AN EFFICIENT VLSI ARCHITECTURE FOR RECURSIVE KARATSUBA-OFMAN MULTIPLIER

G.Sreelakshmi, K.Ramya Prathima B.Harika Devi,
Associate Professor, B.Tech IVthyear, B.TechIVthyear,
ECE,Geethanjali College of Engineering and Technology,Hyderabad.

***Abstract*:** Among the four arithmetic operations multiplication is one of the basic operations. It can be explained as a repeated addition of multiplicand as the value of the multiplier. The finite field multiplication is the basicoperation in all cryptographic applications. It can be performedby using Conventional, Booth, Montgomery and Karatsuba-Ofman'sdivide-and-conquer technique. The Karatsuba-Ofman multiplierreplaces a multiplication by three ones of half-length operandswhich are performed in parallel. Area, power and delay computation of the proposed multipliers are improved.

If 'n' is four or more, the three multiplications in Karatsuba's basic step involve operands with fewer than n digits. Therefore, those products can be computed by recursive calls of the Karatsuba algorithm. The recursion can be applied until the numbers are so small that they can (or must) be computed directly.In this proposed project the recursive Karatsuba algorithm, is implemented using cadence digital encounter tools TSMC 0.18 use symbol micrometer technology for 2-bit, 4-bit, 8-bit, 16-bit, 32-bit, 64-bit and the results are compared with the existing multiplication algorithms – conventional multiplier and Booth's multiplier algorithm.

Keywords: Conventional, Booth, Recursive Karatsuba-Ofman Multipliers

I. INTRODUCTION

In this section, we introduce the fundamental recursive KOA which can successfully be applied to polynomial multiplication. The fundamental Karatsuba-Ofman multiplication is a recursive 'divide-and-conquer' technique. It is considered as one of the fastest way to multiply long numbers. For polynomial multiplication with original Karatsuba method both operands have to be divided into two equal parts. If the length of operands is odd, they have to be padded with leading '0'. Therefore, the KOA becomes recursive. A straightforward application of the KOA requires log2 (n) iteration steps for polynomials of the degree (n-1).

First the n-digit number is split into two (the first part of the number multiplied with some base and addedwith the second part). With the help of intermediate products and base number final result is arrived. AnatolyKaratsuba further reduced the number of multiplication steps by modifying one of the intermediate products wherethe number of multiplication steps can be reduced with the added complexity of addition operations. The complexity ofaddition operations is usually less than the complexity of multiplication operations. Thus the usage of Karatsuba algorithm increased in several advanced fields.

II. EXISTING ARCHITECTURES

1. CONVENTIONAL MULTIPLICATION

Conventional multiplication is much simpler as there is no table of multiplication to remember: just shifts and adds. This method is mathematically correct and has the advantage that a small CPU may perform the multiplication by using the shift and add features of its arithmetic logic unit rather than a specialized circuit.

The method is slow, however, as it involves many intermediate additions. These additions take a lot of time. Faster multipliers may be engineered in order to do fewer additions; a modern processor can multiply two 64-bit numbers with 6 additions (rather than 64), and can do several steps in parallel.

Example:

```
        1001  ( 9 in binary)
        x 1010  (10 in binary)
         =======
          0000
1001
 0000
+ 1001
 ==========
        01011010  (90 in binary)
       ==========
```

2. BOOTH MULTIPLICATION

Booth's multiplication algorithm was invented by Andrew Donald Booth in 1950. This multiplication algorithmmultiplies two signed binary numbers in two's complement notation. Radix-4 booth multiplier algorithm is a modified booth multiplier is used to perform high-speed multiplications using modified booth algorithm. This modified booth multiplier's computation time and the logarithm of the word length of operands are proportional to each other. We can reduce half the number of partial product.

Radix-4 booth algorithm used here increases the speed of multiplier and reduces the area of multiplier circuit. In this algorithm, every second column is taken and multiplied by 0 or +1 or +2 or -1 or -2 instead of multiplying with 0 or 1 after shifting and adding of every column of the booth multiplier. Thus, half of the partial product can be reduced using this booth algorithm. Based on the multiplier bits, the process of encoding the multiplicand is performed by radix-4 booth encoder.

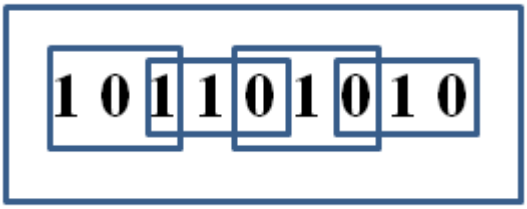

fig1: grouping of bits using radix-4 booth multiplier

The table below shows the functional operation of the radix-4 booth encoder that consists of eight different types of states.

table1: radix-4 booth multiplication

SELECT LINE(ENCODING)	PARTIAL PRODUCTS(OPERATION)
000	Add 0
001	Add multiplicand

010	Add multiplicand
011	Add 2*multiplicand
100	Subtract 2*multiplicand
101	Subtract multiplicand
110	Subtract multiplicand
111	Subtract 0

- **STEPS FOR RADIX-4 BOOTH MULTIPIER ALGORITHM**

1. Extend the sign bit 1 position if necessary to ensure that n is even.
2. Append a 0 to the right of the least significant bit of the booth multiplier.
3. According to the value of each vector, each partial product will be 0, +y, -y, +2y or -2y.

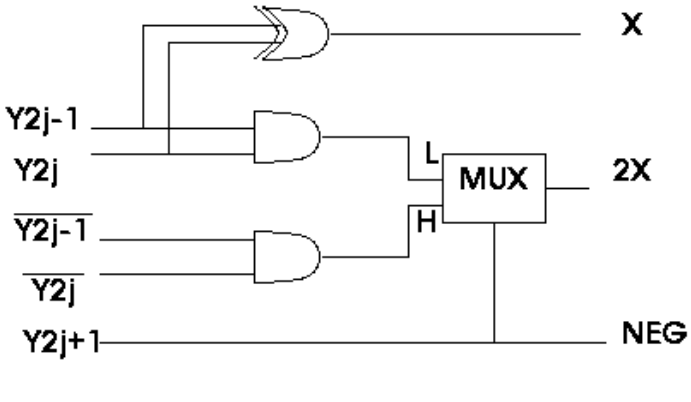

fig 2: booth encoder

III. PROPOSED ARCHITECTURE

1. KARATSUBA MULTIPLICATION

Let A,B be two elements then, both elements can be represented in polynomial basis as,

$$A = \sum_{i=0}^{m-1} a_i x^i = \sum_{i=n}^{m-1} a_i x^i + \sum_{i=0}^{n-1} a_i x^i = x^n \sum_{i=0}^{n-2} a_{i+n} x^i + \sum_{i=0}^{n-1} a_i x^i = x^n A^H + A^L$$

$$B = \sum_{i=0}^{m-1} b_i x^i = \sum_{i=n}^{m-1} b_i x^i + \sum_{i=0}^{n-1} b_i x^i = x^n \sum_{i=0}^{n-2} b_{i+n} x^i + \sum_{i=0}^{n-1} b_i x^i = x^n B^H + B^L$$

The Karatsuba-Ofman Multiplier (KOM) is based on the observation thatthe polynomial product C = A · B can be written as,

$$C = x^{2n}A^HB^H + (A^HB^L + A^LB^H)x^n + A^LB^L$$

$$= x^{2n}A^HB^H + A^LB^L + [(A^H + A^L)(B^L + B^H) - (A^HB^H + A^HB^H)]\, x^n$$

With the computational cost of three n-polynomial multiplications and 4 additions/ subtractions, By applying this strategy recursively, in the every iteration eachdegree polynomial multiplication is transformed into three polynomial multiplications with their degrees reduced to about half of its previous value.

2. RECURSIVE KARATSUBA MULTIPLICATION

In a simplified manner,

Let X, Y are the n-digit integers.

Desired output is T=X*Y.

STEP 1: X =10^n/2.X1 + X2; Y =10^n/2.Y1 + Y2.

(X1, X2, Y1, Y2 each have n/2 digits)

STEP 2: Let U = product of (X1, Y1).

STEP 3: Let V = product of (X2, Y2).

STEP 4: LetW = product of [(X1 + X2),(Y1 + Y2)].

STEP 5: Let Z= W-U-V.

STEP 6: Then T = 10^n.U + 10^n/2.Z + V

Thus T gives the Karatsuba multiplied result of X and Y.

If n isfour or more, the three multiplications in Karatsuba's basic step involve operands with fewer than n digits. Therefore, those products can be computed by recursive calls of the Karatsuba algorithm. The recursion can be applied until the numbers are so small that they can (or must) be computed directly.

2.1 Theoretical Example

(Of Recursive Karatsuba-Ofman Multiplication)

Compute 1234*4321

Sub-problems:

a1=12*43, d1=34*21

e1=(12+34)*(43+21)–a1–d1=46*64–a1–d1

need to recurse…….

First sub-problem:

a1=(12*43)

Sub-sub-problems:

a2=1*4=4, d2=2*3=6

e2=((1+2)(4+3))–a2–d2=11

Answer:$4*10^2 + 11*10 + 6=516$

Second sub-problem:

d1=(34*21)

Sub-sub-problems:

a2=3*2=6, d2=4*1=4

e2=((3+4)(2+1))–a2–d2=11

Answer:$6*10^2 + 11*10 + 4=714$

Third sub-problem:

e1=(46)*(64)–a1–d1

Sub-sub-problems:

a2=4*6=24,d2=6*4=24

e2=((4+6)(6+4))–a2–d2=52

Answer:24*102 + 52*10 + 24-714-516=1714

Final Answer: (1234)*(4321)

=(516)*(104)+(1714)*(102)+714

=5,332,114

IV. SIMULATION RESULTS:

1. Conventional Multiplier

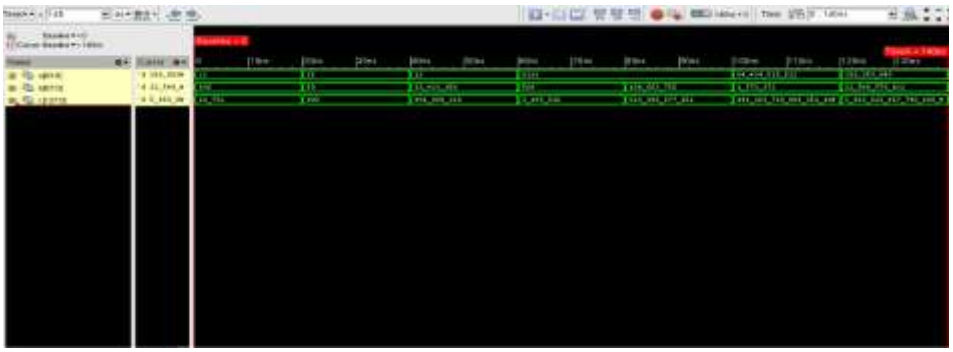

fig.3: conventional multiplier 64 x 64 bit simulation result

The above figure is simulated using cadence tool. It has two 64bit inputs, which are represented as a and b. One 128bit product represented as c.

fig.4: conventional multiplier 64 x 64 bit rtl schematic

The above figure is the result obtained when 64bit conventional code has been synthesized using cadence tool.

2. Booth Multiplier

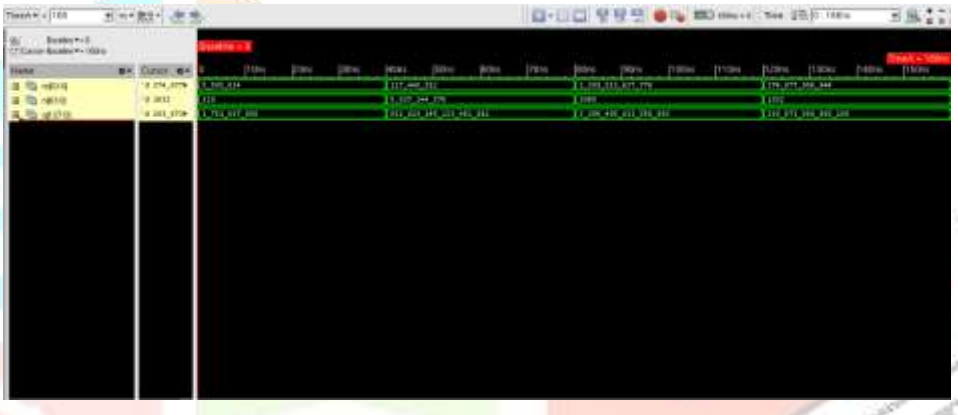

fig 5: booth multiplier 64 x 64 bit simulation result

The above figure is simulated using cadence tool. It has two 64 bit inputs, which are represented as m and n. One 128 bit product represented as q.

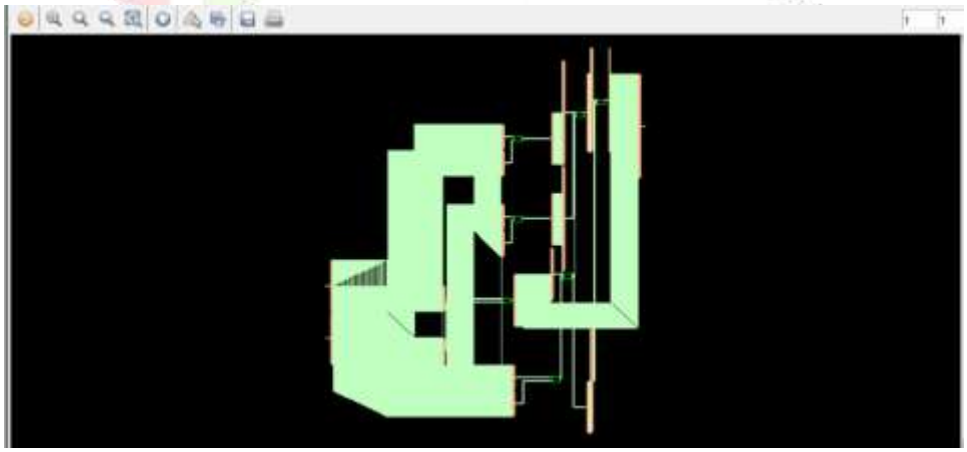

fig. 6: booth multiplier 64 x 64 bit rtl schematic

The above figure is the result obtained when 64 bit booth code has been synthesized using cadence tool.

3. RECURSIVE KARATSUBA-OFMAN MULTIPLIER

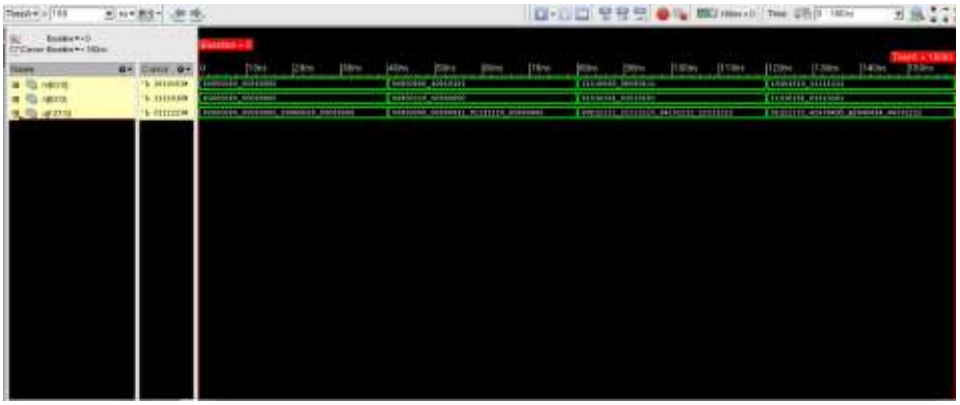

fig. 7: karatsuba-ofman multiplier 64 x 64 bit simulation result

The above figure is simulated using cadence tool. It has two 64 bit inputs, which are represented as m and n. One 64 bit product represented as q.

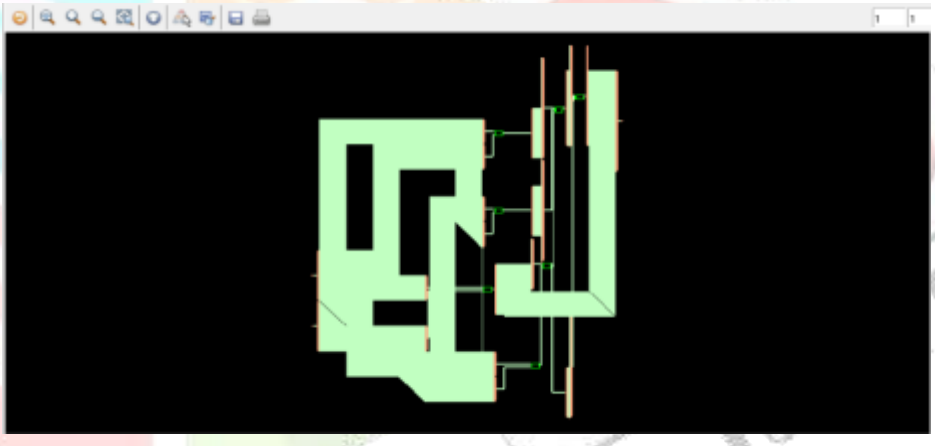

fig. 8: karatsuba-ofman multiplier 64 x 64 bit rtl schematic

The above figure is the result obtained when 64 bit recursive Karatsuba-ofman code has been synthesized using cadence tool.

V. SYNTHESIS REPORT:

(For 64 Bit*64 Bit Multiplication)

table 2: synthesis report comparison table

PARAMETERS	*CONVENTIONAL MULTIPLIER*	*BOOTH MULTIPLIER*	*RECURSIVE KARATSUBA-OFMAN*
POWER (nW)	133652552.198	87856664.115	87560553.919
AREA (um^2)	289610	387559	491662
TIMING (ps)	16404	28987	30297

VI. CONCLUSION:

Recursive Karatsuba-Ofman multiplier reduces the multiplication of two *n*digit numbers to at most single-digit multiplicationsRecursive Karatsuba-Ofman multiplier can successfully be applied to long integers multiplication step.

- The purpose of the proposed approaches is to obtain a

    1. Small Area occupation of the circuit due to the fact that partial

    2. Polynomial multiplications can be applied serially.

- The second purpose is to obtain a high speed computation by performing Parallel multiplication.
- The main purpose is to reduce the power consumption of recursive Karatsuba multiplier than that of the conventional multiplier