Designing a Single-Cycle Implementation

Tom Kelliher, CS 220

Nov. 14, 2011

1 Administrivia

Announcements

Assignment

Read 4.5.

From Last Time

Logic design, overview of MIPS datapath.

Outline

1. Designing separate datapaths.
2. Merging the datapaths.
3. The control unit.

Coming Up

A look at pipelining.

2 Designing Separate Datapaths

Strategy: Look at the major datapath components needed to execute each class of instructions.

Hardware needed:

1. Individual registers (PC).
2. Register file.
3. Memory (data memory).
4. ALU.
5. Sign extend, shift?

2.1 Instruction Fetch

Common to all instruction formats.

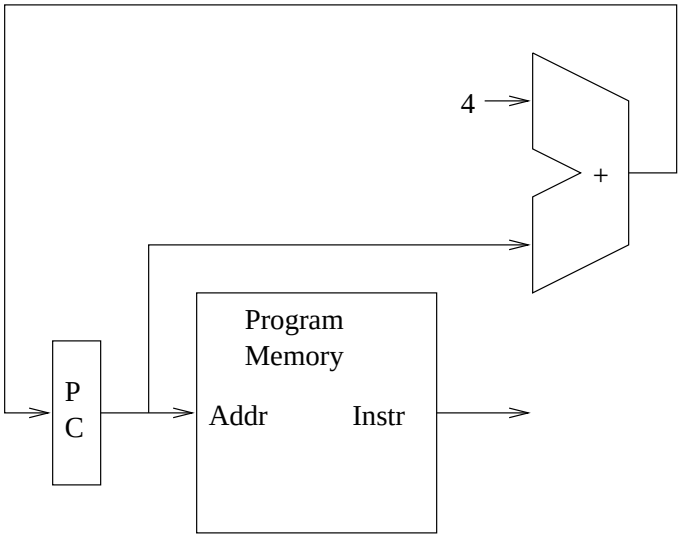
Requirements:

1. Store program.
2. Remember which instruction is to be executed next.
3. Fetch next instruction. (And store.)

Elements:

1. A register for the PC.
2. A memory for the program.
3. An adder to update the PC.

Organization:



The adder is a hardwired ALU.

2.2 R-Format

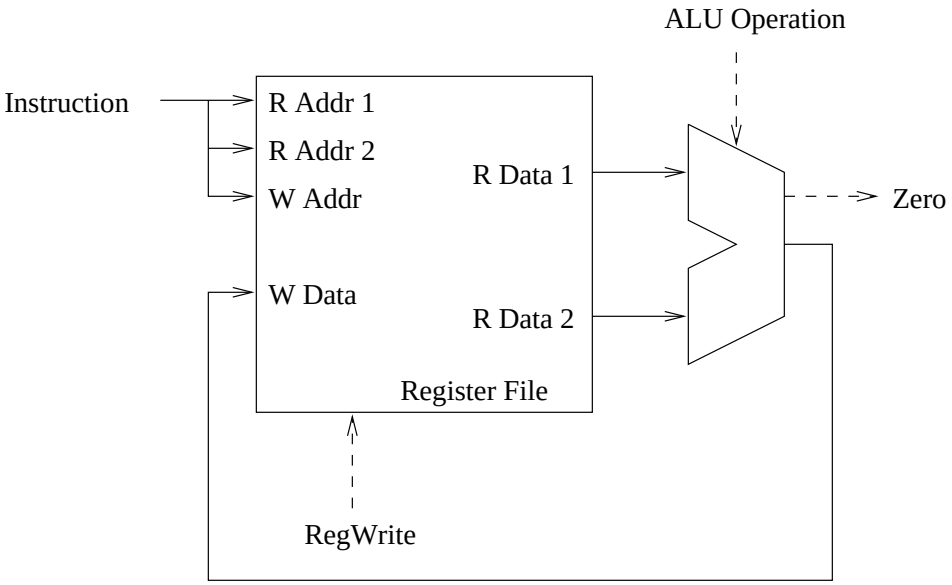
Requirements:

1. Fetch two register operands.
2. Operate.
3. Store result back into a register.

Elements:

1. A 32 word by 32 bit register file with two read ports and a write port.
2. An ALU.

Organization:



Control signals: Write, ALU Control, Zero.

2.3 I-Format

Requirements:

1. Memory reference instructions:
   (a) Add base register and sign-extended offset.
   (b) Transfer data
2. Branch instructions:
   (a) Compare register values.
   (b) Conditionally update PC with PC + 4 + sign-extended, shifted offset.

Elements: R-format elements plus:

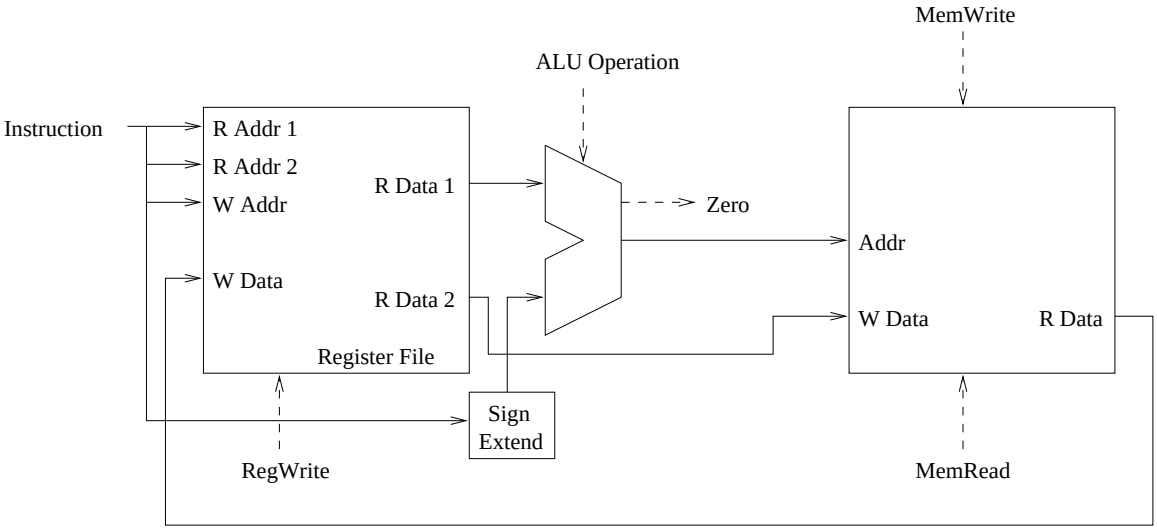
1. Substitution of a register source value with the immediate value.

2. A data memory.

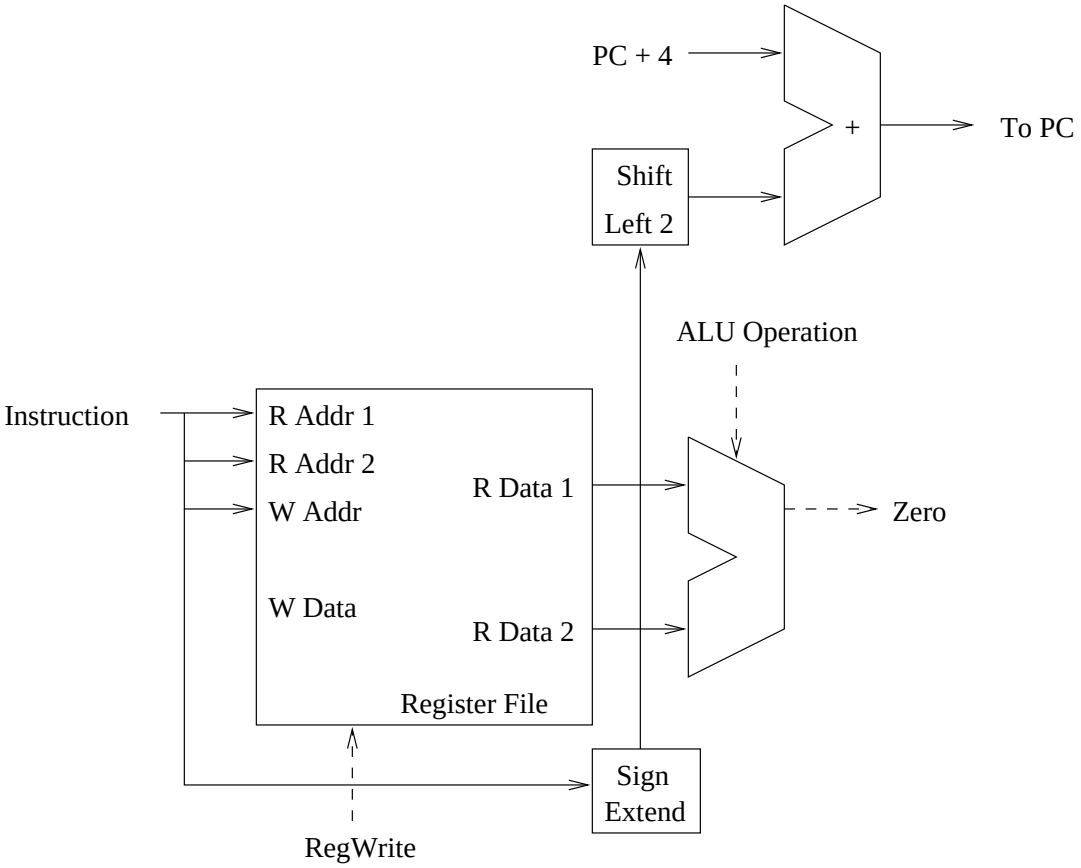
3. A mechanism for loading a branch target address into the PC.

Organizations:

Memory reference instructions:

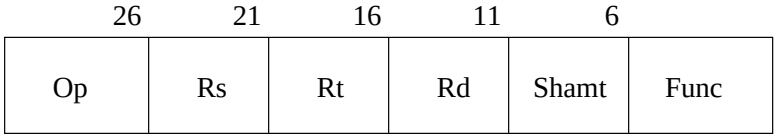


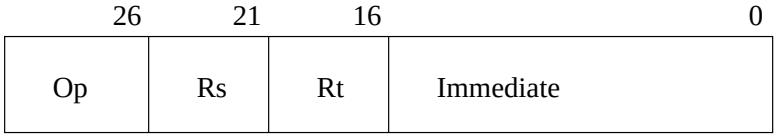
Branch instructions:



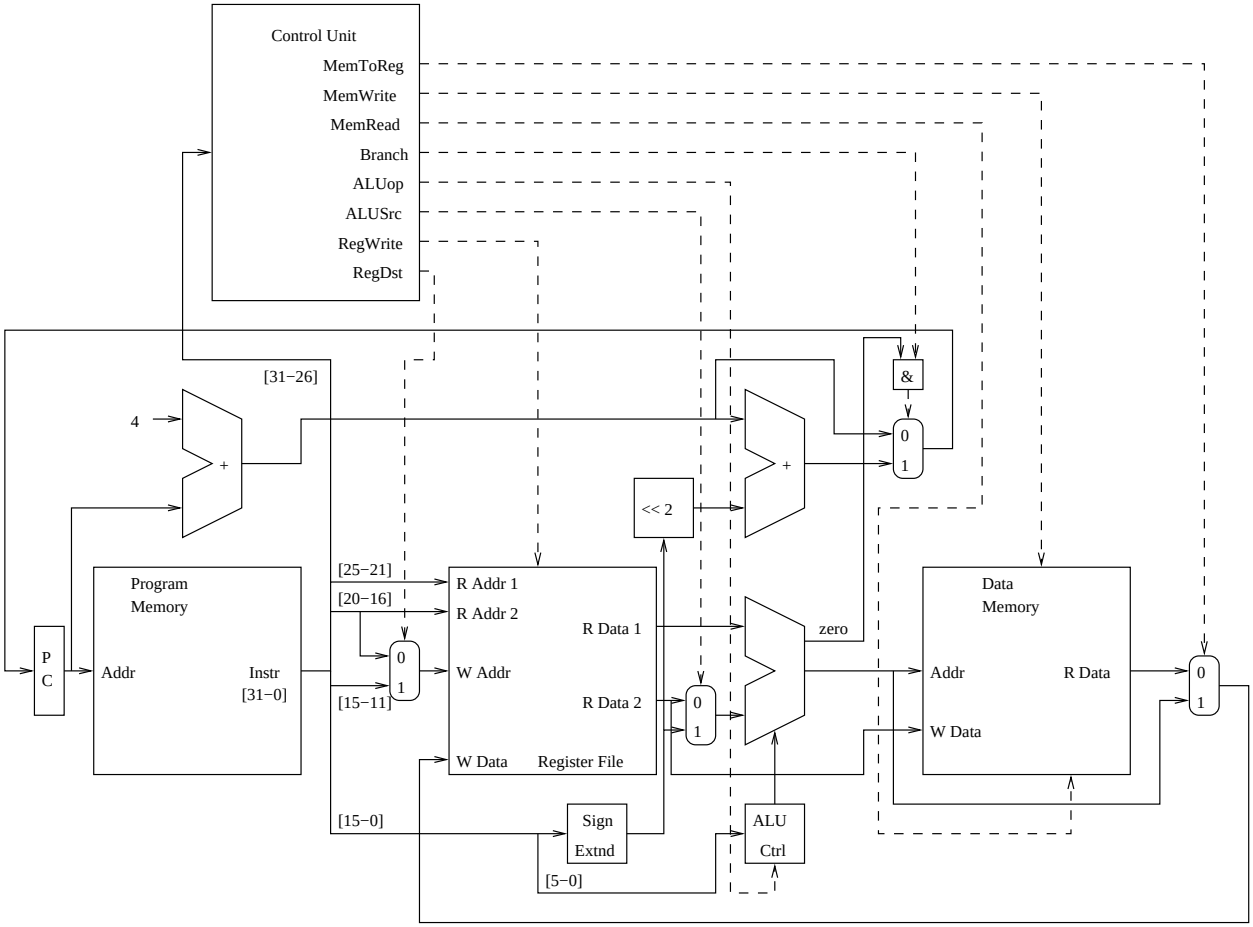
3 Merging the Data Paths

Recall instruction formats:





Merged datapath, with control shown:



3.1 Control signals

1. RegDst — selects rt or rd field as write address.
2. RegWrite — write enable.
3. ALUSrc — selects rd2 or immediate data.
4. PCSrc/Branch — selects PC + 4 or branch target.
5. MemRead — read enable.
6. MemWrite — write enable.
7. ALUOp — Two bits. Add (00), subtract (01) , or refer to FUNC field (10).
8. MemToReg — selects ALU output or memory data to register file write data port.

4 The Control Unit

1. Is it combinational or sequential?

2. Why are its only inputs the opcode bits?

How should the control signals be set (0, 1, x) for each of the following?

1. R-format instructions.

2. lw.

3. sw.

4. beq.