World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:14, No:9, 2020

Error Detection and Correction for Onboard Satellite Computers Using Hamming Code

Rafsan Al Mamun, Md. Motaharul Islam, Rabana Tajrin, Nabiha Noor, Shafinaz Qader

Open Science Index, Electronics and Communication Engineering Vol:14, No:9, 2020 publications.waset.org/10011427.pdf

***Abstract*—**In an attempt to enrich the lives of billions of people by providing proper information, security and a way of communicating with others, the need for efficient and improved satellites is constantly growing. Thus, there is an increasing demand for better error detection and correction (EDAC) schemes, which are capable of protecting the data onboard the satellites. The paper is aimed towards detecting and correcting such errors using a special algorithm called the Hamming Code, which uses the concept of parity and parity bits to prevent single-bit errors onboard a satellite in Low Earth Orbit. This paper focuses on the study of Low Earth Orbit satellites and the process of generating the Hamming Code matrix to be used for EDAC using computer programs. The most effective version of Hamming Code generated was the Hamming (16, 11, 4) version using MATLAB, and the paper compares this particular scheme with other EDAC mechanisms, including other versions of Hamming Codes and Cyclic Redundancy Check (CRC), and the limitations of this scheme. This particular version of the Hamming Code guarantees single-bit error corrections as well as double-bit error detections. Furthermore, this version of Hamming Code has proved to be fast with a checking time of 5.669 nanoseconds, that has a relatively higher code rate and lower bit overhead compared to the other versions and can detect a greater percentage of errors per length of code than other EDAC schemes with similar capabilities. In conclusion, with the proper implementation of the system, it is quite possible to ensure a relatively uncorrupted satellite storage system.

***Keywords*—**Bit-flips, Hamming code, low earth orbit, parity bits, satellite, single error upset.

I. Introduction

SATELLITES have a wide range of applications. There are communication satellites which are able to communicate between a number of remote locations very rapidly. There are navigation satellites also known as the GPS that help people determine their locations and navigate on land, sea and air. There are reconnaissance satellites used for military purposes and to spy on other countries [1]. However, the satellites that are of concern in this paper are called Earth observation (EO) satellites which monitor and gather data on the Earth's surface and beneath it [2]. These data help researchers and analysts understand and monitor global development, natural calamities, climate changes, growth of crops etc. With the constant improvements of these satellites, the information gathered is becoming more sensitive and of utmost importance and hence, needs to be protected no matter the cost [2].

One of the biggest concerns about these satellites is that most of these satellites are in Low Earth Orbits (LEO) and travel through a region of space known as the South Atlantic Anomaly which is an area where the Earth's inner Van Allen radiation belt comes closest to the Earth's surface, leading to an increased flux of energetic particles in this region and exposes orbiting satellites to higher-than-usual levels of radiation [3]-[5]. These high-energy charged particles impart their energy into onboard solid-state memory, flipping a bit and causing incorrect computations that may affect spacecraft performance [5]. Such electronic problems may result in mechanical symptoms, such as loss of attitude and pointing control [3], [5]. Hence, it is necessary to address these errors and correct them to ensure reliability and longevity of the system. In this paper, this issue has been addressed by studying few LEO satellites, and using a very reliable EDAC scheme known as the Hamming Code which uses the concept of parity and parity bits to not only detect and correct an error, but to also locate the bit-position of the error [6], [7]. One of the biggest advantages of Hamming Codes is that it makes forward error correction (FEC), which is a system that makes possible the transmittance of quite reliable information in a simplex system, much affordable [8]. The paper mainly discusses the process of generating these Hamming Codes and checking a received codeword for errors. Using the mechanism, Hamming (16, 11, 4) was generated which proved to be the most effective among other schemes and the comparisons are provided later in the paper. The paper also discusses the limitations of Hamming Code and a way to mitigate this problem. By looking into all these areas, the use of Hamming Codes for EDAC for onboard satellite computers has been finally justified.

The main contributions of this paper are as follows:

- A suitable EDAC scheme and a model for use in satellites have been proposed.
- The anomalies recorded by several LEO satellites have been studied.
- The processes of generating Hamming Codes and syndrome to detect and correct errors have been discussed.
- Comparisons among different EDAC schemes have been shown.
- Limitations of the proposed scheme along with a viable solution have been identified.

The remainder of the paper is organised as follows: Section II reviews the background related to Hamming Codes, studies of some LEO satellites and the justification of using Hamming

Rafsan Al Mamun, Rabana Tajrin and Nabiha Noor are students of Computer Science and Engineering, BRAC University, Dhaka 1212, Bangladesh (e-mail: rafsanzara9805@icloud.com, rabanatajrin22@gmail.com, noornabiha1998@gmail.com).

Shafinaz Qader is a student of Computer Science, BRAC University, Dhaka 1212, Bangladesh (e-mail: shafinaz.q.s.22@gmail.com).

Md. Motaharul Islam is a professor, United International University, Dhaka 1212, Bangladesh (e-mail: motaharul@cse.uiu.ac.bd)

Open Science Index, Electronics and Communication Engineering Vol:14, No:9, 2020 publications.waset.org/10011427.pdf

Codes for such satellites. In Section III, the proposed system has been discussed. Section IV provides the mathematical models, an algorithm for implementation and the generation of Hamming (16, 11, 4) which has been regarded as the most effective version. Section V gives comparison among several EDAC schemes and the limitations along with solutions. Finally, Section VI concludes the paper.

II. Background

The EDAC scheme chosen here, i.e. Hamming Code is to detect and correct errors caused by radiation on an onboard satellite computer during a LEO.

A. Hamming Codes

Hamming Code is a set of error-correction codes that can be used to detect and correct bit errors and is named after Richard Hamming of Bell Labs, who found the algorithm of Hamming Code in 1950. In data communications, Hamming Codes are a family of linear block error correcting codes as seen in Fig. 1 [8], [9].

Hamming Code, similar to most other EDAC schemes, makes use of redundant bits [16]. These are extra bits added to data to check their validity in order to correct them [16]. For each integer $r \geq 2$, there is a code with codeword length $n = 2^r - 1$ and data word length $k = 2^r - r - 1$. Due to the use of more than one parity bit, this scheme can locate the position of the error and self-correct it by inverting the bit. A brief overview of the original Hamming Codes can be seen in Table I [10].

Although the original scheme of Table I allows the detection and correction of only single-bit errors at the same time, double-bit errors can also be detected by the addition of one extra parity bit and the result is an extended Hamming Code [11]. The codewords generated in a Hamming Code scheme can be both systematic and non-systematic, although the paper is more concerned with the latter due its better code rate [6]. The two differ by the positions of the parity bits in the codeword as shown in Table II, using an example of 11 bits.

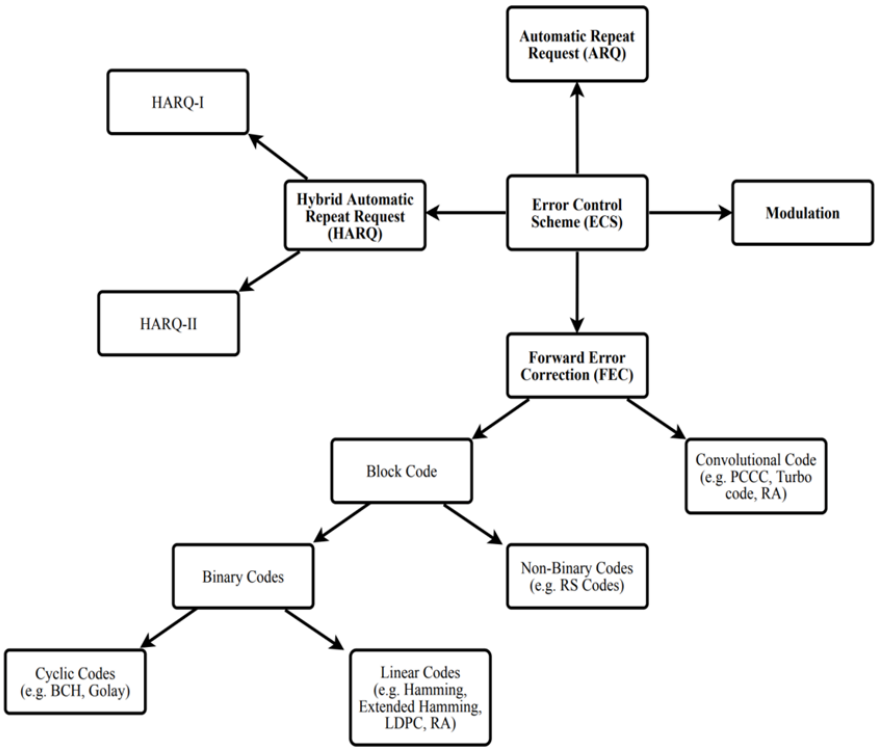

Fig. 1 Classification of Error Control Scheme

TABLE I
CLASSIFICATION OF HAMMING CODES

Named after	Richard W. Hamming
Type	Linear Block Code
Block Length	$n = 2^r - 1$ where $r \geq 2$
Message Length	$k = 2^r - r - 1$
Distance	$d_{min} = 3$

TABLE II
NON-SYSTEMATIC AND SYSTEMATIC CODEWORDS

Systematic	[*k0 k1 k2 k3 k4 k5 k6 k7 r0 r1 r2*]
Non-systematic	[*r0 r1 k0 r2 k1 k2 k3 k4 k5 k6 k7*]

B. Study of Alsat-1

The anomalies recorded by the Alsat-1, the first Algerian micro-satellite launched into LEO [12] have been studied first. These data were observed over a period of 7 years and published in a journal [12]. Alsat-1 is not only a LEO satellite but is also Sun Synchronised Orbit satellite, meaning that Alsat-1 experienced high dosage of radiation. The data recorded by Alsat-1 can be seen from Table III [12].

After studying the data gathered by Alsat-1 and provided in Table III, it is quite evident that single-bit errors are much likely to happen onboard a satellite computer due to effects of

Open Science Index, Electronics and Communication Engineering Vol:14, No:9, 2020 publications.waset.org/10011427.pdf

radiation. From the table, it can also be calculated that there are about 97 single-bit flips on average per day. It can be inferred that most of these errors might have been caused in Alsat-1 during its traversal through the South Atlantic Anomaly, which is again an area where satellites are exposed to very high radiation.

TABLE III
EVENT UPSETS OBSERVED ON ALSAT-I DURING A 7-YEAR PERIOD

System monitored	OBC 386 RAMDISK memory
EDAC code	R-S (256,252)
Memory size	32 Mbytes
Hybrid	SYS84000RKXLI-70 (4M × 8-bit)
Device	Samsung SEC KM684000BLT-5L: 512K × 8-BIT SRAM
Observation period	2510 days November 29, 2002–October 12, 2009
Bit monitored	268435456
No. of single-bit errors	244150 (98.6%)
No. of multiple-bit errors	217 (0.08%)
No. of double-byte errors	2996 (1.21%)
No. of severe errors	230 (0.09%)

C. Study of the CRRES

Next, the anomalies occurring on the Combined Release and Radiation Effects Satellite (CRRES), a NASA satellite launched for investigating fields, plasmas, and energetic particles inside the Earth's magnetosphere, have been studied from its launch on July 25, 1990 to its failure on October 12, 1991. Its findings were also published in a journal [13]. The satellite was launched with a perigee in LEO and apogee near Geosynchronous Equatorial Orbit.

Fig. 2 illustrates the findings of the CRRES [13]. It can be seen that single-error upsets (SEU) occurred mostly when the satellite was travelling in LEO.

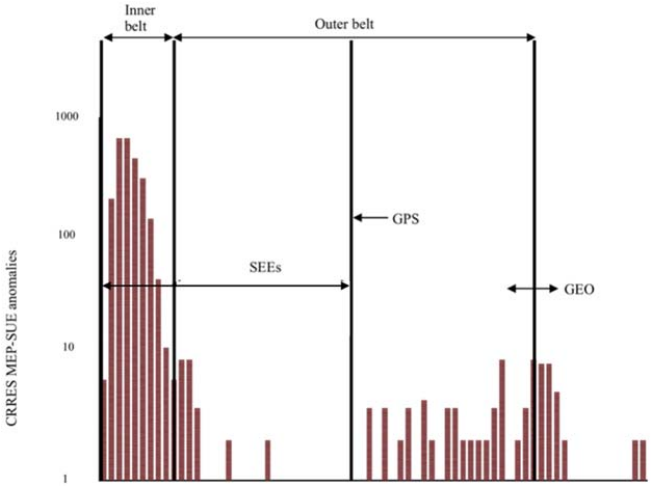

Fig. 2 Anomalies caused by SEU on the CRRES

D. Study of the SOHO

Finally, the anomalies of the Solar and Heliospheric Observatory (SOHO) satellite launched by the European Space Agency reported from April 1996 to August 2001 have been studied from a journal [14]. Events were reported occurring in the solid-state recorder (SSR) of SOHO.

Fig. 3 shows the time plot of the SEUs observed in the SSR with respect to the cosmic ray flux [14]. As it can be seen, although the average upset rate fluctuates around 0.5 SEU/min to 1 SEU/min, there are large peaks going out of the plot which happened during major solar flares. Hence, it is quite evident that the environment in which the satellites reside are most likely to cause erroneous data in the form of single-bit errors.

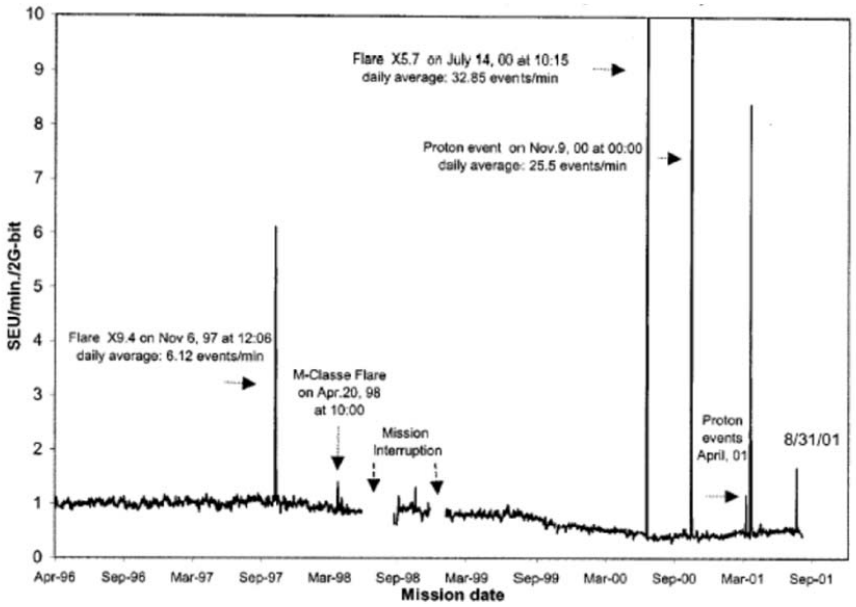

Fig. 3 SOHO SSR upsets during April '96 to August '01; Copyright: [14]

E. Use of Hamming Codes in Satellites

From the studies of the various satellites in LEO, it is very much visible why Hamming Codes are being used as the EDAC scheme for satellites. Hamming Codes, as stated

Open Science Index, Electronics and Communication Engineering Vol:14, No:9, 2020 publications.waset.org/10011427.pdf

earlier, are capable of detecting and correcting single-bit errors on its own and with the addition of an extra parity bit, it can be made to detect double-bit errors as well. Although Hamming Codes are not capable of correcting double-bit flips, it can make sure that the information is discarded and can invoke a request to retransmit the information whenever a double-bit error is detected. Additionally, some other schemes can also be added to correct any burst errors.

Since Hamming Codes meet the needs of the system and are also simple to implement, this scheme has been chosen for detecting and correcting errors onboard a satellite computer.

III. Proposed System

As stated previously, the EDAC scheme used here, i.e. Hamming Codes, ensures that no corrupted information is extracted or stored caused by single-bit errors [8]. The model of this system is illustrated in Fig. 4.

Step1. The satellite captures images and gathers information from the Earth's surface and beyond, and sends them as *k*-bit long inputs to the encoder. It is to be noted that the information at this stage is error free.

Step2. The encoder generates parity bits for the data word with the help of Hamming Code scheme and inserts them at particular positions in the data to form a *n*-bit long codeword and is stored in the onboard computer.

Step3. The data stored in the computers are subject to errors when the satellite passes through regions of high radioactivity as the radioactive particles can pass through the memory cells of the RAM, causing bits to flip.

Step4. The decoder then applies Hamming Code scheme again to calculate the syndrome of the requested data. The syndrome helps the decoder to check and locate any errors present in the codeword. If it finds any single-bit errors, it automatically corrects them. Since an extended version of Hamming Code is being used by adding an extra parity bit, the decoder can detect double-bit flips as well.

Step5. The error-free data are extracted and sent as output. However, if any double-bit flips are detected, the information is discarded and a request is sent to resend the data.

IV. Mathematical Models And Algorithm For Implementation

In order to implement the proposed system, non-systematic Hamming Codes are required. This section aims to highlight the mathematical approach of building the said codewords and an algorithm to be used for implementing the system.

A. Non-Systematic Hamming Codes

The key to the Hamming Code is the use of extra parity bits to allow the identification of errors. In order to turn a data word into a codeword, a special algorithm needs to be followed which is dependent on the type of codeword to be made. Since the use of non-systematic codeword has been proposed for the implementation of the system, from Table II, the following observations about the codeword can be made:

- The bit positions that are powers of two are to be assigned as parity bits. (positions 1, 2, 4 etc.)
- The remaining bit positions are for the data to be encoded. (positions 3, 5, 6, 7, 9, 10 etc.)
- A particular parity bit is responsible for the calculation of its parity using bits in certain positions in the codeword [15]. The choice of these bits follows a specific algorithm that is dependent on the position of the parity bit [15]. For a parity bit in position n, starting from n, the algorithm alternately checks n bits and skips n bits. For example:
  - Position 1: starting from position 1, check 1 bit and skip 1 bit (1, 3, 5, 7, 9, 11 , 13, 15,…)
  - Position 2: starting from position 2, check 2 bits and skip 2 bits (2, 3, 6, 7, 10, 11, 14, 15,…)
  - Position 4: starting from position 4, check 4 bits and skip 4 bits (4 ,5, 6, 7, 12 ,13, 14, 15,…)
  - Position 8: starting from position 8, check 8 bits and skip 8 bits (8, 9, 10, 11, 12, 13, 14, 15,…)
- In order to set a parity, count the total number of ones in the positions checked. If there are odd number of ones, set the parity bit to 1 and 0 otherwise [15].

B. Generator Matrix (G)

This is the matrix used to encode a data word of length *k* to a codeword of length *n*. $G_{(k\times n)}$ is the combination of an identity matrix (I) and a sub matrix (P):

$$G_{k\times n} = [I_{k\times k} \mid P_{k\times r}] \tag{1}$$

C. Parity-Check Matrix (H)

This is the matrix used to decode and correct a codeword. $H_{(r\times n)}$ is the combination of negative transposed matrix of P and of I:

$$H_{r\times n} = [-P^T{}_{k\times r} \mid I_{r\times r}] \tag{2}$$

It is to be noted that the matrices G and H can be made non-systematic from systematic by the help of elementary row operations.

D. Encode and Decode

The matrices G and H made above are used to build the encoder and decoder of a Hamming Code. The encoder would make an *n*-bit codeword from a *k*-bit data word while the decoder would calculate the syndrome required to detect errors in the codeword and extract the error-free data from it. This is done by using the following formulae:

$$codeword_{n\text{–bits}} = \mathrm{mod}_2\,(M_{k\text{–bits}} * G_{k\times n}) \tag{3}$$

$$syndrome_r = \mathrm{mod}_2\,(C_{n\text{–bits}} * H^T{}_{r\times n}) \tag{4}$$

Since an extended Hamming Code is to be used in the system, an additional parity bit is to be added which would be the modular-2 addition of all the codeword bits.

Open Science Index, Electronics and Communication Engineering Vol:14, No:9, 2020 publications.waset.org/10011427.pdf

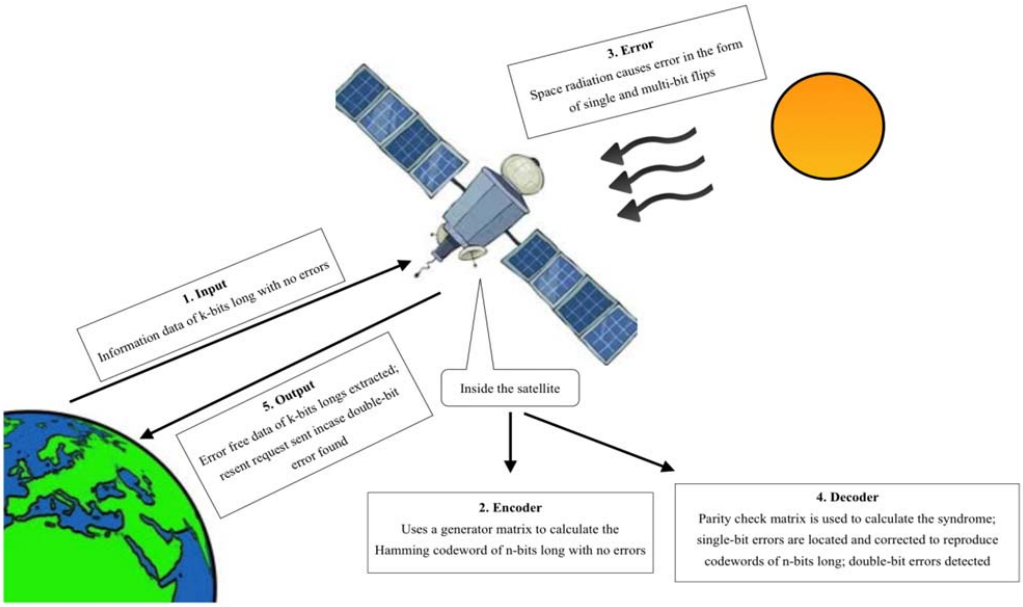

Fig. 4 Overview of the proposed system

INPUT: Message (*msg*)
▷ Define all parameters
$k \leftarrow$ message bits
$r \leftarrow$ parity bits
$n \leftarrow$ codeword bits
$G \leftarrow$ Generator matrix
$H \leftarrow$ Parity-check matrix
Generate the parity bits and the codeword using those bits ▷
Codeword = mod_2 (*msg*⋆*G*)
Introduce errors ▷ flip the bits
Generate syndrome ▷ *Syndrome* = mod_2 (*codeword*⋆H^T)
For $i \leftarrow 1$ to n
do if error found using syndrome
then flip the bit
continue check
if another error found
then discard data
break loop
Extract the error free message bits from codeword and print the message.
Print if any error was found and if found, mention the type, i.e. "Single-bit", "Double-bit"
If no error was found, print "No error".

Algorithm 1: Algorithm to implement Hamming Code scheme on MATLAB

E. Checking for an Error

Upon receiving the codeword, the receiver checks for any errors by verifying each check bit. If the received codeword has no errors, the parity bits are discarded and the data word accepted. If there is any error, the receiver corrects that error to extract the data word that was sent. To do so, the receiver needs to check each parity bit and add the positions that are wrong which would ultimately give the position for the incorrect bit. The receiver can then just invert that bit, discard the parity bits and get the initial data word sent.

F. Algorithm for Implementation

In order to implement Hamming Code, an algorithm has been designed which has been implemented using MATLAB.

For the purpose of this system, the Hamming (16, 11, 4) is preferred since data are stored in double-bytes in satellite storage and also due to its better code rate (see Performance Evaluation). Since it has an extra parity bit, it is the extended version of the Hamming (15, 11) and can detect up to double-bit errors while correcting a single-bit error. Moreover, it uses the nonsystematic codeword making error detection much easier and simpler. A layout of this code is provided in Table IV.

The appropriate generator and parity-check matrices are provided below which have been generated using the formulae mentioned earlier and they will be used for the encoding and decoding purposes:

G_{11x16}

1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1
0	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	1
0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1
1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1

H_{5x16}

1	0	0	0	0	1	1	0	0	1	0	1	1	1	0	1
0	1	1	0	0	1	0	0	1	1	1	1	0	0	0	1
0	0	1	0	1	0	1	1	0	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	1	0	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TABLE IV
LAYOUT OF EXTENDED HAMMING (16, 11, 4)

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Encoded data bits		P1	P2	D1	P4	D2	D3	D4	P8	D5	D6	D7	D8	D9	D10	D11	P16
Encoded data coverage syndrome	P1	x		x		x		x		x		x		x		x	
	P2		x	x			x	x			x	x			x	x	
	P4				x	x	x	x					x	x	x	x	
	P8								x	x	x	x	x	x	x	x	
	P16																x

V. PERFORMANCE EVALUATION

Data being lost or interrupted during the transmission is a common thing. Many error correcting codes have been developed to enable the data to be sent through noisy communication channel without corruption. This increases the throughput of the communication link as now there is no need to re-transmit the corrupted data.

Hamming code is one of the very significant error detecting and correcting codes. As previously mentioned, it works with the help of parity bits which are collection of bits added to the original data to check the validity of the transmitted data. Their popularity lies in their effectiveness when it comes to correcting single-bit errors and detecting double bit errors. The use of Hamming (16, 11, 4) was proposed which gives a codeword of double-byte size, since it is convenient due to the fact that most memory blocks work on a byte standard. Hamming Codes give better results when compared to other similar EDAC schemes.

A. Hamming (16, 11, 4) vs Other Hamming Codes

The Hamming (16, 11, 4) has an improved hamming distance, overhead and code rate than other lower versions of Hamming Codes as seen in Figs. 5 and 6. Since it is an extended version of Hamming Code, it can perform both single-error correction and double-error detection. Due to its improved timing capabilities, it was able to check for errors in an average of 5.669 nanoseconds in 10 different 11-bit data words, illustrated in Fig. 7, although that might vary from computer to computer due to the computer's specifications.

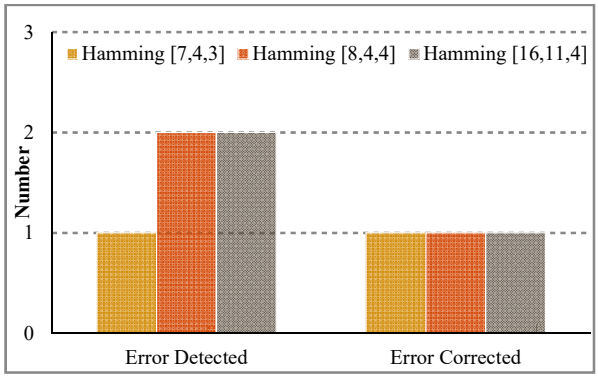

Fig. 5 Comparison between Hamming Codes in terms of errors detected and corrected

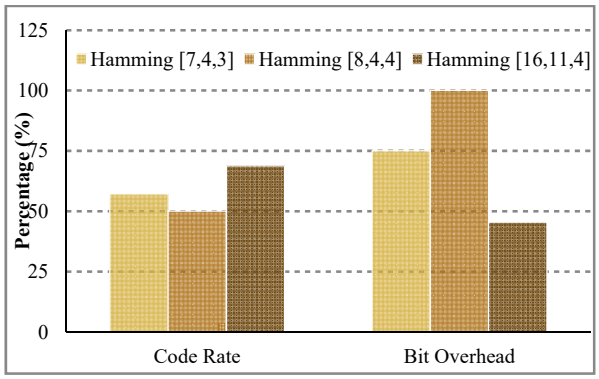

Fig. 6 Comparison between Hamming Codes in terms of code rates and bit overhead

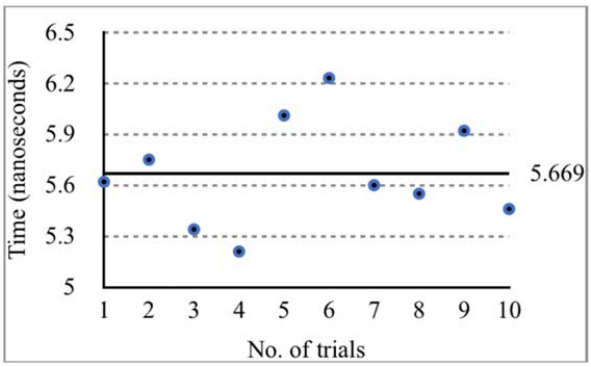

Fig. 7 Timing analysis of Hamming (16, 11, 4)

B. Hamming Code vs CRC

Another effective way of detecting errors is CRC. It possesses many algebraic properties that simplify the encoding and the decoding implementation. While Hamming Code adds binary bits to ensure the validity of the original code, CRC is conceived as the remainder of a polynomial division. Depending on the CRC size, it can detect bursts of errors, which increases the efficiency of the communication. However, there is a major drawback of using CRC. Although it can detect multiple-bit errors, it cannot correct them. Instead, it discards the information and asks the user to resend the data. Hence, for trivial cases such as a single-error, implementing CRC would be time consuming compared to a simple Hamming Code. It is rather better to use CRC together with Hamming Codes only to detect burst errors which is beyond the capabilities of a Hamming Code. Moreover, Hamming Codes can detect a greater percentage of errors per length of code than CRC as the length of codeword increases as can be seen from Fig. 8.

Open Science Index, Electronics and Communication Engineering Vol:14, No:9, 2020 publications.waset.org/10011427.pdf

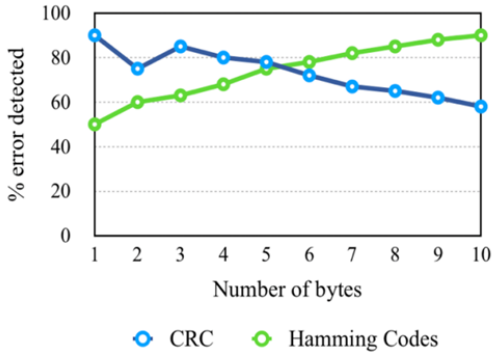

Fig. 8 Error detection performance of CRC and Hamming Code

C. Limitations

As previously discussed, although the extended non-systematic Hamming Codes can detect double-bit errors, they are only capable of correcting single-bit errors. An attempt of correcting multiple-bit errors might cause another correct bit to be changed, resulting in a new corrupted data. Instead, another EDAC scheme such as CRC or Triple Modular Redundancy can be used together with Hamming (16, 11, 4) to detect and maybe correct multiple-bit or burst errors. Another alternative is to discard the data and request for retransmission in case more than one erroneous bit is detected in the codeword.

VI. Conclusion

The field of satellites is evolving and growing at an astonishing pace since it provides a platform to people to push the boundaries of science and technology. With technological development, the memory chip cell architecture is becoming more advanced and the data being gathered by the satellites are becoming more and more valuable and sensitive, thus creating an increasing demand for efficient and advanced EDAC schemes to protect these data. Studying the anomalies registered by Alsat-1, CRRES and SOHO, it could be observed that most of the errors caused in LEO are due to single-bit flips. It could be analysed that errors occurred mostly when the satellites were in the inner radiation belts. Based on this study, the use of non-systematic Hamming Codes was proposed which are capable of correcting single-bit errors. It has been also mentioned that with the addition of an extra parity bit, the extended Hamming Codes could also detect double-bit errors at the same time. Using the formulae mentioned previously, the Hamming (16, 11, 4) was implemented which could theoretically prevent all single-bit errors in a satellite computer. This version of the Hamming Code had a better code rate and overhead, and has also proved to be relatively better in single-error correction and double-error detection compared to other EDAC schemes. Hamming (16, 11, 4) is fast, reliable and simple, and is capable of meeting the problem discussed in this paper.

References

[1] Sun, W., Mba, J. P. S. and Sweeting, P. M. (2001). Micro-minisatellites for affordable EO constellations: RapidEye and DMC. IAA Symposium on Small Satellites for Earth Observation, Berlin, Germany.

[2] Banu, R. and Vladimirova, T. (2006). On-board encryption in earth observation small satellites. 40th Annual 2006 International Carnahan Conference on Security Technology, pp. 203–208, Lexington, KY, USA.

[3] Welser, W., Galvan, D. A., Hemenway, B., Baiocchi, D. (2014). Satellite Anomalies Benefits of a Centralized Anomaly Database and Methods for Securely Sharing Information Among Satellite Operators. Defense Advanced Research Projects Agency (DARPA), USA.

[4] Wren, G. L., and Andrew J. S. (1993). Surface charging of spacecraft in geosynchronous orbit. The Behavior of Systems in the Space Environment: NATO Advanced Study Institute.

[5] Wrenn, G. L., Rodgers, D. J., Ryden, K. A. (2002). A solar cycle of spacecraft anomalies due to internal charging. Annales Geophysicae, Vol. 20, 3/2002, 2002, pp. 953–956.

[6] Elias, P. (1954). Error-Free Coding. Technical Report 285. Massachusetts Institute of Technology, Research Laboratory of Electronics, USA. Retrieved from: https://dspace.mit.edu/bitstream/handle/1721.1/4795/RLE-TR-285-14266170.pdf

[7] Forouzan, B.A. (2007). Error Detection and Correction. Data Communications and Networking. New York, NY: McGraw-Hill.

[8] Hamming, R.W. (1950). Error Detecting and Error Correcting Codes. The Bell System Technical Journal, 29(2): 147-160.

[9] Kadel, R., Islam, N., Ahmed, K. E. U., Halder, S. J. (2018). Opportunities and Challenges for Error Correction Scheme for Wireless Body Area Network—A Survey. Journal of Sensor and Actuator Networks, 8(1):1.

[10] Dr. Humphrys, M. (School of Computing, Dublin City University). Retrieved from: https://www.computing.dcu.ie/~humphrys/Notes/Networks/data.hamming.html

[11] Gill, P., Ruiz, J., Gil-tomás, D., Gracia-morán, J., Itaca, I. and De Valencia, U. P. (2014). Modified hamming codes to enhance short burst error detection in semiconductor memories. Tenth European Dependable Computing Conference, pp. 62–65, Newcastle, UK.

[12] Bentoutou, Y. (2012). A real time EDAC system for applications onboard earth observation small satellites. IEEE Transactions on Aerospace and Electronic Systems, vol. 48, no. 1, pp. 648– 657.

[13] O'Brien, P., Timothy B. G., Joseph E. M., and Richard K. L. (2009). Spacecraft Environmental Anomalies Expert System (SEAES) - IR&D Developments FY06 to FY09, El Segundo, Calif.: Aerospace Corporation, ATR-2009(8073)-3

[14] Harboe-Sørensen, R. et al. (2002). Observation and Analysis of Single Event Effects On-Board the SOHO Satellite. IEEE Trans. Nucl. Sci., vol 49, no. 3, pp. 1345-1350.

[15] Downey, T. (2018). Fundamentals of Computer Systems. University Instructor, Florida International University. Retrieved from: https://users.cs.fiu.edu/~downeyt/cop3402/hamming.html

[16] Rouse, M. (2010). Hamming Code. Writer and Manager, WhatIs.com. Retrieved from: https://whatis.techtarget.com/definition/Hamming-code

Open Science Index, Electronics and Communication Engineering Vol:14, No:9, 2020 publications.waset.org/10011427.pdf