Implementation Of High Speed FIR Filter Based On Ancient Vedic Multiplication Technique

Swapnil Manohar Mehkarkar\textsuperscript{1}, Snehal J. Banarase\textsuperscript{2}

\textsuperscript{1,2}Department of E&TC, G.H.Raisoni College of Engineering and Management, Amravati (444701), Maharashtra, India.

Abstract— This paper proposed the design of high speed FIR Filter using the Vedic Multiplication techniques of Ancient Indian Vedic Mathematics that have been modified to improve performance. The design of an FIR requires three basic building blocks multiplication, addition, signal delay. FIR filter is based on complex arithmetic due to the magnitude and phase responses of the channel impulse characteristics. A high quality filter will in general require more complex number multiplications the output of filter suffers if the multiplier is not fast, so there is a need of high speed multiplier in filter. There are so many multiplication algorithms exist now-a-days having different algorithms and structural levels. Our work proved that Vedic multiplication technique is the best algorithm in terms of speed. Further we have seen that the conventional multiplication have some limitations, so to overcome those limitations a novel approach has been proposed to design the Vedic multiplier. Vedic Mathematics is the ancient Indian system of mathematics which based on 16 Sutras which deals with various branches of mathematics like arithmetic, algebra, geometry, etc, from which Urdhva Triyagbhyam sutra is used for multiplication. It enables parallel generation of intermediate products, eliminates unwanted multiplication steps with zeros. To design proposed high speed filter verilog hardware description language (HDL) has been used.

Keywords— Vedic Mathematics, Vedic multiplier, Urdhva Triyagbhyam Sutra, FIR Filter.

I. INTRODUCTION

Multiplication is an important fundamental function in arithmetic operations. Multiplication-based operations such as convolution, Fast Fourier Transform (FFT), filtering and in microprocessors in its Arithmetic Logic Unit (ALU), etc [1]. In many Digital Signal Processor algorithms multiplication dominates the execution time, so there is a need of high speed multiplier.

Since multiplication is a most important factor in speed of the DSP processors. As there are many new developments in the technology there is a need of higher and higher speed of multipliers. Many researchers are still working on to design multipliers which offers high speed, low power consumption, regularity of layout and hence less area or even combination of them in multiplier [19].

The increased complexity of various applications, demands not only faster multiplier chips but also smarter and efficient multiplying algorithms that can be implemented in the chips. Two most common multiplication algorithms followed in the digital hardware are Array multiplication algorithm and Booth multiplication algorithm. The drawback of these two algorithms is a large propagation delay is associated with it.

The word “Vedic” is derived from the word “Veda” which means the store-house of all knowledge. Like “Ayurveda” which means store-house of all the knowledge regarding Medical field, and “Vedic Mathematics” which means store-house of all the knowledge regarding Mathematics. Vedic mathematics is mainly based on 16 Sutras which deals with various branches of mathematics like arithmetic, algebra, geometry, etc [1],[7]. Jagadguru Shankaracharya Bharati Krishna Teerthaji Maharaja (1884-1960) comprised all this work together and gave its mathematical explanation. After extensive research in Atharva Veda Swamiji constructed 16 sutras (formulae) and 16 Upa sutras (sub formulae). Vedic mathematics deals with several basic as well as complex mathematical operations. The methods of Vedic mathematics are extremely simple and very powerful.

The following are the 16 main sutras or formulae of Vedic math and their meaning in English.
The Urdhva-Tiryangbhyam Sutra from Vedic Mathematics is used for multiplication based on the methodology of Urdhva-Tiryangbhyam Sutra we are designed a new multiplier. This Vedic method of multiplication strikes a difference in the actual multiplication process. It enables parallel generation of intermediate products, eliminates unwanted multiplication steps with zeros. The Urdhva-Tiryangbhyam Sutra is based on the Vertical and Crosswise algorithm of ancient Indian Vedic Mathematics. Multiplier based on Vedic Mathematics is one of the high-speed and low power multiplier.

The step vise architecture of Urdhva Tiryangbhyam Sutra is as shown in Fig.1 for 4x4 binary multiplications.

![Fig. 1 4x4 binary multiplication using Urdhva Tiryangbhyam](image)

The multiplication of two numbers (379×657) using Urdhva Tiryangbhyam Sutra is shown in Fig.2 The least significant digit from both the numbers (i.e. 9 of 379 & 7 of 657) are multiplied vertically and the result is added with the previous carry. Initially the carry is taken to be zero. When there are more lines in one step, all the results are added to the preceding carry. The least significant digit of the number thus obtained result acts as one of the final multiplication result digits and the rest act as the carry for the next step. After these the two digits of LSB side are multiplied crosswise and added with the previous carry. Likewise all the digits are multiplied vertically and crosswise as shown in Fig.2.

<table>
<thead>
<tr>
<th>Sr.No.</th>
<th>Vedic Sutras</th>
<th>English Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Ekadhikina Purvena</td>
<td>By one more than the previous One.</td>
</tr>
<tr>
<td>2.</td>
<td>Gunakasamuchyah</td>
<td>The factors of the sum is equal to the sum of the factors.</td>
</tr>
<tr>
<td>3.</td>
<td>Ekanyunena Purvena</td>
<td>By one less than the previous one.</td>
</tr>
<tr>
<td>4.</td>
<td>Gunitasamuchyah</td>
<td>The product of the sum is equal to the sum of the product.</td>
</tr>
<tr>
<td>5.</td>
<td>Chalana-Kalanabhyam</td>
<td>Differences and Similarities.</td>
</tr>
<tr>
<td>6.</td>
<td>(Anurupye) Shunyamanyat</td>
<td>If one is in ratio, the other is zero.</td>
</tr>
<tr>
<td>7.</td>
<td>Puranapuranaabhyam</td>
<td>By the completion or noncompletion.</td>
</tr>
<tr>
<td>8.</td>
<td>Paraavartya Yojayet</td>
<td>Transpose and adjust.</td>
</tr>
<tr>
<td>9.</td>
<td>Sankalana-vyavakalanabhyam</td>
<td>By addition and by subtraction.</td>
</tr>
<tr>
<td>10.</td>
<td>Yaavadunam</td>
<td>Whatever the extent of its deficiency.</td>
</tr>
<tr>
<td>11.</td>
<td>Sopaantyadavamantyam</td>
<td>The ultimate and twice the penultimate.</td>
</tr>
<tr>
<td>12.</td>
<td>Vyashitsamanstih</td>
<td>Part and Whole.</td>
</tr>
<tr>
<td>14.</td>
<td>Nikhilam Navatascharamam Dashatah</td>
<td>All from 9 and last from 10.</td>
</tr>
<tr>
<td>15.</td>
<td>Shunyam Saamyasamuccaye</td>
<td>When the sum is the same that sum is zero.</td>
</tr>
<tr>
<td>16.</td>
<td>Shesanyanyakena Charamena</td>
<td>The remainders by the last digit.</td>
</tr>
</tbody>
</table>

TABLE I
III. IMPLEMENTATION OF PROPOSED MULTIPLIER

To design 64x64 Bit Vedic Multiplier firstly we have designed 2x2 Bit Vedic Multiplier, with the help of this 2x2 Bit Vedic Multiplier we designed a 4x4 Bit Vedic Multiplier, likewise 8x8 Bit, 16x16 Bit, 32x32 Bit and finally 64x64 Bit Vedic Multiplier.

A. 2x2 Bit Vedic Multiplier

Fig.3. shows the example of 2x2 Bit Vedic Multiplication, here in this example as per the rules of Urdhva-Tiryagbhyam Sutra in first step right most two digits of both the numbers are multiplied vertically. In second step end digits are multiplied crosswise. And in last step left end two digits are multiplied vertically. The carry from the second step is added to the third step, and initial carry is considered as zero.

Urdhva Tiryagbhyam sutra for multiplication strikes a difference in the actual process of multiplication itself. It enables parallel generation of intermediate products, eliminates unwanted multiplication steps with zeros. From Fig.3. we can see that the process of multiplication is a parallel process, to start execution for third step it has nothing to do with step number first and second. That means on the clock pulse all steps are ready to start.

In the design of the proposed Vedic multiplier a 2x2 block is a fundamental block (Basic block) is shown in Fig.4. Also symbol of this fundamental block is shown to be used in 4x4 bit Vedic Multiplier. We know that in binary multiplication basically we AND each two bits in 2-input AND gate. First off all vertical bits (LSBs) are ANDed this will result in the LSB of the result. After that the four bits are multiplied crosswise and then result is added using a half adder. The sum output of the half adder is the next bit of the result right to the LSB. The carry output is also added in half adder with the AND output of the MSBs. The carry of this adder is the MSB of the result [17],[18]. Fig.12 shows the simulation output of Vedic Multiplier for 2x2 Bit using VHDL coding.

B. 4x4 Bit Vedic Multiplier

The design of 4x4 block shown in Fig.5 is a simple arrangement of 2x2 Bit Vedic Multiplier blocks in an optimized manner. The first step in the design of 4x4 Vedic Multiplier block will be grouping the 2 bit of each 4 bit input. These pair terms will form vertical and crosswise product terms. Each input bit-pair is handled by a separate 2x2 Vedic Multiplier the schematic of a 4x4 Vedic Multiplier designed using 2x2 blocks [17],[18].

As the output of 2x2 Bit Vedic Multiplier is 4-Bit and the outputs of two 2x2 Bit Vedic Multiplier are added together with the help of 4-Bit adder. And finally the outputs of two 4-Bit adders are added by 8-Bit adder. So for the multiplication of 4x4 Bit Vedic Multiplier we required four 2x2 Bit Vedic Multipliers, Two 4-Bit Adders and one 8-Bit Adder. Fig.13 shows the simulation output of 4x4 Bit Vedic Multiplier using VHDL coding.
C. 8x8 Bit Vedic Multiplier

The design of 8x8 bit Vedic Multiplier is a similar arrangement of 4x4 bit Vedic Multiplier blocks in an optimized manner as in Fig.5. The first step in the design of 8x8 bit Vedic Multiplier will be grouping the 4 bit (nibble) of each 8 bit input. These quadruple terms will form vertical and crosswise product terms. Each input bit-quadruple is handled by a separate 4x4 bit Vedic multiplier to produce eight partial product rows. These partial products rows are then added in an 8-bit adder [17],[18]. To generate final product bits the output of the 8-bit adder is added together using 16-bit adder. The Fig.6 shows the schematic of an 8x8 Vedic Multiplier designed using 4x4 Vedic Multiplier blocks. Fig.14 shows the simulation output of 8x8 Bit Vedic Multiplier using VHDL coding.

D. 16x16 Bit Vedic Multiplier

The design of 16x16 bit Vedic Multiplier is a similar arrangement of 8x8 blocks in an optimized manner as in Fig.6. The first step in the design of 16x16 bit Vedic Multiplier will be grouping the 8 bit (byte) of each 16 bit input. These lower and upper bytes pairs of two inputs will form vertical and crosswise product terms. Each input byte is handled by a separate 8x8 bit Vedic multiplier to produce sixteen partial product rows. These partial products rows are then added in a 16-bit adder [17],[18]. To generate final product bits the output of the 16-bit adder is added together using 32-bit adder. The Fig.7 shows the schematic of a 16x16 bit Vedic Multiplier designed using 8x8 bit Vedic Multiplier blocks. Fig.15 shows the simulation output of 16x16 Bit Vedic Multiplier using VHDL coding.

E. 32x32 Bit Vedic Multiplier

The design of 32x32 bit Vedic Multiplier is a similar arrangement of 16x16 blocks in an optimized manner as in Fig.7. The first step in the design of 32x32 bit Vedic Multiplier will be grouping the 16 bit (byte) of each 32 bit input. These lower and upper bytes pairs of two inputs will form vertical and crosswise product terms. Each input byte is handled by a separate 16x16 bit Vedic multiplier to produce sixteen partial product rows. These partial products rows are then added in a 32-bit adder. To generate final product bits the output of the 32-bit adder is added together using 64-bit adder. The Fig.8 shows the schematic of a 32x32 bit Vedic Multiplier designed using 16x16 bit Vedic Multiplier blocks. Fig.16 shows the simulation output of 32x32 Bit Vedic Multiplier using VHDL coding.
F. 64x64 Bit Vedic Multiplier

The design of 64x64 bit Vedic Multiplier is a similar arrangement of 32x32 blocks in an optimized manner as in Fig.8. The first step in the design of 64x64 bit Vedic Multiplier will be grouping the 32 bit (byte) of each 64 bit input. These lower and upper bytes pairs of two inputs will form vertical and crosswise product terms. Each input byte is handled by a separate 32x32 bit Vedic multiplier to produce sixteen partial product rows. These partial products rows are then added in a 64-bit adder. To generate final product bits the output of the 64-bit adder is added together using 128-bit adder. The Fig.9. shows the schematic of a 64x64 bit Vedic Multiplier designed using 32x32 bit Vedic Multiplier blocks. Fig.17 shows the simulation output of Vedic Multiplier for 64x64 Bit using VHDL coding.

![Fig.9 Block diagram of 64x64 Bit Vedic Multiplication](image)

G. FIR Filter

In modern electronic systems, complex arithmetic computation plays an important role in the implementation of different Digital Signal Processing (DSP) and scientific computation algorithms. Digital finite impulse response filtering introduces one of many computationally demanding signal processing tasks. FIR filter is based on complex arithmetic due to the magnitude and phase responses of the channel impulse characteristics.

FIR filters are generally characterized by a high order (number of taps) to obtain sharp transition bands that, in case of high speed real time computation, require a lot of resources and have high power dissipation. In particular, for complex FIR filters, the hardware complexity is mostly determined by the number of complex multipliers. So the Complex multipliers determine the performance, area and power dissipation of filters.

In this paper we are designing FIR Filter as an application of Vedic Multiplier. Since to design FIR Filter our first step is to design Complex Number Multiplication using Vedic Multiplier.

1) 64x64 Bit Complex Number Multiplication Using Vedic Multiplier

A Complex Number is a number that can be expressed in the form \(a + bi\) in exactly one way, where \(a\) and \(b\) are real numbers and \(i\) is the imaginary unit, which satisfies the equation \(i^2 = -1\). In this expression, \(a\) is the real part and \(b\) is the imaginary part of the complex number. Complex multiplication is a more difficult operation to understand from either an algebraic or a geometric point of view. Let’s do it algebraically, and let’s take specific complex numbers to multiply, say \(3 + 2i\) and \(1 + 4i\). Each has two terms, so when we multiply them, we’ll get four terms:

\[(3 + 2i)(1 + 4i) = 3 + 12i + 2i + 8i^2\]

The \(12i + 2i\) simplifies to \(14i\), of course. So now what about the \(8i^2\), as we know that \(i\) is a abbreviation for \(\sqrt{-1}\), the square root of \(-1\). In other words, \(i\) is something whose square is \(-1\). Thus, \(8i^2\) equals \(-8\). Therefore, the product \((3 + 2i)(1 + 4i)\) equals \(-5 + 14i\).

\[(3 + 2i)(1 + 4i) = -5 + 14i\]

If we generalize this example, we’ll get the general rule for multiplication,

\[(a + bi) (c + di) = (ac - bd) + (ad + bc)i\]

Fig.18 shows the Simulation output of 64x64 bit complex number multiplication using vedic multiplier techniques. Here for simulation 64 bit real number and 64 bit imaginary number are used, as discussed in following example,

\[
(3584 + 203776 i) \times (896 + 7340032 i) = ([ (3584 \times 896) - (203776 \times 7340032) ]
+ i [ (203776 \times 896) + (3584 \times 7340032) ]
= (3211264 -1495722360832)
+ i (182583296 + 26306674688)
= -1495719149568 + 26489257984 i
\]

2) FIR Filter Based On Vedic Multiplier

The basic operation in digital signal processing is filtering. This operation is widely used in many electronic devices to cancel part of signal that is redundant or damages the signal.
The digital filter is described by difference equation in time domain and by transfer function in frequency domain. A digital FIR filter response of 4-taps is mathematically represented as,

\[ Y_i = X_i \ast W_i + Y_{i-1} \]

Where \( W_i \) are weights of the filter, \( X_i \) is an input signal, \( Y_i \) is an output signal. The direct structure of a FIR filter is shown in Fig.10.

These structures use many multipliers and adders. Because this work is focused on the speed of a FIR filter, in these 4-tap FIR Filter structure is used. This structure uses four adders and four multipliers and is described in Fig.10. And Fig.19 shows simulation output of FIR Filter using 64x64 bit Vedic Multiplier using VHDL coding.

IV. EXPERIMENTAL RESULTS & ANALYSIS

A. Experimental Results

To show the efficiency of proposed Vedic multiplier at 64 Bit level, it has been compared with conventional multiplier structure based on conventional multiplication algorithms at the 64 Bit level. Fig.11. Shows the Simulation output of 64x64 Bit Conventional Multiplier using VHDL coding.

Fig.10 Shows the direct structure of a FIR filter

Fig.11 Simulation output of 64x64 Bit Conventional Multiplier using VHDL coding

Fig.12. Simulation output of Vedic Multiplier for 2x2 Bit using VHDL coding

Fig.13. Simulation output of Vedic Multiplier for 4x4 Bit using VHDL coding

Fig.14. Simulation output of Vedic Multiplier for 8x8 Bit using VHDL coding

Fig.15. Simulation output of Vedic Multiplier for 16x16 Bit using VHDL coding
B. Result Analysis

<table>
<thead>
<tr>
<th>Type Of Multiplier</th>
<th>Time Taken For Multiplication (in ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reference [1]</td>
</tr>
<tr>
<td>2x2 Bit Vedic Multiplier</td>
<td>--</td>
</tr>
<tr>
<td>4x4 Bit Vedic Multiplier</td>
<td>--</td>
</tr>
<tr>
<td>8x8 Bit Vedic Multiplier</td>
<td>15.418 ns</td>
</tr>
<tr>
<td>16x16 Bit Vedic Multiplier</td>
<td>22.604 ns</td>
</tr>
<tr>
<td>32x32 Bit Vedic Multiplier</td>
<td>31.526 ns</td>
</tr>
<tr>
<td>64x64 Bit Vedic Multiplier</td>
<td>--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type Of Multiplier</th>
<th>Time Taken For Multiplication (in ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64x64 Bit Conventional Multiplier</td>
<td>180.864 ns</td>
</tr>
<tr>
<td>64x64 Bit Vedic Multiplier</td>
<td>49.252 ns</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type Of Multiplier</th>
<th>Time Taken For Multiplication (in ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64X64 Bit Vedic Multiplier</td>
<td>49.252 ns</td>
</tr>
<tr>
<td>64X64 Bit Complex Number Multiplication Using Vedic Multiplier</td>
<td>55.472</td>
</tr>
</tbody>
</table>
In this project we proposed the design of 64x64 Bit Vedic Multiplier based on Urdhva Tiryakhyma Sutra of Vedic Mathematics and the design of 64x64 Bit Conventional Multiplier. The design is based on Vedic method of multiplication the worst case propagation delay in the Optimized Vedic multiplier case is 49.252ns. It is therefore seen that the Vedic multipliers are much more faster than the conventional multipliers. FPGA implementation of this multiplier shows that hardware realization of the Vedic mathematics algorithms is easily possible. The high speed multiplier algorithm exhibits improved efficiency in terms of speed.

The time taken for multiplication operation is reduced by employing the Vedic algorithms. Here integrated Vedic Multiplier architecture is proposed for FIR Filter. FIR filter is based on complex arithmetic due to the magnitude and phase responses of the channel impulse characteristics. So here we designed the 64x64 Bit Complex Number multiplications using Vedic Multiplier and using this Vedic Complex Multiplier we design the high speed FIR Filter as an application of the project. In future this work can be used to design Multiply and Accumulate (MAC) unit, Arithmetic and Logical unit (ALU), Digital Signal Processors (DSP), etc.

REFERENCES


