Ryan Li

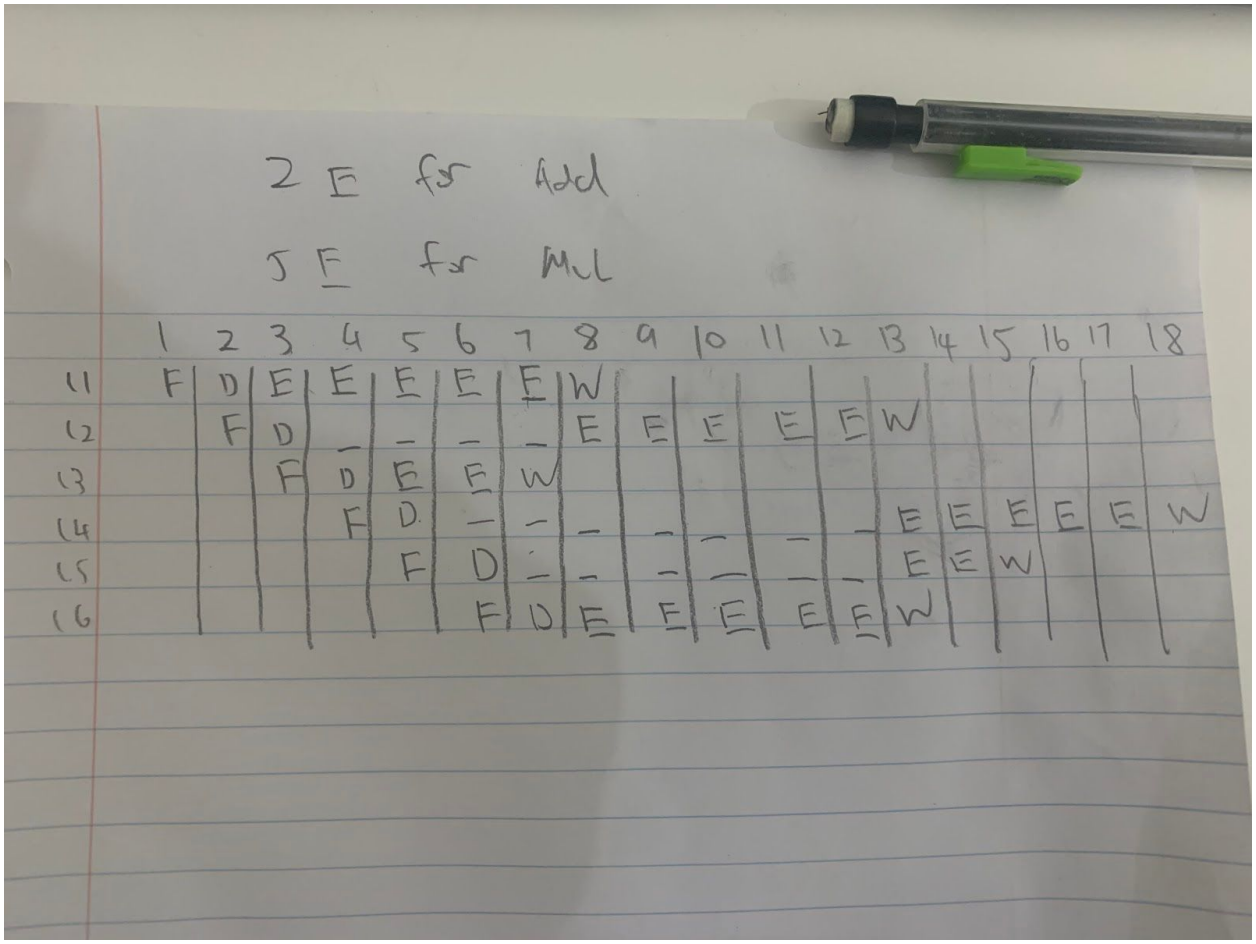
Question 1

1. my favorite topic from the second half of the course is OutofOrder execution. One key takeaway from this class is how difficult being a computer architect would be and how competitive hardware companies must be to keep innovating.
2. cache AMAT: 2 + 0.05 * 200
   TLB AMAT: 1 + 0.1 * 200
3. SIMD will not help code snippet A or B because in A[i][j] = A[i][j-1] * 2 or A[i][j] = A[i - 1][j] * 2 we need to access another element from the array we are currently looping over, so it is not element wise. We cannot partition cleanly the data so that we can run the same instruction on different data at the same time.
4. a) 2 banks because the bank index is 1 bit which can only specify 2^1 banks.
   b) L2 cache cannot serve these requests in parallel because their bank index bits are the same so they cannot both access the same bank at the same time.
5. a) False, non blocking caches reduce miss latency but not hit rate.
   b) True, all these methods aim to increase the number of instructions that can be executed simultaneously
   c) False, atomic instructions complete their execution in a singe step/cycle so nothing else can mess with what they are reading or writing.
   d) False, if all the instructions were serial than this would be the case, but (1-S) of the work is infinitely parallelizable so although each B processor runs at half the speed of the A processor, it provides more instruction level parallelism and will thus have lower than T execution time.

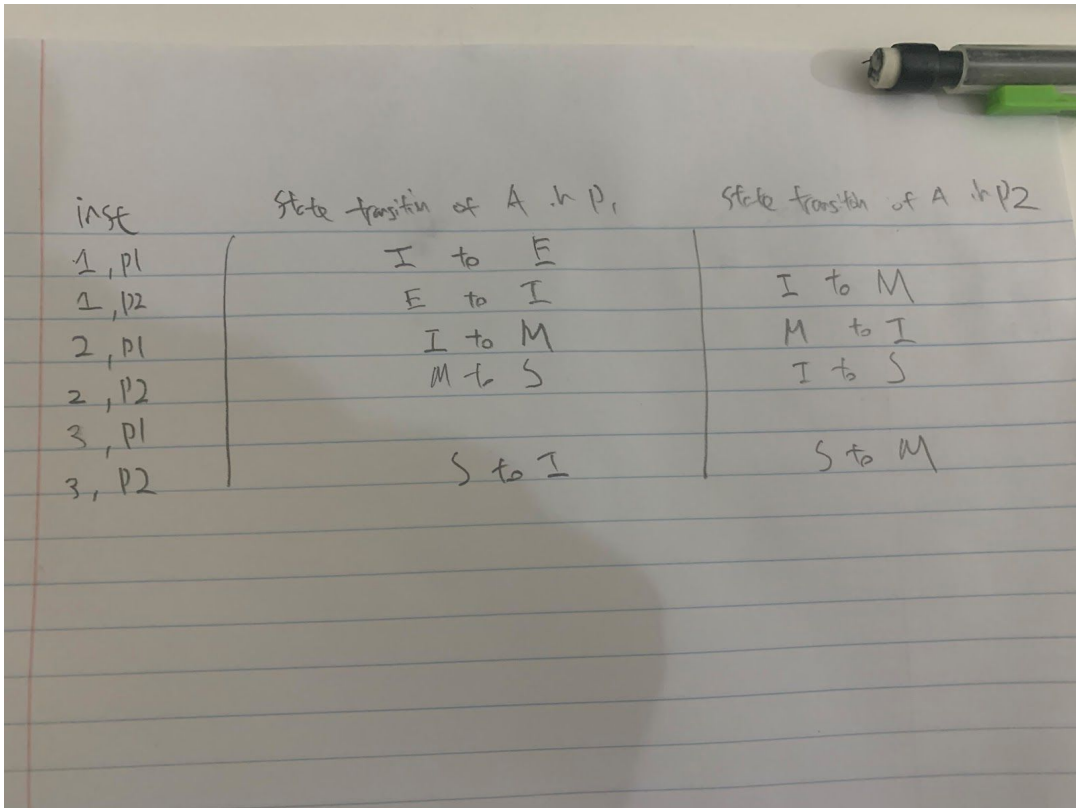
Question 2

1. one entry in the first level page directory can map to 2^10 entries in the second level, each entry in the second level links to a page, page size is 4KB, so 2^10 * 4KB = 4MB.
2. The page directory has 2^10 entries and each entry is 32 bits. It's size is 2^10 * 32 bits. One second level page table has 2^10 entries and each entry is 32 bits so its size also 2^10 * 32 bits. In total, they take 65536 bits which is 8KB.
3. A TLB entry includes 32 bits of PTE, 1 valid bit, 1 dirty bit, 2 LRU bits because it is four way associate, and the rest(28 bits) is tag. This add up to 64 bits or 8 bytes per entry. 1024 * 8 = 8192 bytes.

Question 3


2 E for Add
5 F for Mul
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
I1 F D E E E E E W
I2 F D _ _ _ _ E E E E E W
I3 F D E E W
I4 F D _ _ _ _ _ _ _ E E E E E W
I5 F D _ _ _ _ _ E E W
I6 F D E E E E E W

Question 4


inst
State transition of A in P1
State transition of A in P2
1, P1
I to E
1, P2
E to I
I to M
2, P1
M to I
2, P2
I to S
3, P1
S to I
S to M
3, P2