Digital Logic

CS2102

Sekhar Mandal

August 25, 2020

Outline of the Course

- Number systems, logic gates and Boolean algebra
- Representation, manipulation and minimization of Boolean Functions
- Design of Combinational Circuits
- Design of Sequential Circuits
- Concept of finite state machines
- Digital Integrated Circuits

Number systems

Systematic way to represent and manipulate numbers

Examples of number systems

- Decimal
- Binary
- Roman
- Sexagesimal

Broad Classification of number systems

- Weighted - decimal, binary etc.

-Non-weighted - Roman, Gray code etc.

Base or radix

- Decimal number system is said to be of base 10, because it uses 10 distinct digits (0, 1,2 ...9) and the coefficients are multiplied by power of 10.
- Binary number system has a base of 2, because it uses two digit (0 and 1) and each coefficient is a_i multiplied by 2^i.
- In general, a number system of base (radix) r uses r distinct digits (0, 1, ... (r-1)) and each digit has a weight of some power of r (say r^k.
  $k \geq 0$ for integer part
  $k <$ for the fractional part
  $D = d_{n-1}d_{n-2}\dots d_2d_1d_0.d_{-1}d_{-2}\dots d_m$

Signed and Unsigned Binary Number Representation

- Binary number system is important for designing Digital Circuits.
- Why are binary numbers important?
  - At the low-level, the circuit is implemented using Transistors.
  - A transistor has two stable states, either 'ON' or 'OFF'.
  - As the transistor has two states and can be denoted by two digits (0 or 1) in binary system.
- There are some conventions:
  - Open switch is denoted by 0 and closed switch is denoted by 1.
  - Low voltage is represented by 0 and high voltage is represented by 1.
  - Absence of current is represented by 0 and flow of current is represented by 1.

In this course, we will consider only binary number.
Some conventions:

- Bit $\Rightarrow$ Single binary digit (0 or 1)
- Nibble $\Rightarrow$ Collection of 4 bits
- Byte $\Rightarrow$ Collection of 8 bits
- Word $\Rightarrow$ Collection of 16/32/64 bits

Representing Number in Binary

How many distinct number can be represented using n bits?

- Each bit has two possible states.
- Total number of possible combinations:
  $2 \times 2 \times 2 \ldots upto\ n\ terms = 2^n$

Number can be signed or unsigned.
An unsigned number has only magnitude, no sign bit.
A signed number has both the magnitude and a sign (+ or -)

Unsigned Binary Number

An n-bit binary number system can have 2^n distinct numbers.
$Minimum = 0$ and $Maximum = 2^n - 1$
For example n = 3, there are 8 possible numbers
000 001 010 011 100 101 110 *and* 111
An n-bit binary integer can denoted as:
$b_{n-1} b_{n-2} \ldots b_2 b_1 b_0$
The equivalent unsigned decimal value is:
$D = b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} \ldots + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0$
Each digit position has a weight that is same power of 2.

Signed Integer Representation

There are three possible approaches to represent singed integers.

- Sign-magnitude representation.
- 1's Complement representation.
- 2's Complement representation.

Sign-Magnitude Representation

- For an n-bit number representation, the most significant bit (MSB) represent sign (0 $\Rightarrow$ positive and 1 $\Rightarrow$ negative).
- The rest of (n -1) bits denote the magnitude of the number.
- Range of numbers: $-(2^{n-1} - 1) to + (2^{n-1} - 1)$

Sign-magnitude Representation

A problem in sign-magnitude representation:
There are two possible representation of *zero*
$+0 = 0000$; $-0 = 1000$ (in 4 bits representation)

Decimal	Sign-Magnitude
+7	0111
+6	0110
+5	0101
+4	0100
+3	0011
+2	0010
+1	0001
+0	0000

Decimal	Sign-Magnitude
-0	1000
-1	1001
-2	1010
-3	1011
-4	1100
-5	1101
-6	1110
-7	1111

1's Complement Representation

Basic Idea

- Positive numbers are represented exactly as in sign-magnitude representation form.
- Negative numbers are represented in 1's complement form
- 1's complement of a number is obtained by complementing every bit of a number. (1 to 0 and 0 to 1).
- MSB will indicate the sign of the number.

Decimal	1's complement
+7	0111
+6	0110
+5	0101
+4	0100
+3	0011
+2	0010
+1	0001
+0	0000

Decimal	1's complement
-0	1111
-1	1110
-2	1101
-3	1100
-4	1011
-5	1010
-6	1001
-7	1000

1's Complement Representation

- In 1's complement representation, the numbers ranges from (maximum $= +(2^{n-1} - 1)$) to (minimum $=$ -$(2^{n-1} - 1)$) in n-bits representation.
- Similar to the sign-magnitude representation, 1's complement representation has two zeros.
- Advantage of the 1's complement representation is that subtraction can be done using addition which leads to saving in circuity.

2's Complement Representation

Basic Idea

- Positive numbers are represented exactly as sign-magnitude form.
- Negative numbers are represented in 2's complement form.
- To get the 2's complement of a number, we have to complement each bit of the number (1 to 0 and 0 to 1), and the add 1 to the resulting complement number.
- MSB will indicate the sign of the number (0 $\Rightarrow$ positive and 1 $\Rightarrow$ negative)

2's Complement Representation

Example of 2's complement representation for n = 4

Decimal	1's complement
+7	0111
+6	0110
+5	0101
+4	0100
+3	0011
+2	0010
+1	0001
+0	0000

Decimal	1's complement
-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111

- Range: Maximum $= +(2^{n-1} - 1)$ to Minimum $= -2^{n-1}$
- Advantages of 2's complement representation: (i) Unique representation of zero; (ii) Subtraction can be done using addition which leads to saving circuity.

2's Complement Representation

Additional features of 2's complement representation:

- Weighted number representation with MSB having weight -2^{n-1}.

  $$D = -b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} \ldots + b_1 \times 2^1 + b_0 \times 2^0$$
  $$0101 = 0 + 4 + 0 + 1 = 5$$
  $$1101 = -8 + 4 + 0 + 1 = -3$$
- Shift left by k positions with zero padding multiplies the number by 2^k.

  00001001 = 9 $\Rightarrow$ shift left by 2 positions 00100100 = 36

  11110111= -9 $\Rightarrow$ shift left by 2 positions 11011100 = -36
- Shift right by k positions with sign-bit padding divides the number by 2^k.

  00100100 = 36 $\Rightarrow$ shift right by 2 position 00001001 = 9
- The sign bit can be copied as many times as required at the beginning to extend the size of the number (called sign extension).

  X = 0111 (4 bit number value= 7), sign extend to 8 bit 00000111.