Lecture-1
Wednesday
9/8/23

Overview

CSE 2203: Microprocessors and Micro controllers

Introduction + course profile

Lecture-2
Monday
14/8/23

Microcomputer History

Different microprocessor with features:

What is n-bit microprocessor?

microprocessor size = ALU size

Intel 4004:

→ 1971
→ 4-bit microprocessor
→ 4-bit data bus, 12-bit address bus
⇒ we use 4 wire lines for data computation.
⇒ n-bit address = 2^n address can be define.
→ 4-KB of memory (outside)
→ 45 instructions.

→ Problems:

① Speed

② Word width

Intel 8008 → Intel 8080 → Intel 8085 → 8086/8088

⇒ First modern 8-bit processor

⇒ last 8-bit μprocessor

*** Particular generation ৎ difference.

Microprocessor (***)

→ microprocessor and cpu are used interchangeably.

→ A μprocessor is a multipurpose, programmable, clock-driven, register based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input and process data according to those instructions and provides results as output.

Lecture-3

Wednesday

23/8/23

Introduction to Microprocessor

Assignment: Lecture-3 Topics

→ Submission: Next class

Why we need to learn microprocessor?

General purpose, no special purpose, we use it in any device.

Difference between μprocessor & μcontrollers:

Microprocessor	Microcontroller
1. CPU stand alone, RAM, ROM, i/o separate.	1. CPU, RAM, ROM, memory integrated in one single chip.
2. Designer can decide on the amount of ROM, RAM & i/o ports.	2. Fixed amount of on-chip ROM, RAM, i/o ports.
3. Expensive	3. For application in which cost, power are critical.
4. General purpose.	4. Single purpose.
5. High processing power.	5. low
6. High power consumption	6. low
7. Customizable	7. Not customizable.

8086 Architecture

1. Clock Speed:

→ Per second এ কতগুলো clock cycle produce করতে পারে, তাই clock speed.

→ Fasten the clock speed, Fasten the processing speed.

* 800 MHz = 800×10^6 clock cycles.

2. Uses of tramistors in computer:

→ Building Block of computers

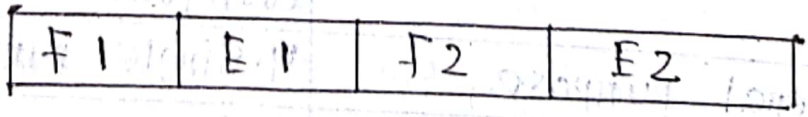
→ memory location = pair of tramistors.

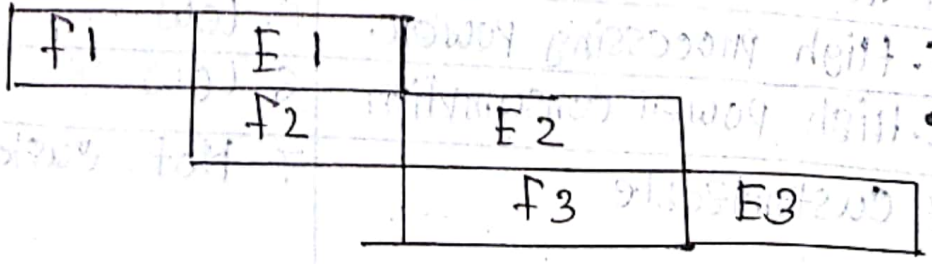
3. Moore's law:

→ integrated circuits double every two years.

4. 8088/86 pipelining:

8085 not pipelined μprocessor.





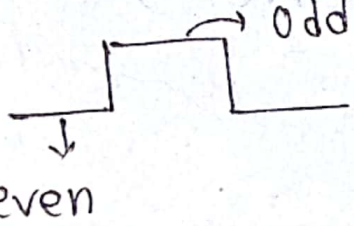
8086 microprocessor :

Features :

1. 20 bit address bus, 16 bit data bus.
2. multiplexed address and data bus.
   ↳ একই wire but অনেক কাজ করতে পারবে।
3. +5v needed
4. 40 pin dual in package

address bus } same bus দিয়ে
data bus } কাজ করবে।

* NMI → Non Muskable interrupt.



8086 internal Architecture : Draw + explain (***)

↓
Block Diagram

Main Unit :

① BIU

② EU

Registers:

*** Flag Registers

0101 0100 0011 1001
0100 0101 0110 1010

1001 1001 1010 0011

Sign flag, SF = 1 (MSB = 1 হলে 1, না হলে 0)

Parity flag, PF = 1 (even no 1)

Carry flag, CF = 0

Auxiliary flag, AF = 1 (First nibble এ carry এসেছে 1)

Zero flag, ZF = 0

Lecture-4
Wednesday
30/8/23

8086 Microprocessor

Bus Interface Unit (BIU):

→ Contain 6 byte instruction queue.

→ Segment Register (CS, DS, ES, SS) → Special purpose register

→ Summation Block, Σ = offset +

→ Instruction point ⇒ current instruction location address.

* Prefetched bytes in a FIFO called Queue.
* EU is ready then call instruction from queue.

Pipelining

* pipelining is kind of parallel processing.

Memory Segmentation

→ 20 bit address bus → can address 2^{20}

* OFFSET size 16 bit

*** Instruction এর physical address কীভাবে calculate করা যায় ?

Starting address 10000 (hardware 0 / code segment)
offset 0029 / IP
10029

Example:

If DS=7FA2H and the offset is 438EH

ⓐ physical address

7FA20
438E
83DAE

ⓑ Lower Range: → starting

7FA20 + 0000 = 7FA20

ⓒ Upper Range of DS: → Ending

7FA20 + FFFF = 8FA1F

ⓓ Show the logical address:

7FA2:438E

Lecture- 5
wednesday
13/9/23

Programming languages

* Machine language → Binary (0,1)
* Assembly "
* High-level "

Machine language:

⇒ Deals with machine directly ⇒ No need for translation
⇒ Fast ⇒ No extra space needed.

Assembly language:

⇒ Uses Hemonies (MUL, ADD, MOV, DIV)
⇒ Faster than high level language.

4 Parts:

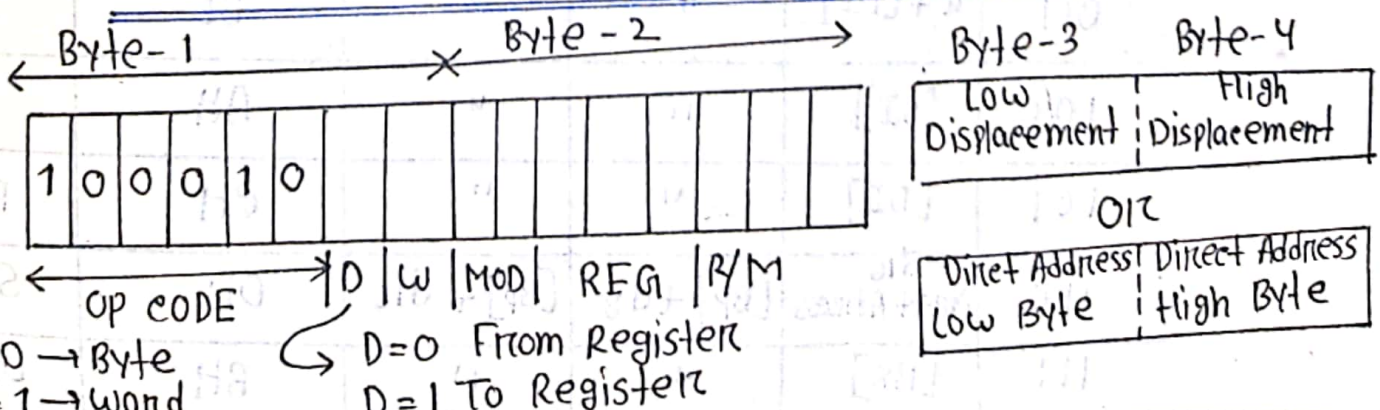
① level ② Opeode ③ Operand ④ comments

② tells the Processor what to do.

③ কার উপর operation করবে।

High-level language: More closer to human.

MOVE Instruction Coding Format and Examples



Register		CODE
W=0	W=1	
AL	AX	000
BL	BX	011
CL	CX	001
DL	DX	010
AH	SP	100
BH	DI	111
CH	BP	101
DH	SI	110

Register Tramfer

Memory Tramfer

MOD / R/M	00 (No Dis)	01 (8 bit Dis)	10 (16 bit Dis)	11 (Reg mode, no Dis)	
				W=0	W=1
000	[Bx]+[SI]	---+d8	---+d16	AL	AX
001	" + [DI]	"	"	CL	CX
010	[BP]+[SI]	"	"	DL	DX
011	" +[DI]	"	"	BL	BX
100	[SI]	"	"	AH	SP
101	[DI]	"	"	CH	BP
110	d16 Direct Address	[BP] +d8	[BP] + d16	DH	SI
111	[BX]	"	"	BH	DI

MOV SP, BX

OPCODE = 100010
D = 0, W = 1
MOD = 11
REG = 011
R/M = 100

MOV SP, BX

OPCODE = 100010
D = 1, W = 1
MOD = 11
REG = 100
R/M = 011

MOV CL, [BX]

OPCODE = 100010
D = 1, W = 0
MOD = 00
REG = 001
R/M = 111

MOV 43H[SI], DH

OPCODE = 100010
D = 10, W = 0
MOD = 01, REG = 110
R/M = 100

*** Assignment → Move instruction all exercise + example. } 18/9/23 → সোমবার Submission.
Color → Language বিচার। }

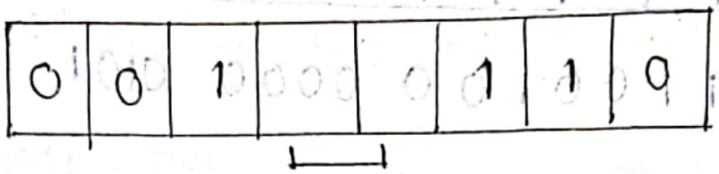
Lecture-6

18/9/23

Monday

MOV CS:[BX], DL

SEGMENT OVERRIDE PREFIX

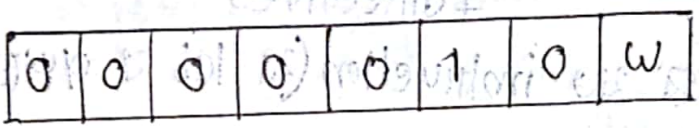


Segment Register

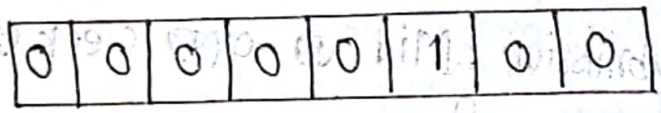
ADD AL, 01H

Segreg	code
CS	01
DS	11
ES	00
SS	10

The format is-

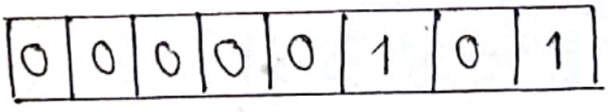


, data byte, data byte



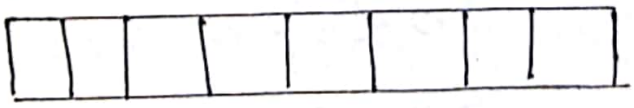
, 0000 , 0001

ADD AX, 3201H



, 01 (Low) Bin → 210 , 32 (High) Bin → 205

ADD AL, 05H



IN AL, 05H → Appendix A তে এগুলি instruction এর format আছে।

*** MOV, ADD, IN

Formate for IN -

1	1	1	0	0	1	0	w

, data byte, data byte

1 1 1 0 0 1 0 0 0000 0101

IN AX, 3205 H

1	1	1	0	0	1	0	1

E5H, 05H, 32H

+ directives

*** Chapter-6 এর ২৩ instruction (৯ম lab এ পর্যন্ত) ২(দুই) Syntax, Explanation, example Assignment.

↳ Submission : Mid এর পরের week এর বুধবার।

Lecture-7
Wednesday
4/10/23

Instruction Timing and Delay Loops

clock cycles

Mov cx, N ; 4 = C_0 → clock cycle overhead

KILL TIME : NOP ; 3
NOP ; 3 ↗ Jump করলে
LOOP KILL TIME ; 17 or 5 → Jump না করে পরের line এ গেলে

$\therefore C_T = C_0 + N(C_L) - 12$

$\Rightarrow N = \frac{C_T - C_0 + 12}{C_L} = \frac{5000 - 4 + 12}{23} = 218 = 0DAH$

1ms delay = 1000 μs delay

8086 μp, clock Speed = 5MHz = 5×10^6 Hz

$n = \frac{1}{T}$

$\Rightarrow T = \frac{1}{n} = \frac{1}{5 \times 10^6} = 0.2\mu S$

0.2 μs → 1 c.c

∴ 1 " → $\frac{1}{0.2}$ "

∴ 1000 " → $\frac{1000}{0.2}$ c.c

= 5000 c.c

2.5 ms Delay

2.5 ms delay = 2.5×10^3 μs

0.2 μs → 1 cc

∴ 1 μs → $\frac{1}{0.2}$ cc

∴ 2.5×10^3 → $\frac{2.5 \times 10^3}{0.2} = 12,500$

$\therefore N = \frac{12,500 - 4 + 12}{23} = 544D = 220H$

Nested LOOP: → অনেক বড় time দরকার হলে

```
MOV BX, COUNT1 ; 4
CNTD1: MOV CX, COUNT2 ; 4 (COUNT1)
CNTD2: LOOP CNTDN2 ; ((17 × COUNT2) - 12) COUNT1
       DEC BX ; 2 (COUNT1)
       JNZ CNTDN1 ; 16 (COUNT1) - 12
```

30 s delay

Lecture-8

5/10/23
Thursday

8086 Addressing Modes

→ Slide

Some important topics:

① Editor
② Assembler
③ Linker
④ Locator
⑤ Debugger
⑥ Emulator

→ Assignment.

Lecture-9

9/10/23
Monday

CT Syllabus: lecture 1-8

- এর instruction detail এ আরো আসবে with example.
- " directives " " " " " " .

=> DB, DW, DD,

=> CMP instruction এর কাজ কী?

CMP Destination, Source

CMP Bx, CX

Condition	CF	ZF
CX > BX	1	0
CX < BX	0	1
CX = BX	0	1

Difference between Conditional Jump and unconditional Jump:

Soln: what is con Jump and uncon Jump

JMP X
+127 to -128

JA
JAE
JG
JB
JBE
+32767 to -32768

+ কার কন) flag এর মান কত তাও mention করা লাগবে।

Exm:

```
CMP AL,100
JAE HEATER_OFF
HEATER_ON: - -
- - - JMP X
- - - - -
HEATER_OFF:
- - - - -
- - - -
```

JB HEATER_ON
JMP HEATER_OFF

* Heater ON থাকলে infinite loop চলবে।

Lecture-10

Wednesday

11/10/23

Procedures

*** Procedures ব্যবহার assembly Program কে আরো সহজ করা যায়।

Classification of procedures:

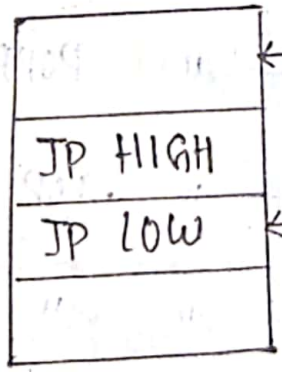
① Direct within-segment near call ; call BP

② Indirect within-segment near call ; CALL WORD PTR [BX]

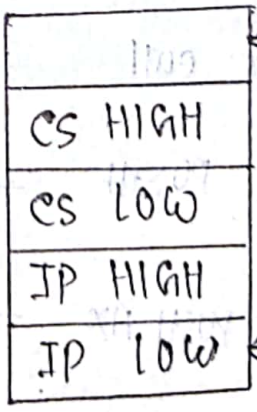
③ Direct inter-segment Far call

④ Indirect " - " "

For Near call:



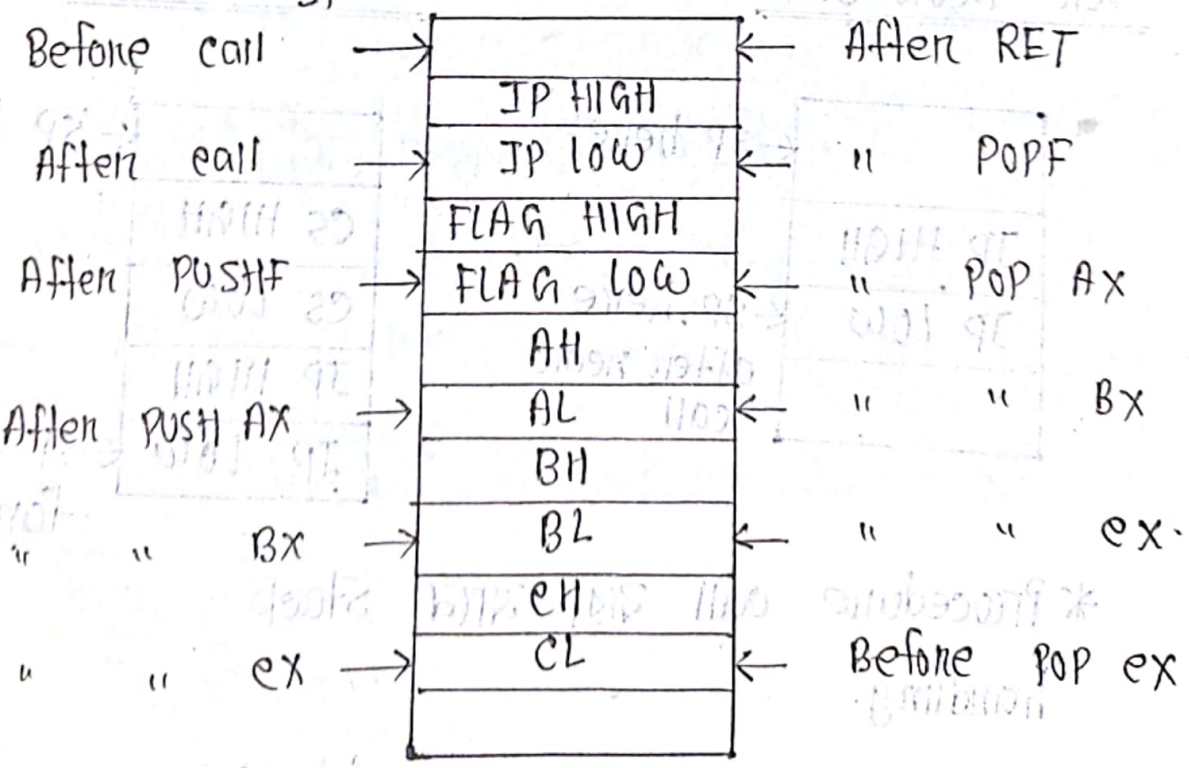
For Far call:



* Procedure call করা মানে Stack handling.

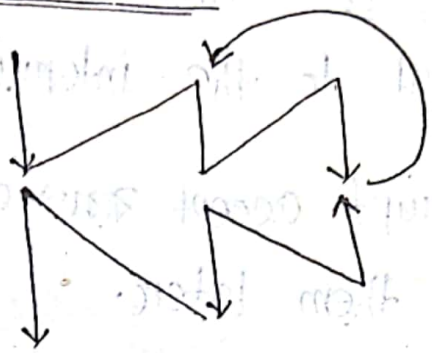
PUSH & POP Instruction:

```
MULTO PROC NEAR
      PUSHF
      PUSH AX
      PUSH BX
      PUSH CX
      :
      POP CX
      POP BX
      POP AX
      POPF
      RET
MULTO ENDP
```



(***) PUSH POP + mapping

Reentrant Procedures:



Interrupt

Sources:

1. Hardware interrupt
2. Software interrupt
3. Error conditions

Lecture-11
10/10/23
Thursday

8086 Interrupts

Interrupts 3 types:

① Software → 256 types
② Hardware → Maskable – INTR – PIN 18
→ Non maskable – NMI – PIN 17

③ Error handling

* Flag important → return বারও POP এর আগে।

Hardware Interrupt: Initiated by external hardware by sending signal to the interrupt pin.

Maskable: Interrupt accept করতে পারে, নাও পারে। Reenable them later.

1 = Unmask → reset

set = 0 = mask

Unmaskable: Interrupt must accept করা লাগবে।

→ Power failure

→ critical timing

Software: Program instructions.

8086 interrupt types (***)

⇒ 256 interrupts are divided into 3 Groups –

① Type 0 to 4

→ Dedicated interrupts.

* প্রত্যেক type এর জন্য 4টা address.

② Type 5 to 31

→ 8086 use করে না।

③ Type 32 to 255

→ User defined

8086 interrupt pointer table/ (***)
vector table

* Type কে 4 দিয়ে গুণ করলে address পাওয়া যাবে।

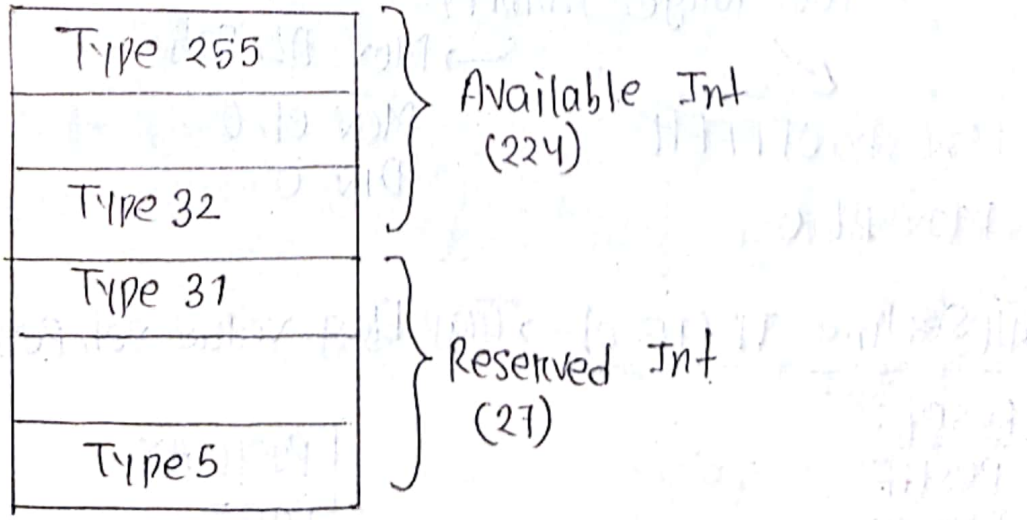
Priority Interrupt:

(***) 8259A Priority Interrupt controller → figure সহ + ব্যাখ্যা করা লাগবে কীভাবে Priority অনুযায়ী interrupt handel করে।

Lecture-12
Monday
30/10/23

Interrupt - 2

Interrupt Pointer Table:



⇒ The starting address of an isp is called interrupt vector / interrupt table.

⇒ The table which contain vector is called Pointer table.

Example 11.1:

$$\text{Address} = 4 \times 50 = (200)_{10}$$
$$= (11001000)_2$$
$$= (C8)_{16}$$

$IP_{50} = 000C8$, $CS_{50} = 000eA$

*** INT vs CALL

Divided by Zero : Type 0 :

→ Too large, infinity

↙

```
Mov AX, OFFFFH
Mov BL, 0
```

↳ Mov AL, 82h

```
Mov el, 0
Div 0
```

Setting TF (TF=0) → Trap Flag value set, Reset

Reset :

```
PUSHF
Pop AX
AND AX, 1111 1110 1111 1111B
```

```
PUSH AX
POPF
```

Set:

```
PUSHF
POP AX
OR AX, 0000 0001 0000 0000B
PUSH AX
POPF
```

Single step int, type-1:

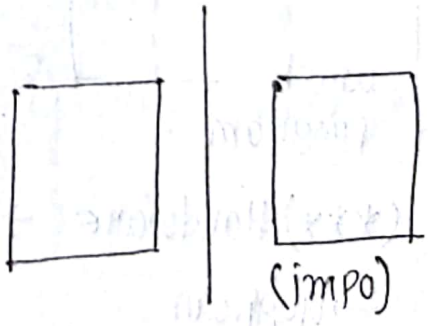
```
PUSHF
MOV BP, SP
OR WORD PTR[BP+0], 0100H
POPF
```

Overflow int, type-4:

```
MOV AL, 64
MOV BL, 64
ADD AL, BL
INTO ; OF=1
```

*** Loop instructions

(***) 8259A Priority int controller:



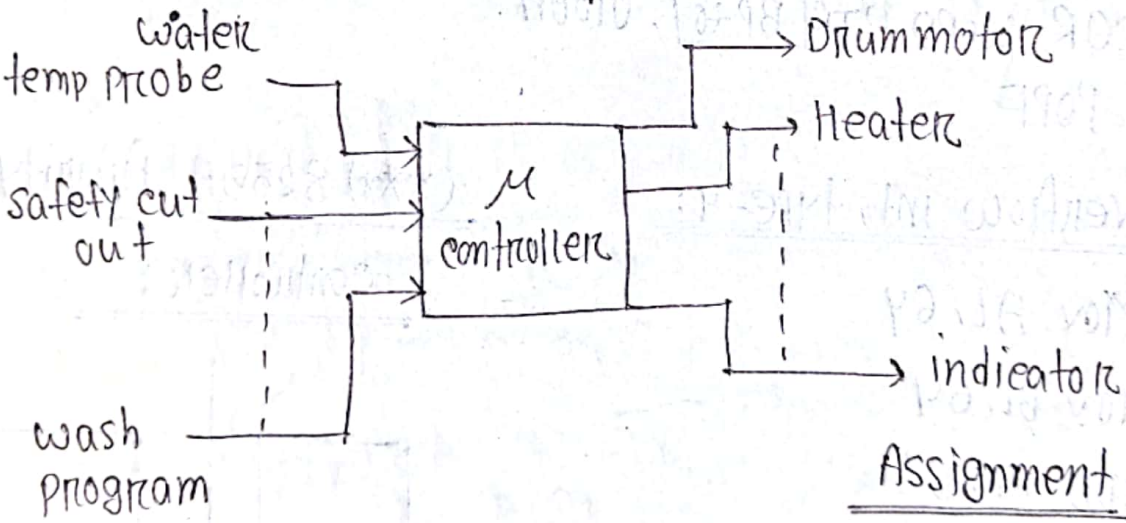
=> Who priority is handled explain with figure?

Lecture-13
31/10/23
Tuesday

Microcontroller Based System Design

[1] Washing machine hardware(***) [what are the hardware components].

① Temperature sensor
② safety cut out switch
③ keyboard
④ Water level gauge
⑤ washing drum
⑥ Power switches
⑦ Heater
⑧ Inlet valve
⑨ Suction pump
⑩ control lamps



(***) Hardware + Software part
→ Diagram
→ flowchart
→ Algo
(***) washing cycle/
(***) Hardware design → circuit diagram

Assignment:
Convert code for 8086.

Leeture-14
Tuesday
7/11/23

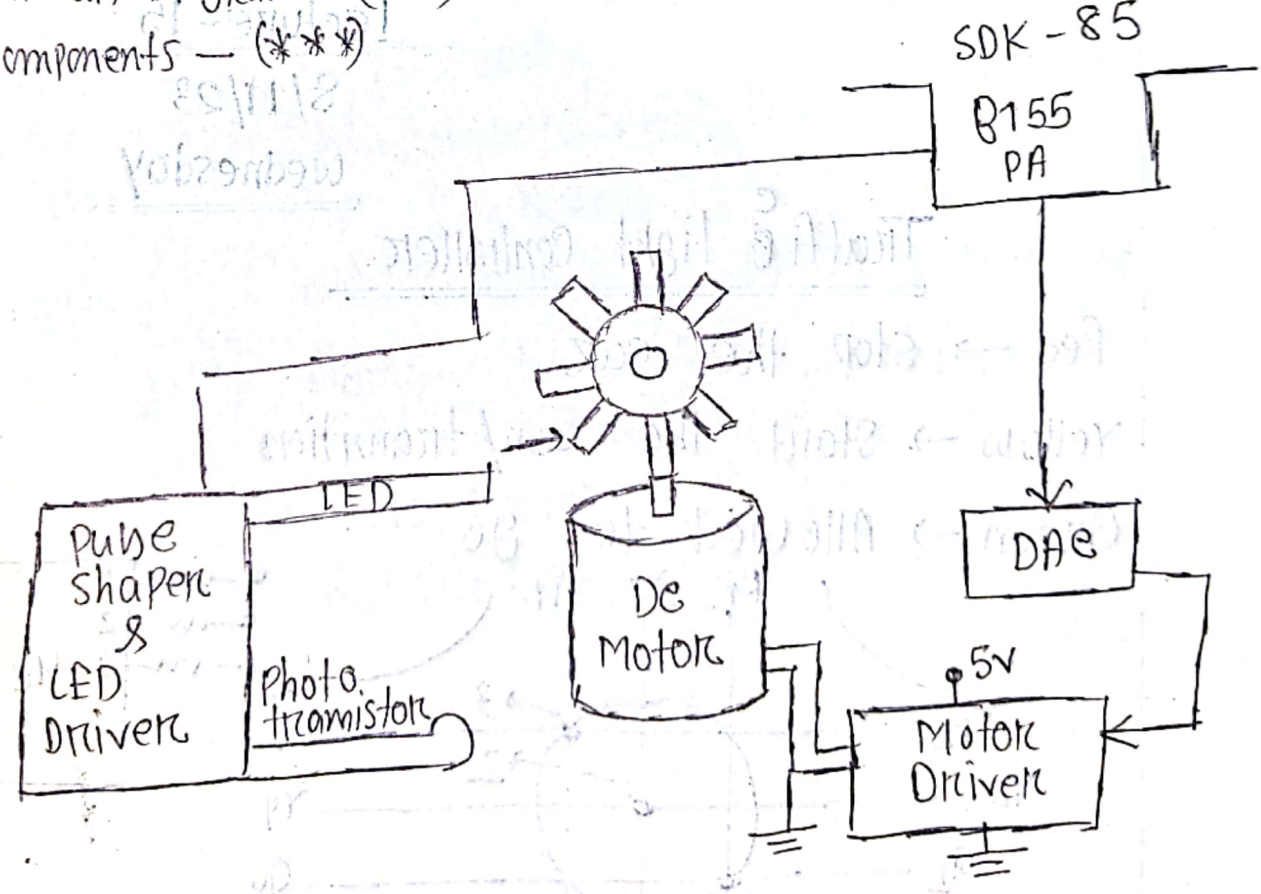
Speed control of De motor

Book: Microesson Prineiples and Applieation (Ajit Pal)

Abstract: The purpose of a motor speed control and take input signal and will drive the De motor.

Speed control of De motor:

Circuit Diagram — (***)
Components — (***)



1 Second → m pulse

n Pulse → 1 rotation

∴ 1 " → 1/n "

∴ m " → $\frac{m}{n}$ rotation

⇒ Counter will count pulse

Software Design: (***)

Lecture-15

8/11/23

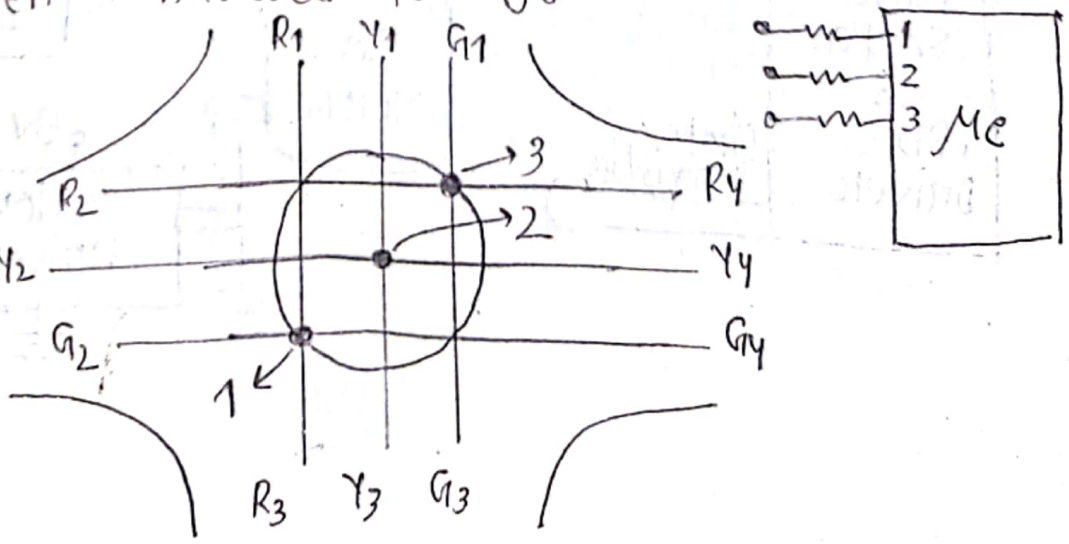
Wednesday

Traffic light Controller

Red → stop the car

Yellow → Start the car/ tramition

Green → Allowed to go



```
setup () {
    Pinmode (1, output);
    Pinmode (2, output);
    Pinmode (3, output);
}
loop() {  digitalwrite (1, HIGH);
          delay (5000); ⟶ 5ms by default
          digitalwrite (1, LOW);
            "      (2, HIGH);
          delay (1000);
          digitalwrite (2, LOW);
            "    (3, HIGH);
          delay ( 5000);
          digitalwrite (3, LOW);
            "     (2, HIGH);
          delay (1000);
          digitalwrite (2, LOW);
}
```

Sadi Sir

Lecture - 16

TOPIC NAME : Procedure

DAY : Wednesday

TIME : DATE : 15/11/23

Passing Parameter to and from Procedures:

*** loop মুখো / Jump এর diff

*** conditional vs Unconditional

*** Jump

*** lab এর topic + Program দেওয়া থাকবে।

→ Theory বুঝবে / Program ও বুঝবে।

① Passing Parameters in registers
② " " " memory
③ " " " pointer
④ " " " stack

} ৪ টা method এর সুবিধা + অসুবিধা।

17 H

17 H → 1×0A + 7 → Binary

AL = 17
0F
07

0001 0111
0000 1111
0000 0111

*** Overflow } push করিবা,
pop করা

GOOD LUCK

#littleGiant_34