The output of the command `getconf -a | grep CACHE` is

```
LEVEL1_ICACHE_SIZE                 32768
LEVEL1_ICACHE_ASSOC
LEVEL1_ICACHE_LINESIZE             64
LEVEL1_DCACHE_SIZE                 32768
LEVEL1_DCACHE_ASSOC                8
LEVEL1_DCACHE_LINESIZE             64
LEVEL2_CACHE_SIZE                  524288
LEVEL2_CACHE_ASSOC                 8
LEVEL2_CACHE_LINESIZE              64
LEVEL3_CACHE_SIZE                  8388608
LEVEL3_CACHE_ASSOC                 0
LEVEL3_CACHE_LINESIZE              64
LEVEL4_CACHE_SIZE
LEVEL4_CACHE_ASSOC
LEVEL4_CACHE_LINESIZE
```

1. There are three levels of cache
2. Cache size are given in the table below

Level	Size
1	32768(instruction cache) + 32768(data cache)=65536 bytes = 64 KiB
2	524288 bytes = 512 KiB
3	8388608 bytes = 8 MiB

3. Associativity of Level-1 cache (Instruction) = 0
   Associativity of Level-1 cache (data) = 8
   Associativity of Level-2 cache = 8
   Associativity of Level-3 cache = 0

4. My CPU has 6 cores. Since L1 and L2 have 6 instances, one for each core we would have
   6 * (64 KiB) + 6 * (512 KiB) + 8 MiB = 8 MiB + 3 MiB + 384 KiB = 11.384 MiB

Assignment - 4

AI20BTECH11006

Problem 2)

256 bytes : cache size

8 bytes block size

No. of blocks = 256/8 = 32 blocks

Index size = log 32 = 5 bits

bytes offset = log 8 = 3 bits

Address size = 12 bits (3 hex digits)

Tag size = 12 - 5 - 3 = 4 bits

tag is first 4 bits, index is next 5 bits.

(a) 0x005 [0000 | 0000 | 0101]

Initially valid bit is 0. So, it will be a miss. Tag 0000 will be stored in 00000

(b) 0x124 [0001 | 0010 | 0100]

Initially valid bit is 0. So, it will be a miss. Tag 0001 will be stored in 00100.

(c) 0x352 [0011 | 0101 | 0010]

Initially, valid bit is 0. So, it will be a miss. Tag 0011 will be stored in 01010.

(d) 0x224 [0010 | 0010 | 0100]

~~Initially, valid bit is 0.~~ The tag at 00100 is 0001. It is a miss. Now, the tag stored there will be 0010

e) 0x126 [0001|0010|0110]

The tag at 00100 is 0010.

It is a miss

Now, the tag stored 00100 will now be 000

f) 0x350 [0011|0101|0000]

The tag at 01010 is 0011, so, it will be a hit.

g) 0x000 [0000|0000|0000]

The tag at 00000 is 0000. It will be a hit

h) 0x025 [0000|0010|0101]

The tag at 00100 is 0001. It will be a miss. The tag at 00100 will now be 0000

i) 0x102 [0001|0000|0010]

The tag at 00000 is 0000. It will be a miss. The tag at 00000 will now be 0001

j) 0x120 [0001|0010|0000]

The tag at 00100 is 0000. It will be a miss. The tag at 00100 will now be 0001

k) 0x200 [0010 | 00000 | 0000]

The tag at 00000 is 0001, ~~the tag at~~ it will be a miss. The tag at 00000 will now be 0010

l) 0x108 [0001 | 00001 | 1000]

The valid bit at 00001 is 0. It is a miss. The tag at 00001 is now 0001

m) 0x0000 [0000 | 00000 | 0000]

The tag at 00000 is 0010. It is a miss. The tag at 00000 will now be 0000

n) 0x202 [0010 | 00000 | 0010]

The tag at 00000 is 0000. It is a miss. The tag will now be 0010

o) 0x128 [0001 | 00101 | 1000]

The ~~tag~~ valid bit at 00101 is 0. It is a miss. The tag will now be 0001

p) 0x350 [0011 | 01010 | 0000]

The tag at 01010 is 0011. It will be a hit.

~~The hit rate is:~~

The number of hits are 3

The number of misses 13

hit rate $= \frac{3}{16} \times 100 = 18.75\%$

miss rate $= \frac{13}{16} \times 100 = 81.25\%$

Problem 3 .

cache size, block size, replacement policy is the same

No. of sets = $\frac{256}{2 \times 8}$ = 16 sets

# of set size = $\log_2 8$ = 3 bits

Index size is now $\log_2 16$ = 4 bits

∴ tag size = 12 - 4 - 3 = 5 bits

The tag is the first 5 bits in each, index in next 4 bits

a) 0x005 [0000 | 00000 | 0101]

valid bit of set 0000 is 0. It is a miss. The set will now store 00000

b) 0x124 [0001 | 0010 | 0100]

valid bit of set 0100 is 0. It is a miss. the set will now store 00010.

c) 0x352 [0011 | 0101 | 0010]

valid bit of set 1010 is 0. It is a miss. The set will now store 00110.

d) 0x224 [0010 | 0010 | 0100]

set 0100 contains tag 00010. It is a miss. The set will store the previous value with low priority & current value with higher priority (least recently used) that is 00100

e) 0x126 : [0001 | 0010 | 0110]

The set 0100 has 00010 & 00100. It is a hit. we make 00010 as recently used tag.

f) 0x350 [0011 | 0101 | 0000]

set 1010 has tag 00110. It is a hit.

g) 0x000 [0000 | 0000 | 0000]

set 0000 has tag 00000. It is a hit

h) 0x025 [0000 | 0010 | 0101]

The set 0100 has tags 00010 & 00100. It is a miss. 00100 was least recently used. So, it will be replaced with 00000

i) 0x102 [0001 | 0000 | 0010]

The set 0000 has tag 00000. It is a miss. The tag 00010 will now be stored in set 0000

j) 0x120 [0001 | 0010 | 0000]

The set 0100 has tags 00000, 00010. ∴ it is a hit. 00010 will become recently used tag.

k) 0x200 [00101 | 0000 | 0000]

set 0000 has 00000 & 00010. It is a miss so it will be replaced by 00100

l) 0x108 [0001 | 0000 | 1000]

set 0001 has valid bit 0. It is a miss

Tag 00010 will now be stored in set 0001

m) 0x000 [0000 | 0000 | 0000]

set 0000 has 00100 & 00010. It is a miss 00000 will replace 00010 in cache as it was least recently used

n) 0x202 [0010 | 0000 | 0010]

The set 0000 has 00000 & 00100. It is a hit. 00100 will become recently used tag.

o) 0x128 [0001 | 0010 | 1000]

The set 0101 has valid bit 0. It is a miss. The set 0101 will now contain tag 00010.

p) 0x350 [0011 | 0101 | 0000]

The set 1010 contains 00110. It is a hit

No. of hits = 6

No. of misses = 10

hit rate $= \frac{100 \times 6}{16} = 37.5\%$

miss rate $= \frac{100 \times 10}{16} = 62.5\%$