Assignment #5
Matthew Park
34017103 o5u7

1.

a.

14	13	12	11	10	9	8	7	6	5	4	3	2	1
CT	CT	CT	CT	CT	CT	CT	CT	CI	CI	CI	CO	CO	CO

b.

0x0362

Tag	Index	Offset
000001101	100	010
00D	4	2

Cache Set:	4
Tag:	D
Hit?	Miss - Conflict
Value:	Unknown

0x7BFE

Tag	Index	Offset
111101111	111	110
1EF	7	6

Cache Set:	7
Tag:	EF
Hit?	Miss - Cold
Value:	Unknown

0x3FC9

Tag	Index	Offset
011111111	001	001
0FF	1	1

Cache Set:	FF
Tag:	1
Hit?	Miss – Cold
Value:	Unknown

0x0361

Tag	Index	Offset
000001101	100	001
D	4	1

Cache Set:	4
Tag:	D
Hit?	Hit
Value:	Unknown

2. See .c code

3.
a. The miss rate should be around 25%. The first cache entry will always be a miss, but when it loads a 16-byte offset cache line that holds 4 byte integers, it will also load the next three integers plus the one it was originally loading. So when sumA reads the next sequential integer of the array, the next 3 integers will be stored into cache. The miss rate is approximately 25%, because the first integer it loads into cache could be right in the middle of the first set instead of the beginning.

b. The miss rate should be 100%. The multi-dimensional array is a row major and sumB is reading the next value of the array as if it were column major. The consequence is that the cache is not being able to experience spatial locality, as the next index of the multi dimension array are always the distance of the size of the rows. Therefore when loading an integer into cache, the integers with consecutive addresses will not be needed anytime soon.

c. The miss rate should be 50%. sumC is reading the array bye a 2x2 square each time. The result is that after the first cold miss, the next integer is the next integer with the consecutive address. This process repeats with the third and fourth integers. In summary, every second integer is return a hit in the cache, or in other words hit 50% of the time.

d. The miss rate should be around 25%. It will follow similarly to question 3a, mainly due because 68 does not divide well into 16. When filling out set of the array, the columns address will be staggered and the bit addresses are long enough that it should not interfere with the previous column.

e. The miss rate should be around 75%. The first column of the array will result in cold misses, but the following three columns will result a third of their columns being hits. This is due to the fact, there are 12 sets (48/4=12) and in hexadecimal, every 12 number will result the digit to be 0, 4, or C. Thus the number of hits for every 4 columns is:

16 hits * 3 columns / 48 (the length of a column) * 4 columns = 25% hit rate, or a 75% miss rate.

f. The miss rate should be around 100%. Number of sets will now be cut in half. From the previous question, a third of every second, third and fourth columns will be hit. This is every 16th (assume the column starts at index 0) to 31st column is a hit. Seeing that the sets will be cut in half and 48/16 = 3, which is greater than 2, all searches in the cache will be have been replaced causing nothing but misses.

This assignment took me a total of 5 hours.