VHDL Code of Vedic Multiplier with Minimum Delay Architecture

Vaiibhav Jindal, Mr. Navaid Zafar Rizvi, Dinesh Kumar Singh

Abstract—The ever demand in enhancing the ability of process or the time of process will be as low as possible. The delay in the processes is affected by the root delay and the logic delay. The logic delay plays main role in calculating the delay of any process. The logic delay of any process is reduced by using minimum hardware architecture. So which architecture of Vedic multiplier used in the process so the delay is minimum.

Index Terms—Vedic multiplication, VHDL, Hardware Design, Half Adder, Full Adder

I. INTRODUCTION

With the latest advancement of VLSI technology the demand for high speed and portable digital signal processing system. Digital signal processing is present in every engineering discipline. In the fastest growing technology, it poses tremendous challenges to the engineering community to make fastest addition and multiplication. Therefore DSP engineers are constantly looking for new architecture with minimum delay.

Multiplication is the most common and important arithmetic operations having wide applications in different areas of engineering and technology. The performance of any circuit is evaluated mainly by estimating the silicon area and speed (delay). Hence, continuous efforts are being made to achieve the same. In order to calculate the square of a binary number, fast multipliers such as Braun Array, Baugh-Wooley methods of two’s compliment, Booth’s algorithm using recorded multiplier and Wallace trees are in use. Recursive decomposition and Booth’s algorithm are the most successful algorithms used for multiplication. Other methods include Vedic multipliers based on ‘Urdhva Tiryagbhyam’ and the “Duplex” properties of ‘Urdhva Tiryagbhyam’. [1]

Vedic mathematics is the name given to the ancient system of mathematics, which was rediscovered, from the Vedas between 1911 and 1918 by Sri Bharati Krishna Tirthaji. The whole of Vedic mathematics is based on 16 sutras [2]. In this paper we will discuss the possible multiplier architecture of Vedic mathematics with its hardware implementation and VHDL code. A simple concept of digital multiplier (referred henceforth as Vedic multiplier) architecture based on the Urdhva Tiryagbhyam (Vertically and Cross wise) Sutra is presented. [3] The organization of this paper is as follows. Section 1 present the basic logic for 4X4 multiplier block and Section 2, explain how the 4X4 multiplier is implemented with minimum hardware use.

Vaiibhav Jindal, School of ICT, Gautam Buddha University, India
Mr. Navaid Zafar Rizvi, School of ICT, Gautam Buddha University, India
Dinesh Kumar Singh, School of ICT, Gautam Buddha University, India

II. VEDIC MULTIPLICATION

4X4 Multiplier: Let’s take two inputs, each of 4 bits, say A0A1A2A3 and B0B1B2B3. As inputs are of 4 bits the result is of maximum 8 bits, or if the input is of maximum n bits the multiplication is of maximum 2n bits.

\[
\begin{array}{c|c|c|c}
A_3 & A_2 & A_1 & A_0 \\
B_3 & B_2 & B_1 & B_0 \\
\hline
S_7 & S_6 & S_5 & S_4 & S_3 & S_2 & S_1 & S_0 \\
\end{array}
\]

The Vedic Sutras along with their brief meanings are enlisted below alphabetically. [3]

1. (Anurupy) Shunyamanyat -If one is in ratio, the other is zero.
2. ChalanaKalanabhyam -Differences and similarities.
3. EkadhikinaPurvena- By one more than the previous One.
4. EkanyunenaPurvena -By one less than the previous one.
5. Gunakasamuchyah-Factors of the sum is equal to the sum of factors.
6. Gunitasmuchyah-The product of sum is equal to sum of the product.
7. NikhilamNavatashcaramamDashatah -All from 9 and last from 10.
8. ParvartyaYojayet-Transpose and adjust.
9. Puranaparanabhyah -By the completion or noncompletion.
10. Sankalana- vyavakalanabhyam -By addition and by subtraction.
11. ShesanyamkenaCharamen -The remainders by the last digit.
12. ShunyamSaamyasamuccay -When the sum is same then sum is zero.
13. Sopantyadvayamantyam -The ultimate and twice the penultimate.
15. Vyastisamansthit -Part and Whole.

Method1:

This Vedic multiplication method based on one of the sutra of Urdhva Tiryagbhyam. In this method S0 is vertical product of bit A0 &B0, S1 the sum of A0B0, and A1*B2, and tens bit of previous sum (S0). All the multiplication is done with the same process. The ten’s and the hundred’s position bit are forward as the carry to next addition. Only the S1 bit of answer will comes from the tens bit S0 in the 4 bit multiplication process. In the n bit multiplication process the 2n-1 bit of answer is calculated like the same way as S7 bit. The 4 bit
multiplication process with different architectures of same method is shown in next section. The multiplication process shown in the figure.1

For the multiplication of single bit a simple AND gate is used; now only the thing is how to make the addition process with minimum hardware so the delay is as low as possible.

The vedic multiplication of 4-bit is process using the following equations

\[
X=a_3a_2a_1a_0 \\
y=b_3b_2b_1b_0 \\
P_0=a_0b_0 \\
P_1=a_1b_0+a_0b_1 \\
P_2=a_2b_0+a_1b_1+a_0b_1+P_1(1) \\
P_3=a_3b_0+a_2b_1+a_1b_2+a_0b_3+P_2(1) \\
P_4=a_2b_1+a_1b_2+a_0b_3+P_3(1) \\
P_5=a_3b_2+a_2b_3+P_4(1) \\
P_6=a_3b_3+P_5(1) \\
Product=P_6&P_5(0)&P_4(0)&P_3(0)&P_2(0)&P_1(0)&P_0
\]

The architecture for the equations are given the third section of this paper.

Method2:
Another Method for the multiplication of 4-bit number is shows in Figure.2. This method is also based on one of the sutra of UrdhvaTiryakbhyam, In this method the 4-bit number is break in two parts higher and lower order bits and 2x2 bit Vedic multiplication is performed as First 2x2 bit multiplier inputs are A1A0 and B1B0. The last block is 2x2 bit multiplier with inputs A3A2 and B3B2. The middle one shows two 2x2 bit multiplier with inputs A1A0 & B1B0 and A1A0 & B3B2. So the final result of multiplication, which is of 8 bit, S7S6S5S4S3S2S1S0.

To get the multiplication bit (S7S6S5S4S3S2S1S0) need of 8 bit adder. If same process is done with n-bits than the number is break into n/2 bits accordingly and 2n bit adder is used to get final result.

III. HARDWARE IMPLEMENTATION

Method1:
The hardware needed for the method 1 is the AND & OR gates, Half Adder (HA) and Full Adder (FA). The basic flow diagram is shown in the Figure.3. In this approach the delay is (Pd\text{AND}+(5xPd\text{FA})+Pd\text{HA})

\[
Pd\text{FA}=2xPd\text{HA}+Pd\text{OR}
\]

Delay Method1= (Pd\text{AND}+11xPd\text{HA}+5xPd\text{OR}) (1)

Now some how to reduce the delay, the design shown in the Figure 4 is with the less delay than the previous design. In this design the groups of FA are replaced by some HA. So, the delay of the path is affected.
From the equation 1&2 it is clear that propagation delay of the circuit is reduced.

For the method 2 of 4 bit Vedic multiplication the hardware requires is 4x2-bit Vedic multiplier and 3x8 bit adder. The architecture of the method 2 is given by KabirajSethiet all shown in Figure.6.Delay is more as 8 bit adder is used, some delay can reduced by using 8 bit adder still the delay is more so it has to be reduced. The 4-bit Vedic multiplier with less hardware is shown in Figure.7.The delay of first architecture of method 2 is(Pd\text{Vedic mul}+3xPd\text{8bit adder})Where Pd\text{Vedic mul} is delay of 2-bit Vedic multiplier can give as Pd\text{AND}+2xPd\text{HA} and Pd\text{8bit adder} is delay of 8-bit adder given as Pd\text{HA} + 7xPd\text{FA}.The delay of method-2 architecture is given as:-

$$\text{Delay Method 2 Architecture 1} = Pd\text{AND} + 47xPd\text{HA} + 15xPd\text{OR}$$

In the second architecture of method 2 the 8-bit adder is replace by 4-bit adder. 4-bit adder needs the less hardware than the 8 bit added so, the path delay is reduced this architecture. The path delay of architecture 2 is (Pd\text{Vedic mul} + 3xPd\text{4bit adder}). Where Pd\text{Vedic mul} is delay of 2-bit Vedic multiplier given as (Pd\text{AND}+2xPd\text{HA}) and Pd\text{4bit adder} is delay of 4 bit adder can be given as (Pd\text{HA} + 3xPd\text{FA}).The delay of this architecture is given as:-

$$\text{Delay Method 2 Architecture 2} = Pd\text{AND} + 23xPd\text{HA} + 6xPd\text{OR}$$

Now it has to be calculating that which architecture has the minimum path delay. To calculating the path delay hardware description code (HDL) is to be written for all architectures and has to be synthesis in xilinx path delay is the addition of logic delay and root delay.The root and logic delays are separately shown in the Figure.8. The delays of all architecture are shown in Table.1 calculating using Xilinx ISE. In the table the total delay is the addition of root and logic delays.

<table>
<thead>
<tr>
<th>Method</th>
<th>Arch.</th>
<th>Delay Manually calculated</th>
<th>Slice Occupied</th>
<th>Total Delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method1</td>
<td>Arch.1</td>
<td>Pd\text{AND} + 1xPd\text{HA} + 5xPd\text{CR}</td>
<td>20</td>
<td>14.310</td>
</tr>
<tr>
<td>Method1</td>
<td>Arch.2</td>
<td>Pd\text{AND} + 16xPd\text{HA} + 4xPd\text{CR}</td>
<td>19</td>
<td>13.155</td>
</tr>
<tr>
<td>Method2</td>
<td>Arch.1</td>
<td>Pd\text{AND} + 47xPd\text{HA} + 15xPd\text{CR}</td>
<td>20</td>
<td>17.829</td>
</tr>
<tr>
<td>Method2</td>
<td>Arch.2</td>
<td>Pd\text{AND} + 23xPd\text{HA} + 6xPd\text{CR}</td>
<td>19</td>
<td>15.779</td>
</tr>
</tbody>
</table>

From the above table it is clear that Architecture 2 of Method 2 is the fast one and its path delay is less than any other architecture.
REFERENCES


