On the Adders with Minimum Tests

Seiji Kajihara and Tsutomu Sasao

Dept. of Computer Science and Electronics, Kyushu Institute of Technology

Abstract

This paper considers two types of n-bit adders, ripple carry adders and cascaded carry look-ahead adders, *with minimum tests for stuck-at fault models. In the first part, we present two types of full adders consisting of five gates, and show their minimality. We also prove that one of the full adders can be tested by only three test patterns for single stuck-at faults. We also present two types of 4-bit carry look-ahead adders and their minimum tests. In the second part, we consider the tests for the cascaded adders, an n-bit ripple carry adder and a 4m-bit cascaded carry look-ahead adders. These tests are considerably smaller than previously published ones.*

1. Introduction

Adders are used in almost all kinds of arithmetic processing units. The basic unit in an adder is a full adder. Fig. 1A and 1B show two realizations of a full adder using EXOR gates. By cascading the full adders, we have a ripple carry adder shown in Fig. 2. This kind of networks is called an iterative logic array [1]. In terms of iterative logic array, a ripple carry adder is referred to as an array, and the full adder is referred to as a cell. The fault assumption in a cell of the iterative logic array is that a faulty cell can change its behavior in any arbitrary way, as long as it remains a combinational circuit. In a single cell fault model (i.e., at most one cell may be faulty), any n-bit ripple carry adder can be tested by eight tests independently of the value of n [2]. In a multiple cell fault model (i.e., more than one cells can be faulty at the same time), any n-bit ripple carry adder can be tested by 11 tests independently of the value of n [2]. The tests for the cascaded adders where each cell is p-bit adder has been also considered in [2]. In this case, the size of the test is $3 \times 2^{p-1}$.

The ripple carry adders are simple and useful when n is small. For example, INTEL 8080 micro-processors used ripple carry adders [3]. However, when n is large, ripple carry adders are slow since the maximum carry propagation time is proportional to n. Thus, carry look-ahead adders are often used for high-speed addition. Fig. 3A and 3B show two realizations of a 4-bit carry look-ahead adder. Becker [4] showed that 6 ($\log_2 n$) - 4 tests are sufficient to test the n-bit carry look-ahead adder, for single stuck-at faults in a given set of basic cells.

In this paper, we consider two types of n-bit adders with minimum tests for stuck-at fault models: Ripple carry adders and cascaded carry look-ahead adders. First, we show the minimality of full adders shown in Fig. 1. They contain the minimum number of gates among adders consisting of only 2-input gates. We also show that the sizes of the minimum tests for single stuck-at faults in the full adders of Fig. 1A and 1B are five and three, respectively. To our knowledge, the full

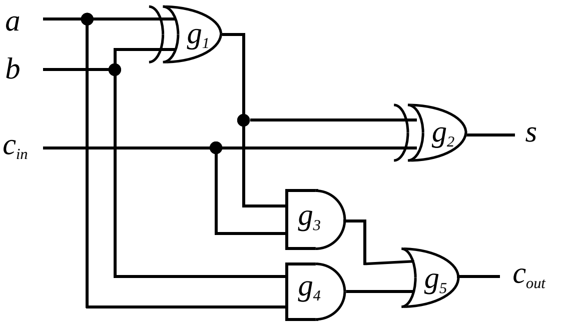

Fig. 1A: Full adder using AND, OR, EXOR gates

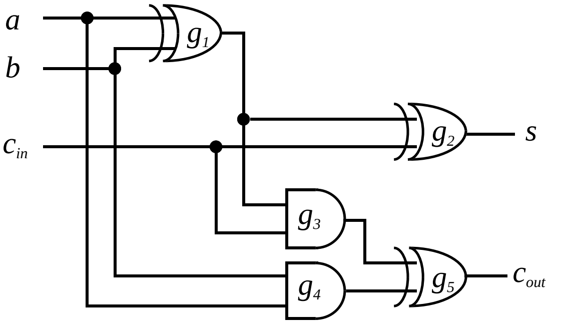

Fig. 1B: Full adder using AND, EXOR gates

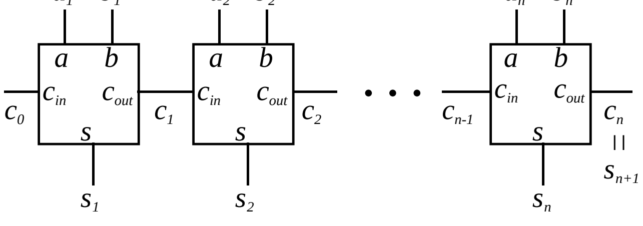

Fig. 2: Ripple carry adder

This research is supported in part by the Ministry of Education, Culture, Science and Sport of Japan.

Proceedings of the 5th Asian Test Symposium (ATS '97)
1081-7735/97 $10.00 © 1997 IEEE

adder in Fig. 1B has the smallest test size. Next, we show that the sizes of the minimum tests for single stuck-at faults in the 4-bit carry look-ahead adders shown in Fig. 3A and 3B are 10 and 12, respectively. Then, we consider the tests for the cascaded adders. In the case of an *n*-bit ripple carry adder, the size of the minimum tests for single faults are the same as that of the full adder, and all multiple stuck-at faults can be detected by six tests. In the case of 4*m*-bit cascaded carry look-ahead adders, we show that the size of the minimum tests are less than 12 for single stuck-at faults. Note that the size of the tests is constant, and is independent of the value of *n* and *m*. These tests are considerably smaller than that of [2], where 3×2^8-1=767 tests are necessary.

2. Adders and their tests

2.1 Minimum full adder

First, we consider two different realizations of full adders, Fig 1A and 1B, both consist of five 2-input gates.

Theorem 1. Consider the network consisting of only 2-input gates. The full adders shown in Fig. 1 contain the minimum number of gates.

(Proof) Here, we will show that there is no full adder with four or fewer 2-input gates. Since we need two gates for sum and carry, the full adder must have the structure as shown in Fig. 4. By considering all possible networks, we can show that it is impossible to realize a full adder. Thus, the minimum adder consists of at least five 2-input gates. (Q.E.D.)

The full adder shown in Fig. 1A is more popular [5] than one in Fig. 1B that contains only AND and EXOR gates. Without using EXOR (EXNOR) gates, we need more gates to realize a full adder. For example, a full adder using eight 2-input gates is shown in [6].

2.2 Tests for the full adder

Under the assumption of the single stuck-at fault model, there is no redundant fault in both of the full adders in Fig. 1A

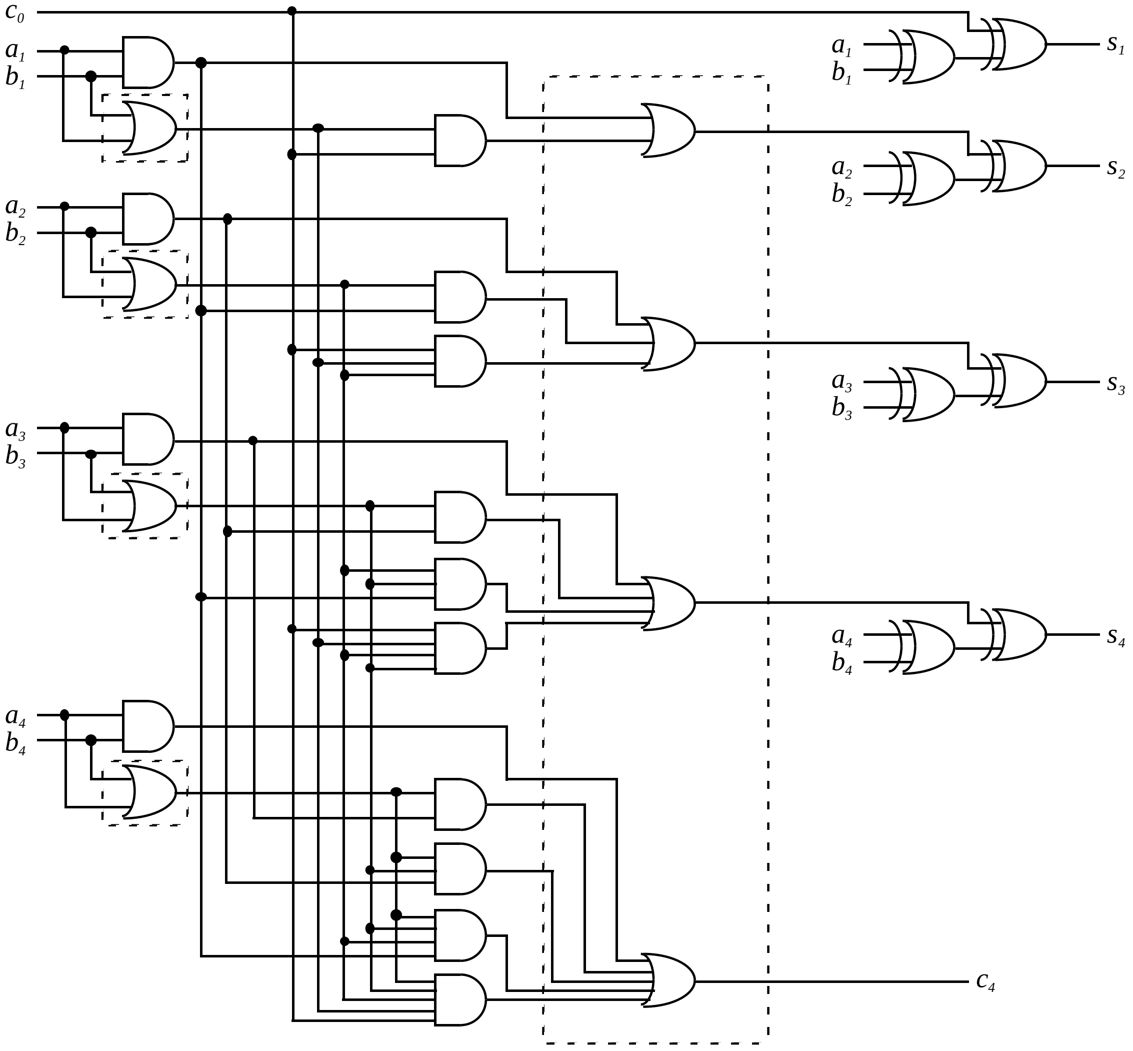

Fig. 3A: 4-bit carry look-ahead adder using AND, OR, EXOR gates

1081-7735/97 $10.00 © 1997 IEEE

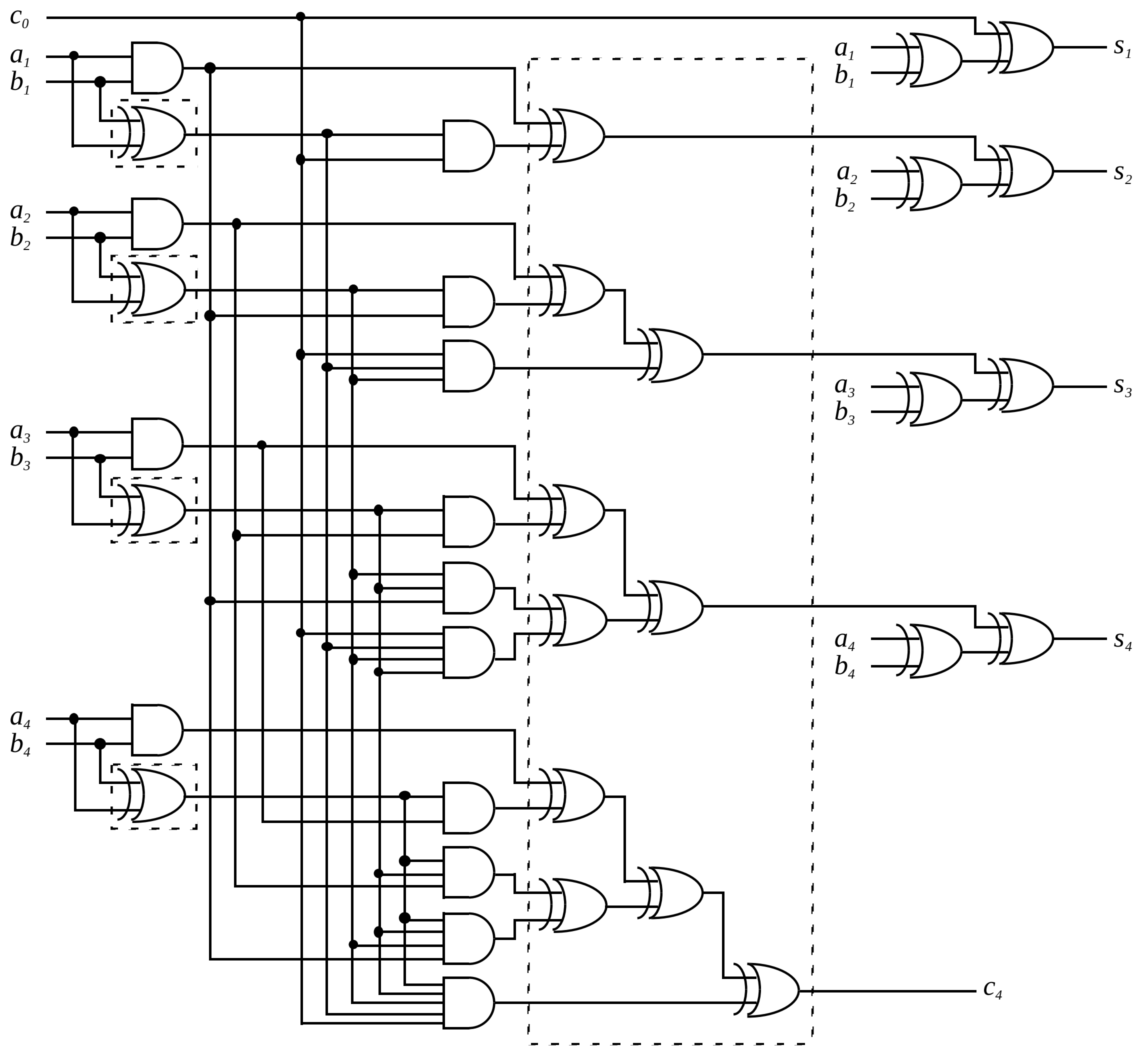

Fig. 3B: 4 bit carry look-ahead adder using AND, EXOR gates

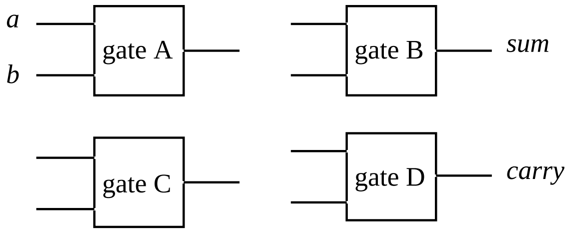

Fig. 4: Proof of minimality of full adder

and Fig. 1B. All the faults can be detected by the tests in Table 1A and Table 1B, where $x, y \in \{0, 1\}$.

Theorem 2. Table 1A and Table 1B give minimum test sets for single stuck-at faults in the full adders in Fig. 1A and Fig. 1B, respectively.

(Proof) For the adder in Fig. 1A, consider the fault set {g_1_g_3/1, a_g_4/1, b_g_4/1, g_4/1, g_3/1}, where g_1_g_3/1 denotes the stuck-at 1 fault on the fanout branch of gate g_1 to gate g_3. Table 2 shows necessary assignments for each fault which are a logic value on signal lines necessarily to detect the fault. Since necessary assignments for any pair of faults have a conflict, it is impossible to generate a test that detects them at the same time. Therefore, the fault set is an independent fault set [7] , and at least five tests are necessary to detect all faults. On the other hand, five tests in Table 1 detect all the faults. Thus, the test set in Table 1A is the minimum for the circuit in Fig. 1A. For the adder in Fig. 1B, at least three tests are necessary to detect all faults because there are 2-input gates. Hence the test set in Table 1B is the minimum for the circuit in Fig. 1B. (Q.E.D.)

To our best knowledge, Fig. 1B is the full adder with the smallest test size. Other full adders using more gates require more tests.

2.3 Carry Look-Ahead Adders

Various carry look-head adders exist. Among them, one shown in Fig. 3A [3] is popular. However, to investigate the testability, we will also consider two other realizations.

As the first variation, consider the circuits where all the OR gates in the inputs (i.e., ones realizing $a_i \vee b_i$) are replaced by EXOR gates. It also works as an adder [8]. As the second variation, consider a circuit in Fig. 3B. This is obtained by replacing all the OR gates in Fig. 3A by two-input EXOR gates. Note that it consists of ANDs and EXORs. This also works as an adder as shown below:

Theorem 3. The circuit where all the OR gates in Fig. 3A are replaced by EXOR gates also works as an adder.
(Proof is in [12])

One might think of yet another variation of the adder: A realization derived by replacing the OR gates in the large dashed box in Fig. 3A with EXOR gates (i.e., OR gates in the small dashed boxes remain), unfortunately, it does not work as an adder.

2.4 Tests for the Carry Look-Ahead Adders

We used a compact test generator [9] to generate a minimal test set and to find a maximal independent fault set of the carry look-ahead adders. For the adder in Fig. 3A, an independent fault set with 10 faults was found, and 10 tests to detect all single stuck-at faults as shown in Table 3A were generated. Thus, the 10 tests form a minimum test set for the adder in Fig. 3A.

Table 1A: Tests for the full adder in FIg. 1A

	In			Out	
test	a	b	c_{in}	s	c_{out}
t_1	0	0	1	1	0
t_2	0	1	0	1	0
t_3	1	0	0	1	0
t_4	1	1	x	x	1
t_5	y	$\bar{y}$	1	0	1

Table 1B: Tests for the full adder in FIg. 1B

	In			Out	
test	a	b	c_{in}	s	c_{out}
t_1	0	1	1	0	1
t_2	1	0	0	1	0
t_3	1	1	1	1	1

Table 2: Necessary Assignments

fault	a	b	c_{in}	g_3	g_4	g_5
$g_1\_g_3/1$	0	0	1	0	0	0
$a\_g_4/1$	0	1	0	0	0	0
$b\_g_4/1$	1	0	0	0	0	0
$g_3/1$	1	1	x	0	1	1
$g_4/1$	y	$\bar{y}$	1	1	0	1

For the adder in Fig. 3B, an independent fault set with 12 faults was found, and 12 tests as shown in Table 3B can detect all single faults. Thus, the 12 tests form a minimum test set for the adder in Fig 3B. This is larger than the size of the minimum tests for the adder in Fig. 3A. One of the considerable reasons is that we used trees of 2-input EXOR gates, instead of multiple-input EXOR gates. For example, a 2-input EXOR gate requires three tests for single stuck-at faults, but a 3-input EXOR gate can be tested by only two tests. Therefore, the use of multiple-input EXOR gates may reduce the number of tests.

3. Minimum tests for cascaded adders

3.1 Tests for ripple carry adders

The ripple carry adder shown in Fig. 2 consists of cells of the full adders, where the primary output, c_{out}, of the full adder is connected to the primary input, c_{in}, of the next full adder. In this section, we show how to construct minimum tests for the ripple carry adder as an array of the full adder.

In order to use the tests of the full adder for the ripple carry adder, it is required to satisfy two conditions below:
(1) to apply the tests of the full adder for each cell.
(2) to observe the output of each cell where fault effect may be propagated.
The following theorems show that these condition can be satisfied.

Theorem 4. Any tests for the full adder can be applied to each cell of the ripple carry adder.
(Proof) For the first cell, the theorem holds. For the ith cell ($i \geq 2$), the signal line c_{i-1} that connects the (i-1) th cell with the i th cell is controllable using the primary inputs, a_{i-1} and b_{i-1}. Hence, the theorem holds. (Q.E.D.)

Theorem 5. If tests for the full adder are applied to each cell of the ripple carry adder, faults in the ripple carry adder are detected.
(Proof) Suppose that tests for the full adder are applied to the ith cell of the ripple carry adder. If effects of a fault in the ith cell are propagated to s_i, the fault can be detected. Even if fault effects are propagated to c_i, it can be detected at the primary output s_{i+1} independently of the logic values of a_{i+1} and b_{i+1}. Thus, any fault in a cell is detected by the tests for the cell, and the theorem holds. (Q.E.D.)

The logic value at c_{out} of the ith cell corresponds to the logic value at c_{in} of the (i+1)th cell. For the array of cells in Fig. 1A, whenever the test t_2, t_3, or t_5 in Table 1A are applied to the ith cell, the same test can be applied to the (i+1)th cell because logic values required at c_{in} of the (i+1)th cell and c_{out} of the ith cell are identical. However, when test t_1 is applied to the ith cell, to the (i+1)th cell, another test which requires logic value 0 at c_{in} should be applied. We achieve it by assigning logic value 0 to x of test t_4. Conversely, when test t_4 is applied to the ith cell, test t_1 is applied to the (i+1)th cell.

As a result, by assigning logic value 0 to x of test t_4 and logic value 0 to y of test t_5, we can obtain five tests as shown in Table 4 to detect all single faults in the ripple carry adder

1081-7735/97 $10.00 © 1997 IEEE

based on the full adders in Fig. 1A. Similarly, we can obtain three tests for the ripple carry adder based on the full adders in Fig. 1B because logic values at c_{in} and c_{out} are identical for every test. These tests obtained for the array are minimum because their sizes are the same as those of the minimum tests for each cell.

3.2 Tests for multiple stuck-at faults in ripple carry adders

Under the assumption of the multiple stuck-at fault model, stuck-at faults are taken into account only at fanout branches and the primary inputs which are called *checkpoints*. This is because every multiple stuck-at fault can be represented by a combination of stuck-at faults on checkpoints [10]. Both of the full adders in Fig. 1A and 1B have 11 checkpoints, i.e., a, b, c_{in}, fanin lines of gate g_1, g_2, g_3, and g_4. If we treat multiple stuck-at faults in the ripple carry adders, we must assume faults on every checkpoints in each cell.

Table 3A: Tests for the carry look-ahead adder in FIg. 3A

	In								Out	
test	a_1	a_2	a_3	a_4	b_1	b_2	b_3	b_4	c_0	c_4
t_1	0	1	1	1	0	0	0	0	1	0
t_2	1	1	1	0	1	1	1	0	1	0
t_3	1	0	0	0	1	0	1	1	1	0
t_4	1	1	0	1	0	1	1	0	0	1
t_5	1	1	1	0	1	0	0	1	0	1
t_6	0	0	0	0	1	1	1	1	1	1
t_7	1	1	0	1	1	1	0	0	1	0
t_8	0	1	1	0	1	0	1	1	0	1
t_9	1	0	0	1	0	0	0	1	1	1
t_{10}	0	0	0	0	1	1	1	1	0	0

Table 3B: Tests for the carry look-ahead adder in FIg. 3B

	In								Out	
test	a_1	a_2	a_3	a_4	b_1	b_2	b_3	b_4	c_0	c_4
t_1	1	1	1	1	0	0	0	0	1	1
t_2	1	0	0	0	1	1	1	1	1	1
t_3	0	1	0	0	1	0	1	1	0	0
t_4	0	1	0	0	1	0	1	0	1	0
t_5	1	1	0	0	1	1	1	1	0	1
t_6	1	0	0	1	1	1	0	0	0	0
t_7	0	1	1	0	1	1	0	0	0	0
t_8	1	1	0	0	1	1	0	1	0	0
t_9	1	1	1	0	0	0	1	0	0	0
t_{10}	1	0	1	0	0	0	0	1	1	0
t_{11}	1	0	1	0	0	1	1	1	1	1
t_{12}	1	0	1	1	1	1	0	1	0	1

Table 4: Minimum tests for the ripple carry adder

test	a_1	b_1	c_0	a_2	b_2	a_3	b_3	a_4	b_4	• • •
T_1	0	0	1	1	1	0	0	1	1	• • •
T_2	0	1	0	0	1	0	1	0	1	• • •
T_3	1	0	0	1	0	1	0	1	0	• • •
T_4	1	1	0	0	0	1	1	0	0	• • •
T_5	0	1	1	0	1	0	1	0	1	• • •

The tests in Table 1A can detect all multiple stuck-at faults in the full adder of Fig. 1A. Any multiple fault which contains stuck-at faults on lines between the primary inputs, a, b, c_{in} and the primary output s are detected at s. And any multiple fault which contains stuck-at faults on fanin lines of gate g_3 and g_4 are detected at c_{out}.

For the ripple carry adder, however, the five tests in Table 4 cannot detect all multiple faults. For example, a multiple fault $(a_2/1, b_2/0, g_1\_g_3/1, a\_g_4/0, b\_g_4/1)$ remains undetected. Although we can produce other 5 tests to detect all single faults by assigning logic value 1 to y in Table 1A, the multiple fault cannot be detected. As any other five tests cannot detect all single faults, at least six tests are required for detecting all multiple faults.

By adding a test such that the logic value of every primary input is 1 to the tests in Table 4, all multiple faults can be detected. It is proven using an idea of the fault analysis method in [11] .

Theorem 6. Six tests in Table 5 detect all multiple stuck-at faults of the ripple carry adder consisting of the full adders in Fig. 1A.

(Proof is in [12])

Since it is impossible to detect all multiple faults by five tests, the tests in Table 5 are minimum. Also, they detect all multiple faults in the ripple carry adder using the full adder in Fig 1B. But they may not be minimum tests for the circuit.

3.3 Tests for single stuck-at faults in 4*m*-bit carry look-ahead adders

By cascading the 4-bit carry look-ahead adders in Fig. 3A or Fig. 3B, we can realize a 4*m*-bit adder. We consider the size of tests for single faults in the cascaded carry look-ahead adder.

As shown in Section 2.4, the size of the minimum tests for the adder in Fig. 3A is 10. However, we could not produce a test set whose size is 10 using the tests in Table 3A. The number of tests which requires logic value 1 at c_{in} is six (i.e., t_1, t_2, t_3, t_6, t_7 and t_9), but the number of tests which produce

Table 5: Minimum tests for multiple faults of the ripple carry adder

test	a_1	b_1	c_0	a_2	b_2	a_3	b_3	a_4	b_4	• • •
T_1	0	0	1	1	1	0	0	1	1	• • •
T_2	0	1	0	0	1	0	1	0	1	• • •
T_3	1	0	0	1	0	1	0	1	0	• • •
T_4	1	1	0	0	0	1	1	0	0	• • •
T_5	0	1	1	0	1	0	1	0	1	• • •
T_6	1	1	1	1	1	1	1	1	1	• • •

Table 6: Tests for the 4m-bit cascaded look-ahead adder

test	a_1	a_2	a_3	a_4	b_1	b_2	b_3	b_4	c_0	a_5	a_6	a_7	a_8	b_5	b_6	b_7	b_8	• • •
T_1	0	1	1	1	0	0	0	0	1	1	1	0	1	0	1	1	0	• • •
T_2	1	1	1	0	1	1	1	0	1	1	1	1	0	1	0	0	1	• • •
T_3	1	0	0	0	1	0	1	1	1	0	1	1	0	1	0	1	1	• • •
T_4	1	1	0	1	0	1	1	0	0	0	1	1	1	0	0	0	0	• • •
T_5	1	1	1	0	1	0	0	1	0	1	1	1	0	1	1	1	0	• • •
T_6	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	• • •
T_7	1	1	0	1	1	1	0	0	1	0	1	1	0	1	0	1	1	• • •
T_8	0	1	1	0	1	0	1	1	0	1	0	0	0	1	0	1	1	• • •
T_9	1	0	0	1	0	0	0	1	1	1	0	0	1	0	0	0	1	• • •
T_{10}	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	1	• • •
T_{11}	0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	0	0	• • •

logic value 1 at c_{out} is five (i.e., t_4, t_5, t_6, t_8 and t_9). Accordingly, if we use the tests in Table 3A without repeating same one, one of the six tests cannot be applied. In order to avoid missing any test, we add an additional test such that logic value 0 is assigned to input c_{in} and logic value 1 at output c_{out} is produced. As a result, we obtain 11 tests for the cascaded look-ahead adder. An example of such tests is shown in Table 6. Note that we have not proven the minimality of the tests in Table 6.

For the 4m-bit carry look-ahead adder realized by cascading the adders in Fig. 3B, we can obtain 12 tests to detect single stuck-at faults. The 12 tests are minimum for the circuit and it can be proven by the similar way to Theorem 4 and 5.

4. Conclusion and comments

In this paper, we presented minimum tests for ripple carry adders and cascaded carry look-ahead adders. Especially, we presented a ripple carry adder that can be tested by only three tests. These tests are considerably smaller than previously published ones. This is due to the compact full adder using more EXOR gates. Without EXOR gates, adders and their test will be more complex. Thus, for the stuck-at fault model, we can say as follows: "EXOR based ripple carry adders are easily testable."

REFERENCES

[1] W. H. Kautz, "Testing for Faults in Cellular Logic Arrays," *8th Annu. Sympo. Switching and Automata Theory*, pp. 161-174, 1967.

[2] W. -T. Cheng, and J. H. Patel, "A Minimum Test set for Multiple Fault Detection in Ripple carry Adders," *IEEE Trans. Comput.*, vol. C-36, no. 7, 891-5, July 1987.

[3] S. Muroga, *Logic Design and Switching Theory*, Wiley-Interscience Publication, pp. 518-520, 1979.

[4] B. Becker, "Efficient Testing of Optimal Time Adders," *IEEE Trans. Comput.*, vol. 37, no. 9, 1113-21, Sept. 1988.

[5] F. J. Hill, and G. R. Peterson, *Introduction to Switching Theory and Logical Design (Third Edition)*, John Wiley & Sons, p. 180, 1981.

[6] Z. Kohavi, *Switching and Automata Theory*, McGraw-Hill, p. 116, 1978.

[7] S. B. Akers, and B. Krishnamurthy, "On the Application of Test Counting to VLSI Testing," *Technical Report No. CR-85-12*, Computer Research Laboratory, Tektronix Laboratories, April 1985.

[8] R. F. Tinder, *Digital Engineering Design*, Prentice-Hall, p. 274, 1991.

[9] S. Kajihara, I. Pomeranz, K. Kinoshita and S. M. Reddy, "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits" *IEEE Trans. on CAD.*, Vol. 14, No. 12, pp. 1496-1504, Dec. 1995.

[10] D. C. Bossen, and S. J. Hong, "Cause-Effect Analysis for Multiple Fault Detection in Combinational Networks," *IEEE Trans. on Comput.*, vol. C-20, pp. 1252-1257, Nov. 1971.

[11] H. Cox, and J. Rajski, "A Method of Fault Analysis for Test Generation and Fault Diagnosis," *IEEE Trans. on CAD.*, vol. 7, pp. 813-833, July 1988.

[12] S. Kajihara, and T. Sasao, "Minimum tests for stuck-at faults of the adders," *IEICE Technical Report, Vol. FTS-96*, Oct. 1997. (to appear)

[13] M. J. Batek, and J. P. Hayes, "Test-set Preserving Logic Transformations," *29th ACM/IEEE Design Automation Conference*, pp. 454-8, 1992.

[14] B. Becker, R. Drechsler, and P. Molitor, "On the Generation of Area-time Optimal Testable Adders" IEEE *Trans. on CAD.*, vol. 14, no. 9, 1049-66, Sept. 1995.

[15] B. Becker, R. Drechsler, R. Krieger, and S. M. Reddy, "A Fast Optimal Robust Path Delay Fault Testable Adder," *European Design and Test Conference ED&TC 96*, pp. 491-498, 1996.

[16] T. -K. Liu, K. R. Hohlin, L. -E. Shiau, and S. Muroga, "Optimal One-bit Full Adders with Different Types of Gates," *IEEE Trans. Comput.*, vol. C-23, No. 1, pp. 63-70, Jan. 1974.

1081-7735/97 $10.00 © 1997 IEEE