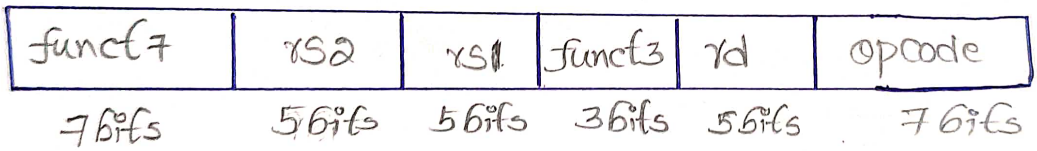
Instruction Type & Formats :-

① R type (Register) ADD rd rs1 rs2 $\Leftrightarrow$ rd $\leftarrow$ rs1 + rs2

Where rd – destination Register

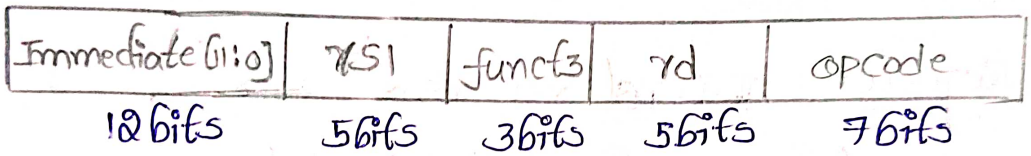
rs1, rs2 – source registers



funct7 & funct3 are additional opcodes.

② I-type (Immediate) load: Memory ⟶ Register

Ex: lw rd, Immediate(rs1) ⇒ rd = Mem[rs1 + Immediate]

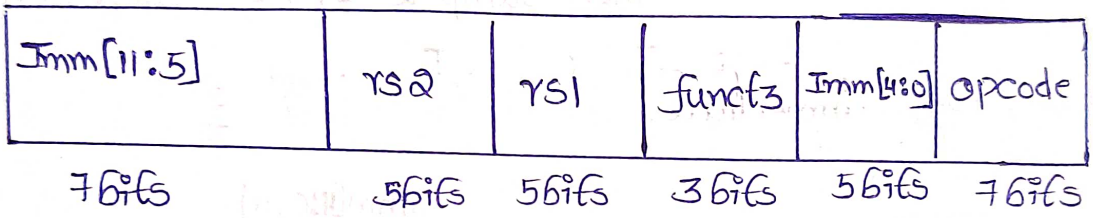


funct3 & opcode Helps to identify this instructions.

S-type (store)

store – Register → Memory

sw rs2, immediate (rs1)



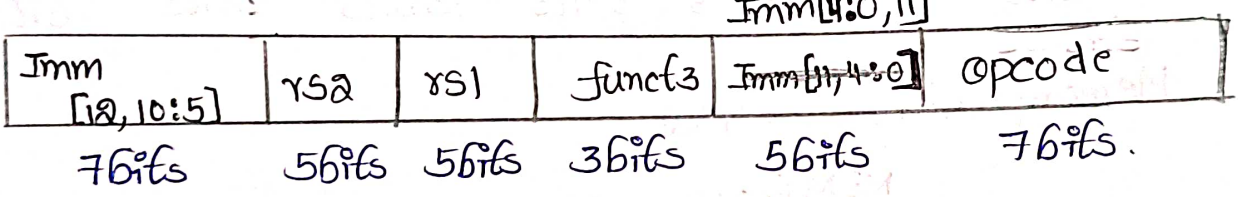
Memory address – rs2

MEM [rs2] = rs1 + Immediate

B Type (Branch type)

beq rs1 rs2 . L ⇒ if (rs1 == rs2)
then Jump to (L + PC)th Instruction

beq rs1, rs2, Immediate

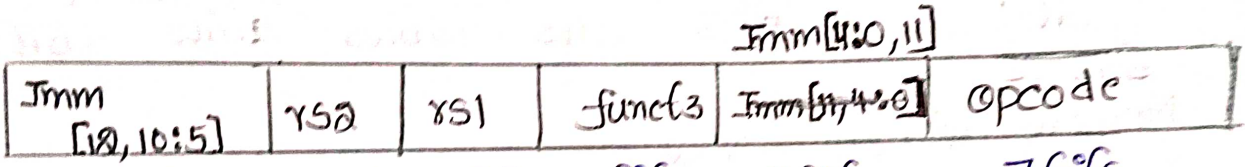


If rs1 ≠ rs2, then immediately
start executing (PC + Immediate)th Instruction.

B Type (Branch type)

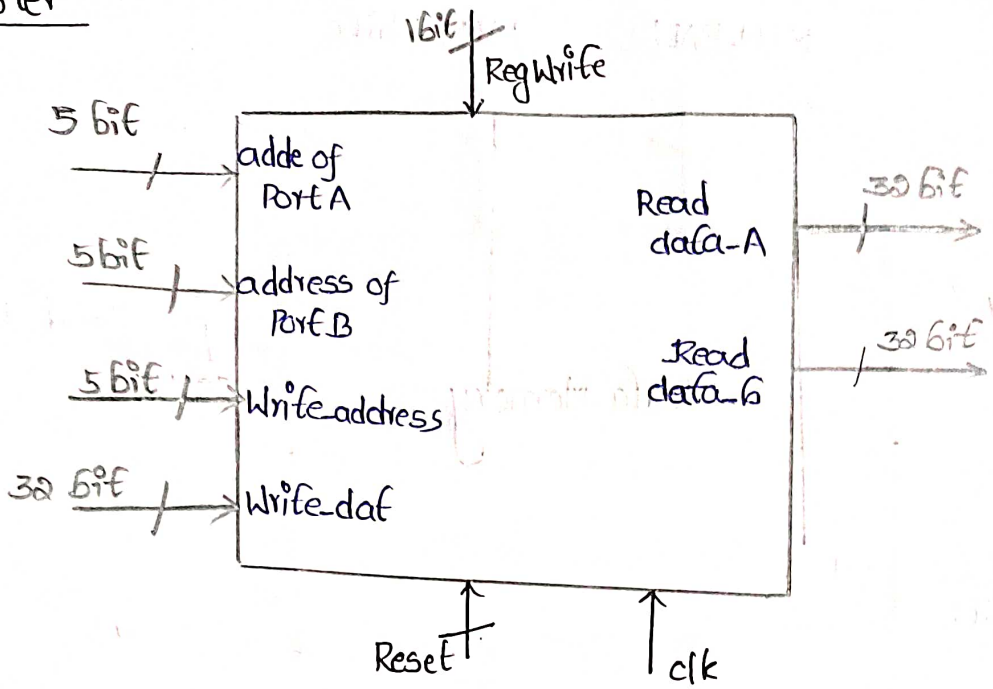
beq rs1 rs2 . L ⇒ if (rs1 == rs2)
then Jump to (L + PC)th Instruction

beq rs1, rs2, Immediate

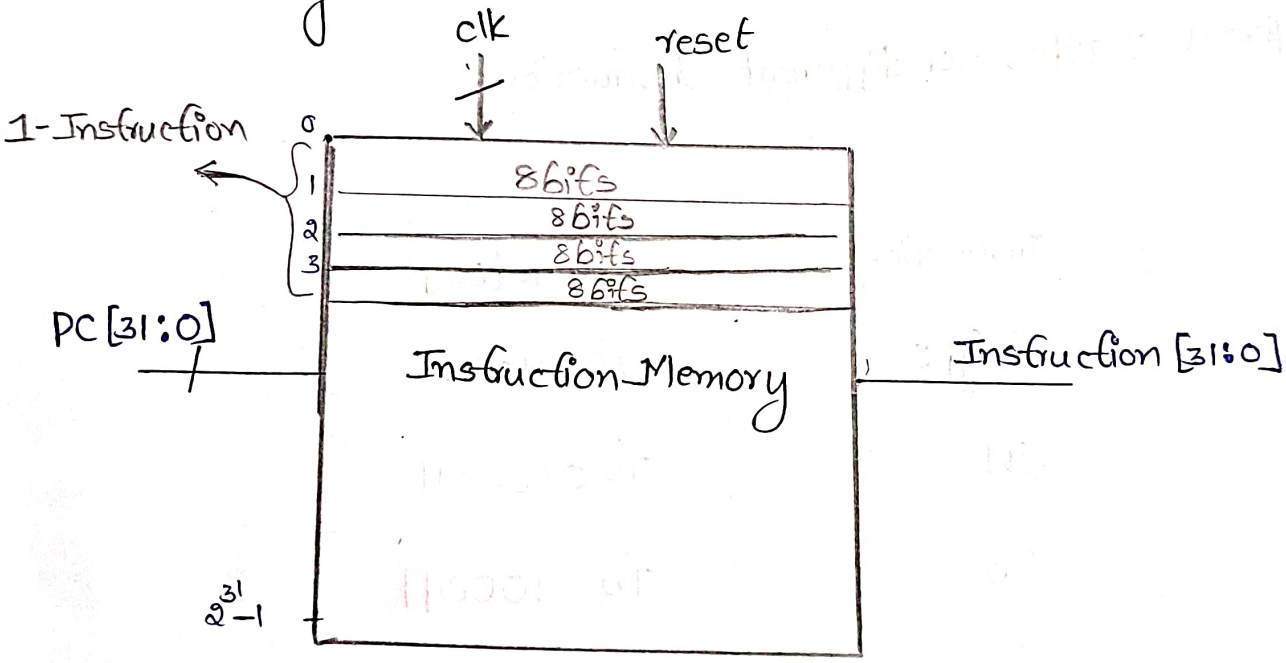


If rs1 ≠ rs2, then immediately
start executing (PC + Immediate)th Instruction.

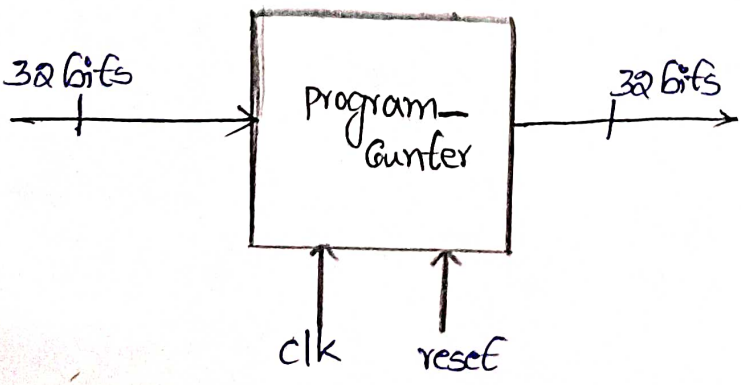
Register



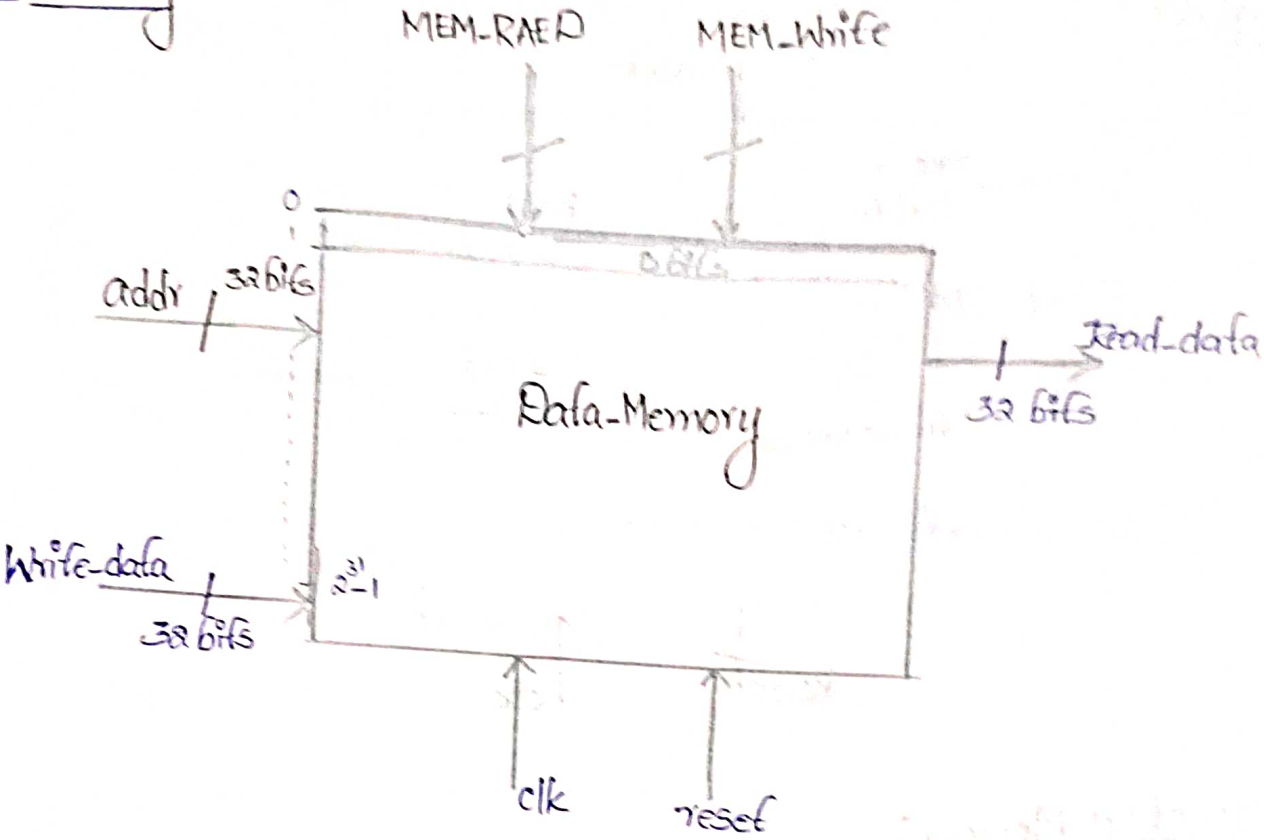
Instruction Memory :



Program Counter :



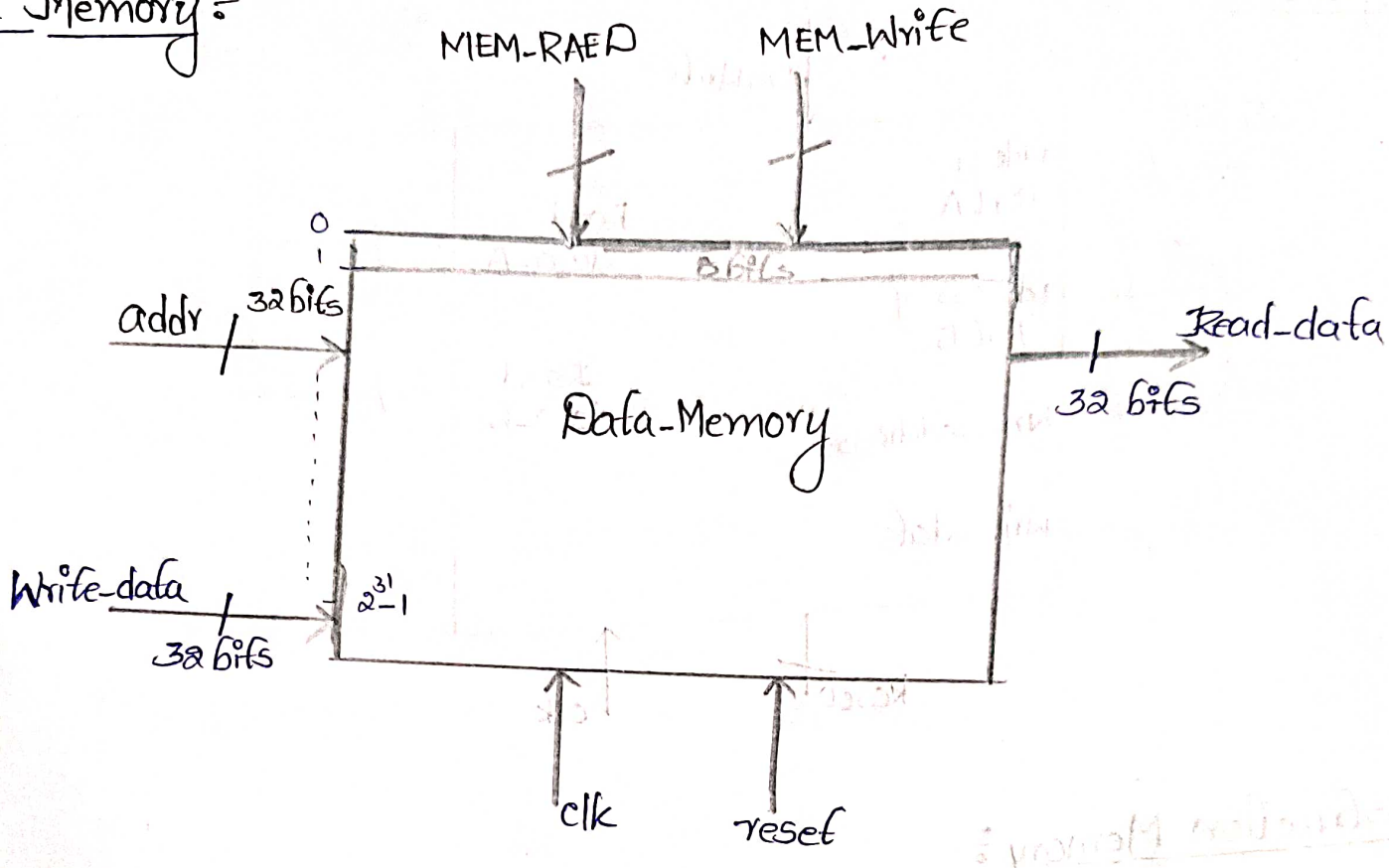
Data Memory :



Different opcodes for different Instructions :

Instruction	opcode [6:0]
R-type	7'b 0110011
SW	7'b 0000011
lw	7'b 0100011
beq	7'b 1100011

Data Memory :

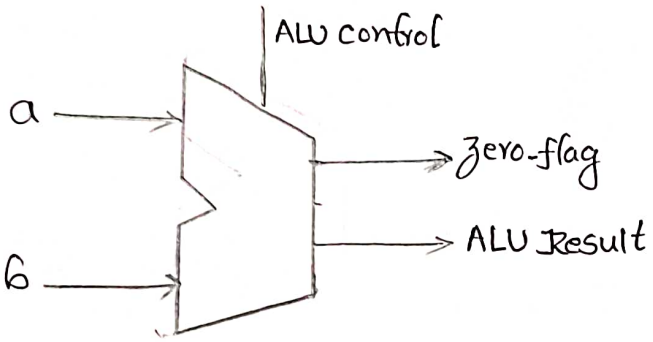


Different opcodes for different Instructions :

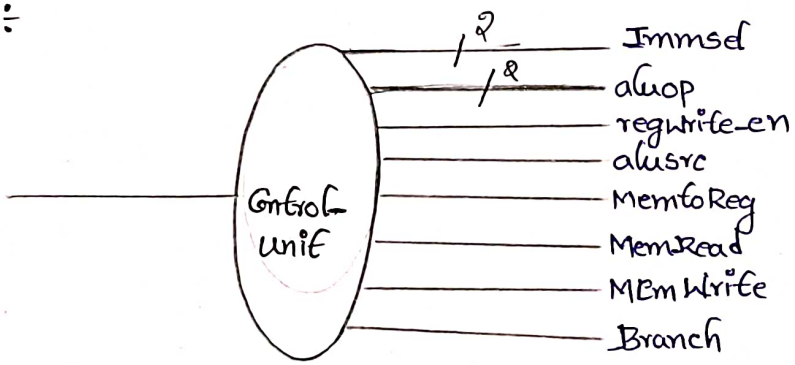
Instruction	opcode [6:0]
R-type	7'b 0110011
SW	7'b 0000011
lw	7'b 0100011
beq	7'b 1100011

ALU:

ALU control lines	Function
0000	AND
0001	OR
0010	ADD
0110	SUB



Control Unit:



Instruction	ALUSrc (0=rs2, 1=imm)	MEM to Reg	RegWrite	MEM Read	MEM write	Branch	ALUop
R-type	0	0	1	0	0	0	10
lw I-type	1	1	1	1	0	0	00
sw	1	x	0	0	1	0	00
beq	0	x	0	0	0	1	01

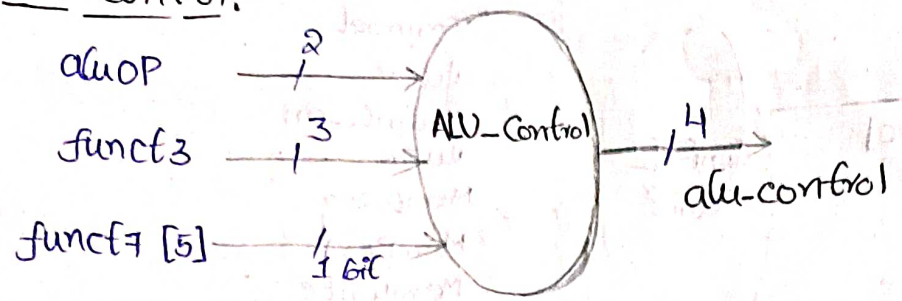
R-type Immsel = 2'bxx

lw — Immsel = 2'b00

store_operation ⇒ Immsel = 2'b01

Beq-operation ⇒ Immsel = 2'b10

alu_control:-



Instruction Opcode	ALUop	operation	Funct7 field	Funct3 Field	desired ALU action	ALU control input
lw	00	load	xxxxxxx	xxx	add	0010
sw	00	store	xxxxxxx	xxx	add	0010
beq	01	branch if equal	xxxxxxx	xxx	sub	0110
R-type	10	add	0000000	000	add	0010
R-type	10	sub	0100000	000	sub	0110
R-type	10	and	0000000	111	and	0000
R-type	10	or	0000000	110	OR	0001

Here, for lw, sw, beq

ALUop are different

So we can make decision on ALUop

But for R-type

ALUop was same

~~funct7 [5], field~~

{funct7 field [5], Funct3 field} - 0000 — add
1000 — sub
0111 — and
0110 — OR