Cache Memory Design

In this lab, we would like to explore the cache performance as we change the cache design parameters. We will do so while executing the provided matrix multiplication code *matrix-vector-multiplication.asm.*

Note that in real benchmarking, a number of different applications are typically used. Additionally, it is likely to be automated in comparison to manually changing the parameters as you will be doing during this lab.

We will consider several memory organization designs as illustrated in the following tables

Parameter	values	Scenario 2	Scenario 3
Cache block size (words)	2-128	2-128	2-128
Cache size (bytes)	128- 1024	128- 1024	512
Placement policy	Direct Mapped	Full Associative	n-Way associative
Set size (blocks)	-	-	2,4,8
Block replacement policy	-	LRU, Random	LRU

Fill in the table(s) provided for every scenario with the achieved ***hit ratio*** using the provided information from the Data Cache Simulator tool (Tools → Data Cache Simulator).

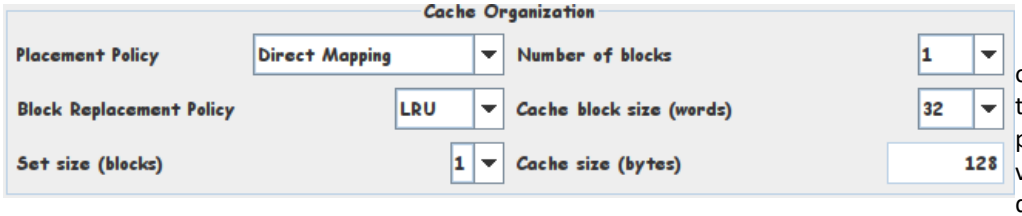
Note that as you change the cache block size, you would need to adjust the number of blocks to maintain a fixed cache size.
Alternatively, you may fill the table vertically by changing the number of blocks while keeping the block size fixed.

After completing data collection phase, you can proceed to the last assignment to answer the questions.

Scenario 1: Direct-mapped cache

In this part, we will investigate direct mapped cache while considering different cache sizes and block sizes.
You need to

1. load the "Data Cache Simulator" tool from MARS Tools menu and connect it to MIPS.
2. Open the provided lab code in MARS
3. Complete the following table by adjusting
    1. number of blocks, and
    2. cache block size
    3. run the provided program



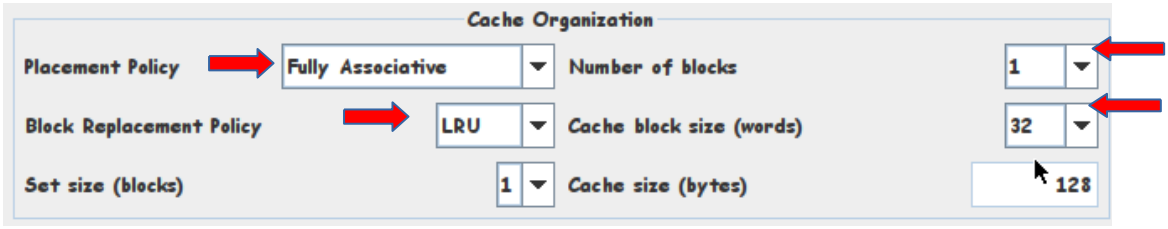
4. copy the provided hit ratio to the right cell in the table that matches the cache size and block size.

4. Repeat until you fill in the whole table.

	Block size (words)	2	4	8	16	32	64	128
Cache Size (bytes)	128	35%	42%	45%	47%	5%	-	-
	256	50%	60%	65%	68%	48%	5%	-
	512	66%	76%	80%	81%	71%	49%	5%
	1024	71%	82%	88%	91%	86%	75%	52%

Scenario 2: Fully-Associative Cache

In this part, we would like to explore fully associative cache with different replacement policies including LRU and random.



LRU replacement

Block size (words)		2	4	8	16	32	64	128
Cache Size (bytes)	128	50%	72%	83%	89%	5%	-	-
	256	75%	87%	94%	97%	89%	5%	-
	512	75%	87%	94%	97%	98%	89%	5%
	1024	75%	87%	94%	97%	98%	99%	94%

Random replacement

Change the replacement policy to random and repeat.

Block size (words)		2	4	8	16	32	64	128
Cache Size (bytes)	128	60%	77%	85%	85%	5%	-	-
	256	69%	83%	90%	92%	86%	5%	-
	512	72%	85%	93%	96%	97%	87%	5%
	1024	75%	87%	93%	97%	98%	97%	91%

Scenario 3: n-way set cache

In this part, we will consider **512**-byte cache memory.

Block size (words)		2	4	8	16	32	64	128
Set Size (blocks)	1-way set	66%	76%	80%	81%	71%	49%	5%
	2-way set	72%	84%	88%	89%	90%	89%	-
	4-way set	75%	87%	94%	97%	98%	-	-
	8-way set	75%	87%	94%	97%	-	-	-