

CS 240, Fall 2014 WELLESLEY CS

Control

A simple implementation

CS 240, Fall 2014 WELLESLEY CS

We built the instrument. Now we read music and play it...

Next: control

Datapath 17-2

CS 240, Fall 2014 WELLESLEY CS

ALU control signals

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR

A_{invert} B_{negate} Op
 (B_{invert} and LSB CarryIn)

Datapath 17-3

CS 240, Fall 2014 WELLESLEY CS

Control the ALU

Note sub-controller for ALU simplifies main control unit.

Control 18-4

CS 240, Fall 2014 WELLESLEY CS

Using ALUOp and Funct field to set ALU control bits

Instruction opcode	ALUOp	Instruction operation	Funct field	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	and	0000
R-type	10	OR	100101	or	0001
R-type	10	set on less than	101010	set on less than	0111

Control 18-5

CS 240, Fall 2014 WELLESLEY CS

From truth table to ALU control

ALUOp	ALUOp	F5	F4	F3	F2	F1	F0	Op
1	0	X	X	X	X	X	X	0010
0	1	X	X	X	X	X	X	0110
1	0	X	X	0	0	0	0	0010
1	X	X	X	0	0	1	0	0110
1	0	X	X	0	1	0	0	0000
1	0	X	X	0	1	0	1	0001
1	X	X	X	1	0	1	0	0111

Control 18-6

CS 240, Fall 2014 WELLESLEY CS

Main control unit

input: 6-bit opcode

output: 9 control bits

Opcode	Instruct
000000	R-format
100011	lw
101011	sw
000100	beq

Control 18-7

CS 240, Fall 2014 WELLESLEY CS

The data path with control signals

Control 18-8

CS 240, Fall 2014 WELLESLEY CS

We start with R-instructions*

Tells us what ALU to perform
 First operand
 Second operand
 Result goes here

Opcode (6) rs (5) rt (5) rd (5) shamt (5) funct (6)

Main Control

To read inputs of register file

To write address for register file

ALU Control

*R-type (for register) or R-format include `add`, `sub`, `and`, `or`, and `slt`.

Control 18-9

CS 240, Fall 2014 WELLESLEY CS

R-instructions data and control signals

Control 18-10

CS 240, Fall 2014 WELLESLEY CS

I-instructions come next

Tells us what ALU to perform
 Base register for lw & sw
 Value read or written for lw or sw
 Offset or address

Opcode (6) rs (5) rt (5) Constant (16)

Decoder generates controls

To read inputs of register file

Sign extended, multiplied by 4 and added to PC

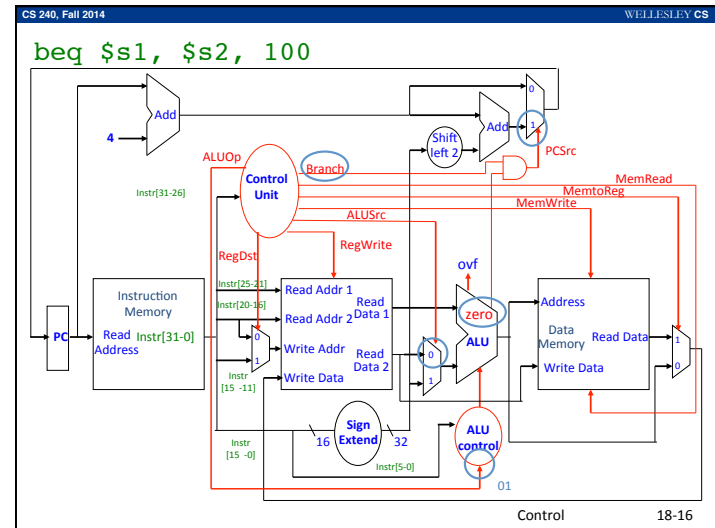
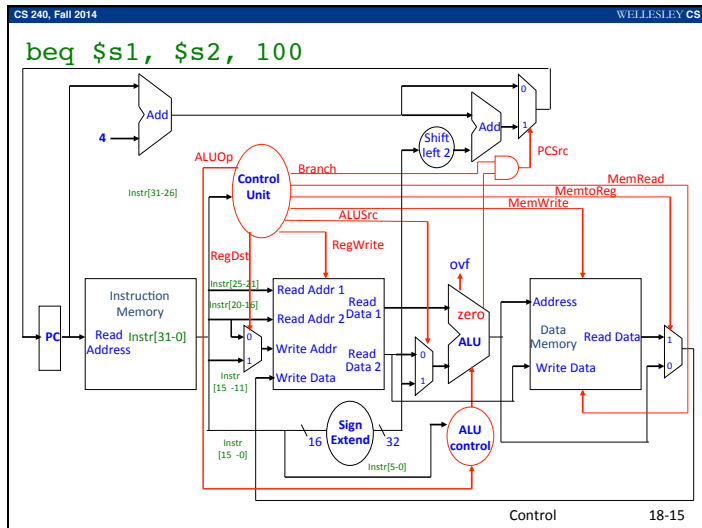
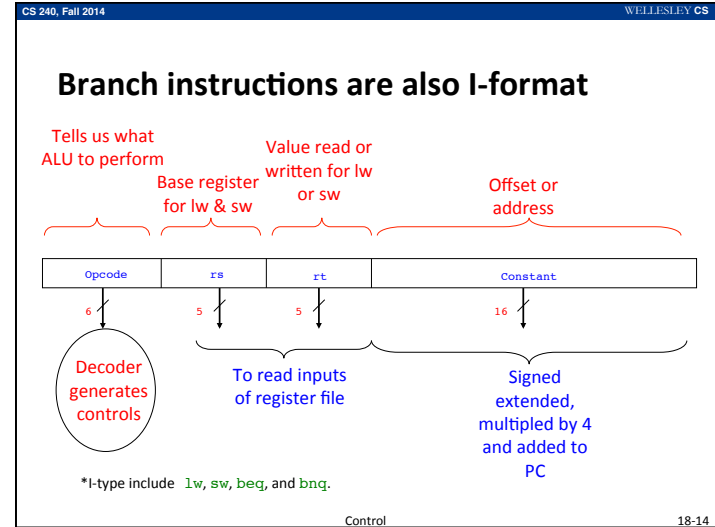
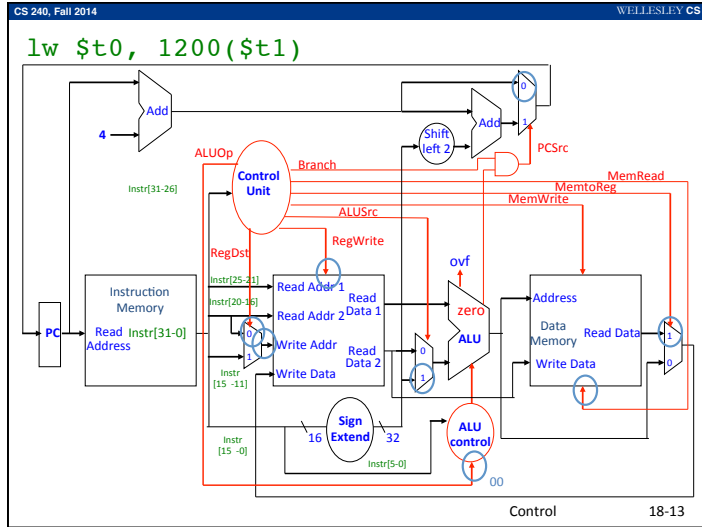
*I-type include `lw`, `sw`, `beq`, and `bnq`. Recall `lw $t0, 1200($t1)`.

Control 18-11

CS 240, Fall 2014 WELLESLEY CS

I-instructions data and control signals

Control 18-12



CS 240, Fall 2014 WELLESLEY CS

Control lines in sum

Instruction	RegDst	ALUSrc	Memto-Reg	Reg Write	Mem Read	Mem Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1

What about j?

Control 18-17

CS 240, Fall 2014 WELLESLEY CS

j 1000

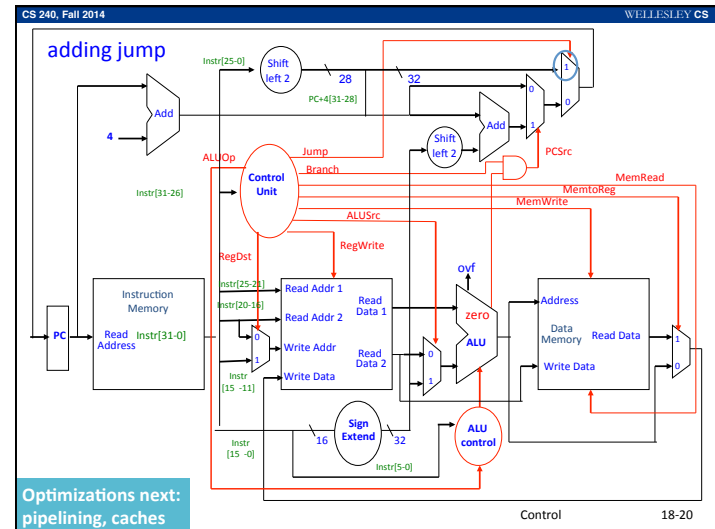
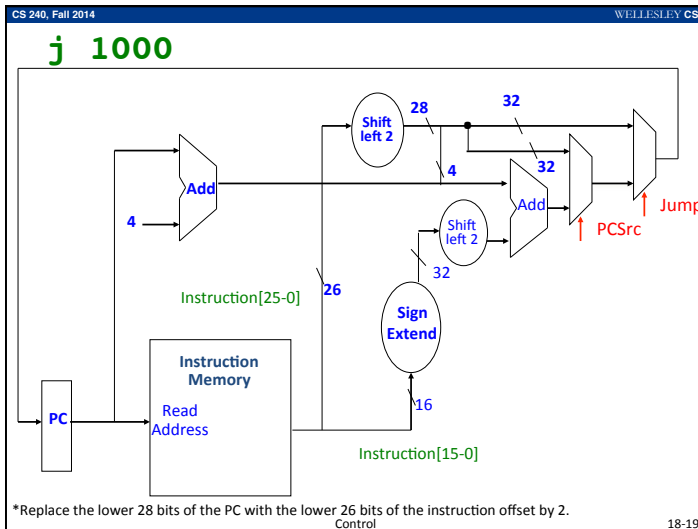
000010
00 0000 0000 0010 0101 0000 0000

6 bits
26 bits

still 32 bits

*Replace the lower 28 bits of the PC with the lower 26 bits of the instruction left shifted by 2 (multiplied by 4).

Control 18-18



CS 240, Fall 2014 WELLESLEY CS

Single cycle pros and cons

Con Inefficient: clock must be timed to accommodate the slowest instruction.

Con May be wasteful of area since some functional units (e.g., adders) must be duplicated since they can not be shared during a clock cycle

Big Pro Simple and easy to understand

Optimizations next: pipelining, caches control 18-21