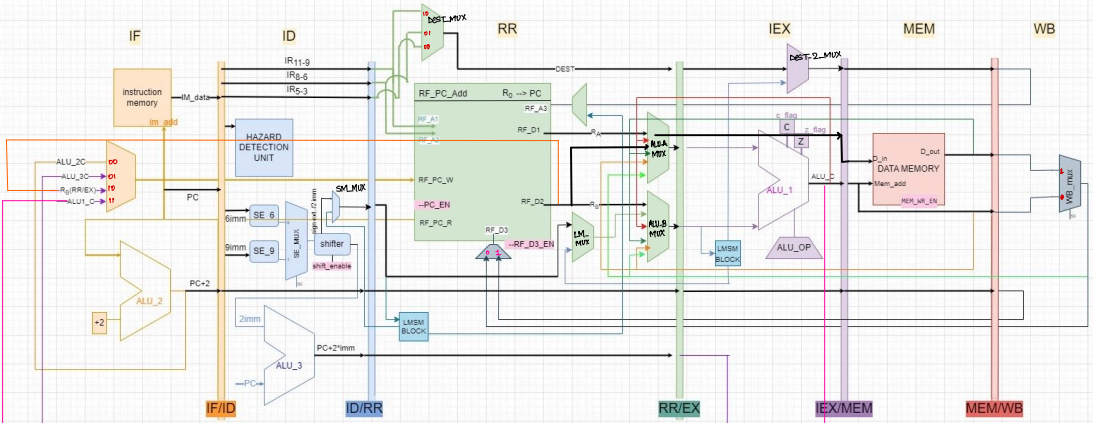
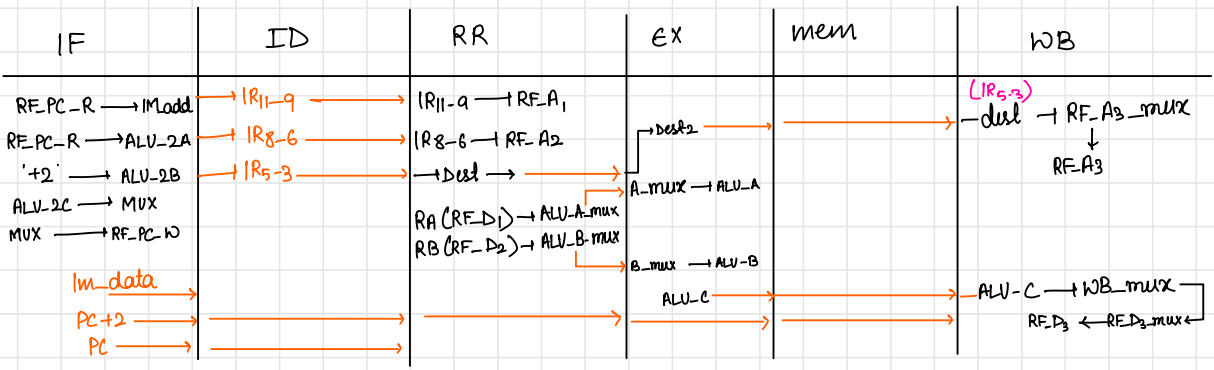
ADD/NAND	0001/0010	RA + RB ⟶ RC (Dest)
ADI	0000	RA + 6imm ⟶ RB (")
LLI	0011	0 + 9imm ⟶ RA (")
LW	0100	M (RB + 6imm) ⟶ RA (Dest)
SW	0101	M (RB + 6imm) ⟵ RA (Din)
BEQ	1000	} if Branch PC' = PC + 2imm
BLT	1001	
BLE	1010	
JAL	1100	: Dest RA ↶ (PC+2) ; PC + 2imm
JLR	1101	: Dest RA ↶ (PC + 2) ; RB
JRI	1111	: to RA + 2imm


IF
ID
RR
IEX
MEM
WB
instruction memory
IM_data
im_add
HAZARD DETECTION UNIT
IR11-9
IR8-6
IR5-3
DEST_MUX
DEST
RF_PC_Add
R0 --> PC
RF_A3
RF_D1
RF_A1
RF_A2
RF_PC_W
--PC_EN
RF_PC_R
RF_D2
RF_D3
--RF_D3_EN
RA
RB
ALU-A MUX
ALU-B MUX
LM_MUX
LMSM BLOCK
ALU_OP
ALU_1
ALU_2
ALU_3
c_flag
z_flag
C
Z
ALU_C
DEST_2_MUX
D_in
D_out
DATA MEMORY
Mem_add
MEM_WR_EN
WB_mux
ALU_2C
ALU_3C
R7(RR/EX)
ALU1_C
PC
PC+2
+2
SE_6
SE_9
6imm
9imm
SE_MUX
SIGN_EXT_2 imm
SM_MUX
shifter
shift_enable
2imm
PC+2*imm
--PC
IF/ID
ID/RR
RR/EX
IEX/MEM
MEM/WB

0001 { ADD, ADC, ADZ, ACA, ACC, ACZ (0000 0000 0000 0010 0010 0010) ; ANC, ACW (0001 0011) }

alu_op

NDU, NDC, NDZ (0100) ; NCU, NCC, NCZ (0101) } 0010



MUX: 00
PC_EN=1

SH_EN=0
SM_MUX=0
SE_MUX=0

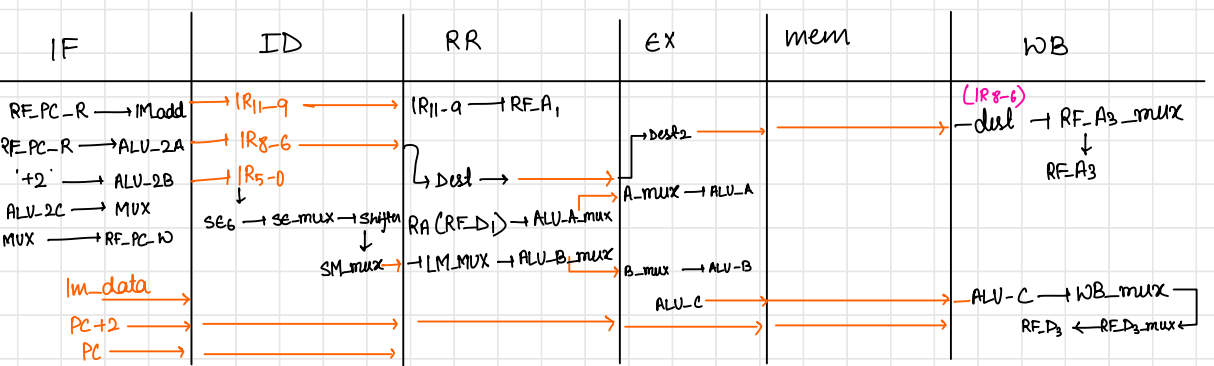
DEST_MUX=00
LM_MUX=0
ALU_A_MUX=000
ALU_B_MUX=000

DEST_2_MUX=0
ALU_C flag = 1
ALU_2 flag = 1

MEM_WR_EN=0

WB_MUX=0
RF_D3_MUX=0
RF_D3_EN=1
RF_A3_MUX=0

0000 ADI ALU_op: 0000 (add) (Ra + 9mm6) → Rb



MUX: 00
PC_EN=1

SH_EN=0
SM_MUX=0
SE_MUX=0

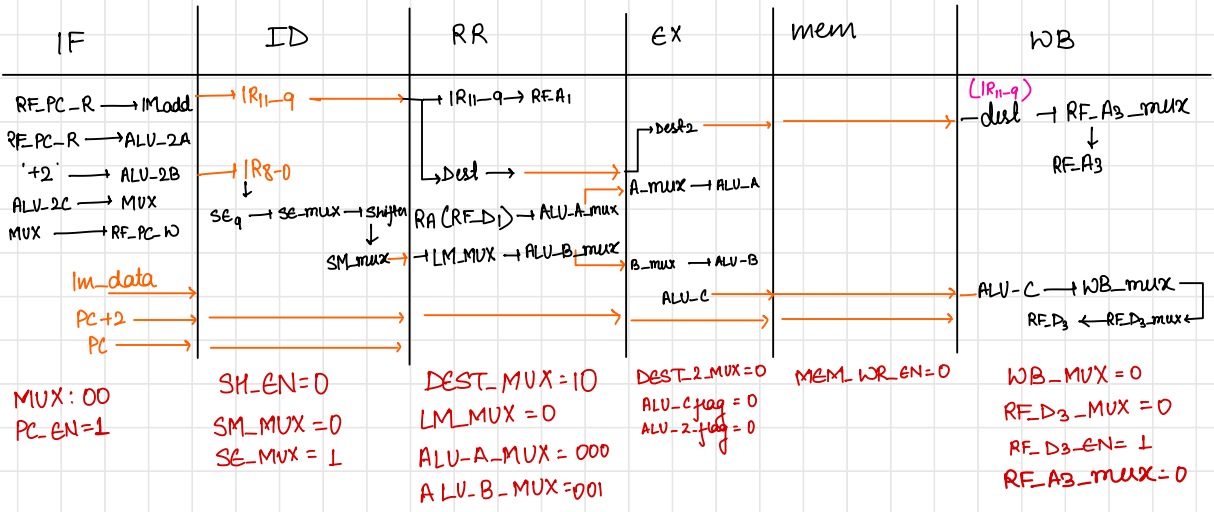
DEST_MUX=01
LM_MUX=0
ALU_A_MUX=000
ALU_B_MUX=001

DEST_2_MUX=0
ALU_C flag = 1
ALU_2 flag = 1

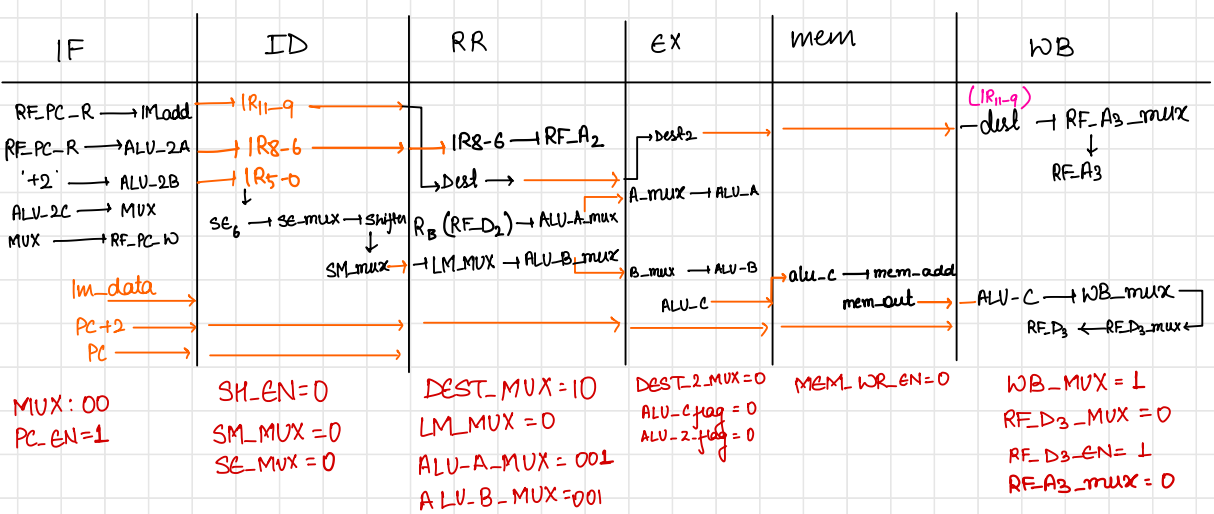
MEM_WR_EN=0

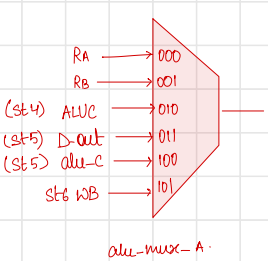
WB_MUX=0
RF_D3_MUX=0
RF_D3_EN=1
RF_A3_MUX=0

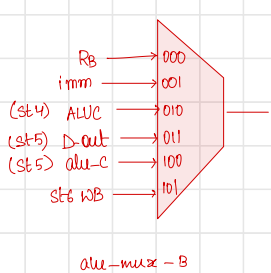
0011 LLI ALU_op : 1001 (LLI) : just returns ALU_B LLI ra, Imm9



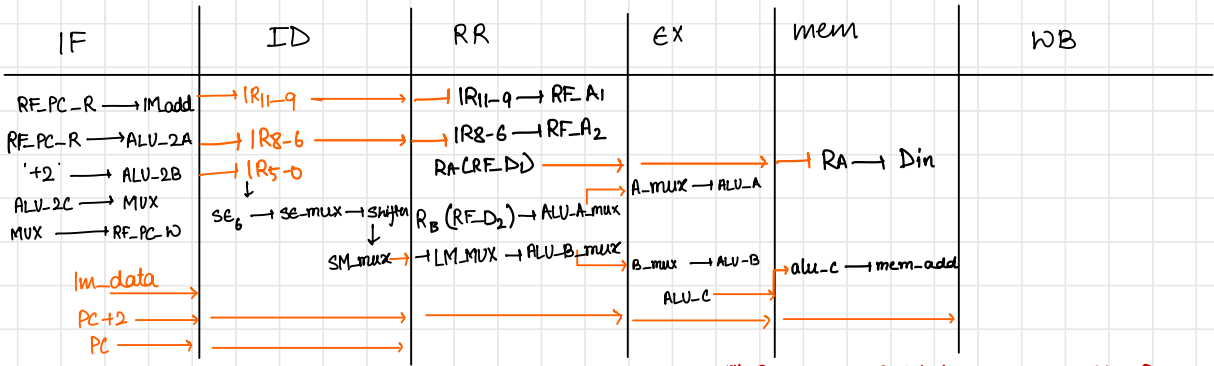
0100 LW alu_op : 0000 (add) M(Rb + Imm6) → Ra





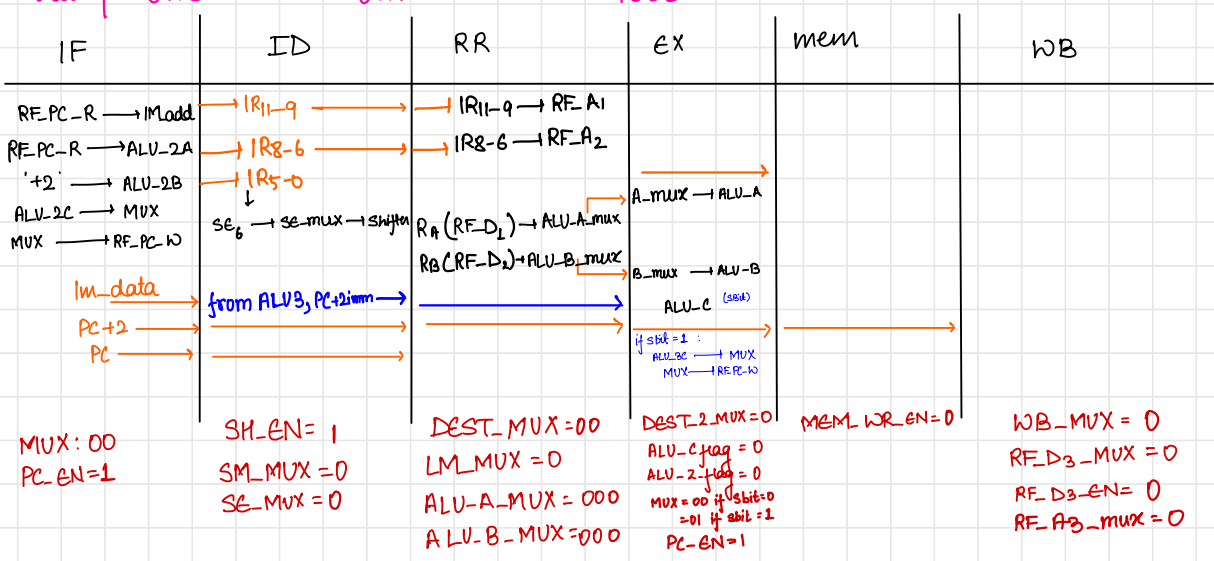


SW 0101 alu_op = add (0000) SW ra, rb, 9mm



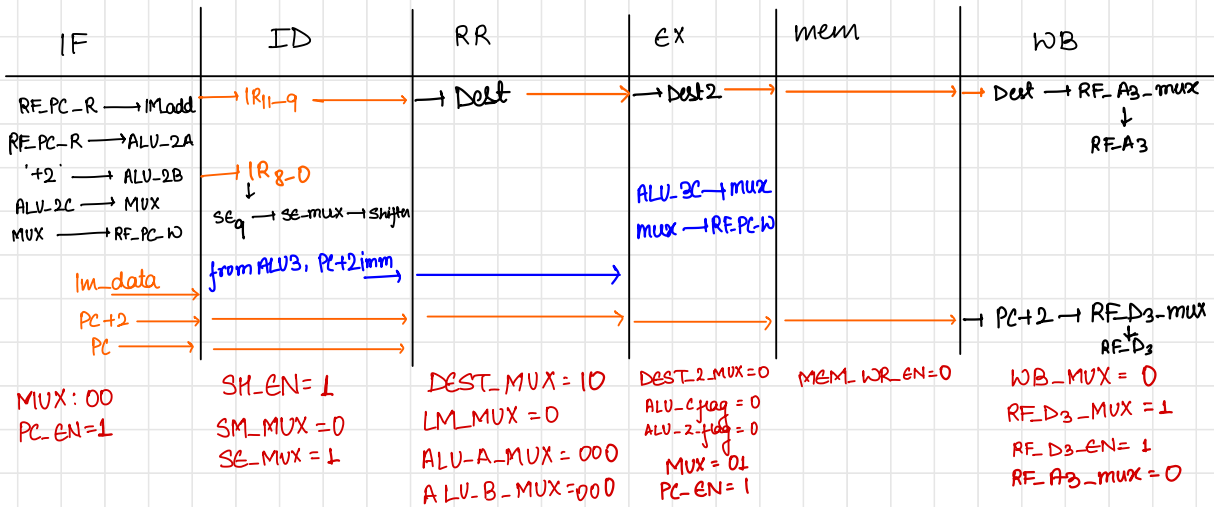
BEQ 1000 BLT 1001 BLE 1010

alu_op 0110 0111 1000



JAL 1100

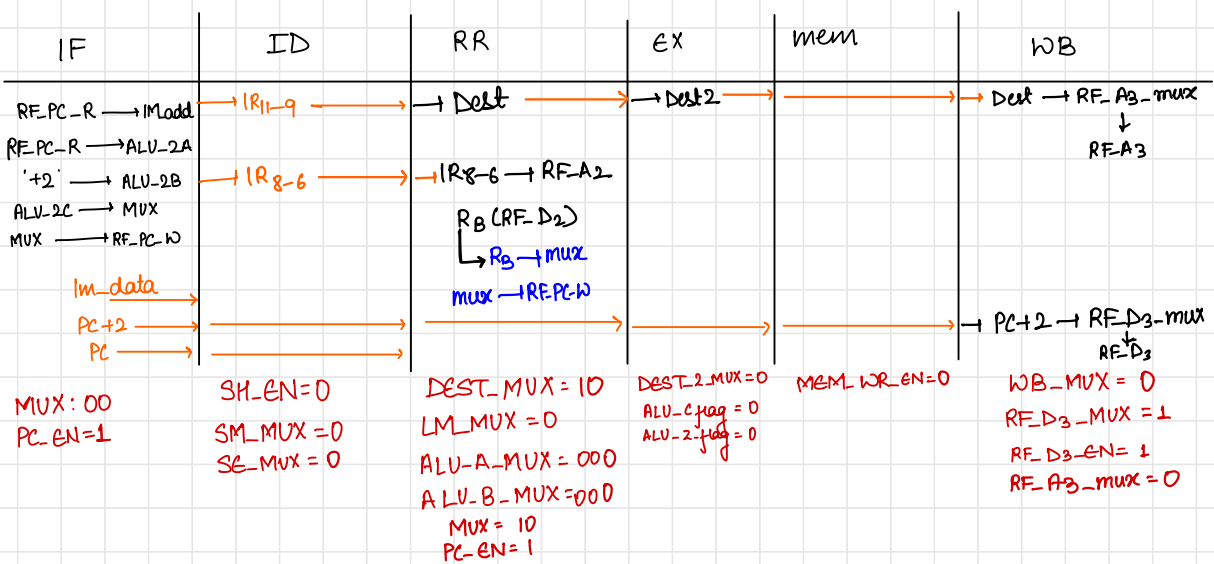
alu-op : nop (1111)



JLR 1101

alu-op : NOP (1111)

jlr ra, rb



JRI 1111 alu-op = 0000 (add) jri, ra, 9mm9