IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 3, Ver. II (May-Jun. 2014), PP 48-53
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

High Speed Vedic Multiplier in FIR Filter on FPGA

Mrs. Pooja .S.Puri[1], Mr. U. A. Patil[2]
[1]*Shivaji university, D. K. T. E. Society's Textile & Engineering Institute Rajwada, Ichalkaranji,*
[2] *D. K. T. E. Society's Textile & Engineering Institute, Shivaji university Rajwada, Ichalkaranji,*

Abstract: *Digital signal processors (DSPs) are very important in various engineering disciplines. Fast multiplication is very important in DSPs for convolution, Fourier transforms etc .In this paper a fast method for multiplication based on ancient Indian Vedic mathematics is used in FIR filter design .The proposed Vedic multiplier is based on Vedic multiplication sutras. Vedic multiplication based on Urdhva tiryakbhyam sutra. This algorithm is applied to digital arithmetic and multiplier architecture is formulated. The coding is done in VHDL (very high speed integrated circuits hardware description language) and synthesis is done using Xilinx ISE series. This Vedic multiplier can bring about great improvement in DSP performance.*
Keywords: *Urdhva tiryakbhyam, Vedic Multiplier, FIR, VHDL*

I. Introduction

The high speed multiplication operation plays vital part in Digital Signal Processor (DSPs) as well as in general processor. Finite Impulse Response (FIR) filter with higher speed is of great importance. FIR filter is also called convolution filter since convolution is the fundamental concept of designing FIR filter. Vedic Mathematics is based on 16 sutras. One of the sutra is Urdhava Tiryagbhyam which delivers a difference in the actual process of multiplication itself. In this paper, we proposed a high speed multiplication operation based on Urdhava Tiryagbhyam sutra to design FIR filter using window. FIR filter has a variety of ways to achieve, with the processing of modern electronic technology, taking use of field programmable gate array FPGA for digital signal processing technology has made rapid development, FPGA with high integration, high speed and reliability advantages, FIR filter implementation using FPGA is becoming a trend. This paper FIR filter is design using vedic multiplier on SPARTAN-3 FPGA.

II. The Vedic Multiplication Method

The proposed Vedic multiplier[1] is based on the Vedic multiplication formulae (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. Vedic multiplication based on Urdhava Tiryakbhyam sutra is discussed below The multiplier is based on an algorithm Urdhava Tiryakbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics.Urdhava Tiryakbhyam 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 with the concurrent addition of these partial products. The parallelism in generation of partial products and their summation is obtained using Urdhava Tiryakbhyam explained in Figure 2. The algorithm can be generalized for n x n bit number. Since the partial products and their sums are calculated in parallel, the multiplier is independent of the clock frequency of the processor. Thus the multiplier will require the same amount of time to calculate the product and hence is independent of the clock frequency. The net advantage is that it reduces the need of microprocessors to operate at increasingly high clock frequencies. While a higher clock frequency generally results in increased processing power, its disadvantage is that it also increases power dissipation which results in higher device operating temperatures. The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure [7]. Due to its regular structure, it can be easily layout in a silicon chip [3]. The Multiplier has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Therefore it is time, space and power efficient. Multiplication of two decimal numbers- 325*738.To illustrate this multiplication scheme, let us consider the multiplication of two decimal numbers (325 * 738). Line diagram for the multiplication is shown in Figure 2. The digits on the both sides of the line are multiplied and added with the carry from the previous step. This generates one of the bits of the result and a carry. This carry is added in the next step and hence the process goes on.

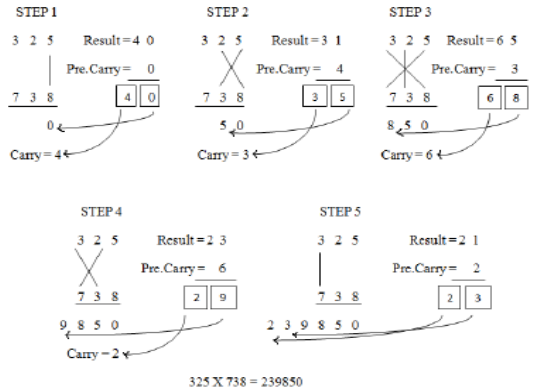

Figure 1: Multiplication of two decimal numbers by Urdhava Tiryakbhyam

Algorithm for 8 X 8 Bit Multiplication Using Urdhava Tiryakbhyam (Vertically and crosswise) for two Binary numbers[8]-

A = A7A6A5A4 A3A2A1A0
X1 X0
B = B7B6B5B4 B3B2B1B0
Y1 Y0
X1 X0
* Y1 Y0

F E D C
CP = X0 * Y0 = C
CP = X1 * Y0 + X0 * Y1 = D
CP = X1 * Y1 = E
Where CP = Cross Product

To illustrate the multiplication algorithm, let us consider the multiplication of two binary numbers a3a2a1a0 and b3b2b1b0. As the result of this multiplication would be more than 4 bits, we express it as... r3r2r1r0. Line diagram for multiplication of two 4- bit numbers is shown in Figure 2 which is nothing but the mapping of the Figure 1 in binary system. For the simplicity, each bit is represented by a circle. Least significant bit r0 is obtained by multiplying the least significant bits of the multiplicand and the multiplier. 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. The same operation continues until the multiplication of the two MSBs to give the MSB of the product. For example, if in some intermediate step, we get 110, then 0 will act as result bit (referred as rn) and 11 as the carry (referred as cn). It should be clearly noted that cn may be a multi-bit number. Thus we get the following expressions:

r0=a0b0*;* (1)
c1r1=a1b0+a0b1*;* (2)
c2r2=c1+a2b0+a1b1 + a0b2*;* (3)
c3r3=c2+a3b0+a2b1 + a1b2 + a0b3*;* (4)
c4r4=c3+a3b1+a2b2 + a1b3*;* (5)
c5r5=c4+a3b2+a2b3*;* (6)
c6r6=c5+a3b3 (7)

With c6r6r5r4r3r2r1r0 being the final product. Hence this is the general mathematical formula applicable to all cases of multiplication. The hardware realization of a 4-bit multiplier is shown in Figure 3. This hardware design is very similar to that of the famous array multiplier where an array of adders is required to arrive at the final product. All the partial products are calculated in parallel and the delay associated is mainly the time taken by the carry to propagate through the adders which form the multiplication array.

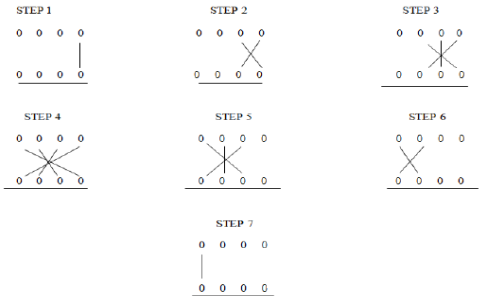

Figure 2: Line diagram for multiplication of 4-bit numbers

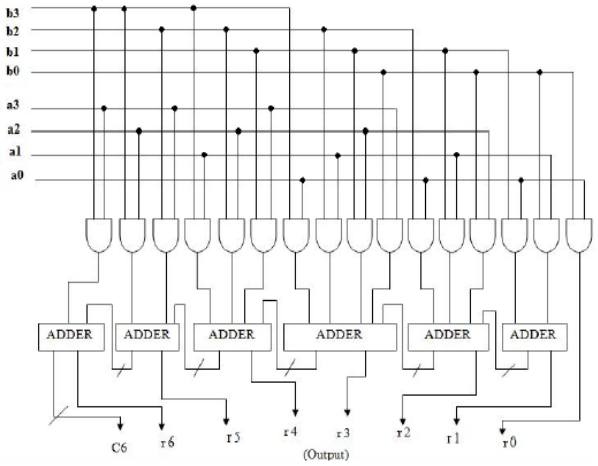

Figure 3: Hardware Architecture Of 4x4 Bits Vedic Multiplier

III. Simulation Results

Table 1 and Table 2 indicate the device utilization summary of the array and Vedic multiplier for 4 bit and 8 bit respectively. Table 3 gives the timing report of the implementation in terms of maximum combinational path delay. While Figure 5 indicates the RTL schematic of the 8 bit Vedic multiplier.

Table 1: Device Utilization Summary

Name of the Multiplier (8bit)	Number of slices	Number of 8 i/ps	Number of IOs	Number of bounded IOBs
Array	71 out of 768	123 out of 1536	32	32 out of 124
Vedic	79 out of 432	133 out of 864	32	32 out of 60

Table 2: Device Utilization Summary

Name of the Multiplier (4 bit)	Number of slices	Number of 4 i/ps	Number of IOs	Number of bounded IOBs
Array	19 out of 768	33 out of 1536	16	16 out of 124
Vedic	15 out of 768	27 out of 1536	16	16 out of 124

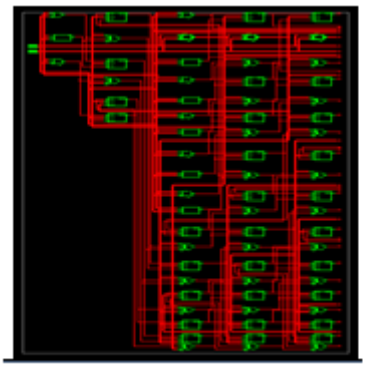

Figure 5: RTL Schematic Of 8x8 Bits Vedic Multiplier

IV. Basic Principle And Structure Of The Filter

Set the unit impulse response h(n) of finite unit impulse response filter as the sequence of length N, the transfer function usually has the following form[7]

$$H(Z) = \sum_{n=0}^{N-1} h(n) * z^{-n} \qquad (1)$$

As we can see from differential equation, FIR filter order is N-1, length N. System output depends on a function of input and has no direct relationship with the past output, it does not contain feedback branch [8]. If the FIR digital filter unit pulse response h(n) is real numbers, h(n) satisfies the odd symmetry or even symmetry, that is

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i) \qquad (2)$$

Differential equation is described as

$$h(n) = h(N-1-n)\ 0 \leq n \leq N-1, \text{ Even symmetry,}$$
$$-h(N-1-n)\ 0 \leq n \leq N-1, \text{Odd symmetry} \qquad (3)$$

In this paper we have used an Odd symmetry, the structure used in paper is shown in Fig [4] [6]

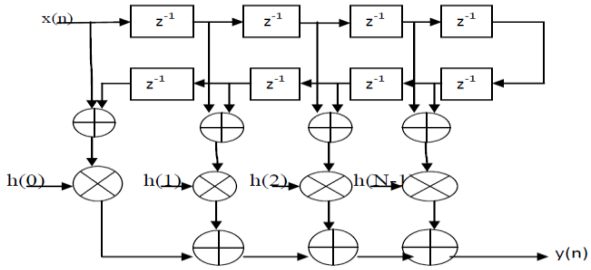

Figure 4: Linear phase FIR filter

4.1 Fir Filter Design

Design of Low Pass FIR Filter of 128 order, the Filter specifications are sampling frequency fs=4.8kHz and cutoff frequency fc=100Hz, Response Type=Low Pass filter, width of data in=12 bits, Pass band attenutation= 3db.The Filter is designed using MAC operation, vedic multiplier is used for multiplication. The filter coefficients are generated by using MATLAB software using window method. In order to expediently deal with the coefficient by FPGA devices, there are two operations to be done: (1) converting the floating-point value into a fixed point value that is called quantification. (2) converting fraction to integer.[6]. The hardware of digital FIR filter is shown in Fig 7. The input a.c signal is given to 12 bits ADC has sampling rate of 10µs , FIR filter using high speed vedic multiplier is stored on FPGA, use of SPARTAN-3,device is XC3S400, the digital output is given to serial DAC-4275, the output can be observe at CRO.

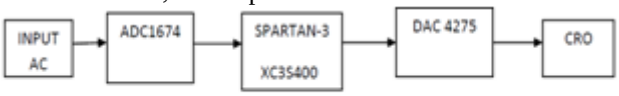

Figure 6: Basic block diagram

4.2 Hardware Results

Below figures indicates the FIR filter on FPGA using high speed vedic multiplier. Fig 9 indicates the oupt put of FIR filter at 45Hz .The filter has cutoff frequency 100Hz so it will pass the signal has frequency below 100Hz and stops above 100Hz ,as indicated in Fig 10.

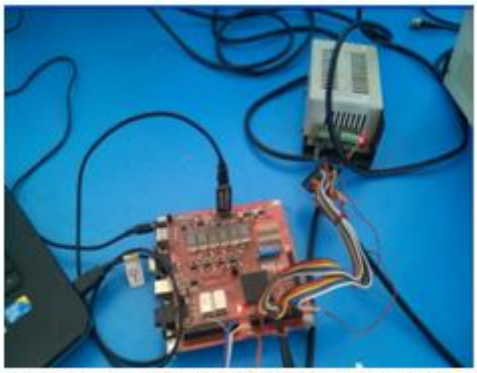

Figure 7: Hardware implementation on FPGA

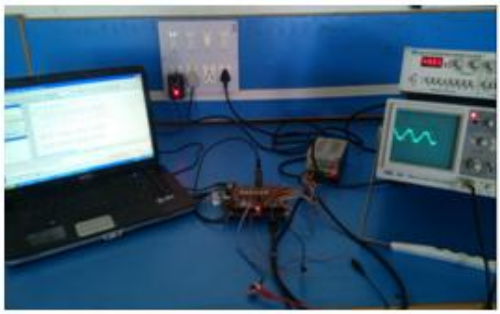

Figure 8: An arrangement for FIR Filter.

Figure 9: Output below cutoff at 45 Hz

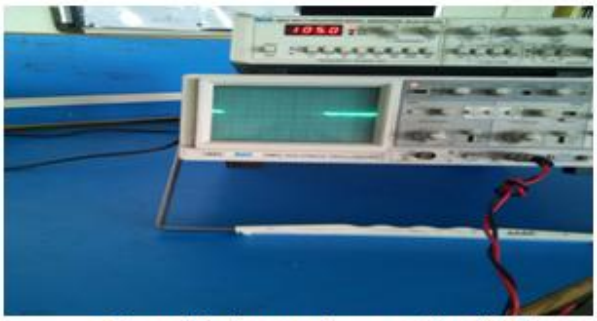

Figure 10: Output above cutoff at 105Hz

V. Conclusion

From the above synthesis report and timing report, it can be inferred that Vedic multiplier achieves higher speed by reducing gate delay. Vedic Multiplier can be further optimized by implementing Carry Save Adder (CSA) so as to increase the speed. Vedic Mathematics gives us a clue of symmetric computation. This paper proposed a design of FIR filter using Urdhava Tiryagbhyam method of Vedic mathematics. The

algorithms of Vedic mathematics are much more efficient than conventional mathematics. It is shown that the Urdhava Tiryagbhyam method is faster than the conventional method The proposed Vedic multiplier proves to be highly efficient in terms of the speed. The main advantage is delay increases slowly as the input bits increases. Most of the important DSP algorithms, such as convolution, discrete Fourier transforms, fast Fourier transforms, digital filters, etc, incorporate multiply-accumulate computations. Since the multiplication time is generally far greater than the addition time, the total processing time for any DSP algorithm primarily depends upon the number of multiplications. Hence, this multiplier can be used to implement the above DSP algorithms.

References

[1]. 1S. S. Kerur, 1Prakash Narchi, 1Jayashree C N, 2Harish M Kittur and 3Girish V A "Implementation of Vedic Multiplier for Digital Signal Processing" International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011

[2]. Jagadguru Swami Sri Bharati Krisna Tirthaji Maharaja, Vedic Mathematics: Sixteen Simple Mathematical Formulae from the Veda, Delhi (1965).

[3]. H. Thapliyal and M. B. Shrinivas and H. Arbania, "Design and Analysis of a VLSI Based High Performance x Low Power Parallel Square Architecture", Int. Conf. Algo. Math.Comp. Sc., Las Vegas, June 2005, pp. 72-76.

[4]. P. D. Chidgup kar and M. T. Karad, "The Imp lementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engg. Edu, vol.8, no.2, 2004.

[5]. Shamim Akhter,"VHDL Implementation Of Fast NXN Multiplier Based On Vedic Mathematics", Jay p ee Institute of Information Technology University, Noida, 201307 UP, INDIA, 2007 IEEE.

[6]. Zhou Ya Feng , Li Yue Hua and Zhu Hao, "Design of 16 order FIR f ilter based on FPGA."The 1st International Conference on Information Science and Engineering (ICISE2009).

[7]. T.-S.Chang.C.Chen,and C.-W.Jen, New distributed arithmetic algorithm and its application to IDCT." IEE Proceedings Circuits,Devices and Systems,
vol.146,no.4,pp.159-163,Aug.1999.

[8]. Charles. Roth Jr. "Digital Systems Design using VHDL," Thomson Brooks/Cole, 7th reprint, 2005

[9]. Sangook Moon,"Elliptic Curve Scalar Point Multiplication Using Radix-4 Booth's Algorithm", International Symposium on Communications and Information Technologies 2004(ISCIT2004),pp. 80-83, Japan, October 26-29,2004.

[10]. A.P. Nicholas, K.R Williams, J. Pickles, "Application of Urdhava Sutra", Spiritual Study Group, Roorkee (India),1984.

[11]. http://www.trivology.com/articles/534