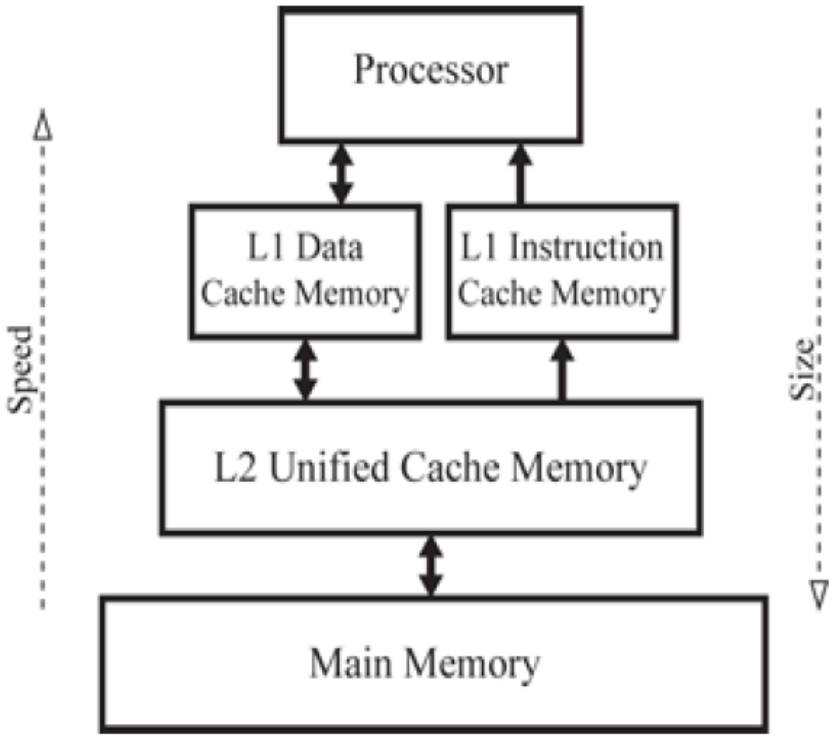
Computer Architecture Lab

Lab12 – Week #12

MPLAB Media Processing Laboratory

컴퓨터 공학과 이성원 교수님 연구실

Cache



- **If accessed data is present in upper level**

Hit: access satisfied by upper level

Hit ratio: hits/accesses

- **If accessed data is absent**

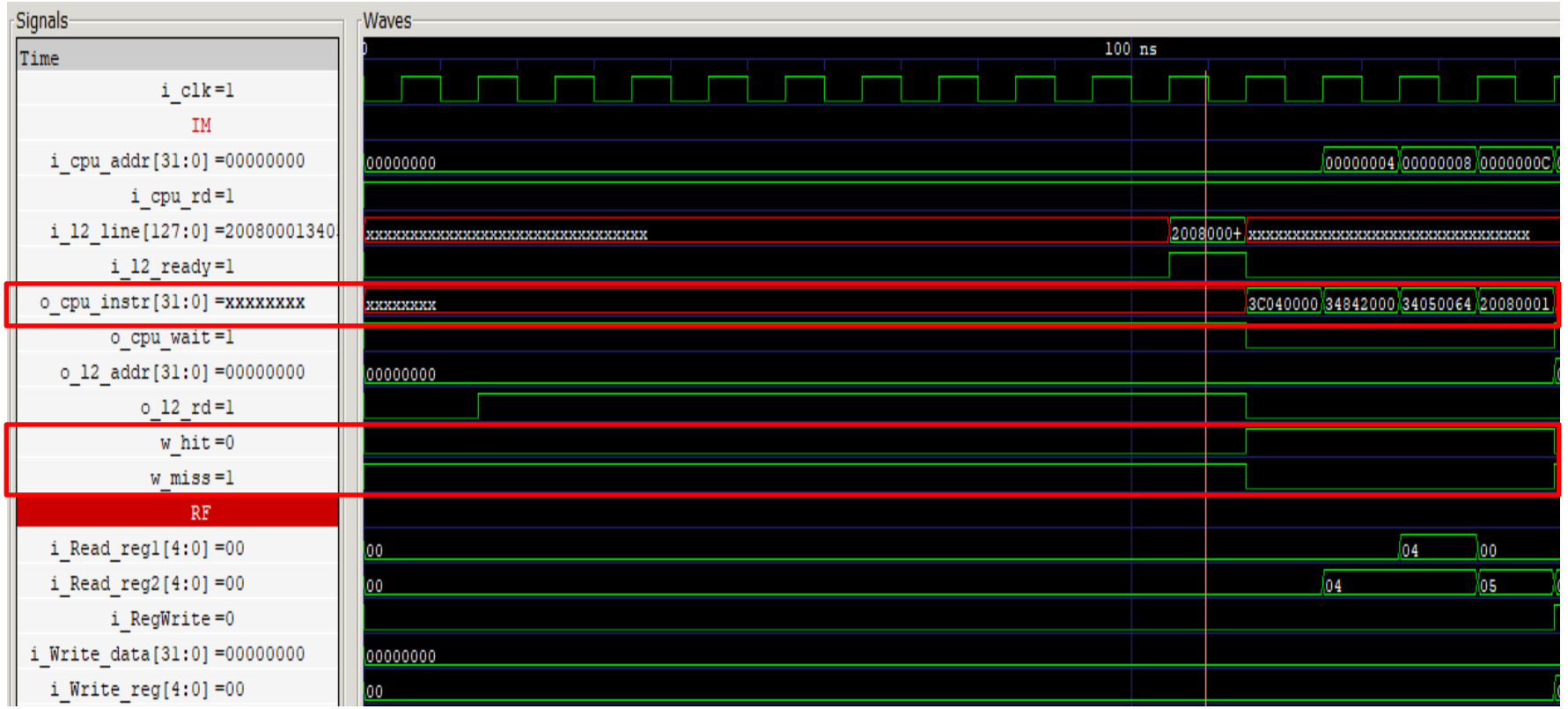
Miss: block copied from lower level

Time taken: miss penalty

Miss ratio: 1 – hit ratio

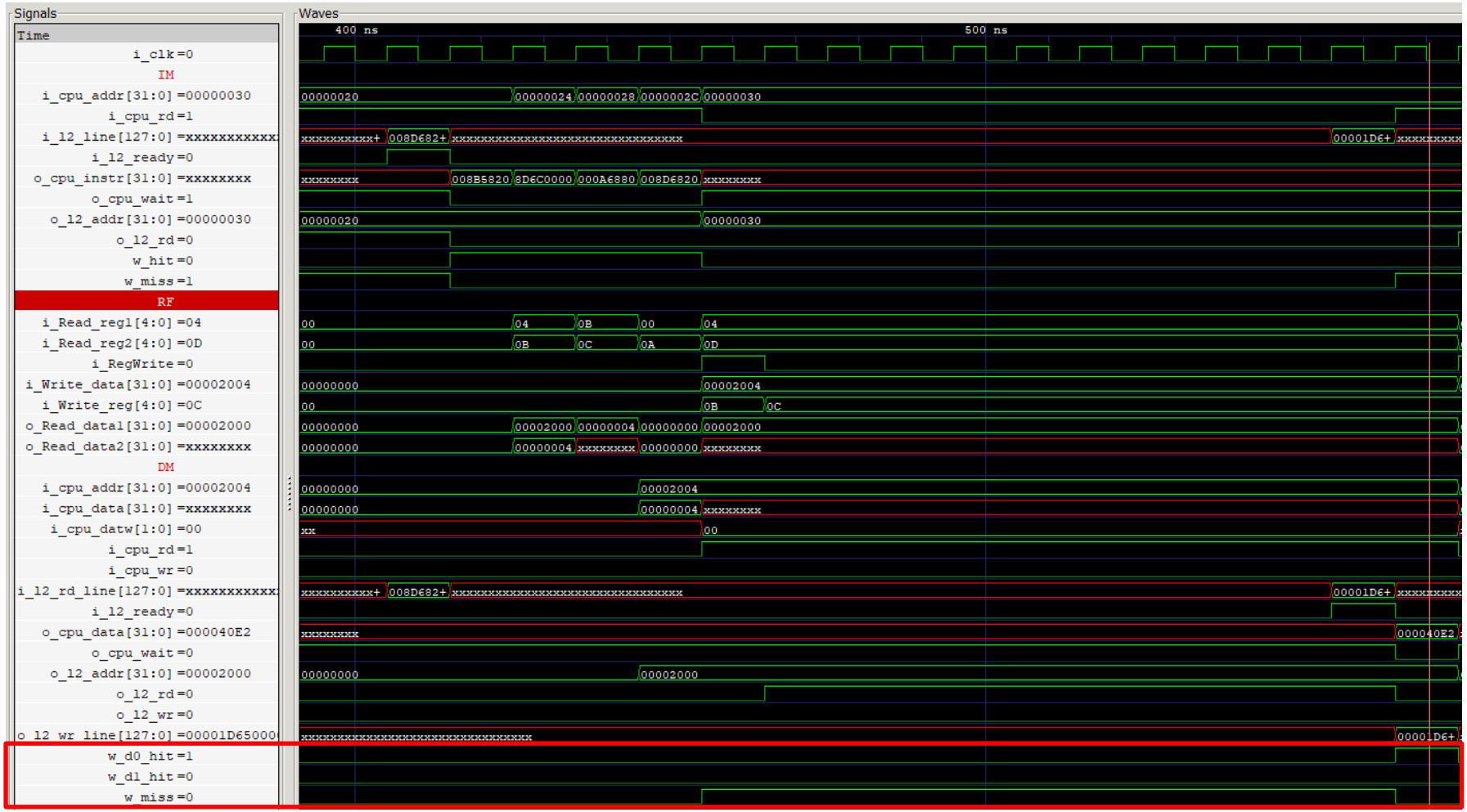
Insertion sort, random access

- How I/D L1 hit operates, how I/D L1 miss operates



Insertion sort, random access

- How I/D L1 hit operates, how I/D L1 miss operates



Insertion sort, random access

- When and how main memory(L2) operates

CINT95 Benchmarks

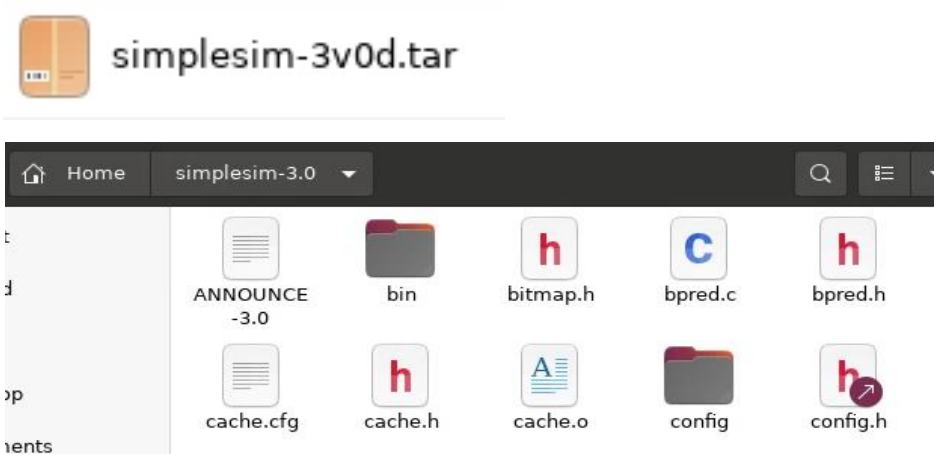
Table 1 – CINT95 Benchmarks

Benchmark	Application Area	Specific Task
go	Game playing	Plays the game Go against itself
m88ksin	Simulation	Simulates the Motorola 88100 processor running Dhrystone and a memory test program
swim	Weather prediction	Solves shallow water equations using finite difference approximations

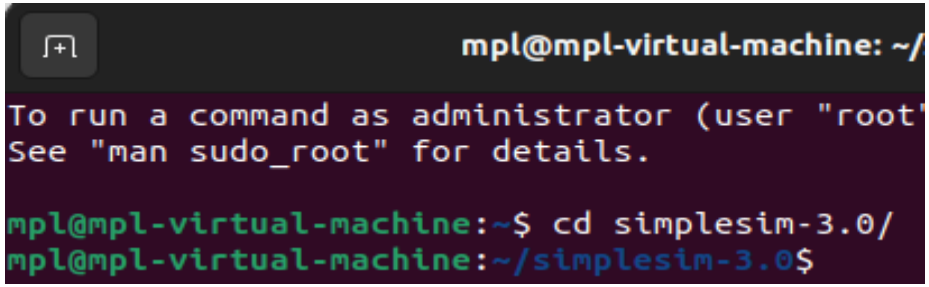
- Change cache configuration
  - ✓ Unified cache / Separate cache
  - ✓ L1 cache size / L2 cache size
  - ✓ Large block size / Small block size
  - ✓ Direct-mapped / Set-associative

Setting SimpleScalar

1. Unzip simplesim-3v0e.tgz file at Home directory



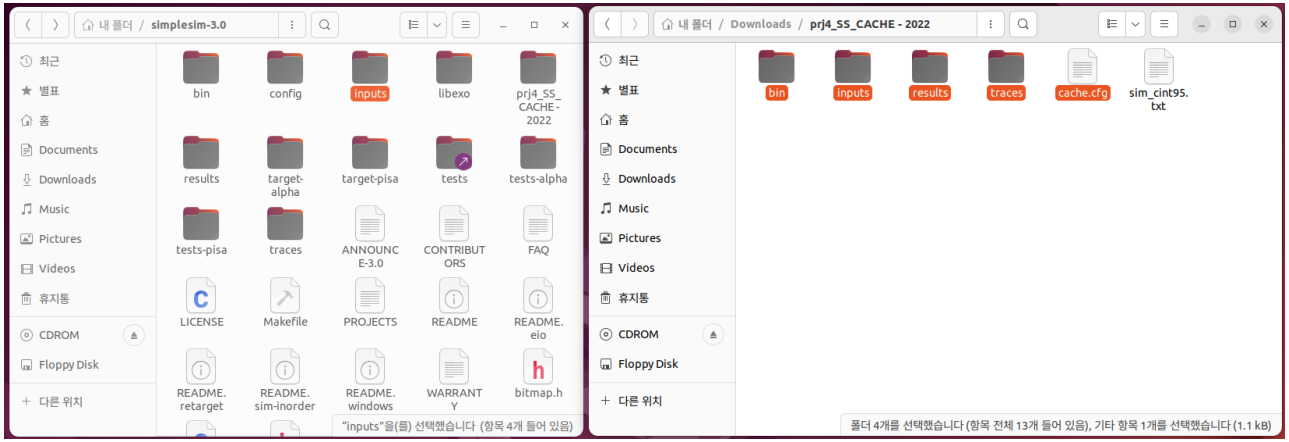
2. (Terminal) cd simplesim-3.0



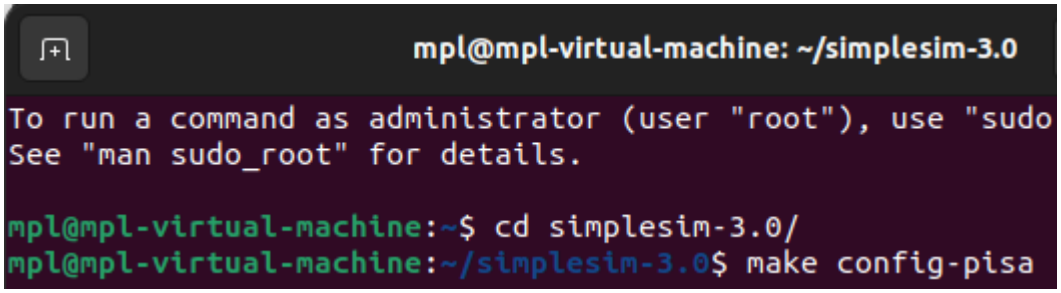
Setting SimpleScalar

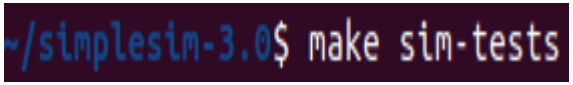
3. Download the Benchmark file in klas

4. Unzip the prj4_SS_CACHE - 2025.zip, Copy and paste in the simplesim-3.0 directory



5. make config-pisa-> make-> make sim-tests





Write configuration file

- To seek help, type the following command
  - ✓ ./sim-cache -h

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

Write configuration file

- Cache setting : gedit ./config/mycache.cfg

[bytes]	Cache Level 1	Cache Level 2
# of sets	32	256
Block size	16	64
Associativity	1	1
Total Cache size	512	16,384

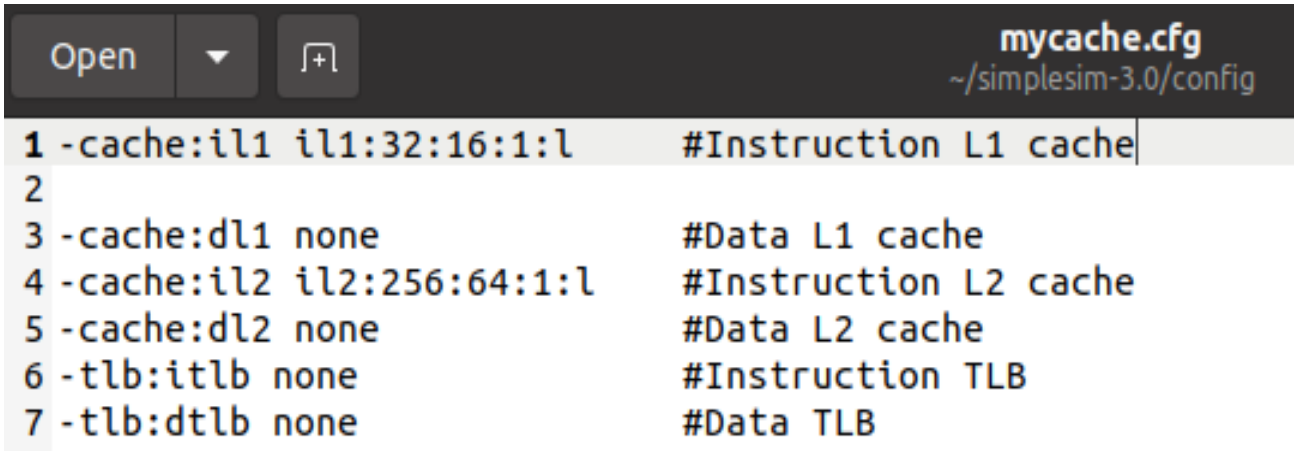
```
-cache:il1 il1:32:16:1:l

-cache:dl1 none                 #Data L1 cache
-cache:il2 il2:256:64:1:l       #Instruction L2 cache
-cache:dl2 none                 #Data L2 cache
-tlb:itlb none                  #Instruction TLB
-tlb:dtlb none                  #Data TLB
```

Typing these

Write configuration file (example)

- Cache setting : gedit ./config/mycache.cfg



How to simulate?

```
mpl@mpl-virtual-machine:~/simplesim-3.0$ ~/simplesim-3.0/sim-cache -config ~/sim
plesim-3.0/config/mycache.cfg bin/cc1.little.ss -O inputs/1stmt.i -o results/cc1
.out 2> traces/cc1.trace
```

- ./sim-cache –config configfile path application path

- go
  - ✓ ~/simplesim-3.0/sim-cache -config ~/simplesim-3.0/config/mycache.cfg bin/go.little.ss 20 9 inputs/2stone9.in 2> traces/go.trace > results/go.out

- m88ksin
  - ✓ ~/simplesim-3.0/sim-cache -config ~/simplesim-3.0/config/mycache.cfg bin/m88ksim.little.ss -c < inputs/ctl.raw inputs/dcrand.lit 2> traces/m88ksim.trace > results/m88ksim.out

- swim
  - ✓ ~/simplesim-3.0/sim-cache -config ~/simplesim-3.0/config/mycache.cfg bin/swim.little.ss < inputs/swim.in 2> traces/swim.trace > results/swim.out

Example of result analysis

- Miss rate
  - ✓ check at .trace file

```
sim: ** simulation statistics **
sim_num_insn                362765 # total number of instructions executed
sim_num_refs                104761 # total number of loads and stores executed
sim_elapsed_time                 1 # total simulation time in seconds
sim_inst_rate           362765.0000 # simulation speed (in insts/sec)
il1.accesses                362765 # total number of accesses
il1.hits                    247961 # total number of hits
il1.misses                  114804 # total number of misses
il1.replacements            114772 # total number of replacements
il1.writebacks                   0 # total number of writebacks
il1.invalidations                0 # total number of invalidations
il1.miss_rate               0.3165 # miss rate (i.e., misses/ref)
il1.repl_rate               0.3164 # 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)
il2.accesses                114804 # total number of accesses
il2.hits                    105675 # total number of hits
il2.misses                    9129 # total number of misses
il2.replacements              8887 # total number of replacements
il2.writebacks                   0 # total number of writebacks
il2.invalidations                0 # total number of invalidations
il2.miss_rate               0.0795 # miss rate (i.e., misses/ref)
il2.repl_rate               0.0774 # replacement rate (i.e., repls/ref)
```

$$miss\ rate_{L1} = \frac{114804}{362765} = 0.316$$

$$miss\ rate_{L2} = \frac{9129}{114804} = 0.079$$

Example of result analysis

- Access lower memory level

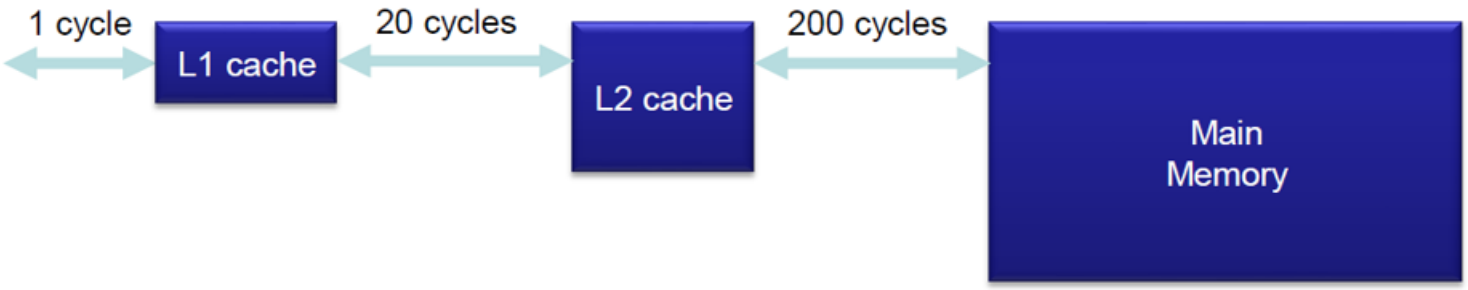
```
sim: ** simulation statistics **
sim_num_insn                362765 # total number of instructions executed
sim_num_refs                104761 # total number of loads and stores executed
sim_elapsed_time                 1 # total simulation time in seconds
sim_inst_rate           362765.0000 # simulation speed (in insts/sec)
il1.accesses                362765 # total number of accesses
il1.hits                    247961 # total number of hits
il1.misses                  114804 # total number of misses
il1.replacements            114772 # total number of replacements
il1.writebacks                   0 # total number of writebacks
il1.invalidations                0 # total number of invalidations
il1.miss_rate               0.3165 # miss rate (i.e., misses/ref)
il1.repl_rate               0.3164 # 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)
il2.accesses                114804 # total number of accesses
il2.hits                    105675 # total number of hits
il2.misses                    9129 # total number of misses
il2.replacements              8887 # total number of replacements
il2.writebacks                   0 # total number of writebacks
il2.invalidations                0 # total number of invalidations
il2.miss_rate               0.0795 # miss rate (i.e., misses/ref)
il2.repl_rate               0.0774 # replacement rate (i.e., repls/ref)
```

If L1 miss, L2 access

If L2 miss, Main memory access

Assumption for AMAT

- L1 cache access time is 1 cycle
- L2 cache access time is 20 cycles
- Main memory access time is 200 cycles



- Average memory-access time(AMAT) = (Hit time) + (Average miss time)
  - ✓ (Average miss time) = (Miss rate) x (Miss penalty)
  - ✓ AMAT of L2 cache
    - AMAT = $L1_{Hit\_time} + L1_{Miss\_rate} \times L1_{miss\_penalty}$
      $= L1_{Hit\_time} + L1_{Miss\_rate} \times (L2_{Hit_{time}} + L2_{miss\_rate} \times L2_{miss\_penalty})$

Unified vs splits

- Unified cache
  - ✓ Data and instructions are stored together
    - Von Neuman architecture
  - ✓ Instructions and data that might map into the same storage locations in a unified cache
  - ✓ Use following, when you make configuration file
    - Ex)

```
-cache:il1 dl1
-cache:dl1 ul1:64:16:1:l

-cache:il2 none          #Instruction L2 cache
-cache:dl2 none          #Data L2 cache
-tlb:itlb none           #Instruction TLB
-tlb:dtlb none           #Data TLB
```

- Split cache
  - ✓ Data and instructions are stored in separate data and instruction caches
    - Harvard architecture
  - ✓ Allows the processor to fetch instructions from instruction cache and data from data cache simultaneously

Simulation 1. Unified vs splits

- Block size = 16, Associativity = 1

- Split cache
  - ✓ Each of instruction cache, Data cache : Number of sets / 2

- Simulate the performance of the cache under the following condition

  - ✓ ./sim-cache –config configfile path application path
  - ✓ Ex) ~/simplesim-3.0/sim-cache -config ~/simplesim-3.0/config/mycache.cfg bin/vortex.little.ss inputs/vortex.raw 2> traces/vortex.trace

# of Sets	Unified cache Miss rate	Unified cache AMAT	Split cache		Split cache AMAT
			Inst. Miss rate	Data Miss rate	
64					
128					
256					
512					

Simulation 2. L1/L2 size

- Block size = 16, Associativity = 1

  - ✓ Use sim-cache to simulate the performance of the cache under the following condition
  - ✓ Do calculate, once each for instruction-miss rate, data miss rate, inst & data unified cache and AMAT

L1I/L1D/L2U	Inst.Miss rate	Data.Miss rate	Unified Cache Miss rate	AMAT
8/8/1024				
16/16/512				
32/32/256				
64/64/128				
128/128/0				

Simulation 3. Associativity

- Block size = 16
  - ✓ Use sim-cache to simulate the performance of the cache under the following condition

# of Sets	Split Cache Miss rate / AMAT							
	1-way		2-way		4-way		8-way	
64								
128								
256								
512								
1024								
2048								

Simulation 4. Block size

- Number of Sets = 512, Associativity = 1
  - ✓ Use sim-cache to simulate the performance of the cache under the following condition

Block size	Unified cache Miss rate	AMAT
16		
64		
128		
256		
512		

Performance

- Assumption
  - ✓ L1 cache access time is 1 cycle, L2 cache access time is 20 cycles, and Main Memory access time is 200 cycles
  - ✓ If increase twice cache size, cycle time increase 4%
  - ✓ If increase twice associativity, cycle time increase 2%
- Find the cache configuration suitable for CINT95 program, provided program using AMAT

Thank You