Control Bus / lines

used to control access & the use of data & address lines.

I/O, Read/write, memory read/write, Interrupt request, Interrupt acknowledge

↓

signals hard

Automated Calculator

Applications of µp

comp that uses µp as CPU

① As microcomputer

② scientific & engineering research

③ Industry (data monitoring system)

④ security system

⑤ Robotics

⑥ Military application system

Architecture of µp based system

① Von-Neuman Architecture / machine (stored program concept)

② Harvard Architecture

① von-Neuman Archi

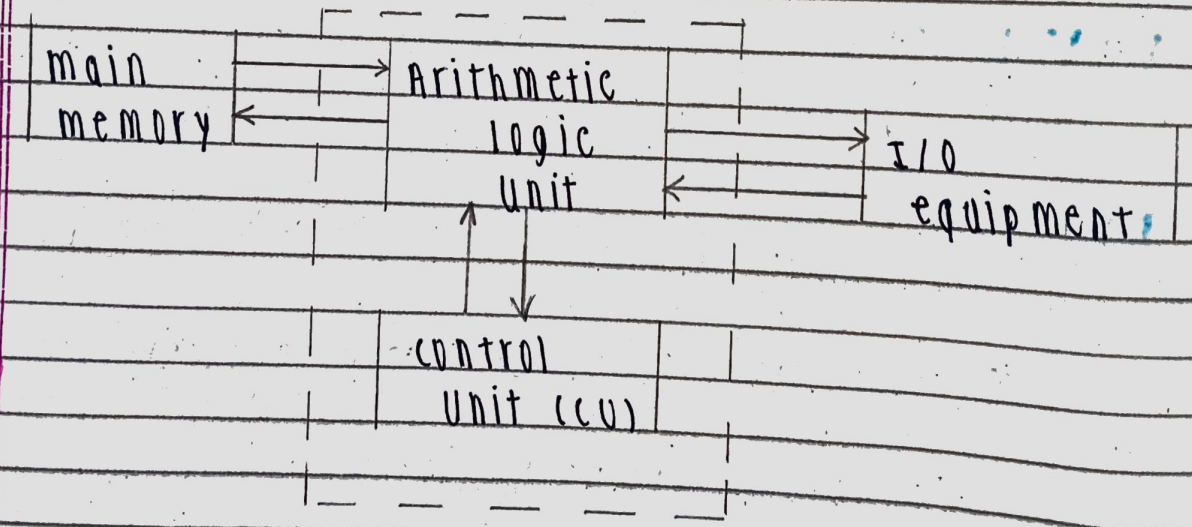


Fig: von-Neuman machine

- Basic Architecture of today's digital computers.
- main memory stores both data & instructions.

Registers used

1) MBR (memory Buffer Registrer)

2) MARC (memory Address Register)

3) IR (Instruction Register)

4) IBR (Instruction Buffer Register)

5) PC (program counter)

6) Accumulator (A) & multiplier quotient (MQ)

Halvard DIY

Diff betn both

DIY

History of computer

- 1st digital computer : ENIAC
  - Electronic Numerical Integrator And Calculator

DIY

Evolution of µp

	Address BUS	Data BUS
4004 µp → 1971 AD	10 bit	4 bit
⋮		
8085 µp → 1976 AD	16 bit	20 bit
8086 µp → 1978 AD	8 bit	16 bit
⋮		
Pentium I		
II		
III		
IV		

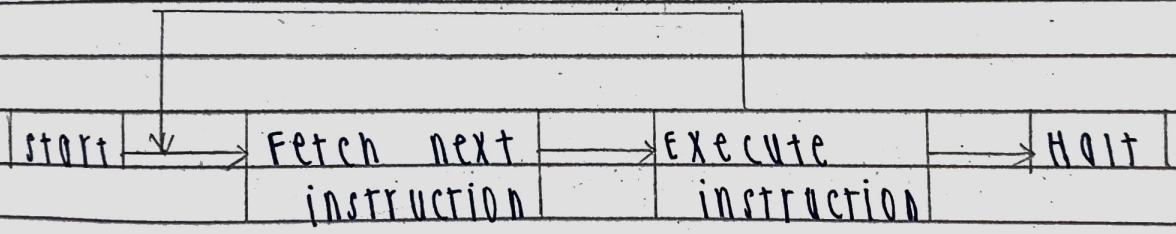
control Unit

① Hardwired CU

② microprogrammed CU

diff betn

processing cycle of stored program computer



Instruction cycle

↓

Fetch cycle + Execute cycle

Processor performs

- Fetch inst^n
- Interpret inst^n
- Fetch data
- Interpret data
- Process data
- write data

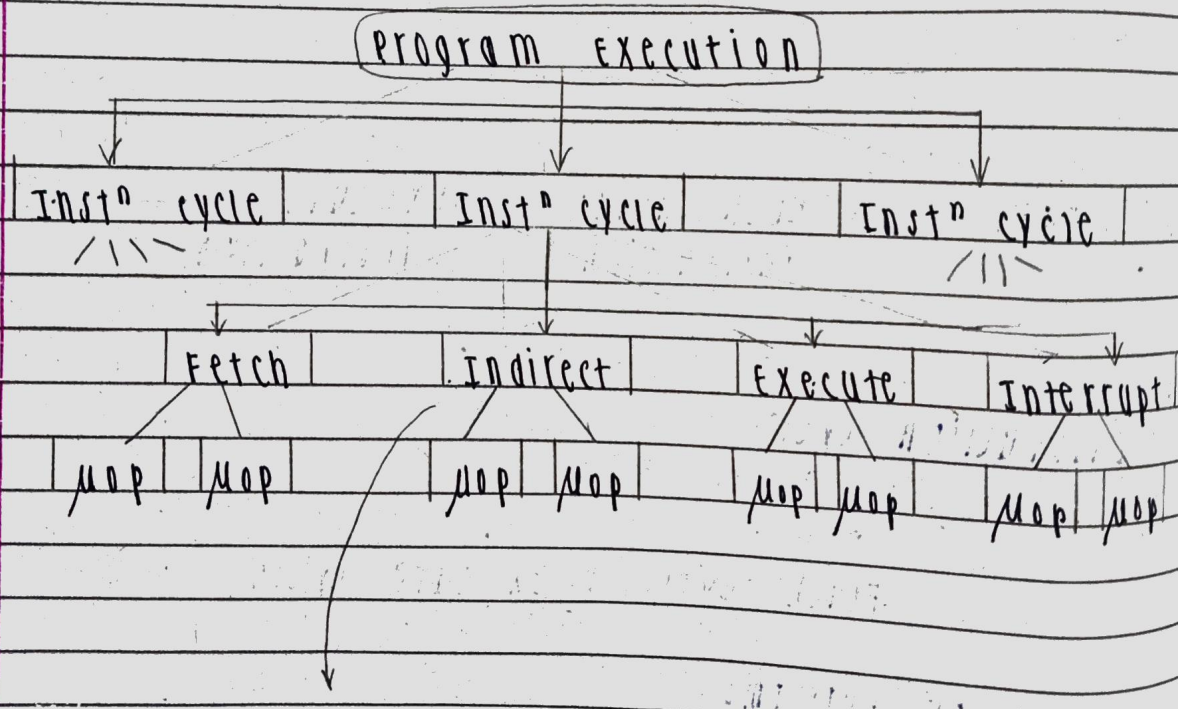
(μop)

micro operations

↳ Each inst^n is made up of shorter subcycles

like fetch, execute, interrupt etc.

known as micro operations.



Data not stored in register are thru data line

Accumulator based µp

Chap : 2

programming with 8085 µp

Programmer's model of 8085 µp

A	8	Flag	8
B	8	C	8
D	8	E	8
H	8	L	8

Stack pointer (SP) 16

Program counter (PC) 16

Data BUS

8 lines

Bi-directional

Address BUS

16 lines

uni directional

1. Accumulator (A)
   - 8 bit register
     - accessible to programmer

2. Registers (B, C, D, E, H, L)
   - 6 general purpose 8 bit registers
   - pairing BC, DE, HL can be done w. 16 bit each.

3. Stack pointer (SP)
   - 16 bit register used as memory pointer
   - Holds address of TOS
     - (Top of stack)
   - Follows LIFO
     - (Last In First Out)

4. Program counter
   - 16 bit register
   - holds address of next inst^n to be executed.

8 marks mq

group of F.F.

0 (Reset)

1 (Set)

5. Flag Register
   - 8 bit register
   - 5 Flags in 8085

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	AC	X	P	X	CY

- CY → carry Flag
- P → parity Flag
- AC → Auxiliary carry Flag
- Z → zero Flag
- S → sign Flag
- X → don't care

D0 – D7
- 8 bit

D7, D6, D4, D2, D0
- 5 Flags

rest d.c.

ⓐ carry Flag
- set if carry occurs in last arithmetic / logical operations

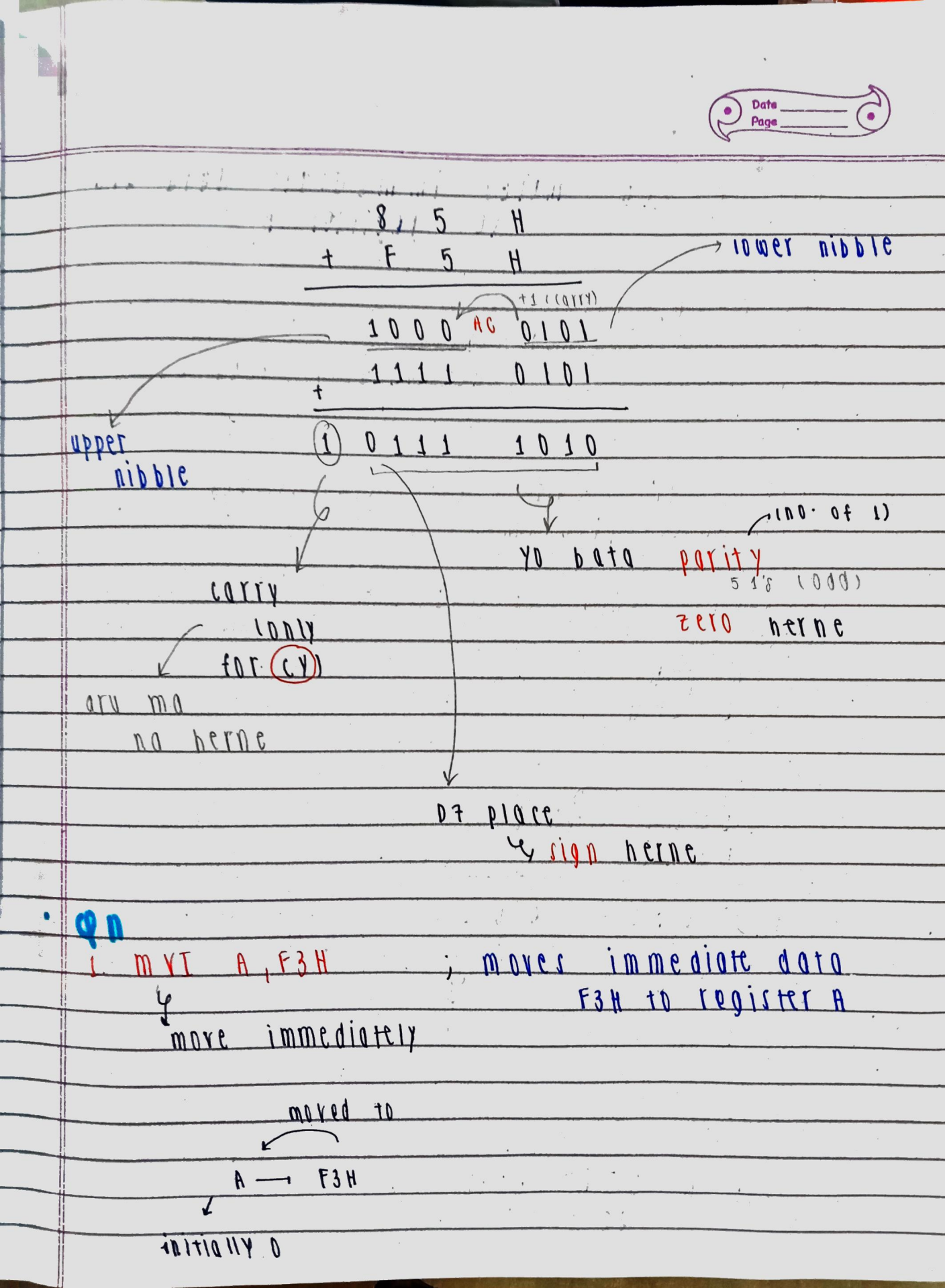
Date
Page

(b) Parity Flag
↳ even parity set ← even no. of 1
↓
else reset

(c) Auxiliary carry flag
↳ If there is carry from lower nibble to upper nibble it sets
↓
else resets

(d) Zero Flag
↳ If o/p is zero then set
↓
else reset

(e) Sign Flag
↳ If D7 bit is 1 (-ve) then set
↓
else reset


8 5 H
+ F 5 H
+1 (carry)
1000 AC 0101
lower nibble
1111 0101
+
upper nibble
1 0111 1010
carry only for (CY)
D7 place
sign herne
YO data parity
(no. of 1)
5 1's (odd)
zero herne
1. MVI A, F3H ; moves immediate data F3H to register A
move immediately
moved to
A → F3H
initially 0

MVI B, 2FH moves immediate data 2F to register B

ADD B ; A ← A+B
↳ Add vayera accumulator mei basxa

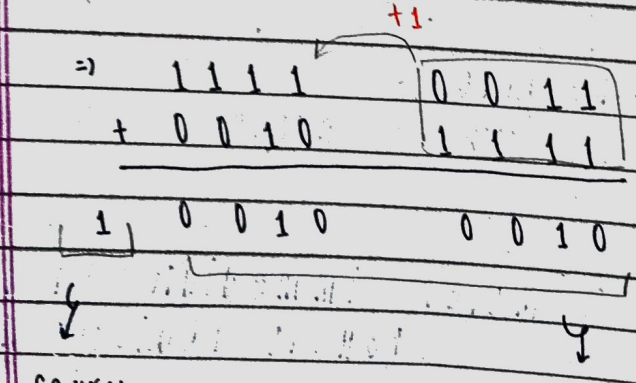
HLT

What is the content of accumulator after execution of program?

Determine status of flags ↓ ani explain gara

Soln

A = F3H
B = 2FH



∴ o/p A = 0010 0010
= 22 H

CY = 1

P = 1 (TWO OR OF 1's)

AC = 1 (lower nibble $\xrightarrow{\text{carry}}$ upper nibble)

Z = 0

S = 0
↳ (D7 = 0)