MGM's College of Engineering and Technology, Kamothe, Navi Mumbai

Department of Computer Engineering

Academic Year -2021-2022

Assignment-2

Course Code: CSC 304
Course Name: Digital Logic & Computer Organization and Architecture
Class: SE-A

1. What is the addition of the binary numbers 11011011010 and 010100101?
a) 0111001000
b) 1100110110
c) 11101111111
d) 10011010011

2. Perform binary subtraction: 101111 – 010101 = ?
a) 100100
b) 010101
c) 011010
d) 011001

3. 100101 × 0110 = ?
a) 1011001111
b) 0100110011
c) 101111110
d) 0110100101

4.Booths algorithm is applied on

a. binary number b. decimal number c. octal number d. hexadecimal number

5.In Booth's non-restoring division algorithm, after performing left shift operation on A,Q registers, if magnitude of A > 0 then ?
a.Q0=0, A=A+M b.A=A+M c.Q0=1 d.A=A-M

6.In Booth's non-restoring division algorithm after performing left shift operation on A,Q register, if sign of A is positive?
a.Q0=0, A=A+M b.A=A+M c.Q0=1 d.A=A-M

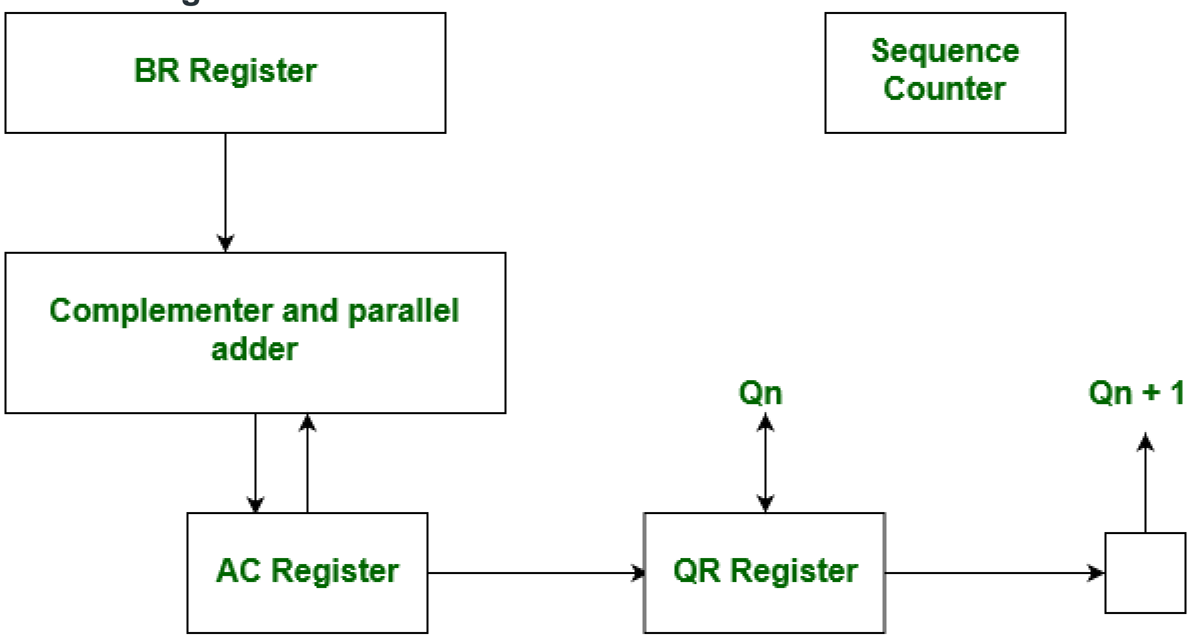
7.In Booth's restoring division algorithm, after performing operations (1) left shift operation on A,Q and (2) A=A -M, if sign of A is positive?
a.Q0=0, A=A+M b. A=A+M c.Q0=1 d.A=A-M

Q2.

1. Explain Booth's Algorithm with flowchart and example

   Ans: Booth algorithm gives a procedure for **multiplying binary integers** in signed 2's complement representation **in efficient way**, i.e., less number of additions/subtractions required. It operates on the fact that strings of 0's in the multiplier require no addition but just shifting and a string of 1's in the multiplier from bit weight 2^k to weight 2^m can be treated as 2^(k+1) to 2^m.

   Booth's Algorithm Flowchart –



Example – A numerical example of booth's algorithm is shown below for n = 4. It shows the step by step multiplication of -5 and -7.

```
MD = -5 = 1011, MD = 1011, MD'+1 = 0101

MR = -7 = 1001

The explanation of first step is as follows: Qn+1

AC = 0000, MR = 1001, Qn+1 = 0,  SC = 4

Qn Qn+1 = 10

So, we do AC + (MD)'+1, which gives AC = 0101

On right shifting AC and MR, we get

AC = 0010, MR = 1100 and Qn+1 = 1
```

2. Multiply (-7) and (+3) using Booth's Algorithm.

Ans:

3. Multiply (17) and (5) using Booth's algorithm.

Ans:

4. Explain Restoring Division algorithm with flowchart.

Ans: Figure below shows the hardware implementation of Restoring Binary Division.

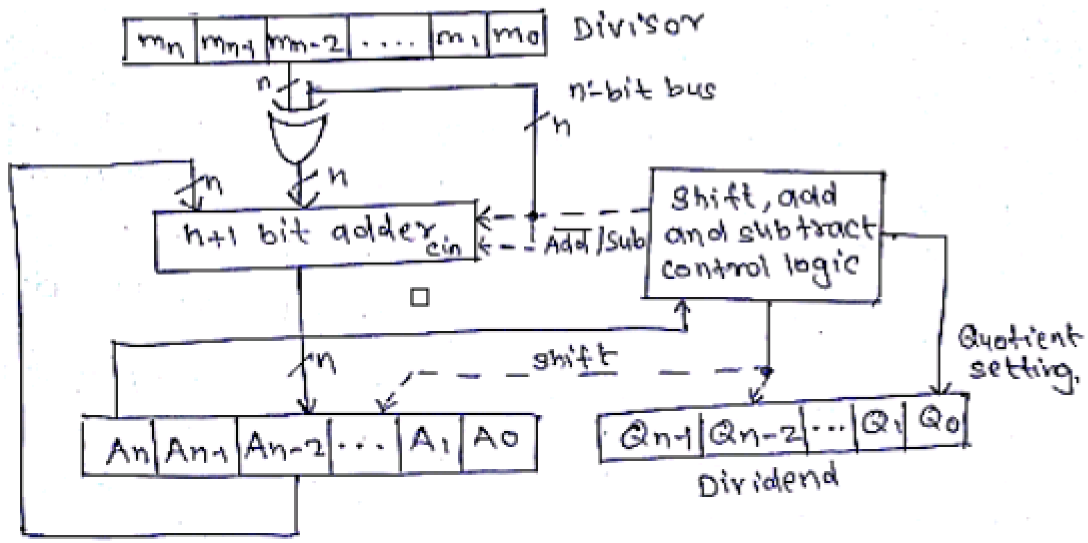


Figure 1: Hardware Implementation for Restoring Binary Division

1) The divisor is placed in M register, the dividend placed in Q register.

2) At every step, the A and Q registers together are shifted to the left by 1-bit

3) M is subtracted from A to determine whether A divides the partial remainder. If it does, then Q_0Q0 set to 1-bit. Otherwise, Q_0Q0 gets a 0 bit and M must be added back to A to restore the previous value.
4) The count is then decremented and the process continues for n steps. At the end, the quotient is in the Q register and the remainder is in the A register.

The steps are as follows:-

Step 1: Shift A and Q left by one binary position

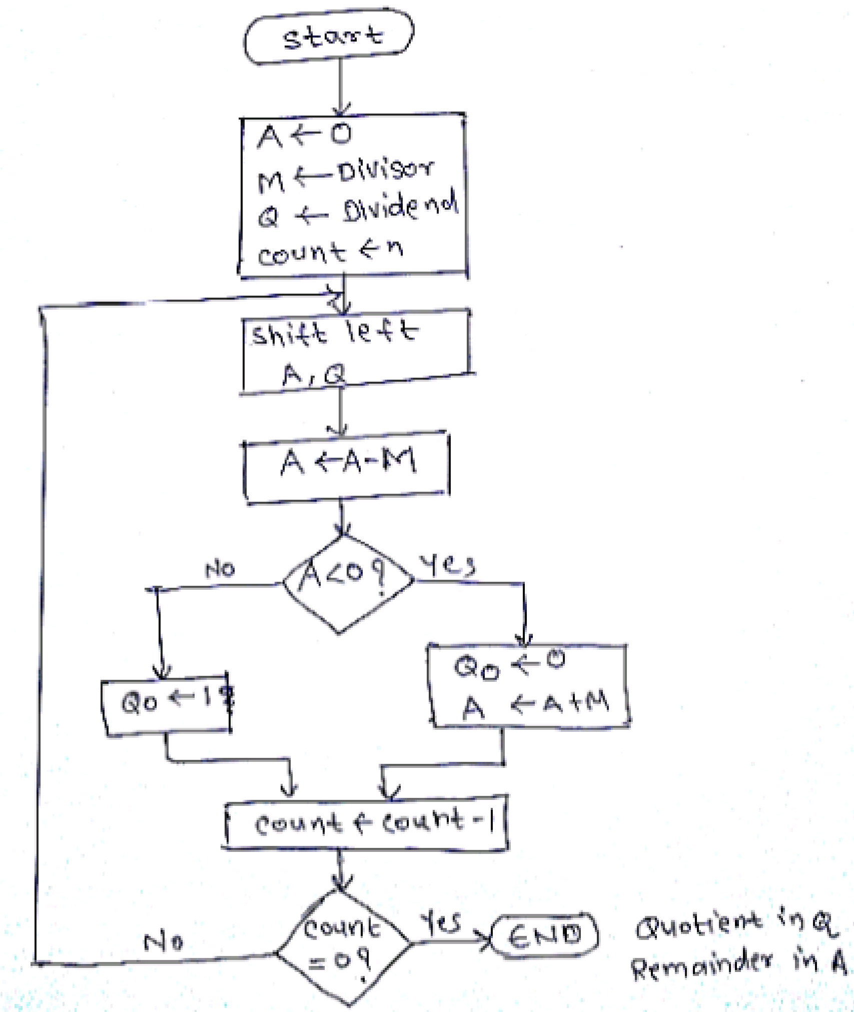
Step 2: Subtract divisor M- from A and place the answer in A (A←A-M)

Step 3: If the sing bit of A is 1, set Q0 to 0 and add divisor back to A, otherwise set Q0 to 2

Step 4: Repeat steps 1,2,3…..n times.

Flowchart:

Figure below shows the flowchart for restoring algorithm



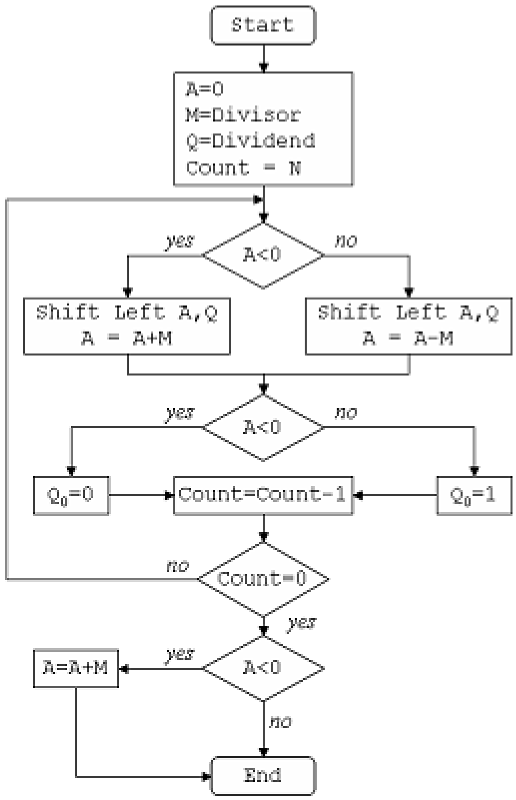
5. Divide 13 by 5 through restoring division method.

Ans:

Step	A	Q	B	Operation
0	0 0000	1101	0101	Initialization
1	0 0001 1 1011 1 1100 0 0101 0 0001	1010 1010	 0101	Shift left A_Q A = A - B A = A + B, Q_0 = 0
2	0 0011 1 1011 1 1110 0 0101 0 0011	0100 0100	 0101	Shift left A_Q A = A - B A = A + B, Q_0 = 0
3	0 0110 1 1011 0 0001	1000 1001	 0101	Shift left A_Q A = A - B Q_0 = 1
4	0 0011 1 1011 1 1110 0 0101 0 0011	0010 0010	 0101	Shift left A_Q A = A - B A = A + B, Q_0 = 0

6. Explain Non Restoring Division algorithm with flowchart.

Ans: Algorithm for Non-restoring division is given in below image :



- A variant that skips the restoring step and instead works with negative residuals
- If P is negative
  1. (i-a) Shift the register pair (P,A) one bit left
  2. (ii-a) Add the contents of register B to P
- If P is positive
  1. (i-b) Shift the register pair (P,A) one bit left
  2. (ii-b) Subtract the contents of register B from P
- (iii) If P is negative, set the low-order bit of A to 0, otherwise set it to 1
- After n cycles
  1. The quotient is in A
  2. If P is positive, it is the remainder, otherwise it has to be restored (add B to it) to get the remainder

7. Divide 11 by 3 through non restoring division method.

Ans:

N	M	A	Q	Action
4	00011	00000	1011	Start
		00001	011_	Left shift AQ
		11110	011_	A=A-M
3		11110	0110	Q[0]=0
		11100	110_	Left shift AQ
		11111	110_	A=A+M
2		11111	1100	Q[0]=0
		11111	100_	Left Shift AQ
		00010	100_	A=A+M
1		00010	1001	Q[0]=1
		00101	001_	Left Shift AQ
		00010	001_	A=A-M
0		00010	0011	Q[0]=1

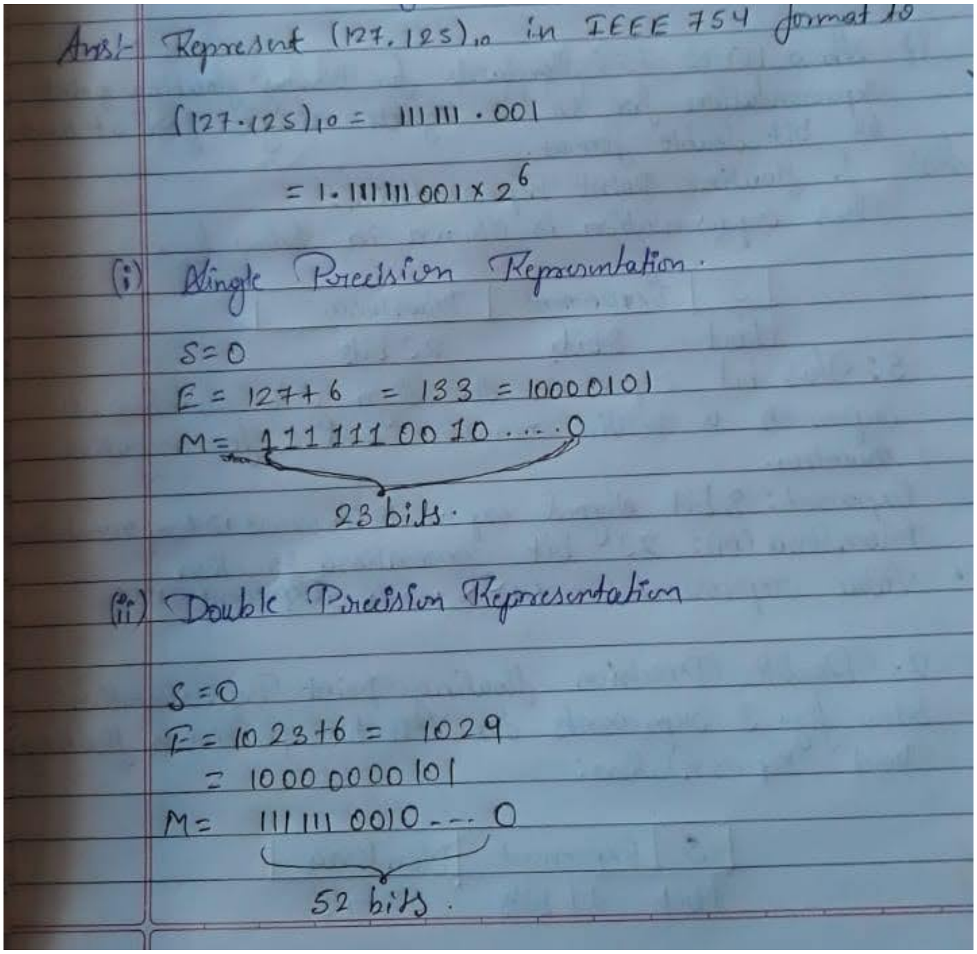
```
Quotient  = 3 (Q)
Remainder = 2 (A)
```

8. Show IEEE 754 standards for binary floating point representation for 32 bit single format and 64 bit double format.
   Ans: There are 4294967296 patterns for any 32-bit format and 18446744073709551616 patterns for the 64-bit format. • The number of representable float data is same as int data. But a wider

range can be covered by a floating-point format due to non-uniform distribution of values over the range

9. Express (127.125) in IEEE-754 single and double precision floating point representation.

Ans:



Ans:- Represent $(127.125)_{10}$ in IEEE 754 format to

$(127.125)_{10} = 111111.001$

$= 1.111111001 \times 2^6$

(i) Single Precision Representation.

$S = 0$

$E = 127 + 6 = 133 = 10000101$

$M = \underbrace{1111110010\ldots0}_{23 \text{ bits}}$

(ii) Double Precision Representation

$S = 0$

$E = 1023 + 6 = 1029$

$= 10000000101$

$M = \underbrace{1111110010\ldots0}_{52 \text{ bits}}$