Name: Vin Sovichhay

ID: e20241261

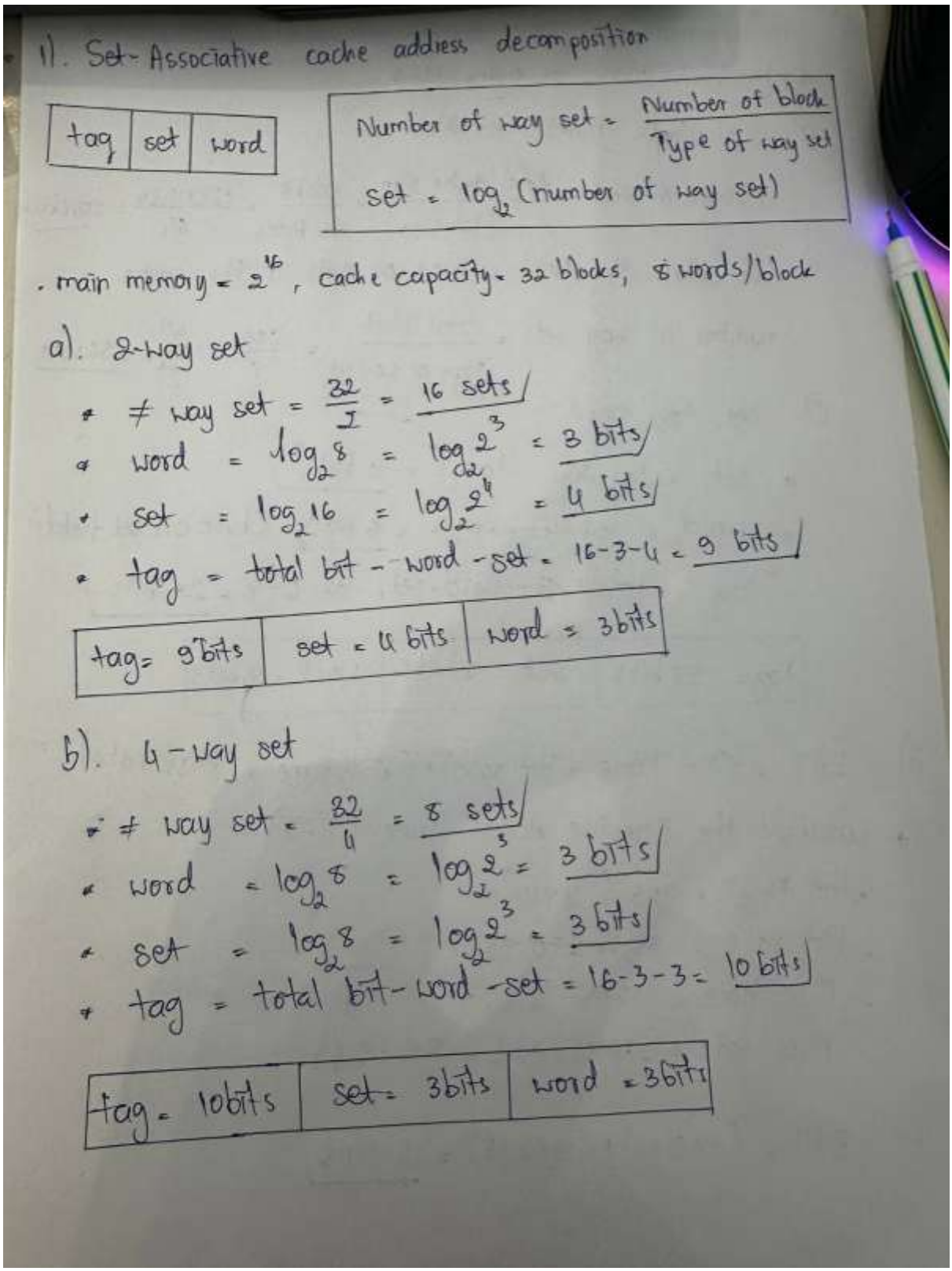
Group: I2-AMS2B

TP8: Set-Associative Cache

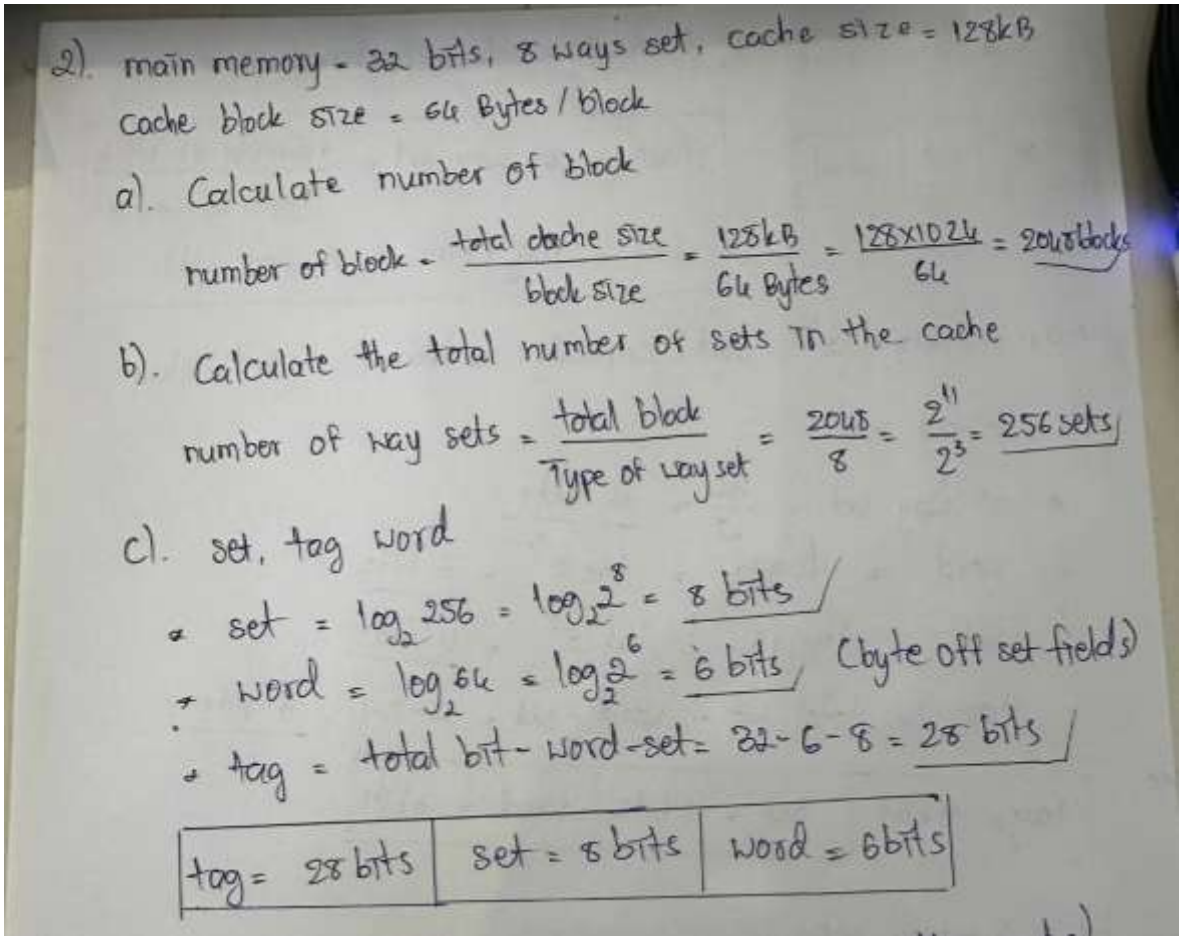
1) **Set-Associative Cache Address Decomposition (Scenario 1)**: Suppose a computer system uses a set-associative cache. The main memory contains 2^{16} words. The cache has a total capacity of 32 blocks, and each cache block contains 8 words.
   a) If this cache is 2-way set-associative, determine the format of a memory address as seen by the cache. Specify the number of bits required for the tag, set index, and word offset fields.
   b) If this cache is 4-way set-associative, determine the format of a memory address as seen by the cache, again specifying the sizes of the tag, set index, and word offset fields.
2) **Set-Associative Cache Address Decomposition (Scenario 2)**: Consider a system with a 32-bit memory address space. It features an 8-way set-associative cache with a total size of 128 KB and a cache block size of 64 Bytes.
   a) Calculate the total number of blocks in the cache.
   b) Calculate the total number of sets in the cache.
   c) Show the format of the 32-bit memory address by specifying the number of bits for the tag, set index, and byte offset fields.
3) **Effective Access Time (EAT)**: A processor accesses memory through a Level 1 (L1) cache. The L1 cache has an access time (hit time) of 2 ns and achieves a hit rate of 95%. Accessing main memory directly takes 100 ns (this is the miss penalty if only L1 cache exists). Calculate the Effective Access Time (EAT) for this single-level cache system.
4) A direct-mapped cache consists of eight blocks. Main memory contains 4K blocks of eight words each. Access time for the cache is 22 ns and the time required to fill a cache slot from main memory is 300ns (this time will allow us to determine the block is missing and bring it into cache). Assume a request is always started in parallel to both cache and to main memory (so if it is not found in cache, we do not have to add this cache search time to the memory access). If a block is missing from cache, the entire block is brought into the cache and the access is restarted. Initially, the cache is empty.
   a) Show the main memory address format that allows us to map addresses from main memory to cache. Be sure to include the fields as well as their sizes.
   b) Compute the hit ratio for a program that loops 4 times from locations 0 to $(67)_{10}$ in memory.
   c) Compute the effective access time for this program.
5) Consider a byte-addressable computer with 24-bit addresses, a cache capable of storing a total of 64K bytes of data and blocks of 32 bytes. Show the format of a 24-bit memory address for:
   a) Direct mapped
   b) Fully associative
   c) 4-way set associative

Solution

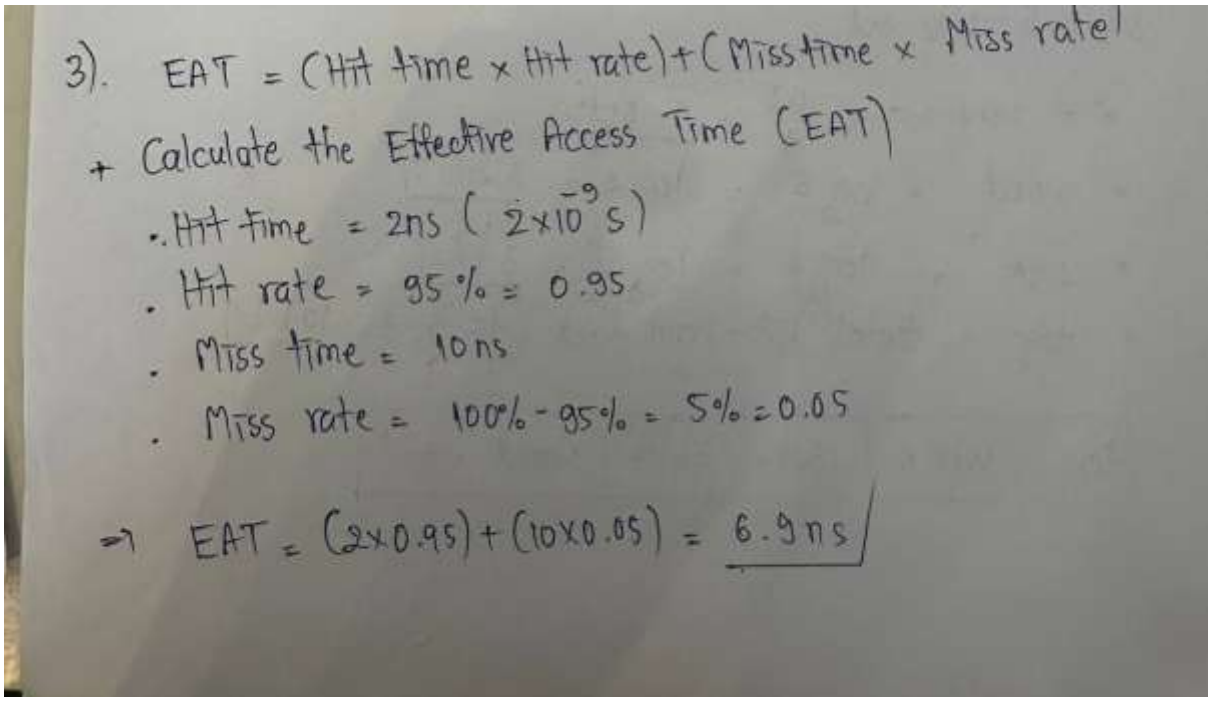
+ Problem 1:



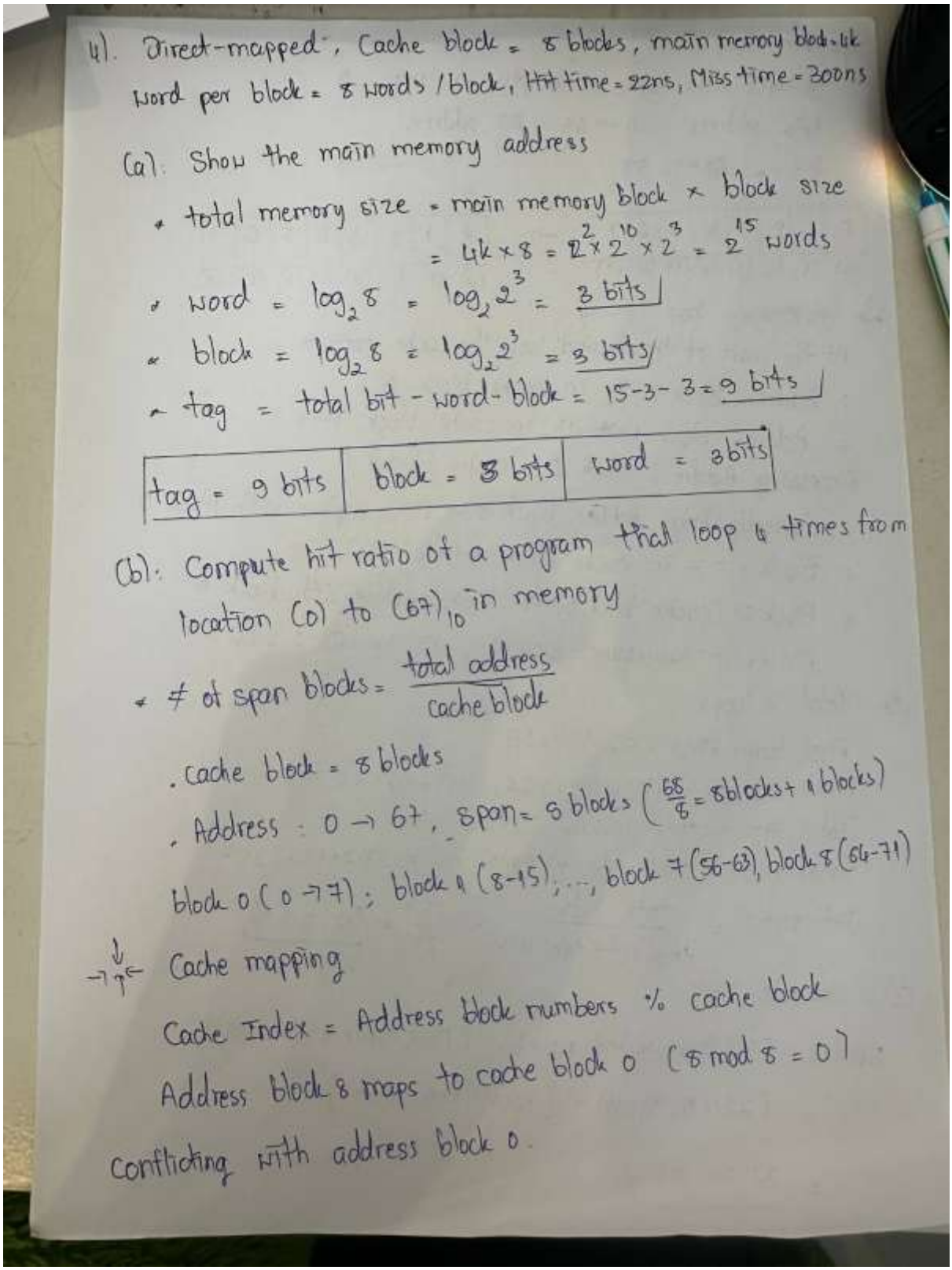
+ Problem 2:



+ Problem 3:



+ Problem 4:



4). Direct-mapped, Cache block = 8 blocks, main memory block = 4k
word per block = 8 words / block, Hit time = 22ns, Miss time = 300ns

(a). Show the main memory address

- total memory size = main memory block × block size
  $= 4k \times 8 = 2^2 \times 2^{10} \times 2^3 = 2^{15}$ words
- word = $\log_2 8 = \log_2 2^3 =$ 3 bits
- block = $\log_2 8 = \log_2 2^3 =$ 3 bits
- tag = total bit - word - block = 15 - 3 - 3 = 9 bits

tag = 9 bits	block = 3 bits	word = 3 bits

(b). Compute hit ratio of a program that loop 4 times from location (0) to $(67)_{10}$ in memory

- # of span blocks = $\frac{\text{total address}}{\text{cache block}}$

. cache block = 8 blocks

. Address : 0 → 67, span = 9 blocks ($\frac{68}{8}$ = 8 blocks + 4 blocks)

block 0 (0 → 7); block 1 (8-15), ..., block 7 (56-63), block 8 (64-71)

→ 7 ← Cache mapping

Cache Index = Address block numbers % cache block

Address block 8 maps to cache block 0 (8 mod 8 = 0)

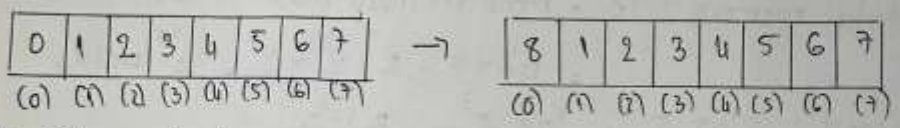
conflicting with address block 0.

- First loop

Initially, the cache is empty ⇒ miss = 9

total address : 0 → 67 = 68 address

hit = 68 - 9 = 59



- Subsequent loop

At the start of the second loop the cache contains:

+ Address block 8 is in cache block 0
+ Address block 1-7 is in cache block 1→7

Accessing Address block 0 (cache block 0)

+ Currently holds Address block 8 → Miss, replace with block 0
+ Block 1→7 in cache block 1→7 (Hits)
+ Block 8 (cache block 0) → Miss, replace with block 0

∴ Misses in subsequent loop = 2, Hits = 68 - 2 = 66

- Total 4 loops

First loop: Miss = 9, Hit = 59

Next 3 loops: Miss = 2 x 3 = 6, Hit = 66 x 3 = 198

Total hit + total miss = 68 x 4 = 272

total miss = 9 + 6 = 15 ⇒ total hit = 272 - 15 = 257

$$\text{Hit ratio} = \frac{\text{total hits}}{\text{total hit + total miss}} = \frac{257}{272} \approx 94.49\%$$

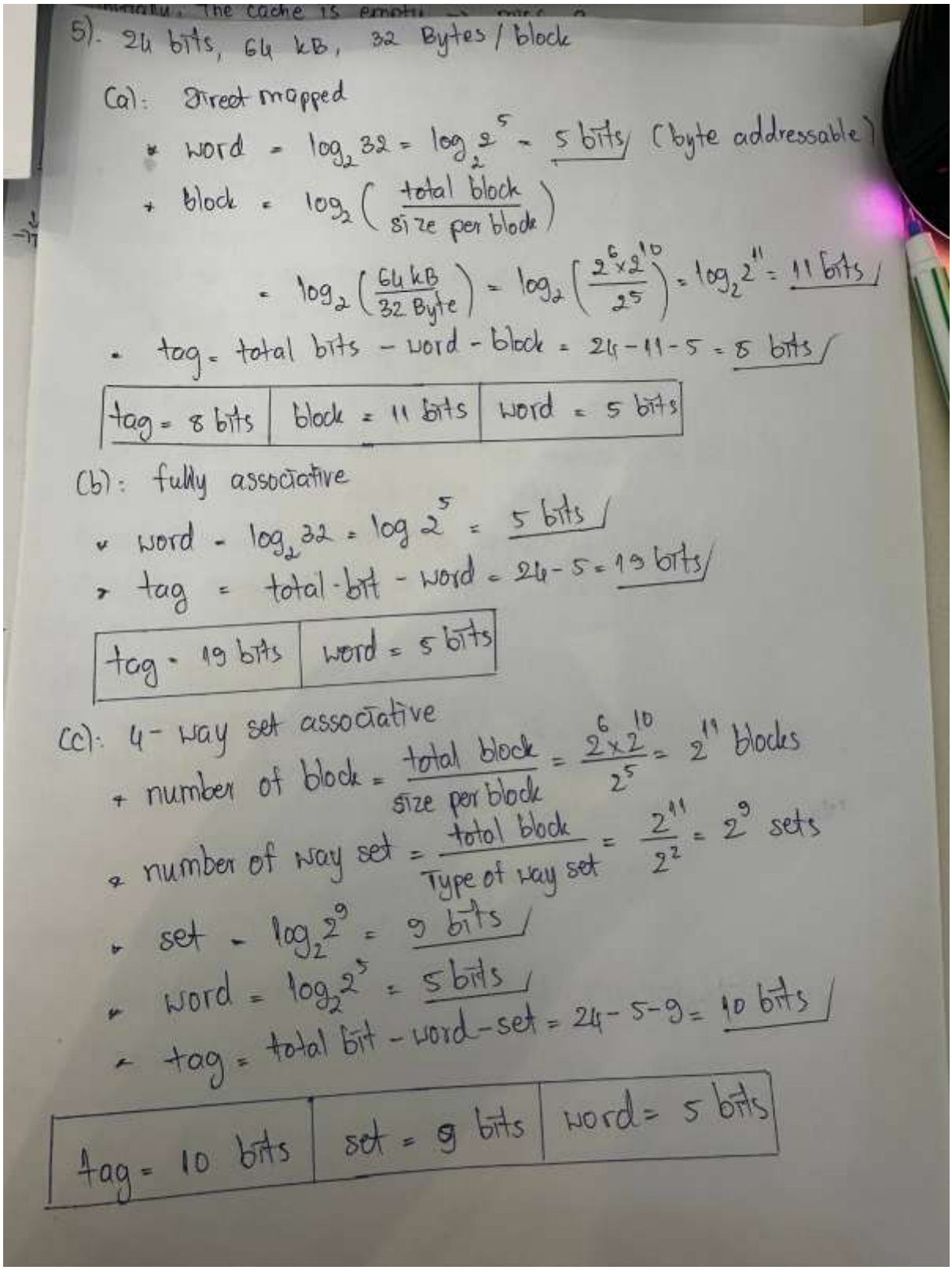
c): Compute EAT

$$EAT = (\text{Hit time} \times \text{Hit rate}) + (\text{Miss time} \times \text{Miss rate})$$

$$= (22 \times 0.9449) + [300 \times (1 - 0.9449)]$$

$$= 37.31 \text{ ns}$$

+ Problem 5:



5). 24 bits, 64 kB, 32 Bytes / block

(a): Direct mapped

- word = $\log_2 32 = \log_2 2^5 = 5$ bits (byte addressable)
- block = $\log_2\left(\frac{\text{total block}}{\text{size per block}}\right)$

$= \log_2\left(\frac{64\,kB}{32\,Byte}\right) = \log_2\left(\frac{2^6 \times 2^{10}}{2^5}\right) = \log_2 2^{11} = 11$ bits

- tag = total bits − word − block = 24 − 11 − 5 = 8 bits

tag = 8 bits	block = 11 bits	word = 5 bits

(b): fully associative

- word = $\log_2 32 = \log 2^5 = 5$ bits
- tag = total bit − word = 24 − 5 = 19 bits

tag = 19 bits	word = 5 bits

(c): 4-way set associative

- number of block = $\frac{\text{total block}}{\text{size per block}} = \frac{2^6 \times 2^{10}}{2^5} = 2^{11}$ blocks
- number of way set = $\frac{\text{total block}}{\text{Type of way set}} = \frac{2^{11}}{2^2} = 2^9$ sets
- set = $\log_2 2^9 = 9$ bits
- word = $\log_2 2^5 = 5$ bits
- tag = total bit − word − set = 24 − 5 − 9 = 10 bits

tag = 10 bits	set = 9 bits	word = 5 bits