IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
ISSN: 2319 – 4200, ISBN No. : 2319 – 4197 Volume 1, Issue 4 (Nov. - Dec. 2012), PP 21-26
www.iosrjournals.org

Parallel Hardware Implementation of Convolution using Vedic Mathematics

[1]Mrs.Rashmi Rahul Kulkarni
[1](Electronics and Telecommunication, Finolex Academy of Management and Technology/Mumbai University, INDIA)

Abstract : *Convolution is fundamental operation of most of the signal processing systems. It is necessity of time to speed up convolution process at very appreciable extent. Here Direct method of computing the discrete linear convolution of finite length sequences is used. The approach is easy to learn because of the similarities to computing the multiplication of two numbers by a pencil and paper calculation. Multipliers are basic building blocks of convolver. Since it dominates most of the execution time, for optimizing the speed, 4×4 bit Vedic multipliers based on Urdhva Tiryagbhyam sutra are used. Convolver has delay of 17.996 ns when implemented on 90 nm process technology FPGA. It also provides necessary modularity, expandability, and regularity to form different convolutions for any number of bits. The coding is done in VHDL (Very High Speed Integrated Circuits Hardware Description Language) for the FPGA , as it is being increasingly used for variety of computationally intensive applications. Simulation and synthesis is done using Xilinx 9.2i.*

Keywords - *Convolution , FPGA ,Vedic Mathematics*

I. INTRODUCTION

Convolution is the most important and fundamental concept in signal processing and analysis. Many of researchers have been trying to improve performance parameters of convolution system. One of the factor in performance evaluation of any system is speed. The core computing process in convolution is always a multiplication routine. Faster addition and multiplication are of extreme importance in DSP. Therefore, engineers are constantly looking for boosting performance parameters of it using new algorithms and hardware. After comparative study of different multipliers, Urdhva Tiryagbhyam sutra is shown to be an efficient multiplication algorithm [3][4] .

In Ref.[1],convolution is carried out by serial processing. They used only one 4×4 bit Vedic multiplier based on Urdhva Tiryagbhyam sutra. Though hardware is less, delay is more as sixteen multiplications are carried out one by one using only single multiplier.

In this paper, convolution of two finite length sequences is computed using Direct method. This method is similar to the multiplication of two decimal numbers, this similarity that makes this method easy to learn and quick to compute [2]. As Vedic multiplier is high speed multiplier among existing multipliers [3],[4], Urdhva Tiryagbhyam algorithm from Vedic mathematics is used for 4×4 bit multiplication and to improve speed parallel processing approach is used.

All required possible adders are studied. All these adders are synthesized using Xilinx9.2i. There delays and areas are compared. Adders which have highest speed and comparatively less area occupied, are selected for implementing convolution.

II. BRIEF LITERATURE SURVEY

In Ref.[1],convolution is carried out by serial processing. They used only one 4×4 bit Vedic multiplier based on Urdhva Tiryagbhyam sutra. Though hardware is less, delay is more as sixteen multiplications are carried out one by one using only single multiplier. Direct method for calculating the linear convolution sum of two finite length sequences is easy to learn and perform. The approach is easy to learn because of the similarities to computing the multiplication of two numbers by a pencil and paper calculation. FPGA implementation is future work [2]. In parallel FIR filter algorithm, the preprocessing, postprocessing and subfilter matrices can be calculated easily with Matlab. Then, Matlab can be used to automatically generate Verilog code for the hardware implementation of this algorithm [5].But in automatically generated code there is no control on architecture level. ROM look up tables can be used to implement the computational modules. Multipliers can be realized using memory based approach. Multiplication of two n bit input variables can be performed by ROM table of size 2 with power 2n entries [6]. But this approach is not efficient in area point of view.CRT algorithm minimizes multiplication operation at cost of increase in addition operations [7]. Parallel implementation improves speed[8]. The sutras in Vedic mathematics are easy to understand, easy to apply and easy to remember. Vedic maths is helpful to software developers as it is more scientific than the normal system of mathematics [9].

III. CONVOLUTION

The behavior of a linear, time-invariant discrete-time system with input signal x[n] and output signal y[n] is described by the convolution sum .Standard equation for convolution ,if x[n] is an N point signal running from 0 to N-1, and h[n] is an M point signal running from 0 to M-1, the convolution of the two: y[n] = x[n] * h[n], is an N+M-1 point signal running from 0 to N+M-2, given by ;

$$y[i] = \sum_{j=0}^{M-1} h[j]\,x[i-j] \qquad (1)$$

Equation(1) is called the convolution sum. It allows each point in the output signal to be calculated independently of all other points in the output signal. The index, *i*, determines which sample in the output signal is being calculated, and therefore corresponds to the left-right position of the convolution machine. Method used here to carry out discrete convolution is Direct method.

3.1. Direct Method

This method for discrete convolution is best introduced by a basic example. For this example, let f(n) equal the finite length sequence (10 11 9 8) and g(n) equal the finite length sequence (15 14 12 13). The linear convolution of f(n) and g (n) is y(n) = f(n) * g (n). This can be solved by several methods, resulting in the sequence y(n) = {150 305 419 498 363 213 104}. This approach for calculating the convolution sum is set up like multiplication where the convolution of f(n) and g (n) is performed as shown in fig. 1.

			g(n) :	15	14	12	13
			f(n):	10	11	9	8
				120	112	96	104
		+	135	126	108	117	
		165	154	132	143		
	150	140	130	120			
y[n]=	150	305	419	498	363	213	104
	y[6]	y[5]	y[4]	y[3]	y[2]	y[1]	y[0]

Figure 1. Convolution by Direct Method

As seen in Fig.(1) computation of the convolution sum, the approach is similar to a pencil and paper multiplication calculation, except carries are not performed out of a column[2].

To get convolution of two sequences, where each sequence consist of 4 samples, sixteen partial products are calculated and afterwards they are added to get convolution sequence y[n]. In this paper, Partial products are calculated by using vedic multiplier based on Urdhva Tiryagbhyam algorithm. Here to minimize hardware, width of each input sample is restricted to 4 bit. Hence maximum possible input sample value would be $(1111)_2$ or $(15)_{10}$ or $(F)_h$. Multiplier required is 4×4 bit. Each multiplier gives 8 bit long partial product. Convolution outputs y[6] and y[0] are direct Partial products. While y[5] obtained after addition of intermediate partial products ,like wise remaining outputs need to be obtained. For addition of required partial products different adders are designed and synthesized, fastest one is selected for implementation.

IV. PROPOSED IMPLEMENTATION

Let two discrete length sequences are x[n] and h[n].Where x[n] = {a3 a2 a1 a0 } and h[n] = { b3 b2 b1 b0 } are convolved. As each sample is four bit long ,each partial product is eight bit long e.g. a0b0, a3b0, a3b3 all are eight bit long. y[n] = x[n] * h[n], in a way as mentioned above. Procedure is rearranged as shown in fig. 2.

x[n]				a3	a2	a1	a0
h[n]				b3	b2	b1	b0
				a3b0	a2b0	a1b0	a0b0
			a3b1	a2b1	a1b1	a0b1	
		a3b2	a2b2	a1b2	a0b2		
	a3b3	a2b3	a1b3	a0b3			
Y[n]	Y6	Y5	Y4	Y3	Y2	Y1	Y0

Figure 2. Convolution of x[n] and h[n]

4 × 4 convolution is implemented in order to keep cost low. This can be extended for N × N convolution.

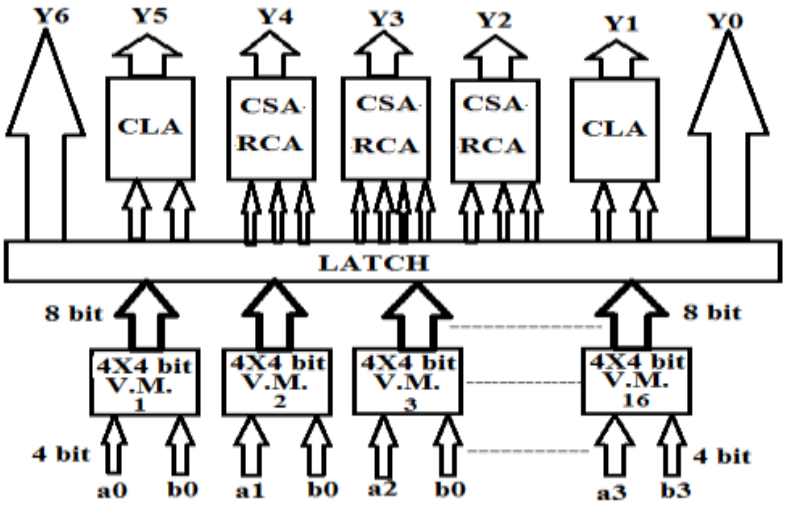

Figure 3. Block Diagram for convolution

As shown in fig. 3., 4 bit long samples are applied to 4X4 bit vedic multipliers (V.M.). Output of each vedic multiplier is 8 bit long partial product. Vedic multiplier uses Urdhva Tiryagbhyam algorithm for multiplication. In parallel processing, to generate sixteen partial products, sixteen vedic multipliers are used to boost speed. To perform further operation of addition, all outputs are latched. To produce Y1 and Y5 carry look ahead adders (CLA) are used and to generate partial products Y2, Y3 and Y4 carry save adders with last stage of ripple carry adder (CSA- RCA) are used. Maximum possible length of Y0 and Y6 is 8 bit , while of Y1 and Y5 is 9 bit. Y2, Y3, Y4 are at the most 10 bit long. The design is built in VHDL and implemented on an FPGA.

4.1. MULTIPLIER

Multiplication was implemented generally with a sequence of additions. There exist many algorithms proposed to perform multiplication, each offering different advantages and having trade off in terms of delay, circuit complexity, area occupied on chip and power consumption. For multiplication algorithms performing in DSP applications, latency and throughput are two major concerns from delay perspective.

A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Selection of speedy multiplier leads to boosting speed of system.

4.2. VEDIC MULTIPLIER

Vedic mathematics is part of four Vedas (books of wisdom). It is part of Sthapatya-Veda (book on civil engineering and architecture), which is an upa-veda (supplement) of Atharva Veda. It gives explanation of several mathematical terms including arithmetic, geometry (plane, co-ordinate), trigonometry, quadratic equations, factorization and even calculus.

His Holiness Jagadguru Shankaracharya Bharati Krishna Teerthaji Maharaja (1884-1960) comprised all this work together and gave its mathematical explanation while discussing it for various applications.

The work presented here, makes use of Vedic Mathematics. "Urdhva Tiryagbhyam Sutra" or "Vertically and Crosswise Algorithm" of Vedic mathematics for multiplication is used to develop digital multiplier architecture. This looks quite similar to the popular array multiplier architecture. This Sutra shows how to handle multiplication of a larger number (N x N, of N bits each) by breaking it into smaller numbers of size (N/2 = n, say) and these smaller numbers can again be broken into smaller numbers (n/2 each) till we reach multiplicand size of (2 x 2) .Thus, simplifying the whole multiplication process.

Let number1 = $(10)_2$ and number2 = $(10)_2$, Their multiplication using Urdhva Tiryagbhyam is shown fig. 4.

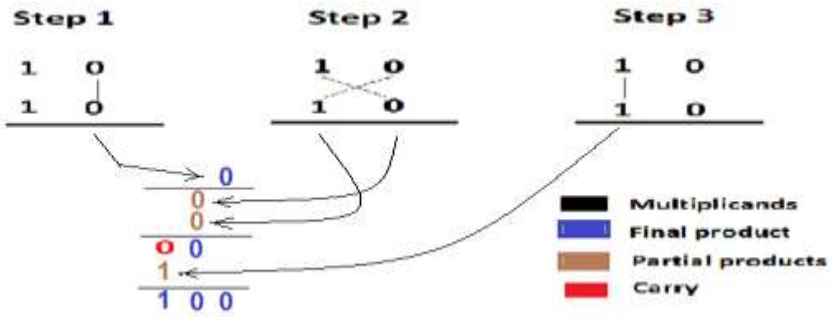

Figure 4. 2 X 2 bit multiplication using Urdhva Tiryagbhyam

4 X 4 bit multiplication can be achieved using 2 X 2 bit. Let Number1= a0 =$(1101)_2$ and Number2= b0 =$(1010)_2$. Split each number into two groups. a0 = {(g1=11)(g0=01)}, b0 = {(g3=10)(g2=10)}.Then stick diagram for this multiplication is as shown in fig. 5.

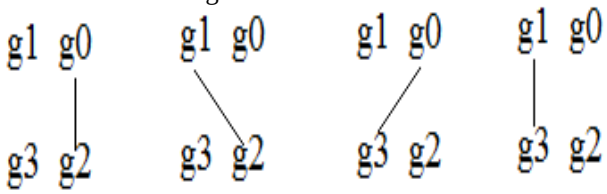

Figure 5. Stick Diagram

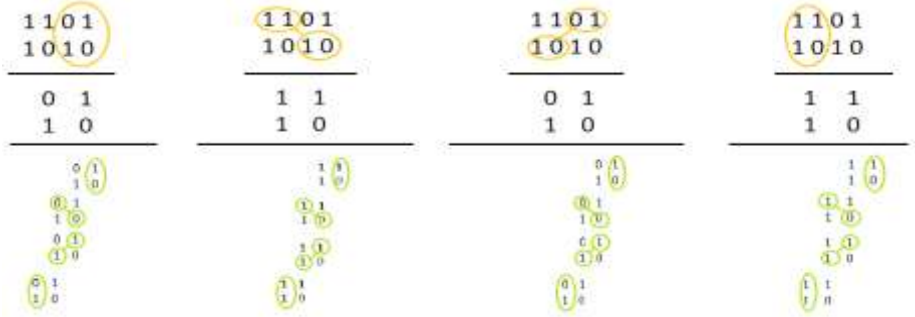

Figure 6. 4 X 4 bit multiplication by Urdhava Tiryagbhyam

4 X 4 bit multiplication carried out using 2 X 2 bit multiplication blocks is shown in fig. 6. All 2 X 2 bit multiplications are carried out simultaneously. Hence speed boosting is achieved.

4.3. SELECTION OF ADDERS

Choice of speedy adder is done by implementing and comparing . Ripple carry adder(RCA) ,Carry look ahead adder(CLA), carry save adder with last stage built by ripple carry adder(CSA-RCA) and carry save adder with last stage built by carry look ahead adder(CSA-CLA) ,for family Spartan 3E.

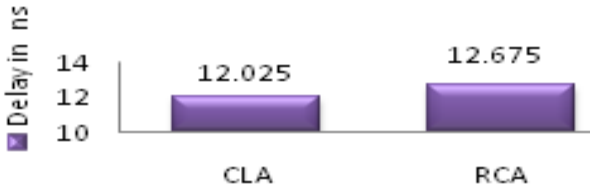

Figure 7. Delay comparison of 8 bit adders for addition of two numbers

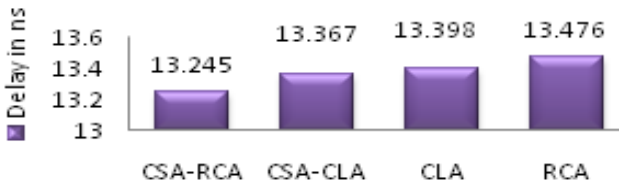

Figure 8. Delay comparison among different adders for addition of three eight bit numbers

Figure 9. Delay comparison among different adders for addition of four eight bit numbers

As per results of comparisons ,for two 8 bit number's addition carry look ahead adder (CLA) is selected. For three and four ,8 bit number's addition carry save adder with last stage built by ripple carry adder is selected (CSA-RCA).

V. DESIGN VERIFICATION

Verification is carried out by ISE simulator. Simulator output is shown in figure 1. Following two input arrays with samples are used for verification:

First input array:[a3 a2 a1 a0] : $15_{10}(F)_h$ $14_{10}(E)_h$ $12_{10}(C)_h$ $13_{10}(D)_h$

Second input array:[b3 b2 b1 b0] : $A_{10}(10)_h$ $B_{10}(11)_h$ $9_{10}(9)_h$ $8_{10}(8)_h$

Expected Convolved output : [conv6 conv5 conv4 conv3 conv2 conv1 conv0] : [96 131 199 1FC 16B D5 68]

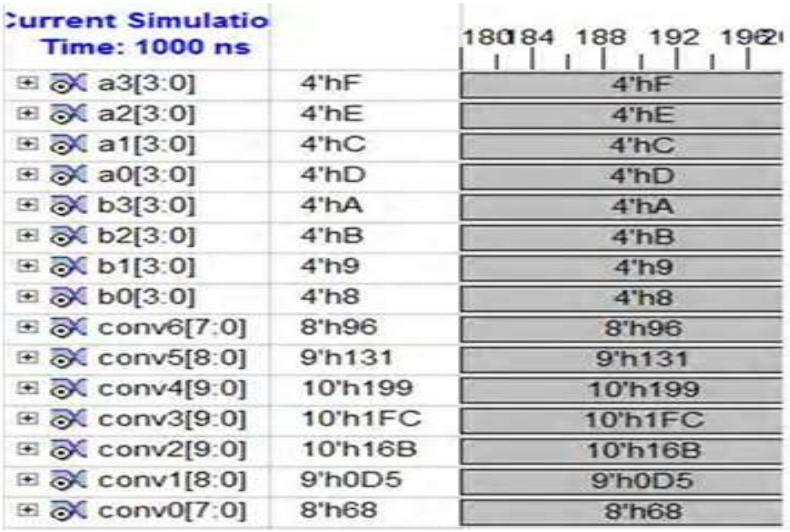

Figure 10. Simulation results of convolver using Vedic mathematics

VI. EVALUATION OF PROPOSED DESIGN

Table 1.and fig.11. shows delay improvement of proposed circuit over the circuit implemented in [1].Functionality is tested using Spartan 3E device family(90 nm process technology) with speed grade -5.

TABLE I.

TABLE II. DELAY COMPARISION FOR PROPOSED DESIGN

Parameters	Reference[1]	Parallel Implementation of Convolver
Delay	183.292 ns	17.996 ns

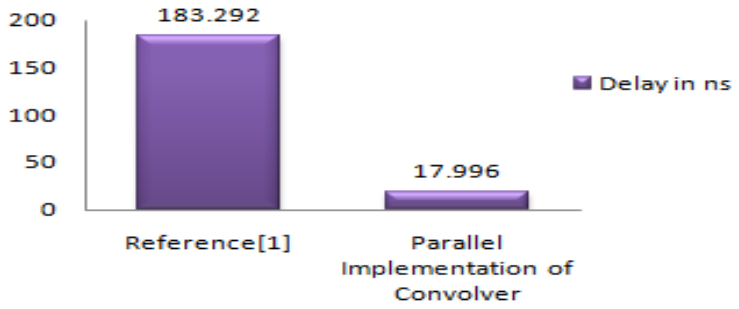

Figure 11. Delay comparison of proposed design

VII. CONCLUSION

This paper, presents speedy implementation of discrete linear convolution. This particular model has the advantage of being fine tuned for any signal processing application. To accurately analyze proposed system, design is coded using the VHDL hardware description language and synthesized it for FPGA products using ISE. The proposed circuit takes about 17ns to complete on 90 nm process technology devices. Similarly, the presented concept can be extended on an NXN case.

Key element behind increasing speed of convolver is multiplier design based on Urdhva Tiryagbhyam sutra of Vedic mathematics and parallel implementation of hardware. Vedic Mathematics is like drops picked up from the ocean of Vedas.

REFERENCES

[1] Rashmi Lomte and Bhaskar P.C., "High Speed Convolution and Deconvolution using Urdhva Triyagbhyam " ,2011 IEEE Computer Society Annual Symposium on VLSI ,p.323 ,July 2011.
[2] John W. Pierre, "A Novel Method for Calculating the Convolution Sum of Two Finite Length Sequences", IEEE transaction on education, VOL.39, NO. 1, 1996.
[3] Honey Tiwari, Ganzorig ankhuyag, Chan Mo Kim,Yong Beom Cho,"Multiplier design based on ancient Indian Vedic Mathematics",IEEE,2008 International Soc Design Conference.
[4] Sumit Vaidya, Deepak Dandekar ,"Delay power performance comparison of multipliers in VLSI circuit design", International Journal of Computer Networks & Communications, Vol 2, No. 4,July 2010.
[5] Chao Cheng , Keshab K. Parhi "Hardware Efficient Fast Parallel FIR Filter Structures Based on Iterated Short Convolution" IEEE, and, IEEE transaction on circuits and systems, VOL. 51, NO. 8, 2004.
[6] Thomas Oelsner ,"Implementation of Data Convolution Algorithms in FPGAs" http://www.quicklogic.com/images/appnote18.pdf
[7] Abraham H. Diaz, Domingo Rodriguez ,"One Dimentional Cyclic Convolution Algorithms With Minimal Multiplicative Complexity", ICASSP.
[8] Mountassar Maamoun,"VLSI Design for High Speed Image Computing Using Fast Convolution- Based Discrete Wavelet Transform", Proceedings of the world Congress on Engineering Vol 1,July 2009.
[9] Pandit Ramnandan Shastri, "Vedic Mathematics", Arihant Publications, p.V.