

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
- a. In doing cache mapping analysis, how many **blocks** would the main memory be partitioned to?
- 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)?
- c. CPU needs to read a byte from main memory address 0x0DB63.
 - i. Which cache block would CPU look 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 0x0DB63?
 - iii. How does the CPU know if the cache block identified in (i) above contains the data that it needs?
 - iv. What is the purpose of the 'offset' field in the cache mapping?
- d. What is the effective access time of the memory in this system?

8.2 Virtual Memory

2. In a processor system with the following characteristics,
- 1 MByte Virtual memory space
 - 64 Kbyte DRAM as main memory
 - Paging scheme used for virtual memory management, Page Table as shown in Table 8.2
 - Virtual Page size = 1 KByte
 - TLB with 4 entries

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

- How many bits are required for each virtual address?
- How many bits are required for each physical address?
- What is the maximum number of entries in the page table in Table 7.2?
- What is the maximum number of valid entries in the page table in Table 7.2?
- With reference to Table 7.2, answer the following. Indicate when a page fault occurs.
 - The compiler mapped the UART routine to virtual address 0x005F0 – 0x006FF, where in the DRAM would you be able to find the UART routine?
 - The compiler mapped the I2C routine to virtual address 0x009C0 - 0x009DF, where in the DRAM would you be able to find the I2C routine?
 - What happens when there is a page fault?
- What memory are the Page Table and TLB resided?
- What is the function and effect of a TLB?

(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?