Snehal Keshav Nalawade (202151160)
Section 2

MULTIPLICATION ALGORITHM

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE Engineering

Submitted by

Snehal Keshav Nalawade **(202151160)**

Under the guidance of

Dr Bhanu Murthy

Indian Institute of information Technology Vadodara
(2021-2022)

Snehal Keshav Nalawade (202151160)
Section 2

Multiplication Algorithm

A multiplication algorithm is an algorithm (or method) to multiply two numbers. Depending on the size of the numbers, different algorithms are used. Efficient multiplication algorithms have existed since the advent of the decimal system.

- Throughout this report, we are discussing the binary multiplication with different bit size. Based on all these criteria the multipliers generally classified as signed and unsigned multipliers.

➢ Booth Multiplication Algorithm

The Booth multiplication algorithm defines a multiplication algorithm that can multiply two signed binary numbers in two’s complement.

Booth’s algorithm contains the addition of one of two predetermined values (A and S) to a product (P) continually and then implementing a rightward arithmetic shift on the product (P). Let us consider the predetermined values to be A and S, and the product to be P. Consider that the multiplicand and multiplier are m and r respectively. Let the number of bits in m and r be x and y respectively.

Example − Find the product of 3 x (-4), where m = 3, r = -4, x = 4 and y = -4.

A = 001100001

S = 110100000

P = 000011000

The loop has to be performed four times since y = 4. P = 000011000

Here, the last two bits are 00.

Therefore, P = 000001100 after performing the arithmetic right shift.

P = 000001100

Here, the last two bits are 00.

Therefore, P = 000000110 after performing the arithmetic right shift.

P = 000000110

Here, the last two bits are 10.

Therefore, P = P + S, which is 110100110.

P = 111010011 after performing the arithmetic right shift.

P = 111010011

Here, the last two bits are 11.

Therefore, P = 111101001 after performing the arithmetic right shift.

The product is 11110100 after dropping the LSB from P.

11110100 is the binary representation of -12.

➢ Karatsuba Multiplier Algorithm

The **Karatsuba algorithm** is a fast multiplication algorithm that uses a **divide and conquer approach** to multiply two numbers. It was discovered by Anatoly Karatsuba in 1960 and published in 1962.

This happens to be the first algorithm to demonstrate that multiplication can be performed at a lower complexity than O(N^2) which is by following the classical multiplication technique. Using this algorithm, multiplication of two n-digit numbers is reduced from O(N^2) to O(N^log3) that is O(N^1.585).
Basically Karatsuba stated that if we have to multiply two n-digit numbers x and y, this can be done with the following operations, assuming that B is the base of m and m < n (for instance: m = n/2)

- First both numbers **x** and **y** can be represented as **x1, x2** and **y1, y2** with the following formula.

$$x=x_1*B_m+x_2x$$
$$y=y_1*B_m+y_2y$$

The product x * y becomes the following product:

$$xy = (x_1*B^m+x_2)(y_1*B^m+y_2)$$

$$xy = x_1*y_1*B^{2m}+x_1*y_2*B^m+x_2*y_1*B^m+x_2*y_2$$

➢ Vedic multiplication Algorithm

The multiplication of two operands using VEDIC multiplier is achieved by multiplication by Vertically and Crosswise and then adding all the results. This multiplication algorithm can be understood using two operands 46 and 33. The operand 33 can be represented as 33 = (3×10 + 3) and 46 can be represented as 46 = (4×10 + 6). The multiplication (46×33) can be represented as (3×6 + 40×3 + 30×6 + 30×40). This multiplication is shown in Figure 1

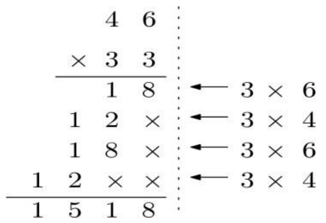

Figure 1: VEDIC Multiplication

- Similar way, this multiplication algorithm can be adopted to implement faster binary multiplier. A 4 -bit binary multiplication is shown below in Figure 2

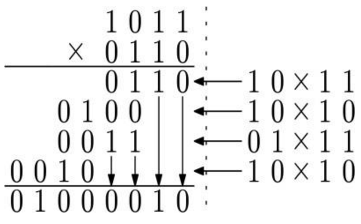

Figure 2: VEDIC multiplication for 4-bit data width.

Snehal Keshav Nalawade (202151160)
Section 2

- **Comparison in terms of Addition and Multiplication**

The number of addition and multiplication decide the speed of operation and the delay. The numbers of these blocks are also capable to increase or decrease the area power etc of the entire design. Comparing other multiplier, Vedic multiplier requires less Multiplication and addition steps.

Multiplier	8x8 bit		16x16 bit		32x32 bit	
	Multiplication	Addition	Multiplication	Addition	Multiplication	Addition
Booth Multiplier	40	26	96	72	288	243
Karatsuba Multiplier	15	10	40	24	198	75
Vedic Multiplier	8	4	16	8	32	16

(Comparison in terms of Addition andMultiplication)

- **Comparisons in Terms of LUT and Delay:-**

	2x2	4x4	8x8
Total no. of LUT	-	24000	28800
LUT used	-	6	20
Delay in (ns)	-	12.85	20.69

(Results of Booth Multiplier)

	2x2	4x4	8x8
Total no of LUT	9112	9312	9582
LUT used	5	14	22
Delay in (ns)	5.658	12.35	17.76

(Results of Karatsuba Multiplier)

	2x2	4x4	8x8
Total no. of LUT	9112	9312	9582
LUT used	4	11	18
Delay in (ns)	5.505	11.35	16.85

(Results of Vedic Multiplier)

LUT - Look Up Table

There are also few other algorithms like :

- Toom-Cook Algorithm : It is an algorithm for multiplying two n digit numbers in O(N^1.46) time, and
- Multiplying Floating Points Numbers

References :- Booth's Multiplication Algorithm - GeeksforGeeks
karatsuba algorithm for big integer multiplication » DrukLearn (drukinfotech.com)
Karatsuba Algorithm (for fast integer multiplication) (opengenus.org)
Toom Cook method for multiplication (opengenus.org)
Multiplying Floating Point Numbers - GeeksforGeeks
Multiplication Algorithm & Division Algorithm Notes - Computer Science Engineering (CSE) (edurev.in)

Snehal Keshav Nalawade (202151160)
Section 2

https://digitalsystemdesign.in/vedic-multiplier/
https://en.wikipedia.org/wiki/Karatsuba_algorithm
https://www.tutorialspoint.com/what-is-booth-multiplication-algorithm-in-computer-architecture

Thank you