BASUDEV GODABARI DEGREE COLLEGE , KESAIBAHAL

Department of Computer Science

"SELF STUDY MODULE"

Module Details :

- **Class – 1st Semester (2020-21) Admission Batch**
- **Subject Name : COMPUTER SCIENCE**
- **Paper Name : DIGITAL LOGIC**

UNIT – 2 : STRUCTURE

2.1 Introduction to Binary Number System
2.2 Addition and Subtraction of Signed number
2.3 Addition/Subtraction Logic Gate
2.4 Design of Fast Adders : Carry – Look Ahead Addition
2.5 Multiplication of Positive Number
2.6 Signed Operand Multiplication of Positive Numbers Both Fast Multiplication and Bit-Pair Multiplication
2.7 Carry – Save Addition and Summands
2.8 Integer Division, Floating Point Numbers and Operations
2.9 Guard Bits and Truncation
2.10 Implementing Floating – Point Operation

Learning Objective

After Learning this unit you should be able to

- Know the Binary number system and there arithmetic Operation.
  Addition, Subtraction, Multiplication, Division etc.
- Know the Multiplication of Positive Number Fast Multiplication and Bit-Pair Multiplication.
- Know the Carry – Save Addition and Summands.
- Able to Understand the Guard Bits and Truncation Implementing Floating – Point Operation.

You Can use the Following Learning Video link related to above topic :

https://youtu.be/wPeQxJD4I9g?list=PLir19lgiavA0EKRRN3xdARy_z44hKhNQc
https://youtu.be/kAnBaQoJkpo
https://youtu.be/NY2gz_Kzc0Q
https://youtu.be/vZ4yF0dEyzU
https://youtu.be/8afbTaA-gOQ

You Can also use the following Books :

- **Digital Electronics** : An Introduction To Theory And Practice by William Gothmann H – USA.
- **Digital Electronics** by John Morris – USA.
- **Digital Electronics** by John Morris – UK.
- Fundamentals of **Digital** Circuits by Anand Kumar – US

And also you can download any book in free by using the following website.

- https://www.pdfdrive.com/

Introduction to Binary Number System:-

Binary is a base-2 **number system**, which only uses two digits (0 & 1). It is a **system** used at the heart of every digital computer, allowing them to encode information, perform arithmetic operations and execute logical control processes.

Addition and Subtraction of Signed number:-

When **adding** two **numbers** with like signs, add the values and keep the common sign.
When **adding** two **numbers** with unlike signs, **subtract** the values and use the sign of the larger-valued **number**. Change the **subtraction** operator to **addition** and change the sign of the **number** that immediately follows.

1. **Addition of signed numbers**

   When **adding** two **numbers** with the same sign (either both positive or both **negative**), add the absolute values (the **number** without a sign attached) and keep the same sign.
2. What is signed subtraction?
   The **subtraction** of two **signed** binary numbers is similar to the addition of two **signed** binary numbers. But, we have to take 2's complement of the number, which is supposed to be subtracted. This is the advantage of 2's complement technique. Follow, the same rules of addition of two **signed** binary numbers.
   Subtracting Positive and Negative Numbers

   Rule 1: **Subtracting** a **positive number** from a **positive number** – it's just normal **subtraction**.
   Rule 2: **Subtracting** a **positive number** from a negative **number** – start at the negative **number** and count backwards.

Adders and Subtractors

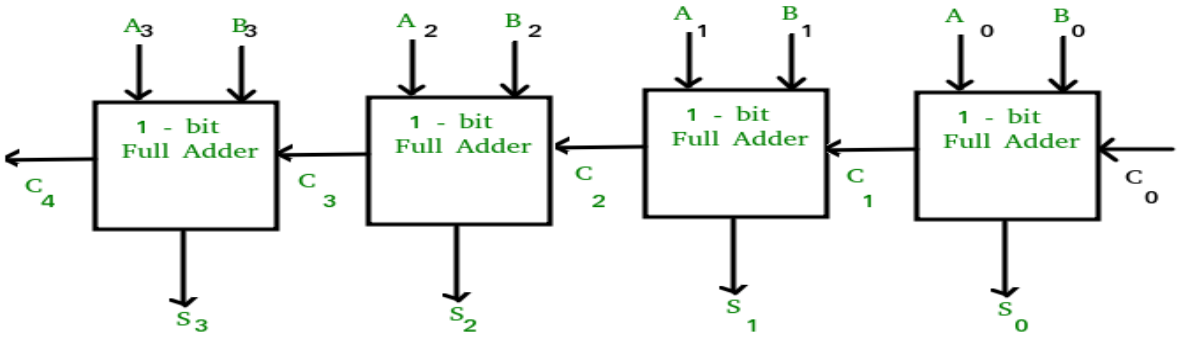
Computers are good at math. Computers, as we've seen, are made out of simple gates. Gates just do simple logic functions like AND and OR, not math like addition and subtraction. How do we reconcile this?
Simple... we make circuits out of logic gates that can do math. In this section we'll have a look at adders and subtractors.
This also provides a few good learning opportunities to bring out some lessons having to do with digital circuit design.
Let's start simply: adding 2 1-bit numbers. Recall from math class that adding numbers results in a sum and a carry. It's no different here. With two one bit numbers we have 4 distinct cases:

1. 0 + 0 = 0 with no carry
2. 0 + 1 = 1 with no carry
3. 1 + 0 = 1 with no carry
4. 1 + 1 = 0 with a carry

Carry Look-Ahead Adder:-

Motivation behind Carry Look-Ahead Adder :

In ripple carry adders, for each adder block, the two bits that are to be added are available instantly. However, each adder block waits for the carry to arrive from its previous block. So, it is not possible to generate the sum and carry of any block until the input carry is known. The block waits for the block to produce its carry. So there will be a considerable time delay which is carry propagation delay.

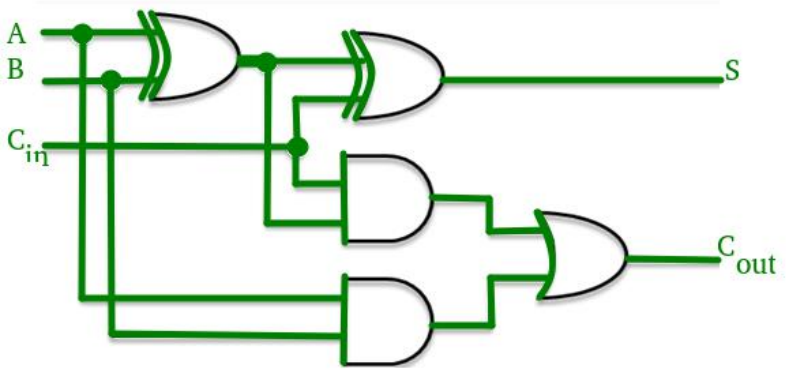


Consider the above 4-bit ripple carry adder. The sum is produced by the corresponding full adder as soon as the input signals are applied to it. But the carry input is not available on its final steady state value until carry is available at its steady state value. Similarly depends on and on . Therefore, though the carry must propagate to all the stages in order that output and carry settle their final steady-state value.

The propagation time is equal to the propagation delay of each adder block, multiplied by the number of adder blocks in the circuit. For example, if each full adder stage has a propagation delay of 20 nanoseconds, then will reach its final correct value after 60 (20 × 3) nanoseconds. The situation gets worse, if we extend the number of stages for adding more number of bits.

Carry Look-ahead Adder :

A carry look-ahead adder reduces the propagation delay by introducing more complex hardware. In this design, the ripple carry design is suitably transformed such that the carry logic over fixed groups of bits of the adder is reduced to two-level logic. Let us discuss the design in detail.



A	B	C	C +1	Condition
0	0	0	0	
0	0	1	0	No Carry Generate
0	1	0	0	
0	1	1	1	
1	0	0	0	No Carry Propogate
1	0	1	1	
1	1	0	1	Carry Generate
1	1	1	1	

What is a positive multiplied by a positive?

Rule 1: A **positive** number **times** a **positive** number equals a **positive** number. This is the **multiplication** you have been doing all along, **positive** numbers **times positive** numbers equal **positive** numbers. For example , 5 x 3 = 15. 5 is a **positive** number, 3 is a **positive** number and **multiplying** equals a **positive** number: 15.

Multiplication Algorithm in Signed Magnitude Representation

Multiplication of two fixed point binary number in *signed magnitude representation* is done with process of *successive shift* and *add operation*.

```
   10111 (Multiplicand)
x  10011 (Multiplier)
  ------
   10111
   10111
  00000
 00000
10111
---------
011011010   (Product)
---------
```

In the multiplication process we are considering successive bits of the multiplier, least significant bit first.
If the multiplier bit is 1, the multiplicand is copied down else 0's are copied down.

The numbers copied down in successive lines are shifted one position to the left from the previous number.
Finally numbers are added and their sum form the product.

The sign of the product is determined from the sign of the multiplicand and multiplier. If they are alike, sign of the product is positive else negative.

QUESTION BANK

MULTIPLE CHOICE. Choose the one alternative that best completes the statement or answers the question.

1) Convert binary 010101 to octal.
A) 25_8 B) 5_8 C) 21_8 D) 15_8

2) Any number with an exponent of zero is equal to
A) itself. B) ten. C) zero. D) one.

3) Convert binary 10101010 to octal.
A) 206_8 B) 522_8 C) 255_8 D) 252_8

4) Convert octal 377 to binary.
A) 11101101 B) 01111011 C) 10110111 D) 11111111

5) Convert octal 54 to decimal.
A) 64_{10} B) 58_{10} C) 76_{10} D) 44_{10}

6) Convert the binary number 0000.1010 to decimal.
A) 0.10 B) 0.55 C) 0.50 D) 0.625

7) Convert octal 36 to binary.
A) 110110 B) 100110 C) 110011 D) 011110

8) The decimal equivalent of binary 1010000 is _____.
A) 96_{10} B) 80_{10} C) 78_{10} D) 84_{10}

9) Convert octal 701 to binary.
A) 11000001 B) 111000001 C) 1000111 D) 111000100

10) Convert the binary number 1011.1110 to decimal.
A) 11.875 B) 11.675 C) 13.875 D) 13.75

11) What is the decimal value of 2^{-2}?
A) 0.5 B) 0.125 C) 0.2 D) 0.25

12) Convert the decimal number 12.125 to binary.
A) 1100.0110 B) 1110.0010 C) 1100.0010 D) 1010.1100

13) Which binary value equals 2^{-2}?
A) 0000.0010 B) 0010.0000 C) 0000.1000 D) 0000.0100

14) Convert the decimal number 6.75 to binary.
A) 0110.1100 B) 0111.1100 C) 0110.0110 D) 0110.1010

15) How many symbols are used in the hexadecimal number system?

A) sixteen B) six C) ten D) twelve

ANSWER

1) A
2) D
3) D
4) D
5) D
6) D
7) D
8) B
9) B
10) A
11) D
12) C
13) D
14) A
15) A

ONLINE QUESTION LINK

https://www.cimt.org.uk/projects/mepres/book9/bk9i1/bk9_1i1.html