A Design of 4X4 Multiplier using 0.18 um Technology

Neha Maheshwari
Department of Electronics and Instrumentation Engineering
SVITS, Indore, M.P., India

Abstract - Designing low power high-speed arithmetic circuit requires a combination of techniques at four levels; algorithm, architecture, circuit and system levels. Digital multipliers are the most commonly used components in many digital circuit designs. Most high performance DSP systems rely on hardware multiplication to achieve high data throughput. There are various types of multipliers available depending upon the application in which they are used. This Paper focuses on an algorithm, called Tree Multiplication, which is suitable for high-speed and low-power applications. The algorithm is symmetric so it's very applicable for binary multiplication, due to the interchangeability of the multiplicand and the multiplier.

A Design of 4X4 Multiplier using 0.18 um Technology is successfully synthesized. Cadence Virtuoso 0.18 um Technology is used for simulation of the Design. The simulated transient output of 4X4 Multiplier is shown. Multiplier circuit work with 3.3 V power supply.

Keywords – Tree multiplier, full Adder, circuit

I. INTRODUCTION

1.1 INTRODUCTION TO MULTIPLIER

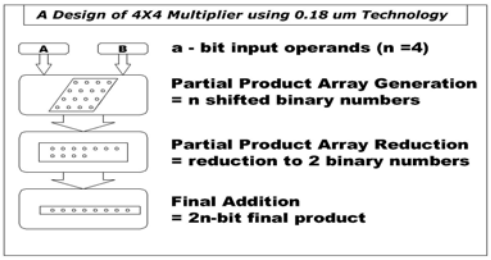
Multiplication is a mathematical operation that at its simplest is an abbreviated process of adding an integer to itself a specified number of times.

The operation of multiplication is rather simple in digital electronics. It has its origin from the classical algorithm for the product of two binary numbers. This algorithm uses addition and shift left operations to calculate the product of two numbers .Based upon the above procedure, we can deduce an algorithm for any kind of multiplication. We can check at the initial stage also that whether the product will be positive or negative or after getting the whole result, MSB of the results tells the sign of the product.

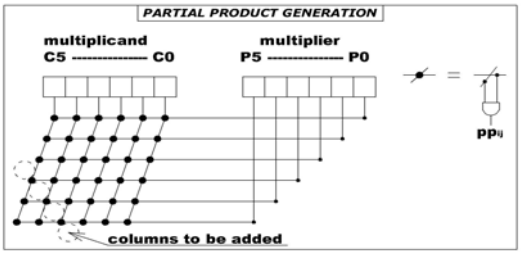
Digital multiplication is a series of bit shifts and bit additions, where two numbers, the multiplicand and the multiplier are combined into the result. Considering the bit representations of the multiplicand*X0X1 X2……Xn*-1 and the multiplier*Y0Y1Y2….Yn*-1, in order to form the product, up to n shifted copies of the multiplicand is to be added for unsigned multiplication. The entire process consists of three steps, partial product generation, partial product reduction and final addition.

Digital multiplication consists of three basic steps, these are:-

1. Generation of Partial Product Array
2. Reduction of Partial Product Array
3. Final addition



In digital multiplication, as an initial step, one needs to generate n shifted copies of the multiplicand, which may be added in the coming stage. The value of the
Multiplier bit determines whether the shifted copy is to be added or not: if the i^{th} bit ($0 \le i \le n-1$) of the multiplieris'1', then the shifted copy of the multiplicand is added. If the bit is'0', it's not added. A logical AND gate can implement this operation, by performing the function AND (x_i y_j) ($0 \le i \le n-1 \& 0 \le y \le n-1$). The resulting values are called partial products.
A trapezoidal structure, called partial product array (PPA), where the partial product bits are arranged in columns to be added in order to form the product. This process is called product array generation.

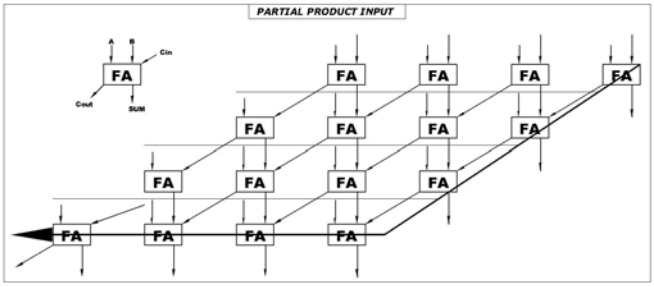


Partial Product Generation

Considering the addition of two bits from two vectors, X and Y, where numerical bit vector representations are of the form X=xn-1xn-2……x1 x0 and Y=yn-1yn-2... y1 y0, conventional full-adder can be used, which takes in three bits and outputs a sum and a carry bit, so the block adds two bits at a given position with the carry in from the previous bit position. Considering the case of adding two bit vectors, two bits are added at the lowest bit position and the carry is propagated to the next bit position. At the higher positions, two inputs and the carry bit are to be combined and a carry out is generated. This rippling technique of adding two n- bit numbers requires O(n) sequential bit additions hence a delay of O(n). For the addition of three n-bit vectors X, Y and Z ,this method can be used to add X and Y, then to add Z to the sum of X+Y; so the total number of bit additions of n shifted copies of an n-bit multiplicand is O(n) where the total delay is O (n2),assuming that the add operations are dependent on the previous ones since the output of earlier operations are inputs to later operations.
Even though the result comes from the combination of all operations, a certain amount of independence exists between each operation, considering the addition on a particular column. All the bits in a column must be added together along with the carry in bits coming from the previous column. Carry save addition implies that addition in separate columns can be performed independently without concerning about carry from previous column. There are several ways to implement addition of partial products in the trapezoidal array. This also forms the basis of classification among parallel multipliers.
In the following section we would discuss the two most commonly used methods
1. Array Multiplier
2. Tree Multiplier

1.2 ARRAY MULTIPLIER

Array multiplier is well known due to its regular structure. In array multiplier, the counters and compressors are connected in a serial fashion for all bit slices of the Partial Product parallelogram. The array topology is a two-dimensional structure that fits nicely on the VLSI planar process. There are several possible array topologies including simple, double and higher order arrays.



The array multiplier originates from the multiplication parallelogram. As shown in Figure, each stage of the parallel adders should receive some partial product inputs. The carry-out is propagated into the next row. The bold line is the critical path of the multiplier. In array multiplier, all of the partial products are generated at the same time. It is observed that the critical path consists of two parts: vertical and horizontal. Both have the same delay in terms of full adder delays and gate delays. For an n-bit by n-bit array multiplier, the vertical and the horizontal delay s are both the same as the delay of an n-bit full adder. For a16- bit multiplier, the worst-case delay is 32 τ where τ is the worst-case full adder delay.

Advantage of the array multiplier comes from its regular structure. Since it is regular, it is easy to layout and has a small size. The design time of an array multiplier is much less than that of a tree multiplier.

Its ease of design for a pipelined architecture. A fully pipelined array multiplier with a stage delay equal to the delay of a 1-bit full adder plus a register has been successfully designed for high- speed DSP applications. Also it can be easily pipelined by inserting Latches after CSA (carry save adders) or after every few rows.

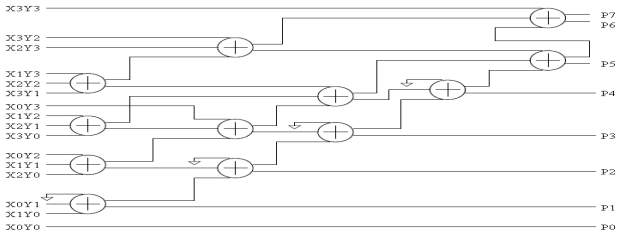
The biggest problem with full linear array multipliers is that they are very large. As operand sizes increase, linear arrays grow in size at a rate equal to the square of the operand size. This is because the number of rows in the array is equal to the length of the multiplier, with the width of each row equal to the width of multiplicand. The large size of full arrays typically prohibits their use, except for small operand sizes, or on special purpose math chips where a major portion of the silicon are a can be assigned to the multiplier array. Another problem with array multipliers is that the hardware is underutilized. As the sum is propagated down through the array, each row of CSA's computes a result only once, when the active computation front passes that row. Thus, the hardware is doing useful work only a very small percentage of the time. This low hardware utilization in conventional linear array multipliers makes Performance gains possible through increased efficiency.

1.3 TREE MULTIPLIER

Trees are an extremely fast structure for summing partial-products. In a linear array, each row sums one additional partial product .As such, linear arrays require order N stages to reduce N partial-products. In contrast, by doing the additions in parallel, tree structures require only order log N stages to reduce N partial products.

The result of the multiplication is obtained by first generating partial products and then adding the partial products. The critical path of a multiplier depends on the delay of the carry chain through all of the adders. Table shows the truth table of a full adder, which is the basic addition process usually employed in a computer to add two numbers together. A and B are the adder inputs, and C is the carry input. The full adder produces a bit of sum and a bit of carry out. It can be observed that a full adder is actually a"one's counter". A , B and C can all be seen as the inputs of[3:2] compressor. The outputs, Carry and Sum, are the encoded output of the three inputs in binary notation .The tree multiplier is based on this property of the full adder. The addition of sum and scan be accelerated by adopting a [3:2] compressor [7]. A [3:2] compressor adds three bits and produces a two-bit binary number whose value is equal to that of the original three. The advantage of the [3:2] compressor is that it can operate without carry propagation along its digital stages and hence is much faster than the conventional adder. In any scheme employing[3:2]compressors, the number of adder passes occurring in a multiplication before the product is reduced

to the sum of two numbers, will be two less than the number of summands, since each pass through an adder converts three numbers to two, reducing the count of numbers by one. To improve the speed of the multiplication, one must arrange many of these passes to occur simultaneously by providing several [3:2] compressors
Thus, the best first step for a tree multiplier is to group the summand into threes, and introduce each group into its own [3:2] compressor, thus reducing the count of numbers by a factor of1.5. The second step is to group the numbers resulting from the first step into threes and again add each group in its own [3:2] compressors. By continuing such steps until only two numbers remain, the addition is completed in a time proportional to the logarithm of the number of summands **Figure** shows a4-bit tree multiplier. It should be noted that[4:2] compression can also be used besides the[3:2] compression. A[4:2] compression can be achieved by combining two fulladders.Fora16-bit multiplier, the worst- case delay is7τfa+τcpawhere τfa is the delay [8]ofa3-2compressor,which is the delay of a full adder, and τcpa is the delay of the fast carry propagate adder. To improve the speed of the multiplication, one must arrange many of these passes to occur simultaneously by providing several [3:2] compressors. Thus, the best first step for a tree multiplier is to group the summands into threes, and introduce each group into its own [3:2] compressors, thus reducing the count of numbers by a factor of1.5.Thesecondstepistogroup the numbers resulting from the first step into threes and again add each group in its own [3:2] compressors. By continuing such steps until only two numbers remain, the addition is completed in a time proportional to the logarithm of the number of summands.



II. CIRCUIT DESIGN

2.I Full Adder

The adder is the basic element in computer arithmetic. It is also the critical element of the multiplier. The truth table of a one-bit full adder is shown on Table .We can derive the Boolean function in sum of products from this truth table as the following:

SUM= ABC+A'B'C+AB'C'+A'BC'
CARRY=AB+BC+AC

A and B are the adder inputs, C is the carry input, SUM is the sum output, and CARRY is the carry output. The generate signal, G, occurs when a carry output (CARRY) is internally generated within the adder. When the propagate signal, P, is true, the carry-in signal (C) is passed to the carry output (CARRY) when C is true. Because the carry-in of the current bit is determined by lower bits of two operands, the delay of the adder depends on the generation of the carry.

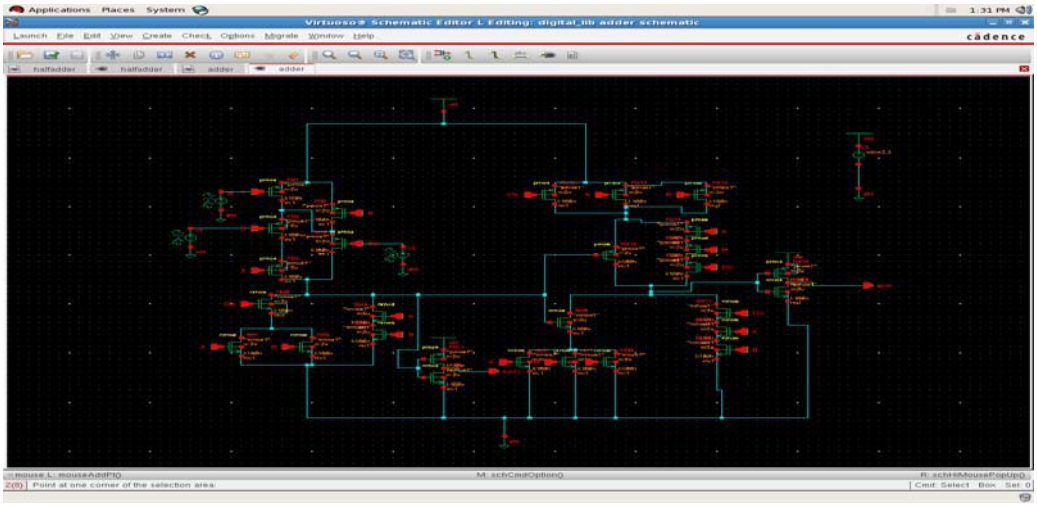
TRUTH TABLE OF FULL ADDER CIRCUIT:

TRUTH TABLE OF FULL ADDER CIRCUIT

A	B	C_i	C_o	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

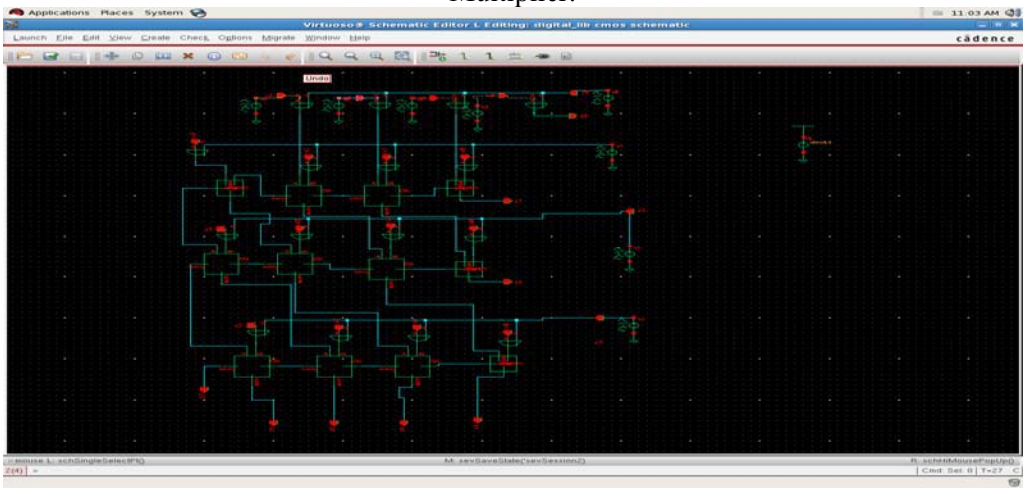
2.2. Schematics

Full Adder:



SCHEMATIC OF FULL ADDER

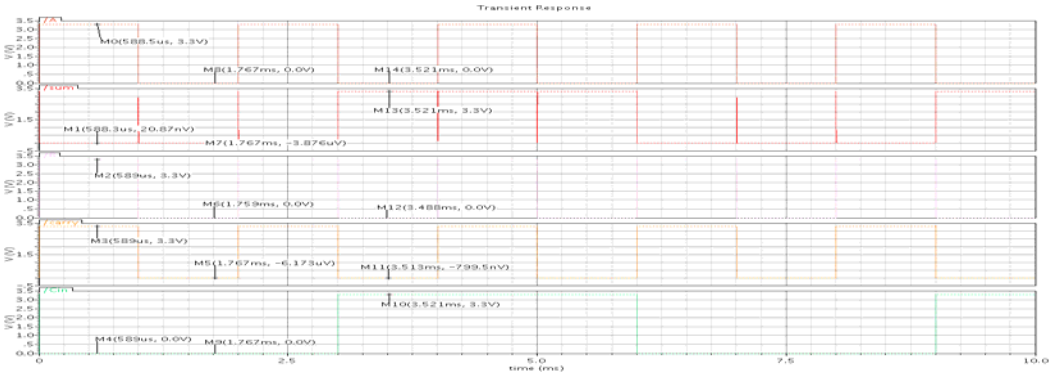
Multiplier:



III. Experiment and Result

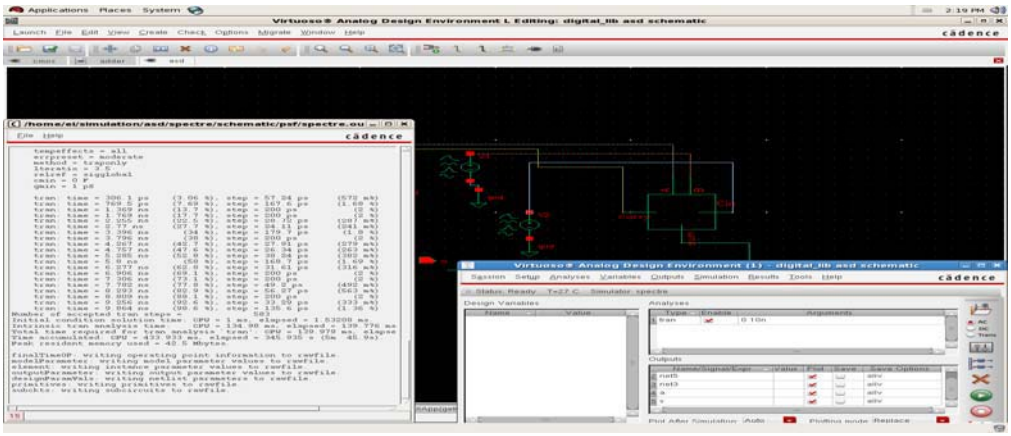
Simulation Results:

1. Analysis of full adder circuit:

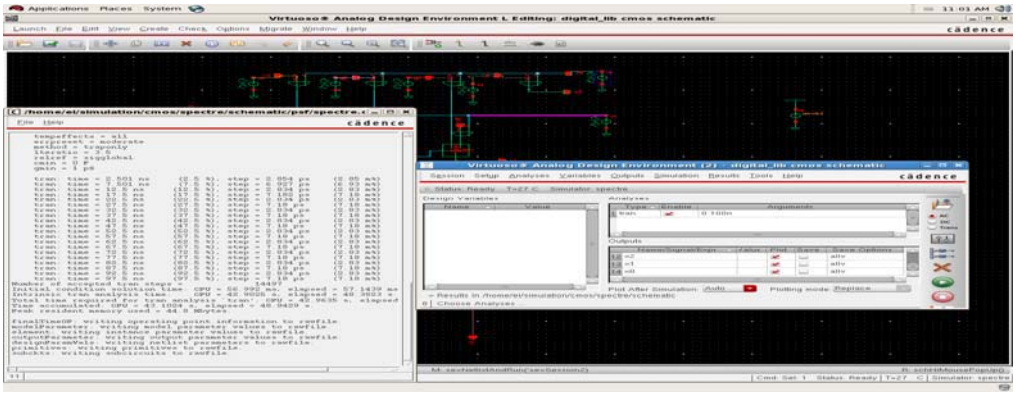


4X4 Multiplier Responce:

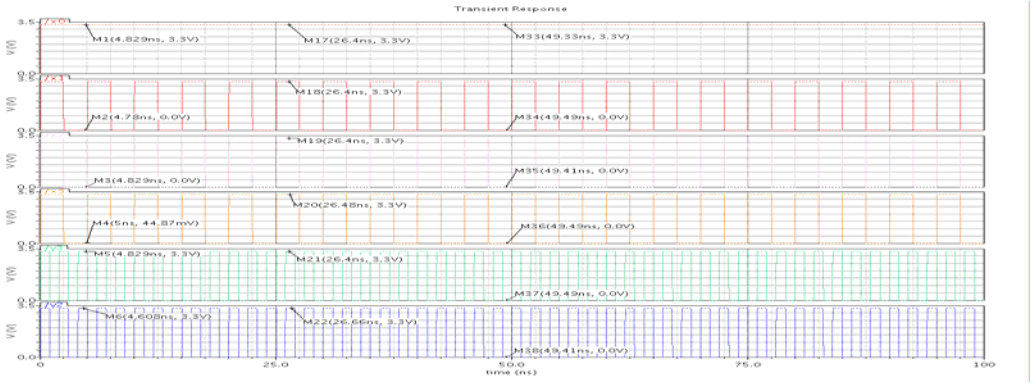
Analysis of Full adder circuit:

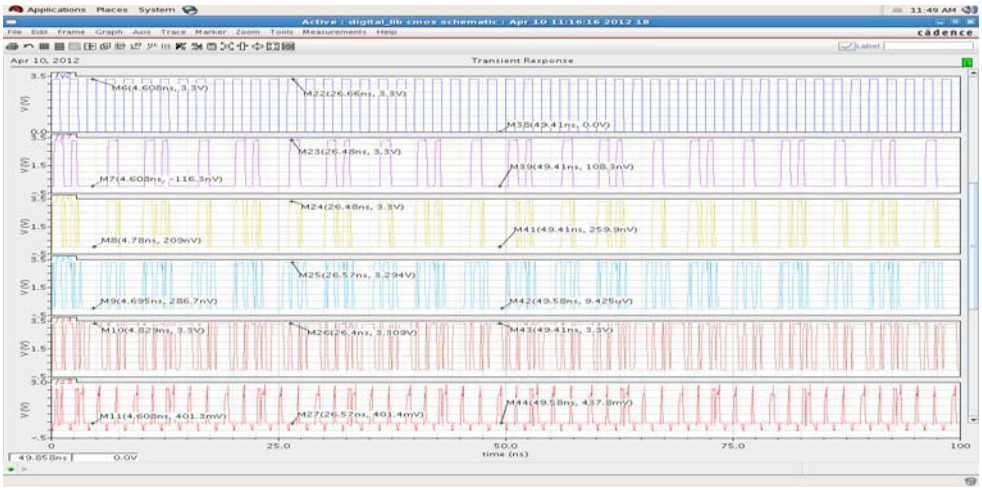


Analysis of 4X4 Multiplier:



Simulated Waveform of 4X4 Multiplier





IV. CONCLUSION

Multiplication is one of the basic functions used in digital signal processing. Most high performance DSP systems rely on hardware multiplication to achieve high data throughput.

A Design of 4X4 Multiplier using 0.18 um Technology is successfully synthesized. Cadence Virtuoso 0.18 um Technology is used for simulation of the Design.The simulated transient output of 4X4 Multiplier is shown. Multiplier circuit work with 3.3 V power supply.

REFERENCES

[1] S.ShahandA.J.Al-Khalili,"Comparisonof32-bitmultipliersforvarious performance measures", *12thInternational conference on Micro electronics* , Tehran,2000.
[2] K. Roy and S. C. Prasad," Low- Power CMOS VLSI Circuit Design*"* ,John Wiley & Sons, 1999
[3] C.S. Wallace."A Suggestion for a Fast Multiplier*", IEEE Transactions on Electronic Computers*,EC-13:14–17,February1964.
[4] P.C.H. Meier," Analysis and Design of Low Power Digital Multipliers", Ph.D. Thesis, Carnegie Mellon University, Dept. of Electrical and Computer Engineering, Pittsburgh, Pennsylvania, 1999.
[5] W. Wolf, "Modern VLSI Design Systems on Silicon*"*, Upper Saddle River: Prentice Hall, 1998.
[6] Biswas, Kelvin, "Multiplexer-Based Array Multipl*iers*" Research work 2005 at university of Windsor
[7] Santaro, "Design and clocking of VLSI Multipliers" *Technical report* October, 1989.
[8] K. Myeir, "Advanced Computer Arithmetic EE486" *IEEE Transactions on Electronic Computers*, EC-13:14–17, Feb,2003.
[9] K. Roy and Yeo, Kiat - Seng, "Low voltage, Low-power VLSI Subsystems", McGraw-Hillpp.124-141.