Announcements

The midterm is Friday
Bring a colored pencil or highlighter

Overview of Content

Basics: Representations -- binary, hex, 2’s complement, ASCII, strings, floating point single/double precision
Ex: What binary number is the hex BEEF 2ABE?

Machine Language: data types, instruction types, MIPS instruction repertoire, alignment
Ex: Load FAB4 into the first argument register
Ex: Let $s0 contain a negative number; write code to shift it right 3 bits, keeping it negative

Assembly Language: register designations, stack, calling conventions, pseudo instructions
Ex: How is bge $4, $5, Address implemented?
Assembly Language Programming

Write short code segments in MIPS Assembly

Hints:

- Plan out 2 basic structure (if-thens, basic loops)
- Write intended code in C before writing Assem
- Comment profusely

```
int fact(int n) {
    int i, f = 1;  /* set i to n; i > 1; i-- */
    f = f * i;  /* set f to 1; */
    for (; i > 1; i--) {  /* exit if done */
        f *= i;
    }
    return f;  /* build factorial */
}
```

Design of 1-cycle MIPS

MIPS design:

- Know all components and their operation
- Know flow of logic -- which components are active when implementing a given instruction
- Be able to specify control signals needed to accomplish specific instructions
- Know why the 1-cycle design underperforms
Final 1-cycle design

Instruction Interpretation
Multicycle design

Regarding the multicycle design

- How did we segue from 1-cycle design to this?
- Added/changed components from 1-cycle design
- Know multicycle design operation for each instruction, and give control settings to make it happen
Stage 1: Instruction fetch & PC increment

IR = Mem[PC]

PC = PC + 4

Register File Read
Stage 3 (R-type): instruction execution

- Save the result in ALUOut
- Take the ALU result from the last cycle...

Stage 4 (R-type): write back

- ...and store it to register "rd"
- Take the ALU result from the last cycle...
Stage 2: Reg fetch & branch target

Stage 3 (beq): Branch completion
Stage 3 (sw): compute effective address

Stage 4 (sw): memory write

...to store data from one of the registers...
Stage 4 (lw): memory read

...to read data from memory...

Use the effective address from stage 3...

...into MDR.

Stage 5 (lw): register write

...and store it in register rt.

Take MDR...