Coding of Checksum components for Increasing the Control Reliability of Data Transmission for Military Applications

NIKOLAOS BARDIS
University of Military Education
[1]Hellenic Army Academy, [2]Hellenic Naval Academy, [3]Hellenic Air Force Academy
Department of Computer Sciences
[1]Vari - 16673, [2]Terma Hadjikyriakou Avenue, Piraeus - 18539, [3]Dekelia Air Base, Tatoi, Metamorfosi 144 51, Greece
bardis@ieee.org

Abstract: -Computer Systems used in Military and other challenging applications, are often exposed to increased levels of electromagnetic radiation. Embedded systems falling in this category often suffer from this exposure due to the operation of the device to which they belong. Consequently data communications within such devices need to be protected against transmission errors. Current general purpose encoding schemes that are used in other communication applications are prohibitively complex for this application. In this paper an innovative extension to the well known checksum concept is proposed that is capable of controlling errors in intra device data transfers. The new technique is shown to be simple enough for implementation and to increase the probability of detection of errors by several orders of magnitude. The scheme is hence shown to be suitable for embedded computing platforms for military and other demanding systems. More specifically, the modified checksum is examined in respect of its suitability for use in schemes for the transient error detection in schemes for reliable data storage within computing systems and it is explained why it is extremely suitable for this application. The modified checksum is also considered in the context of algorithm based fault tolerance schemes and it is again concluded that it can contribute to the overall scheme efficiency and effectiveness. The modified checksum is hence shown to be an algorithmic tool that can significantly contribute to the design of reliable and fault tolerant computing systems, such as the ones used for military systems or other applications that operate in adverse environments.

Key-Words: - Checksum, differential Boolean transformations, error detection, data integrity, fault tolerance

1 Introduction

Computing systems have become an essential, integral component of many aspects of modern society such as military, financial, commercial, and security activities. This fact is particularly important for the military, where secure access to reliable information is absolutely essential and can literally mean the difference between life and death. Military operations involve a multitude of networks and systems operating in different duress conditions, with a wide variety of information needs, objectives, and constraints. Computer system integrity thus acquires an increased significance [12]. Technogenic risks are present on other kinds of activities as well, ranging from home and entertainment appliances to means of transport such as automobiles and airplanes. This is true because all these activities rely on IT devices for their completion. Computer systems on their part critically depend on effective and reliable storage and transmission of information.

The problems surrounding this reliable storage and transmission are well understood for the case of data transfers between computer systems that are physically separated and form computer networks and solutions have been proposed (e.g. [13]).

Modern computer systems are constructed to achieve ever increasing data processing rates. Additionally, they are extremely widely used as embedded components in larger systems. Embedded operation implies exposure to electromagnetic interference form the actual devices in which they operate. There is hence an acute problem of information integrity arising. The speed and interference considerations mentioned earlier on lead to an increase in the probability of errors occurring during information transmission [10]. [2, 11]. The fact that existing Single Error Correction, Double Error Detection codes (SED - DED) are inadequate for current, high reliability applications has been well understood and documented in literature [15]. Solutions that have been proposed include the use of Reed – Solomon codes,

Technological advances in the fields of semiconductors and digital design have permitted an increase in the reliability of all components of the computer systems. However, if the reliability during the transmission and the storage of information is jeopardized, all these advances are at vain and hence measures need to be taken [3]. The continuous increases in the data processing rates involved impose effectiveness constraints on the relevant error control schemes that are becoming increasingly strict and difficult to be met [10]. This effectiveness must be commensurate with the channel characteristics as the bandwidth and the rate of transmitted data. This condition determines the needs for a radical increase of the error control reliability of the means of error control. The increase in processing speed cannot jointly compensate both for the increases of the user requirements and the increased complexity due to error control. Schemes are therefore needed that allow for low in complexity, possibly parallel processing implementations. [7].

Thus, these characteristics specify the urgency and practical importance of the new development and the improvement of the known means for achieving high reliability during data transmission and data storage in the computer systems and telecommunications networks.

In this paper, a new scheme is proposed that is capable of detecting multiple errors in data with very high probability. The new scheme is based on the well known concept of the checksum (c.f. [14]). More specifically this paper is organized as follows. In section 2 the problems surrounding error detection in general and more specifically in the context of data transfers between the processor and peripheral storage devices like the main memory, the hard disk etc. In section 3 existing techniques for error detection during communication are outlined that are based on the checksum principle. The drawbacks and limitations of existing techniques are explained. The use of a new function for calculating a modified checksum is proposed. The modified checksum is shown to be equivalent in the computational complexity it requires, while reducing the probability of errors escaping undetected by several orders of magnitude. In section 4 the concept of Forward Error Correction (FEC) is explained and the use of the Hamming code as a means of achieving this correction is outlined. The basis of an extension of the proposed scheme in the context of FEC is given. In section 5 the application of the new scheme in the context of data storage and transmission is studied and the overheads imposed upon the required processing are assessed. It is hence established that the modified scheme is suitable for use in embedded computing systems and more specifically for systems operating in the adverse environment conditions that exist for military applications. In Section 6 the application of the modified checksum in the context of Algorithm Based Fault Tolerance is considered. Existing algorithm based fault tolerance schemes are examined and the basis of their operation is explained. They are found to require checksums that are capable of reliably detecting errors and requiring a small computational complexity overhead. The modified checksum is hence shown to be suitable as an enabling technology for software based fault tolerant systems. Conclusions are finally drawn and directions for future work are given.

2 Analysis of error detection during data transmission and storage

For guaranteeing the reliable data transmission in communication channels of computer networks a large number of means is used, of which important place occupies the coding of the transmitted information. In the majority of systems the transmission and the storage of information are carried out by blocks and respectively the integrity data transmission or data storage of each block is controlled separately [23].

With the use of special coding it is possible to distinguish two approaches for the correction of the appearing errors:

- the error detection by special codes and their correction by retransmitting the block upon request during the error detection (ARQ-Automatic Repeat Request);

- the correction of the appearing errors due to applications of correcting codes without the repeated transmission (FEC-Forward Error Correction).

The error detection – ARQ approach cannot be applied for the error correction during data storage for simple practical reasons; data are either primary and have just been produced or they are located inside their storage location and are therefore unique with no possibility for retransmission or other equivalent process. The main advantage of ARQ besides the diagrams of the FEC lies in the fact that the error detection requires simpler decoding hardware and smaller redundancy than the error correction method. The implementation of error detection has substantial smaller computational complexity which makes it possible to calculate the error control functions considerably faster.

Furthermore, the effectiveness of ARQ is less and depends on the multiplicity of the appearing errors. This effectiveness I

The choice between the two approaches for the elimination of the appearing errors depends on intensity and the nature of the appearing errors. The basic sources of errors in digital data channels are the inter bit interferences, the externally produced noise and the thermal noise of transmission means [2], [11].

The nature of the appearing errors depends not only on their source, but also on the type of transmission means and on modulation of signals method. Thus, in the ether communication channels the prevailing source of the transmitted errors is the externally produced noise and in this case the intensity of the appearing errors is sufficiently high so that the application of FEC technologies proves to be more preferable. In the wire systems of digital data transmission, in which the intensity of errors is several orders lower in comparison with the wireless channels, the use of ARQ is considered to be more effective [4].

In the cable channels with the sequential data transmission without modulation, the transmitted error has the same nature, and the channels themselves correspond to the binary symmetrical channel model. This model assumes the appearance of erroneous transmission of zero or one with equal probabilities, since the probability p_j that j errors occur during the transmission of the n-bit code is determined for the binary symmetrical channel by the expression [23]:

$$p_m = \binom{n}{j} \cdot p^j \cdot (1-p)^{n-j} \quad (1)$$

where p is the probability of the erroneous transmission of one bit.

For errors detection by the per block data transmission the CRC codes and the CS are used more often [2, 11]. The CS method in comparison with CRC is substantially simpler and ensures the maximum rate of the error control and it is an influential factor on a constant increase in the channel capacity of data transmission.

In contrast to CRC, the structure of the operations which are performed with the check sum calculation, it allows the parallel process which makes it possible to implement effectively this control by hardware, so that the time for the calculation of error control practically will not affect the performance of the data transmission.

Let us denote by $D_1,D_2,\ldots,D_k$ the k codes with n-bit size, which compose the transmission block, and by $D_1',D_2',\ldots,D_k'$ we denote the blocks on the receiver end. Every code D contains n bits: $D=\{d_1,d_2,\ldots,d_n\}$, where $d_1,\ldots,d_n \in \{0,1\}$. The CS's on the receiver and the sender are calculated with the same way: $S_S=D_1 \oplus D_2 \oplus \ldots \oplus D_k$ and $S_R=D_1' \oplus D_2' \oplus \ldots \oplus D_k'$.

The usual check sum assumes that the data transmission is carried out as a block and organized as k codes $D_1,D_2,\ldots,D_k$ with length n bits. In this case the length n of the code is determined by the architecture of the control organization and its value can coincide with the number of simultaneously transferred bits, and it can differ from it.

At the end of the transmission of the data block, the transmitter sends to the receiver the check sum S_S, which is XORed with the check sum that is calculated on the receiver S_R and obtains the differential code $\Delta=S_S \oplus S_R$ of size n. If $\Delta=0$ then we consider that no errors have arisen. For the symmetrical binary channel and, taking into account that in practice the relation $k >> n$ holds, then we can consider that during the transmission of one code only one error can arise.

The low reliability of the error detection of the even error multiplicity is the main disadvantage of the check sum. Actually, the most probable occurrence between them is the two-fold error (situation of appearance of single errors in two from k transmitted codes) and the code Δ can attain only n^2 different values of all the 2^n possible. It is means that for the studied model the usual check sum is ineffectively coding of two-fold error. Because of this the probability P_2 of the non detection of the two-fold error is reasonable high and determined by formula [2], [11], [23]:

$$P_2 = \frac{1}{n} \quad (2)$$

Thus, the reliability level of the error detection when using the check sum can increase due to the coding optimization, i.e., the calculation of check sums on the transmitter and the receiver in the form: $S_S=F(D_1) \oplus F(D_2) \oplus \ldots \oplus F(D_k)$ and $S_R=F(D_1') \oplus F(D_2') \oplus \ldots \oplus F(D_k')$, where F is the function of coding, defined by the system of Boolean functions.

In the paper [1], [9] an orthogonal system of Boolean functions that satisfy the Strict Avalanche Criterion (SAC) has been proposed to use as coding function.

Usually these types of Boolean transformations are used in cryptographic algorithms and their design methods have been developed in [1], [2].

A Boolean function $f(x_1,\ldots,x_n)$ defined on a set Z of all possible 2^n n-tuples of n variables, satisfies the SAC, if a complement of a single incoming n-tuple data bit changes the output of the Boolean function with probability 50%:

$$\forall j \in \{1,...,n\}: \sum_{x_1,...,x_n \in Z} (f(x_1,...,x_j,...,x_n) \oplus f(x_1,...,\overline{x}_j,...,x_n)) = 2^{n-1} \quad (3)$$

If one of the n inputs of the avalanche transformation is changed then half of its outputs will be changed. This means, that there is an "avalanche amplifier" which by changing one of the n-tuple incoming data bit transforms half of the outputs. Because every function of this system satisfies the Avalanche Criterion, these transformations are called "avalanche".

Let's denote with $F(D)$, the Boolean orthogonal avalanche transformation on the n-bits code D. So transformation $F(D)$ consist of orthogonal Boolean functions $f_1(D),f_2(D),\ldots,f_n(D)$, every of which satisfies the Avalanche Criterion. The length of the transformed code $R=F(D)$ is n bits long, as well. The orthogonality of the $F(D)$ transformation indicates the one-to-one correspondence of codes D and R. The avalanche properties of the $F(D)$ transformation indicate that if one bit of the input code D is changed then, on average, $n/2$ bits of the output code $R=F(D)$ will be changed also.

Examle of the Boolean orthogonal avalanche transformation $F(D)$ on the 8-bits code D (n=8) is given below:

$f_1(X) = d_8d_7 + d_8d_6 + d_8d_5 + d_8d_4 + d_8d_2 + d_8d_1 + d_7d_5 + + d_7d_4 + d_7d_2 + d_6d_3 + d_5d_3 + d_5d_1 + d_1$

$f_2(D) = d_8d_6 + d_8d_5 + d_8d_3 + d_8d_1 + d_7d_6 + d_7d_4 + d_7d_2 + d_7d_1 + + d_6d_4 + d_6d_2 + d_5d_4 + d_5d_2 + d_5d_1 + d_4d_1 + d_3d_1 + d_2d_1 + d_2$

$f_3(D) = d_8d_7 + d_8d_5 + d_8d_3 + d_8d_2 + d_7d_6 + d_7d_5 + d_7d_3 + d_7d_2 + + d_7d_1 + d_6d_5 + d_6d_3 + d_6d_2 + d_5d_4 + d_4d_2 + d_3d_2 + d_3 + d_2d_1$

$f_4(D) = d_8d_7 + d_8d_6 + d_8d_4 + d_8d_3 + d_8d_2 + d_8d_1 + d_7d_4 + d_7d_3 + d_7d_2 + d_6d_5 + d_5d_4 + d_5d_3 + d_5d_2 + d_4d_3 + d_4 + d_1d_2 + d_3d_1$

$f_5(D = d_8d_5 + d_8d_4 + d_8d_3 + d_8d_1 + d_7d_2 + d_7d_1 + d_6d_5 + d_6d_4 + + d_6d_3 + d_6d_1 + d_5d_2 + d_5d_1 + d_5 + d_4d_3 + d_4d_2 + d_3d_1 + d_2d_1$

$f_6(D) = d_8d_3 + d_8d_2 + d_7d_5 + d_7d_4 + d_6d_3 + d_6d_2 + d_6 + d_5d_3 + d_5d_2 + + d_5d_1 + d_4d_2 + d_3d_2 + d_2d_1$

$f_7(D) = d_8d_6 + d_8d_5 + d_7d_6 + d_7d_4 + d_7d_3 + d_7d_2 + d_7 + d_6d_4 + d_6d_3 + d_6d_2 + d_6d_1 + d_5d_4 + d_5d_3 + d_5d_2 + d_4d_3 + d_3d_2 + d_3d_1$

$f_8(D) = d_8d_7 + d_8d_5 + d_8d_4 + d_8d_3 + d_8 + d_7d_6 + d_7d_5 + d_7d_4 + d_7d_3 + d_7d_1 + d_6d_5 + d_6d_4 + d_6d_3 + d_5d_2 + d_4d_3 + d_4d_2 + d_4d_1$

Thus, if a single error appears, then $n/2$ bits of the modified checksum will change. If a second error appears then another $n/2$ bits of the modified checksum will change. It is clear that the probability of the masking interaction of $n/2$ erroneous bit pairs is less than the probability of the masking interaction of a single bit pair.

It has been shown that the probability P_{2f} that the dual bit erors will not be detected in case orthogonal avalanche transformation $F(D)$ using is determined as follows:

$$P_{2f} = \frac{1}{\binom{n}{n/2}} = \frac{((n/2)!)^2}{n!} = \prod_{j=0}^{n/2-1} \frac{j+1}{(n-j)} \quad (4)$$

Thus, the probability of detecting dual errors during block transmission using the checksum control shceme orthogonal avalanche transformation $F(D)$, increases by t_2 times in comparion to the ordinary checksum scheme. The numerical value of the t_2 increase is determined by the formula:

$$t_2 = \prod_{j=1}^{n/2-1} \frac{n-j}{j+1} \quad (5)$$

For example, for n=8, the probability that the dual errors will not be detected is decreased by 8.7 times in comparison to the traditional checksum.

In case orthogonal avalanche transformation $F(D)$ using, with the appearance of the two-fold error the code Δ has $n!/((n/2)!)^2$ different values and correspondingly the probability of the non detection of the two-fold error substantially decreases in comparison with the usual check sum.

However in this case the $n!/((n/2)!)^2$ coding variants of the two-fold error are substantially less than the total number of all possible codes $\Delta - 2^n$ and therefore in this case the optimization of coding is not achieved. Consequently, an increase in the reliability of detection of prevailing type errors by check sum can be achieved due to further optimization of its coding via the selection of corresponding functional transformation.

The purpose of this approach is to increase the reliability of error control by using check sum due to development of the functional transformations which optimize its coding for the prevailing forms of errors in the binary symmetrical channel.

3 A novel optimization for reliable data flow based on checksum coding

For the practical implementation of error detection based on coding check sum optimization for detecting the appearing errors a calculation method of the modified check sum is proposed, similarly as in work [5]. In this proposed method as terms are used codes which are obtained from Boolean transformations over the controlled codes. These Boolean transformations consist of a system of *m* Boolean functions with *n* variables:

$$F(D) = \{f_1(D), f_2(D), ..., f_m(D)\} \quad (6)$$

where D is an *n*-bit code: $D=\{d_1,d_2,\ldots,d_n\}$, $\forall j \in \{1,\ldots,n\}: d_j \in \{0,1\}$.

With the appearance of a single error in the j^{th} bit of the code D_i it is transformed into $D_i' = \{d_1,\ldots,d_j \oplus 1,\ldots,d_n\}$ and the differential Δ of the check sum can be represented in the form of the differentials values of the functions $f_1,f_2,\ldots,f_m$ with the variable d_j on the binary tuples $\{d_1,d_2,\ldots,d_{j-1},d_{j+1},\ldots,d_n\}$:

$$\begin{aligned}\Delta &= F(D_i) \oplus F(D_i') = \\ &\{f_1(D_i) \oplus f_1(D_i'), ..., f_m(D_i) \oplus f_m(D_i')\} = \\ &= \{\frac{\partial f_1}{\partial d_j}, \frac{\partial f_2}{\partial d_j}, ..., \frac{\partial f_m}{\partial d_j}\}\end{aligned} \quad (7)$$

The optimization of single error coding in the modified *m* bits check sum can be achieved, if the number of possible values of the code of Δ equals 2^m. Since the number of versions of the single error localization in the code *D* is equal to *n*, so that the single error could be one way coded by check sum it is enough that $m = \lceil \log_2 n \rceil$ holds. In this case the binary code formed by a change in the functions with the appearance of error in the j^{th} bit of the code *D*, i.e., with a change in the variable d_j, is equal to j-1:

$$\forall j \in \{1,...,n\}: \sum_{t=0}^{\lceil \log_2 n \rceil - 1} \frac{\partial f_t}{\partial d_j} \cdot 2^t = j - 1 \quad (8)$$

It is obvious that the condition (8) is satisfied if each of the functions $f_1,f_2,\ldots,f_m$ is linear, and the *q* function f_q includes the variable d_j (i.e., the value of the q bit of the binary number j-1 is equal to one). For example, if *n=8*, then *m=3* and the system of functions which satisfy (7) can be as follows:

$$\begin{aligned} f_1 &= d_2 \oplus d_4 \oplus d_6 \oplus d_8 \\ f_2 &= d_3 \oplus d_4 \oplus d_7 \oplus d_8 \\ f_3 &= d_5 \oplus d_6 \oplus d_7 \oplus d_8 \end{aligned} \quad (9)$$

In this case, the length size of check sum of its coding is substantially lower than the code length size of Δ: $m<n$, however, the probability that the error of any multiplicity larger than one (single errors they are detected always) corresponds to expression (2). For example, the two-fold error is not detected only when both errors occurred in one and the same bit.

In order to decrease the probability of not detecting the multiplicity errors, it is necessary in addition of the Boolean system (8) which forms the set Ξ_1, to use a system of *u* functions which compose the set Ξ_2 so that $F=\{\Xi_1, \Xi_2\}$.

In order to detect the two-fold error with high reliability it is necessary that a number of conditions are fulfilled. Since two errors, localized in different bits of the transferred codes are always detected using the functions of the set Ξ_1, and so for detecting the errors which appear in the same bit on different codes of block it is necessary that the probability of the values agreeing of the differentials functions of the set Ξ_2 with the change of one variable, should be as small as possible or near to zero.

Therefore the functions differentials of this set must not be constant, i.e., the functions must be nonlinear, moreover the functions differentials $f_{m+1},f_{m+2},\ldots, f_{m+u}$, on any of the variables must constitute an orthogonal system of functions.

4 The Modified Checksum and FEC

Forward error correction (FEC) is the twin technology of Error Detection. The objective of FEC is not only to detect errors but also to correct them so that there is no need for information retransmission. More specifically, the aim of FEC is to allow the receiving station to correct transmission errors without having to ask the transmitting station to repeat the frame. Hence, the possible throughput loss due to the redundancy that needs to be introduced by appending length to the useful data, is compensated for by the reduced number of channel exchanges for frame repeats. A well understood FEC code is the Hamming code [15]. If the number of useful information bits in a packet is *k* and the number redundant bits is *l* then the total number of transmitted bits is *m=k+l* and the Hamming codeword is described as (*m,k,l*). The m bits need to be combined with a parity check vector capable of

protecting both the data bits and itself and hence needs to assume all distinct integer values from 0 to $m+l$ or $m+l+1$ distinct values. It is therefore the case that:

$$2^r > m + l + 1, \quad (10)$$

since r bits assume 2^r values. Equation (10) is called the Hamming rule [15].

Using these concepts, the (7, 4, 3) Hamming code occupies 7 load bits in order to protect the 4 useful data bits carried via the use of 3 redundant bits. This protection is effective only against single errors. The hamming code is widely used in telecommunications to protect data against data corruption in noisy channels, in computers to improve the reliability of the data storage and transfer subsystems and in other applications such as data compression to protect from total data loss due to isolated bits being corrupted ([15], [16]). Further sources of errors can be possibly caused by disturbances to the effective operation of the data handling circuitry (i.e. the processing elements that handles the load of encoding or decoding the Hamming code's data protection) due to the effects of external factors. Such external factors include ageing of the devices due to their exposure to adverse conditions and effects of electromagnetic interference originating from sources external or even internal to the device itself [16]. This first group of error generating sources is of course not irrelevant to the safety critical applications that form the target of this research. It is however the second group and more specifically the fact that the interference may be generated from the devices themselves, that makes this type of applications more susceptible to errors and hence in greater need of measures protecting against this type of occurrences.

The construction of Hamming codes for both the encoding and decoding process requires the calculation of the extra control bits that will allow the detection of the errors. Different algorithms have been proposed for this calculation, such as [15] and [17]. On average, these will require e.g. for the (16, 11, 4) code the use of a 16×5 matrix [15] while the (7, 4, 3) code uses a 7×5 matrix. The principle drawback of existing Hamming FEC schemes is precisely this requirement for extensive use of matrix computations. Matrix computations require large amounts of memory and complex calculations for effective indexing. These overheads may be viably tolerated in contexts such as the relatively low speed low distance communications. However, in the case of communication between the main memory and the processor of a modern high clock rate computing system, such overheads may not be tolerable. In this case the number of transactions involved is increased by several orders of magnitude and hence the delay and strain in resources caused by the overheads involved is multiplied accordingly.

It is proposed that the modified checksum be included in the calculation of the Hamming codes in order to reduce the memory and indexing calculations involved. The penalty involved will be a marginal increase in the complexity of the functions that will need to be calculated. The introduction of the modified checksum in FEC schemes will be the topic of a separate publication to appear.

5 Application to reliable storage

The computer industry and microprocessor manufacturers have been continuously pursuing for several decades the design of new devices that are smaller in size and require less power for their operation than their previous products. Such characteristics though imply that newer devices are more susceptible to errors during the data transfers between subsystems due to transient interference [19]. Subsystems that may be exposed to such dangers include the processor, the main memory, the on-chip cache memory, hard disks, DRAMs, and virtually all devices that process or store data. Interference may be caused from external factors like electromagnetic radiation originating from any source from cosmic radiation to adjacent components of the device itself or even the packaging of the circuitry itself [19].

Manufacturers take precautions in order to prevent such occurrences and hence their probability of appearance becomes small but nevertheless non-zero. This fact leads researchers to study various special cases of such errors that are more likely to be encountered. Such cases are an isolated bit being corrupted and multiple erroneous bits appearing in the form of a burst ([14], [15]). The single bit event is usually called single error disruption (SED) or single error upset (SEU) [20]. Elementary probabilistic calculations lead to the conclusion that the event of encountering multiple, randomly distributed errors appears with probability that is exponentially decreasing with the number of such bits.

The reason that the burst error event is separately studied is the fact that an external source of interference might be prolonged in duration and might hence extend for a period much longer than a single transmitted bit. This fact makes the burst

error event more likely than the event of an equal number of erroneous, randomly distributed bits. According to the analysis presented in [19], the rate at which spatial multi – bit errors increases in every new generation of devices and as these devices are forced to fit in smaller space. This is true because a single strike of radiation will effect multiple circuits (e.g. memory cells) simultaneously and is more likely to produce a higher number of erroneous bits. With current 90 nm and 130 nm technologies double bit spatial errors represent between 1% and 2% of all the errors encountered in SRAMs [19]. This figure is likely to increase as the density of the devices on the integrated circuits increases.

The occurrence of an SEU is possible to appear in the context of program instruction words stored in main memory or even while they are inside the processor being encoded [14]. In this event and more particularly in safety critical systems like navigation, guidance, military etc, such an eventuality could have catastrophic consequences. Systems of these types are even more exposed to sources of interference, e.g. the paradigm of an aircraft flying in high altitude where it is exposed to high levels of radiation.

Given the catastrophic consequences described above, measures need to be taken to avert the possibility of an error escaping undetected. It is highly desirable that errors be corrected as well but this is feasible in more restrictive conditions than simple detection [20]. Most approaches will correct the errors in some cases and simply detect the errors in more demanding cases and there is hence no reason to predetermine which one of the two operations will be attempted. The use of error correcting codes (ECCs) is the most usual approach to solving this problem [19]. ECCs however require a large amount of overhead, which in the case of integrated circuitry is translated to a large penalty in silicon area. Considering as an example the case of a double error detect, double error correct code, this would require 14 bits in order to protect 64 bits of data, or an overhead of 22% that will be reflected to the area of the silicon used [19]. More complex schemes (targeted at correcting more errors) would require even more complex solutions. Thus any technological advances in compacting integrated circuits are reversed by the required error correction additional parts. Using layout techniques, such as interleaving, to avoid multi – bit errors does not solve this problem since again that would require more complex connection lines and data regrouping and the overall circuit latency and power consumption is again increased.

For single bit error correction, double bit error detection, the minimum number of overhead bits required is given by the Gilbert-Varshamov bound [14]. The standard approach taken is to find a code that will give a minimum Hamming distance of 3 between all possible code words. The standard code length is for this case $n = 2^m - 1$, where m is the number of redundant bits for the error correction code. As the number of bits increases, the overhead also increases at a much higher rate. The bound gives the number of bits required to support the minimum distance that has been prescribed. If the number of field is q, the length of the code is n, the number of parity checks is m and the minimum distance is d, this minimum distance is guaranteed if

$$\sum_{i=0}^{d-1} (q-1)^i \binom{n-1}{i} < q^m \qquad (11)$$

Using the formula of equation (10), it may be hence concluded that for 48 bits of useful data a minimum of 7 bits is required to produce a minimum distance of 3 between the codewords. It should also be noted that an error correction code is not obviously available and hence designers usually use the redundant bits as parity bits. The use of parity bits offers no possibility of error correction and is a scheme for just detecting the existence of errors [14].

The proposed modified checksum aims to facilitate the process of error prevention within storage systems in two ways. It firstly presents higher reliability in error detection. It is therefore the case that possible errors are less likely to escape undetected. The reliability in stopping errors has a particularly favourable effect in the case of highly critical systems, such as the ones used for military applications. System designs can be hence implemented with more ambitious targets, since the probability of catastrophic failures becomes accordingly smaller. The second improvement exists in terms of the computational complexity involved. As it was analysed earlier on, the additional complexity imposed upon information processing systems for the purpose of error detection and correction. This increase in complexity cannot be compensated for by increase in processor speed, as the problem being studied in this research involves operations that need to take place within the processor itself during the execution of each instruction. Increases in processor speed are therefore irrelevant since the number of

error control calculations to be made will increase accordingly. The quantification of the benefit of the introduction of the modified checksum within a high reliability storage scheme for military applications is a further aspect of this research that will be presented in future publications.

6 Algorithm Based Fault Tolerance

The checksum is additionally used within critical systems in order to protect against faults in other kinds of circuitry as well. The concept is called Algorithm Based Fault Tolerance (ABFT) and aims to use modifications in the application software in order to detect and effectively deal with errors due to faults of the underlying hardware [21]. The principle of operation of ABFT is to create codes corresponding to the data under processing and process these codes together with the data. These codes present linearity properties and are hence such that the code corresponding to the processed data is equal to the result of the processing of the individual data elements, or

$$E\left(F\left(d_1, d_2, ...d_n\right)\right) = F\left(E\left(d_1\right), E\left(d_2\right), ..., E\left(d_n\right)\right) \quad (12)$$

where *E(x)* is the encoding and *F(x)* is the useful data operation. As an example a simple matrix addition may be considered, e.g. $\underline{C} = \underline{A} + \underline{B}$. An ABFT approach might add an extra row of data to the matrices $\underline{A}$ and $\underline{B}$ that could for example have the sums of the corresponding columns as elements. The result of the addition of the augmented matrices now would also be an augmented matrix containing a row which would again contain as elements the sum of the columns of the matrix $\underline{A} + \underline{B}$. This result would indeed be observed only if there were no errors due to hardware failure during the process. This process is illustrated in **Figure 1**.

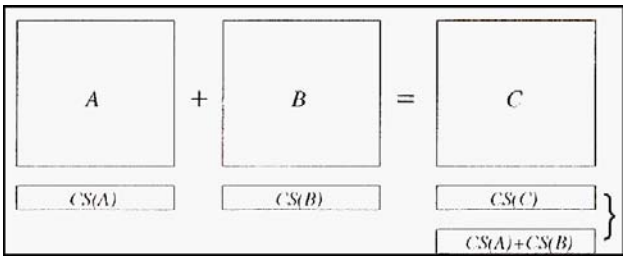

Figure 1: Simple matrix addition checksum example (after [21])

ABFT is a scheme by which computing system reliability is achieved by purely software means without any modifications to the hardware itself. By this method, faults may be corrected or at least detected that may exists in both the data paths and the processing elements involved. In this respect, ABFT may even be considered as a more effective instrument for providing fault tolerance than the hardware mechanisms described in the previous section.

ABFT was first introduced for the case of matrix multiplication executing on systolic hardware [21]. The systolic hardware was used to operate on the encoded data as well as the useful ones. ABFT is also employed for the case of LU factorisation of matrices and matrix inversion. Weighted checksums have been used in the context of data encoding. The sum of squares encoding was introduced to support the FFT and QR factorisation operations. ABFT schemes have been proposed to be applied not only to systolic hardware, but also to general purpose systems as well. These do not require hardware modifications and are purely software. Such schemes were again devised to be used for solving specific classes of problems like matrix multiplication, Gaussian elimination, FFT etc [21].

The software ABFT schemes need compiler support in order to achieve the desired results, by efficiently computing and transforming the checksums. The principle drawback of the ABFT schemes is that they require an additional complexity in order to guarantee reliability. Schemes have been proposed that improve slightly the efficiency by allowing the compiler to identify repeated calculations and reusing the relevant checksum codes [21]. Even though the increase in complexity may be justified due to the fact that reliability is an extremely important factor for safety critical applications, such as the military ones, increases in the efficiency of such systems is always welcome. It is in this respect that the modified checksum may be employed.

The modified checksum is highly suited to ABFT schemes, since it has already been shown to be very efficient in amplifying possible errors and will hence reduce the probability of such errors passing undetected. Furthermore, the modified checksum is also suitable for compiler generated optimisations and simple with respect to the complexity it imposes upon the overall system. The quantification of the modified checksum complexity within an ABFT scheme and the proposal of transformation checksum functions specifically for this purpose are currently being investigated and will be the topic of a future publication.

7 Conclusion

The proposed method is based on the coding optimization to increase the effectiveness of the error detection during data transmission and data

storage using the check sums. This makes it possible to significantly decrease the probability of the mutual masking of even multiplicity errors both in comparison with the usual check sum and the use of SAC transformations for coding of its components [1, 9].

One method for obtaining special differential Boolean functional transformations which optimize the check sum coding of the codes in block are developed. This method is developed from the point of view of the criterion of error detection which appears in the binary symmetrical channel. Examples of the transformations which optimize coding check sum for both the proposed methods are given.

The estimations of the probability of error detecting of different multiplicity are theoretically substantiated.

The structure of the functional transformation is considerably simpler in comparison with the transformations of the CRC method and allows multilevel parallel process on hardware implementation which makes it possible to ensure the high performance of the error control without delays in the process of data transmission [4, 6, 8].

The modified checksum has been shown to be a suitable technology on which schemes for dealing with transient errors in data storage and transfer can be based. This was shown to be true based on considerations regarding the particularities of this problem, the sensitivity in error detection required and the acceptable computational complexity overhead that may realistically be accepted. Similar considerations have shown that the modified checksum is appropriate as an enabling technology for designing fault tolerant computing systems based on the ABFT concept. Research is currently under way in order to complete the proposal for a modified checksum based coding scheme capable of performing FEC. Additionally, the benefits of the modified checksum application in the contexts of transient error management for data storage systems and software based ABFT schemes are being studied, so as to obtain quantitative results that will illustrate the merits of the new technique.

References:

[1] Bardis E.G., Markovskyy A.P. Utilization of of Avalanche Transformation for Increasing of Echoplex and Checksum Data Transmission Control Reliability // *2004 International Symposium on Information Theory and its Applications (ISITA-2004)*.-Parma, Italy, Okt 10-13, 2004.- pp.656-660.

[2] Klove T., Korzhik V. Error Detecting Codes: General Theory and Their Application in Feedback Communication Systems. *Norwell, MA: Kluwer, 1995*. – pp. 433.

[3] Saxena N.R., McCluskey E.J. Extended precision checksums. // *Proc.17-th Intern. Symp. Fault-Tolerant Comput. : FCTS-17,-Pittsburgrh(USA)*.-1987.-pp.142-147.

[4] Bardis N.G, Bardis E.G., Markovskyy A.P., C.Economou, Hardware Implementation of Data Transmission Control based on Boolean Transformation, *WSEAS TRANSACTIONS on COMMUNICATIONS*, Issue 7, Volume 4, ISSN 1109-2742, July 2005, pp: 363 – 371.

[5] Fletcher J. An Arithmetic Checksum for Serial Transmissions // *IEEE Transaction on Communication*.- 1983.- Vol. 30.- № 1.- pp.76-85.

[6] Albertengo G.A., Sisto R.J. Parallel CRC Generation // *IEEE Micro,-1990*.-Vol.11, № 10 - pp.84-91.

[7] Kounavis M.E., Berry F.L. A Systematic Approach to Building High Performance Software-Based CRC Generators.// *10th IEEE Symposium on Computers and Communications (ISCC'05)*, - 2005.- pp. 855-862.

[8] Partrige C., Hughes J., Stone J. Performance of Checksums and CRCs over Real Data // *Proc. SIGCOMM'95 Conf., ASM*, 1995- pp.68-76.

[9] Bardis N.G , Echoplex Error Control System using Avalanche Transformations, *WSEAS Transactions on Communications*, Issue 2, Volume 3, ISSN 1109-2742, April 2004. pp: 741 - 745.

[10] Shu Lin and Daniel J.Costello, Jr, Error Control Coding, *Prentice-Hall, Inc., Englewood Cliffs*, N.J.07632, ISBN 0-13-283796-X, 1983, 603p.

[11] Klove Torleiv, Codes for Error Detection, Serial on Coding Theory and Cryptography – Vol.2, *World Scientific, 2007*, pp. 201.

[12] Federal Business Opportunities Organization, *Request for Information (RFI) - Digital Object Storage and Retrieval (DOSR)*, Solicitation Number: SN08-33, Other Defense Agencies, Defense Advanced Research Projects Agency, Contracts Management Office, https://www.fbo.gov/index?tab=core&s=opportunity&mode=form&id=d163897a448169f22d285b0ad133ce4f.

[13] S. Chessa and P. Maestrini. Dependable and Secure Data Storage and Retrieval in Mobile, Wireless Networks. *Proceedings of the 2003*

International Conference on Dependable Systems and Networks (DSN'03)

[14] Bodnar, L.; Chapelle, G. A single error correction double burst error detection code. *Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on*. Volume 1, 9-12 Nov. 2003 Page(s):1118 - 1121 Vol.1

[15] Jianwu Zhao and Yibing Shi. A Novel Approach to Improving Burst Errors Correction Capability of Hamming Code. Communications, Circuits and Systems, 2007. ICCCAS 2007. International Conference on 11-13 July 2007 Page(s):1193 – 1196

[16] Cardarilli, G.C.; Ottavi, M.; Pontarelli, S.; Re, M.; Salsano, A. Data integrity evaluations of Reed Solomon codes for storage systems [solid state mass memories]. Defect and Fault Tolerance in VLSI Systems, 2004. DFT 2004. Proceedings. 19th IEEE International Symposium on, 10-13 Oct. 2004 Page(s):158 – 164

[17] Kui Cai, and Kees A. Schouhamer Immink. A General Construction of Constrained Parity-Check Codes for Optical Recording. IEEE Transactions On Communications, Vol. 56, No. 7, pp 1070-1079, July 2008

[18] Jack Keil Wolf. An Introduction to Tensor Product Codes and Applications to Digital Storage Systems. Proceedings of 2006 IEEE Information Theory Workshop (ITW'06) pp 6-10, 2006

[19] Bhattacharya, K.; Kim, S.; Ranganathan, N. Improving the reliability of on-chip L2 cache using redundancy. Computer Design, 2007. ICCD 2007. 25th International Conference on 7-10 Oct. 2007 Page(s): 224 – 229

[20] Schiano, L.; Ottavi, M.; Lombardi, F. Markov models of fault-tolerant memory systems under SEU. Memory Technology, Design and Testing, 2004. Records of the 2004 International Workshop on 9-10 Aug. 2004 Page(s): 38 – 43

[21] Chen, G.; Kandemir, M.; Karakoy, M. A data-centric approach to checksum reuse for array-intensive applications. Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on 28 June-1 July 2005 Page(s):316 – 325.

[22] Nikolaos Doukas, Nikolaos Bardis, Effectiveness Data Transmission Error Detection using Check Sum Control for Military Applications, 10th WSEAS Int. Conf. on MATHEMATICAL METHODS, COMPUTATIONAL TECHNIQUES AND INTELLIGENT SYSTEMS (MAMECTIS '08), Corfu Island, Greece, October 26-28, 2008.