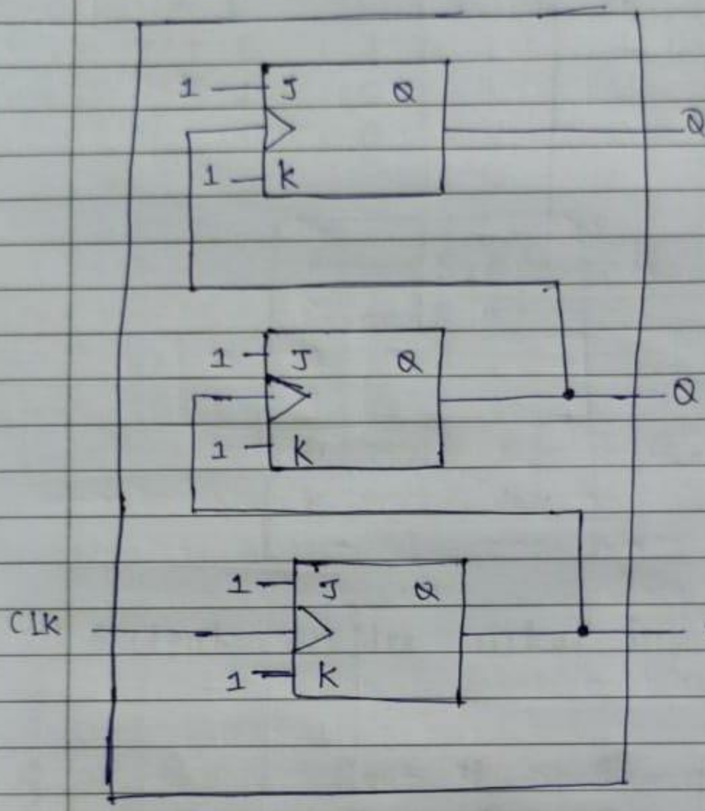
Name - Khushi Nitinkumar Patel.
PRN - 2020BTECS00037.
Branch - CSE.

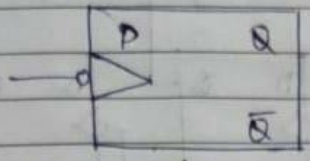
- Asynchronus counter -

Clock is given only to the least significant flip flop, in an asynchronus counter.



| $Q_2$ | $Q_1$ | $Q_0$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 0 | 0 |

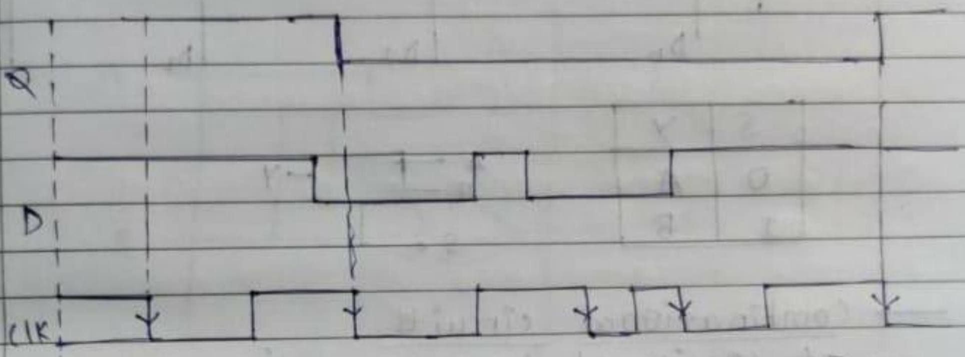
- Shift Registers :-



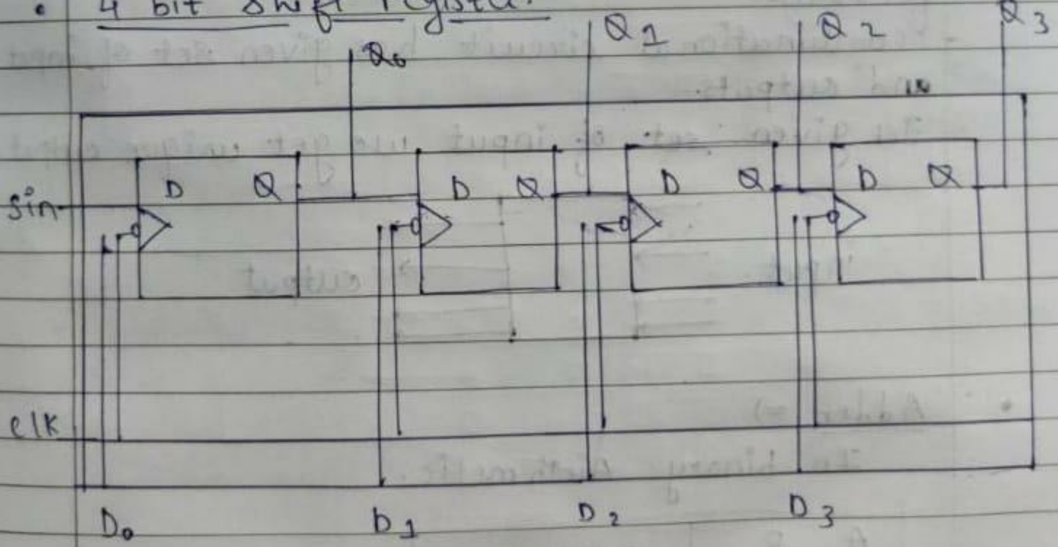
D latch

Truth table :-

| Clk | Q |
|---|---|
| 0, 1, ↑ | Last state |
| ↓ | D |

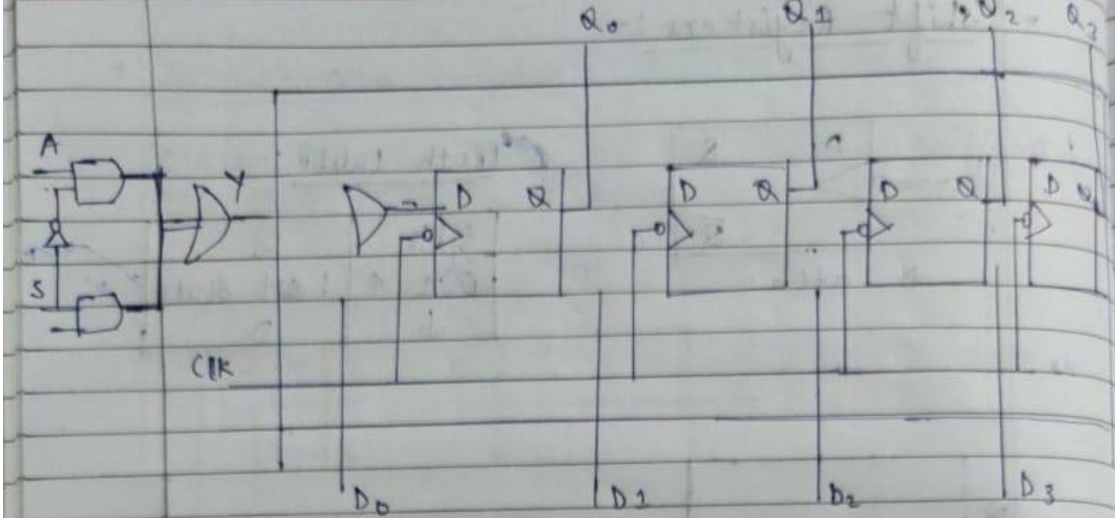


- 4 bit shift register.

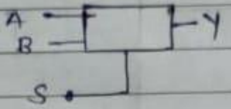


sin = 0

| Q3 | Q2 | Q1 | Q0 |
|---|---|---|---|
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 |



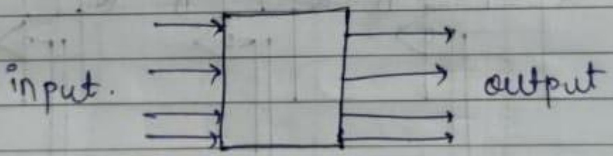
| S | Y |
|---|---|
| 0 | A |
| 1 | B |



→ Combinational circuits.

- It consist of innerconnection of gates to each other and cascading them with no feedback.
- combinational circuit has given set of input and outputs.

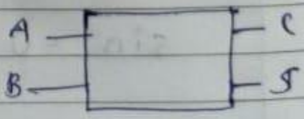
For given set of input we get unique output.



- Adder =)

In binary Airthmetic..

| A | B | carry | sum |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1. |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

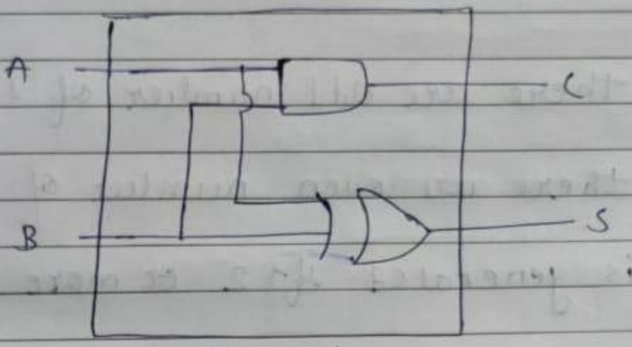


Adding 2, 1 bit number

explaination :

$$\begin{array}{cc} & 0 \\ + & 0 \\ \hline 0 & 0 \\ \uparrow & \uparrow \\ c & s \end{array} \quad \begin{array}{cc} & 0 \\ + & 1 \\ \hline 0 & 1 \end{array} \quad \begin{array}{cc} & 1 \\ + & 0 \\ \hline 0 & 1 \end{array} \quad \begin{array}{cc} & 1 \\ + & 1 \\ \hline 1 & 0 \end{array}$$

If both A & B are 1 then carry = 1

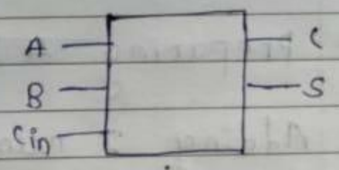


Half Adder.

- Adding 2, 2 bits number.

e.g

$$\begin{array}{ccc} & A_1 & A_0 \\ & B_1 & B_0 \\ C_1 & C_0 & \\ \hline C_1 & S_1 & S_0 \end{array}$$



- Addition of 2 bit number is 3 bit number
- Here we need full adder.

Truth table of full adder =)

| Cin | A | B | Cout | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

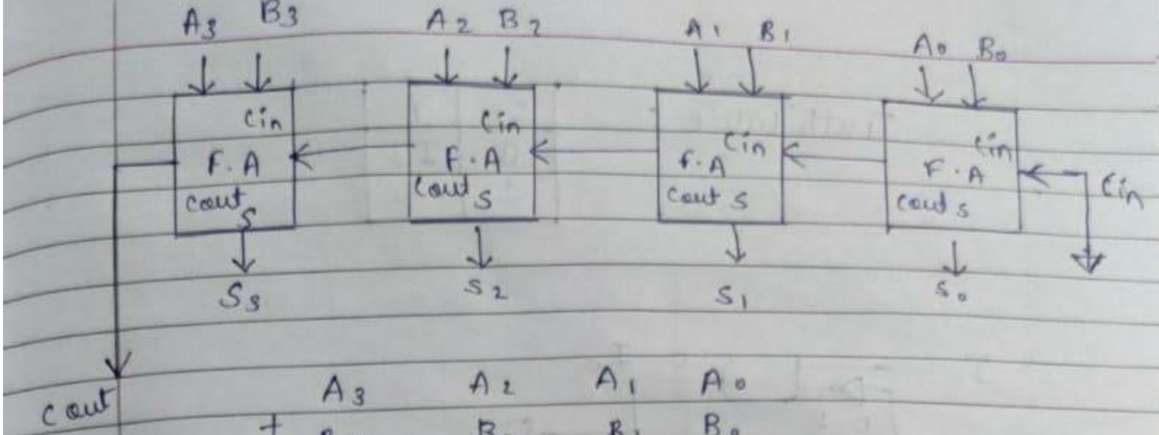
1) when there are odd number of 1's, sum is 1.
2) when there are even number of 1's, sum is 0.
3) carry is generated if 2 or more 1's are there.

- 2 half adder and 1 OR gat is used to prepare full adder.

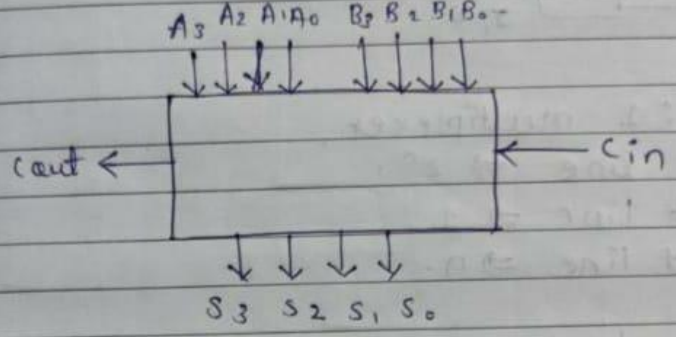
• Preparing 4 bit full adder :-

Adding 2 numbers of 4 bits.

```
e.g      0  1  0  1   ← A.
         1  1  0  1   ← B.
      1  1 . 0  1  Cin
      ______________
      1  0  0  1  0
```



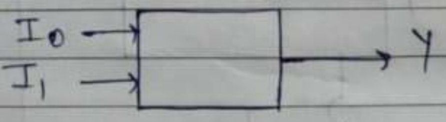
$$\begin{array}{lcccc} & A_3 & A_2 & A_1 & A_0 \\ + & B_3 & B_2 & B_1 & B_0 \\ \hline Cout & S_3 & S_2 & S_1 & S_0 . \end{array}$$



A 4 bit full adder comprises of 1 carry in, 1 carry out, 2 , 4 bit inputs A and B, 1, 4 bit output sum.

→ Multiplexer. ⇒ (many as to 1).

- A multiplexer consists of more than 1 inputs and only single output.

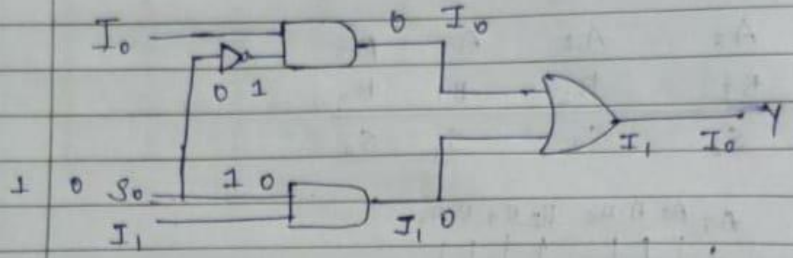


2:1 multiplexer.

- S0 is select line the decides which input should come on the output.

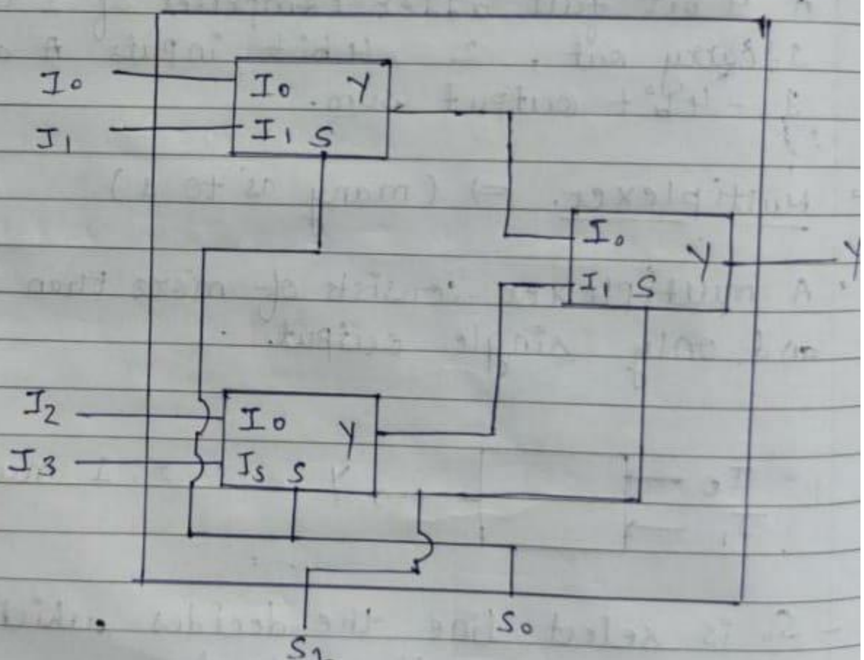
Truth table :-

| $S_0$ | Y |
|---|---|
| 0 | $I_0$ |
| 1 | $I_1$ |



- For $2^n : 1$ multiplexer,
  input line ⇒ $2^n$.
  output line ⇒ 1.
  select line ⇒ n.

- 4 : 1 Multiplexer



1) $S_1 = 0$, Top multiplexer O/P appears on final O/P.

2) $S_1 = 1$, Bottom multiplexer O/P appears on final O/P.

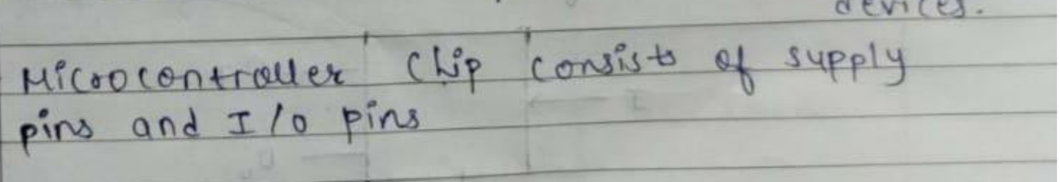
Truth table :-

| $S_1$ | $S_0$ | Y |
|---|---|---|
| 0 | 0 | $I_0$ |
| 0 | 1 | $I_1$ |
| 1 | 0 | $I_2$ |
| 1 | 1 | $I_3$ |

→ $S_0$ decides which output should appear on 1st layer multiplexer.
→ $S_1$ decides final O/P.

- Arduino board ⇒

- It consists of microcontroller chip, it is having interface with computer.



MVI A ⇒ 3E

Microcontroller = Microprocessor + Memory + I/O devices.

- Microcontroller Chip consists of supply pins and I/O pins

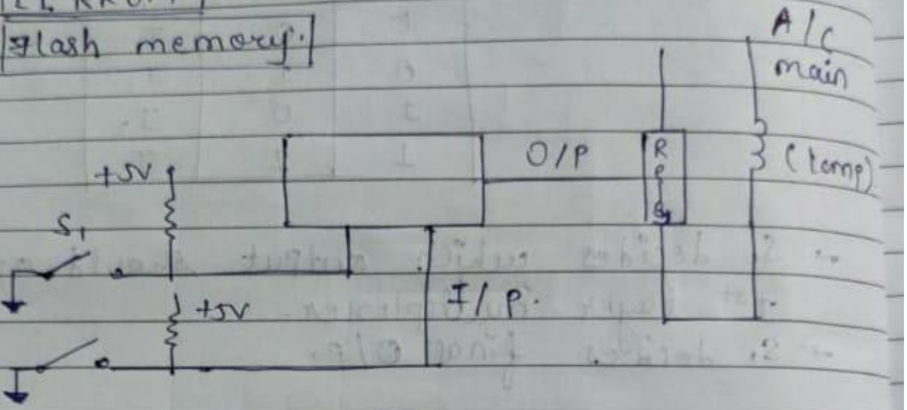
- Microcontroller chip consist of supply pins and I/O pins.

- Power on reset circuit ⇒ Reset the circuit and Microprocessor starts program from Address 0, which is present in memory.

Factory programmed.

EERROM

Flash memory.

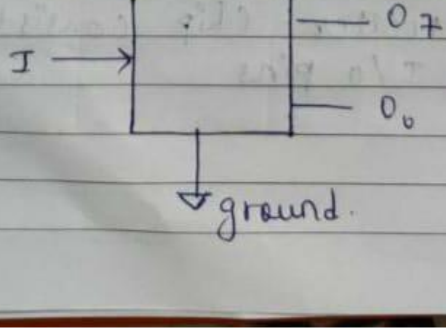


$y = S_1 \oplus S_2.$

→ Arduino board pins.

- Digital I/P pins → user control & Microcontro read.
- Digital O/P pin → Microcontroller controls it.

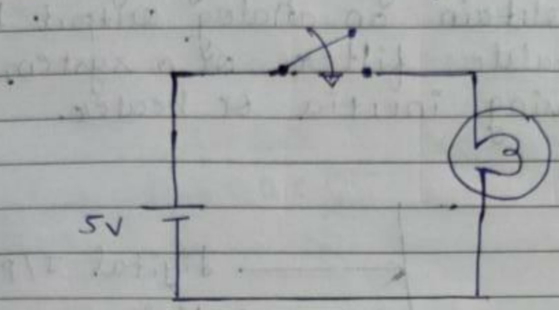
- Analog to Digital ~~controller~~ converter (ADC)=)



Input in range ~~of~~ 0 to 5v is converted into output 00-FF (80H)

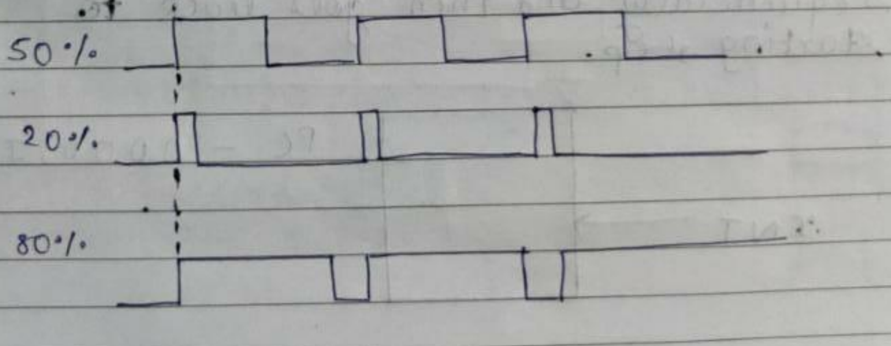
$\frac{5}{255}$ ⇒ 1 UNIT.

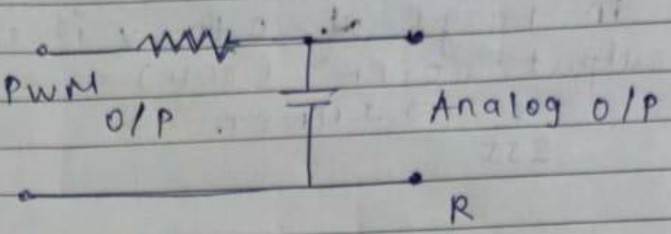
- Analog O/P pin is not available, but output is generator in another form i.e pulse width modulated form (PWM).



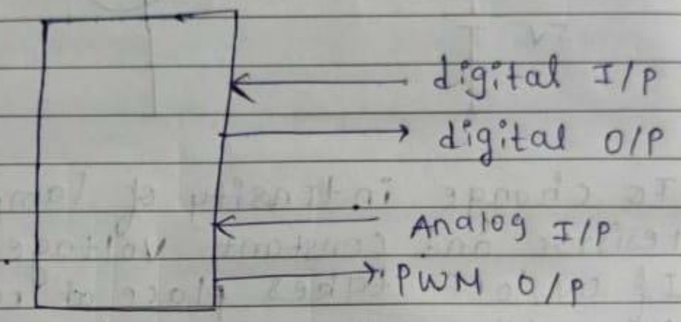
- To change intensity of lamp with no resistor and constant voltage supply.
- If ON/OFF takes place at rapid rate, then filament will be heated at average rate.
- Pulse width Modulation O/P generates a square a square wave where Duty cycle can be from 0% to 100%

Square wave ⇒





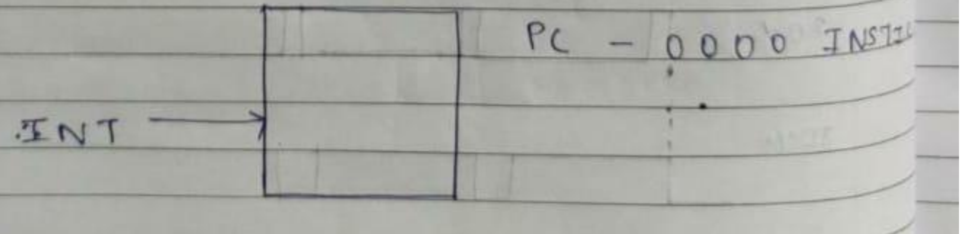
- PWM output pin is a digital output pin in which by controlling its duty cycle it is possible to obtain an analog output by putting suitable filter or a system which itself is having inertia or heater.



~~Ard~~ Arduino board

- Interrupt Inputs:

- A program on arduino board consist of sequence of instruction, which are executed sequencially and then goes back to starting step.



- Interrupt is an input pin which is checked after instruction, automatically.
- It doesn't consume any time.

```
        0000   INS0
           1   INS1      interrupt          Interrupt
           3   INS2      branch address     service routine.
INT →  {0006   INS3        8000  →  INS a1 Putoff Burner
                                        a2 GoBack
```

Scratch PAD.

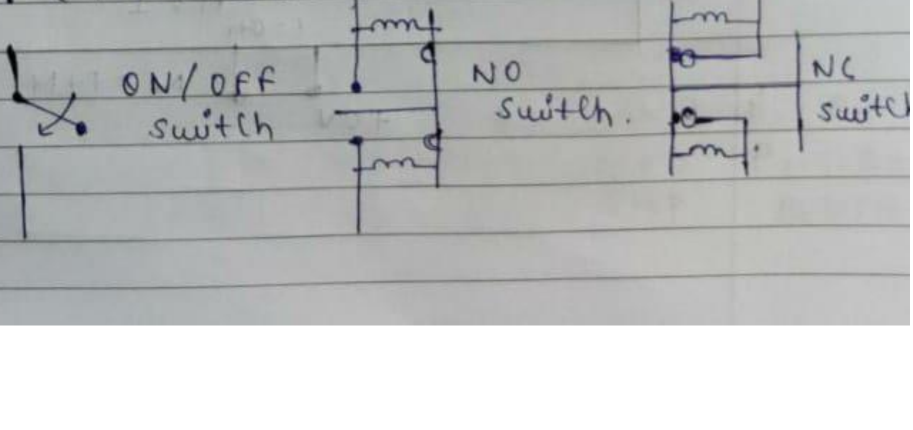
0006

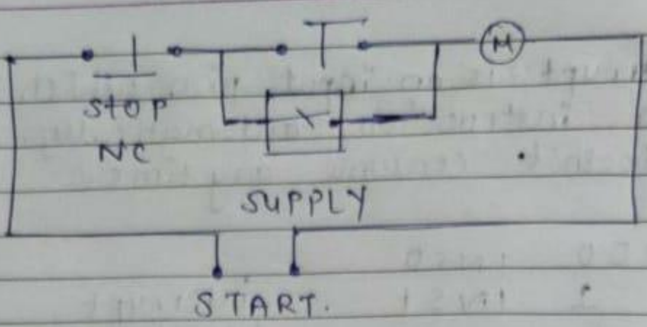
- Arduino programming.

General purpose Digital I/O pins
I/P pin → switch → Manual switch
O/P pin → Read P0.0
Input P1.1

- output pin will be made to logic '1' or logic '0' by micro controller.

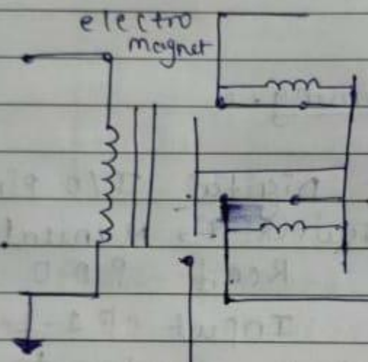
SET P1.2
RESET P2.1
WRITE (P1.3, FLAG)
OUTPUT (P3.2, 0).

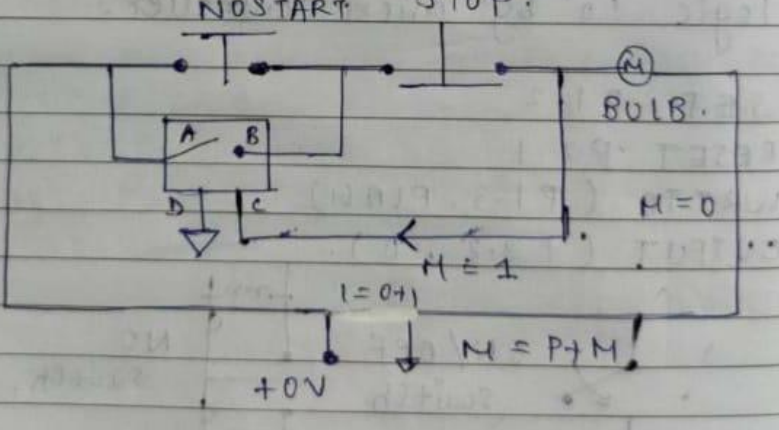




1) Press
⇓
MOTOR START and continues to run even if the start button is released

2) Press STOP
Motor stops and even if you release stop button, motor doesn't stop.



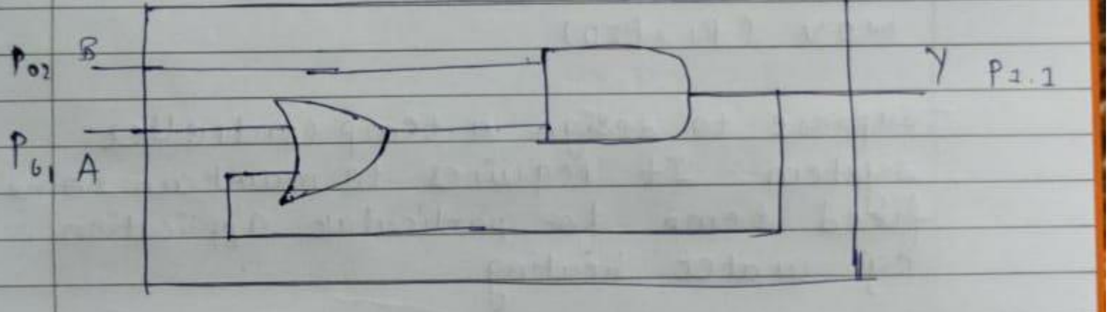


when start button is pushed, motor starts, bulb glows and relay. Using start button, motor can be started and using stop button motor can be stoped.

```
READ  START
READ  STOP
M = (P+M).Q̄
```

label  P = INPUT (START)
$\quad$ Q = INPUT (STOP)
$\quad$ $M = (P+M).\bar{Q}$

| STOP | START | M |
|---|---|---|
| 1 | X | OFF(0) |
| 0 | 1 | ON(1) |
| 0 | 0 | 0/1 |



$Y = C.B$
$Y = (A+Y).B$
$\underset{\overset{\uparrow}{A}}{B_0}$ $\underset{\overset{\uparrow}{B}}{B_1}$ $\underset{\overset{\uparrow}{Y}}{B_2}$ ⇒ internal registers.

```
BO = READ  PO.1
BEGIN : MOV  BO, PO.1      AND  B2, B1
        MOV  B1, PO.2      MOV  P11, B2
        OR   B0, B2        IMP  BEGIN
```

Microcontroller instructions are either airthmetic logic instruction or input / output instructions.

- Analog I/P, PWM O/P.

Analog I/P is there are inbuilt analog to digital converters and it is possible to read that pin.

5V ⟶ 2550

51 ⟶ 255

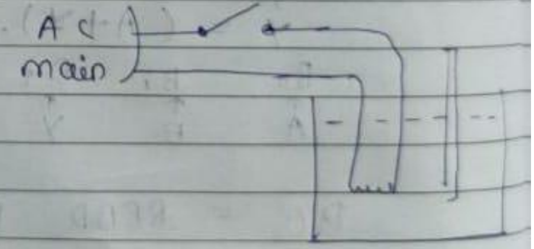
$\frac{5000\,mV}{250} = 20\,mV$

| 3 Bits | Analog. |
|---|---|
| P2.0 | 0 0 |
| P2.1 | 1 |
| P 2.3 | 1 |
| ⋮ | 1 |
| ⋮ | 1 |
| P 2.7 | FF |

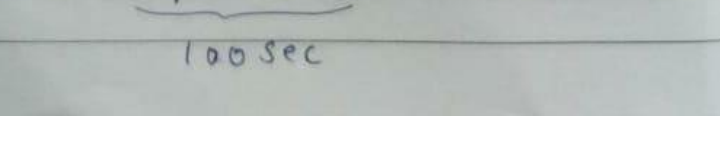
INPUT P2.0
MOV (R1, P20)

- Suppose to design a temp controller system. It requires to maintain some fixed temp for particular application eg water heating.

Maintain
fixed temp
which should be
preseattable



8 switches
00 - FF



(s.P)

user sets set point → Read by Arduino
system process (P.V) → Read by Arduino
changes variables.

SWITCH
O/P Pin ⇐ controlled by Arduino.
(READ)