BOOTH'S ALGORITHM

Name: Shubham Lohan

Roll No: 2019275

Computer Organization

Group 01

I am implementing Booth's algorithm for multiplying two No's. I am implementing this algorithm in **python** and register size of this algorithm is always one more than the maximum bits of the no which user entered to multiply. Here are the following list of functions and variables which are used in the Algorithm.

1. **global Variable:**
    - check – Boolean value, True if only one no is negative otherwise False
    - max_no- Stores the maximum no
    - max_bit- Stores the bits of maximum no
2. **Functions:**

- ➢ **def main ()**: Here is the main driver function to drive the algorithm its basically just to entering the no.

Parameter: Nothing

Return: Nothing

- ➢ **def booths_multiplication :** Here is the main Function of this algorithm and all the calculation of the algorithm goes here.

- **max_bit –** maximum no of bit of a binary no
- then it check according algorithm that Q0[last]+Q_=='01' or '10' or '11' or '00'
- then it follows according to the algorithm

Parameter: string

Return: Nothing

- ➢ **def result() :** Its just for calculating the two's complementing the negative no.

Parameter: string

Return: string

- ➢ **def flipped() :** Its just for flipping the bits

Parameter: string

Return: string

- ➢ **def perform_operation():** It perform the operation by comparing Q and Q-1.

Parameter : string

Return : string

- ➢ **def subtraction():** it calculates the subtraction of two binary no

Parameter : string

Return : string

- **def ASR() :** It performs the arithmetic shift toward right

Parameter : string

Return: string

- **def binAdd():** It calculates the addition of two binary no

Parameter : string

Return : string

- **def covert_Dec_to_binary():** It converts decimal no to binary

Parameter: string

Return: string

- **def twos_complement() :** It calculates the two's complement

Parameter: string

Return: string

- **def spacer() :** It provide the appropriate space for printing.

Parameter: string

Return: string

- **def product_value ():** just for printing purposes.

Parameter : string

Return : string

Note :-

To make the algorithm more efficient, if one of the two no's is zero then it just print the zero.