FPGA Implementation of Convolution using Wallace Tree Multiplier

Madhuraj
Electronics & Communication
Galgotias University
Gr.Noida, India

Prince Kumar Pandey
Electronics & Communication
Galgotias University
Gr.Noida, India

Mayank Kumar
(Asst.Prof.)
Electronics & Communication
Galgotias University
Gr.Noida, India

***Abstract*— This paper is presenting a method to reduce the convolution processing time using hardware computing and implementations of discrete linear convolution of two finite length sequences (NXN). This, implementation method is realized by simplifying the convolution building blocks. The purpose of this research is to speed up the computation. The proposed implementation uses a modified hierarchical design approach, which is efficient and accurate to speeds up the computation; reduces power, hardware resources, . Simulation and comparison to different design approaches show that the circuit uses only 5mw and is faster than what is implemented in [5] and in [1]. In addition, the presented circuit uses less power consumption and has a delay of approx 6ns from input to output. It also provides the necessary modularity, expandability, and regularity to form different convolutions for any number of bits.**

Keywords— Convolution, VHDL, implementations, FPGA, Design and Implementation for discrete linear convolution

I. INTRODUCTION

"Convolution" is an operation involving two functions that turns out to be very useful in many applications. Convolution provides the mathematical framework for Digital signal Processor (DSP). Convolution is the most important and fundamental concept in signal processing and analysis. Filtering of signals is very important in order to determine which one to accept and which one to reject, and all of that is done by convolution. Many image processing operations such as scaling and rotation require re-sampling or convolution filtering for each pixel in the image [3]. Digital images can be modified (through convolution) by neighborhood operations; these operations go beyond point wise operations, and include smoothing, sharpening, and edge detection [2]. Convolution has many applications which have great significance in discrete signal processing. Current high-performance image transformation hardware such as that used in video production ("special effects") devices performs these calculations, but often with compromises in quality, especially for high scale factors. Most such devices (and

indeed most software codes used in computer graphics rendering) have a fixed, sometimes quite small (5 or less) maximum filter width and simply ignore the aliasing and lost information that results. However, as video resolution increases (e.g. high-definition television) and other interactive applications such as graphic arts and page composition become more demanding, the quality of a significantly scaled or transformed picture must remain high[3]. Today, most DSPs suffer from limitations in available address space, or the ability to interface with surrounding systems. The use of high speed FPGAs, together with DSPs, can often increase the system bandwidth, by providing additional functionality to the general purpose DSPs [5].In this paper, a novel method for computing the linear convolution of two finite length sequences is presented. A 4x4 convolution circuit can be instantiated for larger ones. Many approaches have been attempted to reduce the convolution processing time using hardware and software algorithms. But they are restricted to specific applications [6]. Convolution has so many applications which are used in various systems, so it is important to speed up the convolution process. For this we have to change some hardware and software algorithms. Which improve the speed of convolution process and decrease the power dissipation and area used.

This paper is organized as follows. Section II investigates the related convolution algorithm implementation. In section III, circuit implementations are presented. Section IV presents the verification of the proposed design. In section V, evaluation and comparison of the design are presented.

II. BACKGROUND AND REALATED WORK

The main assumption of the consistency principle and the mutual correspondence principle between continuous and digital transformations is that the signal is represented discretely through shift sampling and reconstruction. An image convolution is a filtering step in which an image is the input and a computed image is the output, with each sample of the output image calculated by individually weighting and then constructively and/or destructively summing the samples from some neighborhood of the input image [7]. Mathematically, a convolution is defined as the integral over all space of one function at x times another function at u-x.

The integration is taken over the variable x (which may be a 1D or 3D variable), typically from minus infinity to infinity overall the dimensions. So the convolution is a function of a new variable u, as shown in the following equations. The cross in a circle is used to indicate the convolution operation.

$$C(u) = f(x) \otimes g(x) = \int_{space} f(x)\, g(u - x)\, dx$$

$$= g(x) \otimes f(x) = \int_{space} g(x)\, f(u - x)\, dx$$

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]

Fig1. Example of direct method of convolution

In Fig1. 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.

[2] Presents a direct method of reducing convolution processing time using hardware computing and implementations of discrete linear convolution of two finite length sequences (NXN). It was using array multiplier for multiplication. [12]Present a 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.

Many of researchers have been trying to improve performance parameters of convolution system. One of the factors 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, Wallace tree multiplier is shown to be an efficient multiplication algorithm.

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.

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation. The common multiplication method is "add and shift" algorithm. In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. To achieve speed improvements Wallace Tree algorithm can be used to reduce the number of sequential adding stages. Multiplier in convolution process is one of the most important part which is responsible for the speed of convolution process. To achieve speed improvements Wallace Tree algorithm can be used to reduce the number of sequential adding stages .so we use Wallace tree multiplier here. In Wallace tree architecture, all the bits of all of the partial products in each column are added together by a set of counters in parallel without propagating any carries. Another set of counters then reduces this new matrix and so on, until a two-row matrix is generated. The most common counter used is the 3:2 counters which is a Full Adder. The final result are added using usually carry propagate adder. The advantage of Wallace tree is speed because the addition of partial products is now O (logN). [2] Introduce a method for calculating the linear convolution sum of two finite length sequences that is easy to learn and perform. [11] Presented a design for fast convolve for CDMA signals. This is based on avoiding complex operations such as FFT based convolves. They used substitution of the FFT for a Walsh which reduces the operations three times because it uses only real additions but it requires more hardware like counters, and RAM blocks which increases activity factor. Using image processing functions such as convolution filtering, high performance can be achieved by exploiting parallelism and minimizing hardware cost, but different filter widths and thus potentially different hardware structures are needed for different applications. It is therefore difficult to make a fixed parallel structure efficient. In an application involving spatial scaling of images, for example, a larger filter kernel would be required for large scale factors, a small one for modest scaling. It would be expensive to implement the entire largest desired filter kernel, and wasteful for small scale factors [3]. It is proven that convolution can check all the phase shifts in one step. This is usually done by using the known FFT-based convolution [11]. Each FFT (or IFFT) requires NlogN complex multiplications and NlogN complex additions. Therefore, some algorithm require approximately 3N (logN) +N complex multiplications and 3N (logN) +N additions [1]. Implementing the algorithm in parallel hardware will speed

up the process but the implementation itself is very complex and requires a huge silicon area.

The Digital Convolution is summarized as: first Flip (reverse) one of the digital functions, second Shift it along the time axis by one sample. Third, multiply the corresponding values of the two digital functions. Fourth, sum the products from step 3 to get one point of the Digital Convolution. And finally repeat steps 1-4 to obtain the digital convolution at all times that the functions overlap. 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 ;

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

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. In direct method we take the two discrete finite length sequences and lines the columns up like regular multiplication but rather than carrying the number over to the next column he writes it down in the same column. For example let's say that we are given two discrete finite length sequences x[n] and h[n] where x[n] = {A1 A2 A3 A4} and h[n] = {B1 B2 B3 B4} are convolved, y[n] =x[n]*h[n], in a way that is similar to regular multiplication as shown below:

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			
Y6	Y5	Y4	Y3	Y2	Y1	Y0

As we see direct method of convolution process above which is same as multiplication but the outputs y0, y1, y2, y3, y4, y5 and y6 are add of partial products which is just above no carry is shifted to left and added with other.

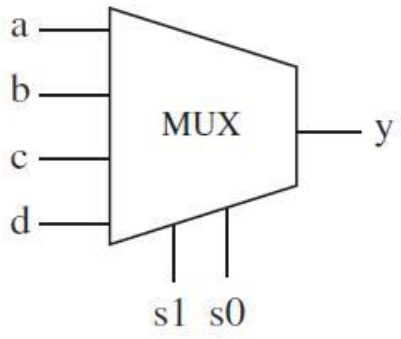

Fig 2: 4*1 multiplexer

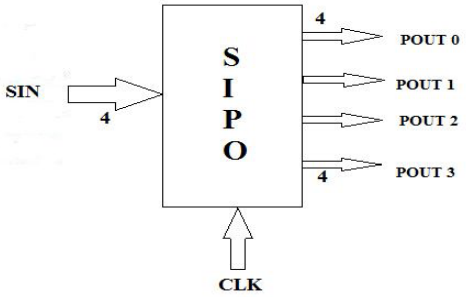

Fig 3: SIPO Block Diagram

The blocks which we are using for convolution process are:
1. Multiplexer
2. Serial input parallel output shift register
3. Wallace tree multiplier
4. Registers

1. Multiplexer

A multiplexer is a device that selects between a numbers of input signals. In its simplest form, a multiplexer will have two signals inputs, one control input and one output. A multiplexer is a device which selects any one of the input from 2^n inputs, and directed to output depending on n-select lines. The higher order multiplexer can be implemented using lower order multiplexers. Sometime it referred as "Multiplexor" or Mux. Block Diagram of Mux is shown in Fig 2.

2. Serial in parallel out Shift Register (SIPO)

A serial in parallel-out shift register converted data from serial format to parallel format. If four data bits are shifted in by four clock pulses via a single wire at data-in, the data becomes available simultaneously on the four outputs Qa to Qd after the fourth clock pulse. The practical application of the serial-in/parallel-out shift register is to convert data from serial format on a single wire to parallel format on multiple wires. The block diagram of serial in parallel out shift register is shown below in which SIN is serial input, CLK is clock input and POUT0, POUT1, POUT2, POUT3 are parallel outputs, after fourth clock pulse. Block Diagram of SIPO is shown in Fig 3.

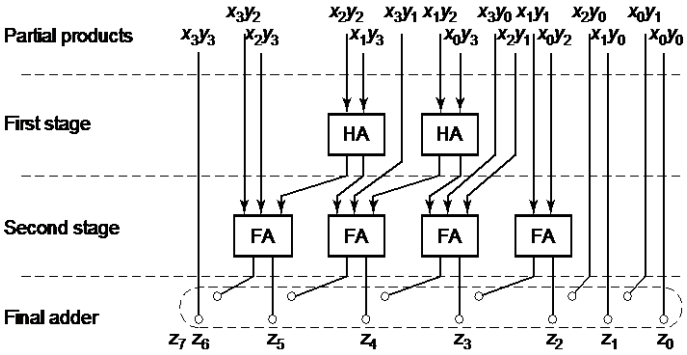

Fig 4: Wallace-Tree Multiplier

3. 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.

Multiplier in convolution process is one of the most important part which is responsible for the speed of convolution process. To achieve speed improvements Wallace Tree algorithm can be used to reduce the number of sequential adding stages .so we use Wallace tree multiplier here. In Wallace tree architecture, all the bits of the partial products in each column are added together by a set of counters in parallel without propagating any carries. Another set of counters then reduces this new matrix and so on, until a two-row matrix is generated. The final results are added using usually carry propagate adder. The advantage of Wallace tree is speed because the addition of partial products is now O (logN). A block diagram of 4 bit Wallace Tree multiplier is shown in below. As seen from the block diagram partial products are added in Wallace tree block. The result of these additions is the final product bits and sum and carry bits which are added in the final fast adder (CRA). In Fig 4 Wallace tree multiplier is shown.

4. Registers

A register is a simple, one-bit memory device, either a flip-flop or a latch. A flip-flop is an edge-triggered memory device. A latch is a level-sensitive memory device. Its basic function is to fold information within a digital system. However a register may also have additional capabilities associated with it. It may have combinational gates that perform certain data processing tasks. The Fig 5 Shows the Block Diagram of register.

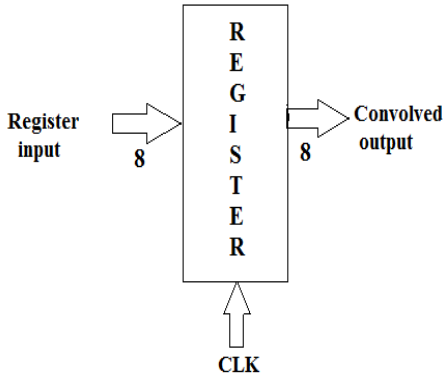

Fig5. Register Block Diagram

III. PROPOSED IMPLEMNTATION CIRCUIT

NXN was selected, and the implementation for 4x4 was prepared in order to have short convolutions that will lead to the lowest implementation cost [10] .The circuit deals with two signals having N values each. They selected N=4 in this implementations. Which consider the two numbers like two arrays having four locations each to store values. Each array is fed into a quadruple 4X1 Mux separately. Hence they can have each signal value up to 4 bit. The basic concept of convolution is to flip, multiply and add. Now for two signals of four values each, we have to flip (invert one of the signals) multiply and then add the values. Block diagram of overall convolution Process is shown in which: The flipping of the values is done by selection of the 4X1 Multiplexer. The output data from the multiplexer is applied to the serial input parallel output block the data will be converted serial to parallel. The output of the SIPO block is connected to the Wallace tree multiplier .Then the data will be stored in the registers. The fig. above shows the overall convolution process.

In this we did some change in our multiplier. The multiplier which we are using is Wallace tree multiplier in this we have to change the carry propagation routine because in convolution process's direct method, carry is not shifted and added towards left all the column are added together separately. so we got 7 outputs. Then all the results are stored in registers. In Fig 6 overall convolution process is shown.

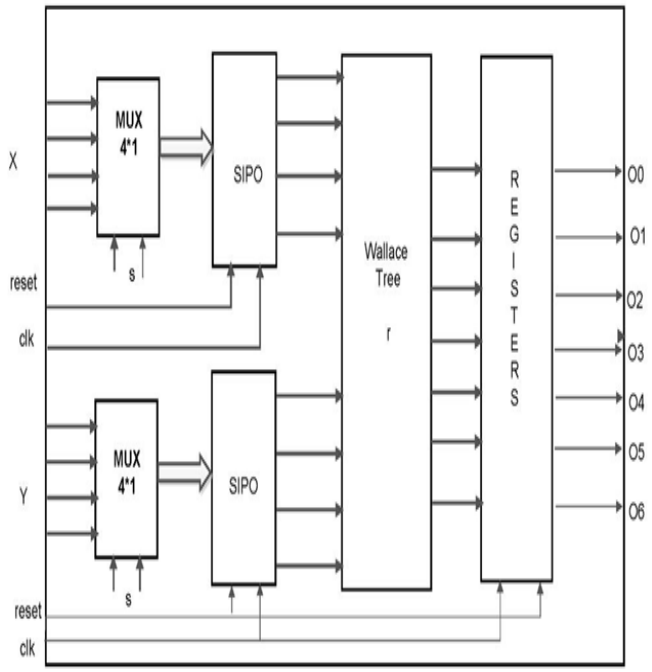

Fig6. Block diagram of overall convolution Process

IV. VERIFICATION OF THE PROPOSED DESIGN

In this we did some change in our multiplier. The multiplier which we are using is, Wallace tree multiplier in this we have to change the carry propagation routine because in convolution process's direct method, carry is not shifted and added towards left all the column are added together separately .so we got 7 outputs. Then all the results are stored in registers. We simulate it then the simulation result which came is shown in Fig 7. The RTL Schematic View of convolution process is shown in Fig 8 in which we can see the components which we used is how connected to each other.

V. EVALUATION AND COMPARISON OF THE DESIGN

Table 1 shows Advanced HDL Synthesis Report in which no. of registers, multiplexers and xors are shown.

Table 1 Advanced HDL Synthesis Report

#Registers	21
Flip-flops	21
#Multiplexers	2
1-bit 4 to 1 multiplexer	2
#Xors	13
1-bit xor2	13

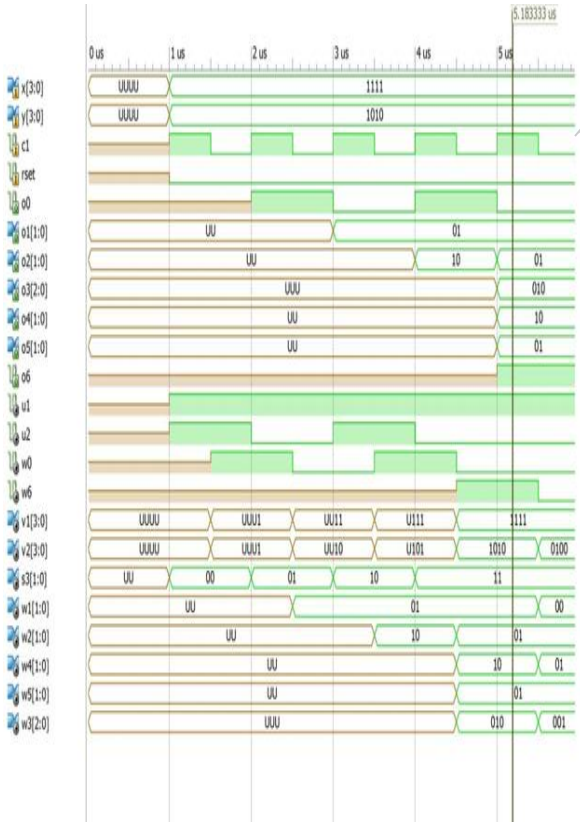

Fig 7: Final simulation result of convolution

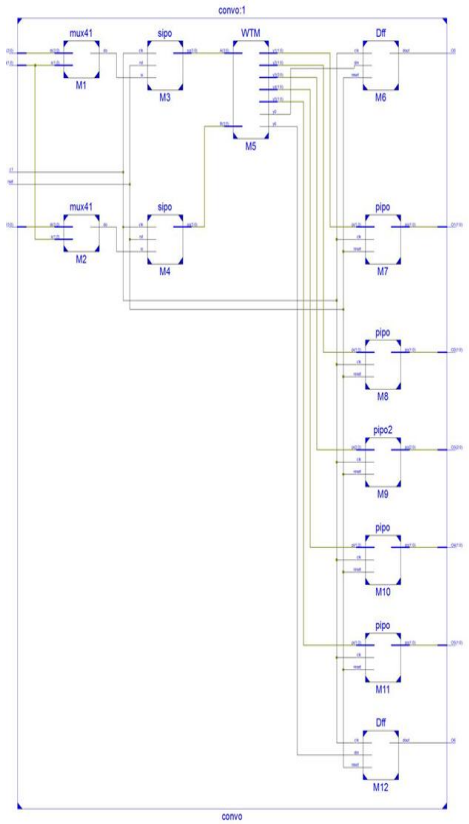

Fig 8. RTL Schematic View of convolution process

Table 2 FPGA results

Logic cells	Numbers
LUT 2	4
LUT4	4
LUT6	9
Flip Flop/Latches	21
Clock Buffers	1
BUFGP	1
IO Buffers	24
IBUF	11
OBUF	13

Table 3 Timing Summary

Minimum period	5.234ns(Maximum Frequency: 191.073MHz)
Minimum input arrival time before clock	2.765ns
Maximum output required time after clock	3.597ns

Table 2 shows the FPGA result and Table 3 shows timing summary of simulation.

CONCLUSION

In this paper, we presented an optimized implementation of discrete linear convolution. This particular model has the advantage to speed up the convolution process. This implementation has the advantage of being optimized based on operation, speed, power and area. To accurately analyze our proposed system, we have coded our design using the VHDL and have synthesized it for FPGA products using ISE, Modelsim and DC compiler for other processor usage. Second, we implemented an illustrative example 4X4 convolver. Similarly, the presented concept can be extended on an NXN case. The functionality of the convolver was tested and verified successfully on a XILINIX SE FPGA and design compiler. the delay come through this is lesser than before.

REFERENCES

[1] 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.
[2] K. Mohammad, "Efficient FPGA implementation of convolution" IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA - October 2009.
[3] R. G. Shoup, "Parameterized convolution filtering in a field programmable gate array," in selected papers from the Oxford 1993 international workshop on field programmable logic and applications on More FPGAs. Oxford, United Kingdom: Abingdon EE&CS Books, 1994, pp. 274–280.
[4] Iván Rodríguez, "Parallel Cyclic Convolution Based on Recursive Formulations of Block Pseudocirculant MatricesMarvi Teixeira", IEEE, transaction on signal processing, 2008.
[5] Thomas Oelsner ,"Implementation of Data Convolution Algorithms in FPGAs" , QuickLogic Europe http://www.quicklogic.com/images/appnote18.pdf
[6] Chao Cheng , Keshab K. Parhi ,"Low-Cost Fast VLSI Algorithm for Discrete Fourier Transform", IEEE,. IEEE transaction on circuits and systems, VOL. 54, 2007.
[7] J. I. Guo, C. M. Liu, and C. W. Jen, "The efficient memory-based VLSI array designs for DFT and DCT," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 37, no. 10, 1992, pp. 723–733.
[8] T. S. Chang, J. I. Guo, and C. W. Jen, "Hardware-efficient DFT designs with cyclic convolution and sub expression sharing",IEEE Trans. Circuits Syst. II, Analog Digital Signal Process., vol. 47, no. 9, 2000, pp. 886– 892.
[9] C. Cheng and K. K. Parhi, "Hardware efficient fast DCT based on novel cyclic convolution structures", IEEE Trans. Signal Process., vol. 54, no.11, 2007, pp. 4419–4434.
[10] 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.
[11] Mrs. Rashmi Rahul Kulkarni, "Parallel Hardware Implementation of Convolution using Vedic Mathematics" 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.