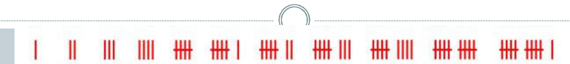
Digital Logic Design

DR YASIR HASHIM

Number Systems

Contents

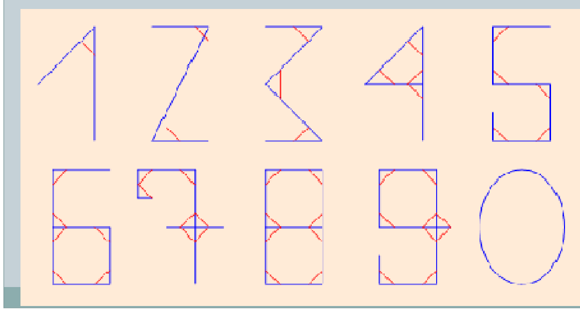


Roman Number:

- I, II, III, IV, V, VI, VII, VIII, IX, X
- L = 50, C = 100, M = 1000, D=500
- Example: DCCCXC = 890

Arabic - Indian

- Number of Angles



The concept of 0

- 1, 2, 3, 4, 5, 6, 7, 8, 9, shift 10
- zero: empty
- 11, 12 ....
- We use 10 base because the first calculus instrument is our 10 fingers.

Structure of a number

$(7\ 4\ 2\ .\ 4\ \ 5)_{10}$ ⟶ Decimal number

This can be written as:

700 + 40 + 2 + 0.4 + 0.05

positional value

$7 \times 10^{2} + 4 \times 10^{1} + 2 \times 10^{0} + 4 \times 10^{-1} + 5 \times 10^{-2}$

coefficient

base or radix

Commonly used number systems

- Binary: base 2
- Octal: base 8
- Decimal: base 10
- Hexadecimal: base 16

Binary Numbers

- The binary number system is used. Binary has a radix of two and uses the digits 0 and 1 to represent quantities.
- The column weights of binary numbers are powers of two that increase from right to left beginning with 2^0 =1: …2^5 2^4 2^3 2^2 2^1 2^0.
- For fractional binary numbers, the column weights are negative powers of two that decrease from left to right: 2^2 2^1 2^0. 2^{-1} 2^{-2} 2^{-3} 2^{-4} …

- 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

Converting Binary to decimal

$(1\,0\,1\ .\,0\,1)_2 \xrightarrow[\text{convert}]{}$ Decimal number

$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$

$= 4 + 0 + 1 + 0 + 0.25$

$= (5.25)_{10}$

Converting decimal to Binary

- You can convert a decimal whole number to binary by reversing the procedure. Write the decimal weight of each column and place 1's in the columns that sum to the decimal number.

Example: Convert the decimal number 49 to binary.

- The column weights double in each position to the right. Write down column weights until the last number is larger than the one you want to convert.

2^6	2^5	2^4	2^3	2^2	2^1	2^0.
64	32	16	8	4	2	1.
0	1	1	0	0	0	1.

Binary Conversions

You can convert decimal to any other base by repeatedly dividing by the base. For binary, repeatedly divide by 2:

Example Convert the decimal number 49 to binary by repeatedly dividing by 2.

You can do this by "reverse division" and the answer will read from left to right. Put quotients to the left and remainders on top.

Answer: 1 1 0 0 0 1 ← remainder
0 1 3 6 12 24 49) 2

Continue until the last quotient is 0

Quotient

Decimal number

base

Binary fraction Conversions

You can convert a decimal fraction to binary by repeatedly multiplying the fractional results of successive multiplications by 2. The carries form the binary number.

Convert the decimal fraction 0.188 to binary by repeatedly multiplying the fractional results by 2.

MSB

0.188 x 2 = 0.376	carry = 0
0.376 x 2 = 0.752	carry = 0
0.752 x 2 = 1.504	carry = 1
0.504 x 2 = 1.008	carry = 1
0.008 x 2 = 0.016	carry = 0

Answer = .00110 (for five significant digits)

Binary Arithmetic

Binary Addition

The rules for binary addition are

0 + 0 = 0	Sum = 0, carry = 0
0 + 1 = 1	Sum = 1, carry = 0
1 + 0 = 1	Sum = 1, carry = 0
1 + 1 = 10	Sum = 0, carry = 1

Example Add the binary numbers 00111 and 10101 and show the equivalent decimal addition.

```
 0 1 1 1
 00111      7
 10101     21
 -----     --
 11100  =  28
```

Binary Subtraction

The rules for binary subtraction are

$0 - 0 = 0$
$1 - 1 = 0$
$1 - 0 = 1$
$0 - 1$: We borrow 1: $10 - 1 = 1$

Example Subtract the binary number 00111 from 10101 and show the equivalent decimal subtraction.

$$\begin{array}{rr} \scriptstyle 1\,1\,1 & \\ 10101 & 21 \\ \underline{00111} & \underline{7} \\ 01110 & = 14 \end{array}$$

Binary Multiplication

The four basic rules for multiplying bits are as follows:

$0 \times 0 = 0$
$0 \times 1 = 0$
$1 \times 0 = 0$
$1 \times 1 = 1$

Example Multiply the binary number 111 from 101 and show the equivalent decimal subtraction.

$$\begin{array}{lrr} & 111 & 7 \\ & \underline{\times 101} & \underline{\times 5} \\ \text{Partial products} \left\{ \right. & 111 & 35 \\ & 000 & \\ & \underline{+111} & \\ & \mathbf{100011} & \end{array}$$

Binary Division

Division in binary follows the same procedure as division in decimal

$0 \div 1 = 0$
$1 \div 1 = 1$

Can't divide by 0

Example $110 \div 11$

$$\begin{array}{rr} \mathbf{10} & 2 \\ 11\overline{)110} & 3\overline{)6} \\ \underline{11} & \underline{6} \\ 000 & 0 \end{array}$$

Complements

- Complements are used for simplifying the subtraction operation and for logical manipulations and representation the negative numbers.
- There are two types of complements for each base-*r* system:
- The r's complement
- The (r-1)'s complement

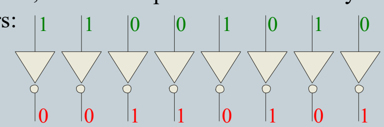
- In case of a binary number
- The (r-1)'s complement = 1's complement
- The r's complement = 2's complement

1's Complement

The 1's complement of a binary number is just the inverse of the digits. To form the 1's complement, change all 0's to 1's and all 1's to 0's.

For example, the 1's complement of 11001010 is 00110101

In digital circuits, the 1's complement is formed by using inverters:



Easy rules for 2's complement:

System 1
- Leave all least significant zeros and the first nonzero digit unchanged
- Replace remaining 1's by 0's and 0's by 1's

System 2
- Take 1's complement of the number
- Add 1 with the LSB (Least significant bit)

2's complement

System 1

0 1 0 0 1 0 1 0 Binary number

1 0 1 1 0 1 1 0 2's complement

System 2

0 1 0 0 1 0 1 0 Binary number

1 0 1 1 0 1 0 1 1's complement

+ 1

1 0 1 1 0 1 1 0 2's complement

2's Complement

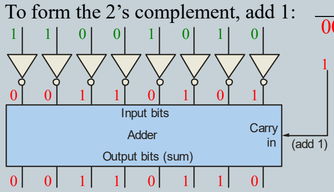
The 2's complement of a binary number is found by adding 1 to the LSB of the 1's complement.

Recall that the 1's complement of 11001010 is

00110101 (1's complement)

To form the 2's complement, add 1: +1

00110110 (2's complement)



9's & 10's complement

Let us take a decimal number 456,
9's complement of this number will be

999
(-) 456
543

10's complement of this no

543
(+)1
544

Decimal digit	9s complement
0	9
1	8
2	7
3	6
4	5
5	4
6	3
7	2
8	1
9	0

Signed Binary Numbers

- There are several ways to represent signed binary numbers. In all cases, the MSB in a signed number is the sign bit, that tells you if the number is positive or negative.
- Computers use a modified 2's complement for signed numbers. Positive numbers are stored in *true* form (with a 0 for the sign bit) and negative numbers are stored in *complement* form (with a 1 for the sign bit).

Signed Binary Numbers

For example, the positive number 58 is written using **8-bits** as 00111010 (true form).

Sign bit — Magnitude bits

Negative numbers are written as the 2's complement of the corresponding positive number.

The negative number –58 is written as:

–58 = 11000110 (complement form)

Sign bit — Magnitude bits

Example

- Assuming that the sign bit = –128, show that 11000110 = –58 as a 2's complement signed number:

Column weights:	–128	64	32	16	8	4	2	1.
	1	1	0	0	0	1	1	0
	–128	+64				+4	+2	= –58

Caution

Make sure to specify whether a binary number is signed or unsigned

For 5 bit:

11001 → -7 (if considered as signed)

11001 → 25 (if considered as unsigned)

Example: C++ Variable types in bytes

Type	Typical Bit Width	Typical Range
char	1byte	-127 to 127 or 0 to 255
int	4bytes	-2147483648 to 2147483647
unsigned int	4bytes	0 to 4294967295
signed int	4bytes	-2147483648 to 2147483647

Octal Numbers

Octal uses eight characters the numbers 0 through 7 to represent numbers. There is no 8 or 9 character in octal.

Binary number can easily be converted to octal by grouping bits 3 at a time and writing the equivalent octal character for each group.

Decimal	Octal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	10	1000
9	11	1001
10	12	1010
11	13	1011
12	14	1100
13	15	1101
14	16	1110
15	17	1111

Example

Express 1 001 011 000 001 110_2 in octal:

Group the binary number by 3-bits starting from the right. Thus, 113016_8

Converting Octal to decimal

$(6\ 0\ 3\ .\ 4)_8 \xrightarrow[\text{convert}]{}$ Decimal number

$6 \times 8^2 + 0 \times 8^1 + 3 \times 8^0 + 4 \times 8^{-1}$

$= 384 + 0 + 3 + 0.5 = (387.5)_{10}$

Hexadecimal Numbers

Hexadecimal uses sixteen characters to represent numbers: the numbers 0 through 9 and the alphabetic characters A through F.

Large binary number can easily be converted to hexadecimal by grouping bits 4 at a time and writing the equivalent hexadecimal character.

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Example

Express 1001 0110 0000 1110_2 in hexadecimal:

Group the binary number by 4-bits starting from the right. Thus, 960E

Why?

$2^3 = 8$ which is the base of Octal number system

So, each octal number can be represented by 3 binary digits

Such as $(5)_8 = (101)_2$ and $(7)_8 = (111)_2$

$2^4 = 16$ which is the base of Hex number system

So, each hex number can be represented by 4 binary digits

Such as $(9)_{16} = (1001)_2$ and $(A)_{16} = (1010)_2$

Converting is easier

Octal to Binary

$(6 \quad 7 \quad 3 \;.\; 1 \quad 2 \quad 4)_8$ to binary

$(110 \quad 111 \quad 011 \;.\; 001 \quad 010 \quad 100)_2$

$(673\,.\,124)_8 = (110\ 111\ 011\ .\ 001\ 010\ 100)_2$

Hexadecimal to Binary

$(3 \quad 0 \quad 6 \;.\; D)_{16}$ to binary

$(0011 \quad 0000 \quad 0110 \;.\; 1101)_2$

$(306.D)_{16} = (001100000110.1101)_2$

Binary to octal

$(10 \quad 110 \quad 001 \quad 101 \;.\; 111 \quad 100 \quad 000)_2$ to octal

$(2 \quad 6 \quad 1 \quad 5 \;.\; 7 \quad 4 \quad 0)_8$

$(10\ 110\ 001101\ .\ 111100000)_2 = (2\ 6\ 1\ 5\ .\ 7\ 4\ 0)_8$

Binary to Hexadecimal

$(10\ 1100\ 0110\ 1011\ .\ 1111\ 0010)_2$ to Hexadecimal

$(2\quad C\quad 6\quad B\ .\quad F\quad 2)_{16}$

$(10\ 1100\ 0110\ 1011\ .\ 1111\ 0010)_2 = (2C6B\ .\ F2)_{16}$

Fraction decimal to Binary

$(0.6875)_{10} \xrightarrow{\text{convert}}$ binary

	Integer		fraction	coefficient
0.6875 X 2 =	1	+	0.3750	$a_{-1} = 1$
0.3750 X 2 =	0	+	0.7500	$a_{-2} = 0$
0.7500 X 2 =	1	+	0.5000	$a_{-3} = 1$
0.5000 X 2 =	1	+	0.0000	$a_{-4} = 1$

$(0.6875)_{10} = (0\ .\ a_{-1}\ a_{-2}\ a_{-3}\ a_{-4}) = (0.1011)_2$

Fraction decimal to Octal

$(0.513)_{10} \xrightarrow{\text{convert}}$ octal

	Integer		fraction	coefficient
0.513 X 8 =	4	+	0.104	$a_{-1} = 4$
0.104 X 8 =	0	+	0.832	$a_{-2} = 0$
0.832 X 8 =	6	+	0.656	$a_{-3} = 6$
0.656 X 8 =	5	+	0.248	$a_{-4} = 5$

$(0.513)_{10} = (0\ .\ a_{-1}\ a_{-2}\ a_{-3}\ a_{-4}\ \ldots) = (0.4065\ \ldots)_8$

Binary Coded Decimal (BCD)

Binary coded decimal (BCD) is a weighted code that is commonly used in digital systems when it is necessary to show decimal numbers such as in clock displays.

The table illustrates the difference between straight binary and BCD. BCD represents each decimal digit with a 4-bit code. Notice that the codes 1010 through 1111 are not used in BCD.

Decimal	Binary	BCD
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	00010000
11	1011	00010001
12	1100	00010010
13	1101	00010011
14	1110	00010100
15	1111	00010101

Example

$(185)_{10} = (0001\ 1000\ 0101)_{BCD}$

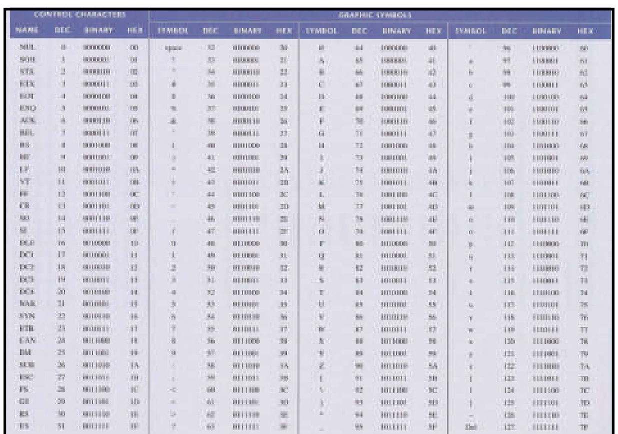
Is the binary equivalent of $(185)_{10}$ will be the same as BCD ?

No

$(185)_{10} = (10111001)_2$

ASCII

- ASCII (The American Standard Code for Information Interchange)is a code for alphanumeric characters and control characters.
- In its original form, ASCII encoded 128 characters and symbols using 7-bits.
- The first 32 characters are control characters, that are based on obsolete teletype requirements, so these characters are generally assigned to other functions in modern usage.



References

1. T. Floyd, “Digital Fundamental”, 10th Ed., USA: PrenticeHall, 2008
2. R.J. Tocci, “Digital Systems: Principles and Applications”, 10th Ed., USA: Prentice-Hall, 2006
3. W. Kleitz, “Digital Electronics: A Practical Approach”, 8th Ed., USA: Prentice-Hall, 2007
4. Begnell and Donovan, “Digital Electronics”, 5th Ed., USA: Delmar Thomson Learning, 2006