CSC 2/451 - Project 1

Yufei Du
<ydu14@ur.rochester.edu>

Princeton Ferro
<pferro@u.rochester.edu>

October 30, 2018

1 Files

All the files we modified, except for `sim_main.cpp`, are in `verilog/`.

2 iCache

Implemented in `verilog/ICACHE.v` as a 32 KB read-only direct-mapped cache with 32 B block size. The current PC is used to index into the cache. $32KB/32B$ is 1024, so the index is 10 bits. Each instruction is 4-byte aligned, so the lower 2 bits can be ignored, while the 3 bits after that will be the number of instructions into the cache entry. Therefore, the lower 5 bits are the offset. This leaves us with $32 - 10 - 5 = 17$ bits for the tag.

The cache table is implemented through an array with size 1024, and each block has the size of 274, including the valid bit with size 1, tag with size 17, and data with size 256.

The 10-cycle miss penalty is achieved by an internal counter that, when nonzero and less than 10, stalls the IF and ID stages. IF and ID don't proceed if the `valid` from the iCache is negative.

Additionally, `sim_main.cpp` was modified at lines 823 and 827 to fetch on aligned addresses.

3 dCache

Implemented in `verilog/DCACHE.v`, but not integrated with the rest of the MIPS pipeline because we could not fully integrate it without causing problems. The dCache was intended to be placed after the MEM stage.

Although the dCache module is not integrated with MIPS pipeline, the module itself is complete. The cache table is implemented through two arrays, each representing a set, with size 512 each. Each block has the size of 276, including the dirty bit with size 1, valid bit with size 1, recently used bit with size 1, tag with size 17, and data with size 256. The recently used

bit is used when choosing the victom for cache replacement. The dCache uses a Least Recently Used (LRU) cache replacement policy, so when the cache needs to replace a cache block, it will replace the one that is not recently used from the two sets.

4 Programs

program	w/o cache		with i-cache	
	cycles	IPC	cycles	IPC
noio	27700	0.0751264	4272	0.487125
file	1307030	0.0728484	107322	0.88719
hello	1308960	0.0731153	106360	0.899821
class	1328990	0.0731209	108262	0.89761
sort	1436460	0.0730435	117767	0.890946
fact12	1515830	0.0731085	123133	0.900002
matrix	1881850	0.0729527	153471	0.89454
hanoi	2762500	0.0730016	228402	0.882948
ical	2519410	0.0859245	227704	0.950704
fib18	4207410	0.0726692	316934	0.964709

The baseline (two left columns) is the "uncached" data, when running with the original MIPS pipeline given to us without any modifications.