Error Detection and Correction Techniques for National Semiconductor's EEPROM Products

National Semiconductor
Application Note 482
February 1987

This application note provides the non-volatile memory system designer who cannot tolerate the very low failure rate associated with National Semiconductor's E^2PROMs, with a method to assure data integrity and extend the life span of the product.

With a minimum additional parts cost, the following error detection and correction techniques allow the designer to extend the usable life of an EEPROM device. The technique is applicable for applications requiring 100,000 or more erase/write cycles per register.

All EEPROMs fail with extended erase/write cycling. National Semiconductor EEPROMs fail in a statistically predictable and well behaved fashion as the number of erase/write cycles increase. The failure of one bit cell does not influence the operation of adjacent bit cells. Since bit failure is a gradual wearout phenomenon which only affects discrete cell locations one at a time, it is possible to apply simple encoding techniques which can determine the locations of cell failures so that faltering memory addresses can be avoided in the future.

Single parity checking is the simplest way to check for errors in a binary code. In a parity checking system an extra-parity-bit is chosen so that the number of 1s in the block of data, including the parity bit, is even. In practice this is accomplished using modulo 2 addition (i.e., $0 + 0 = 0$; $0 + 1 = 1$; $1 + 0 = 1$; $1 + 1 = 0$; $0 + 0 + 1 = 1$; etc.). Modulo 2 addition is quickly accomplished through an exclusive OR gate. When the data is read back, the number of ones are counted and the sum is checked to see if it is odd or even. An odd sum is an indication that an error occurred in the data. This method of single parity checking can detect the occurrence of an error in a block but it cannot be used to determine the exact location of the error to correct the bad data.

A natural extension of single parity checking is the Hamming code. A Hamming code uses several parity checks, instead of just one. This allows errors to be corrected as well as detected. Using bits in blocks of 7, where 4 of the bits are information and 3 are parity allows for error detection and correction of any single bit within the block, including the parity bits themselves.

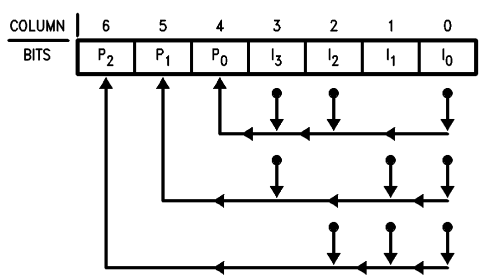

FIGURE 1. Computation Scheme for Parity Bits Using Hamming Code

The initial parity is calculated as shown in *Figure 1.* The parity bits are in columns 4, 5 and 6, while the actual information bits are in columns 0, 1, 2, and 3. The contents of each parity bit comes from summing the contents of a unique combination of three of the four information bits. The parity bit is chosen so that this sum will be an even number when added to the parity bit itself. Notice that each one of the parity bits calculates its contents by using different combinations of the data bits. Every data bit in the block has its information read at least twice. Using this overlapping scheme is what allows the code to correct errors.

Since there are only 4 bits of information there can be only $2^4 = 16$ possible combinations of 1s and 0s. These 16 possible correct combinations are listed on the code word table in Table I. When the encoded block is read back from memory, the same parity coding scheme is used again on the information bits and compared to the original parity bits. This forms what is called a syndrome. If any errors have occurred in the 7-bit block their locations can be determined and the errors corrected. Table II shows the decoding matrix which is used on the syndromes to determine the location of an error. If no errors occurred the syndrome will be 000. Table III shows all the combinations of the 7-bit block. Note that there are only 128 possible variations of 1s and 0s in the block: (7 mistake combinations per block + 1 correct combination per block) $\times$ (16 possible block combinations). All these combinations can be stored in a table and called up quickly to check for possible data errors without the need to even create a syndrome upon reading a word. For example, suppose we want to store the data 1000. From Table I we see that the 7-bit block would be 1111000 after the Hamming code had been applied. If information bit 3 for example goes bad, then the new block would read 1110000. This is case number 112 in Table III, and we see that the correct information is 1000. With Table III available in the computer memory, the received codeword can be corrected automatically. An array of 128 bytes can provide both the corrected information and the syndrome information.

The 7-bit codeword works nicely with National Semiconductor's serial EEPROMs because they are organized as arrays of 16-bit registers. Each 16 bit register is modified or accessed with a simple-serial protocol. The 16-bit unit can be partitioned two eight-bit bytes. Each byte can contain a seven-bit codeword and one-bit flag that indicates whether an error has been previously detected in the byte. This scheme provides one byte of error corrected information per 16-bit register. Slightly more elaborate systems can be used which will detect and correct more errors if additional parity bits are added to the data.

TABLE I. Encoding Table for Hamming Code

	Sixteen Code Words						
	Parity Bits			Information Bits			
	P 2	P 1	P 0	I 3	I 2	I 1	I 0
0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	1
2	1	0	1	0	0	1	0
3	0	1	1	0	0	1	1
4	0	1	1	0	1	0	0
5	1	0	1	0	1	0	1
6	1	1	0	0	1	1	0
7	0	0	0	0	1	1	1
8	1	1	1	1	0	0	0
9	0	0	1	1	0	0	1
10	0	1	0	1	0	1	0
11	1	0	0	1	0	1	1
12	1	0	0	1	1	0	0
13	0	1	0	1	1	0	1
14	0	0	1	1	1	1	0
15	1	1	1	1	1	1	1

TABLE II. Syndrome Decoding Table for Hamming Code

Syndrome			Meaning
0	0	0	No error detected
0	0	1	Check bit 0 in error
0	1	0	Check bit 1 in error
0	1	1	Information bit 2 corrected
1	0	0	Check bit 2 in error
1	0	1	Information bit 1 corrected
1	1	0	Information bit 0 corrected
1	1	1	Information bit 3 corrected

With this added data protection the reliability of EEPROMs can be extended because the probability of two or more cells failing on the same codeword is low. To illustrate the Hamming code, an experiment on 16 devices with 1k bits each was conducted. The experiment results are shown in Table IV. While the first bit failure was detected somewhere between 12,589 and 15,849 cycles, the Hamming code just described would have protected against the loss of data until somewhere between 79,433 and 100,000 erase/write cycles. Notice that 55 bit failures were indicated when the first Hamming code failure was detected. This is to be expected because a Hamming failure will not occur until two or more bits within a particular group of seven bits have failed.

TABLE III. Information Retrieval Table for All Possible Combinations of Single-Bit-Correct Hamming Code

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

TABLE III. Information Retrieval Table for All Possible Combinations of Single-Bit-Correct Hamming Code (Continued)

	Received Codeword							Syndrome Bits			Corrected Information			
32	0	1	0	0	0	0	0	0	1	0	0	0	0	0
33	0	1	0	0	0	0	1	1	0	0	0	0	0	1
34	0	1	0	0	0	1	0	1	1	1	1	0	1	0
35	0	1	0	0	0	1	1	0	0	1	0	0	1	1
36	0	1	0	0	1	0	0	0	0	1	0	1	0	0
37	0	1	0	0	1	0	1	1	1	1	1	1	0	1
38	0	1	0	0	1	1	0	1	0	0	0	1	1	0
39	0	1	0	0	1	1	1	0	1	0	0	1	1	1
40	0	1	0	1	0	0	0	1	0	1	1	0	1	0
41	0	1	0	1	0	0	1	0	1	1	1	1	0	1
42	0	1	0	1	0	1	0	0	0	0	1	0	1	0
43	0	1	0	1	0	1	1	1	1	0	1	0	1	0
44	0	1	0	1	1	0	0	1	1	0	1	1	0	1
45	0	1	0	1	1	0	1	0	0	0	1	1	0	1
46	0	1	0	1	1	1	0	0	1	1	1	0	1	0
47	0	1	0	1	1	1	1	1	0	1	1	1	0	1
48	0	1	1	0	0	0	0	0	1	1	0	1	0	0
49	0	1	1	0	0	0	1	1	0	1	0	0	1	1
50	0	1	1	0	0	1	0	1	1	0	0	0	1	1
51	0	1	1	0	0	1	1	0	0	0	0	0	1	1
52	0	1	1	0	1	0	0	0	0	0	0	1	0	0
53	0	1	1	0	1	0	1	1	1	0	0	1	0	0
54	0	1	1	0	1	1	0	1	0	1	0	1	0	0
55	0	1	1	0	1	1	1	0	1	1	0	0	1	1
56	0	1	1	1	0	0	0	1	0	0	1	0	0	0
57	0	1	1	1	0	0	1	0	1	0	1	0	0	1
58	0	1	1	1	0	1	0	0	0	1	1	0	1	0
59	0	1	1	1	0	1	1	1	1	1	0	0	1	1
60	0	1	1	1	1	0	0	1	1	1	0	1	0	0
61	0	1	1	1	1	0	1	0	0	1	1	1	0	1
62	0	1	1	1	1	1	0	0	1	0	1	1	1	0
63	0	1	1	1	1	1	1	1	0	0	1	1	1	1
64	1	0	0	0	0	0	0	1	0	0	0	0	0	0
65	1	0	0	0	0	0	1	0	1	0	0	0	0	1
66	1	0	0	0	0	1	0	0	0	1	0	0	1	0
67	1	0	0	0	0	1	1	1	1	1	1	0	1	1
68	1	0	0	0	1	0	0	1	1	1	1	1	0	0
69	1	0	0	0	1	0	1	0	0	1	0	1	0	1
70	1	0	0	0	1	1	0	0	1	0	0	1	1	0
71	1	0	0	0	1	1	1	1	0	0	0	1	1	1
72	1	0	0	1	0	0	0	0	1	1	1	1	0	0
73	1	0	0	1	0	0	1	1	0	1	1	0	1	1
74	1	0	0	1	0	1	0	1	1	0	1	0	1	1
75	1	0	0	1	0	1	1	0	0	0	1	0	1	1
76	1	0	0	1	1	0	0	0	0	0	1	1	0	0
77	1	0	0	1	1	0	1	1	1	0	1	1	0	0
78	1	0	0	1	1	1	0	1	0	1	1	1	0	0
79	1	0	0	1	1	1	1	0	1	1	1	0	1	1

TABLE III. Information Retrieval Table for All Possible Combinations of Single-Bit-Correct Hamming Code (Continued)

	Received Codeword							Syndrome Bits			Corrected Information			
80	1	0	1	0	0	0	0	1	0	1	0	0	1	0
81	1	0	1	0	0	0	1	0	1	1	0	1	0	1
82	1	0	1	0	0	1	0	0	0	0	0	0	1	0
83	1	0	1	0	0	1	1	1	1	0	0	0	1	0
84	1	0	1	0	1	0	0	1	1	0	0	1	0	1
85	1	0	1	0	1	0	1	0	0	0	0	1	0	1
86	1	0	1	0	1	1	0	0	1	1	0	0	1	0
87	1	0	1	0	1	1	1	1	0	1	0	1	0	1
88	1	0	1	1	0	0	0	0	1	0	1	0	0	0
89	1	0	1	1	0	0	1	1	0	0	1	0	0	1
90	1	0	1	1	0	1	0	1	1	1	0	0	1	0
91	1	0	1	1	0	1	1	0	0	1	1	0	1	1
92	1	0	1	1	1	0	0	0	0	1	1	1	0	0
93	1	0	1	1	1	0	1	1	1	1	0	1	0	1
94	1	0	1	1	1	1	0	1	0	0	1	1	1	0
95	1	0	1	1	1	1	1	0	1	0	1	1	1	1
96	1	1	0	0	0	0	0	1	1	0	0	0	0	1
97	1	1	0	0	0	0	1	0	0	0	0	0	0	1
98	1	1	0	0	0	1	0	0	1	1	0	1	1	0
99	1	1	0	0	0	1	1	1	0	1	0	0	0	1
100	1	1	0	0	1	0	0	1	0	1	0	1	1	0
101	1	1	0	0	1	0	1	0	1	1	0	0	0	1
102	1	1	0	0	1	1	0	0	0	0	0	1	1	0
103	1	1	0	0	1	1	1	1	1	0	0	1	1	0
104	1	1	0	1	0	0	0	0	0	1	1	0	0	0
105	1	1	0	1	0	0	1	1	1	1	0	0	0	1
106	1	1	0	1	0	1	0	1	0	0	1	0	1	0
107	1	1	0	1	0	1	1	0	1	0	1	0	1	1
108	1	1	0	1	1	0	0	0	1	0	1	1	0	0
109	1	1	0	1	1	0	1	1	0	0	1	1	0	1
110	1	1	0	1	1	1	0	1	1	1	0	1	1	0
111	1	1	0	1	1	1	1	0	0	1	1	1	1	1
112	1	1	1	0	0	0	0	1	1	1	1	0	0	0
113	1	1	1	0	0	0	1	0	0	1	0	0	0	1
114	1	1	1	0	0	1	0	0	1	0	0	0	1	0
115	1	1	1	0	0	1	1	1	0	0	0	0	1	1
116	1	1	1	0	1	0	0	1	0	0	0	1	0	0
117	1	1	1	0	1	0	1	0	1	0	0	1	0	1
118	1	1	1	0	1	1	0	0	0	1	0	1	1	0
119	1	1	1	0	1	1	1	1	1	1	1	1	1	1
120	1	1	1	1	0	0	0	0	0	0	1	0	0	0
121	1	1	1	1	0	0	1	1	1	0	1	0	0	0
122	1	1	1	1	0	1	0	1	0	1	1	0	0	0
123	1	1	1	1	0	1	1	0	1	1	1	1	1	1
124	1	1	1	1	1	0	0	0	1	1	1	0	0	0
125	1	1	1	1	1	0	1	1	0	1	1	1	1	1
126	1	1	1	1	1	1	0	1	1	0	1	1	1	1
127	1	1	1	1	1	1	1	0	0	0	1	1	1	1

TABLE IV. Hamming Code Experimental Demonstration on 16 Devices of 1k Bits Each				
Erase/Write Cycles	**Total Bit Failures**	**Total Codeword Failures**	**Percent Bit Failures**	**Percent Codeword Failures**
1000	0	0	0.00%	0.00%
1259	0	0	0.00%	0.00%
1585	0	0	0.00%	0.00%
1995	0	0	0.00%	0.00%
2512	0	0	0.00%	0.00%
3162	0	0	0.00%	0.00%
3981	0	0	0.00%	0.00%
5012	0	0	0.00%	0.00%
6310	0	0	0.00%	0.00%
7943	0	0	0.00%	0.00%
10000	0	0	0.00%	0.00%
12589	1	0	0.01%	0.00%
15849	1	0	0.01%	0.00%
19953	1	0	0.01%	0.00%
25119	1	0	0.01%	0.00%
31623	3	0	0.02%	0.00%
39811	4	0	0.02%	0.00%
50119	10	0	0.06%	0.00%
63096	16	0	0.10%	0.00%
79433	55	1	0.34%	0.05%
100000	103	3	0.63%	0.15%

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408