Date
Page

DHARMSINH DESAI UNIVERSITY , NADIAD

Faculty Of Technology

ONLINE SESSIONAL EXAMINATION

B.Tech (CE) sem 5th

SUBJECT : MICROPROCESSOR FUNDAMENTAL & PROGRAMMING

Roll No. CE136 Date : 24-Aug-2020

Signature : D.R.Vagh Time : 10:00 AM

Total Pages : 11

Q.2

(a) How does 8086 compute 20 bit Physical address?

→ 20 bit is calculated by segment and offset address.

PA = segment address x 10H + offset

(b) maximum size of ~~offset~~ segment is possible 2^{16} = 65,535 bytes (64 kb) because offset size is 16 bit. Therefore the max. size is 2^{16} memory location

- minimum size of segment is 16 because minimum value of segment is 16 memory location.

(c) - address is specified by CPU and can only be directed from the CPU towards other devices and not the reverse. The data bus is meant for the transfer of data.

- In mn / $\overline{mx}$ pin is high mn this mn used in single Process system.
- To demultiplexing The pin no = 30 of 8085 is used. it is called ALE pin (Address Latch enable).

(d) AAD is used to convert ASCII value to unpacked BCD value. The AAD is only useful when it precedes a DIV Instruction that divides the adjusted value in AX Register.

(e)

(f) 1- MOV [SI], [DI]

we can't transfer data to memory

2- MOV AX, DL

AX is 16 bit and DL is 8 bit so the operand size is not match.

Date
Page

Q.2

(c) Programmable peripheral interface (PPI)

IC - 8255 - It is used to manage automatically handshake operation. It can be programmed to receive an signal from a theriphenal and also send an iut signal and send the ACK signal back to theriphenal at proper time.

- IC - 8255's purpose is programmable I/O device which may be used with microprocessors.
- There are 24 I/o pins which may be individually programmed in 2 different groups of 12.
- 8255 is high performable and industry standard configuration that's why it is compatible with 8086
- It is used in 3 major modes of operation.

8255 can be configured in two modes : 1. BSR mode (Bit set Reset)
2. I/O Mode:

In Io mode three modes available. Mode 0,
Mode 1,
Mode 2.

- Modes are configured by control word

- BSR (Bit set Reset) Mode:

If bit 7 of control word is a logic 0 then 8255 will be configured as BSR mode.

In this mode we can set or Reset the pins.

D7	D6	D5	D4	D3	D2	D1	D0
0	x	x	x	Bit select			S/R

↓ BSR mode

↓ Set = 0

↳ set - 1
Reset - 0

↓

Bit 0
Bit 1
Bit 2
Bit 3
Bit 4
Bit 5
Bit 6
Bit 7

- There are mainly 3 working modes.

Mode 0 - Basic I/O

Mode 1 - strobed I/O

Mode 2 - bidirectional I/O

There is 3 Port A, Port B and Port C

Port	Mode 0	Mode 1	Mode 2
A	✓	✓	✓
B	✓	✓	X
C	✓	X	X

Basic I/O : will control signal from up and pes.

hand shak (strobed) I/O : It will ask up and 8285 that it is possible to transfer data so some data will not be lost.

bidirectional I/O : It is same as strobed I/O but It will work on both in and out bidirectiona

(b)

1. 128kb EPROM using 32K devices.
2. 128kb SRAM using 16K devices.

→ In EPROM required size is 128kb and chip size is 32kb.

so, the required chips

$$n_1 = \frac{128}{32} = 4 \text{ chips}.$$

- Number of sets,

$$\frac{4}{\text{no. of banks}} = \frac{4}{2} = 2$$

- size of set
  = 2 * 32
  = 64 kb

- for EPROM

		Starting address	Ending address
set 2	→	F0000h	FFFFFh
set 1	→	E0000h	EFFFFh

so, we can said the ending address is starts from FFFFFh.

∴ start address = FFFFFh − 0FFFFh
= F0000h

∴ ending address of set 1
= start of set 2 − 1
= EFFFFh − E0000h
= EFFFFh

∴ start address of set 1
= EFFFFh − 0FFFFh
= E0000h

→ In SRAM required size is 128 Kb and chip size is 16 K.

∴ The required chips
$n_2 = \frac{128}{16} = 8$

∴ Number of sets
$= \frac{8}{\text{no. of banks}} = \frac{8}{2} = 4$

∴ size of sets,
= 2 * 16
= 32 Kb

- for SRAM

	Starting address	Ending address
set 1 →	00000h	07FFFh
set 2 →	08000h	0FFFFh
set 3 →	10000h	17FFFh
set 4 →	18000h	1FFFFh

∴ starting address = 00000h

set 1 Ending address = 00000h + set-size
= 00000h + 32 kb
= 07fffh

Q.3

(a) To find the series in AP.

```
data segment
        terms db 50 DUP (0)
        a db ?
        b db ?
        n db ?
data ends

code segment
assume cs : code , ds : data
        MOV ax, data
        MOV ds, ax
        MOV ah, 1
        int 21h
        AND al, 0fh
        MOV a, al
        MOV ah, 1
        int 21h
        AND al, 0fh
        MOV b, al
        MOV ah, 1
        int 21h
        AND al, 0fh
        MOV n, al
```

```
        lea   SI, terms

        MOV   [SI], a
        inc   SI

        MOV   cl, 02h
        XSR   ch, ch

continue:  MOV   al, cl   ;  n
           dec   al       ;  n-1
           MUL   b        ;  (n-1)*b
           add   al, cl   ;  a + (n-1)*d
           MOV   [SI], al
           inc   SI
           cmp   n, cl    ;  if (cl==n)
           JNZ   continue

           int   03h
code ends

end.
```

X —— END OF ANSWER SHEET —— X