EG 211: Computer Architecture November 22, 2020

Assignment 3

Instructor: Prof. Nanditha Rao Ghazi Shazan Ahmad (IMT2019033)

Introduction

The implementation for both the Direct Mapped and Set Associative Cache is done in C++.

Direct Mapped $\implies$ (1) Byte Offset = 2, (2) Tag Bits = 14, (3) Index Bits = 16

4 Way Set Associative $\implies$ (1) Byte Offset = 2, (2) Tag Bits = 16, (3) Index Bits = 14
The Replacement policy used is Least Recently Used(LRU) Algorithm.

Results

Hit Rate = Hits/Total

Miss Rate = Misses/Total

Direct Mapped Cache

	Total	Hits	Misses	Hit Rate	Miss Rate
gcc.trace	515683	483504	32179	93.75	6.25
gzip.trace	481044	320883	160161	66.70	33.3
mcf.trace	727230	7505	719725	1.03	98.97
swim.trace	303193	280738	22455	92.59	7.41
twolf.trace	482824	476770	6054	98.74	1.26

4 Way Set Associate Cache

	Total	Hits	Misses	Hit Rate	Miss Rate
gcc.trace	515683	483871	31812	93.83	6.17
gzip.trace	481044	320883	160161	66.70	33.3
mcf.trace	727230	7508	719722	1.032	98.968
swim.trace	303193	280825	22368	92.62	7.38
twolf.trace	482824	476844	5980	98.76	1.24

Observations

From the above experiment, it is observed that generally the Hit Rate of a 4-Way Associative Cache is greater than the Hit Rate of a Direct Memory Cache.
But, the point to be noted here is that the Hit Rates are largely influenced by the order of instructions given. As seen from the above tables the hit rates of both the Set Associative and the Direct Memory Cache in the gzip.trace file is same but for all the other 4 files there is a slight increase in the Hit Rates in the Set Associative Cache.

Instructions to Run

1. Compile the file using *g++ DM/SA.cpp*, the one which you want to compile.
2. Use *./a.out* to print the ouput in the command line terminal.