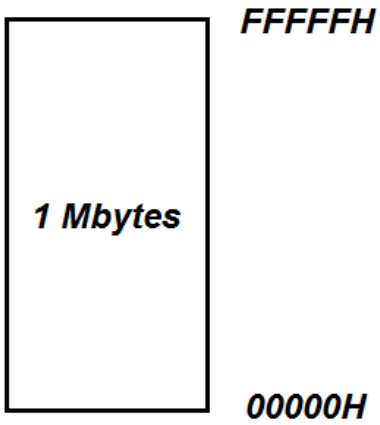
Sheet 1 solution

A) 1)

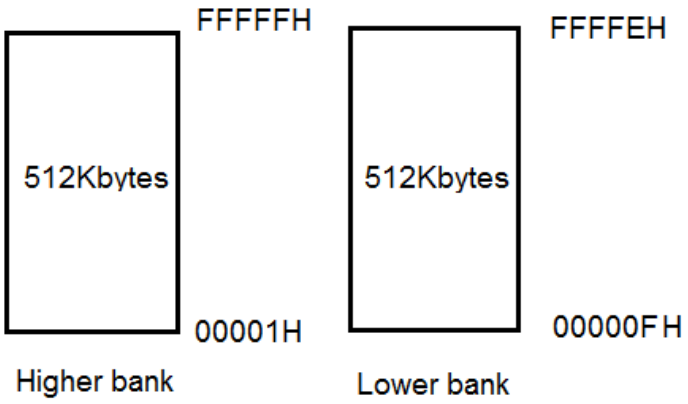
8086	8088
15 lines D_o---D_{15}	8 lines D_o-----D_7
Can address 1 word or 1 byte of data	Can address 1 byte of data
To transfer 16 bits from/to memory it needs 1 operation.	To transfer 16 bits from/to memory it needs 2 operations.

Logical memory is the same for 8086 and 8088:



Logical memory for 8088/8086

- Physical memory for 8088 is the same as its logical memory.
- Physical memory for 8086 is different than its logical memory as it consists of 2 banks each with size 512Kbytes.



Physical memory for 8086

2) 20 address lines A_o to A_{19}.

Memory locations: 1M= 2^{20} locations.

I/O devices that can be accessed: 2^{16}.

3) a) 8086:

Address bus: A_0 to A_{19} (20lines).

Data bus: D_0 to D_{15} (16 lines).

b) 8088:

Address bus: A_0 to A_{19} (20 lines).

Data bus: D_0 to D_7 (8 lines).

4) Control bus function:

- Controls whether the operation a write or read operation.
- Controls whether the memory or an I/O device is accessed.

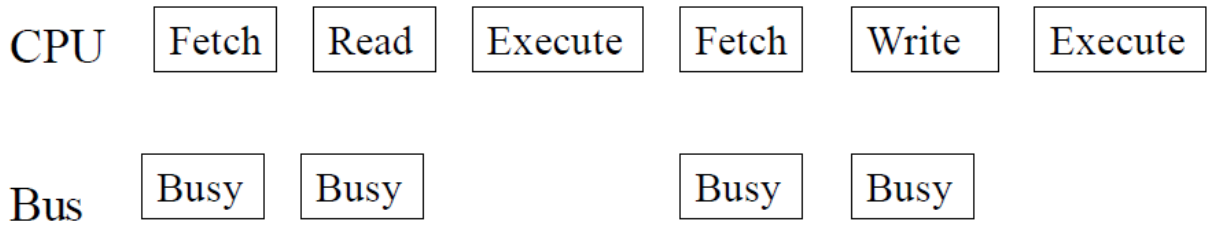
5) 1Kbytes =2^{10} =1024 bytes.

Nibble= 4 bits , byte= 8 bits , word= 16 bits ,

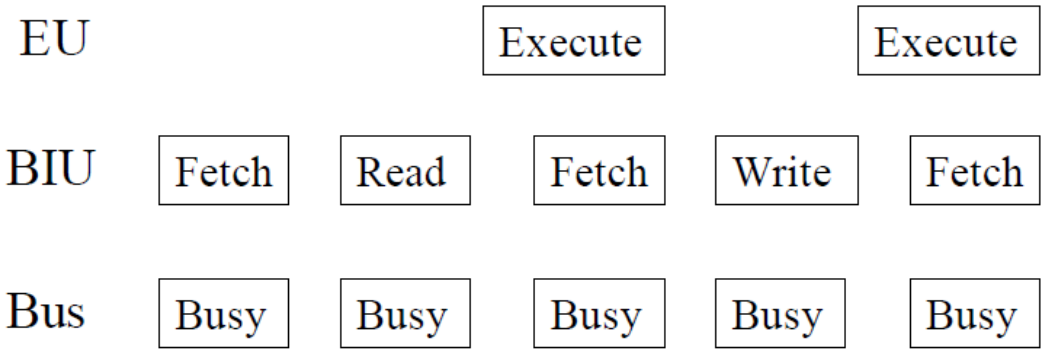
double word= 32 bits.

6)

The normal operation of 8085 (non-pipelined execution) is :



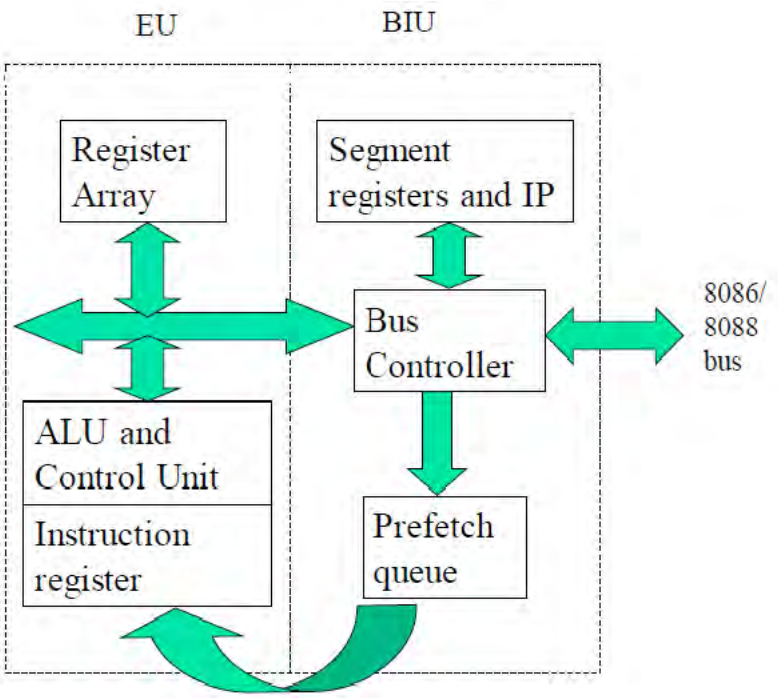
The operation of 8088/8086 (pipelined execution) is:



The advantages of pipelining is :

- Keeping the bus busy all the time means higher efficiency.
- Faster than non-pipelined execution as Both BIU and EU can be working simultaneously without waiting for the completion of the other task (pipelined parallel processing).

7) The basic internal architecture of 8086/8088 microprocessors:



B) 1)

- 16KB= 2^{14}, thus it requires 14 address lines.
- The highest address is 3FFFH (14 ones).

2) The word stored in B00H is FF00H.

3)

00	0A03H
56	0A02H
7C	0A01H
FF	0A00H

C) The primary functions of these registers include:

AX (Accumulator)

- Used with the arithmetic and logic operation.
- Used with the I/O devices

BX (Base)

- Hold the base address of data located in the memory.
- Hold the base address of a table of data referenced by the translate instruction (XLAT).

CX (Count)

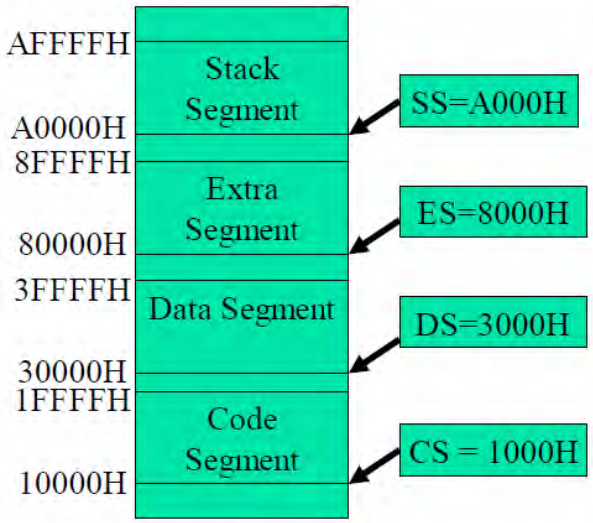
- Used as a counter for certain instructions such as shift rotate and loop.
- Used as a counter for the string operations.

DX (Data)

- Used with the arithmetic instruction such as 16-bit multiplication and division.
- Holds the I/O port number for a variable I/O instruction.

D)

1)Code segment ⇒ stores the program or code.
2)Data segment ⇒ stores the data.
3)Stack segment ⇒ stores data as LIFO stack.
4)Extra segment ⇒ used for string instructions.



E) a) Can be coded.

b) Cannot be coded.

c) Cannot be coded.

d) Can be coded.

e) Can be coded.

f) Cannot be coded.

g) Can be coded.

h) Can be coded.

i) Cannot be coded.

F)

94C2	1001 0100 1100 0010
+	+
323E	0011 0010 0011 1110
0C00	1100 0111 0000 0000

CF=0 AF=1 SF=1 ZF=0 PF=1

G)

1) True.

2) False because the prefetch queue is a FIFO queue 2 bytes wide and 3 locations deep for 8086 while for 8088 it is 1 byte wide and 4 locations deep.

3) True.

Sheet #2

1- a- 4 Segments each of 64KB making 64KBX4 = 256 KB

b- 1- Code segment
2- Data segment
3- Stack segment

2- a- PA = CS X10H + IP = 20000 + 1000 = 21000H

b- PA = SSX10H + SP = 12440H

c- PA = DSX10 + BX = 10010H

d- PA = 0100 X10 + 1234 = 02234H

3- BX = 4F56H ⇒ BL = 56H & BH = 4FH
SP = 0100 , SS = 0200H

SP is at = 0200X10 + 0100 = 02100H

SP now is at 02FE

SP 02100		
02FF	4F	
02FE	56	

4- a- PA = DSX10 + 2000 = 04000H

b- PA = DSX10 + BX = 02200H

c- PA = DSX10 + DI = 02300H

5- a- Logical address = CS:IP = 24F6:634AH

b- PA = CS x 10 + IP = 2B2AAH

c- Lower Range - CS x 10 = 24F60H

d - Upper Range = CS x 10 + FFFF = 34F5FH

6- BX contains 2248H

7- a- PA = SS x 10 + SP = 35000 + FFFE = 44FFEH

b - LA = 3500 : FFFE

c - Lower Range = SS x 10 = 35000H

d - Upper Range = SS x 10 + FFFF = 44FFFH

8- Lower Range = CS x 10 = FFFF0H

Upper Range = CS x 10 + FFFF = 0FFEFH

9- Number of bytes = FFFF - SP + 1 = 100H Bytes
= 100H/2 Words
= 80H words
= 128 words

10- a- True
b- True
c- False
d- False, no POP CS
e- False

11- a- DS = 3245 H

b- SI = 100 H

c- BX = 0004 H

d- PA = DS x 10 + SI = 32450 + 100 = 32550 H

AX = 3245 H + [Content of 32550 H & Content of 32551]

= 3245 + 6A34 = 9C79 H

12-

		START	END
a-	Code Segment	CS x 10 = 12000 H	CS x 10 + FFFF = 21FFF H
	Data Segment	DS x 10 = 10000 H	DS x 10 + FFFF = 1FFFF H
	Stack Segment	SS x 10 = 10600 H	SS x 10 + FFFF = 205FF H
	Extra Segment	ES x 10 = 10300 H	ES x 10 + FFFF = 202FF H

b- Top of stack = SS x 10 + SP = 10A00 H

Bottom of stack = SS x 10 + FFFF = 205FF H

c- # of bytes = FFFF - SP + 1 = 64512 Bytes

d- PA = CS x 10 + IP = 12100 H

13 a- $10_d = 1010 = 1.010\ 2^{3}$ exp - 127

Short format $2^3 = 2$

exponent = 127 + 3 = 130 ⇒ 10000010

10_d = 0 100 0001 0010 0000000000000000000

= 41200000H

Long $2^3 = 2^{exp - 1023}$ ⇒ exponent = 1026

$1026 = 1000000010$

$10d = 0\,1000000\,0010010\ldots\ 0$

$= 4024000000000000H$

Solution Sheet 3

1-a- Register addressing mode
S: BL D: DL

b- Immediate addressing mode
S: 00FF D: AX

c- Register indirect addressing mode
S: AX D: PA = DS x 10 + DI

d- Register indirect addressing mode
S: PA = DS x 10 + SI D: DI

e- Register relative addressing mode
S: CX D: PA = DS x 10 + BX + XY2

f- Register relative addressing mode
S: AH D: PA = DS x 10 + DI + XY2

g- Base relative plus index addressing mode
S: AL D: PA = DS x 10 + BX + DI + XY2

2- a- Not of the same size
b- No data movement between segment registers.
c- No movement between memory locations.
d- Confusion whether 10H is a byte or a word: MOV Byte PTR [BX], 10H
or MOV Word PTR [BX], 10H

4- a - Base relative plus index addressing mode.

PA = SSX10 + BP + DI + LABEL = 20000 + 1000 + 100 + 1234
= 22334H

b - Registo indirect addressing mode

PA = DSX10 + DI = 10000 + 100 = 10100H

c - Register indirect addressing mode
PA = SSX10 + BP = 20000 + 1000 = 21000H

6- a - 8915 H
b - 03C2 H
c - 8918 H
d - 885600 H
e - 8A5710 H
f - 1EH
g - D2C3H
h - C7C03412 H
i - 8B160010 H
j - C7070010 H

7- MOV SI, 0100H
from table 1011 W Reg Data
1011 1 110 0000 0000 0000 0001
= BE 0001H Save into three Bytes

8- Program needs 25 Bytes of memory

Sum up the machine codes of each instruction

Sheet 4 solution

1) a) Copy 03FFH to BX, BX=03FFH.
   b) Copy the address of DATA to AX.
   c) Copy the two bytes addressed by LIST into DI and the next two bytes to DS.
   d) Copy the two bytes addressed by LIST into BX and the next two bytes to ES.
   e) Save the flag register into stack.
   f) Read the data from device with port number 0EEH and put it in AL.

2) a) Mov BX,DATA : BX=DATA , LEA BX,DATA : BX=address of DATA.
   b) In AL,10H : Loads data from I/O device at port number 10H into Al.(Direct (fixed) port addressing).
   In AL,DX : loads data from I/O device at port addressed by DX into Al(indirect port addressing).
   c)same as (b)

3) Mov BX,5000H
   Mov ES,BX
   Mov DI,0000H
   Mov CS,1000H
   CLD
   Mov AL,77H
   Rep StosB

4) Mov Ax,000H
   Mov DS,AX
   Mov ES,AX
   Mov SI,340H
   Mov DI,360H
   Mov CX,[400H]
   CLD
   Rep Movsb

5) Mov BX,0B000H
   Mov DS,BX
   Mov ES,BX
   Mov SI,200H
   Mov DI,600H
   Mov CX,0FAH
   CLD
   Rep Movsb

6) a)
   Mov AX,000
   Mov DS,AX
   Mov [380H],0d
   Mov [381H],1d
   Mov [382H],4d
   Mov [383H],9d
   Mov [384H],16d
   Mov [385H],25d
   Mov [386H],36d
   Mov [387H],49d
   Mov [388H]64d

   b)i)
   Mov AX,000
   Mov DS,AX
   Mov Bx,380H
   Mov AL,[400H]
   XLAT
   Mov [401H],AL

   ii)Same as before.

c)

```
Mov Ax,000
Mov DS,AX
Mov BX,380H
Mov AL,[400H]
XLAT
Mov CL,AL
Mov AL,[401H]
XLAT
ADD AL,CL
Mov [402H],AL
```

7) a) Resisters are not of the same size.
   b) You should add WORD PTR or BYTE PTR to determine whether to increment a byte or a word.(conceptual not a syntax error)
   c) A BYTE PTR or WORD PTR is required to determine whether the addition is done to a byte or a word.

8)

a)

AX=0100H=16d

BL=FDH=-3d

IDIV BL : $AX/_{BL} = 16/_{-3} = -5\frac{1}{3}$

AL=-5d=FBH

AH=1d=1H

AX=01FBH

Note:

AL (or quotient) has the same sign of dividend (AX).

AH (or remainder) has a negative sign when either the divisor or dividend has a negative sign.

b)

AX=-16d

BL=3d

IDIV BL : $AX/_{BL} = -16/_{3} = -5\frac{1}{3}$

AL=-5d=FBH

AH=-1d=FFH

9)

a) AND BX,DX
b) AND DX,[SI]-8
c) OR BP,1122H
d) OR [when],AH
e) XOR [BX],DX
f) XOR DX,[BP]+3C

10)

a) ADD AL,BL
   ADD CL,AL
   ADD DL,CL
   ADD AH,DL
   MOV DH,AH

b) ADD DI,SI
   ADD DI,BP
   SUB AX,DI
   MOV BX,AX

c) MOV AL,CL
   MUL CL
   MUL CL
   MOV CL,AL

d) MOV AL,BL

```
CBW
DIV   CL
MOV  DL,AH
MOV  DH,2H
MUL  DH
MOV  BL,AL
MOV  AL,DL
MUL   DH
MOV  BH,AL
```

e)
```
CBW
MOV   BX,AX
```

f)
```
MOV   BH,7H
MUL    BH
MOV   CX,AX
MOV   BH,5H
MOV   AL,BL
MUL    BH
SUB    CX,AX
MOV   AX,DX
MOV   BH,8H
DIV    BH
SUB    CX,AX
XCHG  AX,CX
```

g)
```
OR    AX,000FH
AND  AX,0FFFH
XOR   AX,0380H
```

h)
```
MOV  AL,66H
MOV CX,NUMBER OF BYTES
CLD
```

```
REPNE  SCASB
```

11)
```
MOV   AX,[100H]
ADD    AX,[200H]
MOV   [300H],AX
MOV   AX,[102H]
ADC    AX,[202H]
MOV   [302H],AX
```

12)
```
MOV   AL,[100H]
MOV   BL,AL
AND    AL,0FH
AND    BL,0F0H
MOV   CL,O4H
SHR    BL,CL
ADD    AL,BL
```

Another solution:

```
MOV  CL.[0100H]
MOV  AL,CL
AND  CL,0FH
AND  AL,0F0H
MOV  BL,10H
CBW
DIV   BL
ADD  AL,CL
```

13)
```
MOV   SI,340H
MOV   DI,340H
MOV   CX,[401H]
MOV   [DI],00H
```

```
AGAIN: MOV  AL,[SI]
       ADD   [DI],AL
       INC   SI
       LOOP  AGAIN
```

14)
```
       MOV   CX,[401H]
       MOV   SI,340H
       MOV   DI,400H
       MOV   [DI],00H
       AGAIN: MOV  AX,[SI]
       ADD WORD PTR[DI],AX
       ADD   SI,2
       DEC   CX
       JNZ   AGAIN
```

15)
```
       MOV   AX,N
       MOV   CX,N-1
       MOV   BX,N
       AGAIN: DEC   BX
       MUL   BX
       LOOP AGAIN
```

Another solution:
```
       MOV   CX,N-1
       MOV   AX,N
       AGAIN: MUL   CX
       DEC   CX
       JNZ   AGAIN
```

16)
```
       MOV   SI,340H
       MOV   AL,00H
```

```
MOV   CX,0AH
AGAIN: MOV   [SI],AL
INC  AL
INC  SI
LOOP AGAIN
MOV  CX,0AH
MOV  SI,340H
AGAIN2: MOV  AL,[SI]
ADD  [SI],AL
INC    SI
LOOP  AGAIN2
```

17) a) SHR DI,1
b) SHL AL,1
c) ROL SI,1
d) RCR DX,1
e) SAR DH,1

18) a)AX=1234H
b)AX=1234H , BX=1234H
c) Since AX= 0001 0010 0011 0100
not AX=1110 1101 1100 1011
then AX=EDCBH
d) AX=0000H
e)AX=1234H
f)AX=1234H, CX=0000H
g)CL=02, AX=1234H
h) since AX=0001 0010 0011 0100
then SHL AX,CL
AX= 0100 1000 1101 0000
=48D0H
i)AX=0001 0010 0011 0100
=1234H

j)AX=0100 1000 1101 0000
=48D0H
CX=0002H

19) MOV DX,3000H
MOV DS,AX
MOV CX,63H (63h=99d)
Y:MOV DL,CL
MOV SI,0B50H
X:MOV AL,[SI]
CMP AL,[SI+1]
JC NO-SWAP
XCHG AL,[SI+1]
MOV [SI],AL
NO-SWAP: INC SI
DEC DL
JNZ X
DEC CL
JNZ Y

20) a) 4d=0100
then SHL AX,2
b) 10d = 1010
SHL AX,1
MOV BX,AX
MOV CL,2
SHL AX,CL
ADD AX,BX

Hardware Portion

Sheet # 1

1. a- ALE is a µP pin which determines whether the bus carries an address (ALE = 1) or data (ALE = 0)

b- They are needed for
1- Multiplexing/Demultiplexing buses
2- Provide required current for proper operations.
3- Isolation between µP and other chips.

c- Refer to class notes.

d- To match between the number of pins of µP and those of memory chips.

e- 8086 has 20 address lines $\Rightarrow \frac{2^{20}}{2^{16}} = 2^4 = 16$ chips
8088 has 20 address lines $\Rightarrow 2^{20}/2^{16} = 16$ chips.
80486 has 32 address lines $\Rightarrow 2^{32}/2^{16} = 2^{16} = 65,536$ chips
Pentium has 32 address lines $\Rightarrow 2^{32}/2^{16} = 65,536$ chips

f&g Refer to class notes

---

2. 1st page C0000 – C07FF H
2nd C0800 – C0FFF H
3rd C1000 – C17FF H
4th C1800 – C1FFF H
5th C2000 – C27FF H
6th C2800 – C2FFF H
7th C3000 – C37FF H
8th C3800 – C3FFF H

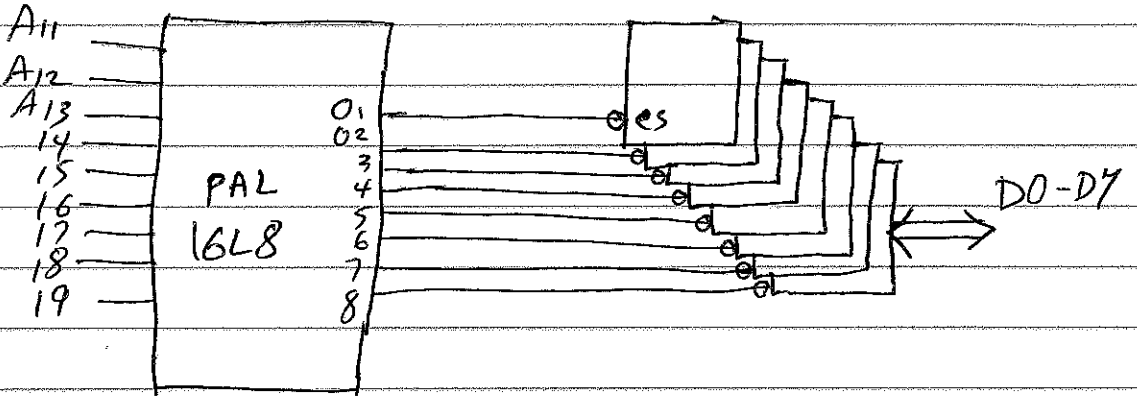
using PLA (16L8) decoder.

$\bar{O}_1 = A_{19} * A_{18} * \bar{A}_{17} * \bar{A}_{16} * \bar{A}_{15} * \bar{A}_{14} * \bar{A}_{13} * \bar{A}_{12} * \bar{A}_{11}$

$\bar{O}_2 = \_\_\_\_\_ * \bar{A}_{13} * \bar{A}_{12} * A_{11}$

$\vdots$

$\bar{O}_8 = \_\_\_\_\_ * A_{13} * A_{12} * A_{11}$



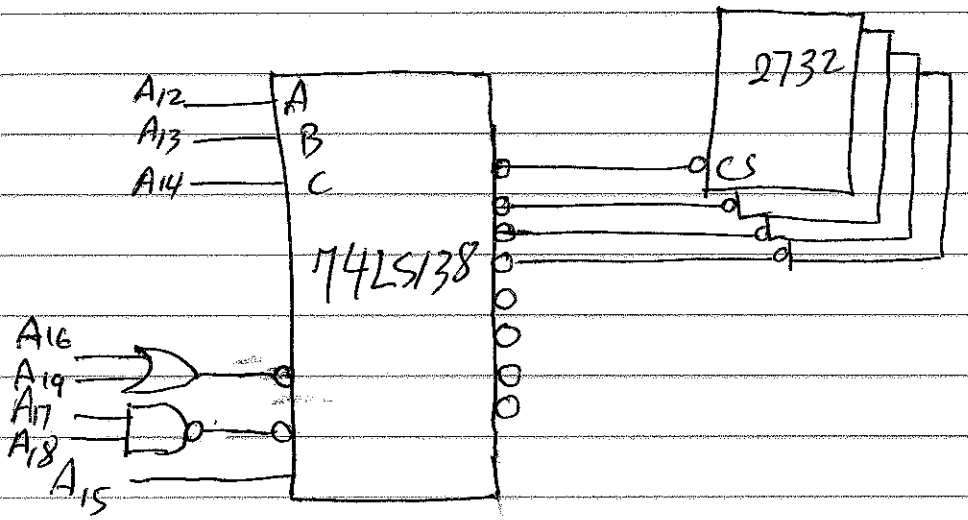
3- 2732 chip is organized as 4K x 8, so it is a 4KB chip. which needs, 2^{12}, 12 address lines from $A_{11} - A_0$.

Given address range = 6BFFF − 68000 + 1 = 4000 = 16KB

Required # of chips = $\frac{16KB}{4KB}$ = 4 Chips.

$A_{19}=0$, $A_{18}=1$, $A_{17}=1$, $A_{16}=0$, $A_{15}=1$

A_{14} A_{13} A_{12} select the active chip.



4- 2732 chip is 4K x 8 needs $A_{11} - A_0$, we require 3 of them.

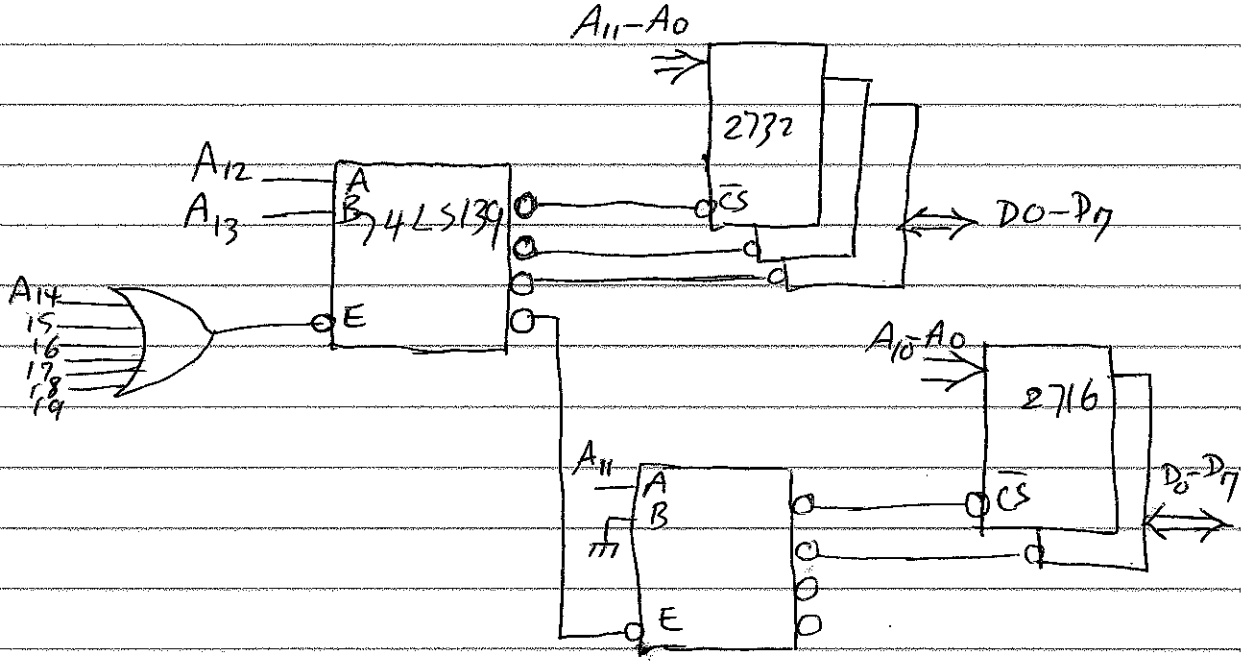
2716 is 2K x 8 needs $A_{10} - A_0$, 2 are required

So, we need to use 2-4 decoder 74LS139

Address range: 00000 → 03FFF

which means $A_{19} \rightarrow A_{14}$ are all zeroes.

chip	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	----	A_1	A_0
1(2732)	0	0	0	0	0	0	0	0	X	X	----	X	X
2(2732)	0	0	0	0	0	0	0	1	X	X	----	X	X
3(2732)	0	0	0	0	0	0	1	0	X	X	----	X	X
1(2716)	0	0	0	0	0	0	1	1	0	X	----	X	X
2(2716)							1	1	1	X	----	X	X



5- a- 2732 = 4K x 8 ⇒ 2^{12} ⇒ $A_{11} \rightarrow A_0$

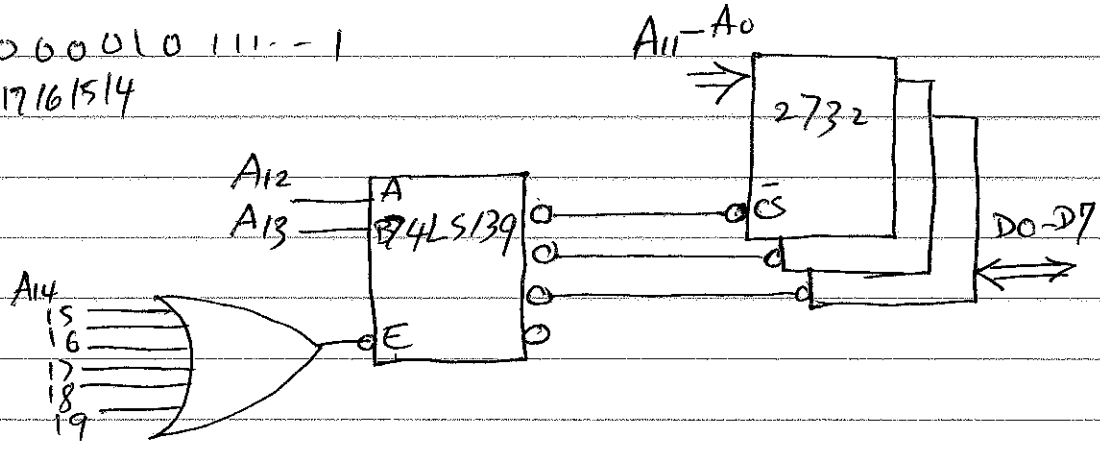
Address range = 02FFF - 00000 + 1 = 3000 = 12KB

∴ # of chips = $\frac{12KB}{4KB}$ = 3 chips (2-4decoder)

S 00000000 ---- 0

E 00000010 111 ---- 1

A_{19} 18 17 16 15 14



b- RAM

$4016 = 2K \times 8 \Rightarrow 2^{11} \Rightarrow A_{10} \rightarrow A_0$

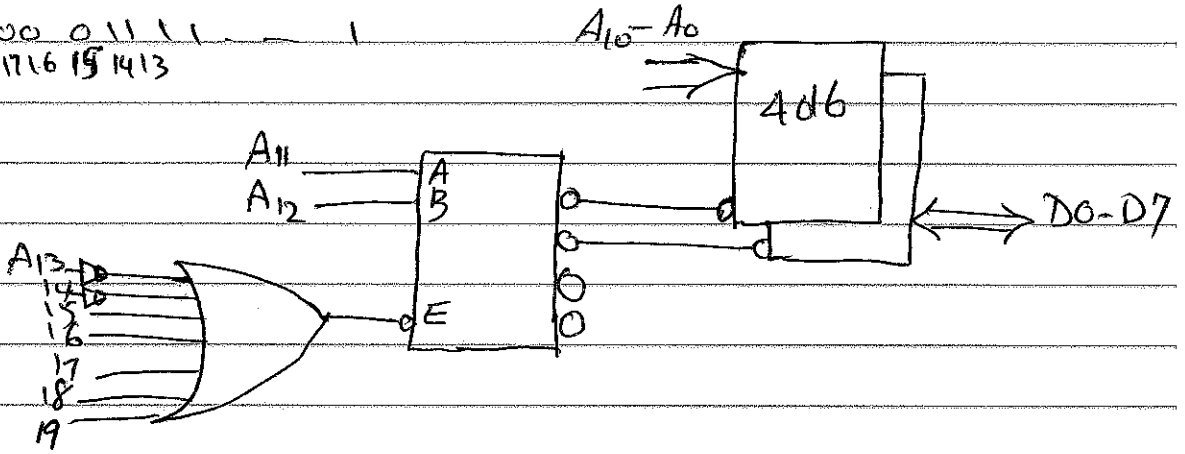
Address range = 06FFF − 06000 + 1 = 1000 = 16^3 = 4KB

# of chips = $\frac{4KB}{2KB}$ = 2 chips (2-4 decoder)

S 00000 110 0 - - - 0

E 00000 011 1 - - - 1

19 18 17 16 15 14 13



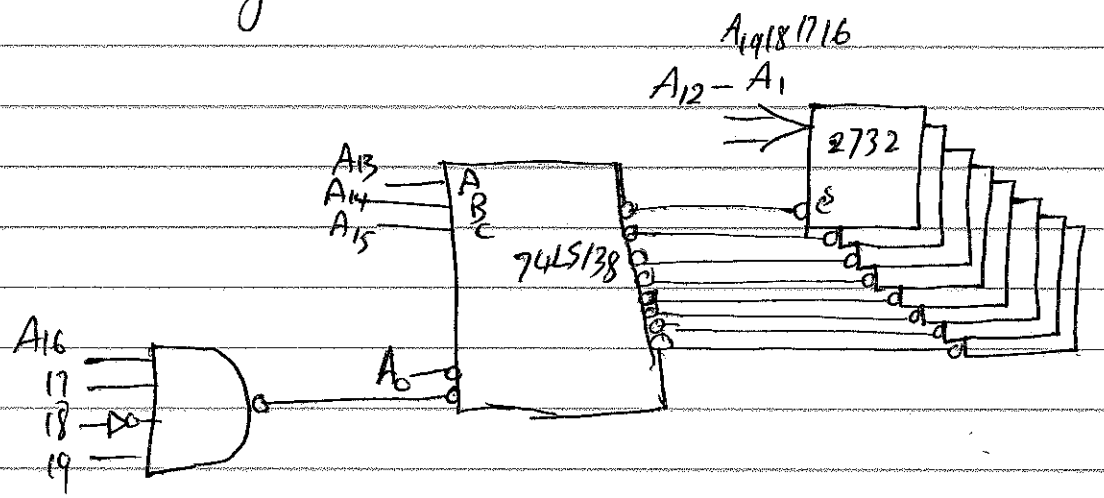
6- 2732 = 4K x 8 $\Rightarrow 2^{12} \Rightarrow A_{11} - A_0$

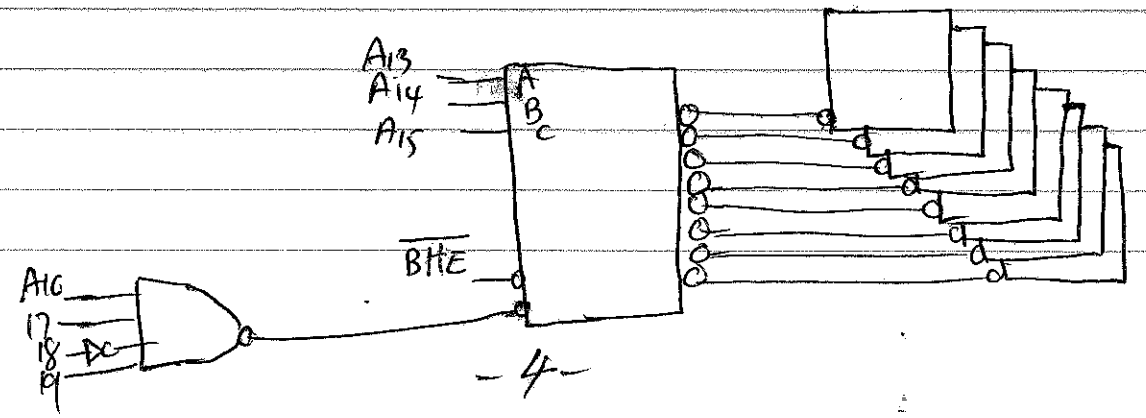
# of chips = $\frac{64KB}{4KB}$ = 16 chips

A0 is used for odd and even chips

Make 8 odd & 8 even chips

Address range starts at B0000 $\Rightarrow$ 1011 0 - - - 0





7- a- See lecture notes

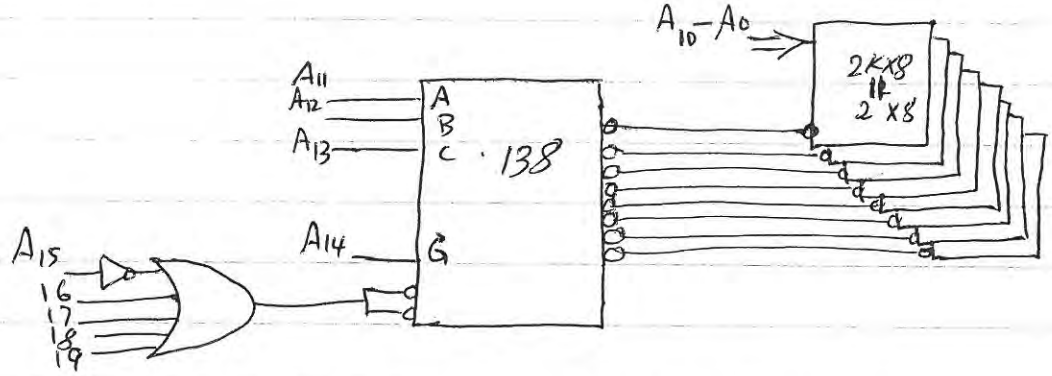
Maximum data rate = 1.25 MB/Seconds

b- ROM0: Total Memory Size = 0FFFF - 0C000 + 1 = 4000H
= 16 KB

# of chips required is 16KB/2KB = 8 chips

1- 74LS138 decoder

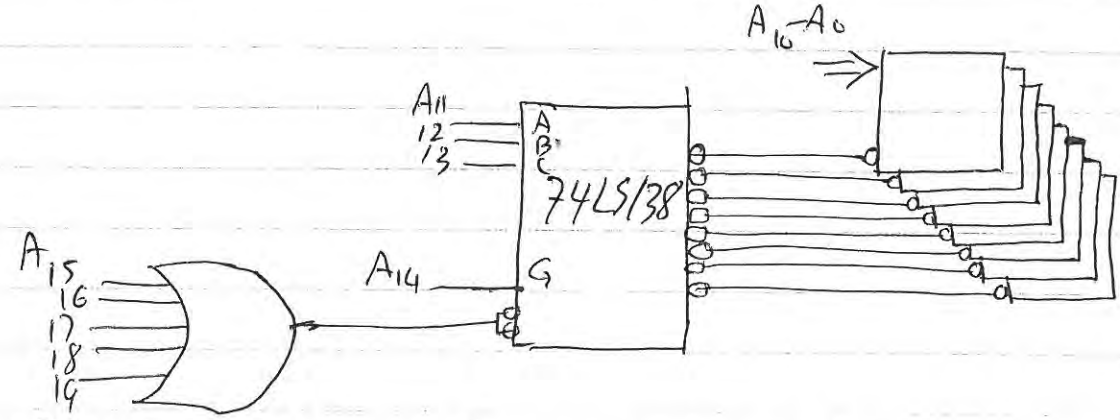
Address range: 0000 1100 0000 0000 0000 → 0000 1111 1111 1111 1111



RAM1: Memory Size = 07FFF - 4000 + 1 = 4000H

C B A = 16 KB = 8 chips

Address Range: 000 0100 0000 0000 0000 → 0000 0111 1111 1111 1111



RAM0 Memory Size = 03FFF + 1 = 04000 = 16KB
= 8 chips

Similar to RAM1 except that A14 has to be inverted before connecting to G.

8- a- Examining the circuit, we find that:

RAM 4016

Address Range: $A_0 \rightarrow \overline{LWR}$; A_1 to $A_{11} \rightarrow A_0$ to A_{10}

A_{12} to $A_{13} \rightarrow$ A & B of 74LS 139

Lowest address: 19 18 17 16 15 14 13 $\rightarrow$ 0 : 00000

0 0 0 0 0 0 0

Required to be 10000, therefore $0\,0\,0\,1\,(0000)_H$

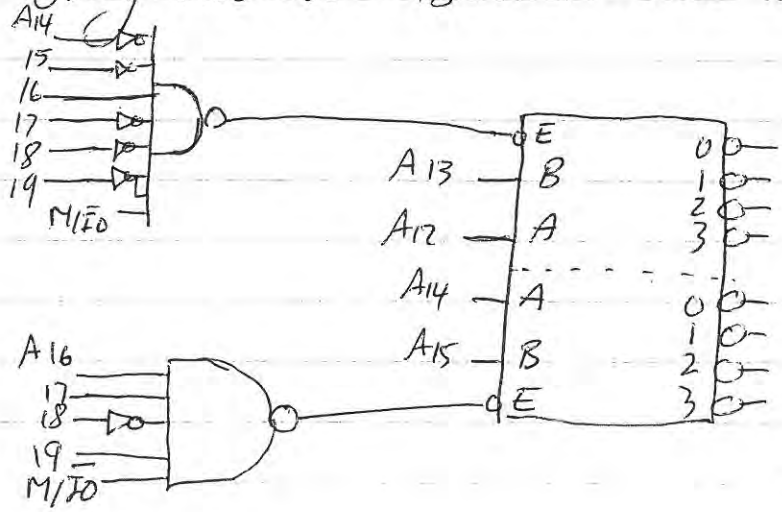
only invert A_{16} and keep the rest as is. $_b$

EPROM 2764

Lowest address: F0000 $\rightarrow (1111)_b\,(0000)_H$

Required to be: B0000 $\rightarrow (1011)_b\,(0000)_H$

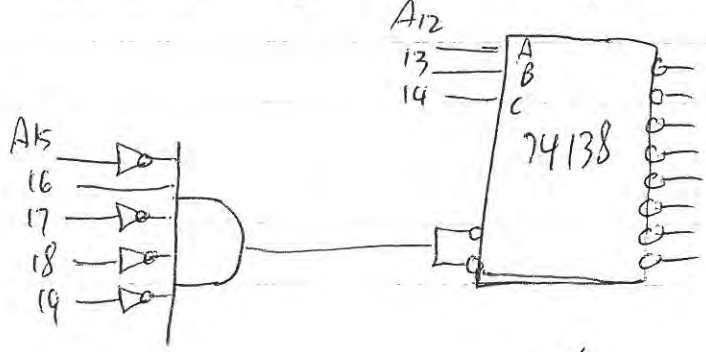
only invert A_{18} and leave the rest as is.



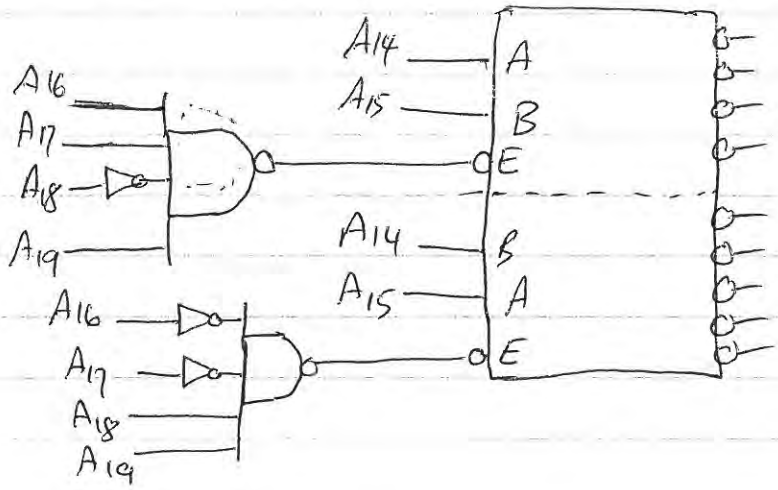
Same Connections

b- Use a 74LS138 decoder for the RAM 4016.

A_0 taken to $\overline{LWR}$, $A_1 \rightarrow A_{11}$ taken to chip memory access, A_{12}, A_{13} & A_{14} for the line decoder A, B, and C, A_{15}, A_{16}, A_{17}, A_{18} & A_{19} are used to select the line decoder in such away starting address is $(10000)_H = (0001)_B\,(0000)_H$



For the EPROM 2764 keep the 74LS139, But remember the total EPROM size is now doubled $2(8 \times 8) = 128KB$
starting address $= (B0000)_H$
Ending address $= (B0000 + 128KB) = CFFFF$
Remark here that you start at B0000 & end at CFFFF, So $(B)_H$ and $(C)_H$ have to be used separately to decode the address as shown below



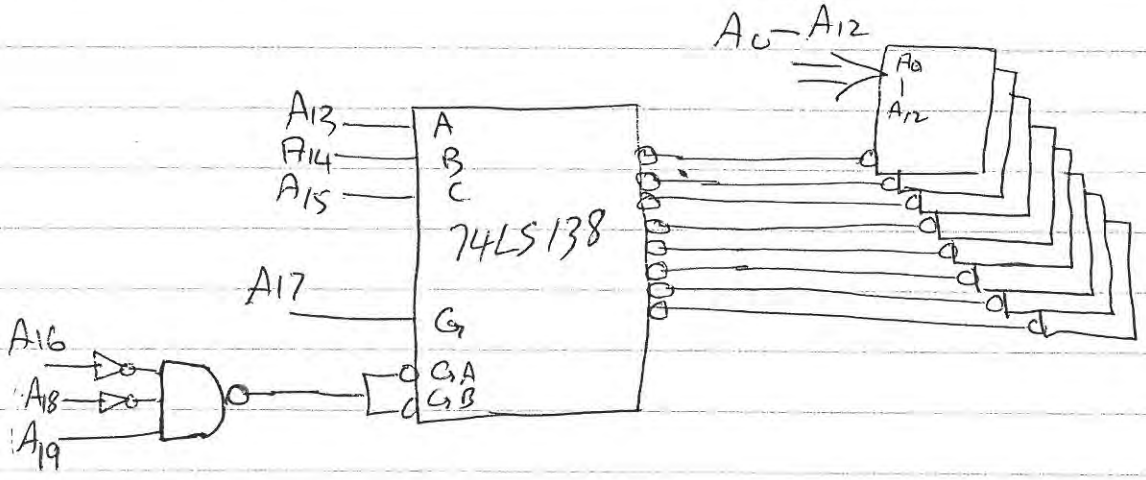
C- Because the total number of chips of RAM is 16 Chips and the Pentium processor has 8 banks, each bank will have 2 chips. Same for the EPROM.

Remember: $A_0 A_1 \& A_2 \rightarrow$ Used to select banks
Address bus of Pentium is 32 ($A_0 \rightarrow A_{31}$)
Bank Selection Signals are $\overline{BE0}$, $\overline{BE1}$, -- $\overline{BE7}$.

9) a- $8KB = 2^3 \, 2^{10} = 2^{13}$

Need $A_0 \rightarrow A_{12}$ to access each memory location in a chip

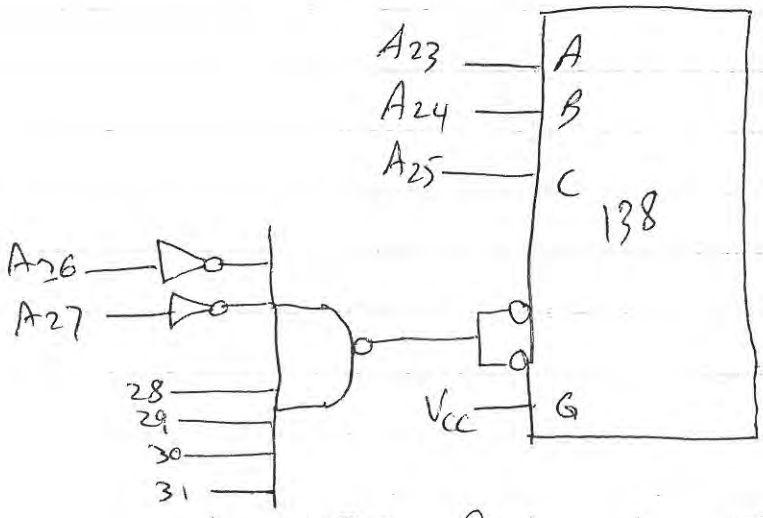
Starting address is $(A0000)_H = (1010)_B (0000)_H$ (19 18 17 16)



b- $1MB = 2^{20}$ needs $A_0 \rightarrow A_{19}$ be used and decode the rest of the 32 address pins.

c- Now use $A_{31} \rightarrow A_{23} \equiv (F00)_H$ to be decoded, i.e. used to be connected to the 138 line decoder.

A_0, A_1 & A_2 are used for bank selection



$A_3 \rightarrow A_{22}$ used to access memory locations.

10- The CRC-Byte is 55ADH

Sheet #2

1. See Class notes

2- First get the addresses

$A_7\ A_6\ A_5\ A_4\ A_3\ A_2\ A_1\ A_0$

1 1 0 1 1 0 0 0

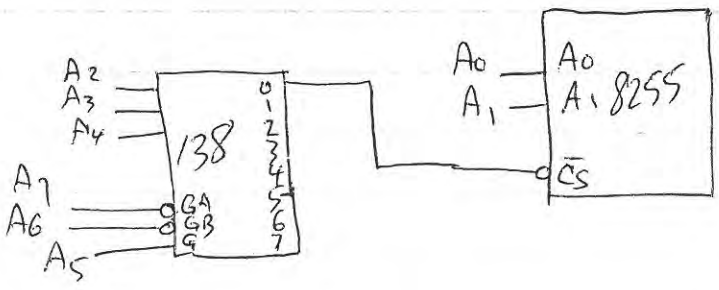
Port A: D8 H
Port B: DA H
Port C: DC H
Command: DE H

Command Byte A 1 0 0 1 0 0 1 0 = 92 H

Program:

```
MOV  AL, 92H
OUT  DEH, AL
IN   AL, D8H
MOV  AH, AL
IN   AL, AH
ADD  AL, AH
OUT  DCH, AL
```

3-a) 20 H = $A_7 A_6 A_5 A_4\ A_3 A_2 A_1 A_0$ = 0 0 1 0 0 0 0 0



Port A: 20
B: 21
C: 22
Command: 23

b- Command Byte A 1 0 0 0 0 0 1 0 = 82 H

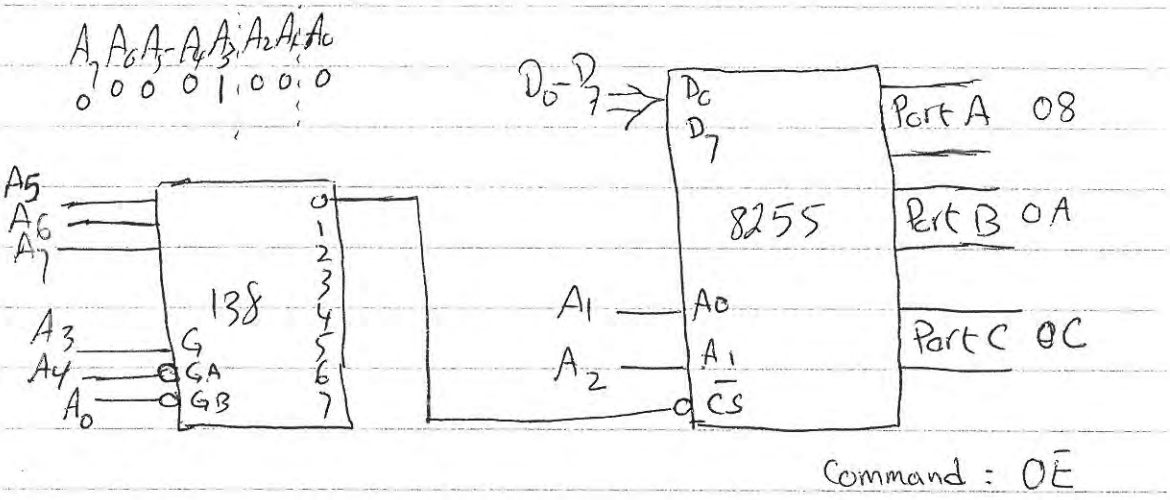
Program

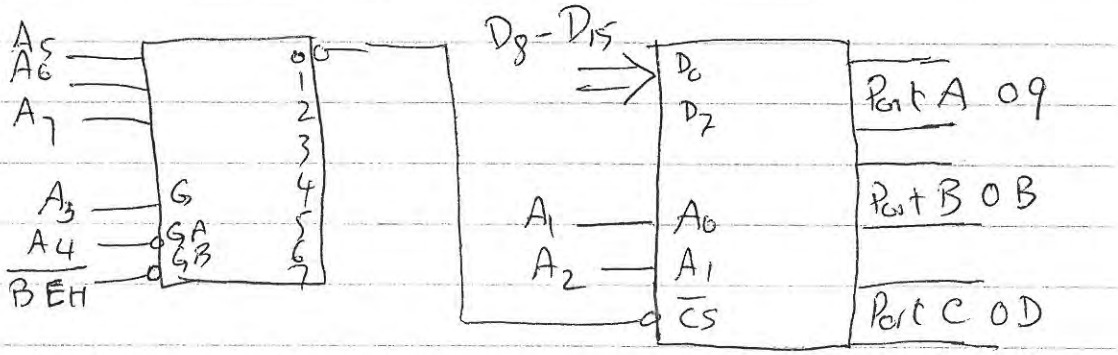
```
MOV  AL, 82H
OUT  23H, AL
```

c- Command Byte A = 10100010 = B2H

```
MOV AL, B2H
OUT 23H, AL
```

d-





A7 A6 A5 A4 A3 A2 A1 A0

0 0 0 0 1 0 0 1

1 0 1 1

1 1 0 1

1 1 1 1

Command 0F