SLIC - SYMBOLIC LAYOUT OF INTEGRATED CIRCUITS

By Dave Gibson and Scott Nance

American Microsystems Inc.
3800 Homestead Road
Santa Clara, California 95051

SUMMARY

The purpose of this paper is to introduce a symbolic layout technique for MOS integrated circuits. We will give a description of symbolic layout, talk about its potential and briefly describe the symbolic layout system we have developed at AMI.

INTRODUCTION

Symbolic layout is a mask design method that uses symbols to represent mask topologies. The symbols are converted to the mask topologies by a computer program. For example, consider a simple 4 mask MOS process where the first mask is diffusion, second mask is gate oxide, third mask is contact cut out and fourth is metal. Figure 1 shows a set of symbols that could be used to draw simple circuits in this process. Using the symbol set in Figure 1 is quite simple. (In actual practice a more complex symbol set is used.) A grid size is chosen and the circuit layout is drawn on grided paper or mylar following two simple rules:

1) If symbols are on adjacent grids, then the topologies they ıepresent are assumed to be connected.

2) If symbols are on non-adjacent grids, they are not connected and their separation meets or exceeds minimum mask layout rules.

Figures 2 and 3 show a simple logic structure and its symbolic and topological layout. The examples illustrate how a freehand symbolic layout can be created in a much shorter time and at a much smaller scale than the same circuit laid out by drawing all topologies in detail.

DISADVANTAGES VS. ADVANTAGES

Symbolic layout techniques [1,2,3] have been around for several years, but for various reasons these techniques have not gained general acceptance in the semiconductor industry. Symbolic layouts are done on a much coarser grid (e.g., 0.5 mils) than standard topological layouts (e.g., 0.1 or 0.05 mils). Therefore, it seems that the "hand draw" layout will be smaller than the symbolic layout of the same circuitry. For example, when drawing a PMOS circuit using symbols on a 0.5 mil grid, it is not possible to take advantage of the minimum line widths and spacings available, whereas the "hand draw" layout can take advantage of these minimum geometries.

If a section of circuitry which was previously "hand drawn" is redrawn with symbols, the area used for the symbolic layout can be as much as 30% greater than the area required for the "hand draw" layout. This happens because the symbolic layout is forced to use the same grouping and positioning of transistors as the "hand draw". We have found that experienced layout designers find ways of making the symbolic layout spacings work to their advantage. A simple example of this is shown in Figure 3 where several of the transistors were placed in the signal lines 'A' and 'B'. Since the circuit is on a much coarser grid, the layout designer is able to see the entire circuit and get a feel for how the total circuit is developing and make appropriate changes. The layout designers can "play" with and redraw a given area several times, improving the packing density each time and still take less time than it would have taken to draw the same area once by the "hand draw" technique. With experienced designers we have found that the actual size increases for symbolic layouts over "hand draw" range from 0 to 10% depending on the type of circuit.

Further studies have indicated that by tailoring a few layout rules and going to a smaller grid, i.e., .25 mil, the disadvantages listed above disappear and the advantages still remain. Drawing times increase by 10% to 20% and the visability of the circuit still remains.

The above holds true for retrofitting a symbolic layout system to an established process. When a new process is proposed, if the layout rules are tailored to meet the requirements of the symbolic layout system then we can have the best of both systems. One such new process that meets these requirements at AMI is VMOS[4].

THE SLIC SYSTEM

A circuit is drawn from a logic diagram with the signal names (node numbers) indicated on the symbol topology. At the same time a logic deck is created from the same logic diagram. The logic deck is similar to a logic simulation deck except that the logic deck includes pads, I/O devices and buffers, and devices that may have electrical characteristics but not logical characteristics. The circuit is digitized, then the topology and node numbers are separated into two different files. The SLIC programs are run on this data (symbol topology, node files, logic deck) until all errors have been found and corrected. Then the symbol data is converted to mask topologies and sent to the pattern generator to generate the actual masks.

After a correction or modification to the circuit is made, the direct feedback achieved by the users when running the various automated programs can be rewarding and/or instructive especially when compared to hand drawn circuits where the feedback loop could be several weeks or longer (after the circuit fails at silicon).

EXPLANATION OF SLIC PROGRAMS

Since the symbols are very simple, this allows very sophisticated C.A.D. programs to be used which do not have to deal directly with the topological complexities until the actual masks are needed. A printer is used to print out the layouts instead of a costly plotter. Design rule checking is reduced to checking symbol-to-symbol adjacency and location. Continuity and logic verification, resistance and capacitance checking are reduced to tracing of pre-defined gates, capacitors and interconnect. Error correction, circuit modification and area relocation can be accomplished in minutes at a remote terminal connected to a central computer.

Figure 4 shows the flow of a circuit through the SLIC system. Figure 5 shows an overview of the data flow through the SLIC system. The programs are written in ALGOL for a Burroughs 6700 computer. An explanation of the seven SLIC programs follows (run times are for a 160x160 mil PMOS chip with 3100 transistors drawn on a .45 mil grid).

CVINPUT: Translates digitized data to the SLIC data base. run time = 1 minute

EDIT: Interactive program that does all data manipulation and updates. It allows for circuit rotation, area shifts, cell insertion, data conversion, separate symbol printer plots, multi-color printer plots and more. run time = 5 seconds to 1 minute

SDRC: Does design rule checking. Checks for all symbol-to-symbol layout rule violations. Symbols may be as many grids apart as is necessary to check the rules. Can be used to correct and/or change symbols. Rules are written in Boolean equation format. For example, the rule to check for corner metal-to-metal violations would be:

? [0,0] = metal [0,0] & (¬(metal[0,1] + metal [1,0]) & metal [1,1]+ etc.)

This rule replaces any metal symbols that are in violation with a question mark (calling further attention to the violation) and prints out a message that a question mark has been placed at location x,y. run time = 30 minutes

COMPARE: Compares the names in the logic deck against the names in the node files and prints out the differences. This guarantees the names the Continuity program checks are consistent. run time = 20 seconds

TRACE: Traces the symbolic layout and produces a net list which includes device sizes. Capacitance and resistance values for each circuit node are calculated and printed out. Shorts between named signal paths are indicated as well as breaks between signal paths with the same name (if the path has been labeled more than once). Finds gates that have multiple source - drain inputs and prints out their location. run time = 3 minutes

CONTINUITY: Generates a net list from a logic deck and compares it to the net list generated by the TRACE program. Compares device sizes. Prints out any differences it finds and arranges print out according to signal names so as to make error correction easier. run time = 3 minutes

S2R: Converts symbol to appropriate topological configuration for each mask level. run time = 10 minutes

All the programs are process independent. A set of "library" files, which may be updated or created by the user, is kept for each process. The "library" files contain all the layout rules, mask connection and expansion characteristics for the symbols. Thus, new symbolic layout schemes may be created and explored or old ones changed without program modification.

CONCLUSION

We have found that SLIC (1) reduces drawing time, (2) reduces digitizing (data capture) time, (3) maintains circuit sizes within 10% of "hand draw", (4) allows for a faster and more accurate checking function, (5) reduces costs. The time from drawing to mask generation can be cut approximately in half with half the man effort and half the cost. The resultant circuits have no design or logic errors and require virtually no turnarounds. Tailoring layout rules and grid sizes can reduce area loss even more.

We have indicated the use of SLIC for MOS processes, however, we believe that SLIC is a general layout tool and could be used for multi-layer, P.C., bi-polar or whatever. The programs have no theoretical limit for number of masks or size of grids.

ACKNOWLEDGEMENT

The authors would like to acknowledge Sharon Hovey and Andy Heninger for their efforts in the development of the SLIC system at AMI.

REFERENCES

1. R.P. Larsen, "Computer-Aided Preliminary Layout Design of Customized MOS Arrays", IEEE Transactions on Computers, Vol. C-20, No. 5, May 1971, pp 512-523.

2. R.P. Larsen, "Interactive Computer-Aided Editing of Custom MOS Device Layout Design", Proceedings of International Microelectronic Symposium, Oct. 1973, San Francisco, Ca.

3. D.E. Barnes and T.L. Davis, "Graphic Layout Systems for Faster Tooling from Faster Designs", Proceedings of Session 4, WESCON Professional Program, Sept. 1975, San Francisco, Ca.

4. T.J. Rogers, W.R. Hiltpold, J. Zimmer, G. Marr, J.D. Trotter, "VMOS ROM", ISSCC Digest of Technical Papers, WPM 6-4, pp 60-61, Feb. 1976.

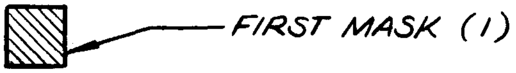
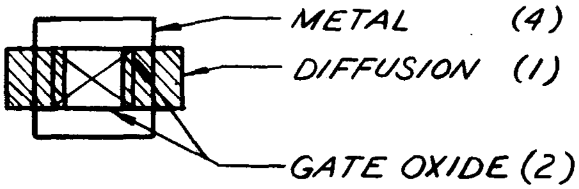
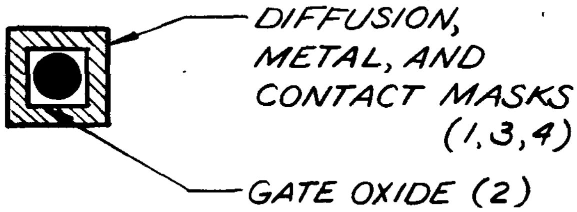
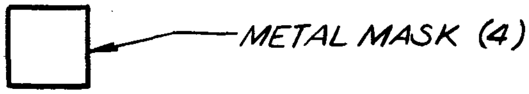
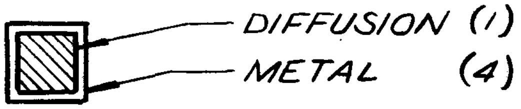
SYMBOL	DEFINITION (DIMENSIONS IN MILS)	GEOMETRY REPRESENTED BY SYMBOL
—	DIFFUSION (0.3 x 0.3)	
×	TRANSISTOR (0.3 x 0.3)	
○	CONTACT (0.3 x 0.3)	 DIFFUSION, METAL, AND CONTACT MASKS (1,3,4) GATE OXIDE (2)
1	METAL (0.4 x 0.4)	
+	METAL, DIFFUSION CROSSOVER	

FIGURE 1.

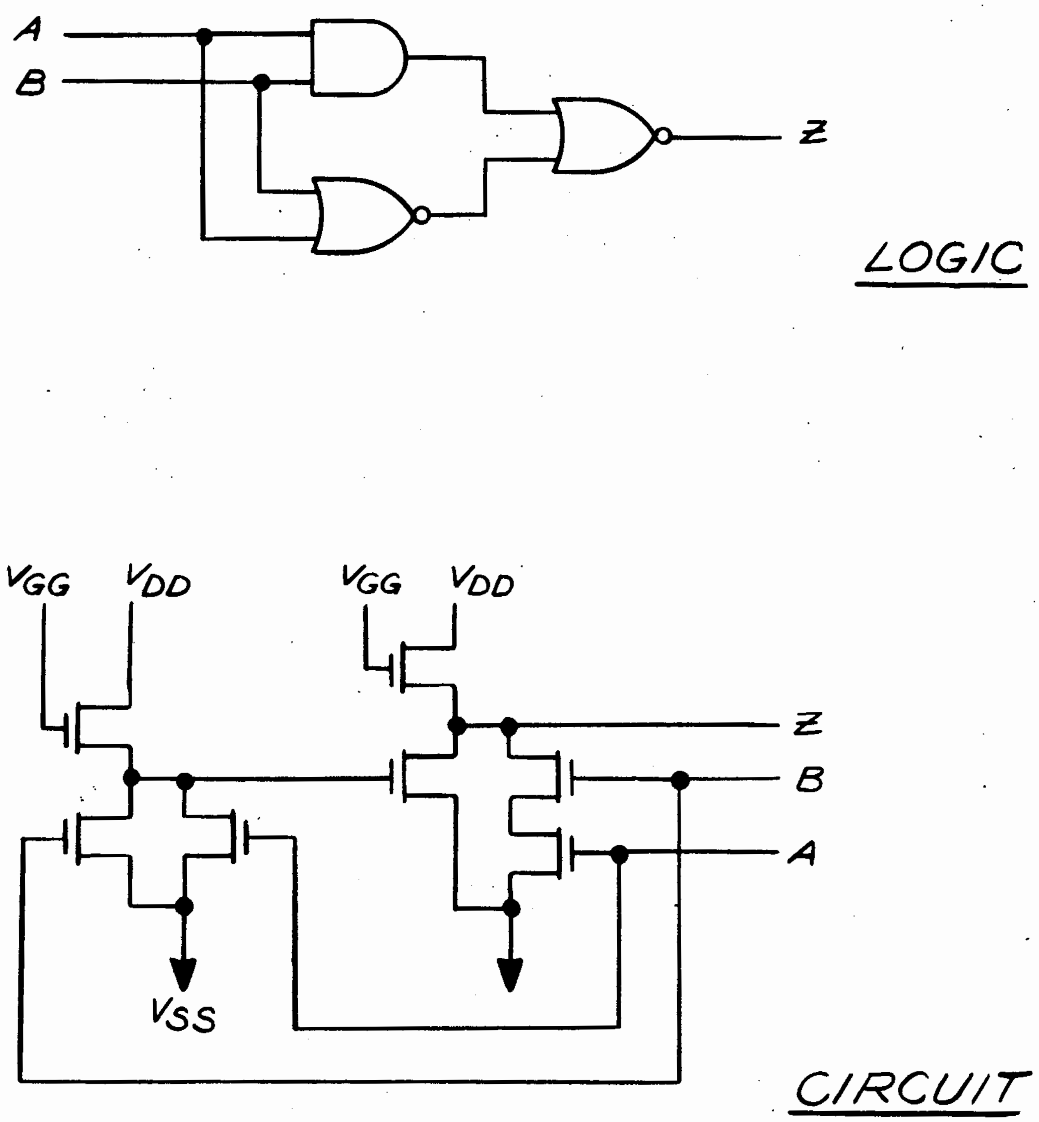
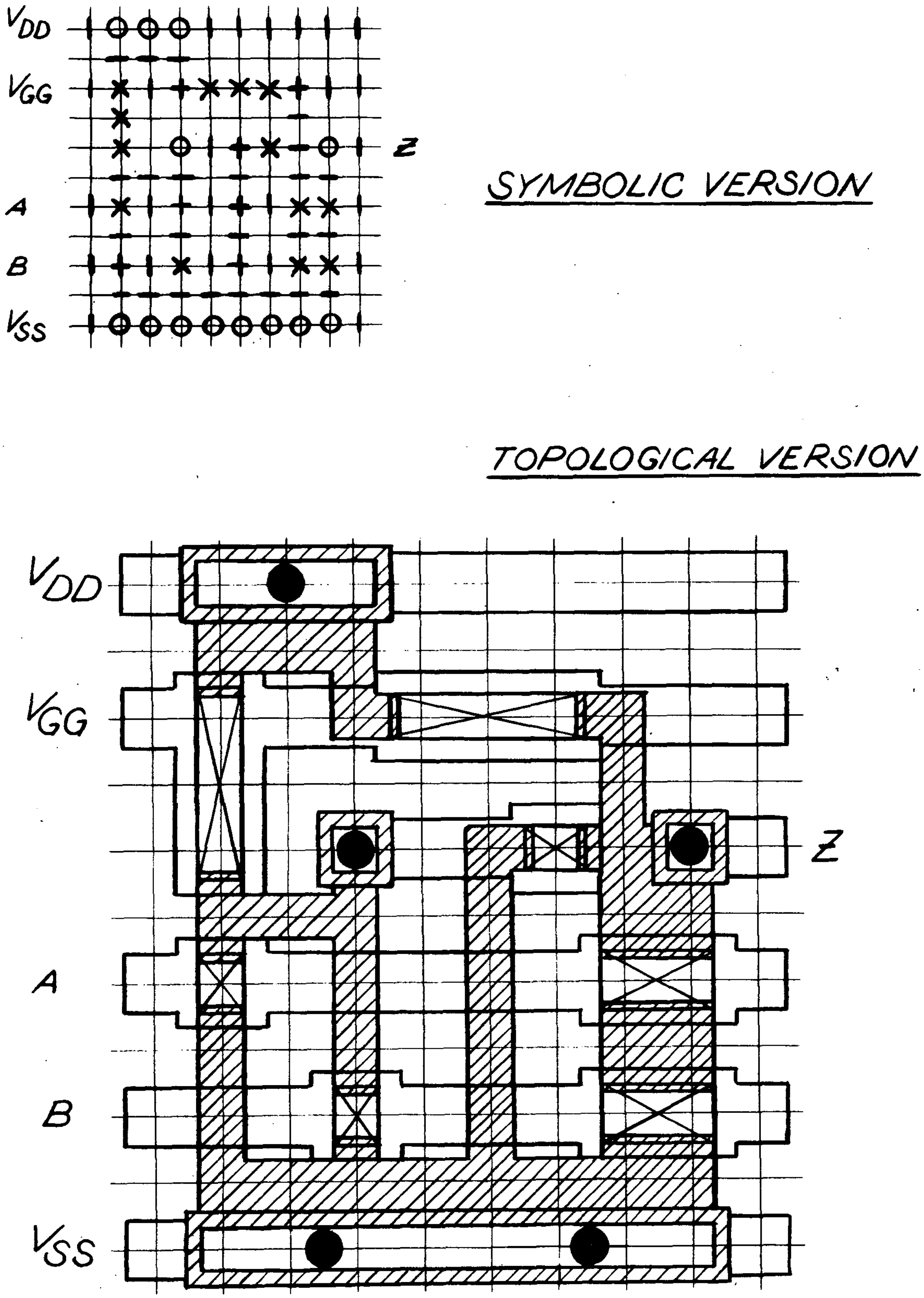


FIGURE 2.



SYMBOLIC VERSION

TOPOLOGICAL VERSION

FIGURE 3.

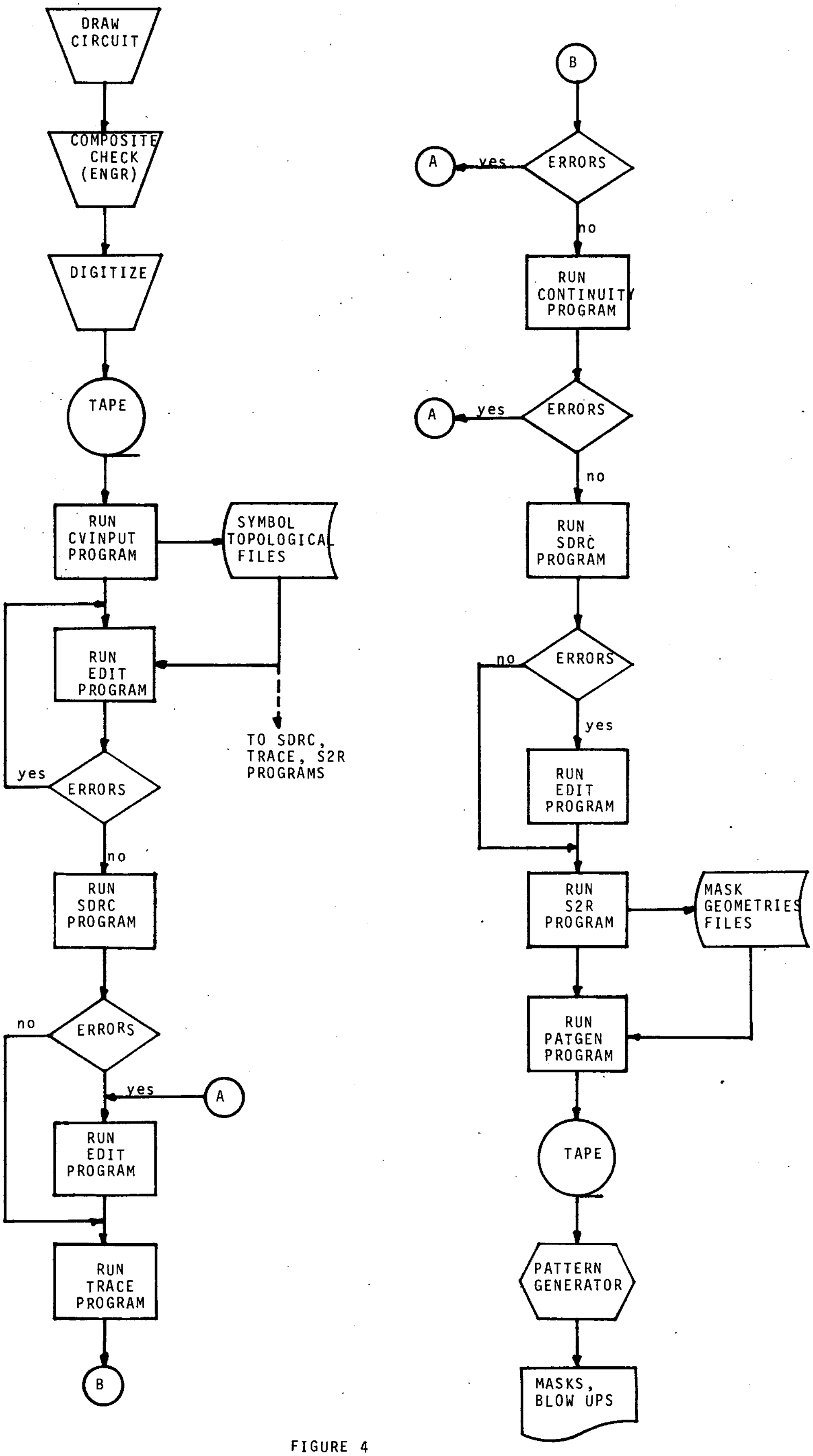

DRAW CIRCUIT
COMPOSITE CHECK (ENGR)
DIGITIZE
TAPE
RUN CVINPUT PROGRAM
SYMBOL TOPOLOGICAL FILES
RUN EDIT PROGRAM
TO SDRC, TRACE, S2R PROGRAMS
yes
ERRORS
no
RUN SDRC PROGRAM
no
ERRORS
yes
A
RUN EDIT PROGRAM
RUN TRACE PROGRAM
B
B
A
yes
ERRORS
no
RUN CONTINUITY PROGRAM
A
yes
ERRORS
no
RUN SDRC PROGRAM
no
ERRORS
yes
RUN EDIT PROGRAM
RUN S2R PROGRAM
MASK GEOMETRIES FILES
RUN PATGEN PROGRAM
TAPE
PATTERN GENERATOR
MASKS, BLOW UPS

FIGURE 4

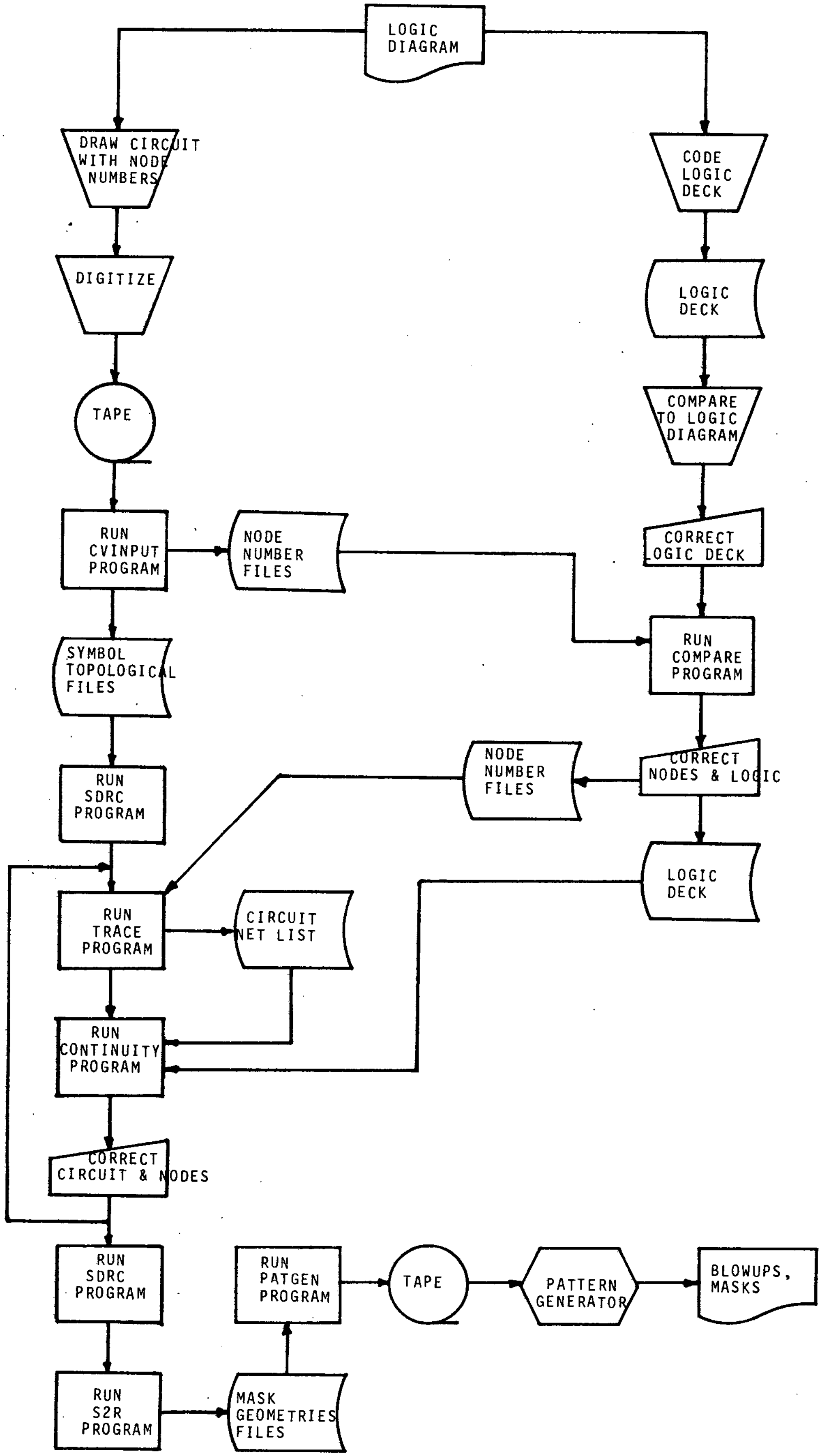

LOGIC DIAGRAM
DRAW CIRCUIT WITH NODE NUMBERS
CODE LOGIC DECK
DIGITIZE
LOGIC DECK
TAPE
COMPARE TO LOGIC DIAGRAM
RUN CVINPUT PROGRAM
NODE NUMBER FILES
CORRECT LOGIC DECK
SYMBOL TOPOLOGICAL FILES
RUN COMPARE PROGRAM
RUN SDRC PROGRAM
NODE NUMBER FILES
CORRECT NODES & LOGIC
LOGIC DECK
RUN TRACE PROGRAM
CIRCUIT NET LIST
RUN CONTINUITY PROGRAM
CORRECT CIRCUIT & NODES
RUN SDRC PROGRAM
RUN PATGEN PROGRAM
TAPE
PATTERN GENERATOR
BLOWUPS, MASKS
RUN S2R PROGRAM
MASK GEOMETRIES FILES

FIGURE 5