



Testing Arithmetic Circuits: Full Adder Based Circuits

Rama Murthy Garimella and C.V. Vismay

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

December 6, 2024

TESTING ARITHMETIC CIRCUITS: FULL ADDER BASED CIRCUITS

Garimella Rama Murthy, Vismay C V,
Department of Computer Science,
Mahindra University, Hyderabad, Telangana, India

ABSTRACT

In this research paper, it is reasoned that for a single full adder (1-bit full adder), the sum and carry outputs are permutation invariant in the three input variables. Further, for 2-bit full adder and arbitrary N-bit full adder, the sum and carry outputs are partially invariant in the input variables. These results enable reducing the number of input combinations (from truth table) for which the 1-bit/N-bit full adder needs to be tested (for correctness)

1. INTRODUCTION:

With the advent of Integrated Circuit (IC) design based on packing transistors and other electronic components, manufacturers were led to Small Scale Integration (SSI), Medium Scale Integration (MSI), Large Scale Integration (LSI) and Very Large Scale Integration (VLSI) schemes. Particularly, digital integrated circuits were commercially very successful. Due to problems in manufacturing process, sometimes the final IC products were found to be defective. But, it is very clear that the defective digital IC packages will drastically affect the success of market penetration. Hence, manufacturers focused onto methods for testing digital ICs.

Exhaustive testing of any combinational circuit with 'N' inputs requires $(2^N)T$ seconds, if 'T' seconds are required for testing the output of circuit for any one combination of Boolean inputs (i.e. any row of the associated truth table). Hence, the time complexity of exhaustive testing in this case is exponential in 'N'. Thus, researchers focused efforts on isolating FAULT MODELS which were realistic and the IC testing based on them is efficient. Other innovative approaches such as Design for Testability (DFT) and Built-in-Self Test (BIST) were also proposed.

Motivated by the problem of efficient digital IC testing, the authors focused on testing arithmetic circuits such as 1-bit full adder, 2-bit full adder and so on. This research paper is organized as follows. In Section 2, Boolean functions and truth table of 1-bit and 2-bit full adders are given. In Section 3, it is reasoned that the outputs of 2-bit/N-bit full adder are invariant under PARTIAL permutation of input variables. The research paper concludes in Section 4.

2. BOOLEAN FUNCTIONS AND TRUTH TABLES FOR 1-BIT AND 2-BIT FULL ADDERS:

This section provides the Boolean expressions and truth tables for both 1-bit and 2-bit full adders.

0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	0
0	0	1	1	0	1	0	0
0	0	0	0	1	0	1	0
0	0	0	1	1	1	0	0
0	0	1	0	1	1	1	0
0	0	1	1	1	0	0	1
0	1	0	0	0	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	1	0
0	1	1	1	0	0	0	1
0	1	0	0	1	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	0	1
0	1	1	1	1	1	1	1
1	0	0	0	0	0	1	0
1	0	0	1	0	1	0	0
1	0	1	0	0	1	1	0
1	0	1	1	0	0	0	1
1	0	0	0	1	1	0	0
1	0	0	1	1	0	1	1
1	0	1	0	1	0	0	1
1	0	1	1	1	1	1	1
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	0	0	1
1	1	1	1	0	1	1	1
1	1	0	0	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	0	1	1	0	1
1	1	1	1	1	0	1	1

3. SYMMETRY AND PARTIAL SYMMETRY RESULTS OF CASCADE OF FULL ADDERS

3.1 Symmetry in Boolean functions:

A Boolean function is said to be symmetric if the outputs remain invariant when the inputs for that function are permuted. A function $f(x_1, x_2, x_3, \dots, x_n)$ is said to be symmetric if the output of this function remains unchanged even when the order of inputs are changed or swapped, i.e., for example, a function having three inputs x_1, x_2, x_3 , can be written as $f(x_1, x_2, x_3) = f(x_2, x_3, x_1) = f(x_3, x_1, x_2)$. This property presents us with the possibility of

reducing the number of test cases to examine to a certain extent instead of exhaustive testing of the combinations of inputs for the IC's.

A 'n' input function of a combinatorial circuit would generate a total of 2^n input combinations. This number of combinations can be reduced from exponential to a linear complexity leveraging the concept of symmetry in Boolean functions. It would become sufficient to test for 'n+1' set of unique combinations based on the number of ones(1's) in the input to validate the function.

3.2 Symmetry in 1-bit Full Adder:

A 1-bit Full Adder takes in 3 inputs (A, B, C_{in}) and produces 2 outputs namely Sum(S) and Carry(C). The Boolean functions for these 2 output terms Sum ($S = A \oplus B \oplus C_{in}$) and Carry ($C_{out} = A.B + C_{in}(A \oplus B)$). These two functions exhibit full symmetry and thus simplifies analysis and testing requiring only (n+1) i.e., (3+1 = 4) unique input combinations (based on the number of ones in the input) instead of eight.

3.3 Symmetry in 2-bit Full Adder:

Extending to a 2-bit Full Adder that can be represented as a serial combination of two 1-bit Full Adder, which takes in a total of five inputs ($A_1, B_1, A_0, B_0, C_{in}$) and produces three outputs, Sum ($S_0 = A_0 \oplus B_0 \oplus C_{in}$), Sum ($S_1 = A_1 \oplus B_1 \oplus C_0$) and a final Carry ($C_1 = A_1.B_1 + C_0(A_1 \oplus B_1)$) along with an intermediate carry generated by the first Full Adder Carry ($C_0 = A_0.B_0 + C_{in}(A_0 \oplus B_0)$) that goes in as the C_{in} for the second Full Adder.

While the Boolean function of the outputs generated by the individual 1-bit Full Adders in the combination retain the symmetric properties with respect to their individual inputs, the hierarchical flow of information from the first 1-bit Full Adder to the second through the intermediate carry function introduces dependencies between the subsets of inputs and breaks the overall symmetry of the circuit.

For example, the final Carry ($C_1 = A_1B_1 + [A_0.B_0 + C_{in}(A_0 \oplus B_0)] (A_1 \oplus B_1)$) expressed in terms of the least significant bits/initial input values tend to not obey the concept of full symmetry. For an example input (A_1, B_1, A_0, B_0) = (0,1,0,0) [C_{in} has been dropped for the example purpose] would give us Sum (S_0) = 0, Carry (C_0) = 0, Sum(S_1) = 1 and final Carry (C_{out}) = 0. According to the concept of symmetry in Boolean functions, changing the order of input variables should still give the same result. This property does not hold good when we permute the input combinations by providing (0,0,1,0) as the inputs. This input combination generates Sum (S_0) = 1, Carry (C_0) = 1, Sum(S_1) = 1 and final Carry (C_{out}) = 1.

3.4 Introducing Partial Symmetry and Testing for Partial Symmetry

Even though we observed that complete symmetry gets violated in the case of a 2-bit Full Adder, we can observe partial symmetry in function at different parts of the circuit. Partially symmetric function can be defined as a function whose subsets of inputs exhibit symmetry internally but show asymmetry when it interacts with other subsets.

The example provided in the previous subsection contains two partially symmetric subsets (A_1B_1) [most significant bits] and (A_0B_0) [least significant bits] where the inputs are symmetric among themselves but not between each other. This can be verified by just permuting the input values within the subset. For example, inputs (A_1, B_1, A_0, B_0) can be tested for partial symmetry if we permute the input values within the same subset (A_1B_1) to be (0,1,0,0) and (1,0,0,0,) which would give us Sum (S_0) = 0, Carry (C_0) = 0, Sum(S_1) = 1 and final Carry (C_{out}) = 0. This holds true for the other subset (A_0B_0) as well.

If there is no scope for full symmetry in the Boolean functions for which the sufficient number of checks would be $(n+1)$ for an 'n' input variable function, we can still keep the number of tests that could be performed significantly lower by checking for the existence of partial symmetry in the circuit. The number of tests required to be deemed sufficient can be formulated as $(n_1 + 1) \times (n_2 + 1) \times \dots \times (n_M + 1)$ number of test cases where n_1, n_2, \dots, n_M represents the number of inputs present inside partially symmetric subsets and M represents the number of partial subsets present.

For a 2-bit adder which showcased partial symmetry, we can observe two partially symmetric subsets where $n_1 = 3$ and $n_2 = 2$. The minimum number of sufficient test cases that could be performed considering partial symmetry would be $(3+1) \times (2+1) = 12$ tests which is a drastic reduction in number compared to the total 32 combinations that had to be checked if tested exhaustively.

4. CONCLUSIONS:

In this research paper, it is proved that for testing a 1-bit/N-bit full adder, only a small number of input {0,1} combinations (from the truth table) can be utilized to test if such a combinational circuit is functioning properly. The results are currently generalized to other interesting combinational circuits

REFERENCE:

1. M. Morris Mano, "Computer System Architecture," Pearson Publishers, Third Edition, 2007