0.Preparation part

```
ceciliazhang@ceciliazhang-VirtualBox:~$ cd /home/ceciliazhang/simplesim/simplesim-3.0
ceciliazhang@ceciliazhang-VirtualBox:~/simplesim/simplesim-3.0$ sudo apt install make
[sudo] password for ceciliazhang:
Reading package lists... Done
Building dependency tree
Reading state information... Done
make is already the newest version (4.1-9.1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 270 not upgraded.
ceciliazhang@ceciliazhang-VirtualBox:~/simplesim/simplesim-3.0$ sudo apt install gcc
Reading package lists... Done
Building dependency tree
Reading state information... Done
gcc is already the newest version (4:7.3.0-3ubuntu2.1).
0 upgraded, 0 newly installed, 0 to remove and 270 not upgraded.
ceciliazhang@ceciliazhang-VirtualBox:~/simplesim/simplesim-3.0$ make config-pisa
rm -f config.h machine.h machine.c machine.def loader.c
symbol.c syscall.c
ln -s target-pisa/config.h config.h
ln -s target-pisa/pisa.h machine.h
ln -s target-pisa/pisa.c machine.c
ln -s target-pisa/pisa.def machine.def
ln -s target-pisa/loader.c loader.c
ln -s target-pisa/symbol.c symbol.c
ln -s target-pisa/syscall.c syscall.c
rm -f tests
ln -s tests-pisa tests
ceciliazhang@ceciliazhang-VirtualBox:~/simplesim/simplesim-3.0$ make
gcc -DDEBUG -o sysprobe sysprobe.c
sysprobe.c: In function ‘main’:
sysprobe.c:263:39: warning: format ‘%d’ expects argument of type ‘int’,
but argument 3 has type ‘long unsigned int’ [-Wformat=]
      fprintf(stdout, "sizeof(int) = %d\n", sizeof(int));
                                      ~^
                                      %ld
sysprobe.c:264:40: warning: format ‘%d’ expects argument of type ‘int’,
but argument 3 has type ‘long unsigned int’ [-Wformat=]
      fprintf(stdout, "sizeof(long) = %d\n", sizeof(long));
                                       ~^
                                       %ld
endian probe results: little
probe flags: -DBYTES_LITTLE_ENDIAN -DWORDS_LITTLE_ENDIAN -DFAST_SRL
-DFAST_SRA -DGZIP_PATH="/bin/gzip"
probe libs: -lm
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c sim-fast.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c main.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c syscall.c
In file included from syscall.c:56:0:
syscall.c: In function ‘sys_syscall’:
syscall.c:1814:41: warning: cast from pointer to integer of different
size [-Wpointer-to-int-cast]
```

```
      iov[i].iov_base = (char *)MD_SWAPW((unsigned)iov[i].iov_base);
                                         ^
machine.h:195:23: note: in definition of macro ‘MD_SWAPW’
 #define MD_SWAPW(X)  (X)
                       ^
syscall.c:1814:24: warning: cast to pointer from integer of different
size [-Wint-to-pointer-cast]
      iov[i].iov_base = (char *)MD_SWAPW((unsigned)iov[i].iov_base);
                        ^
syscall.c:1821:32: warning: cast from pointer to integer of different
size [-Wpointer-to-int-cast]
   mem_bcopy(mem_fn, mem, Read, (md_addr_t)iov[i].iov_base,
                                ^
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c memory.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c regs.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c loader.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c endian.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c dlite.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c symbol.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c eval.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c options.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c stats.c
stats.c: In function ‘print_dist’:
stats.c:651:33: warning: variable ‘imin’ set but not used
[-Wunused-but-set-variable]
   unsigned int i, bcount, imax, imin;
                                 ^~~~
stats.c:651:27: warning: variable ‘imax’ set but not used
[-Wunused-but-set-variable]
   unsigned int i, bcount, imax, imin;
                           ^~~~
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c eio.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c range.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c misc.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c machine.c
cd libexo ; \
make "MAKE=make" "CC=gcc" "AR=ar qcv" "AROPT=" "RANLIB=ranlib"
"CFLAGS=`./sysprobe -flags` -DDEBUG -O0 -g -Wall" "OEXT=o" "LEXT=a"
"EEXT=" "X=/" "RM=rm -f" libexo.a
make[1]: Entering directory
'/home/ceciliazhang/simplesim/simplesim-3.0/libexo'
gcc -DBYTES_LITTLE_ENDIAN -DWORDS_LITTLE_ENDIAN -DFAST_SRL -DFAST_SRA
-DGZIP_PATH="/bin/gzip"  -DDEBUG -O0 -g -Wall -c libexo.c
gcc -DBYTES_LITTLE_ENDIAN -DWORDS_LITTLE_ENDIAN -DFAST_SRL -DFAST_SRA
-DGZIP_PATH="/bin/gzip"  -DDEBUG -O0 -g -Wall -c exolex.c
rm -f libexo.a
ar qcv libexo.a libexo.o exolex.o
a - libexo.o
a - exolex.o
ranlib libexo.a
make[1]: Leaving directory
'/home/ceciliazhang/simplesim/simplesim-3.0/libexo'
gcc -o sim-fast `./sysprobe -flags` -DDEBUG -O0 -g -Wall sim-fast.o
main.o syscall.o memory.o regs.o loader.o endian.o dlite.o symbol.o
```

```
eval.o options.o stats.o eio.o range.o misc.o machine.o libexo/libexo.a
`./sysprobe -libs` -lm
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c sim-safe.c
gcc -o sim-safe `./sysprobe -flags` -DDEBUG -O0 -g -Wall sim-safe.o
main.o syscall.o memory.o regs.o loader.o endian.o dlite.o symbol.o
eval.o options.o stats.o eio.o range.o misc.o machine.o libexo/libexo.a
`./sysprobe -libs` -lm
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c sim-eio.c
gcc -o sim-eio `./sysprobe -flags` -DDEBUG -O0 -g -Wall sim-eio.o main.o
syscall.o memory.o regs.o loader.o endian.o dlite.o symbol.o eval.o
options.o stats.o eio.o range.o misc.o machine.o libexo/libexo.a
`./sysprobe -libs` -lm
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c sim-bpred.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c bpred.c
gcc -o sim-bpred `./sysprobe -flags` -DDEBUG -O0 -g -Wall sim-bpred.o
bpred.o main.o syscall.o memory.o regs.o loader.o endian.o dlite.o
symbol.o eval.o options.o stats.o eio.o range.o misc.o machine.o
libexo/libexo.a `./sysprobe -libs` -lm
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c sim-profile.c
sim-profile.c: In function ‘sim_main’:
sim-profile.c:649:22: warning: variable ‘fault’ set but not used
[-Wunused-but-set-variable]
  enum md_fault_type fault;
                     ^~~~
gcc -o sim-profile `./sysprobe -flags` -DDEBUG -O0 -g -Wall
sim-profile.o main.o syscall.o memory.o regs.o loader.o endian.o dlite.o
symbol.o eval.o options.o stats.o eio.o range.o misc.o machine.o
libexo/libexo.a `./sysprobe -libs` -lm
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c sim-cache.c
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c cache.c
gcc -o sim-cache `./sysprobe -flags` -DDEBUG -O0 -g -Wall sim-cache.o
cache.o main.o syscall.o memory.o regs.o loader.o endian.o dlite.o
symbol.o eval.o options.o stats.o eio.o range.o misc.o machine.o
libexo/libexo.a `./sysprobe -libs` -lm
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c sim-outorder.c
sim-outorder.c: In function ‘ruu_dispatch’:
sim-outorder.c:3723:11: warning: unused variable ‘temp_qword’
[-Wunused-variable]
  qword_t temp_qword = 0;  /* " ditto " */
          ^~~~~~~~~~
sim-outorder.c:3717:7: warning: variable ‘br_taken’ set but not used
[-Wunused-but-set-variable]
  int br_taken, br_pred_taken;  /* if br, taken?  predicted taken? */
      ^~~~~~~~
sim-outorder.c: In function ‘sim_main’:
sim-outorder.c:4455:15: warning: unused variable ‘temp_qword’
[-Wunused-variable]
       qword_t temp_qword = 0;  /* " ditto " */
               ^~~~~~~~~~
sim-outorder.c:4448:17: warning: variable ‘target_PC’ set but not used
[-Wunused-but-set-variable]
       md_addr_t target_PC;  /* actual next/target PC address */
                 ^~~~~~~~~
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c resource.c
```

```
gcc `./sysprobe -flags` -DDEBUG -O0 -g -Wall   -c ptrace.c
gcc -o sim-outorder `./sysprobe -flags` -DDEBUG -O0 -g -Wall
sim-outorder.o cache.o bpred.o resource.o ptrace.o main.o syscall.o
memory.o regs.o loader.o endian.o dlite.o symbol.o eval.o options.o
stats.o eio.o range.o misc.o machine.o libexo/libexo.a `./sysprobe
-libs` -lm
my work is done here...
```

1. What is the performance of running the program, equake, under the default system setup (without changing any simulation parameters) using command: ./sim-outorder -fastfwd 350000000 -max:inst 250000000 equake.ss < equake.in ?

```
ceciliazhang@ceciliazhang-VirtualBox:~/simplesim/simplesim-3.0$ ./sim-outorder -fastfwd 350000000 -
max:inst 250000000 equake.ss<equake.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 -fastfwd 350000000 -max:inst 250000000
equake.ss

sim: simulation started @ Tue Nov 20 16:03:16 2018, options follow:

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

# -config                   # load configuration from a file
# -dumpconfig                # dump configuration to a file
# -h                 false # print help message
# -v                 false # verbose operation
# -d                 false # enable debug message
# -i                 false # start in Dlite debugger
-seed                    1 # random number generator seed (0 for timer seed)
# -q                 false # initialize and terminate immediately
# -chkpt             <null> # restore EIO trace execution from <fname>
# -redir:sim         <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          250000000 # maximum number of inst's to execute
-fastfwd           350000000 # 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 {IDIWB} (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: ** fast forwarding 350000000 insts **
equake00: Reading nodes.
equake00: Reading elements.
sim: ** starting performance simulation **

sim: ** simulation statistics **
sim_num_insn            250000000 # total number of instructions committed
sim_num_refs             81104979 # total number of loads and stores committed
sim_num_loads            56626333 # total number of loads committed
sim_num_stores        24478646.0000 # total number of stores committed
sim_num_branches         65928024 # total number of branches committed
sim_elapsed_time              176 # total simulation time in seconds
sim_inst_rate        1420454.5455 # simulation speed (in insts/sec)
sim_total_insn           263687144 # total number of instructions executed
sim_total_refs           85620994 # total number of loads and stores executed
sim_total_loads          60049176 # total number of loads executed
sim_total_stores      25571818.0000 # total number of stores executed
sim_total_branches       68912684 # total number of branches executed
sim_cycle               148193249 # total simulation time in cycles
sim_IPC                   1.6870 # instructions per cycle
sim_CPI                   0.5928 # cycles per instruction
sim_exec_BW               1.7793 # total instructions (mis-spec + committed) per cycle
sim_IPB                   3.7920 # instruction per branch
```

```
IFQ_count              379237627 # cumulative IFQ occupancy
IFQ_fcount              79908712 # cumulative IFQ full count
ifq_occupancy             2.5591 # avg IFQ occupancy (insn's)
ifq_rate                  1.7793 # avg IFQ dispatch rate (insn/cycle)
ifq_latency               1.4382 # avg IFQ occupant latency (cycle's)
ifq_full                  0.5392 # fraction of time (cycle's) IFQ was full
RUU_count              1517615680 # cumulative RUU occupancy
RUU_fcount              41502815 # cumulative RUU full count
ruu_occupancy            10.2408 # avg RUU occupancy (insn's)
ruu_rate                  1.7793 # avg RUU dispatch rate (insn/cycle)
ruu_latency               5.7554 # avg RUU occupant latency (cycle's)
ruu_full                  0.2801 # fraction of time (cycle's) RUU was full
LSQ_count               487418024 # cumulative LSQ occupancy
LSQ_fcount              14972254 # cumulative LSQ full count
lsq_occupancy             3.2891 # avg LSQ occupancy (insn's)
lsq_rate                  1.7793 # avg LSQ dispatch rate (insn/cycle)
lsq_latency               1.8485 # avg LSQ occupant latency (cycle's)
lsq_full                  0.1010 # fraction of time (cycle's) LSQ was full
sim_slip               2263273292 # total number of slip cycles
avg_sim_slip              9.0531 # the average slip between issue and retirement
bpred_bimod.lookups        70004633 # total number of bpred lookups
bpred_bimod.updates        65928022 # total number of updates
bpred_bimod.addr_hits      64762569 # total number of address-predicted hits
bpred_bimod.dir_hits       64762636 # total number of direction-predicted hits (includes addr-hits)
bpred_bimod.misses          1165386 # total number of misses
bpred_bimod.jr_hits         2911911 # total number of address-predicted hits for JR's
bpred_bimod.jr_seen         2911916 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP       363636 # total number of address-predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP       363637 # total number of non-RAS JR's seen
bpred_bimod.bpred_addr_rate    0.9823 # branch address-prediction rate (i.e., addr-hits/updates)
bpred_bimod.bpred_dir_rate    0.9823 # branch direction-prediction rate (i.e., all-hits/updates)
bpred_bimod.bpred_jr_rate    1.0000 # JR address-prediction rate (i.e., JR addr-hits/JRs seen)
bpred_bimod.bpred_jr_non_ras_rate.PP    1.0000 # non-RAS JR addr-pred rate (ie, non-RAS JR hits/JRs
seen)
bpred_bimod.retstack_pushes      2622134 # total number of address
pushed onto ret-addr stack
bpred_bimod.retstack_pops      2548279 # total number of address popped
off of ret-addr stack
bpred_bimod.used_ras.PP      2548279 # total number of RAS predictions used
bpred_bimod.ras_hits.PP      2548275 # total number of RAS hits
bpred_bimod.ras_rate.PP    1.0000 # RAS prediction rate (i.e., RAS
hits/used RAS)
il1.accesses            274244902 # total number of accesses
il1.hits                268056105 # total number of hits
il1.misses                6188797 # total number of misses
il1.replacements          6188592 # total number of replacements
il1.writebacks                  0 # total number of writebacks
il1.invalidations               0 # total number of invalidations
il1.miss_rate              0.0226 # miss rate (i.e., misses/ref)
il1.repl_rate              0.0226 # 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              81906212 # total number of accesses
dl1.hits                  81850079 # total number of hits
```

```
dl1.misses                56133 # total number of misses
dl1.replacements          55621 # total number of replacements
dl1.writebacks            45384 # total number of writebacks
dl1.invalidations             0 # total number of invalidations
dl1.miss_rate            0.0007 # miss rate (i.e., misses/ref)
dl1.repl_rate            0.0007 # replacement rate (i.e., repls/ref)
dl1.wb_rate              0.0006 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate             0.0000 # invalidation rate (i.e., invs/ref)
ul2.accesses            6290314 # total number of accesses
ul2.hits                6262793 # total number of hits
ul2.misses                27521 # total number of misses
ul2.replacements          23425 # total number of replacements
ul2.writebacks            19350 # total number of writebacks
ul2.invalidations             0 # total number of invalidations
ul2.miss_rate            0.0044 # miss rate (i.e., misses/ref)
ul2.repl_rate            0.0037 # replacement rate (i.e., repls/ref)
ul2.wb_rate              0.0031 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate             0.0000 # invalidation rate (i.e., invs/ref)
itlb.accesses          274244902 # total number of accesses
itlb.hits              274244890 # total number of hits
itlb.misses                   12 # 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           83360800 # total number of accesses
dtlb.hits               83360364 # total number of hits
dtlb.misses                 436 # total number of misses
dtlb.replacements           308 # 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              132784 # program text (code) size in bytes
ld_data_base          0x10000000 # program initialized data segment base
ld_data_size               16384 # 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              1203 # total number of pages allocated
```

```
mem.page_mem              4812k # total size of memory pages allocated
mem.ptab_misses           20663 # total first level page table misses
mem.ptab_accesses      3909621480 # total page table accesses
mem.ptab_miss_rate        0.0000 # first level page table miss rate
```

Comment:

From the statistics, following data could be obtained:

```
sim_num_insn      250000000 # total number of instructions committed
sim_cycle          148193249 # total simulation time in cycles
sim_IPC              1.6870 # instructions per cycle
sim_CPI              0.5928 # cycles per instruction
```

And based on above data, we can get following parameters:

IPC=250000000/148193249 and CPI=1/IPC which are exactly the same result as sim_IPC and sm_CPI.

2. How much is the performance loss if the processor uses in-order execution instead of the default out-of-order execution to run the program?

```
ceciliazhang@ceciliazhang-VirtualBox:~/simplesim/simplesim-3.0$ ./sim-outorder -fastfwd 350000000 -
max:inst 250000000 -issue:inorder true equake.ss<equake.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 -fastfwd 350000000 -max:inst 250000000
-issue:inorder true equake.ss

sim: simulation started @ Tue Nov 20 16:15:24 2018, options follow:

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

# -config                   # load configuration from a file
# -dumpconfig                   # dump configuration to a file
# -h                    false # print help message
# -v                    false # verbose operation
# -d                    false # enable debug message
# -i                    false # start in Dlite debugger
-seed                       1 # random number generator seed (0 for
timer seed)
# -q                    false # initialize and terminate immediately
# -chkpt                <null> # restore EIO trace execution from <fname>
# -redir:sim            <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           250000000 # maximum number of inst's to execute
-fastfwd            350000000 # 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
{IDIWB} (default non-spec)
-decode:width              4 # instruction decode B/W (insts/cycle)
-issue:width               4 # instruction issue B/W (insts/cycle)
-issue:inorder           true # 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: ** fast forwarding 350000000 insts **
equake00: Reading nodes.
equake00: Reading elements.
```

```
sim: ** starting performance simulation **

sim: ** simulation statistics **
sim_num_insn              250000000 # total number of instructions committed
sim_num_refs               81104979 # total number of loads and stores
committed
sim_num_loads              56626333 # total number of loads committed
sim_num_stores         24478646.0000 # total number of stores committed
sim_num_branches           65928024 # total number of branches committed
sim_elapsed_time                315 # total simulation time in seconds
sim_inst_rate           793650.7937 # simulation speed (in insts/sec)
sim_total_insn             251893224 # total number of instructions executed
sim_total_refs             81543049 # total number of loads and stores
executed
sim_total_loads            57063838 # total number of loads executed
sim_total_stores       24479211.0000 # total number of stores executed
sim_total_branches         65928031 # total number of branches executed
sim_cycle                321272084 # total simulation time in cycles
sim_IPC                     0.7782 # instructions per cycle
sim_CPI                     1.2851 # cycles per instruction
sim_exec_BW                 0.7840 # total instructions (mis-spec +
committed) per cycle
sim_IPB                     3.7920 # instruction per branch
IFQ_count                1117739234 # cumulative IFQ occupancy
IFQ_fcount                264166526 # cumulative IFQ full count
ifq_occupancy               3.4791 # avg IFQ occupancy (insn's)
ifq_rate                    0.7840 # avg IFQ dispatch rate (insn/cycle)
ifq_latency                 4.4374 # avg IFQ occupant latency (cycle's)
ifq_full                    0.8223 # fraction of time (cycle's) IFQ was
full
RUU_count                 870668496 # cumulative RUU occupancy
RUU_fcount                        0 # cumulative RUU full count
ruu_occupancy               2.7101 # avg RUU occupancy (insn's)
ruu_rate                    0.7840 # avg RUU dispatch rate (insn/cycle)
ruu_latency                 3.4565 # avg RUU occupant latency (cycle's)
ruu_full                    0.0000 # fraction of time (cycle's) RUU was
full
LSQ_count                 303813606 # cumulative LSQ occupancy
LSQ_fcount                        0 # cumulative LSQ full count
lsq_occupancy               0.9457 # avg LSQ occupancy (insn's)
lsq_rate                    0.7840 # avg LSQ dispatch rate (insn/cycle)
lsq_latency                 1.2061 # avg LSQ occupant latency (cycle's)
lsq_full                    0.0000 # fraction of time (cycle's) LSQ was
full
sim_slip                 1503255186 # total number of slip cycles
avg_sim_slip                6.0130 # the average slip between issue and
retirement
bpred_bimod.lookups        67020644 # total number of bpred lookups
bpred_bimod.updates        65928024 # total number of updates
bpred_bimod.addr_hits      64762571 # total number of address-predicted hits
bpred_bimod.dir_hits       64762638 # total number of
direction-predicted hits (includes addr-hits)
bpred_bimod.misses          1165386 # total number of misses
bpred_bimod.jr_hits        2911911 # total number of address-predicted
```

```
hits for JR's
bpred_bimod.jr_seen         2911916 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP       363636 # total number of
address-predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP       363637 # total number of non-RAS
JR's seen
bpred_bimod.bpred_addr_rate    0.9823 # branch address-prediction rate
(i.e., addr-hits/updates)
bpred_bimod.bpred_dir_rate    0.9823 # branch direction-prediction rate
(i.e., all-hits/updates)
bpred_bimod.bpred_jr_rate    1.0000 # JR address-prediction rate (i.e.,
JR addr-hits/JRs seen)
bpred_bimod.bpred_jr_non_ras_rate.PP    1.0000 # non-RAS JR addr-pred
rate (ie, non-RAS JR hits/JRs seen)
bpred_bimod.retstack_pushes      2548843 # total number of address
pushed onto ret-addr stack
bpred_bimod.retstack_pops      2548279 # total number of address popped
off of ret-addr stack
bpred_bimod.used_ras.PP      2548279 # total number of RAS predictions used
bpred_bimod.ras_hits.PP      2548275 # total number of RAS hits
bpred_bimod.ras_rate.PP    1.0000 # RAS prediction rate (i.e., RAS
hits/used RAS)
il1.accesses             262743371 # total number of accesses
il1.hits                 256554574 # total number of hits
il1.misses                 6188797 # total number of misses
il1.replacements           6188592 # total number of replacements
il1.writebacks                   0 # total number of writebacks
il1.invalidations                0 # total number of invalidations
il1.miss_rate               0.0236 # miss rate (i.e., misses/ref)
il1.repl_rate               0.0236 # 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               81104978 # total number of accesses
dl1.hits                   81048949 # total number of hits
dl1.misses                    56029 # total number of misses
dl1.replacements              55517 # total number of replacements
dl1.writebacks                45359 # total number of writebacks
dl1.invalidations                0 # total number of invalidations
dl1.miss_rate               0.0007 # miss rate (i.e., misses/ref)
dl1.repl_rate               0.0007 # replacement rate (i.e., repls/ref)
dl1.wb_rate                 0.0006 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate                0.0000 # invalidation rate (i.e., invs/ref)
ul2.accesses                6290185 # total number of accesses
ul2.hits                    6262665 # total number of hits
ul2.misses                    27520 # total number of misses
ul2.replacements              23424 # total number of replacements
ul2.writebacks                19349 # total number of writebacks
ul2.invalidations                0 # total number of invalidations
ul2.miss_rate               0.0044 # miss rate (i.e., misses/ref)
ul2.repl_rate               0.0037 # replacement rate (i.e., repls/ref)
ul2.wb_rate                 0.0031 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate                0.0000 # invalidation rate (i.e., invs/ref)
itlb.accesses             262743371 # total number of accesses
itlb.hits                 262743359 # total number of hits
```

```
itlb.misses                 12 # 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          81104978 # total number of accesses
dtlb.hits              81104543 # total number of hits
dtlb.misses                435 # total number of misses
dtlb.replacements          307 # 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             132784 # program text (code) size in bytes
ld_data_base         0x10000000 # program initialized data segment base
ld_data_size              16384 # 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            1203 # total number of pages allocated
mem.page_mem              4812k # total size of memory pages allocated
mem.ptab_misses           20663 # total first level page table misses
mem.ptab_accesses     3857645810 # total page table accesses
mem.ptab_miss_rate       0.0000 # first level page table miss rate
```

Comment:

From above statistics, we know that currently:

sim_IPC 0.7782 # instructions per cycle

sim_CPI 1.2851 # cycles per instruction

, therefor:

speed-up=1.6870/0.7782=2.16782318

Loss=1.6870-0.7782/1.6870=0.53870777

3. The above experiments only perform detailed simulation on 250 million instructions. Based on the simulator running time in Question 1, estimate how long it would take to simulate the program's execution in details from beginning to end using the default configuration. Note: Do not actually run the detailed simulation from beginning to end. It may take hours or even days to finish depending on the speed of your computer.

```
ceciliazhang@ceciliazhang-VirtualBox:~/simplesim/simplesim-3.0$ ./sim-safe equake.ss<equake.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 equake.ss

sim: simulation started @ Tue Nov 20 16:21:23 2018, 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 **
equake00: Reading nodes.
equake00: Reading elements.
equake00: Reading sparse matrix structure.
equake00: Beginning simulation.

CASE SUMMARY
Fault information
  Orientation:  strike: 1.937315
                  dip: 0.767945
                 rake: 1.221730
           dislocation: 29.640788 cm
```

```
Hypocenter: (32.264153, 23.814432, -11.250000) Km
Excitation characteristics
     Time step: 0.002400 sec
      Duration: 9.250000 sec
     Rise time: 0.600000 sec

The source is node 5903 at (32.250000  24.364300  -11.200000)
The epicenter is node 16745 at (32.250000  23.687500  0.000000)

Time step 30
5903: -3.42e-02 -3.11e-02 -4.03e-02
16745: 9.04e-35 -8.75e-34 4.41e-33
Time step 60
5903: -2.46e-01 -2.27e-01 -2.98e-01
16745: 2.45e-27 -2.59e-26 1.30e-25
Time step 90
5903: -7.38e-01 -6.95e-01 -9.22e-01
16745: 4.35e-23 -5.32e-22 2.68e-21
Time step 120
5903: -1.49e+00 -1.44e+00 -1.92e+00
16745: 3.74e-20 -5.61e-19 2.83e-18
Time step 150
5903: -2.37e+00 -2.34e+00 -3.12e+00
16745: 5.89e-18 -1.14e-16 5.76e-16
Time step 180
5903: -3.17e+00 -3.21e+00 -4.26e+00
16745: 3.21e-16 -8.11e-15 4.08e-14
Time step 210
5903: -3.73e+00 -3.88e+00 -5.10e+00
16745: 9.19e-15 -2.76e-13 1.38e-12
Time step 240
5903: -4.00e+00 -4.27e+00 -5.57e+00
16745: 1.91e-13 -5.42e-12 2.70e-11
Time step 270
5903: -4.08e+00 -4.46e+00 -5.78e+00
16745: 3.28e-12 -7.00e-11 3.45e-10
Time step 300
5903: -4.06e+00 -4.52e+00 -5.86e+00
16745: 4.41e-11 -6.45e-10 3.14e-09
Time step 330
5903: -3.98e+00 -4.51e+00 -5.90e+00
16745: 4.49e-10 -4.50e-09 2.18e-08
Time step 360
5903: -3.91e+00 -4.49e+00 -5.96e+00
16745: 3.47e-09 -2.49e-08 1.20e-07
Time step 390
5903: -3.86e+00 -4.48e+00 -6.07e+00
16745: 2.10e-08 -1.13e-07 5.42e-07
Time step 420
5903: -3.85e+00 -4.48e+00 -6.21e+00
16745: 1.02e-07 -4.35e-07 2.08e-06
Time step 450
5903: -3.86e+00 -4.49e+00 -6.36e+00
16745: 4.12e-07 -1.44e-06 6.89e-06
```

Time step 480
5903: -3.89e+00 -4.50e+00 -6.48e+00
16745: 1.40e-06 -4.17e-06 2.01e-05
Time step 510
5903: -3.91e+00 -4.50e+00 -6.57e+00
16745: 4.12e-06 -1.08e-05 5.26e-05
Time step 540
5903: -3.93e+00 -4.50e+00 -6.62e+00
16745: 1.06e-05 -2.53e-05 1.24e-04
Time step 570
5903: -3.95e+00 -4.50e+00 -6.65e+00
16745: 2.40e-05 -5.46e-05 2.69e-04
Time step 600
5903: -3.96e+00 -4.51e+00 -6.67e+00
16745: 4.85e-05 -1.09e-04 5.38e-04
Time step 630
5903: -3.97e+00 -4.52e+00 -6.67e+00
16745: 8.80e-05 -2.06e-04 1.00e-03
Time step 660
5903: -3.98e+00 -4.54e+00 -6.68e+00
16745: 1.43e-04 -3.68e-04 1.74e-03
Time step 690
5903: -3.98e+00 -4.56e+00 -6.67e+00
16745: 2.09e-04 -6.26e-04 2.84e-03
Time step 720
5903: -3.98e+00 -4.57e+00 -6.67e+00
16745: 2.70e-04 -1.02e-03 4.36e-03
Time step 750
5903: -3.98e+00 -4.59e+00 -6.66e+00
16745: 3.01e-04 -1.60e-03 6.31e-03
Time step 780
5903: -3.98e+00 -4.60e+00 -6.65e+00
16745: 2.64e-04 -2.41e-03 8.59e-03
Time step 810
5903: -3.97e+00 -4.60e+00 -6.63e+00
16745: 1.20e-04 -3.46e-03 1.10e-02
Time step 840
5903: -3.97e+00 -4.60e+00 -6.62e+00
16745: -1.70e-04 -4.77e-03 1.31e-02
Time step 870
5903: -3.97e+00 -4.61e+00 -6.61e+00
16745: -6.32e-04 -6.28e-03 1.44e-02
Time step 900
5903: -3.98e+00 -4.60e+00 -6.62e+00
16745: -1.27e-03 -7.90e-03 1.43e-02
Time step 930
5903: -3.98e+00 -4.60e+00 -6.63e+00
16745: -2.07e-03 -9.50e-03 1.22e-02
Time step 960
5903: -3.99e+00 -4.60e+00 -6.64e+00
16745: -2.95e-03 -1.09e-02 7.97e-03
Time step 990
5903: -3.99e+00 -4.60e+00 -6.66e+00
16745: -3.81e-03 -1.20e-02 1.42e-03

Time step 1020
5903: -4.00e+00 -4.60e+00 -6.69e+00
16745: -4.52e-03 -1.26e-02 -7.06e-03
Time step 1050
5903: -4.00e+00 -4.60e+00 -6.70e+00
16745: -4.90e-03 -1.26e-02 -1.68e-02
Time step 1080
5903: -4.00e+00 -4.61e+00 -6.71e+00
16745: -4.81e-03 -1.19e-02 -2.69e-02
Time step 1110
5903: -4.00e+00 -4.61e+00 -6.71e+00
16745: -4.11e-03 -1.04e-02 -3.65e-02
Time step 1140
5903: -3.99e+00 -4.61e+00 -6.70e+00
16745: -2.80e-03 -8.05e-03 -4.47e-02
Time step 1170
5903: -3.98e+00 -4.61e+00 -6.70e+00
16745: -9.48e-04 -4.85e-03 -5.13e-02
Time step 1200
5903: -3.98e+00 -4.61e+00 -6.69e+00
16745: 1.24e-03 -8.07e-04 -5.62e-02
Time step 1230
5903: -3.97e+00 -4.61e+00 -6.69e+00
16745: 3.50e-03 3.90e-03 -5.96e-02
Time step 1260
5903: -3.97e+00 -4.61e+00 -6.69e+00
16745: 5.51e-03 8.96e-03 -6.22e-02
Time step 1290
5903: -3.98e+00 -4.61e+00 -6.70e+00
16745: 7.02e-03 1.39e-02 -6.46e-02
Time step 1320
5903: -3.99e+00 -4.61e+00 -6.71e+00
16745: 7.88e-03 1.84e-02 -6.71e-02
Time step 1350
5903: -3.99e+00 -4.61e+00 -6.71e+00
16745: 8.13e-03 2.19e-02 -7.02e-02
Time step 1380
5903: -4.00e+00 -4.61e+00 -6.72e+00
16745: 8.02e-03 2.45e-02 -7.39e-02
Time step 1410
5903: -4.01e+00 -4.61e+00 -6.72e+00
16745: 7.97e-03 2.61e-02 -7.81e-02
Time step 1440
5903: -4.01e+00 -4.62e+00 -6.71e+00
16745: 8.47e-03 2.74e-02 -8.24e-02
Time step 1470
5903: -4.01e+00 -4.62e+00 -6.71e+00
16745: 9.95e-03 2.88e-02 -8.63e-02
Time step 1500
5903: -4.00e+00 -4.62e+00 -6.70e+00
16745: 1.26e-02 3.08e-02 -8.93e-02
Time step 1530
5903: -4.00e+00 -4.62e+00 -6.70e+00
16745: 1.61e-02 3.38e-02 -9.11e-02

Time step 1560
5903: -3.99e+00 -4.62e+00 -6.71e+00
16745: 2.00e-02 3.78e-02 -9.15e-02
Time step 1590
5903: -3.99e+00 -4.62e+00 -6.71e+00
16745: 2.33e-02 4.27e-02 -9.07e-02
Time step 1620
5903: -3.99e+00 -4.62e+00 -6.72e+00
16745: 2.52e-02 4.78e-02 -8.89e-02
Time step 1650
5903: -3.99e+00 -4.63e+00 -6.72e+00
16745: 2.54e-02 5.25e-02 -8.67e-02
Time step 1680
5903: -3.99e+00 -4.63e+00 -6.72e+00
16745: 2.37e-02 5.59e-02 -8.49e-02
Time step 1710
5903: -3.99e+00 -4.63e+00 -6.72e+00
16745: 2.09e-02 5.72e-02 -8.37e-02
Time step 1740
5903: -3.99e+00 -4.62e+00 -6.72e+00
16745: 1.77e-02 5.58e-02 -8.35e-02
Time step 1770
5903: -3.99e+00 -4.62e+00 -6.71e+00
16745: 1.51e-02 5.16e-02 -8.41e-02
Time step 1800
5903: -3.99e+00 -4.62e+00 -6.70e+00
16745: 1.37e-02 4.48e-02 -8.53e-02
Time step 1830
5903: -3.99e+00 -4.61e+00 -6.70e+00
16745: 1.34e-02 3.64e-02 -8.67e-02
Time step 1860
5903: -3.99e+00 -4.61e+00 -6.70e+00
16745: 1.38e-02 2.75e-02 -8.79e-02
Time step 1890
5903: -3.99e+00 -4.61e+00 -6.71e+00
16745: 1.41e-02 1.93e-02 -8.82e-02
Time step 1920
5903: -3.99e+00 -4.61e+00 -6.72e+00
16745: 1.37e-02 1.28e-02 -8.73e-02
Time step 1950
5903: -3.99e+00 -4.61e+00 -6.72e+00
16745: 1.21e-02 8.32e-03 -8.49e-02
Time step 1980
5903: -4.00e+00 -4.62e+00 -6.73e+00
16745: 9.36e-03 6.00e-03 -8.12e-02
Time step 2010
5903: -4.00e+00 -4.62e+00 -6.74e+00
16745: 5.83e-03 5.39e-03 -7.69e-02
Time step 2040
5903: -4.00e+00 -4.62e+00 -6.74e+00
16745: 2.02e-03 5.86e-03 -7.32e-02
Time step 2070
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: -1.59e-03 6.73e-03 -7.15e-02

Time step 2100
5903: -3.99e+00 -4.61e+00 -6.74e+00
16745: -4.65e-03 7.35e-03 -7.29e-02
Time step 2130
5903: -3.99e+00 -4.61e+00 -6.74e+00
16745: -6.96e-03 7.32e-03 -7.76e-02
Time step 2160
5903: -3.99e+00 -4.61e+00 -6.74e+00
16745: -8.40e-03 6.54e-03 -8.52e-02
Time step 2190
5903: -3.99e+00 -4.62e+00 -6.74e+00
16745: -8.85e-03 5.26e-03 -9.40e-02
Time step 2220
5903: -3.99e+00 -4.62e+00 -6.73e+00
16745: -8.19e-03 4.00e-03 -1.02e-01
Time step 2250
5903: -3.99e+00 -4.62e+00 -6.73e+00
16745: -6.38e-03 3.39e-03 -1.08e-01
Time step 2280
5903: -3.99e+00 -4.62e+00 -6.74e+00
16745: -3.57e-03 3.94e-03 -1.09e-01
Time step 2310
5903: -3.99e+00 -4.62e+00 -6.74e+00
16745: -1.01e-04 5.90e-03 -1.06e-01
Time step 2340
5903: -3.99e+00 -4.62e+00 -6.74e+00
16745: 3.50e-03 9.25e-03 -9.96e-02
Time step 2370
5903: -3.99e+00 -4.62e+00 -6.74e+00
16745: 6.74e-03 1.38e-02 -9.05e-02
Time step 2400
5903: -3.99e+00 -4.61e+00 -6.74e+00
16745: 9.30e-03 1.92e-02 -8.16e-02
Time step 2430
5903: -3.99e+00 -4.61e+00 -6.74e+00
16745: 1.11e-02 2.51e-02 -7.53e-02
Time step 2460
5903: -3.99e+00 -4.61e+00 -6.74e+00
16745: 1.21e-02 3.10e-02 -7.37e-02
Time step 2490
5903: -3.99e+00 -4.61e+00 -6.75e+00
16745: 1.24e-02 3.57e-02 -7.71e-02
Time step 2520
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: 1.18e-02 3.79e-02 -8.48e-02
Time step 2550
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 1.03e-02 3.64e-02 -9.46e-02
Time step 2580
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 7.62e-03 3.03e-02 -1.04e-01
Time step 2610
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 4.18e-03 1.99e-02 -1.10e-01

Time step 2640
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 5.35e-04 7.14e-03 -1.13e-01
Time step 2670
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: -2.56e-03 -5.43e-03 -1.12e-01
Time step 2700
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: -4.49e-03 -1.48e-02 -1.08e-01
Time step 2730
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: -5.05e-03 -1.89e-02 -1.02e-01
Time step 2760
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: -4.49e-03 -1.70e-02 -9.61e-02
Time step 2790
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: -3.29e-03 -1.02e-02 -9.08e-02
Time step 2820
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: -1.89e-03 -7.66e-04 -8.67e-02
Time step 2850
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: -4.83e-04 8.74e-03 -8.44e-02
Time step 2880
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: 9.58e-04 1.62e-02 -8.41e-02
Time step 2910
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: 2.51e-03 2.08e-02 -8.60e-02
Time step 2940
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 4.11e-03 2.26e-02 -9.01e-02
Time step 2970
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 5.51e-03 2.27e-02 -9.57e-02
Time step 3000
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 6.42e-03 2.22e-02 -1.02e-01
Time step 3030
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 6.71e-03 2.20e-02 -1.07e-01
Time step 3060
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 6.49e-03 2.23e-02 -1.11e-01
Time step 3090
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 6.06e-03 2.26e-02 -1.13e-01
Time step 3120
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 5.65e-03 2.23e-02 -1.12e-01
Time step 3150
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 5.31e-03 2.07e-02 -1.08e-01

Time step 3180
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 4.88e-03 1.77e-02 -1.03e-01
Time step 3210
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 4.10e-03 1.38e-02 -9.78e-02
Time step 3240
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 2.83e-03 9.74e-03 -9.29e-02
Time step 3270
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 1.15e-03 6.89e-03 -8.91e-02
Time step 3300
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: -6.79e-04 6.27e-03 -8.66e-02
Time step 3330
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: -2.34e-03 8.46e-03 -8.52e-02
Time step 3360
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: -3.53e-03 1.32e-02 -8.49e-02
Time step 3390
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: -4.03e-03 1.96e-02 -8.57e-02
Time step 3420
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: -3.71e-03 2.58e-02 -8.78e-02
Time step 3450
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: -2.56e-03 3.00e-02 -9.16e-02
Time step 3480
5903: -3.99e+00 -4.61e+00 -6.76e+00
16745: -7.82e-04 3.07e-02 -9.70e-02
Time step 3510
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 1.23e-03 2.73e-02 -1.03e-01
Time step 3540
5903: -3.99e+00 -4.62e+00 -6.77e+00
16745: 2.99e-03 2.07e-02 -1.10e-01
Time step 3570
5903: -3.99e+00 -4.62e+00 -6.77e+00
16745: 4.11e-03 1.27e-02 -1.14e-01
Time step 3600
5903: -3.99e+00 -4.62e+00 -6.77e+00
16745: 4.60e-03 5.60e-03 -1.17e-01
Time step 3630
5903: -3.98e+00 -4.62e+00 -6.77e+00
16745: 4.81e-03 1.51e-03 -1.17e-01
Time step 3660
5903: -3.98e+00 -4.62e+00 -6.77e+00
16745: 5.30e-03 1.42e-03 -1.14e-01
Time step 3690
5903: -3.98e+00 -4.62e+00 -6.77e+00
16745: 6.40e-03 4.99e-03 -1.11e-01

```
Time step 3720
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 7.90e-03 1.08e-02 -1.06e-01
Time step 3750
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 9.03e-03 1.70e-02 -1.03e-01
Time step 3780
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 8.83e-03 2.21e-02 -1.01e-01
Time step 3810
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 6.61e-03 2.52e-02 -1.00e-01
Time step 3840
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 2.45e-03 2.66e-02 -1.01e-01
equake00: 30169 nodes 151173 elems 3855 timesteps

equake00: Done. Terminating the simulation.

sim: ** simulation statistics **
sim_num_insn          165643487723 # total number of instructions executed
sim_num_refs           78603143990 # total number of loads and stores executed
sim_elapsed_time              7318 # total simulation time in seconds
sim_inst_rate         22635076.2125 # simulation speed (in insts/sec)
ld_text_base            0x00400000 # program text (code) segment base
ld_text_size              132784 # program text (code) size in bytes
ld_data_base            0x10000000 # program initialized data segment base
ld_data_size               16384 # 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             10394 # total number of pages allocated
mem.page_mem               41576k # total size of memory pages allocated
mem.ptab_misses            3253141 # total first level page table misses
mem.ptab_accesses      906494436840 # total page table accesses
mem.ptab_miss_rate           0.0000 # first level page table miss rate
```

Comment:

From the first question, it can be figured out that: sim_inst_rate=1420454.5455 which is the simulation speed (in insts/sec).

And based on this rate, I estimate that

simulation time=IC/sim_inst_rate=165643487723/1420454.5455=32.39250426 hours.

4. An advantage of using simulator is that you can vary the processor parameters to see their performance impact. In the default configuration, the processor pipeline (fetch, decode, issue and commit) bandwidth is 4. If all the other parameters are kept the same, how much is the performance improvements when the pipeline bandwidth increases 8? Usually when the pipeline bandwidth increases, the capacity of other hardware components should also increase to have a balanced design. Which component should double its size first, RUU, LSQ, integer ALU, integer multiplier, floating-point ALU, or floating-point multiplier? Use experimental results to support your claim.

```
ceciliazhang@ceciliazhang-VirtualBox:~/simplesim/simplesim-3.0$ ./sim-outorder -fastfwd 350000000 -max:inst 250000000 -fetch:ifqsize 8 -decode:width 8 -issue:width 8 -commit:width 8 equake.ss<equake.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 -fastfwd 350000000 -max:inst 250000000 -fetch:ifqsize 8 -decode:width 8 -issue:width 8 -commit:width 8 equake.ss

sim: simulation started @ Tue Nov 20 19:08:24 2018, options follow:

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

# -config                   # load configuration from a file
# -dumpconfig               # dump configuration to a file
# -h                    false # print help message
# -v                    false # verbose operation
# -d                    false # enable debug message
# -i                    false # start in Dlite debugger
-seed                       1 # random number generator seed (0 for timer seed)
# -q                    false # initialize and terminate immediately
# -chkpt                <null> # restore EIO trace execution from <fname>
# -redir:sim            <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           250000000 # maximum number of inst's to execute
-fastfwd            350000000 # number of insts skipped before timing starts
# -ptrace               <null> # generate pipetrace, i.e., <fname|stdout|stderr> <range>
-fetch:ifqsize              8 # 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 {IDIWB} (default non-spec)
-decode:width              8 # instruction decode B/W (insts/cycle)
-issue:width               8 # 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              8 # 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: ** fast forwarding 350000000 insts **
equake00: Reading nodes.
equake00: Reading elements.
sim: ** starting performance simulation **

sim: ** simulation statistics **
sim_num_insn              250000001 # total number of instructions committed
sim_num_refs               81104979 # total number of loads and stores
committed
sim_num_loads              56626333 # total number of loads committed
sim_num_stores         24478646.0000 # total number of stores committed
sim_num_branches           65928024 # total number of branches committed
sim_elapsed_time                161 # total simulation time in seconds
sim_inst_rate          1552795.0373 # simulation speed (in insts/sec)
sim_total_insn            265867518 # total number of instructions executed
sim_total_refs             85619511 # total number of loads and stores
executed
sim_total_loads            60047786 # total number of loads executed
sim_total_stores       25571725.0000 # total number of stores executed
```

```
sim_total_branches        70003245 # total number of branches executed
sim_cycle                133647851 # total simulation time in cycles
sim_IPC                    1.8706 # instructions per cycle
sim_CPI                    0.5346 # cycles per instruction
sim_exec_BW                1.9893 # total instructions (mis-spec + committed) per cycle
sim_IPB                    3.7920 # instruction per branch
IFQ_count                657804619 # cumulative IFQ occupancy
IFQ_fcount                63017453 # cumulative IFQ full count
ifq_occupancy              4.9219 # avg IFQ occupancy (insn's)
ifq_rate                   1.9893 # avg IFQ dispatch rate (insn/cycle)
ifq_latency                2.4742 # avg IFQ occupant latency (cycle's)
ifq_full                   0.4715 # fraction of time (cycle's) IFQ was full
RUU_count               1496773735 # cumulative RUU occupancy
RUU_fcount                64799727 # cumulative RUU full count
ruu_occupancy             11.1994 # avg RUU occupancy (insn's)
ruu_rate                   1.9893 # avg RUU dispatch rate (insn/cycle)
ruu_latency                5.6298 # avg RUU occupant latency (cycle's)
ruu_full                   0.4849 # fraction of time (cycle's) RUU was full
LSQ_count                469444041 # cumulative LSQ occupancy
LSQ_fcount                16127229 # cumulative LSQ full count
lsq_occupancy              3.5125 # avg LSQ occupancy (insn's)
lsq_rate                   1.9893 # avg LSQ dispatch rate (insn/cycle)
lsq_latency                1.7657 # avg LSQ occupant latency (cycle's)
lsq_full                   0.1207 # fraction of time (cycle's) LSQ was full
sim_slip                2223004454 # total number of slip cycles
avg_sim_slip               8.8920 # the average slip between issue and retirement
bpred_bimod.lookups        72552625 # total number of bpred lookups
bpred_bimod.updates        65928022 # total number of updates
bpred_bimod.addr_hits      64762569 # total number of address-predicted hits
bpred_bimod.dir_hits       64762636 # total number of
direction-predicted hits (includes addr-hits)
bpred_bimod.misses          1165386 # total number of misses
bpred_bimod.jr_hits        2911911 # total number of address-predicted hits for JR's
bpred_bimod.jr_seen        2911916 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP       363636 # total number of address-predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP       363637 # total number of non-RAS JR's seen
bpred_bimod.bpred_addr_rate    0.9823 # branch address-prediction rate
(i.e., addr-hits/updates)
bpred_bimod.bpred_dir_rate    0.9823 # branch direction-prediction rate
(i.e., all-hits/updates)
bpred_bimod.bpred_jr_rate    1.0000 # JR address-prediction rate (i.e.,
JR addr-hits/JRs seen)
bpred_bimod.bpred_jr_non_ras_rate.PP    1.0000 # non-RAS JR addr-pred
rate (ie, non-RAS JR hits/JRs seen)
bpred_bimod.retstack_pushes      2622134 # total number of address
pushed onto ret-addr stack
bpred_bimod.retstack_pops      2549971 # total number of address popped
off of ret-addr stack
bpred_bimod.used_ras.PP      2548279 # total number of RAS predictions used
bpred_bimod.ras_hits.PP      2548275 # total number of RAS hits
bpred_bimod.ras_rate.PP    1.0000 # RAS prediction rate (i.e., RAS
hits/used RAS)
il1.accesses             280432100 # total number of accesses
il1.hits                 274243303 # total number of hits
```

```
il1.misses               6188797 # total number of misses
il1.replacements         6188592 # total number of replacements
il1.writebacks                 0 # total number of writebacks
il1.invalidations              0 # total number of invalidations
il1.miss_rate             0.0221 # miss rate (i.e., misses/ref)
il1.repl_rate             0.0221 # 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             82403617 # total number of accesses
dl1.hits                 82347454 # total number of hits
dl1.misses                  56163 # total number of misses
dl1.replacements            55651 # total number of replacements
dl1.writebacks              45398 # total number of writebacks
dl1.invalidations               0 # total number of invalidations
dl1.miss_rate             0.0007 # miss rate (i.e., misses/ref)
dl1.repl_rate             0.0007 # replacement rate (i.e., repls/ref)
dl1.wb_rate               0.0006 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate              0.0000 # invalidation rate (i.e., invs/ref)
ul2.accesses              6290358 # total number of accesses
ul2.hits                  6262837 # total number of hits
ul2.misses                  27521 # total number of misses
ul2.replacements            23425 # total number of replacements
ul2.writebacks              19350 # total number of writebacks
ul2.invalidations               0 # total number of invalidations
ul2.miss_rate             0.0044 # miss rate (i.e., misses/ref)
ul2.repl_rate             0.0037 # replacement rate (i.e., repls/ref)
ul2.wb_rate               0.0031 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate              0.0000 # invalidation rate (i.e., invs/ref)
itlb.accesses           280432100 # total number of accesses
itlb.hits               280432088 # total number of hits
itlb.misses                    12 # 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            83495696 # total number of accesses
dtlb.hits                83495260 # total number of hits
dtlb.misses                  436 # total number of misses
dtlb.replacements            308 # 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               132784 # program text (code) size in bytes
ld_data_base           0x10000000 # program initialized data segment base
ld_data_size                16384 # 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               1203 # total number of pages allocated
mem.page_mem                 4812k # total size of memory pages allocated
mem.ptab_misses             20663 # total first level page table misses
mem.ptab_accesses       3934367492 # total page table accesses
mem.ptab_miss_rate          0.0000 # first level page table miss rate
```

Comment:

From the question 1, we can obtain that:

```
-fetch:ifqsize           4 # instruction fetch queue size (in insts)
-decode:width            4 # instruction decode B/W (insts/cycle)
-issue:width             4 # instruction issue B/W (insts/cycle)
-commit:width            4 # instruction commit B/W (insts/cycle)
```

which is exactly the same as described.

Digging out the code from files, screenshots about changed options are:

```
621   opt_reg_int(odb, "-fetch:ifqsize", "instruction fetch queue size (in insts)",
622             &ruu_ifq_size, /* default */4,
623             /* print */TRUE, /* format */NULL);

694   opt_reg_int(odb, "-decode:width",
695             "instruction decode B/W (insts/cycle)",
696             &ruu_decode_width, /* default */4,
697             /* print */TRUE, /* format */NULL);

701   opt_reg_int(odb, "-issue:width",
702             "instruction issue B/W (insts/cycle)",
703             &ruu_issue_width, /* default */4,
704             /* print */TRUE, /* format */NULL);

717   opt_reg_int(odb, "-commit:width",
718             "instruction commit B/W (insts/cycle)",
719             &ruu_commit_width, /* default */4,
720             /* print */TRUE, /* format */NULL);
```

which indicates that these parameters are isolated values specifically for “sim-outorder.c”.

Option Name	Original Configuration	Changed Configuration
-ruu:size	16 # register update unit (RUU) size	16 # register update unit (RUU) size
RUU_count	1517615680 # cumulative RUU occupancy	1496773735 # cumulative RUU occupancy
RUU_fcount	41502815 # cumulative RUU full count	64799727 # cumulative RUU full count
ruu_occupancy	10.2408 # avg RUU occupancy (insn's)	11.1994 # avg RUU occupancy (insn's)
ruu_rate	1.7793 # avg RUU dispatch rate (insn/cycle)	1.9893 # avg RUU dispatch rate (insn/cycle)

Option Name	Original Configuration	Changed Configuration
ruu_latency	5.7554 # avg RUU occupant latency (cycle's)	5.6298 # avg RUU occupant latency (cycle's)
ruu_full	0.2801 # fraction of time (cycle's) RUU was full	0.4849 # fraction of time (cycle's) RUU was full
-lsq:size	8 # load/store queue (LSQ) size	8 # load/store queue (LSQ) size
LSQ_count	487418024 # cumulative LSQ occupancy	469444041 # cumulative LSQ occupancy
LSQ_fcount	14972254 # cumulative LSQ full count	16127229 # cumulative LSQ full count
lsq_occupancy	3.2891 # avg LSQ occupancy (insn's)	3.5125 # avg LSQ occupancy (insn's)
lsq_rate	1.7793 # avg LSQ dispatch rate (insn/cycle)	1.9893 # avg LSQ dispatch rate (insn/cycle)
lsq_latency	1.8485 # avg LSQ occupant latency (cycle's)	1.7657 # avg LSQ occupant latency (cycle's)
lsq_full	0.1010 # fraction of time (cycle's) LSQ was full	0.1207 # fraction of time (cycle's) LSQ was full
-res:ialu	4 # total number of integer ALU's available	4 # total number of integer ALU's available
-res:imult	1 # total number of integer multiplier/dividers available	1 # total number of integer multiplier/dividers available
-res:fpalu	4 # total number of floating point ALU's available	4 # total number of floating point ALU's available
-res:fpmult	1 # total number of floating point multiplier/dividers	1 # total number of floating point multiplier/dividers

This attempt shows that by broaden band-width alone, the results only change because of processing like issue have been accelerated.

Select related information from *.cfg files:

```
#
# default sim-outorder configuration
#
# instruction fetch queue size (in insts)
-fetch:ifqsize                  4
# instruction decode B/W (insts/cycle)
-decode:width                   4
# instruction issue B/W (insts/cycle)
-issue:width                    4
# register update unit (RUU) size
-ruu:size                      16
# load/store queue (LSQ) size
-lsq:size                       8
# total number of integer ALU's available
-res:ialu                       4
# total number of integer multiplier/dividers available
-res:imult                      1
# total number of floating point ALU's available
-res:fpalu                      4
# total number of floating point multiplier/dividers available
-res:fpmult                     1
#
# this config matches "-b 2048 -R 16 -L 8 -d dl1:128:32:4:l -i il1:512:32:1:l"
# options on the previous (0.1.0) version of sim-outorder, used for regression
# testing
#
```

```
# instruction fetch queue size (in insts)
-fetch:ifqsize                 4
# instruction decode B/W (insts/cycle)
-decode:width                  4
# instruction issue B/W (insts/cycle)
-issue:width                   4
# register update unit (RUU) size
-ruu:size                     16
# load/store queue (LSQ) size
-lsq:size                     8
# total number of integer ALU's available
-res:ialu                     4
# total number of integer multiplier/dividers available
-res:imult                    1
# total number of floating point ALU's available
-res:fpalu                    4
# total number of floating point multiplier/dividers available
-res:fpmult                   1



#
# relax.cfg     - relax resource restrictions, useful for debugging dependence  related problems.
#
-res:ialu       8 # total number of integer ALU's available
-res:imult      8 # total number of integer multiplier/dividers available
-res:memport    8 # total number of memory system ports available (to CPU)
-res:fpalu      8 # total number of floating point ALU's available
-res:fpmult     8 # total number of floating point multiplier/dividers available
-fetch:ifqsize  16 # instruction fetch queue size (in insts)
-decode:width   16 # instruction decode B/W (insts/cycle)
-issue:width    16 # instruction issue B/W (insts/cycle)
-ruu:size       128 # register update unit (RUU) size
-lsq:size       64 # load/store queue (LSQ) size
```

Take all observation results above into consideration, for the first factor, increase all the pipeline bandwidth, and for the second one, keep the original bandwidth from question 1 and enhance hardwares respectively.