
SI232 Set #13:

Controlling the Single Cycle Implementation (Chapter Five)

1

ADMIN

- **Reading**
 - **Sections 5.5, 5.10, 5.11**
 - **Section 5.6 (first two pages)**

2

Control

Selecting the
and controlling the
based on the

- Outline:
 1. Overview
 2. Controlling the ALU
 3. Controlling multiplexors and register writes

3

Part 1: Control Overview

Example #1: `add $8, $17, $18`

000000	10001	10010	01000	00000	100000
op	rs	rt	rd	shamt	funct

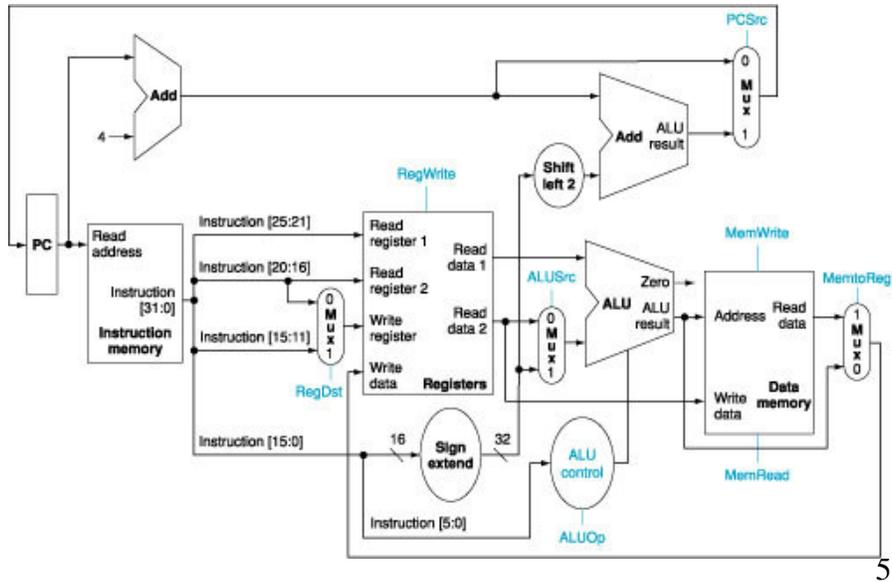
Example #2: `lw $1, 12($2)`

100011	00001	00010	0000000000001100
op	rs	rt	16 bit offset

	add \$8, \$17, \$18	lw \$1, 12(\$2)
A. What should the register file do?		
B. What should the ALU do?		
C. What should the muxes do?		

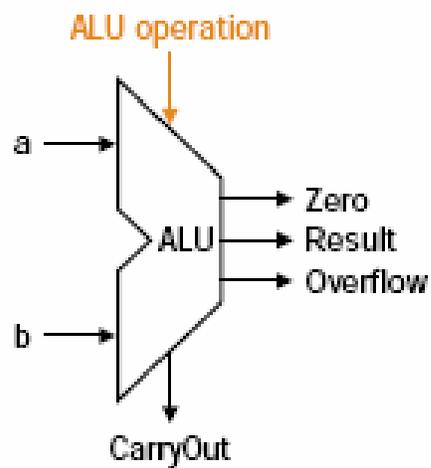
4

Part 1 – Control Overview



Recall: ALU Control and Symbol

ALU Control Lines	Function
0000	AND
0001	OR
0010	Add
0110	Subtract
0111	Set on less than
1100	NOR



Part 2: ALU Control Scheme

Instruction OpCode	Instruction op	Funct Field	Desired ALU action	ALU control input
lw (35)	load word	000000		
sw (43)	store word	000000		
beq (4)	branch equal	000000		
R-type (0)	add	100000		
R-type (0)	subtract	100010		
R-type (0)	AND	100100		
R-type (0)	OR	100101		
R-type (0)	Set on less than	101010		

7

Part 2: ALU Control

- Must describe hardware to compute 4-bit ALU control input given
 1. Instruction type
 - 00 = lw, sw
 - 01 = beq,
 - 10 = arithmetic
 2. Function code (for arithmetic)
- Describe it using a truth table:

ALUOp		Funct field						Operation
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	
0	0	X	X	X	X	X	X	0010
X	1	X	X	X	X	X	X	0110
1	X	X	X	0	0	0	0	0010
1	X	X	X	0	0	1	0	0110
1	X	X	X	0	1	0	0	0000
1	X	X	X	0	1	0	1	0001
1	X	X	X	1	0	1	0	0111

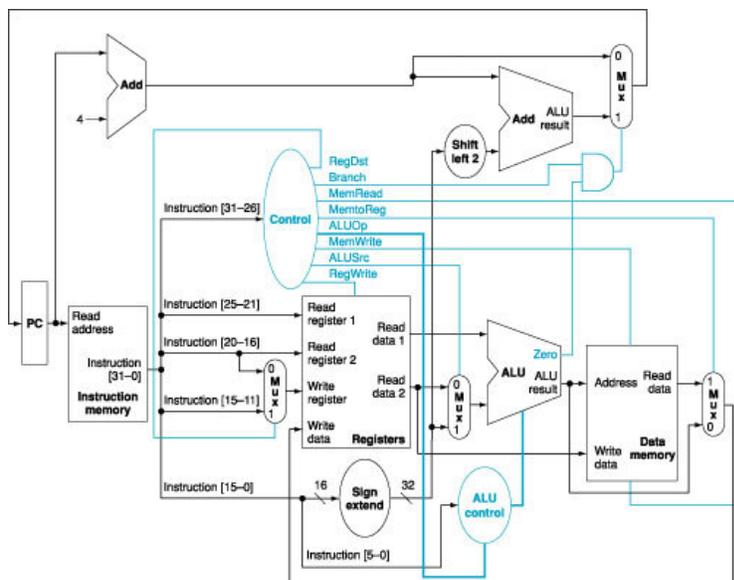
8

Part 3: Main Control

- Set the muxes and register write signals
 - To get data to flow to the right places
 - To store data in the appropriate places
- 7 signals:
 - ALUSrc
 - MemtoReg
 - MemRead
 - MemWrite
 - PCSrc
 - RegDst
 - RegWrite
- Control based on:

9

Part 3 – Main Control



10

Exercise #2

Fill in the correct signal name by looking back at the datapath diagram.

Possibilities: **ALUSrc, MemtoReg, MemRead, MemWrite, PCSrc, RegDst, RegWrite**

Signal Name	Effect when deasserted	Effect when asserted
	The register destination number for the write register comes from the rt field (bits 20-26)	The register destination number for the write register comes from the rd field (bits 15-11)
	None	The register on the Write register input is written with the value on the Write data input
	The second ALU operand comes from the second register file output (Read data 2)	The second ALU operand is sign-extended, lower 16 bits of the instruction
	The PC is replaced by the output of the adder that computes the value of PC+4	The PC is replaced by the output of the adder that computes the value of branch target
	None	Data memory contents designated by the address input are put on the Read Data output
	None	Data memory contents designated by the address input are replaced by the value on the Write Data input
	The value fed to the register Write data input comes from ALU	The value fed to the register Write data input comes from the data memory

13

Exercise #3 – Stretch

- Are all of the control signals needed? If not, how could you eliminate one or more?

14

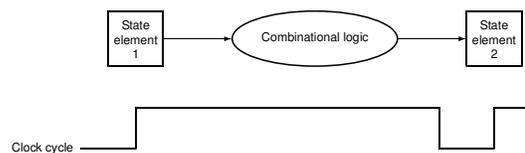
Exercise #4 – Stretch

- What is inefficient about our current design? How would you improve it?

15

Our Simple Control Structure

- All of the logic is combinational
- We wait for everything to settle down, and the right thing to be done
 - ALU might not produce “right answer” right away
 - we use write signals along with clock to determine when to write
- Cycle time determined by length of the longest path

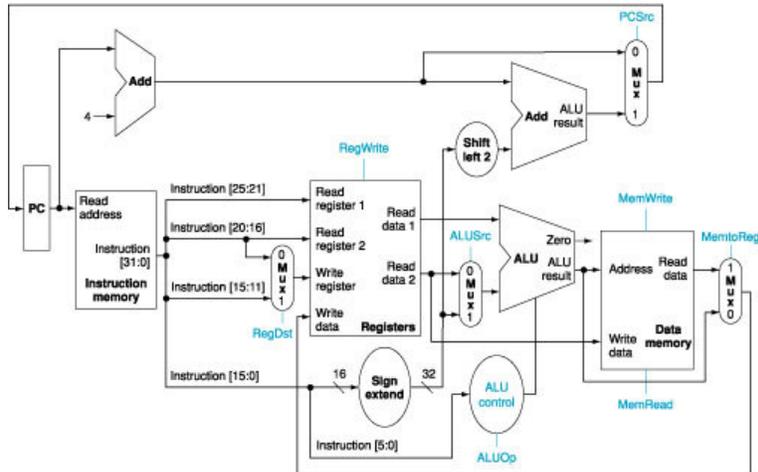


We are ignoring some details like setup and hold times

16

Performance

- Calculate cycle time assuming negligible delays except:
 - Memory (200ps), ALU and adders (100ps),
 - Register file access – read or write (50ps)



17

Performance Calculation

Instruction Class	Functional Units used by the instruction class				
R-type					
lw					
sw					
beq					

Memory (200ps)
 ALU and adders (100ps)
 Register file access – read or write (50ps)

Final Cycle Time?

18

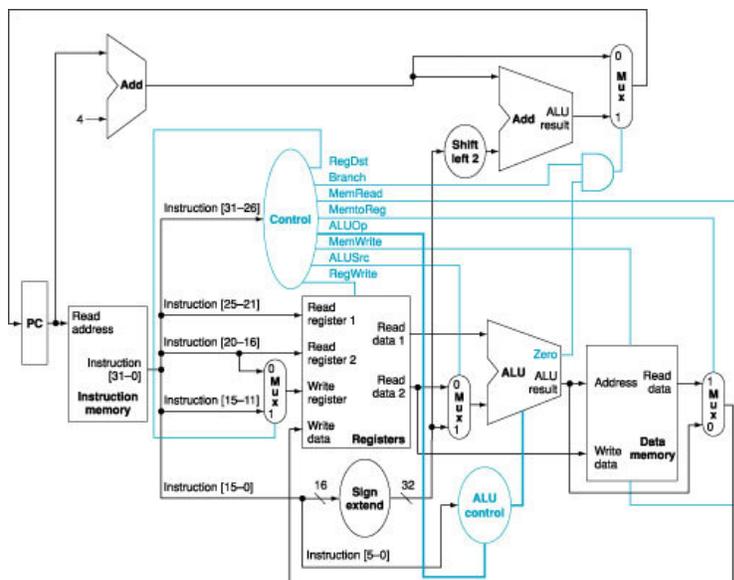
Evaluation – Single Cycle Approach

- Good:

- Bad:

19

Adding Jump?



20