Milestone #7

Milestone Overview

You will augment your pipelined MIPSlite microprocessor so that it correctly processes the control-transfer instructions.

This project milestone is worth 40 points. For full credit, your solution must be completed no later than Thursday, March 23. 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 deliverables for this milestone are:

BranchUnit.c
machine.c

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

Milestone Specifications

The MIPSlite microprocessor recognizes 10 control-transfer instructions:

\{JR, JALR, J, IAL, BEQ, BNE, BLTZ, BLEZ, BGTZ, BGEZ\}

For this milestone, you will add the Branch Unit to your pipelined machine from Milestone #6 so that it correctly processes the control-transfer instructions, although it need not detect or resolve control hazards.

Create a new project subdirectory under your account and copy the appropriate files, including your "machine.c" and the following instructor-supplied files:

/user/cse420/Project/Milestone07/machine*

The following instructor-supplied files are available but must not be copied into your account:

/user/cse420/Project/Milestone07/ClockUnit*
/user/cse420/Project/Milestone07/FetchUnit*
/user/cse420/Project/Milestone07/ControlUnit*
/user/cse420/Project/Milestone07/RegisterUnit*
/user/cse420/Project/Milestone07/DisplayUnit*
/user/cse420/Project/Milestone07/ShiftUnit*
/user/cse420/Project/Milestone07/MathUnit*
/user/cse420/Project/Milestone07/BranchUnit.h

The first seven components behave as specified in the previous milestones.
The following diagram gives the structure of a pipelined implementation of the MIPSlite microprocessor.

In the Fetch Unit, the PC is initialized to zero before the "Running" signal is asserted. Using similar logic, the following fields within the pipeline registers must be initialized to zero:

a) the instruction (32 bits) in the IF/ID pipeline register
b) the instruction (32 bits) and control signals (18 bits) in the ID/EX pipeline register

The inputs to the Branch Unit are:

a) the machine language instruction (32 bits)
b) the incremented PC (32 bits)
c) the contents of REG[rs] (32 bits)
d) the zero flag from the Math Unit (1 bit)

The outputs from the Branch Unit are:

a) the target address for the Fetch Unit (32 bits)
b) the control signal for the Fetch Unit (1 bit)
c) the return address (32 bits)

The Branch Unit will decode the machine language instruction; if the instruction is a control-transfer instruction, the Branch Unit will generate the correct output signals.

For JALR and JAL instructions, the return address is the location of the instruction following the instruction in the delay slot.

For JR and JALR instructions, the target address is contained in the register specified by the "rs" field.

For J and JAL instructions, the target address is derived from the least significant 26 bits of the instruction.

For conditional branch instructions, the target address is derived from the least significant 16 bits of the instruction and is relative to the address of the instruction in the delay slot. The most significant bit of the register specified by the "rs" field and the zero flag from the Math Unit will be used to determine whether the branch condition is true or false.