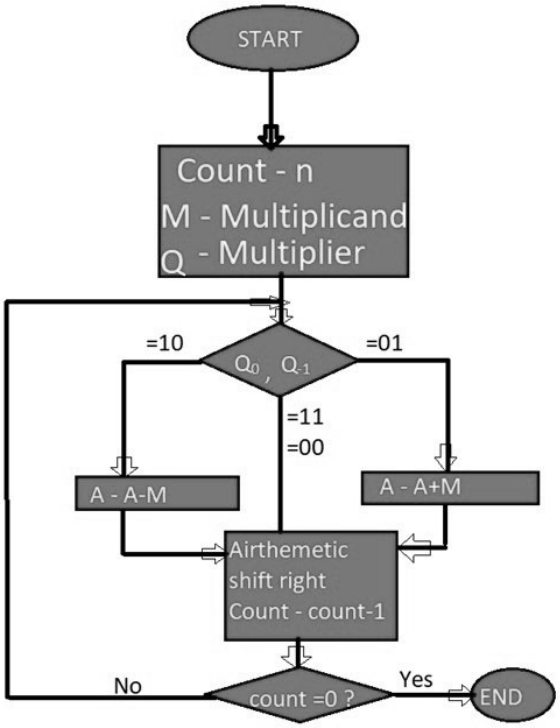
CSE 112
COMPUTER ORGANIZATION

MULTIPLICATION OF TWO INTEGERS USING BOOTH'S ALGORITHM

BOOTH'S ALGORITHM:



THE PROGRAM CODE:

a: ACCUMULATOR initiated as "0000"
n: LENGTH OF BINARY STRING
m: MULTIPLICAND
q: MULTIPLIER
ql: LEAST SIGNIFICANT BIT OF MULTIPLIER
qn: SINGLE BIT INITIALIZED WITH '0'
sb: SIGNED BIT: 1 FOR NEGATIVE: 0 FOR POSITIVE

1) First, the multiplicand and multiplier are taken as input by the user in decimal format. Then, the numbers are converted into binary strings with base 2. The smaller number is taken as the multiplier. A smaller multiplier helps for faster implementation of booths algorithm as the loop is run 'n' times where n is length of multiplier.
2) Both the binary strings are then made to equal lengths by adding extra zeros to the string of lesser length.
3) The length of the strings is taken as 'n'. 'n' is later used as condition for while loop in the algorithm.

4) FUNCTIONS IN THE PROGRAM

- **rightfhift():** This functions right shifts the string passed to it in a circular way.
- **add():** This function is used for addition of two numbers that are passed to it. In this case '**a**' and '**m**'. It is also used for subtraction when 2's complement of a number is passed to it. 2's complement of a number is the negative of a number: a-m=a+(-m)
- **twoscomp():** This function is used to convert a binary number string to its twos complement. It is done by taking 1's compliment by replacing 1 with 0 and 0 with 1 and then adding 1 to it.
- **Subtract():** this function uses the add() and twoscomp() function to subtract two numbers

5) A loop is run 'n' time. Each time 'ql' and 'qn' are checked. If 'ql'='qn', right shift function is called. If 'ql' is 0 and 'qn' is 1, add() function is called and then rightshift() function is called. If 'ql' is 1 and 'qn' is 0. subtract() function is called and then rightshift() function is called. After n steps, the string 'a+q' is the final answer and printed out in binary and decimal formats. Along with answer, signed bit is also printed which is 1 when ans is negative and 0 when answer is positive.

Printing is done at each step.

TEST CASE: MULTIPLYING 3*3

```
enter first number in decimal format:
3
enter second number in decimal format:
3
FIRST BINARY NUMBER IS:  0 0011
SECOND BINARY NUMBER IS:  0 0011
0000     0011    0
0011
Q= 1
Q-1= 0
                                 A=A-M

1101     0011    0
                                 PERFORMING RIGHT SHIFT

1110     1001    1
1001
Q= 1
Q-1= 1
                                 PERFORMING RIGHT SHIFT

1111     0100    1
0100
Q= 0
Q-1= 1
                                 A=A+M

0010     0100    1
                                 PERFORMING RIGHT SHIFT

0001     0010    0
0010
Q= 0
Q-1= 0
                                 PERFORMING RIGHT SHIFT

0000     1001    0

FINAL ANSWER MULTIPLYING:
 0011 and 0011   IS
 0 1001   =  9
MOST SIGNIFICANT BIT IS SIGNED BIT
```

TEST CASE: MULTIPLYING -98*6

```
enter first number in decimal format:
-98
enter second number in decimal format:
6
FIRST BINARY NUMBER IS:  1 1100010
SECOND BINARY NUMBER IS:  0 0000110
0000000          0000110          0
0000110
Q= 0
Q-1= 0
                                  PERFORMING RIGHT SHIFT

0000000          0000011          0
0000011
Q= 1
Q-1= 0
                                  A=A-M

0011110          0000011          0
                                  PERFORMING RIGHT SHIFT

1001111          0000001          1
0000001
Q= 1
Q-1= 1
                                  PERFORMING RIGHT SHIFT

1100111          1000000          1
1000000
Q= 0
Q-1= 1
                                  A=A+M

1001001          1000000          1
                                  PERFORMING RIGHT SHIFT

0100100          1100000          0
1100000
Q= 0
Q-1= 0
                                  PERFORMING RIGHT SHIFT

0010010          0110000          0
0110000
Q= 0
Q-1= 0
                                  PERFORMING RIGHT SHIFT

0001001          0011000          0
0011000
Q= 0
Q-1= 0
                                  PERFORMING RIGHT SHIFT

0000100          1001100          0

FINAL ANSWER MULTIPLYING:
 1100010 and 0000110   IS
 1 1001001100   = - 588
MOST SIGNIFICANT BIT IS SIGNED BIT
```

SUBMITTED BY:
AMAN PRIYADARSHI
2019294