PROJECT WHIRLWIND

Report R-90-1

THE BINARY SYSTEM OF NUMBERS

Submitted to the
OFFICE OF NAVAL RESEARCH
Under Contract N5ori60
Project NR-048-097

Report by
Margaret Florencourt Mann
DIGITAL COMPUTER LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge 39, Massachusetts

Project DIC 6345

January 15, 1946
(revised: February 29, 1952)

Report R-90-1

ABSTRACT

The representation of decimal numbers in the binary system and the processes of binary arithmetic are explained.

THE BINARY SYSTEM OF NUMBERS

I. REPRESENTATION OF NUMBERS

The decimal system takes its name from the fact that it is based on ten digits (0, 1, ... 9) and all numbers are composed of those 10 digits. The binary system, analogously, takes its name from the fact that it is based on 2 digits, (0, 1), and all numbers in the binary system are made up of those 2 digits. The decimal system has a base of 10; the binary system has a base of 2.

Decimal System	Equivalence	Binary System	Equivalence
1	$1x10^0$	1	$1x2^0 = 1$
2	$2x10^0$	10	$1x2^1 + 0x2^0 = 2$
3	$3x10^0$	11	$1x2^1 + 1x2^0 = 3$
4	$4x10^0$	100	$1x2^2 + 0x2^1 + 0x2^0 = 4$
5	$5x10^0$	101	$1x2^2 + 0x2^1 + 1x2^0 = 5$
6	$6x10^0$	110	$1x2^2 + 1x2^1 + 0x2^0 = 6$
7	$7x10^0$	111	$1x2^2 + 1x2^1 + 1x2^0 = 7$
8	$8x10^0$	1000	$1x2^3 + 0x2^2 + 0x2^1 + 0x2^0 = 8$
9	$9x10^0$	1001	$1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 = 9$
10	$1x10^1 + 0x10^0$	1010	$1x2^3 + 0x2^2 + 1x2^1 + 0x2^0 = 10$
11	$1x10^1 + 1x10^0$	1011	$1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 11$
12	$1x10^1 + 2x10^0$	1100	$1x2^3 + 1x2^2 + 0x2^1 + 0x2^0 = 12$
13	$1x10^1 + 3x10^0$	1101	$1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 13$
14	$1x10^1 + 4x10^0$	1110	$1x2^3 + 1x2^2 + 1x2^1 + 0x2^0 = 14$
15	$1x10^1 + 5x10^0$	1111	$1x2^3 + 1x2^2 + 1x2^1 + 1x2^0 = 15$
16	$1x10^1 + 6x10^0$	10000	$1x2^4 + 0x2^3 + 0x2^2 + 0x2^1 + 0x2^0 = 16$
17	$1x10^1 + 7x10^0$	10001	$1x2^4 + 0x2^3 + 0x2^2 + 0x2^1 + 1x2^0 = 17$
20	$2x10^1 + 0x10^0$	10100	$1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 0x2^0 = 20$

Decimal numbers, since they have a base of 10, may be broken up into powers of 10:

e.g. $305.798 = 3x10^2 + 0x10^1 + 5x10^0 + 7x10^{-1} + 9x10^{-2} + 8x10^{-3}$

In the same way, binary numbers, since they have a base of 2, may be broken up into powers of 2:

e.g. $101.011 = 1x2^2 + 0x2^1 + 1x2^0 + 0x2^{-1} + 1x2^{-2} + 1x2^{-3}$

It can be seen from this arrangement of the powers of the bases, that the decimal places (units, tens, hundreds, tenths, hundredths, thousandths, etc.) have a definite relation to the powers of the base 10 in the decimal system. They are numbered off consecutively from left to right, from $+\infty$ to $-\infty$, these numbers corresponding exactly with the powers of the base 10; the decimal point is placed between the units (0) place and the tenths (-1) place.

The binary places (units, twos, fours, eights, sixteens, halves, fourths, eighths, etc.) are also numbered exactly according to the powers of the base 2; the binary point is placed between the units (0) place and the halves (-1) place. Therefore, the place and point arrangement is the same in both decimal and binary systems.

e. g.

Decimal Place and Binary Place No. / No.	$+\infty$ . . .	3,	2,	1,	0.	-1,	-2,	-3,	. . . . $-\infty$
305.798			3	0	5.	7	9	8	
101.011			1	0	1.	0	1	1	

II. CONVERSION OF DECIMAL NUMBERS TO BINARY NUMBERS

In order to convert a number from the decimal system to the binary system, the number must be changed from powers of 10 to powers of 2; therefore, the powers of 2 are taken out of the decimal number. They may be taken out as follows in a brute-force manner, or more simply as shown later in an algorism. Follow the work sheet at the end of the examples.

a. Integers.

Example 1 Given: 18

To Find: Binary Equivalent (First Method)

Method: Extraction of Powers of 2.

Take out of the decimal number, 18, the highest power of 2, = 16 = 2^4. A 1 is now placed in binary place No. 4, corresponding to the power of 2 found. Subtracting 16 from 18 leaves 2. The highest power of 2 in 2 is $2^1 = 2$; therefore, put a 1 in binary place No. 1. Subtracting 2 from 2 leaves 0, so the conversion is completed. Since 4 and 1 were the only powers of 2 in the given number, 1's appear only in binary places 4 and 1. The coefficients of the non-appearing powers must have been zero, so zeros are entered under all the other binary place numbers.

Example 2 Given: 730

To Find: Binary Equivalent (First Method)

Method: Extraction of Powers of 2.

The highest power of 2 in 730 is $2^9 = 512$, so a 1 goes in No. 9
730 - 512 = 218

The highest power of 2 in 218 is $2^7 = 128$, " " " " " No. 7
218 - 128 = 90

The highest power of 2 in 90 is $2^6 = 64$, " " " " " No. 6
90 - 64 = 26

The highest power of 2 in 26 is $2^4 = 16$, " " " " " No. 4
26 - 16 = 10

The highest power of 2 in 10 is $2^3 = 8$, " " " " " No. 3
10 - 8 = 2

The highest power of 2 in 2 is $2^1 = 2$, " " " " " No. 1

No other powers of 2 appear so their coefficients must be zero.

Place No. / No.	10	9	8	7	6	5	4	3	2	1	0.	-1	-2	-3	-4	-5	-6	-7	-8
18	0	0	0	0	0	0	1	0	0	1	0.	0	0	0	0	0	0	0	0
730	0	1	0	1	1	0	1	1	0	1	0.	0	0	0	0	0	0	0	0
0.147	0	0	0	0	0	0	0	0	0	0	0.	0	0	1	0	0	1	0	1

A simpler method of conversion of decimal <u>integers</u> to binary integers is shown in the following algorism. Powers of 2 are taken out of the decimal number by successive divisions by the base 2. The remainders after successive divisions of the number (and its quotients resulting from successive divisions by 2) indicate the coefficients of the powers of the base.

Using the same examples:

Given: Decimal Number 18

To Find: Binary Equivalent (Simpler Method)

Method: Algorism

If there is a 0's power of 2 (2^0) contained in the number, its presence will be indicated by a remainder after the first division of the given number by 2. If there is a 1's power of 2 (2^1) contained in the number, its presence will be indicated by a remainder after the first division of the resulting quotient by 2. If there is a 2^2 contained in the number its presence will be indicated by a remainder after the next division of the resultant quotient by 2, etc. That is, the coefficients of the powers of 2 are the remainders after successive divisions.

2) 18	dividend	
2) 9	quotient	0 remainder = coefficient 0×2^0
2) 4	quotient	1 remainder = coefficient 1×2^1
2) 2	quotient	0 remainder = coefficient 0×2^2
2) 1	quotient	0 remainder = coefficient 0×2^3
0	quotient	1 remainder = coefficient 1×2^4

This same algorism may be applied to the conversion of the decimal number 730 to a binary number:

2) 730	
2) 365	0×2^0
182	1×2^1
91	0×2^2
45	1×2^3
22	1×2^4
11	0×2^5
5	1×2^6
2	1×2^7
1	0×2^8
0	1×2^9

1011011010. = 730.

b. Fractions

Given: .147

To Find: Binary Equivalent

Method: Extraction of Powers of 2

The highest power of 2 in .147 is 2^{-3} = .125 so a 1 goes in No.-3
.147 - .125 = .022

The next power of 2 in order is 2^{-4} = .0625, but this power of 2 is not contained in .022, so coefficient of the (-4) place = 0.

2^{-5} = .03125; not in .022, so 0 in No. (-5).

2^{-6} = .015625; is in .022, so 1 in No. (-6).
.022 - .015625 = .006375

2^{-7} = .0078125; not in .006375, so 0 in No. (-7), etc.

That method of converting a decimal to the binary system always works, but it is laborious and offers many chances for mistakes in the division and subtraction of such long numbers.

There is another method based on the same principle of taking out powers of 2, which, however, is much simpler. Given a decimal: -- by the former method, if it is larger than 2^{-1} (=.5), a 1 goes in the -1 place; if the number (or the remainder after subtraction of .5) is greater than or equal to .25 (=2^{-2}), a 1 goes in the -2 place. However, it is the same thing to say if twice the given decimal is larger than 2 x 2^{-1} (=1), a 1 is put in the -1 place; if 4 times the given decimal is larger than 4 x 2^{-2} (=1), a 1 is put in the -2 place. If 8 times the given decimal is larger than 8 x 2^{-3} (=1), a 1 is put in the -3 place. This is the same thing as doubling the number (or its remainder after a power of 2 is taken out) at each step and comparing it with 1. If the result becomes greater than 1, a 1 is taken out, and the doubling process starts again on the remainder.

Given: .147

To Find: Binary Equivalent

Method: Algorism

The former method started out by asking:

is .147 $\geq$ .5? (If so, a 1 goes in No. -1; if not, a 0 goes in -1.)

is .147 $\geq$ .25? (" " " " " " No. -2; " " " " " " -2.)

is .147 $\geq$ .125? (" " " " " " No. -3; " " " " " " -3.)

This method starts out by asking:

is $2(.147) \geq 1$? (If so a 1 goes in No. -1; if not, a 0 goes in -1.)
is $2x2(.147) \geq 1$? (" " " " " " No. -2; " " " " " " -2.)
is $2x2x2(.147) \geq 1$? (" " " " " " No. -3; " " " " " " -3.)

$2(.147) = .294 \ngeq 1$, therefore, 0 in No. -1
$2 \times 2(.147) = .588 \ngeq 1$, " 0 " No. -2
$2 \times 2 \times 2(.147) = 1.176 > 1$, " 1 " No. -3

$(1.176 - 1.000 = .176)$

$2(.176) = .352 \ngeq 1$, therefore, 0 in No. -4
$2 \times 2(.176) = .704 \ngeq 1$, " 0 " No. -5
$2 \times 2 \times 2(.176) = 1.408 > 1$, " 1 " No. -6

$(1.408 - 1.000 = .408)$

$2(.408) = .816 \ngeq 1$, therefore, 0 in No. -7
$2 \times 2(.408) = 1.632 > 1$, " 1 " No. -8, etc.

This result checks with that shown in detail above. It can also be shown that if a decimal repeats itself in the decimal system, it also repeats itself in the binary system. This method is shown more compactly below:

0.147	
0.294	.0
0.588	0
1.176	1
0.352	0
0.704	0
1.408	1
0.816	0
1.632	1
1.264	1
0.528	0
1.056	1
0.112	0

III. CONVERSION OF BINARY NUMBERS TO THE DECIMAL SYSTEM

Given a number in the binary system, it is always a simple matter to convert it to the decimal system. The converted number is simply the sum of the powers of 2 whose presence in the given number is indicated by 1's in the corresponding binary places.

Binary Place	5	4	3	2	1	0.	-1	-2	-3	-4	Decimal Equivalent
	1	0	1	1	0	1.	0	1	0	1	$1x2^5 + 1x2^3 + 1x2^2 + 1x2^0 + 1x2^{-2} + 1x2^{-4} =$
											$32 + 8 + 4 + 1 + .25 + .0625 =$
											45.3125

The coefficients of the other powers of 2 are zero, so they do not contribute to the converted number.

IV. ARITHMETIC

a. Addition

Since 1 is the largest digit in the binary system, it is evident that any sum larger than 1 must be represented with the aid of carryovers. Therefore, no matter how many 1's are added up in one column, the result under that column must be a 0 or a 1; the rest of the sum is carried over in its binary notation and set up at the head of the adjacent columns to the left as carryover figures. Thus, if a sum of 1's in a column adds up to 6 (which is 110 in the binary notation) a 0 is put at the bottom of the column and the two 1's are put at the head of adjacent columns to the left as carryovers. This is the same as adding the 1's in binary fashion at each step of the columnar addition.

(1 + 1 = 10; 10 + 1 = 11; 11 + 1 = 100; 100 + 1 = 101; 101 + 1 = 110 = 6)

e.g.	1	1	10	2	11	3	111	7	111011	59
	1	1	1	1	1	1	101	5	110111	55
	10	2	11	3	100	4	1100	12	1110010	114

(The small numbers above the examples are carry-over figures put in for ease in following the addition procedure.)

					or, more easily:				
e.g.	111	7	111	7	111	7	111	7	Addends
	011	3	110	6	011	3	110	6	
	101	5	011	3	1010	10	1101	13	
	001	1	111	7	101	5	011	3	
	100	4	010	2	1111	15	10000	16	
	111	7	100	4	001	1	111	7	
	11011	27	11101	29	10000	16	10111	23	
					100	4	010	2	
					10100	20	11001	25	
					111	7	100	4	
					11011	27	11101	29	Sum

b. Subtraction

Subtraction is based on the following rules:

0 from 1 always gives 1, and 1 from 0 always gives 1, but the latter requires "borrowing" from the first column to the left. A 1 in the first column to the left is reduced to 0 by borrowing; a 0 in the first column to the left is reduced to 1, causing the digit in the second column over to be reduced, etc.

10	2	100	4	1000	8	1110	14	1001010	74	Minuend
-1	-1	-1	-1	-1	-1	-1	-1	-0111101	-61	Subtrahend
1	1	011	3	0111	7	1101	13	0001101	13	Remainder

The small numbers show how the digits are changed by borrowing. See also <u>Subtraction</u> under "Complements".

c. Multiplication

Multiplication in the binary system is done exactly as in the decimal system and is based on the multiplication table 0 x 1 = 0, 1 x 1 = 1, 0 x 0 = 0.

```
        101011     or      101011      43 = 2^5 + 2^3 + 2^1 + 2^0       Multiplicand
        101110             101110      46 = 2^5 + 2^3 + 2^2 + 2^1       Multiplier
        000000            1010110     258
       101011            101011      172
      101011            100000010    1978                               Product
     101011             101011
    000000             1001011010
   101011             1010110
   11110111010        11110111010    1978 = 2^10 + 2^9 + 2^8 + 2^7 + 2^5 + 2^4 + 2^3 + 2^1
```

d. Division

Division in the binary system is carried out exactly as in the decimal system.

$1 \overline{)\,1} = 1 \qquad 1 \overline{)\,0} = 0$

```
                                                                Check on Quotient
            0111.1100100001011001          7.782608
   10111) 10110011.0000000000000000    23) 179.000000     2^2 + 2^1 + 2^0   = 7.
           10111                            161                      2^-1   = .5
           0101011                           180                     2^-2   = .25
             10111                           161                     2^-5   = .031250
            0101001                           190                    2^-10  = .000976
              10111                           184                    2^-12  = .000244
             0100100                           60                    2^-13  = .000122
               10111                           46                    2^-16  = .000015
              0011010                          140                             --------
                10111                          138                             7.782607
               00011000                         200
                  10111                         184
                 0000100000                      16
                     110111
                     00100100
                        10111
                       0011010
                         10111
                            11000
                            10111
                             0001
```

It should be noted that in order to get the decimal equivalent of the binary quotient to equal the decimal quotient to 5 decimal places, the binary division had to be carried to 16 binary places.

e. Complements

The ordinary complement of a number in the decimal system is obtained by subtracting the number from the next higher power of 10: e.g. complement of 18 = 100 - 18 = 82. The ordinary complement of a number in the binary system is obtained by subtracting the number from the next higher power of 2: e.g. complement of 5 = $2^3 - 5 = 8 - 5 = 3$. It can be shown that the ordinary complement of a power of 2 is that power of 2 itself. See Example 3.

Another kind of complement of a number is obtained by subtracting the number from any higher power of 2. Notice its use under "Complements, (Subtraction Using Complements).

	(1)	(2)	(3)
Given:	100101	101010	100000

To Find: Binary Complements

Method: Subtract from next higher power of 2.

(1)	1000000	64	(2)	1000000	64	(3)	1000000	64
	-100101	-37		-101010	-42		-100000	-32
	0011011	27		0010110	22		0100000	32

Another method for finding the ordinary complement of a number in the binary system is to interchange all 0's and 1's and add 1.

	(1)	(2)	(3)
Given:	100101	101010	100000

To Find: Binary Complements

Method: Interchange 0's and 1's and add 1.

(1)	No.	100101	number
		011010	interchange 0's and 1's
		1	add 1
		011011	complement
(2)	No.	101010	
		010101	"
		1	
		010110	
(3)	No.	100000	
		011111	"
		1	
		100000	

These results check with those above.

f. Subtraction Using Complements

Instead of subtracting one number from another, it is possible to take a complement of the subtrahend and add that complement to the minuend, provided the power of 2 which was added to the subtrahend in order to get a complement is subtracted from the answer. Practically, subtracting out the added power of 2 means dropping the 1 in the last binary place on the left, if the power of 2 used in getting the complement is greater than that contained in either number. If not, then the power of 2 must be subtracted out by the usual subtraction method.

Regular Subtraction

1011100			Number		-10011	
-001101	Number		Complement	=	01101	(from 2^6)
1001001	Answer		"	=	1101101	(from 2^8)
			"	=	1111101101	(from 2^{11})

Subtraction by Addition of Complements

1011100		1011100		1011100	
01101	Complement	1101101		1111101101	
1101001	Sum	11001001		10001001001	
-100000	2^6	-10000000	2^8	-10000000000	2^{11}
1001001	Answer	01001001		00001001001	

Notice that in the two examples on the right, dropping the last 1 on the left in the sum gives the same result as subtracting out the power of 2 added to get a complement, because the power of 2 added was greater than that contained in either number.

Signed: M. F. Mann
Margaret Florencourt Mann

Approved: JWF
J. W. Forrester

MFM:has:cm