Fault-Secure Interface Between Fault-Tolerant RAM and Transmission Channel Using Systematic Cyclic Codes

Fabrice Monteiro, Stanisław J. Piestrak, Houssein Jaber, and Abbas Dandache
LICM, University of Metz, 7 rue Marconi, 57070 Metz, France
{fabrice.monteiro, piestrak, abbas.dandache}@univ-metz.fr, hussein_jjaber@hotmail.com

Abstract

The problem of designing a fault-secure interface between a fault-tolerant RAM memory system and a transmission channel, both protected against errors using cyclic linear error detecting and/or correcting codes is considered. The main idea relies on using the RAM check bits to control the correct operation of the parallel cyclic code encoder, so that the whole interface has no single point of failure.

1 Introduction

Error detecting codes (EDC) and error-correcting codes (ECCs) have been used for years to increase the reliability of RAM memory [1], [2] and transmission systems [3]. On one hand, high throughput requirements and clocking frequencies that should match those of a microprocessor with which embedded RAM communicates, imply parallel read/write of memory words, each protected by their own checkbits. Nowadays the major source of errors in semiconductor RAM memory systems are temporary faults caused by cosmic radiation (called single event upsets (SEUs)) [4]. On the other hand, the data are often transmitted as large data blocks composed of several memory words forming a single block protected by common check bits. The nature of noise occurring in a transmission channel often is a source of multiple-bit (burst) errors, so that the error model assumed is different and the error-correcting capabilities of codes applied to transmit data generally differ from those used to protect RAM. To date, each of these two important classes of digital systems has been protected separately by some ECCs. Surprisingly, no care has ever been taken to protect an interface between a fault-tolerant memory system and a transmission channel (see Fig. 1).

A typical fault-tolerant RAM memory system is composed of two parts (see Fig. 1): (i) the memory cells storing the data protected by the EDC and/or ECC and (ii) the error detection and correction (EDAC) circuit which verifies the correctness of data read from RAM and, if needed, corrects detected errors. Then, the memory check bits are simply ignored and only the data part is sent back to the processor or to the transmission channel. In the latter case however, before being transmitted, the data are encoded again by some other ECC (different from the one used in RAM), which is more efficient to protect data against transmission errors.

Figure 1. Unprotected interface between fault-tolerant RAM and a transmission channel

An encoder is a critical circuit for the correct (error-free) functioning of a transmission channel, since some errors which could be introduced by faulty circuitry can go undetected, unless it is provided with some concurrent error detection means. A faulty encoder may introduce errors to otherwise correct data read from RAM, so that erroneous data with correctly generated check bits can be transmitted over the channel, with no possibility for error detection or/and correction. Errors can also occur on an unprotected bus between the EDAC circuit and the encoder. Obviously, all these errors cannot be detected, unless the whole interface (the bus and the encoder itself) is implemented as fault-secure. Similarly, internal faults of an unprotected decoder can result in erroneous data at the receiver, despite no errors have occurred during transmission. To date, self-checking encoders for cyclic redundancy check (CRC) and Reed-Solomon codes were proposed in [5] and [6] only.

Here we will present, for the first time in the open literature, how to design a complete fault-secure interface between processor’s RAM memory and the transmission channel which has no single point of failure and which requires less redundancy than duplication with comparison.
2 General Scheme of Fault-Secure Encoder

The FS encoder proposed here (Fig. 2), could be applied to any cyclic linear EDC or ECC used for data transmission (like parity, CRC, and BCH codes). It is assumed that ECC\_RAM is any cyclic linear code generated by the polynomial $G'(X)$ of degree $(m - p)$. This is because, we have found that the cyclic property of the code is essential to avoid an excessive hardware amount needed to implement a parallel FS encoder. Any ECC\_RAM codeword $W_i$ is the concatenation of the $p$-bit data part $Y_i$ and the $(m - p)$-bit check part $Z_i$, i.e. $W_i = (Y_i, Z_i)$ has $m$ bits. In polynomial notation, we have:

$$W_i(X) = Y_i(X) \cdot X^{m-p} + Z_i(X),$$

$$Z_i(X) = (Y_i(X) \cdot X^{m-p}) \mod G'(X).$$

Any data part $D(X)$ of the codeword of ECC\_Transm is a block of $l$ data parts of RAM codewords $Y_i(X)$ ($0 \leq i \leq l - 1$), i.e. $k = l \cdot p$. In polynomial notation, the data stream $D(X)$ is the concatenation of some data slices $D_i(X)$ of $p$ contiguous bits

$$D(X) = \sum_{i=0}^{l-1} D_{up}(X) \cdot X^{(l-1-i)p}, \quad k = lp,$$

where $D_i(X) = d_{k-l-i-p} + d_{k-l-p+1}X + \cdots + d_{k-i-1}X^{p-1}$ and $d_i = 0$ if $i \notin \{0, \ldots, k - 1\}$.

![Figure 2. Fault-secure interface](image)

The transmitted ECC codeword $C(X)$ is the concatenation of the $k$-bit data part $D(X)$ and the $(n - k)$-bit check part $R(X)$, i.e. $C(X) = D(X) \cdot X^{n-k} + R(X)$. The check part $R(X)$ is generated using a generator mod $G(X)$, which is the only common block for both unprotected and FS encoder (although for an FS encoder, it is extended to generate in parallel the vector $R'(X)$ needed for FS operation).

The FS version of an encoder consists of three extra blocks: (i) the RAM check parts accumulator that generates the reference check part $R'(X)$ from $l \cdot (m - p)$-bit check parts $Z_i$ read from RAM; (ii) the generator of reference check bits $R''(X)$ (which is nothing else but the divider of $[R(X) + R'(X)] \mod G'(X)$); and (iii) the self-checking comparator [1] that detects any disagreement between $R'(X)$ and $R''(X)$ in every clock cycle. The FS property of the whole circuit can be proved easily on the basis that there are two different independent paths from the fault-tolerant RAM to the inputs of the comparator. Obviously, all faults that affect a single line of a the comparator are detected, since it is self-checking.

3 Complexity and Performance Evaluation

Table 1 compares the complexity (the number of complex logic blocks (CLBs)) of simple (S) and fault-secure (FS) versions of $p$-parallel FS encoders for sample linear cyclic codes, obtained by synthesising various encoders using Altera’s Quartus II software for the Stratix II FPGA circuit. Two groups of figures were obtained for two classes of CRC transmission codes generated by standard polynomials: $G_1(X) = 1 + X^2 + X^5 + X^{16}$ and $G_2(X) = 1 + X + X^3 + X^4 + X^5 + X^6 + X^7 + X^{10} + X^{11} + X^{14} + X^{17} + X^{18} + X^{23} + X^{24}$, respectively. It was assumed that the fault-tolerant RAM system with $p = 8, 16$, and $32$ data bits was protected by simple parity code for $G_1(X)$ and by cyclic Hamming single error correcting codes generated by the polynomials given in the table for $G_2(X)$. For parity protected RAM the hardware overhead of a FS decoder is only slightly more than 30%, whereas for Hamming codes it is relatively high: from about 69 to 79%, although it still offers savings compared to duplication.

![Table 1. Hardware complexity comparison.](image)

References


