applied sciences

MDPI

Article

Two Dimensional Parity Check with Variable Length Error Detection Code for the Non-Volatile Memory of Smart Data

Cihun-Siyong Alex Gong [1,2,3,*,†], Yung-Chang Chang [4,†], Li-Ren Huang [4], Chih-Jen Yang [4], Kung-Ming Ji [4], Kuen-Long Lu [4] and Jian-Chiun Liou [5,*]

1 Department of Electrical Engineering, College of Engineering, Chang Gung University, Taoyuan 33302, Taiwan
2 Portable Energy System Group, Green Technology Research Center, College of Engineering, Chang Gung University, Taoyuan 33302, Taiwan
3 Department of Ophthalmology, Chang Gung Memorial Hospital, Linkou Branch, Taoyuan 33305, Taiwan
4 Information and Communications Research Laboratories, Industrial Technology Research Institute, Hsinchu 31040, Taiwan; ycchangs@itri.org.tw (Y.-C.C.); lrhuang@itri.org.tw (L.-R.H.); jeff.yang1123@gmail.com (C.-J.Y.); digo@itri.org.tw (K.-M.J.); sone@itri.org.tw (K.-L.L.)
5 School of Biomedical Engineering, College of Biomedical Engineering, Taipei Medical University, Taipei 11031, Taiwan
* Correspondence: alex.mlead@gmail.com (C.-S.A.G.); jcliou@tmu.edu.tw (J.-C.L.)
† These authors contributed equally to this work.

Received: 18 June 2018; Accepted: 21 July 2018; Published: 24 July 2018

Abstract: This paper proposes a novel technology of memory protection for the Non-Volatile Memory (NVM), applied to smart sensors and smart data. Based on the asymmetry of failure rate between the statuses of bit-0 and bit-1 in the non-volatile memory, as a result of the pollution of the radiation of cosmic ray, a two-dimensional parity with variable length error detection code (2D-VLEDC) for memory protection is proposed. 2D-VLEDC has the feature of variable length of redundant bits varied with content of data word in the NVM. The experimental results show that the same error detection quality could be achieved with a 30% redundancy improvement by applying the proposed 2D-VLEDC. The proposed design is particularly suitable for the use of safety-related fields, such as the automotive electronics and industrial non-volatile memories involved in the industrial automation.

Keywords: memory; protection; ECC; parity; non-volatile; variable length; data integrity; smart sensor

1. Introduction

It has been well known that non-volatile memory (NVM) plays an important role in various systems with the embedded or external structures. For smart data and smart sensor applications, data integrity is of primary importance. From automotive to aerospace fields, research continuously discusses the topic of non-volatile memory's failure models. Experimental results show that the erased and programed cells have different threshold voltage (V_{TH}) shifts [1]. For the programed cells, V_{TH} decreases when the floating gate is hit by an ion, resulting in a reduction of the floating gate stored negative charge. On the contrary, in the case of erased cells, no threshold variation is observed. Figure 1 shows one of the results in [1]. Read only memory (ROM) is a semiconductor memory. Its characteristics is its contents cannot be changed or erased once they have been stored in it. Moreover, its contents will not disappear in spite of power-off. It can be either Electrically-Erasable Programmable Read-Only Memory (EEPROM) or NOR FLASH in implementation. In this study, any floating-gate

ROMs can match the characteristics in Figure 1. Two ROM devices can be chosen for verification—the AT24CS of Atmel, and the S25FL128S of CYPRESS. They are popular and easily available.

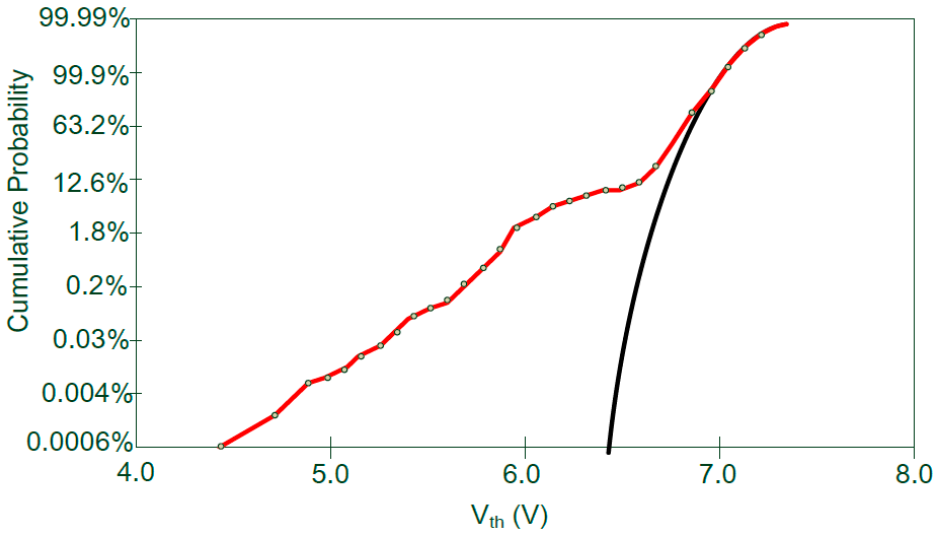

Figure 1. Cumulative distributions of threshold voltages for exposed cells [1].

According to Reference [2], Figure 2 specifies how reading error is caused by the shifted V_{TH} (ΔV_{TH}). When the cell's V_{TH} is lower than read voltage (V_{read}), read error happens because status "0" is read as "1". Status "1" does not have this side-effect, because the ΔV_{TH} of status "1" is almost zero according to the conclusion in Reference [1]. Obviously, the error rates caused by radiation are different between erased cells (status "1"), and programmed cells (status "0"). In Reference [3], a dynamic read reference voltage is used in reading data from non-volatile memory cells. The read reference voltage is calibrated. An Error Checking and Correction (ECC) algorithm is performed to identify whether errors exist in the data for read using the initial read reference voltage level. If errors in the data are identified when read, a pre-determined value is retrieved to adjust the initial read reference voltage level to a new read reference voltage level.

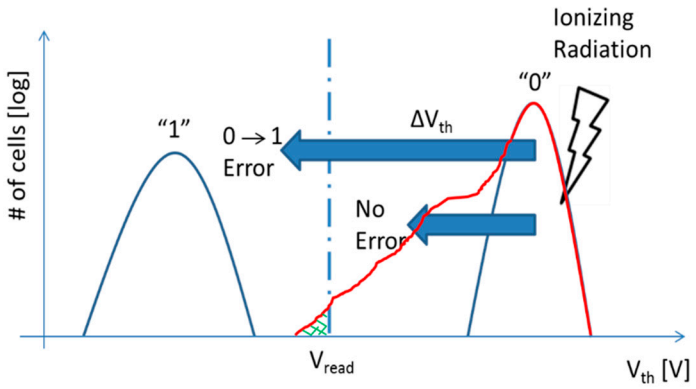

Figure 2. The situation of reading error due to shifted voltage of V_{TH} [2].

In this paper, a novel technology called "Variable Length Error Detection Code" (VLEDC) for the Non-Volatile Memory (NVM) protection is proposed. Although there have been some methods about variable length coding of memory protection [4,5], they were developed without considering the error rate difference between status "1" and status "0" in the memory content. Combined with two-dimensional parity check architecture, the proposed 2D-VLEDC can achieve a high error coverage with significant cost reduction. Compared with the traditional ECC techniques, selectivity can be made

in our design to allow a user to choose either ECC or parity check according to the contents stored in ROM, thereby decreasing the required redundant bits significantly.

This paper was organized as follows: In Section 2, how the 2D-VLEDC works and is applied to NVM is described. Section 3 discusses the experiments on performance comparison of other methods of memory protection with our 2D-VLEDC. Concluding remarks are given in Section 4.

2. Proposed Method

Before discussing 2D-VLEDC, we first introduce a two-dimensional parity (2D-parity) check. Figure 3 describes a ROM whose physical size is 20 × 32 (address 0~19, each address has 32 bits) bits as an example. The addresses from 0 to 16 contain the program codes that Central Processing Unit (CPU) runs with. It is called as data area. The C1 column is *Y*-axis 1-bit parity check of data word at relative address. For example, green square (address 0) in the Figure 3 specifies an even parity coding scheme. In the address 0, only bit (0, 30) from bit 0 to 31 equals to 1, so bit (0, C1) equals to 1. In the same way, address 17 is *X*-axis 1-bit even parity bit for column bits of the program codes. As shown as blue square in the Figure 3, there are totally 15 ones in the column 1 from address 0 to address 16, so the (17, 1) equals to 1 according to even parity rule.

To apply the 2D-parity in a typical memory, the *Y*-axis parity bits (address 18) are stored in the address behind the program codes (address 0 to 16), and *X*-axis parity (address 17). As shown in Figure 3, the *Y*-axis (column C1) bits are stored at the address 18 as presented in the red square. The remaining bits in the address 18 are filled with 0. The storage of *X*-axis and *Y*-axis is called as signature area. In addition, as shown in address 19, the same 2D-parity can be applied to the signature area again to implement self-checking-like scheme to improve the fault tolerant quality.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	C1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
2	1	1	1	0	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	0	1	0	1	1	0	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1
4	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1
8	1	1	1	0	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	0	1	0	1	1	0	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0
13	1	1	1	0	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	0	1	1	1	1	0	0
14	0	1	0	0	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0
15	1	0	1	0	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	1	1	0	1	0	1	0	1	0	1	1	1	1
16	1	1	1	0	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	0	1	0	1	1	0	1
17	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	0	0	0	1	1
18	1	0	1	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
19	0	1	0	1	0	1	0	1	1	1	1	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	0	1	

Figure 3. 2D-parity check applied to non-volatile memory (NVM).

The 2D-parity could detect and correct single bit error in both data area and signature area. Any 2-bit error could be detected. However, the locating of two error bits is not always successful. For example, if there are two error bits in one data word, the *X*-axis shows no change of parity bit even two of bit changes in *Y*-axis parity are found—that is to say, this 2-bit error can be identified but the location of error bit cannot be found, resulting in failure to recover them. As a result, the error handling capability is named as single error correction and double error detection (SECDED).

To compare the undetected error rate of 2D-parity with ECC, a fault injection simulation was applied, as shown in Table 1. In this simulation, 1 to 4 errors were injected and all possible error location combinations in an 8 × 32 size of ROM were evaluated. In other words, for 4-errors fault injection simulation, there are C (8 × 32, 4) operations including injecting fault, decoding, and comparison in the simulation, where C means "combination". Table 1 shows the results of fault injection simulation with 2D-parity and ECC. Undetected error rate equals to (undetected error counts)/C (8 × 32, 4) × 100%. Table 1 show that 2D-parity and ECC can detect all errors from 1 to 3 errors' fault injection. In the 4 errors' fault injection simulation, ECC has effectiveness better (twice) than 2D-parity. The cost overhead of 2D-parity equals to ((8 + 32)/(8 × 32)) × 100% = 15.625%. The cost of ECC equals to ((7 × 8)/(8 × 32)) × 100% = 21.875%. 2D-parity overhead/ECC overhead = 71.43%.

Table 1. Fault injection simulation of ROM size 8 × 32.

No. of Injected Error	Undetected Error of Error Checking and Correction (ECC)	Undetected Error of 2D Parity
1	0	0
2	0	0
3	0	0
4	0.003%	0.007%

One possible way to improve the error detection rate of 2D parity is to include ECC on data word to replace the *X*-axis parity. However, it induces significant increase in redundancy. Here an idea of variable length error detection code is proposed. The probability of read error between 1-to-0 and 0-to-1 is different after NVM in the radiation. It was taken into account to select appropriate *X*-axis redundancy. Figure 4 shows the concept of our VLEDC, the content of data word decides the method of encoding with respective redundant bits. As a result, the length of redundant bits is variable for each data word in NVM.

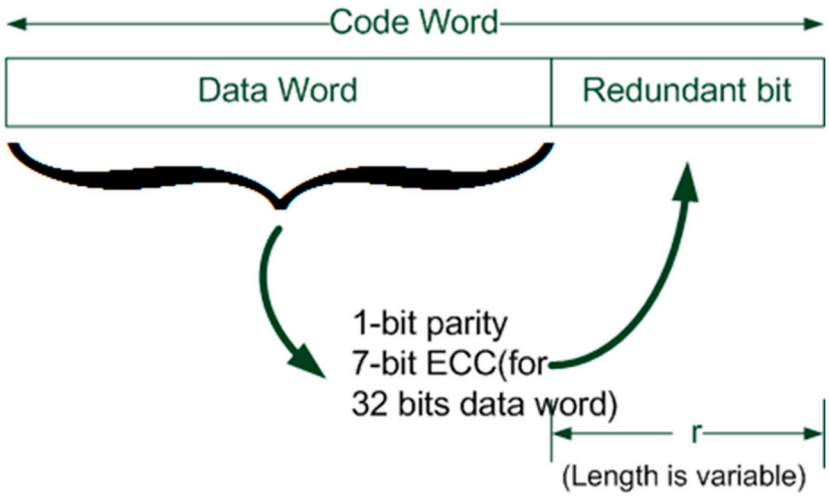

Figure 4. Length of redundant bits depends on the content of data word.

Figure 5 shows the architecture of the proposed 2D-VLEDC, including: Filter, encoder, boot checker and NVM modules. The encoder includes several known memory protection technologies, like 1-bit parity and 7-bits ECC. Before encoding data word, filter module generates encoder-selecting signals by analyzing the zero counts in the data word. A few filter rules could be applied for the filter, such as the number of zero count, the consecutive zero bits, and so on. These threshold values of rule, such as the number of "0" bits, could be adjusted according to the demands of reliability in the system.

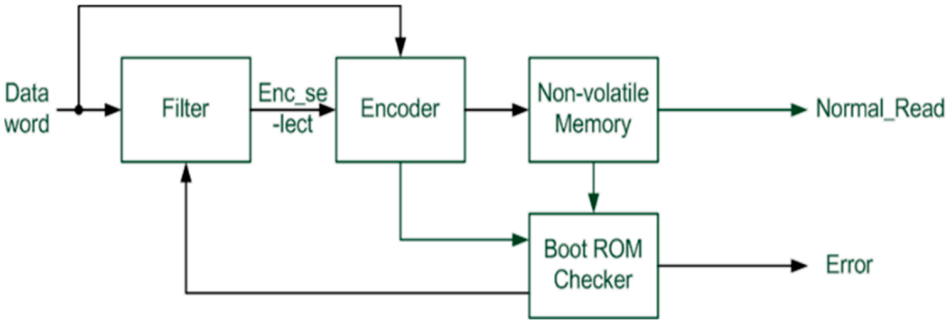

Figure 5. The architecture of the proposed two-dimensional parity with variable length error detection code (2D-VLEDC).

The same as those in the 2D-parity, the generated redundant bits are stored in the signature area. Figure 6 shows that the redundant bits are stored at the addresses from No. 21 to No. 22 as marked in the red box. Address 20 stores the *Y*-axis parity bits. For example, the bit-0 of data word in the address 20 is equal to XOR all bit-0 from address 0 to address 16 as presented in the green box. That is even parity check for all bit-0 from address 0 to address 16.

Before CPU boots up, Boot ROM checker module first performs the boot ROM checking. Boot ROM checker module reads data word, sends data word to filter module to perform encoding process again and finally compares the redundant bit in the NVM with re-calculated ones. If the results of the identical comparison are true, CPU will fetch instructions normally. If the results of the comparison are false, Boot ROM checker module will alarm an error signal to external interface.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	C0	C1	C2	C3	C4	C5	C6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0						
2	1	1	1	0	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	0	1	0	1	1	0	0	0	0	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1						
4	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0							
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1							
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	0							
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1							
8	1	1	1	0	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	0	1	0	1	1	0	0	0	0	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1							
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1							
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1							
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0							
13	1	1	0	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	0	1	1	1	1	0	1	1	0	1	1	0	0	
14	0	1	0	0	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1	1	0	0	1	0	1	0	1	1	0	1	0	0	0	0	1	0
15	1	0	1	0	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	1	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	0	1
16	1	1	1	0	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	0	1	0	1	1	0	0	0	0	1	1	1	1
17	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	0	0	0	1							
18	1	0	1	0	0	1	1	0	0	0	0	1	1	1	1	1	0	1	0	1	0	0	0	1	1	1	1	1	1	1	0	1							
19	1	0	1	1	0	0	1	0	0	0	0	1	0	1	1	1	1	1	0	1	0	0	0	1	1	1	1	0	0	0	0	0							

Figure 6. 2D-VLEDC applied in NVM.

3. Experiments

This section illustrates an experiment to evaluate the performance of the proposed 2D-VLEDC. The performance comparison between other methods of memory protection and 2D-VLEDC are given. Table 2 shows the filer rule applied to the experiments. One-bit parity was applied to those data words with less than half of content is "0". The 7-bit ECC was applied to those data words with more than half of content is "1". As shown in Table 2, the 1-bit parity is used when the number of "0" is 0–15 in an address date. On the other hand, a 7-bit ECC is applied to the system when the number of "0" becomes 16–31.

Table 2. Filter rule of the proposed 2D-VLEDC.

Count of Zero	ECC Encode
0~15	1-bit parity
16~31	7-bit ECC

3.1. Experiment Flow

The experiments were exercised with different ROM sizes from 8 × 32, 16 × 32, 32 × 32, 64 × 32, 128 × 32, to 256 × 32. For each ROM size, 100 trials are exercised, and each trial runs 10^6 times of fault injection and error detection evaluation. The flow chart of the provided experiments is shown in Figure 7 and described step by step as follows:

Step 1. Specify ROM size and target simulation times (sim_times). The parameter "sim_times" is defined as 10^6 times here.

Step 2. Repeat step 3 to step 9 100 times, and go to step 10.

Step 3. Generate a random content of ROM.

Step 4. Based on generated random content in the step 3, generate the encoding results as golden patterns of 1-bit parity, 2D parity, 7-bits ECC and 2D-VLEDC.

Step 5. Repeat step 6 to step 8 10^6 times, and go to step 9.

Step 6. Inject four random errors into the ROM randomly.

Step 7. Perform the decoding flow of 1-bit parity, 2D parity, 7-bits ECC and 2D-VLEDC with the injected-errors ROM.

Step 8. Check error detection of each decoding result. Accumulate the number of un-detected errors into the un-detected error counter and then go to step 4.

Step 9. Log undetected error rate ($Rate_{ud}$) and size overhead of each memory protection technology in the log-files. Go to step 2.

$$Rate_{ud} = (\text{un-detected error times}/\text{sim\_times}) \times 100\% \quad (1)$$

$$\text{size\_overhead} = (\text{redundant bits})/(\text{total bits of memory}) \quad (2)$$

Step 10. Analyze 95% confidence interval from the log-files.

Step 11. Finish experiment. Therefore, there are 100 $Rate_{ud}$ data for each ROM size with 1-bit parity, 2D parity, 7-bits ECC and 2D-VLEDC.

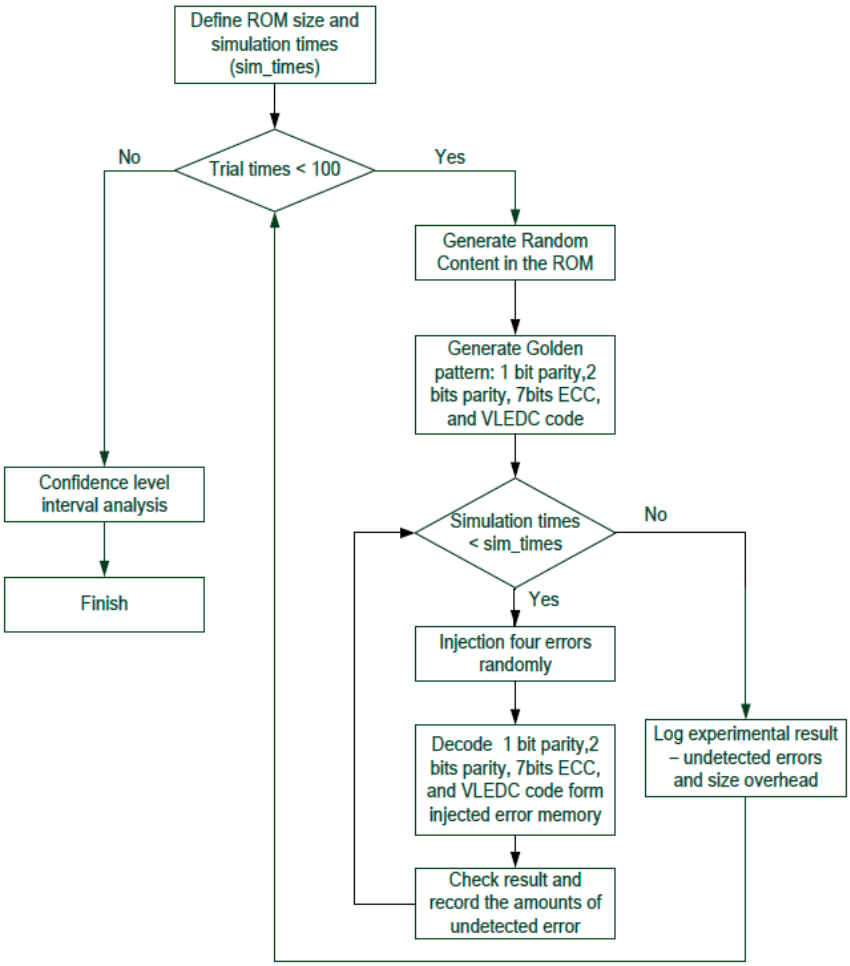

Figure 7. The flow of experiment.

3.2. Performance Comparison

The result of analyzing 95% confidence level interval with 100 $Rate_{ud}$ data for each ROM size are shown in the Table 3. In the Table 3, "Margin of Error" = $1.96 \times \sigma/n^{0.5}$. "StdDev" is the abbreviation of "Standard Deviation". "UDR" is the abbreviation of "Un-Detected error Rate". Each "Average", "StdDev" and "Margin of Error" value comes from the 100 trials, and each trial has 10^6 fault injection simulations. From the 10^6 fault injection simulations, the $Rate_{ud}$ (UDR in the Table 3) from the Equation (1) can be obtained.

Figure 8 shows the undetected error rate of each methodology versus ROM size. The point of the value in the Figure 8 is the field of "Average" in the Table 3 with the process of $\log_{10}$. 2D-VLEDC has better performance at the ROM size 8×32. With the growing of the ROM size, ECC is getting better performance than the 2D-VLEDC. However, the error detection performance has no significant difference between them.

Figure 9 shows the size overhead of each methodology versus ROM size. It was calculated by the Equation (2). The overhead of 2D-parity gradually reduces and approaches the overhead size of the 1-bit parity with the growing of the ROM size. 2D-VLEDC has approximate 10 times improvement in error coverage rate over the 2D-parity, but pays for 10~12% overhead. The results show that 2D-VLEDC needs 15% redundancy while ECC needs 22% redundancy in average. The 2D-VLEDC has not only the improvement of error coverage rate that approximates the coverage of ECC but also has 30% less overhead than the ECC's. For address of n, the required worst-case and best-case redundant bits are $7n$ and n, respectively, under the 2D-VLEDC.

Table 3. The comparison of the overhead of redundant bits. ROM (Read only memory); UDR (Un-Detected error Rate); StdDev (Standard Deviation).

Method/Statistics \ ROM		8 × 32	16 × 32	32 × 32
1-bit_parity UDR (%)	Average	4.88	2.39	1.2
	StdDev	8.14×10^{-2}	3.13×10^{-2}	1.44×10^{-2}
	Margin of Error	1.59×10^{-2}	6.14×10^{-3}	2.83×10^{-3}
2D_parity UDR (%)	Average	1.1×10^{-3}	4.25×10^{-4}	1.66×10^{-4}
	StdDev	3.78×10^{-4}	2.11×10^{-4}	1.17×10^{-4}
	Margin of Error	7.41×10^{-5}	4.15×10^{-5}	2.3×10^{-5}
7-bit-ECC UDR (%)	Average	2.12×10^{-4}	2.8×10^{-5}	5.0×10^{-6}
	StdDev	1.4×10^{-4}	4.94×10^{-5}	2.19×10^{-5}
	Margin of Error	2.76×10^{-5}	9.68×10^{-6}	4.29×10^{-6}
2D-VLEDC UDR (%)	Average	7.7×10^{-5}	2.9×10^{-5}	1.0×10^{-5}
	StdDev	1.4×10^{-4}	5.7×10^{-5}	3.33×10^{-5}
	Margin of Error	2.76×10^{-5}	1.12×10^{-5}	6.53×10^{-6}
Method/Statistics \ ROM		**64 × 32**	**128 × 32**	**256 × 32**
1-bit_parity UDR (%)	Average	6.22×10^{-1}	3.35×10^{-1}	1.93×10^{-1}
	StdDev	6.21×10^{-1}	5.78×10^{-3}	1.36×10^{-3}
	Margin of Error	1.91×10^{-3}	1.5×10^{-3}	5.14×10^{-4}
2D_parity UDR (%)	Average	7.8×10^{-5}	3.85×10^{-5}	2.09×10^{-5}
	StdDev	1.03×10^{-4}	5.26×10^{-5}	5.14×10^{-5}
	Margin of Error	2.02×10^{-5}	1.36×10^{-5}	1.53×10^{-5}
7-bit-ECC UDR (%)	Average	3.0×10^{-6}	1.92×10^{-6}	9.3×10^{-7}
	StdDev	1.32×10^{-5}	1.32×10^{-5}	6.0×10^{-6}
	Margin of Error	3.44×10^{-6}	3.44×10^{-6}	1.82×10^{-6}
2D-VLEDC UDR (%)	Average	1.3×10^{-5}	7.01×10^{-6}	2.32×10^{-6}
	StdDev	5.8×10^{-5}	5.29×10^{-5}	1.52×10^{-5}
	Margin of Error	1.13×10^{-5}	1.37×10^{-5}	4.55×10^{-6}

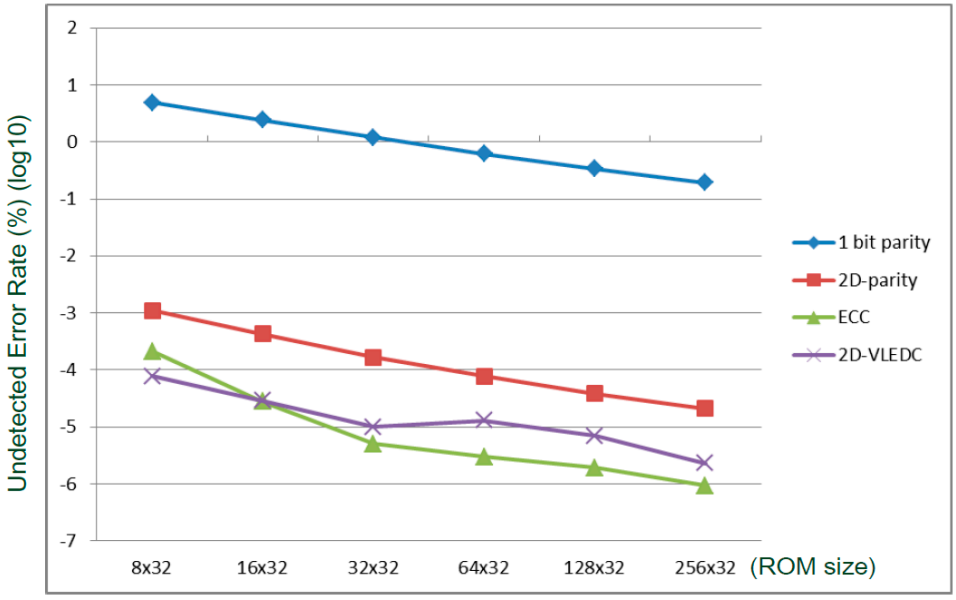

Figure 8. The comparison of error coverage performance.

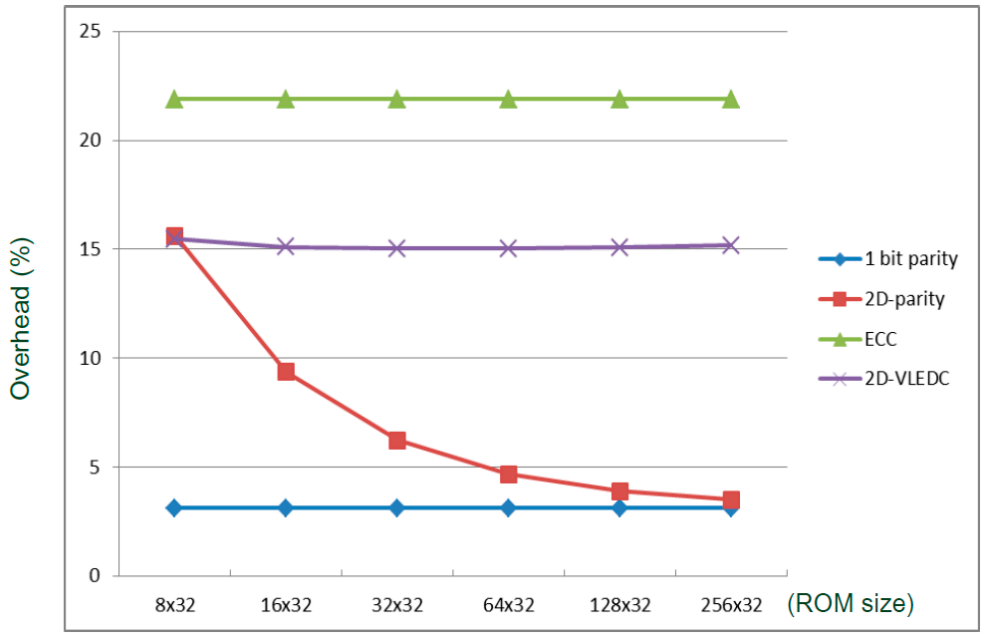

Figure 9. The comparison of the overhead of redundant bits. The overhead was calculated by Equation (2).

4. Conclusions

2D-parity has a good error coverage rate, with very low cost when applied to NVM or ROM protection. Combining with the variable length of the error detection code, the 2D-VLEDC improves flexibility, achieving better tradeoffs between reliability and cost. In our experiments, a very simple rule has been used in accordance with the asymmetric phenomenon between the 1-to-0 and 0-to-1 failures. 2D-VLEDC not only improves error coverage rate but also has the less overhead as compared with other methodologies. NVM cell designer, NVM controller and system designer can co-work and come out an optimized rule to be satisfied with the requirement of reliability. In the future, 2-bits parity, 3-bits checksum or 4-bits CRC could be added into the encoder module. The rule table can also be adjusted according to the requirement of reliability. The more zero counts included in the data word, the more redundant bits applied to memory protections.

Author Contributions: C.-S.A.G. co-proposed the idea, performed the data verification, revised the manuscript, and supervised the work. Y.-C.C. co-proposed the idea and wrote the manuscript. L.-R.H. provided funding and supported the project. C.-J.Y. co-proposed the idea and carried out the experiments. K.-M.J. supported this project. K.-L.L. contributed the technical details. J.-C.L. co-supervised this study.

Funding: This study was funded in part with grants from the Ministry of Science and Technology (MOST), Taiwan, under Grants MOST 105-2221-E-182-039, MOST 106-2221-E-182 -005-, and MOST 107-2221-E-182 -075 -MY2. This work is also supported in part by the Chang Gung Memorial Hospital (CGMH) under the contracts CMRPD2F0103, CMRPD2G0331, and CMRPD2H0041. Financial support provided by Industrial Technology Research Institute is appreciated.

Conflicts of Interest: The authors declare no competing interests.

References

1. Cellere, G.; Pellati, P.; Chimenton, A.; Wyss, J.; Modelli, A.; Larcher, L.; Paccagnella, A. Radiation Effects on Floating-Gate Memory Cells. *IEEE Trans. Nucl. Sci.* **2001**, *48*, 2222–2228. [CrossRef]
2. Gerardin, S.; Bagatin, M.; Paccagnella, A.; Grurmann, K.; Gliem, F.; Oldham, T.R.; Irom, F.; Nguyen, D.N. Radiation Effects in Flash Memories. *IEEE Trans. Nucl. Sci.* **2013**, *60*, 1953–1969. [CrossRef]
3. Ruby, P.D.; Belgal, H.P.; Wakchaure, Y.B.; Guo, X.; Nelson, S.E.; Salmons, S.M. Dynamic Read Channel Calibration for Non-Volatile Memory Devices. U.S. Patent 8,510,636 B2, 1 April 2011.

4. Park, J.; Bhunia, S.; Park, J. VL-ECC: Variable Data-Length Error Correction Code for Embedded Memory in DSP Applications. *IEEE Trans. Circuits Syst. II Express Briefs* **2014**, *61*, 120–124. [CrossRef]
5. Buttigieg, V.; Farrell, P.G. Variable-Length Error-Correcting Codes. *IEE Proc. Commun.* **2000**, *147*, 211–215. [CrossRef]