EF – 306 – 1st Test

Attempt any three. All questions carry equal marks

Q.1 Explain intensity modulation with suitable diagram.

Q2. Explain a laser drive circuit

Q3. What do you understand by linearization. Explain various linearization techniques used with LED.

Q4. Explain various coding scheme and compare them.

Solutions

Shalini Garg – GPC Jodhpur

Q.1 Explain intensity modulation with suitable diagram.

Ans. Intensity modulation is direct modulation technique in which the output power varies in proportion to the modulating signal.

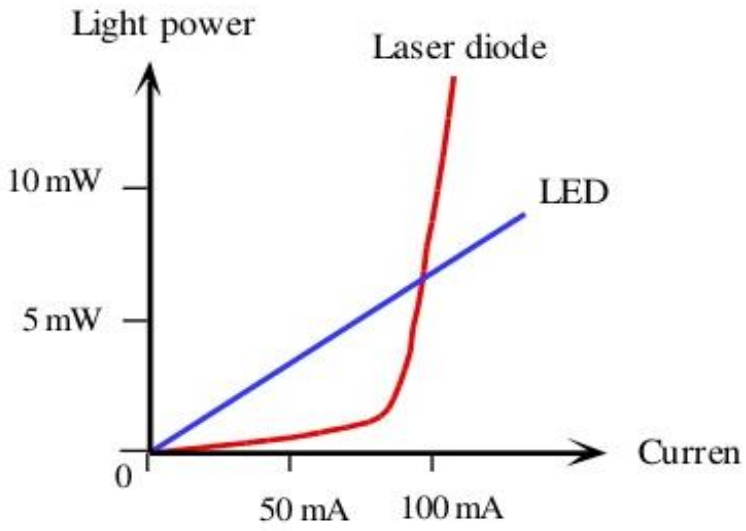



Figure 1 – Relationship between drive current and optical power output for LED and Laser diode

As shown in figure 1, the output power varies linearly with drive current. The result is that total modulating current (i) and resulting optic power (P) can be expressed as.

a) $i = I_{dc} + I_{sp}\sin\omega t$ --------(1)

b) $P = P_{dc} + P_{sp}\sin\omega t$ --------(2)

Where first term represents the d.c bias and 2nd term represents the information signal. $P_{sp}$ is the peak signal power and $I_{sp}$ is the peak signal current.

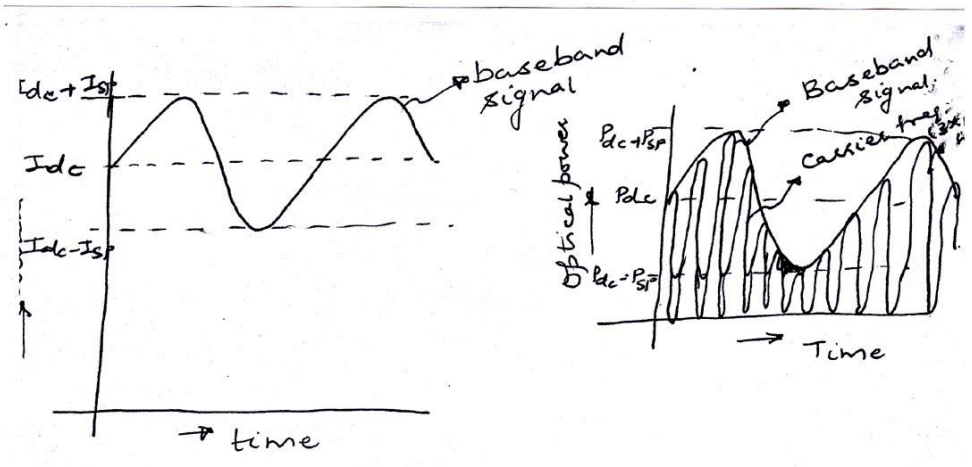



Figure 2- Illustration of (a) Modulating current and (b) Resulting optic power.

The modulation factor m' is the peak current excursion divided by average current.

i.e. $m' = I_{SP}/I_{DC}$

We define optic modulation factor in terms of optic power. Thus

$m = P_{sp}/P_{dc}$

allowing us to write optic power as

c) $p(t) = P_{dc}(1+m\sin\omega t)$.

Although modulating signal has both positive and negative part, a d.c bias current is added to make the modulating current entirely positive as shown in

figure 2, transmitting information as variation in optic power. This is called intensity modulation.

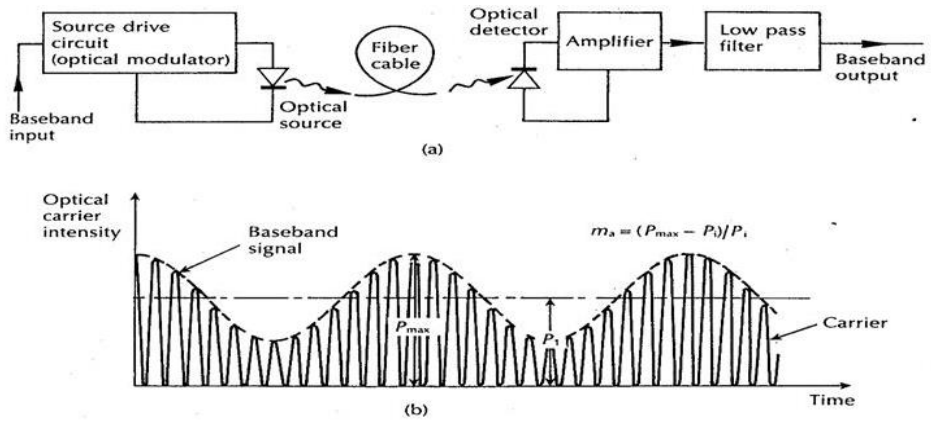



Fig 3 = Analog optical fiber system emptying direct intensity modulation.

At the receiving end, optical power is converted to an electrical current by the detector producing current i(t).

i(t) = R p(t)

where R = detector responsivity (A/W)

The current is amplified and filtered in the receiver. In an intensity modulation system demodulation is not necessary since the base band signal i(t) is already simply proportional to the message.

Q2. Explain a laser drive circuit

The circuit shown in figure 4 using P-channel FET is suitable for high speed digital modulation. Using GaAs MESFET's (Metal Schottky FET) the circuit can modulate laser at the rate more than 1 Gbit/sec. The modulating signal is applied between the gate and the source. The zero voltage applied between gate to source

reduces the channel resistance, the result is that the current flowing through $R_1$ is diverted through FET and almost negligible current flows through laser producing zero output power. For high positive voltage applied between gate to source the channel resistance increases so most of the supplied current passes through the diode, producing high output power.

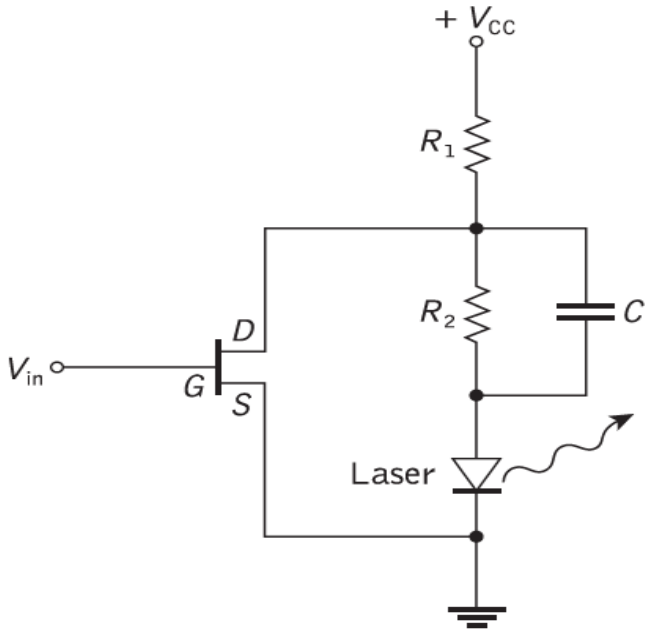



Figure 4 – A shunt drive circuit for use with an injection laser.

The gate voltage is adjusted just below the threshold value of laser. An increased gate voltage turns the laser on. The resistor $R_2$ in series with the diode ensures that that $V_{ds}$ is large enough in both on and off states. The capacitor C improves the switching speed of the circuit. Linearity of laser is not a problem for digital modulator as the diode on current does not depend upon the signal voltage $V_{gs,}$ as long as the voltage is above the minimum level. The disadvantage of the circuit is absence of feedback circuit to control laser bias and drive current in order to control temperature variations and ageing.

Q3. What do you understand by linearization. Explain various linearization techniques used with LED.

Ans. The term linearization means some form of linearization technique incorporated in the circuit to compensate for both LED and drive circuit

nonlinearities. Linearization technique is essential in frequency multiplexed system and baseband video transmission of TV signals as they require the maintenance of extremely low levels of amplitude and phase distortion.

Many techniques are used some of which are illustrated in Figure 5.

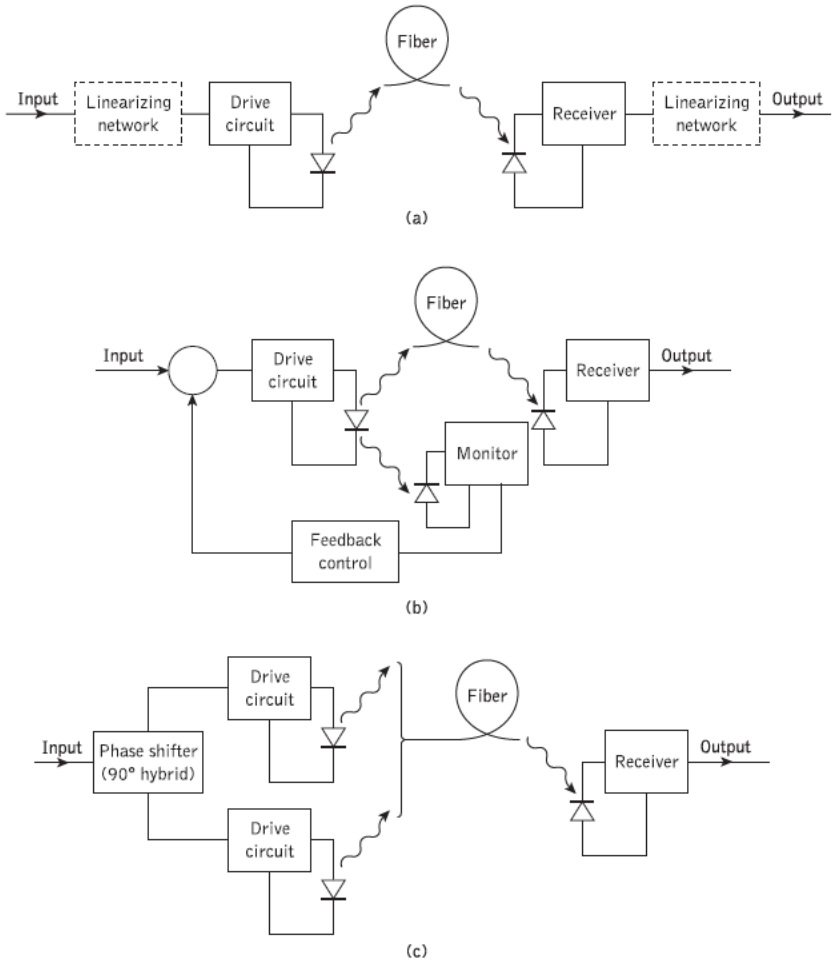



Figure 5 - Various linearization techniques- a) Complementary distortion technique b) Negative feedback compensation technique c) Selective harmonic compensation

- Complementary distortion technique- Figure 5(a) shows the complementary distortion technique where additional nonlinear devices are included in the system. It may take the form of predistortion compensation (before the source drive circuit) or post distortion compensation (after the receiver). This approach has been shown to reduce harmonic distortion by up to 20 dB over a limited range of modulation amplitudes.

- Negative feedback compensation technique- In the negative feedback compensation technique shown in Figure 5(b), the LED is included in the linearization scheme. The optical output is detected and compared with the input waveform, the amount of compensation being dependent on the gain of the feedback loop. Although the technique is straightforward, large-bandwidth requirements (i.e. video) can cause problems at high frequencies
- Selective harmonic compensation - The technique shown in Figure 5(c) employs phase shift modulation for selective harmonic compensation using a pair of LEDs with similar characteristics The input signal is divided into equal parts which are phase shifted with respect to each other. These signals then modulate the two LEDs giving a cancellation of the second and third harmonic with a 90° and 60° phase shift respectively. However, although there is a high degree of distortion cancellation, both harmonics cannot be reduced simultaneously.
- Other linearization techniques include cascade compensation, feedforward compensation and quasi-feedforward compensation.

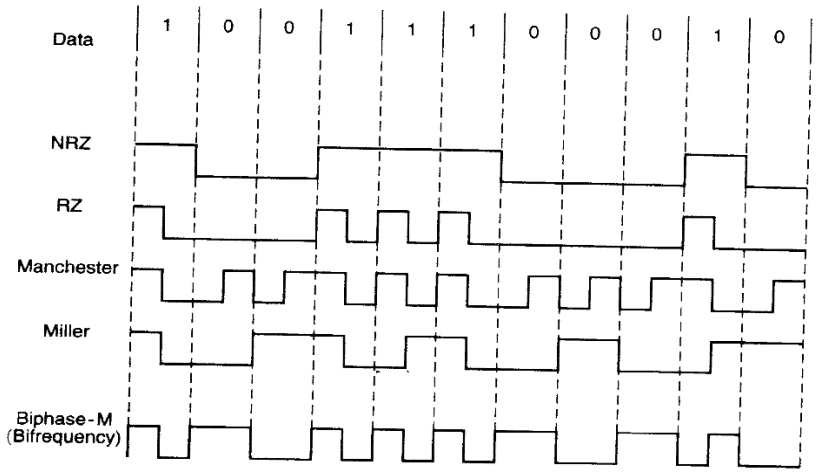
Q.4 Explain various coding scheme and compare them.

Ans. Various coding schemes are-

1. RZ 2. NRZ 3. Miller 4. Manchester 5. Bi-phase 6. Bipolar code
    1. RZ.- Return to Zero- The signal level returns to a nominal Zero level between bits. This takes place even if consecutive 0s or 1s occur in the signal. The signal is self-clocking. It uses twice the bandwidth to achieve the same data-rate as compared to non-return-to-zero format.
    2. NRZ- Non Return to Zero:- Signal does not return to zero but remains at one if two successive 1 bite are transmitted. It is a binary code in which ones are represented by a positive voltage, while zeros are represented by zero condition. NRZ is not inherently a self-clocking signal, so some additional synchronization technique must be used for avoiding bit slips. The NRZ code requires only half the baseband bandwidth required by the RZ format or Manchester.
    3. Manchester coding:- signal level always changes in the middle of a bit interval ; For 0 bit, signal starts out low and changes to HIGH;For 1 bit

signal starts out HIGH and changes to LOW. It is a self-clocking signal with no DC component.

4. Miller coding:- For 1 bit, the signal changes in the middle of bit interval but not at the beginning or end. For a 0 bit, signal level remains constant through a bit interval, changing at the end of it if followed by another 0 but not if it is followed by 1.
5. Bi-phase:- M or Bi frequency coding for a 0 bit, the signal level changes at the start of an interval. For 1 bit, the signal level changes at the start and middle of bit interval.
6. Bipolar code :- is a three level scheme This code provides a pulse whenever the data change. Transmitter switches to full power for half bit interval whenever 0 follows 1.It then returns to half power level and remains there until a 1 appears, at which time power drops to zero for a half bit interval. The power then returns to the median level. In this scheme. dc (average power level will not change, regardless of data pattern. This gives stable operation even with an AG.C receiver.



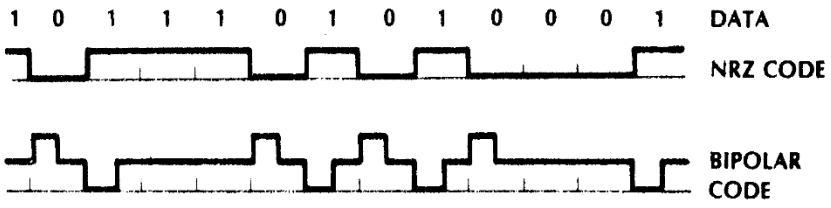



Figure 6- Illustration of various coding schemes

Comparison between various codes.

1. The bandwidth requirement for NRZ code is Minimum (B), while RZ and Manchester coding requires twice as much transmission bandwidth (2B).
2. NRZ requires d.c coupling and changing dc level results in drift. While RZ and Manchester coding requires a.c coupling minimizing drift.
3. For NRZ series of alternating 1's and 0's reveals the clock rate but not all 1's or 0's; For RZ clock rate can be measured for successive 1's but not any other data pattern . Manchester offers the benefits of clock recovery.
4. In case receiver employs AGC, the bipolar coding is best as dc power level will not change regardless of data pattern, giving stable operation. While for all other codes each pulse is amplified by an amount determined by stream of data proceeding it, making receiver difficult to recognize the data bit.

EL 302 -I test

Shalini Garg – GPC Jodhpur

Attempt any three questions. All question carry equal marks.

1. Explain various registers of 8086.

Ans . The BIU of 8086 has four segment registers- CS, DS,SS and ES and the execution unit has nine 16 bit registers AX, BX, CX, DX, SP, BP, SI and DI and flag registers

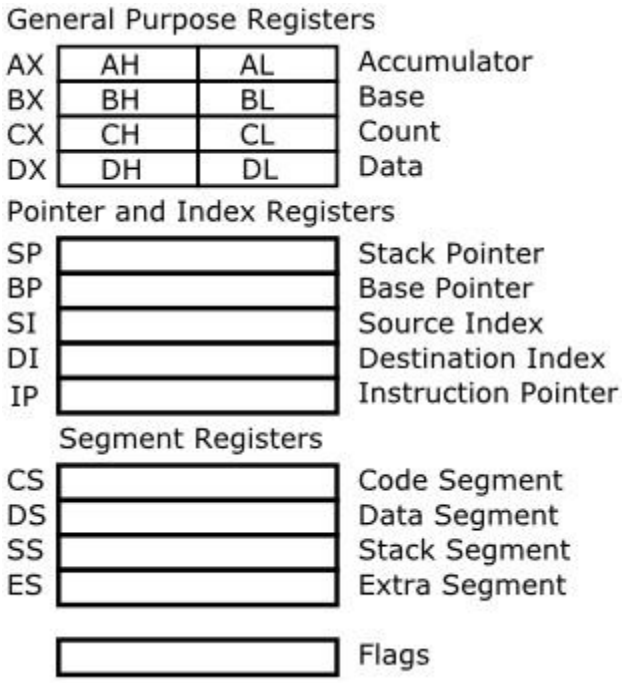
Segment registers

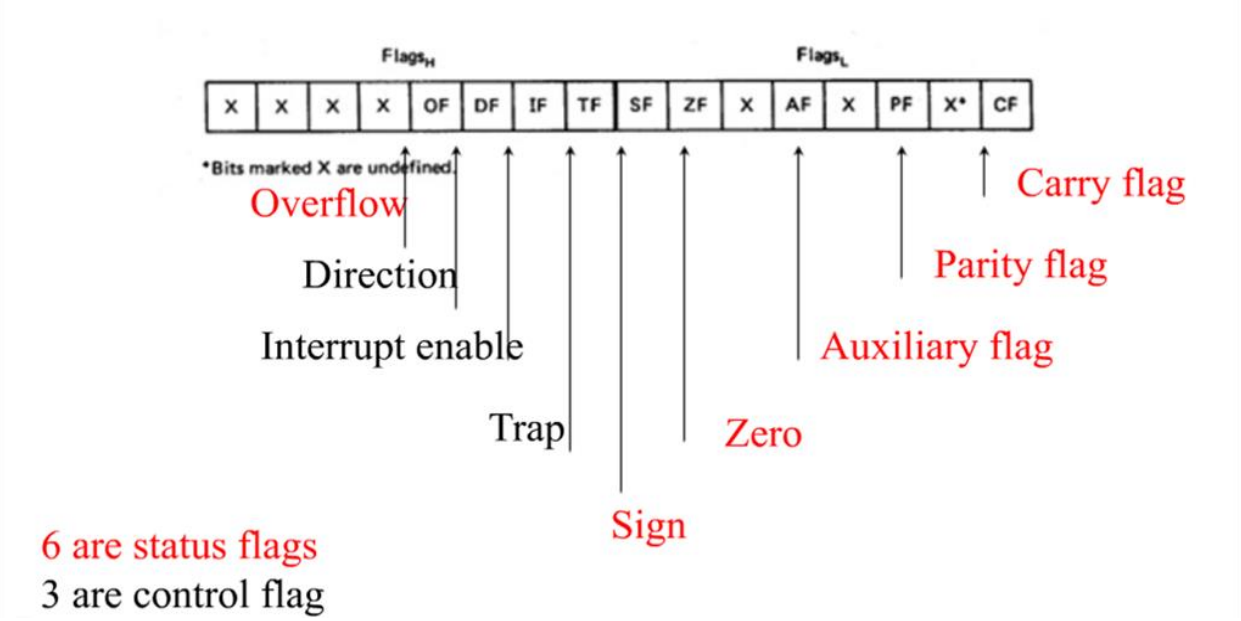
1. CS- The code segment register- it stores all program instructions. The instructions are pointed by 16 bit offset in instruction pointer(IP).
2. DS- The data segment register-It point to current data segment, operands for most instructions are fetched from this segment. The 16 bit content of SI, DI or displacement are used for computing 20 bit physical address.
3. SS- The stack segment- It point to the current stack. SP and BP are default offset register associated with stack.
4. ES- Extra segment-It point to extra segment in which data (in excess of 64 KB pointed by DS) is stored. String instruction always use the ES and DI to determine 20 bit physical address.

General registers- The 16 bit general register AX, BX, CX, BX, can also be used as two 8 bit registers (AH, AL, BH, BL; CH, CL, DH, DL)

1. AX- is called 16 bit accumulator while for 8 bit operation AL is the accumulator
2. BX register is called as base register. This is the only general purpose register whose content can be used for addressing 8086 memory.
3. CX register- It is known as counter register because some instructions such as Shift, Rotate and Loop use content of CX as a counter.
4. DX register- Data register is used to hold 16 bit result (data), Upper word for word by word multiplication or high 16 bit dividend before 32bit by16 bit division.

5. SP and BP :- Stack pointer and base pointer are used to access data in stack.
6. SI and DI:- Source index and Destination index are used for index and addressing.
7. Flag register :- holds the status flags it contains 6 one bit status flag and three one bit control flag






Flag register

1. AF- Auxiliary carry flag-It is set if there is carry from lower nibble to upper nibble (D3 to D4)
2. CF (Carry Flag)- It is set if there is carry from addition or borrow from subtraction.
3. OF (Overflow flag)- is set if there is an arithmetic overflow.
4. SF(Sign flag):- set if MSB of result is 1.
5. PF(Parity flag):- set if result has even parity.
6. ZF(Zero Flag):- is set if result is zero.

Q.2 Compare operating modes of 8086 and draw diagram for maximum mode.

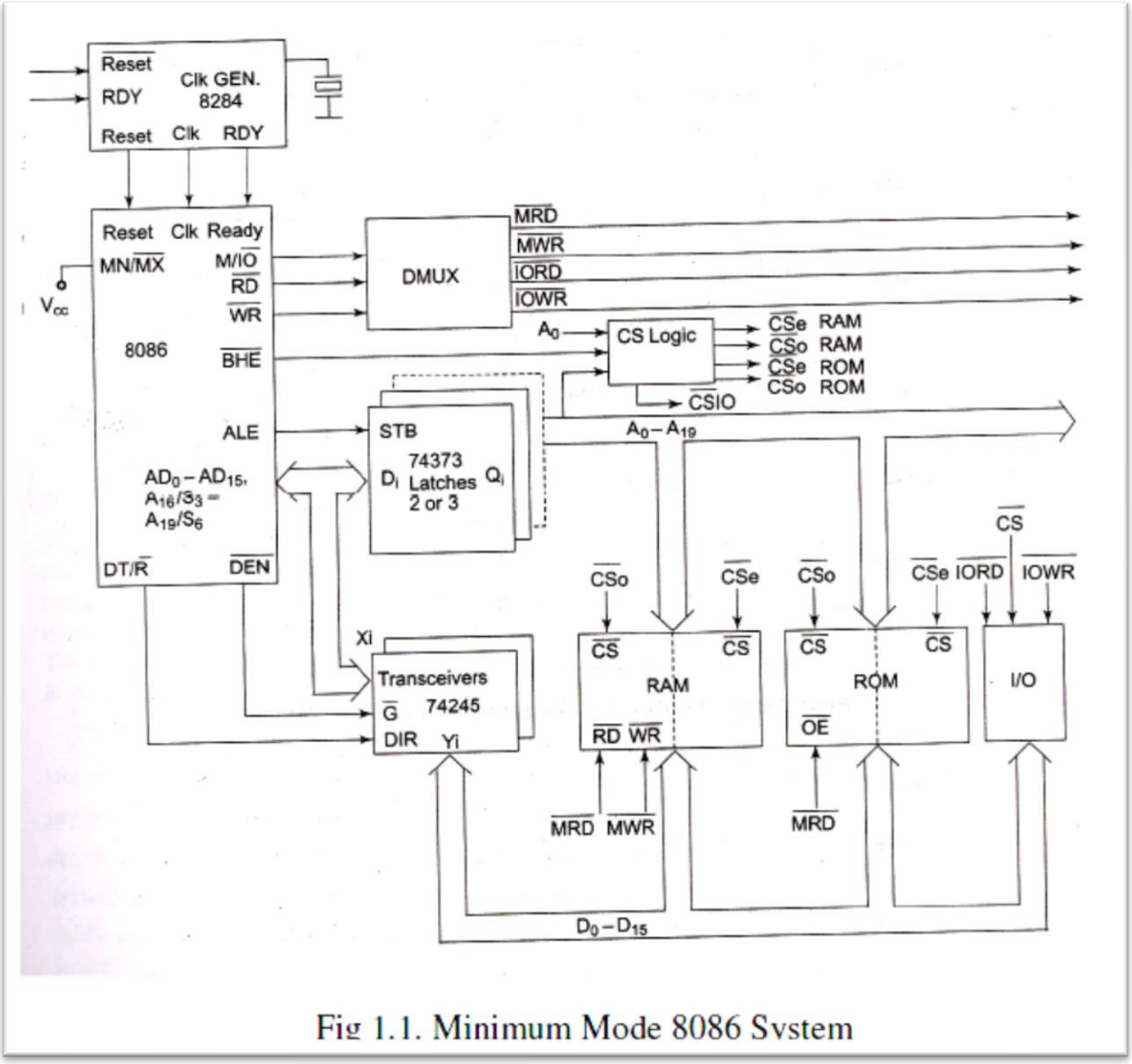



Fig 1.1. Minimum Mode 8086 System

There are two operating modes for 8086- the maximum mode and the minimum mode. The difference between them are as follows-

| S.no. | Minimum mode | Maximum mode |
|---|---|---|
| 1. | MN/$\overline{MX}$ is connected to $+V_{cc}$ | MN/$\overline{MX}$ pin is connected to ground. |
| 2. | Single processor system | Multiprocessor system |
| 3. | ALE for latching address is generated by 8086 itself | External bus controller generates ALE |
| 4. | $\overline{DEN}$ and DT/$\overline{R}$ signals for trans receiver are generated by microprocessor itself | $\overline{DEN}$ & DT/$\overline{R}$ signals are generated by 8288 bus controller. |
| 5. | Control signals like $\overline{RD}$, $\overline{WR}$, are generated by 8086 itself | Control signals like $\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMTC}$, $\overline{IORC}$, $\overline{IOWC}$, $\overline{INTA}$ are generated by 8288 |

Q3. Explain various addressing modes of 8086

Ans. Addressing modes refer to the different methods of addressing the operands. There are 8 different addressing modes in 8086 programming −

1. Immediate addressing mode

The addressing mode in which the operand is specified in the instruction itself. is known as immediate addressing mode.

Example

MOV CL, 12H

This instruction moves 12 immediately into CL register. CL ← 12H

MOV CX, 4929 H; ADD AX, 2387 H; MOV AL, FFH ; Here 4929, 2387 and FF represents immediate data in three instructions respectively.

Register addressing mode -In this mode, operands are specified using registers. This addressing mode is normally preferred because the instructions are compact and the execution is fastest among all instruction forms.

Example:

MOV AX, BX

This instruction copies the contents of BX register into AX register. AX ← BX

3. Direct memory addressing mode-In this mode, address of the operand is directly specified in the instruction. Here only the offset address is specified, the segment being indicated by the instruction.
Example:
MOV CL, [4321H]
This instruction moves data from location 4321H in the data segment into CL.
The physical address is calculated as
DS * 10H + 4321
Assume DS = 5000H
∴PA = 50000 + 4321 = 54321H
∴CL ← [54321H]

4. Register based indirect addressing mode-
In this mode, the effective address of the memory may be taken directly from one of the base register or index register specified by instruction. If register is SI, DI and BX then DS is by default segment register.
If BP is used, then SS is by default segment register.
Example:
MOV CX, [BX]
This instruction moves a word from the address pointed by BX and BX + 1 in data segment into CL and CH respectively.
CL ← DS: [BX] and CH ← DS: [BX + 1]
Physical address can be calculated as DS * 10H + BX.
5. Register relative addressing mode-
In this mode, the operand address is calculated using one of the base registers and an 8 bit or a 16 bit displacement.
Example:
MOV CL, [BX + 04H]
This instruction moves a byte from the address pointed by BX + 4 in data segment to CL.
CL ← DS: [BX + 04H]
Physical address can be calculated as DS * 10H + BX + 4H.
6. Base indexed addressing mode-
Here, operand address is calculated as base register plus an index register.
Example:

MOV CL, [BX + SI]

This instruction moves a byte from the address pointed by BX + SI in data segment to CL.

CL ← DS: [BX + SI]

Physical address can be calculated as DS * 10H + BX + SI.

7. Relative based indexed addressing mode-

In this mode, the address of the operand is calculated as the sum of base register, index register and 8 bit or 16 bit displacement.

Example:

MOV CL, [BX + DI + 20]

This instruction moves a byte from the address pointed by BX + DI + 20H in data segment to CL.

CL ← DS: [BX + DI + 20H]

Physical address can be calculated as DS * 10H + BX + DI + 20H.

8. Implied addressing mode-In this mode, the operands are implied and are hence not specified in the instruction.

Example:

STC

This sets the carry flag.

Q. 4. Explain various Data transfer or Arithmetic instructions with one example each.

Ans. :- Data transfer instructions.

1. MOV operand 1, operand2 → Copies content of operand 2 in operand1.

| S.No. | Possible operand 1 | operand2 |
|---|---|---|
| 1. | Reg | Reg |
| 2. | Reg | 8/16 immediate data |
| 3. | Mem | Reg |
| 4. | Reg | Mem |
| 5. | Mem | 8/16 immediate data |
| 6. | Reg | SReg |
| 7. | Sreg | REg |
| 8. | Sreg | Mem |
| 9. | Mem | Sreg |

Where Reg:-Register (AX, BX, CX, DX, SI, DI,BP, SP)
SReg :- Segment Register
Mem :_ Memory
Note CS cannot be destination register
Eg. (i) MOV AX, 4278
Moves 4278 in register AX.
(ii) MOV [SI], BX
Copies content of BX in Memory Location whose physical address is [DSX10+SI]

2. XCHG operand 1, operand2.
Exchanges content of operand 2 with operand1

| S.no | Possible operand 1 | operand2 |
|---|---|---|
| 1. | REG | Mem |
| 2. | Reg | Reg |
| 3. | Mem | Reg |

Example.
(i) XCHG CL, DH ; [CL] ↔ [DH]
(ii) If CL = 45, DH= 96
After
XCHG CL, DH
DH= 45, CL= 96

3. LAHF → Load AH with Lower byte of flag.
[AH] ← {F(L)}
If F= F024
After
LAHF
AH=24 Ans
4. SAHF → Content of AH are stored into lower order byte of flag register.
5. IN port address.
Content of port whose address is specified in the instruction is copied into accumulator
Example-

IN AL, 40H

[AL] ← {port 40}

Content of port no 40 are copied in AL.

(iii) IN AX, 73H

[AL] ← port 73} ; content of port no 73 are copied in AL

[AH] ← {port 74} And port no 74 into AH

(iv) Indirect port addressing

IN AX, DX

Here DX register specifies port address.

Example-

If DX = 4200

IN AX, DX

Copies content of port no 4200 in AL and port no 4201 in AH.

[AL] ← {4200}

[AH] ← {4201}

Similarly we have

(i) Out port address, AL/AX

(ii) Out DX, AL/AX

The content of accumulator are copied to O/P port whose address is specified in the instruction (i) or to the port whose address is in DX register (ii)

7. LEA operand 1, operand2

Load effective address (offset address) of operand2 into the operand 1

Example -

LEA S1,n

If n is defined as data byte (DB) in DS of memory

DS: 3426 n DB 5

Then SI= 3426

8. LDS reg, Mem.

The 32 bit data from operand 2 is copied into DS and operand 1 (LSB-16bit)
Example -
Let DS= 300; SI= 50

P.A = DS*10+50 = 03050.
Assuming memory location 03050 onwards contains data as shown below.

| | |
|---|---|
| 3053 | 70 |
| 3052 | A4 |
| 3051 | 30 |
| 3050 | 20 |

Then after LDS BX, SI

BL= 20; BH= 30; BX= 3020 and DS= 70A4.

9 LES reg, mem
The 32 bit data from memory is copied into ES and register.

10. XLAT - Reads a byte from look-up table.
[AL] ← {DSX10 + BX + AL $]_M$
Like MOV AL, [AL][BX]

Instruction is useful for translating character from one code to another.

Example-
BX = 7208; AL = 70; DS= 1000
Then P.A = 1000*10+7208+70 = 17278
That is content of memory location 17278 are copied into AL.

Or

Arithmetic instructions.

1. Add operand1, operand2.

| S. No. | Operand1 | Operand2 |
|---|---|---|
| 1. | Reg | Immediate |
| 2. | Reg | Reg |
| 3. | Reg | Mem |
| 4. | Mem | Reg |
| 5. | Mem | Immediate |

Add content of operand 1 and operand 2, result is stored in operand1.
{operand1} ← {operand 1} + {operand2}

Example.

Let AL=52; BL = 7A.

Add AL, BL
[AL]= [AL]+[BL]
i.e.AL = 52+7A = CC
AL = CC

2. ADC operand1, operand2.

[operand1] ← [operand1] + [operand2] + [CY]
Adds with carry operand1 and operand2.

E.g. –

If AL = 52; BL= 7A; CY = 1

ADC AL, BL
[AL] = [AL] + [BL] + [CY] = 52 + 7A +1 = CD Ans.

3. SUB operand 1, operand2

Subtract operand2 from operand1; result in operand1.

[operand1] ← [operand1] – [operand2]

E.g.

AH= 95; CL= 67

SUB AH, CL
[AH] – [AH] – [CL]
= 95-67
67= 01100111
1’s comp = 10011000
2’s comp= +1

10011001
+ 95 = 10010101

1/ 0010 1110

Complement Carry0/2E

AH=2E

4. SSB operand1, operand2
   Subtract with borrow operand2 from operand1
   [operand1] ← [operand1] – [operand2] – [CY]
   SBB AH, CL
   [AH] ← {AH] - [CL] - [CY]
   E.g.
   AH= 95; CL = 67; CY = 1
   AH=95-67-1 = 2D.
   Note – For instruction number 2,3 and 4, the possible operand 1 and 2 are same as in instruction number 1.
5. INC operand. (reg/mem).→
   Increments content of operand by 1 ; operand is either in register or memory.
   if BX = 23FD
   then after INC BX.
   BX= 23FD +1 = 23FE.
6. DEC operand. (reg/mem).
   Decrements content of operand by one; operand is either in register or memory.

   DEC BX.
   if BX = 23FD
   after DEC BX

BX = 23FC

7. NEG operand (reg/mem).

converts operand in 2's complement form; operand is either in register or memory.

E.g.

If CX = 9278

After NEG CX

CX = 9278 = 1001001001111000

1's comp= 0110110110000111

2's + 1

0110 1101 1000 1000

= 6D88

i.e.

CX = 6D88. Ans.

8. CMP operand 1, operand2

operand2 is subtracted from operand1, flags are set according, result is discarded.

operand1 – operand2

flags value according to above result

Operand1 > operand2 CF=0; ZF = 0

operand1 = operand2 ZF = 1; CF = 0

operand1 < operand2 ZF = 0; CF = 1.

9. MUL operand (Reg/ mem); Multiply content of AL with the operand which is either in register or memory.

(i) For Byte *Byte.

MUL BL

AX = AL * BL

Content of AL register are multiplied bydata in BL; result in AX register.

For e.g.

Let AL= 40H; BL = 20H; AX = 40*20 ;

Convert hexadecimal number(H) to decimal number(D) –

40H = 64D; 20H = 32D; 64*32 = 2048D = 800H;

Therefore AX = 0800H Ans.

(ii) Word * word
Content of AX are multiplied by operand;
Result in DX-AX.
MUL CX
DX-AX = AX*CX

10. IMUL operand(Reg/ mem).
Same as MUL except that it performs signed multiplication of operand and accumulator.
E.g.
If AL= F5; CL= 05
AX= F5 * 05
F5 = $-11_{10}$; AX = (-11*05) =$-55_{10}$ = $FFC9_{16}$

11. DIV operand (Reg/Mem)
(i) Word by Byte
AX/Reg/mem; Result – Quotient(Q) in AL; Remainder (R) in AH

E.g. For DIV CL; Divides content of AX with Content of CL, Q in AL; R in AH

If AX = 0800H; CL = 20H; then AX/CL = 800/20;

Convert hexadecimal number(H) to decimal number(D) –

800H = 2048D; 20H = 32D; 2048/32 = 64D; R= 00D;

64D= 40H; therefore after DIV CL ; AL= 40H; AH= 00H

ii (Double Word)/Word

That is (DX-AX)/(Reg/mem)

Divides content of DX-AX with content of register or memory.

Q in AX; R in DX.

12. IDIV operand (reg/mem).

Same as DIV, except it performs signed division.

EL- 207 1st test

Shalini Garg – GPC Jodhpur

Attempt any three

Q.1 Convert the following

a. $2AC_{16} = (?)_{10}$

$12*16^0 + 10*16^1+2*16^2 =$

$12+160+512 = 684$ Ans.

b. $291_{10} = (?)_{BCD.}$

= 0010 1001 0001
-------- ------- -------
2 9 1 Ans.

c. 11111100 is a 2's complement number Find equivalent decimal number.

11111100 = -(00000011+1) = - (00000100)= - 4 Ans.

d. $567.62_{10} = (?)_8$

| | | |
|---|---|---|
| 8 | 567 | |
| 8 | 70 | 7 |
| 8 | 8 | 6 |
| 8 | 1 | 0 |
| | | 1 |

567 = 1067;

0.62*8 = 4.96

0.96*8 = 7.68

0.62 = .47……..

Ans $(567.62)_{10} = (1067.47)_8$

e. $(329)_{10} = (?)_{X-3}$

329

333

6512

0110 0101 1100

Q.2 Explain function of following pins.

a) $\overline{\text{RESET IN}}$ :- A low signal on this pin resets the program counter, buses are tri-stated and the microprocessor is reset.
b) HLDA- It is active high hold acknowledge signal. This signal acknowledges HOLD request.
c) IO/$\overline{M}$ :- This is a status signal used to differentiate between I/O and memory operations. When High, it indicates an I/O operation; when LOW it indicates a memory operation.
d) ALE (Address latch Enable):- This is a positive going pulse generated in first clock cycle of every machine cycle; it indicates that the bits on $AD_0$-$AD_7$ are address bits. This signal is used latch the lower order address from the multiplexed bus and generate a separate set of eight address lines $A_7$-$A_0$.
e) SID (Serial input Data):- In serial transmission, Serial data bit are received on this line.

Q.3 Explain PSW with suitable example.

Ans.:-

PSW (Program Status Word) :- The five flags of 8085 constitute the PSW, reflecting data conditions in accumulator. The bit wise position of flags in flag register/PSW is shown below.

| S | Z | X | AC | X | P | X | CY |
|---|---|---|---|---|---|---|---|

- Sign Flag (S):- The flag is used with signed number. In a given byte, if D7 is 1, the number is viewed as negative number and if it is 0, the number is considered as positive number. This flag is irrelevant for unsigned number.
- Z (Zero Flag):- It is set if the ALU operation result is '0' and the flag is reset if result is not '0'. Modified by result in accumulator as well as other register.
- AC (Auxiliary carry):- When carry is generated by $D_3$ and passed on to digit $D_4$ during arithmetic operation, the AC flag is set.
- P (Parity Flag):- After an arithmetic or logical operation, if the result has an even number of 1's, the flag is set.

- CY (Carry Flag):- It carry is generated as a result of arithmetic operation, CY flag is set, otherwise it is reset.
  E.g.
  If A = 54; B = E7
  After ADD B
  i.e.

```
   A=      01010100
 + B=      11100111
           ----------
         1/00111011
```

  Carry generated → CY= 1
  D7 is 0 → S = 0
  No of 1's in result are 5 i.e. odd no
  → P= 0
  Carry is not generated from $D_3$ to $D_4$ → AC = 0
  Result is not Zero → Z= 0
  The value of PSW is

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0<br>S | 0<br>Z | 0<br>X | 0<br>AC | 0<br>X | 0<br>P | 0<br>X | 1<br>CY |

Figure1 – PSW/Flag register; X indicates don't care condition.
If the don't care conditions X are assumed to be Zero.
PSW = 01

Q.4 Draw bus structure of 8085 and explain various buses.

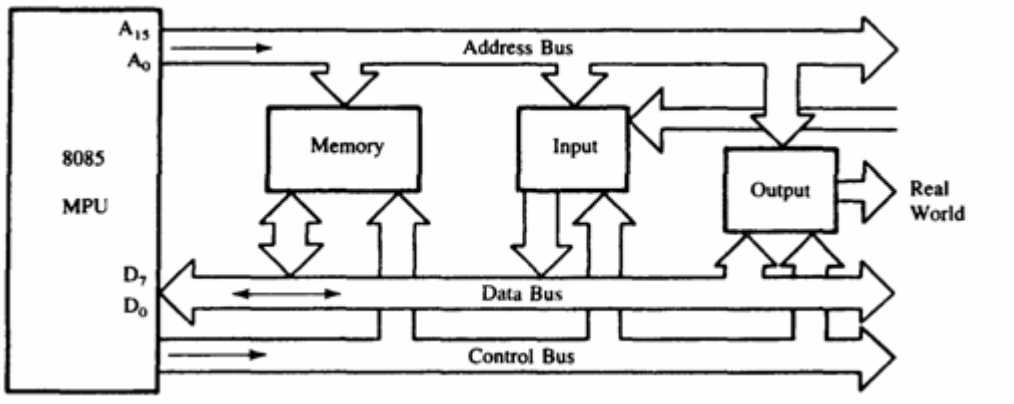



Figure 2- Bus structure of 8085

8085 has three buses: Data, Address and control Bus.

Data Bus:- is group of 8 bidirectional lines used to transfer data between microprocessor and memory (or I/O device). The data bus ($AD_0$-$AD_7$) is multiplexed with lower order address bus.

Address Bus:- is group of 16 lines that are used to send a memory address or device address from microprocessor to memory location or I/O device. It is unidirectional bus.

Control Bus:- are single line generated by the microprocessor to provide timing of various operations. The CPU sends control signal on the control bus to enable the outputs of addressed memory devices or I/O port devices. Some of the control bus signals are as follows:

1.Memory read
2.Memory write
3.I/O read
4.I/O write.