

a) For an n bit address line, we can access 2^n memory locations.

So if we want to access 16K location in memory then,

$$2^n = 16K$$

$$\text{or, } 2^n = 16 \times 2^{10}$$

$$\text{or, } 2^n = 2^4 \times 2^{10}$$

$$\text{or, } 2^n = 2^{14}$$

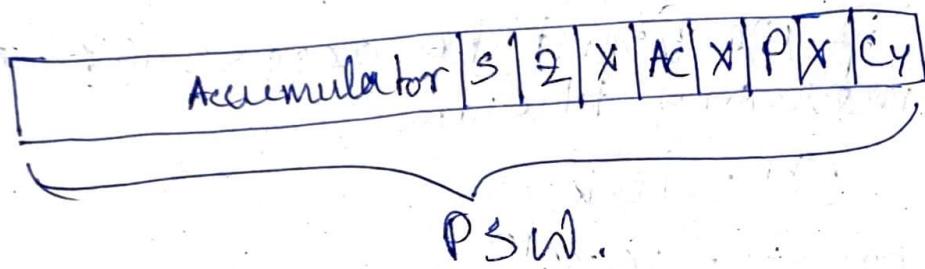
$$\therefore n = 14$$

So, we need 14 address lines to access a memory of 16Kbyte.

b) In 8085 microprocessor HOLD pin is used for ~~handling~~ interrupt handling and facilitates the Direct Memory Access (DMA) mechanism for high speed data transfer between memory & I/O devices. When the external device want to send data to memory directly the DMA controller sends HOLD request to microprocessor and then after completing the current machine cycle microprocessor relinquishes the use of ~~a bus~~ the bus and sends HLDA high signal to indicate that the HOLD request is received and the device can send data.

c) PSW stands for Program Status Word. It is a ~~combi.~~ 16-bit word, a combination of 8-bit accumulator & 8 bit flag register. It stores the result of last arithmetic or logical

operation. The flag register consists of 5 type of flags: S (sign flag), Z (zero flag), AC (Auxiliary carry flag), P (parity flag), CY (carry flag).



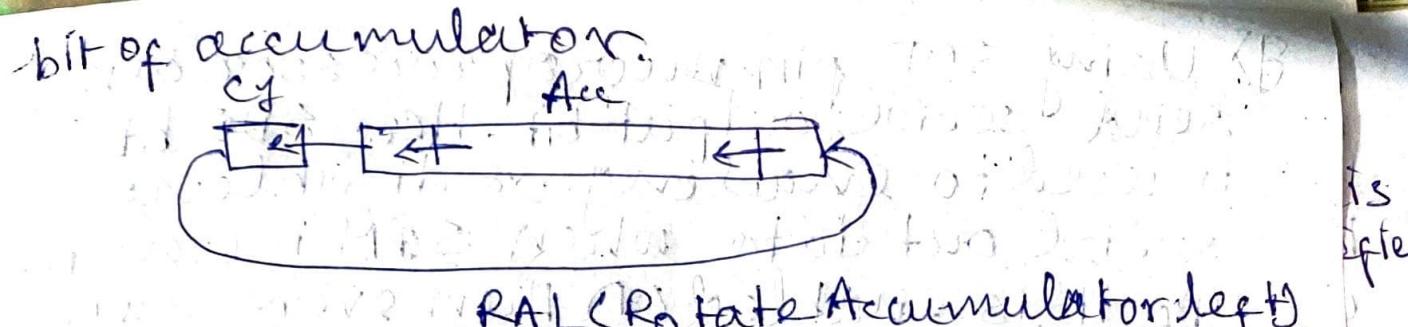
- d) When PCHL instruction is used it is load with the content of HL pair. PC (Program Counter) is a 16 bit register which keeps the address of the next instruction to be executed. After using PCHL, PC loads the content of HL pair and jumps to the location. It is used for unconditional branching or jump.

The reverse of $\text{JMP}(\text{xxxx})_H$ is kind of same. It causes an unconditional jump to the memory location denoted by $(\text{xxxx})_H$. It changes the PC to specific address.

In PCHL the address is load from HL register pair.

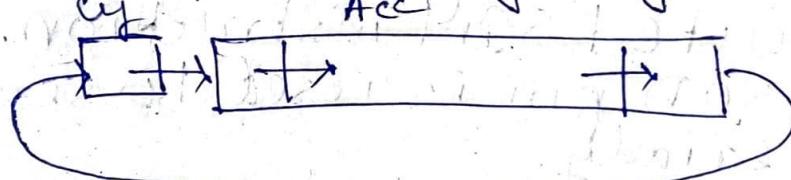
In $\text{JMP}(\text{xxxx})_H$ the address is directly mentioned.

- e) ~~RAT~~ In case of RAL instruction the left most bit of the accumulator is moved to the carry bit/flag. The content of the remaining bits also move to left by 1 bit & the content of the carry is moved to the right most



RAL(Rotate Left Accumulator left).

In case of RAR the right most bit(LSB) of the accumulator is moved to the carry flag/bit & the content of carry is moved to the left most(MSB) position of the accumulator. & remaining bits also move right by 1 bit.



RAR(Rotate Accumulator Right).

- f) The microprocessor 8085 does not have any separate internal memory for storing program codes and data operands. Instead it relies on external memory devices for this purpose. The program code is typically stored in ROM, while data and operands are stored in RAM. The internal registers of the 8085 are used for temporary storage and intermediate calculations during program execution but do not serve as dedicated memory for program code or data storage. Thus, the 8085 requires external memory for its program and data storage need.

g) Using SOD pin microprocessor can send serial output bit. Here SDE bit is used to enabling or disabling serial out data when SPM is executed. If it is enabled then every SPM execution generates 1 data bit which will be 0 or 1 depends on Most significant SOD bit on accumulator, it will be available on SOD line. If we like to send 1 8-bit data serially to other computer, then for every bit we have to execute 1 SPM instruction. In this way SOD pin is used to send data out serially.

SOD	SDE	x x	x R75	MSE	M45	M65	M55
-----	-----	-----	-------	-----	-----	-----	-----

Serial data out
serial data enable

h) Few special purpose registers in 8085 microprocessors:

① Accumulator: It is a 8 bit register, part of ALU. After performing any arithmetical or logical operation result is stored in it.

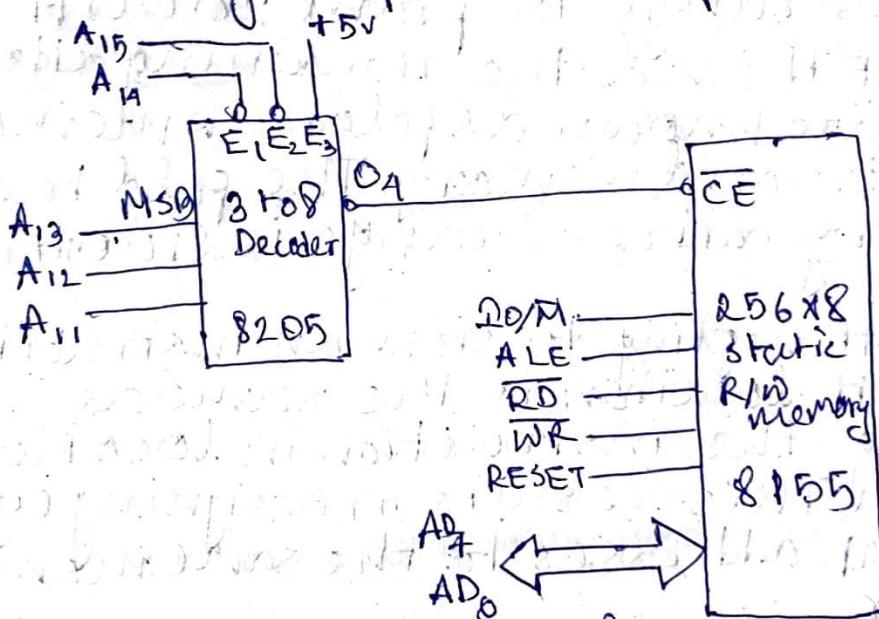
② Flag Register: It is a 8 bit register used to store the status of the last arithmetic or logical operation.

③ Program Counter: It is used to point to the memory address where the next instruction to be executed resides.

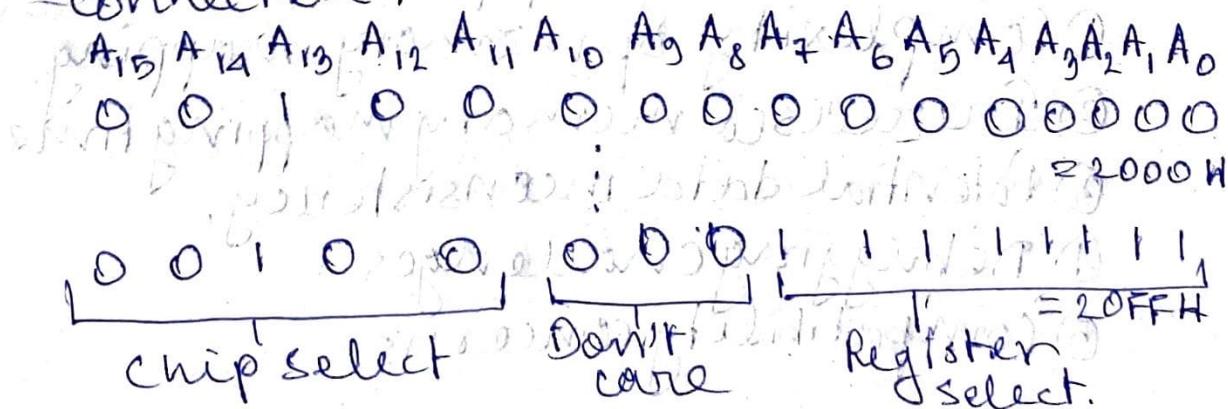
④ Stack pointer: It is used as a memory pointer. It points to a memory location in read/write memory, called the stack. It is incremented/decremented by 2 during PUSH and POP operation.

Q) In microprocessor we know there is 16-bit address line busses address of a specified memory in the Memory chip.

To explain fold back memory with respect to microprocessor 8085 we are here interfacing 8155 chip. Separate address lines are used to select the memory chip and the particular block.



Here A_{15} & A_{14} is used to enable the decoder chip. A_{13}, A_{12}, A_{11} is used to select the output of the decoder. Here A_{13}, A_{12}, A_{11} is 1, 0, 0 respectively to represent output 4. AD_7 to AD_0 is used to address the block from 256 memory location. Here A_8, A_9, A_{10} is in don't care state as these lines are not connected.



Here the address range is from 2000H to 20FFH. Here the three lines A₈, A₉, A₁₀ are in don't care state, so each bit can be taken as 0 or 1 making combination from 000 to 111. Thus each combination can generate one set of complete addresses. The address range given by assuming all the don't care lines as 0 is the primary address. So the primary address range is from 2000H to 20FFH and the remaining address ranges are known as foldback memory or mirrored memory. The fold back memory address range is 2100H to 27FFH.

Attempting to store an instruction in 2100H location is the same as entering the instruction in location 2000H. This results in assigning eight different addresses to the same memory register.

Advantages of fold back memory:

- ① Memory space utilisation
- ② Helps in peripheral integration
- ③ Cost effective design.
- ④ Redundancy and Fault Tolerance.

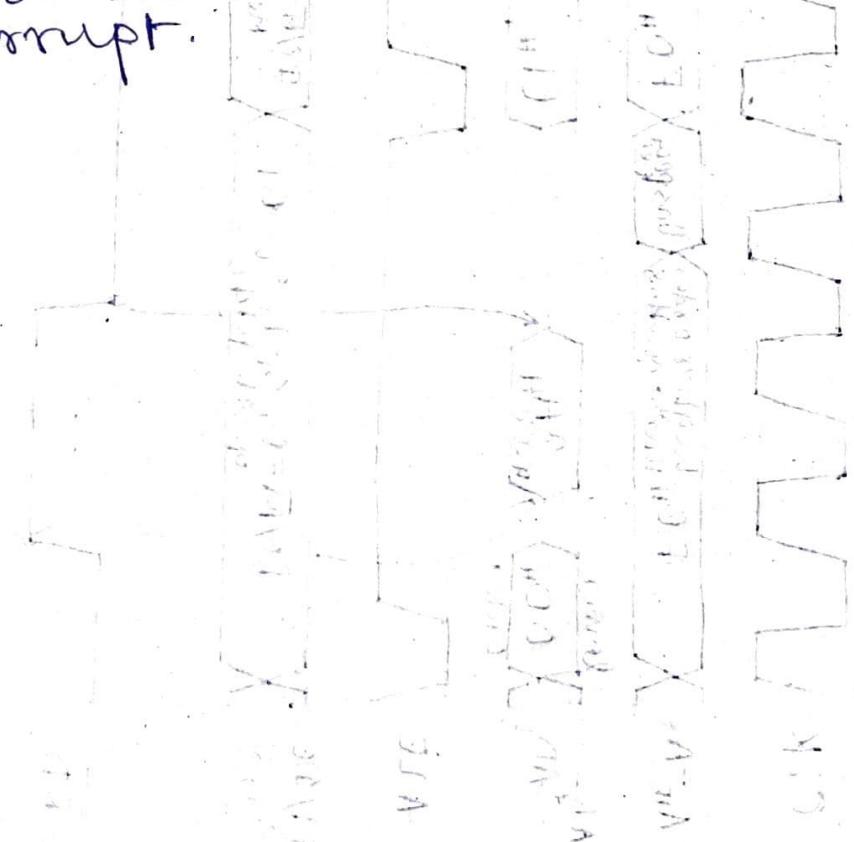
Disadvantages of fold back memory:

- ① Complexity of memory mapping.
- ② Increased memory mapping time.
- ③ Potential data inconsistency.
- ④ Debugging challenges
- ⑤ Compatibility concerns.

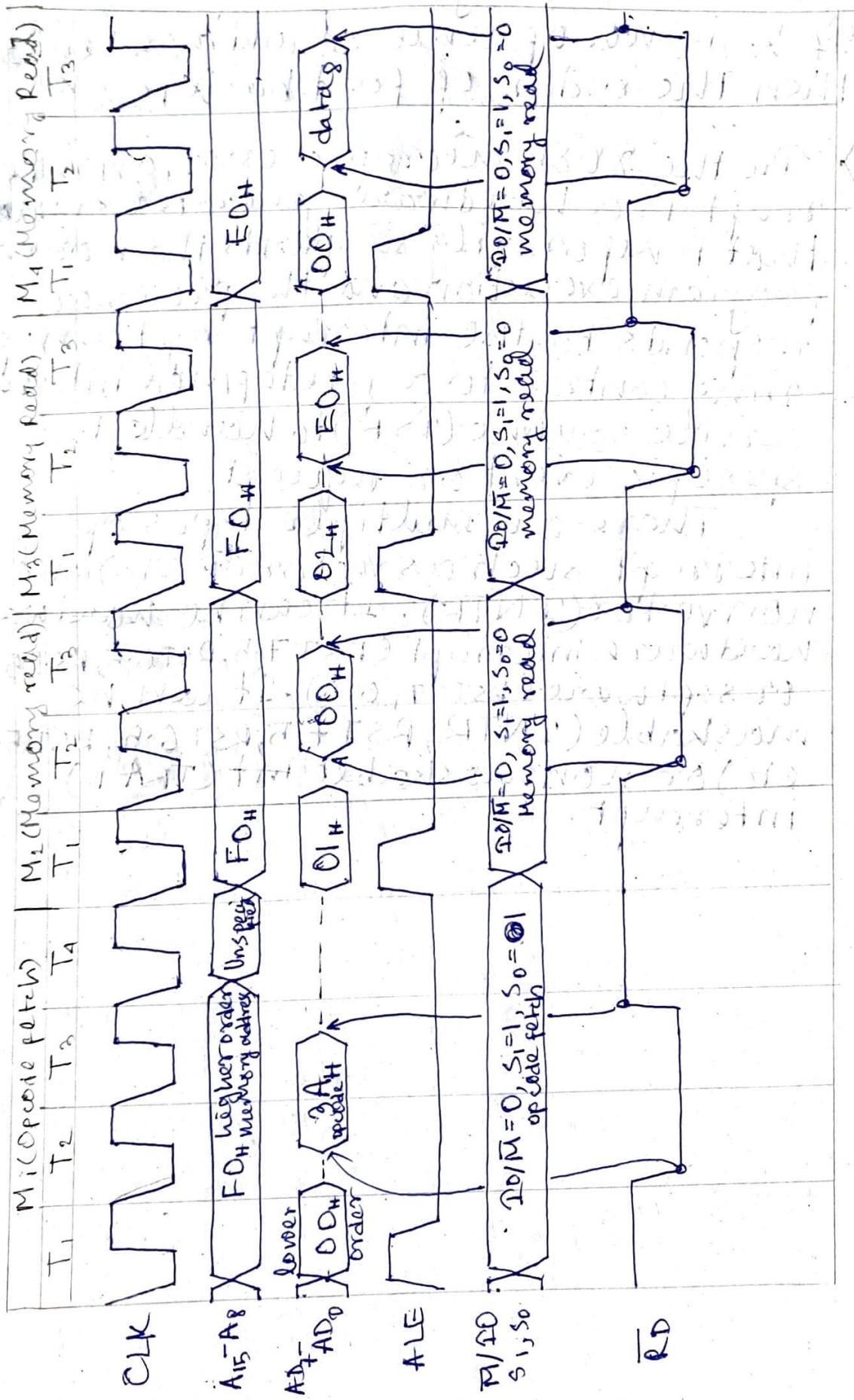
Partial Decoding is used to decode foldback memory address space. Here 3 out of 8 address lines ($A_8 - A_{15}$) are not decoded resulting in multiple addresses. If k is no. of unused address lines then the order of fold back is 2^k .

- b) In the 8085 microprocessor, an interrupt is a hardware-generated signal that temporarily suspends the normal program execution and the processor responds to the interrupt by transferring control to a predefined interrupt service routine (ISR) to handle the specific event or request.

There are multiple types of interrupt such as vector (RSTs) and non vector (NTR). It can be hardware interrupt (RST 7.5, RST 6.5, RST 5.5) or software (RST 7, 6, 5). It can be maskable (NTR, RST 7.5, RST 6.5, RST 5.5 etc.) or nonmaskable INT (TRAP) interrupt.



3) LDAD E000H \Rightarrow BAH - F000H
 00 - F001H
 EO - F002H



(2)

b) The 'PUSH PSW' instruction is used in the Intel 8085 microprocessor to push the contents of Program Status Word (PSW) register onto the stack. PSW consists the content of accumulator & flag register.

Operation of the 'PUSH PSW' instruction:

- ① At first the stack pointer is decremented by 1 and the content of the accumulator is loaded into the memory location pointed by the stack pointer.

$$(\text{SP}) = (\text{SP}) - 1$$

$$A \rightarrow ((\text{SP}))$$

- ② Then again the stack pointer is decremented by 1 and the content of the flag register is load into the memory location pointed by the sp.

$$(\text{SP}) = (\text{SP}) - 1$$

$$F \rightarrow ((\text{SP}))$$

$$\text{The new SP is } (\text{SP}) \leftarrow (\text{SP} - 2)$$

This instruction is a 1 byte instruction.
It is a register indirect instruction.

It takes 3 machine cycles i.e. Opcode fetch & 2 memory reads and the total T states needed is $6 + 3 + 3 = 12$ T-states.

- 4) b) 8085 microprocessor instruction can be classified into various groups:

- ① Data Transfer Group:

These instructions are used to transfer data between registers, memory and I/O ports.

Eg:

MV B, C: Move the content of register C to B.

MVI A, 0X45: Move the immediate value 0X45 to the accumulator.

② Arithmetic Group:

These perform various arithmetic operations like addition, subtraction, increment, decrement etc.

Eg:

ADD B: Add the content of register B to the accumulator.

SUB M: Subtract the content of the memory location pointed by HL from the accumulator.

③ Logical Group:

performs logical operations like AND, OR, XOR etc.

Eg:

ORA A: Perform a logical OR operation b/w the accumulator A itself.

④ Branching Group:

Used for program flow control and branching.

Eg:

JMP 2050H: Unconditional jump to the memory address 2050H.

JZ 3020H: Jump to the memory address 3020H if the zero flag is set.

⑤ Stack group:

These operate on the stack and perform operations like push, pop etc.

Eg:

PUSH B: Push the content of BC register pair onto the stack.

POPD: POP the top of the stack into the DE register pair.

⑥ I/O Groups:

Perform I/O operations.

Eg:

OUT 20H: Output the content of the Accumulator to the I/O port 20H.

IN 80H: Input data from the I/O port 80H into the Accumulator.

⑦ Control Group:

Used for control operation like enabling / disabling interrupts, halting the CPU etc.

Eg.

ED Enable interrupts.

HLT: Halt the CPU until an interrupt occurs.

⑧ Special Group:

These have special functions and perform tasks like exchanging register pairs, adjust decimal values etc.

Eg:

EXCHG: Exchange the content of the DE and HL register pairs.

DAA: Decimal Adjust Accumulator, used after BCD addition/subtraction.

⑨ Interrupt Groups:

These are interrupt instructions

Eg: RST-0, RST1 etc.

a) DAD

The 16 bit content of the specified register pair is added to the contents of the HL register and the sum is saved in the HL register. The content of source register pair is not altered.

This is a 1 byte instruction, 3 machine cycle & 10 states.
If the result is larger than 16 bits the Cy flag is set. No other flag is affected.

Eg: Adding the content of HL register pair to itself

Before DAD H After DAD H

02	42
H	L

$$\begin{array}{r} 0242 \\ 0484 \\ \hline +0242 \\ \hline 0484 \end{array}$$

DAA

The contents of the Accumulator are changed from binary value to two 4 bit binary-coded-decimal value. This is the only instruction that uses the Auxiliary flag to perform the binary to BCD conversion.

If the value of the lower nibble of the accumulator is greater than 9 or the AC flag is set then 6(06) is added to the lower nibble.

If the value of the higher nibble of the accumulator is greater than 9 or the CY flag is set then 6(60) is added to the higher nibble.

It is 1 byte instruction.

It takes 1 machine cycle and 4 T states.

$$\begin{array}{r} \text{Eg: } A = 39_{BCD} = 001111001 \\ + 12_{BCD} \quad \quad \quad \underline{\quad \quad \quad} \\ \hline 51_{BCD} \quad \quad \quad 01001011 \end{array}$$

4 B

The value of lower nibble greater than 9, so 06 is added.

$$\begin{array}{r} 40 = 01001011 \\ + 06 = 00000110 \\ \hline 51 = 01010001 \end{array}$$

④ ①

POP rp

The contents of the memory location pointed by the stack pointer register are copied to the lower order register of the operand. The stack pointer is incremented by 1 & the content of that memory location are copied to the high order register of the operand. The sp register is again incremented by 1.

It is a 1 byte instruction with 3 machine cycles and 10 T-states. ③

Eg.

Before instruction

H	X X	X X	2090	F5
SP	2090	2091	01	
		2092		

After instruction

H	01	F5	L
SP	2092		

5>

a) Demultiplexing of Address/Data bus of microprocessor 8085

8085 microprocessor has 16 bit address bus ($A_{15}-A_0$) are directly connected as output signal through the separate 16 pins. Using this 16 bit address, we can accommodate almost $2^{16} = 64$ kilo main memory locations.

This processor allows 8 bit external data lines called as Data Bus (D_7-D_0). However, for this data bus separate pins are not used. Least significant 8 bits or lower byte of address bus is used for this purpose. In this way same 8 lines of D_7-D_0 is used for both address bus and data bus but not at the same time. Generally it is called as multiplexed Address and Data Bus.

Using this address bus and data bus the main memory system can be organised as $2^{16} \times 8$ or 64KB.

For I/O we also need address but this processor is not providing separate address line, as well as data line for I/O. Same address and data lines are used for both memory & I/O purpose. To know whether current address is for memory or I/O, the CPU is introduced with 1 output

line named as IO/M . If $\text{IO/M} = 0$, then address and data bus is used for memory, and if $\text{IO/M} = 1$, then address and data bus is used for I/O. Atmost this processor can address 256 no. of IO. We need 8 bit address. And that should be generated to 16 bit Address Bus. To resolve this, both lower and higher byte of the address bus is generating same I/O address, therefore we may use lower byte as well as the higher byte as the I/O address.

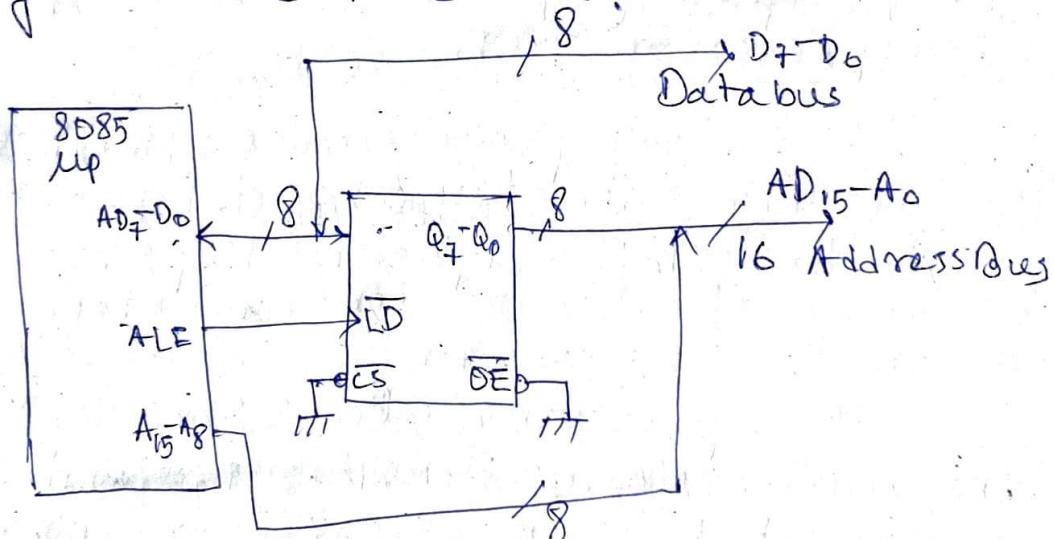


Fig: Address/Data Demultiplexing for 8085 MP.

b) Zero flag:

It is a component of 8 bit flag register. If the result of the last arithmetic operation is 0, then this flag is set otherwise reset.

Eg: MVI A, 10 (load 10H in register A).

$$SUB A \quad (A = A - A)$$

These set of instruction will set the zero flag to 1 as $10H - 10H = 00H$.

Carry flag:

Carry flag is generated when performing n bit operations and the result is more than n bits, then this flag becomes set i.e. 1, otherwise it become reset i.e. 0.

Eg. MVI A, 0C3
MVI B, D5
ADD B

This ~~results at~~ generates a carry which sets the carry flag.

6) Indirect register addressing is a addressing mode in the 8085 microprocessor where the actual memory address is not specified directly in the instruction. Instead, the instruction specifies a register pair, and the cont of that register pair is treated as the memory address where the data is located or where the operation will be performed. This allows for more flexible and dynamic memory access in the program.

In 8085 microprocessor, indirect register addressing is typically used with the HL registers. ~~to~~ This register pair is also known as pointer register. The content of the HL register pair is used as a memory address to access data in RAM.

Eg : LXI H, 4050H

This instruction loads L register with 50H & loads H register with 40H hence HL pair representing 4050H.

MOV A, M instruction
~~In this register~~ indirect register addressing is used. Here M basically means the pointer register HL. Here the data located at the memory address 4050H, which is pointed by HL register, the data is copied/loaded from this 4050H memory address to the Accumulator.

b) HLDA

HLDA is an output control signal generated by the microprocessor in response to a HOLD request from external devices. When an external device needs to access the system buses temporarily, it asserts the HOLD signal to the microprocessor.

- ① When up receives a HOLD request, it suspends its current operations and prepares to release the buses so that the requesting device can gain control of the bus.
- ② The up responds to the HOLD request by issuing the HLDA signal, indicating that it has received the HOLD request and is ready to relinquish control of system bus.
- ③ Upon receiving the HLDA signal from the up, the requesting device can take control of the system buses and perform its data transfer or the other operation.
- ④ Once the external device completes its task it relinquishes control over buses and the up gains the control over buses and end its HOLD state.

ALE

ALE also known as Address Latch Enable, is a pulse generated by the up during the machine cycle to latch the lower order address byte of the address bus.

- ① ALE is active during the first T state of every machine cycle. During the first T-state the lower order address byte are available in the address bus.

- ② During the second T-state⁽⁴⁾ the lower order address bytes is not available as the lower order address bus act as data bus in purpose of opcode fetch, memory write, memory read operation.
- ③ So, we need to latch the address so that the actual address does not change. In this particular case ALE is used to enable address latching.

7>

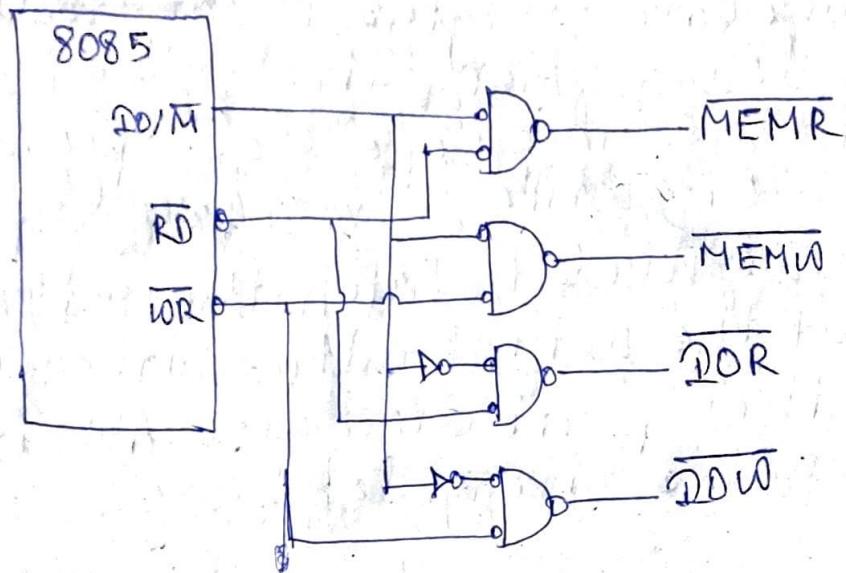
- Q) In 8085 microprocessor we can interface both memory & I/O devices. And we can do read and write operation on memory and I/O also, so it is necessary to generate 2 separate read, write signals for memory and I/O devices.

To implement this three signals are used $\overline{D/M}$, \overline{RD} , \overline{WR} .

- ① $\overline{D/M}$ signal is used to indicate whether it is a memory operation or I/O operation. If it is 0, then it is memory operation. If it is 1, then it is an I/O operation.

- ② \overline{RD} signal is used to indicate, it is a read operation & it is an active low signal.

- ③ \overline{WR} signal is used to indicate, it is a write operation & it is also an active low signal.



<u>Machine Cycle</u>	<u>Status</u>	<u>Control Signal</u>
Memory read	$\overline{IO/M} \quad S_1 \quad S_0$ 0 1 0	$\overline{RD} = 0$
Memory write	0 0 1	$\overline{WR} = 0$
I/O read	1 1 0	$\overline{RD} = 0$
I/O write	1 0 1	$\overline{WR} = 0$

- ① If $\overline{IO/M} = 0$ and $\overline{RD} = 0$,
it is memory read operation
- ② If $\overline{IO/M} = 0$ and $\overline{WR} = 0$,
it is a memory write operation
- ③ If $\overline{IO/M} = 1$ and $\overline{RD} = 0$,
it is an I/O read operation.
- ④ If $\overline{IO/M} = 1$ and $\overline{WR} = 0$,
it is an I/O write operation.

b) Program Counter

The program counter is a 16-bit register in the microprocessor responsible for holding the memory address of the next instruction to be fetched and executed.

Functions:

- ① The PC ensures the proper sequencing of instructions in the program.
- ② The PC controls the flow of the program by pointing to the memory address of the next instructions. It determines the order in which instructions are executed.
- ③ During conditional jumps or subroutine calls, the PC is modified to point to a different memory address, enabling branching to other parts of the program.

Stack Pointer

The stack pointer is a 16-bit register that points to the top of the stack in memory. The stack is a special region in memory used for storing data temporarily, primarily during subroutine call and interrupts.

Functions

- ① The stack pointer manages the operations on the stack, such as push (storing data) and pop (retrieving data).
- ② During a subroutine call, the return address is pushed onto the stack. The stack pointer facilitates this process.
- ③ During interrupt handling, it pushes some register contents onto the stack; stack pointer assists this process.

8)

- a) The Programmable Peripheral Interface (PPI) is a versatile integrated circuit (IC) used in microprocessor-based systems to interface with various peripheral devices. The PPI can be programmed to perform specific tasks.

Functions

- ① Data transfer: The PPI allows the microprocessor to read data from external devices connected to its input ports and enables the CPU to send data to external devices connected to its output port.
- ② I/O address decoding: The PPI is assigned a specific I/O address in the system. When the CPU outputs this address on the address bus, the PPI recognizes and responds accordingly.
- ③ Handshake and Control Signals: PPI provides control signals as RD, WR, CS and AO and handshake signals like ACK and DNTR to facilitate communication and coordination between the CPU and peripherals.
- ④ ~~④~~ BCD/ASCII Conversion: The PPI can be programmed to perform BCD to ASCII conversion and vice versa.
- ⑤ Strobed Input/Output: The PPI can be configured to handle strobed input/output, which is useful in interfacing with devices that require data to be latched or synchronized.
- ⑥ Mode Control: The control register in the PPI allows the CPU to configure each port as either input or output and set other operating modes as needed.

④ Interfacing with Various Peripherals:

(5)

The PPP's flexibility makes it suitable for interfacing with a wide range of peripheral devices such as keyboards, displays, sensors, motors and other input/output devices.

- b) The 8085 has five interrupt inputs. One is called INTR, three are called RST 5.5, 6.5 and 7.5, respectively and the fifth is called TRAP, a nonmaskable interrupt. These last four (RSTS and TRAP) are automatically vectored to specific locations on the memory page 00H without any external hardware. They do not require the INTA signal or an input port, the necessary hardware is already implemented inside the 8085. These interrupts and their call locations are as follows:

Interrupts	Call locations
TRAP	→ 0024H
RST 7.5	→ 003CH
RST 6.5	→ 0034H
RST 5.5	→ 002CH

The TRAP has the highest priority, followed by RST 7.5, 6.5, 5.5.

When these interrupts trigger program counter loads a specific vector location and execute the program.

- c) TRAP is used for power failure recovery routines or to handle system-level errors that requires immediate attention.

Unlike the maskable interrupts, the TRAP cannot be disabled by clearing the Interrupt Enable flip flop, making it a high priority interrupt that is always

serviced when it occurs.

Though hold and trap both are high priority, HOLD has greater priority than trap. Since TRAP is an interrupt and to execute it we need to execute its PSR which is in the memory. As hold signal is activated the database becomes strike states so the trap is unable to execute its corresponding PSR.