CS230: Digital Logic Design and Computer Architecture
Tutorial 11, [Mon 04 Nov, Tue 05 Nov, Thu 07 Nov]

1. (Based on 7.37) Rank each of the following event combinations according to how frequently they are likely to occur. Explain your answer briefly.
   (a) TLB miss followed by page fault followed by cache miss
   (b) TLB miss followed by page hit followed by cache miss
   (c) TLB hit followed by page hit followed by cache miss
   (d) TLB miss followed by page hit followed by cache hit
2. A computer has a 32-bit virtual address space, and a 24-bit physical address space. It has 8KB pages. The TLB is shared between instructions and data. It has 256 entries and is 8-way set associative. The TLB block size is one entry.
   (a) Draw a diagram showing the various TLB fields, and the mapping from virtual memory to physical memory, on a TLB hit.
   (b) How will the TLB fields change if the TLB were fully associative? If it were direct-mapped?
3. Describe the aliasing problem with virtually addressed caches.
4. Mr. Buddhimaan, after having done the computer architecture course, proposes to solve the aliasing problem with virtually addressed caches as follows. On each process context switch, flush the cache. Will this solve the aliasing problem? Why or why not?
5. What is done to prevent the requirement to flush the TLB on each context switch? Explain briefly.
6. **Hit-miss possibilities:** Consider a computer with a TLB, L1 cache, L2 cache, and virtual memory pages. Now, TLB is a cache for page-table entries, L1 is a cache for L2 blocks, L2 is a cache for main memory blocks, and main memory pages are a cache for pages on disk. During program execution, a

particular memory access potentially involves the above four caches, and each of caches can result in a hit or a miss. So overall there are $2^4 = 16$ combinations. List these combinations, and comment on each. Also indicate which combinations are impossible and why.

7. **Stuck at TLB:** Suppose a TLB implements a most-recently-used eviction policy for its entries. Construct a situation where the processor *cannot* execute an instruction. State the necessary conditions under which this will happen.

8. **Non-restartable instruction:** In the context of page-faults, a restartable instruction is one which can be restarted after handling a page-fault which was caused during the instruction's execution. In the MIPS subset we have seen, all instructions are restartable. The Intel IA32 architecture has a single instruction which can perform string copy (strictly speaking, it is memcpy). Prove that this instruction is non-restartable. *Hint: think of some special cases in the specification of the origin and destination strings.*