**Solution description**

1. How did you guarantee that each logical address is translated to the correct physical address?

Using this function extract\_logic\_address(logic\_address, &page\_num, &offset);, we extracted the data from main.cpp

According to address.cpp, we used a sample Logical address: 49454, page number: 193, offset: 46 and received the following output:

logical address: 1100000100101110

page number: 11000001

offset: 00101110

Unit Testing: Now create physical address ...

frame number:193, offset: 46, physical address: 49454

Using the three parameters Page Size, frame number and page offset, we were able to guarantee a physical address

1. How did you implement the page table, physical memory, and TLB?

The TLB which we have used in our project is implemented as an array of pages, with either FIFO or LRU. The TLB points to both the physical memory and page table for data flow purposes. The physical memory contains an array of frame blocks each of which contain an array of single bytes whereas the page table is just a simple array of pages.

1. Does your program realistically and accurately simulate a virtual memory system?

We think our program realistically simulates a virtual memory system, as it only references the page table on a TLB miss and only loads frames from the backing store on a page fault.

1. Did you use the Java operators for bit-masking and bit-shifting?

Yes, it was certainly used

1. When a TLB miss occurs, how do you decide which entry to replace?

When a TLB miss occurs, the entry that gets replaced on a TLB miss would be either FIFO or LRU. We implemented these strategies based on what was discussed in class. The implementations were not completely successful but was close to completion

**Generality and Performance Criteria**

1.Our solution is quite general and everything is used in the most simplistic manner as possible.

2.Our solution is not ground breaking, or significantly more efficient than the next.

3. We do believe that the backing store load pages only when its needed.

4. Yes, because virtual addresses are originally given in 32 bits and are

  eventually converted into 8 bit frame numbers and 8 bit offsets.

**Miscellaneous factors**

(1) Is your code elegant? Our code isn’t the most elegant but we do think that it has some elegance.

(2) How innovative is your solution? Did you try any ideas not suggested here (e.g. a choice of replacement policies for the TLB)? Innovative ideas that go beyond the requirements could receive extra credit.

We designed Replacement factors such as LRU and FIFO which are quite innovative.

(3) Did you document all outside sources?

Reference:

https://bytes.com/topic/c/answers/621985-print-binary-representation

ASSESSMENT: We couldn’t complete entirely due a lack of time, and a lot to try to do with the time that we had.