SDAN 20

DNA 3388T

TRANSIENT UPSET TOLERATION AS AN EMP HARDENING TECHNIQUE

R & D Associates
Box 3580
Santa Monica, California 90403

~~5 August 1974~~
January 1974
Topical Report

CONTRACT No. DNA 001-72-C-0197

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

THIS WORK SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER SUBTASK P99QAXDB001-10.

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, D. C. 20305

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DNA 3388T	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TRANSIENT UPSET TOLERATION AS AN EMP HARDENING TECHNIQUE		5. TYPE OF REPORT & PERIOD COVERED Topical Report
		6. PERFORMING ORG. REPORT NUMBER RDA-TR-2301-010 Jan 1974
7. AUTHOR(s) W. R. Graham J. B. Houston		8. CONTRACT OR GRANT NUMBER(s) DNA 001-72-C-0197
9. PERFORMING ORGANIZATION NAME AND ADDRESS R & D Associates P.O. Box 3580 Santa Monica, California 90403		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NWED Subtask P99QAXDB001-10
11. CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, D. C. 20305		12. REPORT DATE 5 August 1974 13. NUMBER OF PAGES 36
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This work sponsored by the Defense Nuclear Agency under Subtask P99QAXDB001-10.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

EMP
Digital Electronics
Transient Upset
EMP Hardening

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Both transient circuit upset and permanent component damage can produce serious malfunctions in modern electronics equipment. Although the issue is not yet clear, it may be possible to design and select components which have a much higher threshold to permanent damage than to transient upset. If this proves to be possible, then a degree of electromagnetic pulse

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

(EMP) hardness would be achieved through designing systems to be completely insensitive to the effects of transient circuit upset.

A number of techniques have been developed for rendering circuits insensitive to transient upset. However, the capabilities, limitations, and penalties of these techniques are not widely known to the EMP community. Only in rare instances have these techniques been integrated into a system design so thoroughly that the resulting overall system was insensitive to transient upset.

In this paper, the characteristics of the developed techniques for tolerating transient upset are described, various overall system design approaches are discussed, and the relation of specific upset toleration techniques to the system design approaches are illustrated.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION 1

INTRODUCTION

1.1 Objectives.

Both transient circuit upset and permanent component damage can produce serious malfunctions in modern digital electronics equipment. In most cases, it is possible to design digital systems such as aircraft avionics, digital communications, missile guidance sets, etc. in such a way that they are not permanently damaged by the electromagnetic pulse (EMP) environment. Figure 1 shows an example of good housekeeping techniques designed to minimize black-box susceptibility to permanent damage from electromagnetic transients induced on the long, antenna-like cable runs between boxes.

1.2 System.

It is a much more difficult hardening task to design a system so that transient errors cannot be induced in data transmission channels by external electromagnetic interference. One way to protect data stream from transient errors is by extensive shielding. Another is by the use of very high-level signals. However, a considerable improvement in overall system hardening trade-offs may result from designing into the system, an insensitivity to transient errors in digital data transmission. Technologies other than EMP hardening design have developed a variety of methods for achieving such an insensitivity. These techniques should be included in the methods of EMP hardening design.

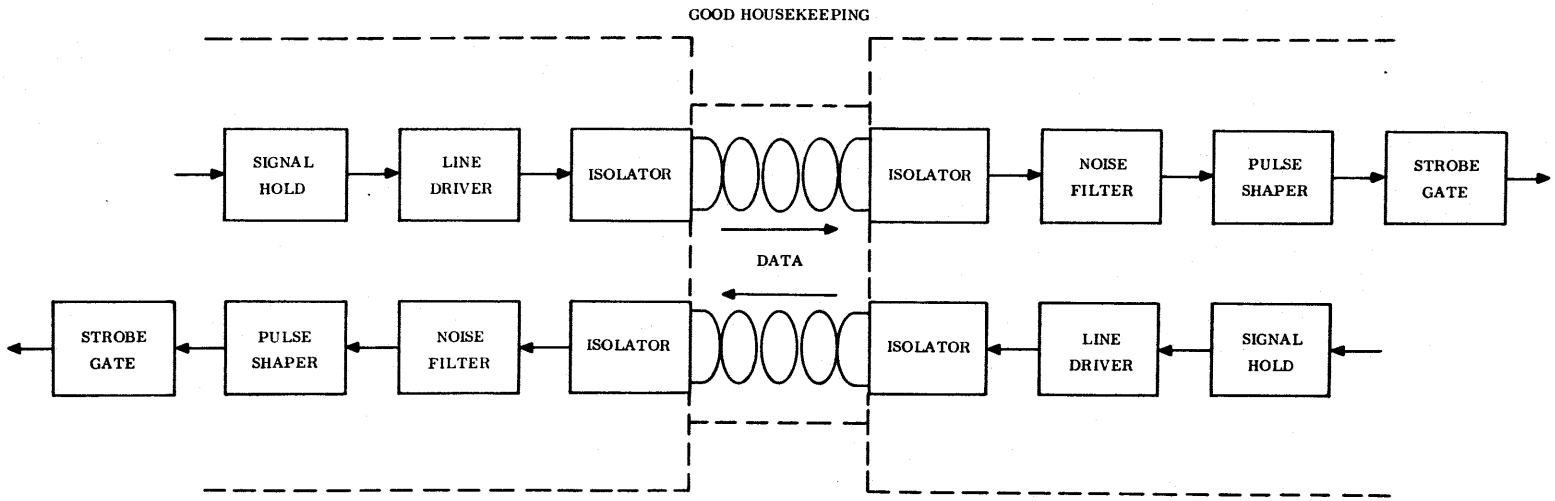


Figure 1. Good housekeeping practice in digital data transmission.

SECTION 2

GROUND RULES

2.1 Objectives.

The ground rules for the system must be established before the system can be designed. The rules should address the following issues:

Rule 1. What responses to a transient error can be tolerated? Some possible responses are: do nothing, skip the data sample, use the previous data, correct the data, retransmit the data, etc.

Rule 2. What kind of error can occur? Do the errors occur in random bits or in a burst? Random bit errors have no relation to each other, while a burst has a maximum number of consecutive bits that can be in error. EMP-induced errors usually are burst errors. The statistics of the errors, including the maximum rate of occurrence, should be specified.

Rule 3. How much delay can be tolerated in processing the signal? In general, the more allowable delay, the more ways data transmission errors can be overcome.

2.2 Design strategies.

When these ground rules have been determined, three alternate design strategies can be evaluated and a selection made of the one best suited to the specific need:

(1) Error toleration.

(2) Error rejection.

(3) Error correction.

Several ways to implement each of these strategies are discussed in the following section.

SECTION 3

ERROR TOLERATION

3.1 Slow system response. If the system is slow enough, then the error might not cause a problem. As long as the error rate is low enough and the error is of short duration (few samples), then the system will not respond to the error. Mechanical and thermal systems often have response times which are much longer than digital signaling times. Occasional errors in a stream of digital control signals to such systems will not affect their performance.

3.2 Signal averaging. By averaging the error with other data samples, as shown in Figure 2, the size of the error is reduced by 1-n, which may permit the system to perform adequately. However, the error in the average persists over n sample times so that the time integral of the error is not affected by averaging. Averaging also adds delay to the system response.

3.3 Signal resolution.

If the signal has n equally spaced levels of resolution, then a single error would disappear with signal averaging using 2n-1 samples.

As an example of how signal resolution and averaging would work, consider input and output signals having three levels having values 1, 2, and 3. The number of samples required to eliminate one error is then five. When the input data is a continuous stream of 1's with an error that forces the 3, the state of the system as a function of time is shown in Figure 3, and there is no error in the output.

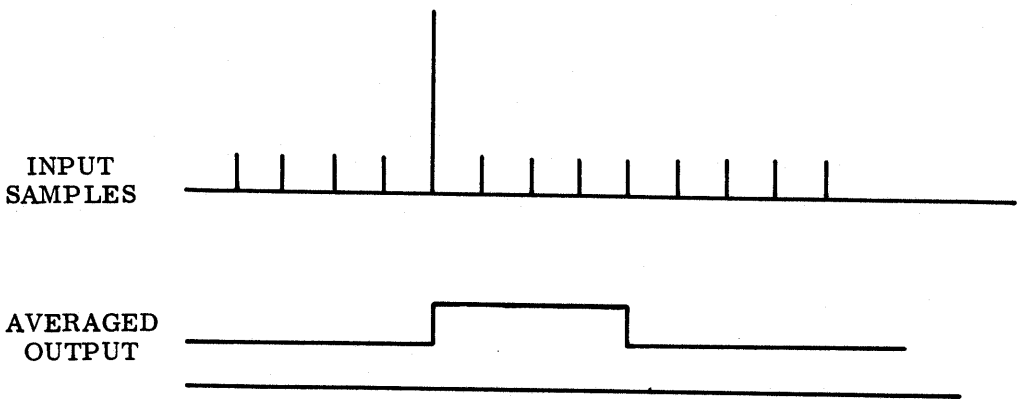


Figure 2. Signal averaging.

Input Data	.	.	.	1	1	1	1	1	1	3	1	1	1	1	1	1	.	.	.
Sum of Samples					.	.	.	5	5	7	7	7	7	7	5	5	.	.	.
Average of Samples					.	.	.	1	1	1.4	1.4	1.4	1.4	1.4	1	1	.	.	.
Output					.	.	.	1	1	1	1	1	1	1	1	1	.	.	.

Delay Caused by Signal Averaging

Figure 3. An example of signal averaging.

SECTION 4

ERROR REJECTION

4.1 Bounds checking. The simplest error-checking system is to check to see if the value represented by the data is reasonable. Physical reality often is the limiting item. The value must be physically possible. Physical bounds often can be placed on the magnitude of the data, the rate of change of the data, or on both.

4.2 Parity checks for a group of signals.

The parity of a group of data bits is set by first counting the number of 1's in the group. An additional bit, called the parity bit, is then added to the group and set to 0 or 1 so that the total number of 1's is odd, if odd parity is used, or even, if even parity is used. Checking the parity of a group of bits for evenness or oddness after the parity bit has been set allows one to determine whether the group has experienced a 1-bit error.

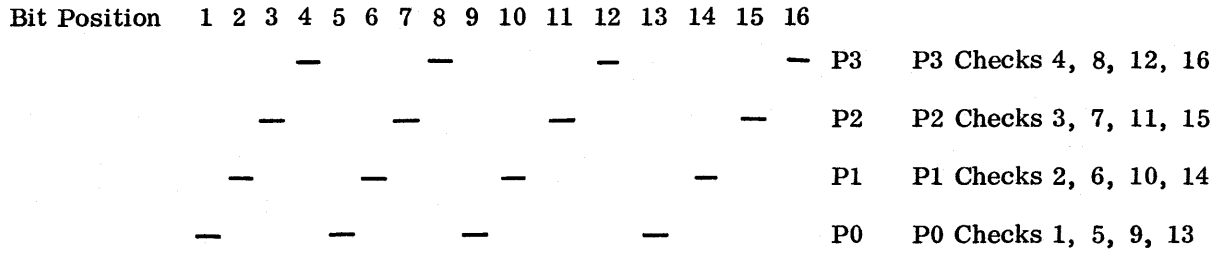
A single parity check will not work in the situation where two or more bits may be in error, since the alteration of an even number of 1's or 0's will not change the parity of the group. If multiple bit errors must be detected, additional parity groups and corresponding parity bits are required, as discussed, for example, in Paragraph 4.3.

The implementation of error detection and rejection techniques generally is not as difficult as the implementation of error-correction systems, which will be discussed later.

4.3 Parity checks for burst error detection.

This method of error detection is used when the errors can occur only in a burst. The burst length is defined as the number of bits from the first bit in error to the last bit in error. An error burst may contain some correct bits.

A simple system to detect burst errors uses the same number of parity checks as the maximum expected burst length. Figure 4 shows an example of such a system that will detect any burst of length 4 or less in a 16-bit word.



(P0, P1, P2, and P3 are included in bits 1-16)

This pattern of parity bits will detect a burst error up to length 4.

Figure 4. Parity check bits for 4-bit burst error detection.

SECTION 5

ERROR CORRECTION

5.1 Store and forward. This method, illustrated in Figure 5, requires storing the data to be sent until a positive acknowledgment is received that the transmission was accomplished without error. The data is transmitted with parity checks which are tested at the receiver. A reply is then sent back to the transmitter, and, if errors were detected, the data is retransmitted. This technique requires considerable hardware and sometimes considerable delay. The data always gets through, but the delay may be excessive. Memory is required in both the transmitter and receiver.

5.2 Data repetition.

This method consists of repeating the data several times and storing the results at the receiver. The groups are compared in corresponding bit locations using majority voting to determine the correct output data. The method requires the data to be stored at both ends of the line. The delay, in this case, is the time required to send all the repetitions of the data.

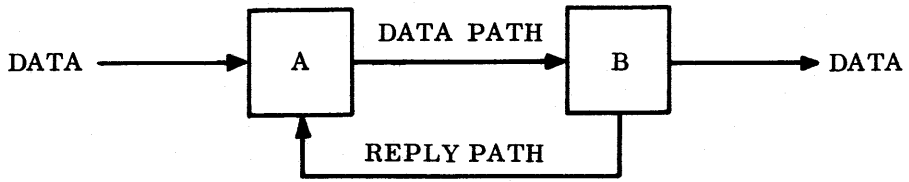
A word is the group of bits that is repeated. A burst error can last as long as the word when the repetition factor is 3. This method requires a single channel, but makes inefficient use of the transmission channel.

5.3 Three channels in parallel. This method uses three lines for each channel, with a majority gate at the receiving end. Errors in any single channel will not affect the output of the gate. A danger in using this method is that a common external influence could disrupt all three channels, resulting in an undetected, uncorrected error in the gate output.

5.4 Three parallel channels with staggered signals. By adding a different delay to each channel, as shown in Figure 6, this method will allow all three lines to be disrupted in the same manner, at the same time, without degrading the data. The unit delay must be greater than the burst length. The data will be delayed by twice the unit delay.

5.5 Check bits for error correction. By adding check bits, both random and burst errors can be corrected. The derivation of error-correcting codes is beyond the scope of this paper.* Error-correcting codes are best used when the

*"Error Correcting Codes." W. Wesley Peterson, MIT Press, 1961.



A

1. STORES THE DATA
2. TRANSMITS WITH CHECK BITS
3. RETRANSMITS IF REPLY INDICATES ERRORS

(REQUIRES A TO HAVE A PROTECTED MEMORY)

B

1. RECEIVES THE DATA
2. CHECKS FOR TRANSMISSION ERRORS
3. SENDS REPLY

Figure 5. Store and forward.

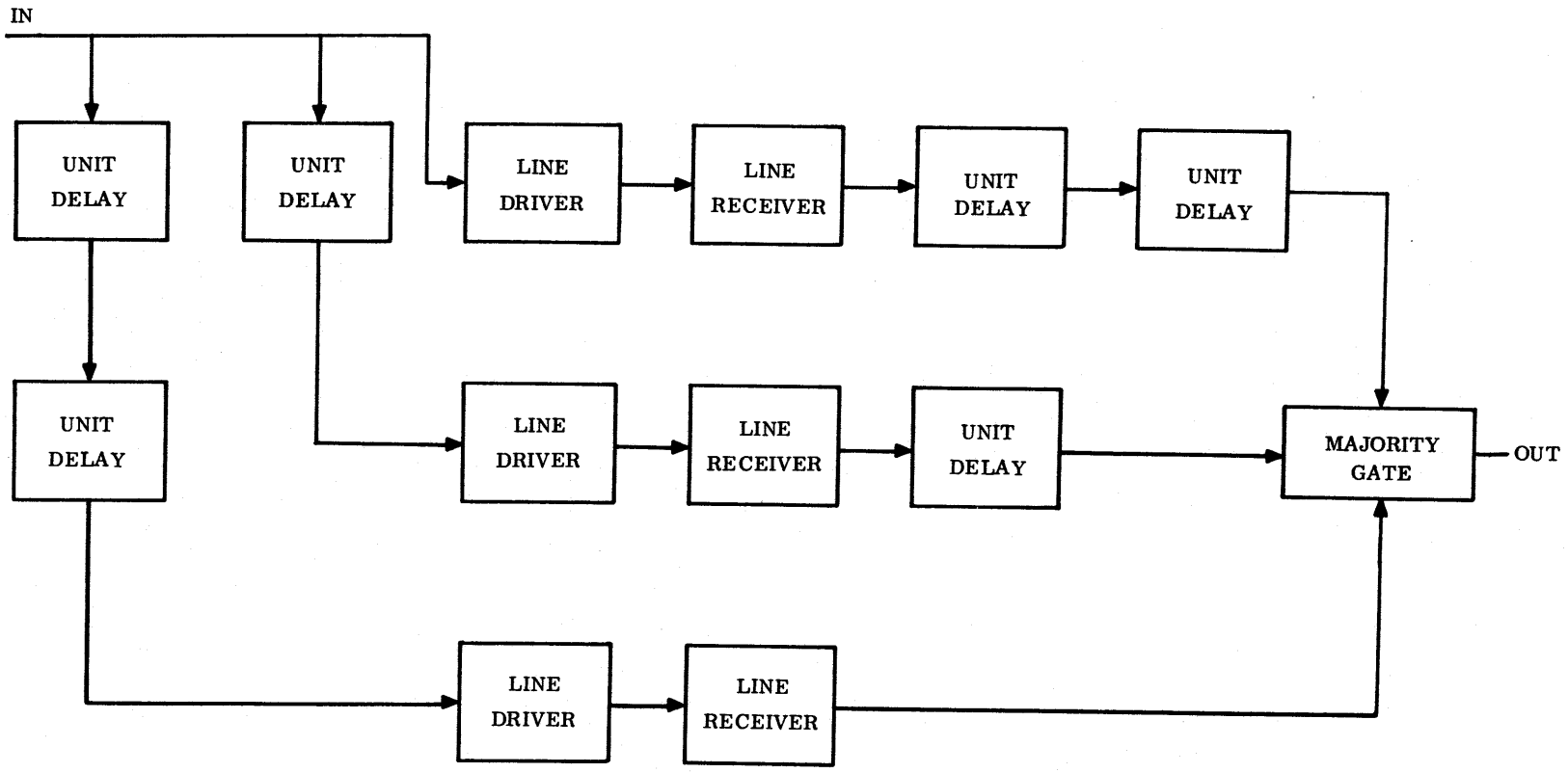


Figure 6. Three lines with delays.

length of the burst or the number of random errors is small compared to the length of the word. An example is shown in Figure 7 of a single bit error-correcting code. If two bits are in error, then the "correction" process always will introduce a third error. It is important to use an error-correction process that will handle the worse-case error situation.

5.6 More general codes for error correction.

The code types mentioned below are the most general for the type of error mentioned. Better codes exist, but are not as general as these. When the errors are expected to be random in nature, a Bose-Chaudhuri code is appropriate; while for single burst errors, a Fire code is used. The last type, Reed-Solomon, is used for multiple bursts of errors in the code word. Peterson's book, mentioned earlier, discusses each of these codes.

Figure 8 shows the characteristics of several random error-correction codes. A simple Fire code is shown in Figure 9. The code is 12 bits long, with 6 data bits, and 6 check bits. It can correct bursts up to length 2 and detect burst errors up to length 4. A few bursts of lengths 3 and 4 can be corrected. Some bursts of length greater than 4 would be detected. The complete table for decoding all combinations of parity failures is shown in Figure 10.

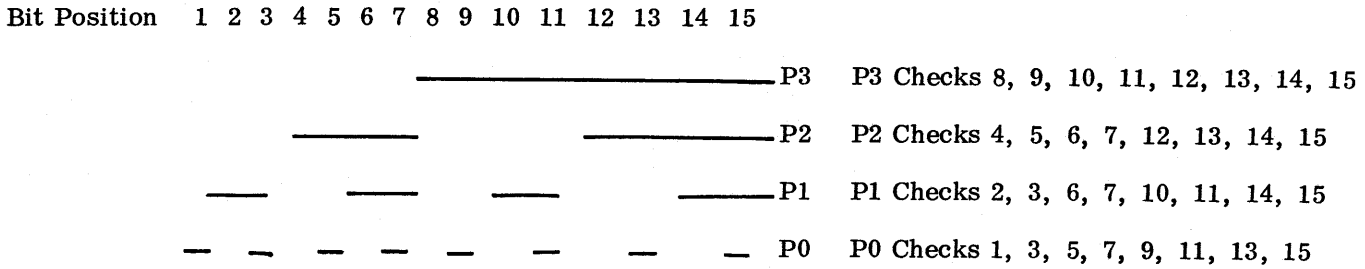
5.7 Burst error correction capabilities. The performance of optimum burst error-correcting codes and Fire codes are shown in Figures 11 and 12. Optimum is defined as having the maximum number of data bits for a given correctable burst length and number of check bits. While optimum codes are the most efficient in terms of information carried per bit, they may require sophisticated logic for encoding and decoding. Thus, one can be faced with a trade-off between data transmission efficiency and coding complexity. The circled dots on Figure 12 are for codes optimum to correct the burst length noted by each dot. The lines are constructed by reducing the number of data bits while keeping the burst length and check bits fixed.

For a Fire code designed to correct a burst error of length c-2 (if c is even), or $\frac{c+1}{2}$ (if c is odd), occurring in a word of length c (2^m-1) (assuming no common factors between c and 2^m-1), c + m check bits are required.

When c is one, and therefore, the burst length is one bit,

$$\frac{\text{data bits}}{\text{word bits}} = \frac{2^m - \frac{m}{2} - 2}{2^m - 1} \quad \text{and} \quad \frac{\text{data bits}}{\text{burst bits}} = 2^m - m - 2.$$

This is shown as the lower dashed wave in Figure 12.



The 15 bits shown include 11 data bits and the 4 parity bits.

Any bit in error will cause one or more parity checks to fail. Each single bit error causes a unique pattern of parity failures which can be interpreted as the binery representation of the position of the bit in error.

Figure 7. Parity check bits for 1-bit error correction.

Number of Information Bits	Total Number of Bits Needed			
	One Bit in Error		Two Bits in Error	
	Total Bits	Optimum	Total Bits	Optimum
1	3	Yes	5	Yes
2	5	Yes	8	Yes
3	8	No	10	No
4	10	No	11	Yes
5	9	Yes	13	No
6	11	Yes	14	Yes
7	13	Yes	--	--
8	14	Yes	--	--
9	15	Yes	17	Yes

Figure 8. Random error-correction capabilities of selected codes.

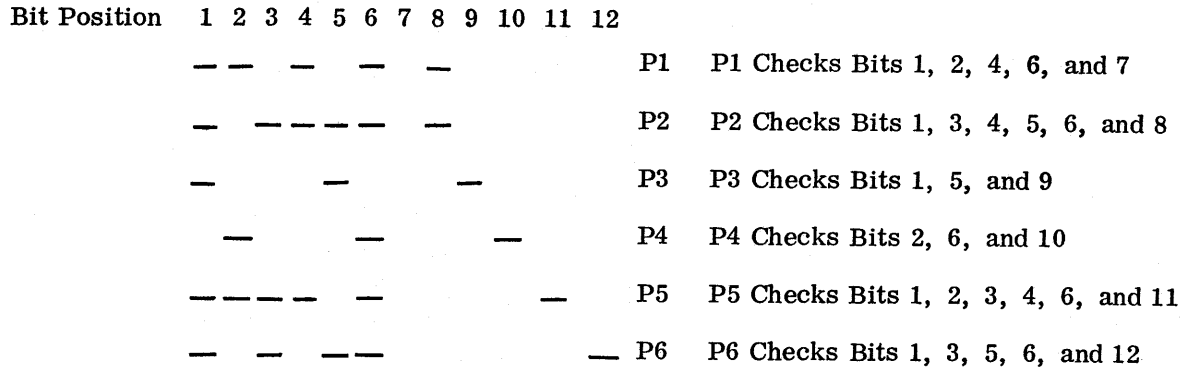


Figure 9. An example of Fire code.

Error Pattern						Error Correction Mask												Burst	Correctable
P1	P2	P3	P4	P5	P6	1	2	3	4	5	6	7	8	9	10	11	12	Length	
X												X						0	No correction required
	X												X					1	Yes
X	X											X	X					1	Yes
		X												X				2	Yes
X		X																1	Yes
	X	X											X	X				3	No
X	X	X																2	Yes
			X												X			3	No
X			X															1	Yes
	X		X															4	No
X	X		X															3	No
		X	X											X	X			4	No
X		X	X															2	Yes
	X	X	X															4	No
X	X	X	X															3	No
				X														4	No
X				X												X		1	Yes
				X														>4	No
	X			X														4	No
X	X			X					X									1	Yes
		X		X														3	No
X		X		X			X								X	X		2	Yes
	X	X		X														1	Yes
X	X	X		X														4	No
			X	X														>4	No
X		X	X	X						X	X							3	No
	X	X	X	X						X	X		X					2	Yes
X	X	X	X	X						X	X		X					4	No
					X												X	4	Yes
X					X			X	X									1	Yes
	X				X													2	Yes
X	X				X													>4	No
		X			X													>4	No
X		X			X					X		X	X					4	No
	X	X			X					X								4	Yes
X	X	X			X					X		X						1	No
			X		X					X		X						3	Yes
X			X		X				X		X	X						3	No
	X		X		X													4	Yes
X	X		X		X		X	X										>4	No
		X	X		X													2	Yes
X		X	X		X													4	No
	X	X	X		X	X	X											>4	No
X	X	X	X		X			X		X	X							2	Yes
				X	X					X	X							4	Yes
X				X	X													2	Yes
	X			X	X						X	X						>4	No
X	X			X	X													2	Yes
		X		X	X													4	No
X		X		X	X				X	X								2	Yes
	X	X		X	X													>4	No
X	X	X		X	X	X												1	Yes
			X	X	X						X		X					3	No
X			X	X	X						X	X						3	Yes
	X		X	X	X						X	X						2	Yes
X	X		X	X	X						X							1	Yes
		X	X	X	X									X				4	No
X		X	X	X	X													4	No
	X	X	X	X	X						X	X		X				4	Yes
X	X	X	X	X	X													4	No

Figure 10. Decoding table for code shown in Figure 9.

Data Bits / Word Bits		Data Bits / Maximum Correctable Burst Length		
Bits/Bits	Decimal	Bits/Bits	Decimal	Optimum
$\frac{9}{15}$	.6	$\frac{9}{3}$	3.0	Yes
$\frac{55}{63}$	.873	$\frac{55}{3}$	18.3	Yes
$\frac{245}{255}$	.961	$\frac{245}{3}$	81.7	Yes
$\frac{1011}{1023}$	.988	$\frac{1011}{3}$	337.0	Yes
$\frac{4081}{4095}$	.997	$\frac{4081}{3}$	1360.3	Yes
$\frac{499}{511}$	.977	$\frac{499}{4}$	124.75	Yes
$\frac{1010}{1023}$	.987	$\frac{1010}{4}$	252.5	Yes
$\frac{4080}{4095}$	.996	$\frac{4080}{4}$	1020.0	Yes
$\frac{6}{12}$	.5	$\frac{6}{2}$	3.0	No
$\frac{8}{15}$	.533	$\frac{8}{2}$	4.0	No
$\frac{51}{63}$	.810	$\frac{51}{3}$	17.0	No
$\frac{234}{255}$	.918	$\frac{234}{4}$	58.5	No
$\frac{985}{1023}$	.963	$\frac{985}{5}$	197.0	No
$\frac{4024}{4095}$	.983	$\frac{4024}{6}$	670.7	No

Figure 11. Selected burst error-correcting codes.

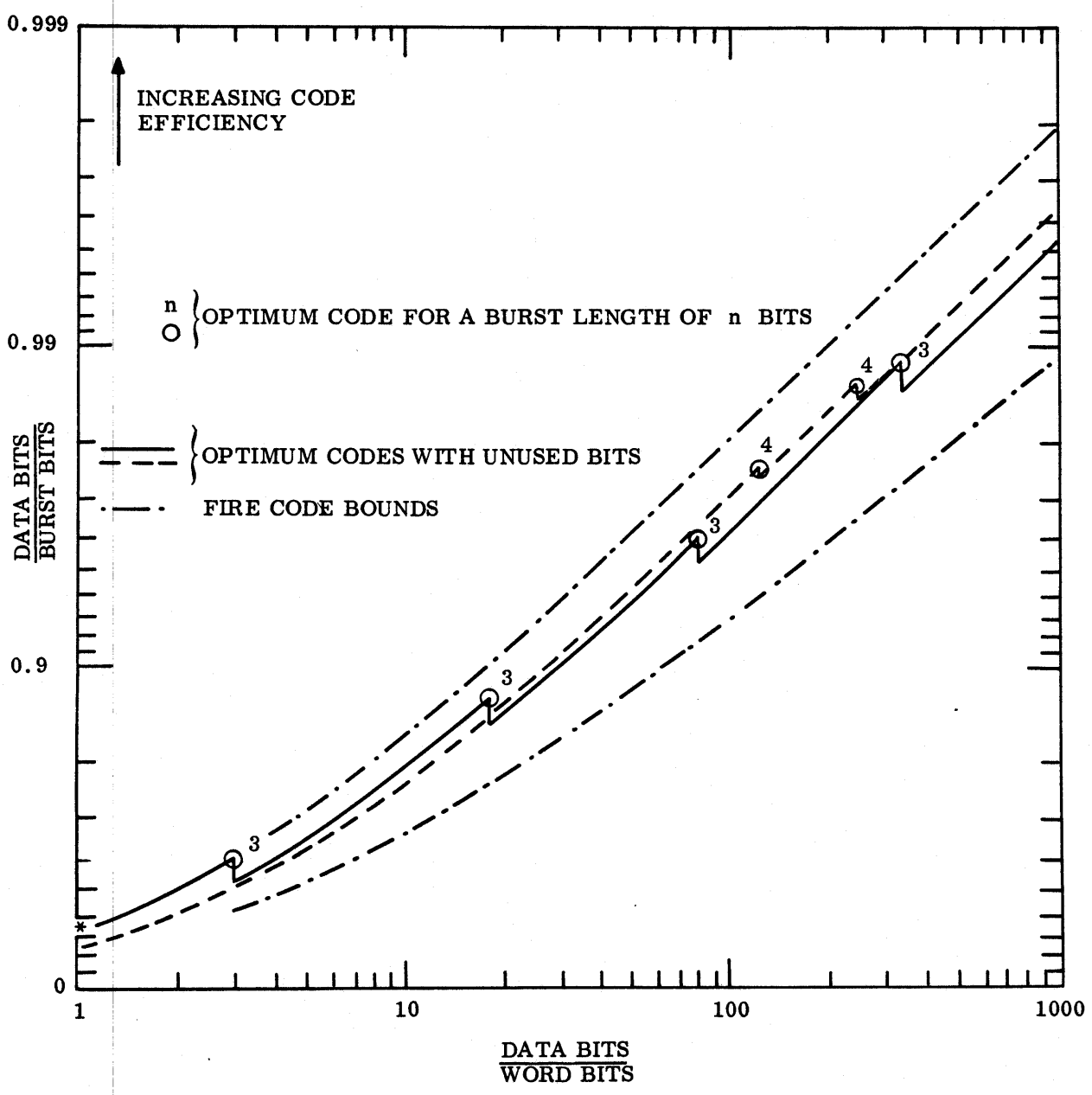


Figure 12. Efficiency of error-correcting codes.

As c becomes large, the ratios become approximately

$$\frac{\text{data bits}}{\text{word bits}} = \frac{2^m - 2}{2^m - 1} \quad \text{and} \quad \frac{\text{data bits}}{\text{burst bits}} = 2^{m+1} - 4.$$

This relation is shown as the upper dashed curve in Figure 12.

The general formulas for Fire codes are:

$$\frac{\text{data bits}}{\text{word bits}} = \frac{c \cdot (2^m - 2) - m}{c \cdot (2^m - 1)},$$

$$\frac{\text{data bits}}{\text{burst bits}} = \frac{c \cdot (2^{m+1} - 4) - 2m}{c + 1} \quad \text{for c odd, and}$$

$$\frac{\text{data bits}}{\text{burst bits}} = \frac{c \cdot (2^{m+1} - 4) - 2m}{c} \quad \text{for c even.}$$

As Figure 12 indicates, the efficiency of error-correcting codes, in terms of data bits/word bits, increases as the ratio of the data bits to the burst length increases. For a fixed burst length, however, as the number of data bits increases, the logic required to perform the error correction becomes more complex. Therefore, a trade-off between data-transmission efficiency and encoding-and-decoding complexity must be made.

SECTION 6

MEMORY

Several of the methods of detecting and correcting errors described above also apply to digital memories. The applicability can be seen through the use of a functional analogy. The write portion of a memory is comparable to the line driver, while the read portion is comparable to the line receiver. The storage mechanism is comparable to the information propagating on the channel. Memories are sometimes bit serial, sometimes bit parallel, and sometimes both. EMP would tend to produce burst errors in serial-system words and random errors in parallel-system words.

SECTION 7

CONCLUSION

Many design techniques have been devised to make digital equipment insensitive to transient errors in data transmission. Each of these techniques has individual advantages and disadvantages which must be evaluated in light of the specific design requirements. While error toleration, rejection, or correction techniques cannot overcome problems of permanent damage to components, these techniques can give systems an additional margin of hardness to transient upset produced by EMP.