SimpleScalar Tool Set

In order to perform fast, flexible, and accurate simulation of modern microprocessors, the SimpleScalar tool set can be used. The built architecture inside this tool set is a derivative of the MIPS architecture, which takes the architecture-specific compiled binaries for examination of seven different architectural aspects of the processor architecture (a.k.a. Super Seven). The examination is done through compiling an user developed application or a benchmark (e.g. from a SPEC's benchmark suite) by the architecture-specific GNU GCC compiler and running it on the processor.

The seven architectural aspects include branch prediction behavior, cache behavior, external I/O trace generator, fast functional simulator implementation, out of order execution, sample functional simulator implementation with profiling, sample functional simulator implementation, and provide high flexibility, portability, extensibility, and performance. The main software components inside the tool set are: `bpred`, `cache`, `config`, `dlite`, `eio`, `endian`, `eval`, `eventq`, loader, machine, main, memory, misc, options, `ptrace`, `range`, `regs`, `resource`, `sim-bpred`, `sim-cache`, `sim-eio`, `sim-fast`, `sim-outorder`, `sim-profile`, `sim-safe`, `symbol`, `syscall`, `sysprobe`.

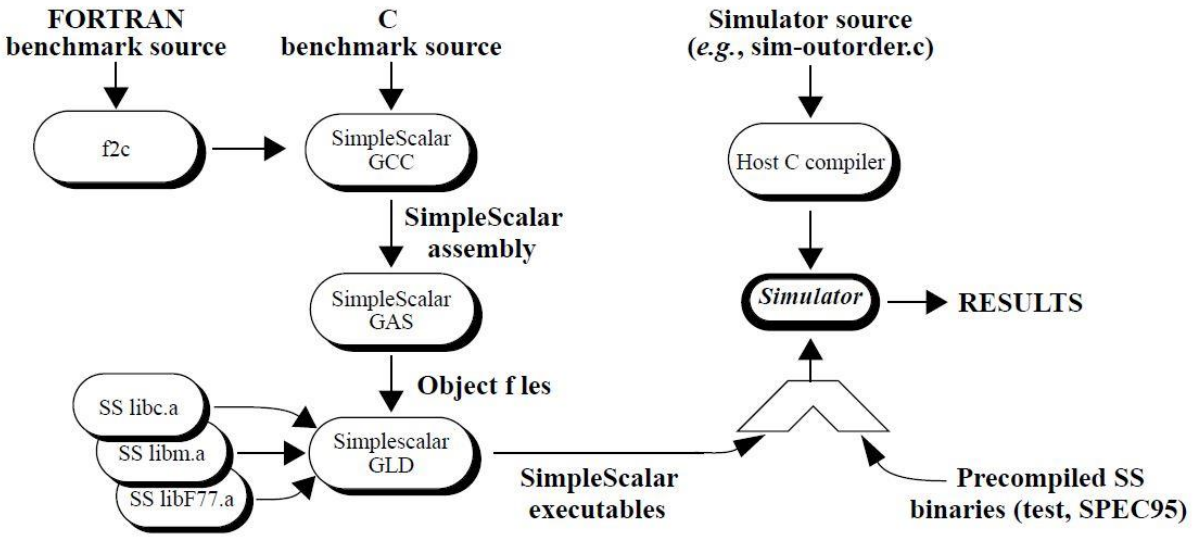

Figure 1. SimpleScalar tool set overview

The SimpleScalar tool set includes both little-endian and big-endian versions of the derivation of the MIPS-IV ISA for improvement of the portability. The differences between the semantics of the SimpleScalar ISA and the MIPS ISA can be described as: (1) having no architected delay slots: loads, stores, and control transfers don't execute the succeeding instruction; (2) loads and stores support two addressing modes that are indexed (Register + Register) and auto increment/decrement; (3) inclusion of a square-root instruction for floating point numbers; and (4) an extended 64-bit instruction encoding.

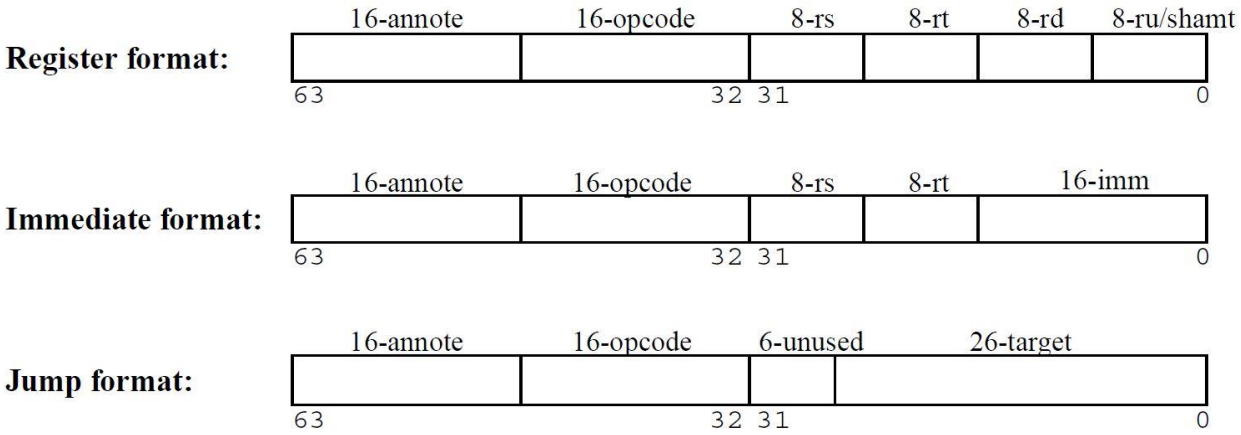

Figure 2. SimpleScalar architecture instruction formats

MCF Benchmark

MCF benchmark that was derived from a ANSI C language based program for single-depot vehicle scheduling problem in public mass transportation. The arithmetic operations in this benchmark are mostly integer-based. During the process of solving problems, the program considers one single depot and a homogenous vehicle fleet. Consequently, the timetabled trips with fixed departure or arrival locations and times according to a line plan and frequented service. There are also pull-out and pull-in trips for leaving and entering the depots in addition to the aforementioned ones. The core of MCF benchmark is the network complex algorithm is a specialized version of the well-known simplex algorithm for network flow problems. For increasing the execution speed, the linear algebra of the general algorithm is replaced by simple network operations.

The executable file of this benchmark takes an input file and generates an output file. The input fie consists of: (a) the number of timetabled and dead-head trips; (b) the starting time and ending time for each timetabled; (c) the cost and the starting and ending timetabled for each dead-head trip. The worst execution time is pseudo-polynomial in the number of timetabled and dead-head trips and in the amount of the maximal cost coefficient. The output file includes check output values describing an optimal schedule computed by the program. Meanwhile, the benchmark execution requires about 100 and 190 megabytes for a 32-bit and a 64-bit architecture.

Execution Commands

(1)

Command = `./sim-bpred -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in`

Description = The "`sim-bpred`" command is used to simulate the branch prediction behavior. In this command, "`Mcf`" shows the executable file, "`inp.in`" shows the input file, and the argument "-dumpconfig" is leveraged for dumping the processor configuration to the "`config_file.config`" file.

(2)

Command = `./sim-cache -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in`

Description = The "`sim-cache`" command is utilized for examining the cache behavior.

(3)

Command = `./sim-eio -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in`

Description = The "`sim-eio`" command is used to simulate external I/O trace generator.

(4)

Command = `./sim-fast -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in`

Description = This command has usage for sample fast functional simulator implementation.

(5)

Command = `./sim-outorder -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in`

Description = In order to simulate the out-of-order version of the SimpleScalar processor, this command can be used.

(6)

Command = `./sim-profile -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in`

Description = This command is used to create a profile along with the sample functional simulator implementation.

(7)

Command = `./sim-safe -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in`

Description = Lack of profile creation is the only difference between this command and the previous one.

Analysis of Outputs

According to the simulation results, the total number of executed instructions for all of the functionalities is 49073291786. The simulation speed in terms of total number of instructions per second for "`sim-outorder`" functionality is 300906.2255. With considering this amount as the reference value, the simulation speed for "`sim-cache`", "`sim-bpred`", "`sim-eio`", "`sim-profile`", "`sim-safe`", and "`sim-fast`" are 12.716, 28.112, 43.184, 43.28, 43.45, 50.905 times higher respectively. As it can be understood from the results, the "`sim-fast`" has the highest speed since more number of instructions are executed in less duration time. The miss rate for L-1 instruction cache, L-1 data cache, and data TLB (i.e. translation lookaside buffer) in the processor in-order mode are 0.6, 0.1, and 0.07 times higher than the out-of-order mode, the miss rate for L-2 cache in the out-of-order mode is 0.025 times higher than the in-order mode, and the miss rate for instruction TLB in both of the in-order and out-of-order modes are ~0.0. Regarding the total size of allocated memory pages, we have the same value for all of the seven architectural aspects that is 97740 KB. Finally, the instruction per cycle (IPC) parameter in the out-of-order mode that is one of the very important parameters for

performance evaluation of a processor architecture is equal to 0.3425. The ideal case is having execution of one instruction per clock cycle.

Argument Commands

1) Load Configuration from a File

Argument = `-config`

2) Dump Configuration to a File

Argument = `-dumpconfig`

3) Random Number Generator Seed (0 for timer seed)

Argument = `-seed 1`

4) Restore EIO Trace Execution from `<fname>`

Argument = `-chkpt`

5) Simulator Scheduling Priority

Argument = `-nice 0`

6) Maximum Number of Instructions to Execute

Argument = `-max:inst 0`

7) Branch Predictor Type {nottaken|taken|bimod|2lev|comb}

Argument = `-bpred bimod`

8) Bimodal Predictor Configuration (<table size>)

Argument = `-bpred:bimod = 2048`

9) 2-Level Predictor Configuration (<l1size> <l2size> <hist_size> <xor>)

Argument = `-bpred:2lev 1 1024 8 0`

10) Combining Predictor Configuration (<meta_table_size>)

Argument = `-bpred:comb 1024`

11) Return Address Stack Size (0 for no return stack)

Argument = `-bpred:ras = 8`

12) BTB config (<num_sets> <associativity>)

Argument = `-bpred:btb 512 4`

13) L-1 Data Cache Configuration, i.e., {<config>|none}

Argument = `-cache:dl1 = dl1:256:32:1:l`

14) L-2 Data Cache Configuration, i.e., {<config>|none}

Argument = `-cache:dl2 = ul2:1024:64:4:l`

15) L-1 Instruction Cache Configuration, i.e., {<config>|dl1|dl2|none}

Argument = `-cache:il1 = il1:256:32:1:l`

16) L-2 Instruction Cache Configuration, i.e., {<config>|dl2|none}

Argument = `-cache:il2 = dl2`

17) Instruction TLB Configuration, i.e., {<config>|none}

Argument = `-tlb:itlb = itlb:16:4096:4:l`

18) Data TLB Configuration, i.e., {<config>|none}

Argument = `-tlb:dtlb = dtlb:32:4096:4:l`

19) Flush Caches on System Calls

Argument = `-flush = false`

20) Convert 64-bit Instruction Addresses to 32-bit Instruction Equivalents

Argument = `-cache:icompress false`

21) Number of Instructions Skipped Before Tracing Starts

Argument = `-fastfwd 0`

22) EIO Trace File Output File Name

Argument = `-trace <null>`

23) Periodic Checkpoint Every “N” Instructions: `<base fname> <interval>`

Argument = `-perdump = <null>`

24) Specify Checkpoint File and Trigger: `<fname> <range>`

Argument = `-dump <null>`

25) Generate Pipetrace, i.e., `<fname|stdout|stderr> <range>`

Argument = `-ptrace <null>`

26) Instruction Fetch Queue Size (in Instructions)

Argument = `-fetch:ifqsize 4`

27) Extra Branch Mis-Prediction Latency

Argument = `-fetch:mplat 3`

28) Speed of Front-End of Machine Relative to Execution Core

Argument = `-fetch:speed 1`

29) Speculative Predictors Update in {`ID`|`WB`} (default non-spec)

Argument = `-bpred:spec_update = <null>`

30) Instruction Decode B/W (Instructions/Cycle)

Argument = `-decode:width 4`

31) Instruction Issue B/W (Instructions/Cycle)

Argument = `-issue:width 4`

32) Run Pipeline with In-Order Issue

Argument = `-issue:inorder false`

33) Issue Instructions Down Wrong Execution Paths

Argument = `-issue:wrongpath true`

34) Instruction Commit B/W (Instructions/Cycle)

Argument = `-commit:width = 4`

35) Register Update Unit (RUU) Size

Argument = `-ruu:size 16`

36) Load/Store Queue (LSQ) Size

Argument = `-lsq:size 8`

37) L-1 Data Cache Hit Latency (in Cycles)

Argument = `-cache:dl1lat 1`

38) L-2 Data Cache Hit Latency (in Cycles)

Argument = `-cache:dl2lat 6`

39) L-1 Instruction Cache Hit Latency (in Cycles)

Argument = `-cache:il1lat 1`

40) L-2 Instruction Cache Hit Latency (in Cycles)

Argument = `-cache:il2lat 6`

41) Flush Caches on System Calls

Argument = `-cache:flush false`

42) Memory access latency (<first_chunk> <inter_chunk>)

Argument = `-mem:lat 18 2`

43) Memory Access Bus Width (in Bytes)

Argument = `-mem:width 8`

44) Instruction/Data TLB Miss Latency (in Cycles)

Argument = `-tlb:lat 30`

45) Total Number of Integer ALU's Available

Argument = `-res:ialu 4`

46) Total Number of Integer Multiplier/Dividers Available

Argument = `-res:imult 1`

47) Total Number of Memory System Ports Available (to CPU)

Argument = `-res:memport 2`

48) Total Number of Floating Point ALU's Available

Argument = `-res:fpalu 4`

49) Total Number of Floating Point Multiplier/Dividers Available

Argument = `-res:fpmult 1`

50) Operate in Backward-Compatible Bugs Mode (for Testing Only)

Argument = `-bugcompat false`

51) Enable All Profile Options

Argument = `-all false`

52) Enable Instruction Class Profiling

Argument = `-iclass false`

53) Enable Instruction Profiling

Argument = `-iprof false`

54) Enable Branch Instruction Profiling

Argument = `-brprof false`

55) Enable Address Mode Profiling

Argument = `-amprof false`

56) Enable Load/Store Address Segment Profiling

Argument = `-segprof false`

57) Enable Text Symbol Profiling

Argument = `-tsymprof false`

58) Enable Text Address Profiling

Argument = `-taddrprof false`

59) Enable Data Symbol Profiling

Argument = `-dsymprof false`

60) Include Compiler-Internal Symbols During Symbol Profiling

Argument = `-internal false`

61) Profile Stat(s) Against Text Addresses (Mult uses ok)

Argument = `-pcstat = <null>`

Simulation Results

(1) sim-bpred

```
sh861201@eustis:~/SimpleScalar/simplesim-3.0/bpred_Dir$ ../sim-bpred -
dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-
3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-
3.0/benchmark/mcf/mcf/inp.in

sim-bpred: SimpleScalar/PISA 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).


sim: command line: ../sim-bpred -dumpconfig config_file.config
/home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf
/home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in


sim: simulation started @ Wed Feb  3 12:27:37 2016, options follow:


sim-bpred: This simulator implements a branch predictor analyzer.


# -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           <null> # 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

-bpred                  bimod # branch predictor type
{nottaken|taken|bimod|2lev|comb}

-bpred:bimod     2048 # bimodal predictor config (<table size>)

-bpred:2lev      1 1024 8 0 # 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>)


  Branch predictor configuration examples for 2-level predictor:

    Configurations:   N, M, W, X

      N   # entries in first level (# of shift register(s))

      W   width of shift register(s)

      M   # entries in 2nd level (# of counters, or other FSM)

      X   (yes-1/no-0) xor history and address for 2nd level index
```

```
    Sample predictors:
      GAg     : 1, W, 2^W, 0
      GAp     : 1, W, M (M > 2^W), 0
      PAg     : N, W, 2^W, 0
      PAp     : N, W, M (M == 2^(N+W)), 0
      gshare  : 1, W, 2^W, 1
  Predictor `comb' combines a bimodal and a 2-level predictor.

sim: ** starting functional simulation w/ predictors **

MCF SPEC version 1.6.I
by  Andreas Loebel
Copyright (c) 1998,1999   ZIB Berlin
All Rights Reserved.

nodes                      : 16555
active arcs                : 244246
simplex iterations         : 182415
flow value                 : 8980173901
new implicit arcs          : 300000
active arcs                : 544246
simplex iterations         : 189170
flow value                 : 8910169940
new implicit arcs          : 300000
active arcs                : 844246
simplex iterations         : 216493
flow value                 : 8650168945
new implicit arcs          : 300000
```

```
active arcs               : 1144246
simplex iterations        : 261464
flow value                : 8570161464
new implicit arcs         : 300000
active arcs               : 1444246
simplex iterations        : 290615
flow value                : 8570159306
new implicit arcs         : 300000
active arcs               : 1744246
simplex iterations        : 318729
flow value                : 8570157650
new implicit arcs         : 300000
active arcs               : 2044246
simplex iterations        : 340078
flow value                : 8570156531
new implicit arcs         : 300000
active arcs               : 2344246
simplex iterations        : 354548
flow value                : 8570156010
new implicit arcs         : 77333
active arcs               : 2421579
simplex iterations        : 361819
flow value                : 8570155949
new implicit arcs         : 1100
active arcs               : 2422679
simplex iterations        : 361826
flow value                : 8570155949
checksum                  : 258659426
```

```
optimal


sim: ** simulation statistics **
sim_num_insn            49073291786 # total number of instructions
executed
sim_num_refs            20768099818 # total number of loads and stores
executed
sim_elapsed_time               5602 # total simulation time in seconds
sim_inst_rate          8759959.2620 # simulation speed (in insts/sec)
sim_num_branches        12052458319 # total number of branches
executed
sim_IPB                      4.0716 # instruction per branch
bpred_bimod.lookups     12052458319 # total number of bpred lookups
bpred_bimod.updates     12052458319 # total number of updates
bpred_bimod.addr_hits   10890361653 # total number of address-
predicted hits
bpred_bimod.dir_hits    10891212540 # total number of direction-
predicted hits (includes addr-hits)
bpred_bimod.misses       1161245779 # total number of misses
bpred_bimod.jr_hits        40089720 # total number of address-
predicted hits for JR's
bpred_bimod.jr_seen        40956662 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP      1250351 # total number of address-
predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP      1266947 # total number of non-RAS
JR's seen
bpred_bimod.bpred_addr_rate    0.9036 # branch address-prediction rate
(i.e., addr-hits/updates)
bpred_bimod.bpred_dir_rate    0.9037 # branch direction-prediction
rate (i.e., all-hits/updates)
bpred_bimod.bpred_jr_rate    0.9788 # JR address-prediction rate
(i.e., JR addr-hits/JRs seen)
```

```
bpred_bimod.bpred_jr_non_ras_rate.PP    0.9869 # non-RAS JR addr-pred
rate (ie, non-RAS JR hits/JRs seen)

bpred_bimod.retstack_pushes     39689717 # total number of address
pushed onto ret-addr stack

bpred_bimod.retstack_pops     39689715 # total number of address
popped off of ret-addr stack

bpred_bimod.used_ras.PP     39689715 # total number of RAS predictions
used

bpred_bimod.ras_hits.PP     38839369 # total number of RAS hits

bpred_bimod.ras_rate.PP    0.9786 # RAS prediction rate (i.e., RAS
hits/used RAS)
```

(2) sim-cache

```
sh861201@eustis:~/SimpleScalar/simplesim-3.0/cache_Dir$ ../sim-cache -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in

sim-cache: SimpleScalar/PISA 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).


sim: command line: ../sim-cache -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in


sim: simulation started @ Tue Feb  2 16:32:36 2016, options follow:


sim-cache: This simulator implements a functional cache simulator.  Cache

statistics are generated for a user-selected cache and TLB configuration,

which may include up to two levels of instruction and data cache (with any

levels unified), and one level of instruction and data TLBs.  No timing

information is generated.
```

```
# -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            <null> # 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
-cache:dl1       dl1:256:32:1:l # l1 data cache config, i.e.,
{<config>|none}
-cache:dl2       ul2:1024:64:4:l # l2 data cache config, i.e.,
{<config>|none}
-cache:il1       il1:256:32:1:l # l1 inst cache config, i.e.,
{<config>|dl1|dl2|none}
-cache:il2                 dl2 # l2 instruction cache config, i.e.,
{<config>|dl2|none}
-tlb:itlb        itlb:16:4096:4:l # instruction TLB config, i.e.,
{<config>|none}
-tlb:dtlb        dtlb:32:4096:4:l # data TLB config, i.e.,
{<config>|none}
-flush                false # flush caches on system calls
```

```
-cache:icompress        false # convert 64-bit inst addresses to 32-
bit inst equivalents
# -pcstat              <null> # profile stat(s) against text addr's
(mult uses ok)

  The cache config parameter <config> 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
                -dtlb dtlb:128:4096:32:r

  Cache levels can be unified by pointing a level of the instruction
cache
  hierarchy at the data cache hiearchy using the "dl1" and "dl2" cache
  configuration arguments.  Most sensible combinations are supported,
e.g.,

    A unified l2 cache (il2 is pointed at dl2):
      -cache:il1 il1:128:64:1:l -cache:il2 dl2
      -cache:dl1 dl1:256:32:1:l -cache:dl2 ul2:1024:64:2:l

    Or, a fully unified cache hierarchy (il1 pointed at dl1):
```

```
      -cache:il1 dl1
      -cache:dl1 ul1:256:32:1:l -cache:dl2 ul2:1024:64:2:l

sim: ** starting functional simulation w/ caches **

MCF SPEC version 1.6.I
by  Andreas Loebel
Copyright (c) 1998,1999   ZIB Berlin
All Rights Reserved.

nodes                      : 16555
active arcs                : 244246
simplex iterations         : 182415
flow value                 : 8980173901
new implicit arcs          : 300000
active arcs                : 544246
simplex iterations         : 189170
flow value                 : 8910169940
new implicit arcs          : 300000
active arcs                : 844246
simplex iterations         : 216493
flow value                 : 8650168945
new implicit arcs          : 300000
active arcs                : 1144246
simplex iterations         : 261464
flow value                 : 8570161464
new implicit arcs          : 300000
active arcs                : 1444246
```

```
simplex iterations         : 290615
flow value                 : 8570159306
new implicit arcs          : 300000
active arcs                : 1744246
simplex iterations         : 318729
flow value                 : 8570157650
new implicit arcs          : 300000
active arcs                : 2044246
simplex iterations         : 340078
flow value                 : 8570156531
new implicit arcs          : 300000
active arcs                : 2344246
simplex iterations         : 354548
flow value                 : 8570156010
new implicit arcs          : 77333
active arcs                : 2421579
simplex iterations         : 361819
flow value                 : 8570155949
new implicit arcs          : 1100
active arcs                : 2422679
simplex iterations         : 361826
flow value                 : 8570155949
checksum                   : 258659426
optimal


sim: ** simulation statistics **
sim_num_insn            49073291786 # total number of instructions
executed
```

```
sim_num_refs            20768099818 # total number of loads and stores
executed
sim_elapsed_time              11890 # total simulation time in seconds
sim_inst_rate          4127274.3302 # simulation speed (in insts/sec)
il1.accesses            49073291786 # total number of accesses
il1.hits                49031987632 # total number of hits
il1.misses                 41304154 # total number of misses
il1.replacements           41303898 # total number of replacements
il1.writebacks                    0 # total number of writebacks
il1.invalidations                 0 # total number of invalidations
il1.miss_rate                0.0008 # miss rate (i.e., misses/ref)
il1.repl_rate                0.0008 # replacement rate (i.e.,
repls/ref)
il1.wb_rate                  0.0000 # writeback rate (i.e., wrbks/ref)
il1.inv_rate                 0.0000 # invalidation rate (i.e.,
invs/ref)
dl1.accesses            20771127189 # total number of accesses
dl1.hits                13498624696 # total number of hits
dl1.misses               7272502493 # total number of misses
dl1.replacements         7272502237 # total number of replacements
dl1.writebacks           1367768305 # total number of writebacks
dl1.invalidations                 0 # total number of invalidations
dl1.miss_rate                0.3501 # miss rate (i.e., misses/ref)
dl1.repl_rate                0.3501 # replacement rate (i.e.,
repls/ref)
dl1.wb_rate                  0.0658 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate                 0.0000 # invalidation rate (i.e.,
invs/ref)
ul2.accesses             8681574952 # total number of accesses
ul2.hits                 3468565682 # total number of hits
```

```
ul2.misses               5213009270 # total number of misses
ul2.replacements         5213005174 # total number of replacements
ul2.writebacks           1207279599 # total number of writebacks
ul2.invalidations                 0 # total number of invalidations
ul2.miss_rate                0.6005 # miss rate (i.e., misses/ref)
ul2.repl_rate                0.6005 # replacement rate (i.e.,
repls/ref)
ul2.wb_rate                  0.1391 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate                 0.0000 # invalidation rate (i.e.,
invs/ref)
itlb.accesses           49073291786 # total number of accesses
itlb.hits               49073291759 # total number of hits
itlb.misses                      27 # total number of misses
itlb.replacements                 0 # total number of replacements
itlb.writebacks                   0 # total number of writebacks
itlb.invalidations                0 # total number of invalidations
itlb.miss_rate               0.0000 # miss rate (i.e., misses/ref)
itlb.repl_rate               0.0000 # replacement rate (i.e.,
repls/ref)
itlb.wb_rate                 0.0000 # writeback rate (i.e., wrbks/ref)
itlb.inv_rate                0.0000 # invalidation rate (i.e.,
invs/ref)
dtlb.accesses           20771127189 # total number of accesses
dtlb.hits               19090713234 # total number of hits
dtlb.misses              1680413955 # total number of misses
dtlb.replacements        1680413827 # total number of replacements
dtlb.writebacks           365182120 # total number of writebacks
dtlb.invalidations                0 # total number of invalidations
dtlb.miss_rate               0.0809 # miss rate (i.e., misses/ref)
```

```
dtlb.repl_rate                0.0809 # replacement rate (i.e.,
repls/ref)

dtlb.wb_rate                  0.0176 # writeback rate (i.e., wrbks/ref)

dtlb.inv_rate                 0.0000 # invalidation rate (i.e.,
invs/ref)

ld_text_base             0x00400000 # program text (code) segment base

ld_text_size                  113136 # program text (code) size in
bytes

ld_data_base             0x10000000 # program initialized data segment
base

ld_data_size                   19060 # program init'ed `.data' and
uninit'ed `.bss' size in bytes

ld_stack_base            0x7fffc000 # program stack segment base
(highest address in stack)

ld_stack_size                  16384 # program initial stack size

ld_prog_entry            0x00400140 # program entry point (initial PC)

ld_environ_base          0x7fff8000 # program environment base address
address

ld_target_big_endian               0 # target executable endian-ness,
non-zero if big endian

mem.page_count                 24435 # total number of pages allocated

mem.page_mem                   97740k # total size of memory pages
allocated

mem.ptab_misses            10414410 # total first level page table
misses

mem.ptab_accesses      237836123985 # total page table accesses

mem.ptab_miss_rate           0.0000 # first level page table miss rate
```

(3) sim-eio

```
sh861201@eustis:~/SimpleScalar/simplesim-3.0/eio_Dir$ ../sim-eio -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in

sim-eio: SimpleScalar/PISA 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).


sim: command line: ../sim-eio -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in


sim: simulation started @ Wed Feb  3 14:20:30 2016, options follow:


sim-eio: This simulator implements simulator support for generating

external event traces (EIO traces) and checkpoint files.  External

event traces capture one execution of a program, and allow it to be

packaged into a single file for later re-execution.  EIO trace executions

are 100% reproducible between subsequent executions (on the same platform.

This simulator also provides functionality to generate checkpoints at

arbitrary points within an external event trace (EIO) execution.  The
```

checkpoint file (along with the EIO trace) can be used to start any SimpleScalar simulator in the middle of a program execution.

```
# -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           <null> # 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 tracing
starts
# -trace               <null> # EIO trace file output file name
# -perdump             <null> # periodic checkpoint every n
instructions: <base fname> <interval>
# -dump                <null> # specify checkpoint file and trigger:
<fname> <range>
```

```
  Checkpoint range triggers are formatted as follows:

    {{@|#}<start>}:{{@|#|+}<end>}

  Both ends of the range are optional, if neither are specified, the
range
  triggers immediately.  Ranges that start with a `@' designate an
address
  range to trigger on, those that start with an `#' designate a cycle
count
  trigger.  All other ranges represent an instruction count range.
The
  second argument, if specified with a `+', indicates a value relative
  to the first argument, e.g., 1000:+100 == 1000:1100.

    Examples:   -ptrace FOO.trc #0:#1000
                -ptrace BAR.trc @2000:
                -ptrace BLAH.trc :1500
                -ptrace UXXE.trc :

sim: ** starting functional simulation **

MCF SPEC version 1.6.I
by  Andreas Loebel
Copyright (c) 1998,1999   ZIB Berlin
All Rights Reserved.

nodes                      : 16555
active arcs                : 244246
```

```
simplex iterations         : 182415
flow value                 : 8980173901
new implicit arcs          : 300000
active arcs                : 544246
simplex iterations         : 189170
flow value                 : 8910169940
new implicit arcs          : 300000
active arcs                : 844246
simplex iterations         : 216493
flow value                 : 8650168945
new implicit arcs          : 300000
active arcs                : 1144246
simplex iterations         : 261464
flow value                 : 8570161464
new implicit arcs          : 300000
active arcs                : 1444246
simplex iterations         : 290615
flow value                 : 8570159306
new implicit arcs          : 300000
active arcs                : 1744246
simplex iterations         : 318729
flow value                 : 8570157650
new implicit arcs          : 300000
active arcs                : 2044246
simplex iterations         : 340078
flow value                 : 8570156531
new implicit arcs          : 300000
active arcs                : 2344246
```

```
simplex iterations        : 354548
flow value                : 8570156010
new implicit arcs         : 77333
active arcs               : 2421579
simplex iterations        : 361819
flow value                : 8570155949
new implicit arcs         : 1100
active arcs               : 2422679
simplex iterations        : 361826
flow value                : 8570155949
checksum                  : 258659426
optimal


sim: ** simulation statistics **
sim_num_insn            49073291786 # total number of instructions
executed
sim_num_refs            20768099818 # total number of loads and stores
executed
sim_elapsed_time               3691 # total simulation time in seconds
sim_inst_rate          13295391.9767 # simulation speed (in insts/sec)
ld_text_base             0x00400000 # program text (code) segment base
ld_text_size                113136 # program text (code) size in
bytes
ld_data_base             0x10000000 # program initialized data segment
base
ld_data_size                 19060 # program init'ed `.data' and
uninit'ed `.bss' size in bytes
ld_stack_base            0x7fffc000 # program stack segment base
(highest address in stack)
ld_stack_size                16384 # program initial stack size
```

```
ld_prog_entry            0x00400140 # program entry point (initial PC)
ld_environ_base          0x7fff8000 # program environment base address
address
ld_target_big_endian              0 # target executable endian-ness,
non-zero if big endian
mem.page_count                24435 # total number of pages allocated
mem.page_mem                  97740k # total size of memory pages
allocated
mem.ptab_misses            10414410 # total first level page table
misses
mem.ptab_accesses      237836123977 # total page table accesses
mem.ptab_miss_rate           0.0000 # first level page table miss rate
```

(4) sim-fast

```
sh861201@eustis:~/SimpleScalar/simplesim-3.0/fast_Dir$ ../sim-fast -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in

sim-fast: SimpleScalar/PISA 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).


sim: command line: ../sim-fast -dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in


sim: simulation started @ Wed Feb  3 15:52:24 2016, options follow:


sim-fast: This simulator implements a very fast functional simulator. This

functional simulator implementation is much more difficult to digest than

the simpler, cleaner sim-safe functional simulator.  By default, this

simulator performs no instruction error checking, as a result, any

instruction errors will manifest as simulator execution errors, possibly

causing sim-fast to execute incorrectly or dump core.  Such is the
```

```
price we pay for speed!!!!

# -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           <null> # redirect simulator output to file
(non-interactive only)
# -redir:prog          <null> # redirect simulated program output to
file
-nice                       0 # simulator scheduling priority

sim: ** starting *fast* functional simulation **

MCF SPEC version 1.6.I
by  Andreas Loebel
Copyright (c) 1998,1999   ZIB Berlin
All Rights Reserved.

nodes                     : 16555
active arcs               : 244246
simplex iterations        : 182415
```

```
flow value                : 8980173901
new implicit arcs         : 300000
active arcs               : 544246
simplex iterations        : 189170
flow value                : 8910169940
new implicit arcs         : 300000
active arcs               : 844246
simplex iterations        : 216493
flow value                : 8650168945
new implicit arcs         : 300000
active arcs               : 1144246
simplex iterations        : 261464
flow value                : 8570161464
new implicit arcs         : 300000
active arcs               : 1444246
simplex iterations        : 290615
flow value                : 8570159306
new implicit arcs         : 300000
active arcs               : 1744246
simplex iterations        : 318729
flow value                : 8570157650
new implicit arcs         : 300000
active arcs               : 2044246
simplex iterations        : 340078
flow value                : 8570156531
new implicit arcs         : 300000
active arcs               : 2344246
simplex iterations        : 354548
```

```
flow value                  : 8570156010
new implicit arcs           : 77333
active arcs                 : 2421579
simplex iterations          : 361819
flow value                  : 8570155949
new implicit arcs           : 1100
active arcs                 : 2422679
simplex iterations          : 361826
flow value                  : 8570155949
checksum                    : 258659426
optimal


sim: ** simulation statistics **
sim_num_insn            49073291786 # total number of instructions
executed
sim_elapsed_time               3142 # total simulation time in seconds
sim_inst_rate          15618488.7925 # simulation speed (in insts/sec)
ld_text_base             0x00400000 # program text (code) segment base
ld_text_size                 113136 # program text (code) size in
bytes
ld_data_base             0x10000000 # program initialized data segment
base
ld_data_size                  19060 # program init'ed `.data' and
uninit'ed `.bss' size in bytes
ld_stack_base            0x7fffc000 # program stack segment base
(highest address in stack)
ld_stack_size                 16384 # program initial stack size
ld_prog_entry            0x00400140 # program entry point (initial PC)
ld_environ_base          0x7fff8000 # program environment base address
address
```

```
ld_target_big_endian              0 # target executable endian-ness,
non-zero if big endian

mem.page_count                24435 # total number of pages allocated

mem.page_mem                  97740k # total size of memory pages
allocated

mem.ptab_misses            10414410 # total first level page table
misses

mem.ptab_accesses      237836123981 # total page table accesses

mem.ptab_miss_rate           0.0000 # first level page table miss rate
```

(5) sim-outorder

```
sh861201@eustis:~/SimpleScalar/simplesim-3.0/outorder_Dir$ ../sim-
outorder -dumpconfig config_file.config
/home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf
/home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in

sim-outorder: SimpleScalar/PISA 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).


sim: command line: ../sim-outorder -dumpconfig config_file.config
/home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf
/home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in


sim: simulation started @ Wed Feb  3 16:48:37 2016, 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           <null> # 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 0 # 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       dl1:128:32:4:l # l1 data cache config, i.e.,
{<config>|none}

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

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

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

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

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

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

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

-cache:flush            false # flush caches on system calls

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

-mem:lat         18 2 # memory access latency (<first_chunk>
<inter_chunk>)

-mem:width                  8 # memory access bus width (in bytes)
```

```
-tlb:itlb        itlb:16:4096:4:l # instruction TLB config, i.e.,
{<config>|none}

-tlb:dtlb        dtlb:32:4096:4:l # data TLB config, i.e.,
{<config>|none}

-tlb:lat                   30 # inst/data TLB miss latency (in cycles)

-res:ialu                   4 # total number of integer ALU's
available

-res:imult                  1 # total number of integer
multiplier/dividers available

-res:memport                2 # total number of memory system ports
available (to CPU)

-res:fpalu                  4 # total number of floating point ALU's
available

-res:fpmult                 1 # total number of floating point
multiplier/dividers available

# -pcstat              <null> # profile stat(s) against text addr's
(mult uses ok)

-bugcompat               false # operate in backward-compatible bugs
mode (for testing only)


  Pipetrace range arguments are formatted as follows:


    {{@|#}<start>}:{{@|#|+}<end>}


  Both ends of the range are optional, if neither are specified, the
entire

  execution is traced.  Ranges that start with a `@' designate an
address

  range to be traced, those that start with an `#' designate a cycle
count

  range.  All other range values represent an instruction count range.
The
```

```
  second argument, if specified with a `+', indicates a value relative
  to the first argument, e.g., 1000:+100 == 1000:1100.  Program
symbols may
  be used in all contexts.

    Examples:   -ptrace FOO.trc #0:#1000
                -ptrace BAR.trc @2000:
                -ptrace BLAH.trc :1500
                -ptrace UXXE.trc :
                -ptrace FOOBAR.trc @main:+278

  Branch predictor configuration examples for 2-level predictor:
    Configurations:   N, M, W, X
      N   # entries in first level (# of shift register(s))
      W   width of shift register(s)
      M   # entries in 2nd level (# of counters, or other FSM)
      X   (yes-1/no-0) xor history and address for 2nd level index
    Sample predictors:
      GAg     : 1, W, 2^W, 0
      GAp     : 1, W, M (M > 2^W), 0
      PAg     : N, W, 2^W, 0
      PAp     : N, W, M (M == 2^(N+W)), 0
      gshare  : 1, W, 2^W, 1
  Predictor `comb' combines a bimodal and a 2-level predictor.

  The cache config parameter <config> 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
                -dtlb dtlb:128:4096:32:r

  Cache levels can be unified by pointing a level of the instruction
cache
  hierarchy at the data cache hiearchy using the "dl1" and "dl2" cache
  configuration arguments.  Most sensible combinations are supported,
e.g.,

    A unified l2 cache (il2 is pointed at dl2):
      -cache:il1 il1:128:64:1:l -cache:il2 dl2
      -cache:dl1 dl1:256:32:1:l -cache:dl2 ul2:1024:64:2:l

    Or, a fully unified cache hierarchy (il1 pointed at dl1):
      -cache:il1 dl1
      -cache:dl1 ul1:256:32:1:l -cache:dl2 ul2:1024:64:2:l

sim: ** starting performance simulation **

MCF SPEC version 1.6.I
by  Andreas Loebel
```

Copyright (c) 1998,1999   ZIB Berlin
All Rights Reserved.

nodes   : 16555
active arcs   : 244246
simplex iterations   : 182415
flow value   : 8980173901
new implicit arcs   : 300000
active arcs   : 544246
simplex iterations   : 189170
flow value   : 8910169940
new implicit arcs   : 300000
active arcs   : 844246
simplex iterations   : 216493
flow value   : 8650168945
new implicit arcs   : 300000
active arcs   : 1144246
simplex iterations   : 261464
flow value   : 8570161464
new implicit arcs   : 300000
active arcs   : 1444246
simplex iterations   : 290615
flow value   : 8570159306
new implicit arcs   : 300000
active arcs   : 1744246
simplex iterations   : 318729
flow value   : 8570157650
new implicit arcs   : 300000

```
active arcs               : 2044246
simplex iterations        : 340078
flow value                : 8570156531
new implicit arcs         : 300000
active arcs               : 2344246
simplex iterations        : 354548
flow value                : 8570156010
new implicit arcs         : 77333
active arcs               : 2421579
simplex iterations        : 361819
flow value                : 8570155949
new implicit arcs         : 1100
active arcs               : 2422679
simplex iterations        : 361826
flow value                : 8570155949
checksum                  : 258659426
optimal


sim: ** simulation statistics **
sim_num_insn            49073291786 # total number of instructions
committed
sim_num_refs            20768099818 # total number of loads and stores
committed
sim_num_loads           18041672686 # total number of loads committed
sim_num_stores         2726427132.0000 # total number of stores
committed
sim_num_branches        12052458319 # total number of branches
committed
sim_elapsed_time             163085 # total simulation time in seconds
```

```
sim_inst_rate           300906.2255 # simulation speed (in insts/sec)
sim_total_insn          58786702113 # total number of instructions
executed
sim_total_refs          25992887011 # total number of loads and stores
executed
sim_total_loads         22737122025 # total number of loads executed
sim_total_stores      3255764986.0000 # total number of stores
executed
sim_total_branches      14110463801 # total number of branches
executed
sim_cycle             143296637320 # total simulation time in cycles
sim_IPC                     0.3425 # instructions per cycle
sim_CPI                     2.9201 # cycles per instruction
sim_exec_BW                 0.4102 # total instructions (mis-spec +
committed) per cycle
sim_IPB                     4.0716 # instruction per branch
IFQ_count             543163217883 # cumulative IFQ occupancy
IFQ_fcount            132402474202 # cumulative IFQ full count
ifq_occupancy               3.7905 # avg IFQ occupancy (insn's)
ifq_rate                    0.4102 # avg IFQ dispatch rate
(insn/cycle)
ifq_latency                 9.2396 # avg IFQ occupant latency
(cycle's)
ifq_full                    0.9240 # fraction of time (cycle's) IFQ
was full
RUU_count             2109935503690 # cumulative RUU occupancy
RUU_fcount            100780909996 # cumulative RUU full count
ruu_occupancy              14.7242 # avg RUU occupancy (insn's)
ruu_rate                    0.4102 # avg RUU dispatch rate
(insn/cycle)
```

```
ruu_latency                35.8914 # avg RUU occupant latency
(cycle's)
ruu_full                    0.7033 # fraction of time (cycle's) RUU
was full
LSQ_count             971656506743 # cumulative LSQ occupancy
LSQ_fcount             66817724062 # cumulative LSQ full count
lsq_occupancy               6.7807 # avg LSQ occupancy (insn's)
lsq_rate                    0.4102 # avg LSQ dispatch rate
(insn/cycle)
lsq_latency                16.5285 # avg LSQ occupant latency
(cycle's)
lsq_full                    0.4663 # fraction of time (cycle's) LSQ
was full
sim_slip              6747556112725567395 # total number of slip
cycles
avg_sim_slip          137499561.7199 # the average slip between issue
and retirement
bpred_bimod.lookups     15290928081 # total number of bpred lookups
bpred_bimod.updates     12052458319 # total number of updates
bpred_bimod.addr_hits   10896784509 # total number of address-
predicted hits
bpred_bimod.dir_hits    10897952939 # total number of direction-
predicted hits (includes addr-hits)
bpred_bimod.misses       1154505380 # total number of misses
bpred_bimod.jr_hits        39772187 # total number of address-
predicted hits for JR's
bpred_bimod.jr_seen        40956662 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP      1250351 # total number of address-
predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP      1266947 # total number of non-RAS
JR's seen
```

```
bpred_bimod.bpred_addr_rate    0.9041 # branch address-prediction rate
(i.e., addr-hits/updates)

bpred_bimod.bpred_dir_rate    0.9042 # branch direction-prediction
rate (i.e., all-hits/updates)

bpred_bimod.bpred_jr_rate    0.9711 # JR address-prediction rate
(i.e., JR addr-hits/JRs seen)

bpred_bimod.bpred_jr_non_ras_rate.PP    0.9869 # non-RAS JR addr-pred
rate (ie, non-RAS JR hits/JRs seen)

bpred_bimod.retstack_pushes     73535045 # total number of address
pushed onto ret-addr stack

bpred_bimod.retstack_pops     58199448 # total number of address
popped off of ret-addr stack

bpred_bimod.used_ras.PP     39689715 # total number of RAS predictions
used

bpred_bimod.ras_hits.PP     38521836 # total number of RAS hits

bpred_bimod.ras_rate.PP    0.9706 # RAS prediction rate (i.e., RAS
hits/used RAS)

il1.accesses            63258957356 # total number of accesses

il1.hits                63229170607 # total number of hits

il1.misses                 29786749 # total number of misses

il1.replacements           29786242 # total number of replacements

il1.writebacks                    0 # total number of writebacks

il1.invalidations                 0 # total number of invalidations

il1.miss_rate                0.0005 # miss rate (i.e., misses/ref)

il1.repl_rate                0.0005 # replacement rate (i.e.,
repls/ref)

il1.wb_rate                  0.0000 # writeback rate (i.e., wrbks/ref)

il1.inv_rate                 0.0000 # invalidation rate (i.e.,
invs/ref)

dl1.accesses            22403117765 # total number of accesses

dl1.hits                15298616499 # total number of hits
```

```
dl1.misses               7104501266 # total number of misses
dl1.replacements         7104500754 # total number of replacements
dl1.writebacks           1334405193 # total number of writebacks
dl1.invalidations                 0 # total number of invalidations
dl1.miss_rate                0.3171 # miss rate (i.e., misses/ref)
dl1.repl_rate                0.3171 # replacement rate (i.e.,
repls/ref)
dl1.wb_rate                  0.0596 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate                 0.0000 # invalidation rate (i.e.,
invs/ref)
ul2.accesses             8468693208 # total number of accesses
ul2.hits                 3251951709 # total number of hits
ul2.misses               5216741499 # total number of misses
ul2.replacements         5216737403 # total number of replacements
ul2.writebacks           1207172414 # total number of writebacks
ul2.invalidations                 0 # total number of invalidations
ul2.miss_rate                0.6160 # miss rate (i.e., misses/ref)
ul2.repl_rate                0.6160 # replacement rate (i.e.,
repls/ref)
ul2.wb_rate                  0.1425 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate                 0.0000 # invalidation rate (i.e.,
invs/ref)
itlb.accesses           63258957356 # total number of accesses
itlb.hits               63258957329 # total number of hits
itlb.misses                      27 # total number of misses
itlb.replacements                 0 # total number of replacements
itlb.writebacks                   0 # total number of writebacks
itlb.invalidations                0 # total number of invalidations
itlb.miss_rate               0.0000 # miss rate (i.e., misses/ref)
```

```
itlb.repl_rate                0.0000 # replacement rate (i.e.,
repls/ref)
itlb.wb_rate                  0.0000 # writeback rate (i.e., wrbks/ref)
itlb.inv_rate                 0.0000 # invalidation rate (i.e.,
invs/ref)
dtlb.accesses           22426137324 # total number of accesses
dtlb.hits               20739675328 # total number of hits
dtlb.misses              1686461996 # total number of misses
dtlb.replacements        1686461868 # total number of replacements
dtlb.writebacks                 0 # total number of writebacks
dtlb.invalidations              0 # total number of invalidations
dtlb.miss_rate                0.0752 # miss rate (i.e., misses/ref)
dtlb.repl_rate                0.0752 # replacement rate (i.e.,
repls/ref)
dtlb.wb_rate                  0.0000 # writeback rate (i.e., wrbks/ref)
dtlb.inv_rate                 0.0000 # invalidation rate (i.e.,
invs/ref)
sim_invalid_addrs               0 # total non-speculative bogus
addresses seen (debug var)
ld_text_base             0x00400000 # program text (code) segment base
ld_text_size                113136 # program text (code) size in
bytes
ld_data_base             0x10000000 # program initialized data segment
base
ld_data_size                 19060 # program init'ed `.data' and
uninit'ed `.bss' size in bytes
ld_stack_base            0x7fffc000 # program stack segment base
(highest address in stack)
ld_stack_size                16384 # program initial stack size
ld_prog_entry            0x00400140 # program entry point (initial PC)
```

```
ld_environ_base          0x7fff8000 # program environment base address
address
ld_target_big_endian             0 # target executable endian-ness,
non-zero if big endian
mem.page_count               24435 # total number of pages allocated
mem.page_mem                 97740k # total size of memory pages
allocated
mem.ptab_misses             6774601 # total first level page table
misses
mem.ptab_accesses      426790608940 # total page table accesses
mem.ptab_miss_rate           0.0000 # first level page table miss rate
```

(6) sim-profile

```
sh861201@eustis:~/SimpleScalar/simplesim-3.0/profile_Dir$ ../sim-
profile -dumpconfig config_file.config
/home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf
/home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in

sim-profile: SimpleScalar/PISA 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).


sim: command line: ../sim-profile -dumpconfig config_file.config
/home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf
/home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in


sim: simulation started @ Fri Feb  5 14:48:23 2016, options follow:


sim-profile: This simulator implements a functional simulator with

profiling support.  Run with the `-h' flag to see profiling options

available.


# -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           <null> # 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
-all                  false # enable all profile options
-iclass               false # enable instruction class profiling
-iprof                false # enable instruction profiling
-brprof               false # enable branch instruction profiling
-amprof               false # enable address mode profiling
-segprof              false # enable load/store address segment
profiling
-tsymprof             false # enable text symbol profiling
-taddrprof            false # enable text address profiling
-dsymprof             false # enable data symbol profiling
-internal             false # include compiler-internal symbols
during symbol profiling
# -pcstat              <null> # profile stat(s) against text addr's
(mult uses ok)


sim: ** starting functional simulation **
```

```
MCF SPEC version 1.6.I
by  Andreas Loebel
Copyright (c) 1998,1999   ZIB Berlin
All Rights Reserved.

nodes                      : 16555
active arcs                : 244246
simplex iterations         : 182415
flow value                 : 8980173901
new implicit arcs          : 300000
active arcs                : 544246
simplex iterations         : 189170
flow value                 : 8910169940
new implicit arcs          : 300000
active arcs                : 844246
simplex iterations         : 216493
flow value                 : 8650168945
new implicit arcs          : 300000
active arcs                : 1144246
simplex iterations         : 261464
flow value                 : 8570161464
new implicit arcs          : 300000
active arcs                : 1444246
simplex iterations         : 290615
flow value                 : 8570159306
new implicit arcs          : 300000
active arcs                : 1744246
simplex iterations         : 318729
```

```
flow value                  : 8570157650
new implicit arcs           : 300000
active arcs                 : 2044246
simplex iterations          : 340078
flow value                  : 8570156531
new implicit arcs           : 300000
active arcs                 : 2344246
simplex iterations          : 354548
flow value                  : 8570156010
new implicit arcs           : 77333
active arcs                 : 2421579
simplex iterations          : 361819
flow value                  : 8570155949
new implicit arcs           : 1100
active arcs                 : 2422679
simplex iterations          : 361826
flow value                  : 8570155949
checksum                    : 258659426
optimal


sim: ** simulation statistics **
sim_num_insn            49073291786 # total number of instructions
executed
sim_num_refs            20768099818 # total number of loads and stores
executed
sim_elapsed_time               3683 # total simulation time in seconds
sim_inst_rate          13324271.4597 # simulation speed (in insts/sec)
ld_text_base             0x00400000 # program text (code) segment base
```

```
ld_text_size                113136 # program text (code) size in
bytes
ld_data_base             0x10000000 # program initialized data segment
base
ld_data_size                 19060 # program init'ed `.data' and
uninit'ed `.bss' size in bytes
ld_stack_base            0x7fffc000 # program stack segment base
(highest address in stack)
ld_stack_size                16384 # program initial stack size
ld_prog_entry            0x00400140 # program entry point (initial PC)
ld_environ_base          0x7fff8000 # program environment base address
address
ld_target_big_endian             0 # target executable endian-ness,
non-zero if big endian
mem.page_count               24435 # total number of pages allocated
mem.page_mem                 97740k # total size of memory pages
allocated
mem.ptab_misses            10414410 # total first level page table
misses
mem.ptab_accesses      237836123993 # total page table accesses
mem.ptab_miss_rate           0.0000 # first level page table miss rate
```

(7) sim-safe

```
sh861201@eustis:~/SimpleScalar/simplesim-3.0/safe_Dir$ ../sim-safe -
dumpconfig config_file.config /home/sh861201/SimpleScalar/simplesim-
3.0/benchmark/mcf/mcf/Mcf /home/sh861201/SimpleScalar/simplesim-
3.0/benchmark/mcf/mcf/inp.in

sim-safe: SimpleScalar/PISA 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).


sim: command line: ../sim-safe -dumpconfig config_file.config
/home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/Mcf
/home/sh861201/SimpleScalar/simplesim-3.0/benchmark/mcf/mcf/inp.in


sim: simulation started @ Fri Feb  5 17:10:52 2016, options follow:


sim-safe: This simulator implements a functional simulator.  This

functional simulator is the simplest, most user-friendly simulator in
the

simplescalar tool set.  Unlike sim-fast, this functional simulator
checks

for all instruction errors, and the implementation is crafted for
clarity

rather than speed.
```

```
# -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           <null> # 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

sim: ** starting functional simulation **

MCF SPEC version 1.6.I
by  Andreas Loebel
Copyright (c) 1998,1999   ZIB Berlin
All Rights Reserved.

nodes                     : 16555
active arcs               : 244246
simplex iterations        : 182415
flow value                : 8980173901
```

```
new implicit arcs          : 300000
active arcs                : 544246
simplex iterations         : 189170
flow value                 : 8910169940
new implicit arcs          : 300000
active arcs                : 844246
simplex iterations         : 216493
flow value                 : 8650168945
new implicit arcs          : 300000
active arcs                : 1144246
simplex iterations         : 261464
flow value                 : 8570161464
new implicit arcs          : 300000
active arcs                : 1444246
simplex iterations         : 290615
flow value                 : 8570159306
new implicit arcs          : 300000
active arcs                : 1744246
simplex iterations         : 318729
flow value                 : 8570157650
new implicit arcs          : 300000
active arcs                : 2044246
simplex iterations         : 340078
flow value                 : 8570156531
new implicit arcs          : 300000
active arcs                : 2344246
simplex iterations         : 354548
flow value                 : 8570156010
```

```
new implicit arcs          : 77333
active arcs                : 2421579
simplex iterations         : 361819
flow value                 : 8570155949
new implicit arcs          : 1100
active arcs                : 2422679
simplex iterations         : 361826
flow value                 : 8570155949
checksum                   : 258659426
optimal


sim: ** simulation statistics **
sim_num_insn            49073291786 # total number of instructions
executed
sim_num_refs            20768099818 # total number of loads and stores
executed
sim_elapsed_time               3669 # total simulation time in seconds
sim_inst_rate          13375113.5966 # simulation speed (in insts/sec)
ld_text_base             0x00400000 # program text (code) segment base
ld_text_size                 113136 # program text (code) size in
bytes
ld_data_base             0x10000000 # program initialized data segment
base
ld_data_size                  19060 # program init'ed `.data' and
uninit'ed `.bss' size in bytes
ld_stack_base            0x7fffc000 # program stack segment base
(highest address in stack)
ld_stack_size                 16384 # program initial stack size
ld_prog_entry            0x00400140 # program entry point (initial PC)
```

```
ld_environ_base          0x7fff8000 # program environment base address
address
ld_target_big_endian             0 # target executable endian-ness,
non-zero if big endian
mem.page_count               24435 # total number of pages allocated
mem.page_mem                 97740k # total size of memory pages
allocated
mem.ptab_misses            10414410 # total first level page table
misses
mem.ptab_accesses      237836123981 # total page table accesses
mem.ptab_miss_rate           0.0000 # first level page table miss rate
```