

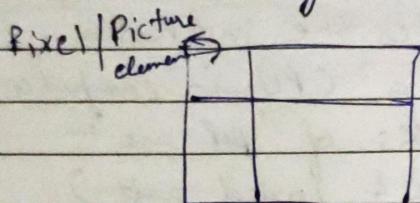
8085 μP

- ★ (b) Bit - A bit is a digit of the binary no. or code
- ★ Nibble - A group of 4 bits / 4 bit Binary no.
- ★ (B) Byte - 8 bits

$$(0110)_2 \rightarrow (0,1)$$

Each digit is a bit in binary system

→ In all other systems it is just called digit.



→ $1 \text{ MB} \rightarrow \text{Mega } (2^{20} \text{ locations})$
 Memory Locations \rightarrow Memory
 ↓
 Byte

★ Word Size - 16 Bit Binary no.

★ Double Word Size - 32 Bit Binary no.

★ Multiple Word - 64, 128 ...

★ Data - Key quantity that is operated by the instruction of program is called data.

★ Address - It is an identification no. of a memory location.

→ 8085 ~~HP~~ uses 16 Bit address.

★ Memory Word Size:- The memory word is the size of the binary information that can be stored on a memory location.
 → The Memory Word Size for 8085 ~~HP~~ is $\underbrace{8 \text{ bit}}_{\substack{\text{1 byte} \\ \text{like in 1MB}}}$.

★ Microprocessor (HP):- HP is a ~~semi~~ programmable semi-conductor device which fetches the instruction & data from the memory Decodes & Executes the instruction.

It is used as a CPU in computers. The basic functional blocks of HP are -
 - ALU (Arithmetic & Logical unit)
 - an array of registers
 - Control Unit

★ The HP is identified by the size of ALU.

* If ALU is 8 bit then HP is 8 bit
 * " .. is 16 bit 16 bits

★ Bus:- Data, Address & Control Signal to Transfer

\downarrow V _{cc}	\downarrow V _{cc}	\downarrow V _{cc}	IC _{RAM}
Data Bus	Address Bus	Control Bus	
K ₁	K ₂	K ₃	
K ₄	K ₅	K ₆	

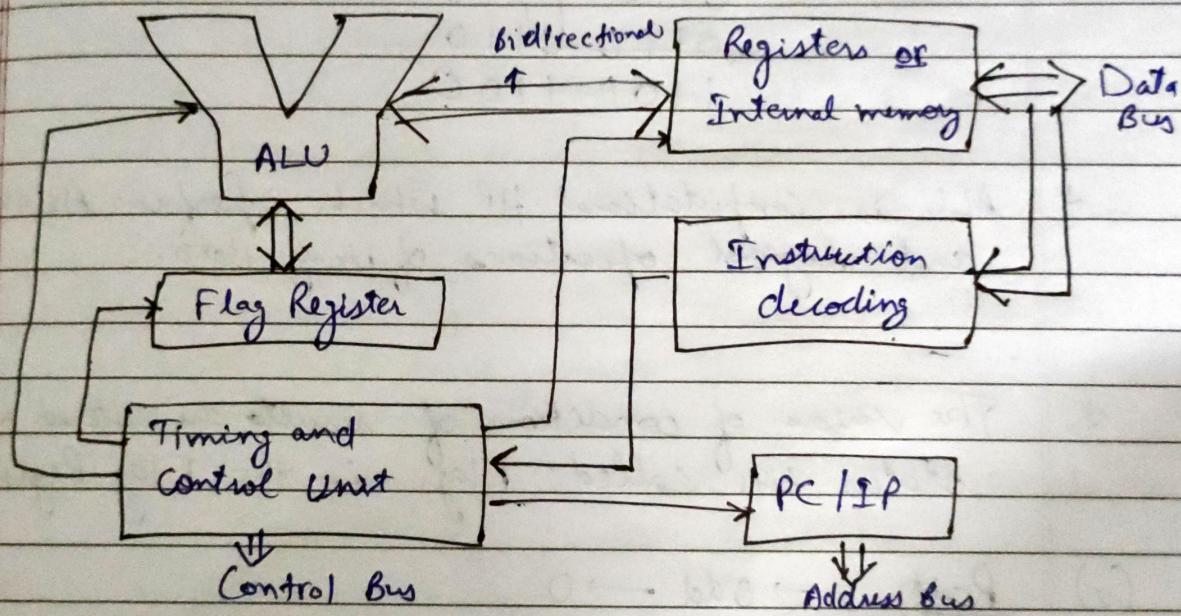
↳ A bus is group of conducting lines that carries data, address & control signals.

• CPU Bus :- The group of conducting lines that are directly connected to the CPU is called a CPU Bus.

* In a CPU Bus the signals are multiplexed.

↓
More than 1 signal is passed through the same line bit at different timing.

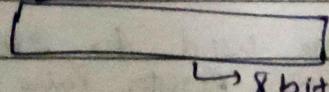
* Functional Building Blocks of 8085 HP :-



→ 8 bit register
Flag:

(i) Sign Flag :-

Result



$$\text{MSB} = 1 \rightarrow S = 1$$

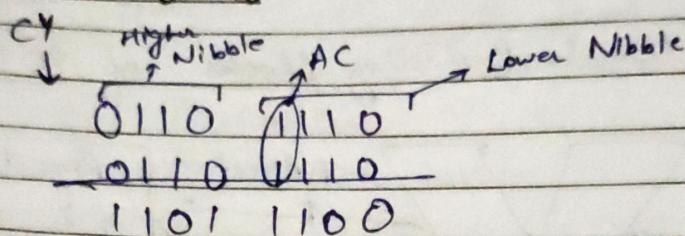
$$\text{otherwise } S = 0$$

(ii) Zero Flag - Z = 1

All bits 0
then Z = 1

(iii) CY (Carry) :-

If MSB generates carry then Flag = 1
otherwise Flag = 0

(iv) AC (Auxiliary Carry Flag) :-

* ALU is Computational IP which performs Arithmetic and logical operations of binary data.

* The various conditions of results are stored as status bits called Flags in the Flag Register.

(v) Parity :- Odd → 0
Even → 1

* A register is an internal storage device & it is known as internal memory.

→ The registers are used to store the data temporarily.

Register Pairs

BC → 16 Bit

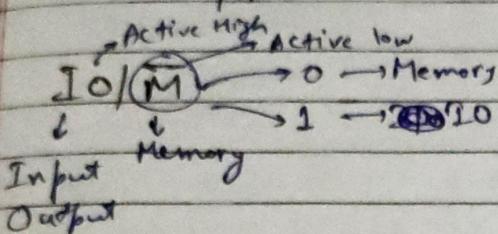
DE → 16 Bit

HL

PC → Address of Next Instruction to be fetched.
↳ 16 Bit for 8085 CPU

PIRKE NO.	
DATE	/ /

- * For any CPU, there will be a set of instructions given by the manufacturer.
- * For doing any useful work with the CPU we have to first write a program using these instructions & store them in a memory device external to the CPU.



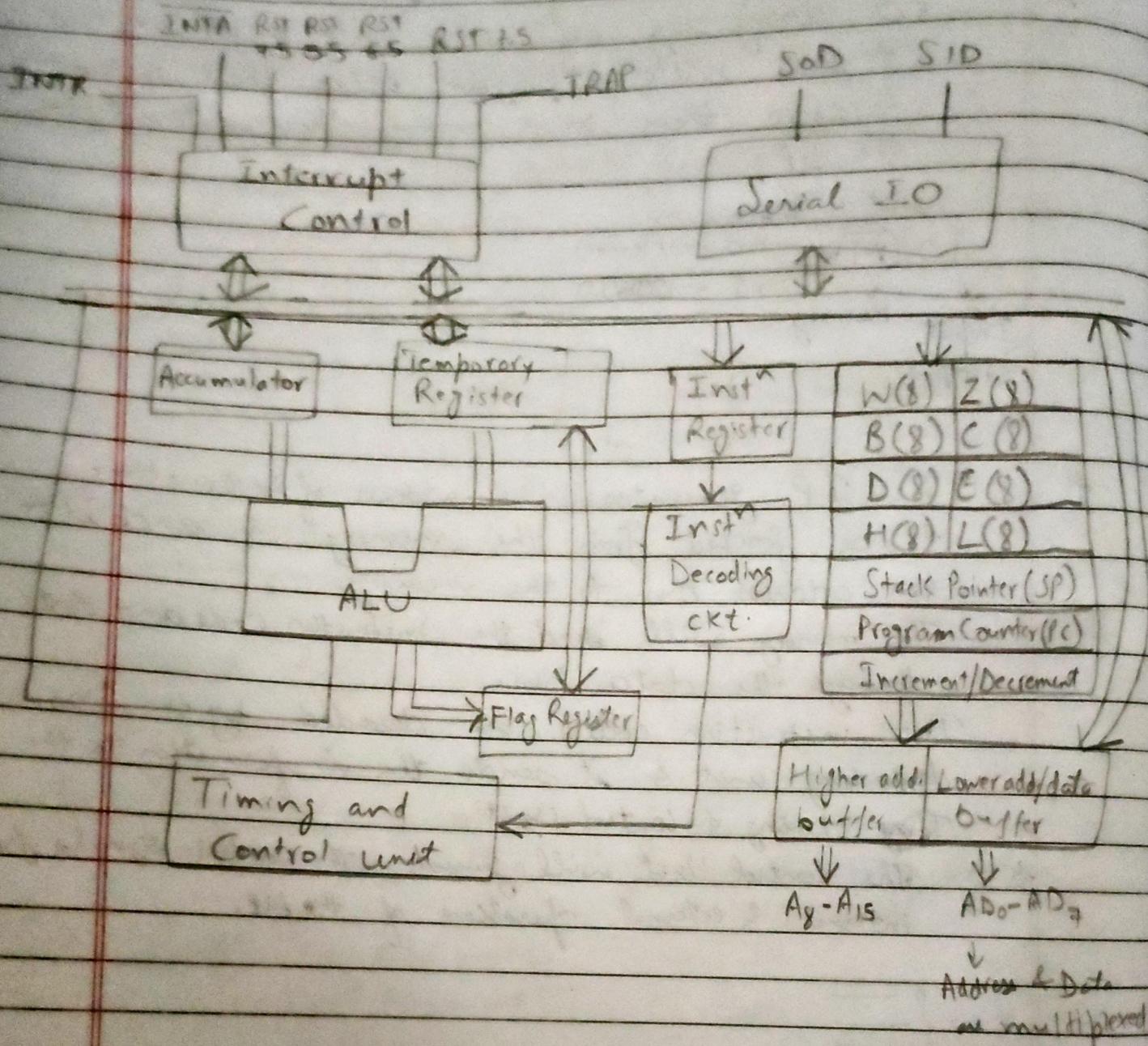
- * The IP generates the address of instructions to be fetched from the memory & sends it to the address bus to the memory. The memory will send the instruction code and data through the data bus. The instruction codes are decoded by the Instruction decoding unit & it sends the information to the Timing & Control Unit.
- * The Control Unit will generate the control signals for internal & external operations of the CPU.

$$Z = 0 \left\{ \begin{array}{l} \text{---} \\ \text{---} \\ \text{---} \end{array} \right.$$

J Z 001017
↓
Jump zero then jump to this address

$$\begin{aligned} A > B &\rightarrow A - B > 0 && \xrightarrow{\text{positive}} \text{Sign Flag} = 1 \\ A = B &\rightarrow A - B = 0 && \rightarrow \text{Zero Flag} = 1 \\ A < B &\rightarrow A - B < 0 \end{aligned}$$

8085 Architecture



8085

ALE (Address Latch Enable)

Related to
Timing +
Control
Unit

→ used for demultiplexing
data from Address or vice versa

Reset IN

RESET OUT

I/O M

RD

WR

READY

HLDA

HOLD

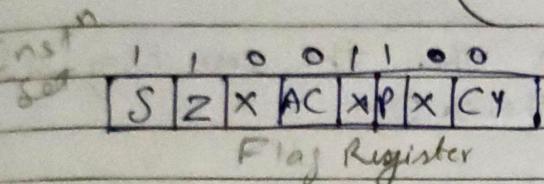
CLOCK (OUT)

S₀

S₁

X₁

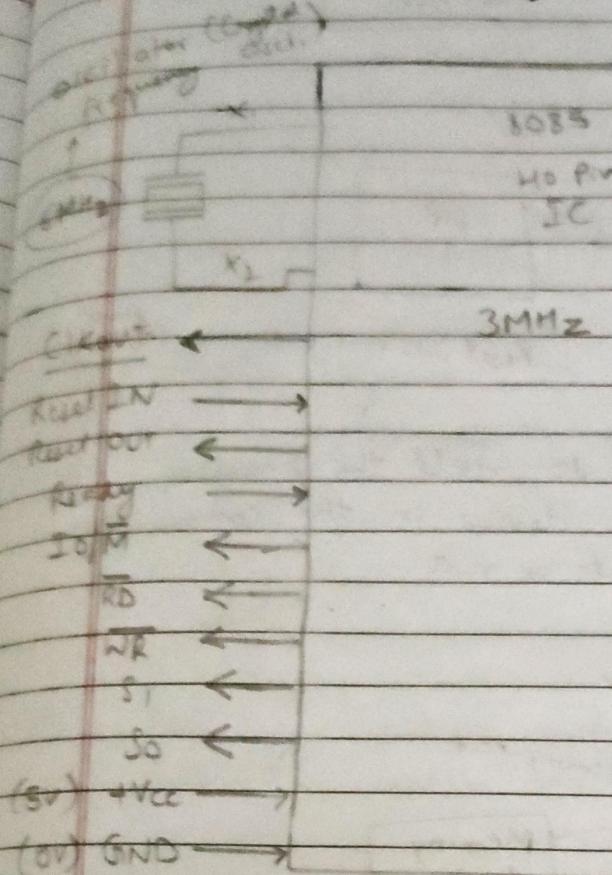
X₂



- * 8085 includes an ALU, a timing & control unit, Instⁿ Register, Decoder circuit, a register array & interrupt control & Serial I/O Control.
- * The operations performed by the ALU of an 8085 CPU are addition, subtraction, Increment/Decrement, Logical AND, OR, ~~exclusive OR~~, complement, Compare, left or right shift.
- * The Accumulator & temporary register are used to hold the data during ~~the~~ Arithmetic & Logical operation. After an operation the result is stored in an accumulator & the flag register is set/reset according to the result.
- * The Accu. & Flag register together is called as program status word (PSW).
 ↪ Group of 16 bit

- * If PSW is 0100H
then Accumulator is 01H (Higher bits)
Flag register is 00H. (Lower bits)
- * There are 5 flags in the flag register.
- * Sign Flag:- If the MSB is 1 then result is -ve, SF = 1
" " , 0 " " +ve, SF = 0
- * Zero Flag:- If the result is 0 then the zero flag is 1.
otherwise 0.
- * Auxiliary Carry:- If the lower Nibble generates the carry then
Add of two
that carry is AC and AC = 1.
- * Parity Flag:- odd no. of 1's then PC = 0
otherwise PC = 1
- * Carry Flag:- If the addition of 2 no.s generates the
carry after adding the MSB bits then that
carry is carry flag, CF = 1

8085 μP Pin diagram/ assignments :-



PAGE NO. / / /
DATE / / /

→ A15 - A8 → MOA

↔ AD₇ - AD₀ → Lower Address

← ALE → Address

← SID → Latch

→ SOD → Enable

← TRAP → Command

← RST 7.5 → 3MHz

← RST 6.5 → Power

← RST 5.5 → 7MHz

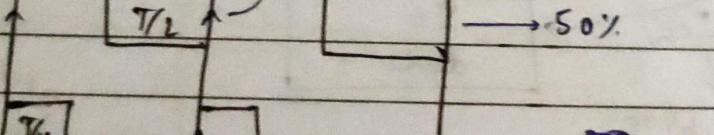
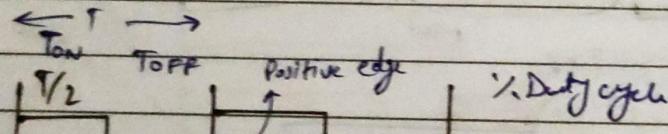
← INTR → Interrupt Request

— INTA → Acknowledge

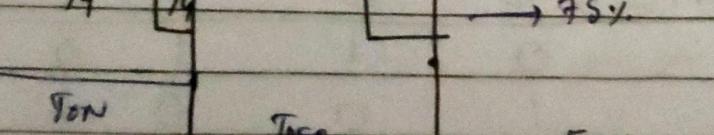
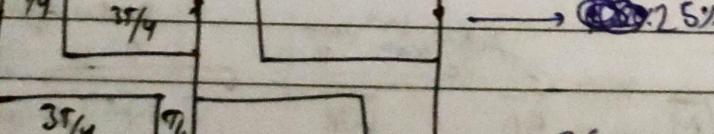
← HOLD → Hold

→ DMA → DMA

→ HLDA → HLDA



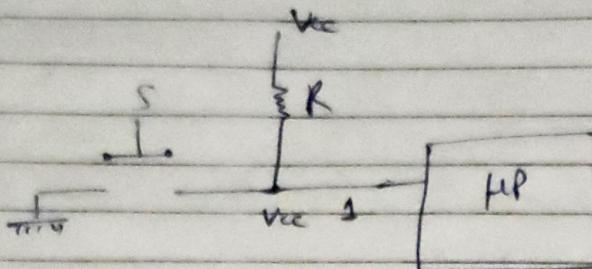
$$f_{CK} = \frac{\text{fcrystal}}{2}$$



$$f = 3 \text{ MHz}$$

then P0 is toggled

$$\% \text{ duty cycle} = \frac{T_{ON}}{T_{ON} + T_{OFF}}$$

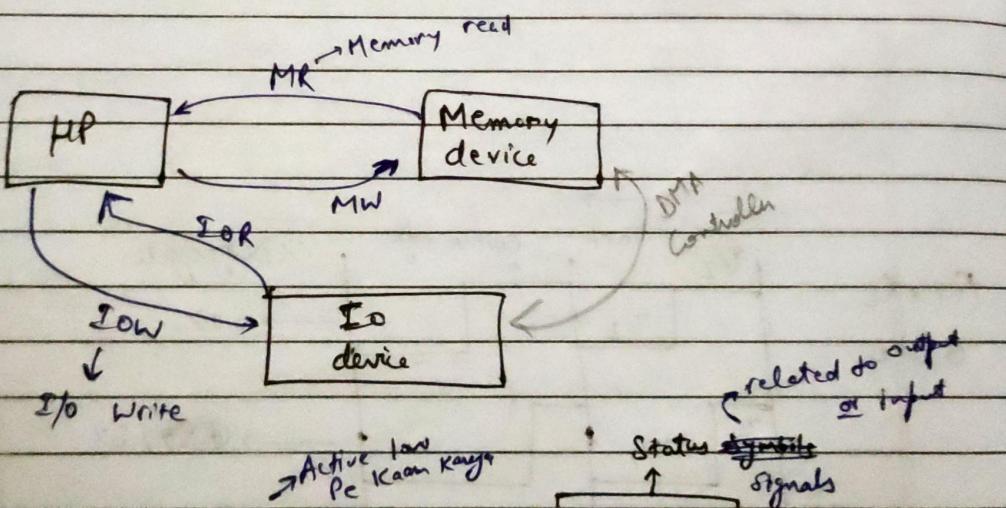


Reset IN → Active low ($\bar{0}$) ^{work}

* Ready → It is used to tell the processor whether other devices in the communication ready to communicate or not.

e.g. Printer

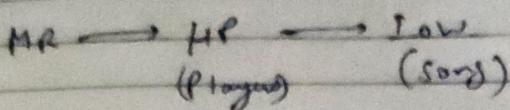
→ For slow devices.



	I0/M	\bar{RD}	\bar{WR}	S ₁ , Input S ₂ , Output	
MR	0	0	1	1	0
MW	0	1	0	0	1
IOR	1	0	1	1	0
IOW	1	1	0	0	1

MR → Input
MD output

e.g. Music

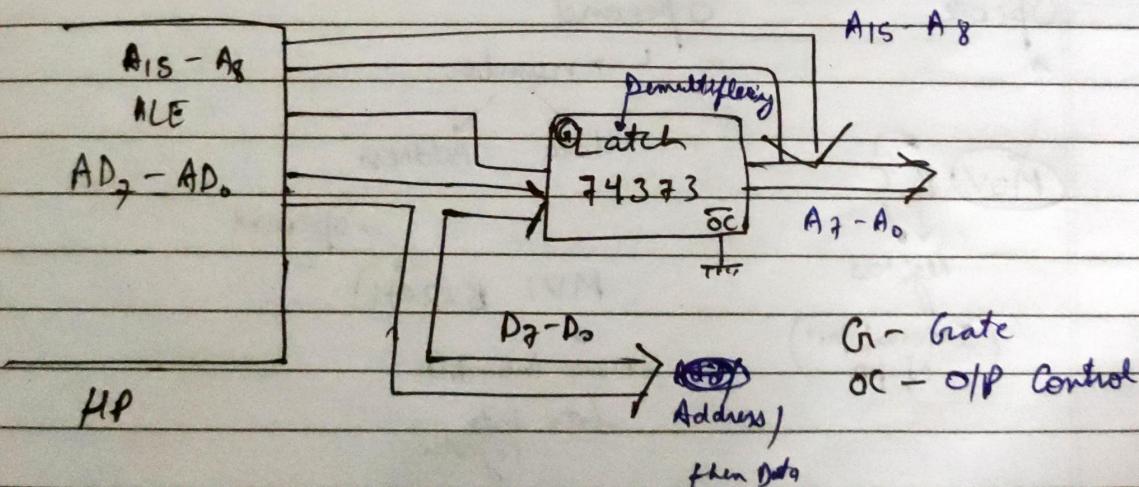


of Keyboard :- FOR \rightarrow HF \rightarrow MW

- * Status signals are defined for HP.

	A_{1S}	Higher Order		A_2	Lower Order		A_0
0	0	0		0	0		0000.H
		:		:		:	
1	1	1		1	1		FFFF..H

- * ALE = 1 , address selected
ALE = 0 , data selected



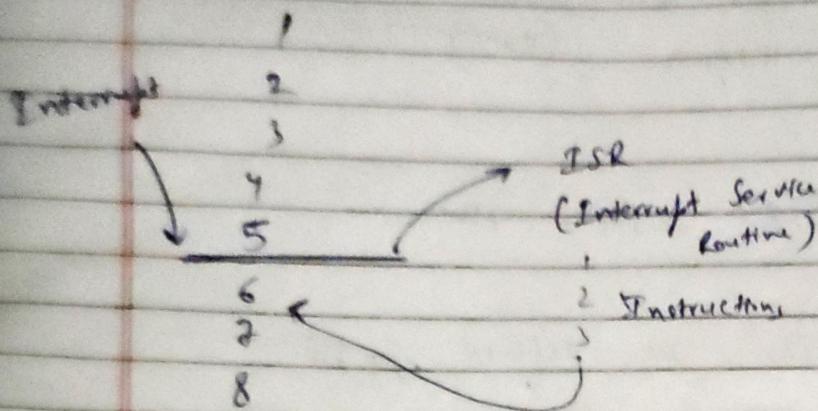
- * Serial Com. It is used for long distance

Bit by Bit Transferred

Slow

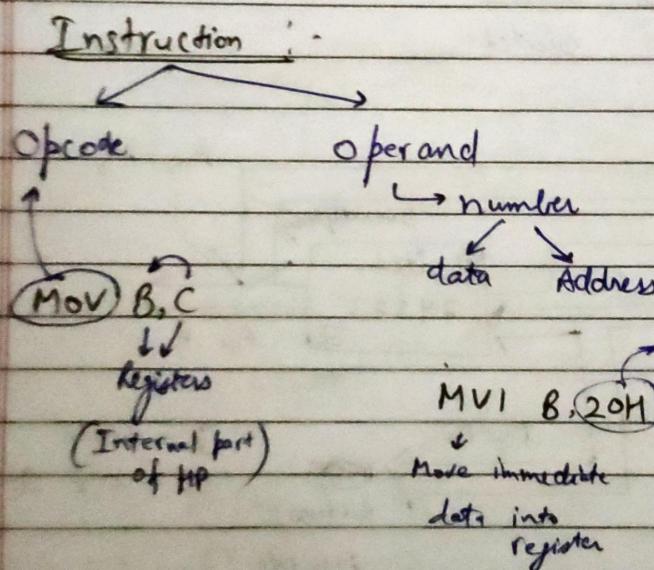
Serial Input device
Serial output device

* Parallel Comm. all transfer in one go
↪ fast



⇒ we have 5 Hardware & 8 Software interrupts.

8085 MP Addressing modes :-



* Registers :-

- General Purpose (A, B, C, D, E, H, L)
- Specific (Stack pointer, PC, Flag Register)

- * Stack → Arrangement of memory locations.
- * Addressing Mode :- The manner in which an operand is given in the instruction

→ Immediate A.M.

→ Data is directly given in the instruction

1 byte for opcode 1 byte for operand

MVI B, 25H (Example) → 2 byte
 MOV B,C (Not an Example)

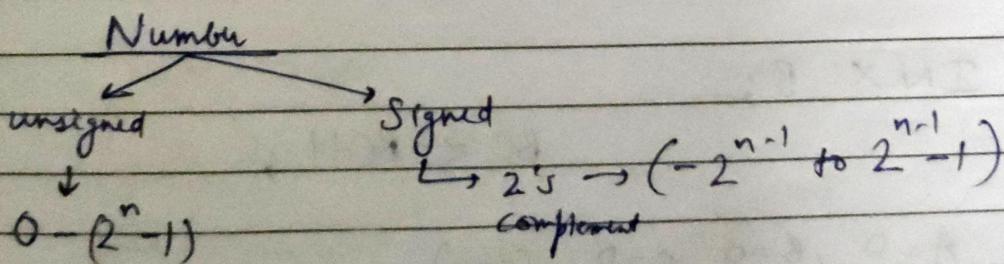
MOV B, 25H → Not valid instruction for 8085 HP

Note :-

$$\begin{array}{r}
 25H \\
 + 3AH \\
 \hline
 5FH
 \end{array}
 \quad
 \begin{array}{r}
 00100101 \\
 00110100 \\
 \hline
 0101\ 1111 \\
 \quad \quad \quad S \quad P
 \end{array}$$

$$\begin{array}{r}
 38H \\
 + 3AH \\
 \hline
 72H
 \end{array}
 \quad
 \begin{array}{l}
 \text{Hexa decimal} \\
 F - 15 \\
 10 - 16 \\
 11 - 17 \\
 12 - 18
 \end{array}$$

1 Carry



→ Dream 11 app banaani
 hai toh jeh no. que
 karegi.

Advantage — You Know Data.

Disadvantage — Memory Size big

Ex:- $LXI \text{ H } B2000H$

↑ ↓ Immediate
Register Pair Data

$\{ \begin{matrix} B & C \\ D & E \\ H & L \end{matrix} \}$ → Used to store 16 bit data

$B = 20H$
 $C = 00H$
 $D = 20H$
 $E = 00H$

* No. is not a part of opcode.

↪ 2) Register Addressing Mode :-

↪ Data is given in register.

Ex:- $\underbrace{\text{MOV } B, C}_{\text{opcode}} ; , B = C$
 ↓
 1 byte

Ex:- $\underbrace{\text{INR } B}_{\text{1 Byte}} ; \quad \leftarrow B+1$

Ex:- $\text{INX } B ;$

$BC \leftarrow BC+1 ;$

Ex:- $A = 0, B = 0, C = 0$ (say)

(a) MVI A, 00H
MVI B, 00H
MVI C, 00H] → 6 Byte

(b) MVI A, 00H
~~MVI~~ MOV B, A
MOV C, A] → 4 Byte

(c) SUB A
MOV B, A
MOV C, A] → 3 Byte → More efficient

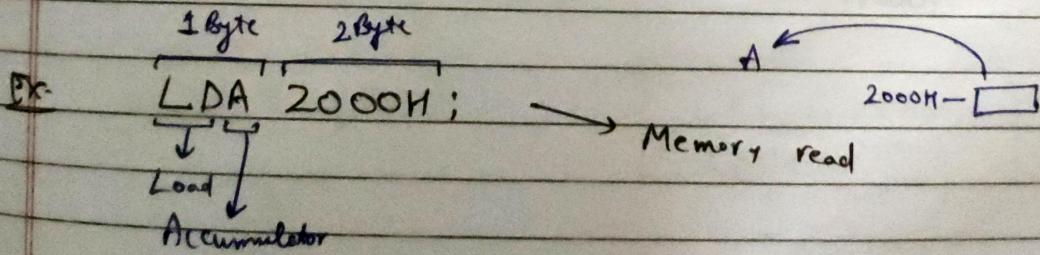
SUB A , ^{Accumulator}
A ← A - A

SUB C
A ← A - C

(d) LXI B, 0000H
MOV A, B] → 4 Byte

→ 3) Direct Addressing Mode :-

→ address is given in an instruction

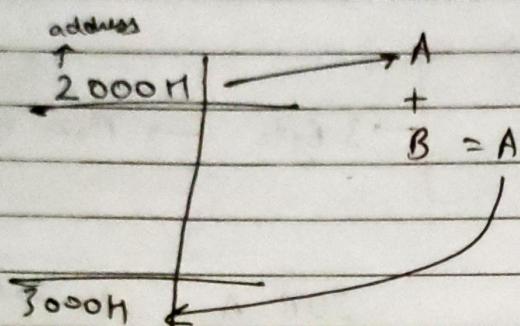


Ex

STA 2000H ; → Memory write
 ↓
 Store
 data from A to 2000H
 (Memory location)

A → [2000H]

Ex



MVI B, 02H → 2 Byte

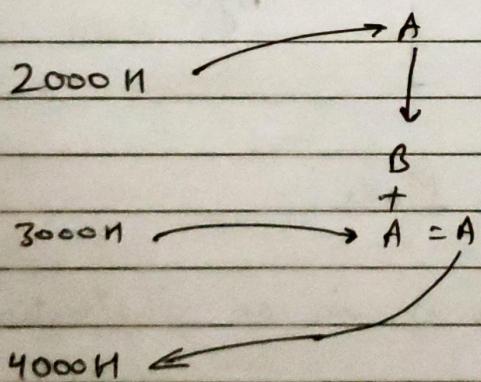
LDA 2000H ; → 3 Byte

ADD A, B ; → 1 Byte

STA 3000H ; → 3 Byte

9 Byte

Ex



PARALLEL	PARALLEL
PARALLEL	PARALLEL
PARALLEL	PARALLEL

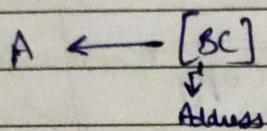
→ 4) Indirect Addressing Mode :-

→ Address is given in register pair
 ↓ ↓ ↓
 16 Bits BC DE HL

2 Byte → Address

Ex LDAX B, ~~4200H~~ → BC Ka data Para hona chahiye isko
 ↓ ↓
 Register Pair
 BC = 4200H implement karne se Pehle

Content of BC is loaded into accumulator.



Ex STAX B, ~~4200H~~

→ Implement Karne se Pehle B aur C Ka data para hona chahiye.

→ BC register pair mein A Ka data store ho jaayega

* Most of the Operations are performed using indirect addressing mode.

LDAX B

LXI B, 2000H

LDAX B

→ 5) Implied Addressing Mode :-

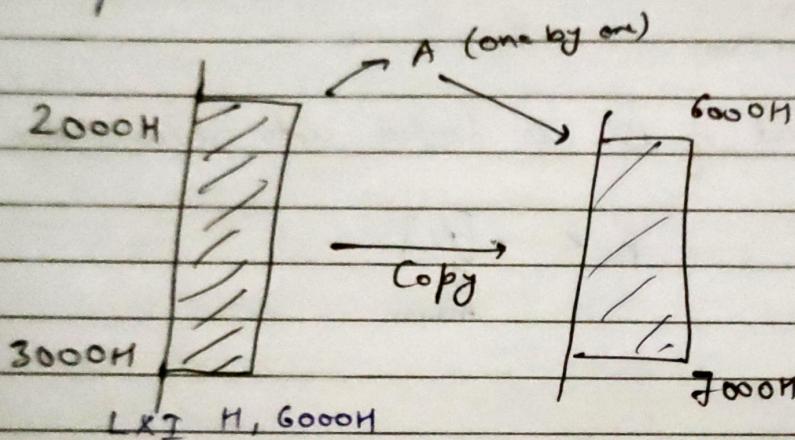
→ No operand (included in opcode)

STC → Set Carry Flag = 1

CMC → Complement Carry Flag = 1

→ No data/Address is specified / operand is not specified.

Ex



LXI B, 2000H

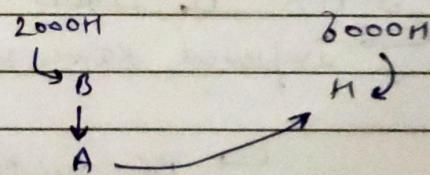
LDAX B

STAX H

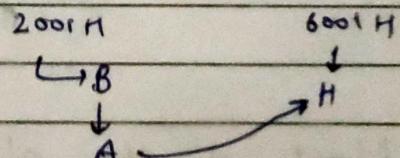
INX B

INX H

Increment



Now increment B^{4H}



Instruction Set of 8085 HP:

Do not affect any flag

8 bits

$2^8 = \text{inst}^n$

→ we will study 74 instⁿ

Categorized into 5 groups

- 1) Data Transfer Group instⁿ :- This group will deal with the transfer of data from one place to another place.

MVI R, 35H

MVI I R, 8 bits data → 2 Bytes instⁿ

→ whenever I is there then next given is data (not address).

8085HP

A

B

C

D

E

H

L

General Purpose Registers

Ex- MVI A, 32H

A ← 32H

2) LX(I) Rp , 16-bits data

Rp → Register Pair

load

BC DE HL

Ex LX2 B, 2000H → 3 Bytes

BC ← 2000H

B

Page No.	
Date	

Name inst? can be done by -

MVI B, 20H } 4 bytes
MVI C, 00H }

Note :- MOV R_d, R_s

→ It can only transfer from register to register

3) MOV R_d, R_s

↓ ↗ Source reg.
Destination reg.

$$7 \times 6 = 42 \text{ opcodes}$$

Ex MOV A,B

A ← B
2SH ↘ 2SH

4) MVI M, data → 2 Byte

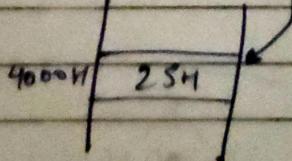
↓
Memory location is given by HL reg. pair

Ex MVI M, 2SH

[HL] ← 2SH

HP

2SH	-
H	2
40	00H

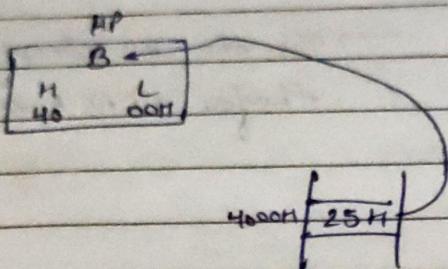


4) In this M is the memory location whose address is specified in H,L register pair

(1 opcode)

5) $MOV R, M$ → 1 Byte
 ↳ (HL) Rp Indirect Addressing mode

Ex $MOV B, M$



* (5 opcodes) are possible

6) $MOV M, R$ → Register Addressing mode
 ↳ (R)

(5 opcodes) are possible

7) $LDA 2000H$ → Direct Addressing mode
 ↓
 Address

$A \leftarrow [2000H]$

8) STA 3000H → Direct Addressing mode
 ↓
 Address

A → [3000H]

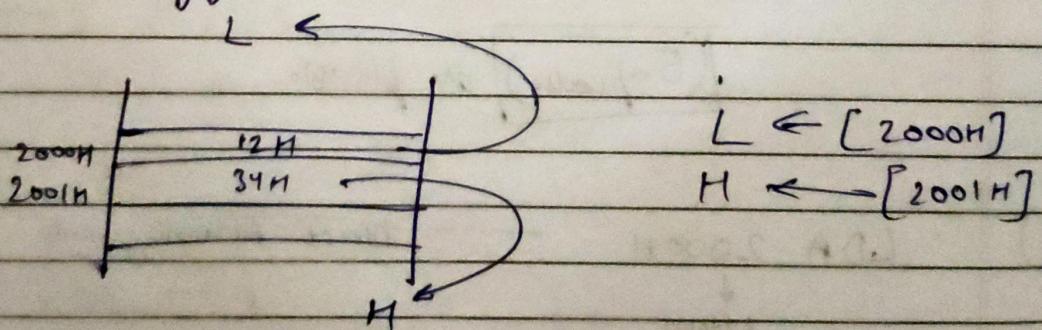
9) LDA X B ;
 1 opcode

Both are same
 LDAX H ;
 Mov A,M → This one we already studied
 therefore LDAX H is not used.

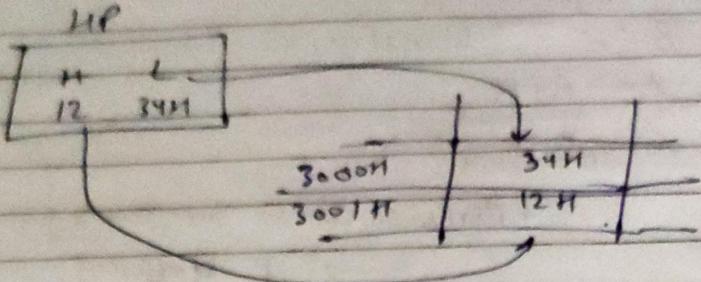
10) STAX D ;

Higher Byte → Lower Byte
 11) LHLD 2000H
 ↓
 Address

Directly
 Address given
 load H & L ko 2000H wali location ke data se
 but humein direct data nahi de rakha. Toh 2000H
 wala L mein jayega aur H mein 2001H wala
 jayega.



12) SHLD 3000H



13) ~~PC HL~~ \Rightarrow $PC \leftarrow HL$ \rightarrow Register Addressing Mode

~~Stores the address of next instⁿ~~

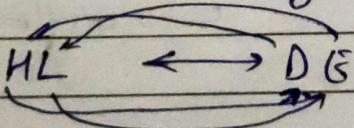
HL
4000H
PCHL
↓
H000H

13) ~~SP~~ SPHL

$SP \leftarrow HL$

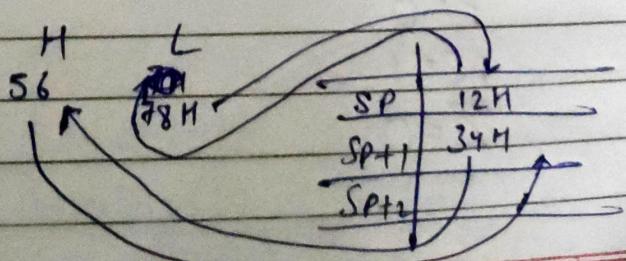
↑ Stack Pointer
SP gives the address of top of the stack
↑ Arrangement of memory location

14) XCHG exchange.



15) XTHL

↓
Exchanging with
Stack Pointer



pFlag may be affected

Arithmetic instruction set :-

1) ADD B :-
 System : 6 inst's
 $A \leftarrow A + B$
 16bit 8bit 8bit
 Result can be more than 8 bits

CF is affected
 carry

2) ADD M :-
 Indirect Addressing Mode
 Memory location
 16bit A $\leftarrow A + [M]$ CF may be affected
 8bit

3) ADD 25H :-
 Immediate 28bit A $\leftarrow A + 25H$
 A.M.

Note :-

$$\begin{array}{r}
 \text{FFH} \\
 + \text{FP H} \\
 \hline
 \text{① FE H} \\
 \text{CF} \quad \text{A}
 \end{array}
 \quad
 \begin{array}{r}
 \text{1111 1111} \\
 \text{1111 1111} \\
 \hline
 \text{① 1111 1110} \\
 \text{CP} \quad \text{F} \quad \text{E}
 \end{array}$$

overflow to go to stage
 FE display here

Sign Flag = $\frac{1}{\downarrow}$
 Seeing in PE

$$P=0$$

$$AC=1$$

$$Z=0$$

$$CF=1$$

4) SUB B
 $A \leftarrow A - B$

5) SUB M
 $A \leftarrow A - [M]$

6) SUB 25H
 $A \leftarrow A - 25H$

All Flags can be affected

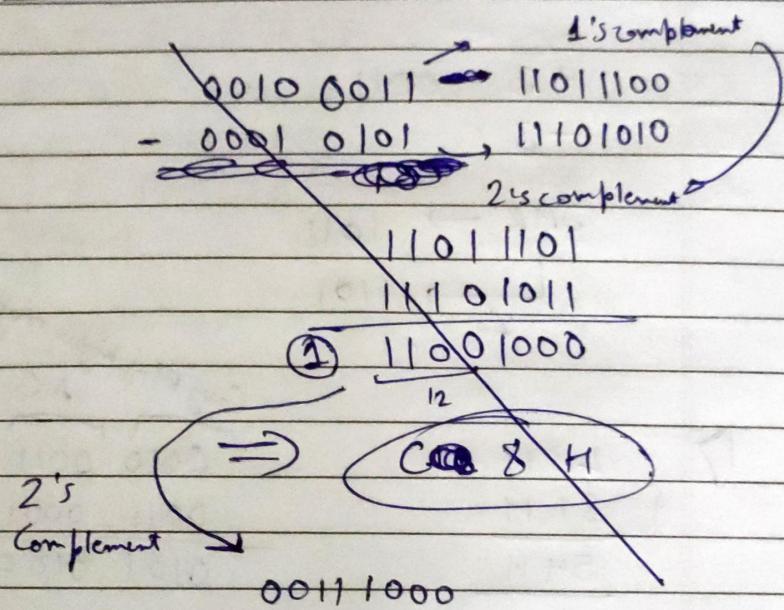
Borrow Flag -

$$\begin{array}{r} A \xrightarrow{\text{2's complement}} \\ + (-B) \xrightarrow{\text{2's complement}} \\ \hline \text{Result } A \xrightarrow{\text{2's complement}} \end{array}$$

Ex 23H
~~-15H~~

08H

Answer change



$+23H \rightarrow (35)_{10} \Rightarrow 00100011 \xrightarrow{\text{SMR}}$

$-15H \rightarrow -(21)_{10} \Rightarrow 10010101 \xrightarrow{\text{2's}}$

$$\begin{array}{r} 0010 0011 \\ 1110 1011 \\ \hline 0000 1110 \end{array}$$

0EH - but you say

0111 1100
0111 1101

PAGE NO.	
DATE	/ /

$$64 + 51 + 16 + 8 + 4 + 1$$

Ex - 23H 0010 0011
→ - 31H ↓
- 54H 1101 1100
aaaa +1

 1101 1101
cf 1100 1110
① 1010 1100 → 2's complement answer
↓ 2's complement
0101 0100
↳ 54

Ex $+ 3 = 0011$
 $- 3 =$

SMR \rightarrow 1011
L \rightarrow 1101
2's

1) 23H
+ 31H

54H

Carry flag higher nibble se check karte hai
A.C. lower nibble se check karte hai

↓
54H Carry Flag = 0
P=0

A.C. = 0

Z = 0

S = 0

2) 37H
+ 29H

60H

$$\begin{array}{r}
 37 \\
 -23H \\
 -31H \\
 \hline
 -54H
 \end{array}$$

7) \rightarrow

$$\begin{array}{r}
 1101\ 1101 \\
 1100\ 1111 \\
 \hline
 \textcircled{1} \ 1010\ 1100 \\
 \text{CF} \quad \downarrow \\
 \text{AC}
 \end{array}$$

S = 1

Z = 0

AC > 1

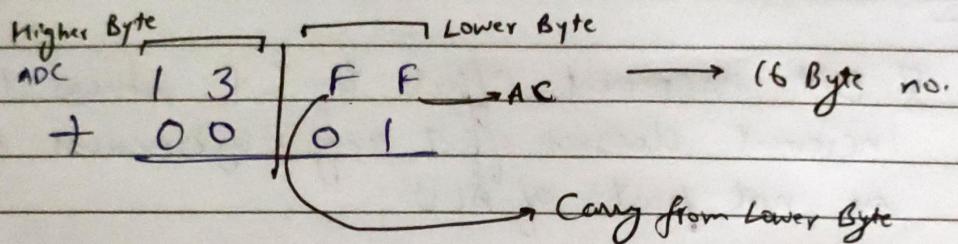
(F = 1)

P = 1

7) $\overset{\leftarrow \text{Add with carry}}{\text{ADC R}}$

$$A \leftarrow A + R + CF$$

* It is used to add two 16 Bit numbers.



8) ADC M \rightarrow 1 byte

$$A \leftarrow A + [M] + CF$$

9) $\overset{\leftarrow 2 \text{ Byte inst?}}{\text{ACI } 25H}$

$$A \leftarrow A + 25H + CF$$

10) $\overset{\leftarrow \text{Subtract with Borrow}}{\text{SBB M}}$ $\overset{\leftarrow \text{Carry in 2's complement system}}{\text{}}$

$$A \leftarrow A - [M] - CF$$

11) SBI 2SH
 $A \leftarrow A - 2SH - CF$

12) SBB R
 $A \leftarrow A - R - CF$

13) INR R $\rightarrow 8\text{bit}$
 $R \leftarrow R + 1$

14) INX Rp $\xrightarrow{\text{Register Pair}} 16\text{bit}$
 $R_p \leftarrow R_p + 1$

$$\begin{array}{r}
 FF \\
 + 1 \\
 \hline
 100
 \end{array}$$

Note -

* 8 bit increment effects the C.F whereas 16 Bit increment doesn't effect any flag because registers are not part of ALU.

15) INR M

\curvearrowright Decrement

16) DCR R

17) DCX Rp

18) DCR M

\curvearrowright Double Addition

19) DAD D

HL is acting as a accumulator

HL $\leftarrow HL + DE$

DAD B

HL $\leftarrow HL + BC$

Page No.	
Date	

→ Multiplication by factor 2
 ADD M
 $M_L \leftarrow M_L + M_L$

20) DAA
 ↳ Decimal Adjust after addition

$$\begin{array}{r}
 \oplus \\
 49H \\
 + 39H \\
 \hline
 82H
 \end{array}
 \quad
 \begin{array}{r}
 45 \\
 + 45 \\
 \hline
 8A
 \end{array}$$

(i) $24H$

$$\begin{array}{r}
 25H \\
 - 49H \\
 \hline
 \end{array}$$

(ii) 2.5

$$\begin{array}{r}
 25 \\
 - 50 \\
 \hline
 4A
 \end{array}$$

in BCD we have to
add 6

in BCD $\rightarrow 0-9 \rightarrow^{10} \rightarrow$ Add 6

in Hexadecimal $\rightarrow 0-15 \rightarrow^{16}$

$$\begin{array}{r}
 0010 \ 0101 \\
 0010 \ 0101 \\
 \hline
 0100 \ 1010
 \end{array}$$

↓ ↓ ↓

Higher lower

$10 \geq 9$ Nibble Nibble
 BCD if $HN > 9$ if $LN \geq 9$
 addition or or
 9+6 AC=1

$$\begin{array}{r}
 0000 \ 0110 \\
 \hline
 0101 \ 0000
 \end{array}$$

\sum

Add 6 or
0110