AFOSR-TR- 91 0596

QUALCOMM, Inc.
10555 Sorrento Valley Road
San Diego, California 92121

FINAL TECHNICAL REPORT FOR THE RESEARCH IN MATHEMATICS AND COMPUTER SCIENCE: CALCULATION OF THE PROBABILITY OF UNDETECTED ERROR FOR CERTAIN ERROR DETECTION CODES

PHASE II

31 May 1991

Submitted To:

USAF, AFSC
Air Force Office of Scientific Research
Building 410
Bolling Air Force Base, DC 20332-6448

Contract No. F49650-90-C-0017

QUALCOMM, Inc.
10555 Sorrento Valley Road
San Diego, California 92121

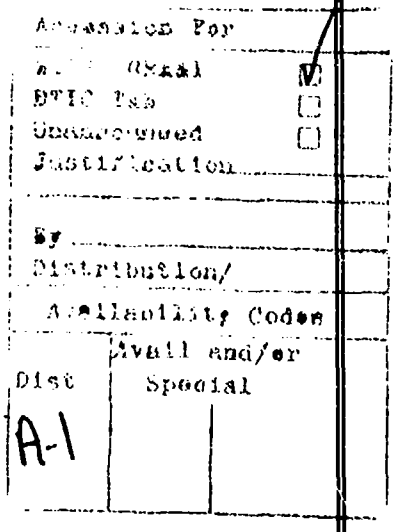
FINAL TECHNICAL REPORT FOR THE RESEARCH IN MATHEMATICS AND COMPUTER SCIENCE: CALCULATION OF THE PROBABILITY OF UNDETECTED ERROR FOR CERTAIN ERROR DETECTION CODES

PHASE II

31 May 1991

Submitted To:

USAF, AFSC
Air Force Office of Scientific Research
Building 410
Bolling Air Force Base, DC 20332-6448



Contract No. F49650-90-C-0017

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	31 May 91	Final Technical Report - 4/1/90 - 5/31/91

4. TITLE AND SUBTITLE

Final Technical Report for the Research in Mathematics and Computer Science:
Calculation of the Probability of Undetected Error for Certain Error Detection Codes (Phase II)

5. FUNDING NUMBERS

Contract Number F49650-90-C-0017
Program Code Number S0514A

61102F 2304/A3

6. AUTHOR(S)

Viterbi, Andrew J. - Wolf, Jack K. - Fredrickson, Lyle J. - Levin, Jeff A. -
Blakeney, Robert D. - Chun, Dexter T.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

QUALCOMM, Inc.
10555 Sorrento Valley Road
San Diego, California 92121-1617

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

USAF, AFSC
Air Force Office of Scientific Research
Building 410
Bolling AFB, DC 20332-6448

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

F49620-90-C-0017

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION /AVAILABILITY STATEMENT

USAF, AFSC
Air Force Office of Scientific Research
Building 410, Room C-124
Bolling AFB, DC 20332-6448
Attention: AFOSR/PKO

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Cyclic Redundancy Check (CRC) codes have become the standard means for detecting errors in messages that have been transmitted over a noisy communications channel. Unfortunately, even the very best CRC codes cannot detect all transmission errors.

In this report, we first describe a hardware device capable of evaluating the random error performance of an important class of CRC codes that are generated by polynomials of the form $g(x) = (x+1)\,p(x)$, where $p(x)$ is a primitive polynomial of degree (R-1). We then introduce a new burst error model and establish an equivalence between the burst and random error performance of cyclic codes. From this, we can extend the random error test results obtained from the hardware device to include burst errors.

Also included in this report is an intuitive look at the factors which lead to good code performance, and an overview of a supplemental hardware device to measure the performance of cyclic codes that are generated by arbitrary polynomials.

14. SUBJECT TERMS

15. NUMBER OF PAGES 27

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39.18
298-102

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

The error detection performance of shortened and unshortened cyclic codes given that random errors have occurred in the communication channel or storage device has been the subject of many previous studies [1-4]. Particular emphasis has been placed on cyclic redundancy check (CRC) codes with generator polynomials of the form $g(x)=(x+1)p(x)$, $p(x)$ a primitive polynomial [4], since several codes in this class have been accepted as international standards [5]. The probability of undetected error for these CRC codes when used in conjunction with a binary symmetric channel with bit error probability p can be determined for any shortened block length [3,4]. Although the probability of undetected error for the unshortened CRC codes do not depend upon the choice of the primitive polynomial $p(x)$, the same is not true for the shortened codes.

It must be emphasized that P_{ud} is dependent upon the statistical behavior of the errors to be detected by the CRC code. In this report we begin with the assumption that we are communicating over a binary symmetric channel (BSC) where the probability of error for each binary digit (bit) is p ($0 \le p \le 1/2$) regardless of whether the bit is a 1 or a 0, and that the errors in different bits are statistically independent of one another. The BSC is a useful model which describes some commonly used communication channels such as antipodal signalling in an additive white Gaussian noise channel with an optimum receiver. Even correlated, symmetric errors approach the BSC model after interleaving at a sufficient depth. However, there are many situations where CRC codes are expected to detect errors which do not obey the BSC model. One such example is when a CRC code is used to detect errors that were miscorrected by a convolutional ECC. In this case, the ECC decoder will produce burst errors (without interleaving). In sections V and VI we will show how the results obtained over a BSC are also applicable to the single burst error channel.

CRC codes may be used at their natural unshortened block length of $N'=2^{R-1} - 1$, or they may be shortened to an arbitrary, reduced block length N. When unshortened, all CRC codes (with generator polynomials of the form described above) with the same values of N' and K perform identically; for any value of p, codes generated by different polynomials will all have the same $P_{ud}(N',p)$. In addition, $P_{ud}(N',p)$ for unshortened codes will not exceed 2^{-R} for $0 \le p \le 0.5$ regardless of the choice of $p(x)$. In contrast, the performance of shortened CRC codes is dependent upon both the choice of $p(x)$ and the shortened block length N. Furthermore, P_{ud} may exceed 2^{-R} for some values of p in the range $0 \le p \le 0.5$.

II. CALCULATING P_{ud}

Exact evaluations of P_{ud} for linear codes are well known [1,2,3,4,7], and the formulas are repeated here without proof. An undetectable error occurs only when the error pattern is a non-zero code word. Hence, the probability of undetected error is the probability that an N-tuple (error pattern) will have the same portion of ones as a code word from the set of all 2^K code words generated by g(x). If A_i is the number of code words generated by g(x) with a Hamming weight of i, then for independent errors,

$$P_{ud}(N,p) = \sum_{i=1}^{N} A_i\, p^i (1-p)^{N-i}$$

,where p=the channel bit error probability.

The set $\{A_i\}$ is known as the weight distribution of the code generated by g(x)=(x+1)p(x), and for unshortened CRC codes with R parity bits, $\{A_i\}$ is independent of the choice of p(x). However, for shortened codes with R parity bits, $\{A_i\}$ is dependent on both p(x) and N.

P_{ud} can also be calculated using the weight distribution of the dual code generated by the parity check polynomial $h(x) = (x^{N'}-1)/g(x)$. This is generally easier since the dual code is an (N,R) code with a total of 2^R code words and R is usually much less than N. If B_i is the number of code words generated by h(x) with weight i, then for independent errors occurring with probability p,

$$P_{ud}(N,p) = 2^{-R} \sum_{i=0}^{N} B_i (1-2p)^i \; - \; (1-p)^N$$

Finally, it has been shown[4] that CRCs generated by $h(x)=(x^{N'}-1)/g(x)$, where g(x)=(x+1)p(x), have a weight distribution $B_i=w_i + w_{N-i}$, where $\{w_i\}$ is the weight distribution of the code generated by $h_2(x)=(x^{N'}-1)/p(x)$. This code is an (N,R-1) code with a total of 2^{R-1} code words. If w_i is the number of code words generated by $h_2(x)$ with weight i, then for independent errors occurring with probability p,

$$P_{ud}(N,p) = 2^{-R} \sum_{i=0}^{N} (w_i + w_{N-i})(1-2p)^i \; - \; (1-p)^N$$

We are now faced with the task of finding the weight distribution of the 2^{R-1} code words generated by $h_2(x)$. This can be done by generating every code word and tallying the number of code words with weight 0,1,2,3, . . . ,N. A very efficient algorithm exists[4] which requires minimal hardware for implementation. This algorithm utilizes two identical linear feedback shift registers (LFSR "A" and "B") to produce two sequential outputs such that any N consecutive bits of each output may be considered a code word, where $R<N<2^{R-1}$. Because they are identical, both LFSRs will generate the same sequence. However, LFSR "B" is intentionally started N cycles (shifts) after LFSR "A"; hence, "B" lags "A" by N cycles. The time separation of N cycles between the two identical sequences forms a code word "window" of length N in which the output of LFSR "A" represents bits entering the window and the output of LFSR "B" represents bits exiting the window. The algorithm makes use of the fact that if we currently know the weight i of the code word in the window, then when the next code word appears after a single shift of both LFSR "A" and "B", it will have a weight of i, i+1, or i-1, depending upon whether "A" (entering window) equals "B" (exiting window), "A" is greater than "B", or "A" is less than "B", respectively, where "A" and "B" are the outputs of each LFSR and take on the values 0 or 1. See appendix B for a description of linear feedback shift register sequences.

III. SPECIAL HARDWARE TESTER

As discussed earlier, the weight distribution of the code generated by $h_2(x)$ can be used to evaluate P_{ud}. However, there are a total of 2^{R-1} -1 non-zero code words (regardless of N) in an (N,R-1) code, and weight calculations can become time consuming even for reasonable values of R. In addition, the amount of hardware required can be reduced considerably when using the algorithm described earlier.

Because of this, weight calculations are processed in a specialized digital Code Weight Computer (CWC). The CWC is capable of evaluating all w_i for any primitive polynomial of degree 2 through 40, inclusive, and for any block length greater than the degree of p(x) up to a maximum of 65535. A polynomial p(x) along with a block length N is loaded into the CWC via an interface card which is connected to a personal computer (PC). The weight calculations consist of counting the number of ones (weight) in each of all possible non-zero code words. Each of the blocks will have a weight i, where $1 \leq i \leq N$. For each weight, a weight enumerator w_i representing the number of blocks with Hamming weight i is updated until all $2^{R-1} - 1$ non-zero blocks have been processed. This will yield N w_i which may be used to evaluate P_{ud}. When done, an interrupt is sent back to the PC, and the $\{w_i\}$ are read from the CWC. At this point, the remainder of $P_{ud}(N,p)$ calculations are performed in software for varying values of p. The majority of the CWC consists of emitter coupled logic (ECL) which allows a system clock speed of 80MHz. Each of the $2^{R-1} - 1$ blocks requires roughly 2 clock cycles to be processed, allowing a 41-bit (R=41) CRC to be tested in under 7 hours. As an example of the speed improvement of the CWC over a SUN workstation running a non-optimized C program, the CWC required 15 seconds to complete an R=30 weight calculation whereas the SUN required 1.5 hours.

Figure 1 shows the block diagram of the CWC. It is comprised of two linear feedback shift registers (LFSR) that have feedback connections corresponding to the non-zero coefficients of p(x), one 40-bit wide by 64K deep read/write memory (RAM) to store the w_i, one 16-bit "address" counter to access the RAM, one 40-bit "tally" counter to increment the w_i as new blocks of weight i are found, one 40-bit dual-purpose "event" counter to establish the block length and to count the 2^{R-1}-1 non-zero code words to be processed, and lastly, control and interface logic to integrate the subsections.

Both LFSR "A" and "B" have identical feedback connections and initial conditions such that they both generate the same maximal sequence determined by primitive p(x). However, LFSR "B" begins running N cycles (shifts) after LFSR "A" has started, thus "B" lags "A" by N cycles, creating a window of length N. The output from both "A" and "B" are used to determine the weight of the code word in this window as follows:

"A"	"B"	Hamming weight
0	0	same weight as previous code word
0	1	one less than previous code word
1	0	one more than previous code word
1	1	same weight as previous code word

The weight of the current code word is stored in an up/down counter which directly addresses a RAM which contains the weight enumerators $\{w_i\}$ as data. The data is incremented via the tally counter such that each RAM location contains an accumulation of the number of code words having a weight of i. During the first N cycles after startup when "A" is running and "B" is stopped, the RAM address increments from zero up to the weight of the first code word at which time LFSR "B" starts. This creates the window with length N and weight i. For the next $2^{R-1}-2$ cycles, the address counter increments, decrements, or remains stable depending upon whether the current code word has a weight that is larger, smaller, or the same as the previous code word, respectively. Thus the weights of all $2^{R-1}-1$ non-zero code words generated by p(x) are tallied. This along with a single all-zero code word ($w_0=1$) yields the complete weight distribution $\{w_i\}$ for all of the code words generated by p(x).

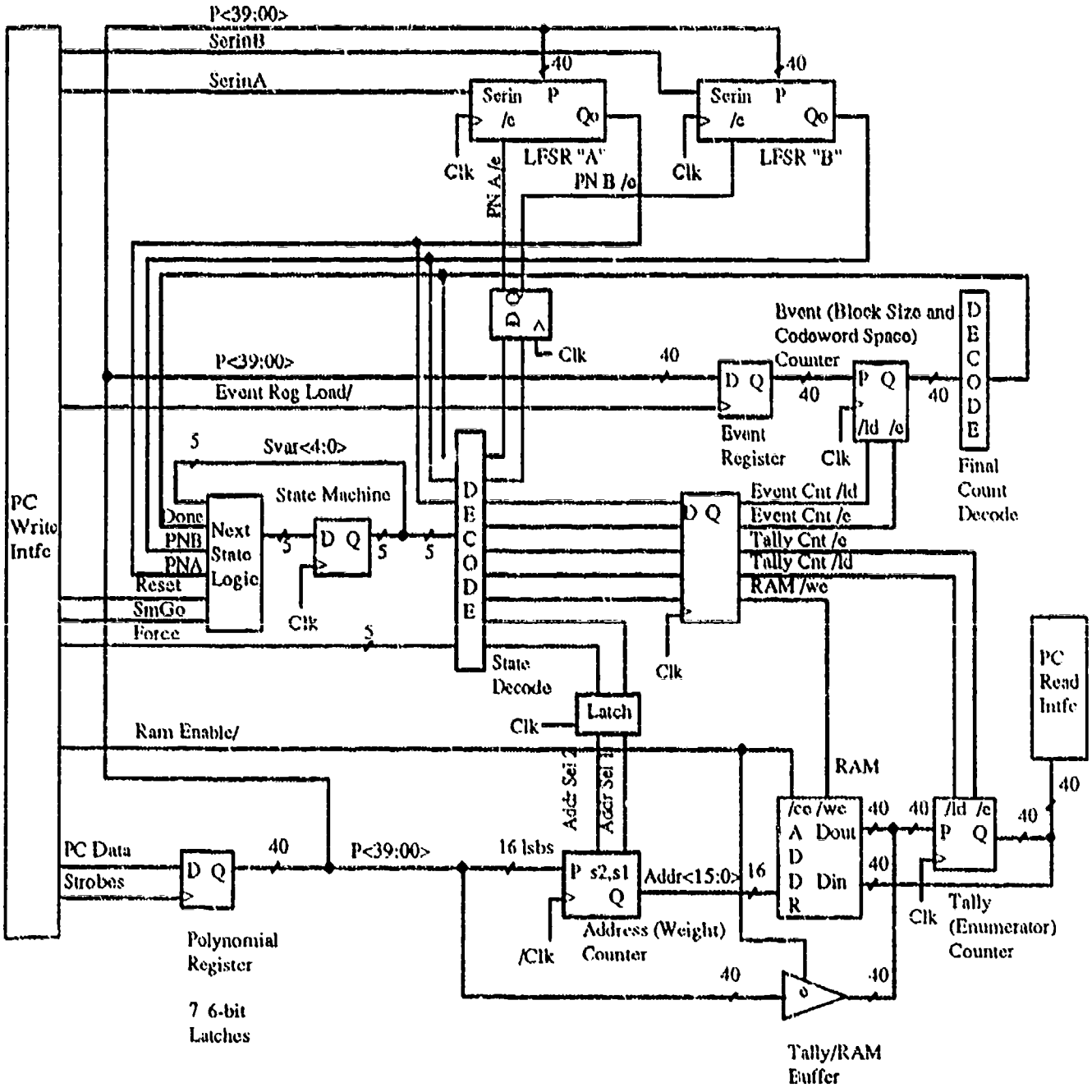


Figure 1. CWC Block Diagram

IV. TEST RESULTS

The objective of the tests performed with the CWC was to determine at least one "good" code having R parity bits in the range $8 \leq R \leq 41$. Our criterion was the ability of the code to be "proper" at a number of block lengths between $(R+1) \leq N \leq 2^{R-2}$. We maintain an earlier[4] definition of "proper" as follows. A code is defined to be proper at block length N if $Pud(N,p) \leq Pud(N,0.5)$ for any $0 \leq p \leq 0.5$. Figures 2 and 3 illustrate the significant difference between proper p(x)=211 and improper p(x)=277. Note that polynomial 277 is proper at N=14.

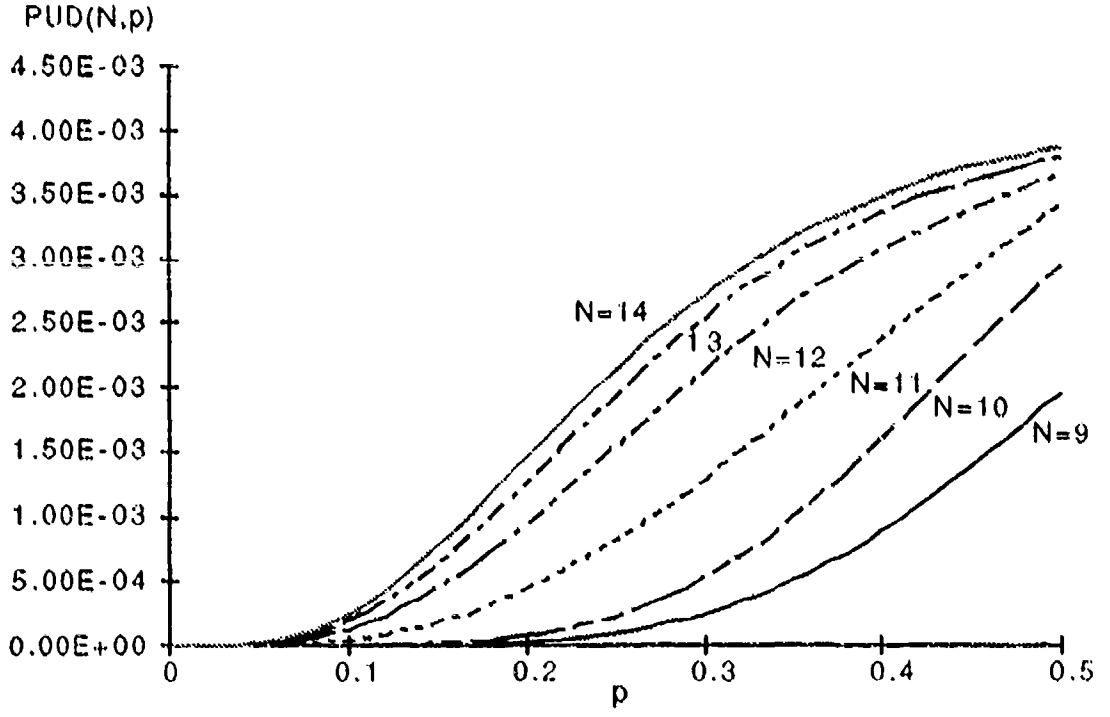


Figure 2. Proper Behavior of Polynomial 211 (octal)

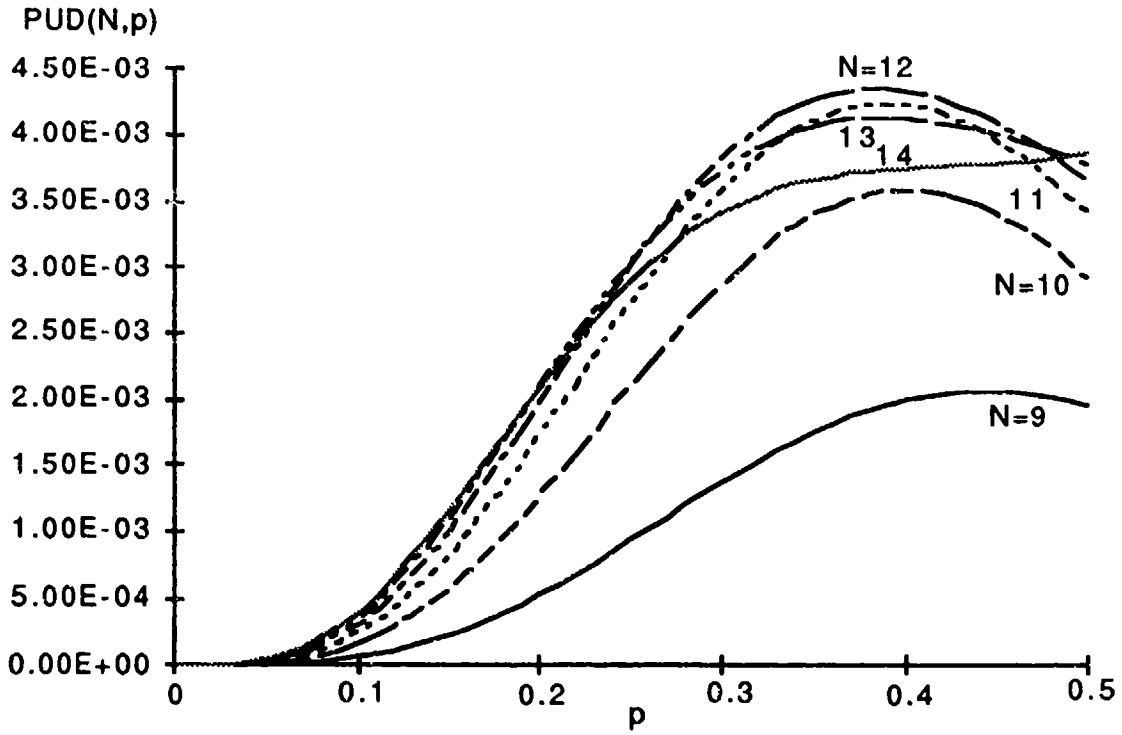


Figure 3. Improper Behavior of Polynomial 277 (octal)

Although the tests did not evaluate p at an infinite number of values, we believe that the chosen values of p are sufficient to eliminate improper codes. Table 1 lists a primitive polynomial for each value of R that has passed our tests. Codes are generated by g(x)=(x+1)p(x).

Table 1.
List of proper CRC primitives (see conditions in text, below)

Parity Bits(R)	Polynomial p(x) (octal) [1]
8	211
9	543
10	1055
11	3471
12	4505
13	15647
14	23231
15	64167
16	103451
17	305667
18	422273
19	1150427
20	2227023
21	6556543
22	10344605
23	27566643
24	42607251
25	134461765
26	345502561
27	426225667
28	1112225171
29	2131556151
30	4660221051
31	16546672375
32	24242142531
33	67346536411
34	114271102221
35	276215750461
36	662342545661
37	1041103456055

The conditions for which the results in table 1 apply are as follows:

Each primitive p(x) was tested at every block length from (R+1) to 2^m, inclusive, where m is the smallest integer such that $2^m > (R+10)$. Next, p(x) was tested at every block length that is a power of two from 2^m up to the lesser of 2^{R-2} or 2^{15}, inclusive. For example, at R=8, block lengths of N={9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,64} were tested.

[1] Octal notation represents the non-zero coefficients of p(x). Thus 211 describes $p(x)=x^7+x^3+1$.

At each block length, $P_{ud}(N,p)$ was evaluated at fifty values of p in the range $10^{-4} \leq p \leq 0.5$. Twenty, linear-scaled values {0.50,0.48,0.46, . . . ,0.16,0.14,0.12} and thirty log-scaled (10 per decade) values $\{0.1,0.1/q,0.1/q^2,0.1/q^3, \ldots,0.1/q^{27},0.1/q^{28},0.1/q^{29},0.0001\}$ were used, where $q=10^{(0.1)}$.

If for all N and p, $P_{ud}(N,p) \leq P_{ud}(N,0.5)$, g(x) was declared "good" and entered into table 1.

The current definition of $P_{ud}(N,p)$ is based on independent, random bit errors occurring with probability p. In the next section, we shall introduce a statistical burst error model and show how the results obtained from the CWC can also be applied to error bursts. Before proceeding, we would like to mention a supplemental, hardware Channel Tester (CT) which is described in appendix A. The CT was originally planned to extend the scope of our tests to include arbitrary cyclic codes operating over real communication channels. Of particular interest was the burst error performance of CRC codes when used in conjunction with error correcting codes. However, the findings of the next two sections extends the capabilities of the CWC to include burst error performance measurement. Therefore, we include the untested CT design only as an appendix to this report.

V. BURST ERROR PERFORMANCE OF BINARY CYCLIC CODES

A statistical model for a single burst of length b is defined whereby the errors are confined to a span of b digits and where the errors within the span occur randomly with bit error probability p. We call such a burst a "(b:p) burst". The conditional probability of undetected error for a shortened or unshortened cyclic code given the occurrence of a single (b:p) burst is shown to be equal to the probability of undetected error for that cyclic code shortened to block length b when this shortened code is used in conjunction with a random error channel with bit error probability p. We investigate the performance of CRC codes with generator polynomials of the form $g(x)=(1+x)p(x)$, $p(x)$ a primitive polynomial. We show that the burst error detection performance of these CRC codes depends upon the choice of the primitive polynomial $p(x)$. In particular, it is shown that 16 bit CRC codes exist which have better burst error detection capabilities than the widely used 16 bit international standards, i.e., the CRC-16 and CRC-CCITT codes. For the random error channel, shortened CRC codes that outperform the commonly used CRC standards (such as the CRC-16 and the CRC-CCITT codes) have been found [3-4].

In this section of the report, we are concerned with the single burst error detection capability of binary, shortened or unshortened cyclic codes. Included in this class are the CRC codes with generator polynomials of the form $g(x)=(x+1)p(x)$.

We consider the following model for a single burst error of burst length b. We assume that errors only occur within a span of b digits, and within that span, the errors occur randomly with bit error probability p ($0 \leq p \leq 1$). We call the above burst, a "(b:p) burst" where b is a positive integer and p is a real number in the range $0 \leq p \leq 1$.

We believe that the (b:p) burst model is useful to describe many real burst error channels. Examples are the errors produced by a decoder of an error correcting code or the errors which are produced by a fading channel.

Note that with this model, if $p < 1$, errors do not necessarily occur at the first and last bit in the burst so that the actual error pattern for a (b:p) burst may span less than b digits. This should be kept in mind when comparing our results with those previously reported in the literature.

Previous results have been published regarding the burst error detection capability of cyclic codes [1-2]. It has long been known that for a cyclic code with arbitrary generator polynomial g(x) of degree R, the fraction of bursts of length b which is undetected by the code is:

0	if $b \leq R$,
$2^{-(R-1)}$	if $b = R+1$, and
2^{-R}	if $b > R+1$.

Here, the definition of a burst of length b differs from our (b:p) model in several ways. For one, with this definition, there is always an error in the first and last (that is, the b-th) digit of the burst. Furthermore, this definition is "non-statistical" in that it refers to fraction of bursts that are not detected rather than the probability of a burst not being detected. Using this definition, the fraction of undetected errors for a burst of length b would equal the conditional probability of undetected error given a burst of length b provided that the 2^{b-2} burst error patterns that have an error in the first and b-th positions are all equally probable. It is our belief that this assignment of probabilities does not describe accurately many real communications or storage channels and that the assignment used in our (b:p) model is a much better model for actual burst errors. We refer to this older model of a burst as the "non-statistical" model.

Let $P_{ud}(p|b))$ be the conditional probability of undetected error for an error detection code given that the code experiences a (b:p) burst. In many applications it may be difficult to know an exact value to assign to p but rather we may know what limits to assign to a range of values: $p_{min} \leq p \leq p_{max}$. We propose to choose that value of p in the range $p_{min} \leq p \leq p_{max}$, for which $P_{ud}(p|b)$ is a maximum. We call this worst case value of p, p*, and we denote the corresponding worst case conditional probability of undetected error for a burst of length b, $P_{ud}(p^*|b)$. Thus:

$$P_{ud}(p^*|b) = \max \left[P_{ud}(p|b)\right], \quad p_{min} \leq p \leq p_{max}$$

The question as to the choice of p_{min} and p_{max} depends on the source of the burst error in the application. It is reasonable, in most applications to set $p_{min} = 0$ and $p_{max}=0.5$ but there are situations where one might consider $p_{max} > 0.5$ (e.g., a phase slip in a P.S.K. demodulator). In this section of the report, unless it is specifically stated to the contrary, we assume that $p_{min} = 0$ and $p_{max} = 0.5$.

For $p_{min} = 0$ and $p_{max} = 0.5$, one would think that $p^*=0.5$ for all codes, but for many codes, including several commonly used international standards, this is not the case. For codes where $p^*=0.5$, the results predicted by the (b:p) burst model give results similar to those predicted by the older burst model. In particular, $P_{ud}(p^*|b) = 0$ for $R \geq b$ and $P_{ud}(0.5|b) = 2^{-R}-2^{-b}$ for $R < b$. However, as we shall see, when p^* is not equal to 0.5, the results predicted by the (b:p) model can differ significantly from those predicted by the older model.

A somewhat surprising result occurs for the case of $p_{max} > 0.5$. In this case, as we shall prove later, for almost every code, if $P_{ud}(p|b) \leq P_{ud}(0.5|b)$, for all $p \leq 0.5$, then $P_{ud}(p|b) > P_{ud}(0.5|b)$ for some $p > 0.5$. Said more simply, for $R<b$, $P_{ud}(0.5|b) = 2^{-R}-2^{-b}$ is almost never an upper bound for $P_{ud}(p|b)$ if $p_{max} > 0.5$.

An important consequence of the (older) non-statisitical definition of a burst is that the fraction of undetected errors of burst length b does not depend upon the specific choice of the generator polynomial of the code but only on the number of parity digits in the code. In particular, for CRC codes with generator polynomials of the form $g(x)=(x+1)p(x)$, $p(x)$ a primitive polynomial, this definition implies that the burst detection capability of these codes does not depend upon the choice of the primitive polynomial $p(x)$. As we shall see, this is not the case using our definition of a single (b:p) burst if p^* is not equal to one half. That is, using our definition, the worst case conditional probability of undetected error for a (b:p) burst for two codes having the same number of parity digits but differing in the choice of their generator polynomial g(x), can differ significantly. In particular we will find CRC codes that outperform the international standards with respect to their ability to detect single (b:p) bursts.

In the next section we prove that for any (unshortened or shortened) cyclic code, the conditional probability of undetected error given a (b:p) burst is equal to the probability of undetected error for that code when used with a random error channel with bit error probability p when the code is shortened to block length b. We then examine some consequences of this result. In particular, we explore the burst error performance of CRC codes with generator polynomials of the form $g(x)=(x+1)p(x)$, with $p(x)$ a primitive polynomial.

VI. RESULTS

Consider any shortened or unshortened cyclic binary (N,K) code (with R = N-K). Define $P_{ud}(N,p)$ as the probability of undetected error for the code when the code is used in conjunction with a binary symmetric channel with channel bit error probability p. Furthermore, define $P_{ud}(p|b)$ as the conditional probability of undetected error for the same code given a (b;p) burst. Then, the following theorem states that for all (N-K) < b ≤ N, the conditional probability of undetected error for the (N,K) code given that a (b:p) burst occurred is equal to the probability of undetected error for the same code, shortened to block length b, when this shortened code is used with a binary symmetric channel with channel bit error probability p. That is, we have the

Theorem: $P_{ud}(p|b) = P_{ud}(b,p)$.

Proof: Since the code is a linear (N,K) code we can assume in the calculation of the conditional probability of undetected error that the all zero code word was transmitted. First assume that a (b:p) burst occurred as the error pattern. This error pattern is such that it has all of its components identically zero except for a span of length b starting in position i and ending in position i+(b-1). Within this span, the coefficients of the error pattern are i.i.d. binary random variables with the probability of a 1 equal to p. We are now concerned with the probability that this error pattern, when input to a linear feedback shift register (LFSR) with feedback connections set in accordance with the coefficients of the generator polynomial g(x), will result in the final contents of this shift register being all zeros. Note that the coefficients of the error vector outside of the range (i,i+b-1) are all zero so that the leading all zero coefficients when input to the LFSR will leave it in the all zero state. Then, the coefficients corresponding to the burst in the interval (i,i+b-1) enter the LFSR and either leave it in the all zero state or not. If the LFSR is left in the all zero state the trailing zeros entering the LFSR will not alter this condition. If the LFSR is left in any other state other than the all zero state, it will not return to the all zero state as a result of the trailing zeros. Thus, the situation is exactly the same as if the code was shortened to block length b, and was used in conjunction with a binary symmetric channel with channel bit error probability p. Q.E.D.

By a careful reading of the above proof, we find that the following corollary is true:

Corollary: All shortened codes obtained from a given cyclic code have the same $P_{ud}(p|b)$ (and thus the same $P_{ud}(p^*|b)$) provided that b is less than or equal to the shortened block length.

These results are interesting in their own right, but they are particularly significant since $P_{ud}(N,p)$ (and thus $P_{ud}(p|b)$) has been computed for a wide variety of codes. The consequences of this computation are described in the next section for the class of CRC codes with generator polynomial $g(x) = (x+1)p(x)$, $p(x)$ a primitive polynomial [4-5]. Using the hardware tester described earlier, $P_{ud}(p^*|b)$ can be computed for any CRC code in this class where $p(x)$ is any primitive polynomial of degree 40 or less.

In the next section, we will always assume that the range of p of interest is $0 \leq p \leq 0.5$. Before proceeding to this section, we explore the consequences of allowing p_{max} to be strictly greater than 0.5. If we compute the derivative of $P_{ud}(b,p)$ (and thus, $P_{ud}(p|b)$) with respect to p and set p=0.5, we obtain $-2^{-(R-1)}B_1 + b2^{-(b-1)}$, where B_1 is the number of code words of Hamming weight 1 in the dual code when the code is shortened to block length b. Setting this derivative to zero and solving for B_1 we obtain the equation $B_1=b2^{R-b}$. Except for the hypothetical case where $B_1=b2^{R-b}$, we have a non-zero derivative at p=0.5. This implies that except for this hypothetical case, the maximum value of $P_{ud}(p|b)$ cannot occur at p=0.5. It should be realized that the same argument can be used to show that, except for this hypothetical case, the maximum value of $P_{ud}(N,p)$ does not occur for the binary symmetric channel at p=0.5. This is true even for the so-called proper codes [2-4] where $P_{ud}(N,p) \leq P_{ud}(N,0.5)$ for all $p \leq 0.5$.

VII. GOOD CRC CODES FOR BURST DETECTION

We first focus on the specific case of CRC codes with 16 parity check bits. Using the older definition of a burst where all 2^{b-2} bursts of length b are equally likely we find that all such codes have the same conditional probability of undetected error. The results are presented in the following table:

Table 2

Conditional Probability of Undetected Error for 16 Bit CRC Codes
Assuming All 2^{b-2} Bursts of Length b Are Equally Likely

Burst Length	Conditional Probability of Undetected Error
$b \leq 16$	0
b=17	$2^{-15} = 3.1 \times 10^{-5}$
$b \geq 17$	$2^{-16} = 1.5 \times 10^{-5}$

We next consider the performance of 16 bit CRC codes with generator polynomials of the form $g(x) = (1+x)p(x)$, where $p(x)$ is a primitive polynomial (of degree 15). In particular, we consider the performance of these codes for a (b:p) burst. We consider three codes, two of which are the commonly used international standards (CRC-CCITT and CRC-16) and a third code which is denoted CRC-16Q*. (This is not the CRC-16Q code referred to in [4].) The generator polynomials of these codes are as follows:

CRC-16 $g(x)=(x+1)(x^{15}+x+1)=x^{16}+x^{15}+x^2+1,$

CRC-CCITT $g(x)=(x+1)(x^{15}+x^{14}+x^{13}+x^{12}+x^4+x^3+x^2+x+1)$

$=x^{16}+x^{12}+x^5+1,$

CRC-16Q* $g(x)=(x+1)(x^{15}+x^{10}+x^9+x^8+x^5+x^3+1)$

$=x^{16}+x^{15}+x^{11}+x^8+x^6+x^5+x^4+x^3+x+1.$

For $p_{max} = 1/2$, the maximum conditional probability of undetected error, $P_{ud}(p^*|b)$, for these three codes is shown below in Table 3. The CRC 16Q* code has $p^*=0.5$ for all burst lengths. It should be remembered that p_{max} is assumed to be 0.5. If p_{max} were taken to be greater than 0.5 than p^* would not equal 0.5 and $P_{ud}(p^*|b)$ would be somewhat higher than the values given in this table.

Table 3

Maximum Conditional Probability of Undetected, $P_{ud}(p^*|b)$, Error for Three 16-Bit CRC Codes

b	CRC-16	CCITT	CRC-16Q*
≤ 16	0	0	0
17	9.3×10^{-5}	9.3×10^{-5}	7.6×10^{-6}
18	1.5×10^{-4}	1.5×10^{-4}	1.1×10^{-5}
19	1.8×10^{-4}	1.7×10^{-4}	1.3×10^{-5}
20	2.0×10^{-4}	1.8×10^{-4}	1.4×10^{-5}
21	2.1×10^{-4}	1.8×10^{-4}	1.5×10^{-5}
22	2.1×10^{-4}	1.8×10^{-4}	1.5×10^{-5}
23	2.0×10^{-4}	1.7×10^{-4}	1.5×10^{-5}
24	1.9×10^{-4}	1.7×10^{-4}	1.5×10^{-5}
25	1.8×10^{-4}	1.6×10^{-4}	1.5×10^{-5}
26	1.7×10^{-4}	1.5×10^{-4}	1.5×10^{-5}
27	1.6×10^{-4}	1.4×10^{-4}	1.5×10^{-5}
28	1.5×10^{-4}	1.3×10^{-4}	1.5×10^{-5}
29	1.4×10^{-4}	1.2×10^{-4}	1.5×10^{-5}
30	1.4×10^{-4}	1.1×10^{-4}	1.5×10^{-5}
31	1.5×10^{-4}	1.1×10^{-4}	1.5×10^{-5}
32	1.6×10^{-4}	1.0×10^{-4}	1.5×10^{-5}

It should be noted that the CRC-16Q* code has almost an order of magnitude improvement in the maximum conditional probability of undetected error, $P_{ud}(p^*|b)$, over the two international standards for all burst lengths in the range $17 \leq b \leq 32$. The old definition of burst error performance which compares the percentage of burst error patterns of length b which are not detected by the code (as given in Table 1) predicts that all three codes have identical performance.

Figure 4 is a plot of $P_{ud}(p|b)$ versus p for the three codes CRC-16, CRC-CCITT and CRC-16Q* for b=20 for p taking values in the range $0 \leq p \leq 1$. Many peculiar phenomena are apparent from these curves. Although, the two international standards have similar behavior for p in the range $0 \leq p \leq 0.5$, they have markedly different behaviors for p in the range $0.5 \leq p \leq 1$. Note that the CRC-CCITT code has two distinct local maxima, one for p below 0.5 and one for p above 0.5. Also, although the CRC-16Q* code has its maximum for p above 0.5, the value at its maximum is not very different from its value at p=0.5.

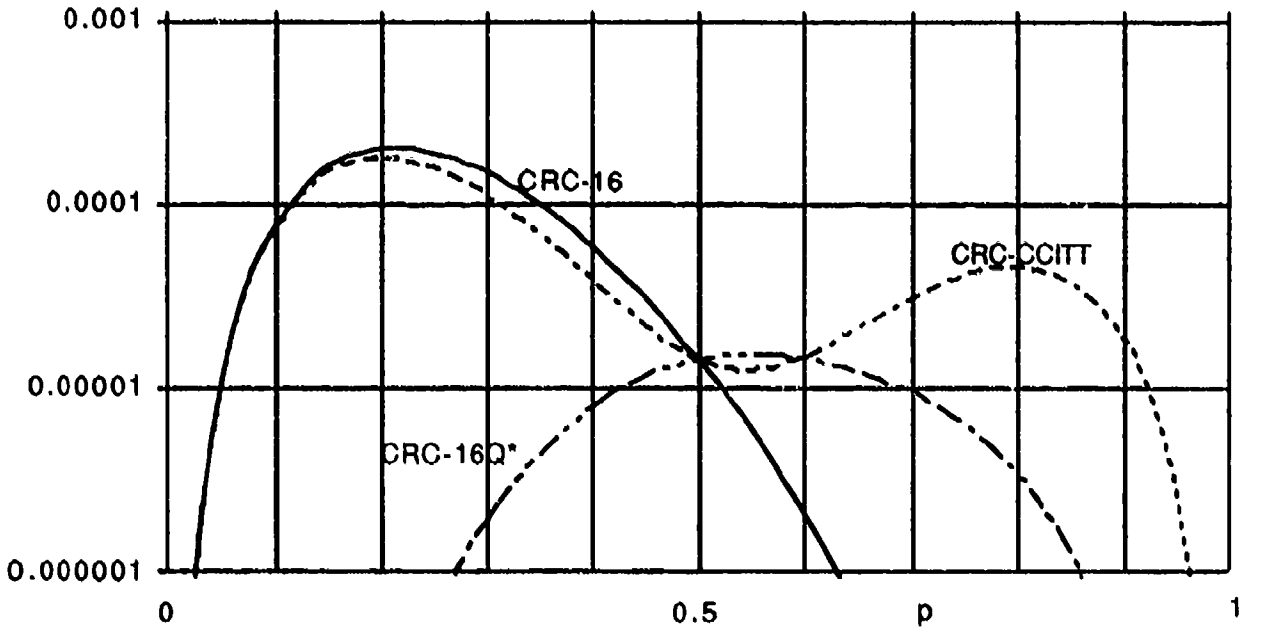


Figure 4. $P_{ud}(p|b=20)$ for CRC-16, CRC-CCITT and CRC-16Q*.

There are many CRC codes with generator polynomials of the form (1+x) times a 15th degree primitive polynomial that have the burst error performance superior to the international standards. The following discussion shows why the generator polynomials chosen for the international standards were poor choices for burst error detection and how other choices result in better codes.

Consider first that we want to detect a burst of length b=R+1. The burst is not detectable if and only if the polynomial which represents this burst is divisible by the generator polynomial of the code. But at this burst length there is only one such burst pattern: namely, the generator polynomial itself. Assume that the generator polynomial has exactly d_1 ones (and R-d_1 zeros): that is, assume that the generator polynomial has Hamming weight equal to d_1. Then, the conditional probability that a (b:p) burst with b=R+1 is not detectable is given by the formula $P_{ud}(p|R+1) = p^{d1}(1-p)^{(b-d1)}$. This equation has its maximum at $p=d_1/b$. If $d_1/b>0.5$, then the maximum value of $P_{ud}(p|R+1)$ for $0 \leq p \leq 0.5$ occurs at p=0.5. If, however, $d_1/b < 0.5$, then the maximum value of $P_{ud}(p|R+1)$ is $P_{ud}(d_1/b|R+1) = (d_1/b)^{d1}(1-(d_1/b))^{(b-d1)}$. For the three codes discussed previously, the values of d_1 and d_1/b are listed in table 4.

Table 4
Values of d_1 and d_1/b for CRC-16, CRC-CCITT, and CRC-16Q*
for an error burst of length R+1

Code	d_1	d_1/b
CRC-16	4	4/17
CRC-CCITT	4	4/17
CRC-16Q*	10	10/17

Even if p_{max} were taken to equal 1 , the CRC-16Q* code would have a smaller probability of undetected error than the two international standards. This is the case since $(d_1/b)^{d1}(1-(d_1/b))^{(b-d1)}$ is smaller for $d_1/b=10/17$ than for $d_1/b=4/17$.

The discussion in the previous paragraph might suggest that the only parameter of importance in choosing the generator polynomial is its Hamming weight. This is not the case, as can be seen by considering longer error bursts. For error bursts of length R+2, one must consider d_2, the Hamming weight of (x+1) times the generator polynomial. Then the conditional probability of undetected error is: $P_{ud}(p|R+2)=2p^{d1}(1-p)^{(b-d1)}+p^{d2}(1-p)^{(b-d2)}$. For error bursts of length R+3 one must consider d_3 and d_4, the Hamming weights of (x^2+1) times the generator polynomial and (x^2+x+1) times the generator polynomial, respectively. The conditional probability of undetected error in this case is given as: $P_{ud}(p|R+3)=3p^{d1}(1-p)^{(b-d1)}+2p^{d2}(1-p)^{(b-d2)}+p^{d3}(1-p)^{(b-d3)}+p^{d4}(1-p)^{(b-d4)}$.

The values of d_1, d_2, d_3, and d_4, for the three 16 bit CRC codes discussed previously are given in Table 5 as well as the maximum values of $P_{ud}(p|17)$, $P_{ud}(p|18)$, $P_{ud}(p|19)$, and $P_{ud}(p|20)$ for these codes for p in the range $0\leq p\leq 0.5$ and $0\leq p\leq 1.0$. The values of d_1, d_2, d_3, and d_4 explain why the primitive polynomial of degree 15 used in the generator polynomial of the CRC-16Q* code is better than the primitive polynomials of degree 15 used in the generator polynomials of the two international standards. In particular, it can be shown that the values of d_1, d_2, d_3, and d_4, for the CRC-16Q* code are near optimal for minimizing $P_{ud}(p|17)$, $P_{ud}(p|18)$, and $P_{ud}(p|19)$ over the entire range $0\leq p\leq 1$.

Table 5

Parameters Related to the Burst Error Detection Capability of Three 16-Bit CRC Codes

Parameter	CRC-16	CRC-CCITT	CRC-16Q*	Range
d_1	4	4	10	
d_2	6	8	10	
d_3	6	8	12	
d_4	6	12	10	
max $P_{ud}(p\|17)$	9.4×10^{-5}	9.4×10^{-5}	7.6×10^{-6}	$0<p<0.5$
max $P_{ud}(p\|17)$	9.4×10^{-5}	9.4×10^{-5}	1.0×10^{-5}	$0<p<1$
max $P_{ud}(p\|18)$	1.5×10^{-4}	1.5×10^{-4}	1.1×10^{-6}	$0<p<0.5$
max $P_{ud}(p\|18)$	1.5×10^{-4}	1.5×10^{-4}	1.3×10^{-5}	$0<p<1$
max $P_{ud}(p\|19)$	1.8×10^{-4}	1.7×10^{-4}	1.3×10^{-5}	$0<p<0.5$
max $P_{ud}(p\|19)$	1.8×10^{-4}	1.7×10^{-4}	1.4×10^{-5}	$0<p<1$

The improvements obtained in the conditional probability of undetected burst error are much more dramatic when one uses more parity bits in the CRC code. For example, consider two CRC codes with 32 parity bits obtained from the following two generator polynomials, both of which are of the form (1+x) times a primitive irreducible polynomial of degree 31:

$$\text{CRC-32} \quad g(x) = (x+1)(x^{31}+x^3+x^2+x+1)$$
$$= (x^{32}+x^{31}+x^4+1)$$
$$\text{CRC-32Q*} \quad g(x) = (x+1)(x^{31}+x^{23}+x^{22}+x^{15}+x^{14}+x^7+x^4+x^3+1)$$
$$= (x^{32}+x^{31}+x^{24}+x^{22}+x^{16}+x^{14}+x^8+x^7+x^5+x^3+x+1).$$

The parameter d_1 for these codes is 4 and 12 respectively. This is the parameter which describes the conditional burst error detecting capability of the code for a burst of length 33. The resulting values for $P_{ud}(p^*|33)$ are: for the CRC-32 code, $P_{ud}(p^*|33)=5.1 \times 10^{-6}$, and for the CRC-32Q*, $P_{ud}(p^*|33)=4.0 \times 10^{-10}$.

VIII. CONCLUSION

A fast and efficient hardware device for determining the weight distribution of certain CRC codes is currently in operation. The weight distribution of codes generated by $g(x)=(x+1)p(x)$ where $p(x)$ is a primitive can be determined for any R up to 41 and any N up to 65535. From this, the exact probability of undetected error when communicating over a random channel can be evaluated.

A new burst error model is proposed for evaluating the burst error detection performance of CRC codes. Based on this model, we find that codes with the same number of parity digits can have very different burst error detecting performance. In particular, a 16 bit CRC code that has an order of magnitude improvement in burst detecting capability over the CRC-CCITT and CRC-16 international standards is given. With this model, evaluating the burst performance of CRCs generated by $g(x)=(x+1)p(x)$ is fast and efficient using the hardware device above.

Insight as to what characteristics of the generator polynomial corresponds with good burst performance is also provided.

ACKNOWLEDGMENT

CRC Research and hardware development was funded by the U.S. Air Force under contract F49650-90-C-0017. The CWC system design was introduced by Lyle Frederickson, Dr. Andrew Viterbi, and Dr. Jack Wolf. The hardware design, simulation, test, and measurement was completed by Rob Blakeney, Jeff Levin, and Dexter Chun.

APPENDIX A : Hardware Device for the Evaluation of Arbitrary Cyclic Codes

Results from the Code Weight Computer (CWC) only apply to CRCs with generator polynomials of the form $g(x) = (x+1)p(x)$ when used over a random or single burst error channel at block lengths up to 65535, where $p(x)$ is a primitive polynomial of degree up to 40.

To allow an even wider search and certification range, a hardware Channel Tester (CT) has been studied to measure the performance of cyclic codes created by an arbitrary generator polynomial $g(x)$ up to degree $R=64$ and at block lengths up to $N=2^{24}$. The CT would be capable of counting the number of undetected errors that occur over existing (real) communication channels or through a channel simulator which would emulate both random and multiple burst channels. The relative frequency of undetected errors could then be used as an estimate of the probability of undetected error (P_{ud}) over the channel. Additional measurements would include channel bit error rate, channel delay, and error burst length within a code word. In conjunction with a personal computer (PC), the CT could provide stand-alone capability including pseudo-random data generation, parity encoding/decoding, receive self-synchronization, and error counting.

Any polynomial up to degree 64 may be used to encode a programmable pseudo-random data source, generating codewords with block lengths of up to 2^{24} bits ($N \leq 16777216$, $N-K \leq 64$). Measurements performed will consist of a) the total number of codewords transmitted, b) the total number of codewords received with errors (including both detected and undetected errors), and c) the total number of codewords received with detected (parity) errors. From these measurements, the probability of block error, of detected block error, and of undetected block error can be computed. In addition, the channel tester will be capable of measuring the delay through the channel, the number of bit errors within an undetected error block, or the longest error burst (number of bits between first and last error in a block) within an undetected error block (the terms codeword and block are used interchangeably).

Figure 5 shows a block diagram of the CT. The process of measuring the probability of undetected block error shall consist of the following procedure:

1. Load and initialize all Linear Feedback Shift Registers (LFSRs), counters, and control logic.
2. Start transmission of preamble PN sequence immediately followed by cyclic encoded PN data.
3. On receive side, acquire synchronization with preamble then switch to error detection of the cyclic encoded PN data.
4. Continuously transmit and receive the PN data, measuring the number of block errors, detected block errors, and blocks sent.
5. Prior to overflow of the block counter (number of blocks sent), the CT shall pause, allowing the Personal Computer (PC) to upload the measurements.
6. After uploading the various counter contents, the PC shall zero the measurement counters then signal the CT to resume processing from item 4, above.

The process of measuring the channel delay and error statistics (bit error rate or longest burst) shall consist of the following procedure:

1. Measure the delay from the start of transmission (beginning of first codeword sent) to the start of reception (beginning of first codeword received).
2. Pause, to allow PC to read delay measurement.
3. Continue, counting the number of bit errors in a block, or the longest burst of errors in a block, depending upon the mode of operation. At the end of each block, pause if an undetected error has occurred, else flush counters and repeat.

The CT will be comprised of a transmit encoder, a receive decoder, and an error recorder as follows: A 24-bit programmable transmit data LFSR will generate a pseudo-random noise (PN) sequence that will serve as the information to be cyclic encoded. This PN sequence will also serve as a preamble to allow a 24-bit receive data synchroni[illegible]/LFSR (programmed with the same PN generator polynomial) to acquire bit and block synchronization. In search mode, the synchronizer will function as a self synchronizing descrambler with shift register state detection to permit code word (block) synchronization with the transmitter. In data mode, the synchronizer will operate as an LFSR, generating an identical, local PN which shall be matched against the received PN to detect data errors. Encoding will be performed by a 64-bit transmit parity LFSR which will be used to generate the parity bits for each code word and may be programmed with any code generator polynomial g(x) up to degree 64. Similarly, a 64-bit receive parity LFSR shall be programmed with the same g(x) and will perform the cyclic decoding - detecting errors in the received blocks. To establish the timing for the (N,K) codewords, programmable 24-bit block size counters shall be used separately in transmit and receive, where N and K are up to 24-bits each.

To keep track of the statistics, three multi-purpose counters will be used; a block counter, block error counter, and detected block error counter will be updated with every codeword received, and shall be read from the PC whenever the CT pauses. The block counter shall be programmable and will allow up to 2^{24} blocks to be tested before pausing the CT. In addition, the block counter will function as either a bit error or error burst counter - measuring the total number of bit errors or the number of bits between the first and last error in each block, respectively. A 16-bit block error counter will accumulate the number of code words that have any errors, and a 16-bit detected block error counter will accumulate the number of code words that have parity (detected) errors. Thus the frequency of undetected errors occurring over the channel can be calculated by taking the difference between the block error count and the detected block error count and then dividing this result by the block count. The detected block error counter will also be used to measure the delay through the channel. Finally, state machine and control logic will integrate the various sections of the CT throughout the synchronization and measurement process.

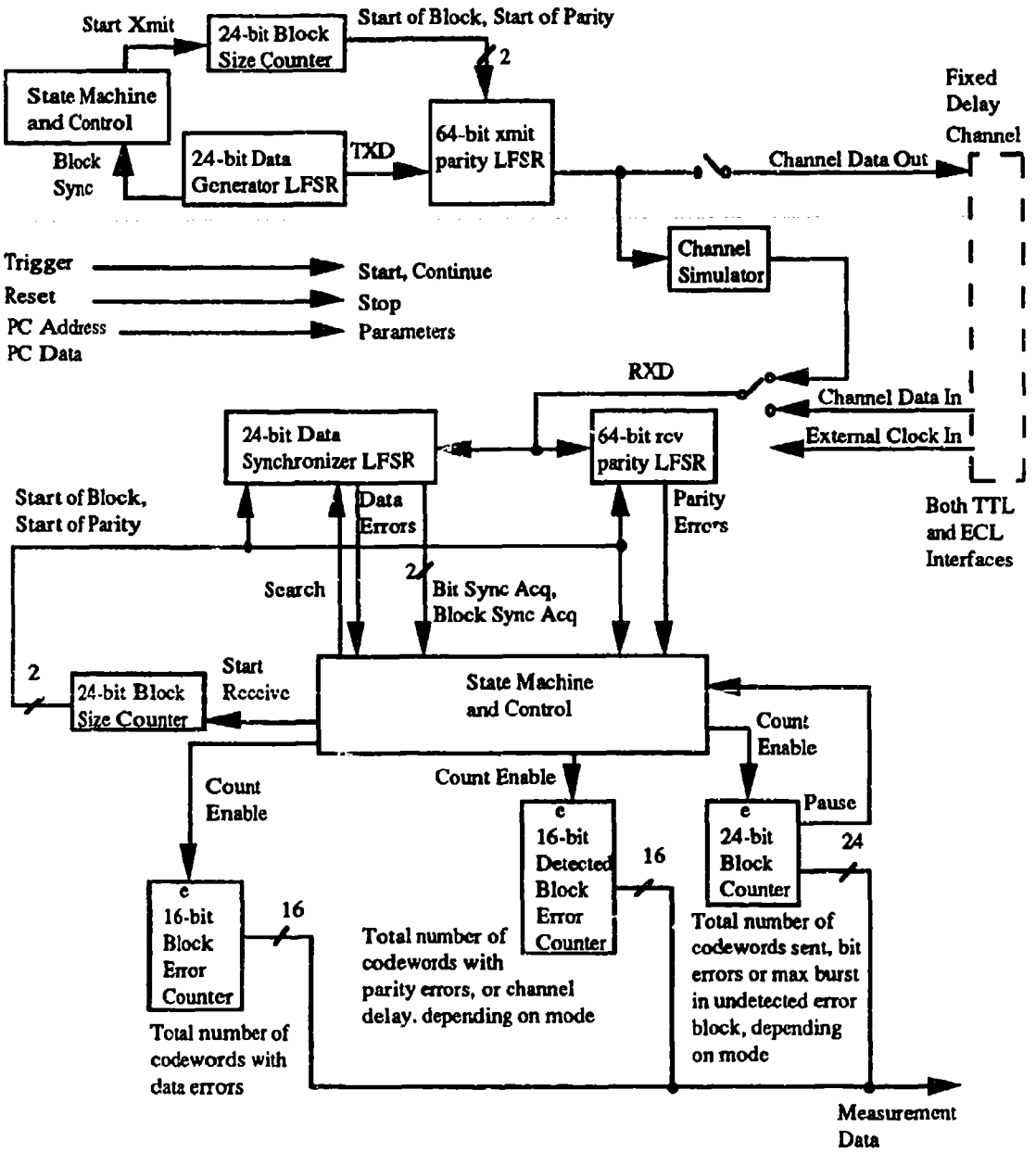
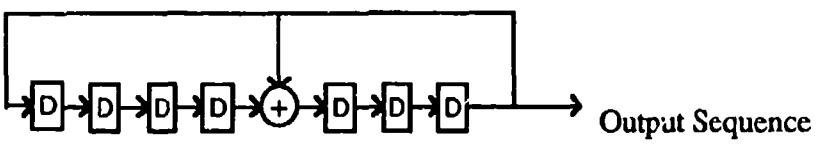


Figure 5. Channel Tester Block Diagram

APPENDIX B : Shift Register Sequences and Code Polynomials

A linear feedback shift register (LFSR) with feedback connections corresponding to the non-zero coefficients of a primitive polynomial p(x) and initialized to any non-zero state will produce a maximal sequence containing all 2^{R-1} -1 non-zero code words generated by $h_2(x)=(x^{N'}-1)/p(x)$ before repeating[4]. When initialized to the all zero state, the LFSR will produce the all zero code word. This accounts for all 2^{R-1} code words generated by $h_2(x)$. The block length N is simply the number of consecutive bits in the sequence that are observed when measuring the weights. This can be visualized as a window of length N panning across the LFSR output sequence one cycle at a time. For an unshortened code, the window will view 2^{R-1} -1 bits at a time for a total of 2^{R-1} -1 unique non-zero code words. For shortened codes, the window will view fewer bits at a time, but will still look at all 2^{R-1} -1 non-zero code words. See figure 6 for an example of an LFSR described by polynomial 211 (octal).

Figure 6. LFSR for polynomial 211.



D = 1 shift delay

+ = modulo-2 adder (exclusive-or)

REFERENCES

1. W. W. Peterson, Error-Correcting Codes, M.I.T. Press, Cambridge, MA, 1961.

2. J. K. Wolf, A. Michelson, and A. Levesque, "On the Probability of Undetected Error for Linear Block Codes", IEEE Trans. Comm., vol. COM-30, pp. 317-324, Feb. 1982.

3. T. Fujiwara, T. Kasami, A. Kitai, and S. Lin, "On the Undetected Error Probability for Shortened Hamming Codes", IEEE Trans. Comm., vol. COM-33, pp. 570-574, June 1985.

4. J. K. Wolf and R.D. Blakeney, II, "An Exact Evaluation of the Probability of Undetected Error for Certain Shortened Binary CRC Codes", MILCOM '88, pp. 15.2.1-15.2.6,

5. A. S. Tanenbaum, Computer Networks, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1981, pg. 132.

[6] J. K. Wolf and D. Chun, "The Single Burst Error Detection Performance of Binary Cyclic Codes", submitted for publication.

7. S. Lin and D. Costello, Error Control Coding Fundamentals and Applications, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1983, pp 85 - 120.

[8] D. Chun and J. K. Wolf, "Special Hardware for Computing the Probability of Undetected Error for Certain Binary CRC Codes and Test Results", submitted for publication.