Lecture-4

Chapter-3.2

Computer Architecture and Organization-
Jhon P. Hayes

Data Representation

Data Representation

- Binary: The computer numbering system.

1 binary digit allows 2^n = 2 codes (0,1)

2 binary digit allows 2^n = 4 codes (00,01,10,11)

3 binary digit allows 2^n = 8 codes (000,…..111)

…………………………………………………

7 binary digit allows 2^n = 128 codes (0000000,…..1111111)

8 binary digit allows 2^n = 256 codes (00000000,…..11111111)

ASCII: American Standard Code for Information Interchange

EBCDIC: Extended Binary Coded Decimal Interchange Code

Data Representation

- Bit: 0 (Off) or 1 (On).

- Byte: 8 bits can make a byte.

- Word: The word is the computer’s basic unit of data, the unit concerned in data storage, processing and transfer.

Data Representation

- Integer:

- Floating Point:

- Character:
  - **ASCII**
  - **EBCDIC**
  - **Unicode**

- Boolean:

Integers

Decimal	Binary
1	00000001
4	00000100
9	00001001
-1	11111111
-4	11111100
-9	11110111

Positive Number Representation

Using Integers: The simplest numbers to consider are the integers. The positive integer numbers are called *unsigned*. And the integer numbers that can also be negative are called *signed*.

For an example the number 13 represents,

$$13 = 1 \times 10^1 + 3 \times 10^0$$

Positive Number Representation

Conversion between Decimal and Binary systems:

1. For an example the binary number 1101 represents the value

$$V = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$V = 13$$

Hence, $(1101)_2 = (13)_{10}$

2. The decimal number 13 represents the value

$$\begin{array}{r|l} 2 & 13 \\ \hline 2 & 6 - 1 \\ \hline 2 & 3 - 0 \\ \hline & 1 - 1 \end{array}$$

Hence, $(13)_{10} = (1101)_2$

Positive Number Representation

Conversion Octal and Hexadecimal Representation:

1. The decimal number 125 represents the octal value

$$
\begin{array}{r|l}
8 & 125 \\
\hline
8 & 15 - 5 \\
\hline
 & 1 - 7
\end{array}
$$

Hence, $(125)_{10} = (175)_8$

2. The decimal number 125 represents the Hexadecimal value

$$
\begin{array}{r|l}
16 & 125 \\
\hline
 & 7 - 13
\end{array}
$$

Hence, $(125)_{10} = (7D)_{16}$

Positive Number Representation

Conversion Octal to Binary and Binary to Hexadecimal Representation:

1. The octal number 175_8 represents the binary value

$$(175)_8 = 001\ 111\ 101$$

Hence, $(175)_8 = (001111101)_2$

2. The binary number $(001111101)_2$ represents the Hexadecimal

$$(001111101)_2 = 0000\ 0111\ 1101$$

$$= 0\ 7\ D$$

Hence, $(001111101)_2 = 7D$

Number in different systems

Decimal	Binary	Octal	Hexadecimal
00	00000	00	00
01	00001	01	01
02	00010	02	02
03	00011	03	03
04	00100	04	04
05	00101	05	05
06	00110	06	06
07	00111	07	07
08	01000	10	08
09	01001	11	09
10	01010	12	0A
11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10
17	10001	21	11
18	10010	22	12

Examples

1. Find out the unsigned integer, real, binary, octal and hexadecimal values of 150 denary number?

2. If the decimal value of B is 66, find out the both decimal and binary values of G and I?

Addition of Unsigned Number

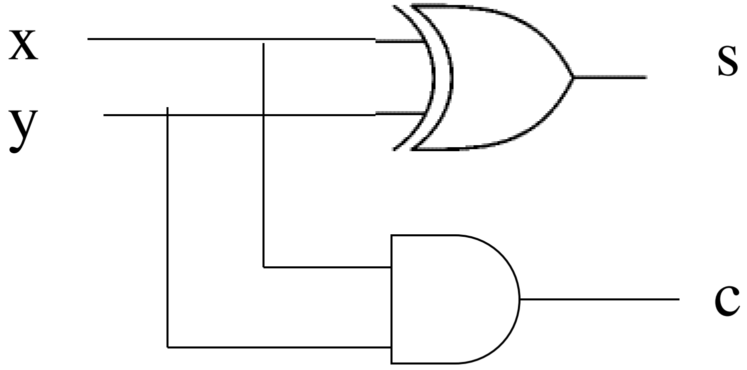
x	0	0	1	1
+y	+0	+1	+0	+1
c s	00	01	10	10

(a) The four possible cases

x	y	Carry c	Sum s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(b) Truth Table for half adder

Addition of Unsigned Number



(c) Circuit

(d) Graphical symbol

Addition of Unsigned Number

c_i	x_i	y_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

(a) Truth table for full adder

Addition of Unsigned Number

c_i \ $x_i y_i$	00	01	11	10
0		1		1
1	1		1	

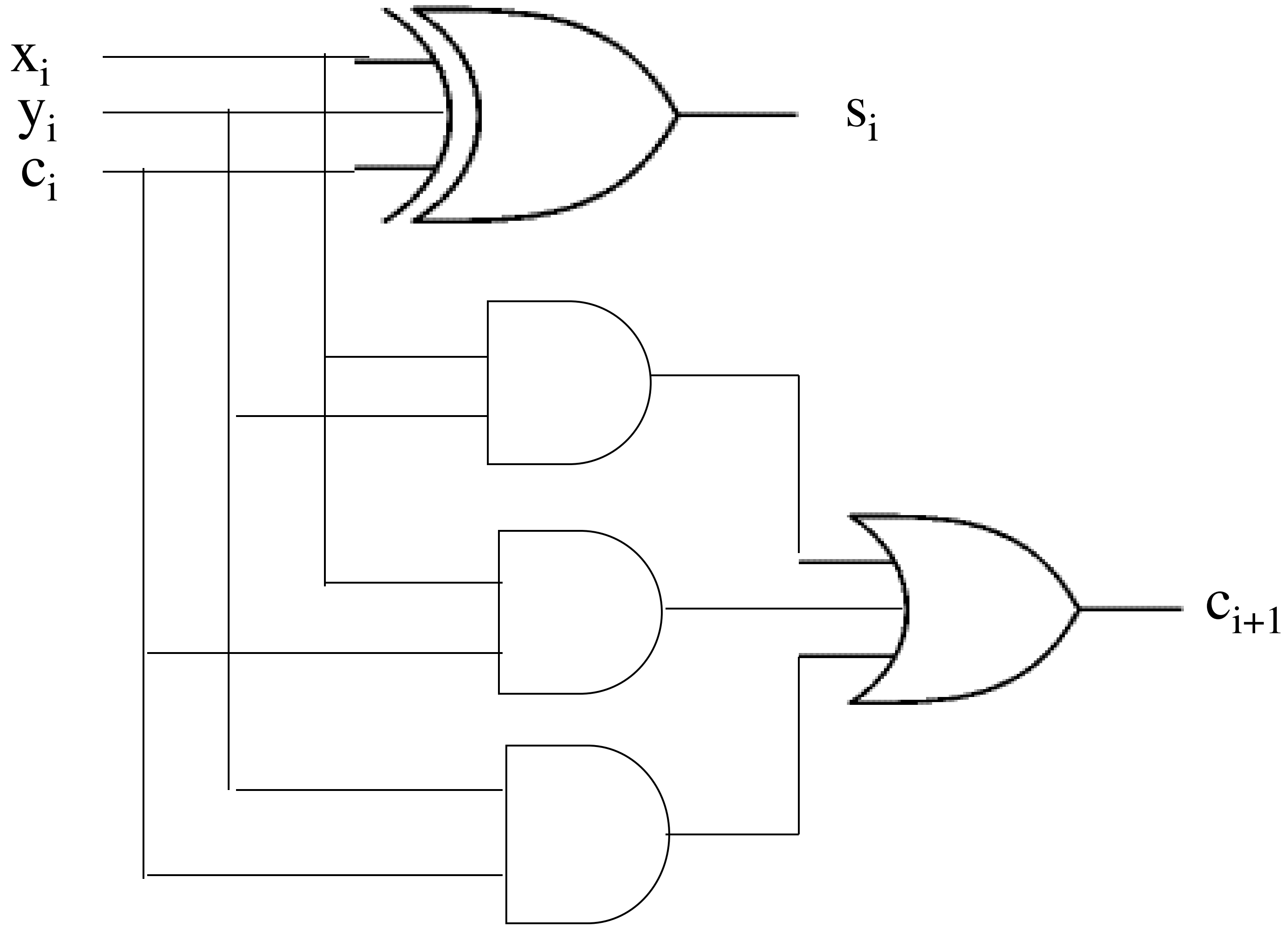
$$s_i = x_i \oplus y_i \oplus c_i$$

c_i \ $x_i y_i$	00	01	11	10
0			1	
1		1	1	1

$$c_{i+1} = x_i y_i + x_i c_i + y_i c_i$$

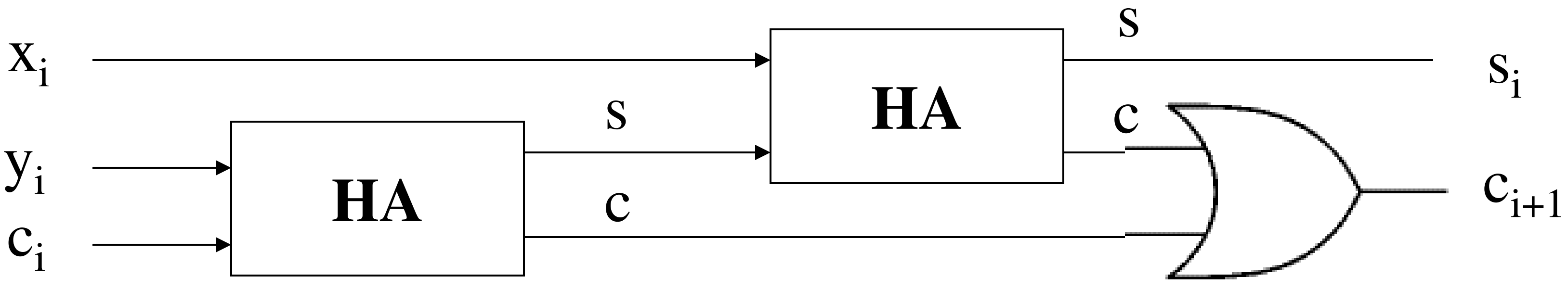
(b) Karnaugh maps

Addition of Unsigned Number

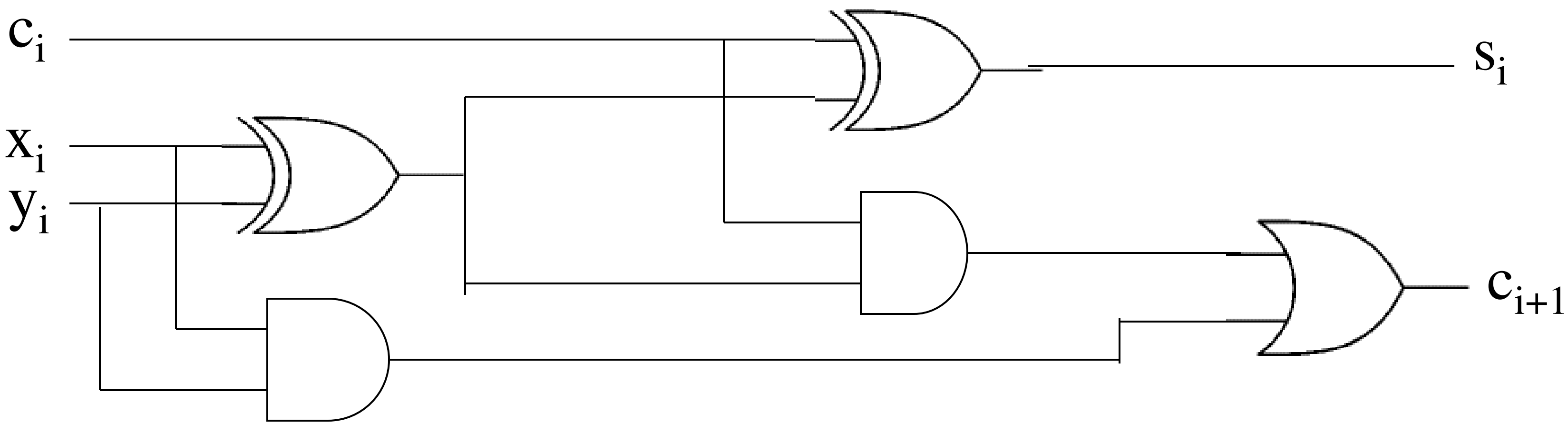


(c) Circuit for Full adder

Decomposed Full Adder



(a) Block diagram



(b) Detailed diagram

Ripple Carry Adder

The signal c_{n-1} is valid after a delay of (n-1)×dt, which means that the complete sum is available after a delay of n×dt. Because of the way the carry signals ripple through the full adder stages, the circuit in figure is called a ***ripple carry adder***.

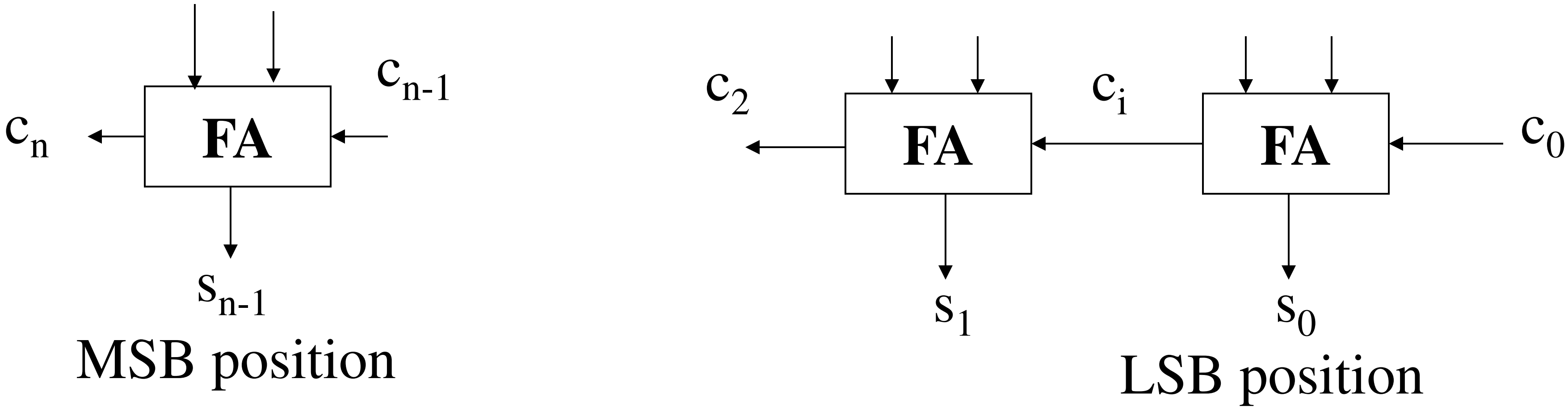


Figure: An n-bit ripple carry adder

Negative Number Representation

Negative numbers can be represented in three different ways:

- Sign and magnitude
- 1’s complement
- 2’s complement

2’s Complements

To Translate a negative denary (base 10) number to binary Using 2’s complements:

- Find the binary value of the equivalent positive decimal Number.

- Change all the 1s to 0 and all the 0s to 1.

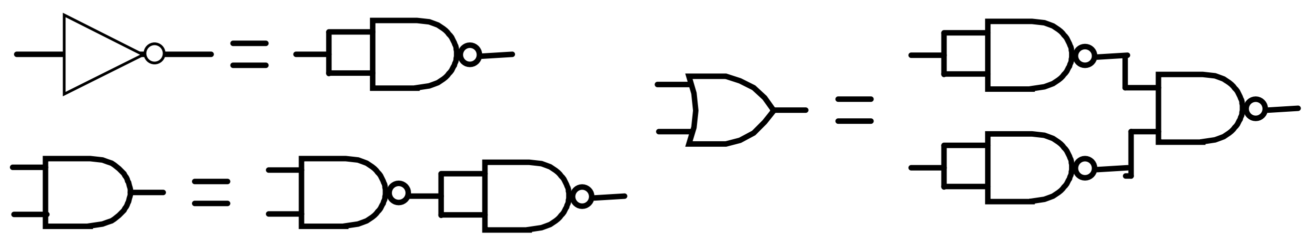
- Add 1 to the result.

Examples of ASCII Codes

Character	ASCII
0	00110000
1	00110001
2	00110010
-	
8	00111000
9	00111001
-	
A	01000001
B	01000010
-	
Y	01011001
Z	01011010
-	
a	01100001
b	01100010

Logic Gate Implementations

- Functionally Complete Set of Gates
  - { + , * , ‘ }
  - { + , ‘ }
  - { * , ‘ }
- Universal Gate
  - NAND



  - Similarly for NOR

Conversion to NAND-Gate Circuits