

Convolution Using Urdhva Tiryagbhyam

Prof. Rashmi Rahul Kulkarni

Department of Electronics and Telecommunication
Finolex Academy of Management and Technology

Abstract-Multiplication is fundamental operation of most of the signal processing systems. Convolution process needs multiplication. To speed up convolution process at very appreciable extent. Multiplier used needs to be speedy. Here Direct method of computing the discrete linear convolution of finite length sequences is used. 4×4 bit Vedic multipliers based on Urdhva Tiryagbhyam sutra also called as “Vertically and cross wise.” are used. Here two different approaches of convolution are compared. For Serial circuit it gives saving in area occupied while for parallel approach, it gives speed improvement 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.

Keywords-Urdhva Tiryagbhyam, FPGA, VHDL

I. INTRODUCTION

Convolution is a mathematical way of combining two signals to form a third signal. It is the single most important technique in Digital Signal Processing. Entire signals and systems and communication depends and revolves around convolution. There are several places where it is crucial. Convolution is similar to cross-correlation. For discrete real valued signals, they differ only in a time reversal in one of the signals. It has applications that include probability, statistics, computer vision, natural language processing, image and signal processing, engineering, and differential equations.

Many of researchers have been trying to improve performance parameters of convolution system. One of the important factor in performance evaluation of any system is speed. The core computing process in convolution is always a multiplication routine. Increase in speed of multiplier directly improves speed of convolution circuit. Different multiplier algorithms are studied and compared, among different multiplication algorithms, 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.

Comparison of required possible is done area and delay wise. Adders which have high speed and comparatively less area occupied, are selected for implementing convolution.

II. IDENTIFY, RESEARCH AND COLLECT IDEA

In Ref.[1],convolution is carried out by serial processing. Direct method is used to find convolution and deconvolution. This design approach efficiently and accurately speeds up computation without compromising with area.

[2]Direct method for calculating the linear convolution sum of two finite length sequences is easy to learn and perform is explained here. 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.

[5]This paper presents an iterated short convolution (ISC) algorithm, based on the mixed radix algorithm and fast convolution algorithm. This ISC-based linear convolution structure is transposed to obtain a new hardware efficient fast parallel finite-impulse response (FIR) filter structure, which saves a large amount of hardware cost, especially when the length of the FIR filter is large. For example, for a 576-tap filter, the proposed structure saves 17% to 42% of the multiplications, 17% to 44% of the delay elements, and 3% to 27% of the additions, of those of prior fast parallel structures, when the level of parallelism varies from 6 to 72. Their regular structures also facilitate automatic hardware implementation of parallel FIR filters. They used automatically generated verilog code using MATLAB, But in

automatically generated code there is no control on architecture level.

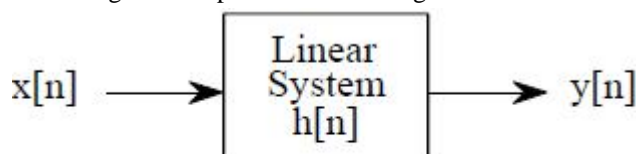
High performance Digital Signal Processing chips have been widely employed to solve signal processing problems. Many of these signal processing solutions can be implemented in a Field Programmable Gate Array (FPGA) instead of a DSP chip. This is possible because the gate densities available in FPGAs have increased rapidly within the last few years and now allow fairly sophisticated DSP algorithms to be implemented within a single chip [6].

[7] In digital signal processing, the design of fast and computationally efficient algorithms has been a major focus of research activity. The objective, in most cases, is the design of algorithms and their respective implementation in a manner so as to perform the required computations in the least amount of time. In order to achieve this goal, parallel processing has also received a lot attention in the research community.

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

Convolution is a formal mathematical operation, just as multiplication, addition, and integration. Addition takes two numbers and produces a third number, while convolution takes two signals and produces a third signal .



$$x[n] * h[n] = y[n]$$

Figure 1. Notation for convolution

Figure 1. shows the notation when convolution is used with linear systems. An input signal, $x[n]$, enters a linear system with an impulse response, $h[n]$, resulting in an output signal, $y[n]$. In equation form: $x[n] * h[n] = y[n]$

we can write the 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:

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

This equation 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.

Consider any discrete time signal $x[n]$. The signal $x[n]$ can be represented as sum of many delayed/advanced and scaled Unit Impulse Signals.

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \delta[n-k]$$

The above expression corresponds to the representation of any arbitrary sequence as a linear combination of shifted Unit Impulses $\delta[n-k]$ which are scaled by $x[n]$. Now if we knew the response of a system for a Unit Impulse Function, we can obtain the response of any arbitrary input.

Direct Method of Convolution:

Consider example, let $f(n)$ equal the finite length sequence (4 5 6) and $g(n)$ equal the finite length sequence (4 4 3 2). 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 $y(n) = (16 36 56 47 28 12)$. This new approach for calculating the convolution sum is set up like multiplication where the convolution of $f(n)$ and $g(n)$ is performed as follows:

$$\begin{array}{r}
 g(n) : \quad 4 \ 4 \ 3 \ 2 \\
 f(n) : \quad \quad * \ 4 \ 5 \ 6 \\
 \hline
 \quad \quad \quad 24 \ 24 \ 18 \ 12 \\
 \quad \quad 20 \ 20 \ 15 \ 10 \\
 \quad 16 \ 16 \ 12 \ 8 \\
 \hline
 y(n) : \quad 16 \ 36 \ 56 \ 47 \ 28 \ 12.
 \end{array}$$

Figure 2. Direct Method of convolution [2]

As seen in the above 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. Above method is used in this project.

By observing above procedure to find out convolution, if one use efficient multipliers and efficient adders, then convolution implemented would be efficient.

IV. MULTIPLIER

A multiplier is one of the key hardware blocks in most digital and high performance systems such as digital signal processors and microprocessors etc. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following- high speed, low power consumption, regularity of layout and hence less area or even combination of them in multiplier. Thus making them suitable for various high speed, low power, and compact VLSI implementations. However area and speed are two conflicting constraints. So improving speed results always in larger areas. Efforts are made here to find trade off solution among the both of them.

V. VEDIC MATHEMATICS

Arithmetic is used by almost everyone, for tasks ranging from simple day to day work like counting to advanced science and business calculations. As a result, the need for a faster and efficient Arithmetic Unit in computers has been a topic of interest over decades. The work presented in this paper, makes use of Vedic Mathematics.

Urdhva Tiryagbhyam sutra for multiplication:

The multiplier is based on an algorithm Urdhva Tiryagbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics. Urdhva Tiryagbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means “Vertically and crosswise”. It is based on a novel concept through which the generation of all partial products can be done and then, concurrent addition of these partial products is achieved. Thus parallelism in generation of partial products and their summation is obtained using Urdhava Tiryagbhyam. The algorithm can be generalized for n x n number. Since the partial products and their sums are calculated in parallel, the multiplier is independent of the clock, if required in circuit.

Now let us see how this algorithm can be used for binary numbers. For example (1101 * 1010) as shown in Figure 3. Firstly, least significant bits are multiplied which gives the least significant bit of the product (vertical). Then, the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added with the product of LSB of multiplier and next higher bit of the multiplicand (crosswise). The sum gives second bit of the product and the carry is added

in the output of next stage sum obtained by the crosswise and vertical multiplication and addition of three bits of the two numbers from least significant position. Next, all the four bits are processed with crosswise multiplication and addition to give the sum and carry. The sum is the corresponding bit of the product and the carry is again added to the next stage multiplication and addition of three bits except the LSB.

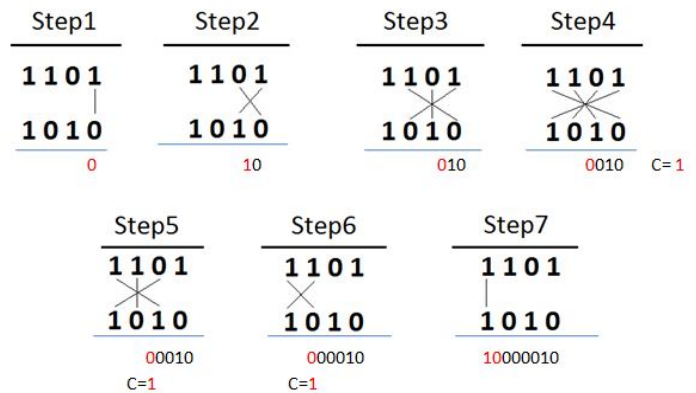


Figure 3: Multiplication by Urdhva Tiryagbhyam algorithm

In this paper above, 4x4 bit multiplication is simplified into 4 , 2x2 bit multiplications that can be performed in parallel. This reduces the number of stages of logic and thus reduces the delay of the multiplier.

VI. IMPLEMENTATION

In this paper, two different approaches of implementation are considered and compared. First is Serial method and second is parallel method approach. For both methods Vedic multiplier is used in circuit. Let us consider two sequences [a b c d] and [e f g h], whose convolution is to be found. Let each number is 4 bit long. Let ‘a’ is constituted by bits a3,a2,a1 and a0. Same is applied for all numbers.

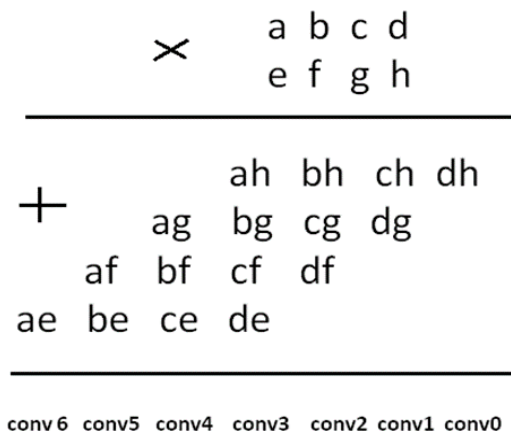


Figure4. shows block diagram of serial hardware

implementation of convolution. In this implementation each time to calculate partial products, only one multiplier is used. Selection of input is done by multiplexer switches. Addition is done by Carry Look Ahead adder (CLA) and for multiple stages addition, Carry save Adder with last stage of ripple carry adder is used(CSA-RCA). Area utilized for this implementation is less as compared to area utilized for parallel implementation

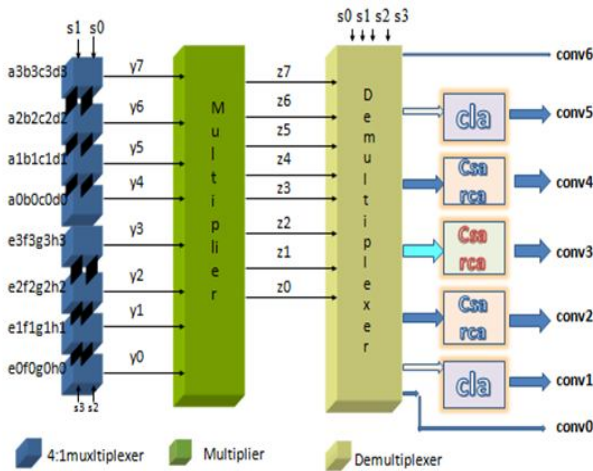


Figure .4. Block Diagram of Serial Hardware Implementation

For parallel hardware implementation above mentioned block diagram is modified. Here multiplexer blocks are not needed. As sixteen product terms are needed, multiplier block is get duplicated for sixteen times. Hence area consumption is more but delay is less. More speed we get in parallel hardware implementation than serial. Figure 5. shows block diagram of parallel hardware implementation.

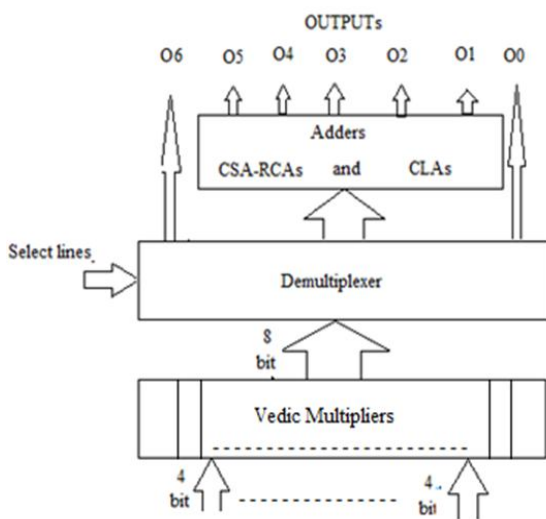


Figure 5: Parallel Hardware Implementation of Convolution

VII. RESULTS

4x4 Vedic Multiplier when simulated for Spartan 3E 11.676 ns delay is obtained.

For 4 stage 8 bit addition, different adders are implemented on Spartan 3E. Following Table.1. explains area and speedwise comparison among these adders. RCA in table indicates Ripple Carry Adder while CLA indicates Carry Look Ahead adder.

Adders	Area			Delay in ns
	Slices	LUTs	IOBs	
Carry Save Adder with RCA	25	43	42	14.338
Carry Save Adder with CLA	26	46	42	14.377
Carry Look Ahead Adder	26	46	42	14.349
Ripple Carry Adder	26	46	42	14.466

Comparison Parameters	Serial Implementation	Parallel Implementation
Delay in ns	159.715	17.9
Slices Utilized	212	340

Parallel implementation utilizes more area than serial implementation while parallel implementation circuit is faster than serial one.

VI. CONCLUSION

This paper presents both possible implementation styles of discrete linear convolution. This particular model has the advantage of being fine tuned for signal processing. Application, where area saving is important, one can go with serial implementation model with compromising speed. But when speed factor is important prefer parallel convolution circuit.

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"
- [7] Abraham H. Diaz, Domingo Rodriguez, "One Dimensional 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.