benchmark with sim-outorder

分析调研报告

2014211304 班

D 组

史文翰 2014211218

林宇辰 2014211223

王剑督 2014211228

崔嘉伟 2014211233

杨　莹 2014211238

徐丹雅 2014211243

郝绍明 2014210123

分工情况

史文翰 2014211218：负责 Linux 操作，解决缺少函数库的问题，负责实验报告的编写和整理

林宇辰 2014211223 **王剑督** 2014211228 **崔嘉伟** 2014211233

杨　莹 2014211238 **徐丹雅** 2014211243 **郝绍明** 2014210123：负责查阅资料等辅助性工作

一、 用 sim-outorder 运行 tests

1、test-math

```
sim: ** starting performance simulation **
pow(12.0, 2.0) == 144.000000
pow(10.0, 3.0) == 1000.000000
pow(10.0, -3.0) == 0.001000
str: 123.456
x: 123.000000
str: 123.456
x: 123.456000
str: 123.456
x: 123.456000
123.456 123.456000 123 1000
sinh(2.0) = 3.62686
sinh(3.0) = 10.01787
h=3.60555
atan2(3,2) = 0.98279
pow(3.60555,4.0) = 169
169 / exp(0.98279 * 5) = 1.24102
3.93117 + 5*log(3.60555) = 10.34355
cos(10.34355) = -0.6068,    sin(10.34355) = -0.79486
x         0.5x
x0.5         x
x     0.5x
-1e-17 != -1e-17 Worked!

sim: ** simulation statistics **
sim_num_insn                   213745 # total number of instructions committed
sim_num_refs                    56902 # total number of loads and stores committed
sim_num_loads                   34108 # total number of loads committed
sim_num_stores               22794.0000 # total number of stores committed
sim_num_branches                38605 # total number of branches committed
sim_elapsed_time                    1 # total simulation time in seconds
sim_inst_rate               213745.0000 # simulation speed (in insts/sec)
sim_total_insn                 233115 # total number of instructions executed
sim_total_refs                  61930 # total number of loads and stores executed
sim_total_loads                 37548 # total number of loads executed
sim_total_stores             24382.0000 # total number of stores executed
sim_total_branches              42799 # total number of branches executed
sim_cycle                      224349 # total simulation time in cycles
```

```
sim_IPC                      0.9527 # instructions per cycle
sim_CPI                      1.0496 # cycles per instruction
sim_exec_BW                  1.0391 # total instructions (mis-spec + committed)
per cycle
sim_IPB                      5.5367 # instruction per branch
IFQ_count                   352193 # cumulative IFQ occupancy
IFQ_fcount                   74011 # cumulative IFQ full count
ifq_occupancy                1.5698 # avg IFQ occupancy (insn's)
ifq_rate                     1.0391 # avg IFQ dispatch rate (insn/cycle)
ifq_latency                  1.5108 # avg IFQ occupant latency (cycle's)
ifq_full                     0.3299 # fraction of time (cycle's) IFQ was full
RUU_count                  1440302 # cumulative RUU occupancy
RUU_fcount                   45178 # cumulative RUU full count
ruu_occupancy                6.4199 # avg RUU occupancy (insn's)
ruu_rate                     1.0391 # avg RUU dispatch rate (insn/cycle)
ruu_latency                  6.1785 # avg RUU occupant latency (cycle's)
ruu_full                     0.2014 # fraction of time (cycle's) RUU was full
LSQ_count                  315260 # cumulative LSQ occupancy
LSQ_fcount                   5107 # cumulative LSQ full count
lsq_occupancy                1.4052 # avg LSQ occupancy (insn's)
lsq_rate                     1.0391 # avg LSQ dispatch rate (insn/cycle)
lsq_latency                  1.3524 # avg LSQ occupant latency (cycle's)
lsq_full                     0.0228 # fraction of time (cycle's) LSQ was full
sim_slip                   1965181 # total number of slip cycles
avg_sim_slip                 9.1940 # the average slip between issue and retirement
bpred_bimod.lookups          44203 # total number of bpred lookups
bpred_bimod.updates          38605 # total number of updates
bpred_bimod.addr_hits        34569 # total number of address-predicted hits
bpred_bimod.dir_hits         35160 # total number of direction-predicted hits
(includes addr-hits)
bpred_bimod.misses            3445 # total number of misses
bpred_bimod.jr_hits           3424 # total number of address-predicted hits for JR's
bpred_bimod.jr_seen           3544 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP          28 # total number of address-predicted hits
for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP          41 # total number of non-RAS JR's seen
bpred_bimod.bpred_addr_rate  0.8955 # branch address-prediction rate (i.e., addr-
hits/updates)
bpred_bimod.bpred_dir_rate   0.9108 # branch direction-prediction rate (i.e., all-
hits/updates)
bpred_bimod.bpred_jr_rate    0.9661 # JR address-prediction rate (i.e., JR addr-
hits/JRs seen)
bpred_bimod.bpred_jr_non_ras_rate.PP  0.6829 # non-RAS JR addr-pred rate (ie,
non-RAS JR hits/JRs seen)
```

```
bpred_bimod.retstack_pushes              3909 # total number of address pushed onto
ret-addr stack
bpred_bimod.retstack_pops                4536 # total number of address popped off of ret-
addr stack
bpred_bimod.used_ras.PP                  3503 # total number of RAS predictions used
bpred_bimod.ras_hits.PP                  3396 # total number of RAS hits
bpred_bimod.ras_rate.PP                0.9695 # RAS prediction rate (i.e., RAS hits/used RAS)
il1.accesses                           256050 # total number of accesses
il1.hits                               239944 # total number of hits
il1.misses                              16106 # total number of misses
il1.replacements                        15595 # total number of replacements
il1.writebacks                              0 # total number of writebacks
il1.invalidations                           0 # total number of invalidations
il1.miss_rate                          0.0629 # miss rate (i.e., misses/ref)
il1.repl_rate                          0.0609 # 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                            56904 # total number of accesses
dl1.hits                                56353 # total number of hits
dl1.misses                                551 # total number of misses
dl1.replacements                           66 # total number of replacements
dl1.writebacks                             61 # total number of writebacks
dl1.invalidations                           0 # total number of invalidations
dl1.miss_rate                          0.0097 # miss rate (i.e., misses/ref)
dl1.repl_rate                          0.0012 # replacement rate (i.e., repls/ref)
dl1.wb_rate                            0.0011 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate                           0.0000 # invalidation rate (i.e., invs/ref)
ul2.accesses                            16718 # total number of accesses
ul2.hits                                15497 # total number of hits
ul2.misses                               1221 # total number of misses
ul2.replacements                            0 # total number of replacements
ul2.writebacks                              0 # total number of writebacks
ul2.invalidations                           0 # total number of invalidations
ul2.miss_rate                          0.0730 # miss rate (i.e., misses/ref)
ul2.repl_rate                          0.0000 # replacement rate (i.e., repls/ref)
ul2.wb_rate                            0.0000 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate                           0.0000 # invalidation rate (i.e., invs/ref)
itlb.accesses                          256050 # total number of accesses
itlb.hits                              256027 # total number of hits
itlb.misses                                23 # 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                         0.0001 # 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 57590 # total number of accesses
dtlb.hits 57580 # total number of hits
dtlb.misses 10 # total number of misses
dtlb.replacements 0 # total number of replacements
dtlb.writebacks 0 # total number of writebacks
dtlb.invalidations 0 # total number of invalidations
dtlb.miss_rate 0.0002 # miss rate (i.e., misses/ref)
dtlb.repl_rate 0.0000 # 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 91744 # program text (code) size in bytes
ld_data_base 0x10000000 # program initialized data segment base
ld_data_size 13028 # 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 33 # total number of pages allocated
mem.page_mem 132k # total size of memory pages allocated
mem.ptab_misses 40 # total first level page table misses
mem.ptab_accesses 2021633 # total page table accesses
mem.ptab_miss_rate 0.0000 # first level page table miss rate

2、test-fmath

sim: ** starting performance simulation **
q=4 (int)x=12 (int)y=29
z=144
z=841
z=13
z=13
l=6
l=36
*lp=216
z=144.000000

q=4 x=12.000000 (int)x=12 y=29.000000 (int)y=29
q = 16 x = 11.700001 y = 23.400000

sim: ** simulation statistics **
sim_num_insn 53504 # total number of instructions committed
sim_num_refs 16347 # total number of loads and stores committed
sim_num_loads 8635 # total number of loads committed
sim_num_stores 7712.0000 # total number of stores committed
sim_num_branches 10353 # total number of branches committed
sim_elapsed_time 1 # total simulation time in seconds
sim_inst_rate 53504.0000 # simulation speed (in insts/sec)
sim_total_insn 58079 # total number of instructions executed
sim_total_refs 17481 # total number of loads and stores executed
sim_total_loads 9398 # total number of loads executed
sim_total_stores 8083.0000 # total number of stores executed
sim_total_branches 11298 # total number of branches executed
sim_cycle 68367 # total simulation time in cycles
sim_IPC 0.7826 # instructions per cycle
sim_CPI 1.2778 # cycles per instruction
sim_exec_BW 0.8495 # total instructions (mis-spec + committed) per cycle
sim_IPB 5.1680 # instruction per branch
IFQ_count 89586 # cumulative IFQ occupancy
IFQ_fcount 18917 # cumulative IFQ full count
ifq_occupancy 1.3104 # avg IFQ occupancy (insn's)
ifq_rate 0.8495 # avg IFQ dispatch rate (insn/cycle)
ifq_latency 1.5425 # avg IFQ occupant latency (cycle's)
ifq_full 0.2767 # fraction of time (cycle's) IFQ was full
RUU_count 359514 # cumulative RUU occupancy
RUU_fcount 10558 # cumulative RUU full count
ruu_occupancy 5.2586 # avg RUU occupancy (insn's)
ruu_rate 0.8495 # avg RUU dispatch rate (insn/cycle)
ruu_latency 6.1901 # avg RUU occupant latency (cycle's)
ruu_full 0.1544 # fraction of time (cycle's) RUU was full
LSQ_count 85631 # cumulative LSQ occupancy
LSQ_fcount 2148 # cumulative LSQ full count
lsq_occupancy 1.2525 # avg LSQ occupancy (insn's)
lsq_rate 0.8495 # avg LSQ dispatch rate (insn/cycle)
lsq_latency 1.4744 # avg LSQ occupant latency (cycle's)
lsq_full 0.0314 # fraction of time (cycle's) LSQ was full
sim_slip 500917 # total number of slip cycles
avg_sim_slip 9.3622 # the average slip between issue and retirement
bpred_bimod.lookups 11563 # total number of bpred lookups
bpred_bimod.updates 10353 # total number of updates

```
bpred_bimod.addr_hits            9123 # total number of address-predicted hits
bpred_bimod.dir_hits             9434 # total number of direction-predicted hits
(includes addr-hits)
bpred_bimod.misses                919 # total number of misses
bpred_bimod.jr_hits               792 # total number of address-predicted hits for JR's
bpred_bimod.jr_seen               833 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP      4 # total number of address-predicted hits
for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP     19 # total number of non-RAS JR's seen
bpred_bimod.bpred_addr_rate    0.8812 # branch address-prediction rate (i.e., addr-
hits/updates)
bpred_bimod.bpred_dir_rate     0.9112 # branch direction-prediction rate (i.e., all-
hits/updates)
bpred_bimod.bpred_jr_rate      0.9508 # JR address-prediction rate (i.e., JR addr-
hits/JRs seen)
bpred_bimod.bpred_jr_non_ras_rate.PP    0.2105 # non-RAS JR addr-pred rate (ie,
non-RAS JR hits/JRs seen)
bpred_bimod.retstack_pushes       942 # total number of address pushed onto
ret-addr stack
bpred_bimod.retstack_pops         930 # total number of address popped off of ret-
addr stack
bpred_bimod.used_ras.PP           814 # total number of RAS predictions used
bpred_bimod.ras_hits.PP           788 # total number of RAS hits
bpred_bimod.ras_rate.PP     0.9681 # RAS prediction rate (i.e., RAS hits/used RAS)
il1.accesses                  63612 # total number of accesses
il1.hits                      59269 # total number of hits
il1.misses                     4343 # total number of misses
il1.replacements               3847 # total number of replacements
il1.writebacks                    0 # total number of writebacks
il1.invalidations                 0 # total number of invalidations
il1.miss_rate                0.0683 # miss rate (i.e., misses/ref)
il1.repl_rate                0.0605 # 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                  16431 # total number of accesses
dl1.hits                      15955 # total number of hits
dl1.misses                      476 # total number of misses
dl1.replacements                 29 # total number of replacements
dl1.writebacks                   22 # total number of writebacks
dl1.invalidations                 0 # total number of invalidations
dl1.miss_rate                0.0290 # miss rate (i.e., misses/ref)
dl1.repl_rate                0.0018 # replacement rate (i.e., repls/ref)
dl1.wb_rate                  0.0013 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate                 0.0000 # invalidation rate (i.e., invs/ref)
```

```
ul2.accesses                 4841 # total number of accesses
ul2.hits                     3975 # total number of hits
ul2.misses                    866 # total number of misses
ul2.replacements                0 # total number of replacements
ul2.writebacks                  0 # total number of writebacks
ul2.invalidations               0 # total number of invalidations
ul2.miss_rate              0.1789 # miss rate (i.e., misses/ref)
ul2.repl_rate              0.0000 # replacement rate (i.e., repls/ref)
ul2.wb_rate                0.0000 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate               0.0000 # invalidation rate (i.e., invs/ref)
itlb.accesses                63612 # total number of accesses
itlb.hits                    63592 # total number of hits
itlb.misses                     20 # 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             0.0003 # 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                16595 # total number of accesses
dtlb.hits                    16585 # total number of hits
dtlb.misses                     10 # total number of misses
dtlb.replacements               0 # total number of replacements
dtlb.writebacks                 0 # total number of writebacks
dtlb.invalidations              0 # total number of invalidations
dtlb.miss_rate             0.0006 # miss rate (i.e., misses/ref)
dtlb.repl_rate             0.0000 # 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                79920 # program text (code) size in bytes
ld_data_base           0x10000000 # program initialized data segment base
ld_data_size                12288 # 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 30 # total number of pages allocated
mem.page_mem 120k # total size of memory pages allocated
mem.ptab_misses 34 # total first level page table misses
mem.ptab_accesses 879930 # total page table accesses
mem.ptab_miss_rate 0.0000 # first level page table miss rate

3、 test-printf

sim: ** starting performance simulation **
%.4x:`0012'
%04x: `0012'
%4.4x: `0012'
%04.4x: `0012'
%4.3x: ` 012'
%04.3x: `0012'
%.*x: `0012'
%0*x: `0012'
%*.*x: `0012'
%0*.*x: `0012'
bad format: "%z"
nil pointer (padded): " (nil)"
decimal negative: "-2345"
octal negative: "37777773327"
hex negative: "fffff6d7"
long decimal number: "-123456"
long octal negative: "37777773327"
long unsigned decimal number: "4294843840"
zero-padded LDN: "-000123456"
left-adjusted ZLDN: "-123456 "
space-padded LDN: " -123456"
left-adjusted SLDN: "-123456 "
zero-padded string: " Hi, Z."
left-adjusted Z string: "Hi, Z. "
space-padded string: " Hi, Z."
left-adjusted S string: "Hi, Z. "
null string: "(null)"
e-style >= 1: "1.234000e+01"
e-style >= .1: "1.234000e-01"
e-style < .1: "1.234000e-03"
e-style big: "1.000000000000000000000000000000000000000000000000000000000e+20"
e-style == .1: "1.000000e-01"
f-style >= 1: "12.340000"
f-style >= .1: "0.123400"
f-style < .1: "0.001234"

```
g-style >= 1: "12.34"
g-style >= .1:"0.1234"
g-style < .1:   "0.00123"
g-style big:    "10000000000000000000"
:new test: 99.85:
  0.10000
  0.10000
x0.5000x
0x1
|          0.0000|    0.0000e+00|                   0|
|          1.0000|    1.0000e+00|                   1|
|         -1.0000| -1.0000e+00|                  -1|
|        100.0000|    1.0000e+02|                 100|
|     1000.0000|    1.0000e+03|                1000|
|    10000.0000|    1.0000e+04|               1e+04|
|    12345.0000|    1.2345e+04|     1.235e+04|
| 100000.0000|    1.0000e+05|              1e+05|
| 123456.0000|    1.2346e+05|     1.235e+05|

Formatted output test
prefix    6d           6o           6x           6X           6u
%-+#0 |-123     |0377     |0xff     |0XFF     |4294967295 |
  %-+# |-123     |0377     |0xff     |0XFF     |4294967295 |
  %-+0 |-123     |377      |ff       |FF       |4294967295 |
    %-+ |-123     |377      |ff       |FF       |4294967295 |
  %-#0 |-123     |0377     |0xff     |0XFF     |4294967295 |
    %-# |-123     |0377     |0xff     |0XFF     |4294967295 |
    %-0 |-123     |377      |ff       |FF       |4294967295 |
     %- |-123     |377      |ff       |FF       |4294967295 |
  %+#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
    %+# |    -123 |    0377 |    0xff |    0XFF |4294967295 |
    %+0 |-00123 |000377 |0000ff |0000FF |4294967295 |
     %+ |    -123 |     377 |       ff |       FF |4294967295 |
    %#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
     %# |    -123 |    0377 |    0xff |    0XFF |4294967295 |
     %0 |-00123 |000377 |0000ff |0000FF |4294967295 |
      % |    -123 |     377 |       ff |       FF |4294967295 |
       (null)
(null)

Formatted output test
prefix    6d           6o           6x           6X           6u
%-+#0 |-123     |0377     |0xff     |0XFF     |4294967295 |
  %-+# |-123     |0377     |0xff     |0XFF     |4294967295 |
```

```
  %-+0 |-123      |377       |ff         |FF         |4294967295 |
   %-+ |-123      |377       |ff         |FF         |4294967295 |
  %-#0 |-123      |0377      |0xff       |0XFF       |4294967295 |
   %-# |-123      |0377      |0xff       |0XFF       |4294967295 |
   %-0 |-123      |377       |ff         |FF         |4294967295 |
    %- |-123      |377       |ff         |FF         |4294967295 |
  %+#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
   %+# |    -123 |    0377 |    0xff |    0XFF |4294967295 |
   %+0 |-00123 |000377 |0000ff |0000FF |4294967295 |
    %+ |    -123 |      377 |        ff |        FF |4294967295 |
   %#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
    %# |    -123 |    0377 |    0xff |    0XFF |4294967295 |
    %0 |-00123 |000377 |0000ff |0000FF |4294967295 |
     % |    -123 |      377 |        ff |        FF |4294967295 |
     (null)
(null)

Formatted output test
prefix    6d          6o          6x          6X          6u
%-+#0 |-123      |0377      |0xff      |0XFF      |4294967295 |
  %-+# |-123      |0377      |0xff      |0XFF      |4294967295 |
  %-+0 |-123      |377       |ff         |FF         |4294967295 |
   %-+ |-123      |377       |ff         |FF         |4294967295 |
  %-#0 |-123      |0377      |0xff       |0XFF       |4294967295 |
   %-# |-123      |0377      |0xff       |0XFF       |4294967295 |
   %-0 |-123      |377       |ff         |FF         |4294967295 |
    %- |-123      |377       |ff         |FF         |4294967295 |
  %+#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
   %+# |    -123 |    0377 |    0xff |    0XFF |4294967295 |
   %+0 |-00123 |000377 |0000ff |0000FF |4294967295 |
    %+ |    -123 |      377 |        ff |        FF |4294967295 |
   %#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
    %# |    -123 |    0377 |    0xff |    0XFF |4294967295 |
    %0 |-00123 |000377 |0000ff |0000FF |4294967295 |
     % |    -123 |      377 |        ff |        FF |4294967295 |
     (null)
(null)

Formatted output test
prefix    6d          6o          6x          6X          6u
%-+#0 |-123      |0377      |0xff      |0XFF      |4294967295 |
  %-+# |-123      |0377      |0xff      |0XFF      |4294967295 |
  %-+0 |-123      |377       |ff         |FF         |4294967295 |
   %-+ |-123      |377       |ff         |FF         |4294967295 |
```

```
  %-#0 |-123     |0377     |0xff     |0XFF     |4294967295 |
   %-# |-123     |0377     |0xff     |0XFF     |4294967295 |
   %-0 |-123     |377      |ff       |FF       |4294967295 |
    %- |-123     |377      |ff       |FF       |4294967295 |
  %+#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
   %+# |   -123 |   0377 |   0xff |   0XFF |4294967295 |
   %+0 |-00123 |000377 |0000ff |0000FF |4294967295 |
    %+ |   -123 |    377 |     ff |     FF |4294967295 |
   %#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
    %# |   -123 |   0377 |   0xff |   0XFF |4294967295 |
    %0 |-00123 |000377 |0000ff |0000FF |4294967295 |
     % |   -123 |    377 |     ff |     FF |4294967295 |
     (null)
(null)

Formatted output test
prefix    6d          6o          6x          6X          6u
%-+#0 |-123     |0377     |0xff     |0XFF     |4294967295 |
  %-+# |-123     |0377     |0xff     |0XFF     |4294967295 |
  %-+0 |-123     |377      |ff       |FF       |4294967295 |
   %-+ |-123     |377      |ff       |FF       |4294967295 |
  %-#0 |-123     |0377     |0xff     |0XFF     |4294967295 |
   %-# |-123     |0377     |0xff     |0XFF     |4294967295 |
   %-0 |-123     |377      |ff       |FF       |4294967295 |
    %- |-123     |377      |ff       |FF       |4294967295 |
  %+#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
   %+# |   -123 |   0377 |   0xff |   0XFF |4294967295 |
   %+0 |-00123 |000377 |0000ff |0000FF |4294967295 |
    %+ |   -123 |    377 |     ff |     FF |4294967295 |
   %#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
    %# |   -123 |   0377 |   0xff |   0XFF |4294967295 |
    %0 |-00123 |000377 |0000ff |0000FF |4294967295 |
     % |   -123 |    377 |     ff |     FF |4294967295 |
     (null)
(null)

Formatted output test
prefix    6d          6o          6x          6X          6u
%-+#0 |-123     |0377     |0xff     |0XFF     |4294967295 |
  %-+# |-123     |0377     |0xff     |0XFF     |4294967295 |
  %-+0 |-123     |377      |ff       |FF       |4294967295 |
   %-+ |-123     |377      |ff       |FF       |4294967295 |
  %-#0 |-123     |0377     |0xff     |0XFF     |4294967295 |
   %-# |-123     |0377     |0xff     |0XFF     |4294967295 |
```

```
   %-0 |-123     |377       |ff        |FF        |4294967295 |
    %- |-123     |377       |ff        |FF        |4294967295 |
 %+#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
   %+# |   -123 |   0377 |   0xff |   0XFF |4294967295 |
   %+0 |-00123 |000377 |0000ff |0000FF |4294967295 |
    %+ |   -123 |     377 |       ff |       FF |4294967295 |
   %#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
    %# |   -123 |   0377 |   0xff |   0XFF |4294967295 |
    %0 |-00123 |000377 |0000ff |0000FF |4294967295 |
     % |   -123 |     377 |       ff |       FF |4294967295 |
     (null)
(null)

Formatted output test
prefix   6d          6o          6x          6X          6u
%-+#0 |-123     |0377     |0xff     |0XFF     |4294967295 |
 %-+# |-123     |0377     |0xff     |0XFF     |4294967295 |
 %-+0 |-123     |377       |ff        |FF        |4294967295 |
  %-+ |-123     |377       |ff        |FF        |4294967295 |
 %-#0 |-123     |0377     |0xff     |0XFF     |4294967295 |
  %-# |-123     |0377     |0xff     |0XFF     |4294967295 |
  %-0 |-123     |377       |ff        |FF        |4294967295 |
   %- |-123     |377       |ff        |FF        |4294967295 |
 %+#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
  %+# |   -123 |   0377 |   0xff |   0XFF |4294967295 |
  %+0 |-00123 |000377 |0000ff |0000FF |4294967295 |
   %+ |   -123 |     377 |       ff |       FF |4294967295 |
  %#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
   %# |   -123 |   0377 |   0xff |   0XFF |4294967295 |
   %0 |-00123 |000377 |0000ff |0000FF |4294967295 |
    % |   -123 |     377 |       ff |       FF |4294967295 |
    (null)
(null)

Formatted output test
prefix   6d          6o          6x          6X          6u
%-+#0 |-123     |0377     |0xff     |0XFF     |4294967295 |
 %-+# |-123     |0377     |0xff     |0XFF     |4294967295 |
 %-+0 |-123     |377       |ff        |FF        |4294967295 |
  %-+ |-123     |377       |ff        |FF        |4294967295 |
 %-#0 |-123     |0377     |0xff     |0XFF     |4294967295 |
  %-# |-123     |0377     |0xff     |0XFF     |4294967295 |
  %-0 |-123     |377       |ff        |FF        |4294967295 |
   %- |-123     |377       |ff        |FF        |4294967295 |
```

```
 %+#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
  %+# |    -123 |    0377 |    0xff |    0XFF |4294967295 |
  %+0 |-00123 |000377 |0000ff |0000FF |4294967295 |
   %+ |    -123 |     377 |       ff |       FF |4294967295 |
  %#0 |-00123 |000377 |0x00ff |0X00FF |4294967295 |
   %# |    -123 |    0377 |    0xff |    0XFF |4294967295 |
   %0 |-00123 |000377 |0000ff |0000FF |4294967295 |
    % |    -123 |     377 |       ff |       FF |4294967295 |
    (null)
(null)
1.234568e+06 should be 1.234568e+06
1234567.800000 should be 1234567.800000
1.23457e+06 should be 1.23457e+06
123.456 should be 123.456
1e+06 should be 1e+06
10 should be 10
0.02 should be 0.02
testing parametric fields: 0.7000

sim: ** simulation statistics **
sim_num_insn                 1813937 # total number of instructions committed
sim_num_refs                  516949 # total number of loads and stores committed
sim_num_loads                 348604 # total number of loads committed
sim_num_stores           168345.0000 # total number of stores committed
sim_num_branches              401630 # total number of branches committed
sim_elapsed_time                   2 # total simulation time in seconds
sim_inst_rate            906968.5000 # simulation speed (in insts/sec)
sim_total_insn               1984229 # total number of instructions executed
sim_total_refs                560618 # total number of loads and stores executed
sim_total_loads               379039 # total number of loads executed
sim_total_stores         181579.0000 # total number of stores executed
sim_total_branches            450155 # total number of branches executed
sim_cycle                     1395617 # total simulation time in cycles
sim_IPC                        1.2997 # instructions per cycle
sim_CPI                        0.7694 # cycles per instruction
sim_exec_BW                    1.4218 # total instructions (mis-spec + committed)
per cycle
sim_IPB                        4.5164 # instruction per branch
IFQ_count                     3424664 # cumulative IFQ occupancy
IFQ_fcount                     724198 # cumulative IFQ full count
ifq_occupancy                  2.4539 # avg IFQ occupancy (insn's)
ifq_rate                       1.4218 # avg IFQ dispatch rate (insn/cycle)
ifq_latency                    1.7259 # avg IFQ occupant latency (cycle's)
ifq_full                       0.5189 # fraction of time (cycle's) IFQ was full
```

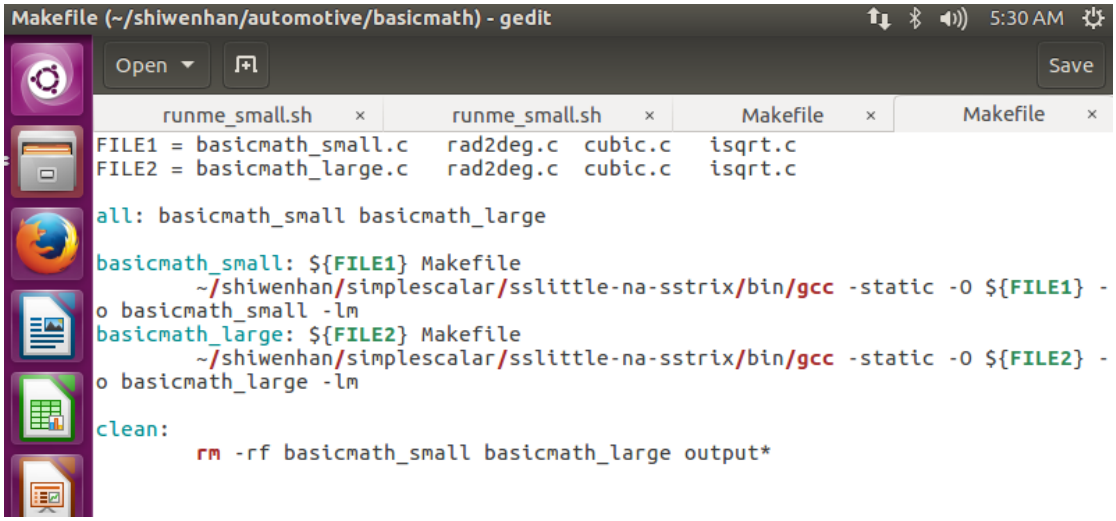
RUU_count 14061759 # cumulative RUU occupancy
RUU_fcount 558671 # cumulative RUU full count
ruu_occupancy 10.0757 # avg RUU occupancy (insn's)
ruu_rate 1.4218 # avg RUU dispatch rate (insn/cycle)
ruu_latency 7.0868 # avg RUU occupant latency (cycle's)
ruu_full 0.4003 # fraction of time (cycle's) RUU was full
LSQ_count 2777672 # cumulative LSQ occupancy
LSQ_fcount 37082 # cumulative LSQ full count
lsq_occupancy 1.9903 # avg LSQ occupancy (insn's)
lsq_rate 1.4218 # avg LSQ dispatch rate (insn/cycle)
lsq_latency 1.3999 # avg LSQ occupant latency (cycle's)
lsq_full 0.0266 # fraction of time (cycle's) LSQ was full
sim_slip 18575793 # total number of slip cycles
avg_sim_slip 10.2406 # the average slip between issue and retirement
bpred_bimod.lookups 464071 # total number of bpred lookups
bpred_bimod.updates 401630 # total number of updates
bpred_bimod.addr_hits 377941 # total number of address-predicted hits
bpred_bimod.dir_hits 380425 # total number of direction-predicted hits (includes addr-hits)
bpred_bimod.misses 21205 # total number of misses
bpred_bimod.jr_hits 29956 # total number of address-predicted hits for JR's
bpred_bimod.jr_seen 31999 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP 363 # total number of address-predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP 2198 # total number of non-RAS JR's seen
bpred_bimod.bpred_addr_rate 0.9410 # branch address-prediction rate (i.e., addr-hits/updates)
bpred_bimod.bpred_dir_rate 0.9472 # branch direction-prediction rate (i.e., all-hits/updates)
bpred_bimod.bpred_jr_rate 0.9362 # JR address-prediction rate (i.e., JR addr-hits/JRs seen)
bpred_bimod.bpred_jr_non_ras_rate.PP 0.1652 # non-RAS JR addr-pred rate (ie, non-RAS JR hits/JRs seen)
bpred_bimod.retstack_pushes 34915 # total number of address pushed onto ret-addr stack
bpred_bimod.retstack_pops 31690 # total number of address popped off of ret-addr stack
bpred_bimod.used_ras.PP 29801 # total number of RAS predictions used
bpred_bimod.ras_hits.PP 29593 # total number of RAS hits
bpred_bimod.ras_rate.PP 0.9930 # RAS prediction rate (i.e., RAS hits/used RAS)
il1.accesses 2116704 # total number of accesses
il1.hits 2054684 # total number of hits
il1.misses 62020 # total number of misses
il1.replacements 61510 # total number of replacements

```
il1.writebacks                    0 # total number of writebacks
il1.invalidations                 0 # total number of invalidations
il1.miss_rate                0.0293 # miss rate (i.e., misses/ref)
il1.repl_rate                0.0291 # 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                524334 # total number of accesses
dl1.hits                    523762 # total number of hits
dl1.misses                     572 # total number of misses
dl1.replacements                77 # total number of replacements
dl1.writebacks                  73 # total number of writebacks
dl1.invalidations                0 # total number of invalidations
dl1.miss_rate                0.0011 # miss rate (i.e., misses/ref)
dl1.repl_rate                0.0001 # replacement rate (i.e., repls/ref)
dl1.wb_rate                  0.0001 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate                 0.0000 # invalidation rate (i.e., invs/ref)
ul2.accesses                 62665 # total number of accesses
ul2.hits                     61524 # total number of hits
ul2.misses                    1141 # total number of misses
ul2.replacements                 0 # total number of replacements
ul2.writebacks                   0 # total number of writebacks
ul2.invalidations                0 # total number of invalidations
ul2.miss_rate                0.0182 # miss rate (i.e., misses/ref)
ul2.repl_rate                0.0000 # replacement rate (i.e., repls/ref)
ul2.wb_rate                  0.0000 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate                 0.0000 # invalidation rate (i.e., invs/ref)
itlb.accesses               2116704 # total number of accesses
itlb.hits                   2116685 # total number of hits
itlb.misses                      19 # 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                528808 # total number of accesses
dtlb.hits                    528798 # total number of hits
dtlb.misses                      10 # total number of misses
dtlb.replacements                0 # total number of replacements
dtlb.writebacks                  0 # total number of writebacks
dtlb.invalidations               0 # total number of invalidations
dtlb.miss_rate               0.0000 # miss rate (i.e., misses/ref)
dtlb.repl_rate               0.0000 # 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 74640 # program text (code) size in bytes
ld_data_base 0x10000000 # program initialized data segment base
ld_data_size 13636 # 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 29 # total number of pages allocated
mem.page_mem 116k # total size of memory pages allocated
mem.ptab_misses 67 # total first level page table misses
mem.ptab_accesses 12667224 # total page table accesses
mem.ptab_miss_rate 0.0000 # first level page table miss rate

二、 构建 automotive 包

下载并解压 automotive.tag.gz
修改 makefile 文件如下：



进行在 GCC 交叉编译器下 make：

```
shiwenhan@ubuntu:~/shiwenhan/automotive/basicmath$ make
~/shiwenhan/simplescalar/sslittle-na-sstrix/bin/gcc -static -O basicmath_small.c
   rad2deg.c  cubic.c   isqrt.c -o basicmath_small -lm
/tmp/ccfsccot.s: Assembler messages:
/tmp/ccfsccot.s:56: Warning: Bignum truncated to 4 bytes
/tmp/ccfsccot.s:60: Warning: Bignum truncated to 4 bytes
/tmp/ccfsccot.s:64: Warning: Bignum truncated to 4 bytes
/tmp/ccfsccot.s: Assembler messages:
/tmp/ccfsccot.s:17: Warning: Bignum truncated to 4 bytes
/tmp/ccfsccot.s:25: Warning: Bignum truncated to 4 bytes
/tmp/ccfsccot.s: Assembler messages:
/tmp/ccfsccot.s:17: Warning: Bignum truncated to 4 bytes
/tmp/ccfsccot.s:21: Warning: Bignum truncated to 4 bytes
/home/shiwenhan/shiwenhan/simplescalar//sslittle-na-sstrix/bin/ld: cannot find -
lm
Makefile:7: recipe for target 'basicmath_small' failed
make: *** [basicmath_small] Error 1
```

发生了-lm 找不到的情况，上网搜索可知，这是交叉编译器缺少 math 库造成的。math 库是 libm.a，可以在任何一个 gcc 中找到，之后利用产品命令将 libc 复制到交叉编译器的 lib 中。

cp libm.a /home/shiwenhan/shiwenhan/simplescalar/sslittle-na-sstrix/lib

```
shiwenhan@ubuntu:~/shiwenhan$ su
Password:
root@ubuntu:/home/shiwenhan/shiwenhan# ^C
root@ubuntu:/home/shiwenhan/shiwenhan# cp libm.a ~/shiwenhan/simplescalar/sslitt
le-na-sstrix/lib
cp: cannot create regular file '/root/shiwenhan/simplescalar/sslittle-na-sstrix/
lib': No such file or directory
root@ubuntu:/home/shiwenhan/shiwenhan# cp libm.a /home/shiwenhan/shiwenhan/simpl
escalar/sslittle-na-sstrix/lib
root@ubuntu:/home/shiwenhan/shiwenhan#
```

之后即可成功 make：

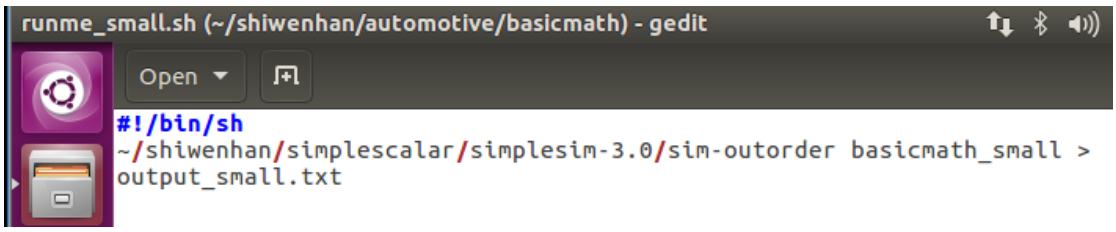
```
shiwenhan@ubuntu:~/shiwenhan/automotive/basicmath$ make
~/shiwenhan/simplescalar/sslittle-na-sstrix/bin/gcc -static -O basicmath_small.c
   rad2deg.c  cubic.c   isqrt.c -o basicmath_small -lm
/tmp/ccmFDqN6.s: Assembler messages:
/tmp/ccmFDqN6.s:56: Warning: Bignum truncated to 4 bytes
/tmp/ccmFDqN6.s:60: Warning: Bignum truncated to 4 bytes
/tmp/ccmFDqN6.s:64: Warning: Bignum truncated to 4 bytes
/tmp/ccmFDqN6.s: Assembler messages:
/tmp/ccmFDqN6.s:17: Warning: Bignum truncated to 4 bytes
/tmp/ccmFDqN6.s:25: Warning: Bignum truncated to 4 bytes
 LibreOffice Calc   Assembler messages:
/tmp/ccmFDqN6.s:17: Warning: Bignum truncated to 4 bytes
/tmp/ccmFDqN6.s:21: Warning: Bignum truncated to 4 bytes
~/shiwenhan/simplescalar/sslittle-na-sstrix/bin/gcc -static -O basicmath_large.c
   rad2deg.c  cubic.c   isqrt.c -o basicmath_large -lm
/tmp/ccgigVIz.s: Assembler messages:
/tmp/ccgigVIz.s:55: Warning: Bignum truncated to 4 bytes
/tmp/ccgigVIz.s:59: Warning: Bignum truncated to 4 bytes
/tmp/ccgigVIz.s:63: Warning: Bignum truncated to 4 bytes
/tmp/ccgigVIz.s:67: Warning: Bignum truncated to 4 bytes
/tmp/ccgigVIz.s:71: Warning: Bignum truncated to 4 bytes
```

生成了如下两个.dat 文件：

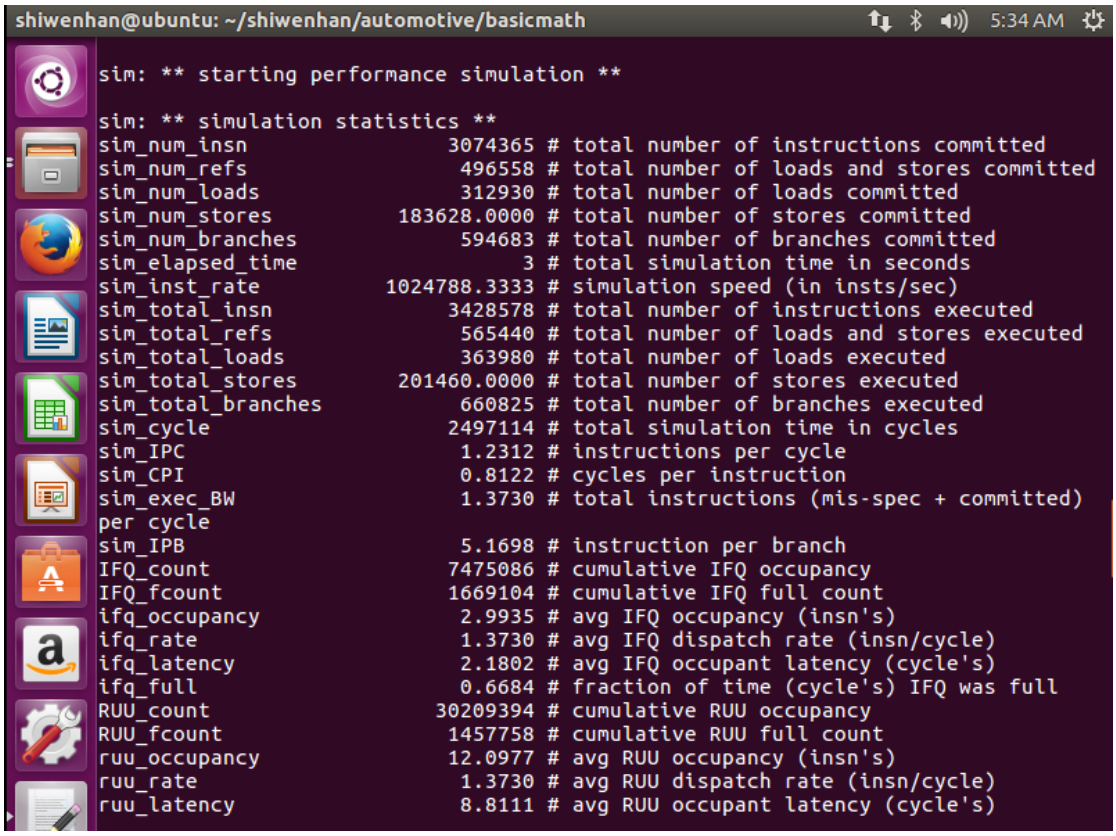
Name	Size	Type	Modified
basicmath_large	247.0 kB	Program	04:36
basicmath_small	247.0 kB	Program	04:36

三、 测试 benchmark

修改 shell 程序如下：



这里仍然以 basicmath_small 为例，执行相应的 shell 脚本 rumme_small.sh：



具体内容为：

sim: ** starting performance simulation **

sim: ** simulation statistics **
sim_num_insn 3074365 # total number of instructions committed
sim_num_refs 496558 # total number of loads and stores committed
sim_num_loads 312930 # total number of loads committed
sim_num_stores 183628.0000 # total number of stores committed
sim_num_branches 594683 # total number of branches committed
sim_elapsed_time 3 # total simulation time in seconds
sim_inst_rate 1024788.3333 # simulation speed (in insts/sec)
sim_total_insn 3428578 # total number of instructions executed
sim_total_refs 565440 # total number of loads and stores executed
sim_total_loads 363980 # total number of loads executed
sim_total_stores 201460.0000 # total number of stores executed

sim_total_branches 660825 # total number of branches executed
sim_cycle 2497114 # total simulation time in cycles
sim_IPC 1.2312 # instructions per cycle
sim_CPI 0.8122 # cycles per instruction
sim_exec_BW 1.3730 # total instructions (mis-spec + committed) per cycle
sim_IPB 5.1698 # instruction per branch
IFQ_count 7475086 # cumulative IFQ occupancy
IFQ_fcount 1669104 # cumulative IFQ full count
ifq_occupancy 2.9935 # avg IFQ occupancy (insn's)
ifq_rate 1.3730 # avg IFQ dispatch rate (insn/cycle)
ifq_latency 2.1802 # avg IFQ occupant latency (cycle's)
ifq_full 0.6684 # fraction of time (cycle's) IFQ was full
RUU_count 30209394 # cumulative RUU occupancy
RUU_fcount 1457758 # cumulative RUU full count
ruu_occupancy 12.0977 # avg RUU occupancy (insn's)
ruu_rate 1.3730 # avg RUU dispatch rate (insn/cycle)
ruu_latency 8.8111 # avg RUU occupant latency (cycle's)
ruu_full 0.5838 # fraction of time (cycle's) RUU was full
LSQ_count 2674964 # cumulative LSQ occupancy
LSQ_fcount 14199 # cumulative LSQ full count
lsq_occupancy 1.0712 # avg LSQ occupancy (insn's)
lsq_rate 1.3730 # avg LSQ dispatch rate (insn/cycle)
lsq_latency 0.7802 # avg LSQ occupant latency (cycle's)
lsq_full 0.0057 # fraction of time (cycle's) LSQ was full
sim_slip 35350244 # total number of slip cycles
avg_sim_slip 11.4984 # the average slip between issue and retirement
bpred_bimod.lookups 680081 # total number of bpred lookups
bpred_bimod.updates 594683 # total number of updates
bpred_bimod.addr_hits 552232 # total number of address-predicted hits
bpred_bimod.dir_hits 552475 # total number of direction-predicted hits (includes addr-hits)
bpred_bimod.misses 42208 # total number of misses
bpred_bimod.jr_hits 27583 # total number of address-predicted hits for JR's
bpred_bimod.jr_seen 29601 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP 5 # total number of address-predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP 2014 # total number of non-RAS JR's seen
bpred_bimod.bpred_addr_rate 0.9286 # branch address-prediction rate (i.e., addr-hits/updates)
bpred_bimod.bpred_dir_rate 0.9290 # branch direction-prediction rate (i.e., all-hits/updates)
bpred_bimod.bpred_jr_rate 0.9318 # JR address-prediction rate (i.e., JR addr-hits/JRs seen)

bpred_bimod.bpred_jr_non_ras_rate.PP 0.0025 # non-RAS JR addr-pred rate (ie, non-RAS JR hits/JRs seen)
bpred_bimod.retstack_pushes 30658 # total number of address pushed onto ret-addr stack
bpred_bimod.retstack_pops 30614 # total number of address popped off of ret-addr stack
bpred_bimod.used_ras.PP 27587 # total number of RAS predictions used
bpred_bimod.ras_hits.PP 27578 # total number of RAS hits
bpred_bimod.ras_rate.PP 0.9997 # RAS prediction rate (i.e., RAS hits/used RAS)
il1.accesses 3607658 # total number of accesses
il1.hits 3543970 # total number of hits
il1.misses 63688 # total number of misses
il1.replacements 63244 # total number of replacements
il1.writebacks 0 # total number of writebacks
il1.invalidations 0 # total number of invalidations
il1.miss_rate 0.0177 # miss rate (i.e., misses/ref)
il1.repl_rate 0.0175 # 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 494096 # total number of accesses
dl1.hits 493493 # total number of hits
dl1.misses 603 # total number of misses
dl1.replacements 91 # total number of replacements
dl1.writebacks 88 # total number of writebacks
dl1.invalidations 0 # total number of invalidations
dl1.miss_rate 0.0012 # miss rate (i.e., misses/ref)
dl1.repl_rate 0.0002 # replacement rate (i.e., repls/ref)
dl1.wb_rate 0.0002 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate 0.0000 # invalidation rate (i.e., invs/ref)
ul2.accesses 64379 # total number of accesses
ul2.hits 63584 # total number of hits
ul2.misses 795 # total number of misses
ul2.replacements 0 # total number of replacements
ul2.writebacks 0 # total number of writebacks
ul2.invalidations 0 # total number of invalidations
ul2.miss_rate 0.0123 # miss rate (i.e., misses/ref)
ul2.repl_rate 0.0000 # replacement rate (i.e., repls/ref)
ul2.wb_rate 0.0000 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate 0.0000 # invalidation rate (i.e., invs/ref)
itlb.accesses 3607658 # total number of accesses
itlb.hits 3607639 # total number of hits
itlb.misses 19 # 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 517999 # total number of accesses
dtlb.hits 517989 # total number of hits
dtlb.misses 10 # total number of misses
dtlb.replacements 0 # total number of replacements
dtlb.writebacks 0 # total number of writebacks
dtlb.invalidations 0 # total number of invalidations
dtlb.miss_rate 0.0000 # miss rate (i.e., misses/ref)
dtlb.repl_rate 0.0000 # 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 90640 # program text (code) size in bytes
ld_data_base 0x10000000 # program initialized data segment base
ld_data_size 12608 # 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 33 # total number of pages allocated
mem.page_mem 132k # total size of memory pages allocated
mem.ptab_misses 53 # total first level page table misses
mem.ptab_accesses 18289774 # total page table accesses
mem.ptab_miss_rate 0.0000 # first level page table miss rate

可得到各种 sim-outorder 的统计变量，具体对应的变量解释请对照实验四，在此不做赘述。