MICROPROCESSORS

→ 80 % Instruction set

10 % Mem, I/O Interfacing

10 % Miscelleneous.

→ Topics:

① Block diag. & pin diag of 8085 µP.

② Mem Interfacing.

③ I/O Interfacing.

④ Instruction, Machine cycles.

⑤ Timing Diagram.

⑥ Instruction set.

⑦ Addressing modes.

∴ **8085 µP**

~~① 16 Address~~ * Features:

① 16 Address Lines

$A_{15} - A_{10}$ → $A_{15} - A_8$ (Higher order Addr. Lines).

$A_{15} - A_{10}$ → $A_7 - A_0$ (Lower order Addr. Lines).

Mem. Capacity $= 2^{16} = 2^6 \cdot 2^{10} = 2^6 \cdot 1\,kB$

$= \boxed{64\ kB.}$

② 8 Data lines:

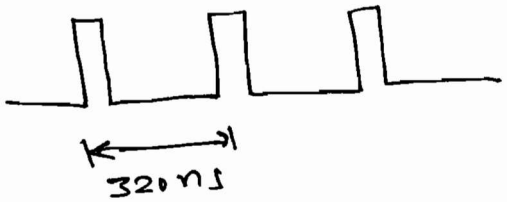
$= \boxed{D_7 - D_0}$

③ Frequency of µP.

$f = 3.072\ MHz$

$\boxed{f = 3.072\ MHz}$; Clock period $T = 1/f$

$\boxed{T = \frac{1}{f} = 320\,ns.}$



④ Von Neumann Architecture.

↓

[Harvard Architecture]

→ In Von Neuemann architecture Programm and data are store into the same memory.

→ In Harvard Architecture Program and data are stored in seperate memories with seperate Buses.

Eg: ARM microcontroller.
Dsp processors.

⑤ 'CISC' Processor.

CISC = Complex instruction set Computer.

↓

X (RISC) = Reduced instruction set computer.

e.g. (i) ARM Controller.
(ii) Dsp processor.

⑥ NMOS Tech.

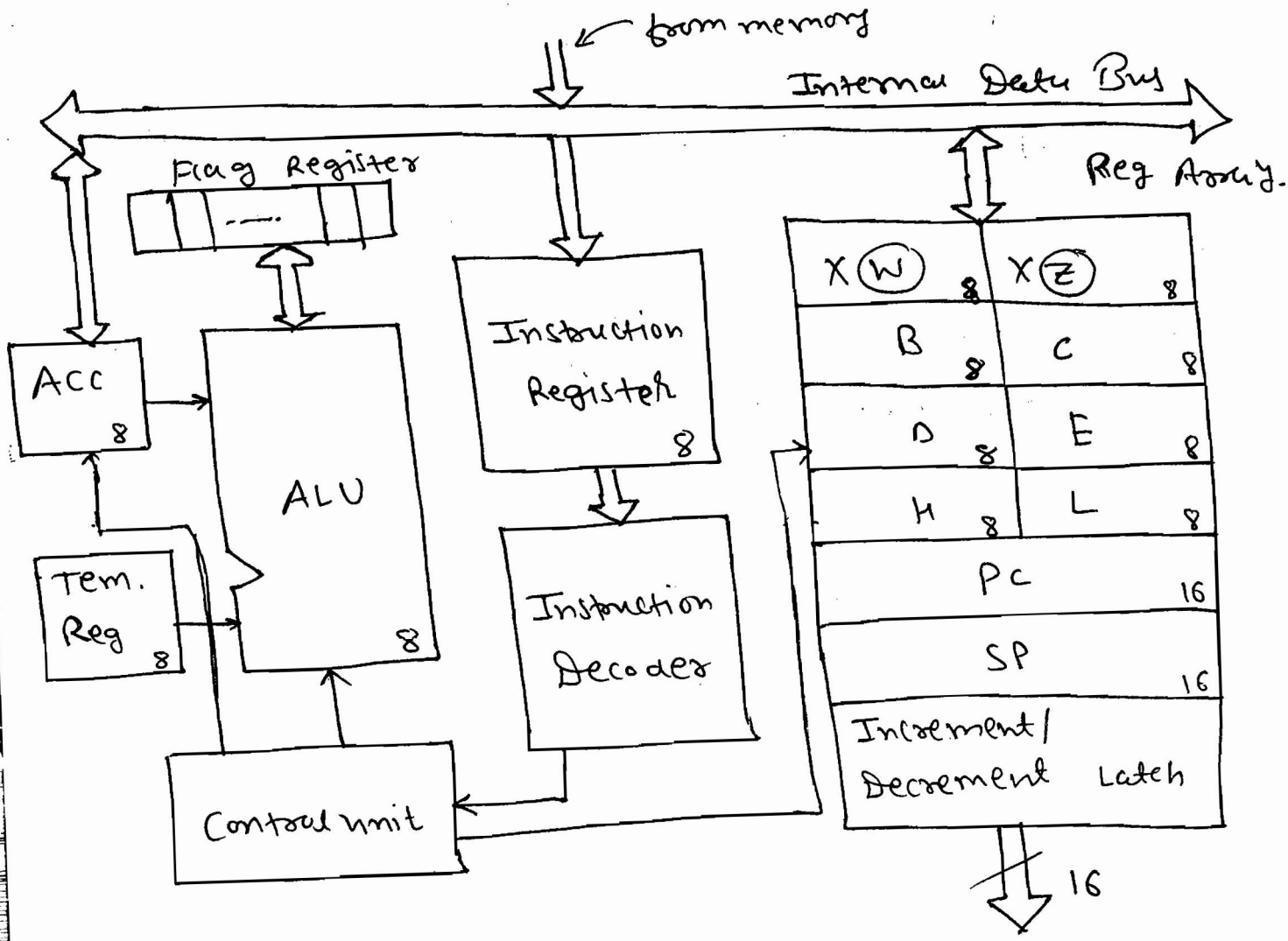
⑦ only +5V supply is used.

⑧ TTL Compatible.

⑨ 8085 up is a 8-bit up because its ALU Capacity is 8-bits.

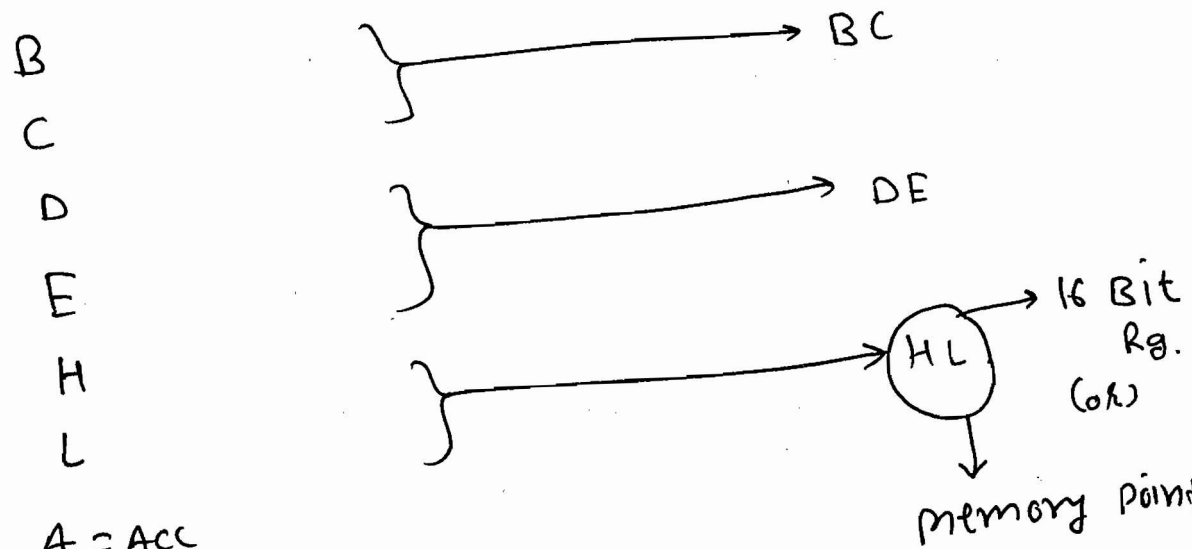
NOTE:
4004, 8008 used +5, -5, +12 supply.

* Block Diagram of 8085 µP.



W, Z = Temp. Reg.

→ 8-bit Register (R) => (7). , Reg. pair | 16 bit Reg. (RP) => (3)



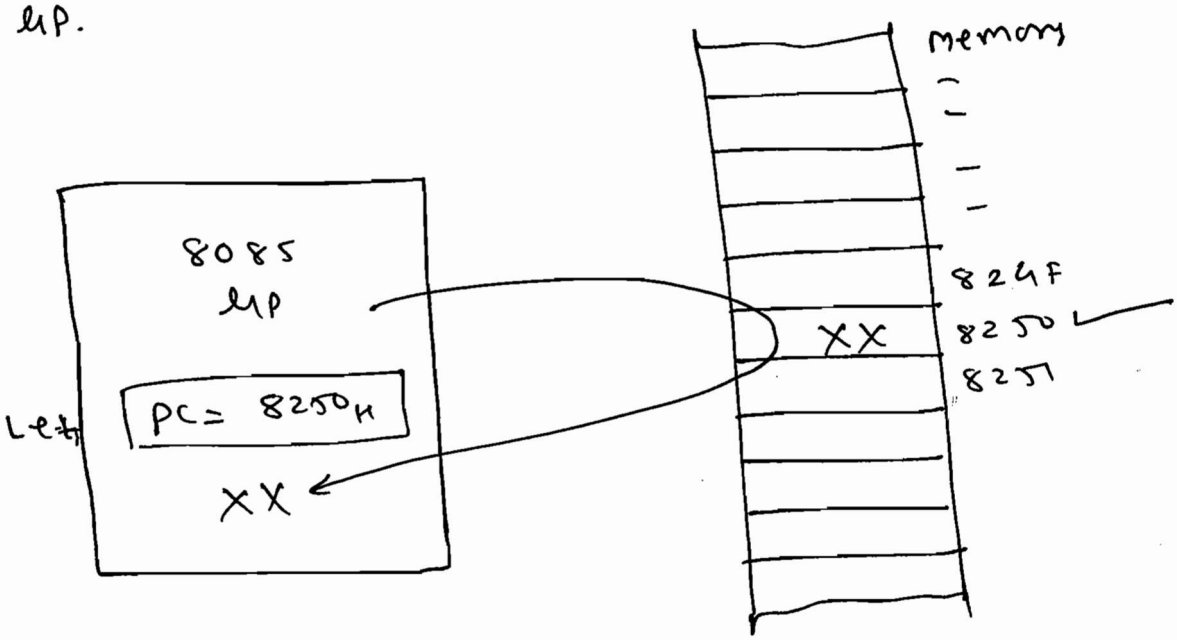
→ PSW_{16} = Program Status Word$_{16}$

= | ACC$_8$ | Flag Register$_8$ |

→ PC: Program Counter.

⇒ Program Counter contains the address of the next instruction.

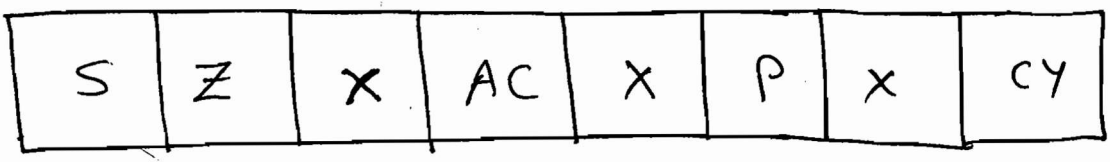
→ It is called IP (Instruction pointer) in 8086 µP.



⇒ SP_{16} = Stack Pointer$_{16}$

* Flag Register:

⇒



S: Sign bflag.

Z: zero flag.

Ap: Parity flag.

CY: Carry flag.

① Sign flag:

→ S=1 if MSB of ALU Result = 1

⇒ S=1 if ALU Result is '-ve'.

② zero flag:

→ Z=1 if ALU operation Resuts = 00_H.

③ Parity flag.

→ P=1 if ALU Resuts has "even parity".

④ CY: Carry Flag:

→ cy=1 if carry occurs during Addition (OR)

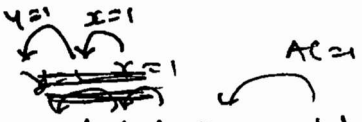
borrow occurs during Subtraction.

⑤ Ac: Aux Carry:

→ it Carry occurs from Lower to Upper Nibble (bit)

NOTE:

→ The Programmer can not excess the Ac flag, µP uses internally for BCD addition.



E.g.

$ED_H = 1110\ 1101_2$

$+\ CB_H = {}^{+}1100\ 1011_2$

$= 1011\ 1000$

$S = 1$

$Z = 0.$

$Ac = 1.$

$P = 1$ (even Parity).

$CY = 1.$

* As $x = 1$, $y = 1.$

⇒ No overflow

(or)

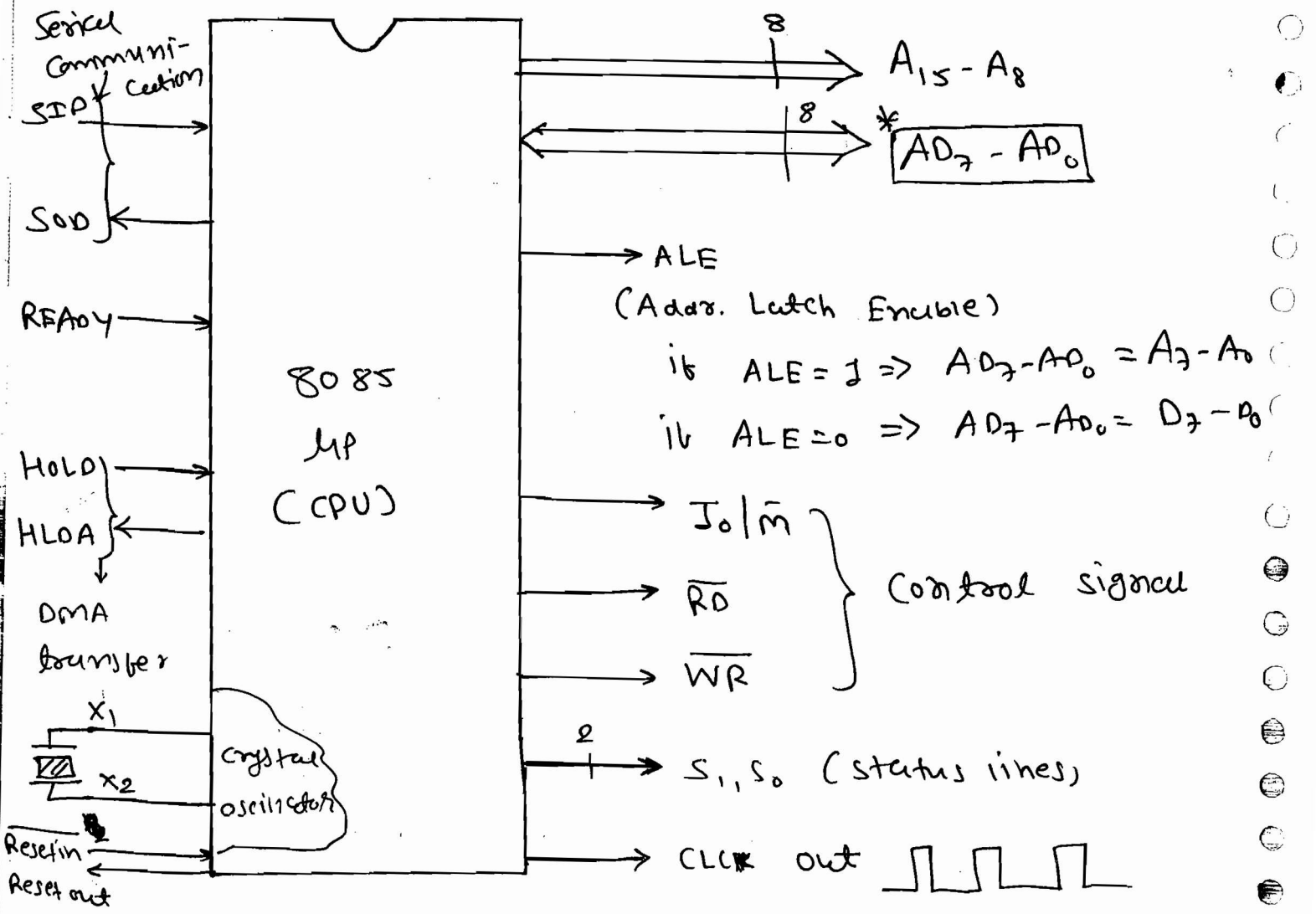
* Two −ve Nos are added.

Result is −ve

⇒ No overflow.

* Pin Diagram of 8085 µP.

⇒



⇒

$IO/\bar{M}$	S_1	S_0	operation
0	1	1	opcode Fetch M/C
0	1	0	mem Read M/C
0	0	1	mem write M/C
1	1	0	IO Read M/C
1	0	1	IO write M/C
1	1	1	Interrupt Ack M/C

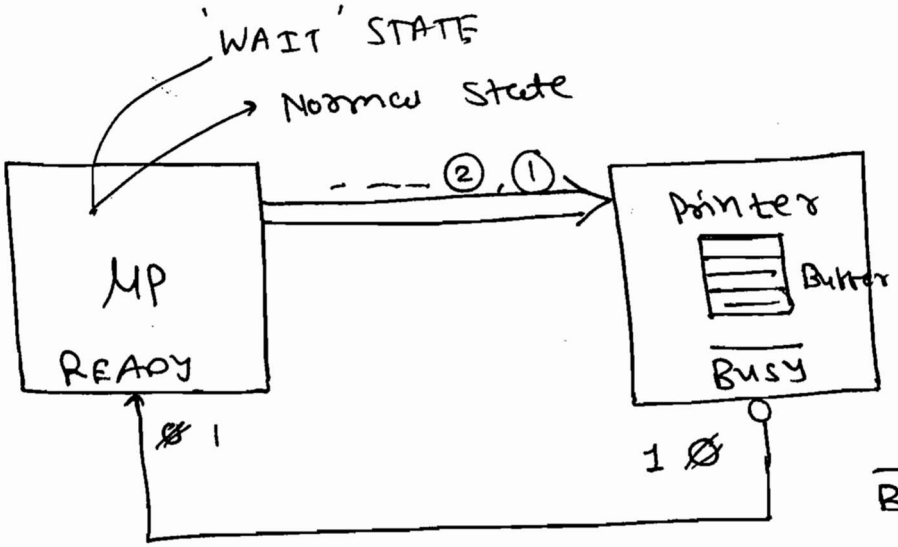
⇒ Multiplexing:

* READY

{ 1 Normal
{ '0' WAIT State

→ Ready pin is used to synchronised the µp with slow speed peripherals and memories.

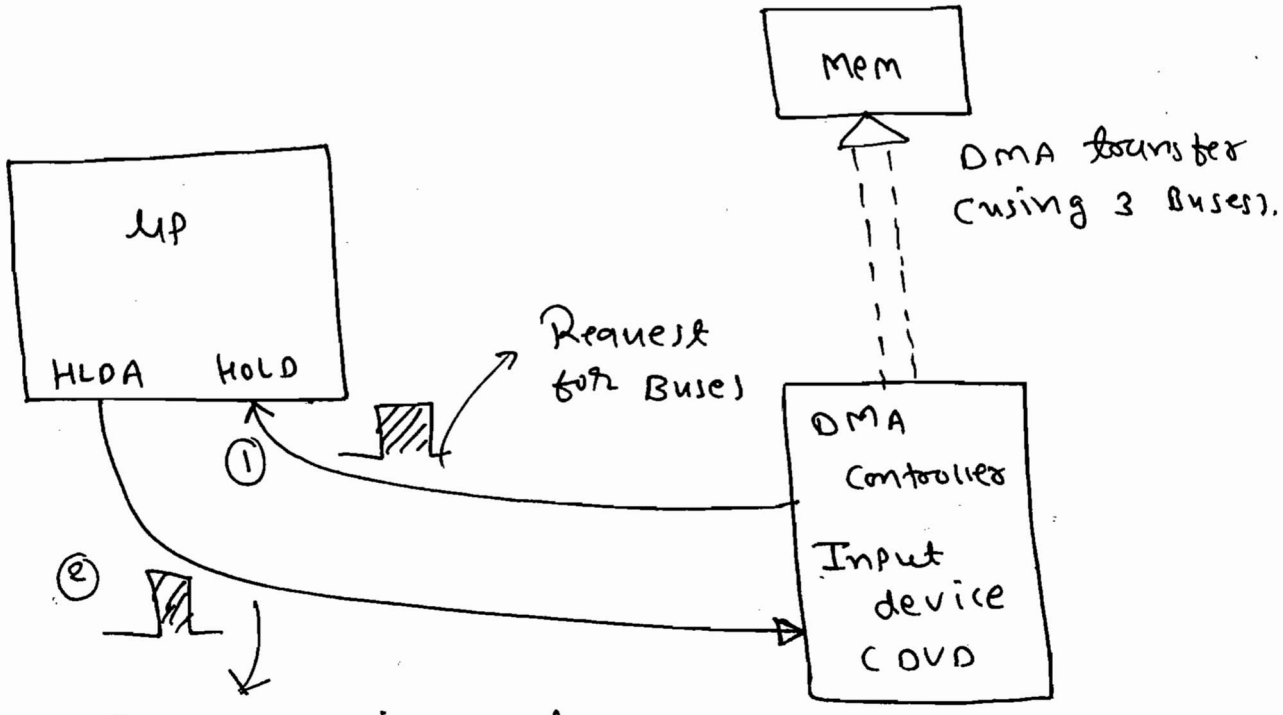
Eg.



$\overline{BSY}$ = 0 printing
= 1 over.

* HOLD and HLDA:

e.g.



→ In DMA transfer µp acquires the Buses from DMA Controller when the HOLD pin becomes LOW

→ Two modes of operation:

ⓐ Burst mode:

→ In this mode the Buses are handover to the microprocessor only after the entire data transfered to memory.

ⓑ Cycle Stealing:

→ In this mode the Control of the Buses switches Back and forth betn µp and DMA Controller.

* X_1 & X_2:

→ 

$f = 3.072$ MHz.

$f_{crystal} = 2 \times f_{\mu p}$

$= 2 \times 3.072$ MHz

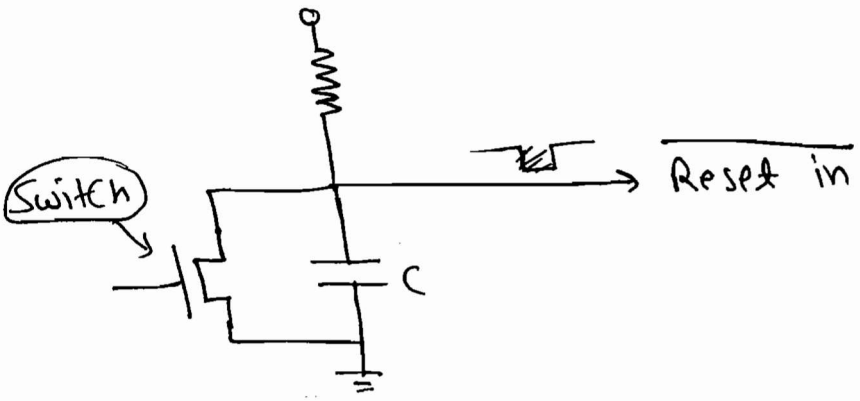
$f_{crystal} = 6.144$ MHz

→ The crystal freq. is divided by two to Convert singie phase clock into a two phase clock because 2 phase MOSFET

MOSFET Shift Register.

* $\overline{\text{Rest in}}$ & $\overline{\text{Re set out}}$.

⇒ Power on Reset



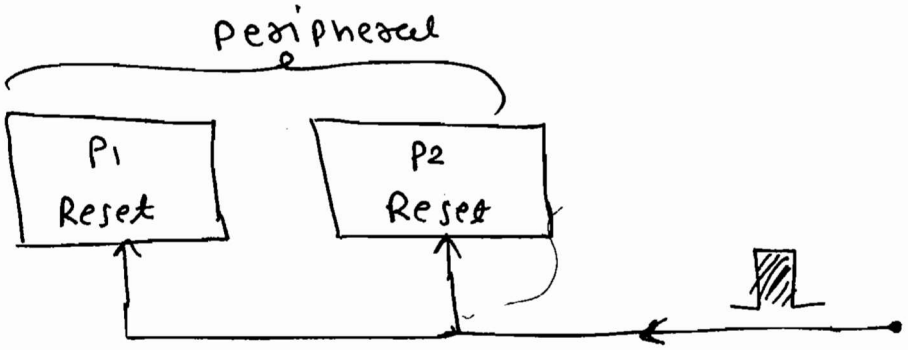
→ when µp is reset

(i) All the Registers are clear including PC.

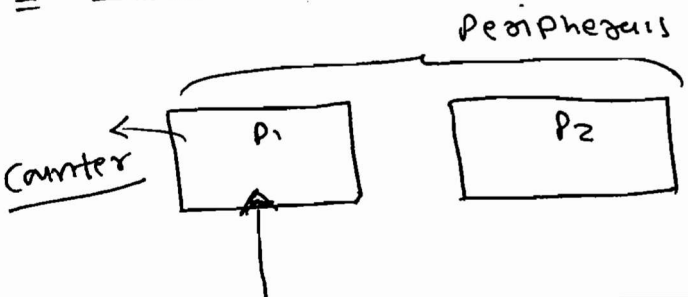
(ii) All the Buses entre into high impedence state.

(iii) µp Processor Fetches its next instruction from memory location 0000H.

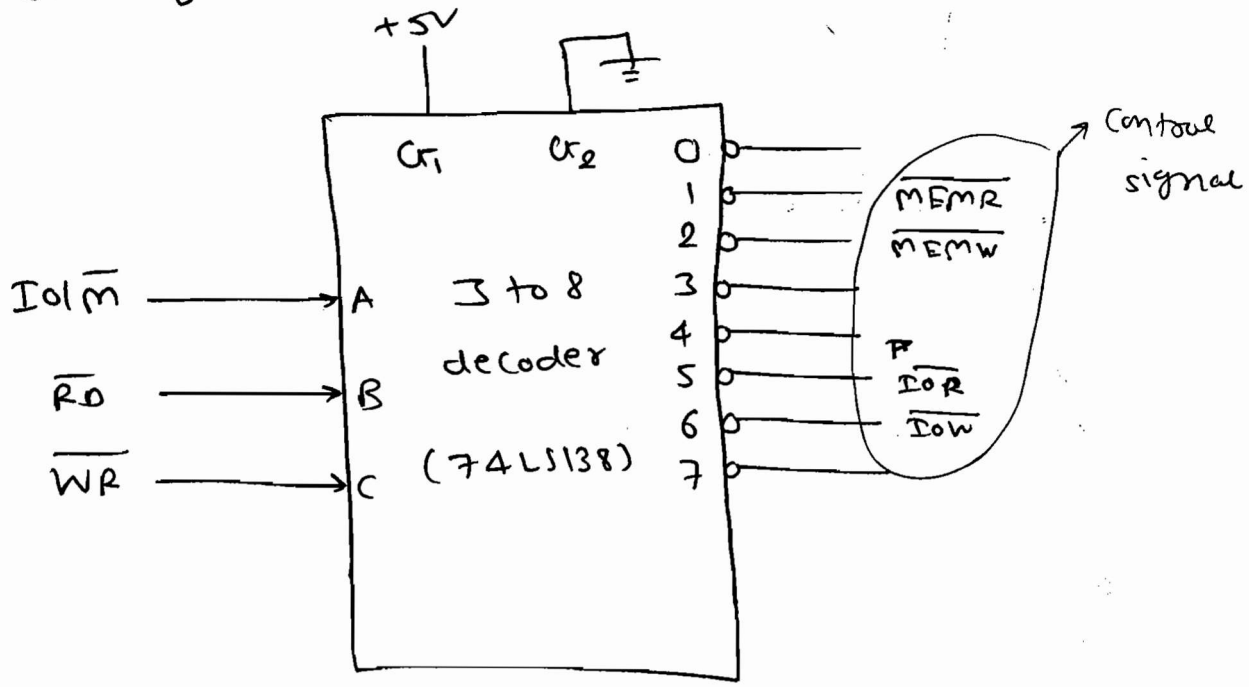
⇒



* CLK out:



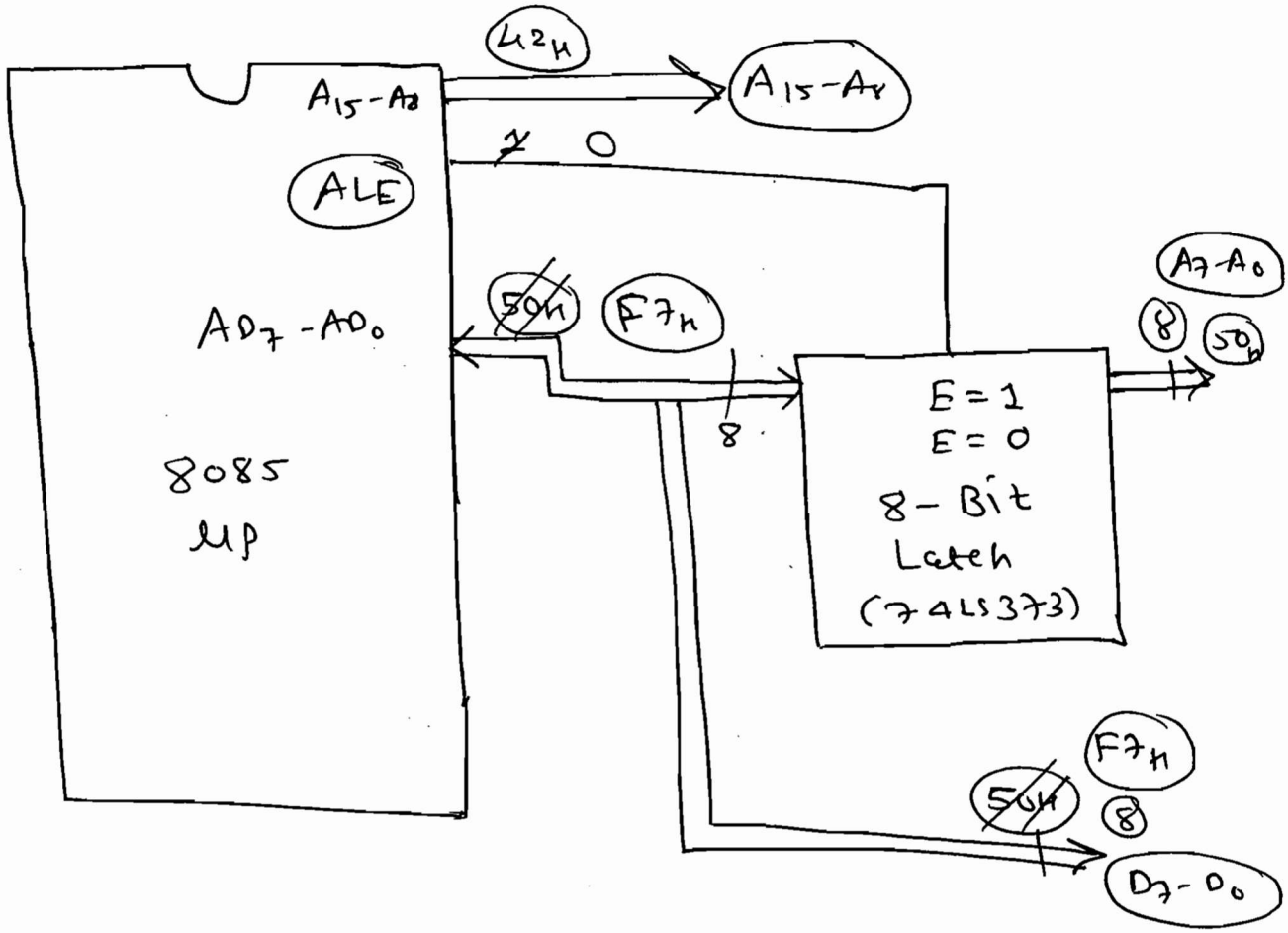
Ex: In the following decoder, determine the number of valid outputs and their functions



Ans:

$IO/\overline{M}$	$\overline{RD}$	$\overline{WR}$	outputs
0	0	0	0 → Invalid
0	0	1	1 → $\overline{MEMR}$
0	1	0	2 → $\overline{MEMW}$
0	1	1	3 → Invalid
1	0	0	4 → Invalid
1	0	1	5 → $\overline{IOR}$
1	1	0	6 → $\overline{IOW}$
1	1	1	7 → Invalid.

* Demultiplexing of $AD_7 - AD_0$.



μP

① Address = 4250H.

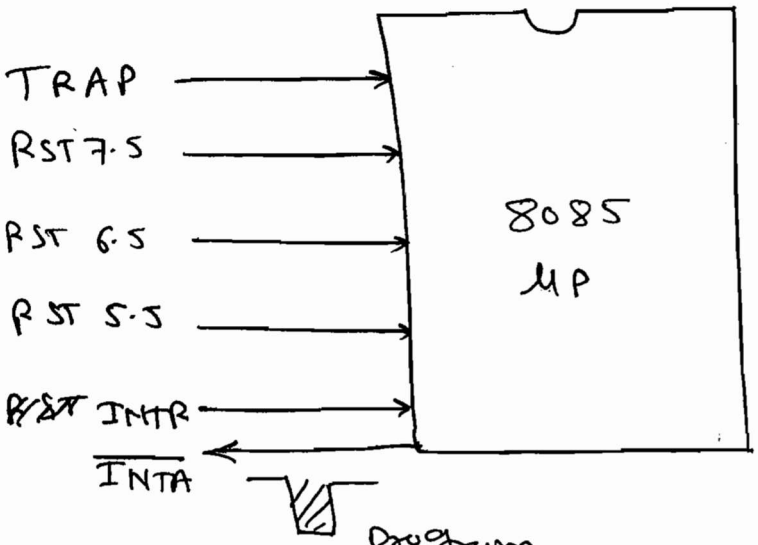
② Data = F7H.

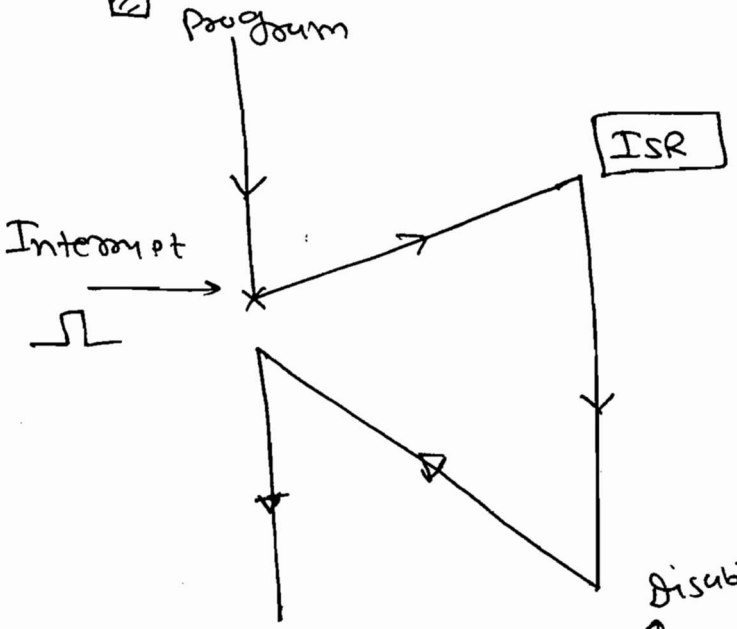
* Latch is enable, when Address is sent on $\boxed{AD_7 - AD_0}$

* Latch is disable, when data is sent on $\boxed{AD_7 - AD_0}$

→ The disadvantges of multiplexing is it reduces the μP speed of operation because of the time taken in Demultiplexing.

* Hardware Interrupts.

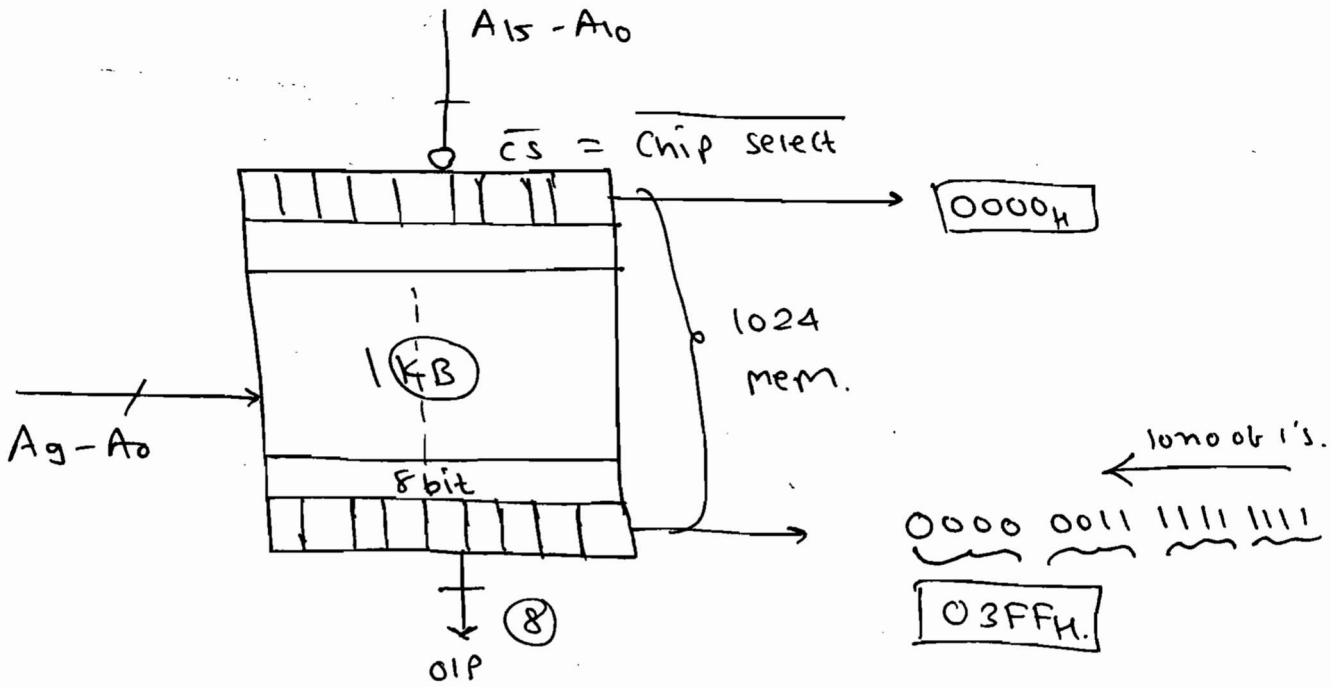




	Priority	Vector Address	Masking	Types of Triggering
TRAP	Highest	0024_H	NMI	Positive edge + Level
(RST 4.5)				
RST 7.5		$003C_H$	Maskable	Positive edge
RST 6.5		0034_H		
RST 5.5		$002C_H$		Level
INTR	Least	NVI		

Disabling

* Memory IC's:



1 KB memory = 1024 x 8

I/P: 2^{10} x 8 O/P

→ if S.A = 3800 then E.A. = 3800 + 03FF = 3BFF$_H$

→ if E.A. = FFFF then S.A. = FFFF$_h$ − 03FF$_H$ = FC00$_H$

Ex-1 Determine the starting address of 2764 memory IC if its ending address is AA5F$_H$.

Ans: NOTE: MEM IC

EPROM: 27 16/8 = 2KB ← RAM: 61 16/8

27 32/8 = 4KB ← 61 32/8

27 64/8 = 8KB ← 61 64/8

27128 = 16KB ← 61 128/8

→ Memory size 8KB

$= 2^3 \times 2^{10} \times 8$

$= 2^{13} \times 8.$

13 → Address lines.

$\therefore$ $AA5F_H$.

E.A. = 0001 1111 1111 1111

1 F F F

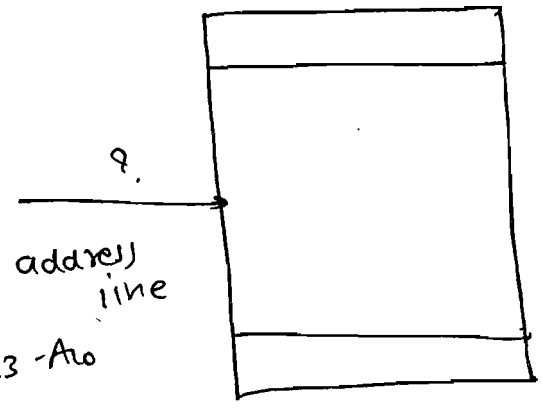
$\therefore$ S.A. = $AA5F_H$

$- 1FFF_H$

$8A60_H$

Ex: Determine the size of the memory whose starting and ending address are $3A00_H$ & $79FF_H$ respectively.

Ans:



$3A00_H - 3A00_H = (0000)_H$

$79FF_H - 3A00_H = (3FFF)_H$

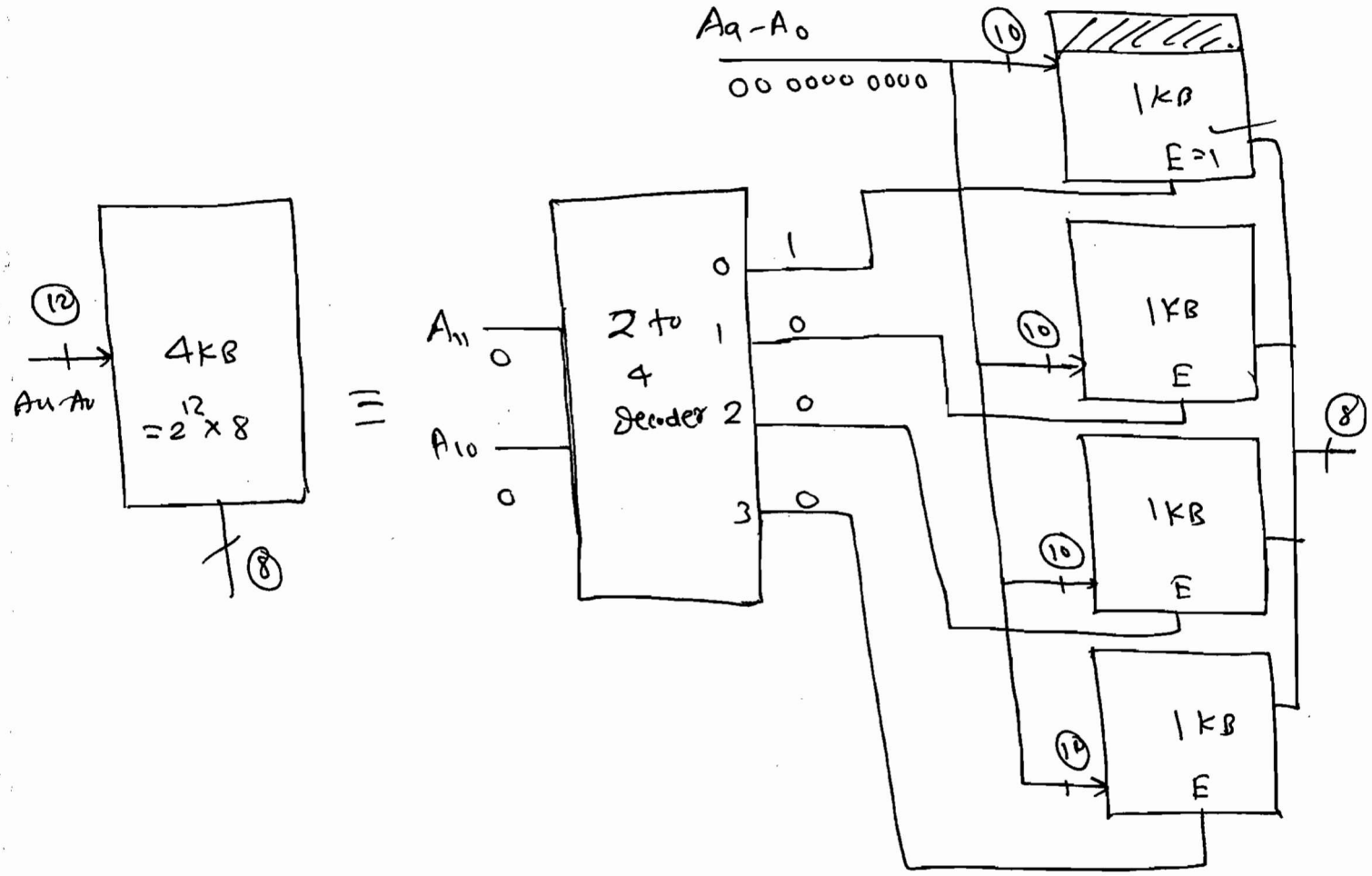
$= 0011\ 1111\ 1111\ 1111$

$= ?$

Hence, Memory size

* Memory Expansion:

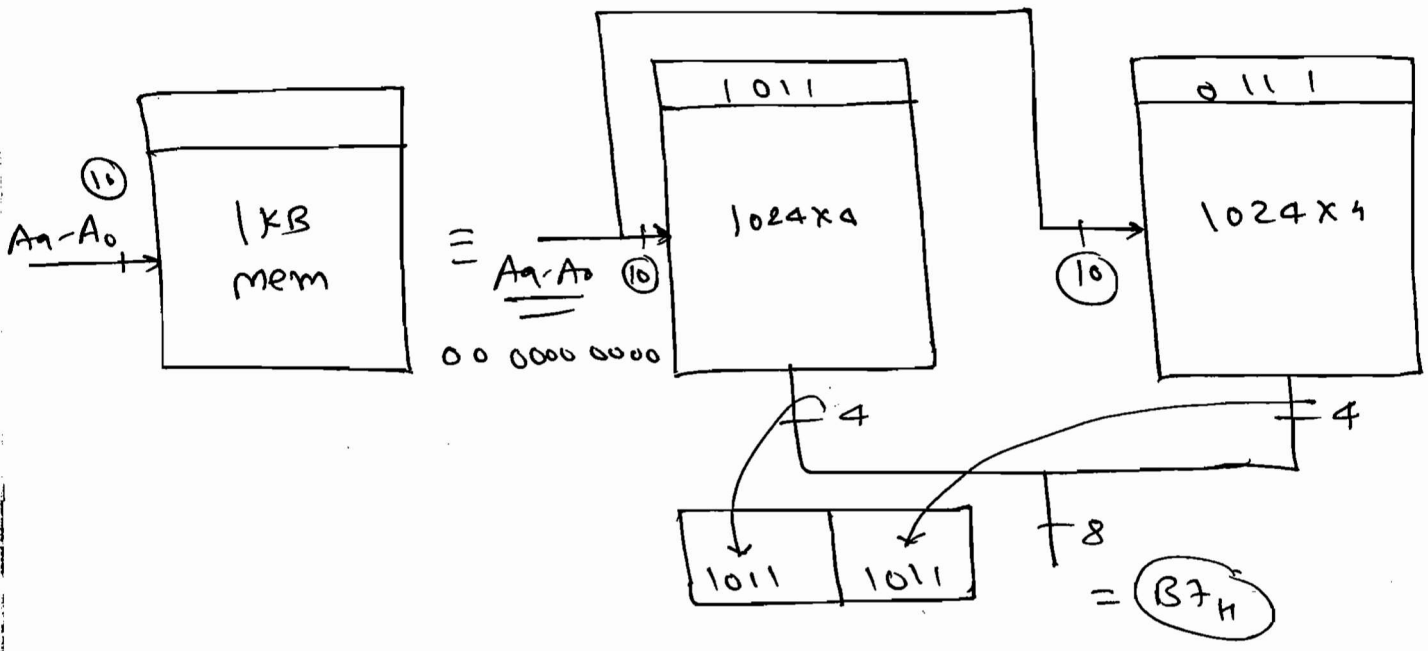
** Construct a 4KB memory by using 1 KB Memory ICs



* Construct a 1KB memory using 1024x 4 mem IC's. → 1KB Mem. = 1024 x 8.

p) Construct a 1KB memory using 1024x4 IC's.

→ 1KB mem = 1024 x 8.



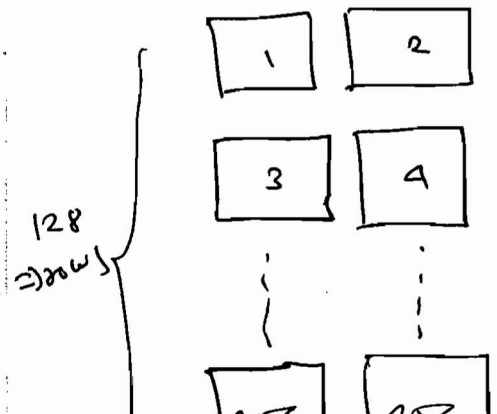
(P)

– How many 256 x 4 memory IC's are require. to construct a 32KB memory.

Ans: No. of mem ICs = $\frac{\text{Required size of mem}}{\text{Given size of mem.}}$

$= \frac{32 \times 1024 \times 8}{256 \times 4.}$

$= 256.$



Ex-2 A digital computer has 20 address lines and 24 data lines. How many memory ICs having 16 address lines and 8 data lines are required to feed the memory of the computer?

Ans:

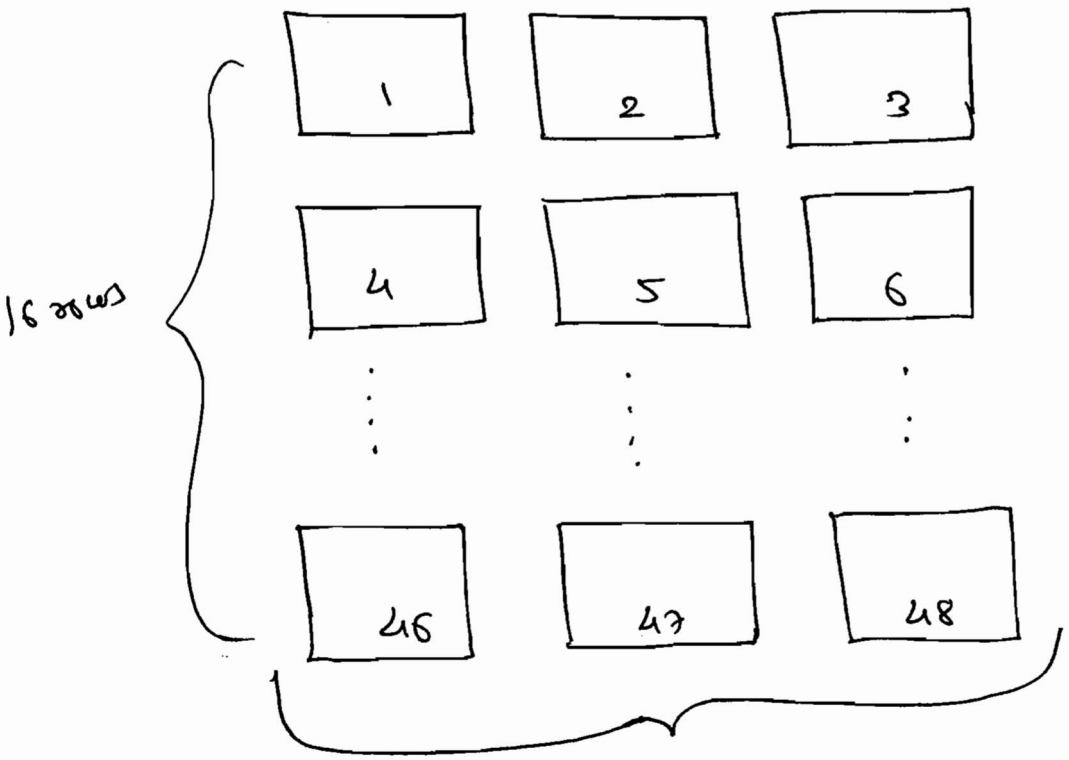
$$\text{No. of mem ICs} = \frac{\text{Required size of mem}}{\text{Given size of mem.}}$$

$$= \frac{2^{20} \times 24}{2^{16} \times 8}$$

$$= 16 \times 3$$

$$= 48.$$

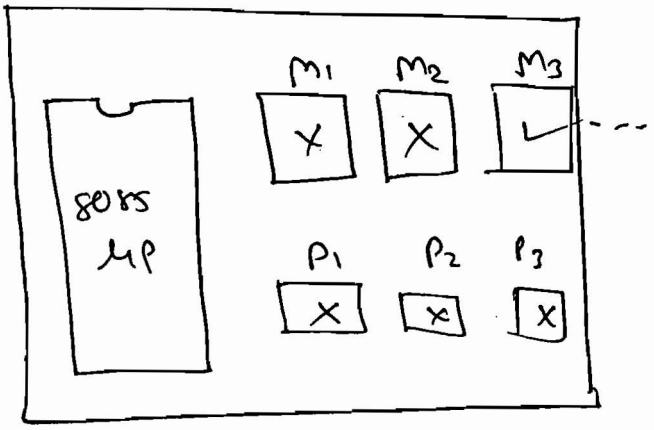
→ 16 rows, 3 columns.

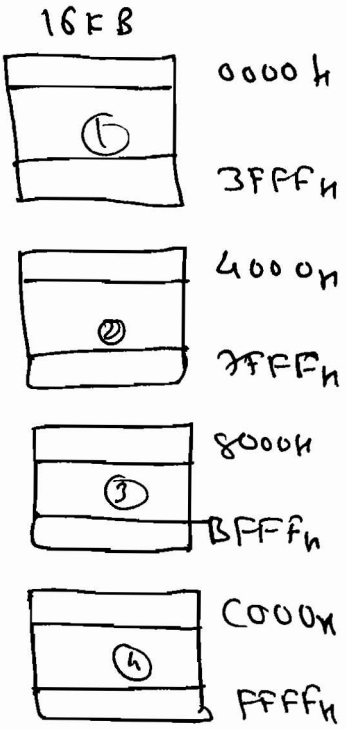


"$2^{20} \times 24$ = memory."

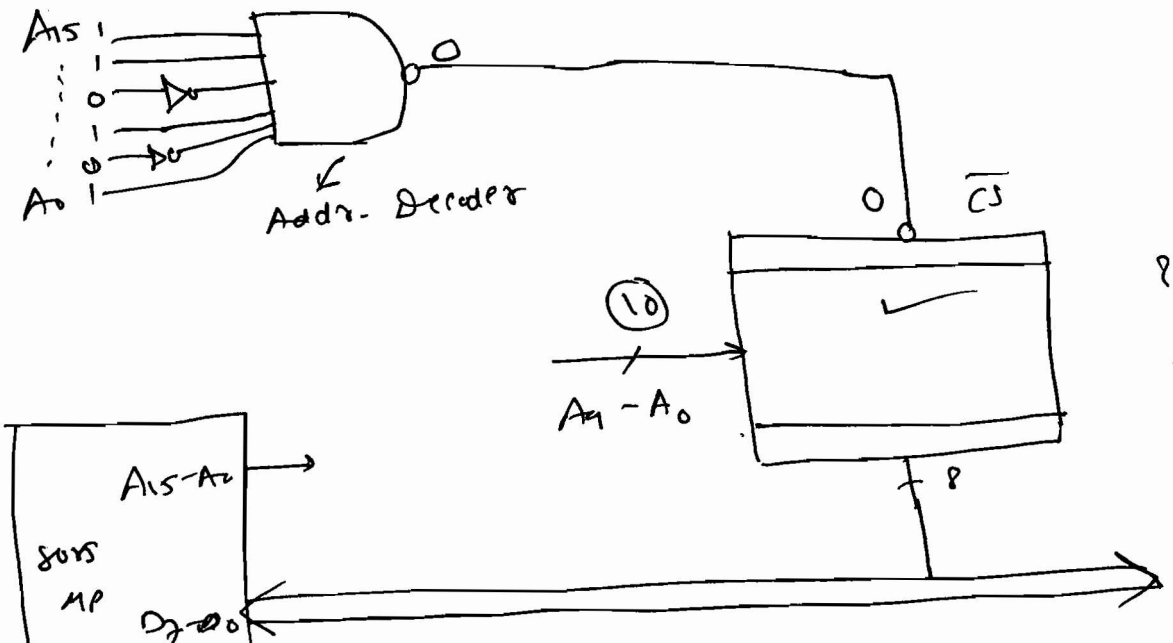
* Memory Mapping:

→ It is the process of assigning address Range to all the memory ICs in a microcomputer.





Ex:1 In a micro computer a 1KB memory IC is interface ~~into~~ the µP as shown in the following figure: Determine the address range of the memory IC



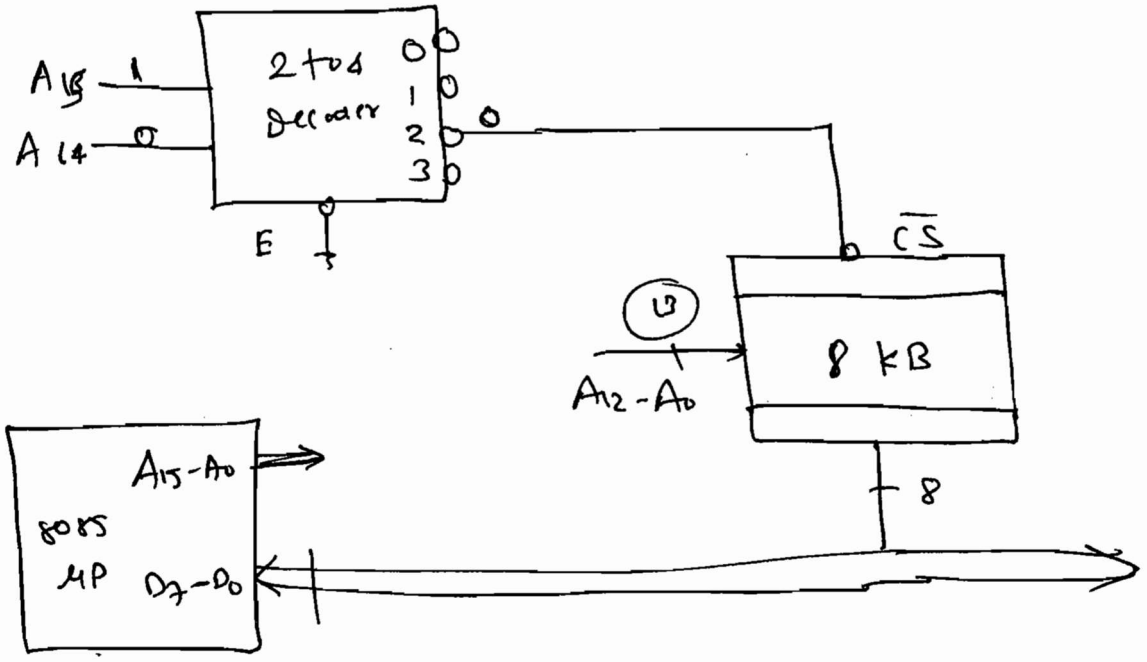
A_{15} ------ A_{10}	A_9 ------------ A_0	
1 1 0 1 01	00 0000 0000	$= D400_H$
1 1 01 01	11 1111 1111	$= D7FF_H$

Addr. Range: $D400_H$ - $D7FF_H$.

→ In the above interfacing as all 16 addr. lines are utilized it is called absolute decoding which results in one to one mapping.

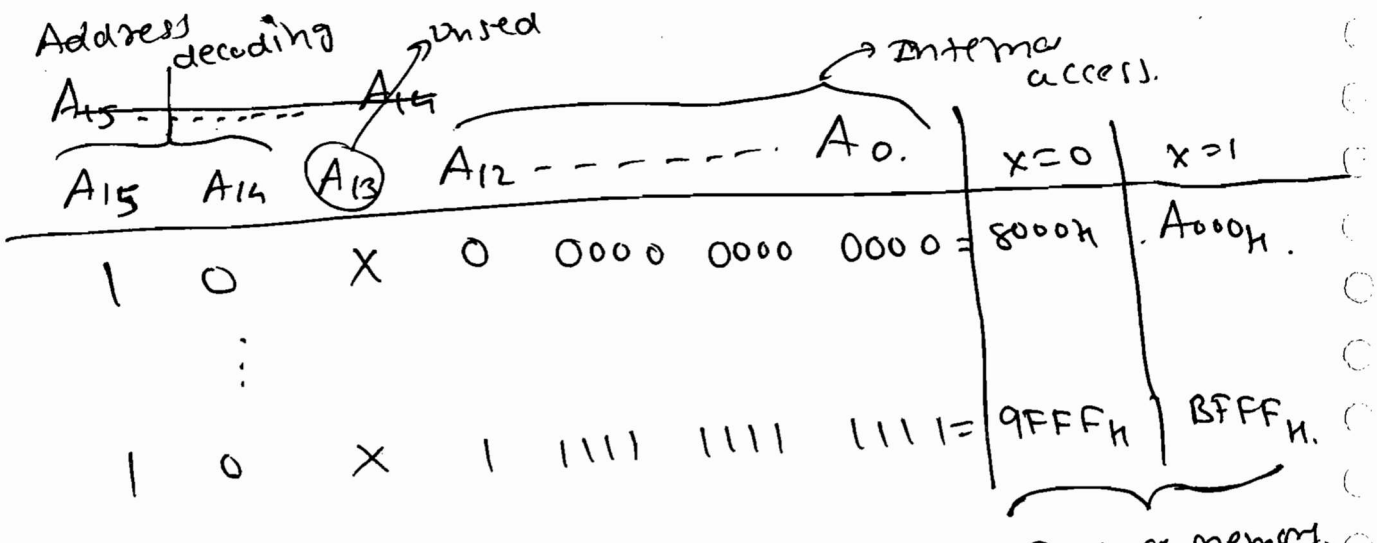
Ex-2 A 8kB memory IC is interface to the microprocessor as shown in the figure. Determine the address range of the memory IC.

Ans:



* A_{13} address line is unused.

~~Ans~~ A_{15} ----------

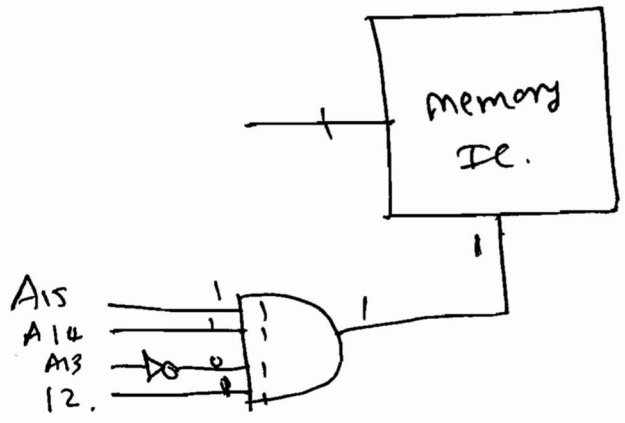


* "A_{-13}" addr line is unused

⇒ Partial Decoding ⇒ many to one mapping.

NOTE: In partial decoding if n address line are unused into results in 2^n foldback memories.

Ex:1 Which of the following indicate the address of the memory IC shown below:



→ mem is selected only when

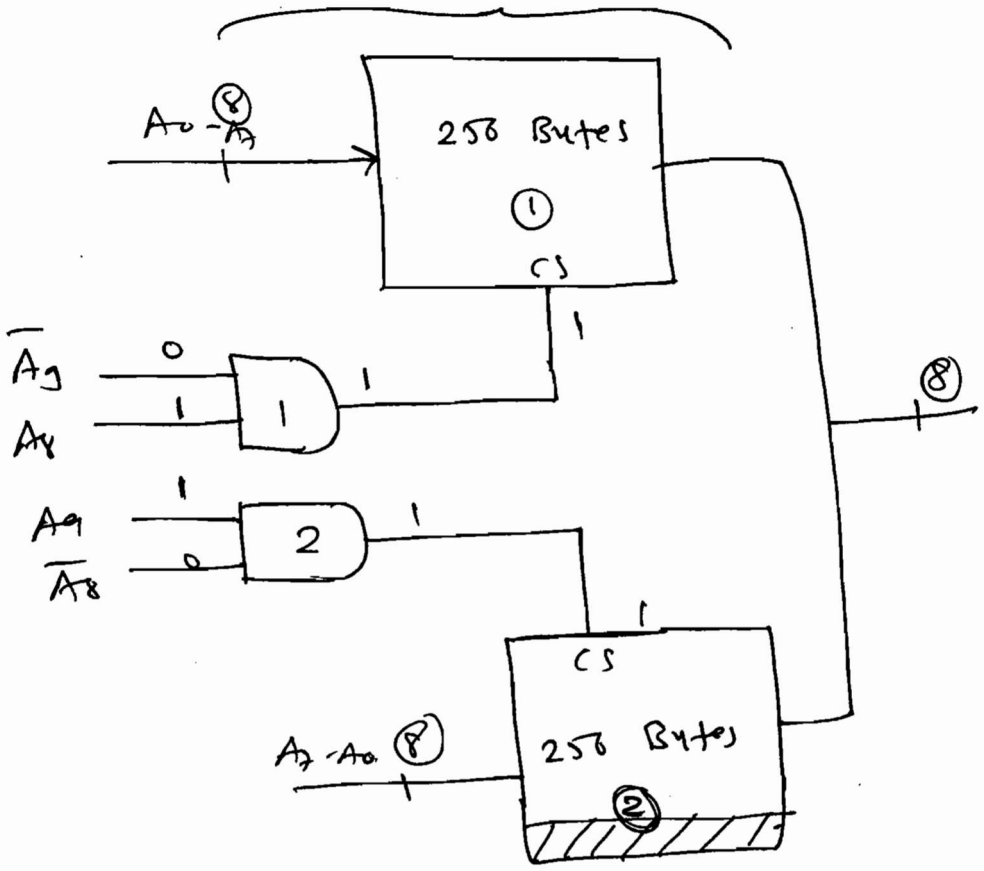
$A_{15}A_{14}A_{13}A_{12} = 1101 = DH.$

(a) $B200_H$ – B3FFH

(b) $F800H$ – F9FFH

(c) ... – DBFFH

Ex-2 Which of the following doesn't indicate the address of the memory IC shown below:



	$A_{15} - A_{10}$	A_9	A_8	$A_7 \ldots A_0$
S.A. =		0	1	0000 0000.
E.A. =		1	0	0000 0000

→ Starting Add. must have $A_9 = 0$, $A_8 = 1$.

→ Ending Add. must have $A_9 = 1$, $A_8 = 0$.

		A_{15}	$A_{14}A_{13}$	$A_{12}A_{11}$	A_{10}	
(a)	$3100_H - 32FF_H \Rightarrow$	3100_H =	0011	0001	0000	0000 Valid
		$32FF_H$ =	0011	0010	0000	0000
(b)	$8D00_H - 8EFF_H \Rightarrow$	8000_H =		1101		valid
		$8EFF_H$ =		1110		
(c)	$F400_H - F5FF_H \Rightarrow$	$F400$ =		0100		Invalid
		$F5FF_H$ =		0101		
(d)	$C900_H - CAFF_H$					
	→	$C900$ =		1001		