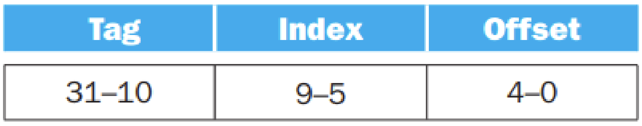
CS202 Computer Organization HW#4

Problem 1.

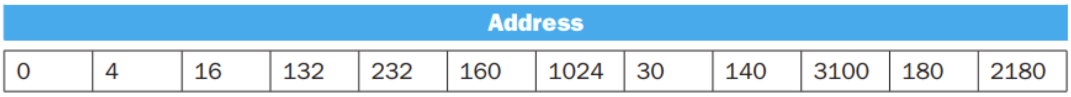
For a direct-mapped cache design with a 32-bit address, the following bits of the address are used to access the cache.



Tag	Index	Offset
31–10	9–5	4–0

1. What is the cache block size (in words)?
2. How many entries does the cache have?
3. What is the ratio between total bits required for such a cache implementation over the data storage bits? (Assume the cache has valid bit, dirty bit and reference bit)

Starting from power on, the following byte-addressed cache references are recorded.



Address											
0	4	16	132	232	160	1024	30	140	3100	180	2180

4. How many blocks are replaced?
5. What is the hit ratio?
6. List the final state of the cache, with each valid entry represented as a record of <index, tag, data>.

Problem 2.

Caches are important to providing a high-performance memory hierarchy to processors. Below is a list of 32-bit memory address references, given as word addresses

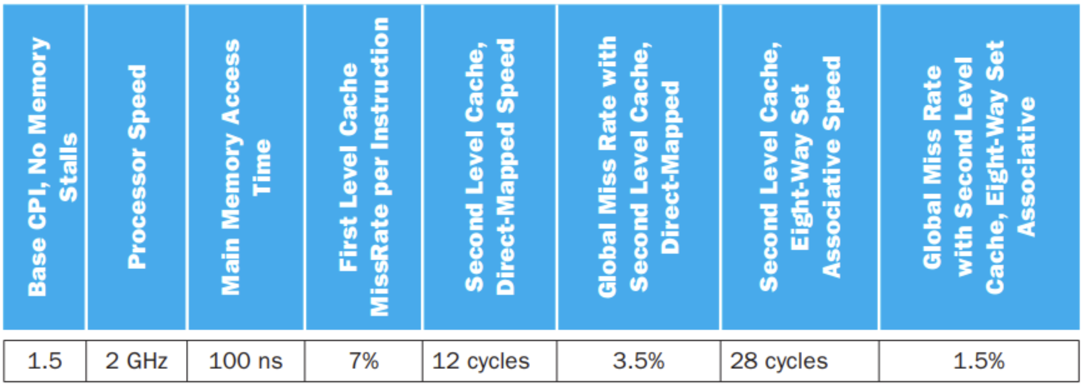
3, 180, 43, 2, 191, 88, 190, 14, 181

The is exercise examines the impact of different cache designs, specifically comparing associative caches to the direct-mapped caches. The address stream is shown above.

1. For a three-way set associative cache with two-word blocks and a total size of 24 words, which bits represent index and which bits represent tag in 32-bit memory address (e.g. Index: 9-5 Tag: 31-10) ?
2. Using the given sequence, show the final cache contents for a three-way set associative cache with two-word blocks and a total size of 24 words. Use LRU replacement. For each reference identify the index bits, the tag bits, the block of set bits, and if it is a hit or a miss.
3. For a fully associative cache with one-word blocks and a total size of 8 words, which bits represent index and which bits represent tag in 32-bit memory address (e.g. Index: 9-5 Tag: 31-10) ?

4. Using the given sequence, show the final cache contents for a fully associative cache with one-word blocks and a total size of 8 words. Use LRU replacement. For each reference identify the index bits, the tag bits, and if it is a hit or a miss.

Multilevel caching is an important technique to overcome the limited amount of space that a first level cache can provide while still maintaining its speed. Consider a processor with the following parameters:



Base CPI, No Memory Stalls	Processor Speed	Main Memory Access Time	First Level Cache MissRate per Instruction	Second Level Cache, Direct-Mapped Speed	Global Miss Rate with Second Level Cache, Direct-Mapped	Second Level Cache, Eight-Way Set Associative Speed	Global Miss Rate with Second Level Cache, Eight-Way Set Associative
1.5	2 GHz	100 ns	7%	12 cycles	3.5%	28 cycles	1.5%

5. Calculate the CPI for the processor in the table using: 1) only a first level cache, 2) a second level direct-mapped cache, and 3) a second level eight-way set associative cache.

Problem 3.

This Exercise examines the single error correcting, double error detecting (SEC/DED) Hamming code.

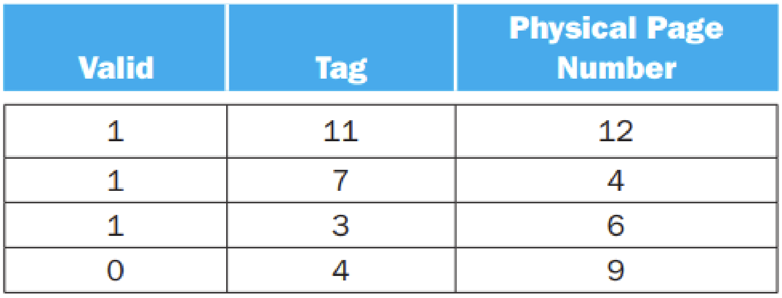
1. What is the minimum number of parity bits required to protect a 128-bit word using the SEC/DED code?
2. Consider a SEC code that protects 8 bit words with 4 parity bits. If we read the value 0x375, is there an error? If so, correct the error.

Problem 4.

Virtual memory uses a page table to track the mapping of virtual addresses to physical addresses. This exercise shows how this table must be updated as addresses are accessed. The following data constitutes a stream of virtual addresses as seen on a system. Assume page size is 4 KiB, a 4-entry fully associative TLB, and true LRU replacement. If pages must be brought in from disk, increment the next largest page number (i.e. If the current maximum physical page number is 12, then the next physical page brought in from disk has a page number of 13).

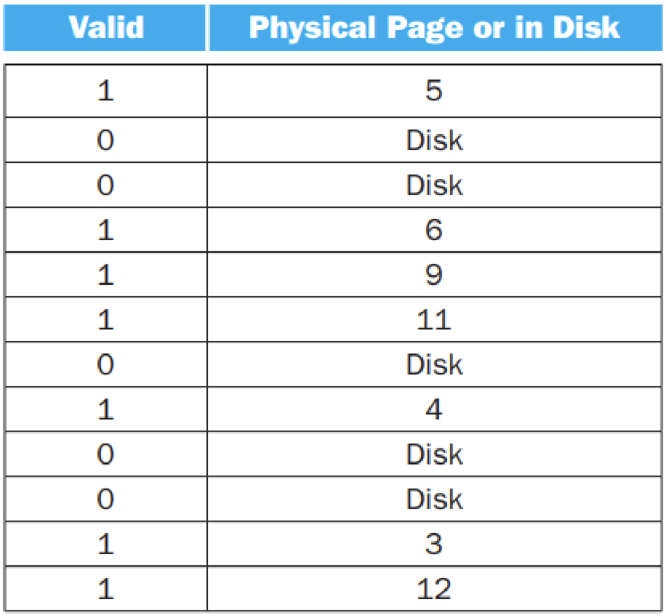
4669, 2227, 13916, 34587, 12608

TLB:



Valid	Tag	Physical Page Number
1	11	12
1	7	4
1	3	6
0	4	9

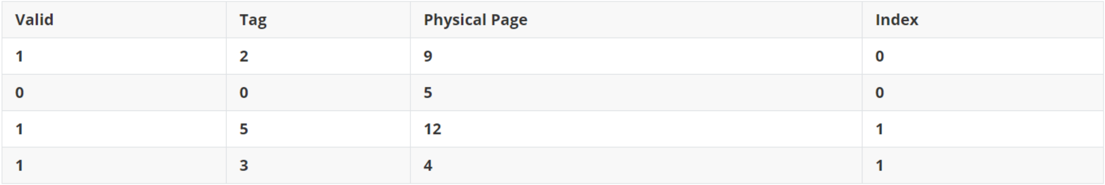
Page table:



Valid	Physical Page or in Disk
1	5
0	Disk
0	Disk
1	6
1	9
1	11
0	Disk
1	4
0	Disk
0	Disk
1	3
1	12

1. Which bits represent virtual page number and which bits represent page offset in 32-bit virtual memory address?
2. Given the address stream shown, and the initial TLB and page table states provided above, show the final state of the system. Also list for each reference if it is a hit in the TLB, a hit in the page table, or a page fault.
3. Show the final contents of the TLB if it is 2-way set associative. **The page table states are provided above and the initial TLB is shown below.** Discuss the importance of having a TLB to high performance. How would virtual memory accesses be handled if there were no TLB?

TLB:



Valid	Tag	Physical Page	Index
1	2	9	0
0	0	5	0
1	5	12	1
1	3	4	1