Sample Questions

1.
   a. $2^{32} / (32 = 2^5) = 2^{27}$
      There are 2^{27} blocks.
   b. $2^{15} / 32 = 2^5 = 2^{10}$
      There are 2^{10} blocks.
   c. Offset: 5 bits because each block is 32 bytes.
      Block field: 10 bits because there are 2^{10} cache blocks.
      Tag field: 17 bits because 32 – 5 – 10 = 17.
   d. Offset: 5 bits because each block is 32 bytes.
      Tag field: 27 bits because 32 – 5 = 27.
   e. Offset: 5 bits because each block is 32 bytes.
      Cache set field: 8 bits because there are $2^{10} / 4 = 2^8$ cache blocks.
      Tag field: 19 bits because 32 – 5 – 8 = 19.

2. 64
   For set associate cache, the address is divided into three parts: tag, set, and offset.
   Since each block is 16 bytes, offset is 4 bits.
   2GB main memory has 31-bit address.
   Tag is 24 bits, which is given.
   Then, the set field is 3 bits because 31 – 24 – 4 = 3.
   Number of cache blocks = 8 KB / 16 = 2^13 / 2^4 = 2^9
   9 – 3 = 6
   Therefore, 2^6 = 64-way

3. 10 or 0xA
   32 blocks of cache: 5 bits for cache block
   Each block is 32 bytes: 5 bits for offset
   2370 = 10 01010 00010 (2)

4.
   a. $8 = 2^3$, 3 bits for page
      $64 = 2^6$, 6 bits for offset
      The virtual address has 9 bits = 3 + 6.
   b. $4 = 2^2$, 2 bits for frames
      $64 = 2^6$, 6 bits for offset
      The physical address has 8 bits = 2 + 6.
   c. 3 bits for 8 virtual pages
   d. 2 bits for 4 frames
   e. 6 bits for a page

5.

Address = 32 bits

Offset = 16 bytes = 2^4, **4 bits** (1 hex digit)

Cache = 4096 bytes = 4 x 1024 = 2^{12} bytes

Since each block is 16 bytes, there are 2^8 cache blocks because $2^{12} / 2^4 = 2^8$.

Cache blocks = **8 bits** (2 hex digits)

Tag = 20 bits because 32 – 4 – 8 = 20.

(0x0FF0 is omitted for readability, as it is the same for all addresses)

Address	Tag	Cache block	hit/miss	Justification
0xFABA	0xF	0xAB	miss	cache is empty
0xFAB1	0xF	0xAB	hit	same tag, memory block in cache
0xEABA	0xE	0xAB	miss	same cache block with different tag → different memory block
0xFAB3	0xF	0xAB	miss	same cache block with different tag → different memory block

6.

Address = 32 bits

Offset = 16 bytes = 2^4, **4 bits** (1 hexadecimal digit)

Cache = 4096 bytes = 4 x 1024 = 2^12 bytes

Since each block is 16 bytes, there are 2^8 cache blocks because 2^12 / 2^4 = 28.

Cache set = **6 bits** because 2^8 / $\mathbf{2^2}$ = 2^6 (divided with 4, as it is a 4-way associative cache)

Tag = 22 bits because 32 – 4 – 6 = 22.

(0x0FF0 is omitted for readability, as it is the same for all addresses)

0xFABA = 11 1110 10 1011 (1010) 0xEABA = 11 1010 10 1011 (1010)

Address	Tag	Cache set	hit/miss	Justification
0xFABA	0x3E	0x2B	miss	cache is empty
0xFAB1	0x3E	0x2B	hit	same tag and cache set
0xEABA	0x3A	0x2B	miss	different tag and same cache set
0xFAB3	0x3E	0x2B	hit	block is in the cache set

7.

Frame #	Valid Bit
1	1
2	1
4	1
16	1
9	1
-	0
-	0
-	0

a.

Since virtual address space is 16MB = 2^4 x 2^{20} = 2^{24}.
The virtual address is 24 bits.

b.

Since physical memory size is 2MB = 2^1 x 2^{20} = 2^{21}
The physical address is 21 bits.

c.

We divide 16MB / 1024 = 2^{24} / 2^{10} = 2^{14}.
There can be 2^{14} entries in a page table in maximum.

d.

The virtual address 0x5F4 = 0101 1111 0100_2.
Offset is 10 bits because each page is 1024 bytes.
01 is the virtual page number.
From the page table, virtual page 1 is mapped at page frame 2.
The physical address is 1001 1111 0100 = 0x9F4.

e.

The physical address 0x400 = 0100 0000 0000_2.
Offset is 10 bits.
01 is the page frame number.
From the page table, page frame 1 has the virtual page 0.
The virtual address is 0000 0000 0000 = 0x0.

8.

```
        # t0 = a, t1 = b
        addi t2, zero, 16     # t2 = 16
        bge t0, t2, else      # a >= 16, go to else
        andi t1, t0, 3        # 3 = 11(binary)
        j end                 # done
else:
        srai t0, t0, 2        # divided by 2^2
end:
```