Q1.

Given parameters:

- Virtual address space: 4GB
- Physical address space: 256MB
- Page size: 2KB
- Page table entry size: 4B
- Cache associativity: 8
- Block size: 2B

a) Maximum size of cache without aliasing?

In a VIPT cache, there is no aliasing when the index is contained within the page offset:

$$\begin{aligned}
&\text{index width} + \text{block offset width} \leqslant \text{page offset width} \\
\iff &\log_2(\#\text{sets}) + \log_2(\text{block size}) \leqslant \log_2(\text{page size}) \\
\iff &\log_2\left(\frac{\text{cache size}}{\text{block size} \cdot \text{associativity}}\right) + \log_2(\text{block size}) \leqslant \log_2(\text{page size}) \\
\iff &\log_2\left(\frac{\text{cache size}}{\text{associativity}}\right) \leqslant \log_2(\text{page size}) \\
\iff &\frac{\text{cache size}}{\text{associativity}} \leqslant \text{page size} \\
\iff &\text{cache size} \leqslant \text{associativity} \cdot \text{page size} \\
\iff &\text{cache size} \leqslant 8 \cdot 2\text{KB} = 16\text{KB}
\end{aligned}$$

The maximum cache size is 16KB.

b) Tag size?

Because the cache is VIPT, the tag size is given by

$$\begin{aligned}
\text{tag size} &= \text{physical address width} - \text{index width} - \text{block offset width} \\
&= \log_2(\text{physical address space}) - \log_2\left(\frac{\text{cache size}}{\text{block size} \cdot \text{associativity}}\right) - \log_2(\text{block size}) \\
&= \log_2(256 \cdot 2^{20}) - \log_2\left(\frac{16 \cdot 2^{10}}{2 \cdot 8}\right) - \log_2 2 = 28 - 10 - 1 = 17 \text{ bits}
\end{aligned}$$

c) Number of virtual pages?

The number of virtual pages is given by

$$\begin{aligned}
\#\text{virtual pages} &= \frac{\text{virtual address space}}{\text{page size}} \\
&= \frac{4\text{GB}}{2\text{KB}} = \frac{4 \cdot 2^{30}}{2 \cdot 2^{10}} = 2 \cdot 2^{20} = 2097152
\end{aligned}$$

d) Size of flat page table?

There is one entry in the page table per virtual page:

$$\begin{aligned}\text{page table size} &= \#\text{virtual pages} \cdot \text{entry size} \\ &= (2 \cdot 2^{20}) \cdot 4\text{B} = 8\text{B} \cdot 2^{20} = 8\text{MB}\end{aligned}$$

e) Size of 2-level page table with 8 bits for the first level?

The first-level table has 2^8 entries.

$$\text{first-level table size} = 2^8 \cdot \text{entry size} = 2^8 \cdot 4\text{B} = 1\text{KB}$$

Each second-level table has $2^{\text{VPN width}-8}$ entries.

$$\text{second-level table size} = 2^{\text{VPN width}-8} \cdot \text{entry size} = 2^{(32-11)-8} \cdot 4\text{B} = 32\text{KB}$$

If the page table is full, there are 2^8 second-level tables.

$$\begin{aligned}\text{page table size} &= \text{first-level table size} + 2^8 \cdot \text{second-level table size} \\ &= 1\text{KB} + 256 \cdot 32\text{KB} = 1\text{KB} + 8\text{MB} = 8193\text{KB}\end{aligned}$$

f) Size of 2-level page table with only one 16th of virtual space in use?

Only one 16th of the entries of the first-level table are valid, but the size of the first-level table is the same as in part *e*. Likewise, the second-level tables have the same size as in part *e*.

Since only one 16th of the entries of the first-level table are valid, there are 16 times fewer second-level tables than in part *e*:

$$\begin{aligned}\text{page table size} &= \text{first-level table size} + \frac{2^8}{16} \cdot \text{second-level table size} \\ &= 1\text{KB} + 16 \cdot 32\text{KB} = 513\text{KB}\end{aligned}$$

Q2.

Parameters:

- associativity: $a \in \{1, 2, 4\}$
- block size: $b \in \{1, 2, 4, 8, 16, 32\}$
- cache size: $s \in \{256, 512\}$
- replacement policy: $r \in \{\text{LRU}, \text{FIFO}\}$

We define $H_i \in \{true, false\}$, $H_i = true$ if the ith access is a hit, and $H_i = false$ otherwise.

A.

#	1	2	3	4	5	6
address	0	2	4	8	16	32

The hit rate is 0.16, so there is exactly 1 hit.

- Because the address sequence is strictly increasing, all misses are compulsory. The parameters a, s, r have no impact on the sequence of hits and misses.

- If $b \leqslant 2$, we have $\neg H_1 \wedge \neg H_2 \wedge \neg H_3 \wedge \neg H_4 \wedge \neg H_5 \wedge \neg H_6$.
  If $b = 4$, we have $\neg H_1 \wedge H_2 \wedge \neg H_3 \wedge \neg H_4 \wedge \neg H_5 \wedge \neg H_6$.
  If $b \geqslant 8$, we have $\neg H_1 \wedge H_2 \wedge H_3$.

The only constraint on the cache parameters is thus $b = 4$.

B.

#	1	2	3	4	5	6	7	8	9	10
address	0	512	1024	2048	1024	2048	512	1536	1024	1536

Note that $1536 = 3 \cdot 512$.

The hit rate is 0.3, so there are exactly 3 hits.

- Because the maximum value of b is 32, some misses are compulsory:

$$\neg H_1 \wedge \neg H_2 \wedge \neg H_3 \wedge \neg H_4 \wedge \neg H_8$$

- The maximum total size of a way in the cache is

$$\max \frac{s}{a} = 512\text{B}$$

  so all addresses that have the same remainder when divided by 512 are mapped to the same index. Here, all addresses are multiples of 512, so there are all mapped to the same index.

- If $a = 1$, all accesses are misses, so we know that $a \geqslant 2$.

- Access 5 is a hit if access 4 does not evict address 1024 from the cache. Access 4 evicts the LRU or FI (first inserted) address, which not 1024 if $a \geqslant 2$.

$$H_5 = (a \geqslant 2)$$

- Access 6 is a hit if access 5 does not evict address 2048 from the cache. Access 5 evicts the LRU or FI address, which not 2048 if $a \geqslant 2$.

$$H_6 = (a \geqslant 2)$$

- Access 7 is a hit if accesses 3, 4, 5, 6 do not evict address 512 from the cache. If $a = 2$, access 4 evicts 512. If $a = 4$, only 4 unique addresses are accessed before access 7, so there is no eviction. Thus,

$$H_7 = (a = 4)$$

- Access 9 is a hit if accesses 6, 7, 8 do not evict address 1024 from the cache. If $a = 2$, access 7 evicts 1024 because it is both the LRU and the FI address. If $a = 4$, access 8 evicts 0 because it is both the LRU and the FI address. So,

$$H_9 = (a = 4)$$

- There are 3 hits, so we know that $\neg(H_5 \wedge H_6 \wedge H_7 \wedge H_9)$ is true, and therefore $a = 2$.
- Access 10 is a hit if access 9 does not evict address 1536 from the cache. Since $a = 2$, access 9 evicts 512 because it is both the LRU and the FI address.

The only new constraint on the cache parameters is thus $a = 2$.

C.

#	1	2	3	4	5	6	7	8	9
address	0	16	32	64	128	256	128	64	0

The hit rate is 0.33, so there are exactly 3 hits.

- Because $b = 4$, we have

$$\neg H_1 \wedge \neg H_2 \wedge \neg H_3 \wedge \neg H_4 \wedge \neg H_5 \wedge \neg H_6$$

- The only possible combination of three hits is thus

$$H_7 \wedge H_8 \wedge H_9$$

- With $a = 2$, access 9 is hit only if there are two or fewer addresses in the sequence that are mapped to index 0.

  If $s = 256$, addresses 0, 128 and 256 are mapped to index 0.
  If $s = 512$, only addresses 0 and 256 are mapped to index 0.
  Thus, we know that $s = 512$.

- With $a = 2$ and $s = 512$, accesses 7 and 8 are also hits for a similar reason as access 9.

The new constraint imposed by sequence C is $s = 512$.

D.

#	1	2	3	4	5	6	7	8
address	0	512	0	1024	0	1536	2048	512

The hit rate is 0.25, so there are exactly 2 hits.

- Because $b = 4$, we know that
$$\neg H_1 \wedge \neg H_2 \wedge \neg H_4 \wedge \neg H_6 \wedge \neg H_7$$
- Because $a = 2$, $b = 4$, $s = 512$, there are 2 bits of offset and 6 bits of index. Since all eight addresses are multiple of $2^{6+2} = 256$, they are all mapped to index 0.
- Access 3 is a miss if access 2 evicts address 0 from the cache. Since $a = 2$, access 2 does not trigger any eviction. Thus,
$$H_3 = true$$
- Access 5 is a miss if access 4 evicts address 0 from the cache. Since $a = 2$, address 0 is the FI (first inserted) but is not the LRU address of the set:
$$H_5 = (r = \text{LRU})$$
- Access 8 is a hit if accesses (3, 4, 5, 6, 7) do not evict address 512 from the cache.

  With $a = 2$, access 4 evicts 0 if $r =$ FIFO or evicts 512 otherwise, and access 5 evicts 512 if $r =$ FIFO. Thus,
$$H_8 = false$$
- The only possible combination of hits and misses is thus
$$H_3 \wedge H_5 \wedge \neg H_8 = (r = \text{LRU})$$

The new constraint imposed by sequence D is thus $r =$ LRU.

Overall, the cache must therefore satisfy
$$\begin{cases} a = 2 \\ b = 4 \\ s = 512 \\ r = \text{LRU} \end{cases}$$

(Optional reading)
[Detailed derivation of the constraints for sequences C and D considered independently]

If you obtained a wrong final answer, the following may be useful for you to pinpoint where your mistake is.

C.

#	1	2	3	4	5	6	7	8	9
address	0	16	32	64	128	256	128	64	0

The hit rate is 0.33, so there are exactly 3 hits.

- We have
$$\neg H_1 \wedge \neg H_3 \wedge \neg H_4 \wedge \neg H_5 \wedge \neg H_6$$
and
$$H_2 = (b = 32)$$

- Access 7 is a hit if access 6 does not evict 128. If $a = 1$, 128 is evicted only if 128 and 256 are mapped to the same index. If $a \geqslant 2$, 128 is not evicted because it cannot be the LRU or the FI address. Thus,
$$\begin{aligned} H_7 &= (a \geqslant 2) \vee ((a = 1) \wedge (\text{way size} \geqslant 256)) \\ &= (a \geqslant 2) \vee \left((a = 1) \wedge \left(\frac{s}{a} \geqslant 256\right)\right) \\ &= (a \geqslant 2) \vee (s \geqslant 256) \end{aligned}$$

- Access 8 is a hit if accesses 5, 6, 7 do not evict address 64.

  If way size $\geqslant 128$B, addresses (128, 256) have different indices than 64, so 64 is not evicted.
  If way size $\leqslant 64$B, addresses (64, 128, 256) have the same index, and 64 is evicted if $a \leqslant 2$.

  Therefore,
$$\begin{aligned} H_8 &= (\text{way size} \geqslant 128) \vee (a = 4) \\ &= \left(\frac{s}{a} \geqslant 128\right) \vee (a = 4) \end{aligned}$$

- Access 9 is a hit if accesses (2, 3, ..., 8) do not evict address 0 from the cache. Until access 9, address 0 is always the LRU and FI of its set.

  If way size $= 512$B, all the addresses have different indices, so 0 is not evicted.
  If way size $= 256$B, address 256 has the same index as address 0, and 0 is evicted by access 6 if $a = 1$.
  If way size $= 128$B, addresses (128, 256) have the same index as address 0, and 0 is evicted only if $a \leqslant 2$.
  If way size $= 64$B, addresses (64, 128, 256) have the same index as address 0, and 0 is evicted only if $a \leqslant 2$.
  If way size $\leqslant 32$B, addresses (32, 64, 128, 256) have the same index as address 0, and 0 is necessarily evicted.

Thus,

$$\begin{aligned}H_9 &= \left(\frac{s}{a} = 512\right) \vee \left(\left(\frac{s}{a} = 256\right) \wedge (a \geqslant 2)\right) \vee \left(\left(\frac{s}{a} = 128\right) \wedge (a = 4)\right) \vee \left(\left(\frac{s}{a} = 64\right) \wedge (a = 4)\right) \\ &= \left(\frac{s}{a} = 512\right) \vee \left(\left(\frac{s}{a} = 256\right) \wedge (a \geqslant 2)\right) \vee ((s = 512) \wedge (a = 4)) \vee ((s = 256) \wedge (a = 4)) \\ &= ((a \leqslant 2) \wedge (s = 512)) \vee ((a = 4) \wedge (s \geqslant 256)) \\ &= (s = 512) \vee ((a = 4) \wedge (s \geqslant 256))\end{aligned}$$

- We note that $H_9 \Rightarrow H_7$ and that $H_9 \Rightarrow H_8$. There are thus only two possible combinations of three hits:

$$(H_2 \wedge H_7 \wedge H_8 \wedge \neg H_9) \vee (\neg H_2 \wedge H_7 \wedge H_8 \wedge H_9)$$

- We have

$$\begin{aligned}&H_2 \wedge H_7 \wedge H_8 \wedge \neg H_9 \\ = (b = 32) \wedge [(a \geqslant 2) \vee (s \geqslant 256)] \wedge &\left[\left(\frac{s}{a} \geqslant 128\right) \vee (a = 4)\right] \wedge \neg\left[(s = 512) \vee ((a = 4) \wedge (s \geqslant 256))\right] \\ &= (b = 32) \wedge [((a \leqslant 2) \wedge (s = 256)) \vee ((a = 4) \wedge (s \leqslant 128))] \qquad (1)\end{aligned}$$

- We have

$$\begin{aligned}&\neg H_2 \wedge H_7 \wedge H_8 \wedge H_9 = \neg H_2 \wedge H_9 \\ &= (b \leqslant 16) \wedge [(s = 512) \vee ((a = 4) \wedge (s = 256))] \qquad (2)\end{aligned}$$

The hit rate is thus 0.33 if and only if the cache satisfies either (1) or (2).

As expected, with $a = 2$ and $b = 4$, (1) $\vee$ (2) reduces to $(s = 512)$.

D.

#	1	2	3	4	5	6	7	8
address	0	512	0	1024	0	1536	2048	512

The hit rate is 0.25, so there are exactly 2 hits.

- Because $b \leqslant 32$, some misses are compulsory:

$$\neg H_1 \wedge \neg H_2 \wedge \neg H_4 \wedge \neg H_6 \wedge \neg H_7$$

- The maximum way size is

$$\max \frac{s}{a} = 512\text{B}$$

  Here, all addresses are multiples of 512, so there are all mapped to the same index.

- If $a = 1$, all accesses are misses, so we know that $a \geqslant 2$.
- Access 3 is a miss if access 2 evicts address 0 from the cache. Since $a \geqslant 2$, access 2 does not trigger any eviction. Thus,

$$H_3 = (a \geqslant 2)$$

- Access 5 is a miss if access 4 evicts address 0 from the cache. If $a = 2$, address 0 is the FI (first inserted) but is not the LRU address of the set. If $a = 4$, access 4 does not trigger any eviction. Thus,

$$H_5 = ((a = 2) \wedge (r = \text{LRU})) \vee (a = 4)$$

- Access 8 is a hit if accesses (3, 4, 5, 6, 7) do not evict address 512 from the cache.

  If $a = 2$, access 4 evicts 0 if $r =$ FIFO or evicts 512 otherwise, and access 5 evicts 512 if $r =$ FIFO.
  If $a = 4$, access 7 evicts 0 if $r =$ FIFO or evicts 512 otherwise.

  Thus,

  $$H_8 = (a = 4) \wedge (r = \text{FIFO})$$

- We can see that $H_8 \Rightarrow (H_3 \wedge H_5)$. The only possible combination of hits and misses is thus

  $$\begin{aligned} H_3 \wedge H_5 \wedge \neg H_8 &= (a \geqslant 2) \wedge [((a = 2) \wedge (r = \text{LRU})) \vee (a = 4)] \wedge \neg [(a = 4) \wedge (r = \text{FIFO})] \\ &= [(a = 2) \wedge (r = \text{LRU})] \vee [(a = 4) \wedge \neg (r = \text{FIFO})] \\ &= (a \geqslant 2) \wedge (r = \text{LRU}) \end{aligned}$$

The constraint on the cache parameters is thus $(a \geqslant 2) \wedge (r = \text{LRU})$.

As expected, with $a = 2$ and $b = 4$ and $s = 512$, the constraint reduces to $(r = \text{LRU})$.