# Homework 4

## Policies:

- The homework is designed to be done in groups of three.
- Using material from the Internet including text and code are permitted provided that the right attribution is made, and that the student understands thoroughly what is being used.
- Homework due by 5PM on Nov 24, 2023.

## Advice

Start early!!

## Understanding Caches and TLB—Sensitivity Analysis of Cache Systems

In this exercise, you will study the behavior of TLB and cache systems. We will use a trace driven simulator for the TLB and for a 2-level cache system. The traces follow the Dinero format (from the University of Wisconsin). The traces to be used in this exercise are provided with the homework document. You will need to write the following components:
- A page allocator. The addresses in the file are all virtual addresses. Upon encountering a virtual address that has not been allocated a physical frame, a new frame and mapping are allocated. A corresponding entry in the page table is made. Feel free to order the physical frames in a list.
- A page table: We will use a 3-level page table of 6/8/6 bits. We will fix the page size to 4KB.
- A simulator for the D1 cache.
- A simulator for the D2 cache.

There are many public codes for TLB and cache simulators that are available. Feel free to use any package if you wish, or partially import some code into your code, or write your own from scratch. You can use whatever programming language you wish.
The operation:
- Each virtual address will be presented to the TLB. If found, the corresponding physical address is given to the L1. If not found, then simulate the page table walk as explained in the lecture. The page table entries are in memory, and therefore will be accessed through the cache. If the address is not allocated in the page table, then your page allocator needs to allocate a physical page, and include all the necessary intermediate level page tables as explained above.

- If the cache contains the item, then you move to the next address. If not, then there is a cache miss. You will need to simulate the fetching of the data as described in the lecture.
- Assume that you have physical starting from address 0, and that the page size is 4KB.

There are three parameters of interest in this exercise:
- The cache line size.
- The size of the TLB.
- The sizes of the L1/L2 cache.

You will estimate the sensitivity of performance to the variation of the three above parameters. The performance will be measured by the number of cache and TLB misses, and by the average latency of memory access. Your assignment is to produce a performance report on the following, with pictures and some analysis. An array of measurements will be most appropriate along the three dimensions described. The sensitivity analysis will show the rate of performance improvement compared to the rate of change in the input parameters. The following cases are to be studied:
- Cache line size of 32-byte, 64-byte and 128-byte.
- Cache size of 32KB and 64KB for the L1, (1 cycle and 2 cycle access time, respectively), and L2 (512K, 1MB and 2MB, with 8, 12 and 16 cycles respectively). Memory is at 100 cycles away.
- TLB of 8 and 16 entries, accessed at 1 cycle. The TLB is fully associative.

Produce the report for the following scenarios:

Case 1- L1 is split in half between instructions and data.

Case 2- Cache replacement algorithm is random, versus LRU. versus FIFO. Use a split L1 cache. Make the L2 4-way set associative.

Case 3- L2 is direct mapped, versus 2-way and 4-way set associative. Use the random replacement policy and a split L1 cache.

The L1 is always direct mapped. Assume a 32-bit physical address space.

In all cases, please report the number of hits and miss for the TLB, L1 and L2. Also, compute the average access time. Compare the impact of the TLB misses compared to data and instruction misses in the cache.

Hint: The assignment looks overwhelming. However, once the simulator is written (or configured if you will import a simulator from outside), then the running of the studies will be very instructive (and rewarding!).