Programming Project #1
Report

Rushikesh Khamkar

Steps Used For Excution

With my eRaider ID and password, I entered the system and made a connection to the HPCC by doing the following actions:

1. Log in into the system using eRaider ID and password
   ```
   ssh -J rkhamkar@ssh.ttu.edu rkhamkar@login.hpcc.ttu.edu
   ```

2. Clone code from the given github link
   ```
   git clone https://github.com/githubyongchen/CS5375.git
   ```

3. cd CS5375/

4. Source file cachesim.c is compiled using the below command which generates a binary, executable file, cachesim
   ```
   gcc cachesim.c -o cachesim
   ```

5. To run the simulator below command is used. Here, argument used is direct to let simulator know it is the direct-mapped cache.
   ```
   ./cachesim direct trace_for_students/trace.stream
   ```

6. To run the simulator below command is used. Here, argument used is fully to let simulator know it is the fully associative cache.
   ```
   ./cachesim fully trace_for_students/trace.stream
   ```

7. To run the simulator below command is used. Here, argument used is n-way to let simulator know it is the n-way set associative cache.
   ```
   ./cachesim n-way trace_for_students/trace.stream
   ```

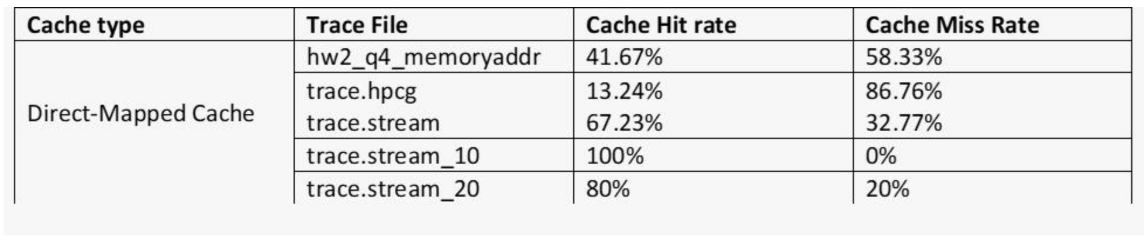
8. vi cachesim.c - to edit code

Part 1. Direct-mapped cache

Every addressed location in main memory that has a direct mapping cache corresponds to a single position in the cache memory. The fact that main memory is substantially bigger than cache memory means that several addresses in main memory correspond to the same spot in cache memory. A line would require 9 bits to be uniquely identifiable in a cache with 512 lines, for example. T tag bits, l line bits, and w word bits are the three components that make up an address in direct mapping. The word bits, which uniquely identify a particular word within a block of memory, are the least important bits.

Miss Rate and Hit Rate is calculated using below formula:
1. totalMemoryAccess = d_cache.hits + d_cache.misses
2. miss_rate = (d_cache.misses / totalMemoryAccess) * 100
3. hit_rate = (d_cache.hits / totalMemoryAccess) * 100

Results of Simulation of Direct mapped associative cache -



Cache type	Trace File	Cache Hit rate	Cache Miss Rate
Direct-Mapped Cache	hw2_q4_memoryaddr	41.67%	58.33%
	trace.hpcg	13.24%	86.76%
	trace.stream	67.23%	32.77%
	trace.stream_10	100%	0%
	trace.stream_20	80%	20%

<u>Part 2. Fully associative and n-way set associative cache</u>

Fully Associative Cache -

A method of cache mapping known as "fully associative mapping" enables mapping of the main memory block to a freely accessible cache line. A completely associative cache would also allow data to be stored in any cache block. No memory address would be compelled to reside in a specific block. Instead of forcing each memory address into a specific block, a fully associative cache allows data to be stored in any cache block.

Results of Simulation of Fully set associative cache -

I have executed fully associative cache with different trace files, and I got different results for the given inputs for Cache Hit and Miss Rate -

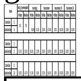
Inputs		hw2_q4_memoryaddr		trace.hpcg		trace.stream		trace.stream_10		trace.stream_20	
		Hit Rate	Miss Rate	Hit Rate	Miss Rate	Hit Rate	Miss Rate	Hit Rate	Miss Rate	Hit Rate	Miss Rate
Cache Size	32	25%	75%								
Cache Line Size	16			85.88%	14.12%	32.60%	67.40%	0.00%	100.00%	15.00%	85.00%
Cache Size	32										
Cache Line Size	32	41.67%	58.33%	86.61%	13.39%	33.05%	66.95%	0.00%	100.00%	20.00%	80.00%
Cache Size	32										
Cache Line Size	128	58.33%	41.67%	91.34%	8.66%	33.35%	66.65%	0.00%	100.00%	20.00%	80.00%

N-Way Associative Cache -

Conflicts are reduced by an N-way set associative cache by offering N blocks in each set where data that maps to that set might be found. Although there are N blocks in a set, each memory address still translates to a particular set. Consequently, a one-way set associative cache is also known as a direct mapped cache. A cache that is divided into units known as a set is known as an n-way set associative cache. And n-blocks can fit in each set.

Results of Simulation of N-way set associative cache -

I have executed N-way associative cache with different trace file and I got different results for the given inputs for Cache Hit and Miss Rate -



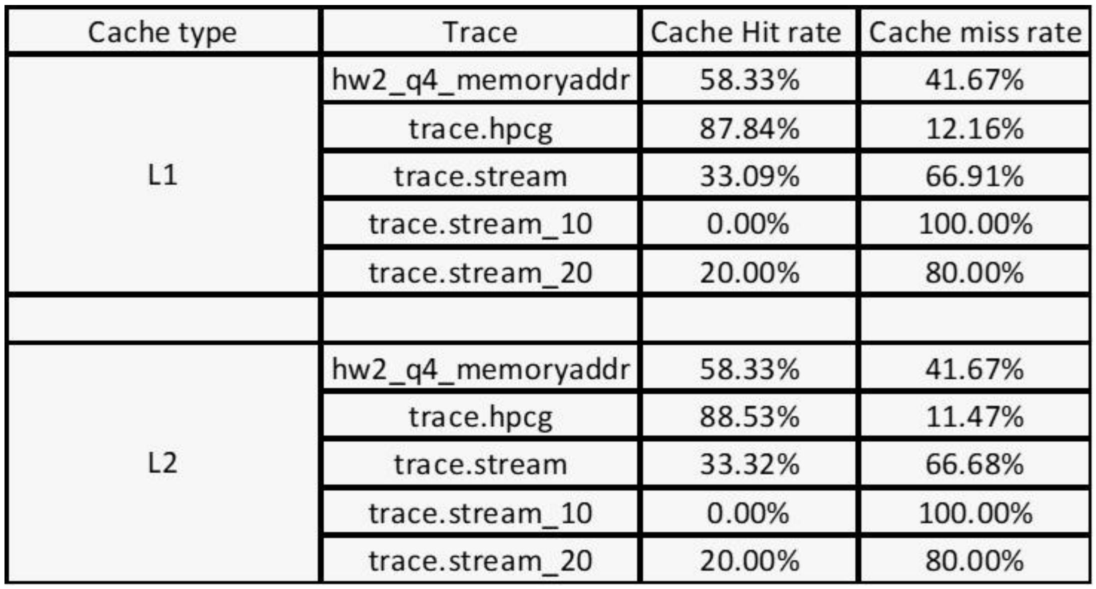
Inputs		hw2_q4_memoryaddr		trace.hpcg		trace.stream		trace.stream_10		trace.stream_20	
		Hit Rate	Miss Rate	Hit Rate	Miss Rate	Hit Rate	Miss Rate	Hit Rate	Miss Rate	Hit Rate	Miss Rate
Cache Size	32										
Cache Line Size	16	25%	75%								
Associativity	2-way			84.78%	15.22%	32.4	67.60%	0.00%	100.00%	15.00%	85.00%
Cache Size	32										
Cache Line Size	32										
Associativity	2-way	41.67%	58.33%	85.44%	14.56%	32.86%	67.14%	0.00%	100.00%	20.00%	80.00%
Cache Size	32										
Cache Line Size	128										
Associativity	2-way	58.33%	41.67%	89.48%	10.52%	32.76%	67.24%	0.00%	100.00%	20.00%	80.00%
Cache Size	32										
Cache Line Size	16										
Associativity	4-way	25.00%	75.00%	85.13%	14.87%	32.40%	67.60%	0.00%	100.00%	15.00%	85.00%
Cache Size	32										
Cache Line Size	32										
Associativity	4-way	41.67%	58.33%	85.82%	14.18%	32.86%	67.14%	0.00%	100.00%	20.00%	80.00%
Cache Size	32										
Cache Line Size	128										
Associativity	4-way	58.33%	41.67%	89.87%	10.13%	32.85%	67.15%	0.00%	100.00%	20.00%	80.00%
Cache Size	32										
Cache Line Size	16										
Associativity	8-way	25.00%	75.00%	85.23%	14.77%	32.40%	67.60%	0.00%	100.00%	15.00%	85.00%
Cache Size	32										
Cache Line Size	32										
Associativity	8-way	41.67%	58.33%	85.91%	14.09%	32.85%	67.15%	0.00%	100.00%	20.00%	80.00%
Cache Size	32										
Cache Line Size	128										
Associativity	8-way	41.67%	58.33%	85.93%	14.07%	32.86%	67.14%	0.00%	100.00%	20.00%	80.00%

Part 3. Two-level cache simulation

Each cache server in a cluster has a two-level cache architecture. A static cache and a dynamic cache made up the cache. Together, static and dynamic caches can increase processing speed by caching data. Despite being on the same processor chip package, a level 2 cache (L2 cache) is a CPU cache memory that is external to and separate from the microprocessor chip core. Early L2 cache designs had them mounted on the motherboard, which made them incredibly sluggish.

Results of Simulation of 2 level cache -

I have executed 2 level cache with different trace file and I got different results for the given inputs for Cache Hit and Miss Rate -



Cache type	Trace	Cache Hit rate	Cache miss rate
L1	hw2_q4_memoryaddr	58.33%	41.67%
	trace.hpcg	87.84%	12.16%
	trace.stream	33.09%	66.91%
	trace.stream_10	0.00%	100.00%
	trace.stream_20	20.00%	80.00%
L2	hw2_q4_memoryaddr	58.33%	41.67%
	trace.hpcg	88.53%	11.47%
	trace.stream	33.32%	66.68%
	trace.stream_10	0.00%	100.00%
	trace.stream_20	20.00%	80.00%

THE END.