EasyChair Preprint

№ 7455

Binary Multiplier Circuit Based on Vedic Mathematics

Jerubandi Ravi Teja

February 13, 2022

Binary multiplier circuit based on Vedic mathematics

***Abstract*: In the literature, Vedic multipliers for 4-bit, 8-bit, 16-bit, and 32-bit are available, but their design involves a larger number of adders, which consumes a considerable portion of the FPGA and causes more delay during implementation. This paper presents a Vedic multiplier based on the Urdhva Tiryagbhyam sutra with certain adjustments in the technique of partial product addition to solve this problem. Xilinx Vivado tool is used to simulate the proposed design, which is written in Verilog HDL. For 4-bit, 8-bit, 16-bit, and 32-bit input, the proposed Vedic multiplier design is simulated. The proposed design's performance is compared to that of existing designs in terms of speed and LUTs. In terms of time and the number of LUTs, this Vedic multiplier has improved.**

Keywords: Vedic multiplier, Verilog HDL, Carry Save Adder, Ripple Carry Adder.

I. INTRODUCTION

In arithmetic and logical operations, multiplication is a crucial fundamental function. Microprocessors, microcontrollers, and a variety of other digital circuits do multiplication. The multipliers using ordinary methods for multiplication require more area and provide more delay during implementation, so the ancient Vedic mathematical formulation can be applied to attain less area and to provide less delay during implementation of multipliers. The main focus of Vedic Mathematics is on sixteen vital principles called as Sutras. The Urdhva Tiryagbhyam sutra is one of the 16 sutras that had been employed in the literature [1-3] to reduce the area and delay of multipliers during implementation. The existing architectures in [1-3] are different only in performing the addition of the partial products. To add the partial products, Carry Save Adders (CSA's) along with OR gate and Ripple Carry Adder (RCA) are used in [1]. Different sizes of RCAs are used in [2]. In [3] three RCAs are used. The main drawback in existing architectures [1-3] is that the hardware to add the partial products is not used efficiently, so they consume a larger part of the FPGA while implementation and provide more delay. Therefore, an architecture is needed that consumes less area and provides less delay. This paper proposes a simple architecture of the Vedic multiplier that uses one CSA for the addition of partial products.

The following is the structure of this paper: Section II explains the Urdhva Tiryagbhyam sutra's approach for Vedic multiplication as well as the architecture of adders. Section III discusses the architectures and disadvantages of existing Vedic multipliers. In Section IV, the proposed Vedic multiplier is presented.In Section V, simulation results are discussed. Finally, Section VI concludes the paper.

II. METHODOLOGY

In the literature [1-3], the Urdhva Tiryagbhyam sutra is used to do multiplication operations. Urdhva Tiryagbhyam is a Sanskrit word that means "vertically and crosswise." The following is the procedure for multiplication utilizing the Urdhva Tiryagbhyam sutra:

In the first step, multiply the left-hand most digit of the multiplicand with the multiplier's left-hand most digit. In the result, the product is written as the first digit.

Multiply the multiplicand's second digit by the multiplier's first digit and the multiplicand's first digit by the multiplier's second digit in the second step, then combine the results. The result is written as the second digit of the result.

The third step is to multiply the multiplicand's second digit with the multiplier's second digit, then write the product as the third digit in the result. All these steps are mentioned on the next page.

11	First step: 1*1 = 1
X 12	Second step: 1*2 + 1*1 = 3
----------	Third step: 1*2 = 2
132	

Urdhva Tiryagbhyam sutra for *N*-bit multiplication:

The multiplier and multiplicand of the input *N*-bit multiplier are divided into two *N*/2 bit values. Assigning inputs to the four multipliers is the first step. The inputs for multiplier 1 will be A [*N*/2: 1:0] and B [*N*/2: 1:0]. The values for multiplier 2 will be A [*N*-1: *N*/2] and B [*N*/2: 1:0], respectively. The inputs for multiplier 3 will be A [*N*/2: 1:0] and B [*N*-1: *N*/2]. The inputs for multiplier 4 will be A [*N*-1: *N*/2] and B [*N*-1: *N*/2].

To add the partial products from the multipliers different adders are used in literature [1-3]. They are discussed below.

RIPPLE CARRY ADDER

In a multiplier, a larger number of full adders are organized in a way that gives the results of an n-bit binary sequence addition operation (partial products). The carry input to the full adder stage is derived from the previous carry output of the adder, and it is repeated till the finish. The delay of adder is $m*T_{n\ bit\ adder}$, where *m* and *n* are the number of bits in the multiplier and multiplicand, respectively. Figure 1 illustrates a 4-bit RCA.

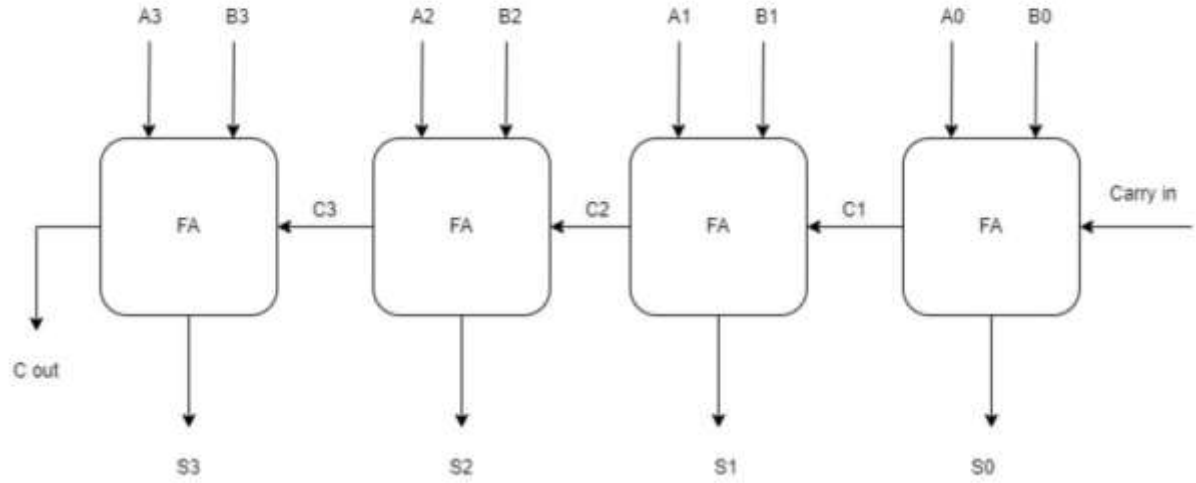

Fig.1. 4 bit RCA

CARRY SAVE ADDER:

A carry-save adder is one the digital adders that allows us to add three inputs at once. This method divides the addition of 3 digits into 2 steps. First compute sum bits and then carry bits and later they both are added to get the final result.
For an example:

X,Y,Z are the inputs and *Sum* is the output of CSA.

X : 1 0 1

Y : 0 1 0

Z : 0 1 1

S: 1 0 0

C: 1 1

Sum: 1 0 1 0

A CSA simply is a full adder with the C_{in} to Z, the S output named as *Sum*, and the C_{out} is just that C_{out} only. Three N bit numbers, X, Y and Z are divided into two numbers, C and S, in CSA. It is worth noting that carry bit C_0 is not generated. Here the delay of circuit is given by $T_{FA} + T_{n\text{-}bit\ RCA}$. Fig. 2 shows 4 bit CSA.

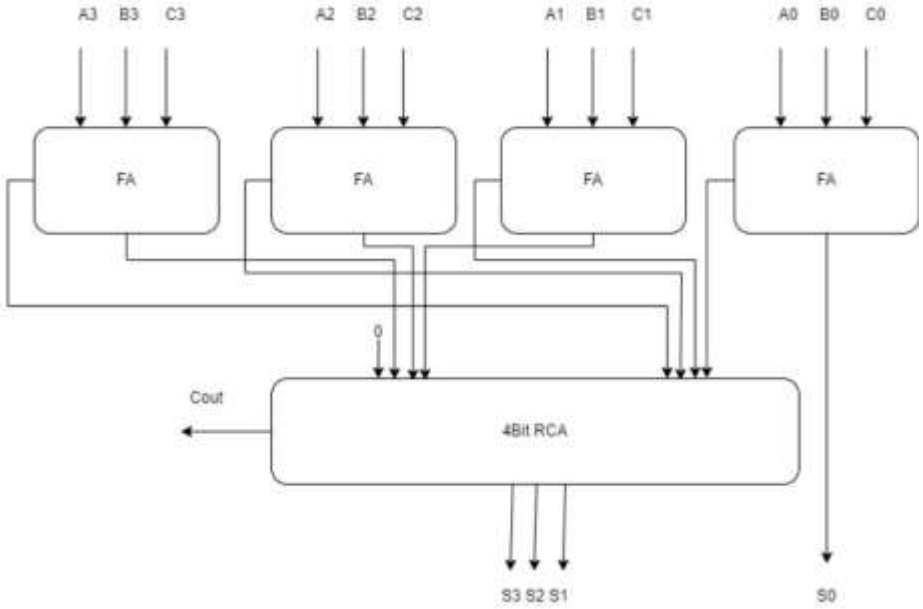

Fig.2. 4 bit CSA

III. ARCHITECTURE AND DISADVANTAGES OF EXISTING MULTIPLIERS

The Fig.3, Fig.4, and Fig.5, respectively, represent the existing Vedic multiplier architectures [1-3].In [1-2], the inputs are a[31:0] and b[31:0]. Whereas in [3] inputs are a[3:0] and b[3:0]. The inputs are divided into equal number of bits and according to Urdhva Tiryagbhyam sutra and multiplication is done.

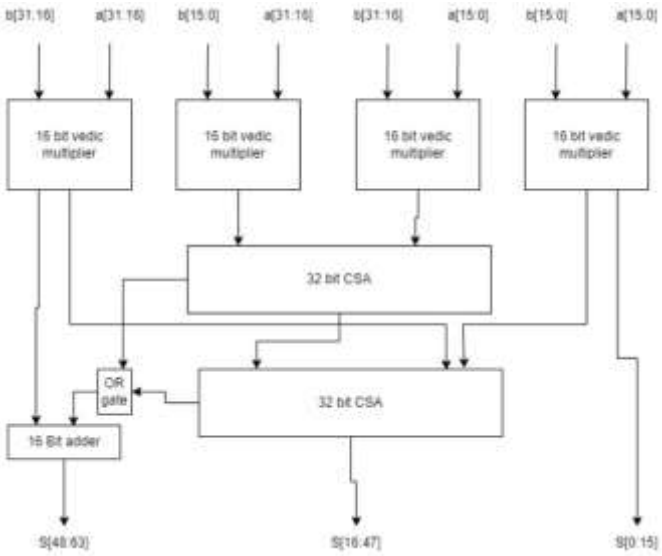

Fig . 3 . Vedic multiplier [1]

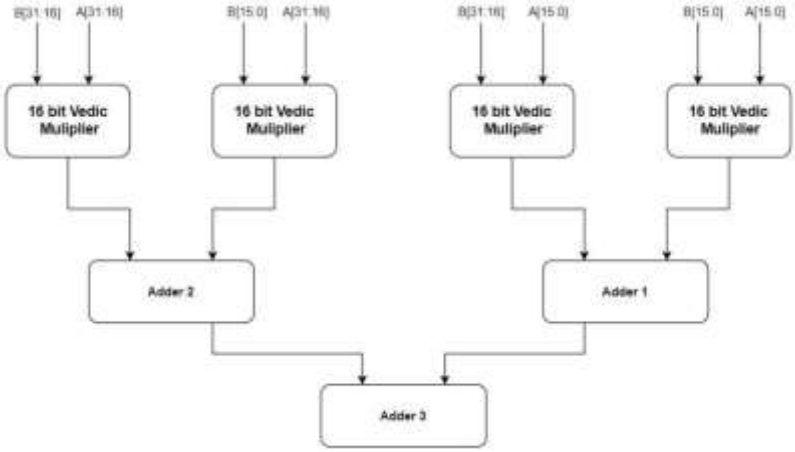

Fig. 4. Vedic multiplier [2]

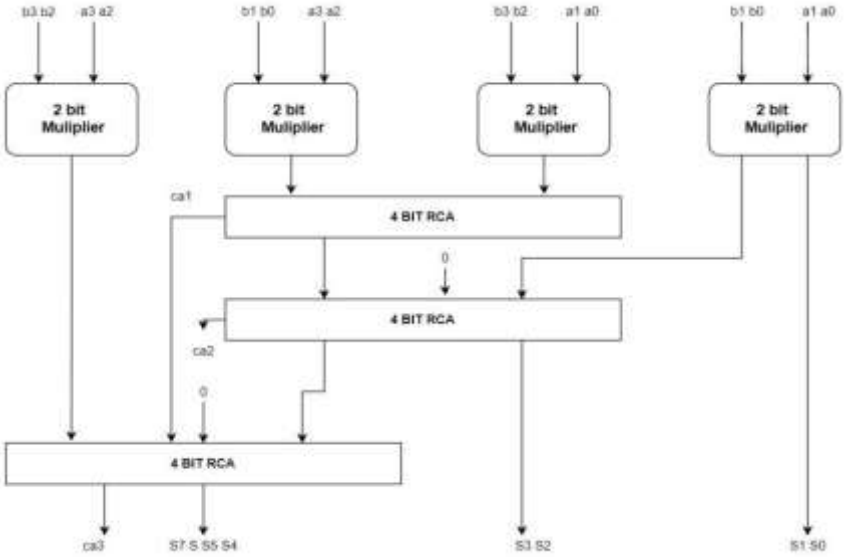

Fig. 5. Vedic multiplier [3]

It can be observed from the existing multipliers (Fig.3 - Fig.5) that the partial products are added using CSAs and Ripple Carry Adders. In [1] two CSA's are used for adding partial products. In [2] different sizes of adders are used for adding partial products. In [3] three RCA's are used to generate the final result. From the existing designs it can be observed that there are many Ripple carry adders and CSAs are used, so they consume a larger part of the FPGA and provide more delay while implementation. Therefore there is a need to reduce hardware, i.e., the number of adders to add the partial products. This paper presents a Vedic multiplier design that utilizes less hardware and provides less delay.

IV. Proposed design

The proposed Vedic multiplier also uses Urdhva Tiryagbhyam sutra for multiplication. In order to reduce the delay and area, the proposed design of multiplier uses one CSA in place of different parallel adders to add the partial products. The proposed design of Vedic multiplication technique is illustrated below for 4-bit inputs *(N* = 4),

$M = M_3 M_2 M_1 M_0$ and $N = N_3 N_2 N_1 N_0$, are 4 bit inputs and $Z_7 Z_6 Z_5 Z_4 Z_3 Z_2 Z_1 Z_0$ is 8-bit output.

$$M_3 M_2 M_1 M_0 \times N_3 N_2 N_1 N_0$$

$$(M_3 M_2) \times (N_1 N_0) \quad (M_1 M_0) \times (N_1 N_0)$$
$$(M_3 M_2) \times (N_3 N_2) \quad (M_1 M_0) \times (N_3 N_2)$$

$$Z_7 Z_6 \qquad Z_5 Z_4 Z_3 Z_2 \qquad Z_1 Z_0$$

The four *N*/2-bit multipliers are used in the intermediate stage calculation and we use four 2 bit multipliers.

$(M_3 M_2) \times (N_3 N_2)$ using 2-bit multiplier give result: $Z_{33} Z_{32} Z_{31} Z_{30}$.
$(M_3 M_2) \times (N_1 N_0)$ using 2-bit multiplier give result: $Z_{23} Z_{22} Z_{21} Z_{20}$.
$(M_1 M_0) \times (N_3 N_2)$ using 2-bit multiplier give result: $Z_{13} Z_{12} Z_{11} Z_{10}$.
$(M_1 M_0) \times (N_1 N_0)$ using 2-bit multiplier give result: $Z_{03} Z_{02} Z_{01} Z_{00}$.
The design of the proposed *N*-bit Vedic multiplier is shown in Fig.6.

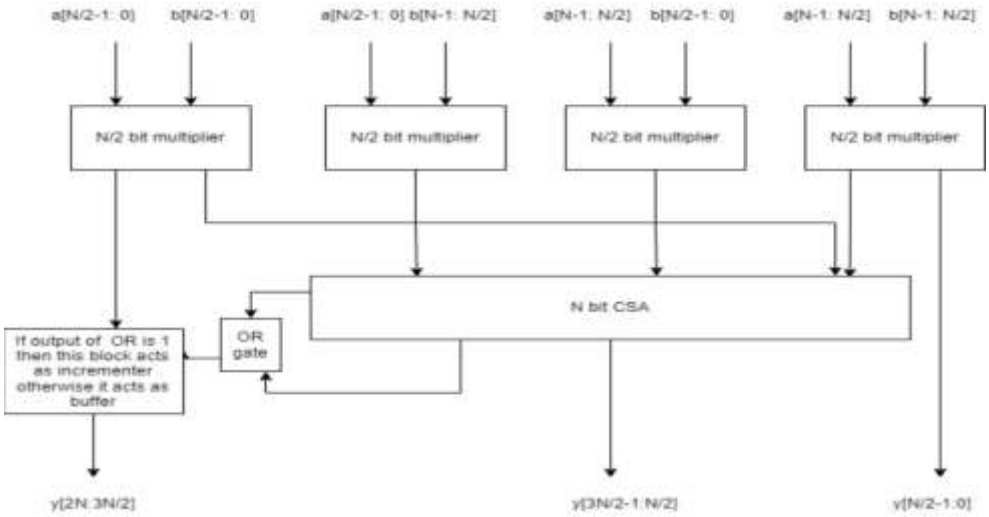

Fig.6. Proposed design of *N*-bit Vedic multiplier

Three *N*-bit input data (here *N*=4) $Z_{23} Z_{22} Z_{21} Z_{20}$, $Z_{13} Z_{12} Z_{11} Z_{10}$ and $Z_{31} Z_{30} Z_{03} Z_{02}$ which are the outputs of four *N*/2 bit multipliers are added with *N*-bit CSA.
Calculation of final product:

- The first *N*/2 bits ($Z_{01} Z_{00}$) from first multiplier directly goes to final stage.

- The *N*/2 LSBs of last multiplier and *N*/2 MSBs of first multiplier as given as one input (Z_{31} Z_{30} Z_{03} Z_{02}) and other two multipliers outputs are given as other two inputs for CSA and the output of CSA gives middle *N* bits of final stage.
- OR operation is performed C_{out} and MSB of CSA output.
- If OR operation is true then the bits i.e., Z_{33} Z_{32} are incremented by 1 else they are directly given to final stage.
- We generate the leftmost *N*/2 bits from the leftmost *N*/2 bit multiplier and middle *N* bits from CSA, as well as the rightmost *N*/2 bits from the rightmost *N*/2 bit multiplier, at the final stage.

V. Results

For comparison purposes, the proposed multiplier and multipliers [1-3] are simulated using the Xilinx Vivado tool. Tables 1 and 2 show the simulation results for 8-bit, 16-bit, and 32-bit multipliers, respectively. Figure 7 shows the output waveforms of a 32-bit multiplier.

.

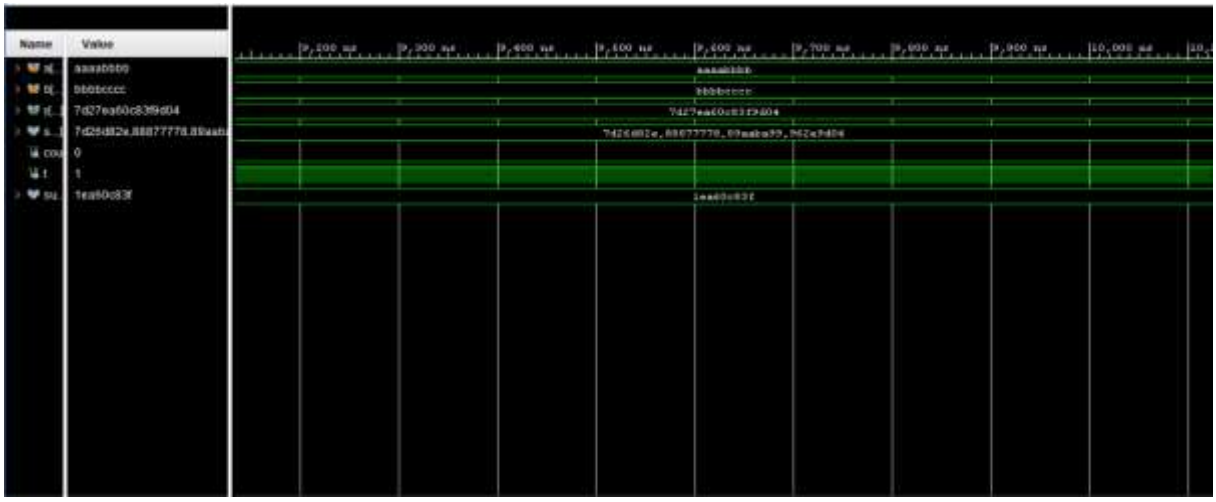

Fig.7. Simulation result of 32-bit Vedic multiplier

As shown in Fig.7, for a 32-bit multiplier the inputs are *A* (*aaaabbbb*) and *B* (*bbbbcccc*) and the output is *Y* (*7d27ea6083f9d04*). The comparison of delay and number of LUTs are shown in table.1 and 2, respectively.

Table 1. Delay comparison

Types of multiplier	8 bit	16 bit	32bit
[1]	24.186ns	43.714ns	68.859ns
[2]	15.2ns	23ns	54.037ns
[3]	28.27ns	56.4ns	110.8ns
Proposed design	15.450ns	22.917ns	39.10ns

Table 2. LUTs comparison

Types of multiplier	8 bit	16 bit	32bit
[1]	201	903	3802
[2]	124	462	1758
[3]	110	450	1722
Proposed design	92	418	1609

The proposed 32-bit multiplier has a 39.10 nanosecond delay and 1609 LUTs. When comparing the proposed design to existing multipliers, it is obvious that the delay and LUTs of the proposed design are greatly reduced.

IV. CONCLUSION

The Urdhva Tiryagbhyam sutra is used for implementation of a Vedic multiplier in this study. As a result, the delay and area are minimised by using the multiplier with the proposed design. When compared to existing multipliers in the literature, the number of LUTs for the proposed 32-bit multiplier is reduced by 57.68%, 08.4%, and 6.5%, respectively. The proposed 32-bit multiplier reduces the delay by 43.21%, 27.64%, and 64.71% respectively when compared to existing multipliers. The proposed multiplier can be employed in microprocessors and microcontrollers because its delay and size are minimized. As a result, the proposed multiplier proved to be more efficient in terms of speed and number of LUTs.

REFERENCE

[1] M. Bala Murugesh, S.Nagaraj, Z J.Jayasree, G.Vijay Kumar Reddy, “Modified High Speed 32-bit Vedic Multiplie Design and Implementation”, Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020),Nagpur.

[2] Akshay Savji, Shruti Oza, “Design and Implementation of Vedic Multiplier”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-6, March 2020.

[3] Poornima M, Shivaraj Kumar Patil, Shivukumar , Shridhar K P , Sanjay H , “Implementation of Multiplier using Vedic Algorithm” , International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2 Issue-6, May2013.

[4] https://en.wikipedia.org/wiki/Carry-save_adder