Computing System Design

Virendra Singh

Professor

Computer Architecture and Dependable Systems Lab

Department of Electrical Engineering

Indian Institute of Technology Bombay

http://www.ee.iitb.ac.in/~viren/

E-mail: viren@ee.iitb.ac.in

CS-226: Digital Logic Design

Lecture 30: 17 April 2021

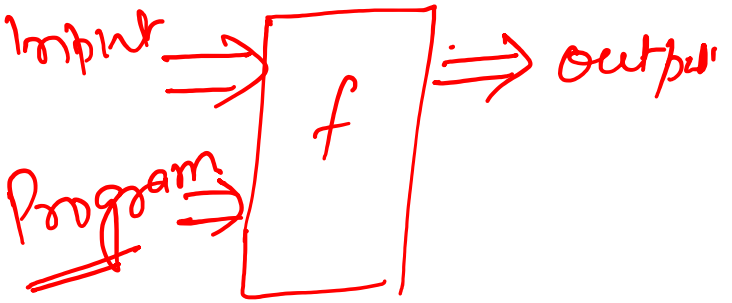
CADSL

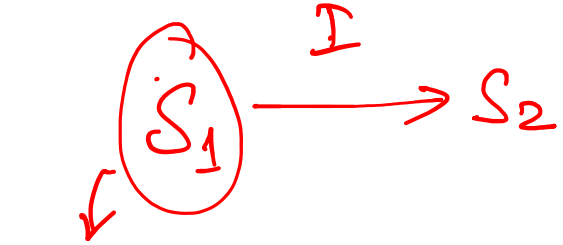
Instruction Set Architecture

- Instruction set architecture is the structure of a computer that a machine language programmer must understand to write a correct (timing independent) program for that machine.

- The instruction set architecture is also the machine description that a hardware designer must understand to design a correct implementation of the computer.

Instruction



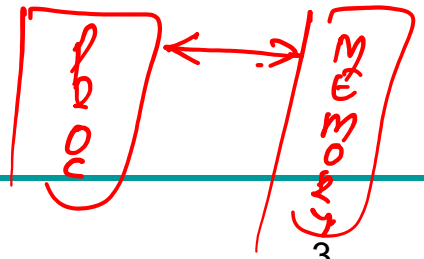


Memory ↑
internal
programmer
visible register

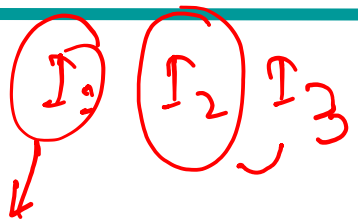
→ Arithmetic / logical

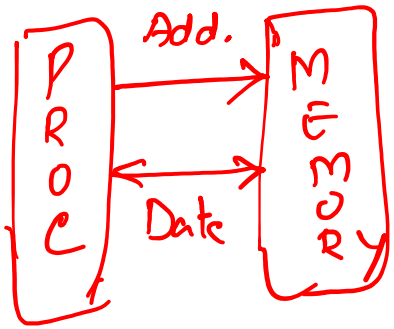
→ Memory communication.

→ Control flow change.



Instruction





Register ⇒ 8 registers

3 bits to address

(1) Arithmetic & logical ⇐ operands & Dest ⇓ Registers

16 bit

4	3	3	3
OPERATION	OPR1	OPR2	DEST

ADD R1, R2, R3 — $R_3 = R_1 + R_2$

AND R1, R2, R3 — $R_3 = R_1 \& R_2$

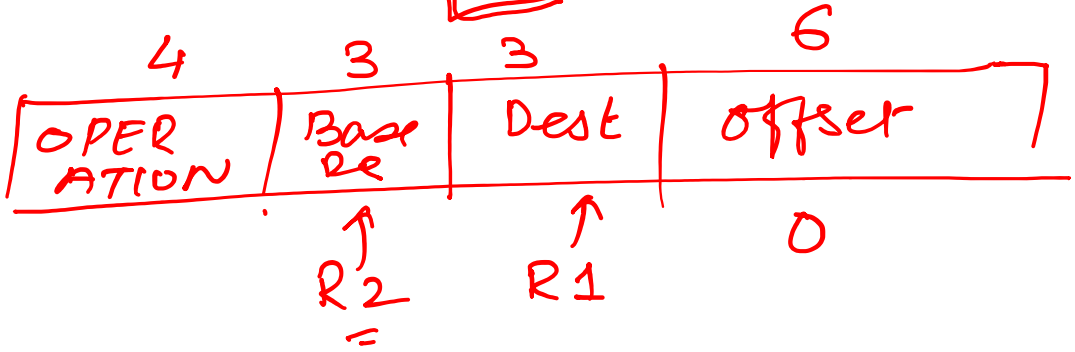
(10) 10 operations ⇒ 4 bit $\lceil \log_2 10 \rceil$

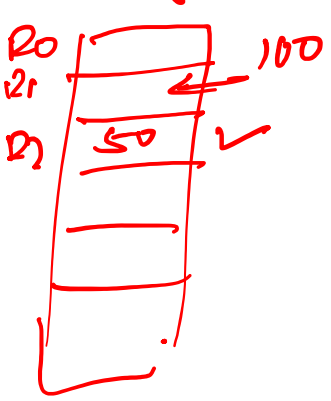
LW Load reg, memory loc. A(R)

Base + displacement

reg

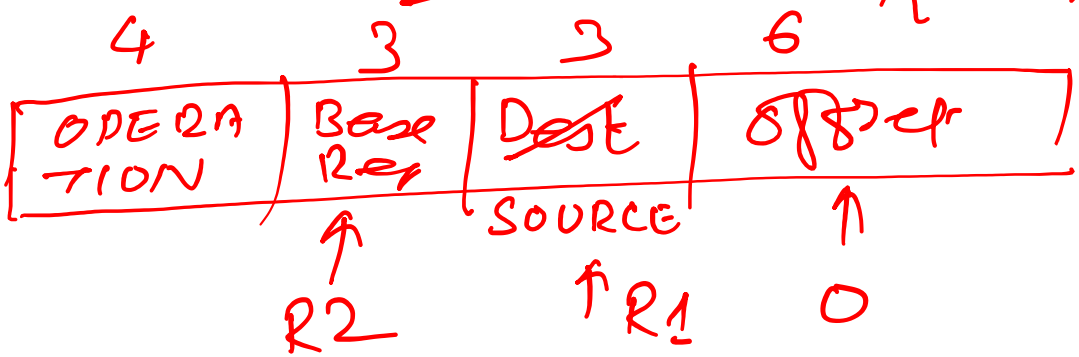
Load R1, (R2) D





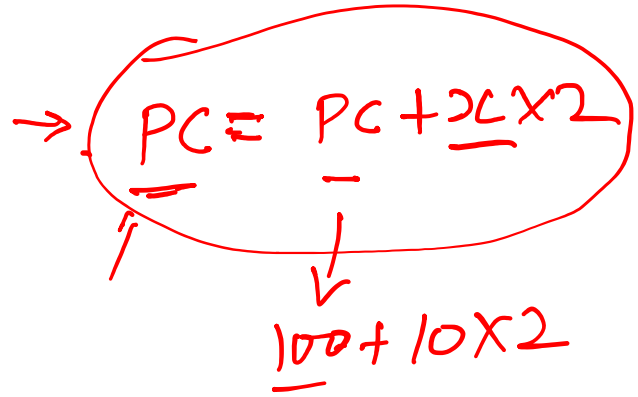
Store R1, (R2), D

M(R2+0) ← R1



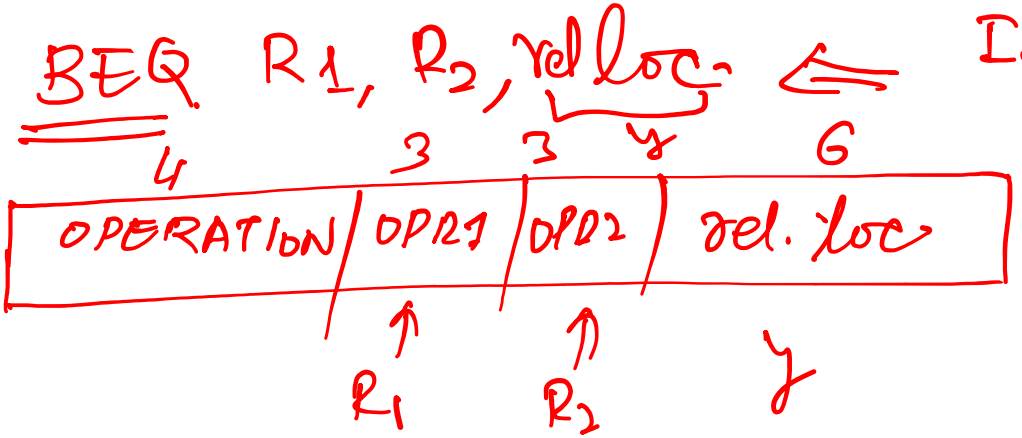
Branch

Unconditional

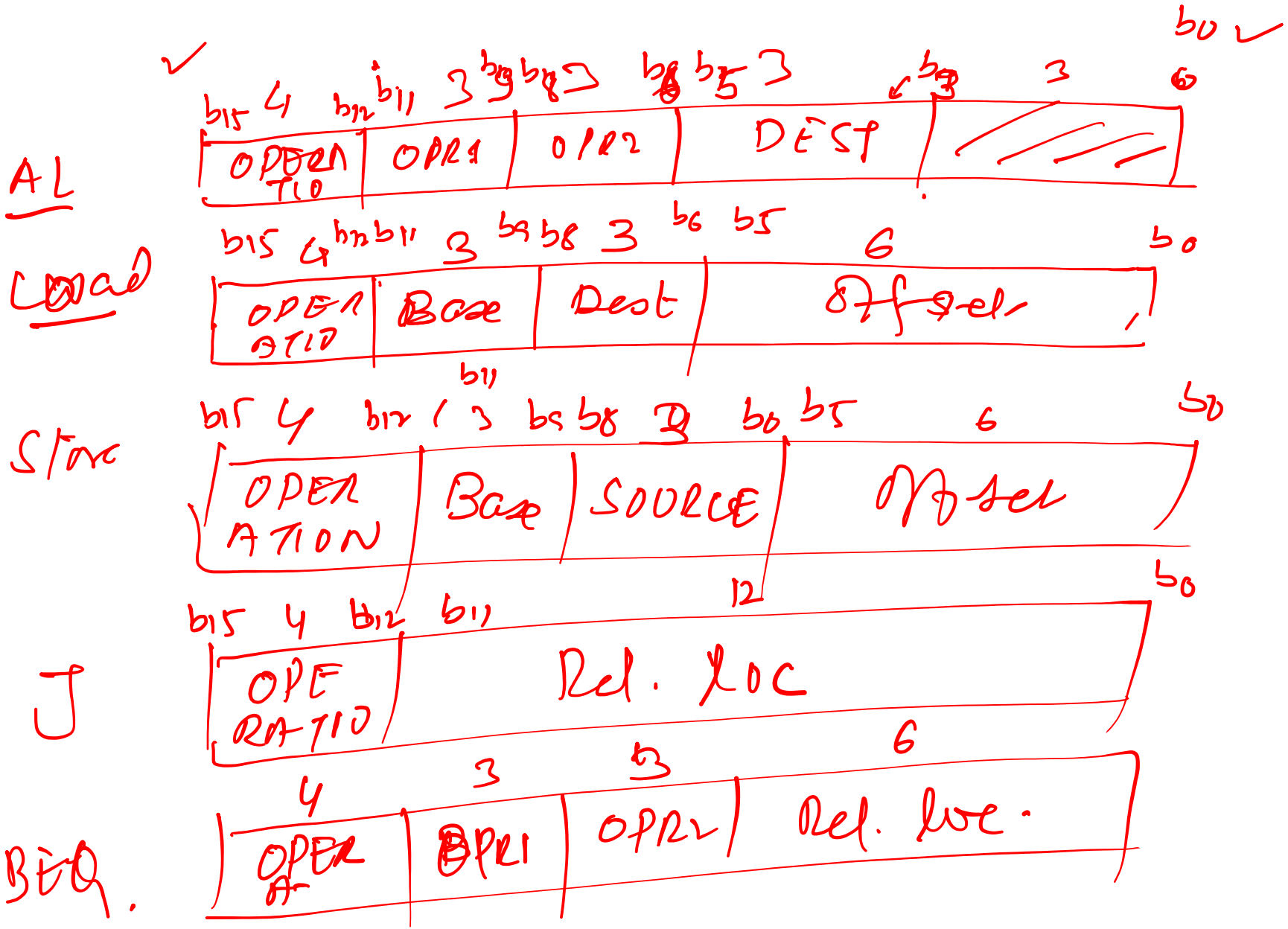


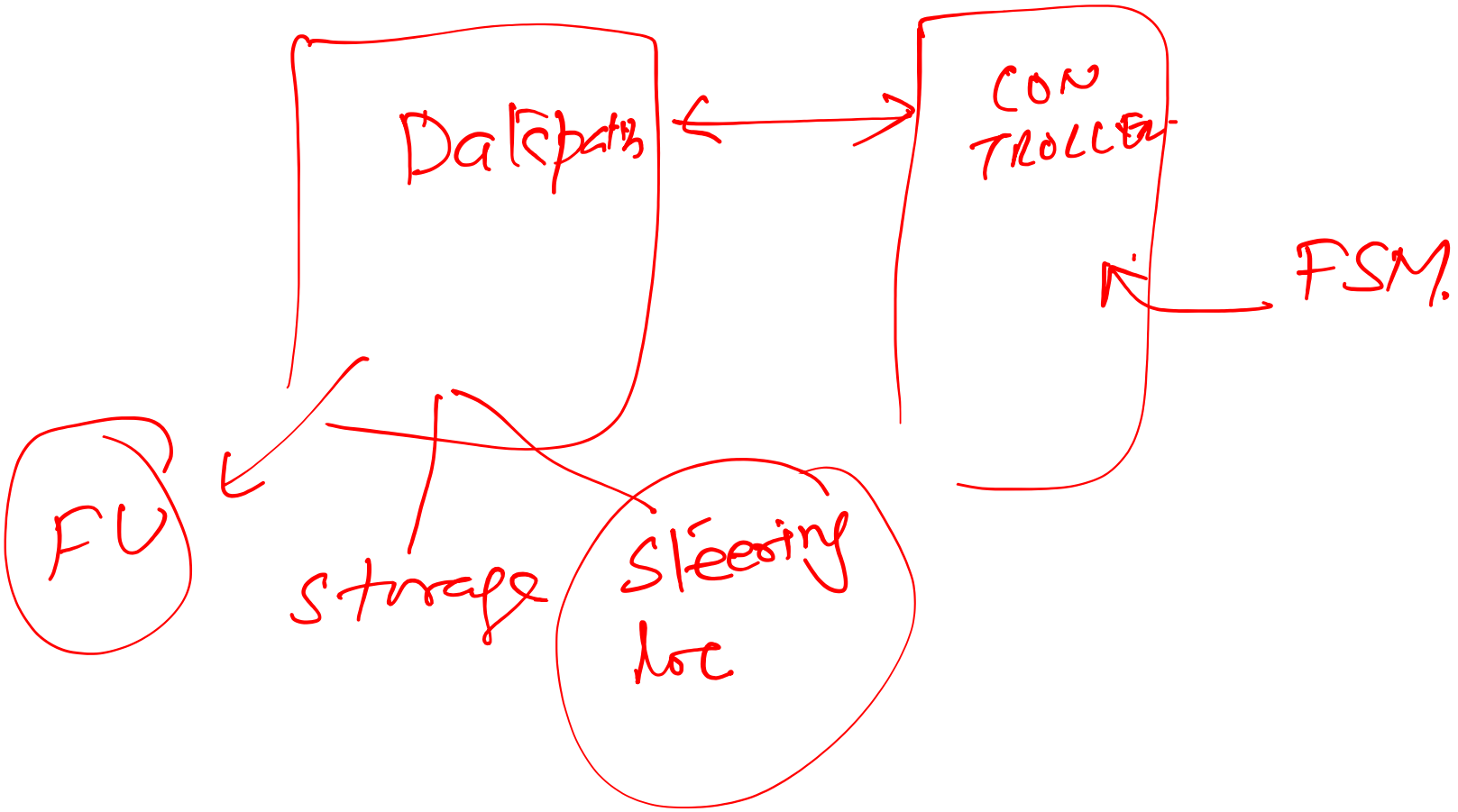
Conditional branch

if (i == j) then

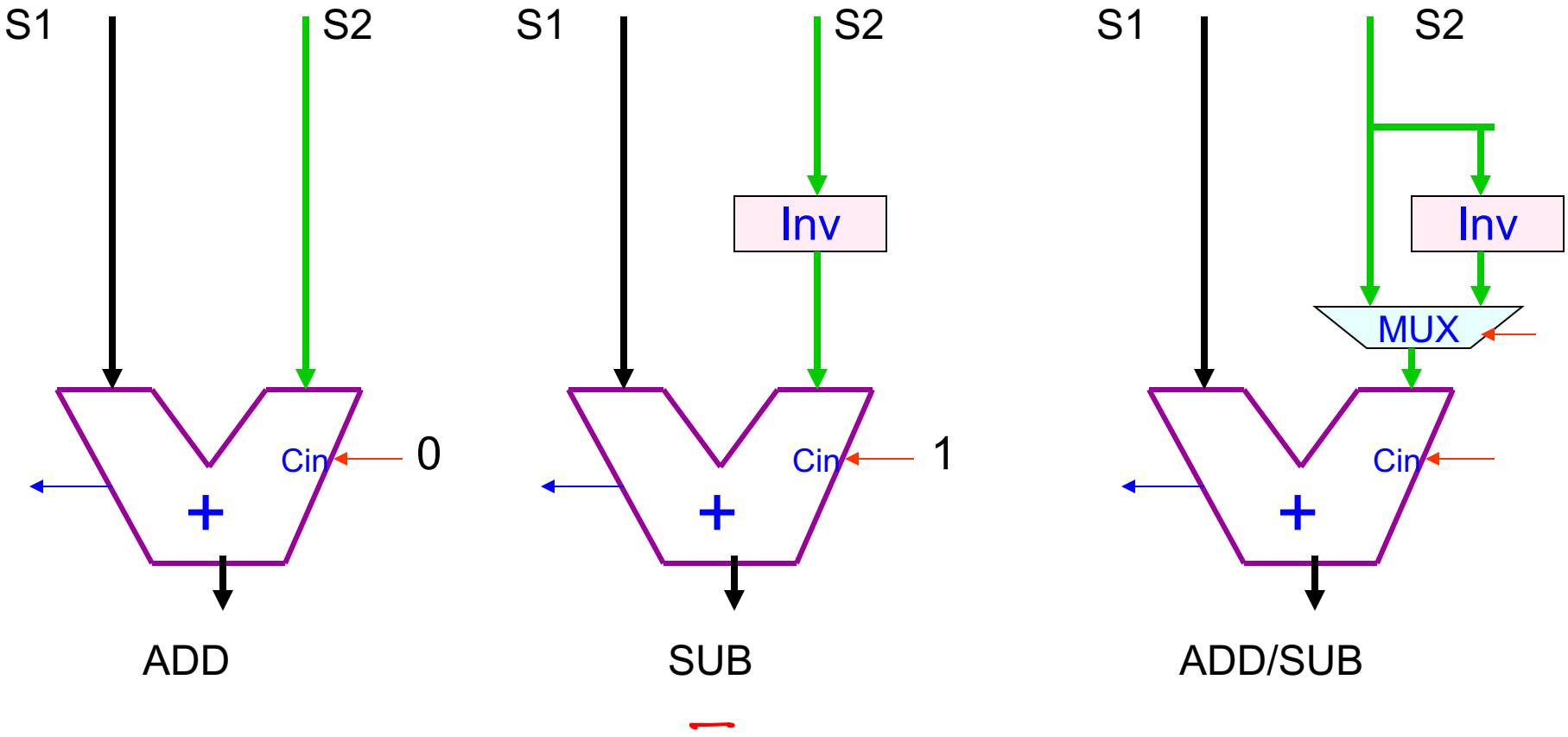


If ($R_1 == R_2$) then
PC = PC + yX2
else
PC = PC + 2

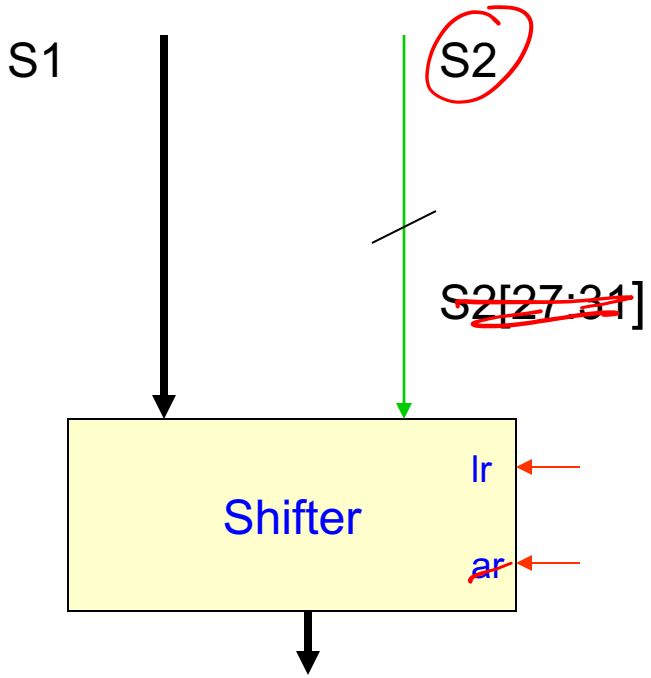

AL
OPERATION
OPR1
OPR2
DEST
Load
OPERATION
Base
Dest
Offset
Store
OPERATION
Base
SOURCE
Offset
J
OPERATION
Rel. Loc
BEQ.
OPERATION
OPR1
OPR2
Rel. loc.


Datapath
CON TROLLER
FSM
FU
straps
Sleeping loc

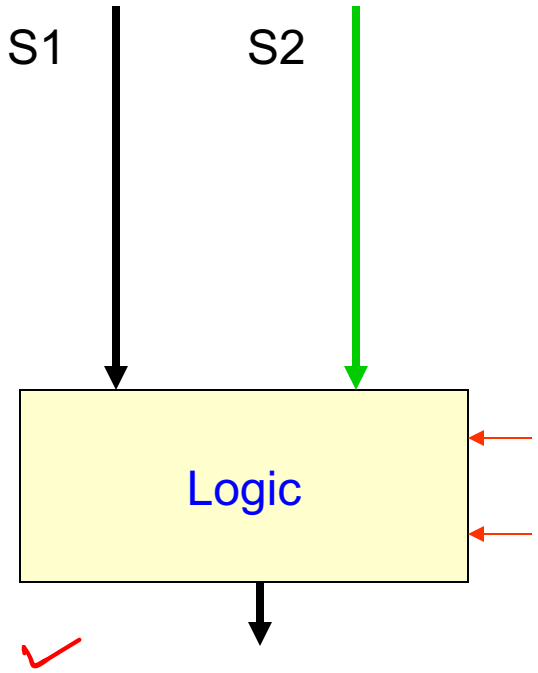
Arithmetic Logic Unit

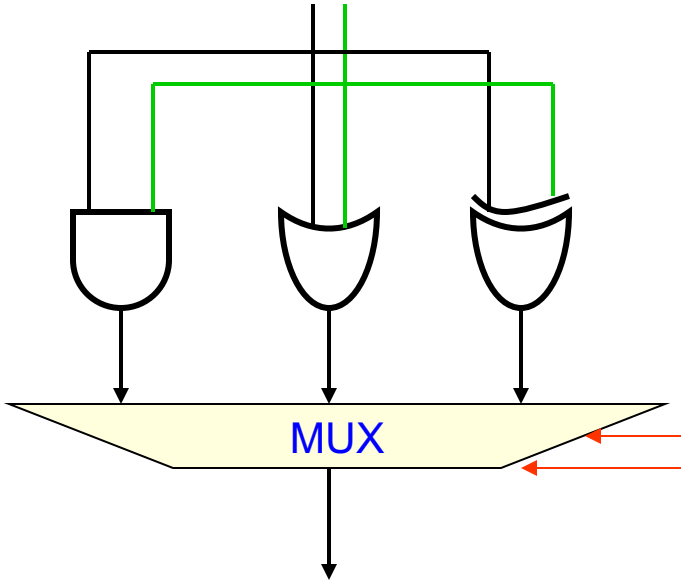


Arithmetic Logic Unit

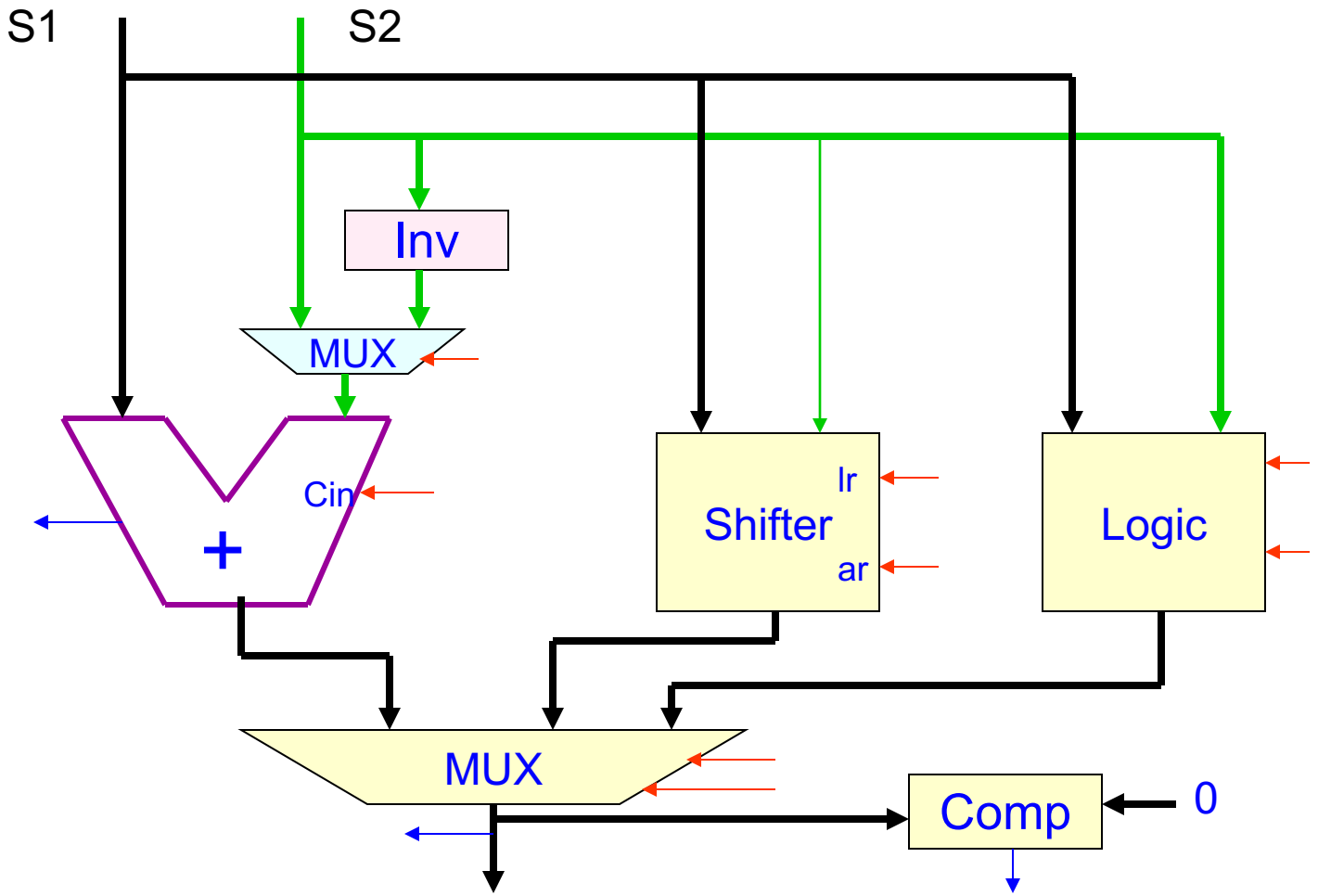


Arithmetic Logic Unit

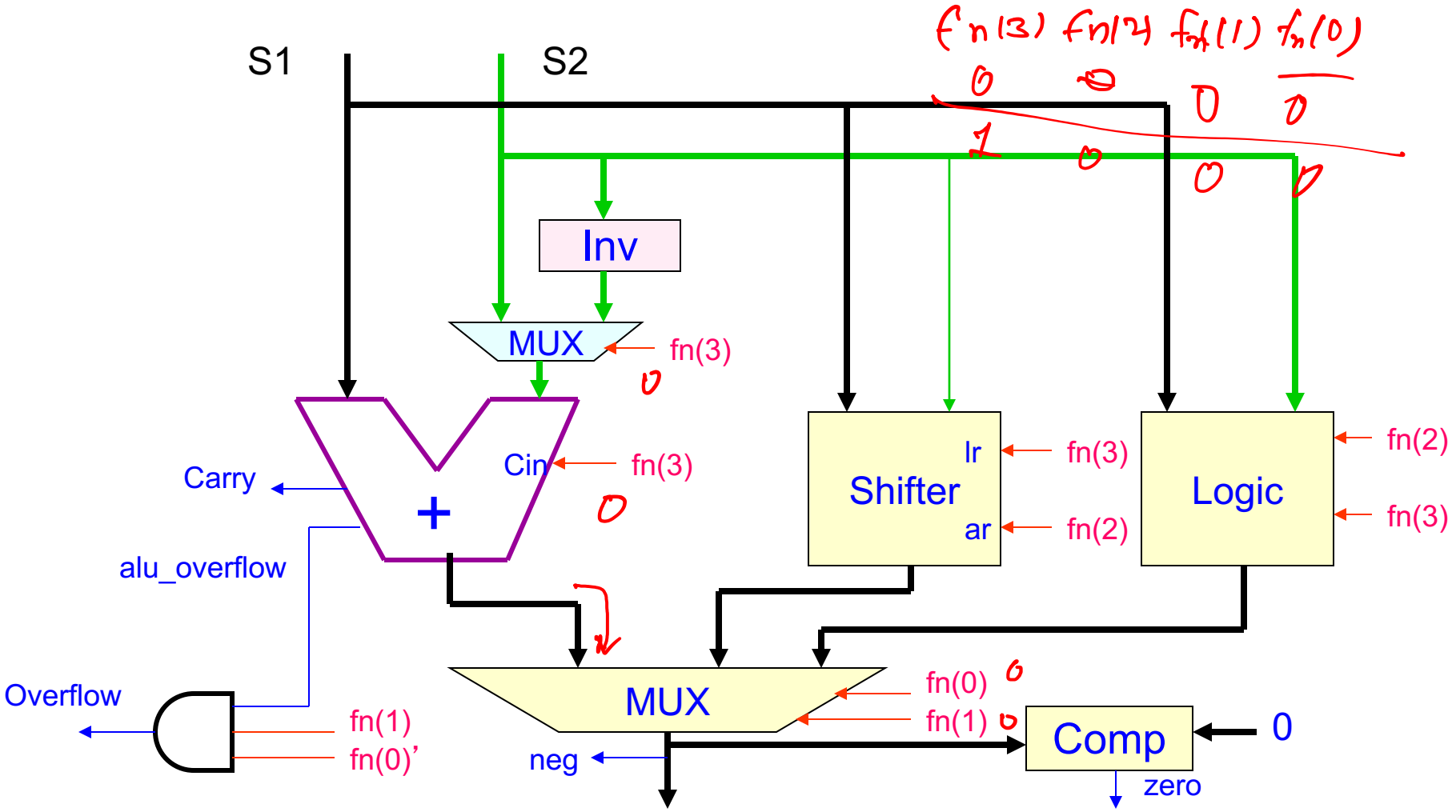




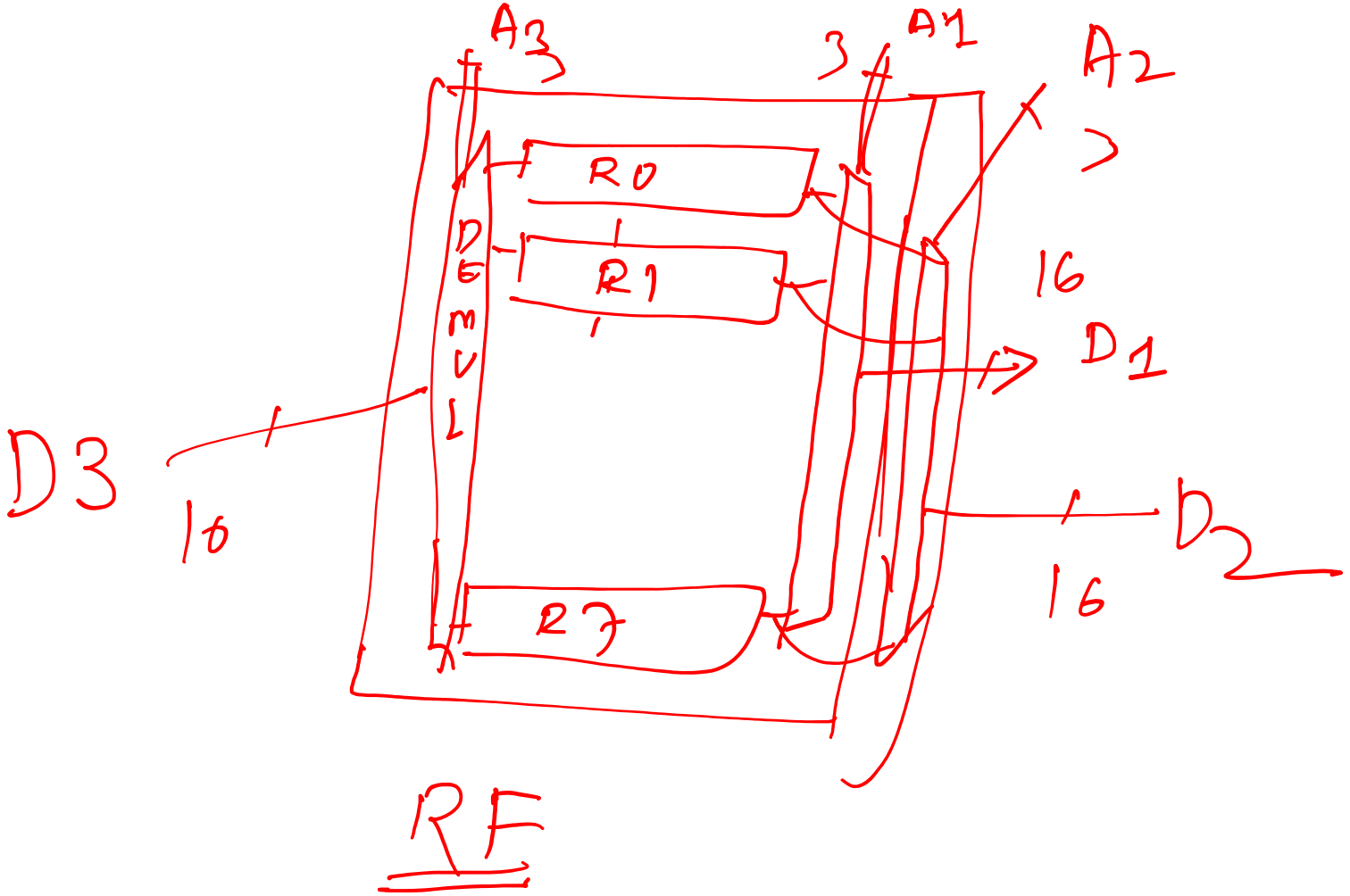
Arithmetic Logic Unit



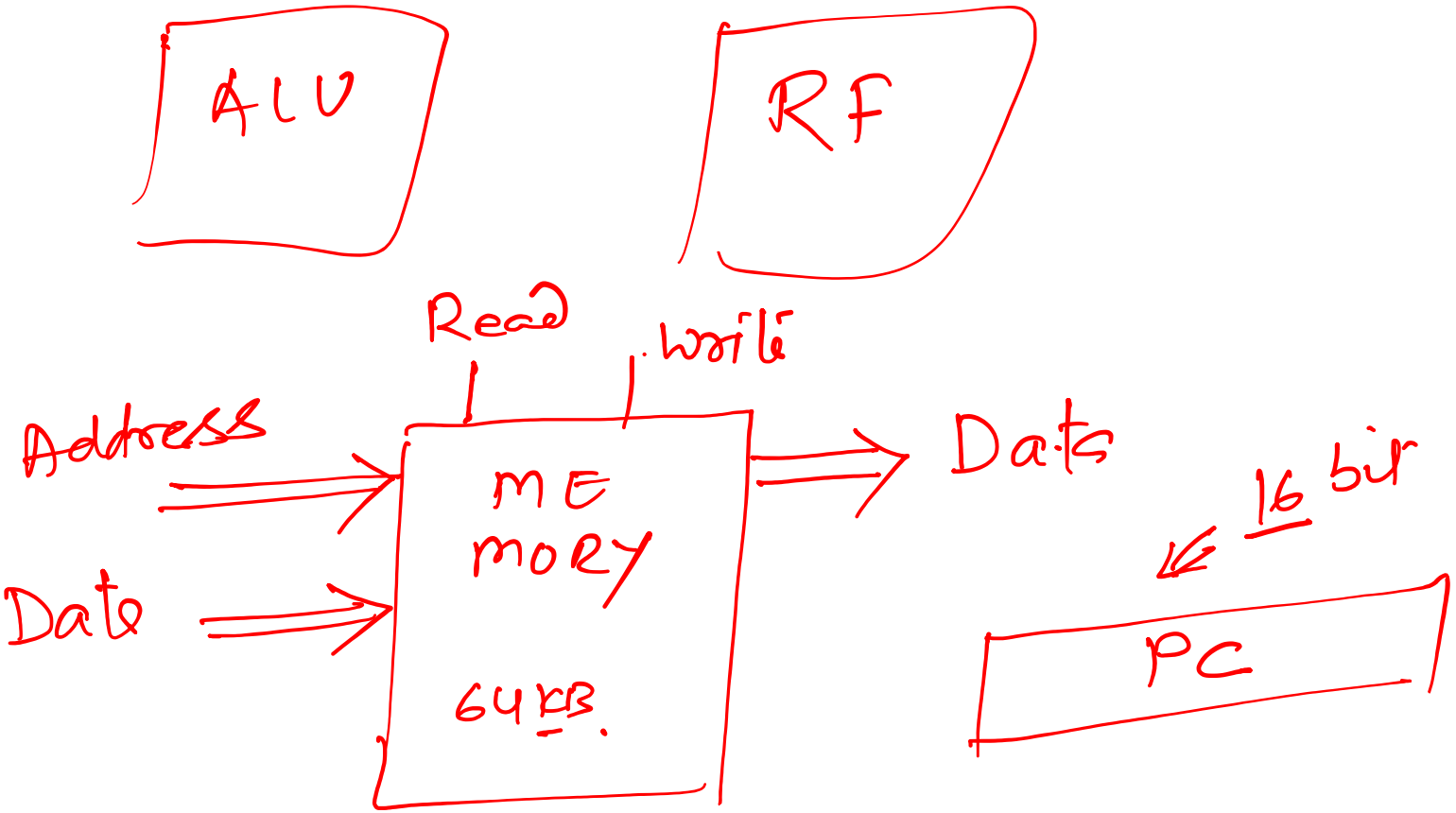
Arithmetic Logic Unit



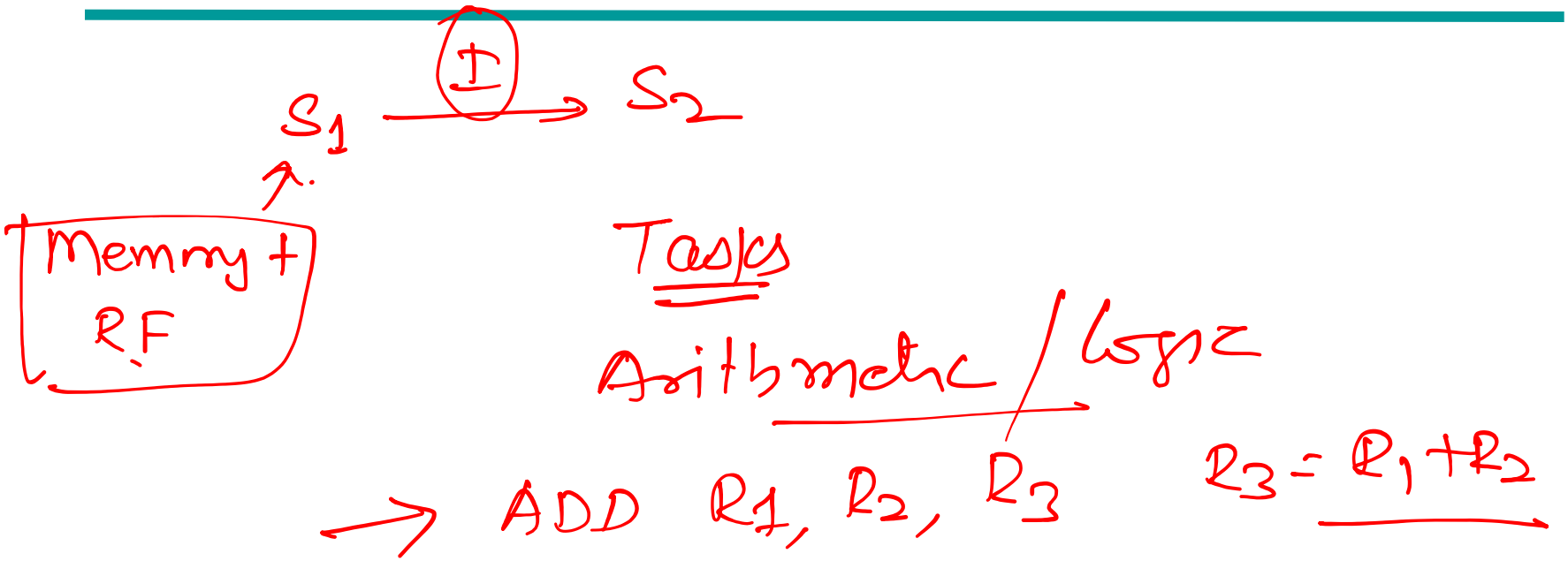
Instruction



Instruction



Instruction



Instruction

AL Instruction

ADD R_1, R_2, R_3

1. Bring instruction from memory (PC) ⇐ S1
2. Understand instruction ⇐ S2

OPERATION	OPR1	OPR2	DEST	///
4	3	3	3	

3. Read operand1 & operand 2 ⇐ S3
4. Compute the instruction (Addition Oper. ⇐ S4)
5. Write the result in R- Dest. reg ⇐ S5
6. update PC PC = PC+2

Instruction

State machine (TASK)

S1

PC → Mem_Add

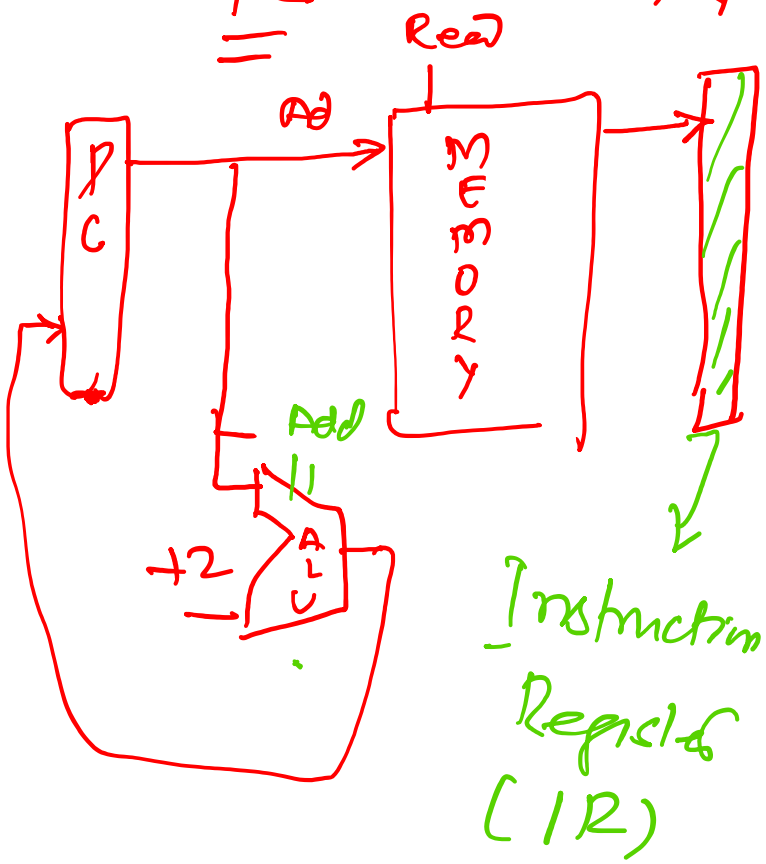
Mem_Data → ir

PC → alu-A

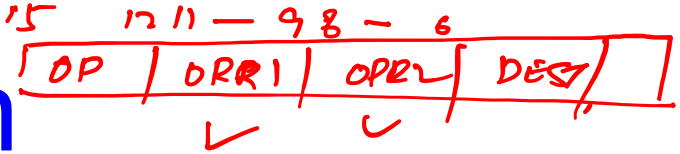
+2 → alu-B

alu-C → PC

- MR
- PC_W
- IR_W
- alu-add



Instruction



S_2^{-1}

t_1 write signal

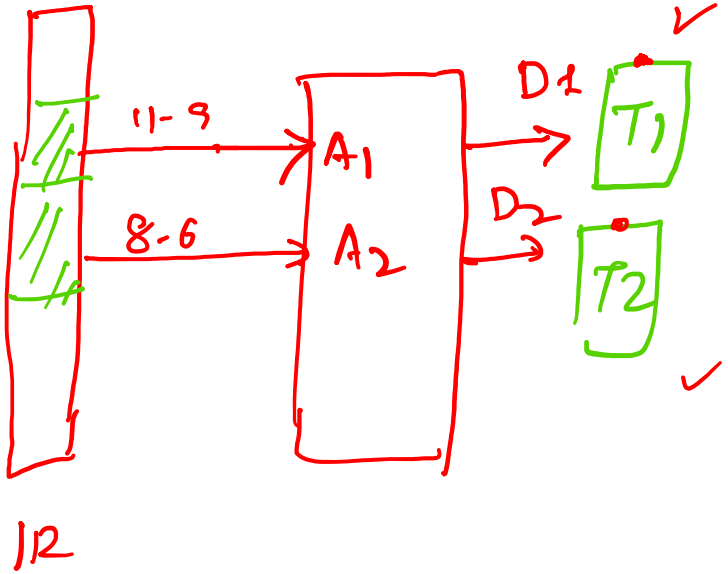
$ir_{11-9} \rightarrow rf\text{-}A_1$

$ir_{8-6} \rightarrow rf\text{-}A_2$

$rf\text{-}D_1 \rightarrow t1$

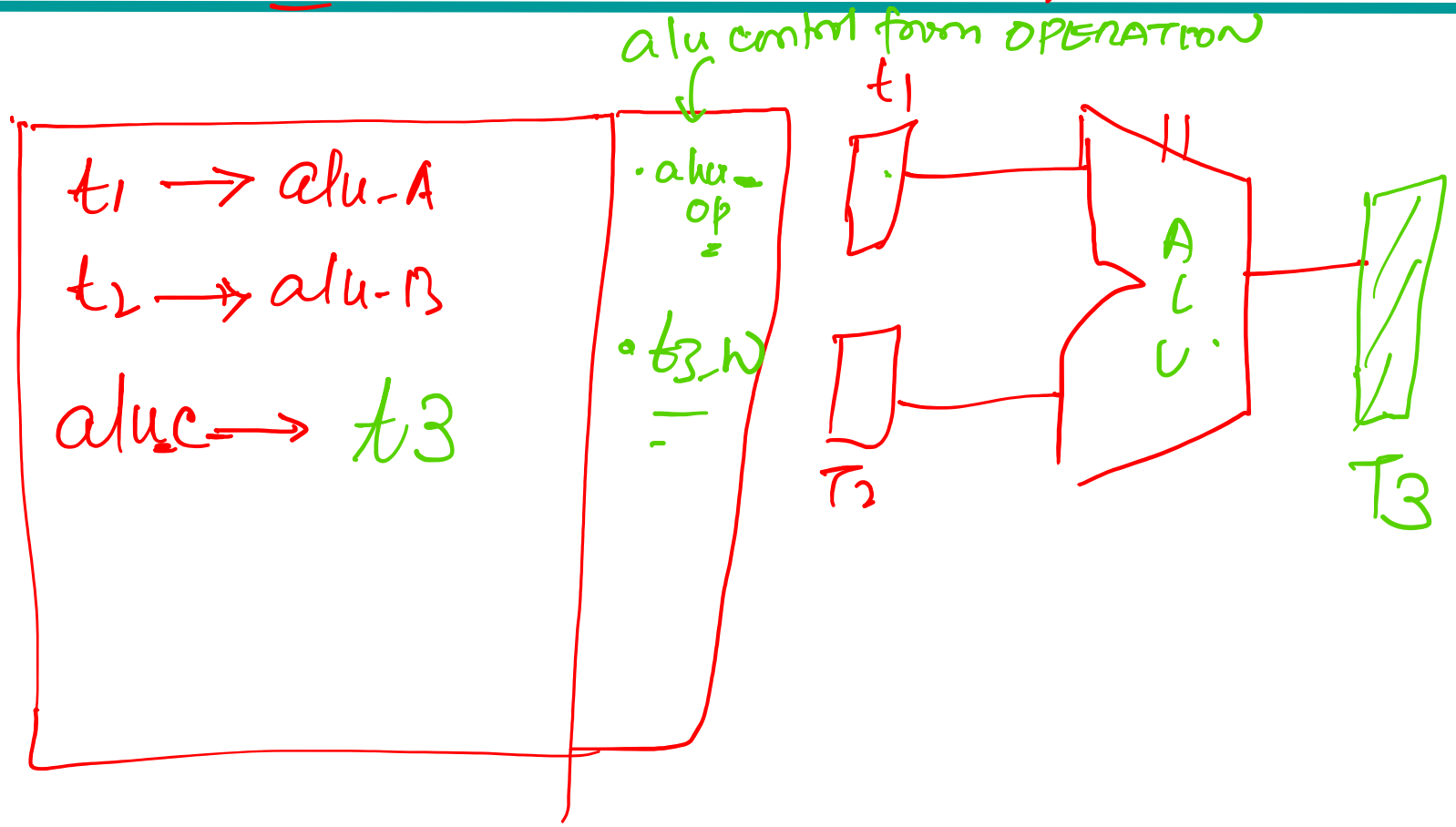
$rf\text{-}D_2 \rightarrow t2$

- t_1-w
- t_2-w



Instruction

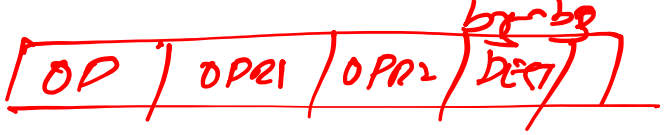
S3 S4 (Execute instruction)

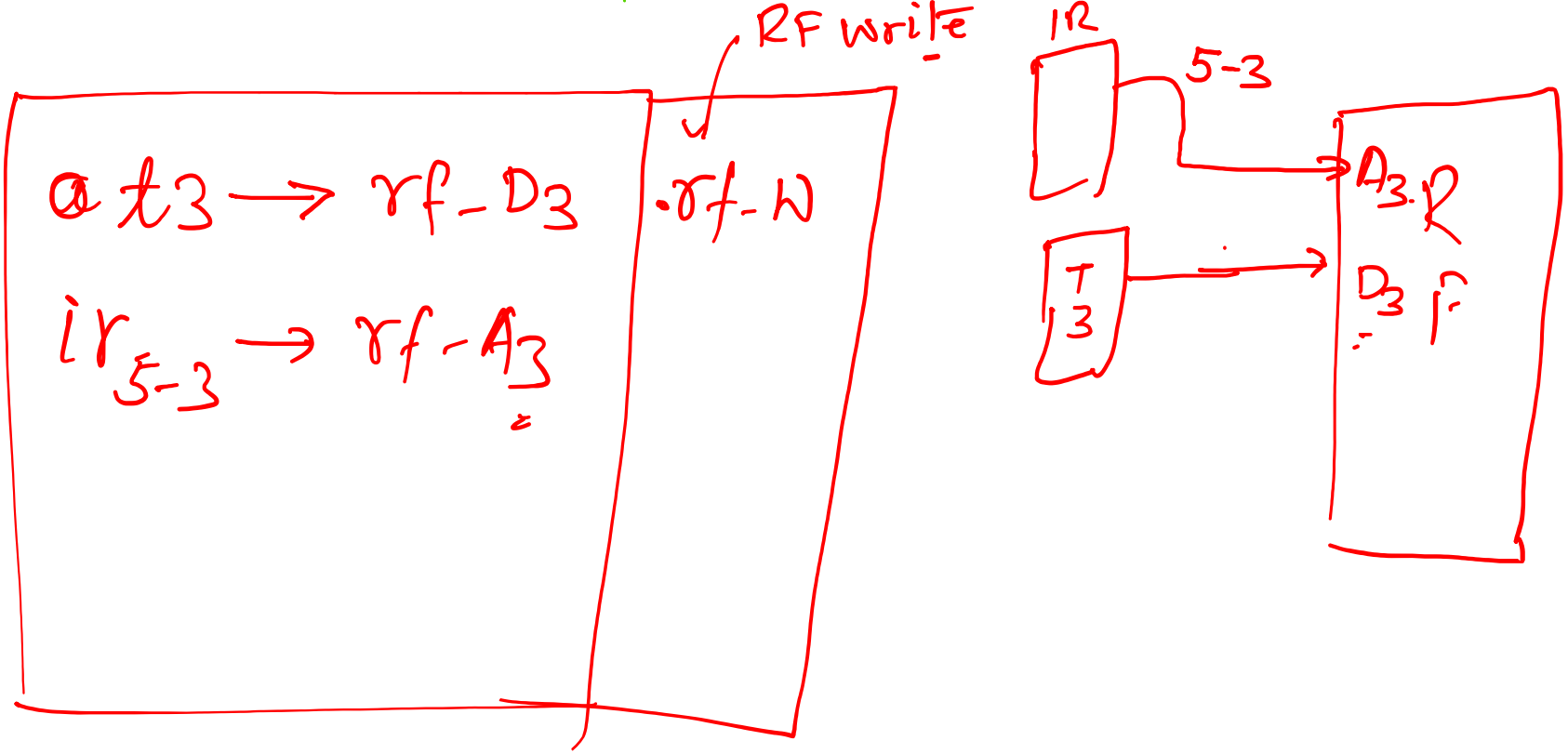


Instruction

S5

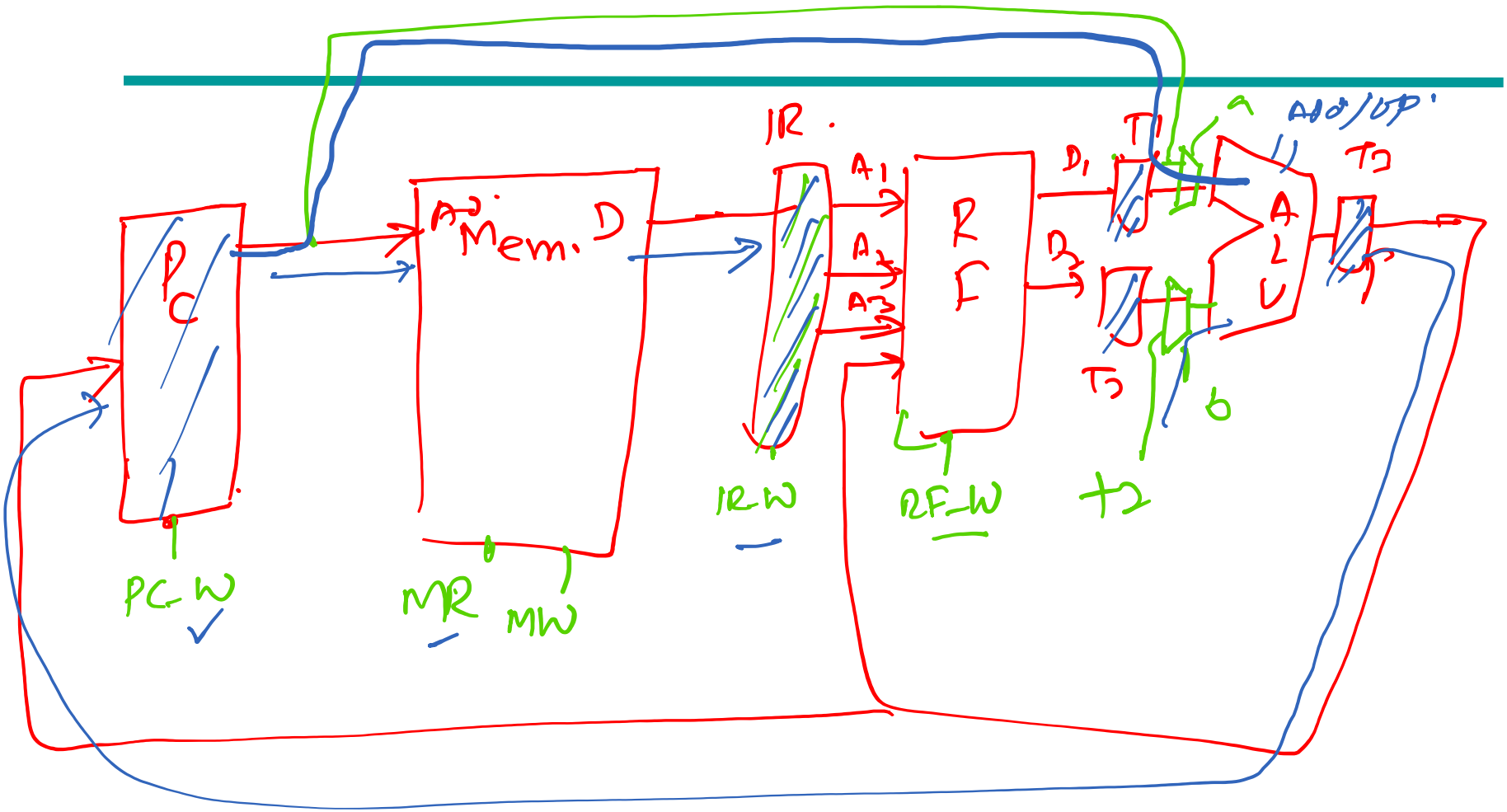
Update the state





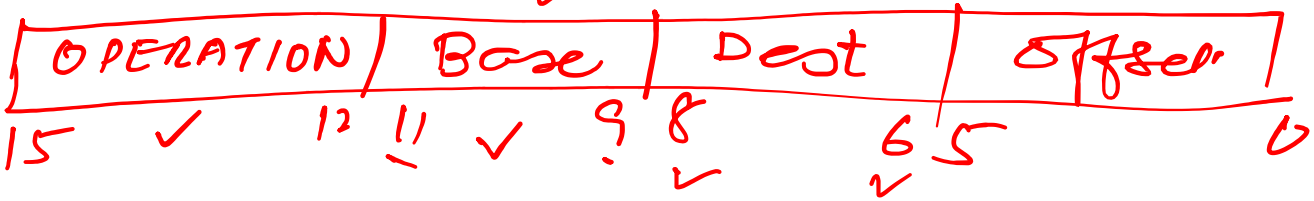
$S_1 \rightarrow S_2' \rightarrow S_4 \rightarrow S_5$

Instruction



Instruction

Load Instruction



SG OPERATIONS

SF ① Read instruction from memory (PC) & update PC.

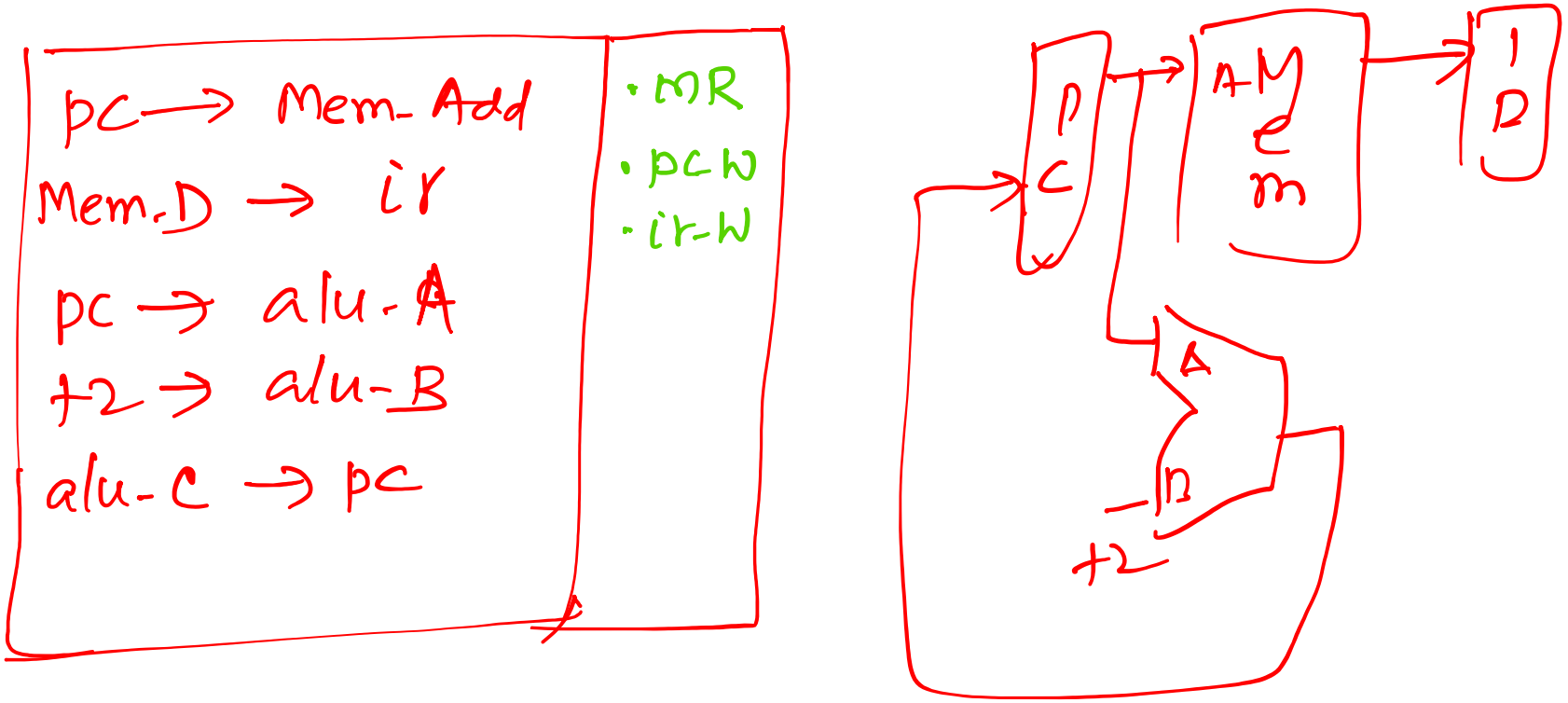
SE ② Understand and read the base address of memory

SD ③ Compute the address of memory (Base add + offset)

SC ④ Read memory

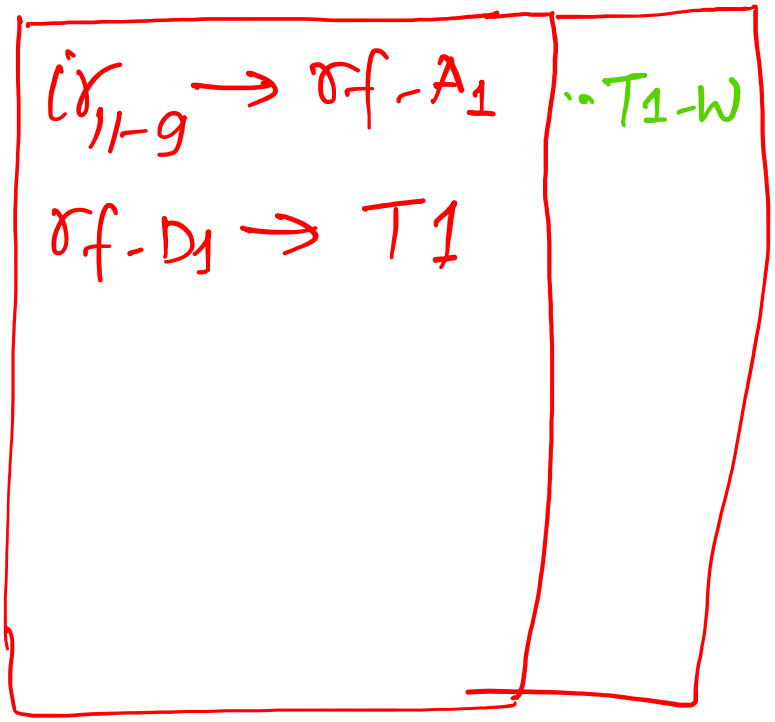
SB ⑤ ~~Write~~ Update the reg.

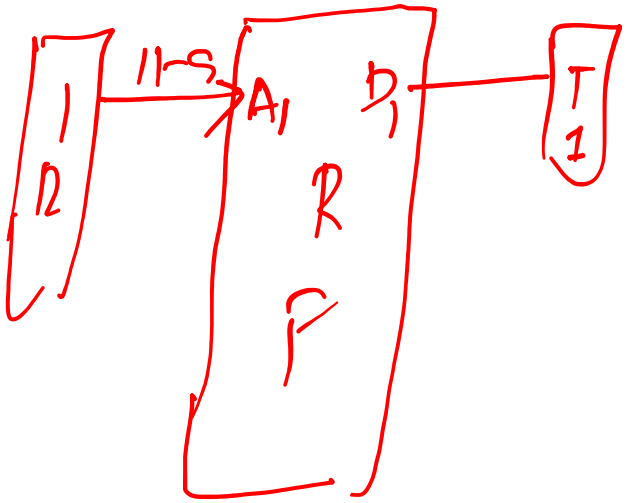
Instruction



Instruction

S7

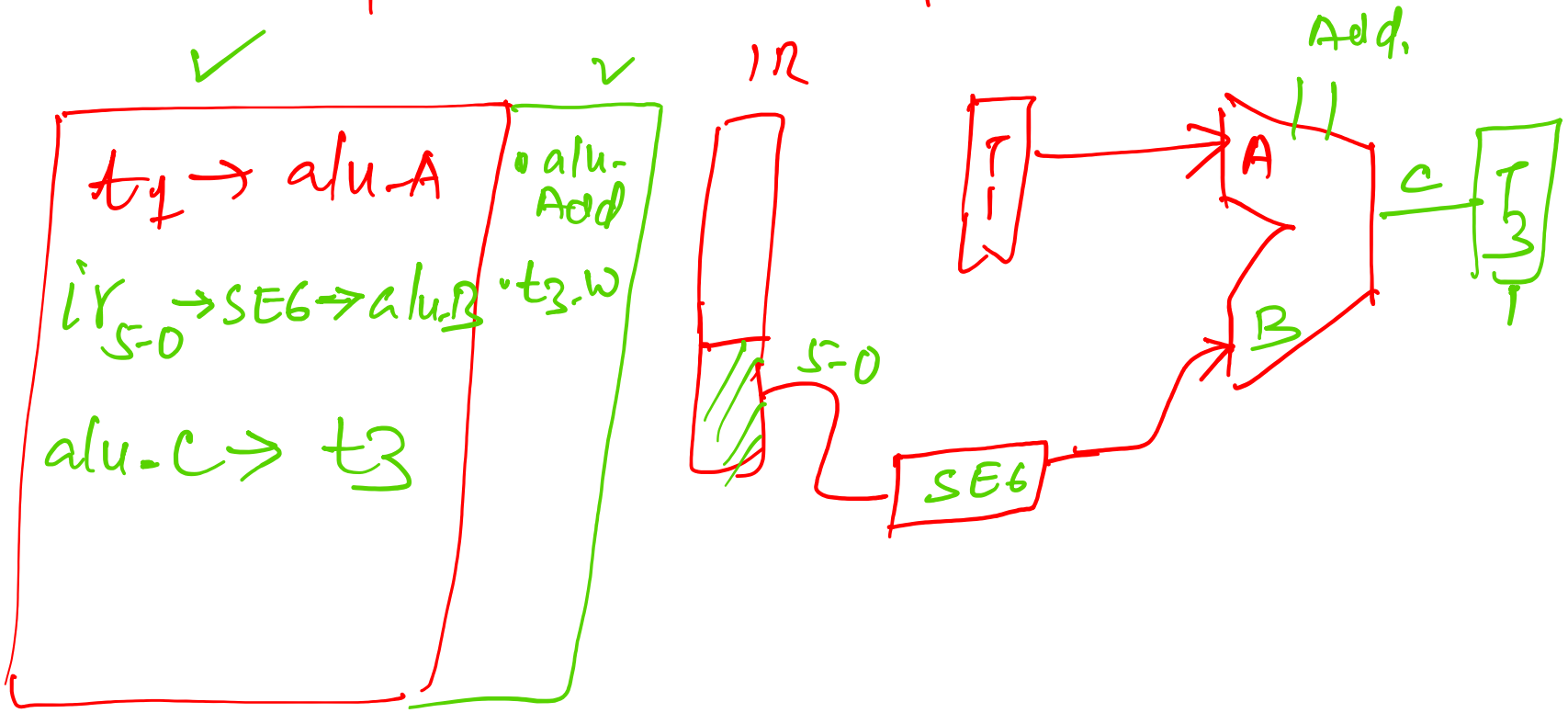




Instruction

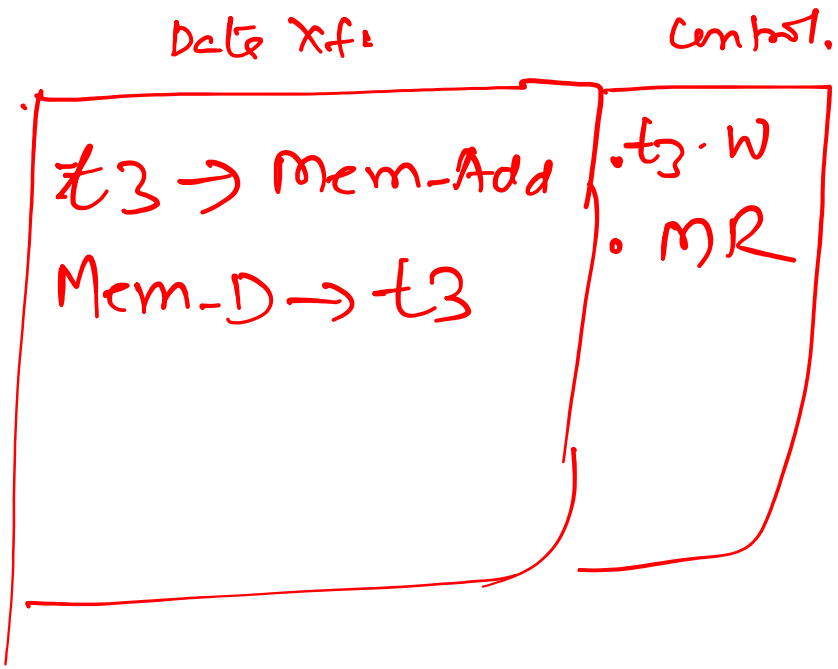
SD

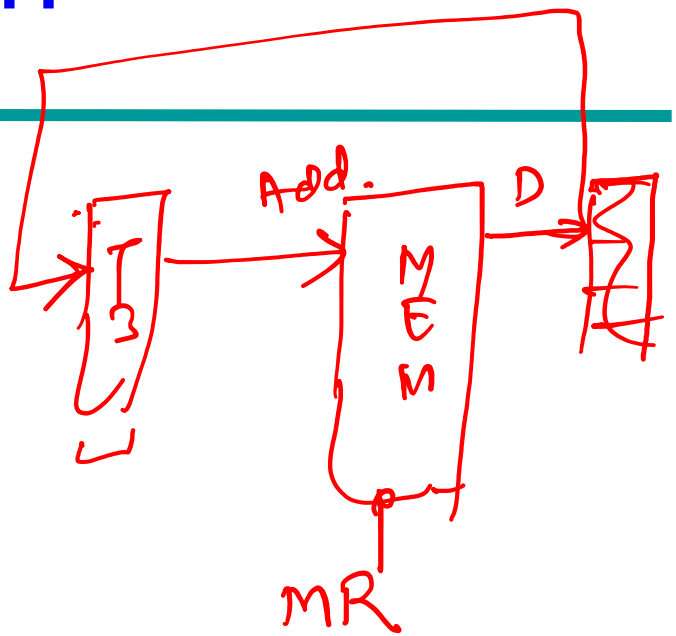
Compute the address of memory



Instruction

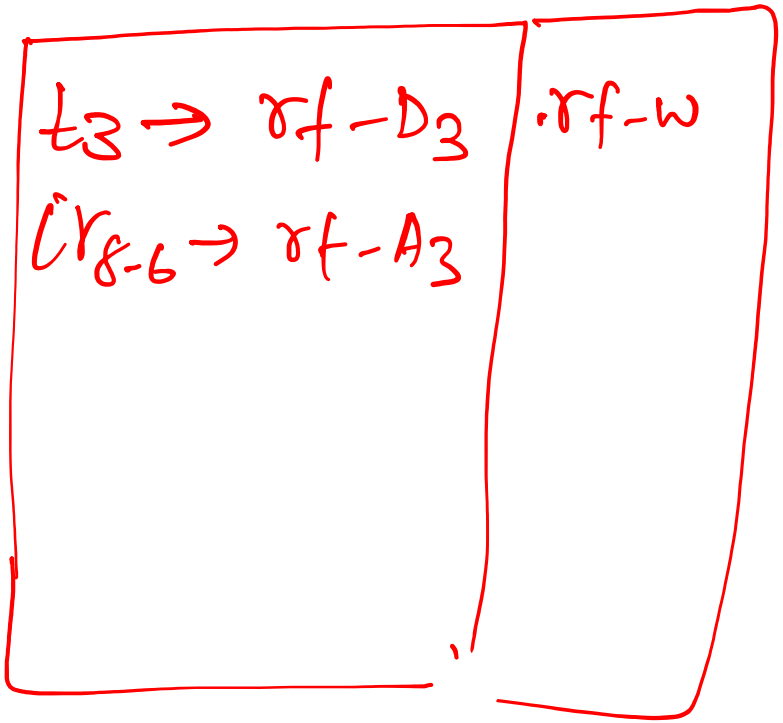
Eg Memory Read.

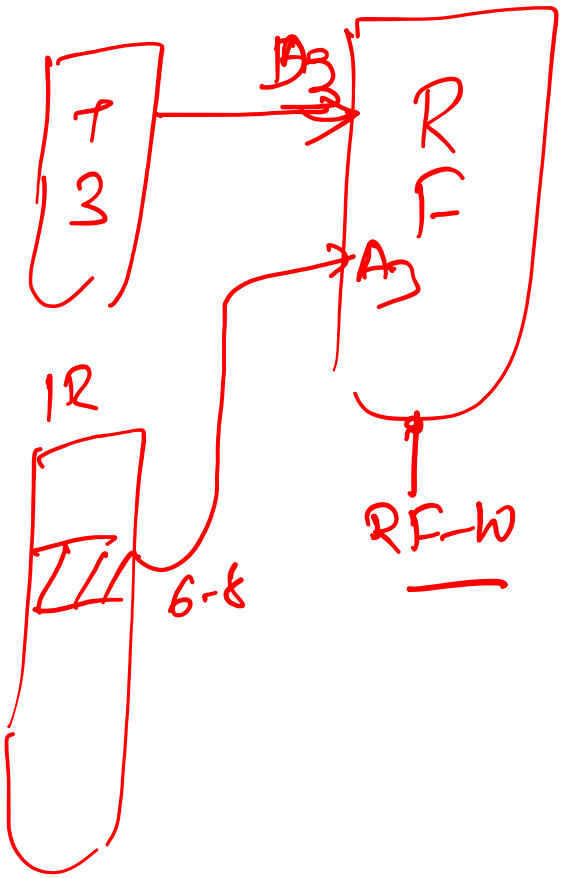




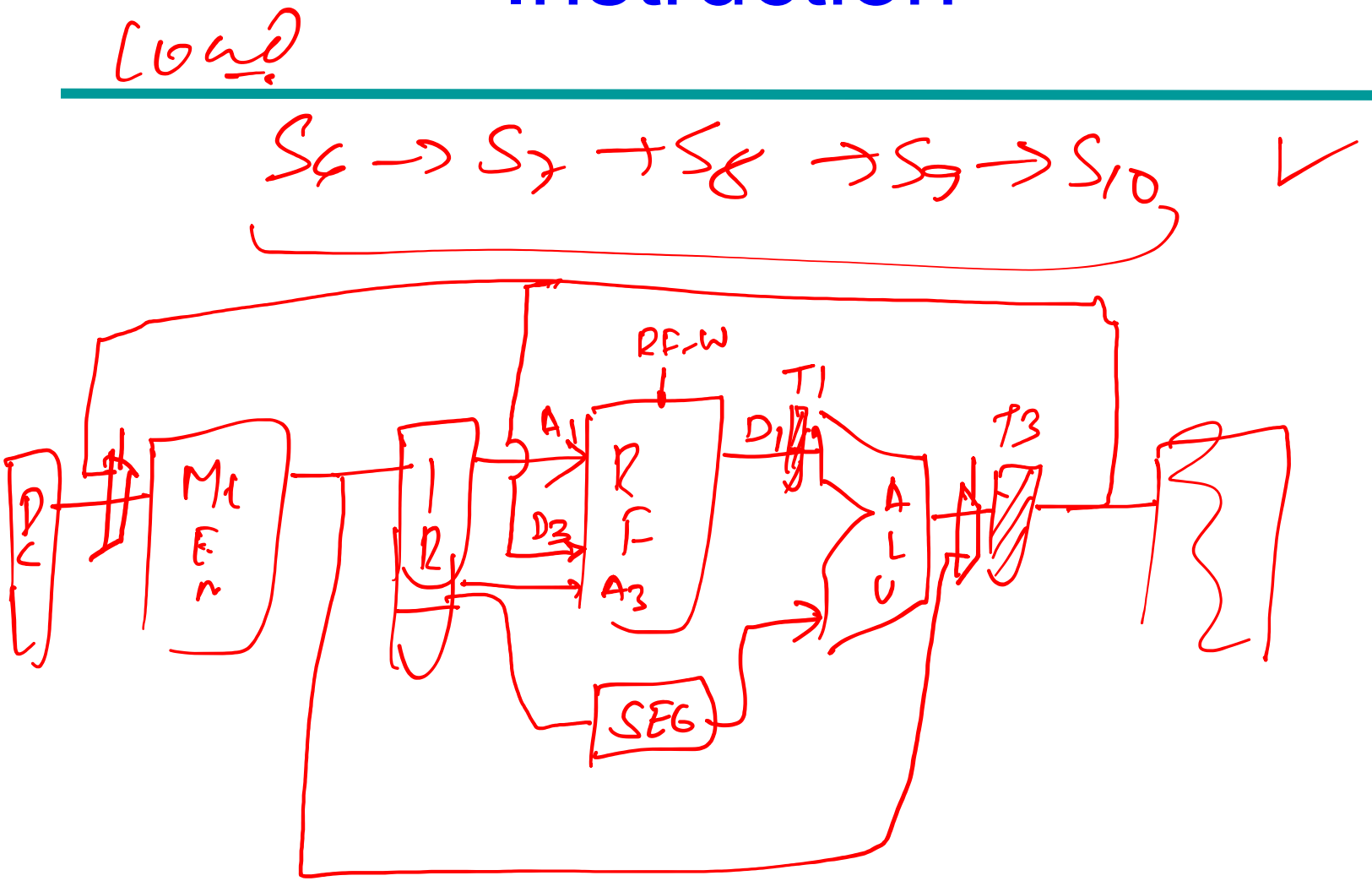
Instruction

S10 Update RF

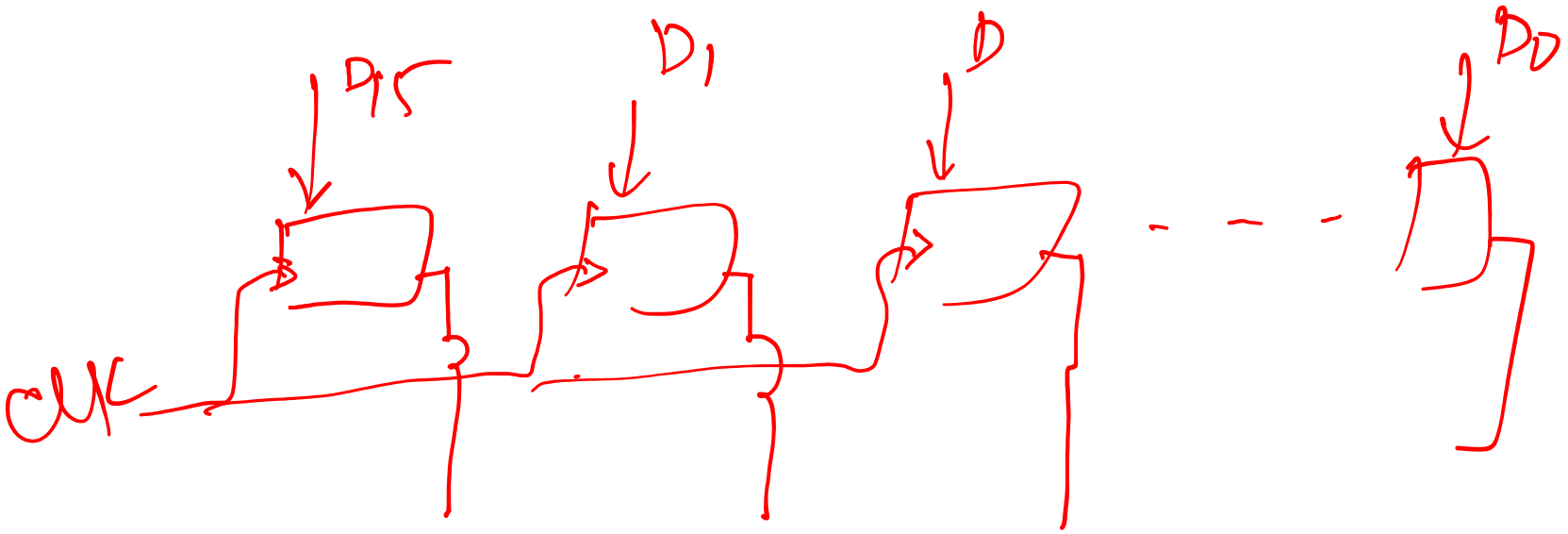




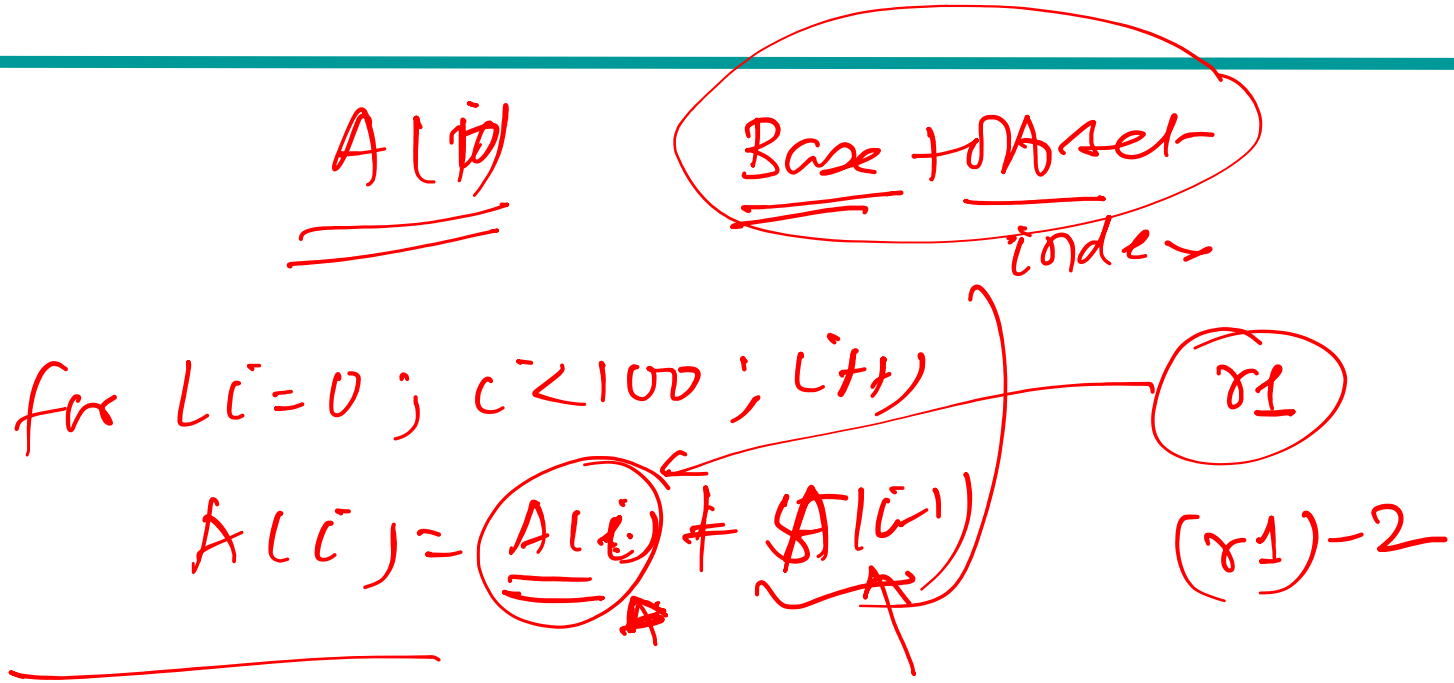
Instruction



Instruction



Instruction



Thank You