Next State via Sequencer

- Most of the control logic is needed just for the next state number
- Our core MIPS system only has 10 states
- More realistic systems would have many more states
- Often the next state is just the previous state + 1
- The start state always comes after the final state in each instruction
- The start state begins the next instruction

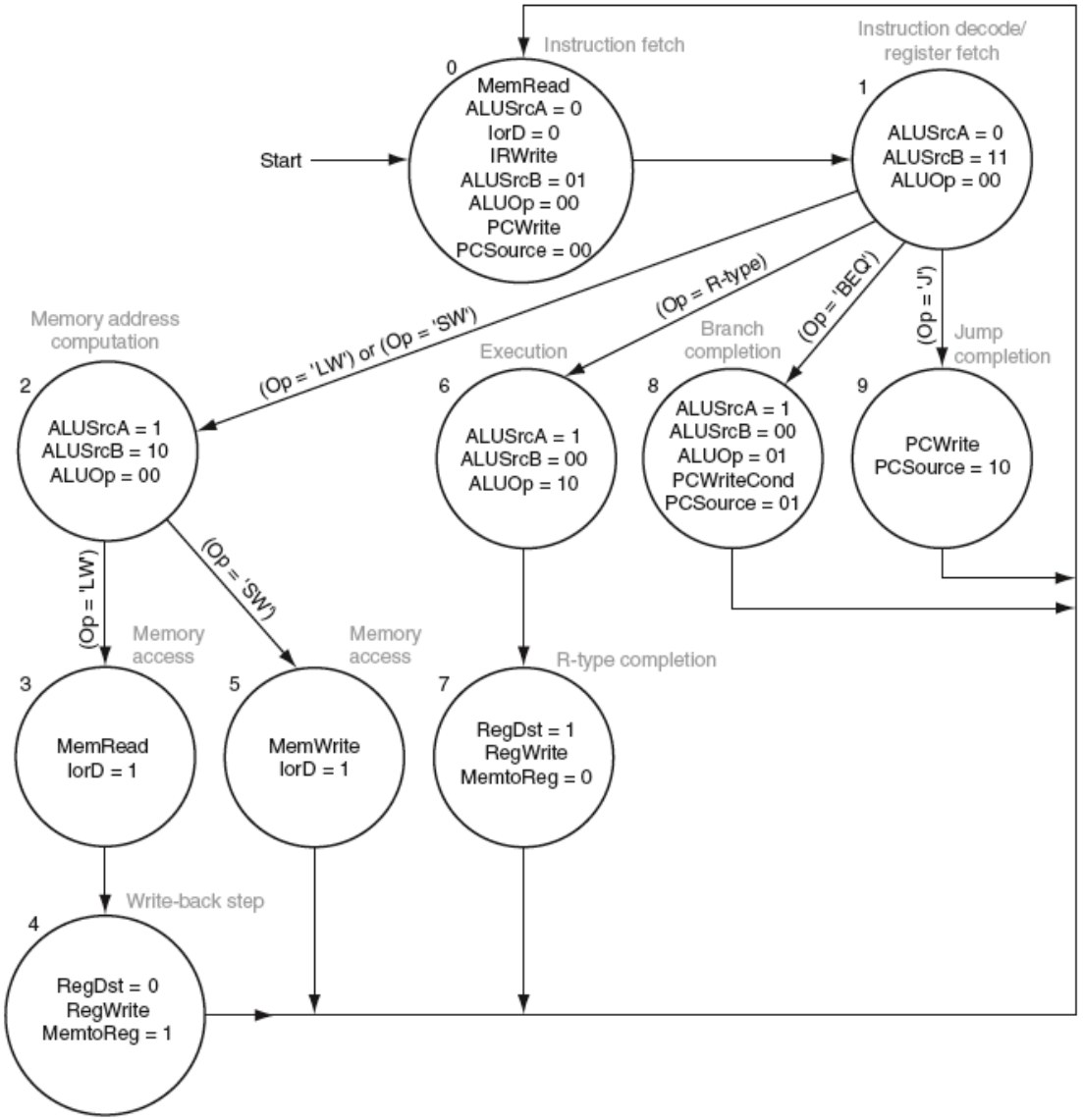
State Diagram

State1 always comes after state0

State4 always comes after state3

State7 always comes after state 6

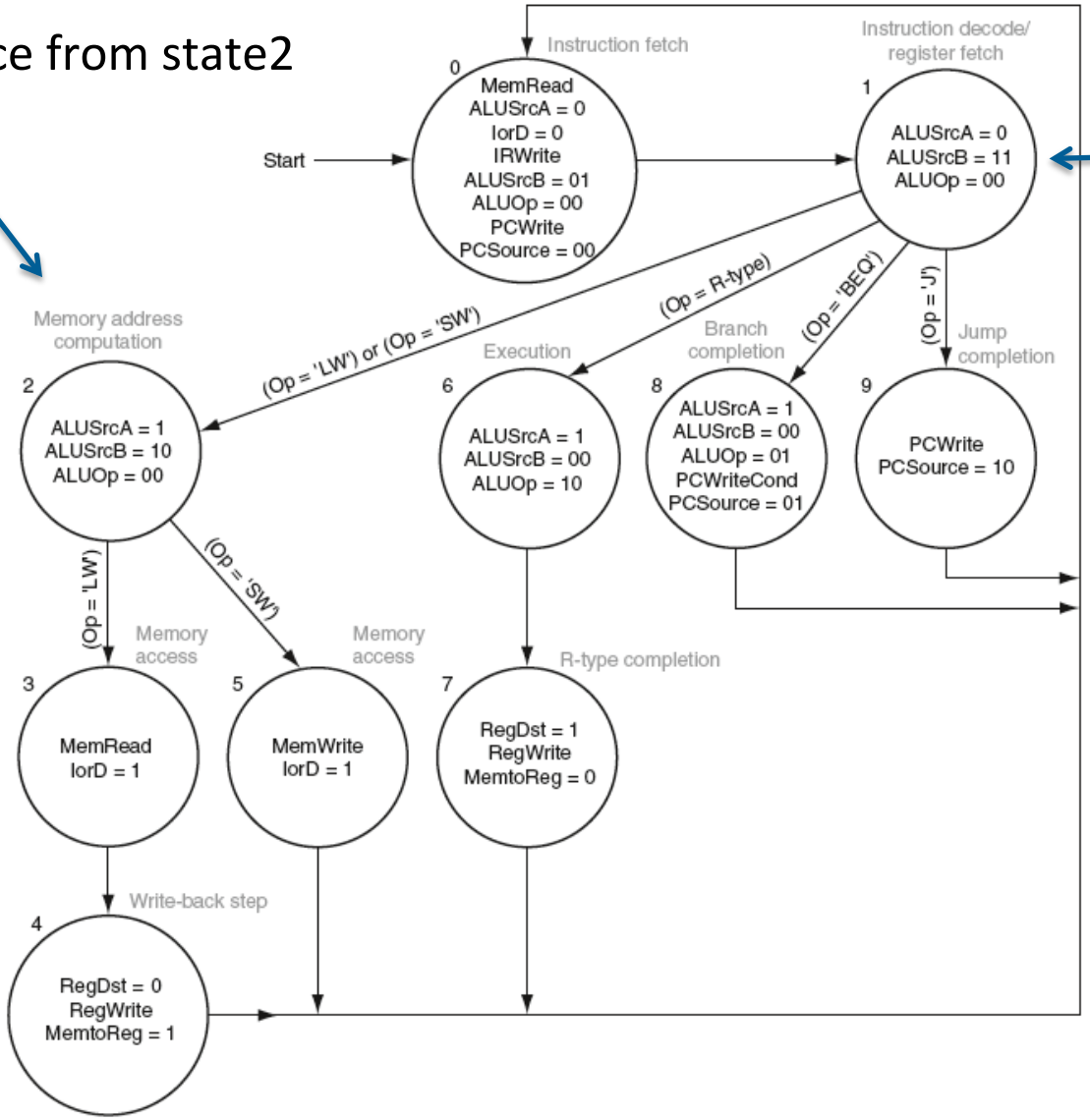
State0 always comes after states 4, 5, 7, 8 and 9



State Diagram

2-way choice from state2

4-way choice from state1



State Transitions

To go to the next sequential state, just increment the current state

Set the state to 0 to go back to the initial state

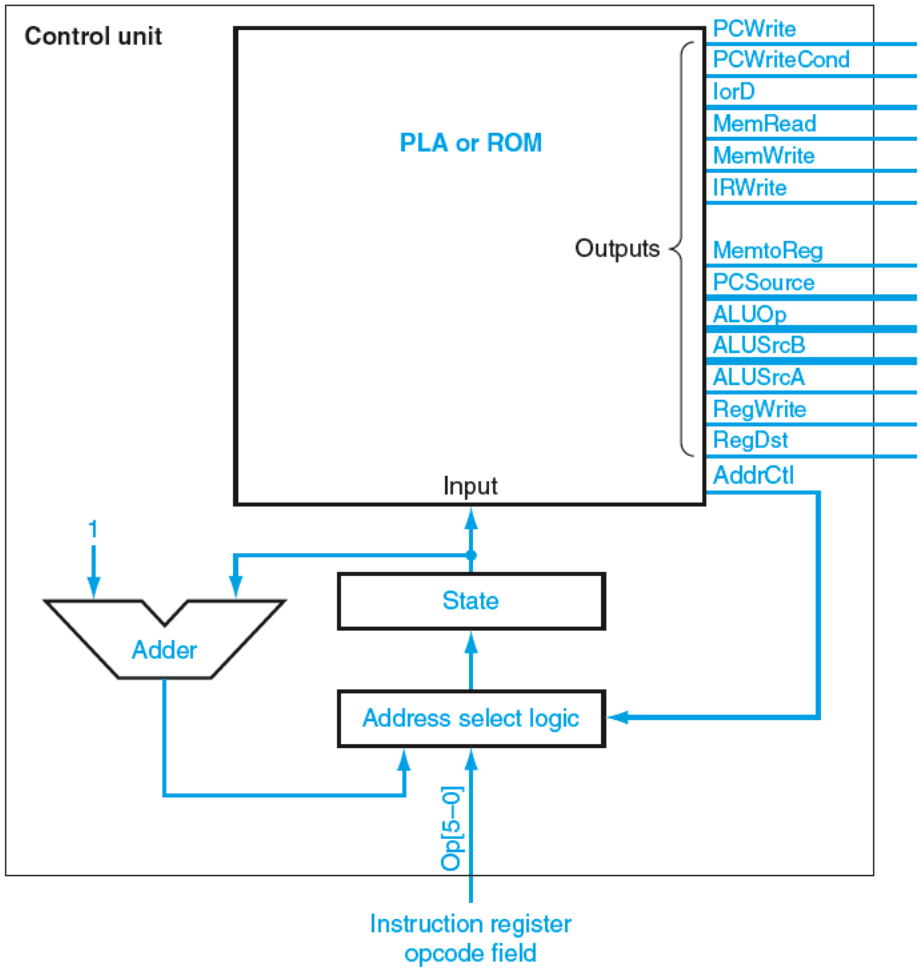
Small lookup tables can be used to choose a non-sequential state

The choice from state 1 and from state2 is based on the opcode

Two ROMs (addressed by opcode) can be used

A 2-bit control signal can be used with a MUX for 4 possible choices

State Register Control



2-bit AddrCtl causes Address select logic to output the next state number

State Register Control

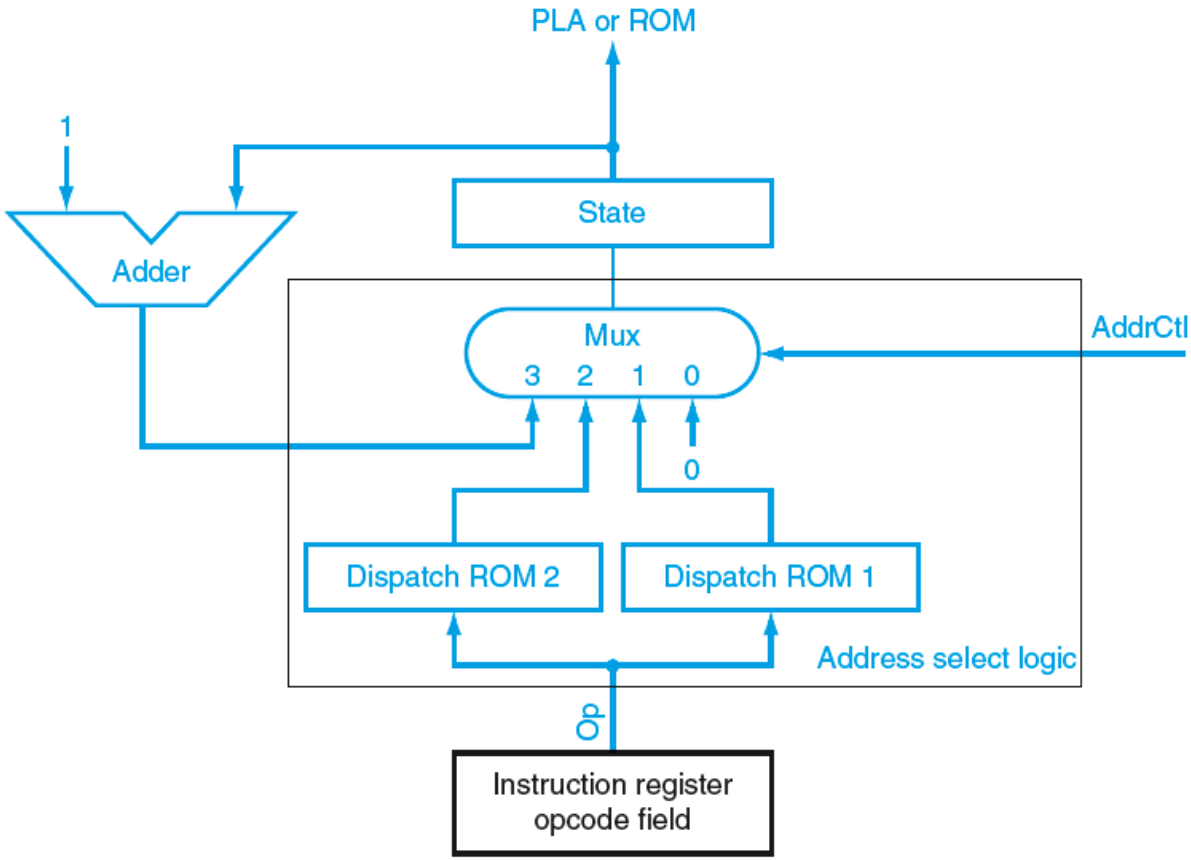
AddrCtl is defined as follows:

AddrCtl value	Action
0	Set state to 0
1	Dispatch with ROM 1
2	Dispatch with ROM 2
3	Use the incremented state

In state1, next state is looked up in dispatch ROM1

In state2, next state is looked up in dispatch ROM2

State Register Control



AddrCtl selects from: 0, ROM1, ROM2 or Output of Adder as next state

Dispatch ROMs

Dispatch ROM 1		
Op	**Opcode name**	**Value**
000000	R-format	0110
000010	jmp	1001
000100	beq	1000
100011	lw	0010
101011	sw	0010

Dispatch ROM 2		
Op	**Opcode name**	**Value**
100011	lw	0011
101011	sw	0101

6-bit opcode as index implies up to $2^6 = 64$ entries

Only 5 entires are used from ROM1

Only 2 entries are used from ROM2

AddrCtl

AddrCtl is determined just by the state number

State number	Address-control action	Value of AddrCtl
0	Use incremented state	3
1	Use dispatch ROM 1	1
2	Use dispatch ROM 2	2
3	Use incremented state	3
4	Replace state number by 0	0
5	Replace state number by 0	0
6	Use incremented state	3
7	Replace state number by 0	0
8	Replace state number by 0	0
9	Replace state number by 0	0

State number determines AddrCtl as well as datapath control bits

Control Word Contents

AddrCtl as well as datapath control bits can be stored in a control memory

State number	Control word bits 17–2	Control word bits 1–0
0	1001010000001000	11
1	0000000000011000	01
2	0000000000010100	10
3	0011000000000000	11
4	0000001000000010	00
5	0010100000000000	00
6	0000000001000100	11
7	0000000000000011	00
8	0100000010100100	00
9	1000000100000000	00

State number serves as address of all 18 bits (16 control + 2 AddrCtl)

Each 18-bit pattern can be viewed as a single “micro-instruction”