PD215 Mechatronics

Week 2

Digital Hardware and Microcontrollers

Digital Hardware and Microcontrollers

- Digital hardware can preform multiple functions in mechatronics systems
  - Signal acquisition and processing
  - System monitoring and control
  - Switching
  - Information display
- In binary logic a variable can take only one of the two states (True or False, 1 or 0)
- Yet a digital device may have to process both logical and numerical data.
- Binary number system is used to perform numerical operations

Types of digital devices

- Combinational logic devices
  - Present inputs completely (and uniquely) determines the present outputs without using any past information (history) or memory.
- Sequential logic devices
  - Output depends on some form of past values of the inputs as well as the present values.
  - Some form of memory would be needed.

Logic Gates

- OR Gate
- AND Gate
- NOT Gate

- NAND Gate
- XOR Gate
- NOR Gate

Design of Logic Circuits

- Identify the input and output
- State the logic as Boolean relation
- Minimize or optimize the relation
- Using basic logic gates sketch the realization

Example

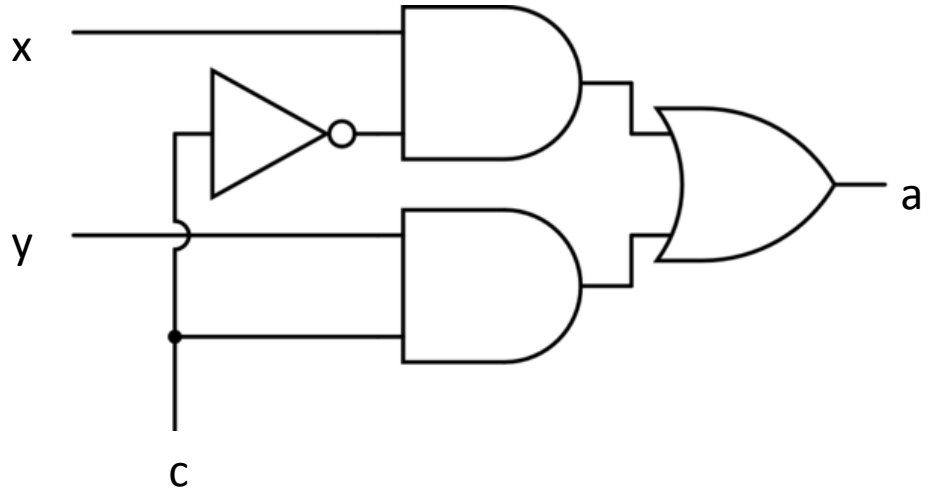
- Multiplexer Circuit
  - Digital multiplexer selects one digital input from a group of input channels based on control signal

Let us say x,y are input channels, a is output, and c is the control signal.

Logic statements

a=x if c=0

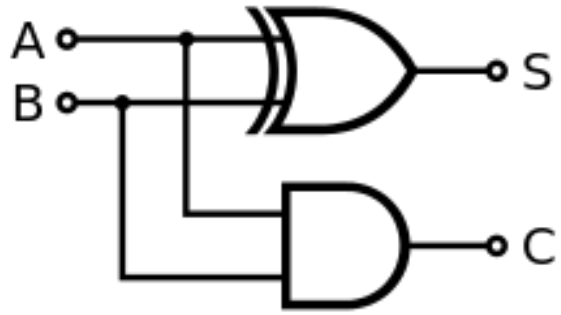
a=y if c=1



$$a = x.\bar{c} + y.c$$

Example 2

Adder Circuits

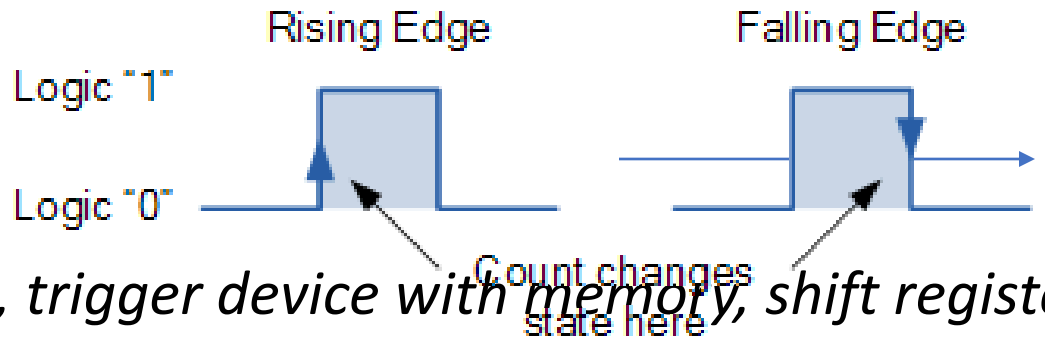


A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

'Half adder'

Sequential Logic Circuits

- Sequential logic circuits depend on past history and timing of the input



e.g counters, trigger device with memory, shift registers, latches, flip-flops

- *Synchronous* operation use a clock signal
- *Asynchronous* operation uses changes in the signal

Flip flops

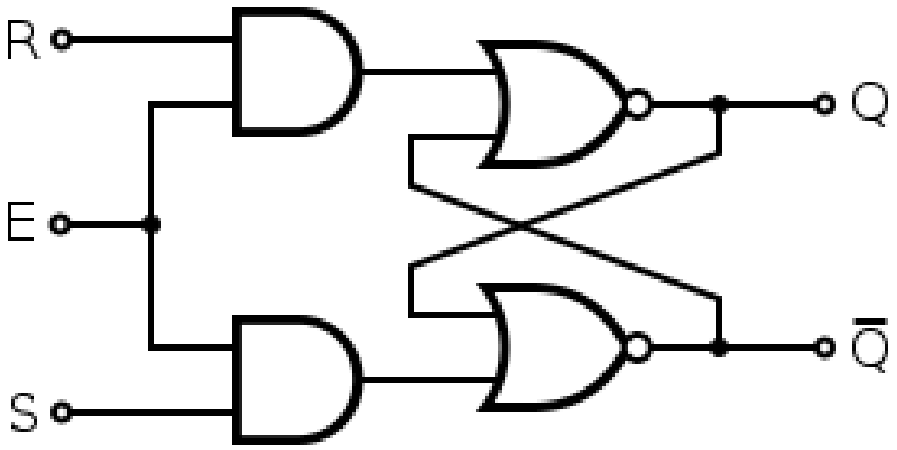
Flip flops are bi-stable devices i.e. the out is either 1 or 0

- RS, JK, D flip flops

Latches

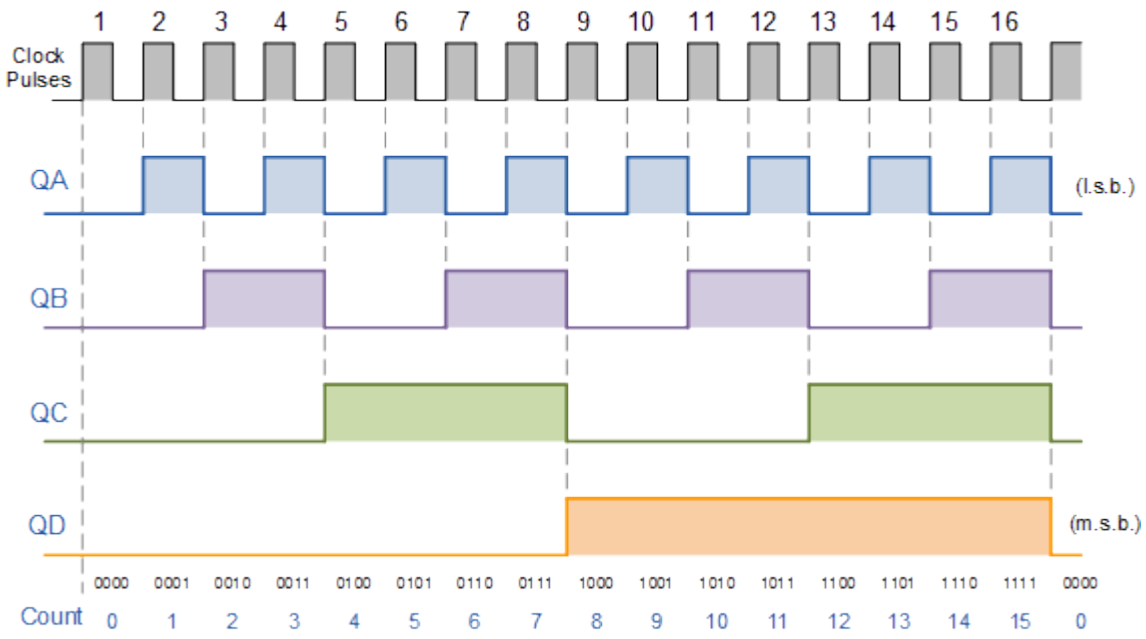
Retains previous state

until reset



A gated SR flipflop circuit diagram constructed from AND gates (on left) and NOR gates

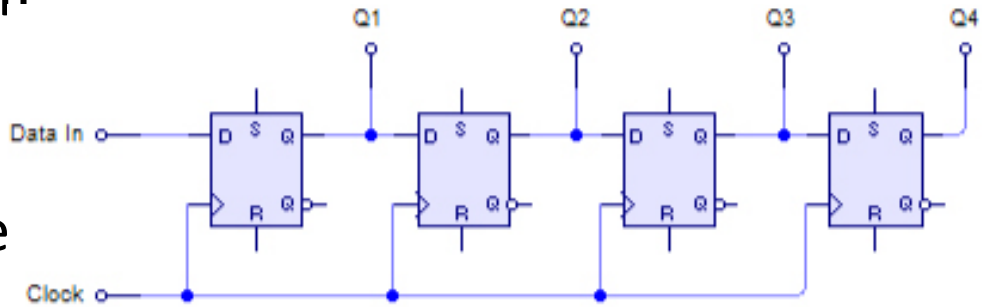
Counters


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Clock Pulses
QA
(l.s.b.)
QB
QC
QD
(m.s.b.)
0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111 0000
Count 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0

Shift Register

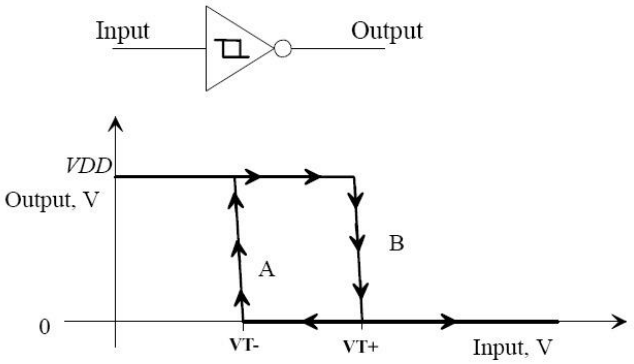
A shift register shifts the stored data in a word one bit at a time.

- Right shift multiples the value by 2 in binary numbe system



Schmitt Trigger

- Triggering element with Hysteresis
- Noisy signal near switching threshold leads to chatter which can be eliminated by Schmitt trigger



Transfer characteristics of a Schmitt trigger

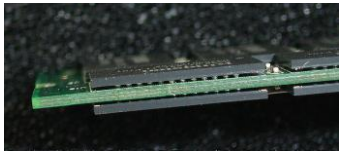
Integrated Circuits and Packaging

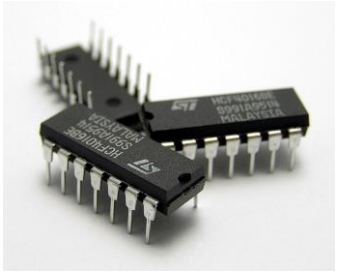
Using modern semiconductor fabrication technology it is possible to have over 10^6 logic gates in a single device.

-VLSI (very large scale integration) 10^4- 10^6 gates

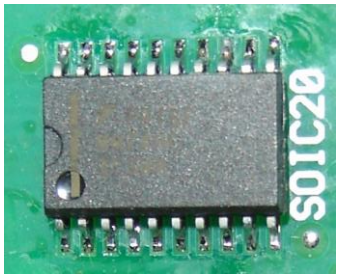
-Because of monolithic manufacturing it is not possible to repair single failed transistor

IC can be packaged in various form factors:



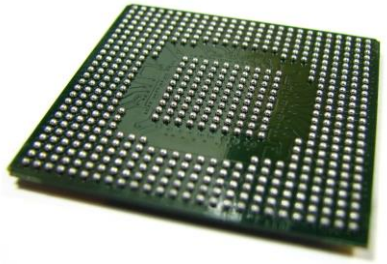


Dual Inline package (DIP)



Small Outline Integrated Circuit (SOIC)

Quad Flat Package (QFP)

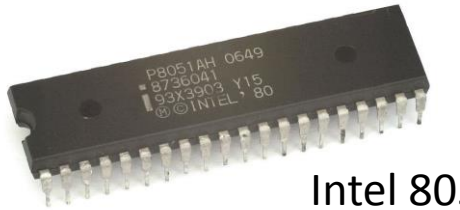


Balls Grid array (BGA)

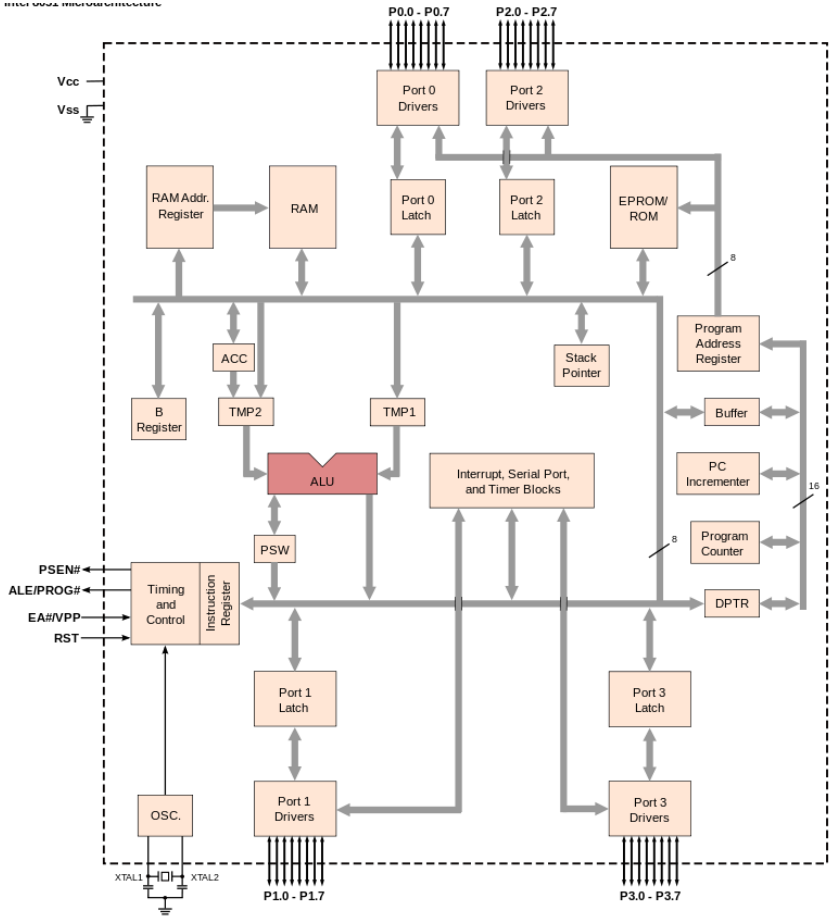
Microcontrollers

Microcontroller is dedicated special purpose digital compute device that is typically ‘embedded’ in application system.

- *Central Processing Unit /microprocessor*
- *Memory for storing data*
- *Input/output circuitry (I/O)*



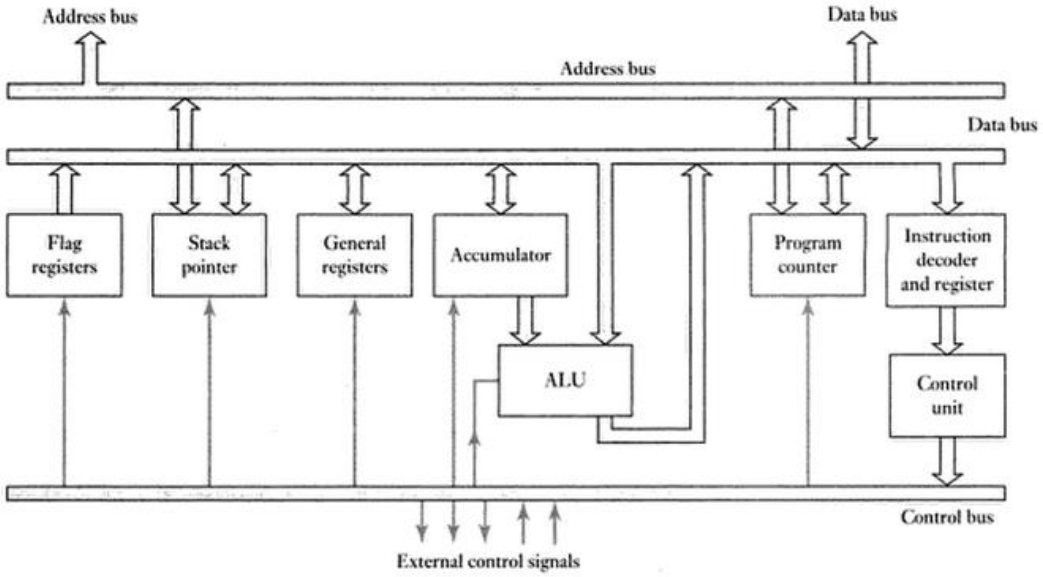
Intel 8051



Microprocessors

Brain of the microcontroller

- Arithmetic Logic Unit (ALU)
- Registers
- Control Unit



General architecture of a microprocessor

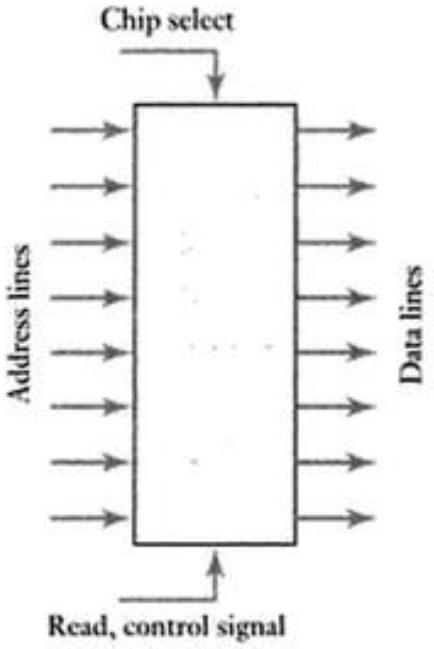
Buses: parallel conductors along which electrical signals are carried
Data Bus, Address Bus, Control Bus

Memory

Non-Volatile memory

- ROM (Read Only Memory)
- PROM (programable ROM)
- EPROM (erasable and programable ROM)
- EEPROM (electrically erasable and programable ROM)

Program stored in ROM is called ***firmware***



ROM

Volatile memory

- RAM (Random Access Memory
- DRAM and SRAM

Program stored in RAM is called ***software***

Software can be loaded in to RAM from other storage devices e.g. Hard disk, floppy or keyboard

Input / Output devices

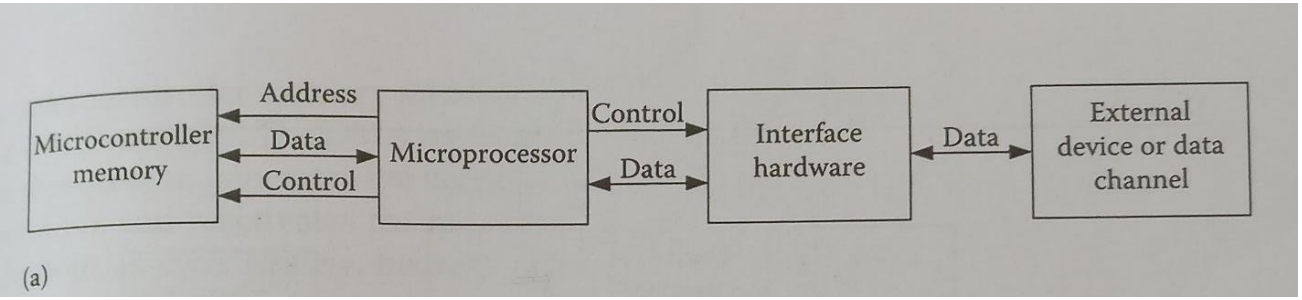
For transfer of data between microprocessor and external world. '**Peripheral devices**' exchange data with the microprocessor.

Programable I/O -polling used by microprocessor to read write data via these I/O

Digital Input/Output

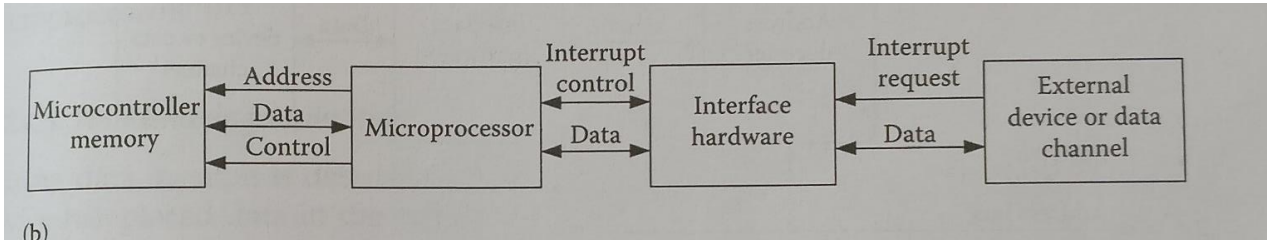
Pulse width modulation (PWM)

Analog Input (via ADC)

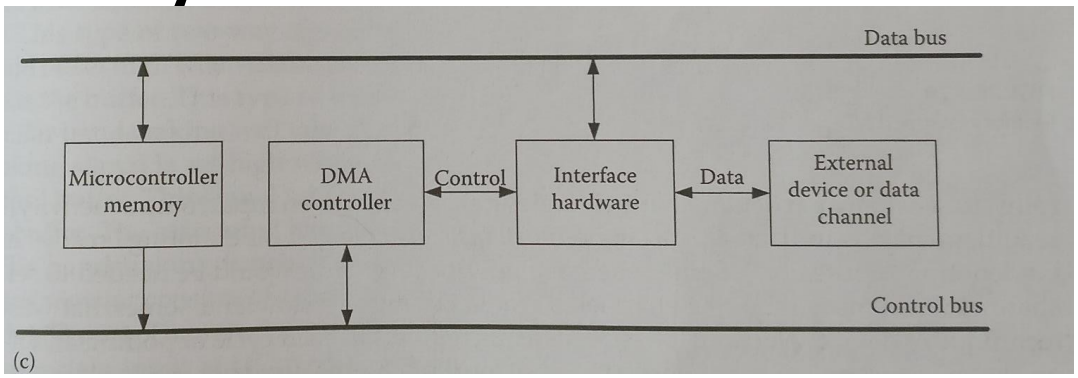


Input / Output devices

Interrupt I/O – Peripheral device sends interrupt request signal to microprocessor. Microprocessor suspends execution of current activity, performs I/O and returns to original (interrupted activity)



Direct Memory Access DMA -



Selecting a Microcontroller

- Number of input/output pins
- Interfaces required
- Memory requirement
- Number of interrupt required
- Processing speed required

Programing a Microcontroller Application

- Define the problem
- Define the **algorithm** to be used
- Represent the algorithm with **flow chart or Pseudocode**
- Translate flow chart/algorithm into instruction microprocessor can process. Assembly language or C conde is converted into **machine code**
- Test and ‘debug’ the program

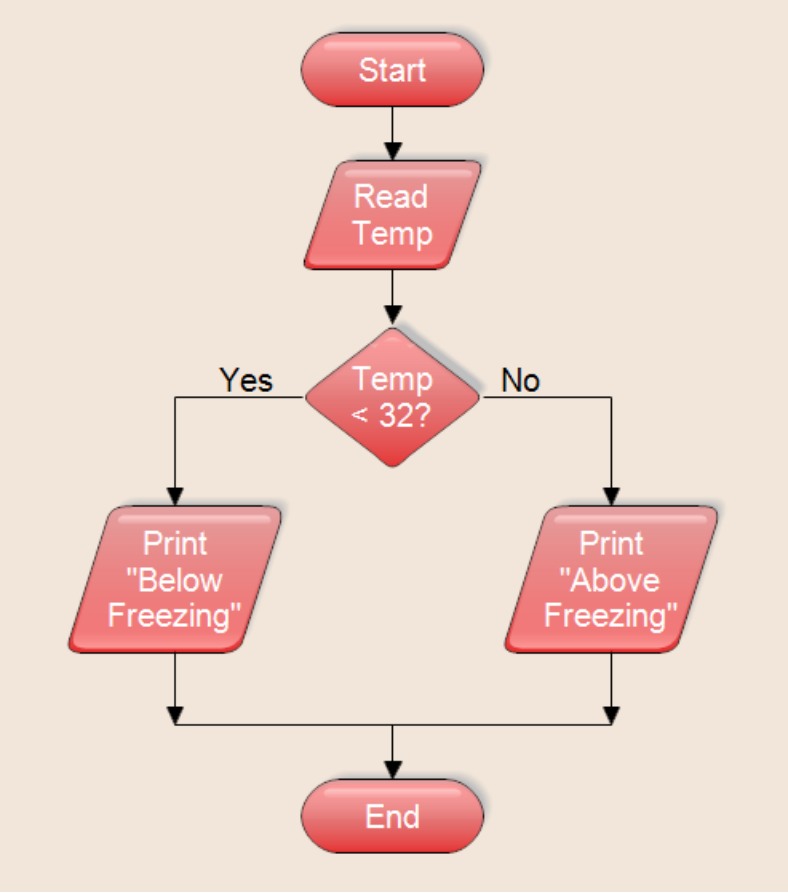
Algorithm

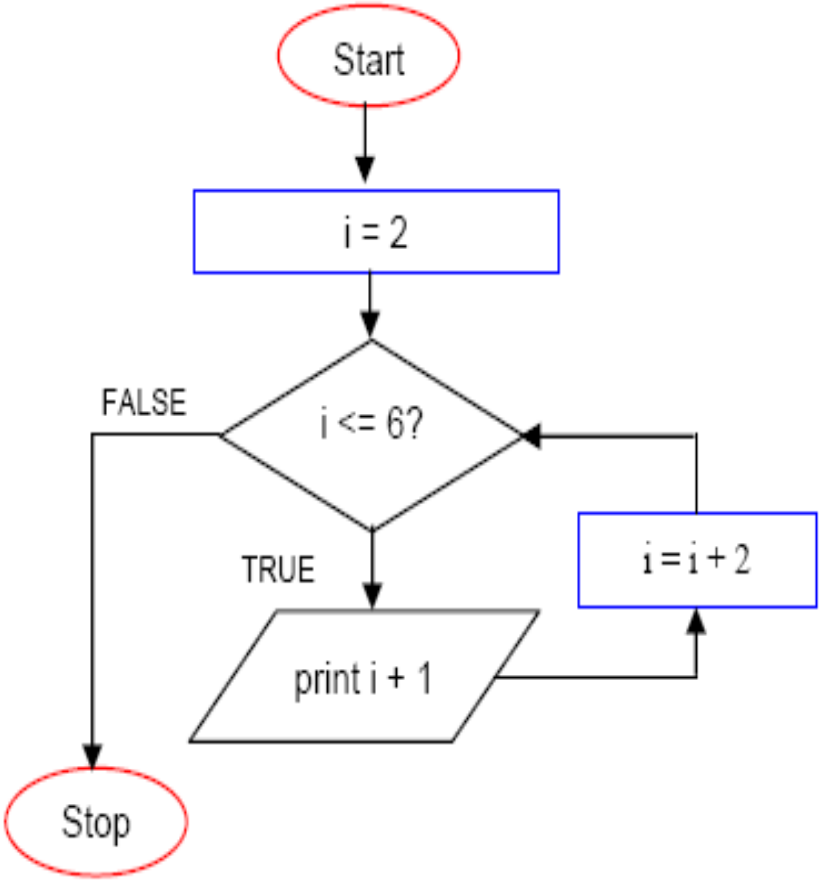
Set of process steps or rules to be followed for solving a problem

Flow Chart can be used to visually represent an algorithm

Start/Stop Process Decision Input

Program flow direction


Start
Read Temp
Temp < 32?
Yes
No
Print "Below Freezing"
Print "Above Freezing"
End


Start
i = 2
i <= 6?
FALSE
TRUE
print i + 1
i = i + 2
Stop

Pseudocode

Write program as sequence of operations of functions with decision steps

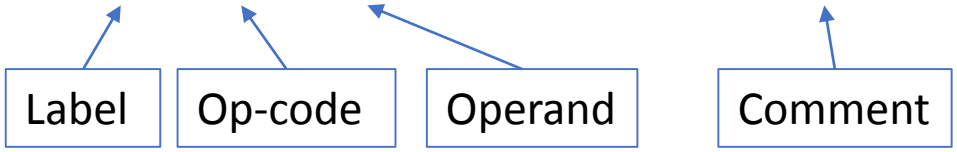
```
BEGIN
        INITIALIZE I =2
        IF I<=6
                PRINT I
                INCREMENT I BY 2
        ELSE
                NOP
STOP
```

Assembly Language

Microprocessor specifically needs ***byte codes*** corresponding to each action.

In assembly language ***mnemonics*** are used represent specific byte code followed by memory locations or data it should use

```
110   LDAA, $0110 ; load
120   ADDA, #10h ; adds 10 hexadecimal to number in Accumulator A
130   DEC, R3       ; Decrement register R3 by 1
```



8051 program

```
                ; Addition of two numbers

NUM1    EQU     20H     ; location of number 1
NUM2    EQU     21H     ; location of number 2
SUM     EQU     22H     ; location for the sum

        ORG     8000H   ; address of start of user RAM
START   MOV     A,NUM1  ; load number 1 into acc. A
        ADD     A,NUM2  ; add number 2 to A
        MOV     SUM,A   ; save the sum to address 22H
        END
```

PIC program

```
                ; Addition of two numbers

Num1    equ     H'20'   ; location of number 1
Num2    equ     H'21'   ; location of number 2
Sum     equ     H'22'   ; location for the sum

        org     H'000'  ; address of start of user RAM
Start   movlw   Num1    ; load number 1 into w
        addlw   Num2    ; add number 2 to w
        movwf   Sum     ; save the sum H'22'
        End
```

Assembly language Instruction sets

Data transfer/movement

Load, Store, Move, Clear

Arithmetic

Add, Decrement, Increment, Compare

Logical

ADD, OR, Exclusive-OR, Logical Shift, Arithmetic Shift, Rotate

Program Control

Jump, Branch, Halt/Stop

Subroutines

Block of program might be used a number of times

- As subroutine may be called number of times there is need to store the program counter in Last In First Out (LIFO) manner.
- Delay subroutine can be used for timing the operations of a microcontroller

---

Lookup Tables

- Indexed addressing can be used to lookup values in a table
- Examples: square of integers can be stored as a table and directly used as needed instead of doing the arithmetic

C Program

C is a ***high level*** programing language which is often used instead of Assembly language

Assembly language is different for different microcontroller while C language is standardized.

main() program is called as an entry point and various subroutines are called from within that

Extensive programing feature are available:

- Branches and Loops
- Pointers
- Header files and libraries of functions

Arduino

The Arduino project started in 2003 as a program for students at the *Interaction Design Institute Ivrea in Ivrea, Italy*

Combination of hardware and software to enable low cost and easy way to build devices that interact with the environment via sensor and actuators

Arduino IDE (Integrated Development Environment) supports c and c++ style programing

setup()

loop()

Thursday lab session

Some practical exercises with Arduino hardware and software

- Create binary clock
- Interrupt and polling switching

Please bring Arduino development board, Laptop with Arduino IDE, breadboard, LEDs and resistor kit