(16 BIT X 16 BIT) BOOTH MULTIPLIER USING VHDL

MUHAMMAD SYAFIQ BIN NORASHID

UNIVERSITI MALAYSIA PAHANG
“I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)”

Signature : ______________________________________________

Name : NOR FARIZAN BINTI ZAKARIA

Date : 7 NOVEMBER 2008
(16 BITS X 16 BITS) BOOTH MULTIPLIER USING VHDL

MUHAMMAD SYAFIQ BIN NORASHID

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor of Electrical Engineering (Hons.) (Electronics)

Faculty of Electrical & Electronics Engineering
University Malaysia Pahang

NOVEMBER, 2008
“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : ____________________________

Author : MUHAMMAD SYAFIQ BIN NORASHID

Date : 7 NOVEMBER 2008
To my beloved parents…
who always pray for me and give me courage to finish this thesis

Also, to those people who gave guided and inspired me throughout my journey.
Thank you for the supports and advices that have been given.
ACKNOWLEDGEMENT

In the name of Allah S.W.T, the Most Gracious, the Ever Merciful. Praise is to Allah, Lord of the Universe and Peace and Prayers be upon His final Prophet and Messenger Muhammad s.a.w.

I would like to take this opportunity to sincerely express my highest gratitude to my supervisor madam Nor Farizan Binti Zakaria for her guidance, ideas and advice during this project.

My sincere gratitude also goes to my colleagues for their aid, ideas and advice, and my friends who had helped me directly or indirectly.

Finally, special thanks extended to my beloved family who had given me moral support and prayed for my success.
ABSTRACT

Nowadays, digital device is very important to all the people in this world. The high speed operation and less space and energy required had made the digital devices more preferred. This project is to design digital system which performed fixed point Booth Multiplier where the design system would be developed using hardware description language (HDL), in this case, VHDL (VHSIC Hardware Description Language), VHSIC stands for Very High Speed Integrated Circuit. The Software used would be Xilinx ISE 10.1 which is the software used to designed digital system for Xilinx manufactured FPGA board. The algorithm to design the system is Booth Multiplier Algorithm. The designed digital system will receive two 16 bits input and processes it to create a 32 bits output with the value of the multiplied inputs data value. Finally, it is proven that the system created can calculate and yield a fixed point multiplied output of the input value.
ABSTRAK

TABLE OF CONTENTS

CHAPTER    TITLE                        PAGES

ACKNOWLEDGEMENT                i
ABSTRACT                      ii
ABSTRAK                       iii
TABLE OF CONTENTS             iv
LIST OF TABLES                vii
LIST OF FIGURES               viii
LIST OF ABBREVIATIONS         ix
LIST OF APPENDICES            x

1 INTRODUCTION
1.1 Overview                  1
1.2 Project Objective         2
1.3 Project Scope             2
1.4 Problem Statement         3
1.5 Project Contribution      3
1.6 Thesis Organization       4
# 2 LITERATURE REVIEW

2.1 Introduction 5
2.2 Booth Multiplier Algorithm 6
   2.2.1 Procedure and Example 6
2.3 VHDL 13
   2.3.1 History of VHDL 13
   2.3.2 Levels of Abstraction 14
   2.3.3 Scope of VHDL 15
2.4 Xilinx Integrated Software Environment (ISE) 10.1 15
   2.4.1 Design Entry 16
   2.4.2 Synthesis 17
   2.4.3 Implementation 17
   2.4.4 Verification 17
   2.4.5 Device Configuration 18

# 3 RESEARCH METHODOLOGY

3.1 Introduction 19
3.2 Overall Methodology 19
3.3 Design Methodology 21
3.4 Booth Multiplier Algorithm and Main Component 23
3.5 Overall Basic Block Diagram 25
3.6 Booth Multiplier Design 26
   3.6.1 Data Path Unit 26
   3.6.2 Control Unit 27
3.7 VHDL Code for Booth Multiplier 27
3.8 Booth Multiplier Block Schematic Diagram 28
RESULT AND DISCUSSION
4.1 Introduction 29
4.2 Booth Multiplier Result and Discussion 29
  4.2.1 Control Unit 30
  4.2.2 Overall Circuit 31
4.3 Performance 32

CONCLUSION AND RECOMMENDATION
5.1 Introduction 33
5.2 Conclusion 33
5.3 Recommendation 34
5.4 Costing and Commercialization 34

REFERENCES 37
Appendices A – D 38 - 65
## LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE NO</th>
<th>TITLE</th>
<th>PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Booth Multiplier Algorithm Rules</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Booth Table (Step 1)</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>Booth Table (Step 2)</td>
<td>9</td>
</tr>
<tr>
<td>2.4</td>
<td>Booth Table (Step 3)</td>
<td>10</td>
</tr>
<tr>
<td>2.5</td>
<td>Booth Table (Final)</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>State, next state condition and output of Control Unit</td>
<td>28</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE NO</th>
<th>TITLE</th>
<th>PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Changing value from decimal to binary</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Cycle 1 calculation (Step 2)</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Cycle 2 calculation (Step 2)</td>
<td>8</td>
</tr>
<tr>
<td>2.4</td>
<td>Cycle 3 calculation (Step 2)</td>
<td>8</td>
</tr>
<tr>
<td>2.5</td>
<td>Cycle 4 calculation (Step 2)</td>
<td>9</td>
</tr>
<tr>
<td>2.6</td>
<td>Cycle 1 calculation (Step 3)</td>
<td>10</td>
</tr>
<tr>
<td>2.7</td>
<td>Cycle 2 calculation (Step 3)</td>
<td>11</td>
</tr>
<tr>
<td>2.8</td>
<td>Cycle 3 calculation (Step 3)</td>
<td>11</td>
</tr>
<tr>
<td>2.9</td>
<td>Cycle 4 calculation (Step 3)</td>
<td>12</td>
</tr>
<tr>
<td>2.10</td>
<td>Levels of abstraction in VHDL</td>
<td>14</td>
</tr>
<tr>
<td>2.11</td>
<td>Scope of VHDL in overall spec of electrical system design</td>
<td>15</td>
</tr>
<tr>
<td>2.12</td>
<td>ISE design flow</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>Flow chart of overall methodology of this project</td>
<td>20</td>
</tr>
<tr>
<td>3.2</td>
<td>Flow chart of design methodology of this project</td>
<td>22</td>
</tr>
<tr>
<td>3.3</td>
<td>Flow chart of Booth Multiplier Algorithm</td>
<td>24</td>
</tr>
<tr>
<td>3.4</td>
<td>Overall basic block diagram</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>Simulation waveform of Control Unit</td>
<td>32</td>
</tr>
<tr>
<td>4.2</td>
<td>Simulation waveform of overall circuit</td>
<td>33</td>
</tr>
</tbody>
</table>
# LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>VHDL</td>
<td>Very high speed integrated circuit Hardware Description Language</td>
</tr>
<tr>
<td>ISE</td>
<td>Integrated Software Environment</td>
</tr>
<tr>
<td>FPGA</td>
<td>Filed-Programmable Gate Array</td>
</tr>
<tr>
<td>VHSIC</td>
<td>Very High Speed Integrated Circuit</td>
</tr>
<tr>
<td>US DoD</td>
<td>United States Department of Defence</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines</td>
</tr>
<tr>
<td>TI</td>
<td>Texas Instruments</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronic Engineer</td>
</tr>
<tr>
<td>ASIC</td>
<td>Application-Specific Integrated Circuit</td>
</tr>
<tr>
<td>RTL</td>
<td>Register Transfer Level</td>
</tr>
<tr>
<td>H/W</td>
<td>Hardware</td>
</tr>
<tr>
<td>S/W</td>
<td>Software</td>
</tr>
<tr>
<td>HDL</td>
<td>Hardware Description Language</td>
</tr>
<tr>
<td>CPLD</td>
<td>Complex Programmable Logic Device</td>
</tr>
<tr>
<td>FSM</td>
<td>Finite State Machine</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>LCD</td>
<td>Liquid Crystal Display</td>
</tr>
</tbody>
</table>
## LIST OF APPENDICES

<table>
<thead>
<tr>
<th>APPENDIX</th>
<th>TITLE</th>
<th>PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>VHDL code of each component in Data Path Unit</td>
<td>38 – 48</td>
</tr>
<tr>
<td>B</td>
<td>VHDL code of Control Unit (Finite State Machine)</td>
<td>49 – 58</td>
</tr>
<tr>
<td>C</td>
<td>VHDL code for Booth Multiplier Design</td>
<td>59 – 63</td>
</tr>
<tr>
<td>D</td>
<td>Full block schematic diagram of (16 bit X 16 bit) Booth Multiplier</td>
<td>64 – 65</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Overview

Digital system nowadays became an important system in this modern era. Analog system was replaced by digital system because digital system can do their processes with high speed operation, less space and energy required. This event happen after the big contribution of the digital system which most commonly used no matter in one’s daily life or in industrial field. Due to the crucial developing of digital system, we cannot deny that the system is very important for now and future.

VHDL which stand for “Very high speed integrated circuit Hardware Description Language” is one of the common techniques for the digital system developing process. The technique is done by program using certain software as a platform which also can perform simulation and analysis of the designed system. The designer only need to describe his digital circuit design in textual form which can erased
without the effort to alter the hardware. The programming language is different compared to other programming language such as C++ language.

VHDL is more preferred because this technique can reduce cost and time, easy to troubleshoot, portable, a lot of platform software support the VHDL function and high references availability.

1.2 Project Objective

The objective of this project is to design a digital multiplier using Booth Multiplier Algorithm in VHDL.

1.3 Project Scope

1. The input will be in 16 bits multiply by 16 bits which will produce 32 bits of accurate multiplied answer.
2. The input and output of the system will only process and produce fixed point value.
3. The system also can accept negative value which is called sign number.
4. VHDL (Very high speed integrated circuit Hardware Description Language) is used as the language for the system.
5. All the process will be running using Xilinx ISE 10.1 software which means the process is simulation only without any hardware implementation.
1.4 Problem Statement

As we know, digital system has been used in daily life or industrial field nowadays because of the benefits compared with analog system. Due to crucial developing of digital system, many new complex digital devices had been design. Some of the devices are called microprocessor, microcontroller or microchip. It is very important to have a very high speed performance in all the devices. Multiplier is one of the most important parts in the devices which can affect the performance of the devices.

So, the high speed and efficient multiplier system is important for the designers of microprocessor, microcontroller and others digital devices. As we know, multiplication operation is not hard to do in decimal number. But, to do the operation in binary number (which used in digital system) is very complex operation.

This project is being done to help create a prototype of digital system design that can operate as multiplier operation that would be implemented into microprocessor, microcontroller and other digital devices.

1.5 Project Contribution

A prototype of 16 bits inputs multiply by 16 bits inputs multiplier using Booth Multiplier Algorithm with accurate 32 bits of output. The prototype is using Xilinx ISE 10.1 software as a platform to design the system which using VHDL as the language for designing process.
1.6 Thesis Organization

This thesis is organized into five chapters. The first chapter introduced the overview, objective, scope and contribution of this project.

Chapter 2 present the related reference studied that being used to do this project. Booth Multiplier Algorithm is also introduced in this chapter.

Chapter 3 would explain about the project methodology which clearly explained about how this project is planned and organized in completing the project.

Chapter 4 presents the result for the system designed and discussion of overall result.

In the final chapter, the project research is summarized and the recommendations for future works are presented here. Costing and commercialization also included in this chapter.
CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, all the literature review that is important to this project will be represented. The literature review will include Booth Multiplier Algorithm (with procedure and example), VHDL and Xilinx Integrated Environment (ISE) 10.1 software. The details of those will be discussed in this chapter.
2.2 Booth Multiplier Algorithm

Table 2.1 show that the Booth Multiplier Algorithm Rules, which very important for this project. The algorithm rules give a procedure for multiplying binary integers in signed –2’s complement representation.

<table>
<thead>
<tr>
<th>Xi</th>
<th>Xi-1</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Shift Only</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Shift Only</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Add (-A) &amp; Shift</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Add (A) &amp; Shift</td>
</tr>
</tbody>
</table>

2.2.1 Procedure and Example

The following example will illustrate how to apply the algorithm to get the exact value of multiplication of \(2_{10}\) multiply by \((-4_{10})\).

- Step 1: Making The Booth Table.
  - i) Change the decimal value to binary value. For the negative decimal value, 2’s complement method is used to get the binary value (Figure 2.1).
ii) Let $A = -4_{10} = 1100_2$ (multiplier)
Let $X = 2_{10} = 0010_2$ (multiplicand)

iii) Load $A$ and $X$ value into the Table 2.2. The additional 4 bits (0000) and 4 cycles (operation) is required because $A$ and $X$ are in 4 bits binary value.

Table 2.2: Booth Table (Step 1)

<table>
<thead>
<tr>
<th>A</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

• Step 2: Identify Operation.

i) The Booth Multiplier Algorithm Rules is applied depend on number of cycle.
ii) When cycle 1, value $i$ will be same as the number cycle (Figure 2.2).

**For Cycle 1:**

$\text{cycle 1) = (i = 1)}$

When $i = 1$, $X(1)$ and $X(0)$ is identified using the rules

$X(0)$ is assumed 0

From the rules, when $X(1) = 0$ and $X(0) = 0$,

The operation is **shift only**.

**Figure 2.2:** Cycle 1 calculation (Step 2)

iii) Repeat the step until $i = \text{Bits number (Maximum)}$ as shown in Figure 2.3, 2.4 and 2.5.

**For Cycle 2:**

$\text{cycle 2) = (i = 2)}$

When $i = 2$, $X(2)$ and $X(1)$ is identified using the rules

$X(2) = 1$ and $X(1) = 0$

From the rules, when $X(2) = 1$ and $X(1) = 0$

The operation is **Add (-A) and Shift**.

**Figure 2.3:** Cycle 2 calculation (Step 2)

**For Cycle 3:**

$\text{cycle 3) = (i = 3)}$

When $i = 3$, $X(3)$ and $X(2)$ is identified using the rules

$X(3) = 0$ and $X(2) = 1$

From the rules, when $X(3) = 0$ and $X(2) = 1$

The operation is **Add A and Shift**

**Figure 2.4:** Cycle 3 calculation (Step 2)
For Cycle 4:

\[(\text{cycle 4}) = (i = 4) = \text{Maximum Bits Number}\]

When \(i = 4\), \(X(4)\) and \(X(3)\) is identified using the rules

\[X(4) = 0 \text{ and } X(3) = 0\]

From the rules, when \(X(4) = 0\) and \(X(3) = 0\)

The operation is **Shift only**.

Figure 2.5: Cycle 4 calculation (Step 2)

iv) Load all the identified operation into the table as shown in Table 2.3.

Table 2.3: Booth Table (Step 2)

<table>
<thead>
<tr>
<th>Bits Number ((i))</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>0</td>
<td>(0)</td>
<td>1</td>
<td>(0)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Additional Bits</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shift Only</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Add (-A) &amp; Shift</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Add A &amp; Shift</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Shift Only</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Step 3: Operation Executed.

i) The operation will be done in order from cycle 1 until cycle 4.

ii) Every answer that we get after every operation is done is used as initial data to do the next operation as shown in Figure 2.6, 2.7, 2.8 and 2.9.
For Cycle 1:
The operation is **Shift only**
Assume initial data is 0000
Because there is no answer (data) from the previous operation

![Diagram showing data shifting and additional bits]

**Figure 2.6**: Cycle 1 calculation (Step 3)

iii) Load the data into the table as shown in Table 2.4.

**Table 2.4**: Booth Table (Step 3)

<table>
<thead>
<tr>
<th>Bits Number ($i$)</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>Additional Bits</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operation</th>
<th>Bit</th>
<th>Bit</th>
<th>Bit</th>
<th>Bit</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift Only</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Add ($-A$) &amp; Shift</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cycle 2</td>
</tr>
<tr>
<td>Add $A$ &amp; Shift</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cycle 3</td>
</tr>
<tr>
<td>Shift Only</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cycle 4</td>
</tr>
</tbody>
</table>

iv) Repeat the same step until the four cycles is completed as shown in Figure 2.7, 2.8 and 2.9.
For Cycle 2:
The operation is Add (-A) & Shift
Initial data = 00000 and A = 1100
-A = 0100 (2’s complement)

Initial Data
0 0 0 0 0
-A + 0 1 0 0
Adding
0 1 0 0 0
Shifting
0 0 1 0 0 0

Figure 2.7: Cycle 2 calculation (Step 3)

For Cycle 3:
The operation is Add A & Shift
Initial data = 001000 and A = 1100

Initial Data
0 0 1 0 0 0
A + 1 1 0 0
Adding
1 1 1 0 0 0
Shifting
1 1 1 1 0 0 0

Figure 2.8: Cycle 3 calculation (Step 3)