ELECTRONICS PAPER II

NUMBER SYSTEMS

BY

SUBADEVI SUBRAMANIAN

12th Electronics
Suba Subramanian
2/3/21

Number Systems

Decimal number system

This system uses 10 digits from 0 to 9. It also has a place-value characteristic. Consider the number 2472. This decimal number can be expressed as $2\times10^3 + 4\times10^2 + 7\times10^1 + 2\times10^0$. ie each digit is assigned a power of ten according to its position. This is called as positional weighting of a number. The number 10, whose powers are considered, is called the base or radix of the decimal number system. Thus base or radix of a number system simply indicates the number of digits that system uses.

Binary number system

This system uses only 2 digits 0 and 1. It is therefore said to have a radix of 2. Each binary digit is called a bit. A number arranged using these bits is called a binary number.

Consider the binary number 1001.11. Each digit position has a weight or value. Since only 2 digits are used, the weights are powers of 2 as shown below.

$$\begin{array}{cccccccc} 1 & 0 & 0 & 1 & . & 1 & 1 \\ 2^3 & 2^2 & 2^1 & 2^0 & & 2^{-1} & 2^{-2} \end{array}$$

Nibble – A group of 4 bits.

Byte - A group of 8 bits

Binary to decimal conversion

This conversion is performed in a very simple way as follows:

(1) Write the given binary number.
(2) Directly under the binary number, write the weight of each bit.
(3) Cancel the weights, where the bit is zero.
(4) Add the remaining weights to get the decimal equivalent.

For instance, binary 1101.11 converts to decimal as follows.

$$\begin{array}{cccccccc} 1 & 1 & 0 & 1 & . & 1 & 1 \\ 2^3 & 2^2 & \cancel{2^1} & 2^0 & & 2^{-1} & 2^{-2} \end{array}$$

$$2^3 + 2^2 + 2^0 + 2^{-1} + 2^{-2} = 8+4+1+0.5+0.25$$
$$= 13.75$$
$$\therefore (1101.11)_2 = (13.75)_{10}$$

Note :- A subscript attached to a number indicates the base of the number.

Decimal to binary Conversion

The method used for converting a decimal integer into binary is called as double-dabble method and its procedure is as follows:

(1) Divide the given number by 2.
(2) Write down the quotient directly below the given number and remainder on right hand side.
(3) Divide the quotient obtained by 2, again writing the new quotient below and remainder on the right side.
(4) Continue this division process till you arrive at a quotient of 0.
(5) The remainders taken in the reverse order from bottom to top, gives the binary equivalent.

eg. In order to convert decimal 13 to its binary form, we proceed as follows:

2	13	
	6	1 ↑
	3	0
	1	1
	0	1

$(13)_{10} = (1101)_2$

For the conversion of decimal fractions into binary fractions, the procedure is as follows:
(1) Multiply the decimal fraction by 2.
(2) The integer from the multiplication is recorded as a carry.
(3) The remaining fraction is again multiplied by 2, writing the integer as a carry.

(4) As the process is unending, it can be continued till the answer is sufficiently accurate. ie upto say 3 places.

(5) The carries obtained by this procedure, taken from top to bottom, gives the binary equivalent.

eg. in order to convert 0.4 into binary, we proceed as follows:

$0.4 \times 2 = 0.8$ with a carry of 0
$0.8 \times 2 = 1.6 = 0.6$ with a carry of 1
$0.6 \times 2 = 1.2 = 0.2$ with a carry of 1
$0.2 \times 2 = 0.4$ with a carry of 0

$$(0.4)_{10} = (.0110)_2$$

For the conversion of the decimal number 10.5 to binary, there are two processes involved. The integer part of the number ie 10 is converted to binary using double dabble method and the fractional part 0.5 is converted to binary through the eated multiplication process. This is shown below.

2	10	
	5	0
	2	1
	1	0
	0	1

$$(10)_{10} = (1010)_2$$

$0.5 \times 2 = 1.0 = 0$ with a carry 1

$0 \times 2 = 0$ with a carry 0

$\therefore (0.5)_{10} = (0.10)_2$

$\therefore (10.5)_{10} = (1010.10)_2$

Hexadecimal number system

This system uses 16 characters, digits from 0 to 9 and alphabets A to F. Hence base of this system is 16. Each digit position corresponds to a power of 16. The weights of the digits are

$\ldots\ 16^2\ 16^1\ 16^0\ .\ 16^{-1}\ 16^{-2}\ \ldots$

Hexadecimal point

Note that here 'A' in hexadecimal means 10 in decimal. Similarly, B=11, C=12, D=13, E=14 and F=15.

Hexadecimal to decimal conversion

To find the decimal equivalent of a hexadecimal number, multiply each digit by its weight and add the resulting products.

eg. 2B9.1

$= 2 \times 16^2 + B \times 16^1 + 9 \times 16^0 + 1 \times 16^{-1}$

$= 512 + 176 + 9 + 0.0625$

$= 697.0625$

$\therefore (2B9.1)_{16} = (697.0625)_{10}$

Decimal to hexadecimal conversion

To perform this conversion, divide the decimal integer by 16, writing the remainder on the right side and quotient below the integer. Divide the quotient by 16, recording the remainder on the right side and new quotient below. Continue dividing the quotient by 16 till a quotient equal to 0 is obtained. The remainders are then read from bottom to top to get the hexadecimal equivalent. When the remainder exceeds 9, it is replaced by an equivalent hexa digit.

To convert decimal fractions into hexadecimal, multiply it by 16. Record the integer obtained from multiplication as a carry. The remaining fraction is again multiplied by 16. The carries taken from top to bottom, gives the hexadecimal equivalent.

eg. $(423.7)_{10}$

16	423	
	26	7
	1	10 = A
	0	1 ↑

$\therefore (423)_{10} = (1A7)_{16}$

$0.7 \times 16 = 11.2 = 0.2$ with a carry $11 = B$ ↓

$0.2 \times 16 = 3.2 = 0.2$ with a carry 3

$0.2 \times 16 = 3.2 = 0.2$ with a carry 3

$\therefore (0.7)_{10} = (.B33)_{16}$

$\therefore (423.7)_{10} = (1A7.B33)_{16}$

Hexadecimal to binary conversion :—

For converting a hexadecimal number into binary, replace each hexa digit by its 4-bit binary equivalent.

eg $(9AF.B)_{16}$

9	A	F	.	B
1001	1010	1111	.	1011

$\therefore (9AF.B)_{16} = (100110101111.1011)_2$

Binary to hexadecimal conversion :—

To find the hexadecimal equivalent of a given binary number, we group 4 bits at a time, starting from the binary point and working both ways. When necessary, 0's are added to complete outside groups. Then each group of 4 is converted into its hexa equivalent.

eg. $(11110.01011)_2$

$\underbrace{0001}_{1}\ \underbrace{1110}_{E}\ .\ \underbrace{0101}_{5}\ \underbrace{1000}_{8}$

$\therefore (11110.01011)_2 = (1E.58)_{16}$

Note :-

The decimal number system is not very useful in digital system. This is because it is very difficult to design electronic equipment so that it can work with 10 different voltage levels. (each one representing one decimal character) On the other hand, it is very easy to design simple, accurate electronic circuits that operate with only 2 voltage levels. Also many electronic components are bistable in nature ie they are in either of the 2 states ON, OFF. These 2 possible states can be indicated by 0, 1. For these above mentioned reasons, almost all digital systems use the binary number system as the basic number system of its operations, although other systems are often used along with binary.

Binary Addition

The procedure for addition is the same as that for decimal numbers. The rules are as follows:

$0+0=0$

$0+1=1$

$1+0=1$

$1+1=10$ ie 0, with carry 1

$1+1+1=11$ ie 1, with carry 1

eg. 0010 + 1010

```
  0 0 1 0
  1 0 1 0
---------
  1 1 0 0
---------
```

Addition proceeds column by column. When the result is 10 or 11, 1 is carried over and added to the left column.

Binary Subtraction

Direct Method

The rules for binary subtraction are as follows:

$0-0=0$

$1-1=0$

$1-0=1$

$10-1=1$

eg. 100 − 001

```
  1 0 0
- 0 0 1
-------
  0 1 1
-------
```

1st column: 10 (after borrow) − 1 = 1

2nd column: 10 (after borrow − 1 (previous borrow) = 1

3rd column: 1 − 1 (previous borrow) = 0

1's complement and 2's complement :-

1's complement of a binary number is the number that results when we change each 0 to a 1 and each 1 to a 0. In other words, 1's complement of 100 is 011.

2's complement is the binary number that results when we add 1 to the 1's complement. ie 2's complement = 1's complement + 1

eg. 2's complement of 1110 = 0010.

Binary subtraction by 1's complement

The following steps are to be followed.

(1) Write the first number from which the subtraction is to be done.
(2) Take 1's complement of the number to be subtracted. Before doing this, see that the number has the same number of digits as the first. If not, add appropriate zeros.
(3) Add the 1's complement to the first number.
(4) If there is a carry, the final answer of subtraction is positive and so add the carry to the remaining digits. If carry is absent, final answer is negative. Take 1's complement of the answer and put a negative sign to it.

eg. (i) 1101 − 10

ie 1101 − 0010

1's complement of 0010 is 1101

$$
\begin{array}{r}
\therefore \quad 1101 \\
+\ 1101 \\
\hline
\text{Carry} \rightarrow \boxed{1}\,1010 \\
+1 \\
\hline
1011 \\
\hline
\end{array}
$$

$\therefore 1101 - 10 = 1011$

(ii) $1011 - 11111$

ie $01011 - 11111$

1's complement of 11111 is 00000

$$
\begin{array}{r}
\therefore \quad 01011 \\
+\ 00000 \\
\hline
01011 \\
\hline
\end{array}
$$

Carry absent. 1's complement of 01011 is 10100.

$\therefore 1011 - 11111$ is -10100.

Binary subtraction by 2's complement

To perform binary subtraction by 2's complement method, the following steps are to be followed:

(1) Write the first number from which the subtraction is to be done.

(2) Take 2's complement of the number to be subtracted.

Before doing this, see that the number has the same number of digits as the first. If not, add appropriate zeros on the left hand side.

(3) Add 2's complement to the first number.

(4) If there is a carry, final answer is positive and to get the answer, simply disregard or neglect the carry.

(5) If there is no carry, final answer is negative. Take 2's complement of the answer obtained and put a negative sign to it.

eg. (i) 101 − 11

ie 101 − 011

2's complement of 011 is 101

$$\therefore \begin{array}{r} 101 \\ +\ 101 \\ \hline \text{Carry} \rightarrow \boxed{1}010 \end{array}$$

Neglecting carry, 101 − 11 = 010.

(ii) 101 − 1111

ie 0101 − 1111

2's complement of 1111 is 0001

$$\therefore \begin{array}{r} 0101 \\ +\ 0001 \\ \hline 0110 \\ \hline \end{array}$$

No carry. 2's complement of 0110 is 1010.

∴ 101 − 1111 = −1010

BCD code (Binary Coded Decimal)

Any decimal number can be represented by an equivalent binary number. The group of 0's and 1's in the binary number can be thought of as a code representing the decimal number. When a decimal number is represented by its equivalent binary number, we call it straight binary coding.

Digital systems all use some form of binary numbers for their internal operation, but the external world is decimal in nature. This means that conversions between the decimal and binary systems are being performed often. The conversions between decimal and binary can become long and complicated for large numbers. For this reason, another means of encoding decimal numbers is sometimes used. If each digit of a decimal number is represented by its binary equivalent, this produces a code called BCD code and since a decimal digit can be as large as 9, ~~only~~ 4 bits are required to code each digit.

According to the BCD code, each digit of the decimal number is represented by a 4-bit binary equivalent. Clearly, only the

4-bit binary numbers from 0000 through 1001 are used.

eg. $(94 \cdot 3)_{10}$

$$\begin{array}{ccc} 9 & 4 & \cdot\ 3 \\ | & | & | \\ 1001 & 0100 & 0011 \end{array}$$

$\therefore (94 \cdot 3)_{10} = (100101000011)_{BCD}$

A BCD number is not the same as a straight binary number. A straight binary code takes the complete decimal number and represents it in binary whereas the BCD code converts each decimal digit to binary individually. Though the length of the BCD number increases, it becomes easy to convert the decimal number.

Advantage — Easy to convert to and from BCD numbers.

Disadvantage — The rules of binary addition do not apply to BCD numbers.

The ASCII code

To get information into and out of a computer, we need to use some kind of alphanumeric code (one for letters, numbers and other symbols). At one time, manufacturers used their own alphanumeric codes, which led to all kinds of confusion. Eventually, industry settled on an input-output code known as the AMERCIAN STANDARD CODE FOR INFORMATION INTERCHANGE (abbreviated ASCII) This code allows manufacturers to standardize computer hardware such as keyboards, printers.

The ASCII code is a 7-bit code whose format is $X_6X_5X_4X_3X_2X_1X_0$ where each X is a 0 or a 1. The table below shows the ASCII code for alphabets, digits and some symbols. Read the table similar to the way you read a graph.

	$X_6X_5X_4$					
$X_3X_2X_1X_0$	010	011	100	101	110	111
0 0 0 0	SP	0	@	P		p
0 0 0 1	!	1	A	Q	a	q
0 0 1 0	"	2	B	R	b	r
0 0 1 1	#	3	C	S	c	s
0 1 0 0	$	4	D	T	d	t
0 1 0 1	%	5	E	U	e	u
0 1 1 0	&	6	F	V	f	v
0 1 1 1	'	7	G	W	g	w
1 0 0 0	(	8	H	X	h	x
1 0 0 1	)	9	I	Y	i	y
1 0 1 0	*	:	J	Z	j	z
1 0 1 1	+	;	k		k	
1 1 0 0	,	<	L		l	
1 1 0 1	–	=	M		m	
1 1 1 0	.	>	N		n	
1 1 1 1	/	?	O		o	

Do not memorise

For instance, the letter 'A' has an $X_6 X_5 X_4$ of 100 and an $X_3 X_2 X_1 X_0$ of 0001
∴ the ASCII code is 1000001
The letter 'a' is coded as 1100001
SP in the table stands for space (blank)
Hitting the space bar of an ASCII keyboard sends this into a microcomputer:
0100000

Decimal number	4-bit binary equivalent.
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

EBCDIC code :- Another code (alphanumeric) used in IBM equipment is the Extended Binary Coded Decimal Information Code. It uses eight bits for each character