

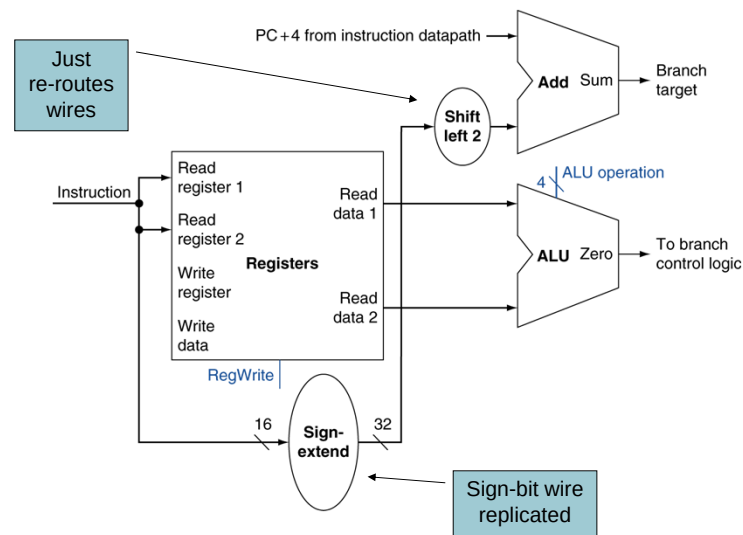
Branch Instructions

- Read register operands
- Compare operands `beq $t1, $t2, offset`
 - Use ALU, subtract and check Zero output
- Calculate target address
 - Sign-extend displacement
 - Shift left 2 places (word displacement)
 - Add to PC + 4
 - Already calculated by instruction fetch



Chapter 4 — The Processor — 33

Branch Instructions



Chapter 4 — The Processor — 34

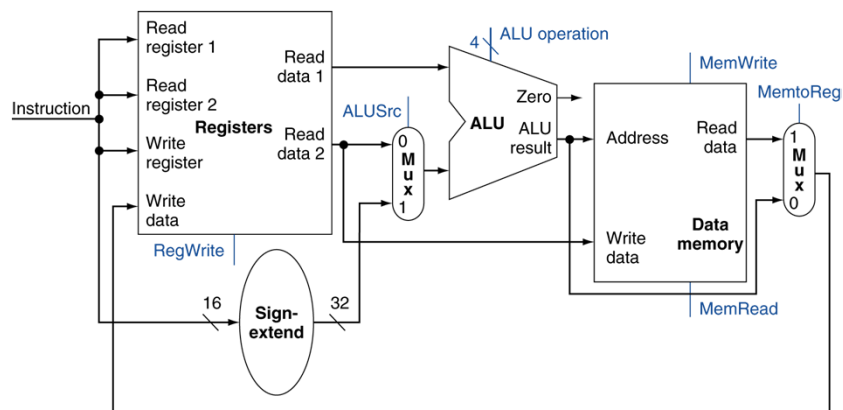
Composing the Elements

- First-cut data path does an instruction in one clock cycle
 - Each datapath element can only do one function at a time
 - Hence, we need separate instruction and data memories
- Use multiplexers where alternate data sources are used for different instructions



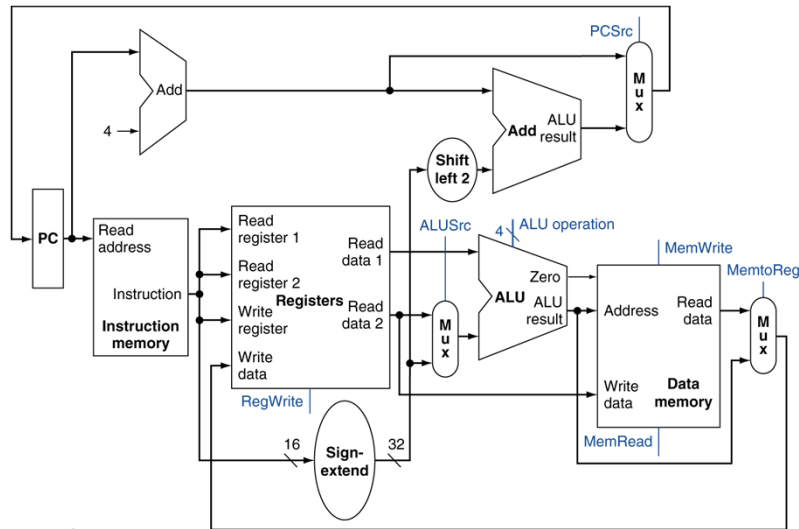
Chapter 4 — The Processor — 35

R-Type/Load/Store Datapath



Chapter 4 — The Processor — 36

Full Datapath



Chapter 4 — The Processor — 37

ALU Control

- ALU used for
 - Load/Store: F = add
 - Branch: F = subtract
 - R-type: F depends on funct field

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

Chapter 4 — The Processor — 38

ALU Control

- Assume 2-bit ALUOp derived from opcode
 - Combinational logic derives ALU control

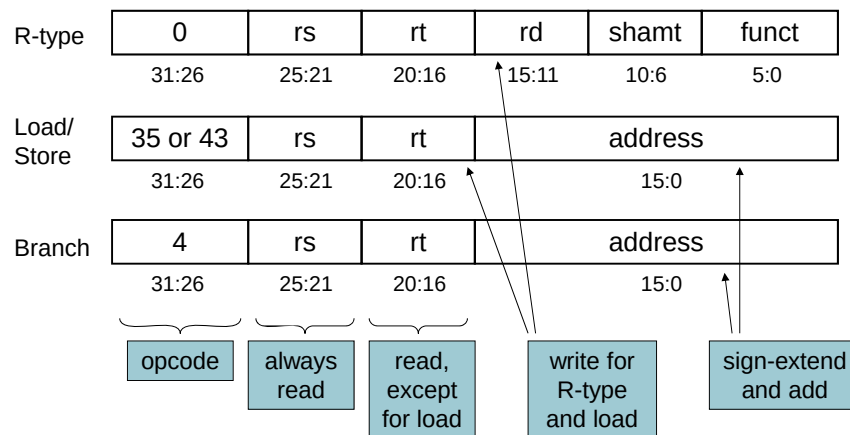
opcode	ALUOp	Operation	funct	ALU function	ALU control
lw	00	load word	XXXXXX	add	0010
sw	00	store word	XXXXXX	add	0010
beq	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
		subtract	100010	subtract	0110
		AND	100100	AND	0000
		OR	100101	OR	0001
		set-on-less-than	101010	set-on-less-than	0111



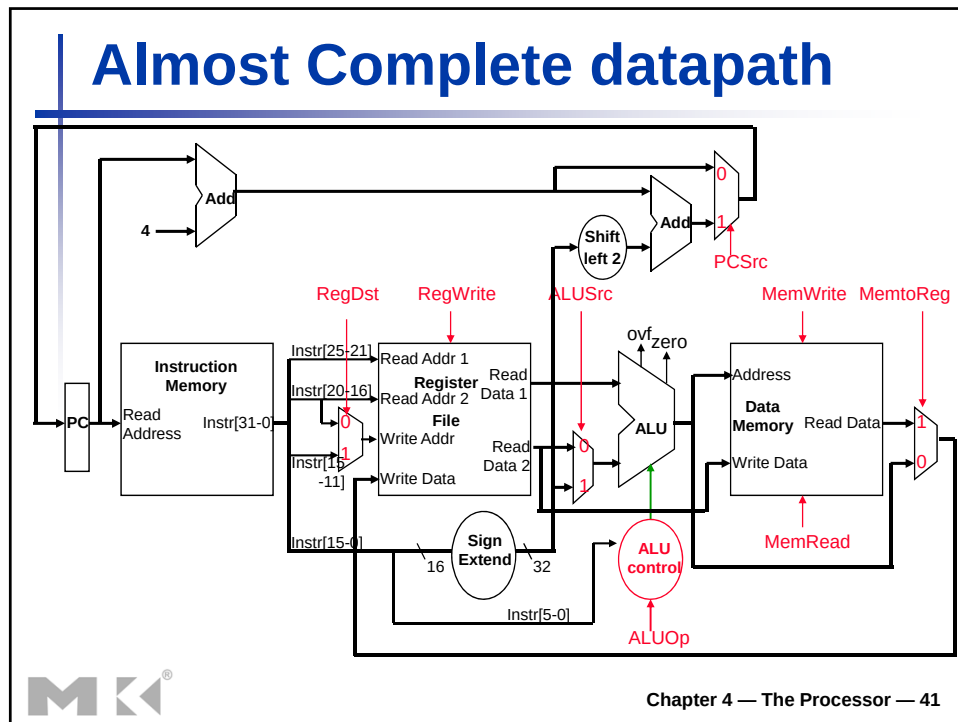
Chapter 4 — The Processor — 39

The Main Control Unit

- Control signals derived from instruction



Chapter 4 — The Processor — 40



Control signals

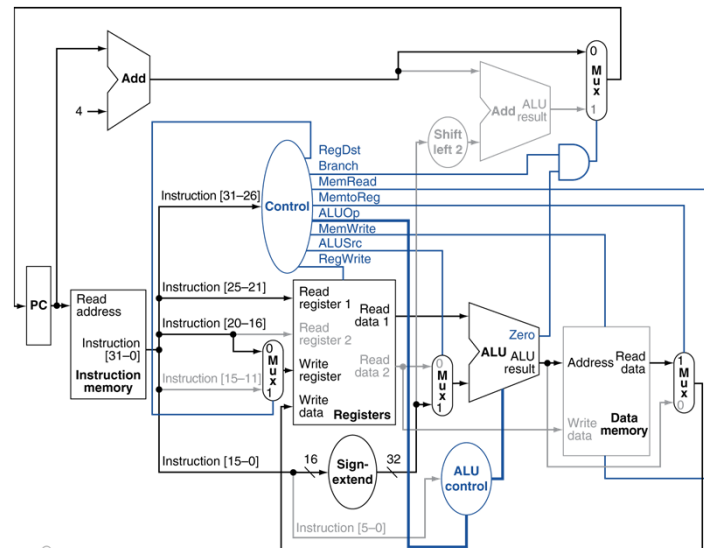
Signal name	Deasserted	Asserted
RegDst	Reg dest number is from rt bits[20:16]	Reg dest number comes from bits[15:11]
RegWrite	NON	Data is written in the register specified by the write reg number
ALUSrc	Register file (2 nd operand)	Sign extended immediate
MemRead	NON	Data memory → Read Data Output
MemWrite	NON	Write data Input → memory
memtoReg	ALU → Register file	Memory → Register file

Chapter 4 — The Processor — 42

Chapter 4 — The Processor — 43

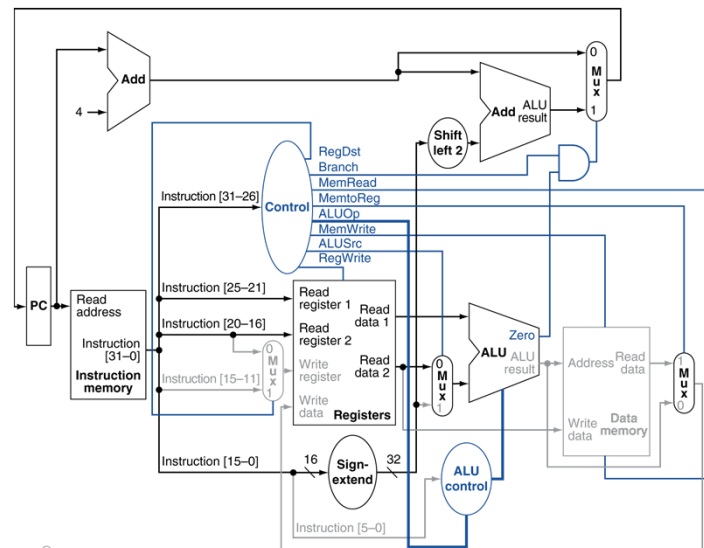
Chapter 4 — The Processor — 44

Load Instruction



Chapter 4 — The Processor — 45

Branch-on-Equal Instruction



Chapter 4 — The Processor — 46

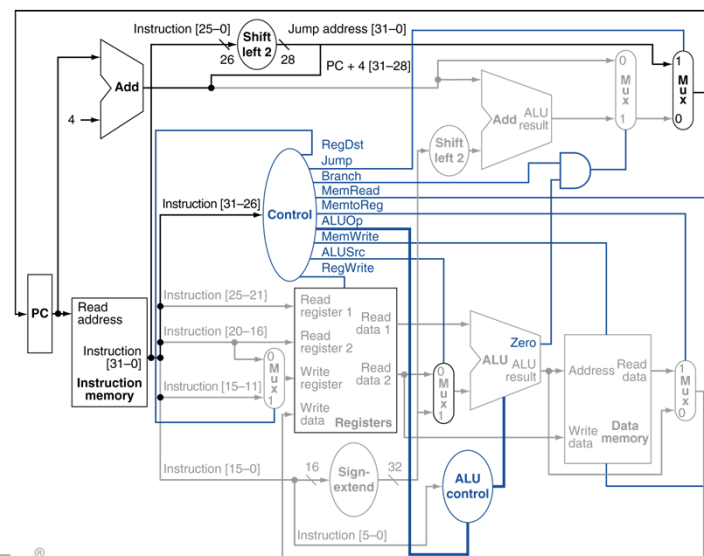
Implementing Jumps

- Jump
- | | |
|-------|---------|
| 2 | address |
| 31:26 | 25:0 |
- Jump uses word address
 - Update PC with concatenation of
 - Top 4 bits of old PC
 - 26-bit jump address
 - 00
 - Need an extra control signal decoded from opcode



Chapter 4 — The Processor — 47

Datapath With Jumps Added



Chapter 4 — The Processor — 48

Performance Issues

- Longest delay determines clock period
 - Critical path: load instruction
 - Instruction memory → register file → ALU → data memory → register file
- Not feasible to vary period for different instructions
- Violates design principle
 - Making the common case fast
- We will improve performance by pipelining

