**COMPUTER NUMBER SYSTEMS**

**EXPONENTS**

When a number (**n**) is raised to an exponent (**a**), it is writtenna. What does this mean? **n** is multiplied by itself **a** times. This is also known as raising **n** to the **a** power. There are two special cases: Any number raised to the first power is the number itself and any number raised to the zero power is one. Some examples:

105 = 10 \* 10 \* 10 \* 10 \* 10 = 100,000

23 = 2 \* 2 \* 2 = 8

52 = 5 \* 5 = 25

81 = 8

40 = 1

21 = 2

60 = 1

**DECIMAL SYSTEM**

The decimal system (base 10) contains ten digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. Decimal’s base is 10. Any positive integer can be represented in expanded notation as a sum of the powers of 10. For example 8253 can be expressed as follows:

825310 = 8 \* 1000 + 2 \* 100 + 5 \* 10 + 3 \* 1

= 8 \* 103 + 2 \* 102 + 5 \* 101 + 3 \* 100

**BINARY SYSTEM and BINARY-TO-DECIMAL CONVERSION**

The binary system has two digits, 0 and 1; it has a base of two. These binary digits (or *bits)* are combined in groups of eight and called *bytes* or groups of four called *nibbles*. A binary number can be written in expanded notation as the sum of each digit times that digit's place value. For example,

1101012 = 1 \* 25 + 1 \* 24 + 0 \* 23 + 1 \* 22 + 0 \* 21 + 1 \* 20

= 1 \* 32 + 1 \* 16 + 0 \* 8 + 1 \* 4 + 0 \* 2 + 1 \* 1

= 32 + 16 + 0 + 4 + 0 + 1

= 53

|  |  |
| --- | --- |
| **POWER OF TWO** | **DECIMAL VALUE** |
| 28 | 256 |
| 27 | 128 |
| 26 | 64 |
| 25 | 32 |
| 24 | 16 |
| 23 | 8 |
| 22 | 4 |
| 21 | 2 |
| 20 | 1 |

**DECIMAL-TO-BINARY CONVERSION**

To convert a decimal number (base 10) to its binary equivalent (base 2), find the largest power of two that is less than the number.

For example, to convert 109 to binary using 8 bits, 26 (64) is the largest power of two that is less than the number. Zero fill to the left of this digit.

0 1 1 0 1 1 0 1

128 64 32 16 8 4 2 1

109 - 64 = 45

45 - 32 = 13

13 - 8 = 5

5 - 4 = 1

1 - 1 = 0

Therefore, 10910 = 0110 11012.

**Practice Problems**

1. Convert the following binary numbers to decimal

(a) 0111 0011

So we count from right to left the results of 20 all the way to 27.

Now we add together all the powers of 2 that are a “1” in the code.  
 So based on this order we have 1 + 2 + 0 + 0 + 16 + 32+ 64 + 0.

When you add those together you get 115 which is our answer!

(b) 0011 0101

Count from right to left the results of 20 all the way to 27.  
 With all of our values in place we add all the powers of 2’s for the “1”s in the code

With that being said we 1 + 0 + 4 + 0 + 16 + 32 + 0 + 0.

When you add them together you get 53 which is the answer!

2. Convert the following decimal numbers to binary.

(a) 91

So we need to find the highest power of 2 that fits into 91.

When we write them out we get 1, 2 , 4, 8, 16, 32, 64, 128. We stop at 64.

64 goes into 91, we put a 1 in its place and then we subtract and get 27

32 doesn’t go into 27 so we put a zero in its place and go to the next and yes 16 does fit so we put a 1 in its place and subtract and get 11. 8 goes into 11 so we put a 1 and subtract and get 3.

2 goes into 3 so we put a 1 and get 1 and then 1 goes into 1 so we put a 1 in its place. When we put it together the binary number it comes to 0101 1011

(b) 255

So the highest power of 2 we have for this number is 128. We put a 1 in its place and subtract 128 from 255 and that leaves us with 127. 64 goes into 127 so we put 1 and subtract 64 from 127 which gives us 63. 32 goes into 63 so we put 1, 63 -32 = 31. 16 goes into 31 so its a 1, 31 - 16 = 15. 4 goes into 15 so we put a 1 and 15 - 4 = 11 which 2 goes into so we place another 1 and 11 - 2 = 9. 1 goes into 9 so we 9 - 1 and we place a 1 in its place. All in all we put the binary numbers together we get 1111 1111

(c) 473

So first thing to note with this is that it goes past the byte size of 8 bits since the max highest power of two is 2^8 = 256. First we 473 - 256 = 217, place a “1” value. Now we 217 - 128 = 89, place a “1”. Next 89 - 64 = 25, place a “1” since 64 goes into 89. 32 doesn’t go into 25 so we skip it and leave a “0”. Next 16 goes into 25 so we 25 - 16 = 9 and place “1”. 8 goes into 9 so we place a 1 and 9 - 8 = 1. Next 4 doesn’t fit into 1 so we place a “0”, neither does 2 so we place a “0” next and then lastly 1 goes into 1 so we place a 1. Now keep in mind we have 9 values here so we technically are working with 2 bytes so we have to write it out fully. The binary code is 0000 0001 1101 1001

Answers to problems:

1. (a) 115 (b) 53

2. (a) 0101 1011 (b) 1111 1111

(c) 0000 0001 1101 1001

Note: spacing in the binary numbers is only used to make answers easier to read.