Totally Self-Checking Carry-Select Adder Design Based on Two-Rail Code

Shao-Hui Shieh and Ming-En Lee
Department of Electronic Engineering,
National Chin-Yi University of Technology,
ssh@ncut.edu.tw, s49713032@student.ncut.edu.tw

Abstract

In this paper, the totally self-checking (TSC) carry-select adder (CSA) design is proposed. The capability of TSC can detect all single stuck-at faults on-line in normal operation mode. The proposed CSA has not only self-checking capability but also reduced transistor count. The design is based on TSMC 0.18um process technology, and a real chip is implemented. The transistor count of proposed totally self-checking CSA design is less than conventional CSA, and even reduced 34.85% compared with [4] for thirty-two bits design. The reduced ratio of transistor-count is proportional to the bit count of totally self-checking CSA to be designed. Our design has other advantages such as high extensibility, non-tree detector structure, and thus having reasonable propagation delay time and can keeping well normal operation in the high-bit design.

Keywords: Totally self-checking, two-rail code, carry select adder, checker, on-line test.

1. Introduction

Arithmetic circuits are very crucial and fundamental building blocks in many integrated circuits (IC) such as dedicated processors and digital signal processors. Therefore, to design the high performance and highly reliable adders has become the key issue and hot spot. In recent years, several types of high performance adder, e.g., carry lookahead adder, carry skip adder, conditional sum adder, carry-select adder (CSA) and so on, have been proposed. Each one adder has different advantages and drawbacks in chip area, delay time, design complexity, and power dissipation [1]. Among these adders, CSA is widely used due to short delay time, easy implementation, low power dissipation and small overhead.

Conventional n-bit CSA module, shown in figure 1, achieves high performance with the penalty of double hardware overhead. To overcome the overhead drawback, CSA is modified with only one carry ripple adder in company with the add-one (add-1) circuit [2-3]. In order to guarantee highly reliability, the capability to online detect faults in carry select adder is extremely key issue. The probability of faults, especially transient fault, occurring in modern integrated circuits has grown significantly due to the shrinkage in transistor feature size. It is well-known that transient faults can only be detected by built-in online fault detection, i.e., self-checking capability. In literature [4], a design technique has been proposed for implementing totally self-checking (TSC) carry select adders by cascading totally self-checking 2-bit adders. These carry select adders are totally self-checking for both permanent and transient single stuck-at faults. However, when the 2-bit totally self-checking carry-select adder is expanded and cascaded to a high-bit adder, a tree structure checker circuit based on 2-pair-2-rail checkers is formed. Obviously, for large bit-count the performance and the required silicon area of tree structure checkers based on 2-pair-2-rail checkers make them a non attractive solution for today applications.

This paper focuses on totally self-checking carry-select adder design based on two-rail code. The resulting carry-select adders have ability to detect errors on-line during the normal operation, and their superiority come from reduced silicon area with lower power consumption.

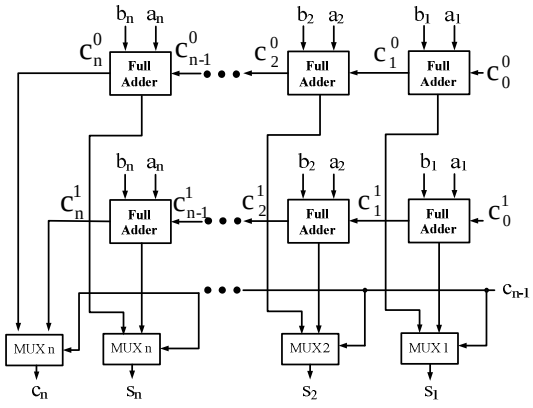

Figure 1. Traditional *n*-bit CSA module

2. Totally Self-Checking Carry Select Adder

Totally self-checking circuits are widely used in many systems with high reliability requirement. A totally self-checking circuit consists of a function circuit performing the required function specified by the chip designer, a check code generator to generate the check-codes to be check from the outputs of

function circuit, and a checker to check whether or not the check-codes be valid or invalid. In this paper, we assume that the function circuit is a two-bit CSA, the check code generator produces two-rail codes, and a two-rail checker has the capability to detect the faults occurring in the two-bit CSA. For selection of adequate CSA as function circuit, conventional CSA is fast enough, but the required silicon area is too large to be acceptable. The modified CSA implementation by using a single carry ripple adder and an add-one circuit with leading zero approach, as shown in figure 2, can effectively reduce the area overhead [2-3], and hence to play the role.

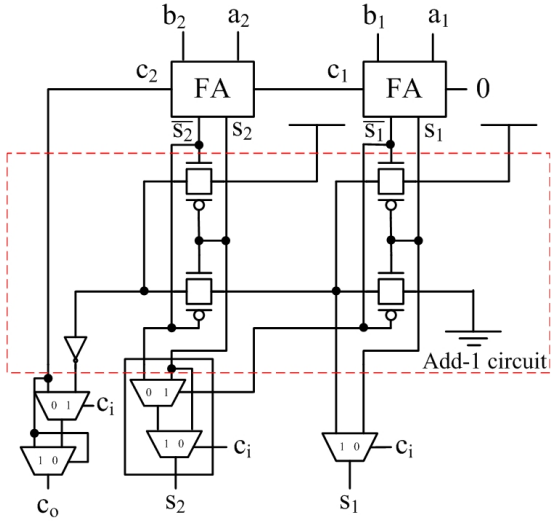

Figure 2. CSA implemented by using add-1 circuit

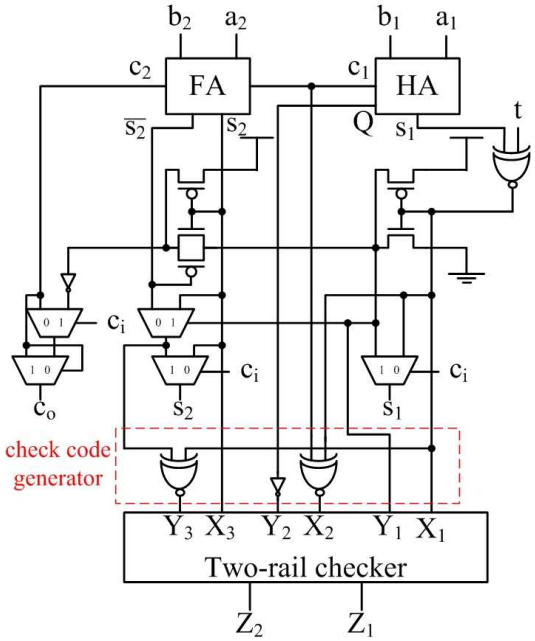

Figure 3. Totally self-checking carry select adder

The gate-level structure of the proposed totally self-checking two-bit carry select adder module is shown in figure 3, where FA denotes the mirror-type CMOS full adder implemented with twenty-eight transistors as sketched in figure 4, a simplified half adder (HA) circuit depicted in figure 5 is used for less significant bit addition due to its carry-in bit always be zero-value, some exclusive-NOR (XNOR) gates as shown in figure 6 are used with an invert to realize the check-codes generator, and a two-rail checker as depicted in figure 7 detects the two-rail check-codes and indicates the function circuit to be work or fail.

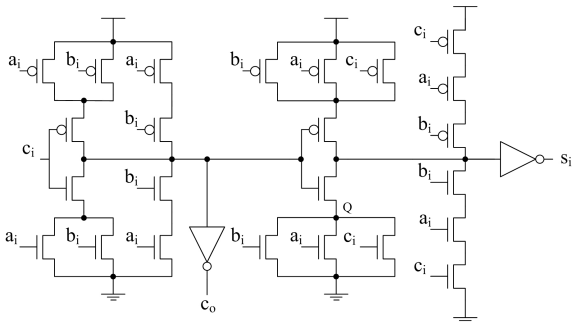

Figure 4. Mirror-type CMOS full adder

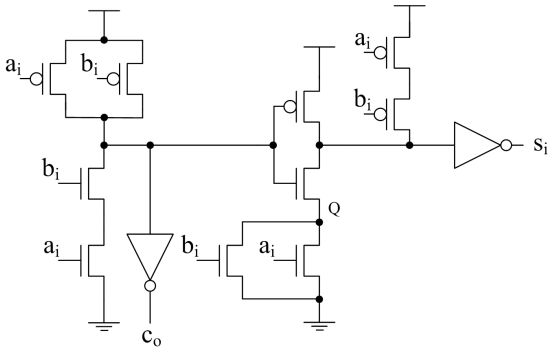

Figure 5. Modified half adder schematic

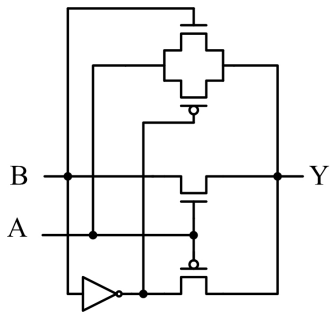

Figure 6. Exclusive-NOR gate

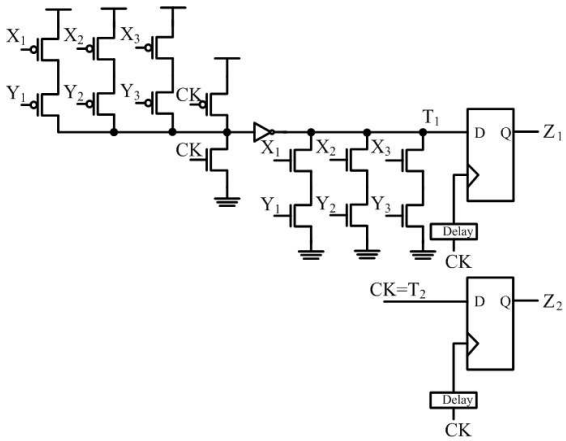

Figure 7. Two-rail checker

The sum-bits ranging from the least significant bit (LSB) to the most significant bit (MSB) of the CSA are first encoded to constitute the three-pair two-rail code denoted as (X_1, Y_1), (X_2, Y_2) and (X_3, Y_3) to perform the check-code generation. In case of the CSA is fault-free, the X_i and Y_i in each pair of the two-rail code must be complementary each other. Therefore, the two-rail checker with the three-pair two-rail code as inputs will export the high voltages for both the corresponding outputs Z_1 and Z_2. If one of the (X_i, Y_i) pair in the two-rail code is detected to have value without complement, the two-rail checker exports low-voltage for Z1 and high-voltage for Z2, the CSA is detected to be faulty. For example, we assume the three-pair two-rail code [(X_1, Y_1), (X_2, Y_2), (X_3, Y_3)] has value [(1, 0), (1, 0), (0, 1)] and imports to the input terminals of the two-rail checker shown in figure 7, both the corresponding outputs Z_1 and Z_2 can be derived to be high voltage. According to the states of Z_1 and Z_2, the CSA is detected to be fault-free in normal operation mode. If the signal of X_1 turns into 0 and the other five signals in the three-pair two-rail code are kept unchanged, the value of Z_1 will turn into 0 but Z_2 is still kept in 1. In this situation, the CSA is detected to be faulty.

3. Totally Self-Checking Function

3.1 Two-Rail Checker

The proposed two-rail checker, shown in figure 7, receives the three-pair two-rail code [(X_1, Y_1), (X_2, Y_2), (X_3, Y_3)] as the check-code input and exports both Z_1 and Z_2 signals as the error indication output. The two-rail checker can detect the presence of any single stuck-at fault in the function circuit under test, i.e., modified two-bit CSA module, on-line. There are two output states of the two-rail checker: the (Z_1, Z_2) equivalent to (0, 1) which is considered as invalid codeword, the CSA under test is faulty; another (Z_1, Z_2) state is (0, 1) which is considered as valid codeword, the CSA under test is fault-free. In the schematic of two-rail checker, nodes T_1 and T_2 may have weak signal states that will degrade the capability of fault detection. Therefore, two D-type flip flops are connected to nodes T_1 and T_2 to modify the drawback.

The waveform analysis of the two-rail checker with clock signal (CK) is shown in figure 8. There are four cases for the waveform analysis. Figure 8 (a) and (b) show the waveforms of nodes T_1 and T_2 under the condition of CAS to be faulty and fault-free, respectively. Due to the weak signals may exist in nodes T_1 and T_2, D-type flip flops are connected. Figure 8 (c) and (d) show the waveforms of the outputs Z_1 and Z_2 to be faulty and fault-free, respectively. In figure 8 (c), both Z_1 and Z_2 will keep high voltage when the CSA under test is faulty. Figure 9 (d) depicts the CSA to be faulty with Z_1 low and Z_2 high voltages.

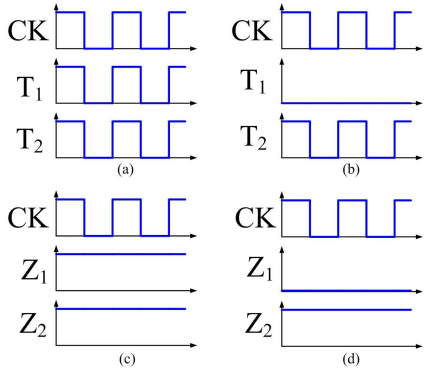

Figure 8. Waveform analysis of checker

3.2 Totally Self-Checking Circuit

In what follows, we will explain how to achieve totally self-checking function in the proposed totally self-checking CSA circuit structure. Figure 9 shows the CSA circuit under test with check code generator in fault injection situations. The *t* signal is used for fault injection into the real totally self-checking CSA chip to demonstrate the capability of totally self-checking, and the voltage of t is kept high in the normal operation mode.

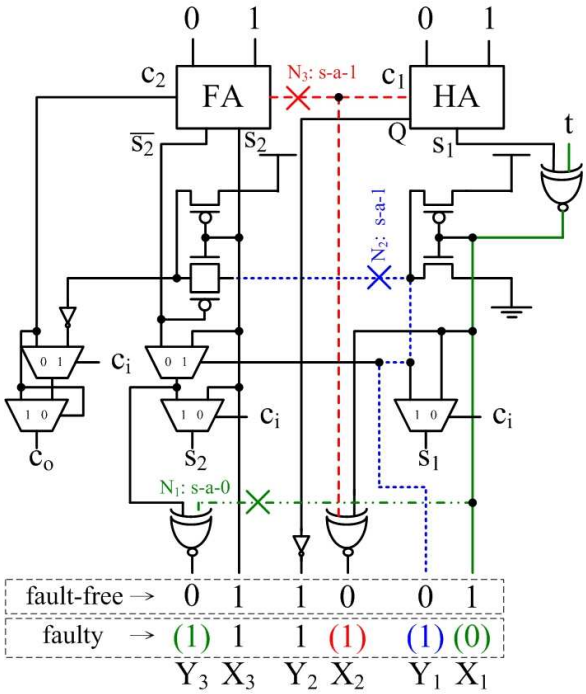

Figure 9. Circuit with fault injection

There are three stuck-at faults injected for instances, node N_1 injected a stuck-at-zero (s-a-0) fault, node N_2 injected a stuck-at-one (s-a-1) fault, and node N_3 also injected a s-a-1 fault. These faults are assumed not happened in the same time due to two or more faults situation seldom happens in the actual system. That why we consider only single fault for the proposed totally self-checking CSA, multiple

faults happened in the same time will not be under discussion.

For illustration, the input vector [a1, b1; a2, b2] of the CSA is assigned to be [1, 0; 1, 0] as shown in figure 9. In normal operation mode, such inputs generate a corresponding three-pair two-rail code [(X_1, Y_1), (X_2, Y_2), (X_3, Y_3)] with value [(*1*, *0*), (*0*, *1*), (*1*, *0*)]. The operation of the totally self-checking CSA circuit is demonstrated as the following by showing how each injected fault passes through the circuit and propagates the error message (invalid codeword) to the checker.

3.2.1 N1 Stuck-at-Zero (s-a-0)

When a stuck-at-zero fault happened on node N_1 in the CSA circuit as shown in figure 9, the signal of the node N_1 appears the wrong value *0* to substitute the normal signal *1*. The XNOR gate receives the wrong signal *0* and another true signal *0* to generate the output signal *1* to Y_3. In this case, the error message is passing through the check code generator to generate an invalid codeword due to both X_3 and Y_3 to be 1. Therefore, the output of the two-rail checker (Z_1, Z_2) is equivalent to (0, 1), and the CSA circuit under test is judged to be faulty.

3.2.2 N2 Stuck-at-One (s-a-1)

Figure 9 illustrates the case of node N_2 injected a stuck-at-one fault. The stuck-at-one fault directly affects the input signal Y_1 of the checker, and the value of Y_1 is forced to change from normal value *0* to error value *1*. Hence, both the inputs X_1 and Y_1 have value 1, and the output of the two-rail checker (Z_1, Z_2) becomes (0, 1). As a result, the CSA circuit under test is detected to be faulty.

3.2.3 N3 Stuck-at-One (s-a-1)

In the case of node N_3 injected a stuck-at-one fault as shown in figure 9, the fault generates an error signal value 1 directly propagating to the XNOR gate. The XNOR gate receives the error signal 1 and another true signal 1 to export a 1 value to X_2. One pair of the two-rail code (X_2, Y_2) is changed to be invalid codeword with value (1, 1). The output of the two-rail checker (Z_1, Z_2) is affected to be (0, 1), so that the CSA circuit under test is confirmed to be faulty.

3.2.4 Fault Injection for Real Chip

How to do fault injection for a real chip implementation based on the proposed totally self-checking CSA is an important key issue for chip measurements. The XNOR-based fault injection module with fault-injection terminal *t* is shown in figure 9. And the design idea of such fault injection circuit comes from the XNOR gate (shown in figure 10) in which output (OUT) value will be the reversed value of input (A) when the fault-injection signal (*t*) is 0, but the output value will be the same as input value when fault-injection signal is 1 and the CSA can be in normal operation without fault injection.

Based on the fault injection module, a faults can be injected on node N_1 in the CSA circuit when the fault-injection signal *t=1* as shown in figure 9, the fault injection case is similar to the node N_1 with stuck-at-zero fault previously described in section 3.2.1, but the fault injection signal affects both the X_1 and Y_3 signals in two pairs, (X_1, Y_1) and (X_3, Y_3), of the three-pair two-rail code. Therefore, the CSA in faulty state is detected by the two-rail checker.

t	A	OUT
0	0	1
0	1	0
1	0	0
1	1	1

Figure 10. XNOR-based fault injection module

4. Experimental Results and Analysis

The design of totally self-checking carry-select adder circuit is based on the TSMC 0.18um process technology of Taiwan Semiconductor Manufacturing Company. The system operating frequency is set to be 200 MHz and power supply voltage is 1.8 V. The operation waveform of totally self-checking carry-select adder is shown in figure 11, in which both the inputs denoted as CK (clock), operands: a1, b1, a2, b2, carry-in cin, and fault-injection signal *t*; and the outputs denoted as sum: s1 and s2, carry-out c2, and the outputs Z_1 and Z_2 of the checker, successively. The simulation waveform shown in figure 11 depicts that both outputs Z_1 and Z_2 of the checker to be high voltage when CSA operates in fault-free condition.

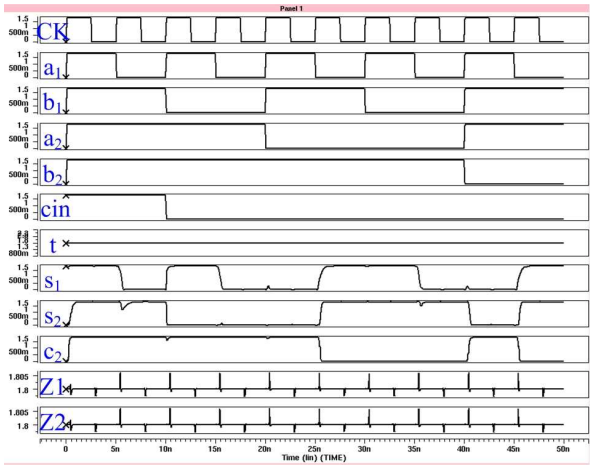

Figure 11. Fault-free system operation waveform

Four system-faulty situations are simulated to verify the results described in section 3.2, and the waveforms are shown in figure 12 to figure 15. All the simulation environments are the same as the setting in experiment for figure 11, but we only

show the waveforms for clock CK, fault-related signals and outputs due to minimizing the number of waveforms. Among the waveforms, the waveform in red rectangle is depicted as in faulty situation.

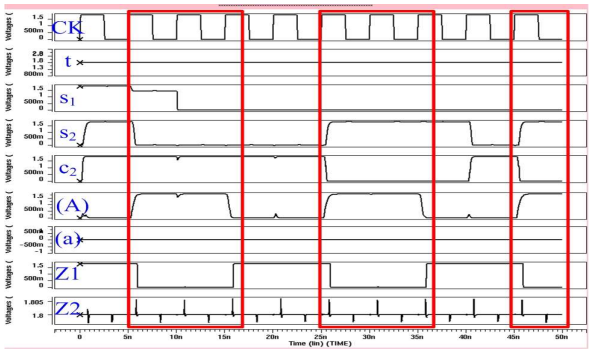

Figure 12. Waveform of N_1 with s-a-0 fault

Figure 12 shows the simulation result of the case when stuck-at-zero fault happened on node N_1 in the CSA circuit as mentioned in section 3.2.1. In the figure, waveform (A) denotes the node N_1 in fault-free state and waveform (a) denotes node N_1 in s-a-0 state.

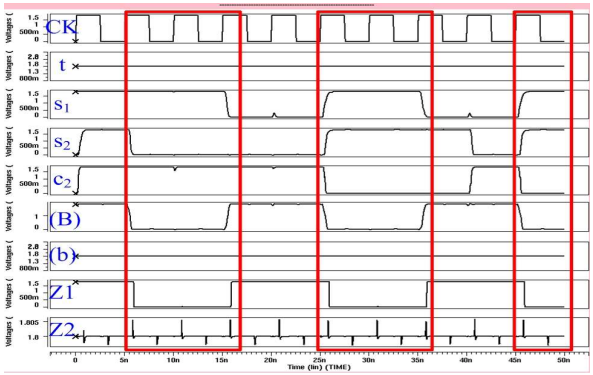

Figure 13. Waveform of N_2 with s-a-1 fault

The simulation waveform for node N_2 stuck at one, as described in section 3.2.2, is shown in figure 13. The waveform (B) stands for the waveform of node N_2 in fault-free state and the waveform (b) stands for the waveform of node N_2 with s-a-0 fault.

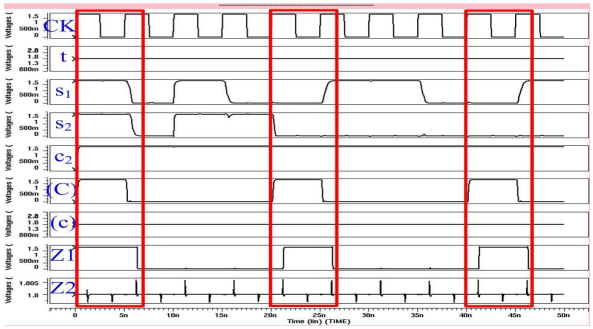

Figure 14. Waveform of N_3 with s-a-1 fault

Figure 14 is the simulation waveform for node N_3 stuck at one as discussed in section 3.2.3. The waveform (C) represents the waveform of node N_3 in fault-free state and the waveform (c) represents the waveform of s-a-0 happened in node N_3.

Figure 15 shows the simulation result of the fault injection by using the fault-injection signal (t) assigned to be 0. The fault-injection is designed for real chip verification for the capability of totally self-checking as mentioned in section 3.2.4.

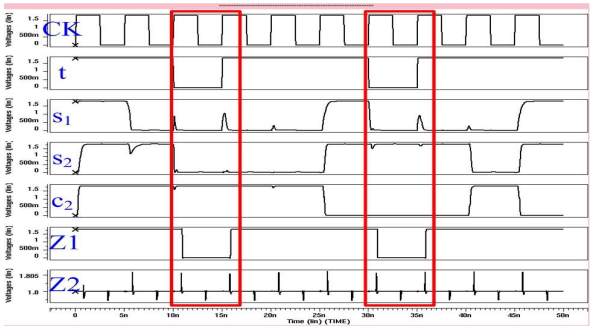

Figure 15. Waveform of fault infection module

5. Real Chip Implementation

The proposed totally self-checking CSA circuit is designed and simulated based on TSMC 0.18 um mixed signal CMOS process technology, all simulation results, from figure 11 to figure 15, show our design is consistent with requirement.

The chip layout of totally self-checking CSA module is shown in figure 16. Table 1 lists the relationship of transistor count versus bit count of the adder for proposed totally self-checking CSA circuit compared with reference designs.

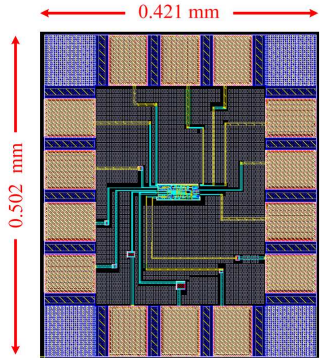

Figure16. Totally self-checking CSA chip layout

The proposed totally self-checking thirty-two bits CSA circuit has advantage of 34.85% reduced transistor-count against that of adder in [4]. The reduced ratio of transistor-count is proportional to the bit count of totally self-checking CSA to be designed.

In addition, the extensibility of our totally self-checking CSA circuit is batter than that of [4]. Since the detector designed in [4] is with tree structure, therefore as the bit count increasing it will generate unacceptable propagation delay and lead to the circuit misjudge. The circuit structure of the proposed detector is non-tree, hence it has reasonable

propagation delay and can keep well normal operation in the same situation.

Table 1. Transistor count vs. bit count comparison

Bit Count	Transistor Count			
	Conventional CSA [1]	[4]	Proposed Design	Reduced Rate
2-bit	148	166	137	17.46%
4-bit	296	340	248	27.05%
8-bit	592	688	470	31.68%
16-bit	1184	1384	914	33.95%
32-bit	2368	2766	1802	34.85%

6. Conclusion

The totally self-checking carry-select adder design is proposed. The design is based on TSMC 0.18um mixed signal CMOS process technology. The experimental results show that our design can work normally in spite of on checking. The high bit totally self-checking carry-select adder can easily be implemented based on proposed totally self-checking two-bit carry-select adder module by appropriately cascading connection. The transistor count of proposed totally self-checking carry-select adder design is less than conventional CSA. For thirty-two bits totally self-checking CSA, 34.85% reduced transistor-count is achieved compared with that of [4]. The reduced ratio of transistor-count is proportional to the bit count of totally self-checking CSA to be designed. Our design has other advantages of high extensibility, non-tree detector structure, and thus having reasonable propagation delay time and can keeping well normal operation in the high-bit design. The experimental results show that our design is valid and effective than previous works.

Acknowledgements

We would like to express our sincere thanks to the Chip Implement Center (CIC) and Taiwan Semiconductor Manufacturing Company (TSMC) for providing the chip manufacturing and other related help (Tapeout number: D35-98B (F)/65 and 18-99A (F)/334) to make our research more intact.

References

[1] Z. Chen and I. Koren, “Techniques for Yield Enhancement of VLSI Adders,” in Proceeding International Conference Application Specific Array Processor, pp. 222–229, Strasbourg, France, 24 – 26 Jul. 1995.

[2] T.-Y. Chang and M.-J. Hsiao, “Carry-Select Adder Using Single Ripple-Carry Adder,” Electronics Letters, Vol.34, Issue 22, pp. 2101-2103, 1998.

[3] Y.-J. Kim and L.-S. Kim, “A Low Power Carry Select Adder with Reduced Area,” IEEE International Symposium on Circuits and Systems (ISCAS 2001), Vol.4, pp. 218-221, 2001.

[4] D. P. Vasudevan, P. K. Lala and J. P. Parkerson, “Self-Checking Carry-Select Adder Design Based on Two-Rail Encoding,” IEEE Transactions on Circuits and Systems I: Regular Papers, pp. 2696–2705, Dec. 2007.

[5] Omana, M. Rossi, D. Metra, C. “Low Cost and High Speed Embedded Two-Rail Code Checker,” IEEE Transactions on Computers, pp. 153–164, Feb. 2005.