ISSN ONLINE (2395-1680)
DOI: 10.21917/ijme.2015.0007

IMPLEMENTATION OF HIGH SPEEDARITHMETIC LOGIC USING VEDIC MATHEMATICS TECHNIQUES

R. Jaikumar[1], P. Poongodi[2] and R. Lavanya[3]

[1]*Department of Electronics and Communication Engineering, RVS college of Engineering and Technology, India*
E-mail: jaikumarer@gmail.com
[2]*Department of Electronics and Communication Engineering, VSB Engineering College, India*
E-mail: poongodiravikumar@yahoo.co.in
[3]*Department of Electronics and Communication Engineering, Sri Ranganathar Institute of Engineering and Technology, India*
E-mail: lavanyaraju4985@gmail.com

Abstract

The Digital signal processing plays a vital role in communication applications. In Digital signal processing, ALU is an important functional unit. The main objectives of VLSI architecture design are the speed and power. Here we are going to design low power, high speed ALU by using vedic mathematics technique. This paper describes the design of more efficient high speed 4 × 4 bit arithmetic logic unit based on Vedic multiplication technique. It can perform arithmetic and logical operations. Generally the digital domain based design depends on the performance of ALU and hence the high performance ALU is predominant. The ALU speed is mainly based on the speed of multiplier. There are so many algorithms used for multiplication technique. Our work has proved that Vedic multiplication technique is the best algorithm in terms of speed. The Vedic multiplication algorithm is based on 16 sutra. Here we are using Urdhva Tiryakbhyam [1].The 4-bit ALU was designed using Vedic mathematics and the performance compared with 4-bit Array multiplier based ALU. The ALU which is shown here is very efficient in terms of speed and power dissipation.

Keywords:

Vedic Mathematics, Low Power, Urdhva Tiryakbhyam, Speed

1. INTRODUCTION

We know that the ALU is a mathematical unit. The mathematical unit performs arithmetic (addition, subtraction, multiplication) and logical operations (AND, OR, INVERTER). That is why the ALU is called heart of microprocessor, microcontroller and digital signal processor. Our proposed 4-bit Vedic ALU performs arithmetic operations and performs logical operations. It performs three arithmetic operations and five logical operations. Nowadays high speed processor requirements are gradually increasing. Recent days processor having more than two ALU (quad core, Core2 duo) to improve the processor speed. So if we improve performance of the internal architecture of the ALU speed mainly depends on the speed of multiplier. Thus we have designed the high speed 4-bit Vedic multiplier using Vedic Mathematics and replaced the existing multiplier unit in ALU. Our Vedic multiplier is very fast and occupies less hardware. The switching activity of the multiplier also reduced leads to reduction of dynamic power consumption [2]. The dynamic power is dominant (around 90 % of total power) in VLSI architecture design.

The proposed ALU is implemented on FPGA hardware. The ALU diagram is given above Fig.1. The input is given to the ALU by using toggle switch on FPGA board and the data can be processed. Then the result can be observed through LED.

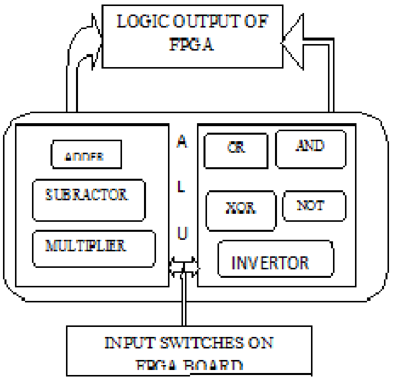


Fig.1. Basic Blocks in ALU Architecture

2. ANCIENT VEDIC MATHEMATICS

Vedic mathematics [2] is the ancient system of Indian mathematics which was re-discovered by Sri Bharatikrishnan Tirthaji (1884-1960). The Sanskrit word Veda means storing of all knowledge. The Vedas are ancient writings whose data is not distributed completely to this world and some of that were missed during several centuries of BC. Hundred years ago Sanskrit scholars translated the Vedic documents and got surprised about the depth of knowledge enriched in that. The system rediscovered by Sri Bharati Krishna is based on sixteen formulas and some sub formulas.

The mathematical problems like algebra, arithmetic, geometry or trigonometry can be solved mentally by using these formulas [8]. The sutras can be related to natural mental functions such as completing a whole, noticing analogies, generalization and so on.

2.1 EXPLANATION OF URDHVA TIRYAKBAYAM METHOD

The proposed our 4-bit Vedic ALU is highly depends upon the multiplier. Here we are using Vedic algorithm which is known as urdhva tiryakbhyam [8] (vertically and crosswise) method. This formula is simply used to multiply two decimal number system. In Our work, the same idea has implemented in binary number system and verified by simulation tool. This is a general formula suitable for all the case of multiplication and it also suitable for implementing in FPGA. Here there is no carry propagation due to addition of partial products. Here the partial protect and the sums are calculated in parallel and our multiplier depends on this sutra.

2.2 MULTIPLICATION OF TWO DECIMAL NUMBER 260 × 756

Urdhva triyakbhyam scheme as shown in below. We consider two decimal numbers 260 and 756 and multiply those numbers (260 × 756) using urdhva triyakbhyam [1] method. The two decimal numbers are multiplied and this result is added from previous step carry. It generates one of the digits of final product and all other digits are carry. Likewise, this process goes on. If more than one line is there in one step then all the results are added with previous carry. In each step the least significant digit act as one of the result digit.

The other digit is acting as carry for next step. Initially, the carry will be considered as zero.

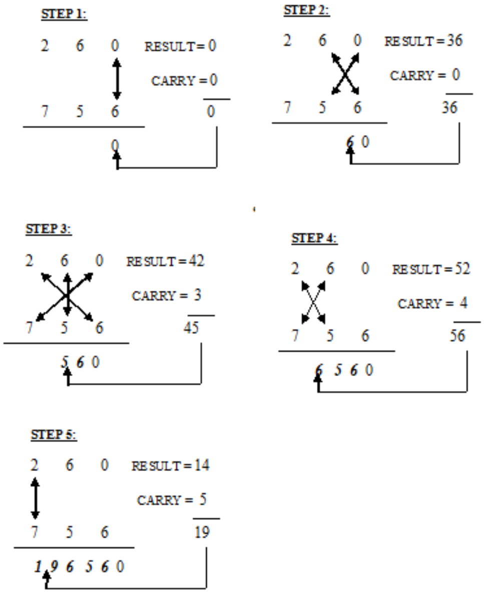


Fig.2. Multiplication of Two Decimal Numbers

The algorithm is also called as vertical cross wise multiplication. The algorithm is efficient for smaller number of multiplication. If the number of bits higher means we may use Nikhilam navatascaramam Dasatah algorithm. But the algorithm [5] has a problem to select the proper base number. The following simple 4 × 4 multiplication example can explain the algorithm.

The procedure will be complex if the number of bits are high, also the propagation delay will increase due to crosswise operation. In above example the * symbol indicates AND operation between the operands and the + symbol indicates the OR operation. Here the multiplicand is 10 and the multiplier is 2. We are getting the result as 20. We can perform any smaller number multiplication using above procedure. The hardware architecture can be obtained by using simple logic gates.

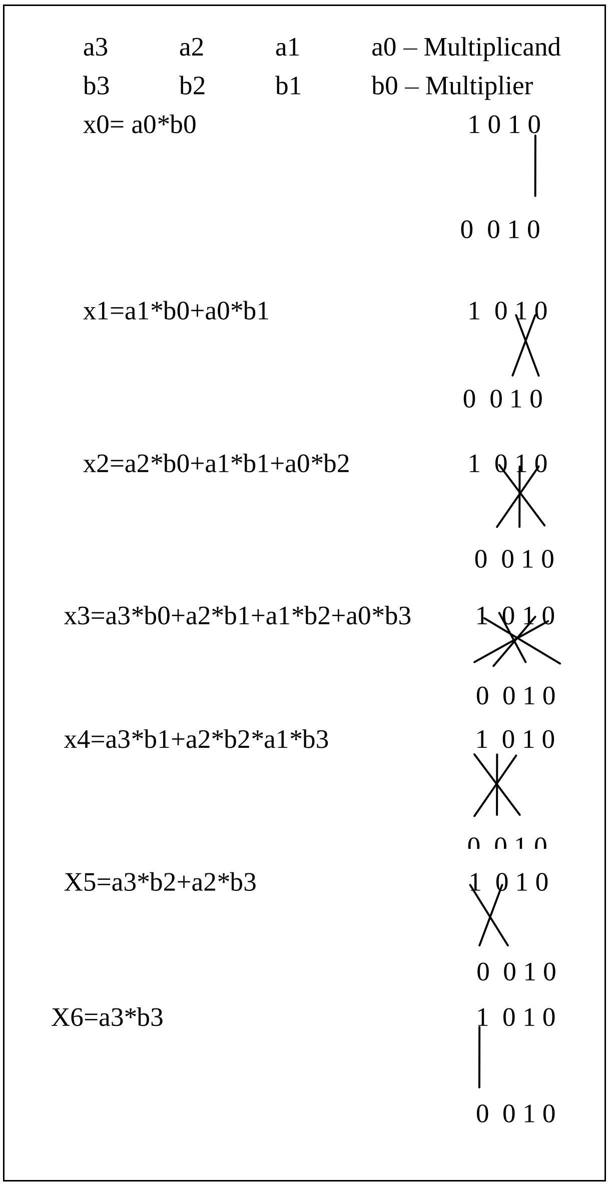


Fig.3. Multiplication of Two 4-Bit Binary Numbers

3. THE PROPOSED MULTIPLIER ARCHITECTURE

The hardware architecture 2 × 2 bit two and 4x4 bit is displayed in following section. Here we are using urdhvatiryakbhyam (vertically and crosswise) sutra for multiply two binary number.

The main advantage of Vedic multiplier is generation of partial products and addition by done by parallel. It will make attractive for binary multiplication and reduces the multiplier delay. This is main motivation behind our work.

3.1 VEDIC MULTIPLIER FOR 4 × 4 BIT MODULES

This proposed Vedic 4-bit multiplier [4] is implemented using AND gate and adder. Here we are using 16 AND gate and 8 full adder and 3 half adder. This module used to reduce path delay. The proposed architecture is shown in Fig.4. Let us consider 4 × 4 multiplication X = x3x2x1x0 and Y = y3y2y1y0.The output produced by 8-bit with include carry bit. The 8-bit result is P = p7p6p5p4p3p2p1p0.

The implemented expressions are

$$P0=x0y0 \tag{1}$$

$$P1=x0y1+y0x0 \tag{2}$$

$$P2=c1+x2y0+x1y1+x0y2 \tag{3}$$

$$P3=c2+x3y0+x2y1+x1y2+x1y3 \tag{4}$$

$$P5=c3+x3y1+x2y2+x1y3 \tag{5}$$

$$P6=c4+x3y2+x2y3 \tag{6}$$

$$P7=c5+x3y3 \tag{7}$$

This formula is suitable for all the cases of multiplication.

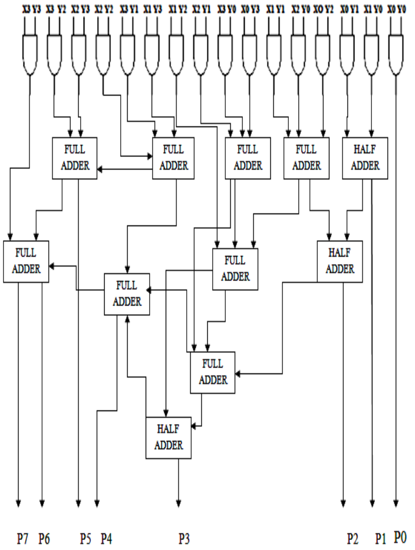


Fig.4. Block Diagram of 4 × 4 Multiplier Module

4. THE PROPOSED VEDIC ARITHMETIC LOGIG UNIT ARCHITECTURE

In this section, we will discuss about our proposed Vedic 4-bit ALU architecture as shown in Fig.5. Two inputs A and B are provided based on the selection line s_1, s_2, s_3 to perform arithmetic and logical operation. Here two different multiplexer such as 8:1 and two 2:1 multiplexer are used. One of the 2:1 multiplexer receives the input B and 2's complement input. If the selection $s_1 = 0$, $s_2 = 0$, $s_3 = 0$ the input B is added with input A directly, otherwise the input A and 2's complement are added together to generate the subtraction output. The 8:1 multiplexer receives the output of all the arithmetic operation and logical operation through this multiplexer the required output can be generated using select line. The generated output may be 4-bit or excess, which is using two state of output. The logical operations produce the result in 4-bits which is displayed in LSB output (3 to 0). The arithmetic operations produce the output which may be excess of 4-bits.This will be displayed in MSB output (7 to 4).The another 2:1 multiplexer is used to select the MSB of adder or multiplier and display it as output (7 to 4). For example if the result of adder is 10010,then the output displayed will be 0011 in LSB output(3 to 0) and the fourth bit '1' will displayed in MSB output(7 to 4).The result of multiplier will also be displayed in the similar way to adder.

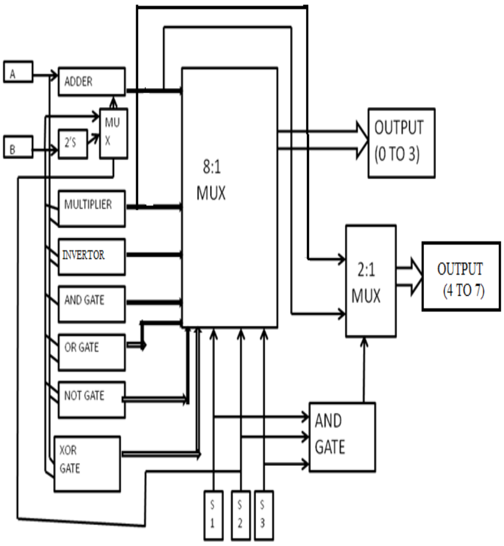


Fig.5. The 4-Bit Vedic ALU Architecture

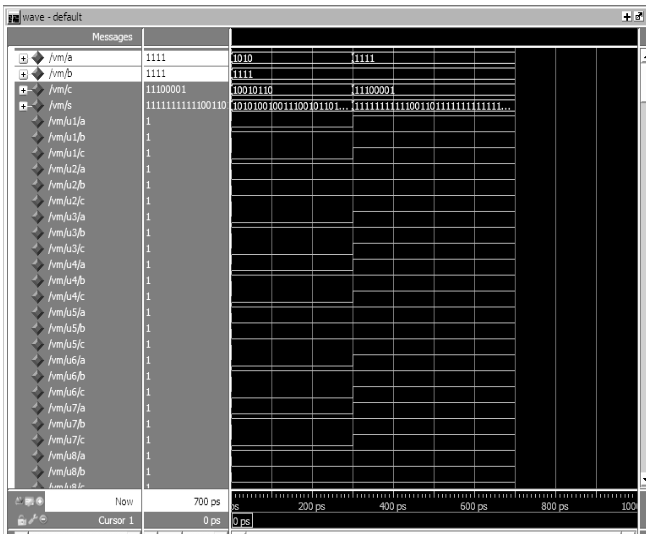
5. THE SIMULATION RESULT AND VERICATION

In this work, the 4 × 4 bit Vedic multiplier and ALU designed using VHDL. Logic synthesis is done using Xilinx version ISE9.2i project navigator and simulator integrated in the Xilinx package PQG208. The performance of circuit is evaluated on the Xilinx device XA3S50 and Xilinx family Automotive Spartan 3 and the simulation is done by using Modelsim 6.3.

5.1 SIMULATION RESULT OF 4-BIT MULTIPLIER

The simulation 4-bit multiplier result we have tested following input bits.

The 4-bit Vedic multiplier input are A = 1111(decimal value is 15) and B = 1111 (decimal value is 15) and the generated output is C = 11100001 (decimal value is 225)



```
/vm/a      1111                  1010                        1111
/vm/b      1111                  1111
/vm/c      11100001              10010110                    11100001
/vm/s      1111111111100110      1010100100111001011 01...   1111111111100110111111111111...
/vm/u1/a   1
/vm/u1/b   1
/vm/u1/c   1
/vm/u2/a   1
/vm/u2/b   1
/vm/u2/c   1
/vm/u3/a   1
/vm/u3/b   1
/vm/u3/c   1
/vm/u4/a   1
/vm/u4/b   1
/vm/u4/c   1
/vm/u5/a   1
/vm/u5/b   1
/vm/u5/c   1
/vm/u6/a   1
/vm/u6/b   1
/vm/u6/c   1
/vm/u7/a   1
/vm/u7/b   1
/vm/u7/c   1
/vm/u8/a   1
/vm/u8/b   1
/vm/u8/c   1
Now        700 ps
Cursor 1   0 ps                  0 ps
```

Fig.6. Simulation Result of 4-Bit Vedic Multiplier

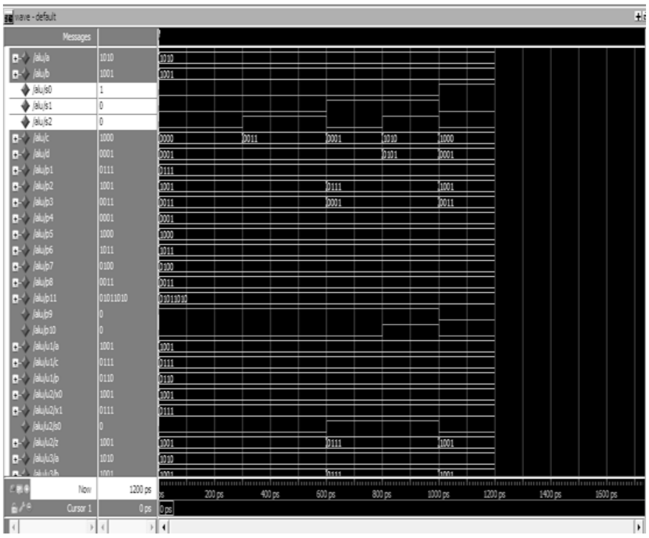


Fig.7. Simulation Result of 4-Bit ALU

5.2 SIMULATION RESULT OF 4-BIT ALU

The simulation result of the 4-bit ALU is discussed here.

The select line is given input is s = 0, s_1 = 0, s_3 = 0, then addition operation is selected and the given inputs are A = 0011 and B = 1001 and the produced output is c = 0011 and d = 0001.

The select line are given input is s = 0, s_1 = 0, s_3 = 1, the subtraction operation is selected and the given inputs are A = 1010 and B = 1001 and the produced output is c = 0001 and d = 0001

6. COMPARISION AND DISCUSSION

Table.1 shows the comparison of the delay time for the array multiplier and Vedic multiplier are 20.117ns, 17.473 respectively. From this, we conclude that the Vedic multiplier has the minimum delay time when compared to other multipliers.

Table.1. Delay Comparison of Multiplier

Multiplier	No of 4 bit slices	No of I/O's	No of LUT's	Delay in ns
Array Multiplier	40	20	45	20.117
Vedic Multiplier	18	16	31	17.473

Table.2 shows the comparison of the path delay for 4-bit Vedic based ALU and array multiplier based ALU. This simulation and synthesize result shows the Vedic ALU delay time is much lower than array based ALU.

Table.2. Comparison of Delay in ALU

ALU	No of 4 bit slices	No of I/O's	No of LUT's	Delay in ns
4*4 Vedic ALU	29	19	51	17.583
4*4 Array ALU	49	23	65	20.227

7. CONCLUSION

This paper represents a better and efficient method of designing the multiplier based on urdhva tiryakbhyamsutra. This multiplication technique reduces the path delay when the multiplication takes place. So that it will be able to execute the calculation with much greater speed as compared to other execution methods. Our proposed 4 × 4 multiplier has a path delay of 17.473ns. This multiplier is used in the 4-bit Vedic ALU. In this method, the performance will be much better, as it will reduce delay time. The path delay of 4-bit Vedic ALU is 17.583 which is relatively lesser than the path delay of the Array based ALU. So, the speed of Vedic ALU is much better than array ALU. If we increase the bitwidth of inputs then we need to perform more crosswise and vertical operations. This will lead to increase the power consumption.

REFERENCES

[1] Amartya Kumar Dutla, "Mathematics in ancient India", *Resonance*, Vol. 7, No. 4, pp. 4-19, 2002.

[2] Anvesh Kumar and Ashish Raman, "Low power ALU design by Ancient mathematics", *The 2nd International Conference on Computer and Automation Engineering*, Vol. 5, pp. 862-865, 2010.

[3] G. Ganesh Kumar and V. Charishma, "Design of high speed Vedic multiplier using Vedic mathematic techniques", *International Journal of Scientific and Research Publications*, Vol. 2, No. 3, pp. 1-5, 2012.

[4] Harpreet Singh Dhillon and Abhijit Mitra, "Reduced-Bit Multiplication Algorithm for Digital Arithmetic", *World Academy of Science, Engineering and Technology*, Vol. 19, pp. 719-724, 2008.

[5] Himanshu Thapliyal and Hamid R. Arabnia, "A Time-Area-Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics", *Proceedings of the International Conference on VLSI*, pp. 434-439, 2004.

[6] D. Krishnaveni and T .G. Umarani, "VLSI Implementation of Vedic multiplier with reduces delay", *International journal of Advanced Technology and Engineering Research*, Vol. 2, No. 4, pp. 10-14, 2012.

[7] Pushpalatha Verma and K. K. Metha, "Implementation of an efficient multiplier based on Vedic mathematics using Eda tool", *International Journal of Engineering and Advanced Technology*, Vol. 1, No. 5, pp. 75-79, 2012.

[8] S. G. Dani, "Ancient Indian mathematics – A conspectus", *Resonance*, Vol. 17, No. 3, pp. 236-246, 2012.

[9] Syed Azman bin Syed Ismail and Pumadai Siva subramaniyam, "Multiplication with the Vedic method", *Procedia - Social and Behavioral Sciences*, Vol. 8, pp. 129-133, 2010.

[10] Christian Soviani, Olivier Tardieu and Stephen A. Edwards, "Optimizing Sequential Cycles Through Shannon Decomposition and Retiming", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 26, No. 3, pp. 456-467, 2007.