CSE 340
Chapter-4

4 Ishmam the cat

CPU time = Instruction count × CPI × clock cycle time.

Two types of risc-V implementation:

(I) A simplified version (slow, too much time wasted)

(II) A more realistic pipelined version (optimized)

Instruction 3 category

→ Main memory associated

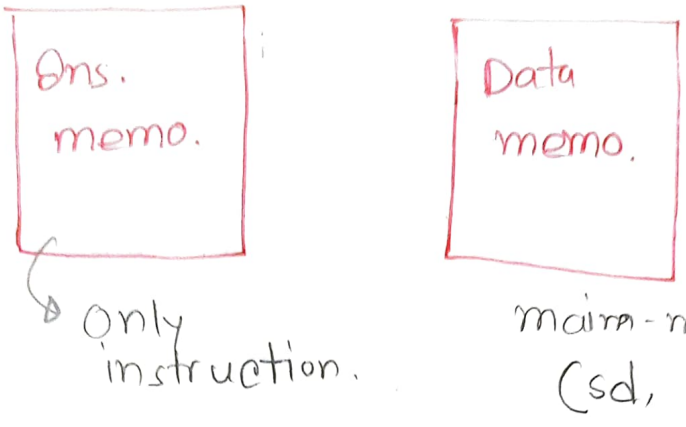
(I) memory reference : ld, sd.

(II) Arithmetic / logical : add, sub, and, or

(III) Control tranfer : beq.
↳ Branching.

Instruction Execution.

PC → program counter register → contains the address of the instruction in the program being executed.

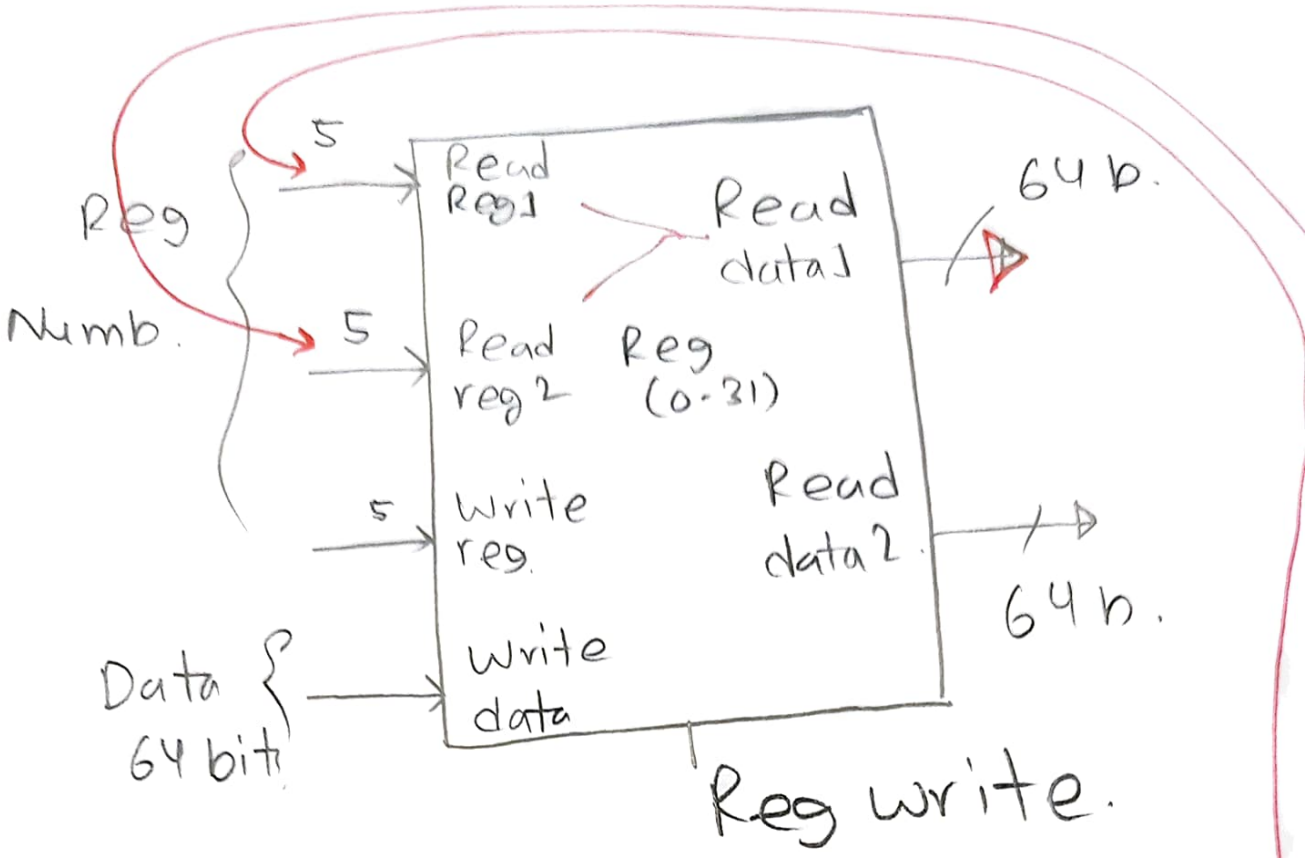


#8000 → add x_{20}, x_{21}, x_0.

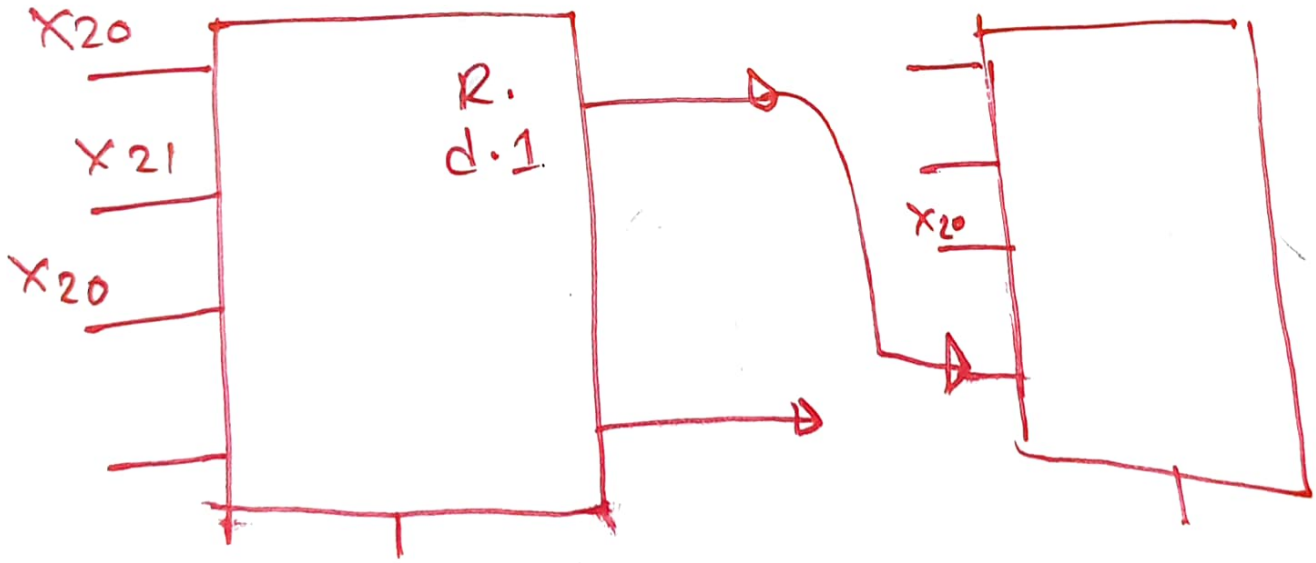
Fetch → Decode (source, destination).

rd						
XXXX	XXX0	0101	1010	0XXX	0011	0XXX
f_7.	rs2		rs1	f_3		XXXX
						op.

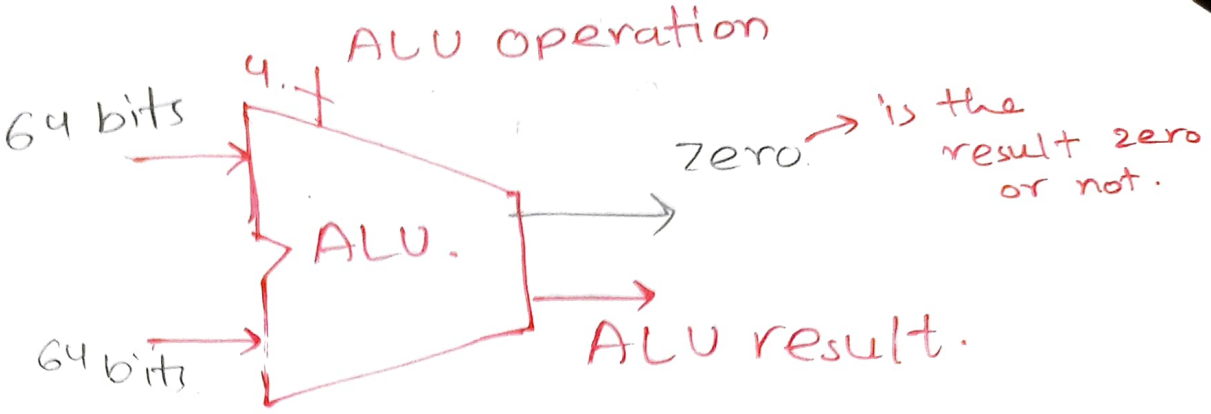
Explaination of negisten file.



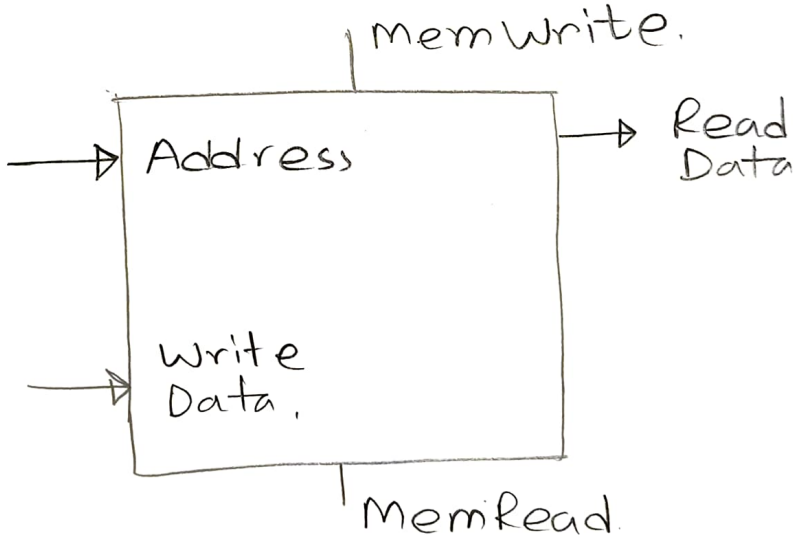
add X_{20}, X



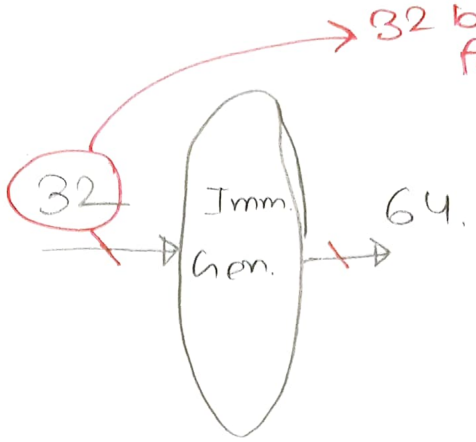
Explaination of ALU.



Data memory & Immediate Generation unit



a. Data Memory Unit.



b. immediate gene. unit

ld X_{21}, 12[X_{22}] → Base + offset

⇒ Actual = X_{22} + 12.

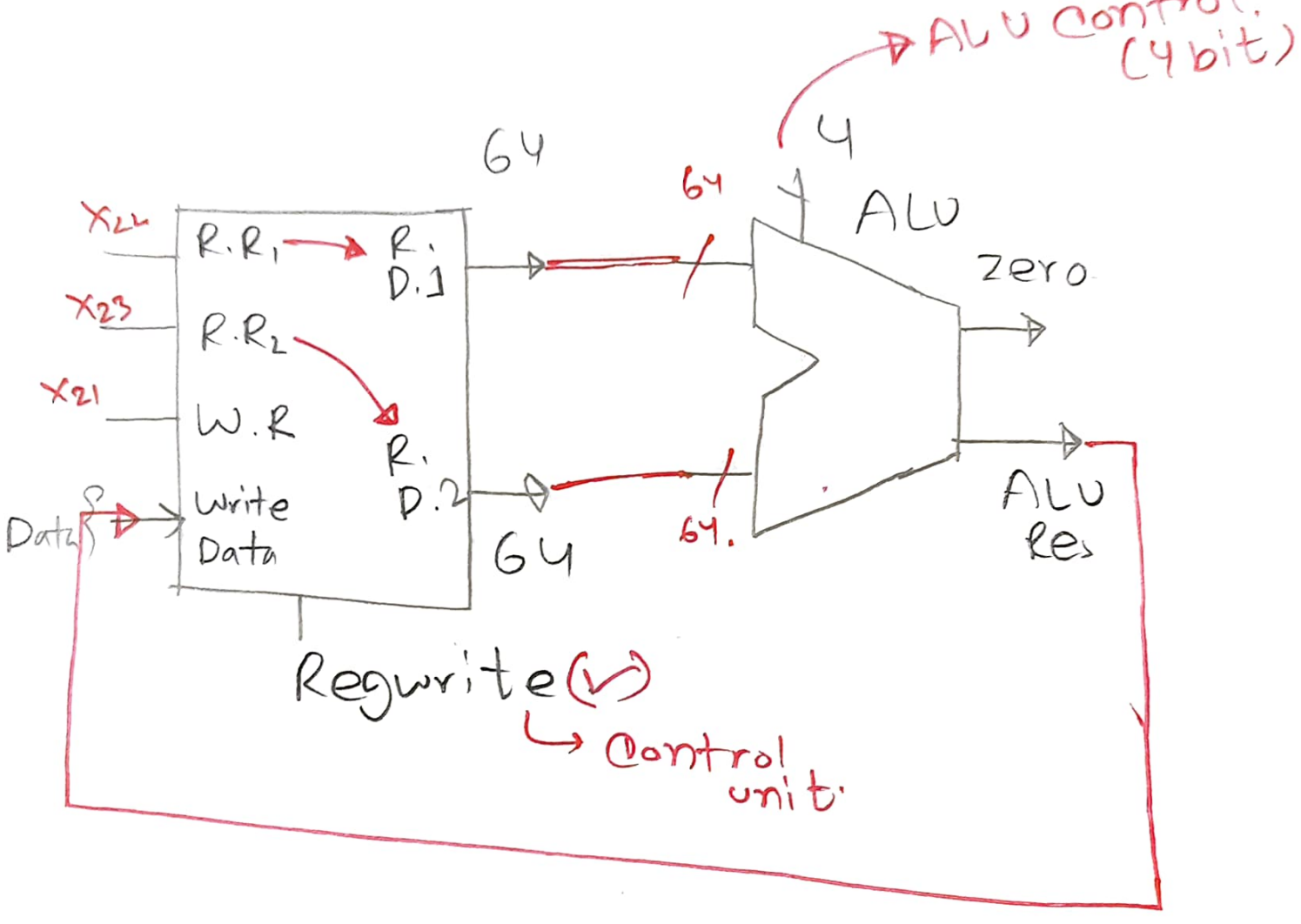
→ 64 bits.

→ 12 bits → IGU → 64 bits.

R-type format.

→ Read two register operand
→ Perform logical operation.
→ write register result.

add X_{21}, X_{22}, X_{23}



Load / Store instruction

- Read registrate operands.

→ Calculate address using 12-bit operand

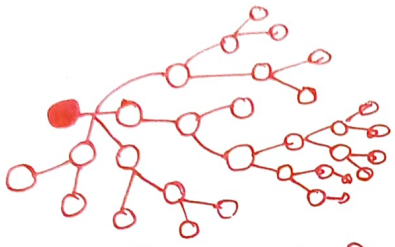
* USE ALU but sign extended ~~oper~~. offset.

→ Load: Read memory and update register

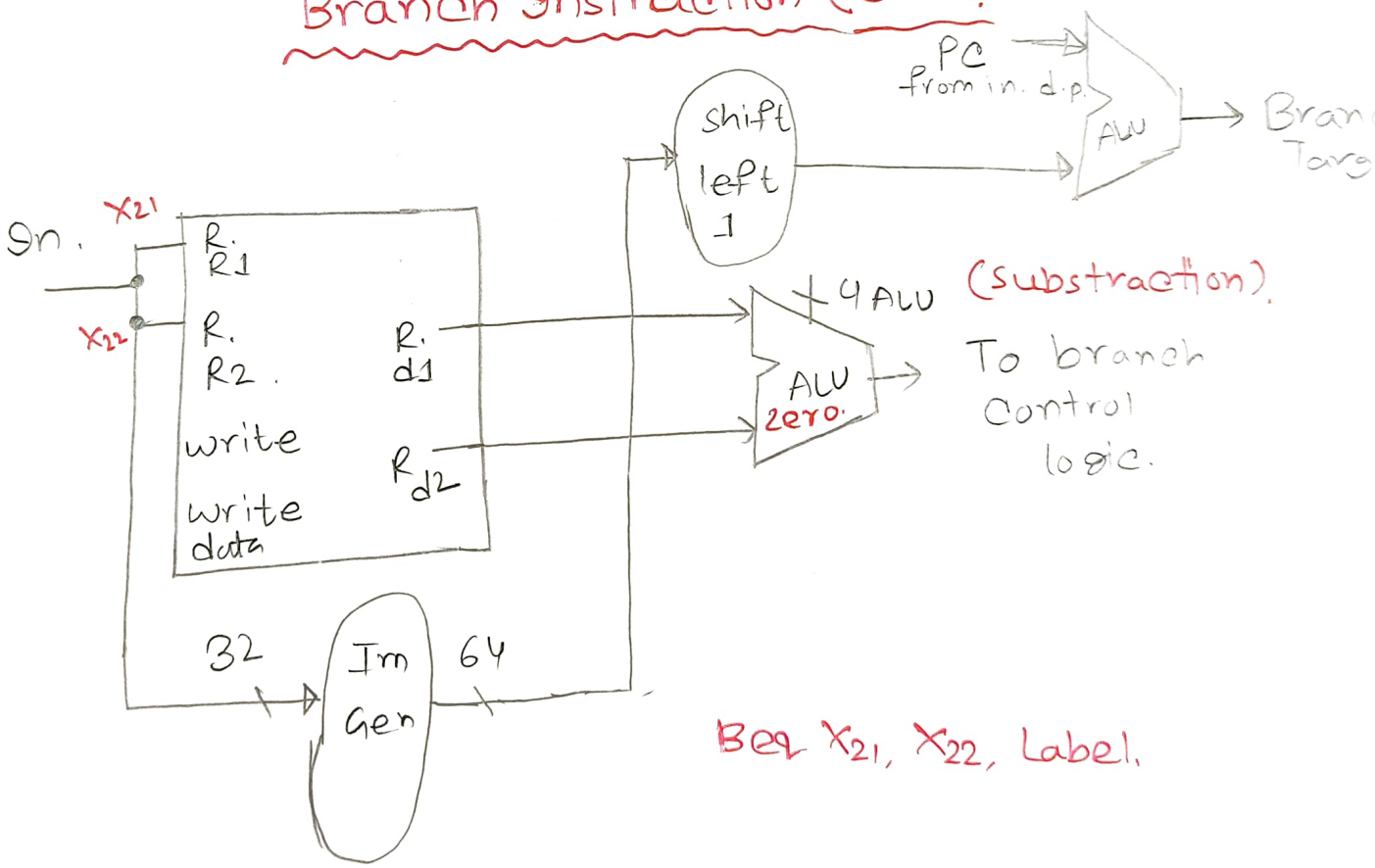
→ Store: write register value for memory.

LD X_{21}, 16 (X_{22})

SD X_{21}, 16 (X_{23})

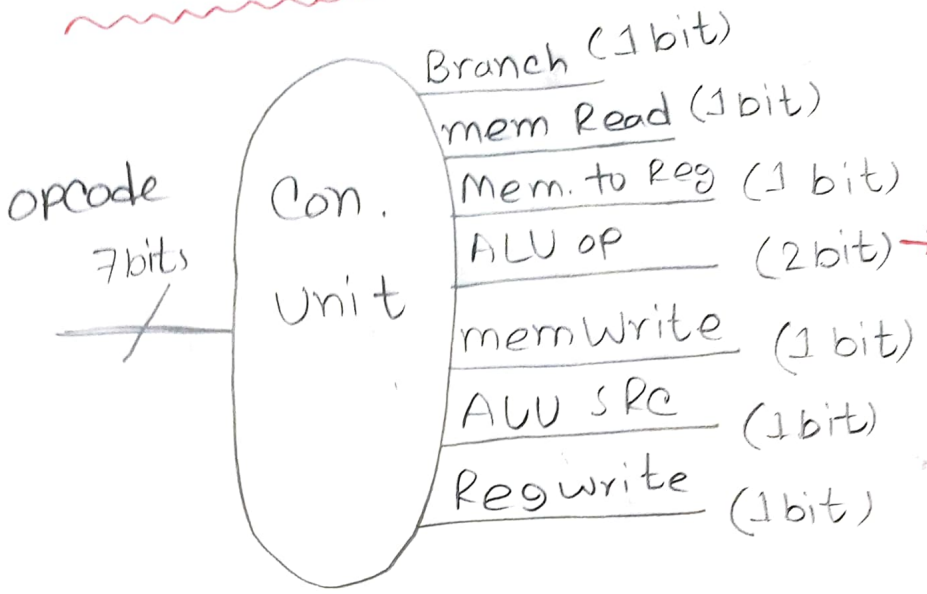


Branch Instruction (BEQ).



Beq X_{21}, X_{22}, Label.

Control unit & ALU cont. unit

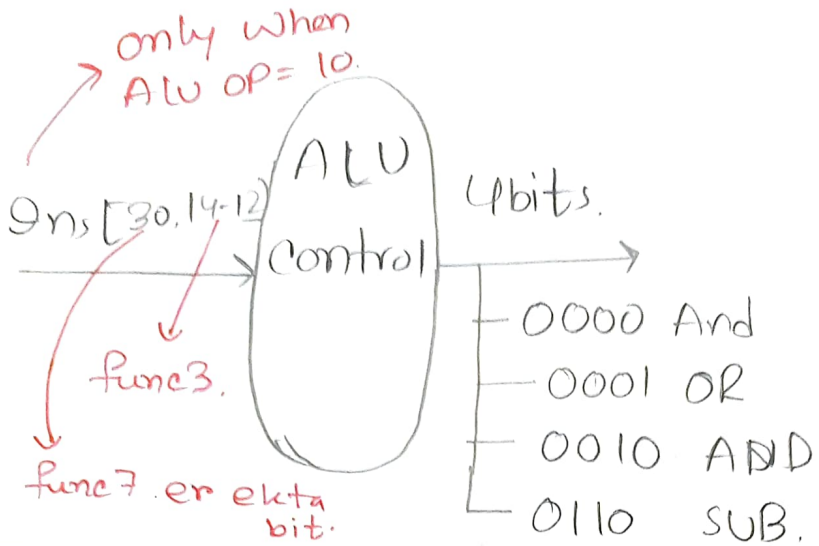


ALU OP → 00 → ld / store / addi

01 → Beq

10 → R-type.

ALU control



অবার এর 6টা bit ই প্রয়োজন হচ্ছে. কোন একটা bit change হলে

Datapath for Instruction Fetch.

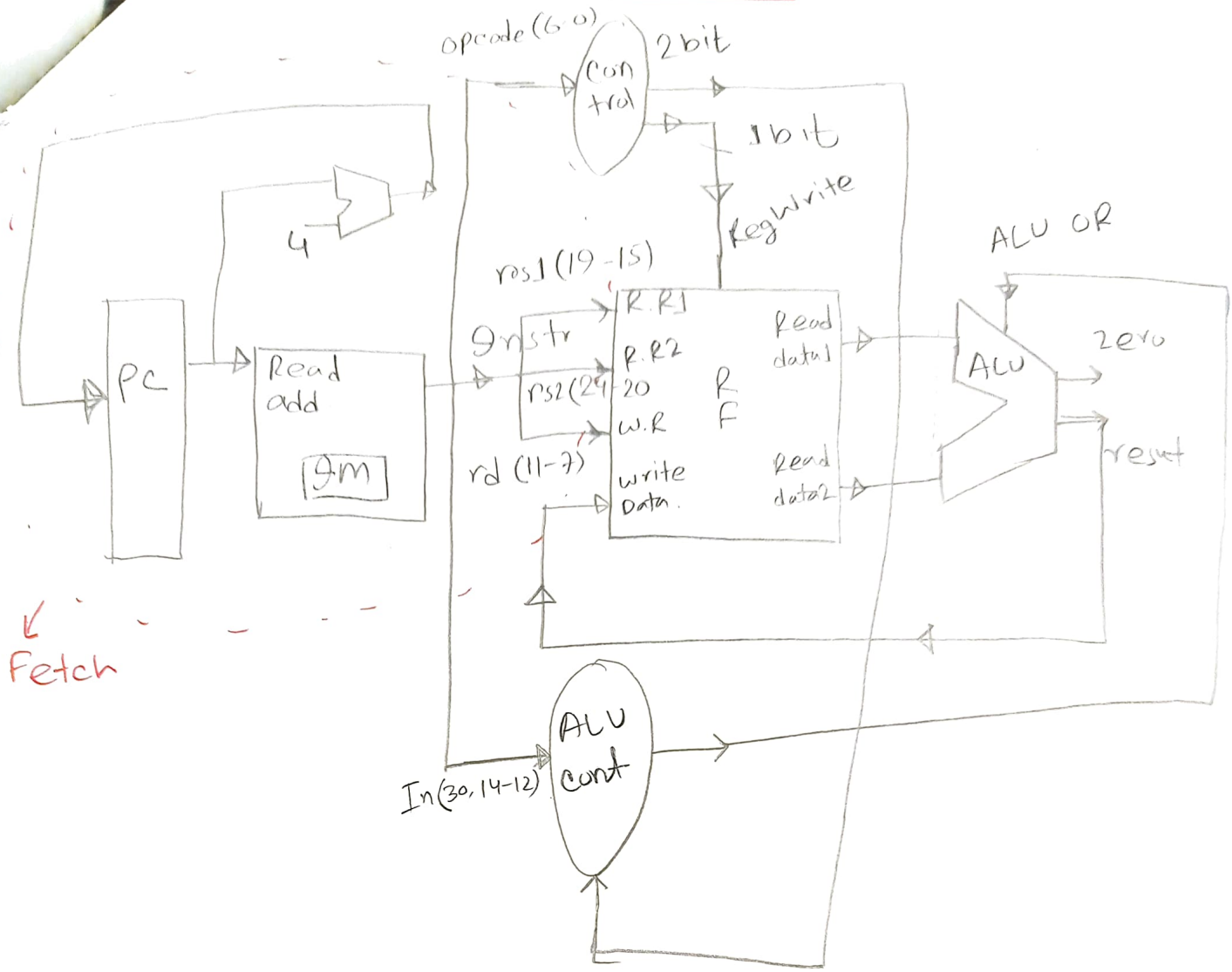
Building Datapaths.

Instruction Fetch:

- Access the program counter.
- Pass the address/location to inst mem.
- Ins memory outputs the respective instruction.
- Increament the PC.

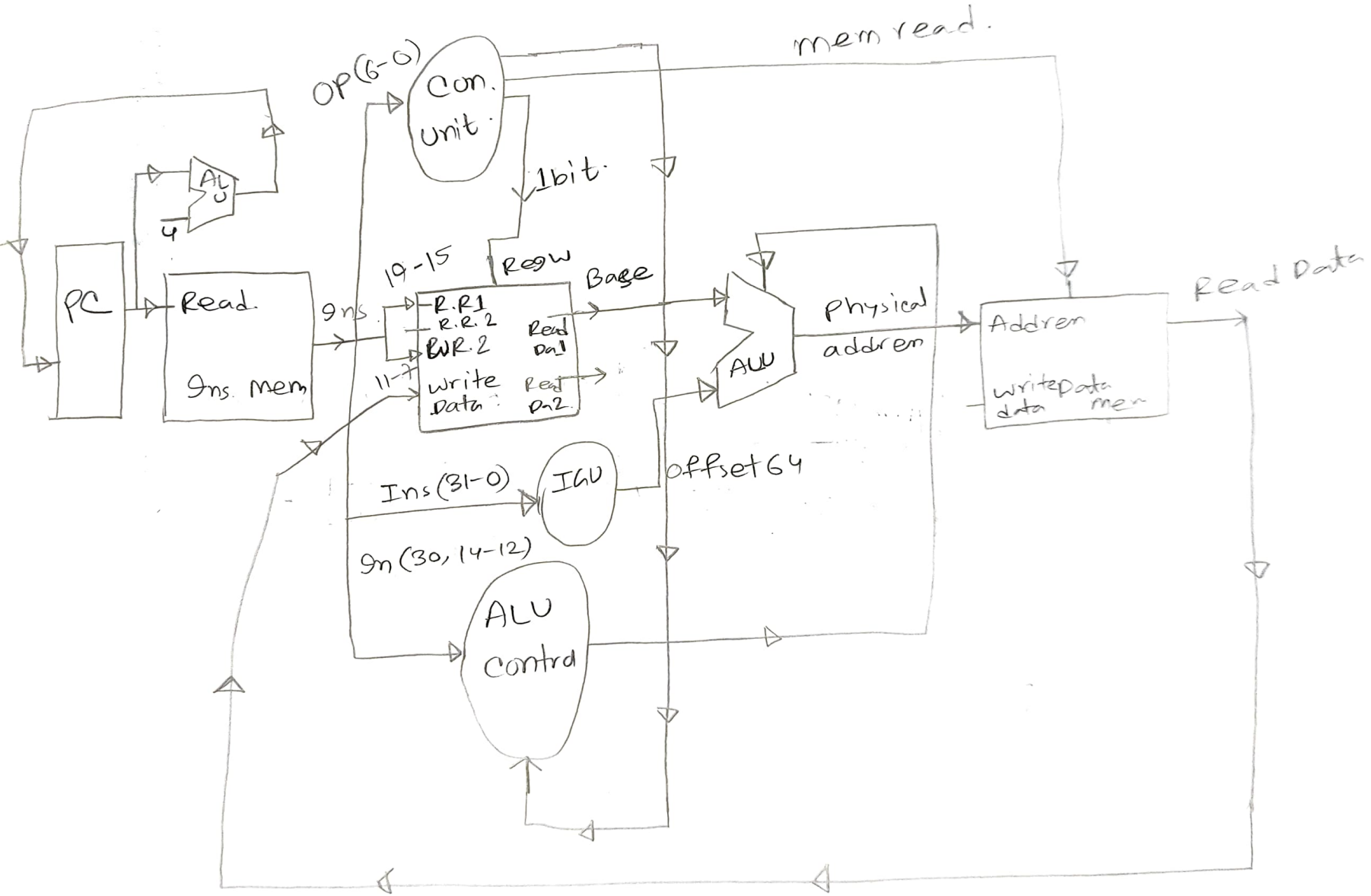
Slow
Ishmam

Datapath - R-type

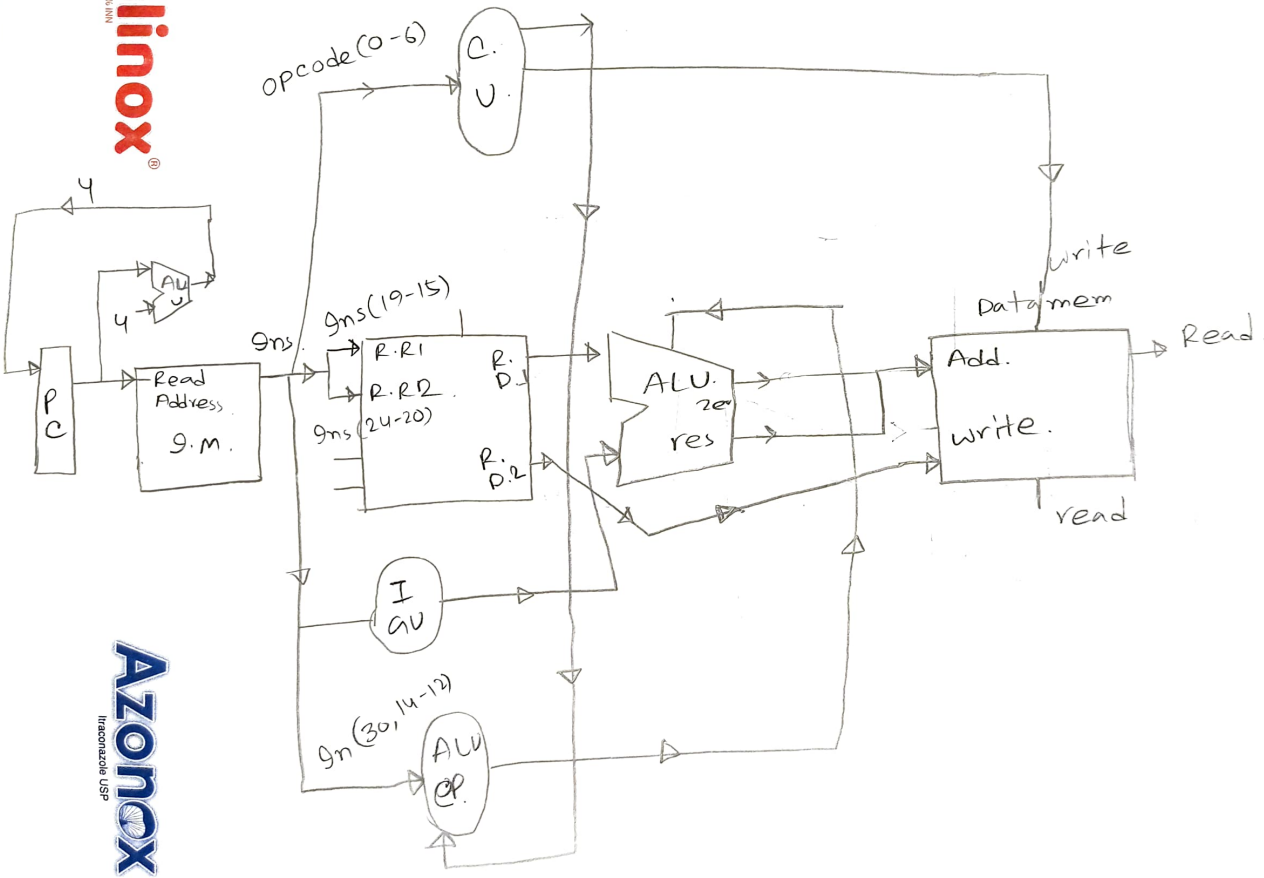


load instruction.

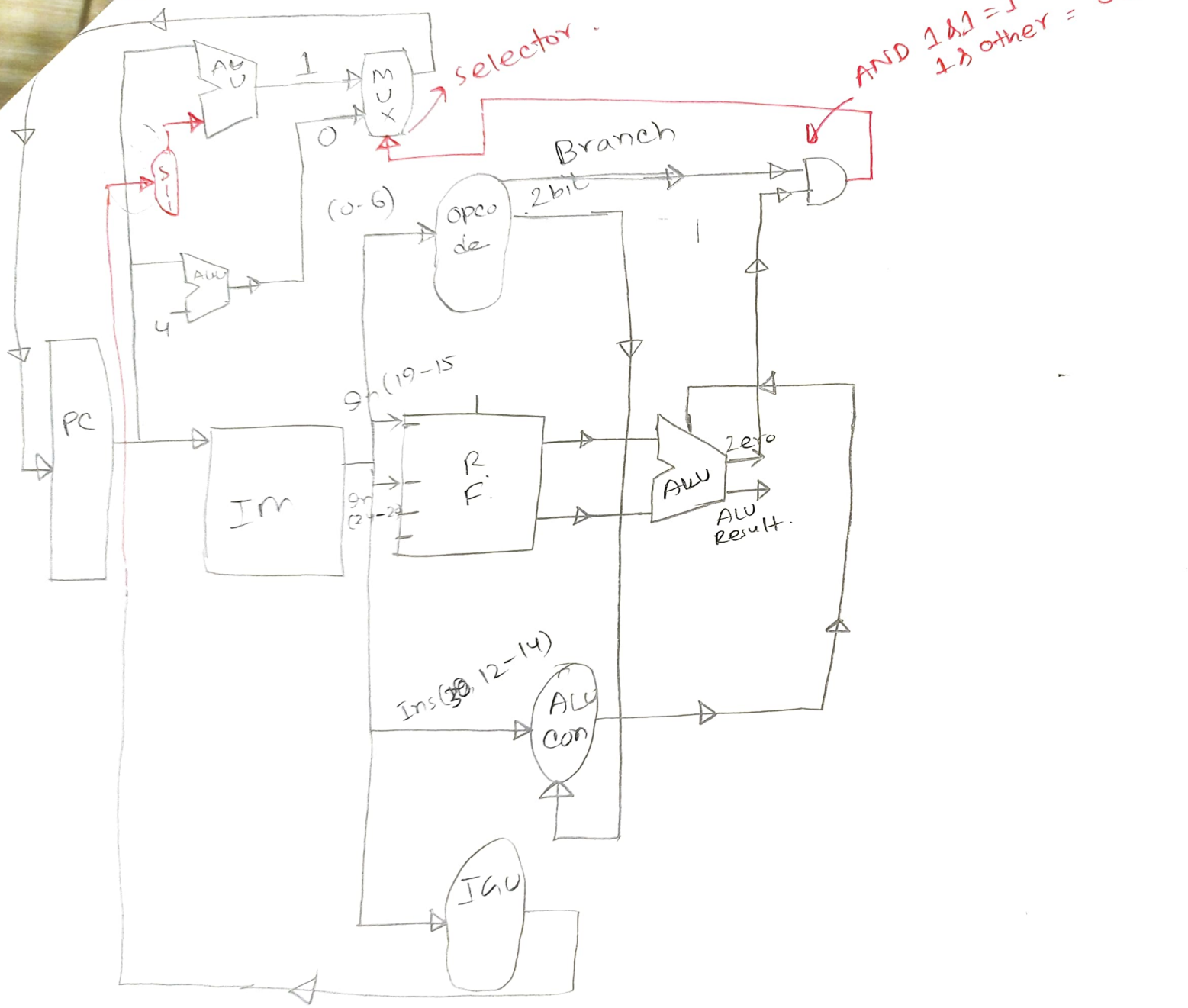
ld X_{21}

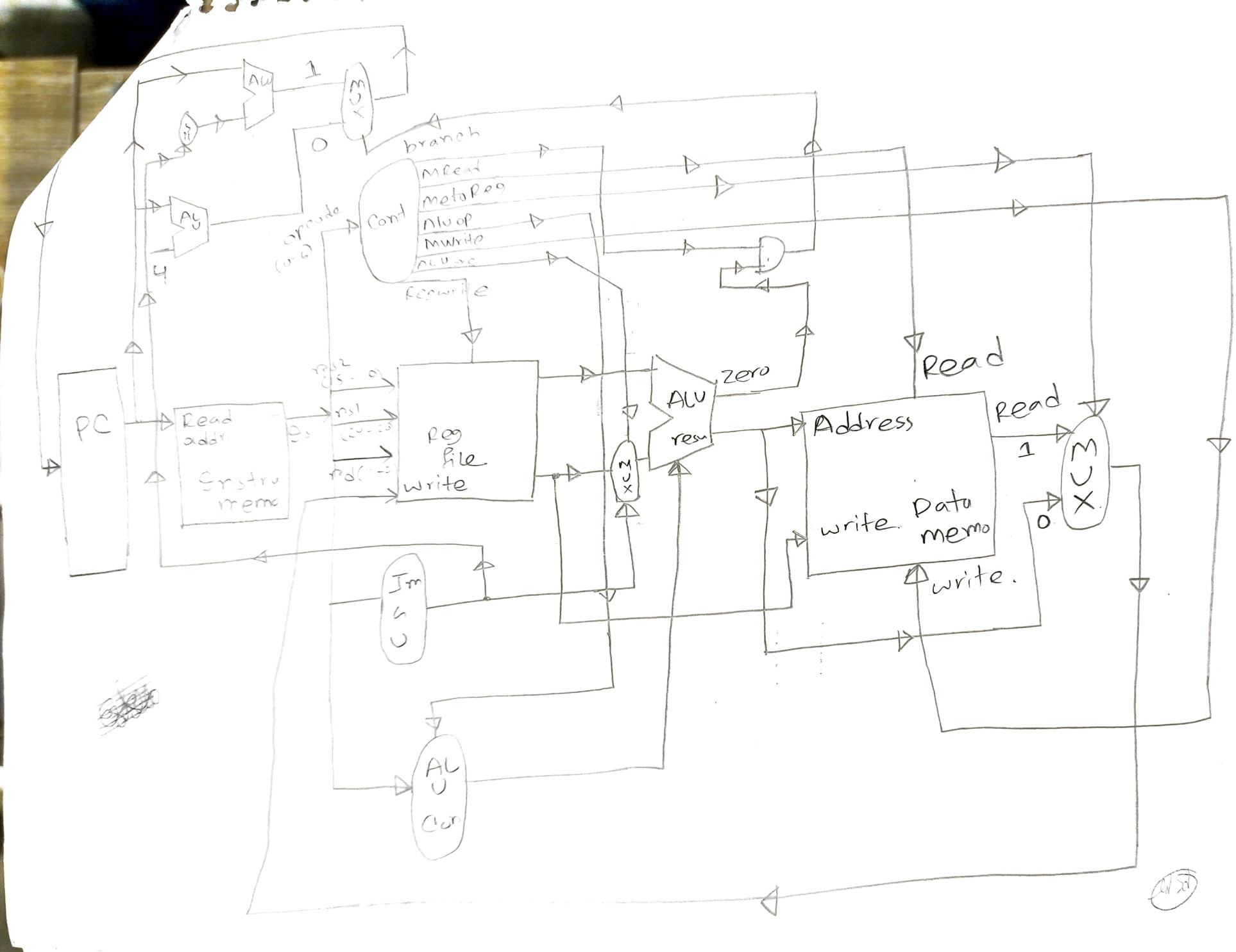


Store Instruction



nstruction


selector.
AND 1 && 1 = 1
1 && other = other.
ALU
1
MUX
0
Branch
2bit
(0-6)
Opco
de
ALU
4
PC
IM
9n(19-15
R.
F.
9n
(2
ALU
zero
ALU
Result.
Ins(12-14)
ALU
Con
IGU


ALU
1
M
U
X
0
branch
MRead
MetoReg
ALUop
MWrite
ALUSrc
RegWrite
Cont
opcode
4
PC
Read addr
Instru mem
rs2
rs1
rd
Reg File
write
ALU
zero
res
Read
Address
write Data mem
write.
Read
1
0
Im
G
U
ALU
Ctr

Pipelining Analogy

Pipelined laundry:

Instruction :

(I) Wash. (W)
(II) dry. (D)
(III) Fold. (F)
(IV) store (S).

Time	6		7		8		9		10		11		12		1	
A	W	D	F	S												
B					W	D	F	S								
C									W	D	F	S				
D													W	D	F	S
Pipe																
A	W	D	F	S												
B		W	D	F	S											
C			W	D	F	S										
D				W	D	F	S									

Clock period calc. for single cycle & pipeline.

(I) Instruction fetch → 10 ps
(II) Register file read/write → 5 ps.
(III) Memory read/write → 6 ps.
(IV) ALU operation → 10 ps

Add _ _ _ → IF + RR + ALU + R.W = 10+5+10+5 = 30 ps

SD _ _ _ → IF + R.R + ALU + M.W = 10+5+10+6 = 31 ps

LD _ _ _ → IF + RR + ALU + MR + R.W = 10+5+10+6+5 = 36 ps.

↳ ld X_{21}, 34(X_{20})

* single cycle datapath এ যে instruction টা run করতে highest time লাগে সেটাই হচ্ছে clock period.

→ Clock period 36 ps.

Pipelined ~~IF, ID~~

IF = 5
ID = 10
EX = 11
Mem = 10
WB = 5

and
or
Add
LD.

→ Clock period.

Clock period calc. for single cycle & pipleline.

(I) Instruction fetch → 10 ps
(II) Register file read/write → 5 ps.
(III) Memory read/write → 6 ps.
(IV) ALU operation → 10 ps

Add - - - → IF + RR + ALU + R.W = 10+5+10+5
= 30 ps

SD - - - → IF + R.R + ALU + M.W = 10+5+10+6
= 31 ps

LD - - - → IF + RR + ALU + MR + R.W =
10+5+10+6+5
= 36 ps.

Ld X_{21}, 34(X_{20})

* single cycle datapath এ যে instruction ~~for~~ run করতে highest time লাগে সেইটাই হচ্ছে clock period.

→ Clock period 36 ps.

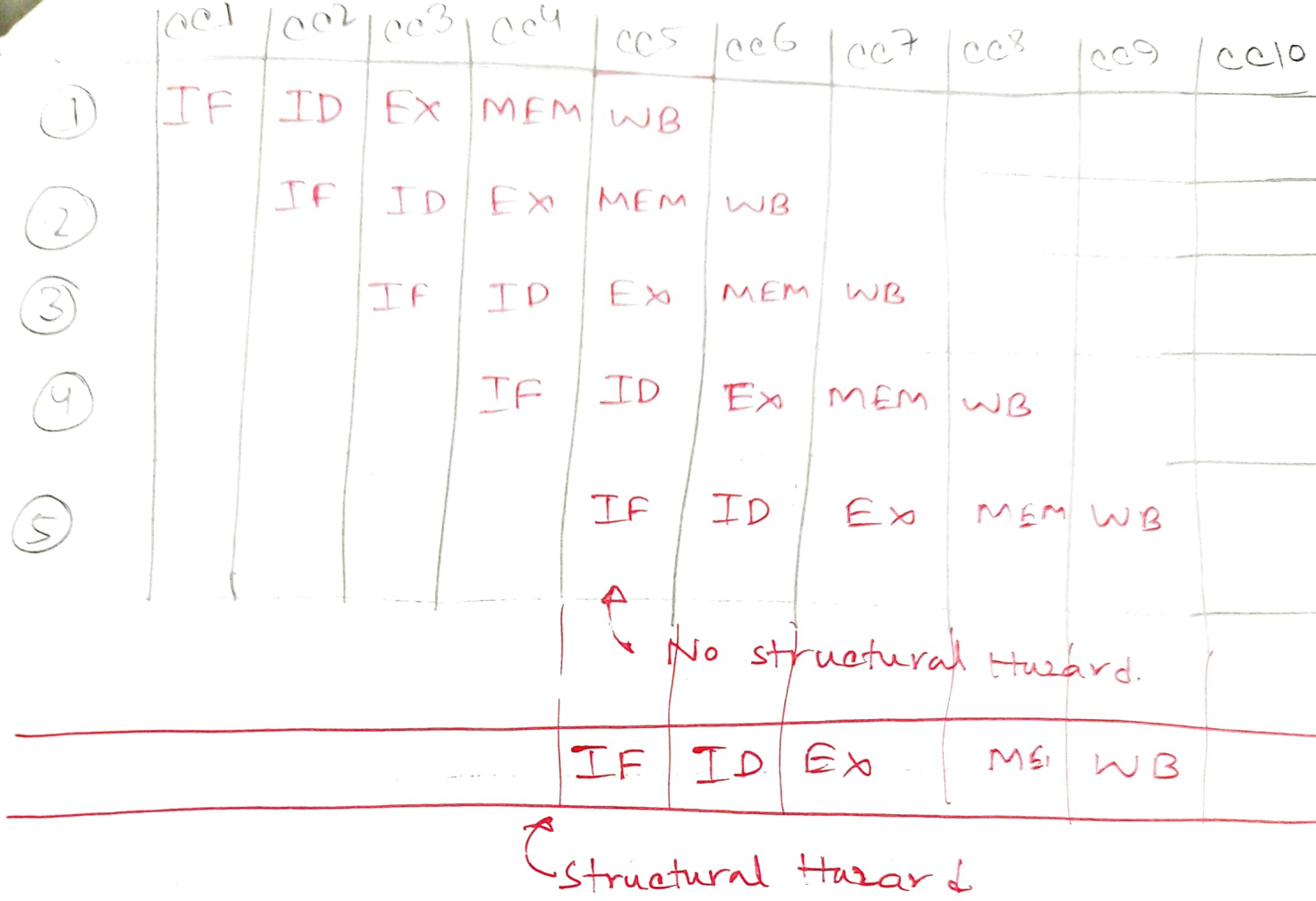
Pipelined ~~If, ID~~ →

IF = 5
ID = 10
EX = 11
Mem = 10
WB = 5

and
or
Add
LD.

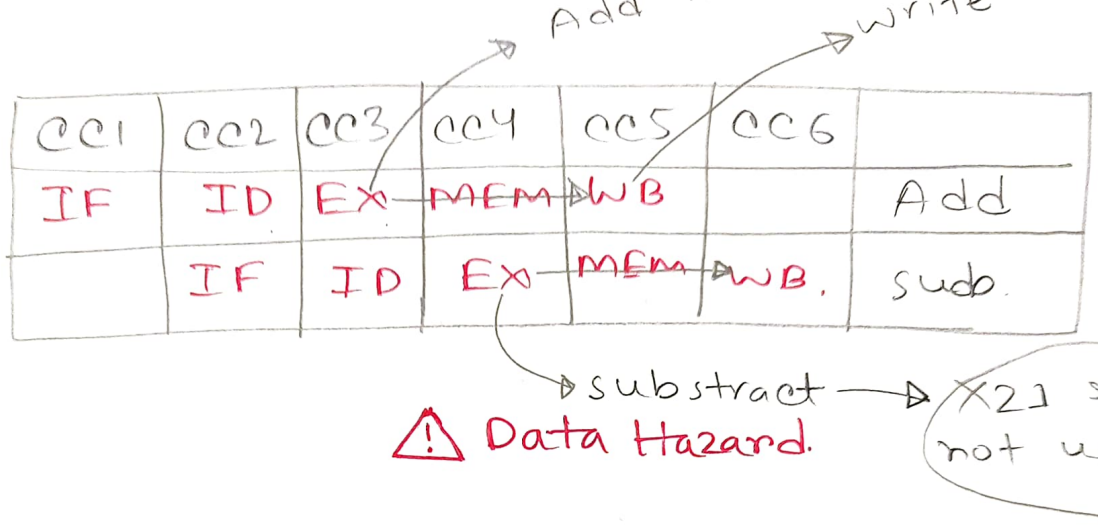
→ Clock period.

Structual Hazard



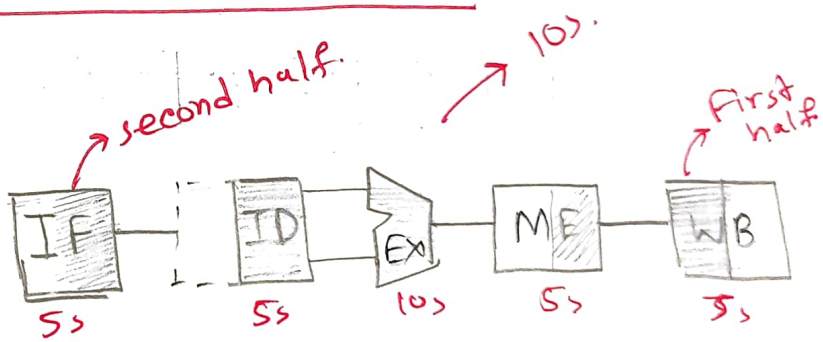
Data Hazard

Add X_{21}, X_{22}, X_{23}.
Sub X_{24}, X_{21}, X_{20}



Resolving Data Hazard using stalling.

ld X_{20}, 40(X_{21})
Add X_{22}, X_{20}, X_{21}
sub X_{22}, X_{22}, X_{20}.
Add X_{22}, X_{22}, X_{21}
slli X_{22}, X_{22}, 5.
Srli X_{25}, X_{25}, 8 → Independent



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
IF	ID	EX	MEM	WB												
	X	X	X	X	X											
		X	X	X	X	X										
			X	X	X	X	X									
			ID	IF	ID	EX	MEM	WB	(Because WB updates in first half							
				X	X	X	X	X	and read is in the second half).							
					X	X	X	X	X							
						IF	ID	EX	MEM	WB						
							X	X	X	X	X					
								X	X	X	X	X				
									IF	ID	EX	ME	WB			
									X	X	X	X	X	X		
										X	X	X	X	X		
											IF	ID	EX	ME	WB	
												IF	ID	EX	ME	WB

Total 18 Clock cycle.

CPI = Clock cycle per instruction

$= 18/6 = 3.$

→ 6 or instruction

Resolving Data Hazard Using Forwarding

Forwarding (Aka Bypassing)

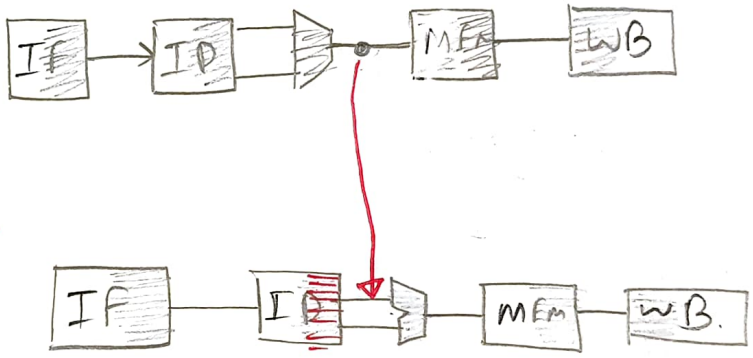
- Use result when it is computed
  - Don't wait for it to be stored in a register
- Requires extra connection

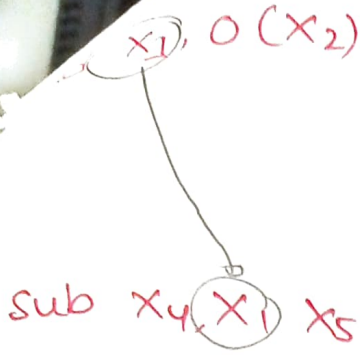
R-type

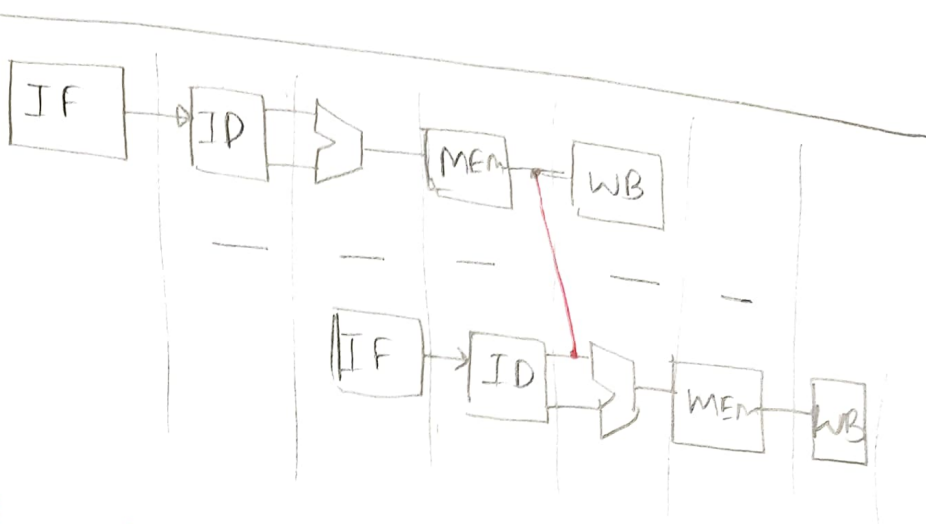
Stall = 8
Forward = 6

add X1, X2, X3.

sub X4, X1, X5







So, only forwarding isn't enough.
we need stall + forwarding.

Code scheduling

rearrang the code sequence to overcome data hazard.

How? ⟹ If one line is independent
(যেখানে stall আছে সেখানে কাজ করে)