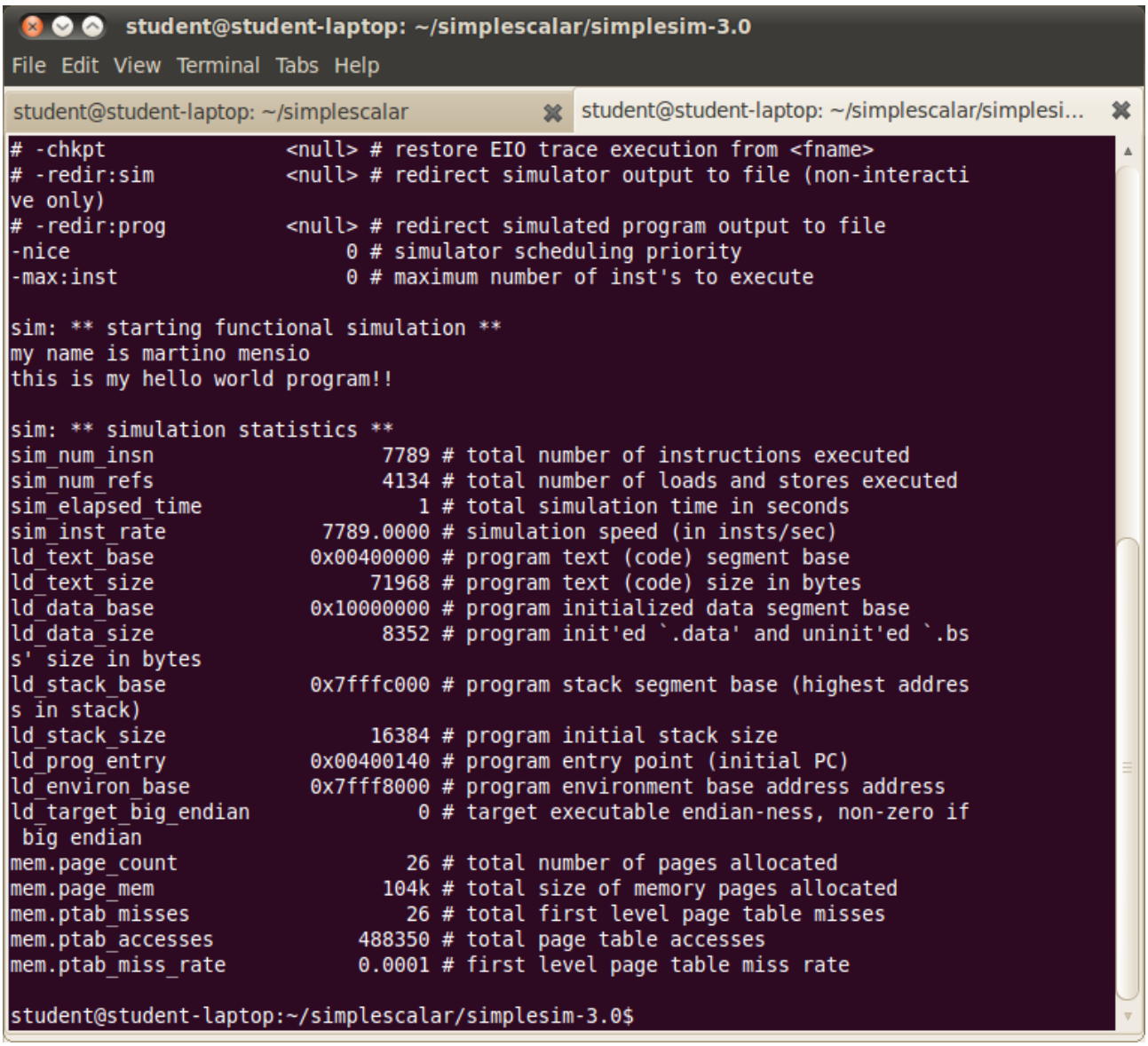
Computer Architectures
2nd part labs – lab 2

1) Sim-safe run



2) Sim-outorder run

```
student@student-laptop: ~/simplescalar/simplesim-3.0
File  Edit  View  Terminal  Tabs  Help
student@student-laptop: ~/simplescalar | student@student-laptop: ~/simplescalar/simplesim-3.0

sim: ** starting performance simulation **
my name is martino mensio
this is my hello world program!!

sim: ** simulation statistics **
sim_num_insn                  7789 # total number of instructions committed
sim_num_refs                  4134 # total number of loads and stores committed
sim_num_loads                  635 # total number of loads committed
sim_num_stores           3499.0000 # total number of stores committed
sim_num_branches              1060 # total number of branches committed
sim_elapsed_time                 1 # total simulation time in seconds
sim_inst_rate            7789.0000 # simulation speed (in insts/sec)
sim_total_insn                8673 # total number of instructions executed
sim_total_refs                4381 # total number of loads and stores executed
sim_total_loads                792 # total number of loads executed
sim_total_stores         3589.0000 # total number of stores executed
sim_total_branches            1207 # total number of branches executed
sim_cycle                    14146 # total simulation time in cycles
sim_IPC                     0.5506 # instructions per cycle
sim_CPI                     1.8162 # cycles per instruction
sim_exec_BW                 0.6131 # total instructions (mis-spec + committed) per cycle
sim_IPB                     7.3481 # instruction per branch
IFQ_count                    12588 # cumulative IFQ occupancy
IFQ_fcount                    2848 # cumulative IFQ full count
ifq_occupancy               0.8899 # avg IFQ occupancy (insn's)
ifq_rate                    0.6131 # avg IFQ dispatch rate (insn/cycle)
ifq_latency                 1.4514 # avg IFQ occupant latency (cycle's)
ifq_full                    0.2013 # fraction of time (cycle's) IFQ was full
RUU_count                    39425 # cumulative RUU occupancy
RUU_fcount                     185 # cumulative RUU full count
ruu_occupancy               2.7870 # avg RUU occupancy (insn's)
ruu_rate                    0.6131 # avg RUU dispatch rate (insn/cycle)
ruu_latency                 4.5457 # avg RUU occupant latency (cycle's)
ruu_full                    0.0131 # fraction of time (cycle's) RUU was full
LSQ_count                    21039 # cumulative LSQ occupancy
LSQ_fcount                    1629 # cumulative LSQ full count
lsq_occupancy               1.4873 # avg LSQ occupancy (insn's)
lsq_rate                    0.6131 # avg LSQ dispatch rate (insn/cycle)
lsq_latency                 2.4258 # avg LSQ occupant latency (cycle's)
lsq_full                    0.1152 # fraction of time (cycle's) LSQ was full
sim_slip                     69299 # total number of slip cycles
avg_sim_slip                8.8970 # the average slip between issue and retirement
bpred_bimod.lookups           1248 # total number of bpred lookups
bpred_bimod.updates           1060 # total number of updates
bpred_bimod.addr_hits          758 # total number of address-predicted hits
bpred_bimod.dir_hits           897 # total number of direction-predicted hits (includes addr-hits)
bpred_bimod.misses             163 # total number of misses
bpred_bimod.jr_hits             64 # total number of address-predicted hits for JR's
bpred_bimod.jr_seen             73 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP           0 # total number of address-predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP           1 # total number of non-RAS JR's seen
bpred_bimod.bpred_addr_rate    0.7151 # branch address-prediction rate (i.e., addr-hits/updates)
bpred_bimod.bpred_dir_rate    0.8462 # branch direction-prediction rate (i.e., all-hits/updates)
bpred_bimod.bpred_jr_rate    0.8767 # JR address-prediction rate (i.e., JR addr-hits/JRs seen)
bpred_bimod.bpred_jr_non_ras_rate.PP    0.0000 # non-RAS JR addr-pred rate (ie, non-RAS JR hits/JRs seen)
bpred_bimod.retstack_pushes          94 # total number of address pushed onto ret-addr stack
bpred_bimod.retstack_pops            77 # total number of address popped off of ret-addr stack
bpred_bimod.used_ras.PP              72 # total number of RAS predictions used
bpred_bimod.ras_hits.PP              64 # total number of RAS hits
bpred_bimod.ras_rate.PP    0.8889 # RAS prediction rate (i.e., RAS hits/used RAS)
il1.accesses                  9167 # total number of accesses
il1.hits                      8643 # total number of hits
il1.misses                     524 # total number of misses
il1.replacements               199 # total number of replacements
il1.writebacks                   0 # total number of writebacks
il1.invalidations                0 # total number of invalidations
il1.miss_rate               0.0572 # miss rate (i.e., misses/ref)
```

```
il1.repl_rate                0.0217 # 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                 4185 # total number of accesses
dl1.hits                     3752 # total number of hits
dl1.misses                    433 # total number of misses
dl1.replacements                8 # total number of replacements
dl1.writebacks                  7 # total number of writebacks
dl1.invalidations               0 # total number of invalidations
dl1.miss_rate                0.1035 # miss rate (i.e., misses/ref)
dl1.repl_rate                0.0019 # replacement rate (i.e., repls/ref)
dl1.wb_rate                  0.0017 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate                 0.0000 # invalidation rate (i.e., invs/ref)
ul2.accesses                  964 # total number of accesses
ul2.hits                      464 # total number of hits
ul2.misses                    500 # 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                0.5187 # 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                9167 # total number of accesses
itlb.hits                    9155 # 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               0.0013 # 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                4191 # total number of accesses
dtlb.hits                    4183 # total number of hits
dtlb.misses                     8 # 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               0.0019 # 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                71968 # program text (code) size in bytes
ld_data_base           0x10000000 # program initialized data segment base
ld_data_size                 8352 # 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                 26 # total number of pages allocated
mem.page_mem                  104k # total size of memory pages allocated
mem.ptab_misses                26 # total first level page table misses
mem.ptab_accesses          518738 # total page table accesses
mem.ptab_miss_rate         0.0001 # first level page table miss rate

student@student-laptop:~/simplescalar/simplesim-3.0$
```

3) Running test-pisa programs

	sim_num_insn	sim_cycle	sim_IPC	sim_CPI
test-fmath	52387	98854	0.5299	1.8870
test-llong	27807	54623	0.5091	1.9644
test-lswlr	8734	17867	0.4888	2.0457
test-math	213487	386448	0.5524	1.8102
test-printf	1729323	3167231	0.5460	1.8315

4) Performance comparison

Conf	FP resources	Fetch speed	Fetch ifqsize	Lsq size	Ruu size	Issue width	Commit width	In order	test-fmath	test-llong	test-lswlr	test-math	test-printf
1	1	1	1	2	2	1	1	Y	1.8870	1.9644	2.0457	1.8102	1.8315
2	1	1	1	2	4	1	1	N	1.5298	1.6085	1.8074	1.4353	1.4485
3	4	1	1	2	4	1	1	N	1.5298	1.6085	1.8074	1.4353	1.4485
4	4	1	1	2	8	1	1	N	1.5169	1.5938	1.8063	1.4227	1.4373
5	1	4	4	4	8	4	4	N	1.1437	1.1923	1.1895	1.0990	1.1231
6	4	4	4	4	16	4	4	N	1.1244	1.1787	1.1861	1.0739	1.1140
7	4	8	8	8	64	4	4	N	1.0851	1.0568	1.1595	1.0551	1.0291

From 1 to 2, only ruu-size is changed: the result is a speedup of all tests because probably the register update unit was the bottleneck.

From 2 to 3 there is no improvement adding more FP resources. These units (fpalu and fpmult) in this configuration are not overloaded.

The biggest improvement is from 4 to 5, where the fetch-speed, fetch-ifqsize, lsq-size, issue-width, commit-width are enhanced.