CCNA 1 v7.0 Curriculum: Module 5 – Number Systems

itexamanswers.net/ccna-1-v7-0-curriculum-module-5-number-systems.html

April 1, 2020

5.0. Introduction

5.0.1. Why should I take this module?

Welcome to Number Systems!

Guess what? This is a 32-bit IPv4 address of a computer in a network: 11000000.10101000.00001010.00001010. It is shown in binary. This is the IPv4 address for the same computer in dotted decimal: 192.168.10.10. Which one would you rather work with? IPv6 addresses are 128 bits! To make these addresses more manageable, IPv6 uses a hexadecimal system of 0-9 and the letters A-F.

As a network administrator you must know how to convert binary addresses into dotted decimal and dotted decimal addresses into binary. You will also need to know how to convert dotted decimal into hexadecimal and vice versa. (Hint: You still need your binary conversion skills to make this work.)

Surprisingly, it is not that hard when you learn a few tricks. This module contains an activity called the Binary Game which will really help you get started. So, why wait?

5.0.2. What will I learn to do in this module?

Module Title: Number Systems

Module Objective: Calculate numbers between decimal, binary, and hexadecimal systems.

Topic Title	Topic Objective
Binary Number System	Calculate numbers between decimal and binary systems.
Hexadecimal Number System	Calculate numbers between decimal and hexadecimal systems.

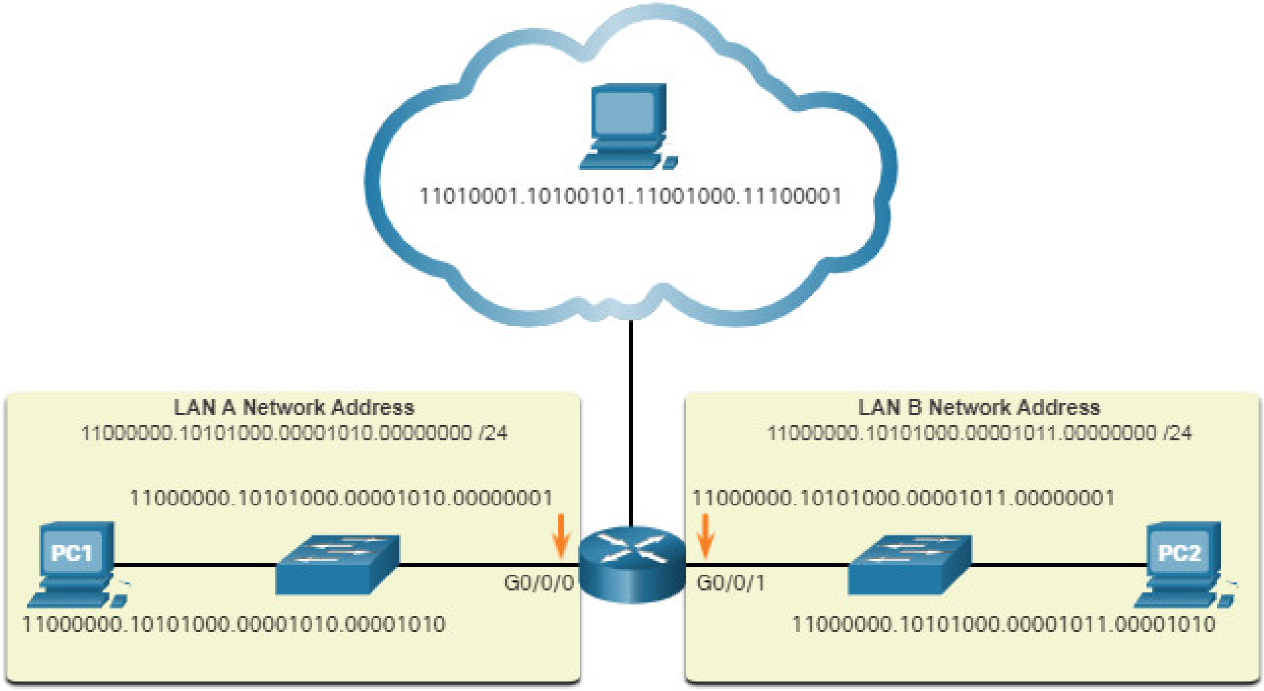
5.1. Binary Number System

5.1.1. Binary and IPv4 Addresses

IPv4 addresses begin as binary, a series of only 1s and 0s. These are difficult to manage, so network administrators must convert them to decimal. This topic shows you a few ways to do this.

Binary is a numbering system that consists of the digits 0 and 1 called bits. In contrast, the decimal numbering system consists of 10 digits consisting of the digits 0 – 9.

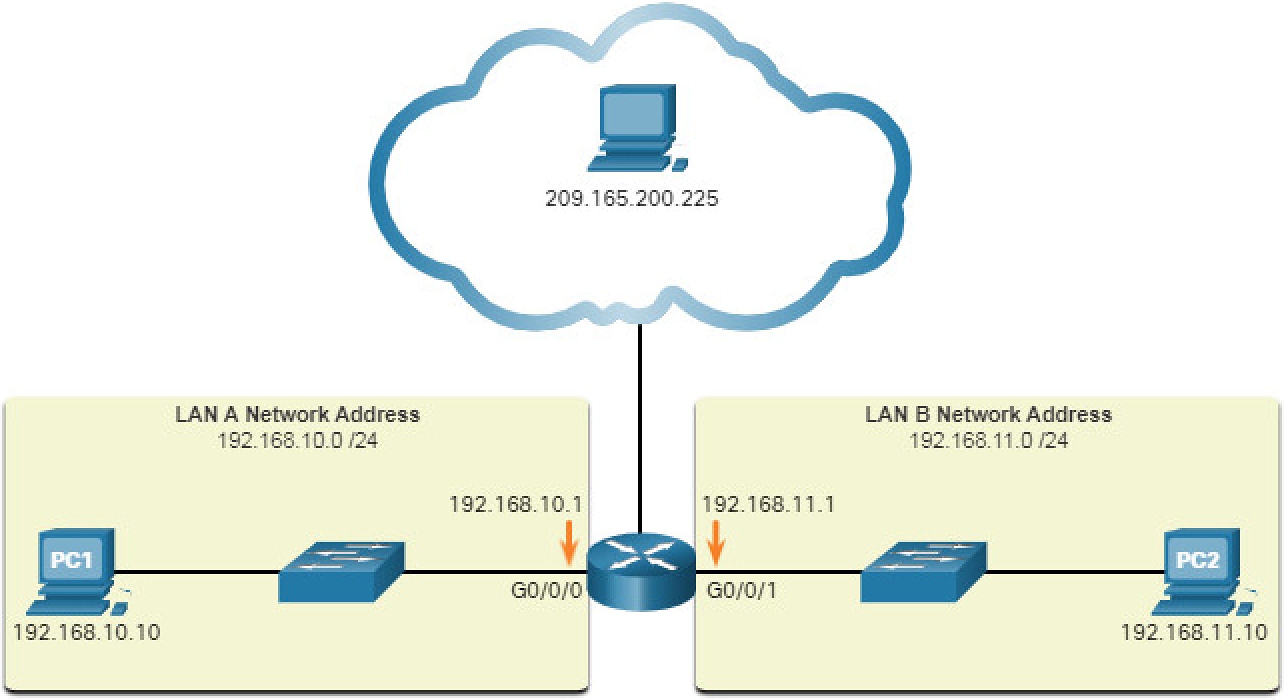
Binary is important for us to understand because hosts, servers, and network devices use binary addressing. Specifically, they use binary IPv4 addresses, as shown in the figure, to identify each other.



Each address consists of a string of 32 bits, divided into four sections called octets. Each octet contains 8 bits (or 1 byte) separated with a dot. For example, PC1 in the figure is assigned IPv4 address 11000000.10101000.00001010.00001010. Its default gateway address would be that of R1 Gigabit Ethernet interface 11000000.10101000.00001010.00000001.

Binary works well with hosts and network devices. However, it is very challenging for humans to work with.

For ease of use by people, IPv4 addresses are commonly expressed in dotted decimal notation. PC1 is assigned the IPv4 address 192.168.10.10, and its default gateway address is 192.168.10.1, as shown in the figure.



For a solid understanding of network addressing, it is necessary to know binary addressing and gain practical skills converting between binary and dotted decimal IPv4 addresses. This section will cover how to convert between base two (binary) and base 10 (decimal) numbering systems.

5.1.2. Video – Converting Between Binary and Decimal Numbering Systems

5.1.3. Binary Positional Notation

Learning to convert binary to decimal requires an understanding of positional notation. Positional notation means that a digit represents different values depending on the "position" the digit occupies in the sequence of numbers. You already know the most common numbering system, the decimal (base 10) notation system.

The decimal positional notation system operates as described in the table.

Radix	**10**	**10**	**10**	**10**
Position in Number	3	2	1	0
Calculate	(10^3)	(10^2)	(10^1)	(10^0)
Position value	1000	100	10	1

The following bullets describe each row of the table.

- Row 1, Radix is the number base. Decimal notation is based on 10, therefore the radix is 10.
- Row 2, Position in number considers the position of the decimal number starting with, from right to left, 0 (1st position), 1 (2nd position), 2 (3rd position), 3 (4th position). These numbers also represent the exponential value use to calculate the positional value in the 4th row.
- Row 3 calculates the positional value by taking the radix and raising it by the exponential value of its position in row 2.
  Note: n0 is = 1.
- Row 4 positional value represents units of thousands, hundreds, tens, and ones.

To use the positional system, match a given number to its positional value. The example in the table illustrates how positional notation is used with the decimal number 1234.

	Thousands	**Hundreds**	**Tens**	**Ones**
Positional Value	1000	100	10	1
Decimal Number (1234)	1	2	3	4
Calculate	1 x 1000	2 x 100	3 x 10	4 x 1
Add them up…	1000	+ 200	+ 30	+ 4
Result	**1,234**			

In contrast, the binary positional notation operates as described in the table.

Radix	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Position in Number	7	6	5	4	3	2	1	0
Calculate	(2^7)	(2^6)	(2^5)	(2^4)	(2^3)	(2^2)	(2^1)	(2^0)
Position value	128	64	32	16	8	4	2	1

The following bullets describe each row of the table.

- Row 1, Radix is the number base. Binary notation is based on 2, therefore the radix is 2.
- Row 2, Position in number considers the position of the binary number starting with, from right to left, 0 (1st position), 1 (2nd position), 2 (3rd position), 3 (4th position). These numbers also represent the exponential value use to calculate the positional value in the 4th row.
- Row 3 calculates the positional value by taking the radix and raising it by the exponential value of its position in row 2.
  Note: n0 is = 1.

- Row 4 positional value represents units of ones, twos, fours, eights, etc.

The example in the table illustrates how a binary number 11000000 corresponds to the number 192. If the binary number had been 10101000, then the corresponding decimal number would be 168.

Positional Value	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Binary Number (11000000)	1	1	0	0	0	0	0	0
Calculate	1 x 128	1 x 64	0 x 32	0 x 16	0 x 8	0 x 4	0 x 2	0 x 1
Add Them Up..	128	+ 64	+ 0	+ 0	+ 0	+ 0	+ 0	+ 0
Result	**192**							

5.1.5. Convert Binary to Decimal

To convert a binary IPv4 address to its dotted decimal equivalent, divide the IPv4 address into four 8-bit octets. Next apply the binary positional value to the first octet binary number and calculate accordingly.

For example, consider that 11000000.10101000.00001011.00001010 is the binary IPv4 address of a host. To convert the binary address to decimal, start with the first octet, as shown in the table. Enter the 8-bit binary number under the positional value of row 1 and then calculate to produce the decimal number 192. This number goes into the first octet of the dotted decimal notation.

Positional Value	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Binary Number (11000000)	1	1	0	0	0	0	0	0
Calculate	128	64	32	16	8	4	2	1
Add Them Up…	128	+ 64	+ 0	+ 0	+ 0	+ 0	+ 0	+ 0
Result	**192**							

Next convert the second octet of 10101000 as shown in the table. The resulting decimal value is 168, and it goes into the second octet.

Positional Value	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Binary Number (10101000)	1	0	1	0	1	0	0	0

Positional Value	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Calculate	128	64	32	16	8	4	2	1
Add Them Up…	128	+ 0	+ 32	+ 0	+ 8	+ 0	+ 0	+ 0
Result	**168**							

Convert the third octet of 00001011 as shown in the table.

Positional Value	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Binary Number (00001011)	0	0	0	0	1	0	1	1
Calculate	128	64	32	16	8	4	2	1
Add Them Up…	0	+ 0	+ 0	+ 0	+ 8	+ 0	+ 2	+ 1
Result	**11**							

Convert the fourth octet of 00001010 as shown in the table. This completes the IP address and produces **192.168.11.10.**

Positional Value	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Binary Number (00001010)	0	0	0	0	1	0	1	0
Calculate	128	64	32	16	8	4	2	1
Add Them Up…	0	+ 0	+ 0	+ 0	+ 8	+ 0	+ 2	+ 0
Result	**10**							

5.1.6. Activity – Binary to Decimal Conversions

5.1.7. Decimal to Binary Conversion

It is also necessary to understand how to convert a dotted decimal IPv4 address to binary. A useful tool is the binary positional value table.

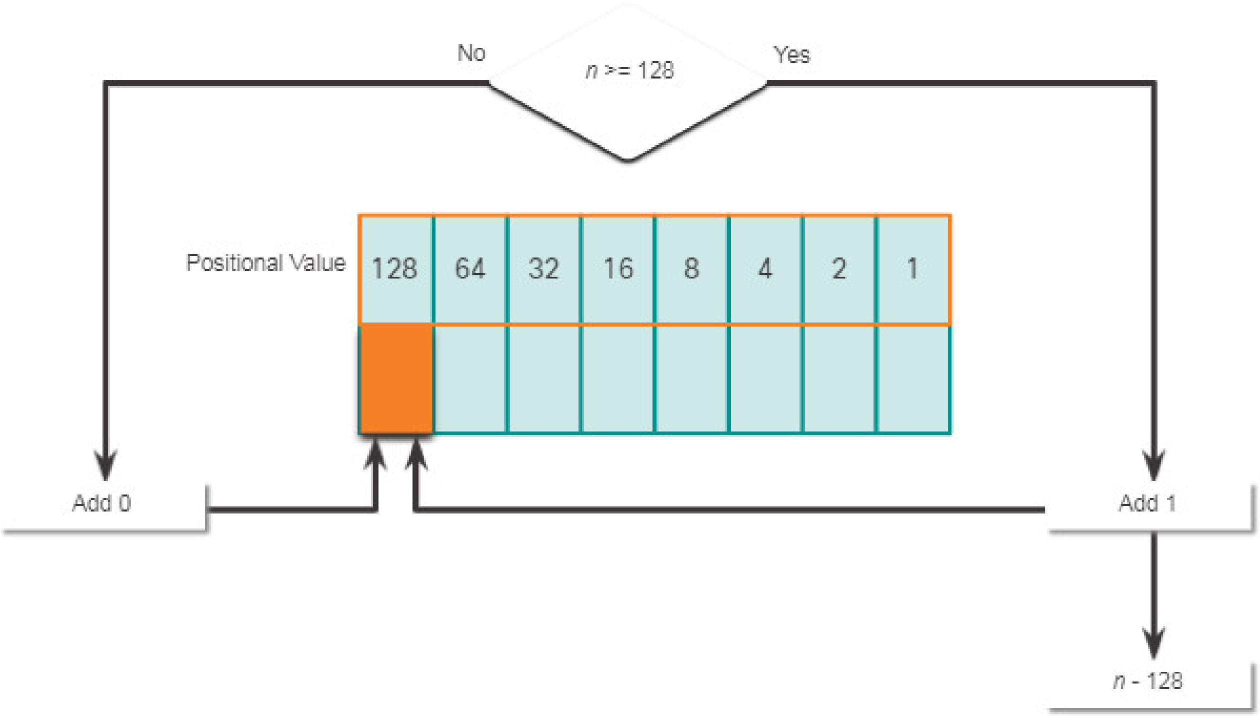
Click each position starting at 128 and work your way from left to right to the 1 position.

- 128
- 64
- 32
- 16
- 8

- 4
- 2
- 1

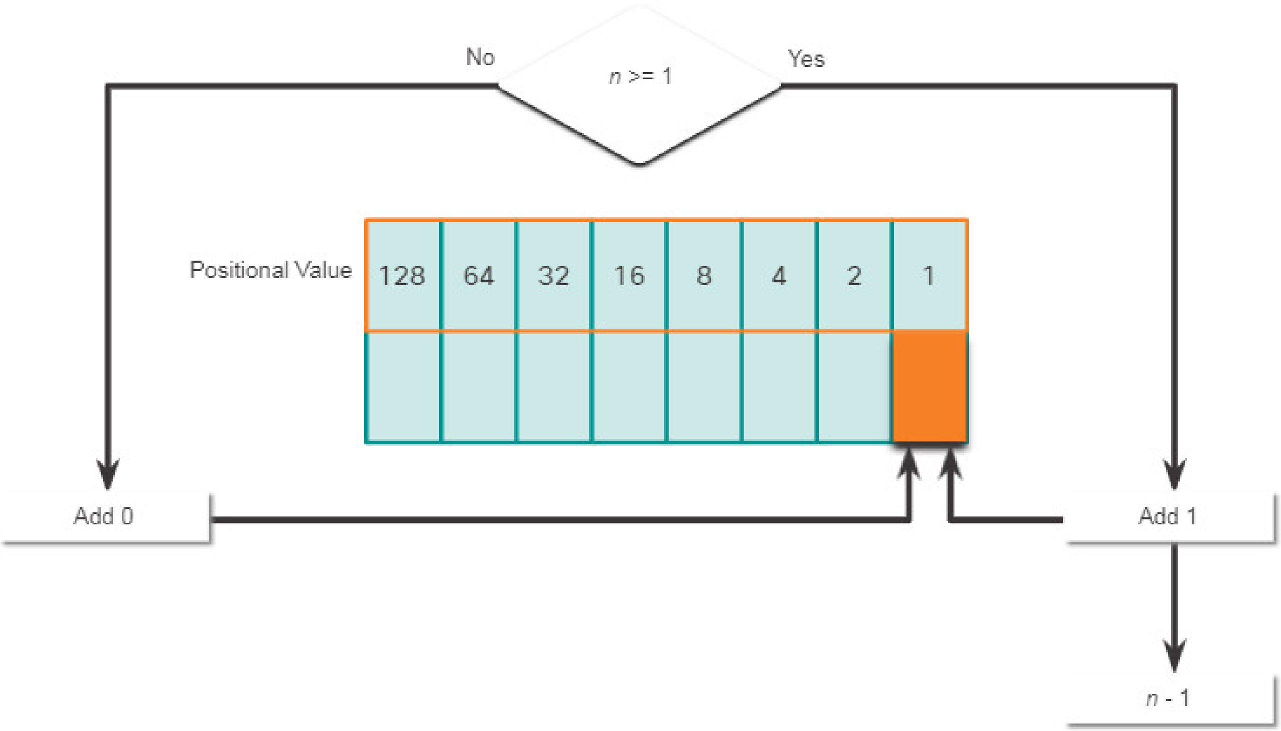
Is the decimal number of the octet (n) equal to or greater than the most-significant bit **(128)?**

If no, then enter binary **0** in the **128** positional value.
If yes, then add a binary **1** in the **128** positional value and subtract **128** from the decimal number.



Is the decimal number of the octet (n) equal to or greater than the last most-significant bit **(1)?**

If no, then enter binary **0** in the **1** positional value.
If yes, then add a binary **1** in the **1** positional value and subtract **1** from the last decimal number.



5.1.8. Decimal to Binary Conversion Example

To help understand the process, consider the IP address 192.168.11.10.

The first octet number 192 is converted to binary using the previously explained positional notation process.

It is possible to bypass the process of subtraction with easier or smaller decimal numbers. For instance, notice that it is fairly easy to calculate the third octet converted to a binary number without actually going through the subtraction process (8 + 2 = 10). The binary value of the third octet is 00001010.

The fourth octet is 11 (8 + 2 + 1). The binary value of the fourth octet is 00001011.

Converting between binary and decimal may seem challenging at first, but with practice it should become easier over time.

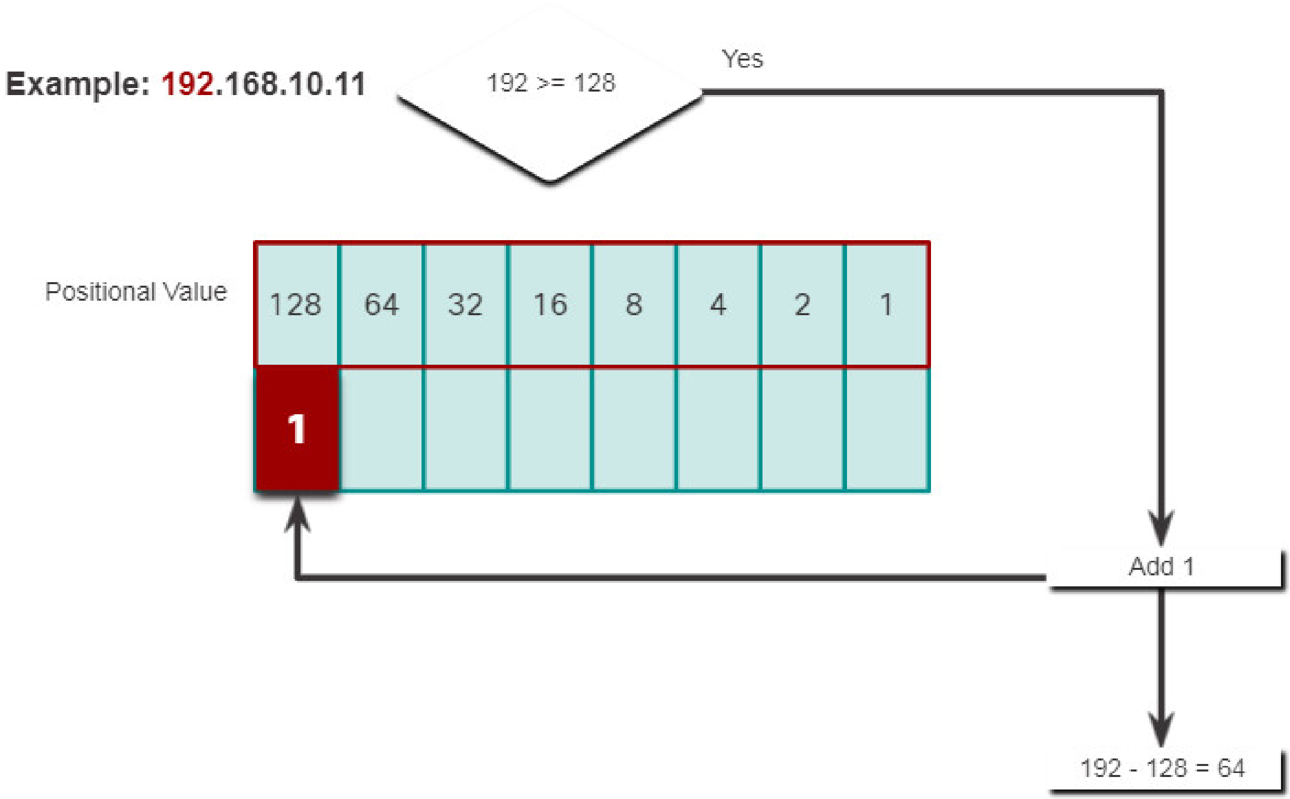
Click each step to see the conversion of the IP address of 192.168.10.11 into binary.

- Step 1
- Step 2
- Step 3
- Step 4
- Step 5

- Step 6
- Step 7
- Step 8
- Step 9
- Step 10
- Step 11

Is the first octet number 192 equal to or greater than the high-order bit 128?

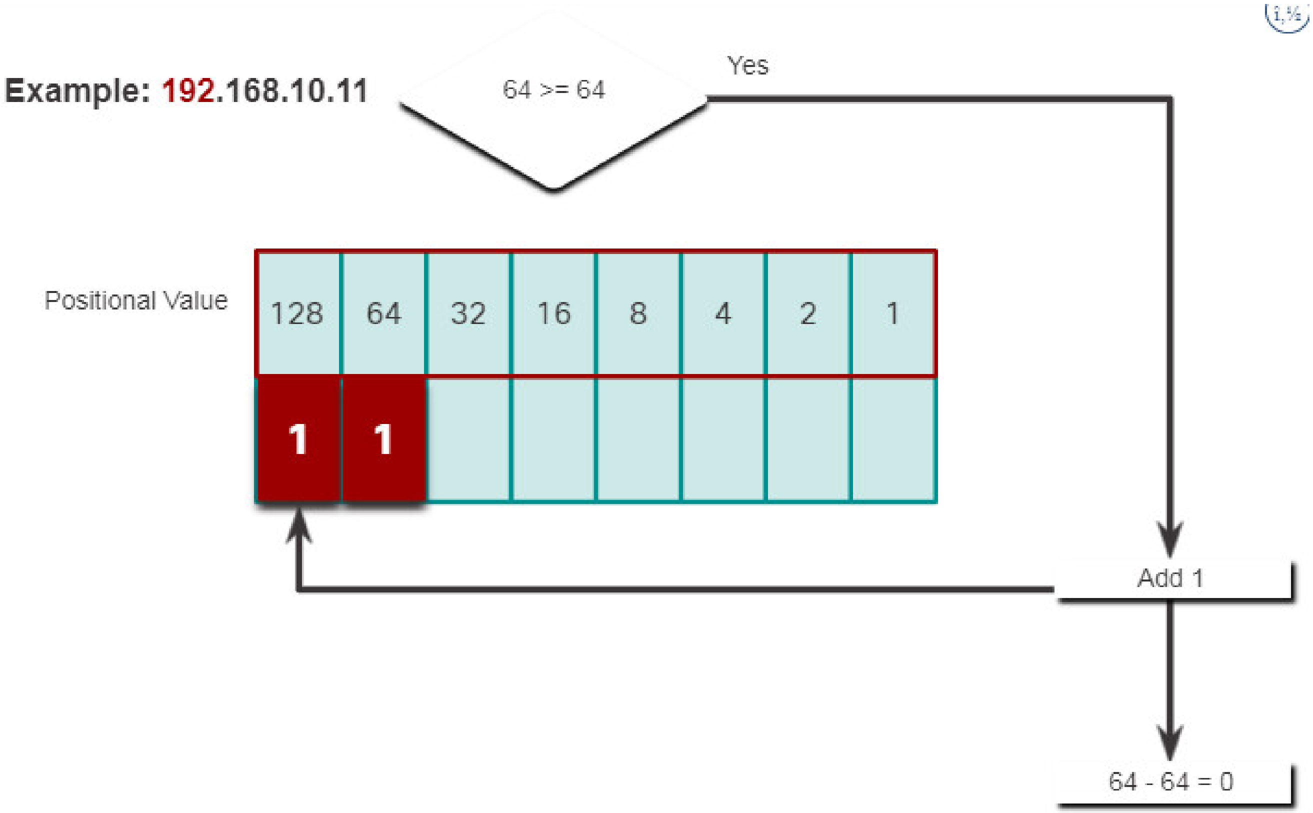
- Yes it is, therefore add a 1 to the high-order positional value to a represent 128.
- Subtract 128 from 192 to produce a remainder of 64.



Positional Value	128	64	32	16	8	4	2	1
	1							

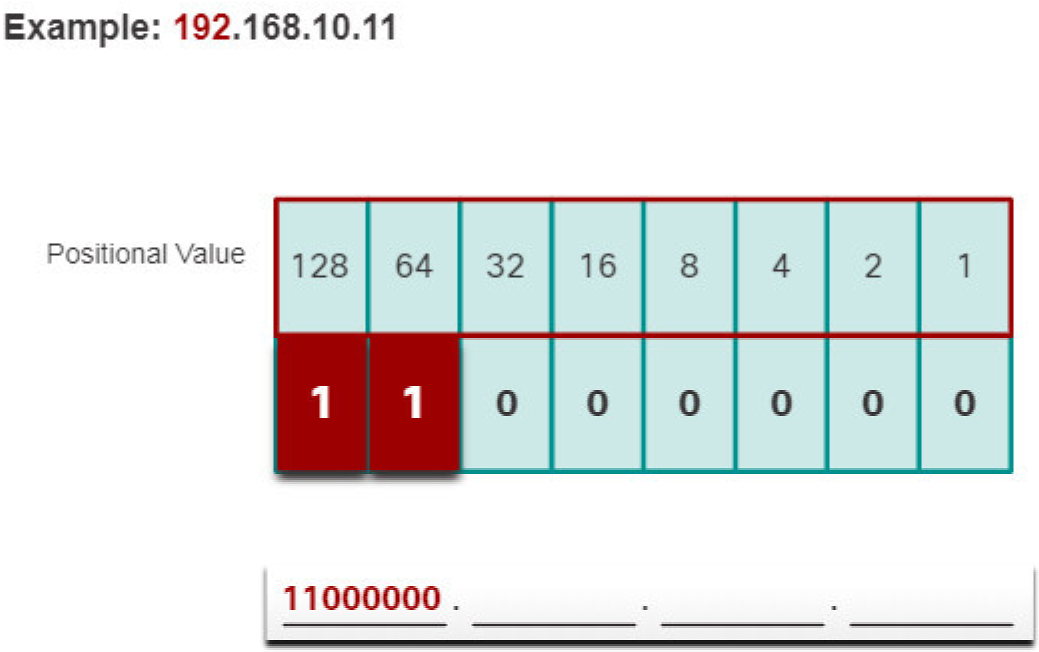
Is the remainder 64 equal to or greater than the next high-order bit 64?

It is equal, therefore add a 1 to next high-order positional value.



Since there is no remainder, enter binary **0** in the remaining positional values.

The binary value of the first octet is **11000000.**

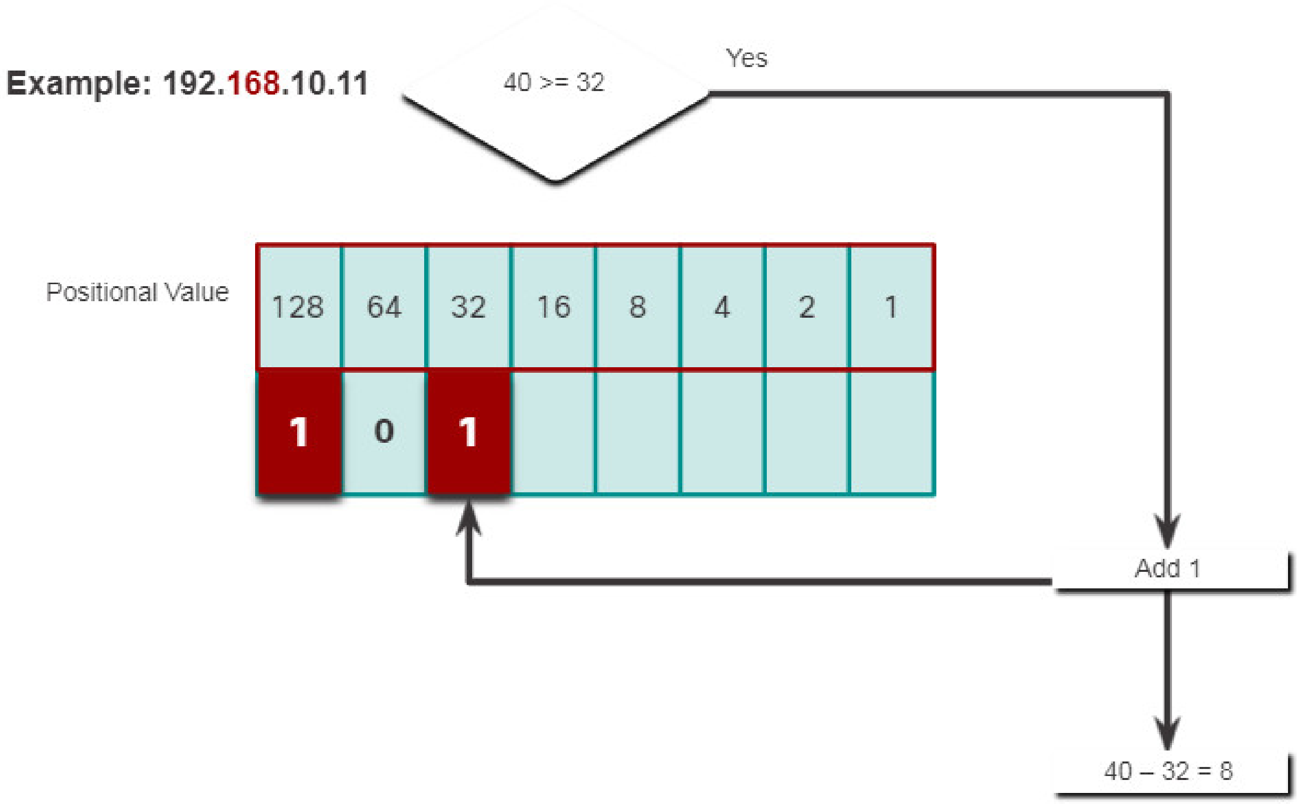


Positional Value	128	64	32	16	8	4	2	1
	1	1	0	0	0	0	0	0

11000000 . ________ . ________ . ________

Is the remainder **40** equal to or greater than the next high-order bit **32?**

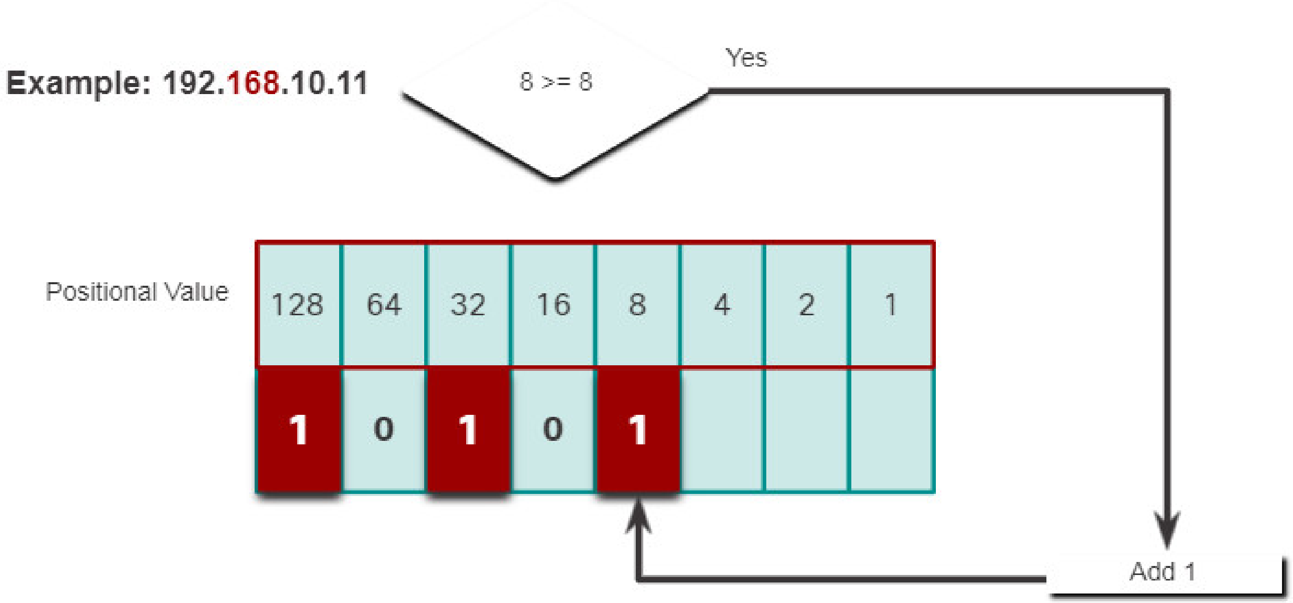
- Yes it is, therefore add a **1** to the high-order positional value to represent **32.**
- Subtract **32** from **40** to produce a remainder of **8**.



Positional Value	128	64	32	16	8	4	2	1
	1	0	1					

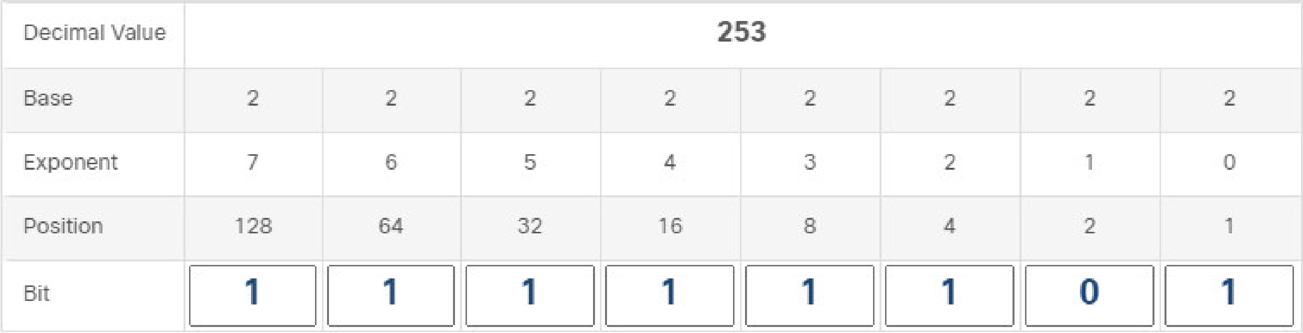
Is the remainder **8** equal to or greater than the next high-order bit **8**?

It is equal, therefore add a **1** to next high-order positional value.



Positional Value	128	64	32	16	8	4	2	1
	1	0	1	0	1			

5.1.9. Activity – Decimal to Binary Conversions



Decimal Value	253							
Base	2	2	2	2	2	2	2	2
Exponent	7	6	5	4	3	2	1	0
Position	128	64	32	16	8	4	2	1
Bit	1	1	1	1	1	1	0	1

5.1.10. Activity – Binary Game

This is a fun way to learn binary numbers for networking.

Game Link: https://learningnetwork.cisco.com/docs/DOC-1803

You will need to log in to cisco.com to use this link. It will be necessary to create an account if you do not already have one.

Mobile Download: https://learningnetwork.cisco.com/docs/DOC-11119

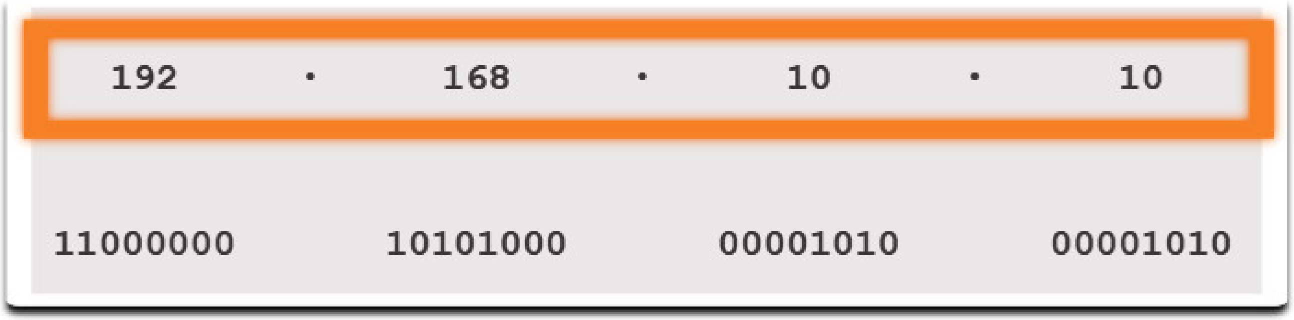
5.1.11. IPv4 Addresses

As mentioned in the beginning of this topic, routers and computers only understand binary, while humans work in decimal. It is important for you to gain a thorough understanding of these two numbering systems and how they are used in networking.

Click each button to contrast the dotted decimal address and the 32-bit address.

- Dotted Decimal Address
- Octets
- 32-bit Address

192.168.10.10 is an IP address that is assigned to a computer.



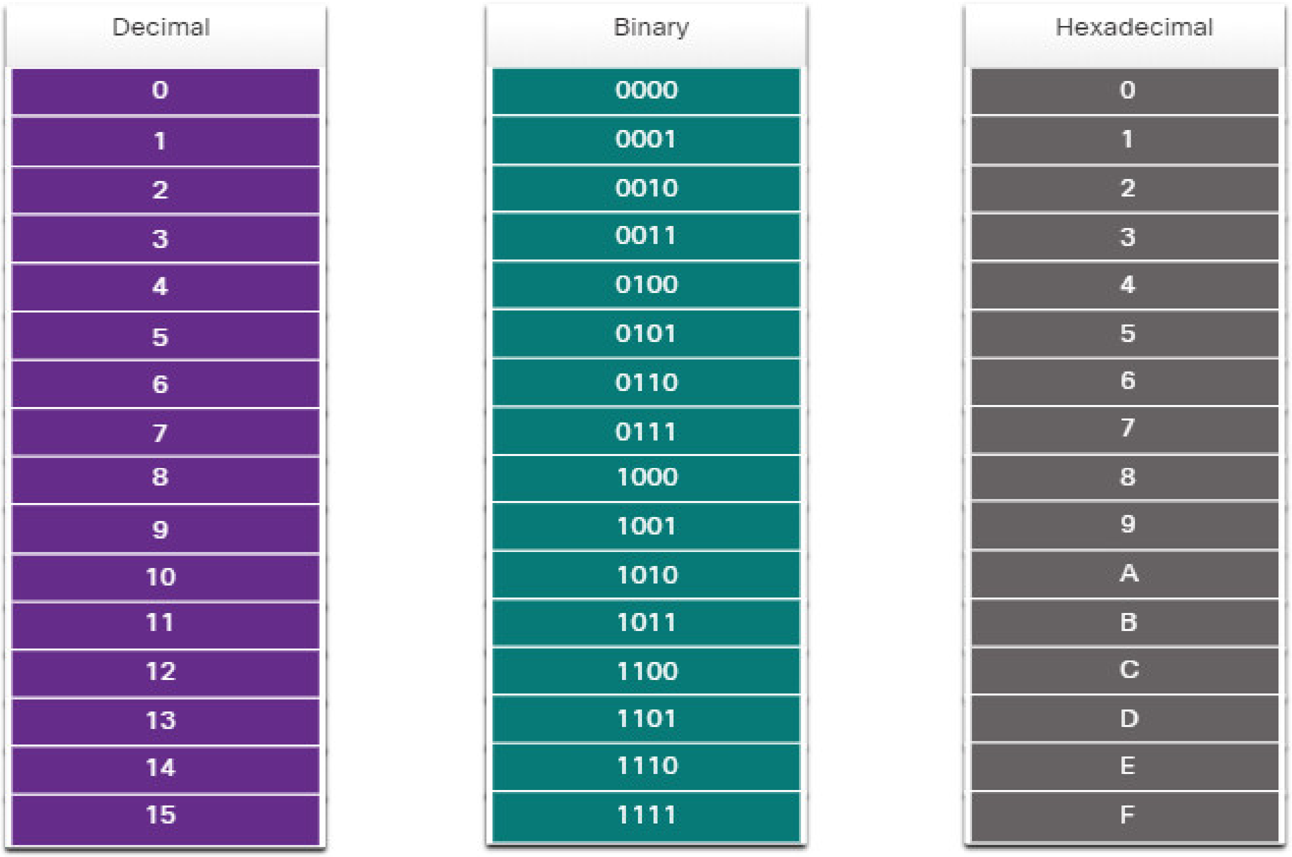
192	.	168	.	10	.	10
11000000		10101000		00001010		00001010

5.2. Hexadecimal Number System

5.2.1. Hexadecimal and IPv6 Addresses

Now you know how to convert binary to decimal and decimal to binary. You need that skill to understand IPv4 addressing in your network. But you are just as likely to be using IPv6 addresses in your network. To understand IPv6 addresses, you must be able to convert hexadecimal to decimal and vice versa.

Just as decimal is a base ten number system, hexadecimal is a base sixteen system. The base sixteen number system uses the digits 0 to 9 and the letters A to F. The figure shows the equivalent decimal and hexadecimal values for binary 0000 to 1111.



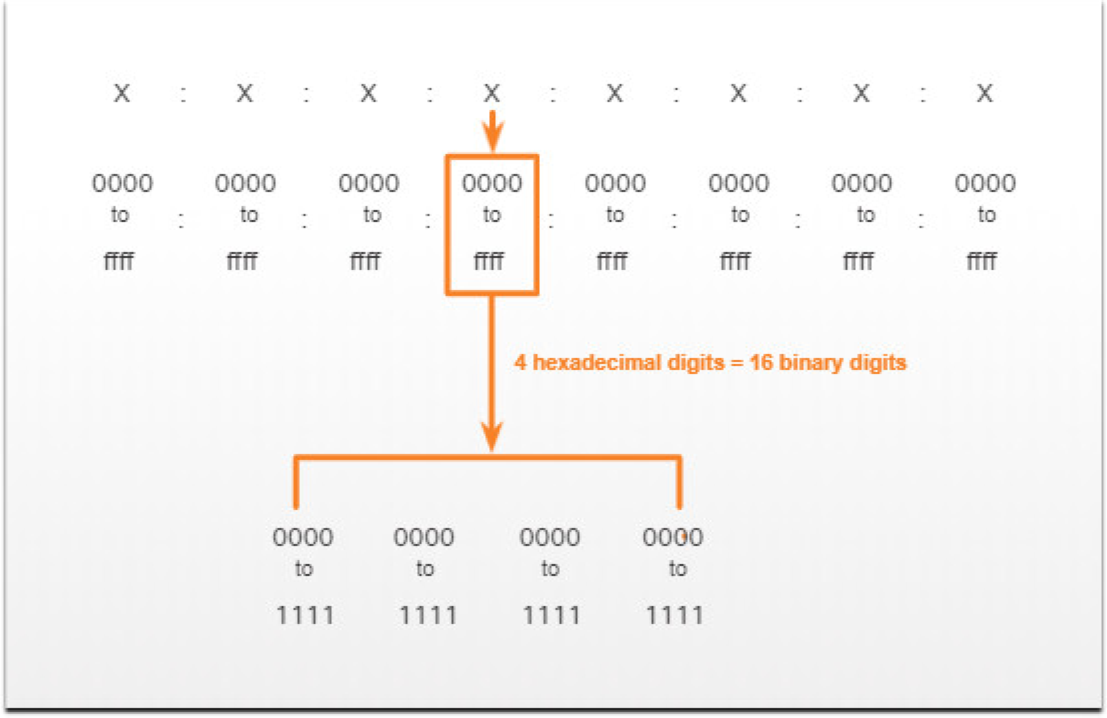
Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Binary and hexadecimal work well together because it is easier to express a value as a single hexadecimal digit than as four binary bits.

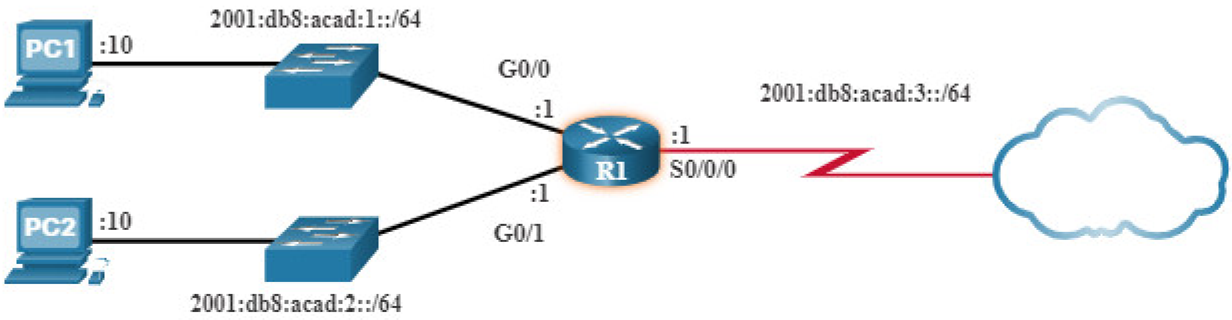
The hexadecimal numbering system is used in networking to represent IP Version 6 addresses and Ethernet MAC addresses.

IPv6 addresses are 128 bits in length and every 4 bits is represented by a single hexadecimal digit; for a total of 32 hexadecimal values. IPv6 addresses are not case-sensitive and can be written in either lowercase or uppercase.

As shown in the figure, the preferred format for writing an IPv6 address is x:x:x:x:x:x:x:x, with each “x” consisting of four hexadecimal values. When referring to 8 bits of an IPv4 address we use the term octet. In IPv6, a hextet is the unofficial term used to refer to a segment of 16 bits or four hexadecimal values. Each “x” is a single hextet, 16 bits, or four hexadecimal digits.


X : X : X : X : X : X : X : X
0000 to ffff : 0000 to ffff : 0000 to ffff : 0000 to ffff : 0000 to ffff : 0000 to ffff : 0000 to ffff : 0000 to ffff
4 hexadecimal digits = 16 binary digits
0000 to 1111
0000 to 1111
0000 to 1111
0000 to 1111

The sample topology in the figure displays IPv6 hexadecimal addresses.



5.2.2. Video – Converting Between Hexadecimal and Decimal Numbering Systems

5.2.3. Decimal to Hexadecimal Conversions

Converting decimal numbers to hexadecimal values is straightforward. Follow the steps listed:

1. Convert the decimal number to 8-bit binary strings.
2. Divide the binary strings in groups of four starting from the rightmost position.
3. Convert each four binary numbers into their equivalent hexadecimal digit.

The example provides the steps for converting 168 to hexadecimal.

For example, 168 converted into hex using the three-step process.

1. 168 in binary is 10101000.
2. 10101000 in two groups of four binary digits is 1010 and 1000.
3. 1010 is hex A and 1000 is hex 8.

Answer: 168 is A8 in hexadecimal.

5.2.4. Hexadecimal to Decimal Conversion

Converting hexadecimal numbers to decimal values is also straightforward. Follow the steps listed:

1. Convert the hexadecimal number to 4-bit binary strings.
2. Create 8-bit binary grouping starting from the rightmost position.
3. Convert each 8-bit binary grouping into their equivalent decimal digit.

This example provides the steps for converting D2 to decimal.

1. D2 in 4-bit binary strings is 1101 and 0010.
2. 1101 and 0010 is 11010010 in an 8-bit grouping.
3. 11010010 in binary is equivalent to 210 in decimal.

Answer: D2 in hexadecimal is 210 in decimal.

5.3. Module Practice and Quiz

5.3.1. What did I learn in this module?

Binary Number System

Binary is a numbering system that consists of the numbers 0 and 1 called bits. In contrast, the decimal numbering system consists of 10 digits consisting of the numbers 0 – 9. Binary is important for us to understand because hosts, servers, and network devices use binary addressing, specifically, binary IPv4 addresses, to identify each other. You must know binary addressing and how to convert between binary and dotted decimal IPv4 addresses. This topic presented a few ways to convert decimal to binary and binary to decimal.

Hexadecimal Number System

Just as decimal is a base ten number system, hexadecimal is a base sixteen system. The base sixteen number system uses the numbers 0 to 9 and the letters A to F. The hexadecimal numbering system is used in networking to represent IPv6 addresses and Ethernet MAC addresses. IPv6 addresses are 128 bits in length and every 4 bits is represented by a single hexadecimal digit; for a total of 32 hexadecimal values. To convert hexadecimal to decimal, you must first convert the hexadecimal to binary, then convert the binary to decimal. To convert decimal to hexadecimal, you must also first convert the decimal to binary.

5.3.2 Module Quiz – Number Systems

Download Slide Powerpoint (PPT)

PPT

CCNA 1 v7.0 Curriculum: Module 5 - Number Systems.pptx

1 file(s) 1.05 MB

Download

Tags:ccna 1 v7 modules