ANALYSIS OF PROCESSOR PERFORMANCE USING SimpleScalar

VELIVELA BHANU PRAKASH

Abstract— Analyzing the effects of processor performance in relation to changes in cache size, Block size and associativity according to the given data. By monitoring the performance, the SimpleScalar simulator may determine changes in cache size, block size, and associativity size using a variety of SPEC2000 benchmarks. The simulators that range from a quick functional simulator to a comprehensive out-of-order issue processor that supports caches and speculative execution. The outcomes demonstrate the performance of several benchmark settings. The compiler, assembler, linker, simulation, and visualization tools in this tool set are for the SimpleScalar architecture.

I. INTRODUCTION

The processor simulators and auxiliary tools that make up SimpleScalar are a group of instructions. Simplescalar to assess the performance effects of a variety of cache setups. The simulator for the CPU is called Simplescalar. Running on Unix/Linux, the simulator is a piece of C-language software. At each cycle, the simulator receives lot of information to duplicate a super scalar machine. Several benchmarks in SPEC2000's default configuration were run because the instructions were changed. The newest generation of CPU-intensive benchmarks, known as SPEC2000, are industry standards. SPEC2000 is used to compare compute-intensive performance across the broadest range of hardware that is currently feasible. Source code benchmarks created from actual user apps were the outcome of the implementation. These benchmarks evaluate the effectiveness of the system under test's CPU, memory, and compiler. Data cache memory, instruction cache memory, data cache associativity, and unified data cache are calculated using the 10 benchmarks gcc, bzip, crafty, eon, swim, mcf, galgel, wupwise, equake and mesa. The default setup for many tasks is the basic configuration. Simulators for both functionality and performance are included in SimpleScalar. Simulators help programmers and compiler writers to test their microcontroller.

II. PROCESS OF EXECUTION/ COMMANDS:

This Simplescalar machine is simulator which runs in Unix/Linux. We have Utilized the UH VPN to connect to UH server. First and foremost, we need to unzip the given files using the command prompt.

For unzipping the files, tar -zxvf simplesim-3voe.tgz command is used.

For unzipping the files, tar -zxvf SPEC2000.tar.gz command is used.

To copy the configuration to the benchmark suite - cp config/default.cfg ../SPEC2000/gcc

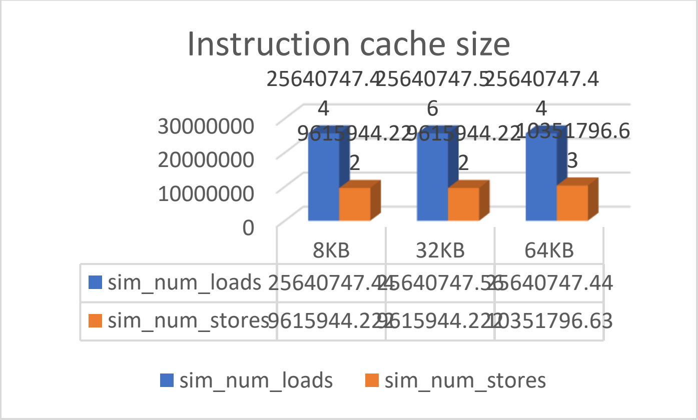
To open a folder – cd folder name

To come out of a folder – cd ..

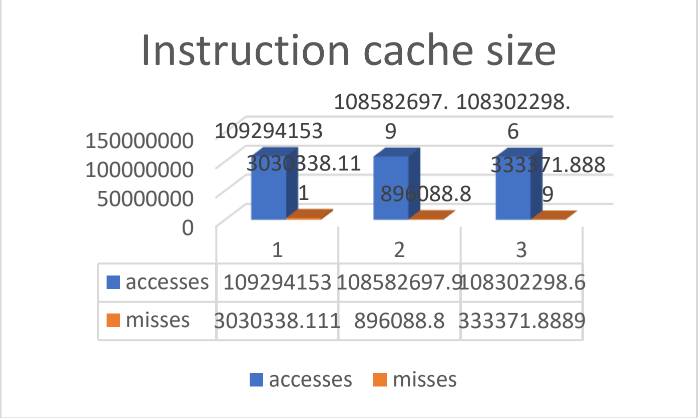
To perform 1000000 instructions in the simulator./sim-outorder -config default.cfg -max:inst 1000000 -fastfwd 1000000

III. OUTCOMES & RESULTS:

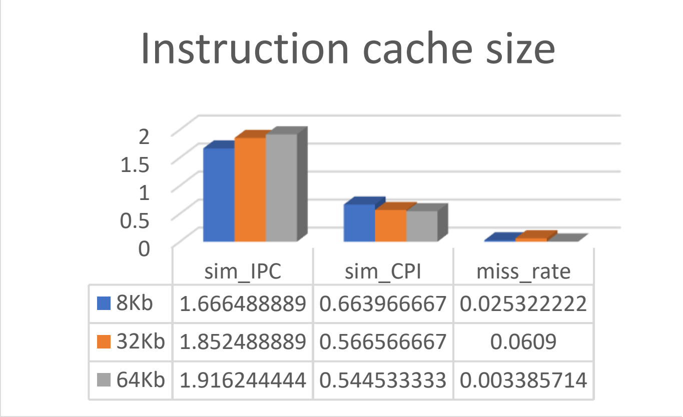
a) Instruction cache size


Instruction cache size

	8KB	32KB	64KB
sim_num_loads	25640747.44	25640747.56	25640747.44
sim_num_stores	9615944.222	9615944.222	10351796.63


Instruction cache size

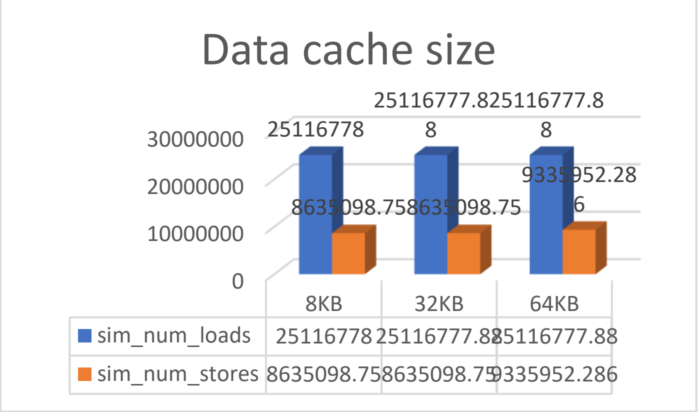
	1	2	3
accesses	109294153	108582697.9	108302298.6
misses	3030338.111	896088.8	333371.8889


Instruction cache size

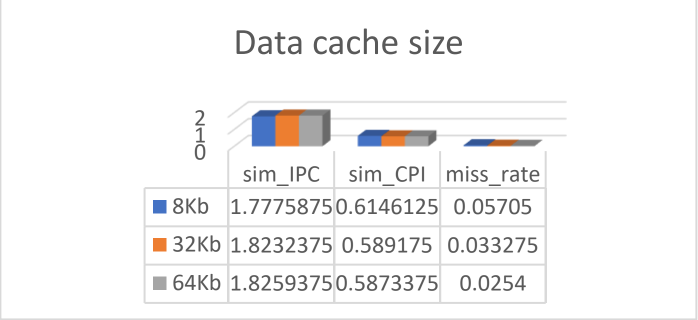
	sim_IPC	sim_CPI	miss_rate
8Kb	1.666488889	0.663966667	0.025322222
32Kb	1.852488889	0.566566667	0.0609
64Kb	1.916244444	0.544533333	0.003385714

First, we need to change adjust the cache size as pre required values. And then by using different benchmarks and configurations, different kinds of performance have been observed. The miss rate has been too small, and it has been observed to be fluctuating. The IPC & CPI are observed to be increasing by increasing in the kb size.

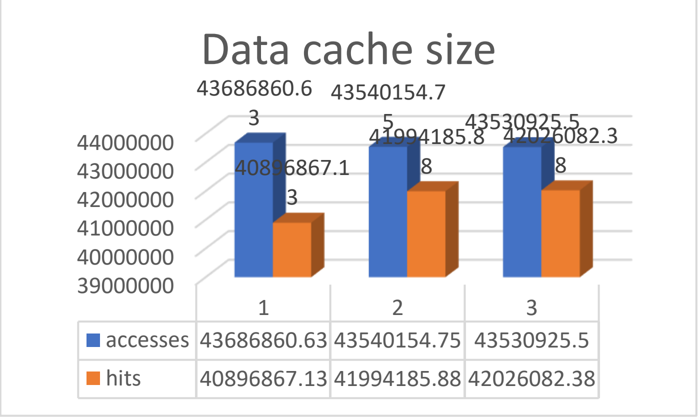
b) DATA CACHE SIZE


Data cache size

	8KB	32KB	64KB
sim_num_loads	25116778	25116777.88	25116777.88
sim_num_stores	8635098.75	8635098.75	9335952.286


Data cache size

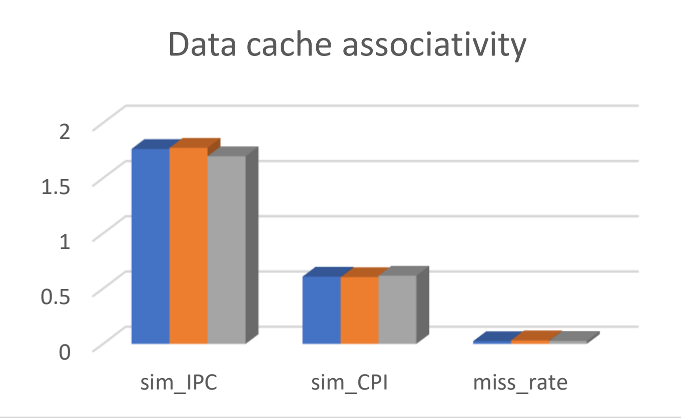
	sim_IPC	sim_CPI	miss_rate
8Kb	1.7775875	0.6146125	0.05705
32Kb	1.8232375	0.589175	0.033275
64Kb	1.8259375	0.5873375	0.0254

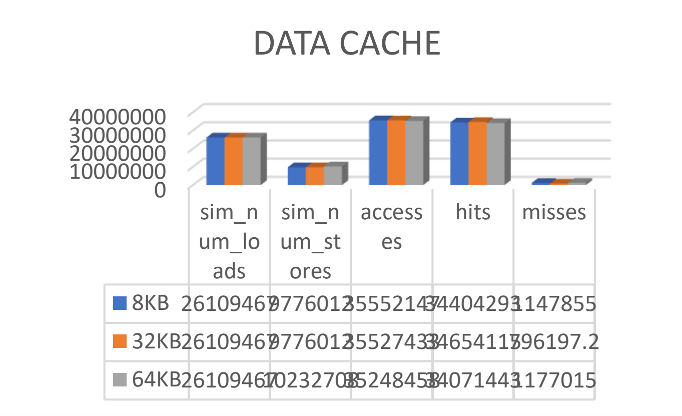

Data cache size

	1	2	3
accesses	43686860.63	43540154.75	43530925.5
hits	40896867.13	41994185.88	42026082.38

Secondly, we need to change the number of set-in data cache and equate it to data cache size. And then by using different benchmarks and configurations, different kinds of performance have been observed. The cache is slightly showing logical segmentation.

c) Data cache associativity





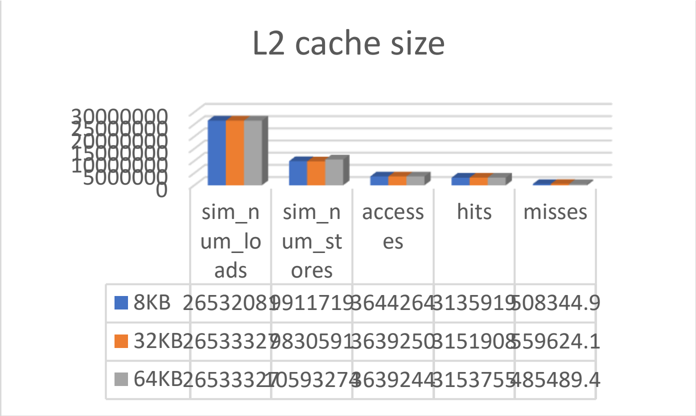
Thirdly, we use Associativity to change and use the benchmarks to get the outcomes. It has been observed that the loads data is same for all the Killo-bytes.

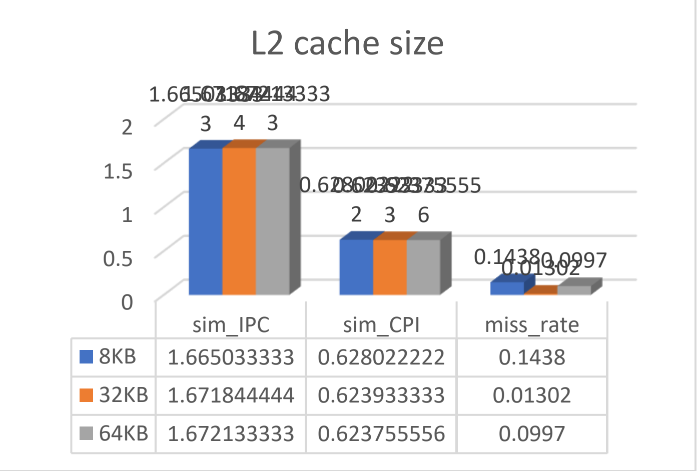
The store values are being changing upto some extent.

Access time and hits are being constantly same.

IPC AND CPI values are as shown in the above observations.

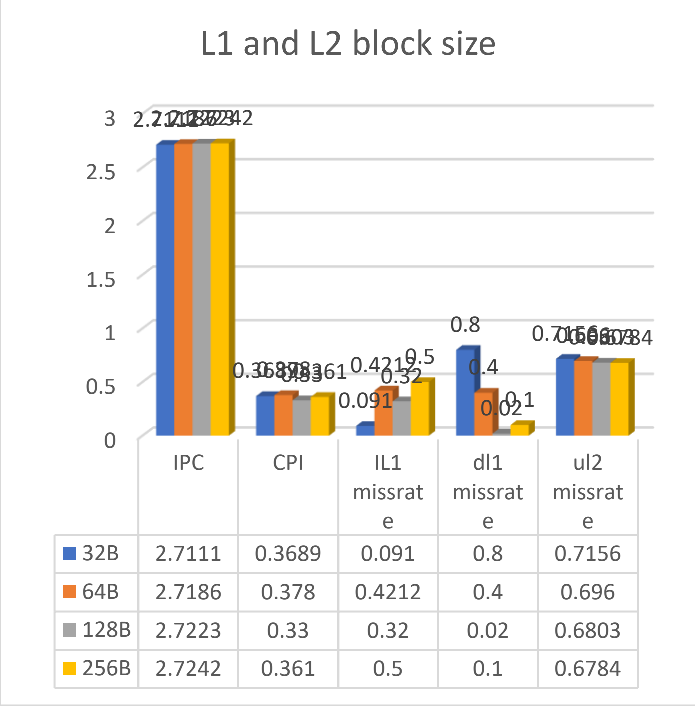
d) L2 cache size



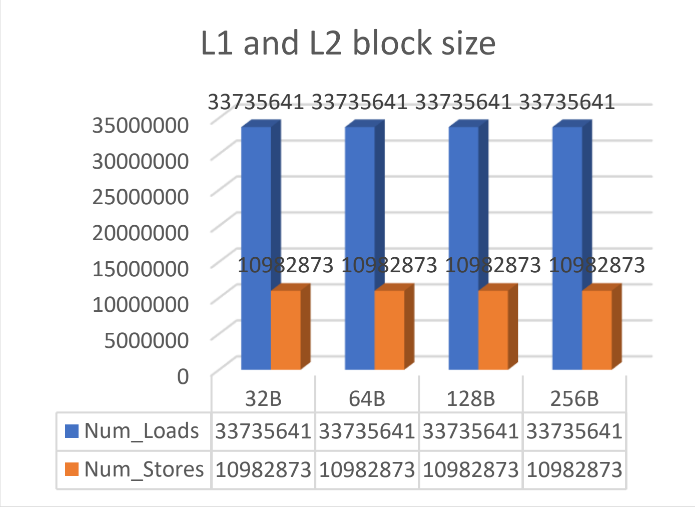


Furthermore, in the l2 data cache we change the number of sets in the data cache keeping the cache size same. This is again performed on different benchmarks and values are recorded.

e) L1 and L2 block size



	IPC	CPI	IL1 missrate	dl1 missrate	ul2 missrate
🟦 32B	2.7111	0.3689	0.091	0.8	0.7156
🟧 64B	2.7186	0.378	0.4212	0.4	0.696
▨ 128B	2.7223	0.33	0.32	0.02	0.6803
🟨 256B	2.7242	0.361	0.5	0.1	0.6784



	32B	64B	128B	256B
🟦 Num_Loads	33735641	33735641	33735641	33735641
🟧 Num_Stores	10982873	10982873	10982873	10982873

Depending on the CPU, the L2 cache size changes. Nevertheless, it often ranges from 256KB to 8MB. Most of the larger L2 cache than 256KB is available in modern CPUs. Now regarded as small. Additionally, some of the most powerful CPUs have a larger L2 memory cache. 8MB. A memory cache that is physically located in the L1 cache utilized to store data and is built into the CPU. latest data obtained by the CPU. on the L2 cache on the other hand, is an external cache memory. Despite being switched on, apart than the CPU chip core same CPU chip packaging.

IV. Conclusion:

Different values are being observed while using the benchmarks and different configurations. Several benchmarks in SPEC2000's default configuration were run because the instructions were changed. The SimpleScalar simulator may determine changes in cache size, block size, and associativity size using a variety of SPEC2000 benchmarks.