

CSCI-2500 Assignment #6: Multicycle Datapath and Control Design

Christopher D. Carothers
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th Street
Troy, New York U.S.A. 12180-3590

November 10, 2009

DUE DATE: Noon, Tuesday, November 17th, 2009

1 Assignment Description

By the end of today's lecture, you will be familiar with the Multicycle Datapath and Control Design for a very small subset of the MIPS instruction set architecture, as indicated in Figures 5.28 (datapath and control) and 5.38 (finite state machine) of the course textbook. **Your task is to modify this design, BOTH THE DATAPATH/CONTROL FIGURE AND FINITE STATE MACHINE, to support the following additional five instructions. Each is worth 20 points.**

1. **swap**: This instruction take two registers as arguments (leave the 3rd register in the instruction unused) and swaps their contents. So for example, `swap $s0, $s1` will swap the value of `$s0` and `$s1` respectively.
2. **exchg**: This instruction take two registers, r_1 and r_2 and a 16 bit signed offset as input. It then uses r_2 as the base address of a memory location, adds the offset contained within the instruction and computes the address, and then "swaps" the value stored at that memory address with the value stored in r_1 . So, for example `exchg $s0, $s1, 64`, take $Mem[$s1 + 64$] \rightarrow $s0$ and $\$s0 \rightarrow Mem[$s1 + 64$]$. Here you need to create/use temporary registers within the datapath to store one of the values prior to overwriting it.

3. **jal**: This is the standard jump-and-link instruction supported by the MIPS instruction set architecture.
4. **lea**: This is a “load effective address” instruction that is very similar to the version contained in the Intel instruction architecture. For our MIPS version of this instruction, we re-purpose the `shamt` field of the R-type instruction to allow the “index” register to be scaled by any value in the range of 0 to 32. However, because we are limited to a 32-bit instruction format, we do not have room to support any offset. So, the form of this MIPS version is: $r_d = r_s + (s * r_t)$ where s is the “scale”, r_t is the index register and r_s is the base address register. The resulting address is stored in r_d . An example of this instruction is: `lea $s0, $s1, $s2, 7`. This results in $\$s0 = \$s1 + 7 * \$s2$.
5. **addm/subm**: This is a new instruction, called, “Add/Sub to memory”. The format is `addm r_d, r_s, r_t`. Here, destination register, r_d is a pointer to memory (absolute address without any offset) where the result will be stored and the other two registers, r_s and r_t , are source registers whose values are either added to subtracted based on the instruction name.

2 What to Modify

So, given the above instructions, you are to modify Figure 5.28 using the drawing software your choice (e.g., Dia, Xfig, Powerpoint, etc.). You must consider what new datapath elements do each of the instructions require on a cycle by cycle basis and if the necessary storage or connective paths between elements do not exist. If not, then you need to add them. Additionally, you need to modify the Finite State Machine, Figure 5.38, to show which Control Signals/Muxes are enabled for each clock cycle during the processing of each new instruction.

Please note, the design constraints are as follows: your Design can only do a memory read or write, 2 register reads and one write in a single clock cycle, additionally you can only do 1 ALU operation in a single clock cycle. Thus, if you need more than 1 ALU or Memory/Register read/write they must be done in different clock cycles. Last, you are not allowed to add more ALUs to your design.

3 HAND-IN INSTRUCTIONS

Print out a hard copy of your design, revised state machine and summary write-up, and hand it in to Dr. Carothers directly, slip under his office door or place in his Lally mailbox. . **It is expected**

that your designs and project write-up will have a high QUALITY about them. Hand written projects will not be accepted as they are impossible to read.