

a) Macros →

When repeated group of insts. is too short to not appropriate to be written as procedure, we use macros.

"It is group of insts. that perform a task. Each time, we call macro in our program the assembler will insert macros in place of call."

b) Procedure →

Whenever we want to execute certain set of insts. repeatedly throughout the program, we can avoid it by two ways -

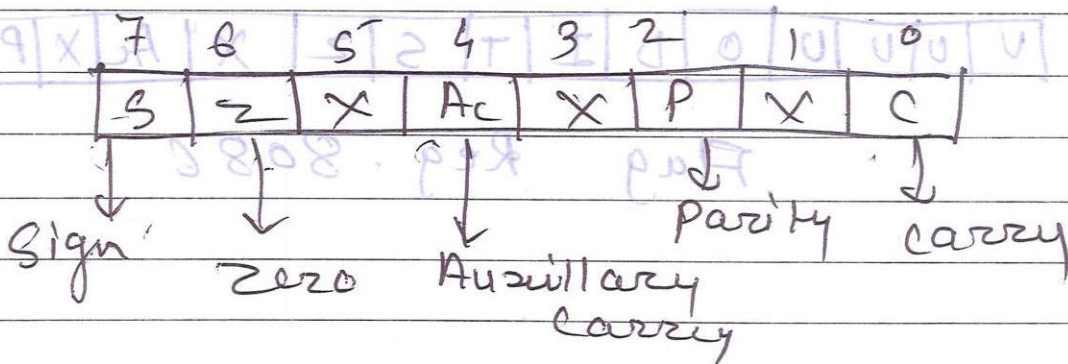
- ① by writing procedure
- ② by macros.

Two types →

- ① Re-entrant Proc.
- ② Re-cursive Proc.

Two directives

- ① PROC
- ② ENDP.



Q1d)

8088	8086
i) 4 Byte Instr queue	i) 6 byte Instr queue
ii) Data bus - 8bit	ii) Data Bus - 16 bit
iii) BHE / ST	iii) SS
iv) A8 - A15	iv) AD8 - AD15

Q2a)

① Type 0
Divide by zero

② Type 1
Single Step Interrupt

③ Type 2
NMI

④ Type 3
Breakpoint Interrupt

⑤ Type 4
Interrupt on overflow

⑥ Type

b)

V	U	U	U	O	D	I	T	S	Z	X	A	X	P	X	C
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Flag Reg. 8086

EPROM

Total EPROM req. = 64 KB

available chip size = 16 KB.

∴ No. of chips = 4

No. of sets = 2

RAM

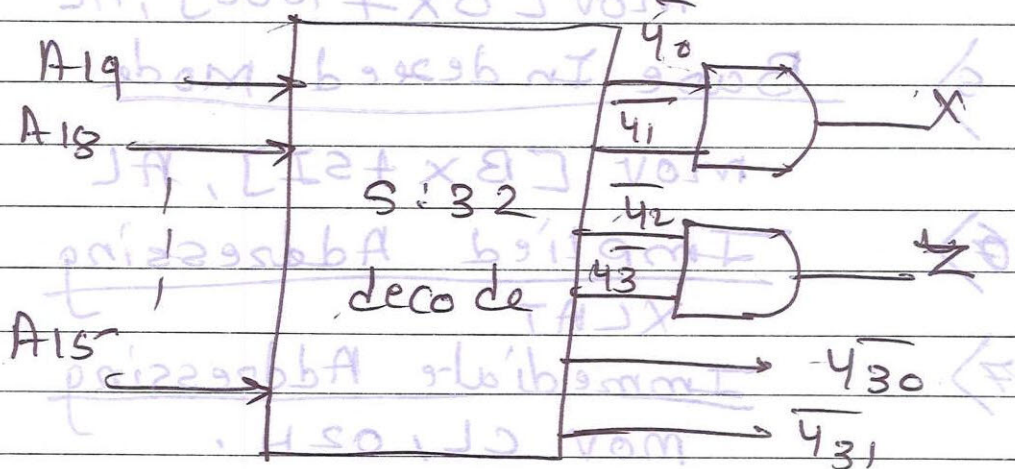
chip size = 32256 = 32 KB.

Req. RAM = 128 KB.

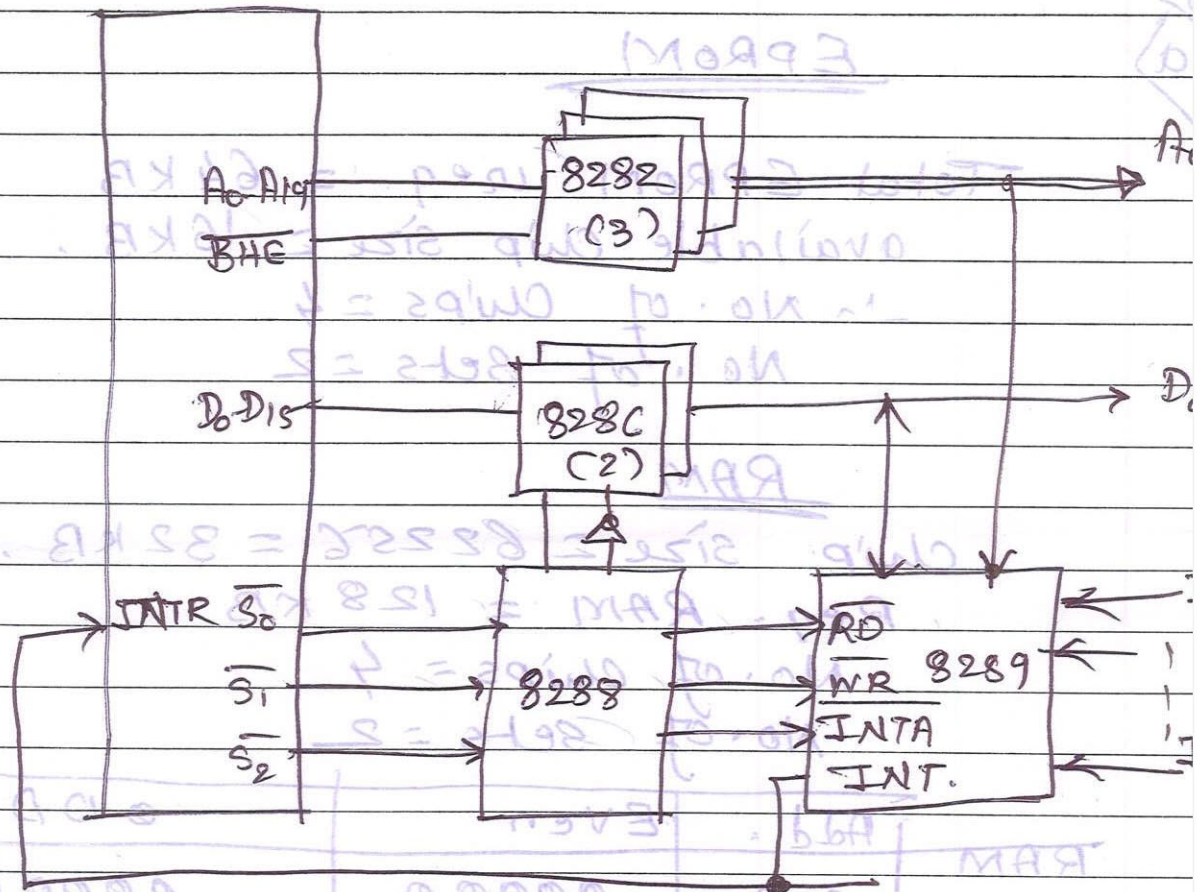
No. of chips = 4

No. of sets = 2

	Add.	Even	ODD.
RAM Set-1	S.A.	000000	00001
	E.A.	00000 0FFFF	00001 0FFFF
RAM Set2	S.A.	100000	10001
	E.A.	1FFFF	1FFFF
ROM Set1	S.A.	F8000	F8001
	E.A.	FFFFE	FFFFF
ROM Set2	S.A.	F0000	F0001
	E.A.	F7FFE	F7FFF



Q3) b)



Q4) a) 1) Register Addressing mode
`mov AX, BX`

2) Direct Addressing Mode
`mov [1234], AL`

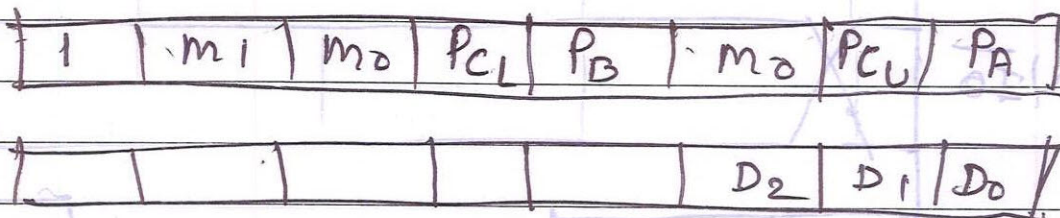
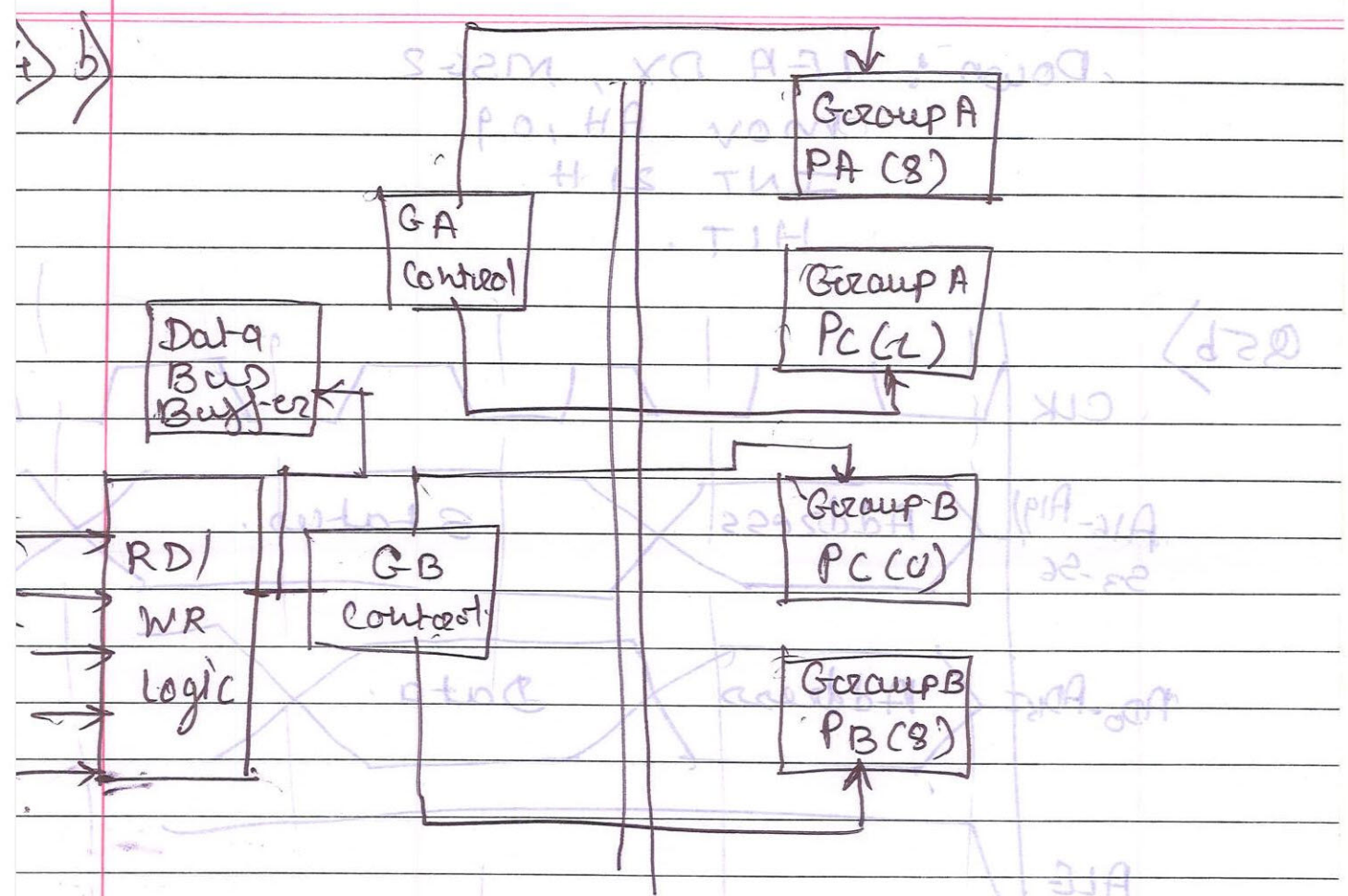
3) Register Indirect Add.
`mov [SI], AL`

4) Register Relative Mode
`mov [BX + 1000], AL`

5) Base Indexed Mode
`mov [BX + SI], AL`

6) Implied Addressing
`XLAT`

7) Immediate Addressing
`mov CL, 02H.`



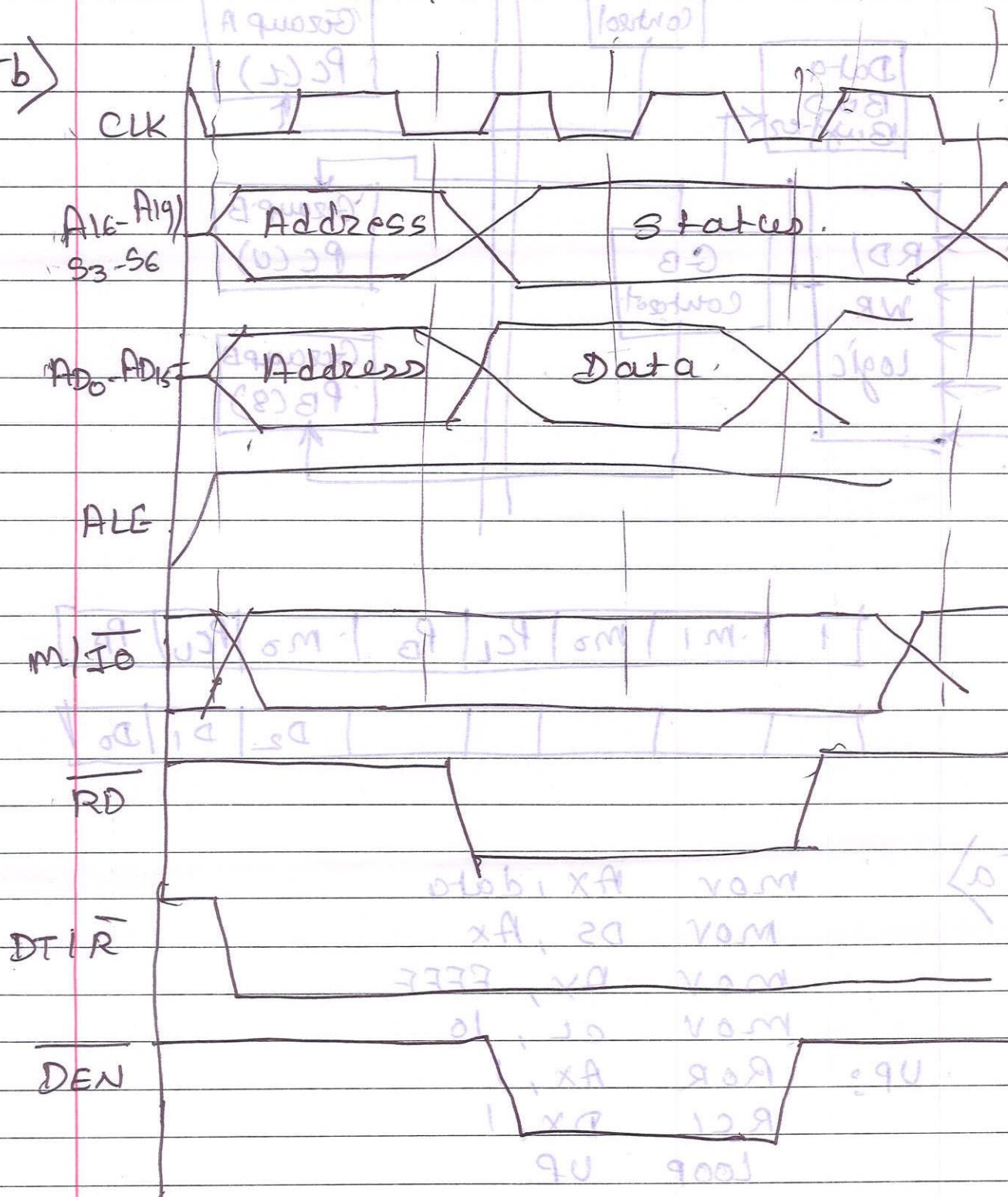
a)

```

mov  AX, data
mov  DS, Ax
mov  AX, FFFF
mov  CL, 10
UP:  ROR  AX, 1
     RCL  DX, 1
     LOOP UP
     CMP  AX, DX
     JNZ  Down
     LEA  DX, MSG
     MOV  AH, 09
     INT  21H
     JMP  Exit
  
```

Down: LEA DX, MSG2
 MOV AH, 09
 INT 21H
 HLT

Q5b)



Q6 a)

TEST →

TEST pin gives status of a wait state required to synchronise the speed between two different devices.

READY →

This pin gives the status of 8086 whether it is ready to receive data from peripheral.

RESET →

Reset pin of a 8086, resets the activity to the beginning of the process.

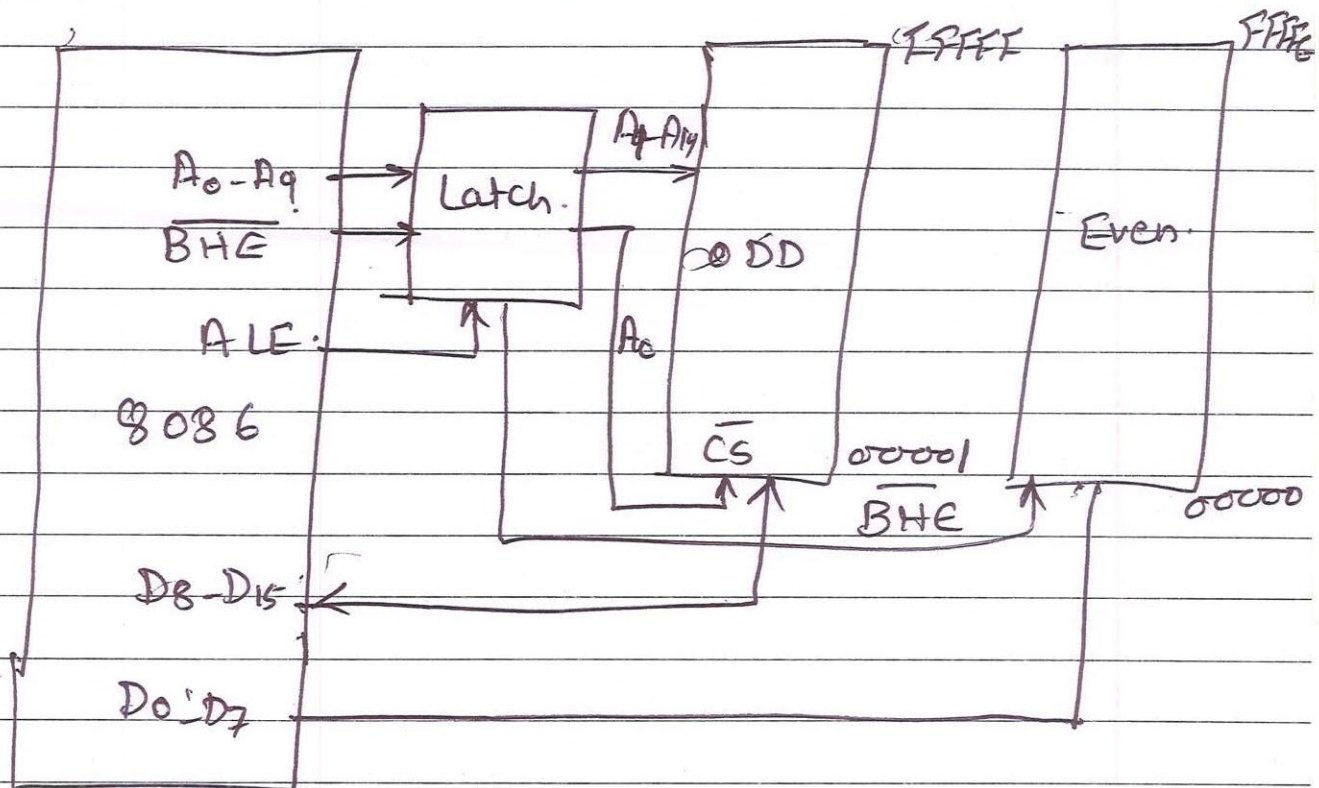
b) Memory Banking

There are two types of bank

① even bank &

② odd bank.

Even bank contains all even addresses whereas odd bank contains all odd addresses.



Q6) c) As the longest & largest instruction that 8086 can have is of 6 byte. Hence,

Q6) d) Assembly Directives → These are the statements that direct the assembler to do something. Following are some examples.

- CODE

This indicates beginning of code segment.

DB

Define byte type variable

