CS161 Final Project
Vulnerability Crawler

Team Number
Terence Kinwah Tam, Cheng-yu Hong, Peter Cheng,
Jonathan Zernik, Gideon Chia, Patrick Bernal

May 4, 2012

1 Frama C is fun

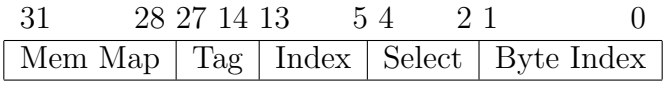
Our project implemented a three pipelined-stages MIPS on Xilinx Vertex 5 FPGA. With only three stages instead of the normal five stages, we have to reconsider the best placements of the various blocks into the pipeline divisions. For optimal performance, we want each stage to have around the same propagation delay, since

$$Clock\ Period = max(T_{stage1}, T_{stage2}, T_{stage3}) \tag{1}$$

2 Detailed Description of Sub-pieces

2.1 Memory Cache

Our cache design follows the recommended scheme on the specifications: a direct-mapped design with write-back and write-allocate. The data and tag portions of the cache are split into two separate block RAMs. Both block RAMs have a 512-bit read/write depth (9-bit index), but the data cache uses a 256-bit block size for optimal access to the DDR2 while the tag cache uses a 16-bit block size (14-bit tag with 1 valid and 1 dirty bit). The 32-bit address is divided as follows: the top 4 bits map to a memory space (BIOS ROM, instruction and data caches, or IO devices), the next 14 bits correspond to the tag, the next 9 bits correspond to the index, the next 3 bits select a 32-bit word in a 256-bit cache block, and the last 2 bits select a byte in a word.



3 Conclusions

From this project, we learned a lot about how the software programs are actually mapped onto hardware, providing us a very concrete understanding of the high level to low level organization of modern computer architecture.

Debugging in this project is the most tedious and challenging part of the process. It does not take us very long to wrap our heads around designing the code, but we would spend tenfold the time in getting the signals hooked up correctly. Choosing between the right part of a signal bus and muxing in a signal or its delayed version are some of the most common bugs. Unlike sequential computer programming that we are so familiar with, coding in Verilog and impacting on FPGA is not just a simple click of the compile button.