VHDL Implementation of Fast Multiplier based on Vedic Mathematic using Modified Square Root Carry Select Adder

Heena Goyal
Department Of Electronics, IGIT, GGSIP University, Delhi, India.

Shamim Akhter, PhD
Assistance Professor, ECE Department
Jaypee Institute of Information Technology, India

ABSTRACT
In this paper, a novel technique for multiplication is presented using Vedic multiplier. Vedic multiplier uses adders and hence making fast adder will increase the overall speed for multiplication. We have done comparative analysis for multiplication using different architectures of adder. For comparison we have taken Carry Select Adder (CSA), Square Root Carry Select Adder (SQRT-CSA). We have proposed Vedic multiplication using Modified SQRT-CSA. VHDL design in proposed and synthesis is performed on Virtex-4 FPGA.

Keywords
Vedic Multiplier, CSA, SQRT-CSA, Modified SQRT-CSA, Binary-to Excess One (BEC) block

1. INTRODUCTION
Many digital signals processing operation requires several multiplication and for the same we need fast multipliers. This paper presents a systematic design methodology for fast and area efficient multiplier based on Vedic mathematics [1].The Multiplier Architecture is based on the Vertical and Crosswise algorithm of ancient Indian Vedic Mathematics. The adder block used during Vedic Algorithm based computation is main source of delay. We have compared multiplication operation different topologies of adder like Ripple Carry Adder (RCA), Carry Look-ahead Adder (CLA), Carry Select Adder (CSA), Square Root CSA (SQRT-CSA) and modified SQRT-CSA.

The organization of this paper is as follows. Section-I present logic for Vedic Algorithm based multiplication operation. Section-II deals with the analysis of Adder using different topologies. Section –III deals with analysis and designing of modified SQRT-CSA. Section-IV deals with the simulation and synthesis results followed by conclusion and references.

2. VEDIC ALGORITHM BASED MULTIPLICATION
The VHDL design of Vedic multiplier is proposed in [1]. It represents the modular design. The basic module is 2x2 multiplier. Using this module, the paper [1] presents the design of 4x4 multiplier based on Vedic logic. Let's analyze 4x4 multiplications of A and B, each of 4-bits. Taking A= $A_3A_2A_1A_0$ and B= $B_3B_2B_1B_0$ and the multiplication result as,

$S_7S_6S_5S_4S_3S_2S_1S_0$. Divide A and B into two parts as A_3A_2 and A_1A_0 for A and B_3B_2 and B_1B_0 for B. Using the fundamental of Vedic multiplication, taking two bit at a time and using 2-bit multiplier block, we can have the following structure for multiplication.

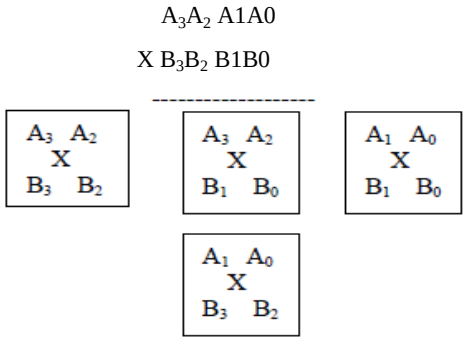

Figure 1. Block diagram for 4x4 bit multiplication [1]

The outputs of first 2x2multiplier block with inputs A_1A_0 and B_1 B_0 are taken $S_{temp13}S_{temp12}S_{temp11}S_{temp10}$.

The outputs of last 2x2 multiplier block with inputs A_3A_2 and B_3B_2 are taken $S_{temp43}S_{temp42}S_{temp41}S_{temp40}$.

We have remaining two blocks of 2x2 Multiplier. First block with inputs A_3A_2 and B_1B_0 and having output as $S_{temp23}S_{temp22}S_{temp21}S_{temp20}$. Second block with inputs A_1A_0 and B_3B_2 and having output as $S_{temp33}S_{temp32}S_{temp31}S_{temp30}$.

The final multiplication result in 8 bit i.e. $S_7S_6S_5S_4S_3S_2S_1S_0$ is obtained as discussed below [1].

Two LSB bits of first 2x2 bit Multiplier output i.e. S_{temp11} and S_{temp10} forms the result bits S_1 and S_0 respectively.

A 4-bit full adder is used to add $S_{temp23}S_{temp22}S_{temp21}S_{temp20}$ and $S_{temp33}S_{temp32}S_{temp31}S_{temp30}$ with carry out from this addition as C_{out1}. The result of this addition is given as one set of 4- bit input to the next adder with the 2nd set of input as $S_{temp41}S_{temp40}S_{temp13}S_{temp12}$. Carry out from this addition is taken as C_{out2} with sum as final result bits $S_5S_4S_3S_2$.

The carry out signals C_{out1} and $C_{out2\ is}$ given to half adder and the outputs of this half adder is taken as $C_{out3}S_{temp5}$. The carry and sum from the half adder i.e. $C_{out3}S_{temp5}$ form one set of input of 2-bit full adder with second set of input as S_{temp43} and S_{temp42} gives the remaining result bit as S_7S_6.

The above logic can be extended for higher order bits in input and the same is explained below.

For NxN multiplication, divide the multiplicand and multiplier into two parts, consisting of (N to N/2-1) bits and (N/2 to 1) bits. For example, for multiplication of A and B, of 16 bit each, if A=0000001010101101, then its part will be 00000010, and 10101101. Similarly divided B is equal parts, i.e. if B= 0010001110101011, then its part will be 00100011 and 10101011. Represent the above mentioned parts of A as

A_M and A_L. Similarly for the input B, it is divided in two parts as B_M and B_L. Now represent A and B as A_MA_L and B_MB_L. For AXB, we can have

$$\begin{array}{l} A_M\ A_L \\ B_M\ B_L \\ \hline A_M \times B_M \downarrow \quad A_M \times B_L \quad A_L \times B_L \downarrow \\ \qquad\quad A_L \times B_M \downarrow \end{array}$$

Figure 2. General representation for Vedic multiplication [1]

Let's say, the overall result is to be represented in $S_{31}S_{30}\ldots\ldots S_1 S_0$. The output of partial product A_M x B_M, is $S_{MM15}S_{MM14}\ldots\ldots S_{MM1}S_{MM0}$. Similarly the outputs of partial products for A_M x B_L, A_L x B_M, and A_L x B_L, are taken as $S_{ML15}\ldots\ldots\ldots S_{ML0}$, $S_{LM15}\ldots\ldots\ldots S_{LM0}$, $S_{LL15}\ldots\ldots\ldots S_{LL0}$ respectively. The complete block diagram for the same is given below [1].

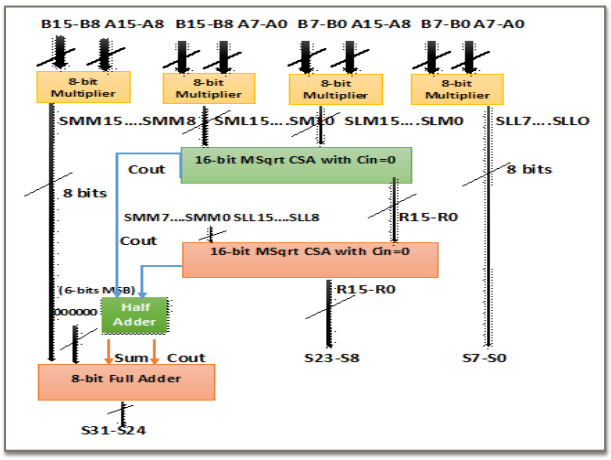

Figure 3. Block Diagram for 16-bit Multiplication Using Vedic Multiplication [1]

3. DIFFERENT TOPOLOGIES FOR ADDER

In our work, we have analyzed the multiplication architecture by using different topologies of adder. We have considered Ripple Carry Adder (RCA), Carry Look-ahead Adder (CLA), CSA, SQRT-CSA [2-4]. In this work, we have proposed Vedic multiplication by using modified SQRT-CSA adder. RCA and CLA are basic adders and can be found in literature very well. In next sub-section, we have explained the working of CSA and SQRT-CSA since it is the basic of Modified SQRT-CSA.

3.1 Working of Carry Select Adder (CSA)

The carry-select adder consists of two ripple carry adders and a set of multiplexers. The block diagram for 4-bit addition using CSA is given in Figure 4. For adding two 4-bit numbers using CSA, we require two 4-bit full adders and that can be Ripple Carry Adder (RCA) or Carry Look-Ahead Adder (CLA). 1st stage computes the 4-bit addition using Cin=0 and 2nd stage computes the same with Cin=1. After the two results are available, the correct sum and carry out is then decided by the multiplexer once the correct carry is known [4]. For designing 16-bit adder, we can cascade the structure shown in Figure.4. It means we need four stages of CSA to create 16-bit CSA adder. The number of bits in each CSA block is uniform. It is made variable in case of SQRT-CSA to get optimal delay [4].

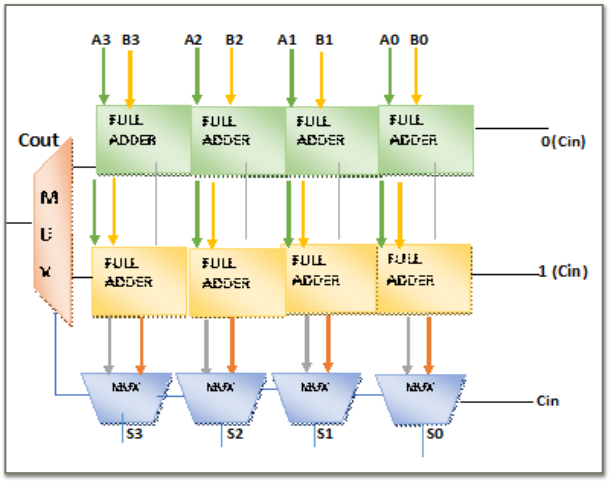

Figure 4. Block Diagram for 4-bit CSA

3.2. Working of Square-Root Carry Select Adder (SQRT-CSA)

A 16-bit CSA with variable size is created by cascading four CSA with variable input size. The block diagram for the same is given below.

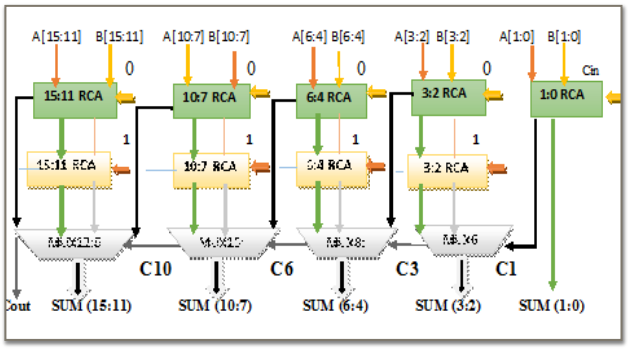

Figure 5. Block Diagram for 4-bit SQRT-CSA [5]

As shown in Figure.5, we have five stages of CSA with different inputs bits to each stage. We have an adder with block sizes of 2-2-3-4-5 respectively. In the above structure the adder module with Cin=1 is replaced with BEC (Binary-To-Excess one) module [2, 5]. This design was named Modified QSRT-CSA [2, 5]. The analysis for the same is given in next section.

4. ANALYSIS AND DESIGNING OF MODIFIED SQRT-CSA

Conventional CSA with Cin=1 block is replaced with binary-to-excess-1 converter (BEC) in the modified SQRT CSA structure. The architecture of 16-bit CSA is configured into five different stages with progressively increasing data size as given in Figure.5. The design steps for 4-bit BEC block is discussed in next section. The design can be extended for other bit size easily.

4.1 Binary-to-Excess-1 Converter (BEC)

This block is responsible for adding one ('1') to the input data so that the RCA with Cin=1 can be replaced. The design for the same is explained below for 4-Bit BEC block which takes four inputs and delivers four bit output.

Table 1: 4-Bit BEC Truth Table

Input				Output			
B_3	B_2	B_1	B_0	X_3	X_2	X_1	X_0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
1	1	1	1	0	0	0	0

The Boolean expressions for X_0, X_1, X_2 and X_3 as derived from K-Map are:

X_0= NOT(B_0)
X_1= XOR (B_0,B_1)
X_2= XOR[B_2, (B_0 AND B_1)]
X_3= XOR[B_3, (B_0 AND B_1 AND B_2)]

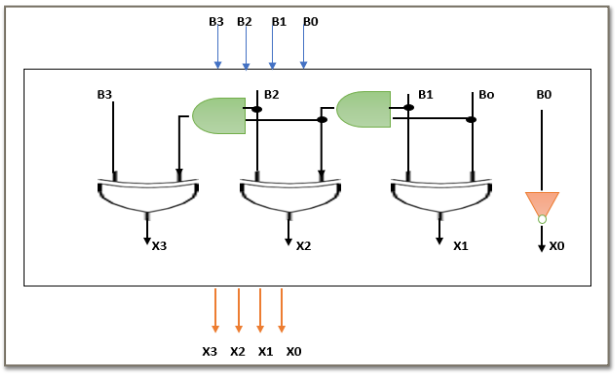

Figure 6: 4-bit binary-to-excess-1 converter [2]

This design can be extended for differnt input size length.

4.2 16-bit Modified SQRT-CSA

The complete block diagram for 16-bit Modified SQRT-CSA based on BEC is shown in Figure 7. The components used in the above designed is coded in VHDL and using structural modelling all block are combined. The BEC input bits are output of adder stage with Cin='0'. For example, for 4-Bit BEC, inputs are {cout sum(6) sum(5) sum(4)} of 3rd stage. Similarly for 5-Bit BEC, inputs are {cout sum(10) sum(9) sum(8) sum(7)} of 4th stage.

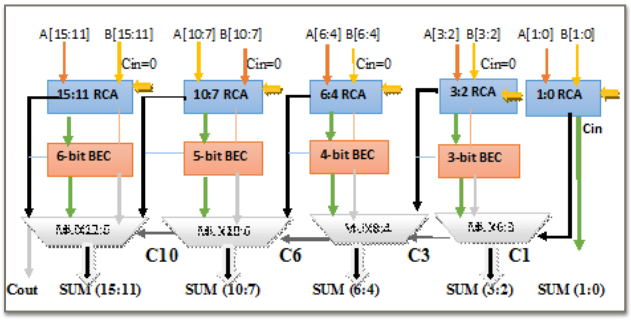

Figure 7: Block diagram of modified 16-bit SQRT CSA [2]

4.3 16-bit Multiplier based on Vedic Logic using Modified SQRT-CSA

As shown in Figure.1, the 4x4 bit multiplication requires basic2x2 multiplier. In addition to this, Vedic multiplication requires two 4-bit Full Adder, and one Half Adder and One 2-bit adder. The Table 2 given below shows the design requirements for the higher order multiplication in terms of input bit size.

Table 2: Design requirements for different input bit length

Multiplication Bit	Multiplier Needed	Adder Requirements		
4x4	2x2	Two, 4-bit FA	One, 2-bit FA	One, HA
8x8	4x4	Two, 8-bit FA	One, 4-bit FA	One, HA
16x16	8x8	Two, 16-bit FA	One, 8-bit FA	One, HA
32x32	16x16	Two, 32-bit FA	One, 16-bit FA	One, HA

Since the advantages of CSA, SQRT-CSA and Modified SQRT-CSA in terms of delay is visible for large sizes of input bit length. We have designed the 4x4, 8x8, 16x16 and 32x32 bit multiplication based on Vedic multiplier using different topologies of adder i.e. CLA, CSA, SQRT-CSA and Modified SQRT-CSA. The simulation is performed in ModelSim and synthesis is done using Xilinx for Vertex 4 (ML 402).

5. SIMULATION AND SYNTHESIS RESULTS

Simulation is performed for different set of input data. Simulation waveform from ModelSim simulator can't be pasted due to large no of signal count. So we have given the equivalent numeric values for two sets of input data.

1st set of input data and result

Input A: 0000 1100 1011 1100 (3260),

Input B: 0000 0000 0100 0011(67),

Result: 00000000000000110101010100110100 (218420)

2nd set of input data and result

Input A: 0000 0100 1011 0000 (1200),

Input B: 0010 0000 0100 0011(8259),

Result: 00000000100101110011101000010000 (9910800)

The synthesis result is tabulated for 8-bit, 16-bit and 32-bit Vedic multiplier based on CLA, CSA, SQRT-CSA, and MSQRT-CSA.

Table 3: Synthesis Result Comparison 8-bit Multiplication

Topologies for 8-bit Vedic Multiplier	Delay(ns)	No of Slices	No of LUT's
CLA	17.567	93	162
CSA	18.240	126	224
SQRT-CSA	20.129	120	213
MSQRT-CSA	20.213	124	221

Table 4: Synthesis Result Comparison 16-bit Multiplication

Topologies for 16-bit Vedic Multiplier	Delay(ns)	No of Slices	No of LUT's
CLA	30.216	411	716
CSA	28.528	541	969
SQRT-CSA	31.056	538	954
MSQRT-CSA	29.867	557	990

Table 5: Synthesis Result Comparison 32-bit Multiplication

Topologies for 32-bit Vedic Multiplier	Delay(ns)	No of Slices	No of LUT's
CLA	55.772	2040	3574
CSA	47.349	2368	4216
SQRT-CSA	53.037	2229	3950
MSQRT-CSA	47.173	2362	4194

It can be seen that Vedic multiplication based on MSQRT-CSA is more efficient as compared to CSA and SQRT-CSA in terms of delay.

6. CONCLUSION

In this paper, a fast technique of digital multiplication is presented using Modified SQRT-CSA. The design is based on Vedic method of multiplication and adder is the main component for the delay. Hence we have done delay analysis of multiplication with different adder topologies and come to conclusion that Modified SQRT-CSA based multiplication is fast. Due to modular design approach complexity gets reduced for large inputs bit length.

7. REFERENCES

[1] Shamim Akhter, "VHDL Implementation of Fast NxN Multiplier Based on Vedic Mathematic," Proc. of 18th European Conference on Circuit Theory and Design (ECCTD), pp. 472-475, Aug. 2007.

[2] Y. He, C.H. Chang, and J. Gu, "An Area Efficient 64-Bit Square Root Carry-Select Adder for Low Power Applications," Proc. of IEEE International Symposium on Circuits and Systems (ISCAS-2005), vol. 4, pp. 4082-4085, May 2005.

[3] N. H. E. Weste, D. Harris, and A. Banerjee, "CMOS VLSI Design: A Circuits and Systems Perspective," Pearson Education, 3rd Edition, 2005.

[4] B. Ramkumar, and H. M. Kittur, "Low-Power and Area-Efficient Carry Select Adder," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 2, pp. 371-375, Feb. 2012.

[5] Shamim Akhter, S. Chaturvedi, and K.Pardhasardi, "CMOS Implementation of Efficient 16-Bit Square Root Carry-Select Adder," 2nd International Conference on Signal Processing and Integrated Networks (SPIN), India, Noida, pp. 891 – 896, Feb 2015.

[6] G.Singh , "Design of Low Area and Low Power Modified 32-BIT Square Root Carry Select Adder", International Journal of Engineering Research and General Science Volume 2, Issue 4, 2014, pp-422-431.