Implementation of 16-Bit Vedic Multiplier Using Modified CSA

B. Vanitha(✉), S. Nagaraj, and B. Sai Kumar

Department of ECE, SVCET (Autonomous), Chittoor, AP 517127, India
vanithav611@gmail.com

Abstract. In this paper we have designed and implemented vedic multiplier using RCA, CSA and MCSA. The main component of DSP is the multiplier. The demand for high-speed multiplier circuitry is enormous. One of the most crucial factors in determining a multipliers effectiveness is speed, power and area. In order to accelerate multiplication using carry save adder, this paper aims based on the urdhva tiryagbhyam algorithm a Vedic multiplier is developed using CSA. One of the fast adder that can be utilised to lessen the total delay associated with addition is CSA. However, because of the dual RCA construction, carry save adder is not an area-efficient one. Using the carry save adder, RCA, half adders, full adder in Verilog HDL, a 16-bit Vedic multiplier is created using Modelsim to simulates and synthesised using Xilinx ISE 14.7. In this project implementation of Vedic Multiplier using carry save adder and comparing it with the Vedic multiplier using Ripple Carry Adder is performed. The synthesis result shows that CSA has 16% greater delay than Vedic multiplier using MCSA. CSA has 44% greater area than MCSA. MCSA has 15% reduced power than vedic multiplier using CSA.

Keywords: Multiplier · Vedic Multiplier · Carry save adder · Modified carry save adder

1 Introduction

The use of electronic devices has skyrocketed in todays still evolving technology area. The size of technological products used by the general public needs to be drastically decreased. Only when the interior circuitry of an electronic device is optimised can the dimensions of the device be decreased. The majority of an integrated circuit (IC) are adders and multipliers. By comparison, it will increase the area of an IC. If we can reduce the adder area, then we can reduce multiplier area. People are currently reluctant to use slow-performing technology, so it is imperative that we create electronic equipment that works more quickly and with less lag. We require equipment with optimised internal circuitry that uses less power.

B. Raj et al. (Eds.): ICETE 2023, AER 223, pp. 919–927, 2023.
https://doi.org/10.2991/978-94-6463-252-1_92

2 Multiplier

A multiplier is a type of combinational circuit that is used in DC to execute multiplication operations. In the area of DSP, the multiplier has a wider range of applications. Compared to addition and subtraction, the multiplication process is more difficult. A multiplier architecture, which houses the adder circuitry, will carry out the multiplication function.

3 Vedic Multiplier

The Vedic multiplier is a multiplication algorithm commonly used in digital circuits that is known for its speed and efficiency. This algorithm is based on the principles of Vedic mathematics, an ancient Indian mathematical system that has been adapted for modern computing applications. The Vedic multiplier works by breaking down the numbers being multiplied into smaller components, and then using specific techniques to perform the multiplication. These techniques are based on simple rules and patterns, making them easy to implement in digital circuits.

One of the key advantages of the Vedic multiplier is its speed. It can perform multiplication in a single clock cycle, which makes it faster than traditional multiplication algorithms. This makes it particularly useful in applications where high-speed multiplication is required, such as in digital signal processing, image processing, and cryptography.

Another advantage of the Vedic multiplier is its scalability. It can be easily extended to handle larger numbers by breaking them down into smaller components and applying the same techniques repeatedly. This makes it a comfortable algorithm is used in a wider applications.

Overall, the Vedic multiplier is a powerful and efficient algorithm for multiplication in digital circuits, offering speed and scalability advantages over traditional multiplication algorithms.

Vedic mathematics is a system of mathematical principles and techniques originating in ancient India. It is based on sixteen sutras, which are concise and easy-to-remember formulas for solving mathematical problems. Here are the sutras:

The Vedic mathematics principles are based on ancient Indian mathematical practices and include a number of algorithms for computation. Some of these algorithms include: Ekadhikena Purvena which involves adding one more than the previous numbers.

Nikhilam Navatashcaramam Dashatah which involves subtracting all numbers from 9 except for the last which is subtracted from 10.

Urdhva-Tiryagbyham which involves multiplying vertically and crosswise Paravartya Yojayet which involves transposing and adjusting.
Shunyam Saamyasamuccaye which involves setting a sum to zero when its equal. Anurupyena which involves proportional multiplication.
Sankalana-Vyavakalanabhyam which involves addition and subtraction. Puranapuranabhyam which involves completion and non completion.
Chalana-Kalanabyham which involves finding differences and similarities. Yaavadunam which involves finding the extent of a deficiency.
Vyastisamanstih which involves finding the relationship between parts and wholes. Shesanyankena Charamena which involves finding the remainders by the last digit.

Sopaantyadvayamantyam which involves finding the ultimate and two times the furthermost.
Ekanyunena Purvena which involves subtracting one less than the before number.
Gunitasamuchyah which involves finding the product of the sum is equal to the products sum.
Gunakasamuchyah which involves finding the factors of the sum is similar to the sum of the factors.
These sutras are versatile and can be used in various mathematical operations. They help to simplify and speed up calculations, making them particularly useful in competitive exams, engineering, and finance.

In this work, 16-bit Vedic multiplier is simulated by Urdhva Tiryagbhyam algorithm.

4 Vedic Multiplier Architecture

Vedic multiplier is a type of multiplier architecture in VLSI design that is based on the ancient Indian Vedic mathematics principles. It is efficient and high speed multiplier architecture that can perform multiplication of two n-bit numbers in just log2(n) stages, compared to traditional multiplication methods that require n stages. The Vedic multiplier architecture consists of several stages, including the following: Pre-processing stage: This stage involves preparing the inputs for multiplication by using techniques such as sign extension, zero-padding, and normalization. Parallel multiplication stage: In this stage, the multiplicand and the multiplier are broken down into smaller sub-multiplicands and sub-multipliers, which are then multiplied in parallel. Carry save addition stage: The previous days partial products obtained from there are added using carry save addition technique, which involves generating two sums and a carry signal. Final addition stage: In this stage, the final product is obtained by adding the results from the previous stage and propagating any carry bits. The Vedic multiplier architecture has several advantages over traditional multiplier architectures, including higher speed, lower power consumption, and smaller area requirements. It is often used in applications that require high speed multiplication, such as image processing, digital signal processing, and cryptography (Figs. 1 and 2).

A. Vedic Multiplier 2-bit

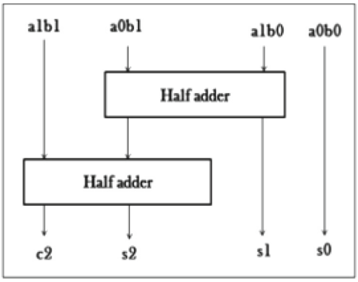


Fig. 1. 2-bit Vedic multiplier.

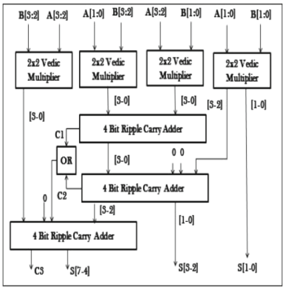


Fig. 2. 4-bit Vedic multiplier.

The given figure illustrates 2 binary numbers A and B, each having two bits denoted as A = a0a1 & B = b0b1. To obtain the LSB of final result .0 of the two numbers are multiplied vertically. In the second stage, the next LSB of A is multiplied by the next higher bit of B crosswise, and the resulting partial product is added to the carry produced by multiplying the MSB of the two numbers. This addition yields the second bit and a carry that are added to the sum of the previous step to obtain the final products third and fourth bits. The below diagram shows the final result of 2 × 2 architecture.

B. Vedic Multiplier 4-bit

This section discusses the 4-bit VM, which operates on two 4-bit numbers, A and B. The multiplier produces an eight-bit result, denoted as C3S6S5S4S3S2S1S0. The Vedic multiplier leverages parallel computation of partial products to reduce delay resulting from increased bit count. Specifically, the 4-bit Vedic multiplier is implemented with 2-bit Vedic multipliers that produce partial outputs. The partial products are then added by three RCA's, each carrying four bits. The carry output from the first 2 RCA is combined using an OR operation and combined as input to the next Ripple Carry Adder. Some of the RCA receive zero inputs, as illustrated in the block diagrams shown in Fig. 3. This approach speeds up computation and minimizes delay in the multiplier.

C. Vedic Multiplier 8-bit

Let us assume two 8 bit numbers are A and B. The final output can be obtained as the 16 bits. From the below diagram implementation of two 8 bit vedic multiplier is understood.

D. 6 × 16 Vedic Multiplier

Figure 4 depicts the architecture of the 16 × 16 block, which is an optimised arrangement of 8 × 8 blocks. The 8 bits (bytes) are grouped for each 16-bit input will be in the first stage in the design of the 16 × 16 block. The vertical and crosswise product terms will be formed by the LSB of two sources. To create sixteen partial product rows, a separate 8 × 8 Vedic multiplier processes each incoming byte separately. To produce final product bits, these products rows are ideally added in a 16-bit RCA. The 8 × 8 Vedic multiplier is used to create the schematic of a 16 × 16 block. The words for the Urdhva cross and vertical products are represented by the partial products. The final product is then produced by using or gate.

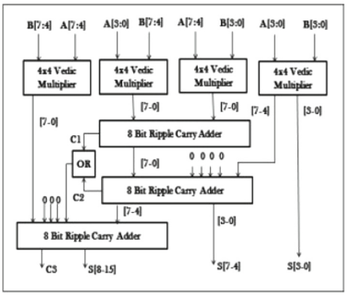


Fig. 3. 8-bit Vedic multiplier.

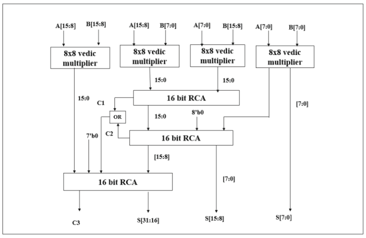


Fig. 4. 16-bit Vedic multiplier.

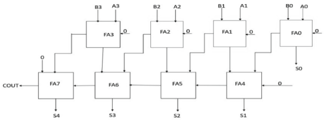


Fig. 5. 4-bit CARRY SAVE ADDER.

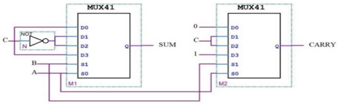


Fig. 6. One-bit full adder used in the structure of Modified CSA.

5 Carry Save Adder

A CSA is a digital circuit used for high-speed addition of multiple binary numbers. It works by first saving the individual bits of each number and then combining them in a parallel fashion to produce the final result. This technique reduces the number of carry operations required, increasing speed and efficiency (Fig. 5).

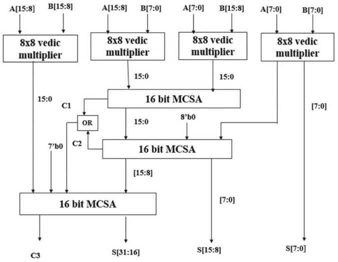


Fig. 7. 8-bit Vedic multiplier

6 Modified Carry Save Adder

A MCSA with MUX is high-speed digital circuit used for addition of multiple binary numbers. It uses multiplexers to reduce the number of partial sum bits generated, reducing circuit complexity and power consumption. The MUX-based MCSA is more efficient than the traditional MCSA, making it a popular choice in modern computing systems (Fig. 6).

7 Proposed Method

Complexity for design gets decreased for larger no of bits and modularity gets increased. By Hardware Description Language the proposed Vedic multiplier is coded, simulated by modelsim and synthesized using Xilinx ISE 14.7. To show the significant improvement in efficiency in terms of speed, Finally the results are compared with Conditional multipliers. To design digital systems multiplier is main block. To implement fast multipliers many algorithms are reported. In order to implement ancient multiplier vedic multiplication is another option. This discuses the vedic multiplier. Urdhva Triyagbhyam is the main Vedic multiplication algorithm. It can be said as universal multiplication formula that can be used in any situation involving multiplication, Vertically and crosswise are the literal translations. With the Vedic multiplier, two operands are multiplied by multiplying vertically and crosswise, and the results are then added. In the place of RCA we have replaced CSA and MCSA to analyse power and delay. A modified carry save adder (MCSA) with MUX is a high-speed digital circuit used for addition of multiple binary numbers. It uses multiplexers to reduce the number of partial sum bits generated, reducing circuit complexity and power consumption. The MUX-based MCSA is more efficient than the traditional MCSA, making it a popular choice in modern computing systems. It is used to reduce the area and delay. Area and delay are designed for efficient multiplier to obtain better performance of the multipliers (Figs. 7 and 8).

8 Methodology

Using the half adder, RCA, a 16 × 16 Vedic multiplier is created using a in Verilog HDL. Verilog HDL is used to create a Ripple carry adder based 16-bit vedic multiplier using adders like half adder and full adder. Design is Simulated by Modelsim and Synthesized

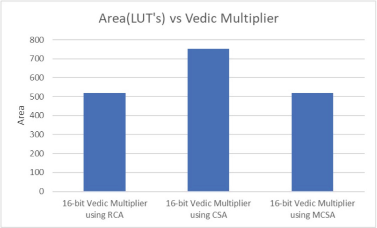


Fig. 8. Comparative analysis of Area for the implemeneted vedic multiplier

with Xilinx ISE 14.7. At the end vedic multiplier Which is designed using both carry save adder and Ripple carry adder are compared for Delay, power and area (number of LUT's) parameters. The carry save adder is not efficient one. Modified Carry Save Adder (MCSA) is designed to reduce the Delay and power than carry save adder (Figs. 9, 10, 11, and 12).

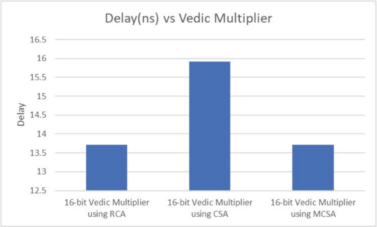


Fig. 9. Comparative analysis of Delay for the implemeneted vedic multiplier.

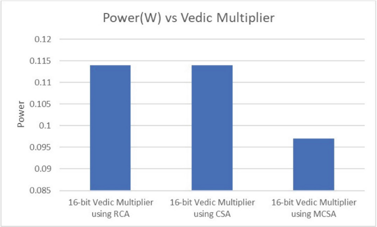
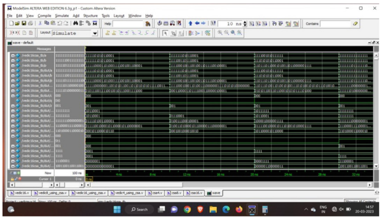


Fig. 10. Comparative analysis of Power for the implemeneted vedic multiplier.



```
Messages
/vedic16csa_tb/a          1111111111111111
/vedic16csa_tb/b          1111110011111111
/vedic16csa_tb/p          1111101111111110
/vedic16csa_tb/dut/a      1111111111111111
/vedic16csa_tb/dut/b      1111110011111111
/vedic16csa_tb/dut/p      1111101111111110
/vedic16csa_tb/dut...     1111101100000100
/vedic16csa_tb/dut...     1111101000000011
/vedic16csa_tb/dut/c      000
/vedic16csa_tb/dut/q      St0
/vedic16csa_tb/dut/z      001
/vedic16csa_tb/dut/...    11111111
/vedic16csa_tb/dut/...    11111111
/vedic16csa_tb/dut/...    1111111000000001
/vedic16csa_tb/dut/...    1110000111100001
/vedic16csa_tb/dut/...    1101000011000010
/vedic16csa_tb/dut/...    000
/vedic16csa_tb/dut/...    St1
/vedic16csa_tb/dut/...    001
/vedic16csa_tb/dut/...    1111
/vedic16csa_tb/dut/...    1111
/vedic16csa_tb/dut/...    11100001
/vedic16csa_tb/dut/...    1001100110011001
Now       100 ns
Cursor 1  0 ns
Project : vedicmux16   Now: 100 ns  Delta: 0
```

Fig. 11. 8-bit Vedic multiplier.

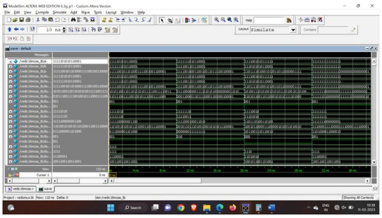


Fig. 12. 16-bit Vedic multiplier.

Table 1. COMPARATIVE ANALYSIS OF VEDIC MULTIPLIERS IN TERMS OF DELAY AND AREA

S.NO	Different Vedic Multipliers	LUT'S	Delay (in ns)	Power (in mW)
1	Vedic Multiplier using RCA	520	13.71	0.114
2	Vedic Multiplier using CSA	751	15.913	0.114
3	Vedic Multiplier using MCSA	520	13.713	0.114

9 Simulation Results and Discussion

A 16 × 16 Vedic multiplier using is CSA designed using Verilog HDL with the adders such as full adder, RCA, and modified CSA. A 16-bit vedic multiplier using RCA is designed using Verilog HDL with the adder such as Half adder, Full adder. The Design is Synthesized using Xilinx ISE 14.7 and Simulated using Modelsim. At the end vedic multiplier Which is designed using CSA, modified CSA and RCA are compared for Delay, power and area (number of LUT's) parameters. Modified CSA is designed to reduce the Delay and power than CSA (Table 1).

10 Conclusion

In this work, RCA, CSA and modified CSA based 16 bit vedic multiplier circuits were presented and analysed. The Vedic Multiplier is implemented by using CSA, and its performance is compared with the Vedic Multiplier implemented using RCA. In this project an implementation of Vedic Multiplier using CSA and it is compared with MCSA. The synthesis results shows that CSA has 16% greater delay than Vedic multiplier using MCSA. CSA has 44% greater area than vedic multiplier using MCSA. MCSA has 15% reduced power than vedic multiplier using CSA.