, XA0 to "00". This enables address loading.
- 2. Set BS1 to "1". This selects high address.
- 3. Set DATA = Address high byte (0x00 0xFF).
- 4. Give XTAL1 a positive pulse. This loads the address high byte.
- H. Program Page
- 1. Give WR a negative pulse. This starts programming of the entire page of data. RDY/BSY goes low.
- 2. Wait until RDY/BSY goes high (See Figure 25-3 for signal waveforms).

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

J. End Page Programming

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

ATmega48/88/168

25.7.14 Reading the Calibration Byte

The algorithm for reading the Calibration byte is as follows (refer to "Programming the Flash" on page 287 for details on Command and Address loading):

- 1. A: Load Command "0000 1000".
- 2. B: Load Address Low Byte, 0x00.
- 3. Set OE to "0", and BS1 to "1". The Calibration byte can now be read at DATA.
- 4. Set OE to "1".

25.7.15 Parallel Programming Characteristics

Figure 25-7. Parallel Programming Timing, Including some General Timing Requirements







Note: 1. The timing requirements shown in Figure 25-7 (i.e., t_{DVXH}, t_{XHXL}, and t_{XLDX}) also apply to loading operation.





Figure 27-4. Active Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)





