International Journal of Advanced Technology in Engineering and Science
Vol. No. 09, Issue No. 07, July 2021
www.ijates.com
ijates
ISSN 2348 - 7550

VEDIC MULTIPLIER WITH HIGH SPEED AND SMALLER AREA USING HYBRID ADDER

Jayamangala Sristi

Assistant Professor, Santhiram Engineering College, Nandyal

Email id: jayamangala.1987@gmail.com

ABSTRACT

The Vedic multiplier is based on the ancient multiplication algorithms (sutras) used in INDIA. This paper is based on the "URDHVA TIRYAGBHYAM SUTRA," which is one of the sutras. These sutras are built to help you measure faster in your brain. Here is a method for reducing power by modifying the design of the Vedic multiplier using some existing methods. The circuits' functionality was checked, and performance parameters such as propagation delay and area were reduced. The 16 x 16 Vedic multiplier is written in Verilog HDL and synthesized with the VIVADO Method from Xilinx. The efficiency is compared to the current Vedic multiplier architecture in terms of area and delay.

Keywords: ***Vedic multiplier, Carry select Adder, Verilog HDL, Xilinx VIVADO tool***

1 Introduction

Multiplication is commonly used in optical signal processing applications, necessitating the use of a high-speed multiplier. Centered on Vedic mathematics, this paper introduces a systematic design methodology for a quick and area efficient multiplier. The Multiplier Architecture is based on the ancient Indian Vedic Mathematics Vertical and Crosswise algorithm. The key cause of delay in the overall multiplication process is the adder block used in Vedic multipliers.

Because of the unavoidable carry propagation chain, the carry-propagation adder (CPA) is frequently part of the critical delay path, limiting overall device efficiency. In a Wallace tree multiplier, for example, the delay of a quick CPA converting the final carry-saved number to its two's complement type is usually 25% to 35% of the total multiplier delay depth. In today's deep sub-micron regime, the value of wire capacitance to gate delay and power consumption is especially pronounced. As a result, it's critical to combine logic structure with circuit technique to reduce CSL's transistor count even further, reducing wire length and simplifying the layout.

1.1 Vedic algorithm-based multiplication

The generalized structure of Vedic multiplier is proposed. The basic module is 2x2 Multiplier. For N x N 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, if A=0000001010101101, then it is divided as 00000010, and 10101101. Similarly if B=

0010001110101011, then it can be divided as 00100011 and 10101011. Representing the divided parts of A as AM and AL. Similarly for the input B, it is divided in two parts as BM and BL. A and B can be represented as AMAL and BMBL. Multiplication operation between A and B can be represented as:

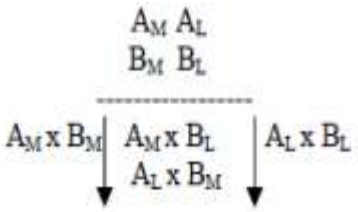


Figure 1: Cross Multiplication Representations

1.2. Multiplication of two decimal numbers by Vedic Mathematics

To begin, let us take two three digit numbers 386 and 512. The product of the two decimal numbers using the Vedic multiplication technique is illustrated in *Fig1*. The multiplication begins by first multiplying the two digits connected through the line. Then, the first digit of the product is stored as the first digit of result and the remaining digit is the carry for the next stage. In the next stage of multiplication again the digits connected to each other through the line are multiplied and the individual products obtained are added to the carry from the previous stage.

Then, again the first digit of the sum obtained is stored as the second digit of the result and the remaining digit is the carry for the next stage. [2]-[4]. This process is repeated again and again and thus the final result i.e. the product of the two numbers 386 x 512 is obtained through the Vedic multiplication technique as shown in below The logic equations for binary to excess-1 converter are

N0 =~ M0

N1 = M0 ^ M1

N2 = M2 ^ (M0 &M1)

N3 = M3 ^ (M0 &M1 &M2)

1.3. 16 X 16 Vedic Multiplier using Modified Carry Select Adder

It describes the 16 X 16 Vedic Multiplier using Modified Carry Adder.The block diagram is shown below. In this multiplier four 8bit vedic multipliers and three 16bit modified carry select adders have been used.

Consider the process of multiplication by taking two numbers X & Y each of 16bits.The two numbers X[15:0] and Y[15:0] are multiplied as per the diagram as shown in the below figure and the final product is obtained. The carry generated by each stage is fed into next stage.

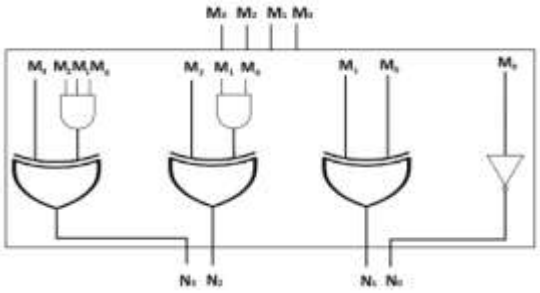
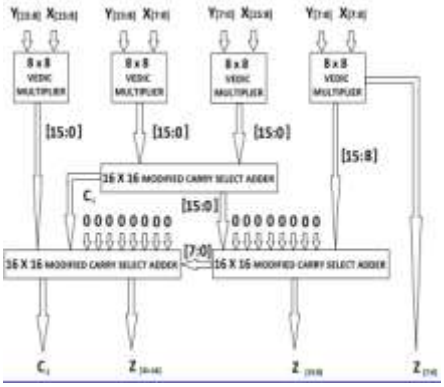


Figure 2: Urdhva Tiryakbhyam Sutrm

adder is used in the intermediate computation, hence fast and area efficient design will enhance the overall performance of multiplication.

2. Existing Method

The modified carry select adder is used as alternative of the ripple carry adder because it incorporates a binary to excess-1 converter as shown in the figure, to reduce the time delay and area occupied in contrastive the ripple carry adder. The performance of the modified carry select adder is better as compared ripple carry adder.



3. Proposed Method

URDHVA TIRYAKBHYAM is the most effective technique in terms of pace among the sixteen sutras in Vedic multiplication techniques. The aim of this paper is to create a hybrid adder multiplier that implements the "URDHVA TIRYAKBHYAM" sutra with improved speed of operation. Adders are commonly used in the critical path of several microprocessor chip blocks. They are used not only in arithmetic logic units (ALUs), but also in other parts of the processor to measure addresses, table indices, and other functions.The success of these adders has a major effect on the efficiency of the digital system. In this paper, we propose a hybrid prefix adder

that improves delay and area used efficiency. Computation time and area are the most critical parameters to consider when testing adder designs.

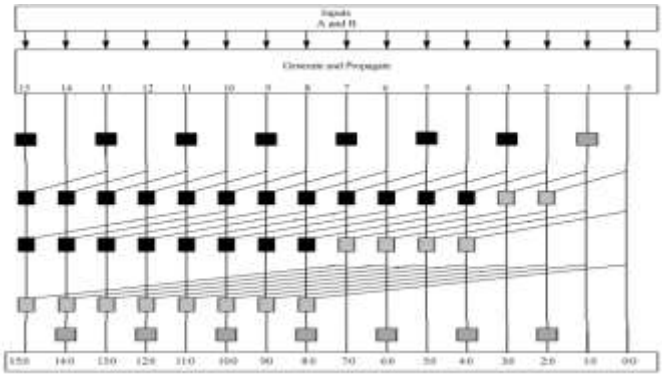
Hybrid-prefix structures are found to be common in high performance adders because of the delay is logarithmically proportional to the adder width. Hybrid-prefix structures basically consists of 3 stages

- Pre computation
- Prefix stage
- Final computation

A. Pre computation in pre computation stage, propagates and generates are computed for the given inputs.

B. Prefix stage in the prefix stage, group generate/propagate signals are computed at eachbit. The black cell (BC) generates the ordered pair, the gray cell (GC) generates only left signal.

C. Final computation in the final computation, the sum and carryout are the final output.



A. Pre computation in pre computation stage, propagates and generates are computed for the given inputs.

B. Prefix stage in the prefix stage, group generate/propagate signals are computed at each bit. The black cell (BC) generates the ordered pair, the gray cell (GC) generates only left signal.

C. Final computation in the final computation, the sum and carryout are the final output.

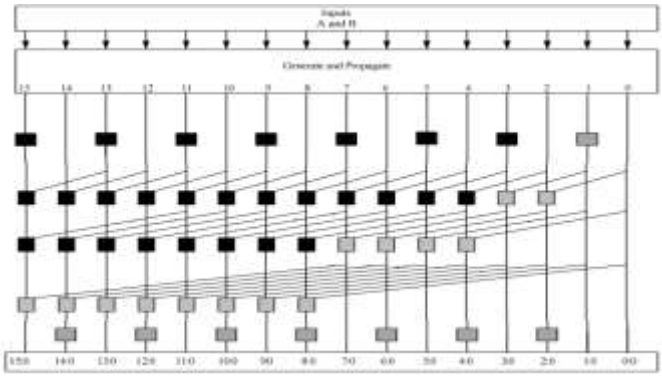


Fig: Hybrid prefix adder

3.1. Hybrid Adder

Parallel prefix adders are used to speed up the binary additions as they are very flexible. The structure of Carry Look Ahead Adder (CLA) is used to obtain parallel prefix adders. Tree structures are used to increase the speed of arithmetic operation. Parallel prefix adders are used for high performance arithmetic circuits in industries as they increase the speed of operation. The construction of parallel prefix adder involves three stages:

1. Pre- processing stage
2. Carry generation network
3. Post processing stage

Pre-processing stage

Generate and propagate signals to each pair of inputs A and B are computed in this stage. These signals are given by the following equations:

Pi=Ai xor Bi (1)

Gi=Ai and Bi (2)

Carry generation network

In this stage, we compute carries equivalent to each bit. Implementation of these operations is carried out in parallel. After the computation of carries in parallel they are segmented into smaller pieces. Carry propagate and generate are used as intermediate signals which are given by the logic equations 3& 4:

$CP_{ij}=P_{i:k+1}$ and $P_{k:j}$ (3)

$CG_{ij}=G_{i:k+1}$ or ($P_{i:k+1}$ and $G_{k:j}$) (4)

The operations involved in fig. 1 are given as:

$CP0=P_i$ and P_j (3(i))

$CG0=(P_i$ and $G_j)$ or G_i (3(ii))

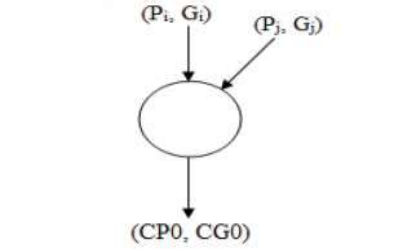


Fig : Carry Generation Network

Post processing Stage

This is the concluding step to compute the summation of input bits. It is common for all the adders and the sum bits are computed by logic equation 5& 6:

Ci-1= (Pi and Cin) or Gi (4)

Si=Pi xor Ci-1 (5)

It describes the 16 X 16 Vedic Multiplier using Hybrid Adder. The block diagram is shown below. In this multiplier four 8bit vedic multipliers and three 16bit modified carry select adders have been used.

3.2. Block Diagram:

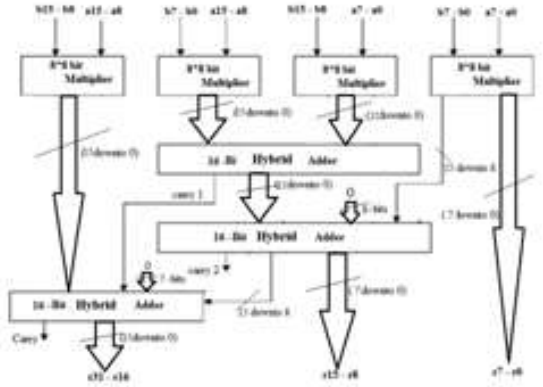


Figure: 16 X 16 Vedic Multiplier using Hybrid Adder

4.Results of Existing Method

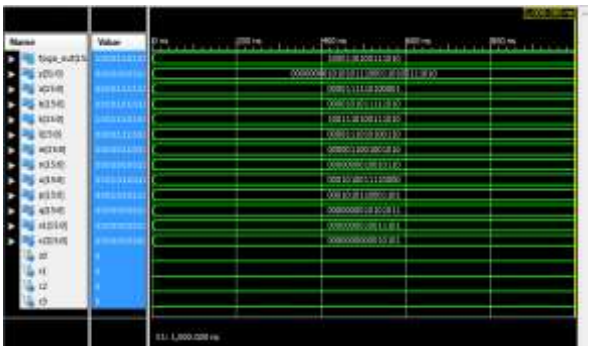
4.1. Area of Modified Carry Select Adder

```
Device utilization summary:
---------------------------

Selected Device : 3s100etq144-4

 Number of Slices:                      355  out of    960    36%
 Number of 4 input LUTs:                619  out of   1920    32%
 Number of IOs:                          64
 Number of bonded IOBs:                  64  out of    108    59%

---------------------------
Partition Resource Summary:
---------------------------
```

4.2. Delay of Modified Carry Select Adder

```
Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================

Cross Clock Domains Report:
--------------------------

=========================================================================


Total REAL time to Xst completion: 15.00 secs
Total CPU time to Xst completion: 15.10 secs
```

4.3. Output of Modified Carry Select Adder



5 Results of Proposed Method

5.1. Area of Hybrid Adder

```
Device utilization summary:
---------------------------

Selected Device : 3s100etq144-4

 Number of Slices:                      344  out of    960    35%
 Number of 4 input LUTs:                634  out of   1920    28%
 Number of IOs:                          69
 Number of bonded IOBs:                  69  out of    108    59%

-----------------------------
Partition Resource Summary:
-----------------------------
```

5.2. Delay of Hybrid Adder

```
Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================

Cross Clock Domains Report:
---------------------------

=========================================================================


Total REAL time to Xst completion: 10.00 secs
Total CPU time to Xst completion: 10.54 secs
```

6.Comparison Table

Vedic multiplier	Delay	Area
Modified carry select adder	15ns	32%
Hybrid adder	10ns	28%

7. Conclusion

As compared to the 16 x 16 Vedic multiplier using adjusted carry select adder, the time delay and area involved in the 16 x 16 Vedic multiplier using Hybrid adder is less. As a result, it has increased speed and can be used in DSP, DIP, and processors to perform faster computations.

REFERENCE'S

1) Kanhe, Aniruddha, Shishir Kumar Das, and Ankit Kumar Singh. "Design and implementation of low power multiplier using vedic multiplication technique." International Journal of Computer Science and Communication 3, no. 1, pp.131-132, 2012.Mehta, Parth, and DhanashriGawali.
2) "Conventional versus Vedic mathematical method for Hardware implementation of a multiplier." In IEEE International Conference on Advances in Computing, Control, & Telecommunication Technologies, 2009, pp. 640-642, 2009.
3) Kumar, Manoj, Sujata Pandey, and Sandeep K. Arya. "Design of CMOS Energy Efficient Single Bit Full Adders." In High Performance Architecture and Grid Computing, pp. 159-168. Springer Berlin Heidelberg, 2011.
4) Anjana, R., B. Abishna, M. S. Harshitha, E. Abhishek, V. Ravichandra, and M. S. Suma."Implementation of vedic multiplier using Kogge-stone adder." In IEEE International Conference on, Embedded Systems (ICES), 2014 pp. 28-31, 2014.