Instruction Type & Formats:

(1) IR type (Register) ADD rd vs1 vs2 > rd - rs1+rs2
Where rd-destination Register
vs1, rs2 - Source registers

func(7	752	751	Junct3	rd	opcode	
7 6°Fs	56%	567fs	36ifs	56765	769Es	

June 7 & June 13 are additional opcodes.

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

&: lw rd, Immediate (rs) -> rd = Mem[75+ Immediate]

Immediate (i) of 751 functs rd opcode 12 bits 56its 36its 56its 76its

Junct3 & opcode Helps to identify this instructions.

as-type (ostore)

offere - Register -> Mernory

SW rsa, immediate (rs1)

Imm[11:5]	rsa	751	Junc 13	Imm[4:0]	apcode
766	56its	56ifs	3676	56i6	767£s

Memory address - 752 MEM [rsa] = 751 + Immediate

B Type (branch type)

beg ys1 ysa. L ⇒ if (ys1==ysa)

then Jump to (L+ PC) th Instruction

beg 751, 752, Immediate

13.5	.5			Imm[4:0,1]	. Also
Imm [ia.io:5]	Y5 a	751	Junct3	Imm (174:0]	opcode
76its	56°6s	5676s	36its	5676s	7696s.

If 751 + 750, then immediately
Oblast executing (PC+ Immediate) the Instruction.

B Type (branch type)

beg rs1 rsa. L ⇒ if (rs1==rsa)

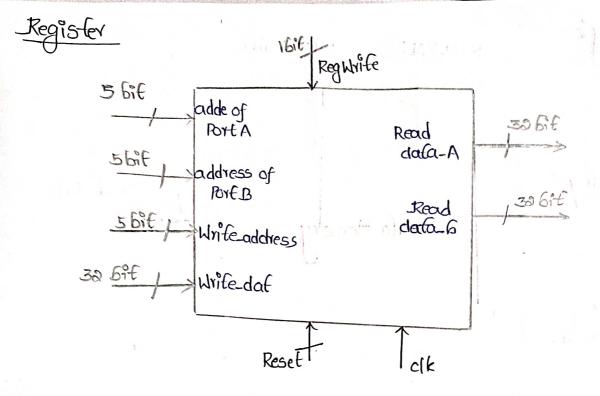
then Jump to (L+ PC) th Instruction

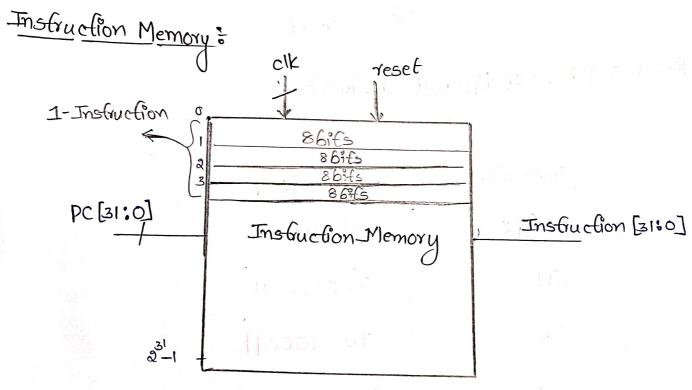
beg 751, 752, Immediate

3421				Inm[4:0,1]	100
Jmm	750	४ ८)	funct3	Imm [1740]	opcode
7696s	56965	56i6s	36765	5676s	7676s.

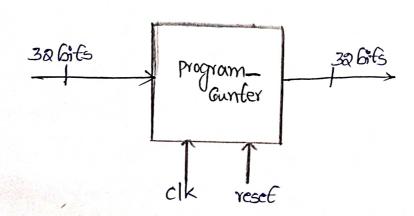
If 851 = 850, then immediately

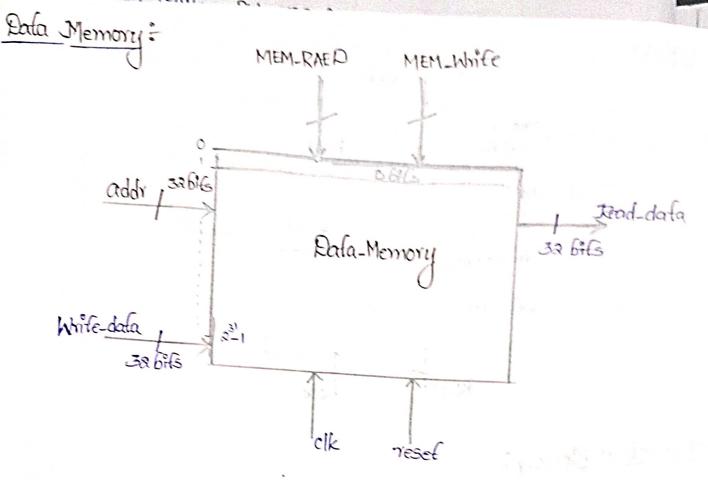
Alast executing (PC+ Immediate) the Instruction.





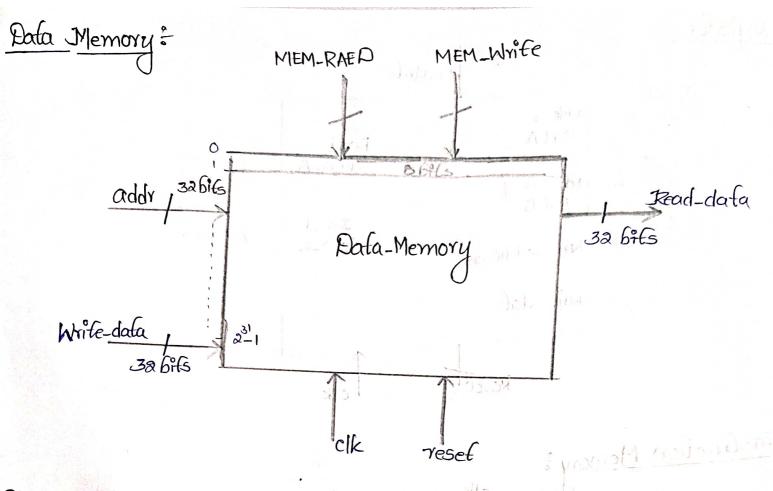
Program Cunter:





Different opcodes for different Instructions:

Instruction	apande [6:0]		
R-type	760110011		
SW	1600000 dF		
lw	1100010 87		
Бед	76 1100011		



Different opcodes for different Instructions;

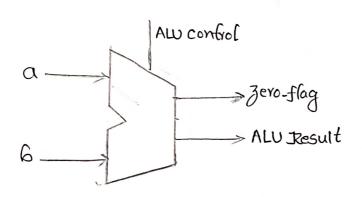
Lotte

Instruction	opcode [6:0]
R-type	100M760110011L
SW	1100000 3F
lw	1100010 8F
beg	76 1100011

[0:10] 9

Colored Will

ALU Control Pines	Function
0000	AND
0001	OR
0010	AAA
0110	SUB

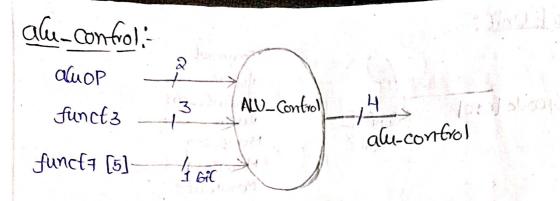


_			
Control Unit:		12_	Immsel
		/ / &	- aluop
		-	— regunite_en — alusto
_	 Gretro		— MemtoReg
	Gregol- unit	<i></i>	— Memikead — MEm Wrife
			— MEm Wrife
,	\ <i></i>		-Branch

Instruction	ALUSTO	MEM 60 Reg	Regulrite	MEM Read	MEM	Branch	ALUOP
R-type	0	0	1	0	0	0	10
luI-Gpe	1	1	1		\Diamond	0	00
SW	1	x	0	0	١	0	00
Бед	0,	×	0	0	0		01

afore-operation => Immsel- 8601

Beg-operation > Immsel = 0610



	Marin Ast.		110		No. of Contract of
AWOP	operation	Funct 7	Functs Field	deshed ALU action	ALU Confol mput
00	load	XXXXXXX	XXX	966	0010
00	Store	XXXXXX	XXX	add	0010
01	Granch if	XXXXXXX	xxx	Sub	0110
10	0.49	0000000	000	add oil	010
10	Sub	0100000	000	Sub	0110
10	and	000000	John John	and to	0000
10	or	0000000	110	oR	0001
	Alvop 00 00 01 10	Alvop operation OO load OO Store OI Granch if equal IO add IO sub IO and	Allop operation Funct 7 field OD load XXXXXXX OD Store XXXXXXX OI Granch if equal XXXXXXX IO add 0000000 IO sub 0100000 IO and 0000000	Allop operation Funct 7 Funct 3 Field Field OD load XXXXXXX XXX OD Store XXXXXXX XXX OI Granch if equal XXXXXXX XXX 10 add 0000000 000 10 sub 0100000 000 10 and 0000000 1111	Allop operation funct 7 Funct 3 desired Alloction OD load XXXXXXX XXX Add OD Store XXXXXXX XXX Add OI Granch if XXXXXXX XXX Sub 10 add 0000000 000 add 10 sub 0100000 000 Sub 10 and 0000000 1111 and 110

Here, for lw, Sw, beg-

ALUOP oure different

So we can make decision on Alvop

But for R-type

Awop was Same

funct 7 [5], fretd