Microprocessor based Design

The ATmega8

Basic Features

ELEMENTZ ENGINEERS GUILD PVT LTD.

ATmega8 - RISC Architecture

- 130 Instructions – Most Single-clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- 64 x 8 Special Function Registers (I/O Registers)
- Up to 16 MIPS Throughput at 16 MHz
- On-chip 2-cycle Multiplier

Nonvolatile Program and Data Memories

- 8K Bytes of In-System Self-Programmable Flash
  10,000 Write/Erase Cycles
- Optional Boot Code Section with Independent Lock Bits
- 512 Bytes EEPROM (100,000 Write/Erase Cycles)
- 1K Byte Internal SRAM
- Programming Lock for Software Security

Peripheral Features

- Two 8-bit Timer/Counters
- One 16-bit Timer/Counter with Capture Mode
- Real Time Counter with Separate Oscillator
- Three PWM Channels
- 6-channel ADC with 10 resp 8 Bit resolution (TQFP: 8 channels)
- Two-wire Serial Interface (TWI)
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with On-chip Oscillator
- On-chip Analog Comparator

Special Microcontroller Features

- Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Five Sleep Modes

I/O and Packages

- 23 Programmable I/O Lines
- 28-lead PDIP, 32-lead TQFP, and 32-pad MLF

Operating Voltages

- 2.7 - 5.5V (ATmega8L)
- 4.5 - 5.5V (ATmega8)

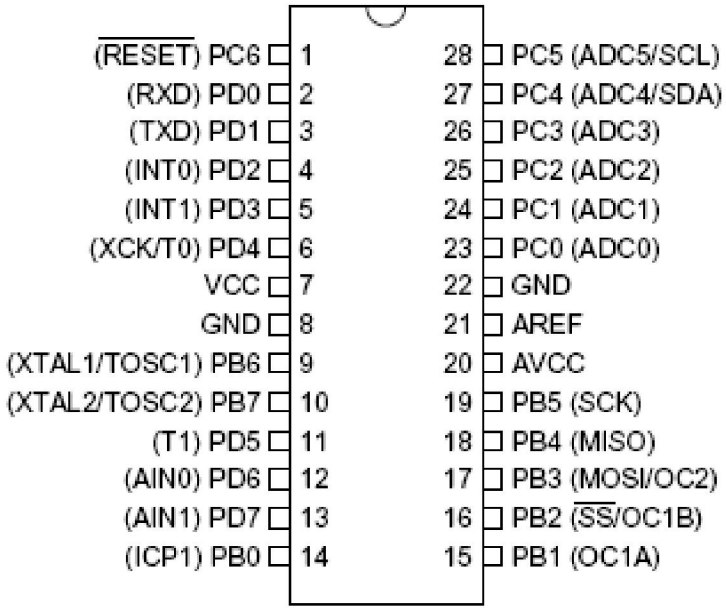
Speed Grades

- 0 - 8 MHz (ATmega8L)
- 0 - 16 MHz (ATmega8)

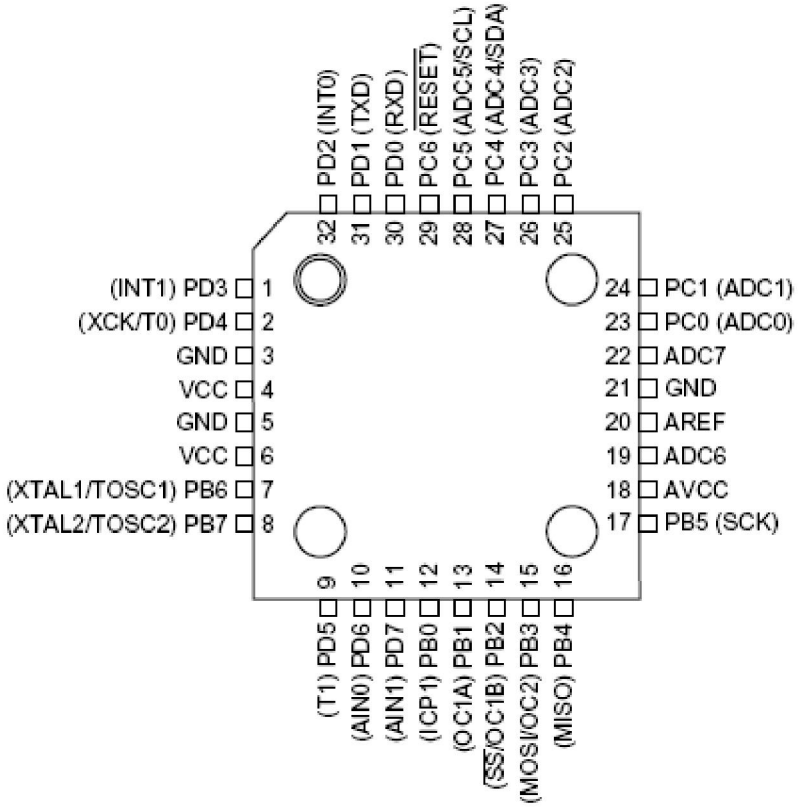
Power Consumption at 4 Mhz, 3V, 25°C

- Active: 3.6 mA
- Idle Mode: 1.0 mA
- Power-down Mode: 0.5 µA

PDIP



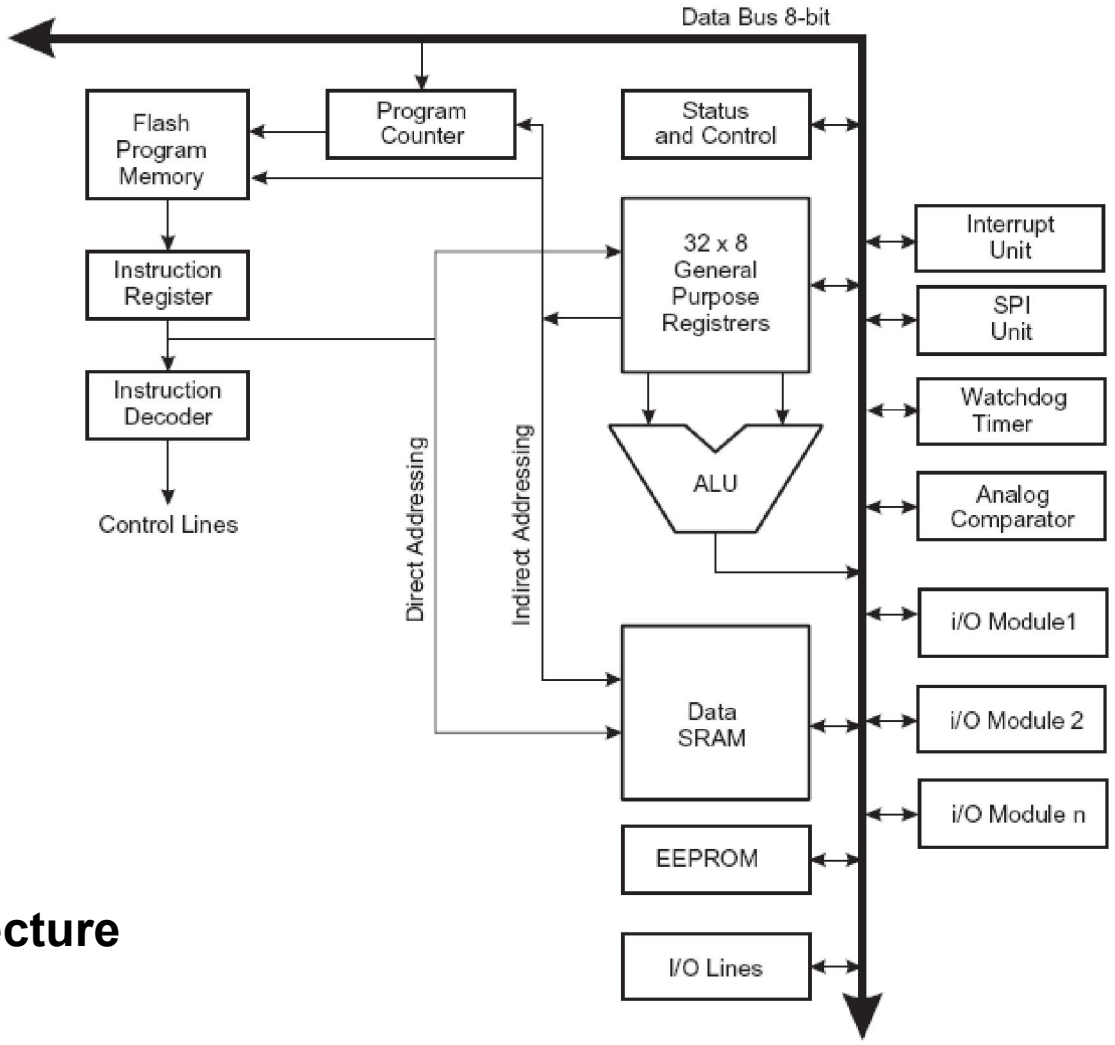
TQFP Top View



ATmega8 Pinout and Packages (DIP and TQFP)

Mega8 CPU Core

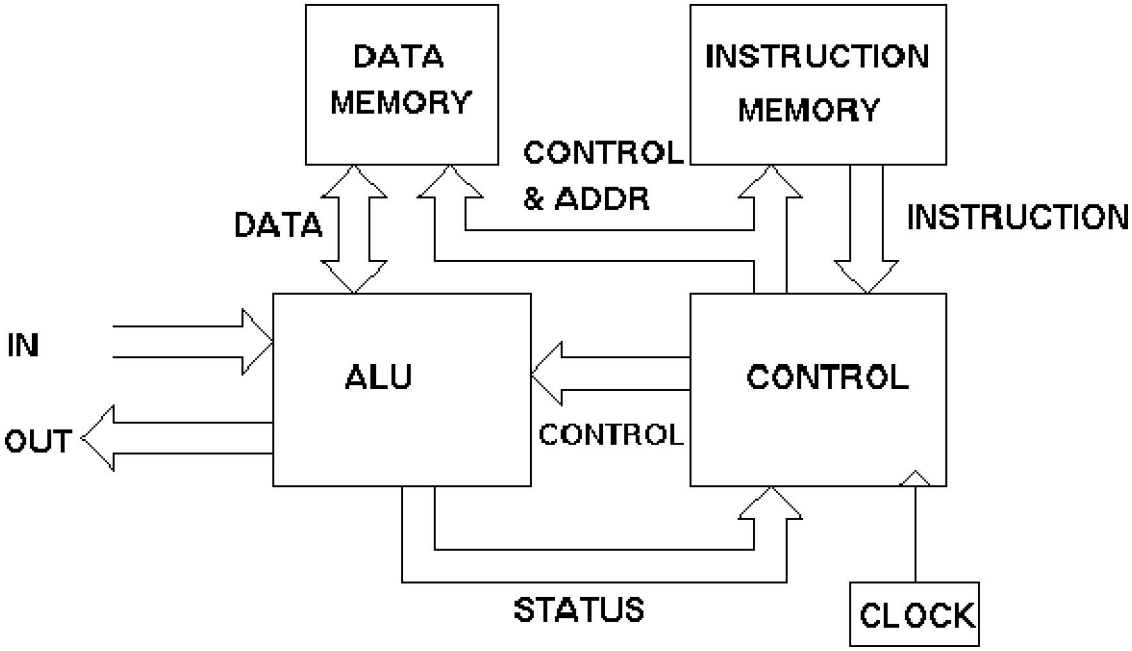
- Seperate Instruction and Data Memories (Harvard)
- all 32 General Purpose Registers connected to ALU
- I/O Modules connected to Data Bus and accessible via Special Function Registers



ATmega8 Core Architecture

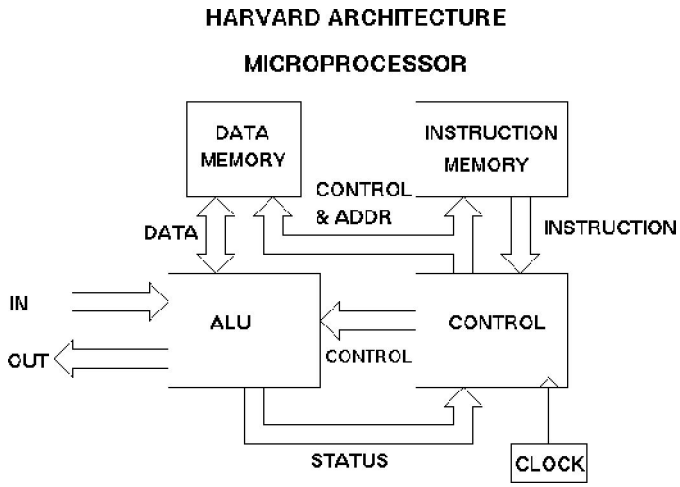
HARVARD ARCHITECTURE

MICROPROCESSOR



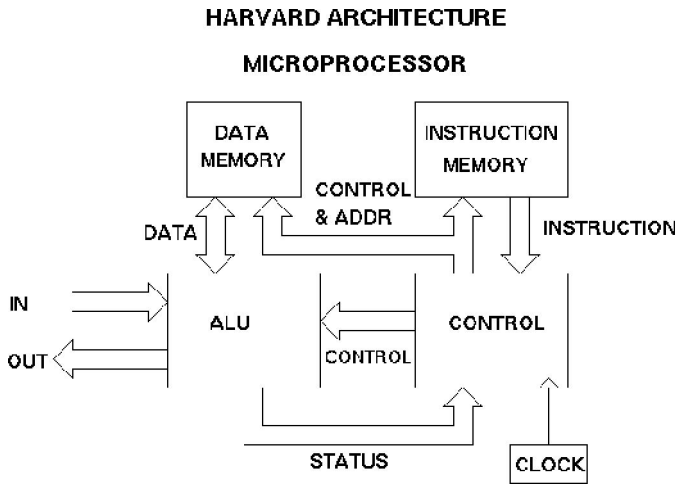
Harvard architecture

- Separate storage and signal pathways for instructions and data.
- History: Harvard Mark I relay-based computer
- word width, timing, and implementation technology of instruction and data memories can differ.
- Contrast: 'Von Neumann' - architecture: Instructions and data use the same signal pathways and memory.



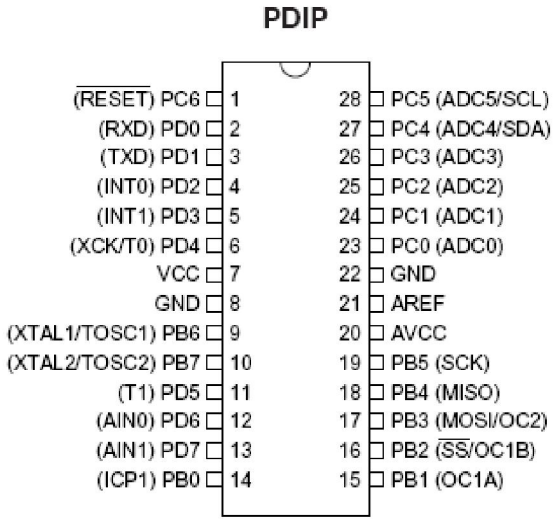
Harvard architecture

- Ability to fetch the next instruction at the same time it completes the current instruction.
- Speed is gained at the expense of more complex electrical circuitry.



In a computer with Harvard architecture, the CPU can read an instruction and data from memory at the same time.

Pin and Port Overview:



GND: Ground (0V)
VCC: Digital Supply Voltage (2,7 – 5,5V)
AVCC: Analog Supply Voltage
connect to low-pass filtered VCC
AREF: Analog Reference Voltage, usually AVCC
/Reset: Low level on this pin will generate a reset

Port B, Port C, Port D:

General Purpose 8 Bit bidirectional I/O - Ports,
optional internal pullup-resistors when configured as input
output source capability: 20mA

Special Functions of the Ports available as configured using the SFRs:

Port D: Uart, external Interrupts, Analog Comparator
Port B: External Oscillator/Crystal, SPI
Port C: A/D converters, TWI

Memory organization

AVR Memory organization:

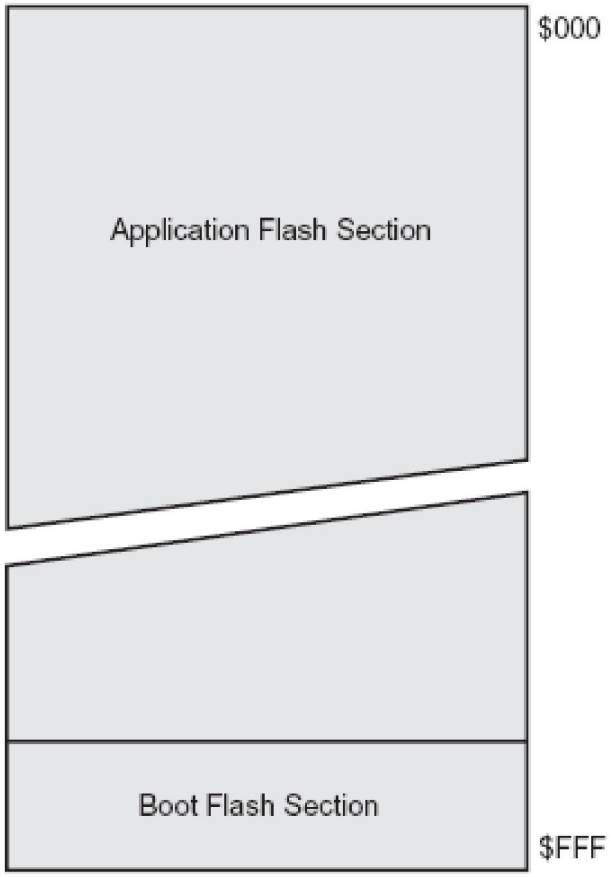
- **Program Flash Memory:**

  On-chip, in system programmable

  8 Kbytes, organized in 4K 16 bit words
  Program Counter (PC) = 12 bits

  Accessible via special instructions: LPM, SPM

  Boot Loader support: Boot Flash Section,
  SPM can be executed only from Boot Flash



AVR Memory organization:

- **EEPROM - Memory:**

  512 Bytes, single Bytes can be read and written

  Special EEPROM read and write procedure using SFRs:

  EEPROM Address Register, EEPROM Data Register,
  EEPROM Control Register
  C – Library Functions available

  Precautions to prevent EEPROM memory corruption:

  - no flash memory or interrupt operations
  - stable power supply

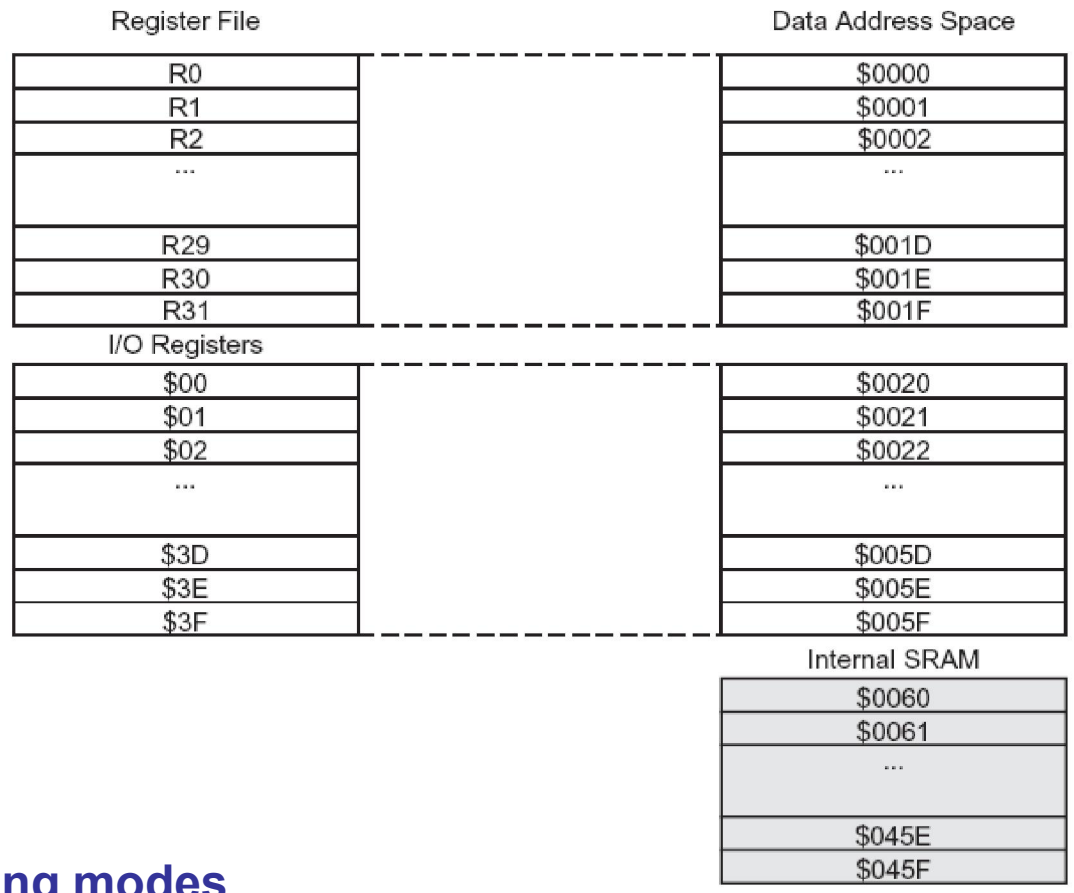
AVR Memory organization:

- **SRAM Data Memory:**

  32 GPR's and
  64 SFR's mapped
  to SRAM memory space

  SFR's accessed
  via in / out instructions
  (I/O-registers)

  1 Kbytes of internal
  SRAM can be accessed
  from address 0x060
  to address 0x45f



5 Direct and indirect addressing modes

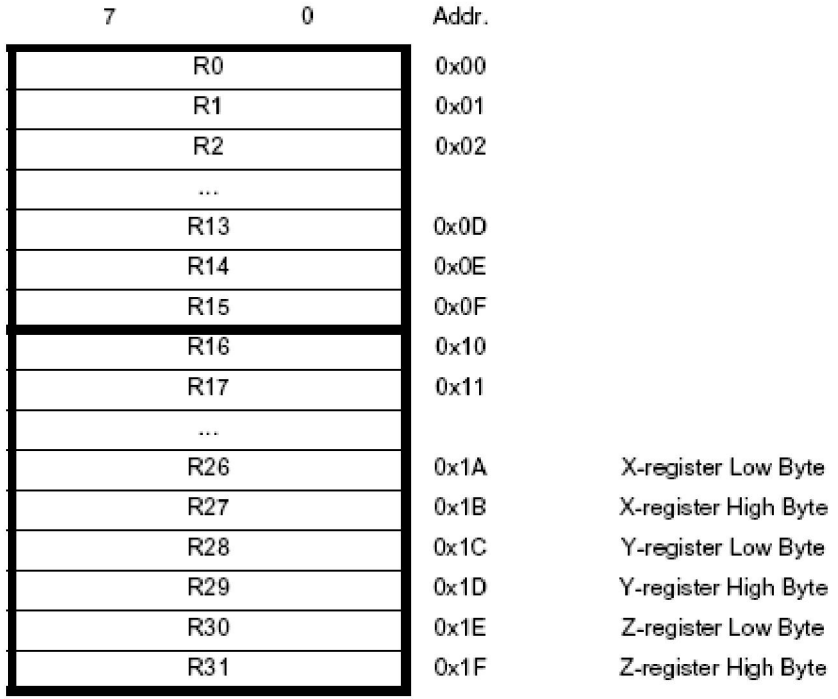
AVR Memory organization:

- **General Purpose Registers:**

Although not being physically implemented as SRAM locations, GPR's can be accessed by SRAM locations

X, Y and Z 16-bit registers can be used for indirect addressing

ALU - Input / output schemes:
- one 8-bit operand, 8-bit result
- two 8-bit operands, 8-bit result
- two 8-bit operands, 16-bit result
- one 16-bit operand, 16-bit result



I/O Memory (SFR) Overview

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	9
0x3E (0x5E)	SPH	–	–	–	–	–	SP10	SP9	SP8	11
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	11
0x3C (0x5C)	Reserved									
0x3B (0x5B)	GICR	INT1	INT0	–	–	–	–	IVSEL	IVCE	47, 65
0x3A (0x5A)	GIFR	INTF1	INTF0	–	–	–	–	–	–	66
0x39 (0x59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	–	TOIE0	70, 100, 120
0x38 (0x58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	–	TOV0	71, 101, 120
0x37 (0x57)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	209
0x36 (0x56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	167
0x35 (0x55)	MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00	31, 64
0x34 (0x54)	MCUCSR	–	–	–	–	WDRF	BORF	EXTRF	PORF	39
0x33 (0x53)	TCCR0	–	–	–	–	–	CS02	CS01	CS00	70
0x32 (0x52)	TCNT0	Timer/Counter0 (8 Bits)								70
0x31 (0x51)	OSCCAL	Oscillator Calibration Register								29
0x30 (0x50)	SFIOR	–	–	–	–	ACME	PUD	PSR2	PSR10	56, 73, 121, 189
0x2F (0x4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	95
0x2E (0x4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	98
0x2D (0x4D)	TCNT1H	Timer/Counter1 – Counter Register High byte								99
0x2C (0x4C)	TCNT1L	Timer/Counter1 – Counter Register Low byte								99
0x2B (0x4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High byte								99
0x2A (0x4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low byte								99
0x29 (0x49)	OCR1BH	Timer/Counter1 – Output Compare Register B High byte								99
0x28 (0x48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low byte								99
0x27 (0x47)	ICR1H	Timer/Counter1 – Input Capture Register High byte								100
0x26 (0x46)	ICR1L	Timer/Counter1 – Input Capture Register Low byte								100
0x25 (0x45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	115
0x24 (0x44)	TCNT2	Timer/Counter2 (8 Bits)								117
0x23 (0x43)	OCR2	Timer/Counter2 Output Compare Register								117
0x22 (0x42)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	117
0x21 (0x41)	WDTCR	–	–	–	WDCE	WDE	WDP2	WDP1	WDP0	41
0x20[1] (0x40)[1]	UBRRH	URSEL	–	–	–	UBRR[11:8]				154
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	152

I/O Memory (SFR) Overview

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x1F (0x3F)	EEARH	–	–	–	–	–	–	–	EEAR8	18
0x1E (0x3E)	EEARL	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	18
0x1D (0x3D)	EEDR	EEPROM Data Register								18
0x1C (0x3C)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	18
0x1B (0x3B)	Reserved									
0x1A (0x3A)	Reserved									
0x19 (0x39)	Reserved									
0x18 (0x38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	63
0x17 (0x37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	63
0x16 (0x36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	63
0x15 (0x35)	PORTC	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	63
0x14 (0x34)	DDRC	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	63
0x13 (0x33)	PINC	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	63
0x12 (0x32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	63
0x11 (0x31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	63
0x10 (0x30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	63
0x0F (0x2F)	SPDR	SPI Data Register								128
0x0E (0x2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	128
0x0D (0x2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	126
0x0C (0x2C)	UDR	USART I/O Data Register								149
0x0B (0x2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	150
0x0A (0x2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	151
0x09 (0x29)	UBRRL	USART Baud Rate Register Low byte								154
0x08 (0x28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	190
0x07 (0x27)	ADMUX	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	201
0x06 (0x26)	ADCSRA	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	203
0x05 (0x25)	ADCH	ADC Data Register High byte								204
0x04 (0x24)	ADCL	ADC Data Register Low byte								204
0x03 (0x23)	TWDR	Two-wire Serial Interface Data Register								169
0x02 (0x22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	170
0x01 (0x21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	169
0x00 (0x20)	TWBR	Two-wire Serial Interface Bit Rate Register								167

Important I/O Registers:
SREG – Status Register

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – I: Global Interrupt Enable

Bit 6 – T: Bit Copy Storage

Bit 5 – H: Half Carry Flag

Bit 4 – S: Sign Bit

Bit 3 – V: Two's Complement Overflow Flag

Bit 2 – N: Negative Flag

Bit 1 – Z: Zero Flag

Bit 0 – C: Carry Flag

Important I/O Registers:
Stack Pointer (SPH and SPL)

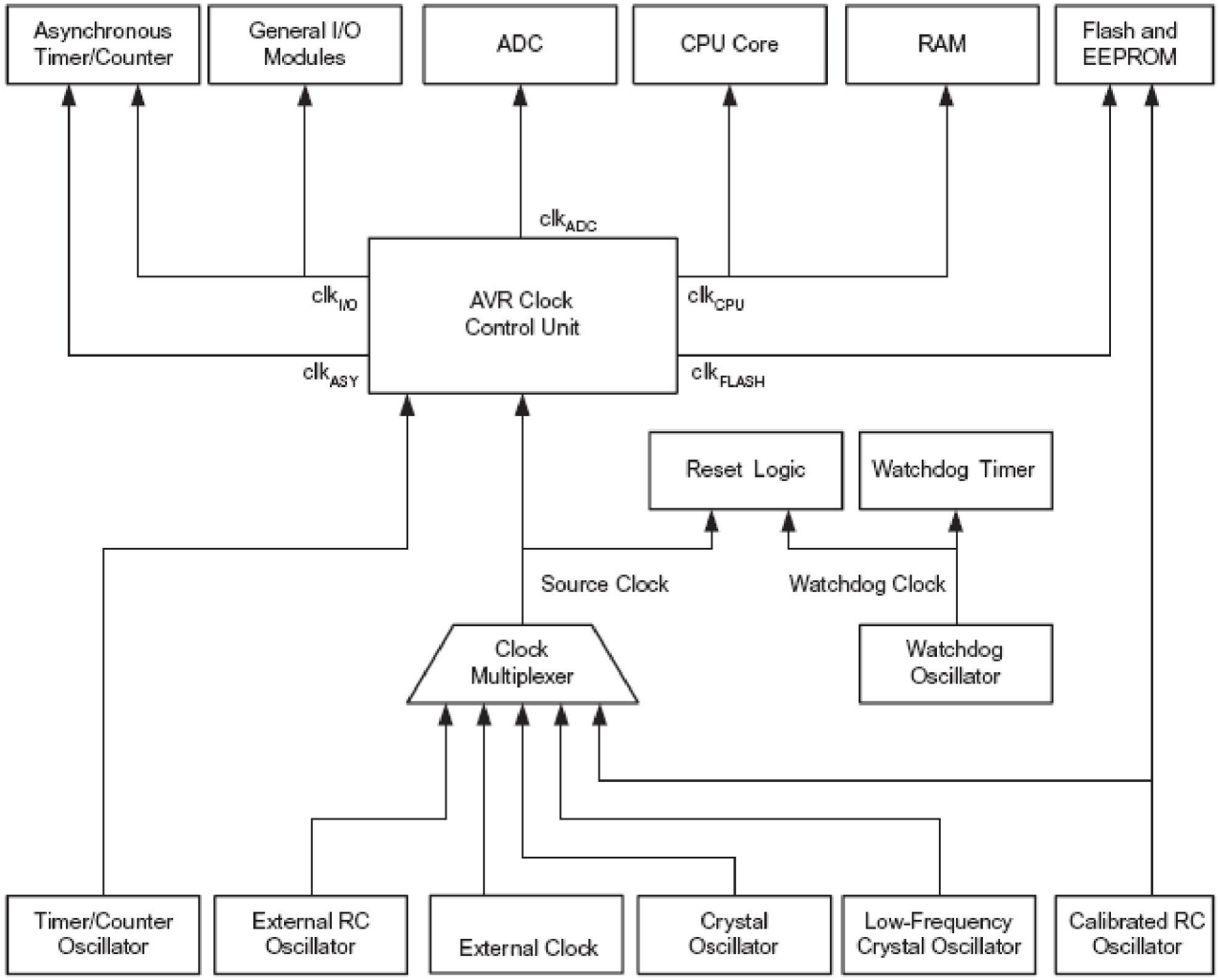
Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

- Stack is a LIFO buffer located in SRAM
- Stack Pointer addresses the current location
- Push and pop instructions write / read from Stack
- Enter or return from subroutines / interrupt routines:

  Address and Parameters transferred via Stack

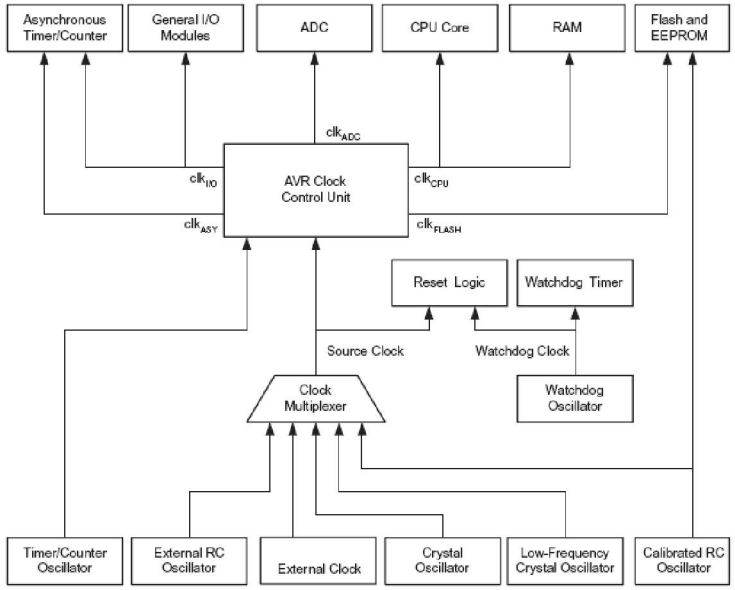
Clock Options

System Clock Options:



System Clock Options:

- Clock Muliplexer selects the clock source according to FUSE settings
- Clock Control Unit distributes clocks clocks can be halted to reduce power consumption
- CPU Clock: CPU, ALU, GPRs
- I/O Clock: Ports, Timers, SPI, UART
- ADC Clock: seperate cock for ADC noise reduction in sleep mode
- Asynchronous Timer Clock: external 32kHz Crystal for realtime clock, keeps timer module running during sleep mode

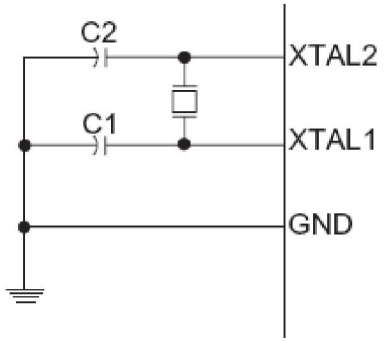


System Clock Options - FUSE bits:

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

- The four CKSEL Bits of the FUSE – Byte select the main Clock Source
- The startup time to stabilize power supply and oscillator can be changed with the SUT fuses
- The device is shipped with CKSEL = 0001 (1 MHZ internal RC oscillator) and SUT = 10 (slowly rising power, 65ms)

System Clock Options - using an external crystal:



CKOPT	CKSEL3..1	Frequency Range(MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals (pF)
1	101(1)	0.4 - 0.9	–
1	110	0.9 - 3.0	12 - 22
1	111	3.0 - 8.0	12 - 22
0	101, 110, 111	1.0 ≤	12 - 22

- CKOPT influences the output swing of the inverting oscillator amplifier (1 = full rail to rail swing, 0 = power save mode)
- For crystals from 3 – 8 MHz set CKOPT = 1 and CKSEL3..1 = 111

System Clock Options - using the internal RC oscillator

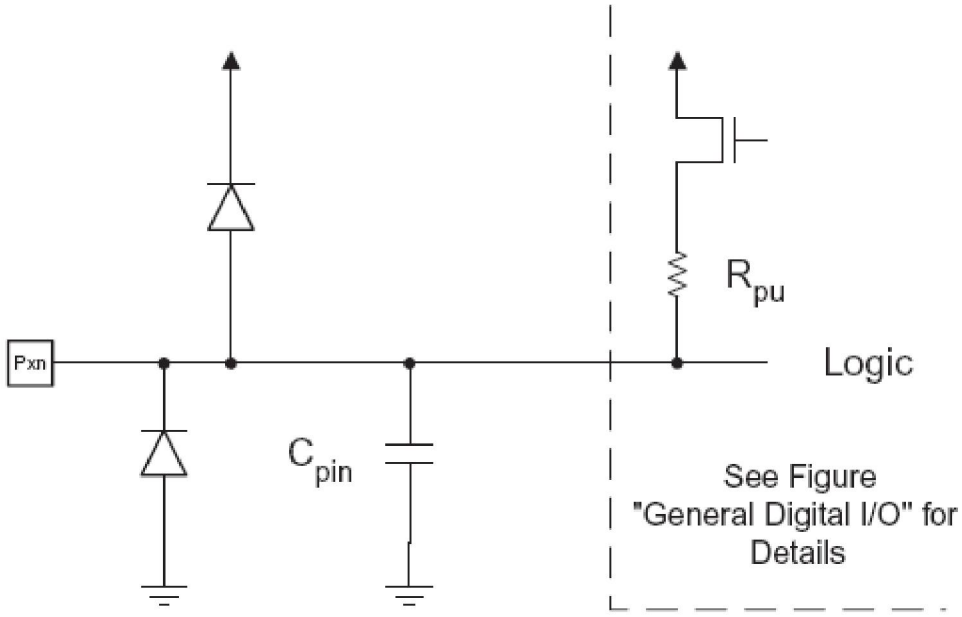
CKSEL3..0	Nominal Frequency (MHz)
0001(1)	1.0
0010	2.0
0011	4.0
0100	8.0

- Fixed 1, 2, 4 or 8 MHz clock
- works without external components
- changes with temperature and operating voltage

detailed information on other clock options, startup times, calibration is found in the ATmega8 data sheet, pp. 23

I/O Ports

I/O Ports



- General Purpose IO : Data Direction Input or Output
- Internal Pullup can be used for Input Pins
- Output driver can source 20mA current
- protection diodes to GND and VCC

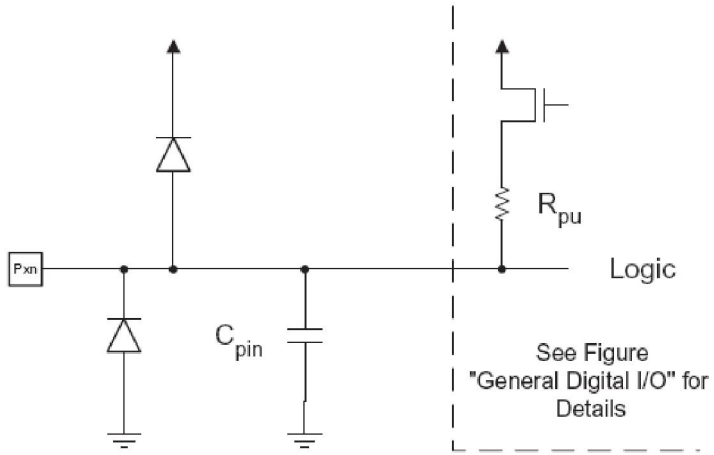
I/O Ports

- 3 I/O-Registers for each port:

  Data Register (r/w):
  PORTB, PORTC, PORTD

  Data Direction Register (r/w):
  DDRB, DDRC, DDRD

  Port Input Pin Register (r):
  PINB, PINC, PIND



The Bits of these registers set the configuration for one Port Pin.

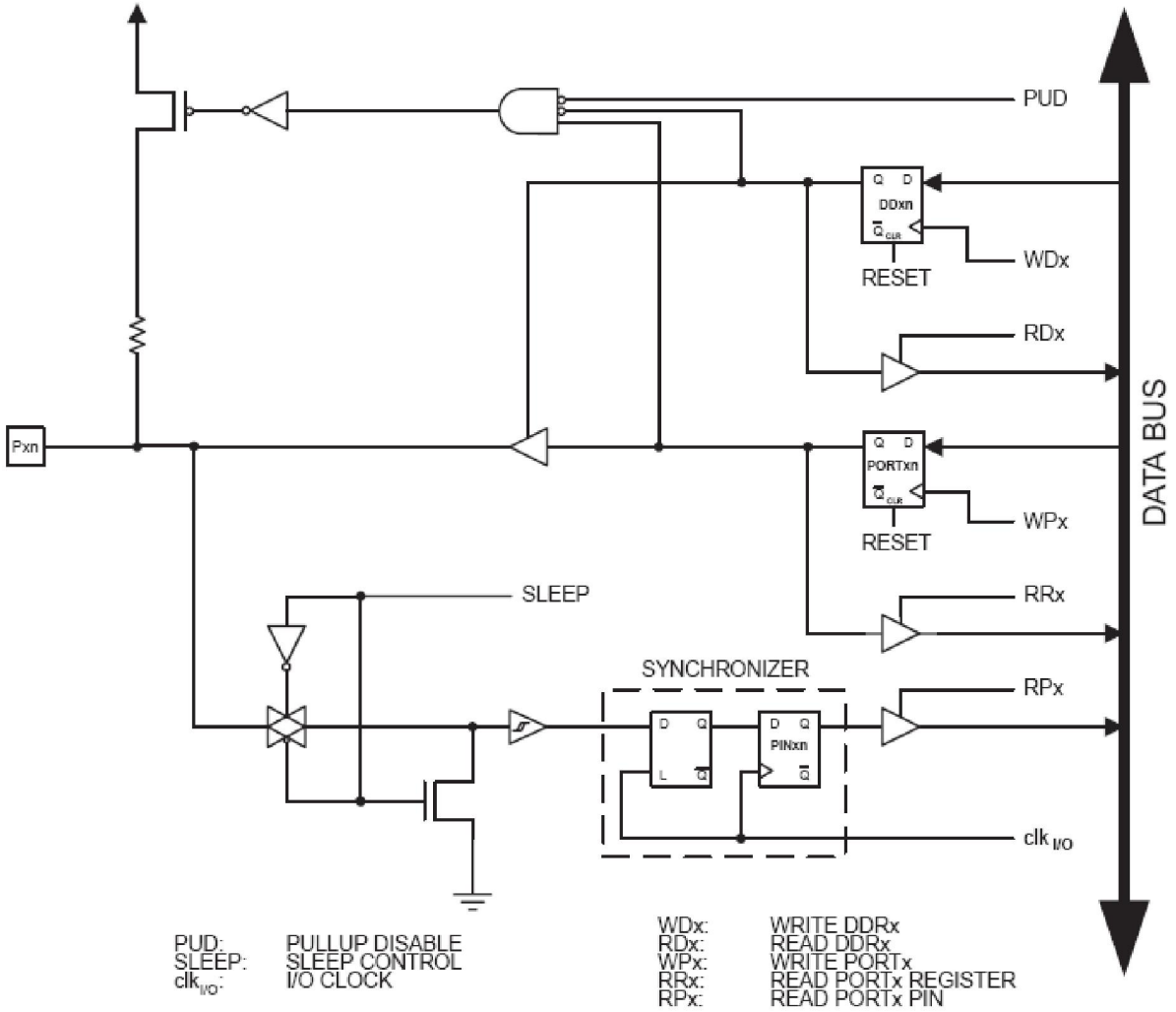
I/O Ports

General Digital IO

Logic of GPIO-Ports:

DDx
PORTx
PINx

Common to all Ports:
Pullup disable (PUD),
SLEEP



I/O Ports – Configuration and usage

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

C-Example 1 - Configure Pin B3 as output, set output level to VCC:

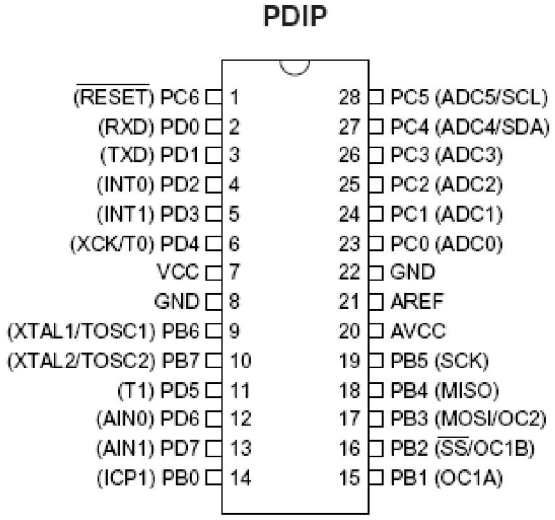
```
DDRB |= (1<<3);  PORTB |= (1<<3);
```

C-Example 2 - Configure Pin D2 as input with pullup, read pin value:

```
DDRD &= ~(1<<2);  PORTD |= (1<<2); uint8_t x = PIND & (1<<2);
```

Alternate Port functions Port B

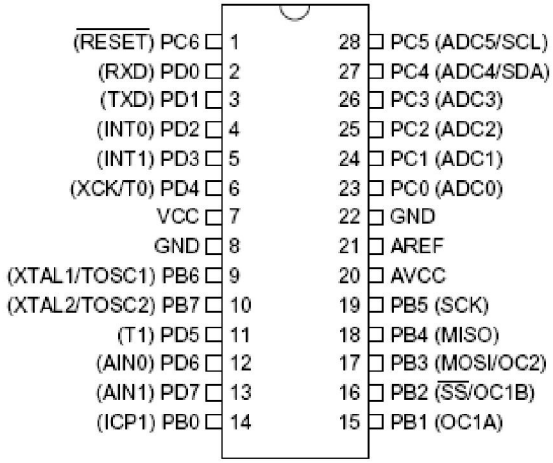
Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1)
PB5	SCK (SPI Bus Master clock Input)
PB4	MISO (SPI Bus Master Input/Slave Output)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2 (Timer/Counter2 Output Compare Match Output)
PB2	$\overline{SS}$ (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output)
PB0	ICP1 (Timer/Counter1 Input Capture Pin)



Alternate Port functions Port C

Port Pin	Alternate Function
PC6	$\overline{\text{RESET}}$ (Reset pin)
PC5	ADC5 (ADC Input Channel 5) SCL (Two-wire Serial Bus Clock Line)
PC4	ADC4 (ADC Input Channel 4) SDA (Two-wire Serial Bus Data Input/Output Line)
PC3	ADC3 (ADC Input Channel 3)
PC2	ADC2 (ADC Input Channel 2)
PC1	ADC1 (ADC Input Channel 1)
PC0	ADC0 (ADC Input Channel 0)

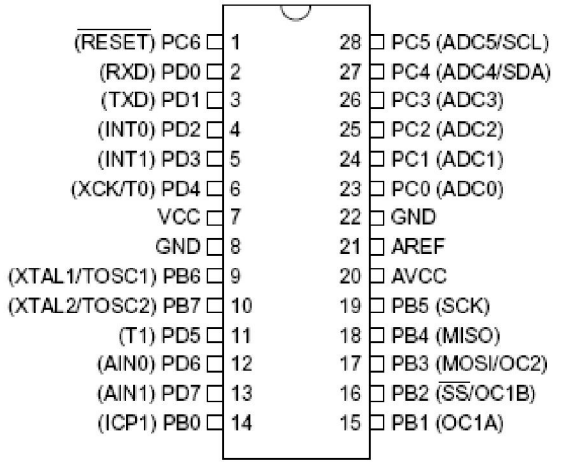
PDIP



Alternate Port functions Port D

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input)
PD6	AIN0 (Analog Comparator Positive Input)
PD5	T1 (Timer/Counter 1 External Counter Input)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

PDIP



Reset- and Interrupt Handling

Interrupt Processing

- several Interrupt Sources:
  External Interrupts, Timer, Bus-Peripherals,
  ADC, EEPROM
- individual Interrupt-Enable bits in the SFR‘s
- global interrupt enable Bit in SREG,
  set with sei() and clear with cli() instruction
- flagged (remembered) and non-flagged interrupt sources
- lowest addresses in program memory reserved
  for the interrupt vector table
- higher priority interrupts have lower addresses

Reset-Vector and Interrupt-Vectors

- Word addresses 0, 1 – 19 in Flash Ram
- When a reset or interrupt occurs, the CPU calls the address
- Install an Interrupt Handler: modify the vector table to jump to your user-handler
- return from interrupt: reti

Vector No.	Program Address(2)	Source	Interrupt Definition
1	0x000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset, and Watchdog Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	TIMER2 COMP	Timer/Counter2 Compare Match
5	0x004	TIMER2 OVF	Timer/Counter2 Overflow
6	0x005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	0x006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	0x007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	0x008	TIMER1 OVF	Timer/Counter1 Overflow
10	0x009	TIMER0 OVF	Timer/Counter0 Overflow
11	0x00A	SPI, STC	Serial Transfer Complete
12	0x00B	USART, RXC	USART, Rx Complete
13	0x00C	USART, UDRE	USART Data Register Empty
14	0x00D	USART, TXC	USART, Tx Complete
15	0x00E	ADC	ADC Conversion Complete
16	0x00F	EE_RDY	EEPROM Ready
17	0x010	ANA_COMP	Analog Comparator
18	0x011	TWI	Two-wire Serial Interface
19	0x012	SPM_RDY	Store Program Memory Ready

Reset-Vector and Interrupt-Vectors

- example shows full featured vector table
- 19 handlers installed
- program execution after reset: jmp RESET ($013)
- Main program is located at $013, beyond the vectors

```
address Labels  Code                    Comments
$000            rjmp    RESET           ; Reset Handler
$001            rjmp    EXT_INT0        ; IRQ0 Handler
$002            rjmp    EXT_INT1        ; IRQ1 Handler
$003            rjmp    TIM2_COMP       ; Timer2 Compare Handler
$004            rjmp    TIM2_OVF        ; Timer2 Overflow Handler
$005            rjmp    TIM1_CAPT       ; Timer1 Capture Handler
$006            rjmp    TIM1_COMPA      ; Timer1 CompareA Handler
$007            rjmp    TIM1_COMPB      ; Timer1 CompareB Handler
$008            rjmp    TIM1_OVF        ; Timer1 Overflow Handler
$009            rjmp    TIM0_OVF        ; Timer0 Overflow Handler
$00a            rjmp    SPI_STC         ; SPI Transfer Complete Handler
$00b            rjmp    USART_RXC       ; USART RX Complete Handler
$00c            rjmp    USART_UDRE      ; UDR Empty Handler
$00d            rjmp    USART_TXC       ; USART TX Complete Handler
$00e            rjmp    ADC             ; ADC Conversion Complete Handler
$00f            rjmp    EE_RDY          ; EEPROM Ready Handler
$010            rjmp    ANA_COMP        ; Analog Comparator Handler
$011            rjmp    TWSI            ; Two-wire Serial Interface Handler
$012            rjmp    SPM_RDY         ; Store Program Memory Ready Handler
;
$013    RESET:  ldi         r16,high(RAMEND) ; Main program start
$014            out     SPH,r16         ; Set Stack Pointer to top of RAM
$015            ldi     r16,low(RAMEND)
$016            out     SPL,r16
$017            sei                     ; Enable interrupts
$018            <instr>  xxx
...             ...       ...
```

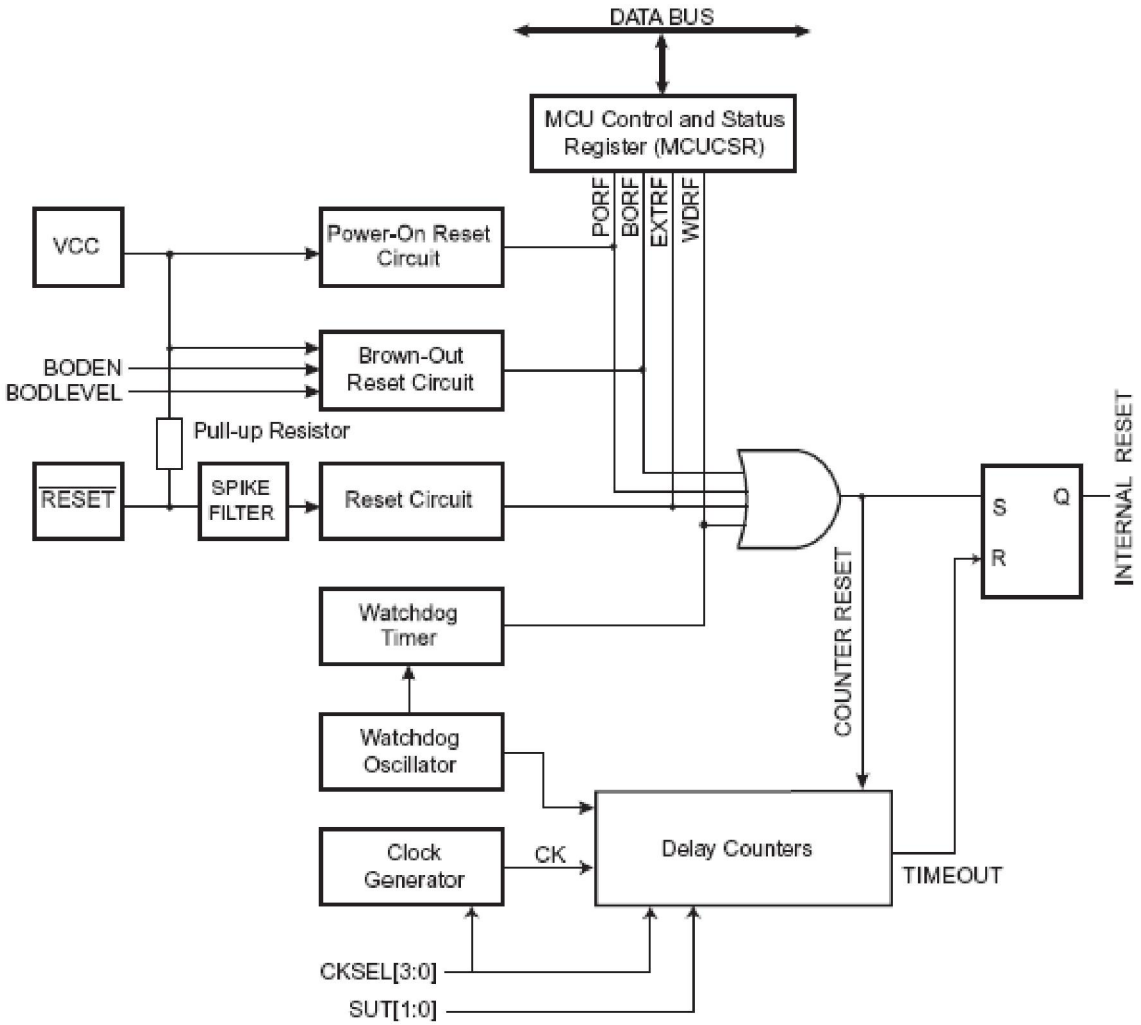
Reset- and Interrupt- Vectors

BOOTRST(1)	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

- Reset vector can be set to the Bootloader section using the BOOTRST fuse bit
- Interrupt vectors can be set to the Bootloader section using the IVSEL bit of the General Interrupt Contol Register

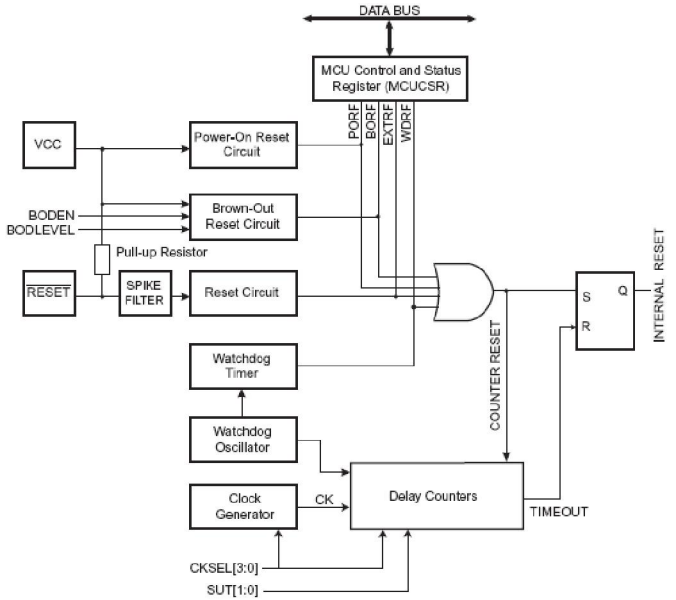
Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	–	–	–	–	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

AVR Reset Sources:



Reset Sources:

- Power-on Reset: supply voltage is below the Power-on Reset threshold
- External Reset: low level is present on /RESET – input pin
- Watchdog Reset: Watchdog Timer enabled and period expires
- Brown-out Reset: Brown-out Detector enabled and supply voltage below threshold



Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	WDRF	BORF	EXTRF	PORF	MCUCSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

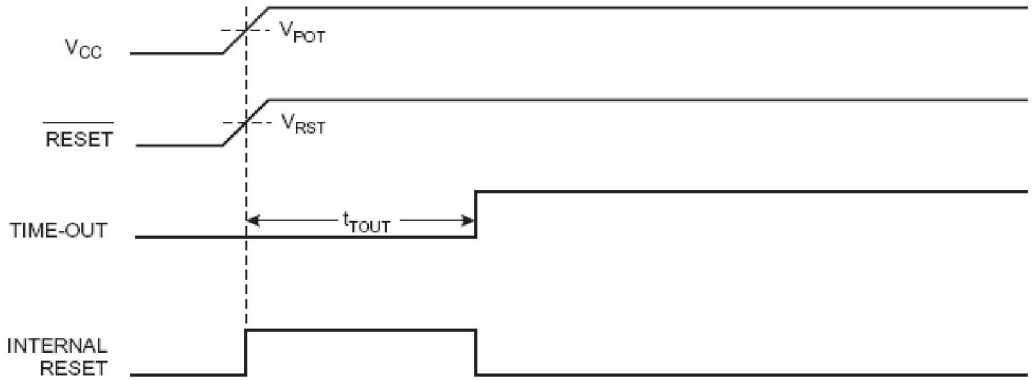
MCUCSR provides information on which reset source caused a CPU reset

Reset Voltage Thresholds

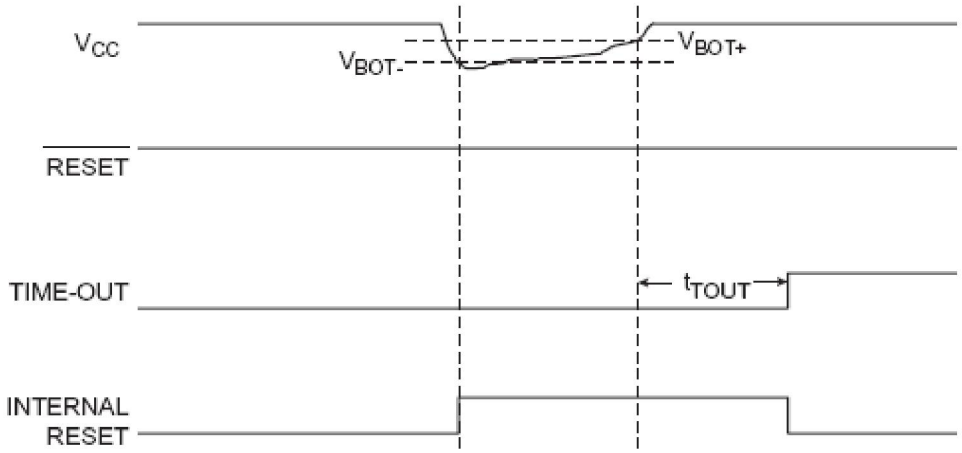
Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{POT}	Power-on Reset Threshold Voltage (rising)(1)			1.4	2.3	V
	Power-on Reset Threshold Voltage (falling)			1.3	2.3	V
V_{RST}	$\overline{\text{RESET}}$ Pin Threshold Voltage		0.1		0.9	V_{CC}
t_{RST}	Minimum pulse width on $\overline{\text{RESET}}$ Pin				1.5	µs
V_{BOT}	Brown-out Reset Threshold Voltage(2)	BODLEVEL = 1	2.4	2.6	2.9	V
		BODLEVEL = 0	3.7	4.0	4.5	
t_{BOD}	Minimum low voltage period for Brown-out Detection	BODLEVEL = 1		2		µs
		BODLEVEL = 0		2		µs
V_{HYST}	Brown-out Detector hysteresis			130		mV

Reset Voltage Thresholds:

Example:
Power-on Reset



Example:
Brown Out Reset



External Interrupts Int0 and Int1:

- Int0 connected to PD2
- Int1 connected to PD3
- asynchronous operation: can wake up CPU
- rising/falling edge or low level can trigger interrupt, defined by Interrupt Sense control – bits of MCUCR SFU

Bit	7	6	5	4	3	2	1	0	
	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
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	

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical 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.

External Interrupts Int0 and Int1:

- Int0 and Int1 have to be enabled by the GICR (+ I-bit in SREG)

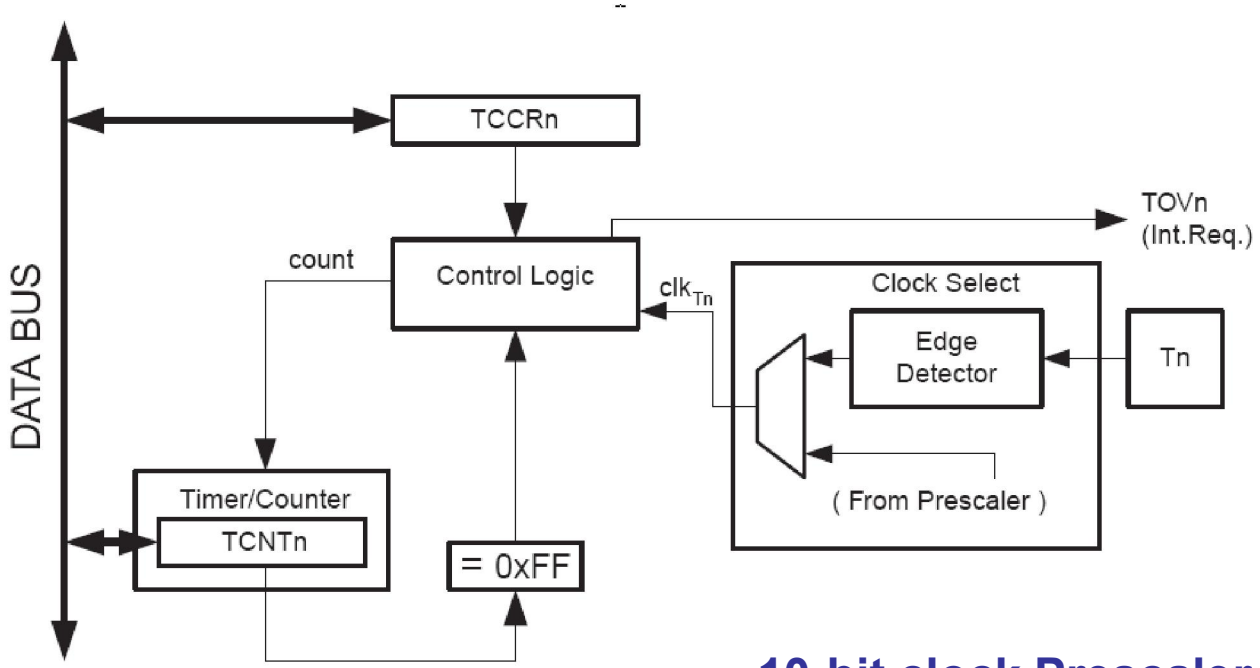
Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	–	–	–	–	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- flagged interrupts: General Interrupt Flag Register (GIFR) indicates when an interrupt request happened

- flags are cleared by executing the interrupt service routine (ISR) or by writing 1 to the flag bit of GIFR

Bit	7	6	5	4	3	2	1	0	
	INTF1	INTF0	–	–	–	–	–	–	GIFR
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

8-bit Timer / Counter0



- 10-bit clock Prescaler
  timer-clk (t0) = clk (IO) / prescaler

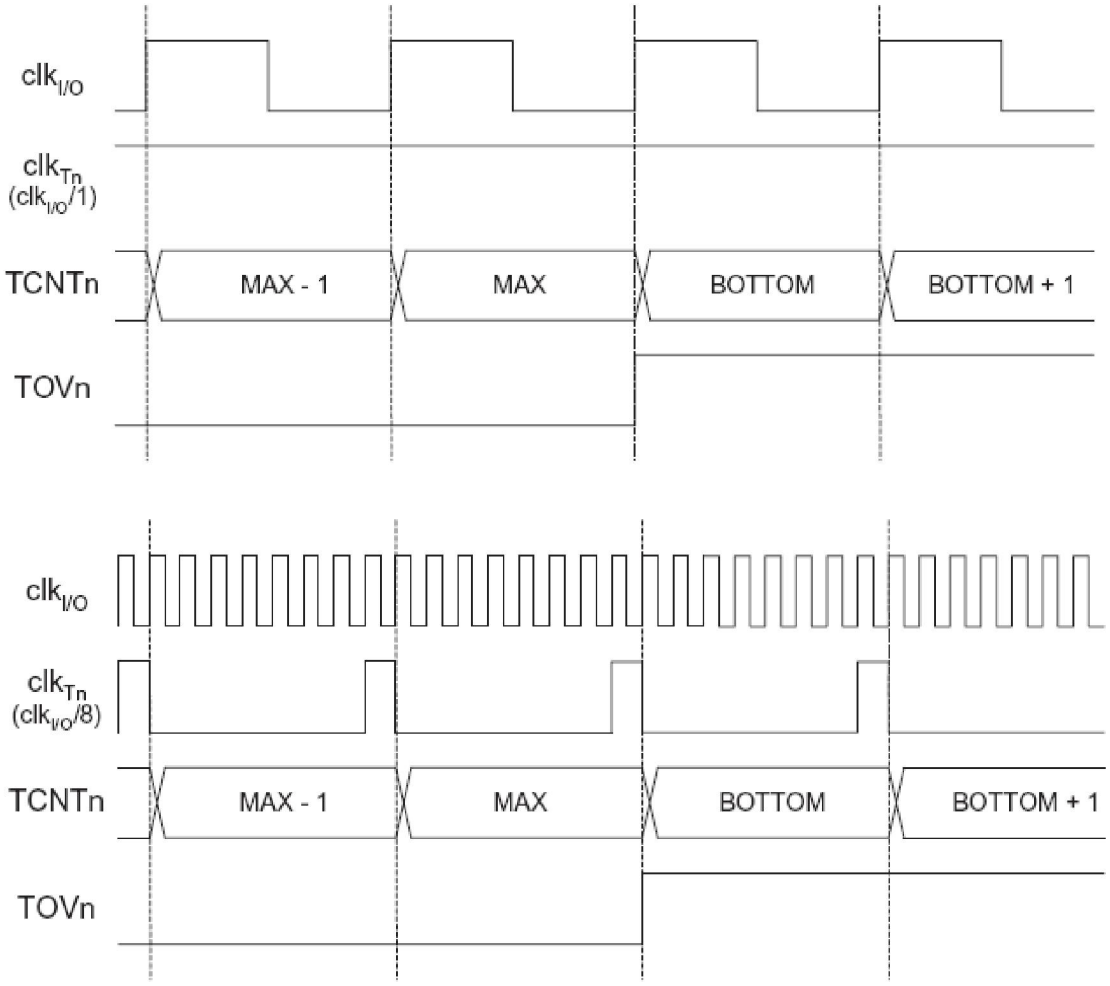
- External clock source T0 connected to PD4
  cannot be prescaled, clk(ext) <= clk (IO) / 2.5

8-bit Timer / Counter0 - prescaler operation

No prescaler

MAX=0xff
BOTTOM=0

Prescaler = 8



8-bit Timer / Counter0 usage

Timer/Counter0 Control Register (TCCR0), Bits CS02-CS00 select Clock Source and Prescaler Value :

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	CS02	CS01	CS00	TCCR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

8-bit Timer / Counter0 usage

Timer/Counter0 Register (TCNT0) :
read/write, incremented per CLK cycle, overflow: 0xff

Bit	7	6	5	4	3	2	1	0	
	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	

- A Reload-Value is used to fine-tune the interrupt interval
- write Reload-Value to TCNT0 in the ISR

8-bit Timer / Counter0 usage

Timer/Counter Interrupt Mask Register (TIMSK) :

Bit 0 : Timer 0 interrupt enable

set 1 to enable timer 0 overflow interrupt (+ I-Bit in SREG)

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	–	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Timer Interrupt Flag Register (TIFR) :

TOV0 indicates a Timer0 overflow, cleared by hardware when the ISR is executed or by writing 1 to the flag

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	–	TOV0	TIFR
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	