CSE 420 Spring, 2017
Semester Project

Milestone #8

Milestone Overview

You will augment your pipelined MIPSlite microprocessor so that it correctly processes the data movement instructions.

This project milestone is worth 40 points. For full credit, your solution must be completed no later than Thursday, April 6. 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:

    MemoryUnit.c
    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 two data movement instructions (LW and SW). Those instructions are used to transfer data between registers and the data memory (RAM). The SW instruction is also used to transfer data to two output devices.

For this milestone, you will add the Memory Unit to your pipelined machine from Milestone #7 so that it correctly processes the data movement instructions, although it need not detect or resolve data hazards.

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

    /user/cse420/Project/Milestone08/machine*

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

    /user/cse420/Project/Milestone08/ClockUnit*
    /user/cse420/Project/Milestone08/FetchUnit*
    /user/cse420/Project/Milestone08/ControlUnit*
    /user/cse420/Project/Milestone08/RegisterUnit*
    /user/cse420/Project/Milestone08/DisplayUnit*
    /user/cse420/Project/Milestone08/ShiftUnit*
    /user/cse420/Project/Milestone08/MathUnit*
    /user/cse420/Project/Milestone08/BranchUnit.h
    /user/cse420/Project/Milestone08/MemoryUnit.h

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

![Diagram of a pipelined implementation of the MIPSlite microprocessor](image)

The inputs to the Memory Unit are:

a) clock pulse for synchronization (1 bit)
b) asserted when instruction reads from the Memory Unit (1 bit)
c) asserted when instruction writes into the Memory Unit (1 bit)
d) effective address (32 bits)
e) operand to be written by SW instructions (32 bits)

The output from the Memory Unit is the operand to be read by LW instructions (32 bits).

Partition 0x0000E000 in the address space represents the data memory. Data movement instructions with effective addresses in the set \{0x0000E000, 0x0000E004, 0x0000E008, \ldots, 0x0000EFFC\} are interpreted as loads from or stores to the data memory. Use a "Ram" component for the data memory, where bits 11:2 of the effective address identify the specific word being accessed (the data memory contains 1024 words of storage).

The MIPSlite microprocessor uses memory-mapped I/O, where specific addresses in the machine’s address space correspond to I/O devices; accesses to those addresses are interpreted as commands to the I/O devices.

SW instructions with an effective address of 0x0000F000 access the output device which displays an ASCII character in the terminal window; SW instructions with an effective address of 0x0000F001 access the output device which displays a 4-byte signed integer in the terminal window. Use a "Stderr" component for the first output device, and a "IntOut" component for the second output device.

The Memory Unit detects data movement instructions with invalid effective addresses, displays an error message in the terminal window, and aborts execution. Use a series of "Stderr" components to produce the error messages.

LW and SW instructions which access an invalid word address in partition 0x0000E000 produce the message "Error1".

SW instructions which access an invalid output device in partition 0x0000F000 produce the message "Error2".

LW and SW instructions which access an invalid partition produce the message "Error3".