1.) Suppose you are creating a cache with a data capacity of 32KBytes where each cache block is 64 Bytes. Assuming a 32-bit address, find the total number of bits needed to store both data and meta-data if the cache is organized as a direct-mapped cache. Perform the same calculations for a 2-way set associative cache. Show your work.

*# blocks = 32*1024/64 = 512*

Bits to indicate where in the block: lg 64 = 6

*Total data bits: 32*1024*8 = 262144*

Direct mapped:

# of sets: 512

Bits to indicate index: lg 512 = 9

Tag bits: 32 - index bits - bits indicating where in block = 32 - 9 - 6 = 17

Per block meta data bits: tag bits + valid bit = 17 + 1

*Total meta data bits: 18 * 512 = 9216*

Total bits: 262144+9216 = 271360

2-way set associative:

# of sets: 512/2 = 256

Bits to indicate index: lg 256 = 8

Tag bits: 32 - 8 - 6 = 18

LRU bits: lg 2 = 1

Per block meta data bits: tag bits + valid bit + LRU bits: 18 + 1 + 1 = 20

*Total meta data bits: 20 * 512 = 10240*

Total bits: 262144+10240 = 272384

2.) How many bits are used for tag, index, and block fields given a 32 bit address and a 4096KByte 8-way set associative cache with 256Byte lines? Show your work.

# of bytes in block offset = lg (block size) = lg 256 = 8 bits for block offset
*Cache capacity = 4096*1024 Bytes*
*# of blocks in cache = capacity / block size = 4096*1024 / 256 = 16384 blocks*
# of sets in cache = # of blocks in cache / associativity = 16384 / 8 = 2048 sets
# of bits needed to specify index = lg (# of sets) = 11
# of tag bits = 32 – index bits – block offset bits = 32 – 11 – 8 = 13

3.) You are given a 64 byte cache with a 4 byte cache block size. It is direct mapped.
a) Label the following references as hits or misses.

Addr (in decimal)	Addr (in hexadecimal)	Hit or Miss
13	D	
116	74	
161	A1	
112	70	
15	F	
258	102	
129	81	

257	101	
75	4B	
256	100	
64	40	
14	E	

b) Show the final state of the cache. (You should specify the range of byte addresses in each row of the cache.)

*# of sets = capacity / (associativity * block_size) = 64 / (1 * 4) = 16*

Addr (in decimal)	Addr (in hexadecimal)	*Block Offset*	*Set #*	Hit or Miss
13	D	*1*	*3*	*M*
116	74	*0*	*13*	*M*
161	A1	*1*	*8*	*M*
112	70	*0*	*12*	*M*
15	F	*3*	*3*	*H*
258	102	*2*	*0*	*M*
129	81	*1*	*0*	*M*
257	101	*1*	*0*	*M*
75	4B	*3*	*2*	*M*
256	100	*0*	*0*	*H*
64	40	*0*	*0*	*M*
14	E	*2*	*3*	*H*

0	*[256-259] -> [128-131] -> [256, 259] -> [64, 67]*
1	
2	*[72, 75]*
3	*[12,15]*
4	
5	
6	
7	
8	*[160-163]*
9	
10	
11	
12	*[112-115]*
13	*[116-119]*
14	
15	

c) What is the miss rate for this stream of references?

9/12

4) You are given a 64 byte cache with a 4 byte cache block size. It is **4-way set associative**. Assume LRU replacement.

a) Label the following references as hits or misses.

Addr (in decimal)	Addr (in hexadecimal)	Hit or Miss

13	D	
116	74	
161	A1	
112	70	
15	F	
258	102	
129	81	
257	101	
75	4B	
256	100	
64	40	
14	E	

b) Show the final state of the cache. (You should specify the range of byte addresses in each row of the cache.)

*# of sets = capacity / (associativity * block_size) = 64 / (4 * 4) = 4*

Addr (in decimal)	Addr (in hexadecimal)	*Block Offset*	*Set #*	Hit or Miss
13	D	*1*	*3*	*M*
116	74	*0*	*1*	*M*
161	A1	*1*	*0*	*M*
112	70	*0*	*0*	*M*
15	F	*3*	*3*	*H*
258	102	*2*	*0*	*M*
129	81	*1*	*0*	*M*
257	101	*1*	*0*	*H*
75	4B	*3*	*2*	*M*
256	100	*0*	*0*	*H*
64	40	*0*	*0*	*M*
14	E	*2*	*3*	*H*

0	*[160, 163] -> [64, 67]*
0	*[112, 115]*
0	*[256, 259]*
0	*[128, 131]*
1	*[116, 119]*
1	
1	
1	
2	*[72, 75]*
2	
2	
2	
3	*[12, 15]*
3	
3	
3	

c) What is the miss rate for this stream of references?

8/12

5.) You are given a 16 byte cache with a 4 byte cache block size. It is **fully associative**. Assume LRU replacement.
a) Label the following references as hits or misses.

Addr (in decimal)	Addr (in hexadecimal)	Hit or Miss
13	D	
116	74	
161	A1	
112	70	
15	F	
258	102	
129	81	
257	101	
75	4B	
256	100	
64	40	
14	E	

b) Show the final state of the cache. (You should specify the range of byte addresses in each row of the cache.)

Addr (in decimal)	Addr (in hexadecimal)	*Block Offset*	Hit or Miss
13	D	*1*	M
116	74	*0*	M
161	A1	*1*	M
112	70	*0*	M
15	F	*3*	H
258	102	*2*	M
129	81	*1*	M
257	101	*1*	H
75	4B	*3*	M
256	100	*0*	H
64	40	*0*	M
14	E	*2*	M

0	*[12, 15] -> [64, 67]*
0	*[116, 119] -> [256, 259]*
0	*[160, 163] -> [128, 131]-> [12, 15]*
0	*[112, 115] -> [72, 75]*

c) What is the miss rate for this stream of references?
9/12