Jawahar Education Societys Annasaheb Chudaman Patil College of Engineering, Kharghar, Navi Mumbai

NAME: PRIYUSH BHIMRAO KHOBRAGADE

PRN NO: 211112018

SUBJECT: DIGITAL LOGIC & COMPUTER ORGANIZATION AND ARCHITECTURE LAB

02

AIM: To implement Booth's algorithm

Practical No. 2

● **Aim**: To implement Booth's algorithm.

● **Objectives**: To implement the operation of the arithmetic unit using the Booths algorithms.

● **Outcomes**: Learner will able to understand the implementation and working of Booth algorithm.

● **Hardware / Software Required:** Any programming language C, Java etc.

● **Theory**:

Booth's Multiplier:

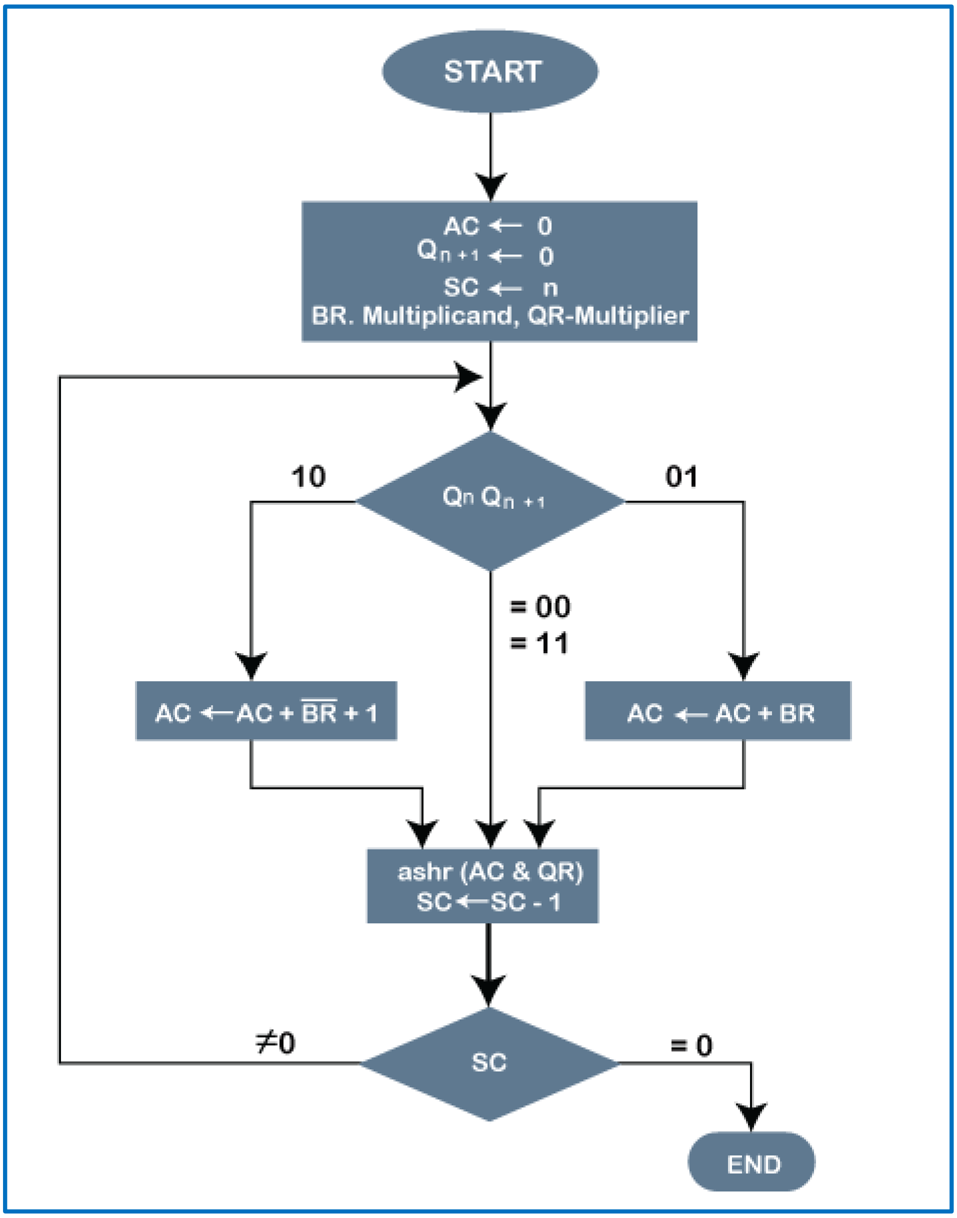
Booth's multiplication algorithm is an algorithm which multiples 2 signed integers in 2's complement. The algorithm is depicted in the following figure with a brief description.

This approach uses fewer additions and subtractions than more straightforward algorithms.

- The multiplicand and multiplier are placed in the m and Q registers respectively.
- A 1 bit register is placed logically to the right of the LSB (least significant bit) Q0 of
- Qregister. This is denoted by Q-1. A and Q-1 are initially set to 0.
- Control logic checks the two bits Q0 and Q-1.
- If the two bits are same (00 or 11) then all of the bits of A, Q, Q are shafted 1 bit to the right.
- If they are not the same and if the combination is 10 then the multiplicand is
- Subtracted from A and if the combination is 01 then the multiplicand is added with A.
- In both the cases results are stored in A, and after the addition or subtraction operation, A, Q, Q-1 are right shifted.
- The shifting is the arithmetic right shift operation where the left most bit namely, An-1
- Is not only shifted An-2 but also remains in An-1. This is to preserve the sign of the number in A and Q.
- The result of the multiplication will appear in the A and Q.

AIM: To implement Booth's algorithm

Algorithm :



AIM: To implement Booth's algorithm

Program Input:

```
import java.util.*;

class BoothsMultiplication
{
static void addition(int A[ ],int B[],int n)
{
	int carry=0,i,j;
	for(i=0;i<n;i++)
	{
		int temp=A[i]+B[i]+carry;
		A[i]=temp%2;
		carry=temp/2;
	}
}

static void subtraction(int A[ ],int B[],int n)
{
	int i,j;
	int M[]=new int[n];
	for(i=0;i<n;i++)
	{
		M[i]=B[i];
	}
	int sum[ ]=new int[n];
	for(i=0;i<n;i++)
	{
		if(M[i]==0)
		M[i]=1;
		else
		M[i]=0;
```

```
        }
        int carry=0;
        int temp=1+M[0];
        sum[0]=temp%2;
        carry=temp/2;
        for(i=1;i<n;i++)
        {
                temp=M[i]+carry;
                sum[i]=temp%2;
                carry=temp/2;
        }
        carry=0;
        for(i=0;i<n;i++)
        {
                temp=A[i]+sum[i]+carry;
                A[i]=temp%2;
                carry=temp/2;
        }
}

static void display(int A[],int B[],int Q[],int Q1,int n)
{
        for(int i=n-1;i>=0;i--)
{
System.out.print(A[i]);
}
System.out.print("\t");
for(int i=n-1;i>=0;i--)
{
System.out.print(Q[i]);
}
```

```
System.out.print("\t");
System.out.print(Q1);
System.out.print("\t");
for(int i=n-1;i>=0;i--)
{
System.out.print(B[i]);
}
System.out.println();
}
public static void main(String args[ ])
{
int b,n,i,j,q,mod;
System.out.println("Enter how many bits you required");
Scanner kbd=new Scanner(System.in);
n=kbd.nextInt();
int Q1=0;
int count =n;
int A[ ]=new int[n];
int B[ ]=new int[n];
int Q[ ]=new int[n];
System.out.println("Enter the multiplicand");
b=kbd.nextInt();
System.out.println("Enter the multiplier");
q=kbd.nextInt();
System.out.println("A"+"\t"+"Q"+"\t"+"Q1"+"\t"+"B");
for(i=0;i<n;i++)
{
mod=b%2;
B[i]=mod;
b=b/2;
}
```

```
for(i=0;i<n;i++)
{
mod=q%2;
Q[i]=mod;
q=q/2;
}
display(A,B,Q,Q1,n);
while(count!=0)
{
if(Q[0]==0&&Q1==1)
{ addition(A,B,n);
display(A,B,Q,Q1,n);
}
else
if(Q[0]==1&&Q1==0)
{
subtraction(A,B,n);
display(A,B,Q,Q1,n);
}
Q1=Q[0];
for(i=1;i<n;i++)
Q[i-1]=Q[i];
Q[n-1]=A[0];
for(i=1;i<n;i++)
A[i-1]=A[i];
count=count-1;
display(A,B,Q,Q1,n);
System.out.println();
}
System.out.println("Multiplication is=");
for(i=n-1;i>=0;i--)
```

```
{
System.out.print(A[i]);
}
for(i=n-1;i>=0;i--)
{
System.out.print(Q[i]);
}
}
}
```

Output

```
C:\WINDOWS\system32\cmd.exe

C:\Users>cd..

C:\>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\>cd C:\Program Files\Java\jdk1.8.0_111\bin\demo

C:\Program Files\Java\jdk1.8.0_111\bin\demo>javac BoothsMultiplication.java

C:\Program Files\Java\jdk1.8.0_111\bin\demo>java BoothsMultiplication
Enter how many bits you required
5
Enter the multiplicand
14
Enter the multiplier
10
A       Q       Q1      B
00000   01010   0       01110
00000   00101   0       01110

10010   00101   0       01110
11001   00010   1       01110

00111   00010   1       01110
00011   10001   0       01110

10101   10001   0       01110
11010   11000   1       01110

01000   11000   1       01110
00100   01100   0       01110

Multiplication is=
0010001100
```

● **Conclusion:** Hence, It is to be concluded that this presentation deals with the design **approach of Booth's algorithm**. Further, we have observed the simulation results of the booth multiplier.