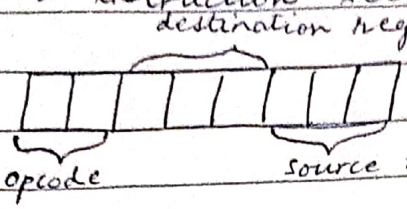
DESIGN AND OPCODE ASSIGNMENT

* 8-bit instruction set



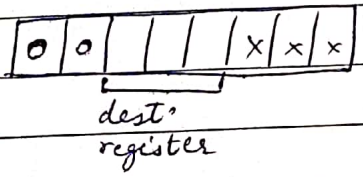
* 8 registers →

A	000	E	100
B	001	F	101
C	010	G	110
D	011	H	111

* opcodes :→

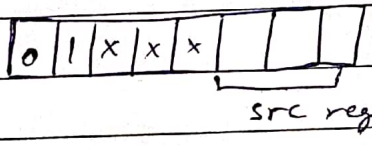
Inst.	OPCODE
IN	00
ADD	01
MOV	10
OUT	11

Ex:→ 1. IN →



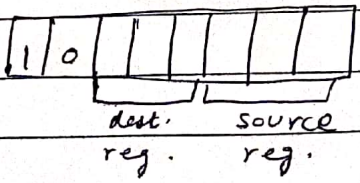
⇒ Mem [dest] <= in;

2. ADD →



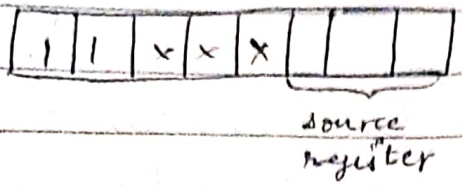
⇒ A ← A + Mem [src

3. MOV →



⇒ Mem [dest.] <= Mem[src

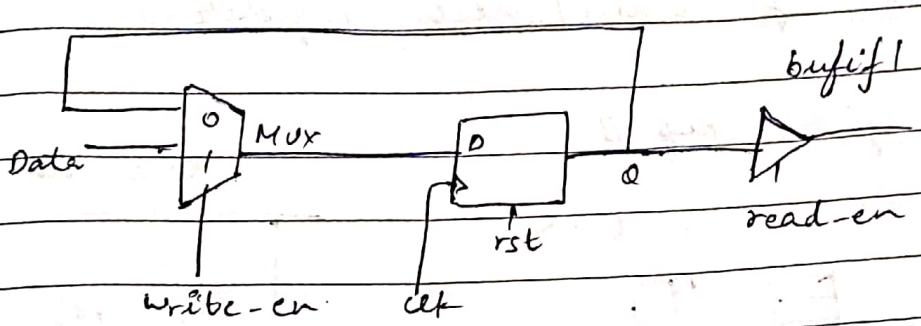
4. OUT



OUT <= Mem [src]

* Reg A => attached to adder. (Accumulator)

Structure for other registers (Control path)



read-en : 1 => data is available for reading.

write-en : 1 => data can be written.

Data path / data line :-

* Common for both i/p & o/p. [serving as bidirectional]

data_line <= in if instr. = IN.

OUT <= data_line if instr. = OUT.

For data transfer b/w registers, one reg is source & the other is destination register.

Logic:

→ The value of source register is written to data line & the value of data line would be ~~read~~ written into ~~into~~ ~~from~~ destination register.

Selection of source/dest. register:-

For eg:- Reg B