CPU Registers :

CPU requires a few registers to carry out its functions.

Type of Registers :

(1) MAR : Memory Address Register

→ Holds the address of memory where operand is stored.

→ Using this address the operand from the memory location can be retrived.

→ MAR size = 12 bits.

(2) Accumulator :

→ contains the initial data to be operated on intermediate data/result and final result of processing operation

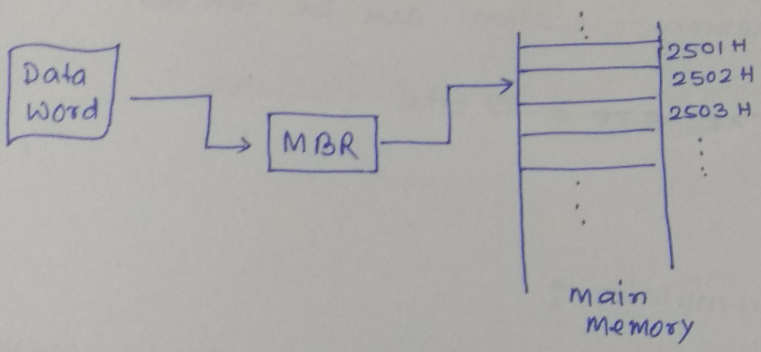
→ The result of arithmetic operation is stored in the Accumulator (ACC) and from Ac it is send to the memory storage thro MBR

③ MBR : Memory Buffer Register

→ It hold the content of memory word to be read from or write into the memory.

→ An instruction word placed in this register is transferred to the IR

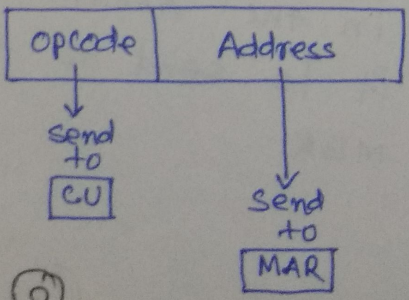
→ The word to be transferred to the Main Memory must through MBR.



IR : Instruction Register

→ It hold the current instruction that is being executed by CPU.

Instruction :



(5) PC : Program Counter

→ It hold the address of next instruction to be executed.

Assume that Instruction are stored in contiguous memory location and read and executed seqentially
Otherwise branch Instruction is encountered.

(6) I/O Register :

→ Used to communicate with I/O devices

→ All i/p information is transferred to this Register by an i/p device

→ All o/p information send to o/p devices is stored in it.

Case study : Lect-(03) / ISA / AJIT PAL / IIT-KGP

How to reduce Instruction length using some Spl. purpose registers.

Example : instruction ⇒ c = a+b;

Instn. Format →

opcode	Addr. of OP1	Addr. of OP2	Destn Address	Addr. of Next Instn
← 1B →	← 2B →	← 2B →	← 2B →	← 2B →

↑ Hex code = 8bit = 1B

If, word length = 2 Bytes.

then, Total length of Instn. = (1+2+2+2+2) = 9 Bytes / Per instn

If word length = 2 Byte then the Instruction (of size 9B) required 9/2 = 5 cc

→ Address of Operand = 2B to 4B.

So, It leads to ↳ ① Instn. size is long

② fetching of instn. is long

Alternative is specify memory address ~~implicitly~~ explicitly rather than explicitly.

↓

use of spl. purpose register
i.e PC

PC → contains the addr. of Next instruction

so, we need not required Addr. of Next Instn field in above.

Now,

IF ⇒

OPcode	Addr. of OP1	Addr. of OP2	Dest. Addr.
1B	2B	2B	2B

Total length = 7 B/ instn
required clock cycle = 7B/2B = 4cc.

Now, try to replace initial operand (ie OP1)

↓

store OP1 to ACC

so, need not require the Addr. of OP1 field.

Since, ACC can store the result of arithmetic operation by the CPU.

So, need not require the Dest. Addr. field.

Finally, IF →

Opcode	Addr. of OP2

⑤

⇒ size of Instn = (1+2) B
= 3B.

Req. clock cycle = 3/2 = 2