Investigating Cache Technology

Objectives

At the end of this lab you should be able to:

- Investigate Directly Mapped cache organization
- Explain what cache hit and miss rates are
- Understand one drawback of Directly Mapped cache type
- Investigate 2-way Set-Associative cache mapping
- Investigate 4-way Set-Associative cache mapping
- Explain the effect of cache size and mapping scheme on cache performance

Lab Exercises - Investigate and Explore

The following exercises require the use of the cache simulator which is part of the CPU-OS Simulator software. The following exercises will use the data cache simulator only.

Exercise 1 – Investigating Directly Mapped cache organization

Create a new program, call it **CacheTest1** and enter the following code:

```
MOV #0, R01
STB R01, @R01
CMP #63, R01
JEQ 31
ADD #1, R01
JMP 6
HLT
```

The above code writes numbers 0 to 63 in memory locations 0 to 63. Run it and observe the contents of the data in memory. To see the memory click on the **SHOW PROGRAM DATA MEMORY…** button.

Click on the **SHOW CACHE…** button to display the data cache window. Make sure the **Stay on top** check box is checked. Now, flush the cache by clicking on the **FLUSH** button and configure the cache with the following settings:

Block Size = 4
Cache Type = Direct Mapped
Cache Size = 16

Now insert the following code below the instruction **JMP 6** in the above code:

```
LDB 0, R00
LDB 1, R00
LDB 2, R00
LDB 3, R00
```

To execute the above LDB instructions individually, double-click on each of the above LDB instructions. Write down what you observe in the table below:

Addr	Data	Hits	Block

Also make a note of the following data displayed in the **Cache Stats** frame:

Hits		**%Hits**	
Misses		**%Misses**	

Note: %Hits = 100 - %Misses

Insert the following LDB instruction after the last LDB instruction above and execute it by double-clicking on it:

LDB 4, R00

Write down the additional contents (i.e. in addition to the above data) of the cache below:

Addr	Data	Block

Briefly explain your observations below:

Exercise 2 – Investigating a disadvantage of Directly Mapped cache

First flush the contents of the cache by clicking on the **FLUSH** button. Then enter the following instructions after the last LDB instruction in the above program:

LDB 16, R00
LDB 32, R00

Next execute only the following three instructions in the above program:

LDB 0, R00
LDB 16, R00
LDB 32, R00

Repeat the above two times and make note of what you observe below:

Addr	Data	Hits	Block

.

Hits		%Hits	
Misses		%Misses	

Briefly comment on your findings below:

Exercise 3 - Investigating Set-Associatively Mapped cache organization

Now configure the cache with the following settings:

Block Size = 4
Cache Type = Set Associative
Cache Size = 16
Set Blocks = 2-way

Insert the following new LDB instructions after the **LDB 4, R00** instruction:

LDB 8, R00
LDB 12, R00

Execute the following set of LDB instructions one after the other in the order listed below:

LDB 0, R00
LDB 4, R00
LDB 8, R00
LDB 12, R00

Write down your observations below (the **Addr** field is filled in for you):

Addr	**Set**	**Block**
0000		
0004		
0008		
0012		

Next re-configure the cache so that the **Set Blocks** is set to **4-way**.

Execute the following set of LDB instructions one after the other in the order listed below:

LDB 0, R00
LDB 4, R00
LDB 8, R00
LDB 12, R00

Write down your observations below (the **Addr** field is filled in for you):

Addr	**Set**	**Block**
0000		
0004		
0008		
0012		

Clear the cache by clicking on the **FLUSH** button. Next execute only the following two LDB instructions in the above program:

LDB 0, R00
LDB 16, R00

Repeat the above three times and make note of what you observe below:

Addr	**Data**	**Hits**

.

Hits		**%Hits**	
Misses		**%Misses**	

Briefly comment on your findings below:

Exercise 4 - Investigating cache size and type on cache performance

Enter the following program, compile it and load it in CPU memory:

```
program CacheTest2
    var a array(120) byte

    i = 199
    for n = 0 to 119
        p = a(n) + a(i)
        i = i - 1
    next
end
```

Now, make sure that the charts are enabled. Also make sure the **Write Policy** option **Write-Back** is selected. Select the correct cache parameters and fill in the **%Misses** in the tables below against each of the cache sizes shown in the tables after running the program to completion which may take up to a minute. Make sure you click on the **RESET PROGRAM** button and slide the speed selector up to the fastest position prior to running the program in each case. Enter % changes in the %misses in the third column in the first two cases only.

Direct mapping (for cache sizes 16 to 256)

16	**32**	**64**	**128**	**256**

2-way set-associative mapping (for cache sizes 16 to 256)

16	32	64	128	256

4-way set-associative mapping (for cache sizes 16 to 256)

16	32	64	128	256

8-way set-associative mapping (for cache sizes 32 to 128)

32	64	128	256

Briefly comment on the results (**Tip**: Compare the bar charts against each other):