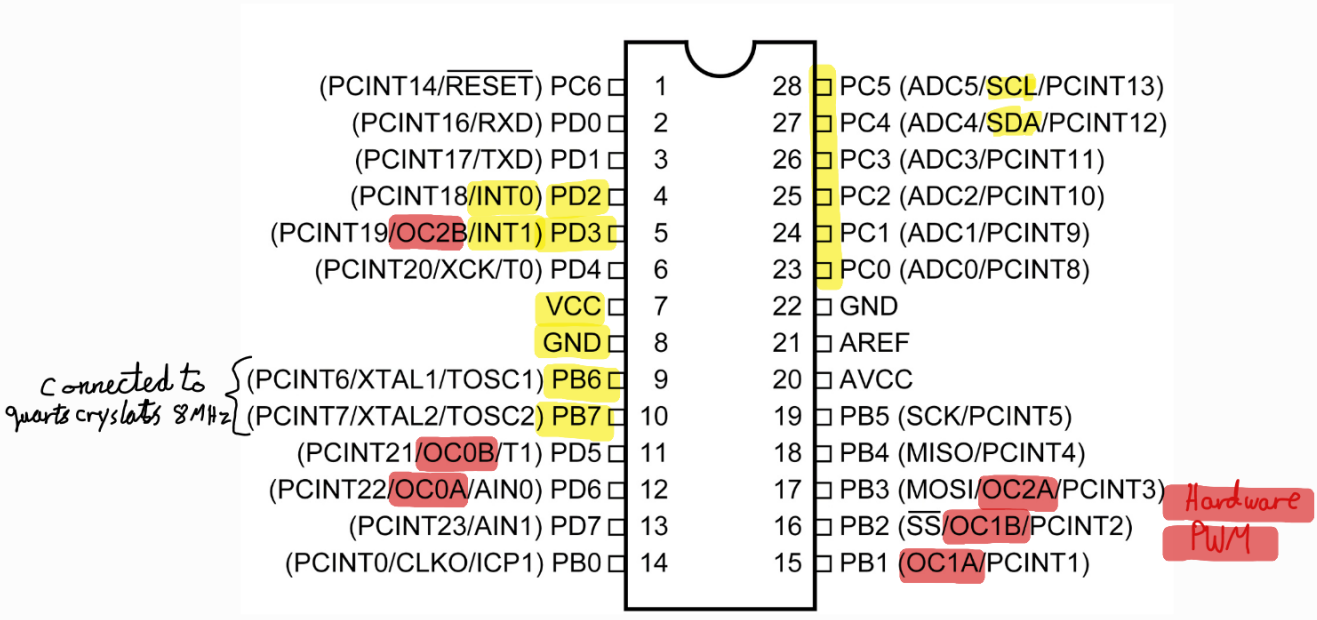
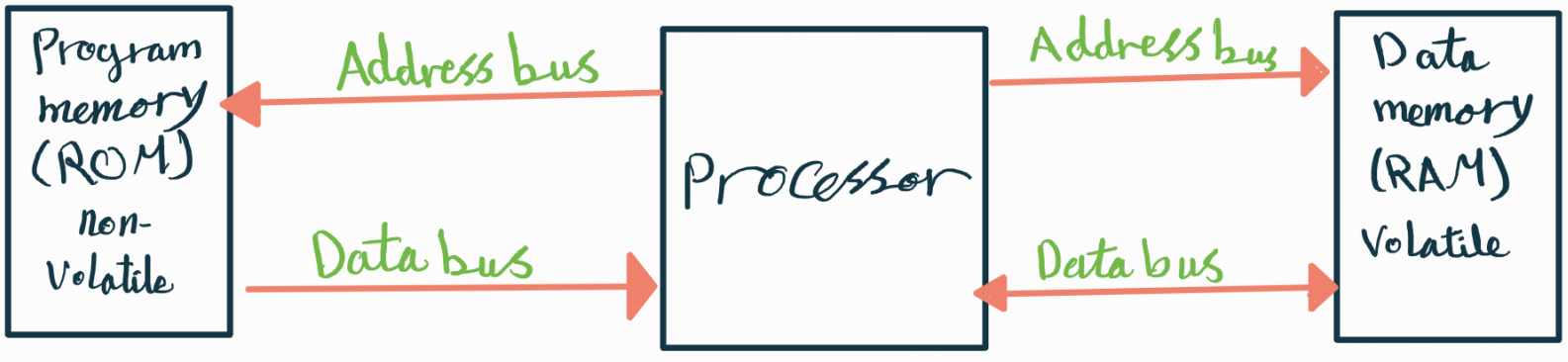
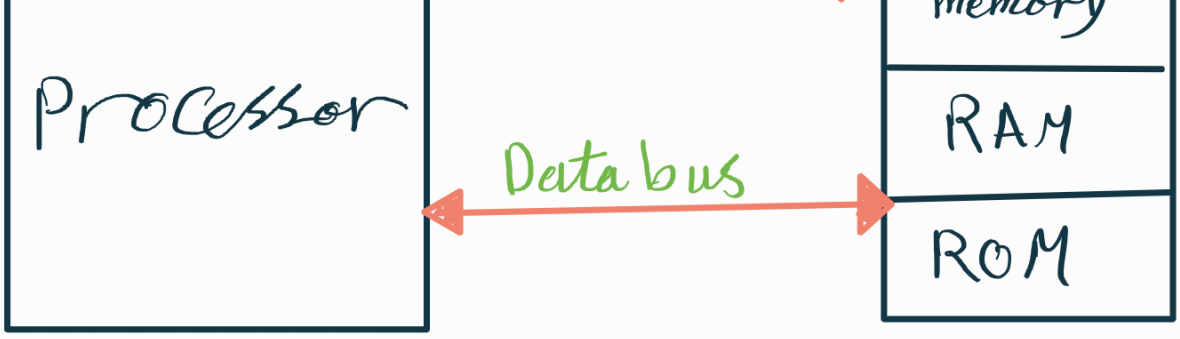


Harvard Architecture: It stores machine instructions & data in separate memory units that are connected by different buses. Thus, each of them has its own data & address bus.



Princeton Architecture (Von Neumann): Has a singal memory system, where several bus cycles are needed to get a complete instruction with the required data, thus it's slower than Harvard.



Microcontroller:

- cheap
- System on chip (SoC)
- All parts on a single chip
- Components can't be modified
- Lower clock speed (1 MHz - 300 MHz)
- Consumes less power
- Perform specific task
  - Thus, the relationship between input & output are predefined.

Microprocessor:

- has only CPU (other parts are required to function)
- high clockspeed (1 GHz - 5 GHz)
- Consumes more power.
- Size/number of external connections depends on the Purpose
- No specific tasks (thus, no predefined relationship between input & output).

Enable input at Pin B3:

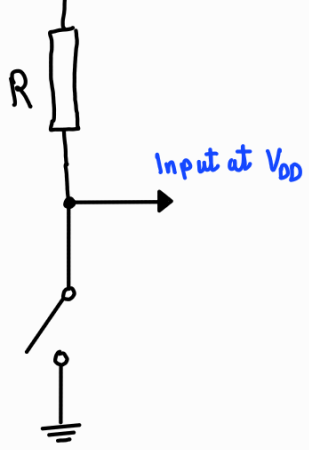
```
DDRB &= ~(1<<DDB3);
```

Pull up resistor enabled:

Switch is open

```
PORTB |= (1<<PB3);
```

Pull up resistor →

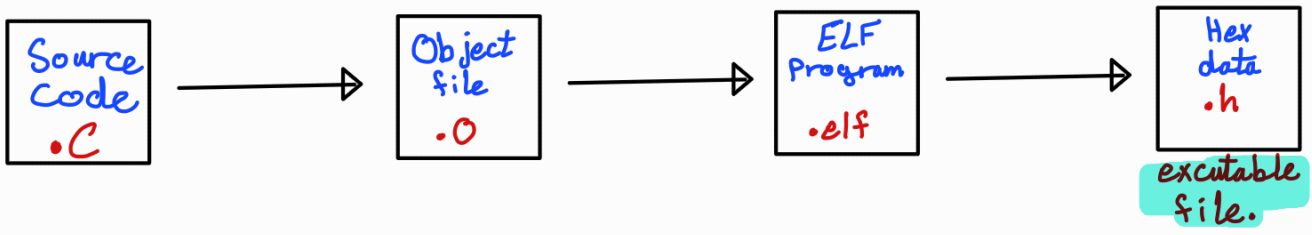


Enable output at pin D2:

```
DDRD |= (1<<DDD2);
```

```
if(~PINB & (1<<PB3)){
    PORTD |= (1<<PD2);}
else{
    PORTD &= ~(1<<PD2);}
```

WDT: (watchdog timer): To keep track of how the microcontroller is running & activate a handling routine in case it gets stuck.



Oversampling: number of samples = $2^{\wedge(2*n)}$ → num. of additional bits of resolution.

Q&A 6.4: For the maximum analog value of 5 V, provide the binary values of registers ADCH and ADCL with ADLAR set to 0 (deactivated) and 1 (activated).

ADLAR = 0 (regular case)	ADLAR = 1 ("left adjusted")
ADCH = _____11	ADCH = 11111111
ADCL = 11111111	ADCL = 11_____

The ADLAR bit affects the presentation of the ADC conversion result in the ADC Data Registers. Usually, the 10-bit ADC conversion result is right adjusted, the preceding six bits are zeros, i.e. max. ADCW value is 0b0000 0011 1111 1111 (consists of ADCH = 0b00000011 and ADCL =

0b11111111). When right adjusted (regular case), ADC value can simply be read over ADCW (C compiled code reads both registers and combines them, behind the scenes, into a 16-bit value) to get a proper 16-bit value with the low 10 bits filled.

ADC 5V → ADCW = 0b 11111111 11000000

If ADLAR bit in the ADMUX register is set, the result is left adjusted. In this case, the max. value of ADCH is 255 (0b11111111 or 0xFF), and the max. value of the lower ADCW byte ADCL is 192 (0b11000000 or 0xC0), due to the six trailing zeros, as ADC result is 10 bits only.

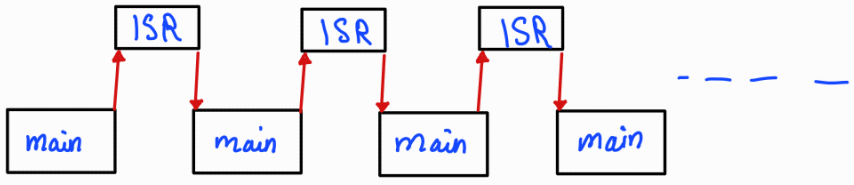
Left adjustment (ADLAR bit is set) is meant for applications that require only 8 bits from the ADC, thus getting all 8 of the most significant bits with no fiddling just by reading ADCH.

*ADCL must be read first (from right)!

ADMUX |=

MUX3...0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5

Interrupt: Is a signal to the processor, emitted hardware or Software indicating an event which needs immediate attention.



ISR: Interrupt Service Routine

*sei(); //enable ISR
*cli(); //disable ISR

Software Interrupts:

TIMER1_OVF_vect, TIMER1_COMPA_vect, SPI, ADC, USART

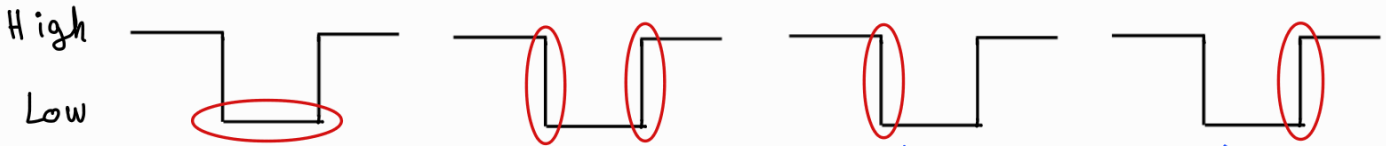
Serial Peripheral Interface

Universal Synchronous & Asynchronous Receiver Transmitter

External Interrupts:

INT0_vect, INT1_vect.

PD2 Pin 4 PD3 Pin 5



While Low (while pressed) | on falling & on rising (once pressed, & once unpressed) | falling edge | rising edge.

Timers/Counters (0, 2, 1)

8bits 16bits

$$f_{ovf} = \frac{f_{clock}}{(P * 2^{n(\text{num of bits})})}$$

P ← Prescaler

Check prescaler for a required frequency: $P_{min} = \frac{f_{clock}}{f_{req.} * 2^{n(\text{num of bits})}}$

→ Prescaler has to be more than P_{min}

$$TCNTX = \frac{f_{clock}}{f_{req.} * P} \cdot \text{Duty cycle}$$

(0, 1, or 2) ←

or $TCNTX = \frac{\text{required delay time}}{T_{clock}} - 1$

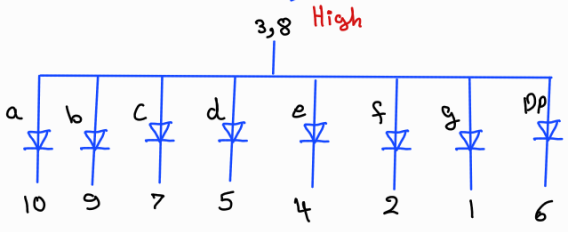
Max. time period of Timer 2 or 0, without counting overflows: devide f_{clock} by maximum available Prescaller (1024 in our case) → $\frac{8 \times 10^6}{1024} = 7812.5$

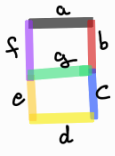
then $T_{max} = \frac{2^{8 \rightarrow \text{num. of bits}}}{7812.5} \rightarrow \simeq 0.0328\,s$

Seven Segements:

SA39 (Active low) → means it's active when connected to Low

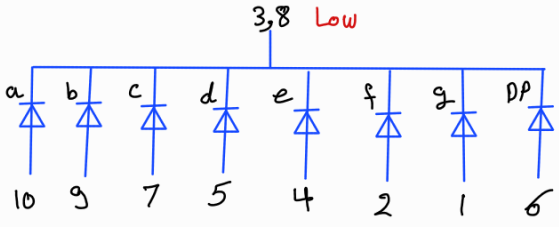
(Common Anode)





SC39 (Active High) → means it's active when connected to high

(Common Cathode)



I2C : Inter Integerated Circuits. (slave address D0)

I2C bus: Was devoloped to realise the serial communication between integerated circuit devices.

I2C has 2 wires (3 with ground): Serial Data (SDA) Pin27, & Serial clock (SCL) Pin28

Types of communication:

1- Serial: Data is sent as a stream of bits, only one at a time

2- Parallel: Many bits are sent at the same time.

EEPROM (Electrically Erasable programable Read-Only Memory) (slave address: A0)

Power & RTC

Batteries:

1- Nickel-Metal Hydride: 1.2 V

2- Zinc: 1.4V

3- Alkaline-MnO_2: 1.5V

4- Silver Oxide: 1.5V

5- Lead acid: 2 V

NiMH & NiCd are the weakest

6 - Lithium - MnO_2 : 3V

RTC: Real Time clock (32768 Hz Crystals)

What is BCD?

Binary coded Decimal: is a System of writing numerals that assigns a four-digit binary code to each digit 0 till 9 in a decimal form.

0 = 0000
1 = 0001
2 = 0010
3 = 0011
4 = 0100
⋮
9 = 1001

93 = 1001 0011
(9) (3)

* From 0X08 till 0X 3F (56 bytes * 8 bits) are RAM inside RTC.

encoding: input (in) is binary, output is BCD:

```
return ((in/10)<<4) | (in%10);
```

82

decoding: input (in) BCD, out binary: (1000 0010)

```
return ((in >>4)*10) + (in & 0b1111)
```

Temperature data Logger

Oversampling: When higher resolution is needed.

UART: Universal Asynchronous Receiver Transmitter.

Baud rate: is the number of symbols transferred per second.