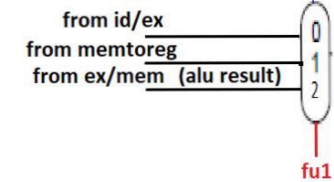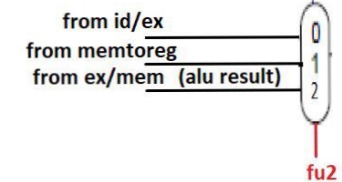**Datapath**

Navid Dehban 810198390
Reza Jahani   810198377

A

from id/ex
from memtoreg
from ex/mem   (alu result)

0
1
2

fu1

B
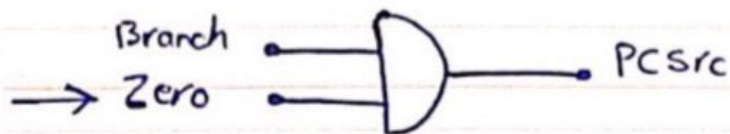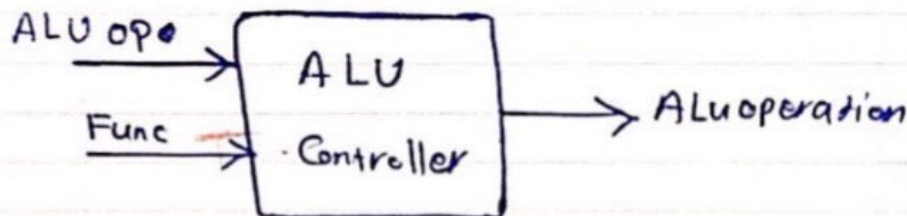
from id/ex
from memtoreg
from ex/mem   (alu result)

0
1
2

fu2

« Controller »

Reza Jahani   810198377

Navid Dehban   810198390

OPC → **Signal Controller** → MemtoReg ، MemRead ، MemWrite
ALUsrc ، RegWrite ، RegDst ، Writesel
Pc1 ، Pc2
→ ALUOP
→ Branch

ALU op → **ALU Controller** → ALUoperation
Func →

Branch •
Zero • —⟩ → PCSrc

ALU Controller
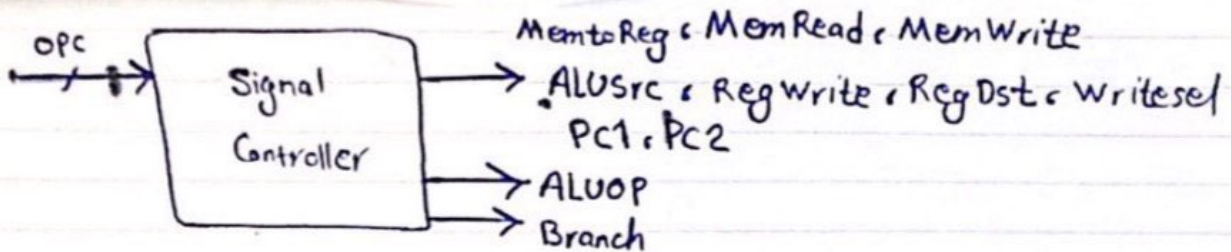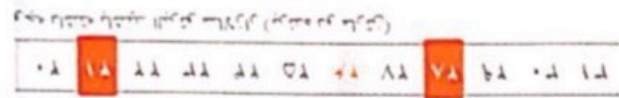
ALUoperation	F
000	and
001	or
010	add
011	sub
111	slt

ALUOP	Func	ALUoperation	
00	————	011	beq
01	————	010	addi Mem.
10	000001	010	RT
	000010	011	
	000100	000	
	001000	001	
	010000	111	
11	————	111	slti

Signal Controller	ALUOP	Mem to Reg	MemRead	MemWrite	ALU src	Reg Write	Reg Dst	Write sel	Pc1	Pc2	Branch
R-type	10	0	0	0	0	1	10	1	X	0	0
addi	01	0	0	0	1	1	00	1	X	0	0
slti	11	0	0	0	1	1	00	1	X	0	0
Lw	01	1	1	0	1	1	00	1	X	0	0
Sw	01	X	0	1	1	X	X	X	X	0	0
beq	00	X	X	X	0	X	X	X	X	0	1
j	X	X	X	X	X	X	X	X	0	1	0
jr	X	X	X	X	X	X	X	X	1	1	0
jal	X	X	X	X	X	1	01	0	0	1	0

Forwarding Unit

1. EX hazard:

if (EX/MEM.RegWrite

and (EX/MEM.RegisterRd != 0)

and (EX/MEM.RegisterRd == ID/EX.RegisterRs)) ForwardA = 10

if (EX/MEM.RegWrite

and (EX/MEM.RegisterRd != 0)

and (EX/MEM.RegisterRd == ID/EX.RegisterRt)) ForwardB = 10

2. MEM hazard:

if (MEM/WB.RegWrite

and (MEM/WB.RegisterRd != 0)

and (MEM/WB.RegisterRd == ID/EX.RegisterRs)) ForwardA = 01

if (MEM/WB.RegWrite

and (MEM/WB.RegisterRd != 0)

and (MEM/WB.RegisterRd = ID/EX.RegisterRt)) ForwardB = 01

----

if (ID/EX.MemRead and

((ID/EX.RegisterRt = IF/ID.RegisterRs) or

(ID/EX.RegisterRt = IF/ID.RegisterRt)))

stall the pipeline