Computer Organization Project

Andrew Lockwood, Austin Alberts

5/5/2017

Introduction

This report encompasses the work of Andrew Lockwood and Austin Alberts on the project of ECEN 4593. This project was designed to expand the working knowledge of the effects of caches and a pipelined CPU. It enhances the coursework by allowing the students to find the optimal point between directly mapped caches and fully associative caches.

Running the Simulation

The code was developed on a Linux platform using C++. We thought C++ would be desired for the inherent ability to have underlying functions, parameters, classes and we both are going into object oriented programming after graduation, so we figured it was a good review. Our simulation is set up for the ability to configure caches on/off, the size of the caches, and which program is desired to run. In order to compile and execute our program, we created a makefile to assist. The makefile will look for all header files in the inc directory and all source code in the src directory. Running *$ make* will create an executable called output that will run our code. When configuration changes are desired caches can be toggled by the variables iCACHEON and dCACHEON defined on lines **<>** in main.cpp. Set it to true to turn the caches on or set it to false to turn them off. In order to switch from program 2 to program 1 the first function in int main on line **<>** change the function from *transfer\_Program2()* to *transfer\_Program1()*. The i and d caches are instances of the same cache class. Their cache size and words per block can be updated through the iCACHE\_SIZE, dCACHE\_SIZE and WORDS variables defined at lines **<>** in main.cpp. The project is complete except implementing dcaches correctly. We include branch forwarding, branches in detection stage, data forwarding, simulation of stalls, and instruction cache. The code runs all combinations of the icache on program 1 and program 2. The dcache runs correctly for WORDS = 1 and write through for program 1.

Self Assessment

Pipeline without caches

Pipeline with instruction cache without early start

Pipeline with instruction cache with early start

Pipeline with data cache and write through

Pipeline with data cache and write back

Pipeline with instruction and data cache with write through

Pipeline with instruction and data cache with write back

Test Results

Program 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| icache size | dcache size | Block Size | Write Policy | i-hit rate | d-hit rate | CPI | Clock cycles |
| None | None | N/A | N/A | N/A | N/A |  |  |
| 128 | 256 | 16 | WT |  |  |  |  |
| 128 | 256 | 16 | WB |  |  |  |  |
| 128 | 256 | 4 | WT |  |  |  |  |
| 128 | 256 | 4 | WB |  |  |  |  |
| 128 | 256 | 1 | WT |  |  |  |  |
| 128 | 256 | 1 | WB |  |  |  |  |
| 64 | 1024 | 16 | WT |  |  |  |  |
| 64 | 1024 | 16 | WB |  |  |  |  |
| 64 | 1024 | 4 | WT |  |  |  |  |
| 64 | 1024 | 4 | WB |  |  |  |  |
| 64 | 1024 | 1 | WT |  |  |  |  |
| 64 | 1024 | 1 | WB |  |  |  |  |

Program 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| icache size | dcache size | Block Size | Write Policy | i-hit rate | d-hit rate | CPI | Clock cycles |
| None | None | N/A | N/A | N/A | N/A |  |  |
| 64 | 512 | 16 | WT |  |  |  |  |
| 64 | 512 | 16 | WB |  |  |  |  |
| 64 | 512 | 4 | WT |  |  |  |  |
| 64 | 512 | 4 | WB |  |  |  |  |
| 64 | 512 | 1 | WT |  |  |  |  |
| 64 | 512 | 1 | WB |  |  |  |  |
| 256 | 128 | 16 | WT |  |  |  |  |
| 256 | 128 | 16 | WB |  |  |  |  |
| 256 | 128 | 4 | WT |  |  |  |  |
| 256 | 128 | 4 | WB |  |  |  |  |
| 256 | 128 | 1 | WT |  |  |  |  |
| 256 | 128 | 1 | WB |  |  |  |  |