



# SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai

Accredited by NAAC-UGC with 'A++' Grade (Cycle III),

Accredited by NBA (B.E - CSE, EEE, ECE, Mech, B.Tech.IT)

COIMBATORE-641 035, TAMIL NADU



## COURSE NAME : 23ITT202 – COMPUTER ARCHITECTURE

### II YEAR/ III SEMESTER

### UNIT – II ARITHMETIC OPERATIONS

#### Topic: Question Bank

#### PART A

1. State the principle of operation of a carry look-ahead adder.  
The input carry needed by a stage is directly computed from carry signals obtained from all the preceding stages  $i-1, i-2, \dots, 0$ , rather than waiting for normal carries to supply slowly from stage to stage. An adder that uses this principle is called carry look-ahead adder.
2. What are the main features of Booth's algorithm?
  - 1) It handles both positive and negative multipliers uniformly.
  - 2) It achieves some efficiency in the number of addition required when the multiplier has a few large blocks of 1s.
3. How can we speed up the multiplication process?(CSE Nov/Dec 2003)  
There are two techniques to speed up the multiplication process:
  - 1) The first technique guarantees that the maximum number of summands that must be added is  $n/2$  for  $n$ -bit operands.
  - 2) The second technique reduces the time needed to add the summands.
4. What is bit pair recoding? Give an example.  
Bit pair recoding halves the maximum number of summands. Group the Booth-recoded multiplier bits in pairs and observe the following: The pair  $(+1 -1)$  is equivalent to the pair  $(0 +1)$ . That is instead of adding  $-1$  times the multiplicand  $m$  at shift position  $i$  to  $+1$  ( $M$  at position  $i+1$ ), the same result is obtained by adding  $+1$  ( $M$  at position  $i$ ).  
Eg: 11010 – Bit Pair recoding value is  $0 -1 -2$
5. What is the advantage of using Booth algorithm?
  - 1) It handles both positive and negative multiplier uniformly.
  - 2) It achieves efficiency in the number of additions required when the multiplier has a few large blocks of 1's.
  - 3) The speed gained by skipping 1's depends on the data.
6. Write the algorithm for restoring division.  
Do the following for  $n$  times:

- 1) Shift A and Q left one binary position.
- 2) Subtract M and A and place the answer back in A.
- 3) If the sign of A is 1, set  $q_0$  to 0 and add M back to A. Where A- Accumulator, M- Divisor, Q- Dividend.

7. Write the algorithm for non restoring division.

Do the following for n times:

Step 1: Do the following for n times:

- 1) If the sign of A is 0, shift A and Q left one bit position and subtract M from A; otherwise, shift A and Q left and add M to A.
- 2) Now, if the sign of A is 0, set  $q_0$  to 1; otherwise, set  $q_0$  to 0. Step 2: if the sign of A is 1, add M to A.

8. When can you say that a number is normalized?

When the decimal point is placed to the right of the first (nonzero) significant digit, the number is said to be normalized.

9. Explain about the special values in floating point numbers.

The end values 0 to 255 of the excess-127 exponent E( are used to represent special values such

as:

When  $E=0$  and the mantissa fraction M is zero the value exact 0 is represented. When  $E=255$  and  $M=0$ , the value ( is represented.

When  $E=0$  and  $M \neq 0$ , denormal values are represented.

When  $E=255$  and  $M \neq 0$ , the value represented is called Not a number.

10. Write the Add/subtract rule for floating point numbers.

- 1) Choose the number with the smaller exponent and shift its mantissa right a number of steps equal to the difference in exponents.
- 2) Set the exponent of the result equal to the larger exponent.
- 3) Perform addition/subtraction on the mantissa and determine the sign of the result
- 4) Normalize the resulting value, if necessary.

11. Write the multiply rule for floating point numbers.

- 1) Add the exponent and subtract 127.
- 2) Multiply the mantissa and determine the sign of the result .
- 3) Normalize the resulting value, if necessary.

12. What is the purpose of guard bits used in floating point arithmetic

Although the mantissa of initial operands are limited to 24 bits, it is important to retain extra bits, called as guard bits.

13. What are the ways to truncate the guard bits?

There are several ways to truncate the guard bits: 1) Chopping 2) Von Neumann rounding 3) Rounding

14. Define carry save addition(CSA) process.

Instead of letting the carries ripple along the rows, they can be saved and introduced into the next row at the correct weighted position. Delay in CSA is less than delay through the ripple carry adder.

15. What are generate and propagate function?

The generate function is given by  $G_i = x_i y_i$  and

The propagate function is given as  $P_i = x_i + y_i$ .

16. What is floating point numbers?

In some cases, the binary point is variable and is automatically adjusted as computation proceeds.

In such case, the binary point is said to float and the numbers are called floating point numbers.

17. In floating point numbers when so you say that an underflow or overflow has occurred?

In single precision numbers when an exponent is less than -126 then we say that an underflow has occurred. In single precision numbers when an exponent is less than +127 then we say that an overflow has occurred.

18. What are the difficulties faced when we use floating point arithmetic?

Mantissa overflow: The addition of two mantissas of the same sign may result in a carryout of the most significant bit

Mantissa underflow: In the process of aligning mantissas, digits may flow off the right end of the mantissa.

Exponent overflow: Exponent overflow occurs when a positive exponent exceeds the maximum possible value.

Exponent underflow: It occurs when a negative exponent exceeds the maximum possible exponent value.

19. In conforming to the IEEE standard mention any four situations under which a processor sets exception flag.

Underflow: If the number requires an exponent less than -126 or in a double precision, if the number requires an exponent less than -1022 to represent its normalized form the underflow occurs.

Overflow: In a single precision, if the number requires an exponent greater than +127 or in a double precision, if the number requires an exponent greater than +1023 to represent its normalized form the underflow occurs.

Divide by zero: It occurs when any number is divided by zero.

Invalid: It occurs if operations such as 0/0 are attempted.

20. Why floating point number is more difficult to represent and process than integer?(CSE May/June 2007)

An integer value requires only half the memory space as an equivalent IEEE double-precision floating point value. Applications that use only integer based arithmetic will therefore also have significantly smaller memory requirement

A floating-point operation usually runs hundreds of times slower than an equivalent integer based arithmetic operation.

21. Give the booth's recoding and bit-pair recoding of the computer.

1000111101000101(CSE May/June 2006)

Booth's recoding

1 0 0 0 1 1 1 1 0 1 0 0 0 1 0 1 0

-1 0 0 +1 0 0 0 -1 +1 -1 0 0 +1 -1 +1 -1

Bit-Pair recoding:

1 0 0 0 1 1 1 1 0 1 0 0 0 1 0 1 0

-2 +1 0 -1 +1 0 +1 1

22. Draw the full adder circuit and give the truth table (CSE May/June 2007)

Inputs			Outputs	
A	B	C	Carry	Sum
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

#### PART B

1. Explain in detail about addition and subtraction of signed number with diagram and example.
2. Illustrate the concept of Carry Look ahead Adder with diagram.
3. Show how to implement full adder by using two half adders and external logic gates.
4. Design and inspect the operation of Full Adder by constructing full adder using 2 half adders.
5. Multiply the following pair of signed 2's complement numbers using the Booth algorithm. Assume that A is the multiplicand & B is the multiplier. A=01101 & B=11010. Compare it with normal multiplication and explain how the concept of Booth recoding of multiplier is effective.

6. Elaborate the concept of Integer division. Describe and analyze the difference between restoring and non-restoring division algorithm & apply any one algorithm execution for Dividend value 1000 and Divisor value 11.
7. Illustrate the advantages of fast multiplication over other multiplication algorithms and apply the two types of fast multiplication concept for the following values. Assume that A is the multiplicand & B is the multiplier. A=01101 & B=11010.