

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

PERIODIC PROGRESS REPORT

MONOLITHIC PARALLEL PROCESSOR

(28 JUNE 1968 TO 27 DECEMBER 1968)

CONTRACT NO. NAS5-11577

Prepared by

RCA ELECTRONIC COMPONENTS
Somerville, New Jersey

for

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

N69-29287	(THRU)	(CODE)	08
(ACCESSION NUMBER)	100	(PAGES)	08
Q# 101566	(NASA CR OR TRX OR AD NUMBER)		

PERIODIC PROGRESS REPORT

MONOLITHIC PARALLEL PROCESSOR
(28 JUNE 1968 TO 27 DECEMBER 1968)

CONTRACT NO. NAS5-11577

GODDARD SPACE FLIGHT CENTER
CONTRACTING OFFICE: A.L. ESSEX
TECHNICAL MONITOR: R.J. LESNIEWSKI

PREPARED BY

RCA ELECTRONIC COMPONENTS
Somerville, New Jersey

PROJECT MANAGER: J. HILIBRAND
PROJECT ENGINEER: A. DINGWALL

for

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

ABSTRACT

A four-bit parallel processor has been designed. The logic design conforms to the contract objectives. The array of 750 devices has been laid out on a chip 145 mils by 155 mils. A composite drawing was completed and rubyliths were prepared. Extensive checking of the rubylith (with the assistance of R. Lesniewski of NASA) was used to reduce the probability of error in the layout. A test matrix was developed to permit testing of the processing and control portions of the parallel processor.

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
I	SUMMARY OF PROGRAM STATUS	1
	A. Introduction	1
	B. Program Objectives	2
	C. Program Status	4
II	PARALLEL PROCESSOR LOGIC DESIGN	5
	A. General Description of Four-Stage Processor . . .	5
	B. General Logic Description	5
	C. Operational Modes	15
	D. Overflow Detection	15
	E. Zero Detection	17
	F. Negative Detection	17
	G. Conditional Operation	18
	H. Instruction Repertoire	18
	I. Serial-Shift Operations	19
	J. Parallel Commands	20
	K. Timing	23
	L. Mode-Independent Switches	24
	M. Mode-Dependent Switches	26
	N. Shift Switches	27
	O. Conditional Operation	27
	P. Overflow Indicator	28

TABLE OF CONTENTS (Cont.)

<u>Section</u>		<u>Page</u>
	Q. Expansion to 16-Stage Processor	29
III	BREADBOARD AND COMPUTER STUDIES	31
	A. Summary	31
	B. Details	31
	C. Computer Studies of Switching Speed	34
IV	COS/MOS FUNCTIONAL LOGIC GATING	47
	A. General Discussion	47
	B. Translation Procedure	50
V	DESIGN AND LAYOUT	57
	A. Introduction	57
	B. Layout Design Considerations and Implementa- tion	60
	C. EXCLUSIVE OR	62
VI	PROCESSING	75
	A. Introduction	75
	B. COS/MOS Fabrication Process	76
	C. Capacitance-Voltage Curves	78
VII	TESTING	81
	A. Introduction	81
	B. Initial Design Fault Isolation	81
	C. Production Test Procedures	81
	D. Parallel Processor Test Procedure	85

TABLE OF CONTENTS (Cont.)

<u>Section</u>		<u>Page</u>
VIII	CONCLUSIONS AND RECOMMENDATIONS	91
IX	PROGRAM FOR NEXT INTERVAL	93

PRECEDING PAGE BLANK NOT FILMED.

LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1	Four-Stage Processor, Lead Requirements	6
2	Logic Diagram	7
3	Implementation of Basic Gates in the Array Design Logic	8
4	Operational Modes	16
5	Schematic Interconnection of Four Four-Stage Chips. .	30
6	Breadboard of the Processor	32
7	Construction of Functional Gates Using the CD4007D as the Basic Building Block	33
8	Input and Output Waveforms for Control Section . . .	35
9	Zero Indicator	36
10	Clock Circuit	37
11	Typical Operating Waveforms	38
12	Example of Form Used to Enter Data Into the Computer	41
13	Computer Calculated Switching Times	45
14	Building Block	48
15	Functional Gating	49
16	Layout of the Parallel Processor	58
17	Lead Configuration for the Four-Bit Slice Using 28- Lead Flat Pack	59
18	Parallel Processor Functional Layout	61

LIST OF ILLUSTRATIONS (Cont.)

<u>Figure</u>	<u>Title</u>	<u>Page</u>
19	A Functional Logic Block: D Flip Flop	63
20	D Flip-Flop Logic Schematic	64
21	Transmission Gate and Inverter Circuit Schematic . . .	65
22	Transmission Gate Layout	66
23	Block Diagram Representation for Layout Development of D Flip Flop	67
24	D Flip-Flop Composite Layout	68
25	Block Diagram for EXCLUSIVE-OR Layout Development . . .	72
26	EXCLUSIVE-OR Composite Layout	73
27	COS/MOS Fabrication Process	77
28	Capacitance-Voltage Curves for Clean and Impure Oxides	79
29	Test Requirement for Three-Input NAND Gate with Inputs Active High	83
30	Test Requirement for Function Gate	84

SECTION I
SUMMARY OF PROGRAM STATUS

A. INTRODUCTION

This first periodic report describes a program whose objective is to design, develop, and fabricate monolithic, complementary-symmetry MOS (COS/MOS), large-scale, parallel processor arrays. The overall program is divided into a 12-month phase and a six-month phase. The work effort to be performed during these two phases is as follows: the fabrication of monolithic, four-bit parallel processor arrays of more than 750 components, i.e., MOS transistors and diodes; and the fabrication of monolithic, 16-bit parallel processor arrays of more than 3000-component complexity. The 16-bit parallel processors are to be fabricated by interconnecting clusters of four-stage arrays on one chip.

The COS/MOS technology to be applied to these parallel processor devices, although relatively new, is well suited to large-scale integration because of the minimal area requirements, low power dissipation, and suitability for functional logic gating. Of special advantage for remote applications are: essentially negligible standby power dissipation (the design goals call for less than 1.2×10^{-5} watts standby power at ambient), and low operational power at significant operational speed (centered design goals are 10×10^{-3} watts at a data transfer rate of 4×10^{-6} seconds).

Successful achievement of the fabrication goals of this program will represent a significant advance in the COS/MOS state of the art. RCA has applied considerable corporate effort to the development of COS/MOS technology. Since August 1967 when the commercial availability of the TA5361 dual three-input NOR gate was first announced, a family of basic integrated circuits has been made available commercially including dual D flip flops, inverters, transmission gates, quad gates, and dual gates. In addition, RCA has available MSI

arrays such as 16-bit memory cells, seven-stage counters, and 18-stage shift registers. RCA's demonstrated experience in COS/MOS design and large-scale production is currently unmatched as evidenced by the fact that RCA is still the sole commercial source for these products. The four-bit and 16-bit parallel processors, however, represent devices having a complexity level more than an order-of-magnitude greater than the COS/MOS devices available commercially.

B. PROGRAM OBJECTIVES

The parallel processor program consists of two distinct phases. The objective of the first phase is to design, develop, and fabricate individual four-bit parallel processor chips. The objective of the second phase is to develop and employ high-yield processing techniques so that monolithic clusters of four good four-bit parallel processor arrays will be found with reasonable yield on silicon wafers. These clusters then are to be interconnected monolithically and packaged to form the 16-bit parallel processor arrays.

Detailed discussions of logic objectives of the parallel processor arrays are given in Section II. In addition to the logic requirements, the specific design features and objectives are as follows:

- a. Size - a monolithic four-stage array approximately 0.145 inch by 0.155 inch.
- b. Power - supply voltage +8 to +12 volts.
- c. Standby power dissipation at ambient - less than 1.2×10^{-5} watts. Operating power 10×10^{-3} watts at a data transfer rate of 4×10^{-6} seconds at ambient with a supply voltage of +10 volts and with 50 picofarads on each of the output lines.
- d. The circuit will operate over the temperature range: $+100^{\circ}\text{C}$ to -50°C .
- e. A "0" is represented by ground potential and a "1" by the supply potential.
- f. The complete circuit will require 27 leads, and will be mounted in a 28-lead ceramic flat pack.

- g. All leads except power leads have a capacitive load not greater than 5 picofarads, including package capacitance.
- h. A decoded command alone will not effect a data transfer. A general transfer (timing) pulse with a decoded command will effect a data transfer.
- i. (1) Each output will be able to drive a 50-picofarad load to within 90 percent of the supply voltage for an output change from "0" to "1" in 4×10^{-6} seconds from initiation of the "Out" instruction.
 (2) Each output will be able to drive a 50-picofarad load to within 10 percent of the supply voltage for an output change from "1" to "0" in 4×10^{-6} seconds from initiation of the "Out" instruction.
- j. Complete execution of non-carry generating instruction other than "Out" will take no longer than 2×10^{-6} seconds from the application of the instruction at ambient with a supply voltage of +10 volts.
- k. Complete execution of a worst case four-bit parallel addition shall take no more than 2×10^{-6} seconds from application of the add instruction at ambient with a supply voltage of +10 volts.
- l. The parallel processor shall be capable of operating in one of four modes, which will be controlled by two lines.
- m. The register should require two timing pulses, one nested in the other; the smaller pulse will be one-half the larger. The circuit is capable of operating with the width of the larger pulse between 2×10^{-6} seconds and 1×10^{-3} seconds.

Design objectives for the 16-bit parallel processor generally are similar to those of the four-bit parallel processor. The following changes, however, result from cascading four units in a single package.

- a. Standby power dissipation at ambient - less than 5×10^{-5} watts.
 Operating power of 40×10^{-3} watts at a data-transfer rate of

4×10^{-6} seconds at ambient, with a supply voltage of +10 volts and 50 picofarads on each output line.

- b. The complete circuit will require approximately 40 leads.
- c. Complete execution of a worst-case 16-bit parallel addition will take no longer than 6×10^{-6} seconds from application of the add instruction at ambient with a supply voltage of 10 volts.

C. PROGRAM STATUS

The logic design, the layout, and the cutting of rubyliths were completed for the parallel processor arrays during this reporting period. Current effort is directed towards final checking and correction of the rubyliths so that wafer fabrication can commence. Tasks still to be completed include high-yield processing of parallel processor wafers, testing, debugging (if required), and packaging of chips.

SECTION II

PARALLEL PROCESSOR LOGIC DESIGN

A. GENERAL DESCRIPTION OF FOUR-STAGE PROCESSOR

The four-stage parallel processor basically is a four-stage shift register that has both serial and parallel access. The logic associated with the register allows parallel-two's complement addition, AND, OR, and EXCLUSIVE OR logic operations; and right, left, or right-cyclic shifts.

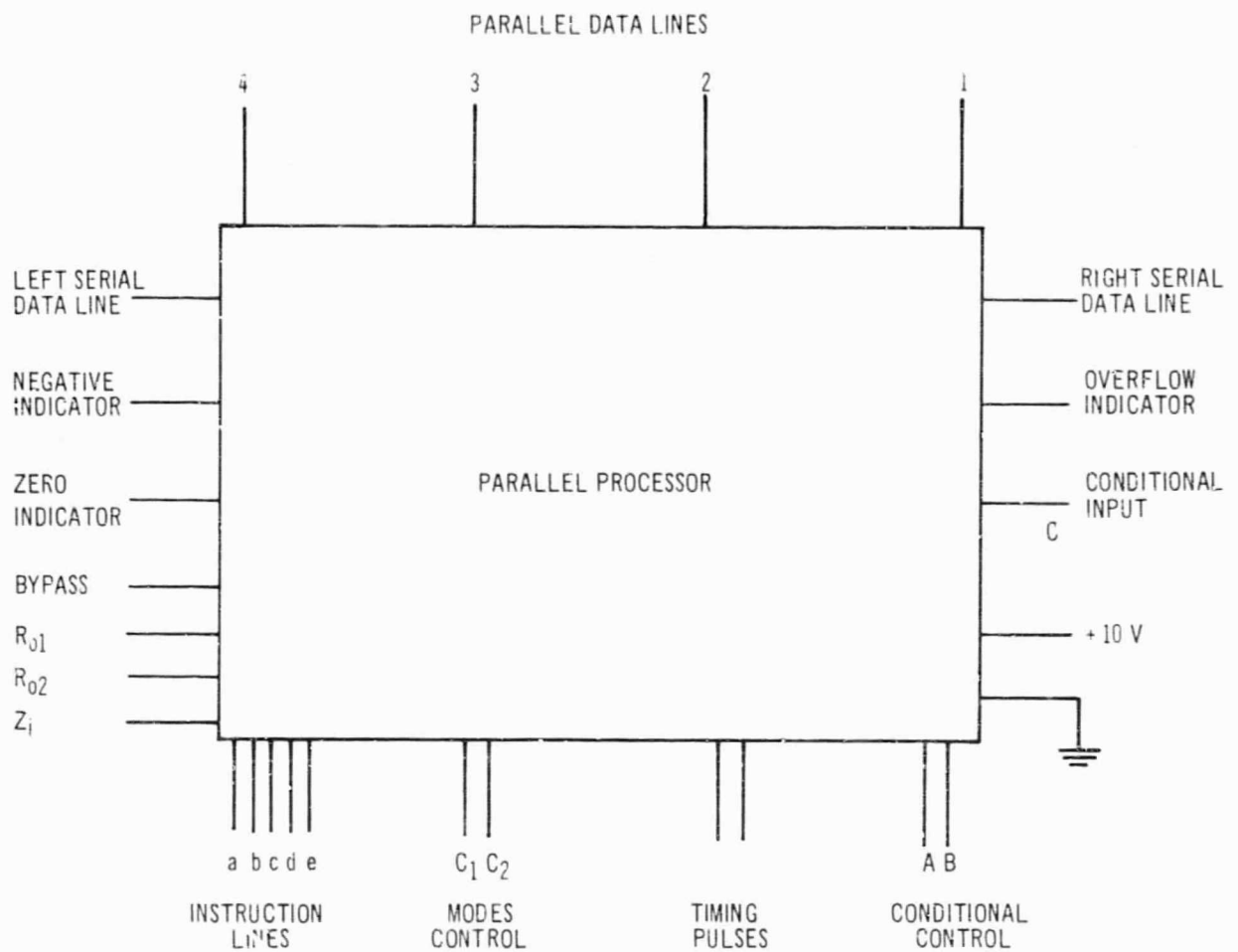
Figure 1 shows the lead requirements for the four-stage processor. All control lines are encoded, with five leads used for instructions; four leads for control; one lead for timing; two leads for power; and five leads for the following conditions:

- a. negative indication
- b. zero indication
- c. overflow indication
- d. overflow input/output
- e. conditional input

Because information will enter and leave on the same line, six leads are required for the four-stage register to transfer data, with four of the leads used for parallel access and the remaining two leads used for serial access. In addition, four leads are available for expansion to a multiple of four (16 for example) stage processor.

B. GENERAL LOGIC DESCRIPTION

The logic configuration for the four-stage parallel processor is shown in Figure 2. Functional gating is used where possible to achieve a low device count. Figure 3 shows the implementation of the basic gates used in the array



01985L

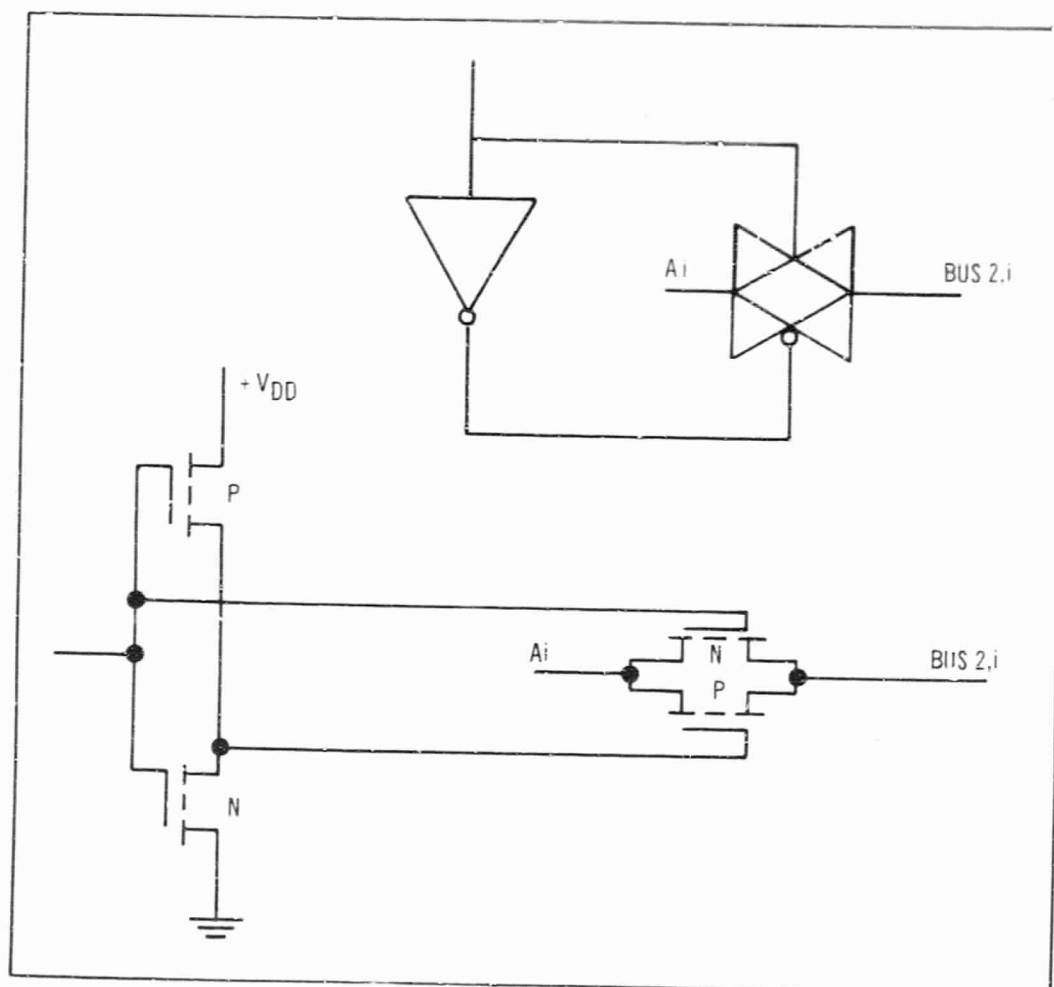
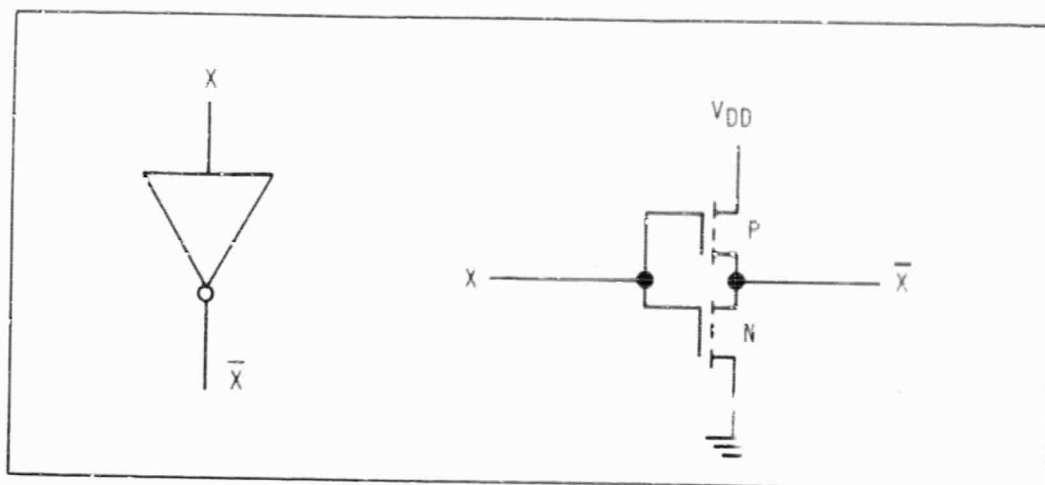
Figure 1. Four-Stage Processor, Lead Requirements



```

NOTE 1 8-BYTE FLIP-FLOP TRANSMITTER B INPUT TO Q OUTPUT ONE LOW-TO-HIGH
CLOCK TRANSITION.
NOT 2 NOT CONNECTED TO MOS FOR 4-BIT PROCESSOR.
NOT 3 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

```



©1988L

Figure 3. Implementation of Basic Gates in the Array Design Logic (Sheet 1 of 7)

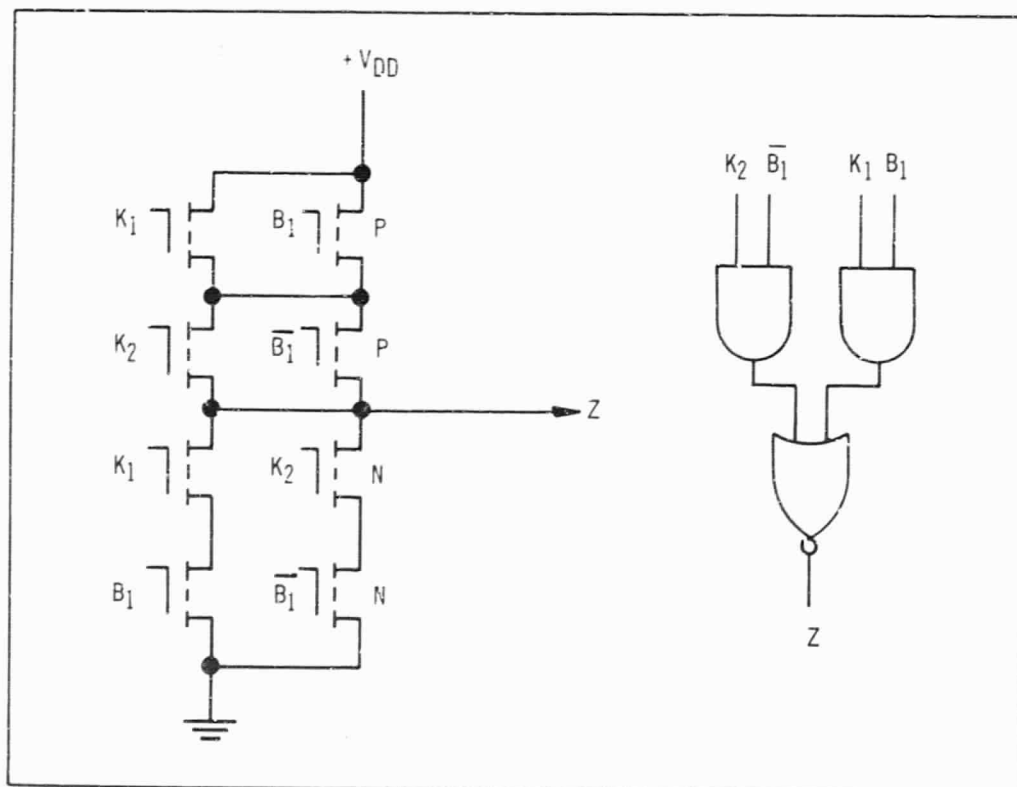
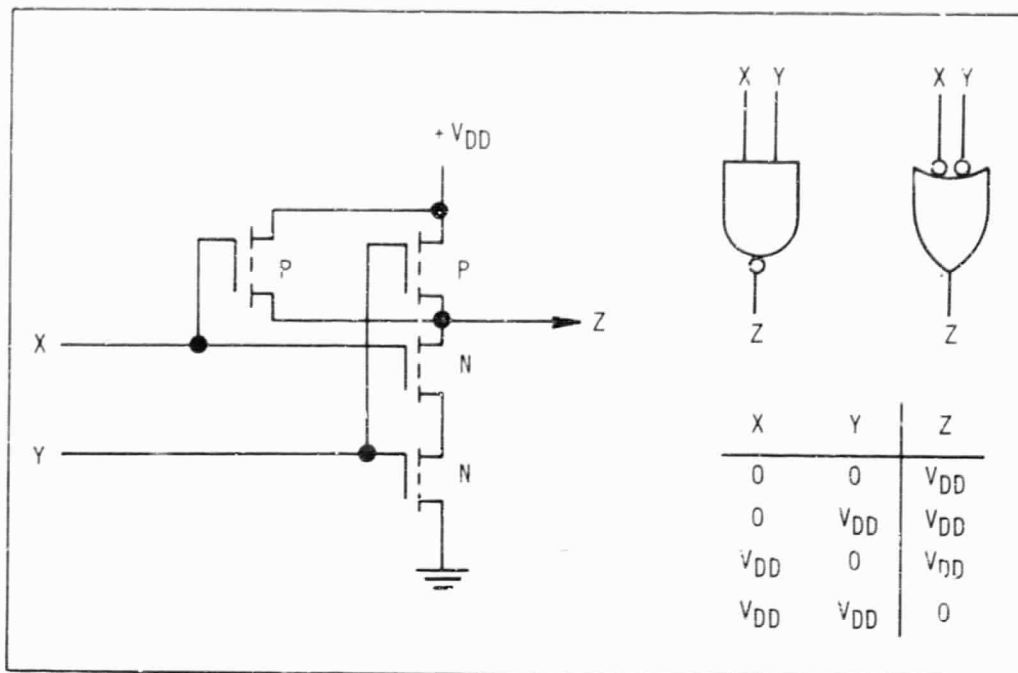
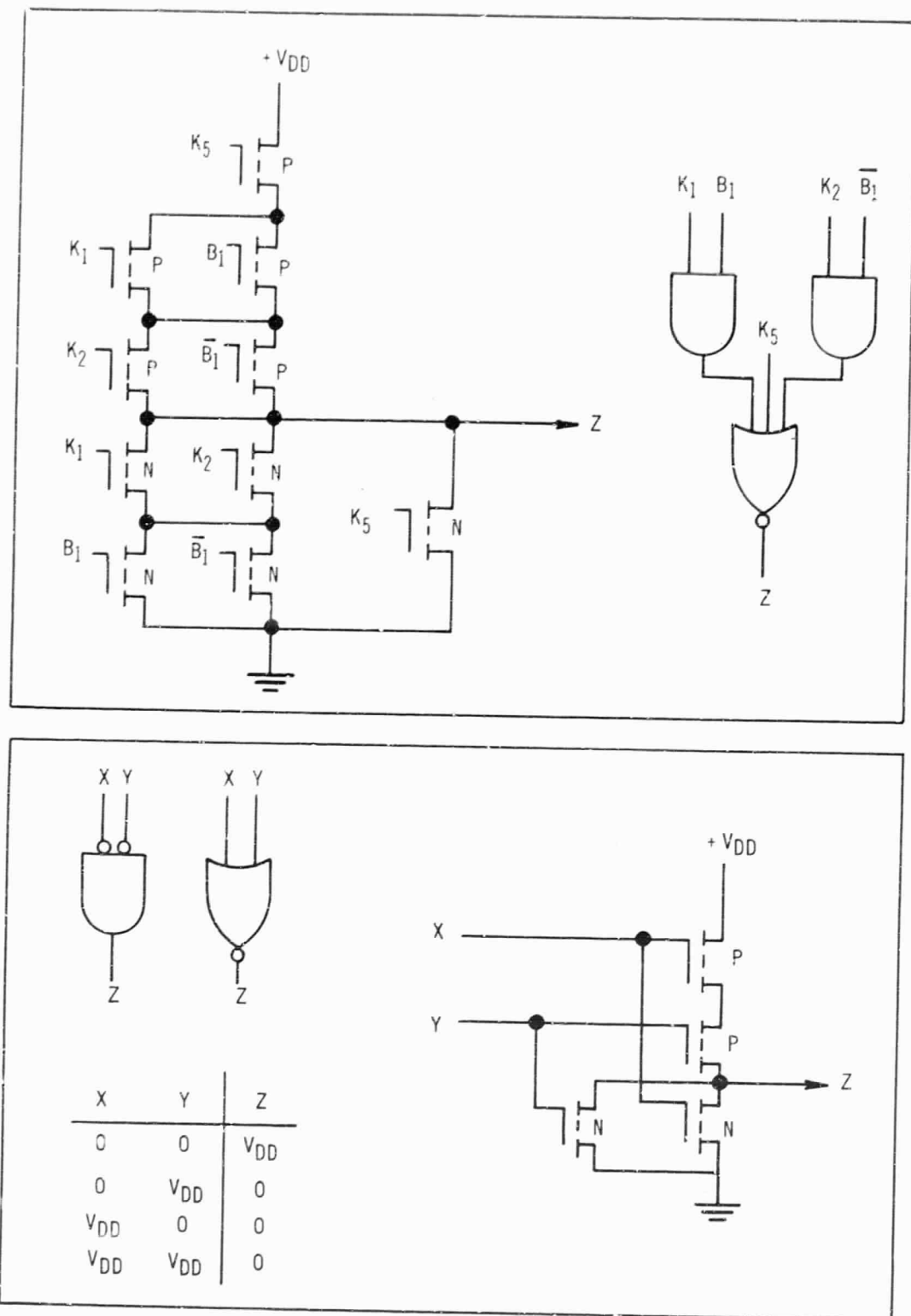
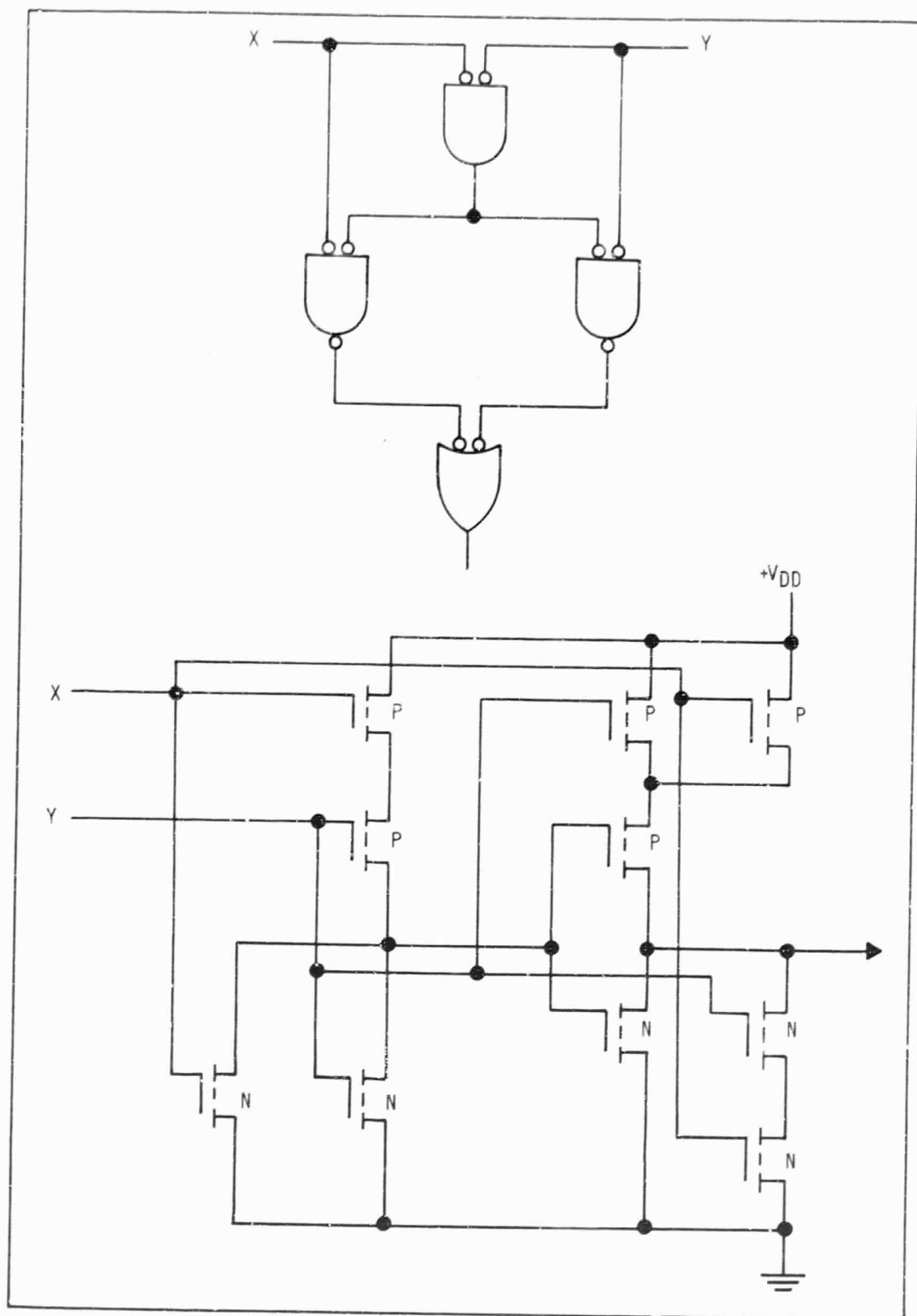


Figure 3. Implementation of Basic Gates in the Array Design Logic (Sheet 2 of 7)



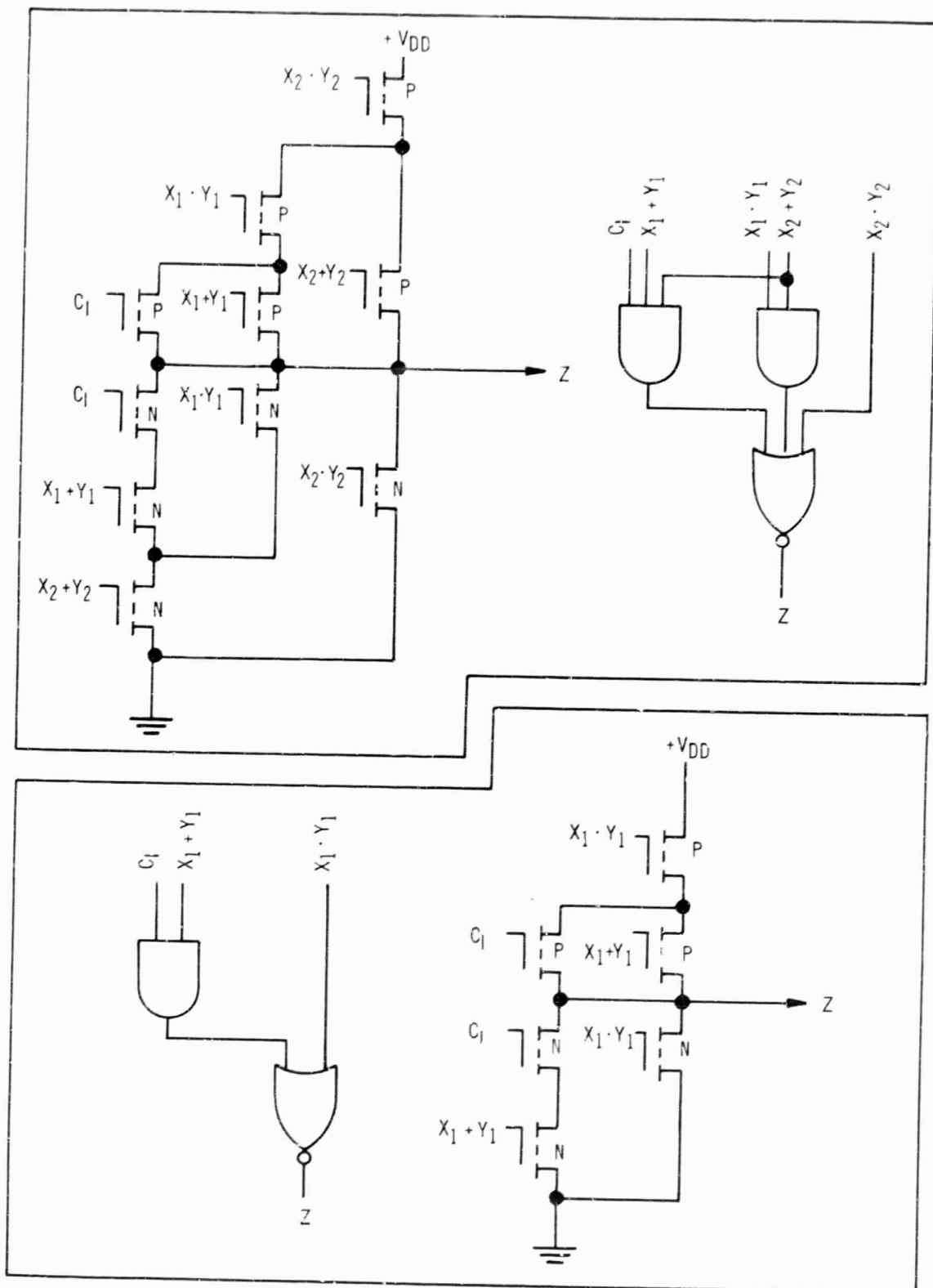
01989L

Figure 3. Implementation of Basic Gates in the Array Design Logic (Sheet 3 of 7)



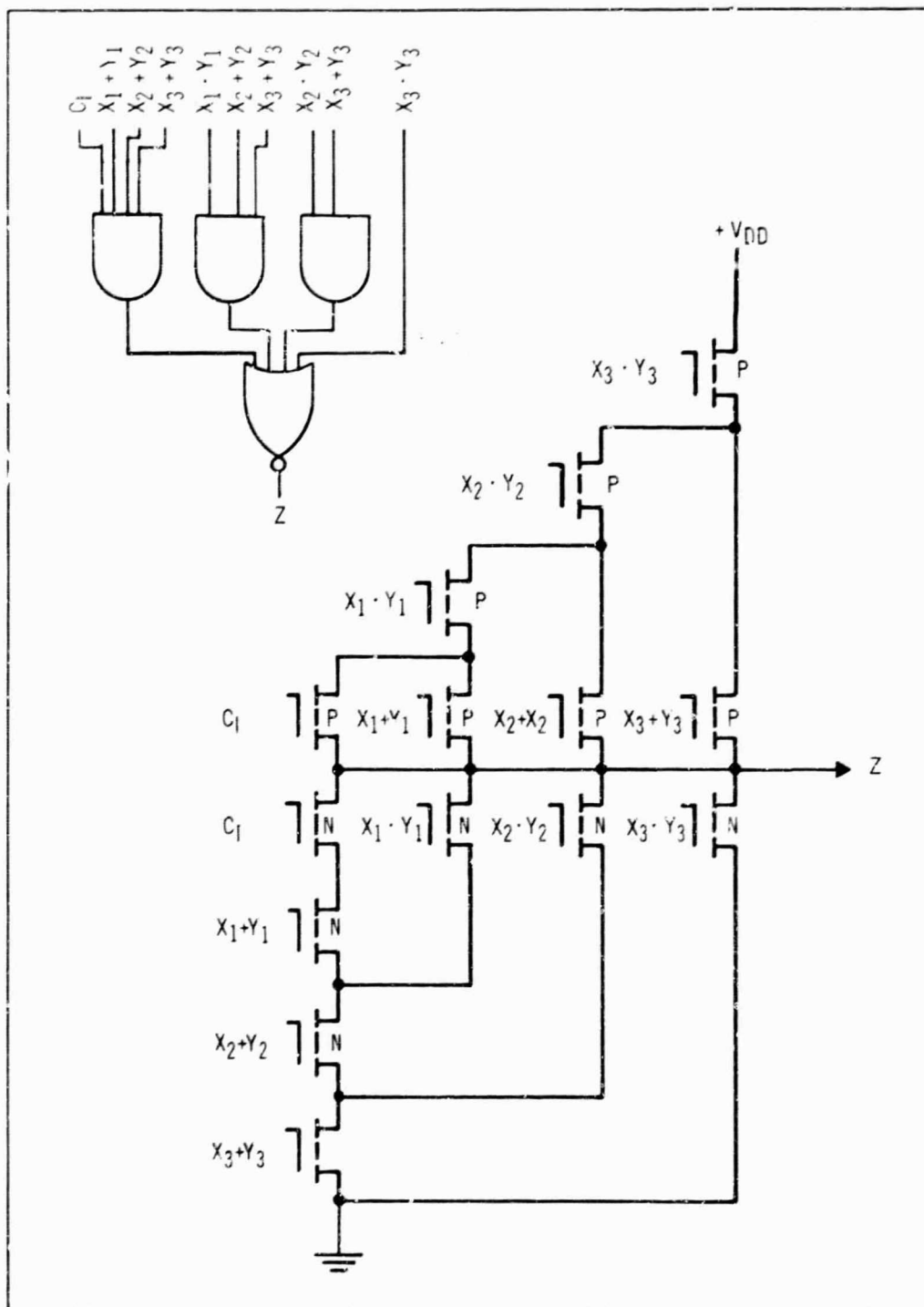
01990L

Figure 3. Implementation of Basic Gates in the Array Design Logic (Sheet 4 of 7)



01991L

Figure 3. Implementation of Basic Gates in the Array Design Logic (Sheet 5 of 7)



01996L

Figure 3. Implementation of Basic Gates in the Array Design Logic (Sheet 6 of 7)

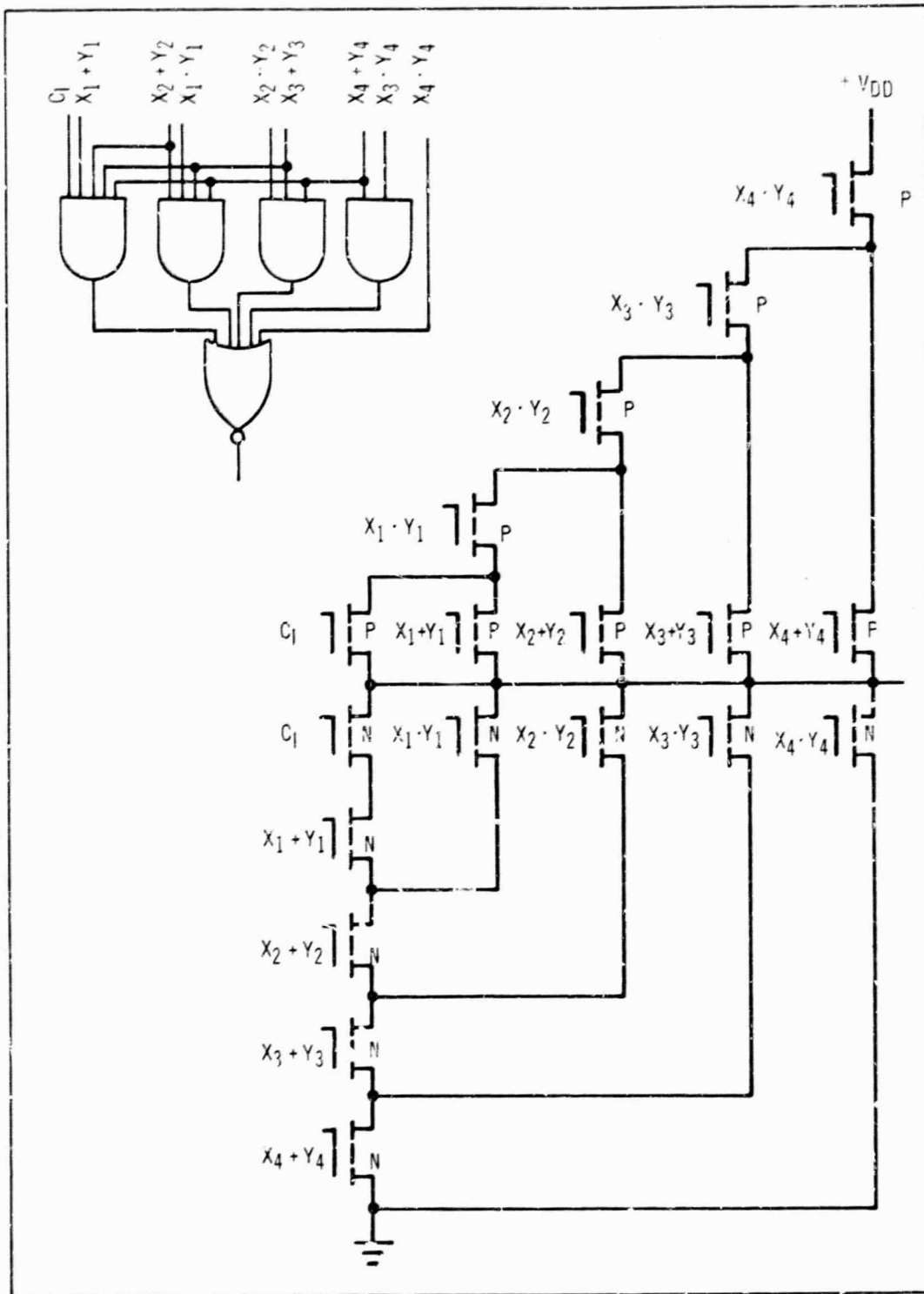


Figure 3. Implementation of Basic Gates in the Array Design Logic (Sheet 7 of 7)

design logic. The hardware requirement is about 750 devices and 27 bonding pads.

C. OPERATIONAL MODES

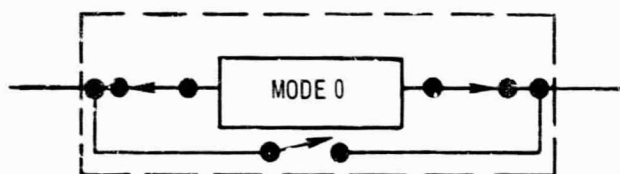
For this discussion mode is defined as the ability of the parallel processor to control the transfer of either serial data or carries due to arithmetic operations. The parallel processor will be capable of operation in one of four modes. For simplicity, consider the parallel processor as a strictly serial device. Serial data can enter or leave either side of the register. Since there is only one lead on either side of the register, the serial transfer must be bidirectional. The manner in which modes control the serial-data lines is as follows:

- a. Mode 0 (A, Figure 4) - Data can enter or leave from either side.
- b. Mode 1 (B, Figure 4) - Data can enter or leave the register on the left side during any serial operation.
- c. Mode 2 (C, Figure 4) - Data can enter or leave on the right side.
- d. Mode 3 (D, Figure 4) - Serial data neither may enter nor leave the register, regardless of the nature of the serial operation; furthermore, the register is bypassed electrically, i.e., there is an electrical bidirectional path from the right serial lead to the left serial lead. The most-significant (leftmost) bit is used as the sign bit.

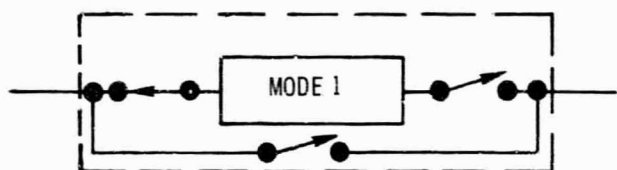
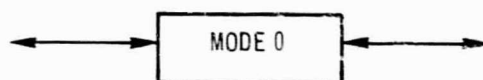
D. OVERFLOW DETECTION

A two's complement overflow is defined as having occurred if the signs of the two initial words are the same and the sign of the result is different while performing the ADD instruction.

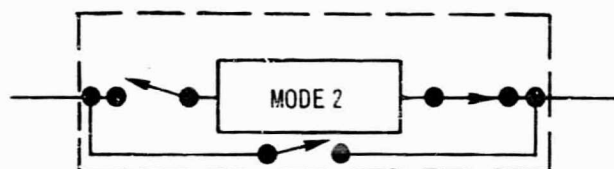
The parallel processor will be capable of detecting and indicating the presence or absence of an arithmetic two's complement overflow. Overflows will be detected and indicated only during operation in Mode 2 or Mode 3. In either mode, only four instructions (AD, SMZ, SM, and SUB) will have the potential of causing a two's complement overflow. If an overflow is detected



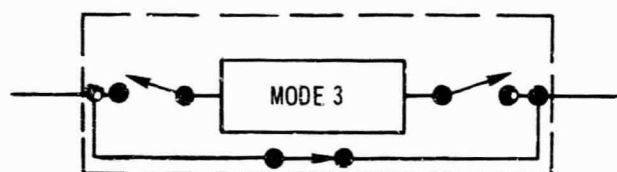
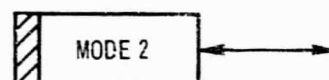
A. MODE 0



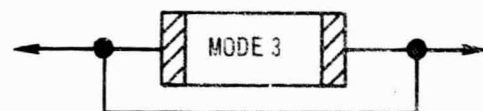
B. MODE 1



C. MODE 2



D. MODE 3



01993L

Figure 4. Operational Modes

and stored by a flip flop, only one of the five instructions (AD, SMZ, SM, SUB or IN) can change the overflow indicator.

Occurrence of a two's complement overflow is represented by a "1" in the overflow flip flop. The absence of an overflow is represented by a "0" in the overflow flip flop. The flip flop will change from zero to one as overflows do not or do occur.

When any one of the three subtraction instructions is used, the sign bit of the data being subtracted will be complemented, and this value will be used in the same manner as one of the initial signs (as in the add instruction) to detect overflows. If these conditions are true, the final sign will be that of the one's complement.

The overflow flip flop will be updated at the same time that the new result is stored in the parallel processor.

E. ZERO DETECTION

The parallel processor will be detecting the condition of all zeros. A "1" on the ZI input, indicating all zeros in the previous stage or that the stage is the least significant state, is required. This operation will be independent of modes. A condition of all zeros will be represented by a "1" on the zero indicator line; otherwise, this line will be zero. If the particular four-bit processor represents the least significant set of bits, ZI should be tied to +V.

F. NEGATIVE DETECTION

The parallel processor will be capable of detecting the presence of a negative number. This operation is independent of modes. If the condition is true, a "1" will appear on the negative indicator line; otherwise, a "0" will appear. A "1" in the most-significant bit position will indicate a negative representation.

C. CONDITIONAL OPERATION

Once the instruction and mode have been applied, only the clock pulse will be required to change the state of the register. If this pulse could be inhibited in the ON condition, all instructions would behave as a NO-OP.

The clock pulse can be constrained by using "conditions." A conditional input is compared with a control line, and a second control line defines whether or not to test the conditional input line. An instruction will be permitted to operate under the following conditions:

- a. Unconditional
- b. The conditional input is positive
- c. The conditional input is negative

H. INSTRUCTION REPERTOIRE

Four encoded lines will be used to represent 16 instructions. A fifth line will be used solely to represent an OUT command. Encoded instructions will be as follows:

- NO-OP
- Left shift
- Right shift
- Rotate (cycle) right
- Input
- Subtract from memory (SM)
- Count up
- Count down
- Clear to zero
- Set to one
- AND
- OR
- EXCLUSIVE OR
- Subtract from zero (SMZ)
- Add (AD)
- Subtract (SUB)

I. SERIAL-SHIFT OPERATIONS

- a. Rotate (cycle) right - This operation is internal. The contents of the register will shift to the right, cyclic fashion, with the leftmost stage accepting data from the rightmost stage, regardless of mode. Data may leave the register serially on the right data line only while the register is in Mode 2 or Mode 0. Data may leave the left data line serially while in Mode 1 or Mode 0.
- b. Right shift - The contents of the register generally will shift to the right under the following conditions:
 - (1) In Mode 0, data may enter serially on the left data line, shift through the register, and leave on the right data line.
 - (2) In Mode 1, data may enter serially on the left data line. The right data line effectively will be open-circuited.
 - (3) In Mode 2, data may leave serially on the right data line. The left data line effectively will be open-circuited. Vacant spaces will be filled with zeros.
 - (4) In Mode 3, serial data neither may enter nor leave the register; however, the contents will shift to the right, and vacated places will be filled with zeros.
- c. Left shift - The contents of the register generally will shift to the left under the following conditions:
 - (1) In Mode 0, data may enter the right data line, shift through the register, and leave on the left data line.
 - (2) In Mode 1, data may leave serially on the left data line. The right data line effectively will be open-circuited. All vacant positions will be filled with zeros.
 - (3) In Mode 2, data may enter serially on the right data line. The left data line effectively will be open-circuited.
 - (4) In Mode 3, data neither may enter nor leave the register; however, the contents will shift to the left, and vacated places will be filled with zeros.

J. PARALLEL COMMANDS

- a. CLEAR - sets register to zero.
- b. SET - sets register to all ones.
- c. OR - processes contents of register with value on parallel-data lines in a logical OR function.
- d. AND - processes contents of register with value on parallel-data lines in a logical AND function.
- e. EXCLUSIVE OR - processes contents of register with value on parallel-data lines in a logical EXCLUSIVE OR function.
- f. IN - loads value on parallel-data lines into register.
- g. OUT - outputs contents of register on parallel-data lines.
- h. SUB:
 - (1) In Mode 1, adds to the contents of the register the two's complement of whatever is on the parallel-data lines. Generated carries may leave on the left serial line. The overflow indicator is not altered.
 - (2) In Mode 2, adds to the contents of the register the one's complement of whatever is on the parallel-data lines. Carries may enter on the right serial line but may not leave on the left data line. The absence or presence of an overflow is registered.
 - (3) In Mode 0, same as Mode 2, except carries may leave on the left data line. The overflow indicator is not altered.
 - (4) In Mode 3, same as Mode 1, except carries may not leave on the left data line. The absence or presence of an overflow is registered.
- i. COUNT UP:
 - (1) In Mode 1, internally adds one to the contents of the register and permits any resulting carry to leave on the left serial-data line. No data enters or leaves either the parallel lines or the right serial line.

- (2) In Mode 2, adds to the contents of the register whatever is on the right serial-data line. No data enters or leaves either the parallel lines or the left serial line.
- (3) In Mode 0, adds to the contents of the register whatever is on the right serial line and permits any resulting carry to leave on the left data line. No data enters or leaves the parallel lines.
- (4) In Mode 3, internally adds one to the contents of the register. No data enters or leaves the register on any serial-data or parallel-data line.

j. COUNT DOWN:

- (1) In Mode 1, internally subtracts one from the contents of the register and permits any resulting carry to leave on the left serial-data line. No data enters or leaves either the parallel lines or the right serial line.
- (2) In Mode 2, subtracts one from the contents of the register and adds to this result whatever is on the right serial-data line. No data enters or leaves the parallel lines or the left data line.
- (3) In Mode 0, subtracts one from the contents of the register and adds to this result whatever is on the right serial-data line and permits any resulting carry to leave on the left data line. No data enters or leaves the parallel lines.
- (4) In Mode 3, internally subtracts one from the contents of the register. No data enters or leaves either the parallel lines or the serial lines.

k. AD:

- (1) In Mode 1, adds the contents of the register to whatever is on the parallel-data lines and allows any resulting carry to leave on the left data line. The right serial-data line is open-circuited. The overflow indicator is not altered.

- (2) In Mode 2, adds the contents of the register to whatever is on the parallel-data lines and the right serial-data line. Any overflows will set the overflow indicator. The left serial-data line is open-circuited. The absence or presence of an overflow is registered.
 - (3) In Mode 0, adds the contents of the register to whatever is on the parallel-data lines and the right serial-data line. Any resulting carry may leave on the left serial-data line. The overflow indicator is not altered.
 - (4) In Mode 3, adds contents of the register to whatever is on the parallel-data line. Any resulting carry will set an overflow indicator. The two serial-data lines are open-circuited. The absence or presence of an overflow is registered.
1. SM - same operation as AD, except the contents of the register are two's complemented during addition in Mode 1 and Mode 3. In Mode 0 or Mode 2, the contents of the register are one's complemented and added to whatever is on the right serial-data line and the parallel-data lines. Overflows occurring in Mode 1 or Mode 0 do not alter the overflow indicator. The presence or absence of overflows is registered on the overflow indicator in Mode 2 or Mode 3.
- m. SMZ:
- (1) In Mode 0, one's complements the contents of the register and adds whatever is on the right serial-data line to the contents of the register. Any resulting carry may leave the left serial line. Any overflow will not alter the overflow indicator.
 - (2) In Mode 1, two's complements the contents of the register and permits any carry to leave on the serial line. Nothing may enter the right serial line. Any overflow will not alter the overflow indicator.
 - (3) In Mode 2, one's complements the contents of the register and adds whatever is on the right serial line to the contents of the

register. Carries may not leave the left serial line. The absence or presence of an overflow will alter the overflow indicator.

- (4) In Mode 3, two's complements the contents of the register. Serial data neither may enter the right serial line nor leave the left serial line. The overflow indicator will be at zero.

- n. NO-OP - The NO-OP condition will inhibit the clock signal before the D-type flip flops.

K. TIMING

Transfer of data is accomplished by using a D-type flip flop (Figure 15) which requires one clock pulse to transfer data on the input into the storage element.

The D-type flip flop consists of two double inverters which may feedback on themselves through transmission gates providing a stable state. When the clock is low, transmission gates "1" and "3" are active and gates "2" and "4" are inactive. This state permits the retention of data by the second inverter pair while allowing the incoming data to define the state of the first inverter pair.

When the clock undergoes a low-to-high transition, the states of all transmission gates are changed. During this transition the flip-flop input becomes isolated and the first inverter pair is stabilized by opening the feedback transmission gate holding the information which was on the data line. Meanwhile, the second inverter pair loses its feedback and a path is established from the first stage. For a set of D flip flops in a shift-register configuration, the effect of this transition is to permit the first stage of each flip flop to store information from the output of the previous flip flop before the second stage of the flip flops changes due to the new flip-flop input. During the high-to-low transition, the new data are transferred to the second inverter pair, in a manner similar to the original transfer, and the normal storage mode is assumed.

L. MODE-INDEPENDENT SWITCHES

The state of the control lines to the processing logic for the 15 operating instructions is shown in Table I. True data from the parallel inputs are gated into the parallel processor when K_1 is high. The pertinent equation is

$$K_1 = \bar{a}cd + ac + c\bar{d} \quad \dots \dots \dots (1)$$

Complementary data from the parallel inputs are gated into the processor by K_2 , which is given by

$$K_2 = bcd \quad \dots \dots \dots (2)$$

True information in the register is gated into the processor by K_3 , which is given by

$$K_3 = \bar{a}\bar{b} + c \quad \dots \dots \dots (3)$$

and the complementary information is gated by K_4 , where

$$K_4 = b\bar{c} \quad \dots \dots \dots (4)$$

Control K_5 is used to set all ones into the processor for one operand. Control K_5 can be gated through for a SET or can be used in COUNT DOWN. The pertinent equation is

$$K_5 = \bar{a}\bar{b}\bar{c} + a\bar{c}\bar{d} \quad \dots \dots \dots (5)$$

The EXCLUSIVE OR can be inhibited when XI is high, which allows the OR operation to be formed. The switching equation is

$$XI = ad \quad \dots \dots \dots (6)$$

The IN transmission gate is used to load the register in parallel, where

$$IN = abc\bar{d} \quad \dots \dots \dots (7)$$

TABLE I. OPERATIONAL CODE FOR THE PARALLEL PROCESSOR ARRAY

Operational Code					Mode Independent								Mode Dependent														
In-struction	a	b	c	d	K ₁	K ₂	K ₃	K ₄	K ₅	XI	IN	SUM	AND	CR	R ₀	R	L	(2,3) R _n	(0,1) R _{T1}	(0,1) L _{T1}	(0,2) R _{T0}	(0,2) L _{T0}	(1,3) L _n	(3,1) Q	(3,1) J	(0,2) CAR	(0,1) M
NOP	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
AND	0	0	0	1	1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1
CNTD	0	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
CNTU	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
SHZ	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
SM	0	1	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
AD	0	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
SUB	0	1	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
SET	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
CLEAR	1	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
XOR	1	0	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
OR	1	0	1	1	1	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
IN	1	1	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
L	1	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0
R	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1	0	0
R ₀	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0	1	1	0	0	1	1	1	0	1	0	0	0

The SUM transmission gate is used to perform all add-type instructions, where

$$\text{SUM} = \bar{a} (b + c) \dots\dots\dots (8)$$

Logic operations are performed by the AND and OR switches, where

$$\text{AND} = \bar{a}\bar{b}cd \dots\dots\dots (9)$$

and

$$\text{OR} = a\bar{b} \dots\dots\dots (10)$$

M. MODE-DEPENDENT SWITCHES

Code definitions for the modes were selected such that there is no need to decode. The definitions selected are as follows:

<u>Mode</u>	<u>C₂</u>	<u>C₁</u>
0	0	0
1	0	1
2	1	0
3	1	1

When high or "1," line C₁ or C₂ indicates which side of the processor is inhibited. Lead C₁ corresponds to the right side; lead C₂ corresponds to the left side.

Transmission gate Q (Figure 2) is used to force a "1" into the carry of the least significant adder during a COUNT UP or SUBTRACT operation and during Mode 1 or Mode 3. The equation is

$$Q = C_1 (\bar{c} + d) \dots\dots\dots (11)$$

A zero is forced into the carry by J, which is given by

$$J = C_1 \cdot c\bar{d} \dots\dots\dots (12)$$

During Mode 0 and Mode 2, a carry is brought in from a previous array by the CAR transmission gate, where

$$\text{CAR} = \bar{C}_1 \cdot \bar{a} \dots\dots\dots (13)$$

A carry may propagate out during Modes 0 and 1 by using an M, which is given by

$$M = \bar{C}_2 \cdot \bar{a} \dots \dots \dots (14)$$

N. SHIFT SWITCHES

The R and L transmission devices are mode independent gates used to perform the right and left shifts. R_N , R_O , R_{T1} , R_{T0} , L_{T1} , L_{T0} , and L_N are mode dependent shift controls. The equations are summarized as follows:

$$R_N = C_2 \cdot abcd \dots \dots \dots (15)$$

$$R_O = abcd \dots \dots \dots (16)$$

$$R_{T,1} = \bar{C}_2 \cdot abc \dots \dots \dots (17)$$

$$L_{T,1} = \bar{C}_2 \cdot abcd \dots \dots \dots (18)$$

$$R_{T,0} = \bar{C}_1 \cdot abc \dots \dots \dots (19)$$

$$L_{T,0} = \bar{C}_1 \cdot abcd \dots \dots \dots (20)$$

$$L_N = C_1 \cdot abcd \dots \dots \dots (21)$$

$$R = abc \dots \dots \dots (22)$$

$$L = P \cdot abcd \dots \dots \dots (23)$$

Note: R denotes right, L denotes left

O. CONDITIONAL OPERATION

The clock pulse operates on condition. Three control lines will be defined to permit conditional instructions. These lines will be labeled A, B, and C. The following truth table defines interactions among A, B, and C:

<u>C</u>	<u>B</u>	<u>A</u>	<u>Permits Operation</u>
0	0	0	Yes
0	0	1	Yes
0	1	0	Yes
0	1	1	No
1	0	0	Yes
1	0	1	No
1	1	0	Yes
1	1	1	Yes

The truth table reduces to the condition that $(A + B \cdot C + \bar{B} \cdot \bar{C})$ data-transfer operation is to take place. This expression combined with the clock pulse accomplishes the data transfer providing the processor is not in the NO-OP state.

P. OVERFLOW INDICATOR

Switch SIGNA operates and puts the truth sum in the register under either one of the following conditions: overflows have not been detected; or the mode of operation and the instruction are such that overflow detection is not needed. This condition can be summarized by

$$\begin{aligned} \text{SIGNA} = & \overline{(\bar{S}_f \cdot S_1 \cdot S_2 \cdot S_f \cdot \bar{S}_1 \cdot \bar{S}_2)} \cdot C_2 \cdot \bar{a}b \\ & + \bar{C}_2 \cdot \bar{a}b + \bar{a}bc \quad \dots \dots \dots (24) \end{aligned}$$

Switch SIGNB operates when an overflow occurs, and a "1" is placed in the overflow flip flop. The complement of the most significant bit sum output also is placed in the register; hence,

$$\text{SIGNB} = (\bar{S}_f \cdot S_1 \cdot S_2 + S_f \cdot \bar{S}_1 \cdot \bar{S}_2) \cdot \bar{a}b \cdot C_2 \quad \dots \dots \dots (25)$$

A zero is placed in the overflow flip-flop when the following condition is true:

$$\overline{(\bar{S}_f \cdot S_1 \cdot S_2 + S_f \cdot \bar{S}_1 \cdot \bar{S}_2)} \cdot \bar{a}b \cdot C_2 \quad \dots \dots \dots (26)$$

The overflow flip flop can only be clocked during ($\bar{a}b \cdot C_2$) or during the IN instruction. Data may be entered or removed from the overflow flip flop, on the OVERFLOW I/O line, and during the IN and OUT commands, respectively.

Q. EXPANSION TO 16-STAGE PROCESSOR

The four-stage parallel processor is designed such that four processors can be interconnected monolithically to form the 16-stage processor. The wafer will be diced such that four operating four-stage processors will form a 16-stage processor. The interconnection scheme is shown in Figure 5. In sections 2 and 3, the C_1 and C_2 mode controls are tied to ground; therefore, these sections are held in Mode 0. In section 1, C_2 is tied to ground; therefore, the section can operate only in Mode 0 or Mode 1. In section 4, lead C_1 is tied to ground; therefore, this section can operate only in Mode 0 or Mode 2. The modes for the 16-stage processor are determined by inputs C_1 and C_2 , as shown.

The bypass leads of sections 1 and 4 are connected as shown in Figure 5. When C_1 and C_2 are both "1," indicating Mode 3 the 16-stage register is bypassed; and the left serial-data line of chip 4 and the right serial-data line of section 1 are connected.

Leads R_{01} of section 4 and R_{02} of section 1 are connected as shown in Figure 5 to allow the rotate operation. R_0 denotes the rotate (cycle shift) function.

The ZERO INDICATOR leads are connected as shown. If a "0" occurs in the 16-stage parallel processor, a "1" will appear on the indicator of the left-most unit.

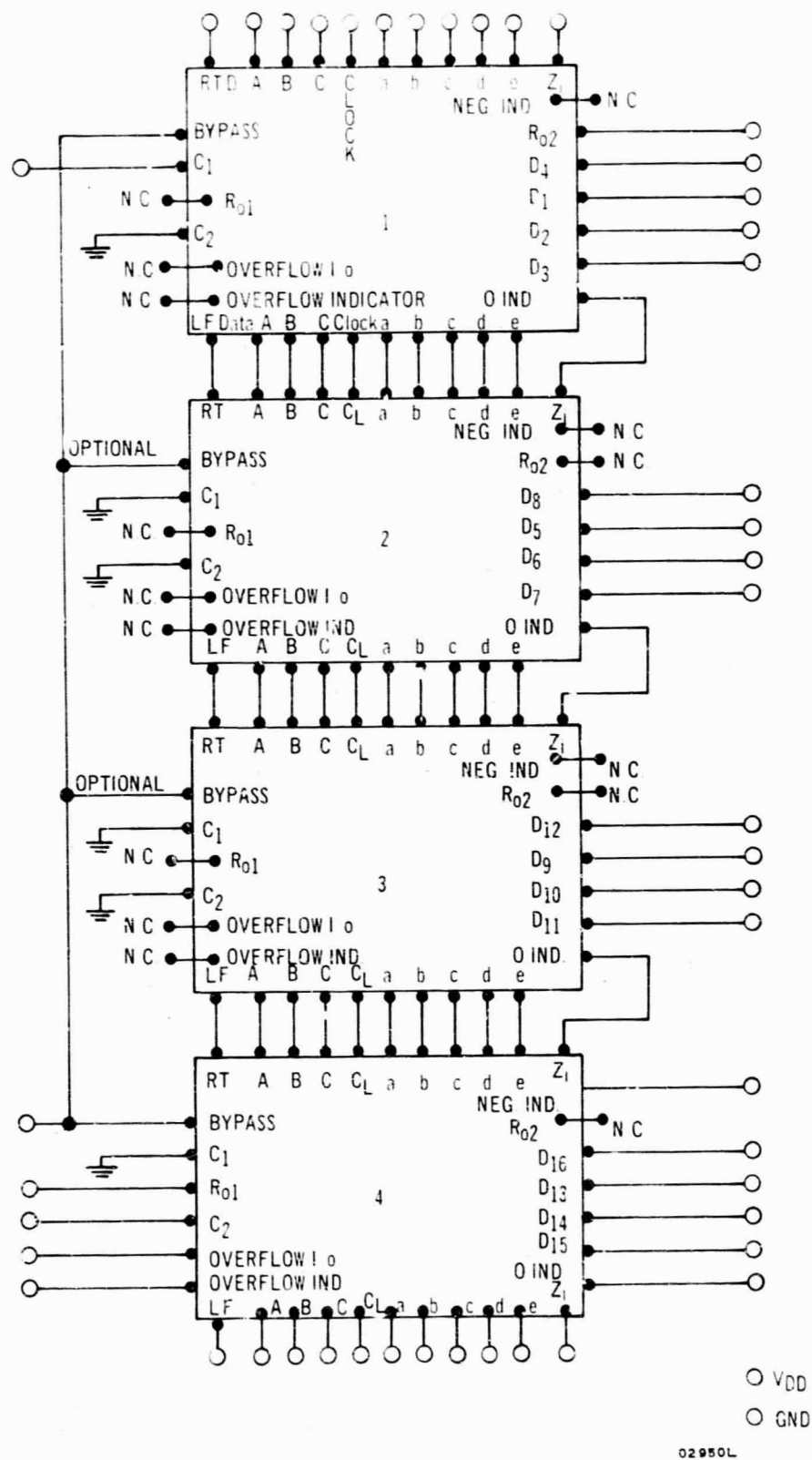


Figure 5. Schematic Interconnection of Four Four-Stage Chips

SECTION III
BREADBOARD COMPUTER STUDIES

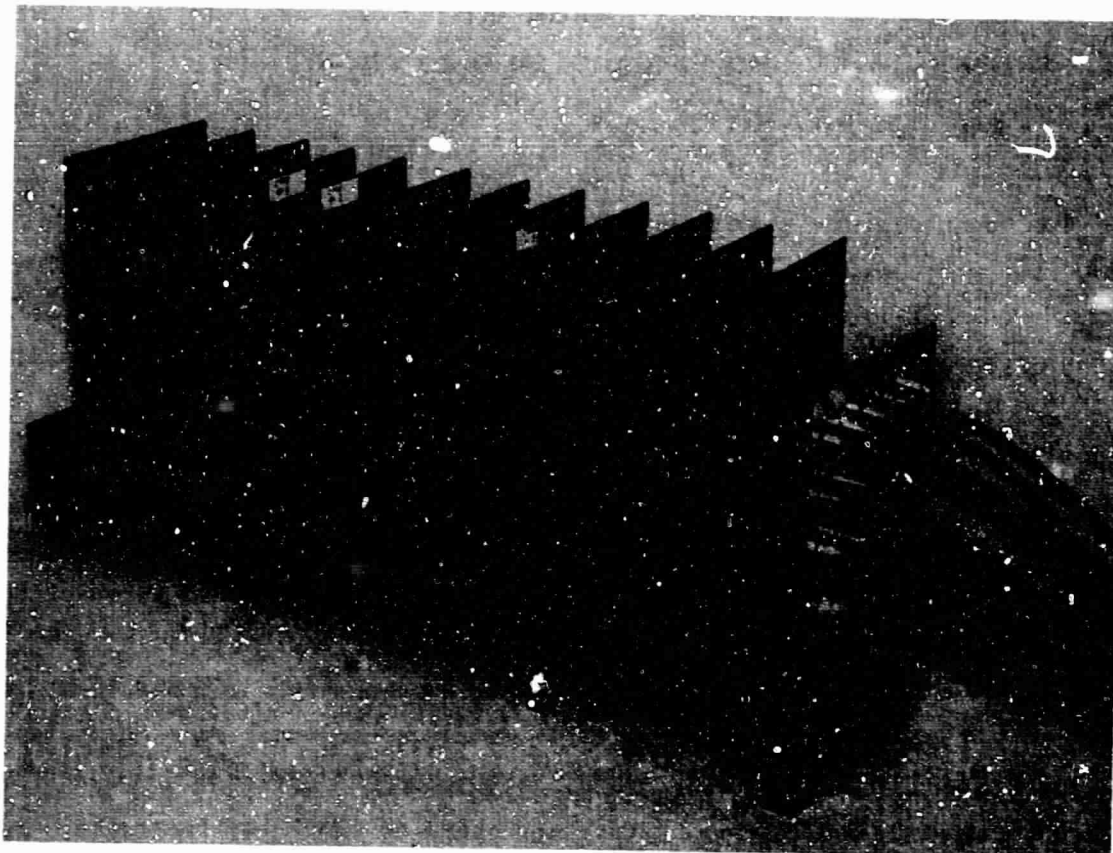
A. SUMMARY

The logic diagram, as received from RCA Burlington, was evaluated by constructing a breadboard (Figure 6) of the entire processor, and by computer programming a portion of the processor. The breadboard verified that the major logic and control functions were correct. The computer runs showed that the proposed device sizes (1.1 mils) were adequate.

B. DETAILS

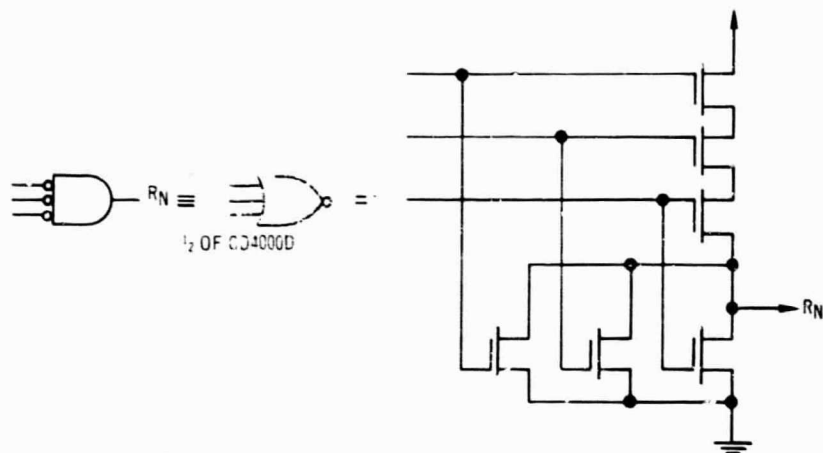
The contract specified that only one stage (plus control logic) of the four-stage parallel processor would be breadboarded. However, in order to verify the total logic package, and to aid in verifying the test program, it was decided to breadboard the entire parallel processor. The breadboard was constructed on 12 printed circuit cards which were mounted on a rack with plug-in sockets. Only commercially available RCA COS/MOS devices were used for the breadboard. The CD4007D was the primary building block for the breadboard, and several CD4000D's, CD4001D's, CD4002D's, and CD4003D's were used to complement its fabrication. A total of 121 COS/MOS packages were required to complete the breadboard.

The control logic was constructed first because it is needed to drive all the other sections. Functional gating was used wherever appropriate so that the breadboard would correspond as closely as feasible to the actual IC. Commercially available COS/MOS NOR gates were used wherever possible. For example, the R_N control required a zero-level-active-NAND gate. Since a zero-active-NAND gate is identical to an active-high-NOR gate, RCA's CD4000D could be used (see A, Figure 7). The CD4007D was the most extensively used building block. This device is essential for constructing the functional gates such as shown in Figure 7.

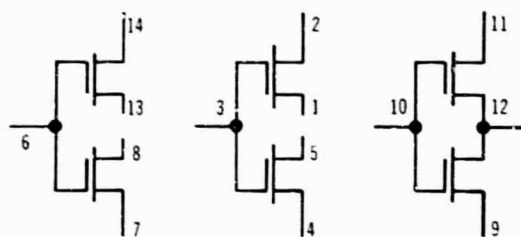


02981P

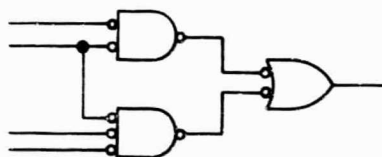
Figure 6. Breadboard of the Processor



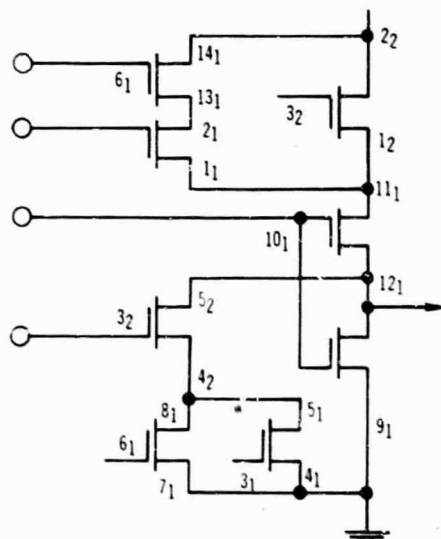
A. SCHEMATIC OF ACTIVE-LOW NAND GATE



B. SCHEMATIC OF BASIC BUILDING BLOCK (CD4007D) IN BREADBOARD



C. FUNCTIONAL GATE IN CONTROL SECTION



NOTE
NUMBERS REFER TO PIN NUMBER
OF CD4007D. SUBSCRIPT REFERS
TO UNIT NUMBER (TWO CD4007D's
REQUIRED)

029-2L

D. IMPLEMENTING THE FUNCTIONAL GATE USING CD4007D

Figure 7. Construction of Functional Gates for the Parallel Processor
Using the CD4007D as the Basic Building Block

After the control logic section was completed, it was checked against the truth table received from RCA Burlington. Figure 8 shows the output waveforms of the control section for all 16 possible input combinations.

The arithmetic and accumulator sections were built one stage at a time and debugged before proceeding. The register in each stage was implemented using the RCA CD4003D dual-data flip flop.

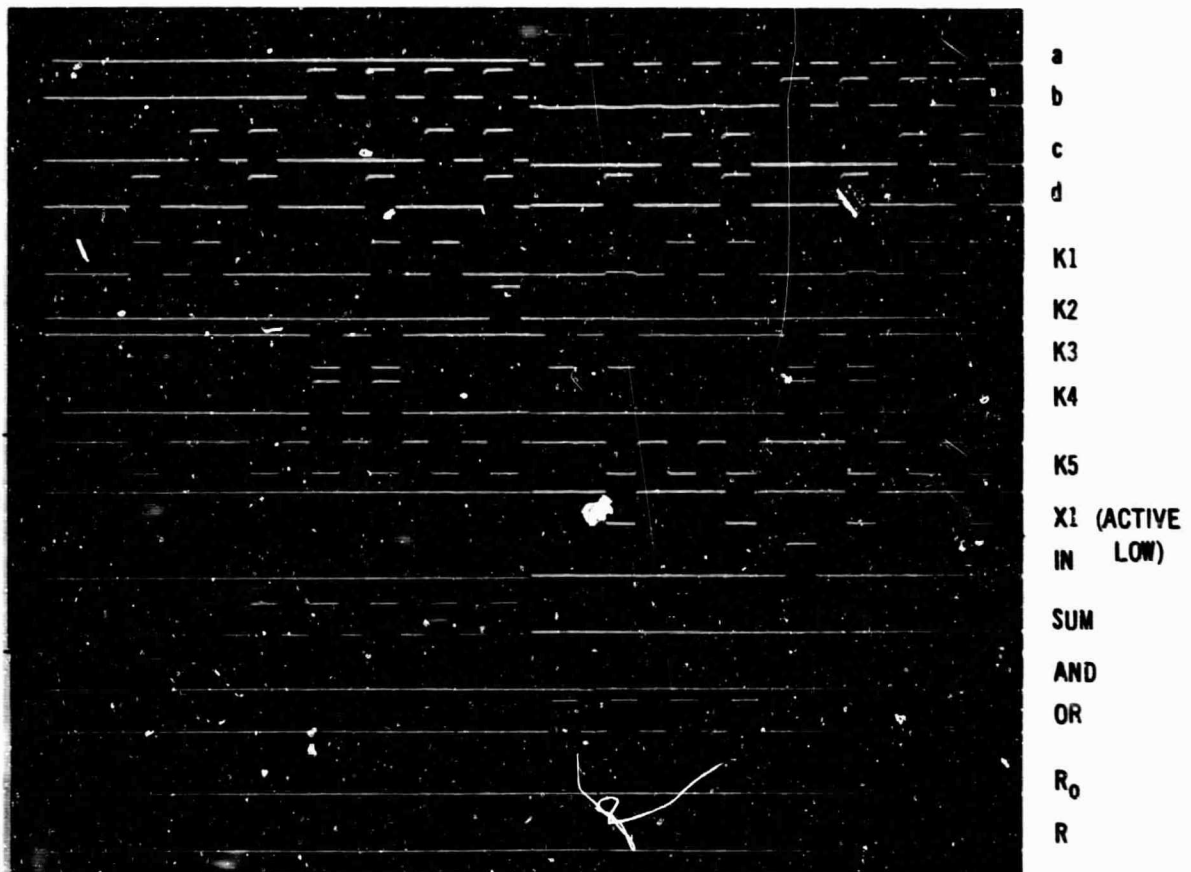
Two minor changes were made in the breadboard so that the logic functions could be fabricated with standard NOR gates instead of building up NOR or NAND gates with the CD4007D. The first change was in the zero indicator. The logic diagram as received from RCA Burlington is shown in A, Figure 9. The breadboarded version is shown in B, Figure 9. The second change involved the clock circuit. The circuit received from RCA Burlington is shown in A, Figure 10. The breadboarded circuit is shown in B, Figure 10.

The complete breadboard was exercised for most major modes of operation such as shift right, shift left, AD, LOGICAL AND, EXCLUSIVE OR, etc. The worst case AD of two four-bit numbers was 1.9 microseconds, or within specifications. Figure 11 shows some typical operating waveforms.

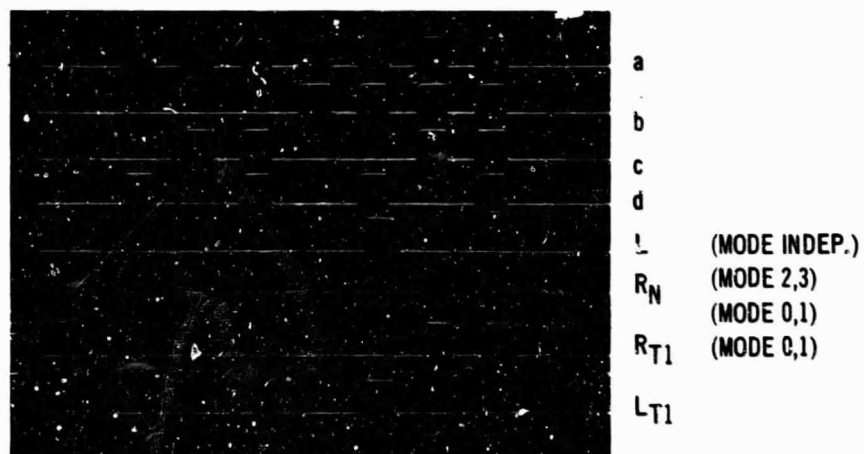
C. COMPUTER STUDIES OF SWITCHING SPEED

A program for computing the switching speed of COS/MOS circuits was developed by A. Feller of RCA Camden, New Jersey.⁽¹⁾ This program was adapted to BTSS II (Basic Time Sharing System) by J. E. Meyer, RCA Princeton, New Jersey. The program enabled RCA engineers in Somerville to submit data to the computer center via the teletype facilities in Somerville, New Jersey. The computer center processed these data in batch mode because of the length of the program. Output of the batch processor was mailed back to Somerville. This last step introduced considerable delay in debugging individual runs, as turn-around time (time from data submission to receipt of mail) was usually 2 to 3 days.

(1) A. Feller, Computer Analysis and Simulation of MOS Circuits, ISSCC, February 20, 1969



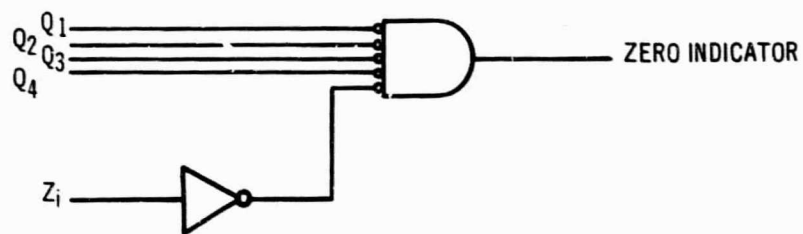
A. CONTROL LOGIC OUTPUT WAVEFORMS FOR THE 16 POSSIBLE CONTROL COMBINATIONS



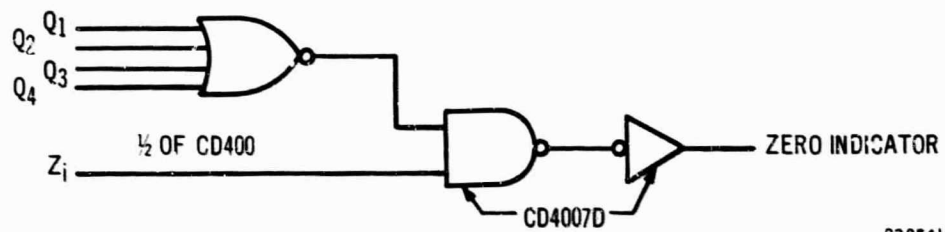
B. MODE-DEPENDENT OUTPUTS FROM CONTROL SECTION

02953P

Figure 8. Input and Output Waveforms for Control Section



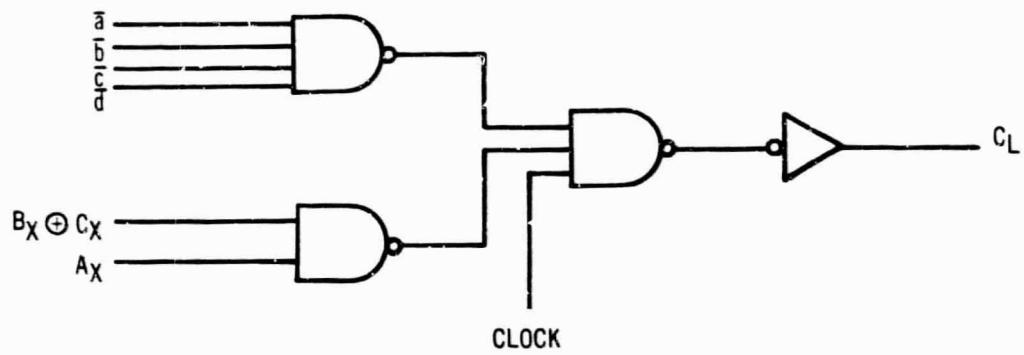
A. ORIGINAL LOGIC DIAGRAM



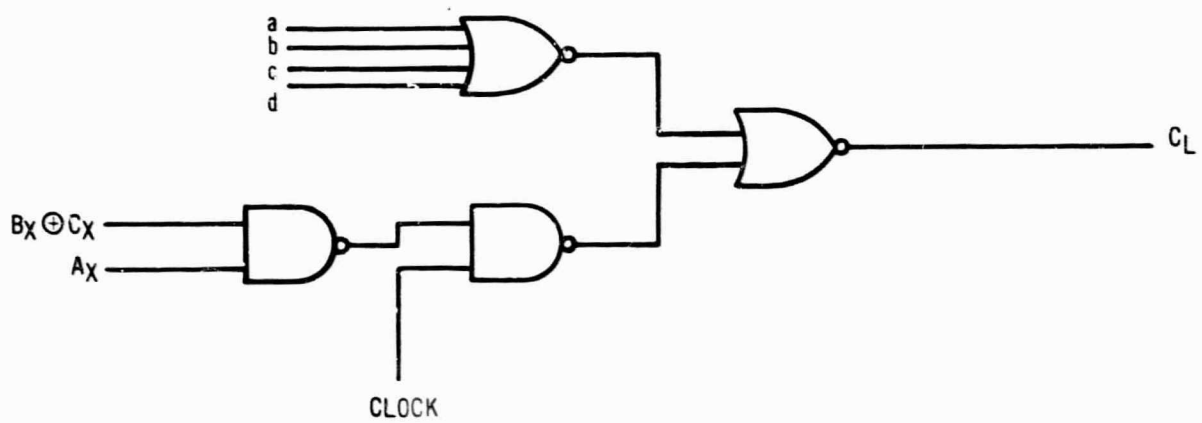
02954L

B. AS BREADBOARDED

Figure 9. Zero Indicator



A. AS ORIGINALLY DESIGNED



B. AS BREADBOARDED

02955L

Figure 10. Clock Circuit



a
b
c
d
CLOCK
D1
D2
D3
D4
Q1
Q2
Q3
Q4

SEQUENCE
1. IN COMMAND = ENTER DATA
2. IN
3. IN
4. ADD COMMAND = ADD $D_i = Q_i$
5. ADD
6. ADD
7. ADD

VERT = 20 V/CM
HOR = 20 μ S/CM

A. ADDITION



a
b
c
d
CLOCK
D1
D2
D3
D4
Q1
Q2
Q3
Q4

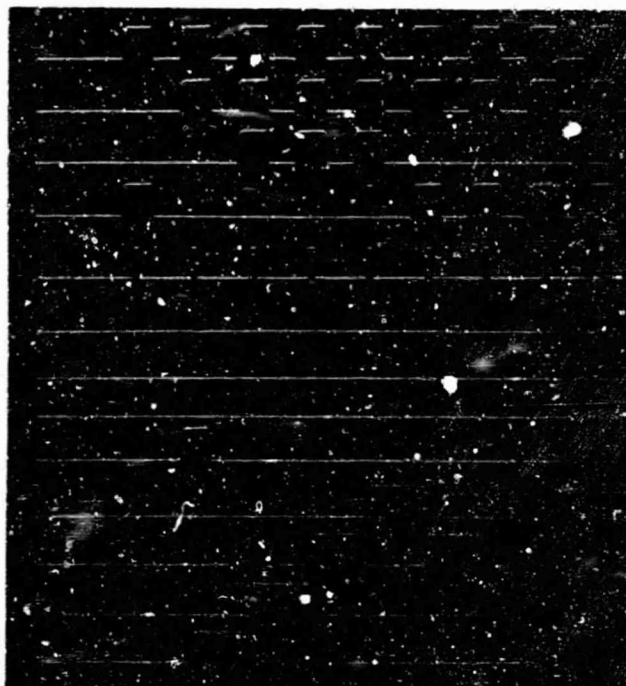
SEQUENCE
1. SET COMMAND
2. CLEAR
3. IN
4. EXCLUSIVE OR
5. EXCLUSIVE OR
6. OR
7. OR
8. AND
9. CLEAR

VERT = 20 V/CM
HOR = μ S/CM

B. EXCLUSIVE OR

07.386P

Figure 11. Typical Operating Waveforms (Sheet 1 of 2)



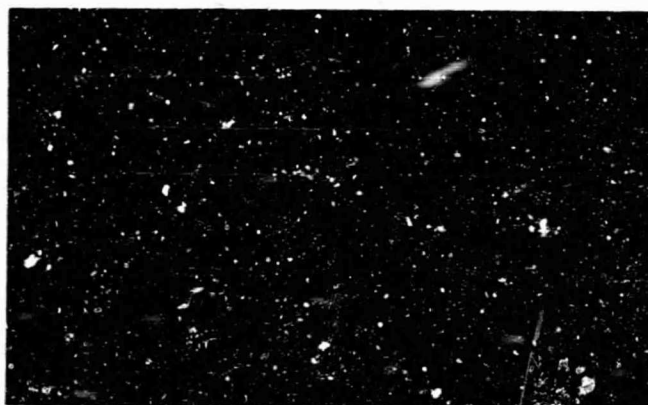
a SEQUENCE
b 1. CLEAR COMMAND
c 2. IN
d 3. SHIFT RIGHT
4. SHIFT RIGHT
5. SHIFT RIGHT
CLOCK 6. SHIFT LEFT
7. SHIFT LEFT
D1 8. SHIFT LEFT
D2 9. SHIFT LEFT
D3
D4 [MODE = 3]
Q1
Q2 VERT = 20 V/CM
Q3 HOR = 20 μ s/CM
Q4

C. SHIFT RIGHT, LEFT



Q1 SEQUENCE
Q2 1. SET
Q3 2. CLEAR
3. SET (COMMAND LINES
4. CLEAR NOT SHOWN)
Q4 5. SET
6. CLEAR

D. SET AND CLEAR



a SEQUENCE
b 1. CLEAR (d LINE NOT
SHOWN)
c 2. IN (ENTER 1 FROM
D1, D3)
CLOCK 3. ADD (D1 = 1 D3 = 1)
(D2 = 0 D4 = 0)
Q1
Q2 SCALE: VERT = 20 V/CM
Q3 HOR = μ s/CM
Q4

E. ADDITION USING MINIMUM CLOCK DELAY

02966 P

Figure 11. Typical Operating Waveforms (Sheet 2 of 2)

Figure 12 is an example of the form in which data are entered into the computer. This form was developed to enable the engineer to simply fill in circuit parameters at his desk, and then turn the form over to a technician for actual typing onto the teletype. This input program asks for (in order): number of independent nodes; number of dependent nodes; and number of resistors, capacitors, p-MOS, and n-MOS. It then asks for resistor and capacitor values and nodes connected to: active device parameter chip parameters; mode (mode 2 = internally programmed pulse; mode 1 = user programmed pulse); pulse parameters; initial node voltages (ask for DC SOLUTION if initial node voltages are unknown; and, finally, printer output intervals and total run time. The format of all of the data described above is changed and written on a simulated tape file called TAPE 3. To process these data, the user types /BATCH 01 02 03 and then tape 3 is batch processed by the main programs written on tapes 1 and 2. (See reference 1 for details of the main program.)

The circuit parasitic elements must be determined before computations can begin because they must be entered separately. The only major parasitic element in most LSI COS/MOS circuits is capacitance-gate to source-drain and drain-substrate (AC GND). Data on these parasitic capacitances is contained in an internal RCA report⁽²⁾. The data were checked by making a comparison with measured data on the capacitance of a simple device, the CD4007D. There was reasonable agreement. When these values were used in the program, they were checked further by running a simple inverter pair the actual width and length of the CD4007D. Pair delay was computed to be 32.3 nanoseconds. The data sheet specifies a minimum of 33 nanoseconds. Minimum speed is the best speed observed in production devices. The median or average speed of a device normally is approximately twice the minimum speed. Hence, knowing the minimum speed enables the median speed of a large lot to be obtained easily. Maximum speed, as specified in the data sheet, is an arbitrarily picked number for testing purposes, and is too imprecise a number to be used in computer studies. There is no practical upper limit on how slow an actual device can be (due to processing differences, defects, etc.), while minimum speed is firmly fixed by physical limitations.

⁽²⁾ D. Burr, Computer Analysis of Switching Speeds in Digital CMOS Circuits, CEL 658-2, August, 1968.

C-MOS TIME-SHARING COMPUTER FORM

F. CARLSON X2628

DIAL 452-2020 TO ACCESS COMPUTER

BEGIN 22

/ON 21245

01/06/69 13.13 22

READY

/CATALPG

*L-INPUT

L-TAPE04

*L-TAPE01

*TESTTAPE

*-PL0T

*L-TAPE02

L-TAPE99

READY

/CODE INPUT

1160

PLS ENTER CIRCUIT DESCRIPTION

TYPE # OF INDEPENDENT NODES, DEPENDENT NODES,
RESISTORS, CAPACITORS, PMOS, NMOS

NI=

ND=

NR=

NC=

NP=

NN=

RESISTOR # =VALUE(KOHMS), NODE, NODE

R 1=

R 2=

R 3=

R 4=

R 5=

R 6=

R 7=

R 8=

R 9=

R 10=

CAPACITOR # =VALUE(PF), NODE, NODE (ALL DEP. NODES MUST HAVE A CAPACITOR)

C 1=

C 2=

C 3=

C 4=

C 5=

C 6=

C 7=

C 8=

C 9=

C 10=

C 11=

C 12=

C 13=

C 14=

C 15=

C 16=

C 17=

/ON 21245

PRESS CONTROL

02057L

Figure 12. Example of Form Used to Enter Data Into the Computer (Sheet 1 of 3)

PMOS # = VT, CH. WIDTH (MILS), CH. LENGTH (MILS), D NODE,
G NODE, S NODE

PMOS 1=
PMOS 2=
PMOS 3=
PMOS 4=
PMOS 5=
PMOS 6=
PMOS 7=
PMOS 8=
PMOS 9=
PMOS10=

NMOS # = VT, CH. WIDTH, CH. LENGTH, D NODE, G NODE, S NODE

NMOS 1=
NMOS 2=
NMOS 3=
NMOS 4=
NMOS 5=
NMOS 6=
NMOS 7=
NMOS 8=
NMOS 9=
NMOS10=

ENTER FOLLOWING CHIP PARAMS----REL PERMITIVITY OXIDE, SILICON,
OXIDE THICKNESS (ANG), N MOBILITY, DONOR CONCENTRATION, P MOBI
LITY, ACCEPTOR CONCENTRATION, GAMMA

EOX=
ESI=
TOX=
UEFFN=
DONORS=
UEFFP=
ACCEPT=
GAMMA=

WHAT KIND OF RUN IS THIS?
MODE 1 OR MODE 2?

ENTER PULSE HEIGHT, WIDTH (SEC), TAU (SEC), & NODE

VH=
TP=
TAU=
NODE=

DO YOU NEED DC SOLUTION?

ENTER INDEP NODE VOLTS AND DEP NODE INITIAL VOLTS
NODE 1 MUST BE MOST +; NODE 2 MOST -
INDEP NODE = CONSTANT VOLTS
NODE 1=
NODE 2=
NODE 3=

02987L

Figure 12. Example of Form Used to Enter Data Into the Computer (Sheet 2 of 3)

DEP NODE = INITIAL VOLTS

NODE 4=
NODE 5=
NODE 6=
NODE 7=
NODE 8=
NODE 9=
NODE10=
NODE11=
NODE12=
NODE13=
NODE14=
NODE15=
NODE16=
NODE17=
NODE18=
NODE19=
NODE20=

ENTER CONTROL INFO---PRINT INTERVAL, STOP TIME,
INITIAL INTEGRATION INTERVAL, MAX EIGENVALUE(SEC OR 1/SEC)

TPRINT=
TSTOP=
TINTEG=
EIGEN=
STOP AFTER *.140
1160

IF, UNDER /CATALOG, A L-TAPE03 APPEARS, TYPE
/DROP TAPE03 · MACHINE WILL ANSWER "READY"

02857L

Figure 12. Example of Form Used to Enter Data Into the Computer (Sheet 3 of 3)

The test circuit was reprogrammed into the computer with slightly lower mobilities and with gate-to-output feedthrough capacitance eliminated. This capacitance was added instead to the capacitance from output to ground. This approximation was necessary to speed up computer run time; the small time constants involved caused the computer to go through an excessive number of integrations. The result of this run was a pair delay time of 35.5 nanoseconds. Since the data sheet specified a minimum of 33 nanoseconds, it was felt that this model was adequate for use in simulating a portion of the parallel processor. Table II lists the capacitance parameters used for the MOS model.

TABLE II. PARASITIC CAPACITANCE VALUES USED FOR
COMPUTER MODEL OF COS/MOS CIRCUITS

<u>Element</u>	<u>Parameter</u>	<u>Value</u>
Each pn pair	Capacitance of gate to substrate or AC ground	0.21 pF/1.1 mil channel width
Each p drain	Capacitance of drain to substrate or AC ground (includes feedthrough capacitance from gate)	0.23 pF/1.21 square mil
Each n drain	Capacitance of drain to substrate or AC ground (includes feedthrough capacitance from gate)	0.28 pF/1.21 square mil

The arithmetic, register, and control portions of the first stage were programmed to check switching speed. The sections actually programmed are shown logically in Figure 13 along with the waveforms at appropriate points. Times indicated are from start of control signal to leading edge of circuit waveform (50-percent points).



Figure 13. Computer Calculated Switching Times.

PRECEDING PAGE BLANK NOT FILMED.

SECTION IV COS/MOS FUNCTIONAL LOGIC GATING

A. GENERAL DISCUSSION

Any complex logic function expressed by a truth table or a Boolean equation may be translated into a hardware schematic by one of two methods: the building-block method, which utilizes standardized AND and OR circuits; or the functional-gating method, which utilizes components to synthesize the function.

The preference within the computer industry traditionally has been for the building-block approach, despite the fact that functional gating requires fewer components to realize the same logic. This preference has been due, in part, to the more complex appearance of functional logic schematics and to the subsequent greater difficulty caused by this complex appearance in debugging and troubleshooting. Compare the identical logic in Figures 14 and 15, for example, in terms of tracing a signal path.

Another traditional reason for avoiding functional logic has been the complexity of gate design; as long as gates are structured of passive resistors and capacitors as well as active devices, this reason is valid. Fan-in, fan-out, load current, drive current, offset voltage, and level shifting all must be considered in conventional gate design, and it has been found preferable to limit building blocks to a few well-designed types that are repeated over and over.

With the advent of complementary MOS array technology, the situation has changed. As will be shown in the following paragraph, the translation of Boolean expressions into COS/MOS functional logic is extremely simple and straightforward. The saving in components is impressive (40 components in Figure 14, for example, compared to 18 components in Figure 15). There is no penalty in circuit-design costs, because only p-and-n active devices are used. Each p-and-n pair constitutes a basic decision-making element, with a single,

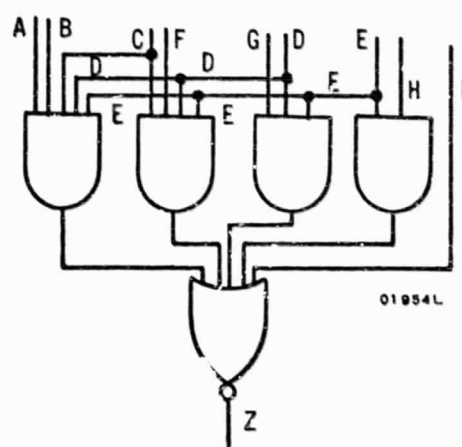


Figure 14. Building Block

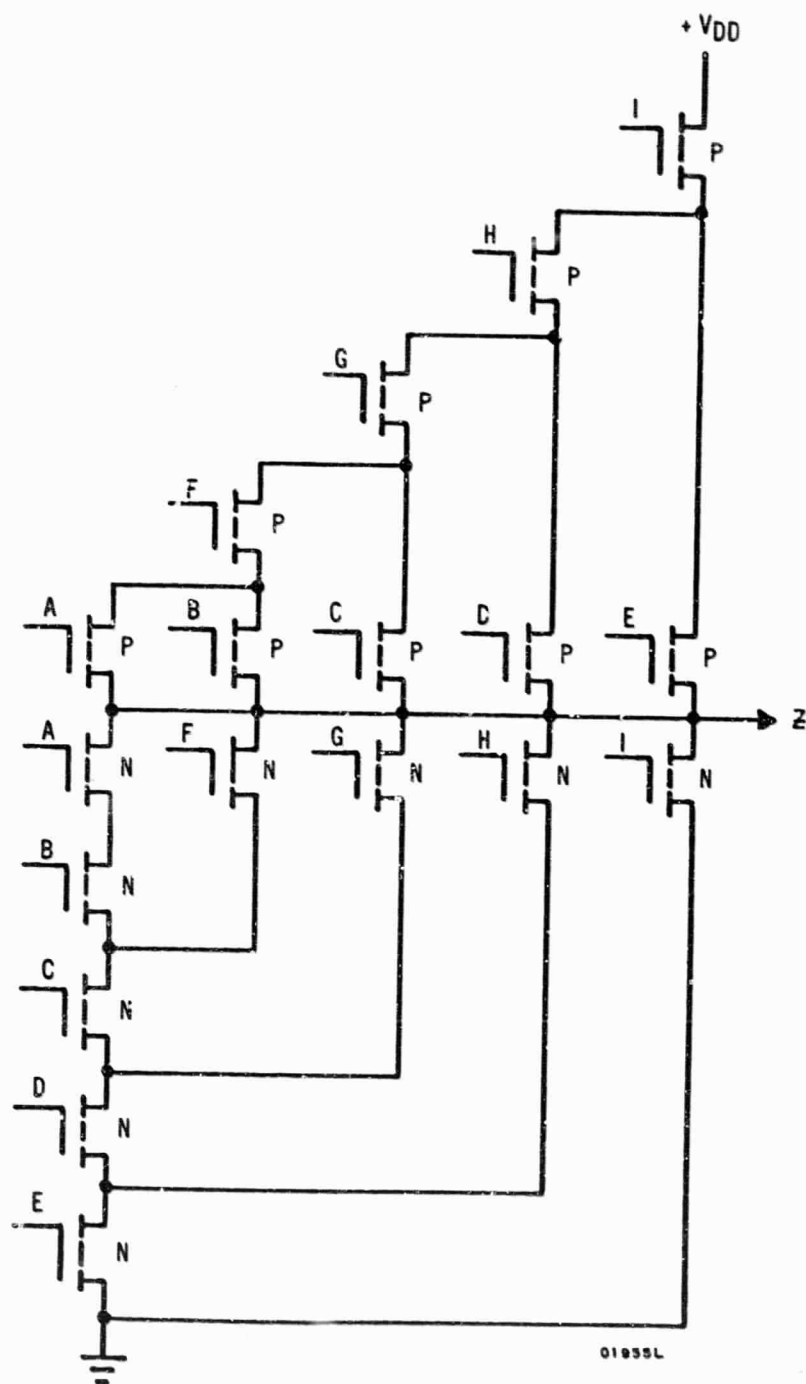


Figure 15. Functional Gating

common input to the gate of each device; they always operate as a pair, with one device on and the other off. There is no load current, the output is at either supply or ground potential, there is no level shifting (since inputs and outputs are at the same potential), and they may be combined in series or parallel configurations with other p-and-n pairs without restriction.

Since there is no access to individual components in an array configuration, the computer builder performs no debugging or troubleshooting on the packaged array. The array is a true "black box" (the pellet whimsically has been called a "black speck"); therefore, the logic can be presented arbitrarily by building-block gates or any other simulative symbols.

B. TRANSLATION PROCEDURE

A typical example will be used to demonstrate the simple step-by-step procedure for translating a Boolean expression into a functional logic schematic. The logic in building-block form is shown in Figure 14. The logic will be recognized as the circuit shown in sheet 7 of Figure 3, with A, B, C, etc., substituted for the actual signals.

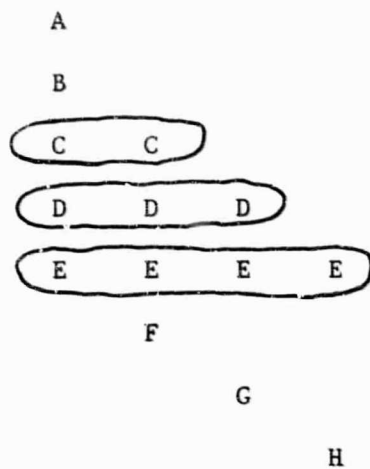
- (1) The Boolean expression is:

$$(A \cdot B \cdot C \cdot D \cdot E) + (C \cdot F \cdot D \cdot E) + (G \cdot D \cdot E) + (E \cdot H) + I = Z \dots \dots \dots (27)$$

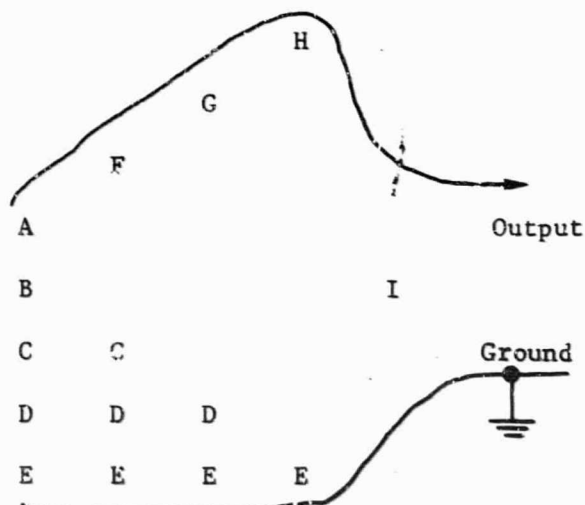
- (2) Rewrite the expression in vertical columns, using a separate column for each term and a separate line for each different variable:

A				
B				
C	C			
D	D	D		
E	E	E	E	
	F			
		G		
			H	
				I

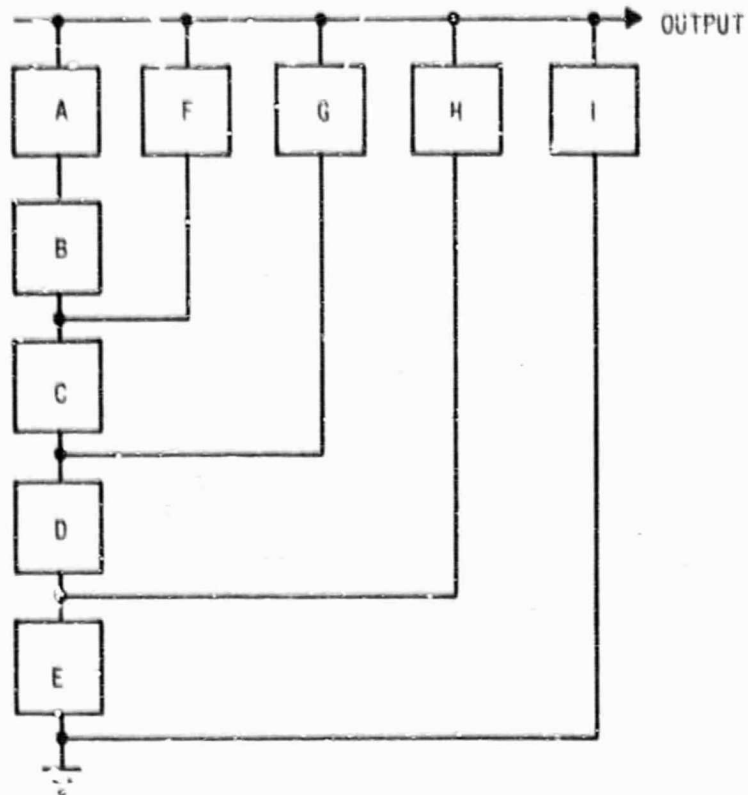
(3) Circle the common variables:



(4) Organize the n devices first. A series string of n units (each vertical column) represents an AND function, since all units must be on simultaneously to close the circuit path. Remember that one end of each n string must be at the output, and one end must be at ground. Since E is common to four of the five columns, choose E to be ground; therefore, F, G, and H will connect to the output, and I (sole variable in its column) must connect to both output and ground:



(5) Box the variables and combine common elements:



Step (5) completes the n section.

Every n unit must have an associated p unit, with the same input variable connected to each pair. The p units also must form a path from supply ($+V_{DD}$) to the output line.

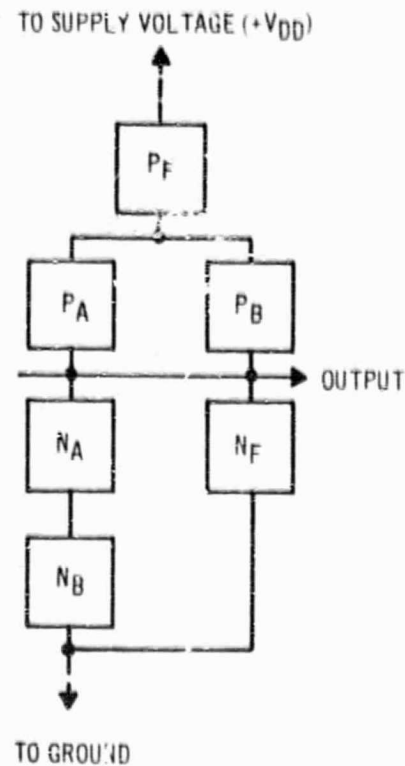
From De Morgan's theorem, the relationship between AND and OR is that of negated variables:

$$\overline{A \cdot B \cdot C} = \overline{A} + \overline{B} + \overline{C}$$

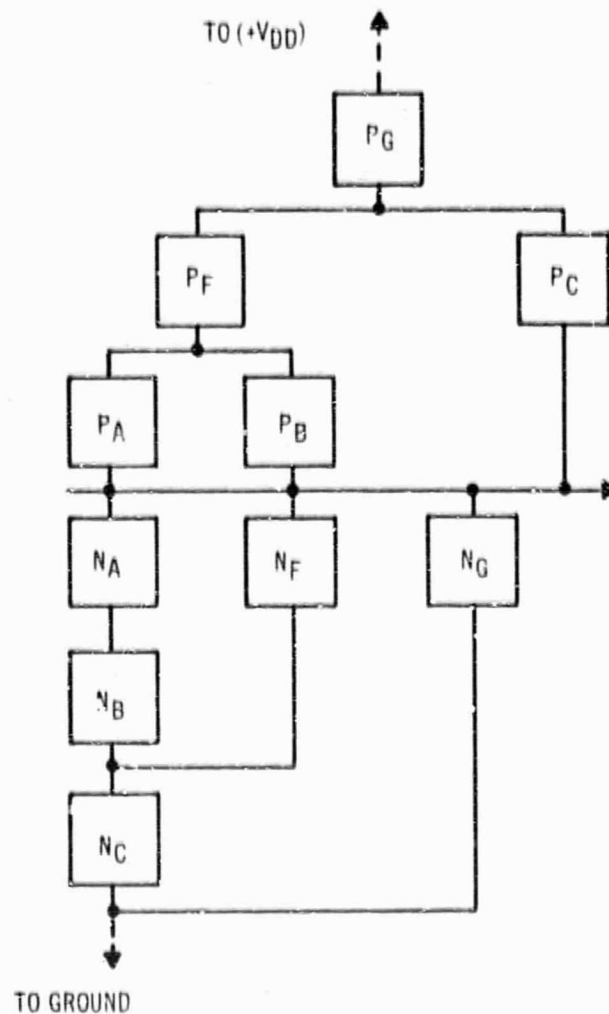
Since the variable that turns on an n device also turns off its companion p unit, the negation is automatic, and the n series string is equivalent to

X

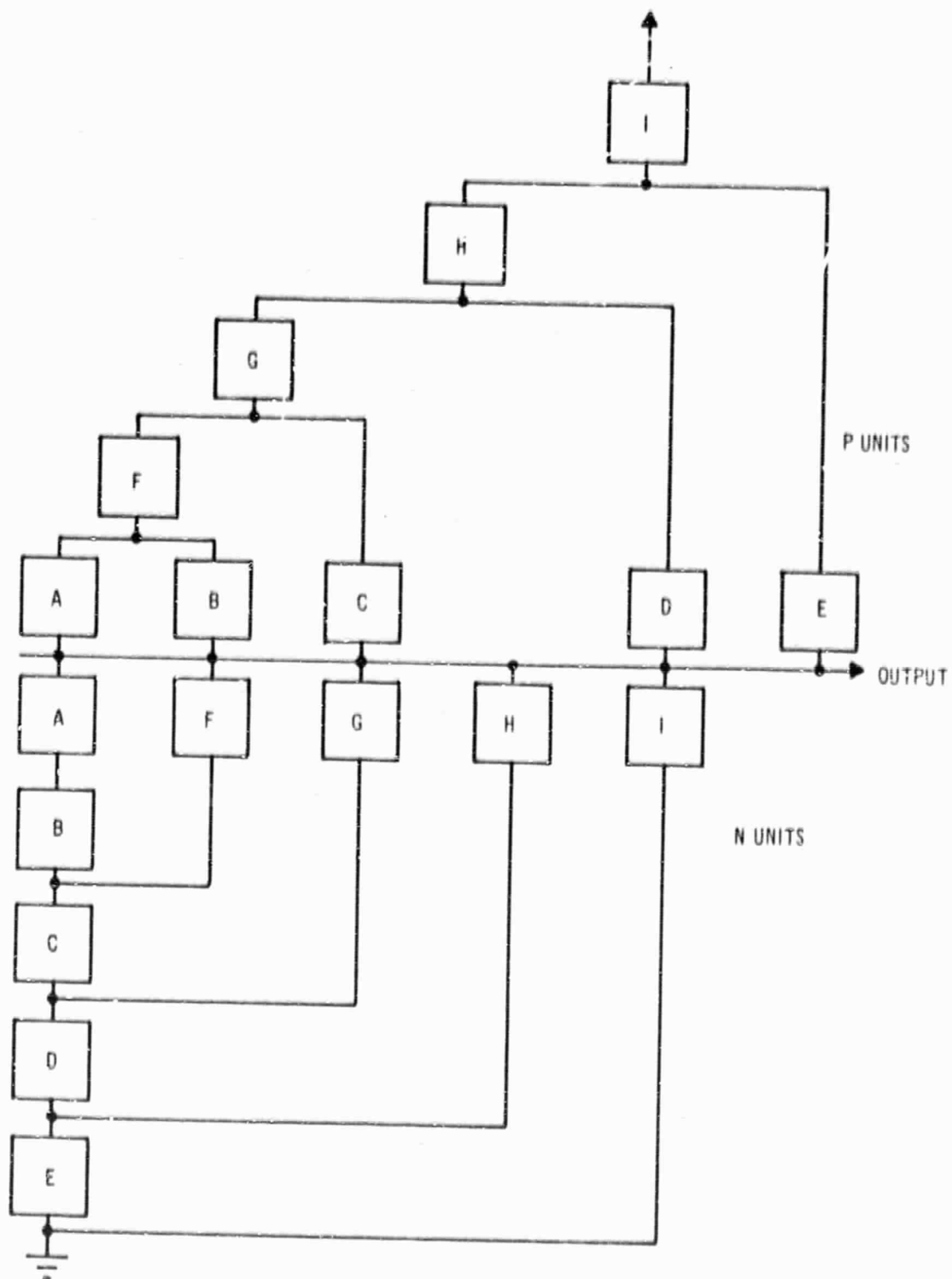
the p parallel units. To derive the p schematic from the n section which was completed in step (5), therefore, it is necessary merely to configure series units in parallel and parallel units in series. N_A is in series with N_B , for example, and the pair is paralleled by N_F . The p functions, therefore, are parallel P_A and P_B in series with P_F :



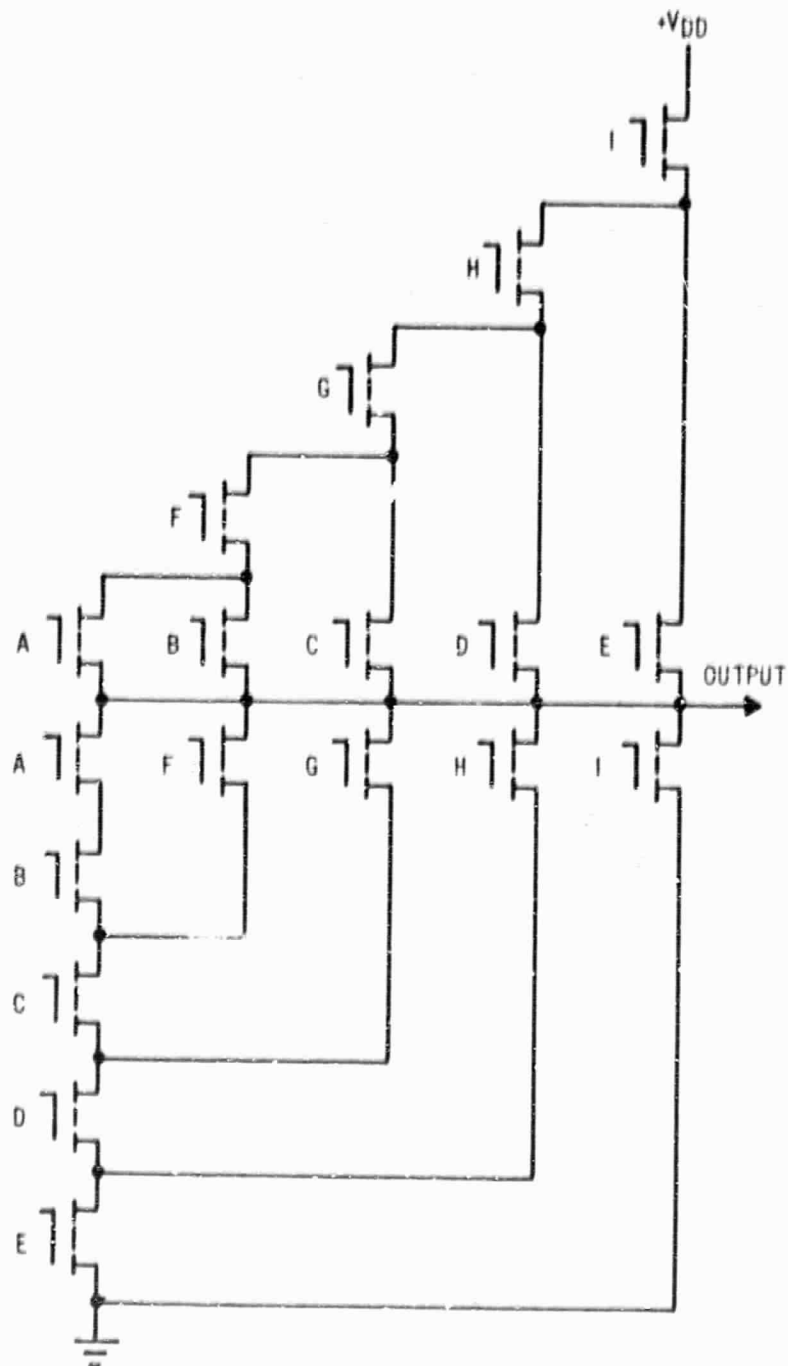
- (6) Similarly, N_C is in series with the N_A , N_B and N_F group, and this combination, in turn, is paralleled by N_G . Replacing series with parallel and parallel with series, P_C will be parallel to P_A , P_B , and P_F , and P_G will be in series with the combination:



- (7) Continue toward completion of the p section, replacing parallel with series and series with parallel:



(8) Redraw with conventional symbols:



SECTION V

DESIGN AND LAYOUT

A. INTRODUCTION

A monolithic four-stage parallel processor array (Figure 16) which forms the basic cell of a 16-stage parallel processor has been designed in partial fulfillment of Phase I requirements of NASA Contract NAS 5-11577 for the Goddard Space Flight Center. The parallel processor basically is a four-stage shift register that is capable of both serial and parallel access. The encoded control logic that is wired in facilitates two's complement addition, AND, OR, and EXCLUSIVE-OR logic operations, as well as right, left, and right cyclic shifts. It also provides for negative, zero, and overflow indications.

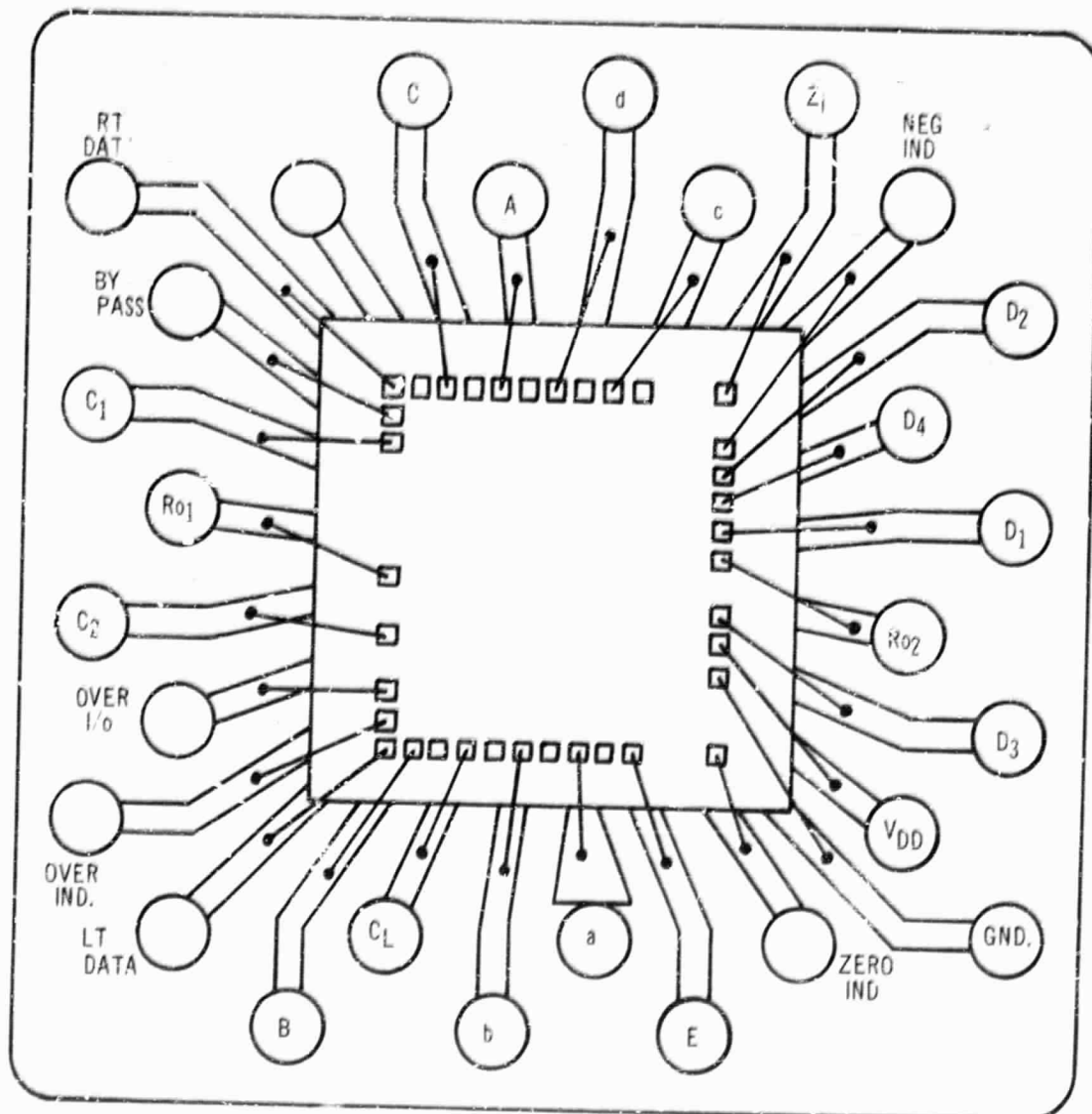
COS/MOS logic and COS/MOS circuit configuration are used to design an array that functionally is a computer on a chip except for memory and buffer circuits. It requires low quiescent power dissipation and provides computing times of less than a microsecond. The unique COS/MOS array design yielded a high packing density, making feasible a circuit containing 750 n-and-p channel devices packed in a chip size of 150 mils square. When completed, this will be the largest COS/MOS array fabricated in industry to date.

Figure 1 shows the lead requirement and Figure 17 shows the lead configuration for the four-bit slice to be packaged in a 28-lead flat-pack integrated circuit. Figure 5 is a schematic layout, indicating the interconnection of four four-stage chips to form a monolithic 16-bit parallel processor. The lead configuration has been chosen such that all the input encoding controls fan either on the top or the bottom of the slice, the input data lines fan to the right, and the remaining indicators and controls fan to the left. Since the control leads are duplicated at the top and the bottom of the slice, this arrangement makes feasible the extension to a 16-stage processor by two-layer metal, beam lead, or simple wire-bonding techniques. The wire-bonding tech-



Figure 16. Layout of the Parallel Processor

02959r



0296CL

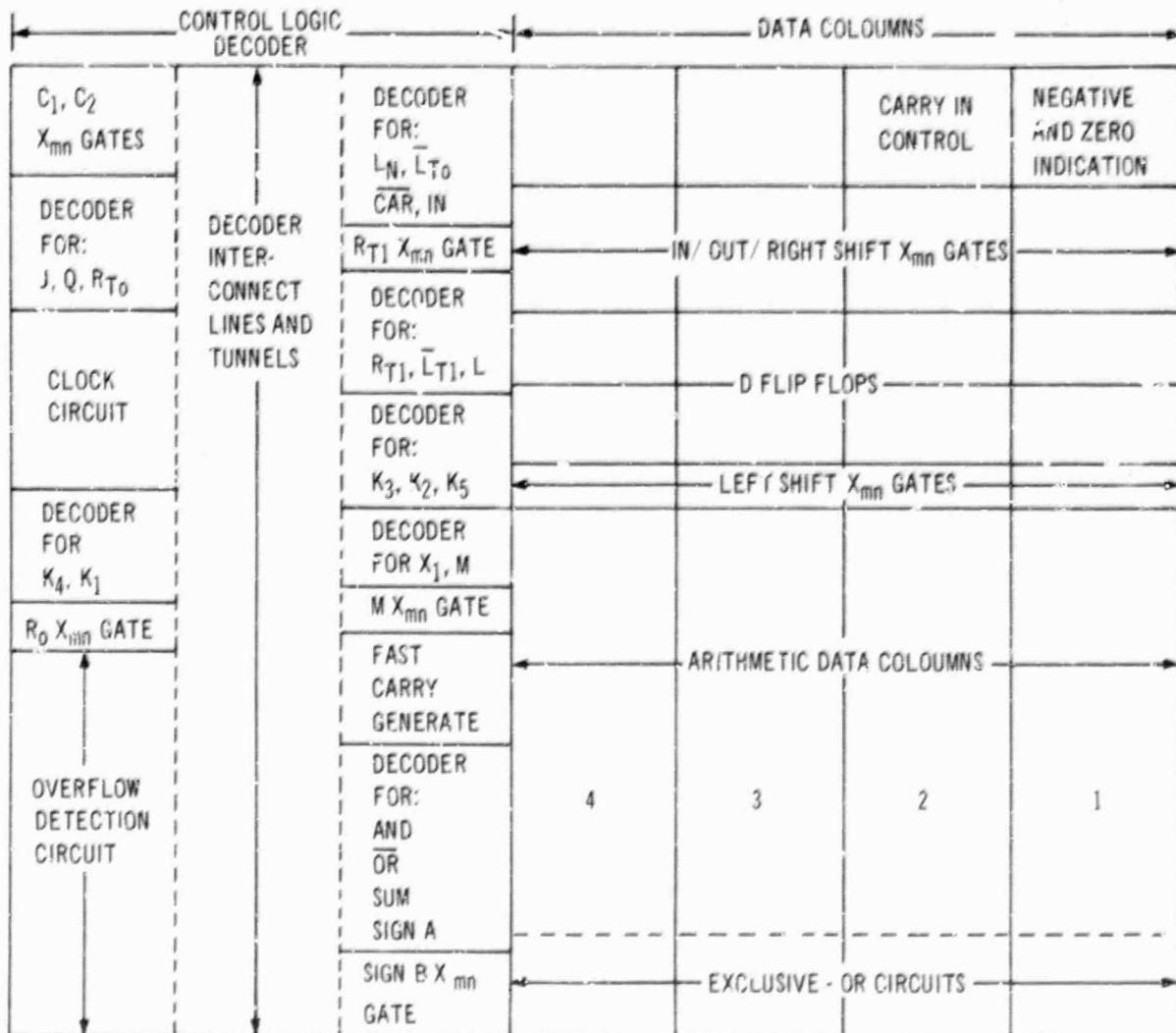
Figure 17. Lead Configuration for the Four-Bit Slice Using 28-Lead Flat Pack

nique in combination with a suitably designed package will be given primary consideration during Phase II of the contract. This layout makes possible monolithic interconnection by wire bonding either between adjacent chips or around nonoperating chips. A special ceramic 40-lead flat pack will be developed for the 16-stage processor during Phase II of the contract. Some preliminary designs are already under consideration.

3. LAYOUT DESIGN CONSIDERATIONS AND IMPLEMENTATION

The circuit schematic of the four-stage parallel processor is shown in Figure 2. The complexity of the array required a systematic design approach. The design philosophy centered on optimizing the array with a minimum of interconnect wiring and tunnels required to feed the decoded control lines to optimized functional logic blocks on parallel data lines. Thus, emphasis was placed on achieving a layout suitable for the control-logic decoder with vertical input instruction lines fanning in from both ends and the horizontal decoded control lines fanning out at appropriate levels to feed the logic blocks in successive columns. Figure 18 is a schematic of the layout with functional logic blocks. Conservative design tolerances, in terms of channel widths, clearances between contact and diffusion edge, clearances between metal lines etc., and particularly between p^+ and n^+ power lines, were followed to minimize leakage. This conservative approach resulted in a design trade-off between yield and chip size. Since precise alignment of all gates in the array is desirable, all the gates were positioned in one direction (horizontal with reference to Figure 18), necessitating a larger chip area. To further facilitate the extension of this layout to Phase II of the contract, bonding pads for input instructions and control lines were provided at both ends and at the top and the bottom of the processor. The 10-percent increase in chip area, which is the result of this feature, is a worthwhile sacrifice considering the advantage gained towards implementing Phase II.

Parts of the four vertical data columns are similar and repetitive. Advantage is taken of similarity and symmetry in the design such that two data columns share a P well, and the two control logic columns with their interconnecting cross tunnels also share a P well. In this way, the design contains a total of only three P wells and thereby minimizes well periphery.



C1961L

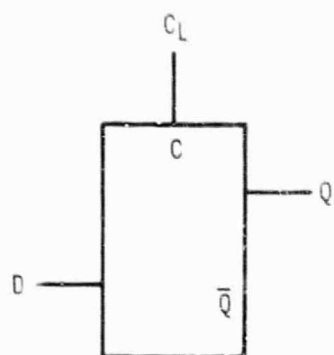
Figure 18. Functional Layout of the Parallel Processor

Two functional logic blocks, the D flip flop and the EXCLUSIVE OR, are discussed here to illustrate the development of the layout. Figure 19 is the logic block diagram of a D flip flop with its truth table. The specification calls for the D flip flop to transfer D input to Q output on low to high clock transition, and for the Q output to be buffered. Figure 20 gives the logic schematic of the two basic components, the transmission gate and the inverter, from which the D flip flop can be built. The complementary circuit configuration for these two components is shown in Figure 21. The transmission gates are closed or opened by the control signal, e.g., clock in this Figure, that is generated elsewhere in the control logic column. Signals from the control logic column feed several transmission gates, and the inverter required with each transmission gate is localized and paired. A method of wiring the combination of inverter and transmission gate is indicated in Figure 22.

A particular advantage of this wiring method is that a number of such building blocks can be stacked up. In this configuration the device source and drain diffusions are found along the vertical, while the signal interconnect lines are found along the horizontal with diffused power-supply connections. The block diagram representation for layout development of the D flip flop (Figure 23) exemplifies this approach. The transmission gates and inverters are positioned appropriately one above the other to facilitate a very compact layout. The composite layout that follows this scheme is illustrated in Figure 24. The cross-hatched areas represent p^+ diffusion regions and the dotted areas represent n^+ diffusion regions. The shaded horizontal areas represent metal interconnect lines. By removing either a cross-hatched or dotted tone from the contact area, which can be seen clearly except for the shading over it, this composite can be used to single out the different masks required for fabrication.

C. EXCLUSIVE OR

This functional logic block, which is repeated on each data line, is a case where the design methods described in the previous paragraph were combined with the functional gating technique discussed initially, and in detail, in



LOGIC BLOCK

D	C _L	Q	$\bar{Q}$
0	┐	0	1
1	┐	1	0
X	┐	Q	$\bar{Q}$

TRUTH TABLE

02969L

Figure 19. A Functional Logic Block: D Flip Flop

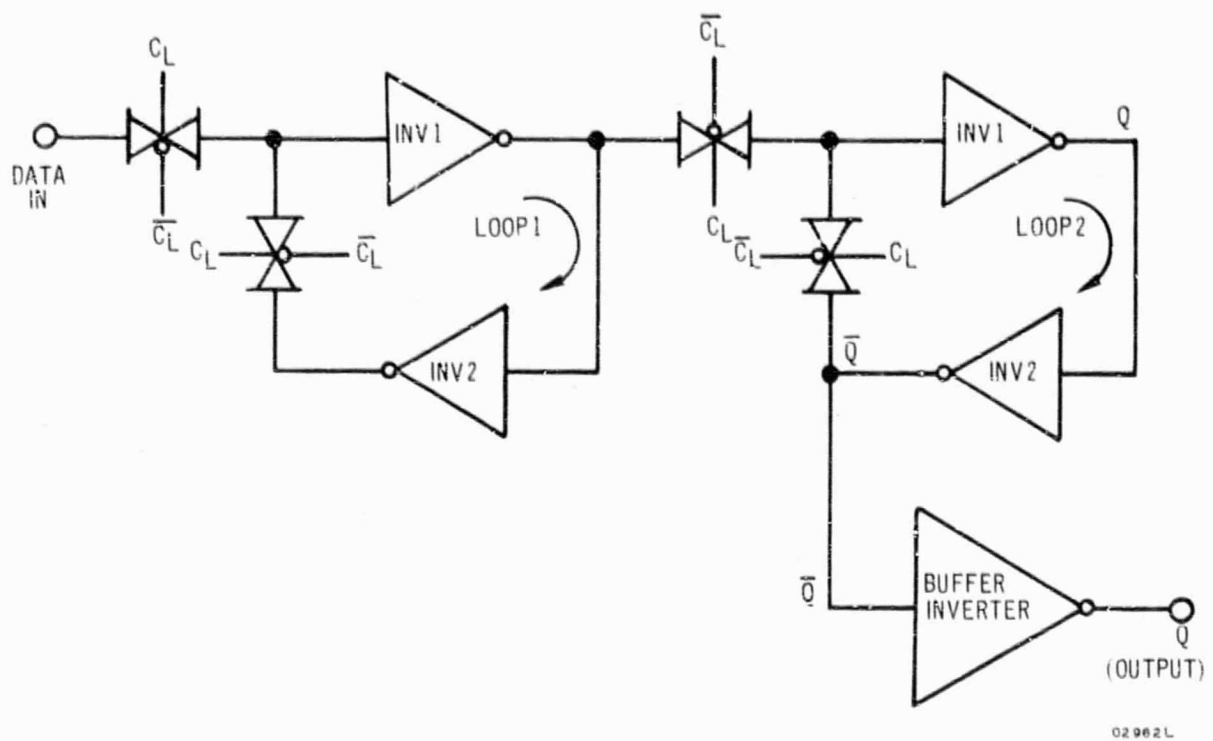
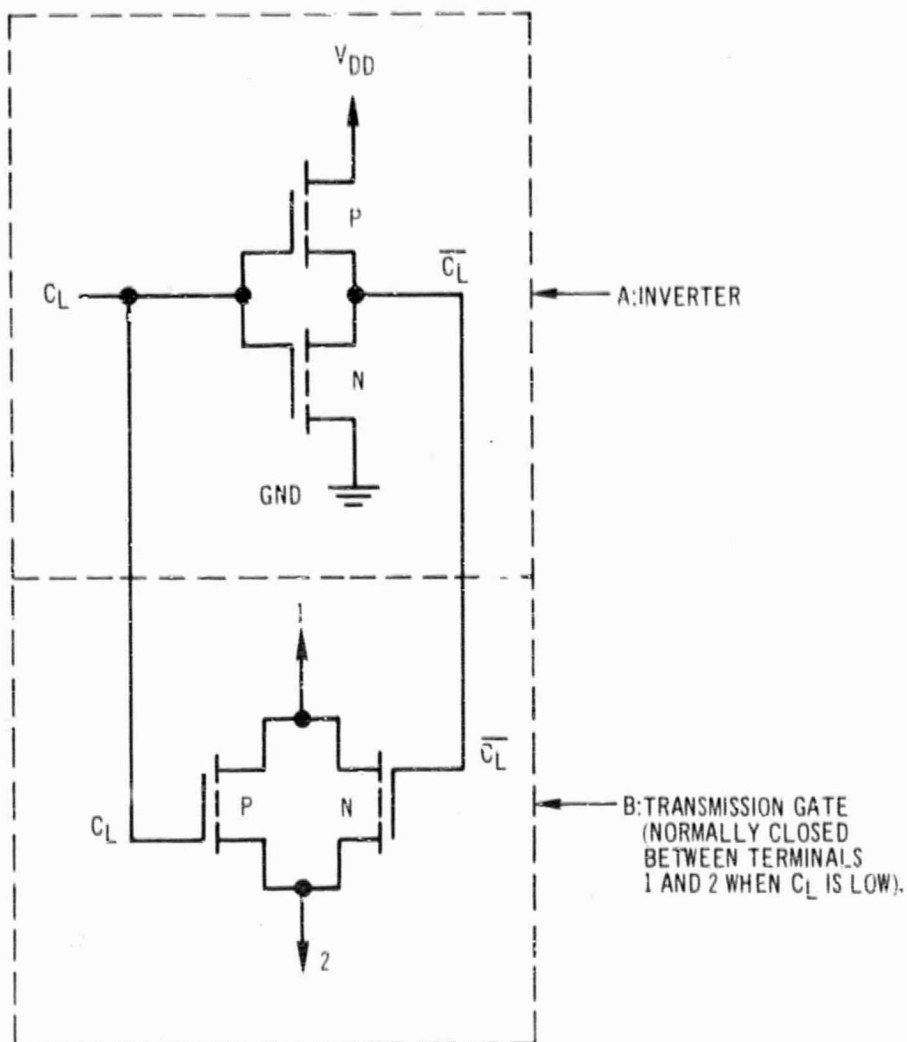
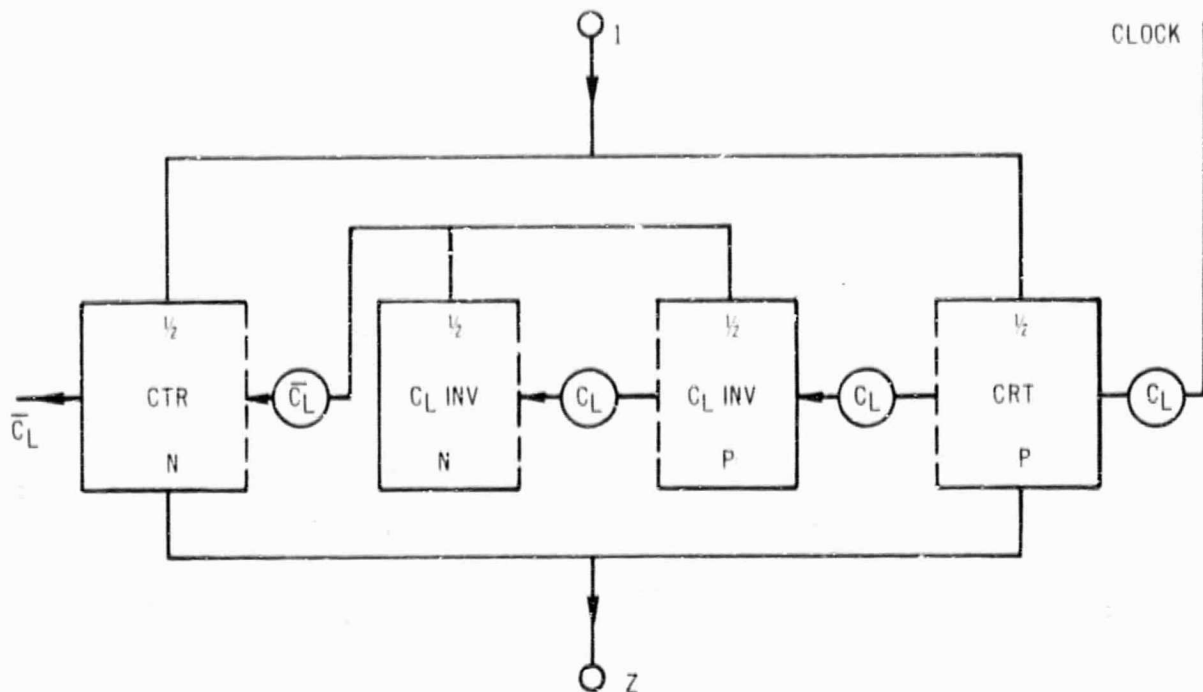


Figure 20. D Flip-Flop Logic, Schematic



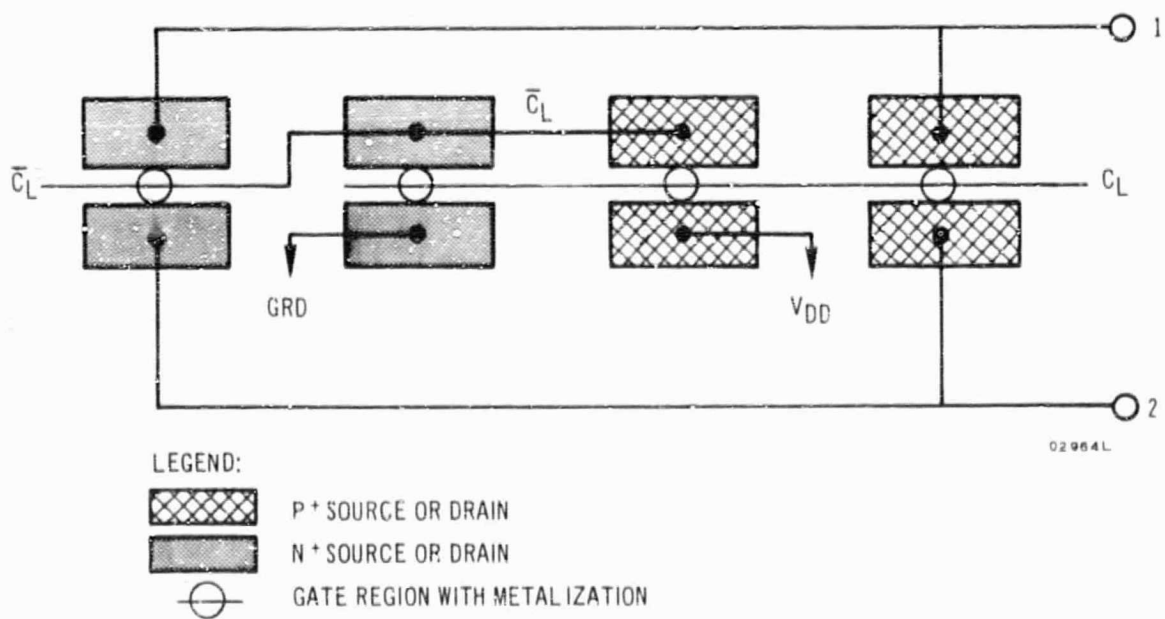
02953L

Figure 21. Transmission Gate and Inverter Circuit, Schematic



NOTE. CTR = NORMALLY CLOSED TRANSMISSION GATE (NORMAL WHEN THE CLOCK IS LOW)

A. BLOCK REPRESENTATION FOR LAYOUT



B. DEVICE TOPOLOGY

Figure 22. Transmission Gate Layout

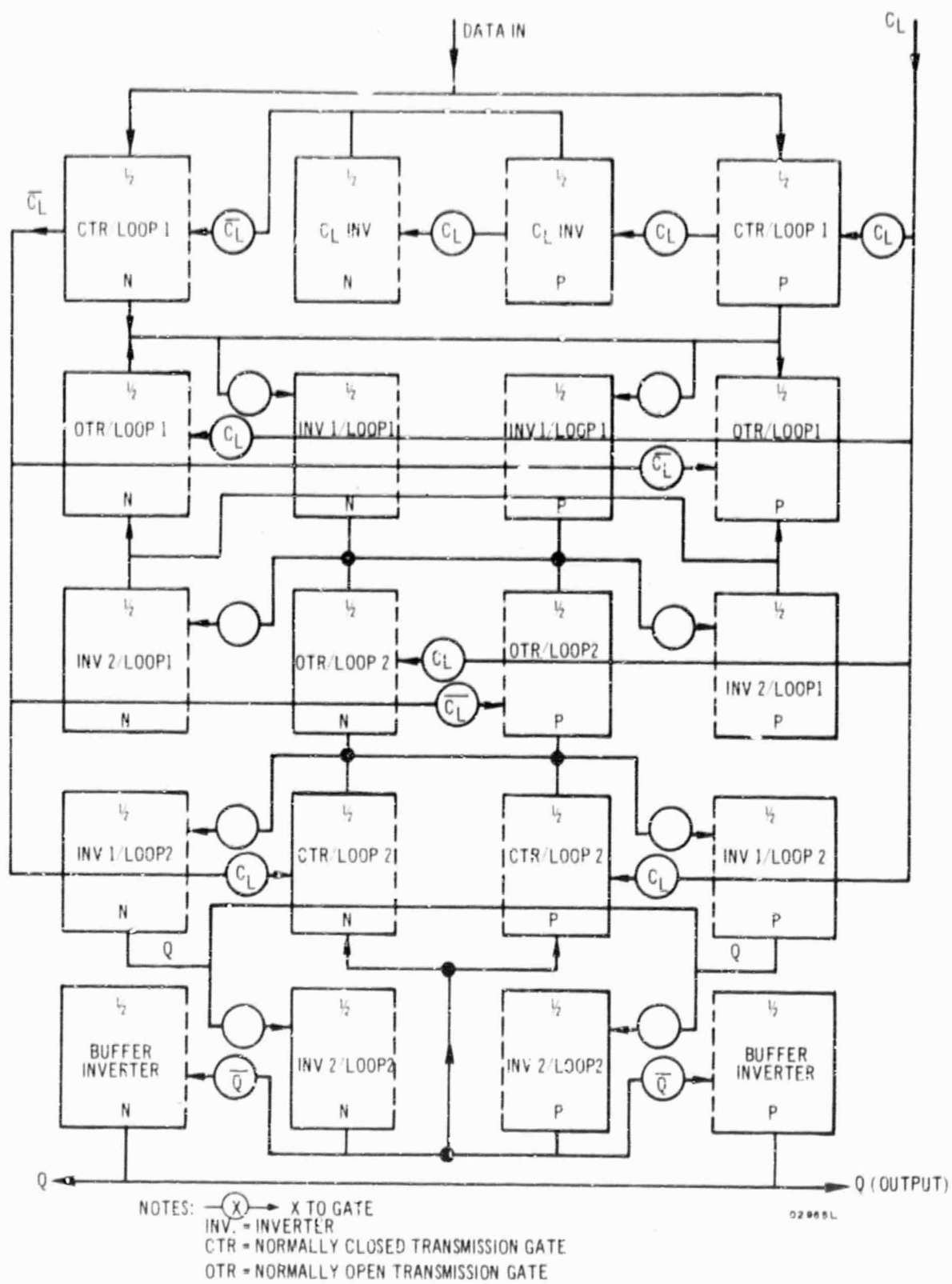
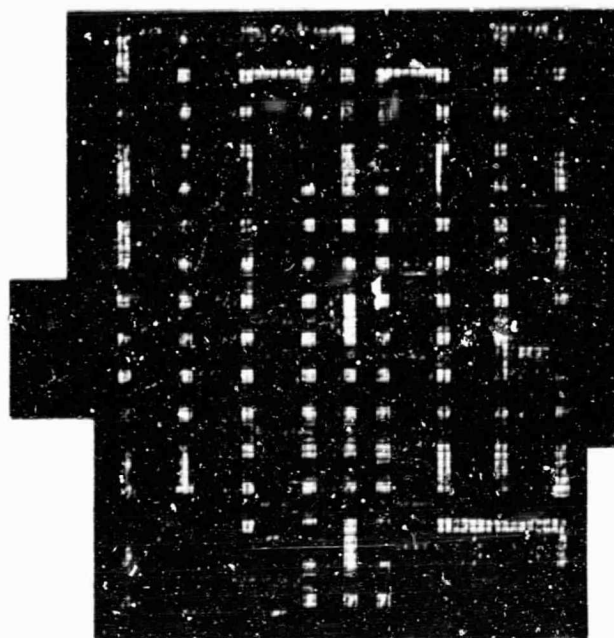


Figure 23. Block Diagram Representation for Layout Development of D Flip Flop



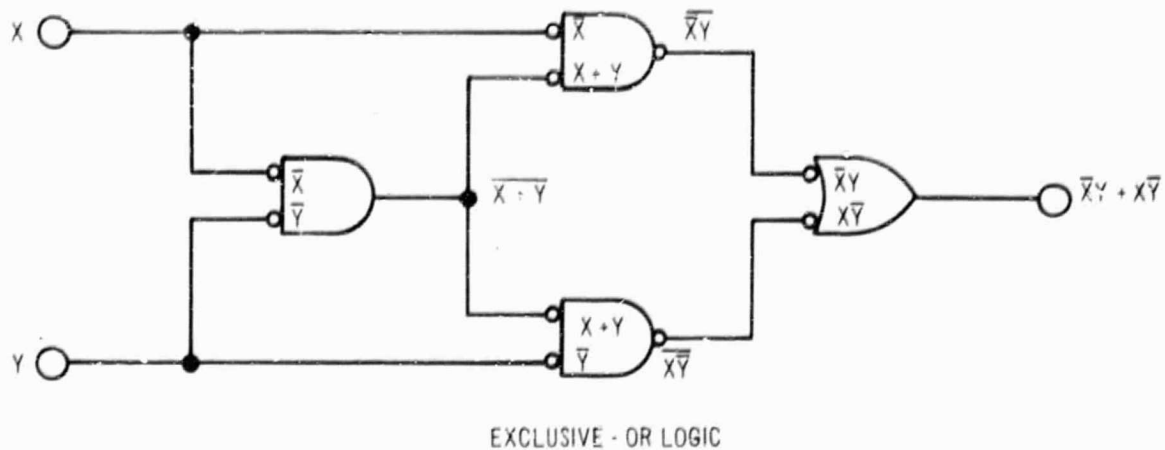
02956P

LEGEND

- : N⁺ REGION
- ⊗ : P⁺ REGION

Figure 24. D Flip-Flop Composite Layout

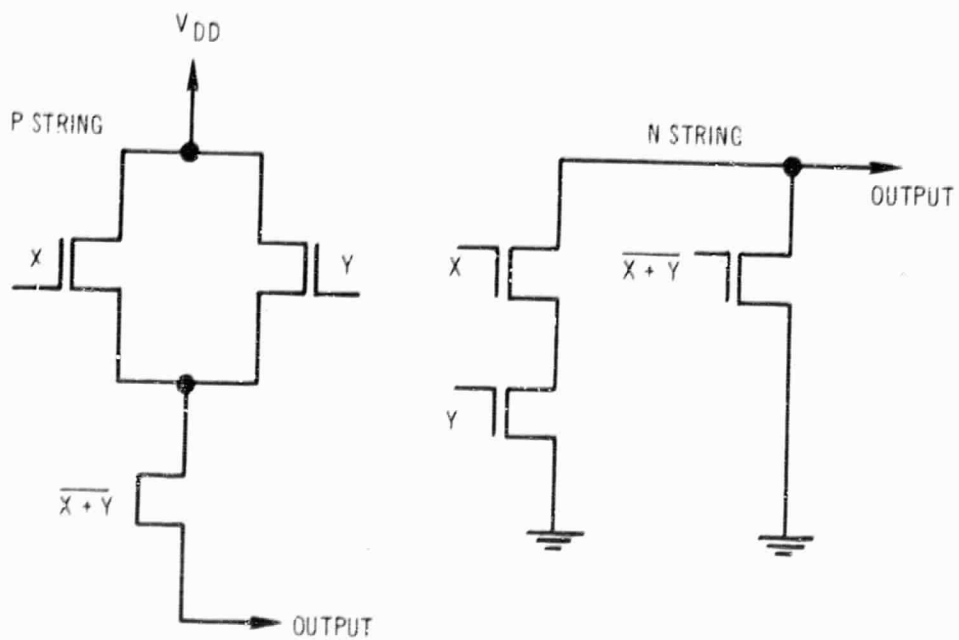
Section III of this report. The logic specified for this relatively simple functional block is as follows:



In addition to NOR and NAND, the logic shown above involves OR gates. OR gates would require a stage of inversion if complementary devices are to be used in assembling the logic. The device count necessarily would go up. To minimize the number of devices required to generate the function, the following technique, using functional gating, is applied.

$$\begin{aligned}
 \text{EXCLUSIVE-OR output} = S &= \bar{X}Y + X\bar{Y} \\
 &= \bar{X}Y + X\bar{X} + X\bar{Y} + Y\bar{Y} \\
 &= \bar{X} (X+Y) + \bar{Y} (X+Y) \\
 &= (\bar{X}+\bar{Y}) (X+Y) \\
 &= \overline{XY} (X+Y) \\
 &= \overline{XY} \overline{\overline{(X+Y)}} \\
 &= \overline{(XY) + (X+Y)}
 \end{aligned}$$

The final reduction is a NOR output which can be functionally designed as indicated below:



Putting the p string and the n string together and generating $\overline{X+Y}$ from a simple NOR gate, the EXCLUSIVE-OR circuit becomes

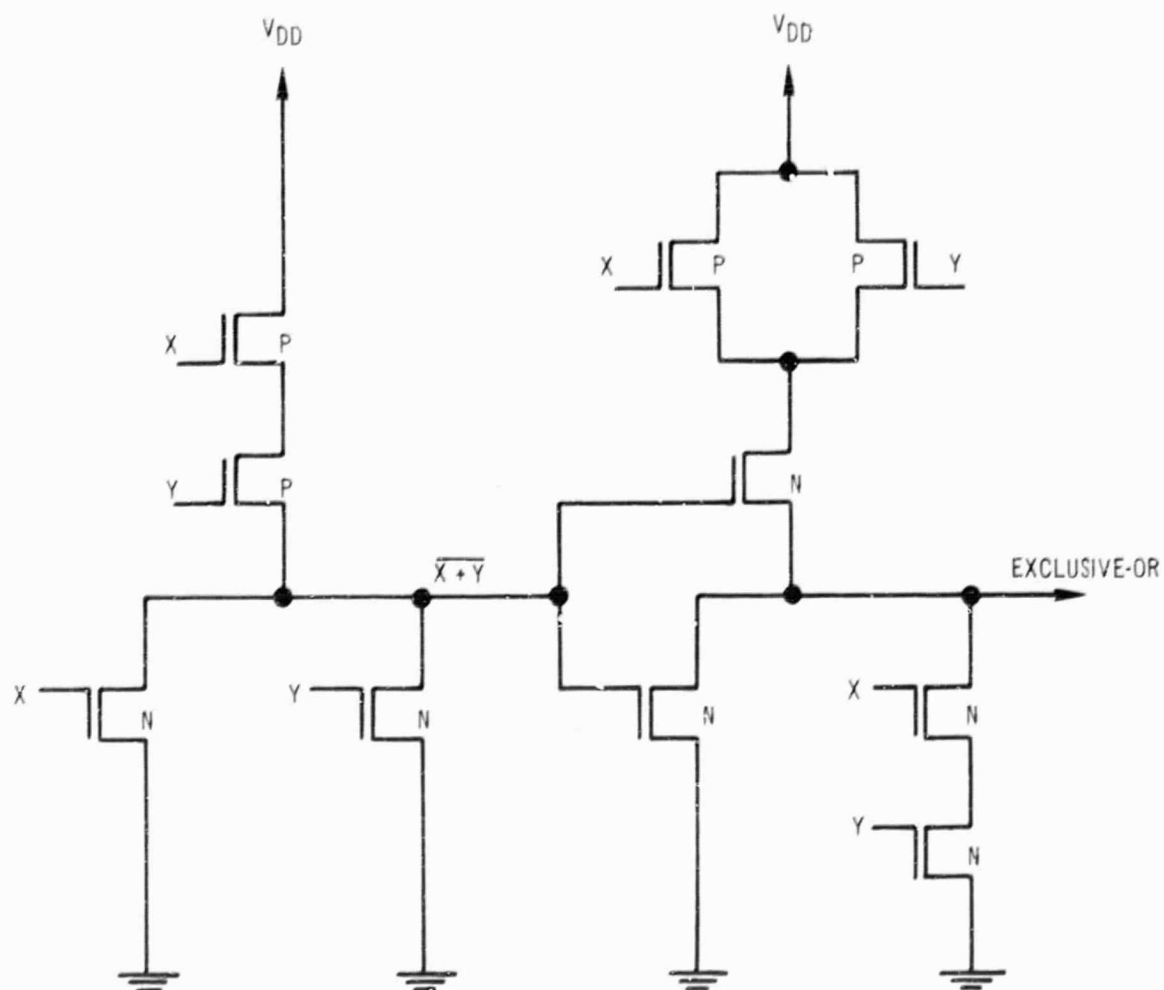
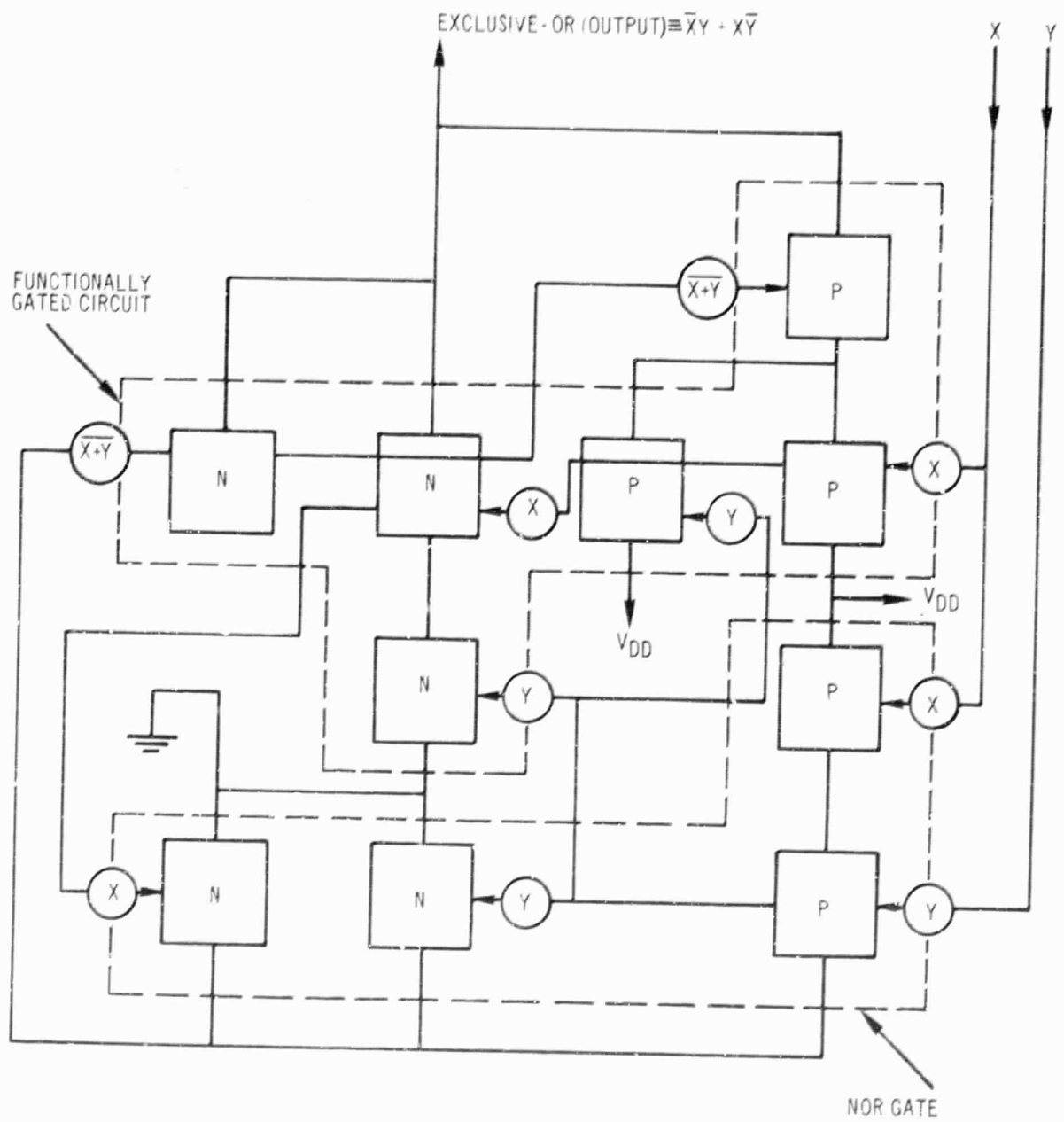
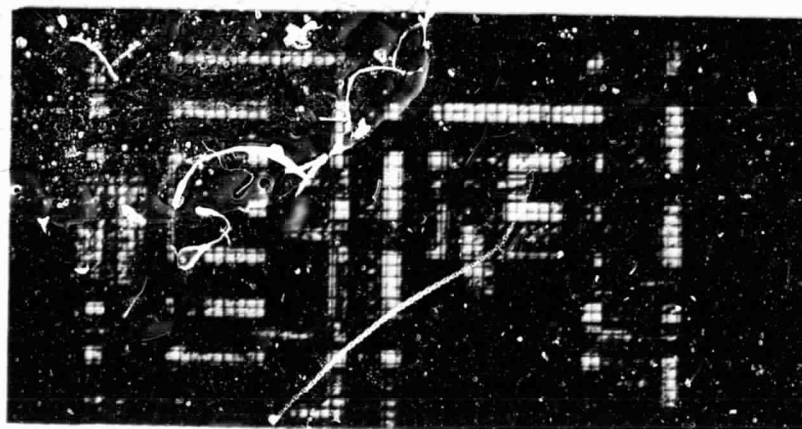


Figure 25 illustrates the device topology for the functionally designed EXCLUSIVE-OR circuit. Figure 26 is the composite layout generated for this logic functional block. The composite employs the design considerations outlined previously for the D flip-flop circuit.



02957L

Figure 25. Block Diagram For Exclusive-Or Layout Development



LEGEND

02958P

- : P⁺ REGION
- : N⁺ REGION

Figure 26. Exclusive-Or Composite Layout

PRECEDING PAGE BLANK NOT FILMED.

SECTION VI PROCESSING

A. INTRODUCTION

Processing of complex, LSI, COS/MOS semiconductor devices requires an unusual degree of manufacturing sophistication. Although processing on the parallel processor circuits has not started yet, it is anticipated that high-yield-fabrication procedures will account for a significant portion of the engineering effort expended in this program. The following processing considerations, pertinent to this program, will require unusual engineering attention:

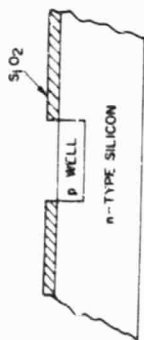
- a. The successful fabrication of 750-component LSI chips for Phase I and 3000-component LSI chips for Phase II would represent a significant challenge for any integrated-circuit technology. The challenge is great in the COS/MOS technology where processing sequences are highly complex.
- b. Successful fabrication of COS/MOS devices requires ultraclean processing to avoid mobile charges in the gate oxide and/or gate-oxide defects.
- c. Unusually high-accuracy alignments and accurate geometric control of components are essential to ensure proper device operation.
- d. The large chip size (0.145 inch by 0.155 inch) is at the limits of present photomask art for generating images without distortion and high resolution.
- e. High-yield techniques are essential to achieve the objectives of the program at a reasonable yield. Special handling and photolithographic fabrications appear to be essential.

B. COS/MOS FABRICATION PROCESS

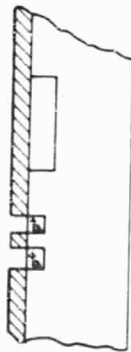
The first step in the fabrication process is shown in A, Figure 27. In this step a low concentration "well" is diffused into the homogeneous N-type silicon which is typically 1 to 2 ohm-cm material. Next, the high concentration regions which form the source and drain of the P-channel devices are diffused as shown in B, Figure 27. In a similar manner, high concentration n-regions are diffused within the well to form the source and drain for the N-channel devices as shown in C, Figure 27. The physical separation between these diffused regions determines the channel length and directly affects device characteristics.

Up to this point, the fabrication of complementary MOS devices is similar to the fabrication of conventional bipolar planar devices, i.e., processing steps for both types of device include silicon dioxide growth, photoresist application, oxide etch, high-temperature deposition and diffusion of impurities, and masking operations. At this point, however, MOS fabrication enters a critical phase because in MOS construction the purity of the silicon dioxide directly over the channel and beneath the metal gate is far more critical than in bipolar devices. Any mobile charges in this oxide due to sodium or calcium impurities move under the influence of the high field between the gate and substrate and can induce uncontrolled leakage currents between the source and drain. These leakage currents were a major obstacle in the development of early MOS enhancement devices.

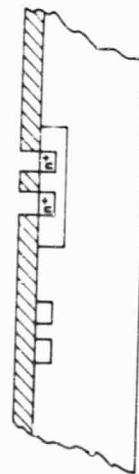
To ensure that a pure channel oxide will exist in the critical region, the existing oxide is etched back to the surface in the source-channel-drain areas as shown in D, Figure 27. The wafer is then thoroughly cleaned, and a thermal oxide is grown in a specially prepared and maintained "clean" furnace. A small amount of phosphorous-doped oxide is usually added as a "getter" during this process to immobilize any stray ions that may intrude on the surface. This step is illustrated in E, Figure 27. Precautions are also taken during the metal evaporation phase to ensure that no contaminants reach the surface of the channel oxide beneath the gate metalization. These special precautions require the use of ultra-pure aluminum wire and a scrupulously clean vacuum chamber. The contact-opening and metal-etch steps are shown in F, Figure 27.



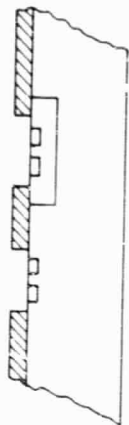
A. DIFFUSION OF A "WELL" INTO N-TYPE SILICON.



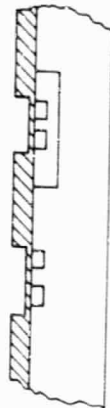
B. DIFFUSION OF P-TYPE REGIONS.



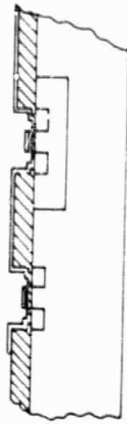
C. DIFFUSION OF N-TYPE REGIONS WITHIN THE P WELL



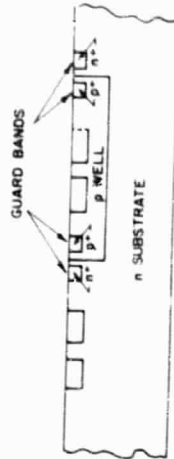
D. ETCHING OF OXIDE IN SOURCE-CHANNEL-DRAIN AREA.



E. GROWING OF THE THERMAL OXIDE WITH PHOSPHORUS DOPING ADDED.



F. CONTACT OPENING AND METAL ETCH.



G. HEAVILY DOPED N AND P GUARD BANDS

01948L

Figure 27. COS/MOS Fabrication Process

C. CAPACITANCE-VOLTAGE CURVES

The capacitance-voltage measurement technique has been valuable in developing the clean-oxide clean-metal process. Briefly, this technique involves the measurement of the capacitance due to the oxide dielectric in series with the capacitance of the surface space-charge layer. Ideally, the induced junction is due solely to the field between the metalized electrode (or channel if this were a gate) and the substrate. As explained previously, however, mobile charges in impure oxide can induce an inversion layer and, therefore, change the bias voltage at which the series capacitance appears. Figure 28 shows capacitance-voltage curves for clean and impure oxides after one minute at 300°C with 10 volts positive bias.

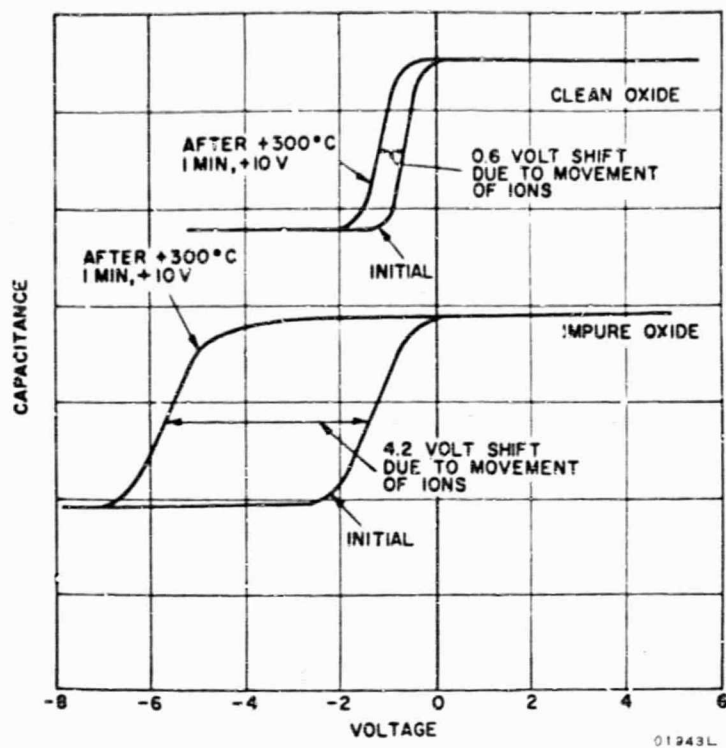


Figure 28. Capacitance-Voltage Curves for Clean and Impure Oxides

PRECEDING PAGE BLANK NOT FILMED.

SECTION VII

TESTING

A. INTRODUCTION

A significant dilemma in the implementation of logic is the problem of verifying the proper operation of the fabricated arrays. Not only is it necessary to fabricate and test volume quantities of an array, but it is also essential to isolate any initial array design faults.

B. INITIAL DESIGN FAULT ISOLATION

When the design and the sample fabrication of an array is complete, it is necessary to verify that all design phases have been performed in a proper and logical fashion. To aid in "debugging" the parallel processor array, a number of internal test probe pads were included, in addition to the ordinary bond pads. These internal test points were strategically located to provide ease in trouble shooting should a failure be revealed in the logic design or in the mask design. By probing these pads it is possible to isolate areas of the array and verify their proper operation. By using these pads, therefore, a fault can easily be located, at least to one gate structure and often to the exact device. This capability greatly facilitates the process of locating a logic or mask error without elaborate rechecking.

C. PRODUCTION TEST PROCEDURES

When it is determined that the masks are correct and functional arrays are producible, a test matrix is required which will exercise each array thoroughly to select functionally usable chips. In this procedure, only the 27 bond pads are accessible to the test words in which the inputs will automatically be sequenced and the outputs verified.

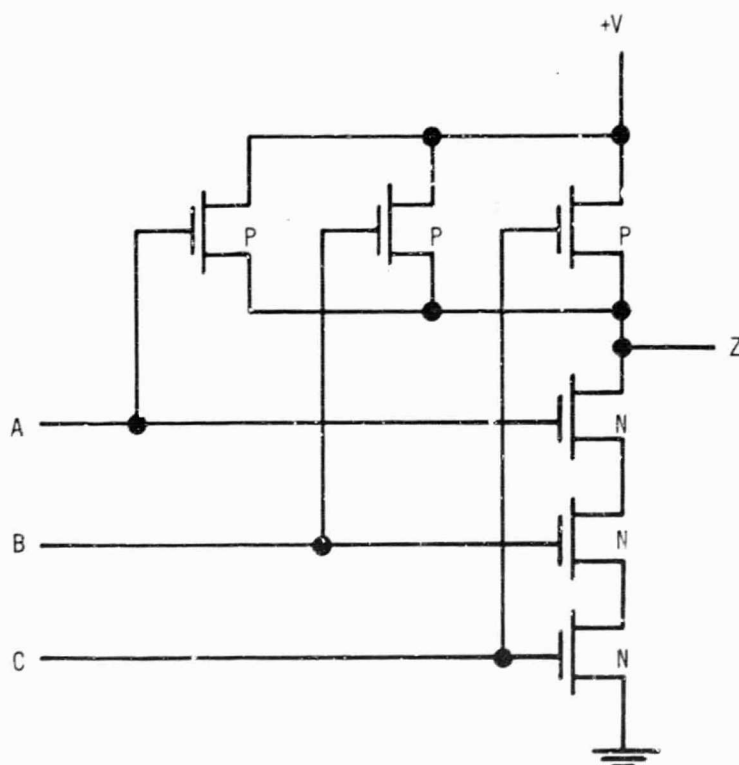
In order to exhaustively test the array, all possible combinations of the 15 required inputs would be applied. This would require 215 over 32,000 test

words to be applied. All outputs would be monitored for each test word. This procedure results in an exorbitant period of test time to provide an indication of array operability. Hence, it behooves the designer to devise a short, concise test sequence which is practical in time, yet will supply assurance that the array is indeed operating as required.

The basic concept leading to the reduced test matrix for the parallel processor involves the verification of proper operation of each pn device pair. To achieve this, each logic gate structure must be made active in all possible modes independently, then made inactive in as few conditions as possible where it can be shown that each device is capable of controlling the logic output.

The procedure is illustrated in Figure 29 where four (4) tests are all that are required to test a three-input NAND gate. In the first test, if any of the p devices are "on," the output will be false, or there will be a DC current path from +V to ground which can easily be detected in the power tests. In the second, third, and fourth tests, an n device that is "on" will be detected in the power test, and the p device that is "off" will show up as a logic error. Also, any error resulting from metalization shorts will be detected in power tests. The extension of this procedure to functional logic structures is quite simple and provides an even greater reduction of "tests per input" than for the NAND/NOR structure. (See Figure 30.)

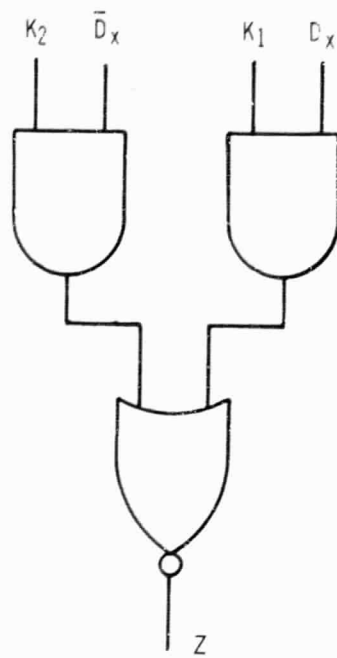
The testing of a bilateral current device (transmission gate) requires a different test technique. In testing the conducting state of a transmission gate in the parallel processor, a sensitized path through the transmission gate was required to change the state of a D flip flop. The changing-of-state requirement is derived from the fact that if the transmission gate was off, the D flip flop input would be open circuited and the charge-storage property of the device would prevent a change in the flip flop for millisecond time durations. To test for nonconducting states, signals are applied such that each data terminal of the transmission gate in question has active signals present which are of opposite state, i.e., "1" and "0." If the gate is conducting, a low resistance path, 10^4 ohms, will be present and can be tested for in quiescent power checks. The quiescent power of a chip should be less than



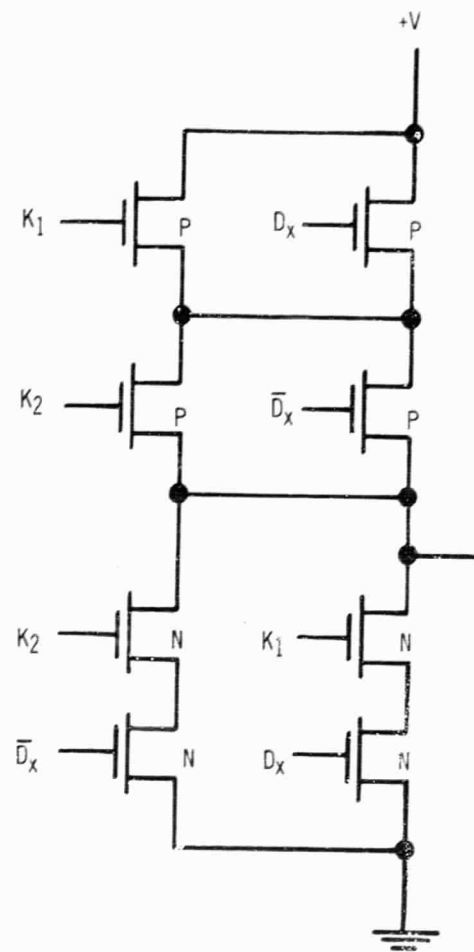
A	B	C	Z	TESTS
+V	+V	+V	G	A _n B _n C _n ON, A _p B _p C _p OFF
G	+V	+V	+V	A _p ON, A _n OFF
+V	G	+V	+V	B _p ON, B _n OFF
+V	+V	G	+V	C _p ON, C _n OFF

NOTE: A_n ⇒ n DEVICE CONTROLLED BY A
A_p ⇒ p DEVICE CONTROLLED BY A

Figure 29. Test Requirement For Three-Input NAND Gate With Inputs Active High.



K_1	K_2	D_x	Z	TESTS
1	0	1	0	$K_{1n}D_{xn}$ ON, $K_{1p}D_{xp}$ OFF
0	1	0	0	$K_{2n}\bar{D}_{xn}$ ON, $K_{2p}\bar{D}_{xp}$ OFF
1	0	0	1	$K_{2p}D_{xp}$ ON, $K_{2n}D_{xn}$ OFF
0	1	1	1	$K_{1p}\bar{D}_{xp}$ ON, $K_{1n}\bar{D}_{xn}$ OFF



02571L

Figure 30. Test Requirement for Function Gate

12 microwatts and the leakage from two opposite state inverters through a transmission gate would be ($P = E^2/R = 1000^2/3 \times 10^4 R = 3$ milliwatts) more than two orders of magnitude above the normal quiescent power. This approach provides a good tolerance margin for detecting this type of failure.

The utilization of this approach allows the testing of parallel data paths simultaneously. Analysis of the individual gate structures along the serial path is achieved by establishing logic paths to and from the structure in question so the required gate input conditions are directly manipulated by the data inputs, and the gate output is directly monitored at the array data outputs.

D. PARALLEL PROCESSOR TEST PROCEDURE

The test procedure (Table III) for the parallel processor was devised using essentially two semi-independent test procedures: one for the processing portion of the logic and one for the control portion of the logic. In testing the processing portion of the array, it is assumed that the control logic is functioning properly, and the control portion testing is based upon outputs of control gates being monitored by using conditions on the inputs to the processor portion, which is assumed correct, to define logic paths. This is essentially a process of establishing "logic paths" to and from the structure to be tested.

The test matrix table has been configured in such a manner as to provide a complete picture of the relationship between a given test word and the specific device structure that is being functionally tested. The information contained in the right-hand column enumerates which particular gate structures are under test and whether these specific gate structures are being programmed for an active or for a specific inactive state to exist at the outputs. Thus, it can be seen that the table provides the user with any and all information that is required to functionally test out a given chip.

With reference to the test matrix table, the first 26 test words are tests of the control portion of the array. For the majority of tests the array is in mode III and all of the 16 processor instructions are examined. For each

PRECEDING PAGE BLANK NOT FILMED:

TABLE III. TEST MATRIX FOR THE P

[illegible][illegible]

SEQUENCE OF EVENTS:

- 1) $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$
- 2) $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$
- 3) $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$
- 4) $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$
- 5) $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$
- 6) $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$
- 7) $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$
- 8) $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$

NOTES: 1)

- NOTES: 1) TEG - Transmembrane glycoprotein
- 2) The α and β subunits of K_{ATP} are the transmembrane proteins. The γ subunit has two extracellular loops.
 - 3) The α subunit is the main pore-forming subunit. It is a β -barrel structure. The β subunit is the γ subunit. It is a β -barrel structure.
 - 4) The α subunit is the main pore-forming subunit. It is a β -barrel structure. The β subunit is the γ subunit. It is a β -barrel structure.
 - 5) The α subunit is the main pore-forming subunit. It is a β -barrel structure. The β subunit is the γ subunit. It is a β -barrel structure.
 - 6) The α subunit is the main pore-forming subunit. It is a β -barrel structure. The β subunit is the γ subunit. It is a β -barrel structure.

- 7) ALL TESTS REQUIRE

- [illegible]

Foldout #1

MATRIX FOR THE PARALLEL PROCESSOR ARRAY

DEPENDENT				CLOCK	DATA	INPUTS	DATA OUTPUTS	TESTS FOR:
CAR	M	SIGN A	SIGN B	C ₁	D ₄ D ₃ D ₂ D ₁ D ₀	D ₄ D ₃ D ₂ D ₁ D ₀	D ₄ D ₃ D ₂ D ₁ D ₀	A - active state, I - inactive state, PWR - power test
0	0	0	0	0	0	0	0	Tests: S ₁ AND A ₁ S ₂ S ₃ S ₄ S ₅ S ₆ S ₇ S ₈ S ₉ S ₁₀ S ₁₁ S ₁₂ S ₁₃ S ₁₄ S ₁₅ S ₁₆ S ₁₇ S ₁₈ S ₁₉ S ₂₀ S ₂₁ S ₂₂ S ₂₃ S ₂₄ S ₂₅ S ₂₆ S ₂₇ S ₂₈ S ₂₉ S ₃₀ S ₃₁ S ₃₂ S ₃₃ S ₃₄ S ₃₅ S ₃₆ S ₃₇ S ₃₈ S ₃₉ S ₄₀ S ₄₁ S ₄₂ S ₄₃ S ₄₄ S ₄₅ S ₄₆ S ₄₇ S ₄₈ S ₄₉ S ₅₀ S ₅₁ S ₅₂ S ₅₃ S ₅₄ S ₅₅ S ₅₆ S ₅₇ S ₅₈ S ₅₉ S ₆₀ S ₆₁ S ₆₂ S ₆₃ S ₆₄ S ₆₅ S ₆₆ S ₆₇ S ₆₈ S ₆₉ S ₇₀ S ₇₁ S ₇₂ S ₇₃ S ₇₄ S ₇₅ S ₇₆ S ₇₇ S ₇₈ S ₇₉ S ₈₀ S ₈₁ S ₈₂ S ₈₃ S ₈₄ S ₈₅ S ₈₆ S ₈₇ S ₈₈ S ₈₉ S ₉₀ S ₉₁ S ₉₂ S ₉₃ S ₉₄ S ₉₅ S ₉₆ S ₉₇ S ₉₈ S ₉₉ S ₁₀₀ S ₁₀₁ S ₁₀₂ S ₁₀₃ S ₁₀₄ S ₁₀₅ S ₁₀₆ S ₁₀₇ S ₁₀₈ S ₁₀₉ S ₁₁₀ S ₁₁₁ S ₁₁₂ S ₁₁₃ S ₁₁₄ S ₁₁₅ S ₁₁₆ S ₁₁₇ S ₁₁₈ S ₁₁₉ S ₁₂₀ S ₁₂₁ S ₁₂₂ S ₁₂₃ S ₁₂₄ S ₁₂₅ S ₁₂₆ S ₁₂₇ S ₁₂₈ S ₁₂₉ S ₁₃₀ S ₁₃₁ S ₁₃₂ S ₁₃₃ S ₁₃₄ S ₁₃₅ S ₁₃₆ S ₁₃₇ S ₁₃₈ S ₁₃₉ S ₁₄₀ S ₁₄₁ S ₁₄₂ S ₁₄₃ S ₁₄₄ S ₁₄₅ S ₁₄₆ S ₁₄₇ S ₁₄₈ S ₁₄₉ S ₁₅₀ S ₁₅₁ S ₁₅₂ S ₁₅₃ S ₁₅₄ S ₁₅₅ S ₁₅₆ S ₁₅₇ S ₁₅₈ S ₁₅₉ S ₁₆₀ S ₁₆₁ S ₁₆₂ S ₁₆₃ S ₁₆₄ S ₁₆₅ S ₁₆₆ S ₁₆₇ S ₁₆₈ S ₁₆₉ S ₁₇₀ S ₁₇₁ S ₁₇₂ S ₁₇₃ S ₁₇₄ S ₁₇₅ S ₁₇₆ S ₁₇₇ S ₁₇₈ S ₁₇₉ S ₁₈₀ S ₁₈₁ S ₁₈₂ S ₁₈₃ S ₁₈₄ S ₁₈₅ S ₁₈₆ S ₁₈₇ S ₁₈₈ S ₁₈₉ S ₁₉₀ S ₁₉₁ S ₁₉₂ S ₁₉₃ S ₁₉₄ S ₁₉₅ S ₁₉₆ S ₁₉₇ S ₁₉₈ S ₁₉₉ S ₂₀₀ S ₂₀₁ S ₂₀₂ S ₂₀₃ S ₂₀₄ S ₂₀₅ S ₂₀₆ S ₂₀₇ S ₂₀₈ S ₂₀₉ S ₂₁₀ S ₂₁₁ S ₂₁₂ S ₂₁₃ S ₂₁₄ S ₂₁₅ S ₂₁₆ S ₂₁₇ S ₂₁₈ S ₂₁₉ S ₂₂₀ S ₂₂₁ S ₂₂₂ S ₂₂₃ S ₂₂₄ S ₂₂₅ S ₂₂₆ S ₂₂₇ S ₂₂₈ S ₂₂₉ S ₂₃₀ S ₂₃₁ S ₂₃₂ S ₂₃₃ S ₂₃₄ S ₂₃₅ S ₂₃₆ S ₂₃₇ S ₂₃₈ S ₂₃₉ S ₂₄₀ S ₂₄₁ S ₂₄₂ S ₂₄₃ S ₂₄₄ S ₂₄₅ S ₂₄₆ S ₂₄₇ S ₂₄₈ S ₂₄₉ S ₂₅₀ S ₂₅₁ S ₂₅₂ S ₂₅₃ S ₂₅₄ S ₂₅₅ S ₂₅₆ S ₂₅₇ S ₂₅₈ S ₂₅₉ S ₂₆₀ S ₂₆₁ S ₂₆₂ S ₂₆₃ S ₂₆₄ S ₂₆₅ S ₂₆₆ S ₂₆₇ S ₂₆₈ S ₂₆₉ S ₂₇₀ S ₂₇₁ S ₂₇₂ S ₂₇₃ S ₂₇₄ S ₂₇₅ S ₂₇₆ S ₂₇₇ S ₂₇₈ S ₂₇₉ S ₂₈₀ S ₂₈₁ S ₂₈₂ S ₂₈₃ S ₂₈₄ S ₂₈₅ S ₂₈₆ S ₂₈₇ S ₂₈₈ S ₂₈₉ S ₂₉₀ S ₂₉₁ S ₂₉₂ S ₂₉₃ S ₂₉₄ S ₂₉₅ S ₂₉₆ S ₂₉₇ S ₂₉₈ S ₂₉₉ S ₃₀₀ S ₃₀₁ S ₃₀₂ S ₃₀₃ S ₃₀₄ S ₃₀₅ S ₃₀₆ S ₃₀₇ S ₃₀₈ S ₃₀₉ S ₃₁₀ S ₃₁₁ S ₃₁₂ S ₃₁₃ S ₃₁₄ S ₃₁₅ S ₃₁₆ S ₃₁₇ S ₃₁₈ S ₃₁₉ S ₃₂₀ S ₃₂₁ S ₃₂₂ S ₃₂₃ S ₃₂₄ S ₃₂₅ S ₃₂₆ S ₃₂₇ S ₃₂₈ S ₃₂₉ S ₃₃₀ S ₃₃₁ S ₃₃₂ S ₃₃₃ S ₃₃₄ S ₃₃₅ S ₃₃₆ S ₃₃₇ S ₃₃₈ S ₃₃₉ S ₃₄₀ S ₃₄₁ S ₃₄₂ S ₃₄₃ S ₃₄₄ S ₃₄₅ S ₃₄₆ S ₃₄₇ S ₃₄₈ S ₃₄₉ S ₃₅₀ S ₃₅₁ S ₃₅₂ S ₃₅₃ S ₃₅₄ S ₃₅₅ S ₃₅₆ S ₃₅₇ S ₃₅₈ S ₃₅₉ S ₃₆₀ S ₃₆₁ S ₃₆₂ S ₃₆₃ S ₃₆₄ S ₃₆₅ S ₃₆₆ S ₃₆₇ S ₃₆₈ S ₃₆₉ S ₃₇₀ S ₃₇₁ S ₃₇₂ S ₃₇₃ S ₃₇₄ S ₃₇₅ S ₃₇₆ S ₃₇₇ S ₃₇₈ S ₃₇₉ S ₃₈₀ S ₃₈₁ S ₃₈₂ S ₃₈₃ S ₃₈₄ S ₃₈₅ S ₃₈₆ S ₃₈₇ S ₃₈₈ S ₃₈₉ S ₃₉₀ S ₃₉₁ S ₃₉₂ S ₃₉₃ S ₃₉₄ S ₃₉₅ S ₃₉₆ S ₃₉₇ S ₃₉₈ S ₃₉₉ S ₄₀₀ S ₄₀₁ S ₄₀₂ S ₄₀₃ S ₄₀₄ S ₄₀₅ S ₄₀₆ S ₄₀₇ S ₄₀₈ S ₄₀₉ S ₄₁₀ S ₄₁₁ S ₄₁₂ S ₄₁₃ S ₄₁₄ S ₄₁₅ S ₄₁₆ S ₄₁₇ S ₄₁₈ S ₄₁₉ S ₄₂₀ S ₄₂₁ S ₄₂₂ S ₄₂₃ S ₄₂₄ S ₄₂₅ S ₄₂₆ S ₄₂₇ S ₄₂₈ S ₄₂₉ S ₄₃₀ S ₄₃₁ S ₄₃₂ S ₄₃₃ S ₄₃₄ S ₄₃₅ S ₄₃₆ S ₄₃₇ S ₄₃₈ S ₄₃₉ S ₄₄₀ S ₄₄₁ S ₄₄₂ S ₄₄₃ S ₄₄₄ S ₄₄₅ S ₄₄₆ S ₄₄₇ S ₄₄₈ S ₄₄₉ S ₄₅₀ S ₄₅₁ S ₄₅₂ S ₄₅₃ S ₄₅₄ S ₄₅₅ S ₄₅₆ S ₄₅₇ S ₄₅₈ S ₄₅₉ S ₄₆₀ S ₄₆₁ S ₄₆₂ S ₄₆₃ S ₄₆₄ S ₄₆₅ S ₄₆₆ S ₄₆₇ S ₄₆₈ S ₄₆₉ S ₄₇₀ S ₄₇₁ S ₄₇₂ S ₄₇₃ S ₄₇₄ S ₄₇₅ S ₄₇₆ S ₄₇₇ S ₄₇₈ S ₄₇₉ S ₄₈₀ S ₄₈₁ S ₄₈₂ S ₄₈₃ S ₄₈₄ S ₄₈₅ S ₄₈₆ S ₄₈₇ S ₄₈₈ S ₄₈₉ S ₄₉₀ S ₄₉₁ S ₄₉₂ S ₄₉₃ S ₄₉₄ S ₄₉₅ S ₄₉₆ S ₄₉₇ S ₄₉₈ S ₄₉₉ S ₅₀₀ S ₅₀₁ S ₅₀₂ S ₅₀₃ S ₅₀₄ S ₅₀₅ S ₅₀₆ S ₅₀₇ S ₅₀₈ S ₅₀₉ S ₅₁₀ S ₅₁₁ S ₅₁₂ S ₅₁₃ S ₅₁₄ S ₅₁₅ S ₅₁₆ S ₅₁₇ S ₅₁₈ S ₅₁₉ S ₅₂₀ S ₅₂₁ S ₅₂₂ S ₅₂₃ S ₅₂₄ S ₅₂₅ S ₅₂₆ S ₅₂₇ S ₅₂₈ S ₅₂₉ S ₅₃₀ S ₅₃₁ S ₅₃₂ S ₅₃₃ S ₅₃₄ S ₅₃₅ S ₅₃₆ S ₅₃₇ S ₅₃₈ S ₅₃₉ S ₅₄₀ S ₅₄₁ S ₅₄₂ S ₅₄₃ S ₅₄₄ S ₅₄₅ S ₅₄₆ S ₅₄₇ S ₅₄₈ S ₅₄₉ S ₅₅₀ S ₅₅₁ S ₅₅₂ S ₅₅₃ S ₅₅₄ S ₅₅₅ S ₅₅₆ S ₅₅₇ S ₅₅₈ S ₅₅₉ S ₅₆₀ S ₅₆₁ S ₅₆₂ S ₅₆₃ S ₅₆₄ S ₅₆₅ S ₅₆₆ S ₅₆₇ S ₅₆₈ S ₅₆₉ S ₅₇₀ S ₅₇₁ S ₅₇₂ S ₅₇₃ S ₅₇₄ S ₅₇₅ S ₅₇₆ S ₅₇₇ S ₅₇₈ S ₅₇₉ S ₅₈₀ S ₅₈₁ S ₅₈₂ S ₅₈₃ S ₅₈₄ S ₅₈₅ S ₅₈₆ S ₅₈₇ S ₅₈₈ S ₅₈₉ S ₅₉₀ S ₅₉₁ S ₅₉₂ S ₅₉₃ S ₅₉₄ S ₅₉₅ S ₅₉₆ S ₅₉₇ S ₅₉₈ S ₅₉₉ S ₆₀₀ S ₆₀₁ S ₆₀₂ S ₆₀₃ S ₆₀₄ S ₆₀₅ S ₆₀₆ S ₆₀₇ S ₆₀₈ S ₆₀₉ S ₆₁₀ S ₆₁₁ S ₆₁₂ S ₆₁₃ S ₆₁₄ S ₆₁₅ S ₆₁₆ S ₆₁₇ S ₆₁₈ S ₆₁₉ S ₆₂₀ S ₆₂₁ S ₆₂₂ S ₆₂₃ S ₆₂₄ S ₆₂₅ S ₆₂₆ S ₆₂₇ S ₆₂₈ S ₆₂₉ S ₆₃₀ S ₆₃₁ S ₆₃₂ S ₆₃₃ S ₆₃₄ S ₆₃₅ S ₆₃₆ S ₆₃₇ S ₆₃₈ S ₆₃₉ S ₆₄₀ S ₆₄₁ S ₆₄₂ S ₆₄₃ S ₆₄₄ S ₆₄₅ S ₆₄₆ S ₆₄₇ S ₆₄₈ S ₆₄₉ S ₆₅₀ S ₆₅₁ S ₆₅₂ S ₆₅₃ S ₆₅₄ S ₆₅₅ S ₆₅₆ S ₆₅₇ S ₆₅₈ S ₆₅₉ S ₆₆₀ S ₆₆₁ S ₆₆₂ S ₆₆₃ S ₆₆₄ S ₆₆₅ S ₆₆₆ S ₆₆₇ S ₆₆₈ S ₆₆₉ S ₆₇₀ S ₆₇₁ S ₆₇₂ S ₆₇₃ S ₆₇₄ S ₆₇₅ S ₆₇₆ S ₆₇₇ S ₆₇₈ S ₆₇₉ S ₆₈₀ S ₆₈₁ S ₆₈₂ S ₆₈₃ S ₆₈₄ S ₆₈₅ S ₆₈₆ S ₆₈₇ S ₆₈₈ S ₆₈₉ S ₆₉₀ S ₆₉₁ S ₆₉₂ S ₆₉₃ S ₆₉₄ S ₆₉₅ S ₆₉₆ S ₆₉₇ S ₆₉₈ S ₆₉₉ S ₇₀₀ S ₇₀₁ S ₇₀₂ S ₇₀₃ S ₇₀₄ S ₇₀₅ S ₇₀₆ S ₇₀₇ S ₇₀₈ S ₇₀₉ S ₇₁₀ S ₇₁₁ S ₇₁₂ S ₇₁₃ S ₇₁₄ S ₇₁₅ S ₇₁₆ S ₇₁₇ S ₇₁₈ S ₇₁₉ S ₇₂₀ S ₇₂₁ S ₇₂₂ S ₇₂₃ S ₇₂₄ S ₇₂₅ S ₇₂₆ S ₇₂₇ S ₇₂₈ S ₇₂₉ S ₇₃₀ S ₇₃₁ S ₇₃₂ S ₇₃₃ S ₇₃₄ S ₇₃₅ S ₇₃₆ S ₇₃₇ S ₇₃₈ S ₇₃₉ S ₇₄₀ S ₇₄₁ S ₇₄₂ S ₇₄₃ S ₇₄₄ S ₇₄₅ S ₇₄₆ S ₇₄₇ S ₇₄₈ S ₇₄₉ S ₇₅₀ S ₇₅₁ S ₇₅₂ S ₇₅₃ S ₇₅₄ S ₇₅₅ S ₇₅₆ S ₇₅₇ S ₇₅₈ S ₇₅₉ S ₇₆₀ S ₇₆₁ S ₇₆₂ S ₇₆₃ S ₇₆₄ S ₇₆₅ S ₇₆₆ S ₇₆₇ S ₇₆₈ S ₇₆₉ S ₇₇₀ S ₇₇₁ S ₇₇₂ S ₇₇₃ S ₇₇₄ S ₇₇₅ S ₇₇₆ S ₇₇₇ S ₇₇₈ S ₇₇₉ S ₇₈₀ S ₇₈₁ S ₇₈₂ S ₇₈₃ S ₇₈₄ S ₇₈₅ S ₇₈₆ S ₇₈₇ S ₇₈₈ S ₇₈₉ S ₇₉₀ S ₇₉₁ S ₇₉₂ S ₇₉₃ S ₇₉₄ S ₇₉₅ S ₇₉₆ S ₇₉₇ S ₇₉₈ S ₇₉₉ S ₈₀₀ S ₈₀₁ S ₈₀₂ S ₈₀₃ S ₈₀₄ S ₈₀₅ S ₈₀₆ S ₈₀₇ S ₈₀₈ S ₈₀₉ S ₈₁₀ S ₈₁₁ S ₈₁₂ S ₈₁₃ S ₈₁₄ S ₈₁₅ S ₈₁₆ S ₈₁₇ S ₈₁₈ S ₈₁₉ S ₈₂₀ S ₈₂₁ S ₈₂₂ S ₈₂₃ S ₈₂₄ S ₈₂₅ S ₈₂₆ S ₈₂₇ S ₈₂₈ S ₈₂₉ S ₈₃₀ S ₈₃₁ S ₈₃₂ S ₈₃₃ S ₈₃₄ S ₈₃₅ S ₈₃₆ S ₈₃₇ S ₈₃₈ S ₈₃₉ S ₈₄₀ S ₈₄₁ S ₈₄₂ S ₈₄₃ S ₈₄₄ S ₈₄₅ S ₈₄₆ S ₈₄₇ S ₈₄₈ S ₈₄₉ S ₈₅₀ S ₈₅₁ S ₈₅₂ S ₈₅₃ S ₈₅₄ S ₈₅₅ S ₈₅₆ S ₈₅₇ S ₈₅₈ S ₈₅₉ S ₈₆₀ S ₈₆₁ S ₈₆₂ S ₈₆₃ S ₈₆₄ S ₈₆₅ S ₈₆₆ S ₈₆₇ S ₈₆₈ S ₈₆₉ S ₈₇₀ S ₈₇₁ S ₈₇₂ S ₈₇₃ S ₈₇₄ S ₈₇₅ S ₈₇₆ S ₈₇₇ S ₈₇₈ S ₈₇₉ S ₈₈₀ S ₈₈₁ S ₈₈₂ S ₈₈₃ S ₈₈₄ S ₈₈₅ S ₈₈₆ S ₈₈₇ S ₈₈₈ S ₈₈₉ S ₈₉₀ S ₈₉₁ S ₈₉₂ S ₈₉₃ S ₈₉₄ S ₈₉₅ S ₈₉₆ S ₈₉₇ S ₈₉₈ S ₈₉₉ S ₉₀₀ S ₉₀₁ S ₉₀₂ S ₉₀₃ S ₉₀₄ S ₉₀₅ S ₉₀₆ S ₉₀₇ S ₉₀₈ S ₉₀₉ S ₉₁₀ S ₉₁₁ S ₉₁₂ S ₉₁₃ S ₉₁₄ S ₉₁₅ S ₉₁₆ S ₉₁₇ S ₉₁₈ S ₉₁₉ S ₉₂₀ S ₉₂₁ S ₉₂₂ S ₉₂₃ S ₉₂₄ S ₉₂₅ S ₉₂₆ S ₉₂₇ S ₉₂₈ S ₉₂₉ S ₉₃₀ S ₉₃₁ S ₉₃₂ S ₉₃₃ S ₉₃₄ S ₉₃₅ S ₉₃₆ S ₉₃₇ S ₉₃₈ S ₉₃₉ S ₉₄₀ S ₉₄₁ S ₉₄₂ S ₉₄₃ S ₉₄₄ S ₉₄₅ S ₉₄₆ S ₉₄₇ S ₉₄₈ S ₉₄₉ S ₉₅₀ S ₉₅₁ S ₉₅₂ S ₉₅₃ S ₉₅₄ S ₉₅₅ S ₉₅₆ S ₉₅₇ S ₉₅₈ S ₉₅₉ S ₉₆₀ S ₉₆₁ S ₉₆₂ S ₉₆₃ S ₉₆₄ S ₉₆₅ S ₉₆₆ S ₉₆₇ S ₉₆₈ S ₉₆₉ S ₉₇₀ S ₉₇₁ S ₉₇₂ S ₉₇₃ S ₉₇₄ S ₉₇₅ S ₉₇₆ S ₉₇₇ S ₉₇₈ S ₉₇₉ S ₉₈₀ S ₉₈₁ S ₉₈₂ S ₉₈₃ S ₉₈₄ S ₉₈₅ S ₉₈₆ S ₉₈₇ S ₉₈₈ S ₉₈₉ S ₉₉₀ S ₉₉₁ S ₉₉₂ S ₉₉₃ S ₉₉₄ S ₉₉₅ S ₉₉₆ S ₉₉₇ S ₉₉₈ S ₉₉₉ S ₁₀₀₀ S ₁₀₀₁ S ₁₀₀₂ S ₁₀₀₃ S ₁₀₀₄ S ₁₀₀₅ S ₁₀₀₆ S ₁₀₀₇ S ₁₀₀₈ S ₁₀₀₉ S ₁₀₁₀ S ₁₀₁₁ S ₁₀₁₂ S ₁₀₁₃ S ₁₀₁₄ S ₁₀₁₅ S ₁₀₁₆ S ₁₀₁₇ S ₁₀₁₈ S ₁₀₁₉ S ₁₀₂₀ S ₁₀₂₁ S ₁₀₂₂ S ₁₀₂₃ S ₁₀₂₄ S ₁₀₂₅ S ₁₀₂₆ S ₁₀₂₇ S ₁₀₂₈ S ₁₀₂₉ S ₁₀₃₀ S ₁₀₃₁ S ₁₀₃₂ S ₁₀₃₃ S ₁₀₃₄ S ₁₀₃₅ S ₁₀₃₆ S ₁₀₃₇ S ₁₀₃₈ S ₁₀₃₉ S ₁₀₄₀ S ₁₀₄₁ S ₁₀₄₂ S ₁₀₄₃ S ₁₀₄₄ S ₁₀₄₅ S ₁₀₄₆ S ₁₀₄₇ S ₁₀₄₈ S ₁₀₄₉ S ₁₀₅₀ S ₁₀₅₁ S ₁₀₅₂ S ₁₀₅₃ S ₁₀₅₄ S ₁₀₅₅ S ₁₀₅₆ S ₁₀₅₇ S ₁₀₅₈ S ₁₀₅₉ S ₁₀₆₀ S ₁₀₆₁ S ₁₀₆₂ S ₁₀₆₃ S ₁₀₆₄ S ₁₀₆₅ S ₁₀₆₆ S ₁₀₆₇ S ₁₀₆₈ S ₁₀₆₉ S ₁₀₇₀ S ₁₀₇₁ S ₁₀₇₂ S ₁₀₇₃ S ₁₀₇₄ S ₁₀₇₅ S ₁₀₇₆ S ₁₀₇₇ S ₁₀₇₈ S ₁₀₇₉ S ₁₀₈₀ S ₁₀₈₁ S ₁₀₈₂ S ₁₀₈₃ S ₁₀₈₄ S ₁₀₈₅ S ₁₀₈₆ S ₁₀₈₇ S ₁₀₈₈ S ₁₀₈₉ S ₁₀₉₀ S ₁₀₉₁ S ₁₀₉₂ S ₁₀₉₃ S ₁₀₉₄ S ₁₀₉₅ S ₁₀₉₆ S ₁₀₉₇ S ₁₀₉₈ S ₁₀₉₉ S ₁₁₀₀ S ₁₁₀₁ S ₁₁₀₂ S ₁₁₀₃ S ₁₁₀₄ S ₁₁₀₅ S ₁₁₀₆ S ₁₁₀₇ S ₁₁₀₈ S ₁₁₀₉ S ₁₁₁₀ S ₁₁₁₁ S ₁₁₁₂ S ₁₁₁₃ S ₁₁₁₄ S ₁₁₁₅ S ₁₁₁₆ S ₁₁₁₇ S ₁₁₁₈ S ₁₁₁₉ S ₁₁₂₀ S ₁₁₂₁ S ₁₁₂₂ S ₁₁₂₃ S ₁₁₂₄ S ₁₁₂₅ S ₁₁₂₆ S ₁₁₂₇ S ₁₁₂₈ S ₁₁₂₉ S ₁₁₃₀ S ₁₁₃₁ S ₁₁₃₂ S ₁₁₃₃ S ₁₁₃₄ S ₁₁₃₅ S ₁₁₃₆ S ₁₁₃₇ S ₁₁₃₈ S ₁₁₃₉ S ₁₁₄₀ S ₁₁₄₁ S ₁₁₄₂ S ₁₁₄₃ S ₁₁₄₄ S ₁₁₄₅ S ₁₁₄₆ S ₁₁₄₇ S ₁₁₄₈ S ₁₁₄₉ S ₁₁₅₀ S ₁₁₅₁ S ₁₁₅₂ S ₁₁₅₃ S ₁₁₅₄ S ₁₁₅₅ S ₁₁₅₆ S ₁₁₅₇ S ₁₁₅₈ S ₁₁₅₉ S ₁₁₆₀ S ₁₁₆₁ S ₁₁₆₂ S ₁₁₆₃ S ₁₁₆₄ S ₁₁₆₅ S ₁₁₆₆ S ₁₁₆₇ S ₁₁₆₈ S ₁₁₆₉ S ₁₁₇₀ S ₁₁₇₁ S ₁₁₇₂ S ₁₁₇₃ S ₁₁₇₄ S ₁₁₇₅ S ₁₁₇₆ S ₁₁₇₇ S ₁₁₇₈ S ₁₁₇₉ S ₁₁₈₀ S ₁₁₈₁ S ₁₁₈₂ S ₁₁₈₃ S ₁₁₈₄ S ₁₁₈₅ S ₁₁₈₆ S ₁₁₈₇ S ₁₁₈₈ S ₁

PRECEDING PAGE BLANK NOT FILMED.

control instruction (R, AND, L, etc.), the data inputs are defined so as many test conditions as possible can be monitored. After the initial 16 words, any untested conditions can be tested.

Upon successful completion of the testing of the control logic, the processing portion of the array is checked. The arithmetic and shifting portion are checked independently. In testing the arithmetic portion, the first EXCLUSIVE-OR structure of the full adders, the carry generation structures, and the input gating structures are tested independently. Tests for AND, OR, and SUM transmission gates, and the second EXCLUSIVE-OR structure are mixed into the above tests.

Testing of the shift register portion is accomplished in a straightforward shifting test. This test checks out the L, R, and R_0 transmission gates and checks the D flip flop operation. Tests for special indicators are checked during the control test portion of the test sequence.

In the test matrix, a column contains information as to what conditions are being tested for in each test word. Examination of this column will reveal that all conditions required by Section II are satisfied.

PRECEDING PAGE BLANK NOT FILMED.

SECTION VIII
CONCLUSIONS AND RECOMMENDATIONS

A four-bit parallel processor has been designed. The logic design conforms to the contract objectives. The array of 750 devices has been laid out on a chip 145 mils by 155 mils. A composite drawing was completed and rubyliths were prepared. Extensive checking of the rubylith (with the assistance of R. Lesniewski of NASA) was used to reduce the probability of error in the layout. A test matrix was developed to permit testing of the processing and control portions of the parallel processor.

It is recommended that the program continue as planned.

PRECEDING PAGE BLANK NOT FILMED.

SECTION IX
PROGRAM FOR NEXT INTERVAL

During the next interval, the four-bit parallel processor unit will be fabricated and tested.