Justin Vargas
Lab - 4L
TPS ACTIVITY 1

1. Cache is a copy of a subset of main memory so that it can be accessed quickly in the future. We need cache in order to improve the speed of tasks. It is quicker for the CPU to check the cache rather than go back and access the main memory.

4. If block size is 16B the offset needs 4 bits. The number of bits in offset as a function of block size is $\log_2 n$, n is the block size. A block size of 1 byte would be practical because then the offset would just be 0.

5. For a 64 block cache size there needs to be 6 index bits. The number of bits in index as a function of number of blocks $\log_2 n$, where n is the number of blocks in the cache.

6. The total size of the cache can be calculated by multiplying the number cache blocks with the size of each block.

7. The tag bits can be used to identify the address in the main memory. Index is the same for every address within a block so it is more efficient to leave it off of the tag to save memory.

8. Given a memory address of 20 bits (during Intel 8086 era), 128B of cache, and 8B block size

a. Main memory size = 2^{20} = 1MB
b. Offset bits = 3
c. Number of blocks in cache = cache size/block size = 128B/8B = 16 blocks
d. Index bits = 4
e. Tag bits = 13
f.

Tag	Valid	Dirty	Data
13 bit	1 bit	1 bit	64 bit

g. Number of bits per row = 13 + 64 + 1 + 1 = 79

TPS ACTIVITY 2

1. The main disadvantage of a direct-mapped cache is that multiple memory addresses map to the same cache index. If two block reference the same location in cache then they are both being accessed. Which will cause them to be continuously swapped in and out. It will introduce the conflict cache miss.

3. Number of sets = number of blocks / associativity = 1024/4 = 256
   Index bits = 8
   Index bit as a function of number of blocks and associativity = $\log_2$(set amount)

4. Given a memory address of 20 bits (during Intel 8086 era), 128B of 2-way cache, and 8B block size

   a. Main Memory = 2^{20} = 1MB
   b. Offset Bits = 3
   c. Blocks in cache = cache size/block size = 128B/8B = 16
   d. Sets in cache = 8
   e. Index bits = 4
   f. Tag bits = 13
   g.

Tag	Valid	Dirty	Data
14 bit	1 bit	1 bit	64 bit

   h. Number of bits per row = 14+1+1+64 = 80 bits

Assignment

1. My CPU has 3 levels of caches. L1 Data, L1 Instruction, L2 Unified, L3 Unified.
2. SIZE: L1 Data = 32KB, L1 Instruction = 32KB, L2 Unified = 256KB, L3 Unified = 6MB.
3. BLOCK SIZE: L1 Data = 64B, L1 Instruction = 64B, L2 Unified = 64B, L3 Unified = 64B.
4. MAPPING: L1 Data = Direct, L1 Instruction = Direct, L2 Unified = Direct, L3 Unified = Complex.
5. L1 tag bits = 17, L1 index bits = 9, L1 offset bits = 6