**Introduction**

* Binary # system: the system the computer works within. It is 1 and 0. When the computer is on it is 1 and when the computer is off it is 0. Many switches within a computer communicate what we want to the actual computer with 1’s and 0’s
* Each chip has transistors, and they transistors do the communication of being on and off
* Humans use the Deca systems: 0,1,2,3,4,5,6,7,8,9. There are 10 numbers
* Computers read the Binary system: 0,1. There are 2 numbers
* Other system is the Hexa system: 0,1,2,3,4,5,6,7,8, A,B,C,D,E,F

**The format**

* The format of the number system is to the exponential power. Meaning whatever the base of the system and we go to the power of that base
* Binary is base 2
* Always starts 2^0, then 2^1, then 2^2,2^3,2^4 and so on. Every time we add a zero, but we keep our base at 2
* 2^0, 2^1, 2^2,2^3,2^4 = 1,2,4,8,16
* ***Side note***: this number system is the base of the numbers of memory and operating systems. For Exp -> 32-bit operating system is just 2^5. Therefore, anything involves the main components of the computers gets its numbers based on the binary numbering system.
* Binary numbering system when written is much wider than Deca and Hexa
* To represent the number 10 you need at least 4 digits (1010), while you need far less digits to represent that number when it is Deca or Hexa
* We still use the binary because this is how regular computers work, because their transistors must be turned on and off and that represent 0 and 1, unless we have an analog computer, which is a computer of multiple states not just the 1s and 0s. The analog computer is what the field of quantum computers is all about
* Deca is base 10
* Always starts 10^0, then 10^1, then 10^2 and so on.
* 1,10,100
* Hexa is base 16
* Always starts 16^0, then 16^1, then 16^2 and so on.
* 1,16,256
* For RAM (Random Access Memory) uses the hexa system, so we do not need to waste space representing the available numbers. This is fine because this is just the short-term memory

Conversion Binary - Deca

* Binary to Deca
* For Exp convert 10110011 to Deca
* The steps to solve it
* Break down the numbers to base 2 starting from right to left like the table below. Then only add the results of the bases that are represented by 1 not 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 2^7=128 | ~~2^6 64~~ | 2^5 32 | 2^4 16 | ~~2^3 8~~ | ~~2^2 4~~ | 2^1 = 2 | 2^0 = 1 |
| 128+32+16+2+1 = 179 | | | | | | | |

* Therefore 10110011 in binary is equal to 179 in Deca
* Deca to Binary
* For Exp convert 296 to Binary
* The steps to solve it
* 1. Write all the binary results in a table.
* 2. Then check if you can subtract the number you want to convert (296) from each cell starting from the left to the right and **keep subtracting** until you reach zero.
* 3. Start with the largest possible number you can subtract from without getting a negative.
* 4. Once you get to zero stop subtracting.
* 5. Then for each value that did not produce negative give its cell a 1 and for each value that gave negative give its cell a 0
* You always need to start your results conversion number with 1 and go from left to right. Therefore, you do not need to include the large numbers that will give a negative to the number you want to cover. For Exp (512 to the 296)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2^9 =512 | 2^8 =256 | 2^7=128 | 2^6 64 | 2^5 32 | 2^4 16 | 2^3 8 | 2^2 4 | 2^1 = 2 | 2^0 = 1 |
| 296-512 = negative, can’t add it | 296-256 =40 | 40-128 = negative, can’t add it | 40-64 = negative, can’t add it | 40-32 =  8 | 8-16 = negative, can’t add it | 8-8 =0, STOP subtracting | - | - | - |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| The binary number is for 296 = 100101000  You do not need to include the 0 of 512, but include every zero that came after the first number that gave a none zero results. Which in this example was 256. You do not need to include the leading zeros | | | | | | | | | |

* Therefore, the binary number of 296 is 100101000

Conversion Binary - Hex

* Binary to Hex
* The Hex allows us to represent a large Binary number to a small Hex value
* Hex can go from 0-15. While a four bits binary number also goes from 0-15
* For Exp convert 1111 0101 1001 1010 to Hex
* The steps to solve it
* 1. Separate the binary bits into sets of 4 to the best you can
* If you do not have a number that is divided equally to 4 bits:
* 1. Start the Binary number from right to left and separate each segment to 4 bits, until you have a last segment that is less than 4 bits. Fill that last segment’s empty spaces with 0s to make them also 4 bits. Then go to step 2 of calculating the sum of each segment
* 2. Calculate the sum of each set
* 3. Conver each summed set to its respective Hex representation and that is the representation of your Binary to Hex 1111 0101 1001 1010 = F 5 9 A
* F = 15 and A =10
* 4. Represent the final number as a Hex number by adding 0x in front of the number
* 0xF59A
* For Exp convert 0x 7B93 to Binary
* The steps to solve it
* 1. Separate each number individually
* 2. Get the bits that represent each Hex number
* 3. Because each the max Binary bit set of 4 bits is equal to 15 which is F in Hex
* 0x 7B93 = 0111 1011 1001 0011

Conversion Decimal - Hex

* Decimal and Hex do not divide to each other because a 16 goes to 10 or 10 goes to 16