Documentation

In this project (Booth's Algorithm) I have calculated multiplication of two binary numbers using Booth's algorithm. There are three cases for multiplication:- both numbers are positive, one is positive and one is negative , and last both numbers are negative.

Code Explanation:-
Firstly I have taken two integer inputs and then check their sign , if both are positive then func_both_positive is called , if one positive and another one is negative then func_one_negative is called and if both are negative then func_both_negative is called.

Working of func_Both_positve :-
In this function I have converted input integers into binary numbers by using decToBinary function and stored the converted input in arraylists named as M and Q. Then, convert M and Q into String and stored in M2 and Q2 respectively.Now if length of M2 is more than Q2 then add 0's in the end of Q2 to make their length equal and if Q2 is more than M2 then add 0's in the end of M2 to make their length equal. Declaring AC String and assigning AC =0 , and making it's equal length as M2 and Q2 by adding 0's in the end and then declaring a count whose value is equal to length of M2 and declaring q1 string equal to "0" . Declaring int registersize which is equal to count in value. Now running a while loop until count becomes 0. And now inside while loop adding first character of Q2 and q1 and then comparing with three condition :-

1st condition if the added value is equal to “10” then i am finding 2’s compliment of M2 by using convert_2scom and storing it into another String named nM and adding AC and nM by using addBinary and then replacing ACwith reversed added value of AC and nM. In some cases length of AC will be more than registersize and then we will remove extra value from AC and make it ‘s length equal to registersize and now performing ASR func by giving arguments AC ,Q2 ,q1 and storing return value of ASR into String array aa and then replacing AC with first value aa[0] and Q2 with aa[1] and q1 with aa[2] and now printing value of reversed values of AC ,Q2 and q1 and making count--.

2nd Condition if the added value is equal to “01” then i am adding AC and M2 by using addBinary and then replacing AC with reversed added value of AC and M2. In some cases length of AC will be more than registersize and then we will remove extra value from AC and make it ‘s length equal to registersize and now performing ASR func by giving arguments AC ,Q2 ,q1 and storing return value of ASR into String array aa and then replacing AC with first value aa[0] and Q2 with aa[1] and q1 with aa[2] and now printing value of reversed values of AC ,Q2 and q1 and making count--.

3rd condition is if the added value is equal to “00” or “11” and then performing ASR func by giving arguments AC ,Q2 ,q1 and storing return value of ASR into String array aa and then replacing AC with first value aa[0] and Q2 with aa[1] and q1 with aa[2] and now printing value of reversed values of AC ,Q2 and q1 and making count--.

Repeating the above process until count becomes zero and then I will be able to print the reversed value of AC + reversed value of Q2 which is result of multiplication of two binary numbers and then atlast I am printing decimal value of binary number.

Working of func_one_negative:-
In this difference is just inside while loop when adding first character of Q2 and q1 and then comparing with three condition and these are
1)if the added value is equal to "10" than replace AC with reversed added value of AC and M2
2)if the added value is equal to "01" then replace AC with reversed added value of AC and nM2 (2's compliment of M2) .
3)this is same
In this for output I am adding 1 to front the output String ,this will be give us negative output.
Other than these, the whole process is same.

Working of func_both_negative:-
In there are two difference first is we are using nQ2 in place of Q2 which is 2's complement of Q2 and other is inside while loop when adding first character of Q2 and q1 and then comparing with three condition and these are :-
1)if the added value is equal to "10" than replace AC with reversed added value of AC and M2
2)if the added value is equal to "01" then replace AC with reversed added value of AC and nM2 (2's compliment of M2) .
3)this is same
Other than these, the whole process is same.

Explanation of ASR function(return type String array):-
I am taking three parameters String AC , String q and String q1 , then declaring a String q11 in it we storing 1st character of q ,declaring an empty string nq and then making for loop (i=1,i<q.length;i++) coping data of q from character at 1 to end of the string and then adding first character of AC at the ending of nq, declaring an empty String nac ,then making for loop (i=1,i<q.length;i++) coping data of AC from character at 1 to end of the string and then adding last character of AC at the ending of the nac.
Making a new string array whose first element is nac ,second is nq and last is q11, and then returns a .

Explanation of decToBinary function(return type arraylist):-
Takes parameter int n, Declaring an arraylist bin(<Integer>) and int i=0 and then making(running) while(n>0) adding n%2 in bin and n=n%2 this while loop runs until n>0, at last returns bin.

Explanation of addBinary function(return type String):-
Takes parameters String a, String b, declaring an empty String empty result , int s=0 , i=a.length()-1 and j=b.length()-1. Now making while(i>=0 || j>=0 || s==1) in it adding a and b , single bit at a time by using s+=((i>=0)? a.charAt(i)-'0' :0),
and then s+=((j>=0)? a.charAt(j)-'0' :0)
and then storing (char)(s%2+ '0') + result in result and making s=s/2 and i-- ,j-- , run loop while(i>=0 || j>=0 || s==1) and at last return result.

Explanation of convert_arraylist_string (return String):-
Takes parameters Arraylist a , declaring an empty string s and then running for loop(i=0;i<a.size();i++) and inside it s+=Integer.toString ((Integer)a.get(i)) , and atlast return a String s.

Explanation of convert_2scom(return type String):-
Takes parameter String s, declaring an empty String ns, making for loop(int i=0;i<s.length();i++) inside it replacing each 0 with 1 and 1 with 0 and storing in ns, and then declaring and empty String one and adding s.length()-1 0's in it , then adding 1 at the end of it and then adding ns and one and then return it .

Explanation of reverse function(return type String):-
Takes argument String ac and then declares an empty string and stores reversed string of ac in s by using for loop (i=ac.length()-1;i>0;i--) and inside it s+=Character.toString(ac.charAt(i)) and then return s.

Explanation of pow function(return type int):-
Take the argument int n and int p, n represents number and p represents power then this function will return n^p.

Explanation of BinaryTodec(return type int):-
Takes the argument String s and then converts the binary number to string using basic concept and then returns the result.

Example:- 1010= (2^0)*1 +(2^1)*0 +(2^2)*1 +(2^3)*0=5

Format of String storing binary number(M2 , Q2):-
In M2 ,Q2 etc ….binary number are stored in the way the I have showed in example :-

1010= (2^0)*1 +(2^1)*0 +(2^2)*1 +(2^3)*0=5

Output format:-
I am printing values like we write on paper. Example :-
00101=(2^4)*0+(2^3)*0 +(2^2)*1 +(2^1)*0 +(2^0)*1=5

Arguments of different function format:-
decToBinary- integer
convert_arraylist_String - Arraylist
reverse -String
addBinary - two String (but here String represents two binary numbers which should be reversed meaning generally in AC , Q2 ,nM we are storing like 1010 but for this function i am giving an argument reversed String).
Convert_2scom- two String (reversed reason same as addBinary)
ASR- three String
Pow-two integer
BinaryTodec- One String

Note :-

1) For finding 2’s complement of binary numbers I am sending a reversed String of the binary number and when convert_2scom returns a value and then also reversing it .
2) Above rules are also applicable on addBinary.
3) If anyone of the number in inputs is negative then i am sending it’s mod in of that number in func_one_negative.
4) If both numbers in input are negative then i am sending mod of those numbers in func_both_negative.
5) Input number should lie in 8bit range.
6) I am printing values of AC , Q2 and q1 after each cycle and lastly I am printing the result of multiplication of two binary numbers.