CDA5105/CDA4150 - Spring 2026
Assignment 2
Memory Hierarchy Simulator

Your assignment is to write a program that simulates the behavior of a memory hierarchy. The hierarchy will include an instruction translation look-aside buffer (ITLB), data translation look-aside buffer (DTLB), page table (PT), instruction cache (IC), and data cache (DC). The program will read a trace of references from standard input and produce statistics about the trace to standard output. The trace of references has the following format:

<streamtype>:<accesstype>:<hexaddress>

where <streamtype> can be the characters:

I - indicates a reference to an instruction
D - indicates a reference to data

<accesstype> can be the characters:

R - indicates a read access
W - indicates a write access

and <hexaddress> is the starting address of the reference expressed as a hexadecimal number. Note that an I (instruction) reference should always have an R (read) accesstype. I have provided an example trace file, *~whalley/cda5155exec/trace.dat*, which is in the appropriate format. You can read information about a reference using the C scanf format **"%c:%c:%x\n"**.

Before processing the trace file, the memory hierarchy simulator should first determine the characteristics of the memory hierarchy to be simulated. This is accomplished by reading a configuration file in the current directory called *trace.config* that specifies the configuration for the ITLB, DTLB, PT, IC, and DC. I have provided an example configuration file in *~whalley/cda5155exec/trace.config* that indicates the required format (note that the values after the colons are the fields that can change). The maximum number of sets that can be specified for the ITLB and DTLB is 256. The maximum number of sets for the IC and DC is 8192. The maximum associativity level is 8 for the TLBs and caches. The maximum number of virtual pages is 8192. The maximum number of physical pages is 1024. The number of sets for the TLBs and caches, the number of virtual pages, the page size, and line size for the caches should all be powers of two. The data line size should be at least 8 and the instruction line size should be at least 4. Your simulator should use an LRU replacement algorithm for the TLBs, caches, and page table. Physical pages should be initially allocated from 0..*n*-1, where *n* is the number of physical pages. The simulator will employ either a no write-allocate and write-through policy or a write-allocate and write-back policy for the DC. When a page fault occurs, you should invalidate the TLB entries and the cache lines that are associated with the page that was replaced. In addition, you will have the option in the configuration file for turning off virtual to physical address translation by assuming that the trace contains physical instead of virtual addresses. Finally, you also have the option of disabling the TLBs.

You can access my executable for the simulation in */home/faculty/whalley/cda5155exec/hierarchy.exe*. Your output should match my format exactly. You will first print

a. the configuration of the memory hierarchy

Next, you will print

b. information about each reference

The fields to be printed for each reference are the virtual address, virtual page number, page offset, reference type, TLB tag, TLB index, TLB result, PT result, physical page number, cache tag, cache index, and cache result. The specified format to print these fields for the full memory hierarchy simulation is given in the form of a C printf format specification below:

"%08x %7x %6x %4s %7x %5x %4s %4s %6x %7x %5x %4s"

Printing this information will also help you debug problems with your simulator. When virtual to physical address

translation or the TLBs are disabled, then the output fields that are not relevant should be left blank.

Next, you will print

c. the number of hits
d. the number of misses
e. hit ratio

for each level of the hierarchy. Finally, you should also indicate the

f. the number of accesses that were reads
g. the number of accesses that were writes
h. the ratio of accesses that were reads
i. the number of accesses for instructions
j. the number of accesses for data
k. the ratio of accesses that were instructions
l. the total number of memory references
m. the total number of disk references

You should upload the source code for your program in Canvas before class starts on February 3. The header comments should identify you, the assignment, and the command you used to compile your program. You should also use readable and consistent indentation and comments throughout the program. Your program should be able to compile and execute on *linprog.cs.fsu.edu*. Below is an example *configuration* file. On the following pages are the *trace data* file and the corresponding *output*.

configuration

```
Instruction TLB configuration
Number of sets: 2
Set size: 1

Data TLB configuration
Number of sets: 4
Set size: 2

Page Table configuration
Number of virtual pages: 64
Number of physical pages: 4
Page size: 256

Instruction Cache configuration
Number of sets: 4
Set size: 1
Line size: 16

Data Cache configuration
Number of sets: 2
Set size: 2
Line size: 8
Write through/no write allocate: y

Virtual addresses: y
TLBs: y
```

trace data

```
I:R:c84
D:R:81c
I:R:14c
I:R:c84
I:R:400
I:R:148
I:R:144
I:R:c80
I:R:008
```

output

```
Instruction TLB contains 2 sets.
Each set contains 1 entries.
Number of bits used for the index is 1.

Data TLB contains 4 sets.
Each set contains 2 entries.
Number of bits used for the index is 2.

Number of virtual pages is 64.
Number of physical pages is 4.
Each page contains 256 bytes.
Number of bits used for the page table index is 6.
Number of bits used for the page offset is 8.

I-cache contains 4 sets.
Each set contains 1 entries.
Each line is 16 bytes.
Number of bits used for the index is 2.
Number of bits used for the offset is 4.

D-cache contains 2 sets.
Each set contains 2 entries.
Each line is 8 bytes.
The cache uses a no write-allocate and write-through policy.
Number of bits used for the index is 1.
Number of bits used for the offset is 3.

The addresses read in are virtual addresses.
```

Virtual Address	Virtual Page #	Page Offset	Ref Type	TLB Tag	TLB Index	TLB Ref	PT Ref	Phys Page #	Cache Tag	Cache Index	Cache Ref
00000c84	c	84	inst	6	0	miss	miss	0	2	0	miss
0000081c	8	1c	data	2	0	miss	miss	1	11	1	miss
0000014c	1	4c	inst	0	1	miss	miss	2	9	0	miss
00000c84	c	84	inst	6	0	hit	none	0	2	0	miss
00000400	4	0	inst	2	0	miss	miss	3	c	0	miss
00000148	1	48	inst	0	1	hit	none	2	9	0	miss
00000144	1	44	inst	0	1	hit	none	2	9	0	hit
00000c80	c	80	inst	6	0	miss	hit	0	2	0	miss
00000008	0	8	inst	0	0	miss	miss	1	4	0	miss

```
Simulation statistics

itlb hits        : 3
itlb misses      : 5
itlb hit ratio   : 0.375000

dtlb hits        : 0
dtlb misses      : 1
dtlb hit ratio   : 0.000000

pt hits          : 1
pt faults        : 5
pt hit ratio     : 0.166667

ic hits          : 1
ic misses        : 7
ic hit ratio     : 0.125000

dc hits          : 0
dc misses        : 1
dc hit ratio     : 0.000000

Total reads      : 9
Total writes     : 0
Ratio of reads   : 1.000000

Total inst refs  : 8
Total data refs  : 1
Ratio of insts   : 0.888889

main memory refs : 14
disk refs        : 5
```

Assignment 2 Suggestions

I recommend that you accomplish assignment 2 in the following stages. The test cases will also follow these stages. For each step compare your output with the output generated by my executable using the Unix *diff* command to ensure your output exactly matches mine.

(1) Copy the *~whalley/cda5155exec/trace.config* file to your directory. Read in this configuration file, calculate the number of index and offset bits for the different portions of the memory hierarchy, and print out the configuration information. Change the information in the configuration file and retest to ensure that the correct configuration information is printed.

(2) Take the following actions within the trace.config file. Mark 'n' after "Virtual addresses:" to indicate that the simulator will not perform virtual to physical address translation. Mark 'n' after "TLBs:" to indicate that the TLBs will be disabled. Implement the simulation of an instruction cache for a direct-mapped organization (set size 1).

(3) Enhance the instruction cache simulation to deal with associativity levels that are greater than 1.

(4) Mark 'y' after "Write through/no write allocate:" to indicate that this write policy will be used. Implement the simulation of the data cache with this write policy.

(5) Enhance the data cache simulation to deal with associativity levels that are greater than 1, where all references are still reads.

(6) Mark 'n' after "Write through/no write allocate:" to indicate that a write back/write allocate policy will be used. Implement the simulation of the data cache with this write policy.

(7) Mark 'y' after "Virtual addresses:" in the configuration file to indicate that you will be processing virtual addresses. Mark 'n' after "TLBs:" to indicate that the TLBs will be disabled. Implement the simulation of the page table.

(8) Mark 'y' after "Virtual addresses:" in the configuration file to indicate that you will be processing virtual addresses. Mark 'y' after "TLBs:" to indicate that the TLBs will be enabled. Implement the simulation of the instruction and data TLBs.

(9) Increment counters during the simulation and print out the summary statistics at the end of the simulation.

(10) Test for invalidating a TLB entry and/or lines in the DC and IC cache when a page is replaced. Check the trace.log file in your current directory that is produced by *~whalley/cda5155exec/hierarchy.exe* to see when these events occur.

(11) Test your solution with larger sets of trace data to see if your output matches mine.

CDA4150 students are not required to implement steps 7, 8, and 10. However, these students will receive extra credit if they do so.