

AD-A070 673

MITRE CORP BEDFORD MASS

F/G 9/2

GENERALIZED CYCLIC CODES FINITE FIELD ARITHMETIC. VOLUME III.(U)

MAY 79 T F ROOME

F19628-78-C-0001

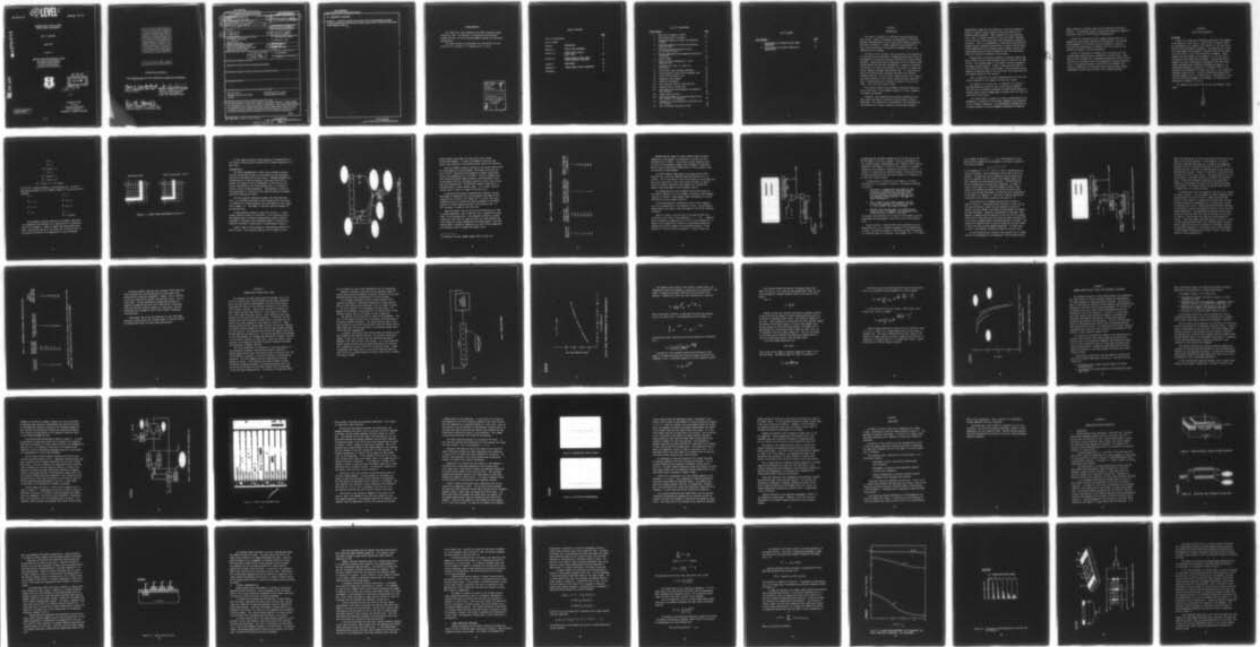
UNCLASSIFIED

MTR-3227-VOL-3

ESD-TR-79-124-VOL-3

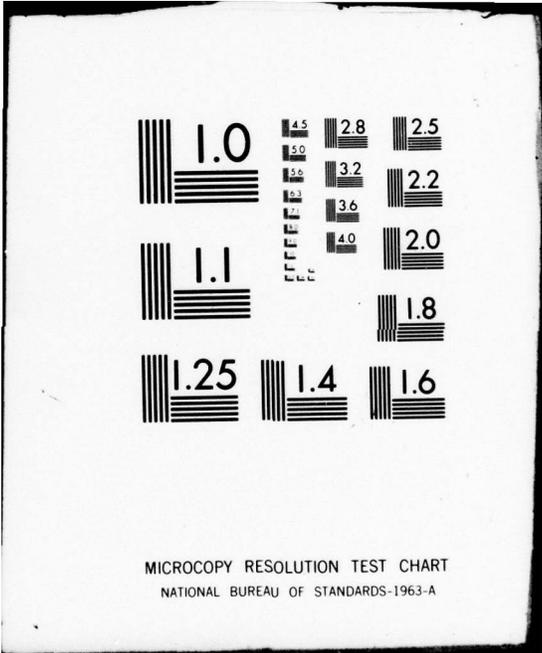
NL

| OF |  
ADA  
070673



END  
DATE  
FILMED  
8-79

DDC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

12 LEVEL II

GENERALIZED CYCLIC CODES  
FINITE FIELD ARITHMETIC

BY T. F. ROOME

MAY 1979

Prepared for

DEPUTY FOR DEVELOPMENT PLANS  
ELECTRONIC SYSTEMS DIVISION  
AIR FORCE SYSTEMS COMMAND  
UNITED STATES AIR FORCE  
Hanscom Air Force Base, Massachusetts



DDC  
RECEIVED  
JUN 29 1979  
B

*[Handwritten signature]*

Project No. 7010

Prepared by

THE MITRE CORPORATION  
Bedford, Massachusetts

Contract No. F19628-78-C-0001

Approved for public release;  
distribution unlimited.

79 06 20 010

DA 070673

DDC FILE COPY

When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Do not return this copy. Retain or destroy.

#### REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

*Paul H. Wendzikowski*

PAUL H. WENDZIKOWSKI, Capt, USAF  
Project Officer

*William M. Smith, Jr.*

WILLIAM M. SMITH, Jr., Col, USAF  
Director, Technological and  
Functional Area Planning  
Deputy for Development Plans

*Ernest L. Hatchell Jr.*

ERNEST L. HATCHELL JR., Colonel, USAF  
Assistant Deputy for Development Plans

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

<b>REPORT DOCUMENTATION PAGE</b>		<b>READ INSTRUCTIONS BEFORE COMPLETING FORM</b>	
<b>18</b> 1. REPORT NUMBER ESD-TR-79-124-VOL-3	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	<b>9</b> technical rept.
<b>6</b> 4. TITLE (and Subtitle) GENERALIZED CYCLIC CODES FINITE FIELD ARITHMETIC . Volume III.		5. TYPE OF REPORT & PERIOD COVERED	
7. AUTHOR(s) <b>10</b> T. F./Roome		<b>14</b> 8. PERFORMING ORG. REPORT NUMBER MTR-3227 - Vol. 3	9. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation P. O. Box 208 Bedford, MA 01730		<b>15</b> 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS F19628-78-C-1001 Project No. 7010	
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Development Plans Electronic Systems Division, AFSC Hanscom AFB, MA 01731		<b>11</b> 12. REPORT DATE MAY 79	13. NUMBER OF PAGES 66
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) <b>12</b> 70p.		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			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  BLOCK CODES                                  MULTIPLE LEVEL LOGIC CHARGE-COUPLED DEVICES                  TRANSVERSAL FILTERS CODING			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) → The implementation of finite field arithmetic processing elements in charge-coupled device technology for cyclic error coding is investigated. The strong coupling between the capabilities of the multiple signal level charge-coupled device delay line and the structures needed for nonbinary finite field arithmetic algorithms is shown to have a potential for realizing resource efficient error control coding computational →			

(over)

235050 *Am*

**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)**

**20. ABSTRACT (concluded)**

→ hardware. Laboratory results are described, and recommendations are made in areas where device characteristics require improvement to better meet the needs of the coding structures. ↖

**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)**

### ACKNOWLEDGMENTS

This report has been prepared by The MITRE Corporation under Project No. 7010. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

The author gratefully acknowledges the contributions to this effort of D. O. Carhoun, R. D. Haggarty and E. A. Palo.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist.	Avail and/or special
<b>A</b>	

## TABLE OF CONTENTS

	<u>Page</u>
List of Illustrations	4
List of Tables	5
Section I	INTRODUCTION 6
Section II	FINITE FIELD ARITHMETIC 9
Section III	CHARGE-COUPLED DEVICE DELAY LINES 28
Section IV	CHARGE-COUPLED DEVICE FINITE FIELD ARITHMETIC STRUCTURES 36
Section V	CONCLUSIONS 48
Appendix A	CHARGE-COUPLED DEVICE PROPERTIES 50
References	66

## LIST OF ILLUSTRATIONS

<u>Figure Numbers</u>		<u>Page</u>
1	Base Field Arithmetic in GF(7)	13
2	Reed-Solomon Rate 1/2 Encoder	15
3	Permutation Method of Finite Field Addition in GF(7)	19
4	Permutation Method of Finite Field Multiplication in GF(7)	22
5	Permutation Method of Multiplication by a Constant in GF(7)	24
6	Encoder Implemented with Permutation Multiplier Tap Weights	25
7	Analysis Model	30
8	Signal to Noise Performance vs. Field Characteristic	31
9	Probability of Error vs. Signal to Noise Ratio	35
10	CCD Multiple Level Storage Performance	38
11	Block Diagram of Finite Field Arithmetic Unit	41
12	Finite Field Arithmetic Unit	42
13	Comparator Input Signals	45
14	Detection Performance	45
A-1	Cross Sectional View of a MOS Capacitor	51
A-2	Potential Well of Single Cell	51
A-3	Three Phase CCD Clock Waveforms and Potential Well Diagrams	53
A-4	Common Input Structure	55
A-5	Amplitude Response of N Transfer CCD with Loss per Transfer $\epsilon$ vs Frequency	62
A-6	Simulation of Time Dispersion in CCD with Loss per Transfer $\epsilon$	63
A-7	Split Electrode Transversal Filter	64

LIST OF TABLES

<u>Table Numbers</u>		<u>Page</u>
1	Requirements for Encoder (Direct Implementation)	17
2	Requirements for Encoder (Permutation Multipliers)	26

## SECTION I

### INTRODUCTION

This report is Volume III of MTR 3227 entitled Generalized Cyclic Codes. Volume III addresses the development of efficient hardware implementations of cyclic block code arithmetic operations described in Volume II. Volume I addresses the application of error correction coding in the context of Command, Control, and Communication (C<sup>3</sup>) systems.

The thrust of this volume is the exploration of the feasibility of directing the new technology of charge-coupled devices toward the development of reliable and resource efficient error control coding hardware. Charge-coupled device technology, with its inherent large packing density and low power consumption, is capable of producing significant size, power and eventually, cost savings in signal processing, imaging, and memory applications [Ref. 1]. These devices, with their discrete time, continuous amplitude shift register storage capability, are very closely aligned with the shift register structures required in error control coding arithmetic operations. Thus, it seems appropriate to examine the details of such a marriage - CCD's and error control coding.

This report focuses on the arithmetic operations required in cyclic block codes. The arithmetic function is a major component in encoders and decoders.

Presently, error detection and correction techniques using block codes are primarily confined to designs based on binary algorithms. This restriction is predicated on the assumption that the availability of binary integrated circuits makes hardware

implementation simple and cost effective, or that the software approach in a general purpose computer allows the flexibility of changing coding parameters. Both assumptions are only partially true. There is a great availability of low cost binary integrated circuits but few of these were specifically designed for coding applications. There is, consequently, an inefficiency associated with designs based on components manufactured for general purpose usage. Coding hardware built with standard digital hardware usually has a large package count and power consumption due to the functionally simplistic structures contained in each package.

Alternatively, the software approach in a general purpose computer suffers from the large overhead required to maintain its flexibility. This overhead appears in the power and space consumed by the general purpose machine. Indeed, in many cases the need for such flexibility in coding is questionable. More important, we are becoming increasingly aware of the significant costs associated with the software design for this approach.

Thus, it seems that the normal implementations do not have substantial potential at this moment for great cost savings.

Our approach is a radical one. It hopefully will perform a marriage between the theory of cyclic block coding arithmetic structures and the emerging technology of charge-coupled devices to produce simple, possibly single chip, structures that will result in significant cost savings.

This report begins in Section II, with an investigation of the mathematical structure of finite fields, especially the properties of the arithmetic operations. The investigation concludes that an alternate approach to direct arithmetic calculations followed by

modulo reduction is possible using cyclic properties of the field elements. This approach requires a counting and comparison algorithm that lends itself to a shift register implementation.

Section III explores the use of charge-coupled devices to implement the shift register portions of the algorithm. The loss and noise parameters of a CCD delay line are included in an analysis along with the requirements of different size finite fields to determine the bounds of reliable performance.

Section IV describes an arithmetic unit architecture for performing either multiplication or addition in a finite field. Laboratory results are shown for a breadboard implementation of the critical portions of an arithmetic unit using commercially available CCD delay lines. The experience gained in testing such an arithmetic unit leads to a definition of areas where device characteristics require improvement to better meet the needs of the coding structures.

SECTION II  
FINITE FIELD ARITHMETIC

BACKGROUND

Linear cyclic codes are conveniently described in the language of the theory of finite fields. Finite fields, or Galois fields, are represented by the symbol  $GF(p^m)$ , where  $p$  is the field characteristic and  $m$  the degree of extension. The characteristic  $p$  must be a prime number and the degree of extension  $m$  must be an integer. If  $m=1$  the field is a base field containing  $p$  elements. These elements are the integers 0 to  $p-1$  modulo  $p$ . If  $m>1$  the field is an extension field containing  $p^m$  elements. The extension field elements are usually represented by polynomials with coefficients from  $GF(p)$ , modulo some irreducible polynomial of degree  $m$ . An irreducible polynomial of degree  $m$  is a polynomial which is not divisible by any polynomial of degree less than  $m$ . The irreducible polynomial cannot be zero. Given an irreducible polynomial  $p(x)$  with the integers 0 to  $p-1$  as coefficients, a representation of the field  $GF(p^m)$  with  $p^m$  elements can be formed. It consists of all polynomials of degree  $m-1$  or less.

For instance, the field  $GF(3^2)$  has the nine elements listed below:

0  
1  
2  
x  
2x  
x+1  
x+2  
2x+1  
2x+2

The elements can be added term by term modulo  $p$  in the ordinary way. An addition of the last three terms produces the result,

$$5x + 5$$

which when reduced modulo 3 leaves

$$2x + 2$$

Multiplication is performed in the ordinary way but requires reduction of each term modulo  $p$  and of the polynomial modulo  $p(x)$  to a polynomial of degree  $m-1$  or less. The simplest approach is to consider  $p(x)=0$  and use this equation to eliminate terms of degree greater than  $m-1$ . Consider multiplying the last two terms in the list of field elements for  $GF(3^2)$ ;

$$(2x + 1)(2x + 2) = 4x^2 + 6x + 2$$

and following this by reduction modulo 3 on each element;

$$x^2 + 2$$

Finally, modulo reduction on this result is performed by the irreducible polynomial  $p(x) = x^2 + x + 2$ . This final reduction is done by setting  $p(x)=0$ , solving for the highest order term, and substituting this into the arithmetic result;

$$p(x) = x^2 + x + 2 = 0$$

$$x^2 = -x - 2 = 2x + 1$$

Upon substitution into the multiplication result the reduced answer becomes

$$(2x + 1) + 2 = 2x + 3 = 2x$$

Base field arithmetic is simpler. Arithmetic in such a field can be performed in the field of all integers followed only by reduction modulo  $p$ .

An example of arithmetic in the base field GF (7) is shown in Figure 1, which presents tables for defined operations. Obviously, arithmetic operations in a base field are much simpler than arithmetic operations in an extension field since no polynomial reduction is required. Volume II shows that good error control coding performance can be obtained with base field codes, and their simpler arithmetic makes them even more attractive.

In any finite field there is at least one field element such that any other non-zero field element can be expressed as a power of this element. This element, called a primitive element  $\alpha$ , is a non-zero integer in a base field or is a non-zero polynomial in an extension field. This polynomial is called a primitive polynomial.

For the base field GF(7) the element 3 is a primitive element and consequently all the non-zero GF(7) elements can be generated as a power of 3, i.e.,

$$3^0 = 1$$

$$3^1 = 3$$

$$3^2 = 9 \bmod 7 = 2$$

$$3^3 = 6$$

$$3^4 = 18 \bmod 7 = 4$$

$$3^5 = 12 \bmod 7 = 5$$

$$3^6 = 15 \bmod 7 = 1 = 3^0 \text{ (check)}$$

For GF ( $3^2$ ) a primitive element is the polynomial  $x+1$ . A list of the successive powers of this element produces all the non-zero field elements;

$$\alpha^0 = 1$$

$$\alpha^4 = 2$$

$$\alpha^1 = x + 1$$

$$\alpha^5 = 2x + 2$$

$$\alpha^2 = x + 2$$

$$\alpha^6 = 2x + 1$$

$$\alpha^3 = 2x$$

$$\alpha^7 = x$$

$$\alpha^8 = 1 \text{ (check)}$$

The polynomial reduction required for arithmetic operations in an extension field obviously is a complication not contained in base field arithmetic. Volume II showed that Reed-Solomon block codes require only the use of one field, and a choice of this to be a base field results in no degradation in rate performance.

ADDITION GF(7)

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

MULTIPLICATION GF(7)

x	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	1	3	5	6
3	0	3	6	5	1	4	2
4	0	4	1	2	6	3	5
5	0	5	3	4	2	6	1
6	0	6	5	4	3	2	1

Figure 1. BASE FIELD ARITHMETIC IN GF (7)

In this report we restrict the discussion of implementations to this class of codes and thus consider only arithmetic operations in a base field.

#### IMPLEMENTATION

The direct implementation of base field arithmetic requires that the calculations be done in the ordinary way and the modulo reduction be performed on the final result. Although mathematically characterized as a division operation that retains only the remainder, the reduction can be implemented in serial form by successive subtraction of the field characteristic, or in parallel form by a bank of window comparators. The complexity of each of these approaches is a function of the number and type of arithmetic operations producing the result.

For the successive subtraction method, the field characteristic  $p$  is subtracted continually from an operand until the result is contained in a window bounded by the zero element and the largest field integer,  $p-1$ . The number of required iterations depends on the operand value.

A single addition requires at most one iteration. A single product requires at most  $p-2$  iterations. If an algorithm such as one commonly used in successive approximation analog-to-digital converters is followed, then the maximum number of iterations for reduction following a single product is approximately  $\log_2(p-1)$ .

Figure 2 shows a block diagram of a Reed-Solomon encoder for a rate  $1/2$  code. This structure requires  $(p-1)/2$  stages of shift

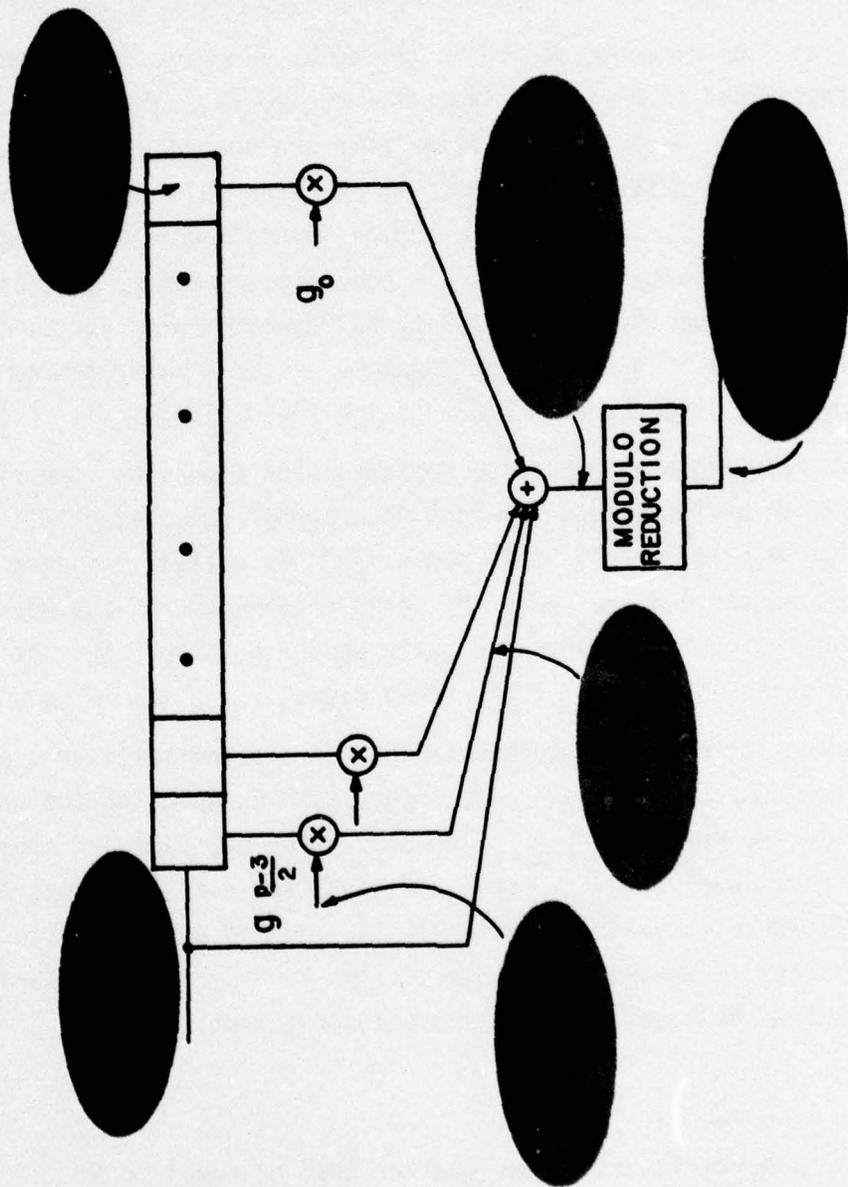


Figure 2. REED-SOLOMON RATE 1/2 ENCODER  
(ORDINARY ARITHMETIC FOLLOWED BY MODULO REDUCTION)

register delay; each stage will carry any of the  $p$  values of the field elements. In terms of arithmetic operations each output value requires  $(p-1)/2$  multiplications and  $(p-1)/2$  additions.

Since the elements stored in the delay line and the coefficients are constrained to be field elements, the largest multiplier output is  $(p-1)^2$ . When the  $(p+1)/2$  outputs are summed, the maximum value of the computation is  $p-1+(p-1)^3/2$ . The shift register must have sufficient dynamic range to contain  $p$  distinguishable levels of the symbol field, but the modulo reduction circuitry must accommodate this maximum value. The additional dynamic range factor is designated here as excess dynamic range. It is a major source of concern since it grows as a power of the field size.

By applying the successive approximation algorithm, the maximum number of iterations for each code symbol calculated will be  $\lceil \log_2((p-1+(p-1)^3/2)/p) \rceil$ ; this number  $\lceil x \rceil^1$  is called the excess computation since it reflects the speed requirement of the modulo reduction circuitry against the basic shift register rate. It too increases as a power of the field size.

Reduction modulo  $p$  may be carried out alternatively by a bank of window comparators, each of aperture size  $p$ , spanning the range from 0 to the maximum value. Each comparator subtracts an appropriate bias level from its input and tests the result against the field aperture to select immediately the reduced output. With this method speed is traded for complexity;  $(p-1+(p-1)^3/2)/p$  comparators are required at most to reduce the encoder output.

---

<sup>1</sup> $\lceil x \rceil$  denotes the least integer greater than or equal to  $x$

TABLE I. REQUIREMENTS FOR ENCODER (DIRECT IMPLEMENTATION)

Field Characteristic, $p$	Maximum Excess Dynamic Range, $20 \log((p-1)^2/2+1)$	Maximum Excess Computation for Successive Subtraction $\lceil \log_2((p-1)^3/2)/p \rceil$	Number of Window Comparators for Parallel Reduction
3	9.5	1	2
5	19.0	3	7
7	25.5	5	16
11	34.1	6	46
13	37.2	7	67
17	42.2	7	121
19	44.2	8	154
23	47.7	8	232

Maximum required values of excess dynamic range and excess computation in the encoder are tabulated as functions of field characteristic in Table 1. The values of excess computation pertain only to the sequential method, and the number of comparators pertain only to the parallel method; values of excess dynamic range are pertinent to both methods.

The excess dynamic range grows as the square of the field characteristic making the reduction much more difficult as  $p$  increases. This is the central issue of the problem we investigate here. Many coding operations, such as the encoder of Figure 2, require multiple arithmetic operations that dramatically increase the dynamic range requirement of the arithmetic section.

The implementation of the arithmetic section must either insert modulo reduction into its arithmetic structure at more frequent intervals to prevent a large dynamic range requirement, or find an alternative to direct modulo reduction.

An alternate method of implementing finite field arithmetic uses cyclic permutations of the elements of  $GF(p)$ . The permutation method reduces the dynamic range requirement of the direct reduction techniques.

Notice in Figure 1 that the rows of the addition table are successive cyclic shifts of the sequence  $0, 1, \dots, p-1$ . Addition may be performed by comparing cyclic shifts of this sequence of integers with an operand until equality is established between the operand and the first element of the permuted sequence. If the operand is  $k$  and the value to be added is  $n$ , then  $k$  shifts will

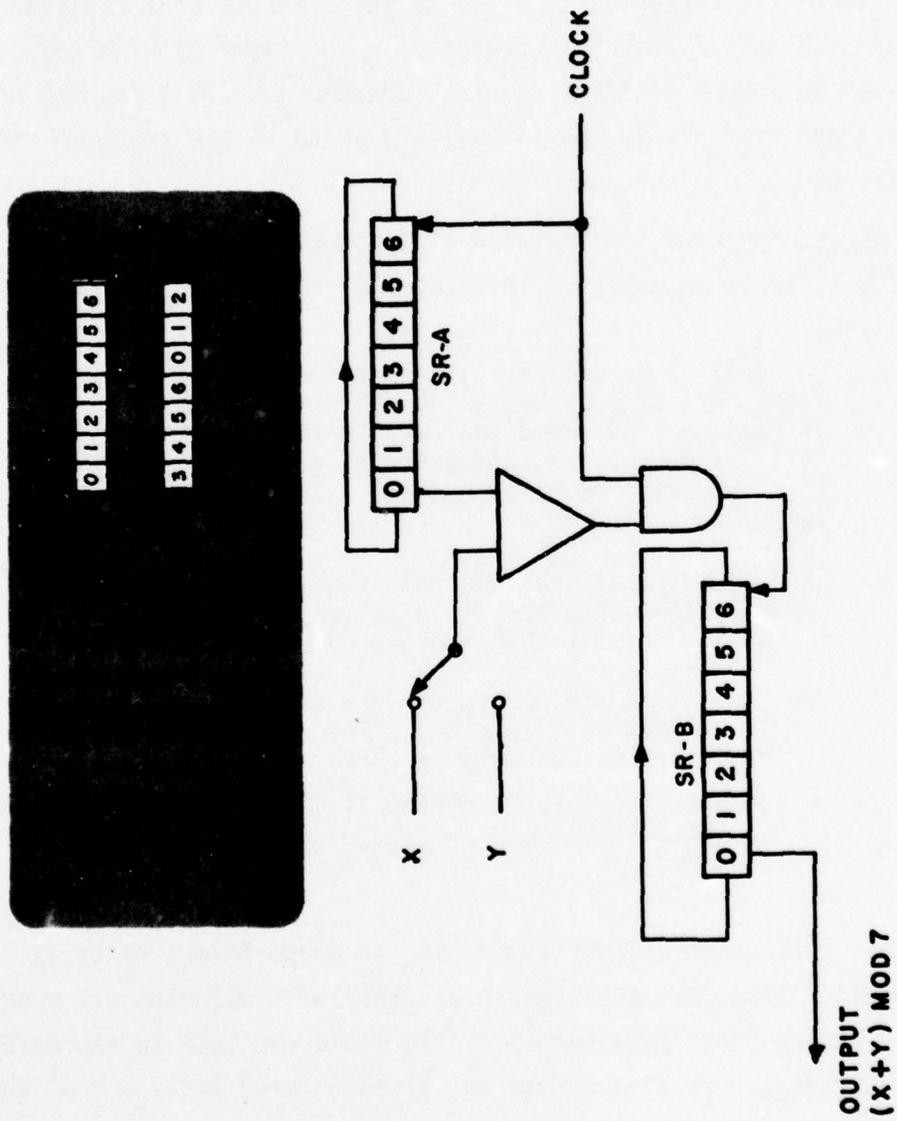


Figure 3. PERMUTATION METHOD OF FINITE FIELD ADDITION IN GF(7)

be performed and the modulo reduced sum will be the value of the  $(n+1^{\text{st}})$  element of the shift sequence. The cyclic permutation of the sequence suggests implementation with a feedback shift register. Addition of the element 3 in GF(7) to the contents of a register loaded with this sequence is depicted in the inset of Figure 3. A register is loaded with the required sequence, shifted and fed back 3 times and effectively has the value 3 added to all register cells and the result modulo reduced.

An architecture for addition of two numbers in a finite field (GF(7)) is illustrated in Figure 3. The operation proceeds as follows:

- a. The value  $X$  is compared with the contents of the first cell of register SR-A, while the contents of registers SR-A and SR-B are shifted to the left in synchronism, and shifting continues until the comparator detects equality. The number of shifts performed is  $X$ .
- b. SR-A is reset to the original sequence, and the input of the comparator is switched to the value  $Y$ . Shift register SR-B remains unchanged.
- c. Registers SR-A and SR-B again are shifted synchronously as step 1 is repeated. The accumulated value in the first cell of SR-B is  $(X+Y) \bmod 7$ .

The process could continue for a series of field elements, and the accumulated sum would appear in the first cell of shift register SR-B.

Multiplication in the base field can be performed by cyclic permutation also, but a different sequence of field elements must be used. The non-zero elements of the field can each be expressed as a distinct power of a primitive field element i.e.,  $\beta = \alpha^i$  where  $i$

is an integer in the set  $0, 1, \dots, p-2$ . Multiplication of two non-zero field elements can be expressed as a product of powers of the primitive element.

$$\beta_i \cdot \beta_j = \alpha^i \cdot \alpha^j = \alpha^{i+j}$$

If the sequence  $(\alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{p-2})$  is used to represent the non-zero field elements cyclic permutation can be used to calculate products. The operation can be regarded as multiplication by addition of exponents, but the values of the elements rather than their exponents can be manipulated directly. Multiplication by the zero element must be accomplished separately. As shown in Figure 4 the integer 3 is a primitive element of  $GF(7)$ ; 3 was chosen as the generating element, and the multiplication table in the inset has been rearranged to reflect the cyclic permutations of the powers of 3 (modulo 7). The inset of Figure 4 depicts multiplication of the register contents by the value  $\alpha^3 = 6$ .

The architecture to implement multiplication in  $GF(7)$  is shown in Figure 4. The multiplication structure is similar to the one used for addition, except that the registers are one cell shorter (the zero element is excluded) and the sequence of elements has been rearranged. Otherwise, the operation is the same: each operand is compared with the contents of the first cell of register SR-A, which is circulated until the values are equal and then reset to the initial sequence before the second operation is entered. The second shift register (SR-B) is used to accumulate the total number of shifts for the cycle and compare operations. If either input is zero, a zero detector switches the output to the zero level.

For multiplication by a constant, only register SR-A is needed. There is no requirement for an accumulator, and the product may be

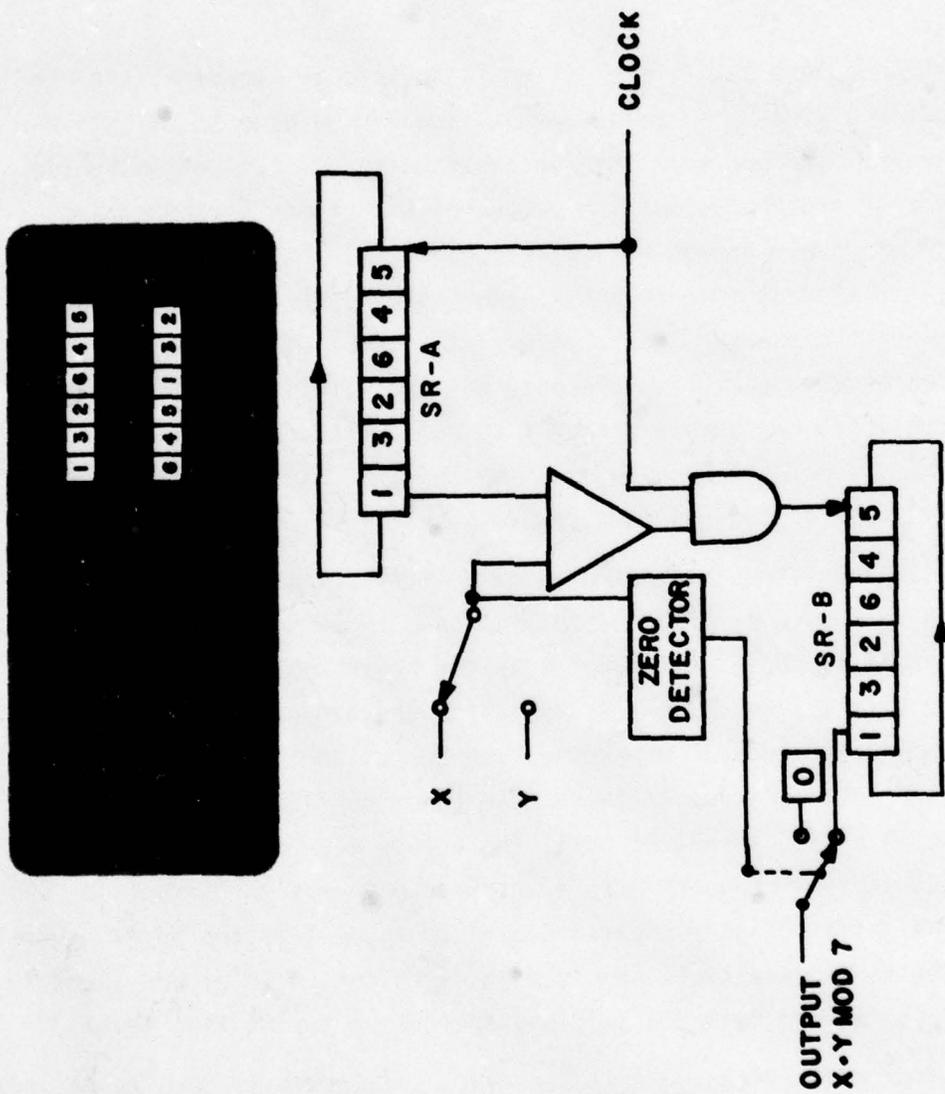


Figure 4. PERMUTATION METHOD OF FINITE FIELD MULTIPLICATION IN GF(7)

taken from the register cell that contained the multiplication constant in the initial sequence. If the operand is  $\alpha^k$ , then  $k$  shifts are required before equality is sensed by the comparator, and each cell has its contents multiplied by the factor  $\alpha^k$ . By selecting the cell from which to take the output we can choose the multiplicative constant. Figure 5 illustrates a structure for multiplication in  $GF(7)$  by the value 4. The second register can also be eliminated in the adder structure for addition of a constant value. Finite field arithmetic performed by the permutation method has the advantage that results do not exceed the largest field element; this directly reduces the dynamic range required in tapped delay line coding structures.

A configuration using the permutation multiplier in an encoder is shown in Figure 6. The tap weights are shown implemented with the multipliers of Figure 5, each  $p-1$  stages long, while summation of their outputs is indicated by an analog adder followed by reduction modulo  $p$  (implemented, for instance, with a successive subtraction circuit). Analog addition with modulo reduction at the output is faster than permutation addition in this multiple operand application. The permutation multipliers produce values within the field and limit the output of the adder to a maximum  $(p-1)(p+1)/2$ , significantly less than the direct approach of Figure 2. Each code symbol calculated requires at most  $p-1$  shifts for the permutation multipliers plus a maximum number of approximately  $\lceil \log_2((p-1)(p+1)/2p) \rceil$  iterations for the modulo reduction by successive subtraction. The maximum values of the parameters of this encoder configuration are listed in Table II for several values of field characteristic.

IA-50,979

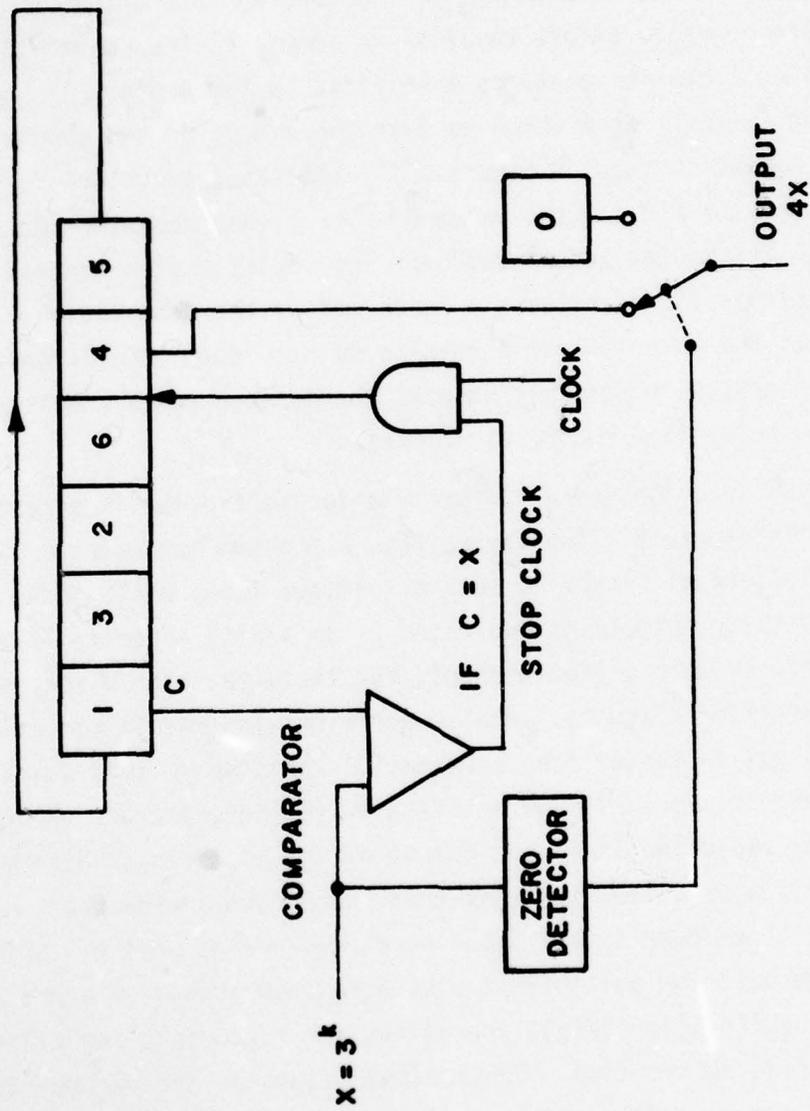


Figure 5 PERMUTATION METHOD OF MULTIPLICATION BY A CONSTANT IN GF (7)

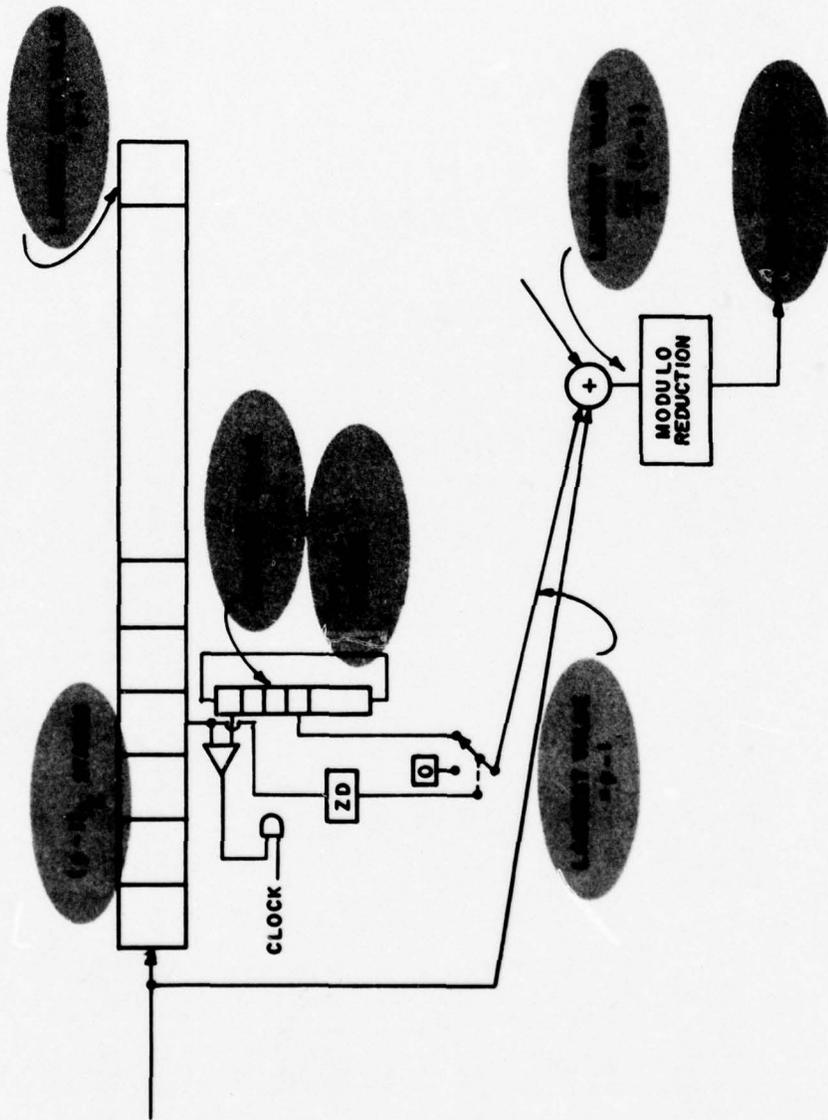


Figure 6. ENCODER IMPLEMENTED WITH PERMUTATION MULTIPLIER TAP WEIGHTS

TABLE II. REQUIREMENTS FOR ENCODER (PERMUTATION MULTIPLIERS)

Field Characteristic, $p$	Maximum Excess Dynamic Range, $20 \log(p+1/2)$	Maximum Excess Computation $[(\log_2(p-1))(p+1)/2p] + p-1$	Total Number of Shift Register Stages, $p(p-1)/2$
3	6.02	3	3
5	9.5	6	10
7	12.04	8	21
11	15.5	13	55
13	16.9	15	78
17	19.0	20	135
19	20	22	171
23	21.5	26	253

Comparison of Table II with Table I indicates a significant reduction in the excess dynamic range if permutation multipliers are used for the tap weights.

The excess dynamic range now only increases linearly with the field characteristic and the possibility of using larger field characteristics seems more reasonable. However, this is achieved by the use of more shift registers and operation cycles. This is a bad trade. The new technology of charge-coupled devices is capable of producing very small shift registers and integrating these into multiple register designs on a single chip, and this technology can be brought to bear on the arithmetic techniques described above.

Additionally the CCD has the potential for very high speed operation, thereby making the increased number of operation cycles an insignificant factor for many coding applications.

### SECTION III

#### CHARGE-COUPLED DEVICE DELAY LINES

The advantage of charge-coupled device technology lies in its ability to achieve small size and weight and low power consumption. One of the most important developments to evolve from the charge-coupled device efforts is the efficient realization of an analog shift register structure. Analog shift registers of length up to 1000 storage locations have been built, with registers of length 100 to 300 stages being more typical. These lengths are more than adequate for our base field coding purposes. In addition, multiple registers have been fashioned on single chips in structures not too distant from those proposed for the encoder of Section II. Image sensing devices and bulk digital memory devices use multiple registers and complex clocking schemes to store and move signal samples throughout the structure. One hundred by one hundred CCD imagers (10K cells) and 64K memories have been built. One can conclude that the technology is more than adequate for implementing permutation arithmetic functions in a few chips or possibly a single chip. There are, however, other questions to be addressed regarding the device's ability to perform the permutation arithmetic function.

This section addresses the usefulness of using CCD registers in the permutation arithmetic architectures defined in Section II. Charge-coupled device delay lines would be used to store and shift the field elements as required in either the addition or multiplication operation. The field elements would be represented as discrete packets of charge. A CCD register would have to handle  $p$  values. The specific problem we address is to quantify how many levels ( $p$ ) can be carried reliably in a CCD shift register with a given level of loss and noise characteristics. This result will bound the field

sizes available for block codes implemented with this technology.

Appendix A presents a discussion of the important aspects of charge-coupled device performance including a discussion of the loss mechanism and the noise characteristics of shift registers.

For arithmetic operation in  $GF(p)$  an upper bound on shift register performance is obtained by examining a  $p$ -stage register, with a  $p$ -level detector and regenerator circuit as shown in Figure 7. The shift register will be assumed to be three phase. The measure of performance is the probability of making an error in detecting a certain level at the output of this  $p$ -stage CCD.

The register will be specified in terms of two parameters. The first is the signal-to-noise ratio of the device, and the second is the device transfer efficiency. The signal-to-noise ratio in this case includes not only noise contributions within the device and its input stage, but also any uncertainty in the detector window levels and in non-ideal regeneration.

In essence, the analysis that follows shows that because of the short length of the required registers, the transfer inefficiency product is small and, in fact, the loss mechanism described in Appendix A has little effect on the results. It is also shown that overall signal-to-noise ratios of about 60-70 dB will be required for a probability of error of  $10^{-10}$  in the smaller fields. Figure 8 summarizes the results and is a plot of the required signal-to-noise ratio for a  $p$ -stage register carrying  $p$  levels with a transfer efficiency of .9999.

IA-50,985

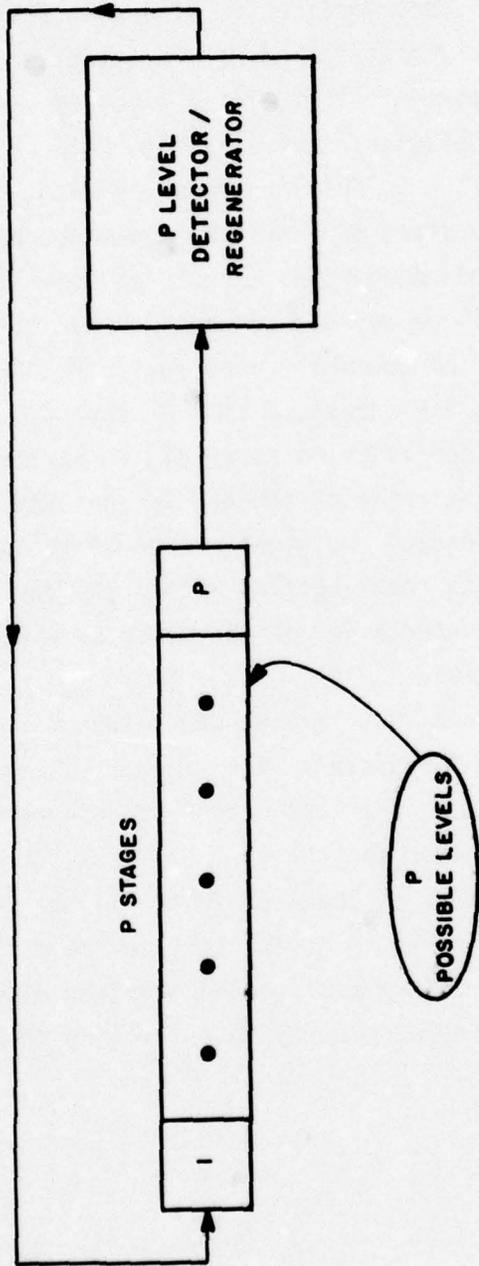


Figure 7. ANALYSIS MODEL

IA-50,986

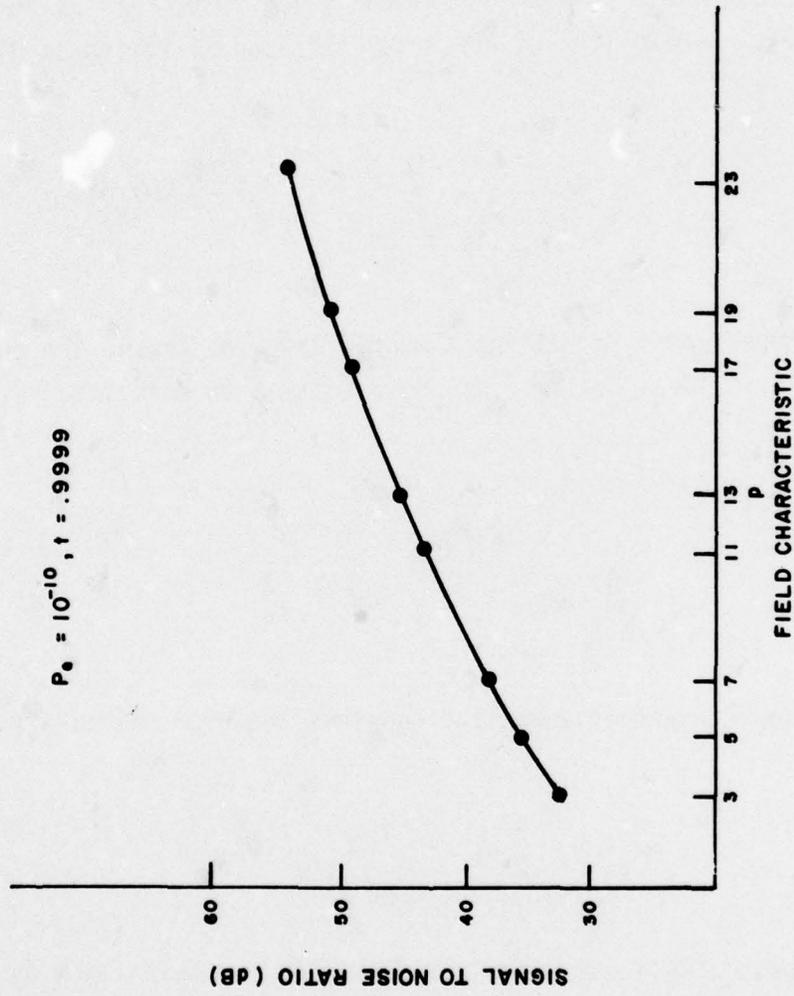


Figure 8. SIGNAL NOISE PERFORMANCE VS FIELD CHARACTERISTIC

The combined noise sources of the register, window levels, and regenerator circuits are assumed to be Gaussian with variance  $N^2$ . The signal is defined as the peak signal value to be carried in the register. Probability of error for this configuration is given by:

$$P_e = \frac{2}{(\sqrt{2\pi})N} \int_{mN}^{\infty} \exp^{-x^2/2N^2} dx$$

which corresponds to setting a window  $2mN$  wide around the expected value in an ideal device. An approximation to this integral is:

$$\int_a^{\infty} \exp^{-x^2/2} dx = \frac{3}{4a} e^{-a^2/2}$$

and making the proper substitution gives the probability of detection error:

$$P_e = \left( \frac{2}{(\sqrt{2\pi})N} \right) \left( \frac{3}{4mN^2} \right) \exp \frac{-m^2 N^2}{2N^2}$$

Considering a unit standard deviation and defining  $m$  as the number of standard deviations away from the expected value that the threshold is set, a simplified probability of error equation results:

$$P_e = \frac{.6}{m} \exp^{-m^2/2}$$

This relation defines the obvious tie between window size ( $2mN \rightarrow 2m$ ) and  $P_e$ . For a given  $P_e$  and maximum signal level, we can find the number of allowable distinct levels to be carried in the device.

This is,

$$K_1 = \frac{1}{2m} \frac{S}{N}$$

However, we must also consider the effect of imperfect transfer in the device and the impact of leaving charge behind and thus contributing to succeeding signal levels. Two things are important in our case: first, because we are concerned with small register lengths, the transfer inefficiency product is low for even somewhat lossy devices; second, for low inefficiency product, the amount lost is approximately the inefficiency product times the signal level and is contained in the next signal sample only. Thus, we can account for the worst case by opening the windows by an amount equal to twice the inefficiency product times the peak signal.

So now the window opening will be:

$$2nN + 2n\bar{\epsilon} S$$

with  $n$  equal to the number of register stages and  $\bar{\epsilon}$  equal to the loss per stage. The maximum number of levels is now equal to:

$$K_2 = \frac{S/N}{2m + 2n \bar{\epsilon} S/N}$$

Solving for  $m$  here and substituting this value in the calculation for  $P_e$  determines the new probability of error relation:

$$P_e = \frac{.6}{S/N \left( \frac{1}{2k_2} - n\bar{\epsilon} \right)} \exp \frac{-S^2}{2N^2} \left( \frac{1}{2k_2} - n\bar{\epsilon} \right)^2$$

In the coding case there are at most  $p$  levels ( $k_2=p$ ), and  $p$  stages ( $n=p$ ) so that  $P_e$  becomes:

$$P_e = \frac{.6}{\left( S/N \frac{1}{2p} - p\bar{\epsilon} \right)} \exp \frac{-S^2}{2N^2} \left( \frac{1}{2p} - p\bar{\epsilon} \right)^2$$

Figure 9 shows probability of error curves for different values of  $p$  (i.e.,  $p$  levels in a  $p$  stage register) and for two values of transfer loss. The results indicate that the small register size required in coding, usually less than  $p$  stages, drives the level of dispersion to the point where it is a second order effect. Transfer efficiencies of .999 per stage are typical.

IA-49, 766

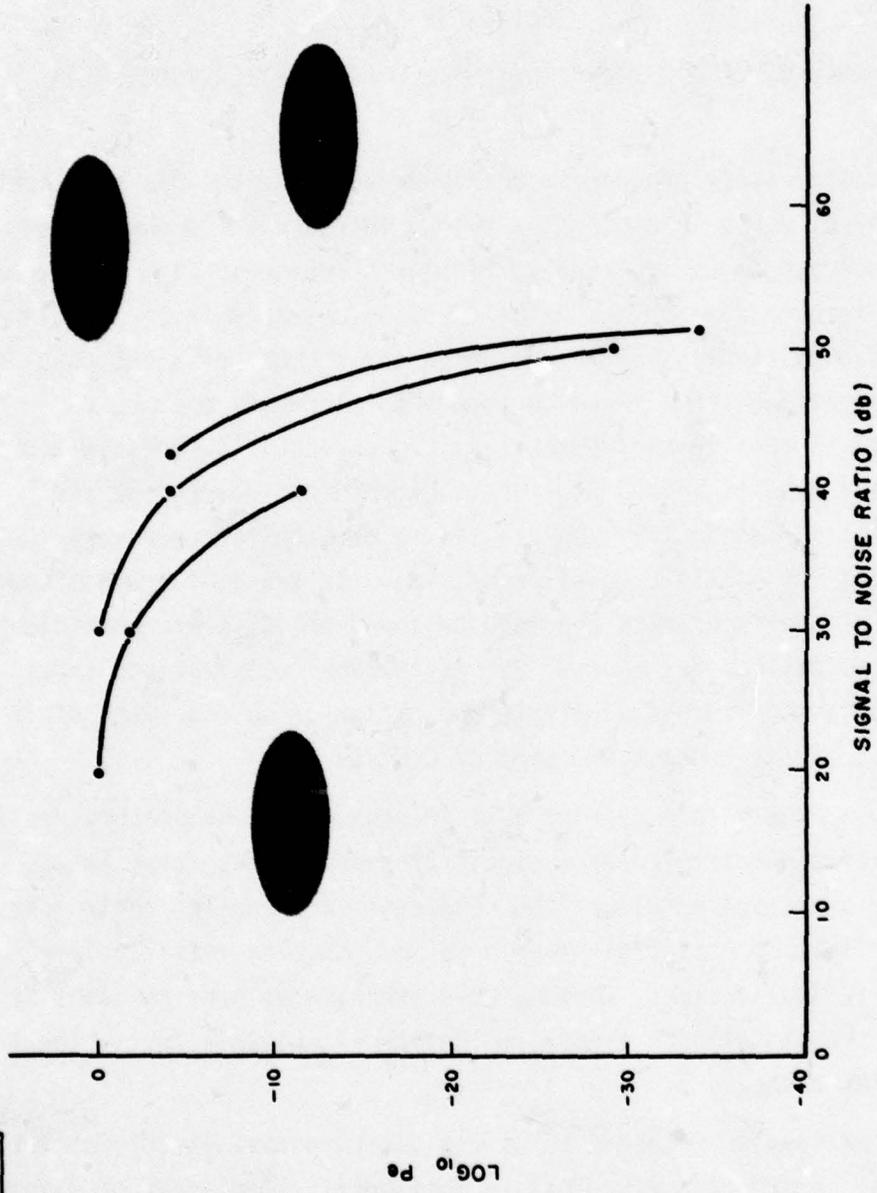


Figure 9 PROBABILITY OF ERROR VS SIGNAL TO NOISE RATIO

## SECTION IV

### CHARGED COUPLED DEVICE FINITE FIELD ARITHMETIC STRUCTURES

The laboratory program in charge-coupled devices has complemented the study of CCD's in coding and helped increase the understanding of various phenomena in these particular structures. Its thrust has been to investigate the potential of doing multiple level signal and data processing in the high signal-to-noise environment of a CCD chip and to highlight the areas in which future developments must be concentrated to realize this potential. In effect it is anticipated that these results should provide the impetus to government and industrial semiconductor laboratories to develop the necessary technology for multiple level processing. In truth, there has been a paucity of published data and results about the discrete multiple level applications for CCD's. Our discussions with various industrial laboratories has turned up little indication of an awareness of this potential signal processing usage of CCD's.

Our approach to providing this impetus in the laboratory was to realize simple multiple level signal processing structures from existing small scale integration components and examine their operation in light of potential usage in a more complex multiple level monolithic application. Through this approach we hope to identify some of the pitfalls in the actual design of multiple level signal processing chips.

Of particular interest to us was the potential structures for base field arithmetic described in Section II. Two areas of major concern are:

- The multiple level signal carrying capacity of charge coupled devices.
- The comparator structure required in the permutation tapped delay line.

Both of these areas impact the reliability with which arithmetic operations can be performed in a charge coupled device.

Of secondary interest in this effort were:

- The matching of levels from input to output of a signal processing structure.
- Implementation of the two-dimensional, independent clocking operation required in the permutation tapped delay line.
- Matching of the transfer characteristics and levels of multiple CCD's needed in the permutation approach.
- Definition of any required modulo reduction circuit.

The laboratory program has tried to estimate the relative importance of these areas, to see if any related work is underway and to identify areas where substantial improvement is required.

The first effort was construction of a CCD delay line test board and the investigation of its response to multiple signals. Figure 10 shows a block diagram of the test setup. The binary feedback shift register generates a code sequence, the D/A converter transforms the sequence into a multiple level sequence, and the sequence enters the CCD register where it is transferred to the output. A comparison of the input and output gives a subjective indication of the corruption of the multiple level sequence.

The register selection was confined to what was commercially available in mid-1975; the Fairchild CCD-311, dual 130 stage analog register, was the only device available. The device can be operated using either of the two 130 stage registers or the two can be time multiplexed to form a single 260 stage device.

In the time multiplex mode the registers share a common output stage and are clocked in synchronism. The data of Figure 10 was

IA-49,771

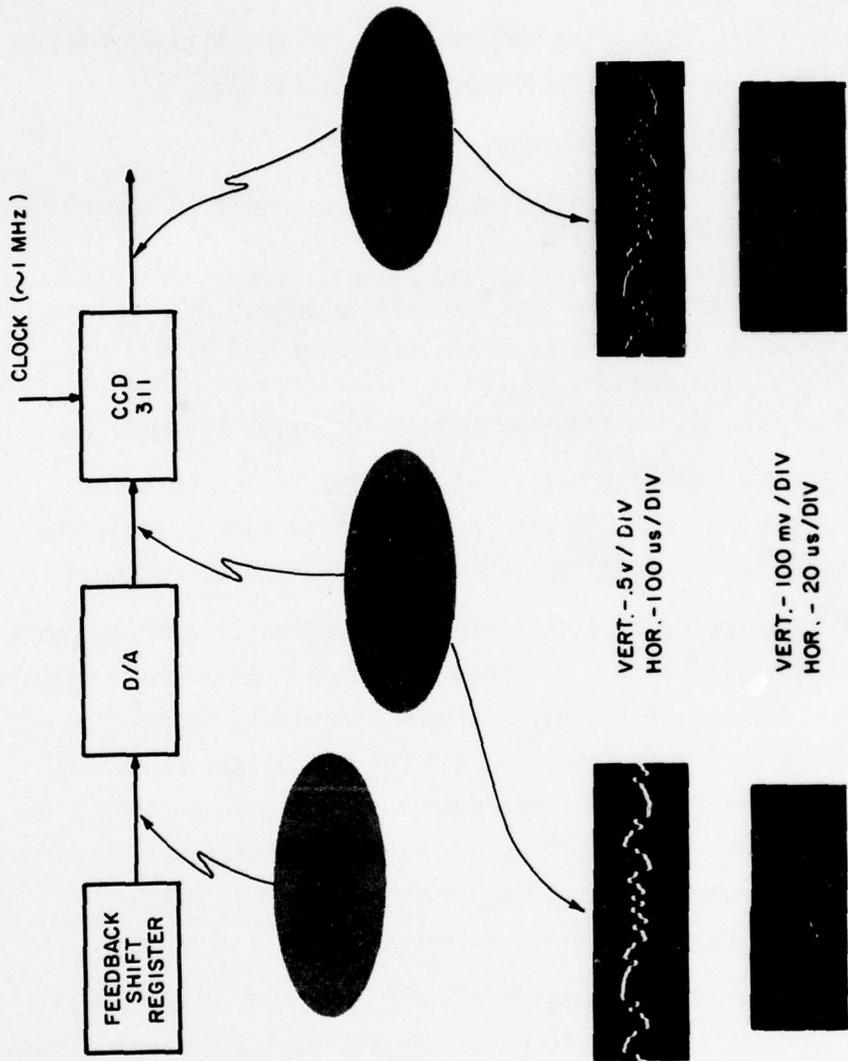


Figure 10 CCD MULTIPLE LEVEL STORAGE PERFORMANCE

taken with only one register active and a clock rate of approximately 1 MHz. Because of conversion limitations of the D/A converter each discrete level is sampled approximately twenty times by the CCD-311, putting a little more than six discrete levels in the device at any one time. The top left photograph in Figure 10 shows one input sequence to the CCD-311 while the top right photograph shows the output sequence. The horizontal scale is uncalibrated at approximately 100  $\mu$ s/div. The vertical scale is .5v/div. Subjectively, it can be seen that the overall fidelity of the input sequence is maintained at the output.

A closer look at a different multiple level sequence is shown in the input and output photographs of the bottom left and bottom right sections of Figure 10. The horizontal scale in this case is uncalibrated at approximately 20  $\mu$ s/div. and the vertical is 100 mv/div. Again the output gives a subjective indication that levels separated by a few tens of millivolts are discernable and probably detectable. No detailed statistical data was taken because the error probabilities that are required preclude any simple testing measures and because the structures implemented here are not the ones desired.

Although no statistical data is available, the results seem encouraging given the excessive length of the device for most of our intended applications. Consequently, the next step was to develop a more complex function to determine the feasibility of multiple level signal processing operations --- in particular some operation that would contribute to our understanding of the implementation of the arithmetic structures of Section II.

The constraint of available devices forced us to consider simplistic structures. There are no available tapped delay line

elements so the direct arithmetic element could not be fashioned. Our approach then focused on understanding the mechanism involved in the permutation approach, especially the interface between the data register and the registers holding the permutation sequence. This requires an investigation of the comparator operation, which is the interface between the two.

The structure that was defined is shown in Figure 11. It could be considered a stand alone finite field arithmetic unit computing additions or multiplies in a given Galois Field but its implementation requires that it be viewed as a test vehicle for examining the comparison circuits and detection phenomena required in the permutation implementation of finite field arithmetic.

The unit again used the CCD-311 registers and initially was confined to multiplication in  $GF(5)$ . Subsequent simple changes could make it perform other arithmetic operations and also work in other base fields. This unit allowed us to investigate not only multiple level arithmetic but also the transient effects of loading a CCD, holding its contents and then reusing that information elsewhere, similar to what would be done in the permutation tapped delay line. This concept although mentioned frequently in the literature and by manufacturers has never been satisfactorily described nor have sufficient guidelines been developed for its use. It does impact significantly on the permutation arithmetic approaches.

A block diagram of the  $GF(5)$  finite field arithmetic unit is shown in Figure 11. A photograph of the unit is shown in Figure 12. Obviously, since this is a radical structure compared with common place LSI components, little help could be obtained in integrating many of the functions that are contained within the design. Consequently, many functions are developed from simple building blocks

IA-49,772

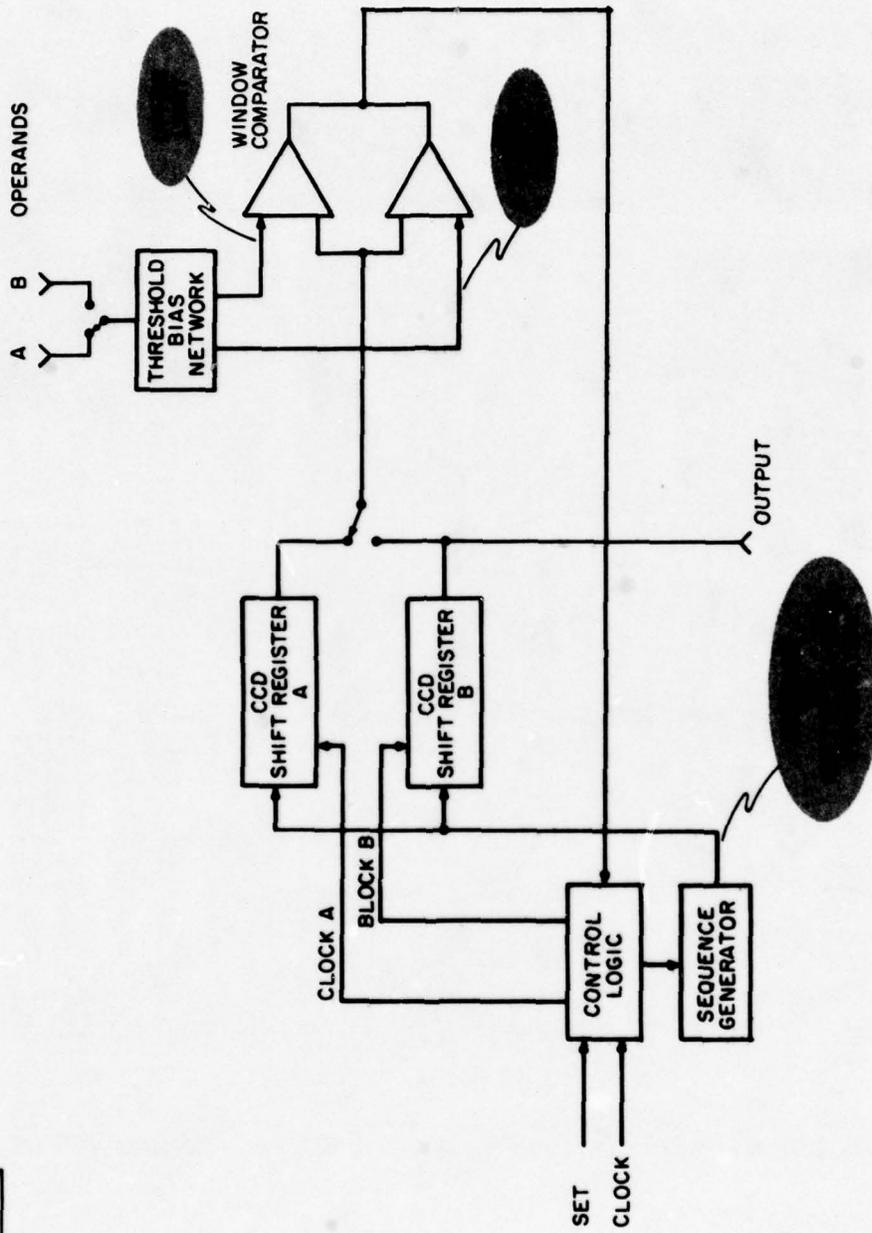


Figure 11 BLOCK DIAGRAM OF FINITE FIELD ARITHMETIC UNIT

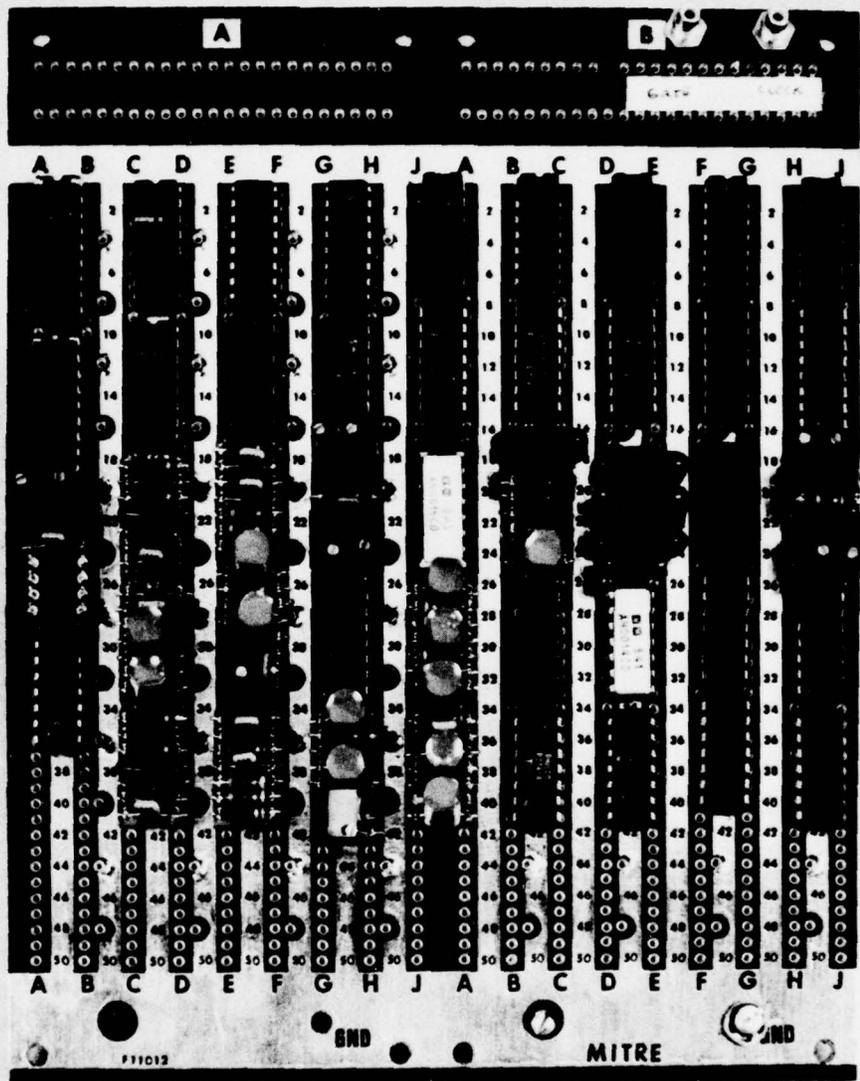


Figure 12 FINITE FIELD ARITHMETIC UNIT

such as gates, flip-flops and operational amplifiers. This results in a physically large structure.

The unit can functionally be broken into three parts, the sequence generator and control logic as one part, the shift registers as another, and the window comparator and bias network as a third part. The operation of the unit begins with the reception of a set pulse which enables the sequence generator and both shift registers. The generator output is one of the multi-level permutations discussed in Section II and is loaded into both shift registers. One of the operands, (A), is gated through to the window comparator, and compared with the contents of the last storage cell of shift register A. The voltage representing A is used to form an upper and lower window limit for the comparator. If the contents of the last cell of the shift register falls within this window, then it is assumed the cell's contents are equal to operand A. If no equality is formed, shift register A shifts its data one place and another comparison is made, continuing until an equality results. By definition, operand A is now stored in the last cell of shift register A.

During this time, shift register B has been holding with its initially loaded sequence. The two switches on the diagram are thrown to sense the last stage of shift register B and operand B. The same process as performed on shift register A is effected. This time, however, shift register A clocks along with shift register B. The result ( $A \times B$  or  $A + B$ ) is contained in the last cell of A when operand B and the last cell of shift register B are equal.

Notice that in this design no feedback of the output to the input is used. The elements of the field are loaded only from the sequence generator. This approach simplifies the design of the register since no regeneration circuits are required and provides

a common source for both registers. It also allows us to focus on the properties of the shift registers themselves and treat the feedback and regeneration as separate problems at a later time. Also, we have chosen to use one register for each operand, loading them all at the same time rather than using one register and reloading for each operand while storing the results in an accumulator. This seemed to be the most efficient use of time for two operands.

The normal operating mode was as a multiplier for GF(5). In this case the multiplication permutation of field elements was loaded into the register. This is the sequence 1, 2, 4, 3.

Because of the excessive delay line length and the resulting large inefficiency product, the input sequence was over sampled by a factor of 10:1 to reduce the signal bandwidth in the device. The CCD-311's were operated in the time multiplex mode (alternating samples) to simplify the clocking requirements.

Comparison of an operand with the output of a CCD register is initiated with a strobe pulse generated in the clocking logic. This strobe pulse is positioned to enable the comparator during one of the ten output samples of each discrete level. This strobing approach was necessary because of the large amount of clock feed-through that appears in CCD circuits. This type of approach should be carried over to future multiple-level comparator designs.

Figure 13 shows oscilloscope traces representing the various inputs to the comparator circuit. The two solid traces represent the upper and lower comparator window levels derived from an arithmetic operand voltage. These levels are spaced about 100 mv apart. The wider pulses above and between the window levels are samples from the charge-coupled device register representing the finite field elements. There are ten samples for each element and the elements of 3, 1, and 2 are visible. In order for an element to be detected

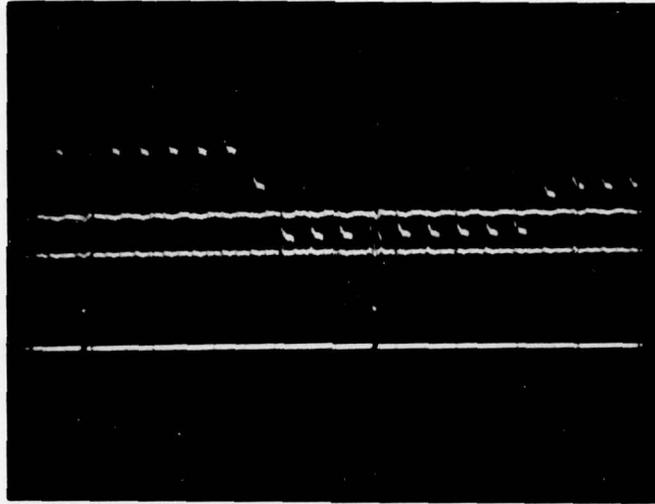


Figure 13. COMPARATOR INPUT SIGNALS

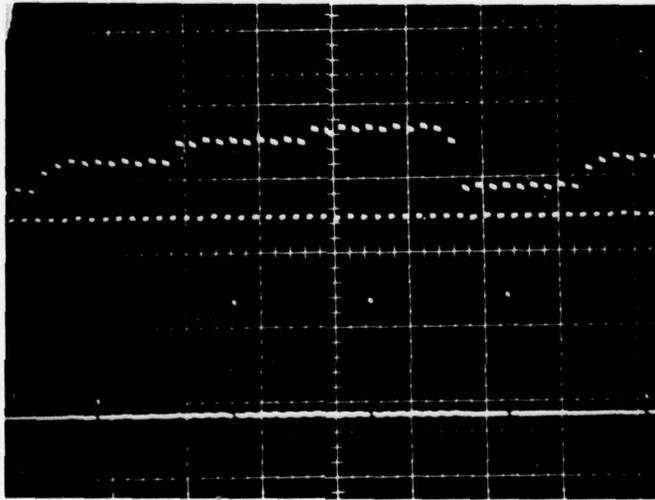


Figure 14. DETECTION PERFORMANCE

IA-50,987

it must appear within the comparator window. The window in this figure is set to match to the element 2 of the multiplication sequence 1, 2, 4, 3. There are, however, only nine samples within the window; one of the samples is actually the sample midway between the samples of the element three and the samples within the window. It is the ninth sample from the left-hand side of the picture. This distortion is an illustration of the effects of charge transfer efficiency caused either by a poor device or more likely in this case by the cumulative effects of a very long register. The oversampling approach used here can help alleviate this problem but at the expense of more clock cycles and increased delay. The strobe pulse used to enable the comparator circuit is shown on the bottom trace. It is positioned within the fifth sample of an element level to avoid the distortion effects caused by poor charge transfer.

An examination of the data samples at this strobe time shows an excessive amount of noise and transient oscillations. This is a major problem with the comparator integrated circuits used in the unit. For small element separations these oscillations can drive data sample values into the comparator voltage window causing a premature match and an erroneous arithmetic answer.

Figure 14 is a photograph of an over sampled sequence of the elements 1, 2, 3, and 4. This represents an addition permutation. The second trace shows the comparator output. The window levels were set to detect element 2 and this is accomplished. Element 2 has a low value during its strobe time indicating a detection while elements 1, 3, and 4 have high values indicating no detection.

In an actual permutation type tapped delay lines the register containing the permutation sequence will be required to hold its contents for some finite amount of time before the comparison takes place. This should put an extra burden on the comparison operation

because charge will build up in the device and increase the value of all output levels. If this operating condition is known a priori then a bias can be inserted to remove it. However, if this increase is not constant across the levels then the variance of each level is increased and detection performance will be poorer.

Attempts were made to store a sequence and then clock it out for comparison. These attempts were not successful due to the increased noise and a significant increase in distortion due to the AC coupling circuits used to set and remove CCD input bias.

The results of the laboratory effort seem to indicate that present technology can probably reliably carry and sense levels that are 50 mv apart. This would limit the CCD permutation approach to fields of 20 to 30 elements. No reasonable hard data can be obtained on the reliability of such arithmetic structures until devices are available that are better tailored to this application.

Also it seems that the integrated strobing comparators available today are only marginally suitable for this application. Their strobing response and noise parameters could be greatly improved and they must be implemented in a technology fully compatible with charge-coupled device fabrication techniques.

There are indications that independent CCD development programs are providing some of the solutions required. For instance, the University of Edinburgh is presently conducting research in the area of absolute matching of the voltage levels into and out of a charge-coupled device processing structure such as a tapped delay line or correlator.

Another indication is the impending development at RADC of a two dimensional analog memory. Research in this area aids the development of the complex clocking required for the arithmetic element.

## SECTION V

### CONCLUSIONS

Arithmetic in finite fields can be implemented in a number of ways. One method is shift and compare method of Section II. This approach has two major advantages: it requires less dynamic range than other approaches and is amenable to large scale integration techniques.

An analysis of the reliability of the implementation of such a shift and compare algorithm in charge-coupled device technology has shown that arithmetic with 20 to 30 symbols in present technology is reasonable.

The laboratory effort identified the following areas as requiring more research.

- Development of short, high quality charge-coupled device registers

- Development of strobing window comparators amenable to large scale integration

- Further investigation of accurate, absolute transfer of element levels throughout the structure.

Future work should be channeled into two areas. First, an investigation must be made of the amplitude error sources (such as CCD input bias and noise build-up), and their effect on the accuracy of the arithmetic. Techniques must be developed to cope with these error sources.

The second area should concentrate on the development of devices to support this effort. In particular, a short register of very low transfer loss should be built to aid in an actual measure-

ment of error probability. Also, a design of an integratable, strobed comparator should be undertaken.

The second area is well suited for research at an Air Force laboratory. The first area should be performed at MITRE in close cooperation with the research at RADC. Specific devices should be fabricated and detailed verification of the error rate performance should be completed.

## APPENDIX A

### CHARGE-COUPLED DEVICE PRINCIPLES

#### a. DESCRIPTION

Charge-coupled device technology is relatively new. Although experiencing tremendous growth in the past few years, the first devices were only described in 1970. Various applications in analog and digital signal processing, digital memory, and visible imaging are continually being expanded and improved. This appendix will provide a simple background description of the operation and important parameters of these devices.

A charge-coupled device is a functional solid-state electronic device which, under the application of a proper sequence of clock pulses, can move quantities of electric charge in a controlled manner across a semiconductor substrate. In essence, it is often viewed as an analog, sampled-data shift register.

The construction of a charge-coupled device evolves from the basic metal-oxide-semiconductor (MOS) configuration. The three elements, layered as shown in Figure A-1, under application of a positive voltage, form a capacitive storage area. Quantities of minority carriers can be confined to the area under the metal electrode.

It is customary to describe the storage capacity by plot of the interface (oxide-silicon) potential of the device as shown in Figure A-2. In Figure A-2, the initial distribution of the interface potential, after application of the voltage, is shown by the dashed line. As minority carriers accumulate near the surface of the bulk silicon, the potential at the interface decreases and the new potential profile in the presence of this charge is shown by the solid line in Figure A-2. The area between the two lines gives a pictorial representation of the amount of charge stored under the electrode. It is

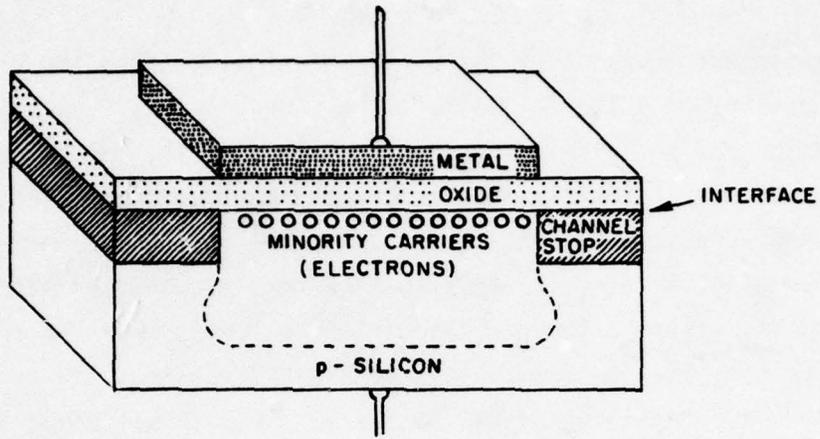
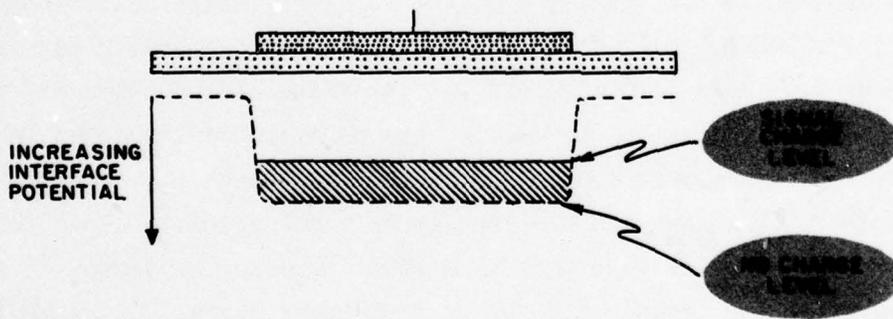


Figure A-1 CROSS SECTIONAL VIEW OF AN MOS CAPACITOR



1A-49,768

Figure A-2 POTENTIAL WELL DIAGRAM OF SINGLE CELL

analogous to water at the bottom of a well, thus, the often used description potential well. (The deeper the well, the larger the interface potential, the more storage capacity in the device.)

Realistically, this storage capacity is transient. If an electrode is pulsed to a high potential, a well is formed, and it immediately starts to fill with thermally generated minority carriers. After a time, the interface potential will be reduced and the well will be full. This time can be up to a few hundred seconds, depending on the purity of the bulk silicon material and the integrity of the fabrication process. Thus, the device operates in a thermal non-equilibrium mode; but for time intervals that are short compared to the relaxation time, it can store analog information represented by the amount of charge in the well.

This single MOS capacitor by itself would not be very useful, but hundreds of them placed close together can form a very important signal processing structure - a delay line. When two MOS capacitors are placed so close together that their potential well profiles overlap, any mobile minority charge will accumulate at the location with the highest interface potential. The interface potential is controlled by the gate voltages and can force minority carriers to move from well to well along the interface of the oxide and the bulk. Sets of metal electrodes are usually tied together in a periodic manner to control the motion of the charge, and several packets of charge can be transferred simultaneously along the delay line. The delay line becomes a shift register. Figure A-3 schematically portrays the transfer of charge from under metal electrodes labeled P1 to metal electrodes labeled P2. Each subdrawing refers to a time slot marked on the waveform diagram. These are the waveforms applied to the metal electrodes to create and control the potential

IA-49,767

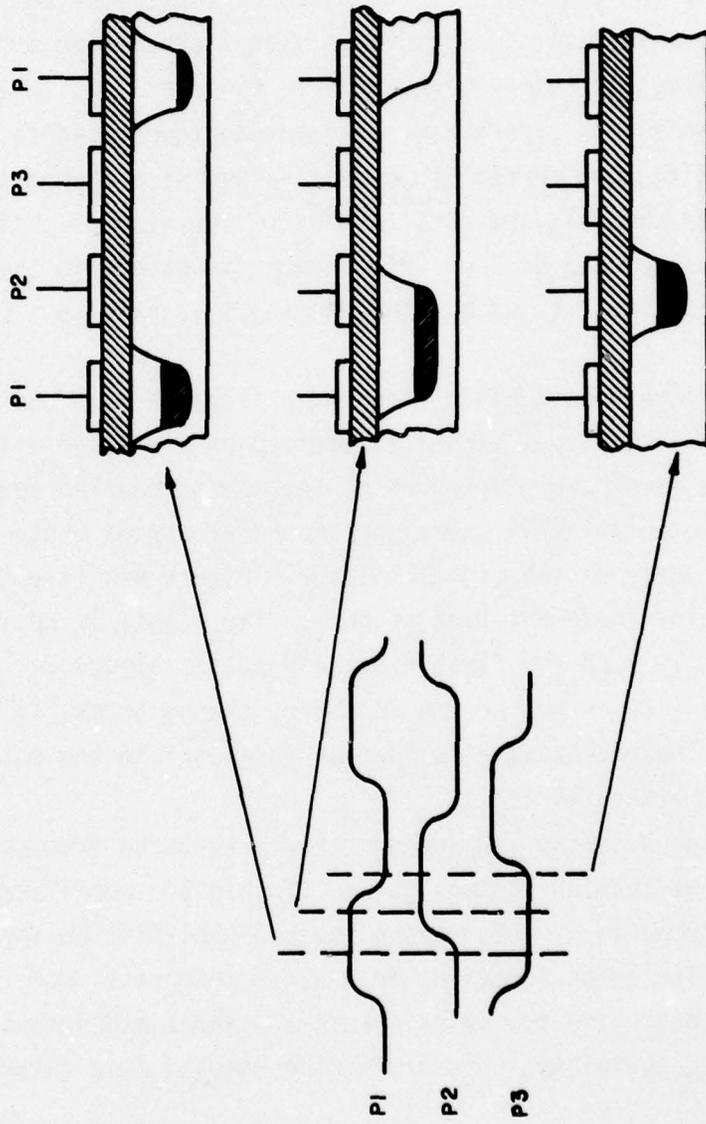


Figure A-3 THREE PHASE CCD CLOCK WAVEFORMS AND POTENTIAL WELL DIAGRAMS

wells and therefore the minority charge motion. Good performance is dictated by strong, smooth coupling between adjacent electrodes.

Some means of inserting and retrieving signal charge must be available for the device to be useful. Electrical charge injection is of the most importance in signal processing applications, but photo-electric charge generation is the most important in imaging applications. There are several techniques used to convert a signal voltage or current to an equivalent size charge packet in the transfer channel, and also to measure the size of these packets at the output of the device. Of primary importance in these structures is the linearity of mapping the input voltage to a packet of charge.

The method most often used today is based on a charge preset mode, whereby the input signal is applied as a voltage difference between the first two electrodes of the charge-coupled device to produce a potential well under the second electrode whose capacity depends linearly on the signal voltage. Figure A-4 illustrates one of the simpler implementation schemes. The signal is applied to the input gate (IG) and the first regularly pulsed electrode (P1). The input diode (ID) is pulsed low to inject charge across IG when P1 is turned on. The packet size is then proportional to the potential difference between P1 and IG.

Charge detection approaches have progressed from very early diode current sensing schemes using off-chip pre-amplifiers, to the present approaches of integrating the pre-amplifier on the chip. This amplifier often functions in a gated integrator mode which can produce outputs of a few volts. Both the input and output schemes use sampling techniques, preserving the sampled data nature of the CCD.

IA-60,978

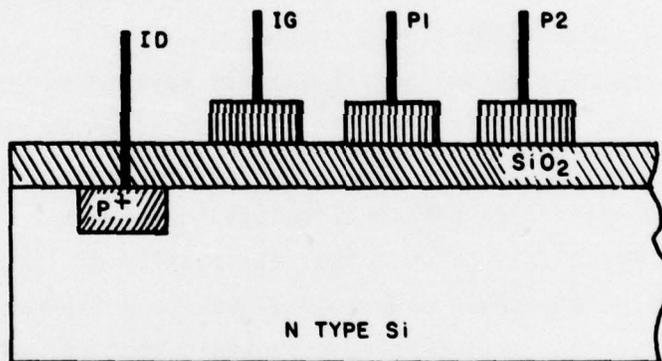


Figure A-4. Common Input Structure

An alternate output technique is to use a floating gate above the channel instead of a diode. This gate senses the size of the signal charge packet by its image charge above the gate. Since the gate electrode in this arrangement is isolated from direct contact with the signal charge and has no well-defined dc path to ground, its potential has to be controlled either by a reset switch or capacitively through a bias electrode. In principle, this approach offers very high sensitivity and signal-to-noise ratio. Additionally, this floating gate approach performs non-destructive sampling of the signal packet and therefore can be used in a number of places throughout the charge-coupled device to realize more complex signal processing structures.

b. PHYSICAL CHARACTERISTICS

Charge-coupled device performance is related to the physical properties within the device. Certain phenomena within the device tend to limit various operational parameters of the signal processing structures and affect the choice of design parameters.

A tradeoff exists between the desirability of large signals, which are easily detectable with good signal-to-noise ratio, and the large cell dimensions or operating potentials that are required for large charge packets. Basically, the amount of charge which can be handled by such devices is a function of the geometry and the applied clocking voltages. This charge storage capacity is of major concern in the multiple level storage designs discussed in this report.

In surface channel CCD's, the charge storage capability is proportional to the clock potential, the active area of the electrode, and the oxide capacitance. The limit in increasing the clock potential is set by the breakdown strength of the oxide layer. For typical values, the maximum charge density, which can be stored at the interface, is about  $10^{13}$  electrons per square centimeter.

The clock waveforms and the transfer electrode configuration also affect the signal handling capability. For instance, using sine waves instead of square waves reduces the signal handling capability to 75%.

However, the most important aspect of a charge-coupled device is its ability to maintain the integrity of the charge packets as they are transferred along the device. The transfer of charge from one cell to another is neither instantaneous nor complete. This non-ideal transfer limits both the speed and the allowable number of transfers of a device. In each transfer, a small amount of charge is left behind, adding to subsequent charge packets and causing a smearing effect on the charge packets.

A parameter called the transfer inefficiency,  $\epsilon$ , represents the fraction of charge not transferred per transfer. This parameter is used to form a common figure of merit by multiplying by the number of transfers in a device; the transfer inefficiency product is  $N\epsilon$ . Good performance is only obtained for inefficiency products much less than one. Thus, either the number of stages must be limited or strict requirements must be placed on the individual transfer loss.

Noise, from both intrinsic and extrinsic sources, appears superimposed on the signal and reduces the accuracy with which information in the device can be retrieved. The four noise sources which contribute to the CCD output are due to the input of signal charge, the output amplifier, dark current, and transfer noise. The first two sources are essentially external to the CCD channel, while the last two sources are intrinsic to the CCD operation.

The noise associated with electrical injection depends strongly on the input structure. Both voltage level and sampling time fluctuations contribute to the input noise source. In the charge preset method described earlier, the mean square thermal noise would be proportional to the capacitance of the input potential well, and thus

to the signal size. Analysis has shown that the noise introduced for a well-designed input structure is very low and good agreement has been obtained in practice (Ref. 6).

In charge-coupled devices, the charge at the detection point appears on a reverse biased diode or on a floating gate and thus essentially on a capacitance. In a properly designed system, the first pre-amplifier stage will determine the limiting noise value. Various techniques, many of them previously associated with nuclear electronic instrumentation, are now being applied to the output circuit design problem.

Charge-coupled devices operate in thermal non-equilibrium and generate carriers that will collect in the potential wells along with the signal charge. This produces a stationary pattern of noise on the signal charge. Obviously, this noise is sensitive to temperature and a rule of thumb says that the noise doubles for every  $8^{\circ}\text{C}$  increase in temperature.

The amount of charge left behind in the transfer process shows random fluctuations as well as a systematic dependence on transfer efficiency and signal size, and thus can be partially modeled as a noise source. Since the noise charge left behind from one cell accumulates in another packet, the noise voltages show a correlation between packets. This random fluctuation is due mainly to interface state noise where minority charge is retained in traps, not allowed to transfer, and finally re-emitted and transferred at a later time. The existence of these traps is due to impurities in the materials.

#### c. SIGNAL PROCESSING STRUCTURES

Analysis and design of CCD signal processing structures are similar to that of digital signal processing structures. Both structures represent discrete time systems. The signals are mathematically

represented as functions of a discrete independent variable, usually time, and have values only at specific times. The major difference lies in representing the signal amplitude; the digital signal amplitude takes on only discrete values; the CCD signal amplitude is represented by a continuous range of values. The majority of the digital signal processing knowledge, with the exception of those issues concerned with discrete amplitude or quantization, is directly applicable to CCD signal processing.

A single transfer in a CCD device can be represented as passing a signal sample through a delay line. The loss mechanism in the device adds an amplitude factor to the output signal. For a p-phase device with loss  $\epsilon$  per transfer, we normally define a single stage as a delay of  $T_c$  seconds with loss  $\bar{\epsilon} = p\epsilon$ . We can write the output of this delay line in response to passing a single signal sample through it of magnitude  $Q_0$  as a sequence of amplitude scaled, delayed unit sample functions:

$$S_0(nT_c) = Q_0 + (1 - \bar{\epsilon}) Q_0 \delta((n-1)T_c) + \\ (1-\bar{\epsilon})(\bar{\epsilon}) Q_0 \delta((n-2)T_c) + \\ (1-\bar{\epsilon})(\bar{\epsilon})^2 Q_0 \delta((n-3)T_c) + \dots$$

From this we can calculate the Z transform of the output function directly, obtaining:

$$S_0(Z) = (1 - \bar{\epsilon}) Q_0 Z^{-1} [1 + \bar{\epsilon} Z^{-1} + \bar{\epsilon}^2 Z^{-2} + \dots]$$

The infinite sum in the brackets can be put in closed form by use of the relation:

$$\sum_{n=0}^{\infty} a^n = \frac{1}{1-a}$$

where  $a = \bar{\epsilon} Z^{-1}$ , leaving

$$S_0(Z) = \frac{(1-\bar{\epsilon})}{1-\bar{\epsilon} Z^{-1}} Z^{-1} Q_0$$

The system function of the single stage delay line is then

$$H^1(Z) = \frac{(1-\bar{\epsilon}) Z^{-1}}{(1-\bar{\epsilon} Z^{-1})}$$

The system function has three basic components:  $Z^{-1}$ , the ideal delay;  $(1-\bar{\epsilon})$ , the transfer efficiency; and the denominator which represents the dispersive effect. This dispersive effect stems from this last term's low pass filter response.

For an N stage delay line, N of these single stage system functions are cascaded to represent the total delay line system function.

$$H^N(Z) = \left[ \frac{1-\bar{\epsilon}}{1-\bar{\epsilon} Z^{-1}} \right]^N Z^{-N}$$

An estimate of the amplitude distortion caused by the factor in brackets can be made by assuming a small value of  $\bar{\epsilon}$  and using an approximation for the bracketed term.

$$H^N(Z) \approx Z^{-N} \exp(N\bar{\epsilon}(Z^{-1} - 1))$$

The transfer inefficiency product is now seen explicitly in the system function. Its value in estimating performance is made more obvious by calculating the frequency response of the delay line, i.e.,

$$z^{-1} \rightarrow \exp(-2\pi j f T_c)$$

Ignoring the phase factors involved, an approximation of the amplitude distortion can be shown to be

$$H^N(f) = \exp(-N\bar{\epsilon}) \exp(N\bar{\epsilon} \cos 2\pi f T_c)$$

This function is plotted in Figure A-5. A simulation of the time output of an N stage delay line dramatically shows the dispersive effect in Figure A-6.

In much the same way, more complex signal processing structures can be developed and analyzed. Of particular importance to us is the transversal filter or tapped delay which occurs naturally as an error correction coding processing structure. Figure A-7a shows the structure of a tapped delay line processor. The structure is simply a delay line with taps nondestructively sampling the signal in each stage, multiplying it by a weight, and summing the results. The equation relating the input and output in such a processor is:

$$v_o(nT) = \sum_{k=1}^N h_k v_{in}(n-k) T_c$$

where  $h_k$  represents the weights.

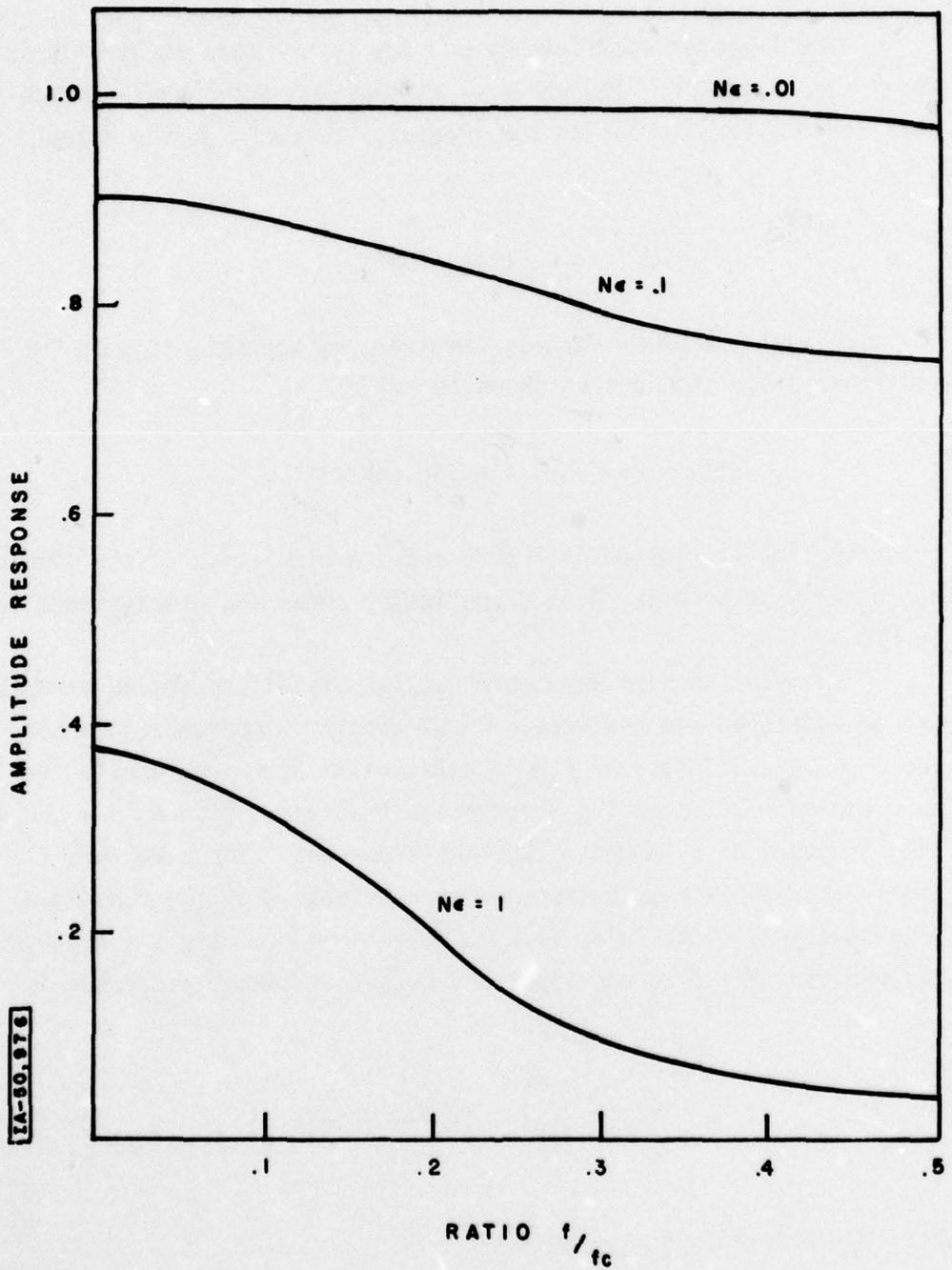


Figure A-5 AMPLITUDE RESPONSE OF N TRANSFER CCD WITH LOSS PER TRANSFER  $\epsilon$  VS FREQUENCY

IA-50,977

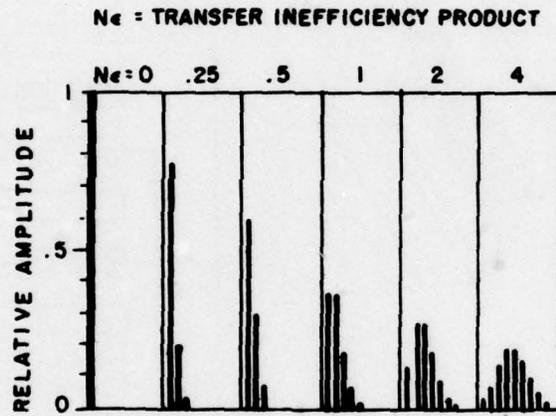


Figure A-6. Simulation of Time Dispersion in CCD with Loss per Transfer  $\epsilon$

1A-49,769

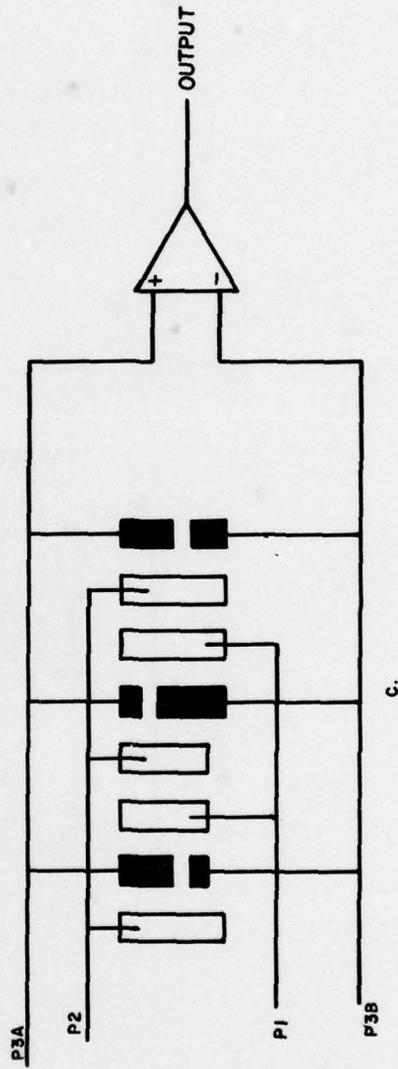
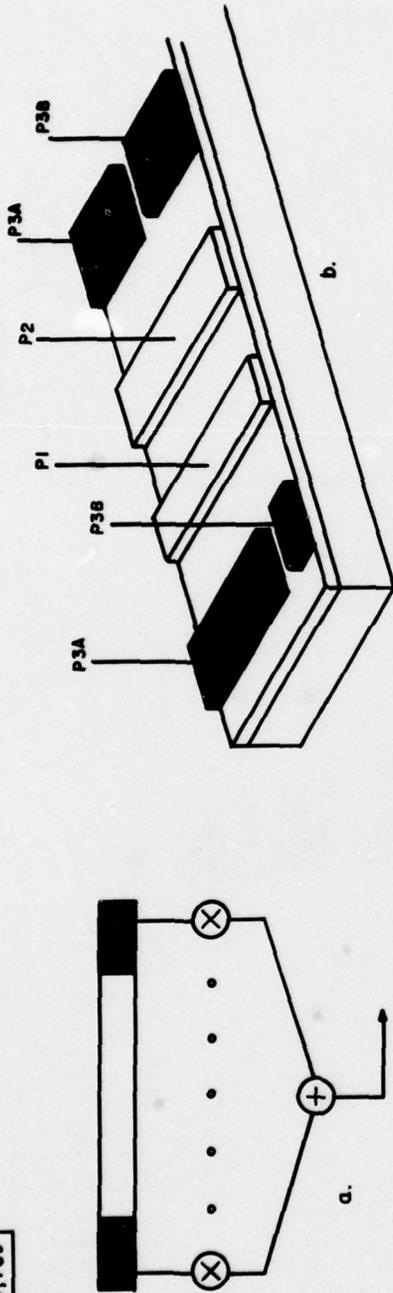


Figure A-7 SPLIT ELECTRODE TRANSVERSAL FILTER

The delay line portion is, of course, a natural structure for charge-coupled devices, but more importantly the technology has matured to the point where the tap weight multiplication and the summation are readily achievable within the transfer electrode structure, thus effecting a small but potentially powerful processing structure.

The manner of achieving the weighting and summing structure is called the split electrode or electrode weighting technique. The technique is illustrated in Figure 7b and 7c.

If all the electrodes in a certain phase, say the third phase, of a CCD delay line are selected and split into two halves, and all the halves on one side of the device are tied together, then the clock current in each side when the signal charge is being transferred into that phase has a component proportional to the signal charge and the area of the electrode. Thus, if the split in the electrodes of the third phase is chosen to have an area on one side proportional to one plus the desired weight and on the other side one minus the desired weight, by summing the currents on each side and subtracting these in a differential amplifier we achieve an output proportional to a sum of the weights times the signal, i.e., the transversal filter characteristic. Indeed, by properly choosing the splits we can achieve positive, negative or zero weights. The accuracy of the tap weights is determined by the tolerances of photolithographic techniques and typically results in weighting errors of about 1%.

Split electrode tapped delay line filters have been built with lengths up to 800 stages and these devices have been self-contained with clock drivers, logic and output amplifier all on the chip. Obviously, this kind of computational power in such a small package can benefit many signal processing applications including error control coding.

#### REFERENCES

1. D. D. Buss and M. F. Tompsett, ed., "Forward-Joint Special Issue on Charge-Transfer Devices," IEEE Journal of Solid State Circuits, Vol. SC-11, No. 1, (February, 1976).
2. T. A. Zimmerman and C. S. Miller, "Charge-Coupled Devices in Signal Processing Systems," Vol. 1, AD-782-574, (Redondo Beach, California,: TRW Systems Group).
3. W. W. Peterson and E. J. Weldon, Jr., Error-Correction Codes, 2d ed. M.I.T. Press, (Cambridge, Massachusetts: M.I.T. Press 1972).
4. J. F. Albarran and D. A. Hodges, "A Charge Transfer Multiplying Digital-to-Analog Converter," 1976 IEEE International Solid-State Circuits, pp 202-203.
5. C. H. Sequin and A. M. Moshen, "Linearity of Electrical Charge Injection Into Charge-Coupled Devices," IEEE Journal of Solid State Circuits, Vol. SC-10, No. 2, (April, 1975), pp 81-92.
6. R. W. Brodersen and S. P. Emmons, "Noise in Buried Channel Charge-Coupled Devices," IEEE Journal of Solid State Circuits, Vol. SC-11, No. 1, (February, 1976), pp 147-155.
7. C. H. Sequin and M. F. Tompsett, "Charge Transfer Devices", suppl. 8 for Advances in Electronics and Electron Physics, (New York: Academic, 1975).