VOL. 10, NO. 1, JANUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

HIGH SPEED APPLICATION SPECIFIC INTEGRATED CIRCUIT (ASIC) DESIGN OF CONVOLUTION AND RELATED FUNCTIONS USING VEDIC MULTIPLIER

Sai Vignesh K. and Balamurugan S. and Marimuthu R.
School of Electrical Engineering, VIT University, Vellore, India
E-Mail: svkrish24@gmail.com

ABSTRACT

ASIC implementation of convolution plays a pivotal role in digital signal processing and analysis. One of the factors in performance evaluation of any system is its speed. In this paper, direct method of computing the discrete linear convolution of finite length sequences was used in order to speed up the process. Convolution related functions such as cross-correlation and auto-correlation were also implemented. Multipliers are the building blocks of a convolution system. Since they dominate most of the execution time, for optimizing the speed, 4×4 bit Vedic multipliers based on 'Urdhva-Tiryagbhyam' (UT) sutra was used. The Verilog HDL coding for the proposed design was done and implemented using Cadence RTL complier with standard 90nm CMOS technology and the results were compared with other conventional methodologies.

Keywords: ASIC, convolution, Vedic multiplier, correlation, parallel processing.

1. INTRODUCTION

Convolution is the fundamental operation in most signal processing systems. It is the necessity of time to speed up the convolution process at an appreciable extent. Faster addition and multiplication are extremely important in DSP. Therefore, engineers constantly look to boost its performance parameters using new algorithms and hardware.

Convolution is carried out by serial processing in [1] using only one 4×4 bit Vedic multiplier based on 'Urdhva-Tiryagbhyam' sutra. Since it uses only one multiplier it consumes less hardware, but the delay was more. In this paper, convolution of two finite length sequences is computed using the direct method. This method is similar to the multiplication of two decimal numbers, hence easier to comprehend [2].

Vedic mathematics is a part of four Vedas. It is a part of Sthapatya-Veda (book on civil engineering and architecture), which is a supplement of Atharva Veda. Explanation for several modern mathematical terms such as arithmetic, geometry, trigonometry, quadratic equations, factorization and even calculus are covered in it. The Vedic mathematics system is based on 16 Vedic sutras (formulae) which describes natural ways of solving a whole range of mathematical problems. In this paper, Urdhva-Tiryagbhyam formula is used in the structural level for faster multiplication.

2. CONVOLUTION

The behavior of a linear, time-invariant discrete-time system with an input signal x [n] and an output signal y [n] is described by the convolution sum. It is donated as y[n] = x[n] * h[n], where h[n] is a harmonic function. 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 is an N+M-1 point signal running from 0 to N+M-2. It is given by:

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

Equation (1) is called the convolution sum. Each point in the output signal is calculated independently of all other points in the output signal. The index, i, determines which sample in the output signal is being calculated. The method used in this paper to carry out discrete convolution is called the direct method.

2.1 Direct method

This method for discrete convolution can be best introduced by a basic example. Let g (n) equal the finite length sequence (1 2 3 4) and h (n) also be equal the finite length sequence (1 2 3 4). The linear convolution of g (n) and h (n) is y (n). This can be solved by several methods, resulting in the sequence y (n) = {1 4 10 16 25 24 16}. Convolution of g (n) and h (n) using direct method is performed as shown in Figure-1.

g (n)			1	2	3	4	
h (n)	x		1	2	3	4	
			4	8	12	16	
+		3	6	9	12		
	2	4	6	8			
1	2	3	4				
y (n)	1	4	10	20	25	24	16
	y[6]	y[5]	y[4]	y[3]	y[2]	y[1]	y[0]

Figure-1. An example for direct method of convolution computation.

From the Figure-1, we can easily understand that computation of the convolution sum, is similar to the multiplication of two decimal numbers.

The two input sequences g (n) and h (n) consist of 4 samples each and hence, sixteen partial products are calculated and afterwards they are added to get convolution sequence y[n]. In this paper, the partial products are calculated by using the Vedic multiplier [4]. In order to minimize hardware, the width of each input sample has been restricted to 4 bits. Therefore, the maximum possible input sample value would be $(1111)_2$ or $(15)_{10}$ or $(F)_{h.}$

The multiplier required for this system is 4×4 bit. The output of each multiplier will be 8 bits long. Convolution outputs y [6] and y [0] are direct partial products. Just as y [5] is obtained after addition of intermediate partial products, the remaining outputs are obtained. For addition of required partial products different adders are designed and synthesized. The fastest one is selected for implementation.

2.2 Cross-correlation and auto-correlation

By definition, cross-correlation is a measure of similarity of two waveforms as a function of a time-lag applied to one of them. Computation of cross-correlation in this paper is done similar to that of convolution using the direct method. The cross - correlation of two signals g (n) and h (n) means to convolve g (n) with the reverse of h (n). For example, if g(n) is equal to the finite length sequence (1 2 3 4) and h(n) is also equal the finite length sequence (1 2 3 4), the cross-correlation of the two signals is obtained by convolving g(n) = (1 2 3 4) and h(n) = (4 3 2 1).

Auto-correlation is the cross-correlation of a signal with itself. It is defined as the similarity between observations as a function of the time lag between them. Here only one sequence is given as input. Auto-correlation of g(n) means to convolve g(n) with the reverse of g(n) itself. Again, this is done by direct method in this paper.

3. PROPOSED ARCHITECTURE

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

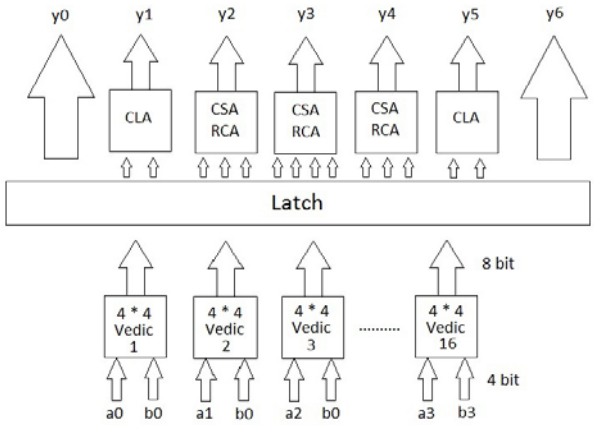

Figure-2. Block diagram of the proposed architecture of convolution.

As shown in Figures 2, 4-bit long input samples are applied to 4x4 Vedic multipliers (V.M.). The output of each Vedic multiplier is 8-bit long. The Vedic multiplier uses Urdhva-Tiryagbhyam sutra for multiplication. In order to generate sixteen partial products, sixteen Vedic multipliers are used to boost the speed. To perform further operation of addition, all outputs are latched using a d-latch. To produce y1 and y5, carry look-ahead adders (CLA) are used. To generate partial products y2, y3 and y4, carry save adders with a last stage of ripple carry adder (CSA- RCA) is used. The design is built in Verilog HDL and implemented using Cadence RTL complier with standard 90nm CMOS technology.

3.1 Vedic multiplier

The potentiality of 'Vedic Mathematics' was reported by 'Sri Bharati Krsna Thirthaji Maharaja', in the form of the Vedic Sutras (formulae) [3]. He explored the mathematical potentials from Vedic primers and showed that the mathematical operations can be carried out mentally to produce fast answers using these sutras.
A few methods of multiplication proposed in Vedic Mathematics are:

a) Urdhva-Tiryagbhyam: Vertically and Crosswise
b) Nikhilam navatashcharamam Dashatah: All from nine and last from ten
c) Anurupyena: Proportionately Vinculum

In this paper, we have used Urdhva-Tiryagbhyam formula for designing the convolution system.

3.1.1 Multiplier architecture

The hardware architecture of 2 x 2 and 4 x 4 Vedic multiplier modules are displayed in the sections below. Here partial product generation and additions are done concurrently. Hence, the architecture is well adapted to parallel processing. This in turn reduces the delay, which is the primary motivation behind this work.

3.1.2 Vedic multiplier for 2x2 bit module

Let us consider two 2-bit numbers A and B where A = a1a0 and B = b1b0 as shown in Figure-3. Firstly, the least significant bits are multiplied which gives the least significant bit of the final product. Then, the L.S.B. of the multiplicand is multiplied with the next higher bit of the multiplier and added with the product of L.S.B. of multiplier and next higher bit of the multiplicand. The sum gives second bit of the final product. The carry is added with the partial product obtained by multiplying the most significant bits. The sum is the third corresponding bit and carry becomes the fourth bit of the final product.

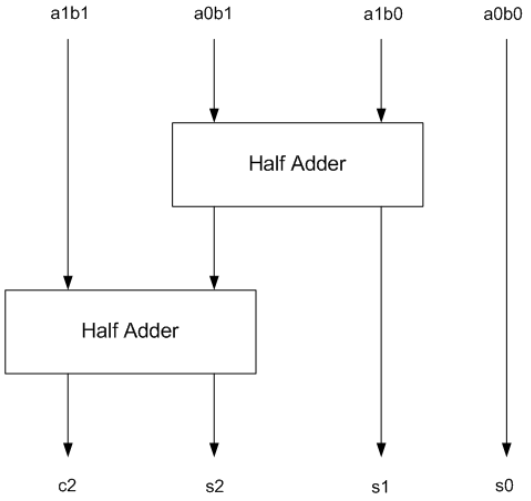

Figure-3. Block diagram of 2x2 Vedic multiplier.

Four input AND gates and two half-adders are used to implement this, as displayed in the block diagram in Figure-3. It is found that the hardware architecture of 2x2 bit Vedic multiplier is similar to the hardware architecture of 2x2 bit conventional Array Multiplier. Hence it is understood that multiplication of 2-bit binary numbers by Vedic method does not make significant impact in multiplier's efficiency.

3.1.3 Vedic multiplier for 4x4 bit module

The 4x4 bit Vedic multiplier module is implemented using four 2x2 bit Vedic multipliers as discussed. Let's take two inputs, say A= a3 a2 a1 a0 and B= b3 b2 b1 b0. The output line for the multiplication result would be - s7 s6 s5 s4 s3 s2 s1 s0. Let's divide A and B into two parts, say (a3a2) and (a1a0) for A and (b3b2) and (b1b0) for B. Each is a 2x2 bit Vedic multiplier. First 2x2 bit multiplier inputs are (a1a0) and (b1b0). The last block is 2x2 bit multiplier with inputs (a3a2) and (b3b2). The middle one shows two 2x2 bit multiplier with inputs (a3a2) and (b1b0) and (a1a0) and (b3b2).

The final 8-bit result will be (s7 s6 s5 s4 s3 s2 s1 s0). For better understanding, the block diagram of 4x4 bit Vedic multiplier is shown in Figure-4.

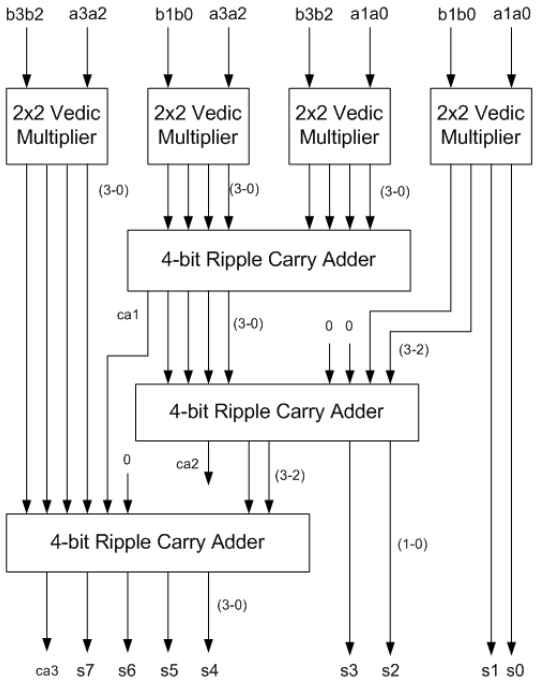

Figure-4. Block diagram of 4x4 Vedic multiplier.

In order to get final product (s7 s6 s5 s4 s3 s2 s1 s0), four 2x2 bit Vedic multiplier (Figure-4) and three 4-bit Ripple-Carry Adders (RCA) are used. The proposed Vedic multiplier can be used to reduce delay. Earlier Vedic multipliers were based on array multiplier structures. This new architecture, as proposed in paper [4] is efficient in terms of speed. The arrangements of RCAs shown in Figure-4, reduces delay.

3.2 Selection of Adders

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), were implemented and compared using Xilinx ISE Design Suite 12.1. As per the comparison results, for addition of two 8-bit numbers, carry look ahead adder (CLA) is selected. For three and four 8-bit numbers addition, carry save adder with last stage built by ripple carry adder is selected (CSA-RCA).

4. RESULTS

4x4 convolution with each input 4-bit long was implemented in Xilinx ISE Design Suite 12.1 and Cadence RTL compiler.

Input given: a0=1, a1=2, a2=3, a3=4, b0=1, b1=2, b2=3, b3=4

Output observed: y0=1, y1=4, y2=10, y3=20, y4=25, y5=24, y6=16

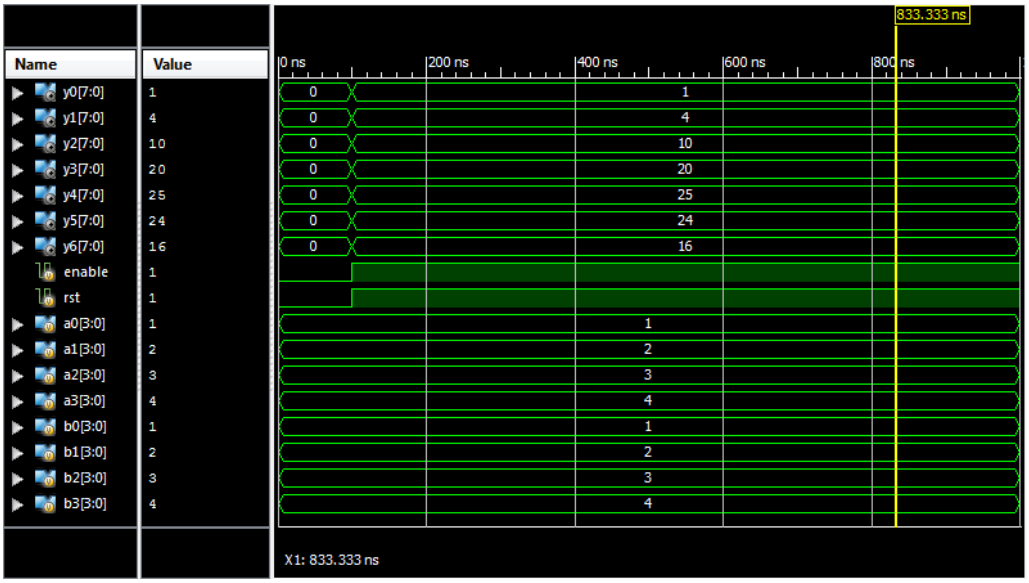

Figure-5. Output observed for convolution with the proposed architecture in Xilinx ISE design suite 12.1.

Table-1. Convolution characteristics with array, booth and proposed Vedic multipliers as observed in Cadence RTL compiler.

	Delay (ps)	**Power** (nW)	**Power-Delay product**
Convolution with array multiplier	3656	386400.71	1412680995.76
Convolution with booth multiplier	3634	261211.56	949242809.04
Convolution with proposed architecture	2757	318595.27	878369916.39

Table-2. Cross-correlation and Auto-correlation with proposed architecture as observed in Cadence RTL compiler.

	Delay (ps)	**Power** (nW)
Cross-correlation with Vedic multiplier	2728	291976.923
Auto-correlation with Vedic multiplier	2670	290444.758

From Table-1, we can observe that the delay for convolution with proposed architecture has been reduced by ~23.13%. The input data was taken in a regular fashion for experimental purpose.

Also, one can note from Table-1 that the power-delay product has been substantially reduced as compared to that of convolution with array and booth multipliers. In electronics, the power-delay product is a Figure of merit correlated with the energy efficiency of a logic gate or logic family. Also known as switching energy, it is the product of power consumption and the input–output delay.

It has the dimension of energy, and measures the energy consumed per switching event.

The delay and power characteristics of cross-correlation and auto-correlation with the proposed model have been tabulated in Table-2.

5. CONCLUSIONS

In this paper, we have design and implement a high-speed implementation of discrete linear convolution and its related applications such as cross-correlation and auto-correlation. The model presented here has the advantage of being used for any signal processing application. To accurately analyse the proposed system, the design was coded using the Verilog HDL and implemented using Cadence RTL complier with standard

90nm CMOS technology. The results were compared with convolution using array and booth multipliers and the proposed architecture with Vedic multiplier proved to be the best in terms of delay and power-delay product. Further, it can be envisaged from the above discussion that the Vedic multiplier proved to be the most critical element in improving the speed of the circuit to compute convolution and related functions.

REFERENCES

[1] Rashmi Lomte and Bhaskar P.C. 2011. High Speed Convolution and Deconvolution using Urdhva Triyagbhyam. IEEE Computer Society Annual Symposium on VLSI. p. 323.

[2] John W. Pierre 1996. A Novel Method for Calculating the Convolution Sum of Two Finite Length Sequences. IEEE transaction on education. 39(1).

[3] J.S.S.B.K.T. Maharaja. 2001. Vedic Mathematics, Motilal Banarsidass Publishers Pvt. Ltd, Delhi.

[4] Poornima M, Shivaraj Kumar Patil, Shivukumar, Shridhar K P, Sanjay H. 2013. Implementation of Multiplier using Vedic Algorithm. International Journal of Innovative Technology and Exploring Engineering. 2(6).

[5] Honey Tiwari, Ganzorig ankhuyag, Chan Mo Kim, Yong Beom Cho. 2008. Multiplier design based on ancient Indian Vedic Mathematics. IEEE International Soc Design Conference.

[6] Sumit Vaidya, Deepak Dandekar. 2010. Delay power performance comparison of multipliers in VLSI circuit design, International Journal of Computer Networks and Communications. 2(4).

[7] Chao Cheng, Keshab K. Parhi. 2004. Hardware Efficient Fast Parallel FIR Filter Structures Based on Iterated Short Convolution. IEEE transaction on circuits and systems. 51(8).

[8] Mountassar Maamoun. 2009. VLSI Design for High Speed Image Computing Using Fast Convolution-Based Discrete Wavelet Transform. Proceedings of the world Congress on Engineering. Vol. 1.

[9] Thapliyal H. and Srinivas M.B. 2004. High Speed Efficient NxN Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics. Transactions on Engineering, Computing and Technology. Vol.2.

[10] Harpreet Singh Dhillon and Abhijit Mitra. 2008. A Reduced - Bit Multiplication Algorithm for Digital Arithmetic. World Academy of Science, Engineering and Technology. Vol. 2.

[11] Parth Mehta and Dhanashri Gawali. 2009. Conventional versus Vedic mathematics method for Hardware implementation of a multiplier. International conference on Advances in Computing, Control, and Telecommunication Technologies. pp. 640-642.

[12] Ramalatha, M. Dayalan, K D Dharani, P Priya and S Deoborah. 2009. High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques. International Conference on Advances in Computational Tools for Engineering Applications. pp. 600-603.

[13] P. Saha, A.Banerjee, A. Dandapat, P. Bhattacharyya. 2011. ASIC design of a high speed low power circuit for factorial calculation using ancient Vedic mathematics. Microelectronics Journal. 42 pp. 1343-1352.

[14] Rashmi Rahul Kulkarni. 2012. Parallel Hardware Implementation of Convolution using Vedic Mathematics. IOSR Journal of VLSI and Signal Processing. 1(4): 21-26.

[15] M. Nagaraju, R. Surya Prakash, B. Vijay Bhaskar. 2013. High Speed ASIC Design of Complex Multiplier Using Vedic Mathematics. International Journal of Engineering Research and Applications. 3(1): 1079-1084.