CCOM 4086 — Exam 2 — May 17, 2004

- **Due by the end of our final period (May 26, 3:45PM)**

1 Introduction

We are going to design a CPU. The CPU will have 16 bit registers and a 16 bit address space. The CPU will have 16 general purpose registers, and no more than 16 instructions.

1.1 R-Format

The R format instructions have a 4 bit opcode, and 3 4 bit register specifications, so `add $2, $3, $4` adds the contents of registers 3 and 4 and places the result in register 2. The machine code will look like:

Op	rd	rs	rt
0000	0010	0011	0100

The remaining R format instructions and their opcodes are:

add	0000
sub	0001
lw	0010
sw	0011

the lw and sw operands compute their target addresses as rs+rt.

1.2 I-Format

I format instructions have the 4 bit opcode, one register and an 8 bit immediate operand:

addi	1000
ldhi*	1001
bz	1010
bnz	1011

The machine code for `addi $2, -3` is:

Op	rd	imm_8
1000	0010	111111101

The ldhi* instruction loads the 8 bit immediate into the upper 8 bits of a 16 bit register. **Implementing ldhi is extra credit.** bz and bnz branch to $\text{PC} + 4 \times \text{imm}_8$ if the named register contians (does not contain) zero.

1.3 J-Format

The final instruction format is the J. A jump instruction has a 4 bit opcode and a 12 bit target address. On a jump, control passes to $\text{PC} + 4 \times \text{imm}_{12}$.

Op	imm_{12}
1111	000000001000

In this exam we will design the full CPU, with pipelining, datapath, and control.

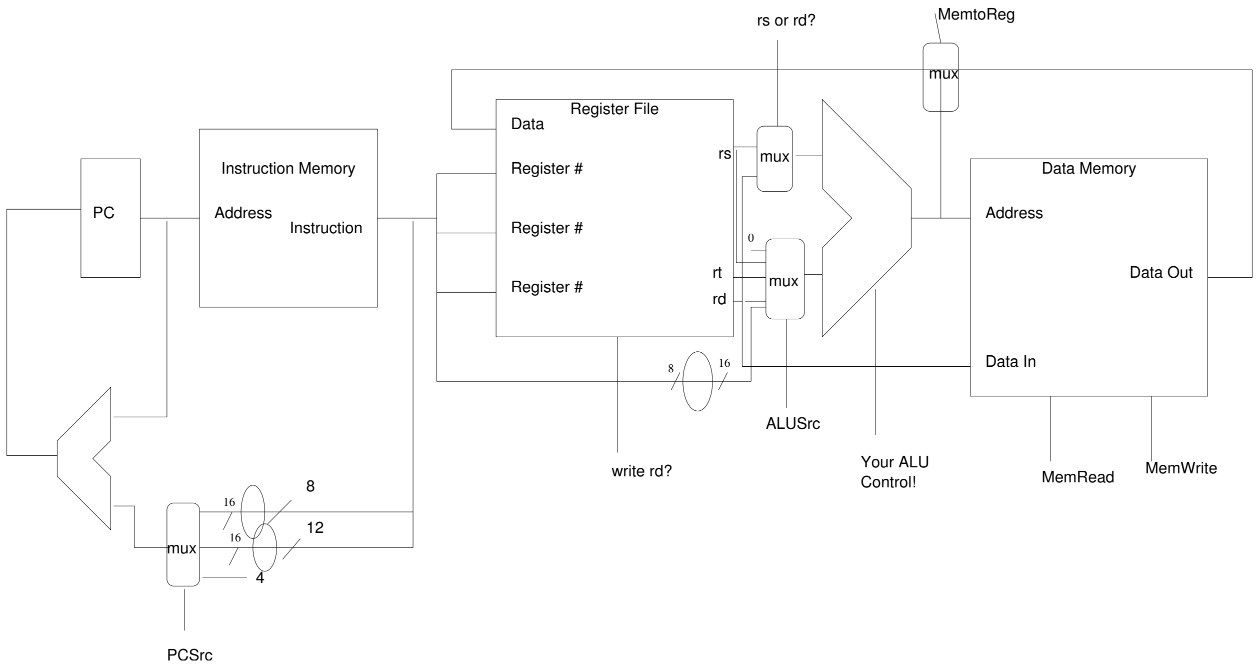
2 Questions

1. Build the 16 bit ALU, including control, to support all our opcodes, (*i.e.,* add, subtract, test for zero, test for not zero). You may use full adders as your building block, but show all additional components.
2. Build the full datapath for the datapath outlined below. Plug your ALU in place. The completed CPU should be similar to Figure 5.29 (See Lecture 14) with the appropriate modifications for our simpler architecture.
3. Specify the full control table for your cpu, and remember to set up the control for the ALU you designed. You may leave out the control for the ldhi instruction (or turn it in for extra credit).
4. Build a 5 stage pipeline for this processor, show the pipeline registers, the sources for the functional units, and the proper control termination. Your completed figure should look like Figure 6.30, (see lecture 18) with appropriate modifications.
5. A Harvard architecture machine has separate instruction and data memories, like ours, and is simpler to implement than a von Neuman architecture that has a single, unified, memory. Show how to simulate a Harvard architecture machine on a machine with one single RAM area, using caches.

3 Extra Credit

1. Construct the datapath and control for the ldhi instruction that loads the upper half of the register with the imm_8 and leaves the lower half unchanged.
2. Show that this processor can compute any function a Pentium 4 can compute. (It's a trick question, you have to prove that both processors are equivalent.)

4 Simplified CPU


rs or rd?
MemtoReg
mux
Register File
Data
Register #
Register #
Register #
rs
rt
rd
mux
0
mux
Instruction Memory
Address
Instruction
PC
Data Memory
Address
Data Out
Data In
8
16
ALUSrc
Your ALU Control!
write rd?
MemRead
MemWrite
16
16
8
12
4
mux
PCSrc