Assignment 2
CDA 5155: Computer Architecture

David Miller
February 1, 2018

The following is a description of the SNERDLEY 1000. Given this information, answer the following questions.

Virtual Memory

1 GB virtual memory
256 MB physical memory
16 KB page size

Instruction TLB

Direct Mapped
64 virtual to physical translations

Data TLB

2-way associative
128 virtual to physical translations

Instruction Cache

Cache line size is 32 bytes
2-way set associative
4 KB of instructions
LRU replacement

Data Cache

Cache line size is 64 bytes
Direct mapped
8 KB of data
write-through/no-write allocate

Secondary Cache

Unified cache
Cache line size is 512 bytes
4-way associative
1 MB in cache
LRU replacement
write-back/write allocate

Problem 1 *How many bits are in a virtual address?*

Virtual memory has 1GB $= 2^{30}$ bytes. Therefore we need 30 bits to for the virtual address.

Problem 2 *How many bits are in a physical address?*

Physical memory has 256 MB $= 2^{28}$ bytes. Therefore we need 28 bits for the physical address.

Problem 3 *How many virtual pages are there?*

Virtual memory can hold 2^{30} bytes and each virtual page takes 16 KB $= 2^{14}$ bytes, therefore we have $2^{30}/2^{14} = 2^{16}$ virtual pages.

Problem 4 *How many bits are in the page offset?*

Each page is 2^{14} bytes, therefore we need $\log_2(2^{14}) = 14$ bits.

Problem 5 *Show how the virtual address is partitioned (i.e. virtual page number and page offset).*

Page Number	Offset
29 14	13 0

Problem 6 *How many physical pages are there?*

Physical memory can hold 2^{28} bytes and each virtual page takes 2^{14} bytes, therefore we have $2^{28}/2^{14} = 2^{14}$ virtual pages.

Problem 7 *Show how the physical address is partitioned (i.e. physical page number and page offset).*

Page Number	Offset
27 14	13 0

Problem 8 *How many entries are in the page table?*

The number of entries is the page table is equal to the number of virtual pages, therefore there are 2^{16} entries in the page table.

Problem 9 *How many bits are required for the tag to access the instruction TLB?*

The bits used for the virtual page number are portioned into bits for the index and bits for the tag. The instruction TLB is direct mapped and has 64 virtual to physical translations (entries), therefore we need $\log_2(64) = \log_2(2^6) = 6$ bits for the index. Then we have 16 - 6 = 10 bits left for the tag. Intuitively this makes sense. If there are 2^{16} virtual pages that must be uniquely identifiable in 2^6 sets, then there has to be 2^{10} tags so that no two elements in the same set can be confused for one another. Mathematically, this is equaivalent to the bijective map

$$f : \underbrace{\{0,1\}^{20}}_{\text{VPN}} \rightarrow \underbrace{\{0,1\}^{14}}_{\text{Tag}} \times \underbrace{\{0,1\}^{6}}_{\text{Index}}.$$

Problem 10 *How many sets are in the instruction cache?*

The instruction cache is 4 KB = 2^2 KB = 2^{12} bytes with 32 byte = 2^5 byte cache line. Since the cache is 2-way associative we have $2 \cdot 2^5$ bytes = 2^6 bytes per set. Therefore there are $2^{12}/2^6 = 2^6$ sets.

Problem 11 *How many bits are required for the tag to access the instruction cache?*

Physical address has 28 bits. Since the cache line size is 32 bytes = 2^5 bytes we need $\log_2(2^5) = 5$ bits for the offset. The instruction cache has 2^6 sets and therefore needs $\log_2(2^6) = 6$ bits for the index. Therefore we have 28 - (5 + 6) = 17 bits for the tag.

Problem 12 *How many sets are in the data cache?*

The data cache is 8 KB = 2^3 KB = $2^1$3 bytes with 64 bytes = 2^6 byte cache line. Since the cache is direct mapped we only have one line per set. Therefore there are $2^1 3/2^6 = 2^7$ sets.

Problem 13 *How many bits are required for the tag to access the data cache?*

Physical address has 28 bits. Since the cache line size is 64 bytes = 2^6 bytes we need $\log_2(2^6) = 6$ bits for the offset. The data cache has 2^7 sets and therefore needs $\log_2(2^7) = 7$ bits for the index. Therefore we have 28 - (6 + 7) = 15 bits for the tag.

Problem 14 *How many sets are in the secondary cache?*

The secondary cache is 1 MB = 2^{10} KB = 2^{20} bytes with 512 bytes = 2^9 bytes cache line. Since the cache is 4-way associative we have $4 \cdot 2^9$ bytes = $2^2 \cdot 2^9$ bytes = 2^{11} bytes per set. Therefore there are $2^{20}/2^{11} = 2^9$ sets.

Problem 15 *How many bits are required for the tag to access the secondary cache?*

Physical address has 28 bits. Since the cache line size is 512 bytes = 2^9 bytes we need $\log_2(2^9) = 9$ bits for the offset. The secondary cache has 2^9 sets and therefore needs $\log_2(2^9) = 9$ bits for the index. Therefore we have 28 - (9 + 9) = 10 bits for the tag.