

Copyright © 1975, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**EFFICIENT RETRIEVAL IN RELATIONAL  
DATA BASE SYSTEMS**

by

**Robert Mark Pecherer**

**Memorandum No. ERL-M547**

**2 October 1975**

**ELECTRONICS RESEARCH LABORATORY**

**College of Engineering  
University of California, Berkeley  
94720**

Efficient Retrieval in Relational Data Base Systems

By

Robert Mark Pecherer

B.S. (University of Michigan) 1967  
M.S. (University of Wisconsin) 1969

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Engineering Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Approved:

.....  
*Michael J. Horne*

.....  
*Ralph M. Kenzie*

.....  
*Luene Wang*

.....  
*[Signature]*  
Committee in Charge

.....

# Efficient Retrieval in Relational Data Base Systems

Ph.D.

Robert Mark Pecherer

Dept. of Electrical  
Engineering and  
Computer Sciences

## ABSTRACT

The retrieval of data from a relational data base is regarded as the explicit formation of the relation defined by an expression in a relational algebra over the data base relations. For a conventional digital computer and a simple, uniform storage representation for relations, a variety of techniques are presented for performing these evaluations efficiently.

Among a large set of operators which map relations to relations that have been proposed, a subset of four operators is shown to provide sufficient relation-defining capability to be considered as a retrieval language. The efficient evaluation of all expressions over these operators and any fixed set of data base relations is explored. Algorithms for the application of each operator are provided whose running times are asymptotically as good or better than previously known solutions. In addition, conditions are described for achieving further speed-up for two of the operators.

Procedures for each of the four operators permit the evaluation of any such algebraic expression through recursive application of operators to data base relations and the intermediate results of the computation. A collection of transformation rules are derived which translate an expression in the algebra to a second expression which defines the same relation independent of the data base. The second expression can usually be evaluated in less time than the first by this recursive evaluation.



Further results are presented for two sets of expressions. For the first set, a correspondence is established between an expression and a class of iterative procedures which evaluate the expression without the use of intermediate relations. Algorithms are given to select that procedure whose running time is minimal or expected minimal within any class. For the second set, it is shown that careful attention to the formation of intermediate results in the evaluation process can produce significant savings in time.

Signature

  
Chairman of Committee

## TABLE OF CONTENTS

	page
ACKNOWLEDGEMENT	1
I. INTRODUCTION	1
II. RELATIONAL ALGEBRAS	10
III. IMPLEMENTATION OF AN *-EVALUATOR	34
IV. EFFICIENCY BY TRANSLATION	44
V. EFFICIENT EVALUATION OF CRA EXPRESSIONS	73
VI. SUMMARY AND FUTURE WORK	80
REFERENCES	84

## Acknowledgement

I am deeply indebted to Professors Michael Stonebraker and Eugene Wong of the University of California, Berkeley for their advice, support and encouragement during the research and writing of this thesis. Also to Professor Ralph McKenzie for reading the manuscript and to Doris Simpson for typing it.

Special thanks to my parents, Benjamin and Miriam and brothers, Michael and Alan for their patience and understanding. And to my wife, Angie for making all of this possible.

Finally, many thanks to friends, fellow students and faculty at Berkeley for creating a most stimulating environment in which to live, work and grow.

Research reported here was sponsored in part by Naval Electronics System Command Contract N00039-75-C-0034.

## Chapter I. INTRODUCTION

### 1.1. Problem Description

A data base system is a computer facility for the storage and retrieval of data from a collection of data called a data base. When the logical representation of the data base conforms to the relational model of data [ 1 ], it is referred to as a relational data base and the system is termed a relational data base system. The logical representation of data in the relational model is as a set of time-varying relations of assorted degrees. (The reader is assumed to be familiar with the concepts and terminology of data base systems; recent texts by Date [ 2 ] and Martin [ 3 ] provide the necessary background material. Formal definitions for relations, etc. appear in 1.4.)

Users of a relational data base system specify interactions to the system (in some language) for the retrieval of data from a relational data base and for the addition, deletion and modification of data in the data base. This thesis is directed to the specific problem of constructing relational data base systems which satisfy user interaction rapidly, particularly for retrievals. Each such interaction will be regarded as a query and a response: a query (supplied by the user) defines a relation from the relational data base and the response (generated by the system) is the relation the query defines.

Additions, deletions and modifications (updates) are not considered. Deletions and modifications often require a retrieval prior to alteration, so that the ability to satisfy retrievals quickly contributes to rapidly satisfying these types of interactions as well.

Alternate approaches to logical data representation such as the

hierarchical view (exemplified by IBM's IMS [ 4 ]) and the network view (typified by the system proposed by the CODASYL Data Base Task Group [ 5 ]) are not discussed. Also excluded are issues pertaining to multiple users, multiple views of the data base and security and integrity constraints.

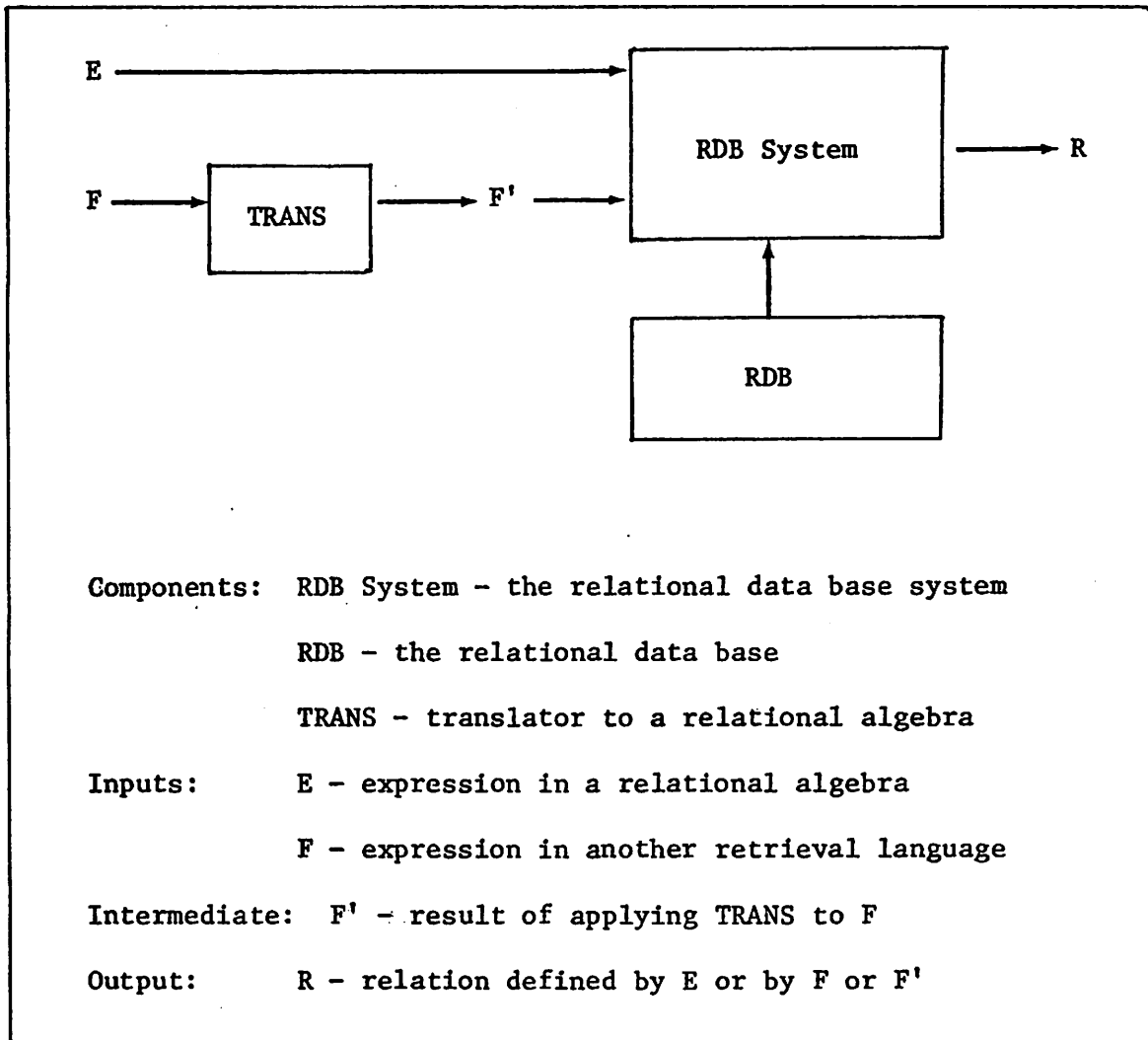
The problem is taken as the efficient evaluation of expressions in a relational algebra over the data base relations. These algebras have been shown to provide extensive relation-defining capabilities [ 6 ], hence they can serve directly as retrieval languages or as interfaces between some other retrieval language (such as ALPHA [ 7 ], SEQUEL [ 8 ], SQUARE [ 9 ], DAMAS [10] and QUEL [11]) and the relational data base system. Figure 1.1 illustrates both uses.

The study begins with the selection of an appropriate relational algebra for data retrieval and then proceeds to derive efficiency techniques for such a system.

## 1.2. Thesis Organization

The thesis consists of six chapters. Chapter I is introductory, serving to describe the problem and the thesis organization, survey previous results and provide notation and terminology for relations.

Chapter II covers relational algebras as retrieval languages. A logical implementation framework is discussed in greater detail using familiar algebraic concepts specific to relational algebras. It is shown that any set of relational operators (e.g. which map relations to relations) defines a set of relations that can be specified (hence retrieved) within such a system. Ten relational operators are considered and a subset of four operators is selected for further study. Extensive examples are provided to demonstrate that the relation-defining



**Fig. 1.1.** Use of a Relational Algebra for Retrieval Specification and as an Interface

capability of the smaller operator set is sufficient for many data retrieval applications.

Chapters III, IV and V investigate procedures for rapid evaluation of algebraic expressions over any set of data base relations and the operators selected in Chapter II. The introductory material of Chapter III describes a hypothetical computer system for implementation, a uniform storage representation for relations and a measure for the running time of procedures which execute in this system environment. These remain in effect throughout III, IV and V.

Additional material in Chapter III derives procedures for the implementation of the relational operators that are as fast or faster than previously described solutions. Conditions are identified under which further speed-up can be achieved for two of these operators.

Chapter IV illustrates how a set of procedures for the operators permit the evaluation of any expression over the data base relations and the operators. A set of transformation rules are presented which translate such an expression to a second expression that defines the same relation. For most of these rules, the second expression can be evaluated at least as fast as the first and usually faster. In addition, a class of expressions is given, and it is shown that any member of the class can be evaluated by nested iteration over the operand relations in any order. Algorithms are described for selecting the order of iteration that is optimal or expected optimal with respect to the measure.

Chapter V presents efficiency techniques for a large and important class of relational algebraic expressions. The class is generated by applying the Codd Reduction Algorithm (CRA) to the set of all relation-defining expressions in the nonalgebraic retrieval language ALPHA. The

CRA translates an ALPHA expression to an expression in a relational algebra that defines the same relation. The class of expressions that results is important in a system which supports ALPHA by using the CRA as a translator as in Figure 1.1. Efficiency is gained by achieving the conditions for operator speed-up (of Chapter III) in the intermediate results of the evaluation. Minor conflicts with two efficiency techniques of Chapter IV are resolved.

Chapter VI is a summary of the thesis and a description of areas in which extension of the work described here is indicated.

### 1.3. Survey of Previous Results

The relational model of data was introduced by E. F. Codd [1]. In subsequent papers [6,7], Codd defines a relational algebra and a relational calculus for manipulating relations, and the non-procedural retrieval language ALPHA based on the relational calculus. An algorithm is given that translates an expression in ALPHA to an expression in the relational algebra over the data base relations that defines the same relation.

The algorithm is discussed in Palermo [12] where it is referred to as the "Codd Reduction Algorithm" (CRA). Viewing the CRA as an interface between users specifying retrievals in ALPHA and a system which evaluates expressions in Codd's relational algebra (as in Figure 1.1), Palermo shows a number of ways to improve the algorithm for faster evaluation. The direction is towards minimizing the volume of I/O transfers in an implemented system. Palermo shows that many data values in the operand relations which are not required for the computation or for the response relation can be discarded almost immediately, thus freeing the primary memory for other purposes. In addition, his use of semi-joins and



indirect joins guarantees that no data value is retrieved more than once in the course of the evaluation. The research reported in this thesis was stimulated in large part by Palermo's results.

Rothnie [10] investigates efficient implementation techniques for a relational data base system supporting his DAMAS retrieval language. In addition to a thorough analysis of various search problems for a data base, Rothnie describes a storage and access method (termed "multiple key hashing") and develops heuristics for the formation of response relations that reduce page faults in a virtual memory environment. The utility of multiple key hashing and the heuristics is demonstrated by experimental results with an implementation of a subset of DAMAS called DAMASjr. The heuristics prove to be beneficial for the simple retrievals allowed in DAMASjr, but generalization to the full defining capability of DAMAS is not attempted.

Smith and Chang [13] describe an interface between Codd's relational algebra and a data-driven relational data management machine that utilizes a number of techniques for performance optimization. That interface (called SQUIRAL for "Smart QUery Interface for A Relational Algebra") seeks to minimize response time through efficiency transformations, coordination or sort orders for relations, maintaining storage locality, use of directories and exploiting disjoint and pipelined concurrency. Five of the eleven transformation rules presented in Chapter IV were independently discovered by those authors and are identified in the text.

Held and Stonebraker [14] investigate a wide variety of storage structures for the computer representation of the data base relations. This work is also directed towards optimizing the performance of a relational data base system, taking into account interactions which

alter the data base as well as the retrieval interactions considered here.

It should be noted that in addition to Rothnie's DAMASjr [10] and an APL implementation of Codd's relational algebra by Palermo [12], three prototype relational data base systems are under development: the ZETA Project at Toronto [15], the INGRES Project at Berkeley [11], and IBM's System R [16]. Information concerning efficiency in these systems has not as yet been published.

#### 1.4. Definitions, Terminology and Notation

The definitions, terminology and notation of this section follow conventional usage as established by Codd [1,6,7].

(1.2) Definition: Let  $D_1, \dots, D_n$  be sets, not necessarily distinct.

A relation  $R$  on  $D_1, \dots, D_n$  is a subset of the Cartesian product

$$D_1 \times D_2 \times \dots \times D_n.$$

A relation  $R$  on  $n$  sets  $D_1, \dots, D_n$  is frequently called an  $n$ -ary relation or relation of degree  $n$ . The degree of relation  $R$  is denoted " $\text{deg}(R)$ ."  $D_j$  is called the  $j$ th underlying domain of  $R$ . An element  $r$  of  $n$ -ary relation  $R$  is called an  $n$ -tuple or simply tuple if no confusion results. The size of a relation is the number of tuples it contains. A tuple variable for  $R$  is a variable that assumes the values of the tuples of  $R$ .

(1.3) Definition: Let  $R$  be a relation over sets  $D_1, \dots, D_n$ . The  $j$ th

domain  $D_j$  of  $R$  is simple if no member of  $D_j$  is itself a set.

(1.4) Definition: A relation  $R$  is first-normal or simply normal if every domain of  $R$  is simple.

The only relations considered in this thesis are normal.

(1.5) Definition: Let  $R$  be an  $n$ -ary relation and  $r$  an  $n$ -tuple of  $R$ . For  $1 \leq i \leq n$ , " $r[i]$ " denotes the  $i$ th attribute of  $r$ , the value that occurs in the  $i$ th domain of  $R$ .

(1.6) Definition: Let  $A = a_1, \dots, a_k$  be a list of integers such that for  $j = 1, \dots, k$  we have  $1 \leq a_j \leq n$ .  $A$  is said to be a domain-identifying list for any relation  $R$  of degree greater than or equal to  $n$ . The length of  $A$  is  $k$ .

If  $R$  is a relation of degree  $n$  and  $A = a_1, \dots, a_k$  a domain-identifying list for  $R$ , " $r[A]$ " designates the  $k$ -tuple  $(r[a_1], \dots, r[a_k])$  for  $r \in R$ .

(1.7) Definition: Let  $A$  be a domain-identifying list for  $R$  without repetition. (E.g.  $A$  contains no duplicate values.) Then " $\bar{A}$ " denotes the domain-identifying list complementary to  $A$  relative to  $R$ .  $\bar{A}$  is a domain-identifying list for  $R$  containing every integer between 1 and  $\text{deg}(R)$  that does not occur in  $A$ . The integers in  $\bar{A}$  appear from smallest to largest.

(1.8) Example: For the 5-ary relation  $R$  and domain-identifying list  $A = 1, 4, 3$ . The domain-identifying list complementary to  $A$  relative to  $R$  is  $\bar{A} = 2, 5$ .

Let  $r$  be an  $n$ -tuple and  $s$  an  $m$ -tuple. The concatention of  $r$  with

$s$  (in that order) is denoted "rs" and designates the  $n+m$ -tuple

$(r[1], \dots, r[n], s[1], \dots, s[m])$ .

## Chapter II. RELATIONAL ALGEBRAS

### 2.1. Introduction

The retrieval of data from a relational data base was described in Chapter I in terms of a query and a response, the query being a relation-defining expression in some retrieval language, the response being the relation defined by the query relative to the data base. In this chapter, a logical framework is described in which a relational algebra over the data base relations serves as a retrieval language. A particular algebra is chosen, and the remainder of the thesis presents techniques for efficiently evaluating the expression of this algebra within the given framework.

An algebra\* is a set of objects  $T$  together with a set of operators  $\Omega$  that map objects in  $T$  to objects in  $T$ . When the objects of  $T$  are relations and  $\Omega$  is a set of relational operators (e.g. which map relations to relations), the algebra is called a "relational algebra."

A set  $S$  of relations and a set  $\Omega$  of relational operators give rise to a relational algebra that contains  $S$  and every relation obtainable from  $S$  by applying the operators of  $\Omega$  repeatedly. Thus any syntactically well-formed expression over  $S$  and  $\Omega$  is the definition of a relation in the relational algebra induced. In this sense, a relational algebra can be thought of as a retrieval language.

In terms of data base systems, any collection of procedures which maps every expression over data base relations  $S$  and operators  $\Omega$  to the

---

\*Formal definitions for the terminology of this section appear in 2.2.

relation it defines can serve as a mechanism for data retrieval from a relational data base. We term a collection of procedures that implements this map an  $\Omega$ -EVALUATOR. The logical framework for this mechanism is illustrated in Figure 2.1 in which the expression  $F$  over  $S$  and  $\Omega$  is the query,  $DB$  is a stored representation of  $S$  (the data base relations), and the response  $R$  is the relation defined by  $F$  relative to  $DB$ .

This approach was proposed by Codd [6]; in [1, 6], Codd defined a large set of relational operators and showed they provide an extensive capability for defining relations from an existing set of relations. Relational operators are studied in this chapter with the intent of selecting a small set with a relation-defining capability sufficient for many data retrieval applications.

The chapter is organized as follows: Section 2.2 provides formal definitions for the concepts and terminology discussed in the Introduction (2.1). In 2.3, three relational operators (JOIN, PROJECTION and DIVISION) are defined and examples are provided to demonstrate their relation-defining capability. Section 2.4 provides useful definitions for comparing relational operators. Seven more operators (PRODUCT,  $\theta$ -JOIN, RESTRICTION,  $\theta$ -RESTRICTION, INTERSECTION, RELATIVE COMPLEMENT and UNION) are defined and it is shown that all operators except UNION can be defined in terms of the others. These results are used in 2.5 to identify 3 operator sets with relation-defining capability equal to the operator set in 2.3. In 2.6, one operator set is selected for further study. Some limitations of the general approach are discussed in 2.7. Section 2.8 is a summary.

All operators discussed here except JOIN and RESTRICTION were defined by Codd [1, 6], these being generalizations of Codd's  $\theta$ -JOIN and

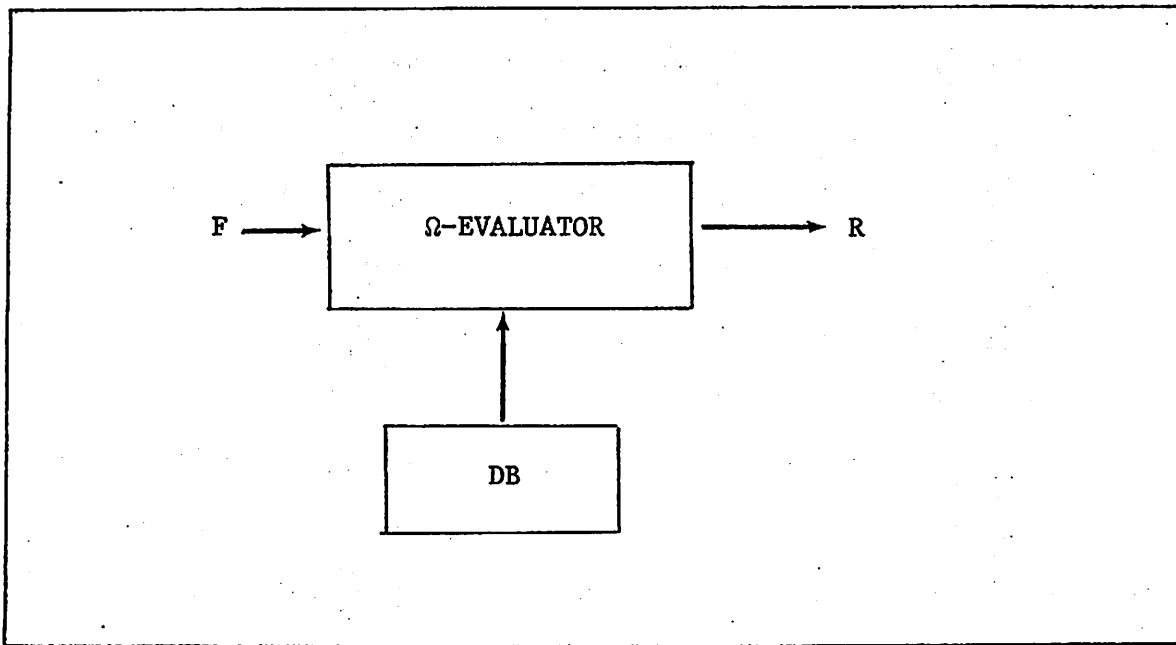


Fig. 2.1. Implementation Framework.

$\theta$ -RESTRICTION. The construction in the proofs of Propositions 2.38, 2.50, and 2.52 are also due to Codd.

## 2.2. Definitions and Terminology

(2.2) Definition: An algebra is a 2-tuple  $(T, \Omega)$  where  $T$  is a set and  $\Omega$  a set of operators which map objects in  $T$  to objects in  $T$ .

(2.3) Definition: A relational operator is a function which maps an ordered set of  $k$  relations to a relation. A relational operator is unary if  $k=1$ , binary if  $k=2$ , and in general  $n$ -ary if  $k=n$ .

The only relational operators considered are unary and binary. The syntax for relational operators is described implicitly in their definitions (see below). A relational expression over relations  $S$  and operators  $\Omega$  is a syntactically well-formed expression over  $S$  and  $\Omega$ . The set of all such expressions over  $S$  and  $\Omega$  is designated  $\xi(S, \Omega)$ .

(2.4) Definition: A relational algebra is an algebra  $(T, \Omega)$  where  $T$  is a set of relations and  $\Omega$  a set of relational operators.

(2.5) Definition: Let  $S$  be a set of relations and  $\Omega$  a set of relational operators. The relational closure of  $S$  under  $\Omega$ , designated  $C(S, \Omega)$ , is the smallest set of relations containing  $S$  and closed under the operators of  $\Omega$ .

(2.6) Definition: Let  $(T, \Omega)$  be a relational algebra. A relational generator for  $(T, \Omega)$  is a subset  $S$  of  $T$  for which  $C(S, \Omega) = T$ .



The concepts of "relational closure" and "relational generator" are the familiar algebraic concepts of "closure" and "generator" applied to relational algebras. Note that for any set  $S$  of relations and any set  $\Omega$  of relational operators,  $(C(S, \Omega), \Omega)$  is a relational algebra for which  $S$  is a relational generator. This algebra is said to be induced by  $\Omega$  over  $S$ .

(2.7) Definition: An  $\Omega$ -EVALUATOR is a collection of procedures which accepts as inputs any  $F \in \xi(S, \Omega)$  and outputs the relation  $R \in C(S, \Omega)$  defined by  $R$ .

Essentially, an  $\Omega$ -EVALUATOR implements a well-defined map from relational expressions over  $S$  and  $\Omega$  to relations in the algebra induced by  $\Omega$  over  $S$ .  $\xi(S, \Omega)$  and  $C(S, \Omega)$  are implicitly defined by  $S$  and  $\Omega$ . So that for a relational data base containing relations  $S$ , the choice of the operator set  $\Omega$  is critical to the value of the data retrieval function served by the  $\Omega$ -EVALUATOR in the framework of Figure 2.1. That is, the choice of  $\Omega$  determines the set of relations that can be retrieved.

The next section defines an operator set  $\Omega_0$  and provides examples to demonstrate the types of queries an  $\Omega_0$ -EVALUATOR can satisfy for a given relational data base.

## 2.3. A Relational Algebra

### 2.3.1. The Operator Set $\Omega_0$

(2.8) Definition: Let  $R_i$  and  $R_j$  be (not necessarily distinct) relations where  $\text{deg}(R_i) = n$  and  $\text{deg}(R_j) = m$ . Let  $r$  and  $s$  be distinct tuple variables ranging respectively over

$R_i$  and  $R_j$ . Let  $f = f(r,s) = f(r[1], \dots, r[n], s[1], \dots, s[m])$  be a 0-1 function of  $r$  and  $s$  where " $r[k]$ " stands for the  $k$ th attribute value of tuple variable  $r$  (see 1.4). The f-JOIN of  $R_i$  and  $R_j$  is denoted by  $R_i[f]R_j$  and is defined as:

$$R_i[f]R_j = \{rs : r \in R_i \wedge s \in R_j \wedge f(r,s) = 1\}.$$

(2.9) Remark:  $R_i[f]R_j$  is a subset of  $\{rs : r \in R_i \wedge s \in R_j\}$  for which  $f$  is a characteristic function. To implement JOIN on a computer, we require that  $f$  be a total recursive function of  $r$  and  $s$ . In addition, we require that  $f$  not involve quantification over  $R_i$ ,  $R_j$  or any other relation. This insures that for  $r$  in  $R_i$  and  $s$  in  $R_j$ , the inclusion of  $rs$  in  $R_i[f]R_j$  can be determined without further access to the data base.

(2.10) Examples of JOIN

$R_i$	<u>1</u>	<u>2</u>	<u>3</u>
	A	1	2
	B	1	3
	C	2	3

$R_j$	<u>1</u>	<u>2</u>
	2	A
	3	A

$R_i[r[1] = s[2]] R_j$

<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>
A	1	2	2	A
A	1	2	3	A

$R_i[r[2] < s[1]] R_j$

<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>
A	1	2	2	A
B	1	3	2	A
A	1	2	3	A
B	1	3	3	A
C	2	3	3	A

$$R_j[r[1] = s[3] \wedge r[2] = s[1]] R_i$$

<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>
2	A	A	1	2

(2.11) Definition: Let  $R_i$  be a relation and  $L$  a domain-identifying list for  $R_i$ . The  $L$ -PROJECTION of  $R_i$  is denoted by  $\pi_L(R_i) = \{r[L] : r \in R_i\}$ .  $L$  is termed the projection list of  $\pi_L(R_i)$ .

(2.12) Remark: In later chapters it becomes necessary to distinguish between a PROJECTION which permutes the domains of the operand relation and one which actually eliminates domains. The former will be referred to as a permuting projection<sup>\*</sup> and the latter as a proper projection. (Projection lists which are domain-identifying lists with repetition are not excluded; however they are of no apparent value for data retrieval and are never considered.)

(2.13) Examples of PROJECTION

$R_i$	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>		$\pi_1(R_i)$	<u>1</u>		
	S	30	M	SF			S		
	J	35	M	LA			J		
	J	45	F	LA			D		
	D	45	F	SJ					
	$\pi_{3,1}(R_i)$	<u>1</u>	<u>2</u>		$\hat{\pi}_{4,3,2,1}(R_i)$	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
		M	S		SF	M	30	S	
		M	J		LA	M	35	J	
		F	J		LA	F	45	J	
		F	D		SJ	F	45	D	

<sup>\*</sup> $\pi_L(R_i)$  will be denoted  $\hat{\pi}_L(R_i)$  if known to be an permuting projection.

(2.14) Definition: Let  $R_i$  and  $R_j$  be relations of degree  $n$  and  $m$  respectively. Let  $A$  and  $B$  be domain-identifying lists for  $R_i$  and  $R_j$  respectively, both of length  $k < n$  and both without repetition. Let  $\bar{A}$  be the domain-identifying list complementary to  $A$  relative to  $R_i$ . The DIVISION of  $R_i$  on domains  $A$  by  $R_j$  on domains  $B$  is denoted by  $R_i[A \div B]R_j$  and defined as:

$$R_i[A \div B]R_j = \{r[\bar{A}] : r \in R_i \wedge \forall s \in R_j \exists t \in R_i (r[\bar{A}] = t[\bar{A}] \wedge t[A] = s[B])\}.$$

(2.15) EXAMPLES OF DIVISION

$R_i$	<u>1</u>	<u>2</u>	<u>3</u>
	X	A	25
	X	A	26
	Y	A	26
	Y	B	3
	X	B	3

$R_j$	<u>1</u>	<u>2</u>	$R_k$	<u>1</u>
	A	25		A
	B	3		

$$R_i[2 \div 1, 2] R_j \quad \underline{1}$$

X

$$R_i[2 \div 1] R_j = \phi$$

$$R_i[2 \div 1] R_k \quad \underline{1} \quad \underline{2}$$

X 25  
X 26  
Y 26

$$\pi_{1,2}(R_i)[2 \div 1] R_j \quad \underline{1}$$

X  
Y

Codd [6] has pointed out that PROJECTION and DIVISION provide an algebraic counterpart to existential and universal quantification in a first-order predicate calculus. Section 2.3.2 provides extensive examples

for a hypothetical set of relations that illustrate this correspondence and show that the operator set  $\Omega_0$  possesses sufficient relation-defining capability to be considered as a retrieval language.

### 2.3.2. Examples of the Use of $\Omega_0$ for Defining Relations

#### 2.3.2.1. A Hypothetical Set of Data Base Relations

Suppose that a set S consists of 4 relations  $R_1, \dots, R_4$  of degrees 3, 2, 3 and 3 respectively with the following domains:

<u>relation name</u>	<u>1</u>	<u>2</u>	<u>3</u>
$R_1$	S#	SNAME	SLOC
$R_2$	P#	PNAME	
$R_3$	J#	JNAME	JLOC
$R_4$	S#	P#	J#

The relations are to be interpreted as follows: A 3-tuple of  $R_1$  describes a "supplier" by number (S#), name (SNAME) and location (SLOC); a 2-tuple of  $R_2$  describes a "part" by number (P#) and name (PNAME); a 3-tuple of  $R_3$  describes a "project" by number (J#), name (JNAME) and location (JLOC); a 3-tuple (x,y,z) of  $R_4$  represents the fact that the supplier with number x supplies the part with number y to the project with number z.

#### 2.3.2.2. Sample Queries

The eight sample queries of this section are provided to demonstrate that the operator set  $\Omega_0$  possesses a powerful capability for defining relations from S. All relations are assumed to be non-empty and that

$$(2.16) \quad \pi_1(R_1) = \pi_1(R_4), \quad \pi_1(R_2) = \pi_2(R_4), \quad \pi_1(R_3) = \pi_3(R_4).$$

Each query is given in English, as a set definition using a first-order predicate calculus, and as an expression in  $\xi(S, \Omega_0)$ .

(2.17a) Find all supplier name/project name pairs.

(2.17b)  $\{x[2], y[2] : x \in R_1 \wedge y \in R_3\}$ .

(2.17c)  $\pi_{2,5}(R_1[I]R_3)$ . (Note:  $I(r,s)$  is identically "1")

(2.18a) Find all supplier name/project name pairs where both are located in the same city.

(2.18b)  $\{x[2], y[2] : x \in R_1 \wedge y \in R_3 \wedge x[3] = y[3]\}$ .

(2.18c)  $\pi_{2,5}(R_1[r[3] = s[3]]R_3)$ .

(2.19a) Find all supplier name/project name pairs where both are located in the same city and the supplier supplies at least one part to the project.

(2.19b)  $\{x[2], y[2] : x \in R_1 \wedge y \in R_3 \wedge \exists z \in R_4 (x[3] = y[3] \wedge x[1] = z[1] \wedge y[1] = z[3])\}$ .

(2.19c)  $\pi_{2,5}((R_1[I]R_3)[r[3] = r[6] \wedge r[1] = s[1] \wedge r[4] = s[3]]R_4)$ .

(2.20a) Find all supplier name/project name pairs where the supplier supplies all parts to the project.

(2.20b)  $\{x[2], y[2] : x \in R_1 \wedge y \in R_3 \wedge \forall z \in R_4 (y[1] \neq z[3] \vee x[1] = z[1])\}$ .

(2.20c)  $\pi_{2,5}(((R_1[I]R_3)[r[4] \neq s[3] \vee r[1] = s[1]]R_4)[7,8,9 \div 1,2,3]R_4)$

(2.21a) Find the names of all suppliers who are the only supplier of any part.

(2.21b)  $\{x[2] : x \in R_1 \wedge \exists y \in R_2 \forall z \in R_4 (y[1] \neq z[2] \vee x[1] = z[1])\}$ .

(2.21c)  $\pi_2(((R_1[I]R_2)[r[4] \neq s[3] \vee r[1] = s[1]]R_4)[6,7,8 \div 1,2,3]R_4)$ .

(2.22a) Find the names of every project for which each part supplied to the project is supplied to some other project as well.

(2.22b)  $\{x[2] : x \in R_3 \wedge \forall y \in R_4 \exists z \in R_4 (x[1] \neq y[3] \vee y[3] \neq z[3] \wedge y[2] = z[2])\}$ .

(2.22c)  $\pi_2(\pi_{1,2,3,4,5,6}((R_3[I]R_4)[r[1] = r[6] \vee r[6] \neq s[3] \wedge r[5] = s[2]]R_4))[4,5,6 \div 1,2,3]R_4)$ .

(2.23a) Find all part names for which the part is supplied to some project and every supplier who supplies this part supplies it to at least one other project as well.

(2.23b)  $\{x[2] : x \in R_2 \wedge \exists y \in R_4 \forall z \in R_4 \exists w \in R_4 (x[1] = y[2] \wedge (y[2] \neq z[2] \vee z[2] = w[2] \wedge y[1] = w[1] \wedge y[3] \neq w[3]))\}$ .

(2.23c)  $\pi_2(\pi_{1,2,3}(\pi_{1,2,3,4,5,6,7,8}(((R_2[I]R_4)[I]R_4)[r[1] = r[4] \wedge (r[4] \neq r[7] \vee r[7] = s[2] \wedge r[3] = s[1] \wedge r[8] \neq s[3]]R_4))[7,8,9 \div 1,2,3]R_4))$ .

(2.24a) Find all part names such that every supplier who supplies the part supplies every project with at least one part.

(2.24b)  $\{x[2] : x \in R_2 \wedge \forall y \in R_4 \forall z \in R_3 \exists w \in R_4 (x[1] \neq y[2] \vee z[1] = w[3] \wedge y[1] = w[1])\}$ .

$$(2.24c) \quad \pi_2(((\pi_{1,2,3,4,5,6,7,8}((R_2[I]R_4)[I]R_3) \\ [r[4] \neq r[1] \vee r[6] = s[3] \wedge r[3] = s[1])R_4) \\ [6,7,8 \div 1,2,3]R_3)[3,4,5 \div 1,2,3]R_4).$$

$$(2.25) \quad \text{Remark: } \pi_A(R_i[f]R_j) = \{r : r \in R_i \wedge \exists s \in R_j (f(r,s) = 1)\}, \\ \text{where } A = 1, \dots, \text{deg}(R_i).$$

$$(2.26) \quad \text{Remark: } (R_i[f]R_j)[A \div B]R_j = \{r : r \in R_i \wedge \forall s \in R_j (f(r,s) = 1)\}, \\ \text{where } A = \text{deg}(R_i) + 1, \dots, \text{deg}(R_i) + \text{deg}(R_j) \\ \text{and } B = 1, \dots, \text{deg}(R_j).$$

#### 2.4. Other Relational Operators

Consider two operator sets,  $\Omega$  and  $\Omega'$  such that  $\Omega$  is a proper subset of  $\Omega'$ . Clearly  $\xi(S, \Omega)$  is a proper subset of  $\xi(S, \Omega')$ , however  $C(S, \Omega)$  and  $C(S, \Omega')$  may be identical. In that case, an  $\Omega$ -EVALUATOR is sufficient to evaluate any  $F$  in  $\xi(S, \Omega')$  provided  $F$  can be translated to a  $G$  in  $\xi(S, \Omega)$  that defines the same relation.

Seven relational operators are defined in this section; six of these are shown to be definable from the  $\Omega_0$  operators of 2.3, so that augmenting  $\Omega_0$  by any of these offers no advantage for data retrieval purposes (other than perhaps user convenience). Section 2.4.1 provides definitions and terminology used in the sequel; 2.4.2 defines the operators and constructively demonstrates their derivability from  $\Omega_0$ .

##### 2.4.1. Definitions and Terminology

(2.27) Definition: Let  $\Omega$  be a set of relational operators and  $S$  a fixed set of relations. Let  $E$  and  $E'$  be expressions in  $\xi(S, \Omega)$ .  $E$  and  $E'$  are S-equal if they evaluate to the same relation in  $C(S, \Omega)$ .



S-equality induces an equivalence partition of  $\xi(S, \Omega)$ ; every expression in an S-equivalence class defines the same relation in  $C(S, \Omega)$ . In general, detecting that E and E' in  $\xi(S, \Omega)$  are in the same S-equivalence class requires the evaluation of each. A more useful equivalence relation exists between two expressions E and E' in  $\xi(S, \Omega)$  if they define the same relation in  $C(S, \Omega)$  for any fixed S.

(2.28) Definition: Let  $\Omega$  be a set of relational operators and  $R_1, \dots, R_p$  be relation names. Let E and E' be expressions in  $\xi(\{R_1, \dots, R_p\}, \Omega)$ . E and E' are strongly equivalent if they define precisely the same relation in  $C(S, \Omega)$  for any fixed set of relations S containing relations  $R_1, \dots, R_p$ .

In Chapter IV, strong equivalence is exploited to select more efficiently evaluable expressions; in this chapter it is used to compare operators.

(2.29) Definition: Let  $R_1, \dots, R_p$  be relation names,  $\Omega$  a set of relational operators and  $w$  any relational operator.  $w$  is  $\Omega$ -derivable if for any  $E \in \xi(\{R_1, \dots, R_p\}, \Omega \cup \{w\})$ , there exists an  $E' \in \xi(\{R_1, \dots, R_p\}, \Omega)$  such that E and E' are strongly equivalent.

(2.30) Remark: If  $\Omega$  is a set of relational operators and  $w \in \Omega$ , then clearly  $w$  is  $\Omega$ -derivable.

(2.31) Remark: If a relational operator  $w$  is  $\Omega$ -derivable and  $\Omega \subset \Omega'$ , then clearly  $w$  is  $\Omega'$ -derivable.

If for any  $F \in \xi(S, \{w\})$  there exists an  $F' \in \xi(S, \Omega)$  such that  $F$  and  $F'$  are strongly equivalent, then for every  $G \in \xi(S, \Omega \cup \{w\})$  there exists a  $G' \in \xi(S, \Omega)$  such that  $G$  and  $G'$  are strongly equivalent. This can be proved by induction on the number of occurrences of "w" in  $G$  and justifies the following proposition:

(2.32) Proposition: Let  $\Omega$  be an operator set and  $w$  a  $p$ -ary relational operator.  $w$  is  $\Omega$ -derivable if and only there exists a  $G \in \xi(S, \Omega)$  that is strongly equivalent to  $w(R_1, \dots, R_p)$  where  $S = \{R_1, \dots, R_p\}$ .

Proof: (omitted)

We next explore the other relational operators defined by Codd [1,6] and demonstrate that all but one are  $\Omega_0$ -derivable. For convenience, we represent operators symbolically: "#", " $\pi$ " and " $\div$ " stand for JOIN, PROJECTION and DIVISION; symbols for the other operators are provided below.

#### 2.4.2. Operators

##### 2.4.2.1. PRODUCT (symbol = "\*")

(2.33) Definition: Let  $R_i$  and  $R_j$  be relations. The PRODUCT of  $R_i$  and  $R_j$  (in that order) is denoted by  $R_i * R_j$  and defined as:  $R_i * R_j = \{rs : r \in R_i \wedge s \in R_j\}$ .

(2.34) Proposition:  $*$  is  $\{\#\}$ -derivable.

Proof:  $R_i * R_j = R_i[f]R_j$  where  $f(r,s) \equiv 1$ .

##### 2.4.2.2. $\theta$ -JOIN (symbol = " $\#_\theta$ ")

(2.35) Definition: Let  $R_i$  and  $R_j$  be relations. Let  $A = a_1, \dots, a_k$  and  $B = b_1, \dots, b_k$  be domain-identifying lists for  $R_i$  and  $R_j$  respectively, and  $\theta$  one of  $\{=, \neq, <, \leq, >, \geq\}$ . The  $\theta$ -JOIN of  $R_i$  on domains  $A$  and  $R_j$  on domains  $B$  is denoted  $R_i[A\theta B]R_j$  and defined as:

$$R_i[A\theta B]R_j = \{rs : r \in R_i \wedge s \in R_j \wedge r[a_1]\theta s[b_1] \\ \wedge \dots \wedge r[a_k]\theta s[b_k]\}.$$

$\theta$ -JOIN utilizes a simpler syntax for the predicate than JOIN, however it also restricts the class of predicates that can be specified. The proof of the next proposition is trivial;  $\theta$ -JOIN will not be further considered.

(2.36) Proposition:  $\#_\theta$  is  $\{\#\}$ -derivable.

#### 2.4.2.3. RESTRICTION (symbol = "E")

(2.37) Definition: Let  $R_i$  be a relation of degree  $m$  and  $g = g(r) = g(r[1], \dots, r[m])$  a 0-1 function of tuple variable  $r$  ranging over  $R_i$ . The  $g$ -RESTRICTION of  $R_i$  is denoted by  $R_i[g]$  and defined as:

$$R_i[g] = \{r : r \in R_i \wedge g(r) = 1\}.$$

The same limitation for the function "f" in the definition of JOIN in 2.3 applies to "g" in the above definition. (See Remark 2.9.)

(2.38) Proposition:  $E$  is  $\{\#, \pi\}$ -derivable.

$$\text{Proof: } R_i[g] = \pi_L(R_i[g]R_j)$$

where  $\text{deg}(R_i) = m$ ,  $L = 1, \dots, m$

and  $R_j$  is any relation.

2.3.2.4.  $\theta$ -RESTRICTION (symbol = " $E_\theta$ ")

- (2.39) Definition: Let  $R_i$  be a relation and  $A = a_1, \dots, a_k$  and  $B = b_1, \dots, b_k$  be domain-identifying lists for  $R_i$ . Let  $\theta$  be one of  $\{=, \neq, <, \leq, >, \geq\}$ . The  $\theta$ -RESTRICTION of  $R_i$  on domains  $A$  and  $B$  is denoted by  $R_i[A\theta B]$  and defined as:
- $$R_i[A\theta B] = \{r : r \in R_i \wedge r[a_1]\theta r[b_1] \wedge \dots \wedge r[a_k]\theta r[b_k]\}.$$

Like  $\theta$ -JOIN,  $\theta$ -RESTRICTION utilizes a simpler syntax for the subsetting predicate and restricts the class of predicates that can be specified. The proof of the next proposition is trivial;  $\theta$ -RESTRICTION will not be further considered.

- (2.40) Proposition:  $E_\theta$  is E-derivable.

The next three operators are the traditional set operators INTERSECTION, RELATIVE COMPLEMENT and UNION.

2.4.2.5. INTERSECTION (symbol = " $\cap$ ")

- (2.41) Definition: Let  $R_i$  and  $R_j$  be relations. The INTERSECTION of  $R_i$  and  $R_j$  is denoted by  $R_i \cap R_j$  and defined as:  $R_i \cap R_j = \{r : r \in R_i \wedge r \in R_j\}$ .

- (2.42) Proposition:  $\cap$  is  $\{\#, \pi\}$ -derivable.

Proof: Consider the following identity:

- (2.43)  $R_i \cap R_j = \{r : r \in R_i \wedge \exists s \in R_j (r[A] = s[A])\}$   
 where  $\text{deg}(R_i) = \text{deg}(R_j) = m$   
 and  $A = 1, \dots, m$

By 2.43 and Remark 2.25

$$R_i \cap R_j = \pi_{\Lambda}(R_i[r[A] = s[A]]R_j)$$

so that  $\cap$  is  $\{\#, \pi\}$ -derivable.

#### 2.4.2.6. RELATIVE COMPLEMENT (symbol = "\")

(2.44) Definition: Let  $R_i$  and  $R_j$  be relations.

The RELATIVE COMPLEMENT of  $R_i$  and  $R_j$  (in that order)

is denoted by  $R_i \setminus R_j$  and defined as:

$$R_i \setminus R_j = \{r : r \in R_i \wedge r \notin R_j\}.$$

(2.45) Proposition:  $\setminus$  is  $\{\#, \div\}$ -derivable.

Proof: Consider the following identity:

$$(2.46) \quad R_i \setminus R_j = \{r : r \in R_i \wedge \forall s \in R_j (r[A] \neq s[A])\}$$

where  $\deg(R_i) = \deg(R_j) = m$

and  $A = 1, \dots, m$

By 2.46 and Remark 2.26

$$R_i \setminus R_j = (R_i[r[A] \neq s[A]]R_j)[B \div A]R_j$$

where  $B = m+1, \dots, 2m$

so that  $\setminus$  is  $\{\#, \div\}$ -derivable.

#### 2.4.2.7. UNION (symbol = "U")

(2.47) Definition: Let  $R_i$  and  $R_j$  be relations, both of degree  $m$ . The

UNION of  $R_i$  and  $R_j$  is designated  $R_i \cup R_j$  and defined

as:

$$R_i \cup R_j = \{r : r \in R_i \vee r \in R_j\} \text{ if for every } r \in R_i$$

and for every  $s \in R_j$  it is the case that  $r[k] \theta s[k]$

(for  $k = 1, \dots, m$  and  $\theta$  one of  $\{=, \neq, <, \leq, >, \geq\}$ )

is either "true" or "false". not undefined.

Else,  $R_i \cup R_j$  is undefined.

The limitation on UNION that every element of the  $k$ th domain of  $R_i$  be  $\theta$ -comparable to every element of the  $k$ th domain of  $R_j$  is termed "union-compatibility" by Codd [6], and for consistency was required for INTERSECTION and RELATIVE COMPLEMENT as well.

(2.48) Proposition:  $\cup$  is not  $\{\#, \pi, \div\}$ -derivable.

Proof: The proof follows trivially from the observation that none of  $\{\#, \pi, \div\}$  produces a relation with any domain that is a proper superset of a domain of its operand relations. So that  $C(S, \{\cup\}) \not\subset C(S, \{\#, \pi, \div\})$  and  $\cup$  is not  $\{\#, \pi, \div\}$ -derivable.

The observation in the proof of Proposition 2.48 applies to every operator in the set

$$\{\#, \pi, \div, *, \#_{\theta}, E, E_{\theta}, \cap, \setminus\}$$

so that the UNION operator cannot be defined in terms of any of these.

The relational algebra originally defined by Codd [6] included all operators defined here except JOIN and RESTRICTION which are generalizations of his  $\theta$ -JOIN and  $\theta$ -RESTRICTION. Using Codd's algebra to define the relation

$$\{rs : r \in R_i \wedge s \in R_j \wedge (r[1] = s[1] \vee r[2] = s[2])\}$$

requires the UNION operator. (E.g.  $R_i[1=1]R_j \cup R_i[2=2]R_j$ .) However, with JOIN instead of  $\theta$ -JOIN, UNION is not required. (E.g.  $R_i[r[1] = s[1] \vee r[2] = s[2]]R_j$ .)

The only relational algebras considered here use JOIN and RESTRICTION, thus eliminating the primary need for UNION. A secondary use of UNION is for combining relations; the operation is more typical for data processing (where it is termed "merge") than for data retrieval. Efficient techniques for taking the UNION of relations (files) are discussed in Knuth [17] and will not be considered here. This also avoids introducing artificial syntactic devices for checking for the semantic "union-compatibility" constraint.

If the UNION of two data base relations needs to be performed, we presume this to take place external to any  $\Omega$ -EVALUATORS considered in this thesis.

## 2.5. Equivalent Operator Sets

The operator set  $\Omega_0$  consisting of JOIN, PROJECTION and DIVISION was defined in 2.3 and shown to provide an extensive relation-defining capability. In this section, the three operator sets

$$\Omega_1 = \{\text{RESTRICTION, PRODUCT, PROJECTION, DIVISION}\}$$

$$\Omega_2 = \{\text{JOIN, PROJECTION, RELATIVE COMPLEMENT}\}$$

$$\Omega_3 = \{\text{RESTRICTION, PRODUCT, PROJECTION, RELATIVE COMPLEMENT}\}$$

are shown to provide the same relation-defining capability as  $\Omega_0$ .

(2.49) Definition: Let  $\Omega$  and  $\Omega'$  be sets of relational operators.

If every  $w \in \Omega$  is  $\Omega'$ -derivable and every  $w' \in \Omega'$  is  $\Omega$ -derivable, then  $\Omega$  and  $\Omega'$  are computationally equivalent.

If two operator sets  $\Omega$  and  $\Omega'$  are computationally equivalent, then for any set of relations  $S$ , any  $F \in \xi(S, \Omega)$  is strongly equivalent to an  $F' \in \xi(S, \Omega')$  and vice versa, so that  $C(S, \Omega) = C(S, \Omega')$ . That is, they

define precisely the same set of relations from S.

$$2.5.1. \quad \underline{\Omega_1 = \{E, *, \pi, \div\}}$$

(2.50) Proposition: # is  $\{E, *\}$ -derivable.

Proof: Let  $R_i$  and  $R_j$  be relations of degree  $m$  and  $n$  and  $f = f(r, s) = f(r[1], \dots, r[m], s[1], \dots, s[n])$  be a 0-1 function of tuple variables  $r$  and  $s$  ranging respectively over  $R_i$  and  $R_j$ .

Then

$$R_i[f]R_j = (R_i * R_j)[g]$$

$$\text{where } g = g(r) = f(r[1], \dots, r[m+n]).$$

(2.51) Proposition:  $\Omega_0$  and  $\Omega_1$  are computationally equivalent.

Proof:  $E$  and  $*$  are  $\Omega_0$ -derivable by Propositions 2.38, 2.34 and Remark 2.31.  $\pi$  and  $\div$  are  $\Omega_0$ -derivable and  $\Omega_1$ -derivable by Remark 2.30. # is  $\Omega_1$ -derivable by Proposition 2.50 and Remark 2.31. Therefore  $\Omega_0$  and  $\Omega_1$  are computationally equivalent.

$$2.5.2. \quad \underline{\Omega_2 = \{\#, \pi, \backslash\}}$$

(2.52) Proposition:  $\div$  is  $\{*, \pi, \backslash\}$ -derivable.

$$\text{Proof: } R_i[A \div B]R_j = \pi_{\bar{A}}(R_i) \backslash \pi_{1, \dots, m-k}(\pi_{\bar{A}}(R_i) * \pi_B(R_j) \backslash \pi_{\bar{A}, A}(R_j))$$

where  $A$  is of length  $k$ ,  $\deg(R_i) = m$ , and  $\bar{A}$  is the domain-identifying list complementary to  $A$  relative to  $R_i$ .

(2.53) Proposition:  $\div$  is  $\{\#, \pi, \backslash\}$ -derivable.

Proof: trivial; replace "\*" in the proof of Proposition 2.52 with "f" where  $f(r, s) \equiv 1$ .



(2.54) Proposition:  $\Omega_0$  and  $\Omega_2$  are computationally equivalent.

Proof: Immediate from Propositions 2.45 and 2.53.

2.5.3.  $\underline{\Omega_3 = \{E, *, \pi, \setminus\}}$

(2.55) Proposition:  $\Omega_0$  and  $\Omega_3$  are computationally equivalent.

Proof: Immediate from Propositions 2.38, 2.34, 2.45, 2.50 and 2.52.

(2.56) Definition: An operator set  $\Omega$  is non-redundant if no  $w \in \Omega$  is  $(\Omega \setminus \{w\})$ -derivable.

It is generally quite difficult to show that an operator set is non-redundant. The following is left as a conjecture:

(2.57) Conjecture:  $\Omega_0, \Omega_1, \Omega_2$  and  $\Omega_3$  are non-redundant operator sets.

2.6. The Operator Set  $\Omega^* = \Omega_1 = \{E, *, \pi, \div\}$

The results of 2.4 and 2.5 indicate that an  $\Omega$ -EVALUATOR for  $\Omega \in \{\Omega_0, \Omega_1, \Omega_2, \Omega_3\}$  is sufficient to evaluate any expression  $F$  in  $\xi(S, \{\#, \pi, \div, E, *, \cap, \setminus\})$  since translation of  $F$  to a strongly equivalent expression  $F'_0 \in \xi(S, \Omega_0)$  or  $F'_1 \in \xi(S, \Omega_1)$  or  $F'_2 \in \xi(S, \Omega_2)$  or  $F'_3 \in \xi(S, \Omega_3)$  can be performed mechanically using the constructions of Propositions 2.34, 2.38, 2.42, 2.45, 2.50, 2.52, and 2.53.

This simplifies the implementation of an  $\Omega$ -EVALUATOR but requires a choice among  $\Omega_0, \Omega_1, \Omega_2$  and  $\Omega_3$ , and that choice is necessarily subjective.

The operator sets  $\Omega_2$  and  $\Omega_3$  were excluded because of the complex expression that results whenever  $R_i[A \div B]R_j$  is translated to an expression using PROUCT, PROJECTION and RELATIVE COMPLEMENT. (See Proposition 2.52.)

Set operators in higher-level programming languages are discussed by Earley [18] and by Schwartz [19].

The choice between  $\Omega_0$  and  $\Omega_1$  reduces to a choice between the JOIN operator and the PRODUCT and RESTRICTION operators. For implementation purposes, the choice is completely arbitrary. For simplicity, PRODUCT and RESTRICTION are preferred since the operation of subsetting a single relation  $R_i$  requires either a RESTRICTION of  $R_i$  or a PROJECTION of a JOIN between  $R_i$  and some dummy relation  $R_j$ . (See Proposition 2.38.)

A second factor in this choice becomes evident in Chapter V. The material of that chapter relies on a translation algorithm from a non-algebraic retrieval language to a relational algebra that employs RESTRICTION and PRODUCT, not JOIN.

Therefore, the operator set  $\Omega_1$  was chosen. This set will be designated as  $\Omega^*$ ; implementation of an efficient  $\Omega^*$ -EVALUATOR is discussed in Chapters III, IV and V.

## 2.7. Limitations

We discuss 2 limitations to the general approach described in this chapter: the first concerns Figure 2.1 and the second concerns a limitation of using an algebra as a retrieval language.

### 2.7.1. Implementation Framework

The logical implementation framework of Figure 2.1 fails to support any user request other than retrieval. E.g. updating, adding or deleting tuples, and data base reorganization including creation or elimination of relations.

To overcome this limitation, we suppose that an  $\Omega$ -EVALUATOR is only a portion of such a system, that the relational algebra is a sublanguage

in a host language which provides facilities for expressing the other types of interactions, and that the system includes procedures for effecting them.

The fact that tuples to be modified or deleted are often specified by attribute value predicates indicates that an  $\Omega$ -EVALUATOR can be an integral portion of such a system and not just an independent part for performing retrievals. That is, an  $\Omega$ -EVALUATOR can often be used to identify the tuples for modification (update) or deletion.

### 2.7.2. Retrieval Specification Using a Relational Algebra

The fact that relational operators map relations to relations indicates that the response to any query posed as an expression in a relational algebra is a relation. The operators defined in this chapter provide no facility for answering queries "about" relations, such as the average of a (numeric) domain or the number of tuples in a relation.

One way around this problem is to define new operators which evaluate to a unary relation of one 1-tuple whose value is the average of a domain, number of tuples in a relation, etc. This approach is used in the data base sublanguages DSL-ALPHA [7], SEQUEL [8], DAMAS [10] and QUEL [11]. Efficient techniques for their implementation are as yet unknown.

We prefer to think of such defining facilities as being part of a host language rather than the data base sublanguage and exclude them from further consideration.

### 2.8. Summary

This chapter investigated relational algebras induced by a set of relational operators  $\Omega$  over a set of relations  $S$ . Ten operators were

defined and compared, and four sets of operators  $\Omega_0$ ,  $\Omega_1$ ,  $\Omega_2$  and  $\Omega_3$  were shown to provide equal relation-defining capability. Examples for  $\Omega_0$  demonstrated that this capability is sufficiently great that the relational algebraic expressions over a set  $S$  of data base relations and  $\Omega_0$  or  $\Omega_1$  or  $\Omega_2$  or  $\Omega_3$  could be considered as a retrieval language in a relational data base system.

Within the logical implementation framework of Figure 2.1, the collection of procedures which implements the map from expressions over data base relations  $S$  and operators  $\Omega$  to the relation defined (relative to the data base) was termed an  $\Omega$ -EVALUATOR. The remainder of the thesis is concerned with constructing an efficient  $\Omega$ -EVALUATOR for  $\Omega = \Omega^* = \Omega_1$ .

Chapter III. IMPLEMENTATION OF AN  $\Omega^*$ -EVALUATOR3.1. Introduction

Chapter II described a logical framework in which an  $\Omega^*$ -EVALUATOR mapping  $\xi(S, \Omega^*)$  to  $C(S, \Omega^*)$  serves as a mechanism for data retrieval from a relational data base for  $S$ . The remainder of this thesis is concerned with the implementation of an  $\Omega^*$ -EVALUATOR that minimizes the time between presentation of  $F \in \xi(S, \Omega^*)$  and delivery of the relation in  $C(S, \Omega^*)$  defined by  $F$  relative to data base relations  $S$  (e.g. the response time).

This chapter describes a hypothetical computer system for implementation, a simple uniform storage representation for relations, and running-time measure for procedures which run in this environment that reflects response time. With this model and measure, efficient procedures for the  $\Omega^*$  operators will be derived. The model and measure are used in Chapter IV and V where higher-level techniques for gaining efficiency in an  $\Omega^*$ -EVALUATOR are explored.

The approach taken in Chapters III, IV and V is to produce analytic results for the implementation of the retrieval portion of relational data base system. That approach necessitates the simplifying assumptions concerning the representation of relations and the running-time measure that appear in 3.2. These results provide a basis for further work in the design of relational data base systems. Extension to more elaborate storage representation and to the data base altering interactions (addition, deletion, update) are clearly indicated. (See Held and Stonebraker [14] and Smith and Chang [13].)

The remainder of this chapter is organized as follows: In section 3.2 we define a model and measure; in 3.3, file activities required by

the  $\Omega^*$ -EVALUATOR and the time required to perform these are described; in 3.4, 3.5, 3.6 and 3.7, efficient procedures for the  $\Omega^*$  operators of RESTRICTION, PRODUCT, PROJECTION and DIVISION are derived; 3.8 is a summary.

### 3.2. Model and Measure

This section contains 3 subsections which describe in order the characteristics of a hypothetical computer system for implementation, the representation of the relational data base and the measure used for comparing algorithms.

#### 3.2.1. A Computer System for Implementation

We consider the implementation of an  $\Omega^*$ -EVALUATOR on a conventional single processor computer with a fast, limited primary memory and a slow, essentially unbounded secondary memory. The secondary storage is considered to be a collection of disks with single channel access to any of the disks. The logic of the processor is assumed to be so fast relative to the time to transfer data between primary and secondary memory, that additional processors would be of no advantage.

No assumptions are made concerning other processes which may run concurrently with the  $\Omega^*$ -EVALUATOR except that where they exist, they can be blocked from modifying the relational data base while the  $\Omega^*$ -EVALUATOR is active.

The analysis that follows is appropriate to both virtual and non-virtual memory system, but is presented in terms of I/O time and not page faults. Extension to the latter requires only a knowledge of page size and the memory management strategy used by any virtual memory system in question. Since processor logic is very fast compared to I/O

time, filling and emptying buffers is assumed to be the primary factor governing response time.

### 3.2.2. Representation of the Relational Data Base

The relational data base is represented by a collection of files in secondary memory, one file for each data base relation. A file for relation  $R_i$  contains  $n_i$  records, one per tuple of  $R_i$ , each of fixed length  $b_i$  bits. If variable-length encoding or data compressions techniques are used, the results that follow are applicable if  $b_i$  is taken as the average length of a record. Each relation is essentially a linear table of tuples with a single retrieval order corresponding to the order in which the tuples were written to the data base files.

It is assumed that files for the data base relations contain no infile pointers, and that no additional accessing structures, such as secondary indexes, are available for accessing the tuples of the data base relations.

#### 3.2.2.1. On the Exclusion of Secondary Indexes

The assumption that no secondary indexes are available is a simplification required for the analysis that follows. The fact that secondary indexes can be logically represented as binary relations indicates that they can be effectively manipulated using a relational algebra. (See Held and Stonebraker [20].) Extension of the results of this thesis for systems employing secondary indexes would be beneficial.

The use and selection of secondary indexes is discussed in Schkolnick [21]. Rivest [22] investigates an important class of retrievals (termed "partical match queries") using scatter storage (hashing) techniques.

### 3.3.3. Measure

Since our specific concern is in minimizing response time and processor logic is very fast compared to transfer time between primary and secondary memory, an appropriate measure is the volume of data transferred as a function of the number of tuples in the operand relations(s). We exclude the response relation from our measure since its' output time is independent of the procedure which produces it.

We say that an algorithm runs in time  $O(f(n))$  for a relation with  $n$  tuples if the actual running time  $T(n)$  is proportional to  $f(n)$ , or equivalently.

$$(3.1) \quad \lim_{n \rightarrow \infty} \frac{f(n)}{T(n)} = k, \quad \text{where } k \text{ is a non-zero proportionately constant.}$$

### 3.3. File Activities of the $\Omega^*$ -EVALUATOR

The measure of section 3.2 includes only the transfer of data between primary and secondary memory. The running time of a procedure to evaluate (say)  $R_i * R_j$  is therefore a measure of the film activity it generates. We identify 3 basic file activities of the  $\Omega^*$ -EVALUATOR:

- (1) scanning a data base relation  $R_i$ .
- (2) searching a data base relation  $R_i$  for tuple  $r_0$ .
- (3) sorting a data base relation  $R_i$  on domains  $L = a_1, \dots, a_k$ .

Scanning relation  $R_i$  consists of sequentially inputting every tuple of  $R_i$  for processing by some procedure. Since the number of bits to represent relation  $R_i$  is proportional to  $n_i$ , the time to scan  $R_i$  is  $O(n_i)$ .

Searching relation  $R_i$  for tuple  $r_0$  requires a sequential scan of  $R_i$  that will require examination of  $n_i/2$  tuples on the average if  $r_0 \in R_i$



and  $n_i$  tuples if  $r_0 \notin R_i$ . Both cases are  $O(n_i)$ .

Since relations are sets with no specified order properties, the result of applying an operator to its operand relation(s) is the same regardless of the order of the representation(s) in the data base. The results of this chapter will show that known orders for the representation of the operand relation(s) can be exploited to reduce running time. Facilities for sorting the table-of-tuples representations are therefore included. A relation  $R_i$  is sorted on domains  $L = a_1, \dots, a_k$  (where  $L$  is a domain identifying list without repetition) if it is the case that for any  $r, s$  in  $R_i$ ,  $r$  precedes  $s$  if and only if  $r[L] \leq s[L]$ . (Character data is sorted lexicographically, or numerically on the encoding). The time to sort relation  $R_i$  is taken to be  $O(n_i \log_2 n_i)$  as shown in Knuth [17], pp. 361-376.

Having established the running times for scanning, searching and sorting, we now consider procedures for application of the operators in  $\Omega^*$ .

#### 3.4. RESTRICTION

The time to evaluate the RESTRICTION  $R_i[g]$  is  $O(n_i)$  since for general "g", every tuple of  $R_i$  must be retrieved and tested with g. The following example indicates that a complete scan of  $R_i$  can be avoided for certain functions "g" and certain orderings of the representation of  $R_i$ , but for general "g", we use the  $O(n_i)$  approximation.

3.2) Example: Suppose  $R_i$  is sorted on domain 1 and  $R_i[r[1] < 10]$  must be evaluated. Since  $r$  precedes  $s$  only if  $r[1] \leq s[1]$ , then  $R_i$  need be scanned only up to the first tuple  $r_0$  such that  $r_0[1] \geq 10$  since every tuple  $r$  following  $r_0$  will have  $r[1] \geq r_0[1] \geq 10$ .

### 3.5. PRODUCT

The time to evaluate the PRODUCT  $R_i * R_j$  is essentially  $O(n_i n_j)$  since every tuple of  $R_i$  must be combined with every tuple of  $R_j$ . Suppose that primary memory buffers are available that can hold  $p$  tuples from  $R_i$  and  $q$  tuples from  $R_j$ . The  $R_i$  buffer must be filled  $n_i/p$  times, and each time it is filled, the  $R_j$  buffer must be filled  $n_j/q$  times. The time required is proportional to

$$(3.3) \quad (n_i/p)(n_j/q)$$

which is  $O(n_i n_j)$ .

It is possible to avoid filling the  $R_j$  buffer once for every filling of the  $R_i$  buffer (except the first) by retaining the last buffer contents from the previous filling. However, the  $O(n_i n_j)$  approximation is still valid.

### 3.6. PROJECTION

The evaluation of the PROJECTION  $\pi_L(R_i)$  is complicated by the fact that a relation is defined as a set. Duplicate occurrences of tuples in relations must therefore be avoided.

When  $L$  is a permutation of  $1, \dots, m$  where  $m$  is the degree of  $R_i$ , then two distinct tuples  $r$  and  $s$  in  $R_i$  map to distinct values  $r[L]$  and  $s[L]$  in  $\pi_L(R_i)$ . So that a sequential scan of  $R_i$  is sufficient to evaluate the permuting projection  $\hat{\pi}_L(R_i)$  (see Remark 2.12 and accompanying footnote). Therefore, any permuting projection of  $R_i$  is an  $O(n_i)$  operation.

With respect to data retrieval, permuting projections are generally used only for rearranging attribute values to a more convenient order in

response relations. Proper projection (Remark 2.12) eliminates domains of the operand relation and is a more important use of the PROJECTION operator.

Two distinct tuples  $r$  and  $s$  in  $R_i$  can have the same value  $r[L] = s[L]$  in the proper projection  $\pi_L(R_i)$ , so that care must be exercised to avoid the output of duplicate tuples in the response relation.

With limited primary memory and a large operand relation  $R_i$ , two distinct tuples  $r$  and  $s$  in  $R_i$  can occur so far apart in the representation of  $R_i$  that the output of duplicate tuples during a sequential scan of  $R_i$  can be avoided only by comparison to tuples previously output. This method can be shown to be  $O(n_i^2)$  for  $R_i$ .

When the tuples of  $R_i$  which map to the same value in  $\pi_L(R_i)$ , the output of duplicate tuples in the response relation can be avoided with comparisons between consecutive tuples, so that  $\pi_L(R_i)$  can be generated with a sequential scan of  $R_i$ . The following proposition provides a sufficiency condition that will be used in the sequel.

(3.4) Proposition: If each subset of  $R_i$  with the same value of  $r[L]$  is consecutively retrievable,  $\pi_L(R_i)$  can be evaluated in time  $O(n_i)$ .

Proof: trivial

The sufficiency condition of Proposition 3.4 is satisfied for the proper projection  $\pi_L(R_i)$  when the representation of  $R_i$  is sorted on  $L$  or on any domains containing  $L$  as a subset (including a permutation of  $L$ ). So that under any of these orderings,  $\pi_L(R_i)$  is an  $O(n_i)$  operation for  $R_i$ .

If the sufficiency condition of Proposition 3.4 is not met for  $\pi_L(R_i)$ ,  $R_i$  can be sorted on  $L$  so that it becomes satisfied to permit  $O(n_i)$  evaluation. The time to sort  $R_i$  is  $O(n_i \log_2 n_i)$ , so that the total evaluation for a proper projection of  $R_i$  using either a sort followed by a projection of a simultaneous sort-and-project is  $O(n_i \log_2 n_i)$ .

### 3.7. DIVISION

The time to evaluate the DIVISION  $R_i[A \div B]R_j$  depends not only on  $n_i$  and  $n_j$ , but also on the number of tuples in the result. In the following derivation, we assume  $n_i \geq n_j$  and that  $s[B]$  assumes  $n_j$  distinct values as  $s$  ranges over  $R_j$ .

From definition 2.12 for DIVISION, we have that for every  $r \in R_i$ :

$$(3.5) \quad r[\bar{A}] \in R_i[A \div B]R_j \iff \forall s \in R_j \exists t \in R_i (r[\bar{A}] = t[\bar{A}] \wedge t[A] = s[B])$$

and

$$(3.6) \quad r[\bar{A}] \notin R_i[A \div B]R_j \iff \exists s \in R_j \forall t \in R_i (r[\bar{A}] \neq t[\bar{A}] \vee t[A] \neq s[B]).$$

To verify the condition on the right hand side (rhs) of 3.5 requires time  $n_j O(n_i)$  since  $s$  assumes  $n_j$  values and it takes  $O(n_i)$  to search  $R_i$  for  $t$ . To verify the condition on the rhs of 3.6 requires time  $O(n_i)$  since  $R_i$  must be searched for  $t$ . If  $m$  is the number of tuples in  $R_i[A \div B]R_j$ ,  $m$  is bounded from below by 0 and from above by  $n_i/n_j$  since for each tuple  $p$  in  $R_i[A \div B]R_j$ , there must be  $n_j$  tuples  $z_1, \dots, z_{n_j}$  with  $z_1[\bar{A}] = \dots = z_{n_j}[\bar{A}] = p$ , and these tuples cannot satisfy the necessary condition on the rhs of 3.5 for any other  $p' \neq p$  in  $R_i[A \div B]R_j$ . This yields the time approximation.

$$(3.7) \quad mn_j O(n_i) + (n_i - mn_j) O(n_i)$$

for the entire computation which is  $O(n_i^2)$  over the range of  $m$ .

When it is known that  $R_i$  is sorted on  $\bar{A}, A$  and  $R_j$  is sorted on  $B$ , then for each  $p \in R_i[A \div B]R_j$ , (at least)  $n_j$  consecutively retrievable tuples  $z_1, \dots, z_{n_j}$  in  $R_i$  will have  $z_1[\bar{A}] = \dots = z_{n_j}[\bar{A}] = p$ , and within this subset, the  $n_j$  tuples that satisfy the role of  $t$  on the rhs of 3.5 will appear in the same order as the  $n_j$  tuples of  $R_j$ . This is illustrated in the following example:

(3.8) Example:  $R_i[2,3 \ 1,2]R_j$

( $R_i$  and  $R_j$  unsorted)

$R_i$	<u>1</u>	<u>2</u>	<u>3</u>	$R_j$	<u>1</u>	<u>2</u>
	A	X	1		Z	3
	C	Z	3		X	1
	B	Z	3			
	C	X	1			
	A	Z	1			
	A	Z	3			
	B	X	2			
	B	Z	1			

( $R_i$  sorted on  $\bar{A}, A=1,2,3$ ;  $R_j$  sorted on  $B=1,2$ )

$R_i$	<u>1</u>	<u>2</u>	<u>3</u>	$R_j$	<u>1</u>	<u>2</u>
	A	X	1		X	1
	A	Z	1		Z	3
	A	Z	3			
	B	X	2			
	B	Z	1			
	B	Z	3			
	C	X	1			
	C	Z	3			

With this arrangement of the representations of the operand relations, evaluation of  $R_i[A \div B]R_j$  can be performed with a sequential scan of  $R_i$  since no tuple of  $R_i$  has to be compared to more than one tuple of  $R_j$ . The time to sort  $R_i$  and to sort-and-project  $R_j$  is  $O(n_i \log_2 n_i)$

since  $n_i \geq n_j$ . The sequential scan of  $R_i$  is  $O(n_i)$ , so that the total time to sort and divide is  $O(n_i \log_2 n_i)$ .

The condition that " $R_i$  is sorted on  $\bar{A}, A$  and  $R_j$  sorted on  $B$ " is sufficient for  $O(n_i)$  evaluation of  $R_i[A \div B]R_j$  but not necessary. The following proposition provides a weaker sufficiency condition that will be used in the sequel:

(3.9) Proposition: If each subset of  $R_i$  with the same value  $r[\bar{A}]$  is consecutively retrievable and the tuples of each subset are retrievable in the same order as the tuples of  $R_j$ , then  $R_i[A \div B]R_j$  can be evaluated in time  $O(n_i)$ .

Proof: trivial

Essentially, DIVISION can always be performed in  $O(n_i \log_2 n_i)$ , and can be performed in time  $O(n_i)$  if the condition in Proposition 3.9 are met.

### 3.8. Summary

In this chapter we established a model on which we consider the implementation of an  $\Omega^*$ -EVALUATOR and provide a measure to compare procedures. Optimal procedures for the basic file activities of scanning, search and sorting are assumed, and these are used to derive efficient procedures for RESTRICTION, PRODUCT, PROJECTION and DIVISION.

In the next chapter, the same model and measure are assumed and we explore efficiency techniques that rely on translation of an expression to an equivalent expression that can be evaluated in less time.

## Chapter IV. EFFICIENCY BY TRANSLATION

This chapter presents techniques for reducing response time for an  $\Omega^*$ -EVALUATOR. The model (hypothetical computer system, table-of-tuples storage representation) and measure of Chapter III are assumed.

The basic approach is to consider the  $\Omega^*$ -EVALUATOR as an executive procedure that evaluates  $F \in \xi(S, \Omega^*)$  by performing a sequence of steps that produces the relation in  $C(S, \Omega^*)$  defined by  $F$  relative to a relational data base for  $S$ . Each step consists of a call to a sub-procedure for one of the  $\Omega^*$  operators (RESTRICTION, PRODUCT, PROJECTION, DIVISION) with relations in secondary memory as arguments. The relations in secondary memory are either data base relations (e.g. in  $S$ ) or intermediate relations formed by previous steps of the evaluation and stored in a workspace.

The sequence of steps is in essence specified by  $F$ . The techniques are used by the  $\Omega^*$ -EVALUATOR to reduce response time by translating  $F$  to  $F' \in \xi(S, \Omega^*)$  for which the sequence of steps produces the same relation in  $C(S, \Omega^*)$  as  $F$ , but in less time. We collectively refer to such techniques as "efficiency by translation."

The following examples illustrates the stepwise evaluation of  $F \in \xi(S, \Omega^*)$  by recursive application of  $\Omega^*$  operators to data base and intermediate relations. It will also be used to motivate the techniques described below.

(4.1) Example: Recursive evaluation of

$$F = \pi_{2,1}((\pi_{2,3,4}(\pi_{1,3,4,5}(R_1)))[3 \div 2]((R_2 * R_3)[g])),$$

where  $\deg(R_1) = 5$ ,  $\deg(R_2) = \deg(R_3) = 2$ .

- step 1  $T_1 = \pi_{1,3,4,5}(R_1)$
- step 2  $T_2 = \pi_{2,3,4}(T_1)$
- step 3  $T_3 = R_2 * R_3$
- step 4  $T_4 = T_3[g]$
- step 5  $T_5 = T_2[3 \div 2]T_4$
- step 6  $F = \pi_{2,1}(T_5)$

Regardless of the efficiency of each step of Example 4.1, 3 improvements are immediately available. First, since

$$(4.2) \quad \pi_{2,3,4}(\pi_{1,3,4,5}(R_1)) = \pi_{3,4,5}(R_1),$$

steps 1 and 2 can be replaced with the single step

$$(4.3) \quad \text{step 1'} \quad T_2 = \pi_{3,4,5}(R_1).$$

Second, the function "g" applied to  $T_3$  in step 4 could be applied to the tuples of  $R_2 * R_3$  as they are formed, thus avoiding the storage and retrieval of  $T_3$ . This results in the single step

$$(4.4) \quad \text{step 2'} \quad T_4 = (R_2 * R_3)[g].$$

Third,  $T_5$  is a relation of degree 2, so that  $\pi_{2,1}(T_5)$  is a permuting projection (Remark 2.12). If the order of the attribute values of the tuples of  $T_2[3 \div 2]T_4$  is reversed prior to output,  $F$  is generated without the storage and retrieval of  $T_5$ . This reduces steps 5 and 6 to

$$(4.5) \quad \text{step 3'} \quad F = \hat{\pi}_{2,1}(T_2[3 \div 2]T_4).$$



The reduction of steps 5 and 6 to step 3' is termed permutation overlap and is possible for any permuting projection. We generalize this as follows:

(4.6) Remark: The permuting projection  $\hat{\pi}_L(T)$  of the intermediate relation T can be formed as efficiently as T.

Similarly, the reduction of steps 3 and 4 to step 2' is termed restriction overlap. We generalize this as follows:

(4.7) Remark: The RESTRICTION T[g] of the intermediate relation T can be formed as efficiently as T.

Remarks 4.6 and 4.7 are trivial ways to reduce evaluation time. The reduction of steps 1 and 2 is representative of a less obvious efficiency technique which will be referred to as an efficiency translation. These techniques rely on the fact that the evaluation of F in  $\xi(S, \Omega^*)$  can be speeded up by evaluating F' in  $\xi(S, \Omega^*)$  where F' or a permuting projection of F' is strongly equivalent (Definition 2.28) to F. In 4.2, eleven transformation rules are derived to map commonly occurring subexpressions to expressions which are in fact either strongly equivalent or strongly equivalent up to permutation. The applicability of these rules for reducing response time is discussed in 4.3. Eight are shown to be positive in benefit, two negative, and one neutral. In 4.4, the neutral transformation rule and the inverse map of one negative rule lead to positive efficiency results for

(4.8)  $\xi(S, \{E, *\})$ ,

the subset of  $\xi(S, \Omega^*)$  consisting of expressions involving only the PRODUCT and RESTRICTION operators.

The techniques presented bear similarities to results described by Palermo [12] and by Smith and Chang [13]. Palermo demonstrated an efficiency translation technique that corresponds to the sixth transformation rule (TR6; see below) in 4.2. That result stimulated the investigation that identified the ten other transformation rules, some of which are quite simple. Using a tree representation for expressions in Codd's relations algebra, Smith and Chang discovered five efficiency transformation that are in fact equivalent to the first, fifth, sixth, ninth and eleventh transformation rules (TR1, TR5, TR6, TR9, TR11) of 4.2.

No attempt is made in this chapter to exploit known orderings of the data base relations or common subexpressions of an expression. Both are considered in [13]; the latter is explored in Breuer [23] for algebraic languages in general.

The  $\Omega^*$  operator procedures are assumed to evaluate  $R_i[g]$ ,  $R_i * R_j$ ,  $\pi_L(R_i)$  and  $R_i[A \div B]R_j$  in time  $O(n_i)$ ,  $O(n_i n_j)$ ,  $O(n_i \log_2 n_i)$  and  $O(n_i \log_2 n_i)$  respectively (where  $n_i$  is the size of  $R_i$ ).

## 4.2. Transformation Rules

Eleven transformation rules are derived in this section through propositions and constructive proofs. The reader is forewarned that not all of these rules represent efficiency translations: applicability is considered separately in 4.3.

### 4.2.1. Strong Equivalence Transformations

(4.9) Proposition: If  $T = (R_i[g_1])[g_2] \in \xi(S, \Omega^*)$ , then there exists a function  $g_3$  such that  $T' = R_i[g_3] \in \xi(S, \Omega^*)$  and  $T'$  is strongly equivalent to  $T$ .

Proof: Let  $g_3 = g_1 \wedge g_2$ .

$$\begin{aligned} T &= \{x : x \in R_i[g_1] \wedge g_2(x) = 1\} \\ &= \{x : x \in R_i \wedge g_1(x) = 1 \wedge g_2(x) = 1\} \\ &= \{x : x \in R_i \wedge g_3(x) = 1\} \\ &= R_i[g_3]. \end{aligned}$$

(4.10) TR1:  $(R_i[g_1])[g_2] \rightarrow R_i[g_3]$ .

(4.11) Proposition: If  $T = \pi_{L_1}(\pi_{L_2}(R_i)) \in \xi(S, \Omega^*)$ , then there exists a domain identifying list  $L_3$  such that  $T' = \pi_{L_3}(R_i) \in \xi(S, \Omega^*)$  and  $T'$  is strongly equivalent to  $T$ .

Proof: Assume  $L_1 = a_1, \dots, a_k$  and  $L_2 = b_1, \dots, b_j$ .

Then  $1 \leq b_p \leq \deg(R_i)$  for  $p = 1, \dots, j$

and  $1 \leq a_q \leq j$  for  $q = 1, \dots, k$ .

Let  $L_3 = b_{a_1}, \dots, b_{a_k}$  so  $1 \leq b_{a_p} \leq \deg(R_i)$  for  $p = 1, \dots, k$ . Therefore  $L_3$  is a domain

identifying list for  $R_i$  and

$$\begin{aligned} T &= \{x[a_1, \dots, a_k] : x \in \pi_{L_2}(R_i)\} \\ &= \{r[b_{a_1}, \dots, b_{a_k}] : r \in R_i\} \\ &= \pi_{L_3}(R_i). \end{aligned}$$

(4.12) TR2:  $\pi_{L_1}(\pi_{L_2}(R_i)) \rightarrow \pi_{L_3}(R_i)$ .

(4.13) Proposition: If  $T = (R_i[A \div B]R_j)[C \div D]R_k \in \xi(S, \Omega^*)$ , then there exist domain identifying lists  $E, F$  such that  $T' = R_i[E \div F](R_j * R_k) \in \xi(S, \Omega^*)$  and  $T'$  is strongly equivalent to  $T$ .

Proof: Assume  $\deg(R_i) = m$ ,  $\deg(R_j) = n$ ,  $R_i, R_j, R_k \neq \phi$ ,

$$\begin{aligned}
A &= a_1, \dots, a_p & \bar{A} &= G = g_1, \dots, g_{m-p} \\
B &= b_1, \dots, b_p \\
C &= c_1, \dots, c_q & \bar{C} &= H = h_1, \dots, h_{m-p-q} \\
D &= d_1, \dots, d_q.
\end{aligned}$$

$$\begin{aligned}
R_1[A \div B]R_j &= \{y[\bar{A}] : y \in R_1 \wedge \forall u \in R_j \exists v \in R_1 \\
&\quad (y[\bar{A}] = v[\bar{A}] \wedge v[A] = u[B])\}
\end{aligned}$$

Therefore

$$\begin{aligned}
T &= \{x[\bar{C}] : x \in R_1[A \div B]R_j \wedge \forall s \in R_k \exists t \in R_1[A \div B]R_j \\
&\quad (x[\bar{C}] = t[\bar{C}] \wedge t[C] = s[D])\} \\
&= \{(y[\bar{A}])[\bar{C}] : y \in R_1 \wedge \forall s \in R_k \exists t \in R_1 \forall u \in R_j \exists v \in R_1 \\
&\quad ((y[\bar{A}])[\bar{C}] = (t[\bar{A}])[\bar{C}] \wedge (t[\bar{A}])[C] = s[D] \wedge \\
&\quad t[\bar{A}] = v[\bar{A}] \wedge v[A] = u[B])\} \\
&= \{(y[\bar{A}])[\bar{C}] : y \in R_1 \wedge \forall s \in R_k \exists t \in R_1 \forall u \in R_j \exists v \in R_1 \\
&\quad ((y[\bar{A}])[\bar{C}] = (v[\bar{A}])[\bar{C}] \wedge (v[\bar{A}])[C] = s[D] \wedge \\
&\quad t[\bar{A}] = v[\bar{A}] \wedge v[A] = u[B])\}.
\end{aligned}$$

Now,  $\exists t \in R_1 \forall u \in R_j \exists v \in R_1 (t[\bar{A}] = v[\bar{A}])$  is identically true by letting  $v$  be  $t$ , so that

$$\begin{aligned}
T &= \{(y[\bar{A}])[\bar{C}] : y \in R_1 \wedge \forall s \in R_k \forall u \in R_j \exists v \in R_1 ((y[\bar{A}])[\bar{C}] = (v[\bar{A}])[\bar{C}] \\
&\quad \wedge v[A](v[\bar{A}])[C] = u[B]s[D])\}
\end{aligned}$$

$$\text{Let } E = a_1, \dots, a_p, g_{c_1}, \dots, g_{c_q} \quad \bar{E} = g_{h_1}, \dots, g_{h_{m-p-q}}$$

$$F = b_1, \dots, b_p, d_1 + n, \dots, d_q + n.$$

Note that  $u[B]s[D] = (us)[F]$

and  $v[A](v[\bar{A}])[C] = v[E]$

and  $(y[\bar{A}])[\bar{C}] = y[\bar{E}]$ , so we can rewrite

$$\begin{aligned}
T &= \{y[\bar{E}] : y \in R_i \wedge \forall z \in (R_j * R_k) \exists v \in R_i (y[\bar{E}] = v[\bar{E}] \wedge v[E] = z[F])\} \\
&= R_i[E \div F](R_j * R_k).
\end{aligned}$$

$$(4.14) \quad \text{TR3} : (R_i[A \div B]R_j)[C \div D]R_k \rightarrow R_i[E \div F](R_j * R_k)$$

(4.15) Proposition: If  $T = R_i[A \div B]\pi_L(R_j) \in \xi(S, \Omega^*)$ , then there exists a domain identifying list  $C$  such that  $T' = R_i[A \div C]R_j \in \xi(S, \Omega^*)$  and  $T'$  is strongly equivalent to  $T$ .

Proof: Assume  $B = b_1, \dots, b_p$      $L = d_1, \dots, d_q$ .

Let  $C = d_{b_1}, \dots, d_{b_p}$ .

$$\begin{aligned}
T &= \{r[\bar{A}] : r \in R_i \wedge \forall s \in \pi_L(R_j) \exists t \in R_i \\
&\quad (r[\bar{A}] = t[\bar{A}] \wedge t[A] = s[B])\}
\end{aligned}$$

$$\begin{aligned}
&= \{r[\bar{A}] : r \in R_i \wedge \forall u \in R_j \exists t \in R_i \\
&\quad (r[\bar{A}] = t[\bar{A}] \wedge t[A] = u[C])\}
\end{aligned}$$

$$= R_i[A \div C]R_j.$$

$$(4.16) \quad \text{TR4} : R_i[A \div B]\pi_L(R_j) \rightarrow R_i[A \div C]R_j.$$

(4.17) Proposition: If  $T = (R_i[A \div B]R_j)[g_1] \in \xi(S, \Omega^*)$ , then there exists a function  $g_2$  such that  $T' = R_i[g_2][A \div B]R_j \in \xi(S, \Omega^*)$  and  $T'$  is strongly equivalent to  $T$ .

Proof: Assume  $\text{deg}(R_i) = m$

$$A = a_1, \dots, a_k \quad \bar{A} = C = c_1, \dots, c_{m-k}$$

$$g_1 = g_1(r[1], \dots, r[m-k]).$$

$$\begin{aligned}
\text{Let } g_2(x) &= g_2(x[1], \dots, x[m]) = g_1(x[c_1], \dots, x[c_{m-k}]) \\
&= g_1(x[\bar{A}]).
\end{aligned}$$

$$\begin{aligned}
T &= \{r : r \in R_1[A \div B]R_j \wedge g_1(r) = 1\} \\
&= \{x[\bar{A}] : x \in R_1 \wedge \forall s \in R_j \exists t \in R_1 \\
&\quad (x[\bar{A}] = t[\bar{A}] \wedge t[A] = s[B]) \wedge g_1(x[A]) = 1\}.
\end{aligned}$$

$$\begin{aligned}
\text{For } r, t \in R_1 \quad r[\bar{A}] = t[\bar{A}] &\Rightarrow g_1(r[\bar{A}]) = g_1(t[\bar{A}]) \\
&\Rightarrow g_2(r) = g_2(t)
\end{aligned}$$

So that we have

$$\begin{aligned}
T &= \{x[\bar{A}] : x \in R_1 \wedge \forall s \in R_j \exists t \in R_1 \\
&\quad (x[\bar{A}] = t[\bar{A}] \wedge t[A] = s[B] \wedge g_2(x) = 1 \wedge g_2(t) = 1)\} \\
&= \{x[\bar{A}] : x \in R_1[g_2] \wedge \forall s \in R_j \exists t \in R_1[g_2] \\
&\quad (x[\bar{A}] = t[\bar{A}] \wedge t[A] = s[B])\} \\
&= R_1[g_2][A \div B]R_j.
\end{aligned}$$

$$(4.18) \quad \text{TR5: } (R_1[A \div B]R_j)[g_1] \rightarrow R_1[g_2][A \div B]R_j.$$

(4.19) Proposition: If  $(\pi_L(R_1))[g_1] \in \xi(S, \Omega^*)$ , then there exists a function  $g_2$  such that  $T' = \pi_L(R_1[g_2]) \in \xi(S, \Omega^*)$  and  $T'$  is strongly equivalent to  $T$ .

Proof: Assume  $\deg(R_1) = m$ ,  $L = a_1, \dots, a_k$  and

$$g_1 = g_1(r[1], \dots, r[k]).$$

$$\text{Let } g_2(x) = g_2(x[1], \dots, x[m]) = g_1(x[a_1], \dots, x[a_k]) = g_1(x[A]).$$

$$\begin{aligned}
T &= \{r : r \in \pi_L(R_1) \wedge g_1(r) = 1\} \\
&= \{x[L] : x \in R_1 \wedge g_2(x) = 1\} \\
&= \pi_L(R_1[g_2]).
\end{aligned}$$

(4.20) TR6:  $(\pi_L(R_i)[g_1] \rightarrow \pi_L(R_i[g_2])).$

(4.21) Proposition: If  $T = (R_i * R_j)[A \div B]R_k \in \xi(S, \Omega^*)$ , then

- (1) there exists a domain identifying list C  
s.t.  $T' = R_i * (R_j[C \div B]R_k) \in \xi(S, \Omega^*)$ ,
- or (2) there exists a domain identifying list D  
s.t.  $T' = (R_i[D \div B]R_k) * R_j \in \xi(S, \Omega^*)$ ,
- or (3) there exist domain identifying lists E, F, G, H  
s.t.  $T' = (R_i[E \div F]R_k) * (R_j[G \div H]R_k) \in \xi(S, \Omega^*)$   
and  $T'$  is strongly equivalent to  $T$ .

Proof: Assume  $\deg(R_i) = m$ ,  $A = a_1, \dots, a_p$  and  $B = b_1, \dots, b_p$

Case 1 holds if  $\min\{a_q : 1 \leq q \leq p\} > m$ ,

Case 2 holds if  $\max\{a_q : 1 \leq q \leq p\} \leq m$ , and

Case 3 holds if  $\min\{a_q : 1 \leq q \leq p\} \leq m$   $\max\{a_q : 1 \leq q \leq p\} > m$ .

We prove only for Case 3; the other cases follow easily.

Assume that for A,  $a_1 < a_2 < \dots < a_p$ . (If this is not the case, we could permute A and B so that it were true.)

Let  $p' = \max\{q : 1 \leq q \leq p \wedge a_q \leq m\}$

$$E = a_1, \dots, a_{p'}, \quad F = b_1, \dots, b_{p'}$$

$$G = a_{p'+1}^{-m}, \dots, a_p^{-m} \quad H = b_{p'+1}, \dots, b_p$$

$$T = \{x[\bar{A}] : x \in R_i * R_j \wedge \forall y \in R_k \exists z \in R_i * R_j \\ (x[\bar{A}] = z[\bar{A}] \wedge z[A] = y[B])\}$$

$$= \{(st)[\bar{A}] : s \in R_i \wedge t \in R_j \wedge \forall y \in R_k \exists u \in R_i \exists v \in R_j \\ ((st)[\bar{A}] = (uv)[\bar{A}] \wedge (uv)[A] = y[B])\}$$

Now,  $\bar{A} = g_1, \dots, g_{m+n-p}$  so that if we let

$$\bar{E} = g_1, \dots, g_{m-p'}$$

$$\text{and } \bar{G} = g_{m-p'+1}^{-m}, \dots, g_{m+n-p}^{-m}.$$

then for  $s \in R_i$ ,  $t \in R_j$ ,  $y \in R_k$  we have

$$(st)[\bar{A}] = s[\bar{E}]t[\bar{G}]$$

$$(st)[A] = s[E]t[G]$$

$$y[B] = y[F]y[H]$$

Therefore

$$\begin{aligned} T &= \{s[\bar{E}]t[\bar{G}] : s \in R_i \wedge t \in R_j \wedge \forall y \in R_k \exists u \in R_i \exists v \in R_j \\ &\quad (s[\bar{E}] = u[\bar{E}] \wedge u[E] = y[F] \wedge t[\bar{G}] = v[\bar{G}] \wedge v[G] = y[H])\} \\ &= \{s[\bar{E}]t[\bar{G}] : s \in R_i \wedge t \in R_j \wedge \\ &\quad \forall y \in R_k \exists u \in R_i (s[\bar{E}] = u[\bar{E}] \wedge u[E] = y[F]) \wedge \\ &\quad \forall y \in R_k \exists v \in R_j (t[\bar{G}] = v[\bar{G}] \wedge v[G] = y[H])\} \\ &= \{s't' : s' \in R_i[E \div F]R_k \wedge t' \in R_j[G \div H]R_k\} \\ &= (R_i[E \div F]R_k) * (R_j[G \div H]R_k) = T' \text{ for case 3.} \end{aligned}$$

In case 1,  $C = a_1^{-m}, \dots, a_p^{-m}$  and in case 2,  $D = A$ .

$$(4.22) \quad \text{TR7: } (R_i * R_j)[A \div B]R_k \rightarrow R_i * (R_j[C \div B]R_k),$$

$$\text{or } (R_i[D \div B]R_k) * R_j,$$

$$\text{or } (R_i[E \div F]R_k) * (R_j[G \div H]R_k).$$

(4.23) Proposition: If  $T = (R_i[A \div B]R_j) * (R_k[C \div D]R_l) \in \xi(S, \Omega^*)$ , then there exist domain identifying lists  $E, F$  such that  $T' = (R_i * R_k)[E \div F](R_j * R_l) \in \xi(S, \Omega^*)$  and  $T'$  is strongly equivalent to  $T$ .

Proof: Assume  $\deg(R_i) = m$ ,  $\deg(R_j) = n$ ,  $\deg(R_k) = z$ ,

$$A = a_1, \dots, a_p$$

$$\bar{A} = g_1, \dots, g_{m-p}$$

$$B = b_1, \dots, b_p$$

$$C = c_1, \dots, c_q$$

$$\bar{C} = h_1, \dots, h_{n-q}$$

$$D = d_1, \dots, d_q$$



$$T = \{xy : x \in R_i[A \div B]R_j \wedge y \in R_k[C \div D]R_l\}$$

$$= \{r[\bar{A}]s[\bar{C}] : r \in R_i \wedge s \in R_k \wedge$$

$$\forall t \in R_j \exists u \in R_l (r[\bar{A}] = u[\bar{A}] \wedge u[A] = t[B]) \wedge$$

$$\forall v \in R_l \exists w \in R_k (s[\bar{C}] = w[\bar{C}] \wedge w[C] = v[D])\}$$

$$T = \{r[\bar{A}]s[\bar{C}] : r \in R_i \wedge s \in R_k \wedge \forall t \in R_j \forall v \in R_l \exists w \in R_k \exists u \in R_l$$

$$(r[\bar{A}]s[\bar{C}] = u[\bar{A}]w[\bar{C}] \wedge u[A]w[C] = t[B]v[D])\}$$

$$\text{Let } E = a_1, \dots, a_p, c_1^{+m}, \dots, c_q^{+m}$$

$$\bar{E} = g_1, \dots, g_{m-p}, h_1^{+n}, \dots, h_{n-q}^{+n}$$

$$F = b_1, \dots, b_p, d_1^{+z}, \dots, d_q^{+z}.$$

Then for  $r \in R_i$ ,  $s \in R_k$ ,  $t \in R_j$ ,  $v \in R_l$  we have

$$r[\bar{A}]s[\bar{C}] = (rs)[\bar{E}]$$

$$r[A]s[C] = (rs)[E]$$

$$t[B]v[D] = (tv)[F], \text{ so that}$$

$$T = \{(rs)[\bar{E}] : r \in R_i \wedge s \in R_k \wedge \forall t \in R_j \forall v \in R_l \exists w \in R_k \exists u \in R_l$$

$$((rs)[\bar{E}] = (uw)[\bar{E}] \wedge (uw)[E] = (tv)[F])\}$$

$$= \{x'[\bar{E}] : x' \in R_i * R_k \wedge \forall t' \in R_j * R_l \exists u' \in R_i * R_k$$

$$(x'[\bar{E}] = u'[\bar{E}] \wedge u'[E] = t'[F])\}$$

$$= (R_i * R_k)[E \div F](R_j * R_l).$$

$$(4.24) \text{ TR8: } (R_i[A \div B]R_j) * (R_k[C \div D]R_l) \rightarrow (R_i * R_k)[E \div F](R_j * R_l).$$

(4.25) Proposition: If  $T = (R_i[g_1]) * (R_j[g_2]) \in \xi(S, \Omega^*)$ , then there exists a function  $g_3$  such that

$$T' = (R_i * R_j)[g_3] \in \xi(S, \Omega^*) \text{ and } T' \text{ is strongly}$$

equivalent to  $T$ .

Proof: Assume  $\deg(R_i) = m$ ,  $\deg(R_j) = n$ , and

$$g_1 = g_1(r[1], \dots, r[m]), \text{ and}$$

$$g_2 = g_2(s) = g_2(s[1], \dots, s[n]).$$

$$\text{Let } g_3 = g_3(t) = g_3(t[1], \dots, t[m+n])$$

$$= g_1(t[1], \dots, t[m]) \wedge g_2(t[m+1], \dots, t[m+n]).$$

$$T = \{rs : r \in R_i[g_1] \wedge s \in R_j[g_2]\}$$

$$= \{rs : r \in R_i \wedge s \in R_j \wedge g_1(r) = 1 \wedge g_2(s) = 1\}$$

$$= \{t : t \in R_i * R_j \wedge g_3(t) = 1\}$$

$$= (R_i * R_j)[g_3].$$

$$(4.26) \quad \text{TR9: } (R_i[g_1]) * (R_j[g_2]) \rightarrow (R_i * R_j)[g_3].$$

#### 4.2.2. p-Strong Equivalence Transformations

The following definition is a variation of Definition 2.19 and is motivated by Remark 4.6

(4.27) Definition: Let  $S = \{R_1, \dots, R_p\}$ ,  $\Omega$  a set of relational operators and  $T_1, T_2 \in \xi(S, \Omega)$ .  $T_1$  is p-strongly equivalent to  $T_2$  if a permuting projection (Remark 2.12)  $\hat{\pi}_L(T_1)$  and  $T_2$  define precisely the same relation in  $C(S, \Omega)$  for any fixed set of relations  $S$  containing  $R_1, \dots, R_p$ .

(4.28) Proposition: If  $T = R_i * R_j \in \xi(S, \Omega^*)$ , then  $T' = R_j * R_i \in \xi(S, \Omega^*)$  and  $T'$  is p-strongly equivalent to  $T$ .

Proof: The proof is trivial. We note that if  $\deg(R_i) = m$  and  $\deg(R_j) = n$ , then  $R_i * R_j = \hat{\pi}_L(R_j * R_i)$  where  $L = n+1, \dots, n+m, 1, \dots, n$ .

$$(4.29) \quad \text{TR10: } R_i * R_j \rightarrow \hat{\pi}_L(R_j * R_i).$$

(4.30) Proposition: If  $T = \pi_{L_1}(R_i * R_j) \in \xi(S, \Omega^*)$ , then there exist domain identifying lists  $L_2, L_3$  such that  $T' = \pi_{L_2}(R_i) * \pi_{L_3}(R_j) \in \xi(S, \Omega^*)$  and  $T'$  is  $p$ -strongly equivalent to  $T$ .

Proof: Assume  $\text{deg}(R_i) = m$ ,  $\text{deg}(R_j) = n$  and  $L_1 = d_1, \dots, d_p$ .

Let  $D = d_{i_1}, \dots, d_{i_p}$  where  $d_{i_1} \leq d_{i_2} \leq \dots \leq d_{i_p}$ , and let  $E = i_1, \dots, i_p$ .

Then  $\pi_E(T) = \pi_E(\pi_{L_1}(R_i * R_j)) = \pi_D(R_i * R_j)$ .

Let  $F = i_{j_1}, \dots, i_{j_p}$  where  $i_{j_1} < i_{j_2} < \dots < i_{j_p}$

and let  $H = j_1, \dots, j_p$ .

Then  $\pi_H(\pi_E(T)) = \pi_H(\pi_E(\pi_{L_1}(R_i * R_j))) = \pi_{L_1}(R_i * R_j) = T$ .

So that  $T = \pi_H(\pi_D(R_i * R_j))$ .

Let  $q = \max\{p' : 1 \leq p' \leq p \wedge d_{i_{p'}} \leq m\}$

and  $L_2 = d_{i_1}, \dots, d_{i_q}$  and  $L_3 = d_{i_{q+1}}, \dots, d_{i_p}$  so that

for  $r \in R_i$  and  $s \in R_j$  we have  $(rs)[D] = r[L_2]s[L_3]$ .

Therefore  $\pi_D(R_i * R_j) = \{r[L_2]s[L_3] : r \in R_i \wedge s \in R_j\}$   
 $= \{r's' : r' \in \pi_{L_2}(R_i) \wedge s' \in \pi_{L_3}(R_j)\}$   
 $= \pi_{L_2}(R_i) * \pi_{L_3}(R_j)$

So that  $T = \pi_H(\pi_{L_2}(R_i) * \pi_{L_3}(R_j))$

$$(4.31) \quad \text{TR11: } \pi_{L_1}(R_i * R_j) \rightarrow \hat{\pi}_L(\pi_{L_2}(R_i) * \pi_{L_3}(R_j)).$$

#### 4.3. The Applicability of Transformation Rules

Given an expression (or subexpression) that appears on the left side of a transformation rule, the decision to apply the rule is essentially a question of whether the expression on the right side can be evaluated more quickly. If it were possible to predict with accuracy the number of tuples that would result from applying operators to data base relations, a detailed analysis of the amount of work necessary to

evaluate the expressions on each side of the transformation rules could be of value. Practical implementation considerations rule this out. Instead we rely on intuitive judgment of the consequences of the transformation rules when the data base relations are large. This somewhat subjective discussion appears in 4.3.1. In 4.3.2 we discuss two conflicts among the rules.

#### 4.3.1. Applicability

Of primary concern in deciding the applicability of a transformation rule are the number of operators and the size of the operands on each side of the rule. Rules which eliminate operators without increasing the size of operands are clearly desirable. Thus TR1 and TR2 are always good. TR3 exchanges a DIVISION for a PRODUCT, and the procedure for DIVISION in Chapter III requires that this product be sorted. By sorting  $R_j$  and  $R_k$  before producing  $R_j * R_k$ , the sort of  $R_j * R_k$  can be avoided; since  $R_j$  and  $R_k$  would have to be sorted for the expression on the left of TR3 and  $R_i[E \div F](R_j * R_k)$  is about the same work as  $R_i[A \div B]R_j$ , we consider TR3 to be a good transformation rule.

TR4 is a good rule because it eliminates a projection and an intermediate relation.

TR5 and TR6 can be considered together. Their applicability depends on the size of  $R_i[g_2]$  compared to  $R_i$ . That is, the evaluation and storage overhead for  $R_i[g_2]$  may be greater than the savings achieved by a reduction of the size of the operand for the PROJECTION or DIVISION. If we employ restriction overlap (Remark 4.7) during the sort of  $R_i$  (for DIVISION) and during the sort-and-project of  $R_i$  (for PROJECTION), the right side in both TR5 and TR6 does not require the use of an

intermediate relation. In this way, TR5 and TR6 will never increase evaluation time and often will decrease it.

TR7 involves the same operators on each side for all 3 cases, and the PRODUCT is always over smaller relations. For large relations, evaluation of  $(R_i * R_j)[A \div B]R_k$  will take more time than  $R_j[C \div B]R_k$  or  $R_i[D \div B]R_k$  or both  $R_i[E \div F]R_k$  and  $R_j[G \div H]R_k$ : Consequently TR7 is a good transformation and TR8 (which reverses TR7) is not.

TR9 is not a good transformation because restriction overlap in the generation of  $R_i[g_1]$  on the left can avoid a sequential scan of the intermediate relation  $R_j[g_2]$  for each  $r$  in  $R_i$  for which  $g_1(r) = 0$ .

TR10 does not appear to offer an increase or decrease in evaluation time. We show in 4.4 how to choose the best way to evaluate the product of 2 or more relations.

TR11 is a good transformation. The PRODUCT can require no more time on the right than on the left, and often will require less. If  $R_i$  and  $R_j$  are large,  $\pi_{L_2}(R_i)$  and  $\pi_{L_3}(R_j)$  can both be evaluated in less time than  $\pi_{L_1}(R_i * R_j)$ .

In summary, all rules except TR8 and TR9 represent efficiency translations. TR1, TR9, TR10 and TR11 are used in 4.4 to increase efficiency over expressions in  $\xi(S, \{E, *\})$ . TR2 and TR3 become highly advantageous in Chapter V. TR5 and TR6 indicate better procedures can be developed for the expressions on the right side of each. TR7 indicates a good way to perform DIVISION when the divided relation is a product. TR4 is of limited applicability.

#### 4.3.2. Conflicting Transformation Rules

Two conflicts among the desirable transformation rules have been identified. Consider for example

$$(4.32) \quad ((R_i * R_j)[A \div B]R_k)[g_1].$$

Applying TR5 to 4.32 yields

$$(4.33) \quad ((R_i * R_j)[g_2])[A \div B]R_k.$$

Applying TR7 to 4.32 yields

$$(4.34) \quad ((R_i[E \div F]R_k) * (R_j[G \div H]R_k))[g_1].$$

The discussion of the previous section indicates that 4.34 is preferable because of the large operand  $R_i * R_j$  in the DIVISION of 4.33. TR7 should thus be given precedence over TR5.

Next consider

$$(4.35) \quad (\pi_L(R_i * R_j))[g_1].$$

TR6 and TR11 respectively yield

$$(4.36) \quad \pi_{L_1}((R_i * R_j)[g_2]) \text{ and}$$

$$(4.37) \quad (\hat{\pi}_L(\pi_{L_2}(R_i) * (\pi_{L_3}(R_j)))[g_1].$$

TR11 should be given precedence over TR6 due to elimination of the projection of a product.

#### 4.4. Efficiency Translations for $\xi(S, \{E, *\})$

Evaluation of the expression

$$(4.38) \quad F = (R_1[g_1] * R_2[g_2] * R_3)[g_3]$$

can be performed by recursive application of PRODUCT and RESTRICTION as follows:

(4.39)    step 1  $T_1 = R_1[g_1]$   
           step 2  $T_2 = R_2[g_2]$   
           step 3  $T_3 = T_1 * T_2$   
           step 4  $F = (T_3 * R_3)[g_3]$

The storage and retrieval overhead of intermediate relations  $T_1$ ,  $T_2$ , and  $T_3$  in 4.39 is unnecessary. The following FORTRAN-like nested iteration avoids this overhead and represents a single-step evaluation for  $F$ :

```
(4.40)      DO 10 I1 = 1, n1
              INPUT (R1,r1)
              IF (g1(r1) = 0) GOTO 10
              DO 10 I2 = 1, n2
                INPUT(R2,r2)
                IF (g2(r2) = 0) GOTO 10
                DO 10 I3 = 1, n3
                  INPUT(R3,r3)
                  IF (g3(r1r2r3) = 0) GOTO 10
                  OUTPUT(F,r1r2r3)
                10 CONTINUE
```

(In 4.40, INPUT(X,x) retrieves the next tuple in X and stores it in variable x; OUTPUT(Y,y) stores the tuple with value in y into the relation Y. Filling and emptying of buffers is assumed to be performed by an I/O manager.)

Any  $F$  in  $\xi(S, \{E, *\})$  can be evaluated by a nested iteration of this type over the operand relations in their order of appearance in

F, with conditional transfers inserted in the obvious locations.

Remark 4.6 indicates that the nested iteration can be performed over the operand relations of F in any order, provided conditional transfers are placed properly. Different orders of iteration will be shown to require different input volumes, hence produce different response times for the same F.

$\xi(S, \{E, *\})$  may be conveniently subdivided into the two disjoint sets

$$(4.41) \quad \xi(S, \{*\})$$

and

$$(4.42) \quad \xi(S, \{E, *\}) \setminus \xi(S, \{*\}).$$

An expression in 4.41 is represented by the generic relational product

$$(4.43) \quad R_1 * R_2 * \dots * R_p.$$

4.4.1. shows how to choose the optimal order of iteration over  $R_1, \dots, R_p$  as a function of  $n_1, \dots, n_p$  and  $b_1, \dots, b_p$ .

An expression F in 4.42 can be converted by repeated application of transformation rule TR9 to the (strongly equivalent) restricted product

$$(4.44) \quad (R_1 * R_2 * \dots * R_p)[g]$$

in which "g" is a conjunction of the RESTRICTION predicates in F.

(Domain-identifying integers in these predicates must be reset since predicates are moved from their position adjacent to their original operand relation; this is trivial to perform.) In 4.3, transformation rule TR9 was shown to increase evaluation time rather than decrease it.



The inverse map of TR9, where it can be performed, is an efficiency translation. (The inverse map of TR9 is not possible for  $(R_1 * R_j)[g_1 \wedge g_2]$ , where  $g_1$  and  $g_2$  each reference attribute values of both  $r$  and  $s$  for  $rs \in R_1 * R_j$ .) Section 4.4.2 demonstrates how to choose an order of iteration over  $R_1, \dots, R_p$  for 4.44 that is expected optimal when the conditional transfers are correctly inserted. The correct insertion represents the inverse map of TR9 for

$$(4.45) \quad (R_{d_1} * \dots * R_{d_p})[g']$$

in which  $d_1, \dots, d_p$  is the order of iteration and  $g'$  is derived from  $g$  in 4.44, again by resetting domain-identifiers in  $g$  to reflect their placement in 4.44.

Projection overlap (see Remark 4.6) is assumed to be applied correctly to arrange tuples back to the original order for 4.43 and 4.44.

#### 4.4.1. Relational Products

Let  $N = \{1, \dots, p\}$ , the subscripts of relations in the generic relational product

$$(4.46) \quad R_1 * \dots * R_p$$

If  $D = d_1, \dots, d_p$  is any permutation of  $N$ , 4.46 can be evaluated by nested iteration (such as 4.40 without conditional transfers) in the order  $R_{d_1}, \dots, R_{d_p}$ . The volume of data input (in bits) for  $D$  is

$$(4.47) \quad n_{d_1} (b_{d_1} + n_{d_2} (b_{d_2} + \dots + n_{d_{p-1}} (b_{d_{p-1}} + n_{d_p} b_{d_p}) \dots))$$

which we denote by  $W(D)$ . The following example demonstrates that different values of  $D$  produce different input data volumes;

(4.48) Example: Nested iteration for  $R_1 * R_2 * R_3$  for

$$n_1 = 300 \quad n_2 = 100 \quad n_3 = 200$$

$$b_1 = 50 \quad b_2 = 100 \quad b_3 = 200$$

$$W(1,2,3) = 1,203,015,000$$

$$W(3,2,1) = 302,040,000$$

The following proposition provides a technique for selecting the optimal iteration order for 4.46.

(4.49) Proposition: Let  $N = \{1, \dots, p\}$  and  $n_1, b_1, \dots, n_p, b_p$  be positive integers. Let  $D = 1, \dots, p$  and  $E = 1, \dots, i-1, i+1, i, i+2, \dots, p$ . Then

$$W(D) \leq W(E) \iff \frac{n_i b_i}{n_i - 1} \geq \frac{n_{i+1} b_{i+1}}{n_{i+1} - 1}$$

Proof:  $W(D) \leq W(E)$

$\iff$

$$n_1 b_1 + n_1 n_2 b_2 + \dots + n_1 \dots n_p b_p$$

$$\leq n_1 b_1 + \dots + n_1 \dots n_{i-1} b_{i-1} + n_1 \dots n_{i-1} n_{i+1} b_{i+1} + n_1 \dots n_{i+1} b_i + \dots + n_1 \dots n_p b_p$$

$\iff$

$$n_i b_i (n_{i+1} - 1) \geq n_{i+1} b_{i+1} (n_i - 1)$$

$\iff$

$$\frac{n_i b_i}{(n_i - 1)} \geq \frac{n_{i+1} b_{i+1}}{(n_{i+1} - 1)}$$

By Proposition 4.49, the evaluation time of  $R_1 * \dots * R_p$  by the iteration scheme of 4.40 for an ordering  $D = d_1, \dots, d_p$  of  $N = \{1, \dots, p\}$  is minimized if and only if

$$(4.50) \quad \frac{n_{d_1} b_{d_1}}{(n_{d_1} - 1)} \geq \frac{n_{d_2} b_{d_2}}{(n_{d_2} - 1)} \geq \dots \geq \frac{n_{d_p} b_{d_p}}{(n_{d_p} - 1)} .$$

This allows the  $\Omega^*$ -EVALUATOR to choose the permutation  $R_{d_1}, \dots, R_{d_p}$  of  $R_1, \dots, R_p$  that minimizes input data volume for the evaluation of

4.46 by nested iteration.

(4.51) Example: Optimal Order of Iteration for  $R_1 * R_2 * R_3$

$$\text{where } n_1 = 300 \quad n_2 = 100 \quad n_3 = 200$$

$$b_1 = 50 \quad b_2 = 100 \quad b_3 = 200$$

$$\frac{n_1 b_1}{n_1 - 1} = 50.17, \quad \frac{n_2 b_2}{n_2 - 1} = 101.01, \quad \frac{n_3 b_3}{n_3 - 1} = 201.05.$$

So that  $R_3 * R_2 * R_1$  is optimal.

#### 4.4.2. Restricted Products

##### 4.4.2.1. Motivation and Assumptions

We consider the evaluation of the generic restricted product

$$(4.52) \quad (R_1 * \dots * R_p)[g].$$

Evaluation is assumed to take place by a nested iteration suggested by the following FORTRAN-like code:

```
(4.53)      DO 200 I1 = 1, n1
              INPUT(R1, r1)
              DO 200 I2 = 1, n2
                INPUT(R2, r2)
                .
                .
                .
```

```

      .
      .
      .
DO 200 Ip = 1, np
INPUT(Rp, rp)
t = r1r2...rp
IF(g(t) = 0) GOTO 200
OUTPUT(F, t)

200 CONTINUE

```

Input data volume for 4.53 is

$$(4.54) \quad n_1 b_1 + n_1(n_2 b_2 + n_2(\dots + n_{p-1}(n_p b_p) \dots)).$$

Suppose that in  $(R_1 * \dots * R_p)[g]$ , the function  $g$  is converted to conjunctive normal form. That is,

$$(4.55) \quad g = g_1 \wedge \dots \wedge g_k.$$

Then  $g_i$  references one or more domains of  $R_1 * \dots * R_p$  and it is trivial to detect which domains are from which operand relations since their degrees are assumed known.

(4.56) Definition: For 4.54 and  $i = 1, \dots, k$

$$H_{g_i} = \{j : \text{one or more domains of } R_j \text{ are} \\ \text{referenced by } g_i\}$$

Suppose  $H_{g_1} = \{1\}$ . Then the schema in 4.53 can be altered to the following where  $h_1$  is derived from  $g_1$ :

```

(4.57)      DO 200 I1 = 1, n1
              INPUT(R1, r1)
              IF(h1(r1) = 0) GOTO 200
              DO 200 I2 = 1, n2
                .
                .
                .
              DO 200 Ip = 1, np
                INPUT(Rp, rp)
                t = r1r2...rp
                IF(g2(t) = 0 ∨ ... ∨ gk(t) = 0) GOTO 200
                OUTPUT(F, t)
              200 CONTINUE

```

If  $P_{g_1}$  is the probability that any tuple in  $R_1$  is in  $R_1[g_1]$ , then the expected data volume for 4.57 is

$$(4.58) \quad n_1 b_1 + n_1 P_{g_1} (n_2 b_2 + n_3 (\dots + n_{p-1} (n_p b_p) \dots)).$$

This represents a savings over 4.53 of

$$(4.59) \quad (1 - P_{g_1}) (n_2 b_2 + n_3 (\dots + n_{p-1} (n_p b_p) \dots)).$$

This value is never negative since  $0 \leq P_{g_1} \leq 1$  and represents a substantial reduction for small  $P_{g_1}$  and large  $n_i$  and  $b_i$ .

The nested iteration corresponding to 4.53 can be arranged to iterate over  $R_1, \dots, R_p$  in any order, just as in 4.4.1. The resulting iteration should be converted to a schema like 4.57 in which functions corresponding to  $g_1, \dots, g_k$  are applied as early as possible. This is essentially the

inverse map of TR9 and is formalized in the following definition and remark.

(4.60) Definition: Let  $F = (R_1 * \dots * R_p)[g_1 \wedge \dots \wedge g_k]$ ,  $N = \{1, \dots, p\}$  and  $G = \{g_1, \dots, g_k\}$ . For the ordering  $D = d_1, \dots, d_p$  of  $N$ , define  $G_i(D)$  for  $i = 1, \dots, p$  as follows:

$$G_i(D) = \{g_j : g_j \in G \wedge_{H_{g_j}} \not\subseteq \{d_1, \dots, d_{i-1}\} \\ \wedge_{H_{g_j}} \subseteq \{d_1, \dots, d_i\}\}$$

(4.61) Remark: For the nested iteration of  $(R_1 * \dots * R_p)[g_1 \wedge \dots \wedge g_k]$  in order  $D = d_1, \dots, d_p$ , functions in  $G_i(D)$  should be applied immediately following the input of tuples from  $R_{d_i}$ .

This implies that any ordering of  $N$  uniquely determines the placement of conditional tests for each function  $g_i$  in 4.55. We represent the nested iteration for  $D$  by the sequence

$$(4.62) \quad R_{d_1}, G_1(D), R_{d_2}, G_2(D), \dots, R_{d_p}, G_p(D).$$

Let  $P_{g_i} =$  the probability that a tuple in  $R_1 * \dots * R_p$  is in  $(R_1 * \dots * R_p)[g_i]$ .

And for  $G' \subseteq G = \{g_1, \dots, g_k\}$ , define  $P(G')$  as follows:

$$P(G') = \begin{cases} 1 & \text{if } G' = \phi \\ P_{g_1} P_{g_2} \dots P_{g_t} & \text{if } G' = \{g_1, \dots, g_t\} \end{cases}$$

If  $P_{g_i}$  and  $P_{g_j}$  are independent for  $i \neq j$ , then the expected data input volume for 4.62 is

$$(4.63) \quad n_{d_1} b_{d_1} + n_{d_1} P(G_1(D)) (n_{d_2} b_{d_2} + n_{d_2} P(G_2(D)) (\dots + n_{d_{p-1}} P(G_{p-1}(D)) (n_{d_p} b_{d_p} + n_{d_p} P(G_p(D)) \dots))).$$

For a given  $G$  and known  $P_{g_1}, \dots, P_{g_k}$ , an ordering  $D$  of  $N$  uniquely determines 4.62 and 4.63, which we denote by  $A(D;G)$  and  $W(D;G)$ . We will show how to derive the ordering  $D'$  of  $N$  that minimizes  $W(D;G)$ . To do so requires that the  $P_{g_i}$  are known and are independent. For implementation purposes, the  $P_{g_i}$  can be approximated. They are generally not expected to be independent, so the technique we present is very approximate. The result is included because it is of theoretical interest and because it provides improvement in the absence of more effective techniques to evaluate  $(R_1 * \dots * R_p)[g]$ .

In the sequel, we assume that  $g$  is in the conjunctive normal form " $g_1 \wedge \dots \wedge g_k$ ", and that  $P_{g_1}, \dots, P_{g_k}$  are known and independent.

The following example illustrates that under the assumptions for  $P_{g_1}, \dots, P_{g_k}$  (i.e. known and independent), different orderings of  $N$  require different input data volumes.

(4.64) Example: Nested iteration for  $(R_1 * R_2 * R_3)[g_1 \wedge g_2 \wedge g_3 \wedge g_4]$

$$\text{where } n_1 = 300 \quad n_2 = 100 \quad n_3 = 200$$

$$b_1 = 50 \quad b_2 = 100 \quad b_3 = 200$$

$$\begin{array}{ll} \text{and } H_{g_1} = \{2\}, & P_{g_1} = .15, \\ H_{g_2} = \{2,3\}, & P_{g_2} = .15, \\ H_{g_3} = \{1,3\}, & P_{g_3} = .5, \\ H_{g_4} = \{3\}, & P_{g_4} = .9. \end{array}$$

So that

$$A(3,1,2;G) = R_3, \{g_4\}, R_1, \{g_3\}, R_2, \{g_1, g_2\} \text{ and}$$

$$W(3,k,2;G) = 272,740,000.$$

$$A(3,2,1;G) = R_3, \{g_4\}, R_2, \{g_1, g_2\}, R_1, \{g_3\} \text{ and}$$

$$W(3,2,1;G) = 7,915,000.$$

4.4.2.2. Technique for  $(R_1 * \dots * R_p)[g_1 \wedge \dots \wedge g_k]$

Assume  $N = \{1, \dots, p\}$  and  $G = \{g_1, \dots, g_k\}$ .  $H_{g_1}, \dots, H_{g_k}$  can be computed. If  $P_{g_1}, \dots, P_{g_k}$  are known and independent, then for any ordering  $D = d_1, \dots, d_p$  of  $N$ ,  $G_1(D), \dots, G_p(D)$  and  $P(G_1(D)), \dots, P(G_p(D))$  can be computed also. We seek an ordering  $D'$  of  $N$  such that

$$W(D'; G) \leq \min\{W(D; G) : D \text{ is an ordering of } N\}.$$

Consider the identity ordering  $D = 1, 2, \dots, p$  of  $N$ , and the ordering  $E = 1, \dots, i-1, i+1, i, i+2, \dots, p$ . The analysis that follows paraphrases the proof of Proposition 4.49.

$$W(D; G) \leq W(E; G)$$

$\Leftrightarrow$

$$\begin{aligned} & n_1 b_1 + n_1 P(G_1(D)) n_2 b_2 + \dots + n_1 P(G_1(D)) \dots n_{p-1} P(G_{p-1}(D)) n_p b_p \\ & \leq n_1 b_1 + \dots + n_1 P(G_1(E)) \dots n_{i-1} P(G_{i-1}(E)) n_{i+1} b_{i+1} \\ & \quad + n_1 P(G_1(E)) \dots n_{i-1} P(G_{i-1}(E)) n_{i+1} P(G_i(E)) n_i b_i \\ & \quad + n_1 P(G_1(E)) \dots n_{i-1} P(G_i(E)) n_i P(G_{i+1}(E)) n_{i+2} b_{i+2} \\ & \quad + \dots + n_1 P(G_1(E)) \dots n_{p-1} P(G_{p-1}(E)) n_p b_p \end{aligned}$$

$D$  and  $E$  are identical in every position except the  $i$ th and  $i+1$ st.

Therefore, for  $j = 1, \dots, i-1, i+2, \dots, p$  we have

$$G_j(D) = G_j(E) \quad \text{and} \quad P(G_j(D)) = P(G_j(E)).$$

And from Definition 4.60 and the fact that

$$P(G')P(G'') = P(G' \cup G'')$$

when  $G' \cap G'' = \emptyset$ , it follows that



$$G_i(D) \cup G_{i+1}(D) = G_i(E) \cup G_{i+1}(E)$$

and

$$P(G_i(D) \cup G_{i+1}(D)) = P(G_i(E) \cup G_{i+1}(E)).$$

So that

$$W(D;G) \leq W(E;G)$$

$$\Leftrightarrow$$

$$n_i b_i + n_i P(G_i(D)) n_{i+1} b_{i+1} \leq n_{i+1} b_{i+1} + n_{i+1} P(G_i(E)) n_i b_i$$

$$\Leftrightarrow$$

$$n_{i+1} b_{i+1} (n_i P(G_i(D)) - 1) \leq n_i b_i (n_{i+1} P(G_i(E)) - 1)$$

$$\Leftrightarrow$$

$$\frac{n_{i+1} b_{i+1}}{(n_{i+1} P(G_i(E)) - 1)} \leq \frac{n_i b_i}{(n_i P(G_i(D)) - 1)}$$

This implies that if  $D' = d_1, \dots, d_p$  minimizes  $W(D;G)$ , then

$$(4.65) \quad \frac{n_1 b_1}{(n_1 P(G_1(D')) - 1)} \leq \dots \leq \frac{n_p b_p}{(n_p P(G_p(D')) - 1)} .$$

This yields a rapid means of sequentially selecting  $e_1, \dots, e_p$  for  $D'$ , the optimal ordering of  $N$ . Recall from Definition 4.60 that

$$G_i(D) = \{g_j : g_j \in G \wedge H_{g_j} \not\subseteq \{d_1, \dots, d_{i-1}\} \\ \wedge H_{g_j} \subseteq \{d_1, \dots, d_i\}\}.$$

So that  $G_i(D) = G_i(d_1, \dots, d_i)$  which is independent of  $d_{i+1}, \dots, d_p$ . That is, it is sufficient to know  $d_1, \dots, d_i$  in order to determine  $G_i(D)$  for any ordering  $D$  with the initial sequence  $d_1, \dots, d_i$ .

The sequential selection of  $e_1, \dots, e_p$  proceeds in  $p$  stages where  $e_q$  is chosen in the  $q$ th stage for  $q = 1, \dots, p$ . The selection rule for  $e_q$  in the  $q$ th stage is:

(4.66)

Set  $e_q$  to  $i_0$  in  $N \setminus \{e_1, \dots, e_{q-1}\}$  to maximize

$$\frac{n_i b_i}{(n_i^P(G(e_1, \dots, e_{q-1}, i)) - 1)}$$

(4.67) Example: Selection of the optimal (expected) iteration for the expression in Example 4.64.

STAGE 1

$$\frac{n_1 b_1}{n_1 - 1} = 50.17$$

$$\frac{n_2 b_2}{n_2^P g_1 - 1} = 714.29$$

$$\frac{n_3 b_3}{n_4^P g_4 - 1} = 223.46$$

SET  $e_1 = 2$ ;

STAGE 2

$$\frac{n_1 b_1}{n_1 - 1} = 50.17$$

$$\frac{n_3 b_3}{n_3^P g_2 g_4 - 1} = 1538.46$$

SET  $e_2 = 3$ ;

STAGE 3

SET  $e_3 = 1$ ;This yields  $D' = 2, 3, 1$ ;
 $A(2, 3, 1; G) = R_2, \{g_1\}, R_3, \{g_2, g_4\}, R_1, \{g_3\}$  and

 $W(2, 3, 1; G) = 6,685.000$  (compare to 4.64).

#### 4.5. Summary

This chapter presented a collection of techniques for reducing response time in an  $\Omega^*$ -EVALUATOR. With the model, measure and  $\Omega^*$  operator procedures of Chapter III assumed, it was shown that any  $F \in \xi(S, \Omega^*)$  can be evaluated by recursively applying these procedures to data base relations or intermediate relations formed in the evaluation of  $F$ . Evaluation time for  $F$  is reduced by translating  $F$  to  $F' \in \xi(S, \Omega^*)$  where  $F$  is strongly equivalent or strongly equivalent up to permutation to  $F'$ , and the evaluation of  $F'$  by the recursive application of operators requires less time than direct evaluation of  $F$ .

For the class of expressions  $\xi(S, \{E, *\})$  involving only RESTRICTION and PRODUCT, it was shown that evaluation can proceed without the storage and retrieval overhead of intermediate relations. Evaluation of  $F \in \xi(S, \{E, *\})$  can be performed by a nested iteration over the operand relations of  $F$  in their order of appearance (left-to-right) in  $F$ . Moreover, the ability to overlap a permuting projection with the generation of tuples of a relation permits this nested iteration to proceed in any order over the operand relations of  $F$ . Simple techniques are provided to select the order of iteration that is optimal or expected optimal with respect to the measure.

### 5.1. Introduction

While an exact characterization of the relation-defining capability of the  $\Omega^*$  operators has never been established, an important step in this direction is an algorithm due to Codd [6] which translates relation-defining expressions in the non-algebraic retrieval language ALPHA to equivalent definitions that are relational algebraic expressions. This translatability implies that the relation-defining capability of the algebra is at least as great as ALPHA, and this may be seen to fall just short of a first-order predicate calculus for finite sets.

A secondary benefit of this algorithm is that any relational data base system supporting retrievals specified in the algebra can also support ALPHA with an interface procedure that implements the translation algorithm. This approach was in fact used by Palermo [12] who referred to the algorithm as the Codd Reduction Algorithm (CRA). The details of ALPHA and the CRA may be found in [6,12]. What is of concern here is the set of algebraic expressions generated by the CRA and how these may be evaluated efficiently. In keeping with Palermo's terminology, we refer to these as CRA expressions.

Every CRA expression is of the form

$$(5.1) \quad \pi_L(G_{p+1}(\dots G_{p+q}((R_{i_1} * \dots * R_{i_{p+q}})[g])\dots))$$

where  $R_{i_1}, \dots, R_{i_{p+q}}$  are data base relations,  $p \geq 1$ ,  $q \geq 0$  and  $G_{p+1}, \dots, G_{p+q}$  are each a PROJECTION or DIVISION. That is, they represent a sequence of PROJECTIONS and DIVISIONS to a restricted product (see 4.4).

A number of efficiency techniques from Chapter IV are applicable

to 5.1, specifically transformation rules TR2 and TR3, and the expected optimal iterative evaluation technique of 4.4 for

$$(5.2) \quad (R_{i_1} * \dots * R_{i_{p+q}}) [g].$$

This chapter demonstrates an evaluation technique for 5.1 that proceeds by forming 5.2 in such a way that the conditions described in 3.6 and 3.7 for  $O(n)$  PROJECTION and DIVISION are met. Consequently, the  $q$  PROJECTIONS and DIVISIONS  $G_{p+1}, \dots, G_{p+q}$  in 5.1 can proceed more rapidly than if  $O(n \log_2 n)$  procedures were required.

The technique is presented in 5.2. It conflicts slightly with results of Chapter IV; these conflicts are described and resolved in 5.3 and 5.4 is a summary.

## 5.2. $O(n)$ PROJECTION and DIVISION for CRA Expressions

We consider only the generic CRA expression

$$(5.3) \quad \pi_L(G_{p+1}(\dots G_{p+q}((R_1 * \dots * R_{p+q})[g])\dots))$$

in which  $q \geq 0$  and for  $j = 1, \dots, q$

$$G_{p+j}(X) \quad \text{is} \quad \pi_{L_{p+j}}(X)$$

$$\text{or} \quad X[N_{p+j} \div M_{p+j}]R_{p+j}$$

$$\text{in which} \quad L_{p+j} = 1, \dots, \mu_{p+j}$$

$$N_{p+j} = \mu_{p+j} + 1, \dots, \mu_{p+j} + \text{deg}(R_{p+j})$$

$$M_{p+j} = 1, \dots, \text{deg}(R_{p+j}).$$

and  $\mu_{p+j} = \text{deg}(R_1) + \dots + \text{deg}(R_{p+j-1})$ . We consider the evaluation of 5.3 to proceed as:

$$\begin{aligned}
 (5.4) \quad T_q &= (R_1 * \dots * R_{p+q})[g] \\
 T_{q-1} &= G_{p+q}(T_q) \\
 &\vdots \\
 T_0 &= G_{p+1}(T_1) \\
 W &= \pi_L(T_0).
 \end{aligned}$$

Note that  $L_{p+j} = \bar{N}_{p+j}$  relative to  $T_j$ . Propositions 3.5 and 3.9 state that the PROJECTION or DIVISION  $G_{p+j}(T_j)$  can be performed in time proportional to the number of tuples in  $T_j$  if

(5.5) "every subset of  $T_j$  with the same value of  $r[L_{p+j}]$  is consecutively retrievable, and the tuples of each subset are retrievable in the same order as the tuples of  $R_{p+j}$ ."

The technique is based on an evaluation procedure for  $T_q, \dots, T_1$  that achieves this condition.

Consider the case when "g" is identically true. Then  $T_j = R_1 * \dots * R_{p+j}$  for  $j = 0, \dots, q$ , and  $T_j = G_{p+j+1}(T_{j+1})$  for  $j = 0, \dots, q-1$ . This is true regardless of whether  $G_{p+j+1}$  is a PROJECTION or DIVISION due to the fact that

$$\begin{aligned}
 (5.6) \quad T_j &= \pi_{L_{p+j+1}}(T_j * R_{p+j+1}) \\
 &= (T_j * R_{p+j+1})[N_{p+j+1} \div M_{p+j+1}]R_{p+j+1}
 \end{aligned}$$

(where  $L_{p+j+1}$ ,  $N_{p+j+1}$  and  $M_{p+j+1}$  are as in 5.3).

If  $T_q$  is generated as  $(R_1 * \dots * R_p) * (R_{p+1} * (\dots * (R_{p+q}) \dots))$ , then for

$$\begin{aligned}
 \text{every } t \in T_{q-1} &= R_1 * \dots * R_{p+q} \\
 &= \pi_{L_{p+q}}(T_q) \\
 &= \pi_{N_{p+q}}(T_q),
 \end{aligned}$$

the subset  $t * R_{p+q}$  appears sequentially in  $T_q$  in the same order as the tuples of  $R_{p+q}$  (providing  $R_{p+q}$  has not been altered). So that the conditions of 5.5 are achieved, and therefore  $T_{q-1} = G_{p+q}(T_q)$  can be evaluated in time proportional to the number of tuples in  $T_q$ . If the tuples of  $T_{q-1}$  are stored in the same order they are encountered in forming  $T_{q-1} = G_{p+q}(T_q)$  and  $R_{p+q-1}$  is unaltered, then the conditions of 5.5 are again met and  $T_{q-2} = G_{p+q-1}(T_{q-1})$  can be produced in time proportional to the number of tuples in  $T_{q-1}$ . Continuing in this fashion (e.g. storing tuples in the order encountered and not altering  $R_{p+q-2}, \dots, R_{p+1}$ ),  $T_{q-3}, \dots, T_0$  can each be produced in time proportional to the number of tuples in  $T_{q-2}, \dots, T_1$ .

When  $g$  is any 0-1 function, the same argument applies to the subsets of  $R_1 * \dots * R_{p+q}, \dots, R_1 * \dots * R_{p+1}$ , so that again,  $G_{p+q}(T_q), \dots, G_{p+1}(T_1)$  can be produced in time proportional to the number of tuples in the operand relation. Only for the last projection " $W = \pi_L(T_0)$ " in 5.4 can the conditions in 5.5 fail.

The importance of this technique is that  $O(n)$  procedures for every PROJECTION and DIVISION are available without any of the overhead for sorting operand relations that would be required for  $O(n \log_2 n)$  PROJECTION and DIVISION. The savings in time are significant for any  $q > 0$  when  $R_1, \dots, R_{p+q}$  are large.

### 5.3. Conflict and Resolution with other Efficiency Techniques

Two efficiency techniques of Chapter IV conflict with the technique of the previous section. Transformation rules TR2 and TR3 (see 4.2, 4.3) indicate that any sequence of PROJECTIONS or of DIVISIONS in  $G_{p+q}, \dots, G_{p+1}$  can be replaced with a single PROJECTION or a single DIVISION. The restricted product formation technique (see 4.4) specifies a different iteration order for  $T_q = (R_1 * \dots * R_{p+q})[g]$  than in 5.2. Use of these efficiency techniques may preclude achieving  $O(n)$  PROJECTION and DIVISION for 5.3. In this section we explore these conflicts and derive resolutions.

#### 5.3.1. Resolution with TR2 and TR3

Suppose that in 5.4 we have for some "j"

$$(5.7) \quad T_{j-1} = \pi_{L_{p+j}}(T_j)$$

$$T_{j-2} = \pi_{L_{p+j-1}}(T_{j-1})$$

or

$$(5.8) \quad T_{j-1} = T_j [N_{p+j} \overset{\circ}{\div} M_{p+j}] R_{p+j}$$

$$T_{j-2} = T_{j-1} [N_{p+j-1} \overset{\circ}{\div} M_{p+j-1}] R_{p+j-1}.$$

Transformation rules TR2 and TR3 indicate that 5.7 can be replaced by

$$(5.9) \quad T_{j-2} = \pi_{L_{p+j-1}}(T_j)$$

and that 5.8 can be replaced by

$$(5.10) \quad T_{j-2} = T_j [E \overset{\circ}{\div} F] (R_{p+j-1} * R_{p+j})$$

$$\text{where } E = N_{p+j-1}, N_{p+j}$$

$$\text{and } F = 1, \dots, \deg(R_{p+j-1}) + \deg(R_{p+j}).$$



Notice that relative to  $T_j$ ,  $L_{p+j-1} = \bar{E}$ . If  $T_q$  has been formed as  $(R_1 * \dots * R_p) * (R_{p+1} * (\dots * (R_{p+q}) \dots))$  and the techniques of 5.2 for  $G_{p+q}, \dots, G_{p+j-1}$  followed, then the condition for  $O(n)$  PROJECTION and DIVISION are achieved for 5.9 and 5.10 provided the tuples of  $R_{p+j-1} * R_{p+j}$  are retrieved as  $R_{p+j-1} * (R_{p+j})$ .  $T_{j-2}$  in 5.9 or 5.10 can be produced in time proportional to the number of tuples in  $T_j$ . Note also that the conditions for  $O(n)$  evaluation of  $T_{j-3}, \dots, T_0$  are not disturbed.

Therefore, TR2 and TR3 can still be used for 5.3 provided the divisor relation for each DIVISION is retrieved properly. The resolution generalizes for 3 or more consecutive DIVISIONS or PROJECTIONS in 5.3.

### 5.3.2. Resolution with Restricted Product Result

The algorithm 4.4.2 indicates that

$$(5.11) \quad T_q = ((R_1 * \dots * R_p) * (R_{p+1} * (\dots * (R_{p+q}) \dots))) [g]$$

may be formed in less time by altering the order of iteration and inserting tests corresponding to  $g_1, \dots, g_k$  where  $g = g_1 \wedge \dots \wedge g_k$ . Generating  $T_q$  by this technique destroys the conditions for  $O(n)$  PROJECTION and DIVISION described in 5.2.

A resolution is possible between these two techniques:

Let  $G = \{g_1, \dots, g_k\}$ . Recall that

$$H_{g_j} = \{i : R_i \text{ is referenced in at least one domain by } g_j\}.$$

(see Definition 4.56) If we let

$$G' = \{g_j : H_{g_j} \subseteq \{1, \dots, p\}\},$$

then the technique of 4.4 can be used in the formation of

$$(R_1 * \dots * R_p) [h_1 \wedge \dots \wedge h_m]$$

where  $G' = \{h_1, \dots, h_m\}$ . This yields a sequence  $D' = d_1, \dots, d_p$  for  $R_1, \dots, R_p$ . Letting  $E' = d_1, \dots, d_p, p+1, \dots, p+q$ , the correct placement of tests for every function in  $G$  is determined uniquely for  $E'$ . Then the formation of  $T_q$  as

$$((R_d * \dots * R_d) * (R_{p+1} * (\dots * (R_{p+q}) \dots))) [g]$$

with tests appropriately placed for  $g_1, \dots, g_k$  saves evaluation time without sacrificing the conditions of 5.5 for  $O(n)$  PROJECTION and DIVISION. Although the most efficient way to form  $T_q$  may be excluded, the extra time for  $O(n \log_2 n)$  PROJECTION and DIVISION is avoided.

#### 5.4. Summary

This chapter has demonstrated an efficiency technique for the CRA expressions in  $\xi(S, \Omega^*)$ . These are in fact the expressions generated by the Codd Reduction Algorithm from the non-algebraic retrieval language ALPHA (over relations  $S$ ) to  $\xi(S, \Omega^*)$ . These expressions become particularly important when retrievals can be specified using ALPHA (or a retrieval language like ALPHA) in place of or in addition to relational algebraic expressions.

Although the technique precludes the full savings that might be achieved by choosing the expected optimal order of iteration to evaluate the intermediate relation

$$(R_{i_1} * \dots * R_{i_{p+q}}) [g],$$

that occurs in every CRA expression, the benefit of performing the PROJECTIONS and DIVISIONS  $G_{p+1}, \dots, G_{p+q}$  in 5.3 by  $O(n)$  procedures instead of  $O(n \log_2 n)$  procedures is expected to more than offset this lost savings.

## Chapter VI. SUMMARY AND FUTURE WORK

6.1. Summary

This thesis has addressed the problem of efficient retrieval from a relational data base. The problem is formalized as the efficient evaluation of expressions in a relational algebra which define new relations from a fixed set of relations. The operation of a relational data base system is considered to be the explicit formation of relations defined by algebraic expressions presented as interactions by the users of such a system. The thesis is a collection of techniques for performing this function efficiently.

Chapter I is an introduction. The study begins in Chapter II with an investigation of operators which map relations to relations and provide the means for the definition of new relations. For a given set of relations  $S$ , any set of relational operators  $\Omega$  induces an algebra containing  $S$  and every relation obtainable by repeated application of the operators to the relations in  $S$ . The set of all algebraic expressions over  $S$  and  $\Omega$  is designated  $\xi(S, \Omega)$  and can be considered to be the definitions of all the relations in the algebra. Designating by  $C(S, \Omega)$  the set of relations that are so defined, the retrieval function is cast as the implementation of the well-defined mapping from  $\xi(S, \Omega)$  to  $C(S, \Omega)$ . A collection of procedures for implementing this map is termed an  $\Omega$ -EVALUATOR, and the problem is thus reduced to the design of an efficient  $\Omega$ -EVALUATOR for an appropriate  $\Omega$ .

Chapter II is essentially a search for an appropriate  $\Omega$ . The appropriateness of a set  $\Omega$  of relational operators is determined by the richness of  $C(S, \Omega)$ , the set of relations that can be defined. By

considering and comparing ten operators, it is shown that the operator set  $\Omega^*$  consisting of only 4 operators (RESTRICTION, PRODUCT, PROJECTION, DIVISION) provides a relation-defining capability sufficient for many data retrieval applications. Thus the problem is further reduced to the implementation of an efficient  $\Omega^*$ -EVALUATOR.

The problem is considered in the environment of a conventional computer system with a simple, uniform storage representation for relations. The description of the computer system, storage representation and a measure for the running time of procedures in this environment appear in Chapter III and remain in effect through Chapters IV and V.

Chapter III investigates procedures for the application of the logical operations of RESTRICTION, PRODUCT, PROJECTION and DIVISION to the physical representation of the data base relations. Procedures are derived whose running times vary as the size of the relations and are asymptotically as good or better than any known procedures that perform the same functions. Additional results in Chapter III show that even more rapid procedures for PROJECTION and DIVISION are possible when the storage representations of the data base relations are known to possess certain order properties.

Chapter IV begins by showing how procedures for RESTRICTION, PRODUCT, PROJECTION and DIVISION permit the evaluation of any  $F$  in  $\xi(S, \Omega^*)$  provided facilities are available for the storage and retrieval of intermediate relations. A collection of transformation rules are given and it is shown that applying these rules to an expression  $F$  in  $\xi(S, \Omega^*)$  yields a second expression  $F'$  in  $\xi(S, \Omega^*)$  that defines precisely the same relations. Most of these rules have the property that  $F'$  can be evaluated in less time than  $F$ . A different technique is presented for a subset of

$\xi(S, \Omega^*)$  consisting of the set of expressions involving just the RESTRICTION and PRODUCT operators. Each such expression can be evaluated by a nested iteration over the operand relations in any order. Simple decision procedures are provided which choose the order of iteration that is optimal or expected optimal with respect to the measure.

Chapter V deals with an important subset of  $\xi(S, \Omega^*)$  called the CRA expressions. These expressions result from the translation of relation-defining expressions in the non-algebraic retrieval language ALPHA to equivalent definitions in  $\xi(S, \Omega^*)$ . They are important in any system which supports retrieval languages like ALPHA in place of or in addition to retrievals specified in a relational algebra. The technique that is presented shows that the conditions (described in Chapter III) for more rapid PROJECTION and DIVISION can be achieved in the intermediate results of the evaluation at no expense in time. Minor conflicts with two efficiency results from Chapter IV are resolved.

These results demonstrate that mechanical conversion of non-procedurally specified interactions to efficient procedural solutions is possible in a relational data base system, at least for the single function of retrieval.

## 6.2. Future Work

Since the investigation is limited to the single function of retrieval, extensions are required before the results can have practical application. First, the functions of data addition, deletion and modification must be considered because a general purpose data base system must support them. Second, other representations for the data base relations need to be explored. Each function will dictate a "best"

storage representation, and the actual choice will undoubtedly be a compromise among them.

Secondary indexes can often be used to advantage in large data bases; an important line of investigation would be to consider all the data base system functions with respect to a linear storage representation supported by secondary indexes.

It is hoped that the results of this thesis for the single function of retrieval and for the linear storage representation will aid in future research in the design of efficient relational data base systems.

## References

- [1] Codd, E.F., "A Relational Model of Data for Large Shared Data Banks," CACM 13, 6, June 1970.
- [2] Date, C.J., An Introduction of Database Systems, Addison-Wesley, 1975.
- [3] Martin, J., Computer Data-Base Optimization, Prentice-Hall, 1975.
- [4] IMS/360, Version 2, System/Application Design Guide, IBM Sh20-0910-3, 1972.
- [5] CODASYL, "Data Base Task Group Report," ACM, October 1969.
- [6] Codd, E.F., "Relational Completeness of Data Base Sublanguages," Courant Computer Science Symposium 6, May 1972; also available as IBM Research Report RJ987.
- [7] \_\_\_\_\_, "A Data Base Sublanguage Founded on the Relational Calculus," Proc. 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control, San Diego, 1971; also available as IBM Research Report RJ893.
- [8] Chamberlin, D.D., and R. F. Boyce, "SEQUEL: A Structured English Query Language," Proc. 1974 ACM-SIGFIDET Workshop on Data Description, Access and Control, Ann Arbor, May 1974.
- [9] Boyce, R.F., D.D. Chamberlin, M.M. Hammer and W.F. King III, "Specifying Queries As Relational Expressions," Proc. ACM SIGPLAN-SIGIR Interface Meeting, Gaithersburg, Maryland, November 1973; also available as IBM Research Report RJ1291.
- [10] Rothnie, J.B., "The Design of a Generalized Data Management System," Ph.D. Dissertation, Dept. of Civil Engineering, MIT, September 1972.
- [11] Stonebraker, M.R. and E. Wong, "INGRES - A Relational Data Base System," ERL Report #M477, University of California, Berkeley, November 1974.

- [12] Palermo, F.P., "A Data Base Search Problem," Fourth International Symposium on Computer and Information Science, Miami Beach, November 1973; also available as IBM Research Report RJ1072.
- [13] Smith, J.M. and P.Y. Chang, "Optimizing the Performance of a Relational Data Base Interface," (Scheduled to appear in) CACM 18, 10, October 1975.
- [14] Held, G.D., M.R. Stonebraker and E. Wong, "INGRES - A Relational Data Base System," Proc. 1975 National Computer Conference, vol. 44, AFIPS, 1975.
- [15] Czarnik, B., S. Schuster and D. Tsichritzis, "ZETA: A Relational Data Base Management system," Proc. ACM PACIFIC '75 Conference, San Francisco, April 1975.
- [16] Chamberlin, D.D., J.N. Gray and I.L. Traiger, "Views, Authorization, and Locking in a Relational Data Base System," Proc. 1975 National Computer Conference, vol. 44, AFIPS, 1975.
- [17] Knuth, D., The Art of Computer Programming, vol. 3, Sorting and Searching, Addison-Wesley, 1973.
- [18] Earley, J., "High Level Operations in Automatic Programming," Computer Science Technical Report #22, University of California, Berkeley, October 1973.
- [19] Schwartz, J.T., "Abstract Algorithms and a Set Theoretic Language for their Expression," New York University, 1970-71.
- [20] Held, G. and M.R. Stonebraker, "Storage Structures and Access Methods in the Relational Data Base Management System INGRES," Proc. ACM PACIFIC '75 Conference, San Francisco, April 1975.
- [21] Schkolnick, M., "Secondary Index Optimization," Proc. 1975 ACM-SIGMOD Workshop on Management of Data, San Jose, May 1975.



- [22] Rivest, R.L., "Analysis of Associative Retrieval Algorithms,"  
IRIA Report #54, IRIA-LABORIA, France, February 1974.
- [23] Breuer, M.A., "Generation of Optimal Code for Expressions via  
Factorization," CACM 12, 6, June 1969.

Copyright © 2013, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

SUCCESSIVE APPROXIMATION ANALOG-TO-DIGITAL CONVERSION  
TECHNIQUES IN MOS INTEGRATED CIRCUITS

by

James Leo McCreary

Memorandum No. ERL-554

9 October 1975

SUCCESSIVE APPROXIMATION ANALOG--TO--DIGITAL CONVERSION  
TECHNIQUES IN MOS INTEGRATED CIRCUITS

by

James Leo McCreary

Memorandum No. ERL-M554

9 October 1975

ELECTRONICS RESEARCH LABORATORY

College of Engineering  
University of California, Berkeley  
94720

## DEDICATION

This manuscript is dedicated to my wife Virginia for her patience and encouragement throughout our years of graduate study.

## ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to Professor P.R. Gray for his patient supervision and ideas throughout the course of this work and to Professor D.A. Hodges for his special guidance and assistance. The recommendations of Professor L.F. Donaghey are also appreciated. In addition the author wishes to thank J. Albarran and G. Smarandoiu for their contributions and suggestions. The discussions with R. Suarez, R. Coen and R. Heald were also helpful. Special thanks go to D. McDaniel and D. Rogers for their technical support in the I.C. lab. The author also expresses his gratitude to Virginia for drafting the figures contained in this manuscript.

## TABLE OF CONTENTS

	PAGE NO.
CHAPTER I INTRODUCTION	1
CHAPTER II PRINCIPLES AND METHODS OF A/D CONVERSION	6
2.1 Introduction	6
2.2 Principles of A/D Conversion	6
2.2.1 Quantization Theory	6
2.2.2 Characterization of the Digital Output	10
2.2.3 Conversion Dynamics	18
2.3 Techniques for D/A Conversion	25
2.3.1 Summation of Binary Weighted Currents	26
2.3.2 Binary Attenuation of Equal Currents	29
2.3.3 Charge Redistribution	29
2.3.4 Integration Types	33
2.4 Techniques for A/D Conversion	36
2.4.1 Serial Methods	36
2.4.2 Successive Approximation	40
2.4.3 Parallel Conversion	41
2.5 Technologies for ADC Components	44
2.5.1 Introduction	44
2.5.2 ADC Component Technology for Charge- Redistribution and Integration Methods	44
2.5.3 ADC Component Technology for Precision Resistor Networks.	45
2.5.4 Comparator	47
2.5.5 Digital Logic and Control	48
2.5.7 Supporting Functions	48

	PAGE NO.
2.5.7 Summary	48
CHAPTER III ALL-MOS, SUCCESSIVE APPROXIMATION, WEIGHTED CAPACITOR, ANALOG-TO-DIGITAL CONVERSION TECHNIQUE -- RADCAP	49
3.1 Introduction	49
3.2 Successive Approximation A/D Conversion	49
3.2.1 A Comparison of Successive Approximation Method and Other Techniques	49
3.2.2 The Successive Approximation Algorithm	50
3.2.3 Precision Component Requirements for DACs	52
3.3 Factors Influencing the Choice of Technology for a Monolithic Realization of a Successive Approximation ADC	52
3.3.1 Advantages of MOS Realization	52
3.3.2 Realization of Precision Attenuator Networks Compatible with MOS Technology	53
3.3.3 VATCAP -- An MOS DAC	55
3.4 A/D Conversion Using Charge-Redistribution on Weighted Capacitors -- RADCAP	58
3.5 Summary	65
CHAPTER IV FACTORS LIMITING ACCURACY IN THE RADCAP CLASS OF ADCs	66
4.1 Introduction	66
4.2 MOS Comparator Input Offset Voltage Cancellation	66
4.3 Effects of Capacitance from the Capacitor Plates to Ground	71



	PAGE NO.
4.4 Temperature Coefficient of Capacitance	78
4.5 Voltage Coefficient of Capacitance	83
4.6 Dielectric Relaxation	85
4.7 Leakage Currents	93
4.8 Parameter Drift	93
4.9 Capacitor Ratio Errors in RADCAP	95
4.9.1 Capacitor versus Resistor Matching	95
4.9.2 Nonlinearity Due to Ratio Errors	97
4.9.3 Uniform Undercut	101
4.9.4 Oxide Gradient	104
4.9.5 Non-uniform Undercut	104
4.9.6 Mask Alignment	108
4.9.7 Errors Due to Interconnect	108
4.9.8 Fringing	110
4.10 Intrinsic Offset	110
CHAPTER V FACTORS LIMITING THE CONVERSION RATE IN RADCAP	115
5.1 Introduction	115
5.2 Factors Limiting the Minimum Acquisition Time When VATCAP is Used as a S/H Circuit	115
5.2.1 Relationship Between Acquisition Time and Sampling Accuracy	115
5.2.2 Minimum Acquisition Time With Offset Cancellation Technique	119
5.3 Factors Limiting the Minimum C/R Time for RADCAP Class of Circuits	122
5.4 Factors Causing Comparator Delay	124

	PAGE NO.
5.5 The Theoretical Minimum Conversion Time	127
5.6 Practical Limitations on Conversion Rate	128
CHAPTER VI DESCRIPTION OF AN EXPERIMENTAL ADC	129
6.1 Introduction	129
6.2 Optimization of MOS Capacitor Geometry	129
6.3 MOS Comparator Realization	132
6.4 Logic Circuit Design	138
6.5 Summary	139
CHAPTER VII EXPERIMENTAL RESULTS	140
7.1 Introduction	140
7.2 Experimental Results of IC1	140
7.2.1 Design of Circuit Layout for IC1	140
7.2.2 Threshold Voltage for the N-MOS Device in IC1	141
7.2.3 Sources of Error for IC1 Due to Fabrication Procedures	143
7.2.4 Data From Capacitance Bridge Measurements for IC1	144
7.2.5 Operation of IC1 in ADC System	148
7.3 Fabrication Modifications Required to Correct Errors Discovered in IC1	150
7.4 Layout Modifications Required to Correct Errors in IC1	151
7.5 Experimental Results for IC2	154
7.5.1 N-MOS Device Parameters for IC2	156
7.5.2 Measurement of Systematic Error in Capacitor Ratios	161

	PAGE NO.
7.5.3 Elimination of Systematic Error by Mask Trim	163
7.5.4 Measurement of Transition Point Voltages	167
7.5.5 Experimental Measurement of Performance Parameters	169
7.5.6 Limitations on Matching Accuracy Due to Random Edge Location	176
CHAPTER VIII CONCLUSION	187
APPENDIX A CALCULATION OF NONLINEARITY DUE TO CAPACITOR VOLTAGE COEFFICIENT	189
APPENDIX B DIGITAL LOGIC CIRCUIT AND EXPERIMENTAL SYSTEM INTERCONNECTIONS	192
APPENDIX C N-MOS ALUMINUM GATE FABRICATION PROCESS	204
APPENDIX D CALCULATION OF CAPACITOR PLATE DUPLICATION SIZE NEEDED FOR UNDERCUT INSENSITIVITY	208
REFERENCES	212

Successive Approximation Analog-to-Digital Conversion  
Techniques in MOS Integrated Circuits

Copyright © 1975

James Leo McCreary

SUCCESSIVE APPROXIMATION ANALOG-TO-DIGITAL CONVERSION  
TECHNIQUES IN MOS INTEGRATED CIRCUITS

Ph.D. James Leo McCreary

Dept. of Electrical  
Engineering and  
Computer Sciences

  
Chairman of Committee

ABSTRACT

This research effort has been directed towards the exploration of new methods of analog-to-digital conversion which are suitable for monolithic construction. A new technique has been developed which utilizes charge-redistribution among binary weighted capacitors. An experimental integrated circuit, fabricated to test the new approach, performed analog-to-digital conversion in 23  $\mu$ s with an accuracy of 10 bits  $\pm$  1/2 of the least significant bit.

## CHAPTER I

INTRODUCTION

Of all the man-made methods of processing information, the fastest, most efficient and least expensive per operation is by electronic circuits. Such circuitry is usually classified as analog or digital depending upon the representation of the information it contains. An analog quantity may assume any level between two extremes and may be continuous in that region but a digital function is only permitted to have a set of discrete values which are usually separated by forbidden regions. The generalized electronic information processing machine, shown in Figure 1.1 must be capable of data flow into or out of the machine according to some control. It may also have processing units and memories. Although much information could be processed by either digital or by analog methods, some data processing is more suited to one particular method. Of course neither technique is best suited for all cases.

The rapid advancement of digital systems over the last 2 decades has been due to increasing applications that require digital methods. Some advantages of digital techniques are higher accuracies and greater immunity to noise. Large computers today are capable of maintaining precisions of the order of 1 part in  $10^{18}$ . In contrast to analog circuits, this accuracy is not degraded or distorted by sequential operations (except for round-off error). Furthermore, digital memory systems have provided more practical means of nearly infinite storage capacity in printed, punched, or magnetic forms. Digital techniques would probably be preferred in applications requiring extensive computations, data manipulation or data management. However, both analog and digital circuits are fabricated in the same basic technologies and are therefore fundamentally capable of the same high operating speeds. Excluding analog circuits requiring coils or transformers both forms can

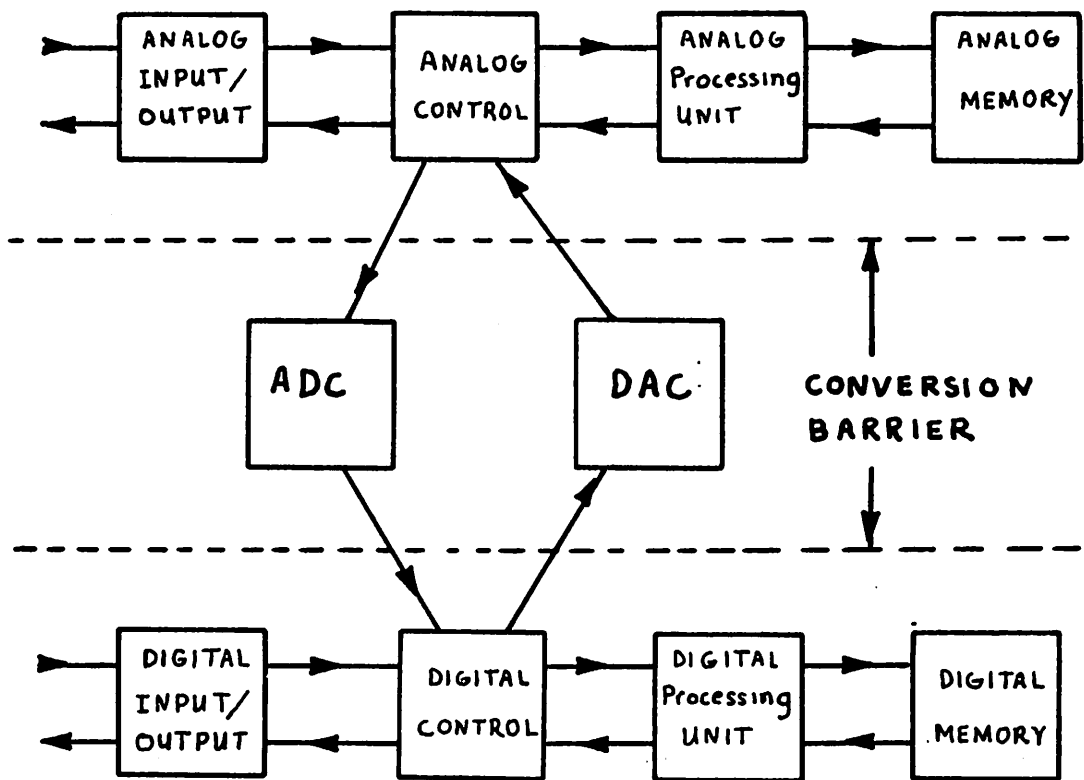


Figure 1.1: Generalized information processing machine.

generally be fabricated as integrated circuits (I.C.s), thereby realizing the advantages of small size, less power consumption, better reliability and lower costs. However, at present there is a greater availability of digital system I.C. components. A high-speed memory can be constructed to the user's specifications by wiring together several widely available I.C.s. Central processors are even available as single units.

We live in a world of analog functions. Data, usually originating as an analog quantity, is transcribed originally by people or by machines into a digital form by a process called analog-to-digital (A/D) conversion. The conversion is a prerequisite for digital processing of analog information. It should be pointed out that the accuracy of such processing cannot be more significant than the accuracy of conversion and that real-time processing cannot occur faster than the converter sampling rate. Two important qualities of the converter are, therefore, the accuracy and the conversion rate. Unfortunately both of these qualities are difficult to achieve without special fabrication techniques which are usually expensive. It is not surprising that the limitations and cost of conversion have inhibited full potential development of the generalized machine of Figure 1. That is, most information processing machines do not freely transit the conversion barrier as would be optimal conceptually. The converters are the only informational links between the two processing sections. Furthermore, the division between them is usually maintained on the premise that once the digital number is available it is more efficient to maintain it in that form until all processing is completed. This has been generally true. Although newer conversion techniques have improved both speed and accuracy, the cost of conversion has remained very high compared with the cost of other functions and therefore conversion has either been minimized or avoided.



It has been the focal point of this research to develop a substantially lower cost A/D converter (ADC) having both a high accuracy and a fast sampling rate. Such a new A/D conversion technique has been developed which can result in low cost circuits having high production yields. The new technique utilizes charge redistribution between binary weighted capacitors and a successive approximation method which rapidly converges an analog voltage into a binary number. In addition a standard N-channel MOS process can be used to implement this new algorithm and to fabricate on the same silicon chip the ADC as well as any additional logic which may comprise a larger system. The feasibility of the approach was investigated by fabricating an experimental integrated circuit which, when supported by a discrete logic system, simulated a complete ADC. The new type of ADC may be described as an "all-MOS, successive approximation, charge Redistribution ADC utilizing weighted CAPacitors" or RADCAP. The test circuit required only 23  $\mu$ s to perform an accurate 10-bit binary conversion thereby verifying the new conversion algorithm [1].

An implied requirement for a lower cost circuit is that it be fabricated using only the standard processing steps required by all circuits. Any special procedures are usually expensive and therefore should be avoided. Hence only conventional photomasking techniques were used in the new design. Since the accuracy of the converter depends strongly upon the ability to construct precisely matched capacitors, the standard photolithographic process was identified as the ultimate limitation upon the conversion accuracy. This was established by gathering extensive data on capacitor ratio accuracy.

In Chapter II the principles and methods of A/D conversion will be discussed. This section is intended to be a survey of only the common

conventional methods and to provide a theoretical background necessary for understanding the new technique.

In Chapter III the successive approximation method of conversion is discussed along with the basic framework for "a MOS charge-redistribution precision Voltage Attenuator utilizing CAPacitors" or VATCAP. A conceptual model of the new RADCAP technique is then illustrated with VATCAP being a component.

The factors limiting accuracy in RADCAP techniques are examined in great detail in Chapter IV. The investigation of these sources of error represents a major effort of this research.

The factors which limit conversion rate for the RADCAP technique are discussed in Chapter V.

The experimental MOS I.C. and the digital logic system are discussed in Chapter VI and the circuit schematics are also given. In addition a precision MOS capacitor design is presented.

In Chapter VII the measurements from the first experimental I.C. and the subsequent design modifications leading to a second I.C. are examined. An experimental set-up is described in which the performance of the new A/D converter was evaluated.

The statement of conclusions from this research is made in Chapter VIII.

## CHAPTER II

Principles and Methods of A/D Conversion2.1 Introduction

The A/D conversion process consists of generalized D/A conversion and comparison operations which convert an analog input into a digitally encoded form utilizing some particular algorithm. In this chapter the most common algorithms and circuit methods of quantization will be examined.

2.2 Principles of A/D Conversion2.2.1 Quantization Theory

The process of A/D conversion consists of two basic operations: generalized D/A conversion and comparison. This division is chosen because common functional blocks are not usually found in all types of converters. In general, however, N-bit ADCs require at least N comparison operations which may be performed in sequence or simultaneously by one or more comparators. Therefore the generalized DAC is the group of circuits which performs all functions other than those of comparison. The basic ADC is shown in a block diagram in Figure 2.1. A generalized DAC usually contains a DAC, digital logic circuits, and perhaps linear amplifiers. The function of this unit can best be understood by examining the digital output code. Assume that hereafter the analog input is a voltage level and that the digital output is binary. Although both of these assumptions are often true, they need not necessarily be true. In linear binary code, the N-bit number,  $B(N) = b_{N-1} b_{N-2} \cdots b_1 b_0$ , where  $b_i$  equals 1 or 0, specifies  $2^N$  binary numbers from zero through  $2^N - 1$  inclusive and is bounded by  $2^N$ . For a uniform ADC a linear correspondence or proportionality exists between a precise analog voltage  $V_i$  and a unique binary number  $B_i$

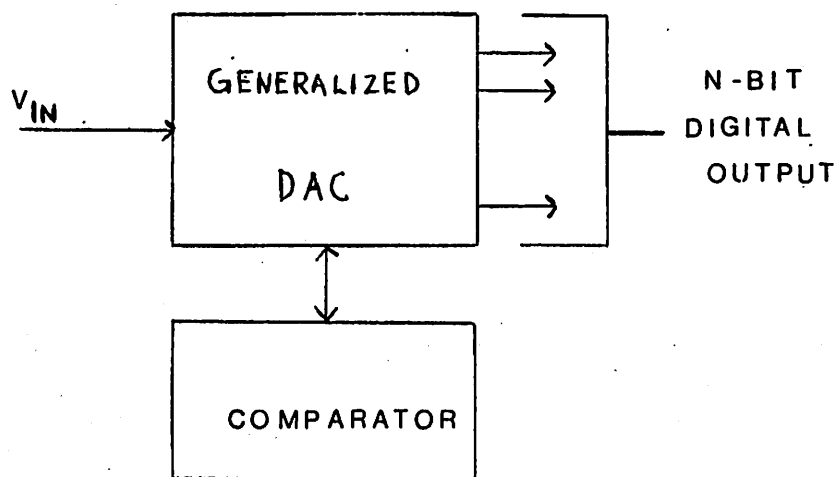


Figure 2.1: Block diagram of an ADC.

such that  $0 \leq V_i < V_R$  and  $0 \leq B_i < 2^N$  where  $i = 0, 1, \dots, (2^N - 1)$ . All input voltages  $V_i$  must be less than  $V_R$  which corresponds to  $2^N$ , the upper bound. Actually,  $V_R$  is a reference voltage which may be generated internally by the converter or else is supplied externally. External references are often chosen to be 5 or 10 volts.

At this point it has been established that for a linear converter a linear relationship must exist between a set of binary numbers  $B_i$  and a set of unique analog values  $V_i$ . Since the analog input  $V_{IN}$  is usually a continuous function, while the binary numbers are a set of discrete values, there must be an uncertainty in the quantization of the input. This uncertainty is a necessary result of correlating a continuous function and a discrete function. Exploring this further, if  $V_R \rightarrow 2^N$ , then  $\frac{V_R}{2^N} \rightarrow 1$ . The symbol " $\rightarrow$ " implies linear correspondence or proportionality. Hence a voltage change of  $\frac{V_R}{2^N}$  is required for a digital change of unity, or conversely,

$$(B_{i+1} - B_i = 1) \rightarrow (V_{i+1} - V_i = \frac{V_R}{2^N}).$$

The smallest change in analog input which can be resolved by a change in binary value is equal to  $\frac{V_R}{2^N}$  and this is defined as the resolution of the converter. Again this is an inherent limitation associated with the process of quantizing a continuous-valued function into a discrete-valued function having a finite number of states.  $\frac{V_R}{2^N}$  may also be called the unit of quantization. Therefore, assuming a uniform or linear encoder the set of values of  $V_i$  may now be computed:

$$V_i = \frac{V_R}{2^N} B_i \text{ for } B_i = 0, 1, \dots, (2^N - 1).$$

A greater question still remains, however, and that concerns the relationship between the real continuous input  $V_{IN}$  and the discrete output  $B_i$ . The answer may be found by examining a segment of an ideal ADC transfer function,  $B_j$  vs  $V_{IN}$ , which is shown in Figure 2.2. The correspondence between  $V_j$  and  $B_j$  for  $j = i-1, i, i+1$  is shown. A transition from  $B_{i-1}$  to  $B_i$  must occur somewhere between  $V_{i-1}$  and  $V_i$  and this is designated  $V_{T_i}$ . A similar transition occurs at  $V_{T_{i+1}}$ . Then  $B_i \rightarrow V_{IN}$  such that  $V_{T_i} \leq V_{IN} \leq V_{T_{i+1}}$ . The quantizing error  $\epsilon$  for  $V_{IN}$  through one unit of quantization from  $V_{i-1} \leq V_{IN} \leq V_i$  is:

$$\epsilon_i = -V_{IN} + V_i \text{ for } B_i$$

and 
$$\epsilon_i = -V_{IN} + V_{i-1} \text{ for } B_{i-1}.$$

For an ideal quantizer the worst case error  $\epsilon_j$ , where  $j = 1, 2, \dots, (2^N - 1)$ , is minimized if all  $\epsilon_j$  are equal in magnitude to the same value  $\epsilon_q$ .

Evaluating the equations for  $V_{IN} = V_{T_i}$  in the interval from  $V_{i-1}$  to  $V_i$ ,

$$\epsilon_q = -V_{T_i} + V_i$$

and 
$$-\epsilon_q = -V_{T_i} + V_{i-1}, \text{ from which}$$

$$\epsilon_q = \frac{V_i - V_{i-1}}{2} = \frac{1}{2} \frac{V_R}{2^N}.$$

$\epsilon_q$  is defined as the maximum quantization error for an ideal, linear converter; and, as demonstrated, this value of error occurs at each transition  $i$ . Of course the particular magnitude of quantization error for any given value of  $V_{IN}$  could be less than  $\epsilon_q$  or even zero if  $V_{IN} = V_i$ . Although  $\epsilon_q$  is smaller for larger  $N$ , it can never be less than 1/2 of the resolution. Thus for an ideal converter, an output of  $B_i$  would imply an analog input

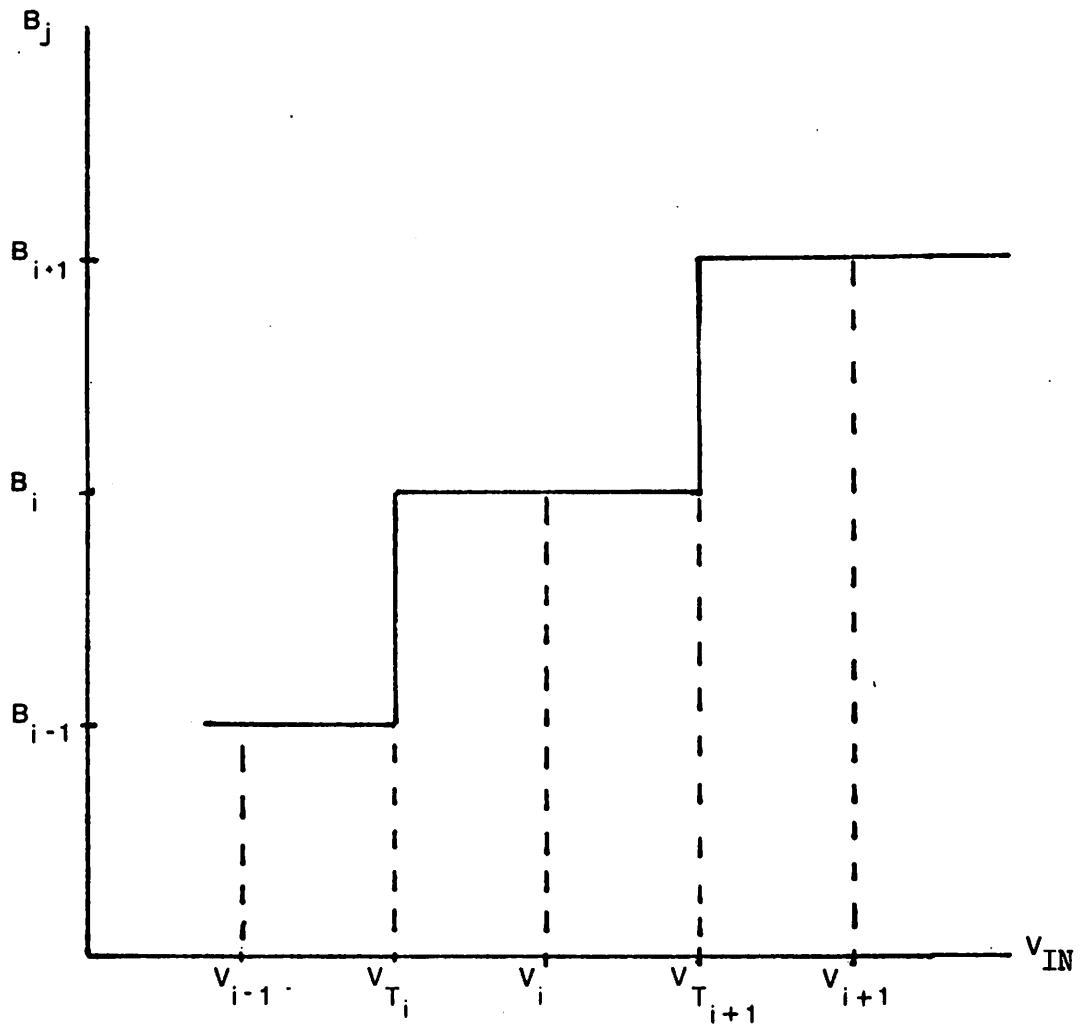


Figure 2.2: A segment of an ideal ADC transfer function illustrating the correspondence between analog input  $V_{IN}$ , the transition voltage  $V_j$  and the digital output  $B_j$ , for  $j = i - 1, i, \text{ and } i + 1$ .

uncertainty of  $\pm \frac{1}{2} \frac{V_R}{2^N}$  :

$$B_i \rightarrow V_i \pm \frac{1}{2} \frac{V_R}{2^N} = V_{IN} = B_i \frac{V_R}{2^N} \pm \frac{1}{2} \frac{V_R}{2^N} = (B_i \pm \frac{1}{2}) \frac{V_R}{2^N} .$$

Moreover, the digital output of an ideal ADC can be no more accurate than  $\pm \frac{1}{2}$  of the least significant bit (LSB). Hence the accuracy of the converter is no better than  $\pm \frac{1}{2}$  LSB even in the ideal case. The transfer function and the quantization error of an ideal linear converter are plotted in Figure 2.3 for a 3-bit converter with a 10 V reference. From observation of states  $B_0$  (000) and  $B_{2^{N-1}}$  (111), the ideal transfer function is characteristically asymmetric at its end points since

$$B_0 \rightarrow V_{IN} \text{ such that } 0 \leq V_{IN} \leq + \epsilon_q$$

but  $B_{2^{N-1}} \rightarrow V_{IN}$  such that

$$(V_{2^{N-1}} - \epsilon_q) \leq V_{IN} \leq (V_{2^{N-1}} + 2 \epsilon_q)$$

Therefore the magnitude of the ideal quantization error of all digital states is no larger than  $\epsilon_q$  except for  $B_{2^{N-1}}$  in which the error may be  $2 \epsilon_q$ . This is a result of exceeding the linear input range of the converter which evidently terminates at  $V_{IN} = V_{2^{N-1}} + \epsilon_q$ .

In conclusion the generalized DAC and comparator must operate upon the analog input voltage thereby resulting in the transfer function illustrated in Figure 2.3.

### 2.2.2 Characterization of the Digital Output

The ideal uniform ADC was characterized by specifying the reference



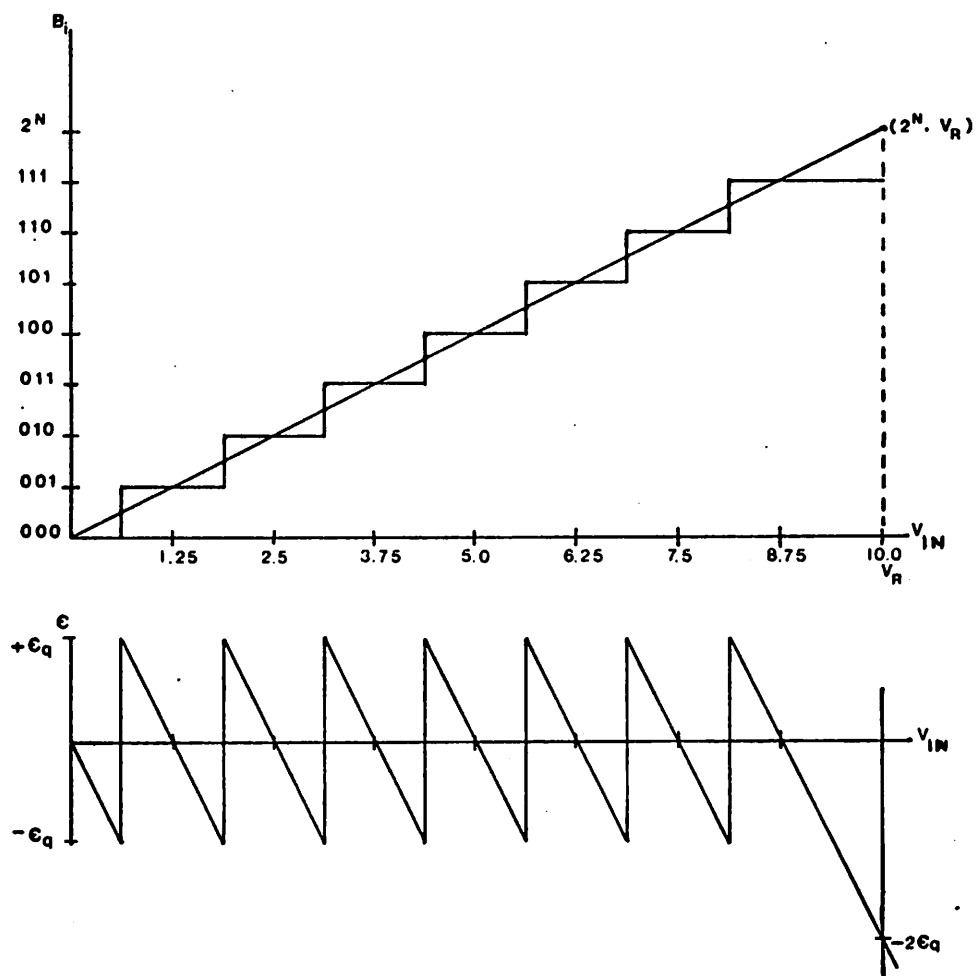


Figure 2.3: The transfer function and quantization error for an ideal ADC.

voltage and the number of bits which then determined the resolution, the accuracy and the maximum quantization error. Except for its endpoints, the transfer function was a regular staircase and each transition occurred at precisely designated points. As might be expected for a non ideal ADC the staircase is not regular and the transition points are not evenly spaced. Although the absolute transition voltage error referenced to  $V_R$  may remain unchanged the same error referenced to a fraction of 1 LSB usually becomes worse as efforts are made to increase the number of bits of resolution. One feature of the ideal converter transfer curve is its linearity. This is identified in Figure 2.3 from the fact that a straight line beginning at the origin will uniformly intersect the midpoint of each vertical transition as well as the boundary point  $(2^N, V_R)$ . This line is called the gain line. Furthermore, within the linear input range, a straight line will intersect all positive-going maxima on the quantization error curve and similarly for all negative-going minima also as illustrated in Figure 2.3. The set of  $2^N - 1$  ideal transition midpoints may be listed:

$$(\text{transition midpoint})_i = B_{T_i} = \left[ \left( B_i + \frac{1}{2} \text{LSB} \right), \frac{V_R}{2^N} \left( i + \frac{1}{2} \right) \right]$$

for  $i = 0, 1, \dots, (2^N - 2)$ . The ideal transfer function has gain which is the slope of the gain line as computed from the coordinates of its endpoints:

$$\text{ideal gain} = \frac{2^N}{V_R} .$$

The transfer function for a real N-bit ADC is shown in Figure 2.4(a). The measured transition voltages  $V_{T_0}$ ,  $V_{T_1}$  and  $V_{T_{(2^N-2)}}$  are recorded. The difference between the real and ideal transition voltages for  $B_{T_0}$  is defined as the offset error and may be computed:  $\text{offset} = V_{T_0} - \frac{1}{2} \frac{V_R}{2^N}$ . The real

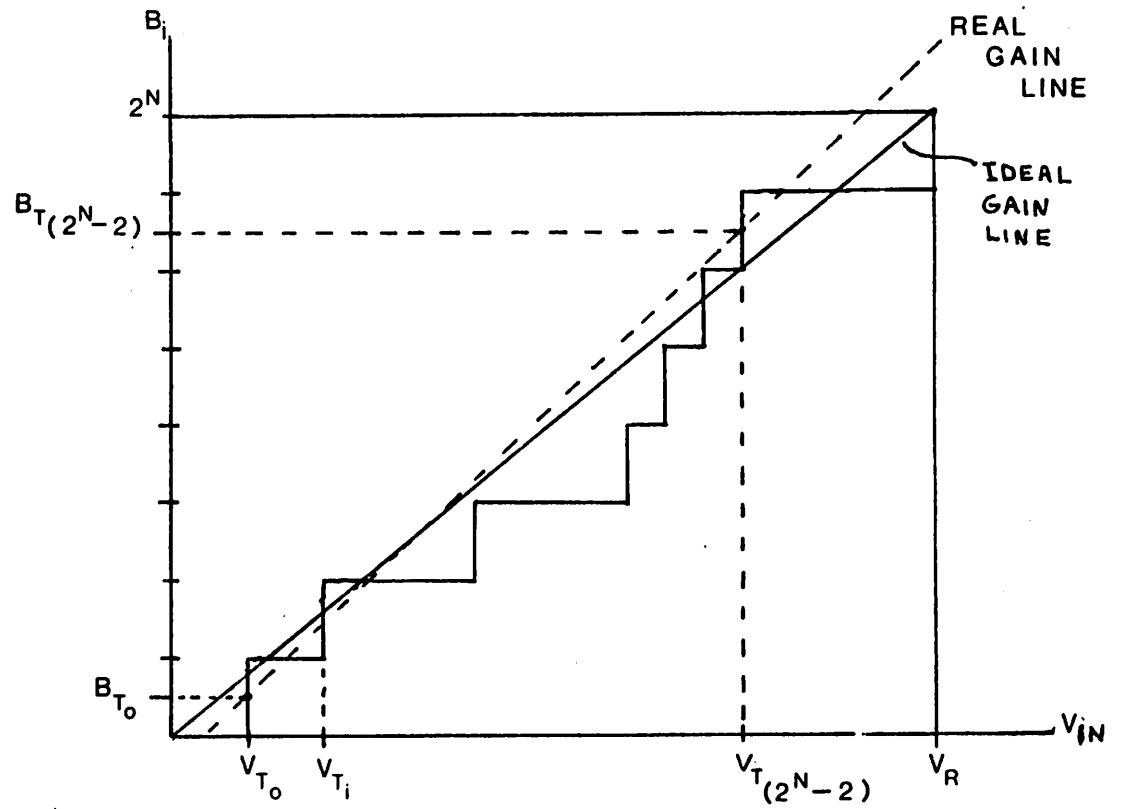


Figure 2.4(a): A real ADC transfer function having both gain and offset errors and nonlinearity.

gain is the slope of the real gain line that intersects the two points

$$\left( B_{T(2^N-2)}, V_{T(2^N-2)} \right) \text{ and } \left( B_{T_0}, V_{T_0} \right):$$

$$\text{real gain} = \frac{2^N - 2}{V_{T(2^N-2)} - V_{T_0}} .$$

A gain error exists which may now be expressed as a percent:

$$\% \text{ gain error} = \frac{\frac{2^N - 2}{V_{T(2^N-2)} - V_{T_0}} - \frac{2^N}{V_R}}{\frac{2^N}{V_R}} \times 100\%$$

If both gain and offset errors are adjusted to zero, as is possible with most ADCs, then the transition midpoints  $B_{T(2^N-2)}$  and  $B_{T_0}$  coincide with their ideal values and the real and ideal gain lines overlap. This is illustrated in Figures 2.4(b) and (c). Now the description of the real transfer function has been reduced to computing transition voltage errors relative to the ideal values. Since the curve which connects the transition midpoints is nonlinear, the parameter which describes it is the worst case deviation from the gain line or the nonlinearity. The nonlinearity may be defined as the worst case deviation of the transition voltage from its ideal value when gain and offset errors have been adjusted to zero. As shown in Figure 2.5 the nonlinearity may be referenced to the analog input axis and expressed as a percent of  $V_R$  or referenced to the digital axis as a fraction of the LSB:

$$\% \text{ nonlinearity} = \frac{\Delta V}{V_R} \times 100 \%$$

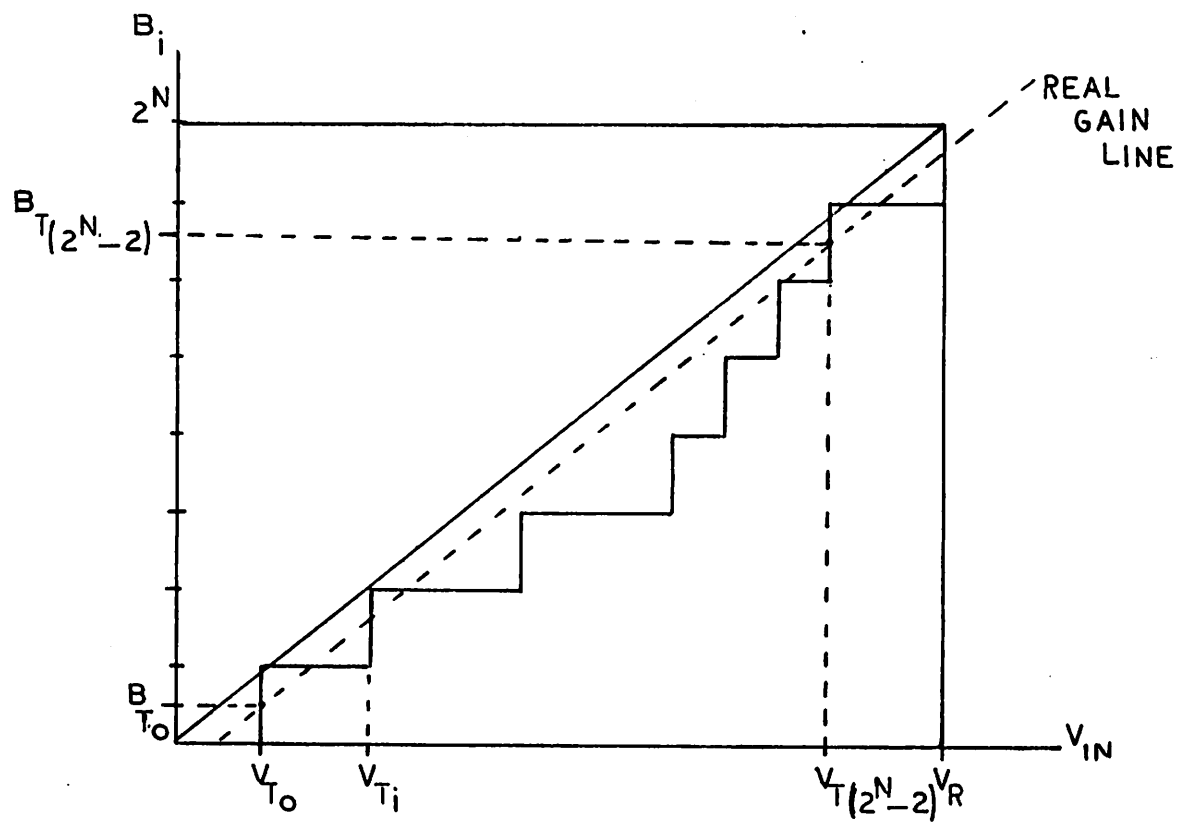


Figure 2.4(b): Gain error adjusted to zero.

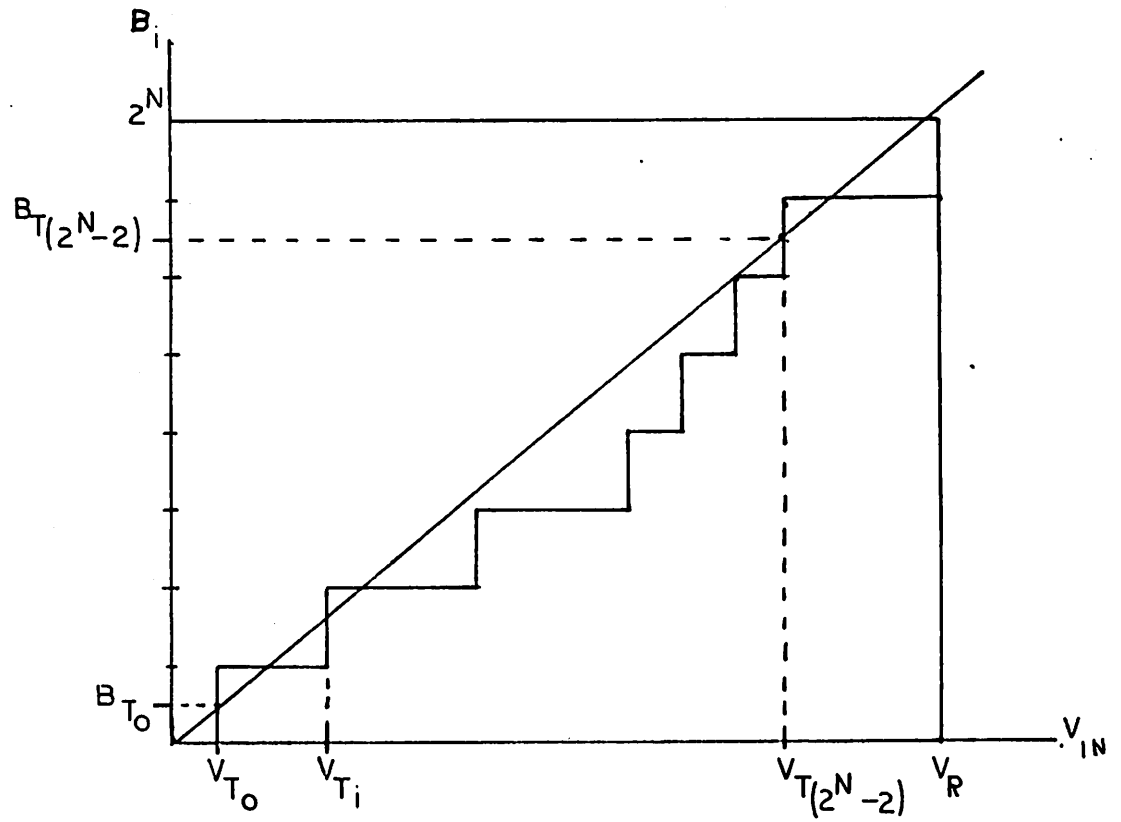


Figure 2.4(c): The ADC transfer function after offset and gain errors have both been adjusted to zero.

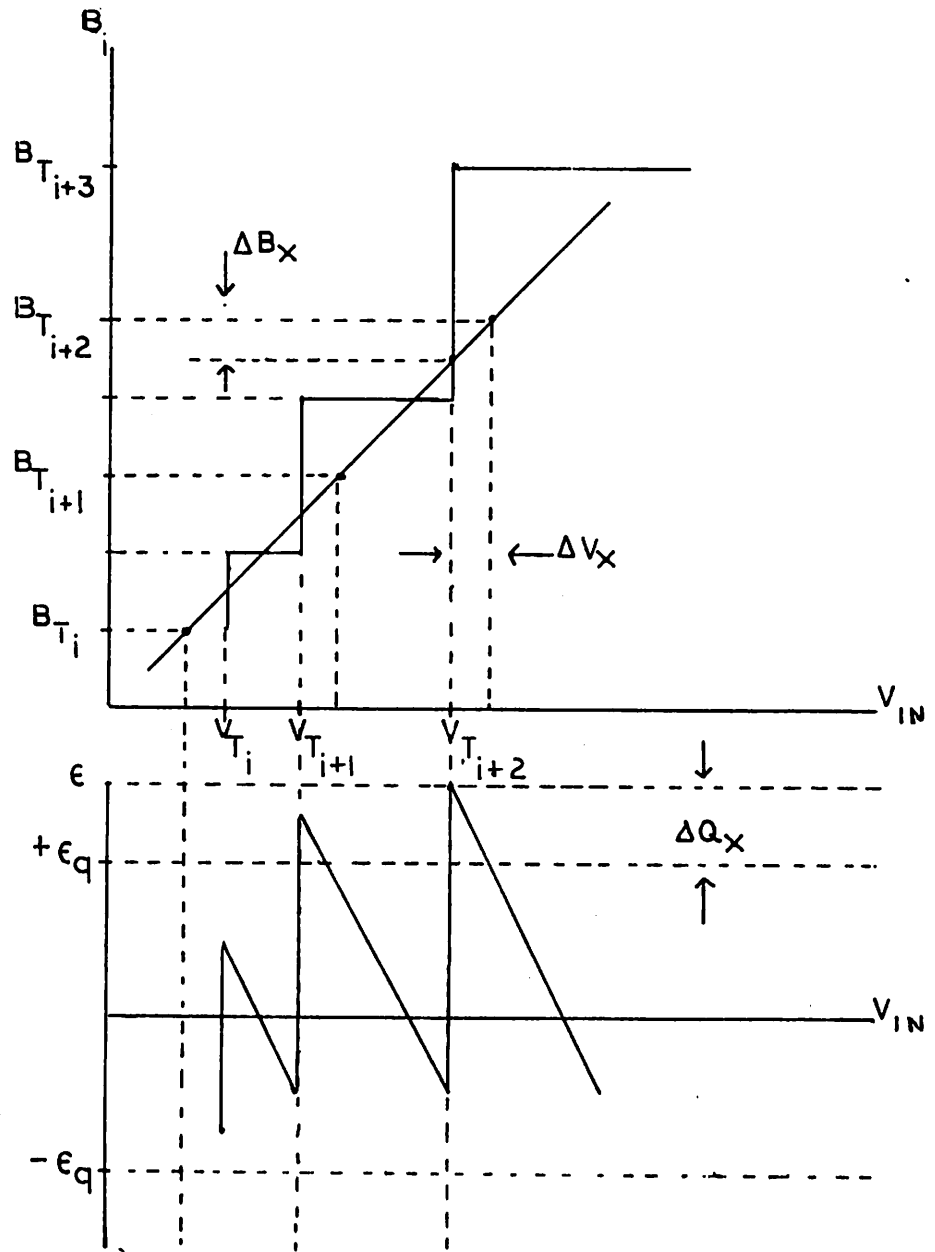


Figure 2.5: Transfer function and quantization error plots illustrating graphical determination of nonlinearity.

or (nonlinearity in bits) =  $\Delta B_x$  LSB.

The corresponding quantization error plotted in Figure 2.5 indicates the nonlinearity directly as shown:

$$(\text{nonlinearity in bits}) = \Delta Q_x \frac{2^N}{V_R} \text{ LSB.}$$

The accuracy of an ADC is defined as the worst case deviation between  $V_{IN}$  and the digital output expressed as a binary fraction of  $V_R$ :

$$\text{accuracy} = \frac{B_x}{2^N} V_R - V_{IN}$$

That is, an ideal converter has an accuracy of  $\pm .5$  LSB due to quantization error; but an encoder with a  $\pm .6$  LSB nonlinearity has an accuracy of only  $\pm 1.1$  LSB. The accuracy of any ADC may be expressed as the sum of quantization, nonlinearity, offset and gain errors.

Although the staircase transfer function for the real converter was shown to be monotonic, that is, uniformly increasing with  $V_{IN}$ , this might not be the case for some converters. This defect is called nonmonotonicity. Another defect, called missing codes, would exist if fewer than  $2^N$  output states were present.

### 2.2.3 Conversion Dynamics

Since time is required for A/D conversion the dynamic performance of the converter will be degraded from its static characteristics especially for high frequency input voltages. One dynamic parameter, the conversion time  $T_c$  is the total time needed to perform one A/D conversion, and  $\frac{1}{T_c}$  is the conversion rate or sampling rate. Two desirable features are a sampling accuracy of  $\pm 1$  LSB for any single sample, and a high sampling rate for a



large input frequency range. The difficulty of achieving both of these will be investigated in the following example.

Let the input signal be a full-scale sine wave of amplitude  $\frac{V_R}{2}$  and frequency  $f$  such that  $V_{IN}(t) = \frac{V_R}{2} \sin 2\pi t$ . If an  $N$ -bit ADC is to convert this signal within 1 LSB, then the maximum rate of change of the input is limited by the unit of quantization and the conversion rate:

$$f = \frac{1}{2^N \pi T_c} .$$

This assumes that the input voltage remains directly connected to the ADC during the entire conversion, hence it must remain stable during that period. From this equation a 10-bit converter with a 100  $\mu$ s conversion period has a maximum input frequency of 3 Hz. However, the input frequency range of the converter may be extended by sampling the signal for a very short period of time and holding that value during the entire conversion. This function is commonly performed by a sample-and-hold circuit (S/H).

In conventional techniques a S/H is interfaced between the signal input and the converter input. Since this is usually realized with 1 or more operational amplifiers the cost is generally high. A simplified S/H circuit is shown in Figure 2.6. The two voltage followers act as unity gain buffers with very low output impedance. The equivalent time constant of the circuit is reduced by the low output resistance of the first follower; therefore, the only significant resistance is that of the closed switch. When the switch is opened the charge remains on the capacitor and its voltage is buffered into the converter. The time required for the S/H circuit to acquire the input voltage is called the acquisition time  $T_{aq}$  which is usually less than the conversion time because of the small time constant

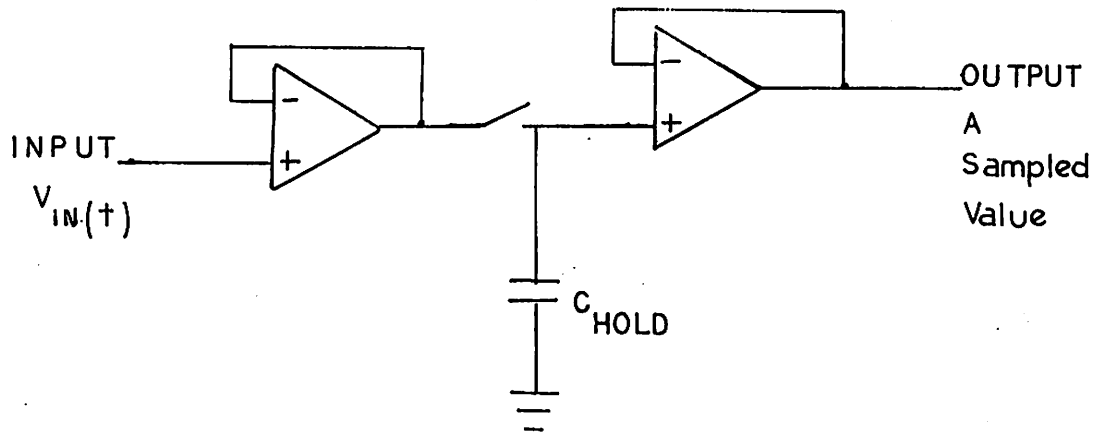


Figure 2.6: A simplified sample-and-hold circuit.

involved. At the end of the acquisition time the output is tracking the input voltage within the specified error range. The time required to open the switch is defined as the aperture time  $T_a$  and random variations in this time are manifested as an uncertainty in the sampled value for rapidly changing input signals. The composite ADC system shown in block diagram form in Figure 2.7 includes the S/H, the generalized DAC, and the comparator. In the timing diagram the acquisition time and the aperture time are contained in the total conversion time even though the operation of the converter does not begin until after  $T_{aq} + T_a$ .

For static inputs the aperture time causes no measurement uncertainty, and the acquisition time is not a limitation because the required sampling accuracy can be achieved if sufficient capacitor charging time is allowed. Therefore the only requirement is that  $T_{aq} \geq M\tau$  where  $M$  is the number of  $C_{HOLD} \times R_{SWITCH}$  time constants  $\tau$  for the desired precision.

For dynamic measurements the maximum input frequency which may be sampled depends upon the criteria used to specify the sampled signal. For example let this criterion be a  $\pm 1$  LSB accuracy in the sampled value (in an idealistic case). In addition it will be assumed that the conversion time  $T_c$  only results in a time delayed output hence this will be ignored with respect to sampling accuracy. For this hypothetical case the maximum input signal frequency is determined by the acquisition time requirements. Figure 2.8 illustrates the equivalent S/H circuit during this time. A sine wave of frequency  $f$  and amplitude  $\frac{V_R}{2}$  has a maximum rate of change equal to  $V_R \pi f$ . For convenience this will now be approximated by a ramp having the same slope:

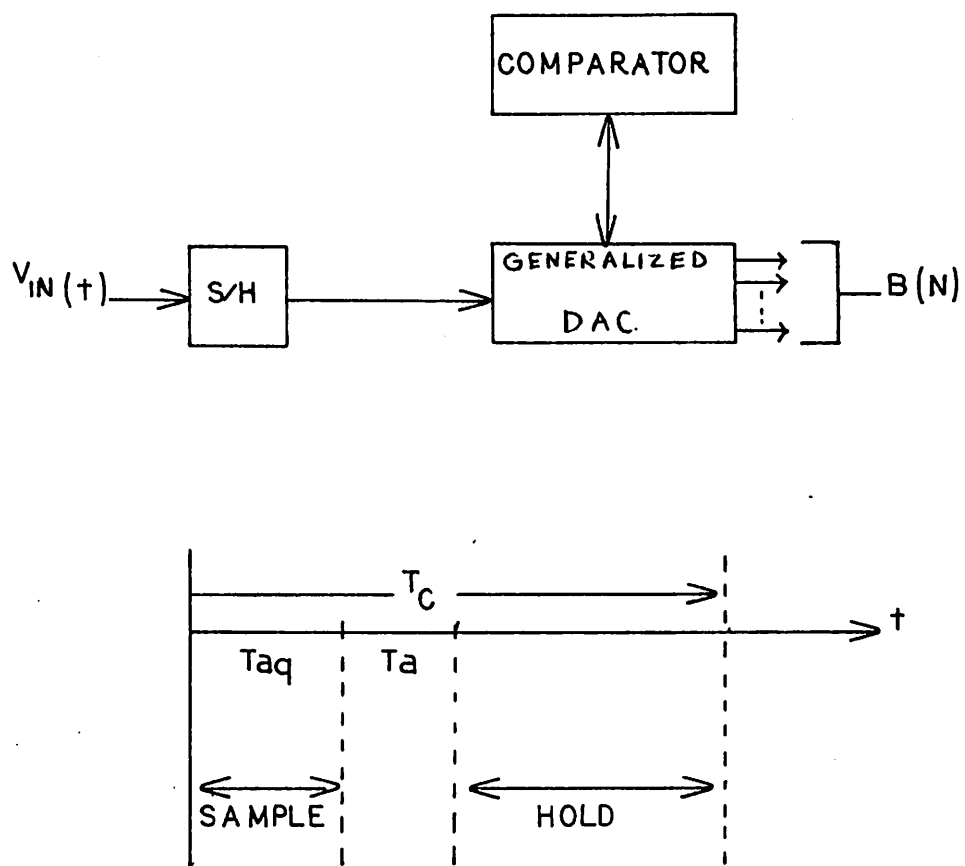


Figure 2.7: A block diagram and system timing diagram of an ADC which includes a sample-and-hold (S/H) circuit.

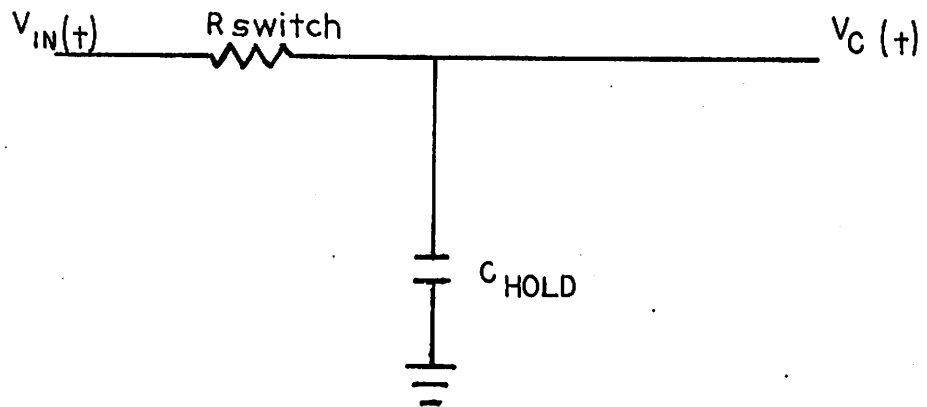


Figure 2.8: A simplified sample-and-hold circuit illustrating the important time constants.

$$V_{IN}(t) = (V_R \pi f)t.$$

Solving this circuit for capacitor voltage  $v_c(t)$  and expressing the error as a fraction of an LSB:

$$\epsilon_{LSB} = 2^N \pi f \tau (1 - e^{-\frac{t}{\tau}}).$$

From this equation the error converges to  $2^N \pi f \tau$  for  $t = \frac{1}{\pi f} \gg \tau$ .

Evaluating this expression for 1 LSB error:

$$\epsilon_{LSB} = 1 = 2^N \pi f \tau$$

from which the maximum input frequency is

$$f_{\max} = \frac{1}{2^N \pi \tau}.$$

Using nominal values of  $C_{\text{HOLD}} = 1000 \text{ pF}$  and  $R_{\text{SWITCH}} = 100 \text{ } \Omega$  for a 10-bit converter, then the maximum frequency which may be sampled to within  $\pm 1$  LSB accuracy by the S/H circuit is 3.1 kHz. The minimum acquisition time for this example is:

$$T_{\text{aq}} = \frac{1}{\pi f_{\max}} = 2^N \tau = 100 \text{ } \mu\text{s}.$$

In practice, however, the acquisition time for this example would be substantially less than this value. This would be true for applications in which signals having much higher frequency components may be sampled if greater dynamic sampling error than  $\pm 1$  LSB can be tolerated (for these frequencies). For example, let the criteria for maximum input frequency be such that the sampled signal is allowed to be attenuated by  $\frac{1}{\sqrt{2}}$  and

phase shifted or time delayed by an arbitrary amount. However the sampled signal is not permitted to be distorted. There are many applications in communications having specifications such as these. It will be shown in Chapter 7 that the S/H circuit (an RC circuit) is actually a low-pass filter for the input signal. Hence from section 5.2.1 the minimum acquisition time required to meet these criteria is

$$T_{aq} = (N+1)\tau \ln 2$$

for which  $f_{max} = \frac{1}{2\pi\tau}$ .

In contrast with the previous example,

$$f_{max} = 1.6 \text{ mhz and}$$

$$T_{aq} = 0.8 \mu\text{s for this case.}$$

It should be pointed out that according to the Nyquist principle a sampling system having a conversion rate of 10 kHz can only recover signals of less than 5 kHz. However there are some data acquisition methods which time-multiplex several 10 kHz converters to produce a sampling system capable of recovering higher frequency signals. An adaptation of this technique referred to as a pipeline converter may be used to perform parallel A/D conversion [2].

### 2.3 Techniques for D/A Conversion

All methods of A/D conversion require a generalized DAC as a component of the ADC. The process of D/A conversion will first be discussed as a necessary prelude to A/D conversion. Although current output converters are available, this discussion will be restricted to voltage output since

only this concept is under study.

### 2.3.1 Summation of Binary Weighted Currents

A simplified form of weighted-current DAC is shown in Figure 2.9. Assuming that the operational amplifier is ideal, the binary ratioed currents are summed at the inverting input. Then

$$V_0 = -\frac{R_0}{R} \left(1 + \frac{1}{2} + \frac{1}{4} + \dots\right) V_R = \left(-\frac{R_0}{R}\right) \frac{V_R}{2^N} B_i; B_i = 0, \dots, 2^N - 1$$

and  $V_0$  becomes a quantized function of a binary number  $B_i$ . The difficulty of implementing this scheme is that the required range of resistor values becomes unmanageable. For example, a 10-bit converter would require that one resistor be of value  $(1024 \pm .05\%) R$ , and this precision is difficult to achieve with most fabrication processes due to the unfavorable aspect ratio of large resistors. In addition the large resistances and associated capacitances would result in slow speeds for some of the switches. A more practical circuit for achieving weighted currents is shown in Figure 2.10 [3]. In this schematic of a 4-bit D/A converter there are 4 current sources which comprise a QUAD. The temperature compensating diode  $D_T$  stabilizes the currents over a given temperature range. In order to maintain the same voltage drop across all base-emitter junctions the current density in each junction must be the same. This is achieved by having the number of emitters proportional to the total emitter current. The binary digit values are input through diodes. A '1' input will reverse bias the diode and the transistor will conduct; however, an '0' will divert all of the resistor current to ground. Although driving current into ground may be somewhat wasteful of power it is necessary in order to avoid thermal gradients when high precision is required. If the QUAD were extended for



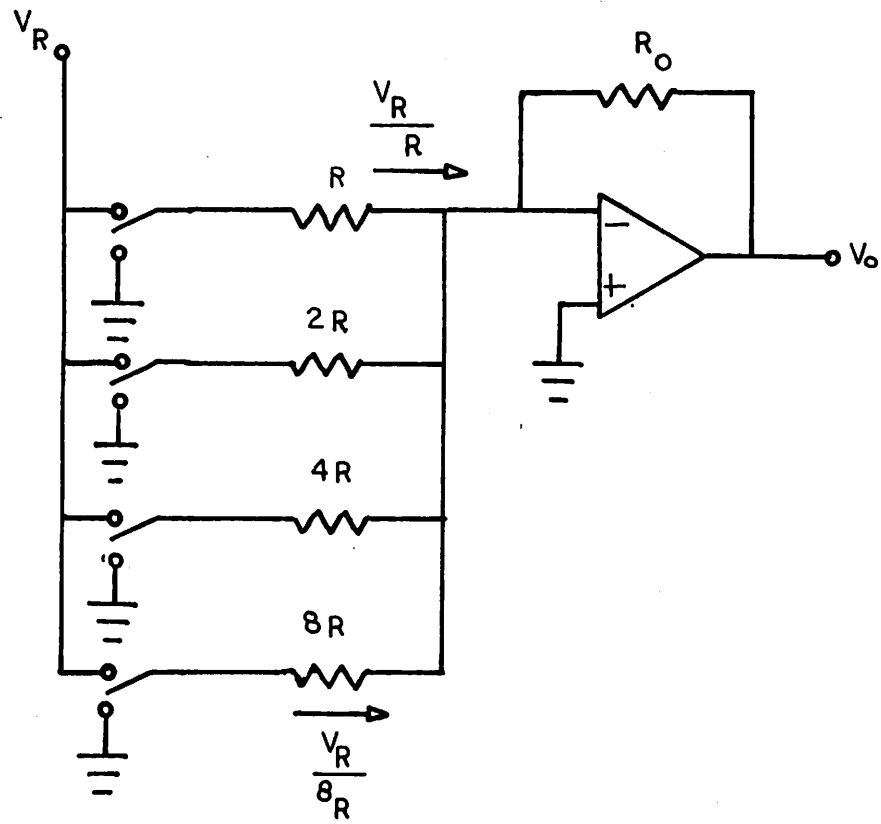


Figure 2.9: A simplified weighted-current DAC.

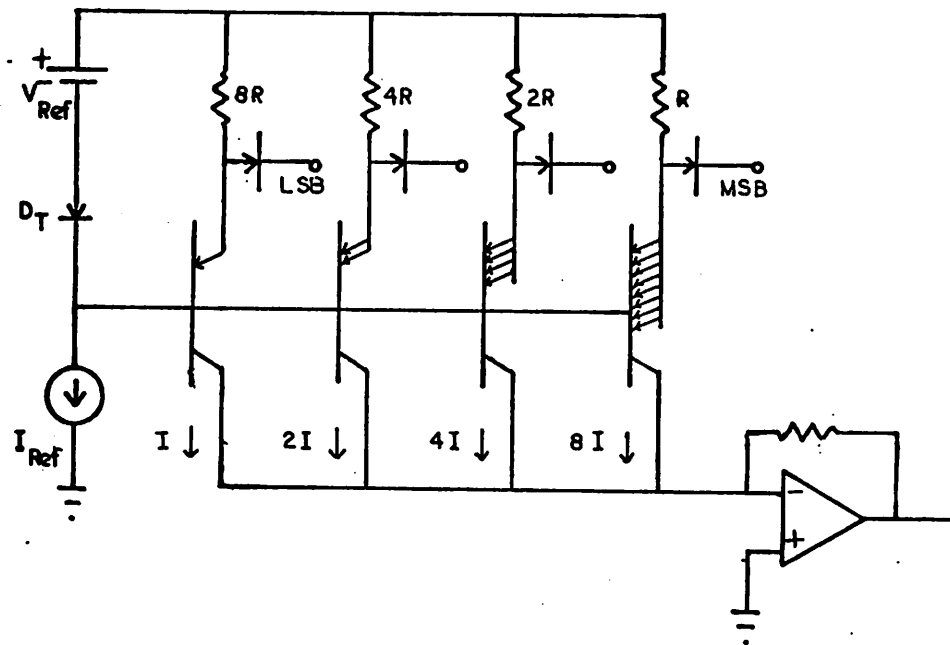


Figure 2.10: A circuit technique for achieving weighted currents.

a 10-bit converter, there would be one transistor requiring 1024 emitters and it would be difficult to retain matching precision for such a device. The actual circuit method groups the current sources in QUADS and interconnects these with current dividers in order to achieve the proper binary weighting. This is done in Figure 2.11. The QUADS are identical thereby avoiding awkward aspect ratios on both transistors and resistors. However, the resistive dividers now contribute to the total error, hence precision must be maintained in these elements too.

A second approach is the use of an R - 2R resistor ladder network to generate binary weighted currents. This is illustrated in Figure 2.12. Binary currents are established by the current division property of the ladder. This technique has the advantage of requiring matching of only two resistor sizes: R and 2R.

### 2.3.2 Binary Attenuation of Equal Currents

Another technique that also employs an R - 2R resistor ladder shown in Figure 2.13 performs binary attenuation of equal-valued currents. If  $I_A = I_B = I_C = I_D = I$  then the output current of the ladder is given by

$$I_{\text{out}} = \frac{I_A}{24} + \frac{I_B}{12} + \frac{I_C}{6} + \frac{I_D}{3} = \frac{I}{24} (1+2+4+8) \text{ and the quantized output voltage is:}$$

$$V_{\text{out}} = \left( -\frac{I}{24} R_{\text{out}} \right) B_i.$$

This method has the advantage of requiring identical current sources and resistors R and 2R which may be more easily matched. On the other hand there are more resistors required.

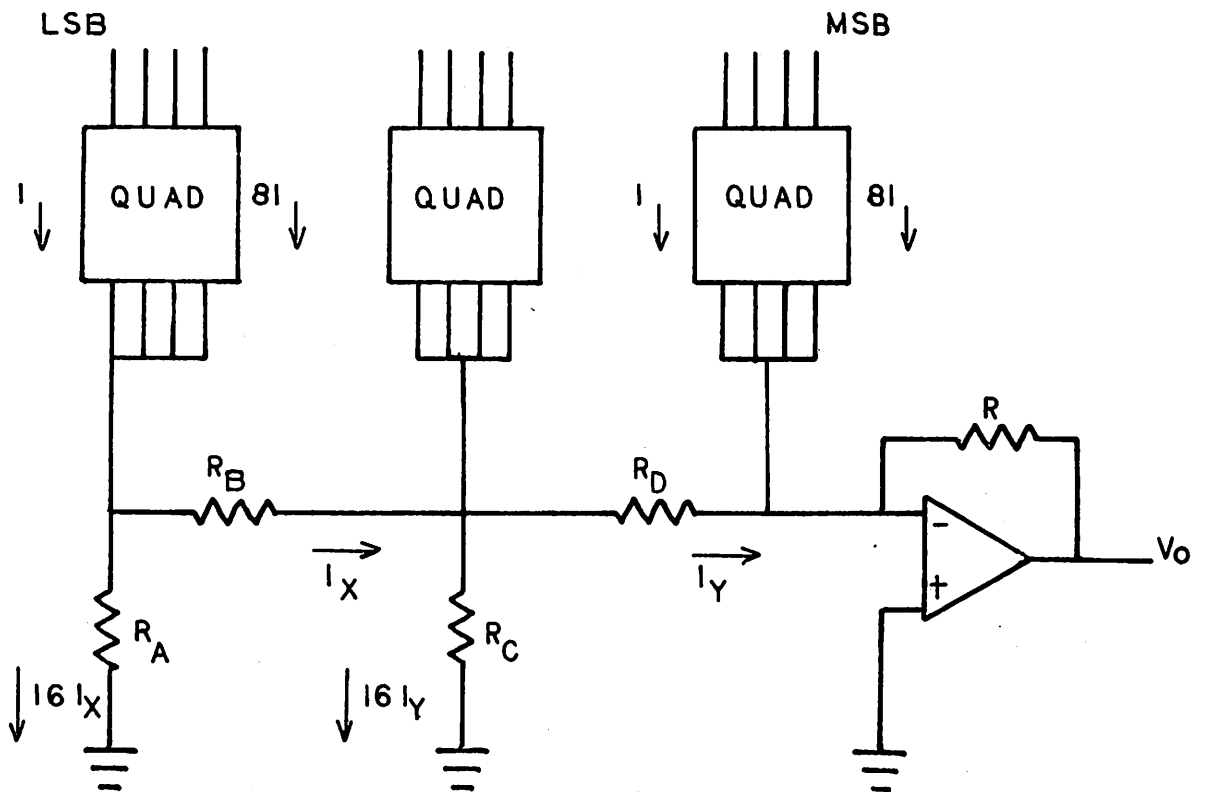


Figure 2.11: Quad current sources with two resistive current dividers.

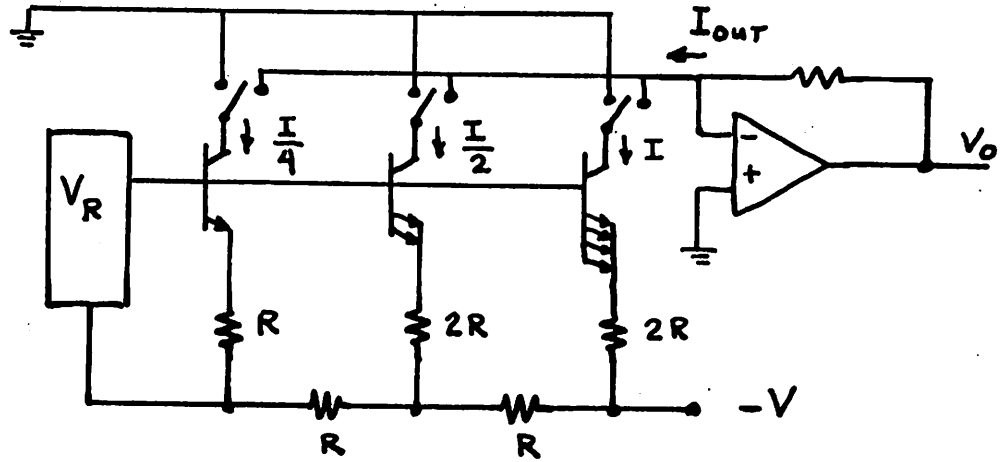


Figure 2.12: A DAC using summation of binary weighted currents generated by an R-2R resistor ladder network.



### 2.3.3 Charge Redistribution

Two capacitors with different initial voltages may be connected together resulting in a final voltage that is dependent upon the total charge. Figure 2.14 illustrates a simplified 2-capacitor circuit. For equal capacitors

$$V_0 = \frac{V_A + V_B}{2}$$

since the initial voltage  $V_A$  equals zero or  $V_R$  and  $V_B$  is initially discharged by S3, then  $V_B$  must be a binary fraction of  $V_R$ :

$$V_B = \frac{V_R}{2^N} B_i = V_0$$

After  $N$  redistributions the output voltage is quantized.

### 2.3.4 Integration Types

A DAC may be conceptually formed by integrating the charge on a capacitor. If a constant current source of value  $I$  charges the capacitor for a discrete number of clock periods,  $T_{CLK} B_i$ , then the final voltage is a quantized function of time:

$$V_0 = \frac{I}{C} T_{CLK} B_i$$

This concept is illustrated in Figure 2.15 in which the counter is initially cleared and counts up to some desired number  $B_i$ . The term  $\frac{I T_{CLK}}{C}$  must remain constant over the range of interest. This places severe tolerance limitations on the three variables involved. For this reason the structure illustrated in Figure 2.15 is not actually used to realize a DAC. However, integration methods which are fundamentally the same as that just discussed are often utilized for ADC circuits. Two particular forms of integration

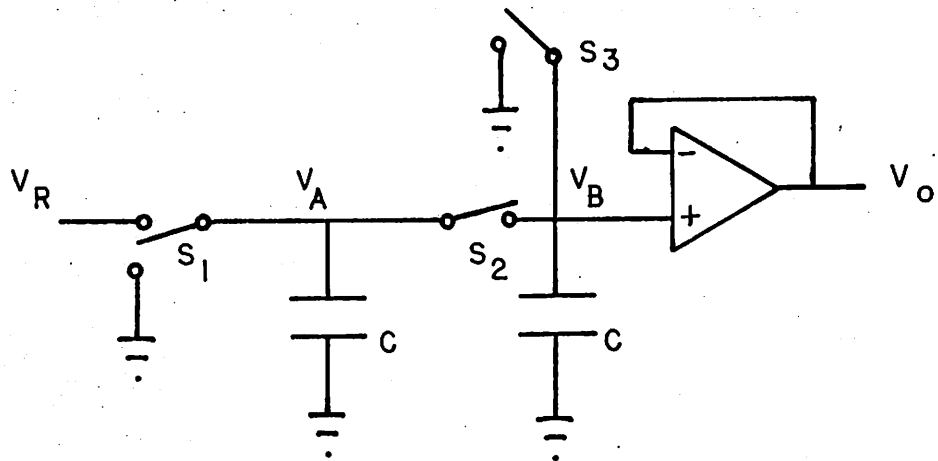


Figure 2.14: A 2-capacitor charge-redistribution DAC.



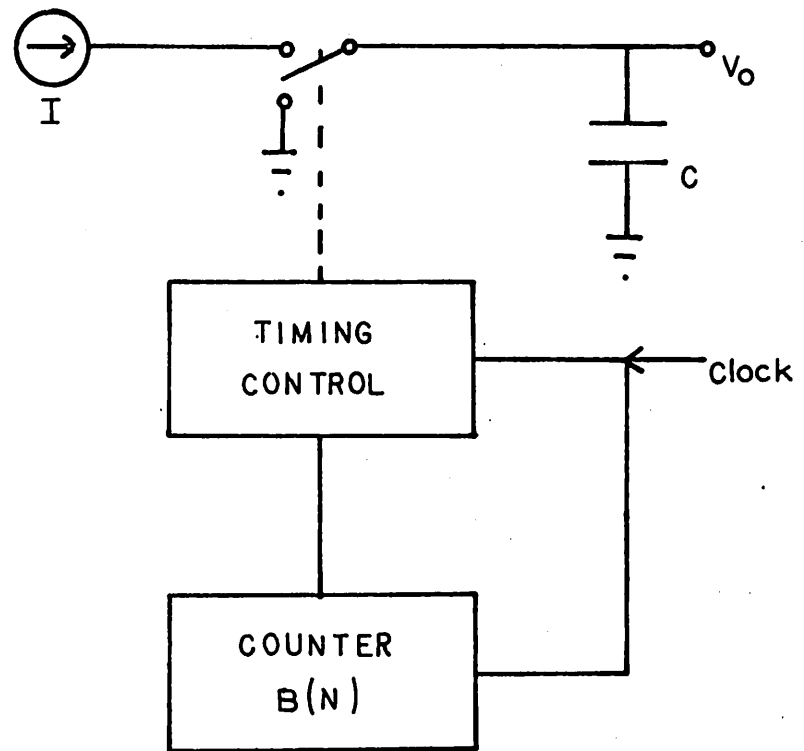


Figure 2.15: An Integration type of DAC.

type conversion circuits are single slope and dual slope converters which are discussed later.

## 2.4 Techniques for A/D Conversion

### 2.4.1 Serial Methods

The simplest method of A/D conversion is a serial method. This technique performs a linear sequential process of conversion such that  $2^N$  comparisons are usually required for a full scale N-bit conversion. A conceptual serial ADC may require a few operational amplifiers, a capacitor and logic circuits as the primary components. A particular advantage is that none of these must be precision components. Although this method is characterized by circuit simplicity, its features include low cost and high accuracy. On the other hand it is very slow. For example, a 12-bit converter with a 5 MHz clock would require 820  $\mu$ s to convert a full-scale input.

Serial ADCs are usually integration type conversion circuits. A particular form of serial ADC using integration methods is illustrated in Figure 2.16 [4]. This circuit may be classified as a single slope type because the slope  $\frac{dv}{dt}$  has only one value of interest I/C. During the regular conversion cycle the capacitor is charged to  $V_{IN}$  and then discharged to the comparator threshold  $V_{TH}$  in time  $T_{IN}$  by the current source I. Then

$$T_{IN} = C(V_{IN} - V_{TH}) = T_{CLK} B_{IN}$$

where  $B_{IN}$  is the binary number of clock periods required to discharge C from  $V_{IN}$  to  $V_{TH}$ . During one phase of operation, the capacitor is charged to  $V_R$  and

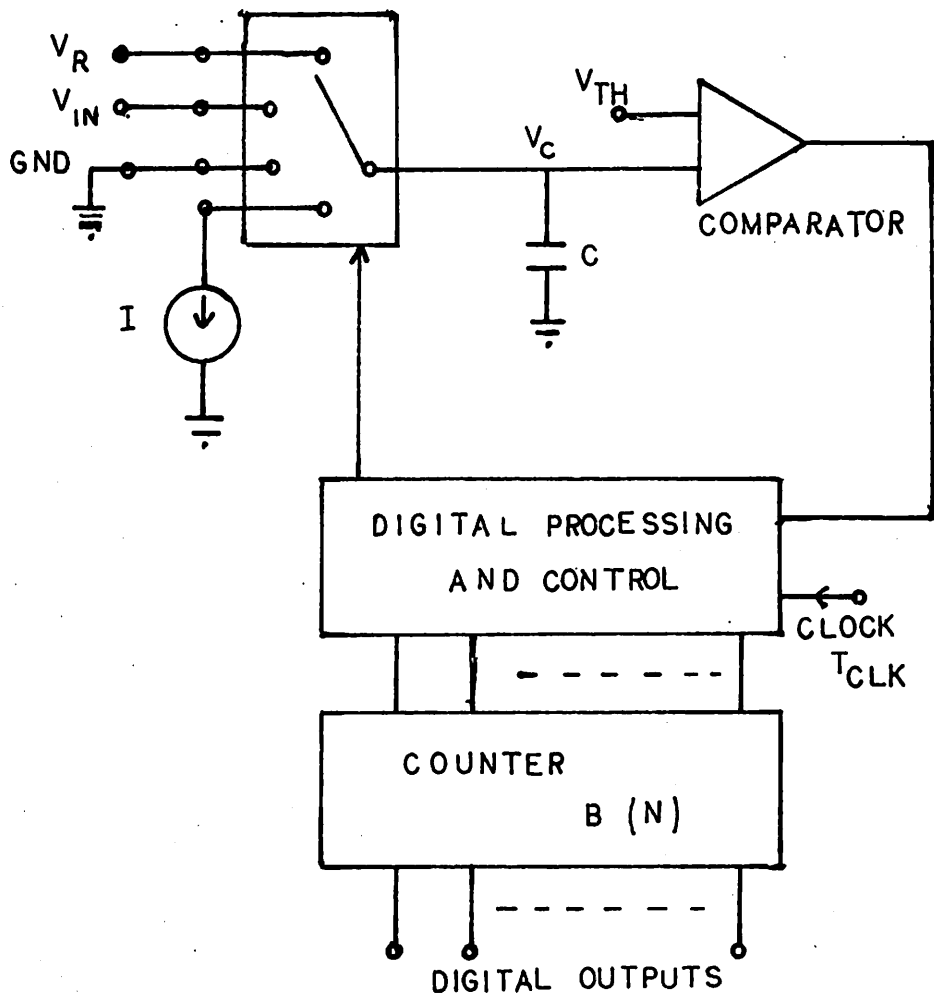


Figure 2.16: A single-slope type of ADC.

$$T_R = C(V_R - V_{TH}) = T_{CLK} B_R.$$

Since the comparator threshold  $V_{TH}$  is negative, ground may also be calibrated:

$$T_G = C(0 - V_{TH}) = T_{CLK} B_G.$$

The digital output is then expressed as a binary fraction of the reference:

$$B_i = \frac{V_{IN}}{V_R} \frac{B_{IN} - B_G}{B_R - B_G}$$

and is independent of the value of  $T_{CLK}$  and  $C$ . The only precision requirements are that  $I$  and  $C$  be constant over the voltage range of interest during the time of each conversion. Disadvantages of this technique are the very slow conversion rate and the need for a division. Other characteristics of the single slope method are low cost, high accuracy and the inclusion of an intrinsic S/H function. Single slope ADCs have not been widely produced commercially since previous designs have required precision components or exact timing control.

Another serial method of A/D conversion which is more widely available than the single slope types is the dual slope ADC. It is shown in Figure 2.17. During a fixed time interval the switch contacts  $V_{IN}$  and the capacitor, which was initially discharged, now charges with a current  $\frac{I_{IN}}{R}$ . When  $V_O$  equals  $V_{TH}$ , the comparator threshold, the counter begins counting up to a fixed number:

$$B_X = \frac{RC (V_X - V_{TH})}{T_{CLK} V_{IN}}.$$

At this point the switch is connected to  $-V_R$  and the counter is reset and begins counting again. The value of the count when  $V_O$  equals  $V_{TH}$  is:

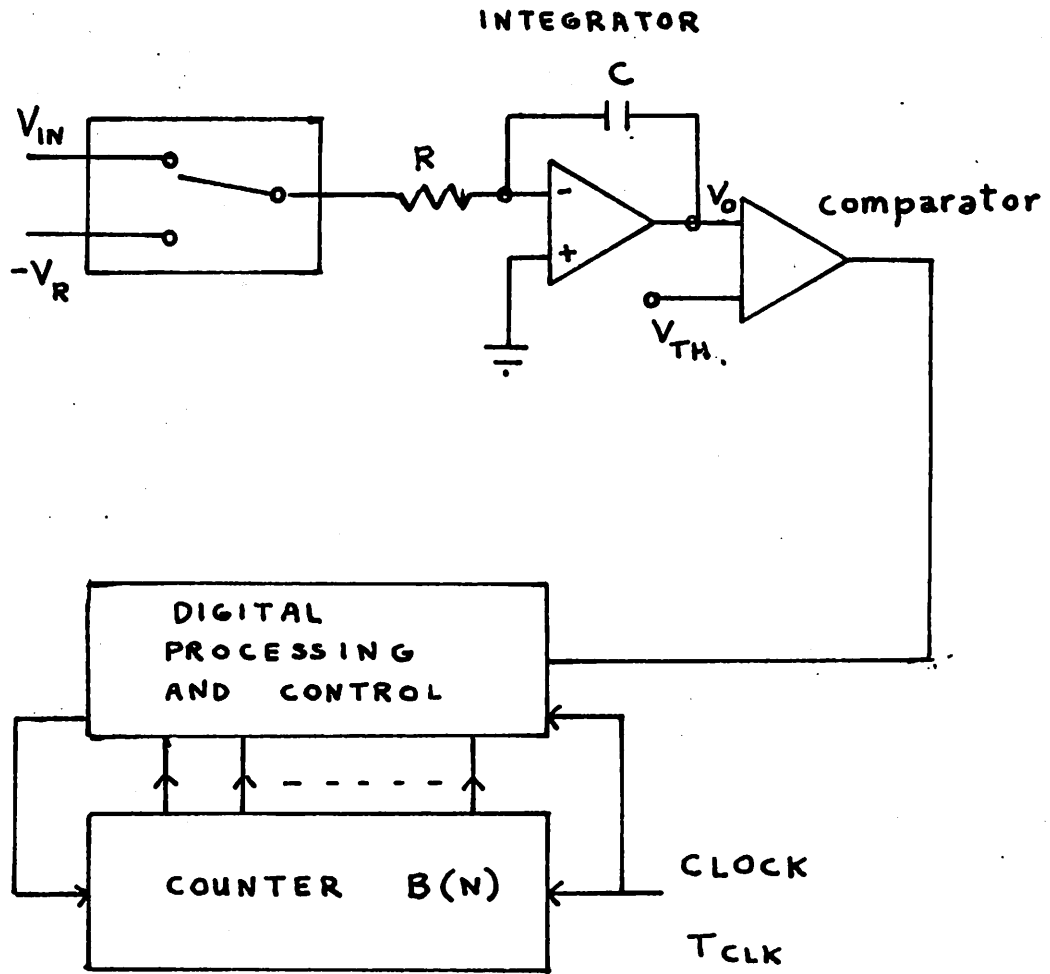


Figure 2.17: A dual slope ADC.

$$B_R = \frac{RC (V_X - V_{TH})}{T_{CLK} V_R} .$$

If  $B_X$  equals  $2^N$ , a fixed count, then the following result is obtained:

$$B_R = 2^N \frac{V_{IN}}{V_R} .$$

The input has been quantized and converted. Furthermore, the accuracy is independent of the value of  $T_{CLK}$ ,  $R$ ,  $C$ , or  $V_{TH}$  and depends only upon the linearity of the integrator and the hysteresis of the comparator, both of which may be controlled. The dual slope converter has similar properties as other serial types: low conversion rates, low cost, and high accuracy. In contrast to the single slope method, the subtractions and divisions are not needed but also there is no S/H function and unfortunately an operational amplifier is required versus a constant current source. Also a negative reference is needed. The dual slope technique has gained wide acceptance in applications having only slowly varying inputs. Commercially available converters of this type are usually multi-chip realizations because the operational amplifier requires a bipolar technology while the digital circuitry is best realized in MOS technology.

#### 2.4.2 Successive Approximation Methods

Successive approximation ADCs are probably more available commercially than any other types and usually have high resolution and fast conversion rates of 10 kHz to 10 MHz. This high speed is achieved by a quantizing algorithm which converges exponentially rather than linearly as in the previous methods. This is accomplished by successive comparisons of the

input with binary fractions of the reference. The control logic operates upon the D/A converter by testing all  $N$  bits sequentially beginning with the MSB. This is done by assuming the bit value is '1' and then comparing the DAC output to  $V_{IN}$ . If this output is less than  $V_{IN}$  then the bit value is actually '1', otherwise, it is returned back to '0'. In this manner  $2^N$  possible binary numbers could be tested in only  $N$  operations. The usual circuit configuration is illustrated in Figure 2.18. The DAC is usually the weighted current source or resistive ladder type. The digital control circuitry required for this scheme is usually quite simple. The accuracy of the output is dependent upon the comparator and the DAC. However, it is usually the resistor matching difficulty in the latter which creates the nonlinearity.

#### 2.4.3 Parallel Conversion

A parallel ADC requires the simultaneous reference generation and comparison for all  $2^N - 1$  transition voltages [5]. Thus  $2^N - 1$  fractions of  $V_R$  must be formed by the converter and delivered to  $2^N - 1$  separate comparators. This is shown conceptually in Figure 2.19 for a 2-bit parallel ADC. The parallel DAC consists of a string of  $2^N$  resistors. Although this guarantees monotonicity at the DAC the offset variations of the comparators may result in missing codes. Also the linearity may become difficult to maintain if the number of resistors becomes very large. In addition the need for  $2^N - 1$  comparators probably precludes monolithic high resolution parallel ADCs because the chip area required increases exponentially with  $N$ . Therefore this method is generally limited to very low resolutions. Parallel ADCs are very expensive however they offer conversion rates up to 25 MHz. Another feature is that they are inherently asynchronous and therefore do not require clock signals.

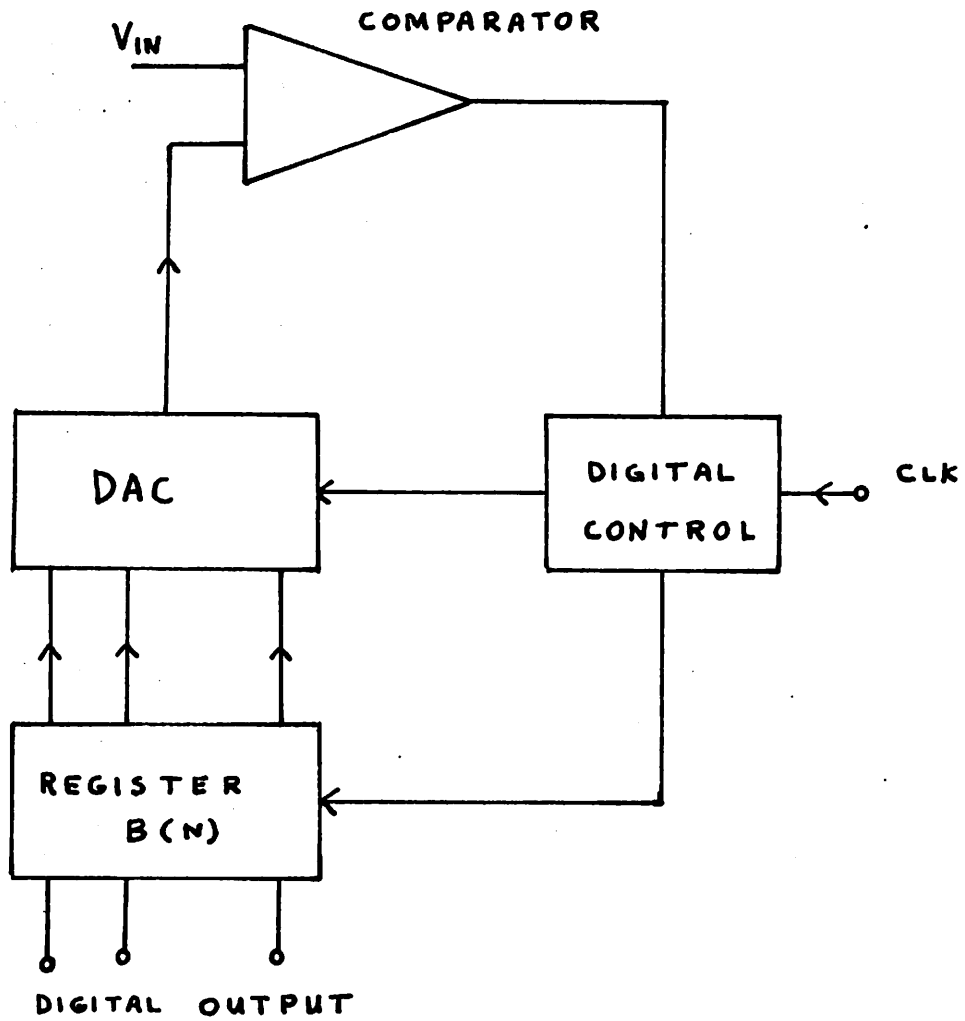


Figure 2.18: A successive approximation ADC.



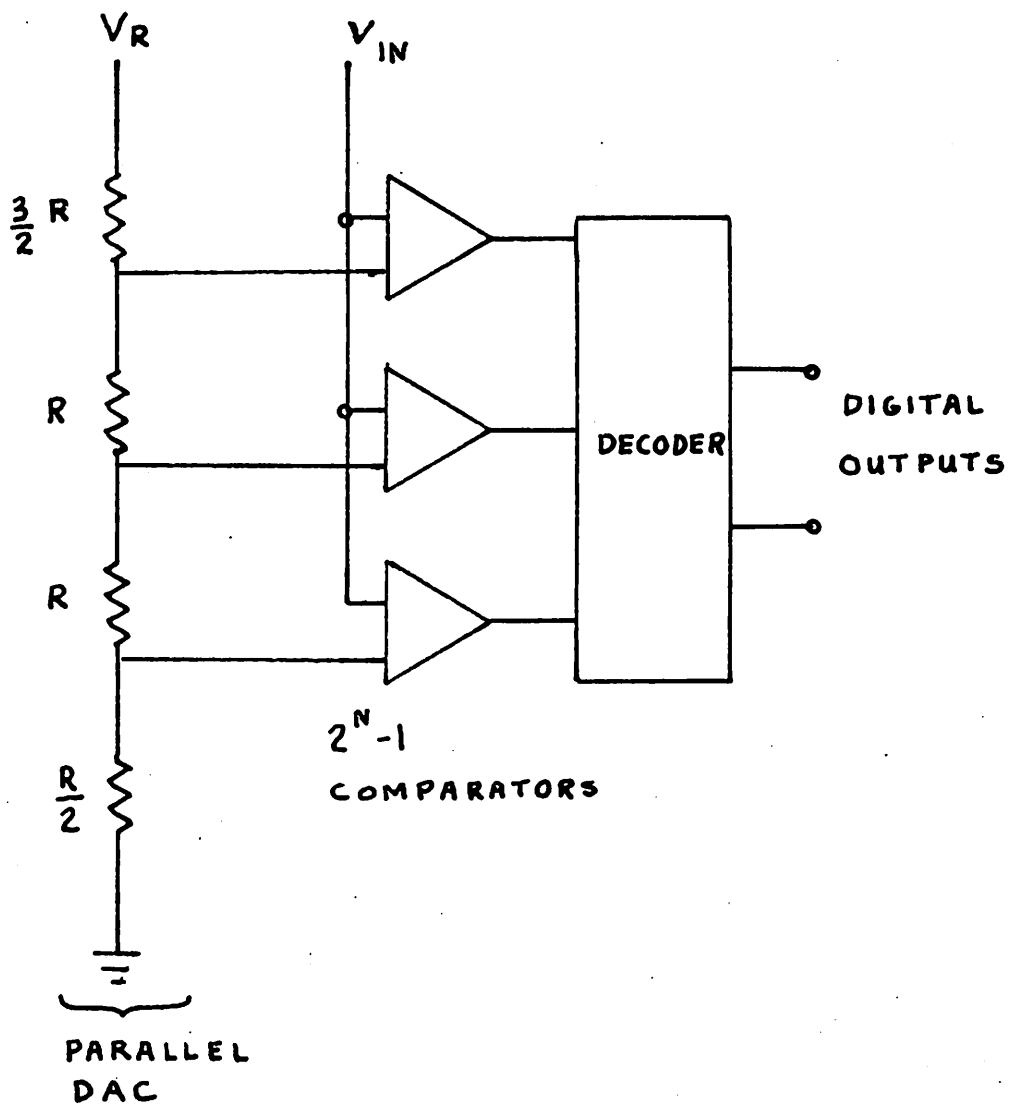


Figure 2.19: A parallel ADC.

## 2.5 Technologies for ADC Components

### 2.5.1 Introduction

Integrated circuit (I.C.) technologies, or fabrication techniques, have had a great impact upon electronic circuits. Linear circuits such as operational amplifiers, comparators, buffers, and current sources which were once made from discrete components are now commonly available as single-chip, bipolar I.C.s. An even greater development of digital I.C.s has occurred due to the low cost, high density advantages of MOS technology. In fact, MOS memories may be constructed from just a few chips while an entire CPU may be purchased as a single monolithic I.C. In contrast to either linear or digital circuits, converters must perform both analog and digital functions. While computational operations with digital circuits can be done with arbitrarily large precisions, such accuracies are not easily achieved in analog circuits, in fact, they are not always required. In order to minimize the error in analog operations, the standard bipolar technology often requires additional complexity such as component trimming and off-chip adjustments or external circuitry. In this section a survey is made of the conventional fabrication methods of ADC components. Methods of realizing comparators, control logic, and supporting functions will also be discussed.

### 2.5.2 ADC Component Technology for Charge Redistribution and Integration Methods

Charge-redistribution (C/R) methods have commonly been utilized in successive approximation ADC techniques, however these circuits have usually contained discrete components such as resistors and capacitors and bipolar op amps. On the other hand it has recently been demonstrated that a single chip, two capacitor C/R ADC may be realized with MOS

technology [6].

Conventional integration methods, used in most serial type ADCs, are usually single slope or dual slope converters. These techniques commonly require op amps and often a discrete capacitor. In contrast a prototype serial ADC using the single slope method has been recently developed using MOS technology and having a potential for single chip realization [7].

### 2.5.3 ADC Component Technology for Precision Resistor Networks

Both the R-2R ladder and the weighted current source methods require precision resistors. Therefore, the following analysis will examine the three different methods of fabricating precision resistors for DAC circuits.

It would be convenient, of course, if diffused resistors could be made accurately since they are compatible with the standard bipolar technology and require no additional processing steps. However, sheet resistance  $\rho_s$  is usually limited to less than 200  $\Omega$ /square, creating difficulty in matching resistors over a wide range of binary values. Therefore, the binary weighted resistor method is not considered practical for high resolutions when diffused resistors are used. On the other hand, the R-2R ladder requires equal resistors of only two values. Therefore this approach is more suitable for diffused resistors. However there are still constraints which require proper design in order to achieve high accuracies. For example, the emitter resistors  $R_E$  of the equal current sources shown in Figure 2.12 must have an IR drop greater than  $2^N$  times the  $V_{BE}$  mismatch. Small currents would be desirable for low power dissipation and small thermal gradients, therefore large resistors having high sheet resistance  $\rho_s$  would be required. On the contrary high resistivities in the R-2R ladder result in greater nonlinearities due to higher voltage coefficient of

resistance (VCR). This arises because the resistors are actually formed with reverse-biased pn junctions having depletion region widths which are dependent upon junction voltage. The effective  $\rho_s$  is therefore somewhat dependent upon the IR drop. Since the depletion region width depends upon the doping gradient at the junction, this effect can be reduced by using low resistivity diffusions in the ladder. A tradeoff arises and the resistors must be fabricated in a manner which minimizes error due to the 2 effects just mentioned. At least one DAC is commercially available which utilizes a diffused R-2R network as part of a 10-bit monolithic bipolar I.C. [8].

In contrast to diffused resistor techniques, impurity ions may be accelerated by an electric field and driven into a silicon surface by a special process called ion implantation. The depth of penetration is generally shallow but the resultant impurity profile and concentration may be controlled. This technique is capable of providing very high resistivity values with dimensional tolerances determined primarily by the photomasking process. Present data indicates that better matching can be achieved than for diffused resistors but at the expense of one implantation. An experimental 10-bit DAC has been built by a commercial manufacturer [9]. It is an R-2R ladder type and is a complete monolithic I.C. utilizing ion implantation and bipolar technology.

A more complex method of resistor fabrication utilizes materials containing nichrome or tantalum or cermet (Cr-SiO) which are deposited as thin-films onto an insulating substrate. Patterns may then be etched to form thin-film resistors. Subsequent oxidation or annealing steps will increase or reduce  $\rho_s$  respectively. By carefully controlling these processes, sheet resistances from 10 to 2500  $\Omega$ /square can be achieved. In contrast

to other types of resistors, thin-films may be trimmed by laser techniques thus enhancing the matching capabilities by one or two orders of magnitude [10]. It would appear that unlimited matching precision could be achieved in this manner; however, there are limitations. For example, the laser beam itself may oxidize or anneal parts of the resistor during the trim causing an uncertainty. Furthermore, materials evaporated by the beam may land on top of previously trimmed resistors thereby changing their values. In addition thin-film resistors have non-zero long-term drift and almost always require passivation layers. In spite of these difficulties, added cost, and complexity, thin-film resistors have generally been preferred over diffused or implanted resistors because they may be trimmed to 13 or 14-bit matching accuracies while the other 2 resistor types cannot be easily trimmed and have been generally limited to 10-bit precisions. Therefore, converters requiring high accuracies (exceeding 10 bits) almost always use discrete or trimmed thin-film or thick-film resistor networks. These may be either R-2R ladders or weighted current source types. The switches and other components of these DACs have been realized in bipolar technology. Some attempts to use CMOS technology and thin-film networks have needed external bipolar comparators and references [11].

#### 2.5.4 Comparator

Most high-speed precision comparators available today are realized in bipolar technology. Although MOS and CMOS comparators do exist they have generally larger offset voltages, slower switching speed and different power supply requirements. Furthermore since most DACs already utilize bipolar circuits there is generally better compatibility with comparators of the same technology. For these reasons bipolar comparators prevail

although MOS comparators have been realized as special purpose sub-units of MOS LSI circuits.

#### 2.5.5 Digital Logic and Control

The most favorable technology for digital control and switching logic is MOS due to its high functional density, low cost, and process simplicity. CMOS technology has advantages of greater logic swing and lower d.c. power but at higher cost and complexity and lower functional density than MOS. Both of these methods require less power and area than bipolar logic and are usually much more desirable for digital circuits except in cases where speed is the ultimate concern.

#### 2.5.6 Supporting Functions

Supporting functions such as amplification, buffering and voltage referencing have been realized almost exclusively in bipolar technology. Precision unity gain and high gain amplifiers are still difficult to achieve in MOS technology. In addition, stable, accurate voltage references have been developed only in bipolar technology.

#### 2.5.7 Summary

In conclusion, ADC components which require high precision circuit elements such as resistor networks, voltage references, op amps and unity gain buffers strongly favor bipolar and thin-film technologies. However, complex digital circuits are more advantageously realized as MOS chips.

## CHAPTER III

All-MOS, Successive Approximation, Weighted Capacitor, Analog-to-Digital Conversion Technique--RADCAP3.1 Introduction

There are many circuit methods which perform A/D conversion as discussed in Chapter II. One particular method which is considered in more detail in this chapter is based on the successive approximation algorithm. This scheme has advantages of high speed operation for the amount of circuit complexity required. In section 3.3 MOS technology is considered as one fabrication method which may be used to realize analog and digital circuits. A particular technique for realizing a successive approximation ADC in MOS technology is by charge-redistribution on binary weighted capacitors, as discussed in section 3.4.

3.2 Successive Approximation A/D Conversion3.2.1 A Comparison of Successive Approximation Method and Other Techniques

The conversion of an analog input voltage  $V_{IN}$  into  $N$  binary bits of resolution requires that the input be compared with a subset of  $(2^N - 1)$  different fractions of the reference  $V_R$ . These fractions actually correspond to transition voltages as discussed in Chapter II. Hence it must be possible in the course of the A/D conversion that any fraction  $(B_i - \frac{1}{2}) \frac{V_R}{2^N}$  (where  $1 \leq B_i \leq 2^N - 1$ ) be generated for comparison with  $V_{IN}$ . These fractions may be generated in the form of a continuous ramp (quantized by a clock or counter) in the case of a serial converter. In this scheme all values of  $(B_i - \frac{1}{2}) \frac{V_R}{2^N}$  may be tested, but only in a sequential or linear manner. That is, after  $V_{IN}$  is tested against  $(B_i - \frac{1}{2}) \frac{V_R}{2^N}$  the next comparison is with  $(B_{i+1} - \frac{1}{2}) \frac{V_R}{2^N}$ . The advantages of the serial method include greater circuit simplicity and fewer precision components

than other methods. However this scheme is very slow since an average of  $\frac{2^N}{2}$  tests may be required for conversion. In contrast, the parallel ADC compares  $V_{IN}$  with  $(2^N-1)$  fractions of  $V_R$  that are simultaneously generated and transmitted to  $(2^N-1)$  different comparators. The parallel converter is extremely fast since only 1 time interval is required for comparison; however, the need for such a large number of components results in a much greater circuit complexity than for other methods. The successive approximation ADC represents a compromise in both speed and complexity compared with serial and parallel methods. The converter must still be capable of generating  $(2^N-1)$  possible fractions of  $V_R$  but only  $N$  tests are required for  $N$ -bit resolution as contrasted with the serial method. This is because each subsequent test except for the first is actually a conditional test which depends upon the outcome of the last comparison. This method is characterized by high conversion rates (though not as high as for parallel converters) and high resolutions with intermediate circuit complexity. Although a particular application may be better suited for either serial or parallel conversion methods, there are many applications for successive approximation techniques and therefore this method has gained wide acceptance.

### 3.2.2 The Successive Approximation Algorithm

Attention is now focused upon the actual sequence of operations in the successive approximation algorithm. This analysis is aided with the flow chart characterizing the algorithm which is shown in Figure 3.1. The operation begins by assuming that the MSB is '1' since initially the "next most significant untested bit" is the MSB. The fraction  $\frac{A}{B} V_R = (\frac{2^N}{2} - \frac{1}{2})$  is thereby generated and compared with  $V_{IN}$ . If  $V_{IN}$  is greater than this



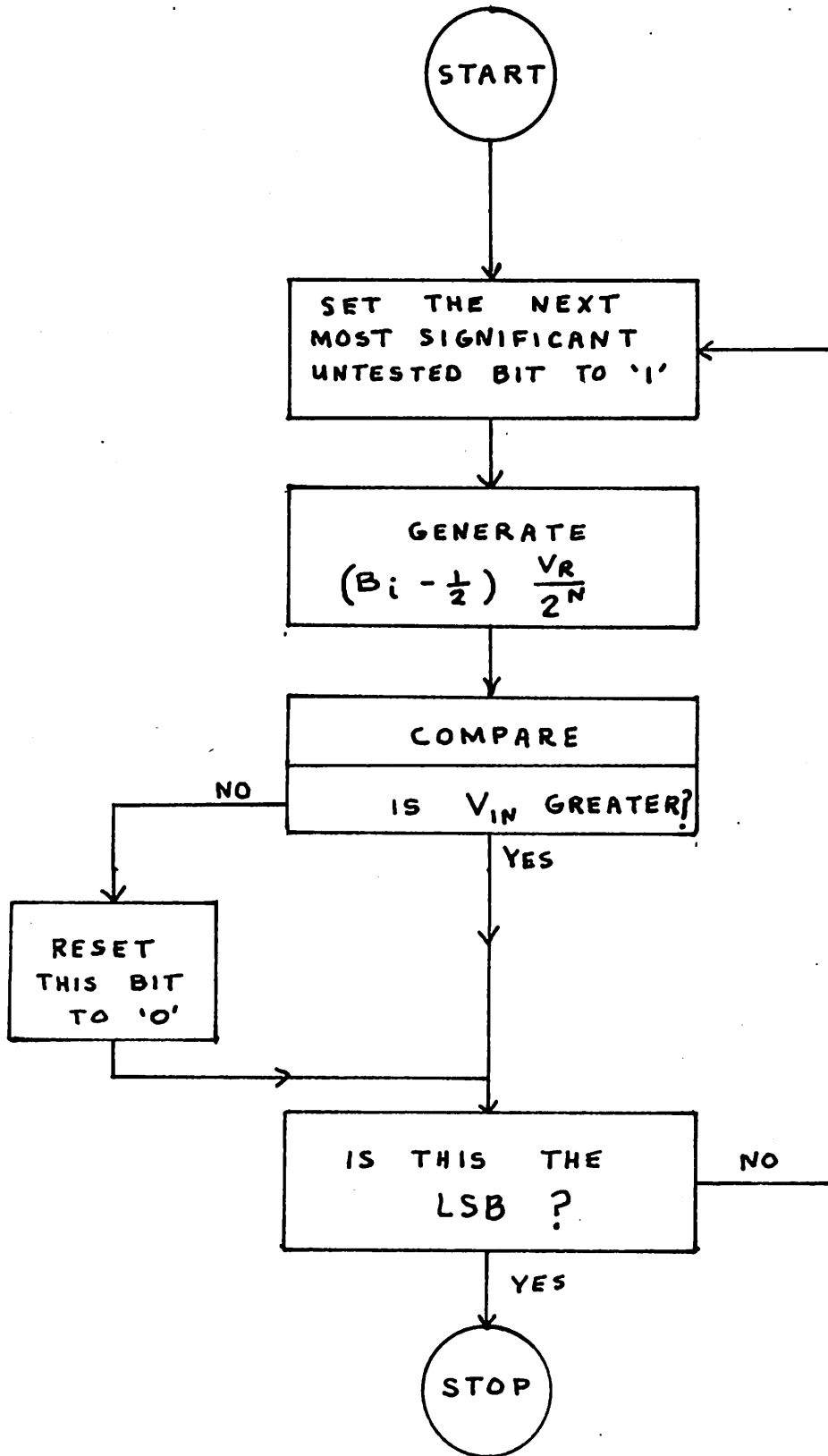


Figure 3.1: The successive approximation algorithm.

voltage then the value of this bit must be '1' as assumed. However if  $V_{IN}$  were less than the generated voltage then the value of the bit under test is reset to '0'. If the bit just determined were the LSB then the conversion is terminated, otherwise the next bit is tested just as before. The successive approximation converter therefore conducts a binary search for the best approximate digital value of  $V_{IN}$  by successively guessing  $V_{IN}$  and then determining whether or not the guess was correct. In this manner the conversion succeeds in  $N$  successive comparisons.

### 3.2.3 Precision Component Requirements for DACs

The most common circuit realization of the successive approximation ADC was previously shown in Figure 2.18. The register, digital control, and comparator may be realized as monolithic I.C.s without serious difficulty; however, the DAC must contain precision components of some kind. These have commonly been precision resistor networks or current sources. For high resolution converters the realization of these precision elements requires design considerations and often additional fabrication complexity. The need for precise components arises from the necessity of generating precise binary fractions of  $V_R$  which are accurate to within 1 part in  $2^N$ . As the desired resolution increases, the allowed component tolerance diminishes.

## 3.3 Factors Influencing the Choice of Technology for Monolithic Realization of a Successive Approximation ADC

### 3.3.1 Advantages of MOS Realization

It was illustrated in section 2.4 that conventional techniques for high speed A/D conversion require high performance analog circuitry such as op amps, precision elements of some kind, and also digital circuits

for counting sequencing and data storage. This has tended to result in multi-chip circuits consisting of one or more bipolar analog chips, possibly a thin-film resistor network, and also a MOS chip to economically perform the digital functions [12]. In converters utilizing precision resistor networks the high degree of resistor matching required has been incompatible with standard I.C. technology for resolutions greater than 8-10 bits. The high cost of precision networks has prevented the realization of high speed low cost converters. Since cost reduction is an industry objective as well as a goal of this research work, considerations must first be given to choice of technology. Lowest fabrication cost on a die area and gate count basis is achieved with MOS technology, having added advantages of high functional density, low power dissipation, and fabrication process simplicity. However, MOS technology has been primarily applied to digital logic. If the required analog processing for the A/D conversion can be performed in MOS technology, a low cost single-chip MOS realization is possible. In addition a MOS ADC has greater potential for future application in that it is process compatible with a MOS micro-processor circuit and therefore could possibly be placed on the same die [13]. Therefore MOS compatibility is desirable.

### 3.3.2 Realization of Precision Attenuator Networks Compatible with MOS Technology

A precision attenuator is necessary for successive approximation A/D conversion. This component performs quantized attenuation of a reference voltage and provides binary fractions of this reference for comparison. The attenuator performs a quantizing function similar to that of a DAC. In conventional successive approximation methods the attenuator usually

contains a weighted resistor or R-2R network. In either case precisely matched current sources and resistors are required. The fabrication of matched-resistor networks and current sources in MOS technology does not appear to be practical for several reasons. First a weighted resistor method would require larger sheet resistivities than could be reasonably achieved with a standard MOS process. Second, even an R-2R ladder would need special design considerations in order to achieve the level of matching required for resolutions in the 10-bit range. Third, a MOS weighted current source or R-2R ladder network would require a MOS device to be used as a current switch. However, the "ON" resistance of the MOS device is much larger than for bipolar junction transistor switches and this resistance would have to be carefully scaled over such a wide range of values that high resolutions would not be easily obtained.

In contrast to its utilization as a current switch, the MOS device, used as a charge switch, has inherently zero offset voltage and as an amplifier has very high input resistance. In addition, capacitors are easily fabricated in metal gate technology. Therefore, one is led to use capacitors rather than resistors as the precision components, and to use charge rather than current as the working medium. This technique, referred to as charge-redistribution, has been used in some discrete component ADCs for many years [14]. However, these converters have required high-performance operational amplifiers which are difficult to realize in single channel MOS technology. Therefore, the design objective of this research centered upon the development of a precision charge-redistribution MOS attenuator which does not require an op amp.

### 3.3.3 VATCAP--A MOS DAC

In this section the utilization of weighted capacitors to perform precision binary attenuation is discussed. The basic component is the precision attenuator VATCAP. It has been shown in section 2.2.1 that for an ideal ADC

$$V_{IN} = B_i \frac{V_R}{2^N} - V_E \text{ where } B_i = 0, 1, \dots (2^N - 1).$$

The error voltage  $V_E$  represents the quantization uncertainty such that

$$-\frac{1}{2} \frac{V_R}{2^N} \leq V_E \leq \frac{1}{2} \frac{V_R}{2^N},$$

and

$$V_E = \frac{B_i V_R}{2^N} - V_{IN}.$$

This equation illustrates the two basic operations which must be performed. First the fraction  $\frac{B_i V_R}{2^N}$  must be generated by VATCAP, which is actually performing a D/A conversion function. Then this fraction must be compared with  $V_{IN}$ . The equation shows this operation as a subtraction which makes  $V_E$  either positive or negative.

The circuit realization of the first operation, the precision attenuation, can be achieved using charge-redistribution. More specifically the formation of the term  $\frac{B_i V_R}{2^N}$  is desired. One way to accomplish this is with charge-redistribution between two capacitors. Assume that the initial voltages  $V_x$  and  $V_y$  in Figure 3.2 are equal to zero and that now  $V_y$  takes on the value  $V_R$ . The final voltage  $V_x$ , resulting from the charge-redistribution is:

$$V_x = V_R \frac{CB}{CA + CB}$$

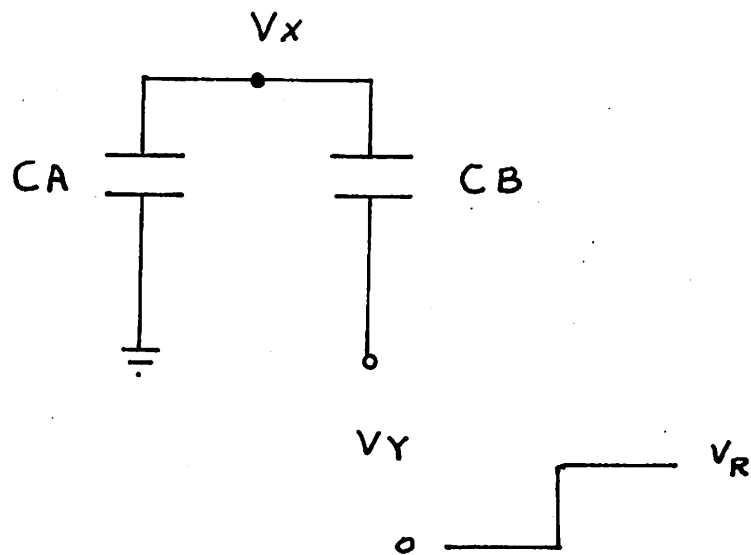


Figure 3.2: A charge-redistribution operation between 2 capacitors.

Furthermore it is desired that  $V_x = B_i \frac{V_R}{2^N}$ , therefore  $\frac{B_i}{2^N} = \frac{CB}{CA + CB}$ . The relationship between CA and CB for any value of  $B_i = 0, 1, \dots (2^N - 1)$  may be determined by arbitrarily setting  $CB = B_i C_1$  where  $C_1$  is referred to as the capacitor of "unity weight" ( $CB = C_1$  if  $B_i = 1$ ). Now CA may be computed,  $CA = (2^N - B_i)C_1$ . Hence both CA and CB are integral multiples of the unity weight capacitor. Significant information can be extracted by expanding the equations for  $C_A$  and  $C_B$ . Since  $B_i$  is an arbitrary binary number from 0 through  $2^N - 1$ ,

$$B_i = D_1 2^M + D_2 2^{M-1} + \dots + D_M 2^1 + D_{M+1} 2^0$$

where  $M < N$ , and D is the binary bit value. Then  $CB = (D_1 2^M + \dots + D_{M+1} 2^0) C_1$  and by expansion of  $2^N$ ,

$$CA = (2^{N-1} + 2^{N-2} + \dots + 2^1 + 2^0 + 2^0 - D_1 2^M - \dots - D_{M+1} 2^0) C_1.$$

From this expression it may be deduced that CA and CB can be configured from a string of binary weighted capacitors plus an additional capacitor of unity weight. Of the members of the string, those which are not used as components of  $C_B$ , must be contained in  $C_A$ . Any reference  $\frac{B_i}{2^N} V_R$  may be generated from a single supply  $V_R$  and an array of binary weighted capacitors having two of unity weight. The error voltage  $V_E$  may now be formed by precharging  $V_x$  to an initial value  $-V_{IN}$  rather than zero. The previous analysis remains valid by superposition. In conclusion the voltage  $V_x$  actually corresponds to the error voltage  $V_E$  which is the desired function.

### 3.4 A/D Conversion Using Charge-Redistribution on Weighted Capacitors--

#### RADCAP

In section 3.3.3 the framework was developed for an MOS precision voltage attenuator, VATCAP, which performed the D/A conversion function. The attenuator was proposed to be an array of binary weighted capacitors with an additional capacitor of unity weight. If a DAC were desired the voltage output of the array after buffering would provide that function. However, an ADC is actually the design objective. In this case it is only necessary to add  $-V_{IN}$  to the top plate of the array initially and then test the sign of  $V_E$  ( $V_E = \frac{B_1 V_R}{2^N} - V_{IN}$ ) after each redistribution. In this section a particular capacitor array structure is examined and the manner in which  $-V_{IN}$  is stored and  $V_E$  subsequently compared with zero is illustrated in detail.

One realization using binary weighted capacitors to perform A/D conversion is illustrated with a conceptual 5-bit version of the converter shown in Figure 3.3 [15]. It consists of a comparator, an array of binary weighted capacitors plus one additional capacitor of weight corresponding to the LSB, and switches which connect the plates to certain voltages. A conversion is accomplished by a sequence of three operations. In the first, the "sample mode" (Figure 3.3), the top plate is connected to ground and the bottom plates to the input voltage. This results in a stored charge on the top plate which is proportional to the input voltage  $V_{IN}$ . In the "hold mode" of Figure 3.4 the top grounding switch is then opened, and the bottom plates are connected to ground. Since the charge on the top plate is conserved, its potential goes to  $-V_{IN}$ . The "redistribution mode," shown in Figure 3.5 begins by testing the value of the MSB. This is done by raising the bottom plate of the largest capacitor to the



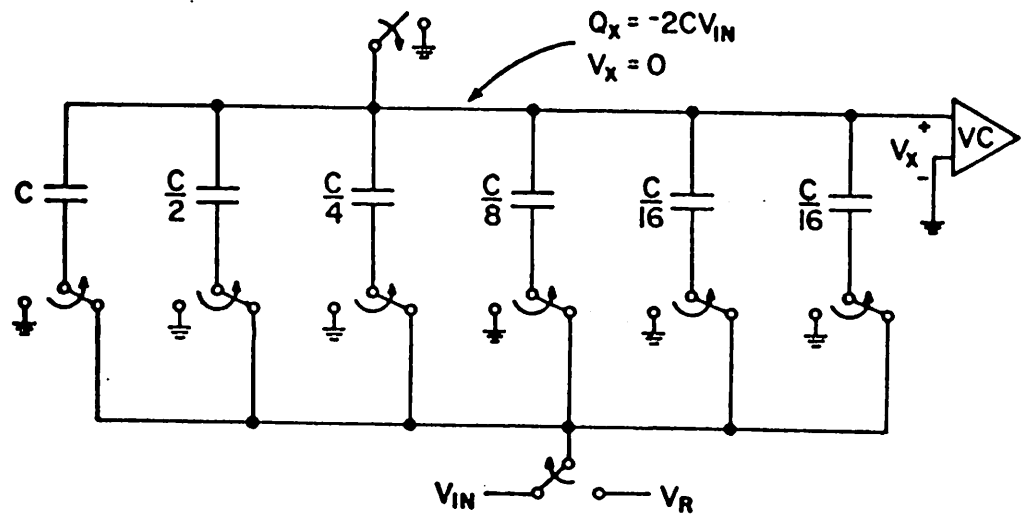


Figure 3.3: The sample mode.

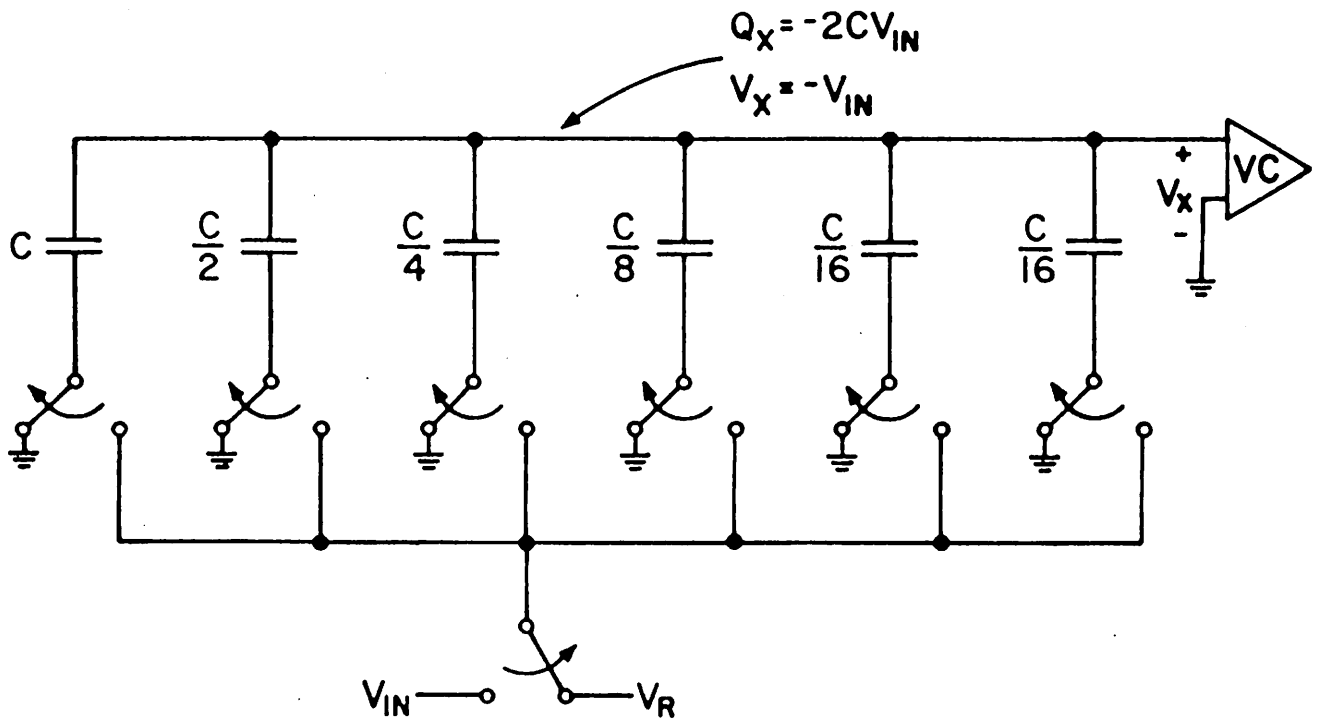


Figure 3.4: Pre-redistribution hold mode.

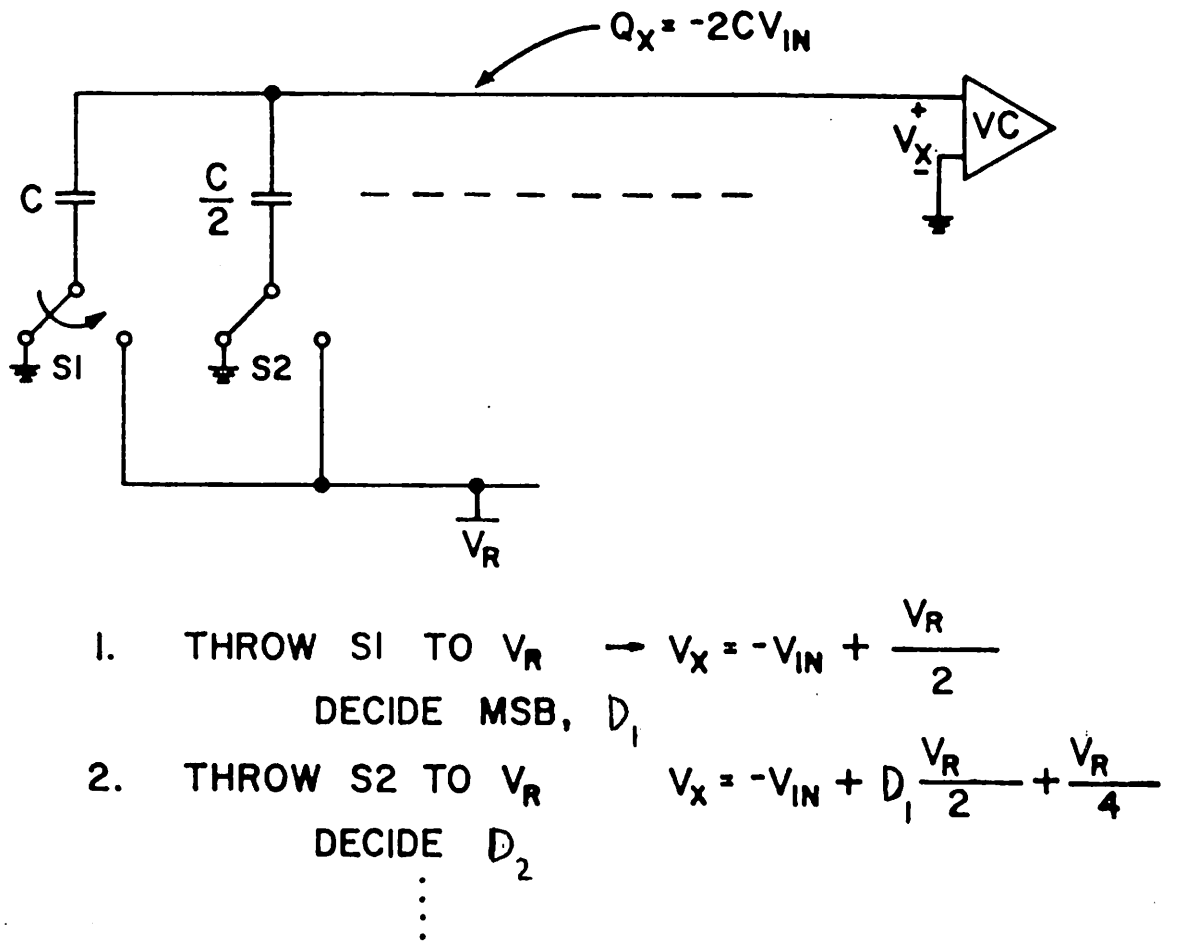


Figure 3.5: Redistribution mode.

reference voltage  $V_R$ . The equivalent circuit is now actually a voltage divider between two equal capacitances. The voltage  $V_x$  which was equal to  $-V_{IN}$  previously is now increased by  $\frac{1}{2}$  the reference as a result of this operation.

$$V_x = -V_{IN} + \frac{V_R}{2}$$

The comparator senses the sign of  $V_x$  and its output is a logic '1' if  $V_x < 0$  and is a '0' if  $V_x > 0$ . This is analogous to the interpretation that

$$\text{if } V_x < 0 \text{ then } V_{IN} > \frac{V_R}{2} \text{ hence the MSB} = 1$$

$$\text{but if } V_x > 0 \text{ then } V_{IN} < \frac{V_R}{2} \text{ therefore the MSB} = 0.$$

The output of the comparator is, therefore, the value of the binary bit being tested. Switch  $S_1$  is returned to ground only if the MSB,  $D_1$  is a zero. In a similar manner, the next MSB is determined by raising the bottom plate of the next largest capacitor to  $V_R$  and checking the polarity of the resulting value of  $V_x$ . In this case however the voltage division property of the array causes  $\frac{V_R}{4}$  to be added to  $V_x$ :

$$V_x = -V_{IN} + D_1 \frac{V_R}{2} + \frac{V_R}{4}$$

Conversion proceeds in this manner until all the bits have been determined. The final value of  $V_x$  is the error voltage  $V_E$ . A final configuration is illustrated in Figure 3.6 for the digital output 01001. Notice that all capacitors corresponding to a '0' bit are completely discharged. The total original charge on the top plates has been redistributed in a binary fashion and now resides only on the capacitors

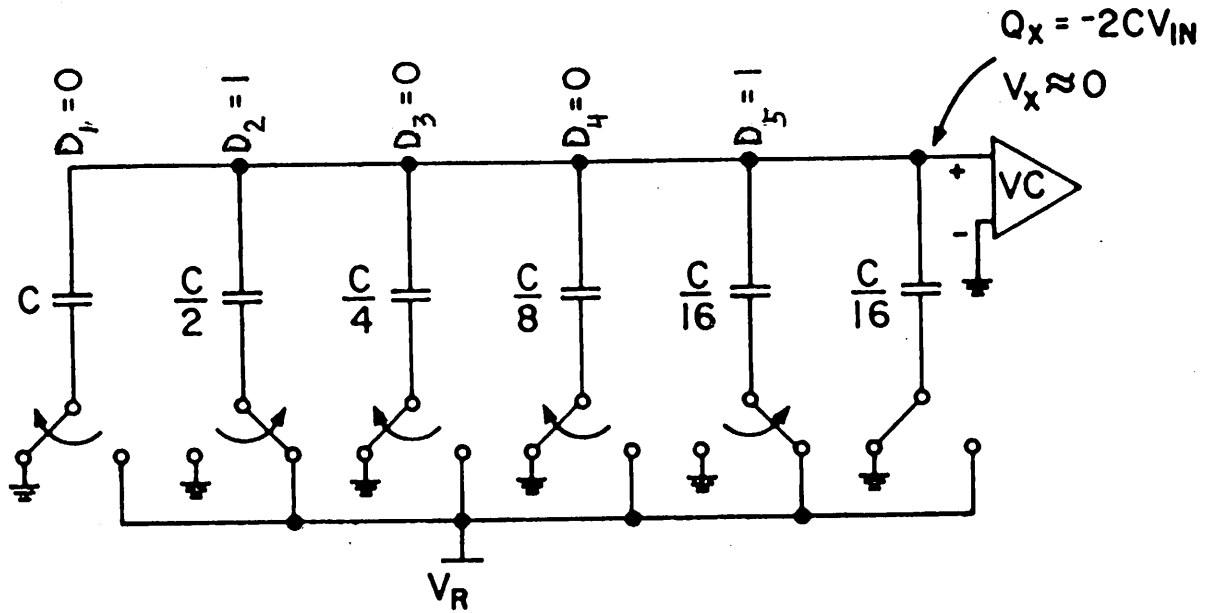


Figure 3.6: The final configuration (example).

corresponding to a '1' bit.  $N$  redistributions are required for a conversion resolution of  $N$  bits. In contrast to earlier charge-redistribution techniques the capacitance of the lower plate switch does not affect the accuracy of the conversion [16]. This fact is evident since the switch capacitance is either discharged to ground or is charged by  $V_R$  but never absorbs charge from the top plate. Therefore, the switch devices can be quite large permitting rapid redistributions. On the other hand the top plate of the array is connected to all the capacitors and to a switch and to the comparator resulting in a large parasitic capacitance from the top plate to ground. The nature of the conversion process, however, is such that  $V_x$  is converged back towards zero -- its initial value. Hence the charge on this parasitic is the same in the final configuration as it was in the sample mode. Therefore the error charge contributed by this parasitic is very near zero as will be further discussed in Chapter IV [17]. Because of this the smallest capacitor may be much smaller than the parasitic and consequently the largest capacitor may be reduced in value proportionally. Furthermore, the initial value of  $V_x$  need not necessarily be zero but can be the threshold voltage of the comparator. This fact allows cancellation of comparator offset by storing the offset in the array during the sample mode. The linearity then is primarily a function of the ratio accuracy of the capacitors in the array.

By only a slight modification of the array switching scheme bipolar voltage inputs can be encoded while still using only the single positive reference. This is achieved by connecting the bottom plate of the largest capacitor to  $V_R$  during the sample mode resulting in a stored charge:

$$Q_x = - C_{TOT} \left( \frac{V_{IN}}{2} + \frac{V_R}{2} \right)$$

Each bit is then tested in sequence just as before except that the largest capacitor is switched from  $V_R$  to ground during its test, while all the other capacitors are switched from ground to  $V_R$ . Also as before a bit value is true if  $V_x$  is negative after the test. The expression for  $V_x$  again converges back towards zero:

$$V_x = -\frac{V_{IN}}{2} + V_R \left( -\frac{D_1}{2^1} + \frac{D_2}{2^2} + \frac{D_3}{2^3} + \dots + \frac{D_{10}}{2^{10}} \right) \approx 0$$

$D_1$  is '0' for  $0 \leq V_{IN} \leq 10V$ , but is '1' for  $-10V \leq V_{IN} \leq 0$ . Therefore  $D_1$  represents the sign bit and its function is to level shift  $V_x$  in order to accommodate negative inputs. Hence a 10-bit conversion is achieved over the input range  $\pm 10V$  with negative numbers expressed in 1's complement.

### 3.5 Summary

The fundamental basis for a successive approximation algorithm using charge-redistribution between binary weighted capacitors was examined in this chapter. The development of a circuit which implements this algorithm and the subsequent verification of the RADCAP method is the subject of later chapters.

## CHAPTER IV

### Factors Limiting Accuracy in RADCAP Type of Circuits

#### 4.1 Introduction

In this chapter the factors limiting accuracy in RADCAP (MOS, successive approximation, charge-Redistribution, ADC utilizing weighted CAPacitors) type of circuits will be examined. The qualitative and quantitative compilation of these effects is essential in order to assess the advantages and disadvantages of various forms of capacitor structures and capacitor networks. The analysis of these effects led to a compatible MOS capacitor design geometry which is optimized to maintain ratio accuracy when conventional photolithography is used. In section 4.2 solutions will be proposed for the problem of input offset voltage cancellation for an MOS comparator. The effects of parasitic capacitance from the capacitor plates to ground will be investigated in section 4.3. Next the significance of temperature and voltage coefficients of capacitance and dielectric relaxation will be evaluated and an estimate will be made of how these affect the ability to design precision capacitors. Effects such as current leakage and parameter drift must also be considered. In addition an assessment is also made for the effects of several other factors which can cause ratio errors, some of which are capacitor oxide gradient and undercutting of the mask which defines the capacitors. Finally a description of the intrinsic offset voltage in RADCAP and its cancellation conclude this chapter.

#### 4.2 MOS Comparator Input Offset Voltage Cancellation

The voltage comparison process is fundamental to A/D conversion. The offset voltage of the comparator is usually manifested as an offset error in the digital conversion. Because of the relatively large gate-source



voltage mismatch in MOS differential amplifiers, the offset voltage of the all-MOS comparator needed for RADCAP must be eliminated as a source of error [18]. This can be accomplished either by digital means or by offset cancellation techniques. The problem is investigated with an ideal comparator shown in Figure 4.1. In this illustration the switching threshold voltage is zero and the gain at the threshold voltage is:

$$|A_c| = \frac{\Delta V_{out}}{\Delta V_{IN}} = \infty.$$

However, a real comparator (of any technology) may be modelled to a first order approximation by the transfer function shown in Figure 4.2. In this illustration the comparator has both input offset voltage  $V_{IOS}$  and output d.c. voltage  $V_{OOS}$ .  $V_{OOS}$  is not actually an offset error but rather is a d.c. switching level intermediate between the two logic levels. Hysteresis has been neglected and finite gain is modeled by the fact that  $|A_c| < \infty$ . Both d.c. voltages may tend to be large for an MOS comparator; however a large value of  $V_{IOS}$  causes a significant translation in d.c. voltage bias at the input. This tends to destroy the usefulness of the comparator for small signal inputs. Therefore the input offset voltage of an MOS comparator must be cancelled.

One particular method of performing offset voltage cancellation is illustrated in Figure 4.3. In this circuit the capacitor is precharged to the switching threshold voltage of the comparator. This is identical to the input offset if  $V_{IOS} = V_{OOS}$  as in the case of logic circuits. If the comparator gain is large in magnitude and negative then the momentary closure of S1 and S2 as shown during an initial precharge cycle forms a feedback path which stabilizes when  $V_x \approx V_{IOS}$ . When both switches are

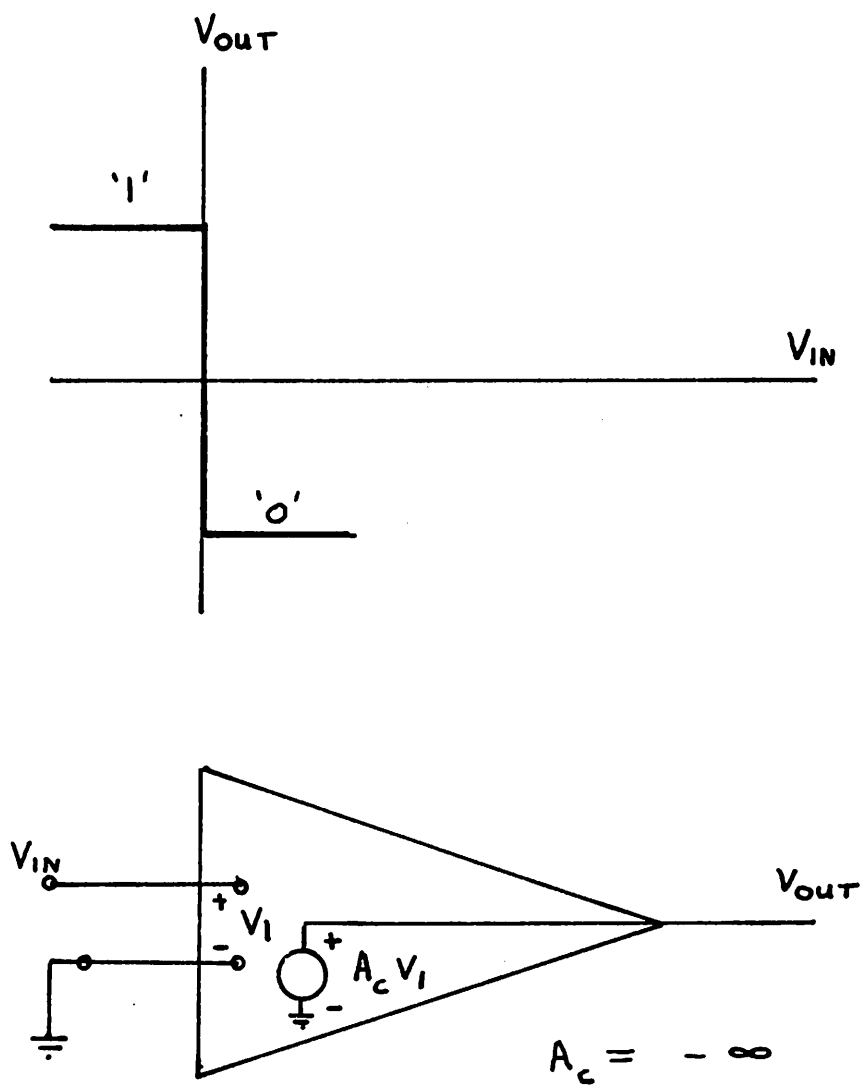


Figure 4.1: An ideal comparator.

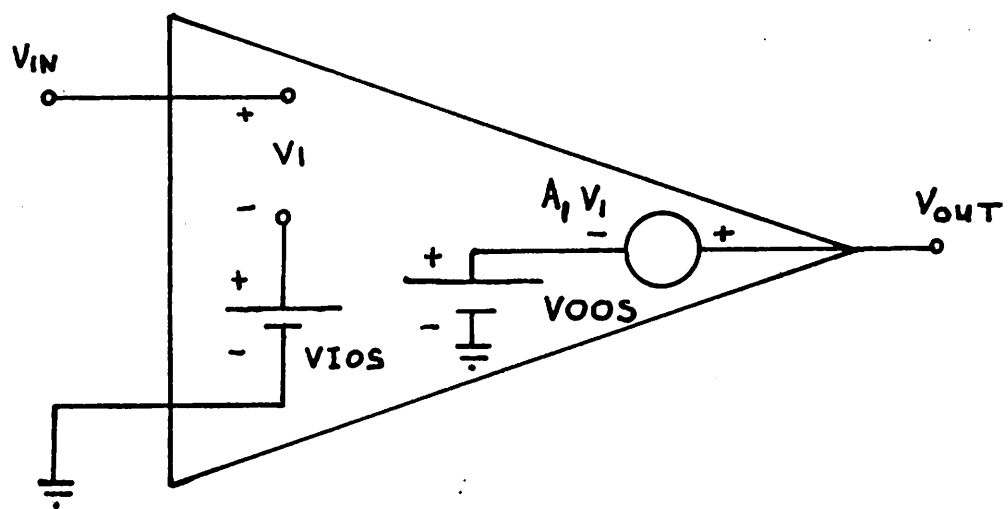
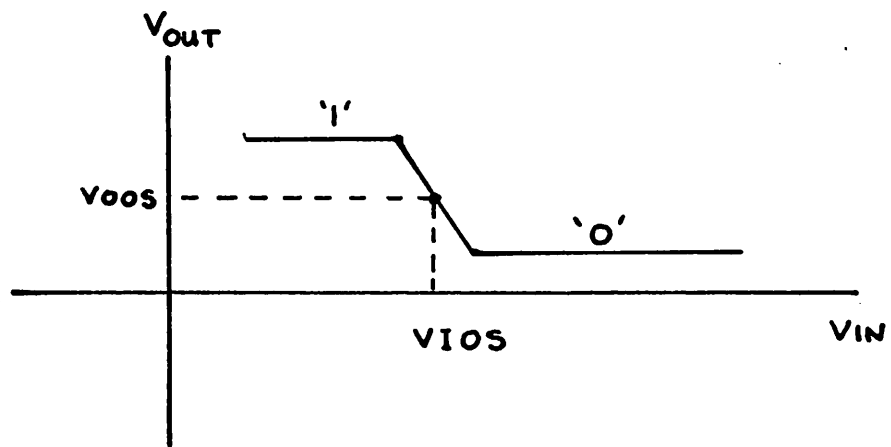


Figure 4.2: A real comparator.

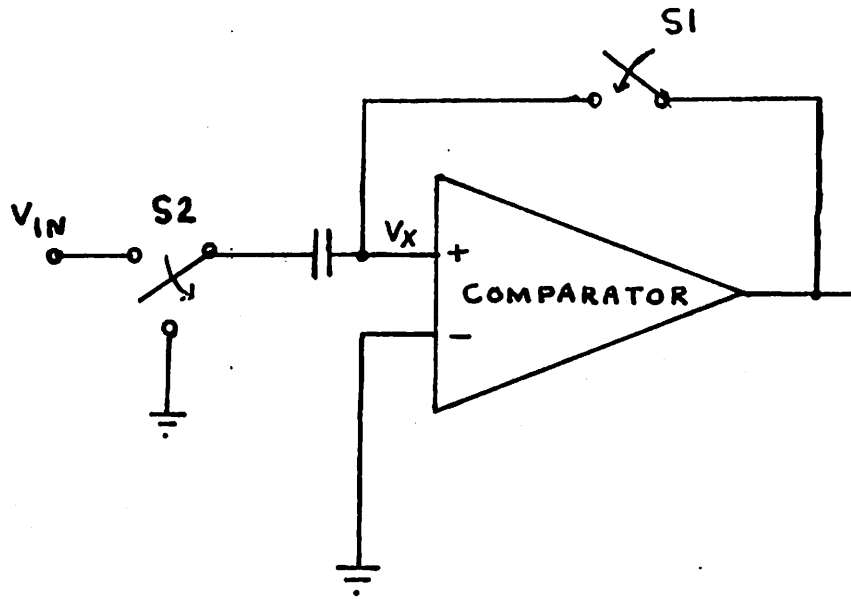


Figure 4.3: Offset cancellation by storing the offset in the capacitor.

opened the capacitor retains its d.c. charge and the effective offset referenced to  $V_{IN}$  is zero. If amplifier gain is low several stages are required in order to realize a high gain comparator. The example, shown in Figure 4.4 illustrates another technique to reduce the effects of input offset voltage. Two gain stages are used but offset cancellation by capacitor storage is performed only for stage A1. Hence there is no offset due to A1. However stage A2 has offset  $V_{IOS2}$ , but when this is reflected back to the input  $V_{IN}$  through A1 the effective offset is:

$$V_{INOS(EFFECTIVE)} = \frac{V_{IOS2} - V_{OOS1}}{A1}$$

If the gain magnitude A1 is large  $V_{INOS(EFFECTIVE)}$  is small. Hence this effective value may be reduced to an acceptable value. Both of the offset cancellation methods just discussed are used for realization of the RADCAP technique.

#### 4.3 Effects of Parasitic Capacitance from the Capacitor Plates to Ground

In RADCAP and VATCAP circuit techniques parasitic capacitance to ground exists at the top and bottom plates of the MOS capacitors. For either class of circuits the parasitic capacitance to ground from the bottom capacitor plates does not affect the accuracy of the charge-redistribution at the top plate. This is true because these parasitics are charged by  $V_R$  and discharged to ground by an MOS device which has no offset voltage when used as a charge switch. Therefore bottom plate parasitic capacitance does not participate in charge-redistribution at the top plate.

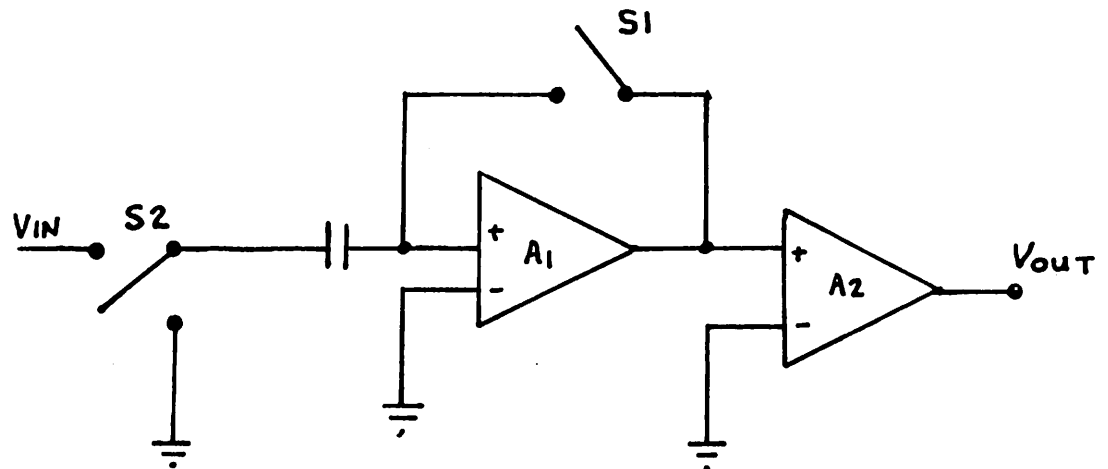


Figure 4.4: Reduction of input offset voltage by capacitive storage and by reflection through a high gain stage.

In VATCAP, however, the parasitic capacitance  $C_p$  to ground from the top plate of the capacitor network results in a charge-redistribution accuracy that is dependent upon the ratio of  $C_p$  to the total capacitance in the array  $C_T$ . The voltage signal  $V_x$  at the top plate in the VATCAP method is desired to be that of a precision attenuator or DAC:

$$V_x = B_i \frac{V_R}{2^N}.$$

However, if  $C_p \neq 0$  as illustrated in Figure 4.5 for a simplified version of VATCAP then

$$V_x = B_i \frac{V_R}{2^N} \left( \frac{1}{1 + \frac{C_p}{C_T}} \right), \text{ in which } C_p \ll C_T.$$

A gain error exists although the linearity and offset are unaffected provided that  $C_p$  is not heavily voltage dependent. One method of correcting this error is to increase the reference voltage to a new value  $V_R'$  such that:

$$V_R = \left( \frac{V_R'}{1 + \frac{C_p}{C_T}} \right).$$

This procedure requires a stable external adjustable reference voltage.

In contrast, RADCAP incorporates VATCAP in such a way that  $C_p$  is not important as was asserted in section 3.4. This will now be demonstrated with the aid of Figure 4.6. As explained in Chapter III the successive approximation algorithm used in RADCAP will force  $V_x$ , the voltage at the top plate, to converge to zero plus or minus  $\epsilon_q$  in the final configuration. For the ideal converter, having an ideal comparator as modelled in Figure 4.1,  $V_x = B_i \frac{V_R}{2^N} - V_{IN}$  and  $-\epsilon_q \leq V_x \leq \epsilon_q$ . This was derived in Chapter II.

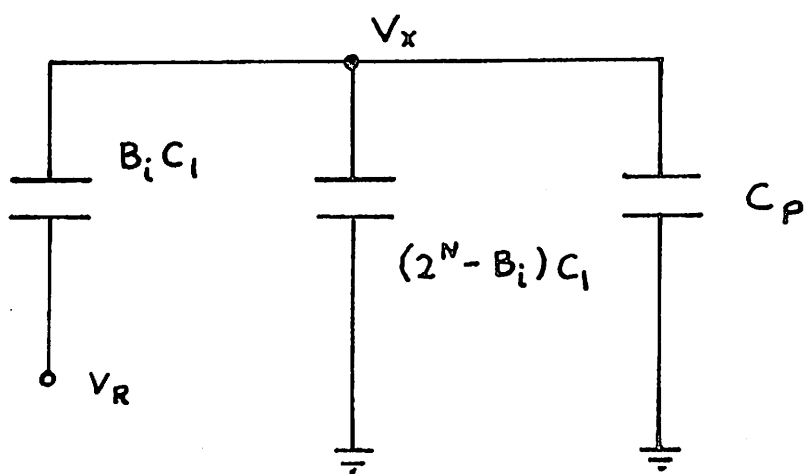


Figure 4.5: An illustration of top plate parasitic capacitance  $C_p$  at node X in the RADCAP circuit technique.  $B_i C_1$  represents the equivalent parallel connection of all capacitors with bottom plates connected to  $V_R$ .



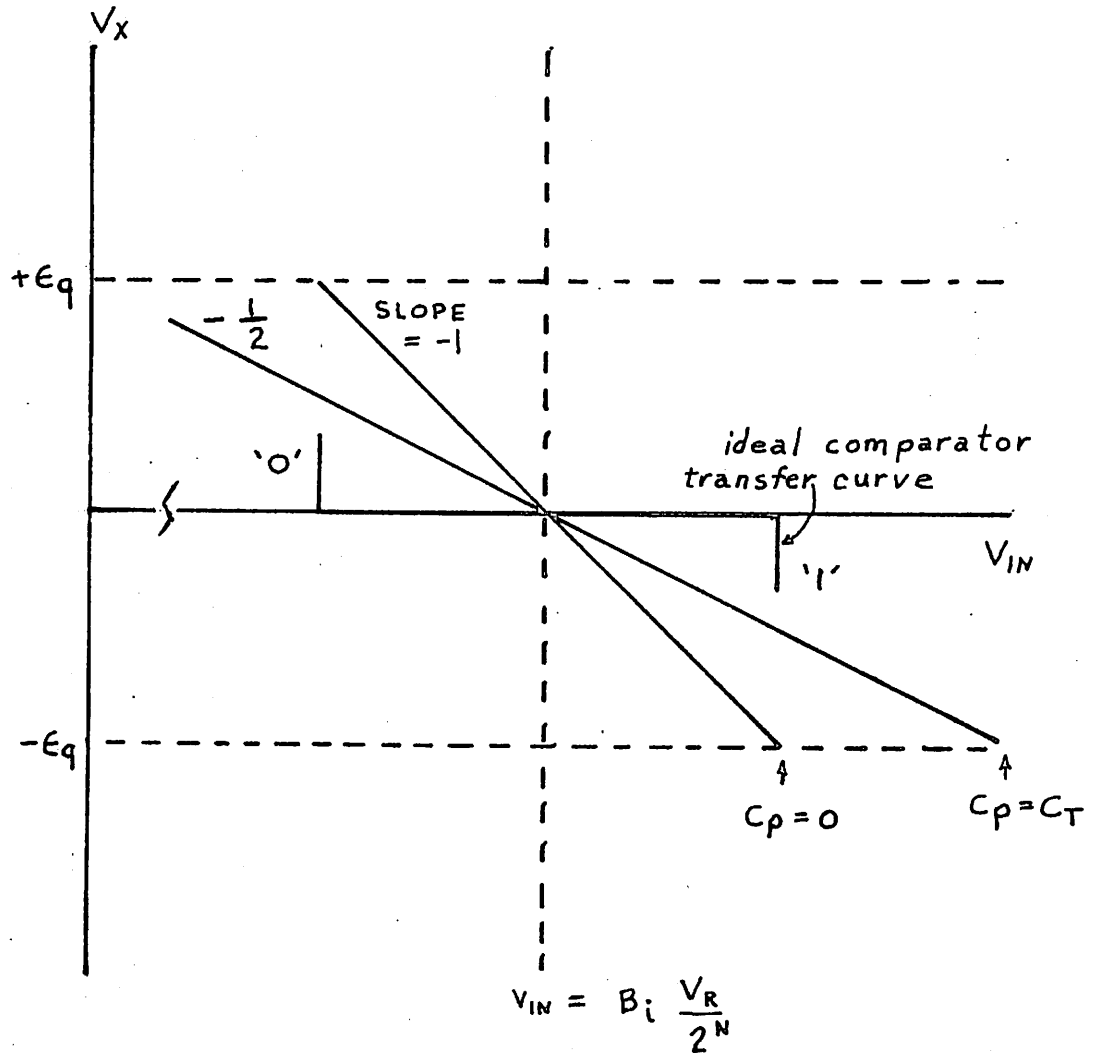


Figure 4.6: An illustration of the effect of  $C_p$  in RADCAP for an ideal comparator.

The expression for the total captured charge  $Q$  at  $x$  is:

$$Q = -2^N C_1 V_{IN} = V_x (2^N C_1 + C_p) - V_R B_i C_1.$$

From this equation and  $C_T = 2^N C_1$ :

$$V_x = \left( B_i \frac{V_R}{2^N} - V_{IN} \right) \frac{1}{1 + \frac{C_p}{C_T}}.$$

Here  $V_x$  represents an effective error voltage. The influence of  $C_p$  may now be determined with the aid of Figure 4.6. This is the plot of  $V_x$  vs  $V_{IN}$  for a small region around  $V_{IN} = B_i \frac{V_R}{2^N}$ . Two lines are plotted which show the behavior of the transfer curve due to  $C_p$ . One line for which  $C_p = 0$  has gain  $\frac{\Delta V_x}{\Delta V_{IN}} = -1$ , and the other for the unrealistically pessimistic case that  $C_p = C_T$ . In this latter case the gain has been reduced to  $-\frac{1}{2}$ . The ideal comparator transfer curve is also superimposed on the  $V_{IN}$  axis at the point  $(B_i \frac{V_R}{2^N}, 0)$ . The intersection of the comparator transfer curve and the transfer function for  $V_x$  is the point at which the ideal comparator switches. From observation this point is independent of  $C_p$ , hence  $C_p$  has no effect upon RADCAP if the comparator is ideal.

A different situation exists however for a real comparator having an uncancelled input offset voltage  $-V_{OS}'$  and a finite gain representing an uncertainty  $\Delta u$ . This is illustrated in Figure 4.7 for  $C_p = 0$  and  $C_p = C_T$ . For  $C_p = 0$  the offset reflected to the  $V_{IN}$  axis is  $V_{OS}'$  and the input uncertainty range is  $\Delta u$  about that offset. However the comparator transfer curve is then translated to the intersection of the lines for which  $C_p = C_T$  and  $V_x = -V_{OS}'$ . This models the unrealistic case in which parasitic capacitance is equal to the total capacitance in the array. From observa-

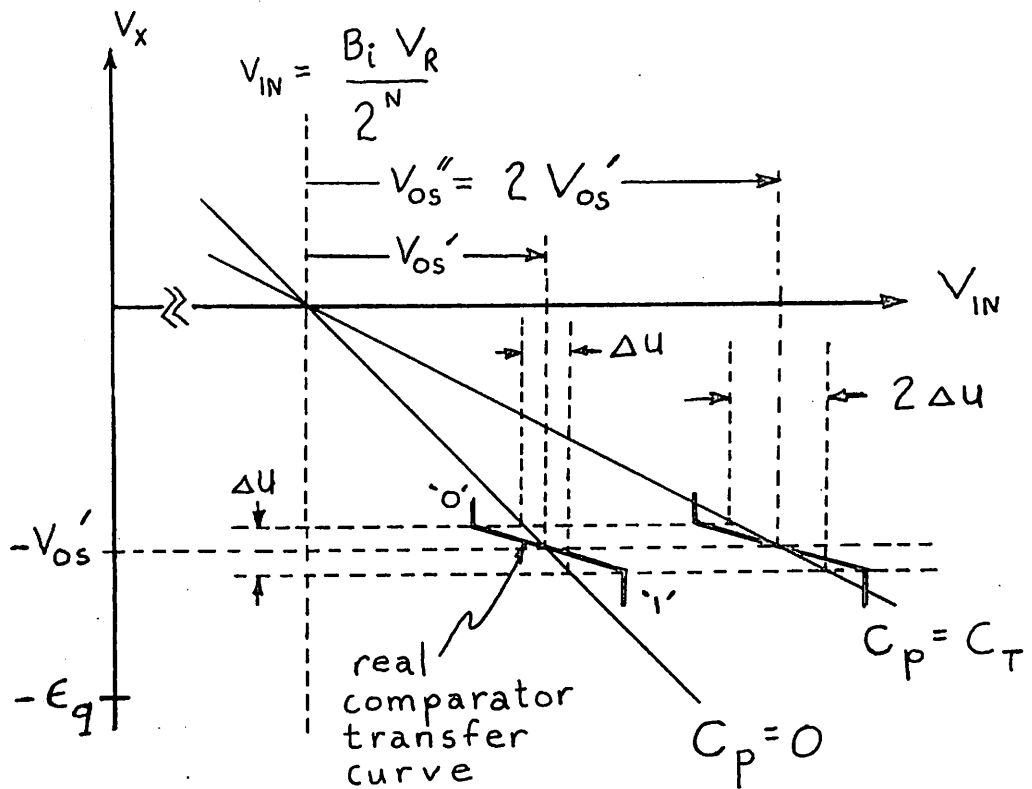


Figure 4.7: An illustration of the effect of  $C_p$  in RADCAP for a real comparator.

tion  $V_{OS}''$  has increased to  $2 V_{OS}'$  and the new uncertainty range is  $2 \Delta u$ . Both changes represent a finite but negligible performance degradation for the RADCAP system. These same results could have been developed by considering VATCAP to have an effective gain of

$$A_{VAT} = - \left( \frac{1}{1 + \frac{C_P}{C_T}} \right)$$

which is the slope of the transfer function of VATCAP:  $\frac{\Delta V_x}{\Delta V_{IN}}$ . Now offset reflection techniques may be applied to the system as modelled in Figure 4.8. Hence, a comparator offset  $-V_{OS}'$ , when reflected to  $V_{IN}$  through VATCAP results in an effective input offset voltage:

$$V_{INOS}(\text{effective}) = - \frac{V_{OS}'}{A_{VAT}} = V_{OS}' \left( 1 + \frac{C_P}{C_T} \right).$$

A similar relationship exists for the uncertainty range when reflected to  $V_{IN}$ . The quantitative effect of  $C_P$  may now be determined. For RADCAP techniques it is estimated that  $\frac{C_P}{C_T} \leq .05$  with normal layout guidelines, therefore it may be deduced that  $C_P$  has virtually no effect upon the RADCAP technique since  $V_{OS}'$  can be made small by offset cancellation methods. In addition the input uncertainty range can be reduced to a sufficiently low level as will be discussed later in Chapter VI.

#### 4.4 Temperature Coefficient of Capacitance

A great advantage of a  $\text{SiO}_2$  dielectric capacitor is its very low temperature coefficient of capacitance (TCC) of 24.8 ppm/ $^{\circ}\text{C}$  [19]. Capacitance values over the range  $-10^{\circ}\text{C}$  to  $140^{\circ}\text{C}$  for both  $\text{SiO}_2$  and  $p^+ n^+$  junction capacitors are shown in Figures 4.9 and 4.10 respectively. The

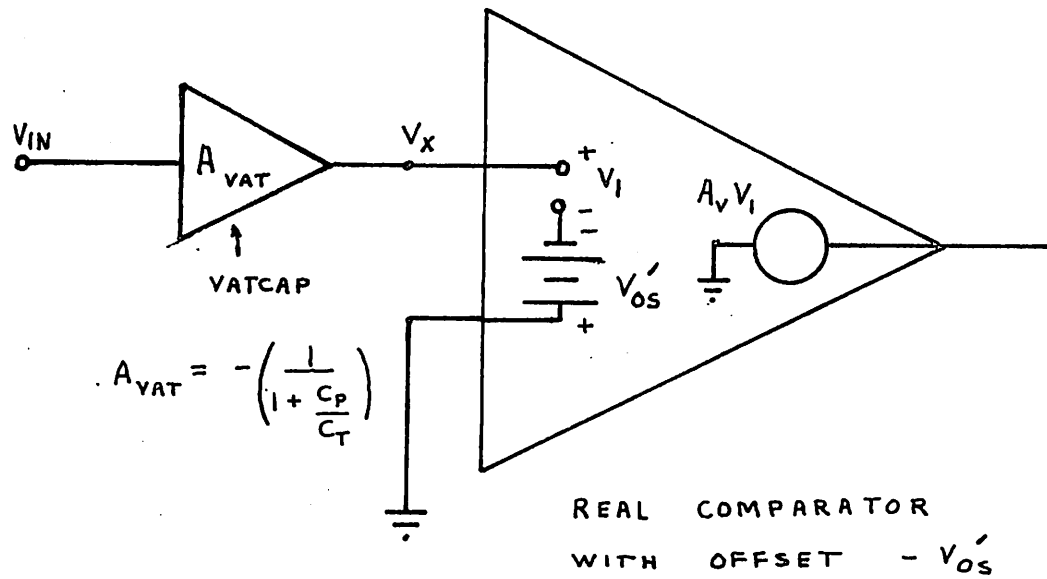


Figure 4.8: Analysis of the effect of  $C_p$  by reflection to the input,  $V_{IN}$ .

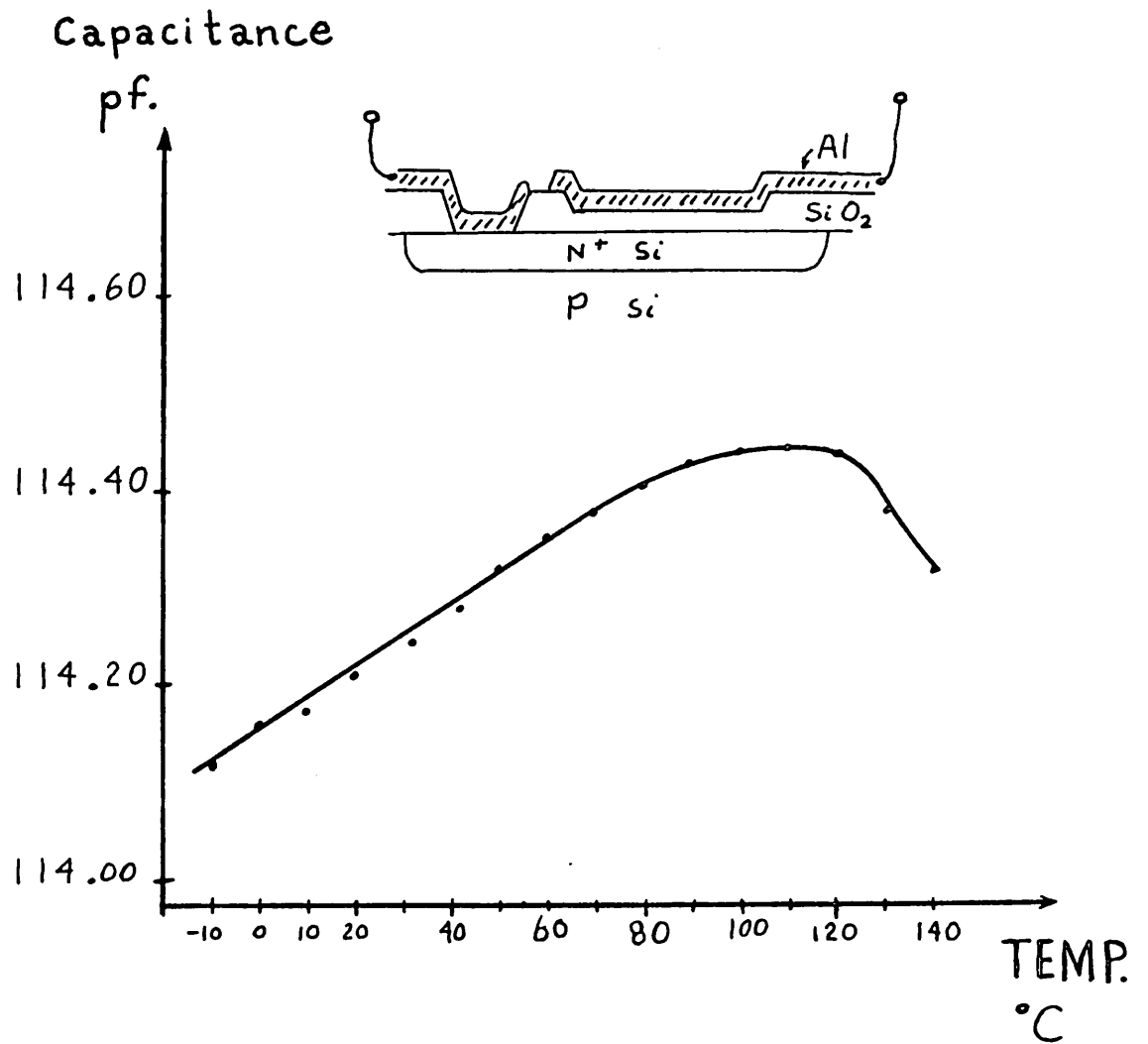


Figure 4.9: A plot of capacitance as a function of temperature for a  $\text{SiO}_2$  capacitor.

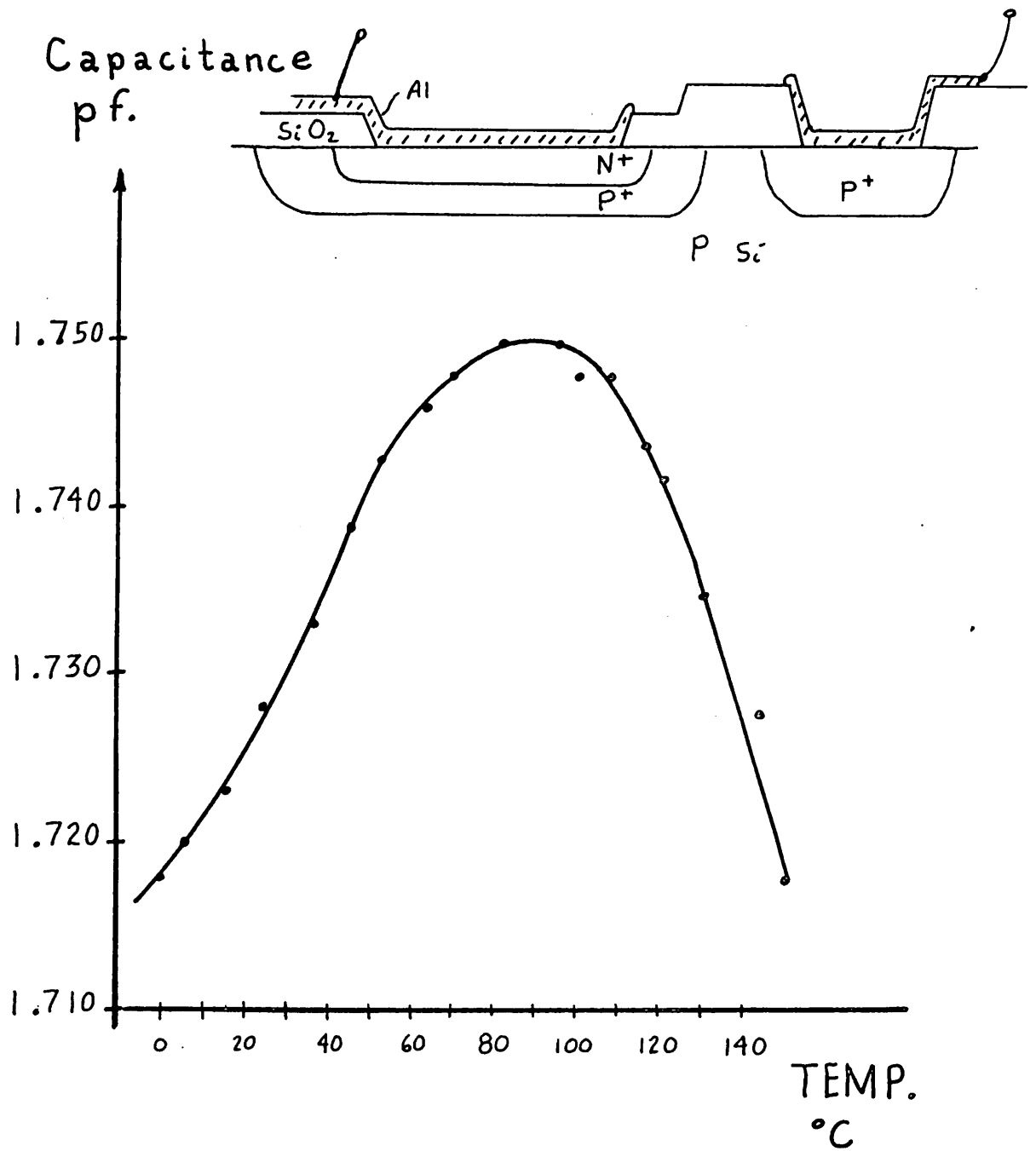


Figure 4.10: A plot of capacitance as a function of temperature for an  $N^+P^+$  junction under reverse-bias conditions. Refer to Appendix C for details relating to diffusion times and temperatures.

TCC of the  $\text{SiO}_2$  capacitor is contrasted with the temperature coefficient of resistance TCR in the range of 1000 to 2000 ppm/ $^{\circ}\text{C}$  for diffused resistors and several hundred ppm/ $^{\circ}\text{C}$  for thin-film and ion-implanted resistors [20]. This data is summarized in Table 4.1. An additional factor influencing circuit dependence upon temperature is that resistive networks used in high speed conversion techniques usually suffer from

Component	Typical
	Temperature Coefficient in ppm/ $^{\circ}\text{C}$
$\text{SiO}_2$ capacitor	+ 24.8
$\text{p}^+ \text{n}^+$ junction capacitor	+ 230
diffused resistor	+ 1500 [21]
thin-film resistor	- 200
ion-implanted resistor	+ 400

Table 4.1 Comparison of Component Temperature Coefficients over the military temperature range.

localized thermal gradients caused by the switching of large currents. Although some measures can be taken to reduce these effects, localized thermal gradients will still be present and will induce component mismatch errors. This situation is not present in RADCAP circuit methods since no large d.c. currents flow in the capacitors, hence no thermal gradients will exist due to the capacitor array. An additional benefit of charge-redistribution is therefore much lower power consumption. In addition the differential temperature coefficient (the variation in temperature coefficient between components) is expected to be less for  $\text{SiO}_2$  capacitors



than for resistors because TCC is an order of magnitude less than TCR. From these considerations it may be concluded that  $\text{SiO}_2$  capacitor matching is less dependent upon local and environmental temperature variations than resistor matching [22]. Since component mismatch has a great effect upon linearity, the temperature coefficient of nonlinearity due to mismatching is low for the RADCAP technique in comparison with resistive approaches.

#### 4.5 Voltage Coefficient of Capacitance

The value of a  $\text{SiO}_2$  capacitor having a metal top plate and a heavily doped  $\text{N}^+$  bottom plate is dependent upon d.c. terminal voltage [23]. This is due to the existence of accumulation, depletion, or inversion layers which may be formed at the  $\text{N}^+$  surface. When the terminal voltage is positive as illustrated in Figure 4.11 the N type surface becomes accumulated with mobile electrons and acts as a low resistance. The capacitance in this case is almost entirely due to the  $\text{SiO}_2$ . However, as the terminal voltage  $V_c$  becomes negative the N type surface begins to become depleted and a very thin, high capacitance, space charge layer is added in series with the  $\text{SiO}_2$  capacitor. The depletion capacitance becomes smaller as the reverse voltage increases and the depletion width widens. Therefore the total series capacitance decreases. At some value of reverse voltage  $V_c$  the surface becomes inverted and the depletion region width reaches its maximum value. This is illustrated by a minimum capacitance for some reverse bias in Figure 4.11. At a larger reverse bias voltage (more negative value of  $V_c$ ) the surface becomes inverted and the total capacitance approaches the same value as that during accumulation. The effect of surface doping concentration of the N type material is also illustrated in Figure 4.11. It may be seen that increased doping reduces the fractional rate of change of capacitance with voltage. The capacitor voltage coefficient  $\alpha$  is defined by the equation  $\alpha = \frac{dC(v)}{dv}$ .

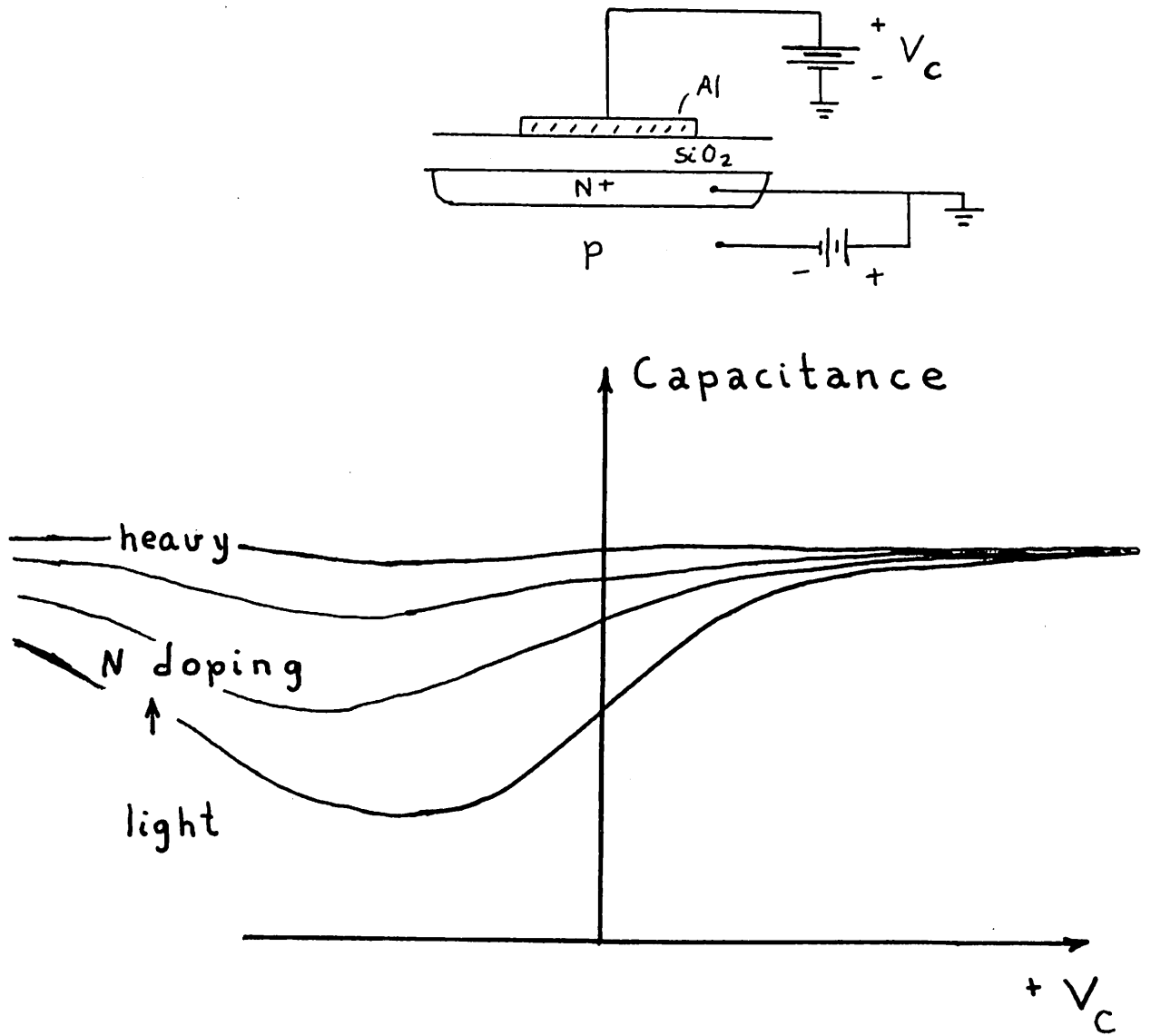


Figure 4.11: The voltage dependence of MOS capacitance.

A lower  $\alpha$  can be obtained by a higher surface concentration. For this reason  $N^+$  silicon is preferred as the capacitor back-plate since higher surface concentrations may be attained for N type impurities than P type impurities. The plot of capacitance versus voltage is shown in Figure 4.12. From this plot the linearly extrapolated  $\alpha$  is 21.9 ppm/volt over the region - 10V to + 10V.

The effect of this value of  $\alpha$  will now be illustrated with the aid of Figure 4.13. Any single charge-redistribution in the array may be modeled by a series combination of 2 capacitors C1 and C2. With the structures indicated:

$$C1(V1) = B_i C_0 (1 - \alpha V1)$$

and 
$$C2(V2) = (2^N - B_i) C_0 (1 + \alpha V2).$$

These equations are solved simultaneously in Appendix A from which the worst case error in V2 is  $\epsilon = -\alpha \frac{V_R^2}{8}$  which occurs for  $V2 = \frac{V_R}{2}$ . The normalized error distribution as a function of V2 is plotted in Figure 4.14. In this graph, for the case that  $V_R = 10$  volts, the error in mV is  $-.3\alpha'$  where  $\alpha' = \frac{\alpha}{22 \text{ ppm/V}}$  a normalization factor. Since the error is a function of V2 which is proportional to  $B_i$  and to  $V_{IN}$ , this error is manifested as a non-linearity error. For the 10-bit experimental RADCAP circuit this corresponds to a worst case nonlinearity of +.03 LSB due to voltage coefficient, and this is an insignificant value. It is expected that a modified fabrication schedule that would increase the  $N^+$  surface concentration would reduce  $\alpha$  by a factor of 3 [24] [25].

#### 4.6 Dielectric Relaxation

When a capacitor array is used as a precision voltage attenuator as in VATCAP, errors in voltage ratios can occur due to

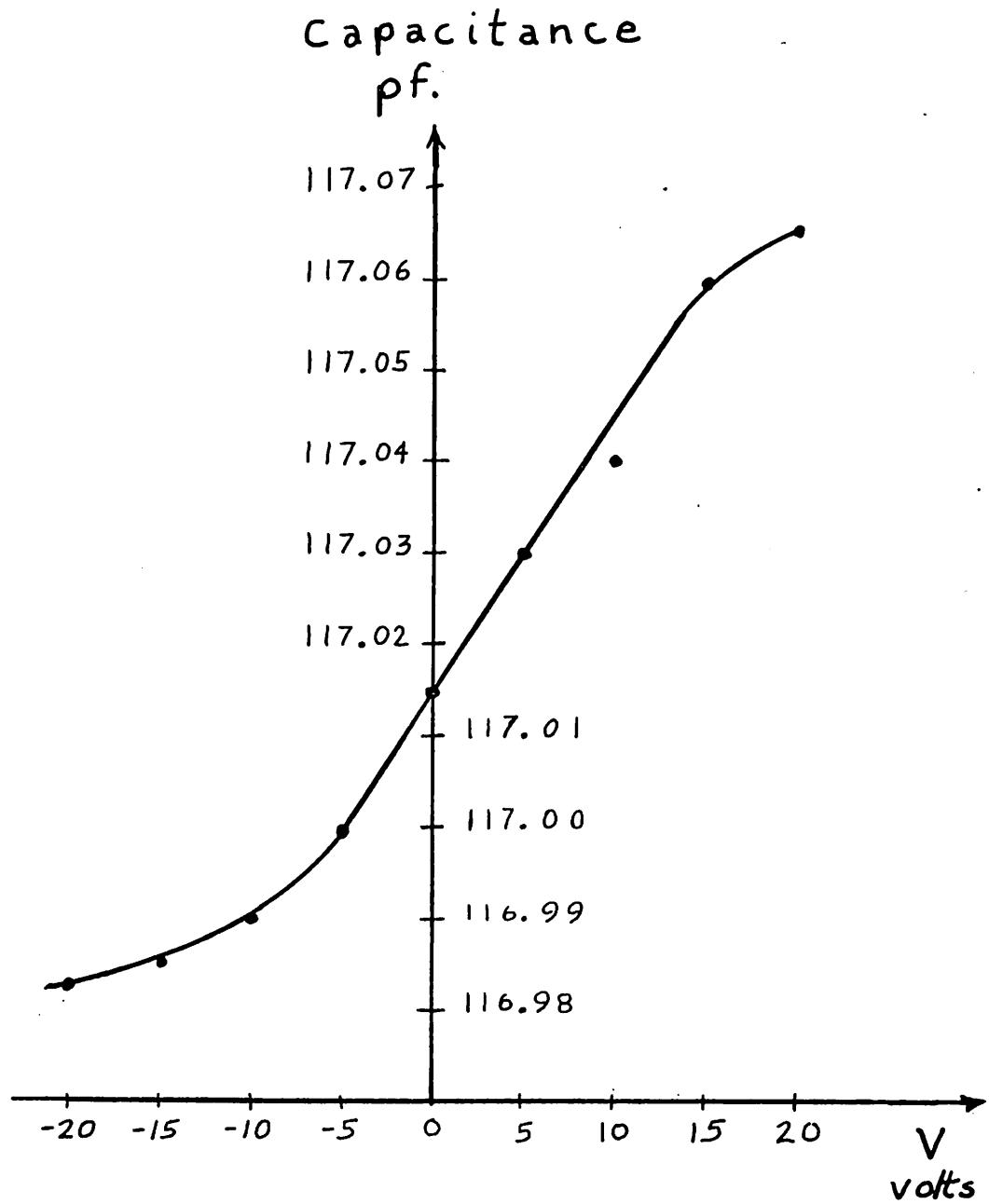


Figure 4.12: The measured voltage dependence of the largest capacitor in the array of an experimental I.C.

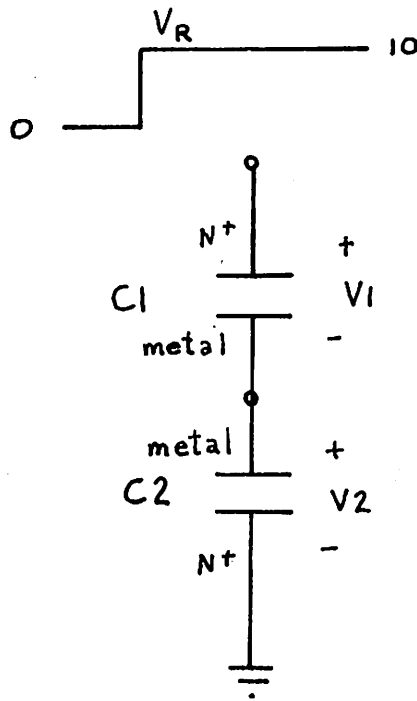


Figure 4.13: An equivalent circuit for the capacitor array.

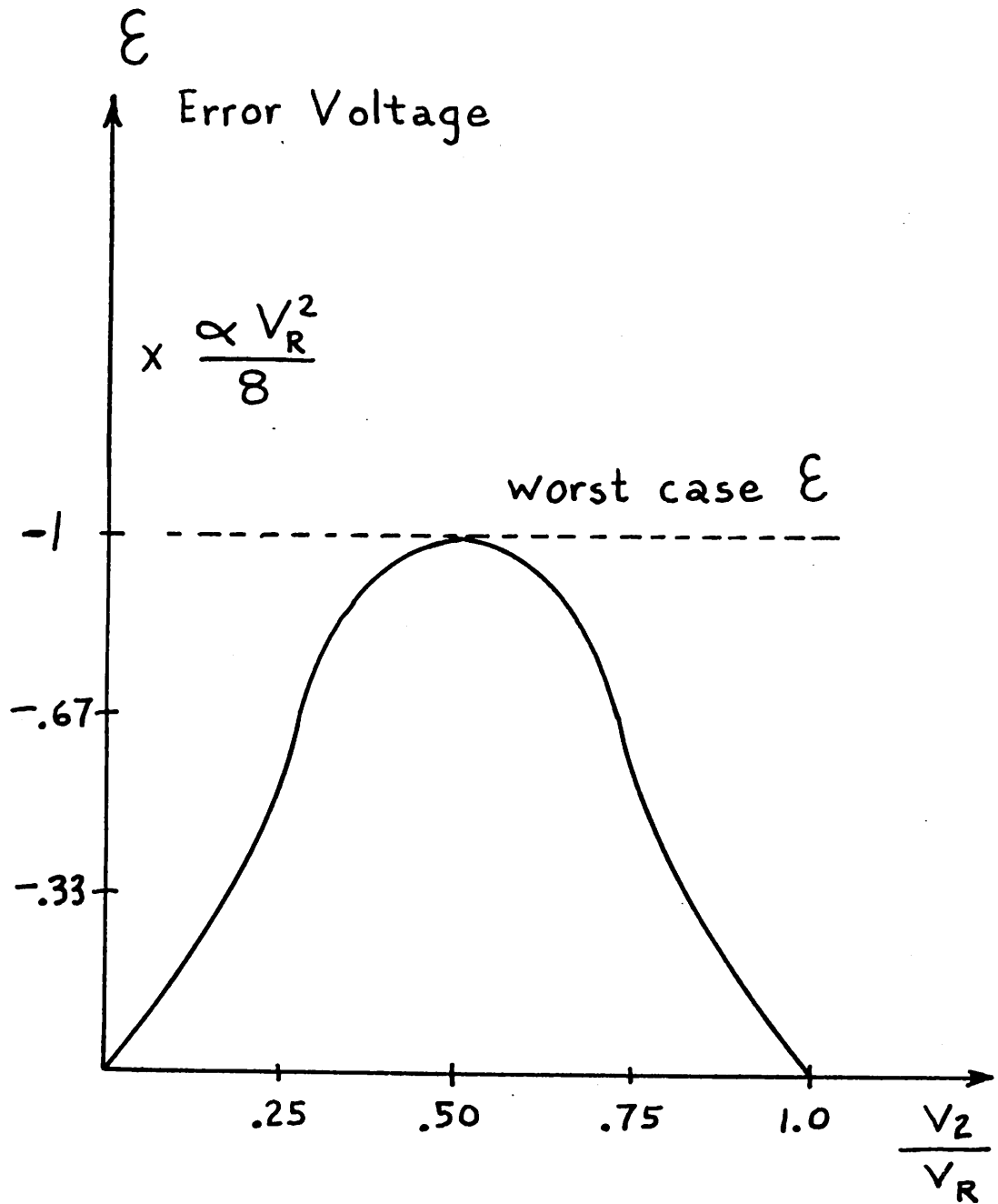


Figure 4.14: The normalized error due to capacitor voltage coefficient  $\alpha$ . The vertical scale normalization factor  $\alpha'$  is defined as  $\frac{\alpha}{\alpha_0}$  where  $\alpha_0 = 22$  ppm/volt.

dielectric relaxation. Consider the two-capacitor circuit of Figure 4.15 having dielectric losses modeled by parallel resistances. In the ideal case  $R_1 = R_2 = \infty$  and

$$V_x = \frac{C_2}{C_1 + C_2} V_R$$

in steady state. But for a real insulator having finite resistivity, the final voltage would be  $V_x = \frac{R_1}{R_1 + R_2} V_R$  which could result in errors. In general, the maximum observation time (or conversion time) for a capacitor network is related to the dielectric relaxation time  $\tau_R$ . For  $\text{SiO}_2$  dielectric capacitors of area  $A$  and oxide thickness  $t$ :

$$\tau_R = RC = \frac{t}{\sigma_{\text{SiO}_2} A} \frac{\epsilon_{\text{ox}} A}{t} = \frac{\epsilon_{\text{ox}}}{\sigma_{\text{SiO}_2}}$$

where the conductivity of  $\text{SiO}_2$ ,  $\sigma_{\text{SiO}_2} < 10^{-16} \Omega \text{ cm}$  and  $\tau_R = 3000 \text{ sec}$ .

From these calculations the observation time must be much less than 3000 sec.

The parallel resistance actually models only one dielectric defect, the steady state generation, drift and recombination of mobile carriers through the oxide due to the high electric field. There are other effects, however, which may cause more serious problems [26]. Polarized molecules in the dielectric may become aligned with the electric field resulting in a residual polarization after rapid discharge. This phenomena is generally not considered significant in  $\text{SiO}_2$  for two reasons; its time constant is much shorter than other circuit delays and the density of polar molecules may be kept to a small value by proper circuit fabrication techniques. Of a more serious consequence, however, is the density of mobile ions in the

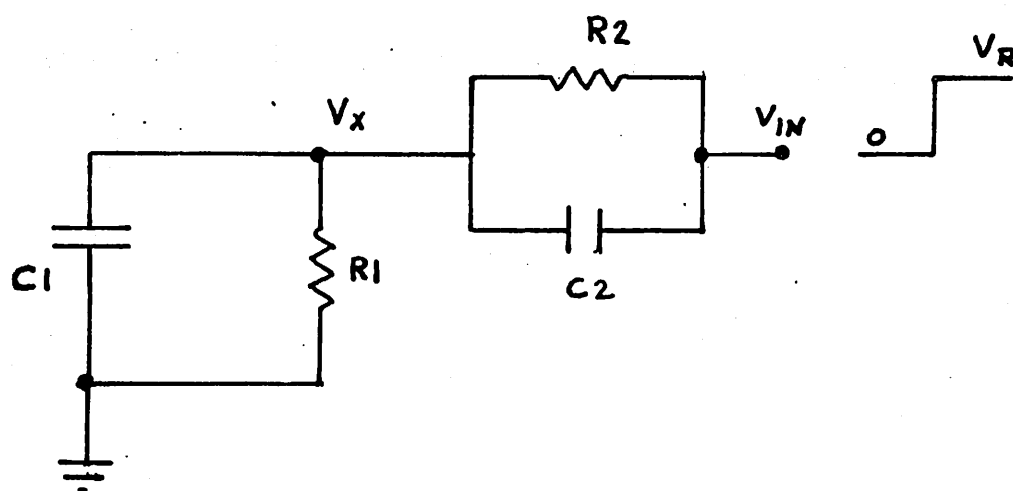


Figure 4.15: A 2-capacitor circuit modeling dielectric losses.



oxide or at its surface [27] [28]. These ions may drift in the electric field and accumulate at the oxide boundaries. This may result in a charge layer which dynamically augments the capacitance value. Furthermore if the concentration of mobile ions is very high, the conductivity of the oxide may appear to be large during the charging or discharging transient. Contamination of the oxide by alkali ions and water molecules enhances this dielectric relaxation phenomena by increasing the magnitude of the defects and reducing the relaxation time.

A typical dielectric relaxation effect is shown in Figure 4.16. These curves represent the voltage on the top plate  $V_x(t)$  as the bottom plate of the largest capacitor is pulsed up to  $V_R$  for 50  $\mu$ s then returned to ground. Figure 4.16(a) illustrates severe dielectric relaxation. This is manifested as an apparent residual voltage of opposite polarity remaining on the capacitor after it has been rapidly discharged. The magnitude of the defect was measured to be 10 mV, decaying to 5 mV after 50  $\mu$ s. The 5 mV error had been previously detected though unexplained and subsequent investigation led to the discovery of the relaxation phenomena. This error alone created a nonlinearity of  $\frac{1}{2}$  LSB. After some experimentation it was found that a mild heat treatment at 150°C to 200°C for 5 or 10 minutes reduced the magnitude of the effect beyond resolution capabilities as shown in Figure 4.16(b). In this case the relaxation effects had vanished within the surrounding system switching noise. These results tend to support the hypothesis that the main component of relaxation effect was associated with moisture trapped within the oxide or at its surface.

In conclusion, data from this study indicates that all contributions to dielectric relaxation combined do not cause significant error at the

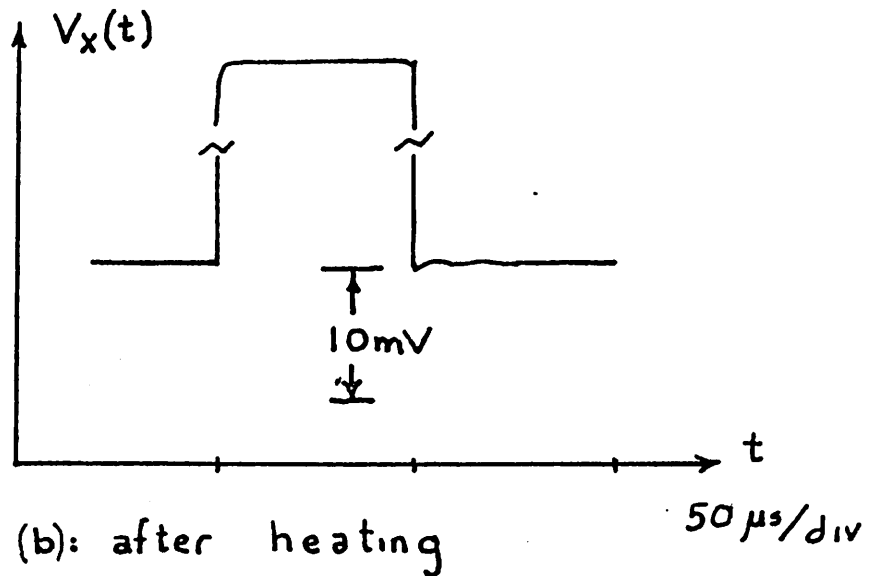
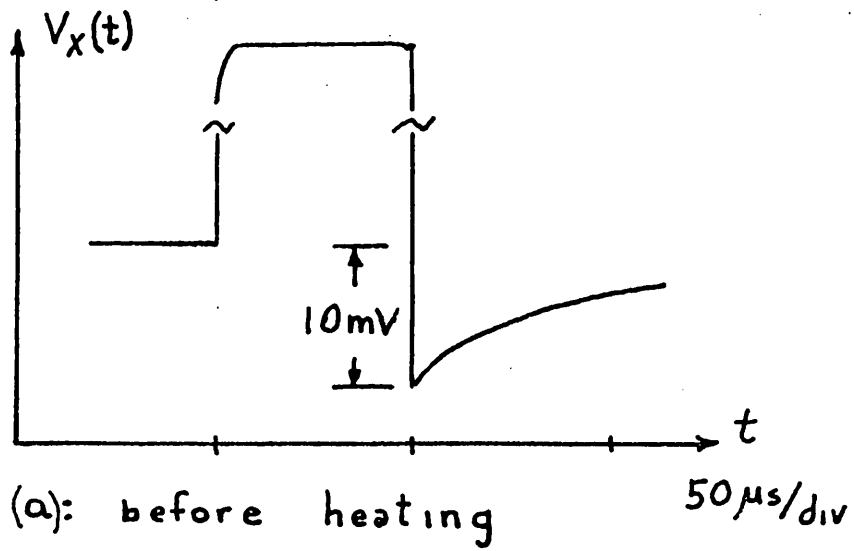
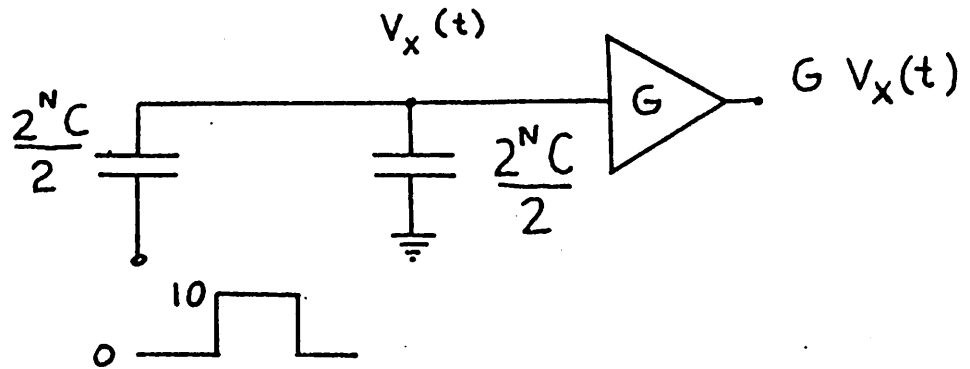


Figure 4.16: An example of a dielectric relaxation effect.

10-bit level.

#### 4.7 Leakage Currents

A large leakage current from a reversed-biased pn junction connected to a capacitor plate can cause an error in capacitor voltage. This could result in an offset error in a circuit employing the VATCAP technique. This problem is especially acute in the case of high leakage junctions ( $> 1\text{nA}/\text{mil}^2$ ) [29]. This is modeled in Figure 4.17. In the capacitor array technique the conversion time must be less than the time required for  $\frac{1}{2}$  LSB error due to leakage. By estimating normal leakage to be  $50(2)^{\frac{(T-25)}{10}}$  pA/mil<sup>2</sup> as a function of Centigrade temperature T, then for the case of a 10 mil<sup>2</sup> junction at 75°C the maximum observation time for a voltage loss of less than  $\frac{V_R}{2^{N+1}}$  is:

$$t = \frac{2^N C_1}{.5\text{nA}(2^5)} \frac{V_R}{2^{N+1}} = \frac{C_1 V_R}{32 \text{ nA}} = 63 \text{ } \mu\text{s}, \text{ for}$$

$C_1$  and  $V_R$  arbitrarily picked to be .2pF and 10 V. In this example the conversion must be completed within 63  $\mu\text{s}$  or else greater than 1/2 LSB offset error will result. Since the nominal conversion time is about 20  $\mu\text{s}$ , for RADCAP leakage effects will not be significant.

#### 4.8 Parameter Drift

Parameter drifts in I.C. devices may result due to environmental temperature changes [30]. In most circuit designs this is not a serious problem if the circuit performance is dependent upon device ratios rather than upon absolute values. On the other hand large d.c. currents in an I.C. can cause localized thermal gradients. In conventional resistor network converters a symmetrical layout is usually required

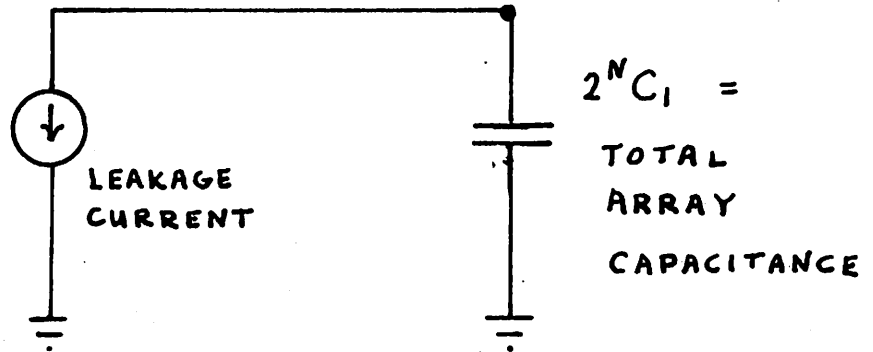


Figure 4.17: A circuit modeling leakage current from the top plate of the array.

to reduce the effects of these gradients. In contrast, there are no d.c. currents flowing in the capacitor network in RADCAP circuit methods hence no thermal gradients are caused by the capacitor array. However power dissipation in the logic circuitry may still cause thermal gradients unless proper design methods are used.

Long term parameter drifts are a characteristic problem with thin-film networks which are used in some converters [31]. Some improvement in stability is usually achieved at the cost of additional passivation layers during fabrication.

#### 4.9 CAPACITOR RATIO ERRORS IN RADCAP

##### 4.9.1 Capacitor Matching versus Resistor Matching

While monolithic circuit technology has had great impact on the cost of many analog circuit functions, such as operational amplifiers, the impact on the cost of A/D and D/A converters has not been as great. This is due to the complexity of a complete converter, and more importantly, to the problem of component matching. Because of the difference between the aspect ratios of diffused resistors of practical value versus those of capacitors, the attainable matching accuracy is higher for capacitors given the same overall die area [32]. The flexibility of capacitor geometry allows them to be made square or even circular so as to optimize matching accuracy. This is illustrated in Figure 4.18 by the fact that the resistors are usually long and thin. The resistor and capacitor are normalized for comparison by the constraint that they both have the same area  $16w^2$  where  $w$  is a small dimension. The aspect ratio 16:1 would be the worst case

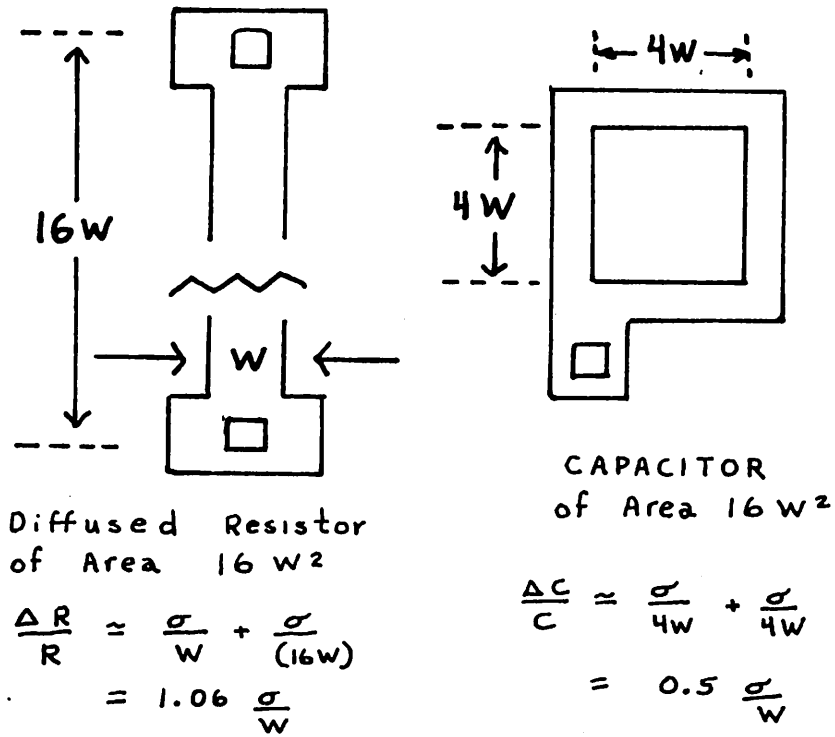


Figure 4.18: A comparison of capacitor matching and resistor matching.

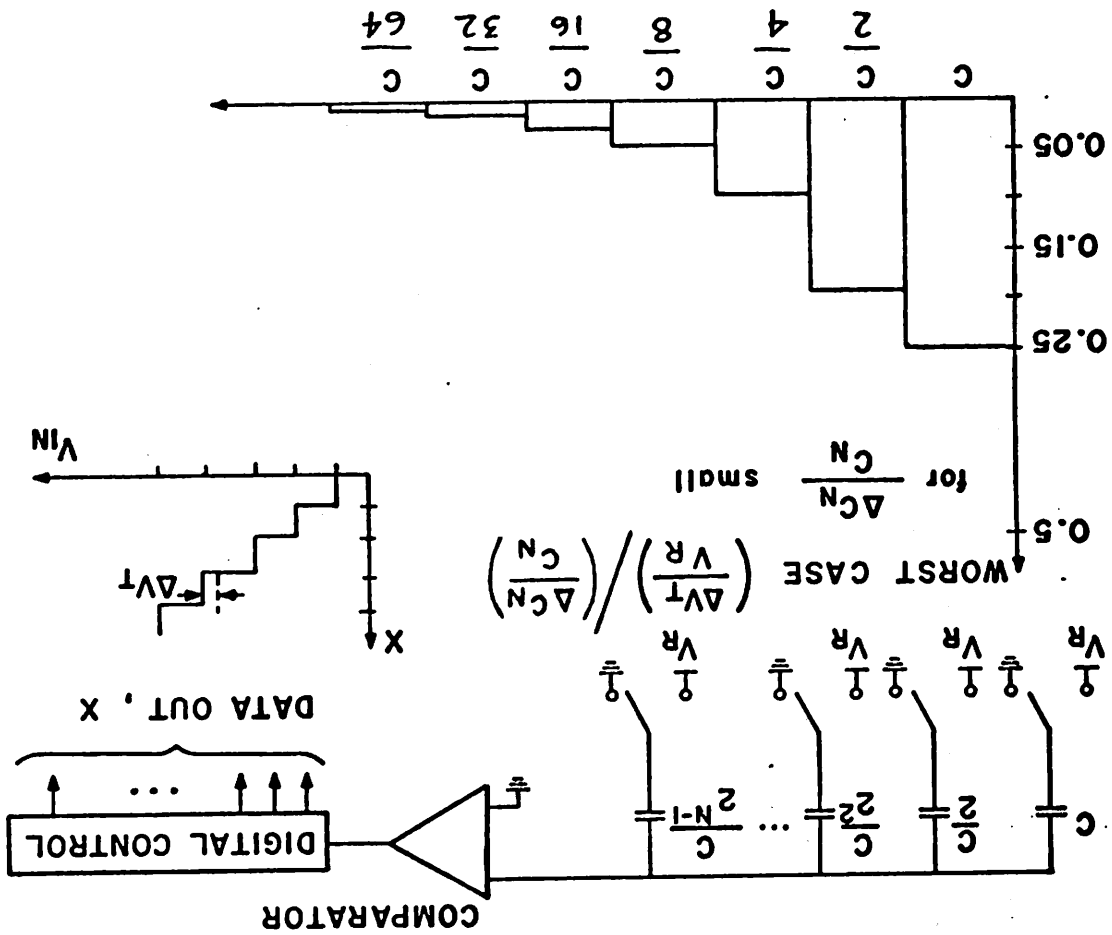
value required for an 8-bit binary weighted resistor string. Then if  $\sigma$  is the uncertainty in line width due to the photolithography, the fractional variation in the resistor value is:  $\frac{\Delta R}{R} = 1.06 \frac{\sigma}{w}$ , but for the capacitor:  $\frac{\Delta C}{C} = 0.5 \frac{\sigma}{w}$ . Hence from a purely geometric argument, the capacitor matching is better by a factor of 2 for this example. Actually there is a third dimensional variable involved and this is the sheet resistance for the resistor and oxide thickness for the  $\text{SiO}_2$  capacitor. Measurements indicate that excellent thin oxide uniformity can be obtained over an entire wafer.

#### 4.9.2 Nonlinearity due to Ratio Errors

Consider the ideal case for a RADCAP type of circuit in which all the capacitors of the converter shown in Figure 4.19 have the precise binary weight values. For this case the digital output  $x$  is a regular staircase when plotted against  $V_{IN}$ , and every transition occurs at a precise value of  $V_{IN}$  designated  $V_T$ . On the other hand, changing one capacitor from its ideal value by a small amount  $\Delta C_N$  causes all transition points to shift somewhat but there will be one worst case transition. The ratio  $\frac{\Delta V_T}{V_R}$  is the normalized worst case fraction deviation in transition point from the ideal. This is also a measure of the nonlinearity. The ratio of this deviation to the fractional change in capacitor value  $\frac{\Delta C_N}{C_N}$  represents the sensitivity of linearity to individual capacitor value. The plot of sensitivity also in Figure 4.19 shows that linearity is very sensitive to a fractional change in the large capacitors, but not very dependent upon similar fractional changes in the smaller capacitors. Therefore the smaller capacitors have greater allowable tolerances. It should be pointed out that actually all capacitors have simultaneous deviations

deviation in individual capacitor values for RADCAP.

Figure 4.19: The sensitivity of A/D conversion linearity to





which cause ratio errors and the worst case combination of these must always be considered.

It will now be shown that even a simultaneous mismatch in the binary ratios of capacitors in the array causes only nonlinearity. It cannot cause a gain error because the end points of the transfer function,  $B_i$  vs  $V_{IN}$ , which is shown in Figure 2.3, are not dependent on capacitor matching. This results from the fact that no final charge-redistribution between capacitors occurs for either zero or full-scale inputs since all capacitors are either fully discharged or fully charged respectively. For the same reason no offset error can arise from capacitor mismatch since the mismatch cannot be manifested unless a charge-redistribution exists in the final configuration. This may be demonstrated analytically with the aid of Figure 4.20 for an N-bit RADCAP circuit. It is illustrated in the figure that for zero or full-scale inputs, the two boundary points in which all capacitors may be paralleled together, the resultant total mismatch error is zero. Another observation is that for  $B_K = 2^N/2$  the total deviation is

$$\sum_{i=2^{N-1}}^{2^N-1} \Delta C_i, \text{ but for } B_K - 1 = 2^{N-1} - 1, \text{ the deviation is } \sum_{i=2^{N-2}}^{1B} \Delta C_i.$$

Since  $\sum_{i=2^{N-1}}^{1B} \Delta C_i$  equals zero then

$$\sum_{i=2^{N-1}}^{2^N-1} \Delta C_i = - \sum_{i=2^{N-2}}^{1B} \Delta C_i.$$

The significance of this is that the real transfer function must also pass through the midpoint of the ideal transfer function (considering only

Real Values	Ideal Values $C_T = 2^N C_1$	Capacitor Deviation
$C_{2^{N-1}}$	$= C_T/2$	$+ \Delta C_{2^{N-1}}$
$C_{2^{N-2}}$	$= C_T/4$	$+ \Delta C_{2^{N-2}}$
$\vdots$	$\vdots$	$\vdots$
$C_{1A}$	$= C_T/2^N$	$+ \Delta C_{1A}$
$C_{1B}$	$= C_T/2^N$	$+ \Delta C_{1B}$
$C_T = \sum_{i=2^{N-1}}^{1B} C_i$	$= C_T$	$+ \sum_{i=2^{N-1}}^{1B} \Delta C_i$
	$\circ \circ \circ \sum_{i=2^{N-1}}^{1B} \Delta C_i = 0$	

Figure 4.20. An illustration that the sum of capacitor deviations equals zero for RADCAP.

capacitor ratio errors). Moreover the generalized result:

$$\sum_{\substack{i = \text{all components} \\ \text{of } B_i}} \Delta C_i = - \sum_{\substack{j = \text{all components} \\ \text{of } \bar{B}_i}} \Delta C_j$$

where  $\bar{B}_i$  is the 1's complement of  $B_i$ , shows that the nonlinearity error at  $B_i$  is the negative of the nonlinearity error at  $\bar{B}_i$ . This may be summarized by stating that the capacitor ratio errors can cause only a nonlinearity, but that the behavior of this nonlinearity is such that the real transfer curve passes through the ideal midpoint and is an odd function about it. This is illustrated graphically in Figure 4.21 by the fact that both the ideal and real curves must intersect at the midpoint and the boundary points (considering only ratio errors which are independent of voltage).

#### 4.9.3 Uniform Undercut

An important source of capacitor ratio error is uniform undercutting of the photoresist which defines the capacitor. Consider two capacitors  $C_4$  and  $C_2$  shown in Figure 4.22 which are nominally related by a factor of 2:  $C_4 = 2C_2$ . During the etching phase of the photomask process a poorly controlled lateral etch occurs called undercut. In most areas of the silicon wafer this effect will be evenly distributed and therefore will be characterized by a uniform reduction of each edge. Let  $\Delta x$  be the undercut length and  $L_4$  be the side length of  $C_4$ , also as shown in Figure 4.22. Then a ratio error is produced between  $C_4$  and  $C_2$  which is proportional to the undercut length:

$$C_4 = 2C_2(1 + \epsilon); \quad \epsilon \approx 4 \frac{\Delta x}{L_4}$$

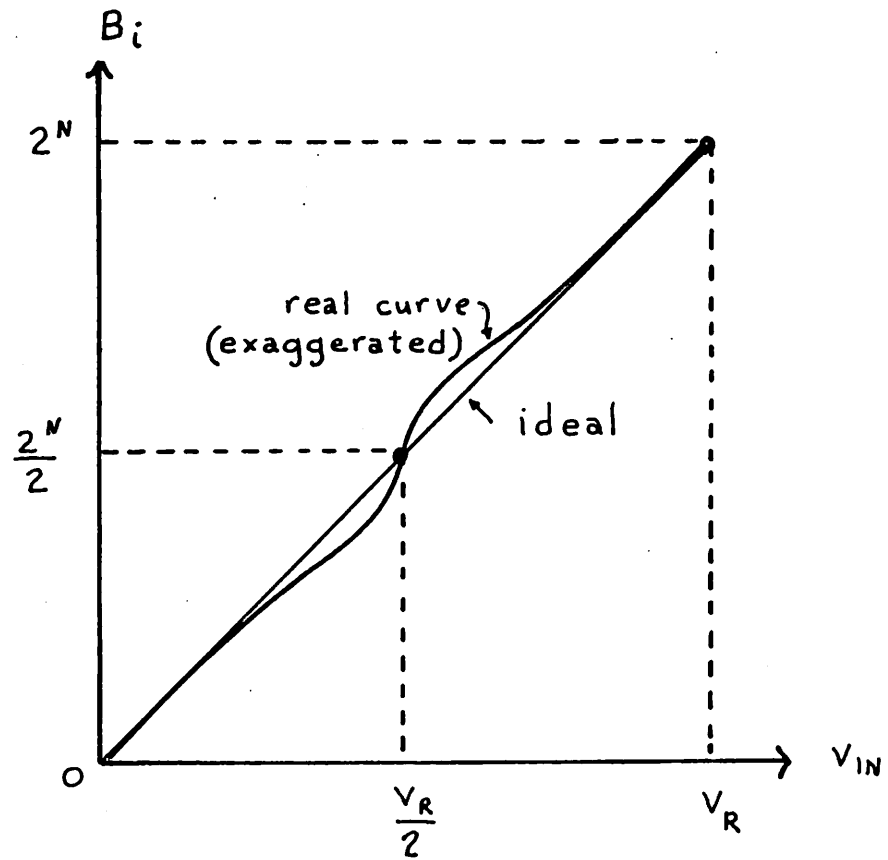
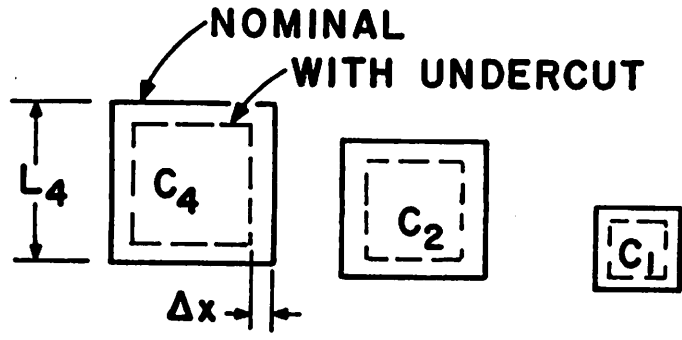


Figure 4.21: An illustration that capacitor ratio errors cause an odd-functioned nonlinearity about the input voltage midpoint.



NOMINAL :  $C_4 = 2 C_2$

WITH UNDERCUT :  $C_4 = 2 C_2 (1 + \epsilon)$ ;  $\epsilon \simeq 4 \frac{\Delta x}{L_4}$

UNDERCUT - INSENSITIVE GEOMETRY

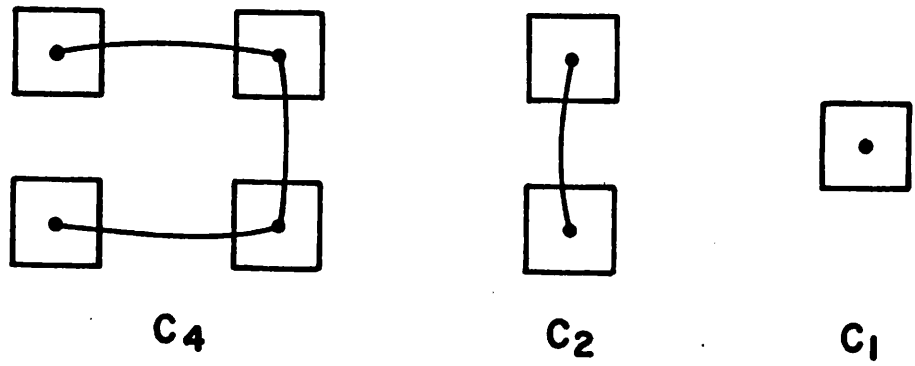


Figure 4.22: Capacitor ratio error due to photomask undercut.

This problem can be circumvented by a geometry such that the perimeter lengths as well as the areas are ratioed. This can be done as seen in Figure 4.22 by paralleling identical size plates to form the larger capacitors. Now the capacitor ratios are not affected by uniform undercut.

#### 4.9.3 Oxide Gradient

Long range gradients in the thin capacitor oxide can also cause ratio errors. These gradients arise from non-uniform oxide growth conditions. If this variation in oxide thickness is approximated as first-order gradient as shown in Figure 4.23, then the resulting ratio error is proportional to the fractional variation in oxide thickness:

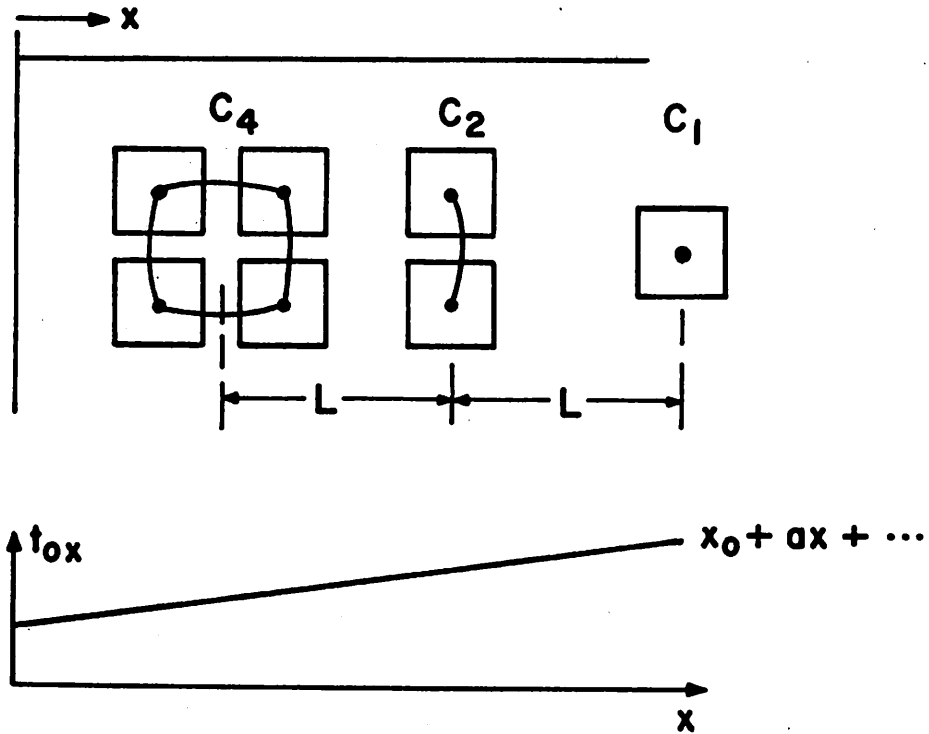
$$C_2 = 2C_1(1 + \epsilon L)$$

$$C_4 = 4C_1(1 + 2\epsilon L); \epsilon = \frac{a}{x_0}.$$

Experimentally, values of 10 to 100 ppm/mil have been observed for the factor  $\epsilon$ . Error from this source can be minimized by improved oxide growth techniques and by a common centroid geometry. This is done in Figure 4.23 by locating the elements of the capacitors in such a way that they are symmetrically spaced about a common center point. In this way the capacitor ratios may be maintained in spite of first-order gradients.

#### 4.9.5 Non-Uniform Undercut

Non-uniform undercut usually appears in three forms as shown in Figure 4.24. Large-scale edge distortion is not clearly understood, however, one mechanism may involve chemical saturation. The etchant solution may become saturated with the etched material in some areas of the chip,



$$C_2 = 2C_1 \left( 1 + \frac{aL}{x_0} \right)$$

$$C_4 = 4C_1 \left( 1 + \frac{2aL}{x_0} \right) \quad ; \quad \frac{a}{x_0} \sim 10 \rightarrow 100 \text{ ppm/mil}$$

### COMMON CENTROID GEOMETRY

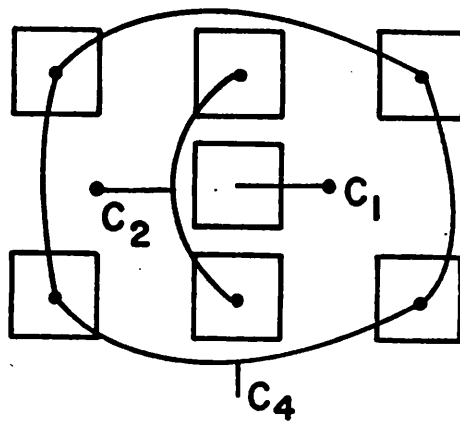
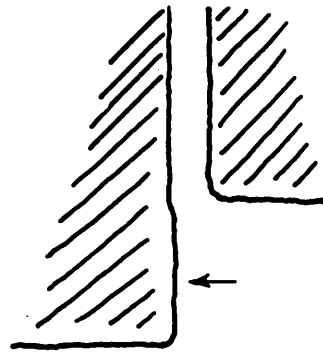


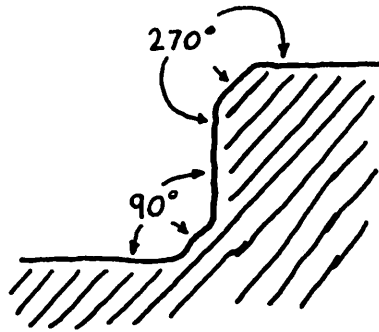
Figure 4.23: Capacitor ratio error due to oxide gradients.



a. LARGE - SCALE  
DISTORTION



b. RANDOM EDGE  
LOCATION



c. CORNER ROUNDING


 material being etched

Figure 4.24: Non-uniform undercutting effects.



causing different etch rates. In addition regional temperature gradients caused by differences in amount of material being etched may cause regional etch rate variations.

The second type of non-uniform undercut is random edge location about which even less information appears in the literature. However, four processes may be involved. First the granular nature of the aluminum may cause local density variations and therefore localized etch rate variations. A "grainy" aluminum is usually caused by evaporation onto a heated wafer. This is usually done to promote adhesion of the metal to the dielectric. A cooled substrate would be a dubious improvement since the temperature of the metallic vapor would remain the same although the cooling-rate of the metal would increase. Another mechanism which might induce a random edge location is the error associated with the photolithography. For example, the glass plate emulsion mask may not have smooth edges. Or, the developed photoresist may have jagged edges due to light interference patterns. Random edge location may also result from localized temperature differences at the edge being etched since etch rate is a function of temperature and the chemical reaction releases energy. Finally, random edge location may result from the formation of stagnant gas bubbles about nucleation sites which retard the etch rate at localized points.

The last form of non-uniform undercut is corner-rounding. One difficulty in making precision capacitors with conventional photolithography is that corners cannot be made sharp. Figure 4.24 illustrates the distortion that results for both  $270^\circ$  and  $90^\circ$  corners. One apparent remedy for minimizing corners is to design circular capacitors. However, this is an unsatisfactory solution for a binary weighted capacitor array because at least 4  $270^\circ$  corners will be required due to the interconnect and mask

alignment tab. A more realistic approach for a common centroid multiple plate capacitor array is one in which there are an equal number of  $270^\circ$  and  $90^\circ$  corners for each capacitor and also that the number of corners be binary ratioed between capacitors. As indicated in the figure this is an approximate solution since the excess area of the  $270^\circ$  corner is not exactly equal to that lost by the  $90^\circ$  corner.

Data indicates that large-scale edge variation may cause significant error at the 10-bit level, therefore, the design must involve control on this mechanism. A first-order control on etch rate is to locate every active edge the same distance from the opposite active edge. Furthermore, any reasonable fabrication changes which may reduce random edge should be considered.

#### 4.9.6 Mask Alignment

The capacitor ratio must be independent of mask alignment. This can be done in conventional photolithography with the aid of an alignment tab which is parallel with the interconnect as shown in Figure 4.25.

#### 4.9.7 Capacitor Ratio Error Due to Interconnect

Capacitors may differ from their design values due to the overlap of the interconnect on the thick oxide over the back electrode of the capacitor. The area of the interconnect which is effectively included in each capacitor value is not ratioed but is rather a constant offset added to each capacitor. Therefore this area must be included in design calculations in order to avoid ratio errors. Any absolute error in this overlap capacitance (as perhaps caused by thick oxide gradients) would cause a capacitor ratio error.

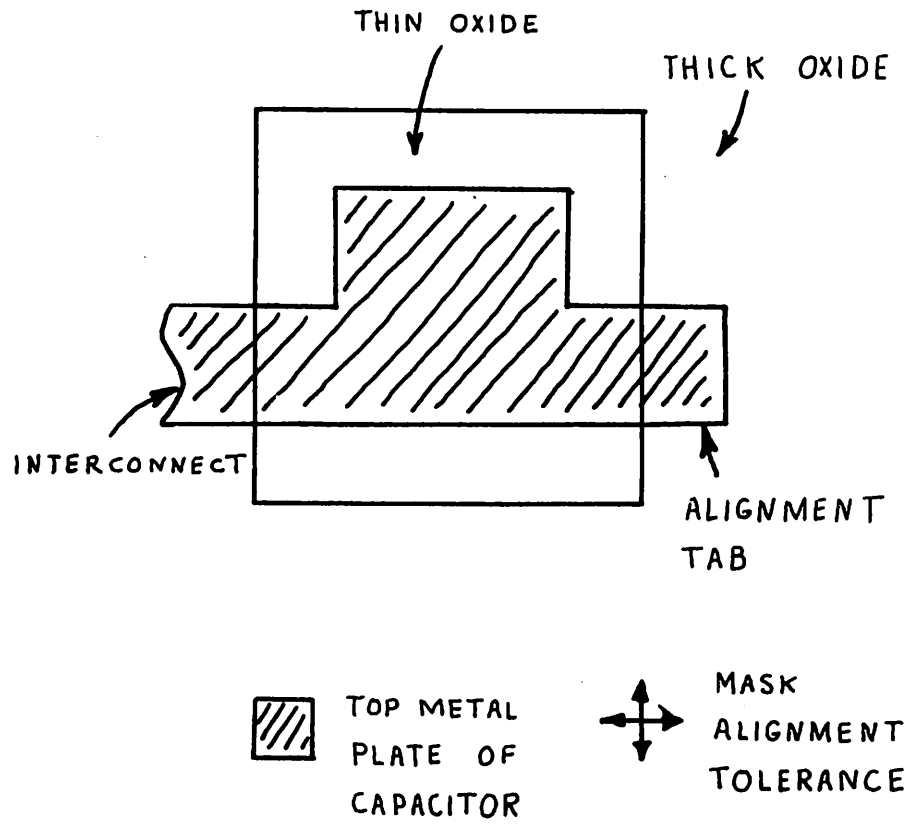


Figure 4.25: The insensitivity of capacitor area to mask alignment errors.

#### 4.9.8 Fringing

A fringe field exists at the edges of the parallel plates as shown in Figure 4.26. The effect of fringing is to increase each dimension of the smallest plate by an amount equal to its thickness [33]. If the capacitor structure is such that this same plate suffers uniform undercut, then there will be a first-order cancellation of undercut and fringing since undercut will tend to decrease the plate area by the same amount that it is increased by fringing. The two effects do not exactly cancel, however if the circuit is designed to be insensitive to undercut then it will automatically be unaffected by fringing.

#### 4.11 Intrinsic Offset

It has been previously stated that a capacitor ratio error does not cause an offset and that the offset due to parasitic capacitance and comparator is small. However an intrinsic offset of  $+\frac{1}{2}$  LSB still exists as indicated in Figure 4.27. The "intrinsic" transfer curve represents that for which the comparator is ideal and its offset has been cancelled as described in section 4.2. With the use of these techniques the comparator transfer function is defined by the equation

$$V_{\text{out}} = 1 \quad \text{if } V_x < V_T$$

but 
$$V_{\text{out}} = 0 \quad \text{if } V_x \geq V_T,$$

where  $V_T$  is the threshold voltage of the comparator. Then

$$V_x = B_i \frac{V_R}{2^N} - V_{IN} + V_T$$

due to comparator offset storage in the array. Hence,

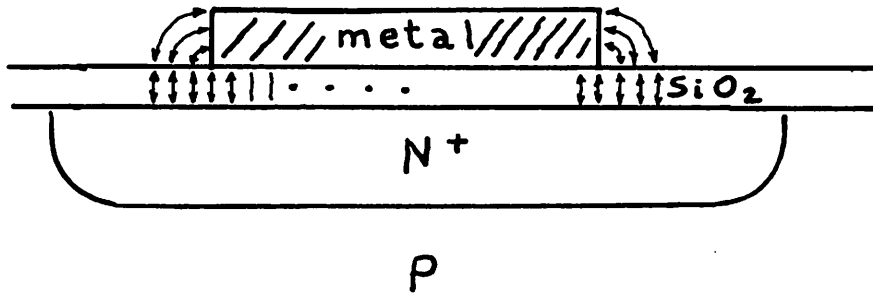


Figure 4.26: Electric field fringing effect.

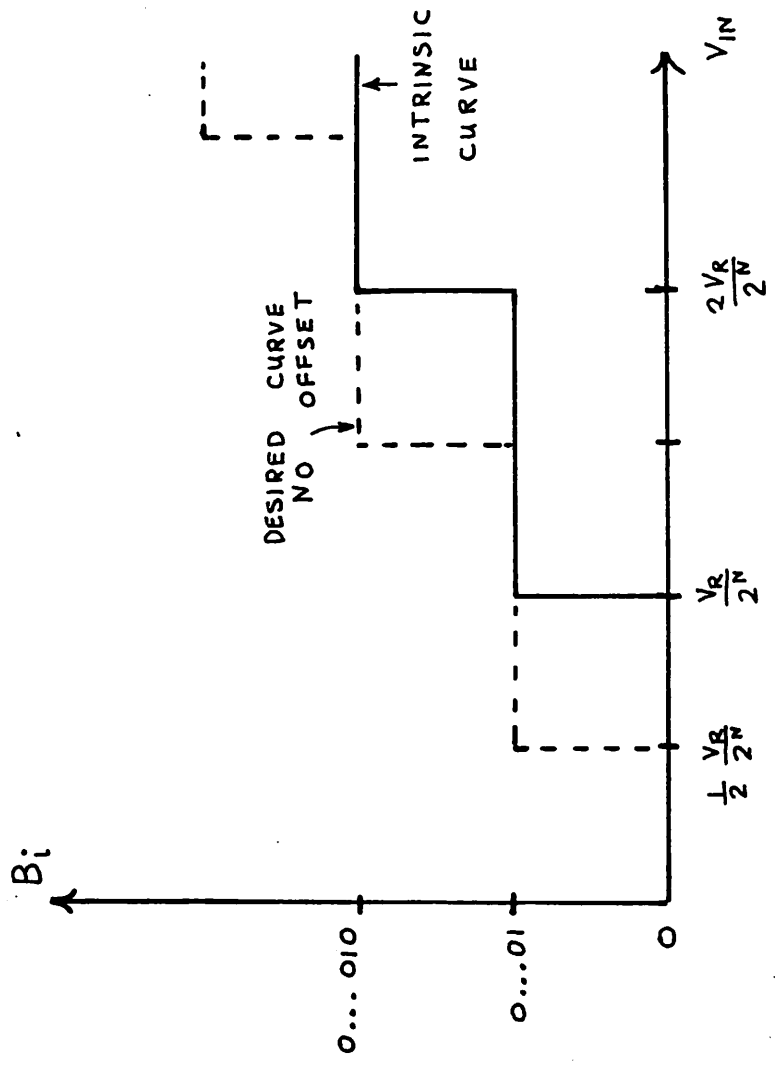


Figure 4.27: The  $\frac{1}{2}$  LSB offset error of the intrinsic transfer function.

$$V_{\text{out}} = 1 \quad \text{if} \quad V_{\text{IN}} > B_i \frac{V_R}{2^N}$$

and therefore  $B_i = 0$  until  $V_{\text{IN}} > \frac{V_R}{2^N}$  at which point the first transition occurs. In view of this, a  $-\frac{1}{2}$  LSB offset adjustment is required to give the "desired" curve which has its initial transition at  $V_{\text{IN}} = \frac{1}{2} \frac{V_R}{2^N}$ . This offset may be added to  $V_{\text{IN}}$  by level shifting the top plate by  $-\frac{1}{2} \frac{V_R}{2^N}$  after the sample mode and comparator offset cancellation are complete, but before the redistribution mode. This may be accomplished on-chip with the extra unity weight capacitor  $C_{1A}$  which already exists in the array as shown in Figure 4.28. The voltage  $V_{1A}$  on the lower plate of  $C_{1A}$  pulses from  $V_{\text{IN}}$  down to  $-\frac{V_R}{2}$  rather than to ground. The voltage  $-\frac{V_R}{2}$  need only be accurate to  $\pm 20\%$  for an offset cancellation accuracy of  $\pm .1$  LSB. Also this voltage is readily available since the substrate bias supply is nominally  $-5$  volts. This technique provides the desired initial transition at  $V_{\text{IN}} = \frac{1}{2} \frac{V_R}{2^N}$ . An implication of offset cancellation is that now the top plate voltage during the final configuration is  $V_x$  such that:

$$V_T \geq V_x = \frac{B_i V_R}{2^N} - V_{\text{IN}} - \frac{1}{2} \frac{V_R}{2^N} + V_T \geq V_T - 2\epsilon_q.$$

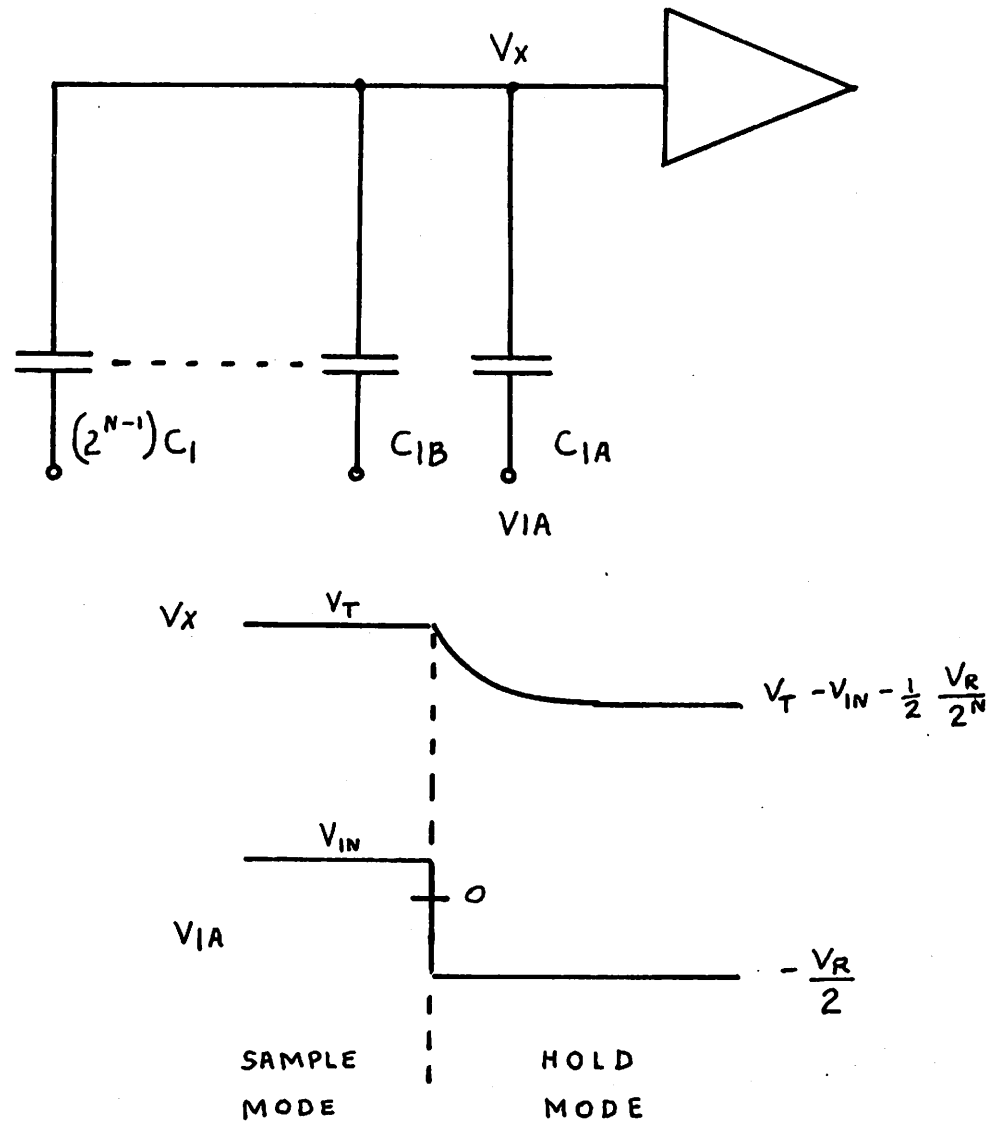


Figure 4.28: Intrinsic offset cancellation.



## CHAPTER V

Factors Limiting Conversion Rate in RADCAP5.1 Introduction

Compared with many conversion techniques the successive approximation method used in RADCAP is capable of rapid conversion. In this chapter the factors limiting the conversion rate are discussed from both theoretical and practical perspectives. In section 5.2 the acquisition time requirements for VATCAP when used as a sample/hold (S/H) circuit are examined. This analysis considers two criteria: the sampling accuracy specification and offset cancellation. This section also investigates the input bandwidth and distortion of high frequency signals. The minimum time required for one charge-redistribution cycle in VATCAP is discussed in section 5.3. An examination of the comparator delay time appears in section 5.4. The proper summation of all of these factors provides the maximum theoretical conversion rate as shown in section 5.5. However, there are practical limitations which dominate when conventional fabrication methods are used. These are discussed in section 5.6.

5.2 Factors Limiting the Minimum Acquisition Time when VATCAP is used as a S/H circuit5.2.1 Relationship Between Acquisition Time and Sampling Accuracy

VATCAP serves two special functions in addition to D/A conversion. It provides offset cancellation for the comparator input stage as explained in Chapter IV and it also performs a S/H function for the input signal. This function results in a sample mode precharge delay referred to as the acquisition time  $T_{aq}$ . The nature of this delay is illustrated in Figure 5.1. The VATCAP attenuator is shown with scaled bottom plate switches. The

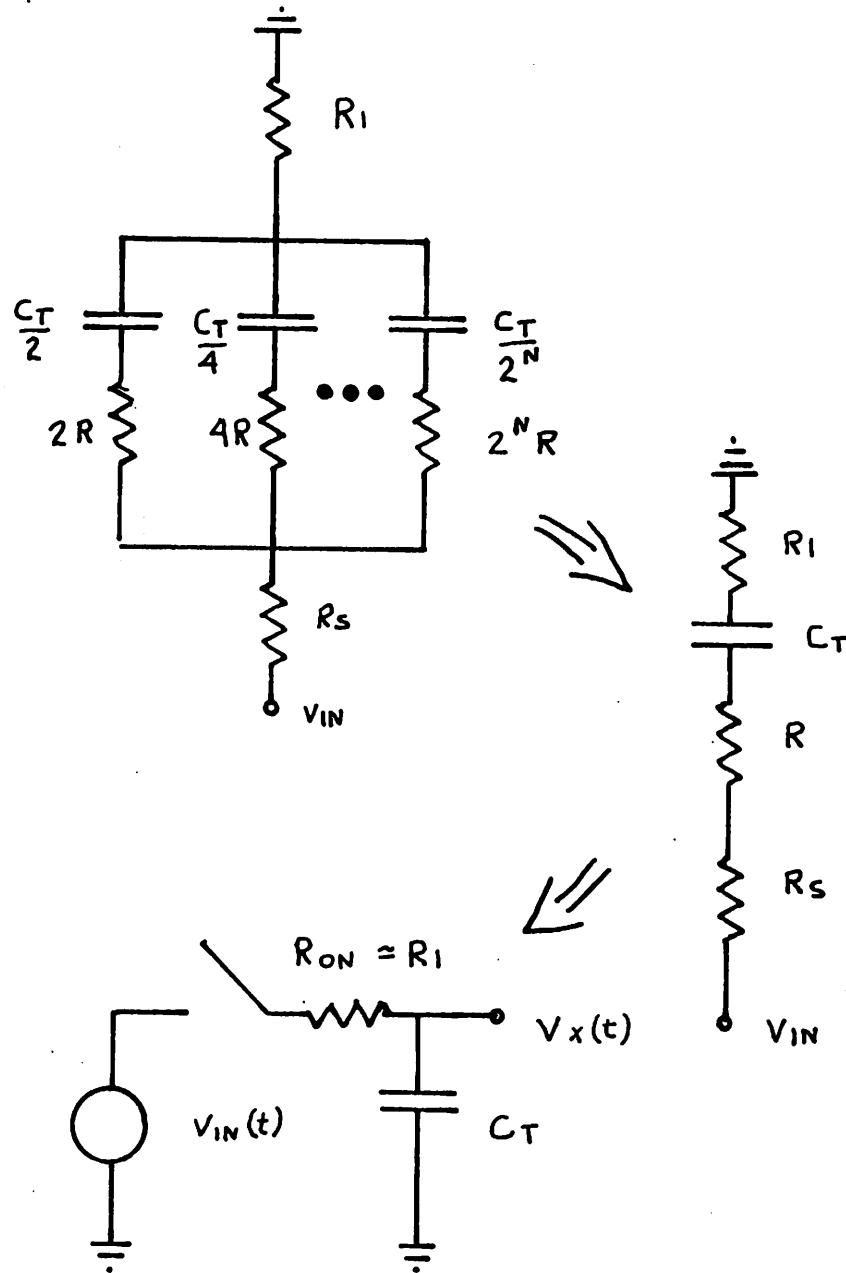


Figure 5.1: The equivalent circuit for VATCAP used as a sample-and-hold circuit during the sample mode precharge.

equivalent S/H circuit is reduced to a series RC circuit.  $R_{ON}$  represents the "ON" resistance of the MOS switches which will be approximately the same value as  $R_1$  the "ON" resistance of the grounding switch because  $R$  and  $R_s$  can be made arbitrarily small compared with  $R_1$ . This assertion is based upon the fact that large devices may be used for lower plate switching without affecting the accuracy of the conversion but the same is not true for the grounding switch. This will be clarified in section 5.2.2. Therefore the time constant during precharge is

$$\tau = [R_s + R + R_1]C_T \approx R_1 C_T.$$

The equation which describes  $V_x(t)$  is

$$V_x(t) = \mathcal{L}^{-1} \left[ \frac{V_{IN}(s)}{1 + \tau s} \right] \text{ where}$$

$\mathcal{L}^{-1}$  is the inverse Laplace transform and  $s$  is a complex number in frequency domain. For the case that  $V_{IN}(t)$  is a step input of amplitude  $V_R$  then:

$$V_x(t) = V_R \left( 1 - e^{-\frac{t}{\tau}} \right).$$

According to this solution the error voltage is  $V_E = -V_R e^{-\frac{t}{\tau}}$ . If the allowable error is  $\frac{V_R}{2^{N+1}}$  or  $\frac{1}{2}$  LSB then the minimum acquisition time is  $T_{aq} = (N+1)\tau \ln 2$ . The solution may also be determined for the case in which the input is a ramp with a d.c. offset voltage  $V_o$ :

$$V_{IN} = \frac{V_R}{T_{IN}} t + V_o \text{ where } \frac{V_R}{T_{IN}}$$

represents the input slew rate. For this case

$$V_x(t) = V_{IN}(t) - \frac{V_R}{T_{IN}} \tau (1 - e^{-\frac{t}{\tau}}) - V_o e^{-\frac{t}{\tau}}.$$

The error voltage is

$$V_E = -\frac{V_R}{T_{IN}} \tau (1 - e^{-\frac{t}{\tau}}) - V_o e^{-\frac{t}{\tau}}.$$

The initial value of  $V_E$  is  $-V_o$  at  $t = 0$  but at  $t = T_{aq} = (N+1)\tau \ln 2$  the error voltage converges to  $-\frac{V_R \tau}{T_{IN}}$ . Since this value is a constant,  $V_x(t)$  may be described as following  $V_{IN}(t)$  but attenuated in peak amplitude and delayed by a time  $\tau$ .

If the input were a sine wave of amplitude  $V_A$  and offset  $V_o$ :

$$V_{IN}(t) = V_A \sin \omega t + V_o,$$

then

$$V_x(t) = \frac{V_A \sin \omega t}{1 + \omega^2 \tau^2} + V_o - \frac{V_A \omega \tau (\cos \omega t - e^{-\frac{t}{\tau}})}{1 + \omega^2 \tau^2} - V_o e^{-\frac{t}{\tau}}.$$

In this example  $V_o$  is used for convenience to represent an initial difference voltage between  $V_x$  and  $V_{IN}$  although this could also have been done with a phase shift. The error voltage at  $t = 0$  is  $-V_o$ . At

$$t = T_{aq} = (N+1)\tau \ln 2$$

$$V_x(t) = \frac{V_A}{1 + \omega^2 \tau^2} [\sin \omega t - \omega \tau \cos \omega t] + V_o.$$

After performing a trigonometric combination  $V_x(t)$  becomes:

$$V_x(t) = \frac{V_A}{\sqrt{1 + \omega^2 \tau^2}} \sin(\omega t - \theta) + V_o$$

where  $\theta = \arctan \omega\tau$ . It is apparent therefore that the S/H function results in amplitude reduction by a factor  $\frac{1}{\sqrt{1 + \omega^2\tau^2}}$  and a phase shift of  $-\theta$  for high frequency signals. The S/H behaves like a low pass filter having a bandwidth  $\frac{1}{\tau}$ . For example if  $\omega = \frac{1}{\tau}$  then the signal amplitude of  $V_x(t)$  is reduced to  $\frac{V_A}{\sqrt{2}}$  and the phase shift is  $-45^\circ$ .

In conclusion, for all three cases considered, the sampled signal  $V_x(t)$  remains undistorted relative to quantization distortion provided that  $T_{aq} \geq (N+1)\tau \ln 2$ . The minimum acquisition time is therefore proportional to the minimum value of  $\tau$ .

#### 5.2.2 Minimum Acquisition Time with Offset Cancellation Technique

It has just been determined that  $T_{aq}$  is proportional to  $\tau$ . It is therefore desirable to determine the minimum value of  $\tau$ . It will be shown that for the RADCAP technique this depends upon the extent to which feedthrough effects can be cancelled. The offset cancellation scheme described in section 4.2 and Figure 4.3 cancels the additional offset error  $V_{FT}$  introduced by the capacitive feedthrough of switch S1. However there are limitations on the magnitude of  $V_{FT}$  since the stages A1 and A2 in Figure 4.4 must remain biased in a linear gain region. This technique will be described in detail in Chapter VI but at this point it is sufficient to model the circuit as shown in Figure 5.2. From this figure  $\tau = R_1 C_T$  as before but the "ON" resistance of switch S1 is given by

$$R_1 = \frac{1}{\frac{W}{L_c} \mu C_{ox} (\Delta V)}$$

where  $\Delta V = V_{OS}(ON) - V_T$  and  $\frac{W}{L_c}$  is the channel width to length ratio [34].

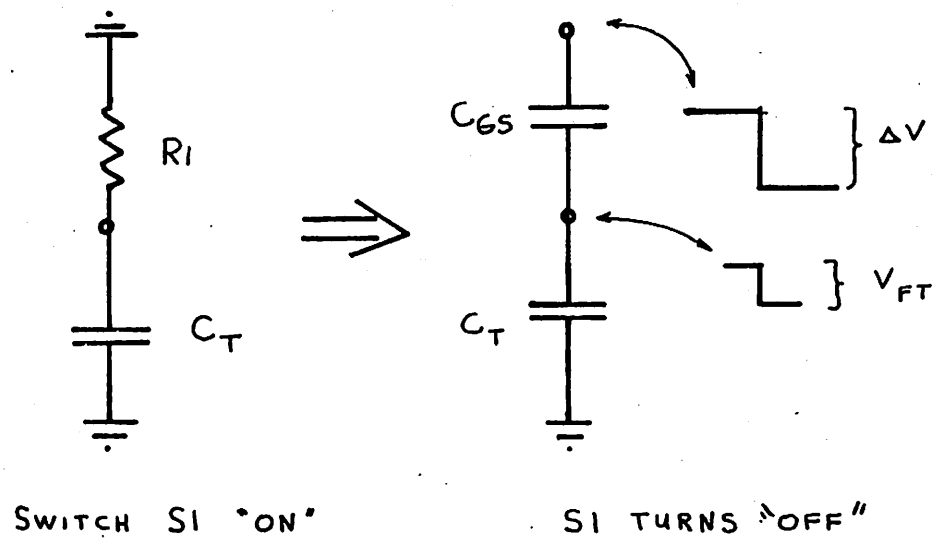


Figure 5.2: An illustration of the capacitive feedthrough of switch S1.

$C_{ox}$  is channel capacitance per unit area of the thin oxide and  $\mu$  is the effective electron mobility for N-MOS devices. Then the total array capacitance  $C_T = A_T C_{ox}$  where  $A_T$  is the total top plate area. Therefore

$$\tau = \frac{A_T}{\frac{W}{L_c} \mu \Delta V} .$$

The effect of capacitive feedthrough when S1 turns off is a voltage drop at the comparator input of:

$$V_{FT} \approx \frac{C_{GS} \Delta V}{C_T} \quad \text{where}$$

$C_{GS}$  is the gate-source capacitance of S1 and is assumed to be much less than  $C_T$ . Using  $C_{GS}$  approximately equal to  $C_{ox} W L_c$  then the product:

$$V_{FT} \tau = \frac{L_c^2}{\mu} \quad \text{where}$$

$L_c$  is the minimum channel length for the given supply voltage to avoid drain-source breakdown. Therefore a trade-off exists between  $V_{FT}$  and  $\tau$ . The smallest value of  $\tau$  is dependent upon the largest value of  $V_{FT}$  which may be cancelled by circuit techniques. Furthermore this trade-off prevents  $R_1$  (the resistance of S1) from being as small as  $R$  or  $R_s$  since the capacitances of these switches do not cause errors. Hence the minimum value of  $\tau$  is

$$\tau = \frac{L_c^2}{\mu V_{FT}}$$

and therefore the acquisition time is

$$T_{aq} = \frac{(N+1)}{V_{FT}} \frac{L_c^2 \ln 2}{\mu}$$

The absolute minimum acquisition time will be estimated for the theoretical limit in which  $V_{FT} = V_{DD} \approx 15$  V and  $L_c = 5 \mu$ ,  $N = 10$  bits, and  $\mu = 500 \frac{\text{cm}^2}{\text{V} \cdot \text{s}}$ . Then for this example  $T_{aq} = 0.25$  ns.

### 5.3 Factors Limiting the Minimum C/R Time for RADCAP Class of Circuits

The equivalent small signal model for VATCAP during C/R is illustrated in Figure 5.3 from which the circuit time constant is

$$\tau_{CR} = R(C_D + C_T).$$

$R$  represents the total resistance of all bottom plate switches in parallel as before and  $C_D$  is the total drain to substrate capacitance of all switches in parallel. An equation for  $R$  is

$$R = \frac{1}{\mu \frac{W}{L_c} C_{ox} \Delta V}$$

as defined in section 5.2. The minimum value of drain capacitance may be expressed as  $C_D = C_{pn} W L_D$  where  $C_{pn}$  is the capacitance per unit area of the drain pn junction and  $L_D$  is the minimum length of the drain diffusion.

Therefore the resultant expression for  $\tau_{CR}$  is

$$\tau_{CR} = \left( \frac{A_T}{A_c} + \frac{C_{pn} L_D}{C_{ox} L_c} \right) \left( \frac{L_c^2}{\mu \Delta V} \right),$$

where  $A_c$  is the total channel area of all switches,  $A_c = W L_c$ . The first term dominates under normal conditions since  $A_T \gg A_c$  but  $C_{pn} L_D$  and  $C_{ox} L_c$



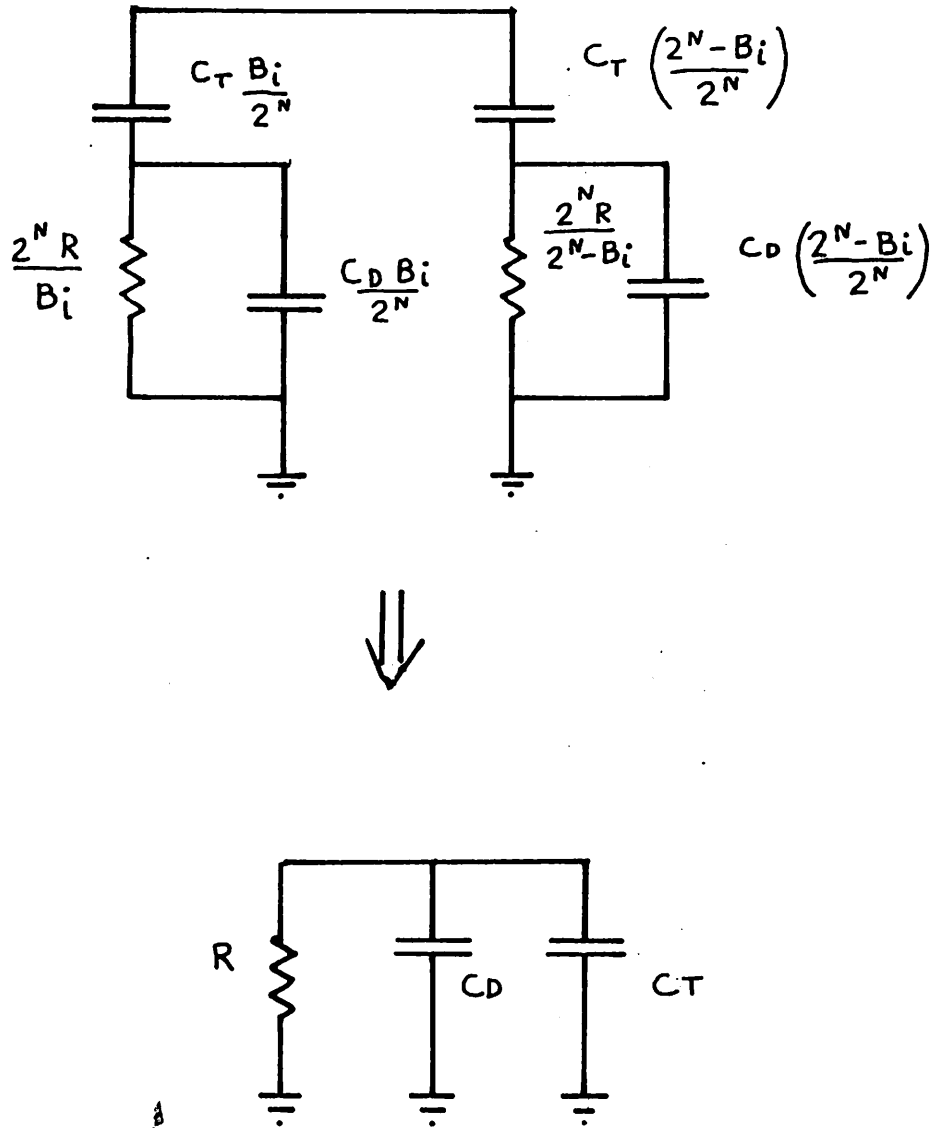


Figure 5.3: The equivalent circuit for VATCAP during the charge-redistribution mode.

may be of the same order of magnitude. In the theoretical limit that the capacitor array could be made small compared with the device size  $\tau_{CR}$  approaches  $\frac{C_{pn}}{C_{ox}} \frac{L_D L_C}{\mu \Delta V}$ . In effect the switches must charge their own capacitances. The total time required for  $N$  C/R cycles, each requiring  $(N+1)\ln 2$  time constants to go to completion, is  $T_{CR} = N(N+1)\tau_{CR} \ln 2$ .

#### 5.4 Factors Causing Comparator Delay

For  $N$  bits of resolution,  $N$  comparisons must be performed by the comparator. It is of interest to examine the fundamental limitations resulting in comparator delay. The comparator must have sufficient gain to resolve the minimum input signal which is  $\frac{V_R}{2^N}$ . The output voltage swing must be approximately  $V_{DD} - 2V_T$  since this must be compatible with the digital logic levels. Therefore the approximate minimum comparator gain required is

$$A = 2^N \frac{(V_{DD} - 2V_T)}{V_R} .$$

Assume for a moment that this gain is realized by direct linear amplification. If slew rate limiting is not a problem then an optimistic transfer function describing this comparator is:

$$A(s) = \frac{A}{1 + s \tau_o}$$

in which  $\tau_o = \frac{A}{2\pi f_T}$  and  $f_T$  is the unity-current-gain frequency of the devices. Hence the time required for  $N$  comparisons each settling to within  $\frac{100}{2^{N+1}}$  % of final value is

$$T_{COMP} = \frac{N(N+1) 2^N \ln 2 (V_{DD} - 2V_T)}{2\pi f_T V_R} \approx \frac{N^2 2^N \ln 2}{2\pi f_T}$$

for  $V_{DD} - 2V_T \approx V_R$ .

From this equation the gain-bandwidth product limitations involved in direct amplification cause a severe comparator delay.

Consider a different approach in which the comparator function is performed by a bistable latch having positive feedback as shown in Figure 5.4 [35]. The gain of the basic amplifier is expressed in terms of a single time constant as before:

$$A(s) = \frac{A}{1 + s\tau}$$

The total transfer function of the feedback amplifier is

$$\frac{V_2(s)}{V_1(s)} = \frac{-A(s)}{1 - (-A(s))^2}$$

for which the approximate time domain solution is:

$$V_2(t) = \frac{\Delta V_L}{2} e^{\frac{t(A-1)}{\tau}}$$

In this expression  $\Delta V_L$  is the initial voltage difference in the latch.

Evaluating the delay time for the full output voltage swing,  $V_2(T_{LATCH}) = V_{DD} - 2V_T$ , then

$$T_{LATCH} = \frac{\tau}{A-1} \ln \left[ \frac{2(V_{DD} - 2V_T)}{\Delta V_L} \right]$$

From this expression the delay time is inversely proportional to  $(A-1)$  hence a large value of  $A$  is desirable. In fact, it would be desirable to realize the full gain of the comparator with the latch circuit due to the speed advantages of positive feedback over linear amplification.

If this were done, the latch would have to switch properly for a minimum differential input signal of  $\Delta V_L = \frac{V_R}{2^N}$  in order that 1 LSB be resolved.

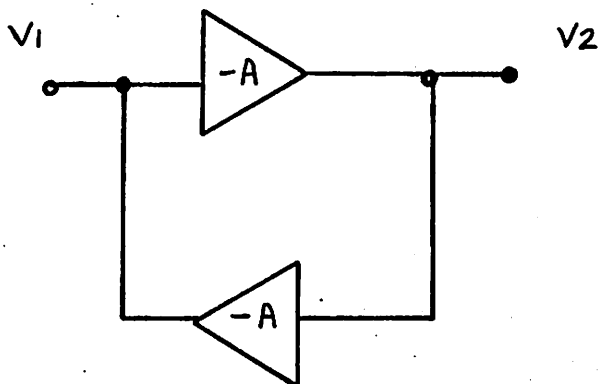


Figure 5.4: A bistable regenerative latch.

For simplicity  $V_{DD} - 2V_T$  is taken to be approximately the value of  $V_R$ ,

$$\text{and } \frac{\tau}{A-1} \approx \frac{1}{2\pi f_T}$$

$$T_{LATCH} \approx \frac{\tau}{A-1} (N+1) \ln 2 \approx \frac{(N+1) \ln 2}{2\pi f_T}$$

Then  $T_{COMP} = N T_{LATCH} \approx \frac{N^2 \ln 2}{2\pi f_T}$ . This quantity is compared with total comparator switching time on page 124. It is found that the regenerative latch switches approximately  $2^N$  times faster than the amplifier. This result implies that ideally the comparator should consist of a latch and that linear amplification is to be avoided since it is a slower process than regenerative latching.

It is instructive to consider the theoretical switching time of the latch by expressing  $\tau$  in terms of device and circuit parameters. By invoking an approach similar to that in section 5.3 the dominant time constant is modeled by a junction capacitance and thin oxide capacitance in parallel with the "ON" resistance of a large device. This analysis will assume for convenience that the amplifier having gain  $-A$  is a simple MOS active load inverter, hence, for a given  $V_{GS}(ON)$  in the load:

$$\tau = C_L \times R_{ON} = \frac{W(L_D C_{pn} + L_C C_{ox}) A^2}{A^2 \mu \frac{W}{L_C} C_{ox} (V_{GS}(ON) - V_T)}$$

Taking an average value of  $\frac{V_{DD} - 2V_T}{2}$  for  $(V_{GS}(ON) - V_T)$  the following expression is obtained:

$$T_{COMP} = \frac{N(N+1)}{A-1} (\ln 2) \frac{L_C^2}{\mu} \left( 1 + \frac{L_D C_{pn}}{L_C C_{ox}} \right) \frac{2}{V_{DD} - 2V_T}$$

It is reiterated that this is a fundamental limitation rather than the nominal delay of a real circuit.

In reality the asymmetry and mismatches in the latch will reduce its ability to resolve small signals. Therefore the smallest practical input signal is determined by the effective input offset voltage of the latch,  $V_{INOS}$ . Hence a linear amplification stage having a minimum gain of approximately  $\frac{V_{INOS}}{V_R} 2^{N+1}$  must precede the latch in order to cancel the offset

reflected to the input.

The results in this section support the conclusion that comparator delay will be largely determined by the amount of linear amplification required prior to the latch. Some reduction in delay can be realized by having this linear stage directly coupled to the comparator input during the C/R cycle. In this way the C/R delay and linear amplification delay may be nearly coincident. In retrospect the comparator having a regenerative latch may be modeled with infinite gain and an input offset voltage as asserted in Chapter IV.

### 5.5 The Theoretical Minimum Conversion Time

The theoretical minimum conversion time will be estimated on the basis that logic and aperture delays could conceivably be made small. In this case the total conversion time is given by:

$$\begin{aligned}
 T_c &= T_{aq} + T_{CR} + T_{COMP} \\
 &= \frac{N+1}{V_{FT}} \frac{L_c^2 \ln 2}{\mu} + \frac{N(N+1)}{\Delta V} \frac{L_c^2 \ln 2}{\mu} \left( \frac{A_T}{A_c} + \frac{C_{pn} L_D}{C_{ox} L_c} \right) \\
 &\quad + \frac{N(N+1)L_c^2 \ln 2}{(A-1)\mu} \left( 1 + \frac{C_{pn} L_D}{C_{ox} L_c} \right) \left( \frac{2}{V_{DD} - 2V_T} \right)
 \end{aligned}$$

The ultimate limit will now be computed with the following assumptions:

$$\Delta V = V_{FT} = V_{DD}; A = 2;$$

$$C_{pn} L_D = C_{ox} L_c;$$

and that  $A_T < A_c$

$$\text{Thus } T_c \approx \frac{(N+1)L_c^2 \ln 2}{V_{DD} \mu} (5N + 1)$$

Using  $L_c = 5 \mu$ ,  $N = 10$  bits,  $V_{DD} = 15V$  and  $\mu = 500 \frac{\text{cm}^2}{V\text{-s}}$  the ultimate limit is approximately:

$$T_c = 22 \text{ ns.}$$

This result indicates that theoretically the A/D conversion time may be quite small. Two important factors which influence this hypothetical limit are feedthrough cancellation and the charging of capacitance. The practical limits are considered in the next section.

### 5.6 Practical Limitations on Conversion Rate

Some practical considerations which limit conversion rate will now be discussed. The inspection of the time constants  $\tau$  and  $\tau_{CR}$  from sections 5.2 and 5.3 indicates that the time constants are directly proportional to the switch "ON" resistance and to the total capacitance  $C_T$ . Therefore increased speed may be achieved by larger switches (over some practical range); however, the switches may not be so large that the maximum instantaneous current is excessive. Additional considerations are therefore power dissipation and chip area. A smaller total capacitance also reduces the time constants but the minimum total capacitance is limited by the resolution properties of the standard photomasking process rather than by the parasitic capacitance. The limitations of the photolithography are manifested as undercut error, corner-rounding, and random edge variations which combine statistically to produce a distribution of errors in total capacitor area with standard deviation  $\sigma_A$ . The minimum value of  $C_T$  is proportional to the minimum total area  $A_T$  such that  $\frac{\sigma_A}{A_T} < \frac{1}{2^{N+1}}$  for  $N$  bits of resolution. This design constraint reflects the fact that higher yields of accurately ratioed arrays will result if the uncertainty in capacitor areas is small compared with the area of the unity weight capacitor.

## CHAPTER VI

Description of an Experimental ADC6.1 Introduction

The techniques described thus far should allow the fabrication of a single-chip MOS ADC. However only the critical portion of the circuit, the capacitor array and the comparator have been fabricated in order to demonstrate the feasibility of the RADCAP technique. The remaining element is the digital control circuitry, the MOS implementation of which is straightforward. The block diagram of Figure 6.1 shows a complete ADC and those components which were fabricated as part of the experimental I.C. The digital circuitry is composed of TTL gates. In this chapter the designs of the experimental chip and the logic system are discussed.

6.2 Optimization of MOS Capacitor Geometry

The realization of a high resolution ADC of the RADCAP type requires the fabrication of precisely matched capacitors. The implications of Chapter IV suggest that the key elements of this realization are multiple plates, common centroid geometry, corner compensation and etch rate control. Another requirement is mask misalignment tolerance. Finally the capacitor itself should depend upon as few variables as possible in order to reduce uncertainties.

Most of the design criteria can be achieved with conventional photolithography with the MOS capacitor structure shown in Figure 6.2. The capacitor top plate is aluminum and its area on top of the oxide covering the  $N^+$  defines the capacitor. The heavily doped  $N^+$  diffusion acts as the bottom plate. Most of the capacitance value is due to the thin oxide, however, a small fraction of the capacitor value is due to interconnect



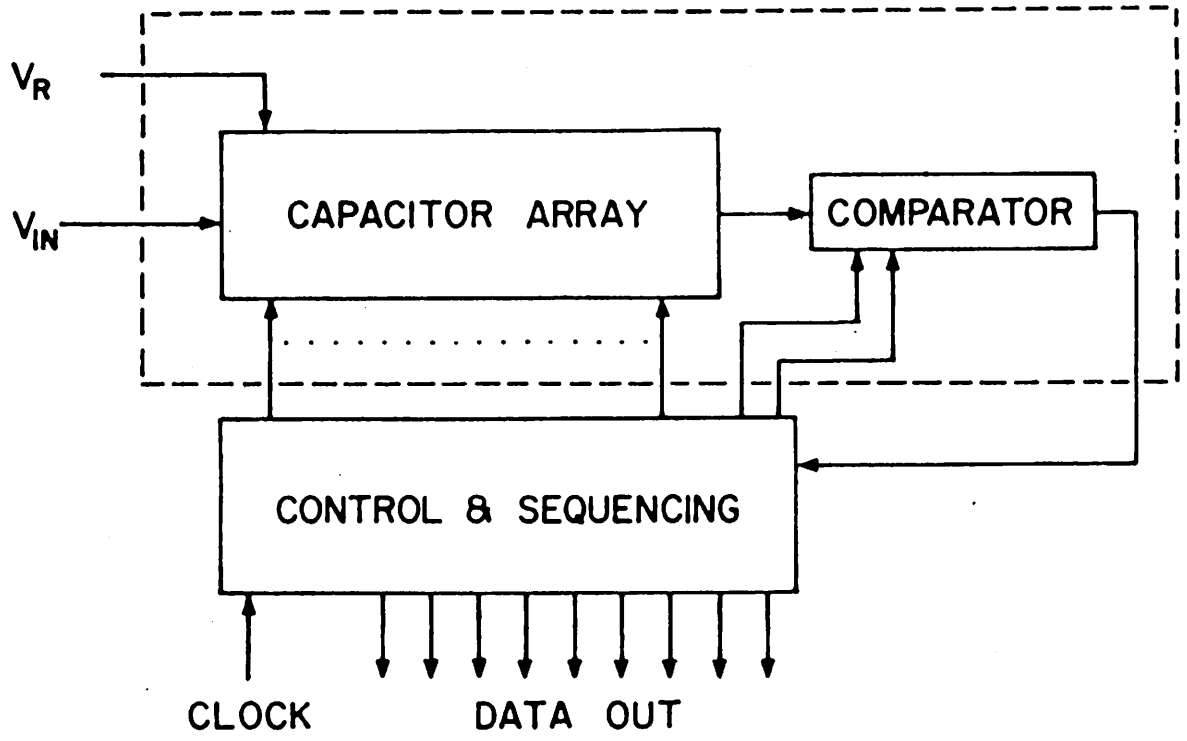


Figure 6.1: A complete ADC. The experimental I.C. is defined by the dashed lines.

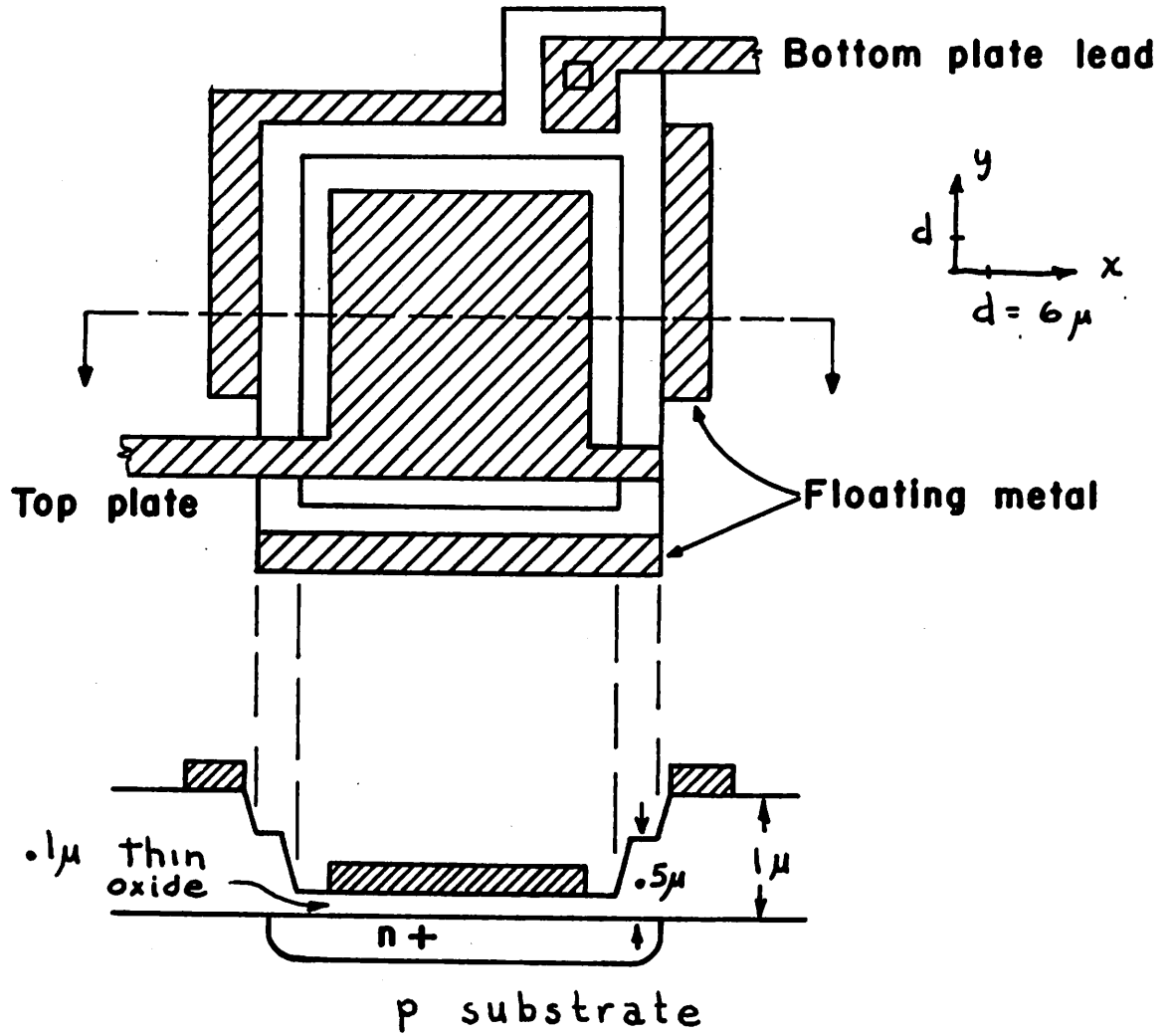


Figure 6.2: MOS precision capacitor structure.

passing over the thicker oxide above the  $N^+$  region. The interconnect over the field oxide constitutes a parasitic capacitance to ground but does not add to the design value of the MOS capacitor. The thin oxide is nominally chosen as  $1200 \text{ \AA}$  since this is the thickest gate oxide desirable. The field oxide is chosen to be  $1 \mu$  since this value results in a field device threshold voltage in excess of 20 volts which is desirable. The  $0.5 \mu$  oxide over the  $N^+$  is arbitrarily picked to provide an intermediate step between oxides but should be thick enough to minimize overlap capacitance. On the other hand this oxide must not be so thick that the effective  $N^+$  drive-in during oxide growth significantly reduces the  $N^+$  surface concentration since this will increase the voltage coefficient of capacitance. The alignment tab shown as an extension of the capacitor plate makes the capacitor value independent of misalignment error  $d$  in the X-axis while no error results from a Y-axis misalignment of  $\pm d$ . The floating metal strip helps to maintain a uniform undercut during the aluminum etch. The actual effect of the strip upon the circuit will be to add parasitic capacitance at the bottom plate which was discussed in Chapter IV. The circuit schematic for the capacitor array is shown in Figure 6.3.

### 6.3 MOS Comparator Realization

The MOS comparator is shown in Figure 6.3 along with the capacitor array. The device aspect ratio,  $\frac{W}{L}$  (channel width divided by channel length), is given for each transistor. The basic operation of the comparator has been outlined in Chapter IV. It consists of one precharge cycle during the sample mode, a hold mode, and ten tests during the redistribution mode.

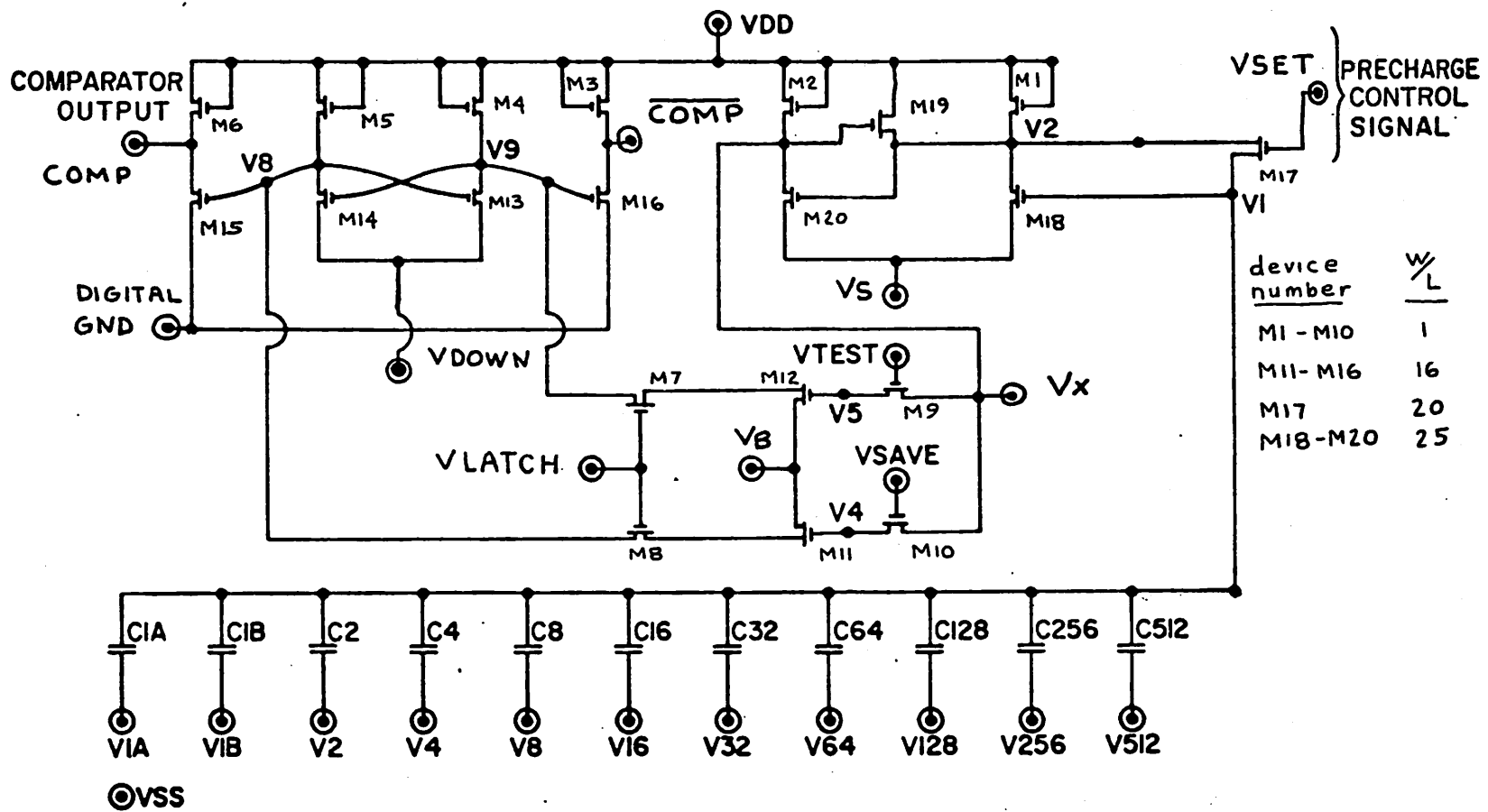


Figure 6.3: A circuit schematic of the experimental I.C.

The sample mode precharge cycle is initiated by a 15 volt VSET signal on the gate of M17.  $V_S$  is a supply voltage nominally chosen to be zero volts. In the worst-case precharge  $V_1$  is negative and  $V_2$  is also low because M17 is in the triode region of operation. This forces M18 off. Since M18-M1 and M20-M2 comprise two gain stages A1 and A2 respectively M20 must also be off because  $V_2$  is low. Therefore the output of A2,  $V_x$ , must be high which turns M19 heavily "ON". M19 is a feedback switch that shunts the output of stage A1 thereby reducing its effective output resistance. This causes a rapid precharge of the array since both M19 and M17 are large devices. When the array charges up positively and both  $V_1$  and  $V_2$  become larger,  $V_x$  will decrease as  $V_2$  increases above the nominal 2 volt threshold voltage of M20. M19 will eventually turn off when  $V_2$  is about 0.5 volt less than the "switching threshold" voltage of stage A2 (that value of voltage for a digital circuit for which  $V_{IN} = V_{out}$  or  $V_2 = V_x$  in this case). However, when this occurs  $V_1$  is nearly equal to  $V_2$  since M17 is heavily "ON" in the "non-saturated" or "triode" region of operation. Therefore a d.c. steady state solution for A1 and A2 is:

$$V_1 = V_2 = V_x = V_{balance}$$

M19 will remain off until  $V_x$  increases by about 2 volts above  $V_2$ . Hence M19 does not destroy the small signal gain about the balance point,  $V_{balance}$ . In fact M19 actually performs an additional useful function by clamping the maximum value of  $V_x$  perhaps reducing the effects of capacitive coupling between the high gain stages and the rest of the circuit. After approximately 1  $\mu$ s or 2  $\mu$ s of precharge the steady state condition is reached and the sample mode precharge is complete.

Both VSET and VSAVE pulse down turning off M17 and M10 which were

heavily "ON" at the completion of the precharge cycle. This initiates the hold mode and it is characterized by the storage of  $V_{\text{balance}}$  at node V4. After the switching transients settle  $V_x$  will be significantly less than  $V_{\text{balance}}$  due to the capacitive feedthrough effects of VSET upon V1:  $V_x = \left( V_{\text{balance}} - V_{\text{FT}}^{A1 A2} \right)$  where A1 and A2 are the gains of stages A1 and A2. This is an apparent offset at V1 and may be cancelled simultaneously along with the intrinsic offset cancellation. The desired value of V1 is:

$$V1 = V_{\text{balance}} - \frac{1}{2} \frac{V_R}{2^N}$$

but after feedthrough of VSET:

$$V1 = V_{\text{balance}} - V_{\text{FT}}$$

hence V1 must be increased by a constant value:  $\Delta V1 = V_{\text{FT}} - \frac{1}{2} \frac{V_R}{2^N}$ . Since this value is small it may be added to V1 by an additional capacitor  $C_Z$ , in the array which is dedicated strictly to offset and feedthrough cancellation. The voltage applied to the bottom plate of  $C_Z$  may be a nominal 10 volt transition, hence the nominal value of  $C_Z$  for a 10 volt reference is:

$$C_Z = C_T \left( \frac{V_{\text{FT}} - \frac{1}{2} \frac{V_R}{2^N}}{10 - V_{\text{FT}} - \frac{1}{2} \frac{V_R}{2^N}} \right).$$

The offset and feedthrough cancellation scheme just described is illustrated in Figure 6.4. It is not necessary for V1 to go to  $-V_{\text{IN}}$ , as previously conceptualized in Chapter 3, unless the converter were operating to accept bipolar voltage inputs. However, for positive input voltages a test upon  $-V_{\text{IN}}$  yields no information. Hence the hold mode ends with  $V1 \approx V_{\text{balance}}$ .

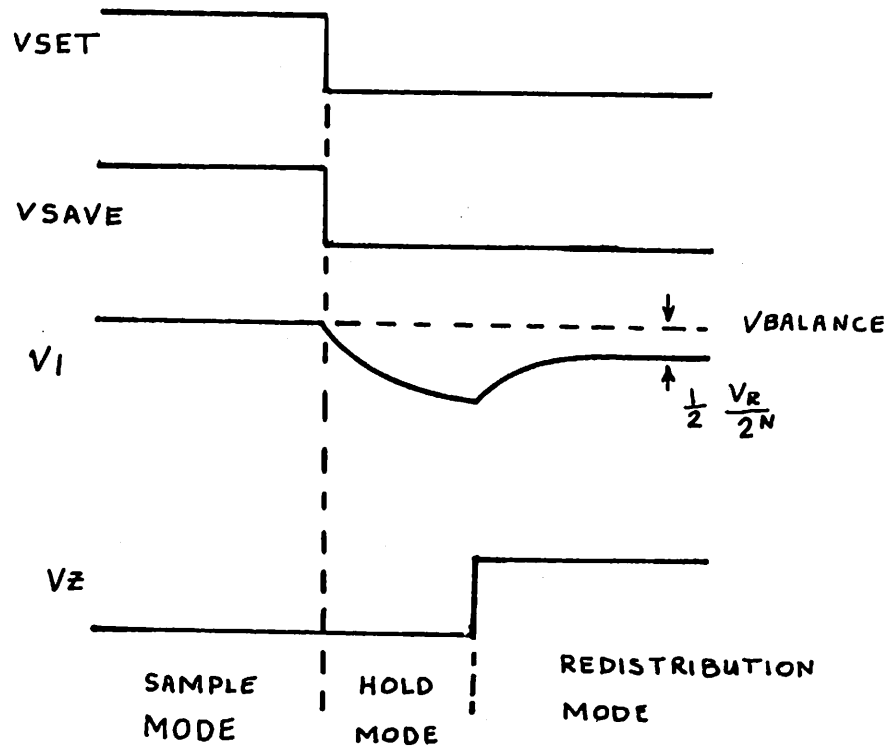
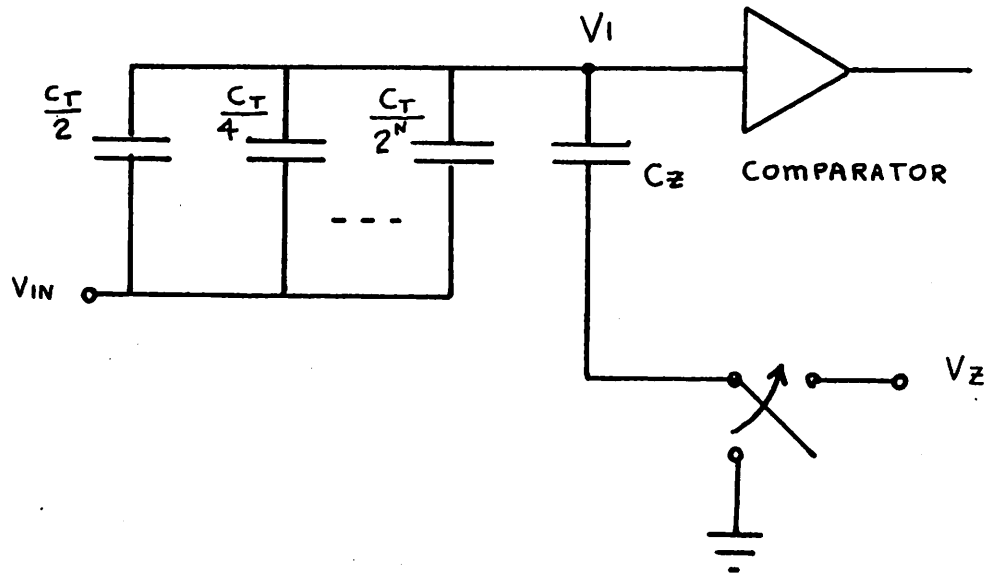


Figure 6.4: Cancellation of intrinsic offset and feedthrough.

The redistribution mode begins with the simultaneous occurrence of 3 events: intrinsic offset cancellation as just described; the switching of the bottom plate of the largest capacitor V512 to  $V_R$ ; and the switching of all other capacitors to ground. In this way the charge-redistribution necessary to test the MSB is performed initially. After this  $V_x$  settles to  $V_x''$  - some amplified value of  $V_1$ . This value is stored at V5 by a pulse on the gate of M9. M9 turns off in order to simulate the same feed-through effects at V5 as M10 had upon V4. The difference signal (V5-V4) is proportional to  $[V_x'' - V_x']$  where  $V_x' = \left( V_{\text{balance}} - \frac{1}{2} \frac{V_R}{2^N} A_1 A_2 \right)$ . During this phase of the redistribution cycle  $V_{\text{DOWN}}$  and  $V_{\text{LATCH}}$  are both high making M13 and M14 "off" but M7 and M8 "ON".  $V_B$  is a fixed d.c. voltage adjusted so that V4 biases M11 in the saturated region. During this time M5 and M4 act as saturated loads for M11 and M12 respectively. Hence these 4 devices comprise a third stage A3 having gain A3. However, this stage is actually a difference amplifier since  $(V_8-V_9) = A_3(V_5-V_4)$ . In contrast stages A1 and A2 form a single-input and single-output amplifier with gain  $A_1 \times A_2$ . The inputs to A1 - A2 are time-multiplexed. This has the great advantage that offsets or mismatches in these two stages do not adversely affect the comparator since these affects are common to both V4 and V5. Only the polarity of  $(V_x'' - V_x')$  is of importance. The signal at V8, V9 is:

$$\begin{aligned} (V_8-V_9) &= G_{A1} \times G_{A2} \times G_{A3} (V_x'' - V_x') \\ &\approx 200 (V_x'' - V_x'). \end{aligned}$$

Since the minimum signal is  $\frac{V_R}{2^N} = 10 \text{ mV}$ , the minimum value of  $(V_8-V_9)$  is 2 volts. M4, M5, M13 and M14 form a bistable latch circuit. In the next



clock cycle  $V_{\text{DOWN}}$  goes low and M14 and M13 turn "ON" and the latch partially regenerates. That is, (V8-V9) is increased in value by this operation. The regeneration becomes total on the next clock cycle when  $V_{\text{LATCH}}$  also goes low turning off the transmission gates M7 and M8. The latch ends in a d.c. state in which (V8-V9) is approximately equal to  $(V_{\text{DD}} - V_{\text{threshold}})$  the full available logic swing. The gain of the entire comparator is therefore infinite. The outputs COMP and  $\overline{\text{COMP}}$  are buffered out of the comparator and into the external digital system. The redistribution mode continues with similar tests until all 10 bits have been tested. The A/D conversion is then complete.

#### 6.4 Logic Circuit Design

The logic circuit performs sequencing, control and data storage functions that are necessary to support the experimental chip as a complete ADC. A detailed system diagram illustrating the functional blocks of the logic circuit is shown in Figure B.1 of Appendix B. A state sequencer containing registers and a counter drives the "Capacitor Signal Generator" and the "Switch Signal Generator." The signals destined for the experimental I.C. require level shifting and buffering to convert them from TTL to MOS logic levels. This function is performed for the capacitor signals by the "CMOS Switches". MOS switches were not placed on chip because the number of bonding pads needed for gate signals would have been excessive. The "Switch Signal Generator" provides the timing signals for the comparator. The final configuration of capacitor signals is clocked into an output buffer and then may be channeled to a suitable display. The details of the logic system including the timing diagrams, state table, and circuit schematics are shown in Appendix B.

## 6.5 Summary

The design philosophy for both the experimental I.C. and the digital logic circuit was that a reasonable effort should be devoted to design flexibility. That is, the chip design should permit numerous methods of recovery in the case of partial circuit failure. This is evident by the number of d.c. levels that are externally adjustable off chip, and the optional outputs and internal bonding pads. In addition the logic system is designed with adjustable width timing signals. Although this philosophy adds some complexity it further enhances the capability for experimental evaluation of the new technique as well as aiding the investigation of errors.

## CHAPTER 7

EXPERIMENTAL RESULTS7.1 Introduction

The experimental results are discussed in 4 sections of this chapter. In section 7.2 the measured data taken for the first experimental I.C., IC1, is examined and the largest sources of error are identified. The subsequent design modifications in both the fabrication schedule and the layout that are required to correct the error are described in sections 7.3 and 7.4. The results of measurements taken from the second experimental IC, IC2, are analyzed in section 7.5. Both IC1 and IC2 have the same circuit schematic as illustrated in Figure 6.3 hence both are 10-bit RADCAP type of ADCs. However, they do not have the same circuit layout or topological geometric configuration. N-channel aluminum gate technology was chosen over p-channel technology due to the higher surface concentration of diffusant resulting in a lower voltage coefficient of capacitance and also due to the higher mobility of electrons over holes. The N-MOS metal gate fabrication schedule is given in Appendix C [36].

7.2 Experimental Results of IC17.2.1 Design of Circuit Layout for IC1

The layout of an integrated circuit is the plane geometrical configuration of the topology of various regions by which a circuit is realized. For IC1 the layout was designed so that  $0.75 \mu$  undercut could be tolerated. This requires the reproduction and parallel connection of smaller plates of identical geometry as discussed in section 4.9.3. In Appendix D an equation is derived for nonlinearity referenced to 10-bit resolution as a function of capacitor reproduction size and uniform

undercut. This equation is plotted in Figure D.1. From this analysis for a VATCAP array it is concluded that a duplication size of 32 will retain 10-bit ratio accuracy for uniform undercut up to  $0.75 \mu$ . If, for example, capacitors in VATCAP are designated C512, C256, C128, ... C2, C13 from largest to smallest, then a duplication of size 32 means that the largest single square capacitor plate has the same dimensions as C32 and that C64 is composed of 2 plates exactly identical in size to C32 but connected in parallel. Similarly, C512 is the parallel connection of 16 plates each identical to C32 in dimensions. This may be seen in the die photo of IC1 which is shown in Figure 7.1. Another layout design feature of IC1 includes common centroid geometry but only for the largest capacitor C512. This may also be seen in the die photo. The absence of available data on oxide growth uniformity of the particular furnace involved in capacitor oxide growth led to an initial assumption that over the region of 1 die the uniformity would be sufficiently good that common centroid geometry is needed only for the largest capacitor. Other sources of error that could not be accurately evaluated prior to wafer fabrication were not included in design considerations. The die photo shown in Figure 7.1 is about 70 mils square. The comparator is located below the capacitor array and next to the test devices. The layout for IC1 is realized on the silicon wafer with glass-emulsion photomasks containing a 400 times reduction of rubilith artwork. This large reduction size was chosen to minimize the effect of linear dimensional uncertainty involved in cutting the rubilith.

### 7.2.2 Threshold Voltage for the N-MOS Device in IC1

The N-MOS devices require a positive threshold voltage  $V_{TH}$  for proper

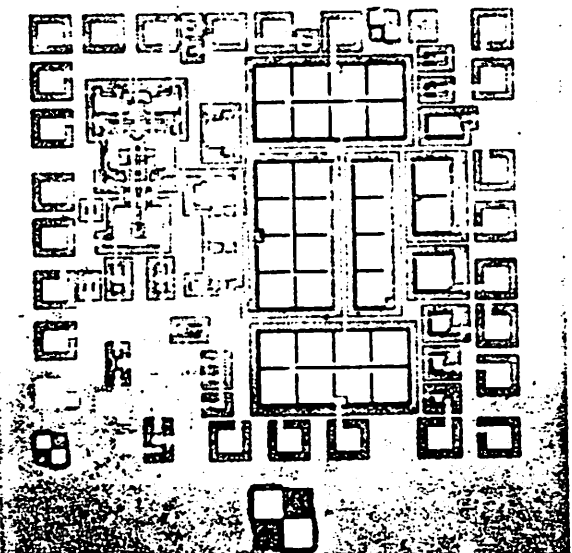


Figure 7.1: Die photo of IC1.

on/off switching action. A nominal threshold voltage of 2v is desirable because the voltage difference at the latch (V8-V9) can be no larger than  $V_{TH}$  from section 6.3, and 2V is a desired value for (V8-V9). There are no other particular device requirements although a large conduction coefficient or gain parameter is advantageous for faster switching. Before sintering the measured N-MOS threshold voltage  $V_{TH}$  for devices on IC1 was about 0 volts which could be increased to about 1 volt with - 10v substrate bias. After sintering  $V_{TH}$  became - .5V and + .5V for zero and - 10V substrate voltage respectively. The reduction in  $V_{TH}$  due to sintering is not clearly understood. However this may be caused by a negative ionic charge associated with moisture between the aluminum and the thin oxide which evaporates upon heating or to mobile positive charge contamination in the metal which migrates to the silicon-silicon dioxide surface during sintering. In any case the largest device threshold voltage is desirable hence the silicon wafer was not sintered. In contrast to sintered metal the unsintered aluminum may be easily removed which facilitates re-use of the wafer.

### 7.2.3 Sources of Error for IC1 due to Fabrication Procedures

Several sources of error were identified during the wafer fabrication process and subsequent wafer probe. The tri-chloroethylene (TCE) which was used during thin oxide growth resulted in visible pitting of the oxide when the ratio of TCE flow to dry  $O_2$  flow was too large [37] [38]. It was also noted that the etch rate of the aluminum appeared to be faster in areas of the wafer for which a lesser amount of metal was being removed by the etchant per unit area. This was hypothesized to be an etchant saturation effect. Furthermore, the uniform undercut varied

between  $1 \mu$  to  $3 \mu$  on the first few wafers produced. This was substantially larger than the expected undercut of  $0.75 \mu$ . Another source of error, observed at the probe station, was a high occurrence of low impedance capacitors. Upon close inspection the cause was determined to be pinholes in the thin oxide which created a low resistance between capacitor plates. In addition to this the capacitance value for a particular size of capacitor varied about 10% across the wafer. This may be modeled by a linear oxide gradient parameter of 100 ppm/mil. The subsequent investigation of the furnace associated with thin oxide growth revealed a defective furnace zone giving rise to a severe temperature gradient in the furnace tube. The sources of error just discussed were considerably larger in magnitude than expected and required fabrication schedule corrections before high precision matching accuracy could be achieved.

#### 7.2.4 Data From Capacitance Bridge Measurements for IC1

IC1 was designed so that each capacitor in the array could be probed using a three-terminal measurement technique which nulled out the effects of stray and parasitic capacitance. Such data was taken for 21 arrays on one wafer and is plotted in Figure 7.2. In this plot the vertical axis represents the error in the binary weight of each capacitor from its ideal binary value. The horizontal axis is broken into regions corresponding to each capacitor. The horizontal segment which bisects each error band is the mean value of error for each capacitor. The arrow defines the standard deviation in the error distribution about the mean value. Several conclusions may be deduced from this data. First there exists a large systematic error in capacitor ratios since the mean values

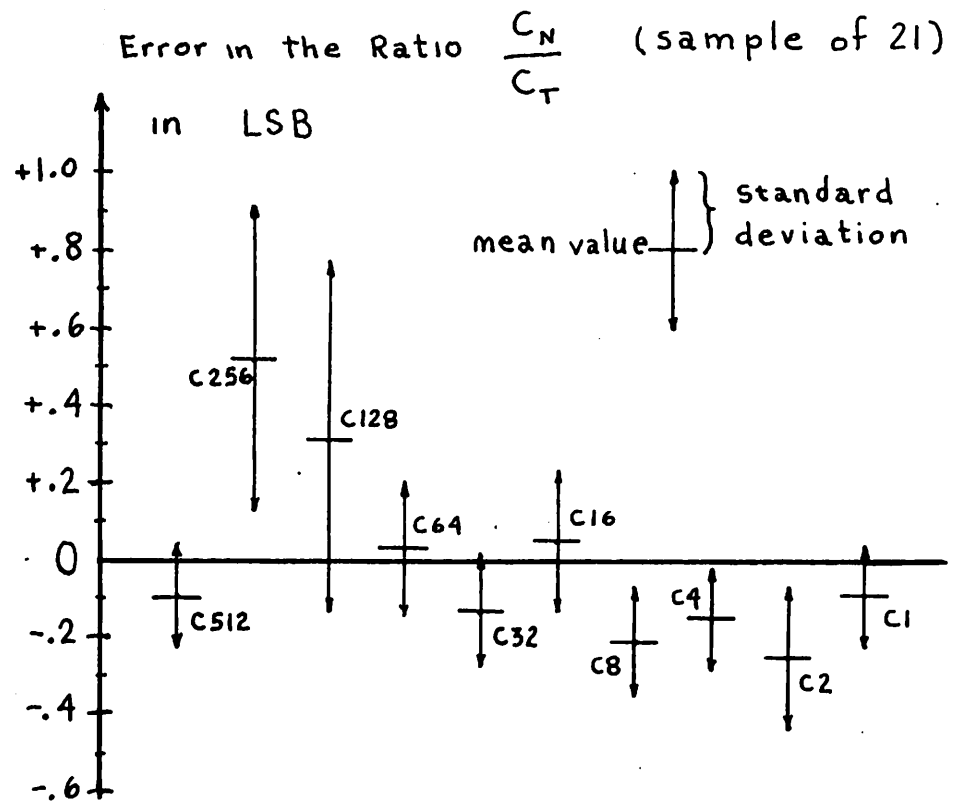


Figure 7.2: The mean value and standard deviations in capacitor ratios for IC1.



in some cases are significantly different from zero. Second the distribution of errors about each mean value is large and is probably due to a combination of factors which vary locally from die to die. There is no clear explanation for the small standard deviation of C512 compared to that for C256 and C128 although the common centroid for C512 could cause a reduced dependence of C512 upon variations in the direction of the oxide gradient.

An attempt is made to identify the sources of error leading to the particular distribution of mean value errors in Figure 7.3. The direction of decreasing oxide thickness is shown in Figure 7.3(a) superimposed upon the capacitor layout for IC1. The approximate direction of the first order gradient was determined by probing capacitors on each die and mapping the change in their absolute values. The effect of this oxide gradient introduces a capacitor ratio error for each capacitor as shown qualitatively in Figure 7.3(b). Since C512 has common centroid geometry to some extent its mean error value is estimated to be less than that for C256 as shown. There are two additional sources of error which are computed analytically and also plotted. Figure 7.3(c) shows the approximate error distribution for  $1 \mu$  uniform undercut. An additional source of error for IC1 was the neglect of overlap capacitance to to interconnect and alignment stubs passing over the thicker oxide over the  $N^+$  regions. It may be seen from the die photo in Figure 7.1 that neither half of C512 requires an alignment stub as do the other capacitors. The ratio error due to this additional capacitance is plotted in Figure 7.3(d). A composite summation of these 3 sources of error when weighted properly could give the observed mean value error distribution which is reproduced in Figure 7.3(e) for comparative analysis. One important conclusion from this study is

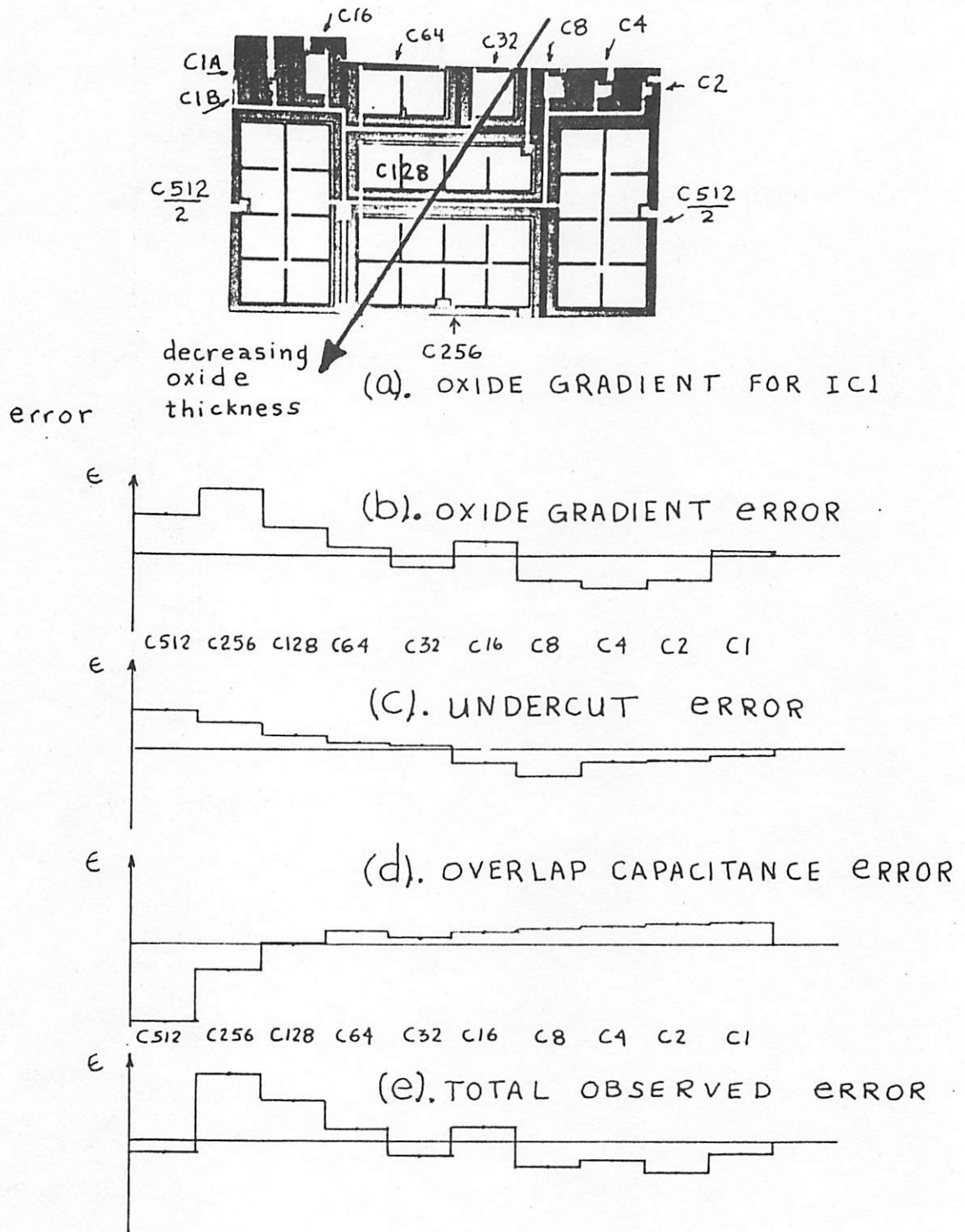


Figure 7.3: Sources of error for IC1.

that the oxide gradient effects must be considerably larger than the other 2 error sources in order to give the observed error distribution in Figure 7.3(e). In addition the error distribution including the systematic error is estimated from Figure 7.2 to be  $\pm 1$  LSB hence the nominal matching accuracy is sufficient for conversion linearity only for 8 or 9 bits of resolution.

#### 7.2.5 Operation of IC1 in the ADC System

Several IC1 circuits were packaged and operated in the ADC system of Figure 6.1. It was verified by measurement that the system performed properly in a logical sense and the comparator offset cancellation functioned correctly. The entire system provided 10-bit resolution for input voltages from 0 to 10 volts.

The linearity of the RADCAP system was determined by measuring the transition voltage at which each individual bit turns on. The deviations in transition voltages were then computed and from these the worst case deviation (WCD) from linearity was determined.

The performance of 3 test circuits was evaluated and compared with their corresponding capacitance bridge measurements. These circuits were sintered to promote stronger contact pads for ultrasonic bonding. Figure 7.4 illustrates a comparison of the distribution of the mean error in capacitor ratios from bridge data to the distribution of mean errors derived by transition voltage measurement. This study indicates the relative reliability of the bridge data as a measure of the conversion nonlinearity. It further illustrates that capacitor ratio error accounts for nearly all of the observed deviation from linearity.

The performance of several circuits was evaluated and the average

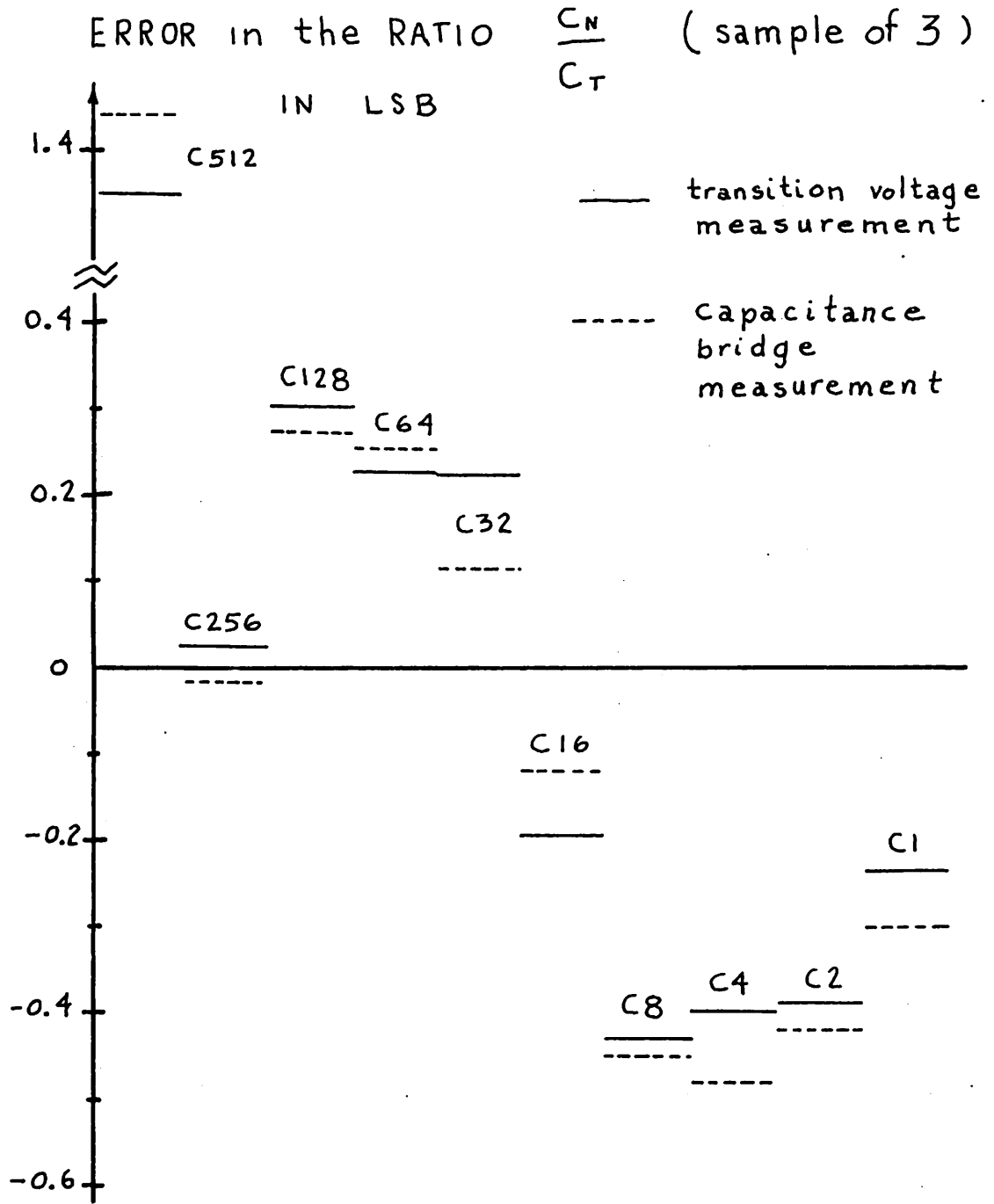


Figure 7.4: A comparison of mean error distributions from capacitance bridge and transition voltage measurements.

WCD from linearity was found to be  $\pm 2$  LSB for 10 bits of resolution. Conversely the maximum resolution for  $\pm \frac{1}{2}$  LSB linearity was between 8 and 9 bits. It was apparent that the systematic error (the mean error) must be eliminated from the system before further evaluation of performance data would be profitable. Analysis of the relative effects of oxide gradient and uniform undercut earlier in this chapter supports the conclusion that greater insensitivity to these two error sources by layout and fabrication modifications was required in order to achieve high yields at  $\pm \frac{1}{2}$  LSB linearity for 10 bits of resolution.

### 7.3 Fabrication Modifications Required to Correct Defects in ICl

Several changes in the fabrication procedure were made in an effort to correct defects discovered in ICl. It was suspected that the N-MOS thresholds were low due to contamination of the vacuum chamber and associated equipment used during the filament evaporation of aluminum [39]. Therefore an electron beam evaporation system was developed and dedicated strictly to MOS in an attempt to reduce positive ion contamination and increase the device threshold voltage. In addition to this the TCE oxide growth with an initial TCE purge to avoid pitting observed with TCE oxides. Another modification intended to reduce damage to the thin oxide was a double exposure of the contact mask after shifting the working plate frame on the mask aligner. This technique insures that dust particles or spot defects will not create pinholes in the thin oxide thereby reducing the number of shorted or low impedance capacitors. In addition to this all contact windows were etched along with thin oxide regions. The advantage of this is that a much shorter etching time than before is needed to open the contact windows. This reduces the chances of photo-

resist lift and undercutting during this last oxide etch. A more radical modification in the fabrication schedule was required to offset the defective zone in the furnace used for thin oxide growth. The new oxide growth procedure required that the wafer lie flat on the boat rather than stand vertical. This was an effort to locate the entire wafer in approximately the same laminar flow streams and to utilize the boat itself as a constant temperature region. Furthermore the wafer was withdrawn from the furnace after each 25% of oxide growth and rotated 90° in an attempt to average the effects of temperature profile and flow stream variations upon oxide growth uniformity. The final fabrication changes involved improvements in photomasking associated with the aluminum. First the working plate for the metal mask was reproduced again with an improved focus for a sharper image. The photoresist development procedure was changed to a spray development for better resolution. Also the aluminum etching procedure was modified to include ultrasonic vibration during the etch in order to remove vapor bubbles from the wafer surface and to provide a better circulation of etchant and more uniform removal of aluminum.

#### 7.4 Layout Modifications Required to Correct Defects in ICl

Evaluation of the measured capacitor matching data indicated that the two largest sources of error were uniform undercut and oxide gradient as discussed in section 7.2. Uniform undercut was estimated between 1  $\mu$  and 3  $\mu$  depending upon the care taken during the etch. From Appendix D the WCD due to this value of undercut is greater than .6 bit. However, if duplication size 8 were used to configure the array a maximum value of 2  $\mu$  undercut would be tolerable. Therefore this duplication size was

chosen for the new layout. Another point of interest is that the array layout design for IC2 did not include in each capacitor that component due to interconnect passing over the diffused  $N^+$  back plates. Actually this component was not neglected since from calculations it results in a ratio error distribution somewhat opposite to that caused by uniform undercut. Hence it was of interest to determine whether the undercut could be so well controlled that the ratio errors could be converged to zero by beginning with an "oversized" array. A theoretical mean error distribution including this interconnect capacitance is shown in Figure 7.5 for various values of undercut. The effect of uniform undercut upon capacitor ratio error is to increase the error in large capacitors and to decrease error for smaller capacitors. From this plot a nominal uniform undercut of  $2 \mu$  would result in minimal total error.

The viability of uniform undercut cancellation rests heavily upon the initial premise that undercut is indeed uniform at every metal edge. In Chapter 4 several mechanisms were proposed which could lead to regional non uniformities in etch rate. One mechanism, etchant saturation, was highly suspect since manifestations of this kind of effect were observed on several occasions. Therefore a layout modification was developed to increase the probability that the etch rate would tend to be uniform at the capacitor metal plate edges. This scheme involved floating metal strips or other metal lines placed the same distance from every capacitor plate edge so that the etchant concentration would tend to be equal at all capacitor plate edges. It can be shown that these floating strips have no adverse affect upon conversion accuracy since they merely constitute a stray capacitance to ground.

It was confirmed from measured data in section 7.2 that thin oxide

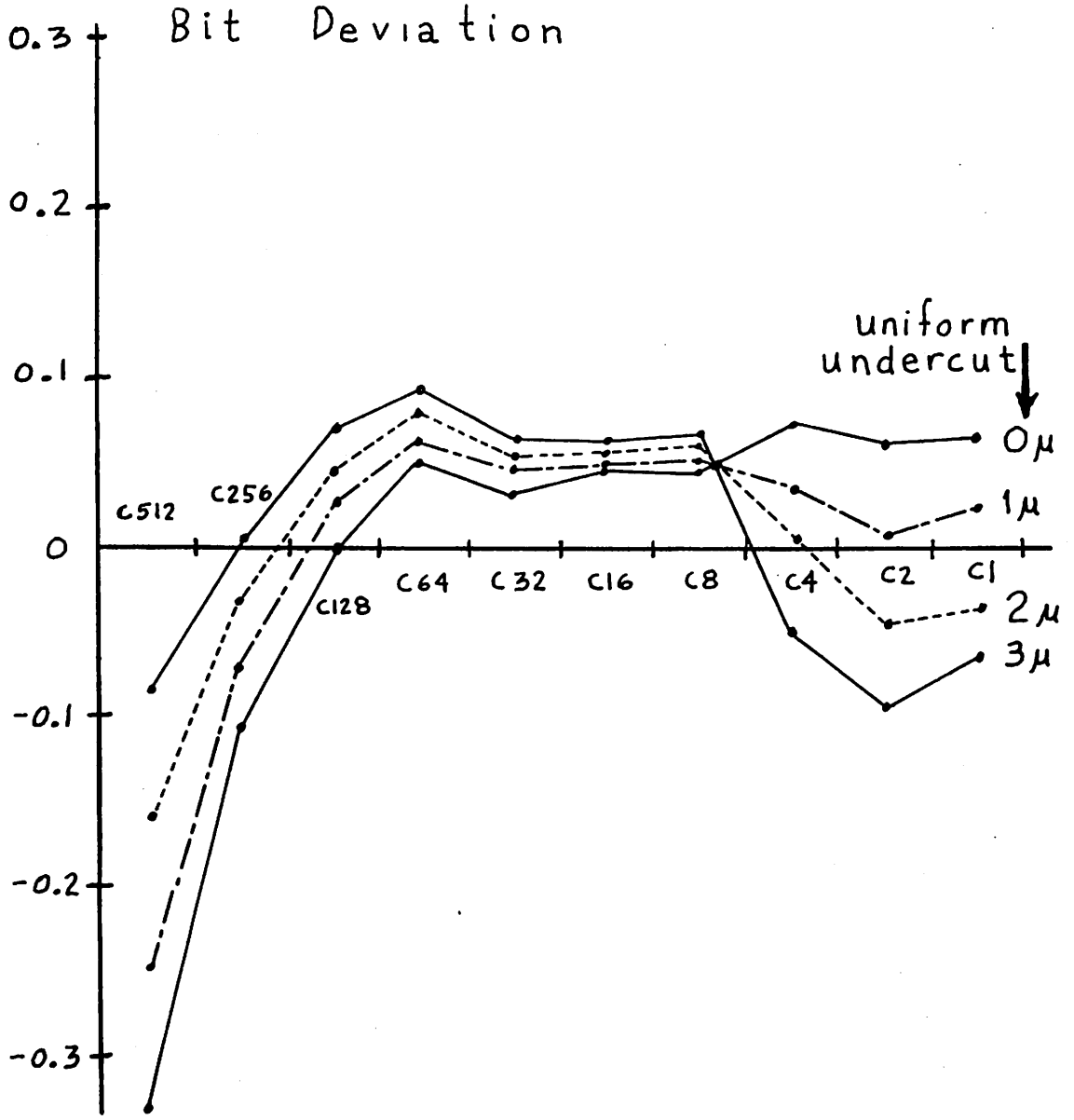


Figure 7.5: Capacitor ratio error distribution as a function of uniform undercut when interconnect capacitance is included.



gradients resulted in large ratio errors. In effect, a more extensive common centroid geometry was needed to reduce the effects of these gradients. A common centroid geometry was therefore chosen for the 5 largest capacitors since the errors in the smaller capacitors were small in absolute value.

Implementation of the layout modifications just discussed resulted in experimental integrated circuit #2 (IC2) shown in the die photo in Figure 7.6. The dimensions of the active area containing the capacitor array is  $75 \times 58$  mils. The common centroid geometry and capacitor reproduction size can be clearly seen along with the floating metal strips. Also incorporated into this design was the existence for each capacitor of an equal number of  $90^\circ$  and  $270^\circ$  corners. This provided a first order cancellation of corner rounding effects.

An additional layout modification was a 200X reduction of the original artwork rather than 400X as used for IC1. While this increased the effect of rubylith cutting errors it had the added advantage of processing independence from commercial reduction facilities. This philosophy was consistent with the expectation that a systematic error would indeed be found but would be subsequently removed by a trim of the artwork and generation of a new working plate.

### 7.5 Experimental Results for IC2

This section contains the measured results for the second experimental integrated circuit. The N-MOS device parameters are examined in section 7.5.1 and the discovery and subsequent elimination of systematic capacitor ratio error is discussed in the next two sections. In sections 7.5.4 and 7.5.5 evaluations are made of the performance of the RADCAP ADC

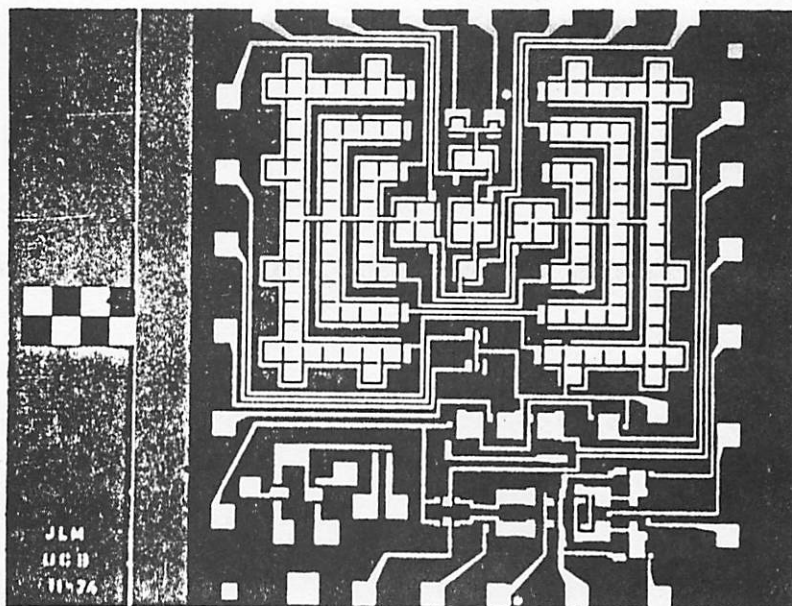


Figure 7.6: Die photo of IC2.

system. In the final section the analysis of remaining error is discussed.

### 7.5.1 N-MOS Device Parameters

The N-MOS device parameters were determined using test structures on IC2. These parameters apply to a p-type (100) substrate having a bulk doping of  $N_B = 3 \times 10^{15}$  which was chosen for experimental work. The curves shown in Figure 7.7 illustrates the variations in drain current versus gate-source voltage,  $I_{DS}$  vs  $V_{GS}$ , for the case in which the drain and gate are connected, hence  $V_{GS} = V_{DS}$ . From left to right the curves are for bulk-source voltages ( $V_{BS}$ ) of 0, - 2.5, - 5, - 10, and - 15 volts. Measured data points are plotted for convenience as shown in Figure 7.8. The vertical axis is scaled to vary as  $\sqrt{I_{DS}}$ . From this plot the extrapolated zero bias threshold voltages (for  $V_{BS} = 0$ ) is obtained from the intersection of the  $V_{GS}$  axis and the best linear approximation for the set of data points:  $V_{T0} = .17$  volt. The slope of this line also gives the conduction factor  $K = \frac{W}{L} \frac{\mu}{2} \frac{\epsilon_{ox}}{t_{ox}} = \frac{I_D}{(V_{GS} - V_{T0})^2}$  and  $K = 120 \frac{\mu A}{V^2}$ . Using  $\frac{W}{L} = 9.4$  for the particular device and  $\frac{\epsilon_{ox}}{t_{ox}} = 3.4 \times 10^{-8} \frac{F}{cm}$  the experimental electron mobility is  $\mu = 750 \frac{cm^2}{V-s}$ . This value is somewhat lower, as expected, than the theoretical value of  $1350 \frac{cm^2}{V-s}$ . In addition the best curve fit for  $V_T$ (effective) vs  $V_{BS}$  is for  $N_B = 1.5 \times 10^{15}$ . This is explained by a diffusion mechanism at the silicon surface called boron depletion which reduces the dopant concentration about 50% in this case [40]. A plot of  $V_T$  (effective) as a function of  $V_{BS}$  is shown in Figure 7.9. The slope of this plot is equal to  $\gamma$ , the bulk threshold parameter. From this data  $\gamma = 0.657 V^{\frac{1}{2}}$  which agrees with the theoretical value of  $0.66 V^{\frac{1}{2}}$  for a bulk surface concentration of  $1.5 \times 10^{15}$ . Figure 7.10 is

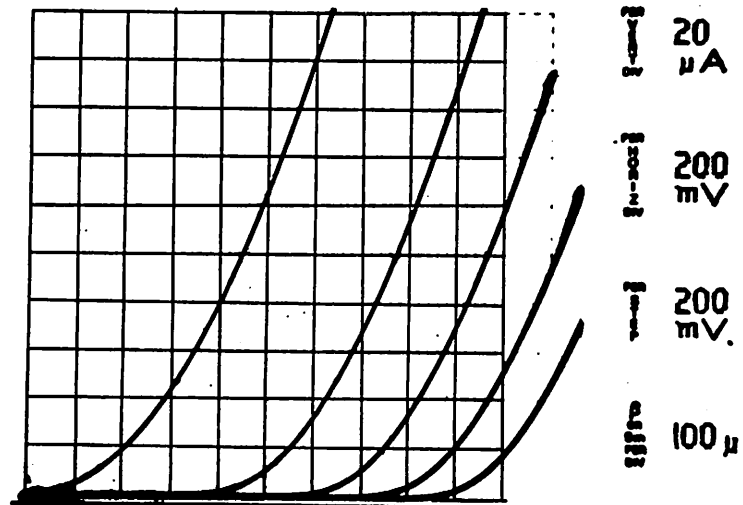


Figure 7.7: The N-MOS drain current characteristics for  $V_{DS} = V_{GS}$  and  $V_{BS} = 0, -2.5, -5, -10,$  and  $-15$  volts.

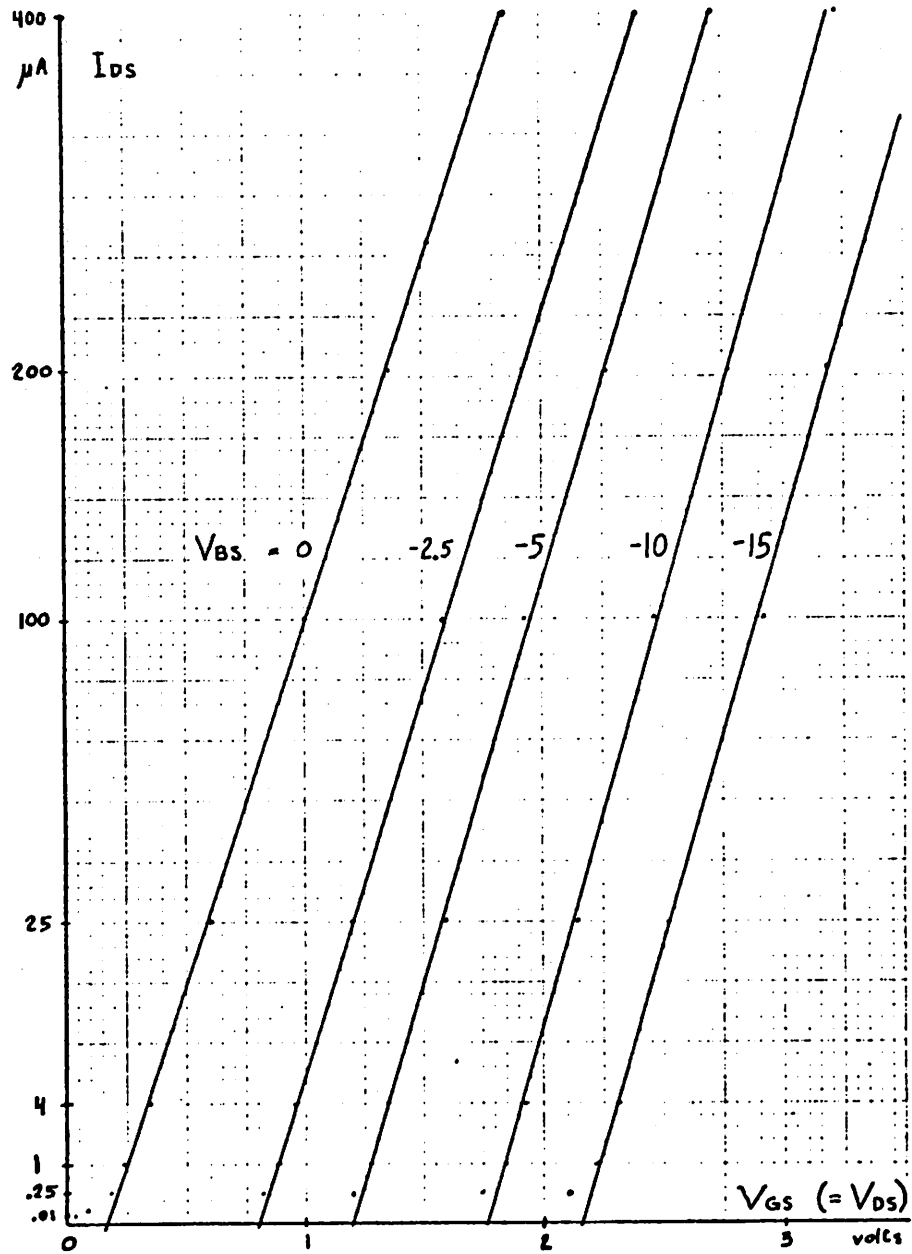


Figure 7.8:  $\sqrt{I_{DS}}$  versus  $V_{GS}$  and the extrapolation of threshold voltage.

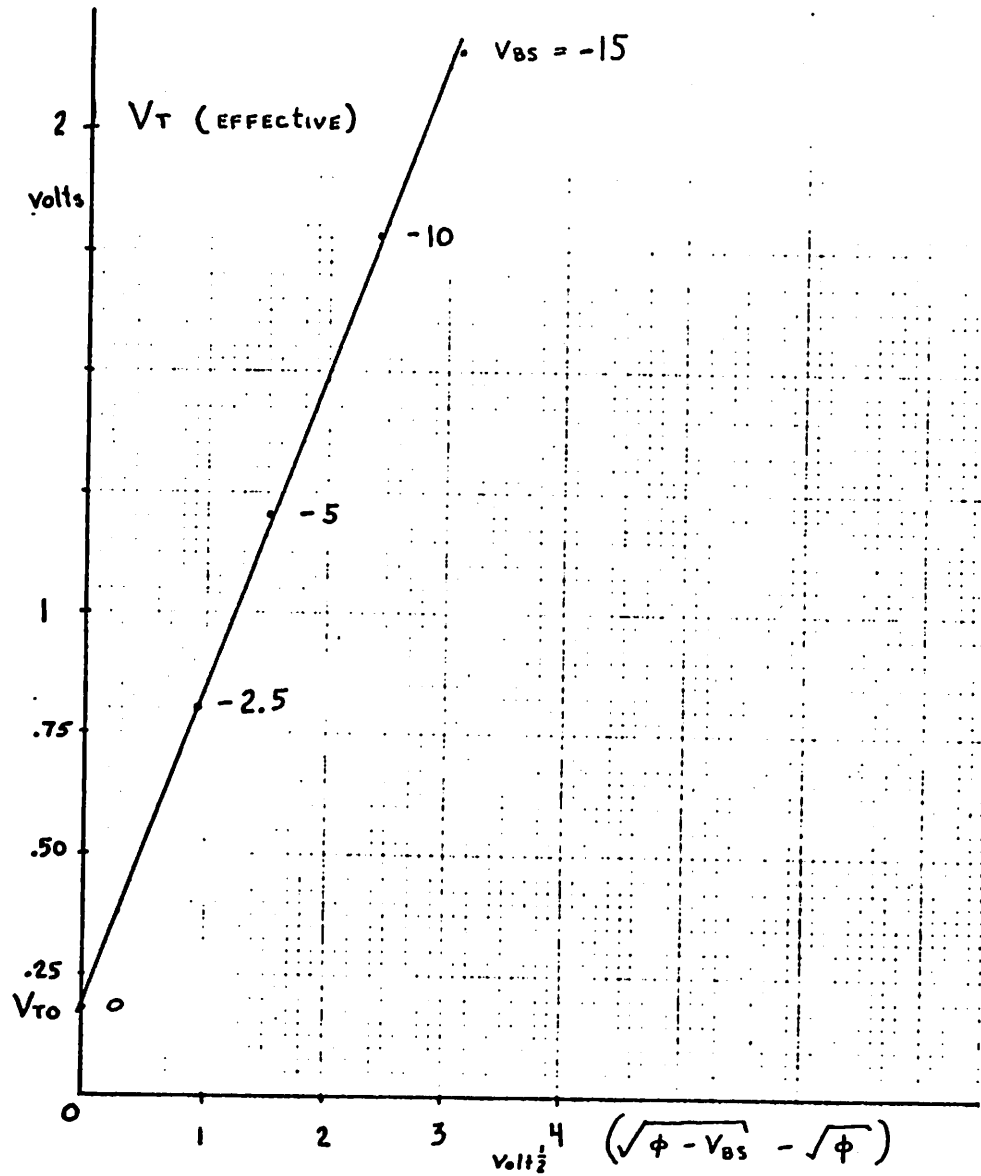


Figure 7.9: The determination of  $V_{T0}$  and the bulk parameter  $\gamma$ .

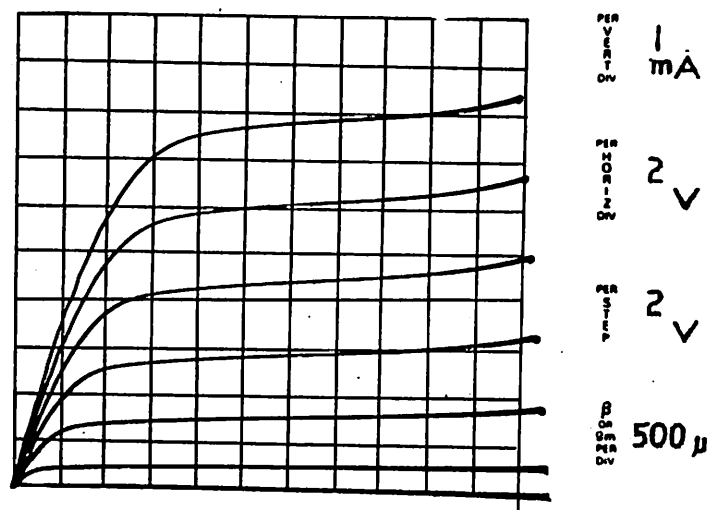


Figure 7.10: N-MOS drain current characteristics.

a photograph of the drain current characteristics  $I_{DS}$  vs  $V_{DS}$ . From this graph the channel length modulation parameter  $\lambda = \frac{.01}{\text{volt}}$ . This corresponds to an Early voltage of  $\frac{1}{\lambda}$ . In conclusion adequate device threshold voltages were realized by improved metal evaporation techniques. The N-MOS devices were enhancement type since the threshold voltage with  $V_{BS} = 0$  is positive. Threshold voltages greater than 2 volts could be obtained with adequate body bias.

### 7.5.2 Measurement of Systematic Error in Capacitor Ratios

Nine capacitor arrays were probed and capacitor values measured. The capacitor ratios were computed and plotted in Figure 7.11. This plot illustrates the distribution in capacitor ratio errors. For each capacitor the mean error ( $m_e$ ) is shown together with the standard deviation ( $\sigma_e$ ) in errors about the mean. A large systematic error is indicated by the fact that  $m_e$  is significantly different from zero. The shape of the mean error distribution fits closely the expected error for oversized capacitors previously shown in Figure 7.11, however the value of the error requires that capacitors be oversized by an amount equal to at least twice the interconnect capacitance. This systematic error could have been caused by the photolithographic distortion in producing the working plates. This process involves a first reduction, a second reduction and a contact print. However, it is more likely that the thin oxide windows were actually larger due to uniform undercut during the oxide etch preceding thin oxide growth. In this case an effectively larger value of capacitance would be due to the interconnect but the error distribution would be precisely as shown in Figure 7.5 for zero undercut but with an increase in vertical scale dimensions. This interpretation



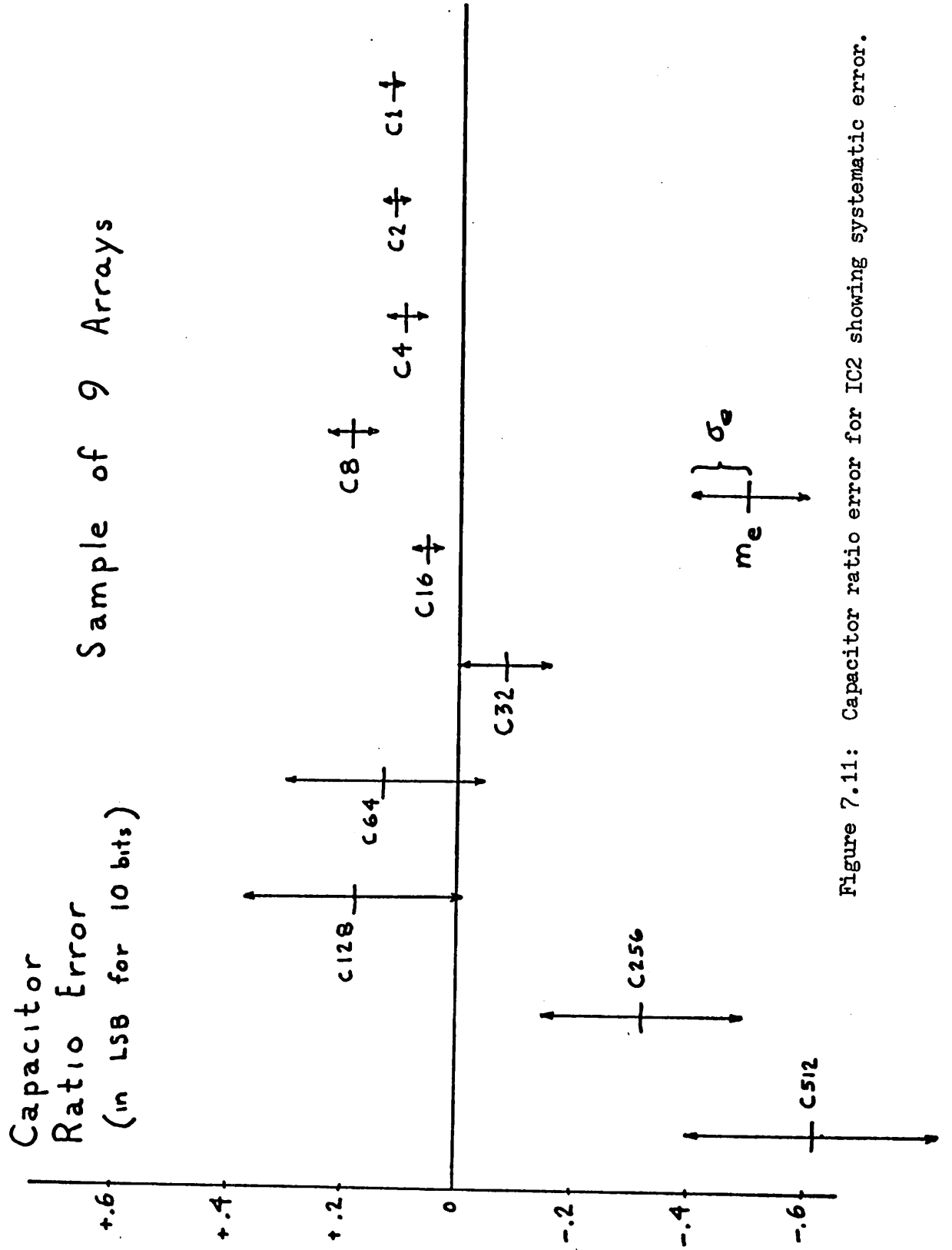


Figure 7.11: Capacitor ratio error for IC2 showing systematic error.

is consistent with the measured data shown in Figure 7.11.

### 7.5.3 Elimination of Systematic Error by Mask Trim

In the previous section the mean value of systematic error was determined for each capacitor by averaging a certain number of samples. The removal of this error was performed by the addition or subtraction of area to each capacitor on the ruby lith defining the metal mask. Each area that was added or subtracted corresponded to the mean systematic error. A new working plate was produced and was used for the fabrication of 3 new wafers. From these a sample of 47 arrays were probed and the capacitor ratios were computed again as in section 7.5.2. The results of these measurements are shown in Figure 7.12. This plot represents the distribution of capacitor ratio errors. The mean error  $m_e$  and the standard deviation  $\sigma_e$  are indicated graphically for each capacitor. From this data it may be deduced that if capacitor ratio error were the only factor affecting yield the yield for  $\pm \frac{1}{2}$  LSB linearity at 8, 9 and 10 bits of resolution would be 98%, 94% and 45% for this sample of 47 arrays. The ability to remove systematic error by a mask trim has been demonstrated.

The correct interpretation of the standard deviation in ratio error ( $\sigma_e$ ) is not clear; however, it is helpful to plot these standard deviations as shown in Figure 7.13. The vertical scale is  $\log \sigma_e$  while the horizontal scale is proportional to binary weight and consequently to the perimeter length ratio (and area ratio) for each capacitor. Assuming for the moment that this error distribution is due to a random mechanism which is uniformly distributed along the capacitor edges then the resultant random variation in capacitor ratio error could be expressed as  $\sigma_e(i) = \sqrt{i} \sigma_e(0)$ , for  $i = 0, 1, 2, 3, \dots, (N-1)$ , if the number of statistically averaged samples

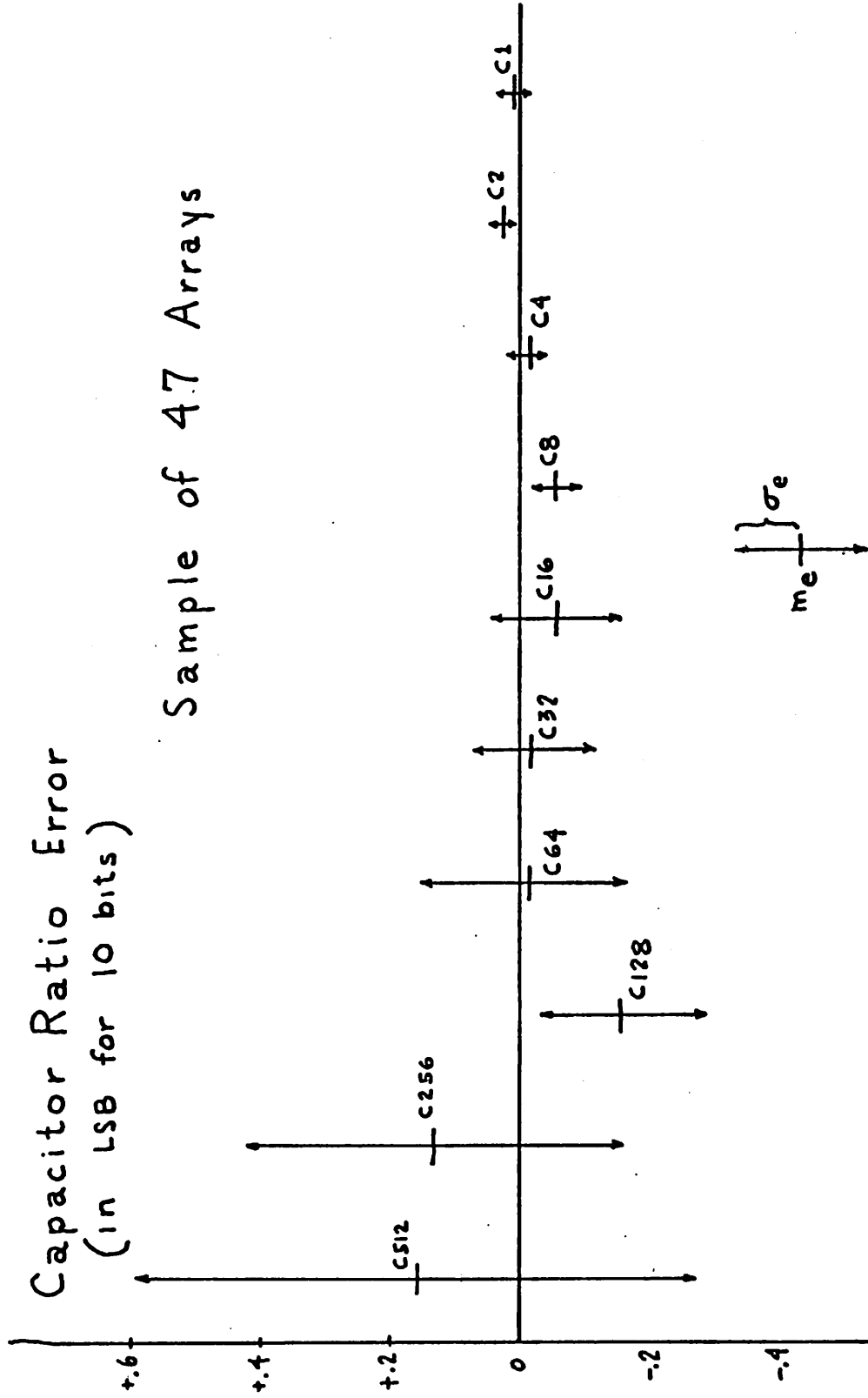


Figure 7.12: Capacitor ratio error after the first mask trim.

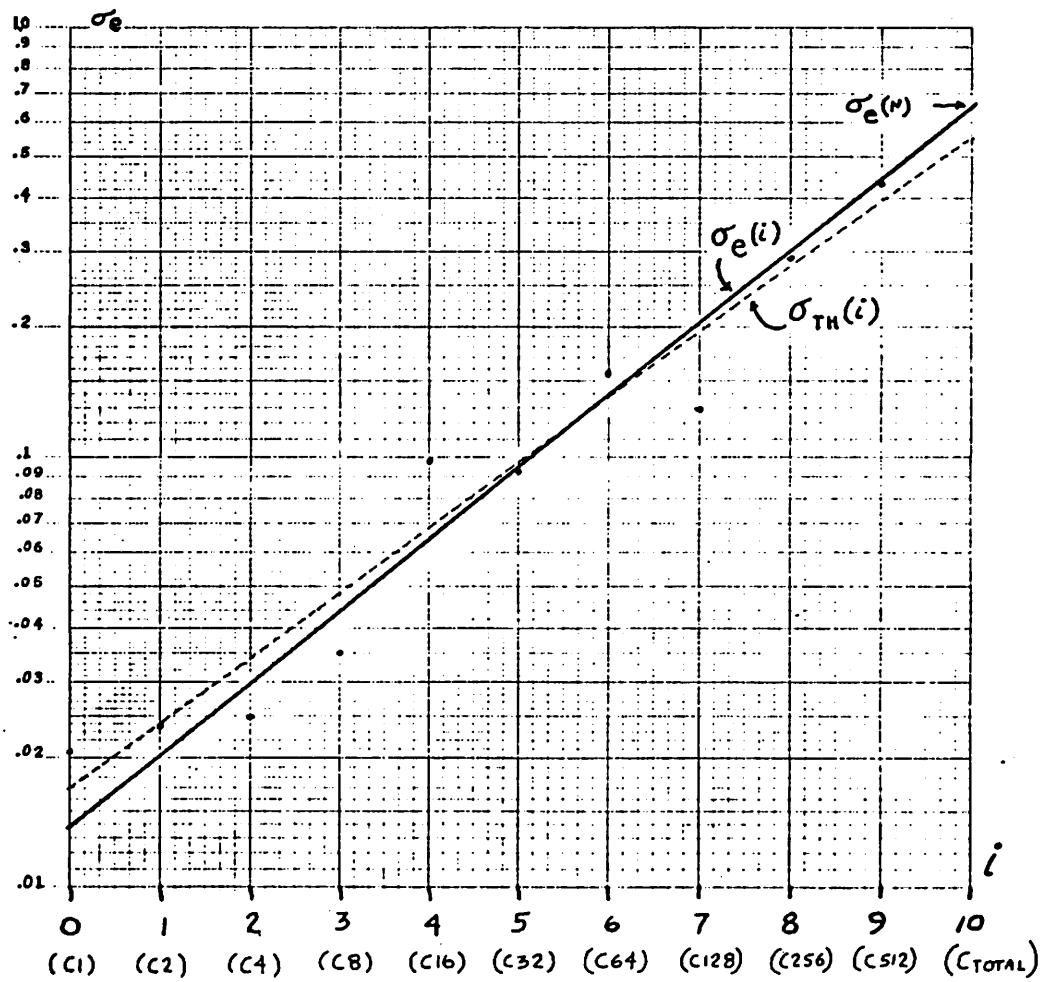


Figure 7.13: The measured standard deviation in capacitor ratio error  $\sigma_e(i)$  compared with a theoretical distribution for a random variable  $\sigma_{th}(i)$ .

is large. This is due to the fact that the standard deviation of a random variable increases by  $\sqrt{2}$  if the sample length (perimeter length in this case) is doubled. From this equation a plot of  $\sigma_e(i)$  versus binary weight should theoretically have a slope of  $\frac{1}{2}$ . This theoretical line denoted as  $\sigma_{TH}(i)$  is superimposed on the graph of  $\sigma_e(i)$  versus binary weight  $i$  and a rough fit is observed. This tends to support the belief that the actual error distribution is due to the presence of a random mechanism which is uniformly distributed along the capacitor edge. It is instructive to pursue this analysis by constructing a line through the more significant data points. This line, denoted as  $\sigma_e(i)$  in Figure 7.13, intersects the line corresponding to  $i = N$  (where  $N = 10$  in this case) at  $\sigma_e(N) = .66$  LSB. This may be interpreted as the extrapolated standard deviation in the total error for all arrays which corresponds to the nonlinearity (assuming that capacitor ratio error is the most significant component of nonlinearity which is true for RADCAP). Hence it may be deduced that 68% ( $\pm$  one standard deviation) of all 47 arrays have a total worst case nonlinearity error  $\sigma_e(N)$  no greater than  $\pm .66$  LSB (for  $N = 10$ ) or that 51% should have a standard deviation of  $\pm .5$  LSB. This can be correlated with the actual findings that 45% have an error less than  $\pm .5$  LSB.

In conclusion the existence of a random variable leading to ratio errors as described in this section requires proper control of geometry in order to minimize the relative effect of this error. That is, if high yield is desirable then large capacitors having long perimeters will tend to reduce the effect of the random variable if the random variable has an absolute effect upon the edges. However, small area capacitors are desirable to save chip area and reduce circuit time constants. There-

fore an "optimal" geometry for the array may be defined as one for which the standard deviation in total array error is

$$\sigma_e(N) \leq \frac{1}{2^{N+1}} \text{ LSB for}$$

68% yield at  $\pm \frac{1}{2}$  LSB linearity. Conversely the optimal total area of the array  $A_{TOT}$  is such that

$$\frac{\sigma_{A_{TOT}}}{A_{TOT}} \leq \frac{1}{2^{N+1}}$$

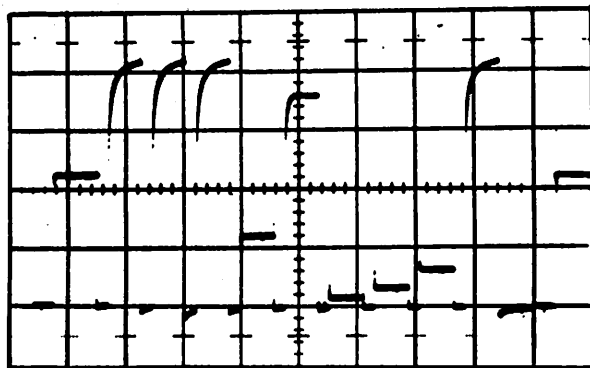
in which  $\sigma_{A_{TOT}}$  is the standard deviation in the area  $A_{TOT}$  as determined by the particular fabrication technique.

#### 7.5.4 Measurement of Transition Point Voltages

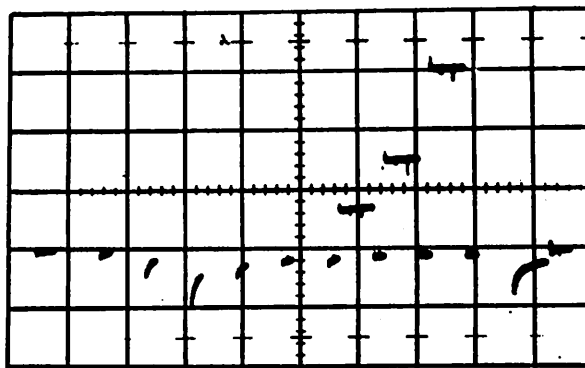
The transition point voltages were measured for IC2 as described in section 7.2.5 for IC1. With the systematic error removed high correlation was expected with the ideal transition points, and with the capacitance bridge data. However, an unexpected additional error was found that was proportionally larger for the larger capacitors. The magnitude of this error was - .5 LSB in the second largest capacitor. The presence of this error was investigated by switching the order in which capacitors were tested and by adding a redundant state between each test. For the following analysis the order of capacitor testing was:

C16 - C128 - C512 - C64 - C8 - C32 - C1 - C2 - C4 - C256

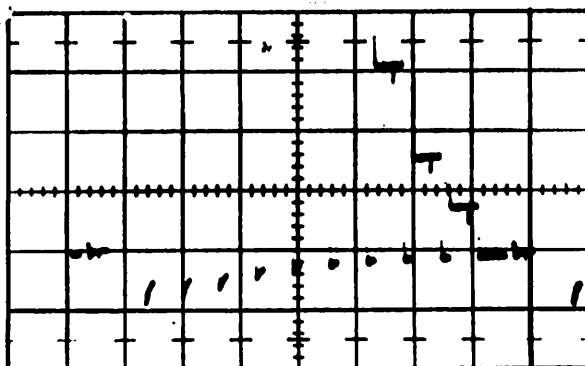
unless otherwise specified. The voltage variation at the top plate of the array is shown in Figure 7.14(a) and with an expanded vertical scale



(a). 80 mV/div ; 50  $\mu$ s/div  
UNORDERED ARRAY



(b). 15 mV/div ; 50  $\mu$ s/div  
UNORDERED ARRAY



(c). 15 mV/div ; 50  $\mu$ s/div  
ORDERED ARRAY

Figure 7.14: The top plate voltage waveform showing dielectric relaxation phenomenon before heat treatment.

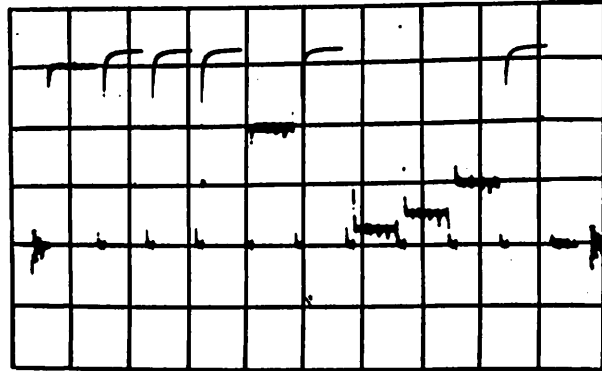
in Figure 7.14(b). This photo shows the top plate voltage variation as each capacitor is charged to  $V_R$  and then rapidly discharged. The presence of an unexpected residual voltage after each negative going transition is evident. Figure 7.14(c) shows the voltage waveform when the correct capacitor order is restored. It was discovered that a mild heat treatment of 200°C for 5 minutes reduced the effect to a negligibly small value. The elimination of this effect for the same circuit as in Figure 7.14 is shown in Figure 7.15(a) and (b) for the unordered capacitor sequence and for the ordered capacitor sequence shown in Figure 7.15(c).

This effect was not detected in capacitance bridge data since these measurements determined only the small signal capacitance; however, the transition voltage measurements were performed with large signals. After heat treatments the transition voltage data correlated with capacitance bridge data as expected. The cause of this effect appears to have been related with moisture at the oxide surface which probably evaporated upon heating.

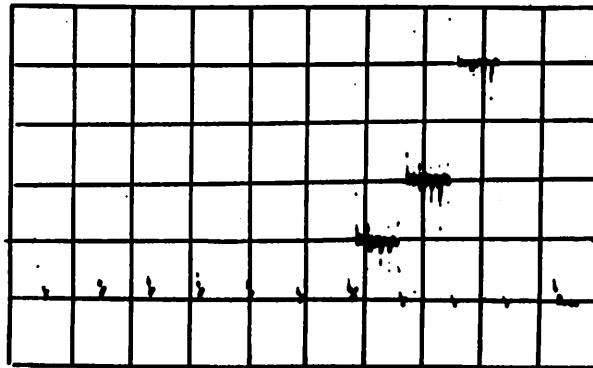
#### 7.5.5 Experimental Measurement of Performance Parameters

When supported by a discrete logic system IC2 became a complete ADC simulating a RADCAP type of circuit. The performance of the complete system was evaluated by first observing the voltage waveforms to and from IC2. The signals VDOWN, VSET, COMP, and  $V_x$  are shown in Figure 7.16(a) as functions of time for an entire conversion cycle. These signals were defined in Chapter VI. The particular digital output shown here is the binary number 1010. Figure 7.16(b) shows the comparator switching waveform. From this photograph the average switching time of the latch is 125 ns. The voltage waveform  $V_x$  at the output of the high

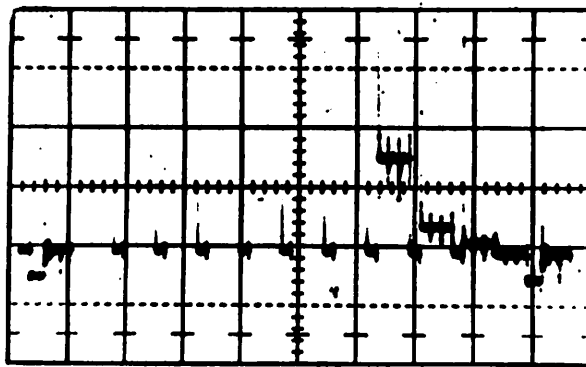




(a). 40 mV/div ; 20  $\mu$ s/div  
UNORDERED ARRAY



(b). 10 mV/div ; 20  $\mu$ s/div  
UNORDERED ARRAY



(c). 30 mV/div ; 20  $\mu$ s/div  
ORDERED ARRAY

Figure 7.15: The top plate voltage waveform showing elimination of dielectric relaxation effect after heating.

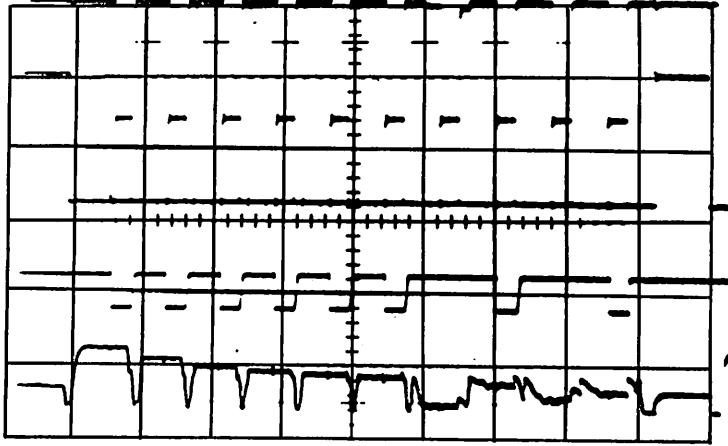


FIGURE 7.16(a): Waveforms : VDOWN (TOP),  
VSET, COMP, AND Vx FOR 1 conversion  
cycle. vert. : 10V/div ; horiz. : 5 $\mu$ s/div.

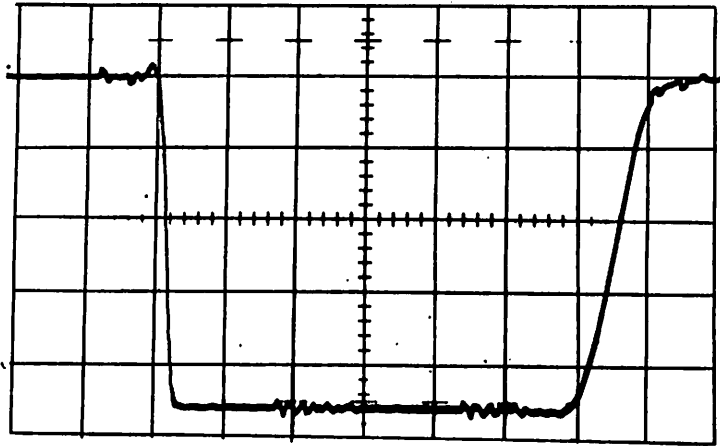


FIGURE 7.16(b) : COMPARATOR SWITCHING  
SIGNAL . vert. : 1V/div ; horiz. : 250ns/div

gain stage is illustrated in Figure 7.17(a) for a full scale 10 kHz triangle wave input. The segment of this waveform showing sample mode precharge is expanded for closer observation in Figure 7.17(b). From this photograph an adequate precharge cycle at this frequency is less than 2  $\mu$ s.

The measurement of nonlinearity was performed using the experimental technique illustrated in Figure 7.18. The output of RADCAP was connected to a 12-bit DAC. Since  $V_{IN}$  was a ramp,  $V_{out}$  of the DAC was a staircase. Both of these waveforms were inputs to a differential amplifier having a sawtooth output corresponding to quantization error plus nonlinearity error. This waveform was recorded on an X-Y plotter and a typical recording is shown also in Figure 7.18. An expanded output is shown in Figure 7.19 and was used for the actual verification of RADCAP performance. This plot enabled a detailed examination of all 1024 states. Since all positive going peaks were between 0 and 10 mV and all negative going peaks were between 0 and - 10 mV the nonlinearity was less than  $\pm .5$  LSB for a resolution of 10 bits. Of 6 units tested three had  $\pm \frac{1}{2}$  LSB linearity for 10 bits while the remaining units had 9 bit resolution. This roughly corresponded to the bridge data for these ICs.

The experimental IC was designed to cancel all offset error except the intrinsic error. In view of this the first transition point should be 1 LSB or 9.7 mV for  $V_R = 10$  v. The typical value measured was about 9 mV with an average uncertainty less than  $\pm 1$  mV. Hence an offset error of less than 2 mV from the design value was observed for all 6 units tested.

An external 10 v reference was supplied to RADCAP hence the measurable gain error of less than 0.05% was due to this supply rather than to the experimental ADC.

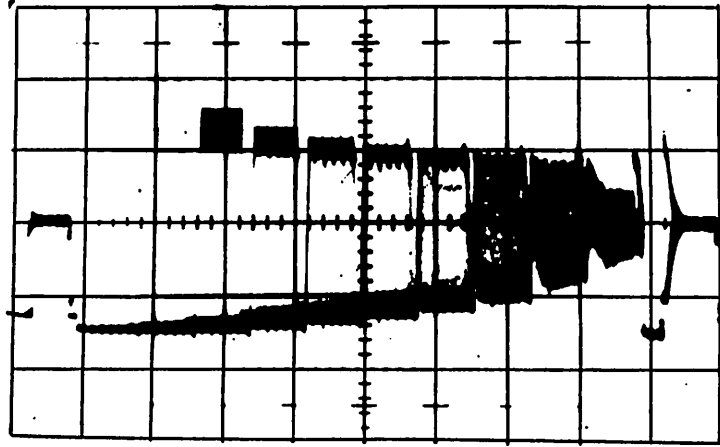


FIGURE 7.17(a):  $V_x$  vs time for  $V_{IN} = 10 \text{ KHz}$   
 FULL SCALE RAMP ;  
 vert. :  $2 \text{ V/div}$  ; horiz. :  $5 \mu\text{s/div}$

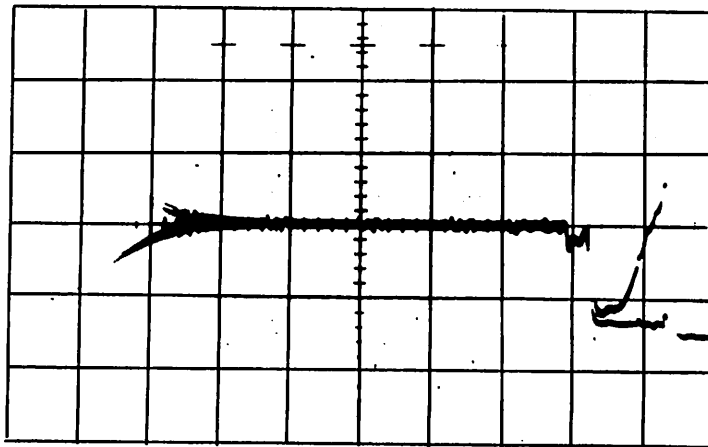
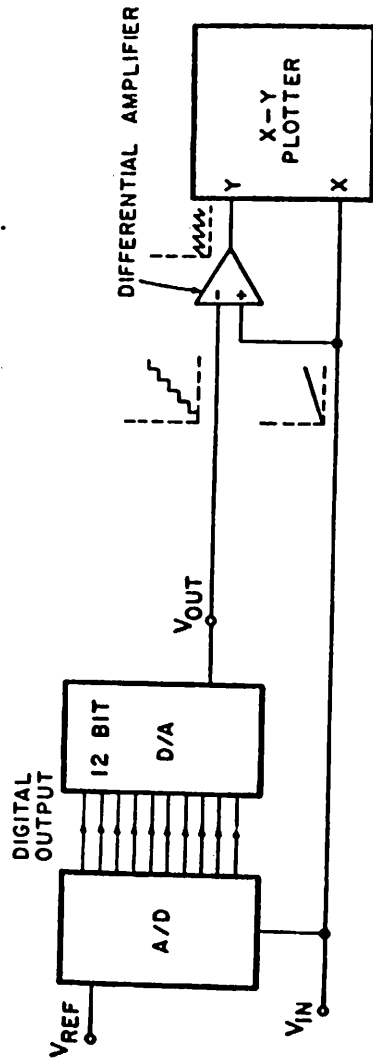


FIGURE 7.17(b):  $V_x$  during Precharge  
 for  $10 \text{ KHz}$  full scale input.  
 vert. :  $2 \text{ V/div}$  ; horiz. :  $0.5 \mu\text{s/div}$ .

### EXPERIMENTAL MEASUREMENT OF ADC NONLINEARITY



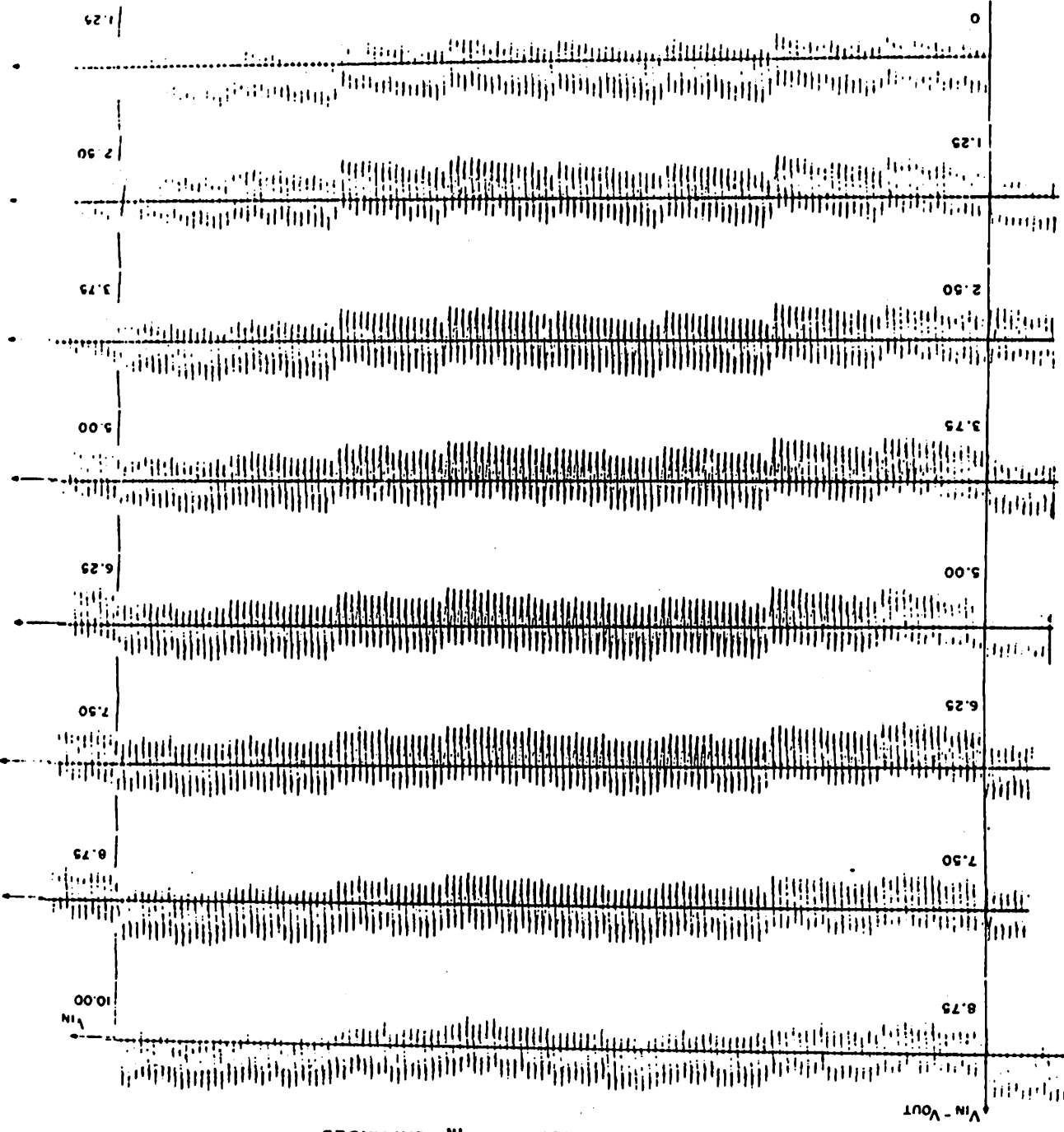
### TYPICAL RECORDING



Figure 7.18: The circuit configuration and typical recording for measurement of nonlinearity.

For all 1024 digital states.

Figure 7.19: An expanded recording showing  $\pm \frac{1}{2}$  LSB linearity



A TYPICAL PLOT OF  $(V_{IN} - V_{OUT})$  VS  $V_{IN}$  EXPANDED

The sample mode acquisition time was the minimum precharge time required for an accurate conversion of a  $\frac{1}{2}$  full-scale voltage step at the input. This was measured at 1.5  $\mu\text{s}$ . The total conversion time was measured at 22.8  $\mu\text{s}$ . A summary of performance specifications is given in Table 7.1.

Additional data was taken which illustrate in greater detail the performance characteristics of the RADCAP system when used to sample high frequency sine waves. Figure 7.20(a) is a photograph of 3 waveforms:  $V_{\text{out}}$  of the DAC (from Figure 7.18),  $V_{\text{IN}}$  (a 3 kHz sine wave), and  $V_x$  (the amplified waveform at the top plate of the array). This figure shows the sampling capability for sinusoids. Figures 7.20(b), (c), (d), (e), and (f) illustrate  $V_{\text{out}}$  (DAC) and  $V_{\text{IN}}$  for different frequencies. In these photos the lack of synchronization between sampling rate and  $V_{\text{IN}}$  results in an apparent continuous band for  $V_{\text{out}}$ . This is convenient since the envelope of  $V_{\text{out}}$  is visible. From these figures it is evident that  $V_{\text{out}}$  may be modeled as attenuated and phase-shifted with respect to  $V_{\text{IN}}$ . The actual recovery of the sinusoidal signal from the output  $V_{\text{out}}$  (DAC) is accomplished by a low pass filter. This was done using a 2-pole 5 kHz low pass filter. The filtered output  $V_{\text{out}}$  (5 kHz filter) was then connected to a distortion analyzer from which a THD (total harmonic distortion) of 0.35% for an input frequency range of 200 Hz to 3.5 kHz was measured. Two photographs taken in this frequency range are shown in Figure 7.21 (a) and (b).

#### 7.5.6 Limitations on Matching Accuracy due to Random Edge Location

In section 7.5.3 a random mechanism operating uniformly along the capacitor perimeters during fabrication was postulated as the cause of

## TABLE OF PERFORMANCE DATA

Resolution	10 bits
Linearity	$\pm \frac{1}{2}$ LSB
Input voltage range	0–10 V
Input offset voltage	2 mV
Gain error	< 0.05% (external reference)
Sample mode acquisition time	2.3 $\mu$ s
Total conversion time	22.8 $\mu$ s

Table 7.1: The measured performance data.



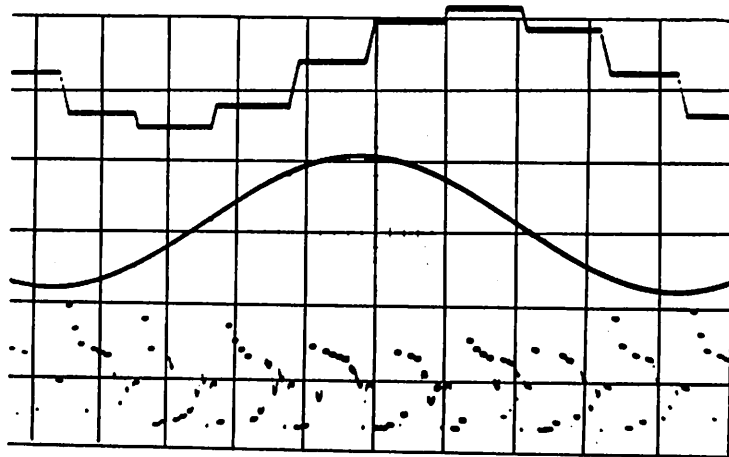


FIGURE 7.20(a):  $V_{out}$  (DAC) (TOP),  $V_{in}$ ,  $V_x$  .  
for 3 kHz full scale sine wave input.  
vert. : 5V/div. ; horiz. : 40 $\mu$ s/div.

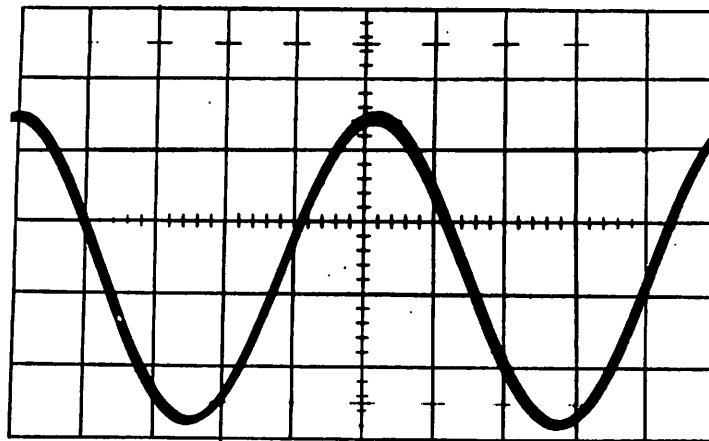


FIGURE 7.20(b):  $V_{out}$  (DAC) shown lagging  $V_{in}$   
for 200 Hz 9V p-p sine wave input.  
vert. : 2V/div ; horiz. : 1ms/div.

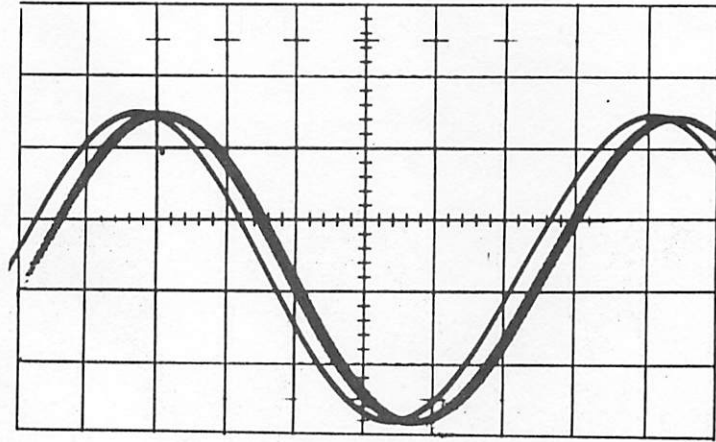


FIGURE 7.20 (c):  $V_{out}$  (DAC) shown lagging  $V_{in}$   
 for 800 Hz, 9V p-p sine wave.  
 vert.: 2V/div.; horiz.: 200  $\mu$ s/div.

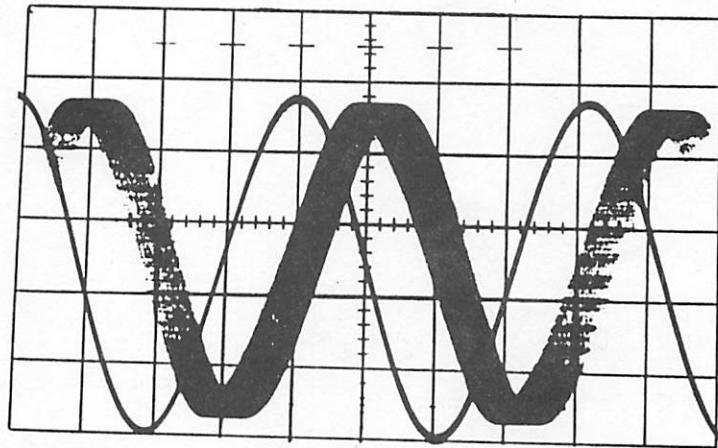


FIGURE 7.20 (d):  $V_{out}$  (DAC) shown lagging  $V_{in}$   
 for 5 kHz, 9.5V p-p sine wave.  
 vert.: 2V/div.; horiz.: 50  $\mu$ s/div.

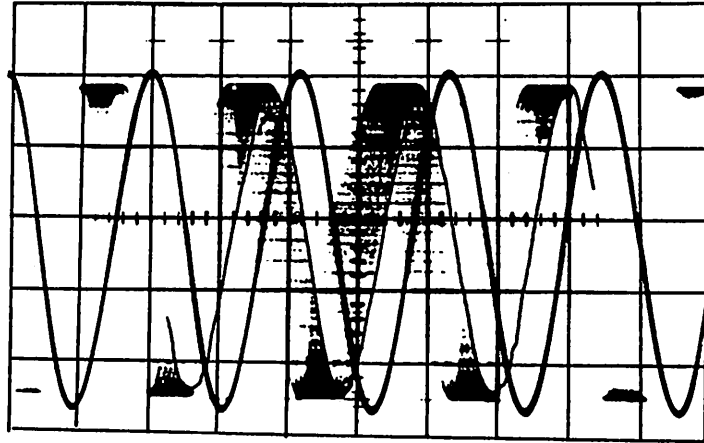


FIGURE 7.20(e):  $V_{out}$  (DAC) shown lagging  $V_{in}$  for a 10 kHz, 9.5 V p-p sine wave input.  
 vert.: 2 V/div ; horiz.: 50  $\mu$ s/div.  
 (image enhanced)

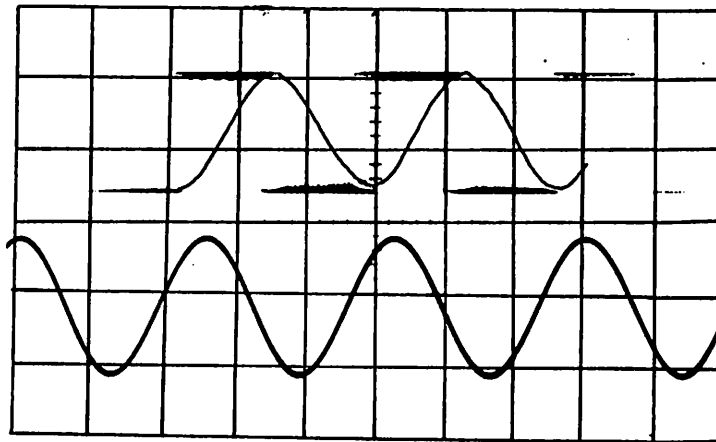


FIGURE 7.20(f):  $V_{out}$  (DAC) shown above  $V_{in}$  for a 20 kHz, 9.5 V p-p sine wave input.  
 vert.: 5 V/div ; horiz.: 20  $\mu$ s/div.  
 (image enhanced.)

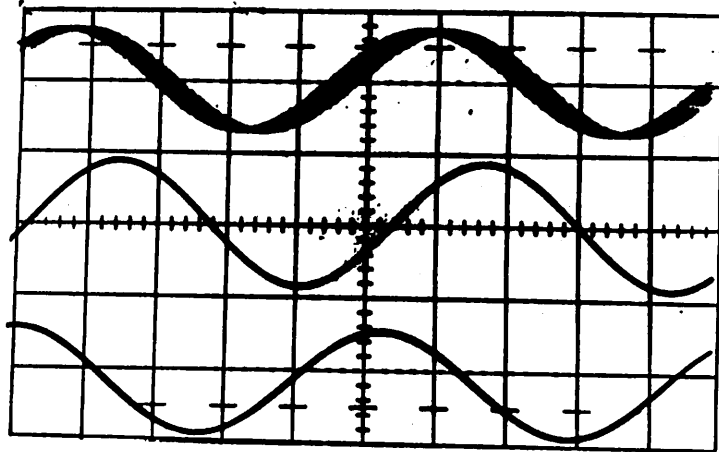


FIGURE 7.21(a):  $V_{out}$  (DAC) (TOP),  $V_{out}$  (5 KHz filter),  $V_{in}$  (1 KHz, 9V p-p) (lower) for 0.35% THD.  
vert. : 5V/div ; horiz. : 200  $\mu$ s/div.

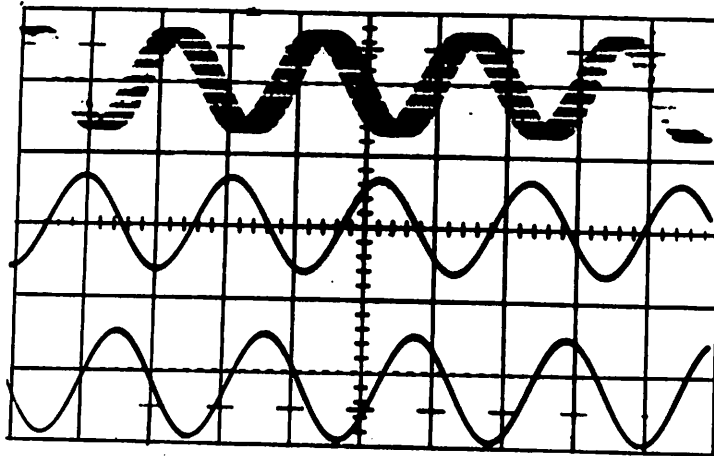


FIGURE 7.21(b):  $V_{out}$  (DAC) (TOP),  $V_{out}$  (5 KHz filter),  $V_{in}$  (1 KHz, 9V p-p) (lower) for 0.35% THD.  
vert. : 5V/div. ; horiz. : 200  $\mu$ s/div.

a random error distribution in capacitor ratios. In section 4.10 random edge location was discussed as a possible source of error. This will now be investigated with the aid of a scanning electron microscope (SEM).

Figure 7.22 shows two pictures of photoresist (PR) edges after development but before etching. In both cases the jagged edges are apparent. The PR was about  $1 \mu$  thick and the approximate random edge location variation is  $.1 \mu$  to  $.2 \mu$ . This defect may arise from the poor resolution capabilities of an emulsion working plate or from the resolution properties of the PR (AZ1350J). Figure 7.23 shows additional examples of the ripples in the PR edges.

In Figure 7.24 the aluminum has been etched but the PR has not yet been removed. A further more serious degradation of the PR is observable. It appears that the etchant also attacks the PR.

Figure 7.25 illustrates the jagged aluminum edges which remain after the metal has been etched and the PR removed.

In conclusion the most significant limitation to increased ratio accuracy is suspected to be the random edge location and this is probably caused by a combination of factors associated with the emulsion working plate, PR resolution, and etchant attack upon the PR as well as non-uniformities associated with etching the aluminum. The observed random error variation was not believed to be significantly dependent upon oxide gradient although this may be a minor factor. In spite of the fact that common centroid was not designed for the smaller capacitors the uniformity in oxide gradient was sufficiently good across the wafer that it would be negligible across one die. In addition the error distribution is not consistent with that which might be expected from an oxide gradient error.

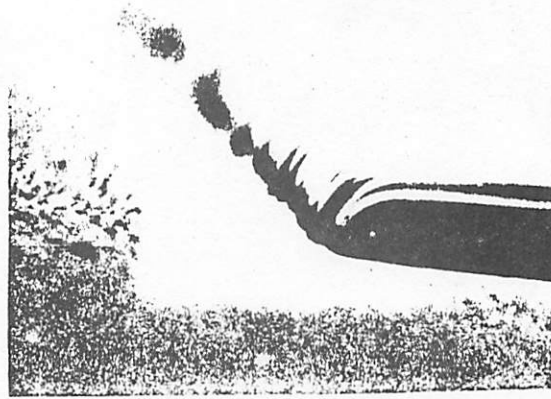


FIGURE 7.22 : PHOTORESIST edges magnified  
10,000 x after development.

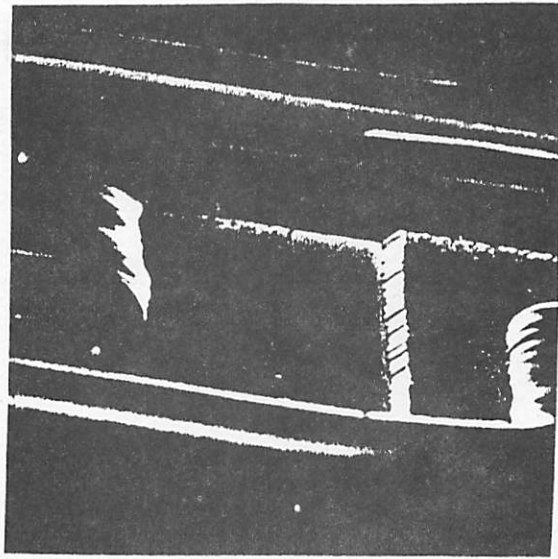
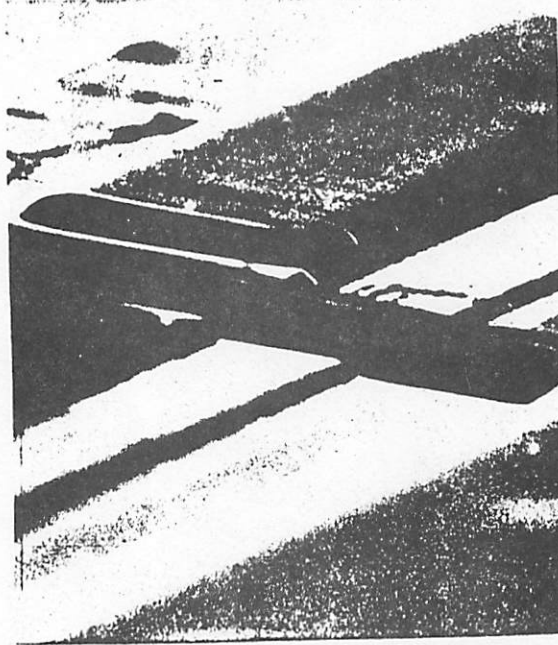
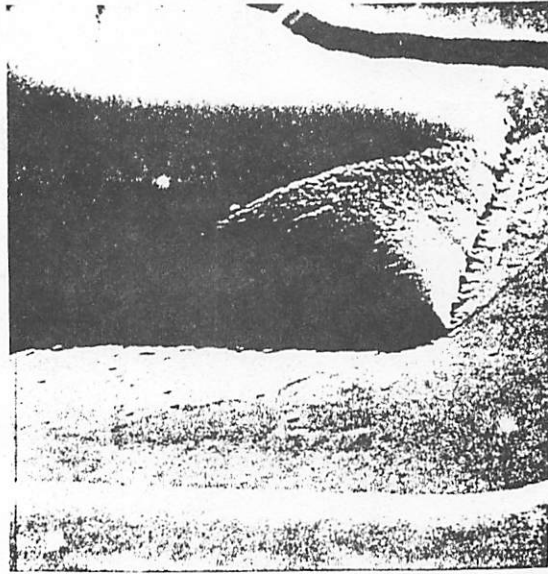


FIGURE 7.23: Developed Photoresist on Aluminum, before etching, showing ripples in resist edges. Magnified 1000x.



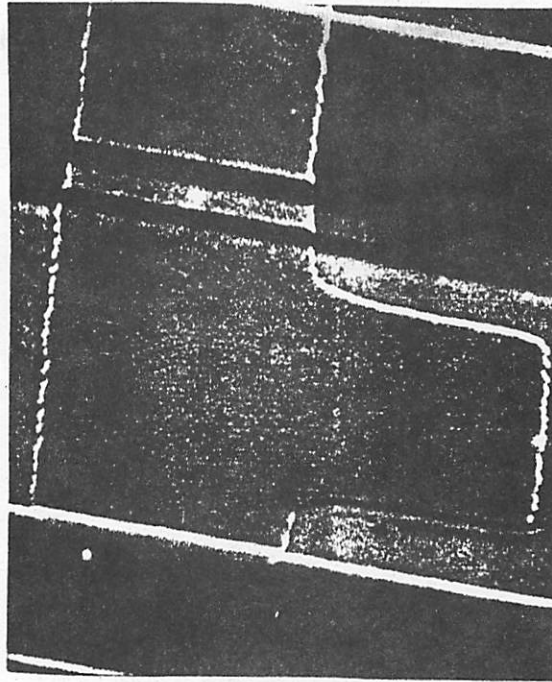
Magnification : 5000 x



Magnification : 12,000 x

FIGURE 7.24 : Photoresist on Aluminum after etching.





Magnification: 3400x



Magnification: 30,000x

FIGURE 7.25 : Aluminum pattern showing random edge location.

## CHAPTER 8

CONCLUSION

The conclusions of this research effort are as follows:

1. The feasibility of single-chip realization of a high speed all-MOS ADC at low cost has been demonstrated. This was done by fabricating an experimental I.C. using N-channel aluminum gate technology. The experimental data indicates that conversion accuracies of 10 bits  $\pm$  1/2 LSB can be achieved at high yield.

2. The principal limitations on the accuracy of this technique are due to the conventional photomasking and chemical etching techniques used in the standard fabrication process. The resolution of the photolithography was identified as the practical limitation upon the ratio accuracy. Some improvement in matching could probably be achieved if special techniques were used to enhance photomasking resolutions such as electron beam exposure and ion beam etching of aluminum [41]. Improvement in matching accuracy could certainly be realized if on-chip trimming techniques were developed.

3. The major practical limitations on the conversion rate are due to the practical minimum value of array capacitance and the accuracy with which feedthrough cancellation can be achieved. Both of these are dependent upon the photolithographic resolution limits of conventional photomasking.

In conclusion, this investigation has demonstrated that, with the addition of an external reference voltage, a single-chip MOS ADC may be realized. It is estimated that this realization would require an active chip area of 90  $\times$  90 mils square and consume 56 mW of power. Table 8.1

contains a summary of these requirements

<u>Component</u>	<u>Area (mils)<sup>2</sup></u>	<u>Power (mV)</u>
Array	55 × 70	1
Comparator	30 × 25	30
Logic & interconnect	60 × 60	25
<hr/>		
Total Chip	90 × 90	56

Power supplies:  $\pm 15V$ ;  $V_R = + 10V \pm .05\%$

Input range: - 10V to + 10V for 20 mV resolution or 0 to  
+ 10V for 10 mV resolution

10-bit conversion time: 23  $\mu$ s

Table 8.1: Estimated single chip RADCAP Realization.

## APPENDIX A

Calculation of Nonlinearity due to Capacitor Voltage Coefficient

As mentioned in section 4.5 the capacitor voltage coefficient  $\alpha$  results in voltage dependent capacitors and this causes nonlinearity. This fact will now be supported by numerical calculations of the resultant error. For the capacitor structure illustrated in Figure 4.7 the equations of interest are:

$$C_1(V_1) = B_1 C_0 (1 - \alpha V_1) = C_X (1 - \alpha V_1)$$

$$\text{and } C_2(V_2) = (2^N - B_1) C_0 (1 + \alpha V_2) = C_Y (1 + \alpha V_2).$$

The change in charge caused by the transient must be equal for both  $C_1$  and  $C_2$ :

$$\Delta Q_1 = \Delta Q_2.$$

$$\text{Then } \int_0^{V_R - V_2} C_1(V_1) dV_1 = \int_0^{V_2} C_2(V_2) dV_2$$

$$\text{and } C_X \left( V_1 - \frac{\alpha V_1^2}{2} \right) \Big|_0^{V_R - V_2} = C_Y \left( V_2 + \frac{\alpha V_2^2}{2} \right) \Big|_0^{V_2}.$$

Solving for  $V_2$  after substituting the limits. The following result is obtained after simplification:

$$V_2 = \frac{C_X}{C_X + C_Y} V_R + \frac{1}{\alpha} \left[ \sqrt{1 + \frac{\alpha V_R B_1}{2^N}} \frac{1 - 2^N}{B_1} - 1 \right]$$

The first term in the equation above is the ideal value of  $V_2$  if  $\alpha$  equals zero. The second term defined as  $\epsilon$  represents the error in  $V_2$  due to voltage coefficient, and is expressed as a function of  $B_i$ .  $\epsilon$  may also be simplified to the expression:

$$\frac{\alpha}{2} \left( \frac{V_R B_i}{2^N} \right)^2 \left( 1 - \frac{2^N}{B_i} \right).$$

Table A.1 lists the error as a function of the digital output for a 10-bit converter with  $\alpha = 22$  ppm/volt.

<u><math>B_i</math></u>	<u><math>\epsilon</math> in millivolts</u>	<u><math>\frac{\epsilon (\alpha)}{\alpha}</math></u>
1024	0	0
992	-.036	$- 1.6 \times 10^3$
960	-.064	$- 2.9 \times 10^3$
896	-.122	$- 5.5 \times 10^3$
768	-.209	$- 9.5 \times 10^3$
640	-.259	$- 11.8 \times 10^3$
512	-.277	$- 12.6 \times 10^3$
384	-.259	$- 11.8 \times 10^3$
256	-.209	$- 9.5 \times 10^3$
128	-.122	$- 5.5 \times 10^3$
64	-.064	$- 2.9 \times 10^3$
32	-.036	$- 1.6 \times 10^3$
0	0	0

TABLE A.1

From the table, the worst case nonlinearity of  $-.3$  mV occurs at  $\frac{1}{2}$  full scale input. This may be generalized to the final result that the worst case error voltage always appears at an input of  $\frac{V_R}{2}$  and its value is:

$$\epsilon = -\alpha \frac{V_R^2}{8}.$$

## APPENDIX B

Digital Logic Circuit

The digital logic circuit is shown in Figure B.1 along with connections to the experimental I.C. There are five logic blocks in addition to the chip: a sequencer, two signal generators, CMOS switches, and a buffer. All signal paths between the six circuits are labeled in the figure. The timing diagram for all signals to the chip are shown in Figure B.2. The timing diagram for capacitor signals is illustrated in greater detail in Figure B.3. The state table necessary to generate the desired timing is shown in Figure B.4. The implementation of the state table results in the circuit schematics for the Sequencer, the Switch Signal Generator, the Capacitor Signal Generator, the CMOS switches, and the Output Buffer which are respectively shown in Figures B.5 through B.9. Figure B.10 illustrates the minimal operating configuration for the experimental chip. The comparator outputs are translated from MOS levels to TTL levels by the CMOS NOR gates shown. Figure B.11 shows the bonding diagram for an experimental chip mounted in a 28 pin DIP.

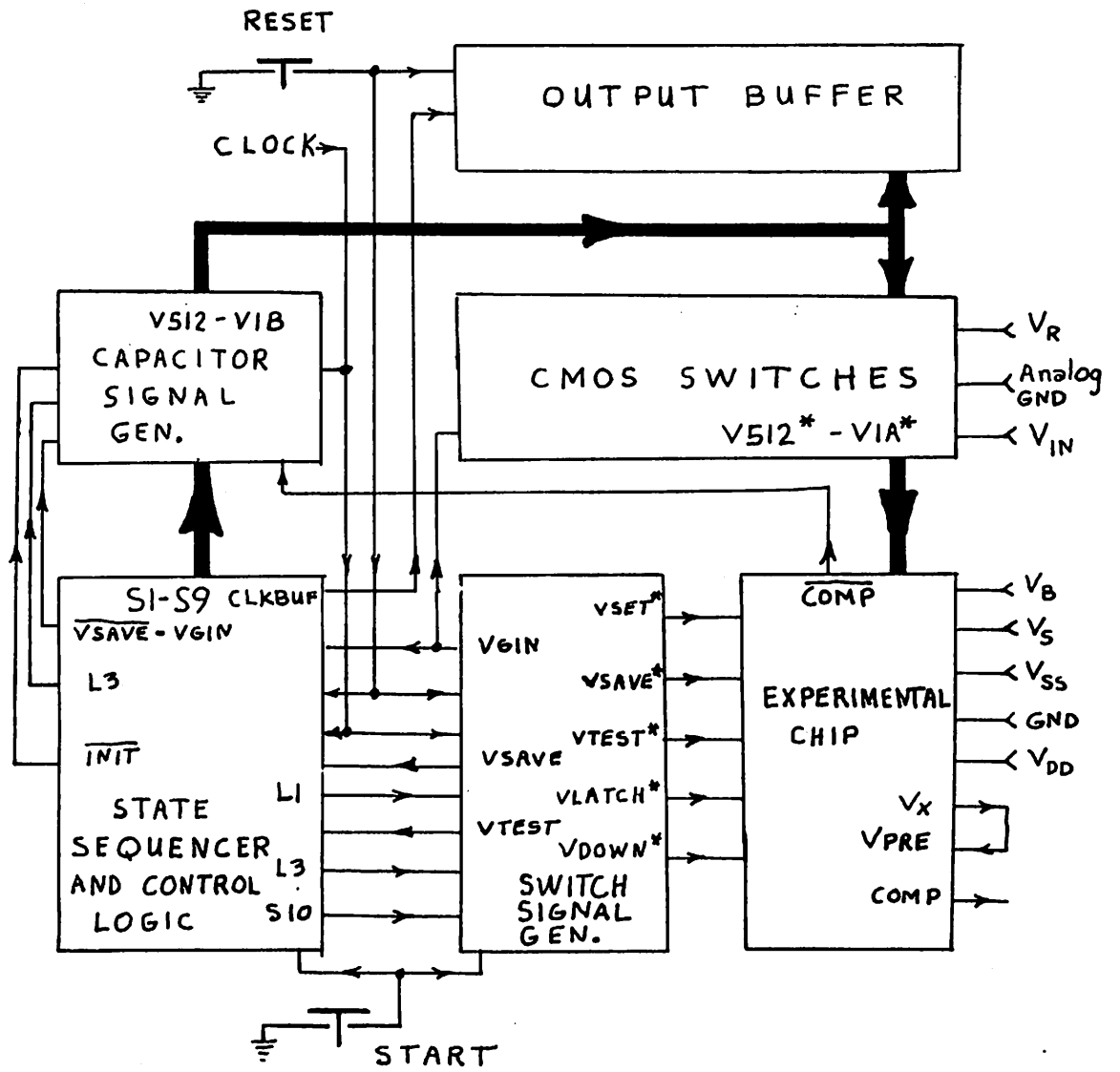


Figure B.1: The complete ADC.



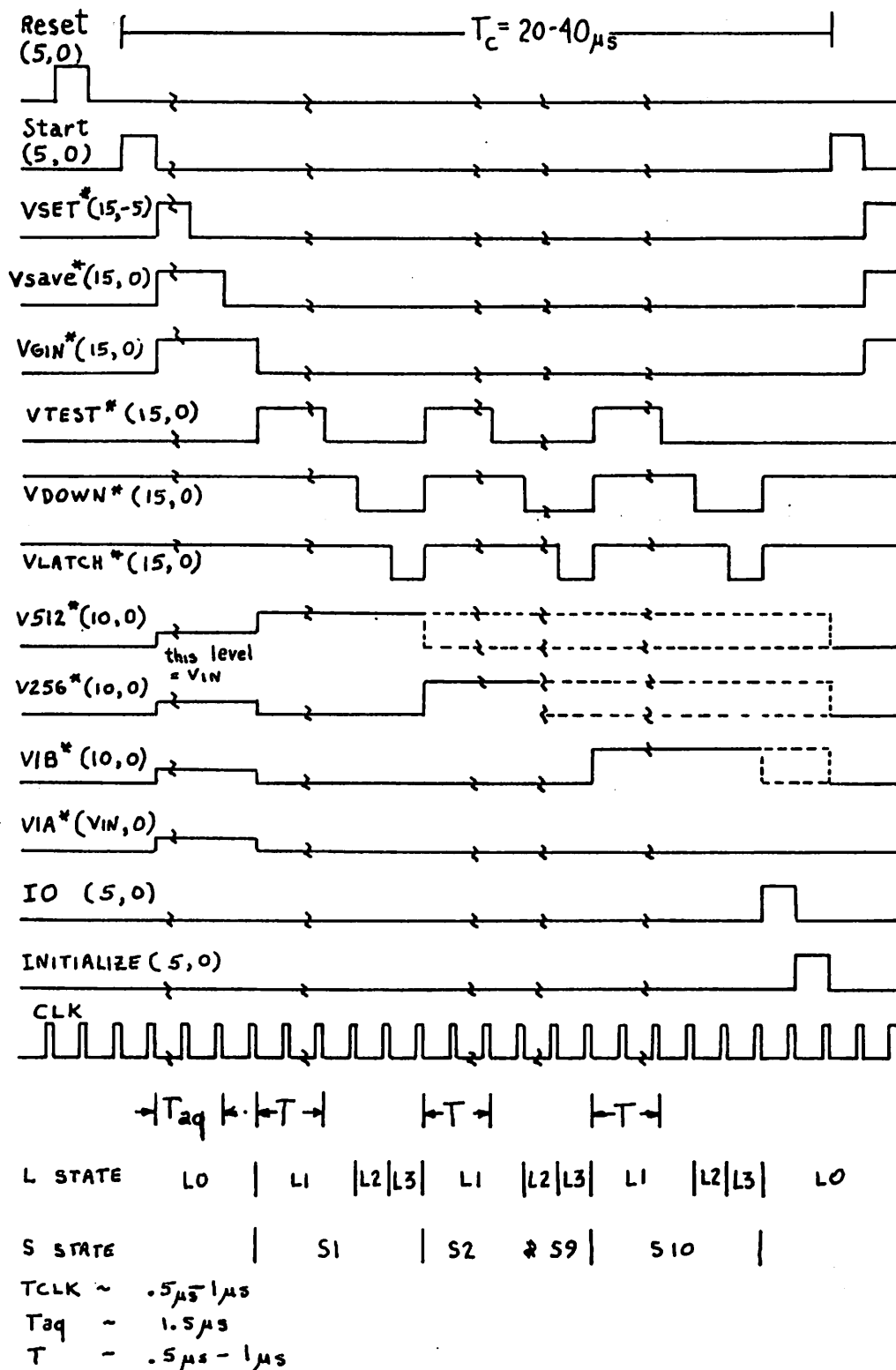
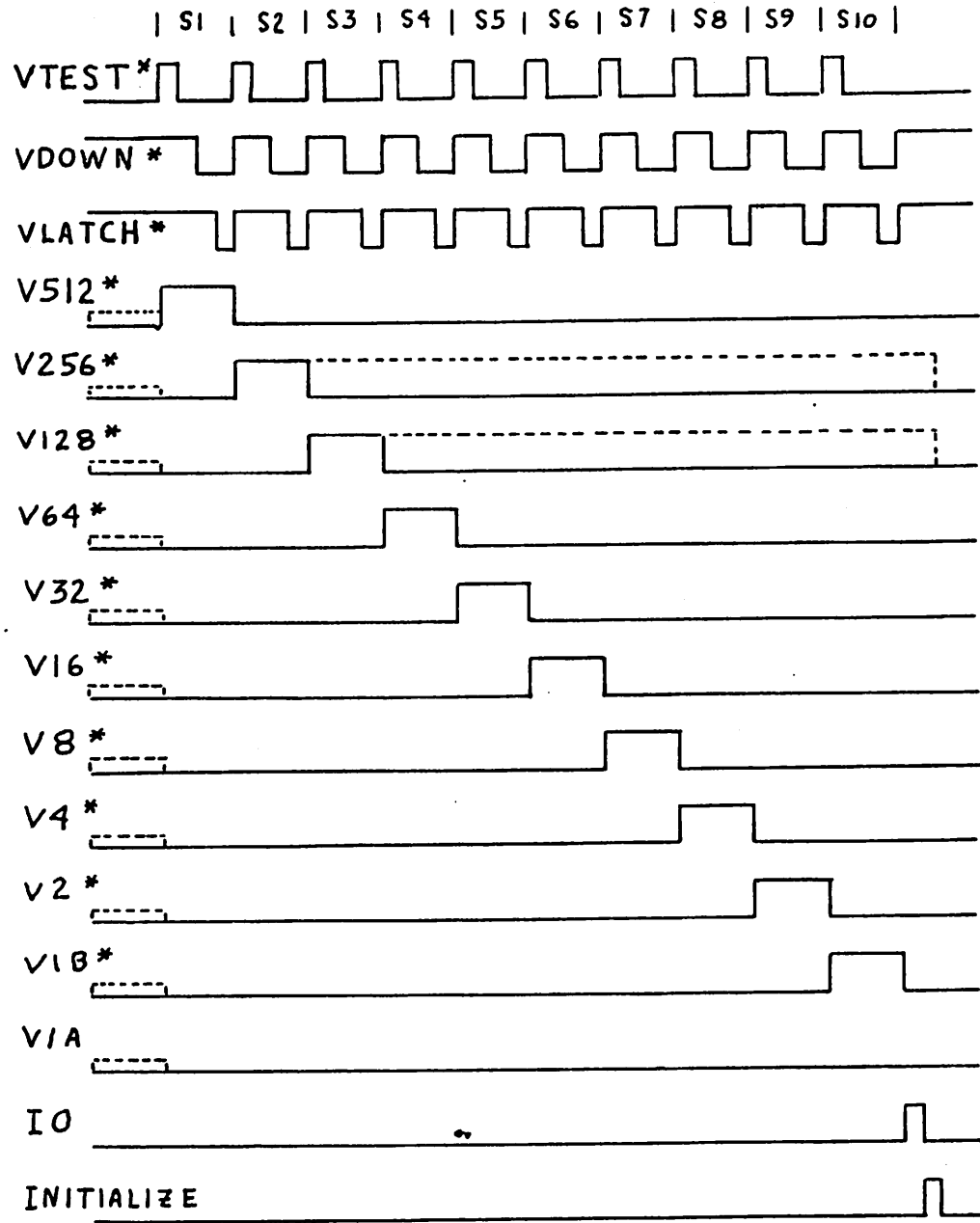


Figure B.2: Timing diagram for signals to experimental I.C.



Solid lines ( $V_{512}^* - V_{1A}^*$ ) are for  $V_{IN} = 0$   
dotted lines are for  $V_{IN} = 3.75v$ .

Figure B.3: An expanded timing diagram.

## STATE TABLE

RESET  $\Rightarrow$  CLEAR: ST1, ST2, VSAVE, VGIN, VTEST,  
 V512, V256... V18, INITIALIZE, IO  
 PRESET: VDOWN, VLATCH

STATE:

	L0	RESET
	L0	
	L0	START ; JVSAVE ; JVGIN ; JVSET
	L0	RVSAVE = $\overline{VSET}$
	L0	RVGIN = $\overline{VSAVE}$ ; JV512 ; JVTEST ; $\overline{VSAVE} \cdot VGIN \rightarrow (L1, S1)$
↑	L1	RVTEST
	L1	RVDOWN ; $\overline{VTEST} \rightarrow (L2)$
S1	L2	RVLATCH
↑	L3	RV512 = $\overline{COMP}$ ; JV256 ; JVDOWN ; JVLATCH ; JVTEST
↑	L1	RVTEST
	L1	RVDOWN ; $\overline{VTEST} \rightarrow (L2)$
S2	L2	RVLATCH
↑	L3	RV256 = $\overline{COMP}$ ; JV128 ; JVDOWN ; JVLATCH ; JVTEST
↑	L1	RVTEST
	L1	RVDOWN ; $\overline{VTEST} \rightarrow (L2)$
S3	L2	RVLATCH
↑	L3	RV128 = $\overline{COMP}$ ; JV64 ; JVDOWN ; JVLATCH ; JVTEST
	:	:
	:	:
	:	:
↑	L3	RV2 = $\overline{COMP}$ ; JV18 ; JVDOWN ; JVLATCH ; JVTEST
↑	L1	RVTEST
	L1	RVDOWN ; $\overline{VTEST} \rightarrow (L2)$
S10	L2	RVLATCH
↓	L3	RV18 = $\overline{COMP}$ ; JVDOWN ; JVLATCH ; JIO ; $L3 \cdot \overline{S10} \rightarrow (L0)$
	L0	JINITIALIZE = IO ; CLKBUF = IO
	L0	(CLEAR V512-V18) = INITIALIZE ; JSTART = INITIALIZE
	L0	START ...
	:	:

Figure B.4: The state table for the digital logic circuit.

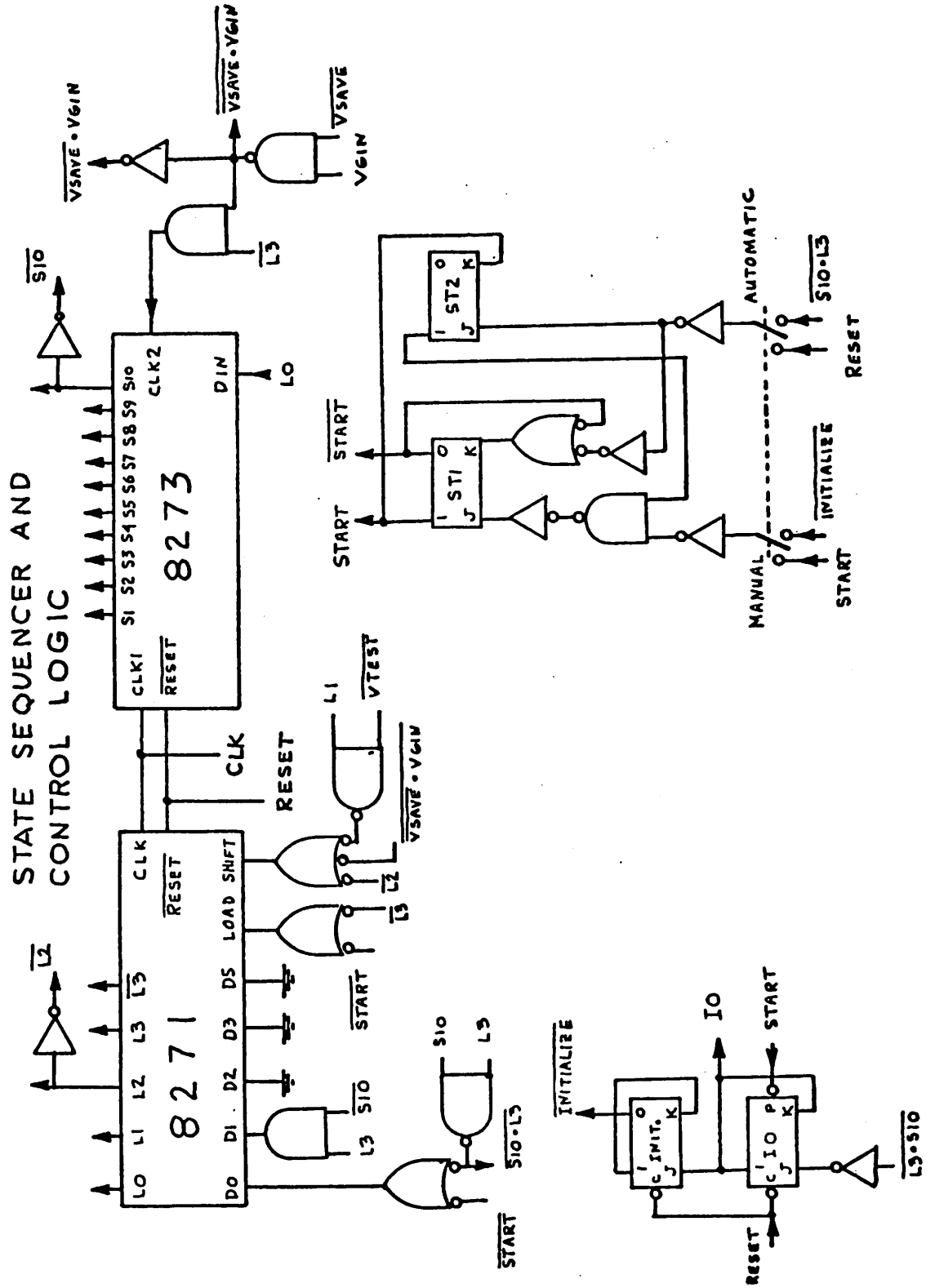


Figure B.5: The logic diagram of the state sequencer and control logic.

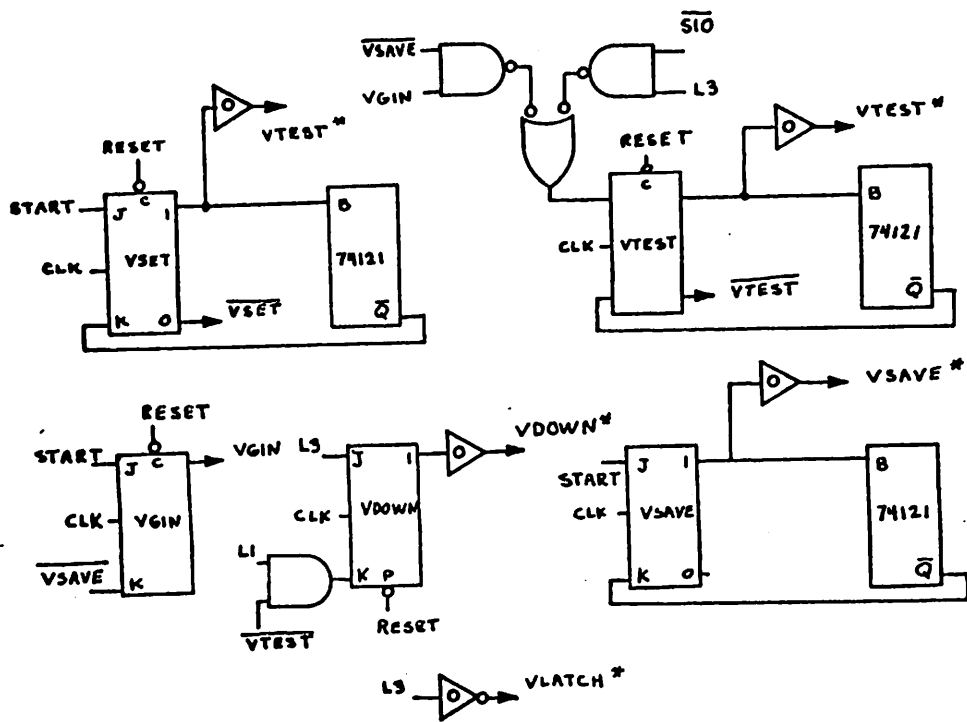


Figure B.6: The switch signal generator.

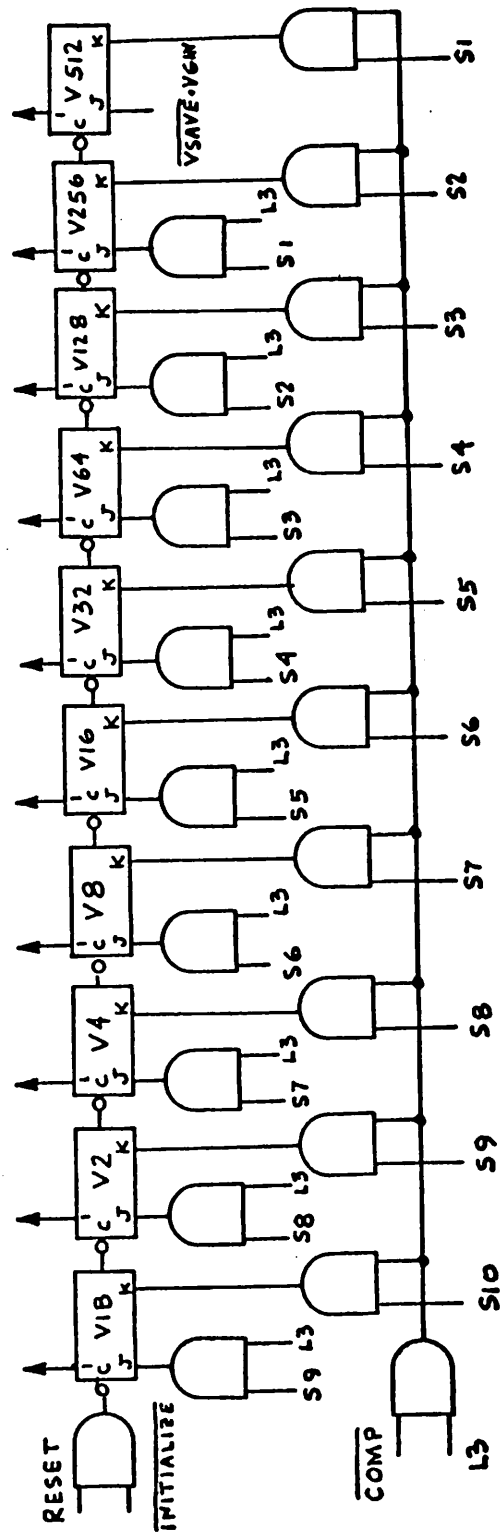


Figure B.7: The capacitor signal generator.

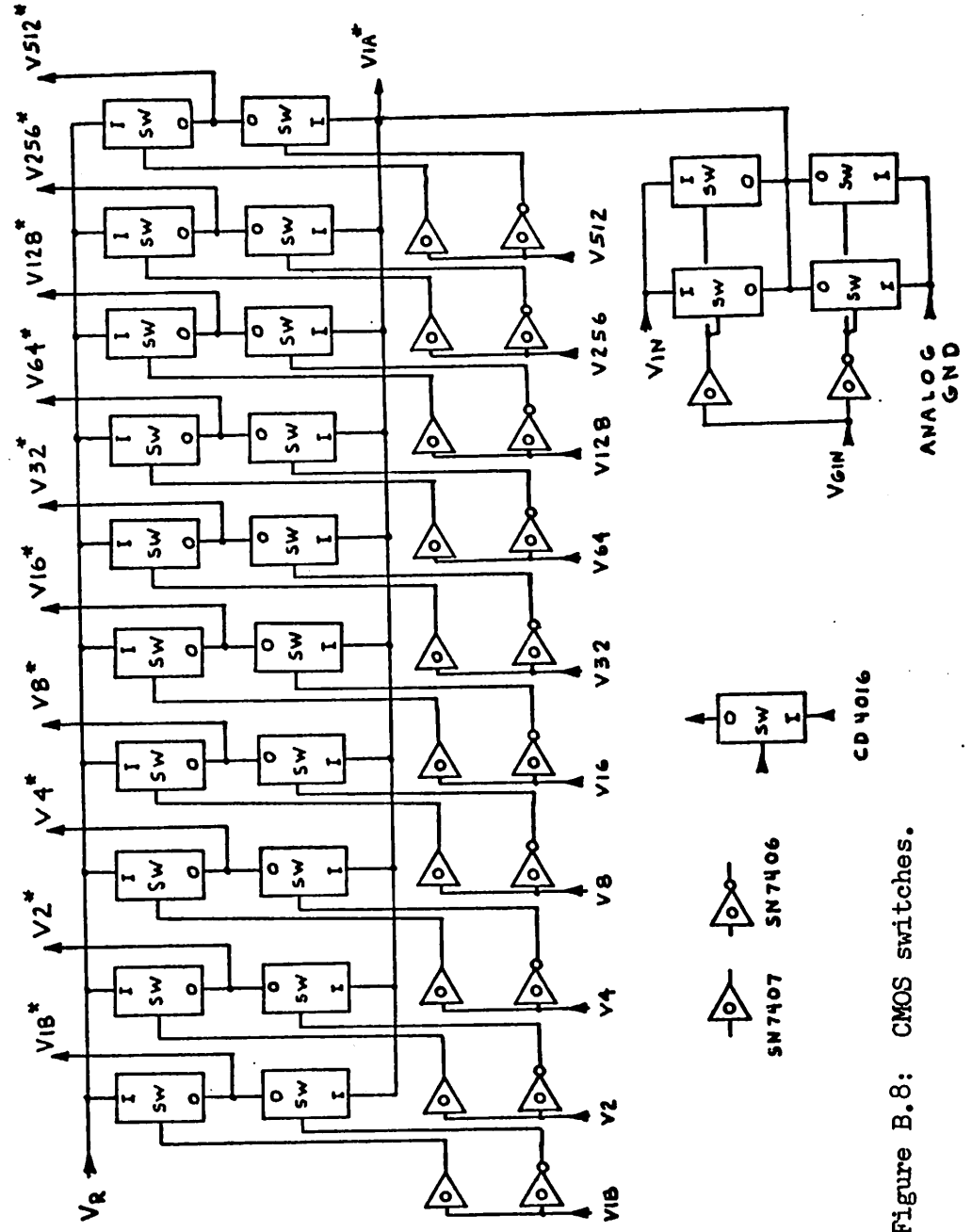


Figure B.8: CMOS switches.

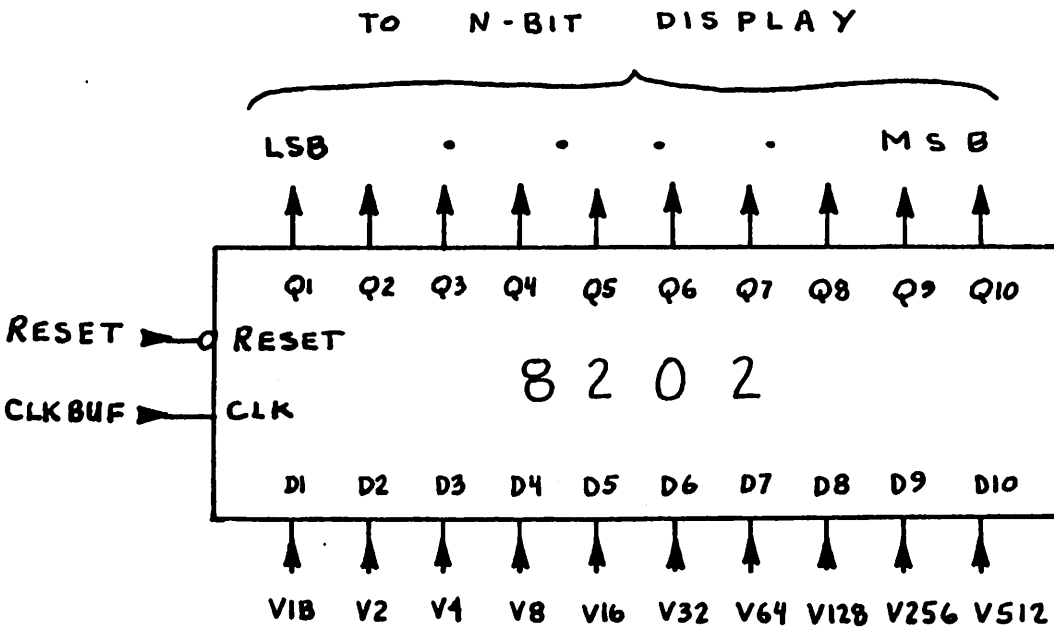


Figure B.9: The output buffer.



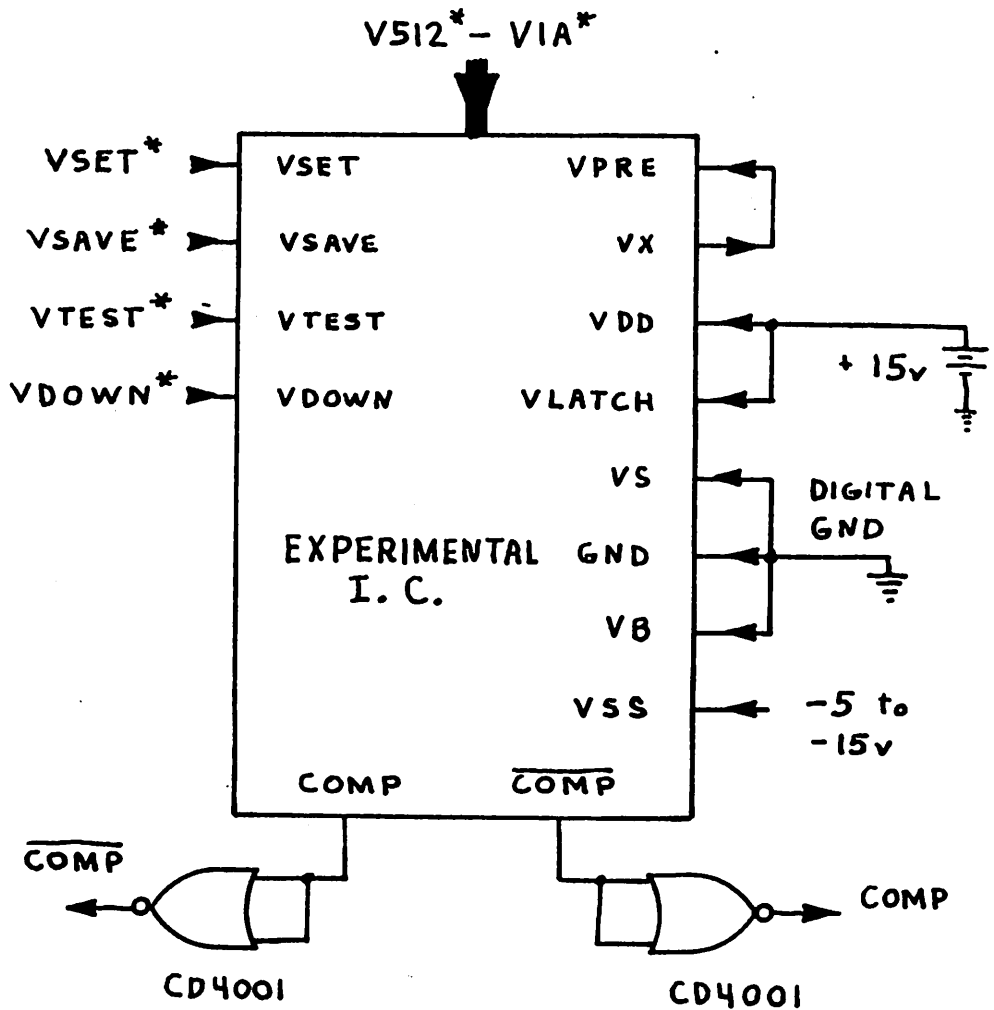


Figure B.10: The minimal operating configuration for the experimental chip.

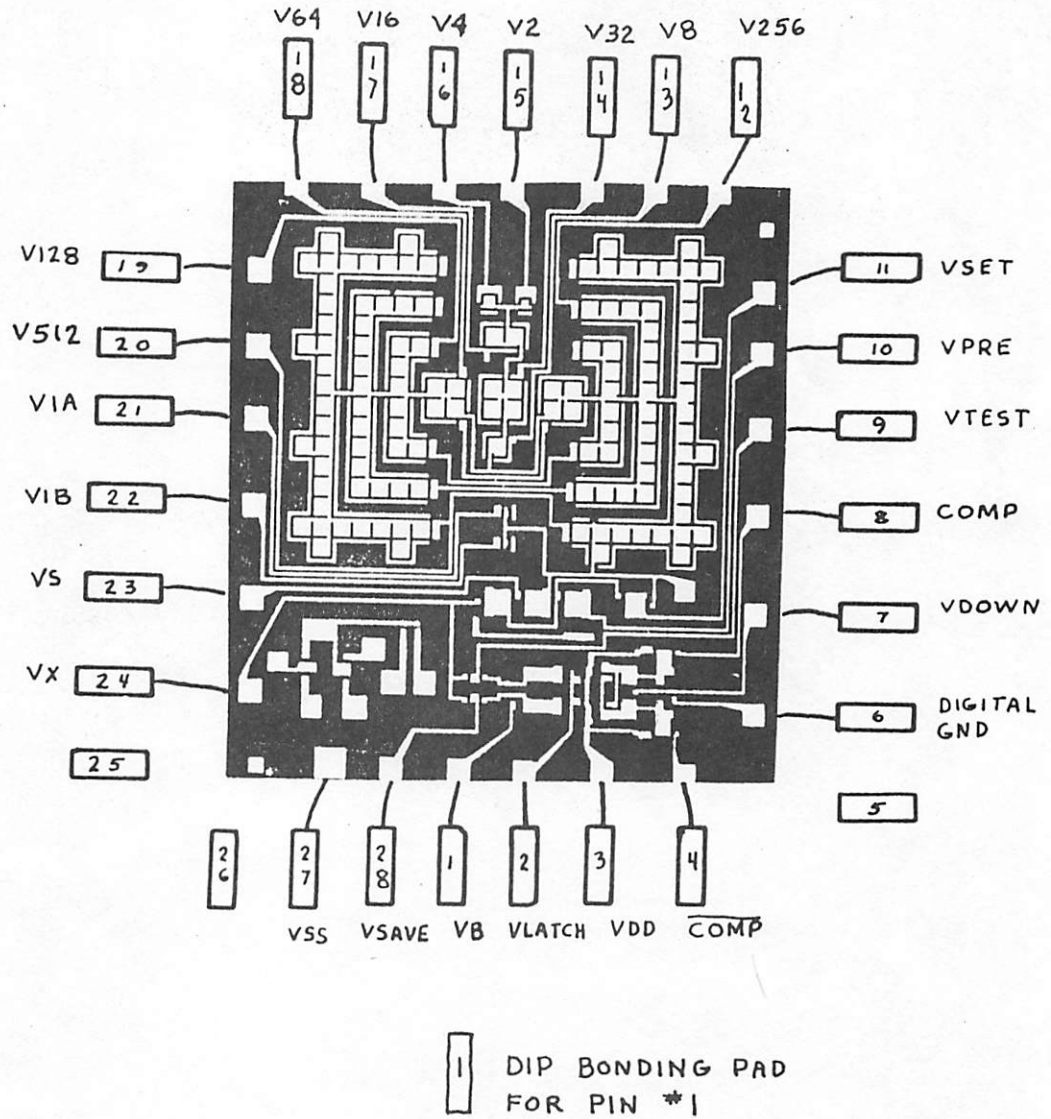


Figure B.11: 28-pin DIP bonding diagram.

## APPENDIX C

## N-MOS ALUMINUM GATE FABRICATION PROCESS

1. p-Type 100 substrate, 3-5  $\Omega$ -cm
2. Clean
  - DI: HF (9:1), dip
  - TCE, 60°C 10 min.
  - Acetone, 2 min.
  - DI rinse
  - RCA Clean:
    - RCAI: 1.  $\text{NH}_4\text{OH}$ :  $\text{H}_2\text{O}_2$ : DI (1:1:5)  
75°C 15 min.
      2. DI Rinse
    - RCAII: 1. HCl:  $\text{H}_2\text{O}_2$ : DI (1:1:6)  
75°C 15 min.
      2. DI Rinse
      3.  $\text{N}_2$  Dry
3. Initial Oxidation; Initial Oxidation Furnace at 1150°C; 0.92  $\mu$  wet oxide;
  - wet  $\text{O}_2$  0.5 l/min. 90 min.
  - $\text{N}_2$  0.65 l/min. 10 min.
4. PR (Photoresist) Step;  $\text{p}^+$  isolation diffusion mask;
  - Kodak 747 (Micro neg) resist; 50 c.s.; 5000 rpm; 30 sec.
  - Air dry 10 min.
  - Prebake; 90°C, 30 min.
  - Expose 3 sec.
  - Spray develop; 30 sec.

- Spray rinse; 20 sec.
  - Postbake; 125°C for 30 min.
  - Oxide etch;  $\text{NH}_4\text{F}$ :  $\text{HF}$  (5:1);  
9.5 min for .92  $\mu$  at .1  $\mu$ /min
  - Strip PR;  $\text{H}_2\text{SO}_4$ :  $\text{H}_2\text{O}_2$  (4:1); 90°C; 5 min.
5. RCA (RCAI and RCAII as in Step 2)
  6.  $\text{p}^+$  Predeposit; Boron Predeposition Furnace @ 950 °C
    - $\text{B}_2\text{H}_6$  0.26  $\ell$ /min
    - $\text{O}_2$  13 cc/min 15 min.
    - $\text{N}_2$  1.3  $\ell$ /min
  7. Cleaning
    - Etch boron glass 12% HF dip (HF:DI, 1:3)
    - RCAII
  8. Oxide over  $\text{p}^+$ ; p-type drive-in Furnace @ 1150°C;
    - grow .4  $\mu$  wet  $\text{O}_2$ ;
    - wet  $\text{O}_2$  0.5  $\ell$ /min 16 min
    - $\text{N}_2$  1.0  $\ell$ /min 10 min
  9. PR step;  $\text{N}^+$  mask; .95  $\mu$  (9.5 min etch)
    - same as in Step 4
  10. Cleaning
    - RCA
  11.  $\text{N}^+$  Predeposit; Phosphorous Predeposition furnace @ 1100°C
    - $\text{O}_2$  0.1  $\ell$ /min } 5 min
    - $\text{N}_2$  1.25  $\ell$ /min }
    - $\text{O}_2$  0.1  $\ell$ /min } 20 min
    - $\text{N}_2$  1.25  $\ell$ /min }
    - $\text{N}_2/\text{POCl}_3$  96 cc/min }

- O<sub>2</sub>      0.1    ℓ/min                      2 min
- N<sub>2</sub>      1.25   ℓ/min

12. Cleaning

- Etch phosphorous glass; 12% HF dip
- DI rinse
- RCAII

13. Oxide over N<sup>+</sup>; N-type Drive-in furnace @ 1100°C; wet O<sub>2</sub>; 0.5 μ;

- wet O<sub>2</sub>      0.5 ℓ/min                      34 min
- N<sub>2</sub>            1.0 ℓ/min                      10 min

14. PR step; gate oxide mask;

.99 μ doped oxide (6.5 min. etch)

15. Cleaning

- RCA

16. Grow gate oxide; N-type drive-in furnace @ 1000°C; dry O<sub>2</sub>; .1 μ;  
wafer horizontal on boat

- O<sub>2</sub>    1.5 ℓ/min                      110 min total time

rotation	Δt
0°	10 min
180°	22 min
90°	33 min
270°	45 min

- N<sub>2</sub>    1.0 ℓ/min                      5 min

17. PR Step; contact window mask;

0.1 μ (1.5 min. etch)

- same as step 4 except mask is double exposed for 2.5 sec. each  
after mask shifted 1 row. (This technique avoids pinholes.)

18. Clean
  - RCAII
19. Aluminum Evaporation; electron beam vacuum chamber;
  - IR Lamp drying 15 min.
  - Evaporation; .3  $\mu$  to .4  $\mu$  Aluminum
20. PR Step; AZ1350J resist; Metallization Mask;
  - IR Lamp heating 10 min
  - Spin coat AZ1350J at 8000 rpm; 30 sec;
  - 10 min air dry
  - Prebake 80°C for 45 min with air circulation
  - Expose 12 sec.
  - Spray develop MF312: DI (1:1); 45 sec.
  - DI Rinse
  - Postbake 20 min; 90°C
  - Aluminum etchant type A; 40°C; ultrasonic agitation;  
(~ 45 sec)
  - 1112 PR Stripper; 50°C; 2 min.
  - DI Rinse, N<sub>2</sub> dry
21. Clean
  - TCE dip
  - Acetone dip
22. Heat treatment; Sintering Oven; 200°C; 5 min
  - N<sub>2</sub>: H<sub>2</sub> (9:1) - forming gas; 1 l/min.

## APPENDIX D

Calculation of Capacitor Plate Duplication Size Needed for Undercut Insensitivity

The effect of uniform undercut upon capacitor ratios will now be computed. This analysis will determine the duplication geometry needed for the larger capacitors. Let  $S$  be the design value of side length of a square plate representing the smallest capacitor  $C_1$ . The uniform undercut ( $u$ ) corresponds to a reduction in the edge length of each capacitor. The general approach used to determine the worst case deviation (WCD) in LSB for the entire array is outlined below:

1. Determine total area  $A_T$ .
2. Determine the resultant binary weight of the  $i$ th capacitor:

$$B_i = \frac{A_i}{A_T} 2^N.$$

3. Determine the deviation from ideal value:

$$\text{Dev}_i = B_i - 2^i$$

$$\text{for } N-1 \geq i.$$

4. Estimate  $\text{WCD} = \text{Dev}_N$ .

For an array having no duplication of smaller plates used in large capacitor construction, the total area is given by:

$$A_T(N,s,u) = (s-u)^2 + (s-u)^2 + (\sqrt{2} s-u)^2 \\ + (\sqrt{4} s-u)^2 + \dots + (\sqrt{2^{N-1}} s-u)^2.$$

However, it has been shown in Chapter IV that it is necessary to design the larger capacitors by paralleling smaller plates of identical geometry in order to achieve insensitivity to uniform undercut. Let the  $k$ th capacitor corresponding to binary weight  $2^k$  be used as the duplication unit. Hence the next largest capacitor would have binary weight  $2(2^k)$  independent of undercut relative to  $2^k$ . Using this approach the total capacitor area is:

$$\begin{aligned}
 A_T(N,k,s,u) &= (s-u)^2 + (s-u)^2 + (\sqrt{2} s-u)^2 \\
 &+ \dots + (\sqrt{2^k} s-u)^2 + 2(\sqrt{2^k} s-u)^2 \\
 &+ 4 (\sqrt{2^k} s-u)^2 + \dots + 2^{N-1-k} (\sqrt{2^k} s-u)^2, \\
 &= 2^N s^2 - 2us[2 + \sqrt{2} + \sqrt{4} + \dots + \sqrt{2^k} (2^{N-k}-1)] \\
 &+ u^2[2^{N-k} + k].
 \end{aligned}$$

The area of a large capacitor of weight  $i$  for  $N-1 \leq i \leq k$  is:

$$A_i(N,k,s,u) = 2^{i-k} (\sqrt{2^k} s-u)^2.$$

The actual binary weight of this capacitor is:

$$B_i(N,k,s,u) = \frac{A_i 2^N}{A_T}$$

and neglecting the terms containing  $u^2$  since  $u \ll s$  then

$$B_i = 2^i + \frac{u}{s} 2^i \left( \frac{2 + \sqrt{2} + \sqrt{4} + \dots + 2^k (2^{N-k}-1)}{2^{N-1}} - 2^{1 - \frac{k}{2}} \right).$$

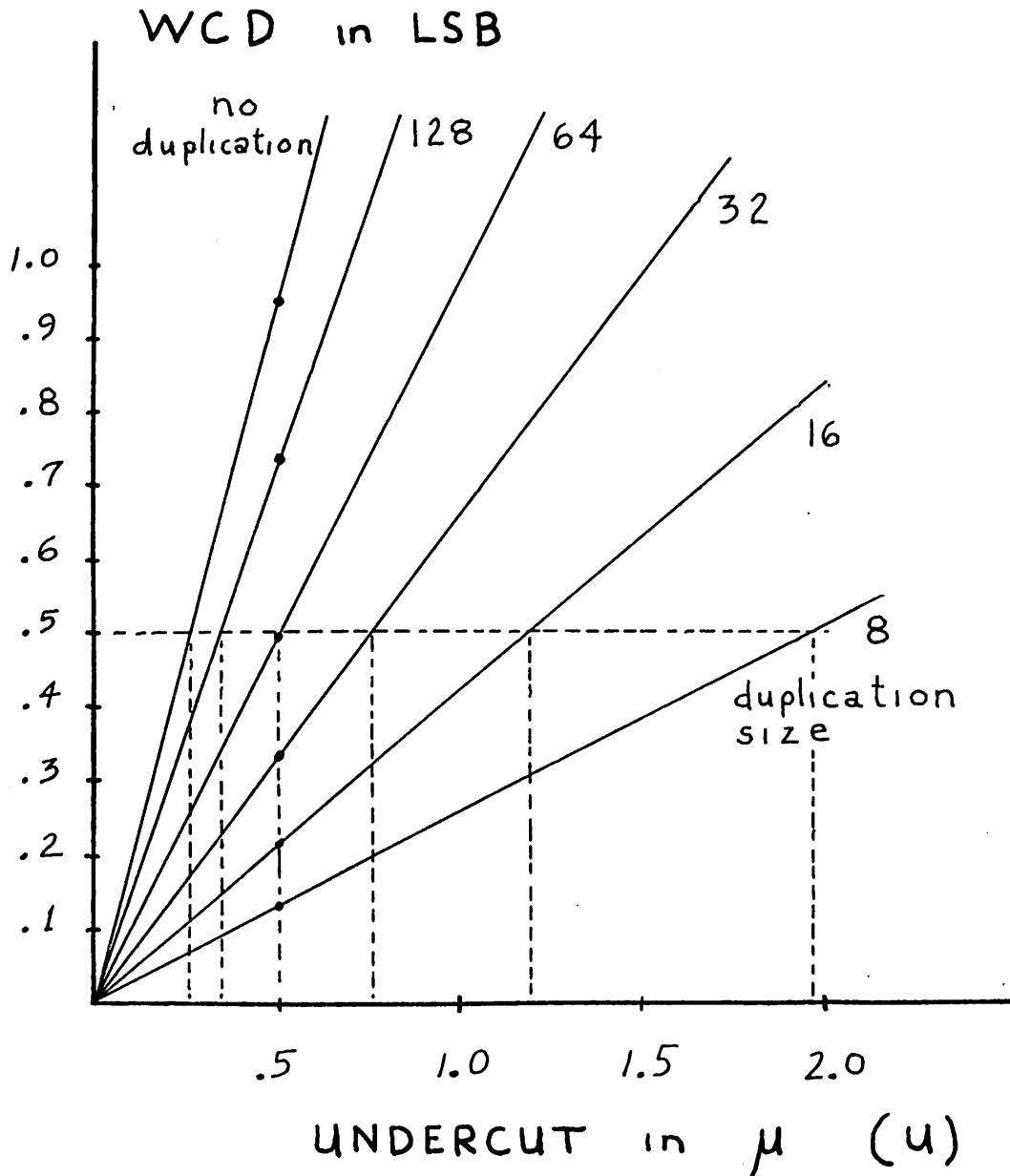


The WCD for the entire array is approximately equal to the second term in this equation evaluated for  $i = N$  and  $1 < k < N-1$

$$\text{WCD}(N,k,s,u) = \frac{2u}{s} \left[ 2 + \sqrt{2} + \sqrt{4} + \dots + \sqrt{2^{k-1}} - \sqrt{2^k} \right].$$

An interesting observation is that although the absolute error becomes smaller as  $N$  becomes larger the WCD in LSB is approximately independent of  $N$  provided that  $N$  is large.

The plot of WCD ( $k,u$ ) versus  $u$  for nominal values of  $N = 10$  bits and  $s = 20 \mu$  is given in Figure D.1. From this graph, an array structure having duplication of capacitor size  $C8$  (for which  $k = 3$  and binary weight equals 8) will retain  $\pm .5$  LSB ratio accuracy in spite of  $2 \mu$  undercut.



Duplication Size	Allowable $u$ for $\pm .5$ LSB
none	.27 $\mu$
128	.35 $\mu$
64	.50 $\mu$
32	.76 $\mu$
16	1.20 $\mu$
8	1.97 $\mu$

Figure D.1: The worst case deviation from linearity (WCD) as a function of duplication size ( $k$ ) and uniform undercut ( $u$ ).

## REFERENCES

1. J.L. McCreary and P.R. Gray, "A High Speed, All-MOS, Successive Approximation, Weighted Capacitor A/D Conversion Technique," ISSCC Digest of Technical Papers, pp. 38-39, Feb. 1975.
2. J. Albarran, Univ. of CA., Berkeley, CA., Private Communications.
3. "Engineering Product Handbook - A/D and D/A Converters," DATEL Systems, Inc., pp. 1-41, 1974.
4. K. Fukahori, "All MOS A/D Converter," M.S. Plan II Report, University of California, Berkeley, CA., 1974.
5. D.F. Hoeschele, Analog-to-Digital/Digital-to-Analog Conversion Techniques, Wiley & Sons, New York, 1968.
6. R.E. Suarez, P.R. Gray, and D.A. Hodges, "An All-MOS, Charge Redistribution, Successive Approximation A/D Conversion Technique," ISSCC Digest of Technical Papers, pp. 194-195, Feb. 1974.
7. G. Smarardoiu and D.A. Hodges, "An All-MOS Analog-to-Digital Converter Using a Constant Slope Approach," European Solid-State Circuits Conference, Canterbury, U.K., IEEE Conf. Pub. 130, pp. 60-61, Sep. 1975.
8. D.J. Dooley, "A Complete Monolithic 10-b D/A Converter," IEEE J. Solid-State Circuits, Vol. SC-8, pp. 404-408, Dec. 1973.
9. G. Kelson, H.H. Stellrecht, and D.S. Perloff, "A Monolithic 10-b Digital-to-Analog Converter Using Ion Implantation", IEEE J. Solid-State Circuits, Vol. SC-8, pp. 396-403, Dec. 1973.
10. Analog-Digital Conversion Handbook, Engr. Staff of Analog Devices Inc., 6th Ed., 1972.
11. "Analogic Announces One-Chip Voltmeter," Electronic Engineering Times, pp. 2, Jan. 1, 1975.

12. J.A. Schoeff, "A Monolithic Analog Subsystem for High-Accuracy A/D Conversion," ISSCC Digest of Technical Papers, pp. 18-19, Feb. 1973.
13. "10-bit CMOS A/D Converter Chip Interfaces with LSI Microprocessors," Electronic Design 2, pp. 81-82, Jan 18, 1975.
14. H. Schmid, Electronic Analog/Digital Conversions, Van Nostrand-Reinhold, New York, 1970.
15. J.L. McCreary and P.R. Gray, "All-MOS A/D Conversion Techniques, Part I," IEEE J. Solid-State Circuits, Dec. 1975.
16. P.R. Gray, Univ. of CA., Berkeley, CA., Private Communication.
17. Ibid.
18. Ibid.
19. "Capacitance and Capacitors," Electronics, pp. 61-66, May 11, 1962.
20. G. Kelson, et al., op. cit.
21. P. Greiff, "Temperature Coefficient of Diffused Resistors," IEEE Proceedings (Corresp.), Vol. 53, pp. 215-216, Feb. 1965.
22. Integrated Silicon Device Technology--Capacitance, Research Triangle Institute, Technical Documentary Report No. ASD-TDR-63-316, Vol. 2, Oct. 1963.
23. A.S. Grove, Physics and Technology of Semiconductor Devices, Wiley and Sons, New York, 1967.
24. P.R. Gray, op. cit.
25. R.E. Suarez, "Analog-to-Digital Conversion in MOS Integrated Circuits," Ph.D. Thesis, University of California, Berkeley, 1975.
26. J.G. Simmons and G.W. Taylor, "Dielectric Relaxation and its Effect on the Isothermal Electrical Characteristics of Defect Insulators," Physical Review B, Vol. 6, No. 12, pp. 4793-4803, Dec. 15, 1972.

27. P.J. Burkhardt, "Dielectric Relaxation in Thermally Grown  $\text{SiO}_2$  Films," IEEE Trans. on Electron Devices, Vol. 13, pp. 268-275, Feb. 1966.
28. R.J. Kriegler and R. Bartnikas, "Dielectric Relaxation in Si -  $\text{SiO}_2$  -  $\text{C}_r$  Structures," IEEE Trans. on Electron Devices (Corresp.), pp. 1010-1011, Nov. 1970.
29. P. Richman, MOS Field-Effect Transistors and Integrated Circuits, Wiley & Sons, New York, 1973.
30. L. Vadasz and A.S. Grove, "Temperature Dependence of MOS Transistor Characteristics Below Saturation," IEEE Trans. on Electron Devices, Vol. 13, pp. 863-866, Dec. 1966.
31. W.M. Penney and L. Lau, Editors, MOS Integrated Circuits, Van Nostrand-Reinhold, New York, 1972.
32. R.E. Suarez, Univ. of CA., Berkeley, CA., private communication.
33. C.H. Sequin, "Fringe Field Corrections for Capacitors on Thin Dielectric Layers," Solid State Electronics, Vol. 14, pp. 417-420, 1971.
34. P. Richman, Characteristics and Operation of MOS Field-Effect Devices, McGraw-Hill, New York, 1967.
35. P.E. Gray and C.L. Searle, Electronics Principles, Physics, Models, and Circuits, Wiley and Sons, New York, pp. 898-904, 1969.
36. B.C. Young, "N-Channel Silicon-Gate MOS Transistor Fabrication Process," M.S. Thesis, University of California, Berkeley, 1974.
37. M. Chen and J.W. Hile, "Oxide Charge Reduction by Chemical Gettering with Trichloroethylene During Thermal Oxidation of Silicon," J. Electrochem. Soc., Vol. 119, pp. 223-226, Feb. 1972.

38. K. Hirabayashi and J. Iwamura, "Kinetics of Thermal Growth of HCl Oxides on Silicon," J. Electrochem. Soc., Vol. 120, No. 11, pp. 1597-1601, Nov. 1973.
39. D.A. Hodges, Univ. of CA., Berkeley, CA., private communication.
40. G. Schottky, "Decrease of FET Threshold Voltage Due to Boron Depletion During Thermal Oxidation," Solid State Electronics, Pergamon Press, Vol. 14, pp. 467-474, 1971.
41. D.A. Hodges, op. cit.