CHAPTER 1

INTRODUCTION

1.1 Background History

Adders are commonly found in the critical path of many building blocks of microprocessors and digital signal processing chips. Adders are essential not only for addition, but also for subtraction, multiplication, and division. Addition is one of the fundamental arithmetic operations. A fast and accurate operation of a digital system is greatly influenced by the performance of the resident adders. The most important for measuring the quality of adder designs in the past were propagation delay, and area.

In array processing and in multiplication and division, multioperand addition is often encountered. More powerful adders are required which can add many numbers instead of two together. The design of a high-speed multioperand adder called Carry-Save Adder (CSA). A ripple carry adder turns into a carry-save-adder if the carry is saved (stored) rather than propagate. The name ‘carry save’ arises from the fact that we save the carry-out word instead of using it immediately to calculate a final sum. The principal idea is that the carry has a higher power of 2 and thus is routed to the next column. Carry save adder is ideal to add several operands together. Thus, it can prevent time-consuming carry propagation and speed up computation. Effort the past shows that 16-bit CSA is the fastest adder within another adders [Chetana Nagendra, Robert Michael Owenz, and Mary Jane Irwin, 1996].

1.2 Application of CSA

Instead of waiting for the carry propagation of the first addition, the idea here is to overlap the carry propagation of the first addition with the computation in the second addition, and so forth, since repetitive additions will be performed by a multioperand adder. After the last addition, the carry propagation delay is then unavoidable and it should be included in the total delay time.

When three or more operands are to be added simultaneously, using two operand adders, the time consuming carry-propagation must be repeated several times. If the number of operands is k, then carry has to propagate (k-1) times. Several techniques for multiple operand addition that attempt to lower the carry propagation have been proposed and implemented.

In the carry-save addition, let the carry propagate only in the last step, while in all the other steps we generate a sum and a sequence of carries separately. Thus, the basic carry save adder (CSA) accepts three n-bit operands and generates two n-bit results, an n-bit sum, and an n-bit carry. A second CSA accept these two bit sequences and another input operand, and generates a new sum and carry. A CSA is therefore, capable of reducing the number of operands to be added from 3 to 2, without any carry propagation.

The sum and carry can then be recombined in a normal addition to form the correct result. This process may seem more complicated and pointless in the above trivial example, but the power of this technique is that any amount of numbers can be added together in this manner. It is only the final recombination of the final carry and sum that requires a carry propagating addition.

The fastest method to add a large set of n-bit inputs using CSAs involves structuring the CSAs into a 3:2 tree structure (i.e., an inverted tree structure in which each internal node has three inputs and two outputs, and the number of levels in the tree is the minimum possible).