An Efficient Design of Vedic Multiplier using New Encoding Scheme

Jai Skand Tripathi
P.G Student, United College of Engineering & Research, India

Priya Keerti Tripathi
P.G Student, Jaypee University of Engineering & Technology, India

Deepti Shakti Tripathi
P.G Student, National Institute of Technology, Rourkela, India

ABSTRACT
This paper presents a design of efficient Digital Vedic Multiplier using the Vedic sutras from ancient Indian Vedic mathematics. If we are looking towards the signal processing, we will find multipliers and adders plays a very important roll. In fact if we make our focus we can see speed of the Digital signal processing systems is mainly dependent on multipliers and adders. A processor requires more hardware and processing time during multiplication rather than addition and subtraction. In this paper we proposed a new digital Vedic multiplier structure based on a new encoding algorithm. We found that this algorithm reduces the number of partial products so reduces the adders. Thus multiplier is going to faster. In this paper we use Xilinx VHDL module for simulation of Encoder.

General Terms
FFT, Urdhva-Triryakbhyam sutra , Hardware complexity,

Keywords
pp_i – ith partial product, Vedic mathematics, Adders, Encoder

1. INTRODUCTION
The primordial and core of all the digital signal processors are arithmetic operations such as multiplication & addition. Adders are simply constructed but constructions of multipliers are complex. Two prominent measurements are associated with multiplication algorithms that are latency and throughput of system. The Latency is defined as the real computation delay of a function and Throughput is the measure of how many computations can be performed during given processing time. The execution time in DSP systems are dependent on multipliers so we need supreme multipliers.

The term Vedic is nascent from the Indian Sanskrit text known as Vedas which means accumulation of intellection. The present Vedic mathematics is due to the 8 years impassable research of Shri Bharati Krshna Tirtha on Vedas [1-2]. After his research of 8 years he finally concluded 16 sutras for Vedic mathematics which enables us to fast multiplication. Actually all the Vedic formula are based on the natural principles on which our mind works. There are various designs of Vedic multiplier [3-4-5]. Vedic multiplier works on the parallel processing and we know parallel processing is faster than serial processing. It has the speed better than simple digital multipliers but it has large number of gates.

2. VEDIC MULTIPLICATION
Vedic mathematics is based on 16 sutras. But in this paper we are using Urdhva-Triryakbhyam sutra for implementation. Urdhva-Triryakbhyam deals with the multiplication of numbers. It is also known as vertically and crosswise technique. Urdhva-Triryakbhyam sutra has been traditionally used for multiplication of decimal numbers. Here we are applying that sutra for digital multiplication. Line diagram for 4-Bit multiplication is shown in Figure-1.

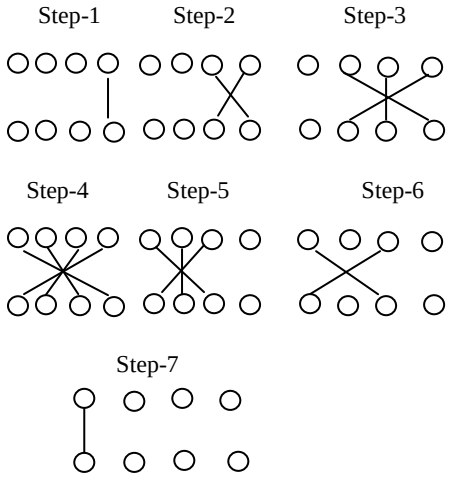

Figure 1:Line diagram of Vedic multiplication

Figure-2 shows the alternate method of Vedic multiplication algorithm. Here each block shows partial product. In Figure-2 we don't show carry but it is present there. Let we have two 8-Bit numbers P & Q. They can be represented as

$P= \sum_{i=0}^{7} a_i * 2\hat{}i$

$Q=\sum_{i=0}^{7} b_i*2\hat{}i$

Y=P*Q

If we have to find the product of P&Q, we have 8 partial products. There product after applying Urdhva-Triryakbhyam sutra and arranging all partial product in order to design a Digital circuit is given in Figure-2.Array multiplication & Vedic multiplication is approximately same with a little difference. The architecture of multiplier based on Urdhva-Triryakbhyam sutra is seen to be similar to array multiplier which uses an array of adders to find final product [6]. Vedic multiplication uses parallel processing. In Figure -2 a_ib_j represents product of ith bit of multiplicand with jth bit of multiplier and c_i represent carry due to ith summation. Equation for every output bit is given in Figure-3.:

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	
a_7b_7	a_7b_6	a_7b_5	a_7b_4	a_7b_3	a_7b_2	a_7b_1	a_7b_0	a_7
a_6b_7	a_6b_6	a_6b_5	a_6b_4	a_6b_3	a_6b_2	a_6b_1	a_6b_0	a_6
a_5b_7	a_5b_6	a_5b_5	a_5b_4	a_5b_3	a_5b_2	a_5b_1	a_5b_0	a_5
a_4b_7	a_4b_6	a_4b_5	a_4b_4	a_4b_3	a_4b_2	a_4b_1	a_4b_0	a_4
a_3b_7	a_3b_6	a_3b_5	a_3b_4	a_3b_3	a_3b_2	a_3b_1	a_3b_0	a_3
a_2b_7	a_2b_6	a_2b_5	a_2b_4	a_2b_3	a_2b_2	a_2b_1	a_2b_0	a_2
a_1b_7	a_1b_6	a_1b_5	a_1b_4	a_1b_3	a_1b_2	a_1b_1	a_1b_0	a_1
a_0b_7	a_0b_6	a_0b_5	a_0b_4	a_0b_3	a_0b_2	a_0b_1	a_0b_0	a_0

Figure 2: Alternate of Vedic Multiplication

$y_0 = a_0 b_0$

$y_1 = a_1b_0 + a_0b_1$

$y_2 = a_2b_0 + a_1b_1 + a_0b_2 + c_1$

$y_3 = a_3b_0 + a_2b_1 + a_1b_2 + a_0b_3 + c_2$

$y_4 = a_4 b_0 + a_3b_1 + a_2b_2 + a_1b_3 + a_0b_4 + c_3$

$y_5 = a_5 b_0 + a_4 b_1 + a_3b_2 + a_2b_3 + a_1b_4 + a_0b_5 + c_4$

$y_6 = a_6b_0 + a_5b_1 + a_4b_2 + a_3b_3 + a_2b_4 + a_1b_5 + a_0b_6 + c_5$

$y_7 = a_7b_0 + a_6b_1 + a_5b_2 + a_4b_3 + a_3b_4 + a_2b_5 + a_1b_6 + a_0b_7 + c_6$

$y_8 = a_7b_1 + a_6 b_2 + a_5 b_3 + a_4b_4 + a_3b_5 + a_2b_6 + a_1b_7 + c_7$

$y_9 = a_7b_2 + a_6b_3 + a_5b_4 + a_4b_5 + a_3b_6 + a_2b_7 + c_8$

$y_{10} = a_7b_3 + a_6 b_4 + a_5b_5 + a_4b_6 + a_3b_7 + c_9$

$y_{11} = a_7 b_4 + a_6b_5 + a_5b_6 + a_4b_7 + c_{10}$

$y_{12} = a_7b_5 + a_6b_6 + a_5b_7 + c_{11}$

$y_{13} = a_6b_7 + a_6b_7 + c_{12}$

$y_{14} = a_6b_7 + c_{13}$

$y_{15} = c_{14}$

Figure 3: Table for partial product generated by figure 2.

3. PROPOSED NEW ENCODING

3.1 Need For New scheme

The multiplication involves two processes first is generation of partial product and second is addition of generated partial product. We can enhance the speed of multiplier either by reducing the number of partial product or by using fast addition algorithms. In this paper we reduce the number of partial products. Although Vedic multiplier works on parallel processing but it has 8 partial products for 8 Bit Multiplier. For final product we have to add all that partial products which take a hues amount of hardware. Therefore delay is very large. In this paper, we introduce a method of encoding which reduces the partial product for 8-bit multiplier to half i.e. 4.

3.2 New Encoding Technique

In this encoding technique we tried to resolve the complexity of a multiplier.Figure-4 shows the grouping of multiplier bits for generation of code. In this algorithm we break a binary number in combination of 2-2 bits starting from LSB and provide a unique code to them according to encoding table given in Table1.

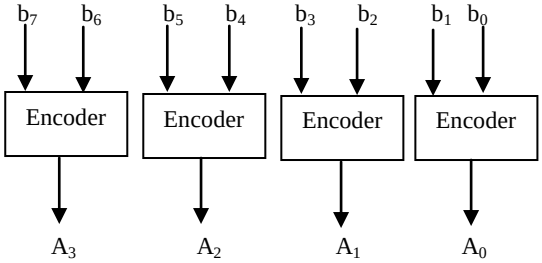

Figure 4: Grouping Bits of New Encoder

Table 1. New Encoding Table

b_{i+1}	b_i	A_i
0	0	0
0	1	1
1	0	2
1	1	3

Algorithm: If A_i ==0

$pp_{(i)}$ = 0;

end;

If A_i ==1

$pp_{(i)}$ = multiplicand;

end;

If A_i ==2

$pp_{(i)}$ = shift the multiplicand 1 position left;

end;

If A_i ==3

$pp_{(i)}$ = sum of $pp_{(i)}$ for code 1 & 2.

Algorithm for this encoding technique involves various steps:

Step1- Take the multiplier number and start grouping 2-2 Bits from LSB.

Step2- Compare the all received 2-bit number with encoding table.

Step3- According to encoding table we apply the multiplicand to the Adder with a shift of 2 bit, 4 bit 6 bit one by one.

Step4- Output of Adder is real product.

3.3 Proposed Multiplier

The block diagram of proposed multiplier architecture is given in Figure-5.It involves very less component and reduces the complexity. From Figure-5 we can see this architecture consists of only a adder, 3 shift registers, and an encoder circuit which can be simply designed. The working of encoder is shown in Figure-4. Multiplier & Multiplicand both are applied to the encoder in order to generate code.

3.4 Design of New Encoder circuit

The encoder is designed on Xilinx VHDL module. Input to this encoder is multiplicand and 2 bit of multiplier from LSB , the output is 1st partial product row then next 2 bits of multiplier is applied while multiplicand is fix it will give second partial product row similarly remaining bits will give 3rd and 4th partial product rows.

3.5 Comparison with various multiplication Algorithms

Vedic multiplier in section 2 there are hues amount of AND & OR gates. This needs more space and increases the cost of Multiplier. Multipliers involves multiplication and addition as given in Table-2[7] From the comparison in Table- 2 it is clear that after using this encoding technique there is a dramatic change in hardware structure After encoding multiplication part vanishes. As we can see in Figure-6 the number of partial product after encoding is just half. So power consumption is less. If we compare our encoder with Booth encoder for unsigned numbers then we find the Same work is done by Radix-4 encoder whose total gate delay is 17.431ns[12] which is large. Radix 4 Booth encoder is more complex than our encoder. It groups 3 bits for encoding out of which on bit is overlapping.

Table 2. Comparison between different Techniques

Bit length	Number Of Calculation					
	Conventional		Vedic		New Vedic	
	M	A	M	A	M	A
4	16	15	16	9	-	5
8	64	77	64	53	-	29

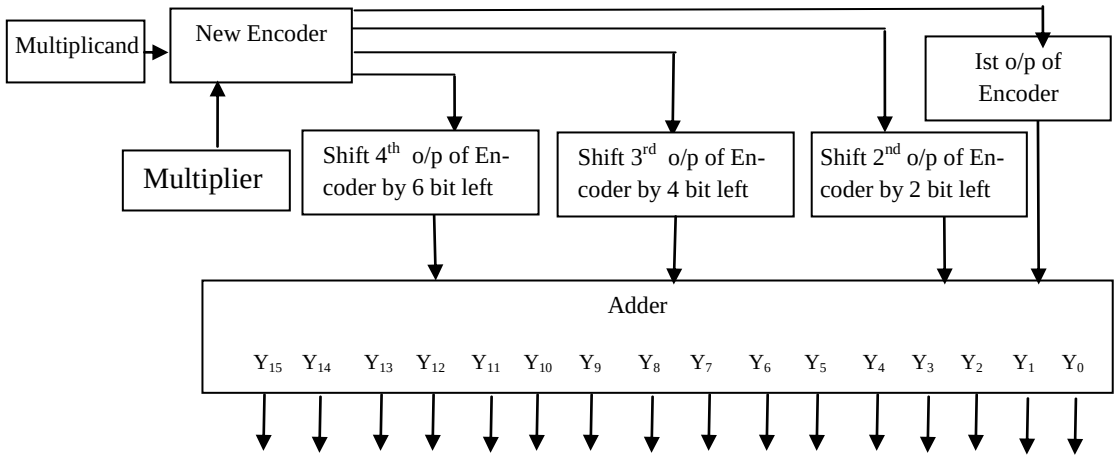

Figure 5: Block level Architecture of proposed Multiplier

From here we can see that after applying encoding technique to Vedic multiplier we get a great reduction. Normally we have 8 partial product row for 8*8 bit Multiplier but here we have 4 Partial product row.

$$
\begin{array}{rcccccccc}
 & a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\
\times & b_7 & b_6 & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \\
\hline
 & PP_{07} & PP_{06} & PP_{05} & PP_{04} & PP_{03} & PP_{02} & PP_{01} & PP_{00} \\
 & PP_{17} & PP_{16} & PP_{15} & PP_{14} & PP_{13} & PP_{12} & PP_{11} & PP_{10} \\
 & PP_{27} & PP_{26} & PP_{25} & PP_{24} & PP_{23} & PP_{22} & PP_{21} & PP_{20} \\
 & PP_{37} & PP_{36} & PP_{35} & PP_{34} & PP_{33} & PP_{32} & PP_{31} & PP_{30} \\
\hline
Y_{15} & Y_{14} & Y_{13} & Y_{12} & Y_{11} & Y_{10} & Y_9 & Y_8 & Y_7 \quad Y_6 \quad Y_5 \quad Y_4 \quad Y_3 \quad Y_2 \quad Y_1 \quad Y_0
\end{array}
$$

Figure 7: Partial Product of 8 bit Multiplier with encoding

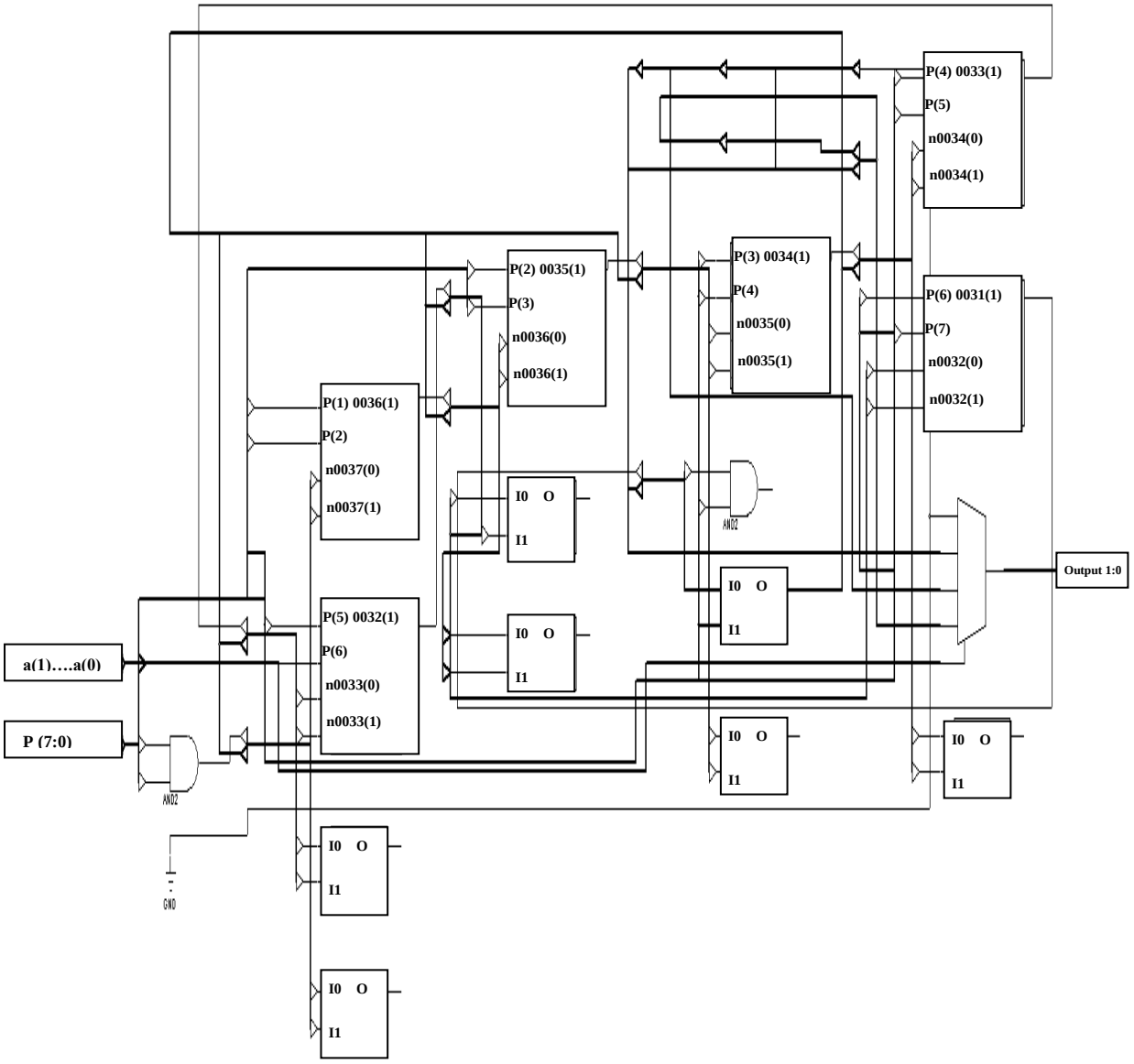

Figure 6: RTL schematic of proposed Encoder

Parameters of Proposed Encoder:
Number of Slices: 19 out of 192 - 9 %
Number of 4 input LUTs: 35 out of 384 - 9%
Number of bonded IOBs : 20 out of 90 - 22%
Total gate delay: 15.118ns (8.260ns logic, 6.858ns route) (54.6% logic, 45.4% route)

Table 3. Comparison with Booth Encoder

Parameter	Booth Radix-4 Encoder	Proposed Encoder
Total Delay	17.431ns	15.118ns
Area	Large	Comparatively Less
Padding of Zero	Required	Not Required
Number of Partial Product row	4	4
Speed	57.36 MHz	66.15 MHz
Hardware Complexity	More due to High Radix & padding	Less

4. CONCLUSION AND FUTURE SCOPE

In this way we can conclude that Vedic multiplier is better than conventional multiplier but after including new encoding algorithm new design is going to be faster. It reduces the complexity associated with Vedic multiplier. Due to reduction in hardware it is going to be less costly than previous structures. Propagation delay in case of multiplier is high but here it is going to be less. We can simply apply this multiplier in various applications such as in implementation of RSA, FFT and in all Digital Signal Processing algorithms. The future work includes is the fixed-width model of this multiplier for low hardware complexity & more reduction in cost and size.

5. ACKNOWLEDGMENTS

The author would like to thanks Prof. B.K.Mohanty & Mr. Dharmendra Kumar for their guidance. The author wish to thank to their family members and close friends for many fruitful discussions.

6. REFERENCES

[1] Swami Bharati Krshna Tirtha, "Vedic Mathematics," Motilal Banarsidass Publishers, Delhi, 1965.

[2] Vedic Mathematics [Online]. Available: http://www. hinduism.co.za/ vedic.htm

[3] Prabir Saha, Arindham Banerajee, Partha Battacharyya, Anup Dhandapat,"High speed design of complex multiplier using Vedic Mathematics", Proceeding of the 2011 IEEE student technology symposium,IIT kharagpur, jan.2011.

[4] Sumit Vaidya & Deepak Dandekar, "Delay-Power performance comparison of Multipliers in VLSI circuit design", International journal of Computer Networks & Communications, July 2010.

[5] Pushpalata Verma, "Design of 4*4 bit Multiplier using EDA Tool", Vol 48 International journal of Computer Application,2012

[6] Harpreet singh Dhillon & Abhijiit Mitra, " A Digital Multiplier Architecture using Urdhava Tiryakbhyam Sutra of Vedic Mathematics" IEEE Conference Proceeding,2008

[7] Nidhi Mittal, Abhijeet Kumar, "Hardware implementation of FFT using vertically and crosswise Algorithm"vol 35 International journal of Computer Application, 2011.

[8] Mr. Abhishek Gupta, Mr. Utsav Malviya, Prof. Vinod Kapse, "A Novel Approach to Design High Speed Arithmetic Logic Unit Based on Ancient Vedic Multiplication Technique" Vol.2 International Journal of Modern Engineering Research. 2012

[9] P.D. Chidgupkar, Mangesh T. Karad,"The implementation of Vedic Algorithms in Digital Signal Processing" Global J. of Education, Vol.8 2004.

[10] Dr. S.M. Khairnar, Ms. Sheetal Kapade, Mr. Naresh Ghorpade "Vedic Mathematics-The cosmic software for implementation of Fast Algorithms"

[11] Mr. Ashish Raman, Anvesh Kumar and R.K.sarin, "High speed Reconfigurable FFT Design By Vedic Mathematics" Vol.1 Journal of computer science and Engineering, 2010.

[12] K.C.Chang "Digital System design with VHDL and synthesis" IEEE Computer society, Willey Publication.