

Arithmetic and logic unit

Unit: 2

Computer Organization &
Architecture

B Tech 3rd Sem



Sanjay Kumar Nayak
CSE
Department



Arithmetic and logic unit:

- Look ahead carries adders.
- Multiplication: Signed operand multiplication,
- Booths algorithm and array multiplier.
- Division and logic operations.
- Floating point arithmetic operation,
- Arithmetic & logic unit design.
- IEEE Standard for Floating Point Numbers

Course Objective

- Analysis of the design of arithmetic & logic unit and understanding of the fixed point and floating-point arithmetic operations.
- Study of various multiplication algorithm and division for signed no. like booth algorithm and others.

- Analysis and design of arithmetic & logic unit and understanding of the fixed point and floating-point arithmetic operations.

COMPUTER ORGANIZATION AND ARCHITECTURE (KCS-302)

CO.K	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
KCS-302.2	2	2	2	2	1	1	-	1	1	1	1	2

COMPUTER ORGANIZATION AND ARCHITECTURE (KCS-302)

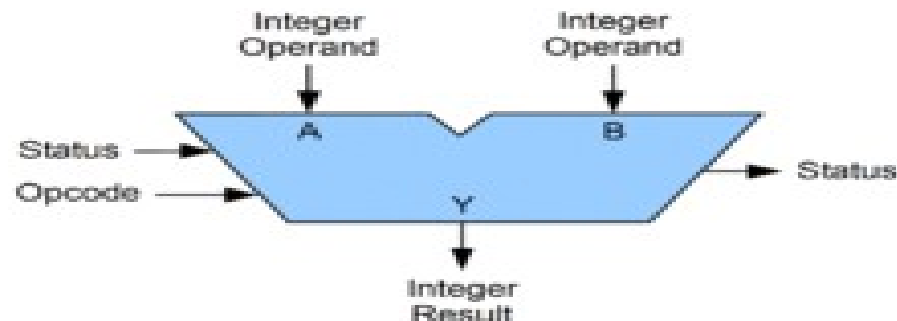
CO.K	PSO1	PSO2	PSO3	PSO4
KCS-302.2	2	2	2	1

Prerequisite and Recap

- **Fundamental of computer**
- **Interconnection of computer**

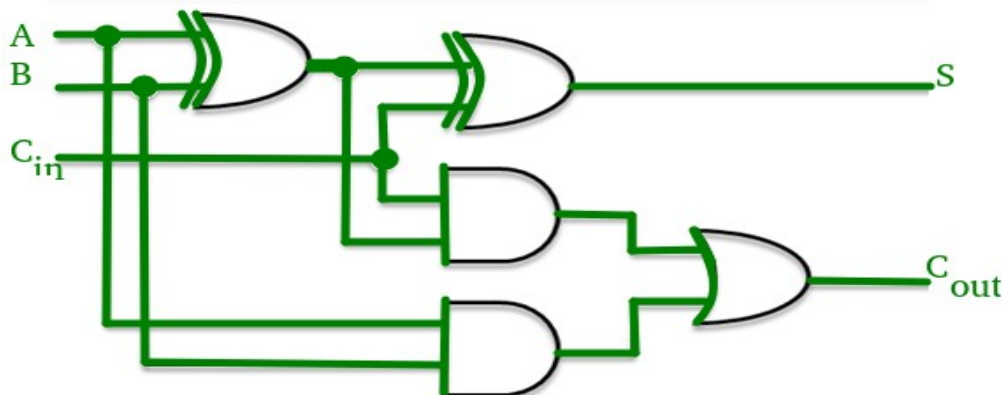
Arithmetic Logic Unit (ALU)

- An **arithmetic logic unit (ALU)** is a digital circuit used to perform **arithmetic** and **logic** operations.
- It represents the fundamental building block of the central processing **unit** (CPU) of a computer. Modern CPUs contain very powerful and complex ALUs.
- In addition to ALUs, modern CPUs contain a **control unit** (CU)
- The inputs to an ALU are the data to be operated on, called operands, and a code indicating the operation to be performed.
- A symbolic representation of an ALU and its input and output signals



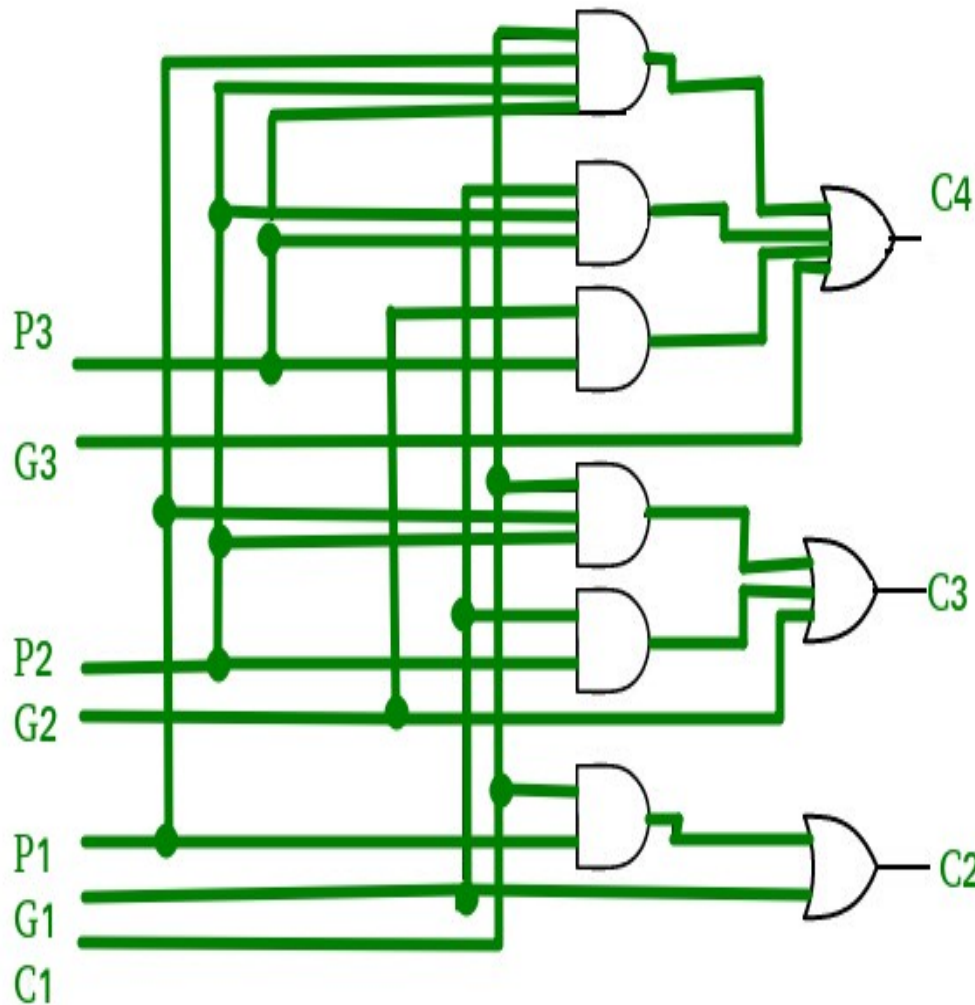
Carry Look ahead adder

- A carry-look ahead adder (CLA) or fast adder is a type of electronic adder used in digital logic. A carry-look ahead adder improves speed by reducing the amount of time required to determine carry bits.
- 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.



Carry Look ahead adder

- The implementation of three Boolean functions for each carry output (C_2, C_3 and C_4) for a carry look-ahead carry generator shown in diagram below figure.



Advantages –

- The propagation delay is reduced.
- It provides the fastest addition logic.

Disadvantages –

- The Carry Look-ahead adder circuit gets complicated as the number of variables increase.
- The circuit is costlier as it involves more number of hardware.

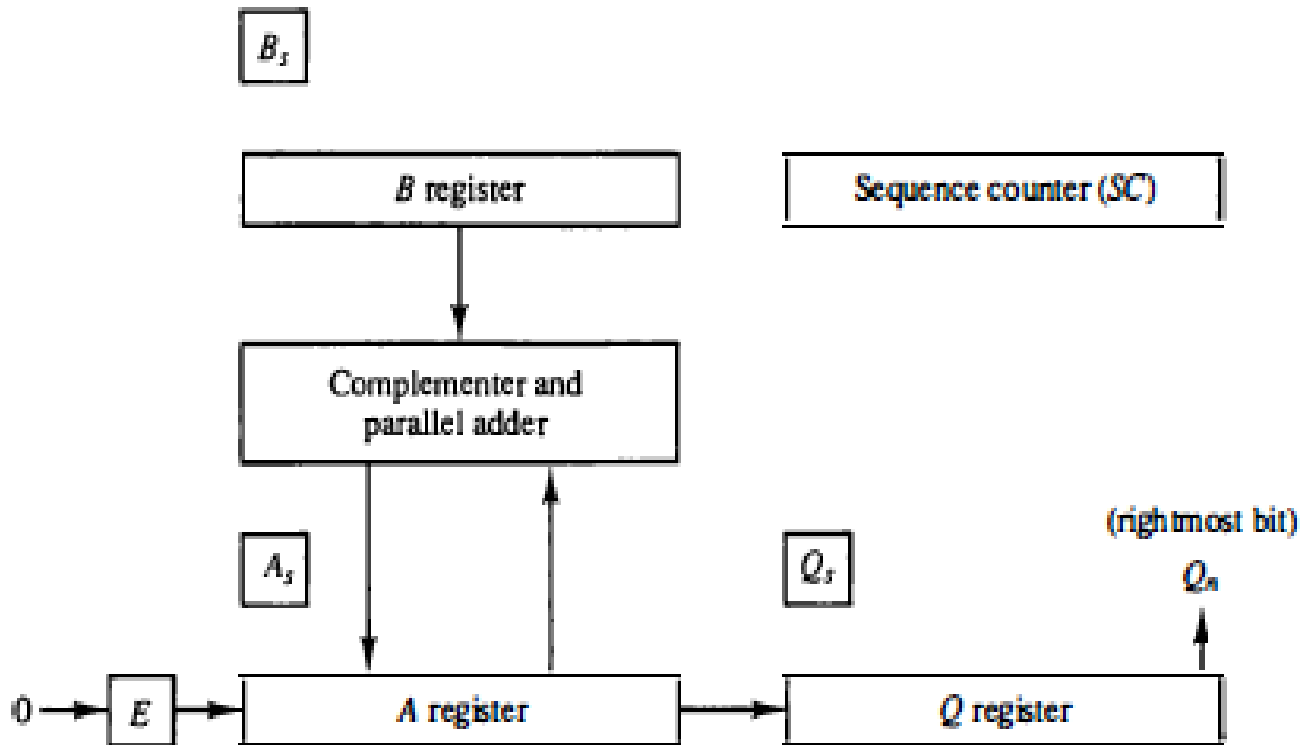
Multiplication: Signed operand multiplication

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

Multiplication for Signed magnitude data

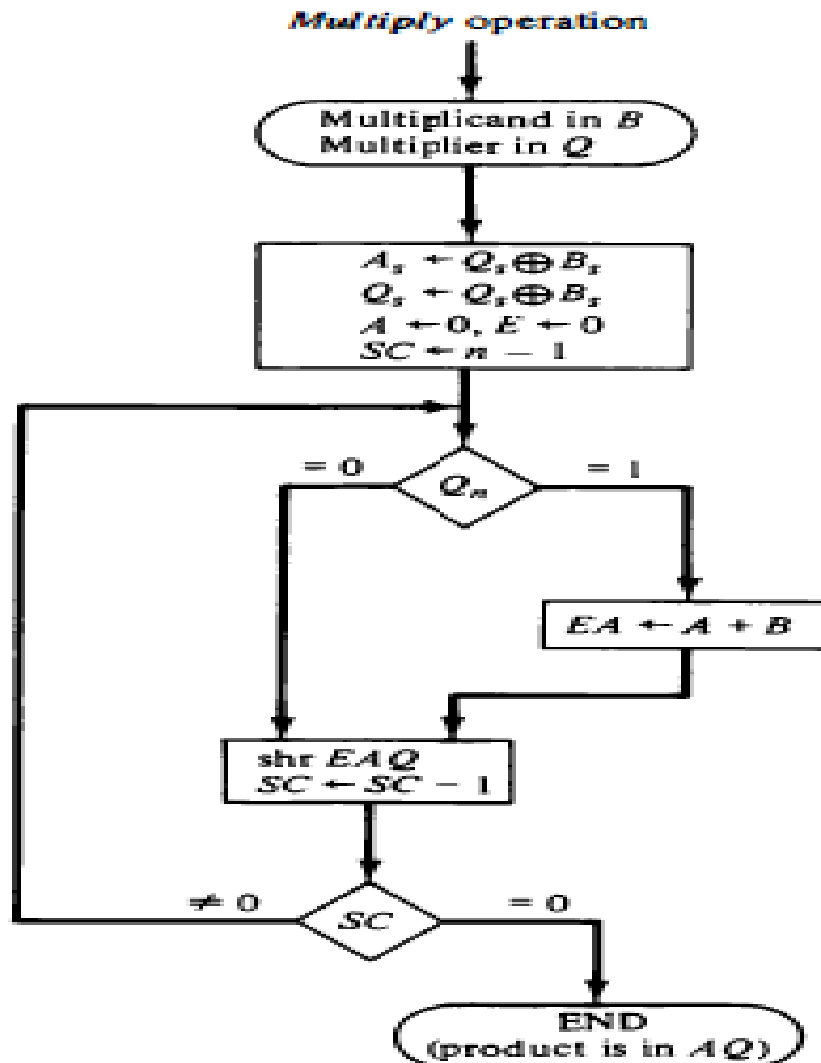
Hardware for multiply operation

Hardware for multiply operation.



Multiplication for Signed magnitude data

Flowchart for multiply operation.



Multiplication for Signed magnitude data

Example

Numerical Example for Binary Multiplier

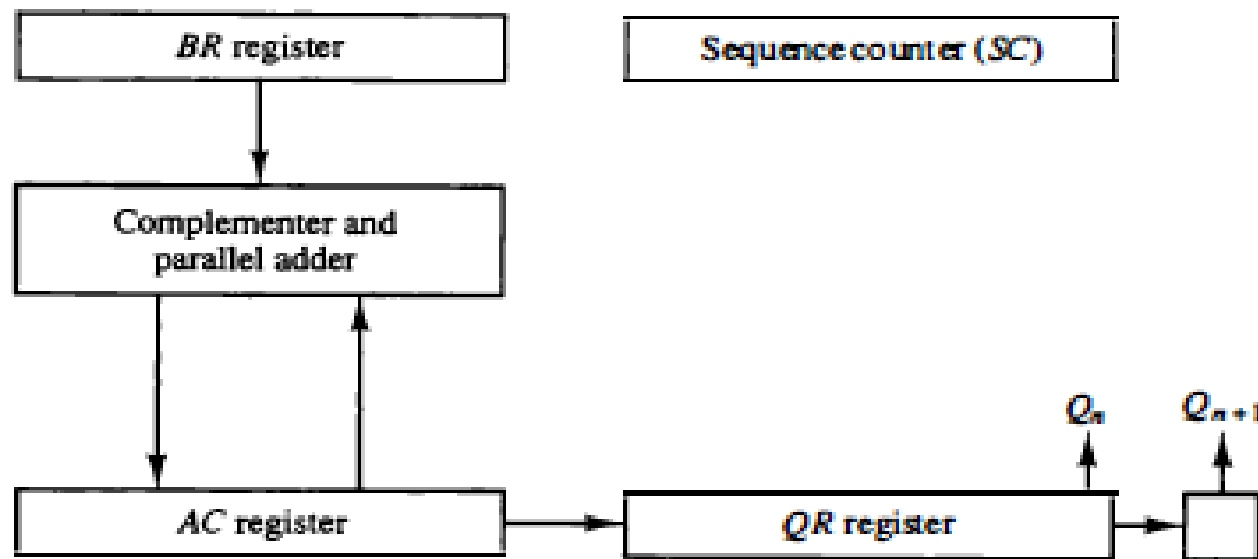
Multiplicand $B = 10111$	E	A	Q	SC
Multiplier in Q	0	00000	10011	101
$Q_n = 1$; add B		<u>10111</u>		
First partial product	0	10111		
Shift right EAQ	0	01011	11001	100
$Q_n = 1$; add B		<u>10111</u>		
Second partial product	1	00010		
Shift right EAQ	0	10001	01100	011
$Q_n = 0$; shift right EAQ	0	01000	10110	010
$Q_n = 0$; shift right EAQ	0	00100	01011	001
$Q_n = 1$; add B		<u>10111</u>		
Fifth partial product	0	11011		
Shift right EAQ	0	01101	10101	000
Final product in $AQ = 0110110101$				

Booth Multiplication Algorithm

Booth Multiplication Method

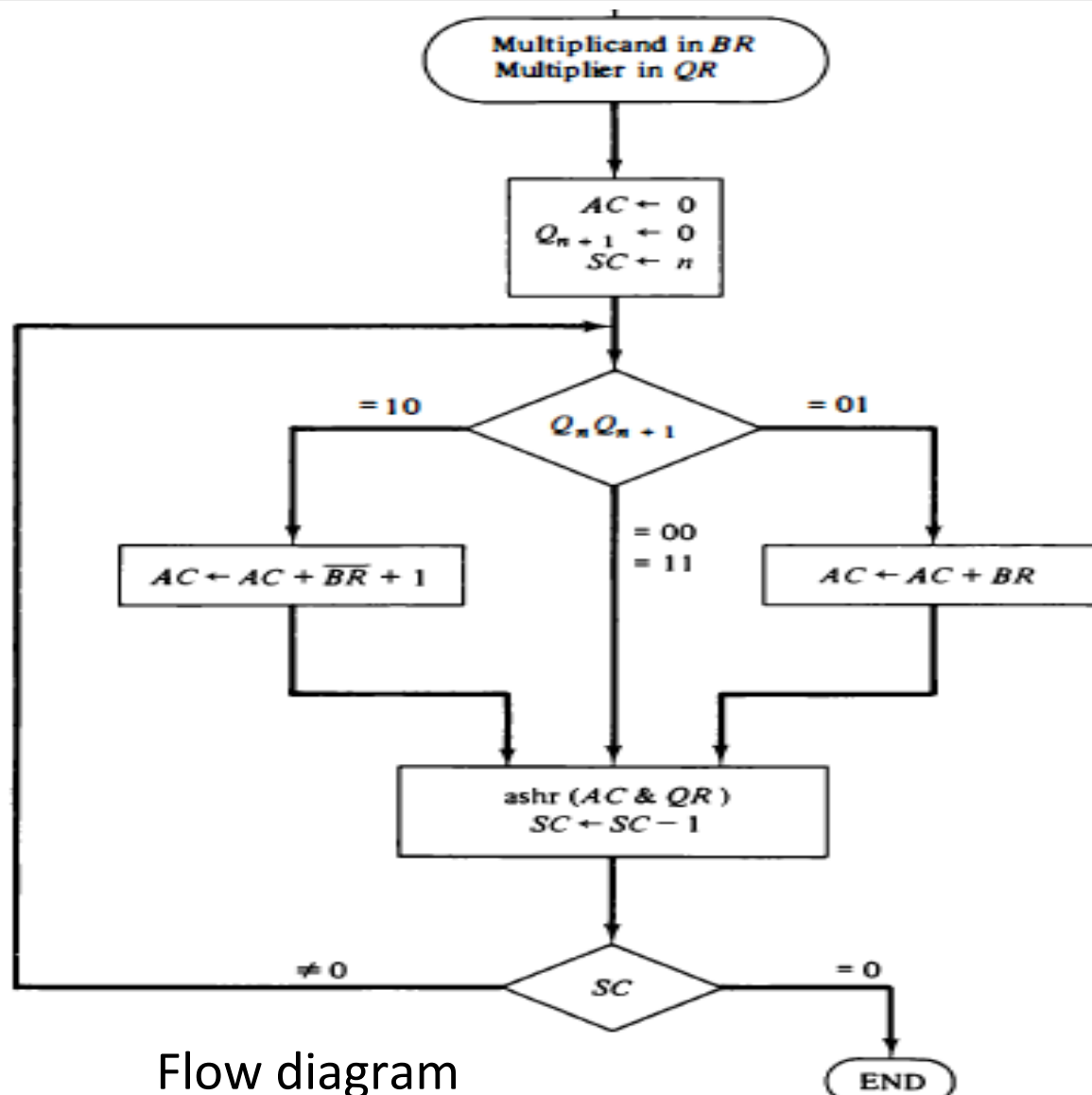
Booth algorithm gives a procedure for multiplying binary integers in signed-2's complement representation.

Hardware for Booth algorithm.



Hardware for booth algorithm

Booth Multiplication Algorithm



Flow diagram

Booth Multiplication Algorithm

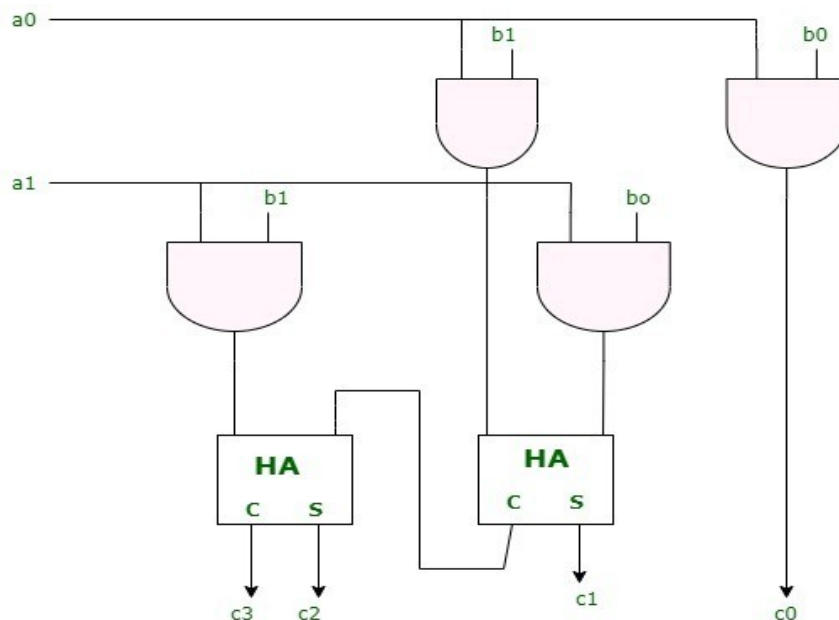
Example

Example of Multiplication with Booth Algorithm

$Q_n Q_{n+1}$	$BR = 10111$ $\overline{BR} + 1 = 01001$	AC	QR	Q_{n+1}	SC
	Initial	00000	10011	0	101
1 0	Subtract BR	<u>01001</u> 01001			
	ashr	00100	11001	1	100
1 1	ashr	00010	01100	1	011
0 1	Add BR	<u>10111</u> 11001			
	ashr	11100	10110	0	010
0 0	ashr	11110	01011	0	001
1 0	Subtract BR	<u>01001</u> 00111			
	ashr	00011	10101	1	000

Array Multiplier

- An **array multiplier** is a digital combinational circuit used for multiplying two binary numbers by employing an array of full adders and half adders.
- This array is used for the nearly simultaneous addition of the various product terms involved.
- The multiplication of two 2-bit numbers as shown in figure. The multiplicand bits are b_1 and b_0 , the multiplier bits are a_1 and a_0 , and the product is -



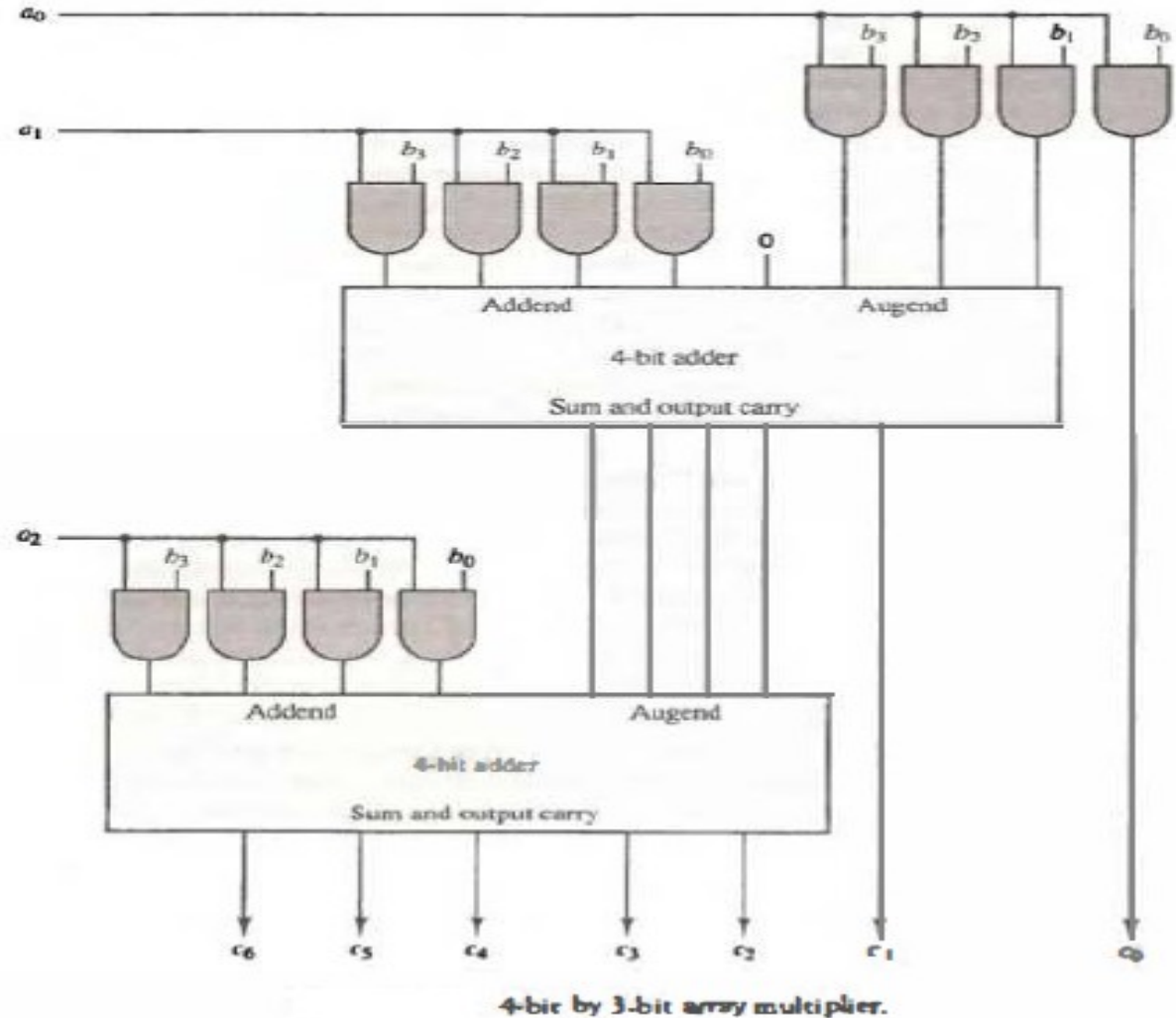
b_1	b_0	
a_1	a_0	
a_0b_1	a_0b_0	
a_1b_1	a_1b_0	
c_3	c_2	c_1
		c_0

Array Multiplier

- 4 bit Array multiplier $B = b_3 b_2 b_1 b_0$ and $A = a_2 a_1 a_0$.

For j multipliers &
 k multiplicands bits ,
 we need

- $j \times k$ AND gates
- $(j-1) \times k$ bit adders
 produce
- A product of $j + k$ bits



➤ Division and logic operations

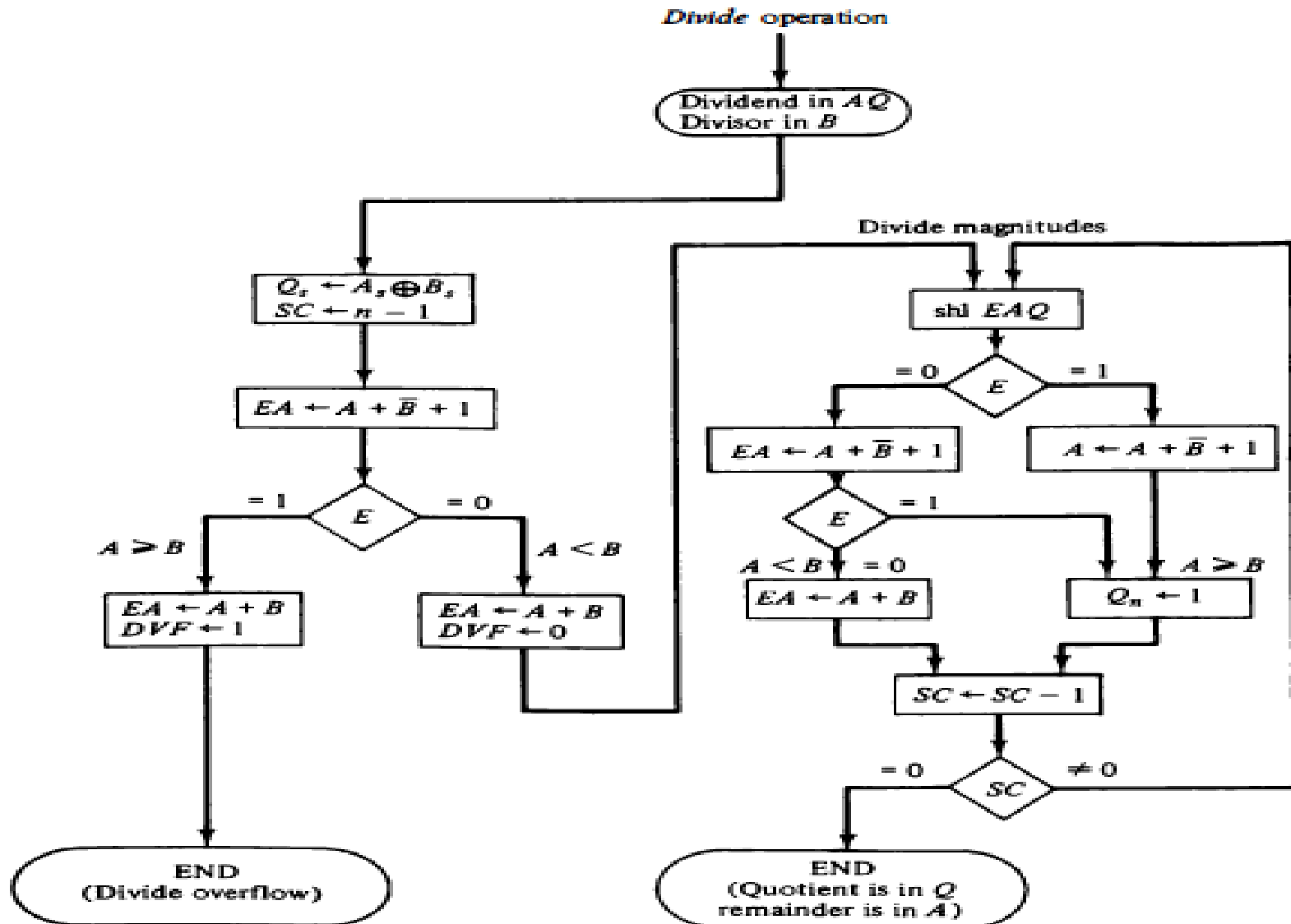
Division

- Division of two fixed-point binary numbers in signed magnitude representation is performed with paper and pencil by a process of successive compare, shift and subtract operations.
- Binary division is much simpler than decimal division because here the quotient digits are either 0 or 1 and there is no need to estimate how many times the dividend or partial remainder fits into the divisor. . _

		0 0 0 0 1 0 1 0 1	Quotient
Divisor	1 1 0 1	1 0 0 0 1 0 0 1 0	Dividend
		- 1 1 0 1	
		1 0 0 0 0	
		- 1 1 0 1	
		1 1 1 0	
		- 1 1 0 1	
		1	Remainder

➤ Division Algorithm

Flowchart for divide operation.



➤ Division Algorithm

Example

Divisor $B = 10001$,

$\bar{B} + 1 = 01111$

	E	A	Q	SC
Dividend:		01110	00000	5
shl EAQ	0	11100	00000	
add $\bar{B} + 1$		01111		
$E = 1$	1	01011		
Set $Q_n = 1$	1	01011	00001	4
shl EAQ	0	10110	00010	
Add $\bar{B} + 1$		01111		
$E = 1$	1	00101		
Set $Q_n = 1$	1	00101	00011	3
shl EAQ	0	01010	00110	
Add $\bar{B} + 1$		01111		
$E = 0$; leave $Q_n = 0$	0	11001	00110	
Add B		10001		
Restore remainder	1	01010		2
shl EAQ	0	10100	01100	
Add $\bar{B} + 1$		01111		
$E = 1$	1	00011		
Set $Q_n = 1$	1	00011	01101	1
shl EAQ	0	00110	11010	
Add $\bar{B} + 1$		01111		
$E = 0$; leave $Q_n = 0$	0	10101	11010	
Add B		10001		
Restore remainder	1	00110	11010	0
Neglect E				
Remainder in A :		00110		
Quotient in Q :			11010	

Example of binary division with digital hardware.

➤ Floating point arithmetic operation

FLOATING POINT Addition & Subtraction

A floating point number in computer registers consists of two parts: a mantissa m and an exponent e . The two parts represent a number obtained from multiplying m times a radix r raised to the value of e ; thus $m \times r^e$

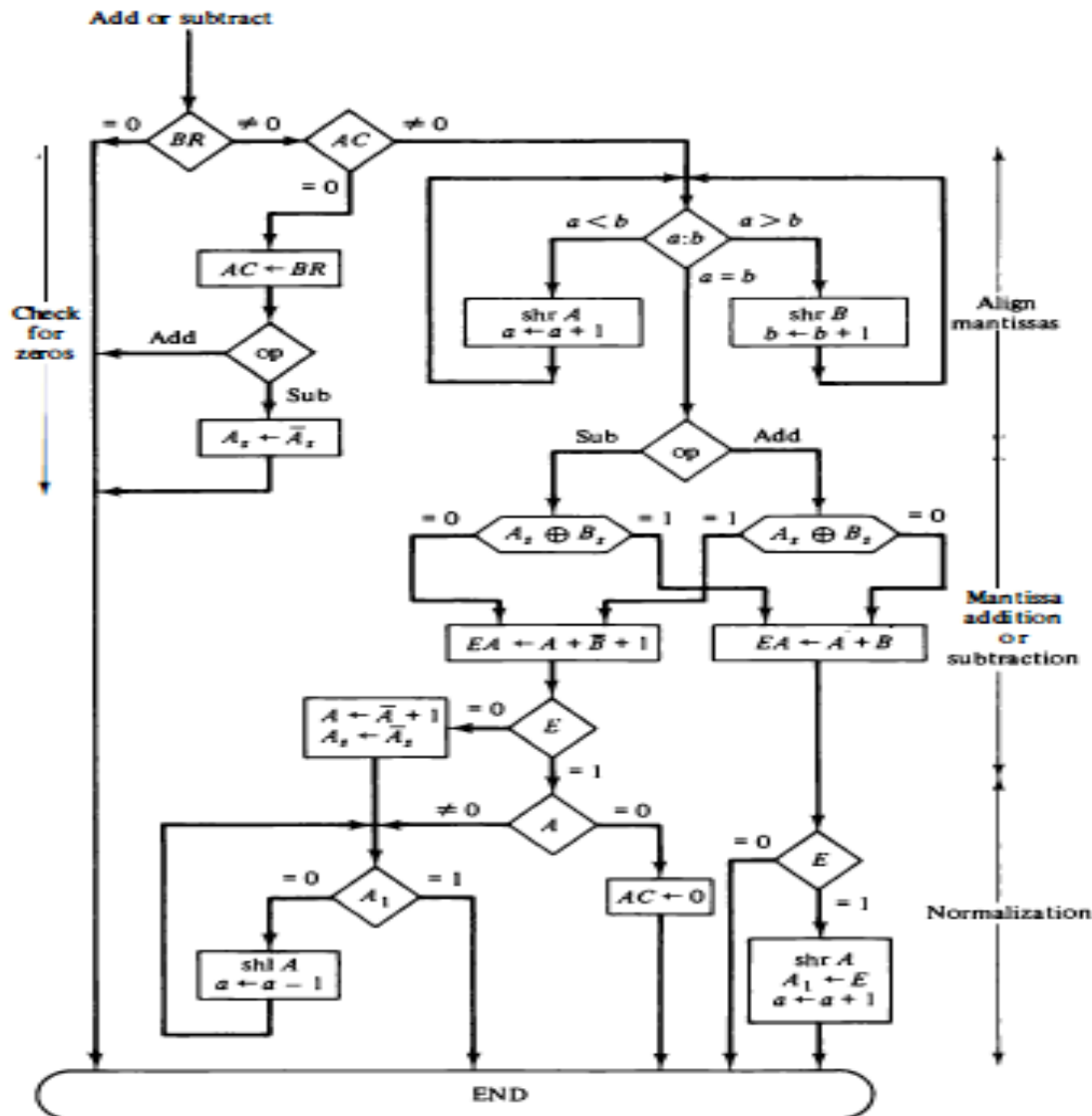
Example $.53725 \times 10^3$

During addition or subtraction, the two floating-point operands are in AC and BR. The sum or difference is formed in the AC. The algorithm can be divided into four consecutive parts:

- Check for zeros.
- Align the mantissas.
- Add or subtract the mantissas.
- Normalize the result.

FLOATING POINT Addition & Subtraction

Floating-Point Arithmetic Operations



➤ Floating point arithmetic operation

FLOATING POINT Multiplication

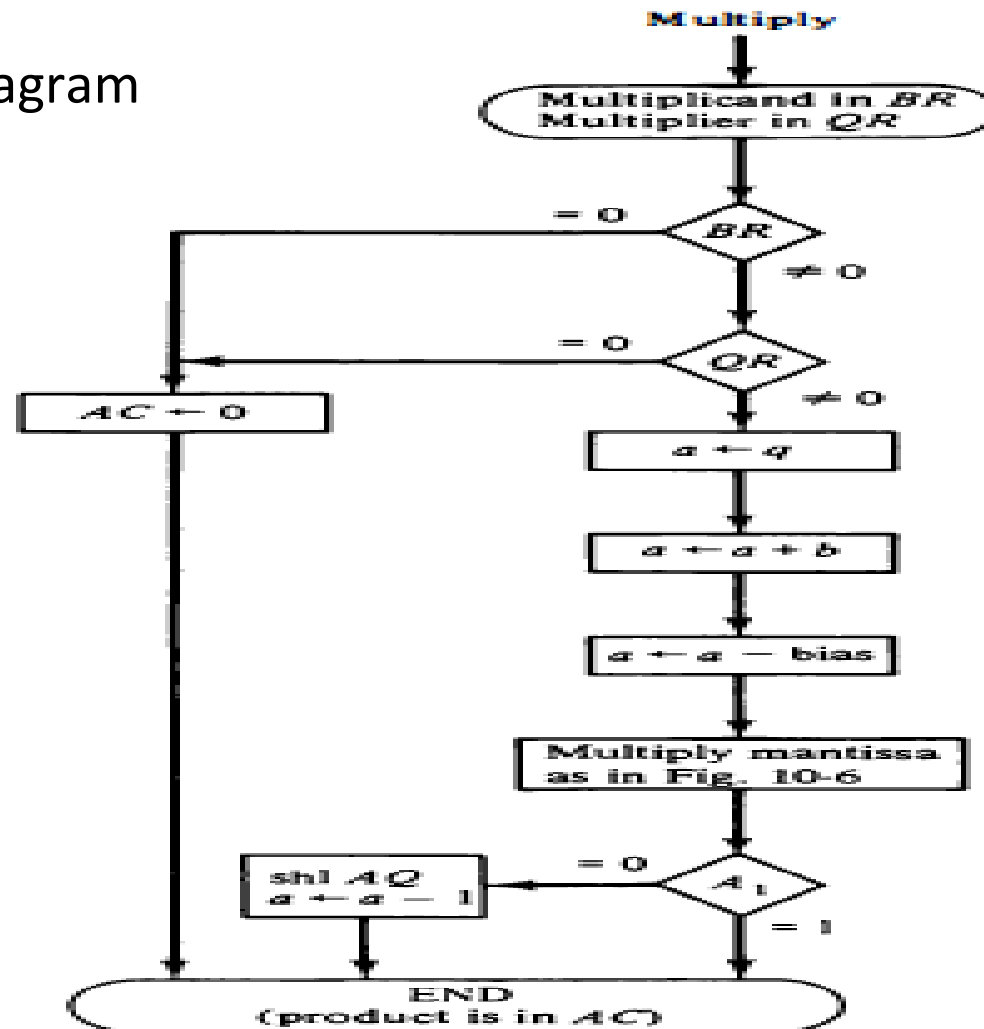
The multiplication of two floating-point numbers requires that we multiply the mantissas and add the exponents. The multiplication of the mantissas is performed in the same way as in fixed-point to provide a double-precision.

The multiplication algorithm can be subdivided into four parts:

- Check for zeros.
- Add the exponents.
- Multiply the mantissas.
- Normalize the product.

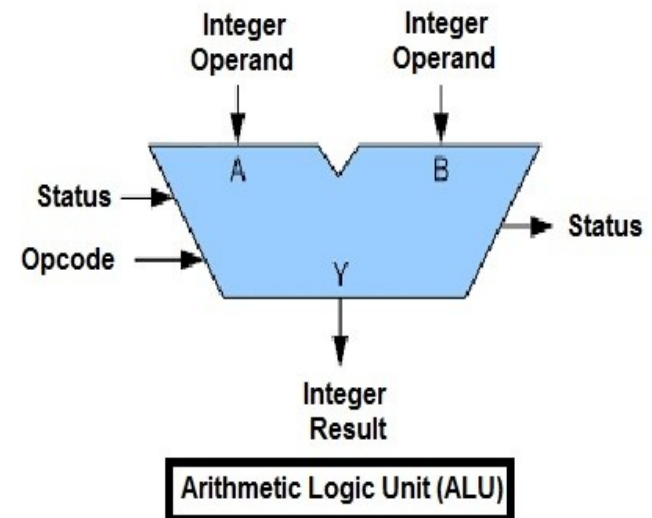
FLOATING POINT Multiplication

Flow diagram



➤ Arithmetic & logic unit design

- Inside a computer, there is an Arithmetic Logic Unit (ALU), which is capable of performing logical operations (e.g. AND, OR, Ex-OR, Invert etc.)
- In addition to the arithmetic operations (e.g. Addition, Subtraction etc.).
- The control unit supplies the data required by the ALU from memory, or from input devices, and directs the ALU to perform a specific operation based on the instruction fetched from the “calculator” portion of the computer. Different operation as carried out by ALU can be categorized as follows
 - **Logical operations**
 - **Bit-Shifting Operations**
 - **Arithmetic operations**



➤ Arithmetic & logic unit design

1-bit ALU Design

- Construct a simple ALU that performs a arithmetic operation (1 bit addition) and does 3 logical operations namely AND, NOR and XOR as shown below.
- The multiplexer selects only one operation at a time. The operation selected depends on the selection lines of the multiplexer as shown in the truth table.

➤ Arithmetic & logic unit design

1-bit ALU Design

Inputs		Outputs
<i>M1</i>	<i>M0</i>	<i>Operation</i>
0	0	SUM
1	0	AND
0	1	OR
1	1	XOR

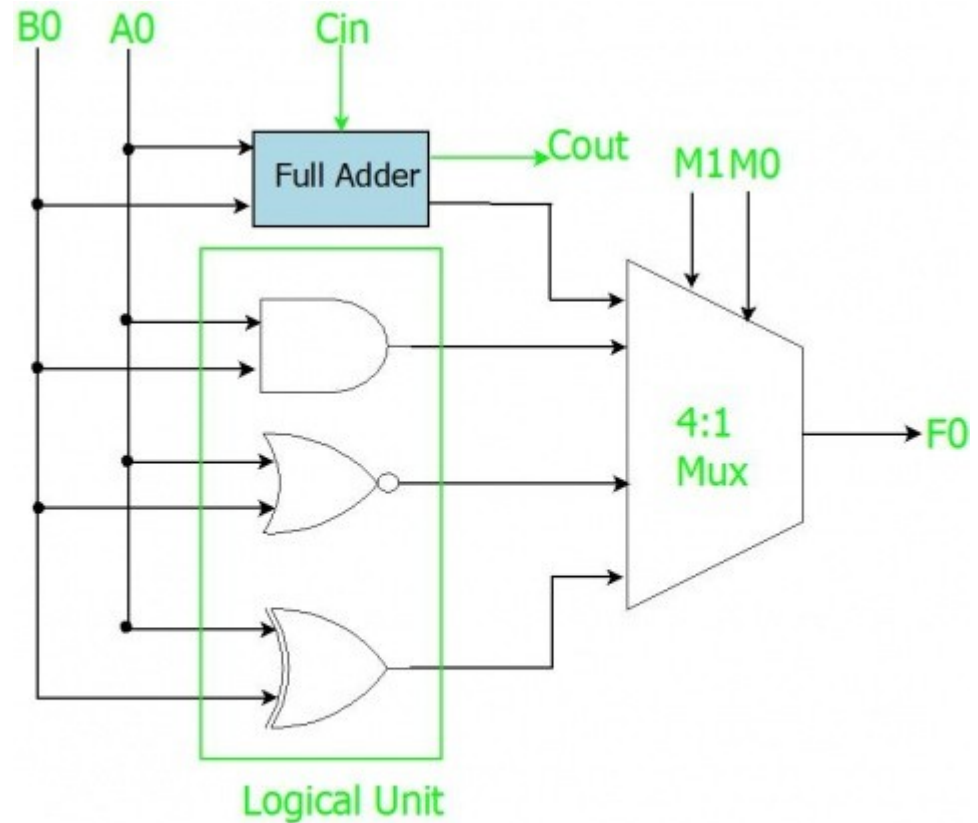


Figure: 1 bit ALU

➤ IEEE Standard for Floating Point Numbers

- The IEEE Standard for Floating-Point Arithmetic (IEEE 754) is a technical standard for floating-point computation which was established in 1985 by the **Institute of Electrical and Electronics Engineers (IEEE)**.

IEEE 754 has 3 basic components:

The Sign of Mantissa –

0 represents a positive number while 1 represents a negative number.

The Biased exponent –

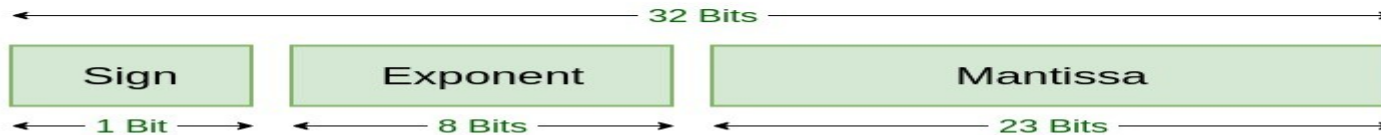
A bias is added to the actual exponent in order to get the stored exponent.

The Normalized Mantissa –

A normalized mantissa is one with only one 1 to the left of the decimal.

➤ IEEE Standard for Floating Point Numbers

IEEE 754 numbers are divided into two based on the above three components: single precision and double precision.



Single Precision
IEEE 754 Floating-Point Standard



Double Precision
IEEE 754 Floating-Point Standard

Faculty Video Links, You tube Courses Details

You tube/other Video Links

- <https://www.youtube.com/watch?v=vcvgvqnH7GA>
- <https://www.youtube.com/watch?v=U62iP8RkZIk>
- <https://www.youtube.com/watch?v=6ToR6vuRb3M>
- <https://www.youtube.com/watch?v=9nkCLdhLDZk>
- <https://www.youtube.com/watch?v=0HiGruw9VcQ>

Daily Quiz

- The result of $0 - 1$ in binary is _____?
- Which flag indicates the number of 1 bit that results from an operation?
- Define IEEE 754 Standard.
- Draw hardware diagram of booth algorithm.
- Draw array multiplier of 2 bit, $A=a_1a_0$, $B=b_1 b_0$.

Weekly Assignment

1. Design array multiplier for $b_1 b_0 \times a_1 a_0$ and $b_3 b_2 b_1 b_0 \times a_2 a_1 a_0$ and also write down the formula for no. of AND gate calculations.
2. Show the contents of registers E, A, Q and SC during the process of division 00001111 by 0011.
3. Show the hardware diagram & flow chart of Booth algorithm for multiplying two numbers.
4. Demonstrate the Carry Look Ahead adder with suitable diagram and advantages.
5. Define IEEE 754 standard for floating point representation with suitable example.

$$(23.175)_{10} \quad \text{b) } (0.6128)_{10} \quad \text{c) } (1000.010101)_2$$

- 1. The 'heart' of the processor which performs many different operations .
a) Arithmetic and logic unit b) Motherboard c) Control Unit d) Memory
- 2. Which of the following is not a bitwise operator?
a) | b) ^ c) . d) <<
- 3. The sign magnitude representation of -1 is _____
a) 0001 b) 1110 c) 1000 d) 1001
- 4. IEEE stands for _____
- 5. The ALU gives the output of the operations and the output is stored in the _____
a) Memory Devices b) Registers c) Flags d) Output Unit
- 6 The IEEE standard followed by almost all the computers for floating point arithmetic
a) IEEE 260 b) IEEE 488 c) IEEE 754 d) IEEE 610
- 7. Which of the following is often called the double precision format?
a) 64-bit b) 8-bit c) 32-bit d) 128-bit

Solution 1 a. 2c. 3 d. 4 Institute of Electrical and electronics engineers. 5 b 6 c 7a

Old Question Papers

111111 111111 1 1

■ ▼ ■ ■ ■ ■

(Following Paper ID and Roll No. to be filled in your Answer Book)

PAPER ID : 1067-NEW

Roll No.

--	--	--	--	--	--	--	--	--	--

B. Tech.

(SEM. IV) EXAMINATION, 2008-09

COMPUTER ORGANIZATION

Time : 3 Hours]

[Total Marks : 100

- Note:**
- (1) Attempt **ALL** questions.
 - (2) All questions carry **equal** marks.
 - (3) Be precise in your answer.
 - (4) **No Second Answer** book will be provided.

Attempt any **four** parts of the following: **5×4=20**

- (a) Represent the following conditional control statement by two register transfer statements with control function :
 if (P=1) then (R1 \leftarrow R2) else if (Q=1)
 then (R1 \leftarrow R3)
- (b) Register A holds the 8-bit binary 11011001. Determine the B operand and the logic micro-operation to be performed in order to change the value in A to :
 - (i) 01101101
 - (ii) 11111101

Old Question Papers

- (c) Design a 4-bit adder-subtractor with complete block diagram.
- (d) What do you mean by high speed adder? Discuss design of high speed adders.
- (e) Write short note on the following :
 - (i) Common Bus system
 - (ii) Bus Arbitration.
- (f) Give IEEE standard for floating point numbers.

2 Attempt any **two** parts of the following: **10×2=20**

- (a) What do you understand by hard wired control unit? Give various methods to design hardwired control unit. Describe one of the design methods for hardwired control unit with suitable diagrams.
- (b) Write short note on the following :
 - (i) Multiple-bus organization
 - (ii) Micro-programmed control unit.
- (c) (i) What do you mean by wide-branch addressing? Explain with example.
 - (ii) Differentiate between a microprocessor and a microprogram? Is it possible to design a microprocessor without a microprogram? - Explain.

3 Attempt any **two** parts of the following: **10×2=20**

- (a) (i) A computer has 32-bit instructions and 12-bit addresses. If there are 250 two-address instructions, how many one-address instructions can be formulated ?

Expected Questions for University Exam

- Explain CLA and design 4bit CLA consists of 3 level of logic.
- Draw the flow chart for multiplying algorithm of sign magnitude data.
- Show the contents of registers E, A, Q & SC during the process of division of 10100011 by 1011
- Explain the IEEE 754 for floating point no. with examples
- Show the flow diagram & multiplication process using Booth's Algorithm for $(-4) \times (+3)$

In previous slides we discuss in details

- Carry Look ahead adder.
- Multiplication: Signed operand multiplication,
- Booths algorithm and array multiplier.
- Division and logic operations.
- Floating point arithmetic operation,
- Arithmetic & logic unit design.
- IEEE Standard for Floating Point Numbers