

A Comparative Study of Construction Techniques of t -Error Correction/All Unidirectional Error Detection Codes

¹Thabit Sultan Mohammed and ²Laith A. Al-Jobouri

¹Department of Electrical and Computer Engineering, College of Engineering, Dhofar University, P.O. Box 2509, 211 Salalah, Oman

²School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK

Article history

Received: 14-07-2014

Revised: 06-09-2014

Accepted: 11-09-2014

Corresponding Author:

Thabit Sultan Mohammed
Department of Electrical and
Computer Engineering, College of
Engineering, Dhofar University,
P.O. Box 2509, 211 Salalah, Oman
Email: thabitsm@yahoo.com

Abstract: The theory and application of (t -EC-AUED) codes was presented and among the methods proposed in literature, the most efficient was chosen and a software was written for encoding and decoding of the codes. Comparison and evaluation of the construction techniques were carried out. A t -Error Correction (EC)/All Unidirectional Error Detection (AUED) codes are constructed by appending a single check symbol to a linear t -EC code to achieve the AUED property.

Keywords: Unidirectional Error Coding, Codes, Error Correction Codes, Error-Detection Codes

Introduction

The use of codes for error correction and detection is long established in the design of large digital systems and provides for the occurrence of faults, which must ultimately develop in any system. For a computer system, an important consideration is the balance between error detection and error correction. Error correction seems to be more desirable than error detection; however, the possibility of erroneous correction exists and has to be guarded against.

One of the most important ways of guarding against erroneous correction is by the use of codes, which detect All Unidirectional Errors (AUE). In (Nikolos *et al.*, 1986) the most important mechanisms for unidirectional errors are described. These errors are likely to occur when a fault in VLSI devices such as ROM and RAM memories exists. Other possible reasons for production of these errors are bridging faults in PLA's, stuck at faults, power supply failures, burst in magnetic recording surfaces due to foreign particles or permanent failure of stages in shift register type memory.

Unidirectional errors can occur in a large number compared to the limited number of random errors likely to occur.

A t -EC/AUED code can be constructed by appending a check symbol to each word of linear binary t -EC code and the first construction technique

was proposed by (Pradhan and Stiffler, 1980). After that many methods have been presented. In (Bose *et al.*, 2007), a design technique for a class of t -unidirectional error detection codes over Z_m is presented. A survey of unidirectional error detection codes for memory systems is presented in (Al-Ani and Al-Shayea, 2010). In (Nikolos *et al.*, 1986) two techniques were proposed in which $(t+1)$ check symbols are added. Andrew (1988) has proposed a more powerful technique than those of (Nikolos *et al.*, 1986). An efficient technique was proposed in (Tao *et al.*, 1988), but it is not as efficient as that of (Andrew, 1988).

In the next section of this research, a definition of the theory of codes and some of the terms related to them are introduced. The other section includes the categories of errors in computers and digital systems. The section followed presents an introduction to the theory of t -EC/AUED codes is presented. In the reset sections we can see a description of a number of construction techniques that are found to be the most efficient. A brief idea about the construction of the software package, the results obtained from running the software are tabulated and discussed and the last section presents conclusions.

Error Control Codes

The theory of control coding is concerned with encoding and decoding of data and the mean of

implementing them in hardware and software. Coding theory is also concerned with n-bits words that are of the form given in Equation 1:

$$X = (a_1 a_2 \dots \dots \dots a_n) \quad (1)$$

where, $a_1, a_2 \dots \dots \dots a_n$ are symbol of a set S .

Let q be the number of symbol that are contained in S . There are q^n possible words of length n , of total of q symbols. A code C is defined to be a subset of all possible code words.

If the set of symbols are elements of a finite field with $q = 2$, then binary codes will be constructed and these codes are the most important, hence they will receive the greatest emphasis in this work.

There are two main types of codes in common use today block codes in which n is fixed and conventional codes in which n is not fixed.

Following are some definitions, which are related to codes.

Definition 1

The Hamming distance between two n -bit words \underline{v} and \underline{w} denoted $d(\underline{v}, \underline{w})$ is defined as the number of places where they differ.

Definition 2

The weight $w(\underline{v})$ of a word \underline{v} is the number of non-zero digits in it.

Linearity forms a common feature for a large number of codes which are referred to as linear codes (Lin, 2004). An (n, k) binary block code C is called linear if its set of 2^k n -tuples code words are a subset of all n -tuples over $GF(2)$ and can be expressed as a linear combination of a set of k basis vectors.

Error Detection and Error Correction

Computers and digital systems are exposed to faults, which are many and varied. The faults lead to errors in data which are usually categorized as follows:

- Symmetric (random) Errors: When both $(1 \rightarrow 0)$ and $(0 \rightarrow 1)$ errors are likely to occur
- Asymmetric Errors: When either $(1 \rightarrow 0)$ or $(0 \rightarrow 1)$ errors occur with different probabilities
- Unidirectional Errors: When both types $(1 \rightarrow 0)$ and $(0 \rightarrow 1)$ errors can occur, but in a particular word all errors are of one type

The statistics of errors are strongly dependent on the way the data is organized and when the

organization of a RAM memory, for example, consists of several bits per card, a defect in a memory card can affect several bits in the same word. Similarly a defect on a tape due to handling or the presence of foreign particles usually effects one track; however, it will cause possibly a large number of errors on this track.

Early errors correcting codes, which were designed under the assumption that errors are random, may not be entirely appropriate for errors which may occur in RAM's or magnetic tapes. This fact has led to the design of a new class of codes, which are capable of detecting unidirectional errors. This category of errors is the most probable to occur in digital systems and devices. Among the all-unidirectional error detection codes are the *m-out-of-n* codes (Pradhan and Stiffler, 1980) and the Berger codes (Pradhan, 1986).

Error Correction/All Unidirectional Error Detection

In spite of their ability of detecting all unidirectional errors, the *m-out-of-n* and the Berger codes, mentioned earlier, are limited in application because they are not able to correct errors; and are incompatible with parity check codes which are the most used codes in computers. Consequently, recent researches has been directed to develop codes that correct all patterns of up to (t) random errors and detect either all or d ($d > t$) unidirectional errors.

In order to study these codes, the following definitions will be made:

Let X and Y be n -tuples over $GF(2)$ then:

- $D(X, Y)$: refers to the Hamming distance between X and Y
- $N(X, Y)$: denotes the number of $(1 \rightarrow 0)$ crossovers from X to Y

The Hamming distance can be expressed in terms of crossovers as per Equation 2:

$$D(X, Y) = N(X, Y) + N(Y, X) \quad (2)$$

Example

If $X = (101110)$ and $Y = (010110)$ then:

$$N(X, Y) = 2, N(Y, X) = 1 \text{ and } D(X, Y) = 3$$

The following theorems describe the error correction/detection properties of binary codes.

Theorem 1

A code C is capable of detecting up to (t) random errors if the condition of Equation 3 satisfies:

$$D(X, Y) \geq t + 1 \text{ for all } X \text{ and } Y \in C \tag{3}$$

Theorem 2

A code C is capable of detecting up to (t) random errors and all unidirectional errors if the condition of Equation 4 satisfies:

$$N(X, Y) \geq 1 \text{ and } D(X, Y) \geq t + 1 \text{ for all } X \text{ and } Y \in C \text{ and } X \neq Y \tag{4}$$

Theorem 3

A code C is capable of correcting up to (t) random errors and detects all unidirectional errors if the condition of Equation 5 satisfies:

$$\text{both } N(X, Y) \geq t + 1 \text{ and } N(Y, X) \geq t + 1 \text{ for all } X, Y \in C \text{ and } X \neq Y \tag{5}$$

Construction Techniques of t -EC/AUED Codes

In all proposed t -EC/AUED design techniques, attempts were made to construct as efficient code as possible. It's however, that most of the proposed techniques are difficult to implement. Such codes are either having no encoding/decoding algorithms or the algorithms given are inefficient.

The construction techniques adopted in this work are found promising and their implementation is not difficult. Their encoding algorithms, which produce systematic t -EC/AUED codes are reviewed and implemented in a software package. The three adopted techniques are featured in that a universal decoding algorithm can be applied on them (Mohammed, 1988).

Two of the algorithms are given in (Nikolos, 1986). In these two techniques, $(t+1)$ check symbols are applied to a t -EC systematic code with Hamming distance $(d_H = 2t + 1)$. The third technique is given in (Andrew, 1988).

For ease of reference, the construction techniques given in (Nikolos, 1986) are referred to as TECH1 and TECH2 in the next sections, while the technique presented in (Andrew, 1988) is referred to as TECH3.

Description of Techniques TECH1 and TECH2

For these two techniques the proposed codes have the form shown in Equation 6:

$$X_i C H_1 C H_2 \dots C H_{t+1} \tag{6}$$

where, X_i is n -tuple from $GF(2)$.

CH_1, \dots, CH_{t+1} are the values of check words corresponding to X_i . They will be calculated using TECH1 or TECH2.

Consider n -tuples words over $GF(2)$ with weights of $0, 1, \dots, n$ and that class i contains all n -tuples of weight i . The classes can be arranged in an ascending order according to their weight.

In order to calculate $CH_1 \dots CH_{t+1}$, partition $(t+1)$ times the pre-arranged classes into check words where the number and size of these check words will be decided by using either TECH1 or TECH2.

TECH1

Step 1: Set $m = 1$

Step 2: Partition the ordered classes $0, 1, 2, \dots, n$ into check words of size $d_H - 2(t - m + 1)$, so that each check word contains consecutive classes with each class appearing in only one check word. Begin the partition from class n . In this way there are $\lceil (n+1) / (d_H - 2(t - m + 1)) \rceil$ check words

Step 3: Give numbers to the check words; start with number 0 to the check word contains class n

Step 4: If $m = t + 1$ end. Else set $m = m + 1$ and repeat from step 2

TECH2

Step 1: Set $m = 1$

Step 2: If $d_H - 2L = 2^u - L$, where $L = t - m + 1$ and $u > 1$, set $R = 2^u$ else $R = d_H - 2L$

Step 3: Partition the ordered classes $0, 1, \dots, n$ into check words of size R , so that each class appearing in only one check word. Start partitioning from class n . In this way there are $\lceil (n+1) / R \rceil$ check words

Step 4: Give numbers to check words, start with number 0 to the check word contains class n

Step 5: If $m = t + 1$ end. Else set $m = m + 1$ repeat from step 2.

Description of Technique TECH3

By this technique a systematic t -EC/AUED code can be constructed by concatenation of any (n, k) t -EC code and a single check symbol.

The necessary conditions for the check symbols is given by Theorem 4 as follows:

Theorem 4

Let $S(s, t) = S_0, S_1, \dots, S_m$ be an alphabet of $m + 1$ symbols which satisfies:

$$N(S_i, S_j) \geq \min(j-i, t+1) \quad (7)$$

for all $S_i, S_j \in CS(s, t)$ with $j > i$

where, $S(s, t)$ is an alphabet of binary s -tuples.

The number of check symbols $S(s, t)$ is related to the i -EC code C of minimum distance $2t + 2$ of even weight such that shown in Equation 8:

$$\max w(X) < 2m \text{ with } X \in C \quad (8)$$

Check Symbol Construction

The construction of the alphabet $S(s, t)$ will be started with $S_0 = (11 \dots 1)$ in the first symbol and cyclic shifts of a block of P adjacent 1's and $(s-p)$ adjacent 0's, which is arbitrary when $p = s$.

Check symbols are constructed by generating S_i successively according to the following algorithm:

```

begin
  i = 0
  Si = (11...1)
  repeat
    R = CSR(Si)
    if TEST(R) = false then
      R = LBT(R)
    end if
    i = i+1
    Si = R
  until (W(Si) = 0)
  Si = (00 .... 0)
end.
    
```

Three functions used in the algorithm are defined as follows:

- CSR is the Cyclic Shift Right (e.g., CSR (111001) = (111100));
- LBT Leading Bit Transition (e.g., LBT (111001) = (110001); and
- TEST (R) = true if $(S_0, S_1, \dots, S_{i+1})$ satisfies the condition of theorem 4, when $S_{i+1} = R$
- TEST (R) = false otherwise

The cardinality (number of check symbols) of the alphabet constructed by TECH3 is given by Equation 9:

$$|S(s, t)| = (s - t - 1) \lfloor s / (t + 1) \rfloor + t + 2 \quad (9)$$

The Software Program

The software package is designed to implement the techniques defined in the previous section (i.e., to

calculate the check symbols). Calculated check symbols are then appended to a t -EC code in order to make a t -EC/ AUED code.

The program interacts either with the user or with a stored data file in order to get the information and check bits of an (n, k) binary code. The generator polynomial of the code and its number of correctable errors (t), are also entered.

The check polynomial is then calculated by the program and the generator and check matrices are produced. One out of three techniques (i.e., TECH1, TECH2 or TECH3) is to be chosen where check symbols are calculated.

The encoding procedure of a t -EC code is executed in the program, where message words of k -information bits are produced and multiplied by the generator matrix. Finally, a t -EC/AUED code is constructed by appending the appropriate check symbol.

Results

The three construction techniques described in the previous section are illustrated in the following case study:

Case Study

Consider the (15,7) systematic 2- EC code, whose minimum Hamming distance $d_H = 2t + 1 = 5$.

By applying the techniques TECH1, TECH2 and TECH3 check symbols will be calculated and hence a 2-EC/AUED code can be constructed.

TECH1

The number of classes is equal to the length of the code ($n = 15$) and the classes will be arranged in an ascending order.

Classes will be divided ($t + 1 = 3$) times into check words. The division is shown in Table 1.

The software program works out the steps of TECH1 in the following order:

Step 1: $m = 1$

Step 2: the classes will be divided into check words of size

$$R = d_H - 2(t - m + 1) = 5 - 2(2 - 1 + 1) = 1$$

So that each check word contains consecutive classes, with each class appears in only one check word.

The division start from class n and the number of check words is calculated as follows:

$$GR = \lceil (n + 1) / R \rceil = \lceil (15 + 1) / 1 \rceil = 16$$

Step 3: The check words are numbered and number 0 is given to the check words contained in class n
 Step 4: Step 2 and 3 are repeated for $m = 2$ and $m = 3$

The calculated check symbols are shown in Table 1. Note that the length of the check symbol is 9 bits.

TECH2

This technique is quite similar to that of TECH1. The only notable difference is that when the size of a check word (R) calculated by the formula:

$$R = d_H - 2(tm + 1) \text{ is found to be equal to } 2^u - 1 \text{ (} u \text{ is an integer), then } R \text{ is set equal to } 2^u$$

This obviously means that the check word will be increased and a code with less number of bits may be generated.

TECH2 is not always superior to TECH1 but becomes so as the value of t increases.

The software program executes the steps of TECH2 as follows:

Step 1: $m = 1$
 Step 2: The classes are divided into check words of size:

$$R = d_H - 2(t - m + 1) = 5 - 2(2 - 1 + 1) = 5 - 2(2 - 1 + 1) = 1$$

The number of check words is: $GR = \lceil (n+1)/R \rceil = 16$

Step 3: The check words are numbered and number 0 is given to the check word contained in class n
 Step 4: Steps 2 and 3 are repeated for $m = 2$ and $m = 3$

Note that when $m = 2$, then $R = 5 - 2(2 - 2 + 1) = 3 = 2^2 - 1$.

In this case, R is set equal 4 instead of 3 classes which lead to a redundancy of 1 bit lower than the code derived from TECH1. This actually implies when the parity check code has a hamming distance of exactly $(2t+1)$. Table 2 shows the calculated check symbols using TECH2. In this table, the length of the check symbols is 8 bits.

TECH3

A systematic 2-EC/ AUED code can be constructed if a single check symbol is appended to each code word of a 2- EC code with even weight.

Consider the (15,7) code with an overall parity check bit added to its code words to make it of distance $2t+2$ and length $n = 16$.

The number of bits in each symbol is governed by the relation Equation 10:

$$|S(s, t)| \geq n/2 + 1 \tag{10}$$

where, $S \lfloor (s, t) \rfloor$ is the number of check symbols in the alphabet. For the (16, 7) code:

$$|S(s, t)| \geq 16/2 + 1 = 9$$

Therefore the length of the check symbol will be determined such that:

$$(s - t - 1) \lfloor s / (t + 1) \rfloor + t + 2 \geq 9$$

Table 1. TECH1

Classes W(x)	Check words			Corresponding check symbols
	CH1	CH2	CH3	
0	15	5	3	111110111
1	14	4	2	111010010
2	13	4	2	110110010
3	12	4	2	110010010
4	11	3	2	101101110
5	10	3	2	101001110
6	9	3	1	100101101
7	8	2	1	100001001
8	7	2	1	011101001
9	6	2	1	011001001
10	5	1	1	010100101
11	4	1	0	010000100
12	3	1	0	001100100
13	2	0	0	001000000
14	1	0	0	000100000
15	0	0	0	000000000

Table 2. Check symbols construction-using TECH2

Classes W(x)	Check words			Corresponding check symbols
	CH1	CH2	CH3	
0	15	3	3	11111111
1	14	3	2	11101110
2	13	3	2	11011110
3	12	3	2	11001110
4	11	2	2	10111010
5	10	2	2	10101010
6	9	2	1	10011001
7	8	2	1	10001001
8	7	1	1	01110101
9	6	1	1	01100101
10	5	1	1	01010101
11	4	1	0	01000100
12	3	0	0	00110000
13	2	0	0	00100000
14	1	0	0	00010000
15	0	0	0	00000000

Table 3. Check symbols S (6, 2) generated using TECH3

S (s, t)	Check symbol
0	11111 <u>1</u>
1	0 <u>1</u> 1111
2	00 <u>1</u> 111
3	000 <u>1</u> 11
4	1000 <u>1</u> 1
5	11000 <u>1</u>
6	11 <u>1</u> 000
7	01 <u>1</u> 000
8	001000 <u>0</u>
9	000000 <u>0</u>

The smallest value of s which can satisfy this relation is 6. Hence, the alphabet S (6, 2) will be generated, as shown in Table 3. Leading bits which are transformed to 0 are underlined. The check symbol has 6 bits.

The t-EC/AUED code C formed by the concatenation of code C and S(s, t), is defined by Equation 11:

$$C = \{[X S_i] : W(X) = 2I, X \in C, S_i \in S(s, t)\} \quad (11)$$

From this case study it can be noticed that, the code constructed using TECH3 is shorter by 1 bit than the code constructed in TECH2 and by 2 bits than that of TECH1. Therefore a more efficient code might be generated when using the construction technique TECH3.

Discussion

The results obtained from the several software runs are tabulated such that the three construction

techniques explained before and the technique given in (Tao *et al.*, 1988), which we will refer to as TECH4, are compared together and discussed.

TECH3 will be used as a reference for this comparison. Tables 4 to 6 show the length of (2, 3 and t-EC/AUED) codes, that are constructed using the four techniques, respectively. In these tables:

N = denotes the length of the code which is used to construct the t-EC/AUED

K = the number of information bits

N = denotes the length of the t-EC/AUED

The total number of check bits is (N²-n). DC1- denotes the difference in check bits in favor of TECH3 when compared to the best code constructed using TECH1 or TECH2:

DC 2 = denotes the difference of check bits in favor of TECH3 when compared to the codes constructed using TECH4

ST = is the result of the following expression Equation 12:

$$ST = |S(s,t)| - (n+1)/2 - 1 \quad (12)$$

Which refers to the number of unused symbols of an alphabet S constructed using TECH3. It also gives an indication to how much the code produced by this technique is efficient.

Table 4. 2-EC/AUED codes

n	k	N'						ST
		TECH1	TECH2	TECH3	TECH4	DC1	DC2	
13	7	22	21	20	21	+1	+1	2
14	8	23	22	21	23	+1	+2	2
17	9	27	27	24	26	+3	+2	0
19	10	29	29	27	28	+2	+1	1
20	11	31	31	28	29	+3	+1	1
21	11	32	32	29	-	+3	-	0
21	12	32	32	29	30	+3	+1	0
26	16	38	37	35	38	+2	+3	0
31	22	43	42	41	43	+1	+2	5
33	22	46	46	43	-	+3	-	4
36	26	49	49	46	48	+3	+2	3
42	32	56	56	52	54	+4	+2	0
50	38	65	64	62	65	+2	+4	2
57	38	72	71	70	-	+1	-	10
63	51	78	77	76	78	+1	+2	7
64	52	80	80	77	79	+3	+3	7
65	53	81	81	78	80	+2	+2	6
76	64	92	92	89	91	+1	+2	1
127	113	145	144	144	145	0	+1	4
144	112	163	163	162	-	+1	-	1

Table 5. 3-EC/AUED codes

n	k	N'						ST
		TECH1	TECH2	TECH3	TECH4	DC1	DC2	
19	8	31	31	28	30	+3	+3	2
20	9	33	33	29	32	+4	+3	2
21	9	34	34	30	-	+4	-	1
22	11	35	35	31	34	+4	+3	1
23	11	36	36	32	-	+4	-	0
23	12	36	36	32	35	+4	+3	0
30	16	45	43	41	42	+2	+1	1
31	16	46	44	42	-	+2	-	0
39	23	55	55	52	55	+3	+3	8
39	32	55	55	52	56	+3	+3	8
43	27	60	60	56	59	+4	+3	6
45	23	62	62	58	-	+4	-	5
51	27	69	68	64	-	+4	+3	2
63	46	82	80	78	79	+2	+1	2
85	64	106	106	102	105	+4	+3	9
127	106	150	148	147	147	+1	0	0
255	231	282	280	281	-	-1	-	2

Table 6. t-EC/AUED codes

n	k	t	N'				ST
			TECH1	TECH2	TECH3	DC1	
63	39	4	85	83	79	+4	3
63	36	5	88	86	82	+4	10
127	99	4	154	152	148	+4	1
127	92	5	158	156	152	+4	14
127	85	6	162	160	154	+4	0
255	223	4	287	285	286	-1	27
255	215	5	292	290	287	+3	3
255	207	6	297	295	291	+4	19
255	199	7	302	299	294	+5	0

Conclusion

For computers and digital systems, one of the ways of guarding against erroneous correction is by the use of codes which detect all unidirectional errors. In order to investigate and compare the several methods and techniques proposed for constructing t -EC/AUED codes, software was developed, by means of which the three most efficient techniques are investigated.

Using the outputs obtained from the software runs, a comparison is made, where the results are tabulated for the most efficient techniques. From the tabulated results it was noticed that, in many cases, TECH3 is the most efficient. In few cases, however, TECH3 fails to be superior to other techniques. The value of ST represents an indication to the efficiency of this technique. When the value of ST is low, efficient codes are constructed. In contrast, higher value of ST leads to less efficient codes.

Another property that favors the codes constructed by TECH3 is the fact that each symbol in the alphabet S is a cyclic shift of adjacent 1's. This property can be invested for the hardware implementation, where the general scheme of the encoder circuit will be a series of shift registers with feedback.

Funding Information

The authors have no support or funding to report.

Author's Contributions

All authors equally contributed in this work.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

- Al-Ani, M. and Q. Al-Shayea, 2010. Unidirectional error correcting codes for memory systems: A comparative study. *Int. J. Comput. Sci.*, 7: 54-61.
- Andrew, R., 1988. Construction of t -EC/AUED codes. *Electron. Lett. Sep.*, 1257-1258. DOI: 10.1049/el:19890635
- Bose, B., S. Elmougy and L.G. Tallini, 2007. Systematic t -unidirectional error-detecting codes over Zm. *IEEE Trans. Comput.*, 56: 564-568. DOI: 10.1109/TC.2007.1014
- Lin, S., 2004. *Error Control Coding: Fundamentals and Applications*. 2nd Edn., Prentice-Hall, ISBN-10: 013042672, pp: 1260.
- Mohammed, T.S., 1988. Application of t -EC/AUED codes on digital computers. M.Sc Thesis, Cranfield Institute of Technology, UK.
- Nikolos, D., N. Gaitanis and G. Philokyrou, 1986. Systematic t -error correcting/all unidirectional error detecting codes. *IEEE Trans. Comput.*, 35: 394-402. DOI: 10.1109/TC.1986.1676782
- Pradhan, D.K. and J.J. Stiffler, 1980. Error correcting codes and self checking circuits. *IEEE Comput.*, 13: 27-37. DOI: 10.1109/MC.1980.1653527
- Pradhan, D.K., 1986. *Fault-Tolerant Computing: Theory and Techniques*. 1st Edn., Prentice-Hall, ISBN-10: 013308230X, pp: 736.
- Tao, D.L., C.R. Hartmann and P.K. Lala, 1988. An efficient class of unidirectional error detecting/correcting codes. *IEEE Trans. Comput.*, 37: 879-882. DOI: 10.1109/12.2237