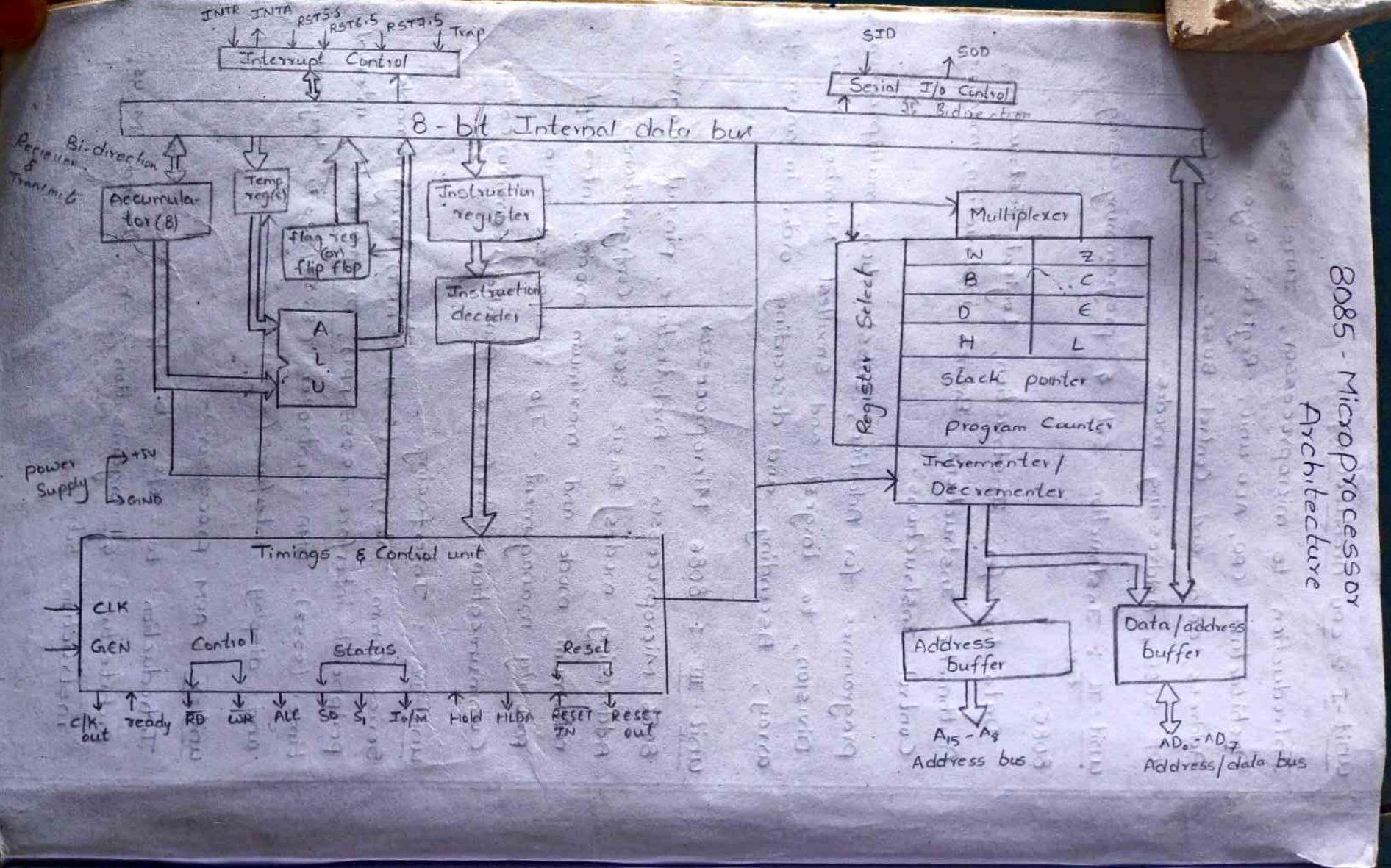

8085 - Microprocessor Architecture
INTR
INTA
RST5.5
RST6.5
RST7.5
Trap
Interrupt Control
SID
SOD
Serial I/o control
Bi-direction
8 - bit Internal data bus
Bi-direction
Register & Transmit
Accumulator (8)
Temp reg(s)
Flag reg (or) flip flop
A L U
Instruction register
Instruction decoder
Multiplexer
Register Selector
W
Z
B
C
D
E
H
L
Stack pointer
Program Counter
Incrementer / Decrementer
power Supply
+5V
GND
Timings & Control unit
CLK
GEN
Control
Status
Reset
CLK out
ready
RD
WR
ALE
S0
S1
IO/M
Hold
HLDA
RESET IN
RESET out
Address buffer
Data/address buffer
A15 - A8
Address bus
AD0 - AD7
Address/data bus

Register :- It stores the data temporarily

Accumulator (A) :- It performs the operations of ALU and stored the final result

ADD B means A + B → A [final result in A]

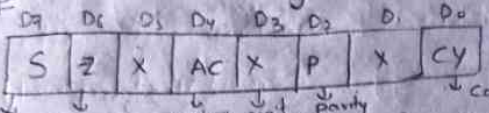
one of the operand to be accumulator, A is Accumulator, B is register

Temp reg :- Temporary ga result store chesukodaniki use chestham.

→ It is non-programmable.

[A+B+c bit count chesukoni ante A+B result ni store chesi c ADD chestham]

flip-flop :- one bit storage device



Total 8 flags

3 flags are don't care either 0 or 1

Remaining 5 flags

Sign flag, Carry flag, Auxiliary carry flag, parity flag, Zero flag

→ If Carry comes after 8 bit ADD then Carry flag is set=1 otherwise reset [Carry flag]

→ parity for error checking / correction

[parity] Even parity flag = set = 1 [even number of 1's]

odd parity flag = Reset = 0

→ 4 bit tarvatha Carry vasthe Auxiliary Carry [Nibble = 4]

[Auxiliary] Carry vasthe set = 1 otherwise Reset = 0

→ Result of operation zero ayithe Zero flag anedhi set = 1 otherwise Reset = 0

→ +ve & -ve :- If the result of operation is -ve then set Sign flag if it is +ve then reset = 0

Instruction Register :- Receive inputs from Internal data bus

Instruction decoder :- are type of code Convert into another

Hexadecimal code from InsRegister Convert into Binary or another type of code [0's and 1's]

Register Array

→ W, Z are non-programmable registers

→ B, C, D, E, H, L Length is 8 bit Generally we use these registers

→ BC, DE, HL Combine ga use chesthe 16 bit

Stack pointer → addressing [16 bit]
↓
first in last out

→ To store the bulk of data we use stacks
→ It point out the Address of the stack. It is working of stack pointer

Program Counter : (16 bit)

→ It stores the Address of next instruction to be executed
→ It incremented by, after execution. [execution ki velle dani address store chesthundhi]

Buffer : Same as input, output will occur.

1 byte = 8 bit

$A_0 - A_{15}$

$A_0 - A_7$ Lower byte loworder bus

$A_8 - A_{15}$ higher byte

$AD_0 - AD_7$ → Multiplexer Address bus
↳ Data store avuthundhi

$A_{15} - A_8$ → Higher order bus
Store address

Control unit

→ It mainly Control's all type of operation
→ Last pin Ground. Max Volt 3.5V - 5V
→ CLK, GEN → flip flops have clock pulses. [CLK, GEN 19 19]
Crystle oscillator generates the frequency [19, 19 pin] No of bits ↑ frequency ↑ [3Mega Hedgz]
→ Reset ← Reset in → program struck avuthundhi [program antha erase avvali anti]
↓ Anni devices Reset cheyyali Ante
only Mp Reset

Active low :- 0 is the Active Avuthundhi [which signals have — it denotes the Active low] Reset in

$\overline{RD}$ → Read Signal → To read the data in data bus
Memory device kani Input device Active avvali, ante $\overline{RD} = 0$

→ $\overline{WR}$ => used for write the data [Mp → data to data transfer cheyyadaniki]

ALE :- Address latch enable [↓ chip size]
differentiate the Address & data from Multiplexer
If ALE is 1 then Address, If ALE = 0 then data

$IO/\overline{M}$:- Input/o devices & Memory devices connected to Mp
$IO/\overline{M} = 1$ I/O operation
$IO/\overline{M} = 0$ Memory operation

→ S_0 & S_1 are Status of operation [either read or write denotes]
→ Ready anedhi '0' ayithe waiting [operation perform cheyyadhu]

→ other devices request the address data of µp through the hold signal [Request anedhi hold dwara vastundhi]

→ HLDA: ⇒ request ni accept chesthe HLDA nundi other devices ki veltundhi

Interrupt:- To break the sequence of execution of main program
Control

INTR → Interrupt Request, signal

Trap → signal, High priority

INTA → Interrupt Acknowledgement [Acceptance is through it]
It is not a signal

parallel → transfer all bit at a time

Serial I/O. Control
↓
output anedhi next daniki Input ga veltundhi

→ bit by bit transfer.

→ SID → receiving

→ SOD → transfer

Microprocessor :-

Microprocessor is a multipurpose programming register base electronic device that reads the binary instruction from a storage device called memory accepts binary data as input and process data according to instruction and provides output

Architecture of 8085 µp :-

* 8085 is an 8-bit µp designed by intel in 1976 using NMOS technology.

It has the following features

→ 8-bit data bus.
→ 16-bit address bus, which can address up to 64KB
→ A 16-bit stack pointer
→ Six 8-bit registers arranged in pairs BC, DE, HL
→ Requires +5V supply to operate at 3.2MHz Single phase clock.

Accumulator :-

It is an 8-bit register used to perform arithmetic, Logical I/o & load/store operations It is connected to internal data bus & ALU

ALU (Arithmetic and Logic unit)

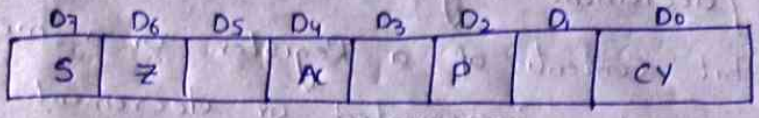
As the name suggest, it performs arithmetic & Logical operations like Addition, Subtraction AND, OR etc on 8-bit data.

Temporary Register :-

It is an 8-bit Register which holds the temporary data of Arithmetic and logical operations

Flag Registers (flip flops) :-

It is an 8-bit bit register having five 1-bit flip flops, which holds either 0 or 1 depending upon the result stored in the accumulator.



S → Sign flag
Z → Zero flag
AC → Auxilary carry
P → parity flag
C → Carry flag

Instruction register and decoder :-

It is an 8-bit register when an instruction is formed from memory then it is stored in the Instruction Register. Instruction decoder decodes the instruction present in the instruction register.

General purpose Registers :-

There are 6 general purpose Registers in 8085 µp i.e B, C, D, E, H & L. Each can hold the 8-bit data. These registers can work in pair to hold 16-bit data and their pairing Combination is like B-C, D-E, H-L

Stack pointer :-

It is also a 16-bit register works like stack which is always incremented / decremented by 2 during push & pop operation

program Counter

It is a 16-bit Register used to store the memory address location of the next instruction to be executed. µp increments the program counter whenever an instruction is being executed.

So that the program Counter point to the memory address of the next instruction that is going to be executed.

Increment / decrement Register :

→ The 8-bit Contents of a Register or a memory location can be incremented or decrement by 1

→ The 16-bit Register is used to inc / dec the Content of program Counter and stack pointer Register by 1

→ Increment or decrement Can be performed on any Register or a memory location.

Address buffer and Address-data buffer :-

The Content stored in the stack pointer and program Counter is loaded into the address buffer and address data buffer to Communicate with CPU. The memory and I/o chip are Connected to these buses, the CPU Can exchange the desired data with the memory and I/o chips.

Address Bus and data bus :-

Data bus Carries the data to be stored. It is bi-directional whereas address bus Carries the location to where it should be stored and it is unidirectional.

It is used to transfer the data & Address I/o devices.

Interrupt Control :-

It Controls the interrupts during a process. When a up is executing a main program and whenever an interrupt occurs, the up shift the Control from the main program to process the incoming request

After the request is completed the Control goes back to the main program.

There are 5 interrupt signals in 8085 µp
INTR, RST 7.5, RST 6.5, RST 5.5, Trap, $\overline{INTA}$

Serial Input / output Control :-

It Controls the serial data Communication by using these two instructions : SID (Serial input data) and SOD (Serial o/p data)

Timing and Control unit :-

It provides timings and Control signal to the µp to perform operations. following are the timing and Control signals, which Control external and internal Circuit

→ Control signals : READY, RD, WR, ALE
→ Status signals : S_0, S_1, $IO/\overline{M}$
→ DMA Signals : HOLD, HLDA
→ RESET Signals :
RESET IN
RESET OUT

Bus Structure :-

→ Microprocessor performs mainly 4 operations
(i) Memory Read (ii) Memory Write
(iii) I/O Read (iv) I/O write.

Memory Read : Reads the data from the memory
Memory Write : Writes the data into the memory
I/O Read : Accept the data from input devices
I/o write : Sends the data to output devices

→ All these operations are part of the Communication process between microprocessor and peripherals [I/o devices or Memory]

→ To Communicate with peripheral or Memory the Up needs to perform the following steps
(i) Identify the peripheral or memory location with it address.
(ii) Transfer binary Information [data & Instructions]
(iii) provide timing or Synchronization Signals.

The Up performs these functions using 3 sets of Communication lines called buses
(i) Address bus
(ii) Data bus
(iii) Control bus

Address bus :-

An address bus is a group of 16 lines identified as A_0 to A_{15}.

→ The Address bus is unidirectional
→ The microprocessor uses the address bus to identify peripheral or memory device.
→ The Address bus is used to Carry a 16 bit address.

→ Number of address lines of µp determines its capacity to identify different memory location.

→ The 8085 µp with its 16 address lines, is capable of addressing 2^{16} (65,536) (64KB) memory locations.

Data bus :-

→ Data bus is group of 8 lines used for data flow.

→ These lines are bi-directional that is data flow in both directions, between µp and memory and peripheral devices.

→ µp uses the data bus to transfer the binary information.

→ The 8 data lines enables the microprocessor to manupulate 8 bit data ranging from 0-0 to F-F [0-255]

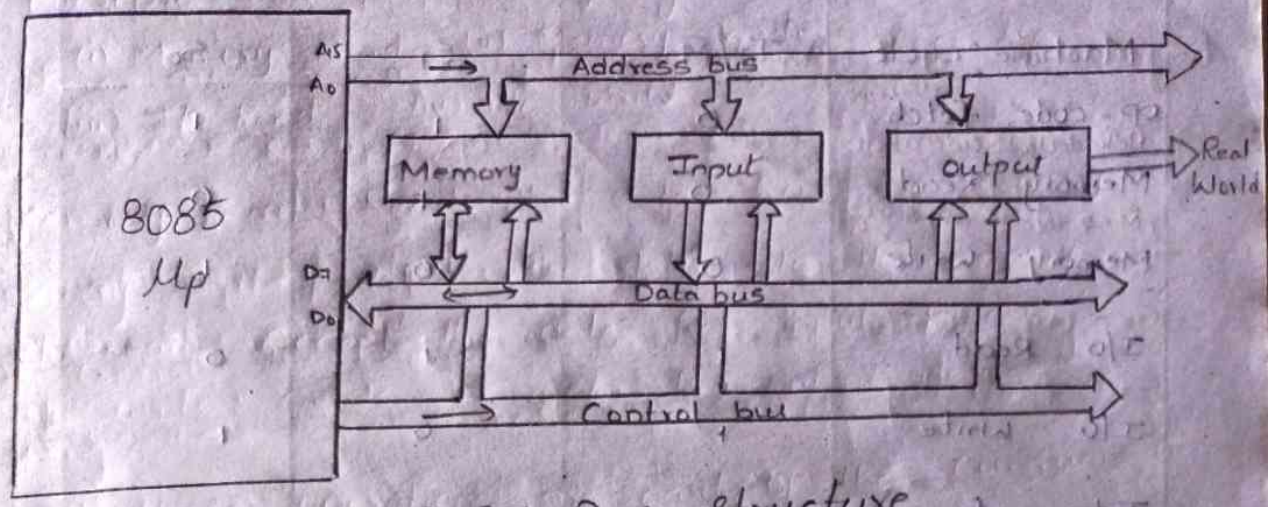
Control bus :-

→ The Control bus is Comprised of various single lines that Carry synchronization signals.

→ These are not group of lines like address or data buses but individual lines that provide a pulse to indicatioe µp operation.

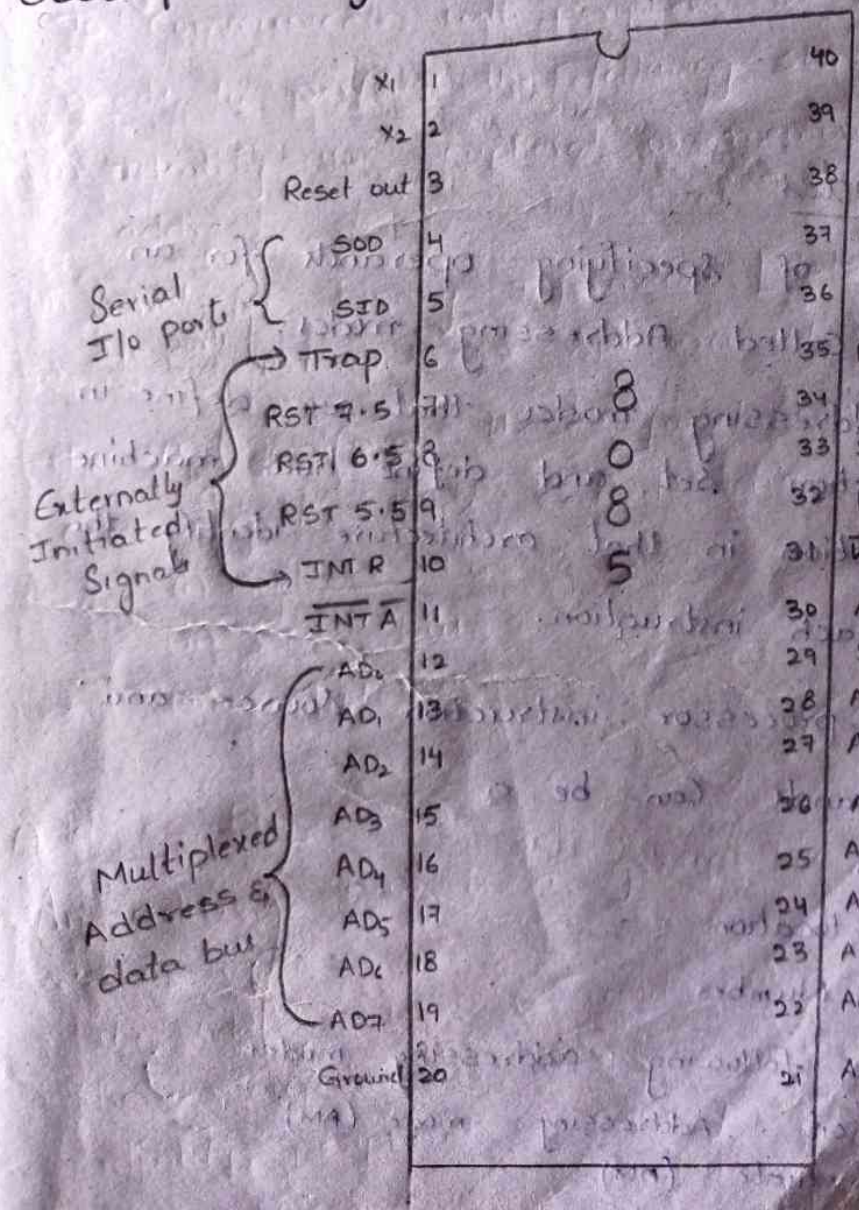
→ µp generates specific Control signals for every operation it performs.

→ This signals are used to identify a device type with which the µp intends to Communicate



8085 Bus structure

8085 pin diagram



Machine cycle	$IO/\bar{M}$	S_1	S_0
op-code fetch (Receive)	0	1	1
Memory Read (Hexadecimal code)	0	1	0
Memory Write ($\bar{M}=0$ then Memory device Select)	0	0	1
I/O Read	1	1	0
I/O Write	1	0	1
Interrupt Acknowledgement	1	1	1

Explanation of pin(No) are same as in 8085 Architecture.

Addressing modes → aka instruction Specify chose mode ni Addressing mode and Hexadecimal Number as input

The various ways of specifying operands for an instruction are called Addressing modes.

→ The various addressing modes that are define in a given instruction set and define how machine language instruction in that architecture identify the operands of each instruction.

→ In any micro-processor instruction source and destination operands can be a

* Register [B, C, D, E, H, L, Accumulator]
* Memory location [0000H - FFFFH]
* 8-bit Number

→ 8085 uses the following addressing modes

(i) Implicit Addressing mode (AM)
(ii) Immediate (AM)
(iii) direct AM
(iv) register AM
(v) register indirect AM

Implicit Addressing mode :-

→ In this mode the operand is hidden and the data to be operated is available in the instruction itself

Eg:- CMA, RRC, RLC
- CMA → Comparison
- RRC → Rotate Right
- RLC → Rotate left

[Related to accumulator] operations are performed as per Accumulator value

Immediate AM

→ In this mode the operand is specified within the instruction itself.

→ If the data is 8-bit then the instruction will be 2-bites. (1) first byte is opcode (2) second byte is operand followed by an 8-bit data

Eg:- MVI B, 45H
- MVI → Mnemonic
- B → operand
- 45H → source
- B → destination
- Whole Instruction Move 45H into B
- B = 45H → operand
- B → Register

LXI H, 3050H

Direct AM :-

→ The data to be operated is available inside a memory location and that memory location is directly specified as an operand.

→ The operand is directly Available in the instruction itself.

Eg:- LDA 2050H (Address)
LHLD 3050H (H register)

Address to data which is stored in 2050H Contain data it can be go through Accumulator LDA

Register AM

→ In this mode the operand is in general purpose register

→ The data to be operated is available inside a register and the register is operand.

∴ The operation is performed within various registers of the µp.

Ex MOV A, B (Register to data transfer)
ADD B A+B→A
(Increment) INR A

Register indirect AM

→ In this mode address of operand is specified by a register pair.

→ The data to be operated is available inside a memory location and that memory location is indirectly specified by a register pair.

BC indicates B
DE = D
HL = H } Register pair

Ex:- LDAX B

LDAX → it denotes register pair

register to Address (data)

X → Register pair