

8. In this exercise we will evaluate the memory access time of a small program under different cache configurations. The program will access the following 20 addresses in order (addresses are given as 8-bit hex numbers for simplicity):

0x00 0x04 0x08 0x0C 0x00 0x04 0x10 0x14 0x40 0x44→  
0x00 0x04 0x48 0x4C 0x08 0x0C 0x00 0x04 0x48 0x4C

In this system one main memory access takes 20ns.

- a) If the system has no cache, how much time will it make all memory accesses in the program given above? (1 point)

$$t_{\text{total}} = N \times t_{\text{mem.}}$$

$$t_{\text{total}} = 20 \times 20\text{ns.} = 400 \text{ ns}$$

- b) As an alternative, it was decided to use a direct mapped cache with capacity of 8 words and a block size of 1. The cache access time for this cache is 2ns. Using the table below, show the final content of this cache memory **after** executing the program above. (2 points)

Location	Content
Set 7	
Set 6	
Set 5	14
Set 4	10
Set 3	<del>0C</del> <del>4C</del> <del>0C</del> 4C
Set 2	<del>08</del> <del>48</del> <del>08</del> 48
Set 1	<del>04</del> <del>44</del> 04
Set 0	<del>00</del> <del>40</del> 00

- c) How many compulsory cache misses were there? (1 point)

There are six compulsory misses: the first four accesses to 00 04 08 0C and then the accesses to 10 14 on the 7<sup>th</sup> and 8<sup>th</sup> cycles.

- d) How many conflict misses were there? (1 point)

There are 10 conflict misses: 8<sup>th</sup> cycle 40 conflicts with 00, 9<sup>th</sup> cycle 44 conflicts with 04, 10<sup>th</sup> cycle 00 conflicts with 40, 11<sup>th</sup> cycle 04 conflicts with 44, 12<sup>th</sup> cycle 48 conflicts with 08, 13<sup>th</sup> cycle 4C conflicts with 0C, 14<sup>th</sup> cycle 08 conflicts with 48, 15<sup>th</sup> cycle 0C conflicts with 4C, 18<sup>th</sup> cycle 48 conflicts with 08, 19<sup>th</sup> cycle 4C conflicts with 0C

e) What is the Miss Ratio for this cache? (1 points)

There are 16 misses out of 20 accesses. So the Miss Rate is  $16/20 = 80\%$

f) How long will it take to make all the memory accesses for the program given above? (2 points)

There are 20 cache accesses each  $2\text{ns} = 2 \times 20\text{ns} = 40\text{ns}$

There are 16 cache misses, each resulting in a memory access  $= 16 \times 20\text{ns} = 320\text{ns}$

Total is  $40\text{ns} + 320\text{ns} = 360\text{ns}$

OR  $\text{AMAT} = t_{\text{cache}} + (\text{MR} \times t_{\text{mem}}) = 2\text{ns} + (0.8 \times 20\text{ns}) = 18\text{ns}$ .

Total time memory access  $\times \text{AMAT} = 20 \times 18\text{ns} = 360\text{ns}$

g) There are four suggestions below. In each case only one parameter of the cache will be changed. Which of the following changes would improve the total memory access time **of this system running the above program**, indicate all that apply? (2 points)

a. Increasing the Capacity from 8 to 16

b. Increasing Block size from 1 to 2

c. Increasing Set Associativity from 1 (direct mapped) to 2

d. Increasing Cache Access Time  $1\text{ns}$  to  $2\text{ns}$

Note:  $1\text{ns} = 0.000\,000\,001\text{s} = 1 \cdot 10^{-9}\text{s}$   
 $1\text{MHz} = 1\,000\,000\text{Hz} = 1 \cdot 10^6\text{Hz}$