ECS 154B
Computer Architecture II
Winter 2009

Multi-Cycle MIPS Control

Appendix C

Single Cycle Control

- Very simple
  - Control signals are functions of opcode and possibly function fields
  - Combinational logic suffices

- Ex: RegWrite
  - Asserted on `R-type`, `lw`
  - Deasserted on `beq`, `sw`, `j`

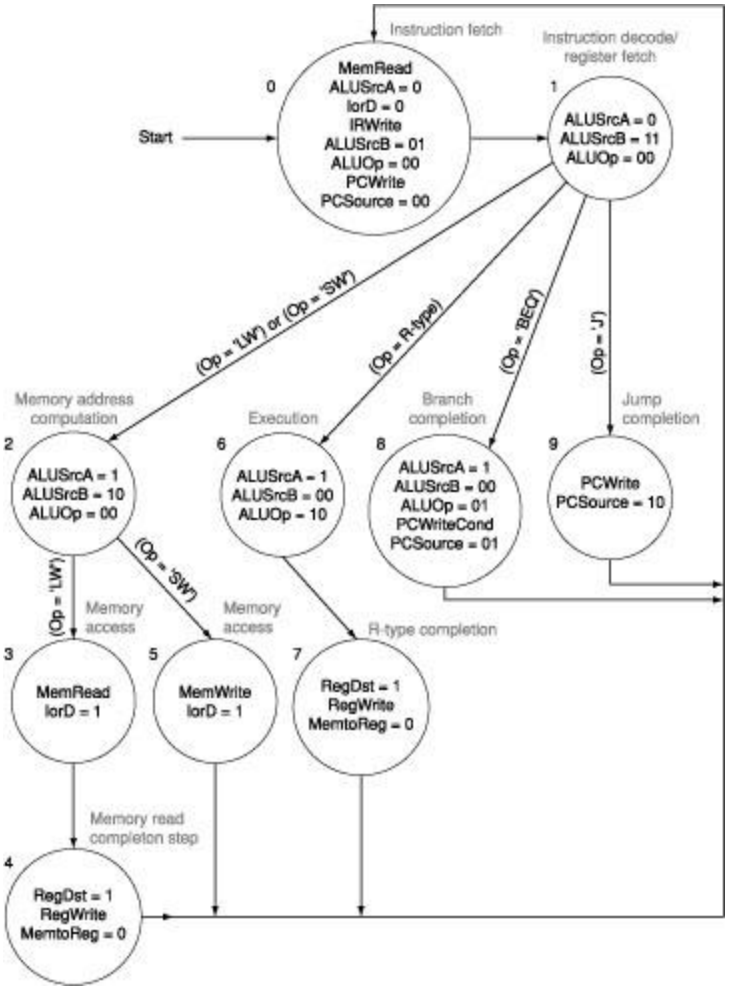
Multi-Cycle Control

- Much harder
  - Control signals depend on instruction and cycle
- Consider RegWrite

	Cycle				
Instruction	Fetch	Decode	Execute	Memory Access	Write Back
R-Type	0	0	0	1	
sw	0	0	0	0	
lw	0	0	0	0	1

- CPU must “remember” what cycle it is in
  - Control unit must maintain state
  - Several ways to do this

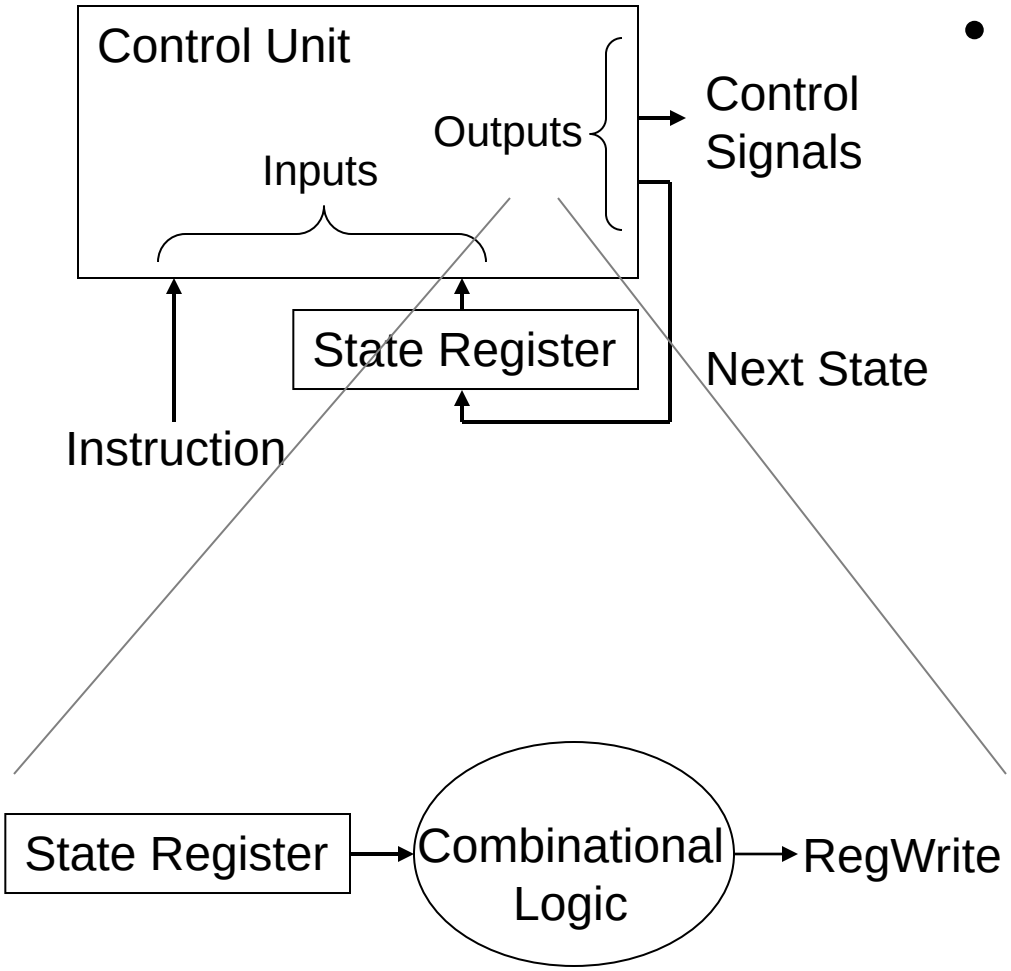
Review


Instruction fetch
Instruction decode/
register fetch
0
MemRead
ALUSrcA = 0
IorD = 0
IRWrite
ALUSrcB = 01
ALUOp = 00
PCWrite
PCSource = 00
Start
1
ALUSrcA = 0
ALUSrcB = 11
ALUOp = 00
(Op = 'LW') or (Op = 'SW')
(Op = R-type)
(Op = 'BEQ')
(Op = 'J')
Memory address
computation
Execution
Branch
completion
Jump
completion
2
ALUSrcA = 1
ALUSrcB = 10
ALUOp = 00
6
ALUSrcA = 1
ALUSrcB = 00
ALUOp = 10
8
ALUSrcA = 1
ALUSrcB = 00
ALUOp = 01
PCWriteCond
PCSource = 01
9
PCWrite
PCSource = 10
(Op = 'LW')
(Op = 'SW')
Memory
access
Memory
access
R-type completion
3
MemRead
IorD = 1
5
MemWrite
IorD = 1
7
RegDst = 1
RegWrite
MemtoReg = 0
Memory read
completon step
4
RegDst = 1
RegWrite
MemtoReg = 0

Multi-Cycle Control Implementation

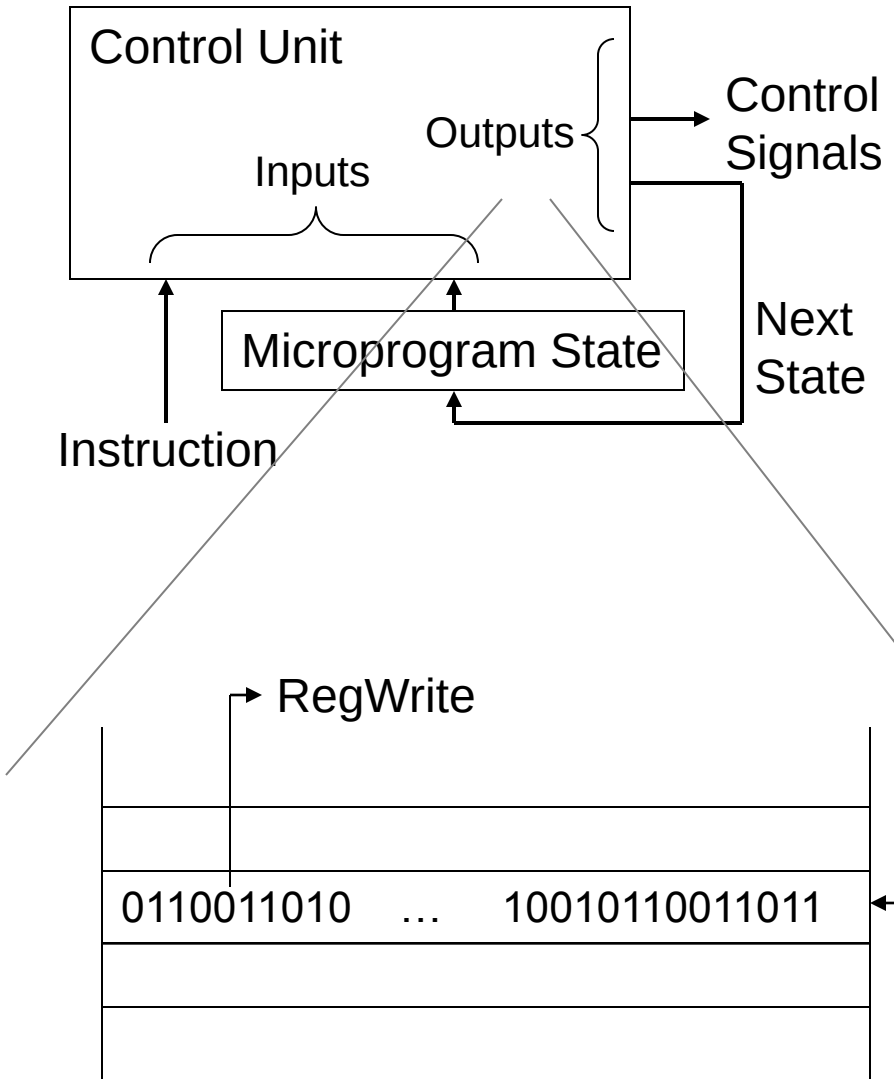
- Two main control implementations
  - State machine
    - Translate finite state machine diagrams to hardware
    - Control signals function of current state
  - Microprogram
    - A small control program runs in parallel to CPU datapath
    - Program outputs are control signals
- Logically similar in many respects
  - Control “remembers” state and changes signals
  - Implementation very different
  - Combinations also possible

State Machine Control



- From digital design:
  - Create state machine
  - Assign state values
  - Derive control signal functions
  - Derive next state functions

Microprogram Control



- Control Unit is now an indexed ROM
  - Memory bits set control signals and next state
  - Microprogram state and instruction select the memory value

Single Microprogram ROM

- Inputs
  - 6 bits from instruction opcode
  - 4 bits from current state
- Outputs
  - 16 bits for control signals
  - 4 bits for next state
- ROM Size
  - 2^{10} 20-bit words
  - Total size: 20 kbits