PGT104 – Digital Electronics

Part 1 – Introduction to Number Systems

Disclaimer:

- Most of the contents (if not all) are extracted from resources available for Digital Fundamentals 10th Edition

Decimal Numbers

The position of each digit in a weighted number system is assigned a weight based on the **base** or **radix** of the system. The radix of decimal numbers is ten, because only ten symbols (0 through 9) are used to represent any number.

The column weights of decimal numbers are powers of ten that increase from right to left beginning with $10^0 = 1$:

$\ldots 10^5\ 10^4\ 10^3\ 10^2\ 10^1\ 10^0.$

For fractional decimal numbers, the column weights are negative powers of ten that decrease from left to right:

$10^2\ 10^1\ 10^0.\ 10^{-1}\ 10^{-2}\ 10^{-3}\ 10^{-4}\ \ldots$

Decimal Numbers

Decimal numbers can be expressed as the sum of the products of each digit times the column value for that digit. Thus, the number 9240 can be expressed as

$$(9 \times 10^3) + (2 \times 10^2) + (4 \times 10^1) + (0 \times 10^0)$$

or

$$9 \times 1{,}000 + 2 \times 100 + 4 \times 10 + 0 \times 1$$

Example

Express the number 480.52 as the sum of values of each digit.

$$480.52 = (4 \times 10^2) + (8 \times 10^1) + (0 \times 10^0) + (5 \times 10^{-1}) + (2 \times 10^{-2})$$

Binary Numbers

For digital systems, 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 $2^0 = 1$:

$$\ldots 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0.$$

For fractional binary numbers, the column weights are negative powers of two that decrease from left to right:

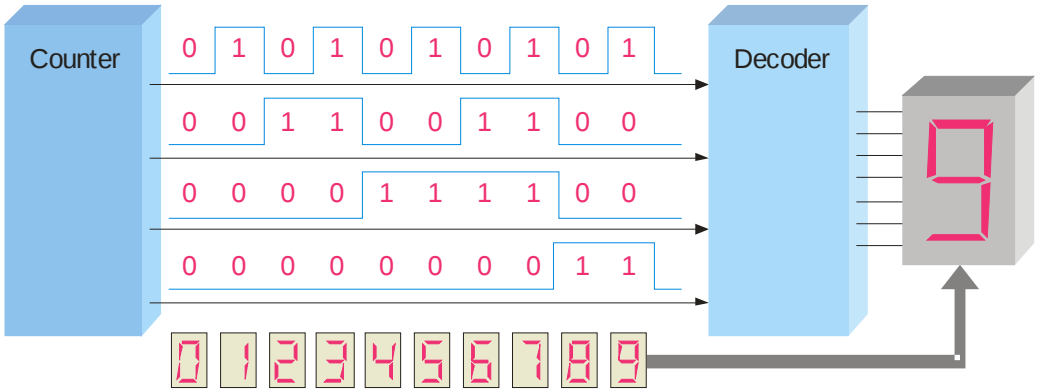
$$2^2\ 2^1\ 2^0.\ 2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}\ \ldots$$

Binary Numbers

A binary counting sequence for numbers from zero to fifteen is shown.

Notice the pattern of zeros and ones in each column.

Digital counters frequently have this same pattern of digits:



Decimal Number	Binary Number
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	1 0 1 0
11	1 0 1 1
12	1 1 0 0
13	1 1 0 1
14	1 1 1 0
15	1 1 1 1

Finite-Bit Representation

- Digital systems have fixed bit-length
  - usually multiple of 8-bit (bytes)
  - limits range of values can be represented
- For unsigned integer,
  - range is from 0 to $2^n - 1$ (*n* is the number of bits)
  - e.g. for a 4-bit integer, the valid range is 0 - 15

Binary Conversions

The decimal equivalent of a binary number can be determined by adding the column values of all of the bits that are 1 and discarding all of the bits that are 0.

Example

Convert the binary number 100101.01 to decimal.

Start by writing the column weights; then add the weights that correspond to each 1 in the number.

2^5	2^4	2^3	2^2	2^1	2^0 .	2^{-1}	2^{-2}
32	16	8	4	2	1 .	½	¼
1	0	0	1	0	1.	0	1

32 +4 +1 +¼ = 37¼

Binary Conversions

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.

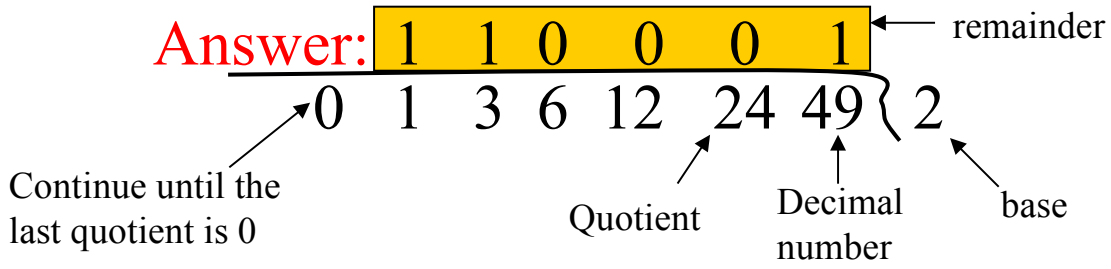
2^6	2^5	2^4	2^3	2^2	2^1	2^0.
64	32	16	8	4	2	1.
0	1	1	0	0	0	1.

Binary Conversions

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

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.



Binary Addition

The rules for binary addition are

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

When an input carry = 1 due to a previous result, the rules are

1 + 0 + 0 = 01	Sum = 1, carry = 0
1 + 0 + 1 = 10	Sum = 0, carry = 1
1 + 1 + 0 = 10	Sum = 0, carry = 1
1 + 1 + 1 = 10	Sum = 1, carry = 1

Binary Addition

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

$$0 - 0 = 0$$
$$1 - 1 = 0$$
$$1 - 0 = 1$$
$$10 - 1 = 1 \quad \text{with a borrow of 1}$$

Example

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

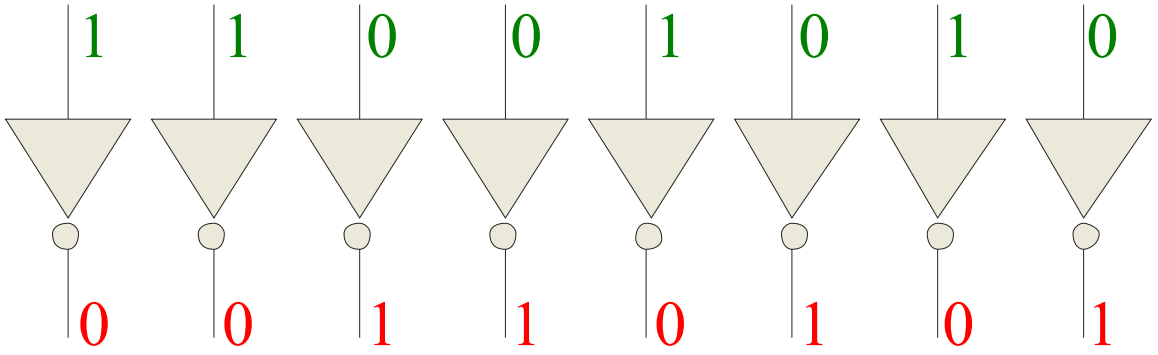
$$\begin{array}{rr} {\scriptstyle 1\ 1\ 1} & \\ 10101 & 21 \\ 00111 & 7 \\ \hline 01110 & = 14 \end{array}$$

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:



2's Complement

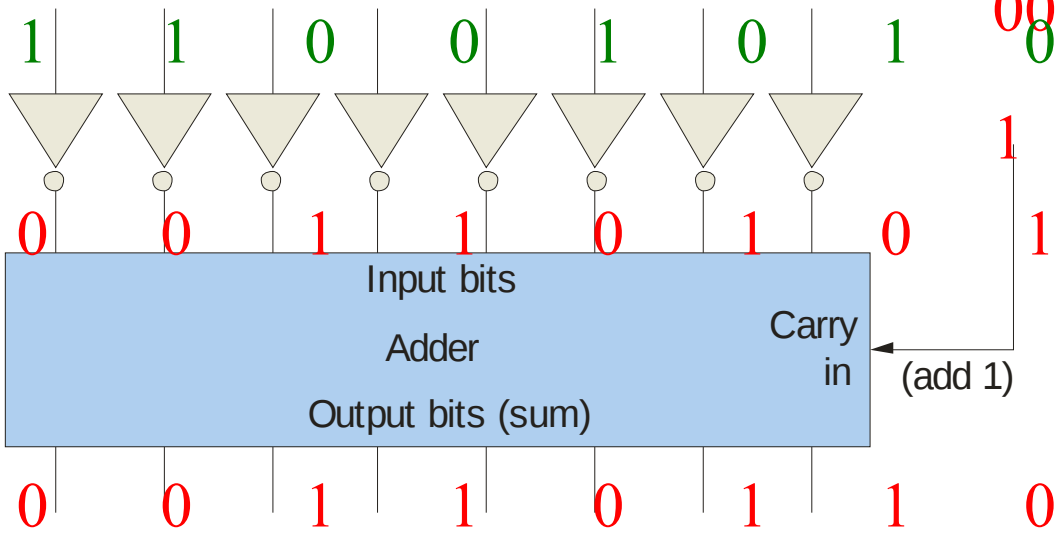
The 2's complement of a binary number is found by adding 1 to the LSB of the 1's complement.

Recall that the 1's complement of 11001010 is 00110101 (1's complement)

To form the 2's complement, add 1:

$$\begin{array}{r} 00110101 \\ +\quad\ 1 \\ \hline 00110110 \end{array}$$

(2's complement)



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).

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

Signed Binary Numbers

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

An easy way to read a signed number that uses this notation is to assign the sign bit a column weight of -128 (for an 8-bit number). Then add the column weights for the 1's.

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

Range for Signed Binary

- Signed binary integer also has limited range
  - for sign-magnitude format, range is from $-(2^{n-1}-1)$ to $(2^{n-1}-1)$
  - e.g. for a 4-bit sign-magnitude integer, the valid range is -7 to 7
- What about 1's complement and 2's complement?

Arithmetic Operations with Signed Numbers

Using the signed number notation with negative numbers in 2's complement form simplifies addition and subtraction of signed numbers.

Rules for **addition**: Add the two signed numbers. Discard any final carries. The result is in signed form.

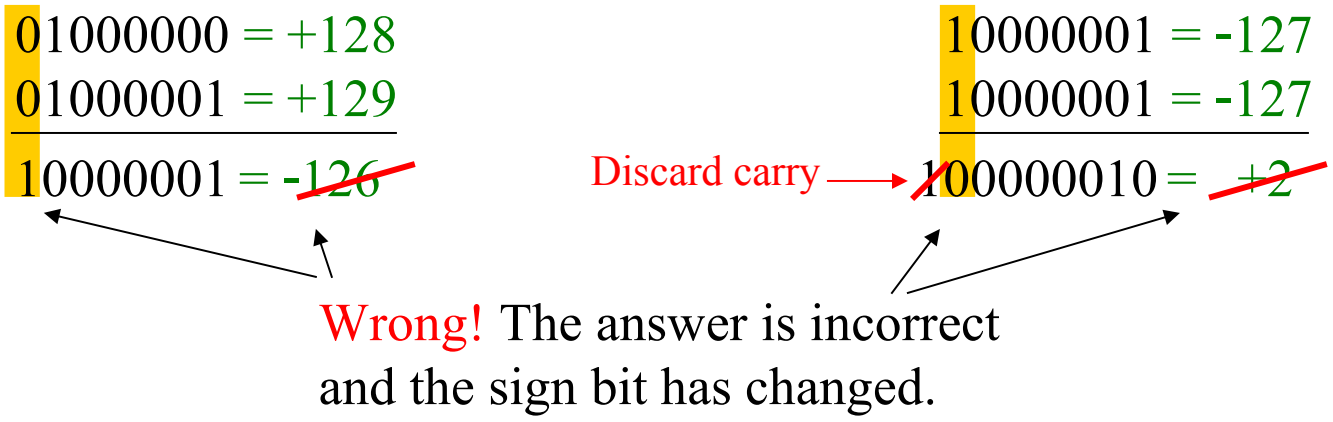
Examples:

00011110 = +30	00001110 = +14	11111111 = -1
00001111 = +15	11101111 = -17	11111000 = -8
00101101 = +45	11111101 = -3	11110111 = -9

Arithmetic Operations with Signed Numbers

Note that if the number of bits required for the answer is exceeded, overflow will occur. This occurs only if both numbers have the same sign. The overflow will be indicated by an incorrect sign bit.

Two examples are:



Arithmetic Operations with Signed Numbers

Rules for **subtraction**: 2's complement the subtrahend and add the numbers. Discard any final carries. The result is in signed form.

Repeat the examples done previously, but subtract:

00011110	(+30)	00001110	(+14)	11111111	(-1)
- 00001111	–(+15)	- 11101111	–(-17)	- 11111000	–(-8)

2's complement subtrahend and add:

00011110 = +30	00001110 = +14	11111111 = -1
11110001 = -15	00010001 = +17	00001000 = +8
100001111 = +15	00011111 = +31	100000111 = +7

Discard carry

Discard carry

Signed Binary Multiplication

- Just like in decimal, use partial product
  - Easier in binary (either sum/add or nothing)
- Only works with positive values
  - Convert if negative (take note of sign for result)
- Result depends on input signs
  - Result is positive if signs are the same
  - Result is negative if signs differ (convert!)

Signed Binary Division

- Simplest method is to subtract
  - Keep track number of subtraction (quotient!)
  - Leftover is remainder
  - Integer results (quotient & remainder)
- Only works with positive values
  - Convert if negative (take note of sign for result)
- Result depends on input signs
  - Result is positive if signs are the same
  - Result is negative if signs differ (convert!)

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.

Example

Express $1001\ 0110\ 0000\ 1110_2$ in hexadecimal:

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

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

Hexadecimal Numbers

Hexadecimal is a weighted number system. The column weights are powers of 16, which increase from right to left.

Column weights { 16^3 16^2 16^1 16^0 : 4096 256 16 1 :

Example

Express $1A2F_{16}$ in decimal.

Start by writing the column weights:

4096 256 16 1

1 A 2 F_{16}

$1(4096) + 10(256) + 2(16) + 15(1) = 6703_{10}$

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

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.

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

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

Octal Numbers

Octal is also a weighted number system. The column weights are powers of 8, which increase from right to left.

Column weights { 8^3 8^2 8^1 8^0 : 512 64 8 1 :

Example

Express 3702_8 in decimal.

Start by writing the column weights:

512 64 8 1

3 7 0 2_8

$3(512) + 7(64) + 0(8) + 2(1) = 1986_{10}$

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

Value and Representation

- Numbering systems are just representations
  - Binary: used by digital systems
  - Decimal: what we are used to
  - Hexadecimal & octal are 'compressed' binary
- Conversion between systems DOES NOT change value!
  - easier to analyze value and/or easier to look at

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

BCD

You can think of BCD in terms of column weights in groups of four bits. For an 8-bit BCD number, the column weights are: 80 40 20 10 8 4 2 1.

Example

What are the column weights for the BCD number 1000 0011 0101 1001?

8000 4000 2000 1000 800 400 200 100 80 40 20 10 8 4 2 1

Note that you could add the column weights where there is a 1 to obtain the decimal number. For this case:

$$8000 + 200 + 100 + 40 + 10 + 8 + 1 = 8359_{10}$$

Quiz

For the binary number 1000, the weight of the column with the 1 is

a. 4

b. 6

c. 8

d. 10

Quiz

The 2's complement of 1000 is

a. 0111

b. 1000

c. 1001

d. 1010

Quiz

The fractional binary number 0.11 has a decimal value of

a. ¼

b. ½

c. ¾

d. none of the above

Quiz

The hexadecimal number 2C has a decimal equivalent value of

a. 14

b. 44

c. 64

d. none of the above

Quiz

When two positive signed numbers are added, the result may be larger that the size of the original numbers, creating overflow. This condition is indicated by

a. a change in the sign bit

b. a carry out of the sign position

c. a zero result

d. smoke

Quiz

The number 1010 in BCD is

a. equal to decimal eight

b. equal to decimal ten

c. equal to decimal twelve

d. invalid