Programming Assignment 4 - Handling Virtual Memory
Design Document

Group 55 - Maxwell Bruce, Samuel Woolledge, Jiapei Kuang

General idea:

Adding virtual memory translation with a TLB to the pipelined RISC-V CPU.

TLB- overview:

All the implementations for TLB are in the TLB.* files with the I-TLB and D-TLB implemented. And there are two functions for TLB, update_entry and get_address.

TLB - structure:

In TLB.h, there are three defined struct for TLB. Each TLB entry contains valid bit, physical page, and virtual page. TLB is restricted to have only 8 entries.

TLB - get_address:

In this function, by bit masking, we get index for the virtual address.

First check if the status(described below) is 0xFF, if so by checking the index with virtual address and valid bit, if there is an address exist, then we output the physical address (32 bit combine by: high 18 bit of physical page and low 14 bit of virtual page). If not, we update the entry by calling update_entry.

If status is not 1, we call update_entry as well.

TLB - update_tlb_entry and handling stalls:

When update_entry is called, it will assign physical address and virtual address since caches will be indexed by both. And we only need to translate higher 18 bits from virtual to physical.

The function will split the virtual address into upper and lower to check and update. Then assign the first level index by shift right 8 bits and shift left 2 bits and add the value of ptbr (page table base register).

Check the status (status is used in pipeline for checking stalls, in fetch stage and in memory stage, it will get the status to see if there will be a stall or not. If there is a miss, the pipeline will stall.) and memory_status and memory_read for first index and second level index are successful.

For the second level index, leave blank for last 12 bits and leave low-order 20 bits by shifting bits. Then add virtual page (without the low order 2 bits which should be matching the physical page).

Check the status and memory_status and memory_read for second index and physical page are successful or not. And Check physical page for invalid entry.

Set entries with virtual page, physical page (32 bit combined by: high 18 bit of physical page and low 14 bit of virtual page), and the valid bit to 1.

Make sure that all bits above bit 31 should be masked off for virtual address and physical address. And lower 2 bit of virtual page number and physical page number are matching.