Notes on binary numbers

Base of a number

We represent numbers by using digits that are placed in a specific order. The representation is formed according to the system we use to represent the number in question. The choice of digits we use within the system, forms its base.

The number of digits used within the system to describe a number is the base of the system.

For example, in the decimal system, numbers are described by using ten digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. So, the base of the system is 10. In the binary system, numbers are described by using two digits: 0 and 1.

To let the reader know which number base is used to represent the referenced number, we may use a prefix, or a suffix to that number. Assuming the number 130 in decimal, popular examples of such notations are:

	Decimal	Binary	Hexadecimal
Prefix	0d130	0b10000010	0x82
Suffix	130_{10}	10000010_{2}	82_{16}

Why binary

Computers use the binary system because it is easy to represent 1 and 0 in a circuit. Possible ways to represent 1 and 0 within a computing system are:

1. The amount of a physical quantity

   The flow of electrical current (power transistors), or the amount of trapped charge (CMOS transistors, flash memory cells, EPROM memory cells, DRAM memory cells) above a specific level represents a 1, whilst the presence of a quantity below another specific level represents a 0. Usually, the two boundaries coincide, but not always. Circuit designers ensure that state between the two boundaries is avoided.

2. The orientation of a magnetic field

   Used in magnetic media (hard disks, tapes, floppy disks). The head of the drive flies above the surface of the medium. During a read operation it detects the orientation of the magnetic field generated by a part of the surface of the device. Since there are only two choices available, one orientation represents 1 and the other 0. During a write operation, the head forces the field to orient itself in the direction required.

3. The polarisation of light reflected on a surface

   Used in CD and DVD media. A UV laser beam is shone on the surface of the medium. Mechanical construction (read only media), or thermal deformation (writeable/re-writeable media) polarises the surface of the medium, thus effecting the reflection of the beam. A 1 is represented when the amount of reflected light exceeds a defined level. A 0 is represented by a low level of reflected light.

4. The position of a charge within a crystal

   Used in high speed, high endurance, non-volatile memories (FRAM). A small amount of charge is trapped in a crystal structure and is forced to stay at either of the extreme positions of the structure. The position chosen to place the charge, represents 1, or 0.

Binary representation of numbers

Numbers are represented by using the digits available within the chosen arithmetic system. Digits are placed next to each other to form the number. The value of each digit used is determined by its position and the base of the number system used. By convention, digits placed at the right-most position of the number carry the least weight, while digits placed at the left-most position of the number carry the maximum weight.

Considering an integer number, the first digit at the right-most position carries base^0 weight, this is the least significant digit. The second digit, carries base^1 weight, the third digit carries base^2 weight and so on. The digit at the left-most position carries the maximum weight and therefore, is the most significant digit.

For the case of the decimal and binary systems, an example of the weights in an integer number is:

			Decimal		
	10^4	10^3	10^2	10^1	10^0
Weight:	10000	1000	100	10	1

			Binary		
	2^4	2^3	2^2	2^1	2^0
Weight:	16	8	4	2	1

An example of the weights in a mixed number carrying two digits for its fractional part, is:

			Decimal		
	10^2	10^1	10^0	10^{-1}	10^{-2}
Weight:	100	10	1	.1	.01

			Binary		
	2^2	2^1	2^0	2^{-1}	2^{-2}
Weight:	4	2	1	0.5	0.25

To calculate the represented number, we multiply the value of the digit used by the weight of its position and then add the results of all multiplications.

Example to calculate the decimal value of the binary number 0b10100110

Digit	1	0	1	0	0	1	1	0
Weight	128	64	32	16	8	4	2	1
Value of digit in decimal	1 * 128 = 128	0 * 64 = 0	1 * 32 = 32	0 * 16 = 0	0 * 8 = 0	1 * 4 = 4	1 * 2 = 2	0 * 1 = 0

Value of number in decimal: 128 + 0 + 32 + 0 + 0 + 4 + 2 + 0 = 166_{10}

Binary addition and subtraction

Binary numbers are added and subtracted in the same way as decimal numbers are. When adding, if the result exceeds the maximum number of the base, the excess value is carried to the digit carrying the next higher weight. When subtracting, if the result subceeds the minimum number of the base, the missing value is borrowed from the digit carrying the next higher weight. Examples of decimal addition and subtraction will be considered to clarify the meaning of “carry” and “borrow”.

Consider the addition of numbers A and B, where A = 0d84 and B = 0d39. The steps taken to calculate the result are:

Step 1: the units of A and B are added: 4 + 9 = 13. The result, 13, is larger than the base of the system, 10. We keep the units of the result, 3 and we carry 1 decade to the addition of the next significant digit.

Step 2: the decades of A and B are added, but this time we also consider the decade that was carried over from step 1. Number A contributes 8 decades, number B contributes 3 decades and the carry from step 1, contributes 1 decade. The result for the decades is: 8 + 3 + 1 = 12

Step 3: we put the above results together and we get the answer:

84 + 39 = 123

Consider the subtraction of numbers A and B, where A = 84 and B = 39. The steps taken to calculate the result are:

Step 1: the units of A and B are subtracted: 4 - 9 = -5. The result, -5, is less than 0. We keep the units of the result, 5 and we borrow 1 decade from the subtraction of the next significant digit.

Step 2: the decades of A and B are subtracted, but this time we also consider the decade that was borrowed to step 1. The 3 decades of number B are subtracted from the 8 decades of number A, and we also subtract the one decade that was borrowed to step 1. The result for the decades is: 8 - 3 - 1 = 4

Step 3: we put the above results together and we get the answer:

84 - 39 = 45

The above steps are considered to be elementary and we perform the calculation without thinking about it. However, they are presented here because they will help us to replicate the process for the cases of binary addition and subtraction.

Binary addition

Binary addition follows the same steps as decimal addition. To help clarify the process, the four rules governing it are summarised below.

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10 (a carry is required)

Consider the addition of numbers A and B, where A = 0b1010 and B = 0b0110. The steps taken to calculate the result are:

Step 1: We add the least significant bit (bit 0) of each of the two numbers: 0 + 0 = 0. So bit 0 of the result is equal to 0.

Step 2: We add the bit 1 of each of the two numbers: 1 + 1 = 10. This time, the result, 10, is greater that the base of the number. A 1 needs to be carried to the next significant bit. So, bit 1 of the result is equal to 0 and a 1 is carried over to the next step.

Step 3: We add the bit 2 of each of the two numbers, but this time we also consider the bit that was carried from step 2. Bit 2 of number A is 0, bit 2 of number B is 1 and the bit carried from step 2 is 1. 0 + 1 + 1 = 10. Again, the result, 10, is greater that the base of the number. A 1 needs to be carried to the next significant bit. So, bit 2 of the result is equal to 0 and a 1 is carried over to the next step.

Step 4: We add the bit 3 of each of the two numbers, and also this time we consider the bit that was carried from step 3, too. Bit 3 of number A is 1, bit 3 of number B is 0 and the bit carried from step 3 is 1. 1 + 0 + 1 = 10. Again, the result, 10, is greater that the base of the number. Because there are no further bits to work with, bit 3 of the result is equal to 0 and an extra bit, equal to 1 is added as the most significant bit of the result.

Step 5: we put the above results together and we get the answer:

0b1010 + 0b0110 = 0b10000

Binary subtraction

Binary subtraction follows the same steps as decimal subtraction. To help clarify the process, the four rules governing it are summarised below.

0 − 0 = 0
1 − 0 = 1
1 − 1 = 0
10 − 1 = 1 (a borrow is required)

Consider the subtraction of number B from number A, where A = 0b1010 and B = 0b0110. The steps taken to calculate the result are:

Step 1: We subtract the least significant bit (bit 0) of each of the two numbers: 0 - 0 = 0. So bit 0 of the result is equal to 0.

Step 2: We subtract the bit 1 of each of the two numbers: 1 - 1 = 0. So bit 1 of the result is equal to 0.

Step 3: We subtract the bit 2 of each of the two numbers: 0 - 1 = 1. Because 1 > 0, a 1 must be borrowed from the subtraction of the next significant bit. So, bit 2 of the result is equal to 1 and a 1 is borrowed from the next step.

Step 4: We subtract the bit 3 of each of the two numbers, but this time we also consider the bit that was borrowed to step 3. The 0 of number B is subtracted from the 1 of number A, and we also subtract the 1 that was borrowed to step 3: 0 - 1 - 1 = 0. . So, bit 3 of the result is 0

Step 5: we put the above results together and we get the answer:

0b1010 - 0b0110 = 0b0100

Representing negative numbers – two's complement

So far, only positive integer numbers were considered, however, a computer needs to perform calculations using negative numbers too. The two's complement operation is the most popular method used to reverse the sign of an integer binary number. Under this scheme, the most significant bit is used to denote the sign of the integer. For an n bit integer, the largest positive number that can be represented is 2^{n-1} -1 and the lowest negative number is -2^{n-1}. So, for an 8 bit number, the largest positive number represented is 128 and the lowest negative number is -127.

The sum of an N bit number and its two's complement is equal to 2^N.

There are two ways to calculate the 2's complement of a binary number:

a. invert all bits of the number and then add one.

b. start from the LSB and work towards the MSB. Copy every bit on the way up to and including, the first 1 encountered. Invert all remaining bits

Examples:

1. Convert 0b01010101:
method a:
Step 1: invert all bits. The intermediate number formed is 10101010.
Step 2: add one: 10101010 + 00000001 = 10101011

method b:
Step 1:copy bit 0. It is equal to 1. The intermediate number formed is:

00000001

Step 2:invert all remaining bits. The number becomes:

10101011

Verification: The addition of an N bit number to its two's complement is equal to 2^N. In this case, we have an 8 bit number, therefore the sum of the number and its two's complement should be 2^8 = 0d256 = 0b100000000.

01010101 + 10101011 = 100000000

therefore, the above conversion is correct.

2. Convert 0b01100100:
method a:
Step 1: invert all bits. The intermediate number formed is 10011011.
Step 2: add one: 10011011 + 00000001 = 10011100

method b:
Step 1: copy bits 0, 1, 2. Bit 2 is equal to 1. The intermediate number formed is:

00000100

Step 2:invert all remaining bits. The number becomes: 10011100

Verification:

01100100 + 10011100 = 100000000

therefore, the above conversion is correct.