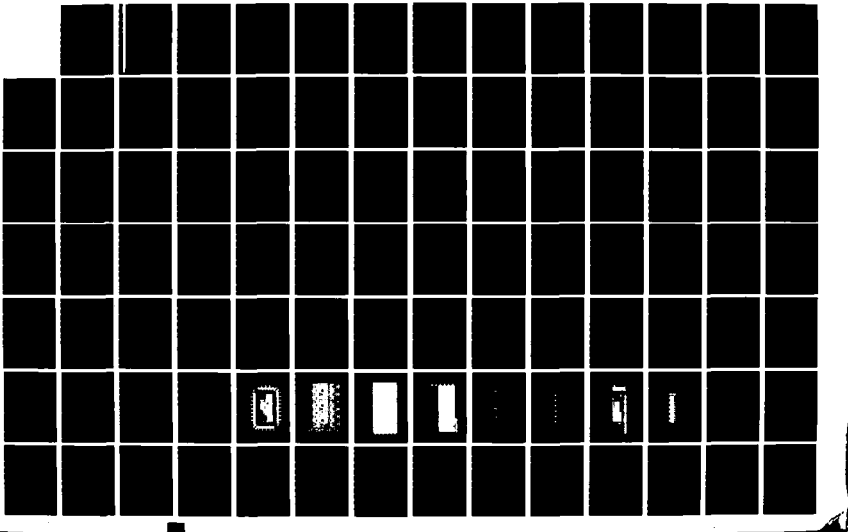
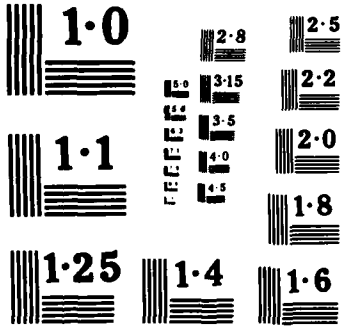
AD-A158 369 BULK CMOS VLSI TECHNOLOGY STUDIES PART 4 DESIGN OF A CMOS MICROSEQUENCER..(U) MISSISSIPPI STATE UNIV MISSISSIPPI STATE DEPT OF ELECTRICAL E.. 1/3

UNCLASSIFIED J D TROTTER ET AL. 17 JUN 85 F/G 9/5 NL





NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

FINAL REPORT

CONTRACT DAAG29-82-K-0167

BULK CMOS VLSI TECHNOLOGY STUDIES

PART 4: DESIGN OF A CMOS MICROSEQUENCER

Principal Investigator

J. Donald Trotter

Associate Investigator

Ajay Kumar Reddy Naini

Mississippi State University
Department of Electrical Engineering
Mississippi State, Mississippi 39762

for
Defense Advance Research Projects Agency
1400 Wilson Ave.
Arlington, VA 22209

for

U. S. Army Research Office
P. O. Box 11211
Research Triangle Park, NC 27709

June 17, 1985

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT This document has been approved for public release and sale; its distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Miss. State University 6b. OFFICE SYMBOL (If applicable) Electrical	7a. NAME OF MONITORING ORGANIZATION Georgia Institute of Technology
6c. ADDRESS (City, State and ZIP Code) Drawer EE Miss. State, MS 39762	7b. ADDRESS (City, State and ZIP Code) 206 O'Keefe Building Atlanta, GA 30332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U.S. Army Research Office 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAG29-82-K-0167
8c. ADDRESS (City, State and ZIP Code) P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) Bulk CMOS VLSI Technology Studies	
12. PERSONAL AUTHOR(S) J. Donald Trotter, Gowni Shiva Prasad	
13a. TYPE OF REPORT Final 13b. TIME COVERED FROM 8-82 TO 2-85	14. DATE OF REPORT (Yr., Mo., Day) 15. PAGE COUNT
16. SUPPLEMENTARY NOTATION	
17. COSATI CODES FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) (Subtitle) Design of a CMOS Microsequencer

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

This thesis discusses the design of a CMOS microsequencer used in a microprogrammed control organization. A preconcieved data path performs most of the data manipulation functions for an LSI computer system. The operations are performed as directed by sequences of control microinstructions, which are fetched from a microcode memory using addresses generated by the microsequencer chip.

A Programmable Logic Array (PLA) is selected in lieu of random logic to control the circuits within the microsequencer. Extensive use has been made of clocked CMOS over classic CMOS to achieve higher layout density and better performance. The chip has been designed using scalable design rules which means it can be fabricated in 1.2 micron or 3 micron technology. The design uses double-layer metal, eliminating the need for extensive poly-interconnect lines.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code) 22c. OFFICE SYMBOL

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

FINAL REPORT

CONTRACT DAAG29-82-K-0167

BULK CMOS VLSI TECHNOLOGY STUDIES

PART 4: DESIGN OF A CMOS MICROSEQUENCER

Principal Investigator

J. Donald Trotter

Associate Investigator

Ajay Kumar Reddy Naini

Mississippi State University
Department of Electrical Engineering
Mississippi State, Mississippi 39762

for
Defense Advance Research Projects Agency
1400 Wilson Ave.
Arlington, VA 22209

for

U. S. Army Research Office
P. O. Box 11211
Research Triangle Park, NC 27709

June 17, 1985

PREFACE

The function of the control unit in a digital system is to initiate sequences of micro-operations which result in ordered operations performed on data stored in the registers of the system. When the control functions are generated by hardware using conventional logic design techniques, the control unit is said to be hard-wired. A second alternative is microprogrammed control which pays a penalty in speed but is an elegant and systematic method for generating the micro-operation sequences in a digital system. This thesis discusses, in Complementary Metal Oxide Semiconductor (CMOS) technology, the design of a microsequencer as part of a microprogrammed control organization.

The material in this report is divided into five chapters and 3 appendices. Chapter 1 provides an overview of the microsequencer and explains some basic concepts of microprogramming. Appendix A provides a detailed functional description of the microsequencer including the functional diagram of the microsequencer and instruction set for the Programmable Logic Array (PLA), which controls the microsequencer logic. Chapter 1 and Appendix A can be read together for a better understanding of the system. Chapter 2 deals with macros, also called subsystems, of the microsequencer. It includes the pin definitions and execution examples of the macros used to build the microsequencer. Appendix B provides a detailed description of macros consisting of functional and logic diagrams.

circuit schematics and floor plans of the macros. Appendix B can be read in support of chapter 2.

Chapter 3, on circuits and layouts, provides the basics of CMOS integrated devices, circuits and fabrication. Appendix C, the cell library, provides the truth-table, logic equation (s), terminal information, logic diagram, circuit schematic and layout of every cell used in building the macros. Appendix C supports chapter 3 and Appendix B. Chapter 4 discusses the functional testing of the microsequencer chip; chapter 5 deals with the next generation microcontroller including suggestions to modify the present microsequencer so that it fits into a redesigned microcontroller.

I would like to acknowledge the extensive help and advice given by Dr. J. D. Trotter throughout this project as supervising professor. Thanks also to the other members of my committee which include Dr. W. A. Hornfeck and Dr. R. C. McCann. Appreciation is also given to Chris Jones for his constructive criticism and helpful discussions.

ABSTRACT

Gowni Shiva Prasad, Master of Science, May 1985

Major: Electrical Engineering, Department of Electrical Engineering

Title of Thesis: Design of a CMOS Microsequencer

Directed by: Dr. J. Donald Trotter

Pages in thesis: 215. Words in Abstract: 128.

ABSTRACT

This thesis discusses the design of a CMOS microsequencer used in a microprogrammed control organization. A preconcieved data path performs most of the data manipulation functions for an LSI computer system. The operations are performed as directed by sequences of control microinstructions, which are fetched from a microcode memory using addresses generated by the microsequencer chip.

A Programmable Logic Array (PLA) is selected in lieu of random logic to control the circuits within the microsequencer. Extensive use has been made of clocked CMOS over classic CMOS to achieve higher layout density and better performance. The chip has been designed using scalable design rules which means it can be fabricated in 1.2 micron or 3 micron technology. The design uses double-layer metal, eliminating the need for extensive poly-interconnect lines.

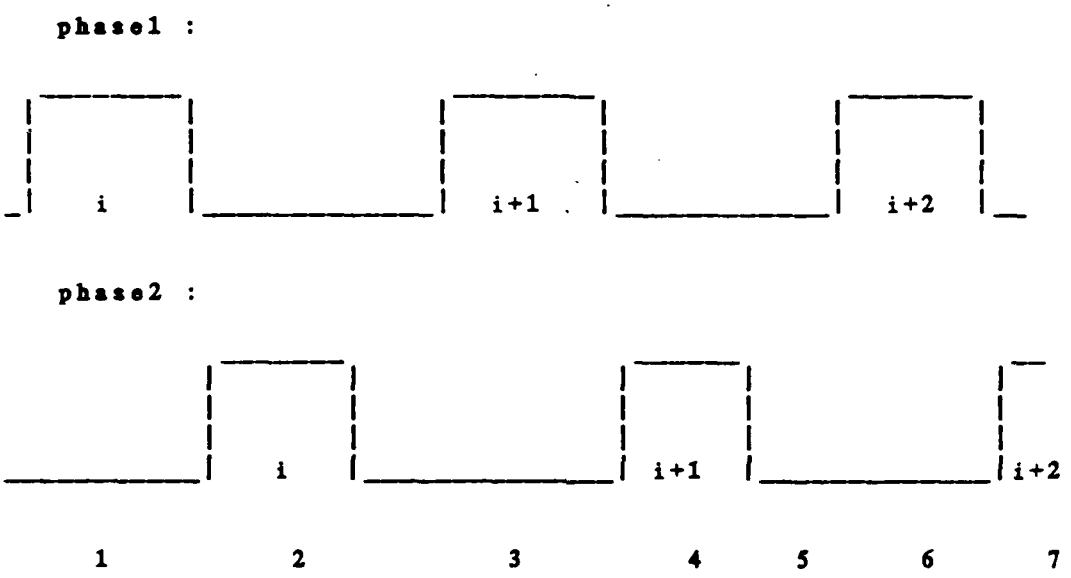
TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1.3. SYSTEM TIMING

This section explains the timing associated with the microsequencer, the data path, and the microstore.

Figure 1.4. TIMING DIAGRAMS



1 : Microsequencer produces microaddress i.

2 : Microstore is accessed for microinstruction i.

3 : Microsequencer produces microaddress i+1; data path phase1 transfer microinstruction i.

4 : Microstore accessed for microinstruction i+1; data path phase2 execute microinstruction i.

5 : Status transfer microinstruction i (status transfer from the data path to the microsequencer).

6 : Microsequencer produces microaddress i+2; data path phase1 transfer microinstruction i+1.

7 : Microstore accessed for microinstruction i+2.

1.2. MICROINSTRUCTION'S NEXT ADDRESS INFORMATION

The following list shows the next address information of the microinstruction

CD : 3 bit condition field to select one of the status bits in a multiplexer (MUX2) of the data path to generate the test bit T. If the selected bit is equal to 1, the T (test) bit is equal to 1; otherwise, it is equal to 0.

BR : 4 bit branch instruction field to execute one of the 16 sequence control instructions, most of which are conditional depending upon the test bit (T), state of the decrementer, or both.

ADF: an explicit address field so that it can be an address source for MUX1 or a count (N) for looping operations.

T is the test flag from the data path in phase1'* phase2' following phase2. The 5 bit op-code is the macro-operation part of the machine instruction.

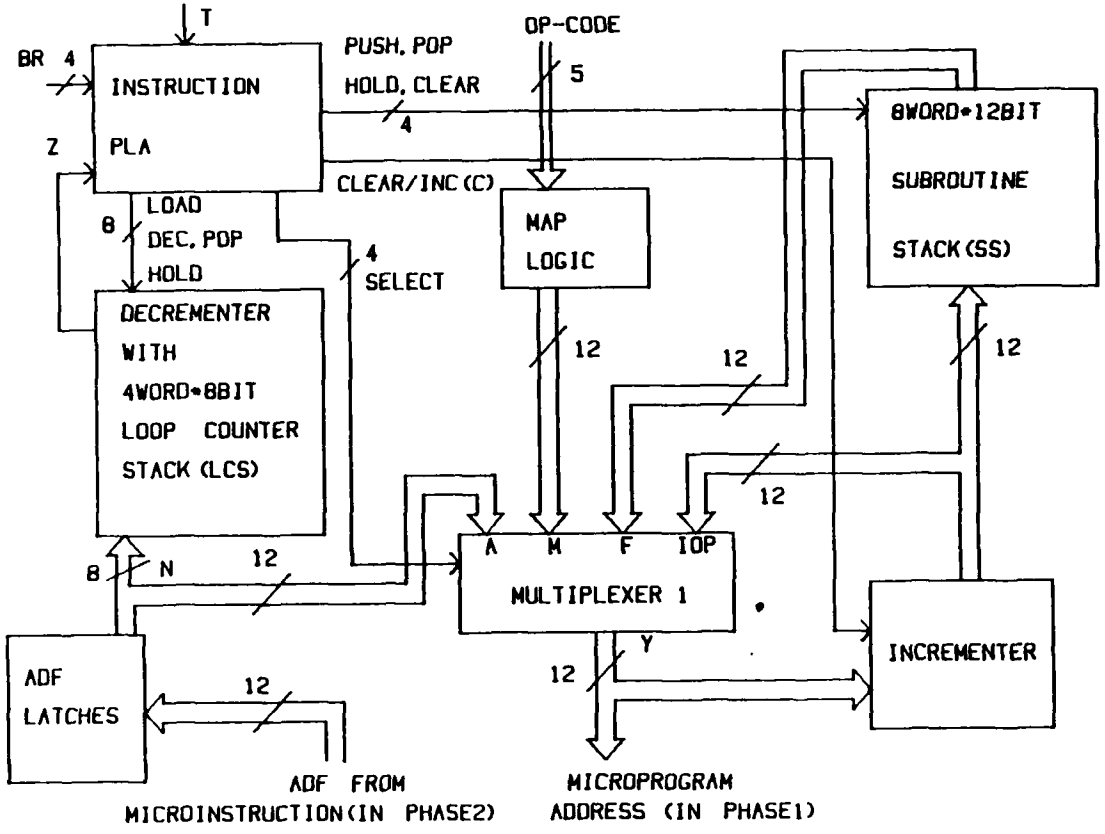


Figure 1.3 BLOCK DIAGRAM OF MICROSEQUENCER

A more flexible mapping scheme is one that uses another ROM or PLA to specify the mapping bits. In this configuration, the bits in the macro-operation specify the address for the mapping memory. The contents of the mapping memory provide the bits for the control memory address. This enables the placement of the routine, that executes the macro-operation, in any desired location in the microcode memory.

Another microcode address operation that could be considered is a form of loop operation. This operation is useful when sections of microcode are to be executed N times, where N can either be a constant specified in the microcode or be the result of a calculation done in the data path. However in the proposed controller the effect of the data path on the controller is limited to the influence of the flag information. The same ADF field of the previous microinstruction is used as an address source for MUX1 or as a count (N) for looping. Only the 8 least significant bits of the ADF field are used for N. One way to implement this instruction is to dedicate one register to be the loop counter and to do conditional branches in the controller based on the result of decrementing the value in that register. With one loop counter, loops could not be nested, and loops could not be used inside of subroutines. With a LIFO stack (also called as a push-down stack), nested loops and loops within subroutines could both be accomodated. Figure 1.3 shows the block diagram of the microsequencer.

The disadvantage with this scheme is the restriction of 2^7 microinstructions per macro-operation. Since the number of microinstructions for a macro-operation is not known at hand, a different mapping scheme is used here, as shown in Figure 1.2.ii. This mapping scheme clears the seven most significant bits and transfers the 5-bit op-code into the 5 least significant address bits. The mapped output (M) becomes another address source for MUX1. This mapping process provides more flexibility to the microprogrammer in that the microinstructions for a macro-operation need not be placed in sequence.

Figure 1.2. MAPPING SCHEMES

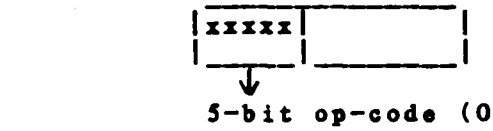


Figure 1.2.i

Mapped output (M) : xxxxx 0000000

op-code mapping bits

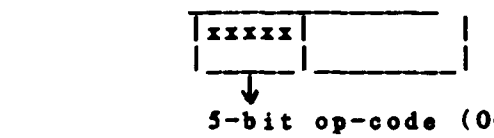


Figure 1.2.ii

Mapped output (M) : 0000000 xxxxx

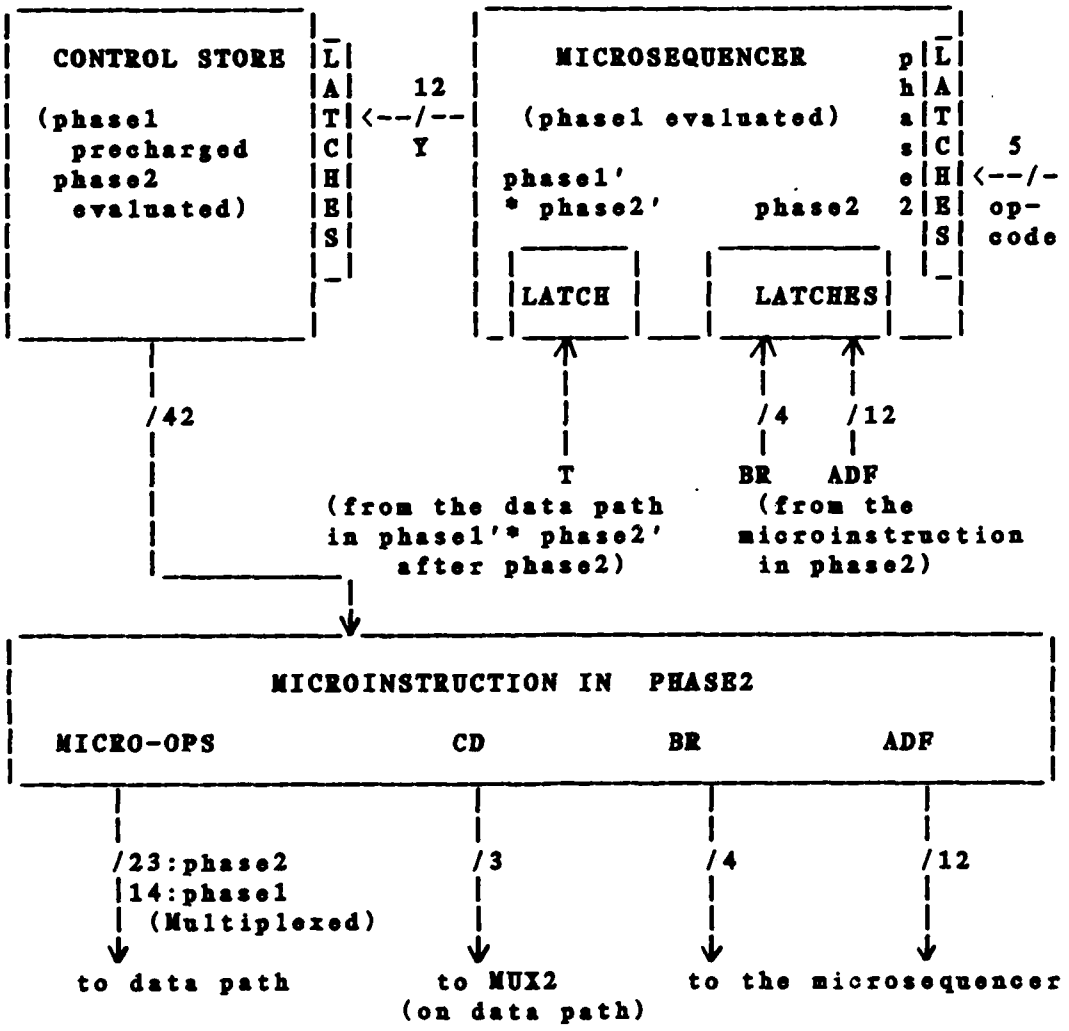
mapping bits op-code

is the jump or branch; so there should be some means of loading possible address value into MUX1. Frequently, many microprograms contain identical sections of code. Microinstructions can be saved by employing subroutines that use common sections of microcode; e.g., the sequence of micro-operations needed to generate the effective address of the operand for a machine instruction is common to all memory reference instructions. This sequence could be a subroutine which is called from within many other routines to execute the effective address computation. To provide such microcode subroutine facilities, provisions must be made for saving return addresses, which is most easily done with a last-in first-out (LIFO) memory stack.

Referring to Figure 1.1, it is seen that the microinstruction contains a field labeled ADF which may be an explicit address by itself. This is one of the address sources for MUX1. Another special type of branch exists when a microinstruction specifies a branch to the first word of a routine for a macro-operation in control memory. The address bits for this type of branch are a function of bits used in the operation (op-code) part of the machine instruction.

One specific mapping process [6] that converts the 5-bit op-code to a 12-bit microprogram address is shown in Figure 1.2.i. In this mapping scheme, the 7 least significant address bits are cleared to zeros while the 5-bit op-code is transferred.

Figure 1.1. MICROPROGRAMMED CONTROL ORGANIZATION



In a microprogrammed controller, the design of the control logic is reduced to encoding sequences of control bit patterns to be stored, along with control memory address sequencing information. The microcode memory may be implemented as either writable or Read-Only Memory (ROM). Each control word of the microcode memory is called a microinstruction and a sequence of words, which together control execution of a macroinstruction, is called a microprogram. The next address information of the microinstruction and the test flag from the data path both act as input data to the microsequencer and the outputs of the microsequencer are the microcode memory address lines. The advantage of a writable control memory is the flexibility of choosing the instruction set of a computer by changing the microprogram under processor control. Figure 1.1 shows one form of a microprogrammed control organization.

The purpose of a microsequencer is to present an address to the control memory so that a microinstruction may be read and executed. The next address logic of the microsequencer determines the specific address source which is to be used in fetching the next microinstruction. There must be a multiplexer (MUX1) which gates one of the various address sources to the control memory. The most common address calculation is to increment the current address by one; therefore the microsequencer should contain an incrementer. The second most important address calculation

CHAPTER 1

MICROSEQUENCER OVERVIEW

1.1. OVERALL CHIP DESCRIPTION

The primary data processsing module of a Large Scale Integrated (LSI) system is the data path chip [3]. The data path is capable of performing a variety of operations on a stream of data supplied from its internal registers or from its I/O ports. However, the data path in itself is not a complete system. An additional component is required to supply the control bits that determine the function of the path during each machine cycle. The overall operations performed on data within the data path are determined by the system controller.

A close examination of the controllers of typical computers reveals the fact that every one either is, or contains within it, a finite state machine. This finite state machine can be implemented by a Programmable Logic Array (PLA) or microprogram controller. It is often desirable for the state machine to be implemented in some writable medium, rather than in the fixed code of a standard PLA and thus patterned permanently in the silicon. Utilizing an engineering prototype model for a computer is such an example application. It has been decided to implement the controller by utilizing a microsequencer and a writable control memory for this research effort.

MUXSEL	Multiplexer Select
N	External Count
NMOS	N-Channel MOS
OC	Operation Code
P	Pass Signal
PCU	Program Control Unit
PLA	Programmable Logic Array
PMOS	P-Channel MOS
RAM	Random Access Memory
ROM	Read-Only Memory
SA	Select A
SF	Select F
SI	Select I
SM	Select M
SPICE	Simulation Program, Integrated Circuit Emphasis
SS	Subroutine Stack
T	Test Bit from Data Path
VDD	Power Supply Voltage
VSB	Source-Body Voltage of an NMOS Transistor
VTP	Threshold Voltage of a PMOS transistor
VTN	Threshold Voltage of an NMOS transistor
W/L	Width/Length Ratio of an MOS Transistor
Y	Microprogram Address
Z	Output of the ZD in DEC
ZD	Zero Detector
Zp	Output of the ZD in PCU

F	Subroutine Stack Top
GND	Ground (Zero Volts)
GRA	Generate Random Address
I	INC LOGIC output
INC	Incrementer
INC CAR CHAIN	Incrementer Carry-Chain
INC LOGIC	Incrementer Logic
I/O	Input/Output
IOP	Incremented output
IRAM	Internal RAM
K	Kill Signal
L	LCR output
LC	Load Count
LCR	Loop Counter Register
LCS	Loop Counter Stack
LCS2	Second word of LCS
LD	Load Decremented output
LIFO	Last-in First-out
LL	Load LCS2
LSB	Least Significant Bit
LSI	Large Scale Integrated
M	Mapped output
MD	Output of the DEC MUX
MOS	Metal Oxide Semiconductor
MSB	Most Significant Bit
MUX1	Multiplexer 1
MUX2	Multiplexer 2

LIST OF ABBREVIATIONS

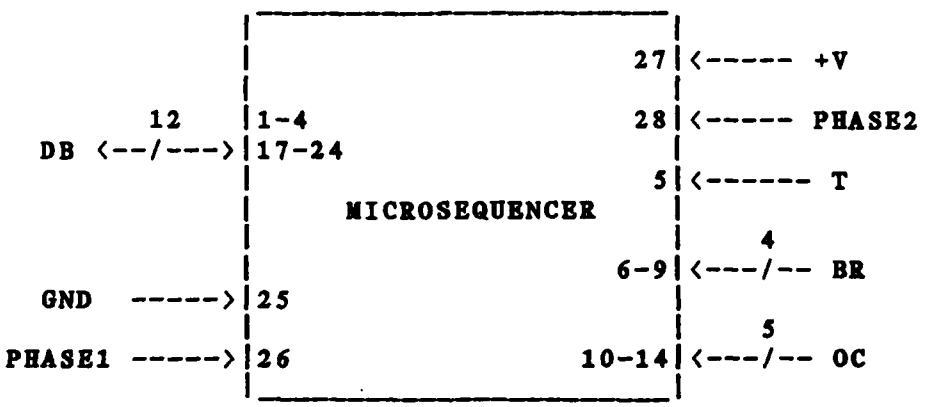
Abbreviation	Name
A	ADF Latches output
ALU	Arithmetic and Logic Unit
B	Branch Field bit
BC	Byte Count
BN	Beta of an N-Channel transistor
BP	Beta of a P-Channel Transistor
BR	Beta ratio of an MOS inverter
C	Clear/Count
CD	Condition Field
CIN	Carry-in
CMOS	Complementary Metal Oxide Semicoductor
COUT	Carry-out
D	DEC LOGIC output
DB	Bi-Directional Data Bus
DEC	Decrementer
DEC CAR CHAIN	Decrementer Carry-Chain
DEC I/P LATCH	Decrementer Input Latch
DEC LOGIC	Decrementer Logic
DEC O/P LATCH	Decrementer Output Latch
DIN	Data-in
DOP	Decremented output
DOUT	Data-out
DPFB	Data Path Flag Bit
EX-OR	Exclusive-OR

LIST OF TABLES

1.4. PIN DEFINITIONS OF THE MICROSEQUENCER CHIP

This section presents the pin diagram of the microsequencer chip followed by a brief description of the pins. For details of these pin definitions refer to chapter 2 on macros.

Figure 1.5. PIN DIAGRAM OF THE MICROSEQUENCER CHIP



DB (pins 1-4 and 17-24) is the 12-bit bidirectional data bus which is used for Y in phase1 and ADF in phase2. For DB (MSB,...,LSB) are (4,...1,24,...,17). Y is the 12-bit microprogram address (microsequencer output) in phase1 and ADF is the 12-bit explicit address field of the microinstruction which is the input to the microsequencer in phase2. The 8 least siginificant bits of the ADF field can be a count for looping, where (MSB,...,LSB) are (24,...,17). T (pin 5) is the test flag from the data path to the microsequencer in phase1'*phase2' following phase2; BR (pins 6-9) is the 4-bit branch instruction field of the

microinstruction which is the input to the microsequencer in phase2. Pin 6 is the LSB and pin 9 is the MSB. OC (pins 10-14) is the 5-bit op-code from the machine instruction in phase2. Pin 10 is the MSB and pin 14 is the LSB.

Note that pins 1-4 and 17-24 are shared by Y and ADF to reduce the pin count of the chip. They are the ADF inputs to the microsequencer in phase2 and Y outputs in phase1. Pins 15,16,29 to 32, and 33 to 36 are not connected. +V is the voltage supply and GND is the ground.

1.5. INSTRUCTION PLA

Irregular combinational functions can be conveniently mapped onto regular structures by means of a Programmable Logic Array (PLA). A PLA is selected in lieu of random logic to control the circuits within the microsequencer because the PLA is a regular structure and combinational functions may be significantly changed without requiring major changes in either the design or layout of the PLA structure. The 4-bit branch instruction (BR) field is used, in conjunction with the address field ADF, to choose the next microinstruction address. The truth-table for the instruction PLA (Table 1.1) is split into two parts, table 1.1.i and table 1.1.ii :

(i) shows the result of each instruction in controlling the multiplexer (MUX1) which determines the Y outputs, the incrementer (INC), and the subroutine stack (SS).

(ii) shows the controls to DEC MUX and Loop counter stack (LCS) in decrementer (DEC). Z is the output of the Zero Detector (ZD) which checks for all zeros in the output of the Loop Counter Register (LCR). Z=1 if LCR has all zeros and Z=0 if LCR has at least one bit which is high.

The binary equivalent of Table 1.1, which has to be mapped onto the PLA, is given in Tables 2.6.i and 2.6.ii, found in chapter 2 with discussions of macros. The detailed explanation of instructions is given in Appendix A. Truth-tables are shown on the following pages. The BR field is denoted by B3B2B1B0 and the following abbreviations are used in the truth tables :

IOP : Incrementer output ; A : ADF latches output ;

M : Mapped output ; F : SS stack top;

TABLE 1.1: INSTRUCTIONS FOR PLA

Table 1.1.i :

BR (B3 -B0)	MNEM ONICS	NAME OF THE BRANCH INSTRUCTION	ZD O/P (Z)	TEST BIT (T)			
				T=0		T=1	
				Y	S. STACK	Y	S.STACK
0	JZ	JUMP TO ZERO	X	IOP	CLEAR	IOP	CLEAR
1	CJS	CONDITIONAL JUMP TO SUB-ROUTINE VIA ADF	X	IOP	HOLD	A	PUSH
2	JMAP	JUMP MAP (EXT)	X	M	HOLD	M	HOLD
3	PUSH	PUSH/COND.LOAD DEC	X	IOP	PUSH	IOP	PUSH
4	RFCT	REPEAT LOOP, *LCR NOT 0*	=0 =1	F *IOP*	HOLD *POP*	F *IOP*	HOLD *POP*
5	RACT	REPEAT ADF, LCR NOT 0	=0 =1	A IOP	HOLD HOLD	A IOP	HOLD HOLD
6	CRTN	COND. RETURN	X	IOP	HOLD	F	POP
7	CJA	COND. JUMP ADF	X	IOP	HOLD	A	HOLD
8	LOOP	TEST END LOOP	X	F	HOLD	IOP	HOLD
9	LDCT	LOAD DEC AND CONTINUE	X	IOP	HOLD	IOP	HOLD
10	CONT	CONTINUE (SEQUENTIAL)	X	IOP	HOLD	IOP	HOLD

Table 1.1.ii :

BR (B3-B0)	T	ZD O/P (Z)	LOOP COUNTER STACK (LCS)	DEC MUX	EQUIVALENT DECREMENTER OPERATIONS
0	X	X	PUSH	LC	LOAD A ZERO COUNT
1	X	X	HOLD*	--	HOLD THE DECREMENTER
2	X	X	HOLD*	--	HOLD THE DECREMENTER
3	0	X	HOLD*	--	HOLD THE DECREMENTER
	1	X	PUSH	LC	LOAD A COUNT (N)
4	X	0	HOLD	LD	DECREMENT
	X	1	POP	LL	POP THE STACK
5	X	0	HOLD	LD	DECREMENT
	X	1	POP	LL	POP THE STACK
6	X	X	HOLD*	--	HOLD THE DECREMENTER
7	X	X	HOLD*	--	HOLD THE DECREMENTER
8	X	X	HOLD*	--	HOLD THE DECREMENTER
9	X	X	PUSH	LC	LOAD A COUNT (N)
10	X	X	HOLD*	--	HOLD THE DECREMENTER

The output of the multiplexer DEC MUX is equal to the external count N if the control LC is high, decremented output DOP if LD is high, and the second word of the loop counter stack LCS2 if LL is high. At any one time only one of the controls LC, LD and LL, to DEC MUX, is active. If there is a blank in the DEC MUX field, then all the controls to DEC MUX are low. Decrement is equivalent to loading the decremented output with LD high while holding the LCS, excluding the stack top (HOLD). HOLD* holds the whole stack in its place. The actual controls issued by the PLA to the stacks are different. For details of these controls and how the stack operations are performed using these controls, refer to chapter 2 on macros.

CHAPTER 2

FUNCTIONAL DESCRIPTION OF MICROSEQUENCER MACROS

2.A. INTRODUCTION

This chapter deals with the following macros :

(1) Incrementer

(2) 8-word by 12-bit subroutine stack (SS)

(3) Decrementer with zero detector (ZD) and 4-word by 8-bit loop counter stack (LCS)

(4) Map Logic

(5) Multiplexer 1 (MUX1) implemented as a bus structure

(6) Instruction PLA with latched inputs and buffered outputs (some of the outputs being gated).

(7) ADF Latches

Each macro is described separately and explained with respect to its pin diagram. Also included are execution examples of different macros with simple logic diagrams. For details refer to Appendix B which contains the general description, functional and logic diagrams, floor plan and the circuit schematic of each macro. Since the logic involved in the macros - Subroutine Stack, Map Logic, MUX1, ADF Latches and Instruction PLA - is simple, only the incrementer and the decrementer are explained with execution examples. The decrementer is explained in two parts: (i) DEC LOGIC and (ii) the complete macro.

Switch representations have been used in place of transfer gates in logic diagrams and sometimes in circuit schematics for simplicity. The analogy between different

types of switches and the corresponding transfer gates is shown below in Figure 2.

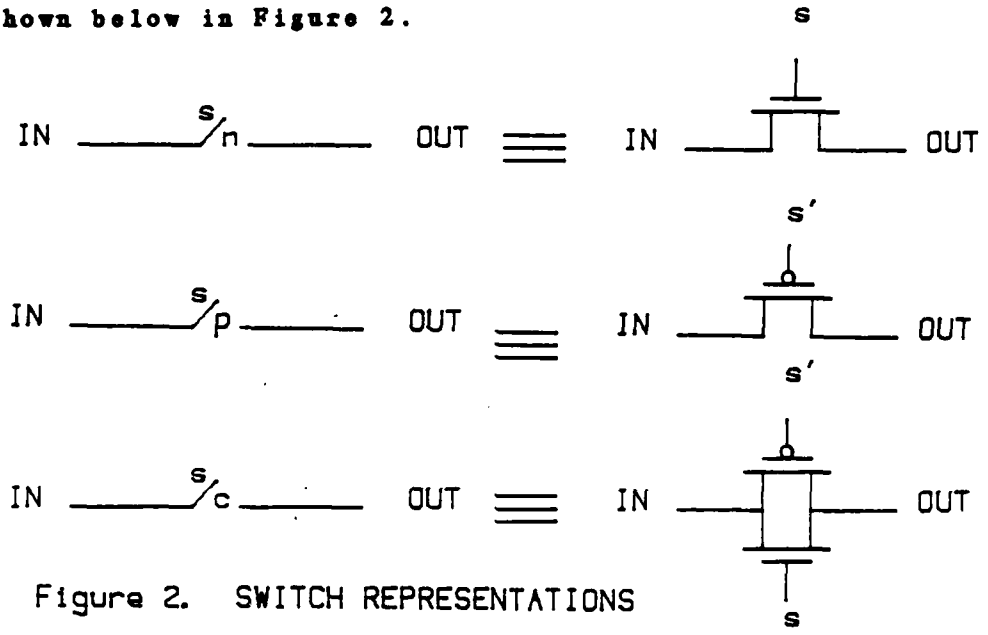


Figure 2. SWITCH REPRESENTATIONS

The type of the switch is denoted by the letter inside the switch; p stands for PMOS, n stands for NMOS, and c stands for CMOS. Whatever the type of the switch is, the switch is closed if the control signal 's' is high (logic 1 for positive logic). S can be a single variable or a boolean expression. GND is the ground (zero volts), VDD is the voltage supply, phase1 and phase2 are two non-overlapping clocks as shown in Figure 1.4 in chapter 1.

Pins 1, 2, 3, and 4 have been reserved for supply, ground, phase1, and phase2, respectively, in all pin diagrams for the sake of consistency. To simplify the 'SIGNAL DEFINITIONS' section, the above mentioned pins are not explained again in the remainder of this chapter. Note that the data is valid during the time shown within the

parentheses in the abbreviation column of all the signal definitions. Refer to Table 2.1, signal defintions of the incrementer, where Y (phase1) in the abbreviation column means that the microprogram address Y is valid in time phase1. Bit '0' is the LSB and bit '11' is the MSB for all data pins except for decrementer data where the MSB is bit '7' since the data is only 8-bits wide as opposed to the 12-bits wide data in all other macros. Also OC_4 is the MSB of the op-code field since the op-code consists only of 5 bits.

The clocks are buffered wherever needed but the buffers are not included in the pin diagrams or logic diagrams since they do not affect the logical operation. For details of the circuits and layouts, refer to chapter 3 on the cell library. The general carry-chain circuit is explained before discussing the incrementer and decrementer carry chains.

2.A.1. CARRY-CHAIN CIRCUIT

It has been decided to implement the fastest possible Manchester-type carry chain having the carry propagation circuit as shown in Figure 2.a.

The carry chain consists of nMOS pass transistors from carry-in (CIN) to carry-out (COUT). The carry chain is precharged high by pMOS transistors, and propagates a low carry signal which it can do quite rapidly. Three signals are used to control the carry-out of each stage. The first is the phaseP' clock which precharges the node associated

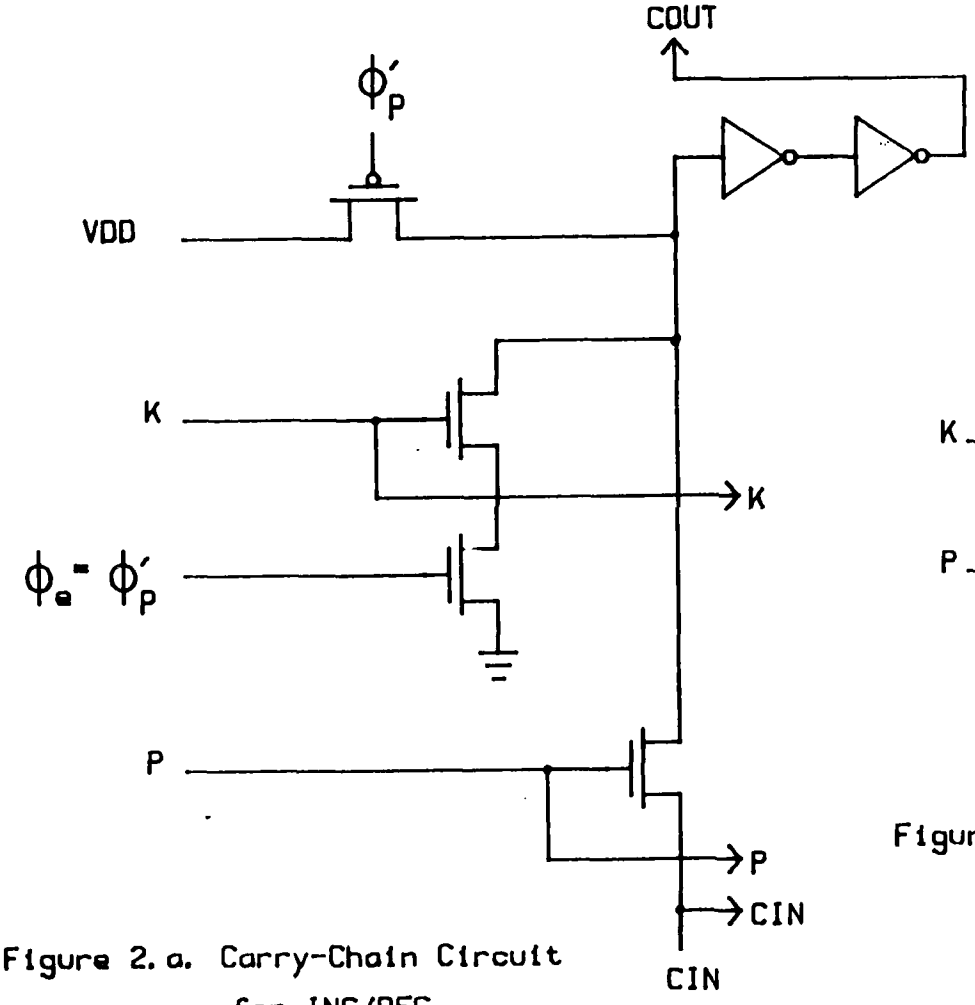


Figure 2.a. Carry-Chain Circuit for INC/DEC

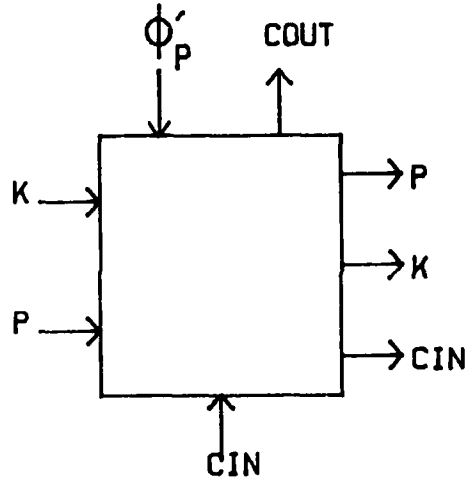


Figure 2.b. Abstraction of Carry-Chain Circuit

with the carry-out of each stage. The second is the carry-kill signal, which simply grounds the carry-out through an nMOS transistor, and a phaseE nMOS evaluation device. PhaseE is the same signal as phaseP' and it is used to make sure that the kill transistor is off during the precharge period. The third is the pass transistor that causes the carry-out to be equal to carry-in. There is no evaluation device added for the pass transistor since both the source and drain of the nMOS pass transistor are precharged high so that the transistor is off even if the input to the pass transistor is high.

The carry-kill (K) and the carry-propagate (P) signals are derived from the inputs to the incrementer or decrementer as explained below and the exact implementation of incrementer and decrementer carry chains are explained in the respective macros. In fact it has been found [5] that nearly all interesting combinations of carry-in and the input signals can be generated using the propagate, kill and the carry-in from each stage. Thus the carry chain can be visualized as a logic block, as in Figure 2.b, with three inputs (carry-kill, carry-in, and carry-propagate), four outputs (propagate, kill, carry-out and carry-in with propagate, kill and carry-in passed through), and a precharge control signal phaseP'.

The carry chain is a part of the normal ALU which can be reduced to incrementer or decrementer functions by designing the necessary functional blocks to (i) combine the

input variables to form the propagate and kill signals, and (ii) combine the propagate, kill and carry-in to form the required output signal (increment or decrement). These steps as implemented in the incrementer and decrementer designs are explained below.

The coding for ALU operations that are commonly found useful shows that for adding two inputs A and B, P is the EX-OR of A and B; K is A'B'. The output is the EX-OR of P and carry-in. To increment a number A, choose B to be all zeros and let the carry-in be equal to 1. This reduces the propagate signal (P) to be equal to the input A and kill signal (K) to be equal to A'. To decrement a number A, B is chosen to be all 1's and the carry-in as 0 so that the propagate becomes A' and kill becomes 0 which means the kill block is not required for the decrementer carry chain. Buffers are interposed in the carry chain occasionally to minimize the propagation delay of the carry through the entire carry chain. After several SPICE * simulations it was been decided to interpose the buffers every fourth stage.

* SPICE is an acronym for Simulation Program, Integrated Circuit Emphasis. It is a general-purpose circuit simulation program for nonlinear dc, nonlinear transient, and linear ac analyses.

2.1. INCREMENTER (INC)

This section shows the pin diagram of the incrementer followed by the signal definitions of the incrementer. An execution example is also presented at the end of the section, to illustrate the logic involved in the incrementer.

Figure 2.1 PIN DIAGRAM OF THE INCREMENTER

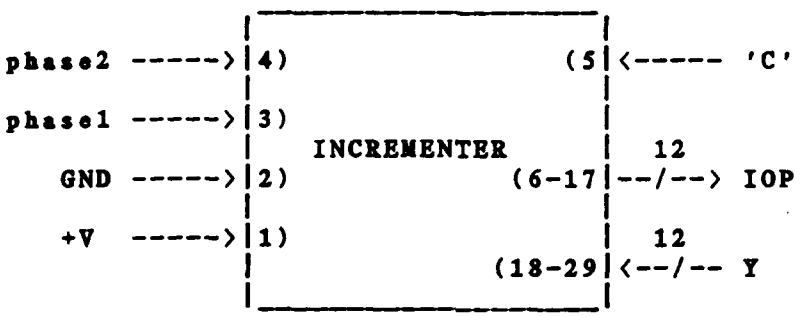


Table 2.1. SIGNAL DEFINITIONS OF THE INCREMENTER

PIN NO(S)	ABBREVIATION	NAME	FUNCTION
6-17	Y (phase1)	Microprogram address	Microsequencer output which is the input to the incrementer. Pin 17 is the LSB. Pin 6 is the MSB.
18-29	IOP (phase1)	Incrementer output	The incrementer output (IOP) is one of the four address sources for the microprogram address(Y). Pin 29 is the LSB. Pin 18 is the MSB.
5	C (phase1)	Clear/ Increment	Control signal from PLA to the incrementer. When high forces the IOP low.

2.1.i. EXECUTION EXAMPLE

Refer to Figure 2.1.i for the execution example of the INC LOGIC.

Instead of 12 bits only 4 bits have been considered for the sake of explanation. Let the value $Y = Y_3\ Y_2\ Y_1\ Y_0 = 1010$. The incrementer consists of an input latch, INC LOGIC (consisting of incrementer carry chain and EX-OR), and an output latch. The inputs (Y) to the carry chain are sampled in phase1, the INC LOGIC is evaluated in phase2 with the output of the INC 1OGIC (I) latched and available (as IOP) in the next phase1. According to the switch level representation, as shown in the figure, the precharge signal (phaseP) for the INC CAR CHAIN is phase2'. But according to the transistor representation the signal that controls the gate of the pMOS transistor is phaseP' which is phase2. The LSB of the INC LOGIC is reduced to that of a simple inverter and a phase2 nMOS switch is added as shown in the figure to make sure that there is no path to ground (in case P_0 is ·0 in phase1) during the precharge period. This is not required at other stages since both the carry-in and the carry-out are precharged high.

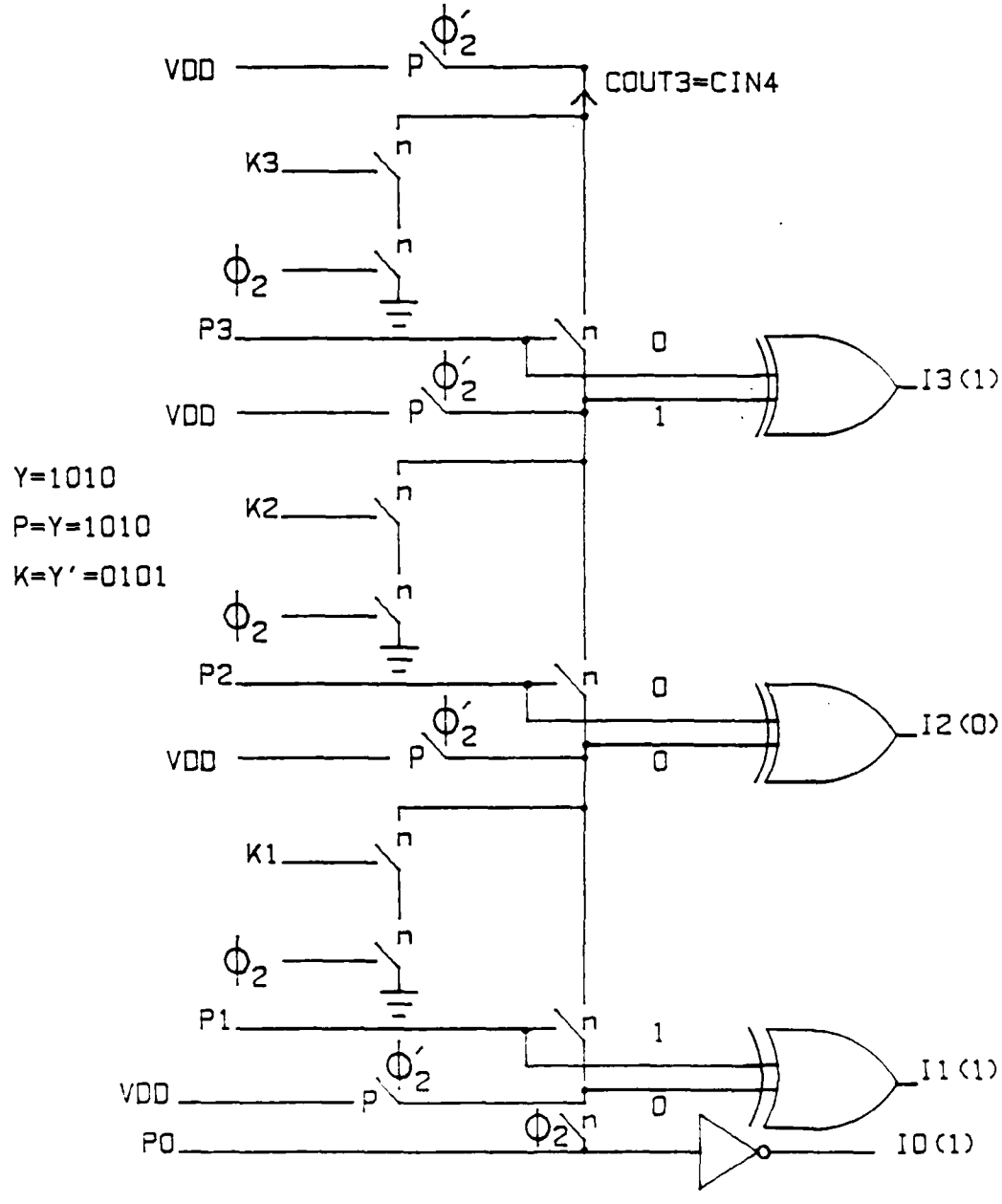


Figure 2.1.1. Execution Example of INC LOGIC

(iii) The switching threshold voltage of a classic CMOS inverter [2], VINV, at which point VIN = VOUT, is given by the following equation:

$$VINV = \frac{VDD + VTP + VTN(BR^{**}1/2)}{(1 + BR^{**}1/2)}$$

where VTP and VTN are the p-channel and n-channel threshold voltages respectively, VDD is the power supply voltage, BR is the ratio of BN (beta of the n-type transistor) to BP (beta of the p-type transistor). The inverter switching point can be changed by changing the relative aspect ratios of the devices. The inverter can detect a '1' input faster if the n-type device is larger than the p-type device, which means the beta ratio should be larger. It can detect a '0' input faster if the P device is larger than the N device i.e. the BR is reduced.

The process used in fabricating these CMOS devices is a p-well process. The general procedure is to start with a p-well on an n-substrate and then synthesize n-type devices in the p-well and p-type devices on the n-substrate. The source-body junction (n+p junction in NMOS devices and p+n junction in PMOS devices) is reverse-biased to restrict the flow of carriers to the channel between source and drain [2].

Figure 3.1.i shows an NMOS device where the body is usually connected to ground. The source-body voltage is denoted by VSB and is non-negative for NMOS devices. The threshold voltage of NMOS transistor is given by the

CHAPTER 3

MICROSEQUENCER CIRCUIT STYLES AND LAYOUTS

This chapter provides the basic descriptions of devices, circuits and fabrication. Basic circuit concepts are explained first, followed by a discussion on clocked CMOS and then illustrating some special circuit applications. Finally, the floor plan of the microsequencer is discussed.

3.1. CIRCUIT CONCEPTS

This section discusses some of the basic CMOS circuit concepts which have been applied in designing the microsequencer. For details of all the circuits and their corresponding layouts, refer to the cell library in Appendix C. It is useful to understand the following points while dealing with circuit design from the device point of view:

(i) The ratio (W/L) of the width, W, to the length, L, of the MOS transistor gate is sometimes referred to as MOS transistor 'aspect ratio'. The gain of the transistor is proportional to the gate aspect ratio, and so is the speed of the logic circuits formed using MOS transistors.

(ii) The product of K'(W/L) is referred to as the 'transistor gain factor', B (beta). K' is the 'process gain factor' which is approximately 2 to 3 times greater for NMOS devices than for PMOS devices.

2.7. ADF LATCHES

This section contains the pin diagram of the ADF Latches along with the table of the signal definitions.

Figure 2.7. PIN DIAGRAM OF THE ADF LATCHES

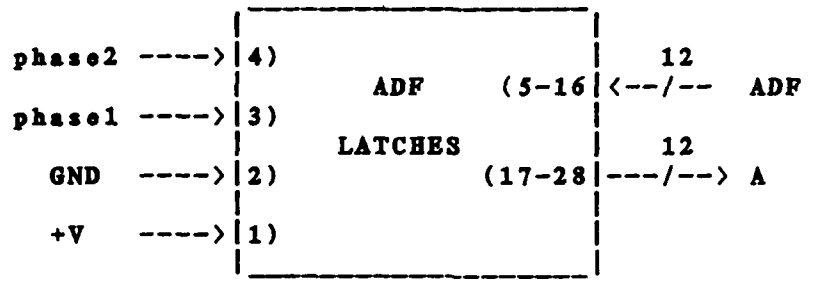


Table 2.7. SIGNAL DEFINITIONS OF THE ADF LATCHES

PIN NO(s)	ABBREVIATION	NAME	FUNCTION
5-16	ADF (phase2)	Explicit address field.	12-bit ADF field of the microinstruction which is sampled in phase2. Pin 16 is the LSB. Pin 5 is the MSB.
17-28	A (phase1)	ADF Latches output	Stable ADF Latches output in phase1. It can be an explicit address source for MUX1 or a count for the decrementer. Pin 21 is the MSB if it is a count; Pin 17 is the MSB if it is an address source. Pin 28 is the LSB in both cases.

Table 2.6.i : (Derived from table 1.1.i in chapter 1.)

	BR FIELD				ZD	T	SELECT ADDRESS				C	CONTROLS FOR SS			
	B3	B2	B1	B0	O/P		SA	SI	SF	SM		ø1F	ø2F	ø1B	ø2B
0	0	0	0	0	X	X	0	1	0	0	1	1	1	0	0
1	0	0	0	1	X	0	0	1	0	0	0	0	1	1	0
					X	1	1	0	0	0	0	1	1	0	0
2	0	0	1	0	X	X	0	0	0	1	0	0	1	1	0
3	0	0	1	1	X	0	0	1	0	0	0	1	1	0	0
					X	1	0	1	0	0	0	1	1	0	0
4	0	1	0	0	0	X	0	0	1	0	0	0	1	1	0
					1	X	0	1	0	0	0	0	0	1	1
5	0	1	0	1	0	X	1	0	0	0	0	0	1	1	0
					1	X	0	1	0	0	0	0	1	1	0
6	0	1	1	0	X	0	0	1	0	0	0	0	1	1	0
					X	1	0	0	1	0	0	0	0	1	1
7	0	1	1	1	X	0	0	1	0	0	0	0	1	1	0
					X	1	1	0	0	0	0	0	1	1	0
8	1	0	0	0	X	0	0	0	1	0	0	0	1	1	0
					X	1	0	1	0	0	0	0	0	1	1
9	1	0	0	1	X	X	0	1	0	0	0	0	1	1	0
10	1	0	1	0	X	X	0	1	0	0	0	0	1	1	0

Table 2.6.ii : (Derived from table 1.1.ii in chapter 1.)

	BR FIELD				ZD	T	CONTROLS FOR THE DECREMENTER							
							LOOP COUNTER STACK (LCS)					DEC MUX		
	B3	B2	B1	B0	O/P		ø1BL*	ø1FL	ø2FL	ø1BL	ø2BL	LC	LD	LL
0	0	0	0	0	X	X	0	1	1	0	0	1	0	0
1	0	0	0	1	X	0	1	0	1	1	0	0	0	0
					X	1	1	0	1	1	0	0	0	0
2	0	0	1	0	X	X	1	0	1	1	0	0	0	0
3	0	0	1	1	X	0	0	1	1	0	0	0	0	0
					X	1	1	0	1	1	0	1	0	0
4	0	1	0	0	0	X	0	0	1	1	0	0	1	0
					1	X	0	0	0	1	1	0	0	1
5	0	1	0	1	0	X	0	0	1	1	0	0	1	0
					1	X	0	0	0	1	1	0	0	1
6	0	1	1	0	X	0	1	0	1	1	0	0	0	0
					X	1	1	0	1	1	0	0	0	0
7	0	1	1	1	X	0	1	0	1	1	0	0	0	0
					X	1	1	0	1	1	0	0	0	0
8	1	0	0	0	X	0	1	0	1	1	0	0	0	0
					X	1	1	0	1	1	0	0	0	0
9	1	0	0	1	X	X	0	1	1	0	0	1	0	0
10	1	0	1	0	X	X	1	0	1	1	0	0	0	0

2.6. INSTRUCTION PLA

This section consists of the pin diagram of the instruction PLA, table of the signal definitions, and the binary equivalent of the truth-table for the instruction PLA (Table 2.6.).

Figure 2.6. PIN DIAGRAM OF THE INSTRUCTION PLA

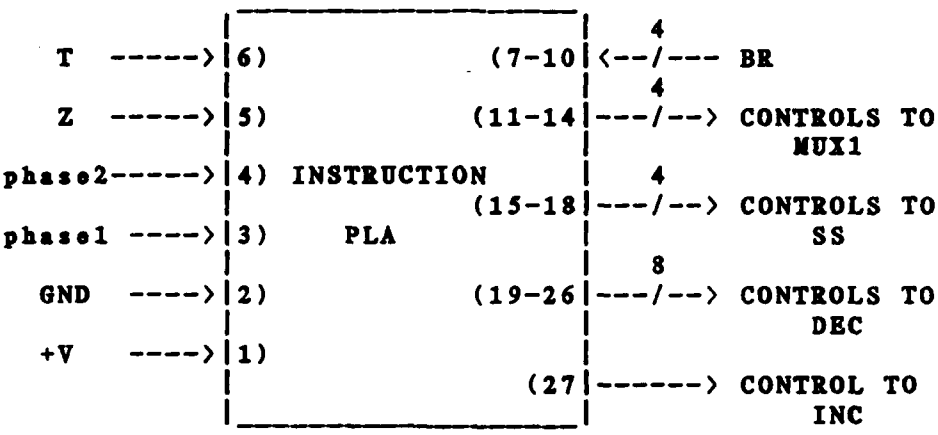


Table 2.6. SIGNAL DEFINITIONS OF THE INSTRUCTION PLA

PIN NO(S)	ABBREVIATION	NAME	FUNCTION
5	Z (phase2)	ZD output	Zero detector output, which is an input to PLA.
6	T (phase2'* phase1')	Test bit	Test bit comes from the data path. Set by the status register according to the 3-bit condition (CD) field of the microinstruction. T is one of the inputs to PLA in phase1.
7-10	BR (phase2)	4-bit branch field	4-bit branch instruction field from the microinstruction. Selects one of the 11 sequence control instructions, most of which are conditional on Z or T.

NOTE : The controls have already been explained in conjunction with other macros of the microsequencer.

2.5. MULTIPLEXER 1 (MUX1)

This section contains the pin diagram and the signal definitions of MUX1.

Figure 2.5. PIN DIAGRAM OF MUX1

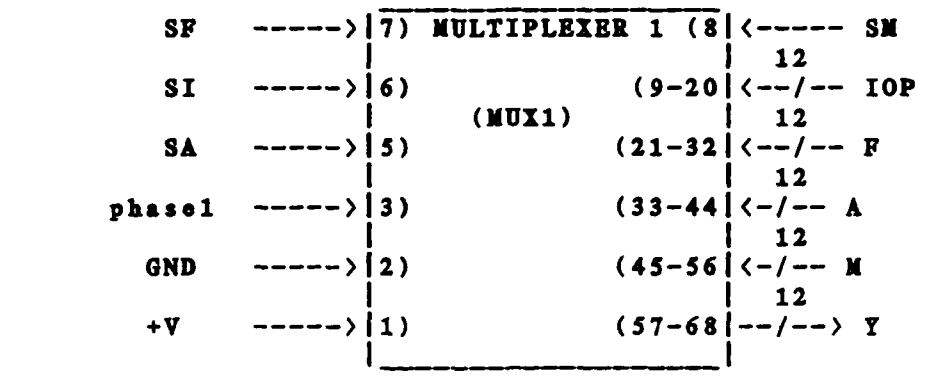


Table 2.5. SIGNAL DEFINITIONS OF MULTIPLEXER 1

PIN NO(s)	ABBREVIATION	NAME	FUNCTION
57-68	Y (phase1)	Microprogram address	MUX1 output, which is the microsequencer output.
45-56	M (phase1)	Mapped output	Map Logic output, which is an address source (M).
33-44	A (phase1)	ADF Latches output	Stable address source for MUX1 in phase1.
21-32	F (phase1)	SS output	SS Stack top (F), which is the return address.
9-20	IOP (phase1)	Incrementer output	Sequential address.
8	SM (phase1)	Select M	MUX1 selects the Mapped output (M).
7	SF (phase1)	Select F	MUX1 selects the return address from SS (F).
6	SI (phase1)	Select IOP	MUX1 selects the output of the incrementer (IOP).
5	SA (phase1)	Select A	MUX1 selects the ADF latches output (A) .

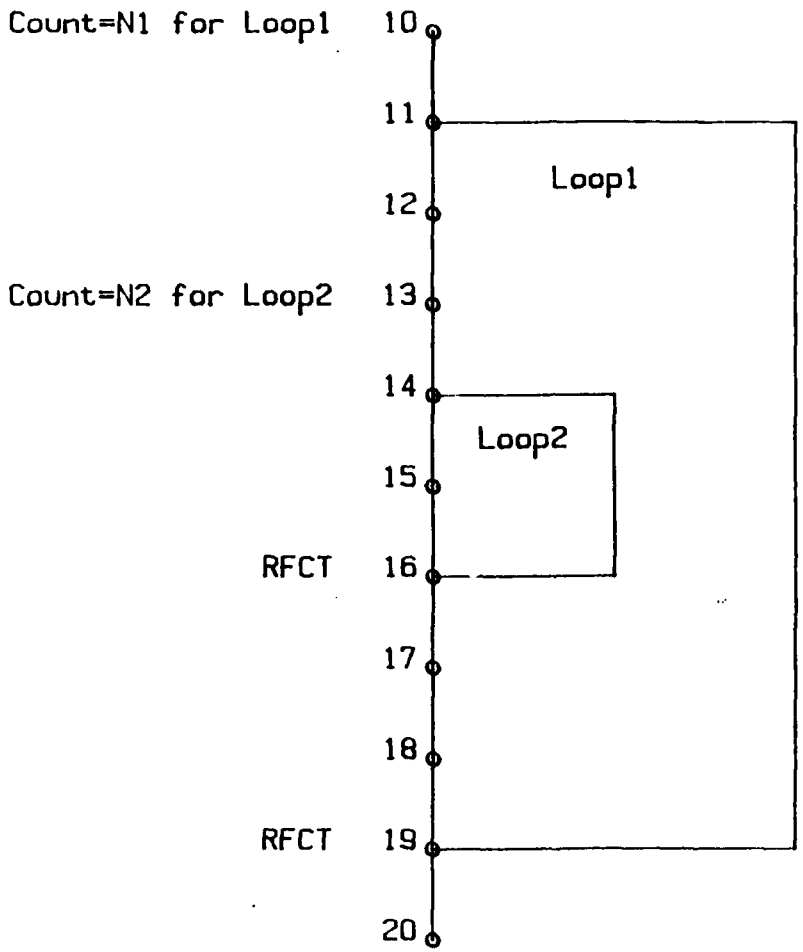


Figure 2.3.iii. Example of Nested Loops

Assume that a count 'N1' is loaded into the DEC at microprogram address 10 and let the RFCT instruction be at locations 16 and 19. Microinstruction at address 12 is executed and at 13 a new count 'N2' is loaded for loop 2, pushing the previously loaded count onto the stack. Refer to Figure 2.3.iii, which shows the nesting.

Loop2 is executed (N2+1) times since zero detection is done prior to decrementation as explained in the instruction set section of Appendix A. When the exit at loop2 occurs, the microprogram control falls through 17 to 19. Since N1 is decremented once, the control goes back to address 11 to execute loop1. When the control goes to address 13, the count N2 is pushed onto the top of the stack again but this time pushing the decremented count (N1-1) onto the stack. In this way Loop2 is executed (N1+1)*(N2+1) times. Refer to Appendix A for the exact sequence of execution of branch instructions.

2.4. MAP LOGIC

This section consists of the pin diagram of the MAP Logic along with the table of the signal definitions.

Figure 2.4. PIN DIAGRAM OF THE MAP LOGIC

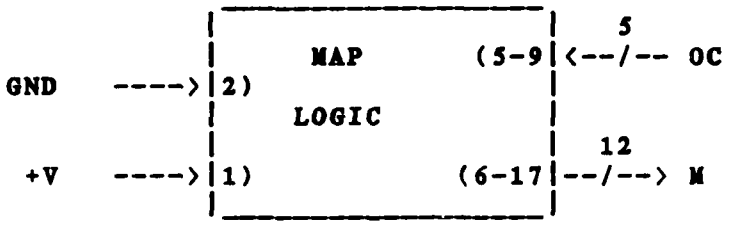
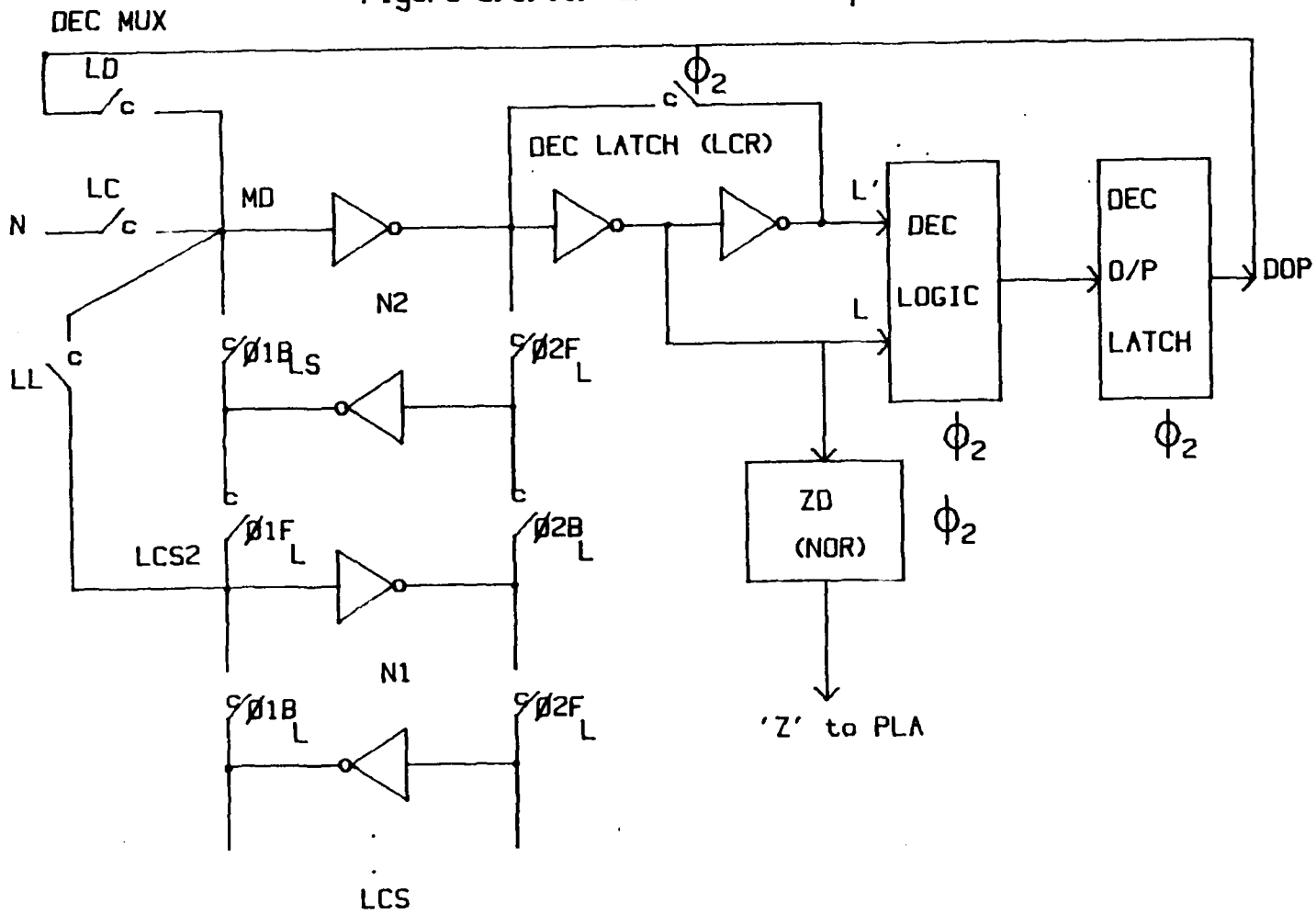


Table 2.4. SIGNAL DEFINITIONS OF MAP LOGIC

PIN NO(s)	ABBREVIATION	NAME	FUNCTION
5-9	OC (phase2)	Operation code (op-code)	Macro-operation part of the machine instruction. Pin 9 is the LSB. Pin 5 is the MSB.
6-17	M (phase1)	Mapped output	The Map logic converts the 5-bit op-code into a 12-bit Mapped output by clearing the 7 most significant bits and transferring the 5-bit op-code into the 5 least significant positions.

Figure 2.3.ii. Execution Example of Decrementer



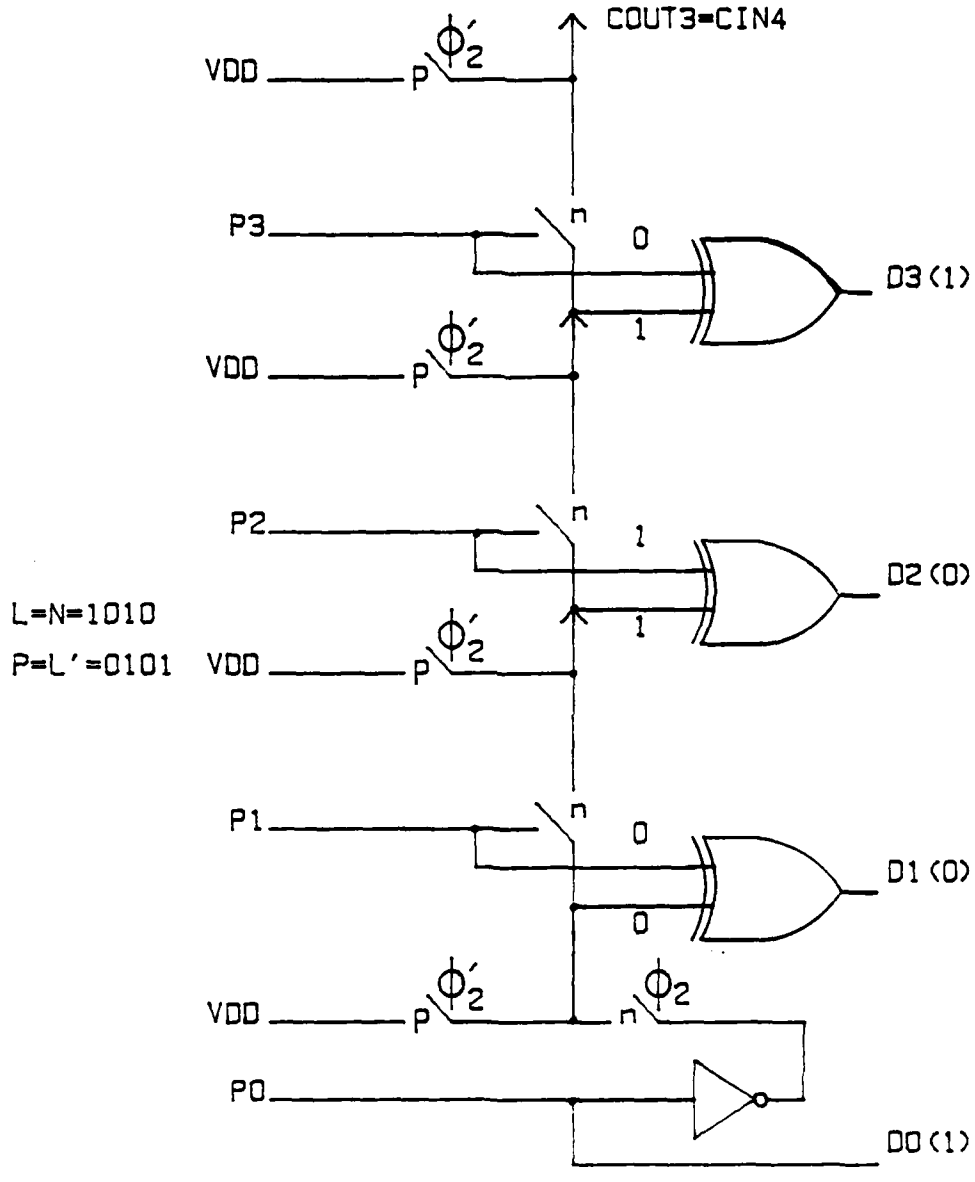


Figure 2.3.1. Execution Example of DEC LOGIC

2.3.i. EXECUTION EXAMPLE OF DEC LOGIC

Refer to Figure 2.3.i.

DEC LOGIC consists of DEC CAR CHAIN and EX-OR as shown in Figure 2.3.i. Instead of 8 bits of the count only 4 bits are considered for explanation. Choose $L = L_3\ L_2\ L_1\ L_0 = 1010$ and $L_3'\ L_2'\ L_1'\ L_0' = 0101$. The carry chain is precharged in phase2' and evaluated in phase2. The LSB is modified to reduce the logic to a simple inverter and a phase2 nMOS switch is added to disable the path to ground which may exist if the LSB of L is 0 when it is sampled in phase1.

2.3.ii. EXECUTION EXAMPLE OF DECREMENTER (DEC)

Refer to Figure 2.3.ii.

ZD is a NOR gate whose inputs are the 8 bits of LCR (L). The 4 main operations of the decrementer are LOAD, HOLD (HOLD* holds the whole stack in its place), DECREMENT (load the DOP back into the DEC while holding the stack, excluding the stack top), and POP as shown below :

OPERATIONS	ACTIVE CONTROLS FOR LCS	STACK OPERATION	FOR DEC MUX
LOAD the DEC (with an external count)	phase1F_L phase2F_L	PUSH	LC * phase1
HOLD the DEC	phase1B_L phase1B_{LS} phase2F_L	HOLD*	—
DECREMENT	phase1B_L phase2F_L	HOLD	LD * phase1
POP (the whole stack)	phase1B_L phase2F_L	POP	LL * phase1

16	phase1F_L (phase1)	LCS phase1 forward	* * Subscript 'L' indicates a control signal from the PLA to the loop counter stack which is a push-down stack similar to SS. The loop counter stack's first word (stack top) is different from that of SS and has a special control phase1B_{LS} for the stack. This control is used with the other four stack controls to control the LCS. The subscript 'L' in all these control signals shows that they are associated with LCS.
17	phase2F_L (phase2)	LCS phase2 forward	
18	phase1B_L (phase1)	LCS phase1 back	
19	phase2B_L (phase2)	LCS phase2 back	
20	phase1B_{LS} (phase1)	Stack top control for LCS	
8-15	N (phase1)	External count (N)	The 8 least significant bits of the ADF field of the microinstruction are used as count for looping. Pin 15 is the LSB. Pin 8 is the MSB.
21	Z (phase2)	Zero detector (ZD) output	The ZD checks for all zeros in the output of loop counter register (DEC LATCH in DEC). Z is high, when the LCR has all zeros. Z is one of the inputs to the PLA. LD is high as long as Z is low, i.e., LCR ≠ 0.

The stack operations are performed as explained below :

PUSH : phase1F_L high in phase1 with phase2F_L high in phase2.

POP : phase1B_L and LL high in phase1 with phase2B_L high in phase2.

HOLD (To hold the stack excluding the stack top) : phase1B_L high in phase1 with phase2F_L high in phase2.

HOLD* (To hold the stack including the stack top) : phase1B_L and phase1B_{LS} high in phase1 with phase2F_L high in phase2.

CLEAR: PUSH a zero.

2.3. DECREMENTER WITH ZD AND 4-WORD BY 8-BIT LCS (DEC)

The following section contains the pin diagram of the decrementer, table of the signal definitions, and an execution example illustrating the decrementer logic.

Figure 2.3. PIN DIAGRAM OF THE DECREMENTER

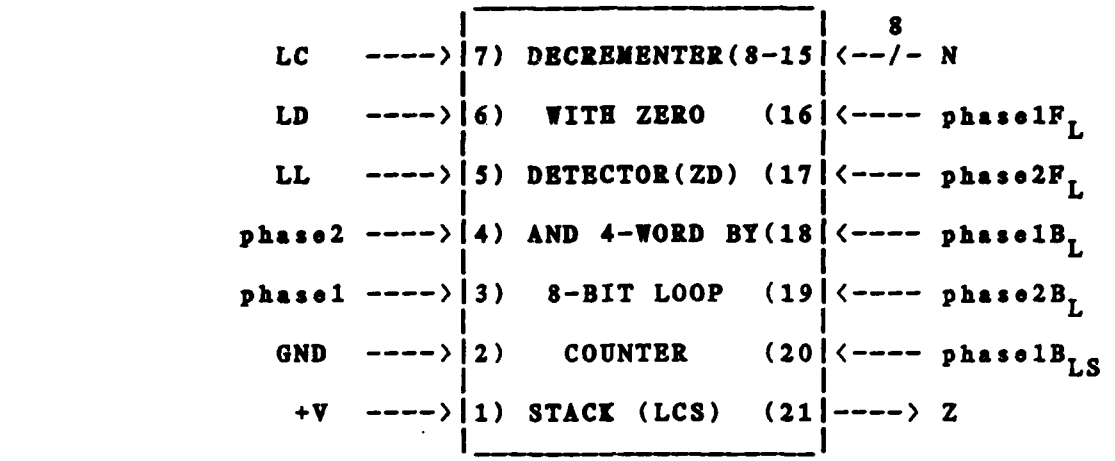


Table 2.3. SIGNAL DEFINITIONS OF THE DECREMENTER

PIN NO(s)	ABBREVIATION	NAME	FUNCTION
6	LD (phase1)	Load decrementer output (DOP)	Control signal from PLA to decrementer. When high, loads the DOP back into the DEC. (Internal operation.)
7	LC (phase1)	Load external count (N)	Control signal from PLA to DEC. When high, loads count N into the DEC.
5	LL (phase1)	Load second word (LCS2) of LCS	Control signal from PLA to decrementer. When high, loads the second word of the LCS into the DEC LATCH. (Internal operation.) Stack top is the first word.

NOTE : Decrementation is performed unconditionally but the decremented output (DOP) is loaded if LD is high. LC, LD, and LL are select controls for the DEC MUX (in DEC) which selects between the external count (N), the DEC output (DOP), and the second word of the LCS (LCS2) to be loaded into the loop counter register (LCR) of the decrementer.

2.2. SUBROUTINE STACK (SS)

This section consists of the pin diagram of the subroutine stack along with the signal definitions.

Figure 2.2. PIN DIAGRAM OF THE SUBROUTINE STACK

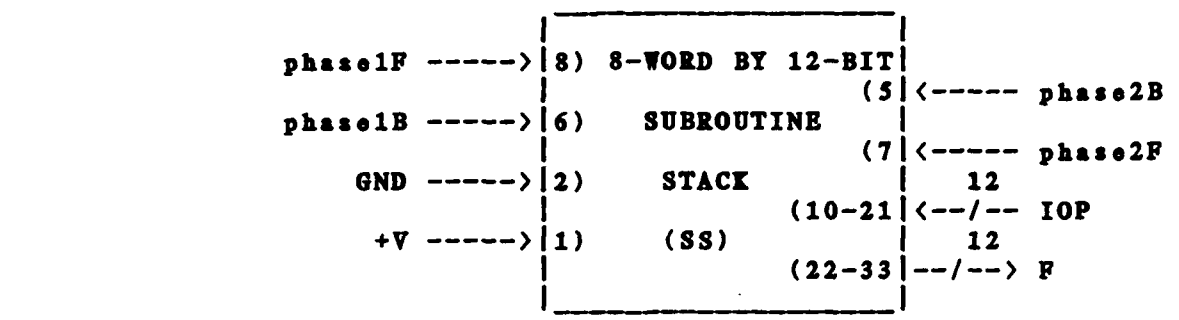


TABLE 2.2. SIGNAL DEFINITIONS OF THE SUBROUTINE STACK

PIN NO(s)	ABBREVIATION	NAME	FUNCTION
10-21	IOP (phase1)	Incrementer output	Input to the subroutine stack. Pin 21 is the LSB. Pin 10 is the MSB.
22-33	F (phase1)	Subroutine stack (SS) output	The SS output (Stacktop) which is the return address. Pin 33 is the LSB. Pin 22 is the MSB.
8	phase1F (phase1)	phase1 forward	* * phase1F, phase2F, phase1B, phase2B are the control signals from PLA to subroutine stack. phase1F, phase1B are active in phase1; and phase2F, phase2B are active in phase2. The stack operations PUSH, POP, HOLD, CLEAR are performed as explained below :
7	phase2F (phase2)	phase2 forward	
6	phase1B (phase1)	phase1 back	
5	phase2B (phase2)	phase2 back	

PUSH : phase1F high in phase1 with phase2F high in phase2.
POP : phase1B high in phase1 with phase2B high in phase2.
HOLD : phase1B high in phase1 with phase2F high in phase2.
CLEAR: PUSH a zero.

equation VT = VTO +]VT where]VT is a function of VSB and p-well doping, VTO is the threshold voltage with VSB=0.

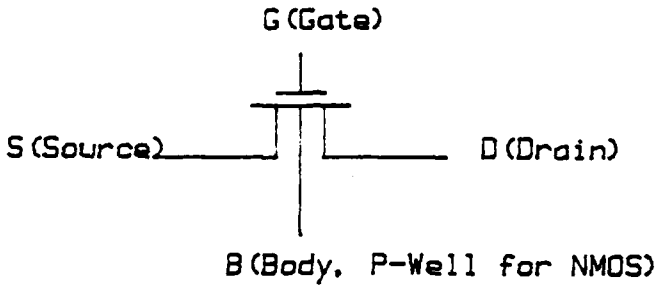


Figure 3.1.1. NMOS Device

The source of the NMOS device can go up to one threshold below the drain voltage when both gate and drain are tied to +V. As the source voltage increases, VSB increases, causing an increase in]VT. Also]VT is high for higher doping density of the p-well. (The doping of the p-well has to be sufficient enough to overcome the substrate doping [2]). Due to these reasons the effective threshold voltage rises with an increase in the source voltage. This phenomenon is called body-effect. Hence the source-follower mode, where the output is the source, is undesirable with NMOS devices in a p-well process. Care has been taken in this project to operate the devices in common source mode where NMOS devices are used for discharging and PMOS devices are used for charging, thus reducing the body-effect problems.

Unless otherwise stated, the W/L ratio of all the devices (both p-type and n-type) is chosen as 2:1, with L=2 lambda being fixed for the technology. Lambda is the basic unit of length measurement in doing layouts and is equal to the fundamental resolution of the fabricating process itself [5]. This ratio has been chosen as a compromise between transistor-gain and layout-density. If the W/L ratio is more than 2, the layout area increases; if it is less than 2, the gain decreases [9]. The latches used throughout the design are classic CMOS D latches [9]. Refer to Figure 3.1.ii for a logic diagram. Input D is sampled when switch S is closed and latched when switch H closes. The output Q is stable when the feedback switch is closed. For the circuit schematic and layout refer to the cell library in Appendix C. Circuits are buffered wherever possible to achieve better voltage transfer characteristics and larger noise margins than unbuffered gates.

3.2. CLOCKED CMOS

The problem with classic CMOS is that it requires a pair of transistors for each gate input. Refer to Figure 3.2.a for an 8-input classic CMOS NOR gate which requires 8 n-transistors and 8 p-transistors. Moreover, the series combination of 8 p-transistors results in very slow circuit operation. Therefore a clocked version of CMOS [7] is used in the design which achieves higher layout density and better performance. Since dynamic (clocked) techniques are utilized, careful considerations of charge splitting [7] are

observed. The signals in MOS devices are stored as charges on capacitors. Charge splitting is a phenomenon of signal losses due to the charges split by parasitic capacitances, which are invariably present in MOS devices. Figure 3.2 illustrates two different versions of clocked CMOS gates which use parallel connected NMOS devices. (There are several other versions [7], but only those used in this design are discussed.) There are two families of clocked CMOS : the ripple variety without the evaluation device, and the gated variety which responds to an evaluation signal. Note that n-type logic is complementary to p-type logic, i.e., parallel p-type logic NAND gate is equivalent to a series n-type logic NAND gate, and vice-versa.

3.2.i. NGa CIRUIT

Figure 3.2.i shows one version of the gated variety of clocked CMOS called the 'NGa' type circuit since it refers to the gated variety, and the fact that its inputs must be driven from an 'active' sourcing output to prevent charge splitting. As illustrated in the figure, the outputs driving this gate must be able to supply current from the positive supply for a '1' input when the evaluation device is turned on. During precharge the output of the NGa gate is precharged high. One input is presumed to be precharged high, resulting in the (common) sources of the logic devices being precharged high to within one threshold below +V. The channel is only weakly induced in this case. When the evaluation device is turned on, the source node is pulled to

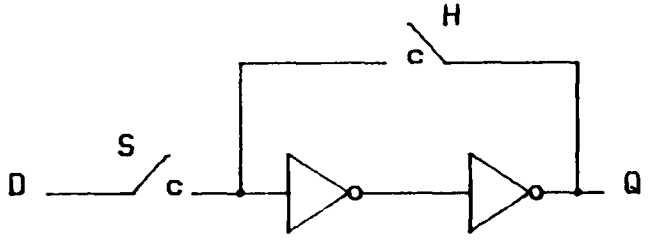


Figure 3.1.ii. Classic CMOS D Latch

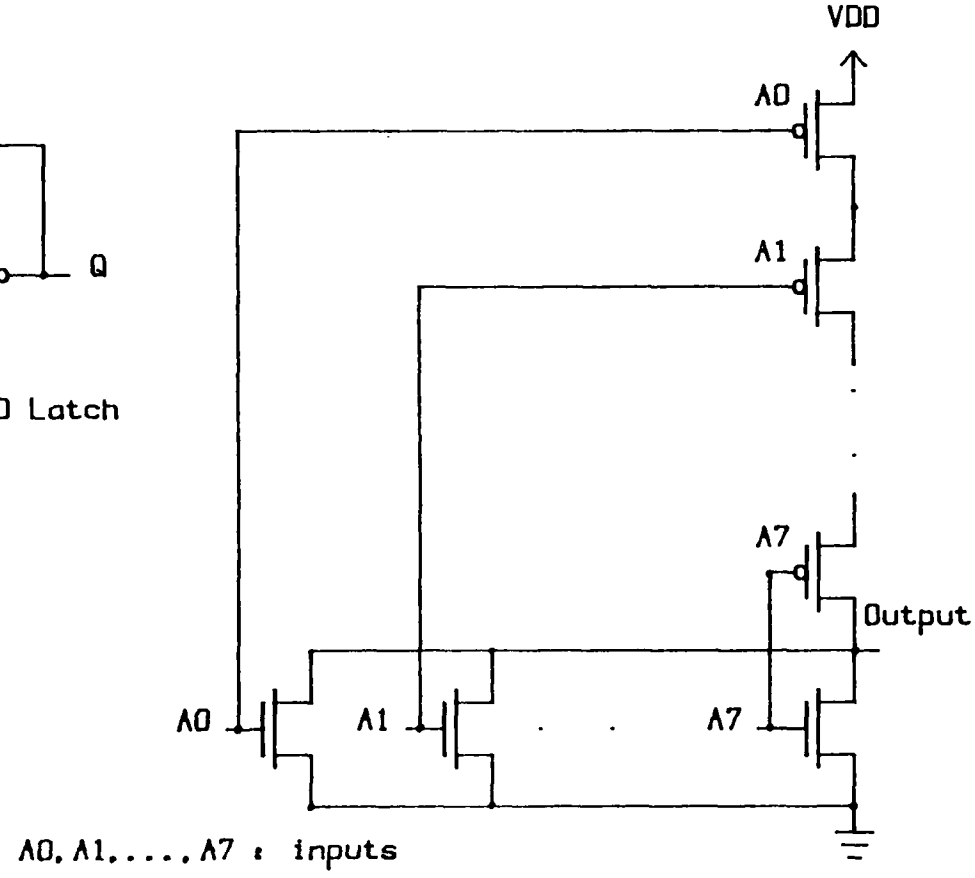


Figure 3.2.a. Eight-Input Classic CMOS NOR Gate

ground with coupling through the full gate capacitance to the input nodes.

Note that if there is no active source for charging this coupling capacitance, then the input signal level is pulled toward ground, reducing the effective drive of the input. The magnitude of this signal loss depends on the effective node capacitance of the input and the coupling capacitance of the gate. A classic inverter or a P-gate can supply the necessary current to provide the full input drive voltage. Note also that the evaluation device can be shared with other NGa gates.

3.2.ii. NGc CIRCUIT

Figure 3.2.ii shows the other version of gated variety called an 'NGc' type circuit which has the evaluation device between the output node and the common drains of the logic devices. The 'c' in NGc refers to the fact that for large fan-in, the circuit requires a large capacitance at the output node to minimize charge splitting. During precharge the output is precharged high and the common drains of the logic devices are presumed to be discharged low. If the inputs are subsequently discharged to ground and then the evaluation device is turned on, only the output capacitance can supply the current to charge the common drain node which has a large capacitance on it. The resulting charge splitting may reduce the ouput signal to an unacceptable level; e.g., if the output is driving a simple inverter (minimum fan-out) and the fan-in is large, then the

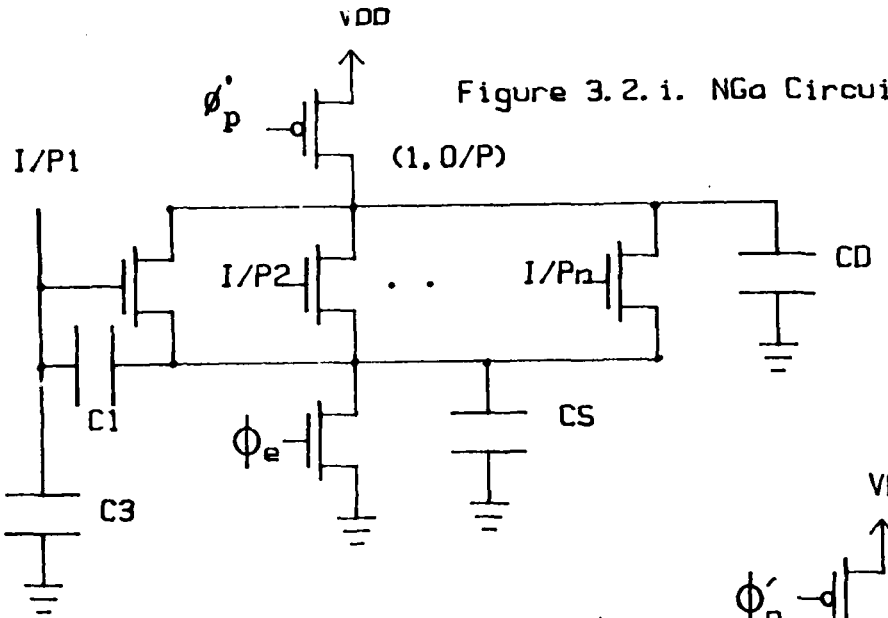


Figure 3.2.i. NGa Circuit

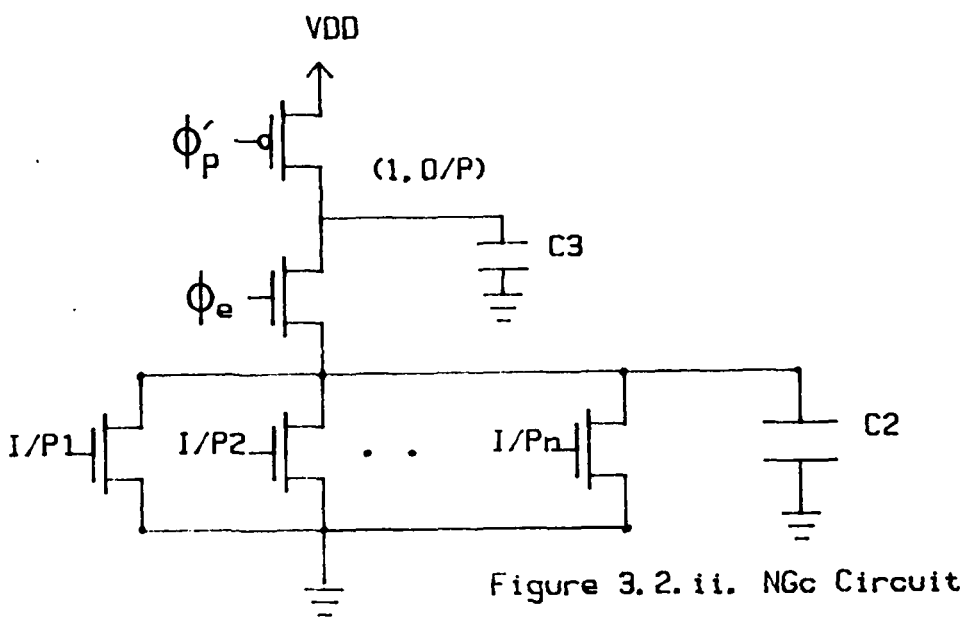


Figure 3.2.ii. NGc Circuit

large effective capacitance on the common drain node may well dominate the small load capacitance, leading to effectively less than '1' output. Both the versions, NGa and NGc, provide the NOR logic function.

3.2.iii and iv. PGa and PGc

Dual versions exist in the P-Gate type but the logic function is NAND, rather than NOR, since parallel p-type logic is equivalent to series n-type logic [9]. PGa type circuits are similar to the NGa forms from a circuit point of view with parallel P devices used instead of parallel N devices; the output is precharged low by an N-transistor and conditionally pulled high by a P-transistor. Refer to Figure 3.2.iii for a PGa circuit. The principal problem with the PGa type circuit is that it cannot supply a good '1' (greater than the switching threshold of the next stage) at the output if not driven from an active source. The PGc type is shown in Figure 3.2.iv and requires a large capacitance at the output node to avoid charge splitting.

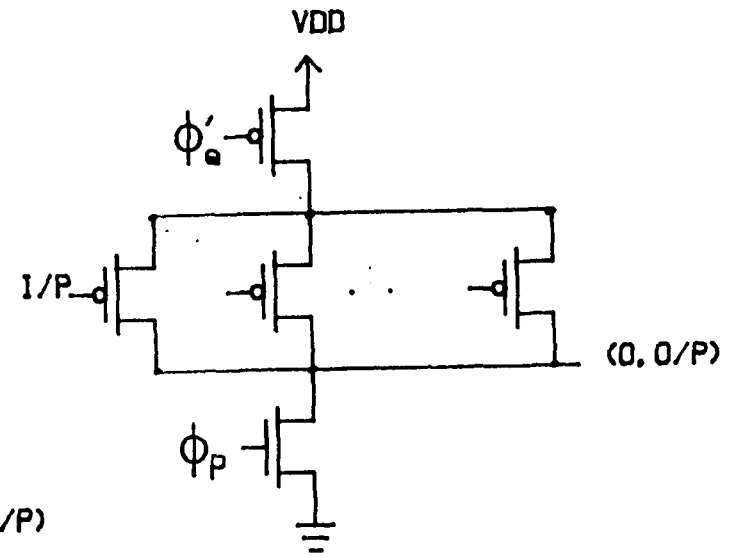
3.3. SPECIAL CIRCUIT APPLICATIONS

This section discusses only special circuit applications intended to illustrate some of the concepts presented in the preceding sections.

3.3.i. CARRY CHAIN

The general carry chain concept has already been explained in chapter 2 on macros, and the INC/DEC carry chains are explained in detail in chapter 2 and Appendix B;

Figure 3.2.iii. PGa Circuit



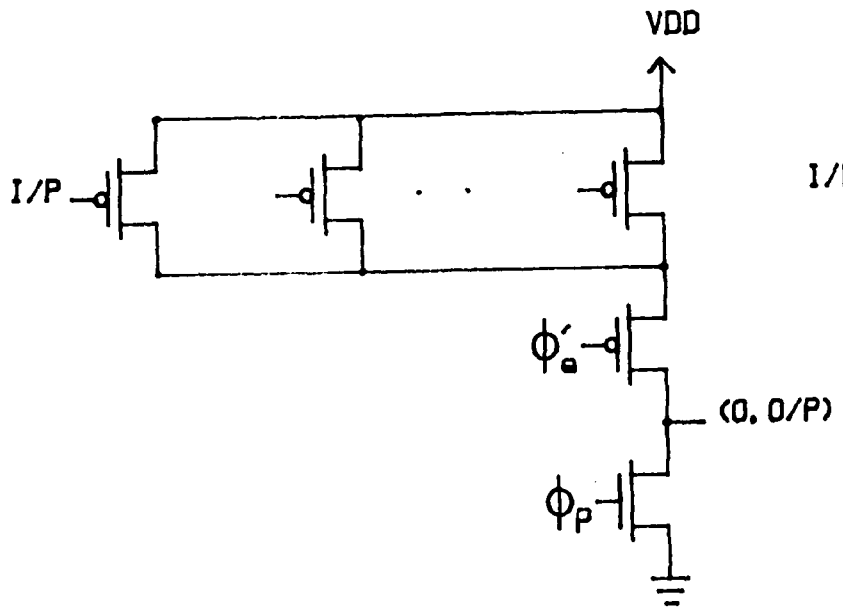


Figure 3.2.iv. PGc Circuit

so the circuit diagram is not presented here. It has been mentioned in chapter 2 that the buffers have been interposed every fourth stage of the carry chain to minimize the propagation delay. This has been verified by SPICE simulation. Also, the aspect ratios of the devices used in the carry chain played an important role in determining the propagation delay. Simulations of the carry chain indicated that the propagation delay reduces with an increase in the W/L ratios of the pass transistors. The W/L ratio of the pass transistors has been chosen to be 3:1 since increasing the W/L ratio further increases the input capacitance on the pass transistors which will subsequently load the latches that drive the carry chain.

The main problem encountered with the carry chain, as already mentioned, is the propagation of the '0' signal. The BR of the first inverter used in the buffers, interposed every fourth stage in the carry chain, is decreased (larger P device) so that it can detect a '0' signal faster. The BR of the second inverter in the buffer has a high beta ratio (larger N device) so that it can detect a '1' signal faster. The output of the buffer is the carry-in to the next stage. This reduces the delay relative to buffers having regular W/L ratios (2:1).

3.3.ii. ZERO DETECTOR (ZD)

The ZD used in the decrementer is an NGa type as discussed earlier in the chapter. Refer to Figure 3.2.i for this circuit diagram. The ZD is an 8-input NOR gate which

checks for all zeros in the loop counter register (LCR). The NGa version has been chosen since the LCR (DEC LATCH) provides the active inputs for the gate. Refer to Figure 2.3 of the decrementer in chapter 2 for the interface between LCR and ZD. Note that this is a clocked NOR gate which uses 8 n-transistors (in parallel), one p-type precharge device and one n-type evaluation device as opposed to the classic CMOS NOR gate which requires 8 n-transistors and 8 p-transistors.

3.3.iii. Programmable Logic Array (PLA)

The basic PLA concept has been explained in chapter 2 and Appendix B. The discussion which follows is on the input/output circuitry for the PLA. Since the inputs needed to be stable by the time the evaluation device is turned on, they are sampled in the appropriate time and are stable in phase1, as explained in the PLA macro of Appendix B.

The inputs to the PLA must be complemented to drive the AND-plane. An initial decision was made to gate the inputs to the AND-plane in phase1 (make them active), as shown in Figure 3.3. This was done to precharge the AND-plane inputs low, which drive series n-type devices, so as to turn them off during the AND-plane precharge period. Hence the evaluation device can be deleted to reduce the number of n-transistors in series and thus reduce the delay in the AND-plane. However there are charge splitting considerations with this scheme.

Refer to Figure 3.4.i for the interface between the AND-plane and the gating devices. During the precharge period the output of the AND-plane is precharged high by the p-type device and all the inputs to the AND-plane are precharged low by the gating device. Let us assume that by the time the evaluation device is turned on, the inputs ABCDE are 11100 so that all the series N transistors are on except the two least significant ones; and the NAND gate output should be '1'. Assume that node X did not get charged, since the n-transistors were off during precharge, so that it stays low. Node X may act as a 'virtual ground', discharging the output 'S' through the conducting n-type devices leading to an incorrect output signal. This problem may arise if the number of series N devices in the AND-plane is large. Also, there should be a gating device for each input and its complement. The inputs must be complemented before they are used by the gating circuitry; so inverters must be added to provide the complements. For these reasons, the concept of gating the PLA inputs was abandoned.

Latching the PLA inputs, providing the complements, and gating the AND-plane circuitry has been selected for this design. The inputs are sampled in phase1' (except for the T bit which is sampled in phase1'*phase2' following phase2) and are stable in phase1. Refer to Figure 3.4.ii for the interface between the AND-plane and the PLA input latches. The evaluation devices are added with this scheme

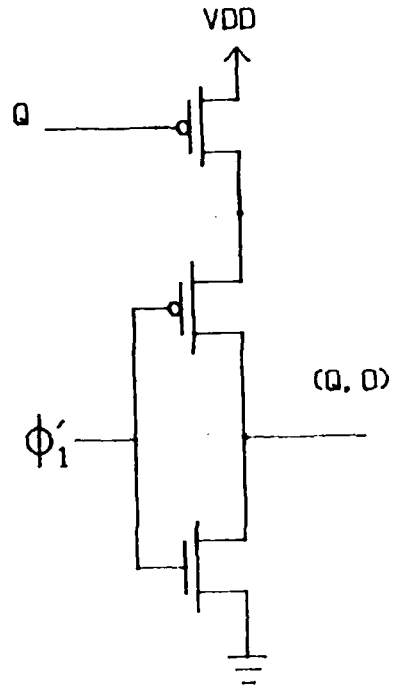


Figure 3.3 Gating Device

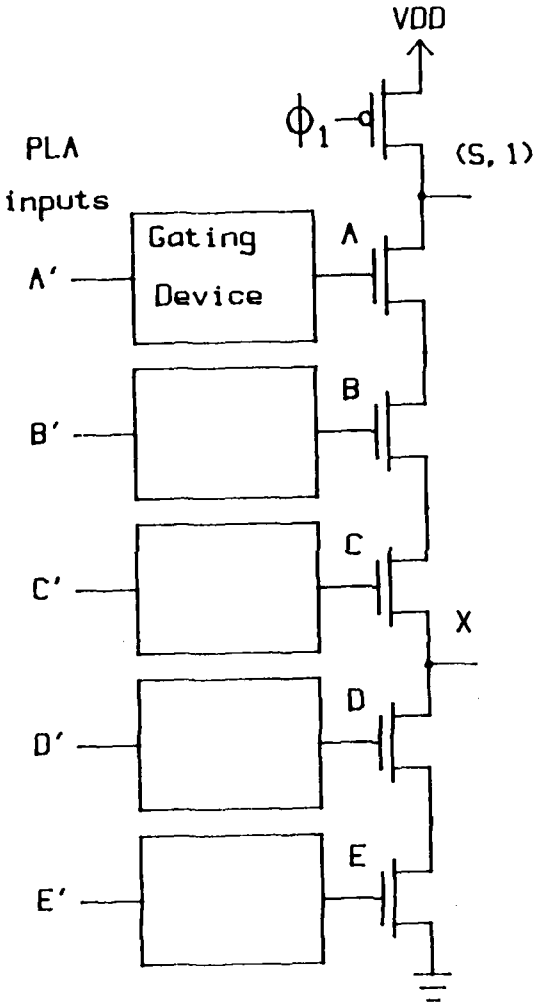


Figure 3.4.i. Gating Devices with PLA inputs

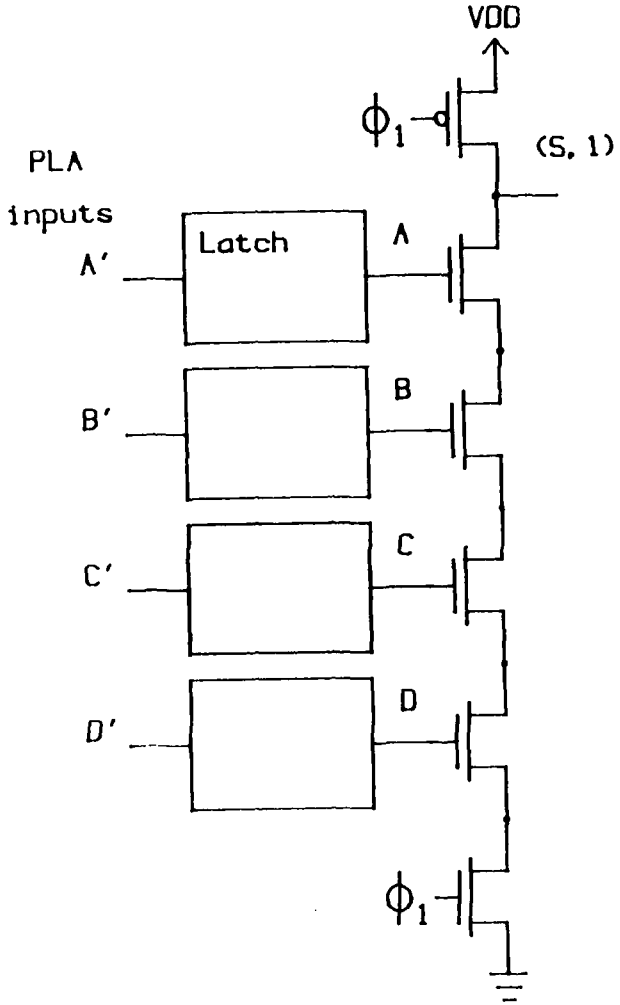


Figure 3.4.ii. Latches for PLA inputs

to ensure that there is no path to ground during the precharge period. Note that the inputs to the AND-plane may be high during precharge since they are driven by latches. This scheme facilitates the series N devices in the AND-plane being precharged before evaluation takes place. This will minimize the problem of incorrect logic due to 'virtual ground', as in the case of gating circuitry. One more advantage with this scheme is that the latches provide the inputs as well as their complements to drive the AND-plane. The disadvantage with this scheme is that the number of series N devices in the AND-plane is large, with the addition of the evaluation device, thus increasing the delay in the evaluation of the AND-plane. The PLA input latches are made of high gain inverters since they have to drive large capacitances present at the input nodes of the n-transistors in the AND-plane. This is done in order to reduce the sampling time since the output of the second inverter in the latch should provide a logic level suitable to turn the first inverter on or off, by the time the feedback switch closes. This has been verified by SPICE simulation.

The OR-plane uses the PGa version to realize the NAND logic. Outputs of the OR-plane go through Hi-Gain buffers to drive the microsequencer logic. The OR-plane is evaluated in phase1 but some of the controls (phase2F, phase2B, phase2F_L, and phase2B_L) have to be valid until phase2. Therefore these buffers are gated in phase2 to

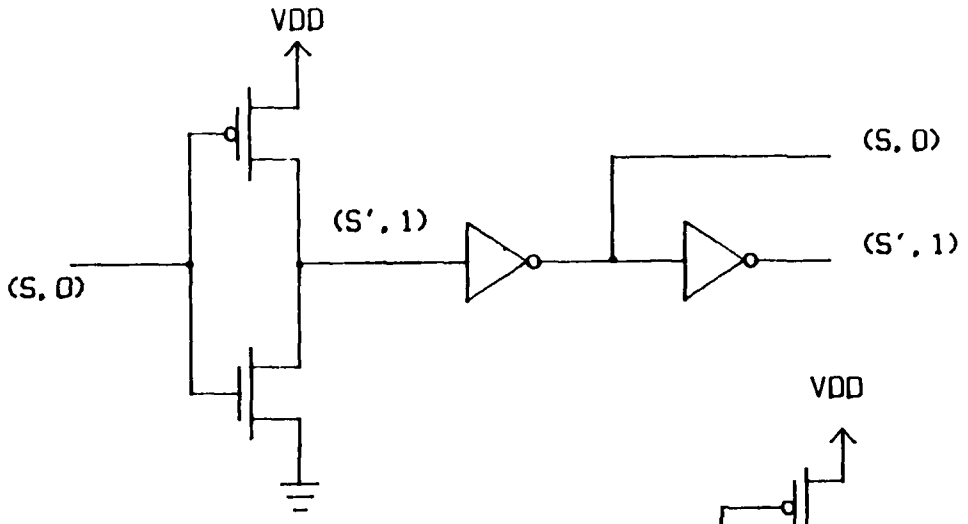


Figure 3.5. Phase1 Buffers for PLA outputs

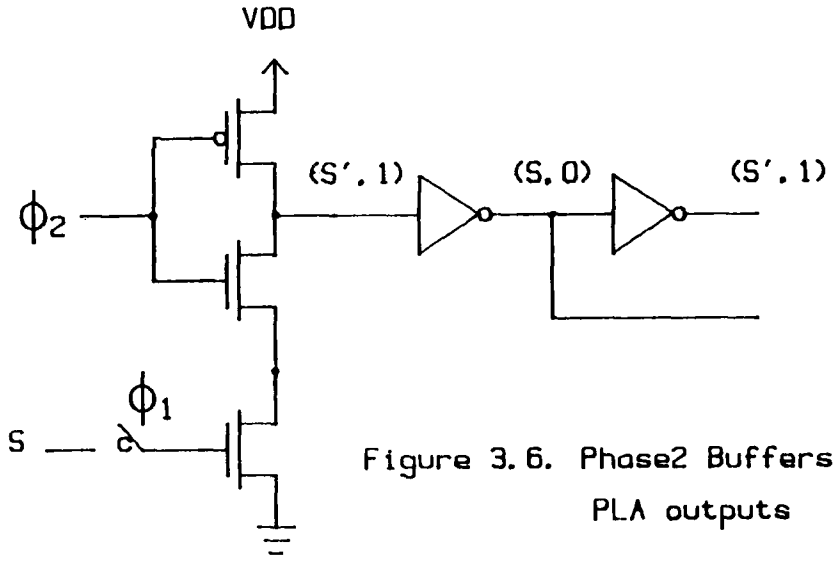


Figure 3.6. Phase2 Buffers for PLA outputs

If the continue intruction executes correctly, the other instructions are tested as follows :

First the instructions CJS, CRTN, CJA, and LOOP should be tested, each with the CONT instruction after the JMAP instruction so that the starting location could be fixed as 10 by loading an op-code of 9 with the JMAP instruction. This makes testing compatible with the explanation of branch instructions in Appendix A where the starting location is always chosen to be 10, except for the JZ instruction. Since CRTN needs a return address it is used in conjunction with CJS. The test code for the above instructions is shown in Table 4.2.

TABLE 4.2.i. TEST CODE FOR CJS AND CRTN.

NAME	BR (phase2)	T (phase1' .phase2')	ADF (phase2)	OC (phase2)	EXPECTED OUTPUT Y (phase1)
JMAP	2	X	0	9	9
CONT	10	X	X	X	10
CONT	10	X	X	X	11
CONT	10	X	X	X	12
CJS	1	0	X	X	13
CJS	1	1	60	X	60
CONT	10	X	X	X	61
CONT	10	X	X	X	62
CONT	10	X	X	X	63
CRTN	6	0	X	X	64
CRTN	6	1	X	X	13

CHAPTER 4

FUNCTIONAL TESTING OF THE MICROSEQUENCER

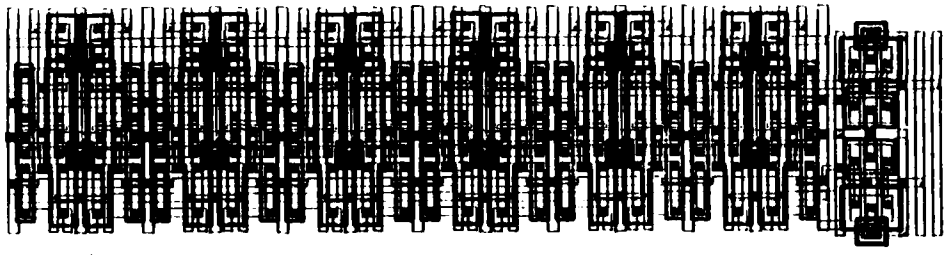
This chapter explains the functional testing of the microsequencer chip. Decimal values have been used in the test codes for convenience. The tables include the expected output (the microprogram address Y), in phase1. Inputs are 4-bit BR field, 5-bit op-code (OC), 12-bit ADF in phase2, and test flag T in phase1'*phase2'. ADF and Y share the same pins with ADF being valid input in phase2 and Y being valid output in phase1.

The eleven branch instructions should be tested, with independent instructions like JZ and JMAP being tested first since these instructions do not depend upon the previous address of the microsequencer. Then the CONT instruction is tested with JZ as well as JMAP. CONT is the most frequently used branch instruction. Table 4.1 shows the test code for the JZ instruction, which includes a zero in the ADF field to clear the LCS, JMAP and CONT instructions. Refer to Appendix A to see the effect of the testing on different blocks of the microsequencer.

TABLE 4.1. TEST CODE FOR CONT WITH JZ AND JMAP

NAME	BR (phase2)	T (phase1' .phase2')	ADF (phase2)	OC (phase2)	EXPECTED OUTPUT Y (phase1)
JZ	0	X	0	X	0
CONT	10	X	X	X	1
JMAP	2	X	X	9	9
CONT	10	X	X	x	10

Figure 3.9.vii. Layout of ADF Latches



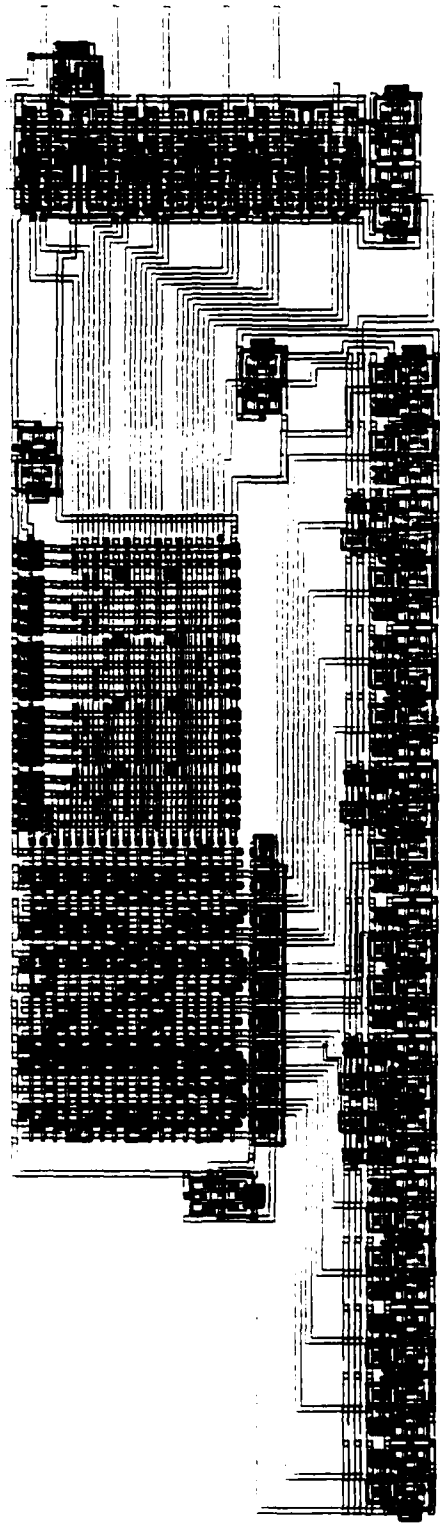
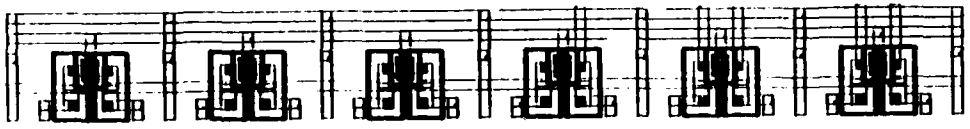


Figure 3.9.vi. Layout of PLA

Figure 3.9.v. Layout of Bus Pre



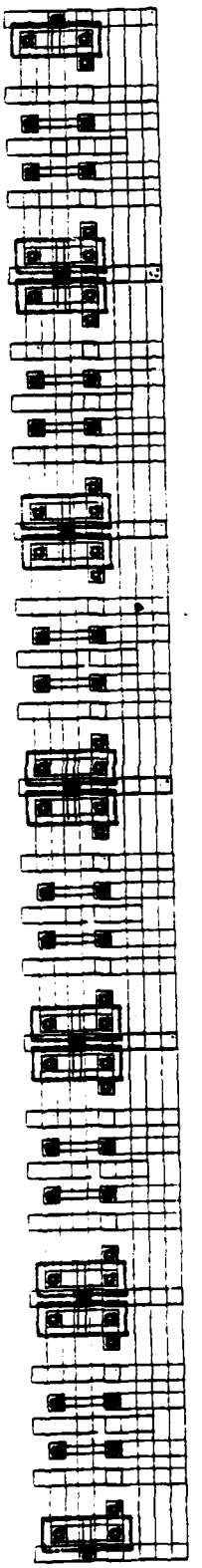


Figure 3.9.iv. Layout of MAP Logic

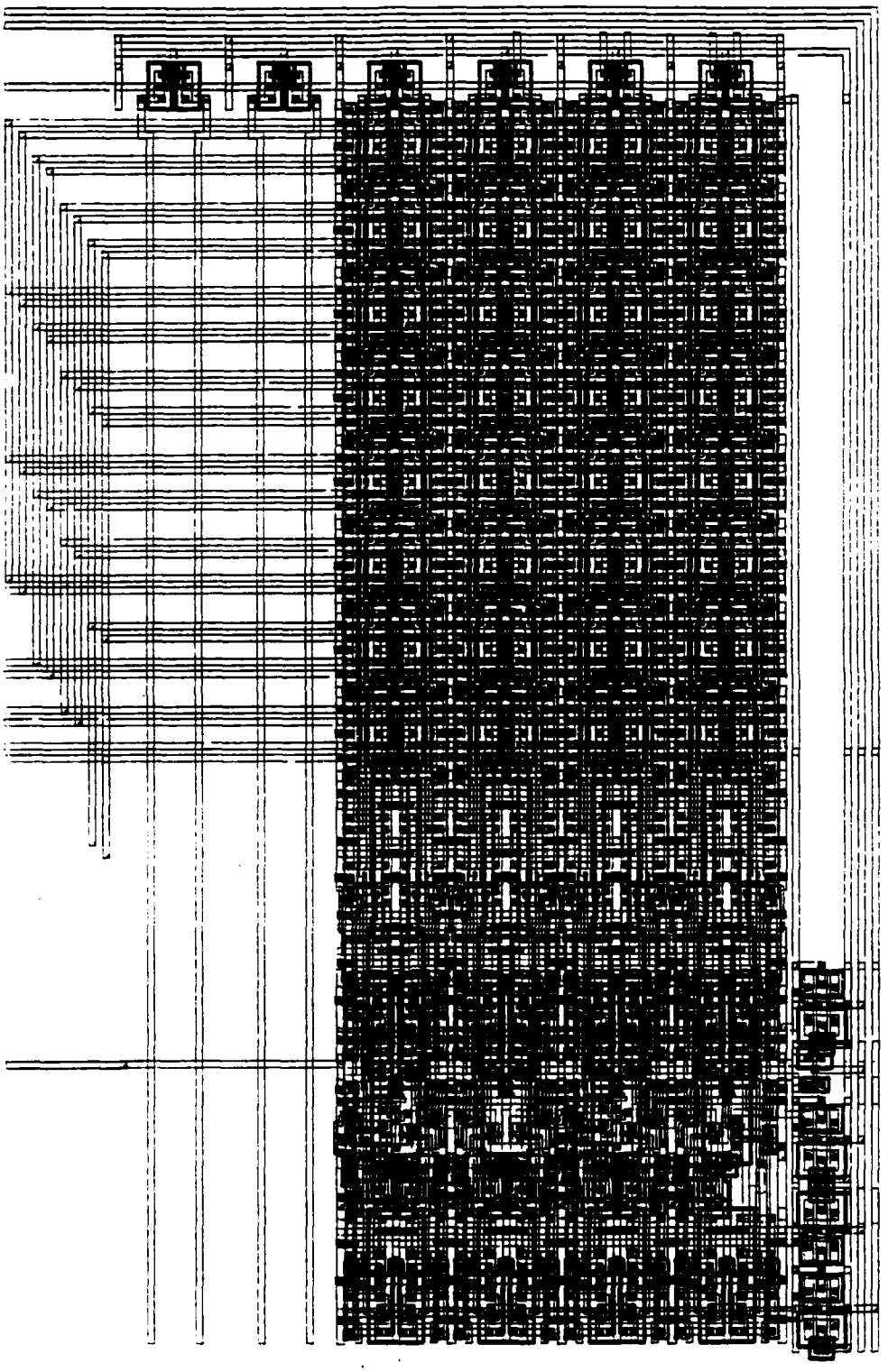


Figure 3.9.111. Layout of Decrementer

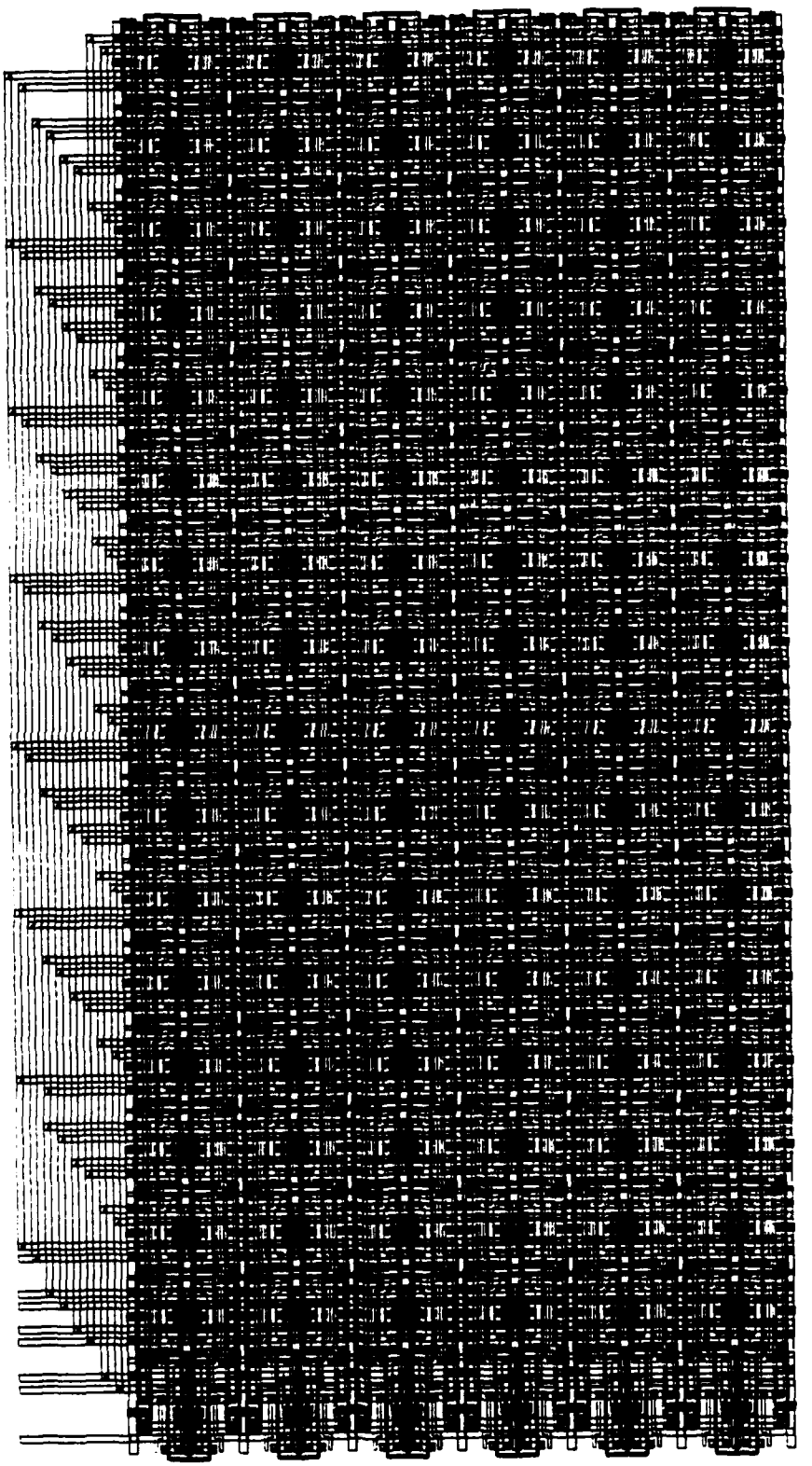


Figura 3.9.11. Layout of Subroutine Stack

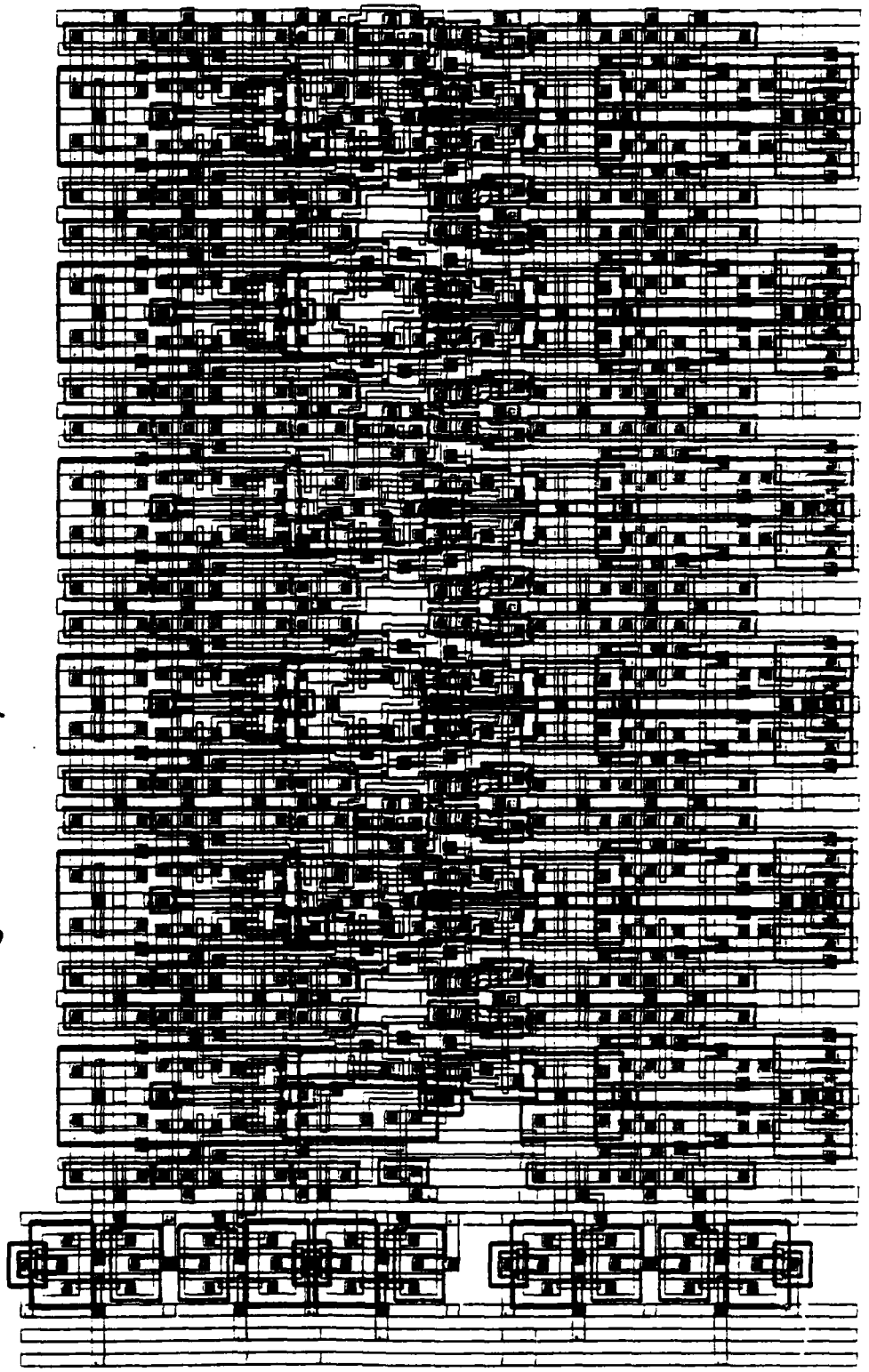


Figura 3.9.i. Layout of Incrementer

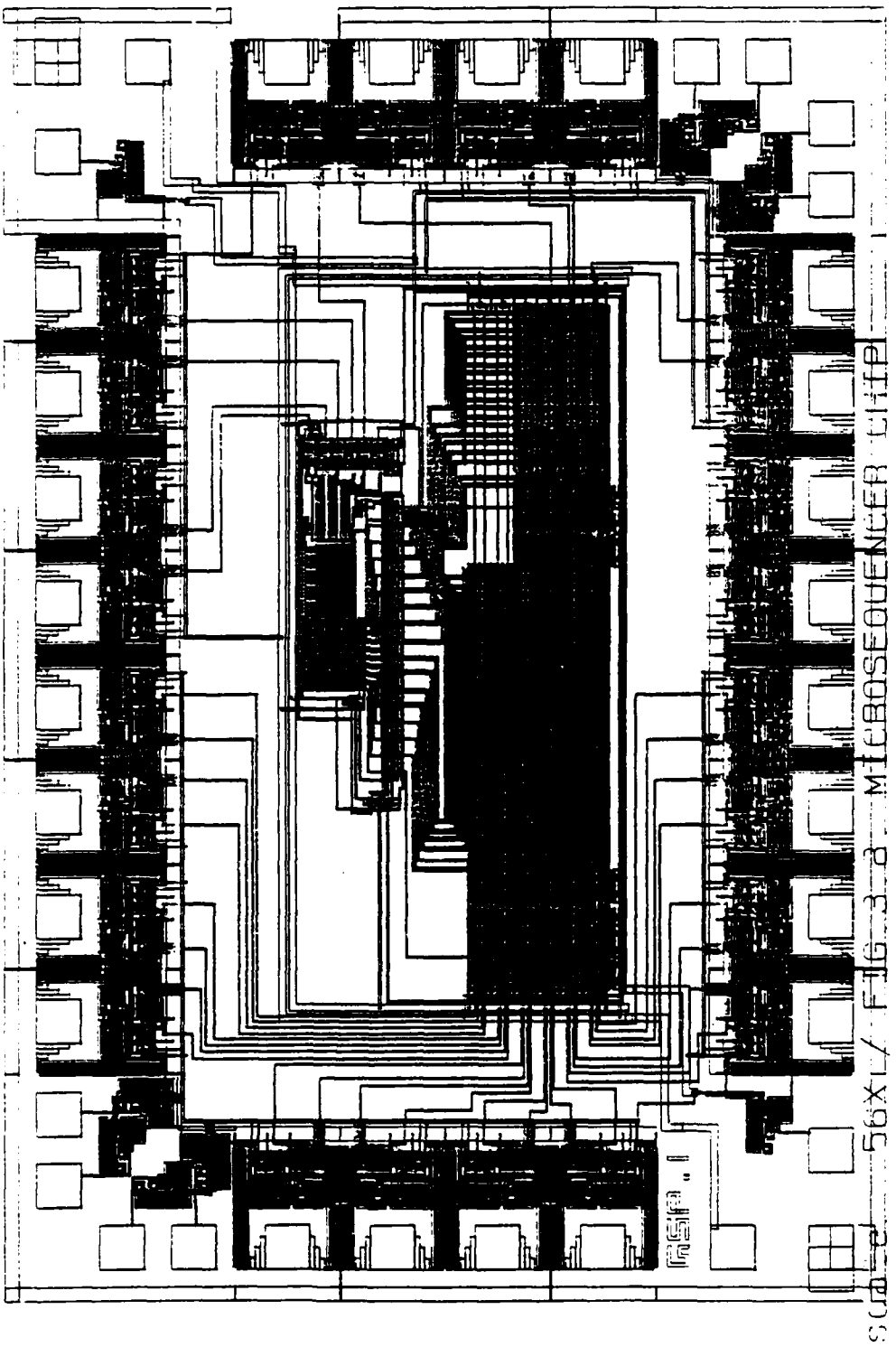


Figure 3.8 Layout of Microsequencer Chip

vertical direction in such a way that they do not interfere with the controls from the PLA. The address bus is precharged using P-gates with a W/L ratio of 4:1 in phase1' and conditionally discharged through N-gates having W/L ratio of 4:1. These ratios were chosen since the bus has a large capacitance.

The final layout of the microsequencer is shown in Figure 3.8 with Figure 3.9 showing the layouts of macros used in the microsequencer.

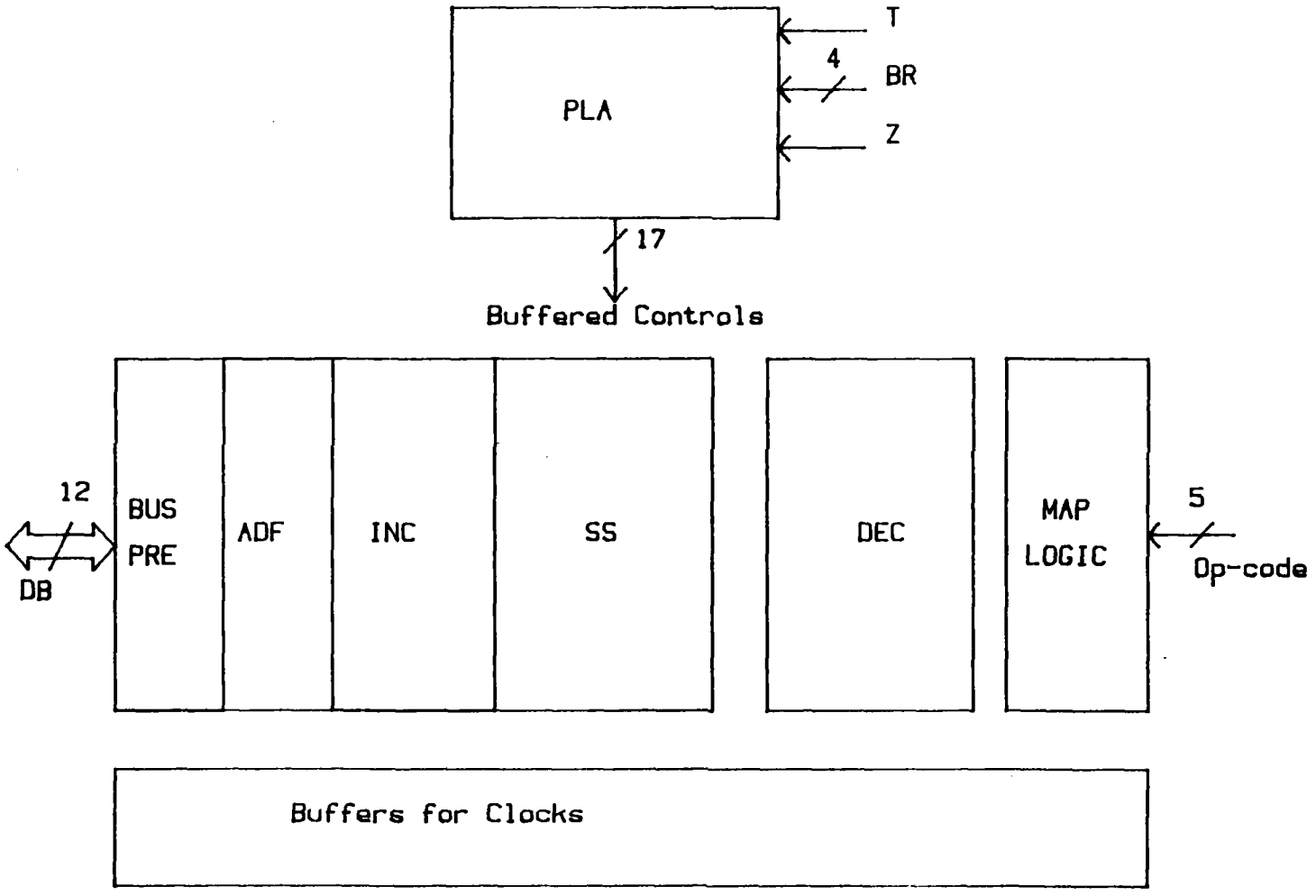


Figure 3.7. Floor plan of Microsequencer

common-drain node is precharged high by a p-type device. The output stays high if the selected address source is high and gets discharged if the selected address source is low. This results in a common source mode of operation with no body-effect.

3.4. FLOOR PLAN OF THE MICROSEQUENCER

Floor plan of the microsequencer is the geometrical and topological arrangements of the microsequencer's macros (also called subsystems). The plan, shown in Figure 3.7, is composed of subsystems built of horizontal slices that are then stacked vertically. The individual macros are then placed horizontally to constitute the final chip. Note that the floor plan shown in the figure does not include the I/O frame, since the I/O frame is a different project [3]. The I/O frame is used by the microsequencer circuit to complete the final chip. For a detailed layout of the microsequencer chip refer to the end of this chapter. Also, the layouts of each macro are included at the end of the chapter. The floor plans of the macros are explained in Appendix B.

The following points were taken into consideration when deciding upon the framework of the plan. Controls from PLA run in first metal [5] in the vertical direction; power supply,ground and data flow in second metal [9] in the horizontal direction. Local interconnections are made out of first metal and poly [5]. Second metal is restricted to global interconnections. Buffers for the clocks, phase1 and phase2, are placed at the bottom to drive the clocks in the

provide the phase2 controls. Refer to Figures 3.5 and 3.6 for phase1 buffers and phase2 buffers (gated in phase2). Since the inputs to the OR-plane are precharged high, a ripple P [7] gate can be used for the OR-plane. Note that the PGa and PGc devices reduce to ripple P type devices if the evaluation devices are removed. Since the input capacitance of the buffer is high, an inverter is added at the output of the OR-plane with devices having a W/L ratio of 2:1. This inverter in turn drives the Hi-Gain buffers with devices having W/L ratios of 6:1. Therefore, with phase1 buffers as shown in Figure 3.5, the OR-plane output ripples through as a phase1 control.

Refer to Figure 3.6 for phase2 buffers where the OR-plane output is stored on the gate capacitance of an n-type transistor during phase1 while the output of the phase2 inverter is precharged high. The inverter in turn drives the Hi-Gain buffer which in effect provides the phase2 controls with complements. These are needed since the controls drive CMOS transmission gates in the microsequencer logic. Note that during phase2 the gated buffer is cut off from the OR-plane, which is evaluated in phase1.

3.3.iv. MULTIPLEXER 1 (MUX1)

Instead of using two controls from the PLA and decoding them into four signals to select one of the four address sources, four select lines are encoded in the PLA to control the MUX1. These lines are precharged low and control the gates of NMOS devices in the multiplexer whose

TABLE 4.2.ii. TEST CODE FOR CJA.

NAME	BR (phase2)	T (phase1' .phase2')	ADF (phase2)	OC (phase2)	EXPECTED OUTPUT Y (phase1)
JMAP	2	X	0	9	9
CONT	10	X	X	X	10
CONT	10	X	X	X	11
CONT	10	X	X	X	12
CJA	7	0	X	X	13
CJA	7	1	60	X	60

The next test should check for instructions PUSH, RFCT, LDCT, and RACT which use the loop counter stack in addition to the subroutine stack. Note that for instruction RFCT to be useful, a count must have been previously loaded by an instruction such as PUSH. Similarly, instruction LDCT must have loaded a count in order for the instruction RACT to be useful. So PUSH is combined with RFCT and LDCT is combined with RACT for testing the above four instructions. The external count for looping is chosen to be 3, which causes the loop to be executed 4 times. Table 4.3 shows the test code for testing the above four instructions.

If these tests, with the JMAP instruction as the starting instruction, fail, then the starting instruction should be JZ followed by CONT instruction. This will start the sequence from location 1 instead of 10 as in the case of starting with the JMAP instruction.

Table 4.2 is divided into three parts: one showing the test code for CJS and CRTN (Table 4.2.i), second with the test code for CJA (Table 4.2.ii) and the third showing

the code for LOOP. Similarly Table 4.3 is divided into two parts: one with the test code for LDCT and RACT (Table 4.3.i) and the other showing the code for PUSH and RFCT (Table 4.3.ii).

TABLE 4.2.iii. TEST CODE FOR PUSH AND LOOP.

NAME	BR (phase2)	T (phase1' .phase2')	ADF (phase2)	OC (phase2)	EXPECTED OUTPUT Y (phase1)
JMAP	2	X	0	9	9
CONT	10	X	X	X	10
CONT	10	X	X	X	11
PUSH	3	1	X	X	12
CONT	10	X	X	X	13
CONT	10	X	X	X	14
CONT	10	X	X	X	15
CONT	10	X	X	X	16
LOOP	8	0	X	X	12
LOOP	8	1	X	X	17

TABLE 4.3.i. TEST CODE FOR LDCT AND RACT.

NAME	BR (phase2)	T (phase1' .phase2')	ADF (phase2)	OC (phase2)	EXPECTED OUTPUT Y (phase1)
JMAP	2	X	0	9	9
CONT	10	X	X	X	10
CONT	10	X	X	X	11
LDCT	9	X	3	X	12
RACT	5	X	12	X	12
RACT	5	1	12	X	12
RACT	5	X	12	X	12
RACT	5	X	12	X	12
RACT	5	X	X	X	13

TABLE 4.3.ii. TEST CODE FOR PUSH AND RFCT.

NAME	BR (phase2)	T (phase1' .phase2')	ADF (phase2)	OC (phase2)	EXPECTED OUTPUT Y (phase1)
JZ	0	X	0	X	0
JMAP	2	X	0	9	9
CONT	10	X	X	X	10
PUSH	3	1	2	X	11
CONT	10	X	X	X	12
CONT	10	X	X	X	13
CONT	10	0	X	X	14
RFCT	4	X	X	X	11
CONT	10	X	X	X	12
CONT	10	X	X	X	13
CONT	10	0	X	X	14
RFCT	4	X	X	X	11
CONT	10	X	X	X	12
CONT	10	X	X	X	13
CONT	10	0	X	X	14
RFCT	4	X	X	X	11
CRTN	6	1	X	X	0

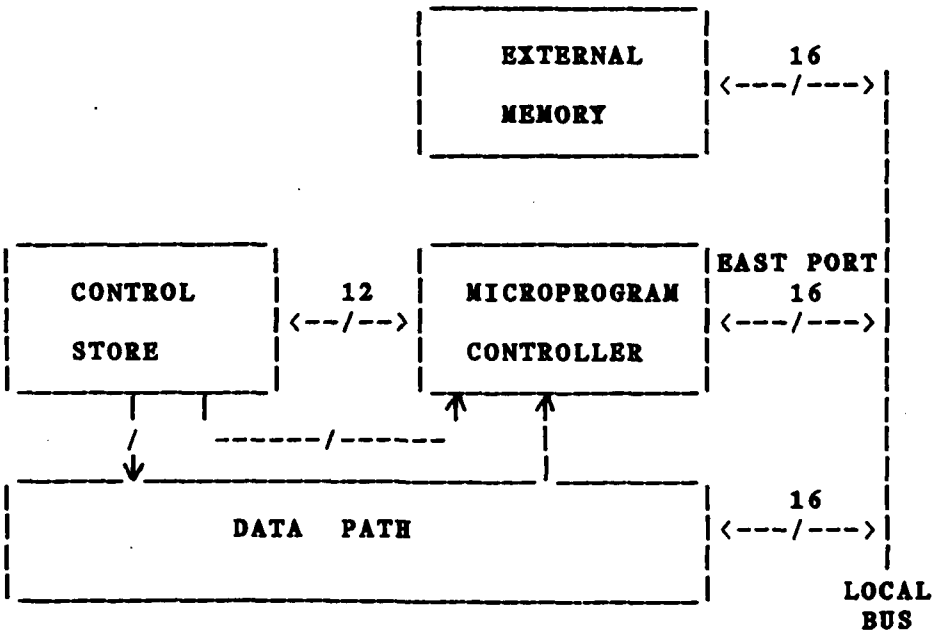
Note that the PUSH instruction is also used in conjunction with the LOOP instuction as shown in Table 4.2.iii. When RFCT exits the loop it pops the loop counter stack as well as the subroutine stack. Prior to a PUSH with the RFCT instruction, the SS and LCS are cleared with the JZ instruction. Therefore, a CRTN is executed unconditionally (by forcing the T input to be 1) at the end of the code to check if the output becomes zero.

CHAPTER 5

A NEXT GENERATION MICROCONTROLLER

The microsequencer chip discussed so far was an experimental version of the microsequencer for a next generation controller which will be explained in this chapter using a system example. This example will illustrate how the controller is integrated into the system. Figure 5.1 shows a system example consisting of four chips :
the microcontroller chip, the data path chip, the control store, and an external read/write (RAM) memory. Suggestions on how to modify the microsequencer to assure compatibilty with the controller are included.

Figure 5.1 SYSTEM EXAMPLE



The main functions of the microcontroller are :

(1) to generate an address for the control store in phase1. This address is generated by the microsequencer.

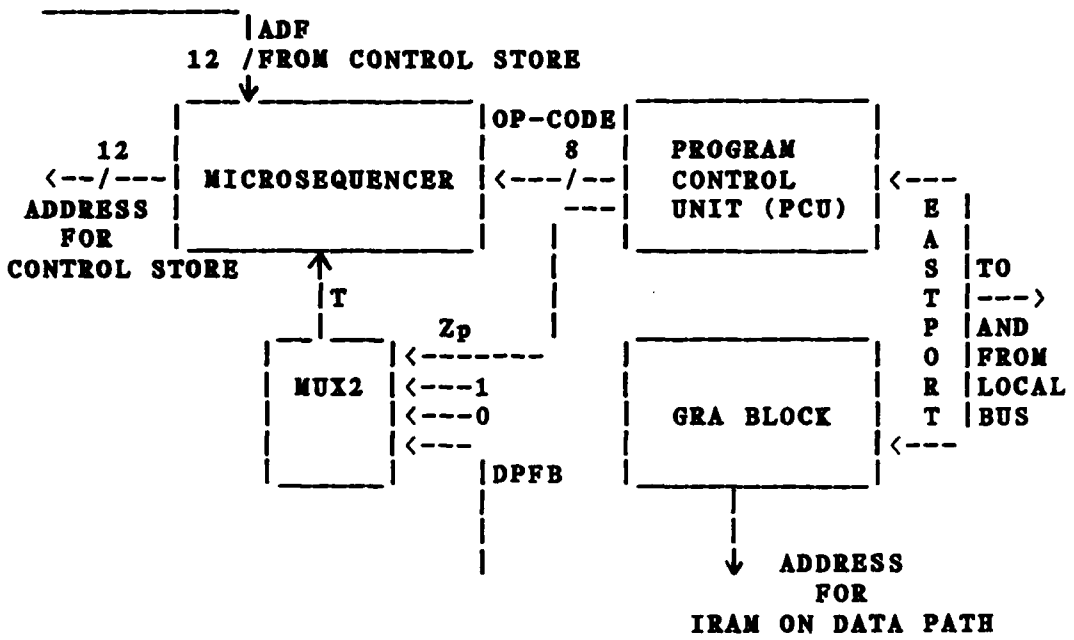
(2) to generate an address for external RAM in phase2. This address is generated by the program control unit (PCU).

(3) to generate an address for the internal RAM, which is on data path chip. This address is generated during phase1 by the Generate Random Address (GRA) Block.

The microcontroller generates the microprogram address in phase1 which fetches the microword from the microstore. In the previous chapters it was mentioned that the microword contained the micro-ops field, to control the data path, and the next address generation field to control the microsequencer. The microword must also provide microbits for controlling the PCU and GRA Block, in addition to the microbits for controlling the microsequencer and data path. It is better to discuss the PCU and GRA Block before deciding on the changes to be made to the present microsequencer to ensure compatibilty with the next generation microcontroller. Refer to Figure 5.2 for a functional description of the microcontroller.

The PCU and GRA Block are not explained in detail since they are part of the next generation controller; so the following discussion deals only with the interfaces between the microsequencer and PCU, and the microsequencer and GRA Block. The program control unit provides addresses to fetch instructions from the external memory and transfer

Figure 5.2. MICROPROGRAM CONTROLLER



blocks of data from the external memory to the internal RAM on the data path. When the PCU fetches an instruction, it transfers the op-code to a register (Instruction Register) which then becomes the input to the MAP Logic in the microsequencer.

The PCU must be provided with the number of data words to be transferred from the external RAM to internal RAM. This is specified as a byte count (BC) which is loaded into a register (BC register) from the external memory during op-code fetch. The instruction also specifies the starting address for the block transfer. The BC is decremented and the block of data is transferred as long as the count is not equal to 0. A zero detector checks for all zeros in the count and the output of the ZD is denoted by Zp (the subscript p stands for PCU, to avoid confusion with the

ZD output, Z, in the DEC macro of the microsequencer). Zp is one of the inputs to a multiplexer (MUX2) which selects between a '1', '0', DPFB (data path flag bit), and Zp. Zp is '1' if BC=0. The MUX2 was part of the data path in the initial version and the output of MUX2 was called the test flag 'T', which is one of the inputs to the PLA in the experimental version of the microsequencer. This 'T' now becomes the DPFB and the output of MUX2 is still denoted by 'T', which will be one of the inputs to the PLA in the microsequencer. The 4-bit condition field (CD) of the experimental version is now split into two fields : (i) a 2-bit MUXSEL field which controls MUX2 and (ii) a 3-bit Flag Bit field which selects one of the status bits on the data path chip to become the DPFB and to be transferred to MUX2 on the microcontroller chip.

There is no interface defined at this point between the microsequencer and the GRA Block; however the GRA Block will be part of the microcontroller chip because there is not sufficient area for implementation of the GRA Block on the data path chip. Another reason is that the GRA Block receives input data from the instructions fetched by the PCU from the external memory. Data is provided to the PCU and the GRA Block through a 16 bit I/O port, to be called east port, which is connected to the local bus as shown in Figure 5.1. The GRA Block and PCU can be controlled by individual PLAs similar to the microsequencer. Inclusion of the GRA Block on the controller chip increases the pin count of the

chip since the GRA Block must provide the address for the internal RAM on the data path. However, these lines could also be used to send status flags from the data path to controller MUX2 when the GRA Block is not sending an address. This will eliminate the need for a multiplexer in the data path; therefore the 3-bit Flag Bit field can be removed. The MUXSEL field can be increased to 4 bits to select between 1, 0, Zp, and one of the flags from the data path. (The MUXSEL field will be 3-bits wide if the number of flags from the data path is less than or equal to 5).

The microcontroller is still under development and the preceding discussions suggest possible modifications to be considered to enhance the overall operation of the system.

APPENDIX A

FUNCTIONAL DESCRIPTION OF THE MICROSEQUENCER

A.1. DISTINCTIVE CHARACTERISTICS

*TWELVE-BIT-WIDE ADDRESS

Can address up to 4096 words of microcode.

*INTERNAL DECREMENTER WITH LOOP COUNTER STACK

Pre-settable 8-bit down counter for repeating instructions and counting loop operations. Stack is used over a single loop counter so that nested loops and loops within subroutines could both be accomodated.The loop counter stack is 4-words deep.

*FOUR ADDRESS SOURCES

Microprogram address may be selected from mapped address, ADF field of the microinstruction, incremented address (sequential), or 8-level push-down (subroutine) stack which stores the return addresses.

*ELEVEN POWERFUL SEQUENCE CONTROL INSTRUCTIONS

Four bit BR field of the microinstruction is used to execute eleven sequence control instructions, most of which are conditional on the test flag 'T' or the zero detector (ZD) output 'Z'.

*MULTIPLEXER 1

The multiplexer (MUX1) for selecting one of the four address sources is implemented as a distributed bus structure.

*TWO PHASE NON-OVERLAPPING CLOCKS

All internal operations of the chip are controlled by two non-overlapping clocks, phase1 and phase2.

*CLOCKED CMOS USED FOR LAYOUT DENSITY AND PERFORMANCE

Clocked CMOS is widely used over classic CMOS for higher layout density and better performance. The microsequencer logic is precharged during phase2 and evaluated during phase1, except for the PLA which is precharged during phase1' and evaluated in phase1.

A.2. GENERAL DESCRIPTION

A microprogrammed control unit should be viewed as consisting of two parts: the control memory that stores the microinstructions and the associated circuitry that control the generation of the next address. The address-generation part is called the microsequencer. The control store contains the binary variables that control the hardware components of the system. The control information is stored in the memory and executed as a stored program. Hence the term firmware is sometimes used for microprogramming since it has a software as well as a hardware flavor.

The microsequencer is also called a microprogram sequencer since it is an address sequencer intended for controlling the sequence of execution of the microinstructions stored in the microprogram memory (also called the microstore or control store). Refer to Figure 1.3 of chapter 1 for the functional (block) diagram of the microsequencer, which is repeated here for convenience. Besides the capability of sequential access, it provides

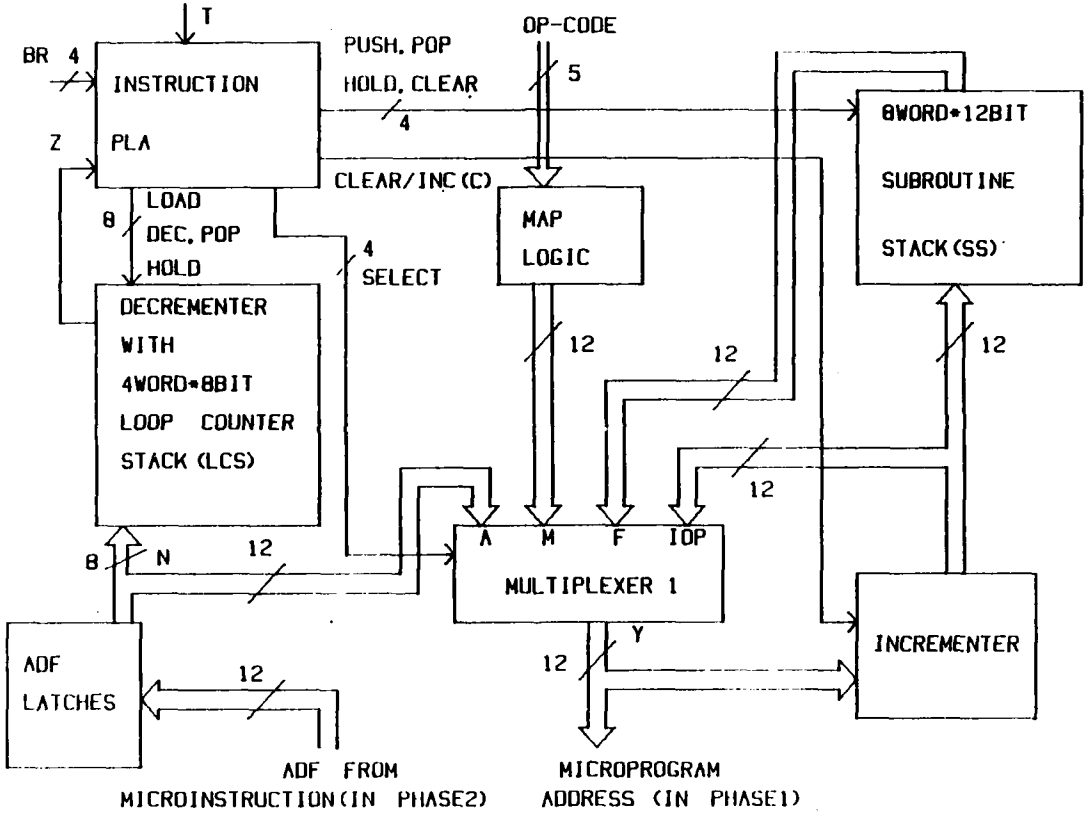


Figure 1.3 BLOCK DIAGRAM OF MICROSEQUENCER

conditional branching to any microinstruction within a 4K word range. A push-down stack is used for microsubroutine return linkage and looping capability. The subroutine stack (SS) has 8 registers which provide 8 levels of nesting subroutines. Microinstruction loop count control is provided with a count capacity of 256. The 12-bit ADF field of the microinstruction is used as an explicit address source for microprogram address or the 8 least significant bits are used as the count (N) for looping. A loop counter stack, which is 4-words deep, is used over a single loop counter register in order to nest loops and to use loops within subroutines.

During each phase1, the microsequencer provides a 12-bit address from one of the four sources:

(1) The incremented output (IOP), which usually contains an address one greater than the previous address,

(2) ADF field of the microinstruction (A) which is an explicit address,

(3) Eight-deep last-in first-out subroutine stack (F), or

(4) Mapped output (M), which is a function of the 5-bit op-code part of the machine instruction (also called macro-operation). A mapping PROM or PLA could be used to map the 5-bit op-code into a 12-bit address source. The scheme used here places zeros in the 7 most significant bits and transfers the 5-bit op-code into the 5 least significant bits of the address.

The instruction PLA controls all other blocks of the microsequencer. The inputs to the PLA are the 4-bit BR field from the microinstruction, test flag 'T' from the data path, and the zero detector (ZD) output 'Z'. Note that DEC stands for decrementer which contains : (i) a latch (DEC I/P LATCH), serving the purpose of a loop counter register, (ii) loop counter stack (LCS), (iii) decrementer logic (DEC LOGIC) consisting of carry chain (DEC CAR CHAIN) and EX-OR circuits, (iv) a latch (DEC O/P LATCH) which latches the DEC LOGIC output (D) and makes available a stable decremented output (DOP) by the time the decrement control (LD) is active, and (v) a multiplexer (DEC MUX) to select between the external count N , the decremented output DOP and the second word (LCS2) of the loop counter stack which contains the previously pushed value. When an external count (N) is loaded into the DEC it goes into the DEC I/P LATCH (loop counter register), which will be denoted as LCR, and also pushed onto the loop counter stack. The value in the loop counter register (LCR) is unconditionally decremented and loaded back into the decrementer (DEC), for further counting only if the decrement control (LD) is high. The ZD checks for all zeros in the loop counter before the value is decremented. Therefore a count (N-1) should be loaded for looping N times. The external count is loaded in phase1, the DEC LOGIC is evaluated in phase2 and the DOP becomes available in the next phase1. For a functional diagram of

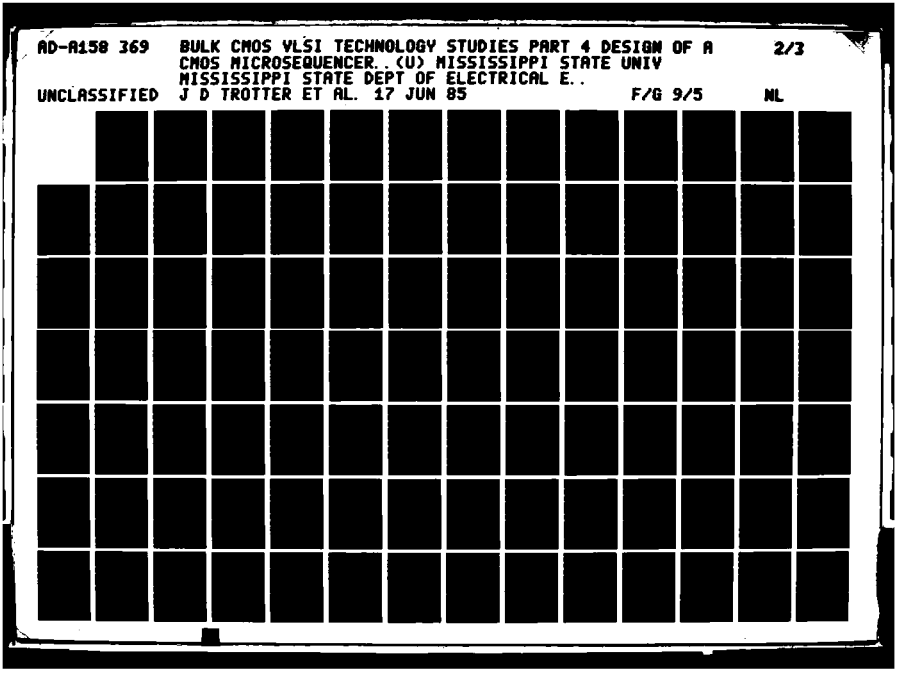
the decrementer refer to Figure B.3.a of Appendix B which is repeated here for convenience.

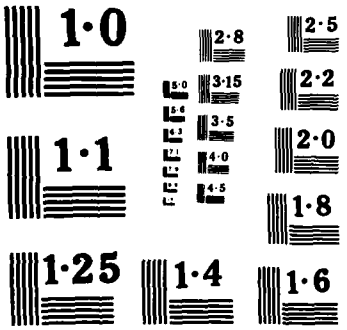
The 12-bit address bus of the microsequencer is bi-directional. It is used for the microprogram address in phase1 and the ADF field of the microinstruction in phase2. This scheme is used to reduce the pin count of the chip. Note that the ADF field is latched in phase2 and becomes a stable address source (A) in phase1. The instruction set is explained below, with execution examples in Figure A.1.

A.3. INSTRUCTION SET

There are eleven instructions which select the address of the next microinstruction to be executed. Four of the instructions (0, 2, 9, 10) are unconditional - their effect depends only on the instruction. Two instructions (4, 5) depend on the zero detector output 'Z' and five instructions (1, 3, 6, 7, 8) depend on the test flag 'T'. The best way to explain the instruction set is to isolate each instruction and review its operation. Figure A.1 shows the execution examples of the eleven instructions. Note that there can be 16 sequence control instructions from the 4-bit BR field but it has been decided to include only the 11 considered most useful at this time.

The examples in Figure A.1 illustrate the microprogram flow as various microinstructions are executed. As an example, the CONTINUE instruction (instruction number 10) simply means that microinstruction 10 is executed, then microinstruction 11 is executed and so on. The microprogram


AD-A158 369
BULK CMOS VLSI TECHNOLOGY STUDIES PART 4 DESIGN OF A CMOS MICROSEQUENCER..(U) MISSISSIPPI STATE UNIV MISSISSIPPI STATE DEPT OF ELECTRICAL E..
2/3
UNCLASSIFIED
J D TROTTER ET AL. 17 JUN 85
F/G 9/5
NL



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

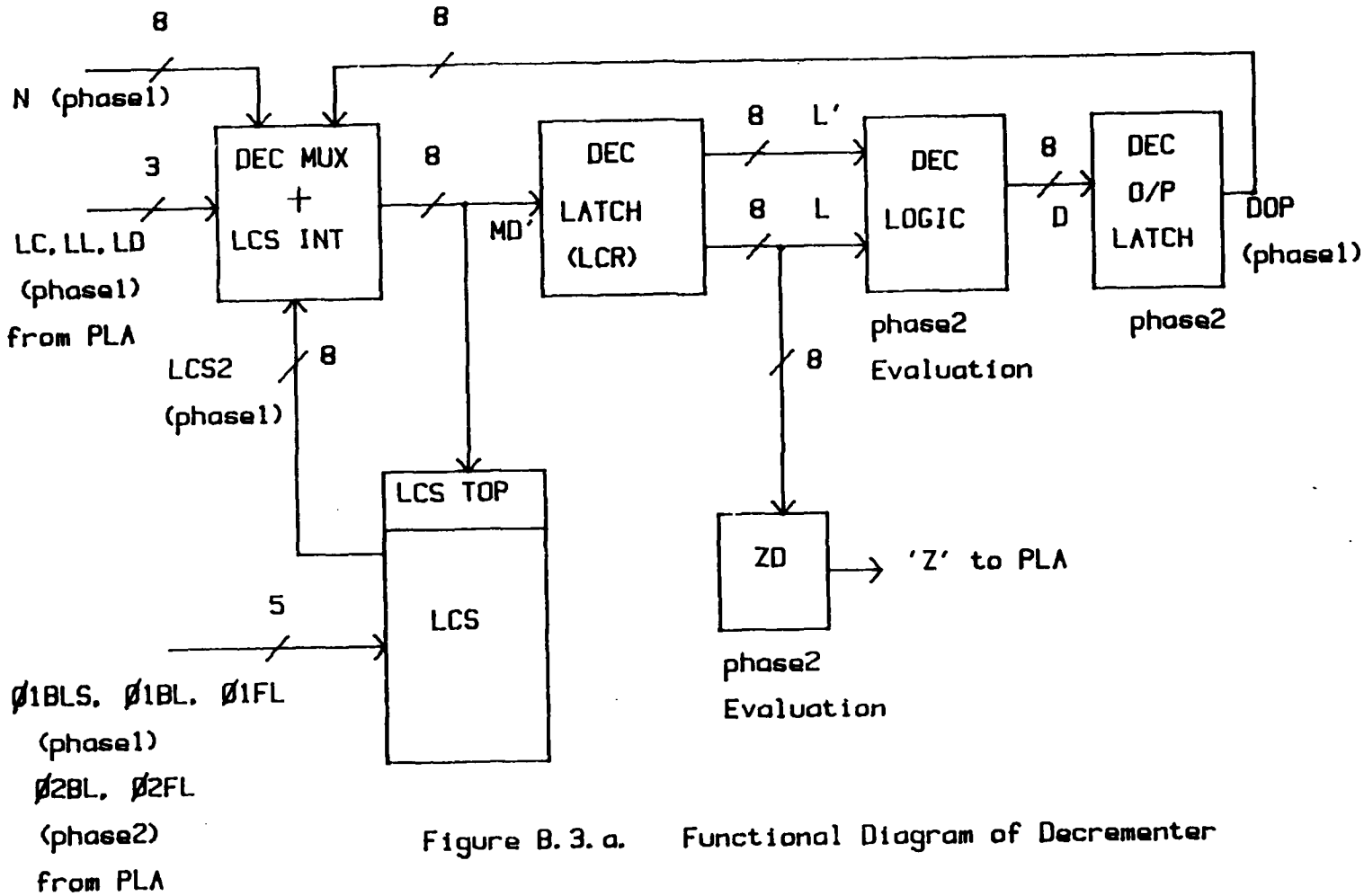


Figure B.3.a. Functional Diagram of Decrementer

addresses were arbitrarily chosen except for the JUMP ZERO instruction which forces the address to zero. Note that a conditional instruction can be executed unconditionally by forcing the T flag to be '1' by selecting the '1' input to the second multiplexer (MUX2), where MUX2 generates the test flag in the data path.

(0) JZ : Jump Zero unconditionally specifies that the address of the next microinstruction is zero.

(1) CJS : Conditional Jump-to-Subroutine via the address provided in the ADF field. This can also be labeled as the 'CALL' instruction. When the contents of address 12 is at the output of the control store, the next address control function is the Conditional Jump-to-Subroutine (CJS). If 'T' is true , CJS is executed by selecting location 60 as the next address with 13 pushed onto the subroutine stack (SS). If the test fails ('T' false) the next address in sequence i.e., address 13 is selected.

(2) JMAP : Jump Map instruction is unconditional which causes the output of the map logic to be executed.

(3) PUSH Push / Conditional Load DEC. This is primarily used for setting up loops in microprogram firmware. The output of the loop counter register (LCR) is made available to the ZD as well as the DEC LOGIC. In Figure A.1 when microinstruction at location 12 is executed, a PUSH will be made onto the SS and the decrementer will be loaded based on the condition. When the PUSH occurs, the value pushed onto the SS is always the next (sequential) address. In this case

this address will be 13. If the test fails, the decrementer (DEC) is not loaded; if it is passed the DEC is loaded with the value contained in the ADF field of the microinstruction. Thus a single instruction can be used to set up a loop to be executed a specific number of times. Instruction 4 (RFCT) will describe how to use the pushed value and the decrementer for looping.

(4) RFCT : Repeat Loop, LCR not equal to 0. For this instruction to be useful, some instruction such as 3 (PUSH) must have loaded a count value into the DEC. This instruction checks to see whether the LCR is equal to 0 or not. This is accomplished by the zero detector (ZD) which is implemented as a NOR gate. The ZD output 'Z' is '1' if the LCR is equal to zero and '0' if the LCR has a non-zero value. If the ZD output is '1', the DOP is fedback into the DEC for further decrementation and the address of the next microinstruciton is taken from the top of the SS. If the ZD output is '0' (the loop exit condition), control falls through the next (sequential) microinstruction by selecting incremented output (IOP); the SS is popped and the LCS is also popped. In Figure A.1 location 10 most likely would contain a PUSH instruction which would have caused address 11 to be pushed onto the SS and the DEC to be loaded with the proper value (N) for looping the desired number of times. RFCT can also be referred to as a 'FOR' loop.

(5) RACT : Repeat ADF, LCR not equal to 0. This instruction is similar to instruction '4' except that the branch address

now comes from the ADF field of the microinstruction rather than from the SS. When the LCR is not equal to 0 (i.e., ZD output is '0'), the DOP is loaded back into the DEC for further decrementation and the next address selected is the ADF field. If the ZD output is '1', the count is popped off the LCS; a pop is not performed on the SS, because it is not used. In Figure A.1, the RACT instruction is shown in a single microinstruction loop as in instruction 12. The address in the ADF field of the microinstruction would be 12. Instruction 11 in this example could be the LOAD DEC AND CONTINUE (LDCT:9) instruction. While this example shows a single microinstruction loop, by simply changing the address in the ADF field of the microinstruction, multi-instruction loops can be performed in this manner for a fixed number of times as determined by the count (N).

(6) CRTN : Conditional Return from Subroutine instruction is used to branch from the subroutine back to the next microinstruction address following the subroutine call. Since this microinstruction is conditional, the return is performed only if the test is passed. If the test fails, the next (sequential) microinstruction is executed. To perform an unconditional Return-from-Subroutine, the conditional Return-from-Subroutine is executed unconditionally. This is done by forcing the 'T' flag to be '1' by selecting the '1' input to MUX2, which is in the data path.

(7) CJA : Conditional Jump ADF instruction derives the branch address from the ADF field of the microinstruction.

It provides a technique for branching to various microprogram sequences depending upon the test flag from the data path.

(8) LOOP : Test End-of-Loop instruction provides the capability of conditionally exiting a loop at the bottom; i.e., this is a conditional instruction that will cause the microprogram to loop, via the SS, if the test is failed else to continue the next (sequential) instruction. In Figure A.1, the LOOP instruction is at address 16. If the test fails, the microprogram will branch to address 12. Address 12 is on the SS because a PUSH instruction has been executed at address 11. If the test is passed at instruction 16, the loop is terminated and the next (sequential) microinstruction at address 17 is executed, which also causes the SS to be popped, thus accomplishing the required stack maintenance. A 'LOOP' instruction can also be referred to as a 'DO' instruction.

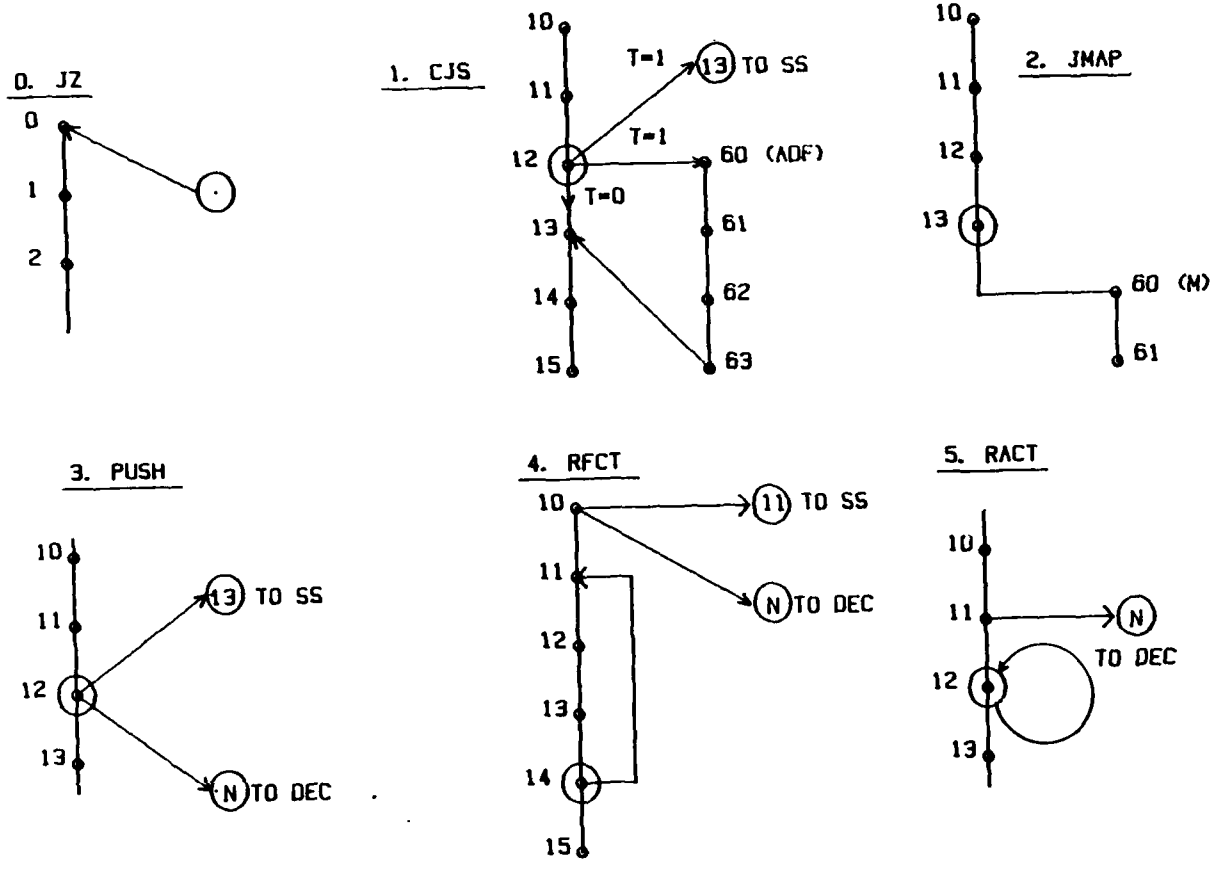
(9) LDCT : Load DEC and Continue instruction enables the DEC to be loaded with the value at its parallel inputs. These inputs are normally connected to the 8 least significant inputs of the ADF field of the microinstruction which (in the architecture described here) serves to supply either a branch address (to MUX1) or a COUNT ('N' for the DEC) depending upon the microinstruction being executed. There are 2 ways of loading the count (N) into the DEC

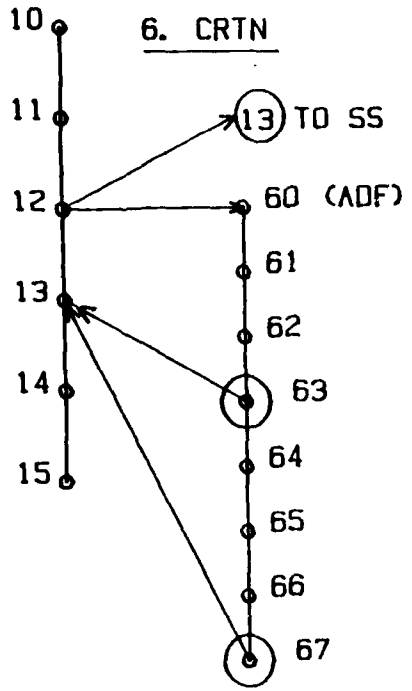
- the explicit load by this instruction :9

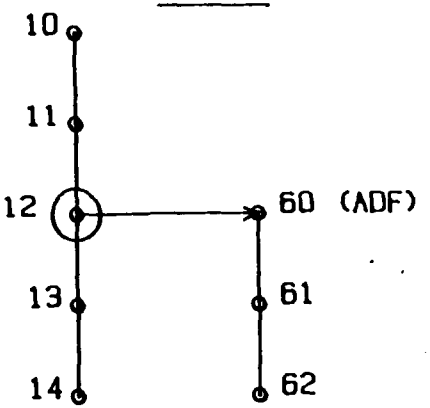
- the conditional load included as part of the PUSH instruction :3.

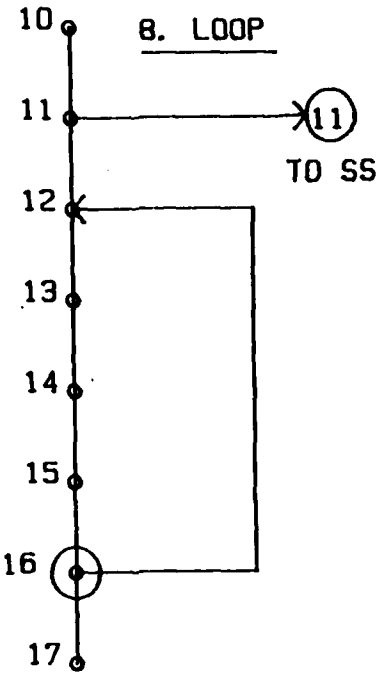
(10) CONTINUE : This simply increments the address so that the next (sequential) microinstruction is executed. This is the simplest instruction and should be the default instruction which the firmware requests whenever no other operation is enabled.

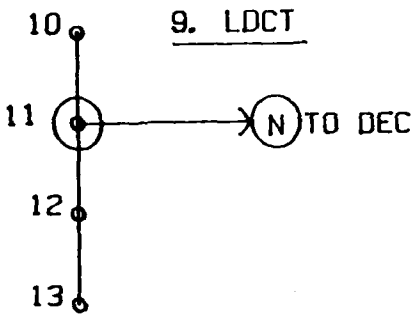
Figure A.1. EXECUTION EXAMPLES OF BRANCH INSTRUCTIONS

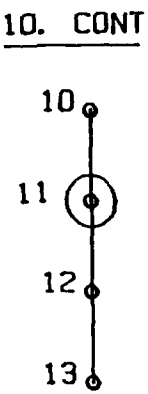



10
6. CRTN
11
13 TO SS
12
60 (ADF)
61
13
62
14
63
64
15
65
66
67


7. CJA
10
11
12
60 (ADF)
13
61
14
62


10
8. LOOP
11
11
TO SS
12
13
14
15
16
17


10
9. LDCT
11
N TO DEC
12
13


10. CONT
10
11
12
13

APPENDIX B

LOGIC DESCRIPTIONS AND CIRCUIT SCHEMATICS OF MACROS

B.1. INCREMENTER

The incrementer output (IOP) is one of the four address sources for the microprogram address. It is also the input to the subroutine stack (SS). The previous output of the microsequencer (Y) is the input to the incrementer and it is sampled during time phase1. The incrementer logic (INC LOGIC) is precharged during phase2' and evaluated during phase2. The output of the INC LOGIC is sampled in phase2 and stable in the next phase1. The INC LOGIC consists of incrementer carry chain (INC CAR CHAIN) and Exclusive-Or (EX-OR). The Clear/Increment (C) control from PLA, when high, pulls the IOP low. When C is low, the incremented output is available. For a detailed layout refer to the cell library in Appendix C. The functional diagram, logic diagram, circuit schematic and the floor plan of the incrementer are shown in Figure B.1.

(a) FUNCTIONAL (BLOCK) DIAGRAM

Refer to Figure B.1.a.

(b) LOGIC DIAGRAM

Refer to Figure B.1.b.

From the figure it is clear that the carry chain is precharged high in phase2' and evaluated in phase2. Since the input to the incrementer input latch (INC I/P LATCH) is sampled in phase1, a phase2 evaluation device is added to

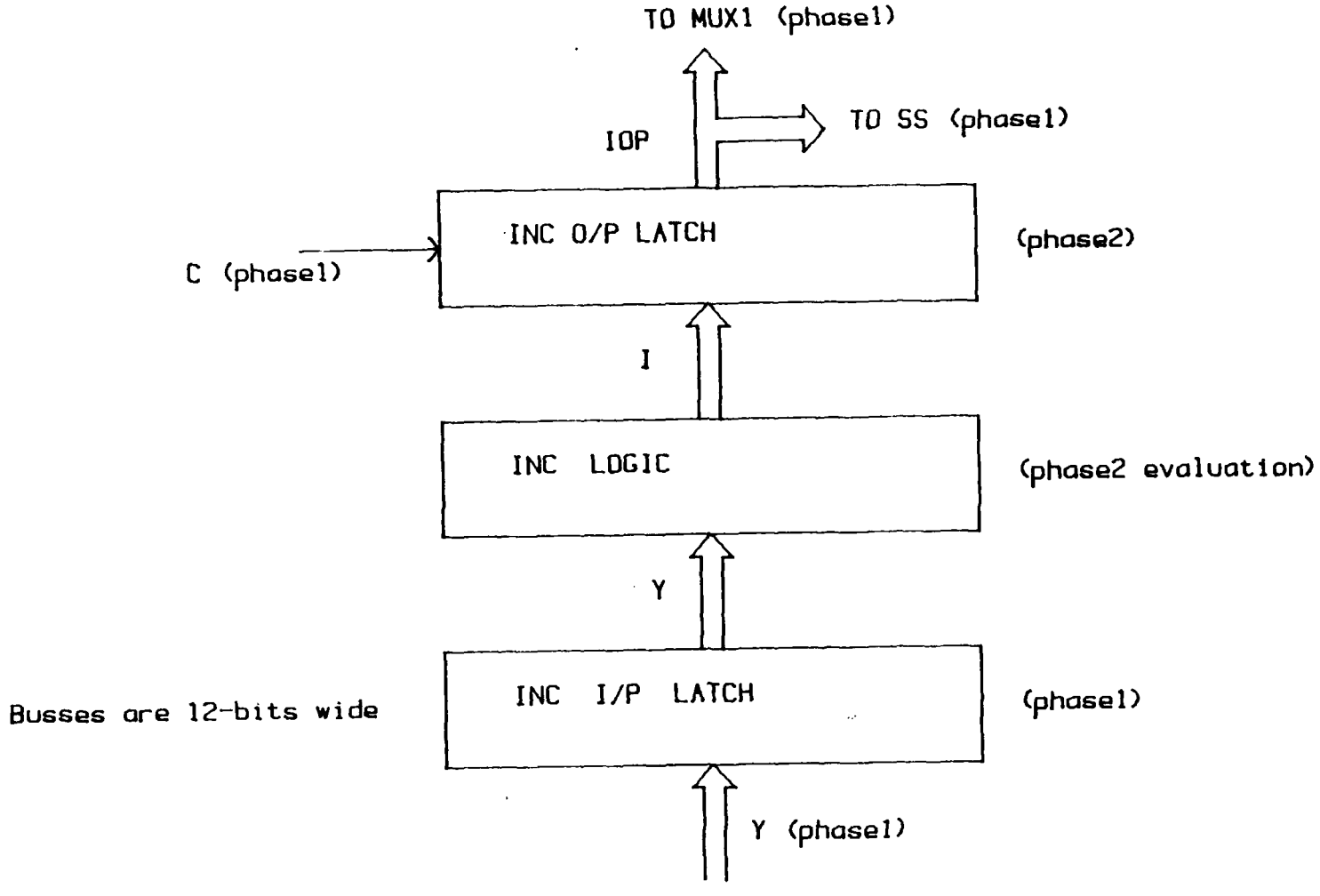


Figure 8.1.a. Block Diagram of Incrementer

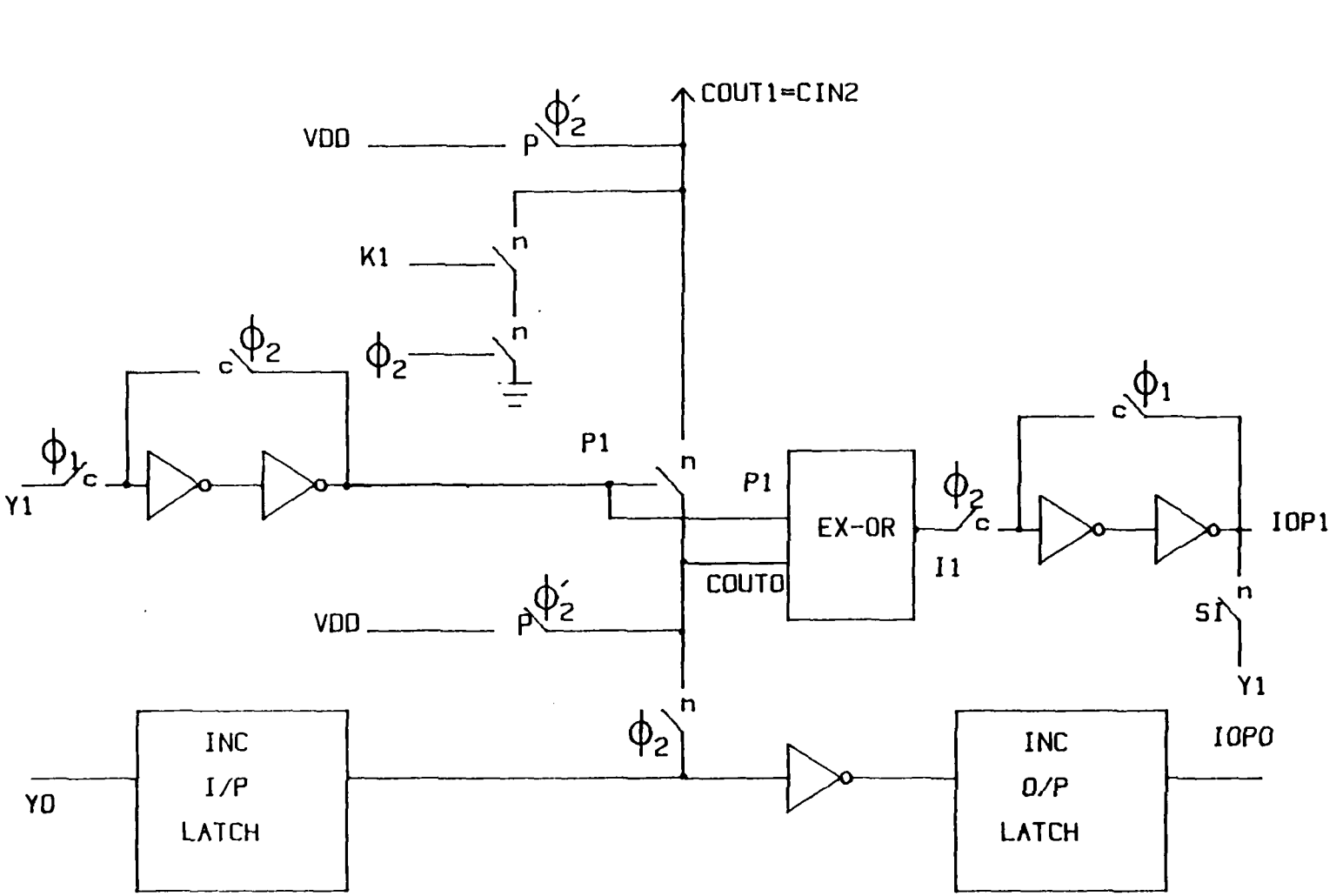


Figure B.1.b. Logic Diagram of Incrementer

the kill transistor, as shown in the figure, to make sure that the kill transistor is not on during the precharge time phase2' which overlaps with the sampling time for the INC I/P LATCH. If the propagate signal (P) is high, the carry in (CIN_1) is propagated to the next stage. If P is '0', which means the kill signal (K) is high, the pass transistor is off and the kill transistor is on, thus pulling down the carry chain output to a logic low level (i.e., $COUT_1$ is '0'). Note that in the case of the LSB the IOP is just the complement of the input Y. The bits Y_2 to Y_{11} are evaluated the same way as bit Y_1. For Y_1 to Y_{11} the INC LOGIC consists of the carry chain and the EX-OR blocks. For Y_0 the INC LOGIC reduces to a simple inverter.

(c) FLOOR PLAN

Refer to Figure B.1.c.

(d) CIRCUIT SCHEMATIC

The circuit schematic of 1 bit of the incrementer is shown along with the LSB in Figure B.1.d.

B.2. SUBROUTINE STACK (PUSH-DOWN STACK)

The stack subsystem is commonly called a last-in, first-out stack (LIFO). It is also known as a push-down (PD) stack. The name subroutine stack comes from the fact that it stores return addresses whenever a subroutine is called. The four principal operations in a full clock period are :

(i) PUSH in a new data word at one end of the array, pushing all previously entered words one word position down in the array.

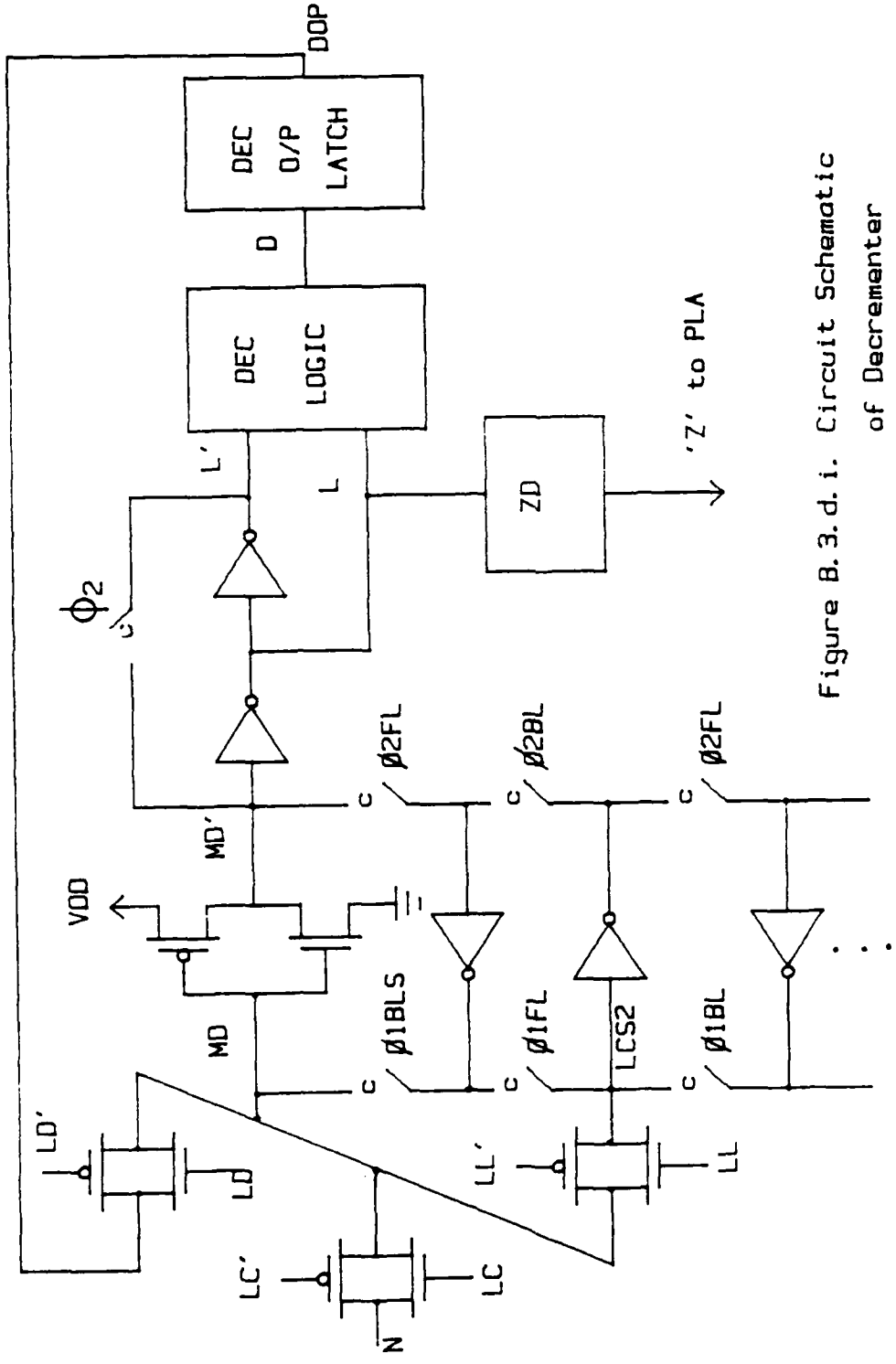


Figure B.3.d.i. Circuit Schematic of Decrementer

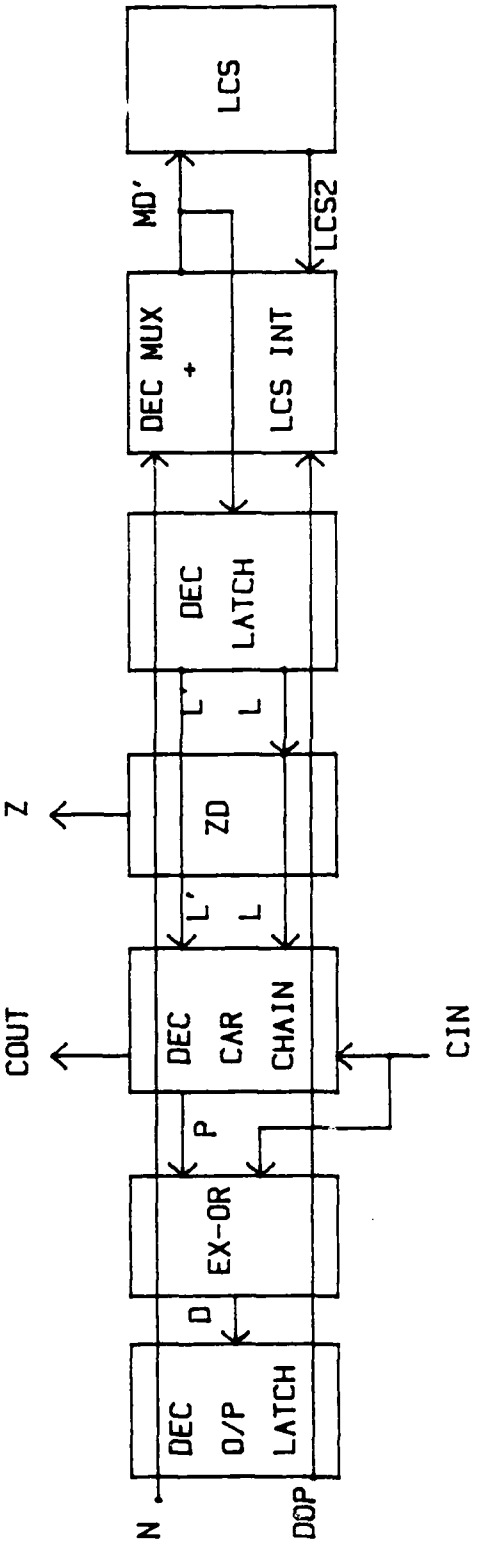


Figure B.3.c. Floor Plan of Decrementer

and LL from PLA and it provides the input (MD) to the DEC LATCH as well as to the LCS.

(c) FLOOR PLAN

Refer to Figure B.3.c.

(d) CIRCUIT SCHEMATIC

Refer to Figure B.3.d.i. for the circuit schematic of the decrementer, excluding the schematics of the DEC LOGIC and ZD which are shown in Figure B.3.d.ii. and Figure B.3.d.iii. respectively. LCS is also a push-down stack similar to SS and is shown with LCS interface in Figure B.3.d.iv. The schematic of the stack is not shown here since it has already been shown in the SS macro.

Note that the schematics of inverters and switches are not repeated for the sake of simplicity. The ZD is precharged high in phase2' and evaluated in phase2. If all the inputs are low, the N-transistors are off and 'Z' stays high. If any one of the inputs is high, then Z is pulled low through the conducting N-transistor.

B.4. MAP LOGIC

A special type of branch exists when a microinstruction specifies a branch to the first word of a routine for a macro-operation in the control memory. The address bits for this type of branch are a funtion of bits used in the operation part of the machine instruction. One specific mapping process which converts the 5-bit op-code into a 12-bit address is shown in Figure B.4.i.

Figure B.3.b.ii. Logic Diagram of DEC LOGIC

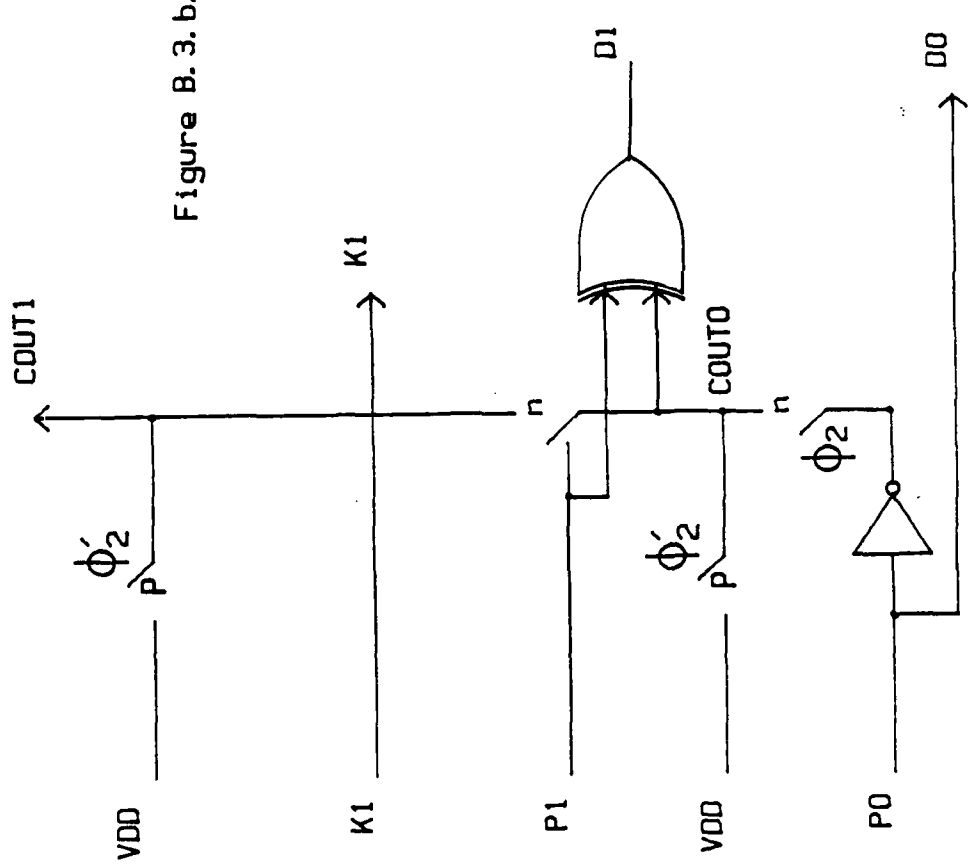
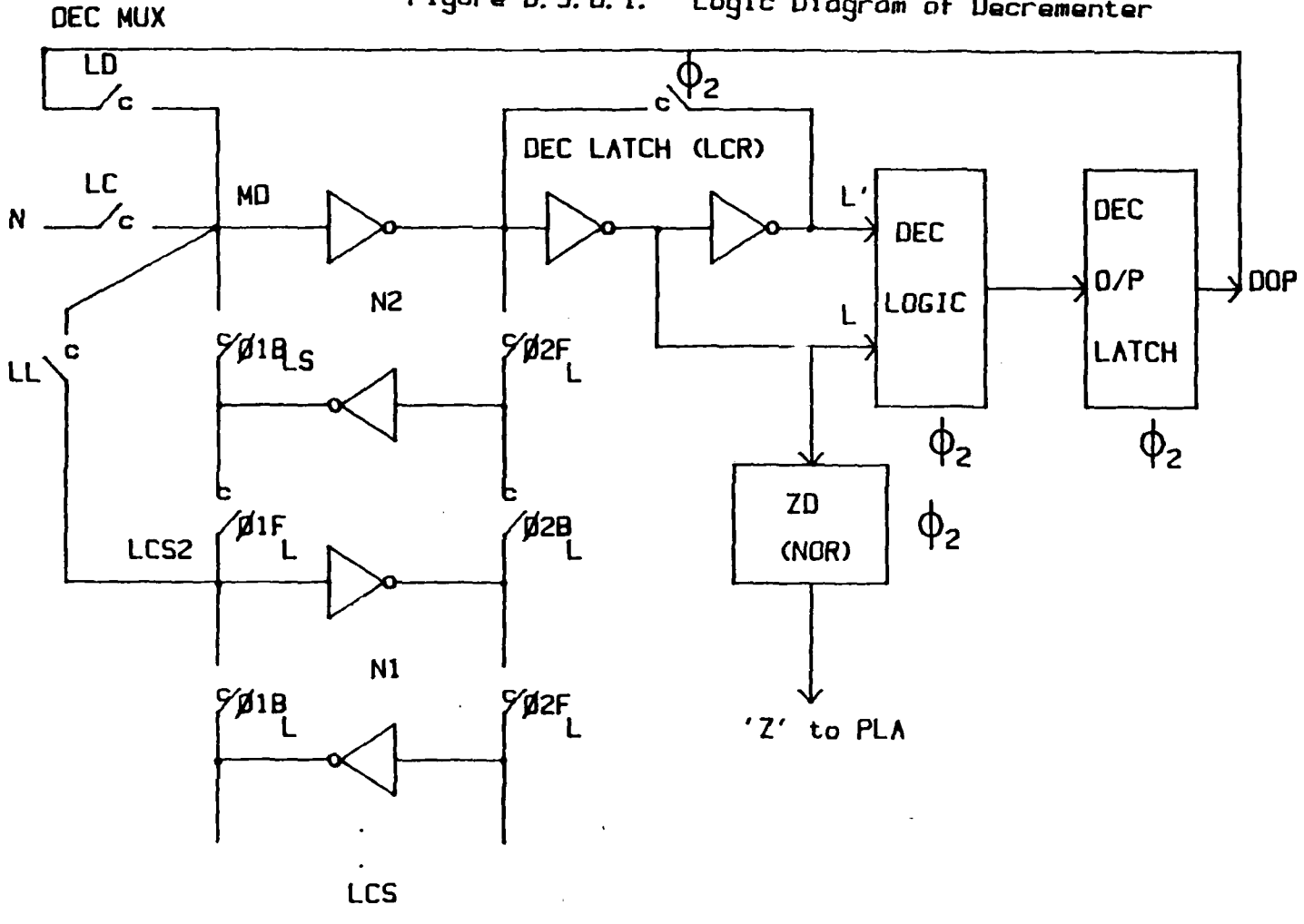


Figure B.3.b.i. Logic Diagram of Decrementer



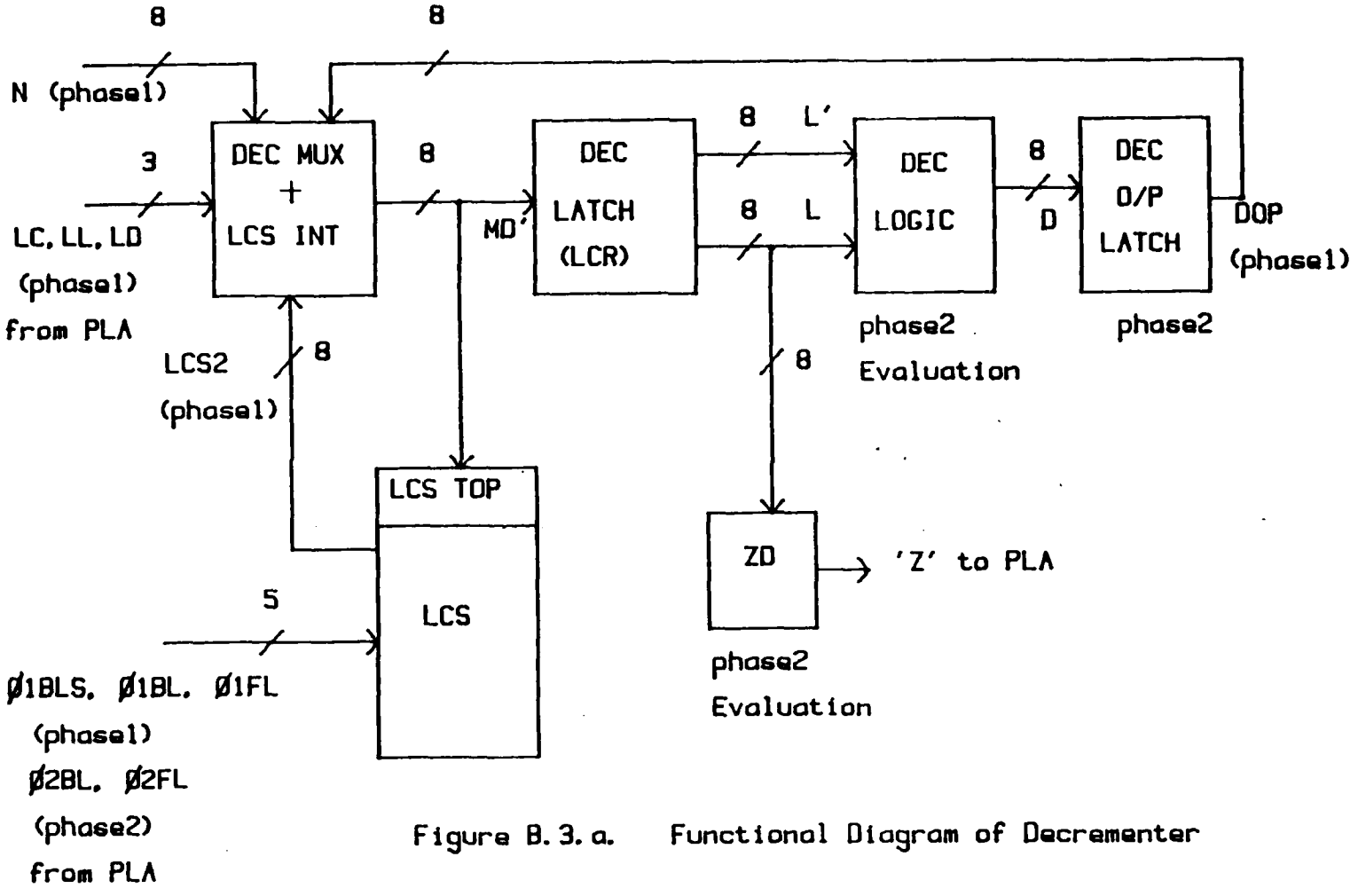


Figure B.3.a. Functional Diagram of Decrementer

checks for all zeros in LCR in phase2 and the ZD output (Z) becomes one of the inputs to the PLA. The count is decremented in phase2, sampled in the DEC O/P LATCH in phase2 and becomes stable in the next phase1. The decremented output is loaded back into the DEC LATCH and LCS stack top, in the same phase1, through the DEC MUX provided the control LD is high. As long as 'Z' is '0', PLA drives LD high. When 'Z' is '1' (i.e., all zeros in LCR), the PLA turns off LC and LD, and pops the LCS which means that the control signal LL is high. When a pop occurs, the second word of the stack (LCS2) is transferred to the top of the stack and also loaded into the LCR. The first word of the stack (stack top) always holds the value being decremented.

(a) FUNCTIONAL DIAGRAM

Refer to Figure B.3.a.

(b) LOGIC DIAGRAM

Refer to Figure B.3.b.i. for the logic diagram of the decrementer which consists of the DEC MUX, LOOP COUNTER STACK (LCS), DEC I/P LATCH (DEC LATCH or LCR), ZD, DEC LOGIC which consists of DEC CAR CHAIN and EX-OR blocks, and the DEC O/P LATCH. Refer to Figure B.3.B.ii. for the logic diagram of DEC LOGIC.

Note that the LCS is also a push-down stack similar to SS but it is different in the stack top and has one more control than SS (Refer to Figure B.3.i). In the case of the LSB the decremented output (DOP) is just the complement of the count 'L'. DEC MUX is controlled by the signals LC, LD

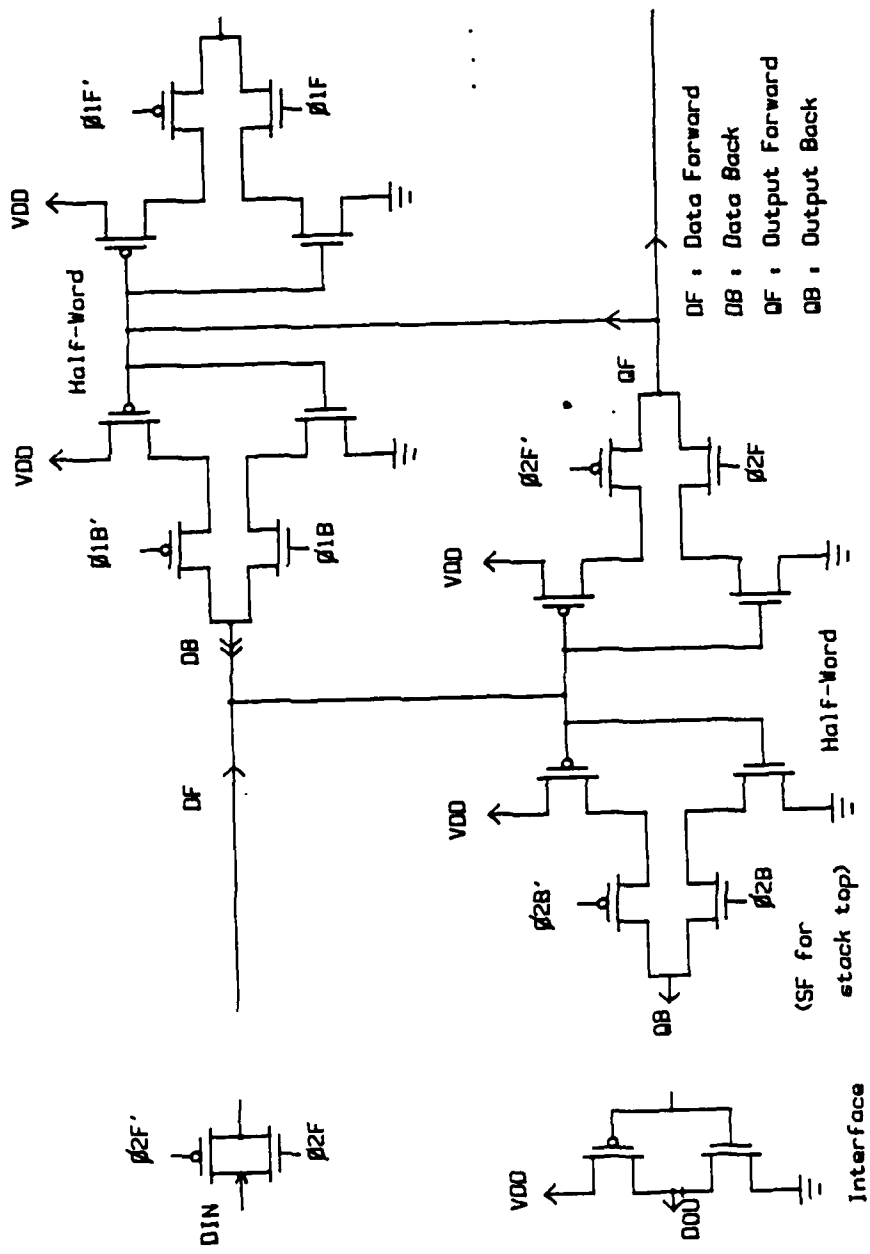


Figure B.2.d. Circuit Schematic of SS

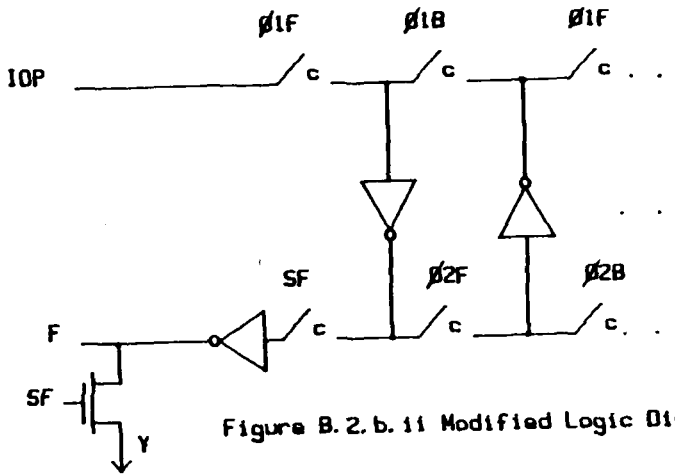


Figure B.2.b.ii Modified Logic Diagram of SS

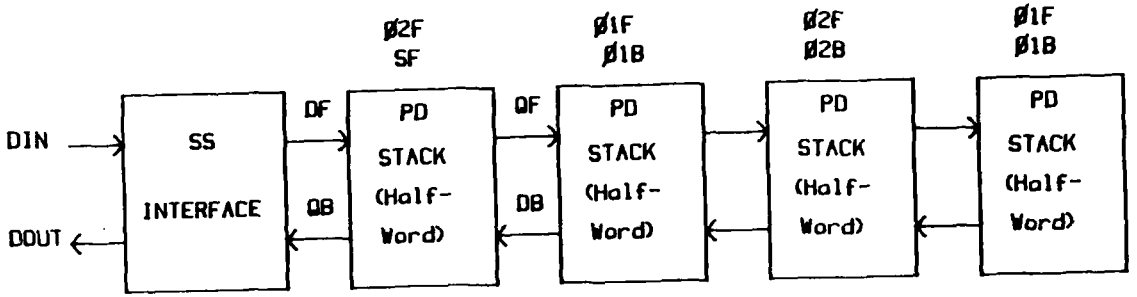


Figure B.2.c. Floor Plan of SS

first inverter, and then closing phase2F during time phase2 with data being transferred to the gate of second inverter. In order for data to be held in place phase1B is closed in phase1 followed by phase2F in phase2, causing the data to recirculate upon itself without shifting. Data (DOUT) is obtained by closing phase1B in time phase1 followed by phase2B in time phase2 (POP operation). But the problem with this is that DOUT is needed in phase1 because the control signal SF which selects the stack top F as the microprogram address comes in phase1. So, instead of signal phase2B to the first word of the stack (stack top), the control signal SF is connected as shown in Figure B.2.b.ii.

(c) FLOOR PLAN

Refer to Figure B.2.C.

Two half-words are stacked together to form one word. This is repeated 8 times to form a stack 8-words deep.

(d) CIRCUIT SCHEMATIC

Refer to Figure B.2.d.

B.3. DECREMENTER WITH ZD AND 4 WORD BY 8-BIT LCS (DEC)

An external count is loaded from the ADF field of the microinstruction in phase1 by driving the control signal LC high. The count MD (output of MUX1) is pushed onto the loop counter stack (LCS) and also sampled by the DEC I/P LATCH which is denoted by DEC LATCH or loop counter register (LCR). The count becomes stable (L) in phase2 and becomes an input to the zero detector (ZD) as well as to the decrementer logic (DEC LOGIC). The ZD is a NOR gate which

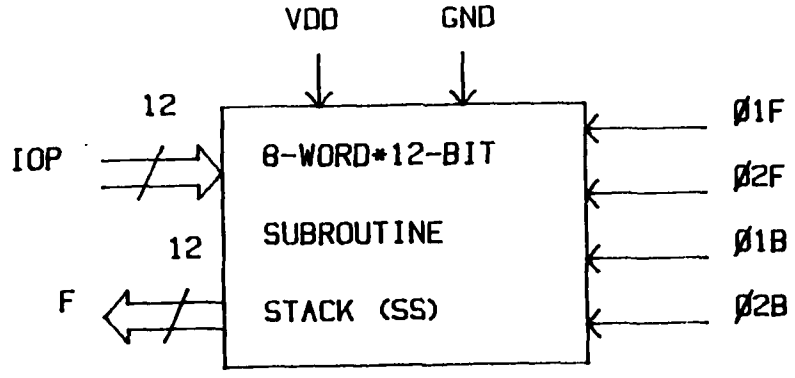


Figure B.2.a. Functional Diagram of SS

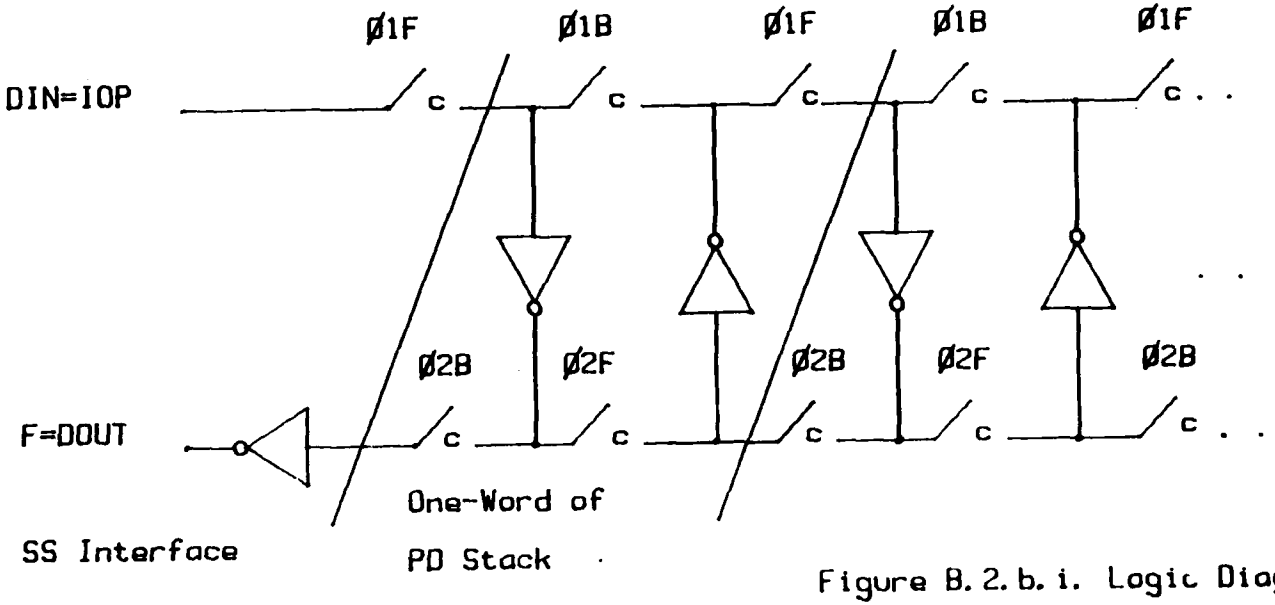


Figure B.2.b.i. Logic Diagram of SS

(ii) POP out a word from the end of the array, pulling up all previously entered words by one word position.

(iii) HOLD the words in their current position.

(iv) CLEAR the top of the stack by pushing a zero. (The whole stack could be cleared by letting the zero ripple through, which is explained below.) The actual controls that the instruction PLA issues to the stack are phase1F, phase2F, phase1B, phase2B. PUSH, POP, HOLD, and CLEAR are performed as explained below :

PUSH : close phase1F in phase1 and phase2F in phase2

POP : close phase1B in phase1 and phase2B in phase2

HOLD : close phase1B in phase1 and phase2F in phase2

CLEAR: close phase1F in phase1 and phase2F in phase2 with the input to the stack top (DIN) equal to 0.
(close phase1F and phase2F at the same time to clear the whole stack, i.e., to initialize the stack. But here it has been decided to clear only the top of stack to simplify the control logic. Moreover it is not necesary to clear the whole stack, since the '0' ripples to the next word in the stack whenever a PUSH operation takes place.)

(a) FUNCTIONAL DIAGRAM

Refer to Figure B.2.a.

(b) LOGIC DIAGRAM

Refer to Figure B.2.b.i.

PUSH is performed by closing phase1F in time phase1, thus transferring the input data (DIN) to the gate of the

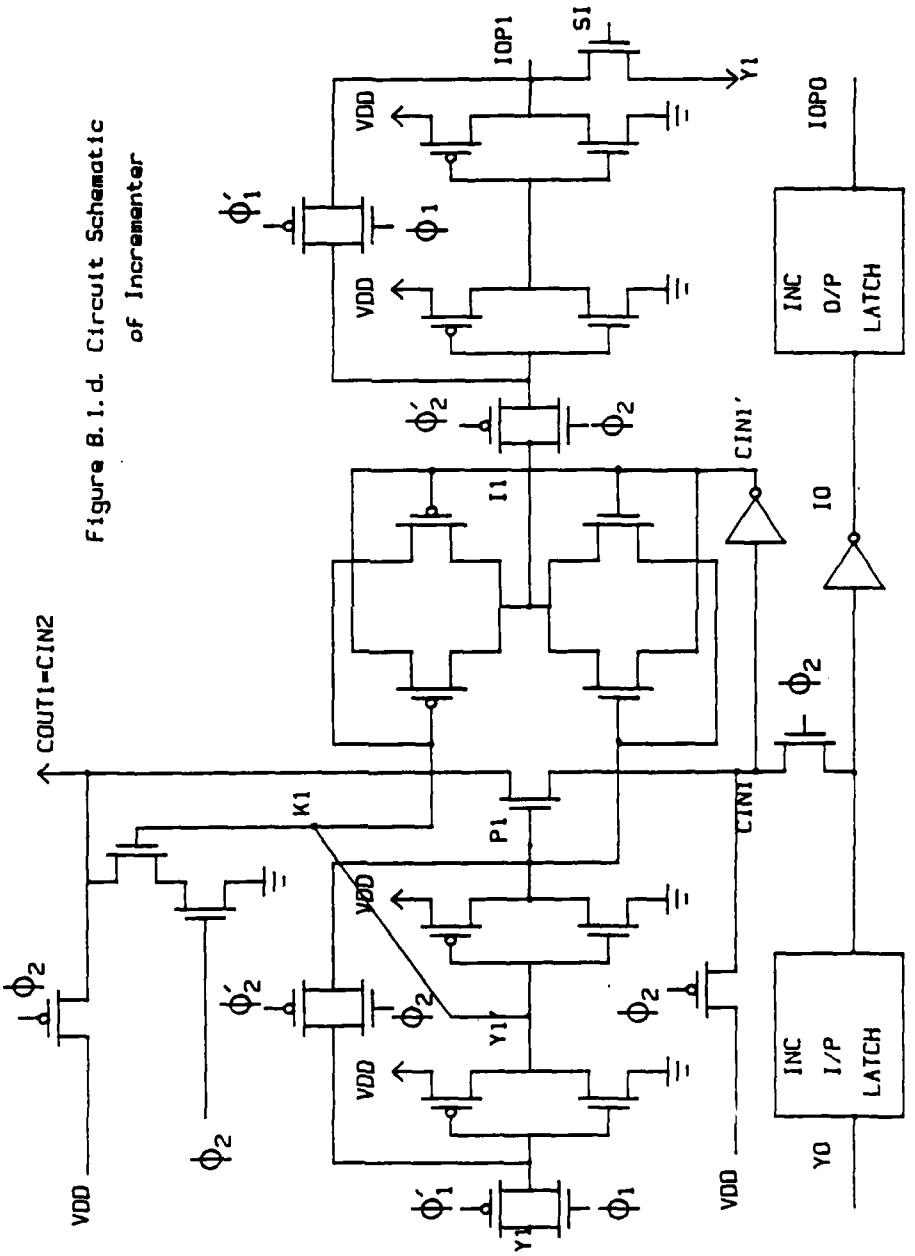


Figure B.1.d. Circuit Schematic of Incrementer

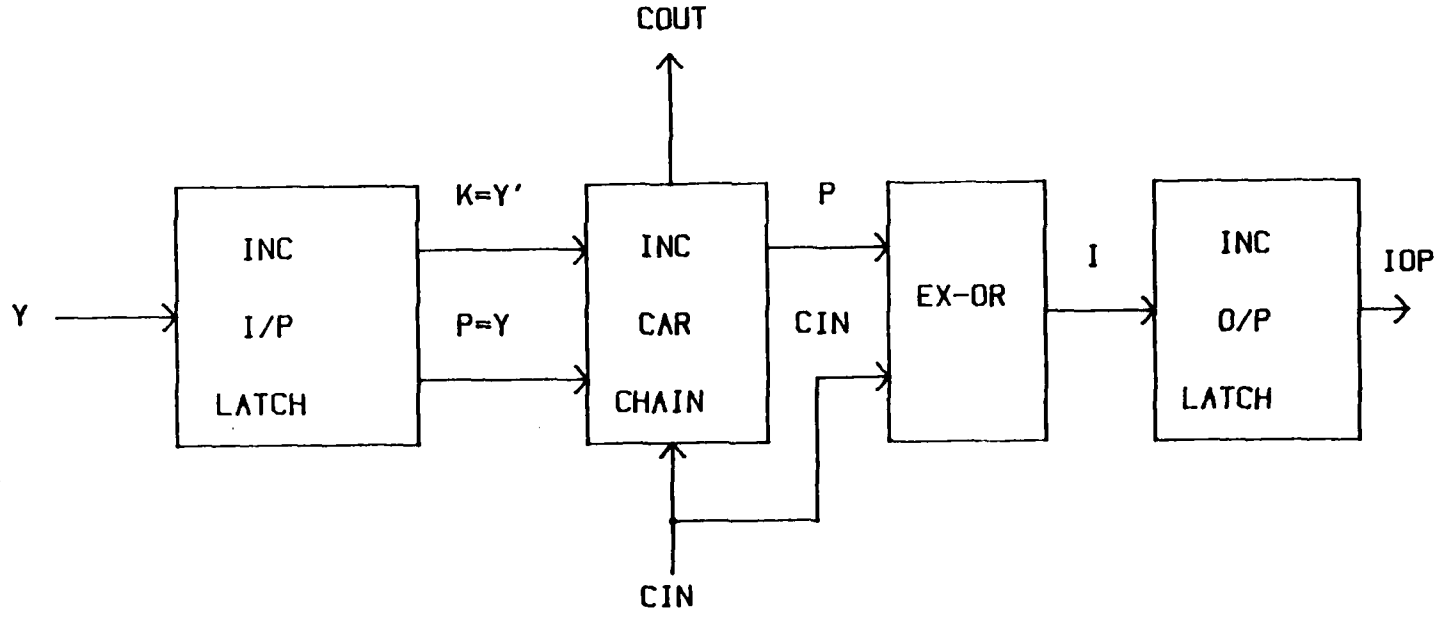


Figure B.1.c. Floor Plan of Incrementer

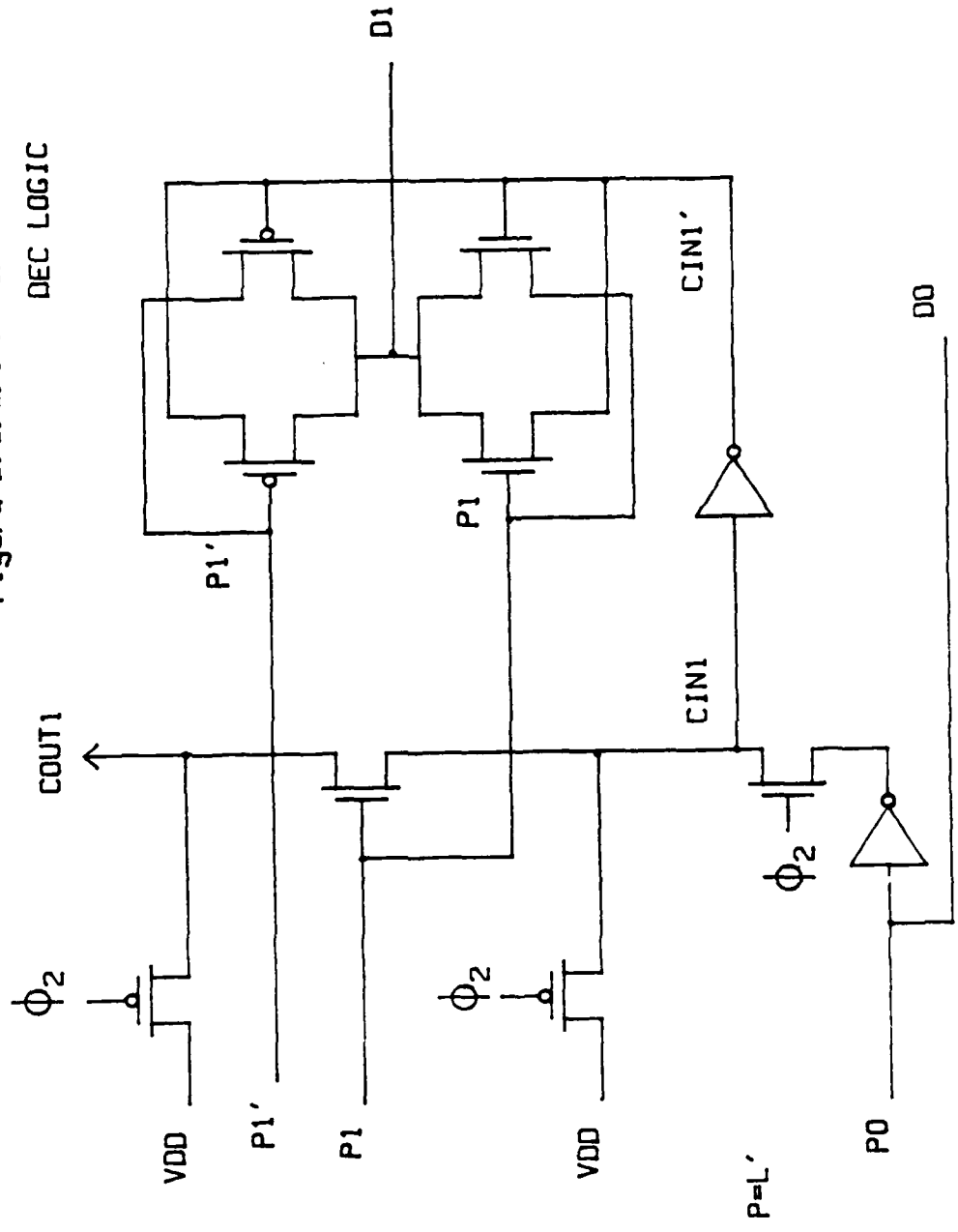


Figure B.3.d.ii. Circuit Schematic of DEC LOGIC

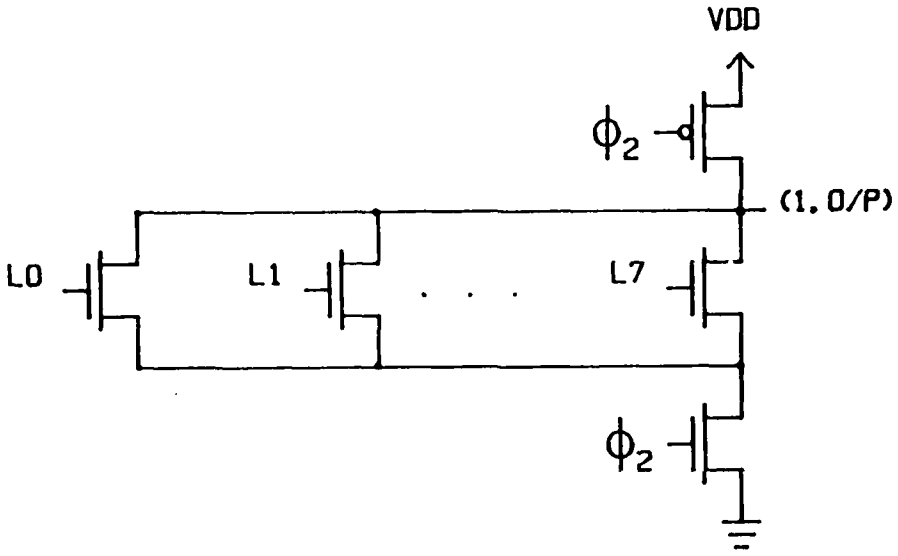


Figure B.3.d.iii. Circuit Schematic of ZD

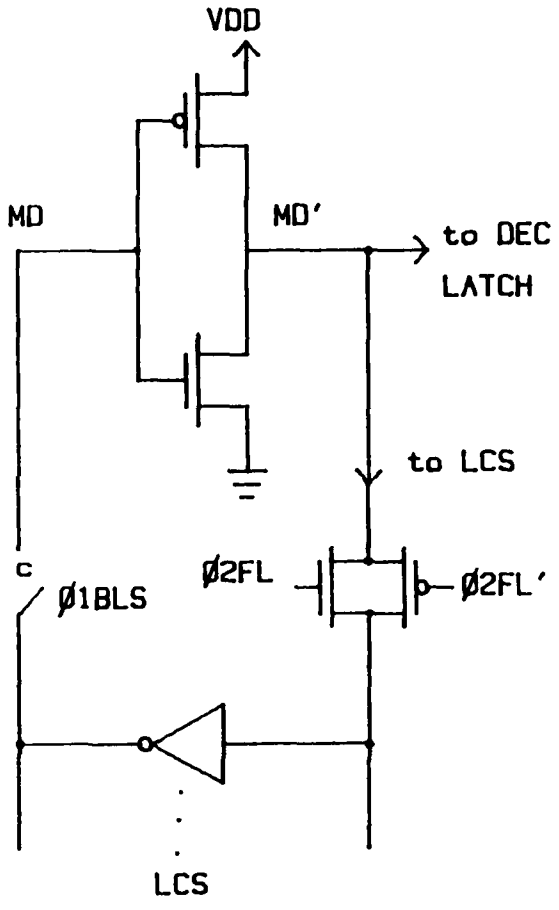
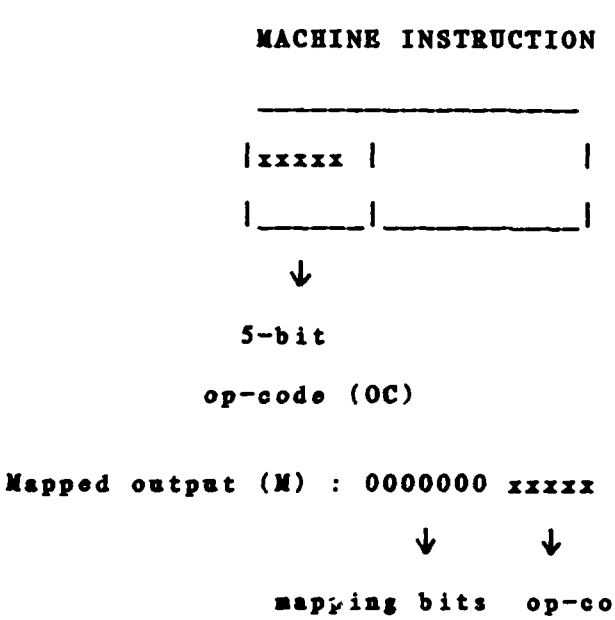


Figure B.3.d.iv. Circuit Schematic of LCS

This mapping consists of clearing the 7 most significant bits and transferring the op-code into the 5 least significant bits. The circuit schematic for MAP LOGIC is shown in Figure B.4.ii.

Figure B.4.i. MAPPING SCHEME



B.5. MULTIPLEXER 1 (MUX1)

MUX1 selects one of the four address sources as the microprogram address (output of the microsequencer). It is implemented as a distributed bus structure which is precharged high in phase1' and evaluated in phase1.

(a) FUNCTIONAL DIAGRAM

Refer to Figure B.5.a.

SA selects the ADF Latches output A.

SI selects the incrementer output IOP.

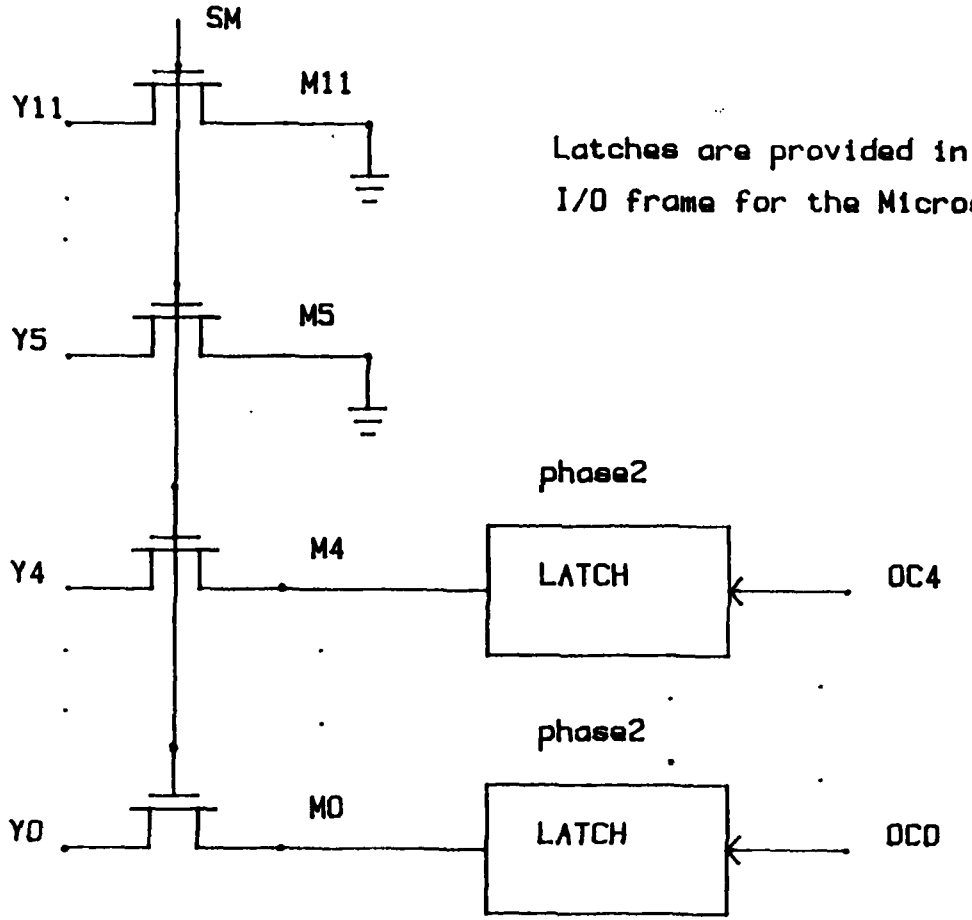


Figure B.4.11. Circuit Schematic of
MAP LOGIC

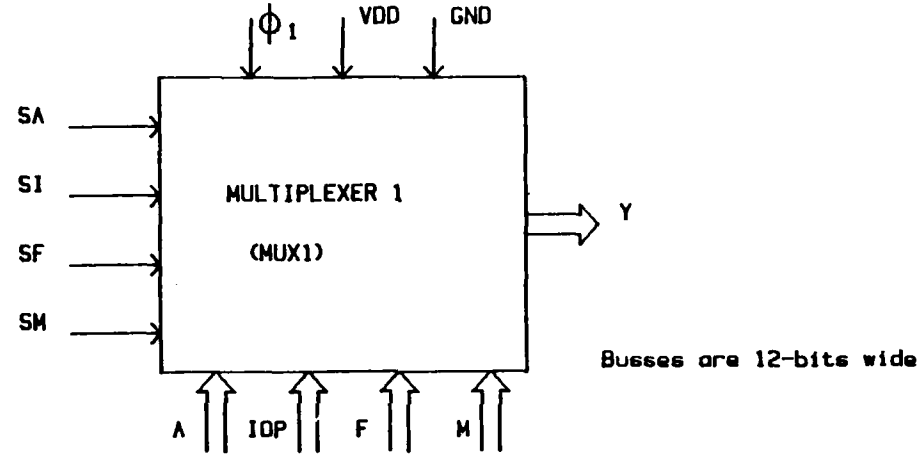


Figure B.5.a. Functional Diagram of MUX1

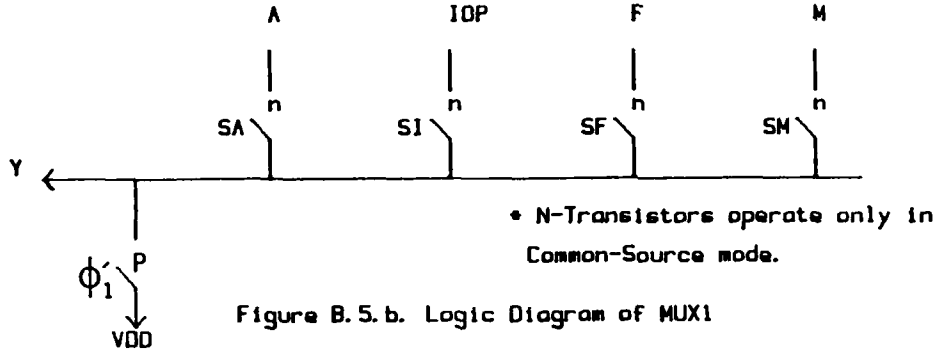


Figure B.5.b. Logic Diagram of MUX1

SF selects the SS stack top F.

SM selects the MAP Logic output M.

(b) LOGIC DIAGRAM

Refer to Figure B.5.b.

(c) FLOOR PLAN

Refer to Figure B.5.c.

(d) CIRCUIT SCHEMATIC

Refer to Figure B.5.d.

Output of MUX1 (Y) is precharged high in phase1' and conditionally pulled low in phase1. At any one time only one of the four controls SA, SI, SF, SM is high, thus discharging the bus conditionally through the selected N-transistors.

B.6. INSTRUCTION PLA

The instruction PLA has six inputs and seventeen outputs and it is evaluated in phase1. The AND-plane has series-N devices which drive the parallel-P devices in the OR-plane. The AND-plane is precharged high in phase1' and the OR-plane is precharged low in phase1'. N-type phase1 evaluation devices are added to the AND-plane so that the inputs need not be precharged. The inputs to the OR-plane are precharged high since they are also the outputs of the AND-plane. So there is no need for the evaluation devices in the OR-plane. The inputs to the PLA enter the chip before phase1. To make sure that the inputs to the PLA are stable by the time the evaluation takes place, the inputs to the

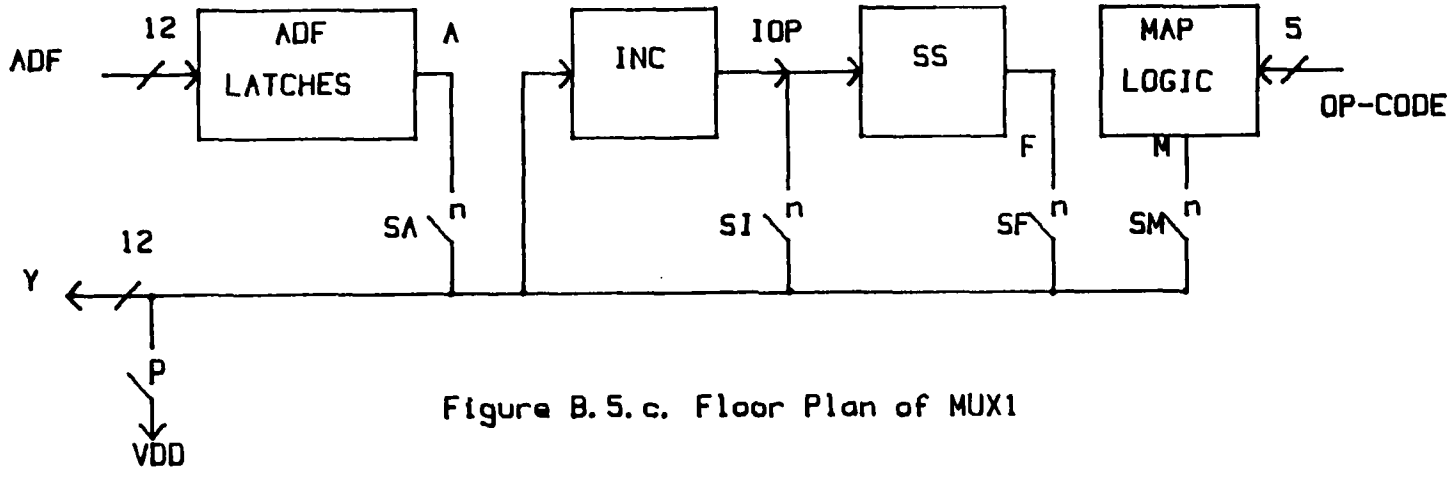


Figure B.5.c. Floor Plan of MUX1

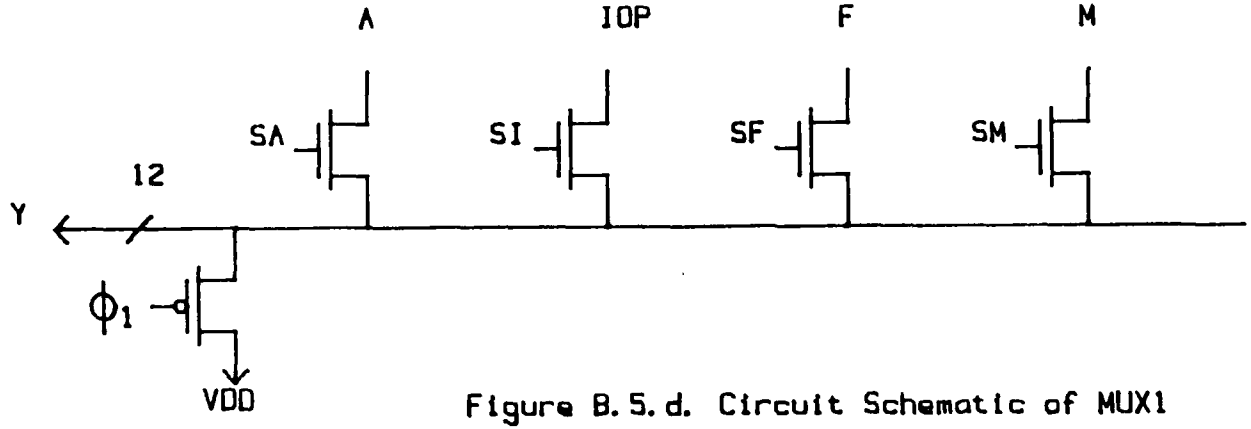


Figure B.5.d. Circuit Schematic of MUX1

PLA, except the test flag 'T', are sampled in phase2 and are stable in phase1. The test bit T is sampled in phase2'*phase1' and is stable in phase1.

The outputs of the PLA are buffered to increase the drive capability. Since the PLA controls the microsequencer logic in both phase times, some of the outputs of the PLA have to go through gated buffers.

(a) FUNCTIONAL DIAGRAM

Refer to Figure B.6.a.

(b) LOGIC DIAGRAM

Refer to Figure B.6.b.i. which shows the basic scheme of the instruction PLA. The actual PLA consists of the AND-plane with 6 inputs and 18 outputs (implicants). The OR-plane has 18 inputs from the AND-plane and 17 outputs which are the controls to all the other blocks in the microsequencer.

Refer to Figure B.6.b.ii. for buffers and gated-buffers. (Some of the buffers are gated in phase2 because they are required to buffer the phase2 controls.)

(c) FLOOR PLAN

Refer to Figure B.6.c.

(d) CIRCUIT SCHEMATIC

Refer to Figure B.6.d.i. for gated and non-gated buffers. The schematic of the AND-OR plane is shown in Figure B.6.d.ii.

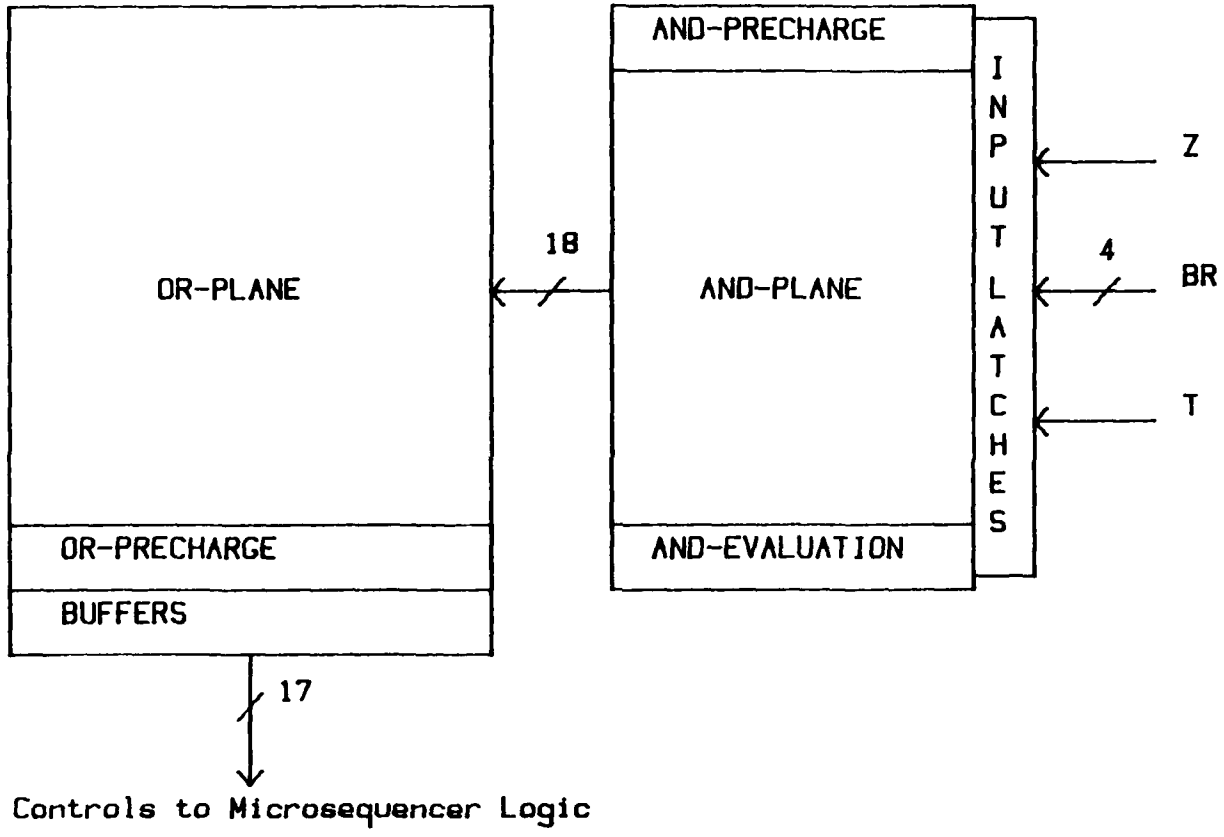
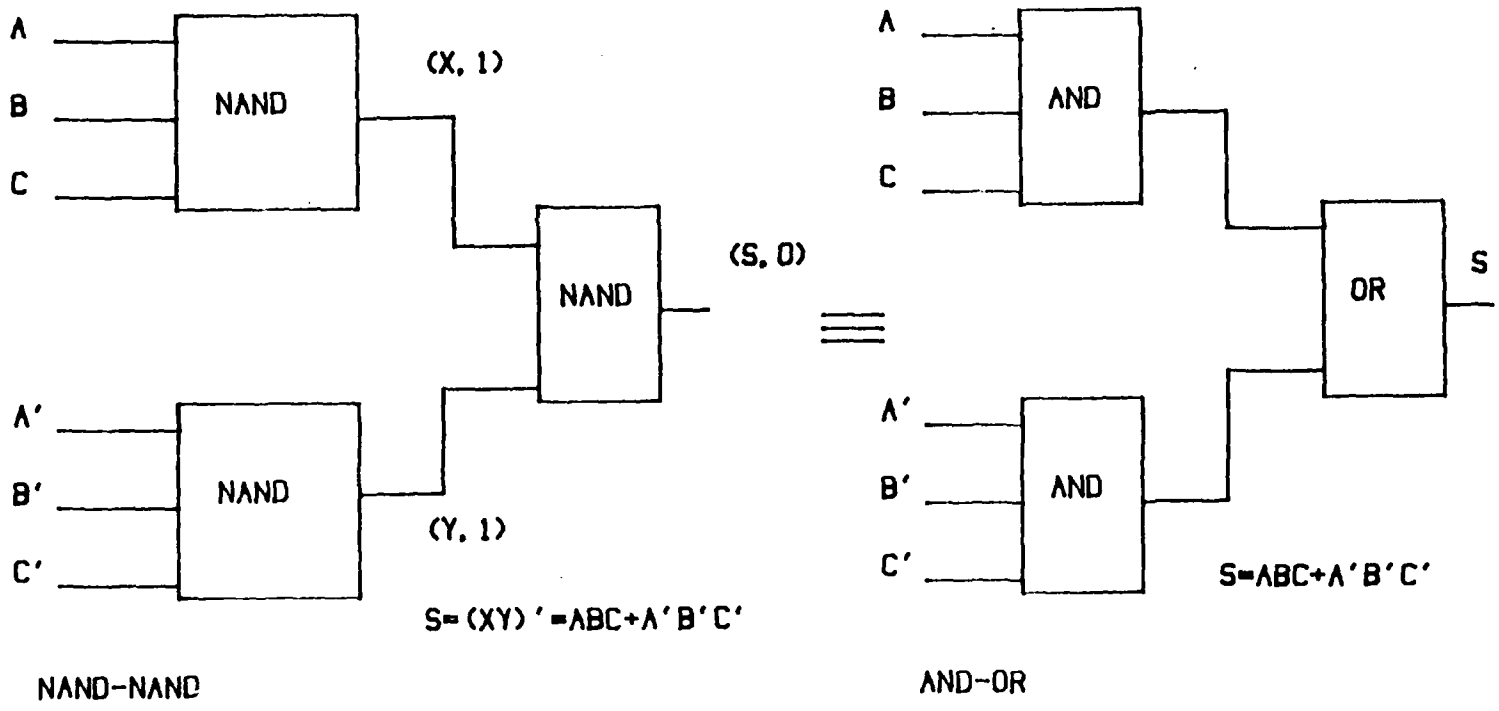
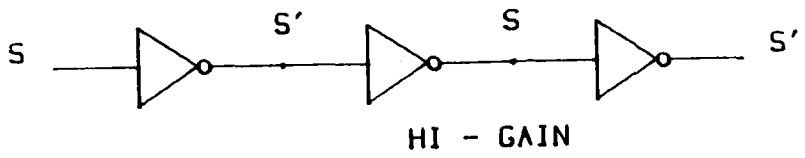


Figure 8.6.a. Functional Diagram of PLA

Figure 8.6.b.1. Logic Diagram of
AND-OR Plane



Phase1 Buffers



Phase2 Buffers

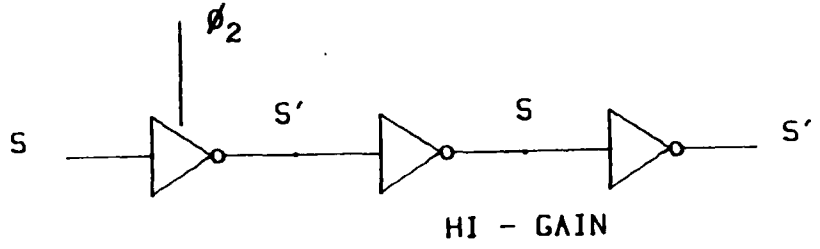


Figure B.6.b.ii. Logic Diagrams of
Buffers for PLA outputs

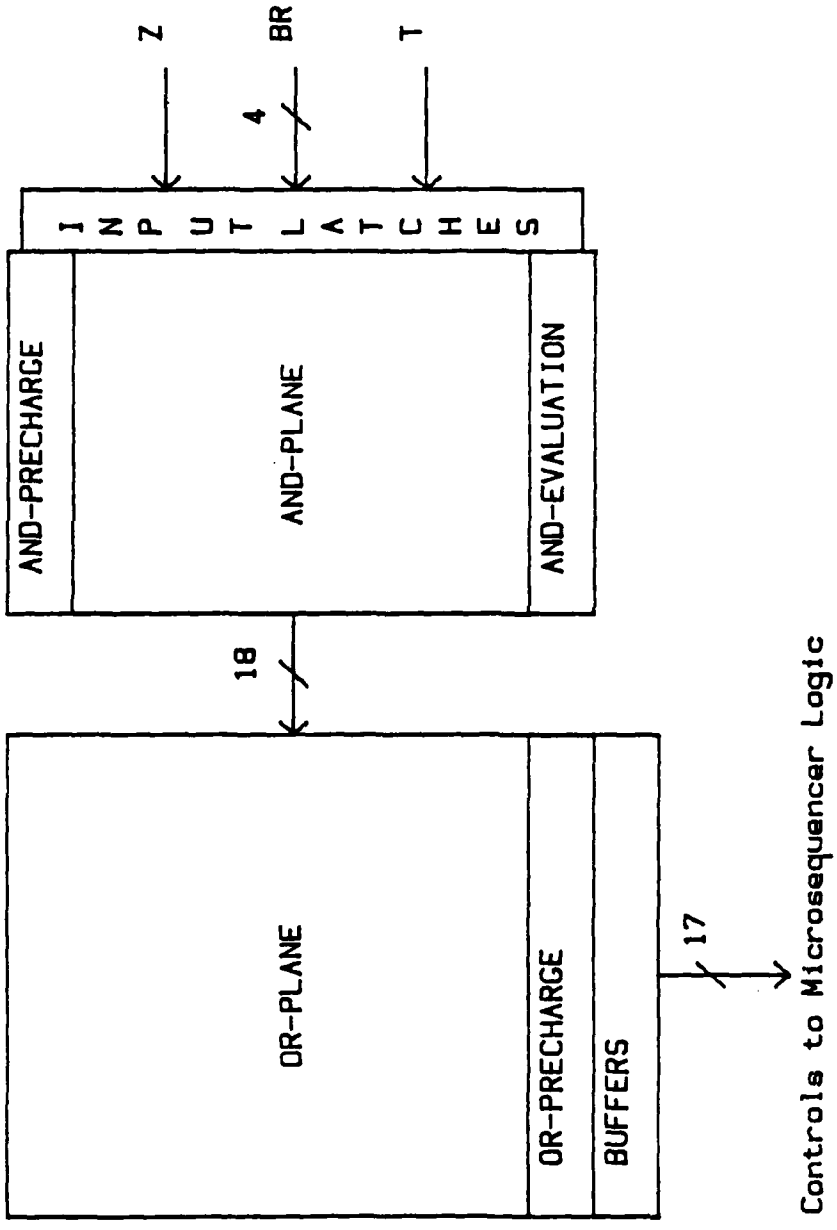


Figure B.6.c. Floor Plan of PLA

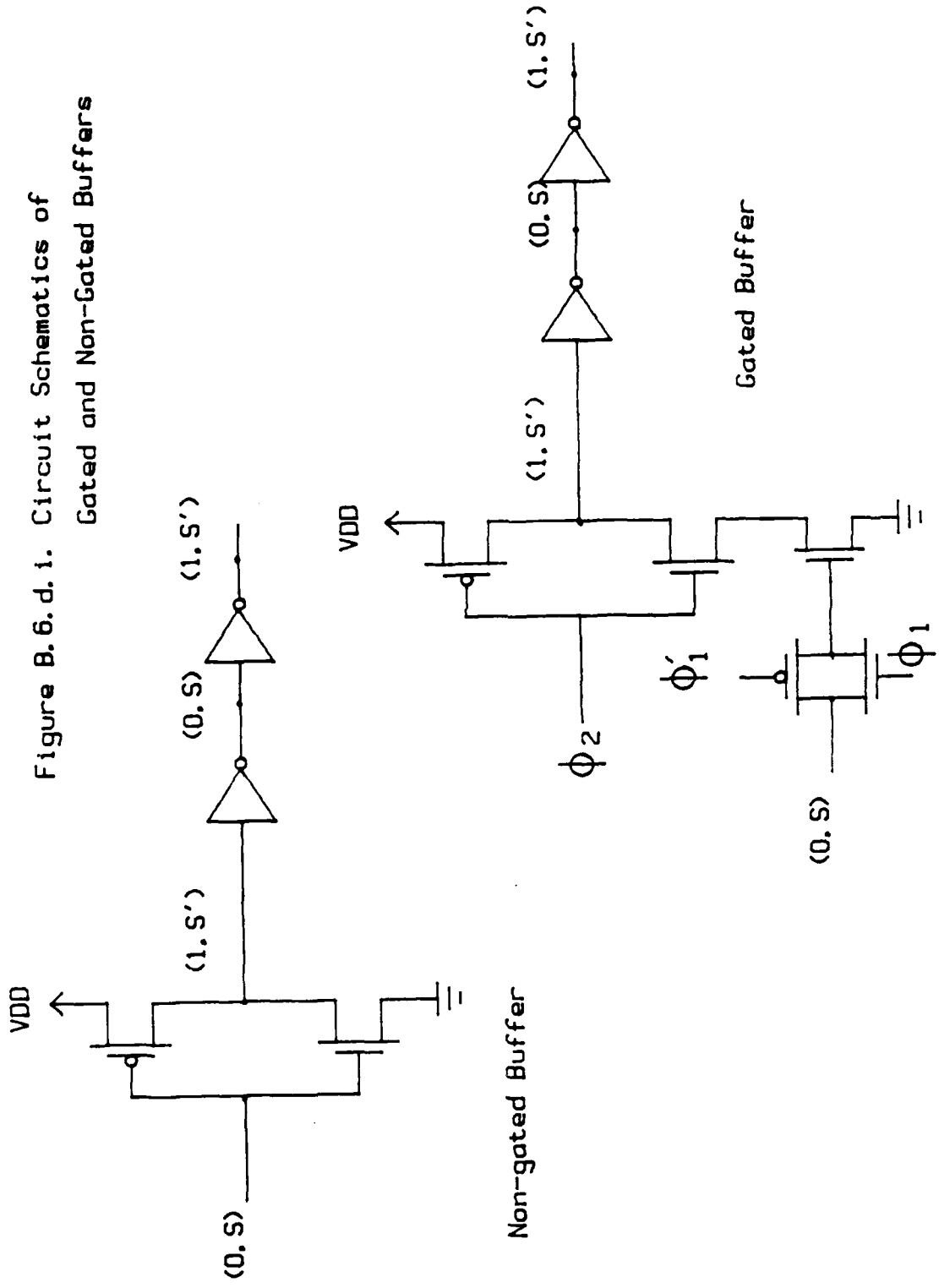


Figure B.6.d.i. Circuit Schematics of Gated and Non-Gated Buffers

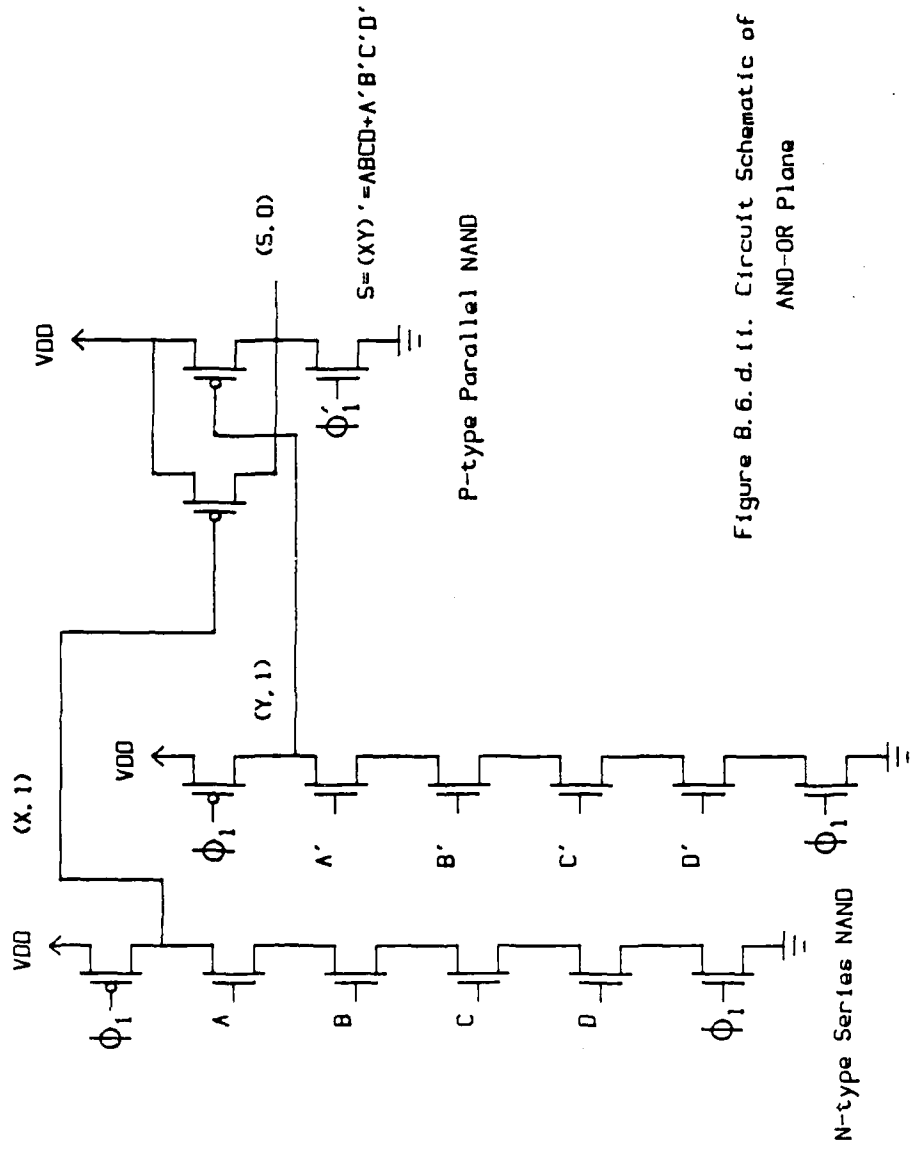
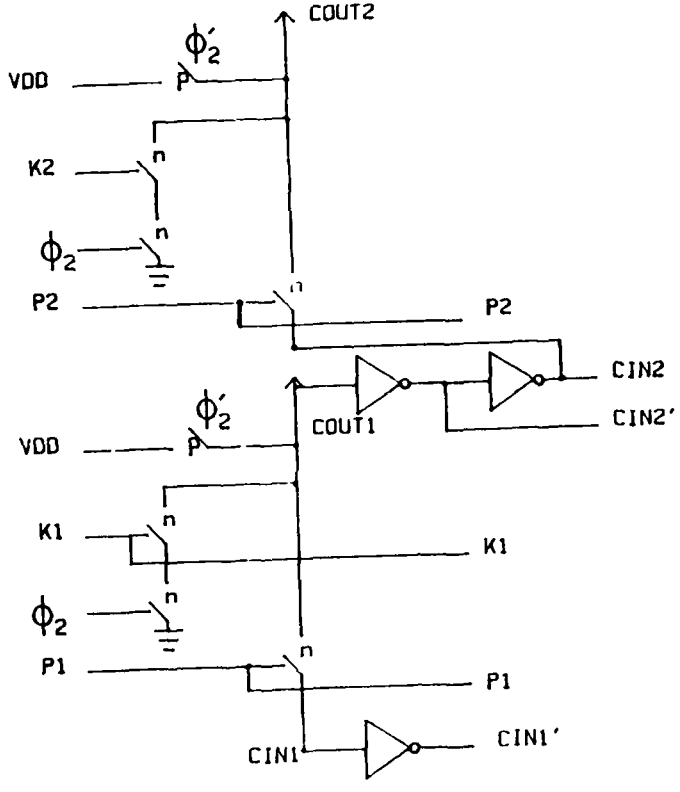
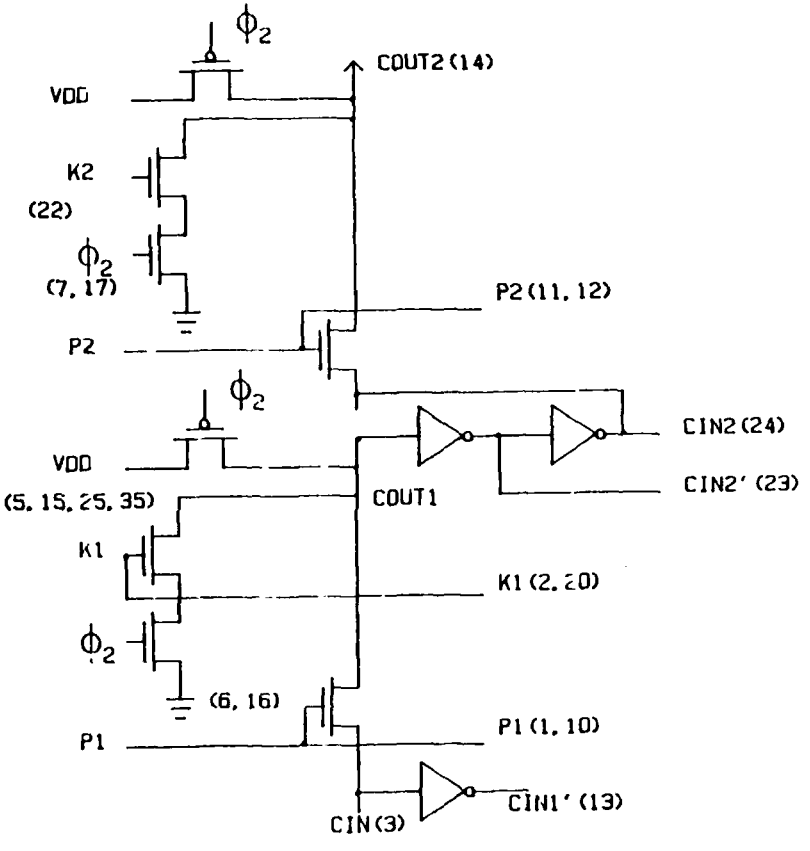


Figure 8.6.d.11. Circuit Schematic of AND-OR Plane



Logic Diagram of INC CAR CHAIN 2



Circuit Schematic of INC CAR CHAIN 2

SCALABLE CMOS CELL FAMILY	NO. : C.2.ii TITLE : INC CAR CHAIN 2 HEIGHT : 64 WIDTH : 48	DATE: 3/25/85 BY : GOWNI

TRUTH TABLE

phase2	P	CIN	COUT
0	x	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

LOGIC EQUATION(S)

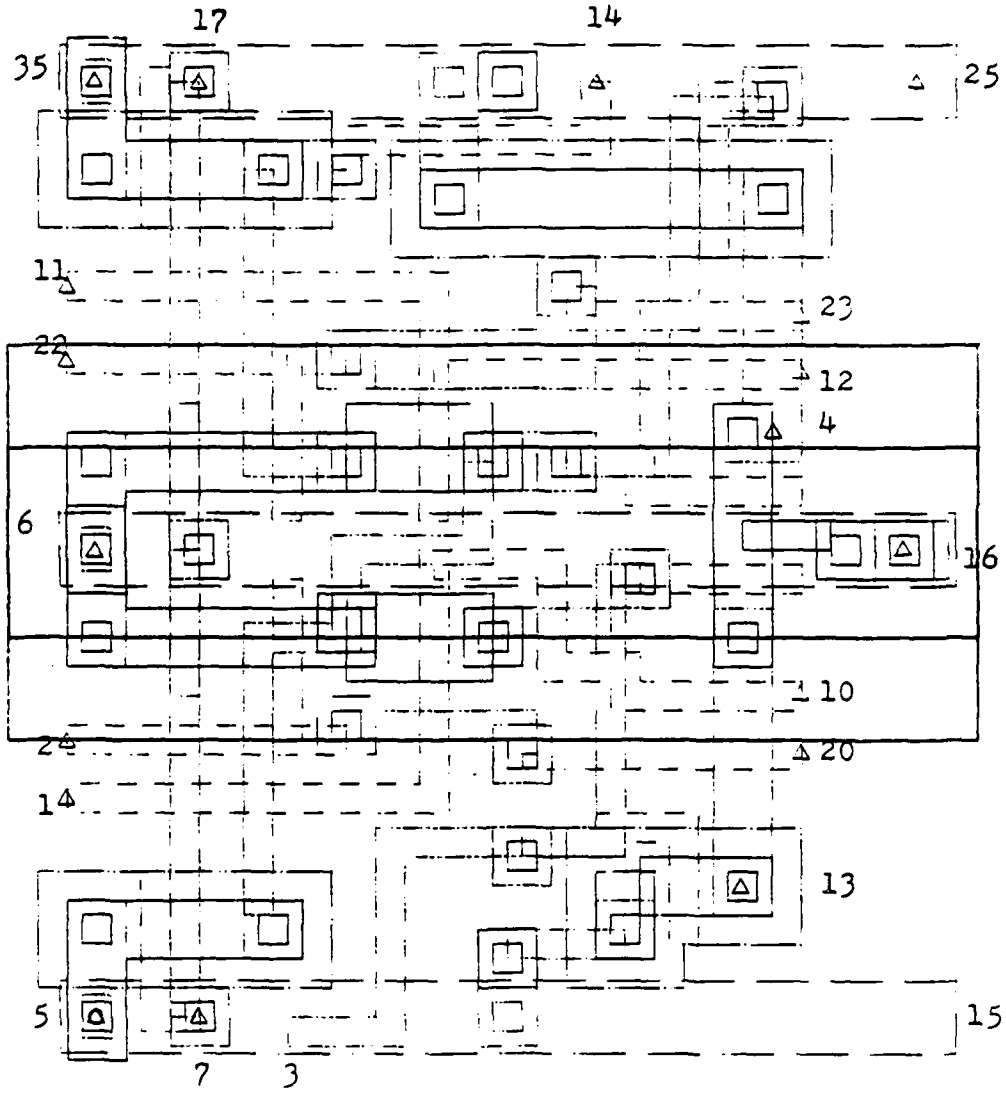
COUT= phase2' +
phase2.(P'+ CIN)

TERMINAL INFORMATION

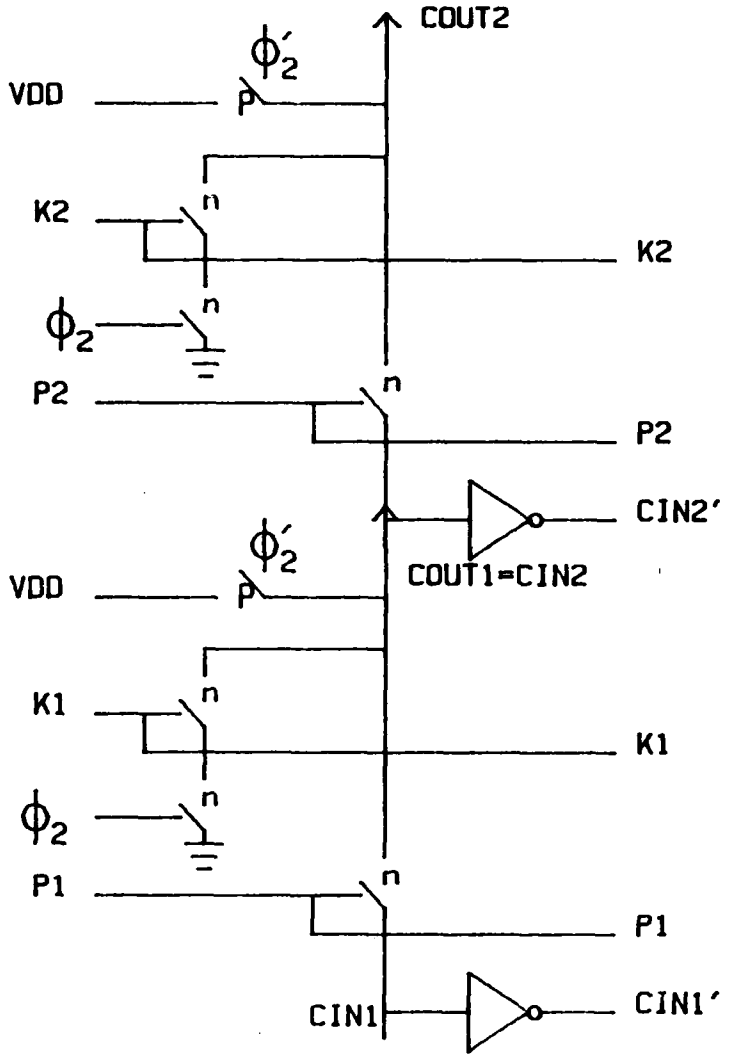
Name	No.	Location	Layer	Abbr.
Pass1	1,10	(-2,13) (44,18)	gp	P1
Pass2	11,12	(-2,50) (41,64)	(gp)(fm,ip,gp)	P2
Voltage Supply	5,15, 25,35	(0,0) (48,0) (48,64) (0,64)	(fm,sm)(sm) (sm)(fm,sm)	VDD
Phase2	7,17	(7,0) (7,64)	fm,ip,gp	ϕ_2
Kill1	2,20	(-2,17) (44,14)	gp	K1
Kill2	22	(-2,45)	gp	K2
Carry-in 1	3	(35,1)	fm,ip,gp	CIN1
Carry-in 1'	13	(51,5)	fm	CIN1'
Carry-in 2	24	(44,49)	gp	CIN2
Carry-out 2	14	(13,64)	fm	COUT2
Carry-in 2'	23	(44,43)	gp	CIN2'
Ground	6,16	(0,32) (39,32)	(fm,sm)(fm)	GND

NOTES

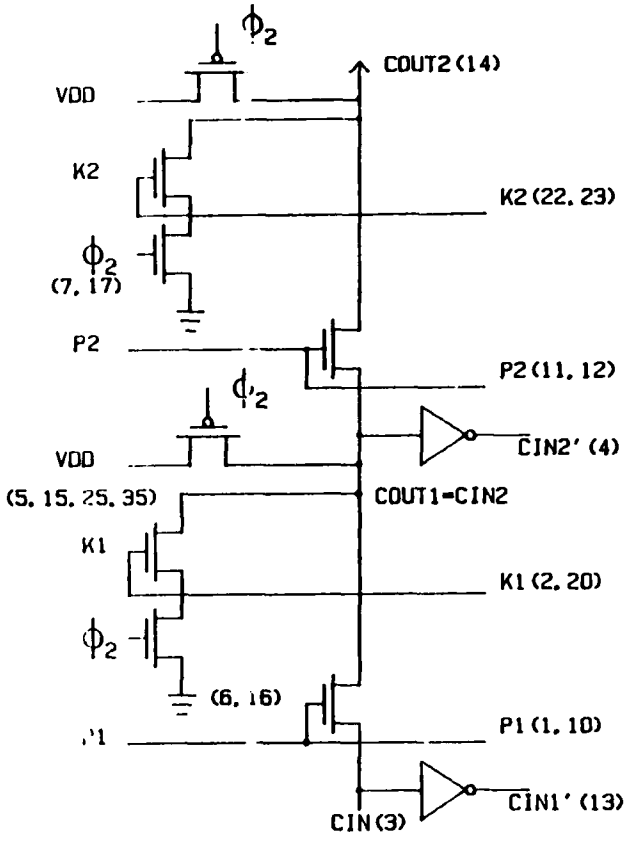
W/L of the pass transistor is 3:1. This carry chain is used every fourth stage with EX-OR circuit being evaluated from P, CIN, and CIN' instead of P, P', and CIN'. Note that the layout is for two bits; one stage contains the buffered CIN and the other is similar to the one in INC CAR CHAIN 1.



scale: 2080X / INC CAR CHAIN 1



Logic Diagram of INC CAR CHAIN 1



Circuit Schematic of INC CAR CHAIN 1

SCALABLE CMOS CELL FAMILY	NO. : C.2.i		DATE: 3/25/85
	TITLE : INC CAR CHAIN 1		BY : GOWNI
	HEIGHT : 64	WIDTH : 48	

TRUTH TABLE

phase2	P	CIN	COUT
0	x	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

LOGIC EQUATION(S)

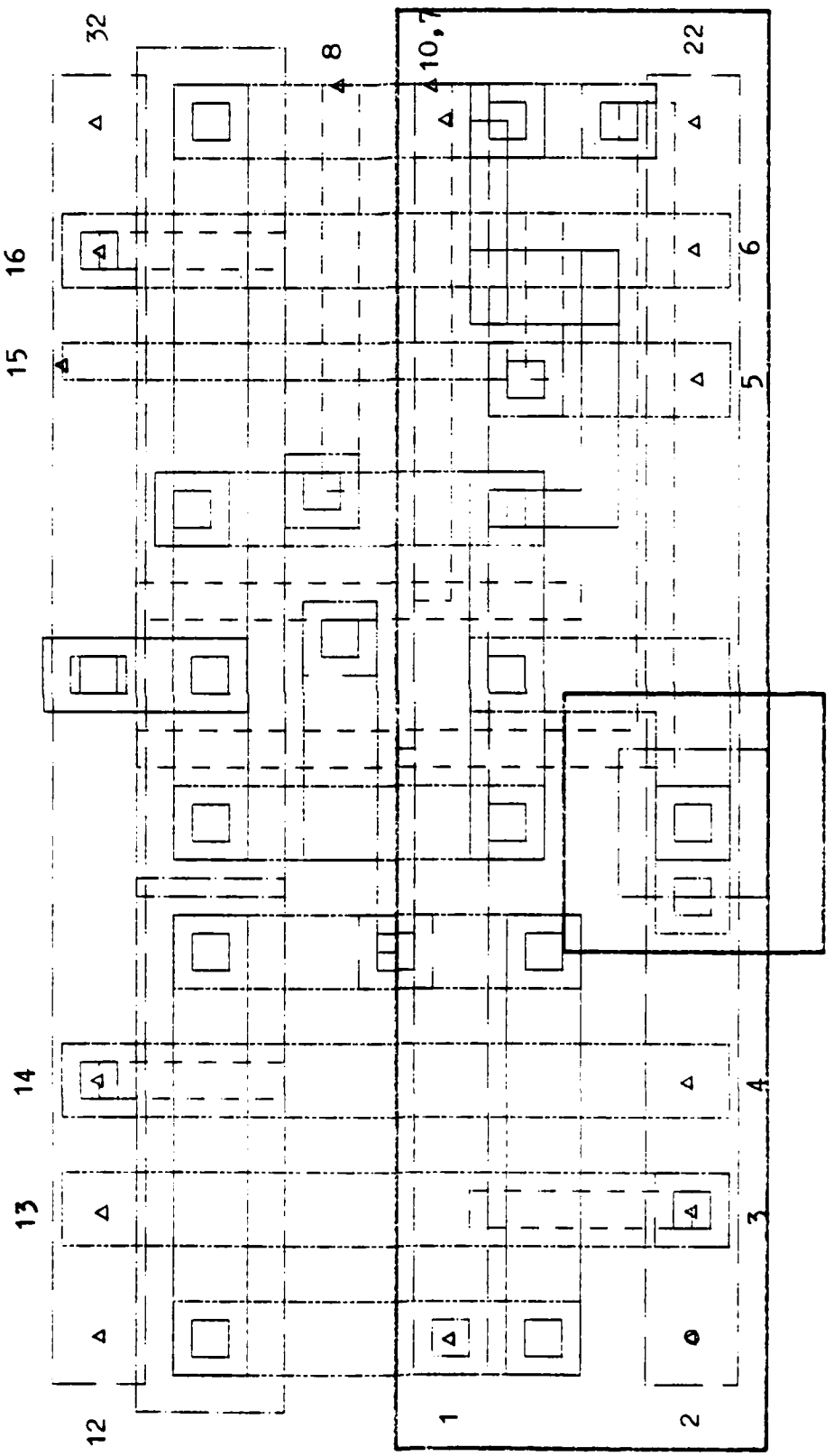
COUT= phase2' +
phase2.(P'+ CIN)

TERMINAL INFORMATION

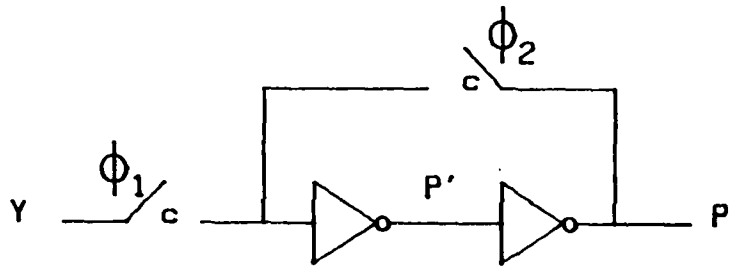
Name	No.	Location	Layer	Abbr.
Pass1	1,10	(-2,15) (48,22)	gp	P1
Pass2	11,12	(-2,50) (48,44)	gp	P2
Voltage Supply	5,15, 25,35	(0,0) (48,0) (48,64) (0,64)	(fm,sm)(sm) (sm)(fm,sm)	VDD
Phase2	7,17	(7,0) (7,64)	fm,ip,gp	ϕ_2
Kill1	2,20	(-2,19) (48,18)	gp	K1
Kill2	22,23	(-2,45) (48,48)	gp	K2
Carry-in 1	3	(14,-1)	fm	CIN1
Carry-in 1'	13	(44,9)	fm	CIN1'
Carry-in 2'	4	(46,40)	fm	CIN2'
Carry-out 2	14	(34,64)	gp	COUT2
Ground	6,16	(0,32) (48,32)	(fm,sm)(fm)	GND

NOTES

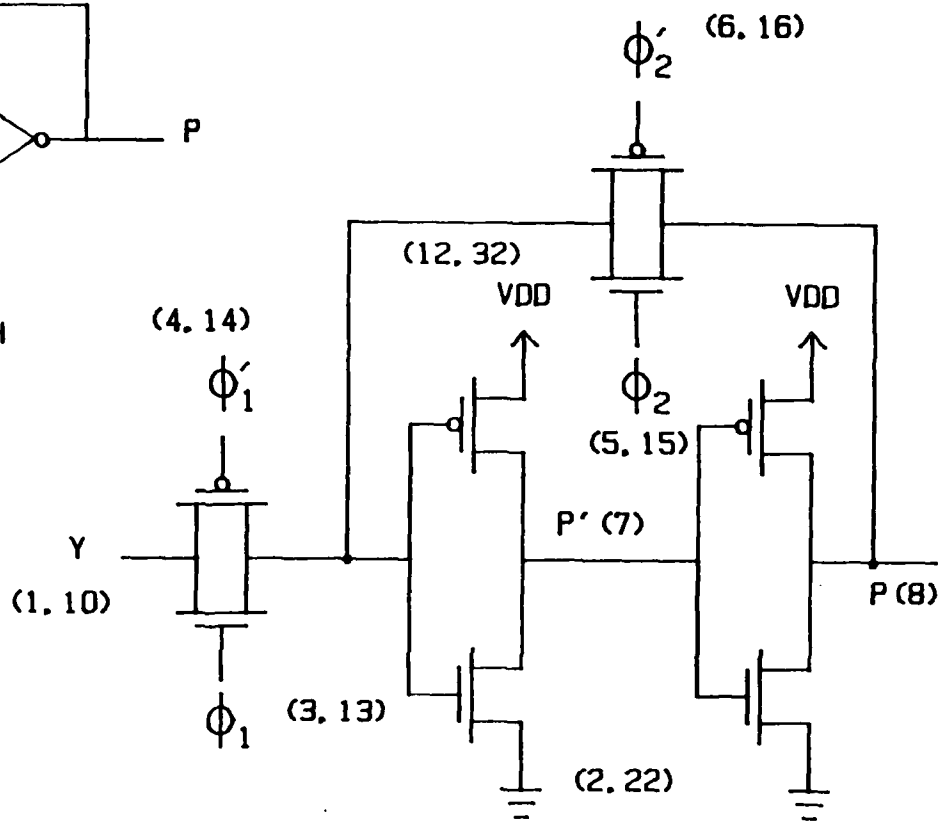
Note that the layout shows 2 bits of the carry-chain; therefore the terminal information section is for two bits. The W/L ratio of the pass transistor is 3:1. The COUT of this stage is buffered (every four stages) to become the CIN of the next stage which is shown in the cell INC CAR CHAIN 2. Also, the CIN of every stage (except the buffered stage) is inverted to provide the active drive for the EX-OR circuit.



scale: 2800x / INC I/P LATCH



Logic Diagram of INC I/P LATCH



Circuit schematic of INC I/P LATCH

SCALABLE CMOS CELL FAMILY	NO. : C.1 TITLE : INC I/P LATCH HEIGHT : 32 WIDTH : 66	DATE: 3/25/85 BY : GOWNI

TRUTH TABLE

phase1	Y	P
0	x	P*
1	0	0
1	1	1

LOGIC EQUATION(S)

P = phase1.Y
+ phase1'.P*

TERMINAL INFORMATION

Name	No.	Location	Layer	Abbr.
Data input	1,10	(0,13) (66,13)	(fm,sm)(sm)	Y
Ground	2,22	(0,0) (66,0)	sm	GND
Voltage Supply	12,32	(0,32) (66,32)	sm	VDD
Phase1	3,13	(7,0) (7,32)	(fm,ip,gp)(fm)	ϕ_1
Phase1'	4,14	(14,0) (14,32)	(fm)(fm,ip,gp)	ϕ_1'
Phase2	5,15	(52,0) (52,32)	fm	ϕ_2
Phase2'	6,16	(59,0) (59,32)	(fm)(fm,ip,gp)	ϕ_2'
Output	8	(68,19)	gp	P
Output'	7	(68,14)	gp	P'

NOTES

This is a classic CMOS 'D' Latch. The input is the previous microprogram address, sampled in phase1, and the output of the latch is stable when the phase2 feedback switch closes.

operating voltages if voltages applied to input or output pins cause forward-biasing of pn junctions within the chip. The resulting high currents may cause permanent damage to the circuits.

The obvious solution to latch-up problems is to prevent junctions from ever becoming forward-biased, and to limit externally applied voltages at levels safely below VC. In practice this proves to be virtually impossible, due to the many spurious signals that occasionally occur in digital systems. Practical solutions to the latch-up problem is to reduce bipolar tansistor current gain, reduce the values of R1 and R2 , and connect p-well to ground and n-substrate to +V at regular intervals. Well contacts [9] at regular intervals reduce the current path for majority carriers in the substrate, thus reducing the IR product (where I is the current and R is the resistance). This results in reduced bias voltage for the bipolar transistors, which turns them off. Circuits connected to input and output pins are most critical with respect to latch-up. When power is switched on and off, voltages applied to the pins of a chip frequently go outside the normal range. This can cause latch-up.

Care must be taken to provide well contacts for each cell. However, if the cell is small (less than 40 lambda [9]), it should at least be designed in such a way so as to share the well contacts of the neighboring cells.

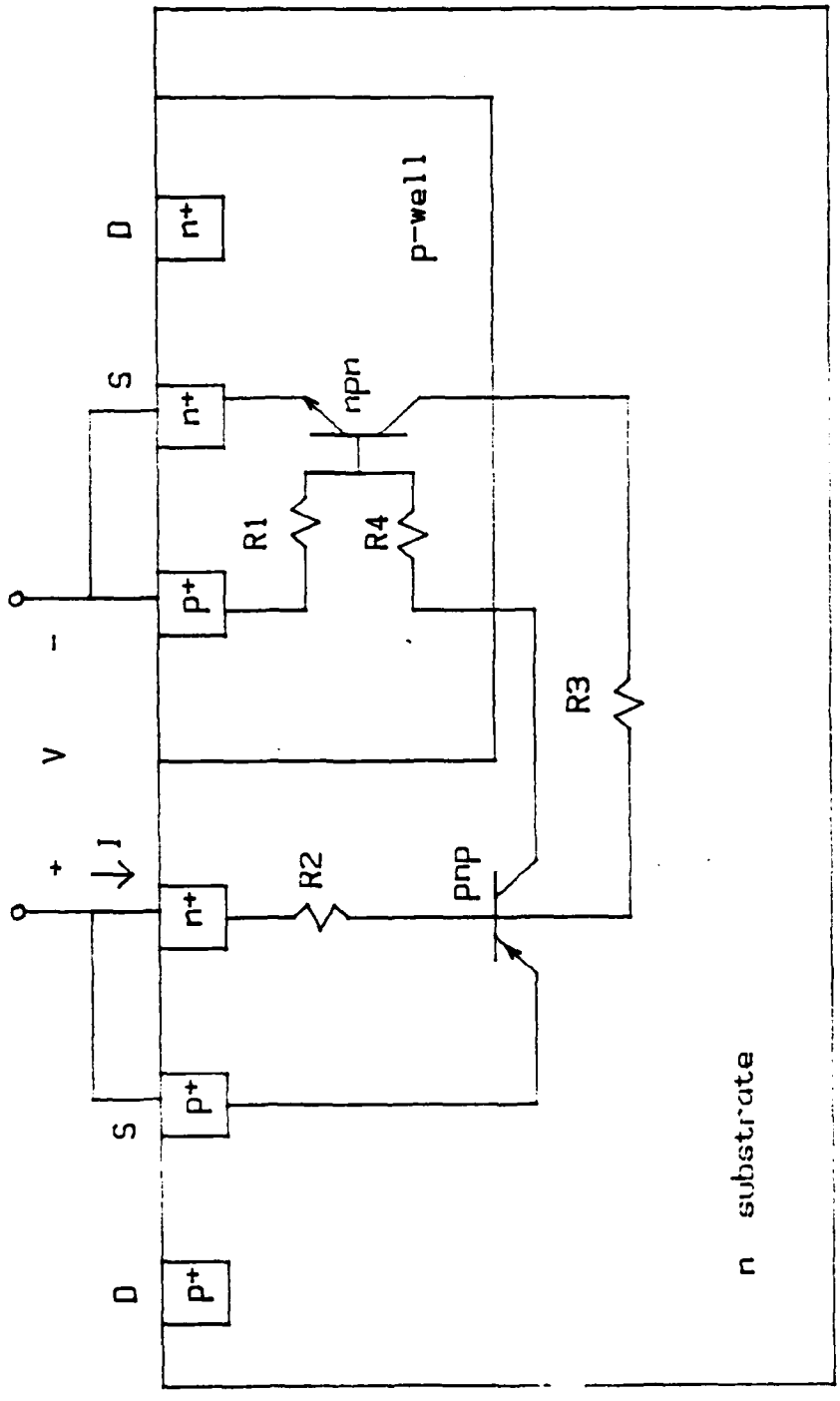


Figure C.a. CMOS Latch-up

both bits (except for the INC LOGIC LSB); however, the truth-table and logic equation(s) sections are included only for one bit of the circuit. X* denotes the previous state of the data X. A '.' between two operands in the logic equation(s) section indicates a logical 'AND' operation and a '+' indicates a logical 'OR' operation. The layouts should be designed in such a way as to avoid the latch-up [2] problem as explained below.

C.a. CMOS LATCH-UP

A classic problem with CMOS integrated circuits is the undesired and potentially troublesome parasitic bipolar transistors which will conduct if one or more of the pn junctions become forward-biased. Refer to FIG.C.a which makes it clear that a pnp transistor is possible with the n-type body as its base, while an npn transistor is possible with an n+ source or drain electrode as its emitter, the p-well as its base, and the n-type substrate as its collector. The parasitic resistors R1 and R2 originate in the bulk semiconductor material of the body and the p-type well. Low values of resistance are desirable in order to make it more difficult to forward-bias junctions.

Above some critical voltage (forward-biased emitter-base junction voltage) VC, both bipolar transistors begin to conduct and the current rises sharply from leakage levels (less than 1 microampere) to a value limited by resistors R3 and R4, often many milliamperes. This phenomenon is known as LATCH-UP. It can occur even at normal

APPENDIX C

CELL LIBRARY

The cell library documents the layouts of all the cells used in this design. It provides the truth-table, logic equation(s), terminal information, and notes section for each layout followed by the logic diagram and circuit schematic for the corresponding layout. Finally, the layout is included with terminal numbers marked on it and the information pertaining to these numbers is obtained in the terminal information section. Circles on the layouts correspond to the exact location of that particular terminal and a triangle indicates that it is the reference point for the layout. The design rules used in this project are scalable and the basic unit of length measurement is called lambda.

The aspect ratio (W/L) of a device is 2:1 unless otherwise specified. L is equal to 1.2 microns or 3 microns, depending upon the process of fabricating the chip. Note that for simplicity the truth-tables do not contain the states that are not allowed. The four main layers [5] used for interconnections are the first metal (fm), the second metal (sm), the gate poly (gp), and the interconnect poly (ip). The terminal information section includes the name of the terminal, the terminal number, connecting media (specified by the layer name) and the abbreviation for the terminal name. The layout for the carry chain includes two bits of the carry chain circuit but the logic is same for

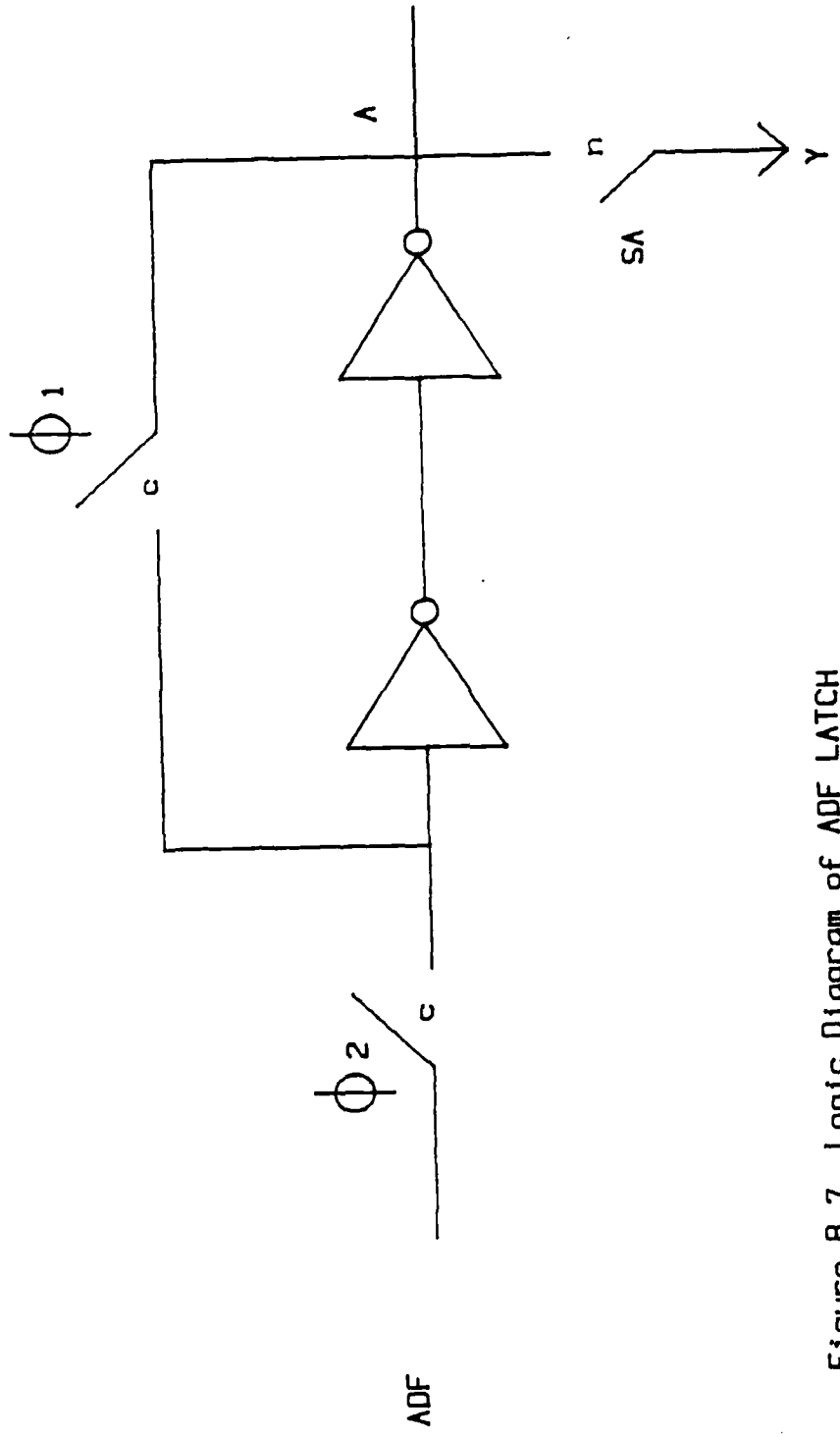


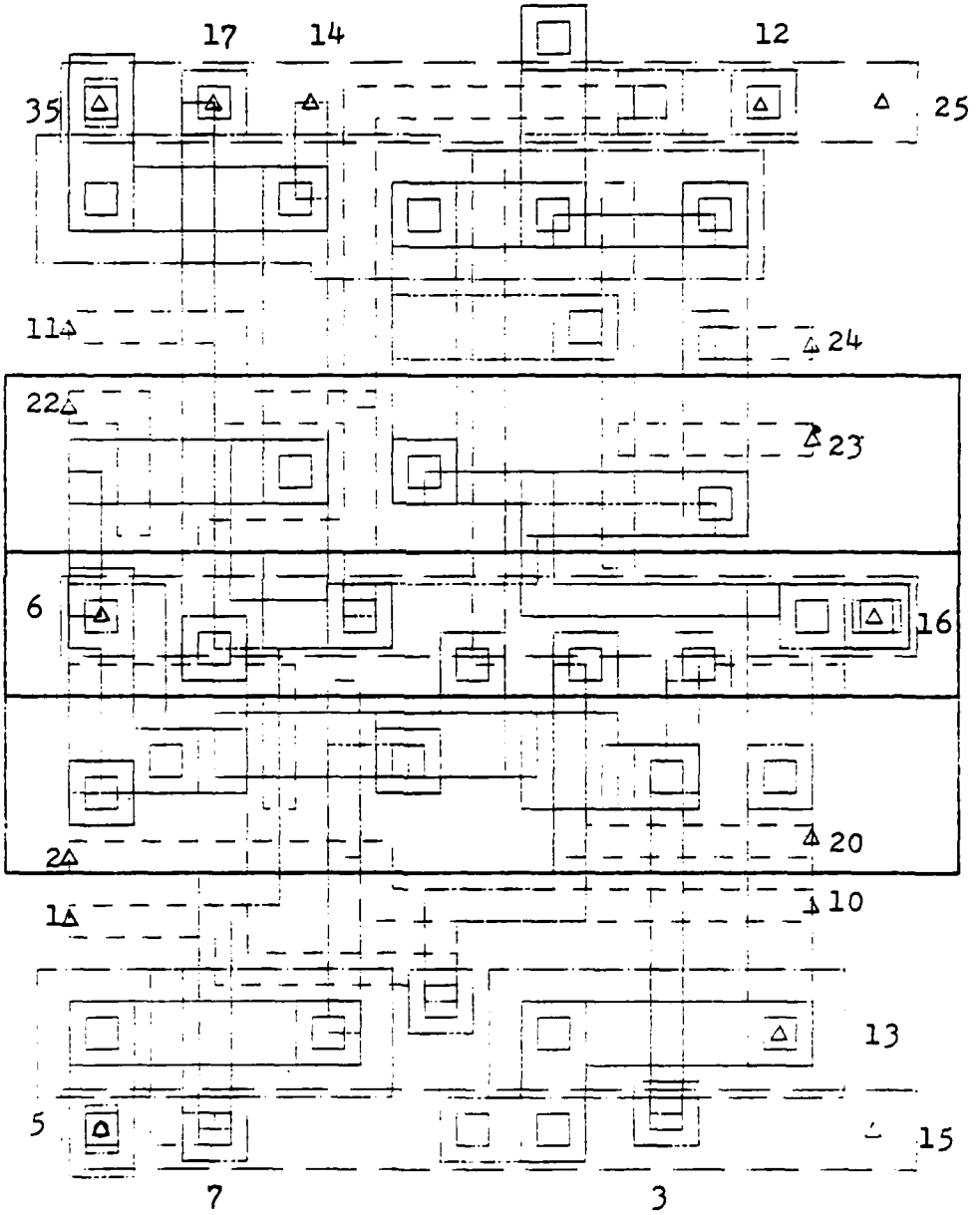
Figure B.7. Logic Diagram of ADF LATCH

B.7. ADF LATCHES

Refer to Figure B.7 for the logic diagram.

The ADF field of the microinstruction could be used as an explicit address source or as a count for looping. It is sampled in phase2 and available as stable output in phase1.

The 12-bit ADF Latches output (A) is used as an address source for MUX1. The 8 least significant bits of 'A' are used as a count (N). The schematic has not been included since it consists of inverters and switches whose schematics have been shown previously with respect to other macros.



scale 2300X / INC CAR CHAIN 2

SCALABLE CMOS CELL FAMILY
NO. : C.3
TITLE : EX-OR
HEIGHT : 32 WIDTH :25
DATE: 3/25/85
BY : GOWNI

TRUTH TABLE

P	CIN	X
0	0	0
0	1	1
1	0	1
1	1	0

LOGIC EQUATION(S)

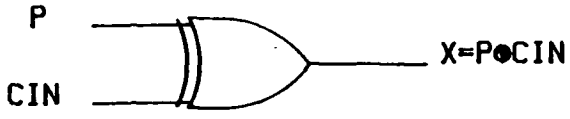
X = P.CIN' + CIN.P'

TERMINAL INFORMATION

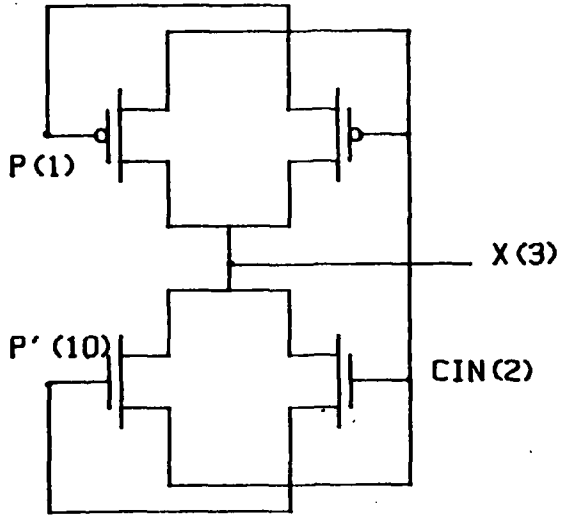
Name	No.	Location	Layer	Abbr.
Pass	1	(-2,23)	gp	P
Pass'	10	(-2,11)	gp	P'
Voltage Supply	4,14	(0,32) (25,32)	(sm)(fm,sm)	VDD
carry-in	2	(0,17)	fm	CIN
Output	3	(27,15)	fm	X
Ground	5,15	(0,0) (25,0)	(fm,sm)(sm)	GND

NOTES

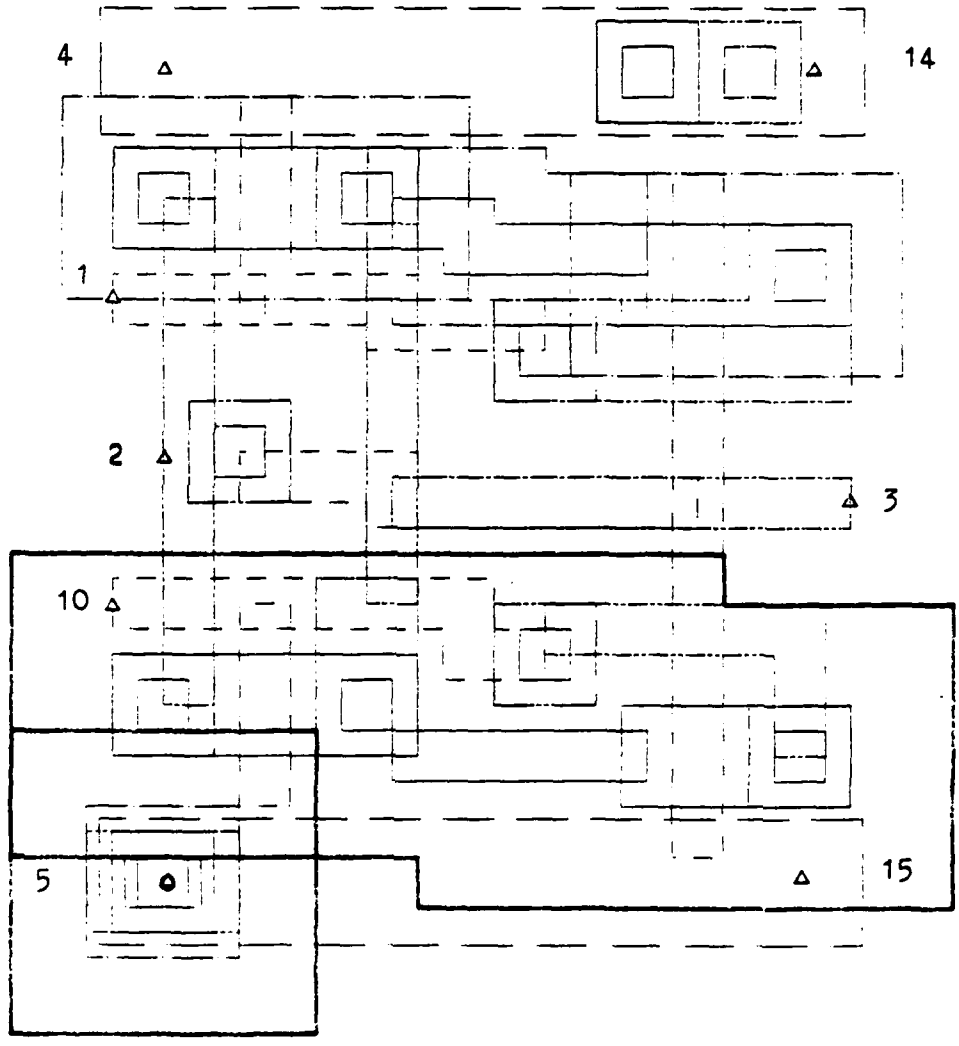
X is the exclusive-OR of P and CIN. Note that the EX-OR is evaluated in a different way with the carry-chain which has its CIN buffered. Note that the logic is same in both cases but the interconnections (between the carry-chain and the EX-OR circuits) are different. This is done in order to save layout area. Note that X is I for INC macro and D for DEC macro.



Logic Diagram of EX-OR



Circuit Schematic of EX-OR



scale. 3600X / EX-OP

SCALABLE CMOS CELL FAMILY
NO. : C.4
TITLE : INC O/P LATCH
HEIGHT : 32 WIDTH : 97
DATE: 3/25/85
BY : GOWNI

TRUTH TABLE

phase2	I	C	IOP	IOP	SA	Y
0	x	0	IOP*	x	0	Y*
0	x	1	0	0	1	0
1	0	0	0	1	1	1
1	1	0	1			

LOGIC EQUATION(S)

$$Y = SA'.Y^* + SA \,.\, IOP$$

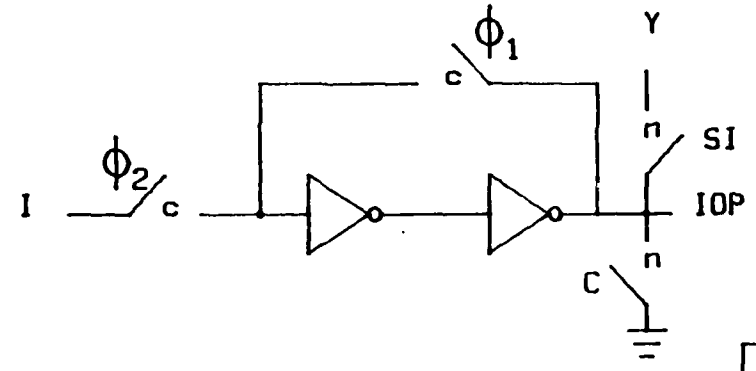
$$IOP = phase2'.C'.IOP^* + C' + phase2 \,.\, C'.I$$

TERMINAL INFORMATION

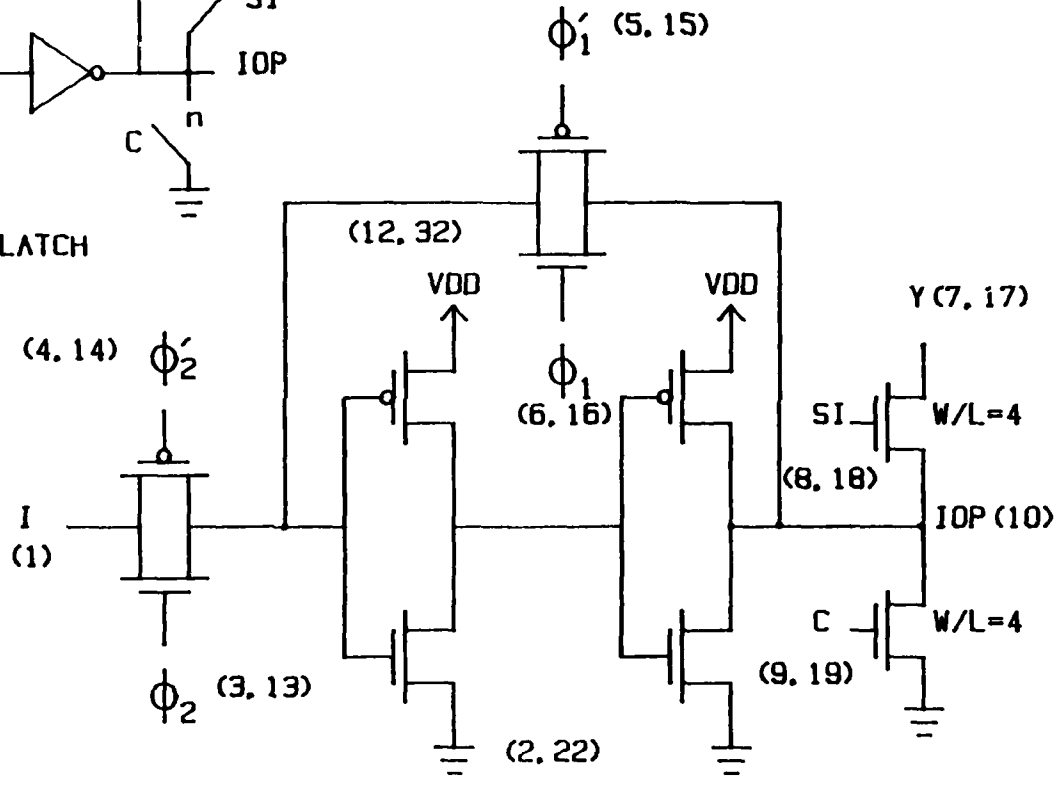
Name	No.	Location	Layer	Abbr.
Data input	1	(0,16)	fm	I
Ground	2,22	(0,0) (97,0)	sm	GND
Voltage Supply	12,32	(0,32) (97,32)	sm	VDD
Phase2	3,13	(7,0) (7,32)	(fm,ip,gp)(fm)	ϕ_2
Phase2'	4,14	(14,0) (14,32)	(fm)(fm,ip,gp)	ϕ_2'
Phase1'	5,15	(52,0) (52,32)	(fm)(fm,ip,gp)	ϕ_1
Phase1	6,16	(59,0) (59,13)	fm	ϕ_1'
Address	7,17	(0,14) (97,14)	sm	Y
Output	10	(94,20)	fm,ip,gp	IOP
Select IOP	8,18	(80,0) (80,32)	(fm,ip,gp)(fm)	SI
Clear/Increment	9,19	(87,0) (87,32)	(fm,ip,gp)(fm)	C

NOTES

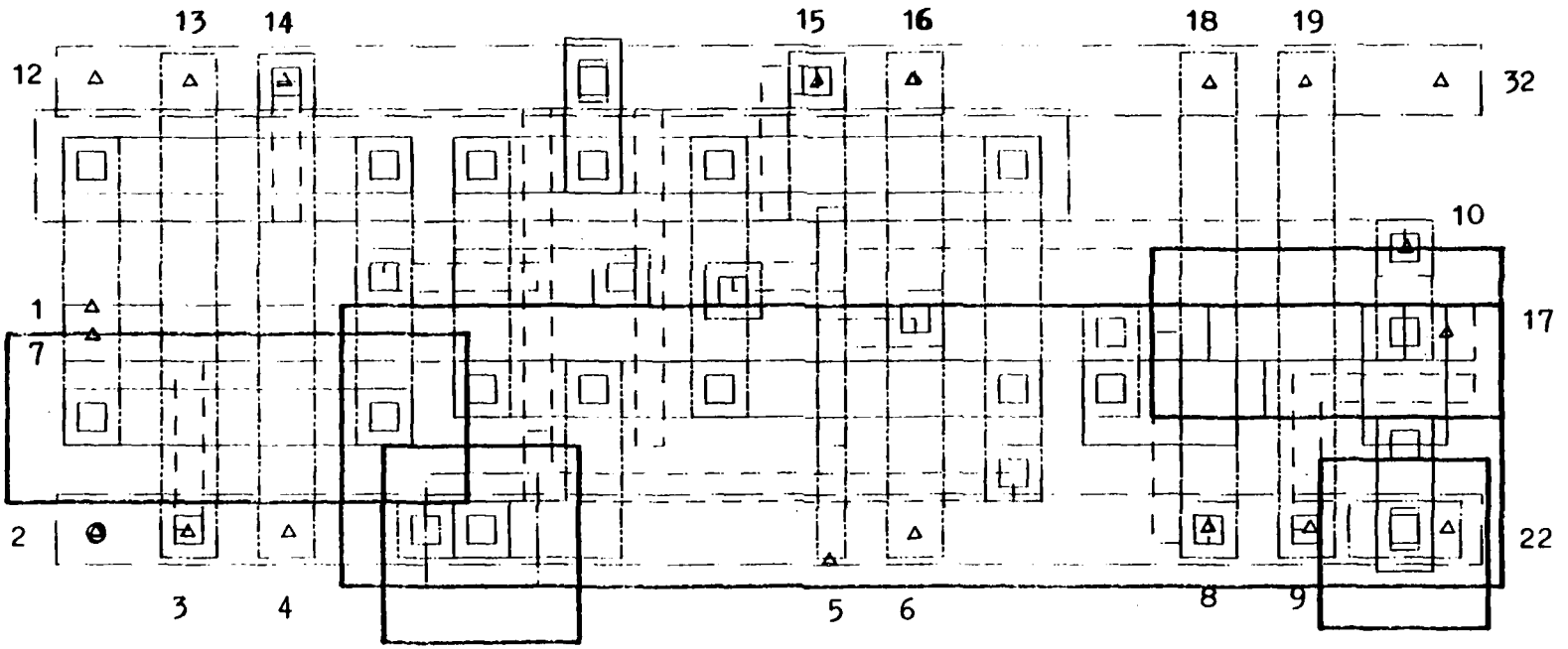
This is a classic CMOS 'D' Latch. The output of EX-OR in the INC LOGIC (I) is the input, sampled in phase2, and is stable when the phase2 feedback switch closes. The Clear/Increment (C) signal is a phase1 control from PLA. The output of the INC O/P LATCH (IOP) is cleared when C is high. The Select IOP (SI) is a control from PLA to MUX1 which is implemened as a distributed bus structure. When high selects the IOP as the microprogram address(Y). W/L ratio of the N-transistor, that clears the IOP, is 4:1 since it has to overcome the P-transistor of the second inverter stage of the INC O/P LATCH. C is 0 in phase2.



Logic Diagram of INC O/P LATCH



Circuit Schematic of INC O/P LATCH



scale. 2000X / INC O/P LATCH

SCALABLE CMOS CELL FAMILY | NO. : C.5 | TITLE : INC LOGIC LSB | HEIGHT : 32 WIDTH : 70 | DATE: 3/25/85 | BY : GOWNI

TRUTH TABLE

phase2	P0	COUT0	I0
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	0

LOGIC EQUATION(S)

COUT0 = phase2' + phase2.P0

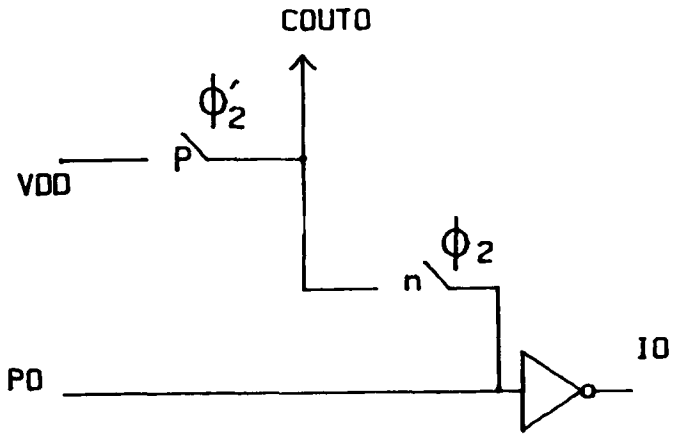
I0 = P0'

TERMINAL INFORMATION

Name	No.	Location	Layer	Abbr.
Pass0	1	(0,14)	fm,ip,gp	P0
Output0	2	(70,15)	fm	I0
Pass1	7	(-2,51)	gp	P1
Kill1	8	(-2,47)	gp	K1
Output1	9	(72,47)	fm	I1
Voltage Supply	5,15, 25,35	(0,0) (0,64) (70,0) (70,64)	sm (sm)(fm,sm)	VDD
Phase2	4,14	(7,0) (7,64)	(fm)(fm,ip,gp)	ϕ_2
Carry-out 1	3	(14,64)	fm	COUT1
Ground	6,16	(0,32) (70,32)	sm	GND

NOTES

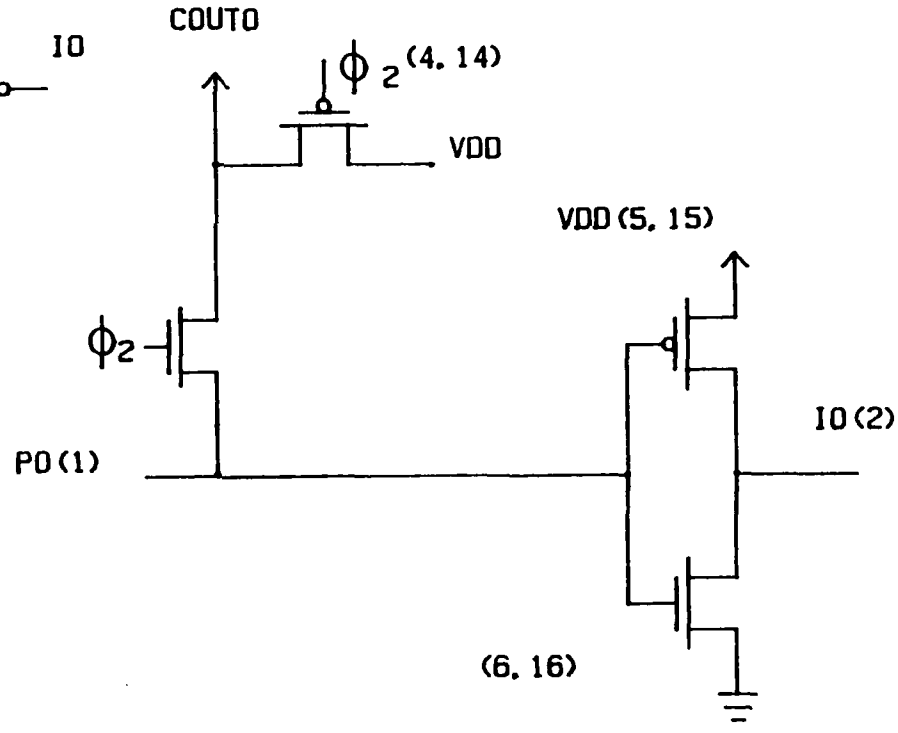
This is the least significant stage of the INC LOGIC in INC macro. The carry-out of this stage is the carry-in to the first stage. The layout shows the INC LOGIC LSB along with the first bit of the carry chain. Hence the terminal information section includes both the stages.

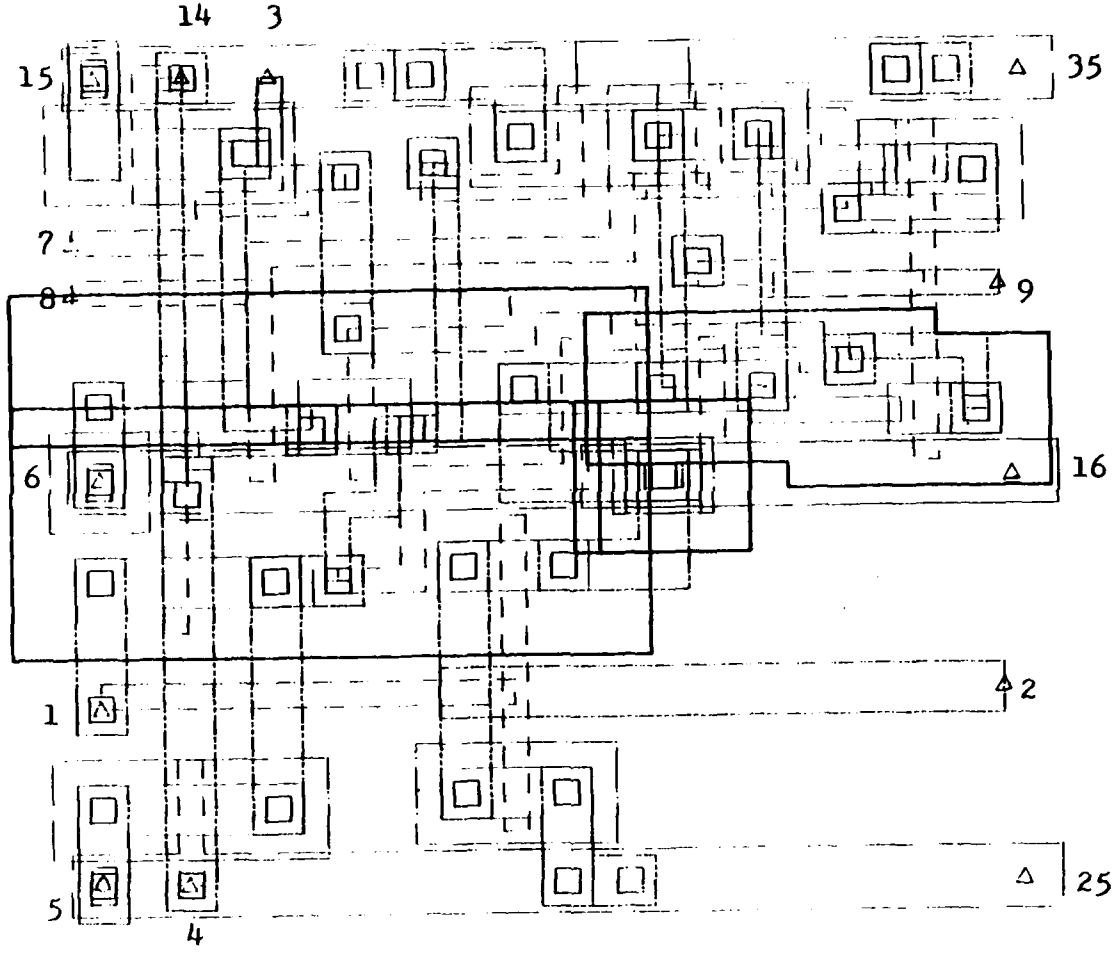


PO=IO

Logic Diagram of INC LOGIC LSB

Circuit Schematic of INC LOGIC LSB





scale: 1790X / INC LOGIC LSB

SCALABLE CMOS CELL FAMILY	NO. : C.6 TITLE : SS INTERFACE HEIGHT : 32 WIDTH : 50	DATE: 3/25/85 BY : GOWNI

TRUTH TABLE

ø1F	IOP	DF	F	SF	Y
0	x	DF*	x	0	Y*
1	0	0	0	1	0
1	1	1	1	1	1

LOGIC EQUATION(S)

Y = SF.F + SF'.Y*

DF = ø1F'. DF*

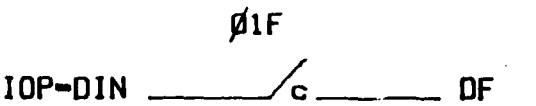
+ ø1F.IOP

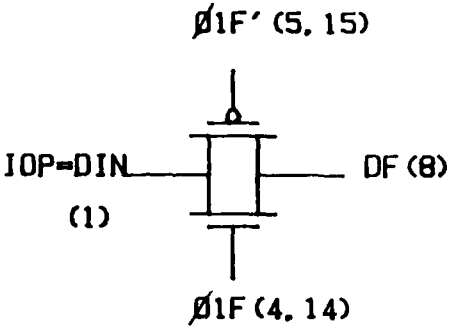
TERMINAL INFORMATION

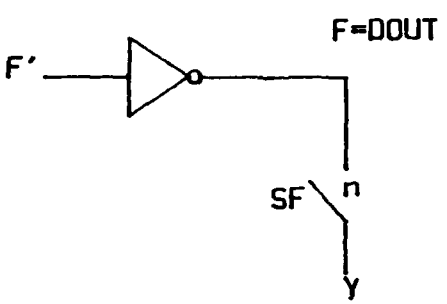
Name	No.	Location	Layer	Abbr.
Input	1	(-2,19)	fm	IOP
Ground	2,22	(0,0) (50,0)	sm	GND
Voltage Supply	6,16	(0,32) (50,32)	sm	VDD
Phase1 Forward	4,14	(36,0) (36,32)	(fm,ip,gp)(fm)	ø1F
Phase1 Forward'	5,15	(43,0) (43,32)	(fm)(fm,ip,gp)	ø1F'
Data to Stack top	8	(50,20)	fm,ip,gp	DF
Data from Stack top	7	(52,14)	gp	F
Select F	3,13	(7,0) (7,32)	(fm,ip,gp)(fm)	SF
Address	9,19	(0,14) (50,14)	(fm,sm)(sm)	Y

NOTES

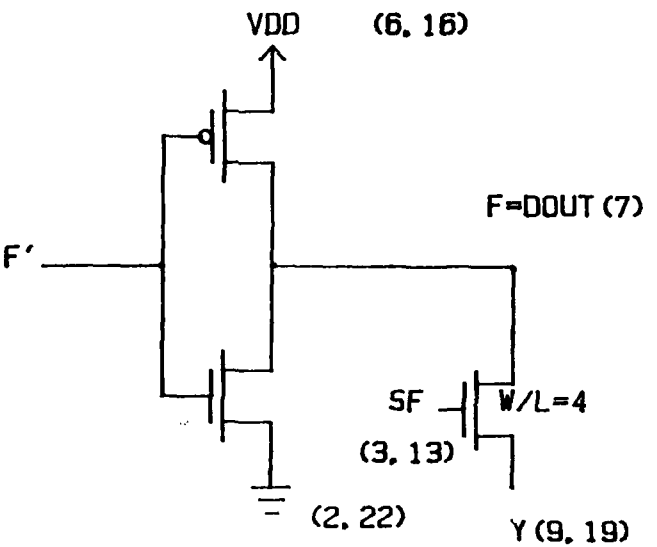
This is the interface between the INC and SS macros.The input is the incremented output (IOP) if it is a PUSH operation or stack top (F) if it is a POP operation. The Select F (SF) is a control from the PLA to MUX1, which when high, selects the stack top (F) as the microprogram address.



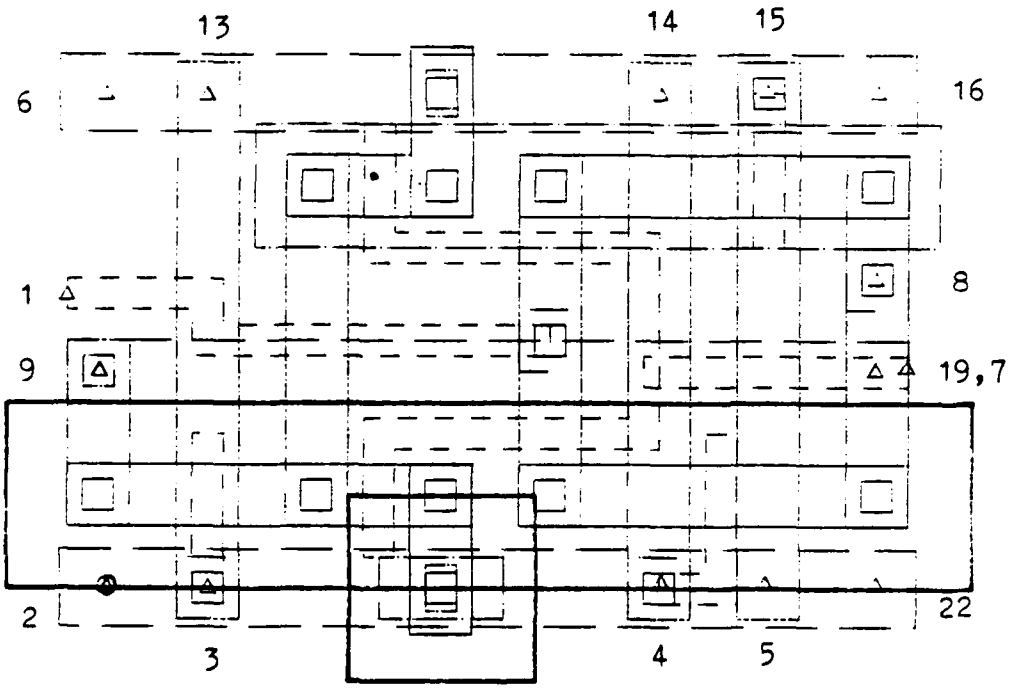




Logic Diagram of SS Interface



Circuit Schematic of SS Interface



scale 2200X / SS INTERFACE

SCALABLE CMOS CELL FAMILY
NO. : C.7
TITLE : PD STACK (1/2_)
HEIGHT : 32 WIDTH : 42
DATE: 3/25/85
BY : GOWNI

TRUTH TABLE

(DF + DB)	∅2B	∅2F	QB	QF
x	0	0	QB*	QF*
x	1	1	not allowed	
0	1	0	0	QF*
1	1	0	1	QF*
0	1	1	QB*	0
1	0	1	QB*	1

LOGIC EQUATION(S)

QB = ∅2B'.QB*
+ ∅2B(DF+DB)

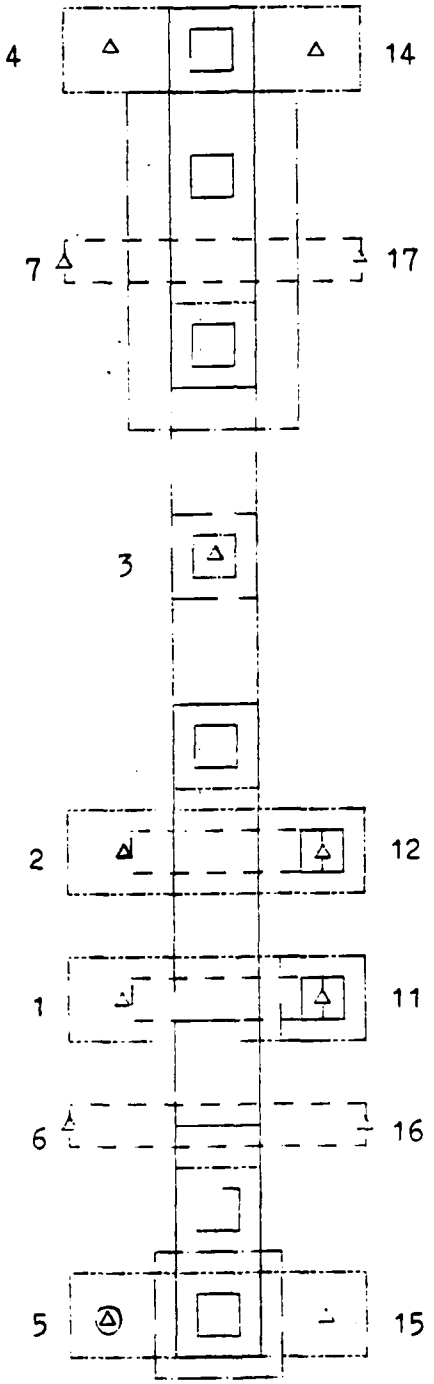
QF = ∅2F'. QF*
+ ∅2F.(DF+DB)

TERMINAL INFORMATION

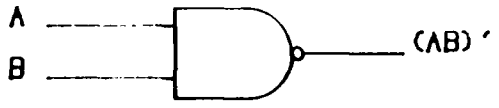
Name	No.	Location	Layer	Abbr.
Input from the left half	1	(-2,19)	gp	DF
Input from the right half	6	(44,14)	gp	DB
Output to the left half	9	(0,13)	fm,ip,gp	QB
Output to the right half	8	(42,20)	fm,ip,gp	QF
Phase2B (Phase1B)	4,14	(14,0) (14,32)	(fm,ip,gp)(fm)	∅2B
Phase2B' (Phase1B')	3,13	(7,0) (7,32)	(fm)(fm,ip,gp)	∅2B'
Phase2F (Phase1F)	5,15	(28,0) (28,32)	(fm,ip,gp)(fm)	∅2F
Phase2F' (Phase1F')	6,16	(35,0) (35,32)	(fm)(fm,ip,gp)	∅2F'
Voltage Supply	12,32	(0,32) (42,32)	sm	VDD
Ground	2,22	(0,0) (42,0)	sm	GND

NOTES

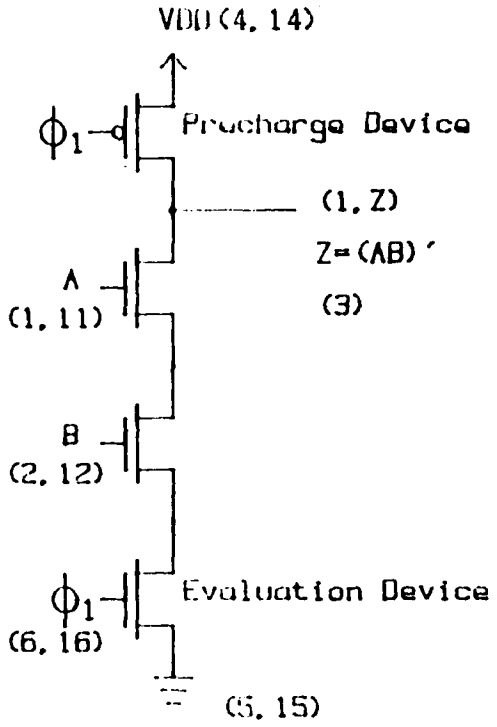
This is half-word of a push-down stack. Names within the parantheses apply to the next half-word of the stack.



scale 3000X / AND



Logic Diagram of AND



Circuit Schematic of AND

SCALABLE CMOS CELL FAMILY	NO. : C.11 TITLE : AND HEIGHT : 60 WIDTH : 10	DATE: 3/25/85 BY : GOWNI

TRUTH TABLE

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

LOGIC EQUATION(S)

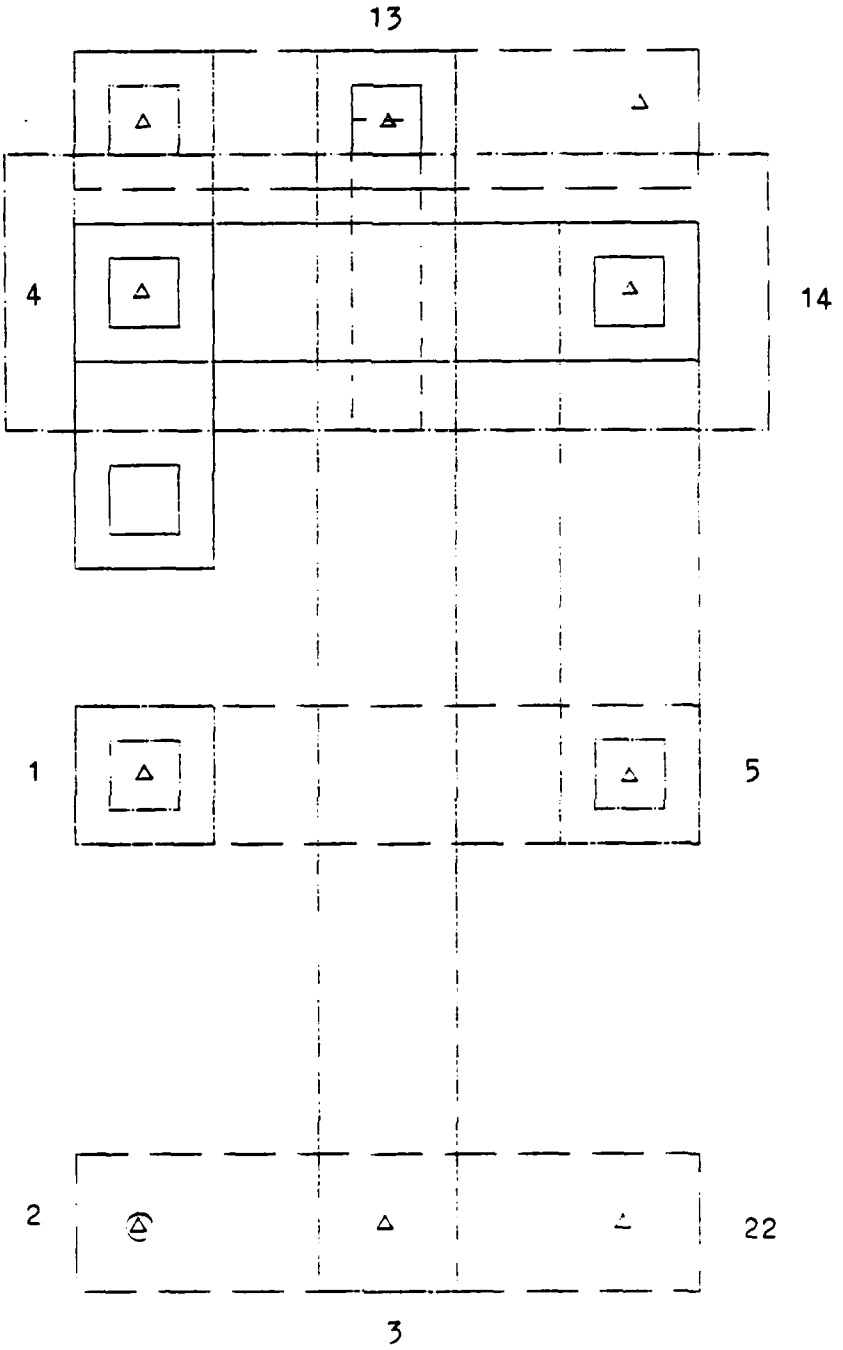
$Z = (A.B)'$

TERMINAL INFORMATION

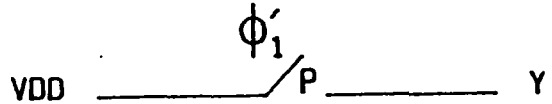
Name	No.	Location	Layer	Abbr.
AND Plane input A	1,11	(10,15) (0,15)	(fm,ip,gp)(fm)	A
AND Plane input B	2,12	(10,22) (0,22)	(fm,ip,gp)(fm)	B
Output	3	(5,36)	fm,sm	Z
Phase1	6,16	(-2,9) (12,9)	gp	ϕ_1
	7,17	(-2,50) (12,50)	gp	
Voltage supply	4,14	(0,60) (10,60)	fm	VDD
Ground	5,15	(0,0) (10,0)	fm	GND

NOTES

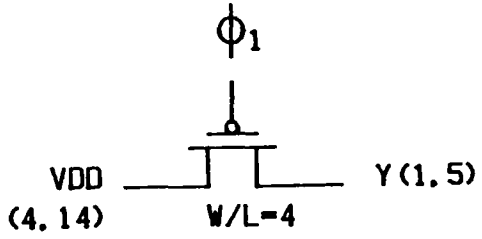
The actual AND-plane in the PLA consists of twelve inputs and eighteen outputs. The AND-plane is n-type and programmed in poly. Well contacts are not shown in this circuit but they are added at regular intervals in the total AND-plane.



scale. 4900X / BUS PRE



Logic Diagram of BUS PRE



Circuit Schematic of BUS PRE

SCALABLE CMOS	NO. : C.10	DATE: 3/25/85
CELL FAMILY	TITLE : BUS PRECHARGE	BY : GOWNI
	HEIGHT : 32 WIDTH : 14	

TRUTH TABLE

phase1	Y
0	1
1	(A+I+F+M)

LOGIC EQUATION(S)

Y = phase1'

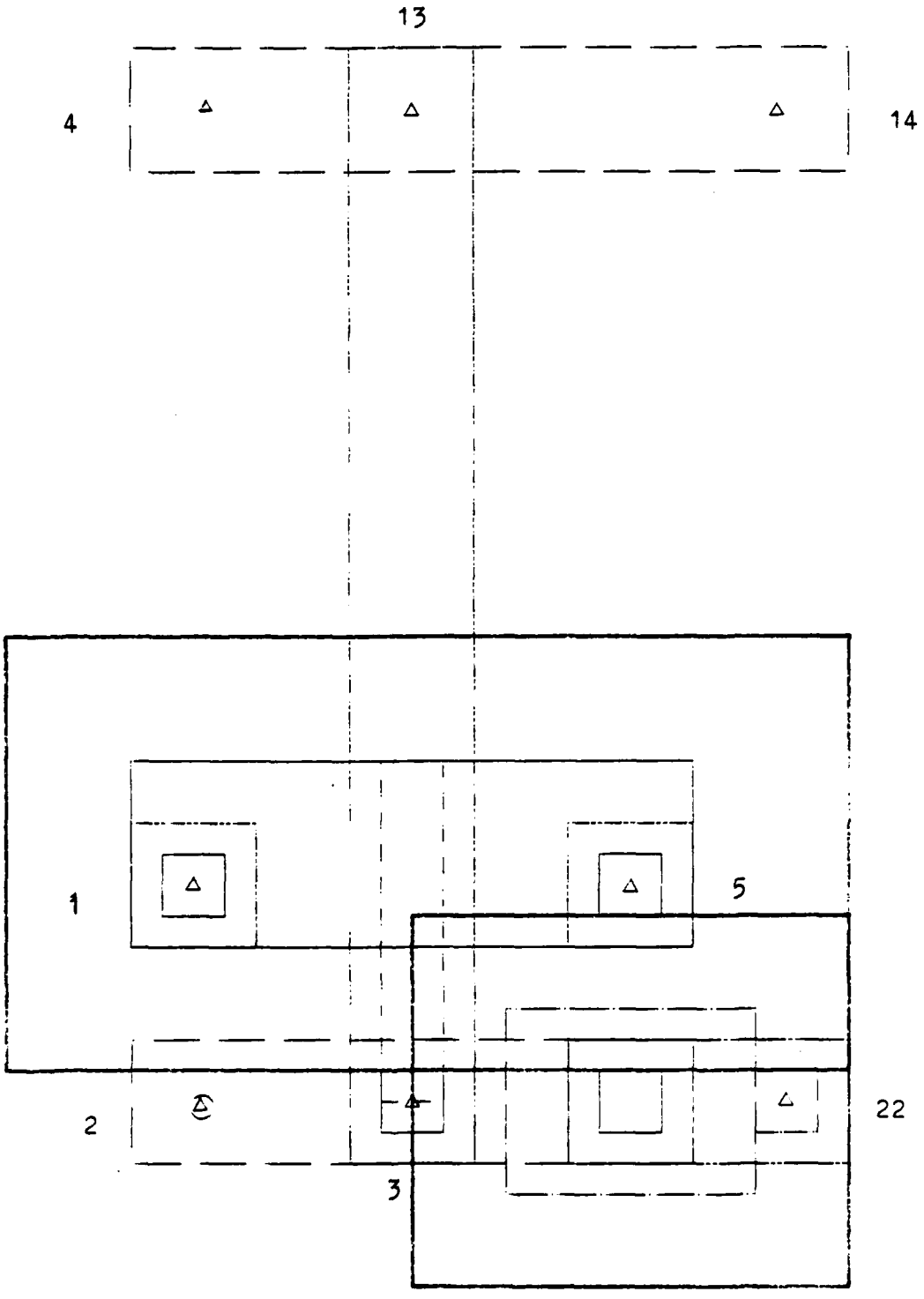
+ phase1.(A+I+F+M)

TERMINAL INFORMATION

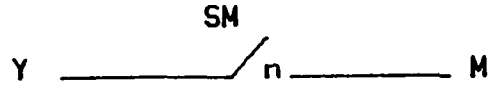
Name	No.	Location	Layer	Abbr.
Address	1,5	(0,13) (14,13)	fm,sm	Y
Phase1	3,13	(7,0) (7,32)	(fm)(fm,ip,gp)	ϕ_1
Voltage Supply	4,14	(0,32) (14,32)	(fm,sm)(sm)	VDD
Ground	2,22	(0,0) (14,0)	sm	GND

NOTES

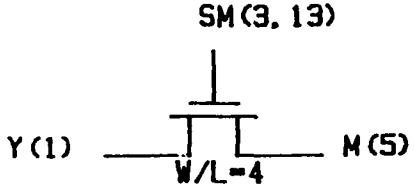
MUX1 is implemented as a distributed bus structure. The output of MUX1, which is the microprogram address Y, runs horizontally as a bus across the microsequencer chip. This address bus (Y) is precharged high in phase1' and conditionally discharged in phase1. The BUS PRE cell is used to precharge the address bus. The W/L ratio of the P-transistor is 4:1 since it has to precharge a large capacitance on the bus.



scale. 5000X / MAP



Logic Diagram of MAP



Circuit Schematic of MAP

SCALABLE CMOS CELL FAMILY
NO. : C.9
TITLE : MAP
HEIGHT : 32 WIDTH : 19
DATE: 3/25/85
BY : GOWNI

TRUTH TABLE

M	SM	Y
x	0	Y*
0	1	0
1	1	0

LOGIC EQUATION(S)

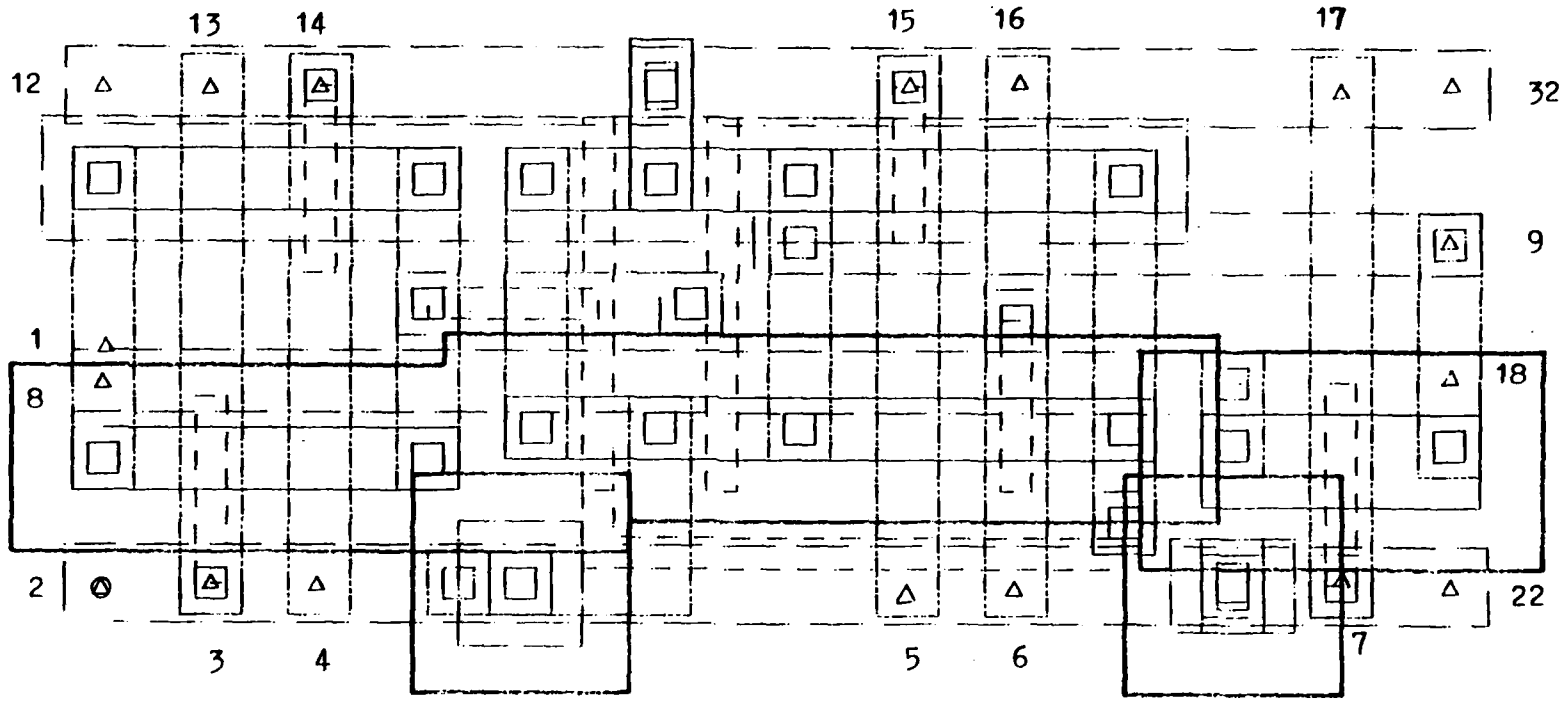
Y = SM'.Y*
+ SM.M

TERMINAL INFORMATION

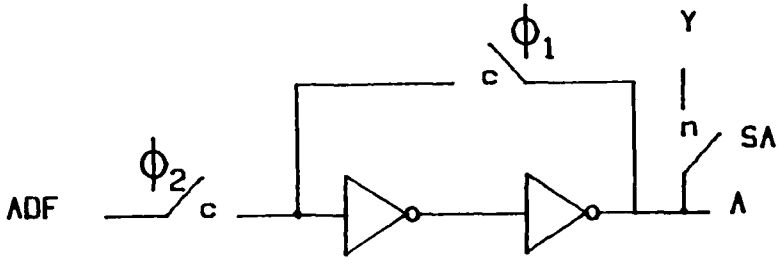
Name	No.	Location	Layer	Abbr.
Address	1	(0,7)	fm	Y
Mapped output	5	(14,7)	fm	M
Select M	3,13	(7,0) (7,32)	(fm,ip,gp)(fm)	S
Voltage Supply	4,14	(0,32) (19,32)	sm	VDD
Ground	2,22	(0,0) (19,0)	(sm)(fm,sm)	GND

NOTES

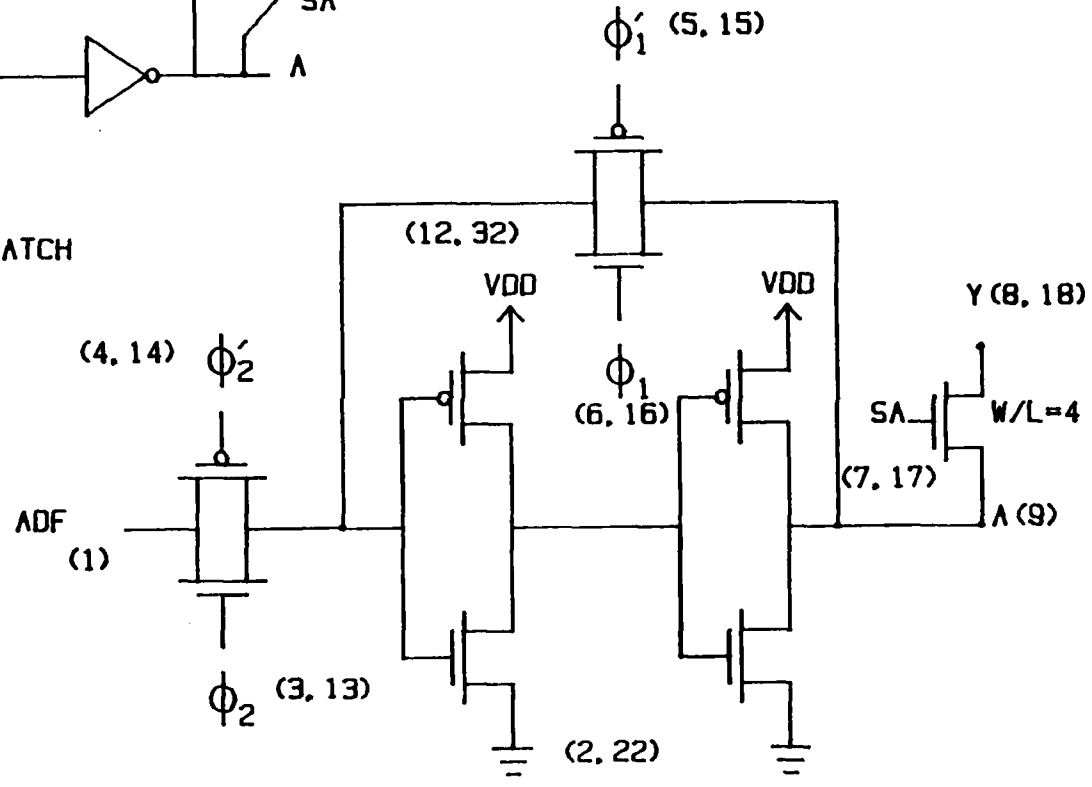
The 5 least significant bits of the mapped output (M) are the op-code bits and the 7 most significant bits are cleared to zero. The Select M (SM) is a control from the PLA to MUX1 in phase1 which, when high, selects the mapped output as the microprogram address (Y). SM is 0 in phase1'.



scale. 2200X / ADF LATCH



Logic Diagram of ADF LATCH



Circuit Schematic of ADF LATCH

SCALABLE CMOS CELL FAMILY
NO. : C.8
TITLE : ADF LATCH
HEIGHT : 32 WIDTH : 87
DATE: 3/25/85
BY : GOWNI

TRUTH TABLE

phase2	ADF	A	SA	A	Y
0	x	A*	0	x	Y*
1	0	0	1	0	0
1	1	1	1	1	1

LOGIC EQUATION(S)

A = phase2'.A*
+ phase2.ADF

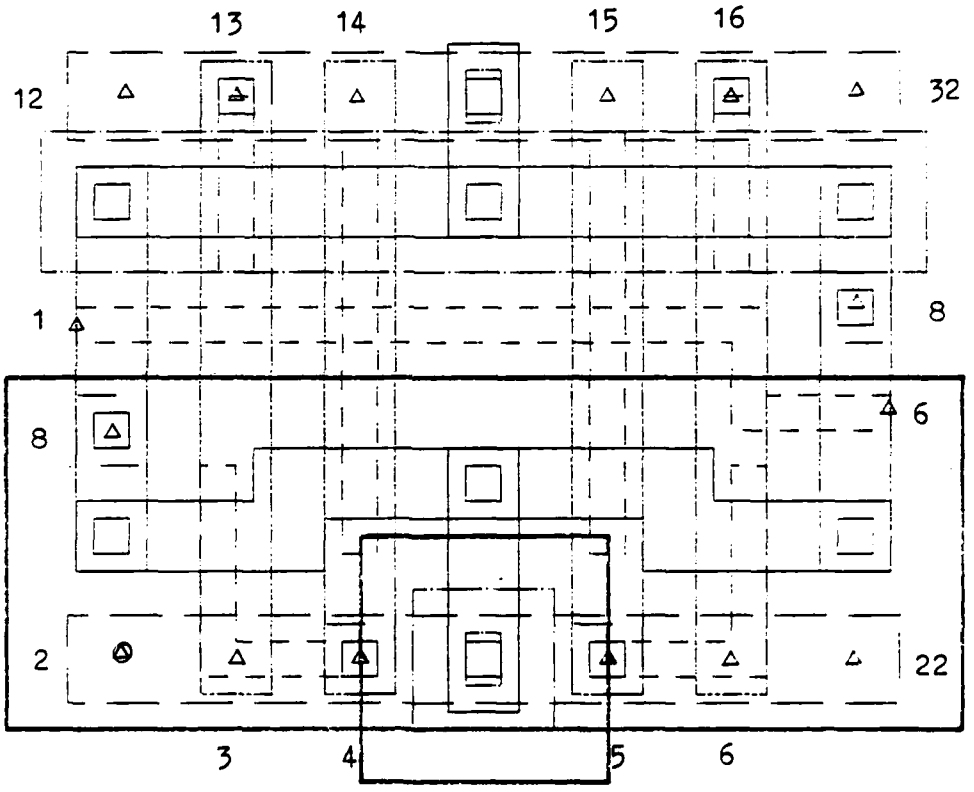
Y= SA'.Y*
+ SA.Y

TERMINAL INFORMATION

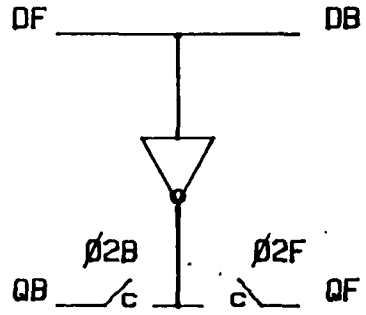
Name	No.	Location	Layer	Abbr.
Data input	1	(0,16)	fm	ADF
Ground	2,22	(0,0) (87,0)	sm	GND
Voltage Supply	12,32	(0,32) (87,32)	sm	VDD
Phase1	6,16	(59,0) (59,32)	fm	ϕ_1
Phase1'	5,15	(52,0) (52,32)	(fm)(fm,ip,gp)	ϕ_1'
Phase2	3,13	(7,0) (7,32)	(fm,ip,gp)(fm)	ϕ_2
Phase2'	4,14	(14,0) (14,32)	(fm)(fm,ip,gp)	ϕ_2'
Address	8,18	(0,13) (87,13)	sm	Y
Output	9	(87,22)	sm,fm	A
Select A	7,17	(80,0) (80,32)	(fm,ip,gp)(fm)	SA

NOTES

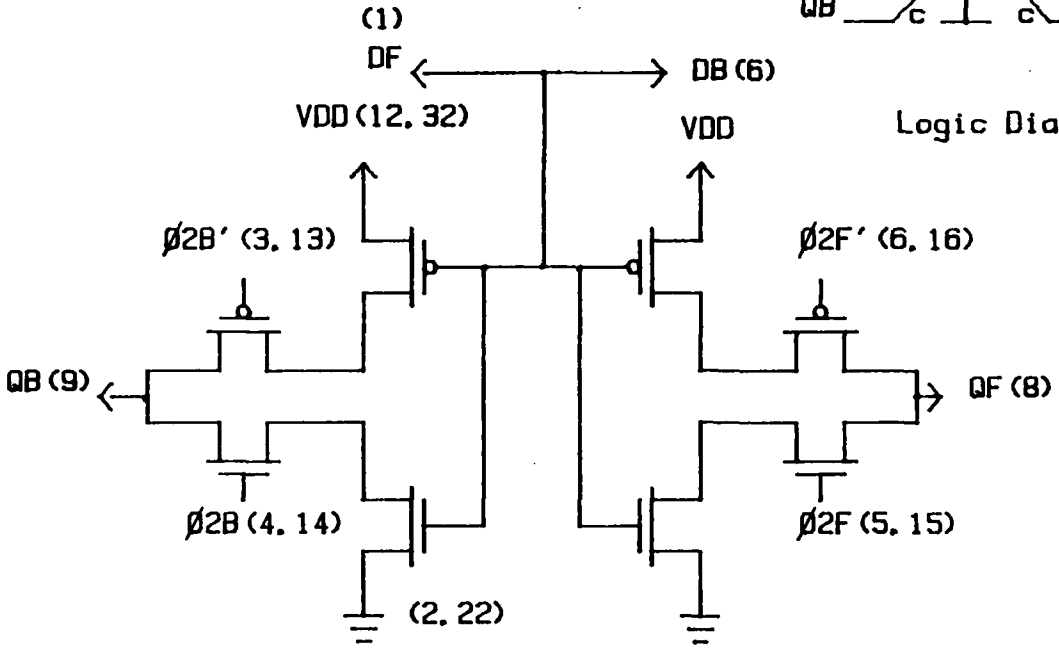
This is a classic CMOS 'D' Latch. The input is the ADF field of the previous microinstruction (ADF), sampled in phase2, and becomes a stable output (A) when the phase1 feedback switch closes. The Se'ect A (SA) is a control from the PLA to MUX1 in phase1 which, when high, selects the ADF Latches output (A) as the microprogram address.



scale. 2500X / PD STACK (1/2)



Logic Diagram of PD Stack(1/2)



Circuit Schematic of PD Stack(1/2)

SCALABLE CMOS CELL FAMILY
NO. : C.12
TITLE : OR
HEIGHT : 39 WIDTH : 8
DATE: 3/25/85
BY : GOWNI

TRUTH TABLE

X	Y	S
0	0	1
0	1	1
1	0	1
1	1	0

LOGIC EQUATION(S)

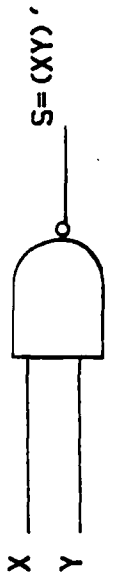
S = (X.Y)'

TERMINAL INFORMATION

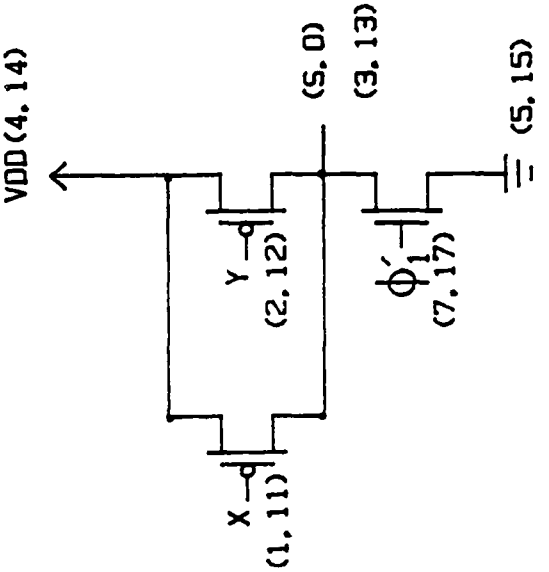
Name	No.	Location	Layer	Abbr.
OR Plane input X	1,11	(-4,27) (12,27)	gp	X
OR Plane input Y	2,12	(-4,35) (12,35)	gp	Y
Output	3,13	(0,0) (0,39)	fm	S
Phase1'	7,17	(0,14) (9,14)	gp	
Voltage supply	4,14	(8,23) (8,39)	(sm)(fm,sm)	VDD
Ground	5,15	(0,0) (10,0)	(sm)(fm,sm)	GND

NOTES

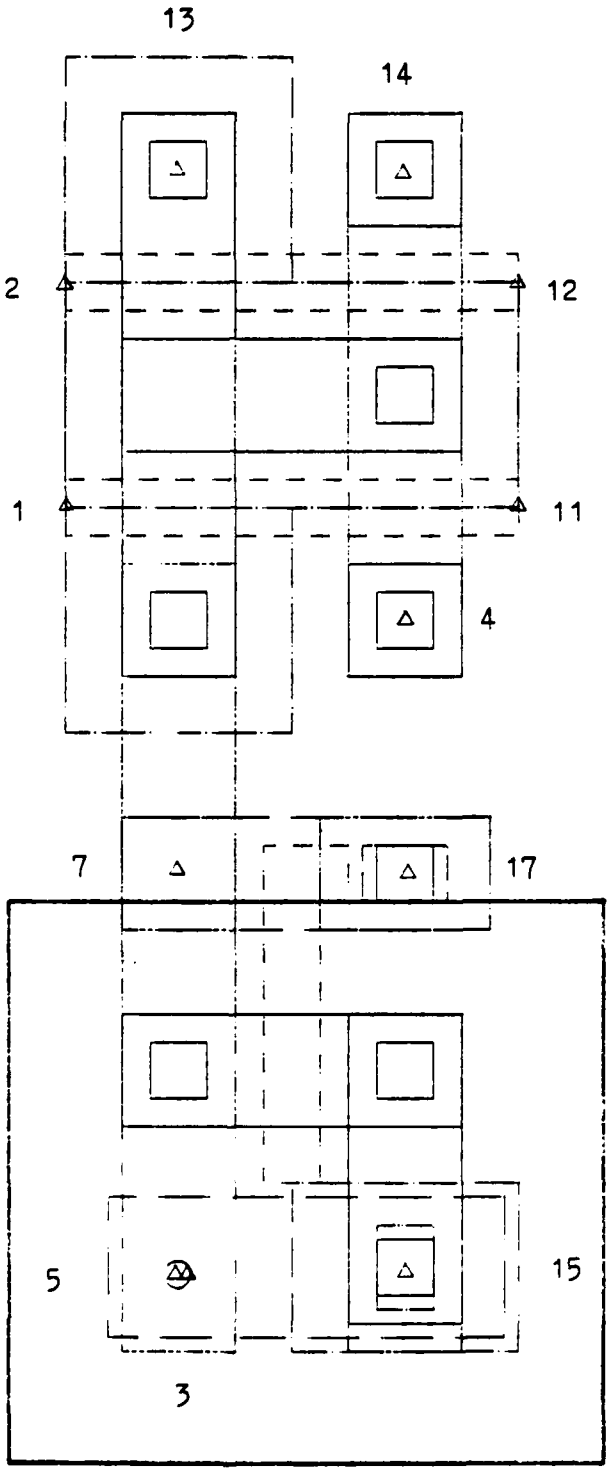
The power supply runs in first metal between two output lines spaced 16 lambda apart. The actual OR-plane has 18 inputs and 17 outputs. The layout shows only one output line. The OR-plane is p-type and programmed in active.



Logic Diagram of OR



Circuit Schematic of OR



scale. 4000X / OR

SCALABLE CMOS CELL FAMILY

NO. : C.13
TITLE : PLA I/P LATCH
HEIGHT : 37 WIDTH : 63

DATE: 3/25/85
BY : GOWNI

TRUTH TABLE

phase2	D	Q
0	x	Q*
1	0	0
1	1	1

LOGIC EQUATION(S)

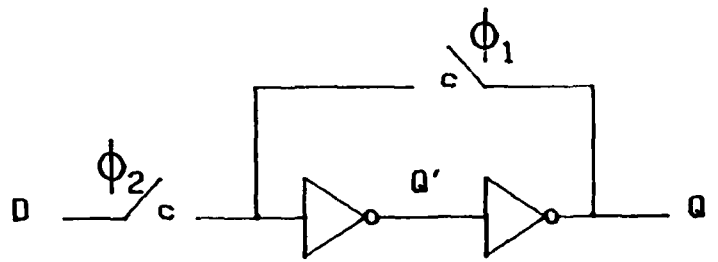
Q = phase2'.Q*
+ phase2.Q

TERMINAL INFORMATION

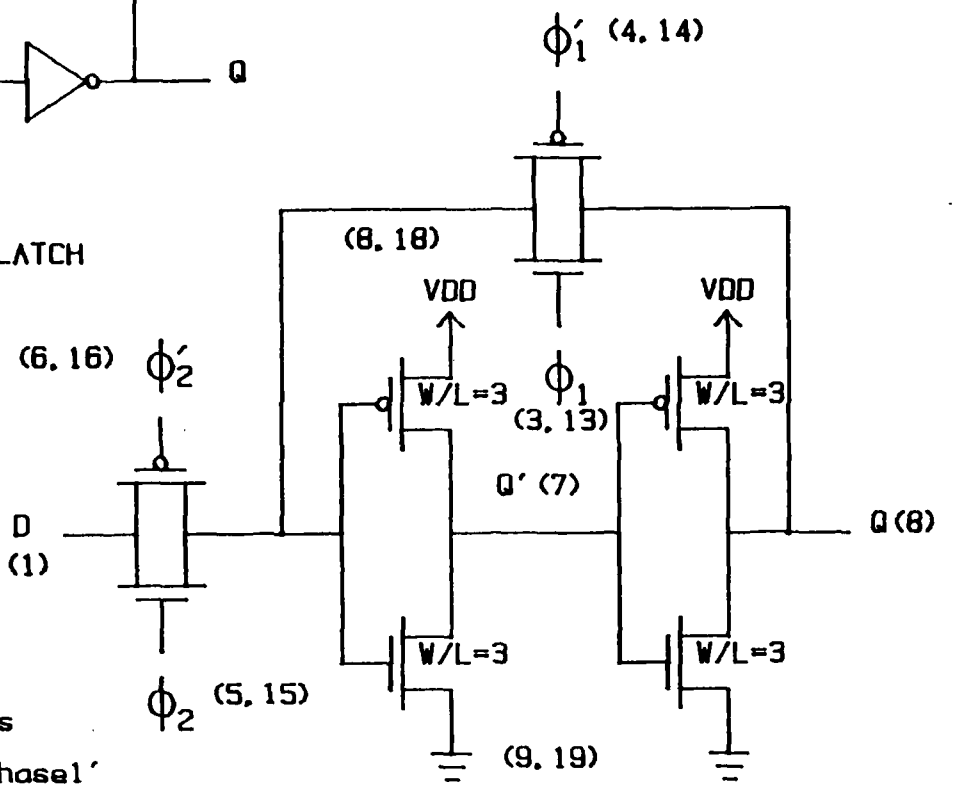
Name	No.	Location	Layer	Abbr.
Data input	1	(63,18)	fm	D
Ground	9,19	(0,0) (63,0)	sm	GND
Voltage Supply	8,18	(0,37) (63,37)	sm	VDD
Phase1	3,13	(7,0) (7,37)	fm	ϕ_1
Phase1'	4,14	(14,0) (14,37)	(fm)(fm,ip,gp)	ϕ_1'
Phase2	5,15	(51,0) (52,39)	(fm,ip,gp)(fm)	ϕ_2
Phase2'	6,16	(57,-2)(58,37)	(fm)(fm,ip,gp)	ϕ_2'
Output	8	(-2,23)	gp	Q
Output'	7	(-2,27)	gp	Q'

NOTES

This is a CMOS 'D' Latch. The input can be one of the four BR field bits or the test flag T. The input is sampled in phase2 and is stable in phase1 when the feedback switch closes. Sampling time is phase2'*phase1' for T input. The devices in the inverters of the latch have W/L ratios of 3:1.

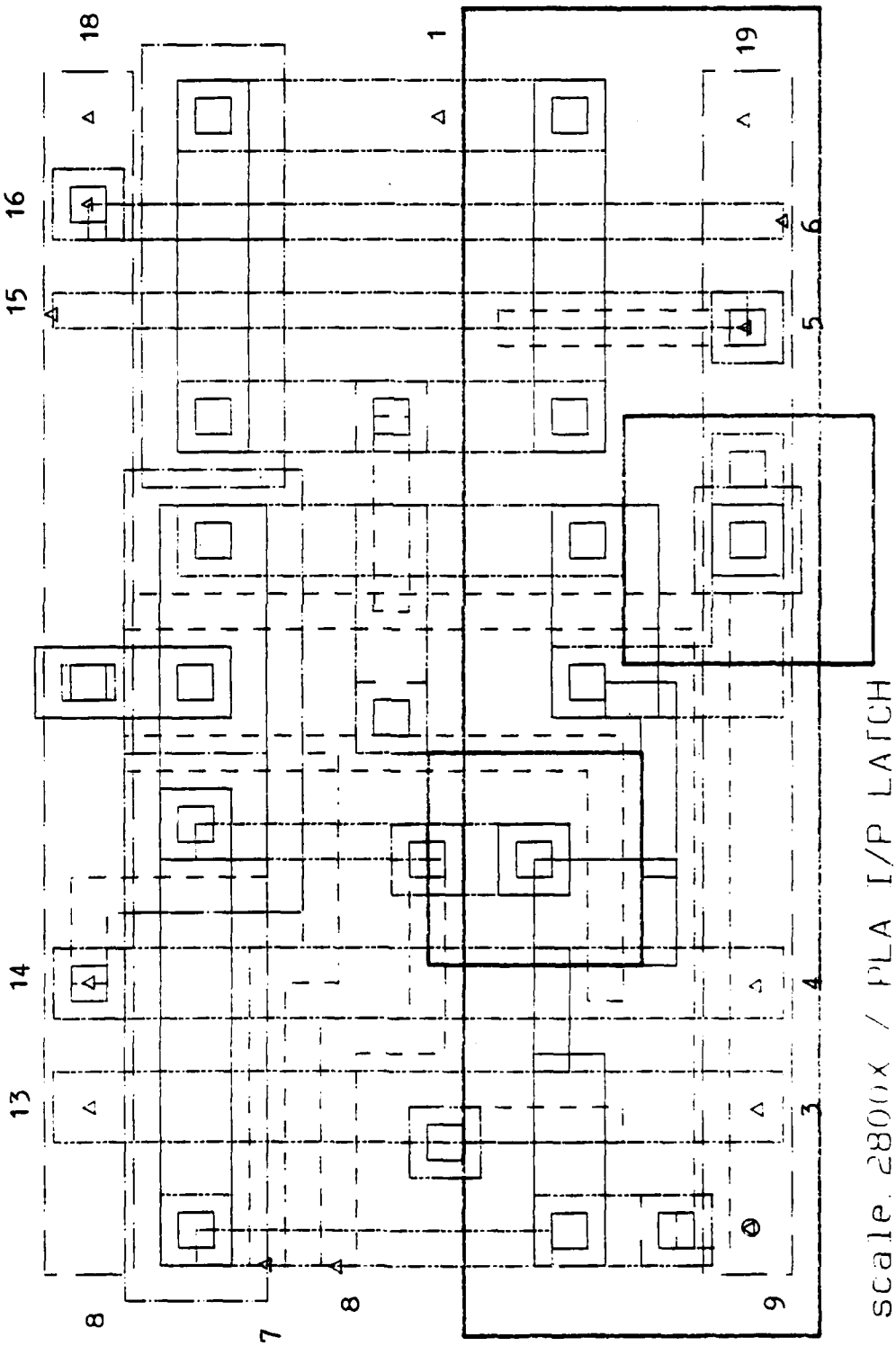


Logic Diagram of PLA I/P LATCH



NOTE: The Sampling Switch is closed in phase2'*phase1' for the T input

Circuit Schematic of PLA I/P LATCH


18
1
19
16
6
15
5
14
4
13
3
8
7
8
9
scale 2800x / PLA I/P LATCH

SCALABLE CMOS CELL FAMILY
NO. : C.14
TITLE : PH1 BUF
HEIGHT : 42 WIDTH : 46
BY : GOWNI
DATE: 3/25/85

TRUTH TABLE

D	Q
0	0
1	1

LOGIC EQUATION(S)

Q = D

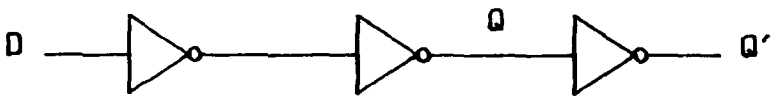
TERMINAL INFORMATION

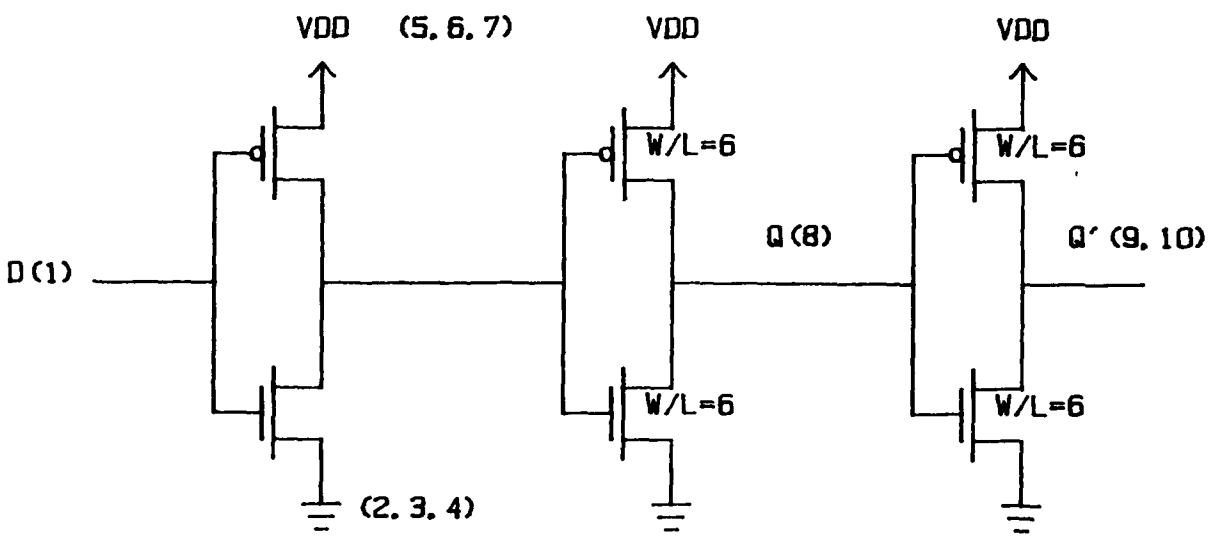
Name	No.	Location	Layer	Abbr.
Data input	1	(23,42)	fm,ip,gp	D
Ground	2,3	(0,30) (46,30)	(sm)(fm,sm)	GND
	4	(46,15)	fm	
Voltage Supply	5,6	(0,0) (46,0)	(fm,sm)(sm)	VDD
	7	(0,15)	fm	
Output	8	(23,0)	fm,ip,gp	Q
Output'	9,10	(16,0) (30,0)	fm	Q'

NOTES

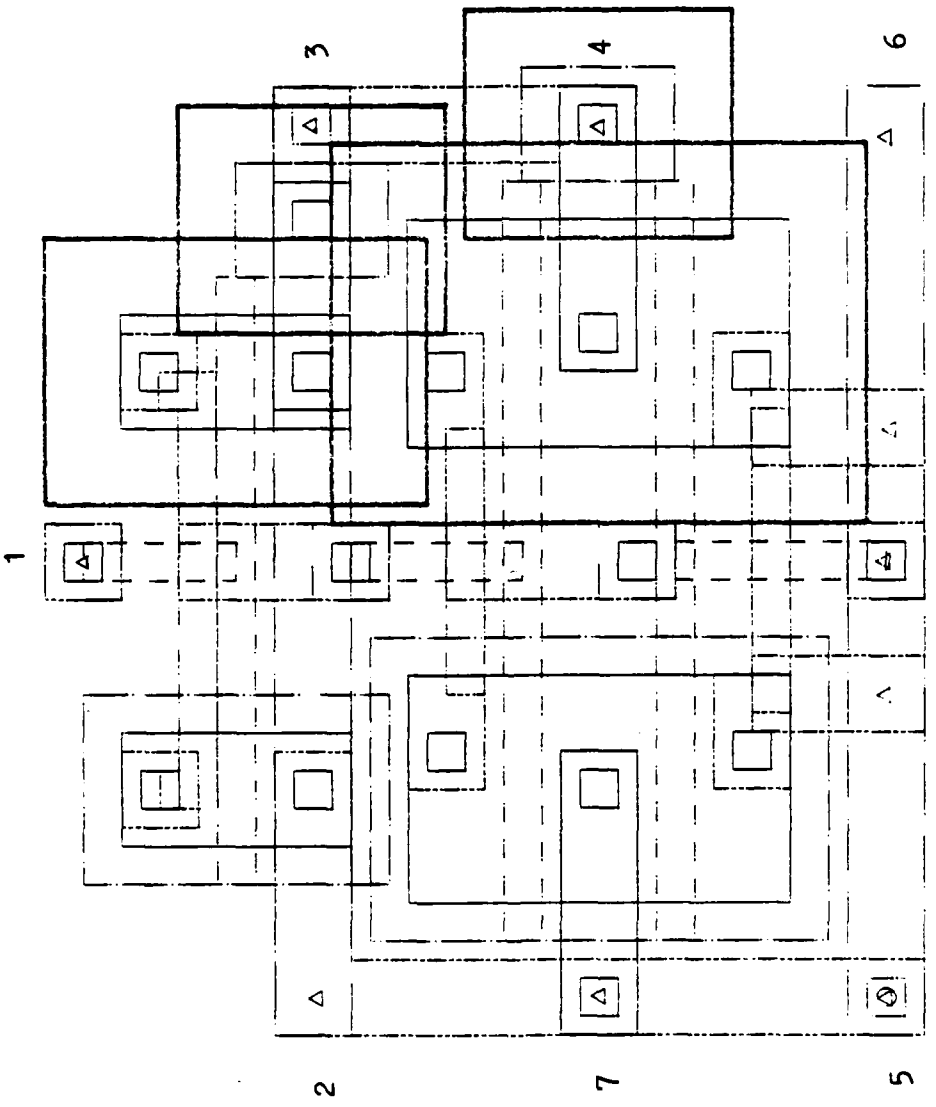
The W/L ratio of the devices used in the inverters of the buffers is 6:1 since the buffers have to drive large capacitances in the microsequencer logic. The inputs to these buffers are the OR-plane outputs and the buffered outputs ripple through in phase1 to control the phase1 microsequencer logic.

Logic Diagram of PH1 BUF





Circuit Schematic of PH1 BUF



scale. 2700X / PH1 BUF

SCALABLE CMOS CELL FAMILY
NO. : C.15 DATE: 3/25/85
TITLE : PH2 BUF BY : GOWNI
HEIGHT : 58 WIDTH : 46

TRUTH TABLE

phase2	D	Q
0	x	1
1	0	0
1	1	1

LOGIC EQUATION(S)

Q = phase2'
+ phase2.D

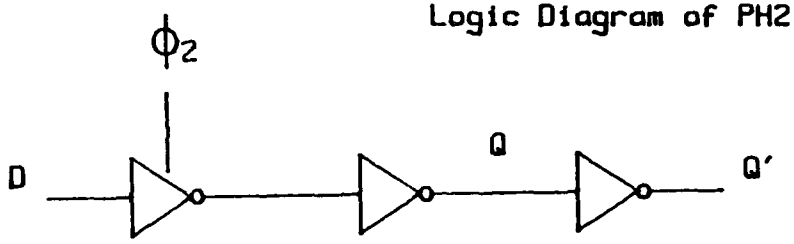
TERMINAL INFORMATION

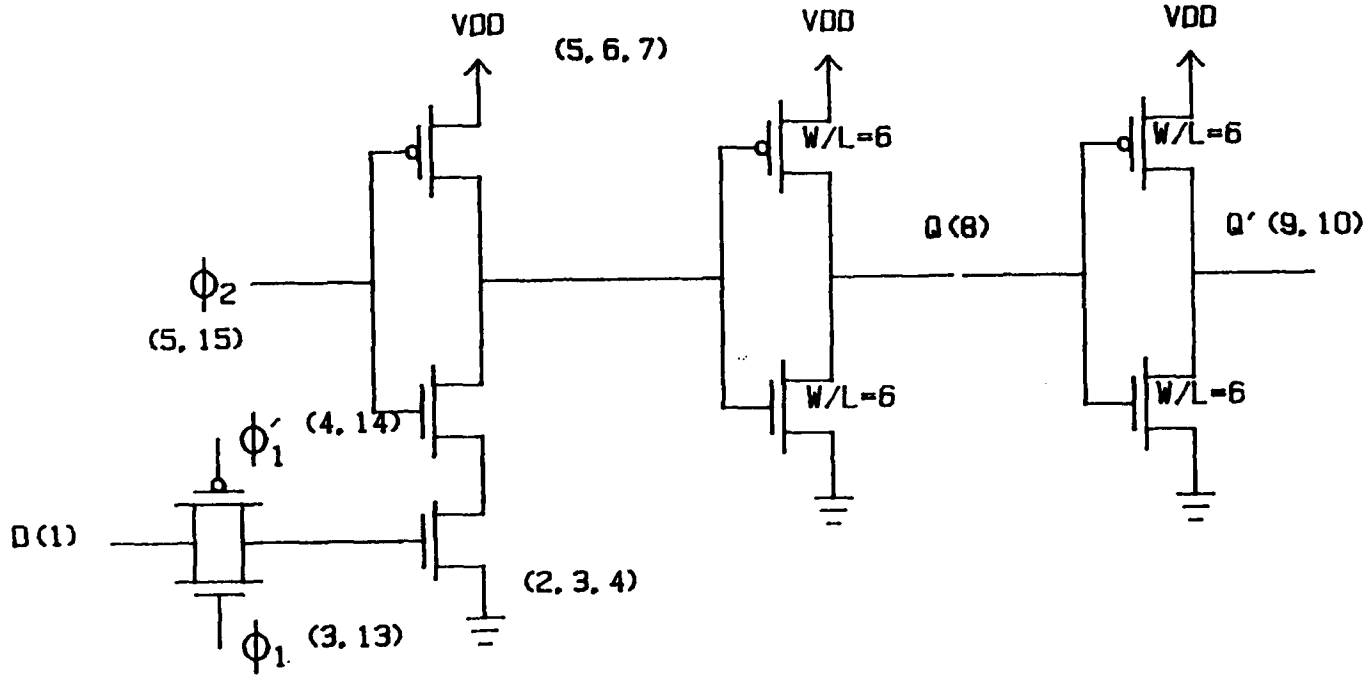
Name	No.	Location	Layer	Abbr.
Data input	1	(23,57)	fm,ip,gp	D
Ground	2,3	(0,30) (46,30)	(sm)(fm,sm)	GND
	4	(46,15)	fm	
Voltage Supply	5,6	(0,0) (46,0)	(fm,sm)(sm)	VDD
	7	(0,15)	fm	
Output	8	(23,0)	fm,ip,gp	Q
Output'	9,10	(16,0) (30,0)	fm	Q'
Phase1	3,13	(0,50) (46,50)	sm	ϕ_1
Phase1'	4,14	(0,58) (46,58)	sm	ϕ_1'
Phase2	5,15	(0,42) (46,42)	sm	ϕ_2

NOTES

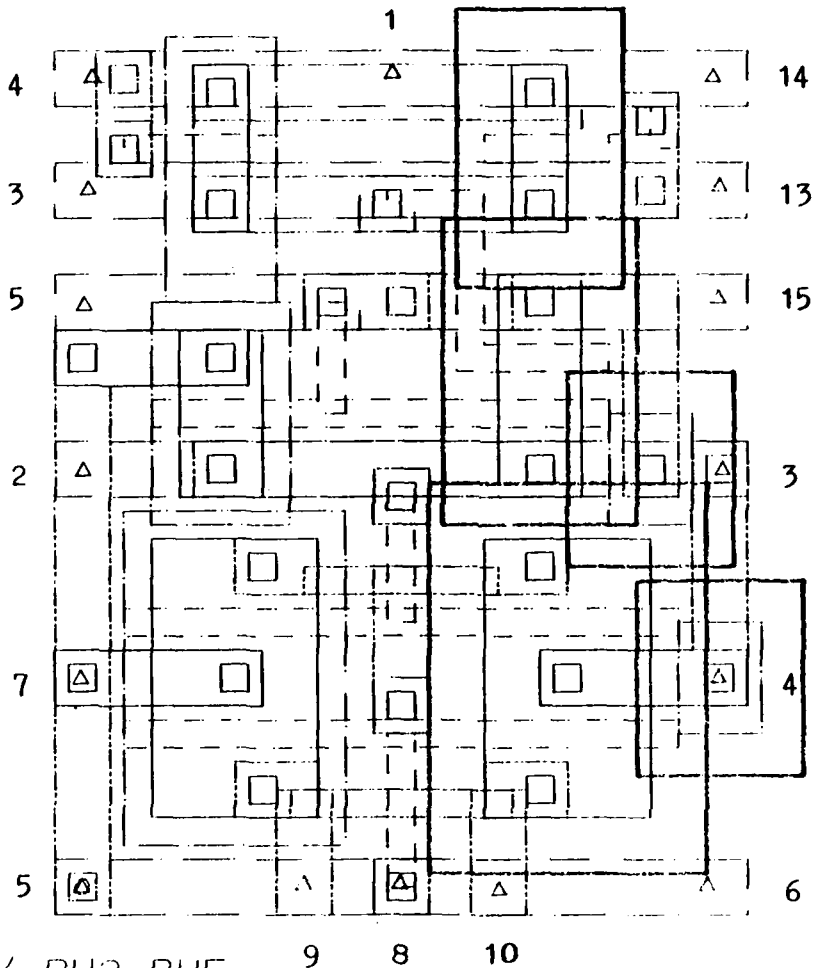
The W/L ratio of the devices used in the inverters of the buffers is 6:1 since the buffers have to drive large capacitances in the microsequencer logic. The inputs to these buffers are the OR-plane outputs and they are gated in phase2 to drive the phase2 controls.

Logic Diagram of PH2 BUF





Circuit Schematic of PH2 BUF



scale. 1970X / PH2 BUF

SCALABLE CMOS CELL FAMILY	NO. : C.16 TITLE : DEC LATCH HEIGHT : 32 WIDTH : 42	DATE: 3/25/85 BY : GOWNI

TRUTH TABLE

MD'	L	L'
0	1	0
1	0	0

LOGIC EQUATION(S)

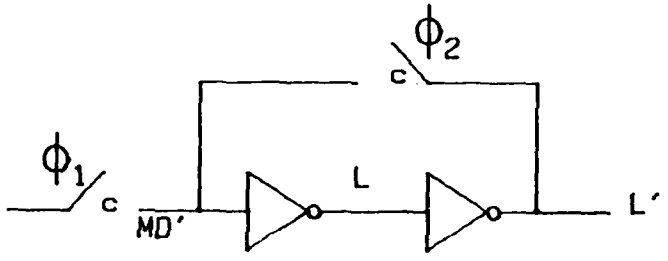
L = MD

TERMINAL INFORMATION

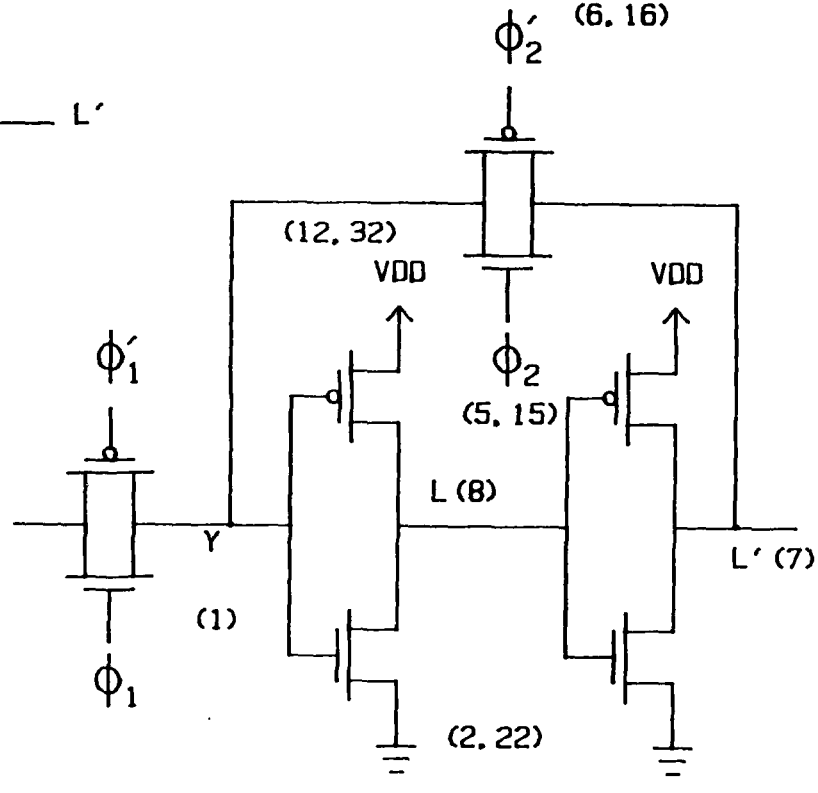
Name	No.	Location	Layer	Abbr.
Data input	1	(2,19)	gp	MD'
Ground	2,22	(0,0) (42,0)	sm	GND
Voltage Supply	12,32	(0,32) (42,32)	sm	VDD
Phase2	5,15	(28,0) (29,34)	fm	ϕ_2
Phase2'	6,16	(35,0) (35,32)	(fm)(fm,ip,gp)	ϕ_2'
Output	8	(44,15)	gp	L
Output'	7	(44,19)	gp	L'

NOTES

This is a classic CMOS 'D' Latch. The input is the inverted output of DEC MUX (MD'), and the output (L) of the latch is stable when the phase2 feedback switch closes. MD is the output of DEC MUX.



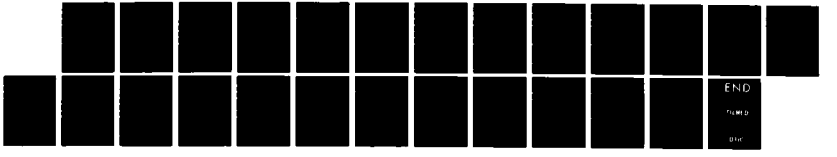
Logic Diagram of DEC LATCH

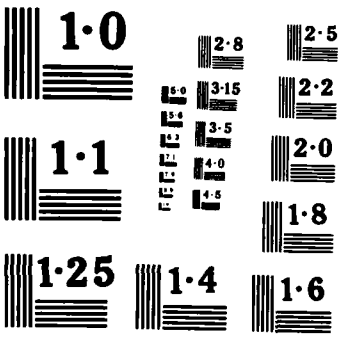


Circuit schematic of DEC LATCH

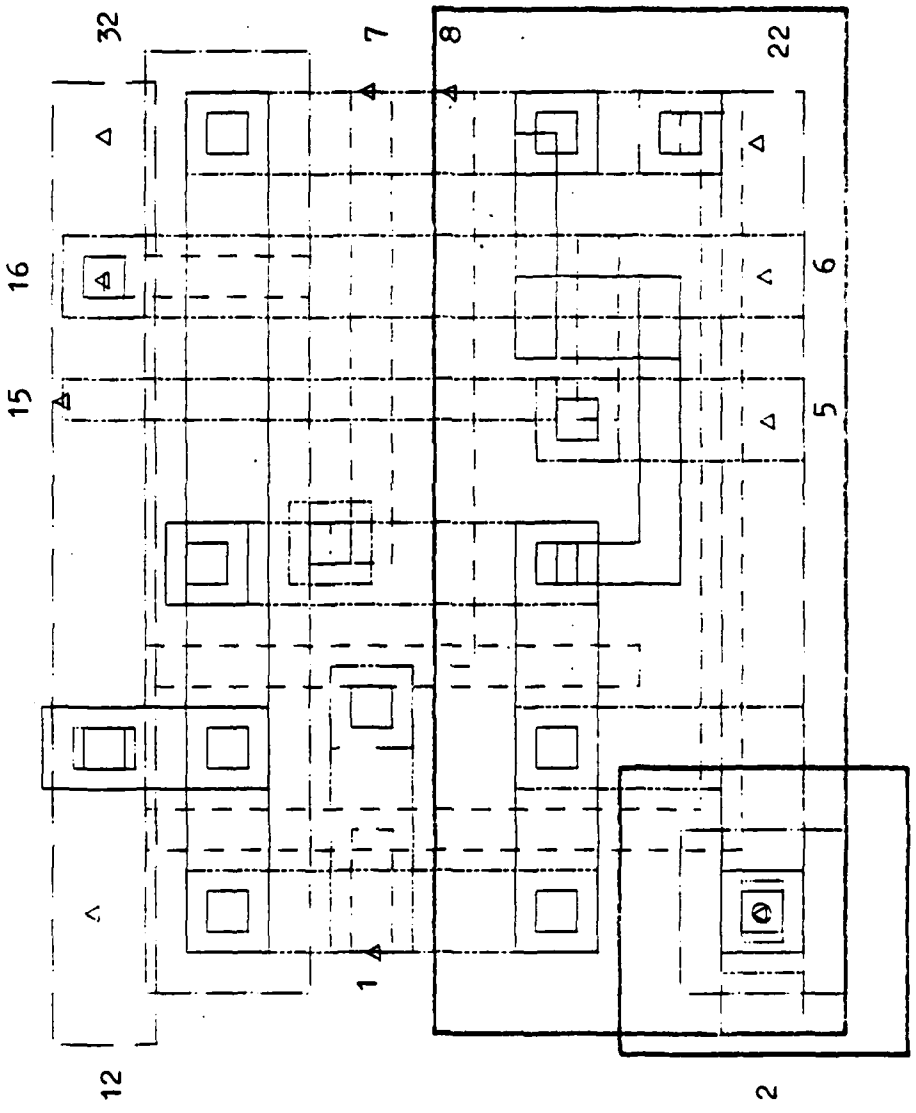
AD-A158 369 BULK CMOS VLSI TECHNOLOGY STUDIES PART 4 DESIGN OF A CMOS MICROSEQUENCER..(U) MISSISSIPPI STATE UNIV MISSISSIPPI STATE DEPT OF ELECTRICAL E.. 3/3

UNCLASSIFIED J D TROTTER ET AL. 17 JUN 85 F/G 9/5 NL





NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART



Scale. 2900X / DEC LATCH

SCALABLE CMOS	NO. : C.17.i	DATE: 3/25/85
CELL FAMILY	TITLE : DEC CAR CHAIN 1	BY : GOWNI
	HEIGHT : 64 WIDTH : 48	

TRUTH TABLE

phase2	P	CIN	COUT
0	x	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

LOGIC EQUATION(S)

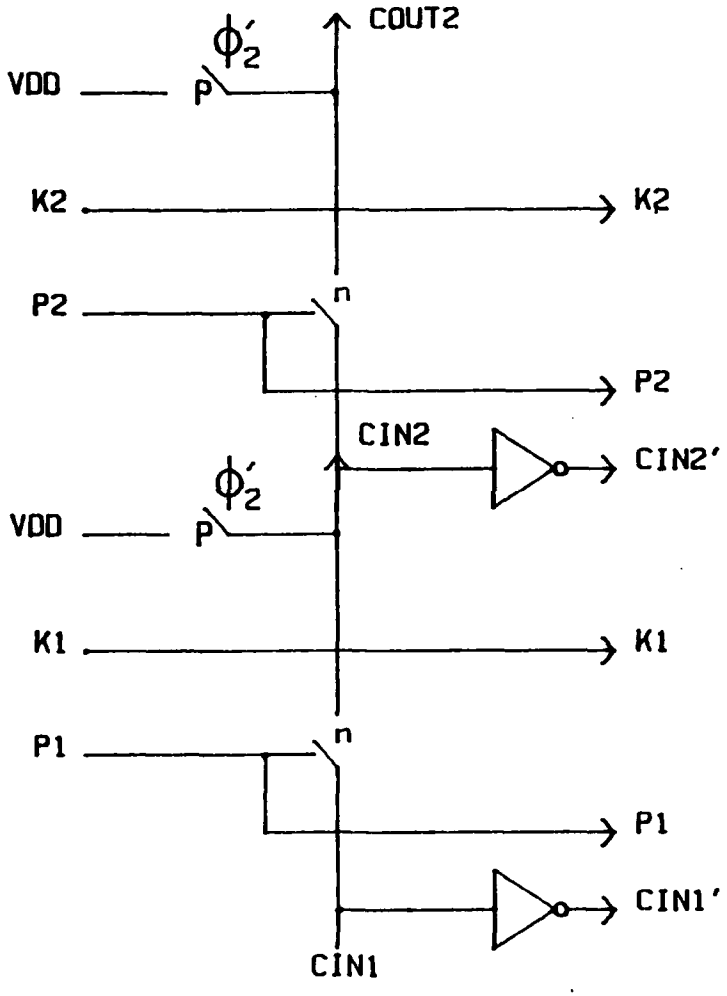
COUT= phase2' +
phase2.(P'+ CIN)

TERMINAL INFORMATION

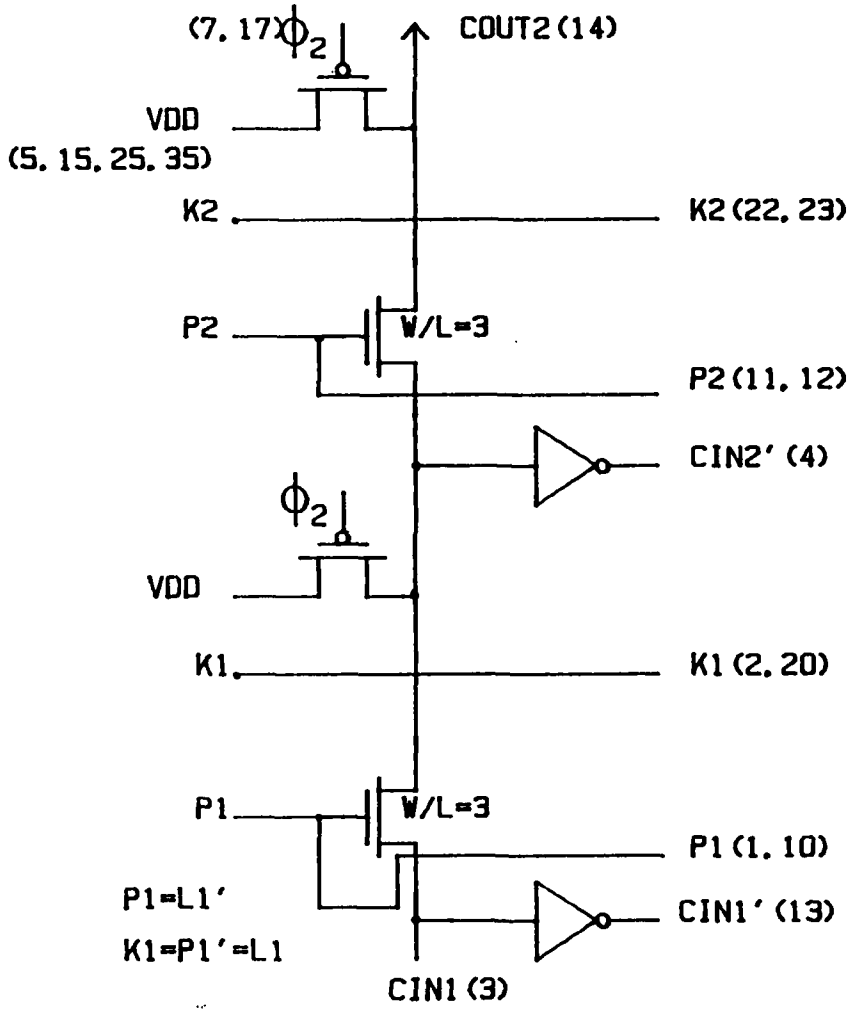
Name	No.	Location	Layer	Abbr.
Pass1	1,10	(-2,15) (48,22)	gp	P1
Pass2	11,12	(-2,50) (48,48)	gp	P2
Voltage Supply	5,15,	(0,0) (48,0)	(fm,sm)(sm)	VDD
	25,35	(0,64) (48,64)	(fm,sm)(sm)	
Phase2	7,17	(7,0) (7,64)	fm,ip,gp	ϕ_2
Kill1	2,20	(-2,19) (48,18)	gp	K1
Kill2	22,23	(-2,45) (48,44)	gp	K2
Carry-in 1	3	(14,-1)	fm	CIN1
Carry-in 1'	13	(44,9)	fm	CIN1'
Carry-in 2'	4	(46,40)	fm	CIN2'
Carry-out 2	14	(34,64)	gp	COUT2
Ground	6,16	(0,32) (48,32)	(fm,sm)(fm)	GND

NOTES

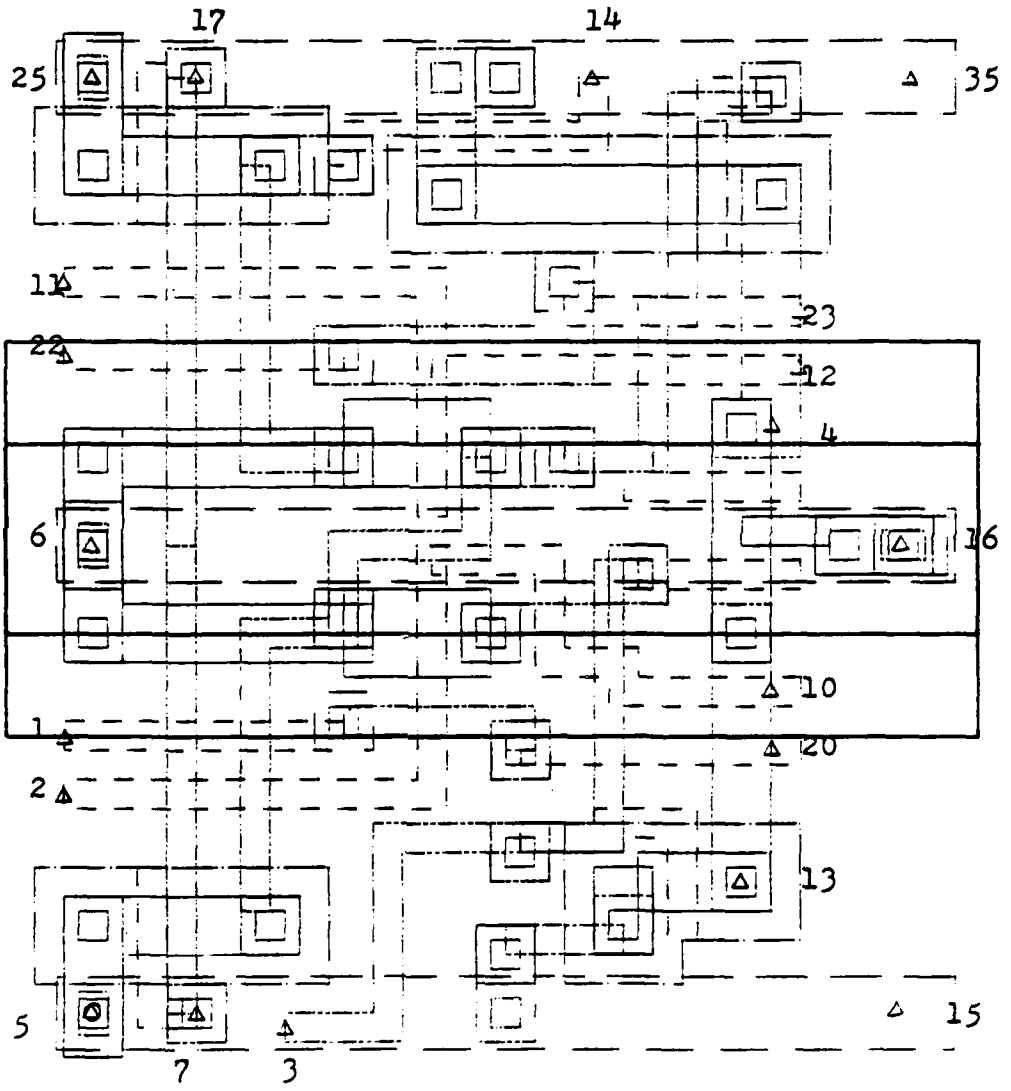
The W/L ratio of the pass transistor is 3:1. The COUT of this stage is buffered (every four stages) to become the CIN of the next stage which is shown in the cell DEC CAR CHAIN 2. This cell is similar to the cell INC CAR CHAIN 1, without the kill and evaluation devices.



Logic Diagram of DEC CAR CHAIN 1



Circuit Schematic of DEC CAR CHAIN 1



scale 2080X / DEC CAR CHAIN 1

SCALABLE CMOS	NO.	: C.17.ii	DATE: 3/25/85
CELL FAMILY	TITLE	: DEC CAR CHAIN 2	BY : GOWNI
	HEIGHT	: 64 WIDTH : 48	

TRUTH TABLE

phase2	P	CIN	COUT
0	x	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

LOGIC EQUATION(S)

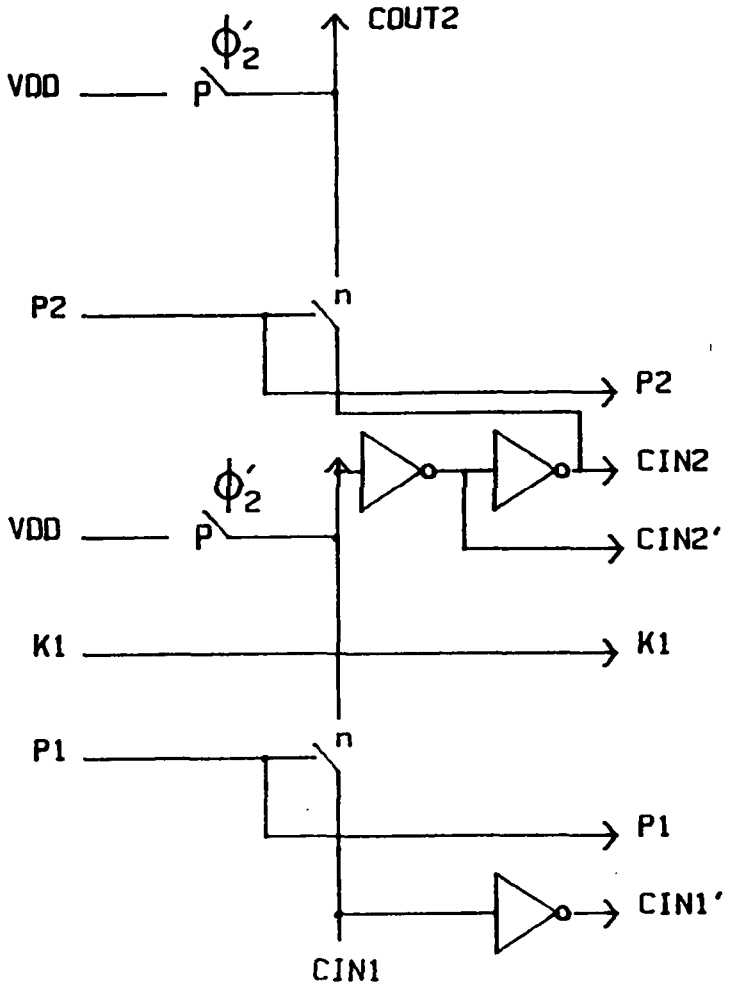
COUT= phase2' +
phase2.(P'+ CIN)

TERMINAL INFORMATION

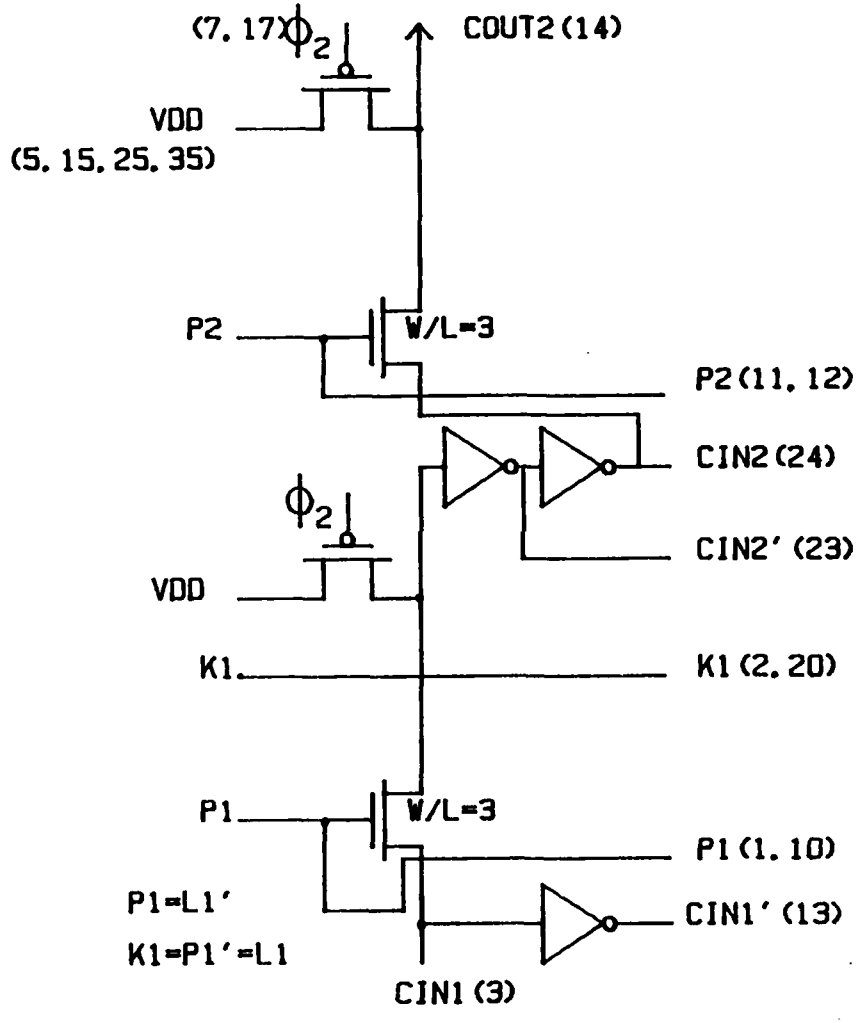
Name	No.	Location	Layer	Abbr.
Pass1	1,10	(-2,13) (44,18)	gp	P1
Pass2	11,12	(-2,50) (41,64)	(gp)(fm,ip,gp)	P2
Voltage Supply	5,15,	(0,0) (48,0)	(fm,sm)(sm)	VDD
	25,35	(0,64) (48,64)	(fm,sm)(sm)	
Phase2	7,17	(7,0) (7,64)	fm,ip,gp	ϕ_2
Kill1	2,20	(-2,17) (44,14)	gp	K1
Carry-in 1	3	(35,1)	fm,ip,gp	CIN1
Carry-in 1'	13	(42,6)	fm	CIN1'
Carry-in 2	24	(44,49)	gp	CIN2
Carry-out 2	14	(13,64)	fm	COUT2
Carry-in 2'	23	(44,43)	gp	CIN2'
Ground	6,16	(0,32) (48,32)	(fm,sm)(fm)	GND

NOTES

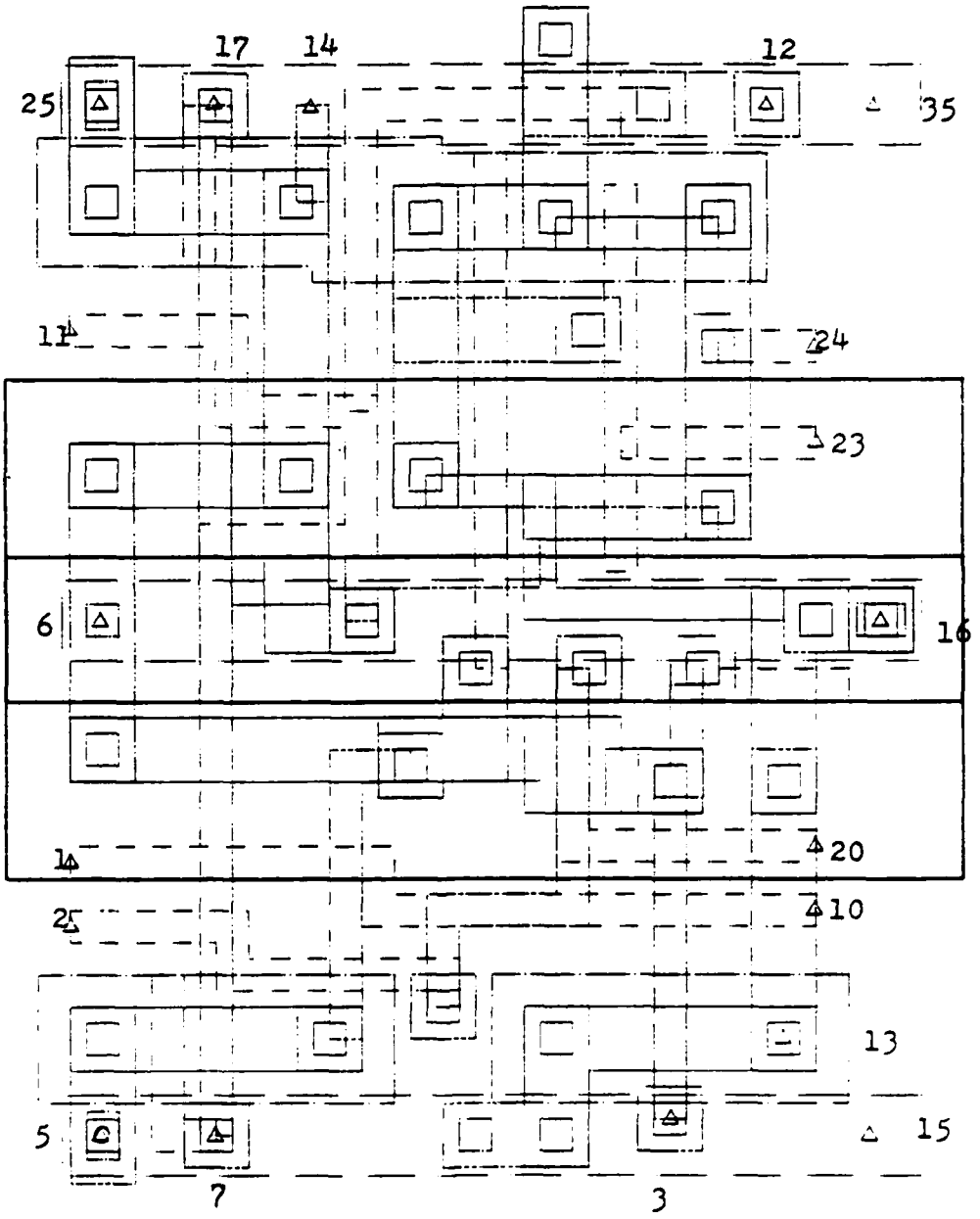
The W/L of the pass transistor is 3:1. This carry chain is used every fourth stage with the EX-OR circuit being evaluated from P, CIN, and CIN' instead of P, P', and CIN. Note that P' is same as K. CIN and CIN' are obtained from the buffer as shown in the logic diagram and the circuit schematic on the following page. This cell is similar to the cell INC CAR CHAIN 2, without the kill and evaluation blocks.



Logic Diagram of DEC CAR CHAIN 2



Circuit Schematic of DEC CAR CHAIN 2



scale. 2330X / DEC CAP CHAIN 2

SCALABLE CMOS CELL FAMILY	NO. : C.18 TITLE : DEC O/P LATCH HEIGHT : 32 WIDTH : 66	DATE: 3/25/85 BY : GOWNI

TRUTH TABLE

phase2	D	DOP
0	x	DOP*
0	0	0
1	1	1

LOGIC EQUATION(S)

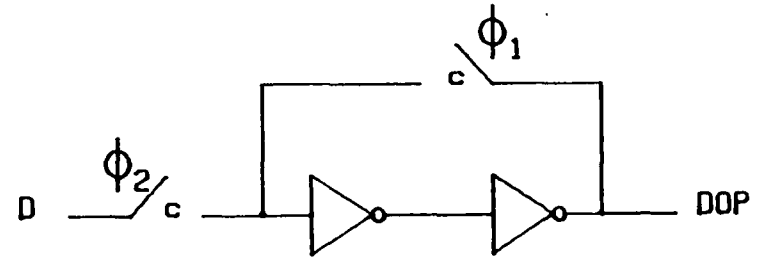
IOP = phase2'.DOP*
+ phase2.D

TERMINAL INFORMATION

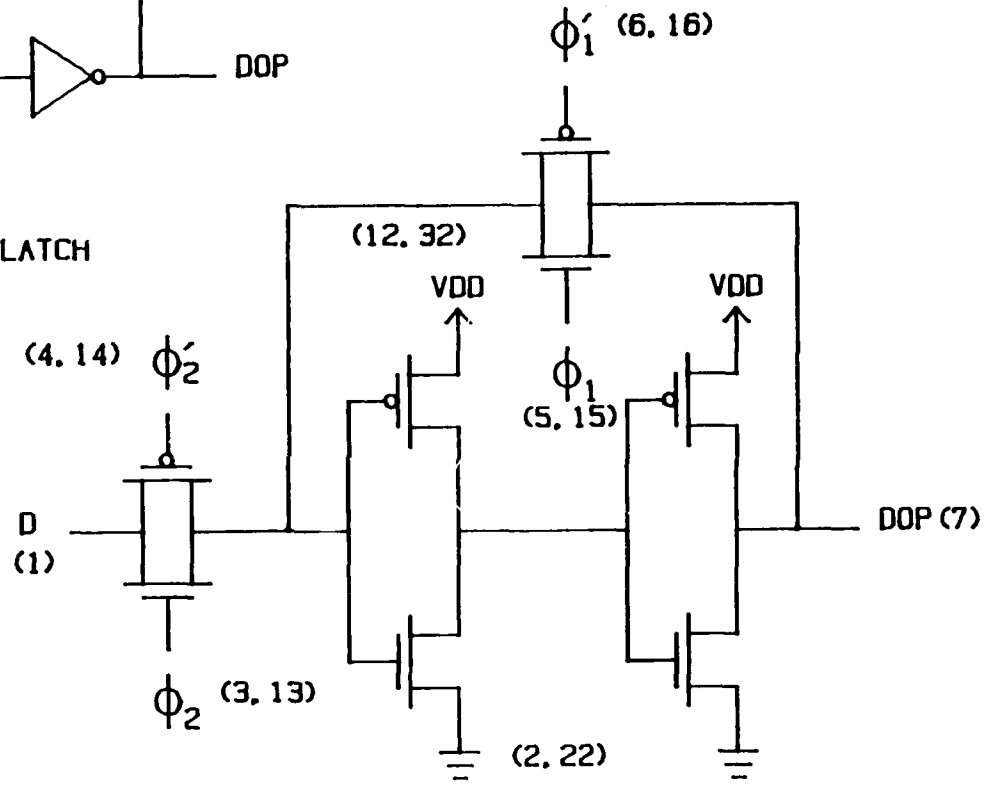
Name	No.	Location	Layer	Abbr.
Data input	1	(66,17)	fm	D
Ground	2,22	(0,0) (66,0)	sm	GND
Voltage Supply	12,32	(0,32) (66,32)	sm	VDD
Phase2	3,13	(59,0) (59,32)	(fm,ip,gp)(fm)	ϕ_2
Phase2'	4,14	(52,0) (52,32)	(fm)(fm,ip,gp)	ϕ_2'
Phase1	5,15	(14,0) (14,32)	fm	ϕ_1
Phase1'	6,16	(7,0) (7,32)	(fm)(fm,ip,gp)	ϕ_1'
Output	7	(66,11)	sm	DOP

NOTES

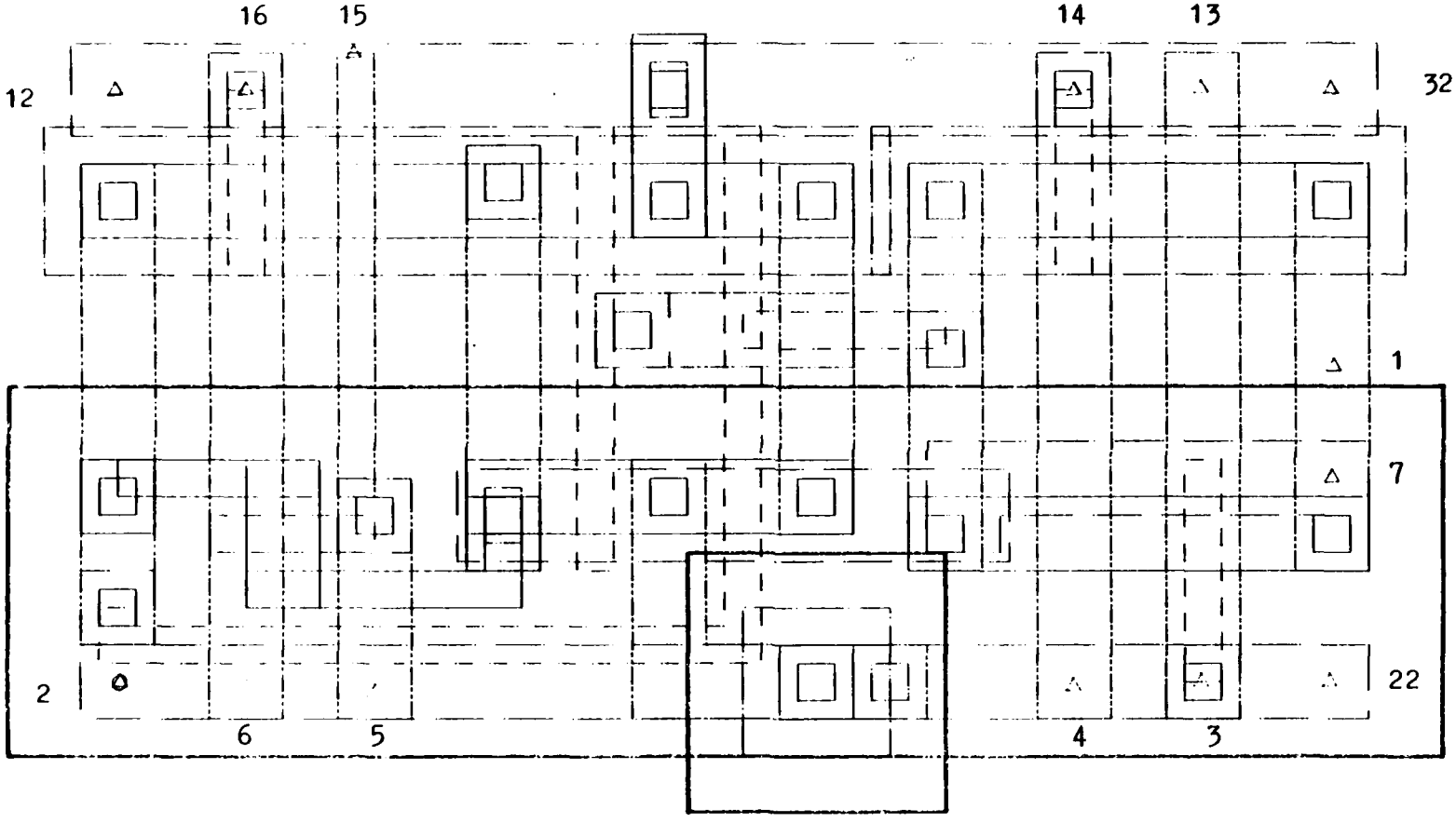
This is a classic CMOS 'D' Latch. The output (D) of the EX-OR circuit in the DEC LOGIC is the input, sampled in phase2 and is stable when the phase1 feedback switch closes.



Logic Diagram of DEC O/P LATCH



Circuit Schematic of DEC O/P LATCH



scale. 2800x / DEC O/P LATCH

SCALABLE CMOS CELL FAMILY | NO. : C.19 | TITLE : DEC LOGIC LSB | HEIGHT : 64 WIDTH : 70 | DATE: 3/25/85 | BY : GOWNI

TRUTH TABLE

phase2	P0	COUT0	D0
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	1

LOGIC EQUATION(S)

COUT0 = phase2' + phase2.P0'

D0 = P0

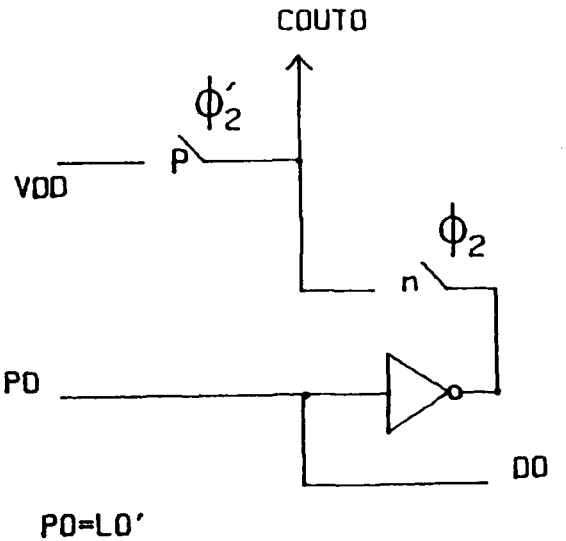
(where P0=L0')

TERMINAL INFORMATION

Name	No.	Location	Layer	Abbr.
Pass0	1	(-2,13)	gp	P0
Output0	2	(70,17)	fm	D0
Pass1	7	(-2,51)	gp	P1
Kill1	8	(-2,47)	gp	K1
Output1	9	(72,47)	fm	D1
Voltage Supply	5,15, 25,35	(0,0) (0,64) (70,0) (70,64)	sm (sm)(fm,sm)	VDD
Phase2	4,14	(7,0) (7,64)	(fm)(fm,ip,gp)	ϕ_2
carry-out 1	3	(14,64)	fm	COUT1
Ground	6,16	(0,32) (70,32)	(fm,sm)(sm)	GND

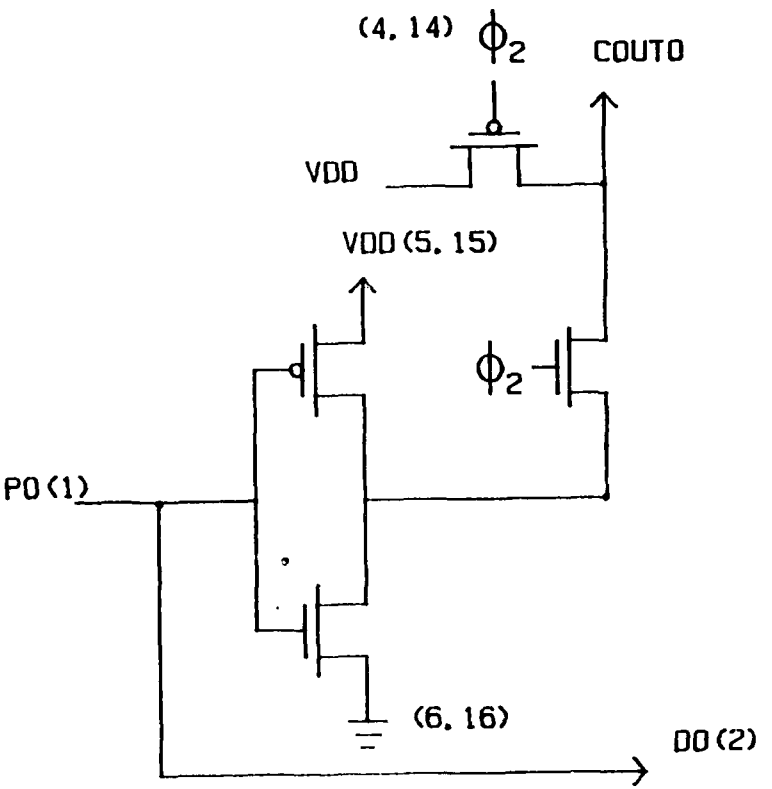
NOTES

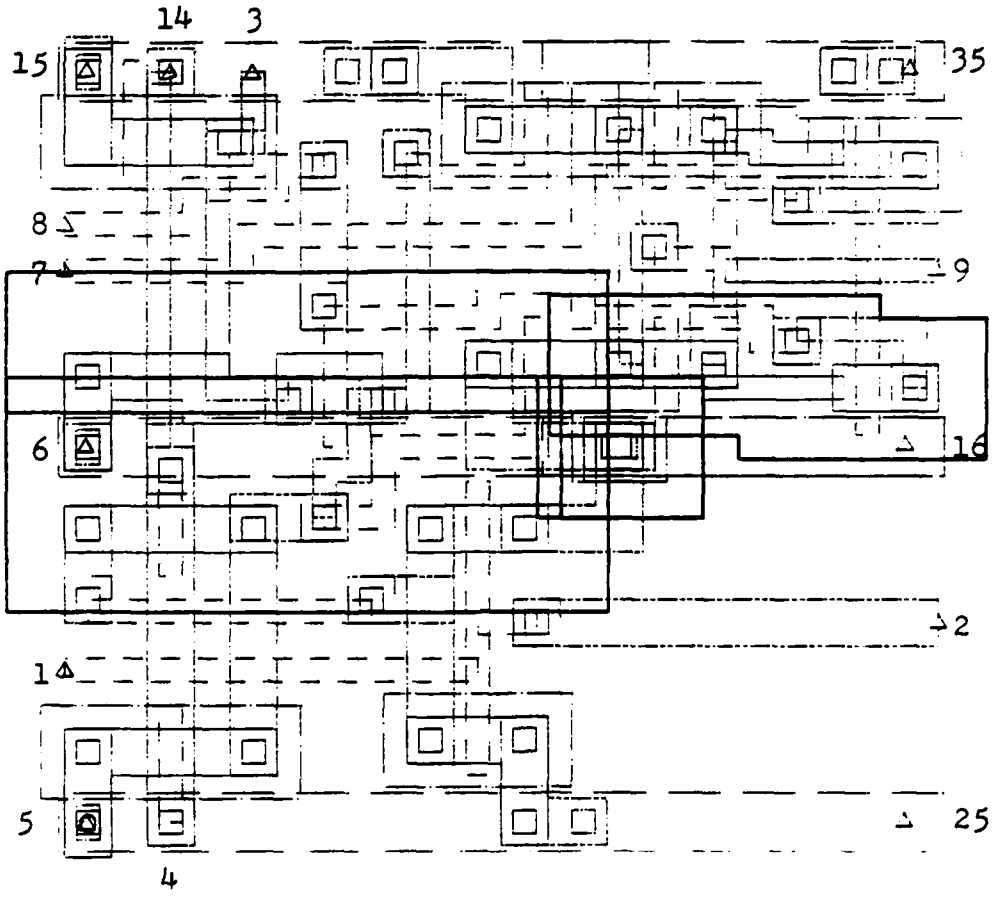
This is the least significant stage of the DEC LOGIC in the DEC macro. The carry-out of this stage is the carry-in to the first stage. The layout shows the DEC LOGIC LSB along with the first bit of the carry chain (similar to the one in DEC CAR CHAIN 1). Hence the terminal information section includes both stages.



Logic Diagram of DEC LOGIC LSB

Circuit Schematic of DEC LOGIC LSB





scale: 1660X / DEC LOGIC LSB

SCALABLE CMOS CELL FAMILY

NO. : C.20
TITLE : ZD
HEIGHT : 32 WIDTH : 8

DATE: 3/25/85
BY : GOWNI

TRUTH TABLE

L7	L6	L5	L4	L3	L2	L1	L0	Z
0	0	0	0	0	0	0	0	1
0	.	.	.	.	.	.	1	0
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
1	1	1	1	1	1	1	1	0

LOGIC EQUATION(S)

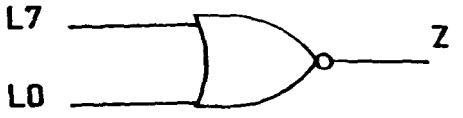
$Z = (L0 + L1 + .. L7)'$

TERMINAL INFORMATION

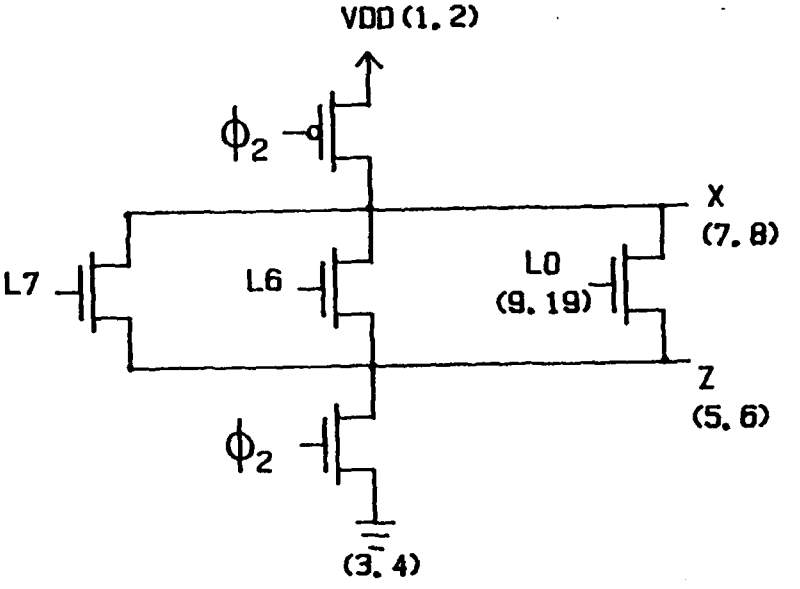
Name	No.	Location	Layer	Abbr.
Voltage supply	1,2	(0,32) (8,32)	sm	VDD
Ground	3,4	(0,0) (50,32)	sm	GND
Common drain node (ZD output)	5,6	(8,0) (8,32)	fm	Z
Common source node	7,8	(0,0) (0,32)	fm	X
Input	9,19	(-2,11) (10,11)	gp	L
Input'	10,20	(-2,15) (10,15)	gp	L'

NOTES

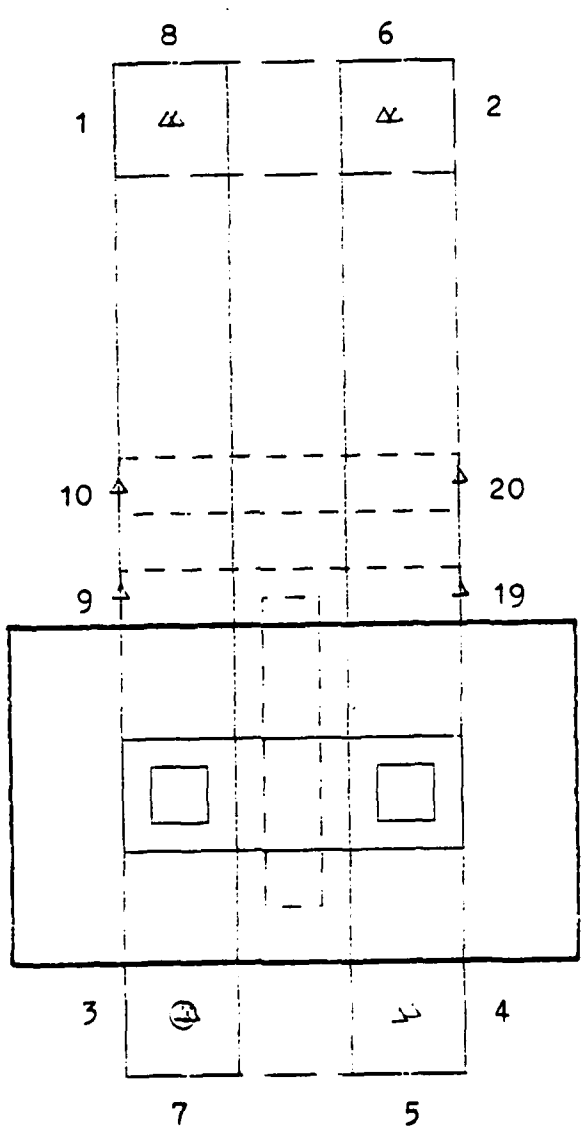
This cell shows only one input of the ZD. Actually, ZD is an array of 8 such cells in positive y direction. The circuit schematic on the next page shows the precharge and evaluation devices whose layouts are shown in the cell ZD LSB. The inputs to the ZD are the 8 bits of LCR (L).



Logic Diagram of ZD



Circuit Schematic of ZD



scale. 4000X / ZD

SCALABLE CMOS CELL FAMILY
NO. : C.21
TITLE : ZD LSB
HEIGHT : 32 WIDTH : 33
DATE: 3/25/85
BY : GOWNI

TRUTH TABLE

phase2	Z
0	1
1	(L0 + L1 ..+L7)'

LOGIC EQUATION(S)

Z = phase2
+ phase2.(L0+L1+..L7)'

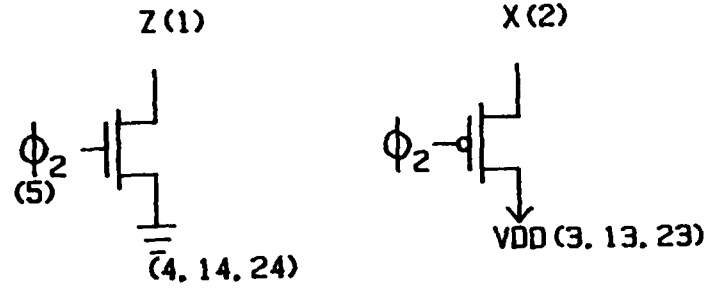
TERMINAL INFORMATION

Name	No.	Location	Layer	Abbr.
Phase2	5	(15,0)	fm	ϕ_2
Voltage Supply	3,13	(0,30) (33,30)	(sm)(fm,sm)	VDD
	23	(33,15)	fm	
Ground	4,14	(0,0) (33,0)	(fm,sm)(sm)	GND
	24	(0,15)	fm	
Common drain node (ZD output)	1	(26,30)	fm	Z
Common source node	2	(8,30)	fm	X

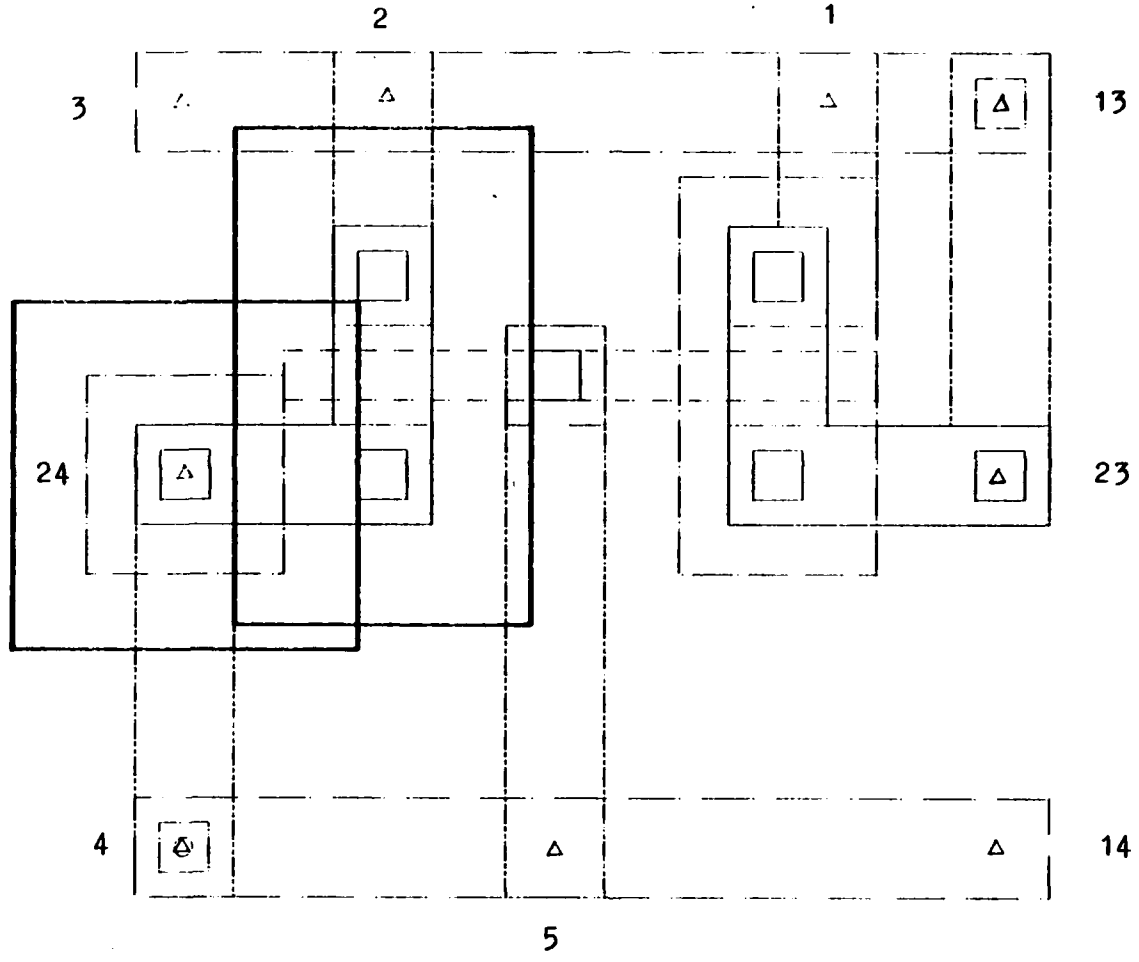
NOTES

This cell consists of the precharge and the evaluation devices for the cell ZD.

Logic Diagram of ZD LSB



Circuit Schematic of ZD LSB



scale. 3500X / ZDLSB

SCALABLE CMOS CELL FAMILY
NO. : C.22
TITLE : DEC MUX + LCS INT
HEIGHT : 32 WIDTH : 113
DATE: 3/25/85
BY : GOWNI

TRUTH TABLE FOR DEC MUX

N	DOP	LCS2	LC	LD	LL	MD
x	x	x	0	0	0	MD*
x	x	0	0	0	1	0
x	x	1	0	0	1	1
x	0	x	0	1	0	0
x	1	x	0	1	0	1
0	x	x	1	0	0	0
1	x	x	1	0	0	1

LOGIC EQUATION(S) FOR DEC MUX

MD = LD.DOP + LC.N
+ LL.LCS2

TRUTH TABLE FOR LCS INT

MD	MD'	ø2F	DF
x	x	0	DF*
0	1	1	1
1	0	1	0

LOGIC EQUATION(S) FOR LCS INT

DF = phase2F'.DF*
+ phase2F.MD'

NOTES

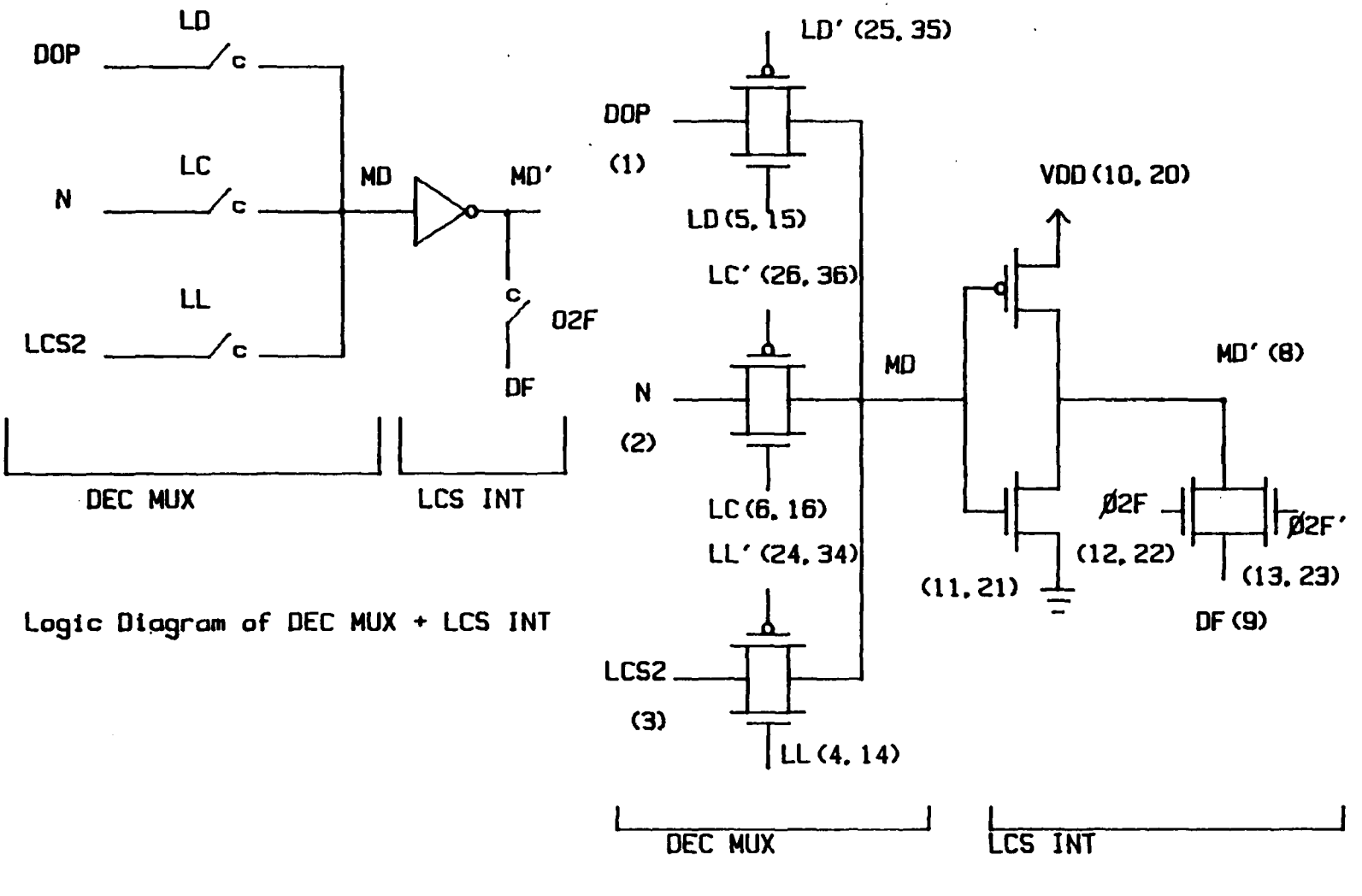
The layout for this cell includes the layouts of DEC MUX and LCS INT. Truth table and logic equation(s) sections for these cells are shown separately but the terminal information section is same as shown on the following page.

SCALABLE CMOS CELL FAMILY	NO. : C.22 TITLE : DEC MUX + LCS INT HEIGHT : 32 WIDTH : 113	DATE: 3/25/85 BY : GOWNI

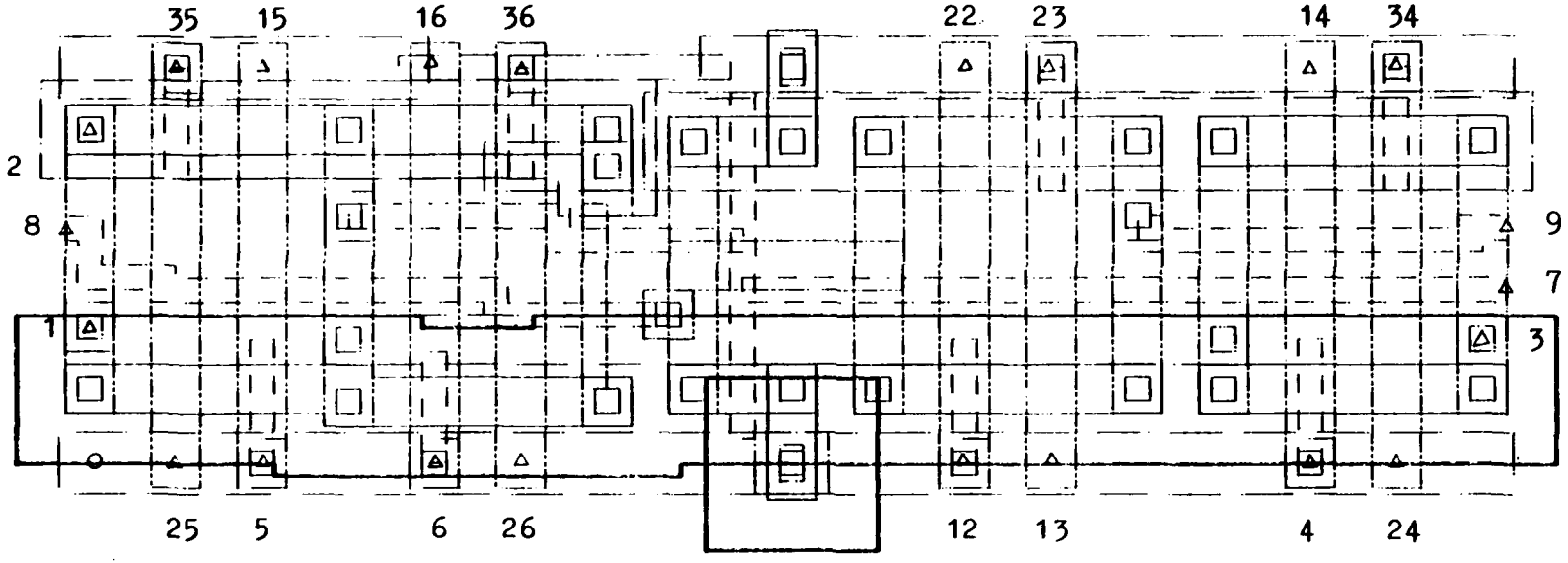
TERMINAL INFORMATION

Name	No.	Location	Layer	Abbr.
Decremented output	1	(0,11)	fm,sm	DOP
External count	2	(-2,24)	sm	N
Second word of LCS	3	(113,10)	fm,sm	LCS2
Load LCS2	4,14	(99,0) (99,32)	(fm,ip,gp)(fm)	LL
(Load LCS2)'	24,34	(106,0) (106,32)	(fm)(fm,ip,gp)	LL'
Load DOP	5,15	(14,0) (14,32)	(fm,ip,gp)(fm)	LD
(Load DOP)'	25,35	(7,0) (7,32)	(fm)(fm,ip,gp)	LD'
Load count	6,16	(28,0) (28,32)	(fm,ip,gp)(fm)	LC
(Load count)'	26,36	(35,0) (35,32)	(fm)(fm,ip,gp)	LC'
Data from LCS top	7	(115,14)	gp	QB
Output to DEC LATCH	8	(-2,19)	gp	MD'
Output to LCS top	9	(115,19)	gp	DF
Voltage supply	10,20	(0,32) (113,32)	sm	VDD
Ground	11,21	(0,0) (113,0)	sm	GND
Phase2F	12,22	(71,0) (71,32)	(fm,ip,gp)(fm)	Ø2F
Phase2F'	13,23	(78,0) (78,32)	(fm)(fm,ip,gp)	Ø2F'

Circuit Schematic of DEC MUX + LCS INT



Logic Diagram of DEC MUX + LCS INT



scale. 1770X / DEC MUX + LCS INT

BIBLIOGRAPHY

1. Bipolar Microprocessor Logic and Interface Data Book, Advanced Micro Devices, 1983.

2. D. A. Hodges and H. G. Jackson, Analysis and Design of Digital Integrated Circuits, McGraw-Hill, New York, 1983.

3. C. W. Jones, A 1.2u CMOS Data Path Chip, Masters Thesis for Mississippi State University, Unpublished, 1985.

4. J. Mavor, M. A. Jack, P. B. Denyer, Introduction to MOS LSI Design, Addison-Wesley, Great Britain, 1983.

5. C. Mead and L. Conway, Introduction to VLSI Systems, Massachusetts, Addison-Wesley, 1980.

6. M. Morris Mano, Computer System Architecture, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

7. J. D. Trotter, Stored Logic Arrays-SLAs implemented with Clocked CMOS, MSFC, Alabama.

8. J. D. Trotter, The Mosaic Transistor Array Applied to Custom Microprocessors, MSFC, Alabama.

9. J. D. Trotter, Solid State Electronics I, Unpublished Class Notes, 1983.

END

FILMED

10-85

DTIC