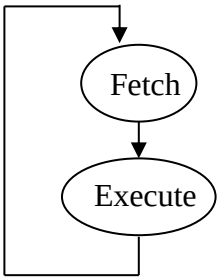
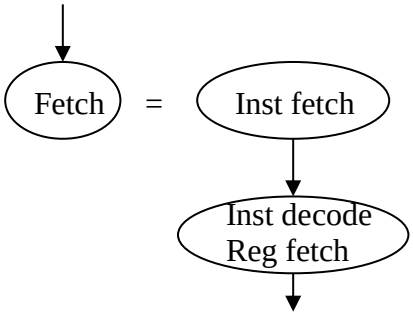
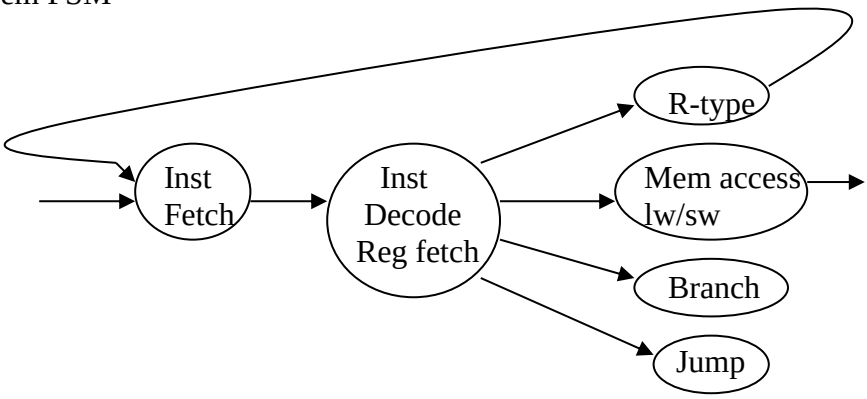


Fetch = Inst fetch + Inst decode and Register fetch



Execute

Mem FSM



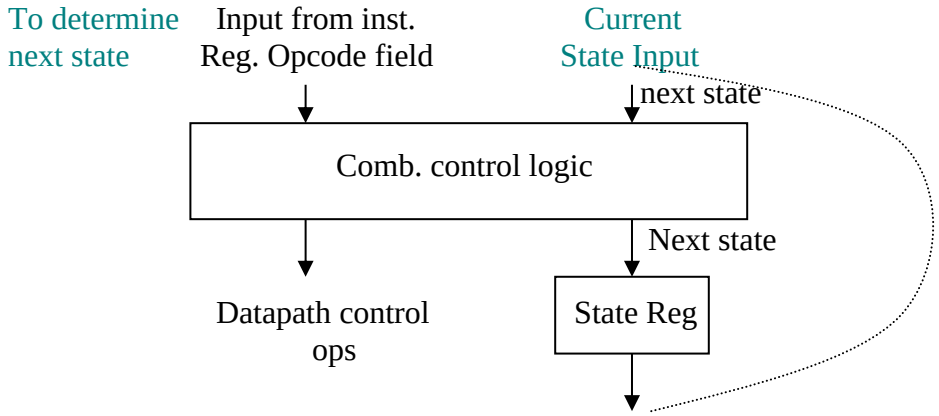
MIPS Controller

Design a FSM controller for MIPS processor with more than six RISC instructions.

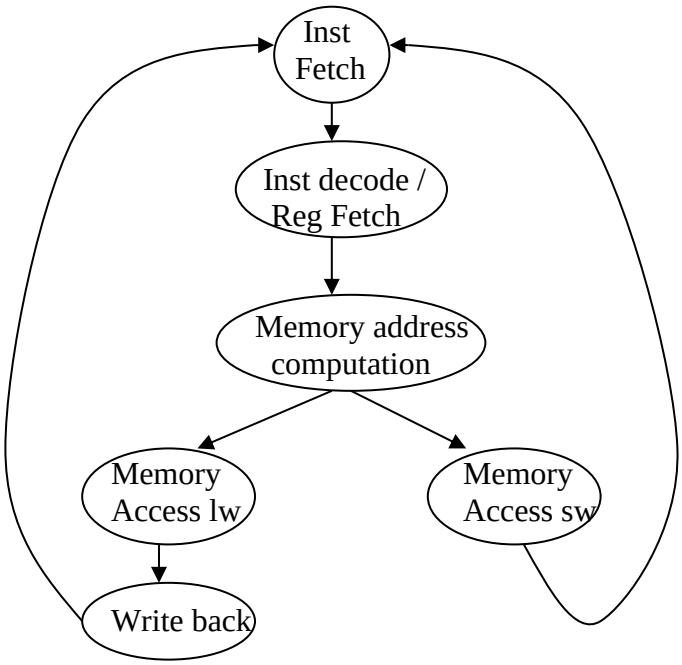
Submit your report with the following:

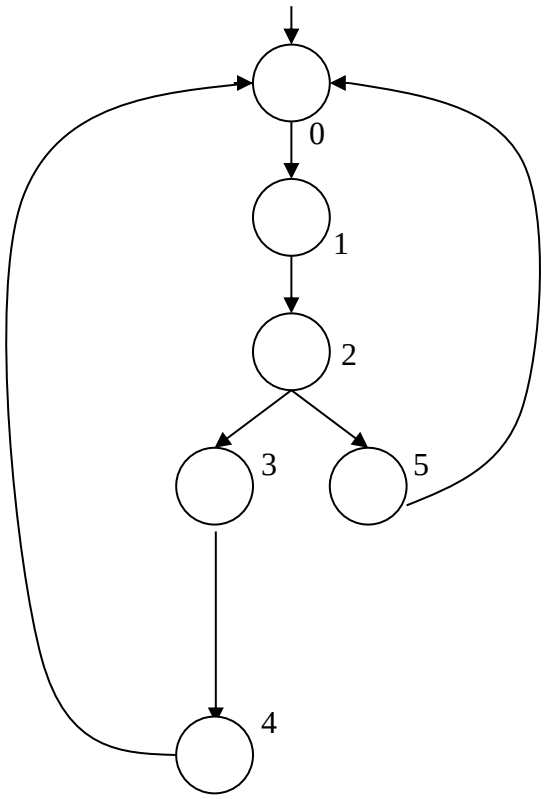
- FSM
- Control signals for each state
- Control signals to Boolean equations
- State equations
- Programmable Logic Array diagram representing control signal outputs using inputs

Implemented using a block of comb. logic & register to hold the current state



lw/sw continued





0 – MemRead
AluSrcA = 0
IorD = 0
IR write
AluSrcB = 01
Aluop = 00
PCwrite
PCSource = 00

1 – AluSrcA = 0
AluSrcB = 11
Aluop = 00

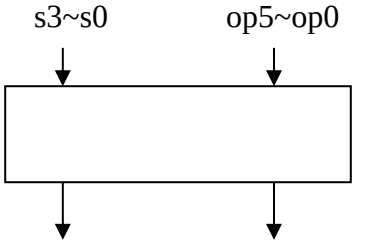
2 – AluSrcA = 1
AluSrcB = 10
Aluop = 00

3 – MemRead
IorD = 1

4 – MemWrite
IorD = 1

5 – RegDst = 0
RegWrite
MemtoReg = 1

Controller



MemRead = State 0 + State 3
AluSrcA = State 2
IorD = State 3 + State 4
IRwrite = State 0
$AluSrcB_0$ = State 0 + State 1
$AluSrcB_1$ = State 1 + State 2
PCwrite = State 0
AluSrcB = State 1 + State 2
MemWrite = State 5
Regwrite = State 5
MemtoReg = State 5

NS Bits
Next State 0 = State 4 + State 5
Next State 1 = State 0
Next State 2 = State 1(lw + sw)
Next State 3 = State 2.lw
Next State 4 = State 3
Next State 5 = State 2(sw)

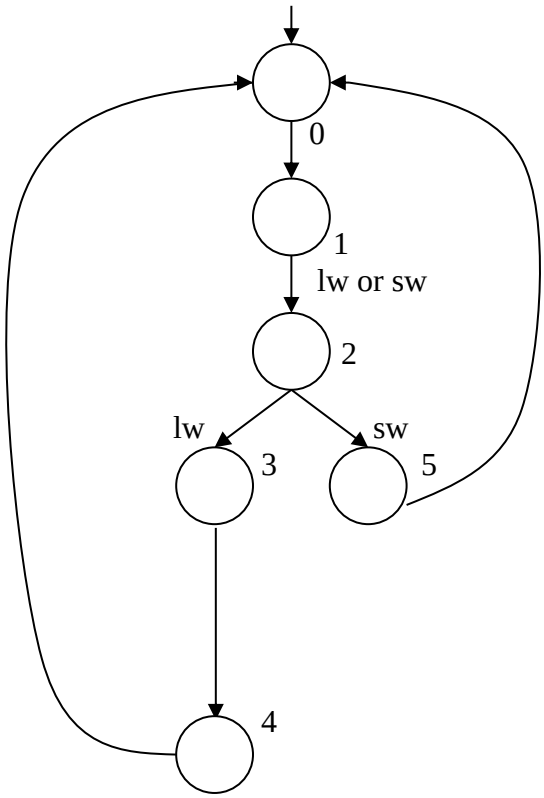
NS2	NS1	NS0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1

NS0 = 1 + 3 + 5
NS1 = 2 + 3
NS2 = 4 + 5

NS0 = State 0 + State 2.lw + State 2.sw
NS1 = State 1(lw + sw) + State 2.lw
NS2 = State 3 + State 2(sw)

FSM Controller for lw/sw

6 states



NS0 should be active whenever NS0 = 1 is true for
Next State 1 + Next State3 + Next State 5

Next State 1 = State 0 = $\overline{S_2S_1S_0}$

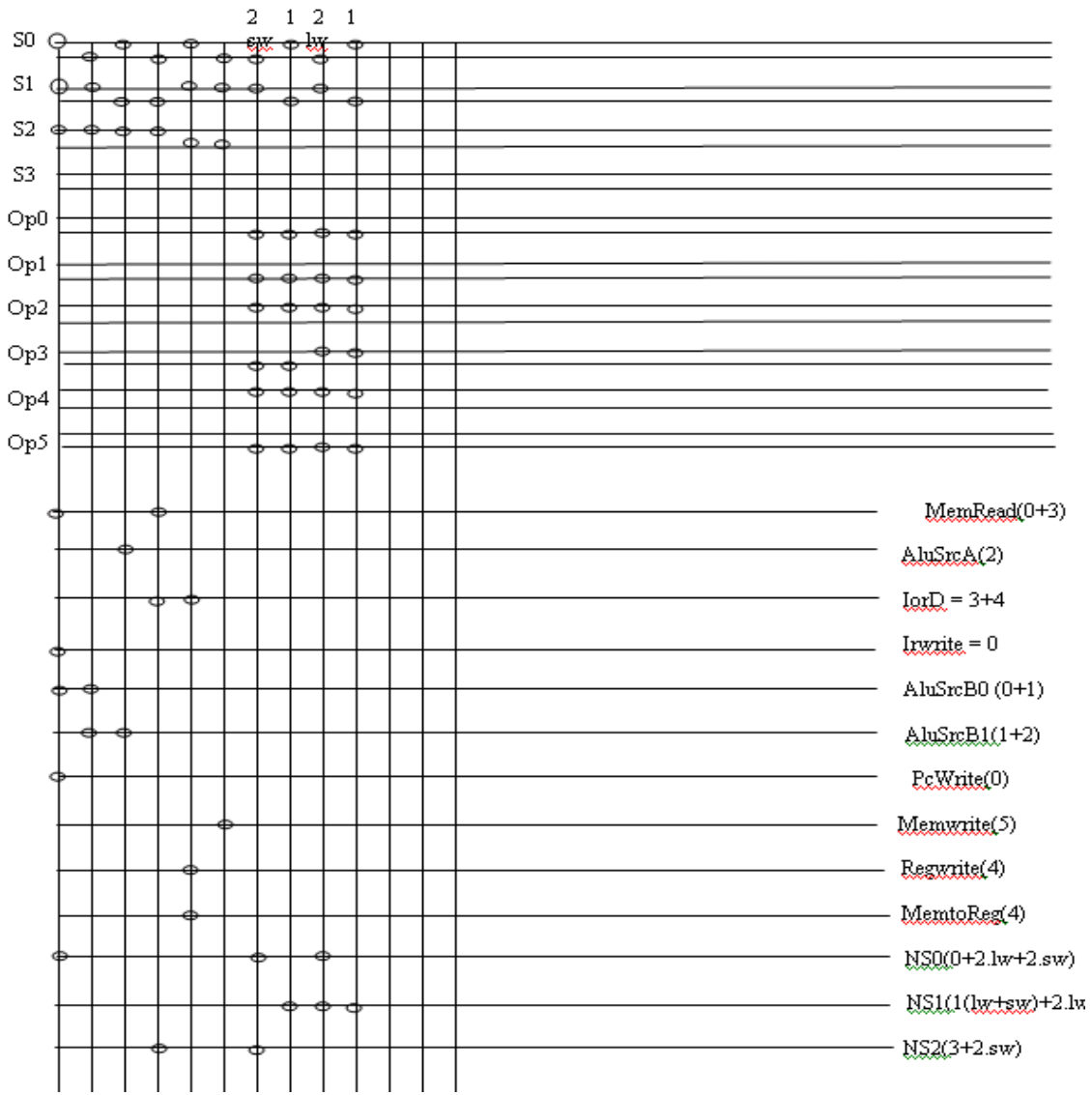
Next State 3 = State 2(lw) $= S_2S_1S_0$ (op5op4op3op2op1op0)

NS0 = 1 + 3 + 5
= State 0 + (State 2 & lw) + (State 2 & sw)

NS1 = 2 + 3
= (State 1 & lw + State 1 & sw) + (State 2 & lw)

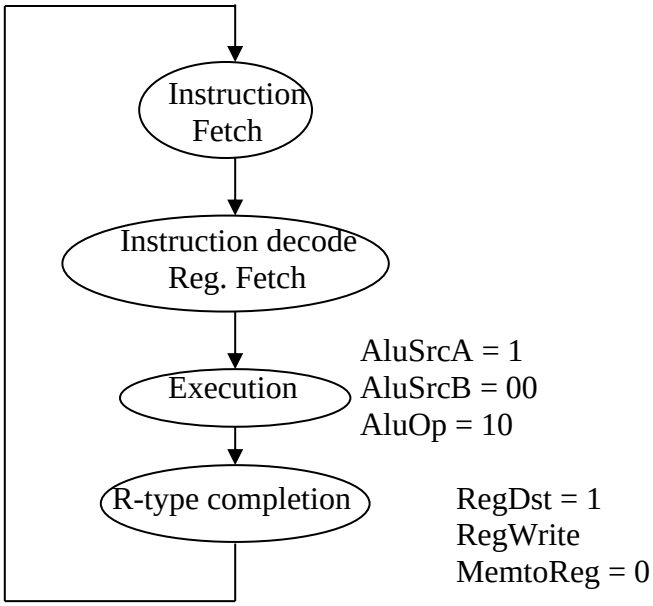
NS2 = 4 + 5
= (State 2 & sw) + State 3

PLA for LW+SW

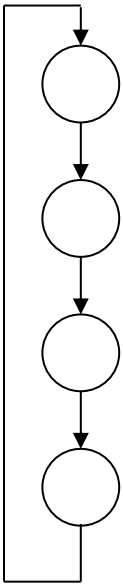

2
1
2
1
sw
lw
S0
S1
S2
S3
Op0
Op1
Op2
Op3
Op4
Op5
MemRead(0+3)
AluSrcA(2)
IorD = 3+4
Irwrite = 0
AluSrcB0 (0+1)
AluSrcB1(1+2)
PcWrite(0)
Memwrite(5)
Regwrite(4)
MemtoReg(4)
NS0(0+2.lw+2.sw)
NS1(1(lw+sw)+2.lw
NS2(3+2.sw)

Lecture 28

R-type instruction continue



R-type



NS1
00
01
10
11

NS0 = 1 + 3

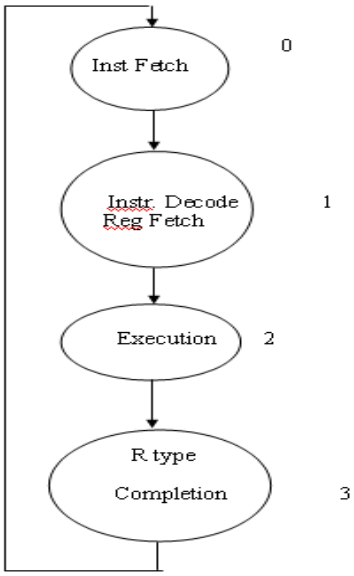
= Current State 0 + Current State 2

= $S_1\overline{S_0} + \overline{S_1}S_0$

NS1 = 2 + 3

= Current State 1 & (op = R-type) + Current State 2

R Type



State 0
- MemRead
- AluSrcA = 0
- IorD = 0
- IRwrite
- AluSrcB = 01
- Aluop = 00
- PCwrite
- PCSource = 00

State 1 (0 0)
- AluSrcA = 0 (0 1)
- AluSrcB = 11 (1 0)
- Aluop = 00 (1 1)

State 2
- AluSrcA = 1
- AluSrcB = 00
- Aluop = 10

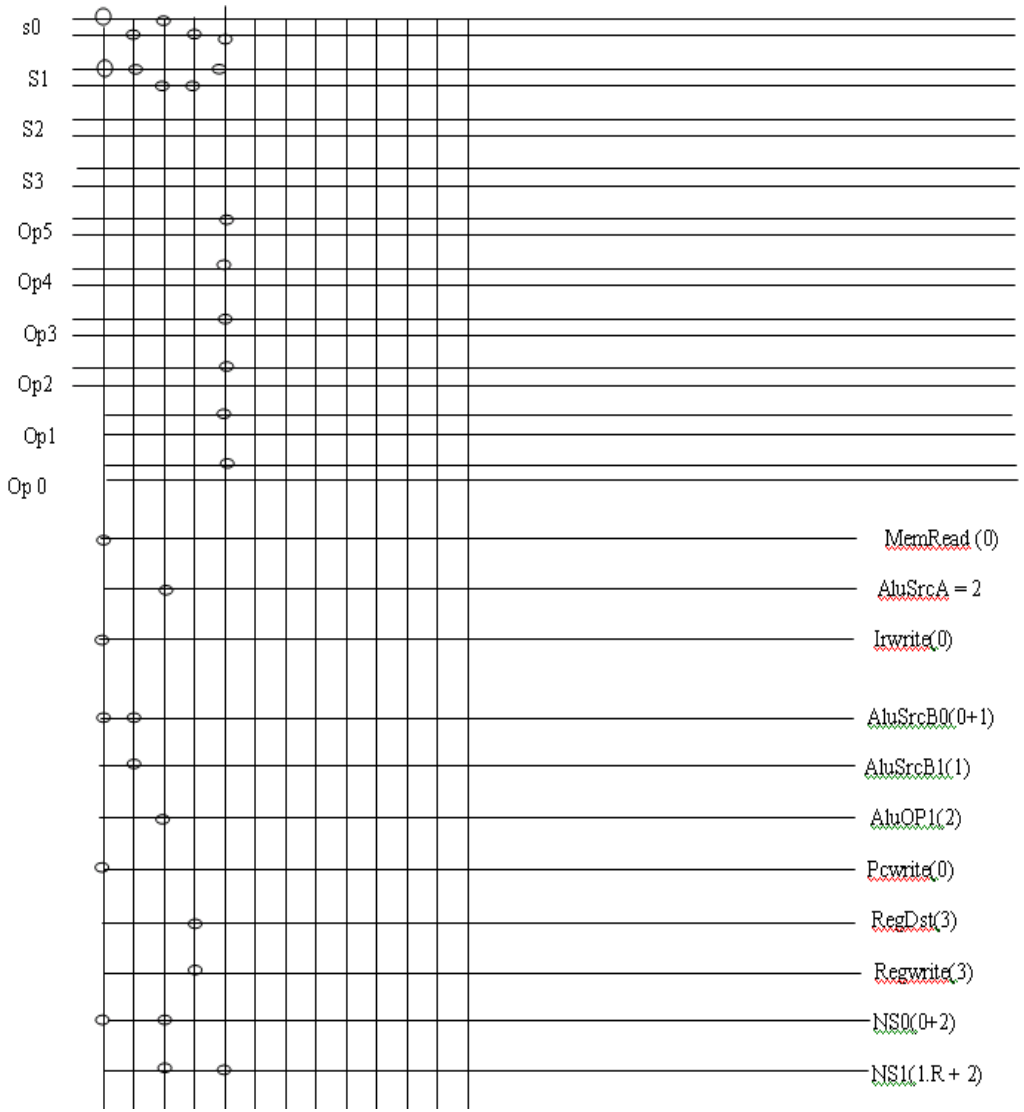
State 3
- RegDst = 1
- RegWrite
- MemtoReg = 0

Nsbits
- Next State 0 = State 3
- Next State 1 = State 0
- Next State 2 = State 1.R
- Next State 3 = State 2

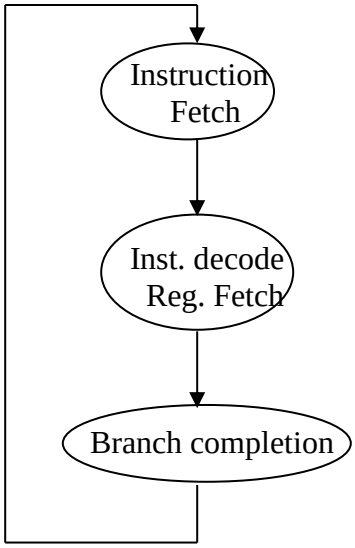
NS0 = 1+3= State 0 + State 2
NS1 = 2+3 = State1.R + State 2

MemRead = 0
AluSrcA = 2
Irwrite = 0
AluSrcB0 = 0+1
AluSrcB1 = 1
Aluop1 = 2
Pcwrite = 0
RegDst = 3
RegWrite = 3
NS0 = 0+2
NS1 = 1.R + 2

PLA for R-Type

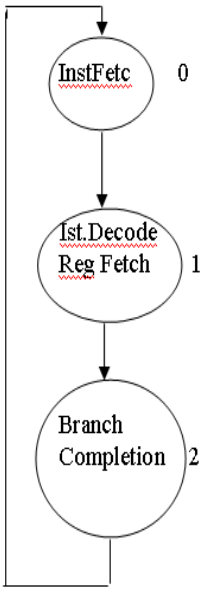


Branch



AluSrcA = 1
AluSrcB = 00
AluOp = 01
PCWrite
PCSource = 01

Branch



State 0

MemRead
AluSrcA = 0
IorD = 0
IRwrite
AluSrcB = 01
Aluop = 00
Pcwrite
Pcsource = 00

State 1

AluSrcA = 0
AluSrcB = 11
Aluop = 00

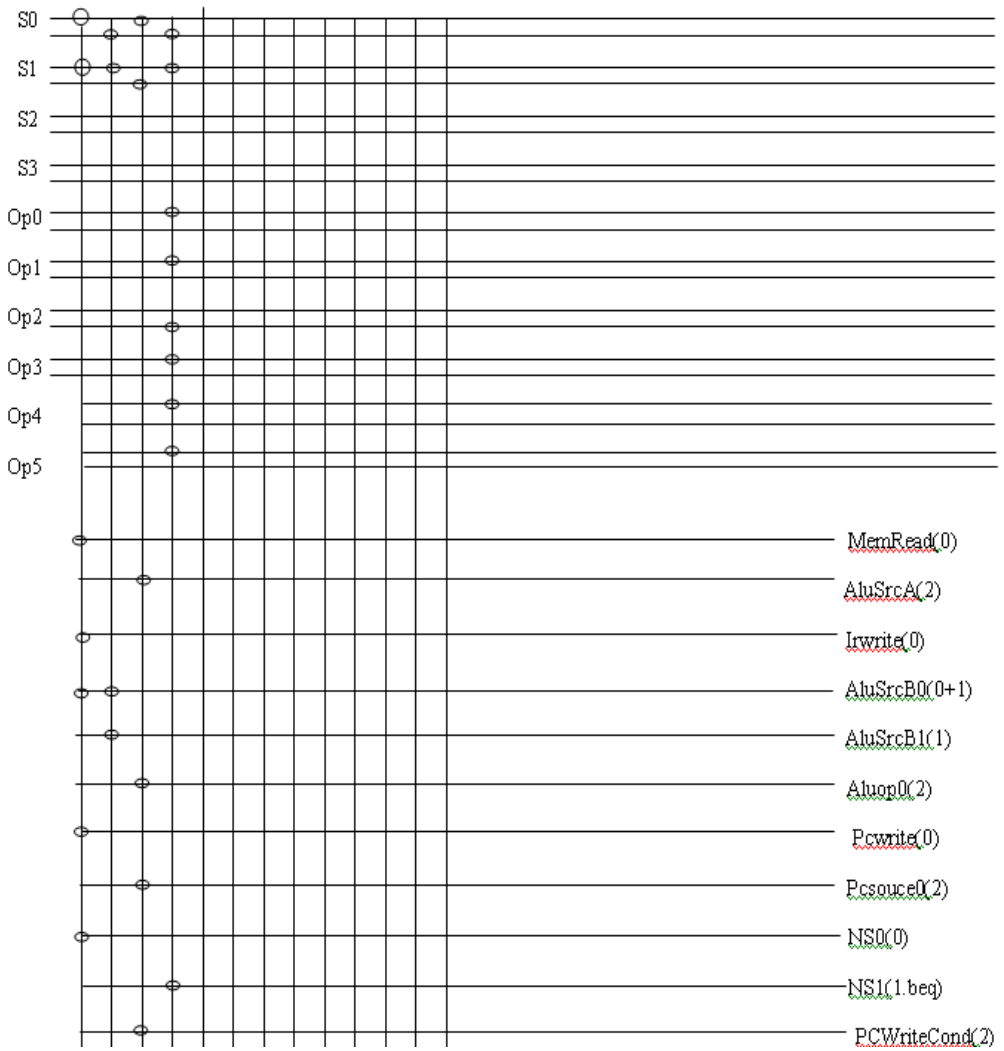
State 2

AluSrcA = 1
AluSrcB = 00
AluOP = 01
PcwriteCond
Pcsource =01

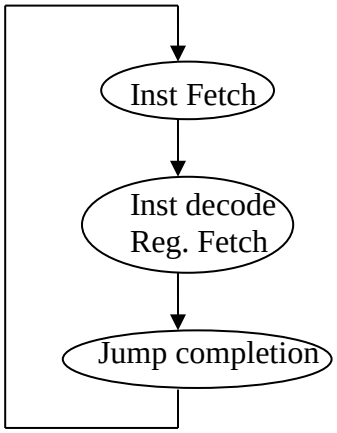
MemRead = 0
AluSrcA = 2
Irwrite = 0

NS0 = 1
NS1 = 2

PLA for beq

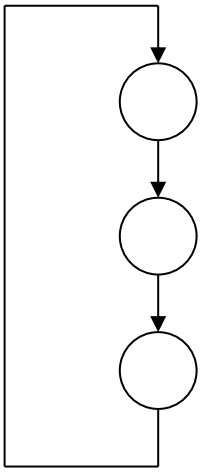

S0
S1
S2
S3
Op0
Op1
Op2
Op3
Op4
Op5
MemRead(0)
AluSrcA(2)
Irwrite(0)
AluSrcB0(0+1)
AluSrcB1(1)
Aluop0(2)
Pcwrite(0)
Pcsouce0(2)
NS0(0)
NS1(1.beq)
PCWriteCond(2)

Jump



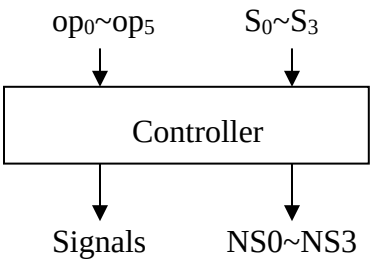
PCWrite
PCSource = 10

Jump

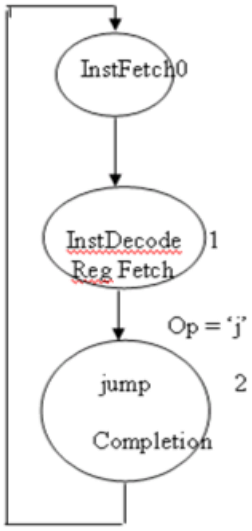


NS0 = 1
NS1 = 2

NS0 = 1 = Current State 0
NS1 = 2 = Current State 1 & opJ



J Type



State 0
MemRead
AluSrcA = 0
IorD = 0
IRwrite
AluSrcB = 01
Aluop = 00
PcWrite
PCSrc = 00

State1
AluSrcA = 0
AluSrcB = 11
Aluop = 00

State 2
Pcwrite
PcSource = 10

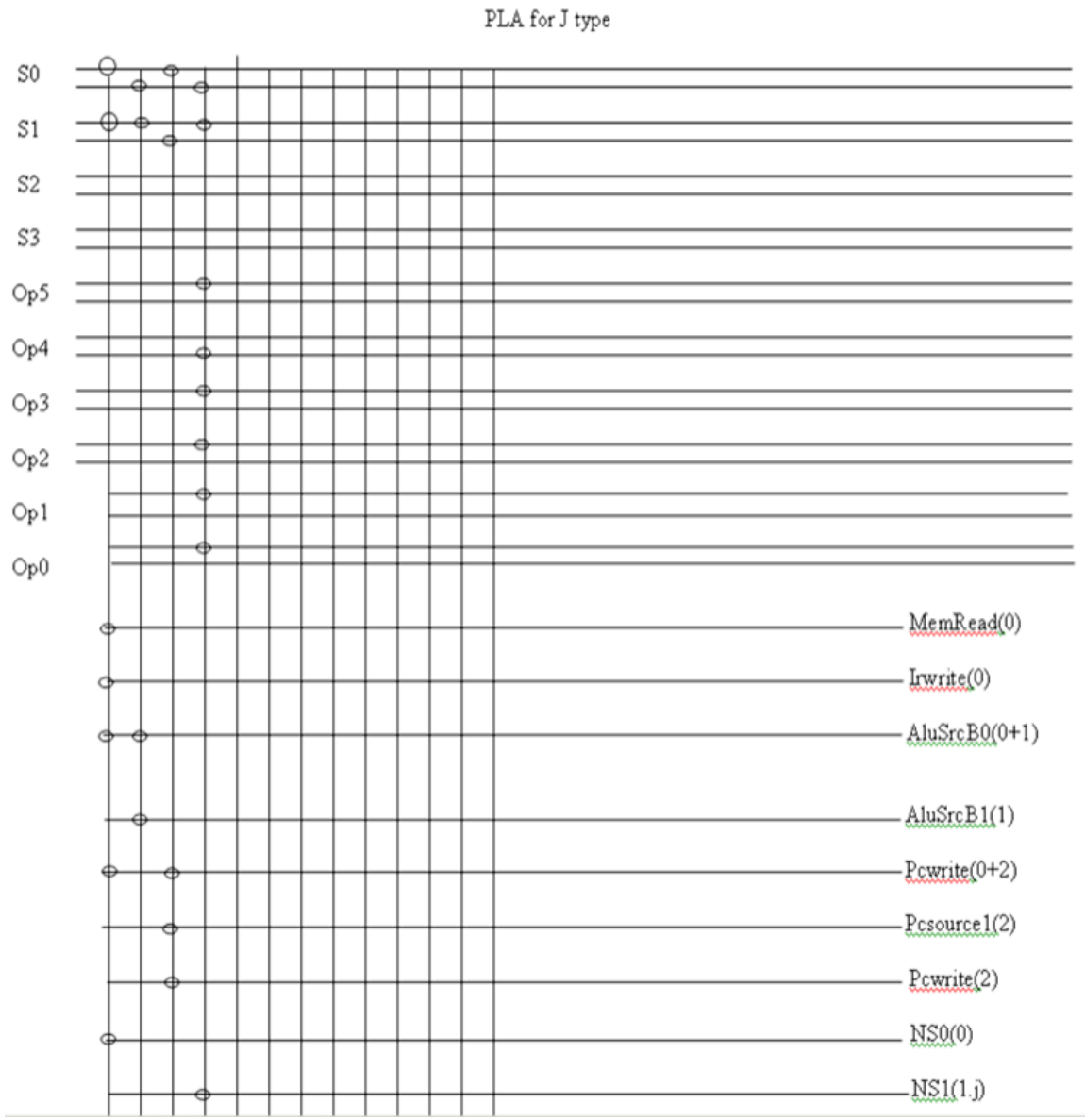
NSBits

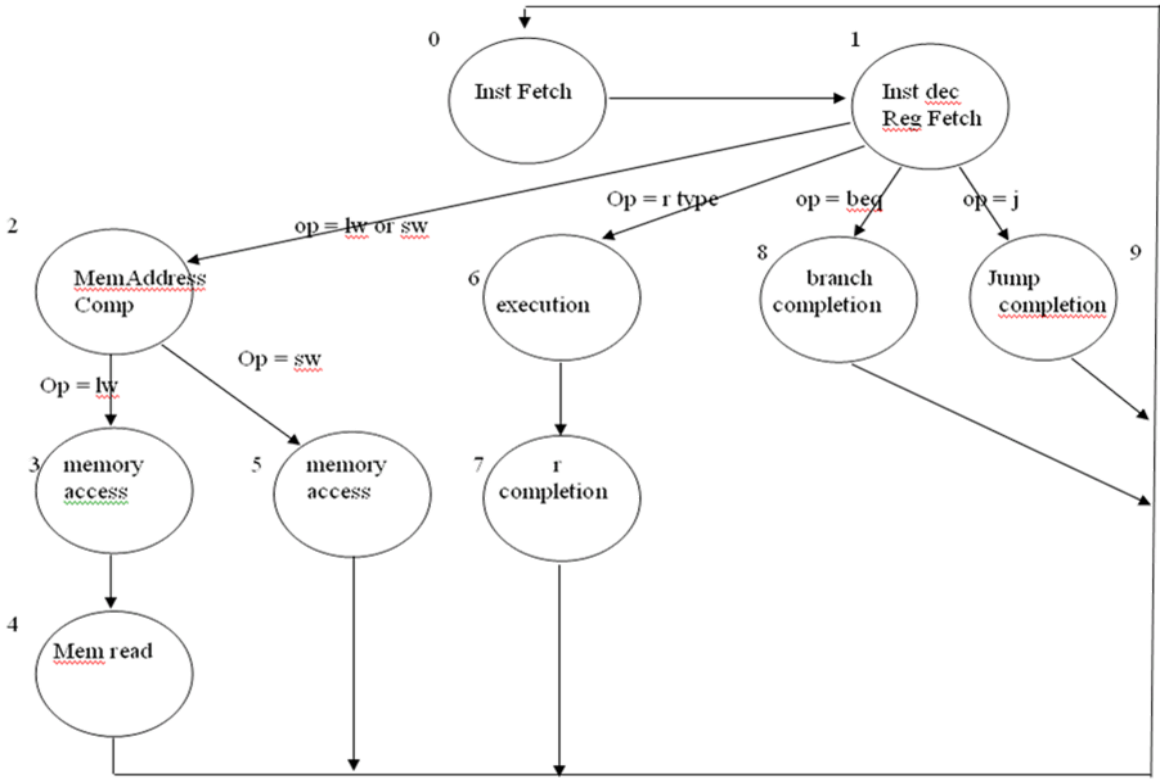
Next State 0 = State 2
Next State 1 = State 0
Next State 2 = State 1.j

NS0 = 1 = State 0
NS1 = 2 = State 1.j

0 0
0 1
1 0

MemRead = 0
Irwrite = 0
AluSrcB0 = 0+1
AluSrcB1 = 1
Pcwrite = 0+2
PcSource1 = 2
PcWrite = 2


PLA for J type
S0
S1
S2
S3
Op5
Op4
Op3
Op2
Op1
Op0
MemRead(0)
Irwrite(0)
AluSrcB0(0+1)
AluSrcB1(1)
Pcwrite(0+2)
Pcsource1(2)
Pcwrite(2)
NS0(0)
NS1(1.j)


0
Inst Fetch
1
Inst dec
Reg Fetch
op = lw or sw
Op = r type
op = beq
op = j
2
MemAddress
Comp
6
execution
8
branch
completion
Jump
completion
9
Op = sw
Op = lw
3
memory
access
5
memory
access
7
r
completion
4
Mem read

State 0

MemRead
AluSrcA = 0
IorD = 0
Irwrite
AluSrcB = 01
Aluop = 00
PCWrite
PCSource = 00

State 1

AluSrcA = 0
AluSrcB = 11
Aluop = 00

State 2

AluSrcA = 1
AluSrcB = 10
Aluop = 00

State 3

MemRead
IorD = 1

State 4

RegDst = 0
RegWrite
MemtoReg = 1

State 5

MemWrite
IorD = 1

State 6

AluSrcA = 1
AluSrcB = 00
Aluop = 10

State 7

RegDst = 1
RegWrite
MemtoReg = 0

State 8

AluSrcA = 1
AluSrcB = 00
Aluop = 01
PcWriteCond
PcSource = 01

State 9

PcWrite
PCSource = 10