* Von Neuman Architecture.

He predicted the computers concept, where before transistors are exist, where instruction data and programming data are stored in same memory.

Central processor unit

Input divice → Control unit

Arthmetic logic unit → output.

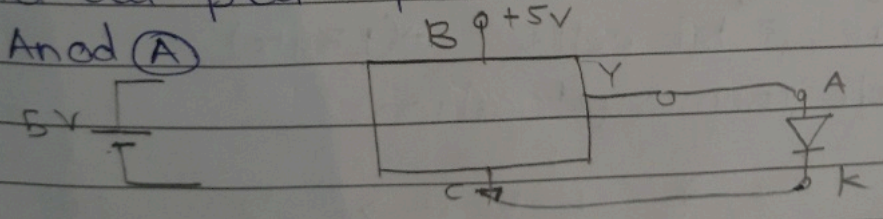
↓ ↑

Memory unit

* ~~In lat.~~
* Microprocessors.

In late 60's ~~mi~~discovery of microprocessor changed entire change in techology.

* Logic 1 - H & logic 0 - L
- Let consider input voltage of +5v & and ground level and out-put :Y and LED bulb having cathode (K) & Anod (A)



- IF Anode & output connected and ground & cathode connected if bulb glows then output is High (1).
- IF LED dosen't glow, then shift Anode at +5 volt & cathode to Y, and then,

(1) if bulb glow the output low (0)

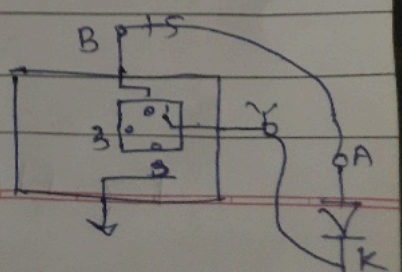
putting threeway switch.

at (Y) switch 1 - ~~to~~ glow

2 - glow

at

3 - Hanging

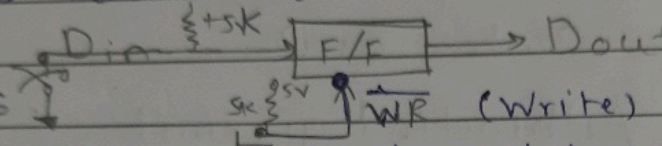


- outputs (High) / Low / tristate
  1 / 0 / Z
- Hanging output goes anywhere (connected)

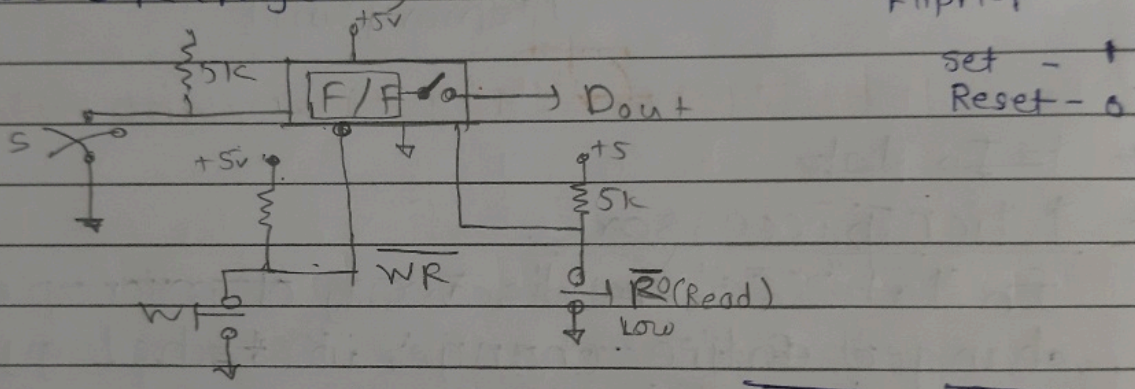
* Memory element :

- Single Bit memory cell :

it ~~is~~ has Din (Data in) & Dout (Data out)



Switch S is not close data putted in ①, the we give pulse WR (Negative).

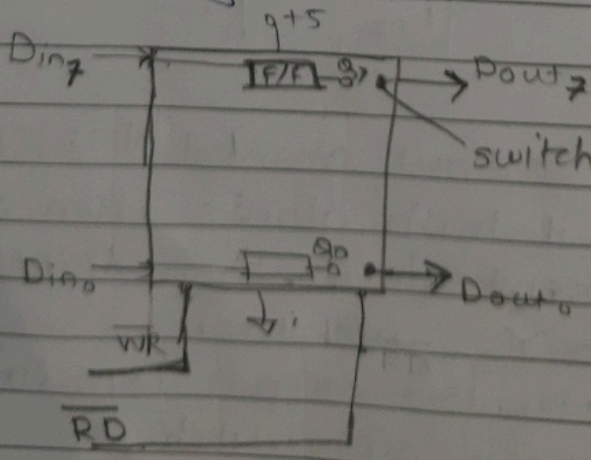


Flipflop
set - 1
Reset - 0

- It have four pins Din, Dout, $\overline{Write}$; $\overline{Read}$
  Using Din & $\overline{WR}$ ⟹ Store single bit
  ⟵ Dout & $\overline{RD}$ ⟹ Retrive ⟵ (Reads)

• Memory organization •

* There are 8 bit cells ⇒ (1 Byte)
- Single byte memory location.



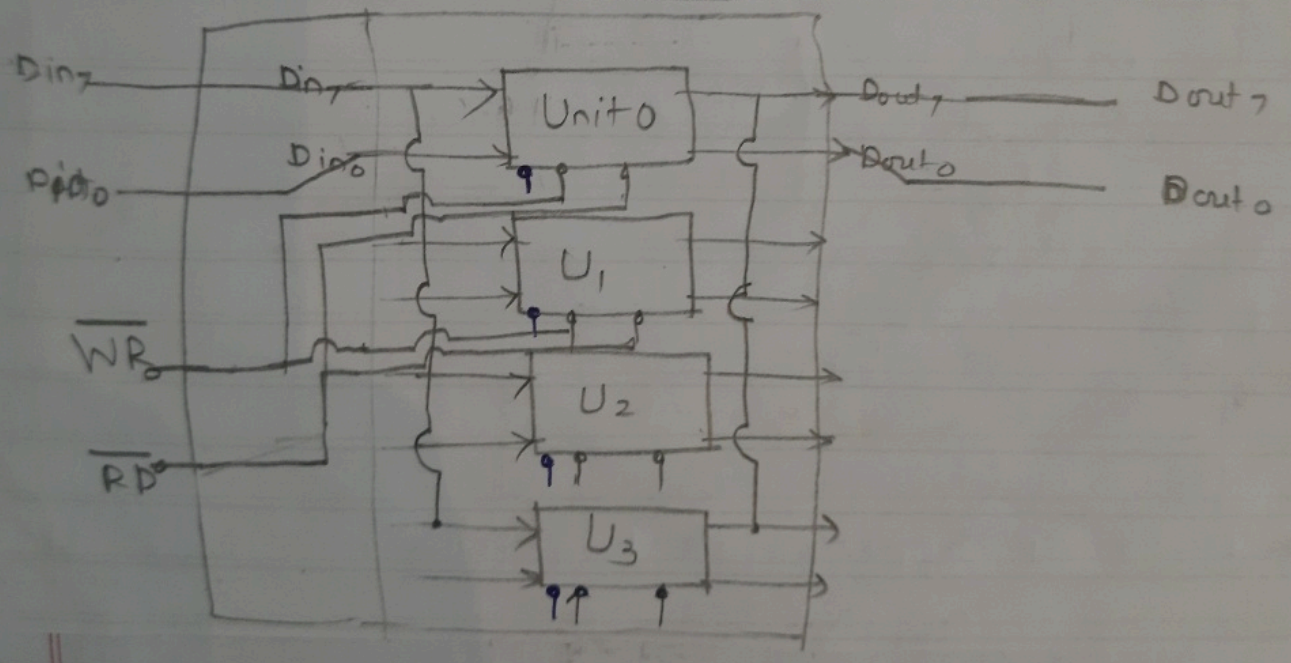
For single byte cells ($\overline{WR}$ & $\overline{IR}$)

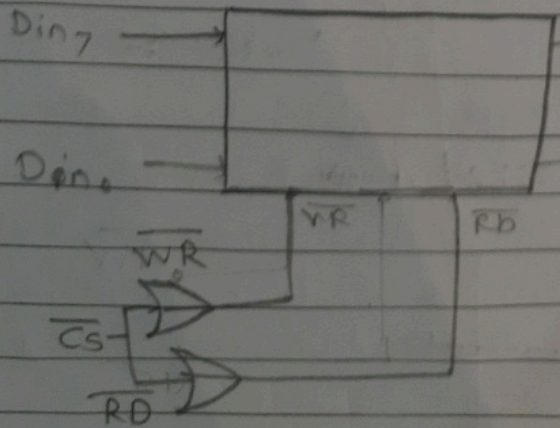
For 1 byte memory cell 8 $\overline{WR}$, & 8 $\overline{RD}$.

- Whatever is in input lines is stored in these 8 location (bits) where ~~rio~~ write bar made low and high again, at time of $\overline{WR}$ input becoming high, whatever is status in inlines is stored in flipflop inside the single bit memory cell at rising age of $\overline{WR}$ pulse.
- When $\overline{RD}$ made low switches ~~open~~ closed (appear
  $\overline{RD}$ made high "——" ~~is~~ open.
- (then output lines hanging lines)

* Construction of one 4 byte memory cell.

- Doctrine - Data inline is 8 and Dout = 8.



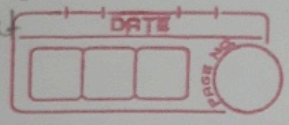


1 byte memory cell

$\overline{CS}$ =

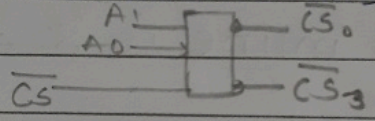
each unit have ground line

Shoul. each memo.unit & one overriding input



- For reading process
  $\overline{CS}$ (chip select) to 0 & $\overline{RD}$ is 0
- For writing process.
  $\overline{CS}$ is 0 & $\overline{WR}$ 0

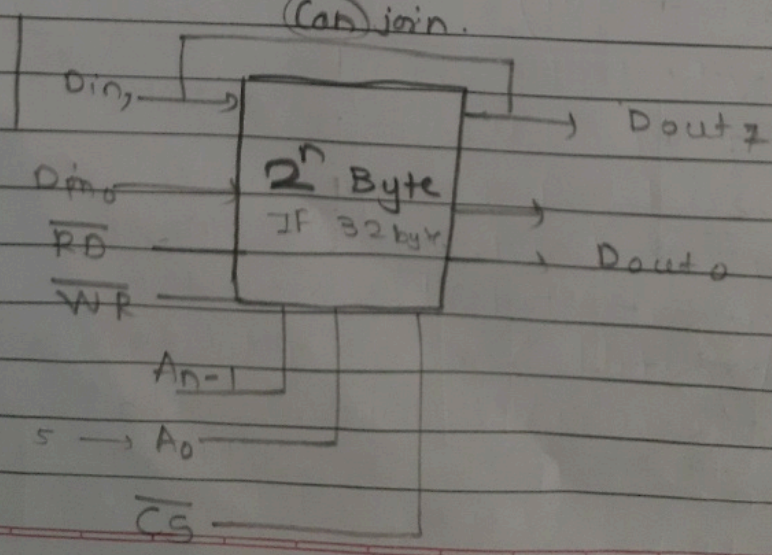
- Combine 8 Data input lines. ~~no~~ or. ~~A0 or 0~~
- " " " outlines.
- " $\overline{RD}$ of all units & $\overline{WR}$ of all units
  : no^r data outlines & inline in 4 byte memory cells = 8
- Each 1 byte memory cells 19 lines of each its. which combin & 4 remaining $\overline{CS}$.
- Only one $\overline{CS}$ (out of ~~4, 1, 3~~) should be 0.



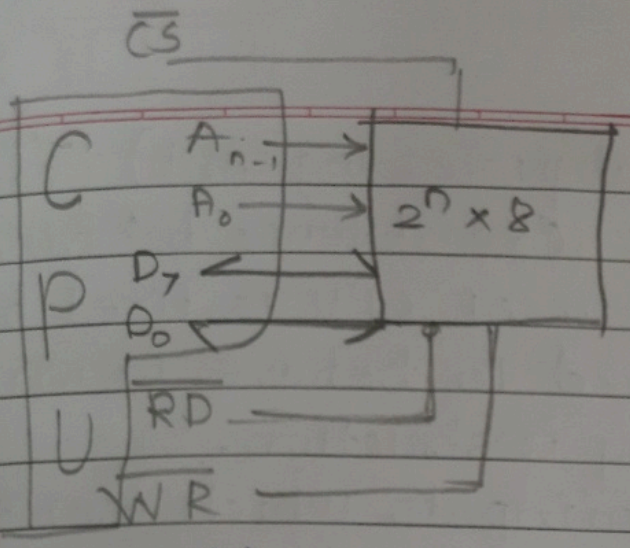
Choose from one of them or none of the units

$\overline{CS}$	A_1	A_0	$\overline{CS_0}$	$\overline{CS_1}$	$\overline{CS_2}$	$\overline{CS_3}$
	0	0	0	1	1	1
	0	1	1	0	1	1
	1	0	1	1	0	1
For none selection	1	1	1	1	1	0
1	X	X	X	X	X	X

- Data of appropriat location come or stored.



Construction of multiple Byte memory cell

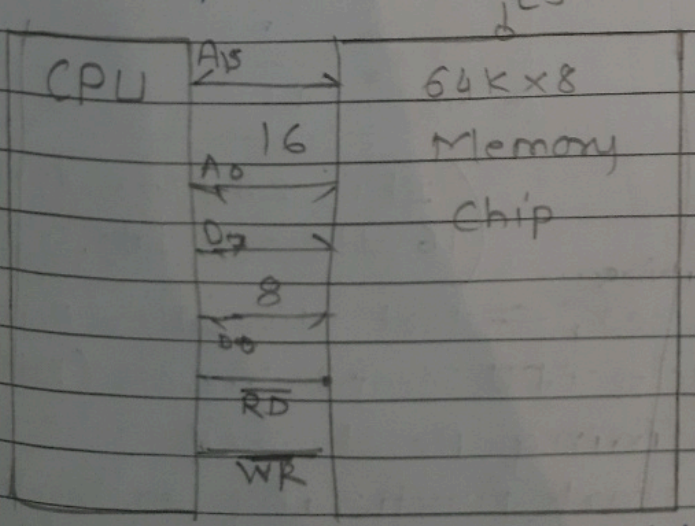


operration of multiple byte memory cell.

- Entity which wants to use these cell First decide which operation he wants to.
- For ~~r~~eading & WR ~~wh~~. it have to put address bars & locations (exact)
- Internally microprocessors have some single byte memory cells and arthmetic logics.

- Single byte memory cells in CPU are ~~resistors~~ called register.
- and program counter.

* Microprocessor (Microcontroller) Architechure. ~~Read~~



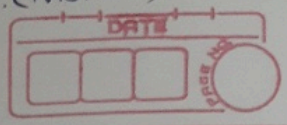
$2^{16} = 64K$

(Device given)

① 16 address line. — for writting must know location.

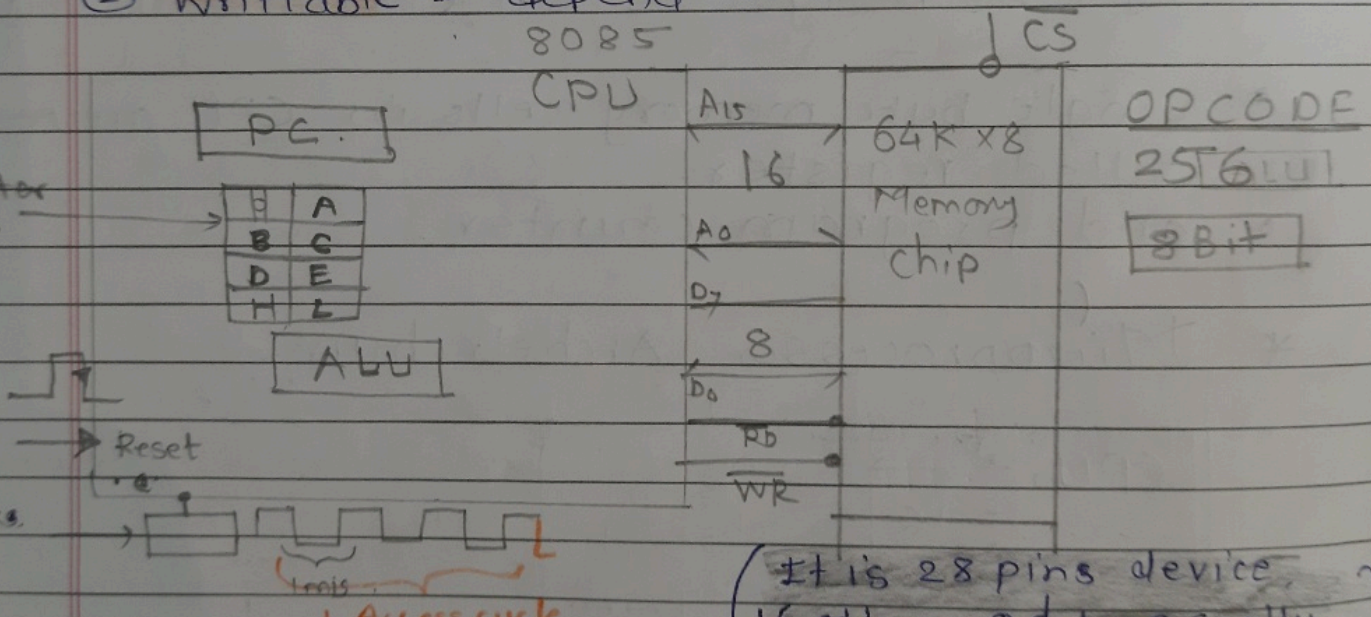
- 8 readline. —

only 1, 0, hanging. (tristate)



CPU (Central processing unit) execute instructions

- non-living entity. (consistence behaviour)
- The manner/behaviour of CPU is fixed but it can varied ~~sometime~~ ~~it is~~ forced to behave in manner as which we want by instruction.
- It operates on semi inputs which user suppose to give.
- In memory chip CPU can Read write in any location if there no power supply the data not stored.
- There is no concept of bank (same data)
- * In memory chip the sections
  1. Pre-written — not depend on power suply. (ROM)
  2. Writtable - depend on



(It is 28 pins device, 16 address, 8 data, 2 supply, 2 control lines)

- ~~Sequences~~ activities
- Clock produce impulses, — start activities (Requirement)
- In three clock cycles CPU comple procees (1 mili access)
- Make it start beigining Reset is (1)
  Reset 0 system start functioning on beside clock.
- CPU contain 8 or 16 bits internal memory Register location
- ALU carry out process (addition / substraction & compliment & incrimant / Decreament) (copyright (Data transfer)
- e.g adding ADD B.

(1 Bit) Carry + Accumilator = A + B.

e.g INRC - Use 8 bits of Register C by ①
DCR D - Use 1 —— " D 1 —— "

CMA

* OPCODE → (ADD C, INR D)

• Using 8 bit OPCOD — solve 256 encod.
Internal registers & ALU are coded by OPCODE
program : Sequence of instructions.
" —— " OPCODEs stored in memory

• After reset, CPU must read.
reset should make Programme counte (PC) zero.

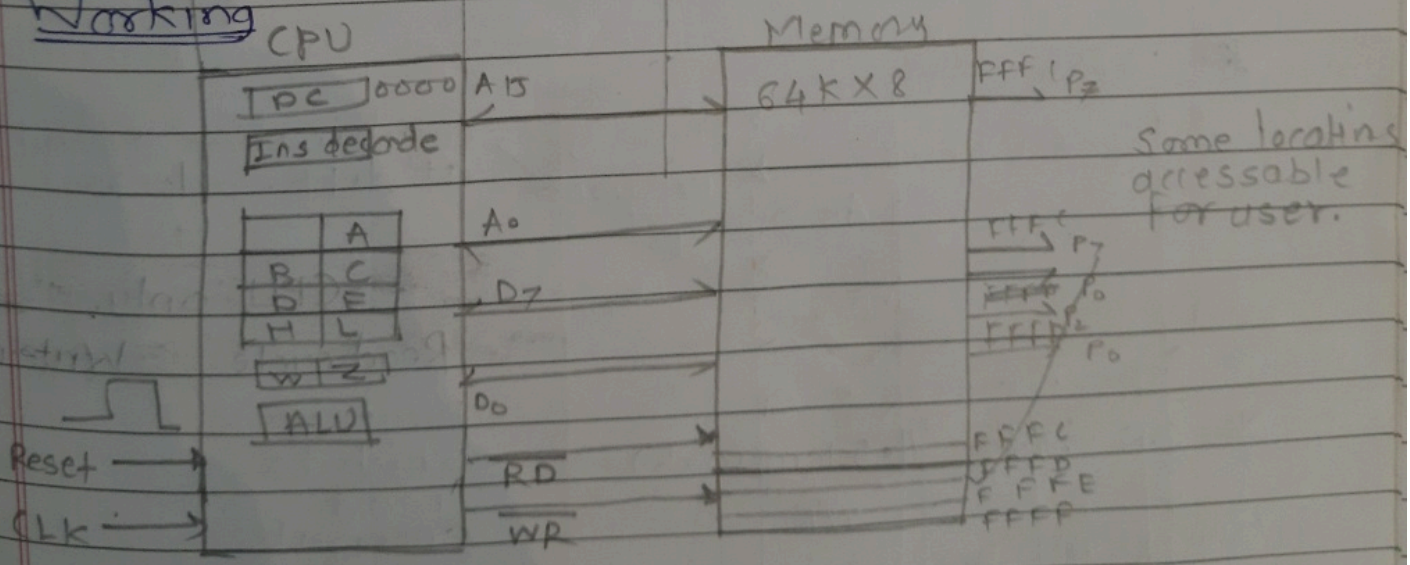
• first cycle after reset ⇒ Read location zero & treat as OPCODE start accordingly mp sequence.

* Data transfer.

e.g MOV C, B , Load Accu (LDA) address 4000 Hex

• 1 byte 2 digits (0, to 9) & A to F 8bit - 2digits 16bits=4.
Now STA 5000 ...

• Working



• Microprocessor reset then PC is 0000, then in 1st machine cycle (3 clk cycle) it would be read cycle the address line data lines Read lines 0 appears on data line.

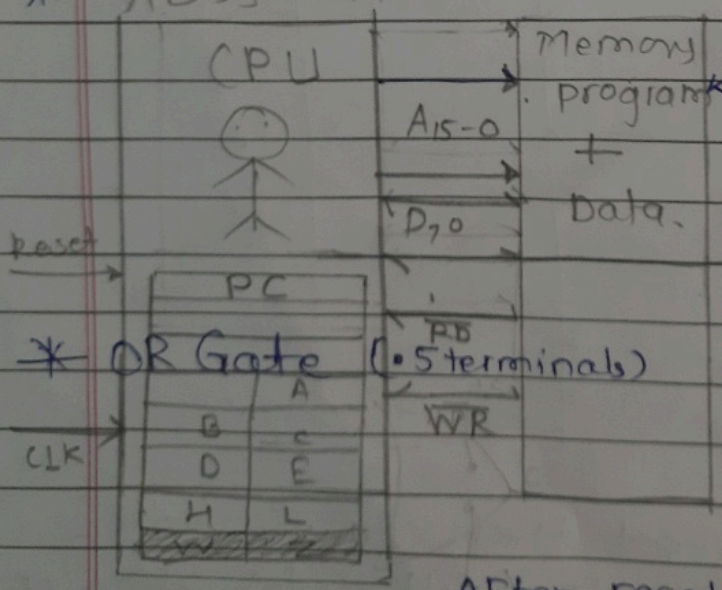
e.g 3A - OPCODE for LDA 4000 goes to instruction decoder.

JMP 6000
C3 0060

e.g.

	Type	Adress	Data, will go
1	Readcy (Fage)	PC → 0000 (give)	[ID] Instruction decode. For LDA 4000
again 2nd	R c.	PC → 0001	00 → register Z.
			00 - data read by 2nd cycle is corre.
			Lower byte of address from where data brought to accu.
3rd	R.c	PC → 0002	00 → W
	R.C	WZ → 4000	33 → A.
	R(F)	PC → 0003	32 → ID — For STA 5000
	R	PC → 0004	00 → Z
	R	PC → 0005	50 → W
Now.	WR cy.	WZ → 5000	A → 33
	R(F)	PC → 0005	C3 → ID
	R	PC → 0006	00 → Z WZ → PC
	R	PC → 0007	60 → W
	R(F)	PC → 6000	1

* AB system



combination of 8 bit location at diff. adress.

CPU only.

Read cycle Write cycle

location which to read CPU put. S. on RD

After reset.

Step. • Read. Prog. Counter Put PC (16 bit) on address lines. & Activate RD & Reads data. and puts it in / writes it on scratch pad.

First write cycle

* (First Hardware cycle should Fix)

eg. W.Z. lines ⇒ on address L⇒ on data lines.
But later it can't guide us.
∴ First cycle is always Read cycle. (Not WR)
→ If First cycle is WR cycle, since it is a first cycle address & data must be from fix entity after that what next is not given.
We don't get any new in CPU

* 1 Hex digit is of 4 bits.

e.g. Refering machine cycle encoder book, in it what is next machine cycle is found.

OPCODE →	00	01	02	--	3A	...	FF
1				--	1/2	- - - - -	
2							
3				--	3/4 (Go back)	- -	

e.g for 3A
LDA 5000

→ Instru. Read.

First cycle * ① $\overline{RD}$ — [PC — Address; Z ← Data] . * ② Read · [P → Add, W ← Data]
③ Read - [WZ → Add, A ← Data]
④ Read [PC → Add, ID ← Data] ⟹