Arithmetic and logic unit

Unit: 2

Computer Organization & Architecture (ACSE0305)

B Tech 3rd Sem

Pradeep Kumar
Assistant Professor
CSE
Department

CSE 3rd Semester Evaluation Scheme

EVALUATION SCHEME
SEMESTER-III

Sl. No.	Subject Codes	Subject Name	Periods			Evaluation Schemes				End Semester		Total	Credit
			L	T	P	CT	TA	TOTAL	PS	TE	PE		
WEEKS COMPULSORY INDUCTION PROGRAM													
1	AAS0301A	Engineering Mathematics III	3	1	0	30	20	50		100		150	4
2	ACSE0306	Discrete Structures	3	0	0	30	20	50		100		150	3
3	ACSE0304	Digital Logic & Circuit Design	3	0	0	30	20	50		100		150	3
4	ACSE0301	Data Structures	3	1	0	30	20	50		100		150	4
5	ACSE0302	Object Oriented Techniques using Java	3	0	0	30	20	50		100		150	3
6	ACSE0305	Computer Organization & Architecture	3	0	0	30	20	50		100		150	3
7	ACSE0354	Digital Logic & Circuit Design Lab	0	0	2				25		25	50	1
8	ACSE0351	Data Structures Lab	0	0	2				25		25	50	1
9	ACSE0352	Object Oriented Techniques using Java Lab	0	0	2				25		25	50	1
10	ACSE0359	Internship Assessment-I	0	0	2				50			50	1
11	ANC0301/ ANC0302	Cyber Security*/ Environmental Science*(Non	2	0	0	30	20	50		50		100	0

Syllabus

UNIT-I	Introduction	8 Hours
Computer Organization and Architecture, Functional units of digital system and their interconnections, buses, bus architecture, types of buses and bus arbitration and it's types. Register, bus and memory transfer. Process or organization, general registers organization, stack organization and address in g modes.		
UNIT-II	ALU Unit	8 Hours
Arithmetic and logic unit: Lookahead carries adders. Multiplication: Signed operand multiplication, Booth's algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation, Arithmetic & logic unit design. IEEE Standard for Floating Point Numbers.		
UNIT-III	Control Unit	8 Hours
Control Unit: Instruction types, formats, instruction cycles and sub cycles (fetch and execute etc.), micro-operations, execution of a complete instruction. Program Control, Reduced Instruction Set Computer, Complex Instruction Set Computer, Pipelining. Hardwire and microprogrammed control, Concept of horizontal and vertical microprogramming, Flynn's classification.		
UNIT-IV	Memory Unit	8 Hours
Memory: Basic concept and hierarchy, semiconductor RAM memories, 2D & 2 1/2D memory organization. ROM memories. Cache memories: concept and design issues & performance, address mapping and replacement Auxiliary memories: magnetic disk, magnetic tape and optical disks Virtual memory: concept implementation, Memory Latency, Memory Bandwidth, Memory Seek Time.		
UNIT-V	Input/Output	8 Hours
Peripheral devices, I/O interface, I/O ports, Interrupts: interrupt hardware, types of interrupts and exceptions. Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct Memory Access. ,I/O channels and processors. Serial Communication: Synchronous & asynchronous communication.		

Course Objective

- The objective of this course is to understand the types of organizations, structures and functions of computer, design of arithmetic and logic unit and float point arithmetic as well as to understand the concepts of memory system, communication with I/O devices and interfaces

Course Outcome

Understand the basic structure and operation of a digital computer system.
Analyze the design of arithmetic & logic unit and understand the fixed point and floating-point arithmetic operations.
Implement control unit techniques and the concept of Pipelining
Understand the hierarchical memory system, cache memories and virtual memory.
Understand different ways of communicating with I/O devices and standard I/O interfaces.

Program Outcome

1.Engineering knowledge
2. Problem analysis
3. Design/development of solutions
4. Conduct investigations of complex problems
5. Modern tool usage
6. The engineer and society
7. Environment and sustainability
8. Ethics
9. Individual and team work
10. Communication
11. Project management and finance
12. Life-long learning

CO-PO Mapping

COMPUTER ORGANIZATION AND ARCHITECTURE (ACSE0305)

CO.K	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ACSE0305.2	2	2	2	2	1	1	-	1	1	1	1	2

Program Specific Outcome

On successful completion of graduation degree, The computer Science & Engineering graduates will be able to:

PSO1: identify, analyze real world problems and design their ethical solutions using artificial intelligence, robotics, virtual/augmented reality, data analytics, block chain technology, and cloud computing.

PSO2: design and develop the hardware sensor devices and related interfacing software systems for solving complex engineering problems.

PSO 3: understand inter-disciplinary computing techniques and to apply them in the design of advanced computing.

PSO 4: conduct investigation of complex problem with the help of technical, managerial, leadership qualities, and modern engineering tools provided by industry sponsored laboratories.

CO- PSO Mapping

COMPUTER ORGANIZATION AND ARCHITECTURE (ACSE0305)				
CO.K	**PSO1**	**PSO2**	**PSO3**	**PSO4**
ACSE0305.2	2	2	2	1

Program Educational Objectives

PEO 1: To have an excellent scientific and engineering breadth so as to comprehend, analyze, design and provide sustainable solutions for real-life problems using state-of-the-art technologies.

PEO 2: To have a successful career in industries, to pursue higher studies or to support entrepreneurial endeavors and to face the global challenges.

PEO 3: To have an effective communication skills, professional attitude, ethical values and a desire to learn specific knowledge in emerging trends, technologies for research, innovation and product development and contribution to society.

PEO 4: To have life-long learning for up-skilling and re-skilling for successful professional career as engineer, scientist, entrepreneur and bureaucrat for betterment of society.

Prerequisite and Recap

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

Content

Arithmetic and logic unit:

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

UNIT OBJECTIVE

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

Arithmetic Logic Unit (ALU)

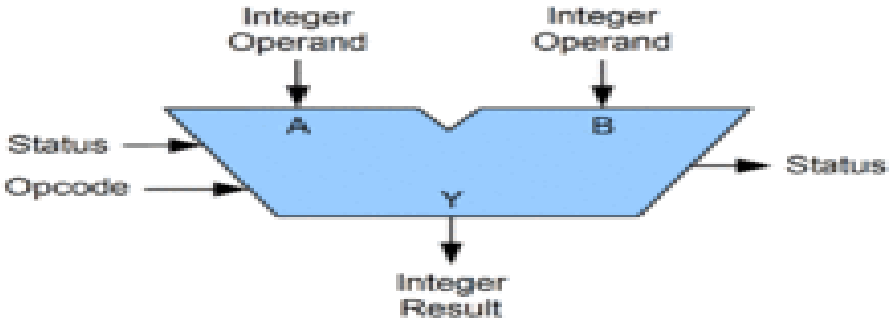
•An **arithmetic logic unit** (**ALU**) is a digital circuit used perform **arithmetic** and **logic** operations.

•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)

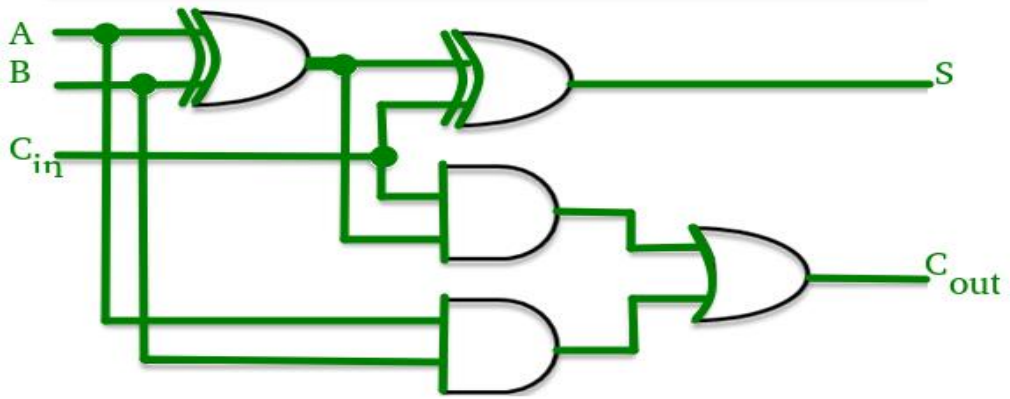
•The inputs to an ALU are the data to be operated on, called operands, and a code indicating the operation to be performed.

•A symbolic representation of an ALU and its input and output signals

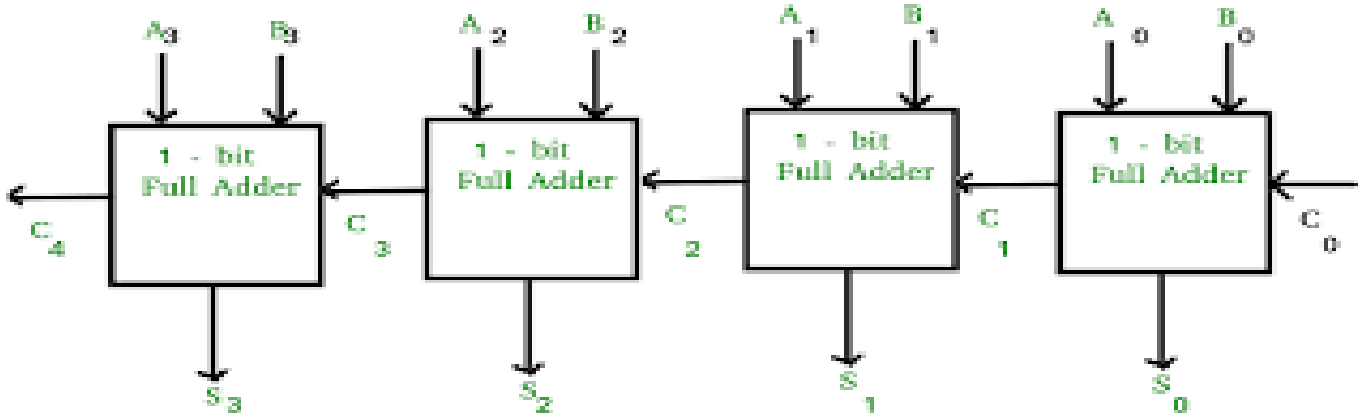


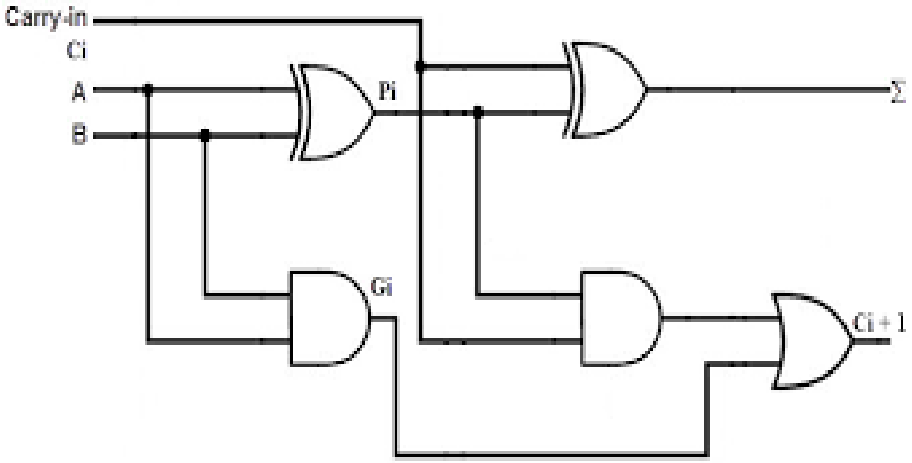
Carry Look ahead adder

- 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





$C_i = G_i + P_i \bullet C_{i-1}$

where

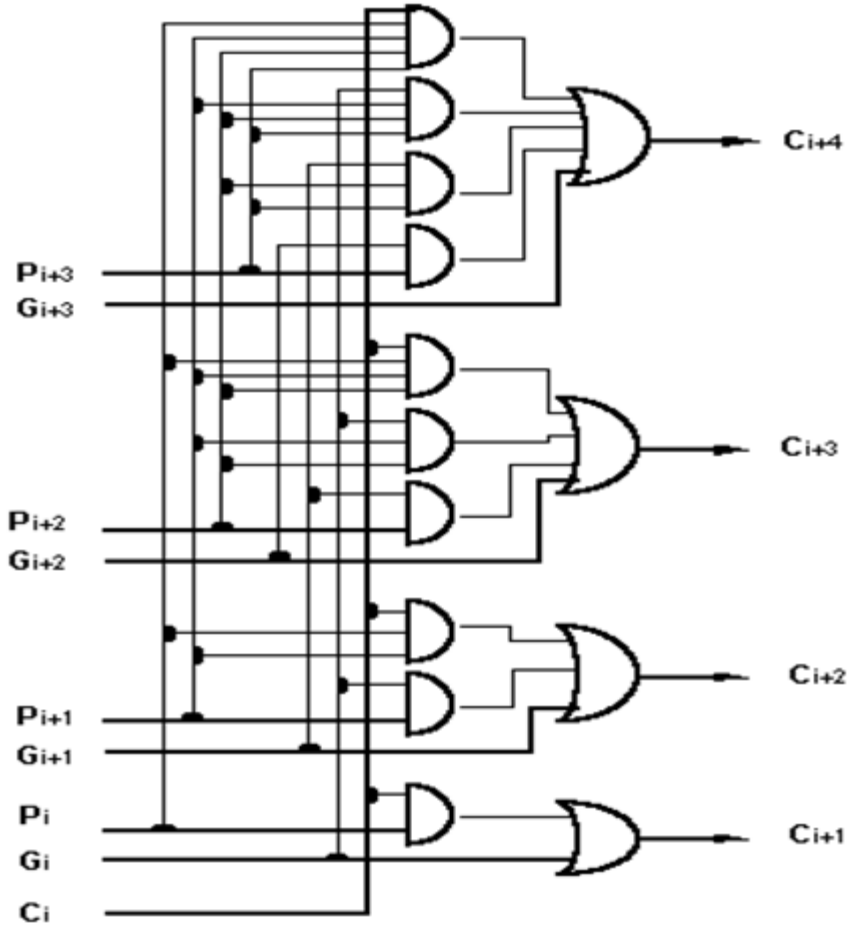
$G_i = A_i \bullet B$
$P_i = A_i + B_i$

The sum S_i is generated by:

$S_i = C_{i-1} \oplus A_i \oplus B_i$
or $C_{i-1} \oplus P_i$ (if $P_i = A_i \oplus B_i$)

Carry Look ahead adder

- The implementation of Boolean functions for each carry output (C_1,C_2,C_3 and C_4) for a carry look-ahead carry generator shown in diagram below figure.



$$c_0 = c_0,$$

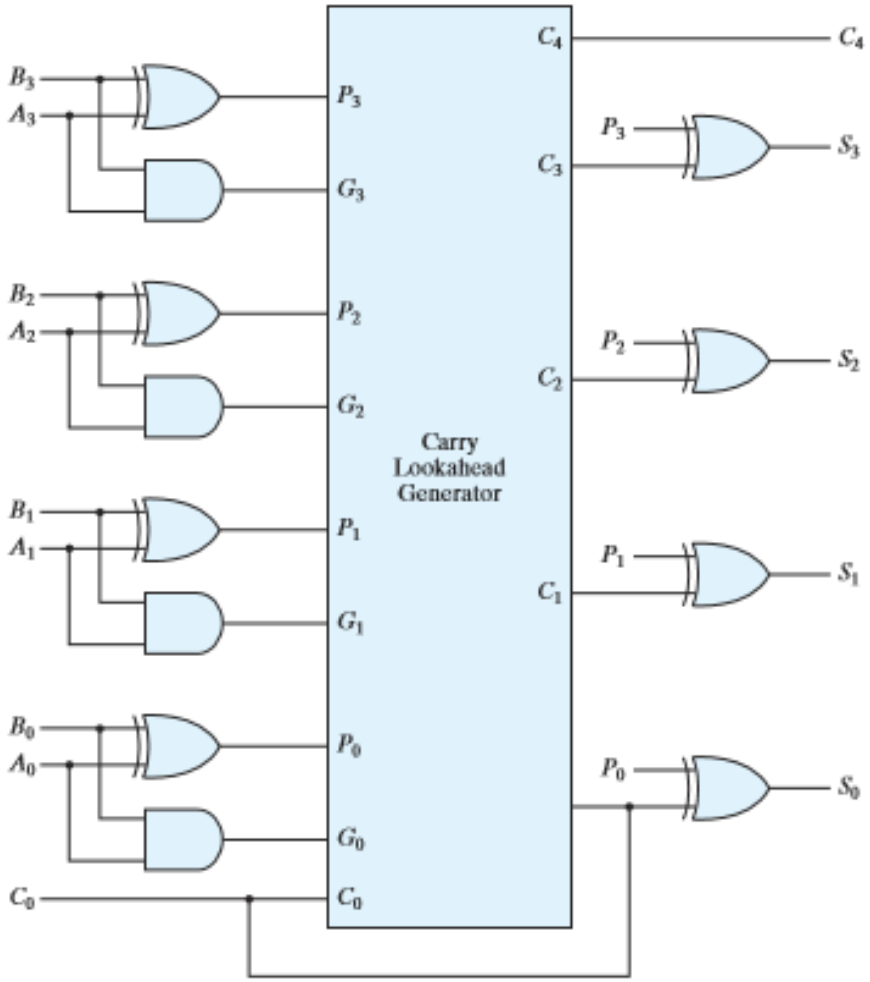
$$c_1 = G_0 + c_0 P_0,$$

$$c_2 = G_1 + G_0 P_1 + c_0 P_0 P_1,$$

$$c_3 = G_2 + G_1 P_2 + G_0 P_1 P_2 + c_0 P_0 P_1 P_2,$$

$$c_4 = G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3 + c_0 P_0 P_1 P_2 P_3.$$

Carry Look ahead adder



Advantages –

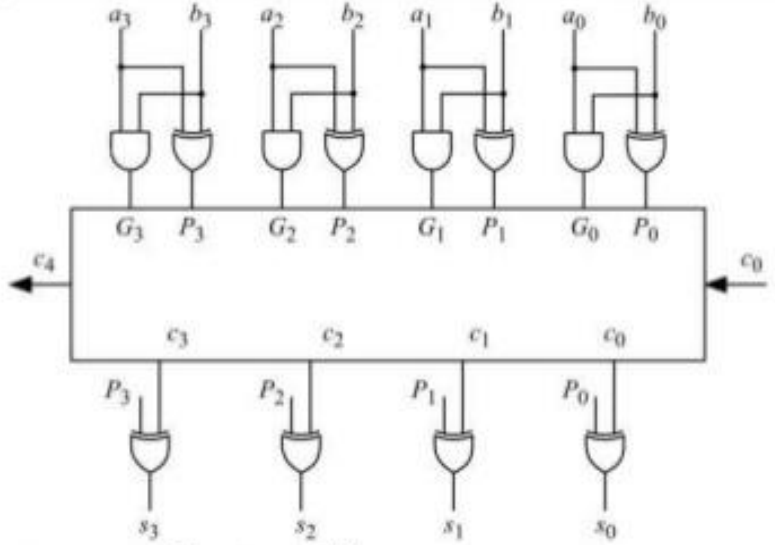
- The propagation delay is reduced.
- It provides the fastest addition logic.

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.

Carry Look ahead adder

Example 4 Bit Adder



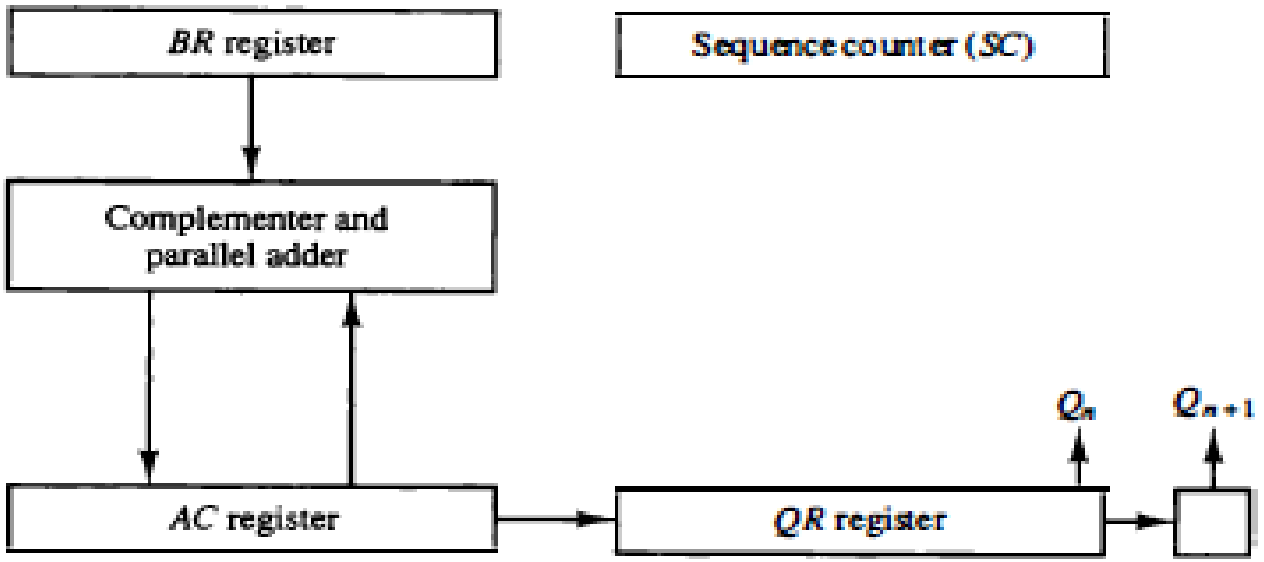
$$c_1 = G_0 + c_0 P_0,$$
$$c_2 = G_1 + G_0 P_1 + c_0 P_0 P_1,$$
$$c_3 = G_2 + G_1 P_2 + G_0 P_1 P_2 + c_0 P_0 P_1 P_2,$$
$$c_4 = G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3 + c_0 P_0 P_1 P_2 P_3$$

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

Booth's Algorithm

Multiplier

Strings of 0's: No addition; Simply shifts

Strings of 1's: String of 1's from m_p to m_q: $2^{p+1} - 2^q$

Example

001110 (14) -> p = 3, q = 1

001110 = $2^{3+1} - 2^1$

M * 14 = $M2^4 - M2^1$

Algorithm

[1] Subtract multiplicand for the first least significant 1 in a string of 1's in the multiplier

[2] Add multiplicand for the first 0 after the string of 1's in the multiplier

[3] Partial Product does not change when the multiplier bit is identical to the previous bit

$110010 = -2^4 + 2^2 - 2^1 = -16 + 4 - 2 = -14$

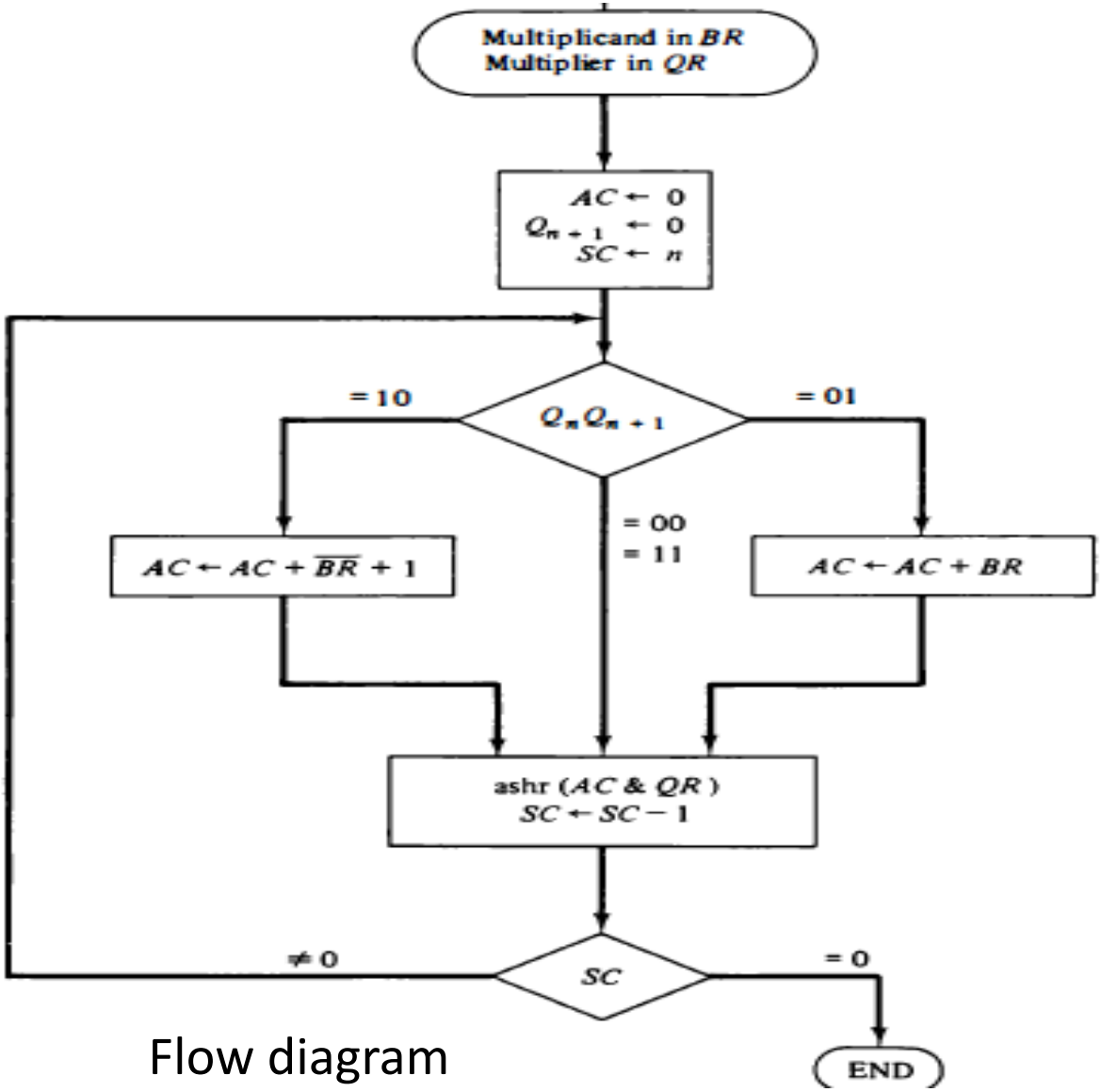
subtract 2^4 Add 2^2 subtract 2

Isha Padhy,CSE Dept, CBIT

-Algorithm is used for multiplying binary integers in signed 2's compl.

- $m^p:2^p$
- multiply multiplicand M by $14(2^{3+1}-2^1)$[multiplier]
- Algorithm also requires checking multiplier bits and shifting of partial product, but prior to shifting following steps may be done:

Booth Multiplication Algorithm



Flow diagram

Booth Multiplication Algorithm

Example

Example of Multiplication with Booth Algorithm

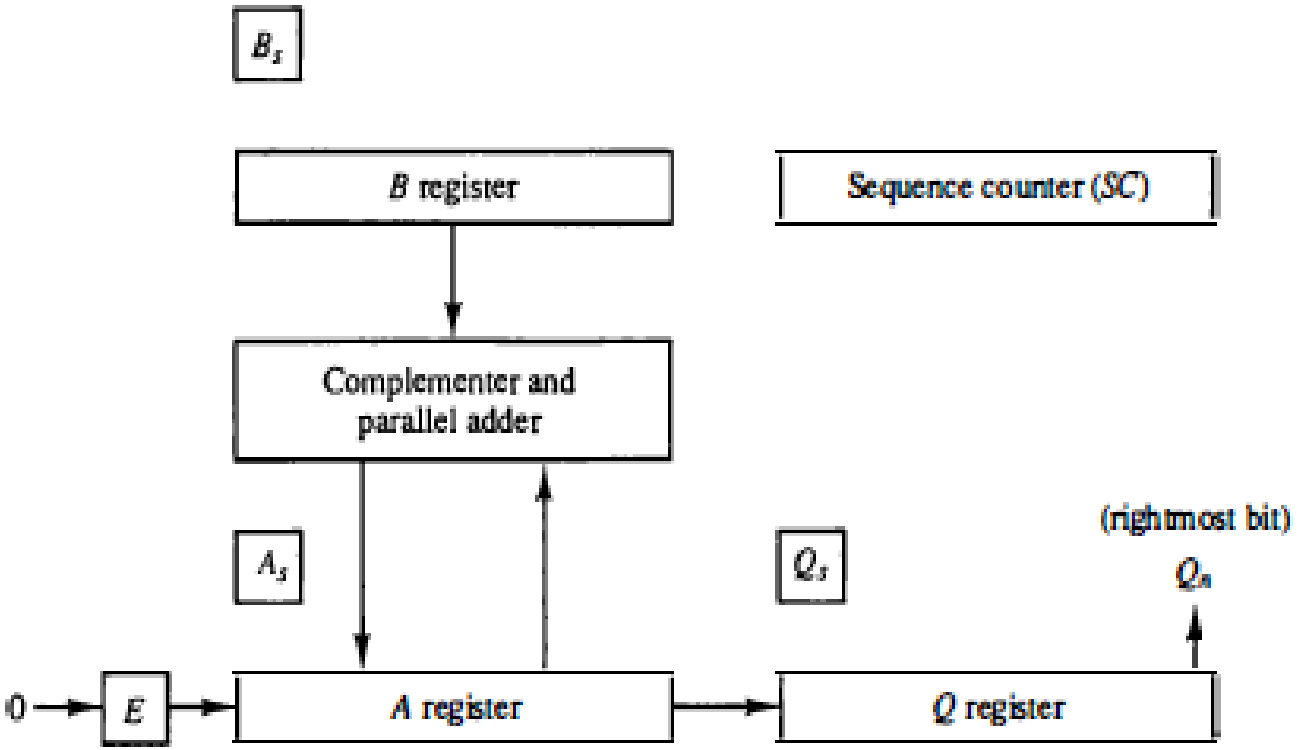
$Q_n Q_{n+1}$	$BR = 10111$ $\overline{BR} + 1 = 01001$	AC	QR	Q_{n+1}	SC
	Initial	00000	10011	0	101
1 0	Subtract BR	01001			
		01001			
	ashr	00100	11001	1	100
1 1	ashr	00010	01100	1	011
0 1	Add BR	10111			
		11001			
	ashr	11100	10110	0	010
0 0	ashr	11110	01011	0	001
1 0	Subtract BR	01001			
		00111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

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.

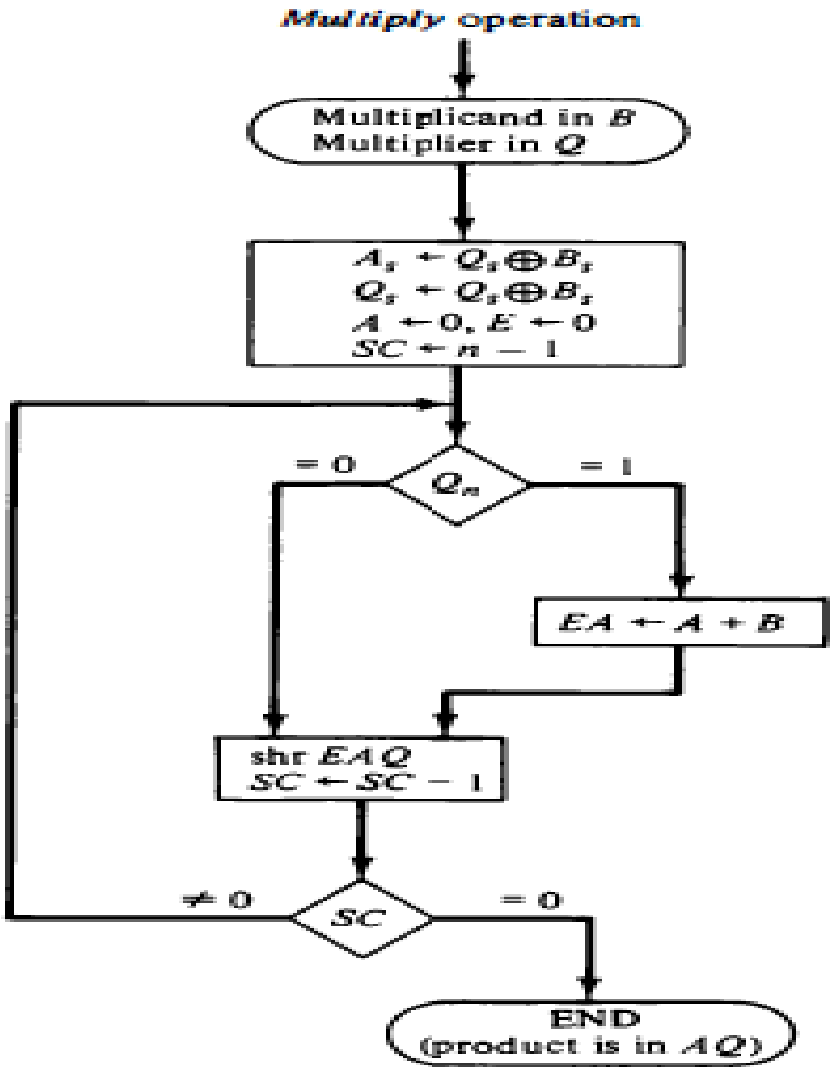
Multiplication for Signed magnitude data

Hardware for multiply operation.



Multiplication for Signed magnitude data

Flowchart for multiply operation.



Multiplication for Signed magnitude data

Example

Numerical Example for Binary Multiplier

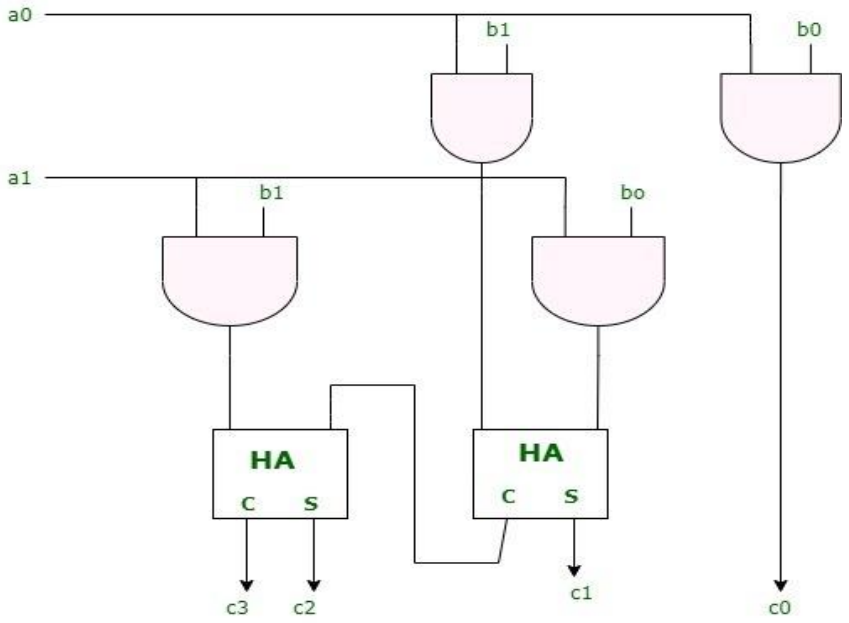
Multiplicand $B = 10111$	E	A	Q	SC
Multiplier in Q	0	00000	10011	101
$Q_n = 1$; add B		10111		
First partial product	0	10111		
Shift right EAQ	0	01011	11001	100
$Q_n = 1$; add B		10111		
Second partial product	1	00010		
Shift right EAQ	0	10001	01100	011
$Q_n = 0$; shift right EAQ	0	01000	10110	010
$Q_n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

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 b1 and b0, the multiplier bits are a1 and a0, and the product is -



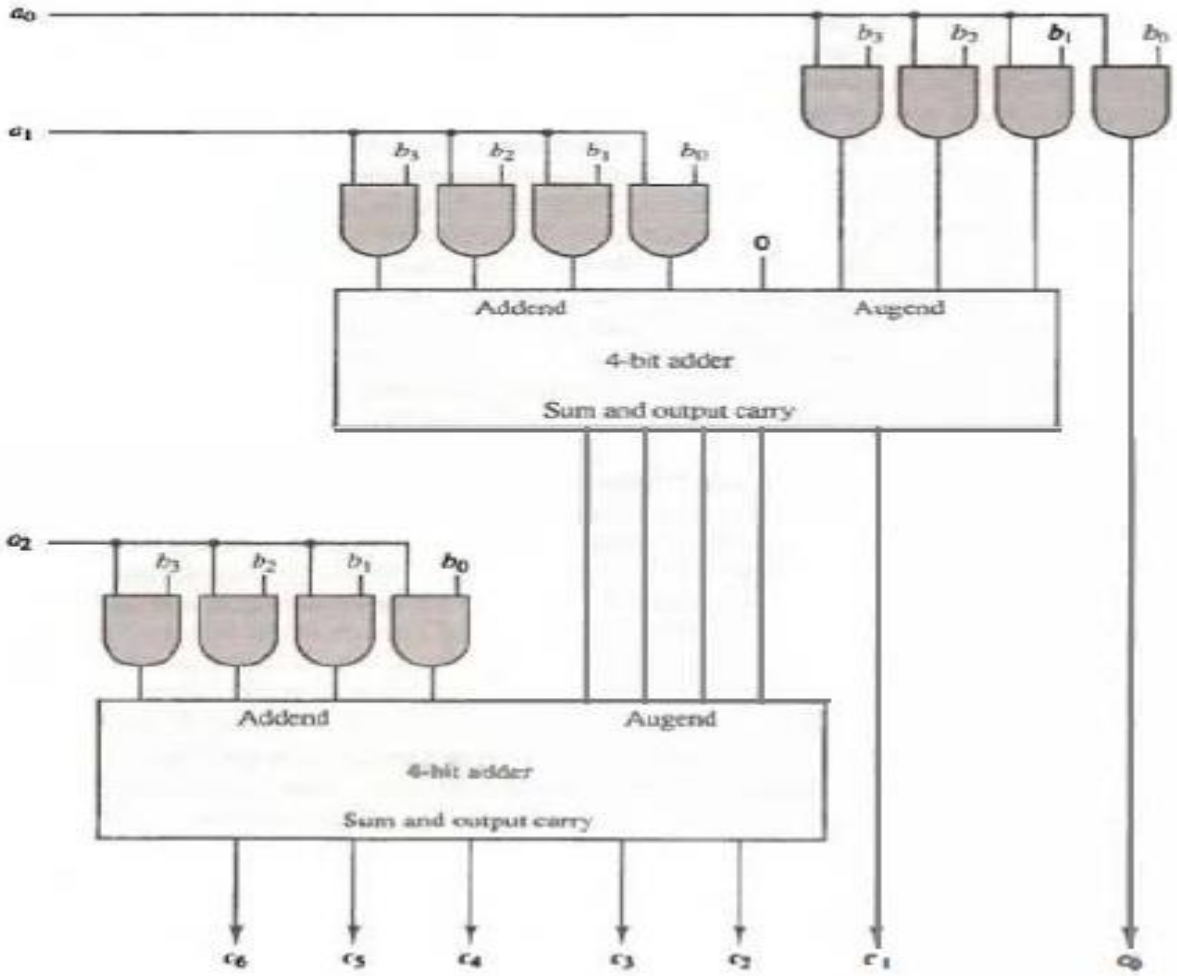
		b1	b0
		a1	a0
		a0b1	a0b0
	a1b1	a1b0	
c3	c2	c1	c0

Array Multiplier

- **4 bit Array multiplier B=b3 b2 b1 b0 and A= a2 a1 a0.**

For **j multipliers** &
k multiplicands bits ,
we needs

- **j x k AND gates**
- **(j-1)x k bit adders** produce
- A product of **j + k bits**



4-bit by 3-bit array multiplier.

➢Fixed Point Division Operations

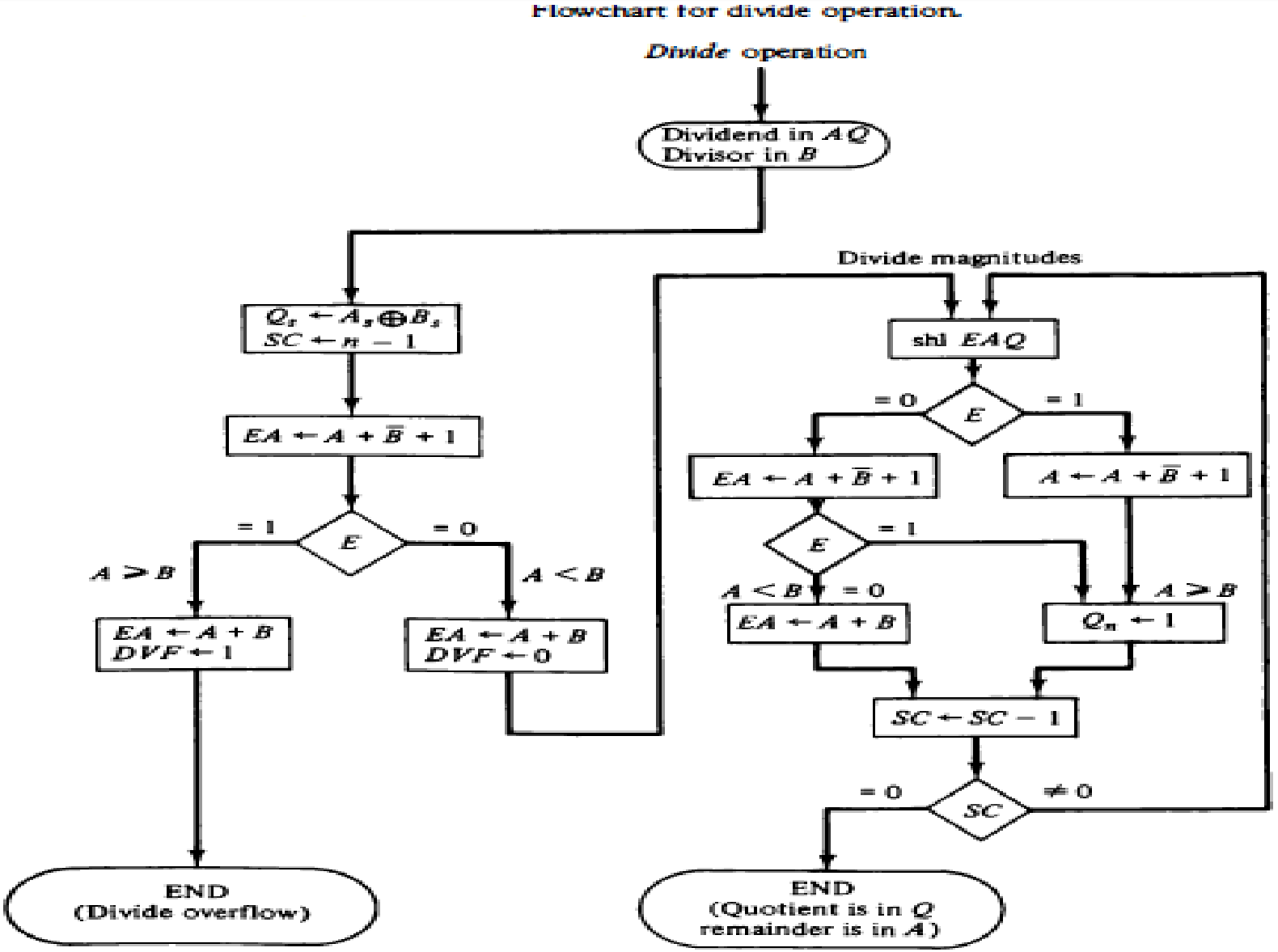
Division Algorithm

•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. .

```
                    00001010 1   Quotient
Divisor  1101 ) 1000 1001 0      Dividend
               -1101
                10000
                -1101
                  1110
                 -1101
                     1           Remainder
```

➢Division Algorithm



➢Division Algorithm

Example

Divisor $B = 10001$, $\overline{B} + 1 = 01111$

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

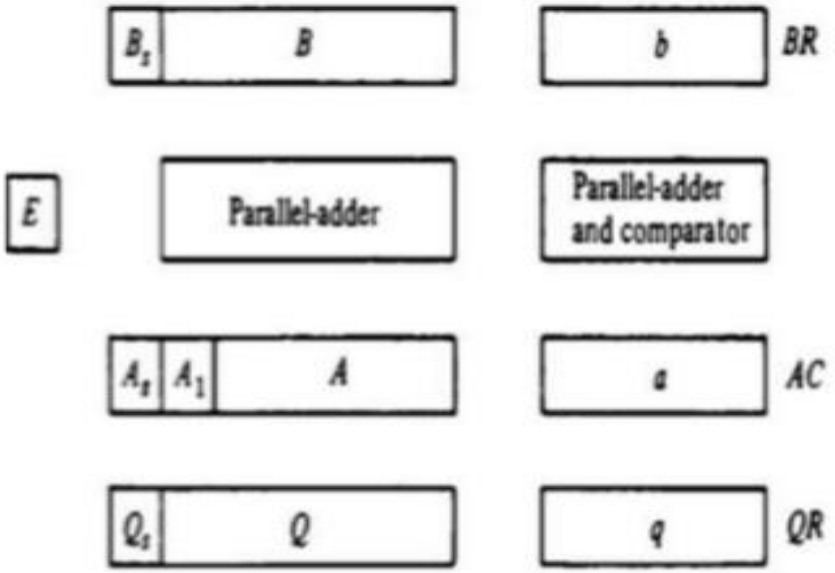
- Check for zeros.
- Align the mantissas.
- Add or subtract the mantissas.
- Normalize the result.

➢Floating point arithmetic operation

Registers for floating point arithmetic operations

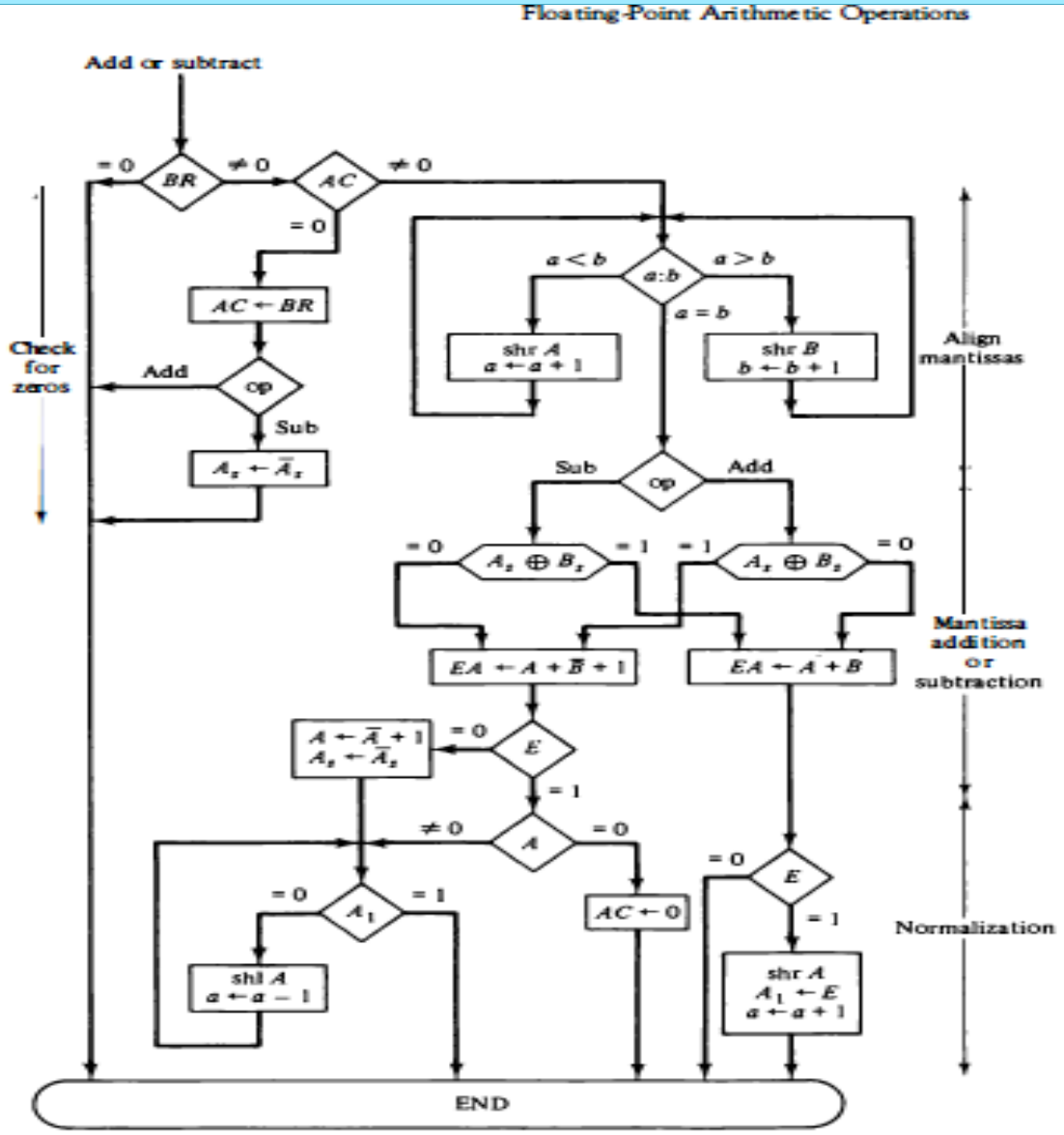
- 3 registers :BR,AC,QR
- All have 2 parts: mantissa(upper case letter), exponent(lower case letter)
- Assumed that each floating point number has a mantissa in signed magnitude and a biased exponent(bias is no. that is added to exponent. Ex exp that ranges from -50 to 49, we consider 00 to 99 as +ve number. Positive exponents range from 99 to 50 and 49 to 00 as negative exponent)
- Comparator: used to compare the +ve exponents to adjust mantissa part for normalization.
- A_s:sign bit of A, Qs: sign bit of Q, Bs: sign bit of B., A1: 1 if the number is to normalized.
- 2 parallel adder, A: mantissa Q+B, E:carry bit

  a: exponent b+q



Isha Padhy,CSE Dept, CBIT

Floating Point Addition & Subtraction



➢Floating Point Multiplication Operation

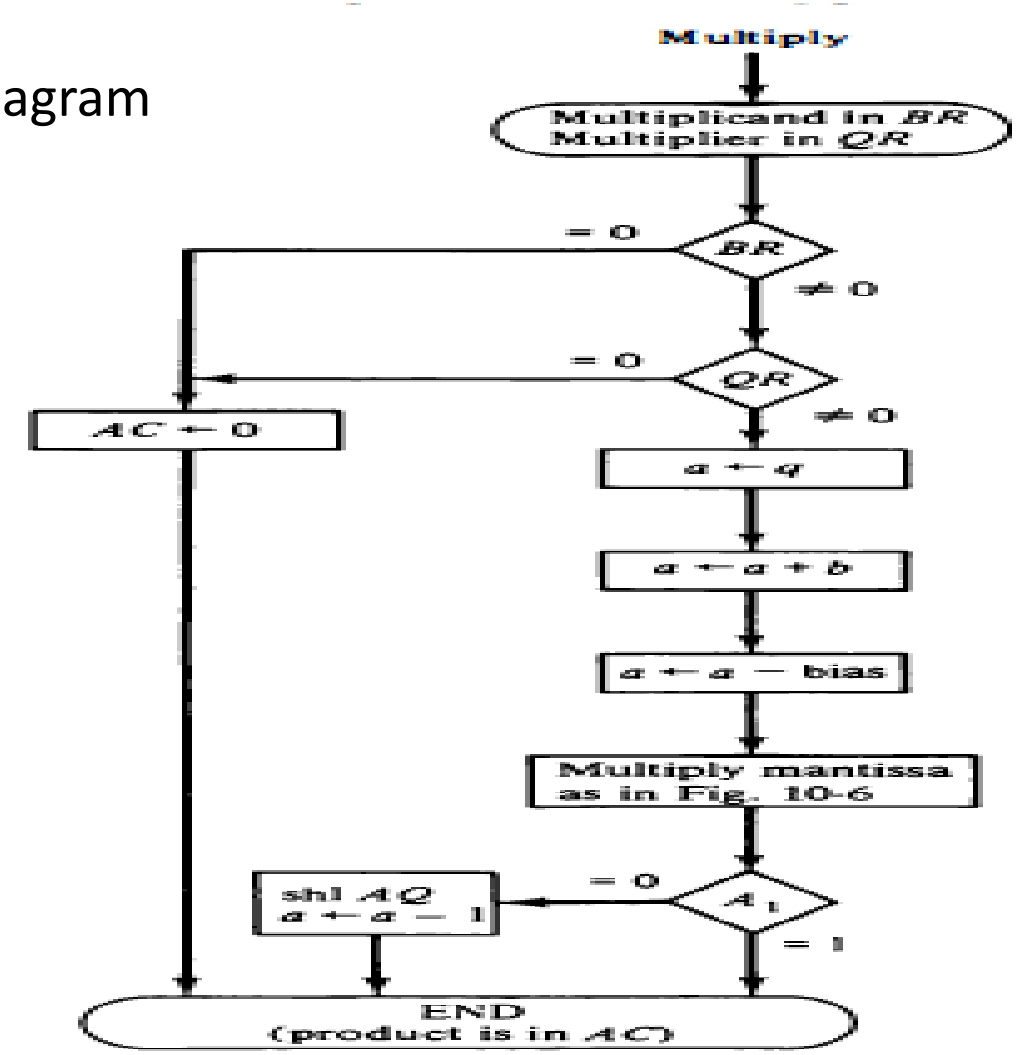
The multiplication of two floating-point numbers requires that we multiply the mantissas and add the exponents. The multiplication of the mantissas is performed in the same way as in fixed-point to provide a double-precision.

The multiplication algorithm can be subdivided into four parts:

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

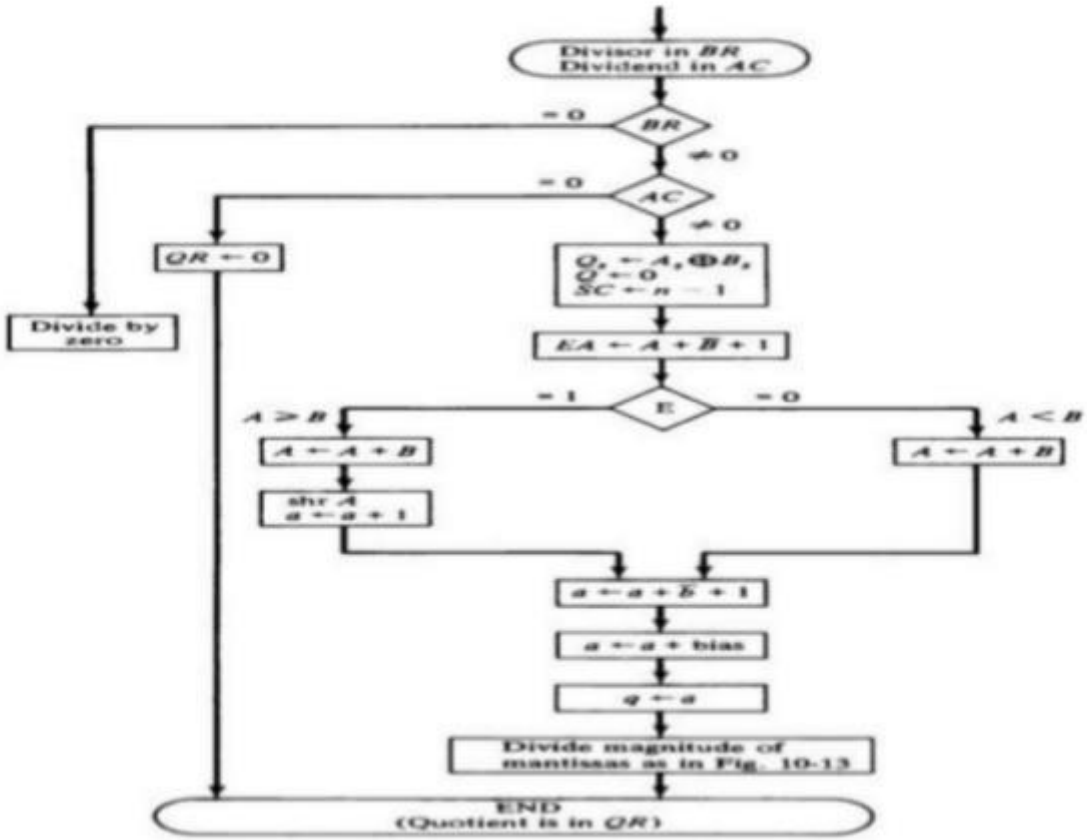
Floating Point Multiplication Operation

Flow diagram



Floating Point Division Operation

Flow diagram



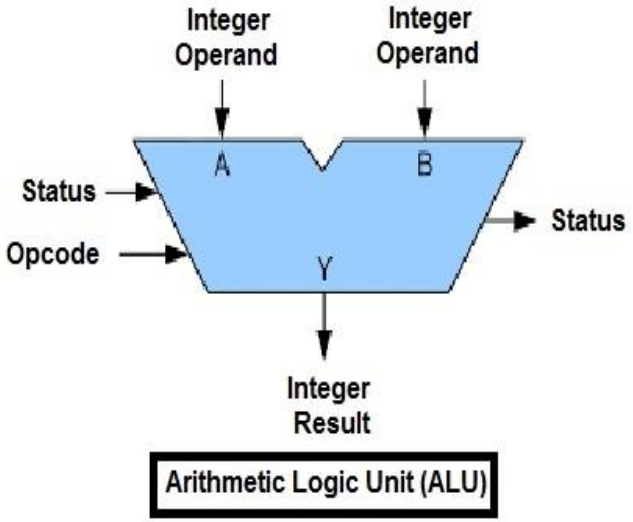
➢Arithmetic & Logic Unit Design

•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 memory. ALU is 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
M1	*M0*	*Operation*
0	0	SUM
1	0	AND
0	1	OR
1	1	XOR

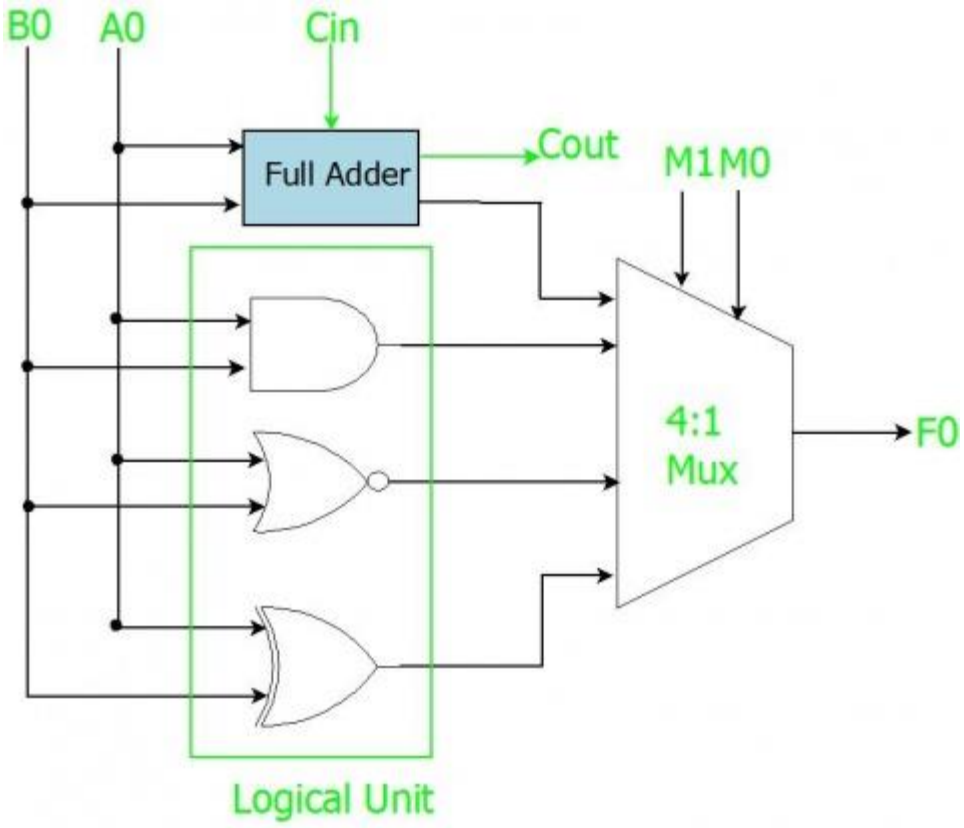


Figure: 1 bit ALU

➢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 –

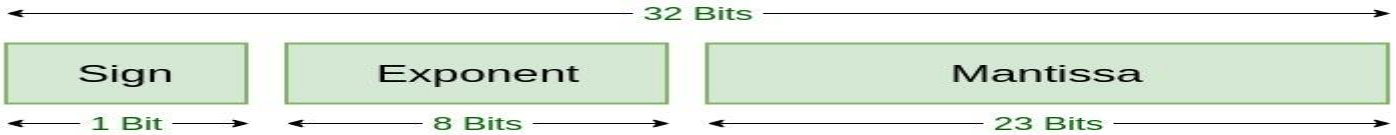
A bias is added to the actual exponent in order to get the stored exponent.

The Normalized Mantissa –

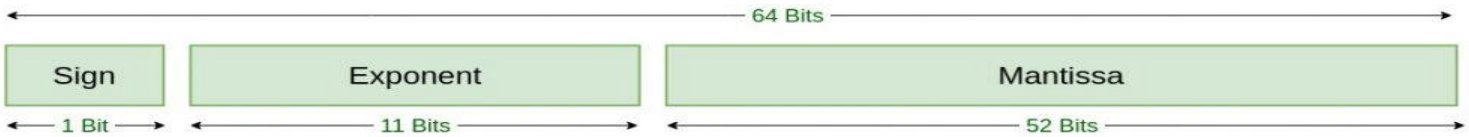
A normalized mantissa is one with only 1 to the left of the binary point like 1.M

➢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

➢IEEE Standard for Floating Point Numbers

TYPES	SIGN	BIASED EXPONENT	NORMALISED MANTISA	BIAS
Single precision	1(31st bit)	8(30-23)	23(22-0)	127
Double precision	1(63rd bit)	11(62-52)	52(51-0)	1023

```
85.125
85 = 1010101
0.125 = 001
85.125 = 1010101.001
       =1.010101001 x 2^6
sign = 0

1. Single precision:
biased exponent 127+6=133
133 = 10000101
Normalised mantisa = 010101001
we will add 0's to complete the 23 bits

The IEEE 754 Single precision is:
= 0 10000101 01010100100000000000000
This can be written in hexadecimal form 42AA4000
```

```
2. Double precision:
biased exponent 1023+6=1029
1029 = 10000000101
Normalised mantisa = 010101001
we will add 0's to complete the 52 bits


The IEEE 754 Double precision is:
= 0 10000000101 0101010010000000000000000000000000000000000000000000
This can be written in hexadecimal form 4055480000000000
```

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=a1a0, B=b1 b0 .

Weekly Assignment

1. Design array multiplier for b1 b0 X a1 a0 and b3 b2 b1 b0 X a2 a1 a0 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 Cary Look Ahead adder with suitable diagram and advantages .
5. Define IEEE 754 standard for floating point representation with suitable example.

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

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

MCQ

- 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

(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.

1. 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 ← R2) else if (Q=1) then (R1 ← 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) X (+3)

Summary

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