Self-Testing Embedded Checkers for Bose-Lin, Bose, and a Class of Borden Codes

Steffen Tarnick

SATCON GmbH
Satellitenkommunikationsgesellschaft
Warthestr. 21
D-14513 Teltow, Germany

Abstract—*A new approach for designing t-UED and BUED code checkers is presented. In particular we consider Borden codes for $t = 2^k - 1$, Bose and Bose-Lin codes. The design technique for all three checker types follows the same principle, which is mainly based on averaging weights and check symbol values of the code words. The checkers are very well suited for use as embedded checkers since they are self-testing with respect to single stuck-at faults under very weak assumptions. All three checker types can be tested by 2 or 3 code words.*

1 Introduction

A Borden code $B(n, t)$ [1] is the set of all length n codewords whose weight is congruent to $\lfloor n/2 \rfloor \bmod (t + 1)$. For $t = 1$, $B(n, t)$ becomes an even/odd parity code if $\lfloor n/2 \rfloor$ is even/odd. A cyclic shift of a Borden code word is a Borden code word as well.

Bose's BUED code $B_1(n, k)$ [2] has n information bits and k check bits. The code is constructed as follows. Let $x^{(i)} = (x^{(i)}_{n-1}, x^{(i)}_{n-2}, \ldots, x^{(i)}_0)$ be the n-bit information bit vector and $x^{(c)} = (x^{(c)}_{k-1}, x^{(c)}_{k-2}, \ldots, x^{(c)}_0)$ the k-bit check symbol. The check symbol $x^{(c)}$ is given by $x^{(c)} = k_0 \bmod 2^k$ where k_0 denotes the number of 0's in the information bit vector. The information and check part are merged but the order of the code word bits do not influence the checker design. For $k = 1$, Bose's code becomes an even/odd parity code if n is even/odd.

Bose-Lin t-UED codes $B_2(n, k)$ [3] are constructed relatively similar to Bose codes. For $k \leq 3$, the check symbol is $x^{(c)} = k_0 \bmod 2^k$ or $x^{(c)} = 2^k - 1 - (k_1 \bmod 2^k)$, where k_0 and k_1 denote the number of 0's and 1's in the information bit vector. For $k \geq 4$ the check symbol has the form $x^{(c)} = (k_0 \bmod 2^{k-1}) + 2^{k-2}$.

In this paper we propose a new approach for designing checkers for Borden codes $B(n, t)$ with $t = 2^k - 1$, Bose codes $B_1(n, k)$, and Bose-Lin codes $B_2(n, k)$. The checker design for all three types of codes is based on averaging code words. The result of the averaging operation is a word of the same code type, but for $k - 1$, which is checked by a lower-level checker for the same code type. For $k = 1$ it is a simple parity checker.

2 Checker Architectures

An essential part of the proposed checkers are averaging circuits. A weight averaging circuit (WAC), shown in Figure 1, consists of two D-flip-flops and n full adders (FA) where the roles of the sum bit and the carry-out bit are exchanged, i.e. the sum output of the i-th full adder is connected to the carry input of the $i + 1$-st full adder, and the carry output of the i-th full adder is the i-th bit of the result of the operation. The D-flip-flops are initialized with two complementary values.

The inputs of a WAC are two n-bit words, x and y. The outputs are the n-bit word z and the two signals d_0 and d_1. With $w(x)$ denoting the weight of a word x, the signals of a WAC satisfy the equation

$$\frac{w(x) + w(y)}{2} = w(z) + \frac{c_n - c_0}{2}.$$

A WAC maps two words $x, y \in B(n, 2^k - 1)$ to a word $z \in B(n, 2^{k-1} - 1)$ and a two-rail signal $d_0 d_1 \in \{01, 10\}$.

For systematic codes, an averaging circuit (AC) based on a similar principle is used. It consitsts of two parts, as shown in Figure 2. The first part performs the weight averaging operation on the information parts $x^{(i)}$ and $y^{(i)}$ of the code words x and y, and yields $z^{(i)}$, the information part of the result z. The second part operates on the check parts $x^{(c)}$ and $y^{(c)}$ and computes their mean value $z^{(c)}$. The least significant sum bit d_1 is the rest of the division $(x^{(c)} + y^{(c)})/2$. The two D-flip-flops have to be initialized complementary. For the signals in an AC the following equations hold:

$$w(z^{(i)}) + \frac{d_0}{2} = \frac{w(x^{(i)}) + w(y^{(i)}) + c^{(i)}_0}{2}$$

$$z^{(c)} + \frac{d_1}{2} = \frac{x^{(c)} + y^{(c)} + c^{(c)}_0}{2} \bmod 2^{k-1}.$$

An AC maps two words $x, y \in B_1(n, k)$ to a word $z \in B_1(n, k-1)$ and a two-rail signal $d_0 d_1 \in \{01, 10\}$. An AC also maps two words $x, y \in B_2(n, k)$ with $k \geq 4$ to a word $z \in B_1(n, k - 2)$ [4].

For all considered codes, if either x or y is a noncode word then z is a noncode word or $d_0 d_1$ is not a two-rail

1530-1591/03 $17.00 © 2003 IEEE

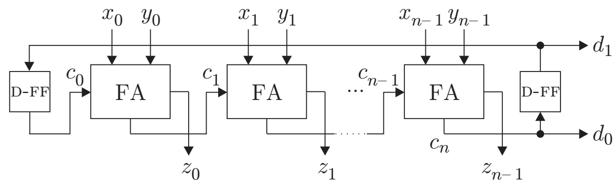

Figure 1: Weight averaging circuit WAC.

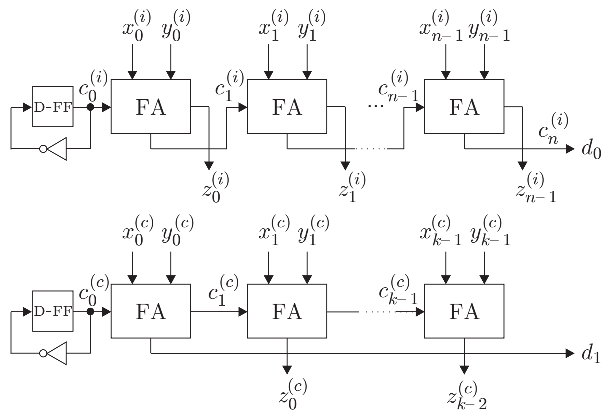

Figure 2: Averaging circuit AC.

signal or both, except for one case: If the most significant check bit $x_k^{(c)}$ (or $y_k^{(c)}$) of a Bose-Lin code word with $k \geq 4$ is affected by a single bit error then this would not be indicated by the averaging circuit. Fortunately, in $B_2(n, k)$ the two most significant check bits are always complementary which can be checked by a two-rail checker.

The proposed architecture of a Borden code checker is shown in Figure 3. The checker works as follows. x is the n bit word that the checker gets from the circuit under check, $x \in B(n, 2^k - 1)$. The n-bit word y is generated by a ring counter (RC) that is initialized with a word of $B(n, 2^k - 1)$ that contains at least one 0 and at least one 1. Therefore, the RC generates a subset of $B(n, 2^k - 1)$. Both words, x and y are the inputs of the weight averaging circuit WAC that computes $z \in B(n, 2^{k-1} - 1)$. z is then the input of a checker for Borden code $B(n, 2^{k-1} - 1)$, that is designed in the same way as the checker that is just explained. For $B(n, 1)$ this checker is a parity checker. The output of this checker and the two-rail signal $d = d_0 d_1$ are fed into the two-rail checker TRC.

We assume that y is a code word of $B(n, 2^k - 1)$. If x is a word of $B(n, 2^k - 1)$ then z is a word of $B(n, 2^{k-1} - 1)$. Both, the output of the checker for $B(n, 2^{k-1} - 1)$ and the output $d = d_0 d_1$ of the weight averaging circuit are two-rail signals which is confirmed by the two-rail checker. If x does not belong to $B(n, 2^k - 1)$ then either $z \notin B(n, 2^{k-1} - 1)$ or/and d is not a two-rail signal. The two-rail checker will get a non-two-rail code word, and an error is indicated.

The architecture of a Bose code checker is very similar to the proposed Borden code checker architecture and also works similarly. The ring counter is replaced by a Bose code word generator (CWG) which is shown in Figure 4,

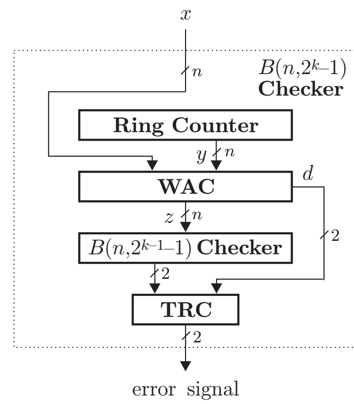

Figure 3: Embedded Borden code checker.

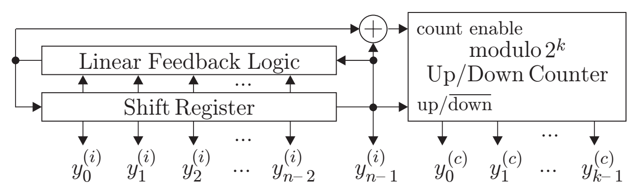

Figure 4: Bose-Lin/Bose code word generator.

the WAC is replaced by an AC, and the $B(n, 2^{k-1} - 1)$ checker is replaced by a $B_1(n, k - 1)$ checker. The Bose CWG generates all words of $B_1(n, k)$, except the all-0 information bit vector and the corresponding check symbol. The checker for Bose code $B_1(n, k-1)$ has the same structure as the checker that is just explained, but for $k - 1$ instead of k check bits. For k = 2 it is a parity checker.

The design of a Bose-Lin code checker is similar to that of a Bose code checker with the following differences. The $B_1(n, k-1)$ checker is replaced by a $B_1(n, k-2)$ checker and the 4-bit TRC is replaced by a 6-bit TRC in order to check also the two most significant check bits of x. The CWG has to be initialized with a Bose-Lin code word. It generates Bose-Lin code words and also noncode words in which the check symbols $y^{(c)}$ differ from the correct check symbols by $\pm 2^{k-1}$. The most significant counter bit $y_{k-1}^{(c)}$ of the CWG is duplicated and inverted.

All three checker architectures are self-testing with respect to every single stuck-at fault, provided that no checker input line gets a constant signal and the code words occur randomly. Depending on the code parameters they can be tested with only 2 or 3 code words [4].

References

[1] J.M. Borden, "Optimal Asymmetric Error Detecting Codes," *Information and Control*, Vol. 53, April 1982, pp. 66–73.

[2] B. Bose, "Burst Unidirectional Error-Detecting Codes," *IEEE Trans. Computers*, Vol. C-35, No. 4, April 1986, pp. 350-353.

[3] B. Bose and D.J. Lin, "Systematic Unidirectional Error-Detecting Codes," *IEEE Transactions on Computers*, Vol. C-34, No. 11, November 1985, pp. 1026–1032.

[4] S. Tarnick, *Self-Testing Embedded Checkers for Bose-Lin, Bose, and a Class of Borden Codes*, Research Report, SATCON GmbH, Teltow, Germany, 2002.