International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2019): 7.583

Realization of Low Power Multiplier Design Based on Vedic Iteration Technique

Charu Verma[1], Ankit Kumar Singh[2], Ravi Tiwari[3]

[1]Shri Shankaracharya Group of Institution, Shri Shankaracharya Technical Campus, Junwani, Bhilai, India
charuverma9876[at]gmail.com

[2]National Institute of Technology, NIT Campus, G.E. Road, Raipur 492010, India
singhankit491[at]gmail.com

[3] Shri Shankaracharya Group of Institution, Shri Shankaracharya Technical Campus, Junwani, Bhilai, India
ravitiwati2605[at]gmail.com

Abstract: ***Vedic mathematics is the ancient methodology of the Indian mathematics which has a unique technique for arithmetic computations based on 16 Sutras (Formula). Transistor level implementation (ASIC) of Vedic Mathematics based 32-bit multiplier for high-speed low power processes is reported in this paper. Simple Boolean logic is combined with Vedic formula, which reduces the partial products and sums generated in one step, reduces the carry propagation from LSB to MSB. The implementation methodology ensures substantial reduction of propagation delay in comparison with Wallace Tree (WTM), modified Both Algorithm (MBA), Baugh Wooley (BWM) and Row Bypassing and Parallel Architecture (RBPA) based implementation which are most used architectures. The functionality of these circuits was checked and performance parameters like propagation delay and dynamic power consumption were calculated by spice spectra using standard 90nm CMOS technology. The propagation delay of the resulting 32x32 multiplier was only ~1.06 us and consumes ~132 uW power. The implementation offered significant improvement in terms of delay and power from earlier reported ones.***

Keywords: Vedic Mathematics, Urdhva-triyakbhyam sutra, FPGA

1. Introduction

Multipliers are extensively used in FIR filters, Microprocessors, DSP, and communication applications. For higher order multiplications a huge number of adders or compressors are to be used to perform the partial product addition. The need of low power and high speed. Multiplier is increasing as the need of high-speed processors are increasing. The Vedic multiplication technique is based on 16 Vedic sutras or aphorism, which are word formulae describing natural ways of solving a whole range of mathematical problems. The mathematical operations using, Vedic method are very fast and requires less hardware, this can be used to improve the computational speed of processors. In this paper we have implemented the Vedic multiplier in VHDL**.** [10]

The Urdhva-triyakbhyam sutra is basically vertically and crosswise. In this method the partial products are generated simultaneously which itself reduces the delay makes this method fast. The method is explained below for two, three-bit numbers A and B where A= a2a1a0 and B=b2b1b0 as given below. Firstly, the least significant bits are multiplied which gives the least significant bit of the products (Vertical). Then, the LSB of multiplier and next higher bit of the multiplicand (Crosswise). The sum gives second bit of the product and vertical multiplication and addition to give the sum and carry. The sum is the corresponding bit of the product and the carry is again added to the next stage multiplication and addition of two bits except the LSB. The same operation continues until the multiplication of the two MSBs to give the MSB of the product.

$$S0 = a0b0; \quad (1)$$
$$C1s1 = a1b0 + a0b1; \quad (2)$$
$$C2s2 = c1 + a2b0 + a1b1 + a0b2; \quad (3)$$
$$C3s3 = c2 + a1b2 + a2b1; \quad (4)$$
$$C4s4 = c3 + a2b2; \quad (5)$$

2. Urdhva Tiryakbhyam Sutra

This sutra is based on "Vertically and Crosswise" multiplication technique. It makes almost all the numeric computations faster and easier. The profit of multiplier based on this sutra over the others is that with the increment in number of bits, area and delay increase at a lower rate in comparison to others. The judgment between conventional and Vedic mathematical realized in VLSI for RSA algorithm, ALU, CURVE encryption etc. It displays that Vedic mathematical method is fast and modest. 16 X 16 multiplier by "Urdhva Tiryagbhyam Sutra" is offered and prolonged by using 'Nikhilam Sutra' 16 X 16 multiplier modules practices two 8 X 8 modules, one 16-bit carry save adder and two 17-bit full adder stages are realized here. The carry save adder raises the speed of addition of partial products. The multiplier is realized in SPARTAN 2 FPGA Device XC2S30 – 5pq208. The 'Urdhva Tiryagbhyam Sutra' plus 'Nikhilam Sutra' multiplication methods are established to be speedy when magnitude of together operands is extra than partial of their extreme values.

The realized multiplier is based on the ancient Vedic multiplication method. At this point the 'Urdhva Tiryakbhyam Sutra' are used for multiplication. The multiplier founded on this technique is equated with the modern multiplier to highpoint the power and speed rewards

in the Vedic Multipliers. To check the Vedic multiplier BIST (Built in Self – Test) is realized and it is established Fault free. The coefficient multipliers are castoff. Outcomes are stored in ROM which rises power consumption. The multiplier is realized using VHDL and Spartan 2G FPGA. The simulation outcomes are offered based on power and time delay. Multiplication is an important fundamental function in arithmetic operations. Multiplication- based operations such as Multiply and Accumulate (MAC) and inner product are among some of the frequently used computation intensive Arithmetic Functions (CIAF) currently implemented in many Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform (FFT), filtering and in microprocessors m its arithmetic and logic unit. Since multiplication dominates the execution tune of most DSP algorithms, so there is a need of high-speed multiplier. Currently, multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip. The demand for high-speed processing has been increasing because of expanding computer and signal processing applications. Higher throughput arithmetic operations are important to achieve the desired performance in many real-time signal and image processing applications. One of the key arithmetic operations in such applications is multiplication and the development of fast multiplier circuit has been a subject of interest over decades. Multiplier based on Vedic Mathematics is one of the fast and low power multipliers, employing this technique in the computation algorithms will reduce the complexity, execution time, power etc. The Vedic based multiplier is compared with binary multiplier (partial products method)

2.1 Objective of the proposed research work

The main objective of this research work is that to implement the Arithmetic operations using Vedic mathematics are a growing trend in the FPGA community. It has become critical to optimize multiplication units for standard FPGA technology. But the FPGA design space is very different from the VLSI design space: thus, optimizations for FPGA environment constrains the design space such that only limited parallelism is effectively exploited to reduce the latency. Obtaining the right balances among the clock speed, the latency and the area in FPGAs is particularly challenging. This article presents the implementation details for binary multiplier and Vedic Multiplier for FPGAs and the implementation of MAC using FPGA technology.

3. Methodology

Now a day, the implementation of Vedic multipliers in different fields is growing up. Ashwanth and Premanandha, 2013 proposed a Urdhva Tiryakbhyam Q-format multiplier using Verilog hardware description language and structural form of coding. The Q15 and Q31 format multipliers are designed using the building block of 8 X 8 and 16 X 16 Urdhva Tiryakbhyam integer multipliers which in turn are made up of 4 X 4 multiplier blocks. Amina Naaz et al. (2014) applied the design of Vedic multiplier using Carry Select Adder (CSLA) for FIR architecture.

3.1 Multiplier Design

3.1.1 Array Multiplier

Array multiplier is an efficient layout of a combinational multiplier. Multiplication of two binary number can be obtained with one micro-operation by using a combinational circuit that forms the product bit all at once thus making it a fast way of multiplying two numbers since only delay is the time for the signals to propagate through the gates that forms the multiplication array. In array multiplier, consider two binary numbers A and B, of m and n bits. There are mn summands that are produced in parallel by a set of mn AND gates. n X n multiplier requires n (n-2) full adders, n half - adders and n2 AND gates. Also, in array multiplier worst case delay would be (2n+1) td. Array Multiplier gives more power consumption as well as optimum number of components required but delay for this multiplier is larger. It also requires larger number of gates because of which area is also increased; due to this array multiplier is less economical. It is a fast multiplier, but hardware complexity is high.

3.1.2 Booth Multiplier

Another improvement in the multiplier is by reducing the number of partial products generated. The Booth recording multiplier is one such multiplier; it scans the three bits at a time to reduce the number of partial products. These three bits are the two bits from the present pair; and a third bit from the high order bit of an adjacent lower order pair. After examining each triplet of bits, the triplets are converted by Booth logic into a set of five control signals used by the adder cells in the array to control the operations performed by the adder cells. International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010 50 To speed up the multiplication Booth encoding performs several steps of multiplication at once. Booth's algorithm takes advantages of the fact that an adder subtraction is nearly as fast and small as a simple adder. From the basics of Booth Multiplication, it can be proved that the addition/subtraction operation can be skipped if the successive bits in the multiplicand are same. If 3 consecutive bits are same, then addition/subtraction operation can be skipped. Thus, in most of the cases the delay associated with Booth Multiplication are smaller than that with Array Multiplier. However, the performance of Booth Multiplier for delay is input data dependent. In the worst case the delay with booth multiplier is on per with Array Multiplier.

4. VHDL Code for Multiplier Design

```
library Ieee;
use ieee.std_logic_1164.all;
Entity vedic_multi_struct4 is
port (a,b:in bit_vector(3 downto 0);
        op:out bit_vector(6 downto 0));
end vedic_multi_struct4;

Architecture vm_struct4 of  vedic_multi_struct4 is
signal
s0,s01,s10,s1,c1,s02,s11,s20,s2,c2,s03,s12,s21,s30,s3,c3,s13
,s22,s31,s4,c4,s23,s32,s5,c5,s33,s6,c6:bit;
```

```
component and2
port(m,n:in bit;l:out bit);
end component;

component xor2
   port(a,b:in bit;c:out bit);
end component;


component and3
port(m,n,o:in bit;l:out bit);
end component;

component xor3
   port(a,b,c:in bit;d:out bit);
end component;


component and4
port(m,n,o,p:in bit;l:out bit);
end component;

component xor4
   port(a,b,c,d:in bit;e:out bit);
end component;

component and5
 PORT(m,n,o,p,q:in bit;l:out bit);
 end component;

component xor5
 port(a,b,c,d,e: in bit;
      f: out bit);
    end component;
    begin
X1:and2 port map(a(0),b(0),S0);
X2:and2 port map(a(0),b(1),S01);
X3:and2 port map(a(1),b(0),S10);
X4:xor2 port map(s01,s10,S1);
X5:and2 port map(s01,s10,c1);
X6:and2 port map(a(0),b(2),S02);
X8:and2 port map(a(1),b(1),S11);
X9:and2 port map(a(2),b(0),S20);
X10:xor4 port map(c1,s02,s11,s20,s2);
X11:and4 port map(c1,s02,s11,s20,c2);
X12:and2 port map(a(0),b(3), s03);
X13:and2 port map(a(1),b(2), s12);
X14:and2 port map(a(2),b(1), s21);
X15:and2 port map(a(3),b(0), s30);
X16:xor5 port map(c2,s03,s12,s21,s30,s3);
x17:and5 port map(c2,s03,s12,s21,s30,c3);
X18:and2 port map(a(1),b(3), s13);
X19:and2 port map(a(2),b(2), s22);
X20:and2 port map(a(3),b(1), s31);
X21:xor4 port map(c3,s13,s22,s31,s4);
x22:and4 port map(c3,s13,s22,s31,c4);
X23:and2 port map(a(2),b(3), s23);
X24:and2 port map(a(3),b(2), s32);
x25:xor3 port map(c4,s23,s32,s5);
x26:and3 port map(c4,s23,s32,c5);
X27:and2 port map(a(3),b(3), s33);
X28:xor2 port map(c5,s33,s6);
X29:and2 port map(c5,s33,c6);
end vm_struct4;
```

5. Result

In this paper we are evaluating the performance of the proposed high speed low power Vedic multiplier by comparing this design with a conventional Array Multiplier and Booth Multiplier. These multipliers are implemented using VHDL to get the power report and delay report the multipliers are synthesized using Xilinx ISE tool and Spartan 2E FPGA is used. The graphical output screen of the proposed design is shown in figure:

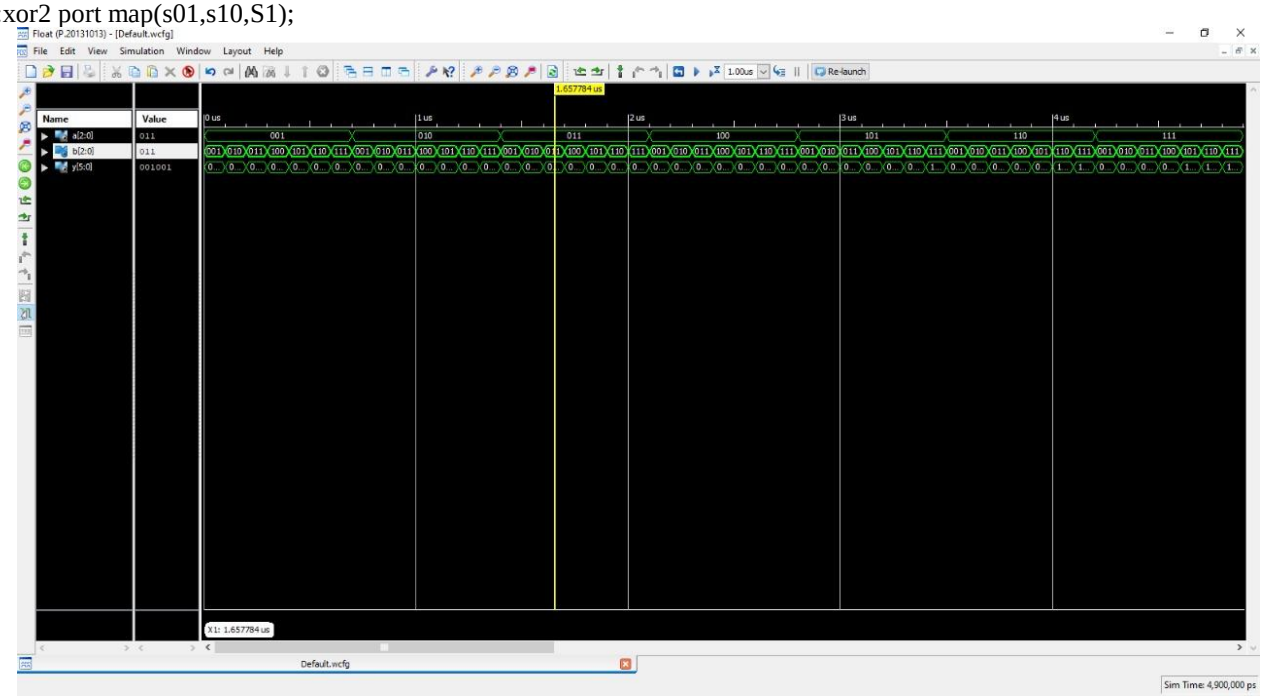

Figure 1: Proposed Design Multiplier Output

References

[1] Jagadguru Swami Sri Bharti Krishna Tirthaji Maharaja, "Vedic Mathematics Sixteen Simple Mathematical Formulae from the Veda", Delhi 1965.
[2] Harpreet Singh Dhillon and Abhijit Mitra, "A Reduced – Bit Multiplication Algorithm for Digital Arithmetic", International journal of Computational and mathematical Science, 2 (2) 2008
[3] Himanshu Thapliyal and M.B Srinivas, "An Efficient Method of Elliptic Curve Encryption Using Ancient Indian Vedic Mathematics, IEEE , 2005.
[4] Charles E. Stroud, "A Designer's Guide to Built-In Self-Test", University of North Carolina at Charlotte, 2002 Kluwer Academic Publishers New York, Boston, Dordrecht, London, Mosco
[5] Goh Keng Hoo, "Verilog Design of Input/Output Processor with Built-In-Self-Test", University Technology Malaysia, April 2007.
[6] Michael L. Bushnell, Vishwani D. Agrawal," Essentials of Electronics Testing for Digital, Memory and mixed signal VLSI circuits", Kluwer Academic Publishers.
[7] T.W. Williams and K. Parker, "Design for Testability-A survey", IEEE Trans. On Computers, January 1982.
[8] Prabha S., Kasliwal, B.P. Patil and D.K Gautam, "Performance Evaluation of squaring Operation by Vedic Mathematics", IETE journal of Research Jan-Feb 2011.
[9] Metin mete OZBLEN, Mustafa GOK, A single/Double Precision Floating-Point Multiplier Design for Multimedia Applications, "Journal of Electrical & Electronics Engineering 2009.
[10] Chetana Nagendra, Robert Michael Owens, and Mary jane Irwin, "Power-Delay Characteristics of CMOS Adders", IEEE Transactions on Very Large-Scale integration (VLSI) Systems, Vol. 2, No. 3, September 1994.
[11] S.S Kerur Prakash Narchi Jayashree C N, Harish M kittur, Girish V A, "Implementation of Vedic Multiplier for Digital Signal Processing," International Journal of Computer Applications (IJCA) 2011.
[12] Aniruddha Kanhe, 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 (IJCSC) Vol.3, No.1 January June 2012.
[13] Kavita Khare, R.P. Singh, Nilay khare, "comparision of pipelined IEEE-754 standard floating point multiplier with unpipelined multiplier" journal of scientific & Industrial Research Vol.65, pages 900-904 November 2006.