https://ntrs.nasa.gov/search.jsp?R=19760015854 2020-03-22T16:07:47+00:00Z

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

NASA CONTRACTOR
REPORT

NASA CR-124210

(NASA-CR-124210) LOGSIM USER'S MANUAL (M&S Computing, Inc.) 84 p HC $5.00 CSCL 09B

N76-22942

Unclas
G3/61 25278

LOGSIM USER'S MANUAL

By C. L. Mitchell and J. F. Taylor

M&S Computing, Inc.
Huntsville, Alabama 35805

January 13, 1972

Prepared for

NASA - GEORGE C. MARSHALL SPACE FLIGHT CENTER
Marshall Space Flight Center, Alabama 35812

MAY 1976
RECEIVED
NASA STI FACILITY
INPUT BRANCH

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO. CR-124210	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
4. TITLE AND SUBTITLE LOGSIM USER'S MANUAL		5. REPORT DATE January 13, 1972
		6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) C. L. Mitchell and J. F. Taylor		8. PERFORMING ORGANIZATION REPORT # 72-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS M&S Computing, Inc. P. O. Box 5183 Huntsville, Alabama 35805		10. WORK UNIT NO.
		11. CONTRACT OR GRANT NO. NAS8-25621
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, D. C. 20546		13. TYPE OF REPORT & PERIOD COVERED Contractor Report
		14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

16. ABSTRACT

This document is intended to serve as the User's Manual for the LOGSIM Program. All program options are explained and a detailed definition of the format of each input card is given. The detail description of the internal logic and flow of the LOGSIM Program is not included in this report, but has been incorporated into a separate document entitled, "Programmer's Manual for LOGSIM." It is recommended that every user review the Programmer's Manual in order to receive an appreciation of the functions performed by the LOGSIM Program; however, it will not be necessary for the user to become intimately familiar with the internal operation of the LOGSIM Program to effectively use the extensive capabilities inherent within the program.

17. KEY WORDS	18. DISTRIBUTION STATEMENT Unclassified - Unlimited

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	84	NTIS

MSFC - Form 3292 (Rev December 1972)   For sale by National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

TABLE OF CONTENTS
(continued)

LIST OF FIGURES

LIST OF FIGURES
(continued)

LIST OF TABLES

1. INTRODUCTION

The Logic Simulation Program (LOGSIM) was designed to serve an important role in computer aided design of electronic circuits, that is, to provide the designer with a means by which to verify the intended logic of a design prior to implementation of the circuit hardware.

A description of the logic network and its input signals is provided by the user as input data. LOGSIM verifies all input data and identifies any errors with diagnostic messages before proceeding with simulation. Simulation consists of switching the output levels of network elements in response to input level changes and recording all such events on magnetic tape. This tape is then processed according to user specifications to produce logic timing diagrams of any or all element outputs.

LOGSIM is capable of simulating very large logic networks within a modest amount of computer core memory. Although it has been written in Fortran IV to minimize machine dependency, its array dimensions are effectively adjusted during run-time to allow each simulation to make optimum use of the available memory. The program has an internal logic library of 15 logic element types to which the user may add new element types by simply defining their operation. In addition, the user may define complex logic functions as read-only-memories (ROMs) with up to 15 inputs each.

LOGSIM is a three-valued logic simulator, in that the unknown (indeterminate) state can be represented and propagated as well as logic HIGH and logic LOW. This feature eliminates the necessity for network initialization and, in conjunction with other special provisions discussed in Section 2, permits simple and complete logic modeling of metal-oxide-semiconductor (MOS) circuitry.

2. LOGSIM PROGRAM OPERATIONS

2.1 LOGSIM Representation of Logic Delay and Decay Times

In order for logic circuitry to be simulated on an ideal binary-valued basis, some relationship must be established between the ideal LOW and HIGH levels and the delayed level transition which occurs in reality. Due to physical constraints of a device, its levels cannot change instantaneously. Figure 2-1 illustrates the delays in switching from LOW to HIGH and HIGH to LOW. The LOGSIM approach is to establish the switching time as the time when the gate in reality has switched 90% of the change between levels. Hence, a gate is considered switched from LOW to HIGH after a rise delay time, t_r, which is the actual time necessary to switch 90% of the way from LOW to HIGH. The same applies for HIGH to LOW switching with a fall time delay, t_f.

The rise and fall delay times are supplied by the user for each circuit element. The user may specify up to 255 different rise delay times and up to 255 different fall delay times for one simulation. These times should be based on the characteristics of the devices which implement the logic. Varying the delay times for different simulations of a logic network will show the user the changes in the logical operation due to delay time variations.

In addition to its capabilities for modeling normal propagation delays, LOGSIM has special provisions for modeling the decay of stored charge on MOS logic circuits. Figure 2-2 illustrates a MOS transmission gate circuit for which such a model is very useful. The output node of the transmission gate is charged by the source when the gate input goes HIGH. This node is expected to hold the charge after the gate input returns to logic LOW. Since leakage resistance will slowly drain the charge, there is some fall decay time, $t_{df,}$ after which the voltage at the output node will have fallen below the logic HIGH threshold level. The output must be considered indeterminate after this decay time has elapsed. The charge decay begins when the transmission gate is turned off by its gate input and the output switches into a charge holding mode.

Section 3.2 describes how the user may define the holding mode as one of the possible output states of MOS logic elements. Either a rise decay time or a fall decay time may be specified for any logic element having a holding mode (both may, in fact, be specified but such a model may not be meaningful).

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

LOGSIM INTERPRETATION OF LOGIC SWITCHING TIME DELAYS

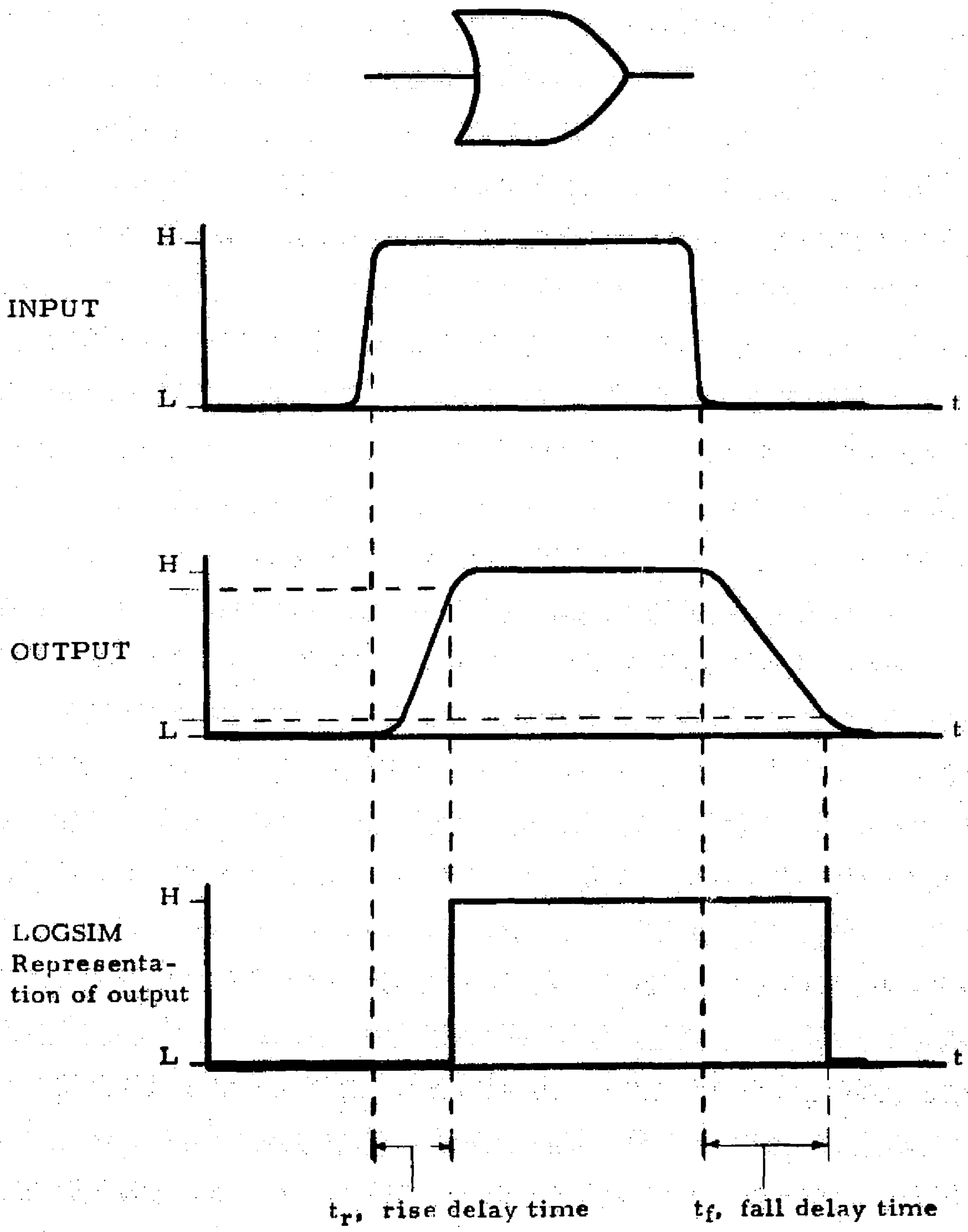
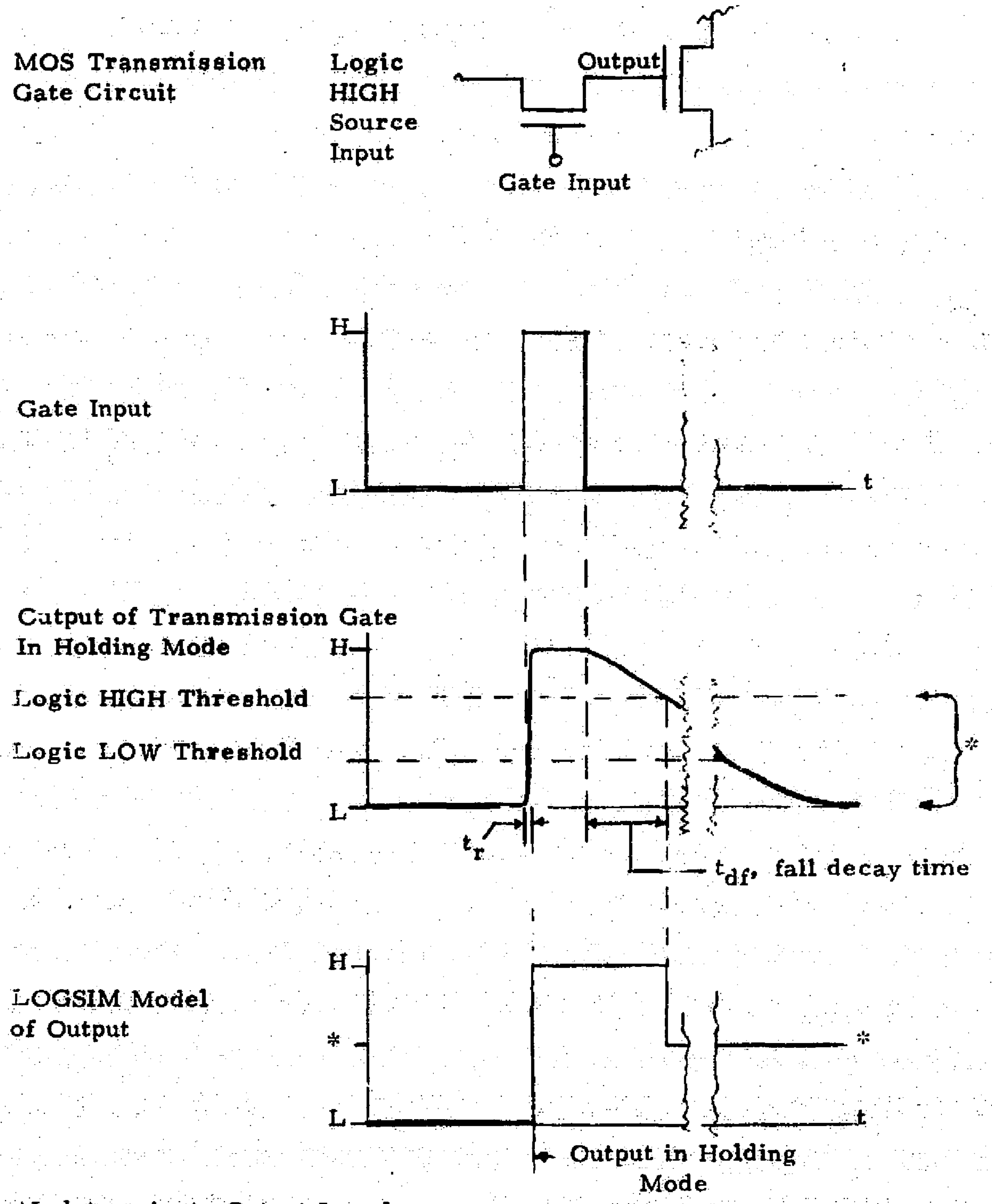


Figure 2-1

LOGSIM INTERPRETATION OF A LOGIC FALL DECAY OF A MOS TRANSMISSION GATE



*Indeterminate Output Level

Figure 2-2

Figure 2-2 illustrates the LOGSIM interpretation of a fall decay time. The output responds to a LOW to HIGH input transition by switching from LOW to HIGH in some rise delay time t_r. However, when the input returns to LOW and the gate's output node is in a holding mode, the output voltage actually drifts slowly toward a LOW state. If a user specifies a fall decay time, the program will interpret it as being the time, t_{df}, in which the output drifts from the HIGH level to the logic HIGH threshold level. After that time LOGSIM assigns the output level an indeterminate state, and it remains at that state until another input pulse causes the output to change to a specific state. The user may choose to ignore the decay of the output, and may omit the decay time specification. The program then interprets the decay time as infinite and the output will hold its previous state indefinitely.

2.2 Sequence of LOGSIM Operations

LOGSIM was designed in the form of three main programs. The three programs may be run together at one time, or may be run at different times, as long as the data to be transferred between programs is preserved. Figure 2-3 illustrates the general flow of execution of the three programs: the Preprocessor, the Simulator, and the Postprocessor.

2.2.1 Preprocessing

During preprocessing the input data is read in the form of 72 column card images, where the first four characters of each record identify the record type. The input device may be changed at any point during the input of data. The Preprocessor Program validates the data and provides diagnostic messages whenever necessary. All of the data will be read and interpreted, regardless of errors, eliminating the need for repetitive runs to locate coding errors. The Preprocessor outputs the image of the data records under appropriate headers enabling the user to quickly determine how the data has been interpreted and if the logic has been specified correctly. As a user's option, the Preprocessor will also generate a connectivity list showing each gate and its loads.

After the data is read and validated, the Preprocessor outputs a parameter record, images of all of the input data records, and a list of all of the logic element names to a data tape to be used later by the Postprocessor. The Preprocessor then packs and converts the circuit data into a form compatible with the Simulator Program's input format. This data is saved on a binary disk file awaiting simulation.

LOGSIM FLOW OF EXECUTION

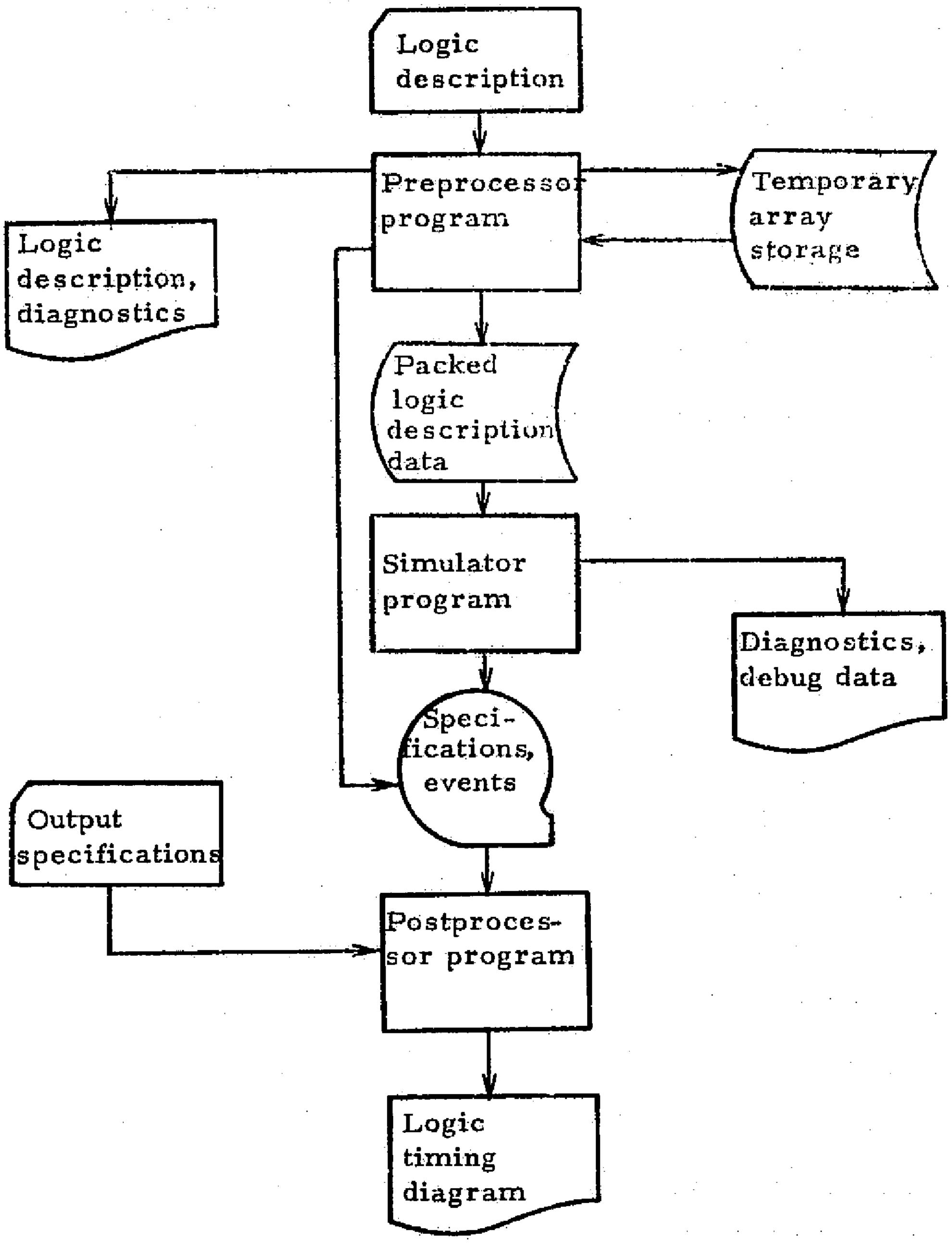


Figure 2-3

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

2.2.2 Simulation

The Simulator accesses only the disk file created by the Preprocessor for its input data. The first record of this file specifies the desired simulation options and the organization of the data arrays describing the logic network. After these arrays have been stored in memory, the program proceeds by initializing logic element outputs. Initial levels of the input signal generators have been specified by the user along with any other element output levels which are known. The program first assigns initial output levels to the logic elements based on the initial generator levels. If some initial levels cannot be determined, program options determine whether simulation is to halt, or to proceed with some elements initially indeterminate. The initial levels which the user has specified are now compared with the level assignments established by the program. Another program option specifies whether inconsistent levels cause simulation to halt, or to continue using the user's initial levels. Upon completion of initialization of the outputs, the program writes the initial level of all logic elements to the data tape being generated for the Postprocessor.

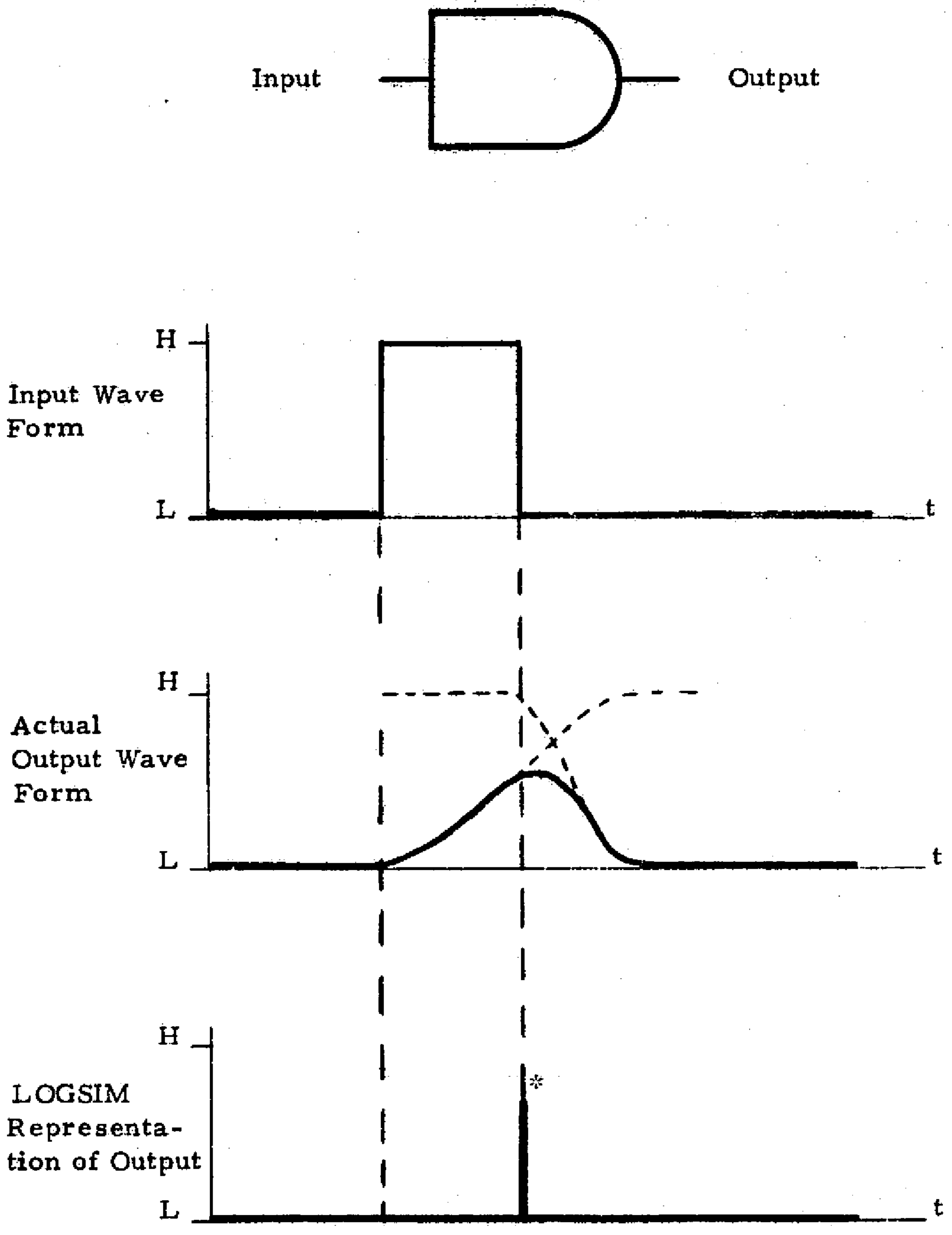
The actual simulation begins by finding the first generator level changes and entering these new levels into an element status table. Each element loading one of the changed signals is then examined to determine if its output is affected. Appropriate delay times are added to the current time to find the times at which the future events are to be scheduled. When all loads have been examined, the current time is advanced to that of the next scheduled events, and the process repeats itself. As the new level of each scheduled event is entered in the element status table, it is also recorded on the events tape for the Postprocessor. Simulation continues until the current time has been advanced beyond the time limit set by the user.

In addition to recording logic events, the simulator detects possible spike conditions which may cause faulty logic operation. Figure 2-4 illustrates how this condition arises when the input of a logic element changes too rapidly for the output to follow. The simulator will always record a notice of the spike condition and maintain the element's output at its original level. The user may choose, at postprocessing, to have these notices printed with the timing diagram output.

2.2.3 Postprocessing

The user specifies the format of the printed output with a small

LOGSIM REPRESENTATION OF A LOGIC SPIKE CONDITION



*Spike notice recorded at this time

Figure 2-4

set of data cards. Since the Postprocessor is completely independent of the preceding program segments, it may be run more than once to reprocess the events tape of a given simulation. Thus, the Preprocessor and Simulator need not be rerun to produce timing diagrams in a new format or of different logic elements.

Among the items specified by the Postprocess data cards, are the characters which the user prefers to have printed indicating HIGH, LOW, and indeterminate levels, and how each is to be shifted on the timing diagram. The user also chooses which element outputs are to be printed, in what order, with what spacing, and over what simulation time intervals.

The levels of any of the circuit elements can be compared to specified reference levels at any time. When the levels do not compare a message is printed, and the output continues unless the user has specified that it be terminated on a mismatch.

The Postprocessor reads the list of circuit element names and their initial conditions from the data tape, and prints a list of the signals to be output before starting a timing diagram. The timing diagram begins with the initial conditions of each element on the timing diagram or optionally a list of all initial conditions. Logic events are read from the data tape and the status of the selected elements is printed for each time within the specified time interval at which an event occurs. The timing diagram is thus compressed in that there is no printout for times at which logic is not changing states (except for the start and end times of each time slot, at which the status is always printed). After each timing diagram is complete, the program determines if another timing diagram is to be printed for another set of element outputs and time slots. If so, the data tape is rewound and the program generates another timing diagram according to the user's specifications. In this manner an unlimited number of timing diagrams can be printed by a single Postprocessor run.

3. PREPARATION OF LOGSIM INPUT DATA

The LOGSIM program has been designed to minimize the user effort required for logic simulation, while retaining flexibility in handling new types of logic components. To minimize transcription errors and simplify interpretation of program inputs and outputs, each element of the user's logic network is identified by a unique alphanumeric name. This name also represents the primary output signal of that gate. The logic network is defined to the program by providing the following data for each component:

- component name
- component parameter specifications
- input signal names

The LOGSIM Preprocessor analyzes this data to identify errors such as duplicated or misspelled element names, missing input data, and inconsistent parameter specifications.

3.1 Program Capabilities and Limitations

The user should be aware of several program limitations while constructing logic networks for simulation. Program capacity is limited by the amount of dimensioned memory, the exact number of elements simulated during one run being a function of the complexity of interconnections, the number of input signal specifications, and size of ROM truth tables. A capacity of 3000 to 5000 elements can normally be expected with the program's current dimensions. Larger networks must be divided into two or more sections for separate simulation. The only other inherent limitation restricts to 255, the number of rise time values, fall time values, rise decay time values, and fall decay time values.

The user should configure his logic so that all functions are performed by components which are either elements of the program's internal library or are specially defined as new gate types or ROMs. It should be noted that all LOGSIM components have a single output. The complement of this output, however, may be specified as an input to other elements by preceding the signal's name with a slash (/). LOGSIM will automatically generate this complement signal and distribute it where required, effectively providing the user with two

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

outputs from the component. Any other forms of multiple output components must be broken up before being defined to the program.

As indicated in Section 2, LOGSIM provides the user with a variety of program options. Table 3-1 provides a summary of these options.

3.2 Component Type Definitions

The standard LOGSIM component library of 15 logic element types spans both bipolar and MOS technologies. These element types are completely described in Tables 3-2 and 3-3. All logic elements used in a simulation run must conform to one of the library types or to a type which is specially defined. Thus some logic functions may have to be synthesized by using logic types available in the LOGSIM library to provide an equivalent logic function. Any number of inputs may be specified for those logic types in Table 3-2 that do not require a fixed number of inputs. All inputs must be specified for logic types with a fixed number of inputs.

New gate (NEWGATE) types may be defined for a particular simulation run by the eight-character sequence, Q1 to Q8, described in Table 3-4, where each Q may represent any of the output states defined. Only the states Q_1 through Q_4 are used to specify an unclocked element. States Q_1 through Q_8 define a clocked element, where Q_1 through Q_4 define the possible outputs when the clock is at a LOW state, and Q_5 through Q_8 define the outputs for a HIGH clock state. Specific logic types which may be defined are:

- Majority logic
- "Any LOW sets" logic
- "Any HIGH sets" logic
- Three input flip-flops
- Five input flip-flops

These are defined in detail in Section 4.1.5. There is no limit to the number of inputs that a new gate type element may have, however, if it has been defined as clocked (Q_1 through Q_8 defined), the last input will be considered the clock.

SUMMARY OF PROGRAM OPTIONS

Options Available at Preprocessing

OPTION	ON/OFF	RESULT
FROMTO	On	A connectivity list is printed during pre-processing.
	Off	Connectivity list is not printed.
IGNORE	On	Simulation continues despite indeterminate initial levels.
	Off	Simulation terminates if all initial levels cannot be established.
NONCON	On	Simulation accepts user specified initial levels if inconsistent with those established by the program.
	Off	Simulation terminates if there are any conflicting initial states
FOLLOW	On	Simulator tries to establish ROM output for any indeterminate inputs.
	Off	Simulator assigns ROM output level indeterminate for any indeterminate inputs.
DEBUG	On	Debug data is printed during simulation.
	Off	No simulator debug data is printed.

Table 3-1a

SUMMARY OF PROGRAM OPTIONS (Continued)

Options Available at Postprocessing

OPTION	ON/OFF	RESULT
PREPIN	On	Preprocessor input data is printed during postprocessing.
	Off	Preprocessor input data is not printed during postprocessing.
POSTIN	On	Postprocessor input data is printed.
	Off	Postprocessor input data is not printed.
SPIKE	On	Spike notices are printed.
	Off	Spike notices are not printed.
INITIAL	On	All initial element levels are printed before first timing diagram.
	Off	Timing diagram element initial levels are printed in each timing diagram
COMSTOP	On	Output ceases and message is printed on first level comparison mismatch.
	Off	A message is printed for each level comparison mismatch.

Table 3-1b

LOGSIM LOGIC ELEMENT TYPE LIBRARY

LOGSIM NAME	LOGIC TYPE	LOGIC DESCRIPTION
OR	OR	Any input HIGH will yield a HIGH output and all inputs LOW will yield a LOW output.
NOR	NOR	Any input HIGH will yield a LOW output and all inputs LOW will yield a HIGH output.
AND	AND	Any input LOW will yield a LOW output and all inputs HIGH will yield a HIGH output.
NAND	NAND	Any input LOW will yield a HIGH output and all inputs HIGH will yield a LOW output.
MOR	MOS NOR	Gate with a clocked load device. The last input listed is treated as the clock.
MORT	MOS NOR with transmission gate	Gate with a clocked load device and clocked internal transmission gate. The last input listed is treated as the clock.
MAND	MOS NAND	Gate with a clocked load device. The last input listed is treated as the clock.
MANT	MOS NAND with transmission gate	Gate with a clocked load device and clocked internal transmission gate. The last input listed is treated as the clock.
CFF	Clocked flip-flop	Clocked, three-input, set-reset flip-flop. The third input is treated as the clock, and the flip-flop is enabled during a negative clock. The first input is assumed to be the set input with the second input assumed to be the reset line. All three inputs must be specified.

Table 3-2

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

LOGSIM LOGIC ELEMENT TYPE LIBRARY
(continued)

LOGSIM NAME	LOGIC TYPE	LOGIC DESCRIPTION
EXOR	EXCLUSIVE OR	Two-input gate in which either input HIGH will yield a HIGH output. Both inputs LOW or HIGH will yield a LOW output.
SRFF	Set-reset flip-flop	Two-input flip-flop. The first input is assumed to be the set input and the second input is assumed to be the reset input. When both inputs are HIGH, the output is undefined.
JKFF	J-K flip-flop	Two-input flip-flop similar to SRFF, but when both inputs are HIGH, the output changes state. The first input is assumed to be the J input and the second input is assumed to be the K input.
MCLK	MOS transmission gate	Two-input flip-flop similar to SRFF, but when both inputs are HIGH, the output changes stage. The first input is assumed to be the J input and the second input is assumed to be the K input.
TJKF	Triggerable J-K flip-flop	Three-input triggerable J-K flip-flop similar to CFF. The third input is assumed to be the clock input, and the flip-flop is enabled on a positive-to-negative transition of the clock.
CJKF	Clocked J-K flip-flop	Five input J-K flip-flop similar to TJKF with direct set and reset lines. The third input is assumed to be the set, the fourth, the reset; and the fifth, the clock. Enabling signals on these lines take effect in zero time.

Table 3-2
(continued)

LOGSIM LIBRARY LOGIC TRUTH TABLES

OR

A	B	out
L	L	L
L	H	H
H	L	H
H	H	H

NOR

A	B	out
L	L	H
L	H	L
H	L	L
H	H	L

AND

A	B	out
L	L	L
L	H	L
H	L	L
H	H	H

NAND

A	B	out
L	L	H
L	H	H
H	L	H
H	H	L

MOR

clock	A	B	out
L	L	L	Hold
	L	H	L
	H	L	L
	H	H	L
H	L	L	H
	L	H	L
	H	L	L
	H	H	L

MORT

clock	A	B	out
L	all states		Hold
H	L	L	H
	L	H	L
	H	L	L
	H	H	L

MAND

clock	A	B	out
L	L	L	Hold
	L	H	Hold
	H	L	Hold
	H	H	L
H	L	L	H
	L	H	H
	H	L	H
	H	H	L

MANT

clock	A	B	out
L	all states		Hold
H	L	L	H
	L	H	H
	H	L	H
	H	H	L

Table 3-3

LOGSIM LIBRARY LOGIC TRUTH TABLES
(continued)

CFF

clock	S	R	out
H	all states		Q_{t-1}
L	L	L	*
	L	H	L
	H	L	H
	H	H	Q_{t-1}

JKFF

J	K	out
L	L	Q_{t-1}
L	H	L
H	L	H
H	H	$\bar{Q}_{t-1}$

EXOR

A	B	out
L	L	L
L	H	H
H	L	H
H	H	L

MCLK

ϕ	A	out
L	L	Hold
	H	
H	L	L
	H	H

TJKF

T^+	J	K	Q_t
H	all states		Q_{t-1}
L	L	L	Q_{t-1}
	L	H	L
	H	L	H
	H	H	$\bar{Q}_{t-1}$

CJKF

T^+	J	K	Q_t
H	all states		Q_{t-1}
L	L	L	Q_{t-1}
	L	H	L
	H	L	H
	H	H	$\bar{Q}_{t-1}$

SRFF

S	R	out
L	L	Q_{t-1}
L	H	L
H	L	H
H	H	*

S_D	R_D	Q
L	L	*
L	H	H
H	L	L
H	H	Q

Table 3-3
(continued)

NEWGATE TRUTH TABLE OUTPUT DEFINITION

Truth Table Output Character String

CLOCK LOW				CLOCK HIGH			
Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8

Table 3-4a

Output State Definitions

STATE, Q_n	DEFINITION
Indeterminate	Level is neither HIGH nor LOW
Low	Logic LOW level
High	Logic HIGH level
Same	Same as current level
Complement	Complement of current level
Holding	Level holding at current level and may decay, see Section 2.1

Table 3-4b

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

Complex logic functions can be described for a simulation run as read-only-memories. Every ROM used in the network must have exactly the number of inputs the ROM was originally defined to have. Each possible combination of input states to a ROM type will then address a unique truth table location. When the ROM type is defined, one of the four possible ROM output states listed in Table 3-5 is assigned to each location of its truth table. ROM elements may have delay and decay time values specified as may other components. In addition, the user may specify rise and fall time delays to be applied to particular inputs of any ROM element. LOGSIM will then delay these inputs as required before allowing them to affect the ROM output. The total delay of such a signal propagating through the ROM will be the sum of the input and output delay. This feature permits convenient simulation of complex circuits having different propagation delays associated with different inputs.

3.3 Time Specifications

Times at which network input generators are to change, times of delay and decay time sets, Postprocessor compare and print times, and the time at which simulation is to terminate should all be specified in the same units. These values must be zero or positive whole numbers less than 2^{31}. The unit of time may be defined to the program for printout with output times. Simulation will always begin with time at zero, although the times for which printout is generated by the Postprocessor are specified by the user.

Since the simulation proceeds from event time to event time, rather than from one time unit to the next, the magnitude of the specified times will not in itself affect the run-time required for simulation. Run-time is related to the number of different times at which events occur. Delay and decay time sets should therefore be rounded to contain no more figures than are in fact significant. This will increase the likelihood of simultaneous events and decrease the required run-time.

3.4 Preparation of the Logic Network Description

Once the user has a network schematic diagram, and some knowledge of the network logic, the network can be conveniently described in an orderly form which will be easily converted to formatted card data. To best illustrate how this is accomplished, an orderly description of a sample MOS logic network is developed in

READ-ONLY MEMORY OUTPUT STATE DEFINITIONS

STATE	DEFINITION
Indeterminate	Level is neither HIGH nor LOW
Low	Logic LOW level
High	Logic HIGH level
Holding	Level holding at current state and may decay, see Section 2.1

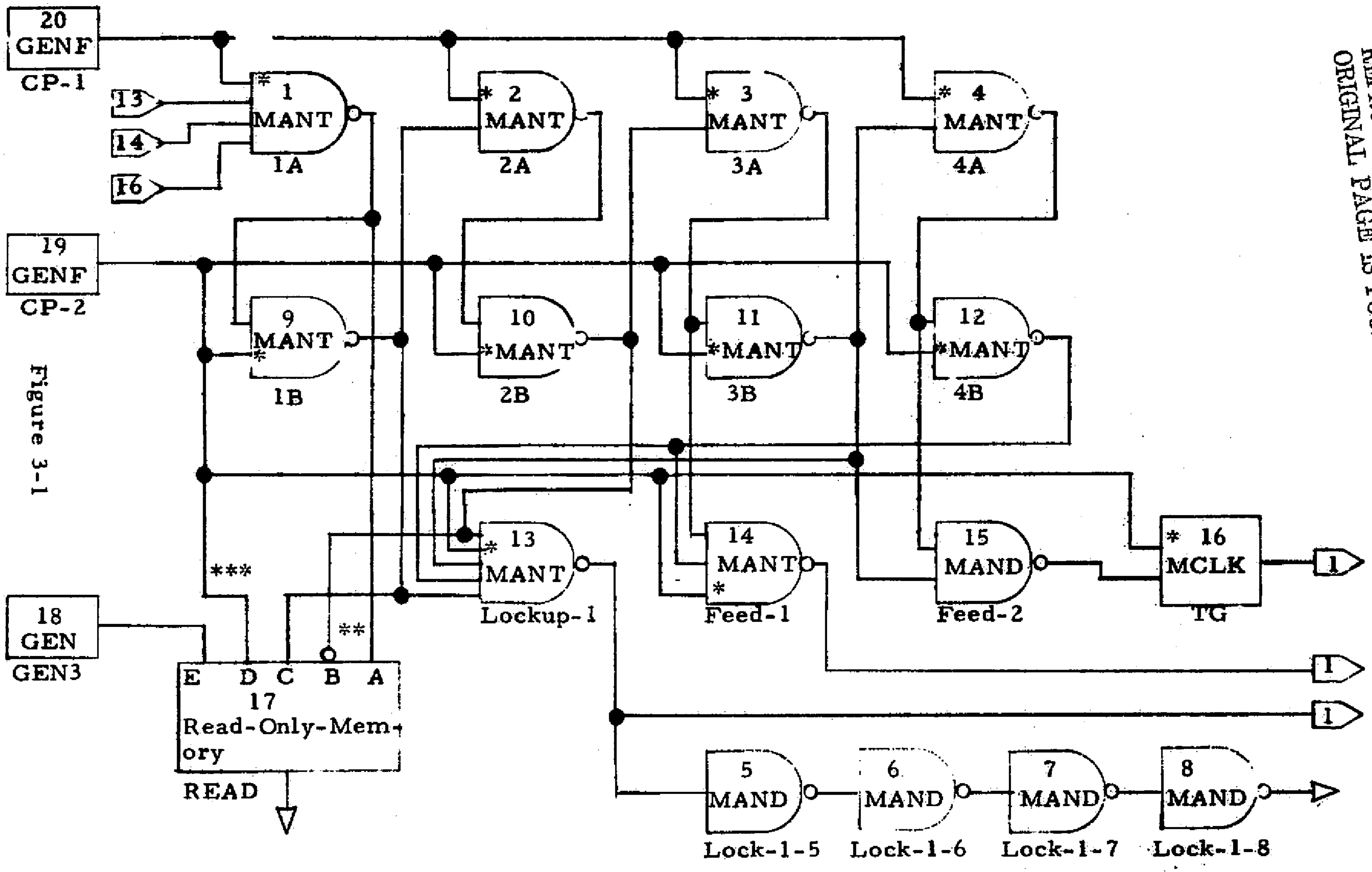
Table 3-5

this discussion. The conversion of the network description to input data and the actual card formats are discussed in Section 4.

The circuit illustrated in Figure 3-1 is a simple MOS logic network made up of 20 elements. Note that each element is numbered for easy reference in this discussion and labelled with the component type names. In addition, a unique name for each element is printed below each element block. Elements 1 through 16 are elements corresponding to the logic library types. Element 17 is a five-input read-only-memory which has the truth table as shown in Table 3-6. Input B of the ROM is the complement of element 10, and the output of element 19 is delayed at the ROM input D. Elements 18 through 20 are signal generators. Generators in LOGSIM represent the input signals driving the network, but are treated as actual network elements. Element 18 is a generator representing a signal pattern as illustrated in Figure 3-2a. Elements 19 and 20 are function generators representing periodic input signals as illustrated in Figure 3-2b and 3-2c respectively. (Subsequently, this type of generator shall be referred to as a generator function, or GENF.) The user will note that the periodic portions of the signals of the generator functions do not occur until some starting times, 25 and 50 respectively, and then each runs periodically for 18 periods.

With this preliminary information, and delay and decay time values which the user sets according to his knowledge of the circuit, an ordered list of the information can be made. The list for the sample MOS logic network is shown in Table 3-7. The user may note that each element, except the automatically constructed delay gate and inverter which input the ROM, is given a name. The name may be 1 to 8 alphanumeric characters, with one exception. Looking back at the circuit diagram in Figure 3-1 the user will see that one input to the ROM is delayed (input D) and another is an inverter signal (input B). The signal from generator function named CP-1 (element 19) is delayed, and the output of a MANT gate named 2B (element 10) is inverted. Two gates will be constructed automatically, representing a delay element and an inverter, by the user merely indicating those inputs to the ROM as a delayed CP-1 output, *CP-1, and an inverted 2B output, /2B (* = delayed, / = inverted). The Preprocessor assigns the name *CP-1 to the delay gate, and the name /2B to the inverter. The exception to the 1 to 8 character name rule is that any gate which inputs an automatically generated delay gate or inverter (such as CP-1 and 2B in this example) may have no more than 7 alphanumeric characters in its name. This allows the delay gate and inverter names to stay within the 1 to 8 character limits.

SAMPLE MOS LOGIC NETWORK



*Clock inputs

**Input B to element 17 is complement of element 10 output

***Input D to element 17 is delayed

Figure 3-1

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

TRUTH TABLE FOR THE READ-ONLY-MEMORY IN THE SAMPLE MOS LOGIC NETWORK

ROM Inputs (Fig. 3-1)					
E	D	C	B	A	Output
0	0	0	0	0	0
0	0	0	0	1	?
0	0	0	1	0	Holding
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	Holding
0	0	1	1	1	1
0	1	0	0	0	Holding
0	1	0	0	1	Holding
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	?
0	1	1	1	0	?
0	1	1	1	0	Holding
0	1	1	1	1	Holding
1	0	0	0	0	0
1	0	0	0	1	1
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	0
1	0	1	1	1	1
1	1	0	0	0	?
1	1	0	0	1	?
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

Table 3-6

Element 18 is a Generator Outputting the Following Signal Pattern

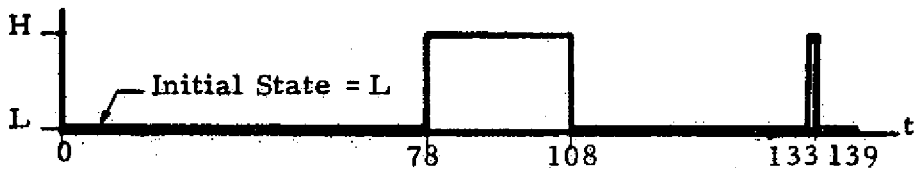


Figure 3-2a

Element 19 is a Function Generator Outputting 18 Repetitions of the Following Periodic Signal, Starting at Time = 25

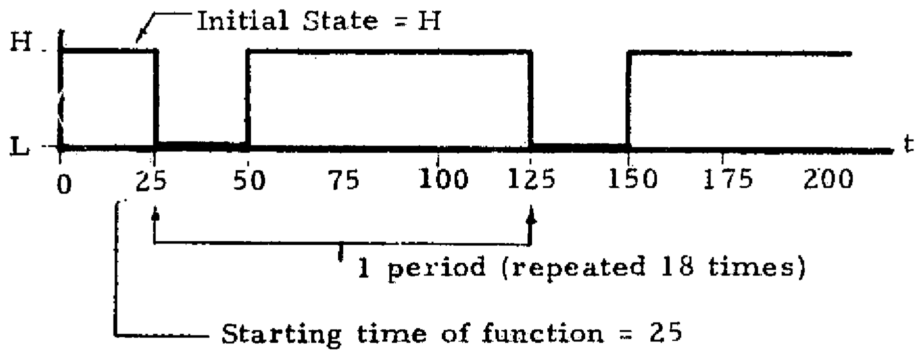


Figure 3-2b

Element 20 is a Function Generator Outputting 18 Repetitions of the Following Periodic Signal, Starting at Time = 50

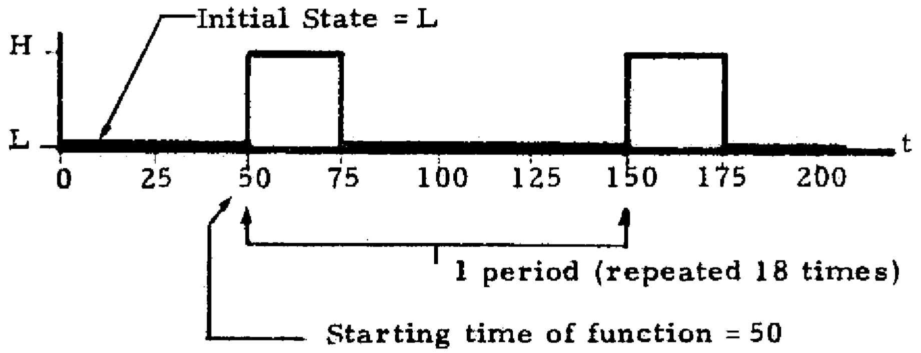


Figure 3-2c

ELEMENT DESCRIPTION LIST FOR THE SAMPLE
MOS LOGIC NETWORK

Gate #	Gate Name	Gate Type	Initial State	Rise Delay	Fall Delay	Rise Decay	Fall Decay
1	1A	MANT	Low	5	5	↑	↑
2	2A	MANT	Low	5	5		
3	3A	MANT	Low	5	5		
4	4A	MANT	Low	5	5		
5	Lock-1-5	MAND	?	0	0		
6	Lock-1-6	MAND	?	0	0		
7	Lock-1-7	MAND	?	0	0		
8	Lock-1-8	MANT	?	1	6	00	00
9	1B	MANT	?	5	5		
10	2B	MANT	?	5	5		
11	3B	MANT	?	5	5		
12	4B	MANT	?	5	5		
13	Lockup-1	MANT	?	5	5		
14	Feed-1	MANT	?	5	5		
15	Feed-2	MAND	?	0	0		
16	TG	MCLK	?	5	5	↓	↓
17	READ	ROM5	?	1	1	109	100
18	GEN-3	GEN	Low				
19	CP-2	GENF	High				
20	CP-1	GENF	Low				

Table 3-7

The five-input read-only-memory was named READ and its logic type named ROM5. The logic type name used for a NEWGATE or a read-only-memory need have no significance other than a name by which it can be referenced. Such names may have 1 to 4 alphanumeric characters, but must not be identical to any of the logic library names.

From the element description list in Table 3-7 three lists of rise and fall delay and decay times can be made. For the sample MOS logic circuit these lists are illustrated in Table 3-8. The lists are formed merely by listing each rise and fall delay and decay time needed to describe the logic elements in the circuit, and assigning an index number to each. A rise delay time of 7 and a fall delay time of 3 are also defined in Table 3-8 to be referenced as the delays for ROM's automatically delayed input (delay gate *CP-2). The index numbers may be any whole numbers from 1 to 255. Note that the component library truth tables in Table 3-3 allow holding modes for MOR, MORT, MAND, MANT, and MCLK component types. For the sample MOS logic network the decay times are assigned as infinite for the MAND, MANT, and MCLK element, and decay times need not be specified. The ROM, however, does have holding modes, and rise and fall decay times are defined.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

DELAY AND DECAY TIME SETS FOR THE SAMPLE MOS LOGIC NETWORK

Rise Delay Times and Index Numbers

Index	Rise Delay Time
1	5
2	0
3	1
4	7

Table 3-8a

Fall Delay Times and Index Numbers

Index	Fall Delay Time
1	5
2	0
3	6
4	1
5	3

Table 3-8b

Decay Times and Index Numbers

Index	Decay Time
1	109
2	100

Table 3-8c

4. DATA CARD FORMATS

Now that the logic has been prepared, and the network has been described, the data must be translated into a form to be input to the LOGSIM programs, namely a deck of punched data cards. As previously mentioned, only two of the LOGSIM programs require data card decks, the Preprocessor and the Postprocessor. Figures 4-1 and 4-2 illustrate the arrangement of data cards for preprocessing and postprocessing. The following discussions illustrate the forms of the data decks for the Preprocessor and the Postprocessor. A description of each card and its format are presented in the order in which the cards must appear in the decks. References will be made frequently to examples of the data cards for the sample MOS logic network discussed in Section 3.

4.1 Preprocessor Data Cards

The following paragraphs describe all of the possible cards which the Preprocessor can use.

4.1.1 Title (NAME) Card - Optional

Description

If the user desires a particular heading to be printed at the top of the preprocessing and postprocessing output documents, a NAME card must be used to specify the heading. If this card is omitted, the words LOGIC SIMULATION will appear as the heading.

Format

Columns 1 - 4	- NAME - required.
9 - 72	- any descriptive heading.

Example

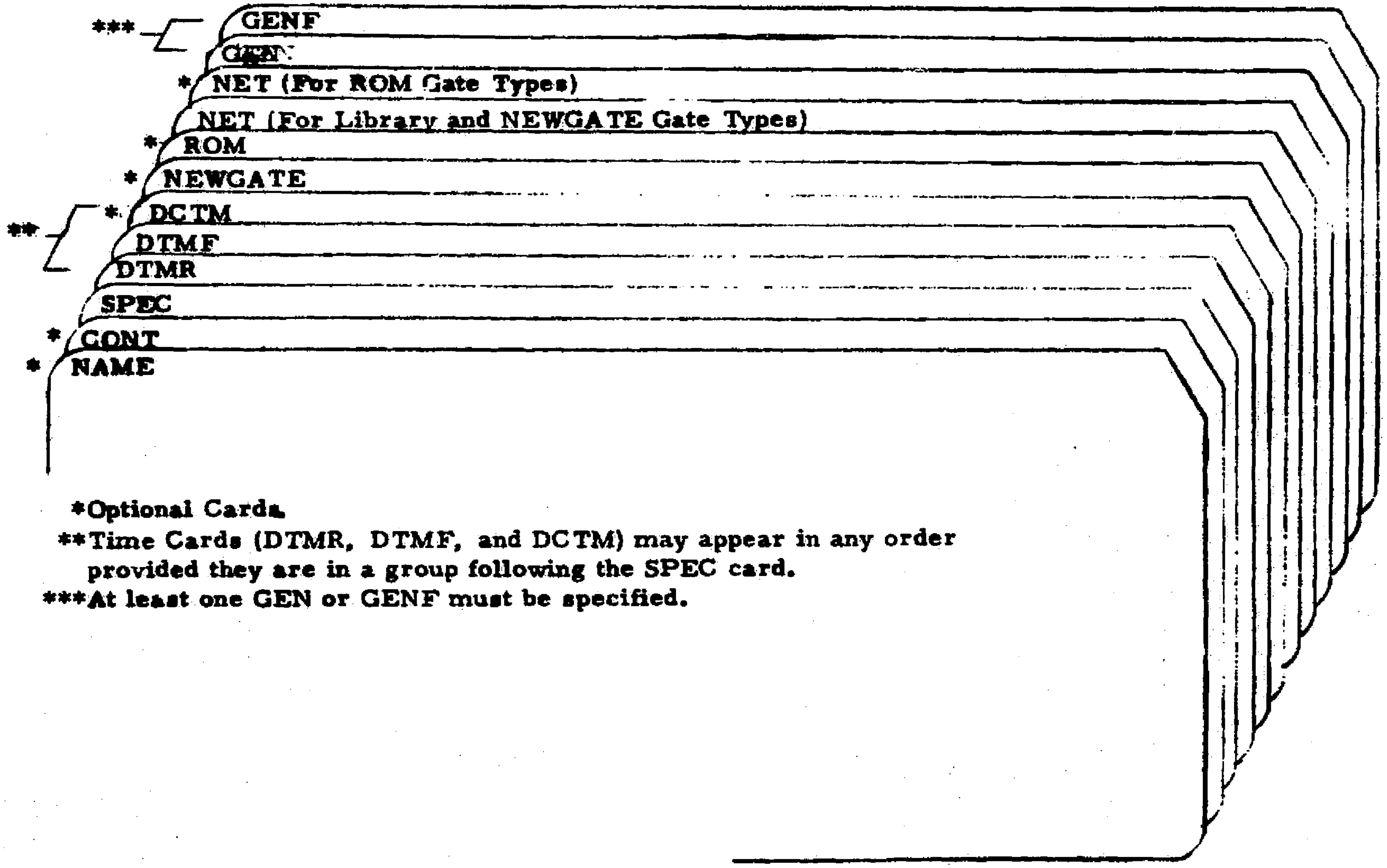
See Figure 4-3 for the data card coding for the sample MOS logic network.

4.1.2 Control (CONT) Card - Optional

Description

The user may choose any of the following preprocessing and simulation

LOGSIM PREPROCESSOR DATA CARD ARRANGMENT



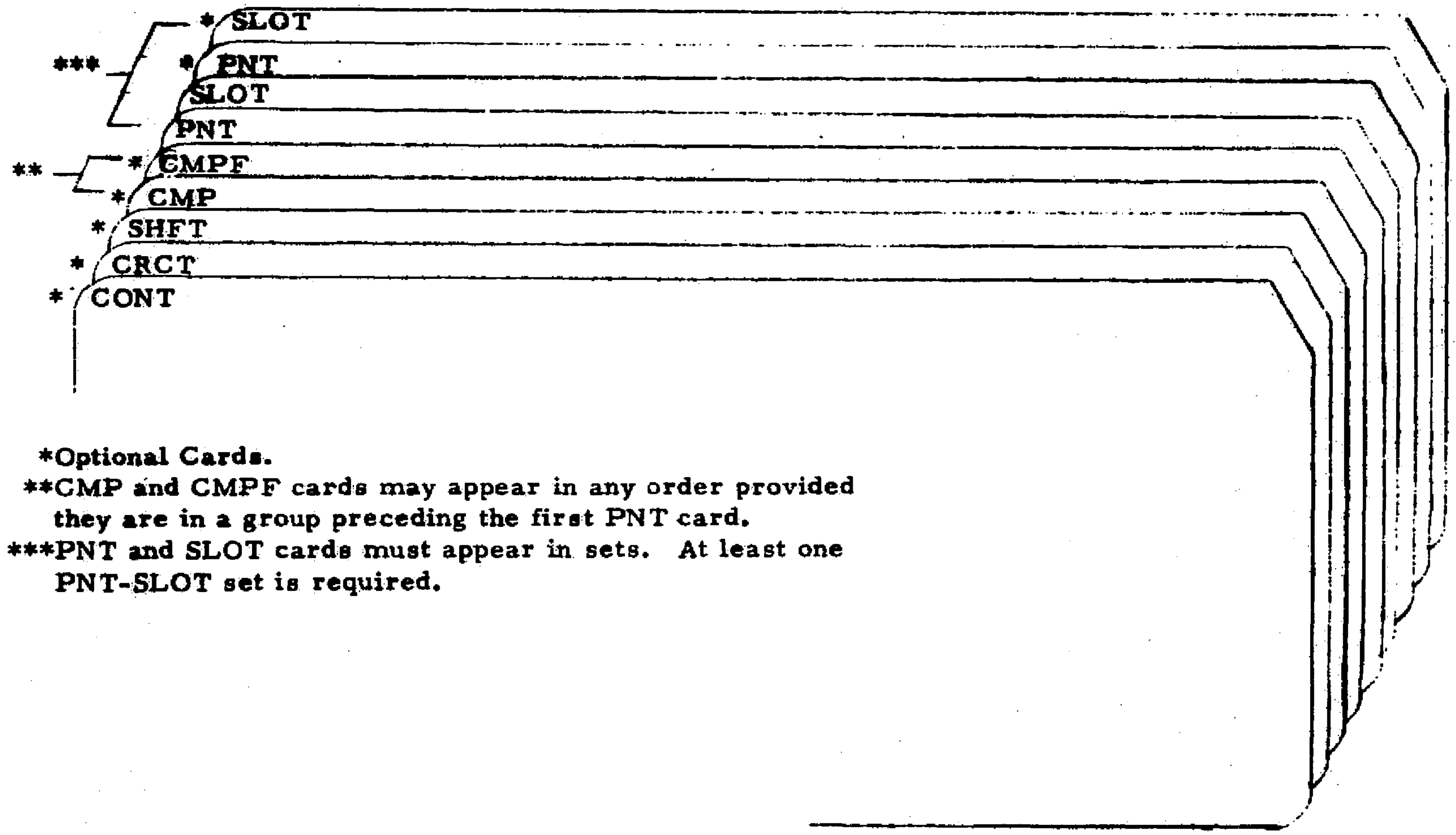
***
GENF
GEN
* NET (For ROM Gate Types)
NET (For Library and NEWGATE Gate Types)
* ROM
* NEWGATE
**
* DCTM
DTMF
DTMR
SPEC
* CONT
* NAME

*Optional Cards.
**Time Cards (DTMR, DTMF, and DCTM) may appear in any order provided they are in a group following the SPEC card.
***At least one GEN or GENF must be specified.

Figure 4-1

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

LOGSIM POSTPROCESSOR DATA CARD ARRANGEMENT



*Optional Cards.
**CMP and CMPF cards may appear in any order provided they are in a group preceding the first PNT card.
***PNT and SLOT cards must appear in sets. At least one PNT-SLOT set is required.

Figure 4-2

DATA CARDS FOR PREPROCESSING OF THE SAMPLE MOS LOGIC NETWORK

```
NAME    ***** LOGIC SIMULATION OF THE SAMPLE MOS LOGIC NETWORK *****
CONT    FROMTO  IGNORE  FOLLOW
SPEC    MSEC 1825,17,3
DTMR    1,5
DTMR    2,0
DTMR    3,1
DTMR    4,7
DTMF    1,5
DTMF    2,0
DTMF    3,6
DTMF    4,1
DTMF    5,3
DCTM    1,100
DCTM    2,109
ROM     1    5ROM50*H100H1HH01**HH01010101**111111
NET     1A          MANT L 1,1                TG,LOCKUP-1,FEED-1,CP-1
NET     2A          MANT L 1,1                1B,CP-1
NET     3A          MANT L 1,1                2B,CP-1
NET     4A          MANT L 1,1                3B,CP-1
NET     LOCK-1-5    MAND * 2,2                LOCKUP-1
NET     LOCK-1-6    MAND * 2,2                LOCK-1-5
NET     LOCK-1-7    MAND * 2,2                LOCK-1-6
NET     LOCK-1-8    MAND * 3,3                LOCK-1-7
NET     1B          MANT * 1,1                1A,CP-2
NET     2B          MANT * 1,1                2A,CP-2
NET     3B          MANT * 1,1                3A,CP-2
NET     4B          MANT * 1,1                4A,CP-2
NET     LOCKUP-1    MANT * 1,1                4B,1B,2B,3B,CP-2
NET     FEED-1      MANT * 1,1                [illegible]A,4B,CP-2
NET     FEED-2      MAND * 2,2                4A,3B
NET     TG          MCLK * 1,1                FEED-2,CP-2
NET     READ        ROM5 * 3,4,1,2            1A,/2B,1B,*CP-2,4,5,GEN-3
GEN    L GEN-3       78,H,10B,L,138,H,139,L
GENF   H CP-2        18        25        25LLLH
GENF   L CP-1        18        50        25HLLL
```

Figure 4-3

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

options by listing them on the control card.

- Gate Loads Connectivity List (FROMTO). This is the option which indicates to the Preprocessor that the connectivity list of the circuit should be printed.

- Initial Conditions Override (IGNORE). This option instructs the Simulator to continue despite any indeterminate initial element levels. If not specified, the Simulator will terminate if all initial states cannot be determined.

- Inconsistent Level Check Override (NONCON) - This option instructs the Simulator to continue even if the initial gate outputs specified by the user are inconsistent with those established by the Simulator, and in such a case, to use those initial conditions specified by the user. If the option is not specified, simulation will terminate if there are any conflicting initial states.

- ROM Indeterminate Inputs Check (FOLLOW). This option specifies that the Simulator, upon encountering any indeterminate inputs to a read-only-memory, should examine all possible output states to determine the effect of these inputs. The ROM output will be considered indeterminate unless all possible outputs are the same state. If this option is not specified, the Simulator will assign the ROM output as indeterminate any time it encounters an indeterminate ROM input. The time required to perform this check may make selection of this option impractical for large ROM networks.

- Simulator Output (DEBUG). Normally the Simulator program does not cause a document to be printed. However, if this option is specified debug data will be output. Normally the user would have little use for this option, but it is included for the programmer's use.

The five preprocessing and simulation options are specified by their names: FROMTO, IGNORE, NONCON, FOLLOW, and DEBUG. The control card may be omitted completely, in which case it is assumed that no options were desired.

Format

Columns 1 - 4	- CONT - required.
9 - 16, 17 - 24, 25 - 32, 33 - 40, 41 - 48	- any of the following option names FROMTO, IGNORE, NONCON, FOLLOW, DEBUG

Any number of the options may be specified in any order. However, the option names specified must be left-justified in one of the five card fields listed above.

Example

For the sample MOS logic network the options FROMTO, IGNORE, and FOLLOW were chosen. See Figure 4-3.

4.1.3 Specifications (SPEC Card - Required)

Description

The purpose of the specifications card is to relay vital parameters to the three LOGSIM programs. The information includes the time unit, maximum simulation time, the number of gates excluding automatically constructed delays and inverters, and the number of generators and generator functions. This is a required card. If it is omitted, the Preprocessor will continue reading cards, searching for a SPEC card.

Format

Columns 1 - 4	- SPEC - required
9 - 12	- time unit to be printed in time column on output, 1 to 4 alphanumeric characters.
Beginning col. 14	- maximum simulation time, number of gates, number of generators. All are required, listed in order, separated by commas, with no internal blanks.

Example

For the sample MOS logic network the parameters are as follows:

- time unit = MSEC (microseconds)
- maximum time = 1825
- number of gates = 17
- number of generators = 3

Figure 4-3 illustrates how these parameters are listed on the SPEC Card.

4.1.4 Time (DTMR, DTMF, DCTM) Cards - Required

Description

The purpose of the time cards is to relay the complete lists of rise delay, fall delay, and decay times to the program. These lists correspond to the lists illustrated in Table 3-8 for the sample MOS logic circuit. Each DTMR card specifies one rise delay time and its index number. The same applies to the DTMF and DCTM cards for fall delay times and decay times respectively. For simulation to occur at least one DTMR and one DTMF card must be present. The time cards may appear in any order as long as they are all together following the SPEC card.

Format

Columns 1 - 4	- DTMR, DTMF, or DCTM - required.
Beginning col. 9	- time index number (any whole number from 1 to 255) and the time value. Both are required, separated by commas, with no internal blanks.

Example

Figure 4-3 illustrates how the information from Table 3-8 is translated to time card specifications for the sample MOS logic network. The user will note that the index numbers in one time set may be repeated in another time set. However, the index numbers within a set, for example the DTMR set, must all be different. Thereby the user is allowed 255 of each type of time card.

4.1.5 New Gate Specification (NEWGATE) Cards - Optional

Description

As previously discussed, the specification of a NEWGATE is possible when the logic types in the internal gate library do not apply to a logic type in a circuit to be simulated. The user may think of a NEWGATE specification merely as a means of extending the internal logic library.

The information required to specify a NEWGATE is simply a name for the logic function, a special code, and the logic truth table. The logic name may be 1 to 4 alphanumeric characters, as long as it is not the same as a library gate type name or another NEWGATE name. The user must specify a special code which indicates how the truth table should be interpreted. Table 4-1 lists the special code characters to be used and their meanings. Tables 4-2, 4-3 and 4-4 further illustrate the interpretation of the special code.

The NEWGATE truth table list may be up to 8 characters (corresponding to the entries in descending order in the output column of a truth table diagram) in length, allowing for clocking of the logic. Each entry must be one of the characters listed in Table 4-5.

Format

Columns	1 - 7	- NEWGATE - required
	9 - 12	- the logic name - required, 1 to 4 alphanumeric characters, left-justified in the field.
	14	- special code character from Table 4-1.
	17 - 24	- 1 to 8 truth table entries, 1 character each from Table 4-5, - required.

Example

Since the sample MOS logic network does not need any NEWGATE specifications there is no NEWGATE card shown in Figure 4-3. Figure 4-4a illustrates how a typical NEWGATE specification might appear.

NEWGATE SPECIAL CODE CHARACTER INTERPRETATIONS

NEWGATE Special Code Characters	Interpretation
ҍ (blank)	No special type. Truth table will be interpreted in normal manner.
F	Three input flip-flop. Truth table interpreted as for library TSKF.
M	Majority logic. See Table 4-2 for truth table interpretation.
L	Any LOW sets. See Table 4-3.
H	Any HIGH sets. See Table 4-4.
C	Five input flip-flop. Truth table interpreted as for library CJKF.

Table 4-1

MAJORITY LOGIC INTERPRETATION

Clock	Inputs	Output
LOW	All inputs LOW	Q_1
	Majority of inputs LOW	Q_2
	Majority of inputs HIGH	Q_3
	All inputs HIGH	Q_4
HIGH	All inputs LOW	Q_5
	Majority of inputs LOW	Q_6
	Majority of inputs HIGH	Q_7
	All inputs HIGH	Q_8

Table 4-2

Note: If there are equal numbers of HIGH and LOW, the Simulator will assign the output as indeterminate.

"ANYLOWSETS" INTERPRETATION

Clock	Inputs	Output
LOW	Any LOW inputs	Q_1
	N/A	Q_2
	N/A	Q_3
	All HIGH inputs	Q_4
HIGH	Any LOW inputs	Q_5
	N/A	Q_6
	N/A	Q_7
	All HIGH inputs	Q_8

Table 4-3

"ANY HIGH SETS" INTERPRETATION

Clock	Inputs	Output
LOW	All LOW inputs	Q1
	N/A	Q_2
	N/A	Q_3
	Any HIGH inputs	Q_4
HIGH	All LOW inputs	Q_5
	N/A	Q_6
	N/A	Q_7
	Any HIGH inputs	Q_8

Table 4-4

NEWGATE TRUTH TABLE CHARACTERS

NEWGATE Truth Table Characters	Interpretation
*	Indeterminate state
0 (zero)	LOW state
1 (one)	HIGH state
S	Same as current state
C	Complement of current state
H	Holding in current state
ƀ (blank)	No entry

Table 4-5

EXAMPLES OF NEWGATE AND MULTICARD ROM SPECIFICATIONS

Typical NEWGATE Specifications

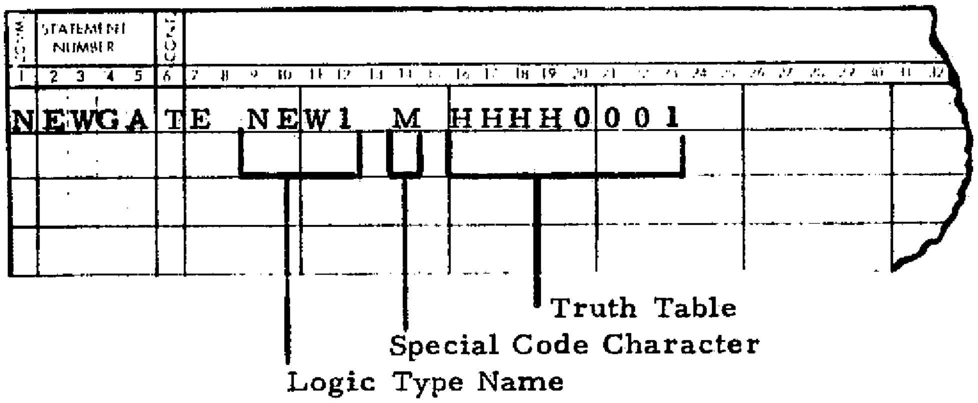


Figure 4-4a

4.1.6 Read-Only-Memory Specification (ROM) Cards - Optional

Description

A ROM specification, like the NEWGATE specification, may be considered an extension of the logic gate library. For a ROM, however, the specification is more flexible. For example, a ROM may have any number of inputs. However, for large numbers of inputs, the specification may become rather extensive. For 16 inputs there are 2^{16} or 65,536 truth table entries, requiring 1172 cards.

A ROM specification must include a logic name, the number of inputs, and a truth table. The logic type name may be 1 to 4 alphanumeric characters, but must not be the same as a library logic name, a NEWGATE name, or any other ROM name. The ROM truth table is specified in the same fashion as a NEWGATE except that it must contain the characters listed in Table 4-6. A ROM specification must contain the exact number of cards required. The cards in a ROM specification must be numbered, beginning with 1 for the first card. This enables the user to easily maintain the order of the cards in the ROM specification.

All of the cards in a ROM specification have the same basic format, except the number of inputs must be specified only on the first card. Each ROM card has space for 56 truth table entries. All 56 spaces must be filled for all cards in a ROM specification except for the last card. Table 4-7 lists the numbers of truth table entries, the number of cards required, and the number of entries on the last card, for the specification of any ROM with 1 to 16 inputs.

Format (First Card)

Columns	1 - 3	- ROM - required.
	5 - 9	- card number (=1 for first card) - required, must be right-justified in the field.
	11 - 12	- number of inputs - required, must be right-justified in the field.
	13 - 16	- name of the ROM - required, 1 to 4 alphanumeric characters, left-justified in the field.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

ROM TRUTH TABLE CHARACTERS

ROM Truth Table Characters	Interpretation
*	Indeterminate state
0 (zero)	LOW
1 (one)	HIGH
H	Holding in current state

Table 4-6

NUMBER OF ROM CARDS AND TRUTH TABLE ENTRIES REQUIRED TO SPECIFY A READ-ONLY-MEMORY WITH "n" INPUTS

Number of Inputs n	Number of Truth Table Entries 2^n	$\frac{2^n \text{ Entries}}{56 \text{ Entries per Card}}$	Number of Cards Needed	Number of Entries on Last Card
1	2	2/56	1	2
2	4	4/56	1	4
3	8	1/7	1	8
4	16	2/7	1	16
5	32	4/7	1	32
6	64	1 1/7	2	8
7	128	2 2/7	3	16
8	256	4 4/7	5	32
9	512	9 1/7	10	8
10	1024	18 2/7	19	16
11	2048	36 4/7	37	32
12	4096	73 1/7	74	8
13	8192	146 2/7	147	16
14	16384	292 4/7	293	32
15	32768	585 1/7	586	8
16	65536	1171 2/7	1172	16

Table 4-7

17 - 72	- 1 - 56 truth table entries, 1 character each from Table 4-6, - required.

Format (Continuation Cards)

If more than one card is required, each additional card has the same format as the first card except the number of inputs in columns 11-12 may be left blank. The truth table is continued on each card beginning in column 17.

Examples

In the sample MOS logic network there is a five-input ROM which must be specified. Referring to Table 4-7 the user will note that a ROM with 5 inputs requires a truth table of 32 entries and can be specified on one card. Figure 4-3 illustrates the necessary ROM card where:

Card number = 1

Number of inputs = 5

Logic type name = ROM5

The truth table follows with 32 entries taken directly from the truth table output column in Table 3-6 and specified using the permissible characters listed in Table 4-6.

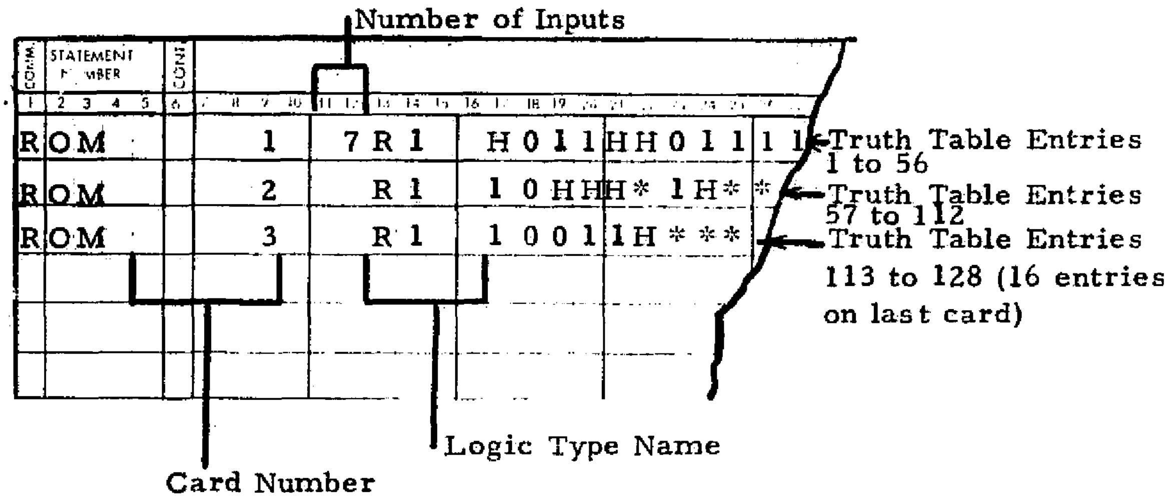
Since the sample ROM specification requires only one card, an additional example is presented illustrating a ROM specification requiring continuation cards. Figure 4-4b illustrates the specification of a seven-input ROM, having 128 truth table entries and requiring 3 cards.

4.1.7 Logic Gate Description (NET) Cards - Required

Description

NET cards are used to describe the types of elements in the circuit which are in the logic library, NEWGATE or ROM categories. The logic library and NEWGATE type NETs must appear in a group before the ROM type NETs. Each NET card must include the name of the specific element, logic type name from the library, NEWGATE, or ROM names, the initial state, if known, rise delay time index, fall

Typical ROM Specification for a Seven-Input Read-Only-Memory



```
ROM     1   7R1   H011HH0111 1
ROM     2    R1   10HHH* 1H* *
ROM     3    R1   10011H***
```

Figure 4-4b

delay time index, and a list of the names of the gates which input to that gate.

In the case of NEWGATE or ROM type gates rise and fall decay time indices may be specified if the NEWGATE, or ROM has a holding mode. If not specified, decay times are assumed infinite. For any gate, if the delay times are omitted, or given as zero, zero delay times are assumed. Several words of caution are applicable at this point.

1) It is not permissible for the user to specify a gate with zero delay times which feeds back into itself.

2) When the Preprocessor reads the delay and decay time indices it interprets 4 integers in the order: rise delay index, fall delay index, rise decay index, fall decay index.

If the user, for example, wishes to omit the fall delay index and the rise decay index, he must specify them as zeros, or the fall decay index will be interpreted as a fall delay index. The user may omit any time indices at the end of the list of 4, without causing any misinterpretation.

The list of inputs to a gate must be in descending order of significance. For example, in the clocked type gates in the library, the clock must always be listed last. In some cases the input list may require a continuation. The last input name on a card to be continued must be followed by a comma. If it is not, the program assumes it is the end of the list and will ignore a continuation card. When an automatically constructed delay gate appears in an input list it must be followed by a rise delay index and a fall delay index.

<u>Format (First Card)</u>

Columns	1 - 3	- NET - required.
	9 - 16	- gate name - required, 1 to 8 alphanumeric characters, <u>left-justified.</u>
	18 - 21	- gate type name - required, 1 to 4 alphanumeric characters, <u>left-justified</u> corresponding to a library, NEWGATE, or a ROM type name.

23	-	initial state - optional, one character from Table 4-8. If omitted, gate's initial state interpreted as indeterminate by the Preprocessor.
25 - 40	-	list of time indices - optional, in order: rise delay index, fall delay index, rise decay index, fall decay index; must begin in column 25, be separated by commas, with no internal blanks. The time indices are the same as specified on the DTMR, DTMF, and DCTM cards.
41 - 72	-	input names list - required. The names must begin in column 41, be separated by commas with no internal blanks. If a continuation of the NET is to follow, a comma should follow the last name on the list. (A name must not be split between cards.)

<u>Format (NET Continuation Cards)</u>

Any continuations must follow the card being continued.

Columns 1 - 4	-	ƀƀƀƀ (blanks) - required.
8 - 72	-	input names list in same manner as for the first card.

<u>Format (Delayed and Inverted Inputs)</u>

Automatically constructed delays and inverters are defined in the input list. If, for example, a delay of some GATE-A is to be listed as the nth input, it appears as follows in the input list.

input n-1, *GATE-A, X, Y, input n+1

where: X and Y are the rise and fall delay indices. The Preprocessor automatically constructs a delay gate and names it *GATE-A. Also, for example, some inverter GATE-B is to be listed as the nth input, and would appear in an input list as follows.

input n-1, /GATE-B, input n+1

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

CHARACTERS REQUIRED FOR LEVEL SPECIFICATION ON NET, GEN, AND GENF CARDS

NET, GEN, & GENF Level Characters	Interpretation
*	Indeterminate state
L	LOW
H	HIGH

Table 4-8

The Preprocessor constructs an inverter named /GATE-B, and assigns it a zero delay time.

Because of the manner in which delay gates and inverters are named, automatically delayed or inverted gates must have a name not exceeding 7 alphanumeric characters.

Fxample

Figure 4-3 illustrates all of the NET cards required for the sample MOS logic network. These cards were coded by referring to the circuit diagram (Figure 3-1), the element description list (Table 3-4), and the lists of delay and decay times (Table 3-5). The user will note that the ROM type element is the last NET specification. It has among its inputs a delay gate, *CP-2, with rise delay and fall delay indices of 4 and 5, and an inverter/2B.

4.1.8 Generator Description (GEN) Cards - (one GEN or GENF required)

Description

The purpose of GEN specifications is to inform the program that certain elements in the circuit are active devices and are generating a signal to the rest of the circuit. In other words GEN cards specify the input signals driving the circuit.

A GEN specification must include an initial state for the generator, the name of the generator element, and a set of generator level changes and change times. The name of a generator element follows the same rule as for other elements in the circuit. It is permissible to continue a GEN specification to more than one card, if the list of change times and level changes is lengthy. An unlimited number of generator changes may be specified.

Format (First Card)

Columns	1 - 3	- GEN - required.
	7	- one character initial state - required, from characters listed in Table 4-8.
	9 - 16	- name of generator - required. The

name is 1 to 8 alphanumeric characters, or 1 to 7 characters if output is to be automatically inverted or delayed. The name must be left-justified in the field.

18 - 72 - list of generator change times and levels - required. The change times must be whole numbers, and the levels are any of those from Table 4-8. They are listed in the following manner:

time 1, level 1, time 2, level 2, etc., separated by commas, with no internal blanks. The last level in the list must be followed by a comma if the list is to be continued on the next card. It is not permissible to split a time-level pair between two cards.

Format (Continuation Cards)

Columns	1 - 4	- ƀƀƀƀ (blanks) - required.
	9 - 72	- continuation of the time level list in same manner as for the first card.

Example

The GEN specification required for the sample MOS logic network is illustrated in Figure 4-3. Referring back to the generator timing diagram illustrated in Figure 3-2, note that the generator has an initial state of LOW. It changes to a HIGH at time = 78, to a LOW at time = 108, and so on. Hence the generator changes list is written as follows:

```
78,H,108,L,138,H,139,L
```

as shown in Figure 4-3.

4.1.9 Generator Function Description (GENF) Cards - (one GEN or GENF required)

Description

The GENF description cards pertain to those generator elements which behave in a periodic manner. The GENF specification must include the name of the generator function element, the initial state, the time at which the periodic pattern begins, the sequence of changes within one period, the time interval between each change, and the number of repetitions of the period. The sequence of changes within one period may be of any length and may require more than one card.

Format (First Card)

Columns 1 - 4	- GENF - required.
7	- one character initial state - required, from Table 4-6.
9 - 16	- generator function name - required. The name may be 1 to 8 alphanumeric characters, or 1 to 7 characters if the generator function is to be automatically delayed or inverted. The name must be left-justified within the field.
17 - 20	- number of repetitions of the period - required. It must be a whole number, of 1 to 4 digits, and must be right-justified in the field.
21 - 30	- starting time of the periodic pattern - required. It must be a whole number, of 1 to 10 digits, and must be right-justified in the field.
31 - 38	- time interval between level changes - required. It must be a whole number, of 1 to 8 digits, and must be right-justified within the field.
39 - 72	- sequence of level changes - required. Each entry in the sequence is one character from Table 4-6. All entries are listed like a truth table; i.e., side-by-side with no separating commas.

Format (Continuation Cards)

The sequence of level changes may be continued on as many additional

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

cards as necessary, in the following manner:

Columns	1 - 4	- b̸b̸b̸b̸ (blanks) - required.
	9 - 72	- continuation of the sequence of level changes in the same manner as on the first card.

Example

The sample MOS logic network has two generator function elements. The waveform of each is illustrated in Figure 3-2. Examining the first one (Figure 3-2b) the following are obvious:

initial state = HIGH

function-starting time = 25

number of repetitions = 18

However, the periodic level sequence and time interval between changes are not so obvious. The level sequence appears to be LOW-HIGH, but there are two time intervals between changes, 25 and 75. The level sequence must be adjusted so there is only one time interval. This is done as follows:

level sequence = LOW-HIGH-HIGH-HIGH

time interval = 25

Although there is no such thing as a HIGH to HIGH level change it must be expressed in that manner so that there is only one time interval. With this in mind the following information can be expressed from the waveform (Figure 3-2c) of the second generator function:

initial state = LOW

function starting time = 50

number of repetitions = 18

level sequence = HIGH-LOW-LOW-LOW

time interval = 25

Figure 4-3 illustrated how this information is expressed as GENF specifications.

4.1.10 Logical Unit Change (CHANGE TO) Cards

Description

As previously mentioned, the data input to the Preprocessor need not be limited to the card reader. The CHANGE TO card allows the data to be read from any other input device. A logical unit number is specified on the CHANGE TO card which instructs the program to read the next data record from the specified logical unit. A control card must be included in the job setup assigning the logical unit numbers to the desired devices.

Format

Columns 1 - 9	-	CHANGE TO - required.
Beginning col. 11	-	logical unit number assigned to the desired input device.

Example

Assume that a user wishes to access a magnetic tape file for all of the delay and decay time sets (DTMR, DTMF, and DCTM specifications) and read all of the other data from cards. Also assume the tape logical unit number assignment is 73 and the card reader is 105. The following illustrates how the data is read and where the CHANGE TO records must appear.

```
NAME, CONT, and SPEC cards          |
CHANGE TO 73                         }   Punched Cards
      ↖                              |
        Last card

DTMR records                         |
DTMF records                         }   Tape Unit #73
DCTM records                         |
CHANGE TO 105                        |
      ↖
        Last record on tape
```

Remaining data	} Punched Cards Unit #105

In short, the last record read before an input device change must be the CHANGE TO specification, and data input in a form other than punched cards must be of the same format as for cards.

4.2 Postprocessor Data Cards

The following paragraphs describe the Postprocessor data cards.

4.2.1 Control (CONT) Card - Optional

Description

The Postprocessor, like the Preprocessor, has a control card for the specification of options. The following postprocessing options may be chosen:

- Preprocessor Input Data List (PREPIN). This option instructs the program to read the Preprocessor input data records from the data tape and output them in a list.

- Postprocessor Input Data List (POSTIN). This option causes the program to generate a list of the postprocessing input data.

- Spike Condition Messages (SPIKE). This option instructs the program to generate a message whenever a spike condition is encountered in the events simulated.

- Output Termination for Compare Mismatch (COMSTOP). The Postprocessor performs spot checks on a gate's output in response to a compare (CMP) or compare function (CMPF) specification. In the normal mode of operation, the Postprocessor generates a descriptive message for each mismatch and continues executing. The COMSTOP option is used to force the Postprocessor to stop execution when a mismatch is encountered.

- Initial Conditions List (INITIAL). This option causes the initial conditions of all of the circuit elements to be output immediately preceding the first timing diagram. If this option is not specified, the initial conditions are output only for the gates which are included in the timing diagram.

The five postprocessing options are specified on the control card by their names: PREPIN, POSTIN, SPIKE, COMSTOP, and INITIAL. If the CONT card is omitted, the program assumes no options were desired.

Format

Columns	1 - 4	-	CONT - required.
	9 - 16 17 - 24 25 - 32 33 - 40 41 - 48	-	any of the following option names: PREPIN, POSTIN, SPIKE, COMSTOP INITIAL.

Any option may be specified in any order, but each must be left-justified in one of the five data fields specified above.

Example

The control options selected for the sample MOS logic circuit were SPIKE, PREPIN, and POSTIN. Figure 4-5 illustrates the control card.

4.2.2 Character (CRCT) Card - Optional

Description

The character card allows the user to choose the character used on the timing diagram to represent each level: indeterminate, LOW, and HIGH. If the CRCT card is omitted, the timing diagram will be output with the following characters:

*	=	indeterminate level
L	=	LOW level
H	=	HIGH level

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

DATA CARDS FOR POSTPROCESSING OF THE SAMPLE MOS LOGIC CIRCUIT

```
                                       FORTRAN STATEMENT                                        IDENTIFICATION
                                                                                                   SEQUENCE
CONT    SPIKE       PREPIN      POSTIN
CRCT    -1+
SHFT    LLR
CMP     READ              1825+
CMP     GEN-3              100+
CMPF    *CP-2          1          28          291+++
PNT     1A          2A          3A          4A          SKIP        LOCK-1-5    LOCK-1-6    LOCK-1-7
PNT     LOCK-1-8    SKIP        1B          2B          3B          4B          SKIP        LOCKUP-1
PNT     SKIP        FEED-1      FEED-2      TG          SKIP        READ        SKIP        *CP-2       *
PNT     /2B         SKIP        SKIF        GEN-3       SKIP        CP-2        CP-1
SLOT                   0        1825
PNT     GEN-3       SKIP        CP-2        SKIP        CP-1        SKIP        SKIP        READ
PNT     SKIP        LOCK-1-8                                                                            **
SLOT                  24         500         501        1000        1001        1500
SLOT                1501        1825
```

* PNT-SLOT CARD SET FOR FIRST TIMING DIAGRAM

** PNT-SLOT CARD SET FOR SECOND TIMING DIAGRAM

Figure 4-5

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

Format

Columns	1 - 4	-	CRCT - required.
	9	-	one character representing indeterminate levels.
	10	-	one character representing LOW levels.
	11	-	one character representing HIGH levels.

Example

The sample MOS logic circuit used the following characters:

- (dash) = indeterminate

| (vertical line) = LOW

+ = HIGH

Figure 4-5 illustrates the format of the CRCT card.

4.2.3 Character Shift (SHFT) Card - Optional

Description

The SHFT card allows the user to indicate the direction the characters, specified by the CRCT card, are to be shifted. The characters used to specify the shift are L for left and R for right. For positive logic the LOW character is normally shifted to the left and the HIGH character to the right, so that when the timing diagram is rotated 90° counter-clockwise, with time increasing from left to right, the HIGH levels will be above the LOW levels. The opposite can be done for negative logic. If the SHFT card is omitted, all of the characters are printed in the left of the column and the output appears as straight lines.

Format

Columns	1 - 4	-	SHFT - required.
	9	-	L or R for indeterminate character shift direction.

10	-	L or R for LOW character shift direction.
11	-	L or R for HIGH character shift direction.

Example

For the sample MOS logic network the following shift scheme was chosen:

indeterminate character (-) shift = L

LOW character (|) shift = L

HIGH character (+) shift = R

Figure 4-5 illustrates this shift specification.

4.2.4 Level Comparison (CMP) Cards - Optional

Description

The user may spot check the level of a certain gate at a given time with a CMP specification. The compare specification instructs the Postprocessor to check a particular gate for a given level at a specified time and generate a message in the case of a mismatch. A CMP card must include the name of the circuit element to be checked, the time, and the level to which its output is to be compared. The user may specify up to 100 comparisons in the CMP form.

Format

Columns	1 - 3	-	CMP - required
	9 - 16	-	name of the circuit element - required, <u>left-justified</u> in the field.
	17 - 26	-	time of comparison - required. Must be a whole number of 1 to 10 digits, <u>right-justified</u> in the field.
	27	-	level - required. The level must be one character, corresponding to the CRCT specification.

Example

The following were chosen for the sample MOS logic circuit:

- Check READ for the + level at time = 1825
- Check GEN - 3 for the + level at time = 100

Figure 4-5 illustrates these CMP specifications.

4.2.5 Compare Function (CMPF) Cards - Optional

Description

A compare function (CMPF) can be specified to make spot comparisons according to a specific pattern. The CMPF specification must include the name of the circuit element to be checked, the starting time for the comparison pattern, the time intervals between comparisons, a sequence of levels to be compared to the output of the element, and the number of repetitions of the checking sequence. The user may specify up to 100 compare functions in one postprocessing run. Like the comparison (CMP) operation, a message is printed only if there is a mismatch.

Format

Columns	1 - 4	-	CMPF - required.
	9 - 16	-	name of the circuit element - required, left-justified in the field.
	17 - 20	-	number of repetitions - required. It must be a whole number of 1 to 4 digits, right justified in the field.
	21 - 30	-	starting time - required. It must be a whole number of 1 to 10 digits, right justified in the field.
	31 - 38	-	time interval between comparisons - required. It must be a whole number of 1 to 8 digits, right-justified in the field.

39 - 70	- sequence of level characters - required. It must be a whole number of 1 to 8 digits, right-justified in the field.

Example

A CMPF was specified for the sample MOS logic circuit to monitor the output of the delay gate *CP-2 for the levels: LOW, HIGH, HIGH, and HIGH at times 28, 57, 86, and 115. Figure 4-5 illustrates this CMPF specification.

4.2.6 Output Print List (PNT) Cards - Required

Description

The PNT cards define the format of the timing diagram by specifying the circuit elements to include in the diagram, and the order in which they are to appear. A timing diagram can accommodate up to 40 columns of output with each column corresponding directly to a circuit element specified on a PNT card. Since a single PNT card can specify 8 circuit elements, a maximum of 5 PNT cards may be required to complete the definition of a timing diagram.

To facilitate the interpretation of a timing diagram and to emphasize particular elements within the diagram, a circuit element or group of elements can be isolated from other timing diagram outputs by specifying blank columns within the diagram. A blank column can be generated by specifying the word SKIP on the PNT card in place of a circuit element name.

An unlimited number of timing diagrams can be defined with PNT cards, but each set of PNT cards defining a timing diagram must be followed by a set of SLOT cards of the format described in Section 4.2.7.

Format

Columns	1 - 3	- PNT - required.
	9 - 16 17 - 24 25 - 32 33 - 40 41 - 48 49 - 56 57 - 64 65 - 72	- element names or SKIP for blank column. The name or SKIP must be left-justified in the fields. Blank fields in the list are not permissible.

Example

For the sample MOS logic network there are two sets of PNT and SLOT cards, as illustrated in Figure 4-5. Each set corresponds to a separate timing diagram.

The first set of PNT cards defines a timing diagram containing every element in the circuit, starting with gate 1A in the first column and ending with CP-1 in the last column. The SKIP notation generates blank columns in positions 5, 10, 15, 17, 21, 23, 26, 27, and 29.

The second PNT set defines a timing diagram for only the generators, generator functions and the unloaded gates, READ and LOCK-1-8 (see Figure 3-1). Blanks are forced in columns 2, 4, 6, 7, and 9.

4.2.7 Output Time Slot (SLOT) Cards - Required

Description

The SLOT cards define the time periods for which the timing diagram is to be printed. A set of SLOT cards is required for each timing diagram and must immediately follow the PNT cards. Each SLOT card may specify up to three time periods with each time period defined by a start time value and an end time value. From 1 to 50 time slots may be specified for each timing diagram on as many SLOT cards as necessary.

Format

Columns	1 - 4	- SLOT - required.
	11 - 20	- start time of first slot, right-justified.
	21 - 30	- end time of first slot, right-justified.
	31 - 40	- start time of second slot, right-justified.
	41 - 50	- end time of second slot, right-justified.
	51 - 60	- start time of third slot, right-justified.
	61 - 70	- end time of third slot, right-justified.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

Example

For the sample MOS logic network, the SLOT specification for the first timing diagram lists only one time slot, from 0 to 1825 (the maximum time). The SLOT specification for the second timing diagram indicates the diagram is to be output for four time slots: 24 to 500, 501 to 1000, 1001 to 1500, and 1501 to 1825.

5. LOGSIM OUTPUT INTERPRETATION

This section discusses and illustrates the printed output supplied by the Preprocessor and Postprocessor portions of the LOGSIM Program. The output of the Simulator is not presented to the user for review but is passed directly to the Postprocessor by way of magnetic tape or disk. The sample MOS logic circuit previously presented is used to illustrate each type of LOGSIM printed output.

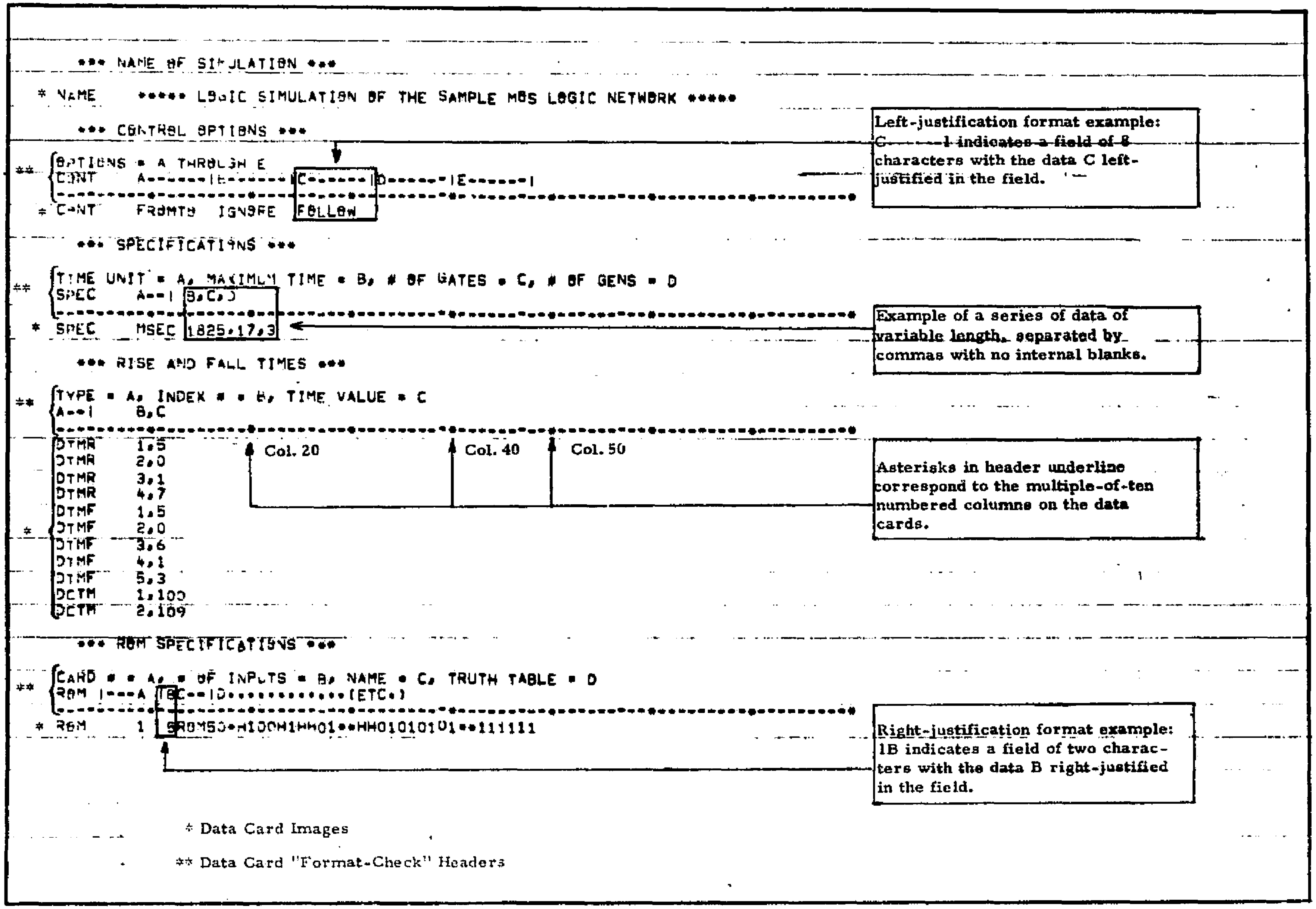
5.1 Preprocessor Output

The Preprocessor printed output serves as a permanent description of the circuit simulated. The output consists of the following five major sections:

- Input Data Page - Each input data record is printed accompanied by an appropriate "format-check" header. Figures 5-1, 5-2, and 5-3 illustrate the data card printout for preprocessing the sample MOS logic circuit.
- Input Diagnostic Page - If input data errors are detected, the errors are listed and the program terminates. Figure 5-4 illustrates the format of the diagnostic page for a circuit with no input data errors.
- Options and Specifications Page - An options and specifications list is compiled and output to enable the user to determine if the proper circuit elements have been preprocessed. Figure 5-4 illustrates the options and specifications list compiled for the sample MOS circuit.
- Delay Gates and Inverters Page - Delay gates and inverters automatically constructed by the Preprocessor are listed. Figure 5-5 illustrates the printed output generated for the delay gate and inverter constructed for the sample MOS logic circuit.
- Connectivity List Page - The Preprocessor generates, as a user option, a connectivity list showing each circuit element and all the circuit elements loading it. The connectivity list for the sample circuit is illustrated in Figure 5-6.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

PREPROCESSOR PRINTOUT : INPUT DATA (SAMPLE MOS LOGIC NETWORK)



```
       *** NAME OF SIMULATION ***

  * NAME     ***** LOGIC SIMULATION OF THE SAMPLE MOS LOGIC NETWORK *****

       *** CONTROL OPTIONS ***

 ** {OPTIONS = A THROUGH E
    {CONT     A-------|B-------|C-------|D-------|E-------|
    {-------------*---------*---------*---------*---------*---------*---------*
  * CONT     FROMTO  IGNORE  FOLLOW

       *** SPECIFICATIONS ***

 ** {TIME UNIT = A, MAXIMUM TIME = B, # OF GATES = C, # OF GENS = D
    {SPEC     A--| B,C,D
    {-------------*---------*---------*---------*---------*---------*---------*
  * SPEC     MSEC 1825,17,3

       *** RISE AND FALL TIMES ***

 ** {TYPE = A, INDEX # = B, TIME VALUE = C
    {A--|     B,C
    {-------------*---------*---------*---------*---------*---------*---------*
    {DTMR     1,5
    {DTMR     2,0
    {DTMR     3,1
    {DTMR     4,7
    {DTMF     1,5
  * {DTMF     2,0
    {DTMF     3,6
    {DTMF     4,1
    {DTMF     5,3
    {DCTM     1,100
    {DCTM     2,109

       *** ROM SPECIFICATIONS ***

 ** {CARD # = A, # OF INPUTS = B, NAME = C, TRUTH TABLE = D
    {ROM  |---A |B|C--|D............(ETC.)
    {-------------*---------*---------*---------*---------*---------*---------*
  * ROM      1 |5|ROM50*H100H1HH01**HH01010101**111111
```

Left-justification format example: ~~C-------| indicates a field of 8~~ characters with the data C left-justified in the field.

Example of a series of data of variable length, separated by commas with no internal blanks.

Col. 20 Col. 40 Col. 50

Asterisks in header underline correspond to the multiple-of-ten numbered columns on the data cards.

Right-justification format example: 1B indicates a field of two characters with the data B right-justified in the field.

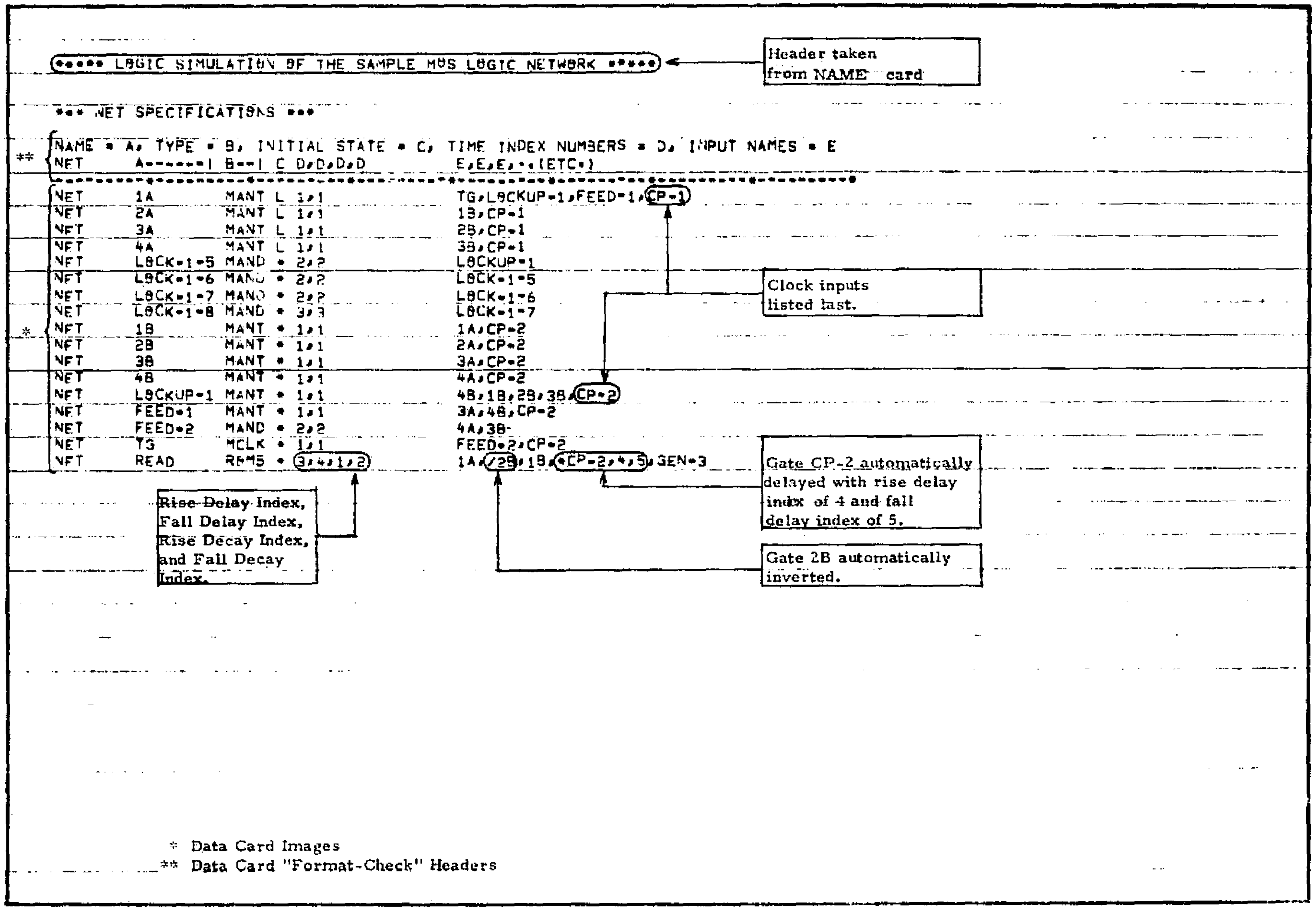
* Data Card Images

** Data Card "Format-Check" Headers

Figure 5-1

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

PREPROCESSOR PRINTOUT : NET INPUT DATA (SAMPLE MOS LOGIC NETWORK)



```
 ***** LØGIC SIMULATIØN ØF THE SAMPLE MØS LØGIC NETWØRK *****

 *** NET SPECIFICATIØNS ***

   NAME = A, TYPE = B, INITIAL STATE = C, TIME INDEX NUMBERS = D, INPUT NAMES = E
** NET      A-------| B--| C D,D,D,D            E,E,E,..(ETC.)
   ---------+---------+-------------+-----------+----------+-------------+----------
   NET      1A        MANT L 1,1                TG,LØCKUP-1,FEED-1,CP-1
   NET      2A        MANT L 1,1                1B,CP-1
   NET      3A        MANT L 1,1                2B,CP-1
   NET      4A        MANT L 1,1                3B,CP-1
   NET      LØCK-1-5  MAND * 2,2                LØCKUP-1
   NET      LØCK-1-6  MAND * 2,2                LØCK-1-5
   NET      LØCK-1-7  MAND * 2,2                LØCK-1-6
   NET      LØCK-1-8  MAND * 3,3                LØCK-1-7
*  NET      1B        MANT * 1,1                1A,CP-2
   NET      2B        MANT * 1,1                2A,CP-2
   NET      3B        MANT * 1,1                3A,CP-2
   NET      4B        MANT * 1,1                4A,CP-2
   NET      LØCKUP-1  MANT * 1,1                4B,1B,2B,3B,CP-2
   NET      FEED-1    MANT * 1,1                3A,4B,CP-2
   NET      FEED-2    MAND * 2,2                4A,3B-
   NET      TG        MCLK * 1,1                FEED-2,CP-2
   NET      READ      RFM5 * 3,4,1,2            1A,/2B,1B,*CP-2,4,5,GEN-3
```

Header taken from NAME card

Clock inputs listed last.

Gate CP-2 automatically delayed with rise delay indx of 4 and fall delay index of 5.

Gate 2B automatically inverted.

Rise Delay Index, Fall Delay Index, Rise Decay Index, and Fall Decay Index.

* Data Card Images
** Data Card "Format-Check" Headers

Figure 5-2

PREPROCESSOR PRINTOUT : GEN AND GENF INPUT DATA (SAMPLE MOS LOGIC NETWORK)

```
    ***** LOGIC SIMULATION OF THE SAMPLE MOS LOGIC NETWORK *****

      *** GENERATOR SPECIFICATIONS ***

    {INITIAL STATE = A, NAME = B, TIMES = C, LEVELS = D
 ** {GFN    A B------( C,D,C,D,C,D......(ETC.)
    {*-----------------------------------------------------------------------*
 *   GFN    L GEN-3     78,H,108,L,138,H,139,L
```

```
    ***** LOGIC SIMULATION OF THE SAMPLE MOS LOGIC NETWORK *****

      *** GENERATOR FUNCTION SPECIFICATIONS ***

    {INITIAL STATE = A, NAME = B, # OF REPITITIONS = C, 1ST TIME = D, INTERVAL = E, LEVEL SEQUENCE = F
 ** {GFNF   A B-------||--C|---------D|-------E|......(ETC.)
    {*-----------------------------------------------------------------------*
 *   GENF   H CP=2       18          25       25LLLH
 *   GFNF   L CP=1       18          50       25HLLL
```

* Data Card Images
** Data Card "Format-Check" Header

Figure 5-3

PREPROCESSOR PRINTOUT : INPUT DATA DIAGNOSTICS PAGE AND OPTIONS AND SPECIFICATIONS LIST
(SAMPLE MOS LOGIC NETWORK)

```
***** LOGIC SIMULATION OF THE SAMPLE MOS LOGIC NETWORK *****

  *** ERROR CHECK ***

NO ERRORS WERE FOUND. PROGRAM WILL CONTINUE.
```

```
***** LOGIC SIMULATION OF THE SAMPLE MOS LOGIC NETWORK *****

  *** OPTIONS AND SPECIFICATIONS ***

INDETERMINATE INITIAL CONDITIONS WILL BE IGNORED. <----------- IGNORE Option specified
GATE LOADS CONNECTIVITY LIST WILL BE PRINTED. <--------------- NONCON Option specified
AN ATTEMPT WILL BE MADE TO ESTABLISH VALUES FOR INDETERMINATE ROM INPUTS. <-- FOLLOW Option specified

MAXIMUM SIMULATION TIME = 1825
TIME UNIT = MSEC
17  LOGIC GATES WERE SPECIFIED, 17  WERE READ.
3  GENS AND GENFS WERE SPECIFIED, 3  WERE READ.
1  DELAY GATES WERE CONSTRUCTED.
1  INVERTERS WERE CONSTRUCTED.
TOTAL ELEMENTS IN THE CIRCUIT = 22
```

Figure 5-4

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

PREPROCESSOR PRINTOUT : LISTS OF DELAY GATES AND INVERTERS AUTOMATICALLY CONSTRUCTED
(SAMPLE MOS LOGIC NETWORK)

```
***** LOGIC SIMULATION OF THE SAMPLE MOS LOGIC NETWORK *****

  *** DELAY GATES CONSTRUCTED ***

   NAME       INPUT     INITIAL STATE *
 -----------------------------------------
  *CP=2       CP=2            1

  *** INVERTERS CONSTRUCTED ***

   NAME       INPUT     INITIAL STATE  *
 -----------------------------------------
  /2B         2B              *
```

* INITIAL STATE REPRESENTATION

* = indeterminate

0 = LOW

1 = HIGH

Figure 5-5

PREPROCESSOR PRINTOUT : CONNECTIVITY LIST (SAMPLE MOS LOGIC NETWORK)

```
***** LØGIC SIMULATIØN ØF THE SAMPLE MØS LØGIC NETWØRK

  *** GATE/LØADS CØNNECTIVITY LIST ***

 GATE          LØADS
---------------------
1A            1B
              READ

2A            2B

3A            3B
              FEED-1

4A            4B
              FEED-2

LØCK-1-5      LØCK-1-6

LØCK-1-6      LØCK-1-7

LØCK-1-7      LØCK-1-8

LØCK-1-8       (NØT LØADED)*

1B            2A
              LØCKUP-1
              READ

2B            3A
              LØCKUP-1
              /2B

3B            4A
              LØCKUP-1
              FEED-2

4B            LØCKUP-1
              FEED-1

LØCKUP-1      1A
              LØCK-1-5

FEED-1        1A

FEED-2        TG

TG            1A
```

```
***** LØGIC SIMULATIØN ØF THE SAMPLE MØS LØGIC NETWØRK

  *** GATE/LØADS CØNNECTIVITY LIST ***

 GATE          LØADS
---------------------
READ           (NØT LØADED)*

*CP-2         READ

/2B           READ

GEN-3         READ

CP-2          1B
              2B
              3B
              4B
              LØCKUP-1
              FEED-1
              TG
              *CP-2

CP-1          1A
              2A
              3A
              4A
```

* Network elements with no loads

(network output nodes).

Figure 5-6

5.2 Postprocessor Output

The Postprocessor printed output describes the input data and setup of the program, and illustrates the results of the logic simulation. The output consists of the following six major sections:

- Postprocessor Input Data Page - The Postprocessor outputs images of the input data cards, along with any diagnostic messages pertaining to the data. Figure 5-7 illustrates input data listing for the sample MOS logic circuit. The generation of the input data page output is a user option controllable by the POSTIN option on the Postprocessor control (CONT) card.

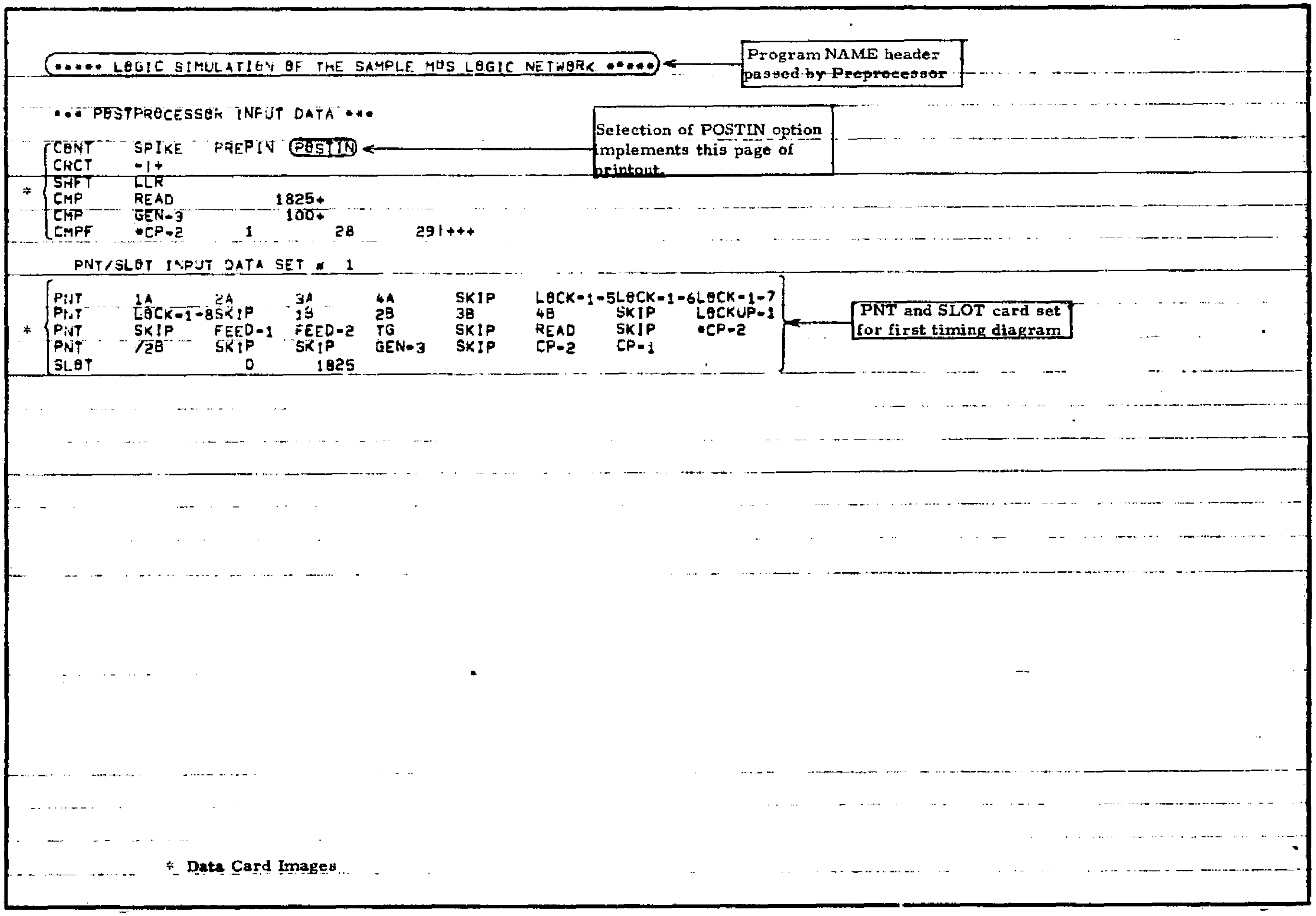
- Preprocessor Input Data Page - Images of the Preprocessor data cards are passed to the Postprocessor and listed by the Postprocessor if the user specifies the PREPIN option on the control (CONT) card. This printed output allows the user to generate a complete reference to the simulation run entirely within the Postprocessor output listing. Figure 5-8 illustrates the Preprocessor input data page for the sample MOS logic circuit.

- Options List Page - The Postprocessor generates an options list identifying the options specified on the Postprocessor control (CONT) card. Figure 5-9 illustrates the options list page for the sample MOS logic circuit.

- Level Comparison and Compare Function List Page - The Postprocessor generates a printed output describing any level comparisons (CMP specifications) and any compare functions (CMPF specifications). Figure 5-10 illustrates the CMP and CMPF specifications for the sample MOS logic circuit.

- Print Matrix Page - A print matrix page appears before each timing diagram, listing each element name on the timing diagram and the time slots which the timing diagram encompasses. Since a timing diagram may consist of up to 40 output columns, there is not a sufficient amount of space to print each element name above its respective output column. Therefore, each column is numbered and related to a particular element name by the print matrix. The print matrix for the first timing diagram of the sample MOS logic circuit is illustrated in Figure 5-11.

POSTPROCESSOR PRINTOUT : INPUT DATA (SAMPLE MOS LOGIC NETWORK)



```
 ***** LOGIC SIMULATION OF THE SAMPLE MOS LOGIC NETWORK *****

 *** POSTPROCESSOR INPUT DATA ***

  CONT     SPIKE   PREPIN  POSTIN
  CRCT     -|+
* SHFT     LLR
  CMP      READ            1825+
  CMP      GEN-3            100+
  CMPF     *CP-2     1            28        29|+++

   PNT/SLOT INPUT DATA SET #  1

  PNT      1A       2A       3A       4A       SKIP     LOCK-1-5LOCK-1-6LOCK-1-7
  PNT      LOCK-1-8SKIP      1B       2B       3B       4B       SKIP     LOCKUP-1
* PNT      SKIP     FEED-1   FEED-2   TG       SKIP     READ     SKIP     *CP-2
  PNT      /2B      SKIP     SKIP     GEN-3    SKIP     CP-2     CP-1
  SLOT                       0        1825
```

Program NAME header passed by Preprocessor

Selection of POSTIN option implements this page of printout.

PNT and SLOT card set for first timing diagram

* Data Card Images

Figure 5-7

POSTPROCESSOR PRINTOUT : PREPROCESSING INPUT DATA (SAMPLE MOS LOGIC NETWORK)

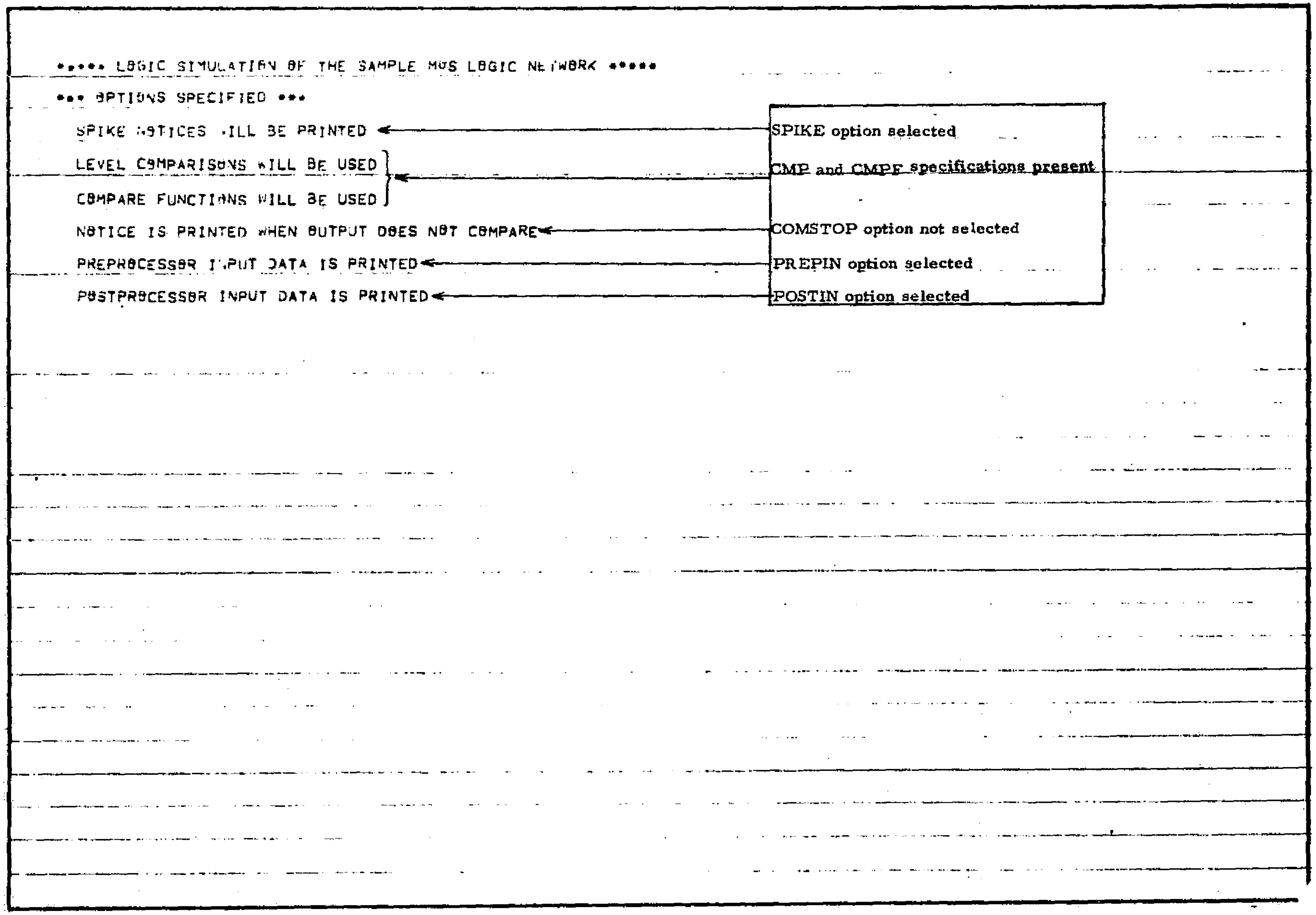
```
***** LOGIC SIMULATION OF THE SAMPLE MOS LOGIC NETWORK *****

*** PREPROCESSOR INPUT DATA ***

CONT    FROMTO  IGNORE  FOLLOW
SPEC    MSEC 1825,17,3
DTMR    1,5
DTMR    2,0
DTMR    3,1
DTMR    4,7
DTMF    1,5
DTMF    2,0
DTMF    3,6
DTMF    4,1
DTMF    5,3
DCTM    1,100
DCTM    2,109
ROM     1  5RAMS *H100H1HH01**HH01010101**111111
NET     1A         MANT L 1,1            TG,LOCKUP=1,FEED=1,CP=1
NET     2A         MANT L 1,1            1B,CP=1
NET     3A         MANT L 1,1            2B,CP=1
NET     4A         MANT L 1,1            3B,CP=1
NET     LOCK=1=5   MAND * 2,2            LOCKUP=1
NET     LOCK=1=6   MAND * 2,2            LOCK=1=5
NET     LOCK=1=7   MAND * 2,2            LOCK=1=6
NET     LOCK=1=8   MAND * 3,3            LOCK=1=7
NET     1B         MANT * 1,1            1A,CP=2
NET     2B         MANT * 1,1            2A,CP=2
NET     3B         MANT * 1,1            3A,CP=2
NET     4B         MANT * 1,1            4A,CP=2
NET     LOCKUP=1   MANT * 1,1            4B,1B,2B,3B,CP=2
NET     FEED=1     MANT * 1,1            3A,4B,CP=2
NET     FEED=2     MAND * 2,2            4A,3B
NET     TG         MCLK * 1,1            FEED=2,CP=2
NET     READ       RAMS * 3,4,1,2        1A,/2B,1B,*CP=2,4,5,GEN=3
GEN   L GEN=3      78,F,103,L,138,H,139,L
GENF  H CP=2        18        25      25LLLH
GENF  L CP=1        18        50      25HLLL
```

Preprocessor data record images printout implemented by selection of Postprocessor PREPIN program control option.

Figure 5-8

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

POSTPROCESSING PRINTOUT : OPTIONS LIST (SAMPLE MOS LOGIC NETWORK)



```
***** LOGIC SIMULATION OF THE SAMPLE MOS LOGIC NETWORK *****
*** OPTIONS SPECIFIED ***
   SPIKE NOTICES WILL BE PRINTED
   LEVEL COMPARISONS WILL BE USED
   COMPARE FUNCTIONS WILL BE USED
   NOTICE IS PRINTED WHEN OUTPUT DOES NOT COMPARE
   PREPROCESSOR INPUT DATA IS PRINTED
   POSTPROCESSOR INPUT DATA IS PRINTED
```

SPIKE option selected
CMP and CMPF specifications present
COMSTOP option not selected
PREPIN option selected
POSTIN option selected

Figure 5-9

POSTPROCESSOR PRINTOUT : LEVEL COMPARISON AND COMPARE FUNCTIONS LIST
(SAMPLE MOS LOGIC NETWORK)

```
***** LØGIC SIMULATIØN ØF THE SAMPLE MØS LØGIC NETWØRK *****

*** CØMPARE LIST ***

GATE NAME  CØMPARE TIME  LEVEL
******************************

READ              1825       +
GEN*3              100       +
```

```
***** LØGIC SIMULATIØN ØF THE SAMPLE MØS LØGIC NETWØRK *****

*** CØMPARE FUNCTIØN LIST ***

GATE NAME  REPITITIØNS  FIRST TIME  INTERVALS           LEVEL SEQUENCE
**********************************************************************

*CP*2             1             28         29     |+++
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

Figure 5-10

POSTPROCESSOR PRINTOUT : PRINT MATRIX FOR FIRST TIMING DIAGRAM (SAMPLE MOS LOGIC NETWORK)

```
***** LOGIC SIMULATION OF THE SAMPLE MOS LOGIC NETWORK *****

*** PRINT MATRIX ***

GATE NAME    NUMBER
-----------------

1A              1
2A              2
3A              3
4A              4
 SKIP
LOCK-1-5        5
LOCK-1-6        6
LOCK-1-7        7
LOCK-1-8        8
 SKIP
1B              9
2B             10
3B             11
4B             12
 SKIP
LOCKUP-1       13
 SKIP
FEED-1         14
FEED-2         15
TG             16
 SKIP
READ           17
 SKIP
*CP-2          18
72B            19
 SKIP
 SKIP
GEN-3          20
 SKIP
CP-2           21
CP-1           22

TIME SLOTS TO BE PRINTED

START TIME    END TIME
---------------------

        0         1825
```

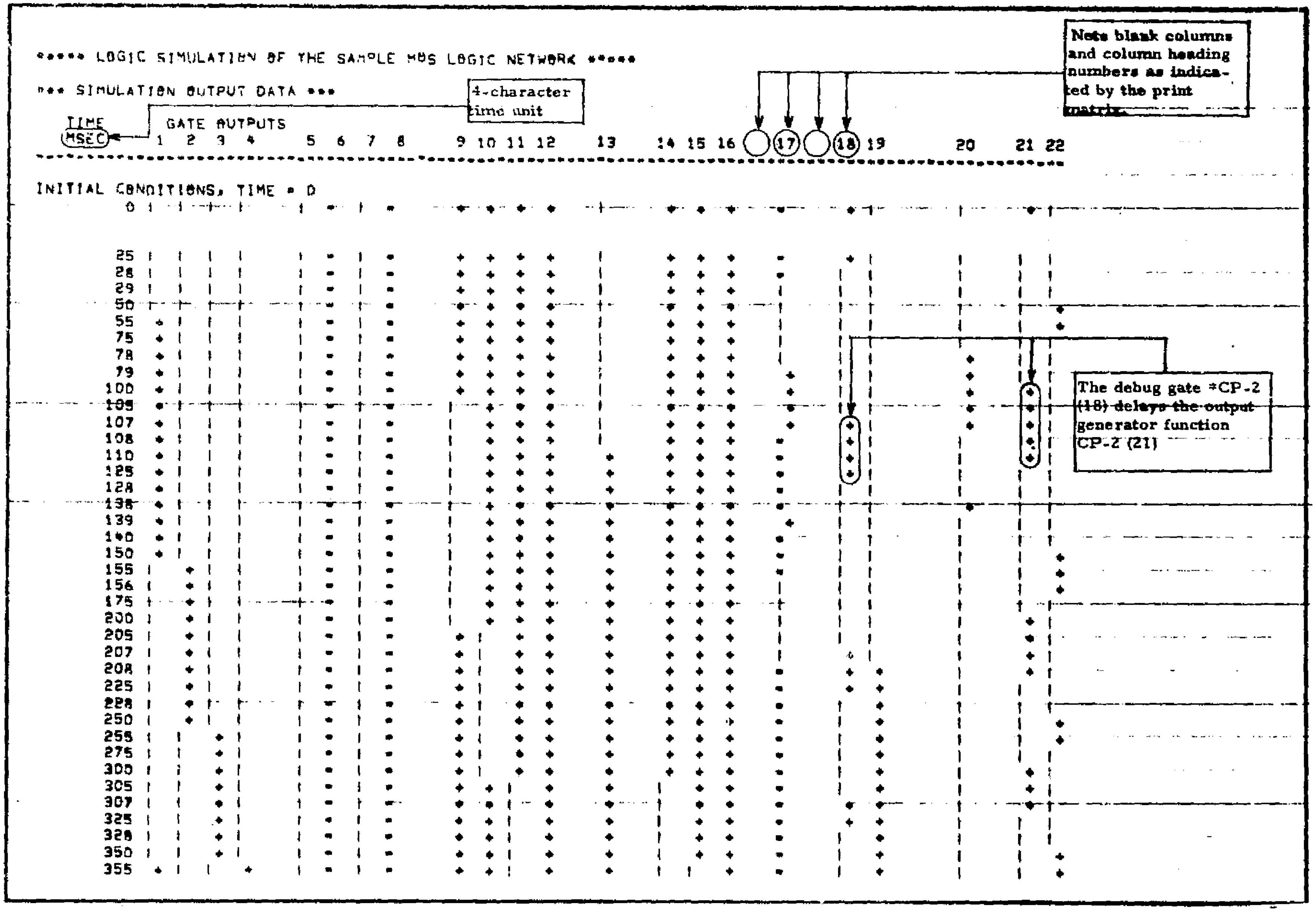
The print matrix numbers refer to the column heading numbers representing each respective element's output on the timing diagram beginning on the following page.

SKIP s appear wherever blank columns have been specified on the PNT cards. Note that the blank columns are not numbered.

Figure 5-11

- Timing Diagram Page - The final output of the Postprocessor is a series of timing diagrams, each preceded by a print matrix. The timing diagrams illustrate the results of the logic simulation. Figure 5-12 presents the first page of the first timing diagram for the sample MOS logic circuit. The characters -, 1, and + denote indeterminate, low, and high respectively.

POSTPROCESSING PRINTOUT : FIRST PAGE OF FIRST TIMING DIAGRAM (SAMPLE MOS LOGIC NETWORK)



```
***** LOGIC SIMULATION OF THE SAMPLE MOS LOGIC NETWORK *****

*** SIMULATION OUTPUT DATA ***

 TIME      GATE OUTPUTS
 MSEC    1  2  3  4     5  6  7  8     9 10 11 12    13    14 15 16    17    18 19       20     21 22
-----------------------------------------------------------------------------------------------------

INITIAL CONDITIONS, TIME = 0
       0
      25
      28
      29
      50
      55
      75
      78
      79
     100
     105
     107
     108
     110
     125
     128
     138
     139
     140
     150
     155
     156
     175
     200
     205
     207
     208
     225
     228
     250
     255
     275
     300
     305
     307
     325
     328
     350
     355
```

Figure 5-12