**1. State the Function of the Following Instructions:**

1. **LHLD (Load H and L Directly)**
   * This instruction loads the **HL register pair** with data from a memory location.
   * Example: LHLD 2500H → Loads H with content of 2501H and L with 2500H.
2. **XCHG (Exchange HL and DE Registers)**
   * Swaps the contents of **HL and DE register pairs**.
   * Example: XCHG swaps (H ↔ D and L ↔ E).
3. **RAL (Rotate Accumulator Left through Carry)**
   * Rotates accumulator left with carry.
   * Example: If A = 11001001 and CY = 1, then after RAL, A = 10010011 and CY = 1.
4. **PUSH B**
   * Pushes **BC register pair** onto the stack.
   * Stack Pointer (SP) decrements twice.
5. **DAD H (Double Add HL with HL)**
   * Adds HL with itself (HL = HL + HL), useful for shifting.
6. **XTHL (Exchange Top of Stack with HL Register Pair)**
   * Swaps the HL register pair with the top two bytes of the stack.
7. **DAA (Decimal Adjust Accumulator)**
   * Converts binary addition result into **BCD (Binary-Coded Decimal)** format.
8. **PCHL (Load Program Counter with HL Contents)**
   * Transfers control to the memory address stored in HL.
9. **RIM (Read Interrupt Mask)**
   * Reads the status of interrupts.
10. **STC (Set Carry Flag)**

* Sets the Carry Flag to 1.

**2. Address Lines for 2K Byte Memory:**

* **Formula:** Number of address lines = log₂(size in bytes)
* **2K = 2048 bytes → log₂(2048) = 11 address lines.**

**3. Starting Address of 8K Memory Chip (Last Address = FFFFH):**

* **8K = 8192 bytes → log₂(8192) = 13 address lines.**
* **Last Address: FFFFH**
* **Starting Address:** FFFFH - (8192 - 1) = **E000H**

**4. Last Address for 4K Memory Chip (Starting Address = 2000H):**

* **4K = 4096 bytes → log₂(4096) = 12 address lines.**
* **Last Address:** 2000H + 4096 - 1 = **2FFFH**

**5. Program to Check if Number at 8000H is Positive, Negative, or Zero:**

assembly

CopyEdit

LDA 8000H ; Load value from 8000H into accumulator

CPI 00H ; Compare with zero

JZ ZERO ; Jump if zero

JM NEGATIVE ; Jump if negative

MVI A, FFH ; If positive, store FFH

STA 8050H

HLT

NEGATIVE:

MVI A, FEH ; Store FEH if negative

STA 8050H

HLT

ZERO:

MVI A, FDH ; Store FDH if zero

STA 8050H

HLT

**6. Reverse Block of Data Stored in Memory Locations XX55H to XX5AH:**

assembly

CopyEdit

LXI H, XX55H ; Source address

LXI D, XX80H ; Destination address

MOV C, 06H ; Counter (6 bytes)

REVERSE:

MOV A, M ; Load byte from source

STAX D ; Store in destination

INX H ; Increment source pointer

DCX D ; Decrement destination pointer

DCR C ; Decrement counter

JNZ REVERSE ; Repeat until counter is zero

HLT

**7. Program to Find the Largest Number in a Given Block:**

assembly

CopyEdit

LXI H, 2050H ; Load starting address

MOV C, M ; Load count (N) into C

INX H

MOV A, M ; Load first number into A

LOOP:

INX H ; Move to next number

CMP M ; Compare A with memory

JNC SKIP ; If A >= M, skip update

MOV A, M ; Update A with the new max

SKIP:

DCR C ; Decrement count

JNZ LOOP ; Repeat until all numbers are checked

STA 2060H ; Store max value

HLT

**8. Program to Sort 8 Numbers in Ascending Order (Starting at 2050H):**

assembly

CopyEdit

LXI H, 2050H ; Load array base address

MOV C, 08H ; Number of elements

OUTER:

MOV D, C ; Store count in D

LXI H, 2050H ; Reset pointer

INNER:

MOV A, M ; Load element

INX H

CMP M ; Compare with next element

JC SKIP ; If already in order, skip swap

MOV B, M

MOV M, A ; Swap

DCX H

MOV M, B

SKIP:

INX H

DCR D

JNZ INNER

DCR C

JNZ OUTER

HLT

**9. Memory System Design for 8085 with 8K EPROM and 8K RAM:**

* **Memory Map:**
  + **EPROM:** 0000H - 1FFFH (8K)
  + **RAM:** 2000H - 3FFFH (8K)
* **Decoder (74LS138) Logic for Chip Selection:**
  + A15 = 0 → Selects EPROM
  + A15 = 1 → Selects RAM

(Draw a **detailed schematic** of **8085 with EPROM and RAM interfacing**)

**10. Memory System Design for 8085 with 4K ROM Using 74LS138 Decoder:**

* **Memory Map:**
  + **ROM:** 0000H - 0FFFH (4K)
* **Address Decoding with 74LS138:**
  + A15 - A12 connected to **Decoder Inputs**
  + A11 - A0 used for **Address Selection**

(Draw a **circuit diagram** showing **ROM interfacing with 8085 using 74LS138**)

**11. Hexadecimal Countdown from FFH to 00H with 1ms Delay Using Register C:**

assembly

CopyEdit

MVI C, 64H ; Load delay count

MVI A, FFH ; Start from FFH

COUNT:

CALL DELAY

OUT 01H ; Output to Port 1

DCR A ; Decrement A

JNZ COUNT ; Repeat until 00H

HLT

DELAY:

DCR C

JNZ DELAY

RET

**12. Square Wave Generation (500µs Period, System Clock = 325ns):**

assembly

CopyEdit

MVI B, 3EH ; Load delay count

LOOP:

XRA A ; Toggle D0

OUT 01H ; Output to Port

CALL DELAY ; Generate delay

JMP LOOP ; Repeat

DELAY:

DCR B

JNZ DELAY

RET

**13. Stack & Stack Pointer in Subroutine Programs:**

* **Stack:** Memory used for temporary storage.
* **Stack Pointer (SP):** A register that stores the stack’s top address.
* **Example:**

assembly

CopyEdit

PUSH B ; Save BC onto stack

POP D ; Restore into DE

**14. Hardware vs Software Interrupts in 8085:**

* **Hardware Interrupts:** Triggered by external devices (TRAP, RST7.5, etc.).
* **Software Interrupts:** Triggered by instructions (RST0 - RST7).

**15. Function of RIM & SIM Instructions:**

* **RIM (Read Interrupt Mask):** Reads interrupt status.
* **SIM (Set Interrupt Mask):** Enables/disables interrupts.