16.482 / 16.561: Computer Architecture and Design
Fall 2013

Homework #7
Due **Monday, 12/2/13**

Notes:
- While typed submissions are preferred, handwritten submissions are acceptable.
- Any handwritten solutions that are scanned and submitted electronically <u>must</u> be clearly legible and combined into a single file—<u>simply sending a picture of each scanned page is not an acceptable form of submission</u>.

1. <u>Virtual memory</u> (30 points) Answer the following questions about a process that uses the page table below:

Virtual page #	Valid bit	Reference bit	Dirty bit	Frame #
0	1	1	0	3
1	1	1	1	1
2	0	0	0	--
3	1	0	0	2
4	0	0	0	--
5	1	0	1	0
6	0	0	0	--
7	1	1	0	5

a. (10 points) Which pages above are candidates to be evicted on a page fault? Which, if any, are better candidates to evict?

b. (20 points) Assuming 8 KB pages, what physical addresses would the virtual addresses below map to? Note that some virtual addresses may not have a valid translation, in which case you should note that address causes a page fault.
- 0xABCD
- 0x1792
- 0x4680
- 0x5701

2. Cache optimizations: Multi-banked/non-blocking caches (40 points) Assume we have
a system containing 64 blocks of memory, labeled B0-B63. (We do not need to know the
block size for this problem.) This system has a 16-line, direct-mapped cache that is
initially empty. Assume we have a program that accesses 20 of these blocks in the
following order, with one access initiated per cycle unless a stall occurs:

    0, 1, 3, 16, 4, 23, 8, 36, 27, 31, 63, 12, 28, 24, 42, 10, 14, 52, 51, 18

Note that, with an initially empty cache and no access to the same block twice, all of
these accesses will miss. You should assume each miss takes 10 cycles to handle. This
problem assesses the impact of several different cache organizations on those misses.

a. (5 points) Calculate the total time for these 20 accesses if the cache is not split into
banks and is therefore a blocking cache (i.e., only one access can be handled at a time).

b. (15 points) Calculate the total time for these 20 accesses if the cache is divided evenly
into four banks. Assume the blocks are not interleaved sequentially, so blocks are
mapped to cache lines using normal direct mapping. In other words, B0 maps to cache
line 0, B1 to cache line 1, and so on. Assume bank 0 holds cache lines 0-3, bank 1 holds
cache lines 4-7, etc.

c. (15 points) Calculate the total time for these 20 accesses if the cache is divided evenly
into four banks and the blocks are interleaved sequentially across those four banks.

d. (5 points) Determine the ideal mapping of memory blocks to cache banks for this
program (i.e., the mapping that gives you the minimum total time for all 20 accesses).

3. Cache optimizations: Early restart/critical word first (30 points) You are studying a
    system with the following memory characteristics. The 32-bit processor contains a
    single 64 KB direct-mapped cache with 256-byte blocks. A 32-bit bus connects the
    cache to 2 GB of off-chip memory. The bus is capable of supplying a word to the
    cache every 250 µs.

Determine the average latency for a cache miss under three conditions:
   • Sequential cache block fill (i.e., start with word 0 and fill all words in block)
   • Early restart
   • Critical word first with early restart

In addition, show the speedup over sequential block fill for both early restart and critical
word first with early restart. (In other words, how much faster is each of the enhanced
techniques than sequential block fill?) You should assume that the requested word is, on
average, in the middle of the cache block (since cache blocks hold an even number of
words, choose the first "middle" word—word 1 in a 4-word block, word 3 in an 8-word
block, etc.).