Computer Organization Project #3

TA in charge: Jinkwon Kim
E-mail: coco@kaist.ac.kr
Office Hour: 16:00 ~ 17:30, Mon. / Wed. (N1, 402)

I. Goal of this project

- ✓ To understand the effect of cache configuration to performance.

II. Work as a Team

- ✓ Project #3 is a team project. You can keep the team of Project #2.
- ✓ If you want to change your team, please let the TAs know the change.

III. Submission and grading

- ✓ This is a reminder to you what to submit.
- ✓ Please submit only one report by each team.
- ✓ For more detailed, please refer to following instructions.
- ✓ You must submit all of followings **(Total 100 points)**:

A. 16 simulation result files **(40 points)** :

Please follow the file name convention.

8 files for cache associativity number

Here is the format: <benchmark_name>_<Nsets>_<bsize>_<way>_assos

(math_32_16_1_assoc, math_16_16_2_assoc, math_8_16_4_assoc, math_4_16_8_assoc, dirent_32_16_1_assoc, dirent_16_16_2_assoc, dirent_8_16_4_assoc, dirent_4_16_8_assoc)

8 files for cache block size.

Here is the format: <benchmark_name>_<Nsets>_<bsize>_<way>_blk

(math_32_8_2_blk, math_16_16_2_blk, math_8_32_2_blk, math_4_64_2_blk, dirent_32_8_2_blk, dirent _16_16_2_blk, dirent _8_32_2_blk, dirent _4_64_2_blk)

B. Analysis report about simulation results **(60 points)**:

The report file name must be

<Project3_StudentID1_Name1_StudentID2_Name2_StudentID3_name3.pdf>.
(e.g. *Project3_20160000_A_20150000_B_20140000_C.pdf*)

You can use either English or Korean as you want.

- ✓ *Compress all these files into a .zip file* and upload the compressed file on KLMS.
- ✓ The submitted file name must be

< Project3_StudentID1_Name1_StudentID2_Name2_StudentID3_name3.zip>.
(e.g. *Project3_20160000_A_20150000_B_20140000_C.zip*)

IV. Due date

- ✓ Nov. 30th (Fri.), 23:59:00
- ✓ Late submission due date: Dec. 1st (Sat.), 23:59:00 (50% point deduction)
- ✓ After the late submission due date, you will get 0 point.

V. Cheating

- ✓ If there are any cheatings in your submission, you will get 0 point.
- ✓ *Followings will be regarded as cheating*:

A. Copying other students' simulation results or reports

B. Modifying other students' results and using them as if they were your own

C. Using other sources without any references excluding your own simulation results

D. All other sorts of inappropriate behaviors

VI. Prerequisite

- ✓ We will use the same virtual machine image from Project #2. (no more setting)
- ✓ *I hope that no one still suffer from the setting.*
- ✓ We use the simplescalar-Alpha (version3.0 same as Project #2).

VII. Tasks

In this project, we run 2 benchmarks with various cache configuration to analyze the effect of that. The L1 data cache configuration will be modified while the L1 Instruction cache configuration is fixed and other caches (like L2) are disabled.

1. Change the configuration file to modify cache configuration.

We use default.cfg file, which is a simplescalar default configuration file.

To examine the file, execute this command:

vi ~/simplesim-3.0/config/default.cfg

~/simplesim-3.0 should be replaced by the your simplescalar directory

As you can see, there are bunch of configuration for cpu.

The only thing we have to focus on is cache configuration.

The cache configuration parameter has the following format:

<name>:<nsets>:<bsize>:<assoc>:<repl>

<name> - name of the cache being defined

<nsets> - number of sets in the cache

<bsize> - block size of the cache

<assoc> - associativity of the cache

<repl> - block replacement strategy, 'l'-LRU, 'f'-FIFO, 'r'-random

Examples: -cache:dl1 dl1:4096:32:1:l

We have to change some cache configuration in default.cfg like this.

	Before modification	After modification
L1 instruction	il1:512:32:1:l	il1:8:16:4:l
L1 data	dl1:128:32:4:l	dl1:8:16:4:l
L2 instruction	ul2:1024:64:4:l	None
L2 data	dl2	None

To reflect this change, just modify your default.cfg file as follow.

```
# l1 data cache config, i.e., {<config>|none}
-cache:dl1              dl1:128:32:4:l

# l1 data cache hit latency (in cycles)
-cache:dl1lat                    1

# l2 data cache config, i.e., {<config>|none}
-cache:dl2              ul2:1024:64:4:l

# l2 data cache hit latency (in cycles)
-cache:dl2lat                    6

# l1 inst cache config, i.e., {<config>|dl1|dl2|none}
-cache:il1              il1:512:32:1:l

# l1 instruction cache hit latency (in cycles)
-cache:il1lat                    1

# l2 instruction cache config, i.e., {<config>|dl2|none}
-cache:il2                     dl2

# l2 instruction cache hit latency (in cycles)
-cache:il2lat                    6

# flush caches on system calls
-cache:flush                 false

# convert 64-bit inst addresses to 32-bit inst equivalents
-cache:icompress             false
```

Figure 1. Before modification

```
# l1 data cache config, i.e., {<config>|none}
-cache:dl1              dl1:8:16:4:l

# l1 data cache hit latency (in cycles)
-cache:dl1lat                    1

# l2 data cache config, i.e., {<config>|none}
-cache:dl2              none

# l2 data cache hit latency (in cycles)
-cache:dl2lat                    6

# l1 inst cache config, i.e., {<config>|dl1|dl2|none}
-cache:il1              il1:8:16:4:l

# l1 instruction cache hit latency (in cycles)
-cache:il1lat                    1

# l2 instruction cache config, i.e., {<config>|dl2|none}
-cache:il2                   none

# l2 instruction cache hit latency (in cycles)
-cache:il2lat                    6

# flush caches on system calls
-cache:flush                 false

# convert 64-bit inst addresses to 32-bit inst equivalents
-cache:icompress             false
```

Figure 2. After modification

To verify your modification, let's run a benchmark and check the modification.

To collect the result file, I recommend you to make result directory.

By executing this command:

mkdir ~/simplesim-3.0/result

To run the simulator with benchmark, execute this command:

~/simplesim-3.0/sim-outorder -config ~/simplesim-3.0/config/default.cfg -redir:sim ~/simplesim-3.0/result/verify ~/simplesim-3.0/tests-alpha/bin/test-dirent

To check the result file, execute this command:

vi ~/simplesim-3.0/result/verify

To make sure that your verify file has the same result as follow:

```
sim-outorder: SimpleScalar/Alpha Tool Set version 3.0 of August, 2003.
Copyright (c) 1994-2003 by Todd M. Austin, Ph.D. and SimpleScalar, LLC.
All Rights Reserved. This version of SimpleScalar is licensed for academic
non-commercial use.  No portion of this work may be used by any commercial
entity, or for any commercial purpose, without the prior written permission
of SimpleScalar, LLC (info@simplescalar.com).

warning: section `.comment' ignored...
sim: command line: /home/cs311/simplesim-3.0/sim-outorder -config /home/cs311/simplesim-3.0/config/default.cfg -redir:s
est-dirent

sim: simulation started @ Thu Nov 15 10:31:32 2018, options follow:

sim-outorder: This simulator implements a very detailed out-of-order issue
superscalar processor with a two-level memory system and speculative
execution support.  This simulator is a performance simulator, tracking the
latency of all pipeline operations.

# -config                     # load configuration from a file
# -dumpconfig                 # dump configuration to a file
# -h                    false # print help message
# -v                    false # verbose operation
# -d                    false # enable debug message
# -i                    false # start in Dlite debugger
-seed                       1 # random number generator seed (0 for timer seed)
# -q                    false # initialize and terminate immediately
# -chkpt                <null> # restore EIO trace execution from <fname>
# -redir:sim     /home/cs311/simplesim-3.0/result/verify # redirect simulator output to file (non-interactive only)
# -redir:prog           <null> # redirect simulated program output to file
-nice                       0 # simulator scheduling priority
-max:inst                   0 # maximum number of inst's to execute
-fastfwd                    0 # number of insts skipped before timing starts
# -ptrace               <null> # generate pipetrace, i.e., <fname|stdout|stderr> <range>
-fetch:ifqsize              4 # instruction fetch queue size (in insts)
-fetch:mplat                3 # extra branch mis-prediction latency
-fetch:speed                1 # speed of front-end of machine relative to execution core
-bpred                  bimod # branch predictor type {nottaken|taken|perfect|bimod|2lev|comb}
-bpred:bimod     2048 # bimodal predictor config (<table size>)
-bpred:2lev      1 1024 8 # 2-level predictor config (<l1size> <l2size> <hist_size> <xor>)
-bpred:comb      1024 # combining predictor config (<meta_table_size>)
-bpred:ras                  8 # return address stack size (0 for no return stack)
-bpred:btb       512 4 # BTB config (<num_sets> <associativity>)
# -bpred:spec_update       <null> # speculative predictors update in {ID|WB} (default non-spec)
-decode:width               4 # instruction decode B/W (insts/cycle)
-issue:width                4 # instruction issue B/W (insts/cycle)
-issue:inorder          false # run pipeline with in-order issue
-issue:wrongpath         true # issue instructions down wrong execution paths
-commit:width               4 # instruction commit B/W (insts/cycle)
-ruu:size                 16 # register update unit (RUU) size
-lsq:size                  8 # load/store queue (LSQ) size
-cache:dl1       dl1:8:16:4:l # l1 data cache config, i.e., {<config>|none}
-cache:dl1lat               1 # l1 data cache hit latency (in cycles)
-cache:dl2              none # l2 data cache config, i.e., {<config>|none}
-cache:dl2lat               6 # l2 data cache hit latency (in cycles)
-cache:il1       il1:8:16:4:l # l1 inst cache config, i.e., {<config>|dl1|dl2|none}
-cache:il1lat               1 # l1 instruction cache hit latency (in cycles)
-cache:il2              none # l2 instruction cache config, i.e., {<config>|dl2|none}
"~/simplesim-3.0/result/verify" 247L, 15465C
```

We are ready to analyze the cache configuration.

2. Analyze the effect of cache associativity.

To analyze the effect of cache associativity, we conduct 4 simulations for each benchmark. Since the benchmark program is small, we fix the total L1 data cache size as 512 byte. (The total cache size is calculated as follow: nsets * bsize * assoc)

Here is the configuration, you have to do. (Note that you have to only change the L1 data cache configuration while the other configuration is fixed to the value in Step 1)

	L1 data	L1 instruction	L2 data	L2 instruction
Configuration 1	Nsets: 32 bsize: 16 way: 1	il1:8:16:4:l (Same as step1.)	None (Same as step1.)	None (Same as step1.)
Configuration 2	Nsets: 16 bsize: 16 way: 2	il1:8:16:4:l (Same as step1.)	None (Same as step1.)	None (Same as step1.)
Configuration 3	Nsets: 8 bsize: 16 way: 4	il1:8:16:4:l (Same as step1.)	None (Same as step1.)	None (Same as step1.)
Configuration 4	Nsets: 4 bsize: 16 way: 8	il1:8:16:4:l (Same as step1.)	None (Same as step1.)	None (Same as step1.)

You have to conduct 4 simulation for each benchmark. We use test-math and test-dirent (‘tests-alpha/bin/test-math’, ‘tests-alpha/bin/test-dirent’)

Each configuration, you have to save the result and set the result file name as follow:

For test-math with cache configuration (Nsets: 32 bsize: 16 way: 1)

math_32_16_1_assoc (which means that math_Nsets_bsize_way_assoc)

You can get this result file by executing this command (after change the L1 data configuration like Step 1):

~/simplesim-3.0/sim-outorder -config ~/simplesim-3.0/config/default.cfg -redir:sim ~/simplesim-3.0/result/math_32_16_1_assoc ~/simplesim-3.0/tests-alpha/bin/test-math

For test-dirent with cache configuration (Nsets: 32 bsize: 16 way: 1)

dirent_32_16_1_assoc (which means that dirent_Nsets_bsize_way_assoc)

You can get this result file by executing this command (after change the L1 data configuration like Step 1):

~/simplesim-3.0/sim-outorder -config ~/simplesim-3.0/config/default.cfg -redir:sim ~/simplesim-3.0/result/dirent_32_16_1_assoc ~/simplesim-3.0/tests-alpha/bin/test-dirent

[What to submit]

① You have to submit 8 files for this task. (Please follow the file name convention. Before executing a program, you have to change the configuration like step1.)

math_32_16_1_assoc, math_16_16_2_assoc, math_8_16_4_assoc, math_4_16_8_assoc,

dirent_32_16_1_assoc, dirent_16_16_2_assoc, dirent_8_16_4_assoc, dirent_4_16_8_assoc

② In report, answer these questions.

1. Find out the best configuration for each benchmark. (just compare the miss rate)
2. Why is this configuration the best?

3. Analyze the effect of block size.

To analyze the effect of cache block size, we conduct 4 simulations for each benchmark. Since the benchmark program is small, we fix the total L1 data cache size as 512 byte. (The total cache size is calculated as follow: nsets * bsize * assoc)

Here is the configuration, you have to do. (Note that you have to only change the L1 data cache configuration while the other configuration is fixed to the value in Step 1)

	L1 data	L1 instruction	L2 data	L2 instruction
Configuration 1	Nsets: 32 bsize: 8 way: 2	il1:8:16:4:l (Same as step1.)	None (Same as step1.)	None (Same as step1.)
Configuration 2	Nsets: 16 bsize: 16 way: 2	il1:8:16:4:l (Same as step1.)	None (Same as step1.)	None (Same as step1.)
Configuration 3	Nsets: 8 bsize: 32 way: 2	il1:8:16:4:l (Same as step1.)	None (Same as step1.)	None (Same as step1.)
Configuration 4	Nsets: 4 bsize: 64 way: 2	il1:8:16:4:l (Same as step1.)	None (Same as step1.)	None (Same as step1.)

You have to conduct 4 simulation for each benchmark. We use test-math and test-dirent (‘tests-alpha/bin/test-math’, ‘tests-alpha/bin/test-dirent’)

Each configuration, you have to save the result and set the result file name as follow:

For test-math with cache configuration (Nsets: 32 bsize: 8 way: 2)

math_32_8_2_blk (which means that math_Nsets_bsize_way_blk)

You can get this result file by executing this command (after change the L1 data configuration like Step 1)::

~/simplesim-3.0/sim-outorder -config ~/simplesim-3.0/config/default.cfg -redir:sim ~/simplesim-3.0/result/math_32_8_2_blk ~/simplesim-3.0/tests-alpha/bin/test-math

For test-dirent with cache configuration (Nsets: 32 bsize: 8 way: 2)

dirent_32_8_2_blk (which means that dirent_Nsets_bsize_way_blk)

You can get this result file by executing this command (after change the L1 data configuration like Step 1)::

~/simplesim-3.0/sim-outorder -config ~/simplesim-3.0/config/default.cfg -redir:sim ~/simplesim-3.0/result/dirent_32_8_2_blk ~/simplesim-3.0/tests-alpha/bin/test-dirent

[What to submit]

① You have to submit 8 files for this task. (Please follow the file name convention. Before executing a program, you have to change the configuration like step1.)

math_32_8_2_blk, math_16_16_2_blk, math_8_32_2_blk, math_4_64_2_blk,

dirent_32_8_2_blk, dirent _16_16_2_blk, dirent _8_32_2_blk, dirent _4_64_2_blk

② In report, answer these questions.

1. Find out the best configuration for each benchmark. (just compare the miss rate)
2. Why is this configuration the best?