



UNIVERSITY
of York

DEPARTMENT OF ELECTRONICS
COMPUTER ARCHITECTURES

Homework Two

Abstract

...

Y3839090

April 27, 2017

Contents

1	Question 1	1
2	Question 2	2
2.1	direct-mapped cache	2
2.1.1	sizes	2
2.1.2	hits and misses	2
2.2	fully-associative cache	2
2.2.1	sizes	2
2.2.2	hits and misses	3
2.3	set-associative cache with 2 blocks per set	3
2.3.1	sizes	3
2.3.2	hits and misses	3
2.4	set-associative cache with 8 blocks per set	3
2.4.1	sizes	3
2.4.2	hits and misses	3
3	Question 3	3
	Appendices	4

List of Figures

1	Flowchart for paged segmentation	5
---	--	---

1 Question 1

A Flowchart of the algorithm for paged segmentation, assuming the presence of separate translation look aside buffers (TLBs) for pages and segments.

2 Question 2

$$\text{addresssize} = 32b$$

$$\text{blocksize} = 64\text{words} = 2048b$$

$$\text{wordsize} = 32b$$

$$\text{cachesize} = 16kB = 131072b$$

$$\text{numblocks} = \text{cachesize}/\text{blocksize} = 64$$

2.1 direct-mapped cache

2.1.1 sizes

$$\text{Offset : } \text{size}(\text{offset}) = (\text{blocksize}/\text{wordsize}) + \text{wordsize} = (256/32) + 4 = 12b$$

$$\text{Index : } \text{size}(\text{index}) = \log_2(\text{cachesize}/\text{blocksize}) - 1 = \log_2(16kB/256B) - 1 = 8b$$

$$\text{Tag : } \text{size}(\text{tag}) = \text{addresssize} - \text{size}(\text{offset}) - \text{size}(\text{index}) = 32 - 12 - 8 = 12b$$

2.1.2 hits and misses

Array A and B map to the same location in the cache and so there is always a cache miss when trying to access one of them. This leads to A and B both having 256 cache misses and 0 cache hits. Array C is mapped to an area in cache that is not occupied by Array A or B and so is cached effectively. It has 4 cache misses and 252 cache hits.

In total this is 516 cache misses and 252 cache hits

2.2 fully-associative cache

2.2.1 sizes

Offset : The same as direct mapped cache = 12b

Index : Fully associative cache does not have an index 0b

$$\text{Tag : } \text{size}(\text{tag}) = \text{addresssize} - \text{size}(\text{offset}) = 32 - 12 = 20b$$

2.2.2 hits and misses

With a fully-associative cache blocks can be placed anywhere. The cache also has more than enough blocks to store the there arrays. Because of these two facts the entirety of all there arrays can be placed in cache, This means that there will only be cache misses when a new block of an array needs to be loaded. Each array will have 4 cache misses and 252 cache hits

In total this is 12 cache misses and 756 cache hits

2.3 set-associative cache with 2 blocks per set

2.3.1 sizes

Offset : The same as direct mapped cache = $12b$

Index : $size(\text{index}) = \log_2(\text{numsets}) - 1 = \log_2(\text{blocksize}/2) - 1 = 4$

Tag : $size(\text{tag}) = \text{addresssize} - size(\text{offset}) - size(\text{index}) = 32 - 12 - 4 = 16b$

2.3.2 hits and misses

2.4 set-associative cache with 8 blocks per set

2.4.1 sizes

Offset : The same as direct mapped cache = $12b$

Index : $size(\text{index}) = \log_2(\text{numsets}) - 1 = \log_2(\text{blocksize}/8) - 1 = 2$

Tag : $size(\text{tag}) = \text{addresssize} - size(\text{offset}) - size(\text{index}) = 32 - 12 - 2 = 18b$

2.4.2 hits and misses

3 Question 3

Appendices

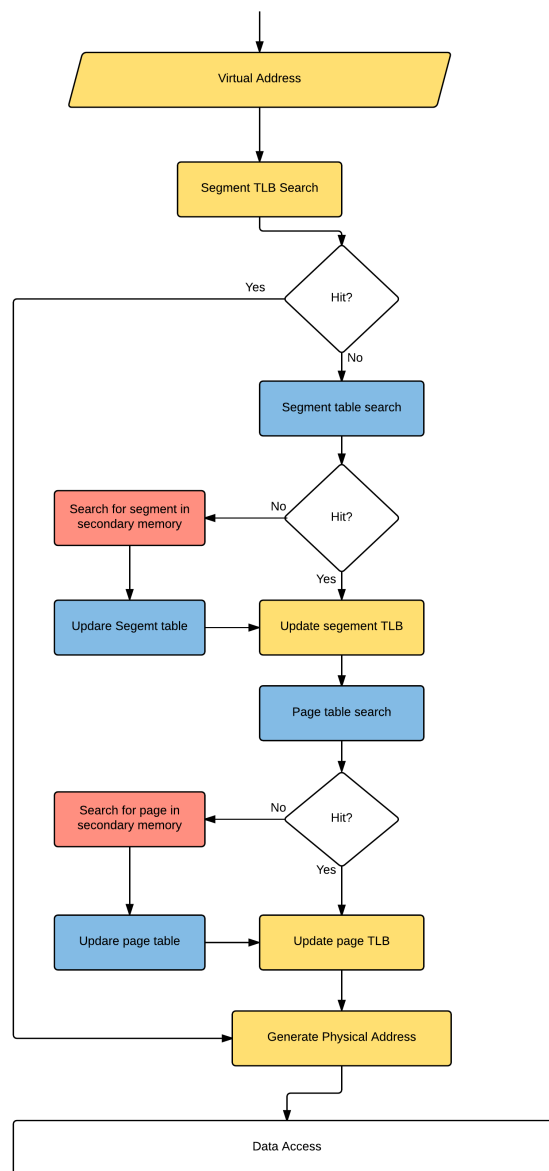


Figure 1: Flowchart for paged segmentation