1. There were 3 hits and 7 misses. My sequence was 3E, 86, 9B, F3, A8, 15, DC, D3, 37, 36. Basically the request would split the 8 bit address into 3bit/5bit. The first 3 bit would correspond to the page table number, from 0 to 7. If the corresponding page table slot was associated with a page frame in physical memory, it was a hit. If not, a new page frame would be loaded in, sometimes kicking out the old one like caches do. There was no page hits because if there was a hit it was caught in the TLB so the page wasn’t checked.
2. 0B, 2B, 4B, 6B, 8B, AB, CB, EB, 0B, 2B
3. The idea is to have more frame numbers in the page table than translations in the buffer. One way to do this would be to decrease the size of the TLB size while keeping the replacement rate of the page table the same. Thus, there may be 4 items in the page table but only 3 in the TLB, resulting in a potential TLB miss while getting a Page Table hit.
4. There are four “threads?” and they all compete for space in memory without communicating with each other. Thread 1 will perform some actions, then thread 2 will ignore what is already in the page file of thread 1, and eventually thread 4 will overwrite some things in the other threads, making it impossible not to completely start over each cycle.