ENPM607 Homework Problem Set No. 3

1. For the system shown in the following figure assume the following list of fixed priorities (in decreasing order of priority)

   **I/O-Bus:** (1) DRUM, (2) DISC, (3) DISPLAY, (4) MULTIPLEXOR  

   **Memory Modules:** (1) I/O-Bus, (2) CPU-Bus  

   and the following request rates (in words/memory-cycle)

<table>
<thead>
<tr>
<th>Unit</th>
<th>Average</th>
<th>Maximum</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>0.7</td>
<td>1.0</td>
<td>100% (if interference is zero)</td>
</tr>
<tr>
<td>DRUM</td>
<td>0.33</td>
<td>0.33 (when used)</td>
<td>55% (transfers only)</td>
</tr>
<tr>
<td>DISC</td>
<td>0.1</td>
<td>0.1 (when used)</td>
<td>40% (transfers only)</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>0.1</td>
<td>0.1</td>
<td>50% (transfers only)</td>
</tr>
<tr>
<td>MULTIPLEXOR</td>
<td>0.05</td>
<td>0.1</td>
<td>30% (transfers only)</td>
</tr>
</tbody>
</table>

   a. What is the average and the maximum required memory bandwidth if there is no interference?  
   b. How many cycles must the multiplexer wait to have access to the I/O-Bus in the worst case?  
   c. What is the memory bandwidth available to the CPU in the case of maximum interference at the memory module level?  
   d. Analyze the improvement in available memory bandwidth that would result from the use of a 4-way interleaving of the memory modules. (Hint: Use Hellerman's formula.)  
   e. What would be the average memory bandwidth if there were only 2-way interleaved modules?  
   f. Suppose a third bus is introduced and connected to all memory modules with I/O devices redistributed so that I/O-Bus 1 serves DRUM and MULTIPLEXOR, and I/O-Bus 2 serves DISC and DISPLAY (with priorities ordered left to right on each bus) and ordered I/O-Bus 1, I/O-Bus 2, CPU-Bus in decreasing priority left to right at the memory access controllers. Assuming 4-way interleaving, what is the average memory bandwidth?

![](image-url)
2. A computer system has a main memory (MM) with \(32K = 2^{15}\) words. It also has a \(4K = 2^{12}\) word cache (or buffer) memory organized in the block-set-associative manner with 4 blocks per set and 64 words per block.

   a. How many bits are in each of the TAG, SET and WORD fields of a MM address?

   \[
   \begin{array}{|c|c|c|}
   \hline
   \text{TAG} & \text{SET} & \text{WORD} \\
   \hline
   \end{array}
   \]

   b. In this organization mark all blocks initially invalid and suppose that the CPU fetches 4352 words from locations 0, 1, 2, \ldots, 4351, in consecutive order. It then repeats this fetch sequence 9 more times. The cache is 10 times faster than the MM, and the memory cycle time is \(\tau\) seconds. Thus, it takes the CPU \(0.1\tau\) seconds to access a word already in the cache. Because every word in a cache block will be fetched once the block is placed in cache and because load through is not used, it takes 64\(\tau\) seconds to move a 64 word block to cache and an additional 6.4\(\tau\) seconds to fetch all 64 words in the block. Thus, assume that it takes 1.1\(\tau\) seconds per word to obtain a word whose block is not already in cache. Note that this program occupies 68 consecutive MM blocks numbered 0-67.

   (i) List all MM block numbers in the above program that map into each of the following cache sets.

   Set 0 ______________________________________
   Set 1 ______________________________________
   Set 2 ______________________________________
   Set 3 ______________________________________
   Set 4 ______________________________________
   Set 5 ______________________________________
   etc.

   (ii) Assume that an LRU algorithm is used for block replacement, and define “improvement factor (IF)” as the ratio:

   \[
   \text{IF} = \frac{\text{time required without cache}}{\text{time required with cache}}.
   \]

   Calculate the improvement factor (IF) resulting from use of the cache on the above program execution.

   \[
   \text{IF} = ______________________________________
   \]
3. A certain program consists of two nested DO loops, a small inner loop and a much larger outer loop. The general structure of the program is given in the figure below. The decimal memory addresses shown delineate the location of the two loops and the beginning and end of the total program. All memory locations in various sections, 17–22, 23–164, 165–239, etc., contain instructions to be executed in straight-line sequencing, and each machine instruction comprises exactly one word. Assume that the computer on which this program is to be executed has a main memory (MM) size of $65,536 = 2^{16}$ words, and assume that the cycle time is $10\tau$ seconds. Suppose that this program is to be run on a computer that has a cache with cycle time of $10\tau$ seconds located between the CPU and the MM. The cache is organized in the direct mapping manner. The cache size is $1024 = 2^{10}$ words and the chosen block size is one word per block. Assume that load through is not used, so that $11\tau$ seconds are required for fetching an instruction into the CPU if it is not already in the cache. Assume that the cache is initially empty.

a. Specify the number of bits in each of the TAG, BLOCK, and WORD fields used by this organization in the interpretation of a MM address.

<table>
<thead>
<tr>
<th>TAG</th>
<th>BLOCK</th>
<th>WORD</th>
</tr>
</thead>
</table>

b. Ignoring the reads and writes associated with operand and result fetching and storing, compute the amount of time needed for instruction fetching in the following code segments, where the first and last addresses specified are inclusive:

i) 165–176: ______________________________________

ii) 177–239: ______________________________________

iii) 240–1046: _____________________________________

iv) 1189–1200: _____________________________________

v) 17–1500: ______________________________________
4. The IBM System 370 Model 165 is equipped with a cache memory which is dynamically maintained by the CPU and is transparent to the programmer.

The characteristics of the system are the following:

**CPU**
cycle time: 80 ns
data path width: 8 bytes (double word)

**Cache**
cache size: 8K bytes
cycle time: 80 ns
access width: 8 bytes/cycle (when a request is made, the first cycle is spent in checking whether the data are in the cache; therefore, 8 bytes can be obtained in 160 ns and 16 bytes in 240 ns)

**Main Memory**
memory size: 512K bytes (minimum configuration)
cycle time: 2 \( \mu s \)
access time: 1.44 \( \mu s \)
interleaving: 4 way (8 consecutive bytes can be obtained from each module)
access width: 32 bytes/cycle (8 bytes \( \times \) 4 modules)

Both the cache and the main memory are partitioned into blocks and columns as illustrated in Figure P4, attached. Any of the blocks in a main memory column can be placed in one of the four blocks in the corresponding cache column. Each element of the address array (see Figure P4) consists of the 13 high-order bits of the main memory address of the 32 bytes of data contained in the corresponding position of the cache. Each column in the cache is associated with a four-entry replacement array (not shown in Figure P4) which controls the replacement activity. When a cache block is referenced, it is put at the top of the list for its particular column. When a block must be reassigned, the entry at the bottom of the list is selected for replacement.

a. Specify whether or not this cache organization is one of the four described in the text and covered in class (they are also discussed in a paper by Conti). If not, comment on the cost and speed differences between it and those four organizations.

b. What fields in the effective address format would you consider to be relevant to cache operation? How long should they be?

c. Describe in a step-by-step fashion how, in your opinion, are these fields used at each memory request issued by the CPU.

d. Why do the entries of the address array consist of 13 bits each? What is the maximum size of the main memory?

(See Figure P4 on page 5)
Buffer Storage Organization (8K)

Figure P4.