16 BIT MULTIPLIER IMPLEMENTATION USING VEDIC MATHEMATICS

P Nagaraju, DVS Chandrababu, T Ravindra,
Department of Electronics and Communication Engineering,
MLR Institute of Technology, Dundigal, Hyderabad, Telangana, India

K Narsimha
Department of Electronics and Communication Engineering,
Vardhaman College of Engineering, Shamshabad, Telangana, India

ABSTRACT

Vedic Mathematics is the old methodology of Indian mathematics which as has a unique technique of calculations based on 16 formulas these are adopted from “Vedas”. On account of those 16 formulas, the partial products and sums are generated in one step which reduces the carry propagation from LSB to MSB. The implementation of the Vedic mathematics and their application to the complex multiplier ensure substantial reduction of propagation delay in comparison with DA based architecture and parallel adder based implementation which is most commonly used architectures. The functionality of these circuits was checked and the performance of parameters like propagation delay and dynamic power consumption was calculated by spice specter using standard 90nm Complementary metal oxide semiconductor technology. The propagation delay of the resulting 16-bit complex multiplier is 4ns and consume 6.5mW power. We achieved 25% improvement in speed from earlier complex multipliers, e.g. parallel adder and DA based architectures.

Keywords: Urdhva Tiryakbhyam, Adders, Vedic Mathematics, Conventional Multipliers, Verilog HDL.


1. INTRODUCTION

Many Digital signal processing (DSP) systems include multipliers as one of core hardware blocks. Multipliers hold a significant role in various DSP applications such as digital filtering, digital communication, and Fast Fourier transform. The common classification of multipliers depending on their architecture involves three types: ‘serial multipliers’, ‘parallel multipliers’ and ‘serial-parallel multipliers’. In this paper, multiplier architecture based on Urdhva tiryakbhyam Sutra, a concept based on Vedic mathematics is discussed. [6]
1.1. Urdhva Tiryakbhyam Sutra
The basic Sutras and Urdhva Tiryakbhyam Sutra in the Vedic Mathematics help to do all the numeric computations in easy and fast manner. The Sutra which we employ in this project is Urdhva Tiryakbhyam (Multiplication)[7]

1.2. Description of sutra
This is the general formula applicable to all cases of multiplication, Urdhva Tiryakbhyam means that “Vertically and Crosswise”, which is the method of multiplication followed.

2. DESIGN OF THE 16X16 MULTIPLIER
We can Design 16-bit multiplier using a 2x2 multiplier, 4x4 multiplier and 8x8 multiplier these three multipliers are explained below. In 16-bit multiplier, we use 4, 6, 8, 12, 16 and 24-bit adders.[1]

2.1. 2X2 Multiplier

Figure 2 Multiplication process of two-bit multiplier

Fig-2 illustrates the steps to multiply two 2 bit numbers Converting the figure to a hardware equivalent we have 3 and gates which will act as 2-bit multipliers and two half adders to add the products to get the final product. Here is the hardware detail of the multiplier[2].
Where "an" and "b" are two numbers to be multiplied and "q" is the product. With this design, we are now ready to code this in Verilog easily using gates and HA(half adders). To make the design more modular we try to write code for HA first and then instantiate it to have the final product.[3]

2.2. 4X4 Multiplier

Using 4 such 2x2 multipliers and 3 adders we can build 4x4 bit multipliers as shown in the figure-4. with Proper instantiating of the 2x2 multipliers and adders and We have to first write code for 4bit and 6-bit adders before writing 4x4 multiplier For a simpler design we have used the "+" operator which is supported by the XST synthesis tool which by default selects a low hardware adder. This architecture follows Wallace tree[5] which reduces the addition levels from 3 to just 2 stages as shown in figure-4, Arrangement of the adders and the addition is explained from the figure shown below[4]

2.3. 8X8 Multiplier

Similar to the previous design of 4x4 multiplier, we need 4 such 4x4 multipliers to develop 8x8 multipliers. Here we need to first design 8bit and 12-bit adders and by proper
instantiating of the module and connections as shown in the figure-6, we have designed an 8x8 bit multiplier. At this point of time, it is necessary to verify the RTL code and we have to check the hardware is as per our design or not. Plan Ahead tool by Xilinx gives a better view of the hardware design with design elaborate option. The figure-6 is the addition tree diagram to know the process for 8x8 multiplier using four 4x4 multipliers.

**Figure 6** Design of 8x8 multiplier using 4x4 multiplier

The multiplication process takes place in 8x8 multiplier as shown in below figure.

**Figure 7** Multiplication process of 8x8 multiplier

### 2.4. 16x16 Multiplier

Similar to the previous design of 8x8 multiplier, we need 4 such 8x8 multipliers to develop 16x16 multipliers. Here we need to first design 16bit and 24-bit adders and by proper instantiating of the module and connections as shown in the figure-8, we have designed a 16x16 bit multiplier. At this point of time, it is necessary for verify the RTL code and checks if the hardware is as per design. Plan Ahead tool by Xilinx gives a better view of the hardware design with design elaborate option.

**Figure 8** Design of 16x16 multiplier using 8x8 multiplier
The multiplication process takes place in 16x16 multiplier as shown in below figure-9

![Figure 9](image-url)  
**Figure 9** Multiplication process of 16x16 multiplier

3. SIMULATION AND SYNTHESIS REPORT

3.1. RTL Schematic Diagrams of Vedic Multiplier  
The below figure-10, is the RTL Schematic diagram of 16x16 multiplier using VedicMathematics Here a, b are the inputs of the 16-bit multiplier and c is the output

![Figure 10](image-url)  
**Figure 10** Schematic Diagrams of Vedic Multiplier

The below figure-11, knows as an internal diagram of RTL Schematic in this we have Four 8-bit multipliers and one 16-bit, two 24-bit adders and each 8-bit multiplier we have 4-bit, 2-bit multipliers in internal.

![Figure 11](image-url)  
**Figure 11** RTL Internal diagram of Vedic multiplier
4. SIMULATION RESULT OF VEDIC MULTIPLIER
The below diagram-12, is the simulation result of final output here we have given input 
a=2345 and b=3243 then output c=7604835 this output is verified according to inputs in this 
way Vedic multiplier multiplies 16-bit numbers. The inputs we can give in the binary form 
also with selecting binary format then we have to select output as binary form.

Figure 12 RTL simulation result of Vedic multiplier

4.1. Design summary of Vedic multiplier
The below table is designed summary of the Vedic 16-bit multiplier in the figure shows no errors if errors show 
program will not execute.

Table Design summary of Vedic multiplier

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Gates</td>
<td>395</td>
<td>398</td>
<td>97%</td>
</tr>
<tr>
<td>Number of Inputs</td>
<td>705</td>
<td>725</td>
<td>93%</td>
</tr>
<tr>
<td>Number of Outputs</td>
<td>64</td>
<td>14</td>
<td>45%</td>
</tr>
</tbody>
</table>

5. APPLICATIONS
- The propagation delay of the resulting (16, 16)x(16, 16) complex multiplier is only 4ns and consume 
  6.5 mW power. We achieved almost 25% improvement in speed from earlier reported complex 
  multipliers, e.g. parallel adder and DA based architectures
- Vedic Mathematics covers Arithmetic Decimal operations in all decimal work, Ratios, Proportions, 
  Trigonometry, Percentages, Averages, Interest, Annuities, Discount, the Centre of Gravity of 
  Mechanics, Solid Geometry, Plane Spherical Trigonometry

6. CONCLUSIONS
The proposed Vedic multiplier architecture shows speed improvements over multiplier 
architecture presented in the 16x16 Vedic multiplier along with other lower bit multipliers 
using Urduha Tiryakbhyam” Sutra found to be better in terms of speed and complexity in the 
gate level design architecture. This approach may be well suited for multiplication of numbers 
with more than 16-bit size.
REFERENCES


