CHAPTER 1

INTRODUCTION

1.1 Project Background

High speed multiplication is another critical function in a range of very large scale integration (VLSI) applications. Multiplications are expensive and slow operations [5]. The performances of many computational problems are often dominated by the speed at which a multiplication operation could be executed [5]. For the purpose of this final year project (FYP), the Wallace Tree multiplier was studied as one of the high speed multiplier designs between all types of multiplier.

The scopes of this project consist of three parts, which are to collect and gather as much information about multipliers especially about the Wallace Tree Multiplier, analyze and construct the schematic, plus designing the source code of the high speed 8-bits x 8-bits Wallace Tree multiplier. From the information gathering, books, journals, webs, and other sources are taken as the references for this project. Next, for the second part, the schematic or the circuit structure for the Wallace Tree multiplier has been analyzed and constructed. The analysis covered the characteristic of the multiplier such as speed, area, and delay. This part also will complete the major objective of this FYP that is to study the 8-bits x 8-bits Wallace Tree multiplier as one of the high speed multiplier of all multiplier types. For the final part of the project, the source code of the high speed 8-bits x 8-bits Wallace Tree multiplier is constructed.

1.2 Multiplication

Multiplication is an important basic arithmetic operation and less common operation than addition, but it is still essential for microprocessors, digital signal processors and graphic engines [1]. Multiplication algorithms are used to illustrate methods of designing different cells so that they fit into a larger structure. Multiplication is logically carried out by a sequence of addition, subtraction and shift operations. It is an expensive and slow operation. VLSI designers have recognized this by dedicating significant area to integer and floating point multipliers. Therefore, high speed multiplication can be achieved by having a high speed multiplier.

1.3 Multiplier

Multipliers are, in effect, complex adder arrays [5]. From the Wikipedia encyclopedia [19], in digital design, a multiplier is a hardware circuit dedicated to multiply two binary values. A variety of computer arithmetic techniques can be used to implement a digital multiplier. Most techniques involve computing a set of partial products, and then summing the partial products together. This process is similar to the method taught to primary school children for conducting long multiplication on base-10 integers, but has been modified here for application to a base-2 (binary) numeral system.

Multiplier design starts with the elementary school algorithm for multiplication [8]. Consider the simple example below:

				0	1	1	0	Multiplicand
			x	1	0	0	1	Multiplier
				0	1	1	0	
		+	0	0	0	0		
			0	0	1	1	0	Partial products
	+	0	0	0	0			
		0	0	0	1	1	0	
+	0	1	1	0				
	0	1	1	0	1	1	0	Product

At each step, we multiply one digit of the multiplier by the full multiplicand; we add the result, shifted by the proper number of bits, to the partial product. When we run out of multiplier digits, we are done [8].

Single-digit multiplication is easy for binary numbers – binary multiplication of two bits is performed by the AND function. The computation of partial products and their accumulation into the complete product can be optimized in many ways, but an understanding of the basic steps in multiplication is important to a full appreciation of those improvements [8].

A high speed multiplier is an electronic computing unit used to provide multiplication processes at a very high speed by incorporating various types of logic circuit. In the high speed multiplier, a partial product is obtained from the initial addition of the multiplicand and the multiplier. Then, the partial product are added together in order to obtain the final product.

1.3.1 Multiplier Topologies

There are two types of multiplier topologies which are regular or serial topology and irregular or parallel topology. The summation of the partial products is done using some variation of a carry-save adder, referred to generally as counters [1]. These counters can be connected by several different methods. As used here, topology refers to implementation differences in the way the counters are connected; the allowable number of wires per wiring channel; and the length of the wires required to connect the counters [1].

1.3.1.1 Regular Topologies

In a regular or serial topology, the counters are connected in a regular pattern that is replicated. The regular connections make the design of the partial product array a hierarchical design. A serial multiplier works in a manner similar to manual multiplication of two decimal numbers, although two binary numbers are multiplied in this case. The only difference between this serial multiplier and manual multiplication is the repeated addition of each multiplicand-multiple, instead of one-time addition of all multiplicand-multiples at the end.

The regular topologies are the topologies most commonly used in custom design, since they provide a compromise between optimization and design effort [1]. The regularity allows designers to build a small group of building blocks that contain connected counters and compressors and then connect these blocks to form the topology. For the regular topologies, the maximum number of counters in series defines the delay of the topology. Regular topologies can be classified as either array topologies or tree topologies [1].

In array topologies, the counters are connected mostly serially [1]. The array topology is a two-dimensional (2D) structure that fits nicely on the VLSI planar process.

There are several possible array topologies, including simple, double, and higher-order arrays [1].

Tree topologies are very fast structure for summing partial products. In a tree, counters are connected mostly in parallel [1]. Although trees are faster than arrays, they both use the same number of counters to reduce the partial products. The difference is in the interconnections between the counters. Trees create a 3D structure. However, integrated circuits are planar; these trees must be flattened to fit in the 2D plane. The flattening is achieved by placing the counters and compressors linearly [1].

1.3.1.2 Irregular Topologies

In an irregular or parallel topology, the counters are connected in order to minimize the total delay, disregarding the ease of laying out the multiplier [1]. They do not have a regular pattern for connecting the counters. There are several types of parallel multipliers in digital design with different speeds, areas, and configurations such as Bough-Wooley array multiplier, Braun array multiplier, Modified Booth multiplier, Wallace Tree multiplier, Dadda multiplier and Ferrari – Stefanelli multiplier. The Bough-Wooley array multiplier was developed to perform two's complement multiplication [6]. It is implements with a full adder tree. The Modified Booth algorithm has been extensively used in multipliers with long operands (> 16 bits). Its principle is based on recoding the two's complement operand to reduce the number of partial products to be added. The reduction of partial products improves the performance of the multiplier [9]. The Wallace Tree multiplier is a high speed multiplier. It is a full adder structured specially for a quick addition of the partial products. The Dadda multiplier operation is similar to that of Wallace Tree multiplier. The Ferrari – Stefanelli multiplier is also called "nests" multiplier as the 2-bit sub-multipliers in the main multiplier, reduce the number of partial products [7]. Although there are many types of parallel multiplier, this project will mainly focusing on the study of the Wallace Tree multiplier as one of the high speed multiplier of all it kinds.