Berger Code Totally Self-Checking Checker Design for Embedded Adder Cores

Shao-Hui Shieh and Wei-Siou Tong
Department of Electronic Engineering,
National Chin-Yi University of Technology,
ssh@ncut.edu.tw; s49713026@student.ncut.edu.tw

Abstract

Totally self-checking (TSC) adder design based on the Berger code is proposed for embedded adder cores. A self-checking circuit can execute on-line testing in normal system operation. TSC can immediately detect the error of an electronic system or a computer to avoid the data damaged or the malfunction of a function circuit. Hence, TSC can enhance the reliability of an electronic system, i.e., using TSC may reduce the harm of the electronic system to the lowest as we wish. A real silicon implementation is given based on the TSMC 0.35μm mixed signal process technology. The experiment results show that transistor count of the proposed checker bits generator (CBG_New) has 59.3% reduced than that of previous MOS design and the power consumption is also reduced to 4.83mW under the condition of the operating frequency at 100 MHz. The lower power is due to eliminating both the charge and discharge times of the weight decoder circuit in our design.

Keywords: Berger code, totally self-checking checker, information bits, checker bits, full adder.

1. Introduction

Highly reliable electronic or computer system is widely used in many important applications such as the navigation system on a space ship, the computer system on an aircraft, the accounting system in a bank, the signal processor in a biochip, and so on. To guarantee the detection of faults, conventional high-reliability electronic systems use off-line testing in which the systems have to stop their normal operation mode in order to enter the test mode. Due to the shrinkage in transistor feature size, the probability of transient faults occurring in modern integrated circuits has grown significantly. However, transient faults can only be detected by using built-in on-line testing approaches. Self-checking capability can fit the requirement, since self-checking circuits are able to detect the faults, including permanent, transient as well as intermittent faults, in normal operation mode. A self-checking circuit can immediately detect the existing faults to avoid the circuit to be damaged or malfunction. Hence, Self-checking circuit can enhance the reliability of the circuit. As a result, the harm or hazard due to the present of all single faults in the circuit can be reduced to the lowest even no any effect as we wish.

Self-checking circuits [1] consist of a function circuit (circuit under test) whose fault-free outputs belongs to a certain error-detecting code (EDC), and a checker that monitors whether or not the outputs of the function circuit to be the valid codes. Error-detecting codes can be classified as systematic and non-systematic. The systematic codes, such as Berger code [2], Bose-Lin code [3] and two-rail code (TRC) [4], are composed of the information part and the check part. The non-systematic codes, such as m-out-of-n (m/n) code [1] and Borden code [5], can not be divided into information part and check part, individually.

Berger codes are the least redundant separable codes among the all unidirectional error detecting (AUED) codes. For N-bit information part of a Berger code, the corresponding check part has length $r = \lceil \log_2(N+1) \rceil$. There are two different encoding schemes, denoted as B_0 and B_1, for Berger code. The B_0 encoding scheme uses the binary representation of the number of 0's in the information bits as the check symbol, whereas the B_1 encoding scheme uses the ones complement of the number of 1's in the information bits as the check one. In this paper, Berger code with the B_1 encoding scheme is adopted for our designs.

2. Totally Self-Checking System

In 1973, totally self-checking (TSC) was formally defined by D. A. Anderson [1]. For a totally self-checking circuit, some inputs vectors are defined as valid inputs (or input code words), and some outputs as valid outputs (or output noncode words), so that the circuit can detect faults by observing whether the outputs are valid or not. Before succeeding presentation, some key terminologies are discussed and defined in the following:

Definition 1: A circuit is *self-testing* (*ST*) for a set of faults *F* if, for every fault in *F*, the circuit produces a noncode word output for at least one code word input.

Definition 2: A circuit is *fault-secure* (*FS*) for a set of faults *F* if, for every fault in *F*, the circuit never produces an incorrect code word output for all code

word inputs.

Definition 3: A circuit is *code-disjoint* (*CD*) if, during fault-free operation, code word inputs map into code word outputs and noncode word inputs map into noncode word outputs.

Definition 4: A circuit is *totally self-checking* (*TSC*) if it is *ST* and *FS*.

The Berger code TSC system, shown in Figure 1, consists of a functional circuit, checker and check bits generator. Outputs of the function circuit are designated as information bits, while corresponding check bits encoded in accordance with information bits are produced by the check bits generator (CBG). Checker can detect faults occurring in the system during the normal operation. In what follows, we focus on design of the 2-bit ripple-carry adder (RCA) as the function circuit, and the checker based on Berger code which has the capability to detect faults in the 2-bit RCA.

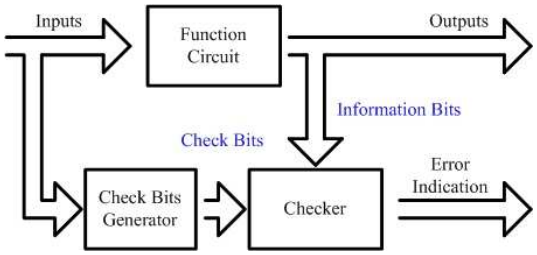

Figure 1. Structure of Berger code TSC system.

3. Design of Berger Code Checker

The proposed Berger code checker is designed based on the *k*-weight threshold circuit as described in following.

3.1 *k*-Weight Threshold Circuit

The *k*-weight threshold circuit [7] is sketched in Figure 2. If the number of 1's in the information bits (x_1, x_2, ⋯, x_n) is equal to or more than *k*, the checker output (V_{out}) is high. Otherwise, the checker output is low. Aspect ratios of these transistors can be expressed as:

$$\frac{W}{L}=\frac{W_{pm}/W_{nm}}{L_{pm}/L_{nm}}=\frac{W_{pm}\cdot L_{nm}}{W_{nm}\cdot L_{pm}} \qquad (1)$$

And also satisfies the following relationship:

$$(k-1)\frac{KP_n}{KP_p}\cdot\frac{2(V_{dd}-V_{tn})V_{IHMIN}-V_{IHMIN}^2}{(V_{dd}+V_{tp})^2}\le\frac{W}{L}$$
$$\le k\frac{KP_n}{KP_p}\cdot\frac{2(V_{dd}-V_{tn})V_{ILMAX}-V_{ILMAX}^2}{(V_{dd}+V_{tp})^2} \qquad (2)$$

The following notation has been used in equation (2):

- V_{IHMIN} (V_{ILMAX}) is the *MINimum High* (*MAXimum Low*) *Input* voltage which is recognized as logic 1 (0) from a driven gate.
- V_{tn} (V_{tp}) is the threshold voltage of NMOS (PMOS) transistor.
- KP_n (KP_p) is the spice parameter for $\mu_n\cdot C_{ox}$ ($\mu_p\cdot C_{ox}$).

Hereafter we will use the following abbreviations:

$$\Psi_H=\frac{KP_n}{KP_p}\cdot\frac{2(V_{dd}-V_{tn})V_{IHMIN}-V_{IHMIN}^2}{(V_{dd}+V_{tp})^2}$$
$$\Psi_L=\frac{KP_n}{KP_p}\cdot\frac{2(V_{dd}-V_{tn})V_{ILMAX}-V_{ILMAX}^2}{(V_{dd}+V_{tp})^2} \qquad (3)$$

Equation (2) can thus be simplified as

$$\Psi_H\cdot(k-1)\le\frac{W}{L}\le\Psi_L\cdot k \qquad (4)$$

In this circuit, all NMOS transistors (nm_1, nm_2, …, nm_n) have the same sizes W_{nmos} and L_{nmos}, so that the *weight* (*k*) can be decided by the transistor-size of control PMOS (*pm*).

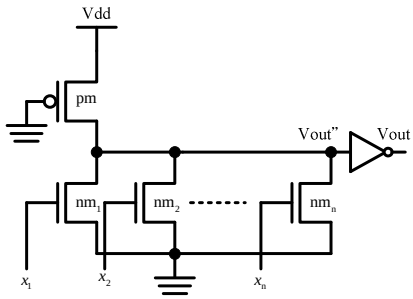

Figure 2. *k*-weight threshold circuit.

3.2 *i* / *i*+1-Weight Threshold Circuit

In the (0 to q)/(1 to q+1)-programmable-weight threshold circuit sketched in Figure 3, the size of each pmos transistors pm_i , $1\leqq i\leqq q$, can be chosen so that the aspect ratio W/L satisfies equation (4). The aspect ratio of pm_I is chosen according to the following relation:

$$\Psi_H\cdot i\le\frac{W_{pm_i}}{L_{pm_i}}+\frac{W_{pm_I}}{L_{pm_I}}\le\Psi_L\cdot(i+1) \qquad (5)$$

The operation of the circuit of Figure 3 depends on the value of the input *I*. If *I*=0, for total *i*-bit of d_i, $i\in\{0,1,...,q\}$, with value equals to 0 and the rest bits to be 1, the circuit is definitely the *i*+1-weight threshold circuit; and hence the circuit behaves as an (1 to *q*+1)-programmable-weight threshold circuit. For *I*=1, in the same condition as described above,

the circuit is obviously the *i*-weight threshold circuit; as a result the circuit behaves as an (o to *q*)-programmable-weight threshold circuit.

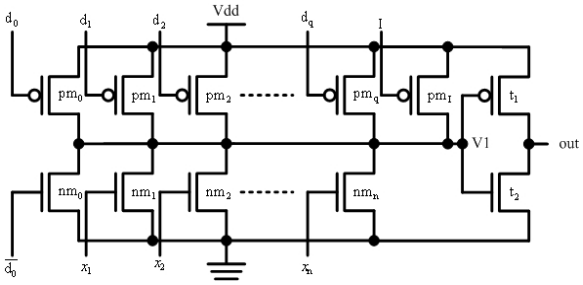

Figure 3. (0 to *q*)/(1 to *q*+1)-programmable-weight threshold circuit.

3.3 Weight Decoder Circuit

The weight decoder circuit, depicted in Figure 4, has Cp to be the system clock signal, check bits C_0 and C_1 to be the input signals, and $\{d_0, d_1, d_2, d_3\}$ to be weight decoder output signals.

When the clock in low-phase, i.e., Cp=0, the PMOS transistors charge the weight decoder output nodes d_0, d_1, d_2, and d_3, to be high levels. In the instance of Cp=1, weight decoder output signals$\{d_0, d_1, d_2, d_3\}$ will be evaluated depending on the value of the check bits C_0 and C_1. The truth table for the weight decoder circuit is shown in Table 1.

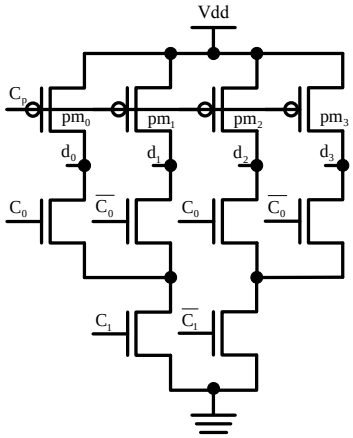

Figure 4. Weight decoder circuit.

Table 1. Truth table for the weight decoder circuit

Check Bits C_1 C_0	Decoder Outputs d_3 d_2 d_1 d_0	Weight of $X_1...X_3$
1_1 1_0	1_3 1_2 1_1 0_0	0
1_1 0_0	1_3 1_2 0_1 1_0	1
0_1 1_0	1_3 0_2 1_1 1_0	2
0_1 0_0	0_3 1_2 1_1 1_0	3

3.4 Berger Code Checker

The proposed Berger code checker [7] is show in Figure 5. C_0 and C_1 belong to the check part of the Berger code word, and x_1, x_2, and x_3 account for the information part. When I=1, module M_0 operates as the (1 to q+1)-programmable-weight circuit and module M_1 operates as the (0 to q)-programmable-weight circuit. Thus, for total i-bit d_i having the zero-value, i.e., d_i=0 for $i \in \{0,1,...,q\}$, the module M_0 behaves the i+1-weight threshold circuit and module M_1 performs the i-weight threshold circuit. However, in the case of I=0, the weights of both module M_0 and module M_1 are exchanged each other.

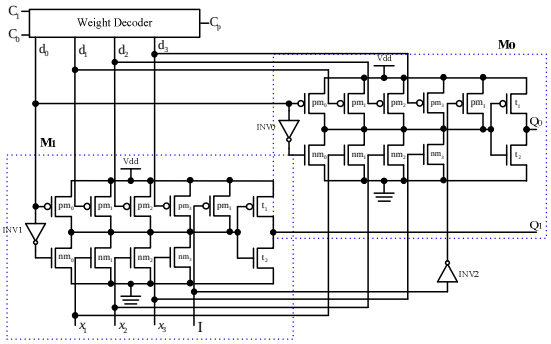

Figure 5. Berger code checker for n=3 and *r*=2.

In what follows, W denotes the Hamming weight (number of ones) of the input vector x_1, …, x_3 and m denotes the weight of d_m which is the output of the weight decoder circuit. Then, the output (Q_0, Q_1) of the checker corresponding to each possible input (x_1, x_2, x_3, C_0, C_1), denoted as (X, C) for short, with different weight is show in Table 2.

Table 2. The outputs of the checker for each possible input

I	Weight	Q_0 Q_1
0	$W<m$	0 0
	$W=m$	1 0
	$W>m$	1 1
1	$W<m$	0 0
	$W=m$	0 1
	$W>m$	1 1

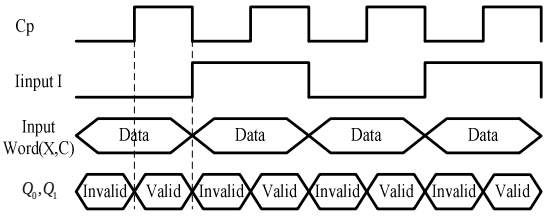

Figure 6. The checker input signal corresponding to the output signal waveforms.

The input I of the checker is driven by the operation frequency equivalent to the half of the Cp

clock signal frequency. The operation waveform of the checker is shown in Figure 6. Obviously, the data (x_1, x_2, x_3, C_0, C_1) must be applied during the low phase (precharge phase) of the clock Cp and remained stable during the high phase (evaluate phase). The response output (Q_0, Q_1) of the checker is valid only during the evaluate phase of the clock Cp.

4. Design of Check Bits Generator

A two-bit ripple carry adder (2-bit RCA), shown in Figure 7, is used as the function circuit under test. The 2-Bit RCA has input signals a_1, b_1, a_0, b_0 and $c_{i,0}$, and corresponding output signals $c_{o,1}$, s_0 and s_1 which are used as the information bits. C_0 and C_1 are check bits in accordance with B_1 encoding scheme for Berger code. According to Table 3, the relationships of inputs a_1, b_1, a_0, b_0 and $c_{i,0}$ with corresponding C_1, C_0 outputs can be derived by *Karnaugh* map simplification as following:

$$\begin{aligned} C_1 = {} & \bar{a}_1 b_1 \bar{a}_0 b_0 c_{i,0} + \bar{a}_1 b_1 a_0 \bar{b}_0 c_{i,0} + \bar{a}_1 b_1 a_0 b_0 \bar{c}_{i,0} + a_1 \bar{b}_1 a_0 \bar{b}_0 c_{i,0} \\ & + a_1 \bar{b}_1 a_0 b_0 \bar{c}_{i,0} + a_1 b_1 \bar{a}_0 \bar{b}_0 c_{i,0} + b_1 a_0 \bar{b}_0 c_{i,0} + \bar{b}_1 \bar{a}_0 b_0 c_{i,0} \\ & + \bar{a}_1 \bar{b}_1 b_0 \bar{c}_{i,0} + a_1 \bar{b}_1 \bar{b}_0 \end{aligned} \quad (6)$$

$$\begin{aligned} C_0 = {} & \bar{a}_1 \bar{b}_1 \bar{a}_0 \bar{b}_0 \bar{c}_{i,0} + \bar{a}_1 b_1 a_0 b_0 c_{i,0} + a_1 b_1 a_0 \bar{b}_0 c_{i,0} + a_1 b_1 a_0 b_0 \bar{c}_{i,0} \\ & + a_1 b_1 \bar{a}_0 b_0 c_{i,0} + a_1 \bar{b}_1 a_0 \bar{b}_0 \bar{c}_{i,0} + a_1 \bar{b}_1 a_0 \bar{b}_0 c_{i,0} + a_1 \bar{b}_1 \bar{a}_0 b_0 \bar{c}_{i,0} \\ & + b_1 a_0 \bar{b}_0 \bar{c}_{i,0} + b_1 \bar{a}_0 \bar{b}_0 c_{i,0} + \bar{b}_1 a_0 b_0 c_{i,0} + b_1 \bar{a}_0 b_0 \bar{c}_{i,0} \end{aligned} \quad (7)$$

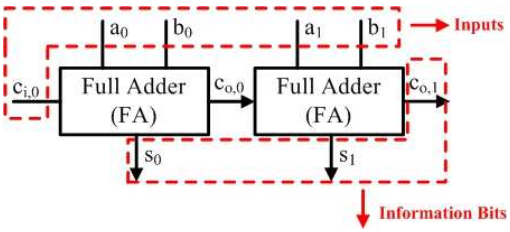

Figure 7. The 2-Bit RCA used as function circuit under test.

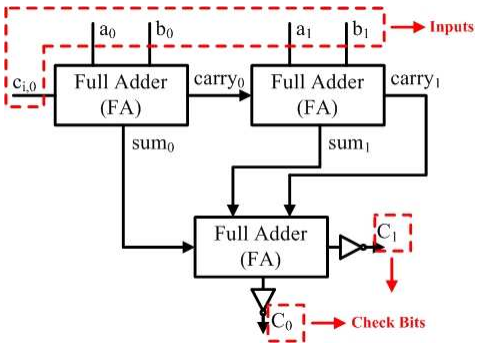

Figure 8. The CBG_New schematic.

To implement a CMOS check bits generator directly based on both equations (6) and (7), it will result in too many disadvantages, such as large transistor-count, long charge and discharge delay time, high power dissipation, and so on. By observing Table 3, the relationship between the complements of check bits (C_1, C_0) and carry and sum outputs (s_0, s_1, $c_{o,1}$) is a full adder (FA) function. As a result, a simple but effective novel design of check bits generator (CBG_New), as shown in Figure 8, is proposed based on a CMOS full adder (FA) [8] with inverters connected to outputs to provide the correct check bits. Our design can reduce both the charge and discharge times of the CBG.

Table 3. Truth table for 2-Bit RCA with check bits.

Inputs a_1 b_1 a_0 b_0 $c_{i,0}$	Information Bits $c_{o,1}$ s_1 s_0	Check Bits C_1 C_0
0_0 0_0 0_0 0_0 $0_{i,0}$	0 0 0	1 1
0_0 0_0 0_0 0_0 $1_{i,1}$	0 0 1	1 0
0_0 0_0 0_0 1_1 $0_{i,0}$	0 0 1	1 0
0_0 0_0 0_0 1_0 $1_{i,1}$	0 1 0	1 0
0_0 0_0 1_1 0_0 $0_{i,0}$	0 0 1	1 0
0_0 0_0 1_1 0_0 $1_{i,1}$	0 1 0	1 0
0_0 0_0 1_1 1_1 $0_{i,0}$	0 1 0	1 0
0_0 0_0 1_1 1_0 $1_{i,1}$	0 1 1	0 1
0_0 1_0 0_0 0_0 $0_{i,0}$	0 1 0	1 0
0_0 1_0 0_0 0_0 $1_{i,1}$	0 1 1	0 1
0_0 1_0 0_0 1_0 $0_{i,0}$	0 1 1	0 1
0_0 1_0 0_0 1_0 $1_{i,1}$	1 0 0	1 0
0_0 1_0 1_0 0_0 $0_{i,0}$	0 1 1	0 1
0_0 1_0 1_0 0_0 $1_{i,1}$	1 0 0	1 0
0_0 1_0 1_0 1_1 $0_{i,0}$	1 0 0	1 0
0_0 1_0 1_0 1_0 $1_{i,1}$	1 0 1	0 1
1_0 0_0 0_0 0_0 $0_{i,0}$	0 1 0	1 0
. . .	. . .	. . .
1_0 1_0 1_0 1_0 $1_{i,1}$	1 1 1	0 0

Table 4. Comparisons with respect to transistor count and power dissipation

CBG (@100MHz, Vdd=3.3 V)	Transistor Count **Tr. #**	Power Dissipation **Pd** (mW)
CMOS	216	0.16
CBG_New	88	0.09

Observing the simulation results shown in Table 4, the proposed CBG_New has approximately 59.26% reduced transistor count (Tr. #), and 43.75% lower power dissipation (Pd) both superior than CMOS design.

5. Berger Code TSC Adder Module

By using the proposed check bits generator

(CBG_New) as shown in Figure 8, a novel Berger code TSC checker for 2-bit embedded RCA core module is sketched in Figure 9. The 2-bit TSC module is easily used as a basic cell to expand itself to *N*-bit TSC RCA core, as shown in Figure 10, for embedded applications.

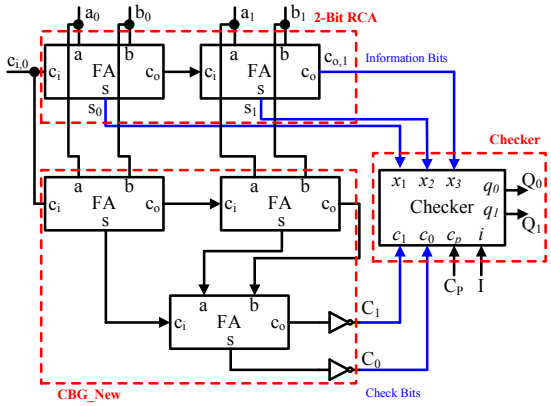

Figure 9. The Berger code TSC 2-bit RCA module for embedded system.

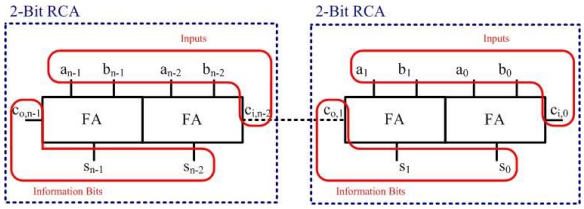

Figure 10. *N*-bit RCA schematic.

6. Experimental Results and Analysis

Based on the TSMC 0.35μm mixed signal process technology, the experiments have been performed on Berger code TSC checker for embedded 2-bit adder module at a clock frequency of 100 MHz with 3.3 V power supply voltage (Vdd). In experiments, we consider the all possible types of the inputs with different weight *W*<*m*, *W*=*m* and *W*>*m*. Post-layout simulation waveforms of the Berger code TSC adder module are shown in Figure 11. The input *I* of the checker is driven by a clock signal with operation frequency to be equal to the half of the Cp clock frequency. The response of the checker, i.e., output (Q_0, Q_1), is valid only during the evaluate phase of the clock. Since the proposed Berger code checker is designed based on ratio logic, both outputs Q_0 and Q_1 have weak signals appeared, buffers are thus added in output terminals to achieve full-swing output (Q_{00}, Q_{11}). The 0.1 pF load capacitance is added at each output to demonstrate the load effect as shown in Figure 12.

A real silicon is implemented based on the TSMC 0.35μm mixed signal CMOS process technology, the chip layout, containing Berger code TSC checker, check bits generator and I/O PADs, is shown in Figure 13. Specification for the real chip is shown in Table 5.

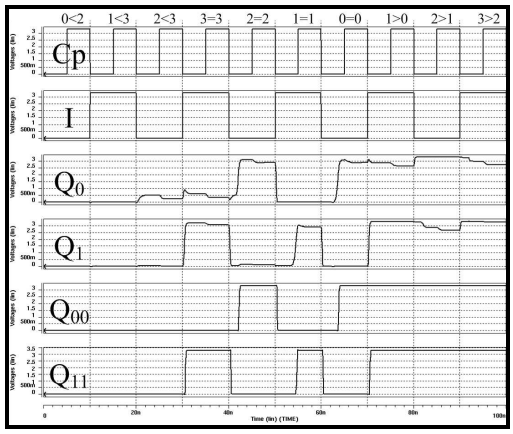

Figure 11. The post-simulation waveform.

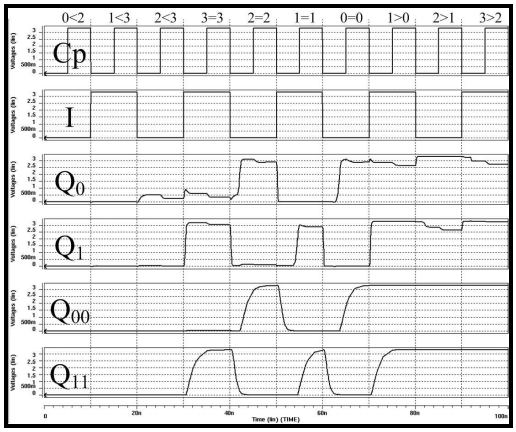

Figure 12. Load effect simulation with 0.1 pF load capacitance.

Table 5. Specification for the proposed Berger code TSC checker

Function	Berger code TSC checker for embedded adder core
Input frequency (MHz)	100
Output frequency (MHz)	50
Average power dissipation (mW)	4.83@100 MHz, Vdd=3.3 V
Tr. #	128
Chip size (mm^2)	65.10×73.05 (without I/O PAD) 1.166×1.166 (with I/O PAD)

For chip measurement, Tektronix AFG3102 arbitrary function generator is used to produce signals Cp and *I*, and the Tektronix DPO4054 oscilloscope is used to display the measured waveforms. Figure 14 shows the chip measurement results in which Tektronix DPO4054 oscilloscope is connected to measured outputs to achieve the waveform.

A rapid prototype for chip measurement and verification is realized by combining the real chip, 2-Bit RCA as function circuit under test, matrix display with interface circuit, on the PCB. When the function circuit is fault-free, matrix display shows the correct message (*O*). However, when the function circuit is with stuck-at-1 fault, matrix display shows an error message (X) as show in Figure 15.

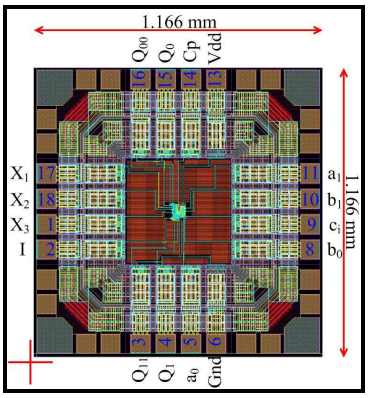

Figure 13. Chip Layout of Berger code TSC checker.

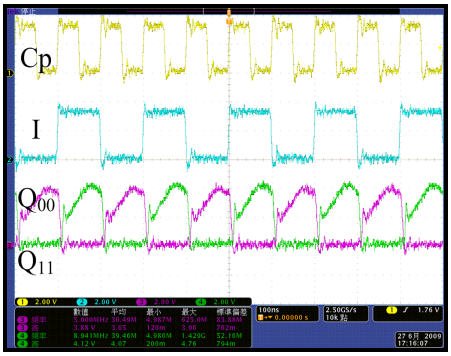

Figure 14. The fault-free with weight three

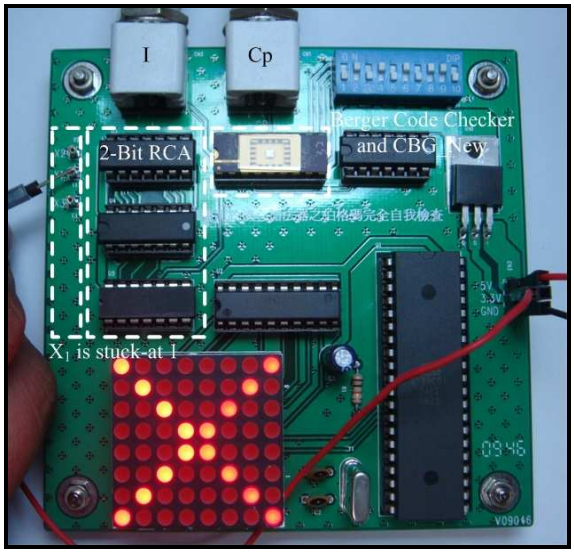

Figure 15. Rapid prototype to demo 2-Bit RCA output node X_1 s-a-1.

7. Conclusions

Berger code totally self-checking (TSC) adder design is proposed based on TSMC 0.35μm process technology, and a real chip is implemented and verified to be work. The experiment results show that the proposed checker bits generator (CBG_New) has 59.3% less transistor count and 43.75% less power dissipation than those of CMOS design. And the power consumption of our design is reduced down to 4.83 mW under the condition of the operating frequency at 100 MHz. The lower power is due to our design can reduce both the charge and discharge times of the weight decoder circuit. The result has proved our design is valid and efficient.

Acknowledgements

We would like to express our sincere thanks to the Taiwan Semiconductor Manufacturing Company (TSMC) and Chip Implement Center (CIC) in Taiwan for providing the chip manufacturing and other related help (Tape-out number: D35-98A-32).

References

[1] D. A. Anderson and G. Metze, "Design of totally self-checking checker circuits for *m*-out-of-*n* codes," *IEEE Trans. Computers*, vol. C-22, pp. 263-269, Mar. 1973.

[2] J. M. Berger, "A not on error detecting codes for asymmetric channels," *Information and Control*, vol. 4, pp. 68-73, Mar. 1961.

[3] B. Bose and D. J. Lin, "Systematic unidirectional error-detecting codes," *IEEE Trans. Computers*, vol. 34, pp. 1026-1032, Nov. 1985.

[4] W. C. Carter and P. R. Schneider, "Design of dynamically checked computers," *Proc. IFIP'68 Congress, Edinburgh, Scotland*, pp878-883, Aug. 1968.

[5] J. M. Borden, "Optimal asymmetric error detecting codes," Information and Control, vol. 53, pp. 66-73, Apr. 1982.

[6] X. Kavousianos and D. Nilolos, "Self-Exercising, Self-Testing k-order Comparators," *Proc. 15th IEEE VLSI Test Symposium*, pp. 216-221, Monterey California, April 27-May 1, 1997.

[7] X. Kavousians, D. Nikolos, G. Foukarakis, and T. Gnardellis, "New Efficient Totally Self-checking Berger Code Checkers," *Integration, The VLSI Journal*, vol. 28, no 1, pp. 101-118, Sep. 1, 1999.

[8] N. H. E. Weste and K. Eshraghian, "Principles of CMOS VLSI design," Addison Wesley, 1993.

[9] X. Kavousianos and D. Nikolos, "Novel Single and Double Output TSC Berger Code Checkers," *Proc. 16th IEEE VLSI Test Symposium*, pp. 348-353, April 1998.