Architecture Simulation

Submission.

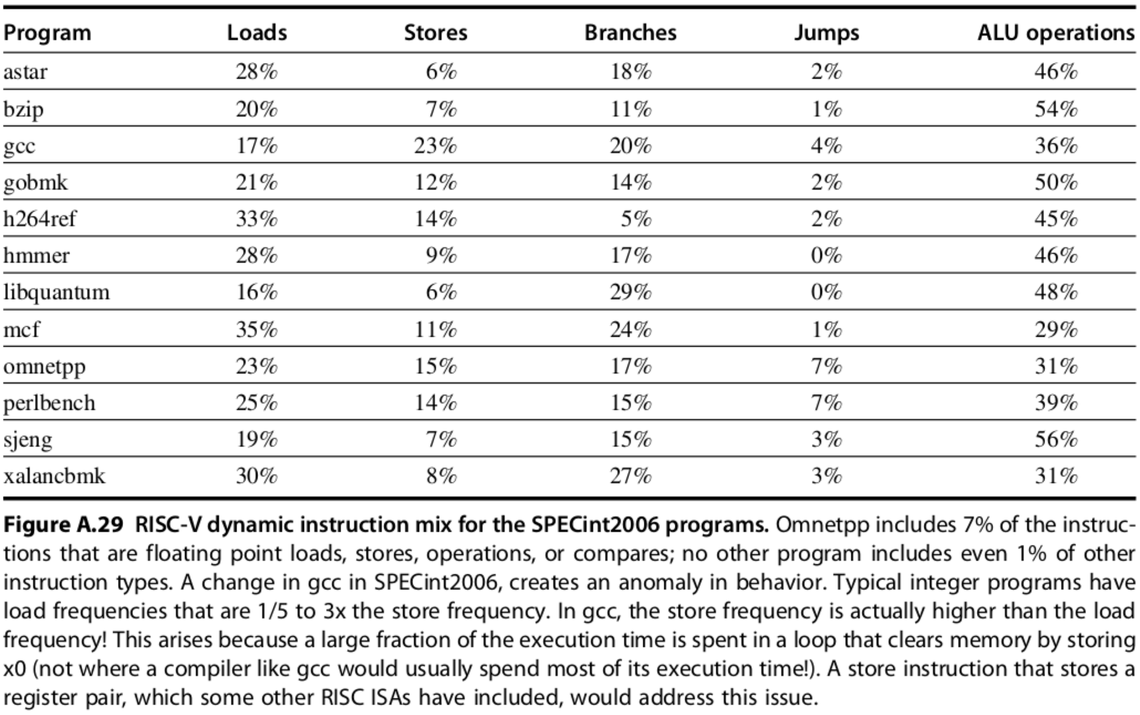
1. Submit a report. Report should contain:
   1. Experimental Setup (what configs, inputs, machine/processor/RAM size/Cache size etc), Inputs, Outputs
   2. Inferences, observations from results

Tools for Q1 and Q2 . Download, Install, and learn to run ChampSim (https://github.com/ChampSim/ChampSim)
Use the traces available on the ChampSim site (DPC-3 traces from the README). Alternatively, use Pin to create more SPEC2006 traces. If you are creating your own traces, download the SPEC CPU 2006 benchmark suite from here here: https://drive.google.com/file/d/1lZHf9S5VmHJ3jqp4uOKScDbcHetdAqg_/view?usp=sharing
[[the ISO file is 3GB]]. SPEC Suit installation manual: https://www.spec.org/cpu2006/Docs/install-guide-unix.html#s5a.

Assignment Questions

Q1.

Classify instructions in the traces as Loads/Stores/Branches/Jumps/ALU ops (Fig. A.29, Page A-42, HP-6e). Fig. A.29 is reproduced below. Use any 10 (5 INT + 5 FP) SPEC benchmarks. List the 10 benchmarks you are using – write max 2 lines the programs you have chosen for this Q.



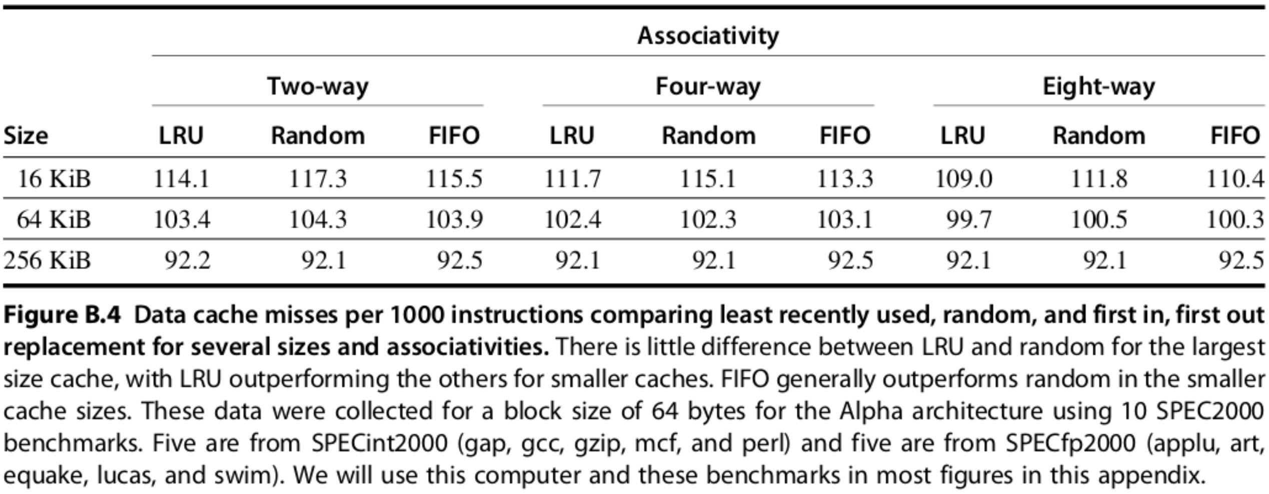
Program	Loads	Stores	Branches	Jumps	ALU operations
astar	28%	6%	18%	2%	46%
bzip	20%	7%	11%	1%	54%
gcc	17%	23%	20%	4%	36%
gobmk	21%	12%	14%	2%	50%
h264ref	33%	14%	5%	2%	45%
hmmer	28%	9%	17%	0%	46%
libquantum	16%	6%	29%	0%	48%
mcf	35%	11%	24%	1%	29%
omnetpp	23%	15%	17%	7%	31%
perlbench	25%	14%	15%	7%	39%
sjeng	19%	7%	15%	3%	56%
xalancbmk	30%	8%	27%	3%	31%

Figure A.29 RISC-V dynamic instruction mix for the SPECint2006 programs. Omnetpp includes 7% of the instructions that are floating point loads, stores, operations, or compares; no other program includes even 1% of other instruction types. A change in gcc in SPECint2006, creates an anomaly in behavior. Typical integer programs have load frequencies that are 1/5 to 3x the store frequency. In gcc, the store frequency is actually higher than the load frequency! This arises because a large fraction of the execution time is spent in a loop that clears memory by storing x0 (not where a compiler like gcc would usually spend most of its execution time!). A store instruction that stores a register pair, which some other RISC ISAs have included, would address this issue.

Q2.

Compare 5 cache replacement policies (LRU, Random, FIFO, and two others from ChampSim of your choice). Reproduce results in the table in Figure B.4 (Page B-10) from the HP-6ed textbook (shown below).

Use any 6 SPEC2006 benchmarks (3INT + 3FP). Champsim provides traces for you (DPC-3 traces from the README). Alternatively, build your own traces using Pin (mini-procedure is in the Champsim README).



	Associativity								
	Two-way			Four-way			Eight-way		
Size	LRU	Random	FIFO	LRU	Random	FIFO	LRU	Random	FIFO
16 KiB	114.1	117.3	115.5	111.7	115.1	113.3	109.0	111.8	110.4
64 KiB	103.4	104.3	103.9	102.4	102.3	103.1	99.7	100.5	100.3
256 KiB	92.2	92.1	92.5	92.1	92.1	92.5	92.1	92.1	92.5

Figure B.4 Data cache misses per 1000 instructions comparing least recently used, random, and first in, first out replacement for several sizes and associativities. There is little difference between LRU and random for the largest size cache, with LRU outperforming the others for smaller caches. FIFO generally outperforms random in the smaller cache sizes. These data were collected for a block size of 64 bytes for the Alpha architecture using 10 SPEC2000 benchmarks. Five are from SPECint2000 (gap, gcc, gzip, mcf, and perl) and five are from SPECfp2000 (applu, art, equake, lucas, and swim). We will use this computer and these benchmarks in most figures in this appendix.

Q3.

CACTI: Download and install Cacti from http://www.cs.utah.edu/~rajeev/cacti6/ . Answer the following questions for a 32nm, 4 bank, 2-way Set Associative, 64B line size, 16KB cache:

(a) How many total sets per bank are there?

(b) What is the default Vdd value?

(c) What are the components of the access time parameter in Cacti? What was the value in your case?

(d) Amongst the Data array and the Tag array, which consumed the most dynamic and leakage power? What are the values?

(e) How large (in mm^2) is the 16KB cache?

(f) Draw the architecture of a 4 bank, 2-way Set Associative, 64B line size, 16KB cache.

Q4.

Offer constructive suggestions about this assignment after you complete all the previous questions. Answer this question based on your experience:

1. Provide a description of any difficulties or misunderstandings that made the assignment unnecessarily difficult, or that led you to waste time.
2. Suggestions of changes that could make this assignment more interesting, more relevant or more challenging in future editions of this course.

****