Question 1: (25 points) A computer system has 16 megabytes of addressable memory and a 128 kbyte, 2-way set-associative cache with 16 bytes per line. The LRU replacement policy is used. Assume the cache is empty and we start executing a program that makes the following sequence of references. Use the table provided to give the tag, set and line number for non-empty lines at the end of the sequence. **Use hexadecimal notation for all numbers.**

Answer:
Cache size = LKN. 128 kbytes = (16)(2)(N), so N = 4096 sets.
Total address bits = \(\log_2\) (16 megabytes) = 24
Address bits for L = \(\log_2\) (16) = 4
Address bits for N = \(\log_2\) (4096) = 12
Address bits for tag = 24 – 4 – 12 = 8

**Answer Sequence**
1. 3C2B20 Allocate line 0 of set 2B2 with tag of 3C
2. 3C2324 Allocate line 0 of set 232 with tag of 3C
3. 172B24 Allocate line 1 of set 2B2 with tag of 17 (line 0 already used)
4. 3C2B28 Hit on line 0 of set 2B2 with tag of 3C
5. 182B27 Allocate line 1 of set 2B2 with tag of 18 (LRU is tag 3C, so displace 17)
6. 258032 Allocate line 0 of set 803 with tag of 25
7. B22328 Allocate line 1 of set 232 with tag of B2
8. 172328 Allocate line 0 of set 232 with tag of 17 (LRU is tag B2, so displace 3C)

Result non-empty lines at the end of the sequence:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Set</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>3C</td>
<td>2B2</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>2B2</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>232</td>
<td>0</td>
</tr>
<tr>
<td>B2</td>
<td>232</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>803</td>
<td>0</td>
</tr>
</tbody>
</table>
Question 2. (15 points) You want to design a 512 kbyte 4-way set associative cache. The data bus width is 64 bits. If you use 256 kbit SRAMs, what chip organization do you need in order to provide fast access to your cache?

- Answer: We need $4 \times 64 = 256$ data outputs from the memory chips in order to read out a word from all 4 lines of a set in parallel. We also need $(512k)(8)$ bits / 256k bits per chip = 16 chips for the cache. These 16 chips need to provide 256 data output lines. $256$ lines / 16 chips = 16 data output lines per chip. So, each chip must be organized as $256k/16 \times 16 = 16k \times 16$. As a check, if we have 16 chips organized as 16k x 16, we get a total of 16 x 16 = 256 data lines and a total of 16 chips x 256 kbits per chip / 8 bits per byte = 512 kbytes of cache memory.

Question 3. You design a processor that provides virtual memory. The internal clock of your processor runs at 400 MHz. A small level 1 cache is implemented within the processor chip with an access time of 1 clock cycle. The internal cache achieves a hit rate of 95%. The virtual memory system uses a two-level lookup through a segment table and a page table. A TLB is implemented internal to the processor chip and can translate an address in 1 clock with a hit rate of 98%. When a full translation is necessary, the segment table hit rate is 90% and the page table hit rate is 25%. Your main memory is 7.5 ns Synchronous DRAM and you run the memory bus clock at 133 MHz. The SDRAM requires 7 clocks from RAS to first data out.

a) (10 points) What is the effective memory access time of your system in nanoseconds?
b) (5 points) What would be the effective memory access time if we turned off virtual memory translation and ran the processor in real mode?

- Answer: a) The internal cache is running at the internal clock rate of 400 MHz. 1 clock at 400 MHz = 2.5 ns access to the internal cache. Initial word access to the SDRAM is 7 clocks x 7.5 ns/clock = 52.5 ns.

\[
t_{\text{eff}} = t_{\text{TLB}} + (1 - h_{\text{TLB}})(t_{\text{ST}} + t_{\text{PT}}) + t_{\text{CACHE}} + (1 - h_{\text{CACHE}}) t_{\text{MAIN}}.
\]

\[
t_{\text{eff}} = 2.5 + (0.02)(2.5 + (.1)(52.5) + 2.5 + (.75)(52.5)) + 2.5 + (.05)(52.5)
\]

\[
= 2.5 + .02(2.5 + 5.25 + 2.5 + 39.375) + 2.5 + 2.625
\]

\[
= 8.6175 \text{ ns}.
\]

- Answer: b) We don’t need the virtual address translation time, so

\[
t_{\text{eff}} = 2.5 + (0.05)(52.5) = 5.125 \text{ ns}.
\]
Question 4. (15 points) Instruction execution time for a processor is 2 cycles and provides an on-chip cache with access time of 1 cycle. Each instruction requires, on average, 1 memory access for the instruction and 2 accesses for data. The internal cache is implemented as a Harvard cache. The processor can overlap instruction execution with access to the internal (L1) cache. The hit ratio to the L1 instruction cache is 92% and the hit ratio to the L1 data cache is 87%. A unified external (L2) cache requires 6 cycles to access and has a global hit ratio of 98%. The main memory requires 25 cycles to access. What is the average MIPS rate of the processor if it is clocked at 400MHz?

- Answer: CPI = CPI_{\text{IDEAL}} + CPI_{\text{FINITE Cache}}. Since we have an internal Harvard cache, we can completely overlap the instruction access of 1 cycle and data access of 2 cycles with instruction execution of 2 cycles for CPI_{\text{IDEAL}}. However, we must account for the miss rate through the memory hierarchy for both instructions and data.

\[
\begin{align*}
\text{CPI}_{\text{IDEAL}} & = 2 \text{ (overlapped with 1 instruction access and 2 data access)} = 2 \\
\text{CPI}_{\text{FINITE Cache}} & = \text{CPI}_{\text{inst miss}} + \text{CPI}_{\text{data miss}} \\
\text{CPI}_{\text{inst miss}} & = 1((.08)(6) + (.02)(25)) = 0.98 \\
\text{CPI}_{\text{data miss}} & = 2((.13)(6) + (.02)(25)) = 2.56 \\
\text{CPI} & = 2 + 0.98 + 2.56 = 5.54 \text{ cycles.}
\end{align*}
\]

At a clock rate of 400 MHz, \(400/5.54 = 72.2 \text{ MIPS}\).

Question 5. (10 points) A processor runs with clock rate of 200 MHz and attempts to make 1 memory reference per clock. The unified cache has an access time of 1 clock and has a hit rate of 95%. It takes 50 ns to access the first word from main memory and 80 ns to load the complete cache line. What is the average utilization of main memory? (Ignore trailing edge effects and contention.)

- At 200 MHz, the processor attempts to make 1 reference per 5 ns. With a hit rate of 95%, we have 19 references at 5 ns and 1 reference at 5 ns + 50 ns. The main memory is utilized while it is loading the complete cache line, which is 80 ns out of the time it takes the processor to make the 20 references. So, the memory utilization is \(80/(20*5 + 50) = 53\%\).
Question 6. (10 points) A new 64-megabit DRAM chip becomes available. You want to use it in a 16 Megabyte DIMM that has a data input/output width of 64 bits. What chip organization do you need?

- If you need 16 Megabytes in the DIMM and each chip is 64 megabits (64/8 = 8 megabytes), you will need 2 chips. A bus width of 64 bits requires each of the two chips to have 32 data lines in parallel. The chip organization must be $\frac{64\text{Meg}}{32 \times 32} = 2 \text{ meg x 32}$.

Question 7. (10 points) A synchronous DRAM is clocked at 133 MHz, and requires 8 clocks from RAS to first data out. The memory bus is 32 bits wide. How long does it take to load your cache line of 64 bytes?

- 64 bytes at 32 bits per word = 16 words to load the cache line. At 133 MHz, the clock period is 7.5 ns. The SDRAM needs 8 clocks for the first word and 1 clock per word for the remaining 15. Therefore, time = 8(7.5) + 15(7.5) = 172.5 ns.