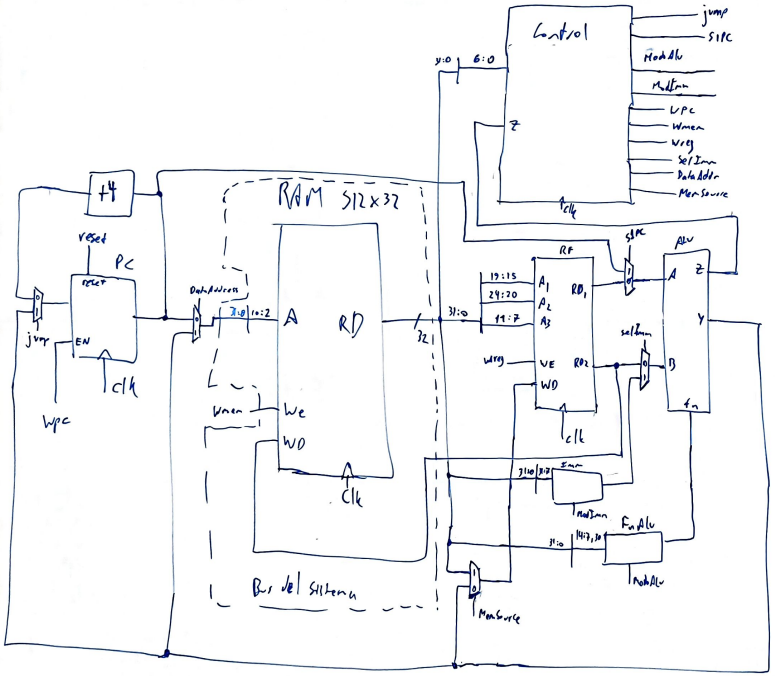


Mal Inm	
000	Comp. = 4
001	I
010	S
011	B
100	U
101	J

jal rd, label

PC ← PC + Inm

rd ← PC + 4

Fetch

Decode

Execute

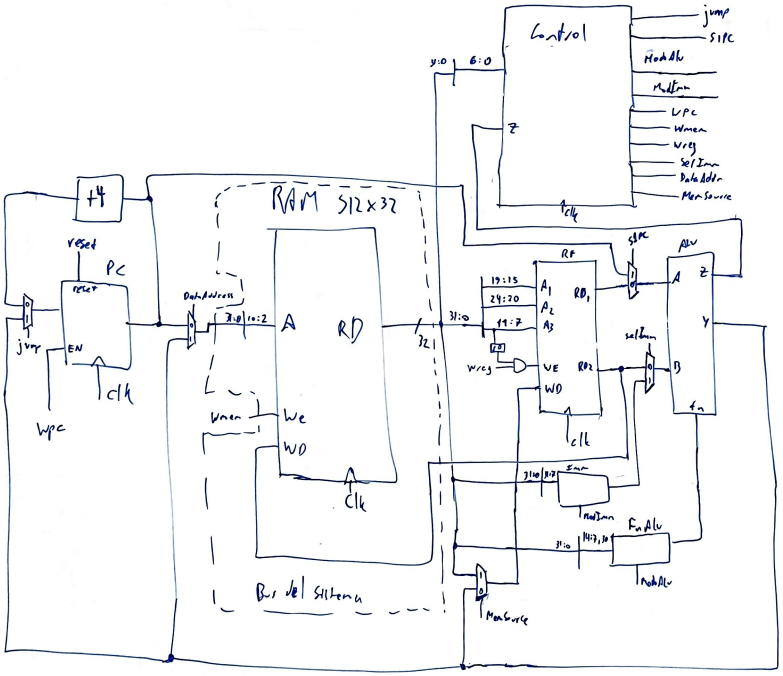
0x8 Suma_5_4:

addi $x_1, x_0, 5$

addi $x_2, x_0, 4$

add x_3, x_1, x_2

add: $x_0, x_0, 0$



jal rd, label

PC' ← PC + Imm

rd ← PC + 4

Mod Imm	
000	Const. = 4
001	I
010	S
011	B
100	U
101	J

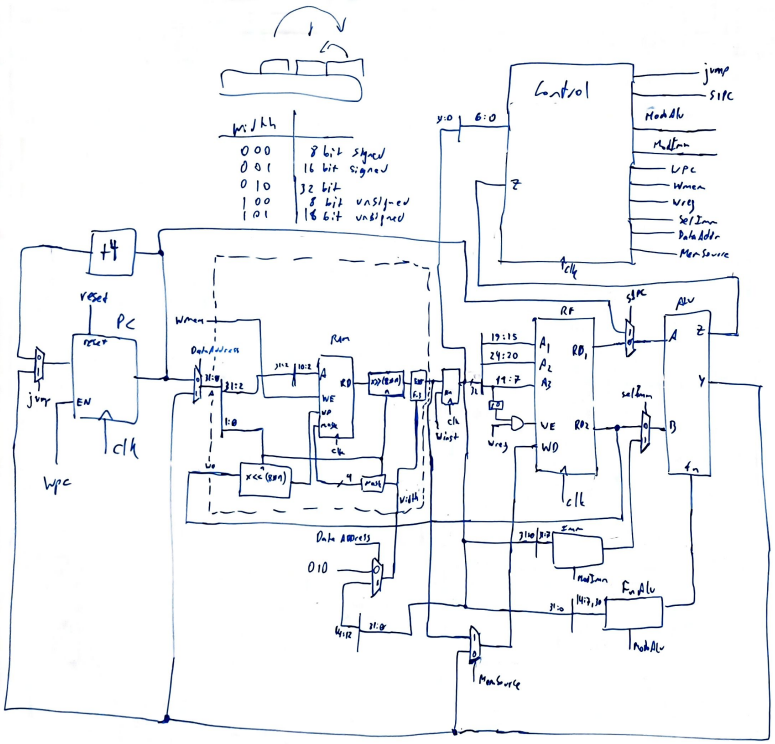
Fetch

Decode

Execute

0x0 Suma_A_B:

```
la  X1, A
LW  X2, 0(X1)
LW  X3, 4(X1)
add X4, X2, X3
SW  X4, 8(X1)
fin: j fin
A:  .word 5
B:  .word 4
R:  .word 0
```



jal rd, label

PC' ← PC + Imm

rd ← PC+4

Fetch

Decode

Execute

0x0 Sum_A_B:

la X_1, A

Lw X_2, 0(X_1)

Lw X_3, 4(X_1)

add X_4, X_2, X_3

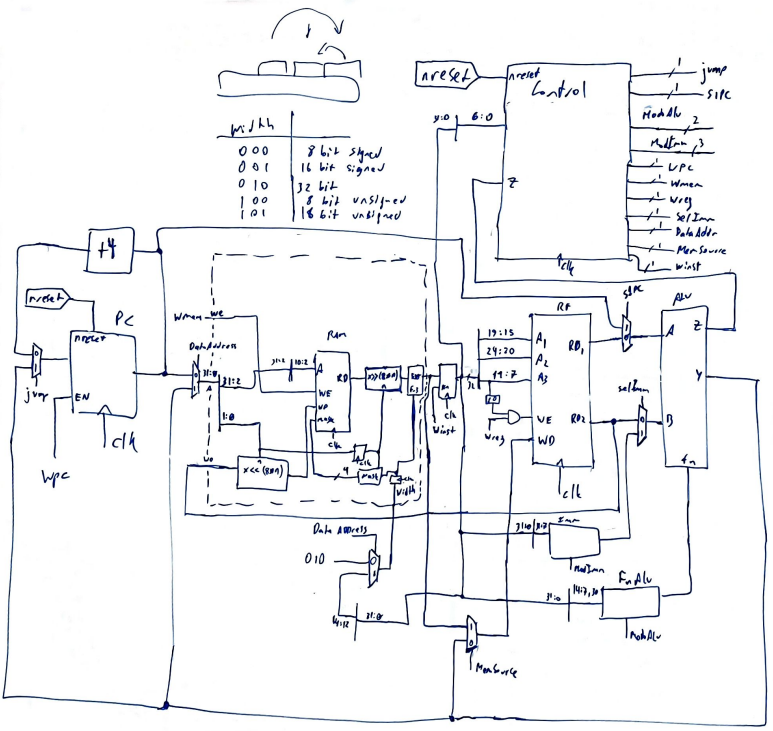
SW X_4, 8(X_1)

fin: j fin

A: .word 5

B: .word 4

R: .word 0



Width	
000	8 bit signed
001	16 bit signed
010	32 bit
100	8 bit unsigned
101	16 bit unsigned

MathAlu	
00	sums
01	addi .. andi
10	add .. and
11	bxx (cmp)

jal rd, label

PC' ← PC + Imm

rd ← PC+4

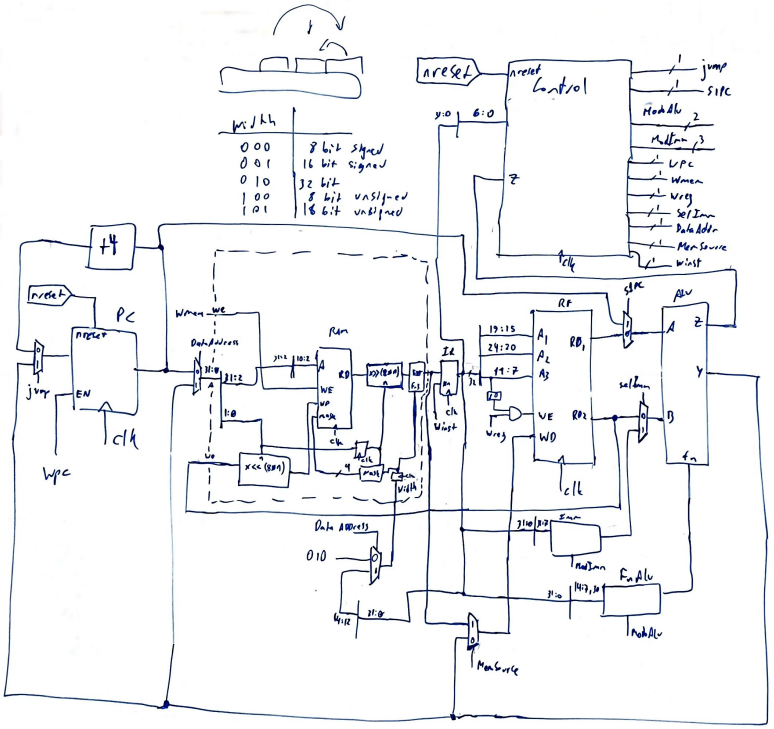
MadImm	
000	Const. = 4
001	I
010	S
011	B
100	U
101	J

Fetch

Decode

Execute

```
.org sum_A_B:
    la  x1, A
    lw  x2, 0(x1)
    lw  x3, 4(x1)
    add x4, x2, x3
    sw  x4, 8(x1)
fin: j fin
A:  .word 5
B:  .word 4
R:  .word 0
```



MemAlu	
00	sum
01	addi .. andi
10	add .. and
11	bxx (cmp)

jal rd, label

PC ← PC + Imm

rd ← PC+4

Multrm	
000	Const. = 4
001	I
010	S
011	B
100	U
101	J

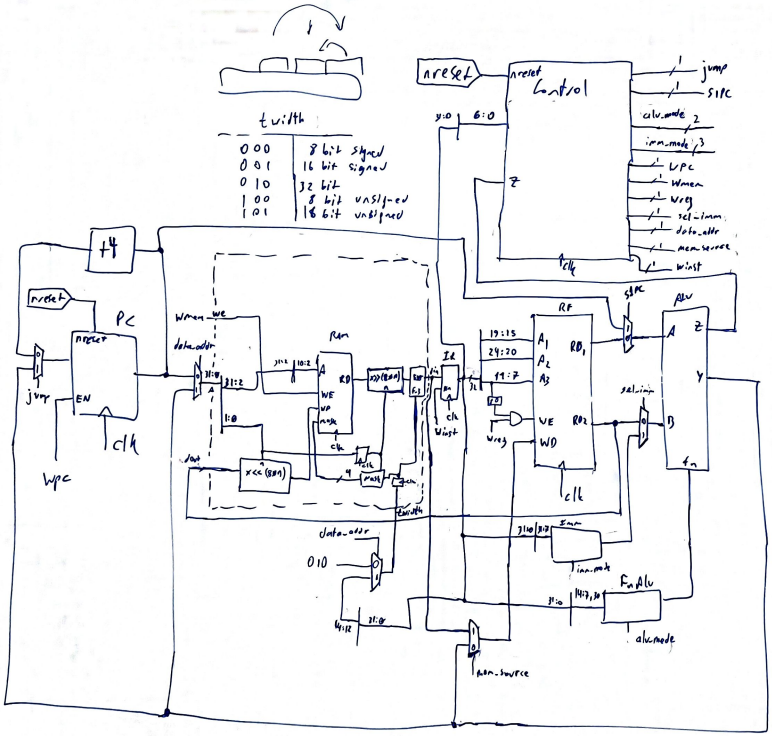
Fetch

Decode

Execute

0x0 Sum_A_B:

```
la  X1, A
Lw  X2, 0(X1)
LW  X3, 4(X1)
add X4, X2, X3
SW  X4, 8(X1)
fin: j fin
A:  .word 5
B:  .word 4
R:  .word 0
```



twidth
000 8 bit signed
001 16 bit signed
010 32 bit
100 8 bit unsigned
101 16 bit unsigned

alu_mode
00 sum
01 addi .. andi
10 add .. and
11 bxx (cmp)

jal rd, label
PC' ← PC + Imm
rd ← PC + 4

imm_mode
000 (cmp.) = 4
001 I
010 S
011 B
100 U
101 J

Fetch
Decode
Execute

0x0 Sum_A_B:
la x_1, A
Lw x_2, 0(x_1)
Lw x_3, 4(x_1)
add x_4, x_2, x_3
SW x_4, 8(x_1)
fin: j fin
A: .word 5
B: .word 4
R: .word 0