

Effectuation of Efficient Convolution and Deconvolution

T.Venkata Sowjanya

M.Tech Student
Department of ECE
Guntur Engineering College,
Guntur, Andhra Pradesh, India.

C.Srinivasa Rao

Assistant Professor,
Department of ECE
Guntur Engineering College,
Guntur, Andhra Pradesh, India.

Abstract:

Convolution and deconvolution place an important role in order to know more information about particular signals and determination of required data about unknown signals, hence necessitate in many DSP applications. In convolution and deconvolution multiplication and division are important elements. Traditional multiplication and division methods reduce the performance of the convolution and deconvolution. There by this paper proposes the advance Vedic methods for multiplication and division like Urdhwa TiryakbyamSutra and division by this fast and accuracy operation can be achieved. The coding is done by using Verilog HDL, simulated using Model-sim and synthesized using XILINX.

1.INTRODUCTION

Convolution provides the mathematical framework for DSP. It is the single most important technique in Digital Signal Processing. Convolution is a mathematical way of combining two signals to form a third signal. Using the strategy of impulse decomposition, systems are described by a signal called the impulse response. In signal processing, the impulse response, or impulse response function (IRF), of a dynamic system is its output when presented with a brief input signal, called an impulse.

More generally, an impulse response refers to the reaction of any dynamic system in response to some external change. It has applications that include statistics, computer vision, image and signal processing, electrical engineering, and differential equations.

One of the most important concepts in Fourier theory, and in crystallography, is that of a convolution. Convolutions arise in many guises, as will be shown below. Because of a mathematical property of the Fourier transform, referred to as the convolution theorem, it is convenient to carry out calculations involving convolutions.

The convolution of f and g is written $f * g$, using an asterisk or star. It is defined as the integral of the product of the two functions after one is reversed and shifted. As such, it is a particular kind of integral transform:

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau \\ = \int_{-\infty}^{\infty} f(t - \tau)g(\tau) d\tau.$$

While the symbol t is used above, it need not represent the time domain. But in that context, the convolution formula can be described as a weighted average of the function $f(\tau)$ at the moment t where the weighting is given by $g(-\tau)$ simply shifted by amount t . As t changes, the weighting function emphasizes different parts of the input function.

More generally, if f and g are complex-valued functions on \mathbf{R}^d , then their convolution may be defined as the integral:

$$(f * g)(x) = \int_{\mathbf{R}^d} f(y)g(x - y) dy = \int_{\mathbf{R}^d} f(x - y)g(y) dy.$$

There are two types of convolution. They are:

- Linear convolution
- Circular convolution

Linear convolution

Convolution is an integral concatenation of two signals. It has many applications in numerous areas of

signal processing. The convolution described above is nothing but linear convolution. The most popular application is the determination of the output signal of a linear time-invariant system by convolving the input signal with the impulse response of the system. Convolution of two signals is equivalent to multiplying the Fourier transform of the two signals.

The linear convolution of two continuous time signals $x(t)$ and $h(t)$ is defined by

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) h(t - \tau) d\tau$$

For discrete time signals $x(n)$ and $h(n)$, the integration is replaced by a summation

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k) h(n - k)$$

II. PROPOSED CONVOLUTION DESIGN

Our proposed design consists multiplexer, SIPO (Serial input parallel output), binary multiplier, Output multiplexer and register.

The block diagram of the proposed architecture is shown in fig.1.

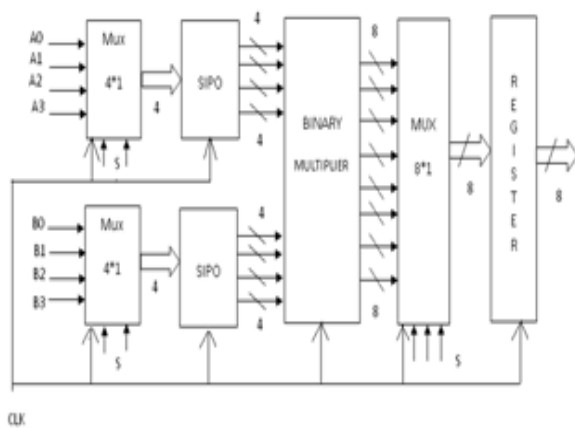


Fig.1: Proposed convolution design

Input multiplier is placed at the input side in order to provide the synchronization between the both inputs and based on the user requirement the input can be

sent at the any point of time (not exactly same order). In order to provide parallel data for the parallel multiplier we introduce the SIPO as the element in-between the MUX and the binary multiplier. In order to provide the output for the serial components register placed as final element.

The multiplier is important element for the calculation of the convolution output. In traditional convolution the multiplication it multiplies the all products even if the products sums are zero. Unlike the traditional convolution designs our proposed method designed in such a way that unwanted multiplications can be avoided hence causes the less power and the less area utilization.

Proposed multiplier design

Vedic Mathematics refers to the technique of Calculations based on a set of 16 Sutras, or aphorisms, as algorithms and their upa-sutras or corollaries derived from these Sutras. Any mathematical problems (algebra, arithmetic, geometry or trigonometry) can be solved mentally with these sutras. Vedic Mathematics is more coherent than modern mathematics.

Multiplier implementation using FPGA has already been reported using different multiplier architectures but the performance of multiplier was improved in proposed design

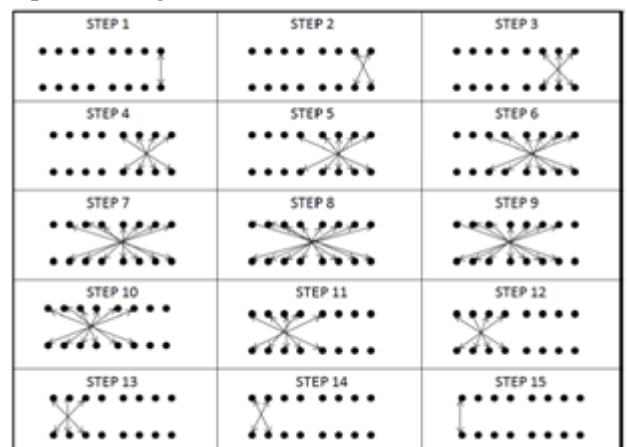


Fig.2: step by step method of multiplying two 8 bit numbers using the Urdhwa TiryakbyamSutra

Let us consider two 8 bit numbers X7-X0 and Y7-Y0, where 0 to 7 represent bits from the Least Significant Bit (LSB) to the Most Significant Bit (MSB).

$$P0 = A0 * B0 \tag{1}$$

$$C1P1 = (A1 * B0) + (A0 * B1) \tag{2}$$

$$C3C2P2 = (A2 * B0) + (A0 * B2) + (A1 * B1) + C1 \tag{3}$$

$$C5C4P3 = (A3 * B0) + (A2 * B1) + (A1 * B2) + (A0 * B3) + C2 \tag{4}$$

$$C7C6P4 = (A4 * B0) + (A3 * B1) + (A2 * B2) + (A1 * B3) + (A0 * B4) + C3 + C4 \tag{5}$$

$$C10C9C8P5 = (A5 * B0) + (A4 * B1) + (A3 * B2) + (A2 * B3) + (A1 * B4) + (A0 * B5) + C5 + C6 \tag{6}$$

$$C13C12C11P6 = (A6 * B0) + (A5 * B1) + (A4 * B2) + (A3 * B3) + (A2 * B4) + (A1 * B5) + (A0 * B6) + C7 + C8 \tag{7}$$

$$C16C15C14P7 = (A7 * B0) + (A6 * B1) + (A5 * B2) + (A4 * B3) + (A2 * B5) + (A1 * B6) + (A0 * B7) + C9 + C11 \tag{8}$$

$$C19C18C17P8 = (A7 * B1) + (A6 * B2) + (A5 * B3) + (A4 * B4) + (A3 * B5) + (A2 * B6) + (A1 * B7) + C10 + C12 + C14 \tag{9}$$

$$C22C21C20P9 = (A7 * B2) + (A6 * B3) + (A5 * B4) + (A4 * B5) + (A3 * B6) + (A2 * B7) + C13 + C15 + C17 \tag{10}$$

$$C25C24C23P10 = (A7 * B3) + (A6 * B4) + (A5 * B5) + (A4 * B6) + (A3 * B7) + C16 + C18 + C20 \tag{11}$$

$$C27C26P11 = (A7 * B4) + (A6 * B5) + (A5 * B6) + (A4 * B7) + C19 + C21 + C23 \tag{12}$$

$$C29C28P12 = (A7 * B5) + (A5 * B6) + (A5 * B7) + C22 + C24 + C26 \tag{13}$$

$$C30P13 = (A7 * B6) + (A6 * B7) + C25 + C27 + C28 \tag{14}$$

$$P14 = (A7 * B7) + C29 + C30 \tag{15}$$

$$P15 = (A7 * B7) \tag{16}$$

III. DECONVOLUTION

Straight method of deconvolution of two finite length discrete-time sequences proposed. Computing of the deconvolution is almost like the traditional long-hand division and polynomial division. Many methods were designed for the calculation of the deconvolution but in our designed an alkalic recursive deconvolution method used for finite length sequences is calculated. The long division recursion method can be accomplished in below manner.

				4 2 3 => f(n)											
4	5	3	4	16	28	34	37	17	12						
				16	20	12	16								
					8	22	21	17	12						
					8	10	6	8							
						12	15	9	12						
						12	15	9	12						
						0	0	0	0						

Fig.3: proposed Deconvolution method

For implementation of division operation vedic mathematic Nikhilam algorithm can be used moreover vedic multiplication based partial products design adopted for the multiplication. For instance, the first example in subsection A may be retrord, solving for f(n) given g(n) and y(n). The sequences are set up in a standardized long division fashion, as shown in Fig.3, but where no carries are executed out of a column [11]. In our proposed algorithm the operation involves normal addition as the part of the operation procedure and the stepwise subtraction which is much different

from the traditional methods, to fulfill overall design of the long grain deconvolution proposed multiplication can be utilized.

IV RESULTS AND DISCUSSION

The fig shows the vedic multiplier based convolution result. By this the partial products can be reduced and as well as the operation time also will be reduced.

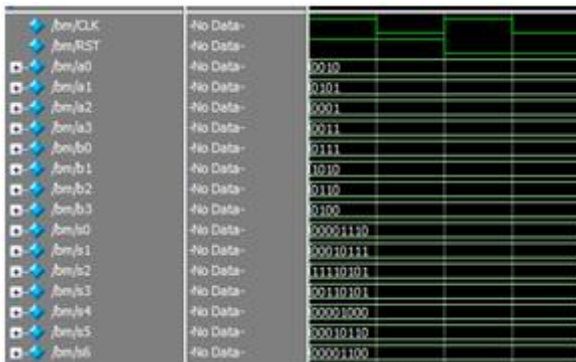


Fig.4: Simulation of convolution

Different inputs applied to the convolution and the outputs of convolution applied to the deconvolution, hence the applied inputs at convolution reproduced at the deconvolution output. Not only for the simulation has it given the better results even for the synthesis. Algorithm coding written in verilog, simulated by Model-sim and synthesized using XILINX.

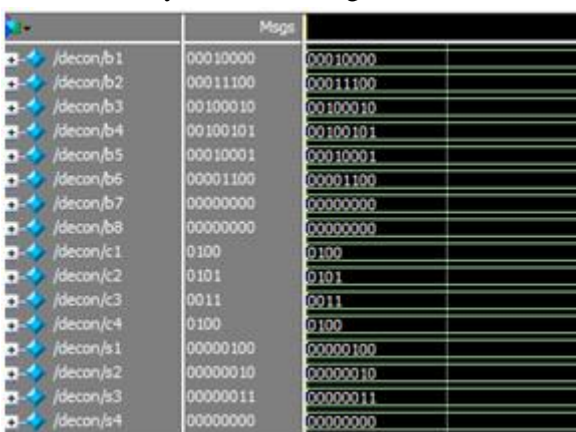


Fig.5: Simulation of Deconvolution

CONCLUSION

Convolution and deconvolution algorithms designed by using the advanced methods. This will be suited for the many applications even if the serial and parallel

processing circuits. In our design we used the advance multiplication and division methods. Vedic methods are the well suited and it meets the all basic requirements like area, power and speed.

REFERENCES

[1] J. G. Proakis and D. G. Manolakis, "Digital Signal Processing: Principles, Algorithm, and Applications," 2nd Edition. New York Macmillan, 1992.

[2] Pierre, John W. "A novel method for calculating the convolution sum of two finite length sequences." Education, IEEE Transactions on 39.1 (1996): 77-80.

[3] Rudagi, J. M., Vishwanath Ambli, Vishwanath Munavalli, Ravindra Patil, and Vinaykumar Sajjan. "Design and implementation of efficient multiplier using Vedic mathematics." (2011): 162-166.

[4] Vaidya, Sumit, and Deepak Dandekar. "Delay-Power Performance Comparison of multipliers in VLSI circuit design." International Journal of Computer Networks & Communications (IJCNC) 2.4 (2010): 47-56.

[5] Akhter, Shamim. "VHDL implementation of fast NxN multiplier based on vedic mathematic." In Circuit Theory and Design, 2007. ECCTD 2007. 18th European Conference on, pp. 472-475. IEEE, 2007.

[6] Thapliyal, Himanshu, and M. B. Srinivas. "High Speed Efficient NxN Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics." Enformatika Trans 2 (2004): 225-228.

[7] Lomte, Rashmi K., and P. C. Bhaskar. "High Speed Convolution and Deconvolution Using Urdhva Triyagbhyam." VLSI (ISVLSI), 2011 IEEE Computer Society Annual Symposium on. IEEE, 2011.

[8] Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja, "Vedic Mathematics." Motilal Banarsidass, New Delhi, India, 1994.



[9] Hanumantharaju, M. C., et al. "A High Speed Block Convolution using Ancient Indian Vedic Mathematics." Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on. Vol. . IEEE, 2007.

[10] Itawadiya, Akhalesh K., et al. "Design a DSP operations using vedic mathematics." Communications and Signal Processing (ICCSP), 2013 International Conference on. IEEE, 2013.

[11] Surabhi Jain, Sandeep Saini "High Speed Convolution and Deconvolution Algorithm" IEEE, 2014.