Bit Boundary Explained

To understand fully what bit boundaries are and how they work, recall that each octet consists of 8-bits and that each bit has an assigned value or "bit value". The corresponding "bit values" for each bit in an octet is described in the table below.

Bit	8	7	6	5	4	3	2	1
Power of 2 (n)	7	6	5	4	3	2	1	0
Bit Value 2^n	128	64	32	16	8	4	2	1

Looking at the table above you will notice that each "bit value" is a power of 2. This is unique in that all subnet masks are either a power of 2 or the sum of powers of 2. For example, take the subnet mask 255.255.255.128, the number 128 is a power of 2. The number 2 raised to the 7^{th} power is 128. How about the subnet mask 255.255.240.0, the number 240 is a sum of the following powers of 2, (2^7) + (2^6) + (2^5) + (2^4) = 240. The number that is not 255 or zero, is either a power of 2 or the sum of multiple powers of 2. Keep this in mind as you read through the rest of the document. Let me back up and talk about classful network boundaries. You should already be familiar with the following Classful network boundaries:

Class A
10.0.0.0/8 mask 255.0.0.0

Class B
172.16.0.0/16 mask 255.255.0.0

Class C
192.168.1.0/24 mask 255.255.255.0

The "major" network boundaries occur at the end of each octet or 8-bits. Lets take a look at these three major network addresses in binary representation.

Class A

00001010.00000000.00000000.00000000

nnnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh

^network boundary

Class B

10101100.00010000.00000000.00000000

nnnnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh

^network boundary

Class A

11000000.10101000.00000001.00000000

nnnnnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh

^network boundaries

When we subnet these address, we "borrow" bits from the host (h) portion of the 32-bit address. In Classful subnetting the subnets had to all be of the same size. Example, subnet the Class C address 192.168.1.0 using a subnet mask of 255.255.255.240. Lets look at this in binary.

192.168.001.000 11000000.10101000.00000001.00000000

255.255.255.240 11111111.11111111.11111111.11110000

^ ^subnet boundary

^network boundary

Notice where the subnet boundary occurs, on the 4th MSB of the forth octet. Recall from the previous table, that the 4th MSB represents a “bit value” of 16. This means that each new subnet must be multiples of 16. You can also calculate the new subnet values using your powers of 2. Since we "borrowed" 4 bits from the host portion, we take 2 to the power of 4 or (2^4) = 16. The new subnets are 0, 16, 32, 48, 64, ...96,....128, ...192, ...240.

Bit Boundary Explained

The position of the last one in the octet is known as the "bit boundary", because all new subnets that are formed must have this bit position and all bits to the left in common. Example, lets take the first new subnet 192.168.1.0/28 and I am going to show all 16 addresses.

192.168.001.000 11000000.10101000.00000001.00000000
255.255.255.240 11111111.11111111.11111111.11110000

192.168.001.000 10101100.00010000.00000001.00000000
192.168.001.001 10101100.00010000.00000001.00000001
192.168.001.002 10101100.00010000.00000001.00000010
192.168.001.003 10101100.00010000.00000001.00000011
192.168.001.004 10101100.00010000.00000001.00000100
192.168.001.005 10101100.00010000.00000001.00000101
192.168.001.006 10101100.00010000.00000001.00000110
192.168.001.007 10101100.00010000.00000001.00000111
192.168.001.008 10101100.00010000.00000001.00001000
192.168.001.009 10101100.00010000.00000001.00001001
192.168.001.010 10101100.00010000.00000001.00001010
192.168.001.011 10101100.00010000.00000001.00001011
192.168.001.012 10101100.00010000.00000001.00001100
192.168.001.013 10101100.00010000.00000001.00001101
192.168.001.014 10101100.00010000.00000001.00001110
192.168.001.015 10101100.00010000.00000001.00001111

^bit boundary

Notice that all bits to the left of the “bit boundary” or the first 28-bits are all the same. IP address are not allowed to cross "bit boundaries". Now in Classful subnetting this was not a problem because all subnets had to be the same size based on the subnet mask or prefix length.

However, in classless subnetting or variable length subnet masking (VLSM), the "bit boundary" concept really becomes very important. Why? Because we can now have subnets of different sizes. For example, we could have subnet masks of /28, /29 and /30 prefix lengths all occurring within the same larger network, but these subnets follow one simple rule, they **MUST** still occur at the proper "bit boundary". Yes another example. Lets take these three new VLSMs and look at them in binary. First, a /28 prefix length is a mask of 255.255.255.240, a /29 prefix is a mask of 255.255.255.248 and the /30 prefix a mask of 255.255.255.252. But, can I subnet the original 192.168.1.0/24 network like this?

192.168.001.000 11000000.10101000.00000001.00000000
255.255.255.252 11111111.11111111.11111111.11111100
^bit boundary

192.168.001.000 10101100.00010000.00000001.00000000
192.168.001.001 10101100.00010000.00000001.00000001
192.168.001.002 10101100.00010000.00000001.00000010
192.168.001.003 10101100.00010000.00000001.00000011
^bit boundary

192.168.001.004 10101100.00010000.00000001.00000100
255.255.255.248 11111111.11111111.11111111.11111000
^bit boundary

192.168.001.004 10101100.00010000.00000001.00000100
192.168.001.005 10101100.00010000.00000001.00000101
192.168.001.006 10101100.00010000.00000001.00000110
192.168.001.007 10101100.00010000.00000001.00000111
192.168.001.008 10101100.00010000.00000001.00001000

192.168.001.009 10101100.00010000.00000001.00001001
192.168.001.010 10101100.00010000.00000001.00001010
192.168.001.011 10101100.00010000.00000001.00001011
192.168.001.012 10101100.00010000.00000001.00001100
^bit boundary

No! Why? Because the default "bit boundary" for a /29 subnet **MUST** occur at multiples of 8, such as 0, 8, 16, ...24,...etc. Notice in the above example the /29 started at 4 and 4 is not a multiple of 8. In addition, the addresses "crossed" over the "bit boundary" as we counted up from 4 to 12. This behavior is not allowed. Therefore, 4 is not the proper "bit boundary" in which to start the /29 subnet. Which brings up a very important issue regarding VLSM, in that you should start with the largest subnet first, then proceed in descending order to the smallest subnet. Like so,

192.168.001.000 11000000.10101000.00000001.00000000
255.255.255.240 11111111.11111111.11111111.11110000

192.168.001.000 10101100.00010000.00000001.00000000 <--- network address
192.168.001.001 10101100.00010000.00000001.00000001
192.168.001.002 10101100.00010000.00000001.00000010
192.168.001.003 10101100.00010000.00000001.00000011
192.168.001.004 10101100.00010000.00000001.00000100
192.168.001.005 10101100.00010000.00000001.00000101
192.168.001.006 10101100.00010000.00000001.00000110
192.168.001.007 10101100.00010000.00000001.00000111
192.168.001.008 10101100.00010000.00000001.00001000
192.168.001.009 10101100.00010000.00000001.00001001
192.168.001.010 10101100.00010000.00000001.00001010

192.168.001.011 10101100.00010000.00000001.00001011
192.168.001.012 10101100.00010000.00000001.00001100
192.168.001.013 10101100.00010000.00000001.00001101
192.168.001.014 10101100.00010000.00000001.00001110
192.168.001.015 10101100.00010000.00000001.00001111 <--- broadcast address
^bit boundary

192.168.001.016 10101100.00010000.00000001.00010000
255.255.255.248 11111111.11111111.11111111.11111000

192.168.001.016 10101100.00010000.00000001.00010000 <--- network address
192.168.001.017 10101100.00010000.00000001.00010001
192.168.001.018 10101100.00010000.00000001.00010010
192.168.001.019 10101100.00010000.00000001.00010011
192.168.001.020 10101100.00010000.00000001.00010100
192.168.001.021 10101100.00010000.00000001.00010101
192.168.001.022 10101100.00010000.00000001.00010110
192.168.001.023 10101100.00010000.00000001.00010111 <--- broadcast address
^bit boundary

192.168.001.024 10101100.00010000.00000001.00010000
255.255.255.252 11111111.11111111.11111111.11111100

192.168.001.024 10101100.00010000.00000001.00010000 <--- network address
192.168.001.025 10101100.00010000.00000001.00010001
192.168.001.026 10101100.00010000.00000001.00010010
192.168.001.027 10101100.00010000.00000001.00010011 <--- broadcast address
^bit boundary

Now, do I have to start the subnets with largest first and proceed in descending order until I reach the smallest subnet? No. I can leave "gaps" or I can put all my /30 subnets at he top of the range or even at the lower end of the range. The only thing I **MUST** do, is remember to start my subnets on the proper "bit boundary". That means for a /28 subnet it must start on any multiple of 16, for a /29 on any multiple of 8, and for a /30 on any multiple of 4. You would do the same for a /27 on any multiple of 32, a /26 on any multiple of 64 and a /25 on any multiple of 128.

In conclusion, the “bit boundary” is an invisible line that occurs just after the last bit in the prefix length or where the last “1” in the subnet mask ends. It separates the subnet bits from the host bits. As we count out the IP addresses, for our newly created subnet, we can not “cross” over this imaginary “bit boundary”. When using VLSM, remember the rule and make sure that the subnets still start on the default “bit boundary” for their specific prefix length or subnet mask. Sorry if this was long winded, but sometimes examples are the best way to help in understanding certain concepts, especially mathematical ones like binary “bit boundaries”. Happy subnetting!

