International Journal of Emerging Engineering Research and Technology
Volume 3, Issue 7, July 2015, PP 122-128
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

Convolution and Deconvolution Algorithm using Vedic Mathematics with Wallace Adder

Shivshankar[1], Pattem sampath kumar[2]

[1]*Department of ECE, Malla Reddy Institute of Technology, Hyderabad, India*
[2]*Department of ECE, Malla Reddy Institute of Technology, Hyderabad, India*

ABSTRACT

The basic blocks in implementing convolution and deconvolution are multipliers and dividers, there are many multipliers but we require high speed multipliers. Presently we have Ancient Indian Vedic multiplication using Urdhvatriyagbhyam and Wallace adder. It's a high speed multiplier and divider also using Nikhilam algorithm. This paper presents the implementation of linear convolution and circular convolution using vedic mathematics and Wallace adder and deconvolution using Nikhilam algorithm in VHDL. Simulation and Synthesis are performed using Xilinx ISE design suit 14.7.

Keywords: Urdhvatriyagbhyam, Nikhilam algorithm, Wallace adder.

INTRODUCTION

A digital signal processor (DSP) is an integrated circuit designed for high-speed data manipulations, and is used in audio, communications, image manipulation, and other data-acquisition and data-control applications. Digital technology such as personal computers (PCs), assist us in many ways: writing documents, spell checking, and drawing. Unfortunately, the world is analog, and electronic analog computers are not as versatile as digital computers. In order to make use of the tremendous processing power that digital technology offers us, we must do the following: · Convert the analog signals into electrical signals, using a transducer (such as a microphone, as shown in the diagram). · Digitize these signals (i.e., convert them from analog to digital using an analog-to-digital converter (ADC)), as shown in the diagram. Once the signal is in digital form, our computer can easily process it through a digital signal processor. The DSP specializes in processing these signals, which makes it slightly different from microcomputers, microcontrollers, and general-purpose microprocessors. After the DSP has processed the signal, the output signal must be converted back to analog form so that we can sense it. This is the digital-to-analog (DAC) conversion stage in the diagram. A loudspeaker, for example, would reproduce analog signals coming from the DAC into sound. So, we can see that to process the signal digitally, we need to convert it at least twice. Is it worth it? As you will see, it really is, at least until someone designs an analog computer as versatile as a digital one.

CONVOLUTION USING VEDIC MULTIPLICATION

The proposed Vedic multiplier is based on the Vedic multiplication for mul(Sutras). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. In this work, we apply the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware.

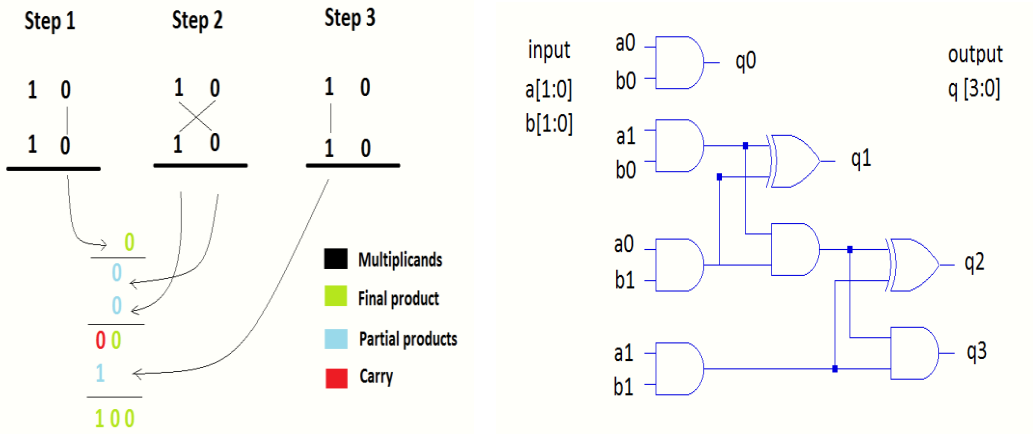
2x2 bit Multiplier

In 2x2 bit multiplier, the multiplicand has 2 bits each and the result o f multiplication is o f 4 bits. So in input the range of inputs goes fro m (00) to (11) and output lies in the set of (0000, 0001, 0010, 0011, 0100, 0110, 1001). Focusing on these facts, a simple design. By using

Address for correspondence:

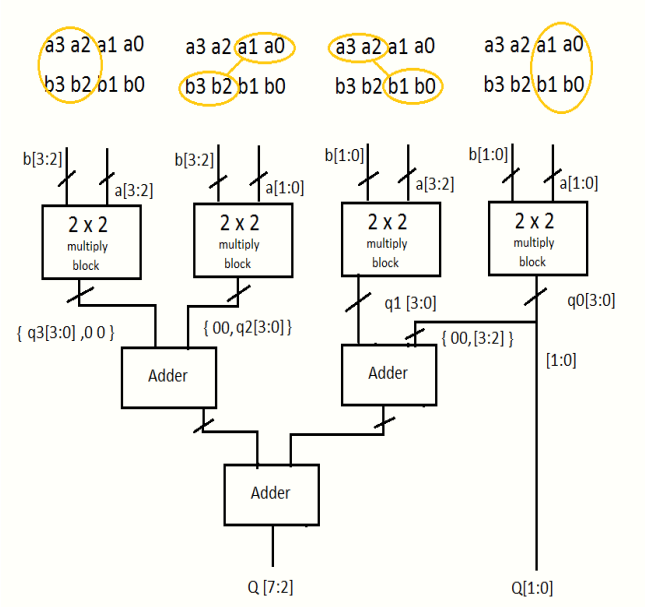
jtinureddy@gmail.com

Urdhvatriyagbhyam, the multiplication takes place as illustrated in Fig. Here multiplicands a and b are taken to be (10) both. The first step in the multiplication is vertical multiplication of LSB of both multiplicand s, then is the second step, that is crosswise multiplication and add it n of the partial pro ducts. Then Step 3 involves vertical multiplication of MSB of the multiplicands and addition with the carry propagated from Step 2.



4x4 Bit Multiplier

The 4x4 Multiplier is made by using 4, 2x2 multiplier blocks. Here, the multiplicands are of bit size (n=4) where as the result is of 8 bit size. The input is broken into smaller chunk s of size of n/2 = 2, for both inputs, that is a and b. These newly formed chunks of 2 bits are given as input to 2x2 multiplier block and the result produced 4 bits, which are the output produced from 2x2 multiplier block are sent for addition to an addition tree.



In this section a novel multiplier architecture [5] based on Urdhva Triyagbhyam Sutra of Ancient Indian Vedic Mathematics is embedded into proposed method of convolution to improve its efficiency in terms of speed and area. This method for discrete convolution using Vedic multiplication algorithm is best introduced by a basic example. For this example, let f(n) equal the finite length sequence (4 2 3) and g(n) equal the

Finite length sequence (4 5 3 4). The linear convolution of f(n)&g(n) is given by

$$y(n) = f(n) * g(n)$$
$$y(n) = \sum_{k=-\infty}^{\infty} f(k)g(n-k)$$

This can be solved by several methods, resulting in the sequence y (n) = (16 28 34 37 17 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:

```
g(n):          4  5  3  4
f(n):       *  4  2  3
-------------------------
            12 15  9 12
         08 10  6  8
      16 20 12 16
-------------------------
y(n): 16 28 34 37 17 12
```

Convolution by Propose Method

The computation of the convolution sum, the approach is similar to multiplication calculation, except carries are not performed out of a column. This first example shows the simplicity of this method and how easily the calculation can be performed. As shown below, this method can be used to check intermediate values the convolution sum is computed using graphical convolution. Multiplication and summation. For a given value of n, the summation is a product of the sequence f(k) and the folded and translated sequence g(n-k).

$$v_n(k) = f(k)g(n-k)$$

The value of the convolution sum for each value of n is

$$f(n) * g(n) = \sum_k v_n(k)$$

The final answer for the graphical convolution method This answer was verified above using the new method. The sequence vn (k), which is an intermediate answer in computing the convolution sum, may also be checked as shown below using the method presented in this paper. the ease in computing the convolution sum for finite sequences using this new method is as fallows.

```
g(n):          4  1  2
f(n):       *  2  3  4      k
-------------------------------
v(1):          16 04 08      1
v(0):       12 03 06         0
v(-1):   08 02 04            -1
-------------------------------
y(n):    08 14 23 10 08

n:       -1  0  1  2  3
```

Circular Convolution

Circular convolution has many applications and is usually introduced to electrical engineering students in a digital signal processing. The novel method for computing linear convolution using Vedic mathematics from above subsection is easily modified for circular convolution. This method of computing circular convolution is best illustrated by example.

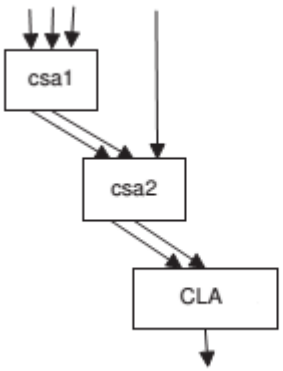
Let f(n) = (2 3 1 0) and g(n) = (4 5 2 1). The circular convolution of f(n) and g(n) is given by

$$y(n) = f(n) * g(n)$$
$$y(n) = \sum_{k=0}^{N-1} f(k)g((n-k) mod N)$$

y(n)= (13 13 23 23) where N is the length of the sequences. This circular convolution calculation may be performed similar to the method for linear convolution from above subsection. The multiplier architecture is implemented using Vedic algorithm. The location of the triangle of bold faced numbers is repositioned for circular convolution compared with linear convolution. The location of these numbers is due to the circular translation in circular convolution. The far left value in the circular convolution solution corresponds to y (N - 1).where N is the length of the sequence

	2	3	1	0
⊗	4	5	2	1
	2	3	1	0
	6	2	0	**4**
	5	0	**10**	**15**
	0	**8**	**12**	**4**
	13	13	23	23
	y(3)	y(2)	y(1)	y(0)

Wallace Adder



In this we are using carry save adders and cla adders in csa adder using full adders. we are having 4 bit Wallace adder first three inputs given to csa1 Sum, carry, forth input given to csa2 then sum and carry given to cla then we will get output.

DECONVOLUTION

A direct method is also presented for the deconvolution of two finite length discrete-time sequences. This deconvolution method is similar to computing long-hand division and polynomial division, just as the direct convolution method is similar to multiplication. Many other deconvolution methods are available. In this section, a basic recursive deconvolution method for finite length sequences is computed. This recursion can be carried out in a manner similar to long division.

		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	

Deconvolution by Proposed Method

Division operation is implemented by using Nikhilam algorithm based on Vedic mathematics while to obtain partial products Vedic multiplier is used. For instance, the first example in subsection A may

be reworked, solving for f(n) given g(n) and y(n). The sequences are set up in a fashion similar to long division but where no carries are performed out of a column.

Vedic Divider

Nikhilam Algorithm

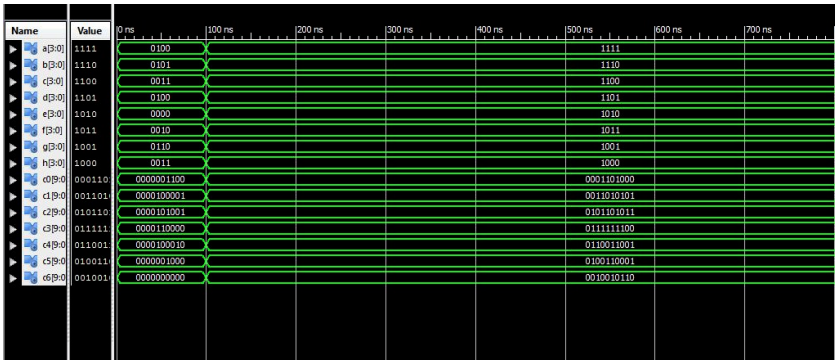
The Nikhilam sutra goes as follows: Nikhilam Navatascaramam Dasatah, literally meaning all from 9 and the last from 10. To illustrate the method further, we will take an example. Let us work out 123/8.

```
8      2
1 2  | 3
0 2  | 8
-------
1 4  | 11
1 5  | 3
```

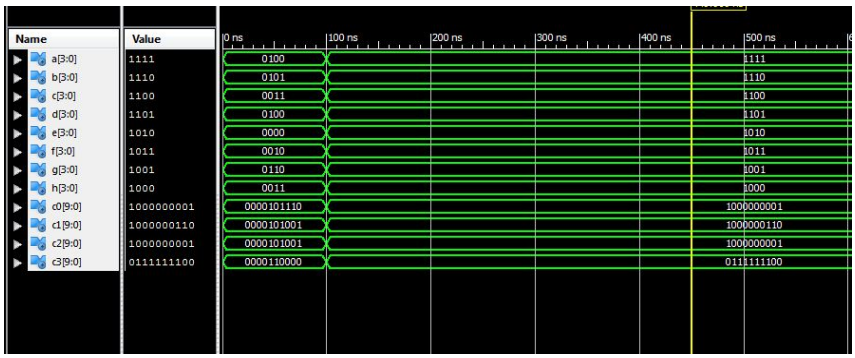
The first line consists of the denominator followed by its 10's complement (2 is 8's 10's complement). The numerator has been divided by a "j" such that there are as many digits to the right of the "j" as there are digits in the denominator. We then put a zero under the first digit of the numerator. Now add up the digits in that column of the numerator to get a sum of 1 (1 + 0 = 1). Multiply it by the 10's complement to get 2 (1 x 2 = 2). Put that under the second digit of the numerator. The sum of the digits under the second column is 4. Multiplying this by the 10's complement gives us 8. Put the 8 under the third digit of the numerator, right of the "j". Now we add up the numbers under the columns. Note that there is no carry over from the right of the "j" to the left of it. Following the rules on how to deal with a remainder greater than the denominator, we divide the remainder by the denominator and add the new quotient to the original quotient and retain the new remainder as the final remainder. This method is extended for other numbers. Nikhilam division algorithm just involves the addition of numbers which is very much different from the traditional division technique including multiplication of big numbers by the trial digit of the quotient at each step and subtracts that result from dividend at each step

SIMULATIONWAVEFORMS

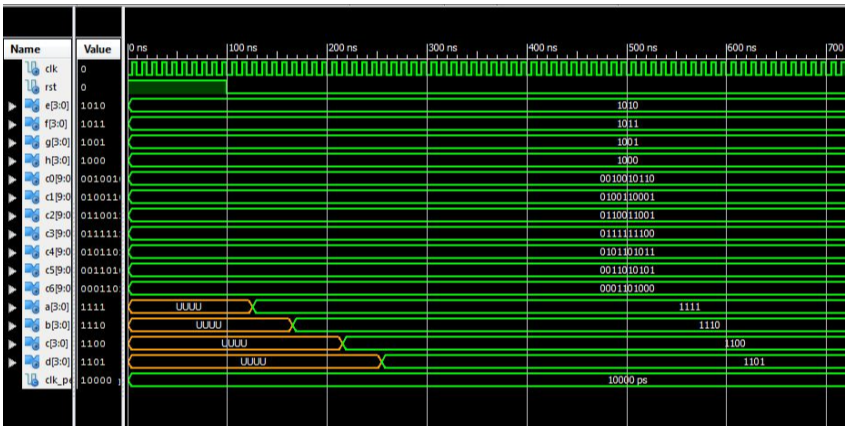
Convolution



Circular convolution



Deconvolution



CONCLUSION

The main focus of this paper is to introduce a method for calculating the linear convolution, circular convolution and deconvolution with the help of Vedic algorithms that is easy to learn and perform. The execution time and area of the proposed method for convolution using vedic multiplication algorithm is compared with that of convolution with the simple multiplication is less. From the simulated results

REFERENCES

[1] J. G. Proakis and D. G. Melonakos, "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, Summit, 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. ECCTD2007. 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. 2. 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.

AUTHORS' BIOGRAPHY

Shivshankar has completed his B.E in electronics and communication engineering from AIET Gulbarga, VTU Belgaum, and Karnataka affiliated college in 2013.he is pursuing his M.Tech in VLSI and Embedded system from MALLA REDDY INSTITUTE OF TECHNOLOGY, JNTUH affiliated college

Pattem Sampath Kumar is an associate professor and HOD of ECE department at Malla Reddy institute of technology, Hyderabad. He received his bachelor degree in electronics and communication engineering from SNIST and M.E from VCE and pursuing PhD FROM JNTUA his research interests in WSN and VLSI