

Forum:[Hack Forums](#)

Thread: [Decimals,Bits and Hexadecimals | Calculating](#)

Tafelpoot - December 31, 2013, 6:12 am

Decimals,Bits and Hexadecimals | Calculating

I advice you to learn the basics of Decimals,Bits and Hexadecimals first, Here is a Tutorial that will teach you everything you need to know for this one.

[Decimals,Bits and Hexadecimals | A numeral system explained](#)

Binair to Decimal

You can use the graphical view of binair numbers to transform them to decimal numbers.

Therefore you need to put the binair numbers above the power of 2. The power of 3 wit a 1 below needs to be add up.

Decimal to Binair

Transforming of a number in decimal notation to binair notation is a bit harder. You can't easily recognize the power of 2 in a number.

You would need to be a math genius to see that 91 equal to $64 + 16 + 8 + 2 + 1$.

The last bit in a byte is easy to recognize. The numbers 2, 4, 8, 16, ... are all even, By summing up you can only make even numbers.

If a number is odd the binair notation will end on a 1, Otherwise on a 0. This is the basic algorithm to convert decimal to binair notation. The rules are below.

Here is an example on how the rules work for the number 5 :

5 is odd, Write down a 1 (...1), Minus 1(4) en divide by 2 gives 2.

2 is even, Write down a 0 (...01), Divide by 2 gives 1.

1 is odd, Write down a 1 (...101), Minus 1, This gives 0.

We are now at 0, The result is 101.

Binair calculations

With binair numbers you can also calculate. There are rules that are similar to the math rules for decimal numbers. In this thread i will not give an extensive lesson.

This is not needed because the computer already does the math for you.

We discuss a example, Summing of binair numbers. Later on we will use the same example to show how the machine does these calculations.

Summing up of 2 bits is easy. The answer is always written in 2 bits.

Only the right one looks weird. You would think that there would be just $1+1=2$.

When you keep following this rule, The summing up will be the same as with decimal numbers. Here is an example :

Overall we don't write the numbers with a specific amount of numbers, We regularly use as much as we need. You could write 000011 instead of 11, because the 0's upfront do not matter.

Computer systems use for integers a specific amount of bits, Normally 16, 32 or 48. If it's needed, 0's will be added to the front of the number. The number 5(101) would be shown like this in 16 bits : 0000000000000101.

In the scheme you can see how big numbers with 16,32 or 48 bits can get. With a number of 16 bits there are 65.536 possible values, You can count from 0 to 65.536.

A number with 32 bits has 4.294.967.295 possible values.

The standard amount of bits of numbers can also give problems. In the above example you could see that the outcome from a sum could give more numbers (5 bits) then the summed up numbers(4 bits). If you sum up 2 numbers of 16 bits, You can get a result that needs 17 bits. This is not possible an error will occur. You can compare this to the meter in your car. If it says 999999 and you drive 1 more Mile, It will become 000000.

Of course it needs to be 1000000, But the number in the front is not there, This is called overflow. Mayby you have seen a similar error in a spreadsheet or while programming.

Here is a little program i made in Excel which calculates Binair notation to Decimal notation.

Download link :

https://mega.co.nz/#!BBN10TCC!MdKLRj6_cxjlboH22ZkFTSouDrRyjyg511OW3PYla9g

This was the Tutorial on how to do calculations with Binair and Decimal notation.
The next tutorial i will be talking about ASCII and UNICODE.

Thanks for readding
Leave some feedback

Regards VB.