Note: Do not refer to the processor configuration in the case study notes for this tutorial. A smaller system will be used instead.

8.1 Cache

1. Given a processor system with the following characteristics
   - Processor has a direct-mapped cache with 32 cache blocks and a cache size of 512 bytes.
   - Cache Memory Access time = 5ns.
   - Cache Hit rate = 0.9
   - 64Kbyte DRAM used as the main memory.
   - DRAM Memory access time = 200ns

cache block size =16,offset = 4bits

32 cache blocks, 5 BLK bits

MM address bit = 16 bits for 64KB

TAG bits = 16-5-4 = 7

a. In doing cache mapping analysis, how many **blocks** would the main memory be partitioned to?

32 ✗ Number of main memory blocks = 64KB/16B = 4096

b. What is the format of a memory address as seen by the cache (i.e. determine the sizes of the tag, block and offset fields)?

7 : 5 : 4 ✓

0x0DB63 = 1101 1011 0110 0011
=block 10110b = Block 22

c. CPU needs to read a byte from main memory address 0xDB63.

0xBA03

1101 1011 0110 0011
t b o

i. Which cache block would CPU looked at to search for the required data?

ii. How many main memory blocks could potentially be mapped to the same cache block as that of 0xDB63?

1 ✗

iii. How does the CPU knows if the cache block identified in (i) above contains the data that it needs?

It compares TAG value of cache block 0 and the TAG value from 0xDB63 MM address. If it is a match, a cache hit occurred and the data is correctly identified. If it is different, the data in cache block 0 did not come from MM block where the data resides.

iv. What is the purpose of the 'offset' field in the cache mapping?

offset determines the which byte from the start of the cache block is being accessed

For 0xDB63, the byte is the byte 3 of block 22. First byte of the block is byte 0.

d. What is the effective access time of the memory in this system?

For 0xDB63, the TAG value for the cache block is equal to 1101 101b

MM

effective access time = 0.9 x 5 x 10^-9 + 0.1 x (5 x 10^-9+200 x 10^-9) = 25 x 10^-9 s ✓

ii) Number of MM blocks =4096. Number of cache blocks = 32 => each cache block is a potential destination to 4096/32 = 128MM

8.2 Virtual Memory

2. In a processor system with the following characteristics,
   - 1 MByte Virtual memory space ⇒ 10 bits for page as offset is 1KB
   - 64 Kbyte DRAM as main memory ⇒ 6 bits for frame
   - Paging scheme used for virtual memory management, Page Table as shown in Table 8.2
   - Virtual Page size = 1 KByte => offset field is 10 bits
   - TLB with 4 entries

1MB/1KB = 1024

64KB/1KB = 64

Table 8.2 – Page Table

Virtual Page Number	Valid Bit	Page Frame Number
0	1	1
1	1	2
2	0	-
3	1	16
4	1	9

a. How many bits are required for each virtual address? 20

b. How many bits are required for each physical address? 16

c. What is the maximum number of entries in the page table in Table 8.2? 5K

Number of pages = virtual mem space/virtual page size = 2^20B/1024 = 1048576B /1024 = 1024

d. What is the maximum number of valid entries in the page table in Table 8.2? 4 ✗ $2^6 = 64$

e. With reference to Table 8.2, answer the following. Indicate when a page fault occurs.

10 bits ~ 6 bits

(i) The compiler mapped the UART routine to virtual address 0x005F0 – 0x006FF, where in the DRAM would you be able to find the UART routine?

0000 01 ..... ⇒ 1

Virtual Page 1 mapped to Frame 2 and valid bit is 1

0000 10 .....

10011

In 0x05F0 - 0x06FF

In 0x09C0 - 0x09DF

(ii) The compiler mapped the I2C routine to virtual address 0x009C0 - 0x009DF, where in the DRAM would you be able to find the I2C routine? 0000 10 ⇒ 2

Page fault

(iii) What happens when there is a page fault?

The OS is triggered to load the required page from storage memory to Physical Memory

f. What memory are the Page Table and TLB resided?

processor

Main memory

g. What is the function and effect of a TLB?

to speed up page translation process. TLB stores a list of most recently/ frequently used address translation entries in the page table as a subset of Page Table.

(Not necessary to be covered during tutorial)

[Optional, but students are encouraged to attempt these questions]

3. Consider a system with the following characteristics.

- Direct mapped cache of 32 cache blocks and cache block size of 32 bytes
- Cache uses Physical Address for address mapping
- Virtual Memory page size 2048 bytes
- Virtual Memory size is 1Mbyte. Physical Memory size is 64KByte
- Extracts of Page Table (valid entries)
  - Virtual Page 0 → Physical Frame 9
  - Virtual Page 1 → Physical Frame 3
  - Virtual Page 2 → Physical Frame 5
  - Virtual Page 3 → Physical Frame 2
  - Virtual Page 4 → Physical Frame 7
- The main program is 5KByte in size and starts at virtual address 0x01006

(i) Assuming that the compiler allocates the program sequentially in the virtual memory, what is the physical address of the start and end of the main program?

(ii) Which cache block should the CPU check in the cache for the start of the main program? What is the corresponding TAG value used to check for cache hit/miss?

Physical Address used for tagging.
TAG: BLK:OFFSET = 6:5:5
Start address = 0x2806
=0010 1000 0000 0110
TAG value = 001010 = 0xA
BLK = 00000 = 0

Virtual Address(Start) = 0x01006,end = 0x02405 (5KB size = 5120B =
0001 0100 0000 0000)

0x1006 => 0001 0000 0000 0110 =>page 2 =>physical frame 5(101b)
Physical address = 0010 1000 0000 0110 =0x2806

0x2405 =>0010 0100 0000 0101 =>page 4=>physical frame 7(111b)
Physical Address = 0011 1100 0000 0101