Milestone #5

Milestone Overview

You will develop the initial stages of a pipelined MIPSLite microprocessor.

This project milestone is worth 40 points. For full credit, your solution must be completed no later than Thursday, March 2. A 10% penalty will be applied for each day that your solution is late (based on the date when your files are submitted via the "handin" system).

Milestone Deliverables

The deliverable for this milestone is:

    machine.c

Be sure to use the specified file name, and to submit your file for grading via the "handin" system.

Milestone Specifications

The MIPSLite microprocessor recognizes 31 machine language instructions:

- R-format shift instructions: \{SLL, SLLV, SRL, SRLV, SRA, SRAV\}
- R-format data manipulation: \{ADD, SUB, AND, OR, XOR, NOR, SLT\}
- I-format data manipulation: \{ADDI, SLTI, ANDI, ORI, Xori, LUI\}
- I-format data movement: \{LW, SW\}
- Control-transfer instructions: \{JR, JALR, J, JAL, BEQ, BNE, BLTZ, BLEZ, BGTZ, BGEZ\}

For this milestone, you will revise an instructor-supplied skeleton to begin your development of a pipelined implementation of the MIPSLite microprocessor.

Create a new project subdirectory under your account and copy the following files:

    /user/cse420/Project/Milestone05/machine*

The instructor has supplied complete and correct versions of the following:

    /user/cse420/Project/Milestone05/ClockUnit.o
    /user/cse420/Project/Milestone05/FetchUnit.o
    /user/cse420/Project/Milestone05/ControlUnit.o
    /user/cse420/Project/Milestone05/RegisterUnit.o
    /user/cse420/Project/Milestone05/DisplayUnit.o

The Clock Unit, Fetch Unit and Control Unit behave as specified in the previous milestones.

The Register Unit uses the register numbers encoded in machine language instructions, along with control signals from the Control Unit, to access the general-purpose registers and selectively update them. The Display Unit displays the contents of the pipeline registers in the terminal window.

Edit your copy of "machine.c" (an incomplete version of the MIPSLite microprocessor) to implement the missing functionality.
The following diagram gives the basic structure of a pipelined implementation of the MIPSLite microprocessor.

1. The IF/ID pipeline register will capture all outputs of the Fetch Unit.
2. The ID/EX pipeline register will capture all of the bits from the IF/ID pipeline register, as well as the output of the Control Unit and the output of the Register Unit (the 32-bit contents of registers "rs" and "rt").
3. The pipeline registers have the same signals as both inputs and outputs (the registers simply capture and hold the signals temporarily). Of course, the "Register" component in "sim" requires different names for the inputs and outputs, so you will need to adopt a reasonable naming convention.
4. Use probes to display the contents of the pipeline registers. It will be necessary for you to read and interpret the values displayed on the probes throughout the semester, so use a reasonable and readable layout.
5. Use a temporary signal (with the value 0x1) to enable the PC and pipeline registers.
6. Use the correct bits from the instruction in the IF/ID pipeline register to select the two registers which will be the outputs of the Register Unit.
7. Use the correct bits from the instruction in the ID/EX pipeline register, as well as the control signals in that pipeline register, to selectively update the Register Unit.
8. Use a temporary signal (with the value 0x89abcdef) as the operand to be selectively written into the Register Unit.