A SURVEY REPORT ON CARRY SAVE ADDER

NANCY ANAND 17050
dtjnancy@gmail.com

IIIT KALYANI,WEST BENGAL

September 30, 2018

Abstract

Carry-save-adder (CSA) is one of the most widely used types of operations for implementing fast computations of arithmetics in industry. It was reported that typical arithmetic computations found in industry designs were optimized using CSAs, producing designs with on average over 20 percent faster timing (without any area penalty). Being stimulated by the results, we analyze and formulate the problem of allocating CSAs for an arithmetic expression, and present an effective algorithm which constructs a functionally equivalent CSA tree with a minimal timing. Specifically, we solve the CSA allocation problem in two steps:

(1) allocating a delay-optimal CSA tree for the multi-bit inputs of the arithmetic expression and
(2) determining the assignment of the single-bit inputs to carry inputs of CSAs which leads to a minimal increase of delay of the CSA tree obtained in step (1).
For a number of arithmetic expressions found in real designs, it was demonstrated that our approach is very effective and produces designs with up to 54 percent faster timing and up to 42 percent smaller area.

1 INTRODUCTION

A carry-save adder is a type of digital adder, used in computer micro architecture to compute the sum of three or more n-bit numbers in binary. It differs from other digital adders in that it outputs two numbers of the same dimensions as the inputs, one which is a sequence of partial sum bits and another which is a sequence of carry bits. In large-integer arithmetic, addition is a very rare operation, and adders are mostly used to accumulate partial sums in a multiplication.

A carry save adder consists of a ladder of stand-alone full adders. The n-bit CSA consists of n disjoint full adders (FAs) where each of which computes a single sum and carry bit based on the corresponding bits of the three input numbers. It consumes three n-bit input integers to be added and produces two outputs, n-bit partial sum and n-bit carry. Unlike the normal adders such as ripple carry adder, a CSA consists of multiple one-bit full adders without any carry chaining. Carry save adder also known as (3, 2) counter where the addends are three. 3:2 counter can be used to speed up the summation of three or more addends. It sums three 4-bits inputs, and returns the result as two 4-bits output.

A CSA is a different thing altogether. A CSA, instead of trying to solve an addition problem, it solves a different problem. All a CSA does is convert the problem of adding three numbers together into a problem of adding two numbers together.

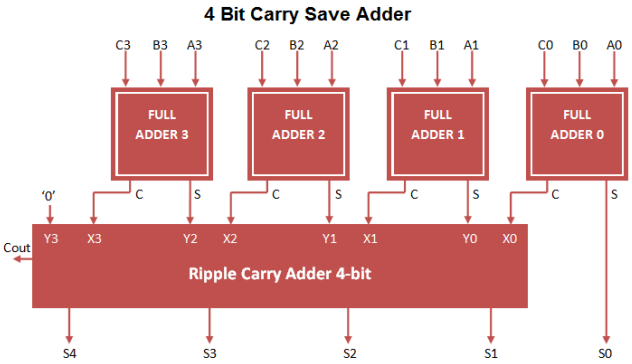

Figure 1: 4-BIT CARRY SAVE ADDER

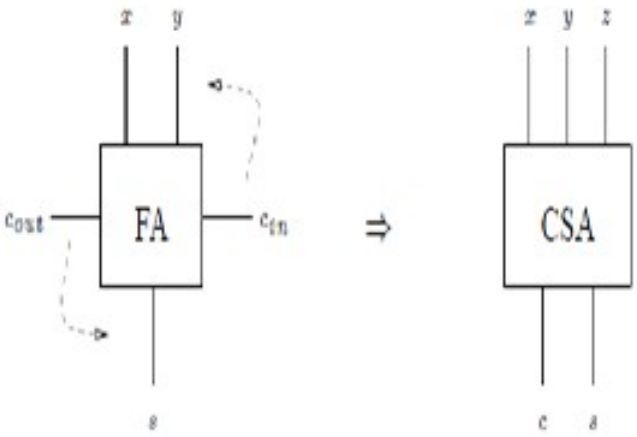

Figure 2: BLOCK DIAGRAM OF 1 BIT CARRY SAVE ADDER

2 PAST DEVELOPMENT OF CARRY SAVE ADDER

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, multi operand 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 CarrySaveAdder (CSA). A ripple carry adder turns into a carry-save-adder if the carry is saved (stored) rather than propagate.

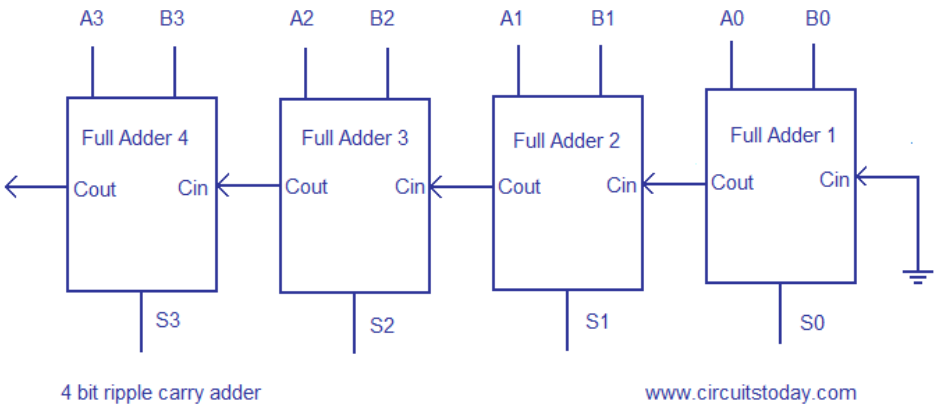

Figure 3: Block diagram of ripple carry adder

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. Efforts in the past show that 16-bit CSA is the fastest adder within another adders [Chetana Nagendra, Robert Michael Owenz, and Mary Jane Irwin, 1996].

3 CURRENT STATUS

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).

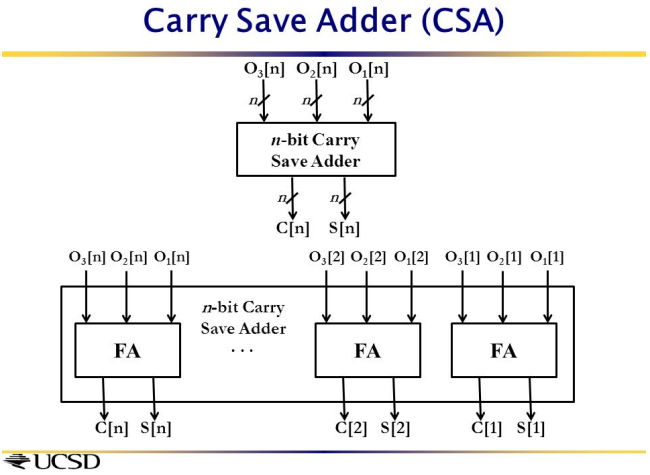

Figure 4: Block diagram of n-bit carry adder

4 CONCLUSION

In this survey report on carry save adder, it has been efficiently and in great detais discussed, the basics concept of carry save adder, how it converts the problem of addition of three numbers into a problem of addition of two numbers and then solves it. It's structure has also been clearly depicted through different block diagrams, for example, a 4-bit carry save adder, 1-bit CSA and in genral, also the n-bit carry save adder. the basic principle of operation of carry save adders is that it take 3 n-bit integers as input and produces two outputs, namely, n-bit partial sum and a caryy output. It's usefulness in multi operand addition makes it applicable in array processing and in multiplication and division. One of the major advantages of this type of adders include that it can prevent time-consuming carry propagation and speed up computation.

5 REFERENCES

THEORY RESOURCES:
1."Carry Save Adder",https://en.wikipedia.org/wiki/Carry-save-adder
2.A Hashmi, "Introduction to history and applications of Carry save adder",http://dspace.unimap.edu.my/
3.A Hashmi, "Literature of carry save adder", http://dspace.unimap.edu.my/dspace/bitstream/123456789/
4."Utilization of carry save adder in arithmetic operation",Junhyung Um,Taewhan Kim, https://ieeexplore.
IMAGE RESOURCES:

1."n-bit carry save adder",https://slideplayer.com/slide/7043635/
2."1-bit carry save adder", https://www.ijser.org/paper/Design-of-Braun-Multiplier-with-Kogge-Stone-Adder-Its-Implementation-on-FPGA.html
3."4-bit carry save adder", https://allaboutfpga.com/carry-save-adder-vhdl-code/