MAT JOURNALS

Journal of Electronic Design Engineering
Volume 3 Issue 3

Delay optimized 16 X 16 bit Vedic Multiplier

Sri vidya B.V [#1], ***Kirankumar.T*** [*2]

[#*] Faculty, Department of Telecommunication, Dayananda Sagar College of Engineering, Bangalore, India

[1] srividyabv@gmail.com

[2] kiran.thodmal@gmail.com

Abstract

In this paper a comparative study of multiplier is done for speed. The concept used is "UrdhvaTiryagbhyam" algorithm for ancient Indian Vedic mathematics which is utilized for multiplication to improve the speed and area. The approached architecture for a 16X16 multiplier uses 8X8 multiplier along with parallel Carry Save Adders. The carry save adder in the multiplier architecture increases the speed of addition of partial products. The 16×16 Vedic multiplier is coded in VHDL, synthesized and simulated using Xilinx ISE 10.1 software. This multiplier is implemented on Spartan 2 FPGA device XC2S100-5tq144 and also on Virtex-4 vlx15sf363-12. The performance of the proposed algorithm is evaluated based on its speed and device utilization, when implemented on FPGA. We observe, that the speed has enhanced in the proposed multiplier design as compared to the existing multipliers [3]and [4].

Keywords: Vedic Mathematics, Algorithm, Multiplication Multipliers, VHDL, FPGA

INTRODUCTION

Many digital signals processing operation requires several multiplication and for the same, we need very fast multiplier for a wide range of requirements for hardware and speed. Vedic mathematics was rediscovered in the early twentieth century from ancient Indian sculptures (Vedas).The Multiplier Architecture is based on the Vertical and Crosswise algorithm of ancient Indian Vedic Mathematics. Vedic mathematics simplifies and optimizes the conventional mathematical algorithm.

In [3], the authors have proposed a new multiplier based
on an Vedic algorithm for low power and high speed applications. Their multiplier architecture is based on generating 8bit partial products and their sums by using two16bit parallel adders. The n bit parallel adder have a propagation delay of approximately 2nD[5]. In their architecture the second 16 bit parallel adder waits for the sum to be generated from the first 16 bit parallel adder.

In [4] the authors have proposed architecture for squaring a number using parallel adders and also have compared the performance of Vedic multiplier and Booth algorithm. They have concluded that Vedic multiplier is faster than Booth algorithm.

In the proposed work, a new multiplier architecture is proposed using Carry Save adders for the "Urdhva Tiryagbhyam" Sutra. These carry save adders do not propagate carry. Hence there is an overall reduction in the delay.

An organization of this paper is as : section 2 describes the Vedic multiplier using "Urdhva Tiryagbhyam" sutra. Section 3 shows proposed architecture using carry save adders for 16 bit multipliers. Section 4 describes results, comparison with [3], [4] and finally

conclusion and references are given in section 5 and
6 respectively.

VEDIC MULTIPLIER USING 'URDHVA TIRYAGBHYAM'SUTRA.

The "Urdhva Tiryagbhyam" sutra is a Vedic multiplier that can be applied to all cases of multiplication. These words "Urdhva " and "Tiryagbhyam " are derived from Sanskrit Literature, which means, vertically cross wise multiplication.
This "Urdhva Tiryagbhyam" sutra for multiplication is illustrated for binary numbers and is as shown in Figure 1. The two binary numbers are represented as a3a2a1a0 and b3b2b1b0. The generation of the partial products using the concept of parallelism is as detailed in Figure 1.

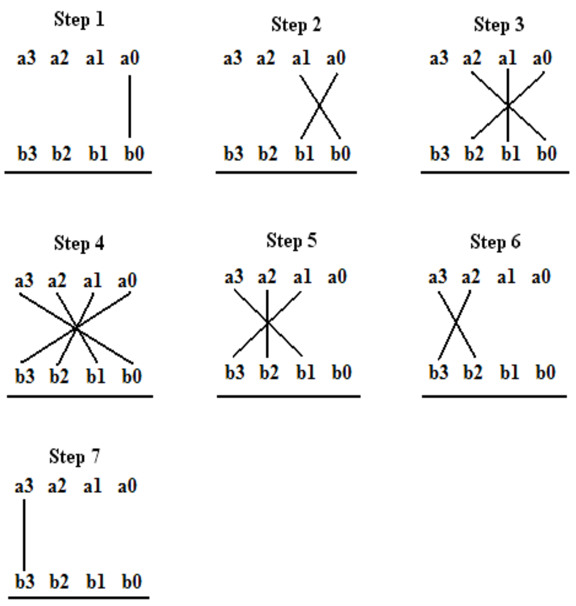


Fig 1: *Line diagram for multiplication of Two 4-bit numbers*

The result of this multiplication would be more than 4bits and is represented as r7r6r5r4r3r2r1r0. Thus we get the following expression.

r0 = a0b0;(1) c1r1=a1b0+aob1;(2) c2r2=c1+a2bo+a1b1+a0b2;(3) c3r3=c2+a3b0+a2b1+a1b2+aob3;(4) c4r4=c3+a3b1+a2b2+a1b3;(5) c5r5=c4+a3b2+a2b3;(6) c6r6=c5+a3b3(7)
r7=c7(end around carry);(8)

Where r_n is the result bit and c_n is the carry. This parallel approach requires 16 XOR gates.

The proposed 16 bit multiplier uses a structured architecture where 8 bit multiplier is the main module and the four bit Vedic multiplier module is used as a sub module in 8 bit multiplier.

The implementation of a basic 4bit multiplier module uses parallel architecture which consists of 8 Half adders and 7 full adders. The result is obtained after 7 stages. This architecture is implemented using VHDL on Xilinx 10.1ISE. When synthesized using Device Spartan2 XC2s100 -5 tq 144 produced a delay of 21 ns, and utilized 18 out of 1200 slices and 33 out of 2400 LUTs.

The motivation for this work was obtained by comparing
the Booth Multiplier with Vedic Multiplier. The Vedic Multiplier uses only16 XORs as against Booth Multiplier which used 7 four bit adders and twelve 4:1 Multiplexers. More over a delay reduction of 20% and 50% was obtained for
4bit and 8bit multipliers respectively.

PROPOSED ARCHITECTURE FOR 16 X 16 MULTIPLIER.

The architecture of 16x16 Vedic multiplier using "Urdhva Tiryagbhyam."Sutra is shown in Fig.2. The 16x16 Vedic multiplier architecture is implemented using four 8x8 Vedic multiplier modules, 3 stages of 8bit carry save adders as in figure 3.

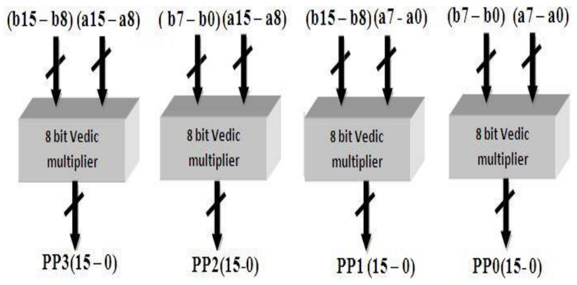


Fig 2: *Block diagram for individual multiplication result*

Each of the eight bit multipliers are implemented using the basic 4 bit multiplier module. Multiplication Result
=PROD(31-24) & PROD(23-16) & PROD(15-8) & PROD(7-
0),where & = concatenate operation. The proposed architecture uses three stages of 8-bit carry save adders to generate the final
32-bits product. The carry from the third stage is ignored.

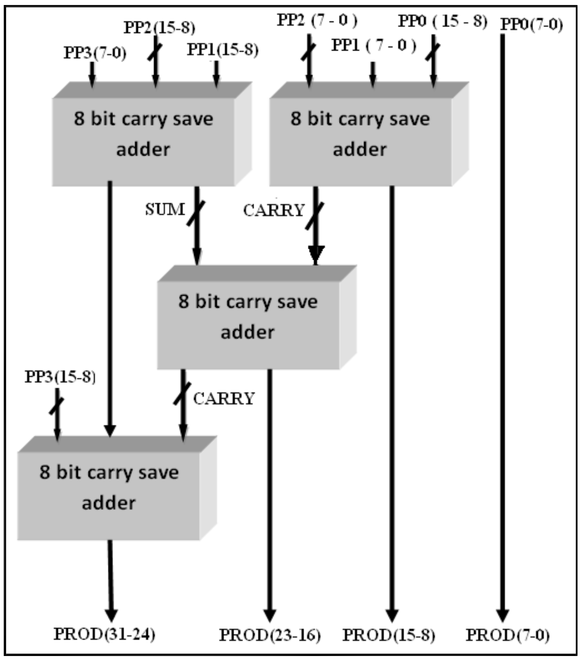


Fig 3: *Complete block diagram for proposed architecture*

A carry save adder is different from the conventional adders, since the carry generated is saved and not propagated during the generation of the partial sum bits. These saved carry bits are added to the partial sum bits to obtain the final sum.

The total gate delay of the carry save adder is O(m + log (n + m)), where m is number of inputs each being n bits long. Using Carry save adders in the proposed algorithm the overall speed has been significantly improved. Addition cannot be performed in time in parallel adder because the carry information must be propagated. In carry save addition, we decline any carry information from directly passing on the next stage[6].

RESULTS AND DISCUSSION-

Table-1 displays the comparison of synthesis results of the proposed 16x16 Vedic multiplier with the proposed architecture of [3].

Table-1. *Comparison of Synthesis Results of proposed 16x16 Bit Vedic Multiplier with [3].*

Device Spartan2 XC2S100: -5 tq144	16 X 16 architecture [3]	16 X 16 proposed architecture
Delay	74.852 ns	52.281 ns
Number of Slices	412/1200 (34%)	483/1200 (40%)
Number of LUTs	716/2400 (30%)	855/2400 (35%)
Levels of Logic	35	24

Table-2 displays the comparison of synthesis results of the proposed 16x16 Vedic multiplier with the proposed architecture of [4].

Table-2. *Comparison of Synthesis Results of proposed 16x16 Bit Vedic Multiplier with [4].*

Device Virtex 4vlx15sf363-12	16 X 16 architecture [4]	16 X 16 proposed architecture
Delay	33.391 ns	18.18 ns

Table-3 *HDL Synthesis Report of the proposed VedicMultiplier.*

Device: Spartan2 XC2S100: -5 tq144	**4 bit**	**8 bit**	**16 bit**
XORs	9- XOR 2 7- XOR 3	36- XOR 2 84- XOR 3	144- XOR 2 560- XOR 3
Delay	21ns	32.287 ns	52.281 ns
Levels of Logic	9	14	24

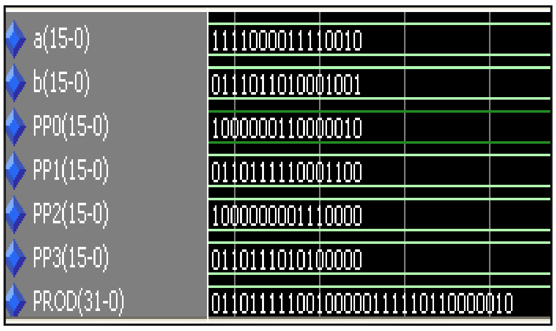
From Table-1 the proposed architecture for a 16 X 16 multiplier is found to have a delay of 52.281 ns., with 20.291 ns for the logic and 31.990 ns for routing. The speed is optimized compared to [3] which has 24.752ns for the logic and 47.1 ns for routing , giving a delay reduction of 30% .

From Table -2 the proposed architecture for a 16 X 16 multiplier is found to have a delay of 18.181ns compared to the delay found in [4]. This gives a delay reduction of 46 %.
The Implementation for a 16 X 16 multiplier is done in stages of 4 bit and 8 bit multiplier. In 8 bit multiplier 4 bit modules are used and in 16 bit implementation 8-bit multiplier are used as sub modules.

Table -3 summarizes the work done. The 4 bit multiplier uses a total of 16 XORs using parallel architecture as indicated in figure 1. The 8 bit multiplier uses 120 XORs.

Simulation result

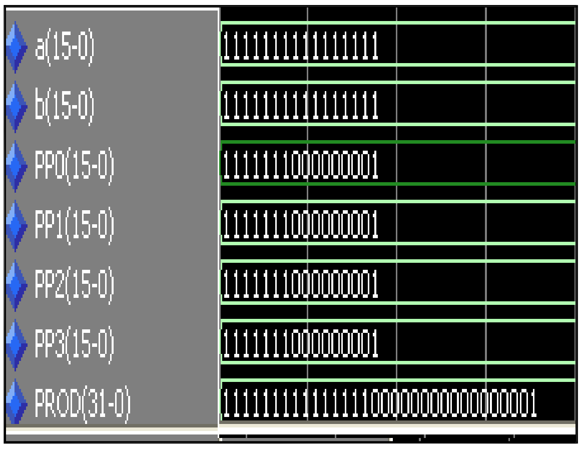
For designing a 16x16 multiplier the coding is done using VHDL. The algorithm is verified for all possible input combination. For fast operation, 3 stages of carry save adder is being used.



```
a(15-0)     1111000011110010
b(15-0)     0111011010001001
PP0(15-0)   1000000110000010
PP1(15-0)   0110111110001100
PP2(15-0)   1000000001110000
PP3(15-0)   0110111010100000
PROD(31-0)  01101111100100000111110110000010
```

Fig 4. *Simulation result for 16 bit proposed multiplier*

Figure 4 shows the Model SIM results for 16 bit proposed multiplier for a =6168210, b = 3034510 yielding a product of 871740290 10.
Similarly for a = b = 6553510 , the product is 429483622510 as per figure 5.
In figures 4 and 5 PP0,PP1,PP2,PP3 each is a 16 bit partial product obtained from 8 X 8 multiplier.



```
a(15-0)     1111111111111111
b(15-0)     1111111111111111
PP0(15-0)   1111111000000001
PP1(15-0)   1111111000000001
PP2(15-0)   1111111000000001
PP3(15-0)   1111111000000001
PROD(31-0)  11111111111111100000000000000001
```

Fig 5. *Simulation result for 16 bit proposed multiplier*

CONCLUSION

The proposed multiplier architecture shows an improvement in speed when compared to the existing systems [3] [4]. The 16x16 Vedic multiplier using "Urdhva Tiryagbhyam "Sutra implemented using carry save adders have resulted in terms of 30% to 46% reduction in delay with 15% increase in area. It can be well suited for multiplication of numbers with more than 16 bit size. This work can be further extended for area optimization.

REFERENCES

1. Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho,“Multiplier Design based on Ancient Indian Vedic Mathematics”, International SoC,Design Conference, Nov. 2008, pp.65-68.
2. agadguru Swami Sri Bharati Krisna Tirthaji Maharaja, “Vedic mathematics”, Motilal Banarsidass Publishers Pvt. Ltd, Delhi, 2009.
3. Shamim Akhter “ VHDL implementation of fast NxN multiplier based on Vedic mathematic” , Circuit theory and design 2007, ECCTD 2007, 18th European Conference, Issue date 27-30 Aug. 2007, pg 472-475.
4. Prabha S Kasliwal, BP Patil, DK Gautam, “Performance Evaluation of Squaring Operation by Vedic Mathematics”, IETE Journal of research, Year 2011, Volume 57, Issue 1, Pg 39-41.
5. http:// www.asic-world.com/digital /arithmetic2.html
6. Prof. Loh, CS3220- Processor design, “carry save addition”, Spring 2005.