

CPU Design

R-format (All arithmetic and shift instructions)

Op	Destination Reg	Source Reg	Shamt	Fn
6 bits	5 bits	5 bits	10 bits	6 bits

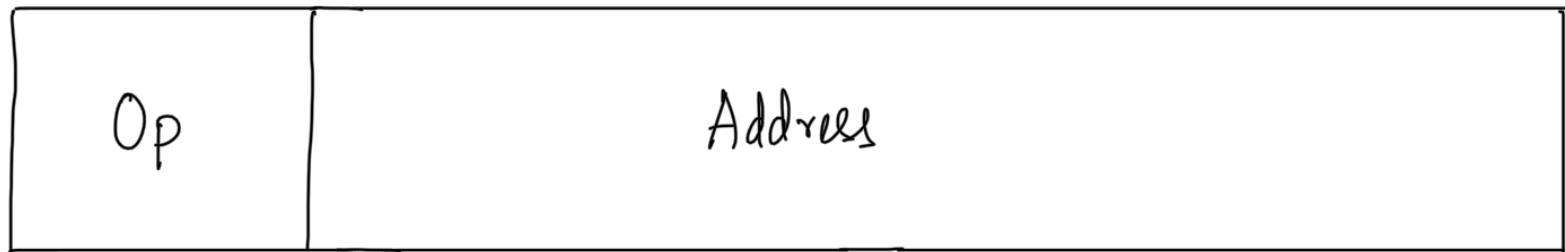
I-format

Op	Destination Reg	Source Reg	Immediate / offset
6 bits	5 bits	5 bits	16 bits

PC relative B-format

Op	Register	Address
6 bits	5 bits	21 bits

Abs B-format (Absolute address)



6 bits

26 bits

Encoding

Instruction

Op code

Fn

Add (R)

0

1

Comp (R)

0

2

And (R)

0

3

XOR (R)

0

4

Diff (R)

0

5

Shift left logical (R)

0

6

Shift right logical (R)

0

7

Shift into logical variable (R)

0

8

0011001000000000

Shift right logical (register) (R)	0	9
Shift right arithmetic (R)	0	10
Shift right arithmetic variable (R)	0	11
Unconditional branch (Abs B)	15	-
Branch register (PC rel B)	16	-
Branch on less than 0 (PC rel B)	17	-
Branch on flag 0 (PC rel B)	18	-
Branch on flag not 0 (PC rel B)	19	-
Branch and link (Abs B)	20	-
Branch on carry (Abs B)	21	-
Branch on no carry (Abs B)	22	-
Add Immediate (I)	63	-

Complement Immediate (I)	62	-
Load Word (I)	61	-
Store Word (I)	60	-
End	45	-

- * R format encoded instructions have been op codes 0 and 1
- * Op code 1 is used for shift related instructions.
- * Branch and immediate have separate formats. This is because in given ISA all branch operations use only one register (at most).
- * All immediate instructions have been given different encodings as there is no fn part.
- * The Branch instructions op code starts from 15. We have left a few op codes if any new R format instructions will be added later with

new op codes .

* We have two formats for branch instructions.

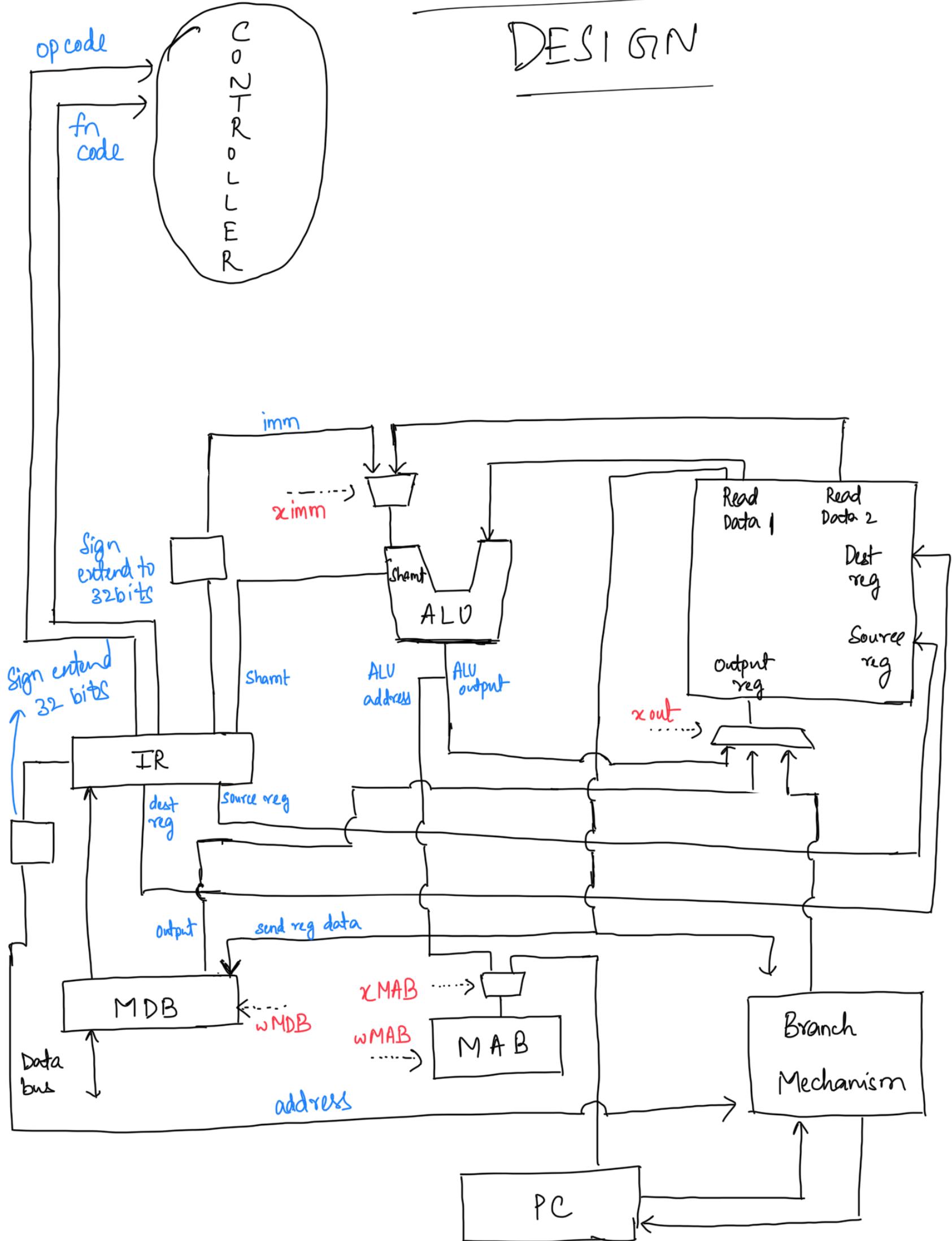
One which has space for a register (PC rel B format).

other works for absolute address .

Branch formats dont have fn codes as more space is required for fitting the address .

This separation is done so that branch instructions which dont involve register can utilize the space for address .

INITIAL DESIGN



Instruction Fetch

Instruction Decode

MAB \leftarrow PC

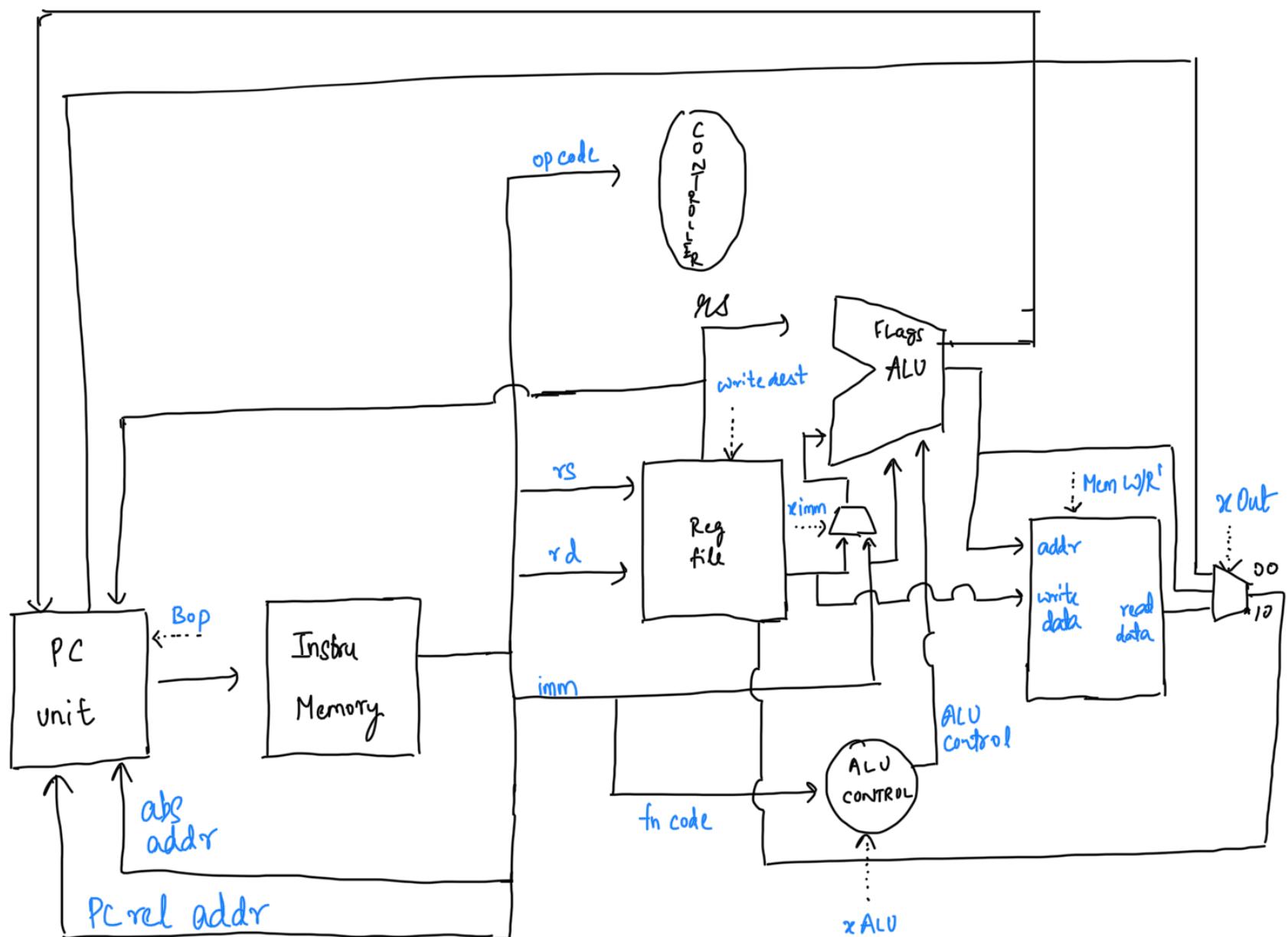
Examine op code

MDB \leftarrow Mem

and set ALU op code
accordingly.

IR \leftarrow MDB

FINAL DESIGN



<u>op code</u>	<u>Bop</u>	<u>xOut</u>	<u>ximm</u>	<u>xALU</u>	<u>writedest</u>	<u>Mem w/R</u>
0	111	01	0	001	01	0
15	000	01	0	000	00	0
16	001	01	0	000	00	0
17	010	01	0	000	00	0
18	011	01	0	000	00	0
19	100	01	0	000	00	0
20	000	00	0	000	11	0
21	101	01	0	000	00	0
22	110	01	0	000	00	0
63	111	01	1	010	01	0
62	111	01	1	011	01	0
61	111	10xx	1	010	1000	0
60	111	01	1	010	00	1
45	111	10	X	XXX	10	0

xpc : 00 \Rightarrow No branch

01 \Rightarrow Branch but no use of flags

10 \Rightarrow Branch using flag

11 \Rightarrow addr coming from register

$xOut$: 00 \Rightarrow write to reg using PC

01 \Rightarrow write to reg from RTU

10 \Rightarrow write to reg from mem

$\chi_{imm} :$ 0 \Rightarrow second operand is register

1 \Rightarrow second operand is immediate value

write : 00 \Rightarrow dont write to reg

dest 01 \Rightarrow write to rs

10 \Rightarrow write to rt

11 \Rightarrow write to zl

Mem w/R¹ : 1 \Rightarrow write to memory

0 \Rightarrow read from memory

$\chi_{ALU} :$ 000 \Rightarrow branch instruction

001 \Rightarrow R-format instruction

010 \Rightarrow Immediate instruction that requires add

011 \Rightarrow Immediate instruction that requires complement.

ALU

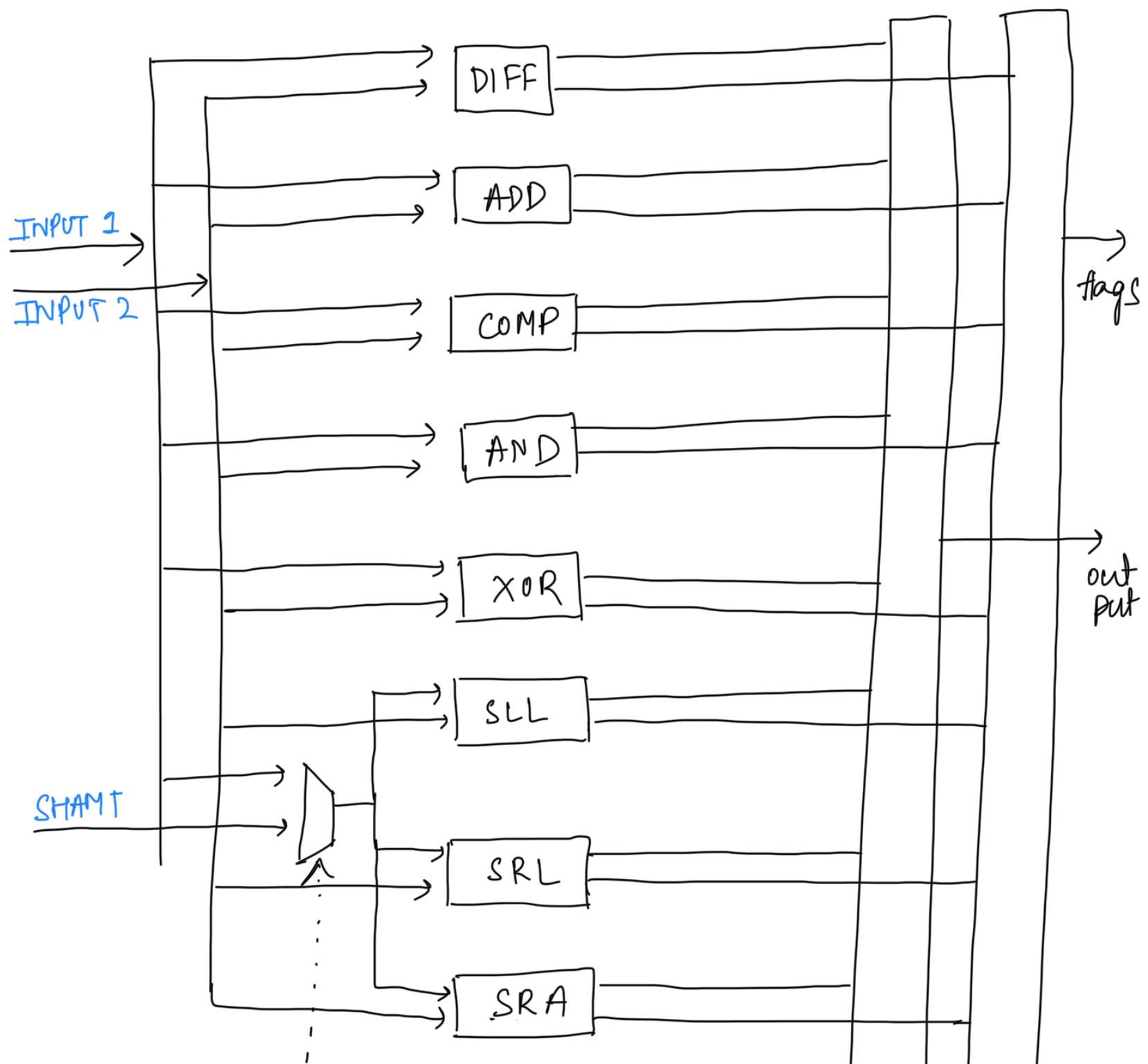
<u>x ALU</u>	<u>Fn code</u>	<u>ALU control</u> \Rightarrow
000	X	0111
001	1	0001
001	2	0010
001	3	0011
001	4	0100
001	5	0000
001	6	1110
001	7	1101
001	8	0110
001	9	0101
001	10	1111
001	11	0111
010	X	0001
011	X	0010

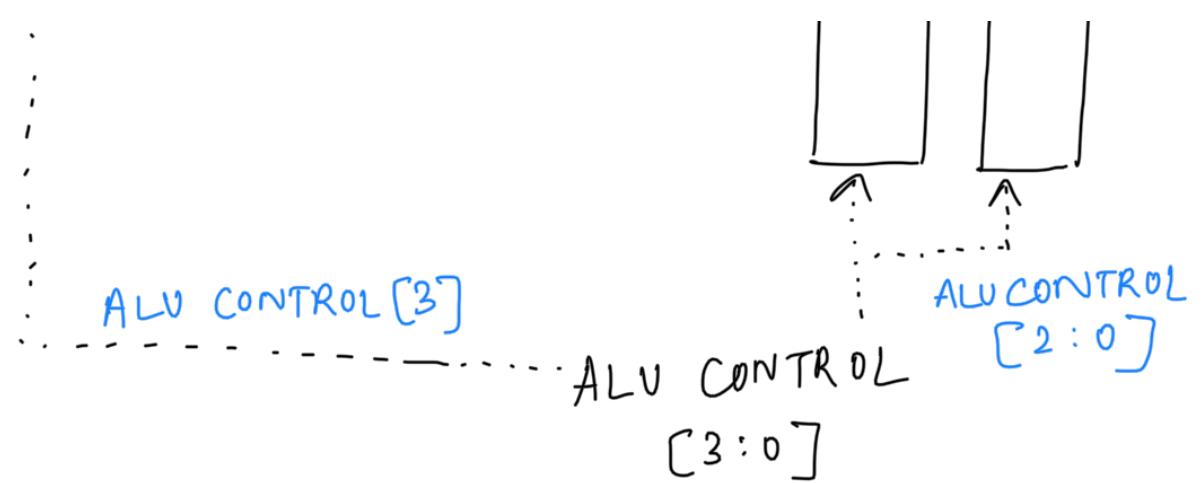
ALU operations

001 010 011 100 101 110 111
 ADD, COMPLEMENT, AND, XOR, SRL, SLL, SRA
 000
 diff

↳ 8 operations + 1 bit for knowing
 whether Shamt
 required or not

⇒ ALU control → 4 bits

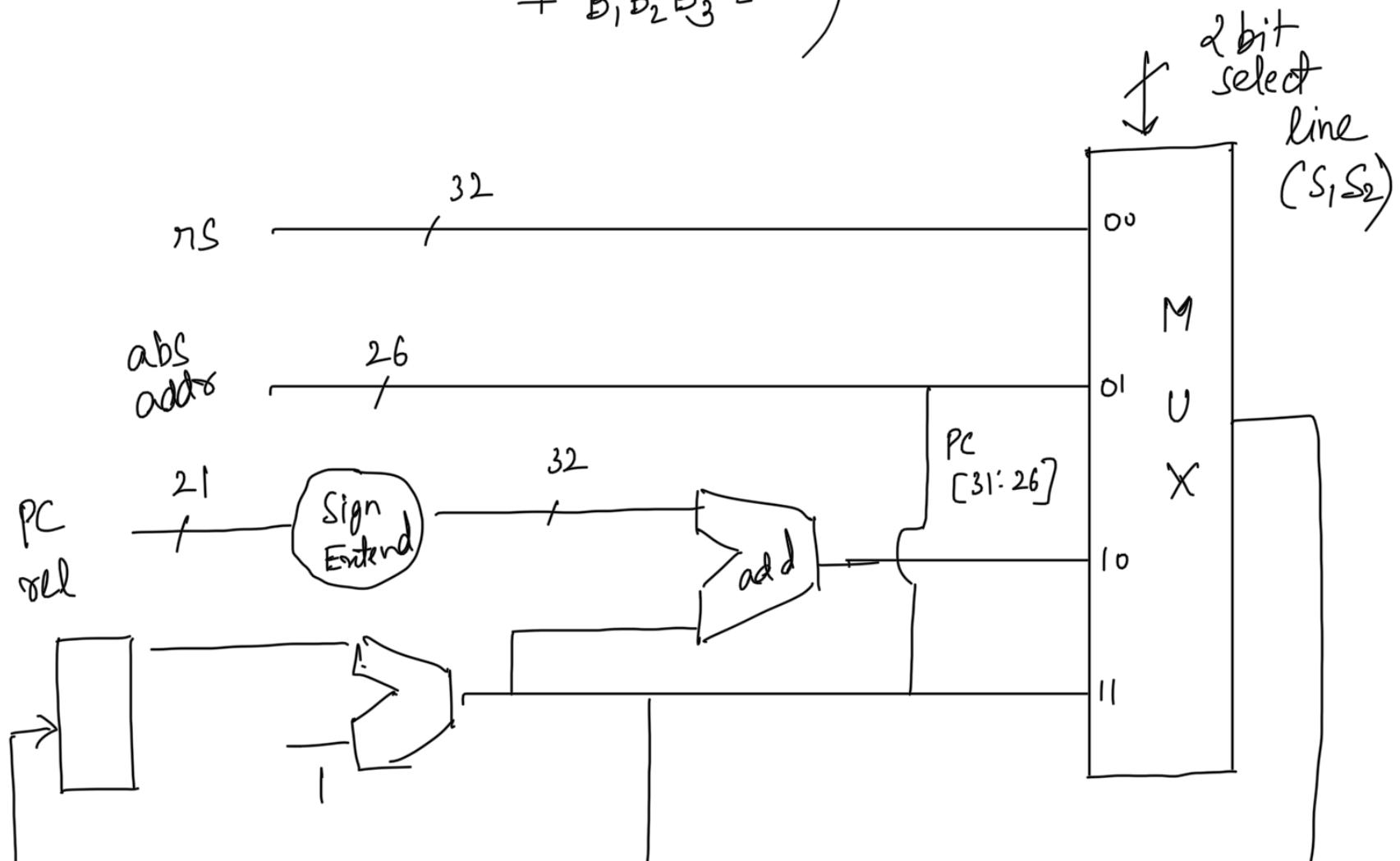




Formula for sel 1 and sel 2

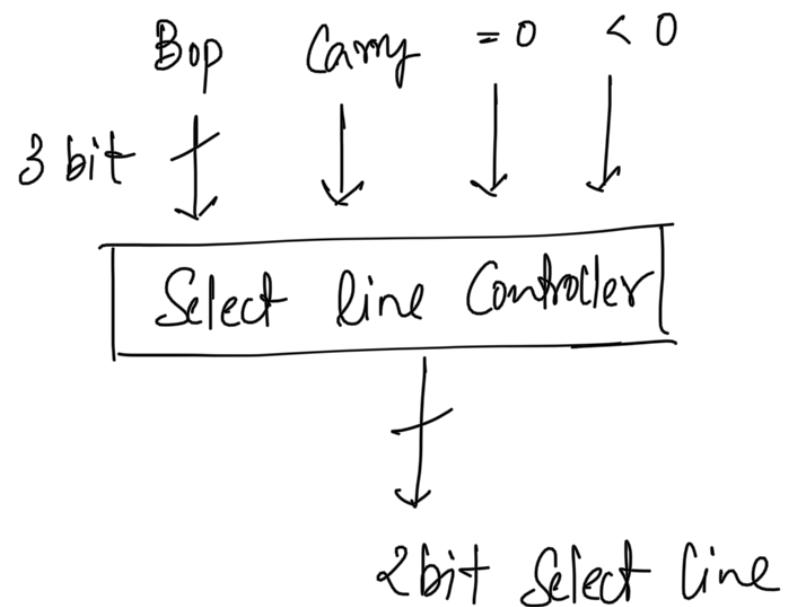
$$7 \left(\bar{B}_1 \bar{B}_2 \bar{B}_3 + \bar{B}_1 \bar{B}_2 B_3 + B_1 \bar{B}_2 B_3 C + B_1 B_2 \bar{B}_3 \bar{C} \right) = \text{sel 1}$$

$$7 \left(\bar{B}_1 \bar{B}_2 B_3 + \bar{B}_1 B_2 \bar{B}_3 \bar{E}L + \bar{B}_1 B_2 B_3 \bar{E} \bar{L} + B_1 \bar{B}_2 \bar{B}_3 \bar{E} \right) = \text{sel 2}$$





Output to Regfile



B op 1	B op 2	B op 3	carry	=0	<0	sel1	sel2
0	0	0	-	-	-	0	1
0	0	1	-	-	-	0	0
0	1	0	-	0	1	1	0
0	1	0	-	-	0	1	1
0	1	1	-	1	0	1	0
0	1	1	-	0	-	1	1
1	0	0	-	0	-	1	0
1	0	0	-	1	0	1	1
1	0	0	1	1	-	-	0
1	0	1	0	-	-	1	1
1	1	0	0	-	-	0	1
1	1	0	1	-	-	1	1
1	1	1	-	-	-	1	1

S flag L flag O > One's comp Two's comp S flag L flag
 0 0 - - - 0 0 0

B op - 0

000 - unconditional branch and Branch and Link

001 - Branch Register

010 - Branch on less than 0

011 - Branch on flag 0

100 - Branch on flag not 0

101 - Branch on carry

110 - Branch on no-carry

111 - No sequentially next instruction