NASA CONTRACTOR REPORT

NASA CR-1668

NASA CR-1668

THRESHOLD LOGIC IMPLEMENTATION OF A MODULAR COMPUTER SYSTEM DESIGN

by D. Hampel, J. H. Beinart, and K. J. Prost

Prepared by
RCA CORPORATION
Somerville, N. J. 08876
for Electronics Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • OCTOBER 1970

1. Report No.	2. Government Accession No.	3. Recipient's Cat.
NASA CR-1668		

4. Title and Subtitle	5. Report Date
THRESHOLD LOGIC IMPLEMENTATION OF A MODULAR COMPUTER SYSTEM DESIGN	October 1970
	6. Performing Organization Code
7. Author(s) D. Hampel, J. Beinart, K. Prost	8. Performing Organization Report No.
9. Performing Organization Name and Address ~~RCA Corp.~~ Radio Corp of America Advanced Communications Lab. Rt. 202, Somerville, N. J. 08876	10. Work Unit No.
	11. Contract or Grant No. NAS12-2130
	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Several portions of logic of the NASA Modular Computer, to be built with LSI, were designed with threshold logic. Some of the designs were compared with TTL NAND gate implementations. The logic designs for each function were characterized by total component count, connections, speed and power. The threshold logic circuits used arrays of current switching modules to realize such functions as parity, parallel shift and rotate logic, decoding, registers and counters. The designs were based on a family of integrated threshold gates developed in 1968-1969 and which were shown to be amenable to LSI. New circuit techniques are shown to simplify logic where, traditionally, pure threshold logic has not shown advantage.

Depending on the number of components used for a TTL NAND gate, in an array, the component savings provided by threshold logic ranged from slightly less than 2 to 1 to over 3 to 1. Gate interconnection complexity was shown to be reduced by 5 to 1 in the parity circuit to 1.5 to 1 for a switch. Logic function delays were generally decreased by almost 3 to 1 with the threshold logic while the power dissipation to achieve this was ~~also reduced anywhere from 1.5 to 1 to 3 to 1.~~

17. Key Words (Selected by Author(s))	18. Distribution Statement
Threshold Logic Large Scale Integration	Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price *
Unclassified	Unclassified	72	$3.00

1. Computer Logic

TABLE OF CONTENTS

FIGURES

APPENDIX

TABLES

THRESHOLD LOGIC IMPLEMENTATION OF A MODULAR COMPUTER SYSTEM DESIGN

By D. Hampel, J. H. Beinart, K. J. Prost

RCA
Defense Advanced Communications Lab.
Somerville, N. J.

1. SUMMARY

Several portions of logic of the NASA Modular Computer, to be built with LSI, were designed with threshold logic. Some of the designs were compared with TTL NAND gate implementations. The logic designs for each function were characterized by total component count, connections, speed and power. The threshold logic circuits used arrays of current switching modules to realize such functions as parity, parallel shift and rotate logic, decoding, registers and counters. The designs were based on a family of integrated threshold gates developed in 1968-1969 and which were shown to be amenable to LSI. New circuit techniques are shown to simplify logic where, traditionally, pure threshold logic has not shown advantage.

Depending on the number of components used for a TTL NAND gate, in an array, the component savings provided by threshold logic ranged from slightly less than 2 to 1 to over 3 to 1. Gate interconnection complexity was shown to be reduced by 5 to 1 in the parity circuit to 1.5 to 1 for a switch. Logic function delays were generally decreased by almost 3 to 1

with the threshold logic while the power dissipation to achieve this was also reduced anywhere from 1.5 to 1 to 3 to 1.

2. INTRODUCTION

The objective of this program was to test the applicability of threshold logic to an existing computer design; to see what advantages, if any, would result from using integrated threshold gate arrays compared to TTL arrays.

The advent of threshold gates which are integrable in large scale arrays (Ref. 1) has led to the ability to realize the theoretical advantages of threshold logic with the full application of present technology.

The circuit philosophy is based on the use of current switches which can be interconnected to form a variety of threshold gates. Since the current switches are in themselves like emitter-coupled logic (ECL) blocks, and hence can perform an "OR" function,the threshold gates, in fact, are able to realize OR-THRESHOLD functions (combinational OR-AND gates if the threshold is set to its maximum). Also, these current switches are shown in configurations which hardly resemble classical threshold gates but which realize many of the computer functions most effectively.

These circuit concepts were applied to functions from different areas of the Modular Computer and a number of factors used in the evaluation of

logic circuits were calculated. The same criteria were applied to conventional TTL implementations and the results compared.

The impact of component savings (the total of all integrated devices-transistors, diodes and resistors) was demonstrated in a couple of cases by doing detailed component layouts and interconnect patterns as would be used in integrated circuit mask preparation. All the logic functions studied had a significantly lower (power x speed) product when implemented with threshold circuits. Since these factors can be traded off to some extent, either much lower power or much higher speed could be realized compared to conventional saturated logic circuits.

The help of Dr. R. O. Winder of the RCA Laboratories, Princeton, N. J., is acknowledged. His work in threshold logic has been instrumental toward the developments leading to this study as well as providing helpful suggestions on this program.

3. THRESHOLD LOGIC CIRCUIT CHARACTERISTICS

The integrated threshold gates use current switches, summing resistors and emitter followers. Their functions are the following:

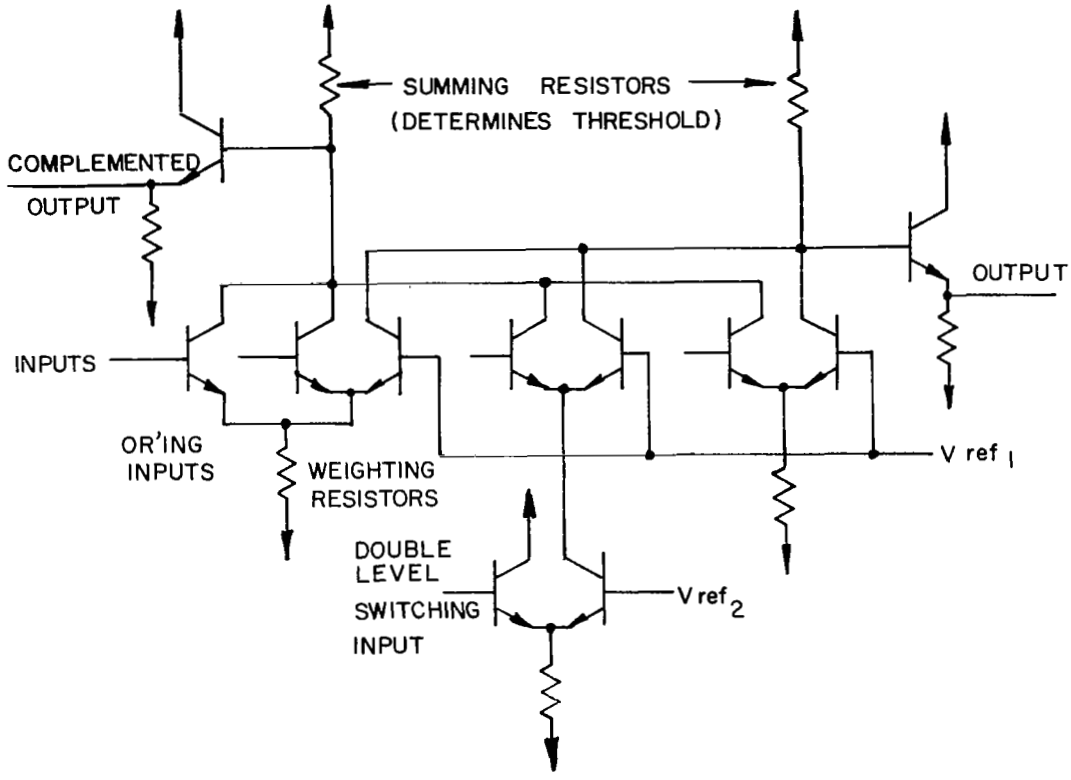
-Current switches; for isolating inputs, comparing them to a reference and controlling well-defined currents used for summing

-Summing resistors; for converting currents to voltages which are above or below (binary 1 or 0) the reference

-Emitter followers; for level shifting and providing drive capability

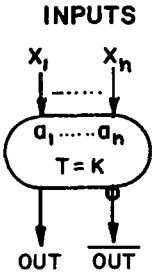
Figure 3-1 shows how these elements are connected in the general case. The logic symbols for various combinations on these elements is given in Figure 3-2. Generally, the four types of functions represented can be combined.

For example, OR'ing inputs can be used on double level gates, etc. Also, virtual OR (wired OR) can be done on any gate's outputs. Saturation is avoided in all circuits by the use of appropriate clamp circuits-cross connected diodes in most circuits. Hence, the circuits are naturally high in speed. The nature of the circuitry is such that complementary outputs are always available if desired. Excellent noise immunity stems from the use of the current switch. All inputs are always compared to a reference and do not interfere with the output signal until they are close (within a few millivolts) to the reference. Noise immunity as well as signal swing depends on gate size, threshold setting, and the power supply used. Typically, noise immunity approaches 50% of the signal swing. The signal swing is about 12% (minimum) of the power supply. Hence, for a 5 volt supply, signal swing is 600 mV and noise immunity almost 300 mV. Since current switching is used, noise sources are minimized; power transients are very small.

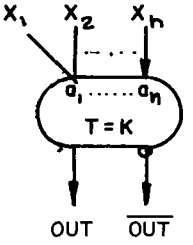


GENERAL THRESHOLD LOGIC CIRCUIT

Fig. 3-1



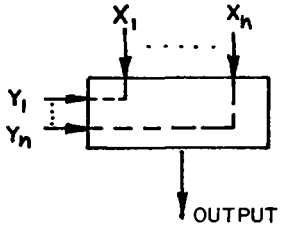
GENERAL THRESHOLD GATE



EXP. OF GATE WITH OR'ING INPUT $(X_1 + X_2)$ WITH WEIGHT a_1

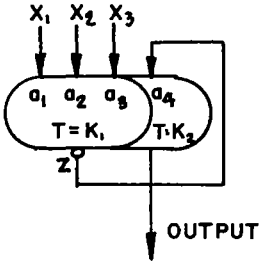
a_1 = WEIGHT OF INPUT – REPRESENTS A CURRENT SWITCH WITH a_1 CURRENT.

k = THRESHOLD OF THE GATE (COULD BE 1 THRU h)



EXP. OF DOUBLE LEVEL GATE

Y_1 CONTROLS X_1, ETC.



EXP. OF DOUBLE FUNCTION GATE WHERE OUTPUT IS DETERMINED BY $a_1 - a_4$ STATES AND Z IS DETERMINED BY $a_1 - a_3$ STATES.

THRESHOLD GATE SYMBOLS USED IN LOGIC DIAGRAMS

Fig. 3-2

The circuits are not unlike emitter-coupled logic (ECL) in concept, except that current summing is done in combination with OR'ing on the inputs enabling the realization of OR-THRESHOLD functions and OR-AND functions. Virtual OR'ing on the outputs is, of course, also possible.

There is no penalty in using current switches this way as long as resistor ratios are held fairly close. Actually for all of the gate types used in this study, some critical ratios are $\pm$3%; many other ratios are $\pm$5% or higher. Moderate geometries readily achieve this. Fairly good V_{be} matching is also necessary but this is a natural attribute of integration. Differences of $\pm$20 mV pose no problem and merely cut in slightly to the noise immunity. Such differences in V_{be} have a negligible impact on circuit yield.

The circuits track with power supply and temperature over a large range. For example, if an LSI chip were designed at a supply of four volts in magnitude (+4 or -4), it would work at any value from 3.2 to 6 volts. The circuits work for -25°C to +125°C; they are actually faster at the higher temperature.

The natural higher speed of these circuits compared to TTL can be used to advantage to further minimize component count. One example is in the arithmetic unit; carry ripple through would be acceptable in place of carry lookahead.

Two innovations were applied to the basic threshold gate designs resulting in improved implementation of some of the MC logic. These were.

1. The use of a threshold gate for performing two different functions of its inputs simultaneously; one on its true output and another (different one) on its complemented output. This technique was used in the parity circuits in the incrementer of the L1 character and in the arithmetic circuit.

2. The application of double level switching to the threshold gate. This technique, commonly applied to current mode logic, permits switching and multiplexing functions to be realized with very low component count and power dissipation. Its use was shown in the CAU switch logic, flip-flop circuits and in much of the L1 character.

These circuits are described in the Appendix.

3.1 Circuit Flexibility

The improvements realized in the threshold circuit approach stems, in a large measure, from its flexibility. Current switches, used for logic and control of gates are combined in double level switching circuits with OR-ing, AND-ing, or other threshold settings. Clusters of current switches can be connected to perform major functions - both combinational and sequential - in themselves. The standard approach is to synthesize these functions from rather inflexible predefined blocks such as NAND or NOR gates.

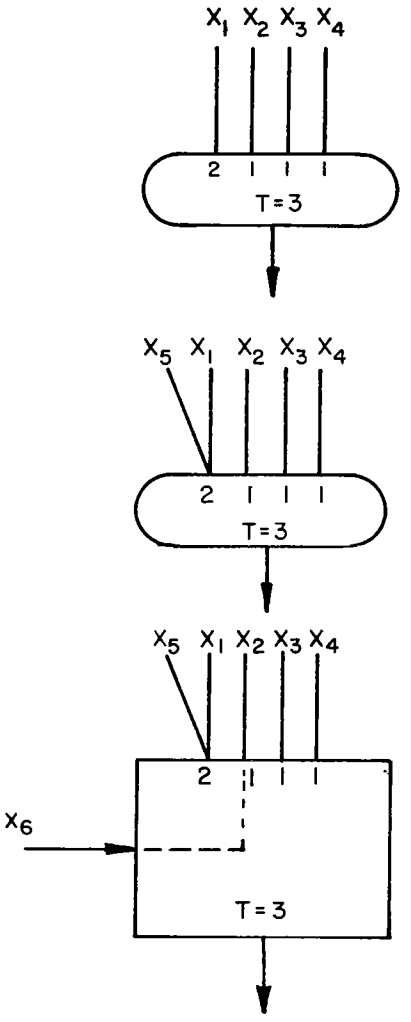
3.2 Examples

The circuit and logic design become indistinguishable from one another in an array of current switching modules as we are using them; in TTL, the circuit of each gate is precisely the same. For example, assume a threshold gate has been specified for a particular function of 4 variables X_1, X_2, X_3 and X_4; (2, 1, 1, 1; T = 3). The function is $[X_1(X_2 + X_3 + X_4) + X_2X_3X_4]$. A fifth variable can be introduced, X_5, as a transistor in parallel with the X_1 input. The function is now $[(X_1 + X_5)\ (X_2 + X_3 + X_4) + X_2X_3X_4)]$. By adding another variable, X_6, as a second level switch for the X_2 input, the function is $[(X_1 + X_5)\ (X_2X_6 + X_3 + X_4) + X_2X_6X_3X_4]$. This sequence of gates is shown in Figure 3-3.

In each modification a component or two was added to the initial gate. Without this technique, in conventional NAND gate designs, a gate or two would have to be added to get from the original function to the desired function. Using these techniques on pure threshold gates allows the designer to add control functions on the component or circuit level rather than the gate level.

3.3 Implication of Flexibility

One of the interesting features of threshold gates is their universality. That is, one gate can generally be used to realize a number of different useful functions either by biasing inputs or changing weights. Thus, it is a more powerful building block. Less block types are required and less total blocks would be used when compared to a Boolean block. (A way



CIRCUIT FLEXIBILITY

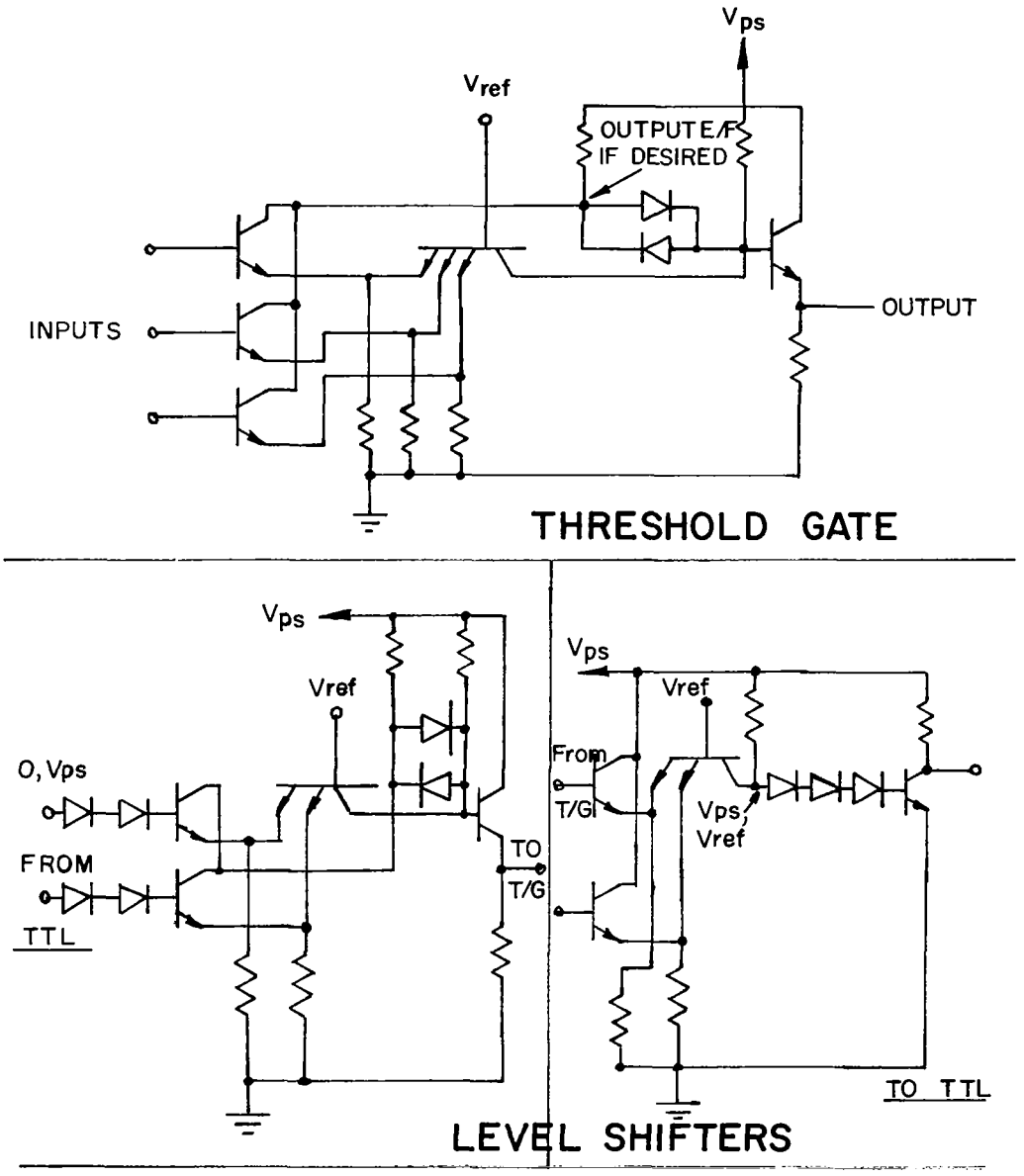
Fig. 3-3

of showing this is to place a number of gates in a package with all inputs and outputs accessible; then, a system could be built with fewer package types and fewer total packages if the package contains threshold gates.) [Ref. 3] This feature is brought out in the customized LSI arrays considered in the Modular Computers in the sense that the array can be considered as a mosaic of building blocks. What we are saying is that a given size array will be capable of being metallized to implement more functions than a similar size array composed of NAND gates for example. The implication is that the characters (realistically a package of chips) of the MC can be realized with a fewer number of chips - chips with the same diffusion masking and first layer of metallization pattern but only with different second level of metal.

3.4 Compatibility

By using a +5 volt supply for these new logic gates, they can be easily made compatible with +5 volt TTL with use of simple level shifters. A threshold gate is shown in Figure 3-4. It is drawn here in a form which resembles a TTL gate. Figure 3.5 shows the circuits required to get into and out of an array of such threshold gates. A stage of logic can be done in the level translation process.

It appears possible to have an array of threshold gates working at their own internal levels which can be buffered on the periphery of a chip with level shifters (such as those shown) to work with some existing family



COMPATIBILITY

Fig. 3-4

logic. The level shifters can provide a level of logic in the process. In most cases, such a chip will still have preserved the advantages of threshold logic.

3.5 Testing

The fact that all the threshold circuits use a reference for defining the binary signal at every input to a gate offers a degree of freedom in array testing. By bringing the internally generated reference voltage to an external pin, its value can be varied, in testing so that all gates can be forced to a high or low state. Several reference points can be brought out to selectively force different logic paths or registers to known states so that simplified, meaningful tests can be accomplished.

In the test and diagnostics area, the major problems arise with the sequential circuitry in an array[1]. If the states of these circuits can be easily controlled considerable improvement in test and diagnostic routines are possible. The threshold levels of the sequential circuits of this study provide such a flexibility.

3.6 Circuit Comparisons

Comparisons were made for various pieces of logic, in the modular computer, between the threshold logic and conventional TTL NAND gate

1. In private discussion with H. Jacobwitz, RCA Project Engineer on NASA, ERC Test and Diagnostics Program.

implementations. The following points must first be emphasized:

-The threshold logic circuits are basically non-saturating current switching circuits. Hence, a gate's delay is naturally less than a TTL gate delay. Furthermore, since fewer delays are normally encountered between input and output, the apparent speed for a function is even greater. It is seen that power dissipations were comparable in many cases, while speeds were 4 times higher with the threshold circuits. These designs were based on an existing integrated threshold gate. In an optimum design these two factors can be traded off to save power, by increasing circuit impedance levels.

-The number of components per TTL gate within an LSI environment will be less than the number in a gate used for driving off the chip. The performance of a TTL gate is dependent on its circuit complexity (noise immunity, speed, fan-out, etc.). The number of components-sum of all resistors, transistors and diodes-per TTL gate could range from 5 to 9. In the T/L only total components and not gate count is of importance. The component count of a threshold gate is higher (but not by much) for just doing a simple Boolean function. Compare Figure 3-4 with a 3-input TTL gate. However, in the designs examined, a gate is never used for just NAND or NOR - it does complex combinational logic within itself. The addition of relatively few components accomplishes what multi-gate networds are required for. Frequently, the use of these gates obviate the need for the complemented input of a variable. Other times, it needs complemented inputs but since the gates provide complemented outputs, component count

is hardly affected. In TTL an inverter uses about the same circuitry as a NAND gate.

-A triggerable flip-flop assembled from NAND gates was assumed to contain 6 TTL gates. There are custom circuits which use fewer components than required by 6 gates, but these frequently result in a sacrifice of reliability (such as specifying clock rise times, less tolerance to power supply variations, etc.).

-The noise sources in an array of current switches are much less than in TTL primarily due to elimination of output circuit current transients.

-Connections were used as a factor in comparisons. This is defined as the total number of inputs to all clusters of component recognized as a gate. This is then a measure of the 2nd layer of metallization density. Its minimization is considered to result in higher reliability of a chip. (Elimination of the 2nd level of metal would be the ultimate, but in the complexity of the chips postulated for this study this is impossible.) The first level of metal contains power, ground, and component interconnect. (In the case of T/L it also contains an internally generated reference source.) To the extent that the second layer of metal is minimized, there are fewer chances for shorts.

4. APPLICATION OF THRESHOLD LOGIC TO NASA MODULAR COMPUTER (MC)

The NASA MC (Ref. 2) will consist of a number of modules, connectable to operate as three parallel processors or to operate as a single processor. In the latter case, combinations of modules from the three possible processors

can be patched to form the single processor. The modules perform the functions of arithmetic units, control units, memory units and input-output units. These modules or units are connected to one another through swithces under control of the configuration assignment unit (CAU). The availability of good modules is determined by means of hardware-software tests.

All of the units (or modules) are, in turn, made up of building blocks defined as characters. A functional character set (10 characters) has been specified. This group of logic arrays forms a self-sufficient family of blocks with which the units of the MC are synthesized.

Threshold logic was applied to three different designs:

1. A specific subcharacter circuit-parity generator and checker
2. A complete character in detail - the L1, General Logic
3. Portions of the Configuration Assignment Unit which consists, in a large part of: P2 characters, Up/Down Counters; P3 characters, Switch; and G1 characters, Register Storage.

For each of these areas, the following was determined for a specific Threshold Logic design:

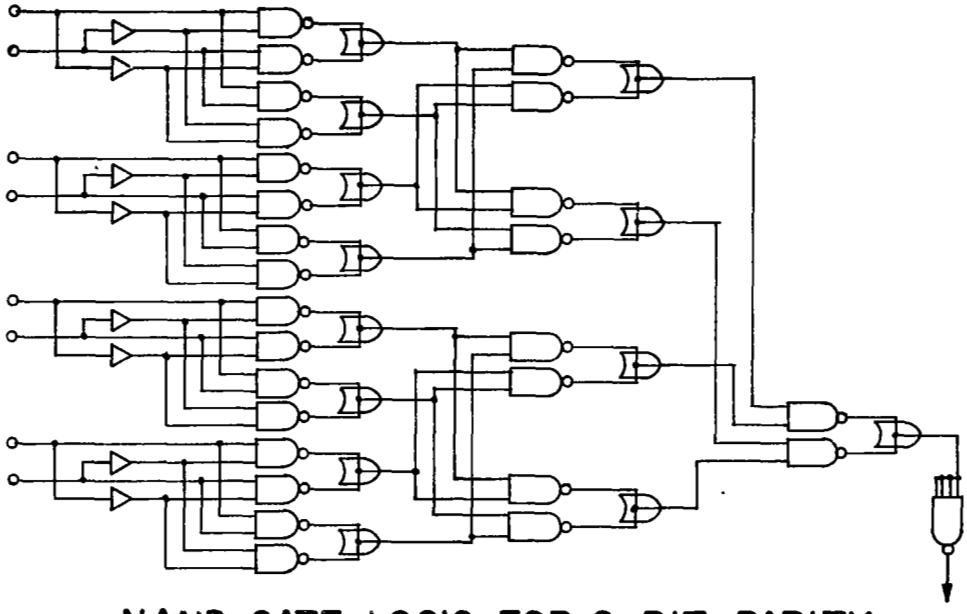
- o Total Power
- o Circuit Delay
- o Component layout to estimate chip area requirements

Comparisons were made with TTL designs. For the case of the parity circuits a detailed design was available. The L1 character had an approximate gate total with which to compare. The CAU switch circuit (P3 character) design was known, but the TTL gate count for the other CAU functions were estimated.

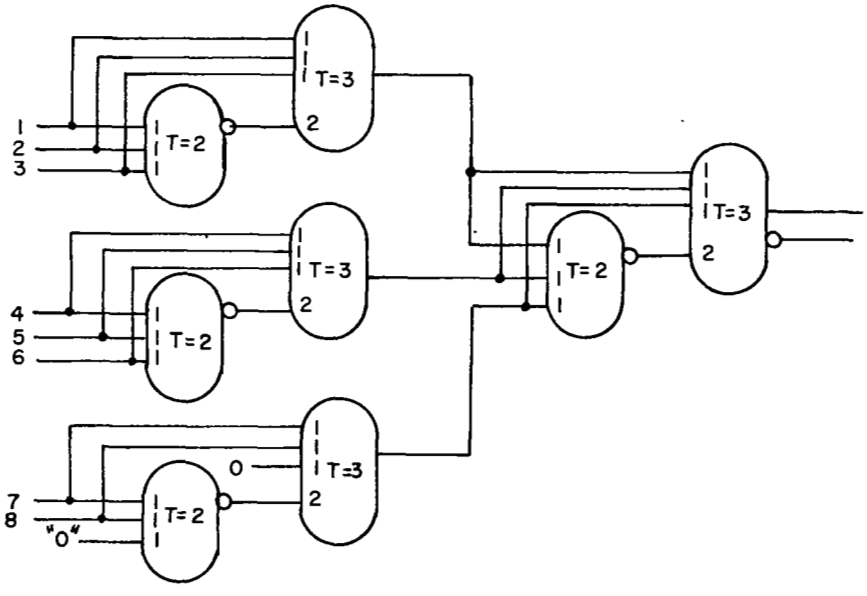
4.1 Parity Circuits

Parity circuits have traditionally lent themselves to efficient T/L implementation (since they are symmetrical functions). Designs are available for minimum delay or minimum gate performance for any number of bits for parity generation or checking. In this study, a minimum component 8-bit parity circuit is required. The speed will automatically be better than any TTL design because of the circuits used. Component count and not gate count is of prime importance in these circuits (as well as the others to be designed) since, in LSI, this is the determining factor in chip size.

A very straightforward approach, and one which gives good results, is to use the well-known T/L full-adder (with carry out not used) two-gate circuit to realize parity for 3 bits at a time. This is shown in Figure 4.1-1 along with the TTL NAND gate realization. A simplification of this approach, involving the use of the 3-input majority gate for two functions simultaneously (one each on its complementary summing resistors) represents the ultimate in circuit simplicity and performance. This is shown in Figure 4.1-2. The three input clusters in the adder design were duplicated - for the (1, 1, 1; T = 2) gate and the (2, 1, 1, 1; T = 3) gate. This duplication

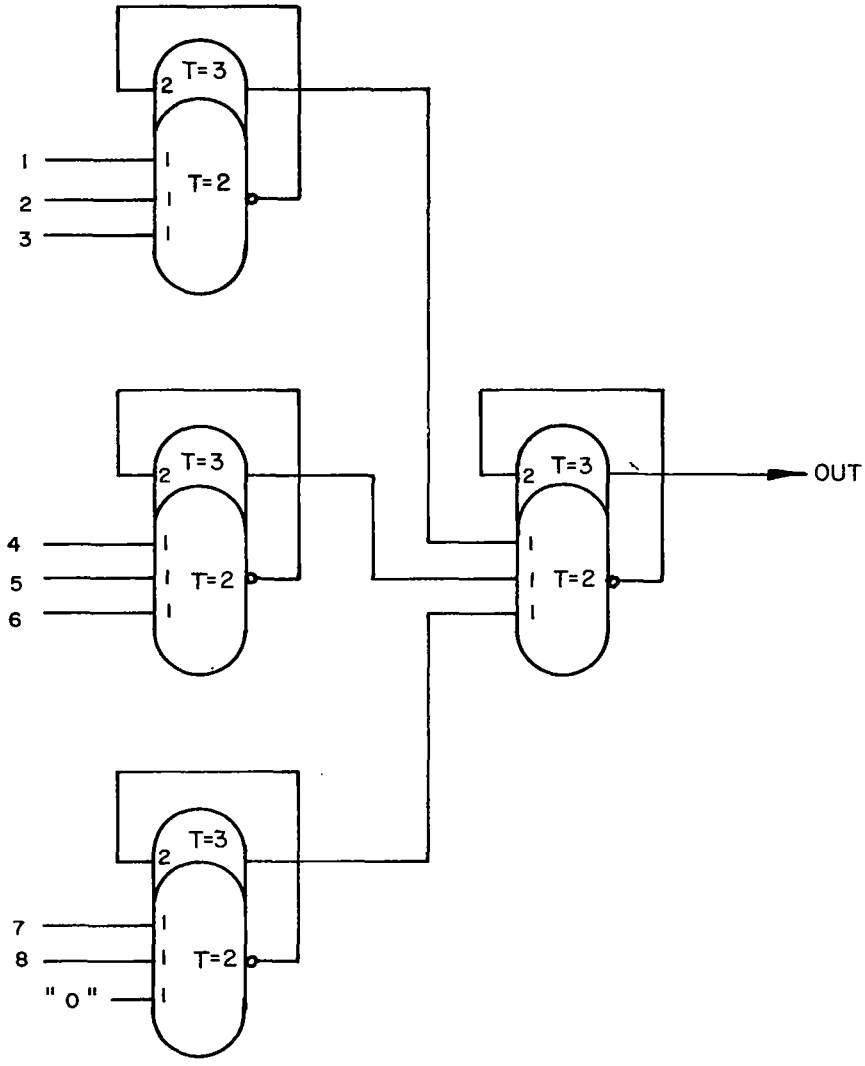


NAND GATE LOGIC FOR 8-BIT PARITY
(From Parity Logic Board SPC-102 Dwg. 32801)



THRESHOLD GATE LOGIC FOR 8-BIT PARITY

Fig. 4.1-1



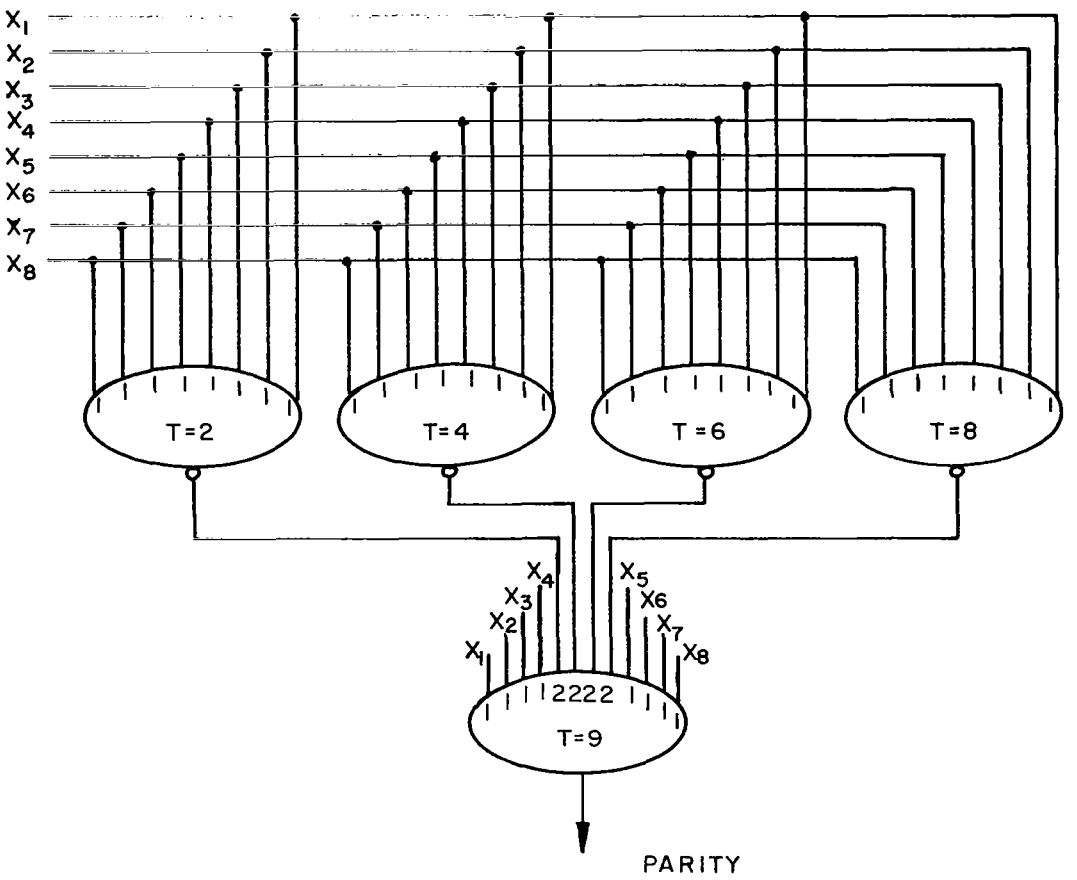
IMPROVED THRESHOLD LOGIC PARITY CIRCUIT
(8 BIT)

Fig. 4.1-2

is avoided by allowing the 3 input current switches to provide an output on their inverted side to feed a double weighted switch. This, in turn, sums on a resistor which is also used for the true side summing of the 3 input switches. A more detailed description of this circuit is given in the Appendix. This approach results in a 8-bit parity circuit which uses but 79 components; it does the job of 35 TTL gates.

The circuit delay of the threshold gate implementation of Figure 4.1-1 and Figure 4.1-2 are essentially the same and are composed of four stage delays. A circuit which gives two stage delays is shown in Figure 4.1-3. Here, there are 5 large gates with total component count significantly greater than that of the design given. Also, note the multiplicity of connections for each input of four of the gates - this parallelism is in effect responsible for the increased speed. A four-gate design (minimum gate, not shown) again has much greater component count and connection requirements. These points are emphasized here to show that a threshold logic design is not necessarily attractive for a given situation unless a suitable approach is chosen.

Layout for Parity Circuit.--Having completed the circuit design and logic design, a component layout for integration was designed to establish chip area requirements. The circuit topology was governed by the following factors:



MINIMUM DELAY PARITY CIRCUIT

Fig. 4.1-3

Transistor Geometry

Single emitters	2.2 x 2.6 mils
Multi-emitters (2 each)	2.6 x 3.6 mils
Contact openings	0.2 x 0.4 mils min.

Resistors

For 2% ratios	1.5 mils min. width
For emitter-follower loads	0.7 mils width

Metal

Two layer metallization

Min. width	0.3 mils
Min. spacing	0.3 mils

Bonding Pads

4 x 4 mils

Figure 4.1-4 shows the component layout with single emitter transistors, two-emitter transistors and both wide and narrow resistors. The two overlays show first and second layer metallization patterns respectively. The first layer distributes ground, power and reference supplies, as well as some gate interconnections.

The total area (without bonding pads since this circuit would probably be within a chip) is 1500 sq. mils. All reference bias and clamp circuitry

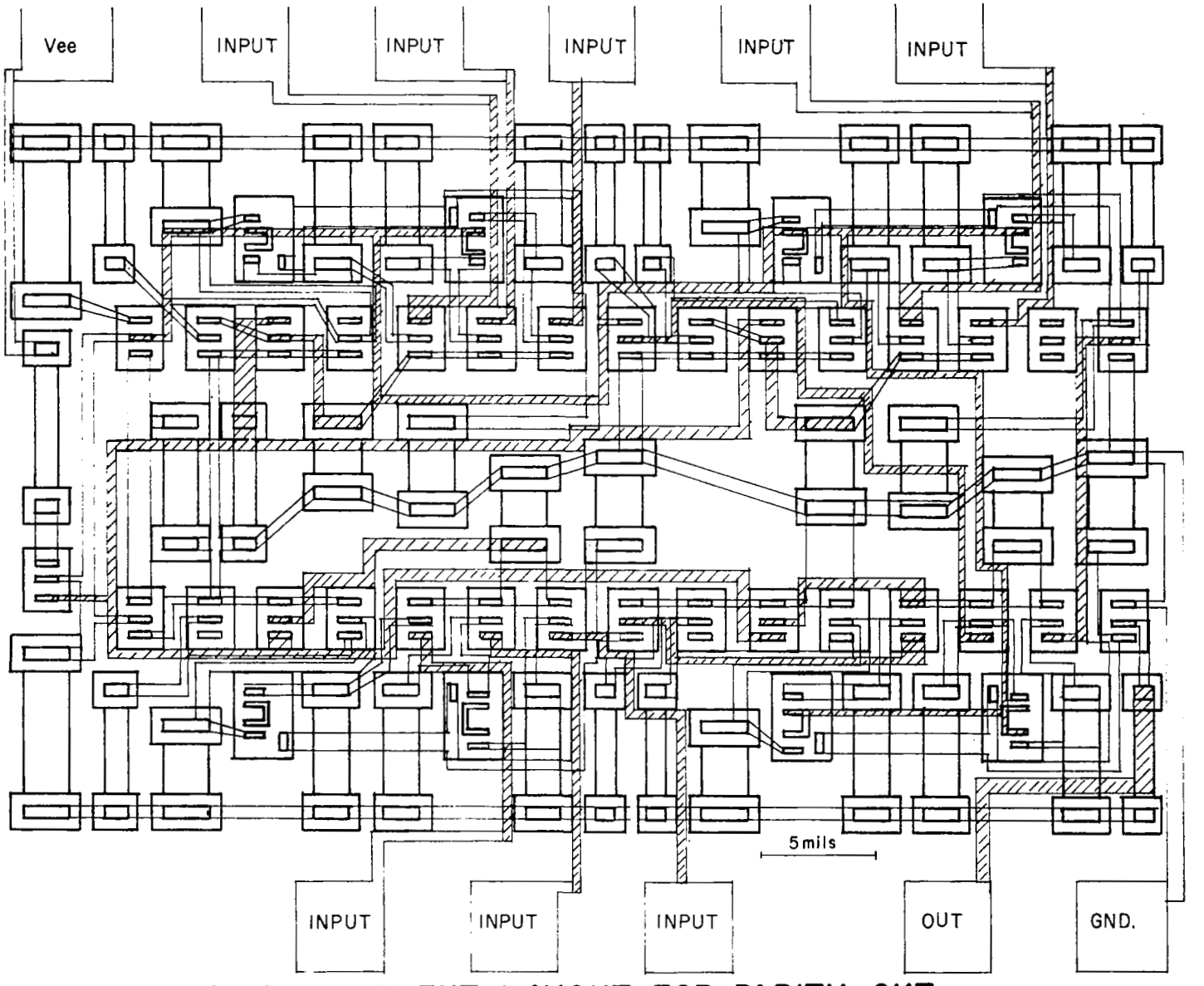


Fig 4.1-4 COMPONENT LAYOUT FOR PARITY CKT.

is included and the topology has not been optimized; notice the unused area in Figure 4.1-4. Nevertheless, this circuit replaces 35 TTL gates implying an equivalent of 43 sq. mils per gate.

The pertinent guide lines provided by NASA for LSI were:

Diffusion Geometry Separation	0.5 mil
High Frequency Geometry	0.2 mil
Diffusion for Resistors	1.0 mil
Metallization Min. Spacing	0.2 mil
Metallization Max. Width	0.2 mil
Bonding Pads	4 x 5 mil

These figures generally fall within those initially postulated and would result in slightly smaller layout areas for the parity circuit and the L1 logic of Section 4.2

The following, Table I, summarizes the characteristics of the parity circuit and compares them to the TTL design.

4.2 L1 Character

The L1 character provides the basic logic functions selectable by the microprogram. It is 8 bits wide and contains the following logic:

Bussing

Decoding

Rotate, Shift, Complement

SUMMARY OF CIRCUIT CHARACTERISTICS

8-Bit Parity Circuit

	No. of Gates	No. of Components	No. of Connections	Avg. Delay (Nano-sec.)	Avg. Power (mW)	Avg. (Power x Speed) 10^{-12} (watt-sec)	Chip Area (sq. mils)
Threshold Logic	4	79	11	32	248	7,950	1500
TTL	35	210-280	61	87	366	32,000	-

TABLE I

Incrementer

L Register

Gating to Output Bus

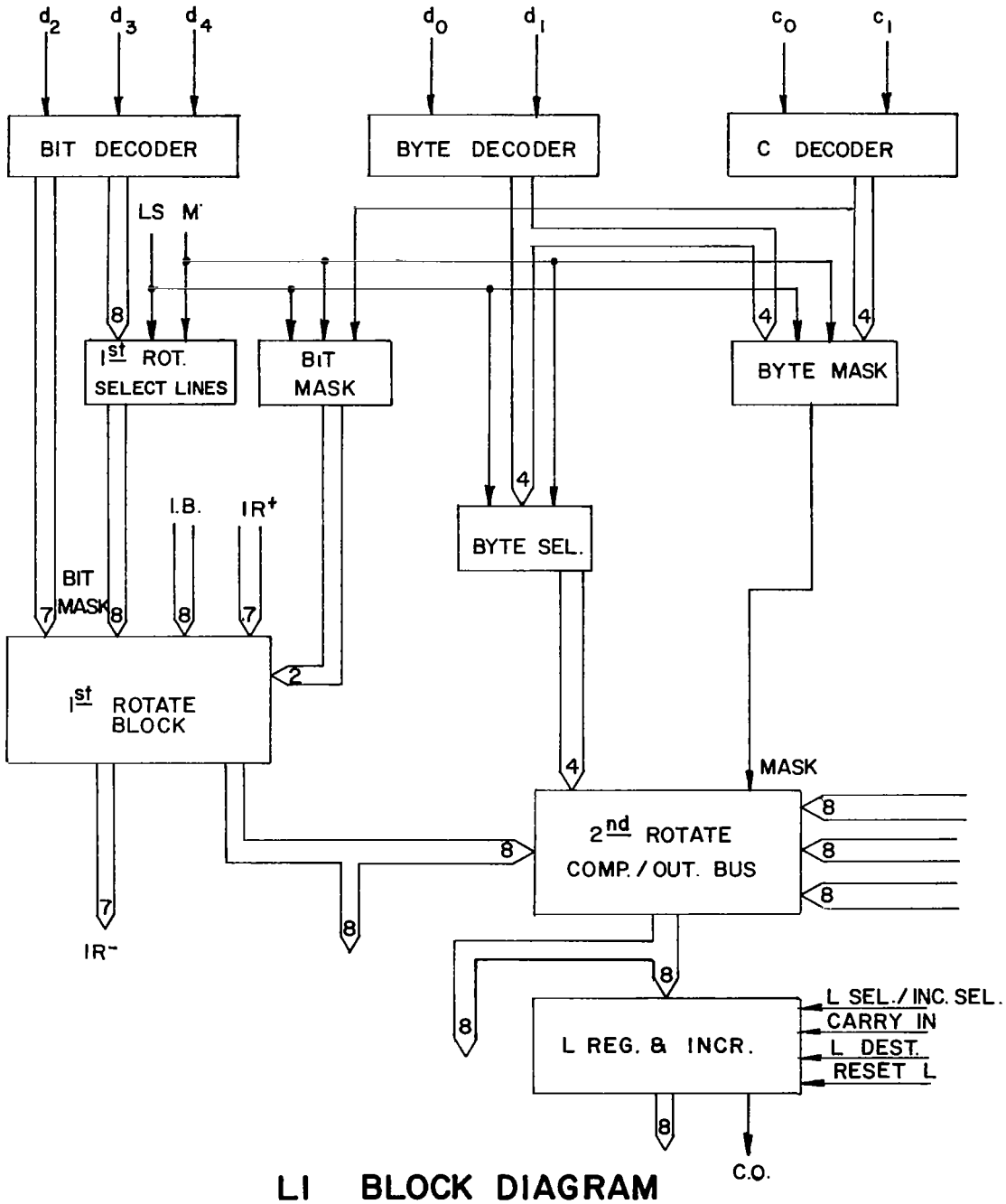
Shifts and rotates are implemented in a single step for 1 to 31 positions (rather than serially) for fast manipulation of data.

A block diagram of the L1 as it will be synthesized with threshold gates is given in Figure 4.2-1. The top portion contains the decode and control, while the lower portion contains the general logic functions.

The following sections briefly describe each of the blocks and reference the circuit diagrams which realize each function.

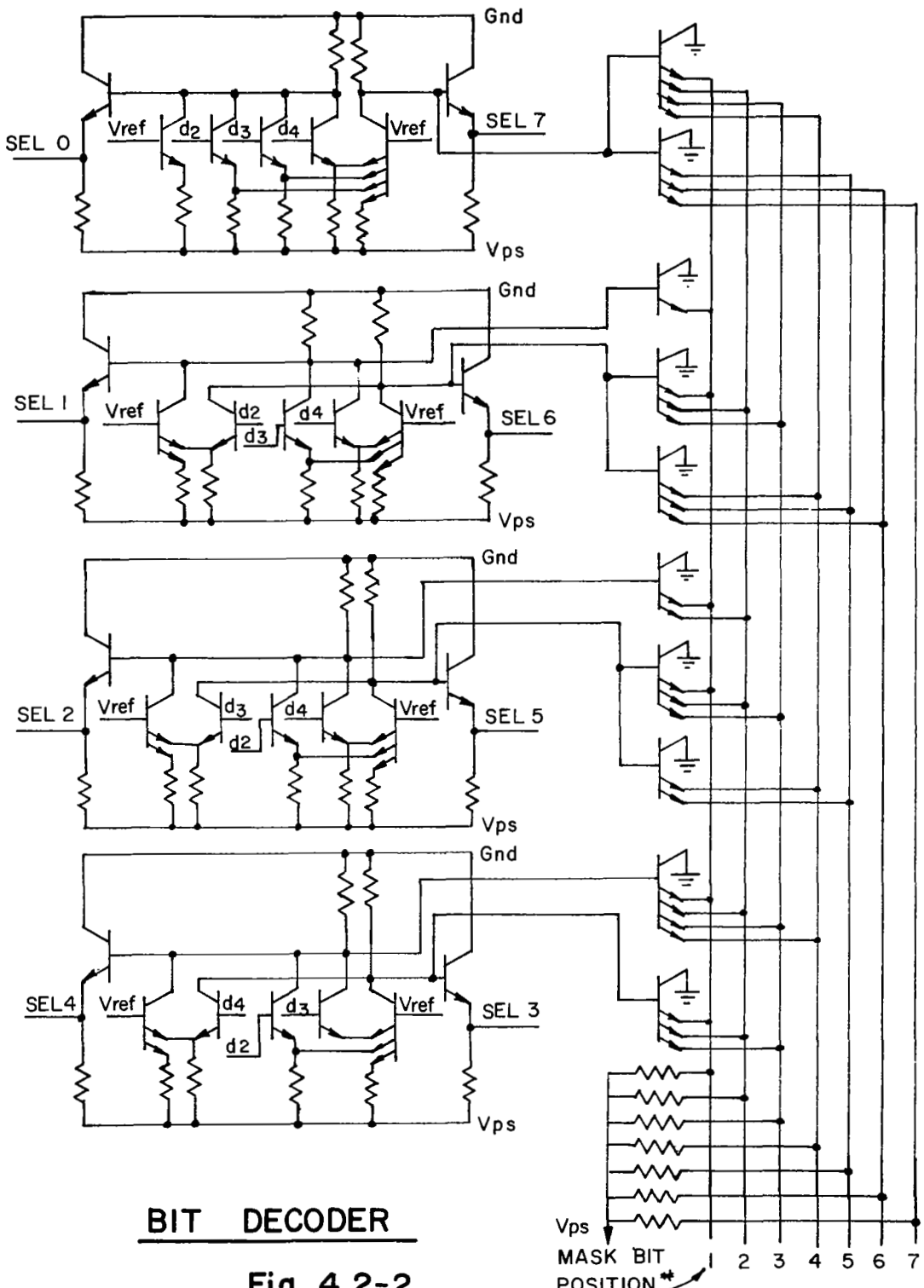
Bit Decoder (Figure 4.2-2).--This decoder has a three-line input (d2, d3 and d4) and has two output functions. The first set of output lines is a "one of eight" decode, and it is used for the 1st rotate select lines. The second set of output lines (Bit Mask) is used to force the output of the 1st rotate to "0". The number of bits that are forced to a "0" is determined by the bit decoder while the bit mask determines from what direction the blanking will occur.

These functions are accomplished with AND/NAND gates in combination with a matrix formed by interconnecting the emitter follower outputs of the gates.



LI BLOCK DIAGRAM

Fig. 4.2-1



BIT DECODER

Fig. 4.2-2

First Rotate Select Lines (Figure 4.2-3).--This block has the 8-line input from the bit decoder, the LS and m inputs of the microinstruction, and it decodes for a rotate, shift left (S/L) or shift right (S/R) command. If a rotate or S/L is decoded, the output of the bit decoder is fed into the first rotate; if the command is for a S/R, the 2's complement of the bit decoder is fed into the 1st rotate select lines.

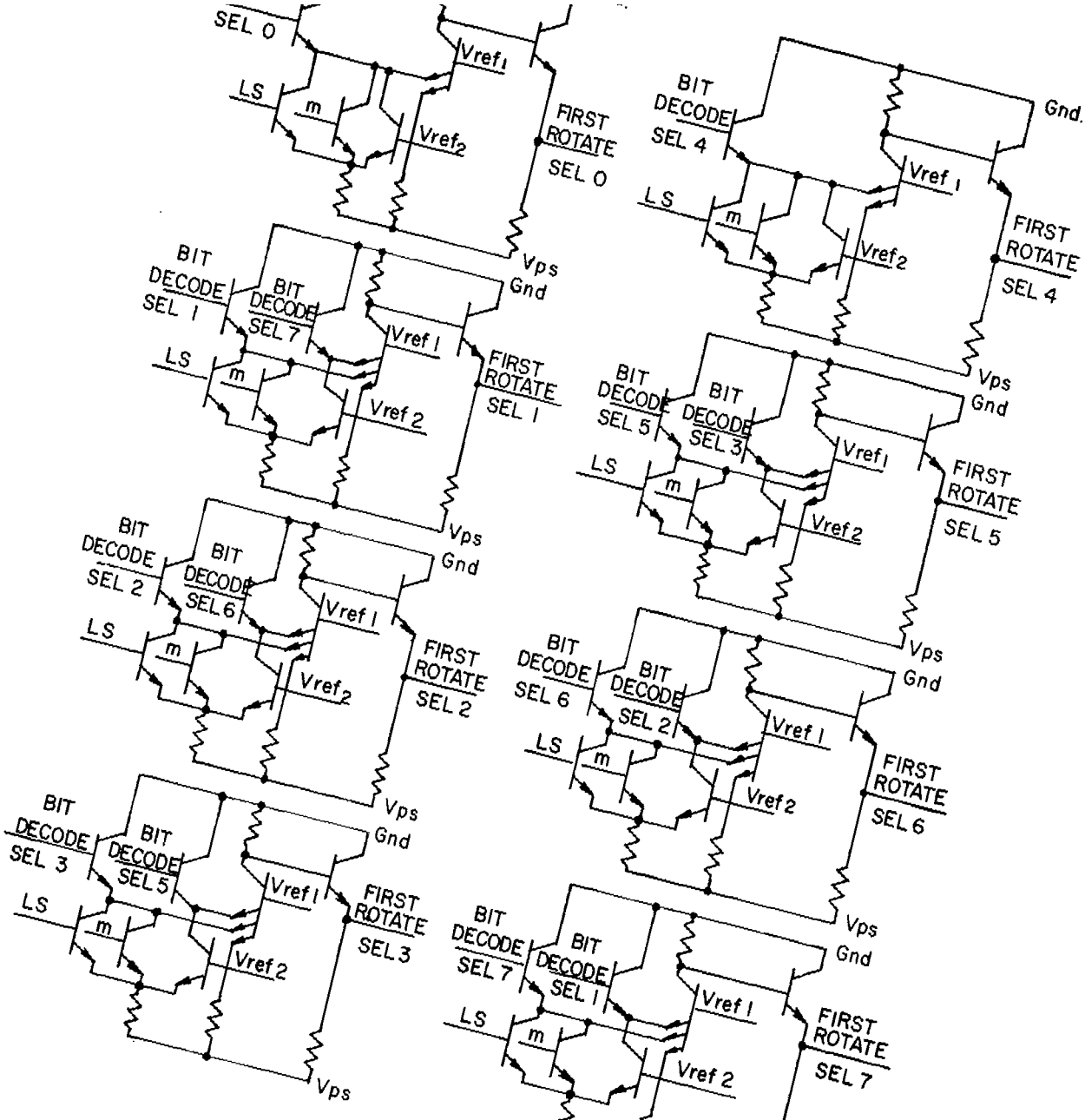
This circuit uses double level switches to effectively generate the rotate commands by AND-OR logic.

Decoder [Byte and C] (Figure 4.2-4).--Both decoders have a 2-line binary input and a one of four output. Output of the byte decoder is used as the inputs to the byte select and the byte mask blocks.

The C decoder determines what bit positions the L1 character is located in (up to 4 L1 characters may be connected together). Listed below are the C inputs and the bits assigned to that code.

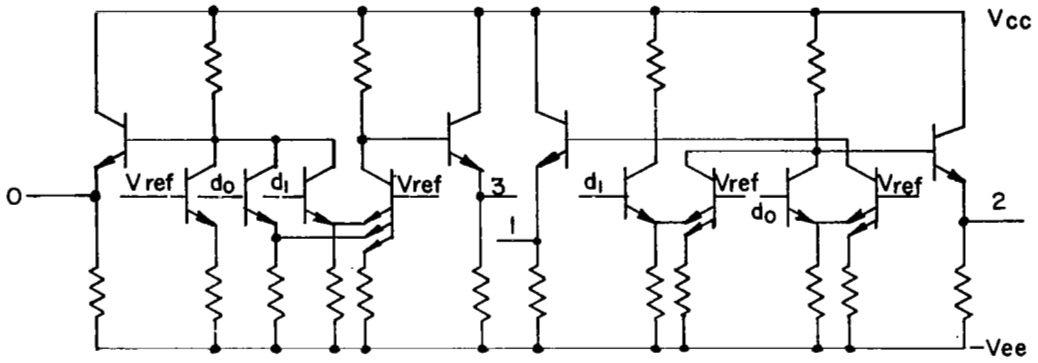
Input	Bit Positions	Byte
00	1-8	0
01	9-16	1
10	17-24	2
11	25-32	3

Bit Mask (Figure 4.2-5).--This circuit is used only during shift commands and it selects the proper L1 character to mask. For a S/L the mask is used to blank from the L.S.B. to the M.S.B and vice-versa for a S/R. The



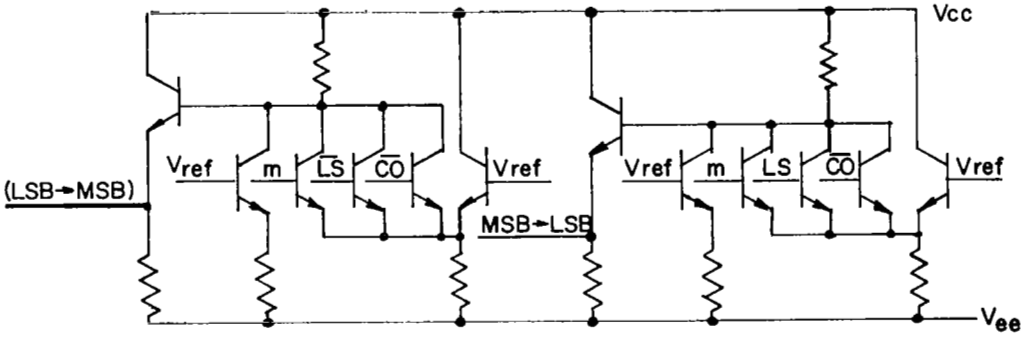
FIRST ROTATE SEL. LINES

Fig. 4.2-3



DECODER (for BYTE & C)

Fig. 4.2-4



BIT MASK

Fig. 4.2-5

number of bits to be masked is determined by the bit mask output lines of the bit decoder.

Both the functions of 4.2-4 and 4.2-5 are realized with the same type of circuit as that of 4.3-2.

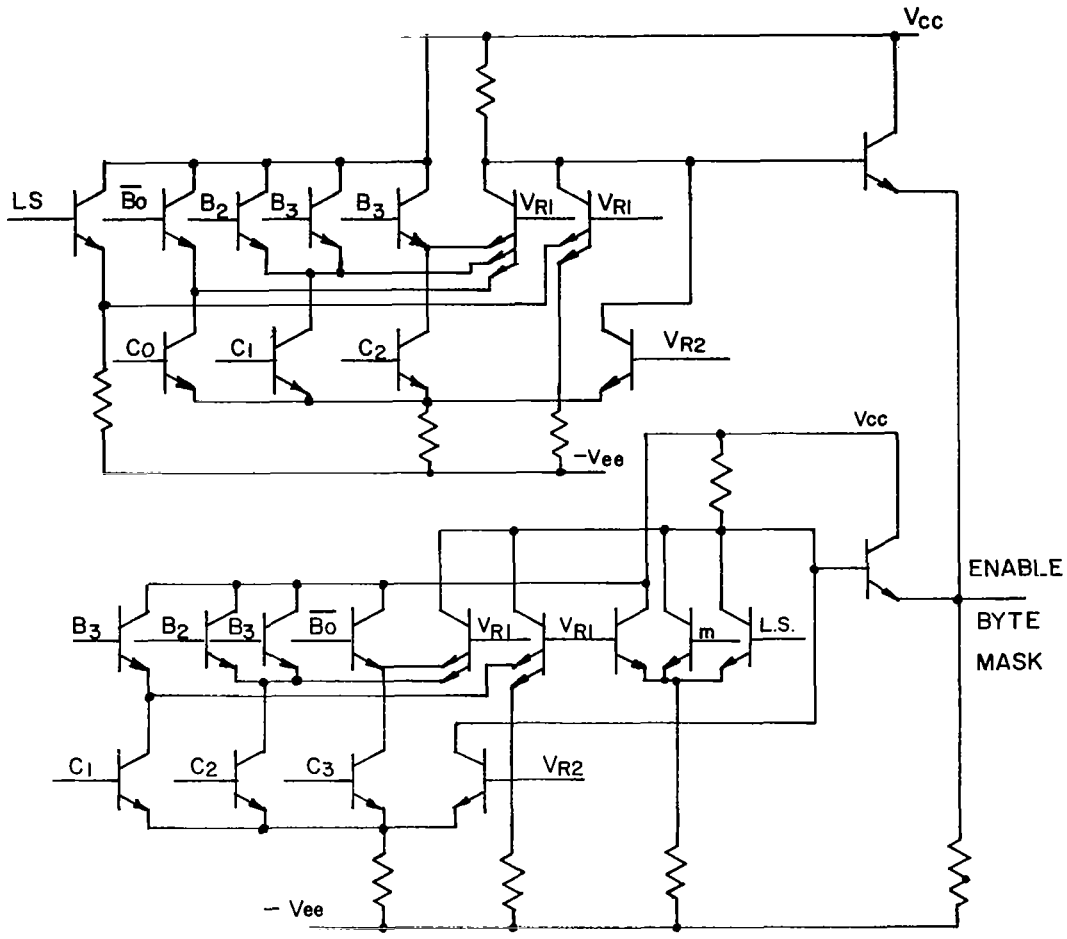
Byte Mask (Figure 4.2-6).--This block is similar to the bit mask with the exception that it may be applied to more than one character at any given time and it masks the entire byte.

Byte Select (Figure 4.2-7).--This block is similar to the 1st rotate select lines. During a rotate or shift command, the function is the same as the 1st rotate, but during a complement (do $\rightarrow$ d4 = 0, LS = 0 and m = 0) all four output lines are forced to a "0".

These two byte functions use double level switches.

1st Rotate (Figure 4.2-8).-- This block is comprised of 8 switching circuits and it has 15 data input lines and 17 control lines. The output of this block is one of the data lines, depending upon the select lines, or it is a logical zero if a mask command is generated. The mask command overrides the select lines, and it can be used to force up to 7 of the output lines to zero.

Eight basic switches of this type shown in Figure 4.2-8 are connected together to form the 1st Rotate.



BYTE MASK GENERATOR

Fig. 4.2-6

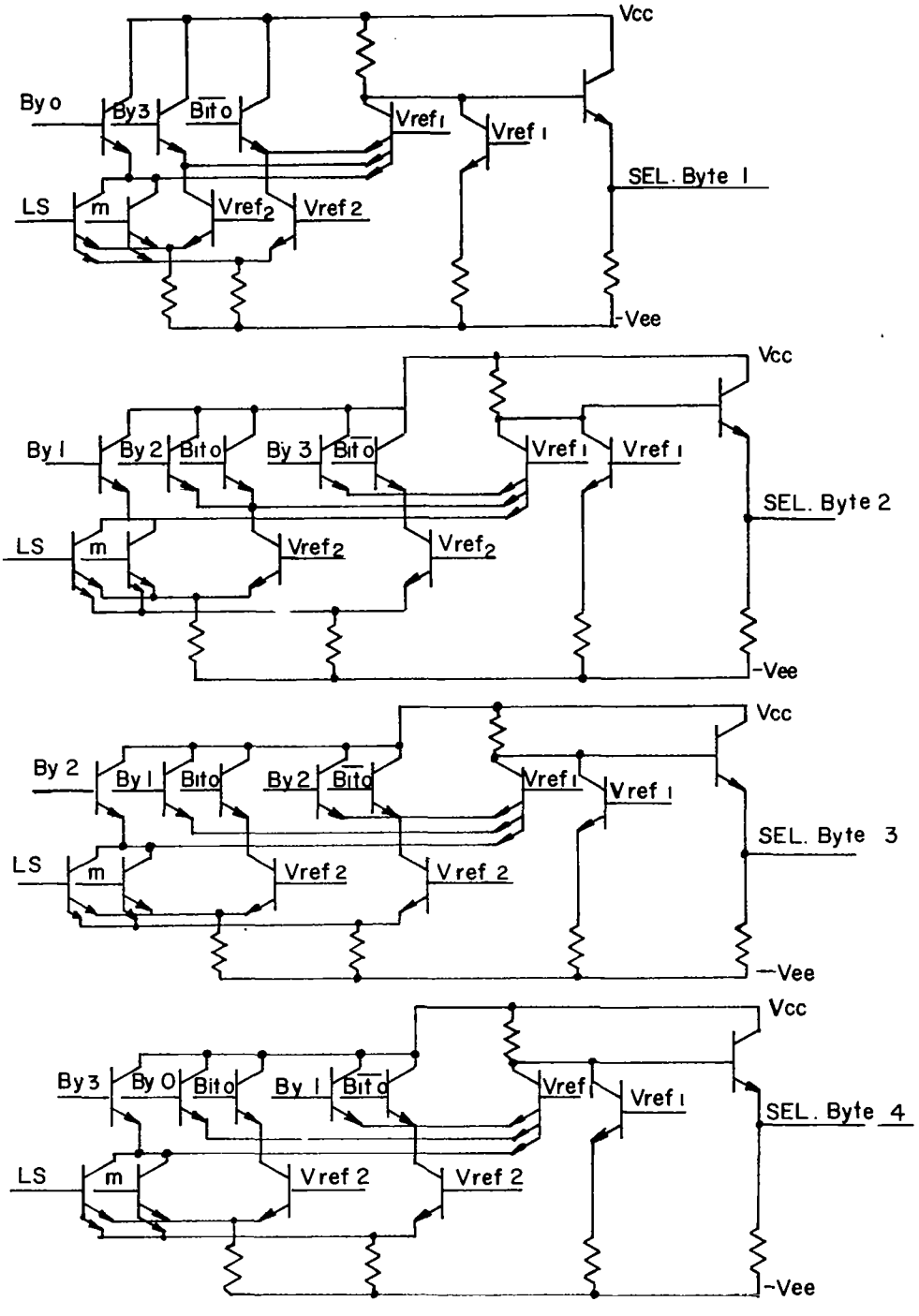
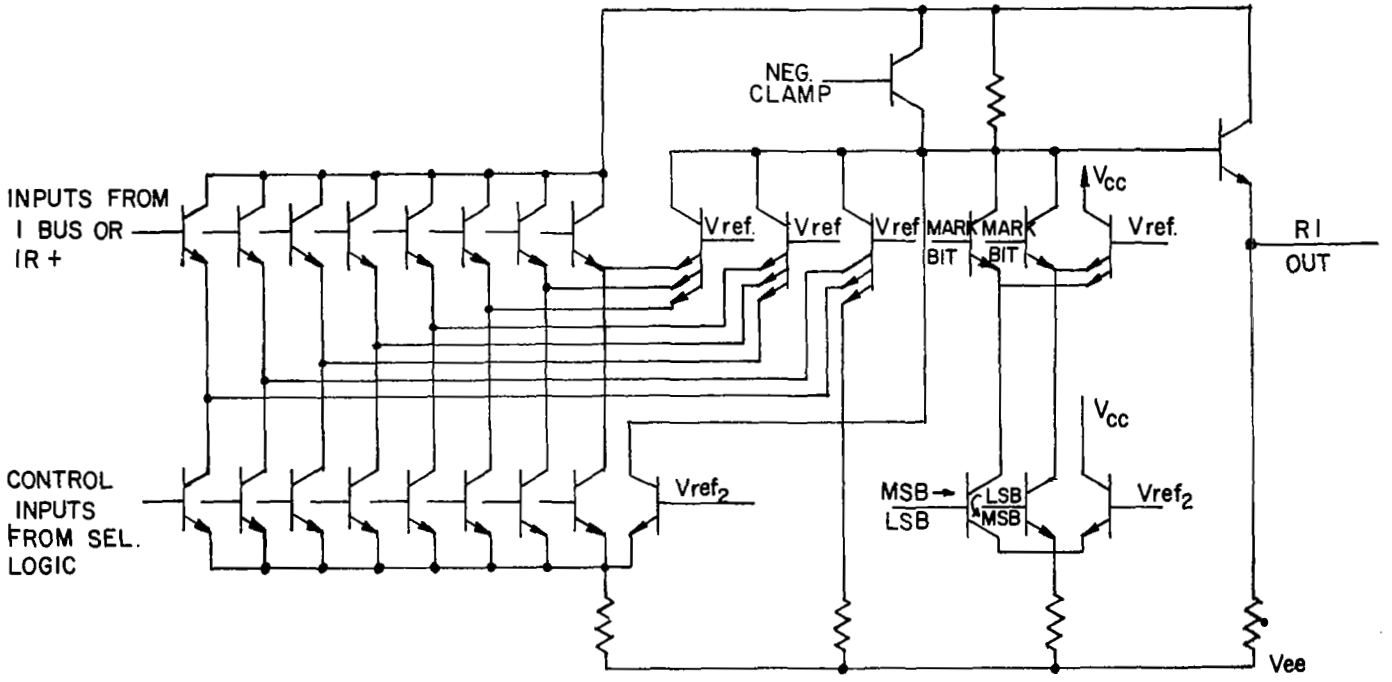


Fig. 4.2-7 BYTE SELECT



SWITCH USED FOR 1ST ROTATE LOGIC

Fig. 4.2-8

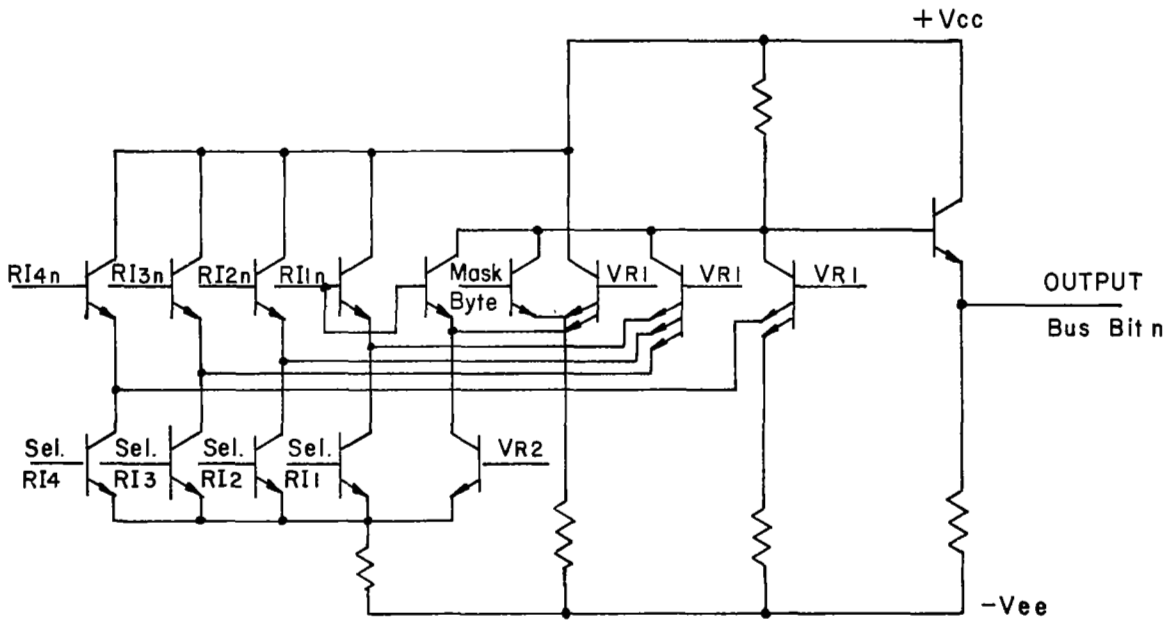
2nd Rotate (Figure 4.2-9).--This switch is used to select one out of four inputs, and is similar in operation to the switch used for the 1st rotate.

There are 32 inputs to this switch, RI, (output of 1st rotate), RI_2, RI_3 and RI_4, and each of these buses contain 8 bits. Five control lines are used in this switch, four select lines, and one mask line. The select lines determine which input bus is to be gated onto the output lines, and the mask input is used to blank out the entire byte (this input overrides the select inputs).

The L1 character must also gate the complement of the input bus (IB) to the output bus upon command. The only special provision for the complement function is made in the 2nd rotate block.

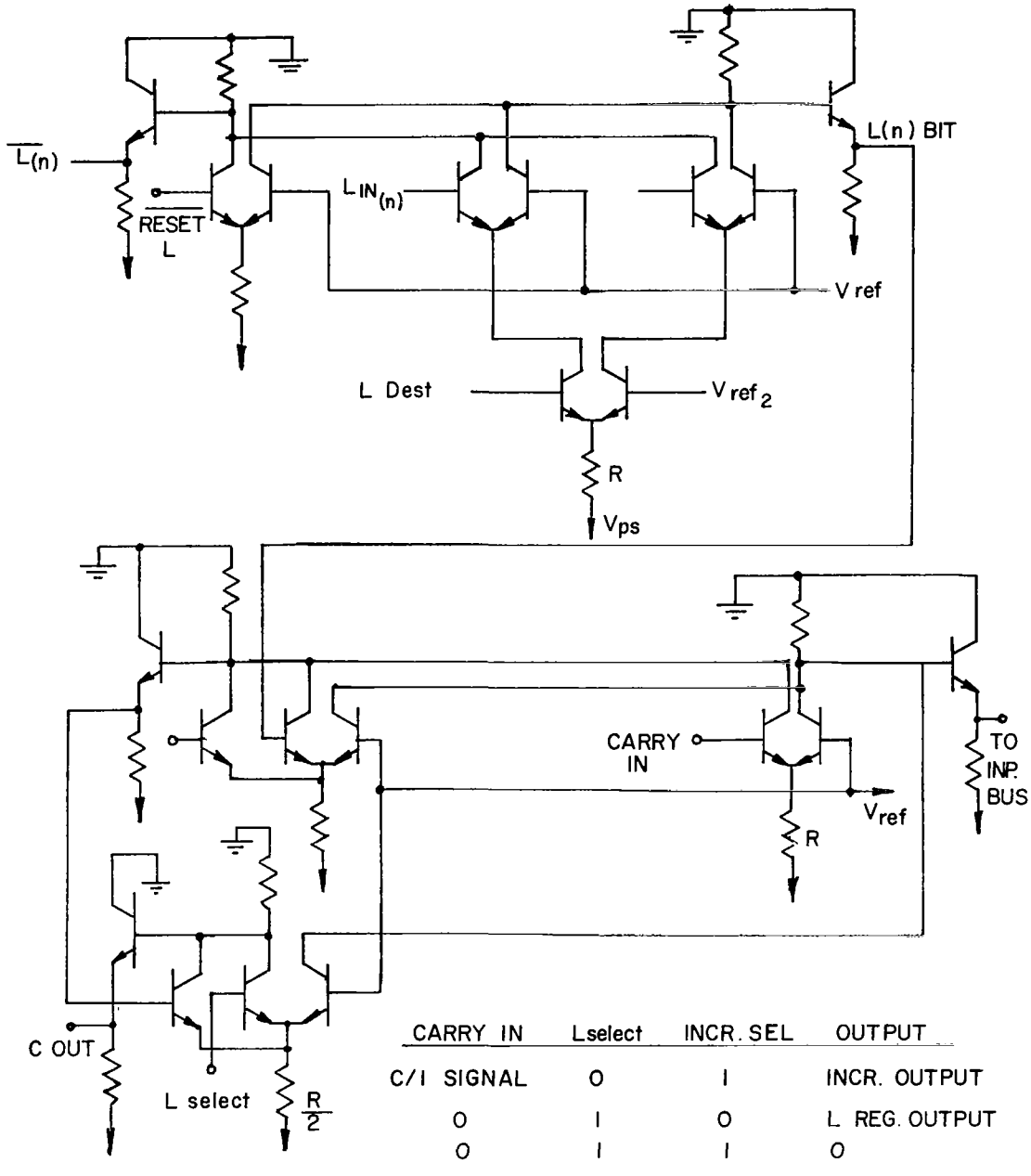
If a complement command (do $\rightarrow$ d4 = 0, LS = 0 and m = 0) is decoded, the select "0" line is "hi" in the 1st rotate block (this gates the IB into the 2nd rotate) and all of the mask generators and the byte select outputs are a logical "0". The 2nd rotate selects the RI_1 bus and the complement of this input is gated to the output bus.

L-Register and Incrementer.--This circuit uses a gateable flip-flop as described in the Appendix (the set function is not required) for the L Register and a modified threshold logic adder for the Incrementer. A register stage is shown in the top part of Figure 4.2-10 and an incrementer stage in the bottom half.



SWITCH USED FOR 2nd ROTATE

Fig. 4.2-9



CARRY IN	Lselect	INCR. SEL	OUTPUT
C/I SIGNAL	0	1	INCR. OUTPUT
0	1	0	L REG. OUTPUT
0	1	1	0

CIRCUIT FOR 1 BIT L-REGISTER AND INCREMENTER

Fig. 4.2-10.

The output is the "exclusive-or" function of the L bit and carry-in bit. A carry lookahead (and AND function of all preceeding L bits and carry-in) can be incorporated in the final design and is considered in the final circuit layout and comparison figures. However, with these circuits, advantage can be taken of their higher speed potential (by using higher power in the carry propagation chain) to eliminate the need for full look ahead.

Control transistors on the incrementer, OR'd with the L bit and the double weighted switch, will cause the output lead to represent 0, the L register state or the incremented output. (Note the slight difference in coding in this design for controlling the output of this unit.) This is an example of the flexibility of this modular switching approach-incrementing and selection is done within a cluster of modules not much more complex than a single gate, as exemplified in Section 3.

Figure 4.2-11 gives the logic schematic for the L-Register and Incrementer.

Table II summarizes the threshold logic implementation of the L1. The delay through L1 is 30 or 45 nano-seconds depending on whether an L Register and Incrementer cycle is included. The TTL comparison figures for the L1 are not known on a functional basis. However, a total NAND gate count of 300 has been estimated. This leads to the performance for TTL shown on the bottom of the table.

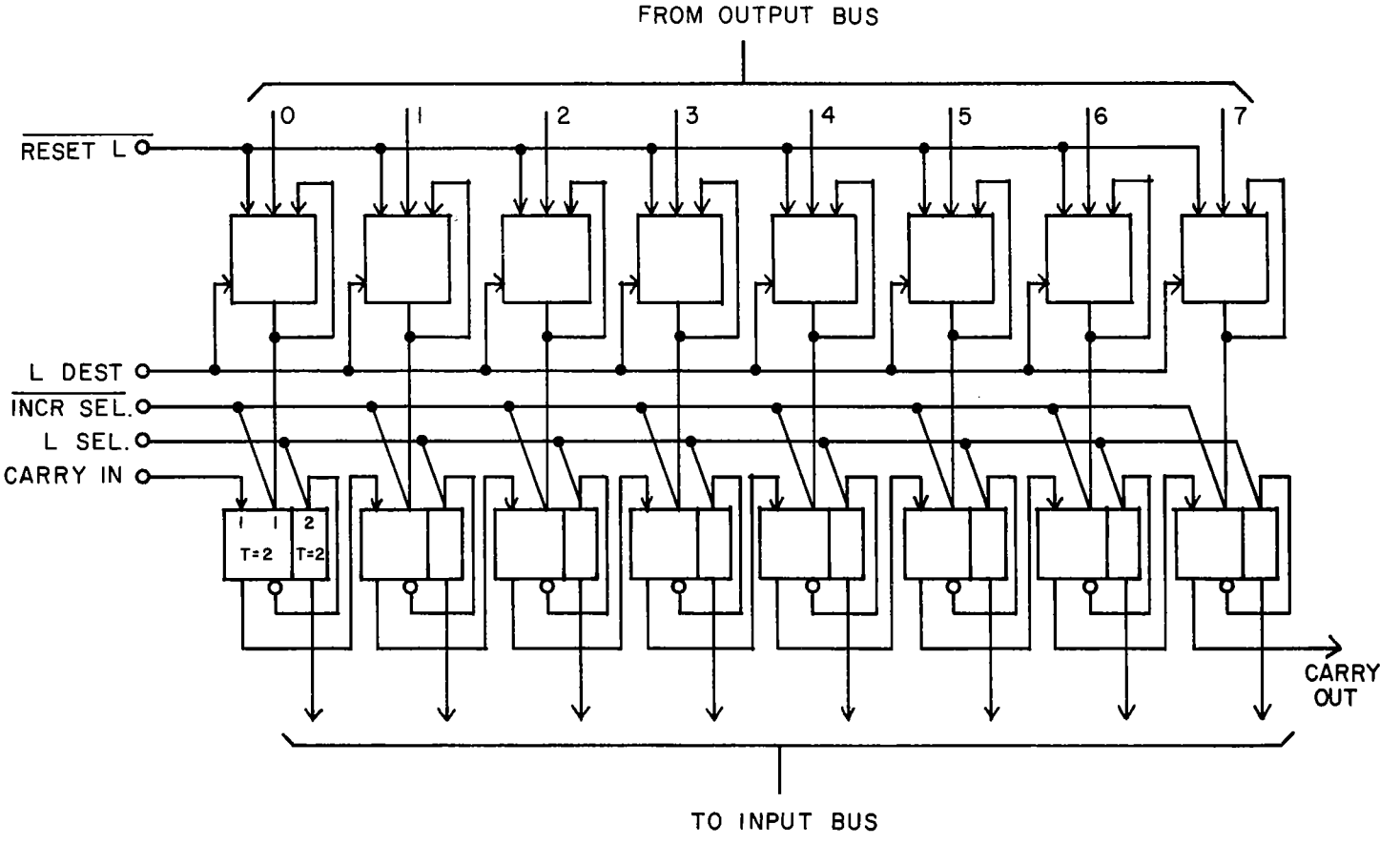


Fig. 4.2-11

LOGIC FOR L-REGISTER AND INCREMENTER

L1 CHARACTER

	Number of Components	Number of Connections	Avg. Delay nano-sec.		Avg. Power mW	Avg. Power x Speed Product 10^{12} watt-sec.
Bit Decoder	91	12	A*	7.5	220	1650
Byte Decoder	32	4	A*	7.5	80	600
C Decoder	32	4	A*	7.5	80	600
1st Rot. Sel.	86	30	B*	7.5	160	1200
Bit Mask	20	6	B*	7.5	40	300
Byte Mask	37	17	B*	7.5	52	390
Byte Sel.	70	26	B*	7.5	112	840
1st Rot.	264	160	C	7.5	224	1680
2nd Rot.	160	72	D	7.5	224	1680
L Reg. and Incr.	323	64	E	15	628	9420
Total T/L	1133	409	*30 or 45		1820	54,600 or 81,800
Total TTL (Approx. 300 gates)	2100	-	*60 or 90		3000	180,000 or 270,000

Note: (*x) in Avg. Delay Column indicate parallel operation

(total delay is Ripple thru delay time of L1 character.)

TABLE II

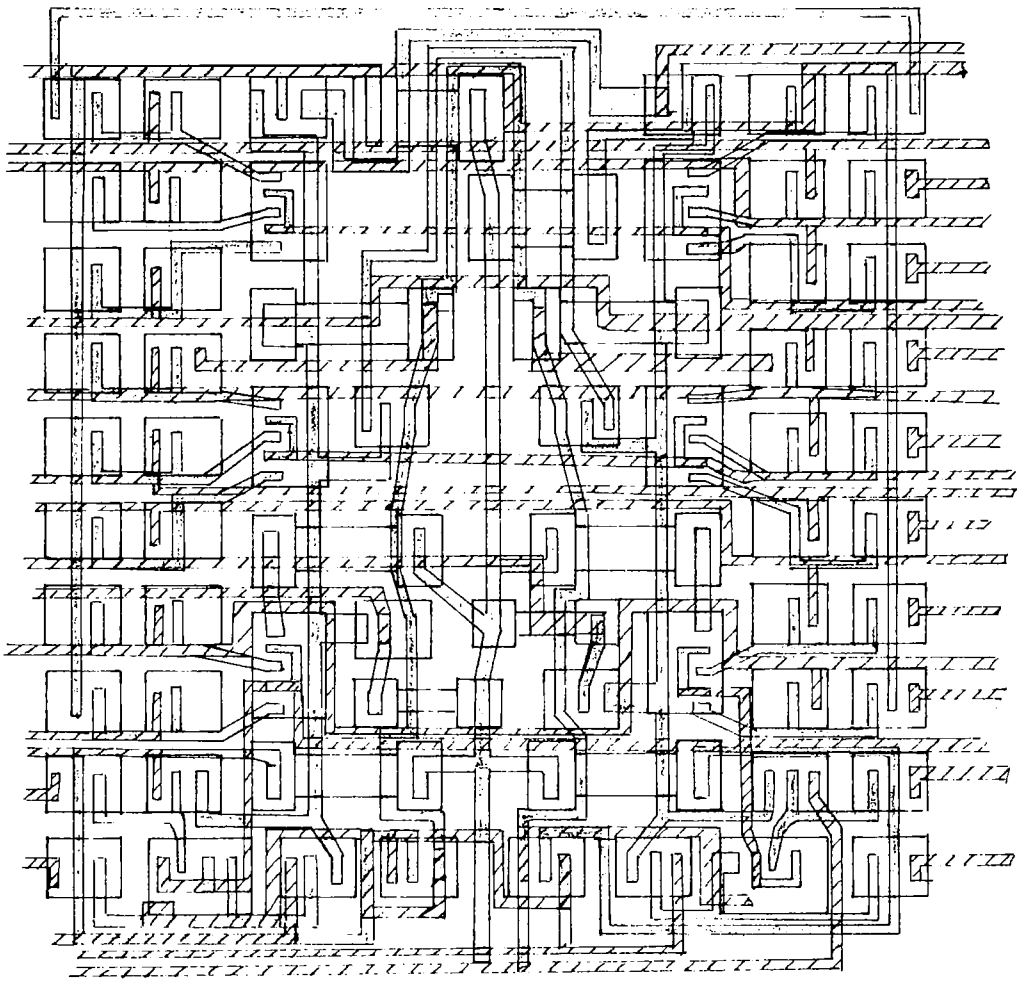
Circuit Layout for L1.--The fabrication assumptions given for the Parity Circuit Layout (Section 3.1) will also apply for the L1 character. Layout work was accomplished for the general logic portions (1st Rotate, 2nd Rotate and L Register and Incrementer but not for the decode functions). Figure 4.2-12 is the first rotate: (both layers of metal are on layout), and Figure 4.2-13 is the L Register and Incrementer. The rotate blocks each contain two switches, and the first level of metal is used for all interconnections of the two switches (power, references, ground and gate interconnections). The second level of metal is used for all I/O lines and interconnections between different cells. Both levels of metal will be fairly populated due to the large number of I/O lines and and also due to the fact that there are many common lines (line select, I.R. and input bus).

The register and incrementer is shown for one stage only and the two levels of wiring are all shown on one view.

The area requirements for the various blocks are given in the chart below:

Logic Block	Area Required Components (per 2 Bits)	Total Area With Interconnection (per 2 Bits)
1st Rotate	30 x 30 mils	40 x 40 mils
2nd Rotate	30 x 20 mils	35 x 25 mils
L Register and Incrementer	30 x 20 mils	30 x 20 mils

For the complete L1, these figures must be multiplied by four.

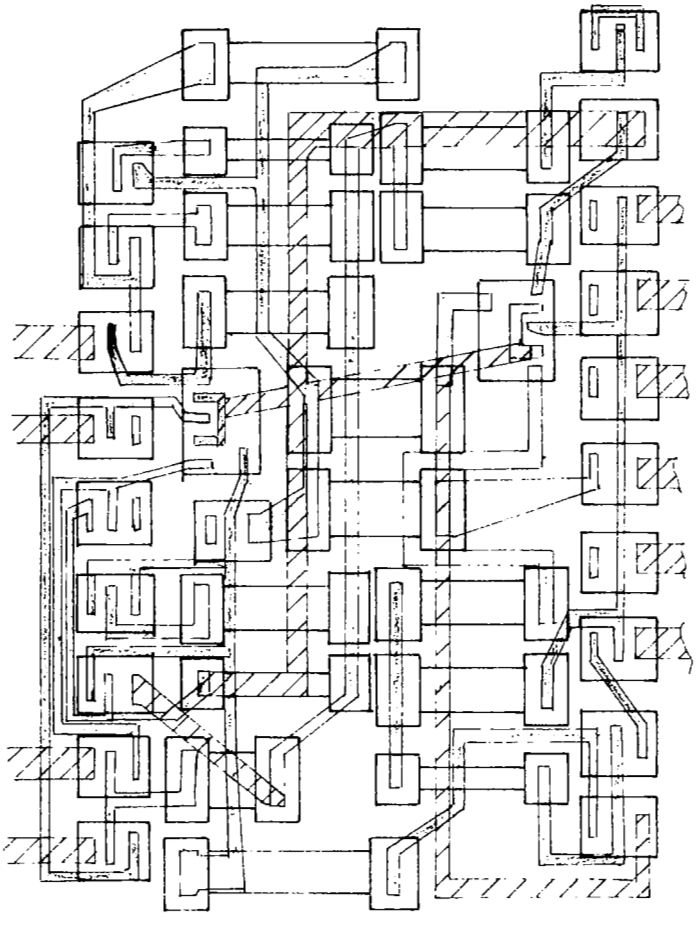


NOTE: 2ND LEVEL WIRING INDICATED BY SLASHED LINES

5 Mils

COMPONENT LAYOUT & METALLIZATION FOR 1ST ROTATE

Fig. 4.2-12



NOTE: 2ND LEVEL WIRING INDICATED BY SLASHED LINES

5 Mils

L REGISTER & INCREMENTER

Fig. 4.2-13

4.3 CAU Logic

Three circuits, accounting for the bulk of the CAU, were designed with threshold logic. These are given in the following sections.

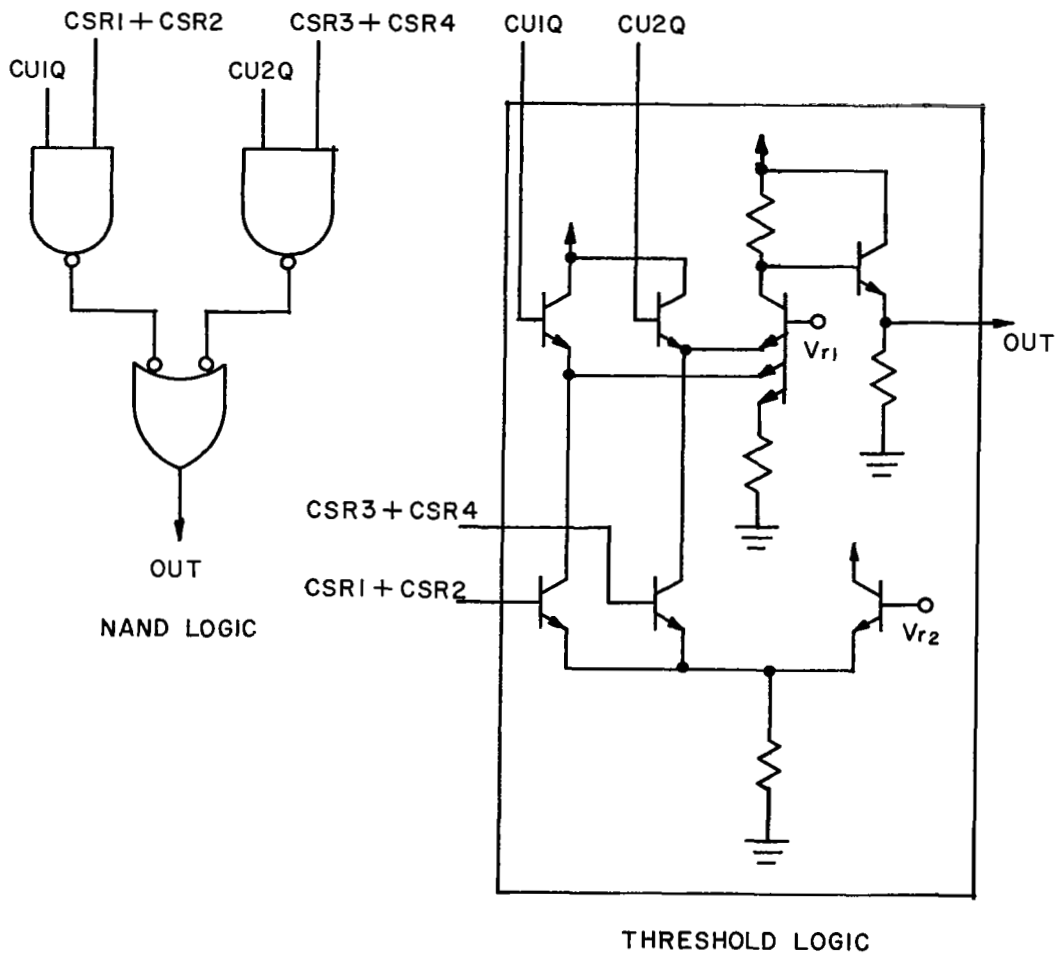
One of the most important circuit functions in the CAU is the switching circuit which interconnect the computer modules. Redundancy of switches (as well as the whole CAU) is not possible in the sense that it is in the computer modules themselves. Hence, the CAU deserves increased reliability considerations.

CAU Switch Logic.--Figure 4.3-1 shows the NAND gate and threshold gate realizations of a 2-input switch. The NAND gate design is from the actual MCB. (In actuality, 3-input switches will be used.) Two control signals determine which one of the two module inputs (in this case CU1Q, CU2Q) are connected to the output bus.

The data summarizing the performance of the two approaches is given in Table III.

Search Mode Logic (SML).--Figure 4.3-2 shows a bit path of the SML, again as taken from MCB logic, in both NAND and threshold logic implementations. This function involves the capability of gating any one of four inputs into a storage register and then into the Configuration Selection Register (CSR). The comparison for this function is given in Table III.

Parallel Up-Down Counter.--Another major portion of the CAU is the idle-time counter formed by P2 characters. These are up-down counters.



CAU SWITCH

Fig. 4.3-1

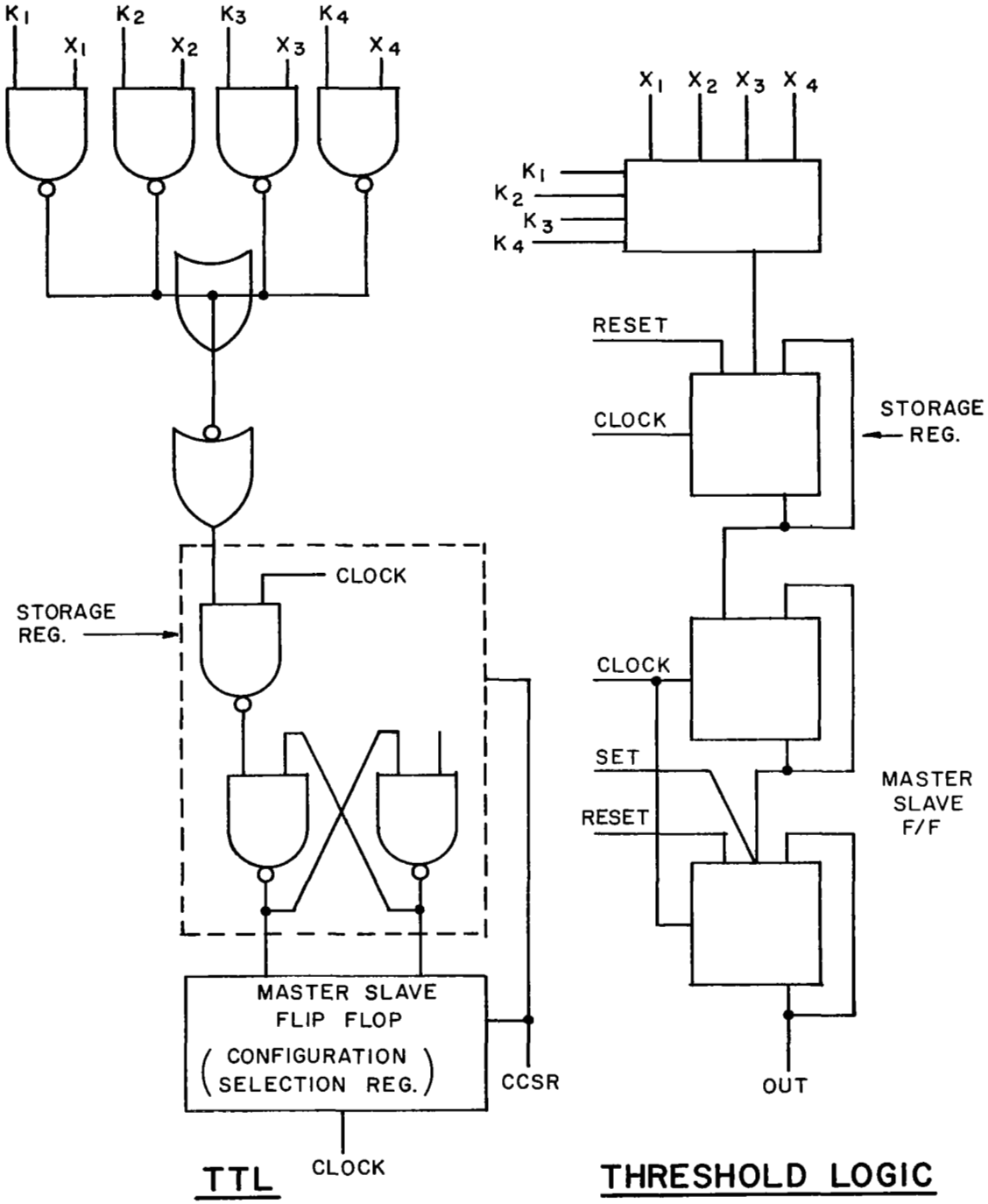

K1
K2
K3
K4
X1
X2
X3
X4
STORAGE REG.
CLOCK
MASTER SLAVE FLIP FLOP
(CONFIGURATION SELECTION REG.)
CCSR
CLOCK
TTL
CAU LOGIC
RESET
CLOCK
STORAGE REG.
SET
MASTER SLAVE F/F
OUT
THRESHOLD LOGIC

Fig. 4.3-2

A preliminary design of a 8-stage counter, such as the one used in the P2 was performed. There was no provision for decoding any control inputs since this was not part of the original tasks and detailed information was not available. The only design implemented was a parallel counter which was capable of counting forward or reverse and contained byte look-ahead logic. This design was carried out due to the extensive use of this counter in the CAU.

The basic flip-flop is the one described in the Appendix, the $\div 2$ circuit, with the addition of a set and a reset input applied to the slave unit (Figure A-3). All necessary bias circuits are considered in the design. The counter is comprised of 8 individual $\div 2$ circuits with a common clock line for the slave, while the clock for the master is gated externally to the flip-flop. In order for the master clock to be generated, one of the following conditions must exist:

1. The count up command is generated and all previous flip-flop outputs are in the "1" state, or

2. The countdown command is generated and all previous flip-flop outputs are in the "0" state.

During the period when the master clock is inhibited, the element is held in the storage mode ($Cl < Vref_2$) and therefore the output of the slave will remain the same, regardless of the state of the clock since the output is now the same as the input. Most of the inhibit logic is comprised

of "AND-OR" gates which increase in complexity as it progresses to each higher order flip-flop.

The number of components, power, etc. is listed in Table III.

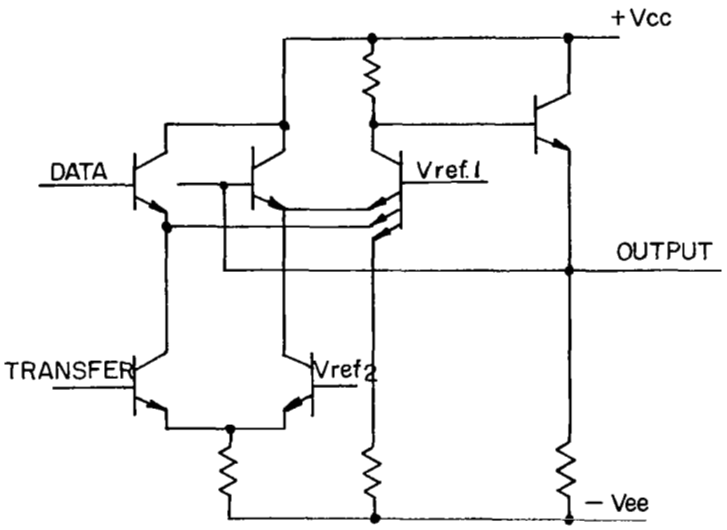
4.4 L2 Character - Arithmetic Unit

A preliminary design of the L2 character was performed using available information on its function. The designs include arithmetic logic, registers and bussing.

Although the L2 was not included in the original tasks, it is shown here since many of its circuits have been designed as part of the other functions and since it seems particularly adaptable to threshold logic.

The circuit for the register (Figure 4.4-1) is the storage element described in previous sections and contains one data input, one output and a transfer line that would be common to the entire register. There is no provision for direct set or reset. A total of 16 stages are required for the L2 character.

The basic requirement for the adder is: full addition, mod 2 addition, transfer of the A register and an initial carry input on the lowest order bit. The full adder circuit used is a modified parity generator;



STORAGE REGISTER USED FOR L2 CHARACTER

Fig. 4.4-1

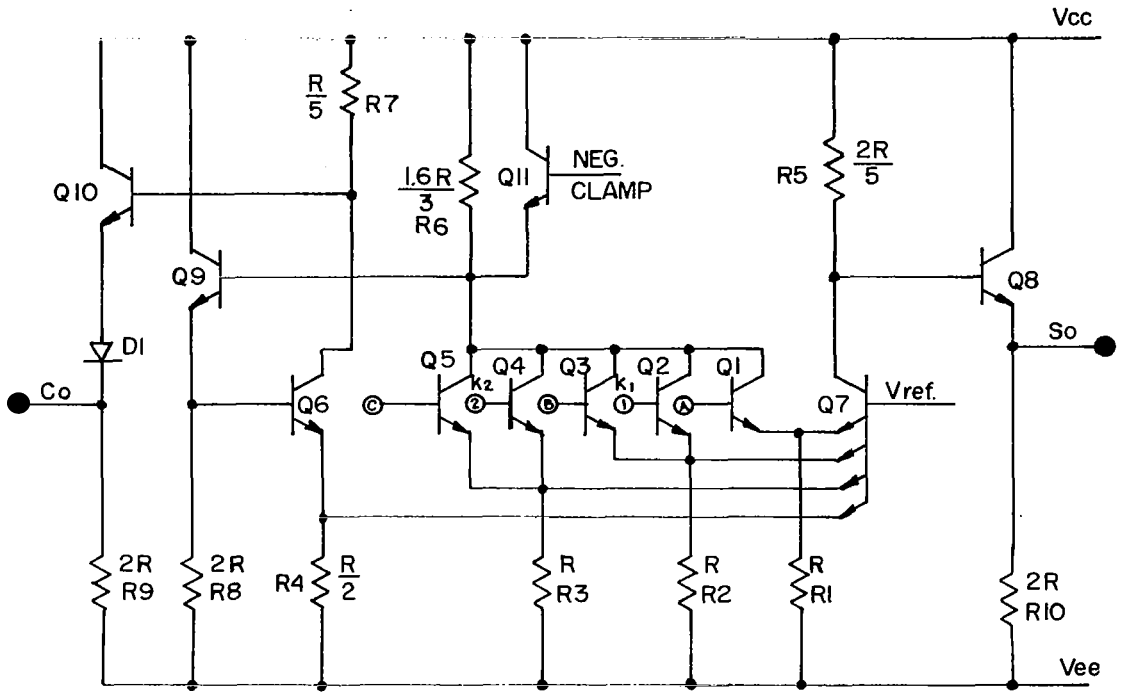
CAU LOGIC

	Number of Components	Number of Connections	Avg. Delay nano-sec.	Avg. Power (mW)	Avg. Power x Speed Product 10^{12} (watt-sec.)
		CU-MU-SWITCH (1 Switch)			
TTL	28	6	26	46.5	1210
Threshold Logic	11	4	7.7	20	154
		SEARCH MODE LOGIC (1 Bit Path)			
TTL	126	35	60	140	8400
Threshold Logic	57	20	22.5	104	2340
		UP-DOWN COUNTER (8 Stages)			
Threshold Logic	216	32	8.5/stage	52/stage	442/stage
		INHIBIT LOGIC (Estimated)			
Threshold Logic	200	100	7.5		

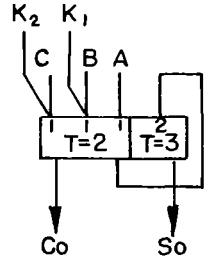
TABLE III

a carry output signal was generated and two control inputs were added (Figure 4.4-2). The operation of the adder is similar to the parity circuit (excluding the control input for the present). If either 0 or 1 of the inputs are "hi" (A, B or C_{in}), Q6 is on (its base being more positive than Vref) and the voltage drop across R5 plus the V_{be} drop of Q10 and D1 is sufficient to cause Cout to be "lo" (more negative than Vref). If 2 or more inputs are hi, then Q6 is off and Cout is now 2 V_{be} drops below gnd (-1.5 V or at a "hi" level). This is equal to the carry out function or AB+AC+BC. The Sout is the same as it is for the parity circuit. The two control inputs, K1 and K2, are OR-ing inputs with the B and C_{in} inputs, respectively, and are used to control the operation of the adder as shown below.

Add	K1 and K2 held lo-normal operation
Sub	K1 and K2 lo, initial carry in = "1" and $\overline{B}$ gated into the L2 character-normal operation
$\overline{\text{EXCLUSIVE OR}}$	K1 = 0, K2 = 1; this control overrides the C_{in} input on all bits and the adder performs the $\overline{\text{excl. or}}$ between A and B. If $\overline{A}$ can be gated into the L2 character than the exclusive-or function can be performed
TRANSFER A	K1 - K2 ≠ "1" B and C_{in} inputs and the output of the adder is now determined by the information in the "A" register



FUNCTION	CONTROL		INIT. CARRY IN
	K_1	K_2	
ADD	0	0	0
SUB.	0	0	1
EXCL. OR	0	1	X
TRAN. "A"	1	1	X



LOGIC SYMBOL

TOTAL N° OF COMPONENTS = 23

FULL ADDER

Fig. 4.4-2

The complete adder is shown in Figure 4.4-3, and it includes four threshold gates to perform internal carry-look ahead. The propagation delay between C_{in} to C8 is approximately 30 nano-seconds, C_{in} to S8 is approximately 52 nano-seconds and the delay for 32 bits (C_{in} -- S32) would be on the order of 142 ns. If the delay times are excessive, then the carry lookahead can be modified to perform byte parallel lookahead.

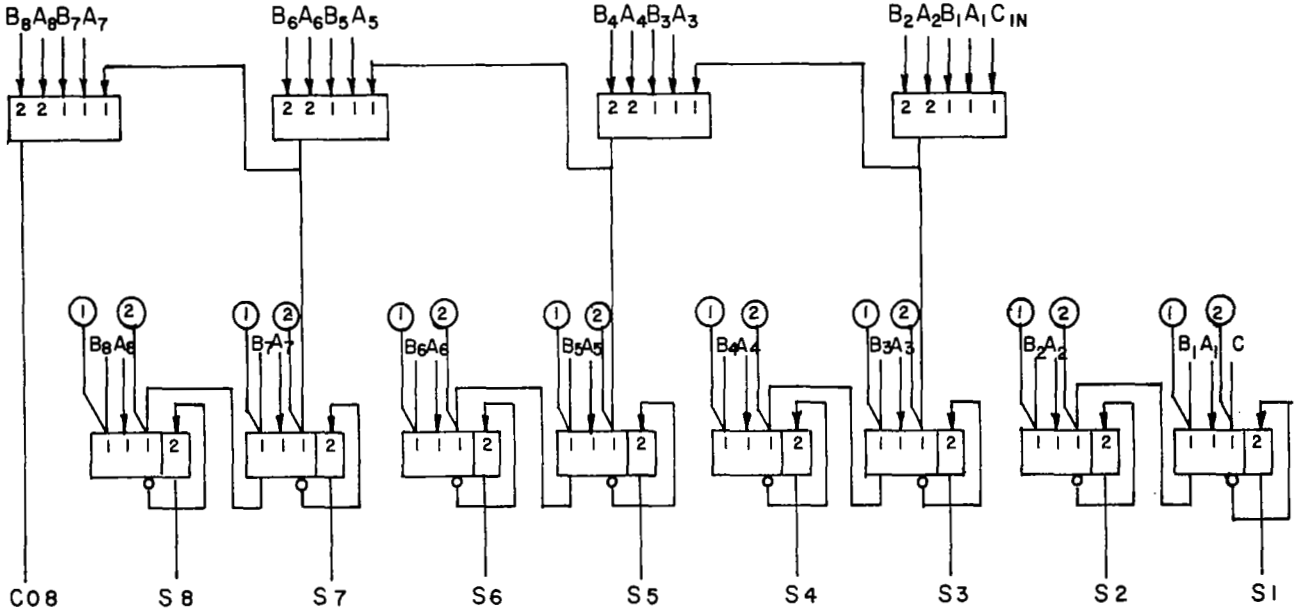
A complete breakdown on the number of components, speed, etc., is given in Table IV.

5. CONCLUSIONS

The efficiency of logic to perform a given function can be measured by circuit complexity and energy (power x speed)* necessary to calculate the function. In LSI, circuit complexity relates to total component count, the component geometries and their connectivity. These, in turn, dictate the all important chip area and metallization complexity. Less second layer metal means less to go wrong.

It has been shown that the use of threshold logic provides benefits in all areas. The threshold circuits were composed of non-saturating current switches.

*Frequently, it is one or the other of these which is the determining factor for specified performance.



8 STAGE ADDER

Fig. 4.4-3

L2 CHARACTER

Number of Components	Number of Connections	Avg. Delay nano-sec.	Avg. Power (mW)	Avg. Power x Speed Product 10^{-14} (watt-sec.)
160		REGISTERS (16)		
160	32	7.5	20/stage 320 Total	150/stage
		ADDER		
236	60	$C_{in} \rightarrow C_{08}$=30	720	
		$C_{in} \rightarrow C_{08}$=52.5		37,800/byte
		$C_{in} \rightarrow S_{32}$=142		410,400 for 4 byte

TABLE IV

This was done by doing detailed logic and circuit designs on selected, representative portions of the Modular Computer System, and comparing the results, most of the cases, with saturated NAND gate designs. Comparisons with conventional current mode (emitter coupled logic) OR/NOR gates would show still greater advantage in favor of threshold logic techniques.

Some of the NAND gate logic design characteristics were conjectured and will depend on their ultimate implementation in LSI. (Table V gives the comparison data, where available, summarized from Tables I-III.

o In no case were any of the designs worse in any respect to a TTL design.

o It appears that for functions such as decoding, minimum advantage is obtained. This is expected with threshold logic.

o For the bulk of functions in this computer (switching, parallel shift and rotate, counting) component count is reduced by a factor of 2 to 2.5. These functions have traditionally been weaker spots for threshold logic, but some new techniques have given these advantages.

o As expected symmetrical type functions (parity circuits, arithmetic functions) provide the greatest component savings - a factor of 3 or more.

o Connection complexity is substantially reduced. In those cases where computer partitioning has resulted in the use of characters with

a large number of I/O connections and only one or two levels of logic, the savings in connections is primarily among the components comprising a gate. In other cases, it shows on the inter-gate level (parity circuit).

o One of the most surprising results is the fact that higher speeds are possible with lower power for everything studied.

o These circuits work at their own, rather unique, internal signal levels. However, shifting to and from saturated logic levels is easy and would not appreciably alter savings offered by threshold logic.

o The implications of 2 or 3 to 1 savings in complexity is important if the function on a 10,000 sq. mil chip had to be realized on a 20,000 tp 30,000 sq. mil chip. The technology is such as to always favor a small monolithic die.

o Testing of threshold vs. NAND gates, although somewhat hypothetical at this time appears to be easier with the threshold techniques for two reasons. First, there is less to test and second,having control of a reference voltage provides a degree of freedom in testing and diagnosis.

Any advances in the technology - smaller, faster transistors, smaller resistor geometries for given accuracy, etc. - can be utilized by the threshold logic as well as any other logic. Standard integration fabrication processes can be used (gold doping is not required).

In custom integrated designs, circuit "tricks" are frequently used to simplify a circuit with little or no effect on the outside world. This is particularly true in MOS logic arrays and in memory arrays (bi-polar

and MOS). These "tricks" are defined as a variation from the use of pure Boolean gates to realize a function or the use of a component (or more) in combination with a conventional gate to perform a bigger function than the gate can do by itself. The techniques of this study can be viewed as an extension of this idea to a much larger degree in combination with classic threshold logic theory. It is the application of this theory which allows the systematic synthesis of functions; variations appear for further circuit simplification.

It seems that with the application of threshold logic, the performance of bi-polar technology could significantly be increased to aid in the solution of some LSI problems.

5.1 Implication

A savings in component count (and hence chip area) can affect the design to varying degrees. For example, assume that chips can be fabricated with high reliability in sizes only up to 120 x 120 mils. If a T/L design used 80 x 80 mils and a TTL design the full 120 x 120 mils, the improvement will result in better yields simplicity and less power. This is all to the good and would represent an improvement. If, however, a T/L design used 120 x 120 mils and TTL then needed 2 chips, the improvement is of a much higher order. We are now decreasing external connections - a more important reliability factor than a decrease in area or 2nd layer metal connectivity.

Also, the function capacity per chip will affect the logic character partitioning.

Threshold logic per se for the computer functions studied is used in its pure form in the parity circuits and incrementer of the L1. It would also be used to tremendous advantage in L2-the arithmetic character. In most other cases, the gates have their thresholds set for the AND function so that basic OR-AND logic is done. The current switch modules of the gates are used for flip-flops (as in the registers and counters) digital multiplexing and switching (as in the combinational Rotate and Shift functions.)

Switching functions which have previously been weak areas for threshold logic have been realized with compatible circuit techniques and with comparable component savings and efficiency over conventional methods. This is viewed as a breakthrough in overall LSI design.

5.2 Comparison With MOS

It is acknowledged that due to the smaller geometries of MOS transistors and the use of MOS devices in place of resistors, that the packing density of MOS arrays is greater than for bi-polar arrays. However, when considering the improvements in bi-polar device and circuit techniques (threshold logic) this difference is becoming smaller. As a matter of fact there is conceivably no difference in some of the functions studied. Also, it appears that packages may be pin limited due to a combination of partitioning and physical factors so that improved packing density, after a while, has little importance.

Whereas, the bi-polar designs shown can work at clock rates of 10 MHz

to 25 MHz or higher the MOS speed would be a maximum of about 5 MHz although improvements can be expected in this area.

Finally, G. R. Madland, President Integrated Circuit Eng. Corp. emphasized the reliability of bi-polar IC's in "MOS Integrated Circuit-The Designer's Dilemma," WESCON, August 1969 in a comparison between the two technologies. Although there are more processing steps, these steps are not generally as critical as those in MOS and they are, by now, well proven.

Power.--Many of the functions shown with bi-polar threshold logic do not take appreciably more power than MOS circuits would if they had to work at the required speeds; in fact, it might take less. This includes C/MOS! The dynamic power for a C/MOS gate is $P_d = CV^2f$. (The static power may be neglected for C/MOS but not necessarily for P/MOS). As an example of this assume:

$C = 10$ pf

$V = 10$ V

$f = 5$ MHz

then $P = 5$ mW

For the L1 character, the First Rotate and Second Rotate functions would require about 70 gates each; hence, power per rotate function could reach 359 mW.

The power with the threshold gate switching circuits was 224 mW at a

delay of about 7.5 nano-seconds. This power is fairly constant, independent of logic states. The MOS power, of course, depends on how many gates switch each cycle. In the rotate logic this can vary from 1/8 to 7/8 of the maximum. For estimating purposes, assume a 33% duty cycle per gate. Then the bi-polar power would be no more than twice than C/MOS power. This power difference would be less for the arithmetic logic and parity circuits. The C/MOS would show much higher power savings in the latter stages of the Idle Time Counter and in the decoding functions. The point is, that for these speeds, MOS circuits should not be chosen for power savings alone.

The threshold circuits power when compared to TTL has already been shown in some of the estimates. The L1, TTL design, has not been detailed, but overall power savings should be about 2 to 1.

SUMMARY TABLE

	No. of Components	No. of Connections	Avg. Delay (nano-sec.)	Avg. Power (mW)	Chip Area (sq. mils)
			8-BIT PARITY		
Threshold Logic	79	11	32	248	1500
TTL	210-280	61	87	366	-
			L1 CHARACTER		
Threshold Logic	1133	-	30-45	1820	-
TTL	2100	-	60-90	3000	-
			CAU SWITCH		
Threshold Logic	11	4	7.7	20	
TTL	28	6	26	46	
			CAU SML		
Threshold Logic	57	20	22.5	104	
TTL	126	35	60	140	

TABLE V

REFERENCES

1. "Integrated Threshold Logic," RCA Final Report AFAL-TR-69-195, August 1969.

2. "Research in the Effective Implementation of Guidance Computers with Large Scale Arrays," Second Interim Report to ERC, NASA, July 1969, Hughes Aircraft Co., J. J. Pariser, et al.

3. "Threshold Gate Building Block," S. Cohen and R. O. Winder, IEEE Transations on Computers, Vol. C-18, Number 9, September 1969.

APPENDIX

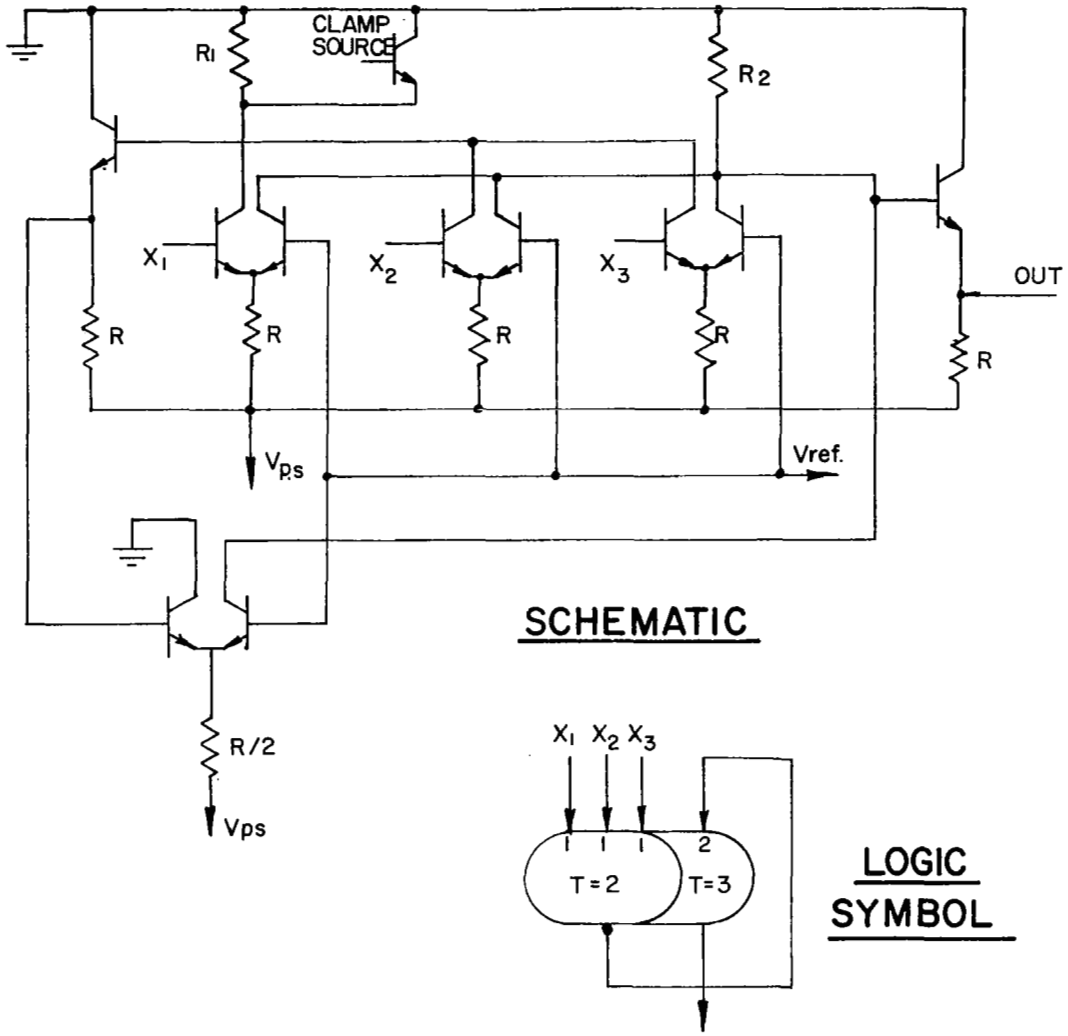
A-1 Improved Threshold Gate Circuit

Frequently, one threshold gate feeds a second threshold gate where the second gate has (among others) the same set of inputs as the first gate. This is true in parity circuits and adders. To simplify the threshold circuit realization of such functions, the complemented side sum resistor is used for the first gate's function while the true side sum resistor is used for the second gate's function. This latter resistor then sums the first gate and one or more modules representing the additional inputs to the second gate.

A schematic illustrating this for a 3-input parity function is shown in Figure A-1, with its logic symbol.

The Resistor R1 calculates the inverted majority function of the 3 inputs and feeds a double weighted module. This module then feeds R2 which is also fed by the 3 inputs. The value of R2 is such as to calculate the 3/5 majority function. If 2 or 3 inputs are hi, the bottom decision switch contributes a double current through R2; otherwise, it contributes zero current. Therefore:

# of Inputs Hi	Unit of Current thru R2	Output
0	3	lo
1	2	hi
2	3	lo
3	2	hi



IMPROVED THRESHOLD LOGIC PARITY CIRCUIT (3 BIT)

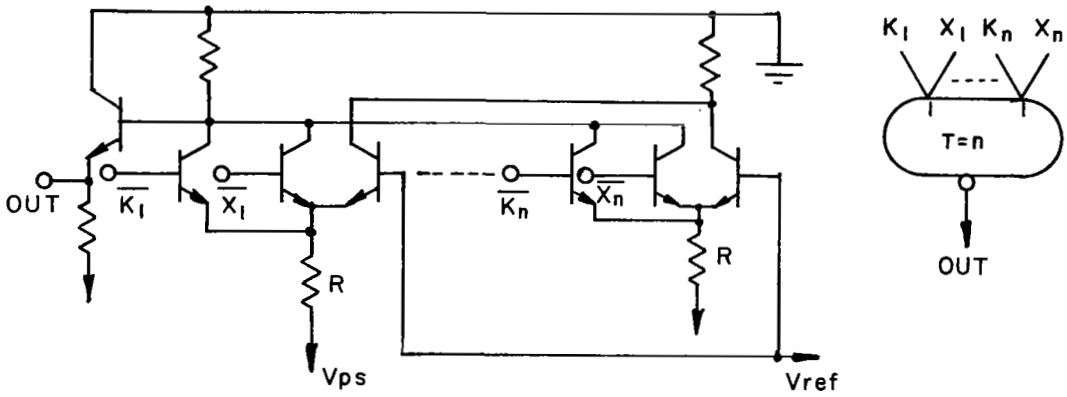
Fig. A-1

The output represents odd parity. The sum point varies between 2 and 3 hi and so doesn't require any clamp control. The 3-input majority side is clamped by a transistor so as to prevent the input transistors from saturating. The same temperature and power supply variation immunity apply to this circuit as to the original dual-threshold gate circuits for this function.

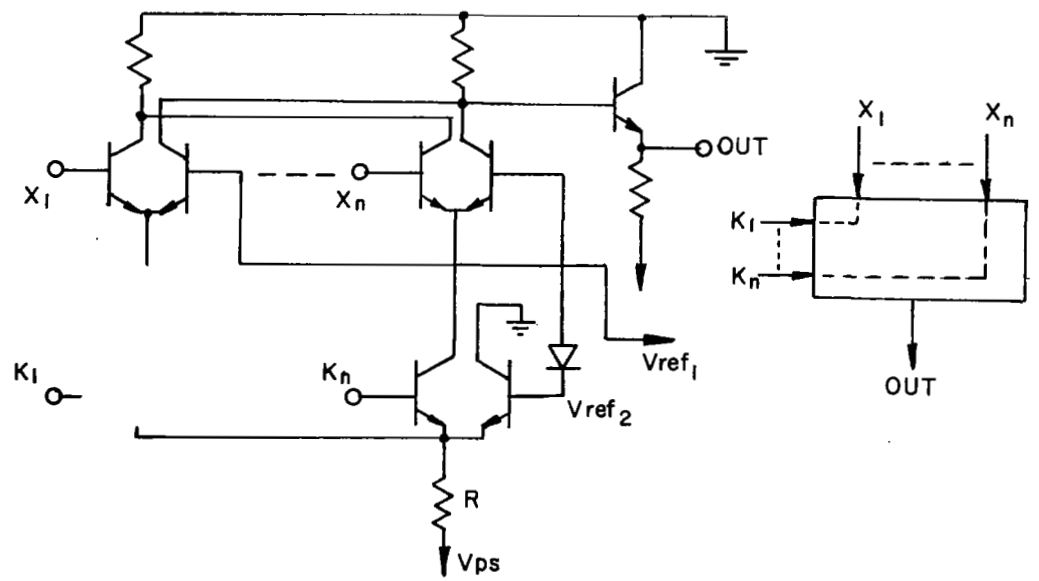
Three decision switches with associated power drain and connections have been eliminated compared to a conventional threshold logic design. This doubles the improvement (over an already 3 to 1 improvement, with straightforward threshold gates) in connections, components, etc., as brought out in Section 3.1 when compared to NAND gate realizations.

A-2 Switching Circuits (Figure A-2)

Up to now, switching circuits of the type shown in Figure a have used OR'ing inputs with the threshold set at its maximum so that OR-AND gates were realized. If all but one of the control signals, K_i, are lo, then the output is X_i. This circuit uses n units of power and required inverted inputs. In Figure b is shown an alternative circuit approach. The X inputs are brought into upper decision switches and the control inputs, K, are brought into lower decision swithces. Signals and the reference voltage brought into bottom switches are displaced by a diode or V_{be} drop from the levels of the upper switches. This circuit uses 1 unit of power, and does not require complemented inputs. It is realized with the modules used for threshold logic and is otherwise compatible with the gates reported on



(a) SWITCHING CIRCUIT



(b) IMPROVED SWITCHING CIRCUIT

Fig. A-2

previously. OR'ing inputs can be used for logic and control on upper or lower switches giving added flexibility. Saturation is avoided with simple clamp circuits as previously described and speeds are comparable to single level threshold circuits.

A-3 Flip-Flops With Switching Circuits

Figure A-3 shows a basic flip-flop (F/F) circuit and the logic for using it as a S-R gateable F/F or a master-slave F/F. In principle, when the clock (which is always fed to a lower level switch) is hi, control is exerted by the input - the output takes on the value of the input. When the clock goes lo, the output, which is fed back to one of the upper switches, controls, and so the circuit is in its storage phase. With the addition of OR'ing inputs, additional switches, logic can be done in conjunction with storage. Complementary outputs are available by simply adding a resistor and F/F on the inverted side. A master-slave can be used as a binary counter by feeding the inverted output of the slave back to the input of the master.

In master-slave circuits (used for shift registers as well as counting) the reference voltage for the master, $Vref_2$ is designed to be slightly different from that of the slave so that their are no specifications on clock rise or fall times.

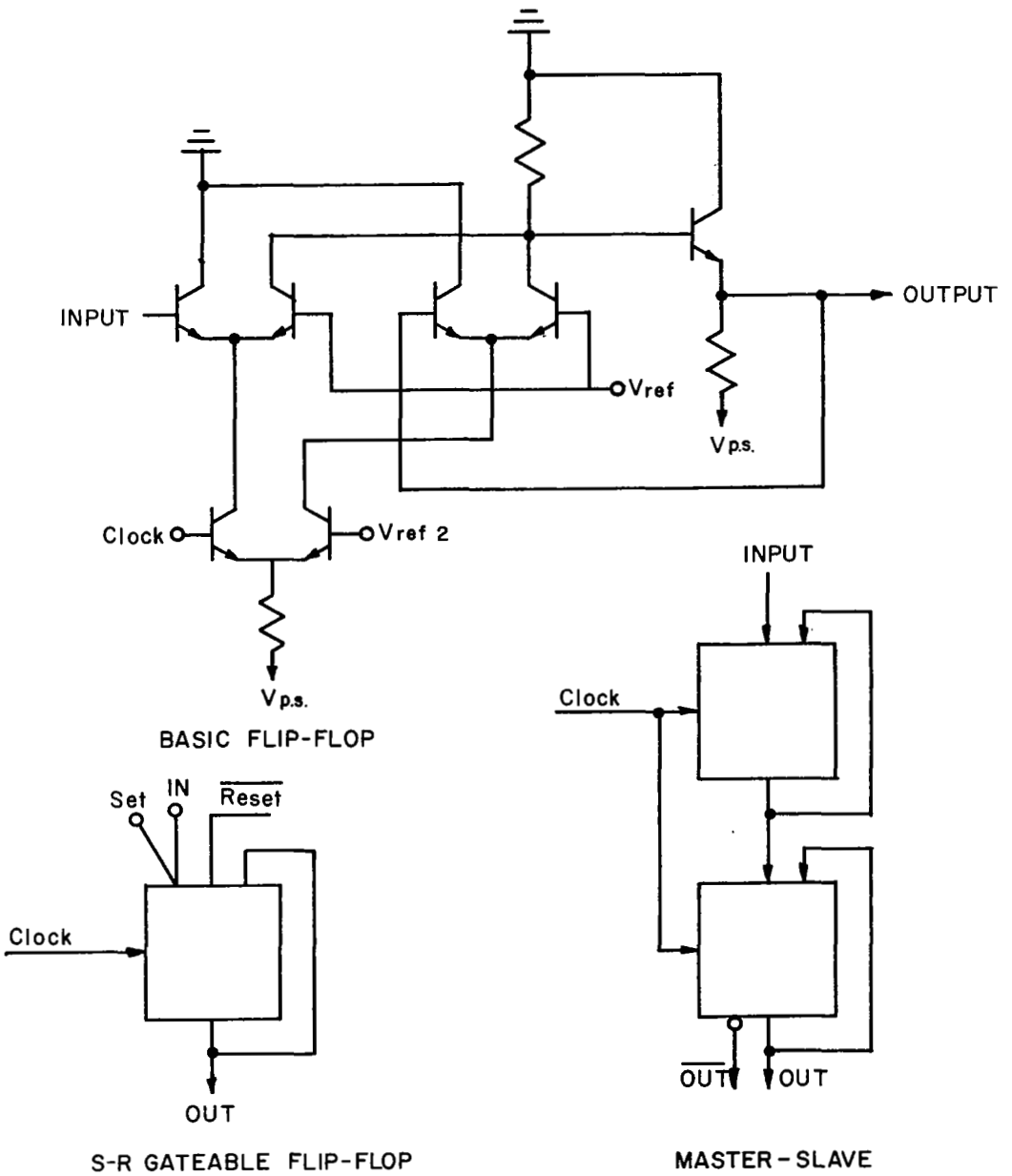


Fig. A-3

FLIP-FLOP CIRCUITS USING THRESHOLD SWITCHES

NASA-Langley, 1970 — 8 CR-1668