Controlling the Multi-cycle Implementation

Tom Kelliher, CS 220

Nov. 12, 2007

1 Administrivia

Announcements

Assignment

Read 5.6 and 6.1.

From Last Time

Multi-cycle datapath.

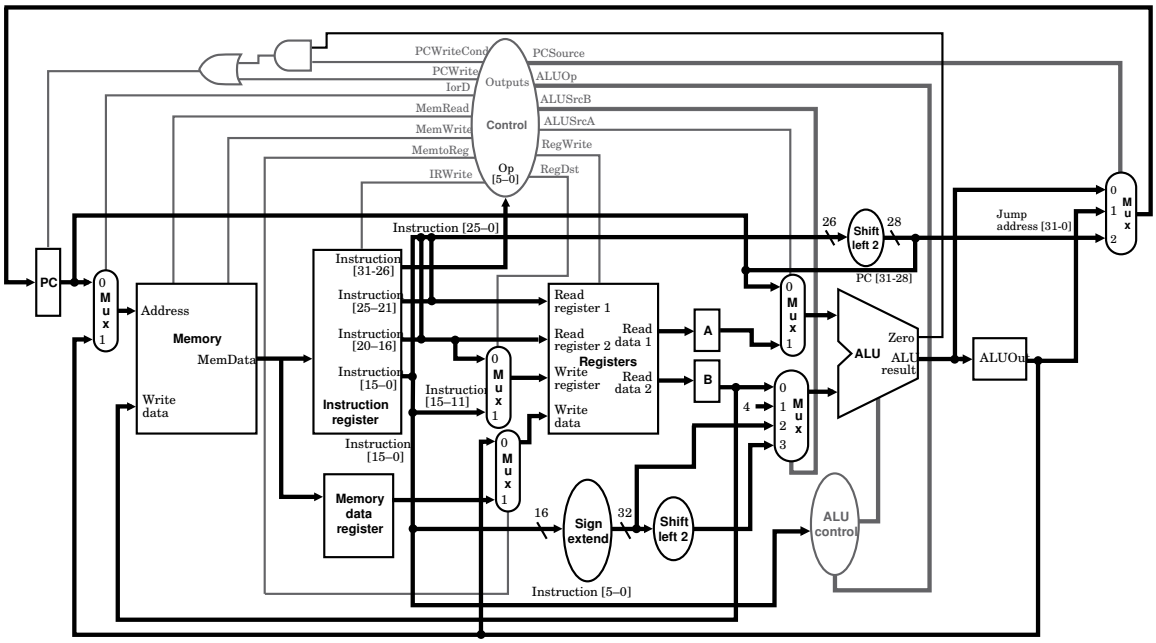
Outline

1. The complete multi-cycle datapath and control signals.
2. Class exercise to design individual state machines to handle the following instructions: R-format, beq, lw, sw, jump.
3. Development of the overall state machine.

Coming Up

Exceptions; introduction to pipelining.

2 The Complete Datapath, Revisited



Control signals:

1. IRWrite: Write a value from memory to the instruction register.
2. RegDst, RegWrite.
3. ALUSrcA: Choose between PC and Rs.
4. MemRead, MemWrite, MemtoReg.
5. IorD: Choose between PC and ALUOut for memory address.
6. PCWrite: Load a new value into PC.
7. PcWriteCond: Load a new value into PC if zero is active.
8. ALUOp.
9. ALUSrcB: Choose between Rt/Rd, 4, sign-extended immediate, sign-extended shifted immediate.
10. PCSource: Choose between PC + 4, ALUOut (branch target address), jump address

3 The Instruction Cycle

1. Steps: Fetch, decode, execution/completion.

2. Instructions: R-format, memory reference, conditional branch, jump.

3.1 Fetch

Common to all instructions.

1. Load IR.

2. Increment PC.

3.2 Decode

Common to all instructions.

1. Load A (rs field) and B (rt field) from register file.

2. Load ALUOut with branch target.

These are "optimistic" optimizations which *do no harm*, even if they are not needed/do not make sense for the current instruction.

3.3 Execution, Memory Address Computation, Branch Completion

Instruction classes go their own way.

1. R-format:

(a) Perform ALU operation, loading ALUOut.

2. Memory reference:

   (a) Use ALU to compute sum of base & offset, loading ALUOut.

3. Conditional branch:

   (a) If `zero` load PC with computed branch target in ALUOut.

   This instruction has completed.

4. Jump:

   (a) Load PC with jump address.

   This instruction has completed.

3.4 R-Format Completion, Memory Access

Just two instruction classes remaining.

1. R-format:

   (a) Load register file (rd field) from ALUOut.

   This instruction has completed.

2. Memory reference:

   (a) sw: Memory location specified by address in ALUout written with value stored in B.

   This instruction has completed.

   (b) lw: MDR loaded from memory location specified by address in ALUOut.

3.5 LW completion

1. Load register file (rt field) from MDR.

4 Designing the Control Unit

Design the state machine necessary for controlling the datapath.