SUMMARY REPORT
IMPLEMENTATION INVESTIGATIONS
OF ZVA/FVA COMPRESSION ALGORITHMS

IMSC 669224

1 July 1965 through 28 February 1966

Contract NAS 8-5110, TP 2-831815

Prepared by: H. N. Massey
W. E. Smith

Date: 28 February 1966

FOREWORD

This report describes work performed by Lockheed Missiles & Space Company during the period between 1 July 1965 and 28 February 1966 under Phase V of Contract NAS 8-1150 entitled "Advanced Telemetry System with Adaptive Capability," for the George C. Marshall Space Flight Center, National Aeronautics and Space Administration, Huntsville, Alabama.

CONTENTS

CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS

Section 1
SUMMARY

The purpose of the study discussed in this report was to investigate design techniques for implementing specific compression algorithms. The work reported here is an extension of prior studies, References 4 and 14. Specifically, the goals of this investigation were:

1) To make a complete paper design of the "high speed" logic section of a ZVA/FVA Hybrid Data Compressor.

2) To evaluate the design and construct breadboards and test.

3) To investigate integrated circuit logic families for use in the low speed logic section of the ZVA/FVA Hybrid Data Compressor.

4) To investigate methods of implementing the reference memory with a storage capacity to meet the requirements of the Hybrid ZVA/FVA Data Compressor.

Much of the logic required to implement a Hybrid ZVA/FVA Data Compressor has been proven in the design of the ZFN Telemetry Data Compressor, Reference 15. In this study only the logic which had not previously been implemented was breadboarded. Because this additional logic operates at a faster clock rate, it is referred to in this report as the "high speed" logic section.

The breadboarded hybrid ZVA/FVA data compressor, hereafter referred to as the breadboard, successfully performed both the FVA and ZVA algorithms over high and low temperature extremes. Fairchild Diode Transistor Micrologic was used to perform the logic functions in the breadboard. These integrated circuits performed as specified by Fairchild and are recommended for use in future

flight units. Also, to evaluate alternate source possibilities, circuits manufactured by both Philco and Stewart Warner were used in the breadboard. While modules from Philco and Stewart Warner were not used as extensively as the ones manufactured by Fairchild, they also operated as specified and there was no interface problem between modules manufactured by the three companies.

The breadboard has a single channel capability to accept parallel inputs from an input signal simulator, perform either an FVA or ZVA algorithm implementation, and provide parallel outputs to a buffer memory. However, the breadboard does not include a buffer memory, and the outputs go instead to output monitoring circuitry. The breadboard data compressor, the input signal simulator and the output monitoring circuitry are all mounted in an aluminum carrying case.

A complete hybrid data compressor including reference and buffer memories and all associated logic would require approximately 650 logic modules. Over half the modules would be required to operate at a maximum clock rate of 288 KC, with the remainder operating with a clock rate of 2.8 MC. The Fairchild Diode Transistor Logic could be used in both the high speed area (2.8 MC) and the low speed area (288 KC). However, a survey was made of the integrated circuits available to determine if a logic line compatible with the Fairchild DTμL line was available which could perform the low speed functions, but would require much less power. A Fairchild low power DTL family was selected, because it consumes significantly less power than other logic families.

The reference memory in the hybrid data compressor is required to have a minimum of 37 bits of DRO storage and 6 bits of NDRO storage per data source. The buffer memory requirements are identical to the buffer memory in the ZFN Telemetry Data Compressor.

Methods were investigated for implementing a reference memory with a storage capacity to meet the requirements of the Hybrid ZVA/FVA Data Compressor.

Preliminary evaluation was made of a new memory organization, capable of meeting the increased storage capacity requirement. This new organization appears to be more flexible and more economical than previous organizations of memories with both DRO and NDRO storage capability. Results of the preliminary evaluation indicated that this new organization should be practical. Another part of the reference memory investigation was a study of possible improvements by use of integrated circuits for circuit functions within the memory. It was determined that at least three available integrated circuit sense amplifiers have suitable thresholding characteristics. However, unless power can be removed from these circuits for a large part of the time, their power dissipation will be too high (125 to 200 milliwatts per circuit with approximately 60 circuits required per system).

Recommendations are detailed in section 7.

Section 2
BASIC ALGORITHM RULES

The rules implemented in the breadboard are discussed in this section.

2.1 ZVA Rules

The basic rules for the Zero Order, Variable Corridor, Artificial Preceding Sample Transmitted (ZVA) algorithm are listed below. These rules are direct excerpts from Chapter II of Reference 1.

Basic rules for ZVA redundancy removal:

a. "The occurrence of a non-redundant sample requires that a new corridor be established. This is accomplished by drawing lines of zero slope through the end points of the tolerance range placed around the non-redundant sample.

b. "The requirement for a subsequent sample to be redundant and thus discarded is that one end of the tolerance range placed about the sample fall within the corridor. Note that the sample itself is not required to be within the corridor. Each redundant sample modifies the corridor extended to the next sample in the following way. The new corridor consists of that part of the previous corridor which is overlapped by the tolerance range placed about the redundant sample.

c. "If the tolerance range placed about a sample does not overlap the corridor, the sample is non-redundant, but is not transmitted. Rather, the midpoint of the corridor used to analyze this sample, actually the predicted value

c. Continued

of the sample, is transmitted for the preceding sample. Hence, the transmitted sample is not a real-data sample but an artificial sample."

A pictorial example of the application of these rules is shown in Fig. 2-1. These basic ZVA rules are modified in accordance with paragraphs 2.2.6, 2.2.7 and 2.2.8 of Reference 3.

2.2 FVA Rules

The basic rules for the First Order, Variable Corridor, Artificial Preceding Sample Transmitted (FVA) algorithm are listed below and are direct excerpts from Chapter II of Reference 1. Two modifications of these rules are made in the breadboard. These modifications are discussed following the basic rules.

2.2.1 Basic Rules for FVA Redundancy Removal:

a. "The occurrence of a non-redundant sample requires that a new corridor be established. This is accomplished by placing a tolerance range around the non-redundant sample and drawing two straight lines. The first passes through the artificial preceding sample, which is transmitted, and the upper end of the tolerance range; the second passes through the sample and the lower end of the tolerance range.

b. "The requirement for a subsequent sample to be redundant and discarded is that one end of the tolerance range placed around the sample fall within the corridor. Note that the sample itself is not required to be within the corridor. Each redundant sample modifies the corridor extended to the next sample in the following way. If

b. Continued

one of the boundary lines of the corridor does not pass through the tolerance range placed around the redundant sample, a new boundary is established which passes through the nearest end point of the tolerance range.

c. "If the tolerance range placed about a sample does not overlap the corridor, the sample is non-redundant but is not transmitted. Rather an artificial value for the preceding sample is transmitted, which is simply the predicted value for the previous sample. This value is the midpoint of the corridor at the time corresponding to the previous sample, where the corridor is the one used for analysis of the non-redundant sample."

A pictorial example of the application of these FVA rules is shown in Fig. 2-2. These basic FVA rules are modified in accordance with paragraphs 2.2.6, 2.2.7 and 2.2.8 of Reference 3.

2.2.2 Modification of Basic FVA Rules Using Approximate Division

The implementation of any first order compression algorithm requires a capability for division. To perform the required division (division by an arbitrary positive whole number) requires a significantly large number of logic modules. A modified first order compression algorithm, which does not require full division for its implementation, has been developed under the LMSC Independent Development Program. This implementation requires many fewer logic modules than are required to implement the first order algorithm with full division. The simulated modified compression algorithm replaces each full division in the original compression algorithm with a division by the (logarithmically) nearest integral power of two. The advantage of this

modified algorithm lies in the fact that binary arithmetic division by integral powers of two may be accomplished by simple shifting operations, thus reducing by a considerable amount the total number of logic modules and the time required for division.

Because this modified first order algorithm is only an approximation of the first order algorithm using full division, one might expect its performance to be less efficient in comparison to the first order algorithm using full division.

Several runs on real telemetry data were made, (in the LMSC Independent Development Program) using the IBM 7094 to simulate both the modified and full compression algorithms. Two points were readily apparent from these runs. First, both modified and full compression algorithms, on a given run on the same data, produce almost identical numbers of output points; i.e., almost identical word compression ratios. (These output points are not produced at identical sample times by the two compression algorithms.) Second, the peak error criterion must be discarded when using the modified algorithm. However, the rms error incurred using the modified algorithm is only slightly greater than that incurred using full division algorithms.

In summary, the performance of the approximate algorithm seems to be comparable with the full divide algorithm in every respect, except for the peak error criterion. The peak error criterion was exceeded in so few instances that the reconstructed data resulting from the modified algorithms is as "acceptable" as that from the full algorithms.

The modified division routine was incorporated into the breadboard. However, there are four ways in which the modified division incorporated in the breadboard differs from the modified division used in the computer runs. First, in the breadboard the nearest power of two was used for division, while in the computer runs the nearest power of two (logarithmically) was used. The table below illustrates the differences.

Computer Run		Breadboard	
n	n Rounded Off to	n	n Rounded Off to
2	2	2	2
3 → 5	4	3 → 5	4
6 → 11	8	6 → 11	8
12 → 22	16	12 → 23*	16
23 → 45	32	24 → 47*	32
46 → 63	64	48 → 63	64

* Different from the computer run

Second, the breadboard does not have capability to carry decimal fractions, while in the computer runs, fractions were used in the computations. Third, the maximum run length in the breadboard was limited to 63, while the run length was unlimited in the computer. However, the breadboard can be expanded to incorporate longer run lengths. Fourth, the FVA algorithm was not used in the computer studies. The algorithm used in the computer studies was a First Order Interpolator with a Disjoined Line Segment (FOIDIS). A description of this algorithm can be found in Reference 2.

While the effects of the above modifications on the system performance have not been completely determined, it is believed that their effect on the system performance is minor. To verify this belief, either computer runs in which the breadboard is simulated or the use of the breadboard to compress actual telemetry data would be very beneficial in analyzing system performance.

2.2.3 Modification of Basic FVA Rules by Limiting the Range of the Corridor Boundaries

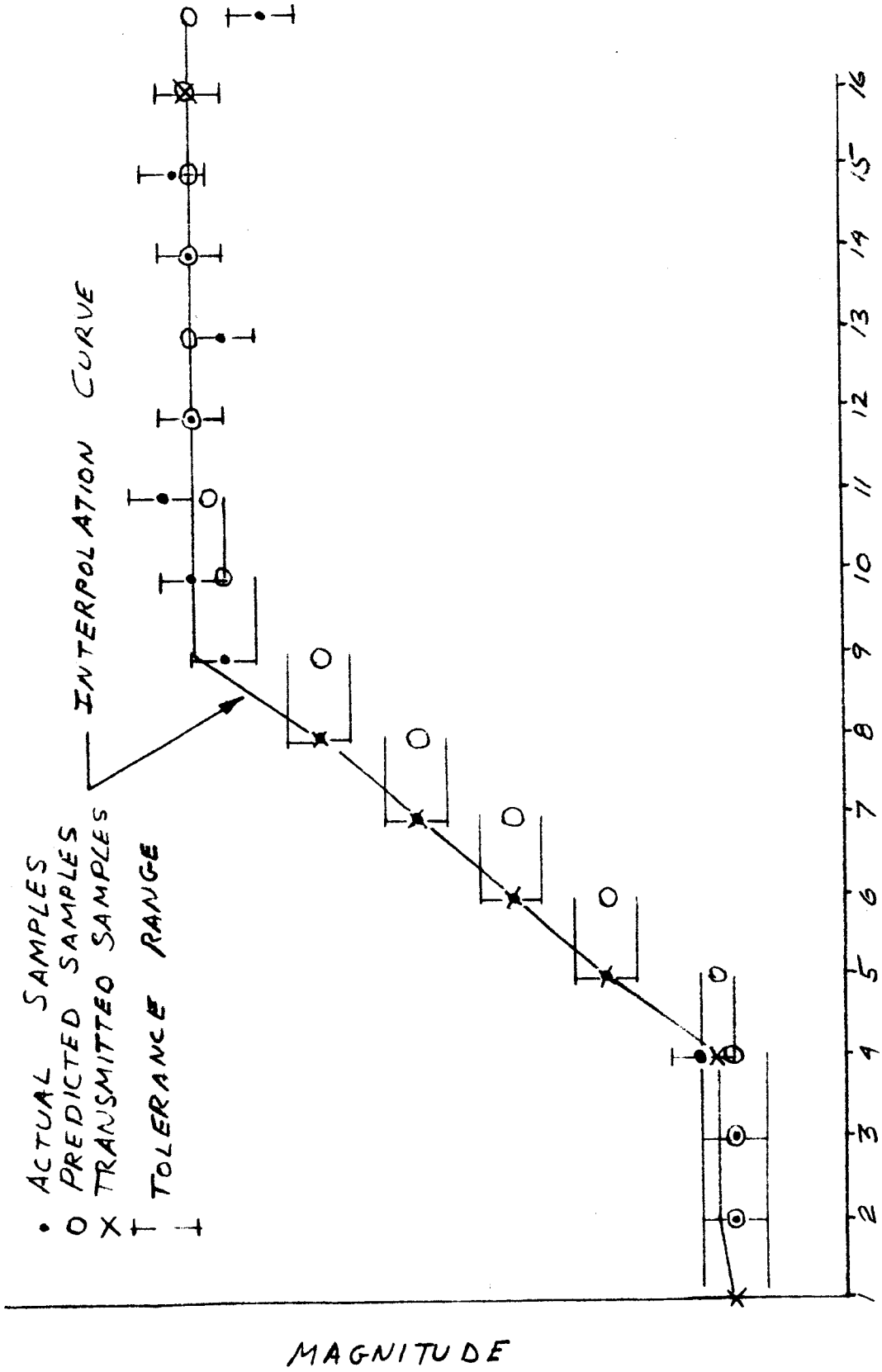
To completely follow the basic FVA rules in implementing the FVA algorithm, the calculation of the corridor boundaries requires provisions in the reference memory for storing negative numbers and for storing positive numbers up to twice full scale. If these provisions are not made or deviations from the FVA

rules are not made, errors as great as one-half scale may occur. These errors would occur when both the computed limits of the corridor either exceed full scale or are negative. In the breadboard when both the computed limits of the corridor either exceed full scale or are negative, a data point is inserted in the buffer which is not computed according to the basic FVA rules. There are two different situations which may occur, and they are described below:

1. If a data point has just been inserted in the buffer and both of the new corridor boundaries are either negative or greater than full scale, the previous actual data point is inserted in the buffer. In this special case the breadboard implements the First Order, Variable Corridor, Preceding Sample Transmitted (FVP) algorithm. This algorithm was described in LMSC 8-39-65-1, Reference 14.

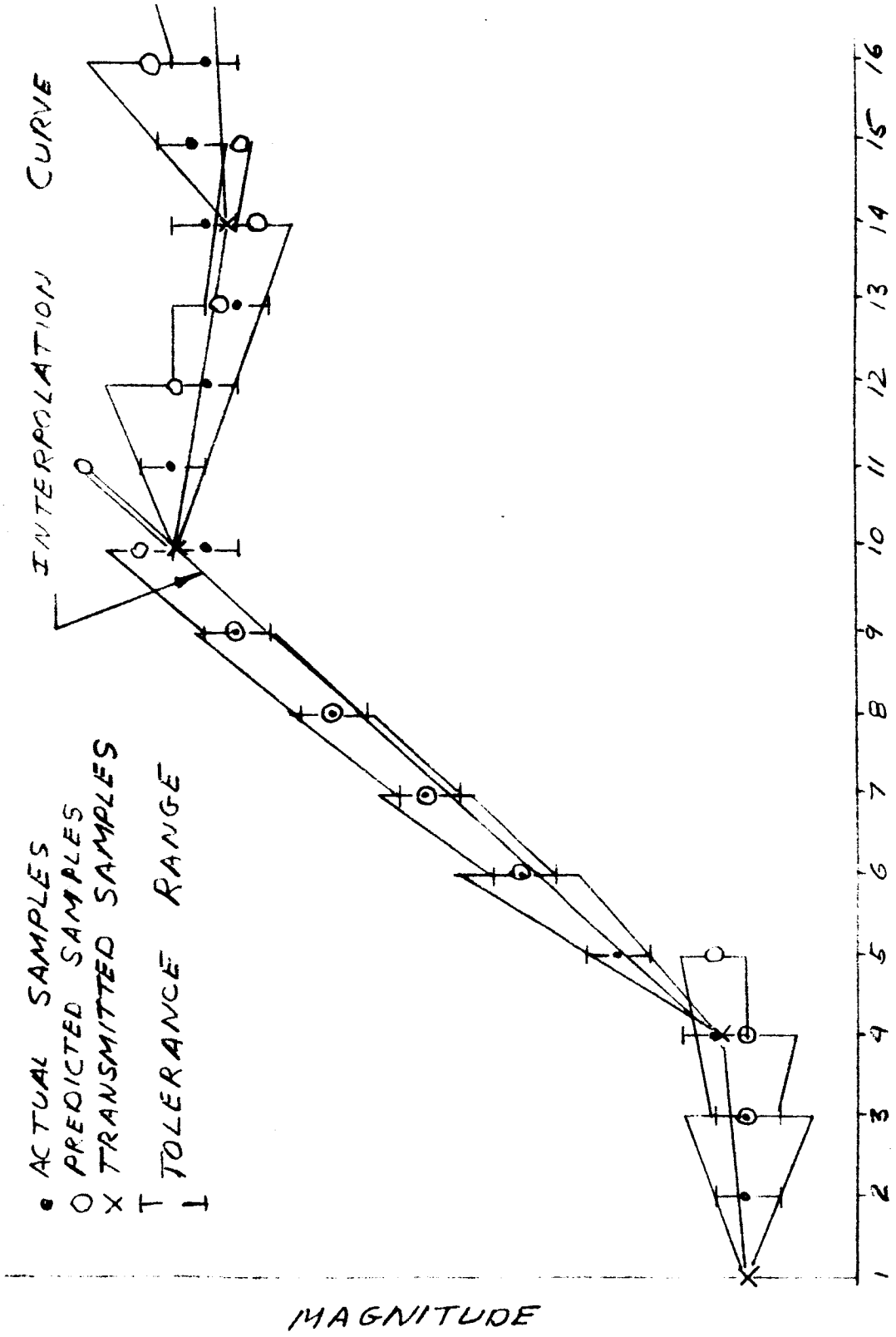
2. If a data point has not just been inserted in the buffer (run length of 2 or greater) and both of the corridor boundaries are either negative or greater than full scale, the previous lower corridor boundary is inserted in the buffer. Note that this point is an artificial point which is not computed according to the FVA rules.

These deviations from the FVA rules were taken because they result in significant hardware savings in meeting special situations in data behavior and cause no increase in the peak error criterion. These deviations may cause a decrease in word compression ratio; however, this decrease should be very slight.



ZERO ORDER, VARIABLE CORRIDOR, ARTIFICIAL PRECEDING SAMPLE TRANSMITTED - ZVA

FIG. 2-1



FIRST ORDER, VARIABLE CORRIDOR, ARTIFICIAL PRECEDING SAMPLE TRANSMITTED -- EVA

FIG. 2-2

Section 3
ZVA/FVA HYBRID DATA COMPRESSOR DESCRIPTIONS

The hybrid data compressor may be divided into three hardware sections:

1) The buffer and reference memories and associated circuitry.

2) The "high speed" logic section. This section consists of the logic which operates off the 2.8 MC clock and is shown in the flow diagram, Fig. 3-1, and the block diagram, Fig. 3-2.

3) The "low speed" logic section. This section consists of the logic which operates at a maximum clock rate of 288 KC and consists of hardware areas that have been implemented in the ZFN Telemetry Data Compressor.

The breadboard consists only of the "high speed" logic section and a small portion of the "low speed" logic section.

3.1 Buffer Memory Requirements

The buffer memory required by the hybrid data compressor is identical to the one used in the existing ZFN Data Compressor, developed by LMSC for MSFC. A functional description of the required buffer memory may be found in Reference 3. When operating the breadboard, feedback from the buffer memory to the breadboard is simulated by the test panel. Also, inputs to the buffer memory from the breadboard go instead to the test panel.

3.2 Reference Memory Requirements

The reference memory required by the hybrid data compressor is similar to the one used in the existing ZFN Data Compressor and described in Reference 3. However, 43 bits of storage per data source are required in the hybrid data compressor, compared with only 16 bits in the ZFN system. The reference memory requirements for the implementation of the FVA algorithm are discussed below.

Between data samples the reference memory must contain the last transmitted value, the upper and lower boundary of the corridor and the length of the corridor (or information with which to calculate these values). A rather straightforward approach is used in the proposed reference memory. The initial point of the corridor (Y_o), the upper boundary of the corridor (Y'_n), the lower boundary of the corridor (Y''_n) and the number of samples (n) since the last transmitted value would be stored directly in the reference memory.

The reference memory should contain 37 bits of DRO storage and 6 bits of NDRO storage per data source. Assignments of the bits are shown below.

n	Number of Samples	6 DRO
Y_o	Initial point of the corridor	10 DRO
Y'_n	Upper Corridor Boundary	10 DRO
Y''_n	Lower Corridor Boundary	10 DRO
R. S.	Redundant Sample Tag	1 DRO
	TOTAL DRO BITS	37

K	Tolerance Value	3 NDRO
A. S.	Algorithm Selection	1 NDRO
P	Priority Tag	1 NDRO
	Reference memory program bit	1 NDRO
	TOTAL NDRO BITS	6
	TOTAL REFERENCE MEMORY BITS	43

The Y_o and n reference memory locations are not used when the ZVA algorithm is being implemented.

When operating the breadboard the reference memory is simulated by either the test panel or the temporary storage registers in the breadboard, depending on the mode of the simulated input (static or dynamic). In the discussion of the flow diagram that follows, it is assumed that a complete hybrid data compressor (including buffer and reference memories, and the "low speed" logic section, as well as the "high speed" logic section) is present. Differences between the breadboard and a complete system are pointed out during the discussion.

3.3 Discussion of Flow Diagram

In the flow diagram, Fig. 3-1, the numbers at the corners of each block represent the logic sequence of the operation indicated in the block. During logic step 1, the new data value is inserted in register Y_n and the reference memory is read, with the information contained in the reference memory being transferred to the corresponding registers; i.e.,

Y'_n	is transferred to register Y'_n
Y''_n	is transferred to register Y''_n
Y_o	is transferred to register Y_o
n	is transferred to counter n
K	is transferred to register K
P	is transferred to the P flip flop
R. S.	is transferred to the R. S. flip flop
A. S.	is transferred to the A. S. flip flop

In the breadboard the K and P outputs of the reference memory are simulated by the test panel. The A. S. flip flop is also simulated by the test panel. Provisions for the reference memory program bit and the redundant sample tag are not included in the breadboard. When the test panel is programmed for a

static input, the Y_n', Y_n'' and Y_o outputs of the reference memory are simulated by the test panel and the n output of the reference memory is forced to equal 2. When the test panel is programmed for dynamic inputs, the Y_n', Y_n'', Y_o and n outputs of the reference memory are stored in the corresponding temporary storage registers. These registers are not reset during logic step 16; therefore the proper information is retained in the registers.

During logic step 2, the algorithm selection flip flop is examined and it is determined whether the data sample will be processed by the FVA or the ZVA algorithm.

3.3.1 FVA Implementation

When the algorithm selection flip flop is in the FVA state, the following will occur. During logic step 2, registers Y_n' and Y_n'' are examined to see if Y_n' and Y_n'' both equal all 1's (full scale) or if both Y_n' and Y_n'' equal all zeros. If neither condition exists, a count of one is added to the n counter; if the n counter then contains a count of 63, the sequence jumps to logic step 7_3. If not, the buffer queue length is examined, and if the queue length is near zero, the sequence jumps to logic step 7_3. If the buffer queue length was not near zero, the buffer fullness information and the priority flip flop are examined to determine if the data sample should be rejected because of buffer fullness. (When the buffer queue length exceeds the programmed level, no data points from low priority channels are allowed to be inserted in the buffer.) If the sample was not rejected, the tolerance value K is adjusted (depending on the buffer queue length) and the adjusted tolerance is inserted in register C. (The tolerance values are increased when buffer queue length exceeds certain programmed levels.) Also during logic step 2, the redundant sample flip flop is examined along with the buffer queue length. If the R.S. flip flop is a 1 and the buffer queue length is less than a programmed level, an artificial sample is inserted in the buffer. If the sample was not rejected because the buffer queue length was excessive or was not accepted because of the redundant sample process, the normal computation process is followed, starting with logic step 3.

The redundant sample logic is not included in the breadboard; therefore, the above decision process does not take place. Also, only two levels of buffer fullness are programmable.

During logic step 3 the information in register Y_n (data sample) is added to the information in register C (adjusted tolerance) and the result is stored in register A. If the result is greater than full scale, register A is set equal to full scale. During logic step 4 the information in register C (adjusted tolerance) is subtracted from the information in register Y_n (data sample) and the result is stored in register B. If the result is negative, register B is set equal to zero. During logic steps 5 and 6 the test is made to determine whether or not any point within the tolerance range placed around the data sample, Y_n, falls within the established corridor, (i.e., in between Y_n' and Y_n''). If the tolerance range placed around the data sample does not overlap the corridor, the sample is non-redundant and the next logic step is 7_3.

When logic step 7_3 is reached, the decision to calculate an artificial sample and insert the artificial sample in the buffer memory has been made. For a data sample processed by the FVA algorithm the decision to calculate an artificial sample could have been made for one of the following reasons:

1. The information in the n counter was equal to 63. (logic step 2)

2. Redundant Sample logic (logic step 2).

3. Buffer empty (logic step 2).

4. No point within the tolerance range placed around the data sample fell within the established corridor (logic step 5 or 6).

If an artificial sample is to be inserted in the buffer, path 7_3 through 14_3 is followed. During logic step 7_3 the midpoint of the corridor is calculated and stored in register A. In the breadboard, decimal fractions are not retained. Therefore, if A was calculated to have a coefficient of 1 in the 2^{-1} location, the 2^{-1} term is simply dropped. During logic steps 8_3, 9_3 and 10_3 the artificial data value for the preceding sample is calculated and stored in register B. Also, during logic step 8_3 register C is set equal to K, and during logic step 10_3 the n counter is set equal to zero.

During logic step 11_3 the artificial data value (contents of register B) is inserted in the buffer for transmission, the calculation of the new upper corridor boundary is started and the n counter is advanced to 1. During logic step 12_3 the calculation of the new upper corridor boundary is completed and stored in register Y_n'. If the information in Y_n' is greater than full scale, register Y_n' is set equal to full scale; if the information in register Y_n' is negative, register Y_n' is set equal to zero. During logic steps 13_3 and 14_3 the lower corridor boundary is calculated and stored in register Y_n''. If the lower corridor boundary is greater than full scale, register Y_n'' is set equal to full scale; if the lower corridor boundary is negative, register Y_n'' is set equal to zero. During logic step 14_3 the artificial data value which was put in the buffer during 11_3 (the new initial point of the corridor) is inserted in register Y_o.

If the upper corridor boundary is zero or the lower corridor boundary is full scale, a special situation exists. If this situation occurs, an accurate artificial sample cannot be calculated, nor can an accurate determination of redundancy be made at the next sample time. To relieve this problem a deviation is made from the basic rules for the FVA algorithm. At the next sample time the preceding data sample is transmitted. Ordinarily the preceding sample has not been stored in the reference memory and therefore is not available for transmission at the next sample time. However, when this special situation exists, Y_n, instead of the contents of register Y_o, is inserted in the reference memory location Y_o. Thus during logic step 15, Y_n', Y_n'', Y_n and n are written

in the reference memory, with Y_n being written in the Y_o location. If the new upper corridor boundary did not equal zero or the new lower corridor boundary did not equal full scale, Y_n', Y_n'', Y_o and n are written in the reference memory during logic step 15. During logic step 16 all registers and control flip flops are set equal to zero. In the breadboard, when the input is dynamic, registers Y_n', Y_n'', Y_o and the n counter are not set equal to zero.

If during logic steps 5 and 6 it was found that the tolerance range placed around the data sample fell within the established corridor, the data sample (Y_n) is considered redundant and the corridor must be extended to sample time n + 1. During logic step 7 the need to modify the upper boundary of the corridor is determined. If it does need to be modified, this is done and the upper boundary extended during logic steps 8_1, 9_1 and 10_1. If the upper corridor boundary does not need to be modified, it is extended during logic steps 8, 9 and 10. If the upper corridor boundary was calculated to be greater than full scale, register Y_n' is set equal to full scale; if the upper corridor boundary was negative, register Y_n' is set equal to zero. Also during logic steps 10 and 10_1 Y_n'' is inserted in register Y_n.

During logic step 11 the need to modify the lower corridor boundary is determined. If it does need to be modified, this is done and the lower boundary extended during logic steps 12, 13 and 14. If the corridor boundary does not need modification, it is merely extended during logic steps 12_1, 13_1 and 14_1. If the lower corridor boundary was calculated to be greater than full scale, register Y_n'' is set equal to full scale; if the lower corridor boundary was negative, register Y_n'' is set equal to zero.

If the lower corridor boundary is equal to full scale or the upper corridor boundary is equal to zero, an accurate artificial sample cannot be calculated, nor can an accurate determination of redundancy be made at the next sample time. In this case the contents of register Y_n (i.e., Y_n'' was placed in Y_n during logic step 10) are inserted in the reference memory in the Y_o location during logic step 15_1.

If during logic step 2 the data sample was rejected because of priority, the next operation occurs at logic step 8_2. During logic steps 8_2, 9_2 and 10_2 the upper corridor boundary is extended to sample number n + 1. This sequence is identical to the sequence followed by logic steps 8, 9 and 10. Also, during logic step 10_2, the information in register Y_n'' is placed in register Y_n. During logic steps 12_2, 13_2 and 14_2 the lower corridor boundary is extended to sample number n + 1. This sequence is identical to the sequence followed by logic steps 12_1, 13_1 and 14_1.

Following logic step 14_1, if the lower corridor boundary is equal to full scale or the upper corridor boundary is equal to zero, an accurate artificial sample cannot be calculated, nor can an accurate determination of redundancy be made at the next sample time. In this case the contents of register Y_n (i.e., the lower corridor boundary was placed in Y_n during logic step 10_2) are inserted in the reference memory in the Y_o location during logic step 15_1.

If the information in register Y_n was inserted in the Y_o memory location, this fact is detected during logic step 2 (the next time data from this channel is processed) by checking register Y_n'' for all ones and register Y_n' for all zeros. (These are the conditions necessary for the information in register Y_n to be inserted in the Y_o memory location in the first place.) If the above condition exists during logic step 2, the logic sequence jumps to logic step 10_6. During logic step 10_6 the contents of register Y_o are inserted in register B. Path 11_3 through 14_3 is then followed.

Equations in the format of $\frac{A+B}{N} + C = D$ appear several times in the flow diagram -- specifically in paths 8 through 10, 8_1 through 10_1, 8_2 through 10_2, 8_3 through 10_3, 12 through 14, 12_1 through 14_1 and 12_2 through 14_2. A more detailed flow diagram of this implementation is shown in Fig. 3-3. A, B, C and D are 10-bit natural binary numbers, and $N = 2^M$ where M is 1, 2, 3, 4, 5 or 6.

A modified form of 2's complement arithmetic is used in this sequence. The first step is to subtract A-B; the result may be positive or negative. If A-B

is positive, the result is in natural binary format. Next the result, E, is divided by N, with 2^0 being the least significant bit retained. C is then added to $\frac{A-B}{N}$. Since both C and $\frac{A-B}{N}$ are positive, the result, G, must also be positive, and it is possible for the result to be greater than full scale, $2^{10}-1$. However, in the breadboard D is limited to a maximum figure of $2^{10}-1$. If G is greater than $2^{10}-1$, this is detected by examining the carry flip flop, and D is set to full scale, $2^{10}-1$. If G is less than full scale, D=G.

If the result of the first step is negative, the result, E, is in 2's complement format. E is then divided by N and the result, F, is in 2's complement format. 2^0 is the least significant bit of F. Next C is added to F. Since F is negative and C is positive, the result, G, may be either positive or negative. If G is negative, this fact is detected by examining the borrow flip flop (same physical flip flop as the carry flip flop), and D is set equal to zero. If G is positive, D=G.

3.3.2 ZVA Implementation

All of the possible flow paths that could be followed in implementing the FVA algorithm were discussed above. The Y_o and n reference memory locations are not used when the ZVA algorithm is being implemented. Logic step 1 is the same for the ZVA implementation as described for the FVA implementation. When the algorithm selection flip flop is in the ZVA state, the following will occur. Logic step 2 is the same as for the FVA implementation except for the following:

1) The test for Y_n' equal to zero and Y_n'' equal to full scale is not made.

2) The test for n equal to 63 is not made.

If the data sample is rejected for buffer fullness, the logic sequence jumps to logic step 15. If not, during logic steps 3 through 6 the data sample is examined for redundancy. These steps are identical to the ones for the FVA implementation.

When logic step 7_3 occurs, the decision to calculate an artificial data point and insert the artificial data point in the buffer has been reached. For a data sample processed by the ZVA algorithm the decision to calculate an artificial data point could have been made for one of the following reasons:

1) Redundant Sample logic (logic step 2).
2) Buffer empty (logic step 2).
3) No point within the tolerance range placed around the data sample fell within the established corridor (logic step 5 or 6).

If an artificial data point is to be inserted in the buffer, the artificial data point is calculated and stored in register A during logic step 7_3. For the ZVA implementation the artificial data point is simply the mid-point of the corridor. During logic step 10_4 the value of the artificial data point is inserted in register B, and then during logic step 11_3 the artificial data point is written in the buffer memory.

During logic step 12_4 the new upper corridor boundary is calculated, and during logic step 14_4 the new lower corridor boundary is calculated. During logic step 15 Y_n' and Y_n'' are written in the reference memory. Note that memory locations Y_o and n are not used for the ZVA implementation. During logic step 16 all registers and decision flip flops are cleared to zero. In the breadboard, when the input is dynamic, registers Y_n' and Y_n'' are not set equal to zero.

If, during logic steps 5 and 6, it was found that the tolerance range placed around the data sample fell within the established corridor, the data sample Y_n, is considered redundant and the next step is logic step 7. During logic step 7 the need to modify the upper corridor boundary is determined. If the upper boundary needs to be modified, it is modified during logic step 10_5.

During logic step 11 the need to modify the lower corridor boundary is determined. If the lower boundary needs to be modified, it is modified during logic step 14_5.

The above discussion has covered all possible flow paths in the ZVA implementation.

3.4 "High Speed" Logic Section

Fairchild Diode Transistor Micrologic (DTμL) was used in implementing the "high speed" logic section. Reasons for using this logic family are given in Reference 4. The January 1966 issue of the magazine Electronic Products lists the following eight companies as second sources for the Fairchild DTμL line: General Micro-Electronics, ITT Semiconductor, Motorola, National Semiconductor, Philco, Signetics, Stewart-Warner, and Texas Instruments, Incorporated. The majority of the modules used in the breadboard were produced by Fairchild. However, a small sampling of Philco and Stewart Warner modules were intermingled with the Fairchild units.

3.4.1 High Speed Logic Modules

The following types of circuits were used:

930 -- Dual Four-Input Gate
932 -- Dual Four-Input Buffer
933 -- Dual Four-Input Extender
945 -- Clocked Flip Flop
946 -- Quad Two-Input Gate
948 -- Clocked Flip Flop

Complete information and specifications for these modules can be found in the Fairchild data sheets and application notes, References 5, 6, 7, 8, 9, 10 and 11. A brief discussion of the use of these modules is included below:

930 and 946 NAND Gates: Throughout this report the more positive of the two logic signals will represent a logical "1", and the more negative signal will represent a logical "0". Using this positive-logic form the 930 and 946 become NAND gates. NAND gates perform the AND function and invert the result. The logic symbol, pin locations, and truth table for these gates are shown in Fig. 3-4. The 930 gate is provided with input extension terminals for use when fan-in greater than 4 is needed. The 933 Dual Four-Input extender may serve this function, or silicon diodes such as the IN914 may be used. In the breadboard IN916 diodes, 933 Dual Four-Input extenders, and a diode array manufactured by Siliconix were used. All seemed to work equally well. The outputs of two or more 930 or 946 gates may be paralleled directly to produce a wired OR function. This connection proved very useful and is used quite extensively throughout the breadboard.

933 Dual Four-Input Extender: This module is two diode arrays and is used to increase the fan-in to 930 or 932 modules. The logic symbol pin locations and the truth table are shown in Fig. 3-5.

932 - Dual Buffer: The 932 Dual Four-Input Buffer is a power NAND gate used to drive high fan-out and/or high capacity loads. The logic symbol, pin locations and truth table are shown in Fig. 3-5. The 932 buffer cannot be used in the wired OR configuration as the 930 or 946 can. Only one 932 buffer is used in the breadboard, and it is used as a clock driver.

945 and 948 Clocked Flip Flop: The clocked flip flop features an AND gate input, permitting operation in either the R-S or J-K mode. The logic symbol, pin numbers and truth tables are shown in Fig. 3-6. The following three paragraphs are direct quotes from Reference 11 and discuss the clocked flip flop truth tables of Fig. 3-6.

"The Trigger input does not appear in the synchronous entry table, but its influence is implicit. Inputs are stated at time t=n, and outputs are stated at time t=n+1. The demarcation between times n and n+1 is an excursion of the Trigger input from low to high and back to low again. This unit time delay due to clocking is typical of synchronous operation.

"In response to signals applied to the direct inputs, both the Clocked Flip Flop outputs may be concurrently high. However, the outputs are always complementary in response to synchronous entry. Each pair of synchronous inputs must be high to affect the output. In the event that all synchronous inputs are high simultaneously, an unpredictable output condition will result. This ambiguity is characteristic of R-S Flip Flops and must be avoided. The twin Set and Clear inputs of DTμL Clocked Flip Flops simplify the resolution of R-S characteristic ambiguity.

"Since the Clocked Flip Flop outputs are always complementary in response to synchronous entry, they may be cross-connected to one of each of the Set and Clear inputs to assure that the ambiguous input condition can never occur. The terminal characteristics of a Clocked Flip Flop so connected are those of a J-K Flip Flop. The term J-K is applied to a flip flop which toggles whenever its inputs are both simultaneously active at clock time ....."

The specifications for these modules are given in the Fairchild data sheets, References 5, 6, 7, 8, 9 and 10, and are considered too complicated to be repeated in this report. The breadboard was designed following the Fairchild specifications. However, exact values of stray capacitance are not known and were estimated depending upon the physical routing of the wire. Propagation delays in a chain of four or more cascaded gates were calculated using typical rather than worst-case propagation delay times. The longest cascaded chain of gates in the breadboard is 6.

3.4.2 Notes on Logic Schematics

This subsection discusses the principal features of the logic schematics and is intended as an aid to understanding the schematics. Explanation of the terminology used in describing the logic signals can be found in Appendix A. The schematics are Figs. 3-7, 3-8 and 3-10 through 3-20.

The block diagram for the complete hybrid data compressor is shown in Fig. 3-2. The portion shown within the dotted lines indicates the part of the system included in the breadboard. The sizes of blocks in the diagram do not indicate the complexity or number of modules that make up the blocks. For example, the adder-subtractor consists of four modules, while the timing and control consists of approximately half of the breadboard, or about 130 modules. The numbers on the lines between the blocks indicate the logic step or steps during which information flows between the two blocks, and the arrowhead indicates the direction of flow. This flow of information between the blocks is routed by the timing and control circuits. In the following paragraphs each schematic is discussed, the relationship between the schematic and the block diagram is mentioned and the function of each module is described.

In the breadboard the free-running clock is simulated by an external pulse generator. However, in a flyable system the clock would be internally generated. A suitable clock circuit for this purpose has been developed on the LMSC Universal Electronic System program. The schematic of this circuit is shown in Fig. 3-7. For the hybrid data compressor a 2.8 MC crystal would be used, and the component values may be changed slightly.

The basic timing generation logic is on boards X, V and T. Referring to the schematic of board X, Fig. 3-8, pin 43 is the "data ready" input. This input is a "1" (high voltage) when a new data sample is ready to be processed. The 2.8 MC clock input is on pin 37. This clock is free-running and is non-synchronous with respect to the "data ready" input. The first high-going-low

transition of the 2.8 MC clock input, after the "data ready" input has become a "1", will set flip flop M to a "1", which in turn sets flip flop I to a "1". This enables gates N8 and N11, and the 2.8 MC clock flows through gates N8 and N11. Gate N11 is the clock input to the 12-state generator, flip flops A, B, E and F. The state sequence of flip flops A, B, E and F is shown below:

A	B	E	F	State #
0	0	0	0	1
1	0	0	0	2
0	1	0	0	3
0	0	1	0	4
1	0	1	0	5
0	1	1	0	6
0	0	1	1	7
1	0	1	1	8
0	1	1	1	9
0	0	0	1	10
1	0	0	1	11
0	1	0	1	12
0	0	0	0	1

Gate C6 decodes state 1 and gate C8 decodes state 11. When gate C6 decodes state 1, the synchronous Clear inputs to flip flop D are primed. At the next clock pulse flip flop D goes to the "0" state and gate G8 is enabled, allowing the clock pulse to flow through gate G8 and gate K6. The output of K6 is called "10 clocks" and is one of two basic system clock lines that feed all of the shift registers. When state 11 is decoded by gate C8, the synchronous Set inputs to flip flop D and the synchronous Clear input, C1, to flip flop H are primed. The following clock pulse causes flip flop H to go to the "0" state and flip flop D to go to the 1 state, inhibiting gate G8. The "10 clocks" output is forced to a "0". Referring to the timing diagram Fig. 3-9, there were 10 clock pulses on the output of gate K6 before it was inhibited. When flip flop H went to the "0" state, gate J8 was enabled, allowing one clock pulse through gate K8. This output is called "11th pulse" and is the second of two basic system clock lines that feed most of the shift registers. The next clock pulse sets flip flop H back to the "1" state, which in turn inhibits gate J8

as well as advancing the 16-state generator to the next state. The 16-state generator consists of flip flops T, S, R and Q plus gates L6 and L8. The state sequence of flip flops T, S, R and Q is shown below:

$T(2^0)$		$S(2^1)$	$R(2^2)$	$Q(2^3)$	State #
1	0	0	1	0	1
0		0	1	0	2
0		1	1	0	3
1		1	1	0	4
1		0	0	0	5
0		0	0	0	6
0		1	0	0	7
1		1	0	0	8
1		0	0	1	9
0		0	0	1	10
0		1	0	1	11
1		1	0	1	12
1		0	1	1	13
0		0	1	1	14
0		1	1	1	15
1		1	1	1	16
1		0	1	0	1

Gates J6 and N3 decode state #16 and prime the synchronous inputs of flip flop I. When flip flop H goes to the "1" state during state 16, flip flop I is set to the "0" state, inhibiting gates N8 and N11. This inhibits all clock pulses throughout the system until the "data ready" input again becomes a "1". Waveforms on board X are shown in Fig. 3-9.

The 16 states of the 16-state generator on board X are decoded on boards V and T. These 16 decoded states are combined with various control and decision signals to form timing signals that correspond one for one to the logic steps shown in the flow diagram, Fig. 3-1.

Also on board T are three control flip flops. Gates P6 and S11 are connected as a set-reset flip flop. Gate P6 is a "1" when the decision to transmit an artificial data point has been made. Flip Flop L stores the answer to the questions asked in logic steps 7 and 11. The state of this flip flop controls

the flow path 7 through 14. Flip flop K is a "1" if the data sample is rejected because of priority.

Board R contains the input and output circuits of register A. As is shown on the block diagram, data flows in and out of register A many times, and the many gates on board R control the inflow and outflow of data. Stages 2^9, 2^8, 2^7, 2^1 and 2^0 of register A are located on board R, while stages 2^6 through 2^2 are located on board B. Below is a list of the functions performed by each of the gates on board R.

Gate T8	If the result of logic step 3 is greater than full scale, gate T8 detects this fact and allows the "11th pulse" to flow through, which sets register A to full scale (all ones).
Gates J6, J8 and R8	Provide 10 shift pulses during logic steps 3, 5, 7_T, 8, 8_1, 8_2, 8_3, 10_T, 13_3 and 14_T so that information either flows into or out of register A.
Gates J3, J11 and R6	Provide divide shift pulses during logic steps 9, 9_2, 13, 13 and 13_2 so that the information in register A is divided by n.
Gates F6 and M8	Provide one extra shift pulse during logic step 7_3 so that the result in register A is divided by 2.
Gates I6 and I11	Provide the data in register A to the B inputs of the adder during logic steps 5 and 7.
Gates Q8, I8 and I3	Provide the data in register A to the A inputs of the adder during logic steps 10, 10_1, 10_2, 14, 14 and 14_2.

Gates Q6, M8 and M11	Provide the inverted data in register A to the A inputs of the adder during logic steps 8_1, 8_3, 10_3, 12_3 and 14_3.
Flip Flop G	This flip flop is called the $A2^{10}$ flip flop and is used for the purposes listed below: (1) Stores whether the result of logic steps 8_1, 8_2, 12, 12_1 and 12_2 was positive or negative. (2) Serves as the 2^{10} stage of register A during logic steps 11_3 and 13_3 and therefore performs the multiplication of A by 2.
Gates K6 and K8	Provide the subtractor outputs to $A2^{10}$ during logic steps 11_3.
Gates M6, K3 and K11	Provide the subtractor outputs to $A2^{10}$ during logic step 13_3.
Gates O6 and O8	Provide the CFF_* output to $A2^{10}$ during logic steps 8, 8_2, 12, 12_1 and 12_2.
Gates O3 and F8	Provide 10 shift pulses to $A2^{10}$ during logic steps 11_3 and 13_3. The output of 0_3 could replace 11_3 on inputs K_4 and K_{10}. If this were done, the need for gates K_3 and K_{11} would be eliminated.

Gate F11	Provides a shift pulse to $A2^{10}$ during logic steps 8, 8_2, 12, 12_1 and 12_2.
Gates F3 and O11	Clears $A2^{10}$ to the "0" state at the end of logic steps 9, 9_3, 12_3 and 16.
Gates T6, L6 and L8	Provide the state of $A2^{10}$ as an input to $A2^9$ during logic steps 9, 9_2, 11_3 and 13_T.
Gates L3 and L11	Provide the adder outputs as an input to $A2^9$ prior to the 11th pulse during logic steps 3 and 7_3.
Gates P6 and P8	Provide the state of CFF_* as an input to $A2^9$ during the 11th pulse of logic step 7_3.
Gates S6, H6 and H8	Provide the output of the adder as an input to $A2^9$ during logic steps 8, 8_2, 12, 12_1 and 12_2.
Gates S8, H3 and H11	Provide the inverted outputs of the adder (i.e., subtractor outputs) to $A2^9$ during logic steps 5, 7, 8_1 and 8_3.

Board P contains the input and output circuits of register Y'_n. Also, the CFF_* flip flop, the 8,12 ± flip flop, and three timing circuits are on board P. Stages 2^9, 2^8, 2^7, 2^6 and 2^0 of register Y'_n are on board P, while stages 2^5, 2^4, 2^3, 2^2 and 2^1 are on board AA. Below is a list of the functions of each of the gates on board P.

Gates M3 and M11	Provide the inverted data in register Y'_n to the A inputs of the adder during logic steps 7, 8 and 8_2.
Gates Q6, M6 and M8	Provide the data in register Y'_n to the B inputs of the adder during logic steps 6, 7_3, 10, 10_1, 10_2 and 10_3.

Gates K6 and K8	If the result of logic steps 10, 10_1, 10_2 or 12_4 is greater than full scale, gates K6 or K8 detect this and allow the 11th pulse to flow through and set register Y'_n to full scale (all ones).
Gates G8, G3, J3, J6, J11 and J8	Provide a Set signal to each stage of register Y'_n that is on board R. Individual gates are required for each stage because each stage can be separately set to the "1" state by the reference memory sense amplifiers. Therefore, a common Set signal, as is used for register A, cannot be used for register Y'_n.
Gates F3 and F11	Provide divide shift pulses during logic steps 9_1 and 9_3 so that the information in register Y'_n is divided by n.
Gates F6, F8 and 06	Provide 10 shift pulses during the logic steps in which data is shifted into or out of register Y'_n.
Gate Q8	If the result of logic steps 10, 10_1 or 10_2 is negative, gate Q8 detects this and provides a pulse to set register Y'_n to zero.
Gates G8 and G6	Reset register Y'_n to zero.
Gates D, H and L	These gates route the inputs to $Y'_n 2^9$ and insure that only one input is applied at a time.
Gate S	Provides various timing signals as labeled.
Flip Flop P	This flip flop is the CFF_* and is not reset by the 11th pulse.

Flip Flop T	This flip flop stores whether the result of logic steps 8_T and 12_T is positive or negative.
Gate R6	If the result of logic step 4 is negative, gate R6 detects this and provides a pulse to set register B to zero.
Gate 08	Detects if Y'_n and Y''_n equal zero.

Board M contains the input and output circuits for the C register, portions of the C register, the adder-subtractor and some control circuits. The 2^9, 2^3, 2^2, 2^1 and 2^0 stages of register C are on board M, while stages 2^8 through 2^4 are on board B.

Gates J3 and J6	Provide the data in register C to the B inputs of the adder during logic steps 3, 4, 11_3, 12_4, 13_3 and 14_3.
Gate M	These four gates write the inputs to $C2^9$ and insure that only one input is applied at a time.
Gate P3	Provides shift pulses to register C.
Gate P11	Provides clear pulse to register C.
Gate P8	Provides CFF_* -- this is required for fan-out reasons.

Gates Q, R and S, along with flip flop T, comprise the adder-subtractor circuits. Dual outputs for both the sum and the no-sum outputs are provided for fan-out reasons. Flip Flop T is the carry flip flop. The circuit mechanizes the following equations:

$$\text{Pin 39 and 47} = A\overline{BC} + \overline{A}B\overline{C} + \overline{AB}C + ABC$$
$$\text{Pin 31 and 33} = \overline{A}BC + A\overline{B}C + AB\overline{C} + \overline{ABC}$$

When this circuit is used for addition A+B, the sum is on pins 39 and 47, and the not-sum is on pins 31 and 33. When the circuit is used for subtraction $\overline{A}$-B, the difference is on pins 31 and 33, and the not-difference is on pins 39 and 47. Note that when subtraction is performed, the minuend is applied to the inverted A inputs and the subtrahend is applied to the B inputs.

Flip flops N, L, H, G, D and C are control flip flops. If the result of logic step 12_3 is negative, gate K8 detects this fact and provides a signal to set register Y'_n to zero. If the result of logic step 12_3 is greater than full scale, gate K6 or J8 detects this fact and provides a signal to set register Y'_n to full scale. If the result of logic step 14_3 is negative, gate J11 or 08 detects this fact and provides a signal to set register Y''_n to zero. If the result of logic step 14_3 is greater than full scale, gate 06 detects this fact and provides a signal to set register Y''_n to full scale.

Board K contains the approximate divide circuits. Flip flops F, E, A, B, C and D are connected as a ripple-carry counter. During logic step 1 the n count is transferred to this counter from the reference memory, and during logic step 2 the count is increased by one through gate Q8. Gate S6 detects if the count is 63. Gates M, I, H, J and G decode the count in the counter to one of six ranges listed below:

		n COUNTER BETWEEN
N10	"0"	0 and 2
010	"0"	3 and 5
P10	"0"	6 and 11
L10	"0"	12 and 23
K10	"0"	24 and 47
K12	"0"	48 and 63

Flip flops R, N, O, P, L and K are connected as a shift register. The range gates insert a "1" into one of the flip flops. During logic steps 9_T, 13, 13_1 and 13_2 gate S8 is enabled, and the "10 CLOCKS" flow through gates S8, T6 and Q11 and shift the "1" in the register toward Flip Flop R. When Flip Flop R becomes a "1", gate S8 is inhibited, stopping the flow of clock pulses through S8. Depending on which shift register stage was set to a "1", either 1, 2, 3, 4, 5 or 6 clock pulses flowed through gate S8. These pulses are outputted on pin 15 and are called $\overline{\div \text{ clocks}}$. The number of pulses corresponding to the state of the n counter is listed below:

STATE OF n COUNTER	# OF $\overline{\div \text{ CLOCKS}}$	$\div$ BY
0 - 2	1	2
3 - 5	2	4
6 - 11	3	8
12 - 23	4	16
24 - 47	5	32
48 - 63	6	64

Gates T8 and T11 clear the shift register to "0". Gates Q3 and Q6 clear the counter to "0".

Board H contains the input and output circuits for registers Y_o and Y''_n, as well as parts of registers Y_o and Y''_n. Stages 2^9, 2^8, 2^7, 2^6 and 2^0 of register Y''_n are on board H, while stages 2^5, 2^4, 2^3, 2^2 and 2^1 are on board B. Stages 2^4, 2^3, 2^2, 2^1 and 2^0 of register Y_o are on board H, while stages 2^8, 2^7, 2^6, 2^5 and 2^4 are on board D.

Gates Q8, M3 and N11 — Provide the data in register Y_o to the B inputs of the adder during logic steps 8_T, 12, 12_1 and 12_2.

Gates M6 and M8 — Provide for recirculating the information in register Y_o.

Gates M3 and M11	Provide the data in register B to the input of $Y_o 2^9$ during logic step 14_3.
Gates Q6 and N6	Provide shift pulses to register Y_o.
Gate N8	Clears register Y_o to zero.
Gates S6, J3 and J8	Provide the data in register Y''_n to the $\bar{A}$ inputs of the adder during logic steps 5, 11, 12 and 12_2.
Gates O3, J6 and J11	Provide the data in register Y''_n to the A inputs of the adder during logic steps 7_3, 14 and 14_2.
Gate O6	Provides shift pulses to register Y''_n.
Gates P and L	These eight gates route the inputs to $Y''_n 2^9$ and insure that only one input is applied at a time.
Gate T8	If the result of logic step 14, 14_1 or 14_2 is greater than full scale, gate T8 detects this and allows the 11th pulse to flow through and set register Y''_n to full scale (all ones).
Gates R3, R6, R8 and R11	These four gates provide a Set signal to each stage of register Y''_n. Individual gates are required for each stage because each stage can be separately set to the "1" state by the reference memory sense amplifier. Therefore, a common Set signal, as is used for register A, cannot be used for register Y''_n.

Gate S8	If the result of logic step 14, 14_1 or 14_2 is negative, gate S8 detects this fact and allows the 11th pulse to flow through and clear register Y_n'' to zero.
Gate T6	If the result of logic step 14_4 is negative, gate T6 detects this fact and allows the 11th pulse to flow through and clear register Y_n'' to zero.
Gates 08 and 011	Clear register to Y_n'' to zero.

Board F contains the input and output circuits for registers Y_n and B as well as parts of registers Y_n and B. Stages 2^9, 2^8, 2^7, 2^6 and 2^0 of register Y_n are located on board F, while stages 2^5 through 2^1 are located on board D.

Gates Q6, M3 and M11	Provide the data in register Y_n to the $\bar{A}$ inputs of the adder during logic steps 4, 13 and 14_4.
Gates Q8, M6 and M8	Provide the data in register Y_n to the A inputs of the adder during logic steps 3, 11_3 and 12_4.
Gates R6 and J3	Provide the shift pulses to register Y_n.
Gate J6	Clears register Y_n to zero during logic step 16.

Stages 2^9, 2^8, 2^7, 2^6 and 2^0 of register B are located on board F, while stages 2^5 through 2^1 are located on board B.

Gates J11 and R8	Provide the shift pulses to register B.
Gates T3, 03 and 06	Provide the data in register B to the $\bar{A}$ Inputs of the adder during logic steps 6 and 12.

Gates S6, 08 and 011	Provide the data in register B to the B inputs of the adder during logic steps 11, 12_3, 14 and 14_3.
Gates S8, T6, T8, L3, L6, L8 and L11	These gates route the inputs to $B2^9$ and insure that only one input is applied at a time.
Gates J8, P3, P6, P8 and P11	These gates transfer the data in register B to the buffer memory during logic step 11_3. Note that this data is the artificial data point. Since there is no buffer memory in the breadboard, the output of these gates goes to the test panel.
Gate T11	Gate T11 is a timing circuit, whose output is 4.

Boards D and B contain the center sections of registers Y_n, Y_o, Y''_n, A, B and C. The breadboard was initially designed for 5-bit data words and was later expanded to incorporate 10-bit data words. When this was done there was no space on the original boards for the additional register stages. Therefore, these register stages are separated from the rest of their respective registers.

Board AA contains stages 2^5, 2^4, 2^3, 2^2 and 2^1 of register Y'_n, gating associated with register Y'_n, and various independent gating functions.

Gate P11	This gate inverts the $\overline{\div \text{ CLOCK}}$ signal for use on boards AA, B, P and R.
Gate I8	This gate detects when register Y''_n is to be shifted by the 10 CLOCKS.
Gate K8	Gate K8 is a timing gate and provides 9, $9_3 + 12_3$ signal to board R.

Gates K3 and K6	These gates provide signal 12_3·11th pulse to board M.
Gates D3 and D6	These gates are connected as a flip flop. If register Y_n' and Y_n'' are both equal to full scale, Flip Flop D is set such that pin D_6 is a "1" during logic step 2.

The need for the circuits discussed below was not realized until testing of the breadboard was well under way. Therefore, these circuits are well separated from associated circuits.

Gate D11	If D_6 is a "1", gate D11 prevents logic step 10_3 from occurring.
Gates G6 and G8	If D_6 is a "1", these gates provide the data in register Y_o to the input of $B2^9$. This enables logic step 10_6 to occur.
Gates H8, H6 and H3	Provide the data in register Y_n to 2^9Y_o during logic step 15_1.
Gates L3, L6 and H11	Provide the data in register Y_n'' to the input of Y_n2^9 and $Y_n''2^9$ during logic steps 10, 10_1 and 10_2.
Gates L8 and L11	Provide the data in register Y_n to the input Y_n2^9 at all times except during logic steps 10, 10_1 and 10_2.

3.5 "Low Speed" Logic Section

The "low speed" logic section consists of hardware areas which have been implemented in the ZFN Telemetry Data Compressor using Texas Instruments series 51 logic modules. Although the series 51 modules were usable in the ZFN system, they are not fast enough to be used in the "high speed" logic section of the hybrid data compressor. The Fairchild DTμL family could be used to implement the "low speed" logic section as well as the "high speed" section. However, the speed of the Fairchild DTμL line is not required in the "low speed" section. A survey was conducted of the integrated circuit field to select a family or families of logic modules for use in the "low speed" section of the hybrid data compressor. The obvious choice would seem to be the TI series 51 family. However, the series 51 cannot be specified to drive the Fairchild DTμL modules and therefore was eliminated from consideration. Major considerations in the survey are discussed below.

Temperature

The specifications required that the system operate from -20°C to +85°C. An expanded temperature range from -55°C to +125°C was used in evaluating the integrated circuits. Integrated circuits are generally specified over the -55°C to +125°C temperature range, not the -20°C to +85°C temperature range. Thus, comparisons are easier to make over the wider temperature range. A more important reason for using the wider temperature range in evaluating integrated circuits is that a significant built-in design margin results when the modules are used over limited temperature range.

Speed

Propagation delays equal to or less than 250 nanoseconds for the gates and 1 microsecond for the flip flop were required.

Power

The only purpose of using a logic family other than the Fairchild DTμL line in the "low speed" section is power savings. Since the system is designed for the space environment, the lower the power the better. A general rule of thumb for digital logic circuits states, "the higher the speed the higher the power consumed". Thus power savings is traded for high speed. However, in the "low speed" section of the hybrid data compressor, high speed is not required, and the trade can be made.

Fan-In and Fan-Out

Most families have approximately the same fan-in and fan-out capabilities. Any flip flop with a fan-out greater than five was considered acceptable. Any gate with a fan-out greater than eight was considered acceptable.

Interface with Fairchild DTμL

Since the "low speed" modules and the "high speed" modules must interface with one another, the "low speed" modules must interface with the Fairchild DTμL.

Second Sources

Second and third sources are a major plus factor.

Cost

The cost of integrated circuits has been constantly lowered in the past, and all indications are that the cost will continue to be lowered. There is very little difference in price between most families at the present time, and the lowest-price family today may or may not be the lowest-price family six months from now. For these reasons cost was not a major consideration.

The following logic families were considered the best:

Fairchild	Low Power DTL
Motorola	MC 200 series
Signetics	SE 400 series

All of these families could interface with the Fairchild DTμL and all met the temperature, speed, fan-in and fan-out requirements. The Fairchild family does not have a second source at this time. In fact, the family has not been publicly announced, but Fairchild representatives indicate that it will be announced shortly. Generally, after a logic family is announced second sources soon follow. Also, the logic form factors of the Fairchild line are limited at this time. The power consumed by each of the families is listed below.

	POWER CONSUMED	
	GATE	FF
Fairchild	1.0 mw	4.8 mw
Motorola	6.0 mw	1.6 mw
Signetics	6.5 mw	1.4 mw

The estimated number of circuits in the "low speed" section is 130 flip flops and 540 gates. The power required by each of the three families, as well as the Fairchild DTL, is calculated below.

Fairchild Low Power DTL

$$\text{Power} = 1.0\ (540) + 4.8\ (130) = 1.06 \text{ watts}$$

Motorola

$$\text{Power} = 6.0\ (540) + 16\ (130) = 5.3 \text{ watts}$$

Signetics

$$\text{Power} = 6.5\ (540) + 14\ (130) = 5.3 \text{ watts}$$

Fairchild DTL

$$\text{Power} = 8.2\ (540) + 50\ (130) = 8.8 \text{ watts}$$

It is felt that the shortcomings of the Fairchild low power family will be corrected in time (second sources and logic form factors). Because the power required by the Fairchild low power family is significantly less than either Motorola or Signetics, the Fairchild line is recommended for use in the "low speed" section. The logic symbol, pin locations and truth table for these modules are shown in Fig. 3-21.

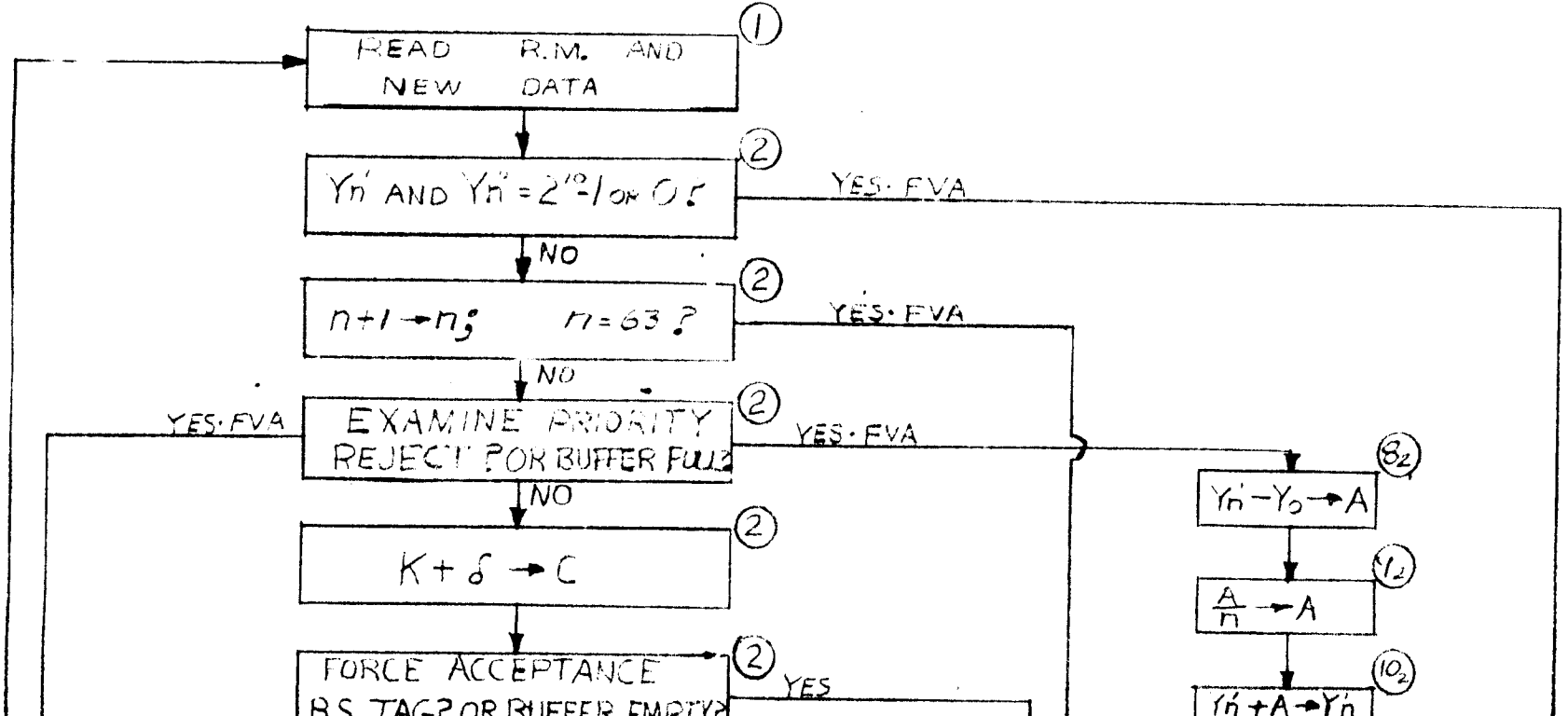
3.6 Construction Features of the Breadboard

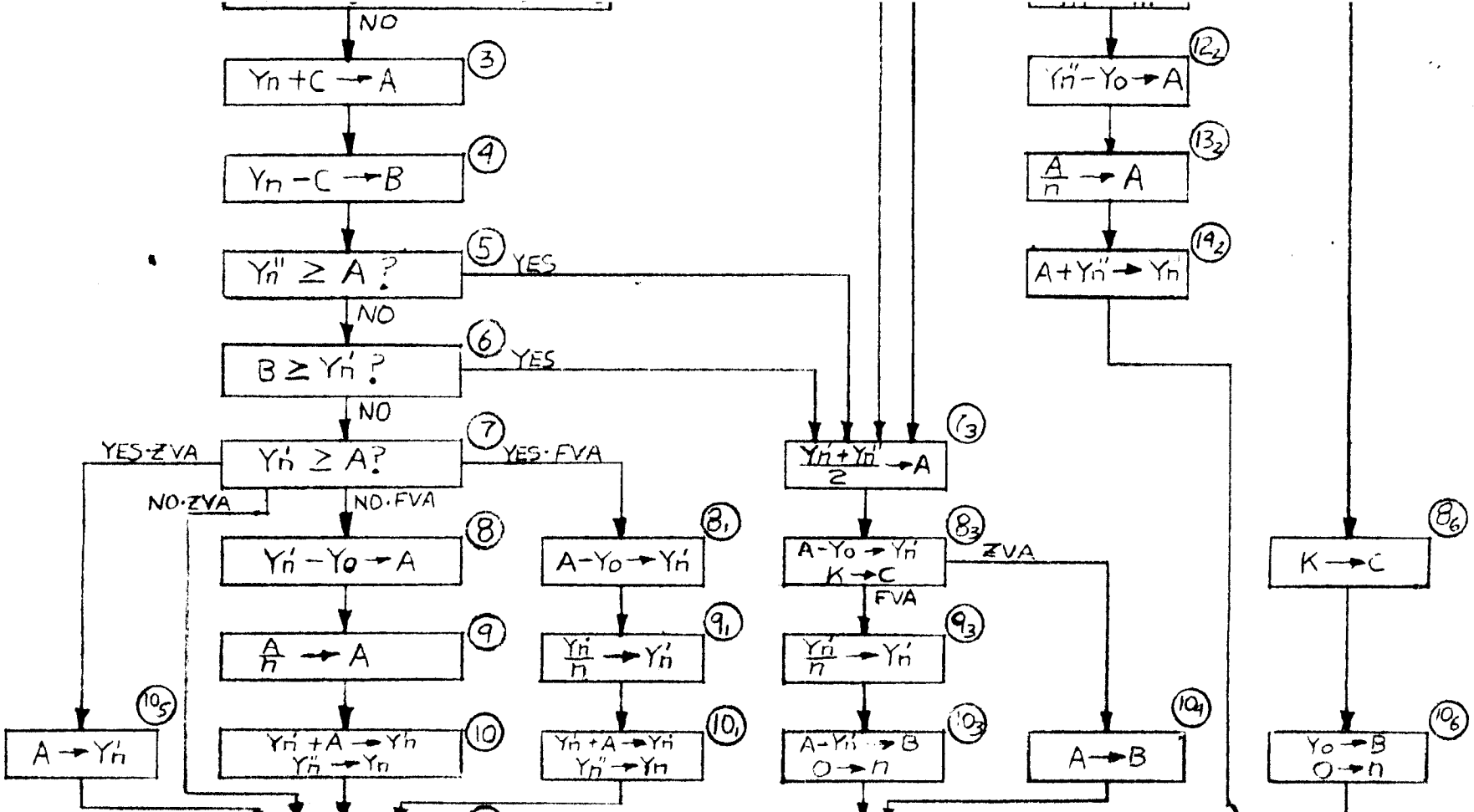
The breadboard is constructed using etched circuit boards with full-temperature-range flat-pack integrated circuit modules soldered in place. Interconnection of the modules is made by point-to-point wiring. A 66 pin connector is mounted to the bottom edge of each board and the boards are plugged into the top of a wire-wrap receptacle plate. Interconnections between circuit boards are made by wiring between pins on the bottom of the wire-wrap receptacle plate. External electrical connections to the test panel are made by wires from board Z. Figure 3-22 shows the breadboard and the test panel, and Fig. 3-23 shows the breadboard with board Z removed. Figure 3-24 is a view of the bottom of the wire-wrap receptacle plate showing the interconnection wiring between boards.

Figure 3-25 shows the module side of a typical board, board F. The letters underneath the modules signify the module location. The numbers around module J indicate the pin numbering of the modules. Starting with the right-most pin, the external connector pins are numbered 1 through 66, from right to left as shown. Figure 3-26 is a view of the wiring side of board F, showing the ground plane and point-to-point wiring.

GENERAL FLOW DIAGRAM

ZVA-FVA HYBRID DATA COMPRESSOR

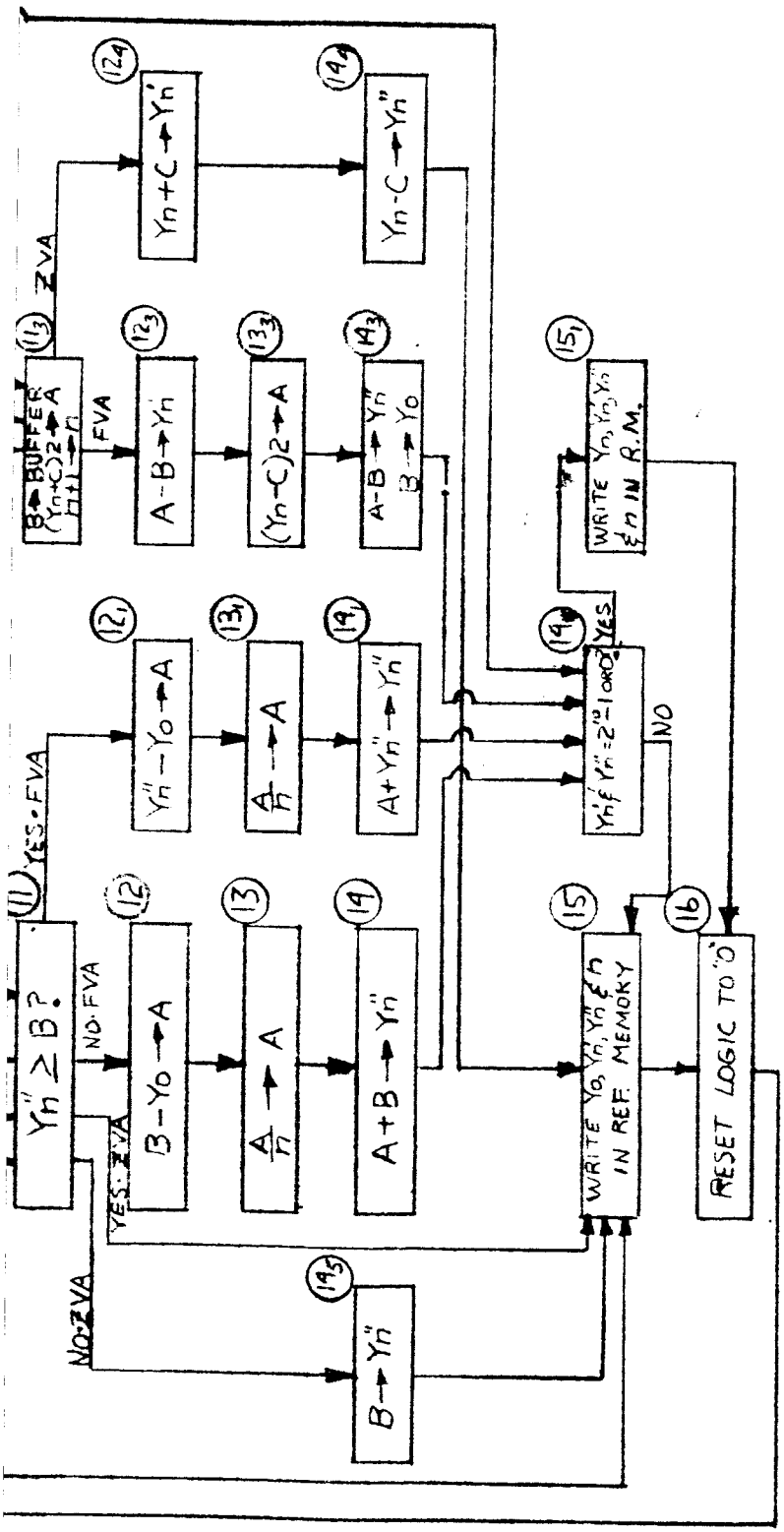



NO
$Y_n + C \rightarrow A$
3
$Y_n - C \rightarrow B$
4
$Y_n'' \geq A$?
5
YES
NO
$B \geq Y_n'$?
6
YES
NO
$Y_n' \geq A$?
7
YES·ZVA
YES·FVA
NO·ZVA
NO·FVA
$Y_n' - Y_0 \rightarrow A$
8
$\frac{A}{n} \rightarrow A$
9
$Y_n' + A \rightarrow Y_n'$
$Y_n'' \rightarrow Y_n$
10
$A \rightarrow Y_n'$
10_5
$A - Y_0 \rightarrow Y_n'$
8_1
$\frac{Y_n'}{n} \rightarrow Y_n'$
9_1
$Y_n' + A \rightarrow Y_n'$
$Y_n'' \rightarrow Y_n$
10_1
$\frac{Y_n' + Y_n''}{2} \rightarrow A$
13_3
$A - Y_0 \rightarrow Y_n'$
$K \rightarrow C$
8_3
ZVA
FVA
$\frac{Y_n'}{n} \rightarrow Y_n'$
9_3
$A - Y_n' \rightarrow B$
$0 \rightarrow n$
10_3
$A \rightarrow B$
10_4
$Y_n'' - Y_0 \rightarrow A$
12_2
$\frac{A}{n} \rightarrow A$
13_2
$A + Y_n'' \rightarrow Y_n'$
19_2
$K \rightarrow C$
8_6
$Y_0 \rightarrow B$
$0 \rightarrow n$
10_6

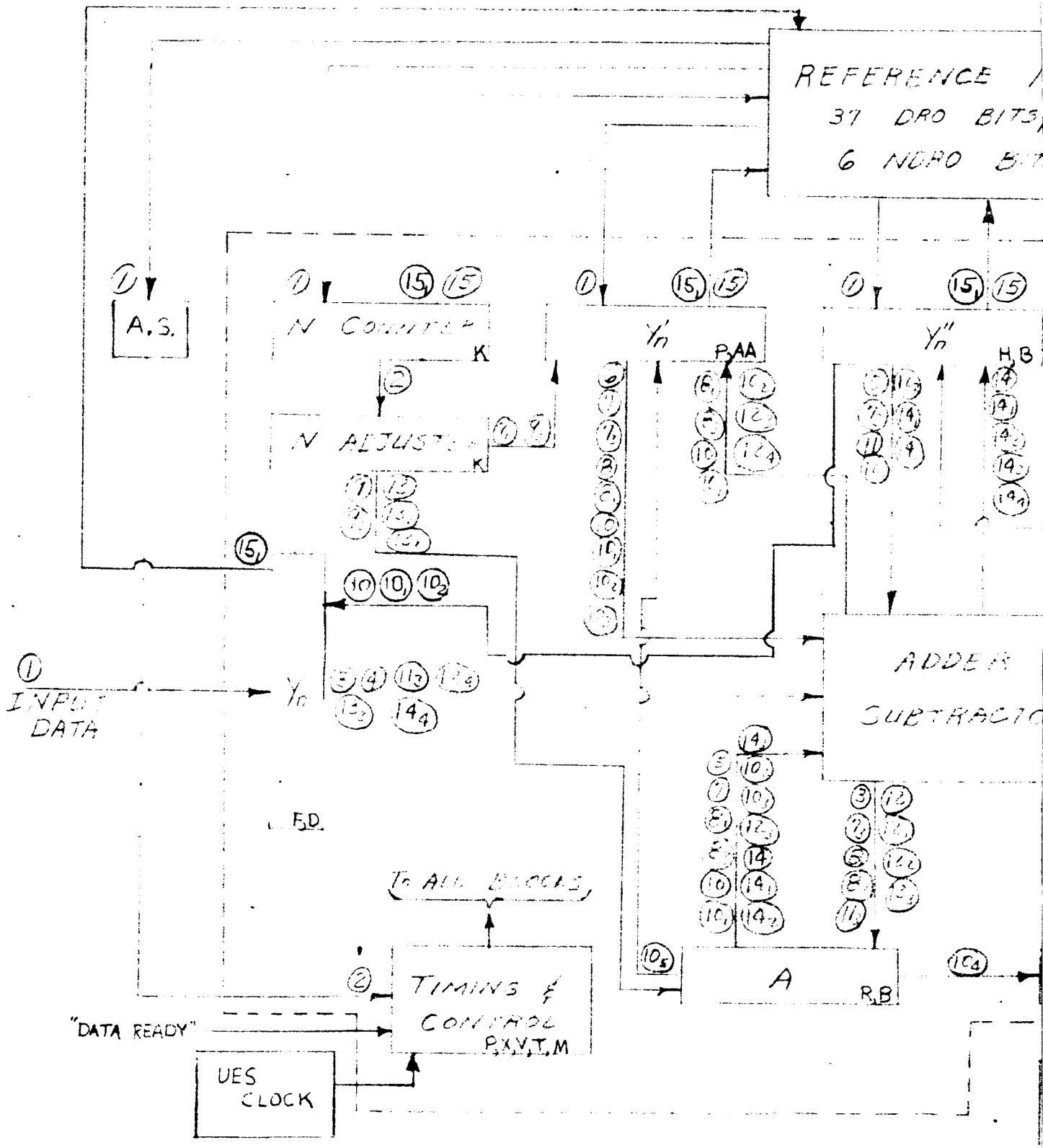
LMSC
669224

FIG. 3-1
GENERAL FLOW DIAGRAM

DRAWN BY [illegible]
DATE 7 FEB 1966



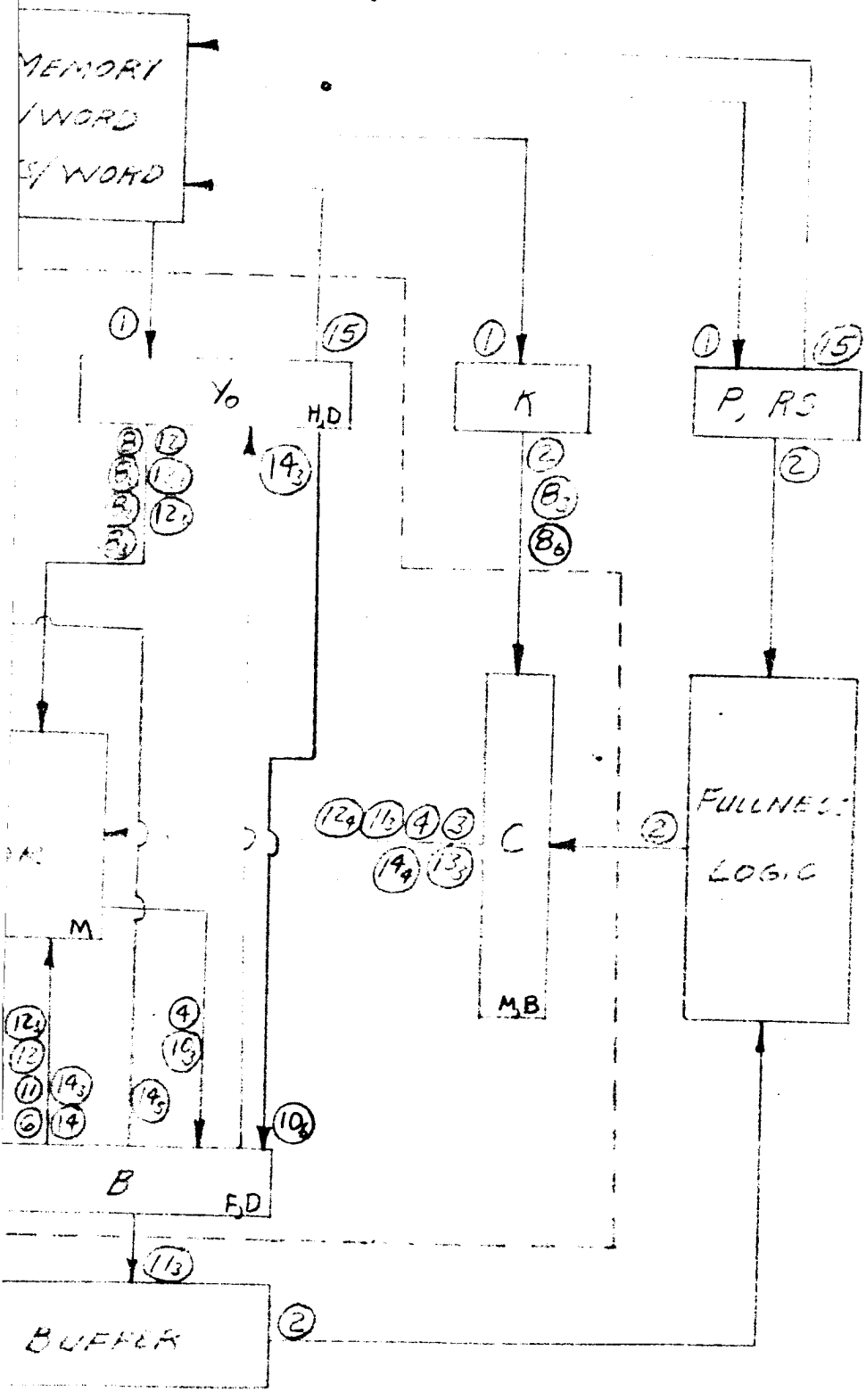
-ARTIAL BLOCK DIAG



NOTE: 1) FEASIBILITY BREADBOARD INCLUDE
2) LETTERS IN LOWER RIGHT CORNER OF BLOCK INDICA

ZVA/FVA DATA COMPRESSOR

FIG. 3-2

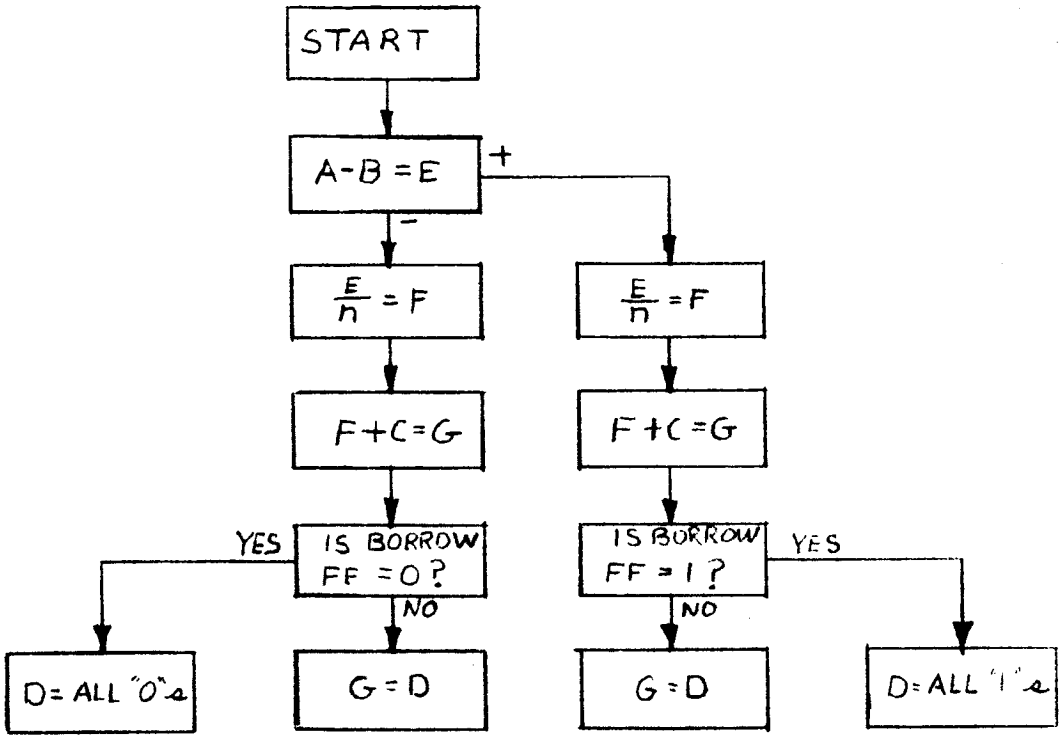


THE BLOCKS SHOWN WITHIN THE DOTTED LINE.
TES BOARD LOCATION OF BLOCK.

FLOW DIAGRAM IMPLEMENTING $\frac{A-B}{n}+C=D$

FIG. 3-3

DRAWN BY W.A LEISER
DATE 3 FEB 1966

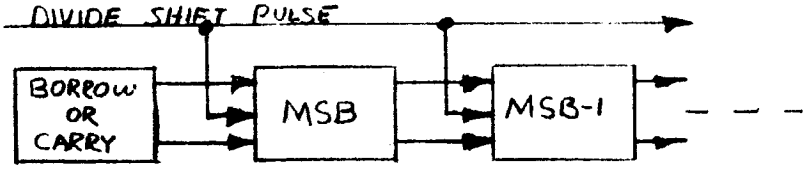


NOTES:

① A, B, C, & D ARE 10 BIT NATURAL BINARY NUMBERS.

② $n = 2^m$ WHERE $M = 1, 2, 3, 4, 5,$ OR 6.

③ WHEN SHIFTING TO DIVIDE THE STATE OF THE BORROW (CARRY) FF IS NOT CHANGED, AND THE INPUT TO THE MSB IS THE BORROW (CARRY) FF, AS SHOWN BELOW.



TWO 4 BIT EXAMPLES ARE SHOWN BELOW

	B	2^3	2^2	2^1	2^0	2^{-1}
A=13	0	1	1	0	1	
B=9	0	1	0	0	1	
	0	0	1	0	0	
n=2	0	0	0	1	0	
C=4	0	0	1	0	0	
D=6	0	0	1	1	0	

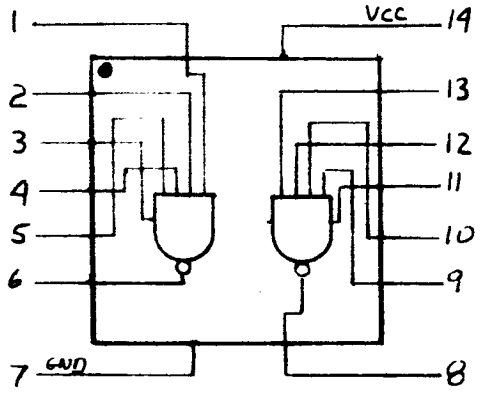
	B	2^3	2^2	2^1	2^0	2^{-1}
A=9	0	1	0	0	1	
B=13	0	1	1	0	1	
	1	1	1	0	0	
n=2	1	1	1	1	0	
C=4	0	0	1	0	0	
D=2	0	0	0	1	0	

DTuL 930 AND 946 GATES

FIG. 3-4

DRAWING NO. [illegible]
DATE 3 FEB 1966

930

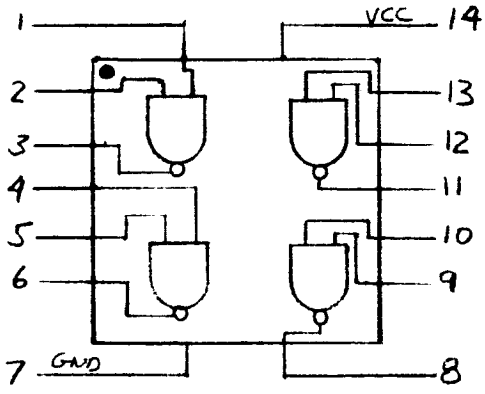


LOGIC EQUATIONS

$6 = \overline{1 \cdot 2 \cdot 4 \cdot 5}$

$8 = \overline{9 \cdot 10 \cdot 12 \cdot 13}$

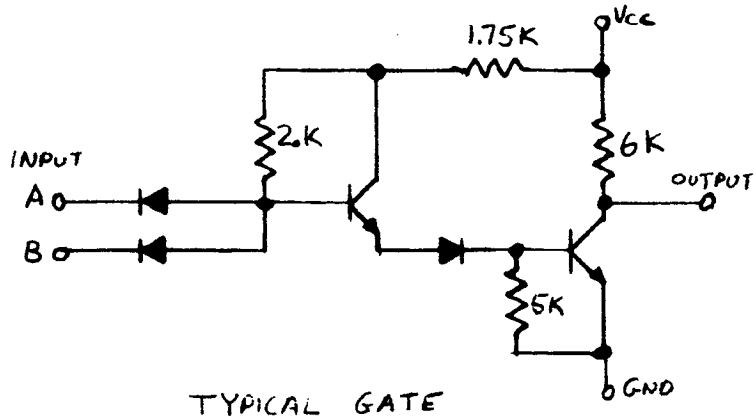
946



LOGIC EQUATIONS

$3 = \overline{1 \cdot 2}$ $\quad 8 = \overline{9 \cdot 10}$

$6 = \overline{4 \cdot 5}$ $\quad 11 = \overline{12 \cdot 13}$



TYPICAL GATE SCHEMATIC

TRUTH TABLE FOR GATES

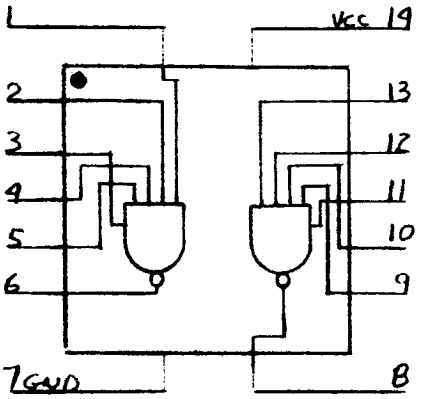
INPUT		OUTPUT
A	B	
0	0	1
0	1	0
1	0	0
1	1	0

"0" = LOW VOLTAGE

"1" = HIGH VOLTAGE

DTuL 932 AND 933
FIG. 3-5

932

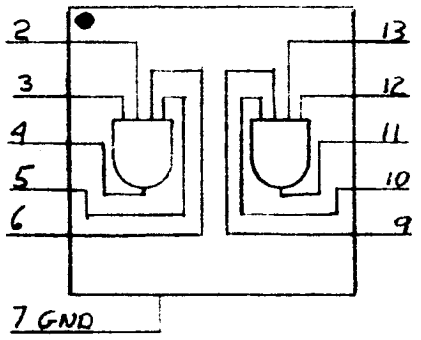


LOGIC EQUATIONS

$6 = \overline{1 \cdot 2 \cdot 4 \cdot 5}$

$8 = \overline{9 \cdot 10 \cdot 12 \cdot 13}$

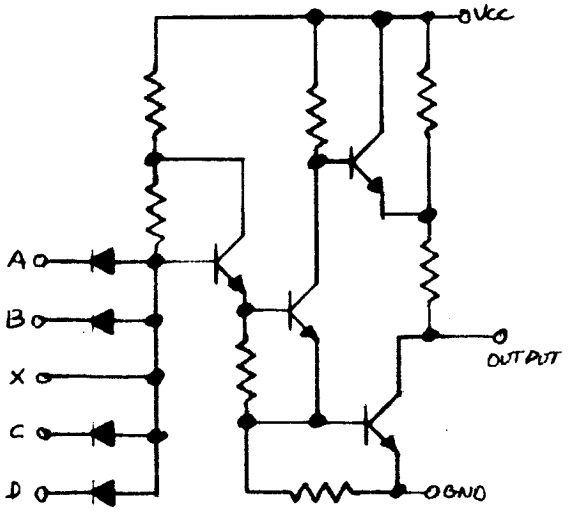
933



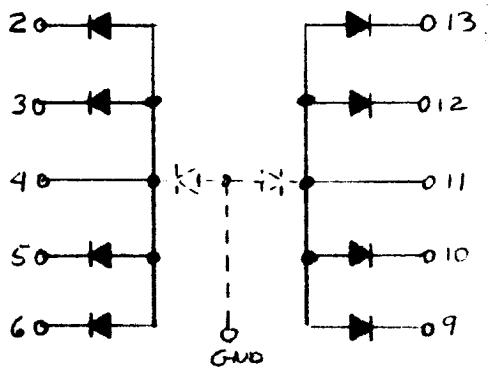
LOGIC EQUATIONS

$4 = 2 \cdot 3 \cdot 5 \cdot 6$

$11 = 9 \cdot 10 \cdot 12 \cdot 13$



SCHEMATIC 932
(ONE SIDE)

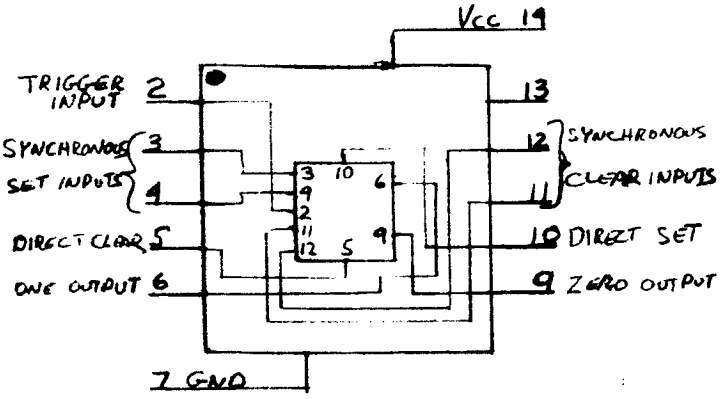


SCHEMATIC 933

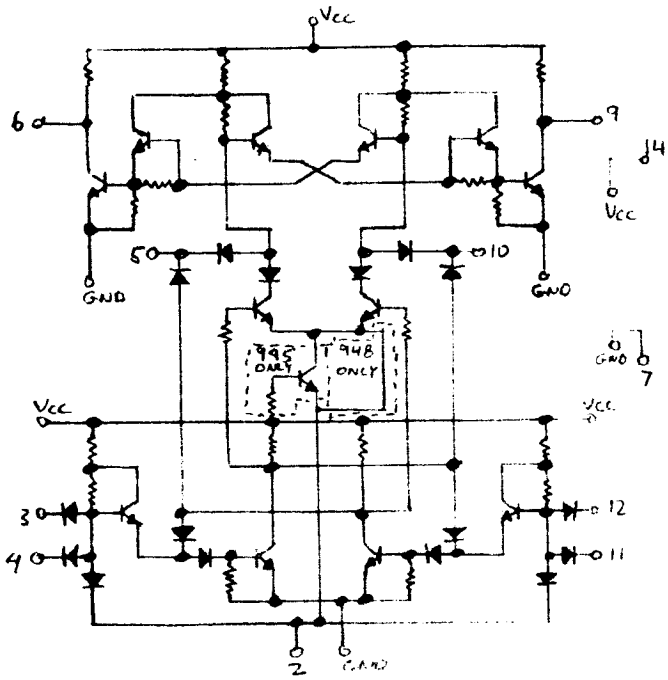
DTuL 945-948 FLIP-FLOPS

FIG. 3-6

DRAWN BY [illegible]
DATE 3 FEB 1966



THE SMALL SQUARE IS THE FLIP FLOP LOGIC SYMBOL USED IN THIS REPORT.



SCHEMATIC

TRUTH TABLES

ASYNCHRONOUS ENTRY

DIRECT INPUTS SET	DIRECT INPUTS CLEAR	OUTPUTS ONE	OUTPUTS ZERO
1	1	X	$\overline{X}$
1	0	0	1
0	1	1	0
0	0	1	1

"1" = HIGH VOLTAGE
"0" = LOW VOLTAGE
X = IMMATERIAL EITHER "1" OR "0" WITH EQUAL EFFECT.

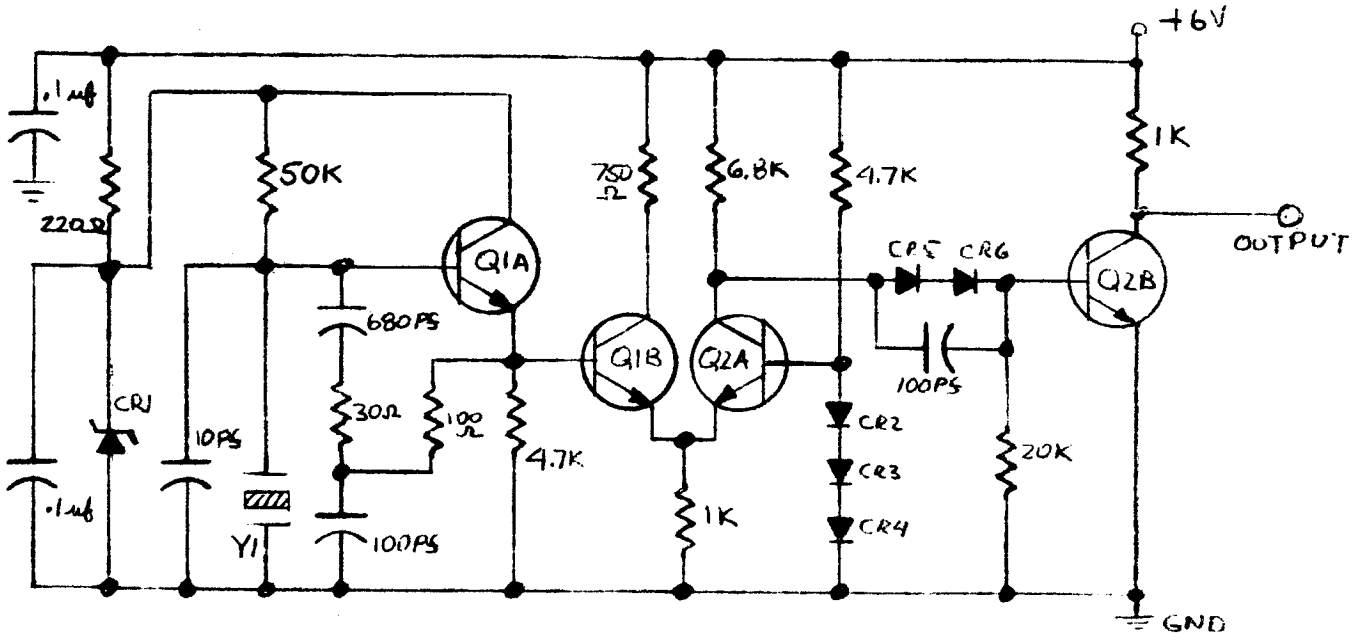
SYNCHRONOUS ENTRY

SYNCHRONOUS INPUTS AT $t=n$ SET_1	SET_2	$CLEAR_1$	$CLEAR_2$	OUTPUTS AT $t=n+1$ ONE	ZERO
0	X	0	X	X	$\overline{X}$
0	X	X	0	X	$\overline{X}$
X	0	0	X	X	$\overline{X}$
X	0	X	0	X	$\overline{X}$
0	X	1	1	0	1
X	0	1	1	0	1
1	1	0	X	1	0
1	1	X	0	1	0
1	1	1	1	AMBIGUOUS	

UES CLOCK CIRCUIT

FIG. 3-7

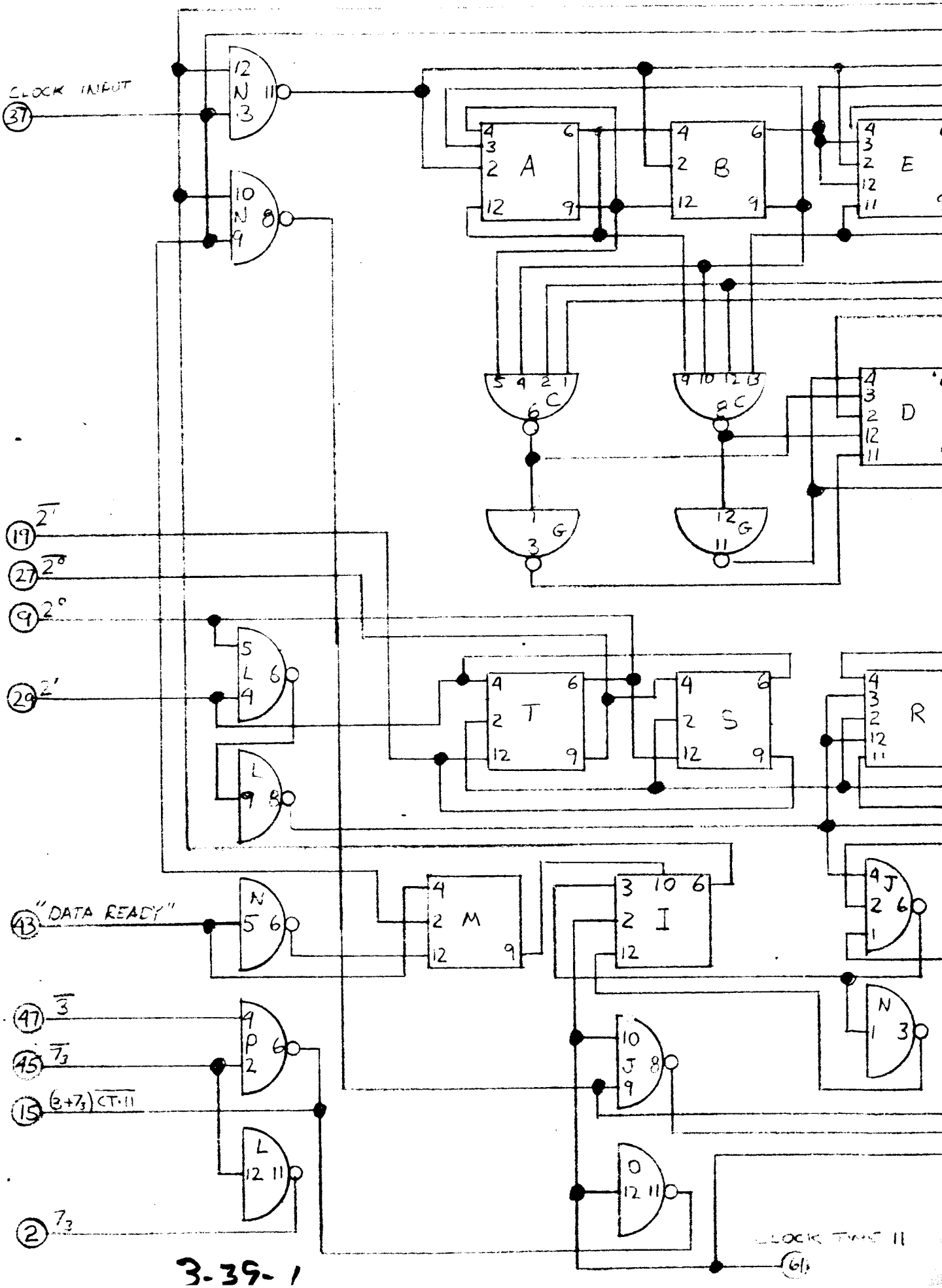
DRAWN BY W.A. LEISER
DATE 2 FEB 1966



CR1- PD 6001A
CR2-CR6 PD290
Y1 CR18A

Q1, Q2 MD 1126F (DUAL)

ALL RESISTORS TYPE RC05


CLOCK INPUT
37
"DATA READY"
43
47
45
15
2
19
27
9
29
61
A
B
E
C
D
G
T
S
R
M
I
J
N
P
L
O
CLOCK TIME 11

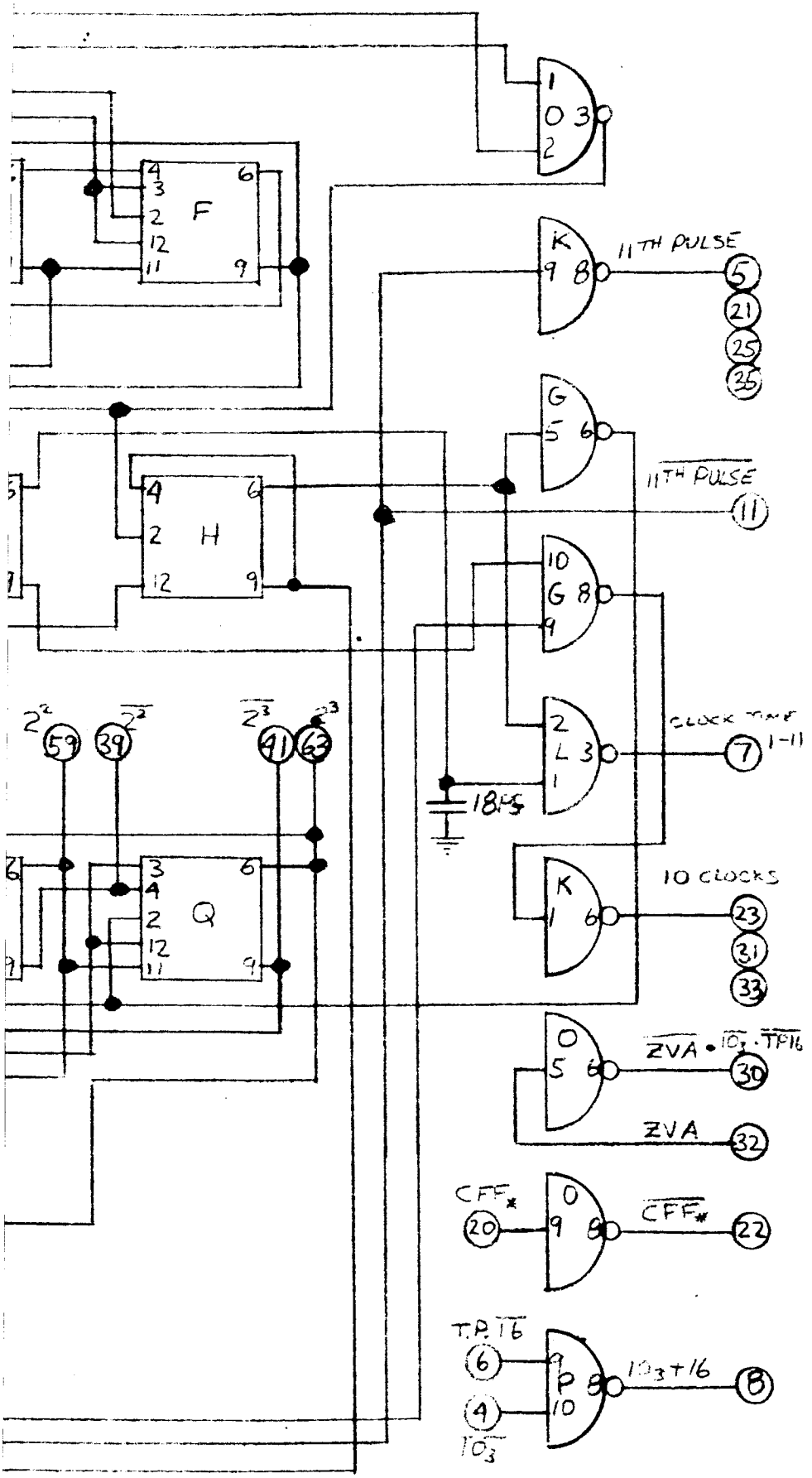
BOARD X

TIMING BOARD #J

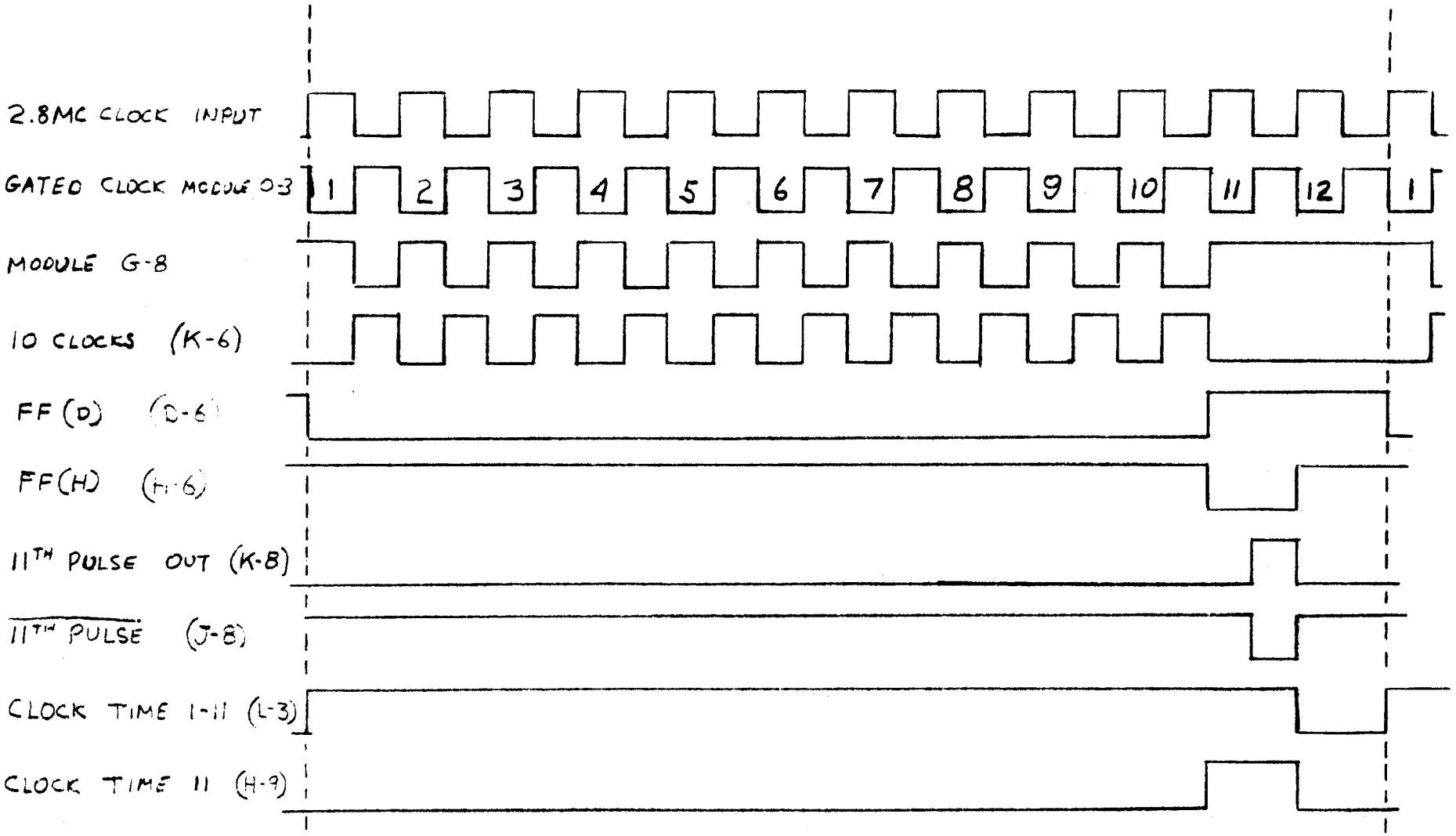
FIG. 3-8

SEE NOTE 1	
A-948	K-932
B-948	L-946
C-930	M-945
D-945	N-946
E-948	O-SW946
F-948	P-930
G-946	Q-945
H-945	R-945
I-945	S-945
J-930	T-945

DRAWN BY W.A. [illegible]
DATE 26 JAN [illegible]



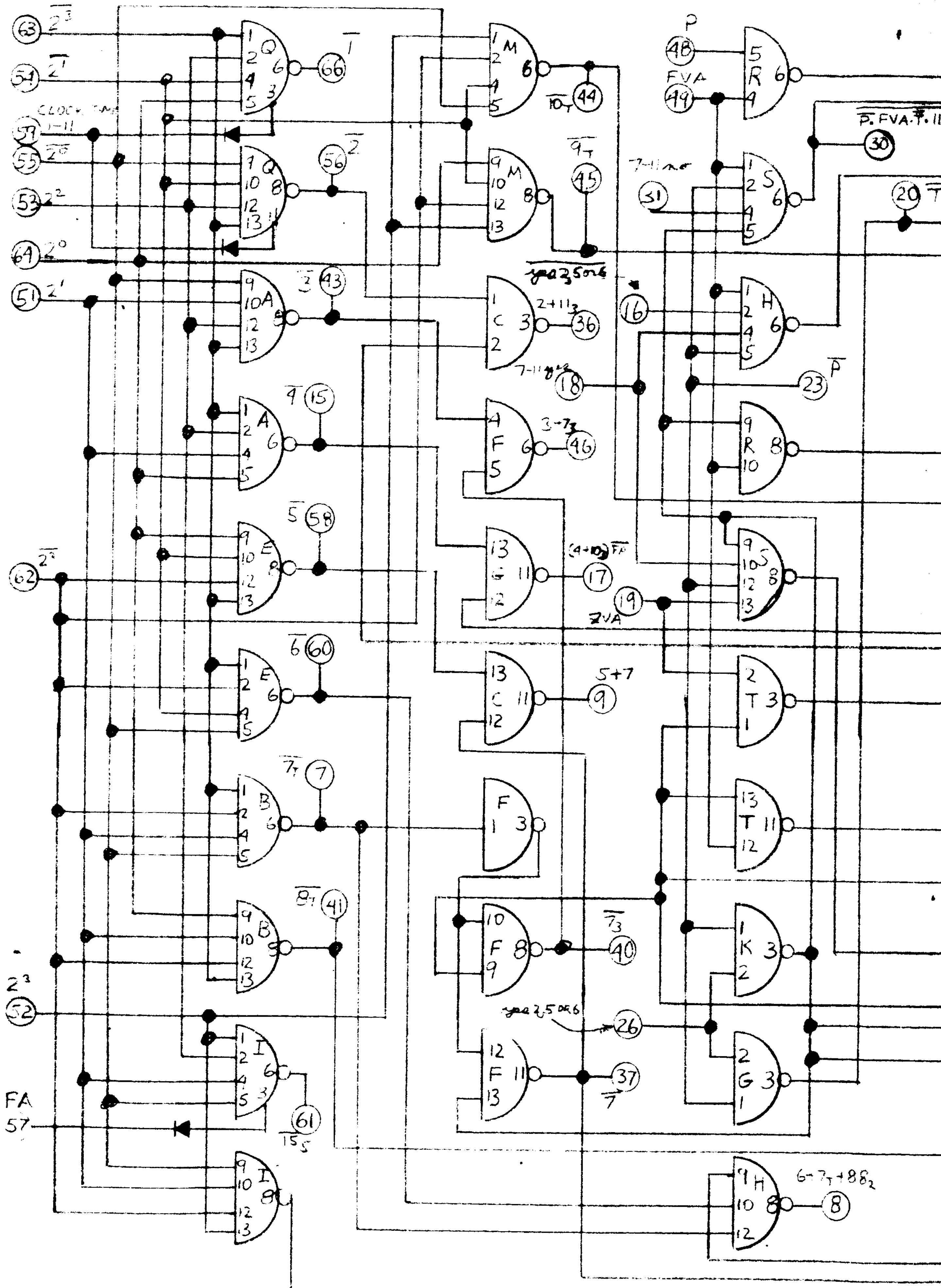
NOTE 1 TABLE INDICATES MODULE TYPE AND LOCATION ON THE BOARD



BOARD X WAVEFORMS

FIG. 3-9

LMSC 669224



63
51
57
CLOCK
55
53
64
51
62
2 3
52
FA
57
66
56
43
15
58
60
7
41
61
44
45
36
46
17
9
40
26
37
P
48
FVA
49
30
31
20
16
18
23
19
ZVA
6-7T+88₂
8
5+7
Q
M
R
S
H
A
C
F
G
E
T
B
K
I

LMSC
669224

BOARD V

TIMING BOARD "2

FIG. 3-10

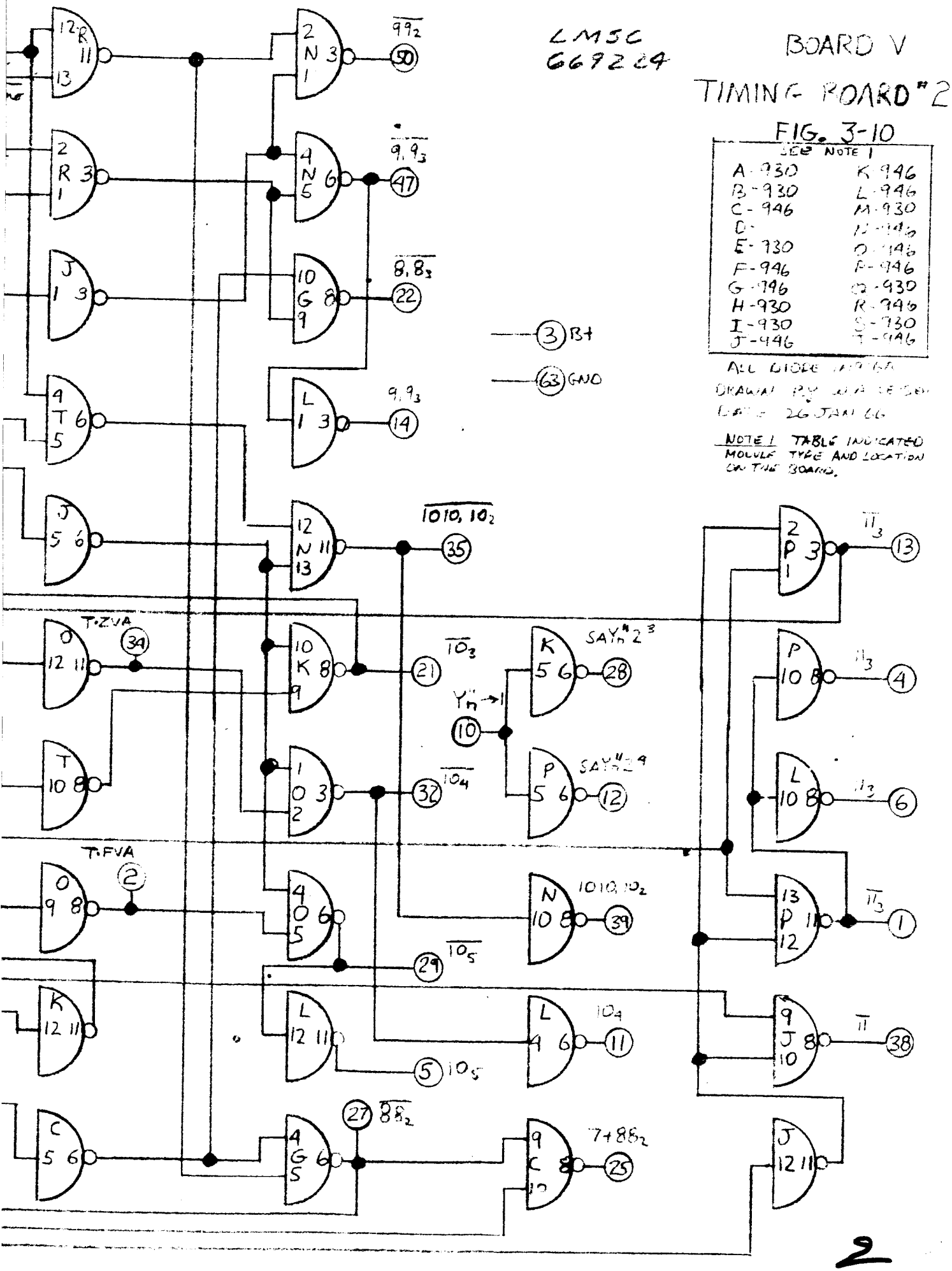
SEE NOTE 1	
A - 930	K 946
B - 930	L - 946
C - 946	M - 930
D -	N - 946
E - 930	O - 946
F - 946	P - 946
G - 946	Q - 930
H - 930	R - 946
I - 930	S - 930
J - 946	T - 946

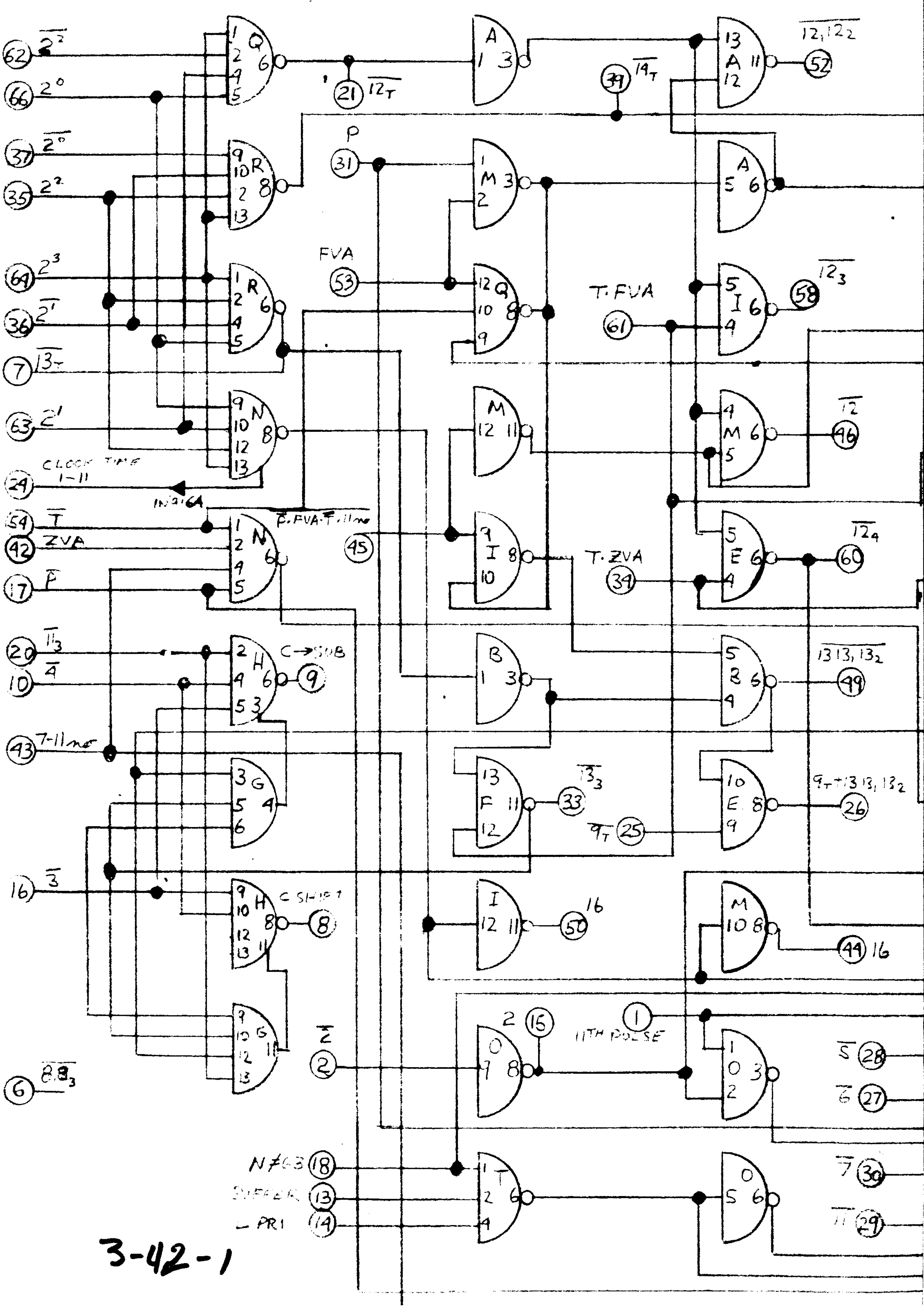
ALL DIODE [illegible]

DRAWN BY [illegible]

DATE 26 JAN 66

NOTE 1 TABLE INDICATED MODULE TYPE AND LOCATION ON THE BOARD.





2²
2⁰
2³
2¹
13_T
CLOCK TIME 1-11
T
ZVA
P
11₃
4
7-11 no
3
8.8₃
12_T
14_T
P
FVA
T·FVA
T·ZVA
12₁ 12₂
12₃
12
12₄
13 13₁ 13₂
13₃
9_T + 13 13₁ 13₂
9_T
16
C→SUB
C SHIFT
2
11TH PULSE
5
6
7
11
N≠63
PR1

3-42-1

LMSC
669224

BOARD T

TIMING BOARD #3

FIG. 3-11

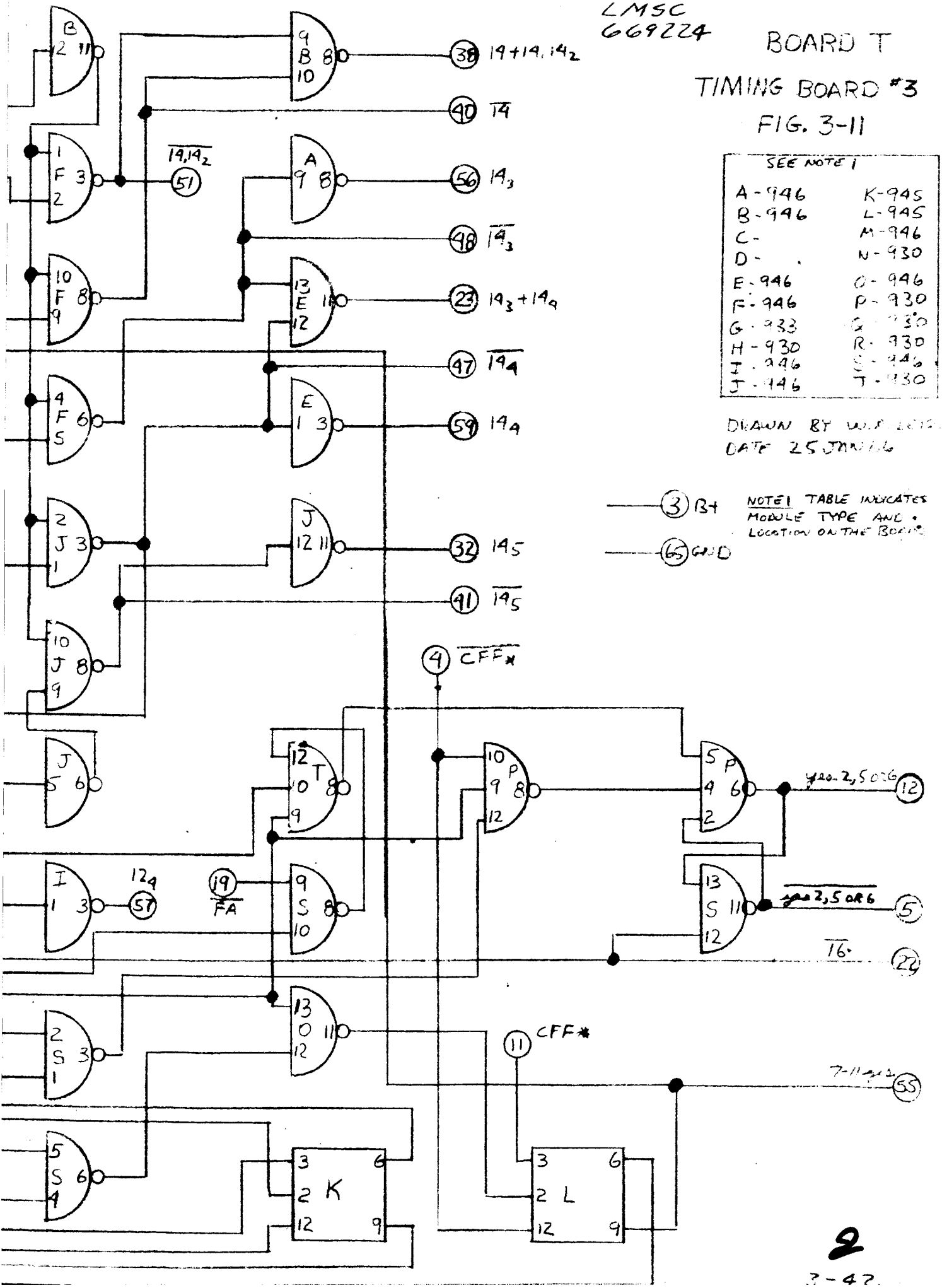
SEE NOTE 1	
A - 946	K - 945
B - 946	L - 945
C -	M - 946
D -	N - 930
E - 946	O - 946
F - 946	P - 930
G - 933	Q - 930
H - 930	R - 930
I - 946	S - 946
J - 946	T - 930

DRAWN BY [illegible]
DATE 25 JAN 66

(3) B+

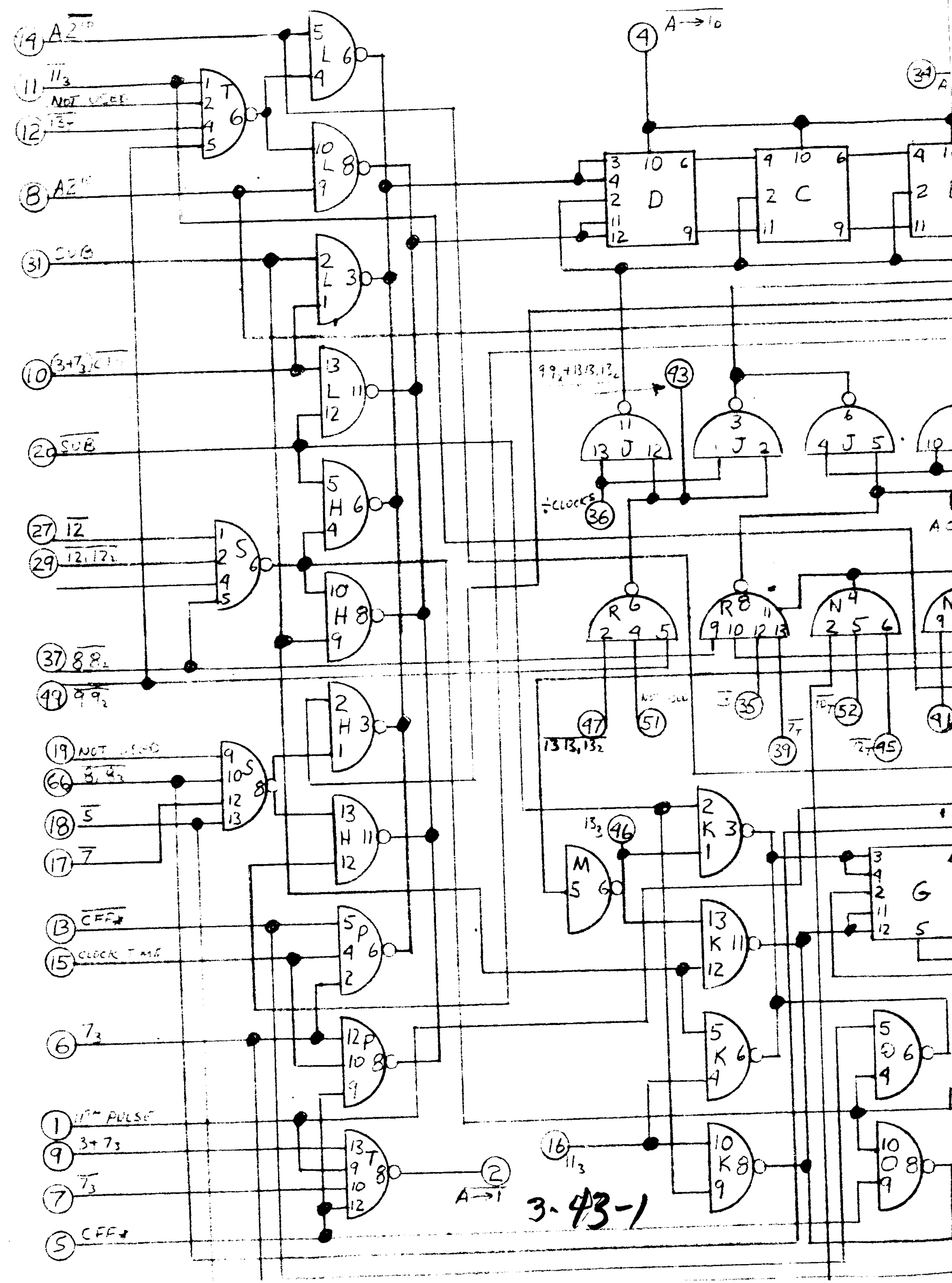
(65) GND

NOTE 1 TABLE INDICATES MODULE TYPE AND LOCATION ON THE BOARD



2

3-42



A→10
NOT USED
SUB
÷CLOCKS
CLOCK TIME
11th PULSE
3+7₃
A→1
3-43-1

LMSC
6G9224

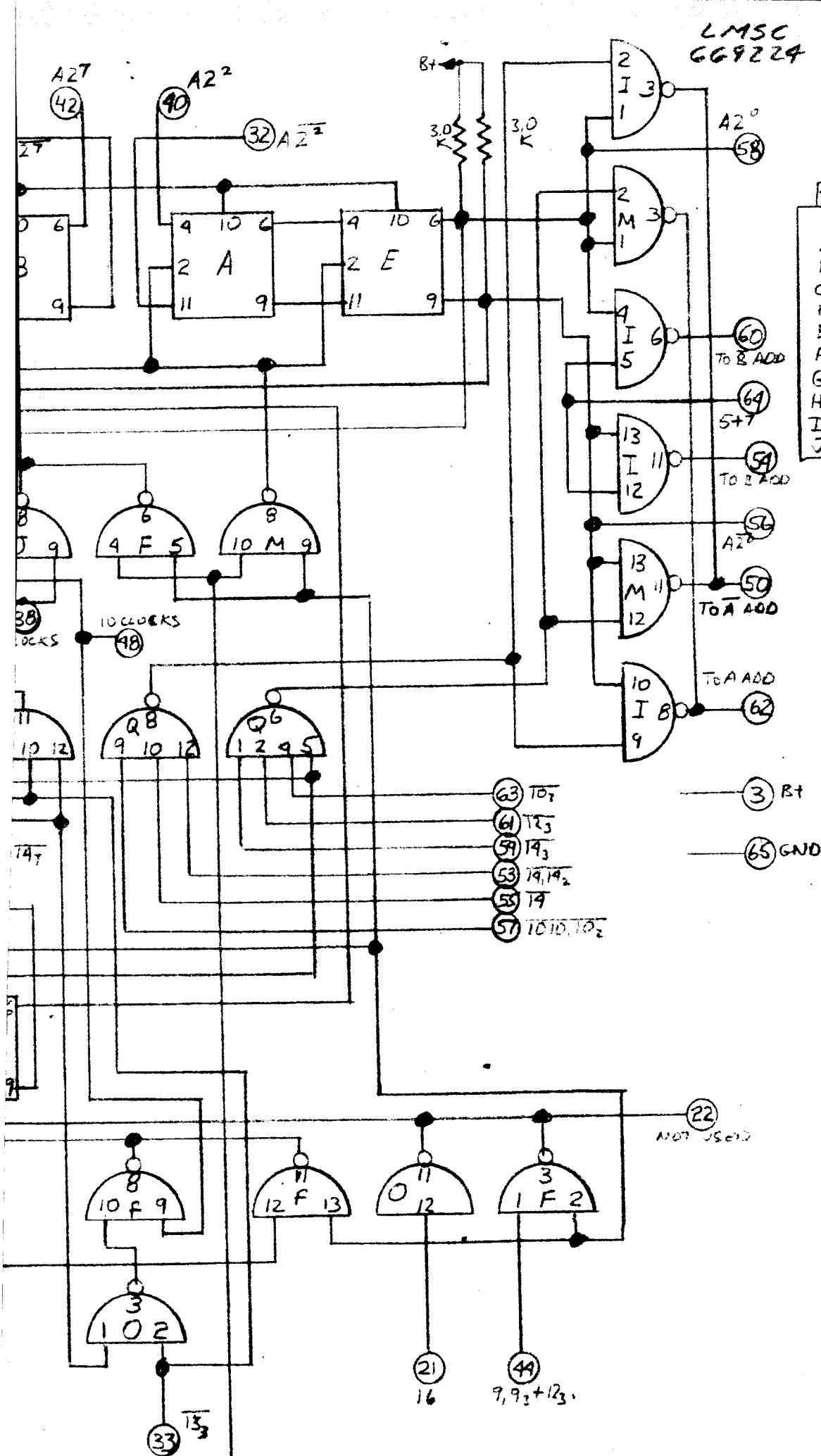
FIG. 3-12
BOARD R
PARTS OF
REGISTER A

SEE NOTE 1

A-945	K-SW946
B-945	L-SW946
C-945	M-SW946
D-945	N-SW933
E-PL945	O-SW946
F-SW946	P-930
G-945	Q-930
H-946	R-930
I-946	S-930
J-SW946	T-930

DRAWN BY W. A. LEE
DATE 29 JAN 66

NOTE 1 TABLE INDICATES MODULE TYPE AND LOCATION ON THE BOARD.



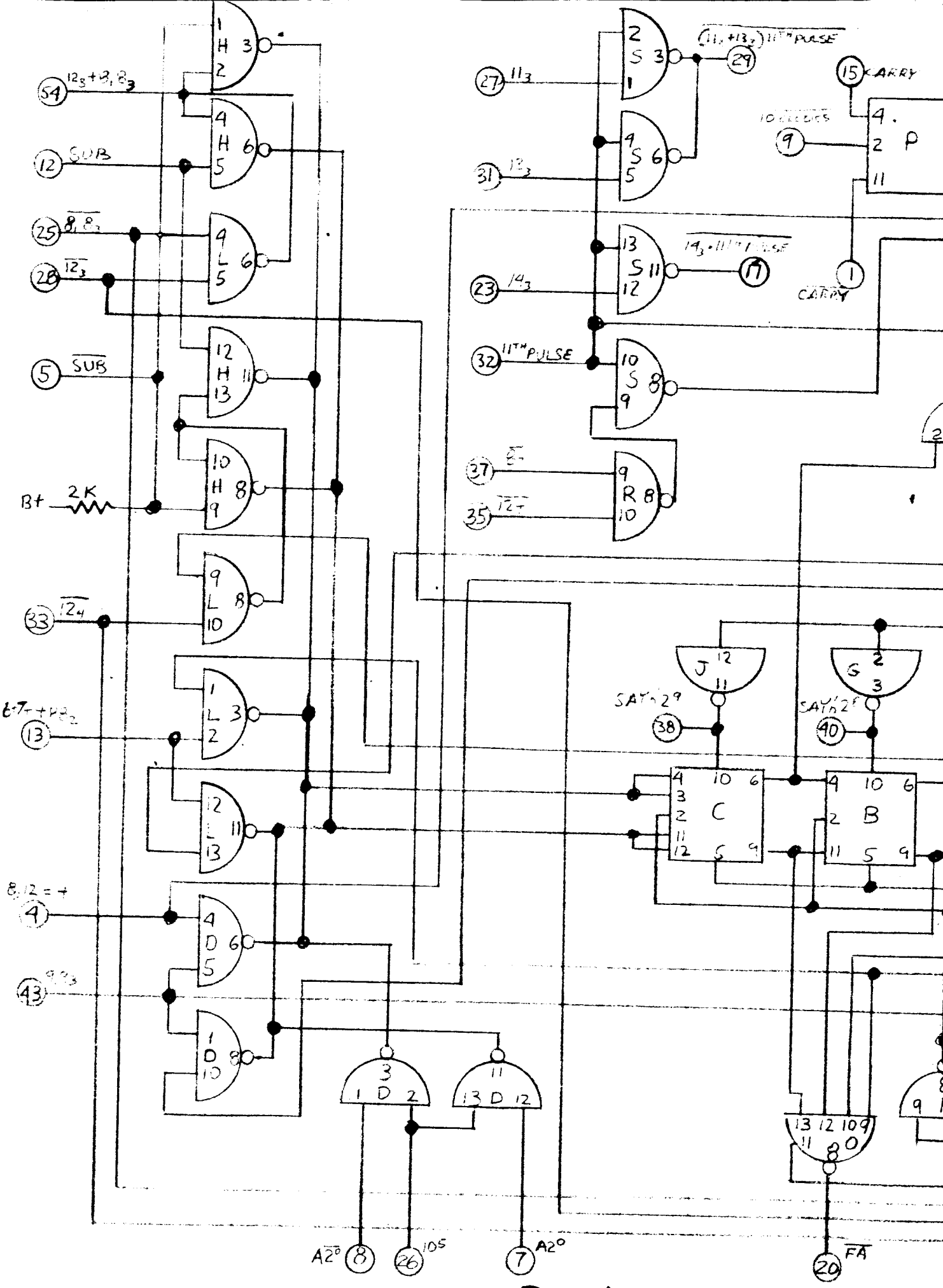
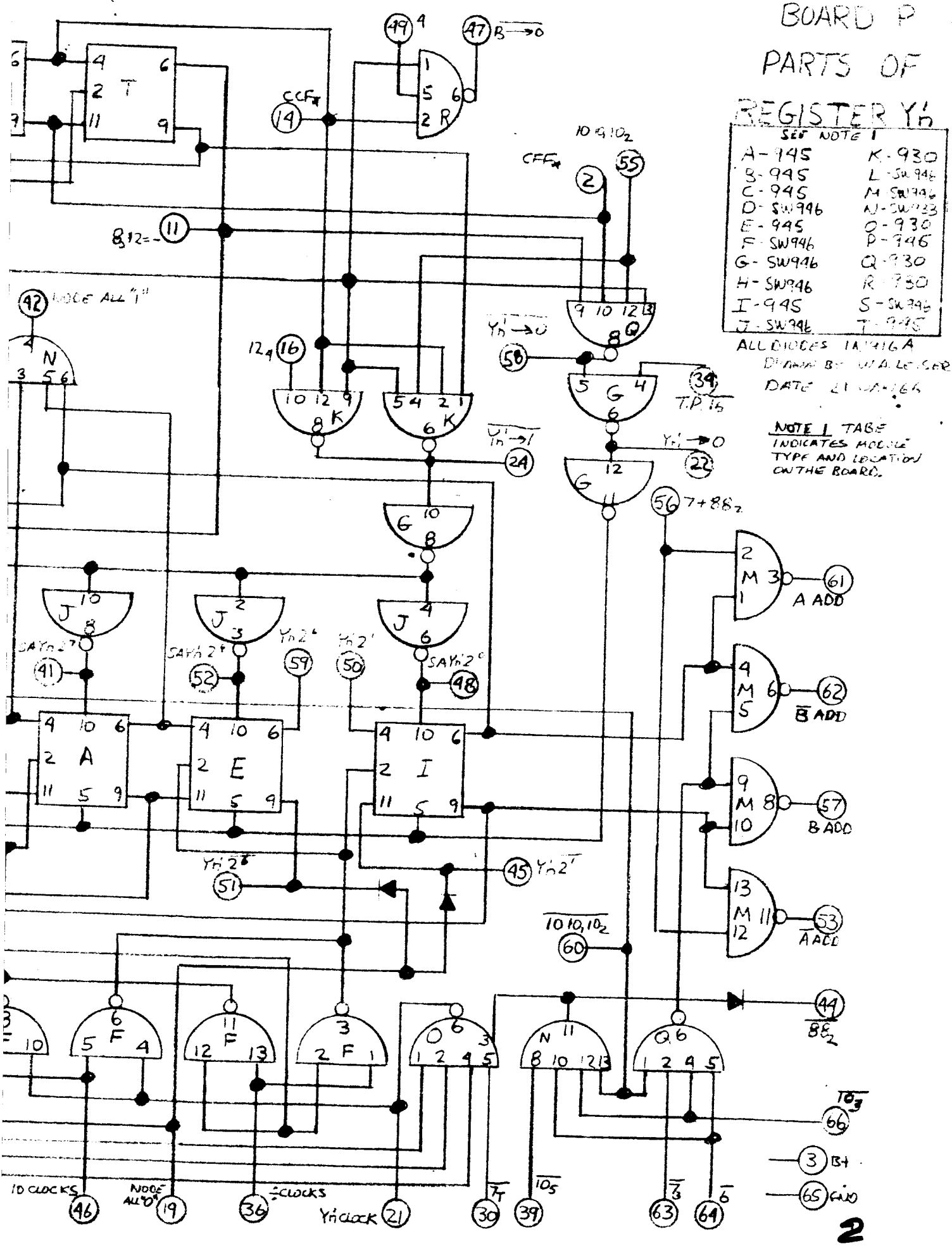

54
12
SUB
25
28
5
SUB
B+
2K
33
13
4
43
27
31
23
32
37
35
15
CARRY
9
1
CARRY
38
40
SAT'n 2^9
SAT'n 2^8
A2^0
8
26
7
A2^0
20
FA
3-44-1

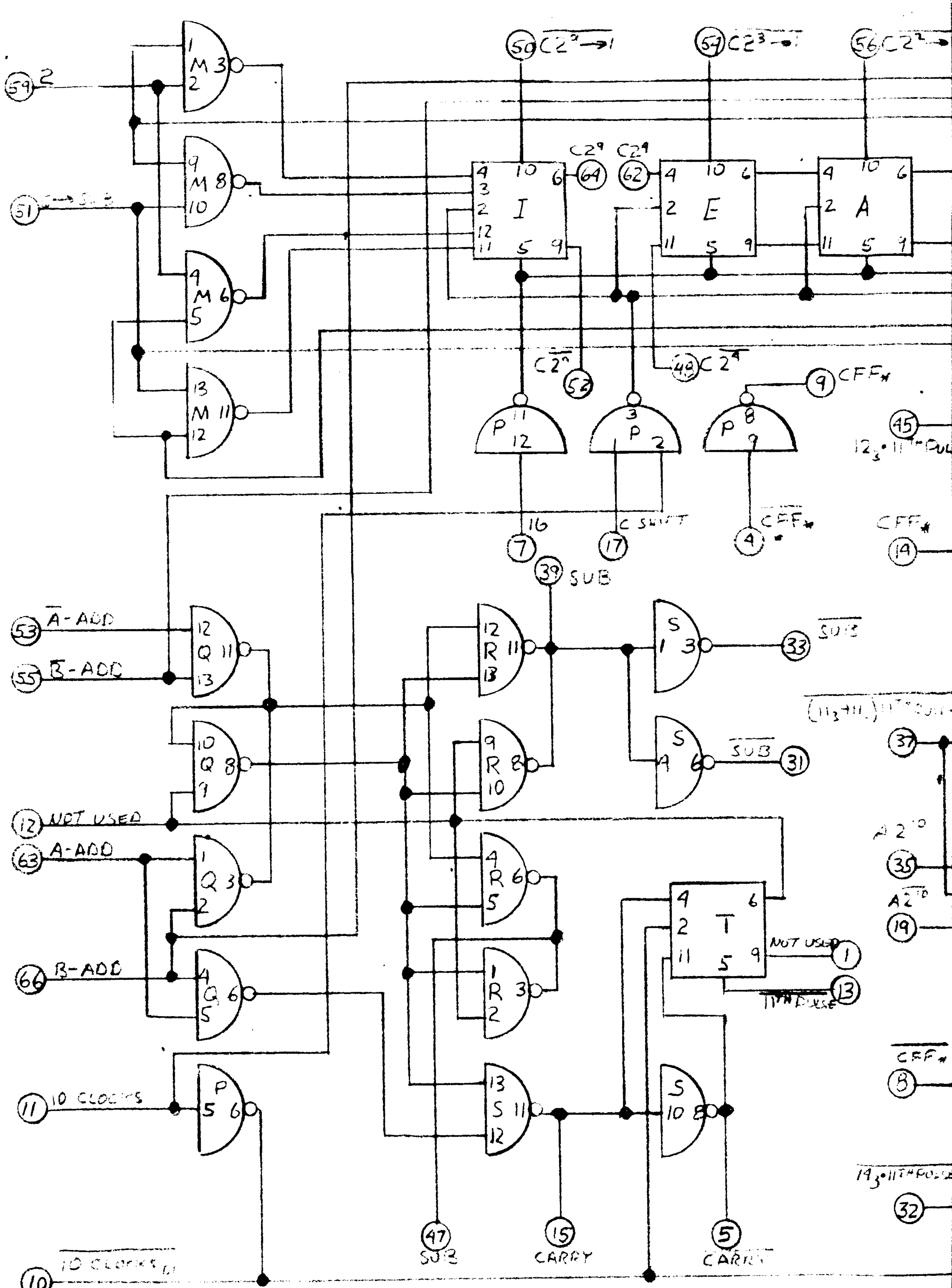
FIG. 3-13
BOARD P
PARTS OF
REGISTER Yn

SEE NOTE 1	
A-945	K-930
B-945	L-SW946
C-945	M-SW946
D-SW946	N-SW933
E-945	O-930
F-SW946	P-946
G-SW946	Q-930
H-SW946	R-930
I-945	S-SW946
J-SW946	T-945

ALL DIODES 1N916A
DRAWN BY W.A. LEISER
DATE 21 JAN 66

NOTE 1 TABE INDICATES MODULE TYPE AND LOCATION ON THE BOARD.

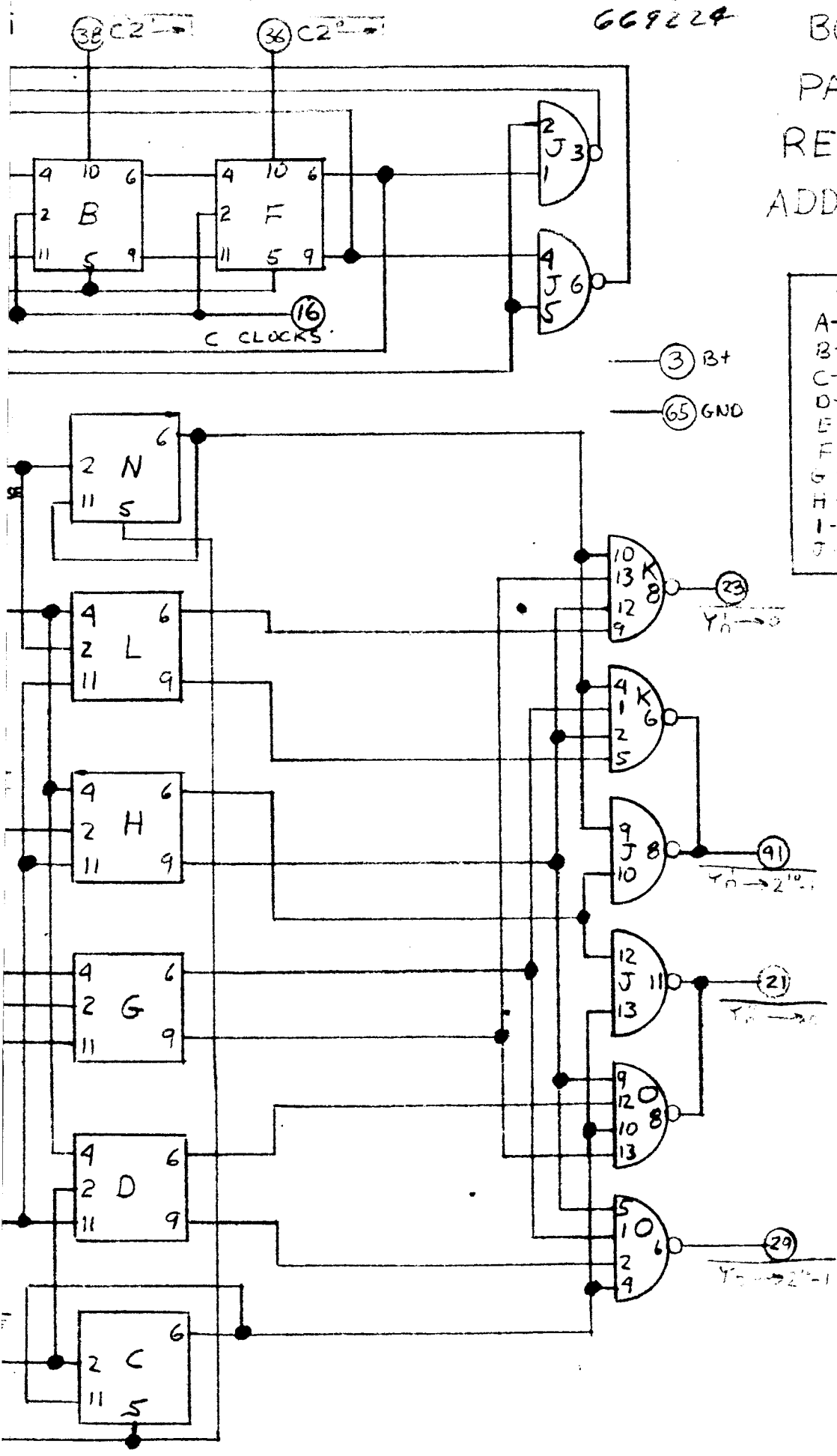




2
C 2⁹→1
C 2³→
C 2²→
C2⁹
C2⁹
I
E
A
C 2⁹
C 2⁴
CFF*
CFF*
CFF*
C SHIFT
SUB
A-ADD
B-ADD
NOT USED
A-ADD
B-ADD
10 CLOCKS
SUB
SUB
SUB
T
NOT USED
11TH PULSE
SUB
CARRY
CARRY
10 CLOCKS
A 2¹⁰
A 2¹⁰

LMSC
6G9224

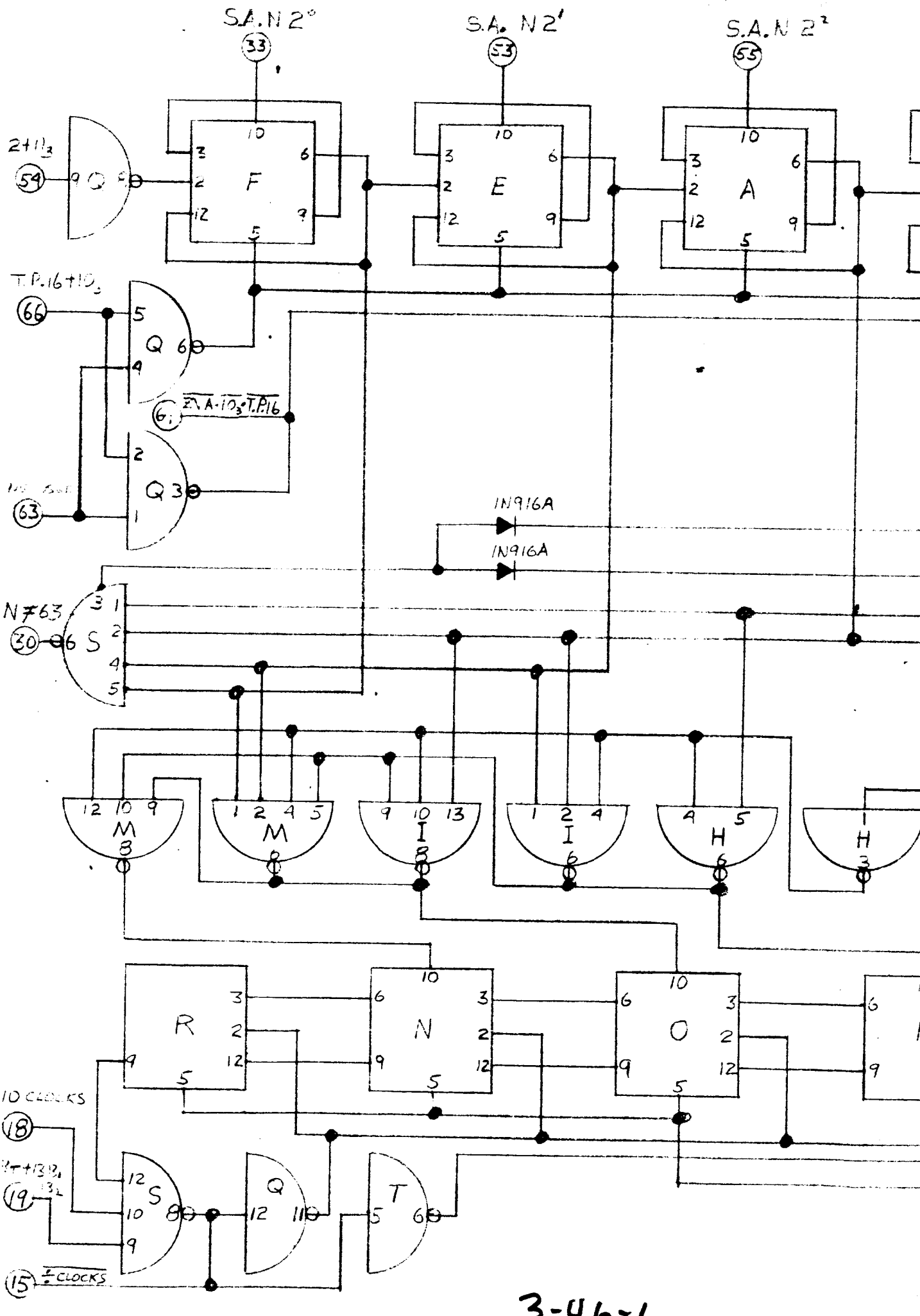
FIG. 3-14
BOARD M
PARTS OF
REGISTER C
ADDER-SUBTRACTER



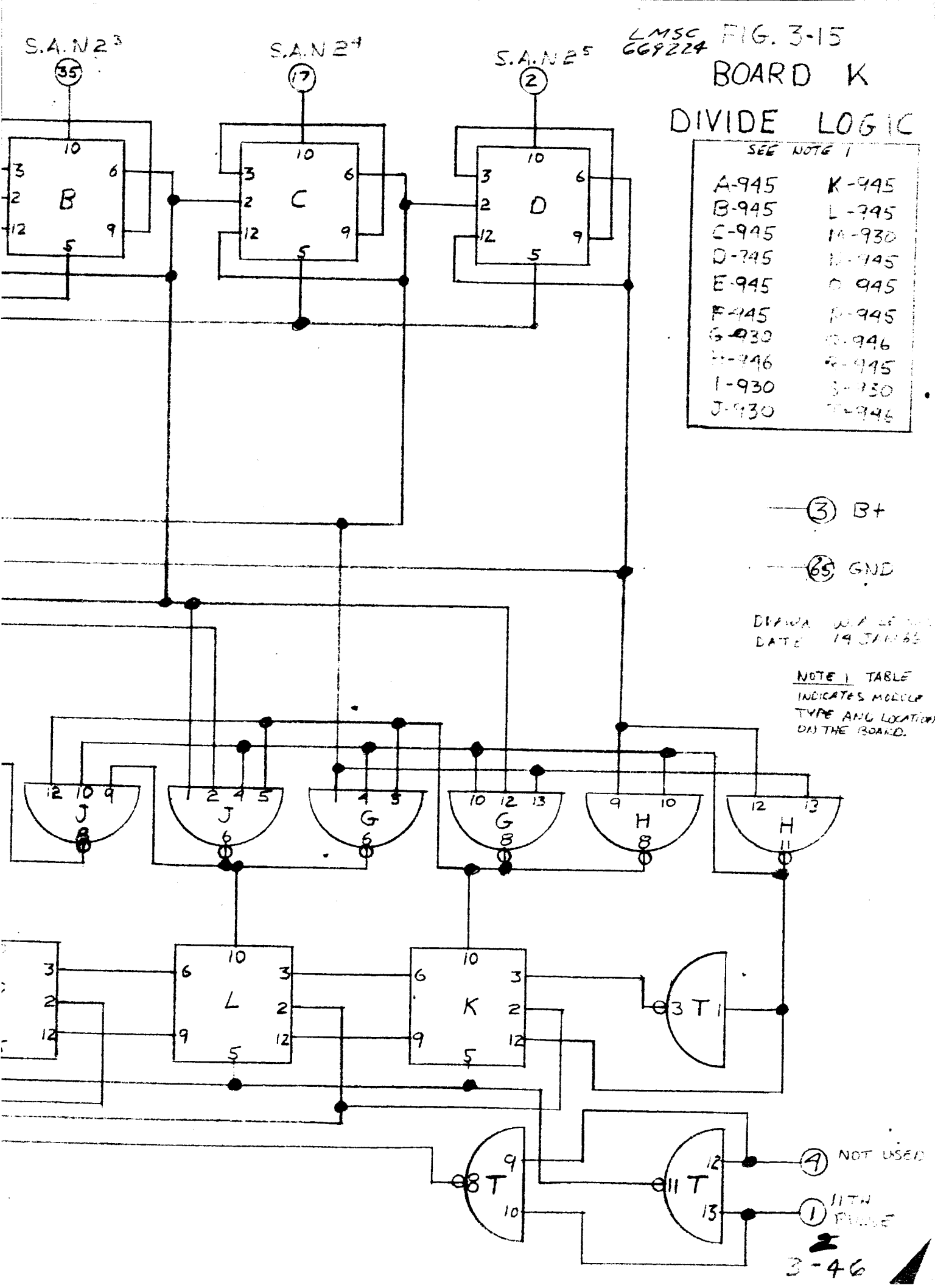
SEE NOTE 1	
A-945	K-930
B-945	L-945
C-945	M-946
D-945	N-945
E-945	O-930
F-945	P-946
G-945	Q-946
H-945	R-946
I-945	S-946
J-946	T-945

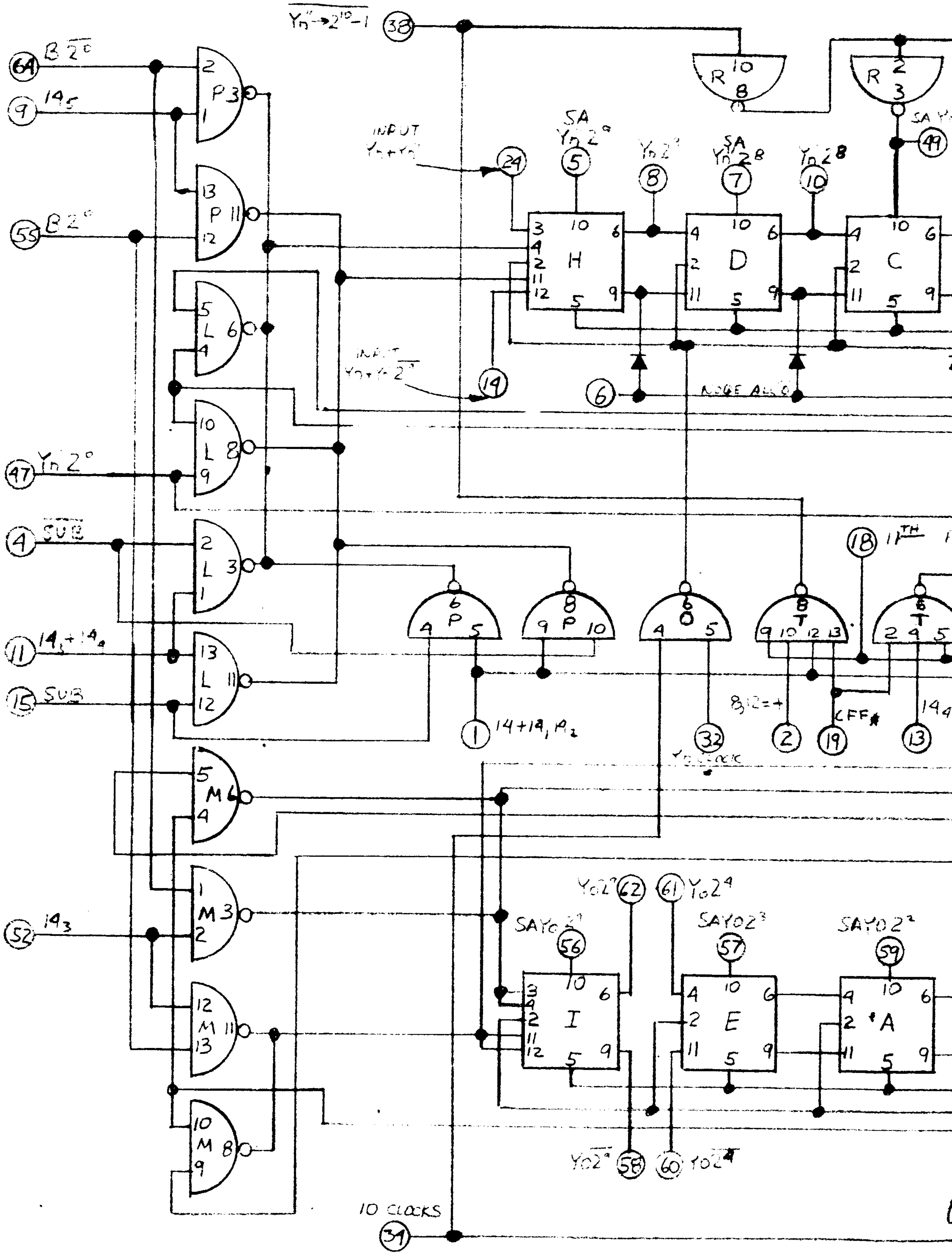
DRAWN BY W.A. LEESE
DATE 21 JAN 66

NOTE 1 TABLE INDICATES MODULE TYPE AND LOCATION ON THE BOARD.


S.A. N 2°
S.A. N 2'
S.A. N 2²
33
53
55
F
E
A
2+11₃
54
T.P.16+10₃
66
Q 6
Q 3
63
IN916A
IN916A
N≠63
30
S
M
M
I
I
H
H
R
N
O
10 CLOCKS
18
19
S
Q
T
15
÷ CLOCKS

3-46-1


S.A.N2³
S.A.N2⁴
S.A.N2⁵
LMSC 669224
FIG. 3-15
BOARD K
DIVIDE LOGIC
SEE NOTE 1
A-945
B-945
C-945
D-945
E-945
F-945
G-930
H-946
I-930
J-930
K-945
L-945
M-930
N-945
O-945
P-945
Q-946
R-945
S-930
T-946
B+
GND
NOTE 1 TABLE INDICATES MODULE TYPE AND LOCATION ON THE BOARD.
NOT USED
3-46



10 CLOCKS
NODE ALL 0
SAYo2³
SAYo2²
SUB
H
D
C
I
E
A

FIG. 3-16 BOARD H PARTS OF REGISTERS

Y_n'' & Y_0

SEE NOTE 1

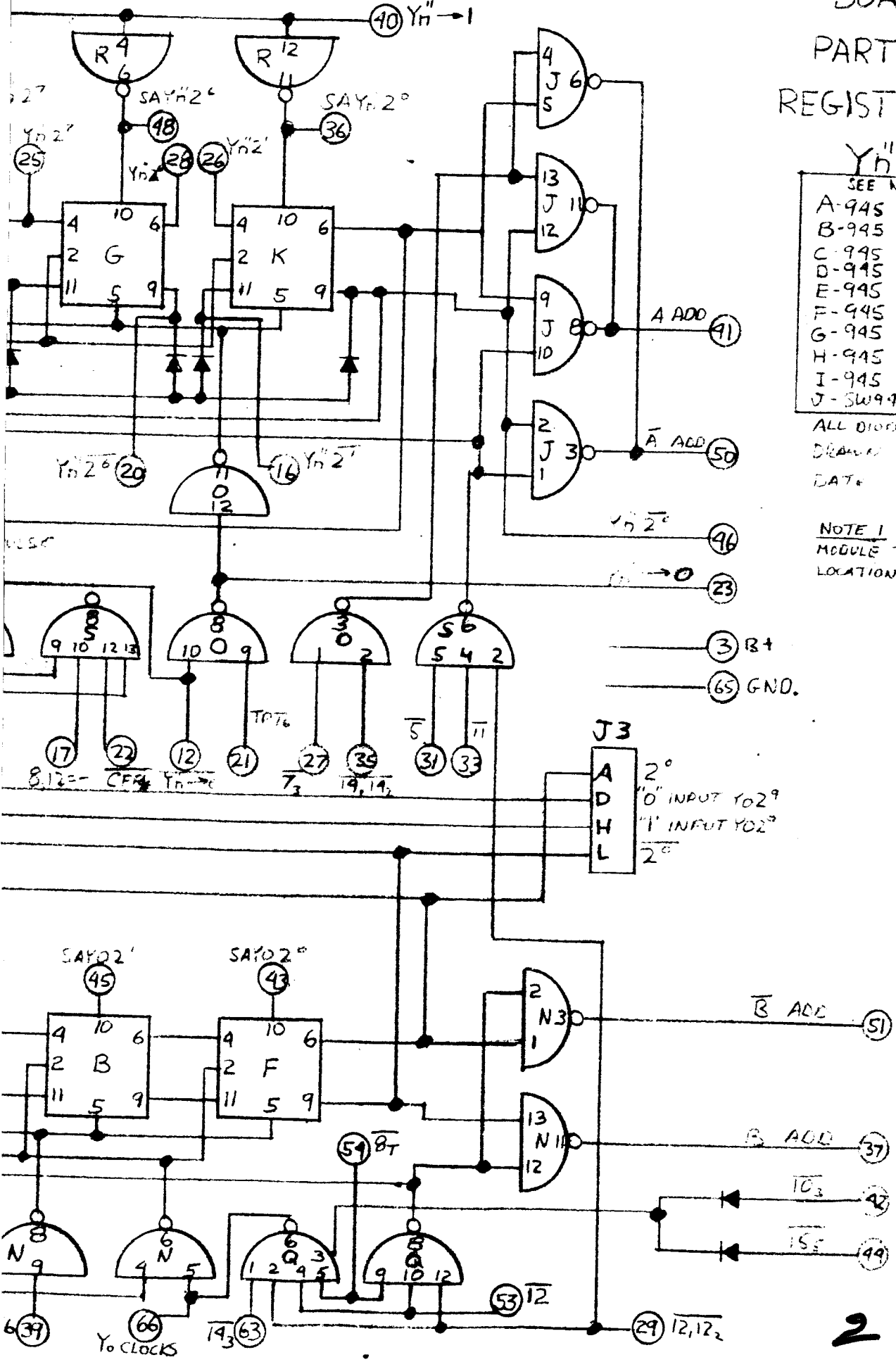
A-945	K-945
B-945	L-SW946
C-945	M-SW946
D-945	N-SW946
E-945	O-SW946
F-945	P-SW946
G-945	Q-930
H-945	R-SW946
I-945	S-930
J-SW946	T-930

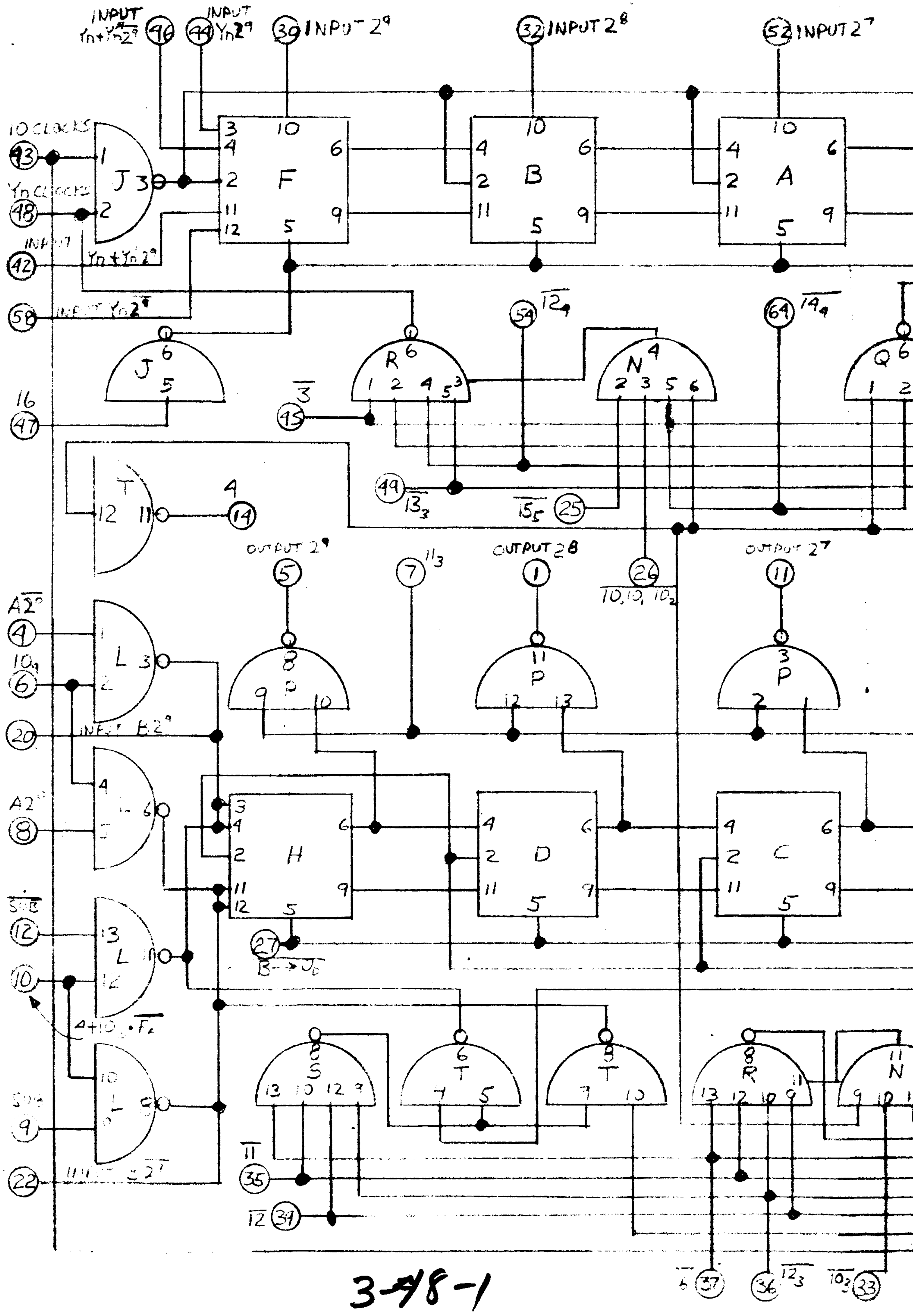
ALL DIODES 1N916A

DRAWN BY W.A. [illegible]

DATE 20 JAN 66

NOTE 1 TABLE INDICATES MODULE TYPE AND LOCATION ON THE BOARD.





INPUT Yn+Yn2⁹
INPUT Yn2⁹
INPUT 2⁹
INPUT 2⁸
INPUT 2⁷
10 CLOCKS
Yn CLOCKS
INPUT Yn+Yn2⁹
INPUT Yn2⁹
F
B
A
J
R
N
Q
T
L
H
D
C
P
S
OUTPUT 2⁹
OUTPUT 2⁸
OUTPUT 2⁷
INPUT B2⁹
A2⁰
B→J0

3-48-1

LMSC 669224

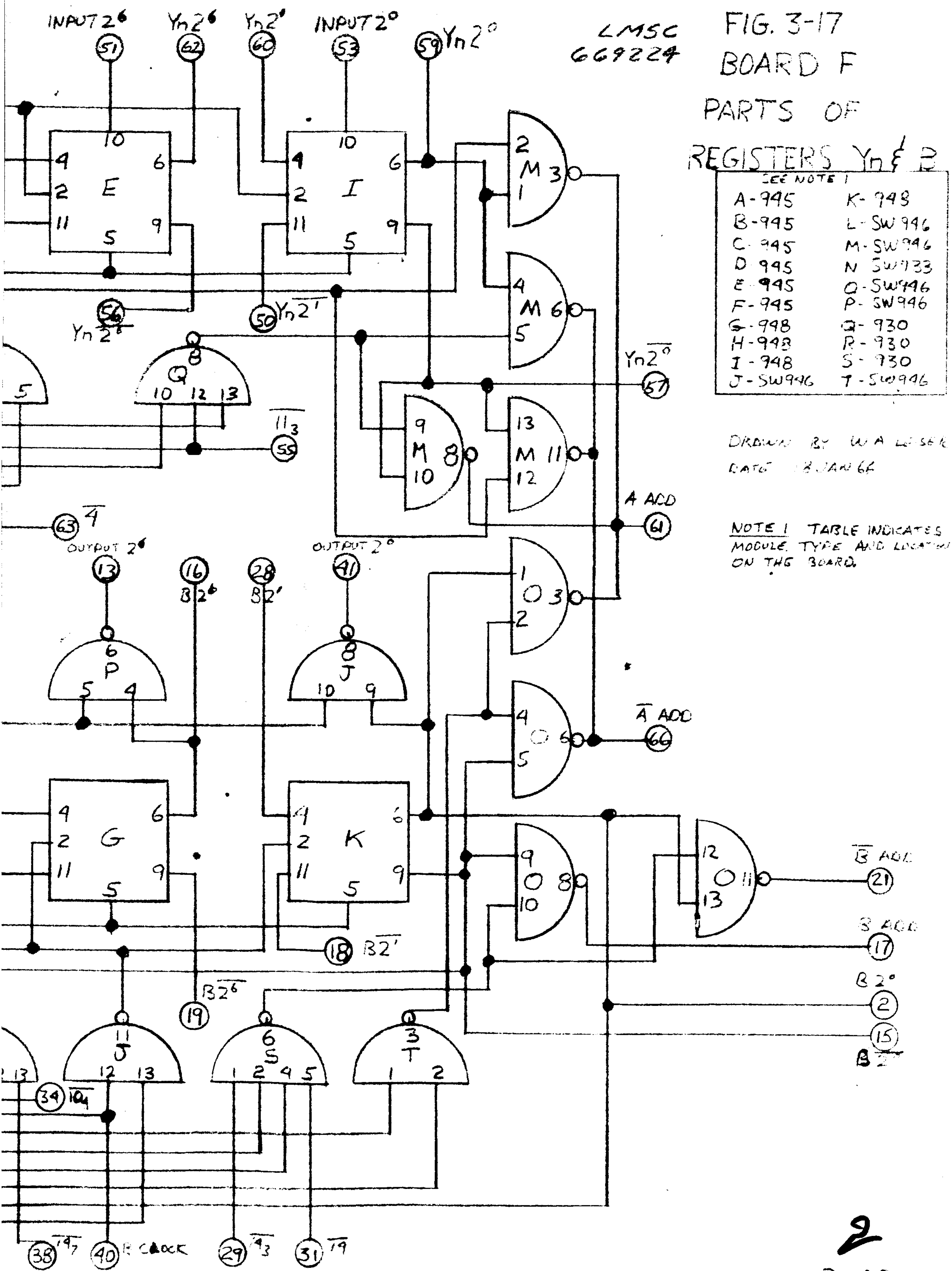
FIG. 3-17 BOARD F PARTS OF REGISTERS Yn & B

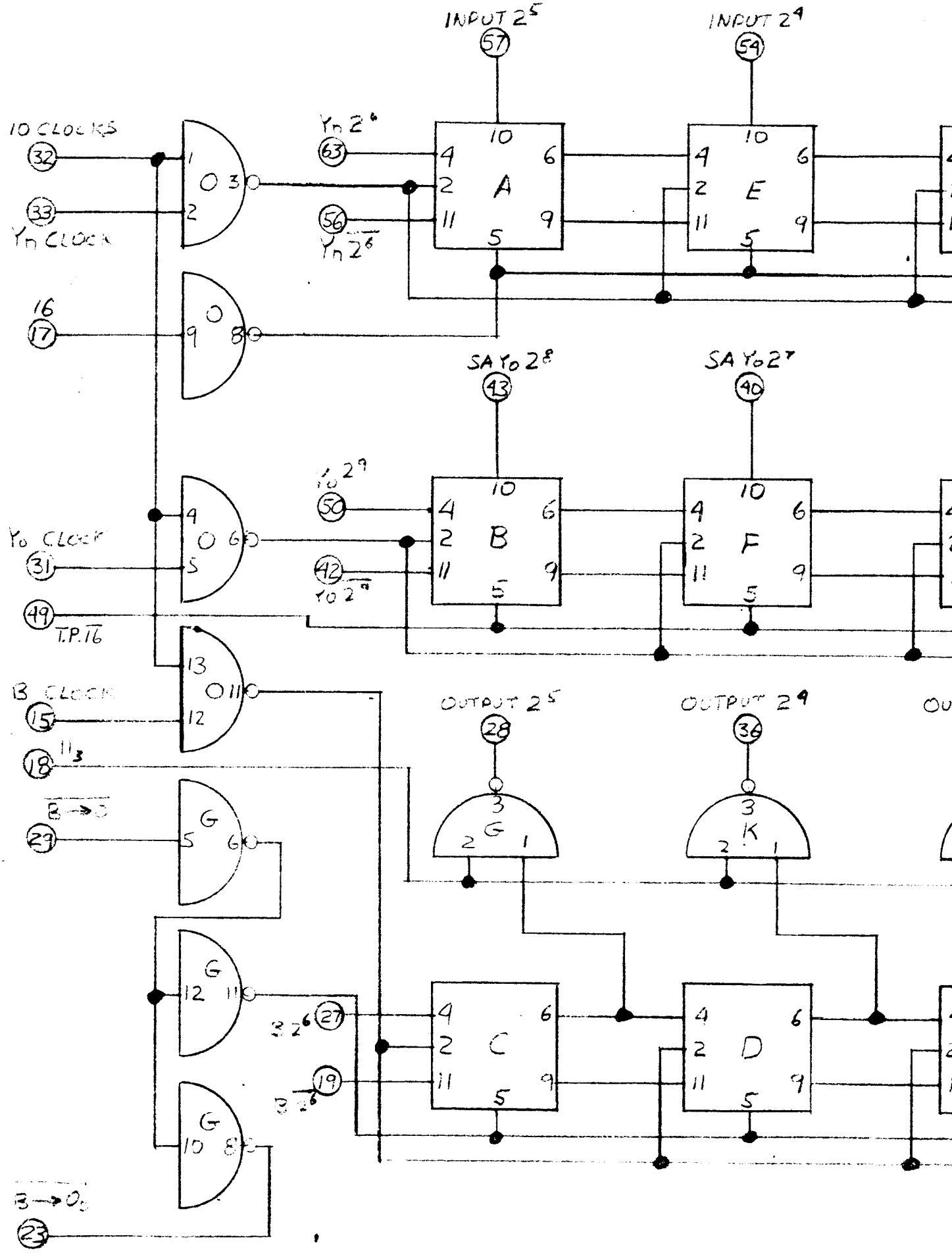
SEE NOTE 1

A-945	K-948
B-945	L-SW946
C-945	M-SW946
D-945	N-SW933
E-945	O-SW946
F-945	P-SW946
G-948	Q-930
H-948	R-930
I-948	S-930
J-SW946	T-SW946

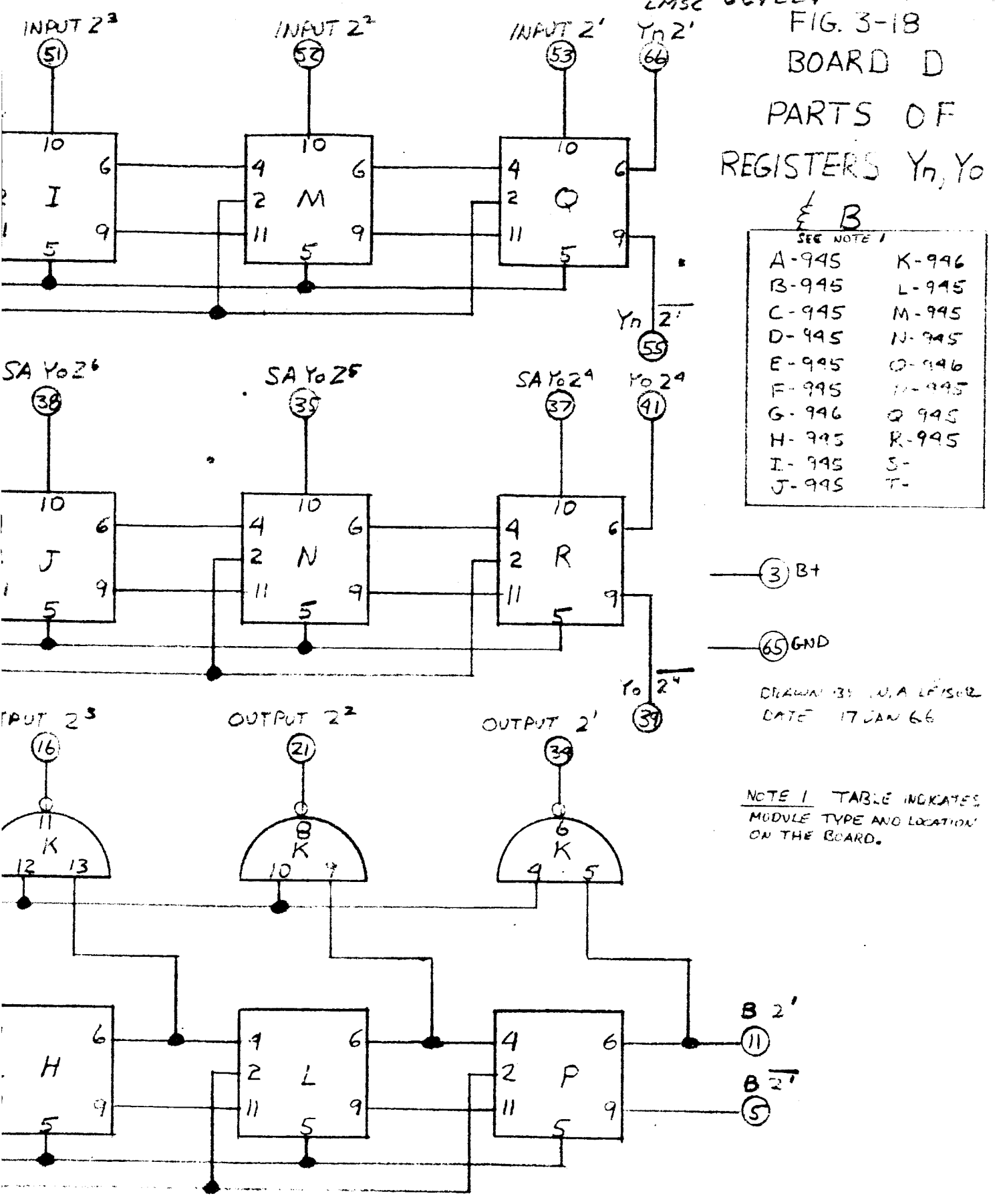
DRAWN BY W A LESER
DATE 13 JAN 66

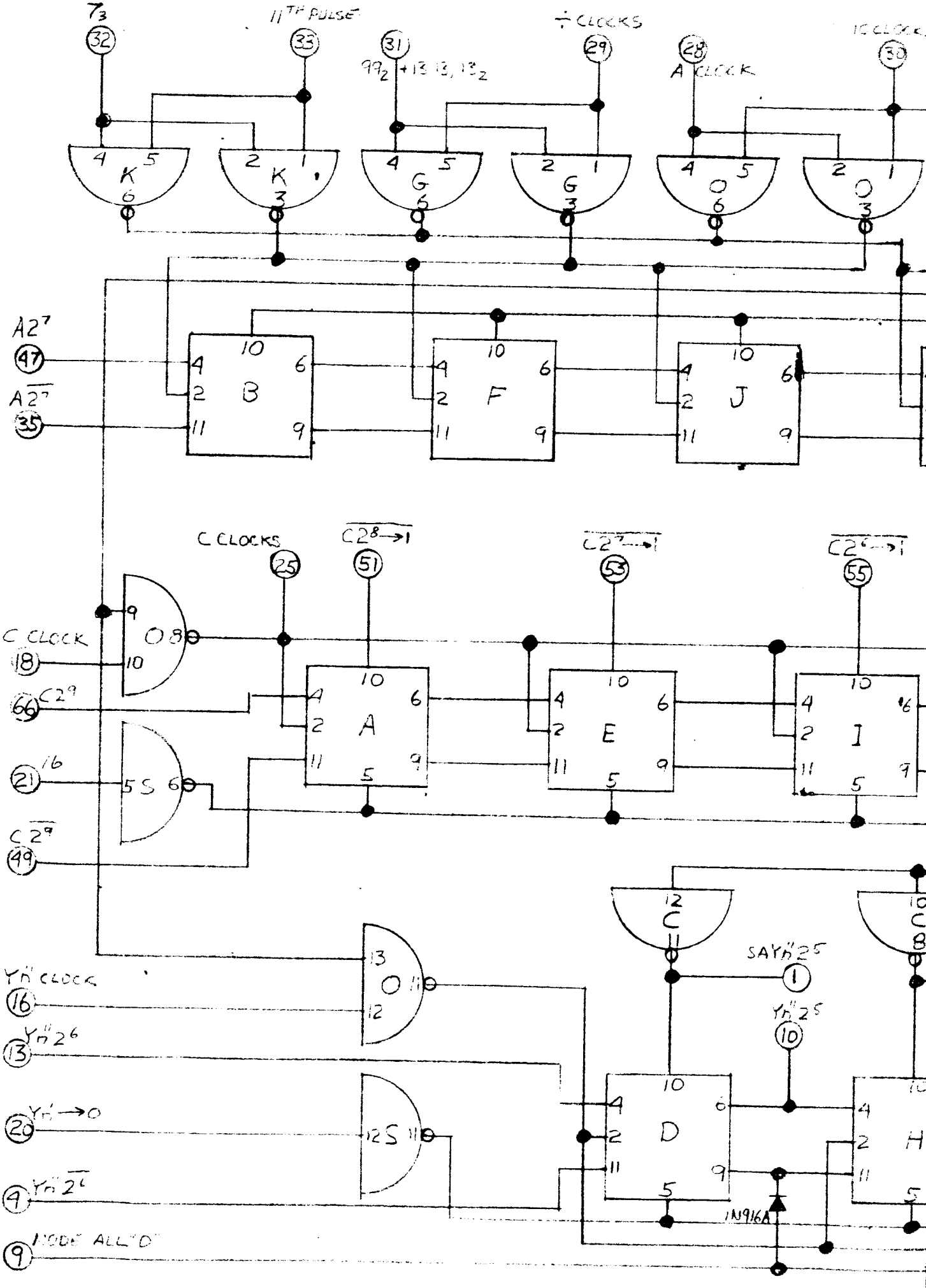
NOTE 1 TABLE INDICATES MODULE TYPE AND LOCATION ON THE BOARD.



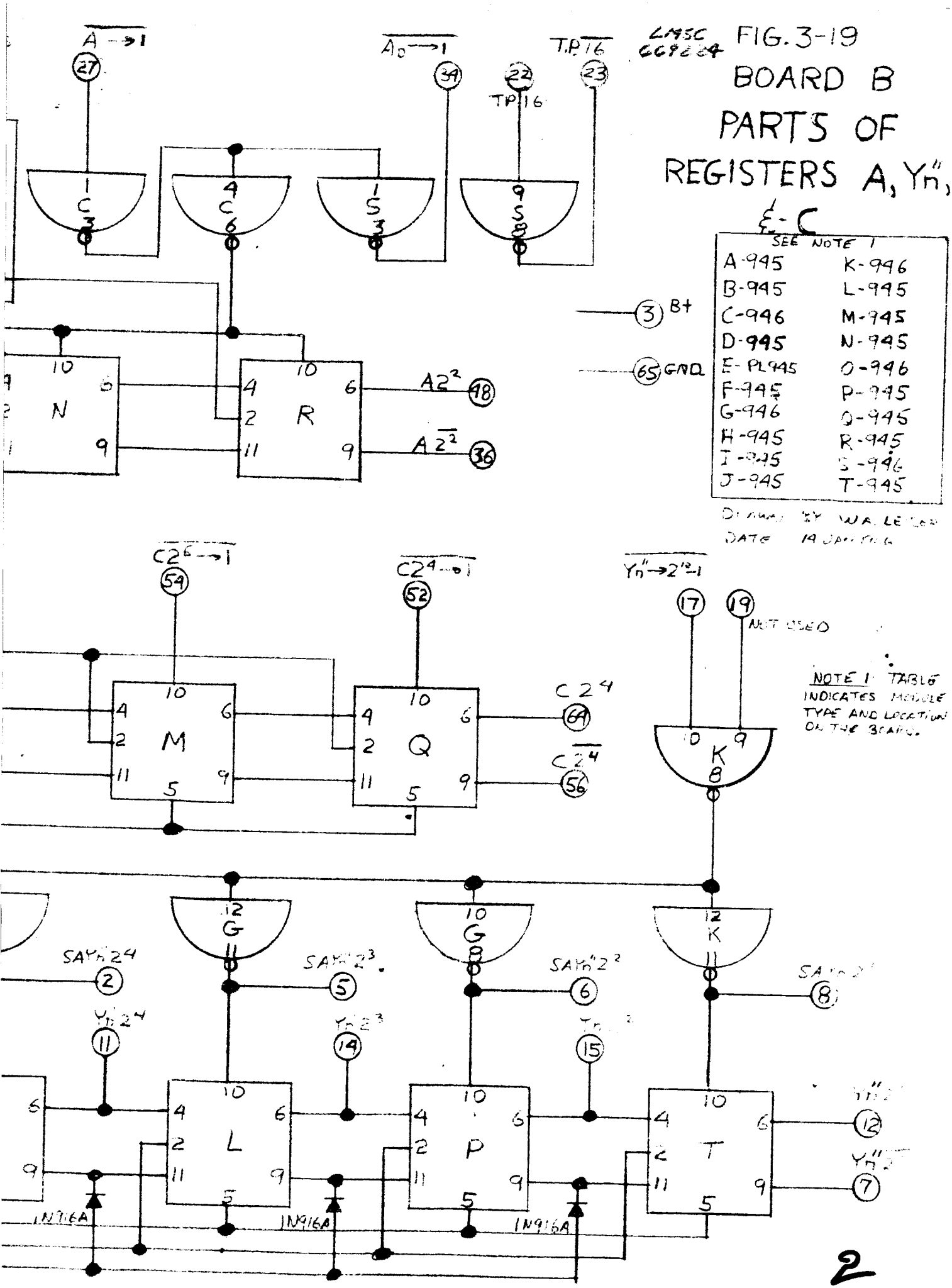

INPUT 2⁵
INPUT 2⁴
10 CLOCKS
Yn 2⁶
Yn CLOCK
Yn 2⁶
SA Yo 2⁸
SA Yo 2⁷
Yo 2⁹
Yo CLOCK
Yo 2⁹
T.P.16
B CLOCK
OUTPUT 2⁵
OUTPUT 2⁴
B → 0
3 → 0₂
A
B
C
D
E
F
G
K

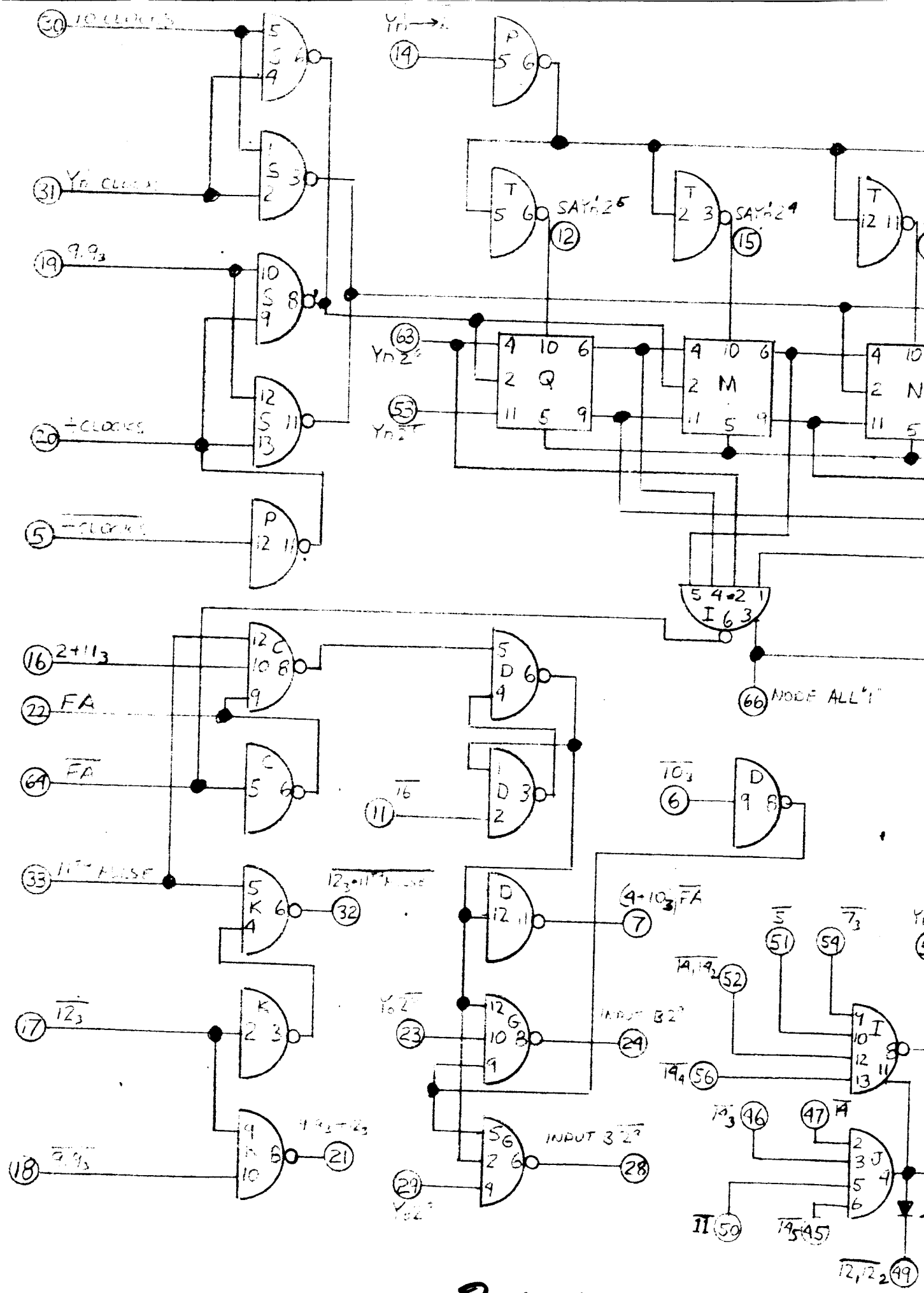
3-49-1


INPUT 2³
INPUT 2²
INPUT 2¹
Yn 2¹
FIG. 3-18
BOARD D
PARTS OF
REGISTERS Yn, Yo
& B
SEE NOTE 1
A-945
B-945
C-945
D-945
E-945
F-945
G-946
H-945
I-945
J-945
K-946
L-945
M-945
N-945
O-946
P-945
Q-945
R-945
S-
T-
Yn 2¹
SA Yo 2⁶
SA Yo 2⁵
SA Yo 2⁴
Yo 2⁴
B+
GND
Yo 2⁴
DRAWN BY W.A. LEISER
DATE 17 JAN 66
OUTPUT 2³
OUTPUT 2²
OUTPUT 2¹
NOTE 1 TABLE INDICATES MODULE TYPE AND LOCATION ON THE BOARD.
B 2¹
B 2¹


T3
32
11TH PULSE
33
31
99₂ +13 13, 13₂
÷ CLOCKS
29
28
A CLOCK
30
K
G
O
A2⁷
47
A2⁷
35
B
F
J
C CLOCKS
25
C28→1
51
C27→1
53
C26→1
55
C CLOCK
18
C29
66
16
21
C29
49
A
E
I
Yn" CLOCK
16
Yn"26
13
Yn'→0
20
Yn"2⁶
9
MODE ALL "0"
9
C
D
H
SAYn"25
1
Yn"25
10
1N916A

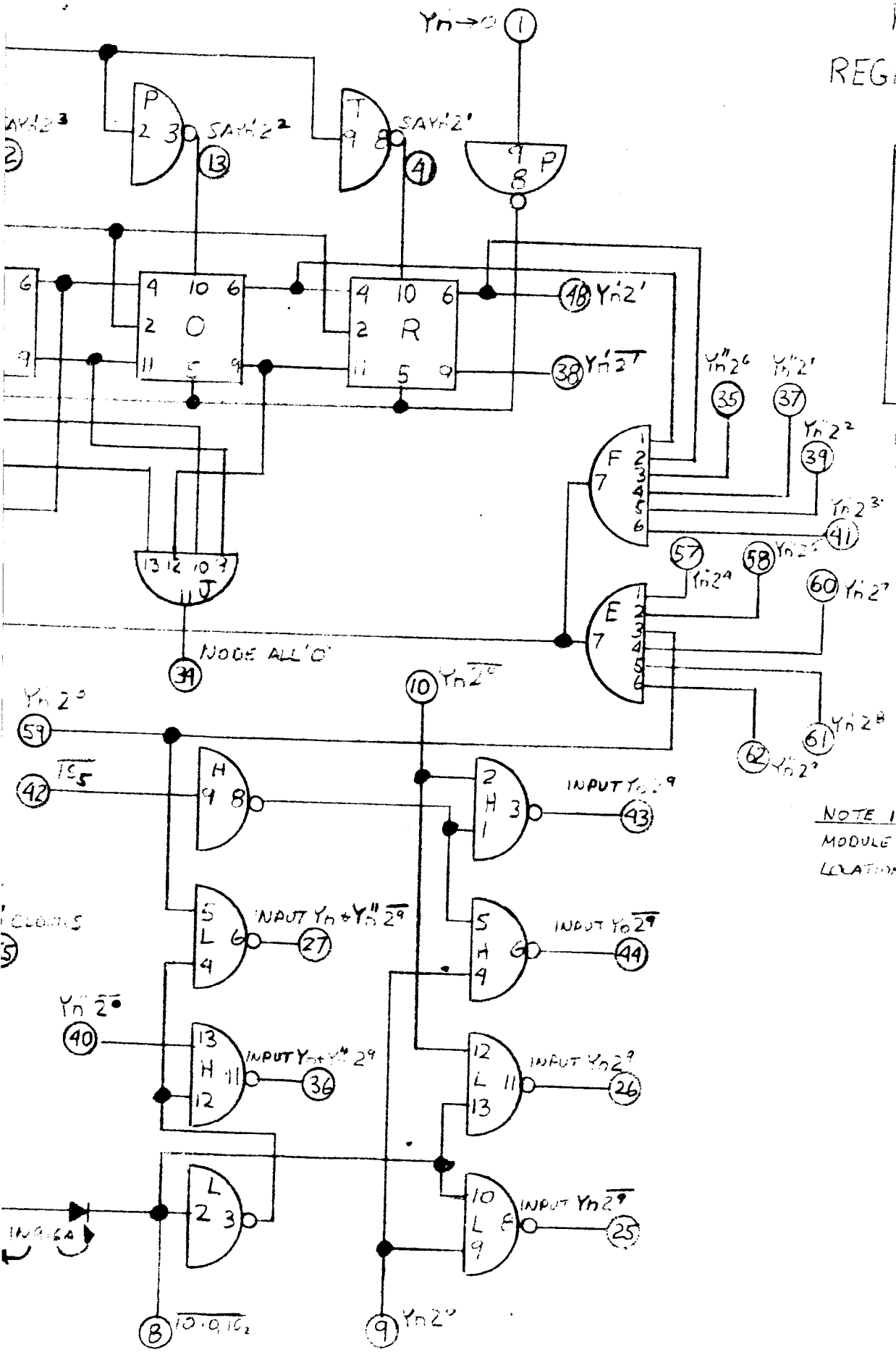
3-50-1


A → 1
A0 → 1
T.P.T6
LM5C
CC82C4
FIG. 3-19
BOARD B
PARTS OF
REGISTERS A, Yn'',
E-C
TP16
SEE NOTE 1
A-945 K-946
B-945 L-945
C-946 M-945
D-945 N-945
E-PL945 O-946
F-945 P-945
G-946 Q-945
H-945 R-945
I-945 S-946
J-945 T-945
B+
GND
A2²
A2̄²
C2⁵ → 1
C2⁴ → 1
Yn'' → 2'' - 1
NOT USED
C 2⁴
C2̄⁴
NOTE 1: TABLE INDICATES MODULE TYPE AND LOCATION ON THE BOARD.
SAYn'2⁴
SAYn'2³
SAYn'2²
Yn'2⁴
Yn'2³
IN916A
IN916A
IN916A
2
3-50


SAY n 2^5
SAY n 2^4
MODE ALL "1"
INPUT B2^3
INPUT 3 2^3

LMSC
669224

FIG. 3-20
BOARD AA
PARTS
REGISTER Yn &
MISC.



SEE NOTE 1	
A-	K-946
B-	L-946
C-930	M PL946
D-946	N PL946
E-DIODE ARRAY	O PL946
F-DIODE ARRAY	P 946
G-930	Q PL946
H-946	R PL946
I-930	S 946
J-SN933	T 946

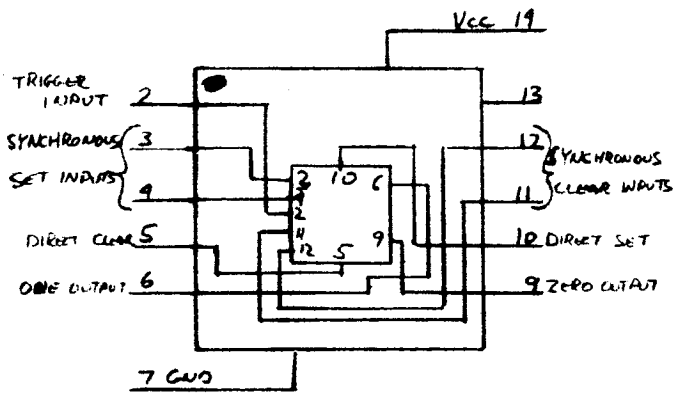
DRAWN BY AA
DATE 27 JAN

2

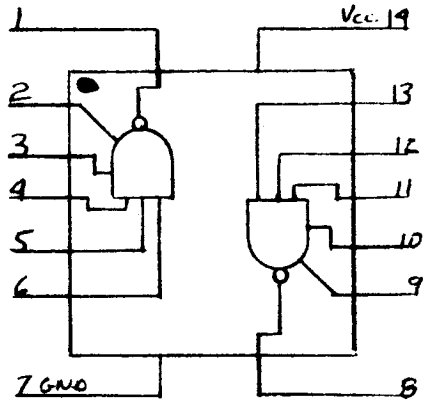
LPDTuL 9040 AND 9042
FIG. 3-21

DRAWN BY W.A. LEIGH
DATE 9 FEB 1966

9040



9042



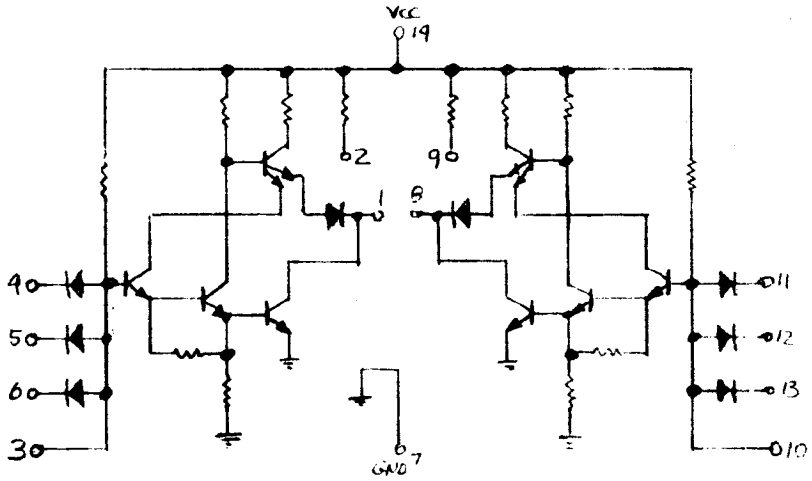
LOGIC EQUATION

$1 = \overline{4 \cdot 5 \cdot 6}$

$8 = \overline{11 \cdot 12 \cdot 13}$

NOTE

THE 9040 TRUTH TABLES ARE IDENTICAL TO THE DTuL 945 TRUTH TABLES - SEE FIG. 3-6.



SCHEMATIC 9042

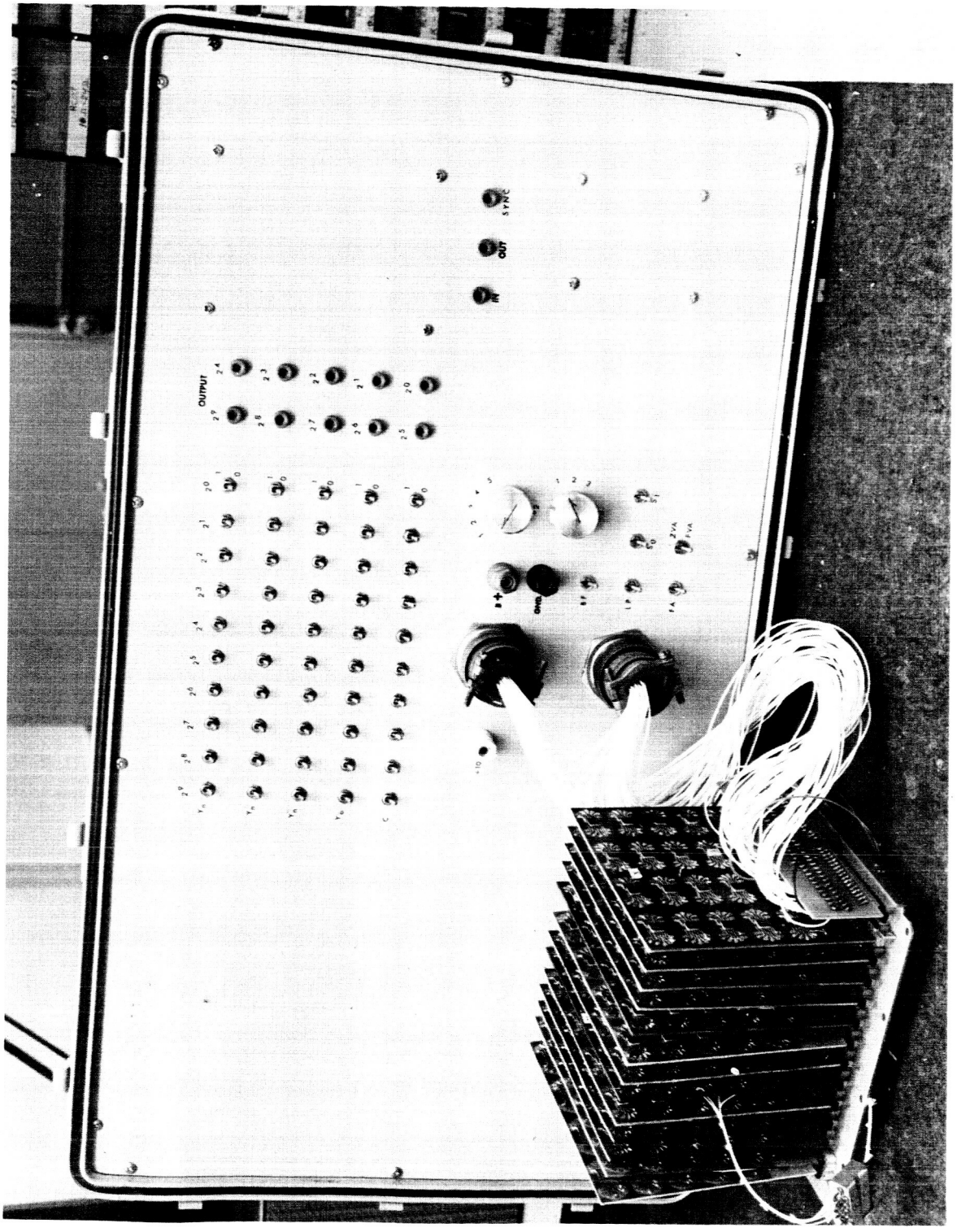


Figure 3-22
ZVA/FVA Breadboard and Test Panel

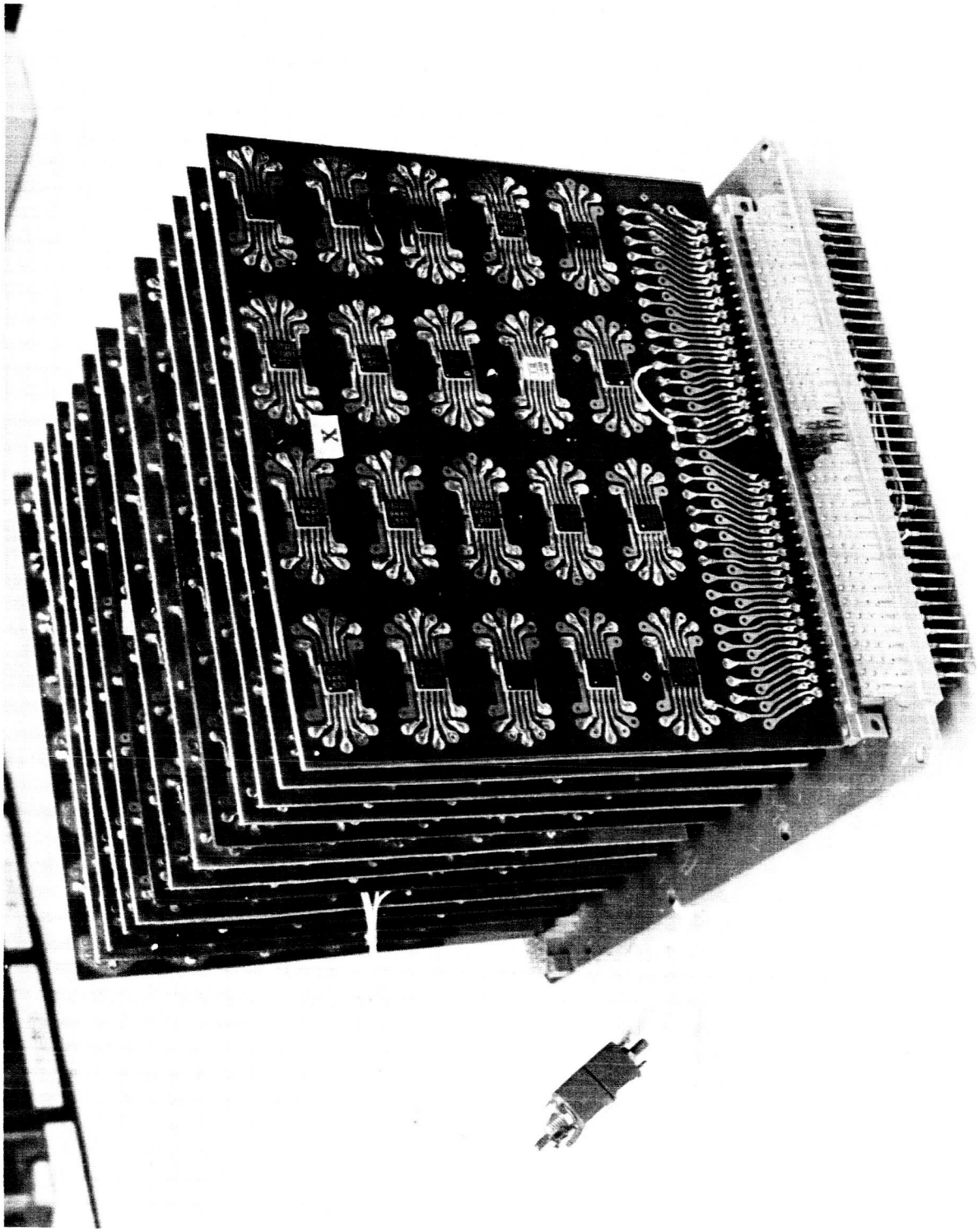


Figure 3-23
ZVA/FVA Breadboard
Board Z Removed

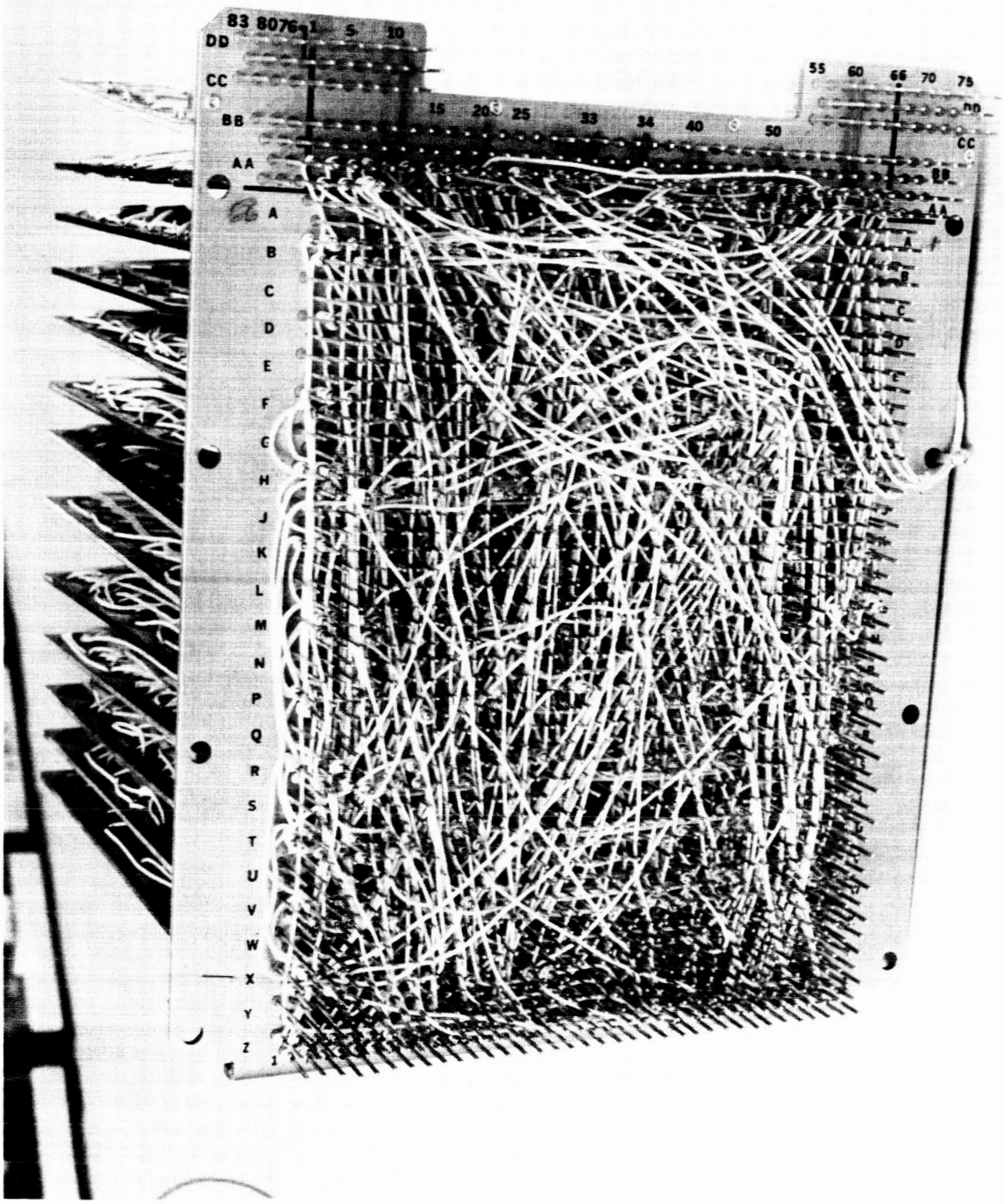


Figure 3-24
ZVA/FVA Breadboard
Interconnection Wiring

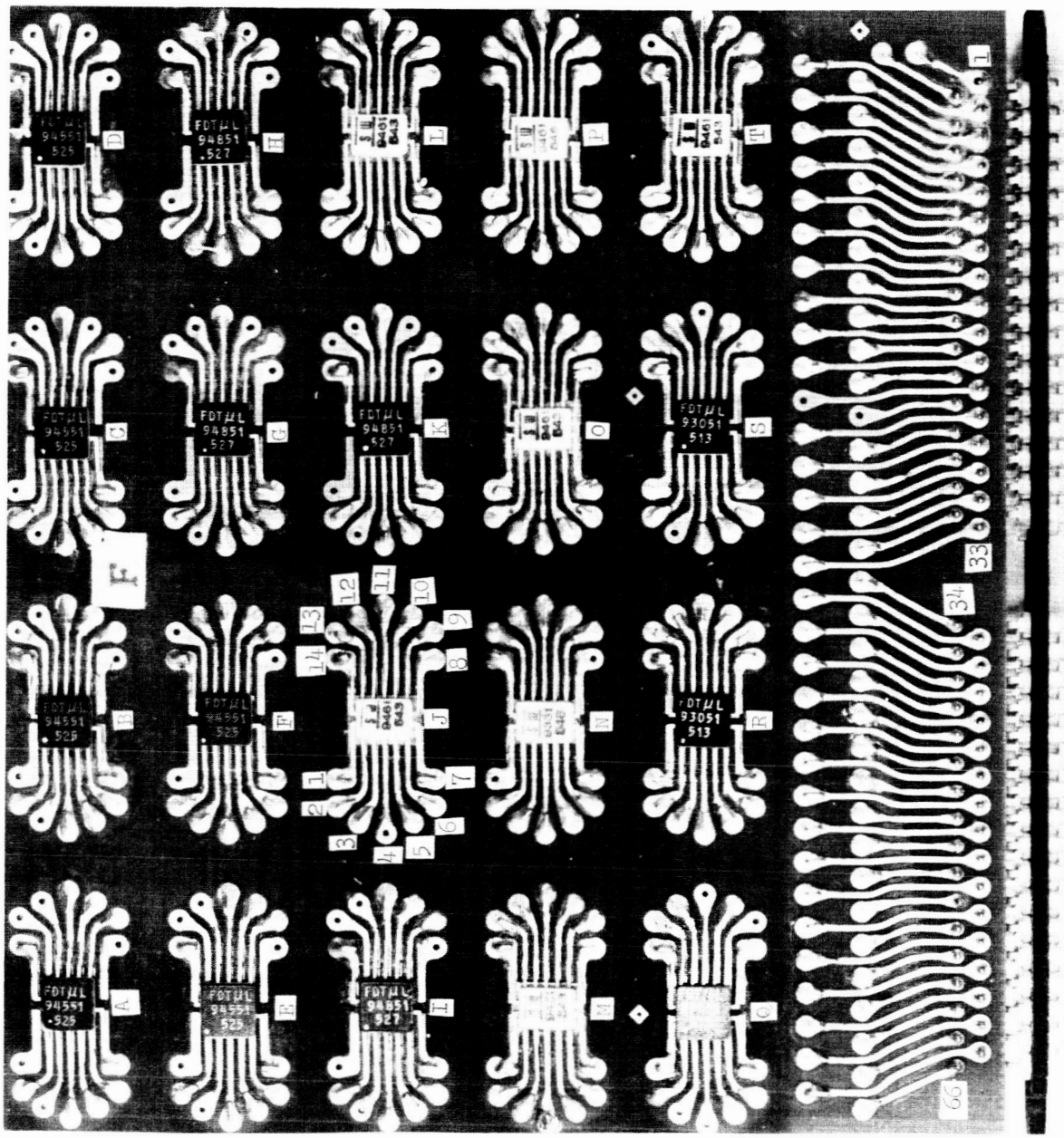


Figure 3-25
Board F - Module Side

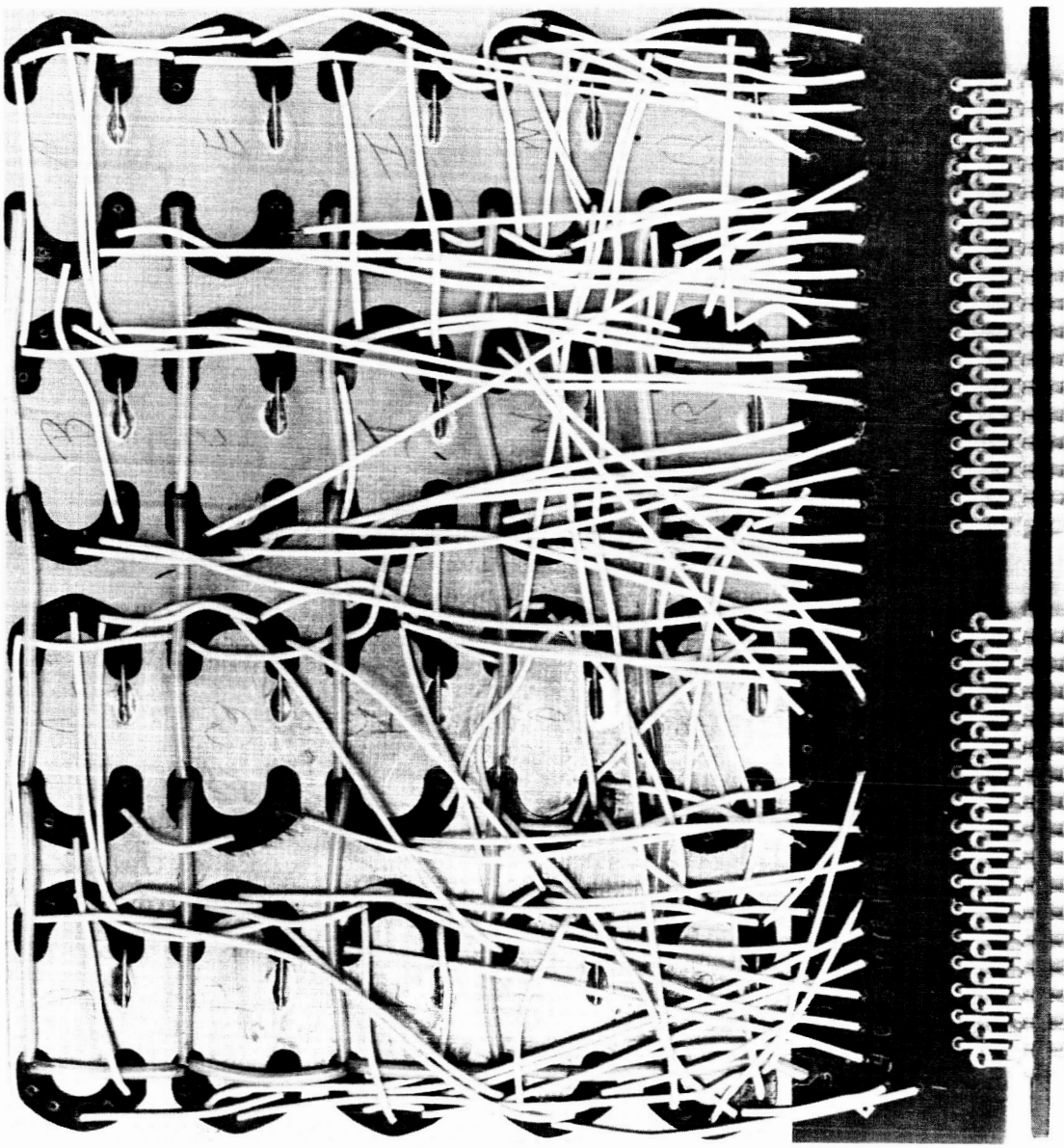


Figure 3-26
Board F - Wiring Side

Section 4
SIGNAL SIMULATOR AND OUTPUT MONITORING EQUIPMENT

4.1 General Discussion

The signal simulator provides new data points in parallel to the breadboard. These data points may be in the form of a ramp, an exponential stairstep, or a constant value. Also, the signal simulator provides the tolerance value, selects the algorithm, provides high or low priority information, simulates two levels of buffer fullness, and can force an output each sample time. If the data point provided is a constant, Y_o, Y'_n and Y''_n are also provided as inputs to the breadboard.

The output monitoring circuit is a ten bit D-A converter which is used to convert the ten bit parallel breadboard output to a pulsed analog voltage. Also, test jacks are provided for monitoring each of the ten bit breadboard outputs. A ten bit D-A converter also converts the data input from the signal simulator to the breadboard to an analog voltage. The two D-A converter outputs can be simultaneously displayed on a dual trace scope for comparison. Figure 4-1 is a block diagram showing the interconnection between the signal simulator, the breadboard, the D-A converters and the scope.

4.2 Detailed Discussion

The schematics of the signal simulator are shown in Figs. 4-2 through 4-5. Board 1 consists of a ten bit ripple carry counter (modules A through H plus L and K) and a ten bit shift register (modules I and J and modules M through T). When switch S_2 is in position 2, the A sides of the counter are routed through S_2 to the inverters. The ten bit digital output of the inverters is converted to an analog voltage by the input D-A converter. When switch S_2 is in position 2, the resultant output of the D-A converter is a positive-going ramp. When S_2 is in position 3, the $\overline{A}$ sides of the counter are routed through S_2 to the inverters. In this case the input D-A converter output is a

negative-going ramp. When S_2 is in position 4, the B sides of the shift register are routed through S_2 to the inverters. In this case the input D-A converter output is a positive-going stairstep exponential. When S_2 is in position 5, the $\overline{B}$ sides of the shift register are routed through S_2 to the inverters. In this case the input D A converter output is a negative-going stairstep exponential.

Modules P, L, Q, T, R and S on Board 2 and switch S_3 constitute the feedback logic and the input logic for the shift register on Board 1. This logic causes the shift register to sequence through the 11 states listed below:

STATE #	B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
1	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0
4	1	1	1	0	0	0	0	0	0	0
5	1	1	1	1	0	0	0	0	0	0
6	1	1	1	1	1	0	0	0	0	0
7	1	1	1	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1	0	0	0
9	1	1	1	1	1	1	1	1	0	0
10	1	1	1	1	1	1	1	1	1	0
11	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0

Modules I and J on Board 2 divide the commutation rate by 4. Modules K, O, M and N and switches S_0 and S_1 constitute the input logic for the counter on Board 1.

Gates H3 and H6 provide the transfer pulse to 18 other gates on Board 2. Gates A8, A11, B3, B6, B8, B11, C3, C6, C8 and C11 insert the new data word (inverter outputs) into register Y_n in the breadboard during logic step 1. Note that the input D-A converter output is the analog equivalent of the new data word. Gates A3 and A6 provide a transfer pulse to the Y_n'' switch row. These pulses transfer the Y_n'' switch positions to the Y_n'' register in the breadboard during logic step 1. When Switch S_2 is in position 1, these pulses simulate the reference memory

output. Gates H8 and H11 perform a similar function for register Y'_n. Likewise, gates D3 and D6 perform a similar function for register Y_o, as does gate D11 for the K flip flops.

The output D-A converter, Fig. 4-6, converts the output of the breadboard to a pulsed analog output.

4.3 Operation Instructions

Equipment required:

One Signal Simulator
One Breadboard
One 2.8 MC pulse generator
One Dual Trace Scope
One 0 to +6 volt DC, 2 amp. variable power supply
One +10 volt DC 500 Ma power supply

Operation Steps:

1) Connect J1 and J2

2) Connect all grounds to the black ground terminal. This includes the black wire from board Z of the breadboard, the 0 to +6 volt power supply return, the +10 volt power supply return and the pulse generator return.

3) Connect the red wire from board Z to the red terminal (B+). Also connect the plus output of a 0-6 volt DC power supply. The supply should be initially at 0 volts and slowly increased to +5 volts while monitoring the current of the supply. The signal simulator plus the breadboard should draw approximately 1.5 amps.

CAUTION

Do not apply more than +6 volts or any negative voltage to the B+ terminals. If this is done all modules may be eliminated.

4) Apply +10 volts to the white terminal.

5) Connect the blue wire from board Z to a 2.8 MC pulse generator. The pulse should vary from 0 volts to +4 volts. (CAUTION -- DO NOT EXCEED +6.0 VOLTS.) The pulse should be +4 volts about 1/3 of the time and 0 volts about 2/3 of the time.

6) Select the algorithm by placing the algorithm selection switch in either the FVA or ZVA position.

7) If it is desired to force an output every sample period, place the FA switch in the up position. Otherwise, place it in the down position.

8) Select the channel priority by placing the LP switch in the up position for low priority or in the down position for high priority.

9) Select one of two buffer fullness levels by the placing of the BF switch in the up position for the higher level and in the down position for the lower level. If both the LP and BF switches are up, the channel will be rejected. If either or both are down, operation will be normal.

10) Select the tolerance range by placing the $C2^1$, $C2^2$ and $C2^3$ switches in one of the positions shown below:

$C2^1$	$C2^2$	$C2^3$	Tolerance
0	0	0	±0 bits or 0.0%
1	0	0	±2 bits or 0.2%
0	1	0	±4 bits or 0.4%
1	1	0	±6 bits or 0.6%
0	0	1	±8 bits or 0.8%
1	0	1	±10 bits or 1.0%
0	1	1	±12 bits or 1.2%
1	1	1	±14 bits or 1.4%

Note that the up position is a "1" and the down position is a "0". Switches $C2^0$, $C2^4$, $C2^5$, $C2^6$, $C2^7$, $C2^8$ and $C2^9$ are not connected and therefore their position is immaterial.

Static Input

1) Place S_2 in position 1. The positions of S_o, S_1 and S_3 are immaterial.

2) Program Y_n (the data input), Y_n', Y_n'' and Y_o as desired.

Dynamic Input

1) Place all Y_n, Y_n', Y_n'' and Y_o'' switches in the down position. If this is not done, the results will be invalid.

2) Select the input waveform by placing switches S_o, S_1, S_2 and S_3 in one of the combinations listed below:

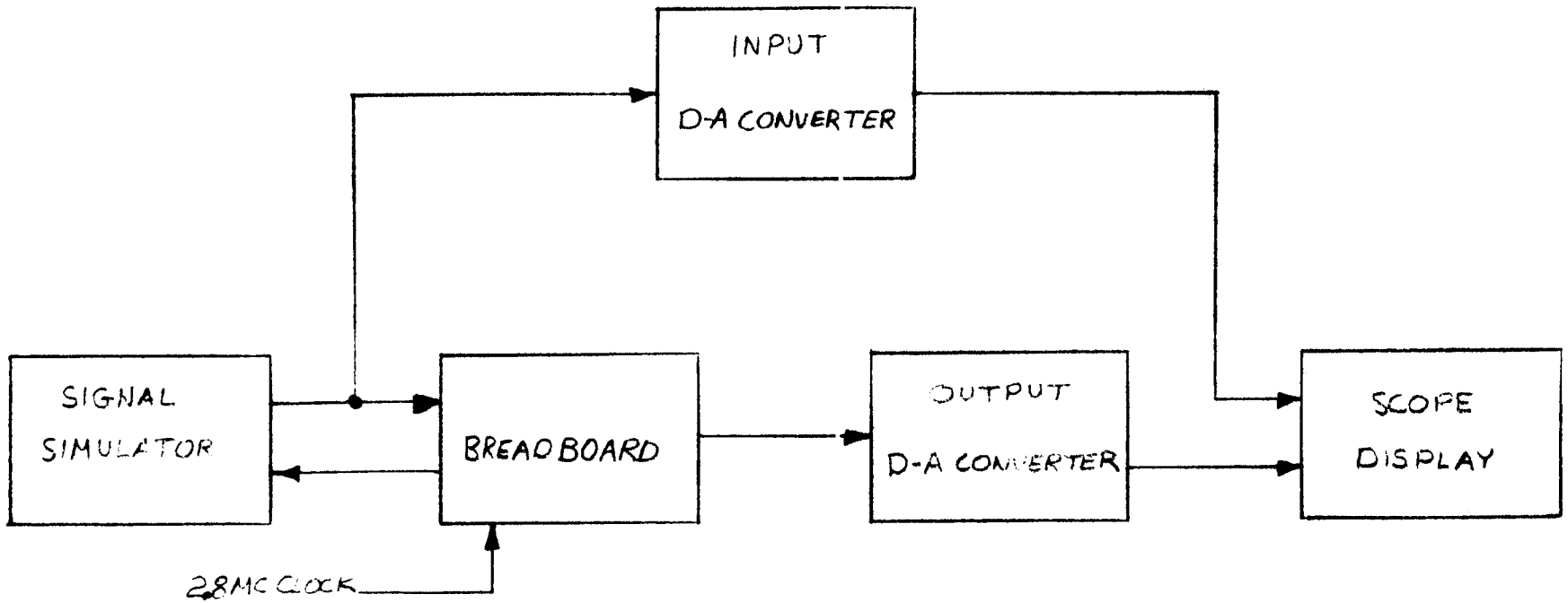
	S_o	S_1	S_3	S_2
Positive Ramp	down	down	x	2
Positive Ramp 2 slopes	up	down	x	2
Positive Ramp 1 slope	down	up	x	2
Negative Ramp 1 slope	down	down	x	3
Negative Ramp 2 slopes	down	up	x	3
Positive exponential 1 sample per step	x	x	1	4
Positive stairstep exponential (2 samples per step)	x	x	2	4
Positive stairstep exponential (4 samples per step)	x	x	3	4
Negative exponential 1 sample per step	x	x	1	5
Negative stairstep exponential 2 samples per step	x	x	2	5
Negative stairstep exponential 4 samples per step	x	x	3	5

x = immaterial

Output Monitoring

1) Static input -- The 10 parallel outputs, the pulsed output from the output D-A converter and the output of the input D-A converter may be observed on the scope. Sync should be obtained from a signal in the breadboard, such as $\overline{1}$ on pin V66 in the breadboard.

2) Dynamic inputs -- Sync the scope to the sync jack on the test panel. Using a dual trace scope, such as the Tektronix 545 or 585, both the input and the output can be observed simultaneously. The output is inverted with respect to the input.



BLOCK DIAGRAM
BREADBOARD AND SIGNAL SIMULATOR
FIG. 4-1

LMSC 669224

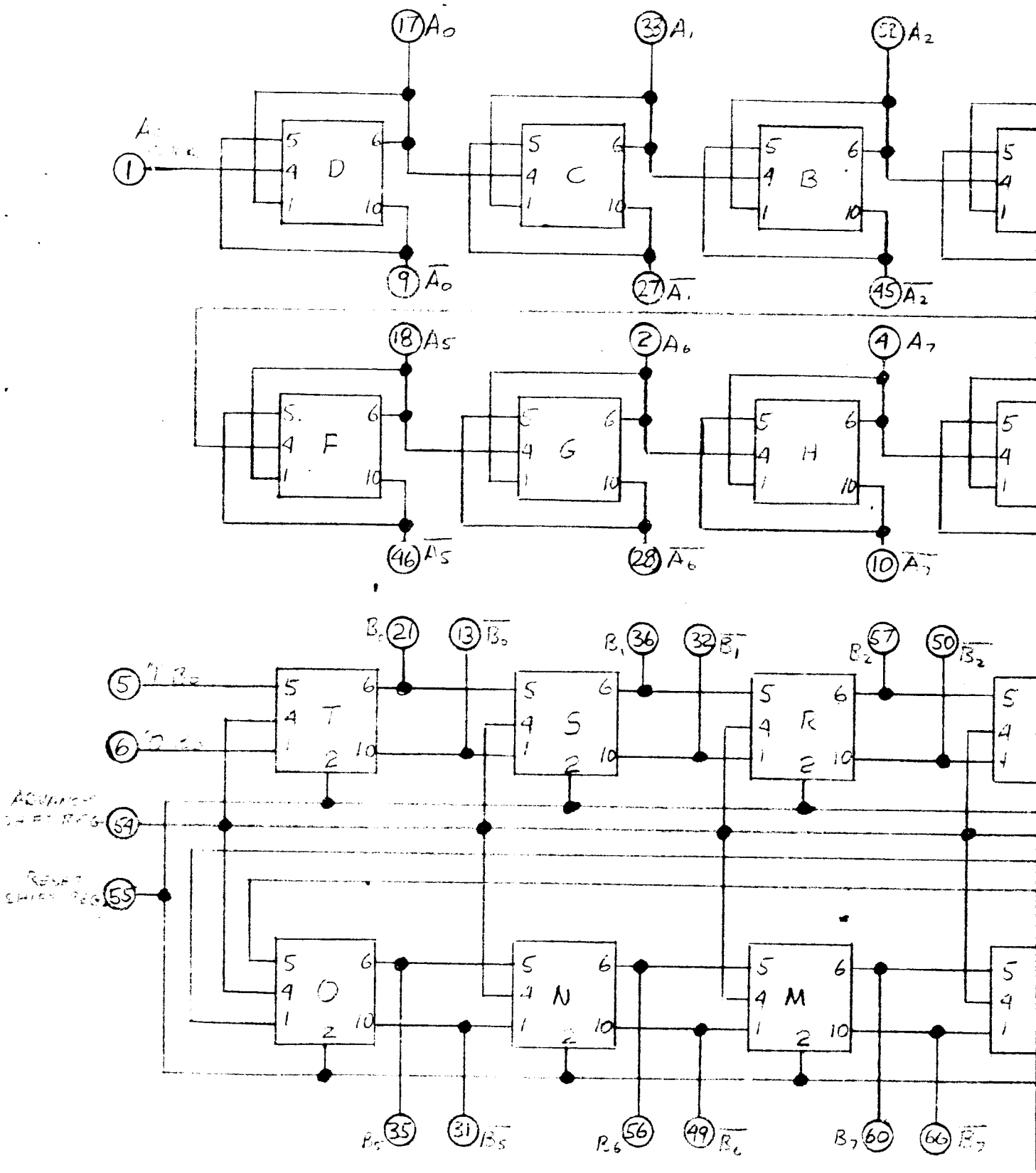
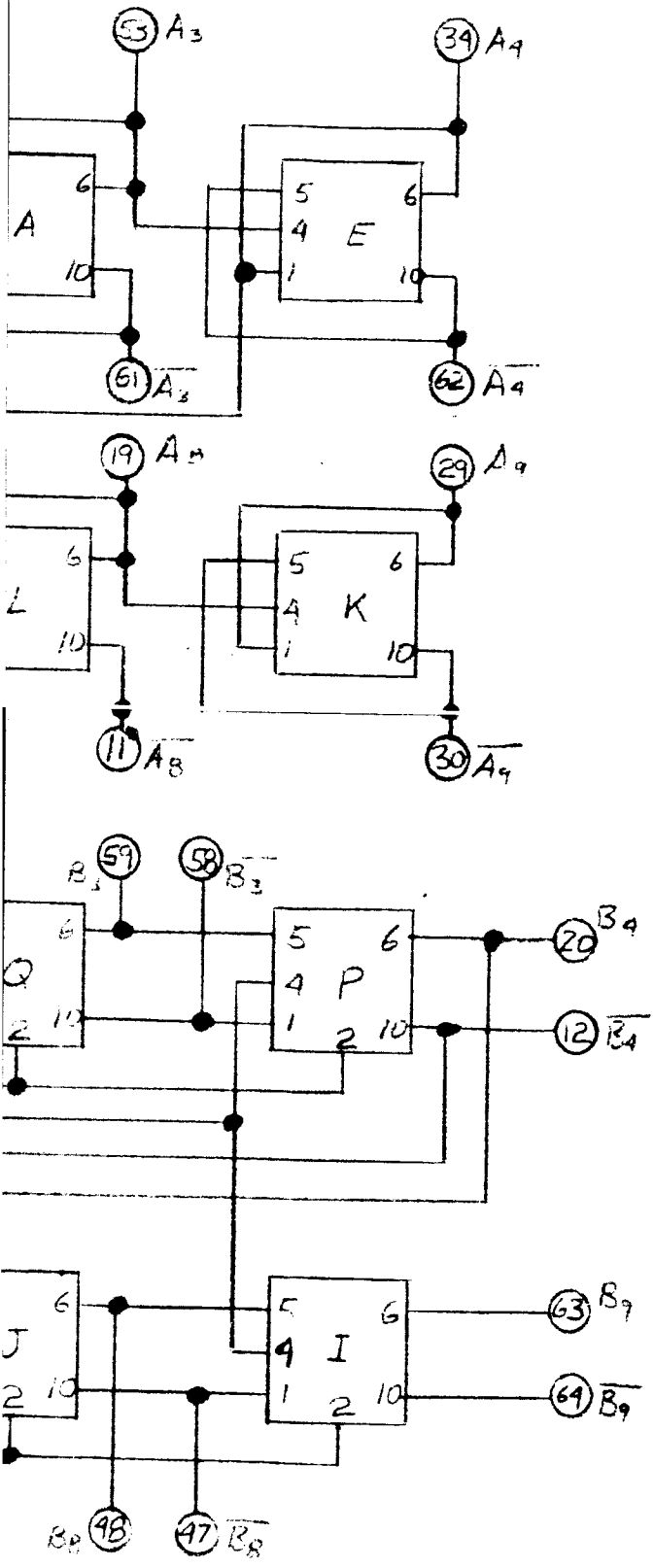

17 A_0
33 A_1
32 A_2
1
D
C
B
9 $\overline{A_0}$
27 $\overline{A_1}$
45 $\overline{A_2}$
18 A_5
2 A_6
4 A_7
F
G
H
46 $\overline{A_5}$
28 $\overline{A_6}$
10 $\overline{A_7}$
B_0 21
13 $\overline{B_0}$
B_1 36
32 $\overline{B_1}$
B_2 57
50 $\overline{B_2}$
5
6
T
S
R
ADVANCE SHIFT REG
54
RESET SHIFT REG
55
O
N
M
B_5 35
31 $\overline{B_5}$
B_6 56
49 $\overline{B_6}$
B_7 60
66 $\overline{B_7}$

FIG. 4-2

TEST PANEL BOARD #1



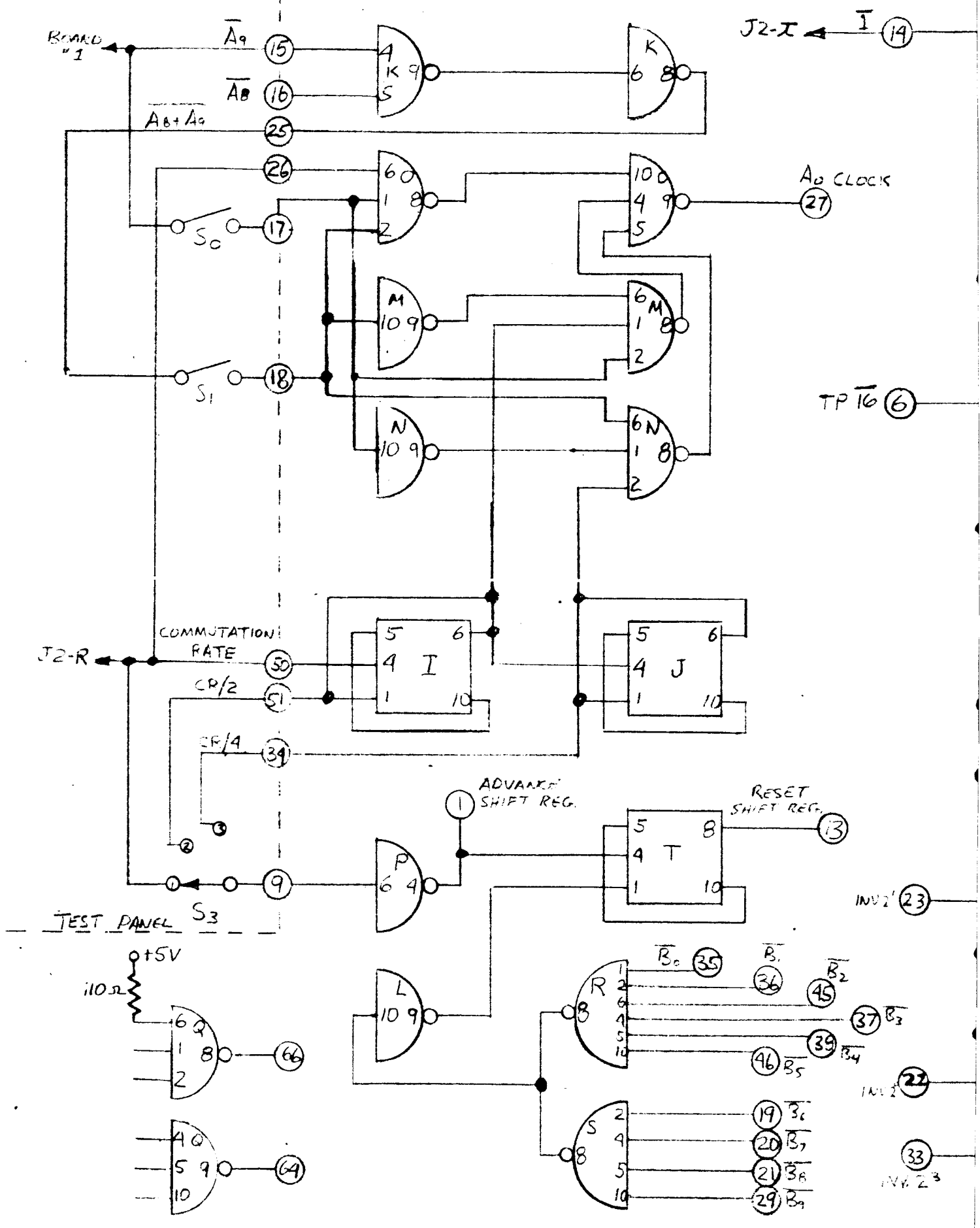
3 B+

65 GND

ALL MODULES SERIES 51 T.I. FLIP FLOPS

DRAWN BY W.A. [illegible]

DATE 31 JAN 66


BOARD "1
$\overline{A_9}$
15
$\overline{A_8}$
16
$A_8 + A_9$
25
26
17
S_0
18
S_1
J2-I
I
19
A_0 CLOCK
27
TP $\overline{16}$
6
COMMUTATION RATE
J2-R
50
CP/2
51
CP/4
34
ADVANCE SHIFT REG.
1
RESET SHIFT REG.
13
INV 2'
23
9
S_3
TEST PANEL
+5V
110Ω
66
64
$\overline{B_0}$
35
$\overline{B_1}$
36
$\overline{B_2}$
45
$\overline{B_3}$
37
$\overline{B_4}$
39
$\overline{B_5}$
46
INV 2
22
$\overline{B_6}$
19
$\overline{B_7}$
20
$\overline{B_8}$
21
$\overline{B_9}$
29
INV 2³
33

LMSC
669224

FIG. 4-3
TEST PANEL
BOARD # 2

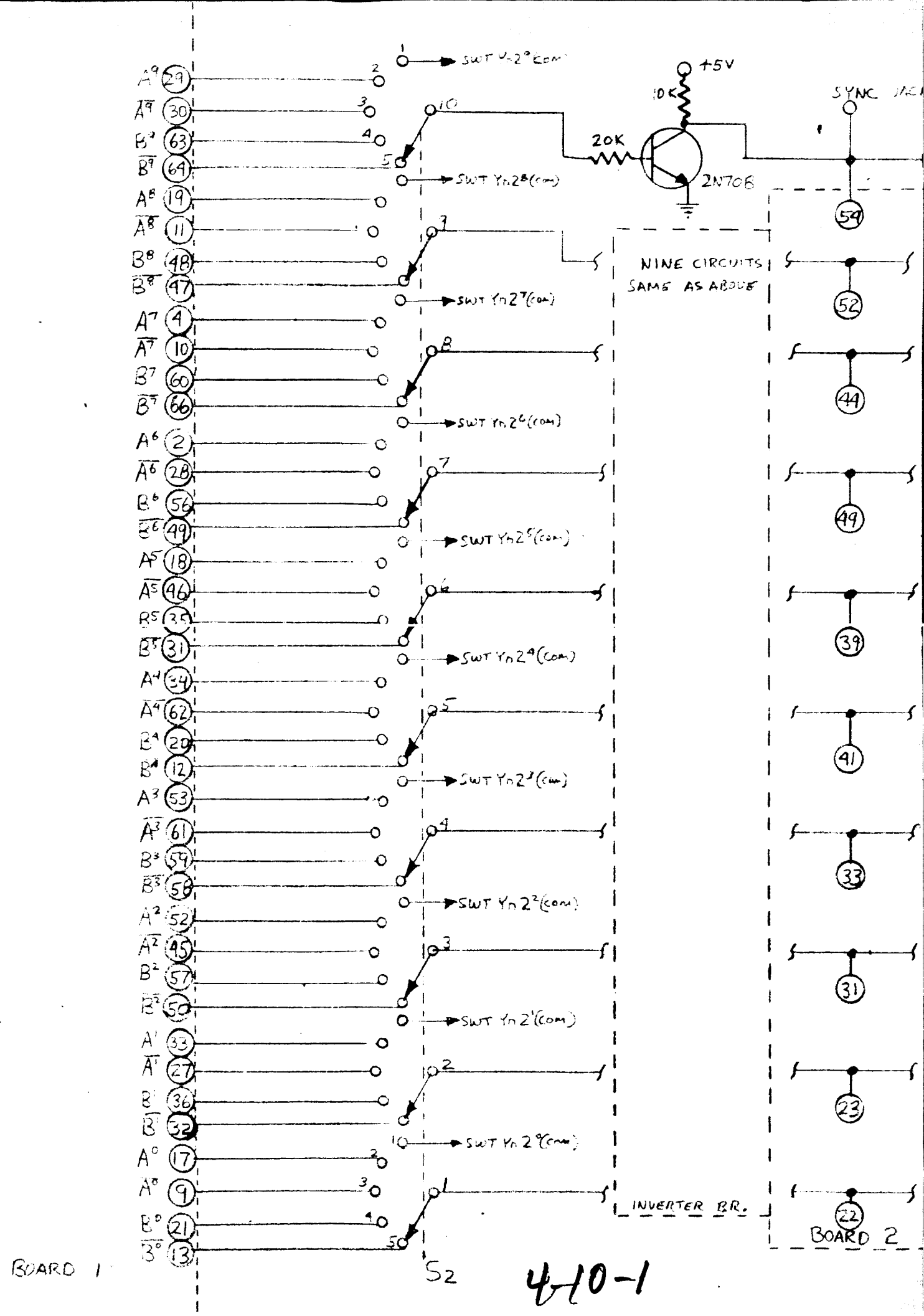
A- 946
B- 946
C- 946
D- 946
E-
F-
G-
H- 946
I- SN323
J- SN323
K- SN325
L- SN325
M- SN514
N- SN514
O- SN514
P- SN517A
Q- SN514
R- SN329
S- SN329
T- SN323

DRAWN BY W.A. LEE
DATE 31 JAN 1966

Yn' SWTS. (4)
Yn' SWTS (5)
S2-12-1 (8) TP 16
C SWTS. (7)
TO SWTS (12)
YO SWTS. (10)
INPUT 2' (24)
INPUT 2° (11)
INPUT 2³ (30)

INV 2² (31) INPUT 2² (32)
INV 2⁵ (39) INPUT 2⁵ (43)
INV 2⁹ (41) INPUT 2⁹ (40)
INV 2⁷ (44) INPUT 2⁷ (48)
INV 2⁶ (49) INPUT 2⁶ (47)
INV. 2⁹ (54) INPUT 2⁹ (55)
INV 2⁸ (52) INPUT 2⁸ (53)
Yn'' SWTS (63)
Yn' SWTS (62)

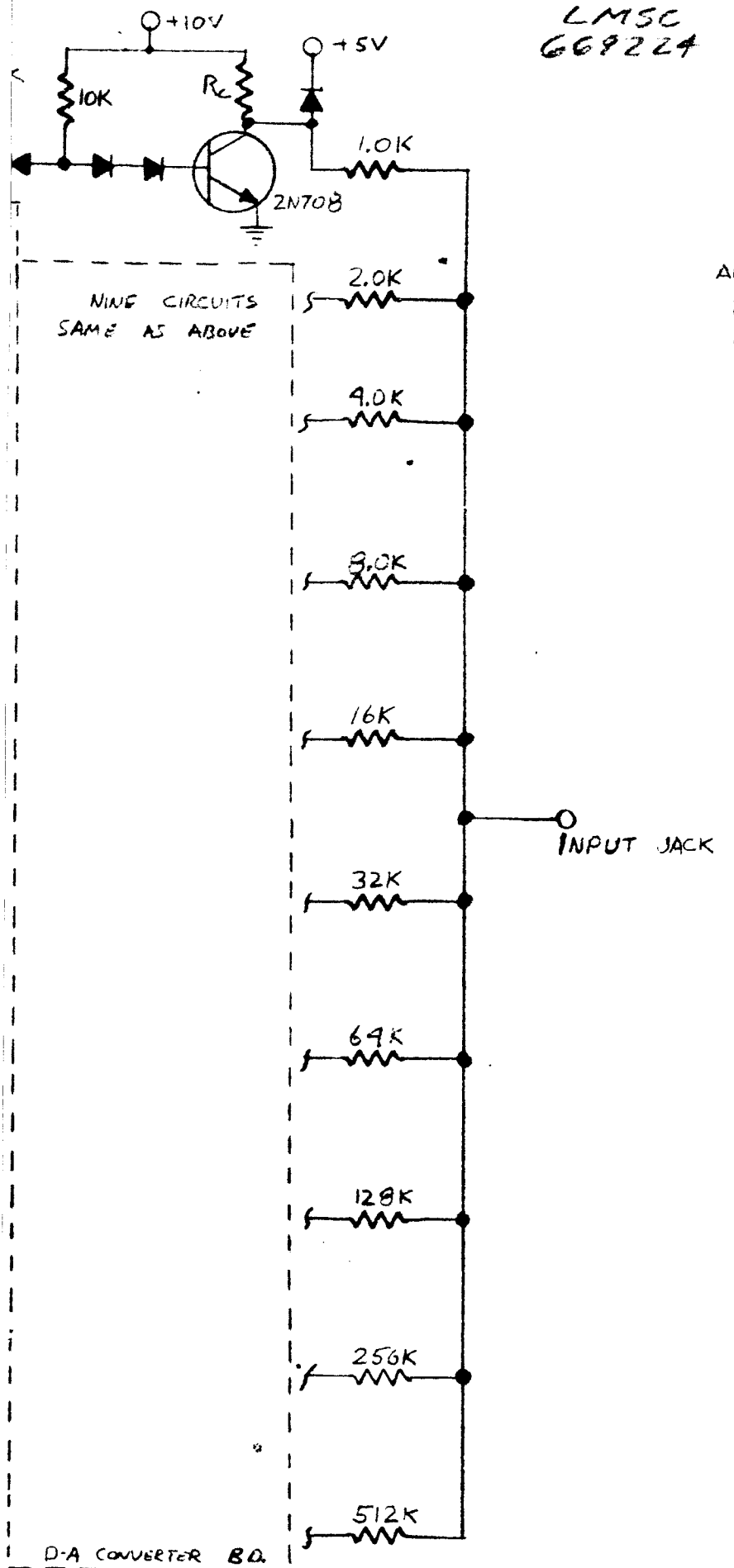
(3) B+
(65) GND.


+5V
10K
20K
2N706
SYNC
SWT Yn2⁹(com)
SWT Yn2⁸(com)
SWT Yn2⁷(com)
SWT Yn2⁶(com)
SWT Yn2⁵(com)
SWT Yn2⁴(com)
SWT Yn2³(com)
SWT Yn2²(com)
SWT Yn2¹(com)
SWT Yn2⁰(com)
NINE CIRCUITS SAME AS ABOVE
INVERTER BR.
BOARD 1
BOARD 2
S2
4-10-1

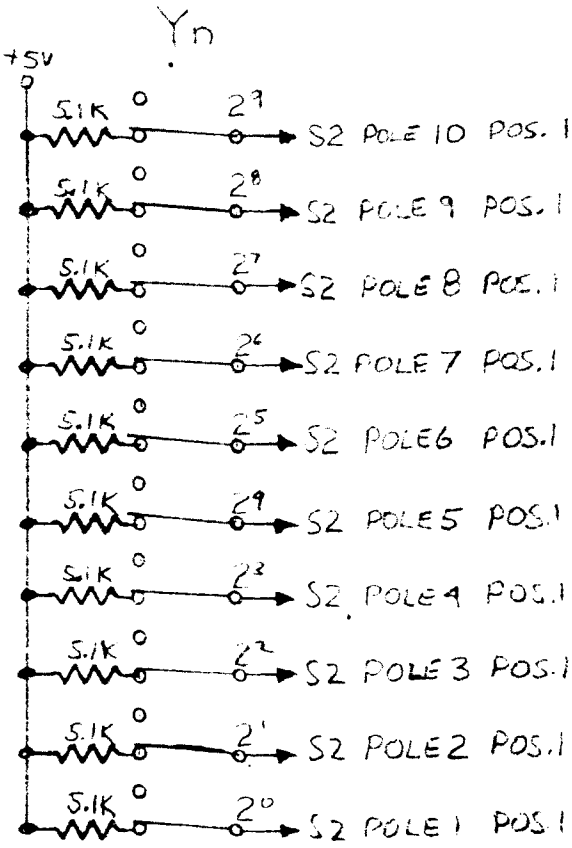
LMSC
669224

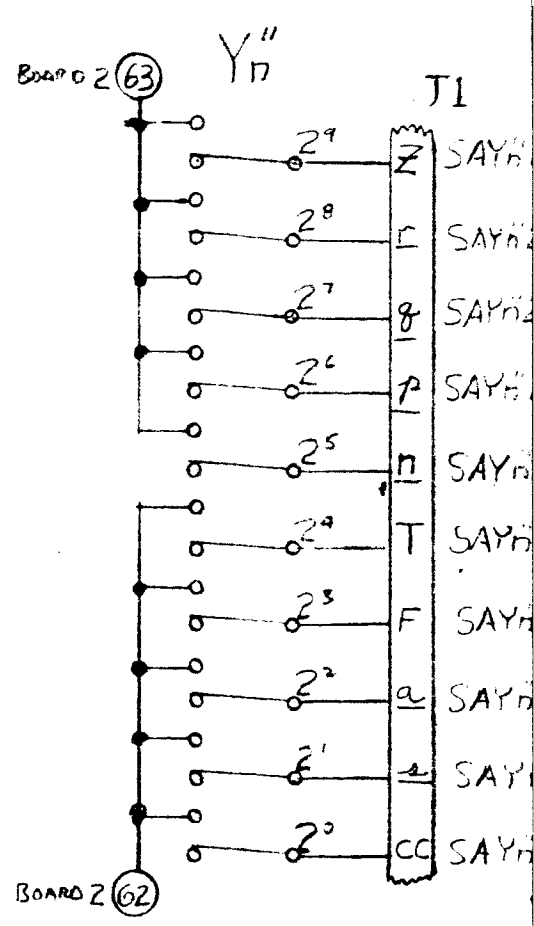
FIG. 4-4
TEST PANEL INPUT D-A CONVERTER AND RELATED CIRCUITS.

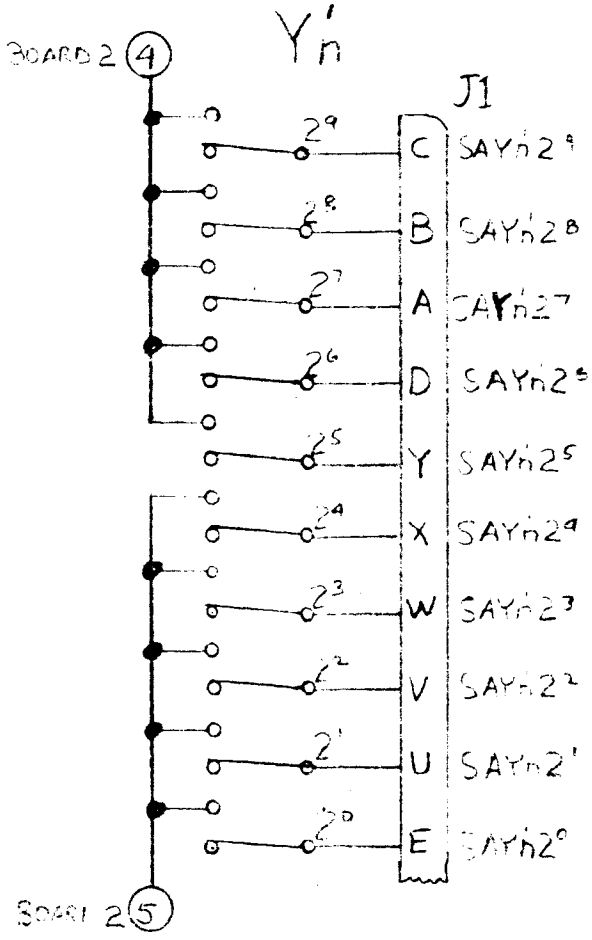
ALL DIODES 1N916A
DRAWN BY W A LEISER
DATE 1 FEB 1966

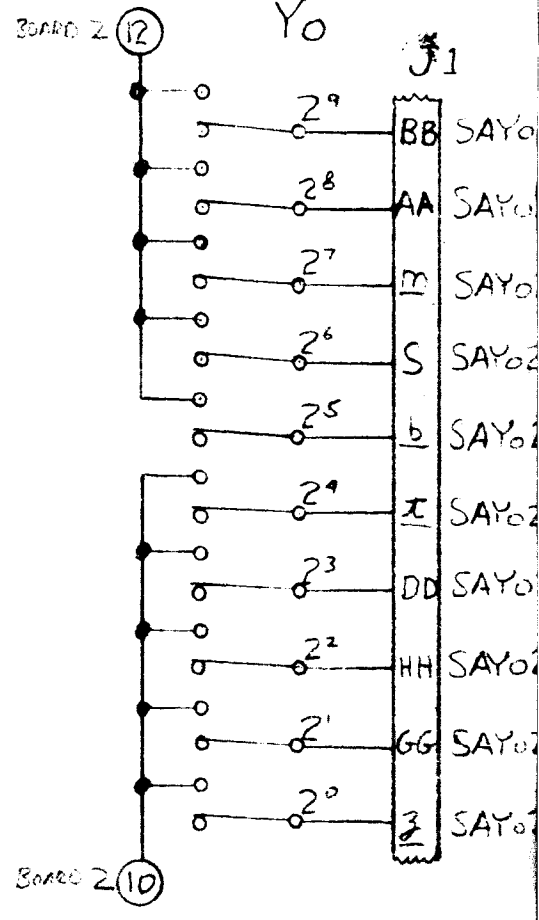


CKT	Rc
2^9	750 Ω
2^8	910 Ω
2^7	1.0K
2^6	1.1K
2^5	1.2K
2^4	1.2K
2^3	1.2K
2^2	1.2K
2^1	1.2K
2^0	1.2K


Yn
+5V
5.1K
2^9 S2 POLE 10 POS. 1
2^8 S2 POLE 9 POS. 1
2^7 S2 POLE 8 POS. 1
2^6 S2 POLE 7 POS. 1
2^5 S2 POLE 6 POS. 1
2^4 S2 POLE 5 POS. 1
2^3 S2 POLE 4 POS. 1
2^2 S2 POLE 3 POS. 1
2^1 S2 POLE 2 POS. 1
2^0 S2 POLE 1 POS. 1


Yn"
BOARD 2 (63)
J1
Z
r
g
p
n
T
F
a
s
CC
BOARD 2 (62)

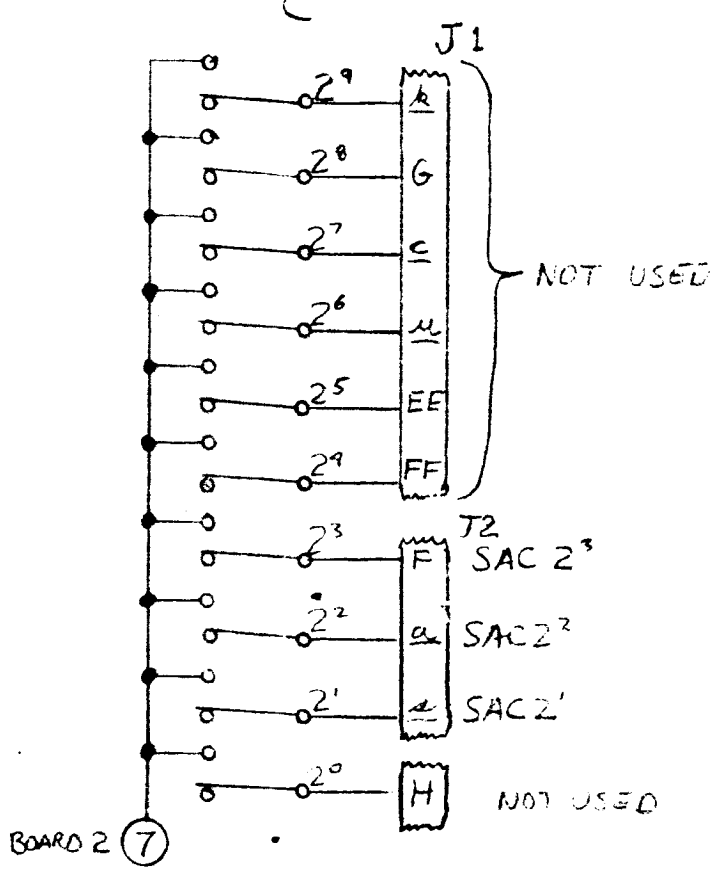

Yn'
BOARD 2 (4)
J1
C
B
A
D
Y
X
W
V
U
E
BOARD 2 (5)

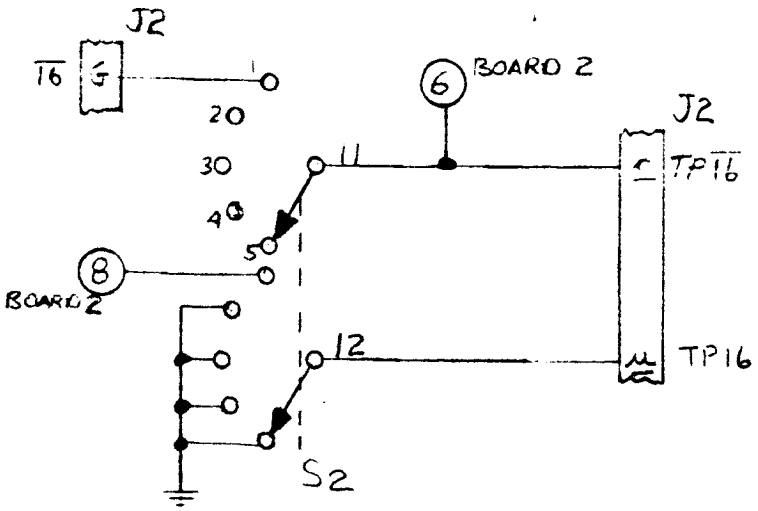

Yo
BOARD 2 (12)
J1
BB
AA
m
S
b
x
DD
HH
GG
3
BOARD 2 (10)

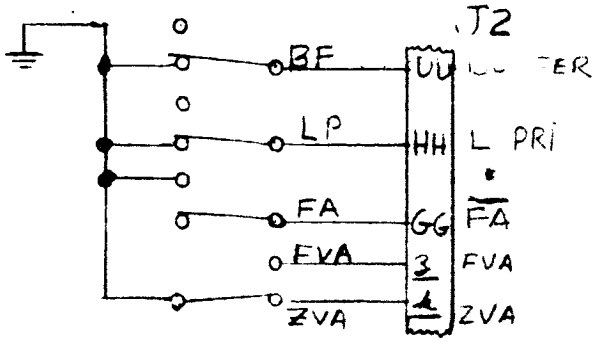
LMSC
669224

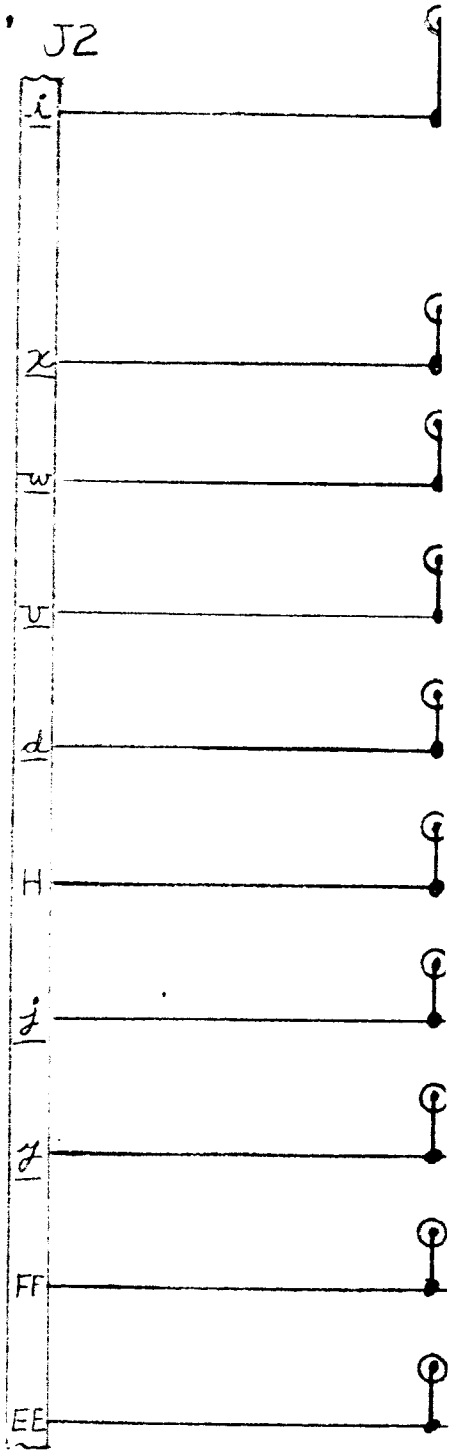
FIG. 4-5
TEST PANEL SWITCHES.

DRAWN BY W.A. LOISON
DATE 1 FEB 1966





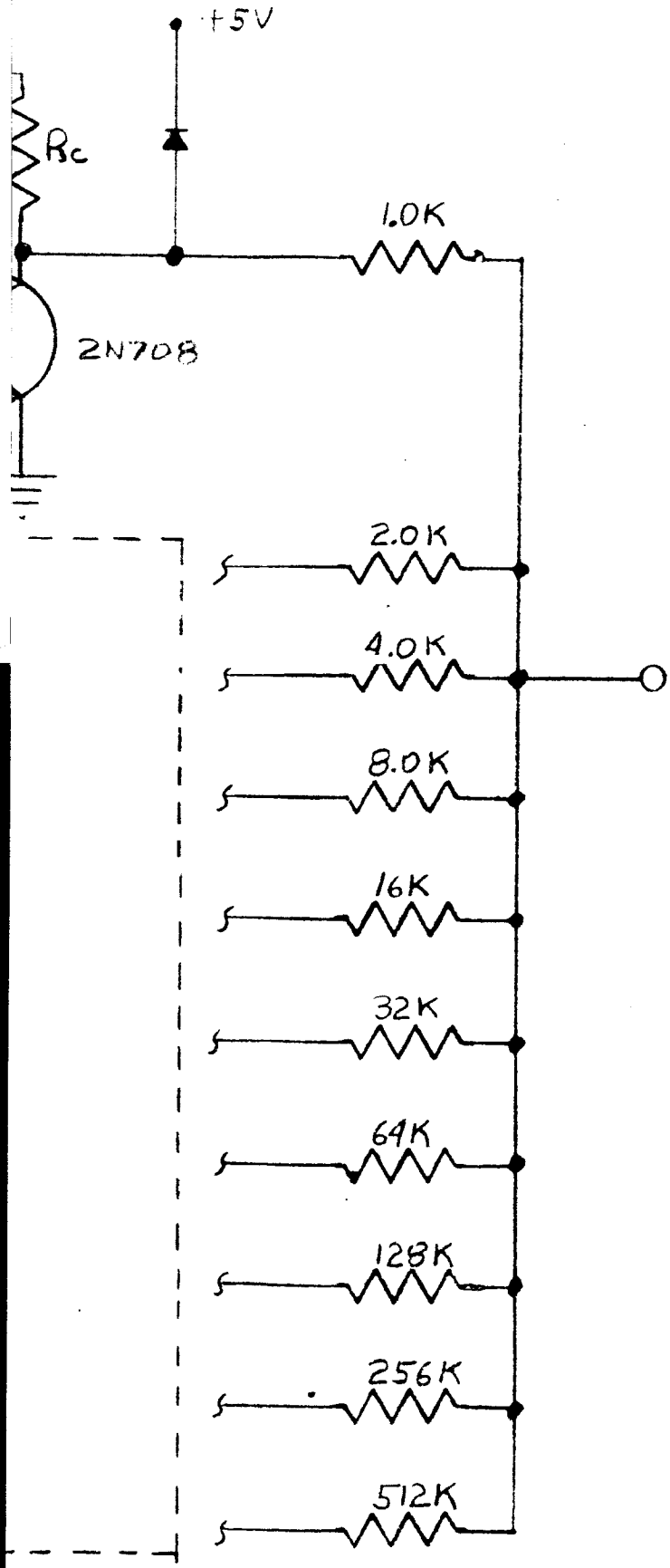



J2
i
x
w
v
d
H
j
z
FF
EE

LMSC
666224

FIG. 4-6

OUTPUT
D-A CONVERTER
TEST PANEL



CKT.	Rc
2^9	750Ω
2^8	910Ω
2^7	1.0K
2^6	1.1K
2^5	1.2K
2^4	1.2K
2^3	1.2K
2^2	1.2K
2^1	1.2K
2^0	1.2K

ALL DIODES IN716A

Section 5
REFERENCE MEMORY INVESTIGATION

5.1 General

To implement the hybrid ZVA/FVA compression algorithms, a reference memory with an increased storage capacity is required (as compared to the reference memory previously mechanized for the ZFN compression algorithm). In the evaluation of methods to achieve the increased storage capacity, the retention of a combination of DRO and NDRO storage was considered highly desirable. There were two methods which were considered feasible. These methods are discussed in paragraphs 5.2 and 5.3 below.

Possible improvements of memory mechanization, by use of available integrated circuits, was also investigated. This investigation was concentrated on the suitability of available core memory sense amplifiers for use in memories of flyable data compressors. The sense amplifier investigation is discussed in paragraph 5.4.

5.2 Extension of Bit Capacity of the Linear-Select Memory

The memory developed for the ZFN flyable data compressor has a combination DRO and NDRO storage capability. The addressing of the memory cores is by a linear-select method (all cores, related to a stored word, are selected by a current pulse in a single address wire). The selection of a specific memory word is made through sequential-operating core-diode current-steering address switches and associated regulated-current drivers. This address circuitry was designed to have adequate performance margins when driving 11 DRO and 5 NDRO memory cores. To mechanize the ZVA/FVA algorithms, approximately 37 DRO and 6 NDRO bits are required. To drive this larger number of cores, revision of the core-diode address switch is required. Also, the regulated-current drivers may require circuit design changes.

From the factors discussed, up to this point, it would appear that the ZVA/FVA memory requirements can be met by minor design revisions to the address driving circuitry. However, complications arise from a problem that has developed in the availability of two-hole cores for NDRO storage. At this time, there is no proven source of two-hole cores which meet the memory core requirements previously established for the flyable ZFN Data Compressor. (A "proven source", as used here, is a memory core producer capable of showing evidence of ever having previously produced cores of the required characteristics and able to assure future deliveries of these cores.) To use a two-hole core, of different characteristics than previously used, requires that additional evaluation of available two-hole cores be made and that a different toroidal (DRO) core be evaluated. Also, the regulated-current drivers and core-diode address switches may require further circuit revisions.

The extension of the linear-select, DRO-NDRO memory, to the required storage capacity is feasible. However, this may not be the best approach. A preliminary evaluation of an alternate approach was made which seems to offer a better solution to the requirements.

5.3 Coincident-Current DRO-NDRO Memory

The alternate approach, to meet the hybrid ZVA/FVA memory requirements, involves a scheme to provide a combination of DRO and NDRO storage capability within a coincident-current addressed memory. This organization would be more flexible for meeting possible future revisions of storage requirements and should be more economical to make. A preliminary evaluation of this scheme was made by driving a small sample of two-hole cores with a combination of current pulses which simulate the proposed driving scheme. The results from this preliminary evaluation indicated a good probability of success in obtaining the desired performance of a memory system. There should be further evaluation, by test of a wired array of two-hole cores (such as a 32x32 core array), prior to building a complete memory system.

This coincident-current addressed, DRO-NDRO storage, memory scheme is shown in Fig. 5-1. Its use would have the following advantages and restrictions:

a) A combination of DRO and NDRO storage, with only one additional drive circuit required above that required for the same total number of bits stored in a DRO fashion, could be achieved. (This added "clear" drive circuit and one "inhibit" drive circuit per NDRO bit are not used after NDRO information is loaded into memory. Therefore, it may be plausible to locate these circuits in ground-based test equipment. The actual number of drive circuits and power required for a combination DRO/NDRO vehicle-borne memory would then be less than for an all DRO memory.)

b) Additional cost of NDRO capability, over the cost of DRO storage, will essentially be limited to the higher cost of two-hole cores and the cost of stringing one additional wire through each two-hole core (conventional DRO cores have four wires per core -- NDRO requires five wires per core).

c) The combination of DRO/NDRO storage would be expandable. Additional bits of DRO and/or NDRO storage could be added or removed without functional redesign.

d) Before loading any new NDRO information into the memory, all old NDRO information must be "cleared" out. Therefore, any change in stored NDRO information requires complete reloading. (Complete reloading is required if (a) and (b) are to be achieved. By adding more drive circuits and two more wires through each NDRO core, the change of a single word of NDRO storage, without complete reloading, would be possible.)

e) When new NDRO information is loaded, all stored DRO information is destroyed.

5.4 Integrated Circuit Sense Amplifiers

The investigation into the possibility of using an integrated circuit sense amplifier, within flyable data compressor memories, was essentially limited to a survey of vendor specifications. It was determined that there are three (3) integrated circuit sense amplifiers available which appear to meet input-output performance requirements. These are: Fairchild Type μA711, Sylvania type SA-10, and Texas Instrument type SN5500. (Evaluation tests which included these types were conducted by others at IMSC for a similar requirement.)

For those integrated circuits which meet input-output performance requirements, there remains one major drawback. The power required is 125 to 200 milliwatts per circuit. With a total of approximately 60 circuits required, for a hybrid ZVA/FVA data compressor system, 7.5 to 12 watts of regulated power would be required for memory sense amplifiers. If a suitable arrangement can be worked out for switching power to the sense amplifiers, then total sense amplifier power consumption can be held to a more reasonable 0.5 watt or less. This appears to be feasible with the Fairchild type μA711 and Sylvania type SA-10. The Texas Instrument type SN5500 has a one-shot included which appears to make power input switching impractical. The design and test of power switching should be a goal of future memory work.

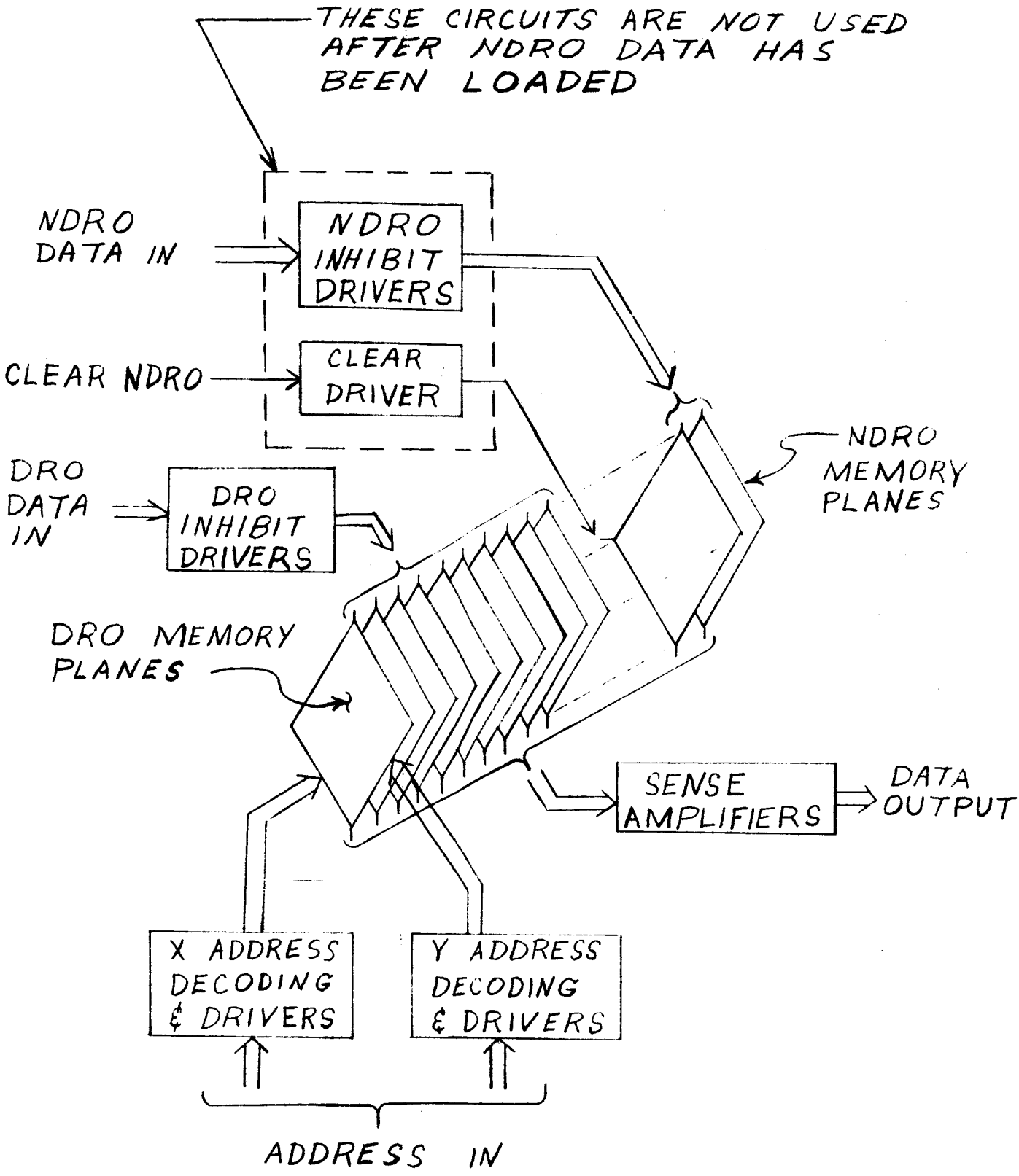


FIGURE 5-1 MEMORY WITH BOTH DRO & NDRO STORAGE CAPABILITY, USING COINCIDENT CURRENT ADDRESSING

Section 6
TEST RESULTS AND CONCLUSIONS

The basic objectives of the breadboard testing program were:

1) To verify that the detailed operation of the system was as planned and to determine approximate design margins.

2) To evaluate the approach to the logic design and to evaluate the performance and peculiarities of the logic families used.

3) To observe the compatibility between logic modules of different manufacturers.

Tests were conducted by applying controlled inputs and monitoring the breadboard outputs as well as various internal waveforms. Inputs to the breadboard consisted of both static and dynamic inputs. The static inputs simulated the reference memory outputs and the new data input and did not vary from sample to sample. These static inputs were most useful while trouble-shooting. The dynamic inputs simulated either ramp or exponential waveforms. When using the dynamic inputs, the information that would normally be stored in the reference memory was retained in the breadboard registers. (Registers Y_o, Y'_n and Y''_n, along with the n counter, were not reset during logic step 16.)

The system requirements dictate that the breadboard operate at a sample rate of 14.4 KC (clock rate of 2.8MC) and over the temperature range from -20°C to +85°C. The breadboard performed both the FVA and ZVA compression algorithms at a clock rate in excess of 2.8 MC and at temperatures below -20°C and above +85°C.

The B+ supply voltage, the clock and the temperature were varied in different combinations until the breadboard failed to perform. For example, the supply voltage was lowered from a maximum of +6.0 volts until failure. At no time

was the supply voltage greater than + 6.0 volts. The breadboard operation was arbitrarily limited to a maximum clock rate of 3.5 MC. The difference between the required range of operation and the point of failure indicates the degree of design margin. The results, summarized below, indicate a good design margin.

NOMINAL SUPPLY VOLTAGE	5.0 VOLTS
NOMINAL CLOCK RATE	2.8 MC
NOMINAL TEMPERATURE	-20°C to +85°C

Temperature	Mode of Operation	Range of Supply Voltage For Proper Operation	Range of Clock Rate For Proper Operation
+ 25°C	ZVA	6.0 to 3.9 volt	< 3.5 MC
+ 25°C	FVA	6.0 to 4.3 volt	< 3.5 MC
+ 90°C	ZVA	6.0 to 3.9 volt	< 3.5 MC
+ 90°C	FVA	6.0 to 4.4 volt	< 3.5 MC
+120°C	ZVA	6.0 to 3.8 volt	< 3.5 MC
+120°C	FVA	6.0 to 4.5 volt	< 3.0 MC
- 22°C	ZVA	6.0 to 4.5 volt	< 3.5 MC
- 22°C	FVA	6.0 to 4.5 volt	< 3.5 MC
- 55°C	ZVA	6.0 to 4.8 volt	< 3.0 MC
- 55°C	FVA	6.0 to 4.8 volt	< 3.0 MC

In general, the cause of failure was that propagation delays required between clock pulses were longer than the time between clock pulses. For example, with a temperature of + 120°C, the supply voltage of + 4.5 volts and the clock rate at 3.0 MC, the worst case propagation delay (propagation delay of 6 gates and one flip flop) was equal to 333 nanoseconds, the time between clock pulses. If the supply voltage was lowered, the clock rate increased, or the temperature increased, the worst case propagation delay increased and the breadboard failed to perform the FVA algorithm. Conversely, if the voltage supply was increased or the temperature decreased, the breadboard operated at a higher clock rate.

The results in the table do not in any way guarantee the system performance of this design nor represent the limit at which this design can be made to operate. The table shows only the range over which the breadboard was operated. The system design should be capable of operating over the required temperature range and at the maximum commutation rate.

The performance of the logic modules was considered satisfactory. Some of the results are listed below:

1) The Fairchild modules operated as specified by the data sheets. There were no peculiar modes of operation observed which were not covered on the data sheets. However, a significant number (approximately 10%) of Fairchild flip flops were defective when first tested. This indicates either that the flip flops were defective when received, or that they were damaged when soldered on the boards. This is not considered a major problem and, in either event, can be corrected by standard procedures. There were no failures of gates, expanders or clock drivers.

2) The Philco and Stewart Warner modules operated identically to the Fairchild modules. There was no interface problem observed between modules manufactured by the three companies and there were no failures of Philco or Stewart Warner modules. The quantities and types observed of each of these modules were limited, and further evaluation of Philco and Stewart Warner modules would be desirable. The number and types of modules used in the Breadboard are given below.

Manufacturer	No. of Type 930 Used	No. of Type 932 Used	No. of Type 933 Used	No. of Type 945 Used	No. of Type 946 Used	No. of Type 948 Used	Total Used
Fairchild	42	1	0	91	49	8	191
Stewart Warner	0	0	5	0	28	0	33
Philco	0	0	0	7	0	0	7
TOTAL	42	1	5	98	77	8	231

3) The logic ground was distributed on the printed circuit boards by means of a ground plane. Ground interconnections between boards were accomplished by means of number 26 wire connected between pins 65 of each board. Voltage variations between grounds on different boards were measured to be as high as 0.2 volts. However, there was no circuit malfunction traceable to noise problems. This indicates that the logic modules were relatively insensitive to ground noise. The ground plane proved quite satisfactory and should be used as part of the ground distribution system in future designs. The ground interconnection between boards is considered the weak link in the grounding scheme. On future designs it is recommended that larger wire be used for ground interconnection and that a pin near the center of the board, such as pin 34, be used as the ground pin. Also the B+ voltage pin should be near the center of the board.

4) The rise time of the logic signals depends upon the loading and the stray capacitance as described in the Fairchild data sheets. Whenever a low propagation delay is required, precaution must be taken to physically locate the source and the load close to one another, thus minimizing the stray capacitance. This was not always possible in the breadboard, and in three instances the stray capacitance was large enough to cause the system to malfunction. To remedy this situation a 3K resistor was placed from the signal line to B+. This reduces the time constant (load and stray capacitance) and shortens the rise time, thus decreasing the propagation delay of the signal. This problem can be eliminated by proper layout of the circuitry. However, it is possible that there would be instances in future designs where all fast circuits could not be placed close together and the additional 3K loading resistors would be required.

5) In several instances, in the breadboard, the loading on modules was in excess of the specified maximum (11 loads instead of specified maximum of 8). The purpose of doing this was to evaluate the modules under these conditions and determine if the operation deteriorated. In all cases the modules operated as if they were driving a normal load. This indicates that a certain amount of design margin exists in the logic modules when used as specified. While the modules in the breadboard operated properly under excess loading, this is not recommended. The loading rules as specified in the Fairchild data sheets should be followed.

Additional test data on the Fairchild DTμL logic family may be found in References 12 and 13.

Section 7
RECOMMENDATIONS

The results of this work indicate several areas which warrant additional breadboarding and new or additional investigation. Recommended future work is discussed below:

7.1 Additions to the ZVA/FVA Breadboard

Make additions to the advanced compression algorithm breadboard system to achieve the general performance that would be desired of the later flight system. Emphasis would be placed on a capability of processing post-flight data and monitoring of performance. However, as much of the breadboard addition as is practical would be designed to perform functions required of a flight system and provisions would be made for further revision or expansion to include all functions of a flight system. Subtasks are:

7.1.1 Specifications

Develop and write preliminary performance specifications.

7.1.2 Processing Logic

Add to ZVA/FVA breadboard the additional logic required to increase its functional capability to include functions which are considered applicable for a vehicle-borne data compression system. This includes additions in the areas of: (a) input timing, (b) self-contained high-speed clock, (c) redundant sample routine logic, and (d) generation of code words.

7.1.3 Reference Memory

Add a reference memory to the ZVA/FVA breadboard to permit processing of data from multiple sources. The functional areas included in this subtask are:

(a) channel address counter, (b) memory addressing logic, (c) gating for memory data input, (d) memory control logic and (e) memory core arrays along with current drivers and sense circuits. Also investigate the feasibility of increasing the NDRO storage in the reference memory to provide an all-stored addressing scheme. A reference memory with a stored-address capability would eliminate the need to reprogram the reference memory by hardwire changes when the multiplex format is changed. The reprogramming would be accomplished by simply changing the addresses stored in the reference memory.

7.1.4 Buffer Memory

Add a buffer memory and associated circuits to buffer the accepted data samples and provide a compressed PCM output at a constant output rate. Logic will be included to prevent buffer overflow and underflow and to provide outputs indicating buffer fullness. The functional areas included in this subtask are: (a) memory core arrays with current drivers and sense amplifiers, (b) "read" and "write" logic, (c) buffer input gating, (d) output timing, (e) output register, (f) fullness monitoring and detection logic.

7.1.5 Special Input Test Equipment

This subtask includes the design, fabrication and test of special input test equipment for use with the ZVA/FVA breadboard. The primary function of this test equipment is to provide a means for entering information into the NDRO part of the reference memory.

7.1.6 Output Monitoring Equipment

This subtask includes the design, fabrication and test of special equipment for use with the ZVA/FVA breadboard, capable of detection, holding, displaying, and D-to-A conversion of information from any two data channels.

7.2 Computer Simulation

Simulate the FVA algorithm, as implemented in the breadboard, on a computer to determine the effects on system performance of the modifications made on the FVA algorithm. This includes evaluating the limited run length, the approximate division including the discarding of any decimal fractions, and the limited range of the corridor boundaries. These results could also be obtained by processing post-flight data through the breadboard.

7.3 Analysis of Reconstructor Requirements for ZVA/FVA Compression Algorithms

Perform an engineering analysis of the functional requirements of a reconstructor system to reconstruct data which has been compressed using a ZVA and/or FVA compression algorithm. This task would include a description of the functional requirements of the reconstructor.

Section 8
LIST OF REFERENCES

1. Simpson, Griffen, James and Bjorn, A Study of Redundancy Removal for Saturn Telemetry Data, Tech. Report #5, Systems Engineering Group, University of Alabama, Nov 1964.

2. Lockheed Missiles & Space Company, Techniques and Applications of Data Compression, LMSC 8-51-63-2, Jun 1963.

3. Lockheed Missiles & Space Company, Operation Manual - Advanced Telemetry System With Adaptive Capability (Data Compressor), LMSC 8-39-65-4, 1 Nov 1965.

4. Lockheed Missiles & Space Company, Advanced Algorithm Investigation - High Speed Integrated Circuit Survey, LMSC 8-39-65-3, 19 Aug 1965.

5. Fairchild Semiconductor, DTμL Composite Data Sheet, SL-17, Sept 1965.

6. Fairchild Semiconductor, DTμL 930 Dual Gate Element, Dec 1964.

7. Fairchild Semiconductor, DTμL 932 Dual Buffer Element, May 1965.

8. Fairchild Semiconductor, DTμL 933 Dual Four-Input Extender Element, Jul 1964.

9. Fairchild Semiconductor, DTμL 945 DTμL 948 Clocked Flip-Flop, Mar 1965.

10. Fairchild Semiconductor, DTμL 946 Quad Two-Input Gate Element, Aug 1964.

11. Fairchild Semiconductor, Diode-Transistor Micrologic, Application Bulletin 107 Rev. 1, May 1965.

12. Naval Air Development Center, Crane, Indiana, Information on Microelectronics for Navy Avionics Equipment U-Notes No. 7, NAD Crane Report No. QE/C 64-469, 1 Aug 1964.

13. Naval Air Development Center, Crane, Indiana, Information on Microelectronics for Navy Avionics Equipment U-Notes No. 13, NAD Crane Report No. 65-574, 1 Aug 1965.

14. Lockheed Missiles & Space Company, Preliminary Implementation Studies of Advanced Compression Algorithms, LMSC 8-39-65-1, 30 Jun 1965

15. Lockheed Missiles & Space Company, Annual Technical Report - Advanced Telemetry System With Adaptive Capability, LMSC 657617, 30 Jul 1963.

APPENDIX A
EXPLANATION OF SYMBOLS

This Appendix contains explanations of the symbols used in the wiring tables and schematic diagrams.

Data Input

The ten data input wires are described as INPUT 2^9, INPUT 2^8, etc. These signals originate in the test panel.

Sense Amplifier Outputs

The outputs of the reference memory sense amplifiers are described as SA Y'_n 2^2, SA Y_o 2^7, etc. The Y'_n and Y_o refer to the data word and the 2^2, 2^7 refer to the bit location. In the breadboard there is no reference memory, and these signals originate in the test panel.

Test Panel Signals

When test panel switch S_2 is in position 1, TP16 and $\overline{TP16}$ are present. TP16 is a "1" during logic step 16 and a "0" at all other times. $\overline{TP16}$ is a "0" during logic step 16 and a "1" at all other times. If test panel S_2 is not in position 1, TP16 is a "0" and $\overline{TP16}$ is a "1" at all times.

ZVA is a "1" when the data is being processed by the ZVA algorithm and a "0" otherwise. FVA is a "1" when the data is being processed by the FVA algorithm and a "0" otherwise. Both the FVA and ZVA wires originate in the test panel.

$\overline{FA}$ - when this wire is a "0", an artificial data sample will be calculated for some reason other than the normal redundancy test.

If the L Pri wire is a "1", the data channel is a low priority channel. In the breadboard this wire originates in a test panel switch. If the buffer wire is a "1", the buffer is full. In the breadboard this wire originates in a test panel switch.

Timing Signals

Examples of timing signals are 10_5, $\overline{10\ 10_1\ 10_2}$, 10_T, $6 + 7 + 88_2$. The symbol 10_5 means that the wire is a "1" during logic step 10_5 and a "0" at all other times. The symbol $\overline{10\ 10_1\ 10_2}$ means that the wire is a "1" at all times except logic during steps 10, 10_1 or 10_2 and a "0" during these logic steps. The symbol 10_T means that the wire is a "1" at all times that the 16-state generator is in state 10, regardless of which path of the flow diagram is being followed, and a "0" at all other times. The symbol $6 + 7 + 88_2$ means that the wire is a "1" during logic steps 6, 7, 8 or 8_2 and a "0" at all other times. The above examples illustrate the various formats that describe timing symbols.

16 State Generators

The "1" outputs of the four stages of the 16-state generator, located on board X, are symbolized by 2^0, 2^1, 2^2 and 2^3, while the "0" outputs are symbolized by $\overline{2^0}$, $\overline{2^1}$, $\overline{2^2}$ and $\overline{2^3}$.

Control Signals

P - the "1" side of the priority flip flop. When this wire is a "1", the data sample has been rejected because of low priority. $\overline{P}$ is the "0" side of the priority flip flop.

7-11 yes is the "yes" side of the 7-11 control flip flop. If the answer to logic step 7 or logic step 11 is yes, flip flop 7-11 is set such that the 7-11 yes wire is a "1". 7-11 no is the zero side of the 7-11 flip flop.

CFF_* is the "1" output of the CFF_* flip flop.
$\overline{CFF}_*$ is the "0" output of the CFF_* flip flop.

If the answer to one or more of the questions asked during logic steps 6, 5 or 2 is yes, control flip flop yes 2, 5, 6 is set such that the yes 2, 5 or 6 output is a "1".

If the T wire is a "1", the decision to calculate an artificial data point has been made. The logic equation for T is $\overline{P}$·Yes 2, 5 or 6.

Register Outputs

There are seven 10-bit shift registers in the breadboard. They are registers A, B, C, Y_n, Y'_n, Y''_n and Y_o. The least significant flip flop of the shift register is described as the 2^0 flip flop, the next least significant flip flop being the 2^1 flip flop, on up to the most significant flip flop which is called 2^9 flip flop. When these points are described in the wiring list and the schematics, the register name is listed first, followed by the position in the register. For example, the "1" output of the least significant stage of the A register is described as $A2^o$, while the "0" output of the same flip flop is described as $\overline{A2^o}$. Likewise Y_n2^6, $Y''_n\overline{2^9}$, etc.

Adder-Subtractor Signals

SUB and $\overline{SUB}$ are the outputs of the adder-subtractor. To A ADD, to $\overline{A}$ ADD, to B ADD and to $\overline{B}$ ADD are the inputs to the adder-subtractor. CARRY and $\overline{CARRY}$ are the carry (borrow in the case of subtraction) outputs of the adder-subtractor.

Divide Signals

÷ CLOCKS are the clock pulses used in the divide operation in logic steps 9 or 13.

Board X Signals

The definitions of the following can be found in the notes of schematic board X.

"10 CLOCKS"
11th PULSE
CLOCK TIME 1-11
$\overline{\text{CLOCK TIME 1-11}}$
DATA READY
CLOCK TIME 11

Other Signals

NODE ALL "0" is the common connection of the outputs of the diode expander gates that detect when the data in both registers Y'_n and Y''_n are all zeros.

$\overline{B \rightarrow 0}$ is a "0" when register B is being set equal to zero.

8, 12 + is the "1" output of the 8, 12 flip flop. 8, 12 - is the "0" output of the 8, 12 flip flop.

Y'_n 0 is a "1" when register Y'_n is being set equal to zero.

$\overline{Y'_n \rightarrow 1}$ is a "0" when register Y'_n is being set equal to full scale.

NODE ALL "1" is the common connection of the outputs of the diode expander gates that detect when the data in both registers Y'_n and Y''_n are full scale.

$\overline{Y'_n \rightarrow 0}$ is a "0" when register Y'_n is being set equal to zero.

If the wire C→Sub is a "1", the data in the C register is applied to the B inputs of the subtractor.

If the wire C Shift is a "1", 10 shift pulses are applied to register C.

$N \neq 63$ is a "1" when the state of the N counter is not 63.

$\overline{A \rightarrow 1}$ is a "0" when register A is being forced to full scale.

$\overline{A \rightarrow 1_D}$ is $\overline{A \rightarrow 1}$ delayed by two inverter stages.

A CLOCK is a "1" when register A is being shifted by 10 clocks.

Appendix B

This Appendix contains the interconnection wiring lists for the breadboard and the test panel. Pages B-2 through B-14 contain the interconnections between circuit boards in the breadboard. Page B-15 is the wiring list between board Z in the breadboard and the test panel. Pages B-16 through B-19 contain the test panel wiring lists.

LMSC 669224

INPUT BOARD Z

A	FUNCTION	TO		REMARKS
1	INPUT 2^9	F30		
3	+8			
5	INPUT 2^6	F51		
7	INPUT 2^4	D54		
9	INPUT 2^2	D52		
11	INPUT 2^0	F53		
13	SA Yn" 2^8	V28	H7	
15	SA Yn" 2^6	H48		
17	SA Yn" 2^4	B2		
19	SA Yn" 2^2	B6		
21	SA Yn" 2^0	H36		
	OUTPUT 2^8	F1		
	OUTPUT 2^6	F13		
27	OUTPUT 2^4	D36		
	OUTPUT 2^2	D21		
31	OUTPUT 2^0	F41		
33	SA Yn 2^8	P40		
35	SA Yn 2^6	P52		
37	SA Yn 2^4	AA15		
39	CLOCK INPUT	X37		
41	TEST PAN 16	H39	B22	
43	DATA READY	X43		
45	FVA	V49	T53	
47	$\overline{16}$	T22	AA11	
49	SA Yn 2^3	AA2		
51	SA Yn 2^1	AA4		
53	SA Yo 2^9	H56		
55	SA Yo 2^7	D40		
57	SA Yo 2^5	D35		
59	SA Yo 2^3	H57		
61	SA Yo 2^1	H45		
63	$\overline{C\ 2^1 \rightarrow 1}$	M38		
65	GND			

FROM	FUNCTION	TO	REMARKS
2	INPUT 2^8	F32	
4	INPUT 2^7	F52	
6	INPUT 2^5	D57	
8	INPUT 2^3	D51	
10	INPUT 2^1	D53	
12	SA Yn 2^9	V12	H5
14	SA Yn" 2^9	H49	
16	SA Yn" 2^5	B1	
18	SA Yn" 2^3	B5	
20	SA Yn" 2^1	B8	
22	OUTPUT 2^9	F5	
24	OUTPUT 2^7	F11	
26	OUTPUT 2^5	D28	
28	OUTPUT 2^3	D16	
30	OUTPUT 2^1	D34	
32	SA Yn 2^9	P38	
34	SA Yn 2^7	P41	
36	SA Yn 2^5	AA12	
38	BUFFER F	T13	
40	L PRI	T14	
42	$\overline{FA}$	T19	P20 AA64
44	TEST PAN $\overline{16}$	X6	
46	ZVA	T42	V19 X32
48	$\overline{T}$	V66	
50	SA Yn 2^2	AA13	
52	SA Yn 2^0	P48	
54	SA Yo 2^8	D43	
56	SA Yo 2^6	D38	
58	SA Yo 2^4	D37	
60	SA Yo 2^2	H59	
62	SA Yo 2^0	H43	
64	$\overline{C\ 2^2 \rightarrow 1}$	M56	
66	$\overline{C\ 2^3 \rightarrow 1}$	M54	

REVISIONS

WIRING DIAGRAM– MATRIX INTERCONNECTION LIST

WIRING LIST

BREADBOARD MATRIX ROW Z

B-2

ATWAC

CODE 66887 SHEET

PC BOARD X

IMSC-660224

A	FUNCTION	TO		REMARKS		FROM	FUNCTION	TO	REMARKS
1						2	T_3	R6	B32
3	B+					4	$\overline{10_3}$	V21	R63, P66 F33, M4C A4C
5	11th pulse	T1	H18			6	TP·T6	P34	H21
7	CLOCK TIME I-II	V59	T29			8	$10_3 + 16$	K66	
9	2^0	V64	T66			10			
11	$\overline{11^{th}}$ PULSE	M13				12			
13						14			
15	$(3+T_3)\overline{CT\cdot 11}$	R10				16			
17						18			
19	$\overline{2^1}$	V54	T36			20	CFF*	P14	M14
21	11th pulse	B33	AA33			22	$\overline{CFF*}$	M8	
23	10 CLOCKS	D32	B30			24			
25	11th pulse	R1	K1			26			
27	$\overline{2^0}$	V55	T37			28			
29	2^1	V51	T63			30	$\overline{ZVA\cdot 10_3\cdot TP16}$	KCA	
31	10 CLOCKS	M11	K18	H34, F43		32	ZVA	Z46	T42, V19
33	10 CLOCKS	R48	P46	AA30		34			
35	11th pulse	P32				36			
37	CLOCK INPUT					38			
39	$\overline{2^2}$	V62	T62			40			
41	$\overline{2^3}$	V63				42			
43	DATA READY	Z43				44			
45	$\overline{T_3}$	P63	V40	R7, H27	AA54	46			
47	$\overline{3}$	V43	T16	R35, F45		48			
49						50			
51						52			
53						54			
55						56			
57						58			
59	2^2	V53	T35			60			
61	CLOCK II TIME	R15				62			
63	2^3	V52	T64			64			
65	GND					66			

REV STATUS

WIRING DIAGRAM– MATRIX INTERCONNECTION LIST

WIRING LIST
BREADBOARD MATRIX ROW X
B-3

ATWAC

CODE 04887 | SHEET

11-66

PC BOARD V

LMSC-669224

A	FUNCTION	TO		REMARKS		FROM	FUNCTION	TO	REMARKS
1	$\overline{11_3}$					2	$T \cdot FVA$	T61	
3	B+					4	11_3	D18	
5	10_5	P26				6	11_3	R16	R27 F7
7	$\overline{7_T}$	P30	R39			8	$6+7_T+8B_2$	P13	
9	$5+7$	R64				10	$Y_n'' \rightarrow 1$	H40	
11	10_4	F6				12	SA Y_n'' 2^4	H5	Z12
13	$\overline{11_3}$	T20	R11	F55		14	$9, 9_3$	P43	AA19
15	$\overline{4}$	T10	F63			16	Yes 2,5nG	T5	
17	$(4+10_3)\overline{FA}$	F10	V17			18	7-11 yes	T55	
19	$\overline{ZVA}$	T42	Z46	X42		20	$\overline{T}$	T54	
21	$\overline{10_3}$	R63	P66	F33,X4	AA6	HA2 22	$\overline{8, B_3}$	T6	P25 R66
23	$\overline{P}$	T17				24			
25	$7+8B_2$	P56				26	Yes 2,5ORG	T12	
27	$\overline{8B_2}$	R37	P44			28	SA Y_n'' 2^3	H7	Z13
29	$\overline{10_5}$	P39				30	$\overline{P \cdot FVA \cdot T \cdot 11_{no}}$	T45	
31	$7-11_{no}$	T43				32	$\overline{10_4}$	F34	
33						34	$T \cdot ZVA$	T34	
35	$\overline{10\ 10, 10_2}$	R57	P60	F26, AA8		36	$2+11_3$	K54	AA16
37	$\overline{7}$	T30	R17			38	$\overline{X1}$	T29	AA50, H33, F35
39	$10\ 10, 10_2$	P55				40	$\overline{7_3}$	P63	R7, H27, AA54, X45
41	$\overline{8_T}$	P37	H54			42			
43	$\overline{3}$	T16	R35	F45, X47		44	$\overline{10_T}$	R52	
45	$\overline{9_T}$	T25				46	$3+7_3$	R9	
47	$\overline{9, 9_3}$	AA18				48	P	T31	
49	FVA	T53	Z45			50	$\overline{79_2}$	R49	
51	$2'$	X29	T63			52	2^3	X63	T64
53	2^2	X59	T35			54	$\overline{2^T}$	X19	T36
55	$\overline{2^0}$	X27	T37			56	$\overline{2}$	T2	
57	FA	AA22				58	$\overline{5}$	T28	R18 H31 AA51
59	CLOCK TIME 1-11	X5	T24			60	$\overline{6}$	T27	P64 F37
61	$\overline{15_C}$	H44	F25	AA42		62	$\overline{2^2}$	X39	T62
63	$\overline{2^3}$	X41				64	2^0	X9	T66
65	GND					66	$\overline{1}$	Z48	

REVISIONS

WIRING DIAGRAM– MATRIX INTERCONNECTION LIST

WIRING LIST

BREADBOARD MATRIX ROW V

B-4

ATWAC

CODE 86887 SHEET

PC BOARD T

LMSC 669224

M	FUNCTION	TO		REMARKS
1	11th pulse	X5	H18	
3	B+			
5	Yes 2,5,6	V16		
7	$\overline{13_T}$	R12		
9	C → SUB	M51		
11	CFF*	R5	M9	H19
13	BUFFER F	Z38		
15	$\overline{2}$			
17	$\overline{P}$	V23		
19	$\overline{FA}$	Z42	P20	AA64
21	$\overline{12_T}$	P35	R45	
23	$14_3 + 14_4$	H11		
25	$\overline{9_T}$	V45		
27	$\overline{6}$	V60	P64	F37
29	$\overline{11}$	V38	AR_{50}	H33, F35
31	P	V48		
33	$\overline{13_3}$	R33	F49	
35	2^2	V53	X59	
37	$\overline{2^0}$	V55	X27	
39	$\overline{14_T}$	R41	F38	
41	$\overline{19_5}$	AA45		
43	7-11 mc	V31		
45	$\overline{P \cdot FUA \cdot T \cdot 11mc}$	V30		
47	$\overline{14_4}$	AA56	F64	
49	$\overline{12\,13,15_2}$	R47		
51	$\overline{14,14_2}$	AA52	R53	H35
53	FVA	V49	Z45	
55	7-11 sec	V18		
57	12_4	P16		
59	14_4	H13		
61	T·FVA	V2		
63	2'	V51	X29	
65	GND			

FROM	FUNCTION	TO	REMARKS
2	$\overline{2}$	V56	
4	CFF*	R13	P2, M4 H22
6	$\overline{8B_3}$	V22	P25 R66
8	C SHIFT	M17	B18
10	$\overline{4}$	V15	F63
12	Yes 2,5,6	V26	
14	L PRI	Z40	
16	$\overline{3}$	V43	R35 F45, X47
18	NF.63	K30	
20	$\overline{11_3}$	V13	H11, F55
22	$\overline{16}$	Z47	AA11
24	CLOCK TIME 1-11	V59	X5
26	$9_T + 13\,13, 13_2$	K19	
28	$\overline{5}$	V58	K18 H61 AA51
30	$\overline{7}$	V37	R17
32	14_5	H9	
34	T·ZVA	V34	
36	$\overline{2'}$	V54	X19
38	$14 + 14, 14_2$	H1	
40	$\overline{19}$	AA47	R55, F31
42	ZVA	V19	Z46 X32
44	16	R21	M7 F47
46	$\overline{12}$	R27	H53 F37
48	$\overline{14_3}$	AA46	R59, H63, F29
50	16	D17	B21
52	$12, 12_2$	AA49	R29 H29
54	$\overline{T}$	V20	
56	14_3	P23	H52
58	$\overline{12_5}$	P28	R61, F36, AA17
60	$\overline{12_4}$	P33	F54
62	$\overline{2^2}$	V62	X39
64	2^3	V52	X63
66	2^0	V64	X9

REV STATUS

WIRING DIAGRAM–MATRIX INTERCONNECTION LIST

WIRING LIST
BREADBOARD MATRIX ROW T

B-5

ATWAC

CODE 06887 | SHEET

4-11-66

PC BOARD R

IMSC 669224

FROM	FUNCTION	TO		REMARKS
1	11th pulse	X25	K1	
3	B+			
5	CFF*	M9	H19	T11
7	$\overline{7_3}$	P63	V40	H27, AH54 X45
9	$3 + 7_3$	V46		
11	$\overline{11_3}$	V13	T20	F55
13	$\overline{CFF_*}$	P2	M4	H22, T4
15	CLOCK "TIME"	X61		
17	$\overline{7}$	R17	V37	
19				
21	16	T44	M7	F47
23				
25				
27	$\overline{12}$	T46	H53	F39
29	$\overline{12, 12_2}$	T52	AA49	H29
31	SUB	M47	P12	
33	$\overline{13_3}$	T33	F49	
35	$\overline{3}$	T16	V43	F45, X4
37	$\overline{88_2}$	V27	P44	
39	$\overline{7_T}$	P30	V7	
41	$\overline{14_T}$	T39	F38	
43	$99_2 + 1313, 13_2$	B31		
45	$\overline{12_T}$	P35	T21	
47	$\overline{1313, 13_2}$	T49		
49	$\overline{99_2}$	V50		
51				
53	$\overline{14, 14_2}$	AH52	T51	H35
55	$\overline{14}$	AA47	T40	F31
57	$\overline{10\ 10, 10_2}$	P60	V35	F26, AA3
59	$\overline{14_3}$	AH46	T48	H63, F29
61	$\overline{12_3}$	P28	T58	F36, AA17
63	$\overline{10_3}$	V21	P66	F33, X4 AA6
65	GND			

FROM	FUNCTION	TO	REMARKS
2	$\overline{A \to 1}$	B27	
4	$\overline{A \to 1_D}$	B34	
6	7_3	B32	X2
8	$A\ 2^{10}$	M35	
10	$(3+2_3)\overline{CT\ 11}$	X15	
12	$\overline{13_T}$	T7	
14	$A \cdot \overline{2^{10}}$	M19	
16	11_3	V6	P27 F7
18	$\overline{5}$	V58	T28 H31 AA51
20	$\overline{SUB}$	M31	P5
22			
24			
26			
28			
30			
32	$A\ \overline{2^2}$	B36	
34	$A\ \overline{2^7}$	B35	
36	÷ CLOCKS	P36	B29 AA20
38	A CLOCK	B28	
40	$A\ 2^2$	B48	
42	$A\ 2^7$	B47	
44	$9, 9_3 + 12_3$	AA21	
46	13_3	P31	
48	10 CLOCKS	X33	F46 AH30
50	To H ADD	P53	M53, H50, FG6
52	$\overline{10_T}$	T44	
54	To B ADD	P57	M55, H37, F17
56	$A\ \overline{2^0}$	P8	F4
58	$A\ 2^0$	P7	F8
60	To $\overline{B}$ ADD	P62	M55, H51, F21
62	To A ADD	P61	M63, H41, FG1
H4264	5 + 1	V9	
66	$\overline{8, 63}$	P25	V22 T6

REV STATUS

WIRING DIAGRAM— MATRIX INTERCONNECTION LIST

WIRING LIST
BREADBOARD MATRIX ROW R
B-6

ATWAC

CODE 06887 | SHEET

11-66

PC BOARD P

LMSC 669224

FROM	FUNCTION	TO	REMARKS
1	CARRY	M5	
2	$\overline{CFF_*}$	R13	M4 H22, T4
3	B_T		
4	$8,12 = *$	H2	
5	$\overline{SUB}$	R20 M31	
6			
7	$A\,2^0$	R58 F8	
8	$A\,\overline{2^0}$	R56	F4
9	$\overline{10\ CLOCKS}$	M10	
10			
11	$8,12 = -$	H17	
12	SUB	R31	M47
13	$G_T 7_T + 88_2$	V8	
14	CFF_A	X20	M14
15	CARRY	M15	
16	12_4	T57	
17	$19_2 \cdot 11^{th}$ pulse	M32	
18			
19	NODE ALL "0"	H6 AA34 B9	
20	$\overline{FA}$	AA64	T19, Z42
21	Y_n' CLOCK	AA31	
22	$Y_n' \rightarrow 0$	AA1	
23	19_3	T56 H52	
24	$\overline{Y_n \rightarrow 1}$	M41	AA14
25	$\overline{8,8_2}$	T6 V22	R66
26	10^5	V5	
27	11_3	V6 R16	F7
28	$\overline{12_3}$	T58	R61 F36, AH17
29	$(11_3 + 13_3)$ 11^{th} pulse	M37	
30	$\overline{7_T}$	V7	R39
31	13_3	R46	
32	11^{th} pulse	X35	
33	$\overline{T2_4}$	T60 F54	
34	T.P. T6	X6	H21
35	$\overline{12_T}$	T21 R45	
36	÷ CLOCKS	R36	B29, AA20
37	$\overline{B_T}$	V41 H54	
38	SA X_n' 2^9	Z32	
39	$\overline{10_5}$	V29	
40	SA Y_n' 2^8	Z33	
41	SA Y_n' 2^7	Z34	
42	NODE ALL "1"	AA64	
43	$9,9_3$	V14 AAA	
44	$\overline{88_2}$	R37	V27
45	Y_n' $\overline{2^7}$	AA38	
46	10 CLOCKS	R48	X33, AA30
47	$\overline{B \rightarrow 0}$	D29	
48	SA Y_n' 2^0	Z52	
49	4	F14	
50	Y_n' 2^1	AH48	
51	Y_n' $\overline{2^6}$	AA53	
52	SA Y_n' 2^6	Z35	
53	$\bar{A}$ ADD.	F66 H50	M53, R50
54	$12_3 + 8,8$		
55	$10\ 10,10_2$	V39	
56	$7 + 88_2$	V25	
57	B ADD	F17 H37	M66, R54
58	$\overline{Y_n' \rightarrow 0}$	M23	
59	Y_n' 2^6	AA63	
60	$\overline{10\ 10,10_2}$	R57	V35 F26, AH8
61	A ADD	F61 H41	M63, R62
62	$\overline{B}$ ADD	F21	H51, M55, R60
63	$\overline{5_3}$	V40 R7	H27, AA54, X45
64	$\overline{6}$	V60	F37 T27
65	GND		
66	$\overline{10_3}$	V21	R63 F33, X4, AH6

REV STATUS H42

WIRING DIAGRAM— MATRIX INTERCONNECTION LIST

WIRING LIST

BREADBOARD MATRIX ROW P

B-7

ATWAC

CODE 86887 SHEET

1-11-66

PC BOARD M

IMSC 669224

ROM	FUNCTION	TO		REMARKS
1	$\overline{CFF}$			
3	B+			
5	$\overline{CARRY}$	P1		
7	16	T44	R21	F47
9	CFF_*	R5	H19	T11
11	10 CLOCKS	X31	K18	H34, F43
13	$\overline{11^{th}\ pulse}$	X11		
15	CARRY	P15		
17	C SHIFT	T8	B18	
19	$A\ \overline{2^{10}}$	R14		
21	$\overline{Y_n'' \rightarrow 0}$	H12		
23	$\overline{Y_n' \rightarrow 0}$	P58		
25				
27				
29	$\overline{Y_n'' \rightarrow 2^{10}-1}$	H38	B17	
31	$\overline{SUB}$	R20	P5	
33	$\overline{SUB}$	H4	F12	
35	$A\ 2^{10}$	R8		
37	$(11_3+13_3)\ 11^{th}$ pul	P29		
39	SUB	H15	F9	
41	$\overline{Y_n' \rightarrow 2^{10}-1}$	P24	AAM	
43				
45	$\overline{12_3 \cdot 11^{th}}$ pulse	AA32		
47	SUB	R31	P12	
49				
51	C→SUB	T9		
53	$\overline{A}$ - ADD.	F66	H50	P53, R50
55	$\overline{B}$ - ADD	F21	H51	P62, R60
57				
59	2			
61				
63	A - ADD	F61	H41	P61, R62
65	GND			

FROM	FUNCTION	TO	REMARKS
2			
4	$\overline{CFF_M}$	P2	R13, H22, I4
6			
8	$\overline{CFF_*}$	X22	
10	$\overline{10\ CLOCKS_0}$	P9	
12	CFF		
14	CFF_M	P14	X20
16	C CLOCKS		
18			
20			
22			
24			
26			
28			
30			
32	$\overline{19_3 \cdot 11^{th}\ pulse}$	P17	
34			
36	$\overline{C\ 2^0 \rightarrow 1}$		
38	$\overline{C\ 2^1 \rightarrow 1}$	Z63	
40			
42			
44			
46			
48	$C\ \overline{2^4}$	B56	
50	$\overline{C\ 2^9 \rightarrow 1}$		
52	$C\ \overline{2^9}$	B49	
54	$\overline{C\ 2^3 \rightarrow 1}$	Z66	
56	$\overline{C\ 2^2 \rightarrow 1}$	Z64	
58			
60			
62	$C\ 2^4$	B64	
64	$C\ 2^9$	B66	
66	B-ADD	F17	H37 P57 R54

REV STATUS

WIRING DIAGRAM-MATRIX INTERCONNECTION LIST

WIRING LIST
BREADBOARD MATRIX ROW M

B-8

ATWAC

CODE 06887 | SHEET

1-11-66

PC BOARD K

IMSC 669224

FROM	FUNCTION	TO		REMARKS	FROM	FUNCTION	TO	REMARKS
1	11 th PULSE	X25	R1		2	S.R. N 2^5		
3	B+				4			
5					6			
7					8			
9					10			
11					12			
13					14			
15	$\div$ CLOCKS	AA5			16			
17	SA N 2^4				18	10 C. CLOCKS	X31	M11, H34, F43
19	9_T + 13 13, 13_2	T26			20			
21					22			
23					24			
25					26			
27					28			
29					30	N ≠ 63	T18	
31					32			
33	SA N 2^0				34			
35	SA N 2^3				36			
37					38			
39					40			
41					42			
43					44			
45					46			
47					48			
49					50			
51					52			
53	SA N 2^1				54	2 + 11 3	V36	AA16
55	SA N 2^2				56			
57					58			
59					60			
61					62			
63					64	$\overline{2VA \cdot 10 \cdot TP1C}$	X30	
65	GND.				66	10_3 + TP16	X8	

REV STATUS

WIRING DIAGRAM— MATRIX INTERCONNECTION LIST

WIRING LIST
BREADBOARD MATRIX ROW K
B-9

ATWAC

CODE 06887 | SHEET

1-11-66

IMSC 669224

BOARD H

FROM	FUNCTION	TO		REMARKS
1	$14 + 14_1 M_2$	T38		
3	$B+$			
5	SA Y_n'' 2^9	V12	Z12	
7	SA Y_n'' 2^8	V28	Z13	
9	14_5	T32		
11	$14_3 + 14_4$	T23		
13	14_4	T59		
15	SUB	M39	F9	
17	8, 12 = −	P11		
19	CFF*	M9	R5	T11
21	TP $\overline{16}$	P34	X6	
23	$Y_n'' \rightarrow 0$	B20		
25	Y_n'' 2^7	AA60		
27	$\overline{7_3}$	R7	V40	P63, AA54 X45
29	$\overline{12, 12_2}$	R29	AA46	T52
31	$\overline{5}$	R18	T28	V58, AA51
33	$\overline{11}$	T29	V38	AA50, F35
35	$\overline{14, 14_2}$	R53	T51	AA52
37	B ADD	F17	M66	P59 R54
39	TP16	B22		
41	A ADD	F61	M63	P61, R62
43	SA Y_0 2^0	Z62		
45	SA Y_0 2^1	Z61		
47	$Y_n'' 2^0$	AA59		
49	SA Y_n'' 2^7	Z14		
51	$\overline{B}$ ADD	F21	M55	P62, R60
53	$\overline{12}$	R27	T46	F39
55	$B2^0$	F2		
57	SA Y_0 2^3	Z59		
59	SA Y_0 2^2	Z60		
61	Y_0 2^4	D41		
63	$\overline{14_3}$	F29	R59	T48, AA46
65	GND			

FROM	FUNCTION	TO	REMARKS
2	8, 12 = +	P4	
4	$\overline{SUB}$	M33	F12
6	NODE ALL "0"	AA34	P19 B9
8	Y_n'' 2^9	AA62	
10	Y_n'' 2^8	AA61	
12	$\overline{Y_n'' \rightarrow 0}$	M21	
14	INPUT Y_n & $Y_n'' 2^9$	F46	AA27
16	Y_n'' $\overline{2^1}$	B7	
18	$11^{\pm 4}$ pulse	T1	X5
20	Y_n'' $\overline{2^6}$	B4	
22	$\overline{CFF*}$	M4	P2, R13, T4
24	INPUT Y_n & Y_n^9	F42	AA36
26	Y_n'' 2^1	B12	AA37
28	Y_n'' 2^6	B13	AA35
30			
32	Y_n CLOCK	AA55	B16
34	10 CLOCKS	X31	M11, K18, F43
36	SA Y_n'' 2^0	Z21	
38	$\overline{Y_n'' \rightarrow 2^0 - 1}$	M29	B17
40	$Y_n'' \rightarrow 1$	V10	
42	$\overline{10_3}$	AA6	F33, P66, R63, V21, X4
44	$\overline{15_5}$	F25	AA42, V61
46	Y_n'' $\overline{2^0}$	AA40	
48	SA Y_n'' 2^6	Z15	
50	$\overline{A}$ ADD	F66	M53, P53, R50
52	14_3	P23	T56
54	$\overline{8_T}$	P37	V41
56	SA Y_0 2^9	Z53	
58	Y_0 2^1	D42	
60	Y_0 $\overline{2^4}$	D39	
62	Y_0 2^9	D50	
64	$B\ \overline{2^0}$	F15	
66	Y_0 CLOCK	D31	

REV STATUS

WIRING DIAGRAM— MATRIX INTERCONNECTION LIST

WIRING LIST
BREADBOARD MATRIX ROW H

B-10

ATWAC

CODE 06887 | SHEET

-11-66

PC BOARD F

IMSC 669224

FROM	FUNCTION	TO		REMARKS		FROM	FUNCTION	TO	REMARKS
1	OUTPUT 2^8	Z23				2	B 2^0	H55	
3	B+					4	A $\overline{2^0}$	P8	R56
5	OUTPUT 2^9	Z22				6	10_4	V11	
7	11_3	P27	R16	V6		8	A 2^0	P7	R58
9	SUB	H15	M39			10	$(4+10_3)\overline{FA}$	V17	AA7
11	OUTPUT 2^7	Z24				12	$\overline{SUB}$	H4	M33
13	OUTPUT 2^6	Z25				14	4	P49	
15	B $\overline{2^0}$	H69				16	B 2^6	D27	
17	B ADD	H37	M66	P57, R54		18	B $\overline{2^1}$	D5	
19	B $\overline{2^6}$	D19				20	INPUT B 2^9	AA24	
21	B ADD	H51	M55	P62, R60		22	INPUT B $\overline{2^9}$	AA28	
23						24			
25	$\overline{15_5}$	AA42	H44	V61		26	$\overline{10_0, 10_2}$	AA8	P60, R52, V35
27	B → O_D	D23				28	B $\overline{2^1}$	D11	
29	$\overline{14_3}$	H63	R59	T48, AA46		30	INPUT 2^7	Z1	
31	$\overline{14}$	R55	T40	AA47		32	INPUT 2^8	Z2	
33	$\overline{10_3}$	P66	R63	V21, X4	AA6	H42 34	$\overline{10_4}$	V32	
35	$\overline{11}$	H33	AA50	V38, T29		36	$\overline{12_3}$	R61	T58 P26, AH17
37	$\overline{6}$	P64	V60	T27		38	$\overline{14_7}$	R41	T39
39	$\overline{12}$	H53	T46	R27		40	B CLOCK	D15	
41	OUTPUT 2^0	Z31				42	INPUT Y_n & Y_n "2"	AA36	H24
43	10 CLOCKS	H34	K18	M11, X31		44	INPUT Y_n 2^9	AA26	
45	$\overline{3}$	R35	V43	T16, X47		46	INPUT Y_n & Y_n "$\overline{2^9}$"	AH27	H4
47	16		M7	R21, T44		48	Y_n CLOCK	D33	
49	$\overline{13_3}$	R3	T33			50	Y_n $\overline{2^1}$	D55	
51	INPUT 2^6	Z5				52	INPUT 2^7	Z4	
53	INPUT 2^0	Z11				54	$\overline{12_4}$	P33	T60
55	$\overline{11_3}$	R11	T20	V13		56	Y_n $\overline{2^6}$	D56	
57	Y_n $\overline{2^0}$	AA10				58	INPUT Y_n $\overline{2^9}$	AA25	
59	Y_n 2^0	AA9				60	Y_n 2^1	D66	
61	A ADD	H41	M63	P61, R62		62	Y_n 2^6	D63	
63	$\overline{4}$	T10	V15			64	$\overline{14_4}$	AA56	T47
65	GND					66	$\overline{A}$ ADD	H50	M53, P53, R50

REV STATUS

WIRING DIAGRAM– MATRIX INTERCONNECTION LIST

WIRING LIST
BREADBOARD MATRIX ROW F
B-11

ATWAC

CODE 06887 | SHEET

1-11-66

P.C. BOARD D

IMSC 669224

FROM	FUNCTION	TO		REMARKS	FROM	FUNCTION	TO	REMARKS
1					2			
3	B+				4			
5	$B\overline{2'}$	F18			6			
7					8			
9					10			
11	B 2'	F28			12			
13					14			
15	B CLOCK	F40			16	OUTPUT 2^3	Z28	
17	16	T50	B21		18	11_3	V4	
19	$B\overline{2^8}$	F19			20			
21	OUTPUT 2^2	Z29			22			
23	$\overline{B \rightarrow O_8}$	F27			24			
25					26			
27	B 2^6	F16			28	OUTPUT 2^5	Z26	
29	$\overline{B \rightarrow O}$	P47			30			
31	Y_0 CLOCK	H66			32	10 CLOCKS	X23	B30
33	Y_n CLOCK	F48			34	OUTPUT 2'	Z30	
35	SH Y_0 2^5	Z57			36	OUTPUT 2^4	Z27	
37	SA Y_0 2^4	Z58			38	SA Y_0 2^6	Z56	
39	Y_0 $\overline{2^7}$	H60			40	SA Y_0 2^7	Z55	
41	Y_0 2^8	H61			42	Y_0 $\overline{2^9}$	H58	
43	SA Y_0 2^8	Z54			44			
45					46			
47					48			
49	T.P $\overline{T6}$	B23			50	Y_0 2^9	H62	
51	INPUT 2^3	Z8			52	INPUT 2^2	Z9	
53	INPUT 2'	Z10			54	INPUT 2^4	Z7	
55	Y_n $\overline{2'}$	F50			56	Y_n $\overline{2^6}$	F56	
57	INPUT 2^5	Z6			58			
59					60			
61					62			
63	Y_n 2^6	F62			64			
65	GND				66	Y_n 2'	F60	

REV STATUS

WIRING DIAGRAM– MATRIX INTERCONNECTION LIST

WIRING LIST
BREADBOARD MATRIX ROW D
B-12

ATWAC

CODE 06887 | SHEET

PC BOARD B

LMSC 669224

FROM	FUNCTION	TO		REMARKS		FROM	FUNCTION	TO	REMARKS
1	SA Y_n'' 2^5	Z16				2	SA Y_n'' 2^4	Z17	
3	B+					4	Y_n'' $\overline{2^6}$	H20	
5	SA Y_n'' 2^3	Z18				6	SA Y_n'' 2^2	Z19	
7	Y_n'' $\overline{2^1}$	H16				8	SA Y_n'' 2^1	Z20	
9	NODE ALL "0"	AA34	H6	P19		10	Y_n'' 2^5	AA58	
11	Y_n'' 2^4	AA57				12	Y_n'' 2^7	H26	AA37
13	Y_n'' 2^6	H28	AA35			14	Y_n'' 2^3	AA41	
15	Y_n'' 2^2	AA39				16	Y_n'' CLOCK	AA55	H32
17	$\overline{Y_n'' \rightarrow 2^{10-1}}$	H38	M29			18	C CLOCK	M17	T8
19						20	$Y_n'' \rightarrow 0$	H23	
21	16	D17	T50			22	TP 16	H39	
23	$\overline{T.P.16}$	D49				24			
25	C CLOCKS					26			
27	$\overline{A \rightarrow 1}$	R2				28	A CLOCK	R38	
29	÷ CLOCKS	AA20	P36	R36		30	10 CLOCKS	D32	X23
31	$99_2 + 1313,13_2$	R43				32	T_3	R6	X2
33	11th pulse	X21	AA33			34	$\overline{A_0 \rightarrow 1}$	R4	
35	$A\overline{Z^7}$	R34				36	$A\overline{Z^2}$	R32	
37						38			
39						40			
41						42			
43						44			
45						46			
47	$A Z^7$	R42				48	$A Z^2$	R40	
49	$C\overline{Z^9}$	M52				50			
51	$\overline{C Z^8 \rightarrow 1}$					52	$\overline{C Z^4 \rightarrow 1}$		
53	$\overline{C Z^7 \rightarrow 1}$					54	$\overline{C Z^5 \rightarrow 1}$		
55	$\overline{C Z^6 \rightarrow 1}$					56	$C Z^4$	M48	
57						58			
59						60			
61						62			
63						64	$C Z^4$	M62	
65	GND					66	$C Z^9$	M64	

REV STATUS

WIRING DIAGRAM– MATRIX INTERCONNECTION LIST

WIRING LIST
BREADBOARD MATRIX ROW B

ATWAC

CODE 66887 | SHEET

1-11-66

PC BOARD AA

IMSC 660224

FROM	FUNCTION	TO		REMARKS
1	$Y_n' \rightarrow 0$	P22		
3	B+			
5	÷ CLOCKS	K15		
7	$(4+10_2)\overline{FA}$	F10	V17	
9	$Y_n 2^0$	F59		
11	$\overline{16}$	T22	Z47	
13	SA $Y_n' 2^2$	Z50		
15	SA $Y_n' 2^4$	Z37		
17	$\overline{12_3}$	F36	R61	T58, P28
19	$9, 9_3$	P43	V14	
21	$9, 9_3 + 12_3$	R44		
23	$Y_0 \overline{2^0}$	H Connector		
25	INPUT $Y_n \overline{2^7}$	F58		
27	INPUT Y_n & $Y_n'' \overline{2^9}$	F46	H14	
29	$Y_0 2^0$	H Connector		
31	Y_n' CLOCK	P21		
33	11th pulse	B33	X21	
35	$Y_n'' 2^6$	B13	H28	
37	$Y_n'' 2^1$	B12	H26	
39	$Y_n'' 2^2$	B15		
41	$Y_n'' 2^3$	B14		
43	INPUT $Y_0 2^9$	H Connector		
45	$\overline{14_5}$	T41		
47	$\overline{14}$	F31	R55	T40
49	$\overline{12, 12_2}$	H29	R29	T52
51	$\overline{5}$	H31	R18	T28, V58
53	$Y_n' \overline{2^6}$	P51		
55	Y_n'' CLOCKS	B16	H32	
57	$Y_n'' 2^4$	B11		
59	$Y_n'' 2^0$	H47		
61	$Y_n'' 2^8$	H10		
63	$Y_n' 2^6$	P59		
65	GND			

FROM	FUNCTION	TO	REMARKS
2	SA $Y_n' 2^3$	Z49	
4	SA $Y_n' 2^1$	Z51	
6	$\overline{10_3}$	H42	F33, P66, R63, V21, X4
8	$\overline{10\ 10, 10_2}$	F26	P60, R57, V35
10	$Y_n \overline{2^0}$	F57	
12	SA $Y_n' 2^5$	Z36	
14	$Y_n' \rightarrow 1$	M41	P24
16	$2 + 11_3$	K54	V36
18	$\overline{9, 9_3}$	V47	
20	÷ CLOCKS	B29	P36 R36
22	FA	V57	(should be 3 input
24	INPUT B 2^9	F20	
26	INPUT $Y_n 2^9$	F44	
28	INPUT B $\overline{2^9}$	F22	
30	10 CLOCKS	P46	R48, X33
32	$\overline{12_3 \cdot 11}$ pulse	M45	
34	NODE ALL "0"	H6	P19 B9
36	INPUT Y_n & $Y_n'' 2^7$	F42	H24
38	$Y_n' \overline{2^1}$	P45	
40	$Y_n'' \overline{2^0}$	H46	
42	$\overline{15_5}$	F25	H44 V61
44	INPUT $Y_0 \overline{2^9}$	H connector	
46	$\overline{14_3}$	F29	H63, R59, T48
48	$Y_n' 2^1$	P50	
50	$\overline{11}$	F35	H33 V38 T29
52	$\overline{14, 14_2}$	H35	K53 I54
54	$\overline{7_2}$	H27	R7 V40, P63, X
56	$\overline{14_4}$	F64	J47
58	$Y_n'' 2^5$	B10	
60	$Y_n'' 2^7$	H25	
62	$Y_n'' 2^9$	H8	
64	$\overline{FA}$	P20	T19 Z42
66	NODE ALL"	P42	

REV STATUS

**WIRING DIAGRAM-
MATRIX
INTERCONNECTION
LIST**

WIRING LIST
BREADBOARD MATRIX ROW AA

B-14

ATWAC

CODE 86887 | SHEET

WIRING LIST

BOARD Z TO P1 & P2 OF TEST PANEL

BOARD Z	CONNECTOR	BOARD Z	CONNECTOR
1	P2-M	34	P1-A
2	P2-L	35	P1-D
3	NC	36	P1-Y
4	P2-K	37	P1-X
5	P2-N	38	P2-DD
6	P2-h	39	NC
7	P2-g	40	P2-HH
8	P2-f	41	P2-u
9	P2-e	42	P2-GG
10	P2-J	43	P2-R
11	P2-P	44	P2-c
12	P1-Z	45	P2-z
13	P1-r	46	P2-k
14	P1-q	47	P2-G
15	P1-p	48	P2-t
16	P1-n	49	P1-W
17	P1-T	50	P1-V
18	P1-F	51	P1-U
19	P1-a	52	P1-E
20	P1-s	53	P1-BB
21	P1-CC	54	P1-AA
22	P2-i	55	P1-m
23	P2-x	56	P1-S
24	P2-w	57	P1-b
25	P2-v	58	P1-t
26	P2-b	59	P1-DD
27	P2-H	60	P1-HH
28	P2-j	61	P1-GG
29	P2-y	62	P1-z
30	P2-FF	63	P2-s
31	P2-EE	64	P2-a
32	P1-C	65	NC
33	P1-B	66	P2-F

TEST PANEL WIRING LIST-1

SY'_n2^9	J1 - C
SY'_n2^8	J1 - B
SY'_n2^7	J1 - A
SY'_n2^6	J1 - D
SY'_n2^5	J1 - Y
SY'_n2^4	J1 - X
SY'_n2^3	J1 - W
SY'_n2^2	J1 - V
SY'_n2^1	J1 - U
SY'_n2^0	J1 - E
SY''_n2^9	J1 - Z
SY''_n2^8	J1 - r
SY''_n2^7	J1 - q
SY''_n2^6	J1 - p
SY''_n2^5	J1 - n
SY''_n2^4	J1 - T
SY''_n2^3	J1 - F
SY''_n2^2	J1 - a
SY''_n2^1	J1 - s
SY''_n2^0	J1 - CC
SY_o2^9	J1 - BB
SY_o2^8	J1 - AA
SY_o2^7	J1 - m
SY_o2^6	J1 - S
SY_o2^5	J1 - b
SY_o2^4	J1 - t
SY_o2^3	J1 - DD
SY_o2^2	J1 - HH
SY_o2^1	J1 - GG
SY_o2^0	J1 - z

$SC2^9$	J1 - k
$SC2^8$	J1 - G
$SC2^7$	J1 - c
$SC2^6$	J1 - u
$SC2^5$	J1 - EE
$SC2^4$	J1 - FF
$SC2^3$	J2 - F
$SC2^2$	J2 - a
$SC2^1$	J2 - s
$SC2^0$	J1 - H
B2 55	J2 - M
B2 53	J2 - L
B2 48	J2 - K
B2 47	J2 - N
B2 43	J2 - h
B2 40	J2 - g
B2 30	J2 - f
B2 32	J2 - e
B2 24	J2 - J
B2 11	J2 - P

Output Jack 2^9	Output D-A-2^9	J2 - i
Output Jack 2^8	Output D-A-2^8	J2 - x
Output Jack 2^7	Output D-A-2^7	J2 - w
Output Jack 2^6	Output D-A-2^6	J2 - v
Output Jack 2^5	Output D-A-2^5	J2 - d
Output Jack 2^4	Output D-A-2^4	J2 - H
Output Jack 2^3	Output D-A-2^3	J2 - j
Output Jack 2^2	Output D-A-2^2	J2 - y
Output Jack 2^1	Output D-A-2^1	J2 - FF
Output Jack 2^0	Output D-A-2^0	J2 - EE

TEST PANEL WIRING LIST-2

B1 - 63	S2 10 4			J2 G	S2 11 - 1
B1 - 48	S2 9 4			$SY_n 2^9$	S2 10 - 1
B1 - 60	S2 8 4			$SY_n 2^8$	S2 9 - 1
B1 - 56	S2 7 4			$SY_n 2^7$	S2 8 - 1
B1 - 35	S2 6 4			$SY_n 2^6$	S2 7 - 1
B1 - 20	S2 5 4			$SY_n 2^5$	S2 6 - 1
B1 - 59	S2 4 4			$SY_n 2^4$	S2 5 - 1
B1 - 57	S2 3 4			$SY_n 2^3$	S2 4 - 1
B1 - 36	S2 2 4			$SY_n 2^2$	S2 3 - 1
B1 - 21	S2 1 4			$SY_n 2^1$	S2 2 - 1
B1 - 64	S2 10 5	B2 - 29		$SY_n 2^0$	S2 1 - 1
B1 - 47	S2 9 5	B2 - 21		B1 - 29	S2 10 - 2
B1 - 66	S2 8 5	B2 - 20		B1 - 19	S2 9 - 2
B1 - 49	S2 7 5	B2 - 19		B1 - 4	S2 8 - 2
B1 - 31	S2 6 5	B2 - 46		B1 - 2	S2 7 - 2
B1 - 12	S2 5 5	B2 - 38		B1 - 18	S2 6 - 2
B1 - 58	S2 4 5	B2 - 37		B1 - 34	S2 5 - 2
B1 - 50	S2 3 5	B2 - 45		B1 - 53	S2 4 - 2
B1 - 32	S2 2 5	B2 - 36		B1 - 52	S2 3 - 2
B1 - 13	S2 1 5	B2 - 35		B1 - 33	S2 2 - 2
B2 - 54	INV 2^9	Input D-A-2^9	Sync Jack	B1 - 17	S2 1 - 2
B2 - 52	INV 2^8	Input D-A-2^8		B1 - 30	S2 10 - 3 , So-Com, B2 - 15
B2 - 44	INV 2^7	Input D-A-2^7		B1 - 11	S2 9 - 3 B2 - 16
B2 - 49	INV 2^6	Input D-A-2^6		B1 - 10	S2 8 - 3
B2 - 39	INV 2^5	Input D-A-2^5		B1 - 28	S2 7 - 3
B2 - 41	INV 2^4	Input D-A-2^4		B1 - 46	S2 6 - 3
B2 - 33	INV 2^3	Input D-A-2^3		B1 - 62	S2 5 - 3
B2 - 31	INV 2^2	Input D-A-2^2		B1 - 61	S2 4 - 3
B2 - 23	INV 2^1	Input D-A-2^1		B1 - 45	S2 3 - 3
B2 - 22	INV 2^0	Input D-A-2^0		B1 - 27	S2 2 - 3
				B1 - 9	S2 1 - 3

TEST PANEL WIRING LIST-3

B2 - 4,	Y'_n 2^9 - 2^5 switches normally open
B2 - 5,	Y'_n 2^4 - 2^0 switches normally open
B2 - 7,	C 2^9 - 2^0 switches normally open
B2 - 12,	Y_o 2^9 - 2^5 switches normally open
B2 - 10,	Y_o 2^4 - 2^0 switches normally open
B2 - 63,	Y''_n 2^9 - 2^5 switches normally open
B2 - 62,	Y''_n 2^4 - 2^0 switches normally open
BF - COM,	J2 - DD
LP - COM,	J2 - HH
FA - COM,	J2 - GG
FVA-ZVA N.O.,	J2 - Z
FVA-ZVA N.C.,	J2 - k
B2 - 27	B1 - 1
B2 - 26	B2 - 50
B2 - 6	S2 - 11 - COM - J2 - c
B2 - 8	S2 - 12 - 1
B2 - 3, INV (+5), B1 - 3,	DA1 (+5), DA2 (+5), +5 Jack
B2 14	J2 - t 5.1KBUSS Y_n Swts
S2 - 12 - COM	J2 - μ
B2 - 64	B1 - 6
B2 - 25	S1 - COM
B2 - 18	S1 - N.O.
B2 - 17	S_o - N.O.
B2 - 34	S2 - 1 - 3
B2 - 51	S3 - 1 - 2
B2 - 50	S3 - 1 - 1 J2 - R
B2 - 9	S3 - 1 - COM
B2 - 1	B1 - 54
B2 - 13	B1 - 55
B2 - 66	B1 - 5

TEST PANEL WIRING LIST-4

Output D-A		Output Jack
Input D-A		Input Jack
GND Jack,	B1 - 65,	B2 - 65, O-D-A, I-D-A, INV (GND), S2-12-2 THRU 5, BF-N.C., LP-N.C., FA-N.O., FVA-ZVA-COM