**The George Washington University**
School of Engineering & Applied Science
Electrical & Computer Engineering Department

**Instructor:** Prof. Louri **Semester:** Fall 2022
**Course:** Computer Architecture & Design ECE 6005 / ECE 4535

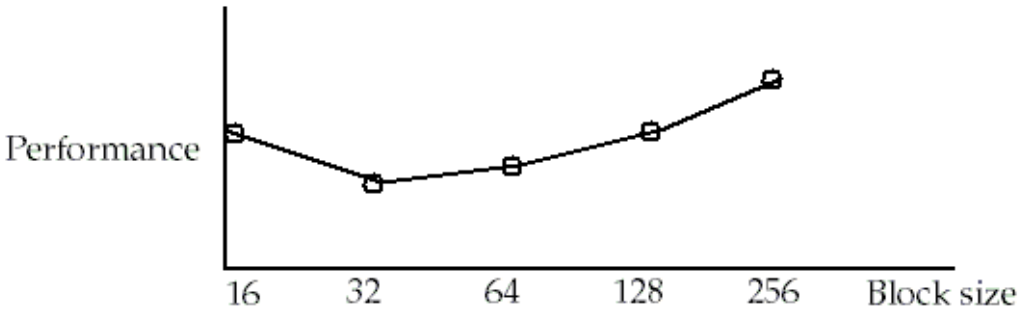
**Lab Assignment 2**

**Due Date: October 12**

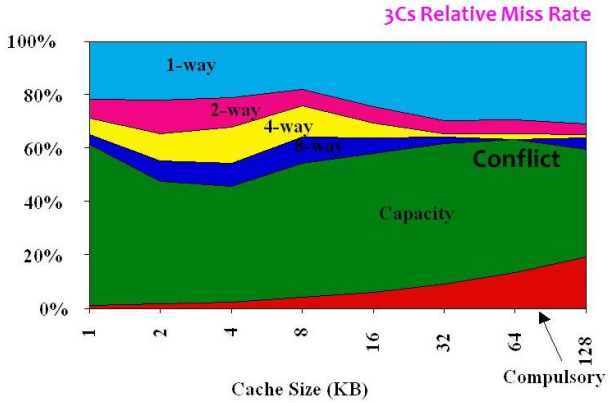
## Problem 1 (60)

Each question has its shell script named by its number.

a) According to the simulation result 1-a.txt, we could see that we can decrease the miss rate by using larger block size cache, but if the block size goes too big the miss rate will go up again, which happened because small block size has high miss rate, while bigger block size has lower miss rate yet bigger miss penalty at the same time.




b) According to the simulation result 1-b.txt, the miss rate will decrease with the set number of associativity.




c) According to the simulation result 1-c.txt, the miss rate gets the smallest when the replacement policy is LRU.

d) According to the simulation result 1-d.txt, the miss rate of instruction cache and data cache are dependent on applications. In average, the unified cache can get good performance.

e) According to the simulation result 1-e.txt, the miss rate will not be changed by write policy, and take cc1.din for example, the AMAT is $0.0405 \times 50 + (1 - 0.0405) = 3.1645$.

f) I couldn't find detailed cache information of any CPU product. So I can only do simulations. According to the simulation result 1-f.txt, the results of optimal cache setup and miss rate is as follows:

```
Trace:cc1.din
Optimal Cache type:i
Optimal Cache size:8'KB'
Optimal Block size:128
Optimal Associativity:64
Optimal Replacement policy:l
Optimal Miss rate:0.0199
Trace:spice.din
Optimal Cache type:d
Optimal Cache size:8'KB'
Optimal Block size:128
Optimal Associativity:64
Optimal Replacement policy:l
Optimal Miss rate:0.0034
Trace:tex.din
Optimal Cache type:i
Optimal Cache size:1'KB'
Optimal Block size:32
Optimal Associativity:8
Optimal Replacement policy:l
Optimal Miss rate:0.0000
```

## Problem 2 (40)

a) According to the data sheet, the STM32H7 family have 16KB of L1 data cache and 16KB of instruction cache.

**b) A system with a 256K-byte unified cache:**

Test command:

```
[yihui@redhat-vlab01 d4-7]$ ./dineroIV -l1-usize 256k -l1-ubsize 32 -l1-uassoc 2 -l1-urepl f -informat d
<cc1.din
```

Test result:

```
Demand miss rate            0.0080
```

Cache schematic:


**A system with a 64K-byte L1 cache and a 192K-byte L2 cache**:

Test command (since the cache size has to be power of 2, so I'm using 256kb here):

```
[yihui@redhat-vlab01 d4-7]$ ./dineroIV -l1-usize 64k -l1-ubsize 32 -l1-uassoc 2 -l1-urepl f -l2-usize 25
6k -l2-ubsize 32 -l2-uassoc 2 -l2-urepl f -informat d <cc1.din
```

Test result:

```
Demand miss rate                0.5764
```

Cache schematic:


**A system with a 32K-byte L1 instruction cache and a 32K-byte L1 instruction cache and 192K-byte L2 data cache** (since the cache size has to be power of 2, so I'm using 256kb here):

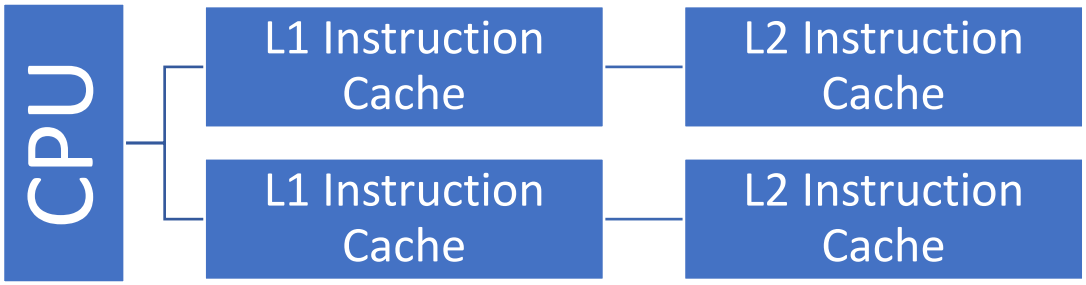
Test command:

```
[yihui@redhat-vlab01 d4-7]$ ./dineroIV -l1-isize 32k -l1-ibsize 32 -l1-iassoc 2 -l1-irepl f -l1-isize 32
k -l1-ibsize 32 -l1-iassoc 2 -l1-irepl f -l2-dsize 256k -l2-dbsize 32 -l2-dassoc 2 -l2-drepl f -informat
 d <cc1.din
```

Test Result:

```
Demand miss rate                0.0080
```

Cache schematic:



From the test result we could see that the miss rate of L1 cache won't be changed by following L2 or L3 cache, that is because the aspect that multilevel cache can improve is the miss penalty, but we can only see the miss rate from the result.