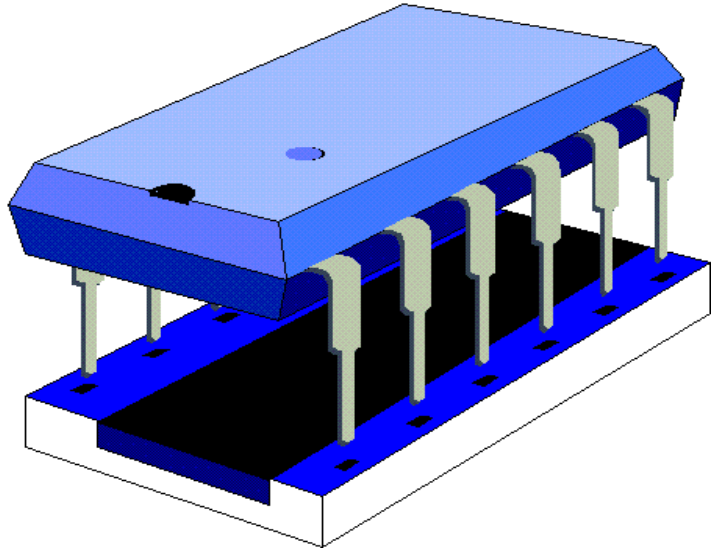
Digital Electronics



Number Systems and Codes Part 2

Grayson Himunzowa.

Objectives

- At the end of this chapter the student should be able to
- Understand different number systems in use.
- Convert one number system to another number system.
- Understand codes used in digital systems.
- Carry out binary arithmetic.
- Negate binary numbers

Binary Arithmetic

- Binary arithmetic is essential part of all the digital computers and many other digital systems.
- Hence it is important to explore the various arithmetic procedures involved.
- These are addition, subtraction, multiplication and division.

Binary arithmetic - Addition

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 BUT WITH CARRY IN (OVERFLOW) TO NEXT (MORE SIGNIFICANT) DIGIT TO LEFT

with carry in	sum	Carry
1 + 0 + 0 =	1	0
1 + 0 + 1 =	0	1
1 + 1 + 0 =	0	1
1 + 1 + 1 =	1	1

Binary Subtraction

- There are four rules for binary subtraction

0 - 0 = 0
1 - 1 = 0
1 - 0 = 1
10 - 1 = 1

Exercise

evaluate

$0 - 1 =$

Binary arithmetic

- Multiplication is similar to decimal multiplication. It is simpler than decimal multiplication because only 0s and 1s are involved. The rules of multiplication are show below:

  0 x 0 = 0

  0 x 1 = 0

  1 x 0 = 0

  1 x 1 = 1

Binary Division

- It is similar to decimal division

  Example

  101010 / 000110 = 111

Prove it by working it out.

Exercise

- What is the binary result of $11100_2 + 10011_2$?
- What is the binary result of $101_2 - 11_2$?
- What is the binary result of $111_2 \times 101_2$?

Hexadecimal addition

- Three rules

1. In any given column of an addition problem, think of the two hexadecimal digits in terms of their decimal values. For instance, $5_{16} = 5_{10}$ and $C_{16} = 12_{10}$.
2. If the sum of these two digits is 15_{10} or less, bring down the corresponding hexadecimal digit.
3. If the sum of these two digits is greater than 15_{10}, bring down the amount of the sum that exceeds 16_{10} and carry a 1 to the next column.

Example

Add the following hexadecimal numbers:

(a) $23_{16} + 16_{16}$ **(b)** $58_{16} + 22_{16}$ **(c)** $2B_{16} + 84_{16}$ **(d)** $DF_{16} + AC_{16}$

(a)

$$\begin{array}{r} 23_{16} \\ +16_{16} \\ \hline 39_{16} \end{array}$$

right column: $3_{16} + 6_{16} = 3_{10} + 6_{10} = 9_{10} = 9_{16}$

left column: $2_{16} + 1_{16} = 2_{10} + 1_{10} = 3_{10} = 3_{16}$

(b)

$$\begin{array}{r} 58_{16} \\ +\ 22_{16} \\ \hline 7A_{16} \end{array}$$

right column: $8_{16} + 2_{16} = 8_{10} + 2_{10} = 10_{10} = A_{16}$

left column: $5_{16} + 2_{16} = 5_{10} + 2_{10} = 7_{10} = 7_{16}$

(c)

$$\begin{array}{r} 2B_{16} \\ +\ 84_{16} \\ \hline AF_{16} \end{array}$$

right column: $B_{16} + 4_{16} = 11_{10} + 4_{10} = 15_{10} = F_{16}$

left column: $2_{16} + 8_{16} = 2_{10} + 8_{10} = 10_{10} = A_{16}$

(d)

$$\begin{array}{r} DF_{16} \\ +\ AC_{16} \\ \hline 18B_{16} \end{array}$$

right column: $F_{16} + C_{16} = 15_{10} + 12_{10} = 27_{10}$

$27_{10} - 16_{10} = 11_{10} = B_{16}$ with a 1 carry

left column: $D_{16} + A_{16} + 1_{16} = 13_{10} + 10_{10} + 1_{10} = 24_{10}$

$24_{10} - 16_{10} = 8_{10} = 8_{16}$ with a 1 carry

Hexadecimal subtraction

- Convert the hexadecimal number to binary,
- Take 2’s complement of the binary,
- And perform subtraction,
- Then the result is converted back to hexadecimal.

Representing Negative Numbers

- A simple solution is to append a sign bit to the left hand end of a binary number
- 0 – positive 1 – negative
- Binary code now comprises sign (left most bit) and magnitude (remaining bits)
- 0 111 might represent $+7_{10}$
- 1 011 might represent -3_{10}

continue

- There are problems with manipulation of *sign and magnitude* representation – it is not possible to perform simple arithmetic operations that can produce the correct signed result.
- Hence, the standard method of representing negative numbers (signed integers) is 2's complement signed number representation.

2's Complement

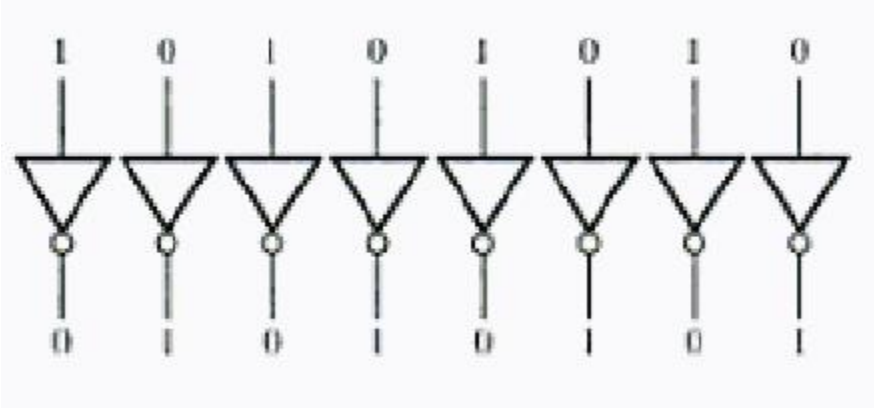
- To find the 2's complement representation of a negative number:

1. find 1's complement by inverting each bit of a binary number.
2. then add 1 to the inverted binary number.

- To find the 2's complement representation of -15_{10}
- Convert to binary $15_{10} = 01111_2$
- Invert this, i.e. $01111_2 \Rightarrow 10000_2$
- Add 1, i.e. $10000_2 \Rightarrow 10001_2$

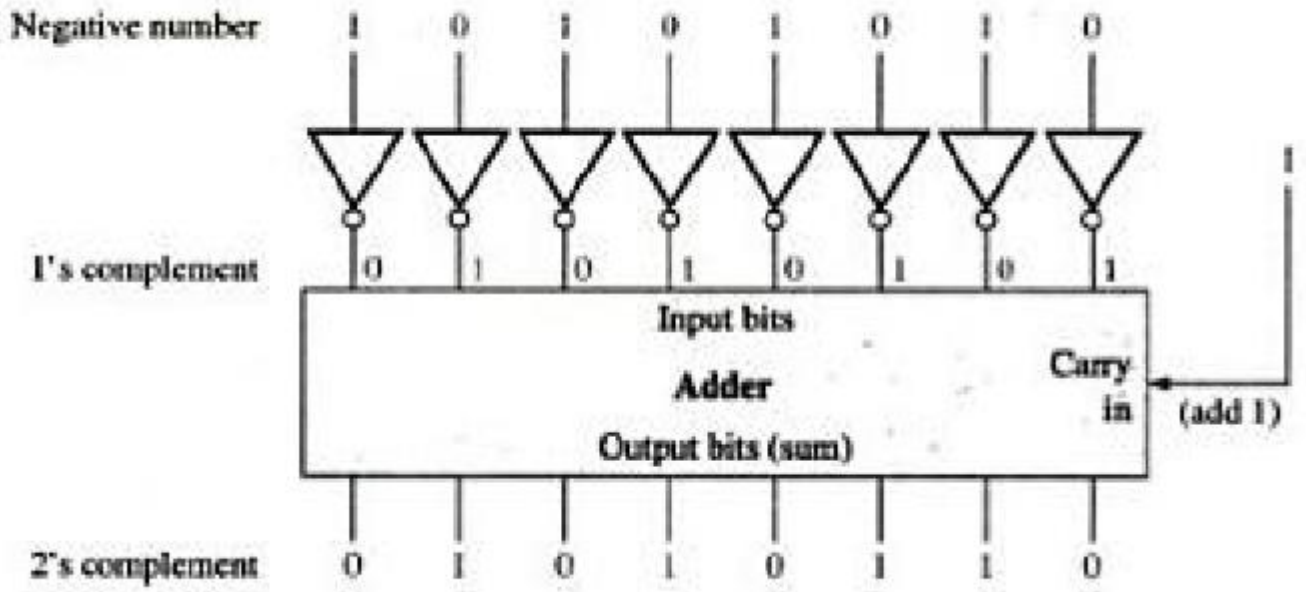
Continue

- Hardware converter for 1's complement consists of inverters arranged according to the size of a word in bits.



Continue

- 2’s complement hardware



2's Complement Register Sizing

- It's important to consider the full number of bits being used to represent a value (representing -15_{10} requires 5 bits)
- 2's complement representation incorporates a sign bit
- For example, a four bit register can accommodate a range of -8 to +7 2's complement numbers.
- In general a range could be worked out from the relation $-2^{(n-1)} < + (2^{(n-1)} - 1)$
- Where n is the number of bits.
- 2's complement representation leads to the following range of values that can be represented using 4 bits

Range for 4-bit 2's Complement

Binary	Decimal
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

2's complement is the standard way of representing signed integers

Range for 8-bit 2's Complement

- The range of values that can be represented as 8-bit signed integers is
- -128 to 127 inclusive

0 0000000	0
0 0000001	1
0 0000010	2
0 1111111	127
1 0000000	-128
1 0000001	-127
1 1111110	-2
1 1111111	-1

Exercise3

- What range of values can be represented as 16-bit signed integers?

Exercise4

- What range of values can be represented as 32-bit signed integers?

2’s Complement Arithmetic

Subtraction using 2’s complement

1. Take 2’s complement of a subtrahend (number to be subtracted).

2. Add the result to minuend (a number from which it is to be subtracted).

3. Ignore the carry if generated.

Decimal value of 2's complement number

- The decimal value of 2's complement is determined by the following relation:

$$\text{decimal value} = -b_n R^n + \ldots + b_1 R^1 + b_0 R^0$$

Continue

- Examples

Determine the decimal values of the signed binary numbers expressed in 2's complement:

(a) 01010110 **(b)** 10101010

Examples

1. Find 2's complement of 110_2

   soln

   step1: find the 1's complement of 110_2

   so we invert 110 = 001

   step2: add 1 to inverted 110

   001 + 1 = 010

Exercises5

- Work out the following problems:

1. Determine the 1's complement of each binary number:
   (a) 00011010 (b) 11110111 (c) 10001101
2. Determine the 2's complement of each binary number:
   (a) 00010110 (b) 11111100 (c) 10010001

Exercise6

- Perform each one of the following subtractions of the signed numbers using 2's complement method.

(a) 00001000 − 00000011 (b) 00001100 − 11110111

(c) 11100111 − 00010011 (d) 10001000 − 11100010

Floating point number

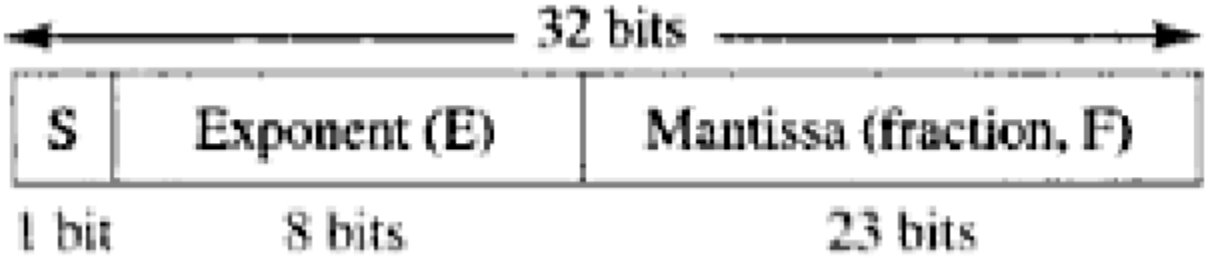
- It is difficult to represent a very large integer with its fraction in computer because computer memory capacity is finite.
- The answer to the above problem is the use of floating point number representation, which has the capacity of representing very large number without increase in the register size.

Continue

Floating point number = sign * mantissa * 10E

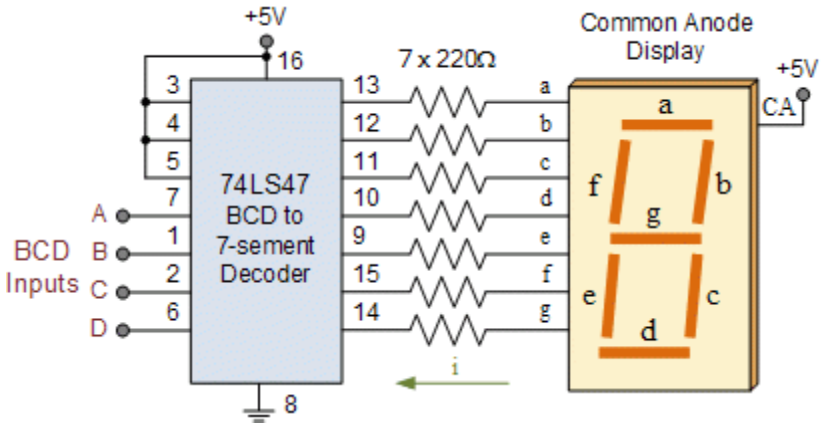
For binary floating point numbers, the format is defined by ANSI/IEEE Standard 754-1985 in three forms:

single-precision, double-precision, and extended-precision



BCD Code

- Used for decoding binary numbers to decimal numbers



Binary Coded Decimal - BCD

- It is a way to represent decimal number in bits or binary
- Each decimal digit is represented by a binary code of 4 bits
- With 4 bits, sixteen numbers can be represented but only 10 of these are used.
- Commonest BCD code is 8421
  - 8421 represents the weights of each of the 4 bits
  - six combinations are not used - 1010, 1011,1100, 1101, 1110, 1111
- Main advantage - ease of conversion between decimal and BCD
- To express any decimal number in BCD, simply replace each decimal digit by the appropriate 4 bit code

Binary Coded Decimal - BCD

Decimal BCD	8421 BCD	5211 BCD	Aitken 2421
0	0000	0000	0000
1	0001	0001	0001
2	0010	0011	0010
3	0011	0101	0011
4	0100	0111	0100
5	0101	1000	0101
6	0110	1001	0110
7	0111	1011	0111
8	1000	1101	1110
9	1001	1111	1111

Convert the two decimal number 47_{10} and 92_{10} into BCD code.

Gray code

- The gray code is unweighted and it is not an arithmetic code.
- The important feature of the code is that, it exhibits only a single bit change from one code word to to the next sequence.
- The property is important in many applications, such as position encoders, where error vulnerability increase with the number of bit changes between adjustment numbers in a sequence.
- Or a signal that is prone to signal corruption in telecommunication systems.

Binary to Gray code conversion

- The left most digit or bit of the binary number is also the left most bit of the gray code.
- The second left most bit of the GRAY code is obtained by modulo 2 of addition of the left most and second left most bits of the binary number.
- The third left most bit of the GRAY code is obtained by modulo 2 of addition of the second left most and third left most bits of the binary number.
- The fourth left most bit of the GRAY code is obtained by modulo 2 of addition of the third left most and fourth left most bits of the binary number.
- This rule is applied until the all the bits of the binary number are considered.

Modulo 2 addition

The modulo 2 addition of two numbers is

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 0$

Example

- Convert 0111_2 to GRAY code

  soln

  0 is the most left most bit

  0+1=1 is the second left most bit

  1+1=0 is the third left most bit

  1+1=0 is the third left most bit

0100_2

Exercise7

- Convert the following binary numbers to their equivalent GRAY codes:

  i) 0111_2

  ii) 10111_2

4-bit Gray Code

DECIMAL	BINARY	GRAY CODE	DECIMAL	BINARY	GRAY CODE
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

GRAY code to Binary

- The left most bit of the binary number is obtained by dividing the left most bit number of the GRAY code number.
- The second left most bit of the binary number is obtained by dividing the sum of left most bit number and the second left most bit of the GRAY code number.
- The rule is applied until all the bits are considered.

General formulation

The bits are arranged as follows:

$a_n/2, (a_n + a_{n-1})/2, \ldots (a_n + a_{n-1} + \ldots + a_0)/2$

Example

- Convert the following gray code number to binary number:

  11010_2

Soln

10011_2

Exercise8

- Convert the following GRAY code numbers to binary:

i) 1111_2

ii) 1000_2

iii) 1110_2

iv) 1101_2

v) 1001_2

ASCII Code

- **ASCII**, stands for American Standard Code for Information Interchange.
- It's a 7-bit character code where every single bit represents a unique character of about 256 characters.

Assignment2

- Exercise6
- Exercise7
- Exercise8

End