

Implementation for Minimization of Computation Time of Partial Products for Designing Multipliers using Tabulated Vedic Mathematics

Nashrah Fatima
Research scholar
Department of Electronics &
Communication
NIIST College, Bhopal

Taha Tanveer
Research scholar
Department of Electronics &
Communication
TCST College, Bhopal

Brahmi Shrman
Asst, Professor
Department of Electronics &
Communication
NIIST college, Bhopal

ABSTRACT

Multiplication is the most time consuming process in various signal processing operations like Convolution, Circular Convolution, auto-correlation, and Cross Correlation, Image processing applications such as edge detection, microprocessors arithmetic and logical units etc. The multiplier circuit uses adder logic which reduces speed of operation due to its path propagation delay. The designs used offer tradeoffs between Computational time area, latency and throughput for performing multiplication. In this paper an ancient Indian Vedic mathematics sutras base multiplier have been proposed with Urdhva Tiryakbhyam sutra. In this methodology the length of input sequence used for multiplication varies from two to eight.

Keywords

Multiplier circuit

1. INTRODUCTION

Multiplication is a process which is frequently used kernel operation in a wide variety of graphics, image processing, robotics, and signal processing applications. A microprocessors base system performance is determined by the performance of the multiplier. Addition is the fundamental operation of multiplication, thus adders are the building blocks of multipliers which reduces multiplication speed due to its path propagation delay. Thus the multiplier is usually slowest and area consuming element in the system. Hence optimization of speed and area of the multiplier is most important design issue. This paper is discuss the methodology on the design of multiplication operation base on Vedic mathematics by using Vedic Urdhava-Tiryagbhyam Multiplication Sutra. The use of vedic mathamatics based on the natural principles on which the human mind works in multiplication process. It reduces the typical calculations in conventional mathematics to very simple one. Urdhava-Tiryagbhyam Sutra is one of the effective and efficient multiplication formula which is applicable for all types of multiplication.

It is a common multiplication formula valid for all cases of multiplication, which literally means “Vertically and crosswise”. In this method the calculation of all partial products can be made with the concurrent addition of its partial products. The algorithm can be generalized for $n \times n$ bit number. Since the partial products and their sums are

calculated in parallel, the multiplier is independent of the clock frequency of the processor.

2. LITERATURE REVIEW

Alex Pappachen James, Dinesh S. Kumar, and Arun Ajayan work on Threshold Logic Computing: Memristive CMOS Circuits for Fast Fourier Transform (FFT) and Vedic Multiplication. They design and developed memristive threshold logic gates are used for designing fast Fourier transform (FFT) and multiplication circuits useful for modern microprocessors [1]. Their implementation have lower area requirements and higher power dissipation, and in comparison with other memristive-CMOS threshold logic gates the proposed cell indicate lower area requirements and lower power dissipation [1].

Yogita Bansal, Charu Madhu, Pardeep Kaur work on High Speed Vedic Multiplier Designs on Urdhva Tiryakbhyam sutras. In their work the cmos layout of multiplier and adder cell is design [2]. Aravind E Vijayan, Arlene John, Deepak Sen work on 8-bit Vedic Multipliers for Image Processing Application. Their design implemented on Xilinx Virtex 4 based FPGA board and the performance was evaluated using Xilinx ISim simulator [4]. A comparative analysis has been performed and the proposed architecture was found to be efficient both in terms of area and total propagation delay [4].

3. METHODOLOGY

To illustrate this methodology, let us consider the multiplication of different length of two decimal numbers by Urdhva-Tiryakbhyam method. Here implementation between input sequence of length two to eight is shown. The new proposed method for the calculation of Urdhva Tiryakbhyam method is shown below. In this technique, the numbers to be multiplied let us say b_i and a_i are written on the consecutive sides of the square table. On partitioning the square into rows and columns, each row/column belongs to one of the digit of the two numbers to be multiplied such that every digit of one number has a small square common to the digit of other number. These small squares are further divided into two equal parts by crosswise lines. Now the each digit of one number is multiplied with every digit of second number and two digit products are placed in their corresponding square.

1. When the length of input sequence is Two

Length of input sequence $x(n)$ is $= L = 2$ Length of input sequence $h(n)$ is $= M+1 = 2$

Length of output sequence $y(n)$ is $= L+M+1-1 = 3$

Multiplicand $= [a0, a1]$

Multiplier $= [b0, b1]$

Let us consider the two finite input sequences of length two as 99 X 99 the result is 9801.

	9		9	
9	81		81	
9	81		81	
<hr/>				
	8			
	1	162	81	
<hr/>				
	8	17	10	1
<hr/>				
	9	8	0	1

Figure 1.1: Two bit example

Algorithmically can be stated as

	a0		a1	
b0	a0b0		a1b0	
b1	a0b1		a1b1	
s0	s00	s01	s02	
s1	s10	s11	s12	s13
s2	s20	s21	s22	s23

Figure 1.2: Two bit algorithm

Then their product is evaluated as shown

$s00 = a0b0$

$s01 = a0b1 + a1b0$

$s02 = a1b1$

2. When the length of input sequence is Three

Length of input sequence $x(n)$ is $= L = 3$

Length of input sequence $h(n)$ is $= M+1 = 3$ Length of output sequence $y(n)$ is $= L+M+1-1 = 5$

Multiplicand $= [a0, a1, a2]$

Multiplier $= [b0, b1, b2]$

Let us consider the two finite input sequence is of length three 0 as 999X999 the result is 998001.

	9	9	9	
9	81	81	81	
9	81	81	81	
9	81	81	81	
<hr/>				
s0	81	162	243	162 81
s1	8	17	26	19 10 1
s2	9	9	7	10 0 1
s3	9	9	8	0 0 1

Figure 2.1: Three bit example

Algorithmically can be stated as

a0 a1 a2

b0	b0a0	b0a1	b0a2	
b1	b1a0	b1a1	b1a2	
b2	b2a0	b2a1	b2a2	
s0	s00	s01	s02	s03 s04
s1	s10	s11	s12	s13 s14 s15
s2	s20	s21	s22	s23 s24 s25
s3	s30	s31	s32	s33 s34 s35

Figure 2.2: Three bit algorithm

Then their product is evaluated as shown

$s00 = a0b0$

$s01 = b1a0 + b0a1$

$s02 = b2a0 + b1a1 + b0a2$

$s03 = b2a1 + b1a2$

$s04 = b2a2$

As in the result of row s1, s2, s3 the digits on the both sides of the line are multiplied and added with the carry from the previous step. This generates one of the bits of the result and a carry. This carry is added in the next step and hence the process goes on. If more than one line is there in one step, all the results are added to the previous carry. In each row s1, s2, s3, least significant bit acts as the result bit and all other bits act as carry. For example, if in some intermediate step, we will get 243, then 3 will act as result bit and 24 as the carry.

3. When the length of input sequence is Four

Length of input sequence $x(n)$ is $= L = 4$ Length of input sequence $h(n)$ is $= M+1 = 4$

Length of output sequence $y(n)$ is $= L+M+1-1 = 7$

Multiplicand $= [a0, a1, a2, a3]$

Multiplier $= [b0, b1, b2, b3]$

Let us consider the two finite input sequence is of length three 0 as 9999 X 9999 the result is 99980001.

	9	9	9	9	
9	81	81	81	81	
9	81	81	81	81	
9	81	81	81	81	
9	81	81	81	81	
<hr/>					
s0	81	162	243	324	243 162 81
s1	8	17	26	35	28 19 10 1
s2	9	9	9	7	9 10 0 1
s3	9	9	9	7	10 0 0 1
s4	9	9	9	8	0 0 0 1

Figure 3.1: Four bit example

Algorithmically can be stated as

	a0	a1	a2	a3	
b0	b0a0	b0a1	b0a2	b0a3	
b1	b1a0	b1a1	b1a2	b1a3	
b2	b2a0	b2a1	b2a2	b2a3	
b3	b3a0	b3a1	b3a2	b3a3	
s0	s00	s01	s02	s03	s04 s05 s06
s1	s10	s11	s12	s13	s14 s15 s16 s17

s2	s20	s21	s22	s23	s24	s25	s26	s27
s3	s30	s31	s32	s33	s34	s35	s36	s37
s4	s40	s41	s42	s43	s44	s45	s46	s47

Figure 3.2: Four bit algorithm

Then their product is evaluated as shown

$$s00=b0a0$$

$$s01=b1a0+b0a1$$

$$s02=b2a0+b1a1+b0a2$$

$$s03=b3a0+b2a1+b1a2+b0a3$$

$$s04=b3a1+b2a2+b1a3$$

$$s05=b3a2+b2a3$$

$$s06=b3a3$$

As in the result of row s1, least significant bit of 81,62,43,24,43,62,81 of row s0 acts as the result bit and the other entire bits act as carry. In row s2 least significant bit of 8, 17, 26, 35, 28, 19, 10, 1 of row s1 acts as the

result bit and all other bits act as carry. In row s3 least significant bit of 9, 9, 9,7,9,10,0,1 of row s2 acts as the result bit and all other bits act as carry. In row s4 least significant bit of

9,9,9,7,10,0,0,1 of row s3 acts as the result bit and all other bits act as carry. The final result of multiplication is shown in row s4.

4. When the length of input sequence is Five

Length of input sequence $x(n)$ is $= L = 5$

Length of input sequence $h(n)$ is $= M+1 = 5$ Length of output sequence $y(n)$ is $= L+M+1-1 = 9$

Multiplicand = [a0, a1, a2, a3, a4]

Multiplier = [b0, b1, b2, b3, b4]

Let us consider the two finite input sequence is of length three 0 as 99999 X 99999 the the result is 99997300001.

	9	9	9	9	9					
9	81	81	81	81	81					
9	81	81	81	81	81					
9	81	81	81	81	81					
9	81	81	81	81	81					
9	81	81	81	81	81					
s0	81	162	243	324	400	324	243	162	81	
s1	8	17	26	35	44	30	28	19	10	1
s2	9	9	9	9	7	2	9	10	0	1
s3	9	9	9	9	7	2	10	0	0	1
s4	9	9	9	9	7	3	0	0	0	1

Figure 4.1: Five bit example

Algorithmically can be stated as

	a0	a1	a2	a3	a4					
b0	ab0a0	b0a1	b0a2	b0a3	b0a4					
b1	b1a0	b1a1	b2a2	b2a3	b2a4					
b2	b2a0	b2a1	b2a2	b2a3	b2a4					
b3	b3a0	b3a1	b3a2	b3a3	b3a4					
b4	b4a0	b4a1	b4a2	b4a3	b4a4					
s0	s00	s01	s02	s03	s04	s05	s06	s07	s08	
s1	s10	s11	s12	s13	s14	s15	s16	s17	s18	s19
s2	s20	s21	s22	s23	s24	s25	s26	s27	s28	s29
s3	s30	s31	s32	s33	s34	s35	s36	s37	s38	s39
s4	s40	s41	s42	s43	s44	s45	s46	s47	s48	s49

Figure 4.2: Five bit algorithm

Then their product is evaluated as shown

$$s00=b0a0$$

$$s01=b1a0+b0a1$$

$$s02=b2a0+b1a1+b0a2$$

$$s03=b3a0+b2a1+b1a2+b0a3$$

$$s04=b3a1+b2a2+b1a3$$

$$s05=b3a2+b2a3$$

$$s06=b3a3$$

As in the result of row s1, least significant bit of 81,162,243,324,400,324,243,162,81 of row s0 acts as the result bit and all the other bits act as carry. In row s2 least significant bit of 8,17,26,35,44,30,28,19,10,1 of row s1 acts as the result bit and all the other bits act as carry. In row s3 least significant bit of 9,9,9,9,7,2,9,10,0,1 of row s2 acts as the result bit and all the other bits act as carry. In row s4 least significant bit of 9,9,9,9,7,2,10,0,0,1 of row s3 acts as the result bit and all the other bits act as carry. The final result of multiplication is shown in row s5.

	9	9	9	9	9	9					
9	81	81	81	81	81	81					
9	81	81	81	81	81	81					
9	81	81	81	81	81	81					
9	81	81	81	81	81	81					
9	81	81	81	81	81	81					
9	81	81	81	81	81	81					
s0	81	162	243	324	405	486	405	324	243	162	81
s1	8	17	26	35	44	53	46	37	28	19	10
s2	9	9	9	9	9	7	9	9	9	19	1
s3	9	9	9	9	9	7	9	9	10	9	1
s4	9	9	9	9	9	7	9	10	0	9	1
s5	9	9	9	9	9	7	10	0	0	9	1
s6	9	9	9	9	9	8	0	0	0	9	1

Figure 5.1: Six bit example

Algorithmically can be stated a

	a0	a1	a2	a3	a4	a5					
b0	b0a0	b0a1	b0a2	b0a3	b0a4	b0a5					
b1	b1a0	b1a1	b1a2	b1a3	b1a4	b1a5					
b2	b2a0	b2a1	b2a2	b2a3	b2a4	b2a5					
b3	b3a0	b3a1	b3a2	b3a3	b3a4	b3a5					
b4	b4a0	b4a1	b4a2	b4a3	b4a4	b4a5					
b5	b5a0	b5a1	b5a2	b5a3	b5a4	b5a5					
s0	s00	s01	s02	s03	s04	s05	s06	s07	s08	s09	s10
s1	s10	s11	s12	s13	s14	s15	s16	s17	s18	s19	s110
s2	s20	s21	s22	s23	s24	s25	s26	s27	s28	s29	s210
s3	s30s	31	s32	s33	s34	s35	s36	s37	s38	s39	s310
s4	s40	s41	s42	s43	s44	s45	s46	s47	s48	s49	s410
s5	s50	s51	s52	s53	s54	s55	s56	s57	s58	s59	s510
s6	s60	s61	s62	s63	s64	s65	s66	s67	s68	s69	s610

Figure 5.2: Six bit algorithm

Then their product is evaluated as shown

$$s00=b0a0$$

$$s01=b1a0+b0a1$$

$$s02=b2a0+b1a1+b0a2$$

$$s03=b3a0+b2a1+b1a2+b0a3$$

$$s04=b4a0+b3a1+b2a2+b1a3+b0a4$$

$$s05=b5a0+b4a1+b3a2+b2a3+b1a4+b0a5$$

$$s06=b5a1+b4a2+b3a3+b2a4+b1a5$$

5. When the length of input sequence is six

Length of input sequence $x(n)$ is $= L = 6$ Length of input sequence $h(n)$ is $= M + 1 = 6$

Length of output sequence $y(n)$ is $= L + M + 1 - 1 = 11$

Multiplicand $= [a0, a1, a2, a3, a4]$

Multiplier $= [b0, b1, b2, b3, b4]$

Let us consider the two finite input sequence is of length three 0 as 999999 X 999999 the result is 99999800091.

$$s07=b5a2+b4a3+b3a4+b2a5$$

$$s08=b5a3+b4a3++b3a4$$

$$s09=b5a4+b4a5$$

$$s10=b5a5.$$

Similar calculation will generate the final multiplication result in row s6. Since their partial products and sums are calculated in parallel, it is independent of clock frequency. By the Vedic method, efficient multipliers for N bit multiplier systems can be developed from smaller bit multipliers like 2x2 multipliers. The multipliers and adders

used are developed from basic logic gates. The implementation and simulation was carried out in VHDL and ISim respectively. The first step carried out was the preparation and simulation of the various subsystems required to build the Vedic multiplier.

4. CONCLUSION

In this propose work, the different type multiplication operation use in convolution, circular convolution, cross correlation, auto correlation is possible using Vedic mathematics. Fast computations of multiplication operations of two finite length sequences implemented

with the help of this propose algorithm is possible using VHDL language. The operations bas on Urdhava-Triyagbhyam method of Vedic mathematics reduces the processing time as compare to conventional multiplication process. The algorithm provides a platform to solve DSP operation mentally and easily for small length of sequences.

5. REFERENCES

- [1] Alex Pappachen James, Dinesh S. Kumar, and Arun Ajayan "Threshold Logic Computing: Memristive-CMOS Circuits for Fast Fourier Transform and Vedic Multiplication" IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, 5 Jan 2015 Vol PP issue 99.
- [2] Yogita Bansal, Charu Madhu, Pardeep Kaur "High Speed Vedic Multiplier Designs a Review" Proceedings of 2014 RAECS UIET Panjab University Chandigarh, pp no. 06 – 08 March, 2014.
- [3] Diptendu Kumar Kundu¹, Supriyo Srimani², Saradindu Panda³, Prof. Bansibadan Maji⁴ " Implementation of Optimized High Performance 4x4 Multiplier using Ancient Vedic Sutra in 45 nm Technology" IEEE Second International Conference on Devices, Circuits and Systems (ICDCS) 2014.
- [4] Aravind E Vijayan, Arlene John, Deepak Sen "Efficient Implementation of 8-bit Vedic Multipliers for Image Processing Application" IEEE International Conference on Contemporary Computing and Informatics (IC3I) 2014 pp no. 544-550.
- [5] P.Mehta and D. Gawali, "Conventional versus Vedic mathematical method for Hardware implementation of a multiplier." In Proceedings IEEE international conference on Advances in Computing, Control and Telecommunication Technologies. Trivandrum, Kerela, Dec.28-29 T 2009, pp.640-642