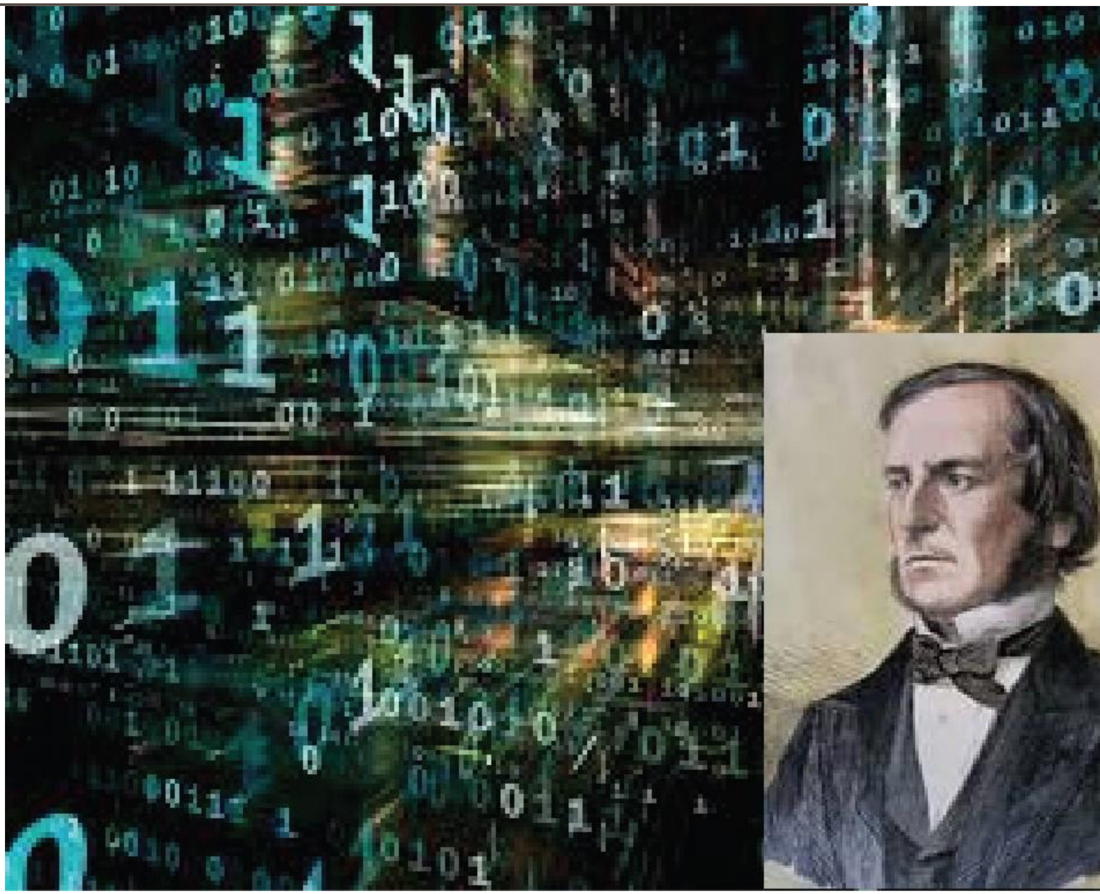
DIGITAL CIRCUITS

Week-5, Lecture-3
Number System

Sneh Saurabh

31st August, 2018



Digital Circuits: Announcements/Revision

Digital Circuits

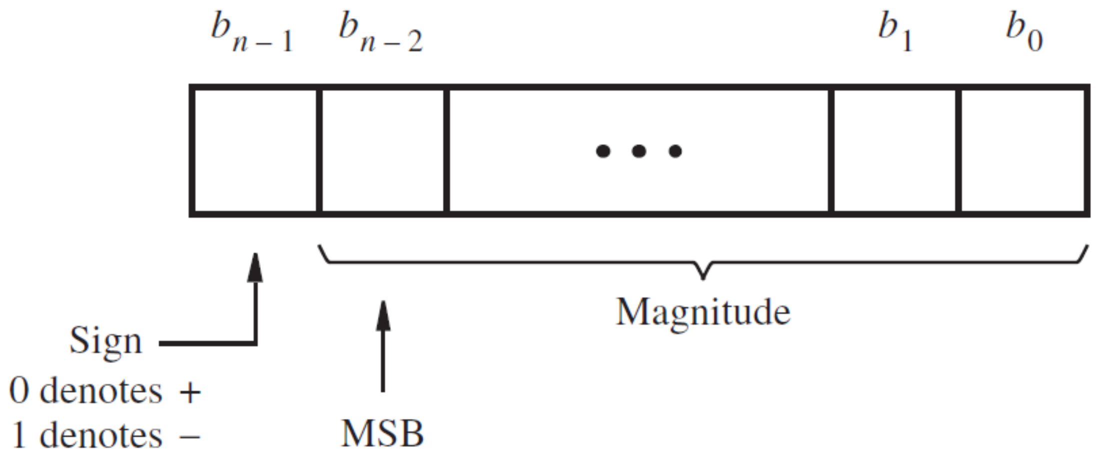
Number System

Digital Circuits

Signed Numbers

Signed Number Representation in Binary

- Positive numbers are represented by positional number system (as explained)
- Negative numbers can be represented in three ways:
  1. Sign-and-magnitude
  2. 1's complement
  3. 2's complement
- Sign is represented by the leftmost bit



Signed Number Representation: Sign and Magnitude

- Sign is represented by the leftmost bit
- Sign bit is 0 for positive numbers and 1 for negative numbers

- Example: In four-bit representation
  - $+5\text{: } 0101$
  - $-5\text{: } 1101$

- Easy to understand
- Difficult to implement in hardware compared to other systems

Signed Number Representation: 1's complement

- Let the number be represented with n −bits
- Let the negative number be K and the corresponding positive number be P
- In 1's complement $K = (2^n - 1) - P$

Problem: Given $n = 4$. Represent -5 using 1's complement

Solution:

- $(5)_{10} = (0101)_2$
- $(-5)_2 = (2^4 - 1)_{10} - (0101)_2 = (15)_{10} - (0101)_2$
- $= (1111)_2 - (0101)_2 = (1010)_2$

- Can be obtained by simply complementing each bit of the number (including sign bit)

Signed Number Representation: 2's complement

- Let the number be represented with n −bits
- Let the negative number be K and the corresponding positive number be P
- In 2's complement $K = 2^n - P$

Problem: Given $n = 4$. Represent -5 using 2's complement.

Solution:

- $(5)_{10} = (0101)_2$
- $(-5)_2 = (2^4)_{10} - (0101)_2 = (16)_{10} - (0101)_2$
- $= (10000)_2 - (0101)_2 = (1011)_2$

- Can be obtained by simply adding $+1$ to the 1's complement

To quickly find 2's complement:

- Start looking the bits from right to left: copy all bits that are zero and the first bit that is one; then simply complement rest of the bits

Example:

- Given Number: $10110\ \ 100$
- 2's complement: $01001\ \ 100$

Signed Number Representation: A table

Table 5.2 Interpretation of four-bit signed integers.

$b_3b_2b_1b_0$	Sign and magnitude	1's complement	2's complement
0111	+7	+7	+7
0110	+6	+6	+6
0101	+5	+5	+5
0100	+4	+4	+4
0011	+3	+3	+3
0010	+2	+2	+2
0001	+1	+1	+1
0000	+0	+0	+0
1000	−0	−7	−8
1001	−1	−6	−7
1010	−2	−5	−6
1011	−3	−4	−5
1100	−4	−3	−4
1101	−5	−2	−3
1110	−6	−1	−2
1111	−7	−0	−1

Digital Circuits

Arithmetic of signed numbers

Addition of signed number: Sign and Magnitude

- Addition of numbers with the same sign: use adder and retain the sign-bit

- Addition of numbers with opposite sign:
  - Sign of result depends on the absolute value of the two numbers
  - Comparator will be required
  - Subtraction is required

- Hardware for addition of numbers in sign/magnitude form is not efficient
- Sign and magnitude representation is not used in computers

Addition of signed number: 1’s complement

Table 5.2 Interpretation of four-bit signed integers.

$b_3b_2b_1b_0$	Sign and magnitude	1’s complement	2’s complement
0111	+7	+7	+7
0110	+6	+6	+6
0101	+5	+5	+5
0100	+4	+4	+4
0011	+3	+3	+3
0010	+2	+2	+2
0001	+1	+1	+1
0000	+0	+0	+0
1000	−0	−7	−8
1001	−1	−6	−7
1010	−2	−5	−6
1011	−3	−4	−5
1100	−4	−3	−4
1101	−5	−2	−3
1110	−6	−1	−2
1111	−7	−0	−1

$$\begin{array}{r} (+5) \\ +\ (+2) \\ \hline (+7) \end{array} \qquad \begin{array}{r} 0101 \\ +\ 0010 \\ \hline 0111 \end{array} \qquad\qquad \begin{array}{r} (-5) \\ +\ (+2) \\ \hline (-3) \end{array} \qquad \begin{array}{r} 1010 \\ +\ 0010 \\ \hline 1100 \end{array}$$

$$\begin{array}{r} (+5) \\ +\ (-2) \\ \hline (+3) \end{array} \qquad \begin{array}{r} 0101 \\ +\ 1101 \\ \hline 1\ 0010 \\ 1 \\ \hline 0011 \end{array} \qquad\qquad \begin{array}{r} (-5) \\ +\ (-2) \\ \hline (-7) \end{array} \qquad \begin{array}{r} 1010 \\ +\ 1101 \\ \hline 1\ 0111 \\ 1 \\ \hline 1000 \end{array}$$

- There is a carry out from sign-bit position
- Carry out from sign-bit can be added to LSB to get correct result
- Extra addition will be required in certain cases

Addition of signed number: 2's complement

Table 5.2 Interpretation of four-bit signed integers.

$b_3b_2b_1b_0$	Sign and magnitude	1's complement	2's complement
0111	+7	+7	+7
0110	+6	+6	+6
0101	+5	+5	+5
0100	+4	+4	+4
0011	+3	+3	+3
0010	+2	+2	+2
0001	+1	+1	+1
0000	+0	+0	+0
1000	−0	−7	−8
1001	−1	−6	−7
1010	−2	−5	−6
1011	−3	−4	−5
1100	−4	−3	−4
1101	−5	−2	−3
1110	−6	−1	−2
1111	−7	−0	−1

$$\begin{array}{r} (+5) \\ +\ (+2) \\ \hline (+7) \end{array} \qquad \begin{array}{r} 0101 \\ +\ 0010 \\ \hline 0111 \end{array} \qquad\qquad \begin{array}{r} (-5) \\ +\ (+2) \\ \hline (-3) \end{array} \qquad \begin{array}{r} 1011 \\ +\ 0010 \\ \hline 1101 \end{array}$$

$$\begin{array}{r} (+5) \\ +\ (-2) \\ \hline (+3) \end{array} \qquad \begin{array}{r} 0101 \\ +\ 1110 \\ \hline 1\,0011 \end{array} \qquad\qquad \begin{array}{r} (-5) \\ +\ (-2) \\ \hline (-7) \end{array} \qquad \begin{array}{r} 1011 \\ +\ 1110 \\ \hline 1\,1001 \end{array}$$

ignore　　　　ignore

- There is a carry out from sign-bit position, which can be ignored
- Irrespective of sign of the operand, the same adder circuit can be used

Subtraction of signed number: 2’s complement

Table 5.2 Interpretation of four-bit signed integers.

$b_3b_2b_1b_0$	Sign and magnitude	1’s complement	2’s complement
0111	+7	+7	+7
0110	+6	+6	+6
0101	+5	+5	+5
0100	+4	+4	+4
0011	+3	+3	+3
0010	+2	+2	+2
0001	+1	+1	+1
0000	+0	+0	+0
1000	−0	−7	−8
1001	−1	−6	−7
1010	−2	−5	−6
1011	−3	−4	−5
1100	−4	−3	−4
1101	−5	−2	−3
1110	−6	−1	−2
1111	−7	−0	−1

- Find 2’s complement of subtrahend and then add to minuend

$$\begin{array}{r} (+5)\\ -\,(+2)\\ \hline (+3)\end{array} \qquad \begin{array}{r} 0101\\ -\,0010\\ \hline \end{array} \quad \Rightarrow \quad \begin{array}{r} 0101\\ +\,1110\\ \hline 10011\end{array}$$

ignore

$$\begin{array}{r} (-5)\\ -\,(+2)\\ \hline (-7)\end{array} \qquad \begin{array}{r} 1011\\ -\,0010\\ \hline \end{array} \quad \Rightarrow \quad \begin{array}{r} 1011\\ +\,1110\\ \hline 11001\end{array}$$

ignore

Subtraction of signed number: 2’s complement

Table 5.2 Interpretation of four-bit signed integers.

$b_3b_2b_1b_0$	Sign and magnitude	1’s complement	2’s complement
0111	+7	+7	+7
0110	+6	+6	+6
0101	+5	+5	+5
0100	+4	+4	+4
0011	+3	+3	+3
0010	+2	+2	+2
0001	+1	+1	+1
0000	+0	+0	+0
1000	−0	−7	−8
1001	−1	−6	−7
1010	−2	−5	−6
1011	−3	−4	−5
1100	−4	−3	−4
1101	−5	−2	−3
1110	−6	−1	−2
1111	−7	−0	−1

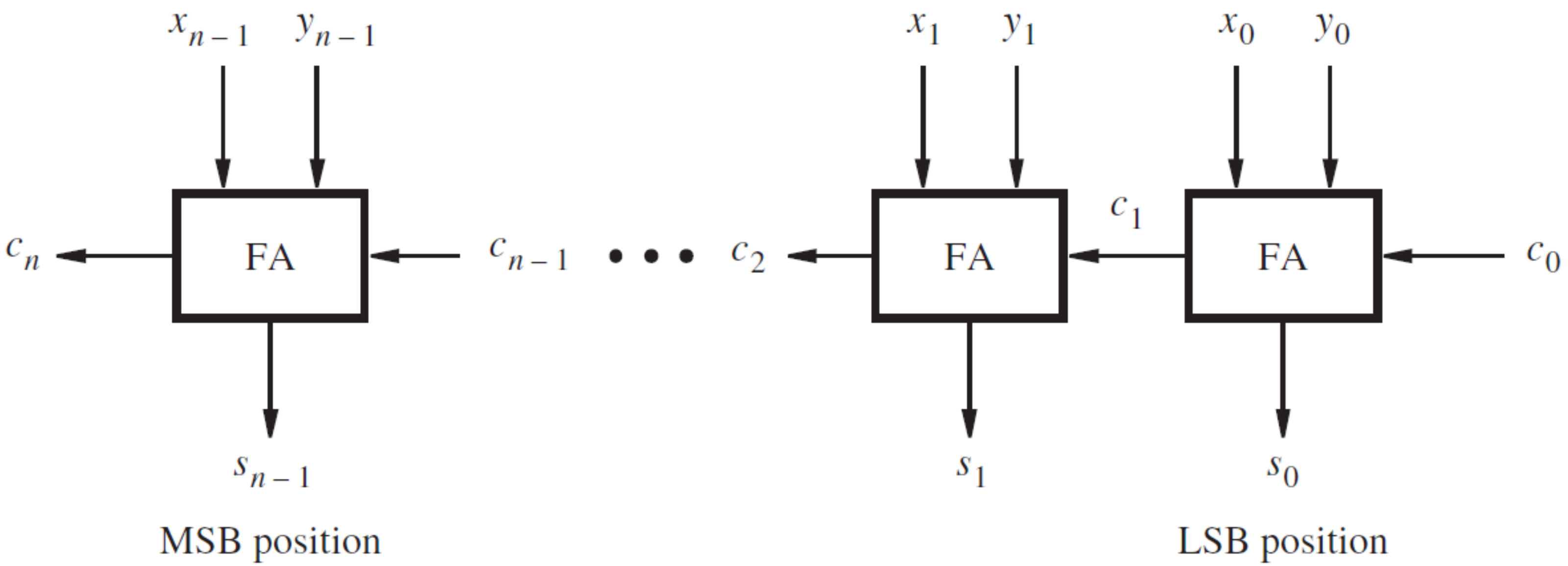
- Find 2’s complement of subtrahend and then add to minuend

$$\begin{array}{r} (+5) \\ -\ (-2) \\ \hline (+7) \end{array} \qquad \begin{array}{r} 0\,1\,0\,1 \\ -\ 1\,1\,1\,0 \\ \hline \end{array} \quad \Rightarrow \quad \begin{array}{r} 0\,1\,0\,1 \\ +\ 0\,0\,1\,0 \\ \hline 0\,1\,1\,1 \end{array}$$

$$\begin{array}{r} (-5) \\ -\ (-2) \\ \hline (-3) \end{array} \qquad \begin{array}{r} 1\,0\,1\,1 \\ -\ 1\,1\,1\,0 \\ \hline \end{array} \quad \Rightarrow \quad \begin{array}{r} 1\,0\,1\,1 \\ +\ 0\,0\,1\,0 \\ \hline 1\,1\,0\,1 \end{array}$$

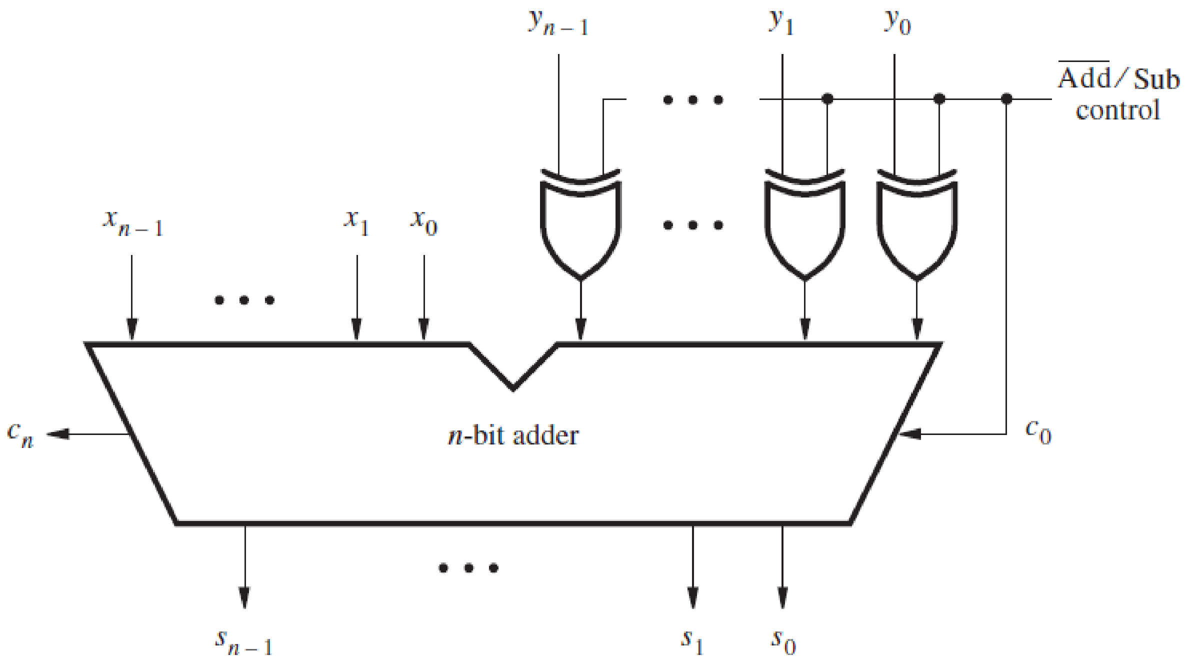
- Subtraction operation can be realized using addition operation (after taking 2’s complement of subtrahend)

N-bit adder (recap…)



- To add two unsigned numbers, circuit is designed similar to what is done in hand-calculation
- Least Significant Bit (LSB) is on the right and Most Significant Bit (MSB) is on the left
- Bits are added starting from right using Full Adders
- Carry bits propagate from right to left

Adder and Subtractor unit



Hardware is shared between addition and subtraction: reduces complexity and cost

For addition:

- $\overline{Add}/Sub = 0$: $y_0, y_1, \ldots$ reaches adder as it is
- $c_0 = 0$: carry-in is 0 for the LSB
- Normal addition is done

For subtraction:

- $\overline{Add}/Sub = 1$: $y_0, y_1, \ldots$ gets inverted before adder
- $c_0 = 1$: carry-in is 1 for the LSB
- 2's complement of Y is obtained

Digital Circuits

Binary Codes

Binary codes: Introduction

- A binary number of $n - digits$ can be represented by n binary circuit elements, each having an output signal corresponding to 0 and 1
- Binary codes are patterns/group of zeros and ones
- Any discrete information that is distinct within a group can be represented using binary codes
- Binary codes merely change the symbol representing data: the meaning of data is not changed
- Motivation for using binary code is ease of manipulation, storage or transmission of data

Binary codes: Number of bits

- An $n - bit$ binary code is a group of $n - bits$ that assumes up to 2^n combination of zeroes and ones
- Each combination represent one element of the set that is being encoded: code should be unique for each element
- Though minimum number of bits required to represent 2^n elements is n, there is no maximum number of bits that can be used in encoding (some bits may be unused)