

UNCLASSIFIED

AD 273 741

*Reproduced
by the*

**ARMED SERVICES TECHNICAL INFORMATION AGENCY
ARLINGTON HALL STATION
ARLINGTON 12, VIRGINIA**



UNCLASSIFIED

NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

273 741

273741

AS AD No.

RADC-TR-61-159

LINEAR RECURRENT BINARY ERROR-BURST-CODER

Richard W. Weeks

Electronics Research Laboratory
Endowment & Research Foundation
Montana State College
Bozeman, Montana

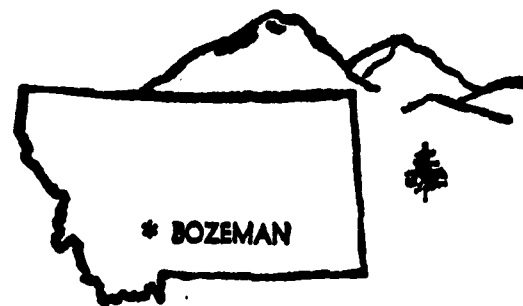
FINAL REPORT

Contract No. AF 30(602)-2315

October 23, 1961

Prepared for

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
GRIFFISS AIR FORCE BASE, NEW YORK
ATTN.: RAUAT/MR. MILES BICKELHAUPT



**electronics
research
laboratory**

ENDOWMENT AND RESEARCH FOUNDATION AT
MONTANA STATE COLLEGE, BOZEMAN, MONT.

RADC-TR- 1-199

**LINEAR RECURRENT
BINARY ERROR-BURST-CODER**

Richard W. Weeks
Wm. L. Kilmer
Daniel N. March
Charles Sykes

Electronics Research Laboratory
Endowment & Research Foundation
Montana State College
Bozeman, Montana

FINAL REPORT
Contract #AF 30(602)-2315

October 23, 1961

Prepared for
ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
GRIFFISS AIR FORCE BASE, NEW YORK
ATTN: RAUAT/MR. MILES BICKELHAUPT

TABLE OF CONTENTS

	<u>Page</u>
Abstract	
Introduction.....	1
I. The Coding System.....	2
The Generalized Coder.....	2
The Generalized Decoder.....	3
II. Instrumentation of the Code.....	5
III. Operation of Equipment.....	7
A. Detail Logic of the Encoder.....	7
B. Detail Logic of the Decoder.....	13
IV. Testing Procedure.....	16
V. Test Results.....	18
VI. Interpretation of Test Data.....	30
VII. Equipment Characteristics.....	32
VIII. Operation of Equipment.....	33

ABSTRACT

This report describes the construction and testing of a binary error-burst-correcting channel simulator and decoder. The reader is referred to Report number 4 of the RADC TN-61-79 and Report number 3 of RADC TN-60-49 under contract no. AF30(602)-1915 for a detailed description of the general code of which this equipment tests the special case described below. It should be emphasized that this report contains, for the most part, a description of the equipment and the results of some initial experimentation.

The code this equipment instruments has every 4th digit a parity check digit, and can correct almost all error bursts that span 9 to 12 consecutive coded message digits, depending upon the phase of the burst. A guard space between bursts of at least 72 consecutive non-corrupted coded message digits is required, where the first counted of these is a check digit. Perfect decoding is achieved if the guard space contains at least 132 consecutive non-corrupted coded message digits.

The equipment was constructed from standard binary logic units. The channel simulator (herein after called the encoder) is not general owing to the linearity of the code and the chosen message sequence of all ZEROs. It merely has the ability to generate random errors and parity check digits, open the channel gate during a burst and close it during a guard space, and preserve all phase relations. The decoder has the ability to compare information digits with parity digits, establish a correction sequence and locate its largest axis, and correct the digits which are in error.

The testing procedure uses two electronic counters in conjunction with encoder-decoder. One is used to count errors entering the decoder; the other to count those leaving the decoder. Thus, the error correcting efficiency is easily calculated.

Test results show perfect correction with a guard space of 132 digits. With guard lengths below 132, better correction efficiency is achieved if the probability of an error during a burst is high. An analytical interpretation of test data is given in the text of the report.

INTRODUCTION

In the second report under Rome Contract No. AF30(602)-1915 entitled "Some Results on Linear-Recurrent Binary Burst-Correcting Codes", an error correcting code is given which is suited for digital communication systems that accept random binary message sequences, encode them into binary coded message sequences, and transmit them over channels which corrupt the coded message in bursts. It is assumed that during each burst, the probability of any particular digit being corrupted is p ($0 < p < 1$) and is independent of any other digits being corrupted. It is also assumed that there is at least a certain required guard space, or number of consecutive non-corrupted coded message digits between every pair of successive bursts.

The report gives an example of a burst-correcting code system which allows for every fourth binary digit in the coded message sequence to be a parity check digit. These parity check digits are generated from a combination of widely separated information digits in such a way that if errors occur in short bursts, with sufficiently long guard spaces between these bursts there is enough information correctly transmitted to detect and correct the errors.

It was conjectured that the particular code system explained was at least an "almost best" linear recurrent code with respect to its ratio of maximum correctable burst-length to required-minimum-guard length. However, the quantitative significance of "almost" was not known. Since the situation was quite complex, it was impossible to give an accurate analytical evaluation of "almost" in terms of strict probability considerations. Hence, an empirical test was needed to make a numerical evaluation of this term.

The purpose of this project therefore, was to construct a channel simulator (encoder) and decoder to instrument the type of code mentioned. A random noise generator could be used to provide the system with random errors. Electronic counters could be used to count errors entering

and leaving the decoder. Then, tests could be run to determine the decoder correction efficiency as a function of minimum required guard space with the maximum allowable burst length held constant. The guard space would be decreased in steps from a value which gave perfect correction to the conjectured minimum number of coded message digits for the code. Thus, an empirical evaluation of the "almost" term discussed alone would be given in terms of correction efficiency. This evaluation would be made for several values of error probability ranging from zero to one. Thus, a wide range of statistical data could be utilized to evaluate important characteristics of the code.

I. THE CODING SYSTEM

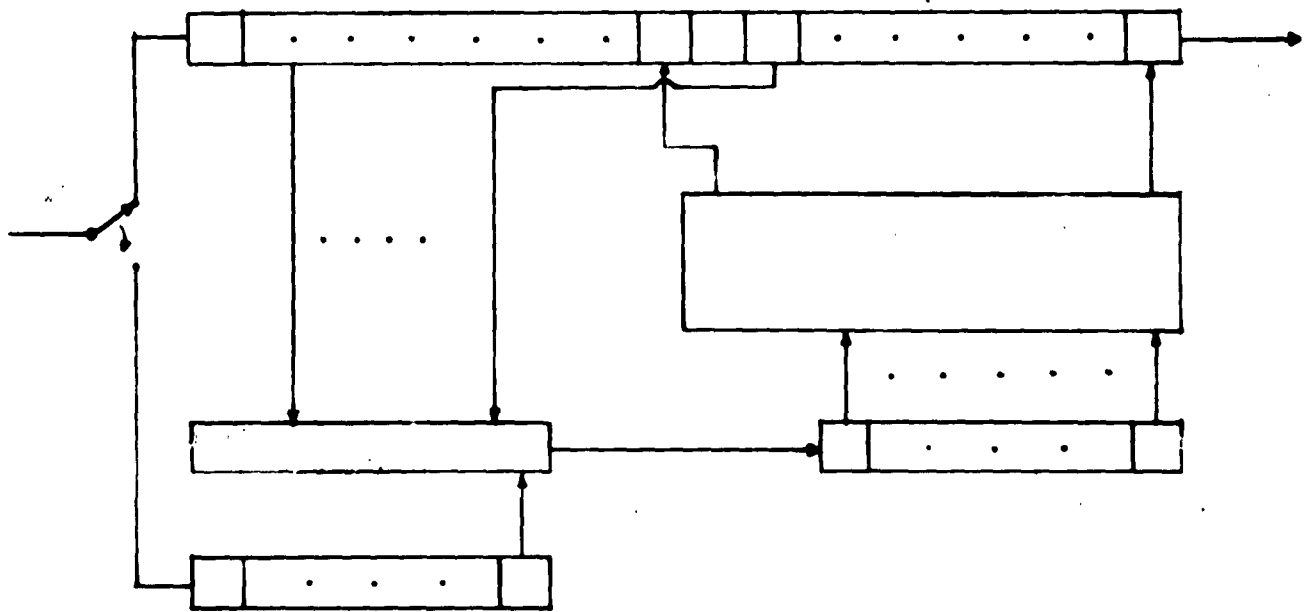
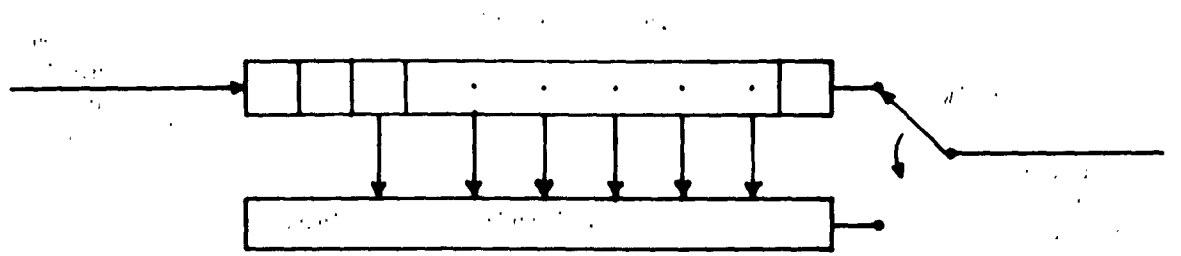
The code to be realized allows for every 4th digit in the coded message sequence to be a parity check digit, which corrects almost all error bursts that span a length of < 9 to 12 consecutive coded message digits. (9 to 12 depending upon the "phase" of the burst, i.e., whether the first corrupted digit of the burst is a check digit, the digit immediately following a check digit, etc.) The code requires a guard space of at least 72 consecutive non-corrupted coded message digits between bursts (where the first counted of them must be a check digit). Perfect correction is guaranteed if successive bursts are separated by a guard space of at least 132 error-free digits.

The Generalized Coder

In Figure (1a) a random sequence of binary message digits is applied to the input of a 57 place shift register, R, in blocks of 3 digits per block. After each successive block is situated in R, the parity check circuit, P, forms a binary parity check digit of such a value that the modulo 2 sum of it and the digits in the 1, 11, 19, 28, 38 and 48 cells of R is zero. Then, the check digit is transmitted. Thus, the output of the coder consists of blocks of three message digits from R followed by single parity check digits from P.

The Generalized Decoder.

In Figure (1b) the "received coded message sequence" is assumed to be the coded message sequence out of Figure 1a) altered by channel noise. The function of the decoder is to first check the parity relations that should hold in the received message, and hereby generate a parity check sequence, S. Then, when the first ONE of an unsatisfied parity relation, corresponding to an error in the coded message sequence, arrives in the 18th cell of R_4 a pattern in this subsequence is recognized and correlated with an error configuration in the information digits of R_3 . The corrupted digits in the cells of R_3 are corrected allowing the correct message sequence to be shifted out of R_3 .



II. INSTRUMENTATION OF THE CODE

It was proposed that an encoder-decoder be constructed to test error correction efficiency as a function of guard space and burst error probability for the code system explained above. The guard space would be varied in small increments from the lower bound of 72 to the perfect correction guard space of 132 digits. Also, the burst error probability would be adjusted in increments from zero to one. Since it is not important, due to the linearity of the code, that any specific message be coded, it was decided that a code consisting of all ZEROS would be used. In this case, any ONE present at the input of the decoder would be considered a corrupted digit and any ONE present at the output of the decoder would be considered an uncorrected error. By this method, the correction efficiency of the decoder could be tested without having to build a complete encoder capable of instrumenting any binary sequence.

Strictly speaking, an error probability greater than one half has no significance since if the error probability is $p \geq 1/2$, then, it is more appropriate to assume that the opposite binary digit occurred but with probability of error being $p^1 = (1-p)$. However, since the actual test message used in this work was one consisting of all ZEROS, the conversion of any digit to a ONE corresponded to an error. The probability of converting a ZERO to a ONE was made adjustable from zero to one and for this reason the point of view of an error probability from zero to one will be maintained throughout this report.

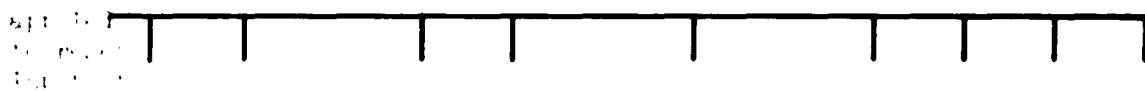
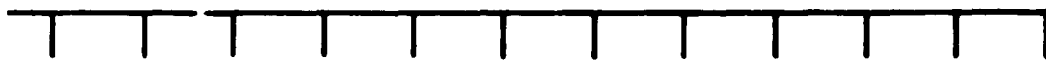
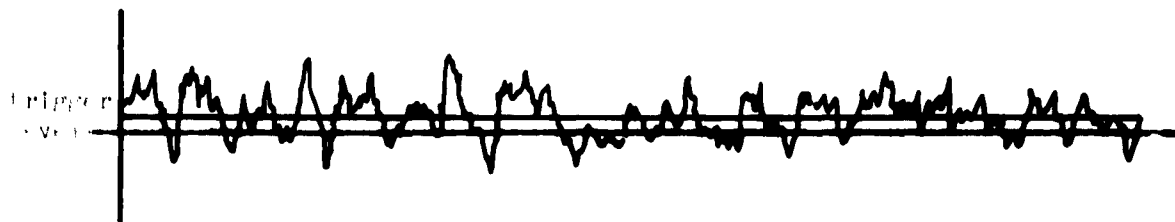


Figure 1

Timing diagram for a digital system

III. OPERATION OF EQUIPMENT

Both the encoder and decoder are operated as a synchronous system from an external or internal system clock. The equipment will operate reliably at clock repetition rates up to 20 kilocycles per second.

In the encoder the system clock is divided by four by the burst phase counter so that every fourth timing pulse is a parity check digit clock pulse. The other three pulses represent the clock pulses for the first, second, and third information digits. (See Fig. 3)

A. Detail Logic of the Encoder

Refer to drawing 102-2. The output of a voltage noise source is applied to the input of a Schmitt trigger which generates a random asynchronous binary sequence, (See Fig. 2). The probability, p , of a ONE being present at any time can be varied from $0 < p < 1$ by adjusting the output level of the noise source. The Schmitt trigger output is fed into a synchronizing "and" gate A_1 admitting only pulses which occur in time phase with the system clock. This output is applied to the set input of flip flop F_1 . The Schmitt trigger output is also inverted and gated with the system clock A_4 and applied to the reset input of flip flop F_1 . At the output of flip flop F_1 a synchronous binary level sequence exists. This sequence will be completely random, assuming no periodic output from the voltage noise source, if the sampling rate is appreciably less than $\frac{1}{2B}$ per second (B = bandwidth of noise). As explained in the section on "instrumentation of the code", the correct sequence is to consist entirely of ZEROS so that the presence of a ONE at the output of F_1 represents an error.

To insure that the system is operating within the prescribed limits dictated by the code, it is necessary that an error burst contain no more than nine to twelve digits (twelve if the first error is a parity digit, eleven if the first error is a first information digit, etc.). Also, each error burst must be followed by a selected

guard space. Thus, the function of the channel gate A_6 is to open for an error burst and then close for a prescribed guard length. Control for this gate is received from the level of flip flop F_4 .

Consider circuit operation (DWG 102-2) starting just prior to the last guard space count. The burst length counter has been counting to 4 and has been reset by each delayed third information digit. When the guard space counter finishes its count, the output resets flip flop F_4 which in turn opens channel gate A_6 and coincidence gate A_3 . The output of the guard space counter also serves to reset all of its cells so that a new count may begin on the next timing pulse.

The timing diagram of Fig. 3 shows a block of pulses containing 12 digits. The parity digit occurs at time t_1 and the first second and third information digits occur at t_2 , t_3 , and t_4 respectively.

When the first ONE, corresponding to an error, occurs at the output of F_1 , flip flop F_3 is set and gate A_7 is closed. This prevents the burst length counter from being reset. This first ONE may occur at time t_1 in which case the burst length counter would just have been reset and would count all 12 digits t_1 ----- t_{12} before giving an output. If the first ONE occurred at $t = t_2$ the burst count could only be 11 digits long. If the first ONE occurred at $t = t_3$ the burst could only be 10 digits long. Finally, if the first error occurred at $t = t_4$, the burst could only last for 9 digits.

When flip flop F_3 is set, "AND" gate A_4 is opened in order that the first ONE may reset flip flop F_2 . The output from F_2 then conditions coincidence gate A_5 so that the guard space counter will be reset on the third information digit, I_3 , delayed clock pulse. Each delayed I_3 clock pulse used for resetting the guard space counter is further delayed through D_1 and then sets F_2 provided a ONE is not present at the reset input of F_2 . If it is, the flip-flop will not be set. In essence, this says that the guard space counter reset is enabled as long as an error is present between reset pulses (i.e., the delayed I_3 clock pulses) and the total number of pulses since the

first ONE has not exceeded the 0 to 12 maximum. If, for example, there are no more errors present after the first ONE, the guard space count would begin its count on the next parity check clock pulse.

As soon as the burst length counter has finished its count its output sets flip-flops F_4 and F_3 . Delay D_3 and D_7 are merely present to eliminate race conditions. The state change of F_4 disables channel gate A_6 and gate A_3 . This prevents any errors from passing through to the decoder during the guard space and disables the set input of F_3 . The resetting of F_3 (from the burst length counter output) enables coincidence gate A_7 while disabling gate A_2 . The disabling of gate A_2 serves to prevent flip-flop F_2 from being set which, in turn, closes "AND" gate A_5 and prevents the guard space counter from resetting. At the same time, when "AND" gate A_7 is enabled, the reset to the burst length counter is allowed to function.

"AND" gate A_2 also has the clock delayed as one of its inputs. This is merely to change the output of A_2 from level to pulse logic so that each error present in the Y sequence is capable of resetting flip-flop F_2 . Delay D_4 is present to cover switching times required by F_4 , A_3 , and F_3 . Similarly AND gate A_6 has the clock delayed as one of its inputs. This is also present to change from level to pulse logic thereby providing pulses for the decoder input at R_1 and R_3 . Delays D_9 and D_{10} at the reset busses of the burst length and guard space counters serve to widen the pulse and furnish drive for the resets.

The function of the burst phase counter is to divide the clock pulses by 4, thereby giving an output corresponding to a parity digit clock pulse. Outputs from the two flip-flops of the burst phase counter are applied as inputs to OR gate O_2 . O_2 gives an output level which is present except during the time between I_3 and the following parity check clock pulse. (See Fig. 4)

This output and its negation are used to control AND gates A_8 and A_9 . This action steers the three information digits from the control gate A_6 (its output is in phase with the delayed clock) into information register R_1 and the parity check digits into Parity Check register R_2 of the decoder.

The output from OR gate O_2 along with its negation control AND gates A_{10} and A_{11} . The output from A_{10} is three delayed information digit clock pulses while the output from A_{11} is the delayed parity check clock pulse. These outputs are further delayed by delays D_5 and D_2 . The output from D_5 provides the advance for the information shift register R_1 . Likewise, the output from D_2 provides the advance for the parity check shift register R_2 . (Refer to Fig. 4) A manual reset is provided so that the cells of R_1 and R_2 may be reset.

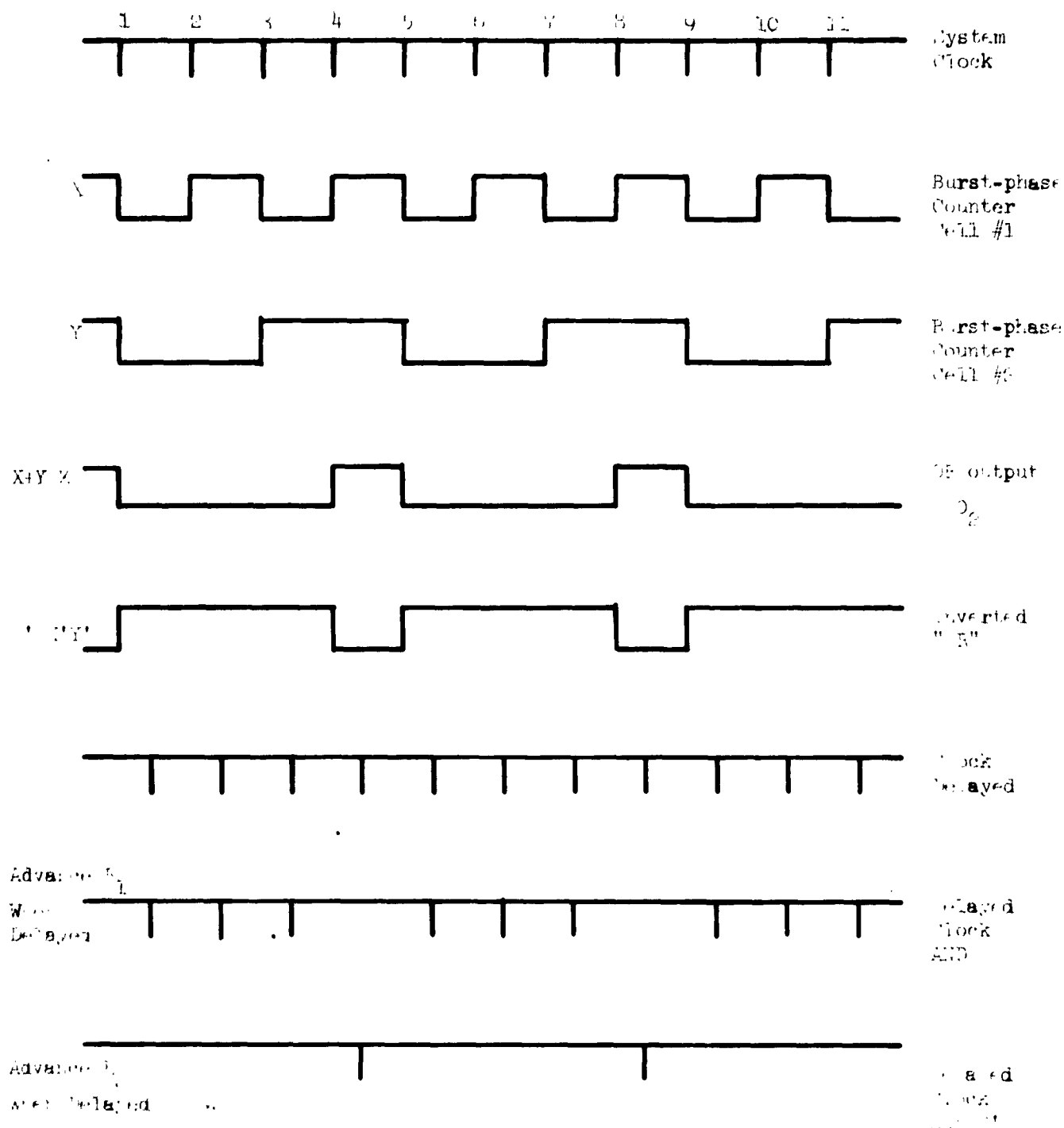


Figure 4
Timing Diagram for Burst-Phase Counter

B. Detail Logic of Decoder

As digits arrive at the decoder, the information digits are switched to the shift register R_1 while the parity checks are inserted in the R_2 shift register. After each third information digit enters R_1 , the parity relations that were checked by the original parity checks are again compared. This is accomplished by adding, modulo two, the 3, 11, 19, 28, 38, and 48 places of the shift register R_1 . The modulo two adder uses comparators which produce a ONE output if one and only one of the input stages from the R_1 register is in the ONE stage. In other words, the comparators are used as "EXCLUSIVE OR" gates. As the comparators are in cascade, double inverters are inserted between them to not only amplify the signals but also give the output as well as its negation.

The newly formed parity relation is now compared to the original received parity digit. As this comparison is a modulo two addition, this parity relation is simply included in the modulo two sum as another element.

If the parity relation is not satisfied, a ONE will appear as the output of the mod-two adder when compared with the original parity digit. This ONE will be shifted into the S sequence shift register R_4 providing there is not a blocking pulse from the axis erase circuitry. This ONE is shifted by actuating a six micro second delay with the advance pulse that advances R_2 and R_4 . If all the above conditions are met, the ONE will be inserted into R_4 six microseconds after the digits in R_4 have been advanced one cell (place).

As the S sequence shift register has no significance until the last place, number eighteen has a ONE in it, nothing is actuated until such time as this occurs. Comparators are used for the axis of symmetry location as the digits on both sides of all axes must be compared. The comparators used on the eighteenth place will not yield a ONE output until both the eighteenth place of R_4 and the other compared place are ONE. This results in a common AND gate. The other

comparators in the array are such that a ONE is put out if the two shift register cells have either both ONES or both ZEROS (i.e., using conventional Boolean algebra notation, where A means a ONE in cell A and a A^1 means a ZERO in cell A) the output is $AB + A^1B^1$ for any cells A and B. Due to the load not always being constant on the comparators, various levels of output were found and were eliminated by changing the inputs to $AB^1 + A^1B$ followed by inverter amplifiers as isolation and amplification. (Note that the inverse of $AB^1 + A^1B$ is $AB + A^1B^1$)

To have a particular axis of symmetry all the digits that are compared must produce ONE outputs from the comparators. Therefore, the necessary comparator outputs are put through multiple AND gates (i.e., for axis number 5, comparators M, N, P, R, and S must be ONE). Each multiple AND gate in this section is followed by two inverters. The two-fold purpose of these inverters is: 1) to isolate and amplify the pulses and, 2) to give the inverted output of the particular axis.

There most likely will be more than one axis that has all the necessary comparators in the ONE state. Of these, the largest one is the true or desired axis, and circuitry is included to eliminate any smaller axes. This circuitry is initiated by a pulse, either from R_h delayed, or a delayed pulse from a pulse generator or the clock. The eliminating circuitry is designed such that no "race" conditions exist and the logic involved can be followed through easily. Each axis, excepting number 7, has an emitter follower for isolation after the axes eliminating circuitry.

Coming from the axis elimination circuitry is a single axis in the ONE state. This axis proceeds and is combined, using AND gates, with the digits to the right of the axis in R_h . This is done as only the ONES to the right of the axis show where digits are to be complemented. The ZEROS which may exist to the right of the axis show that the corresponding digit in the information is correct. The total logic of this section could have been done with less circuitry had OR gates been used, but the loss of signal and multiple level voltages made the

more elaborate AND gate arrangement more practical. As an example, if the major axis is at #2, and if the fifteenth position in R_4 is a ONE, then the information digit in the 53rd place of R_3 must be complemented. However, it is seen that this 53rd digit can also be complemented due to an axis at either #1 or #3 and the fifteenth position of R_4 in the ONE state. Therefore, diode isolation for each axis is necessary at the complementing input of the information digits. Also, note that no AND gate action is necessary for the axis of symmetry and the eighteenth position of R_4 this position must always be ONE before an axis is found. Hence, the axes also go directly to the diode isolations for particular digits in R_3 .

The S sequence digits, after an axis has been found and correction has taken place, must be erased as no other true axes are to be found among them. The number of digits to be erased depends on the size of the axis. With an axis at #1, #2, or #3, all eighteen S sequence digits plus the next three digits contain information of no further use. The next three digits are possibly corrupted due to the burst that was just corrected. Hence, a lead is brought from the isolation of information digit #54 which contains either of all three axis concerned; continued through a pulse delay and widener; and proceeds to erase the eighteen digits in R_4 plus actuating the three stages in the erase storage shift register. When these three shift register places are actuated, they close the AND gate A_{21} for three S sequence shifts. For an axis at #4, #5, or #6, the total S sequence register must be erased but not the next three digits. With an axis at #7, only the last fifteen digits of the S sequence need to be erased.

The decoder, after erasing, is now ready to receive another burst of errors. Any digits that were not corrected or were erroneously complemented will be shifted on in the information register and finally out at the 70th place. If a counter is connected to the advance output on this stage, the digits in error will be counted. Occasionally, the 70th place will have an error in it and this error will be corrected there. This will not produce a pulse on the advance output that will be counted.

IV. TESTING PROCEDURE

After the encoder and decoder were constructed, a test setup was arrived at. The following questions were deemed important.

1. Test for maximum repetition rate of equipment for reliable operation.
2. Vary guard length in steps from lower bound of 72 to upper of 132. For each step obtain efficiency readings for error probabilities ranging from zero to one.

A testing arrangement was arrived at as shown in Figure 5.

The function of the noise source was to provide a completely random signal for sampling by the internal schmitt trigger. The output level was variable for altering error probability. The external system clock was set to operate at 10 kilocycles per second and the frequency of the output of the synchronous AND gate, A_1 , was measured. The error probability for this condition was then given as:

$$p = \frac{\text{counts from } A_1 \text{ during 10 second intervals}}{\text{ten times the clock frequency}}$$

Ten readings of p were taken at each setting and a statistical average was taken. It was found that deviation from the average was no more than one percent.

For each probability the system was allowed to run until approximately twenty thousand errors were totalized on counter A. The number of uncorrected errors were totalized on counter B.

The correction efficiency for each probability and guard space was then determined by:

$$\%EFF = (1 - \frac{\text{uncorrected errors}}{\text{total errors}}) \times 100$$

At each probability setting efficiencies were determined at guard space values of 72, 75, 78, 84, 90, 96, 100, 108, 120, and 132 digits.

Sets of data were taken for probability values of 0.06, 0.125, 0.30, 0.038, 0.434, 0.4, 0.585, 0.68, 0.76, 0.875, and 0.12. Graphs showing these results are given in the next section.

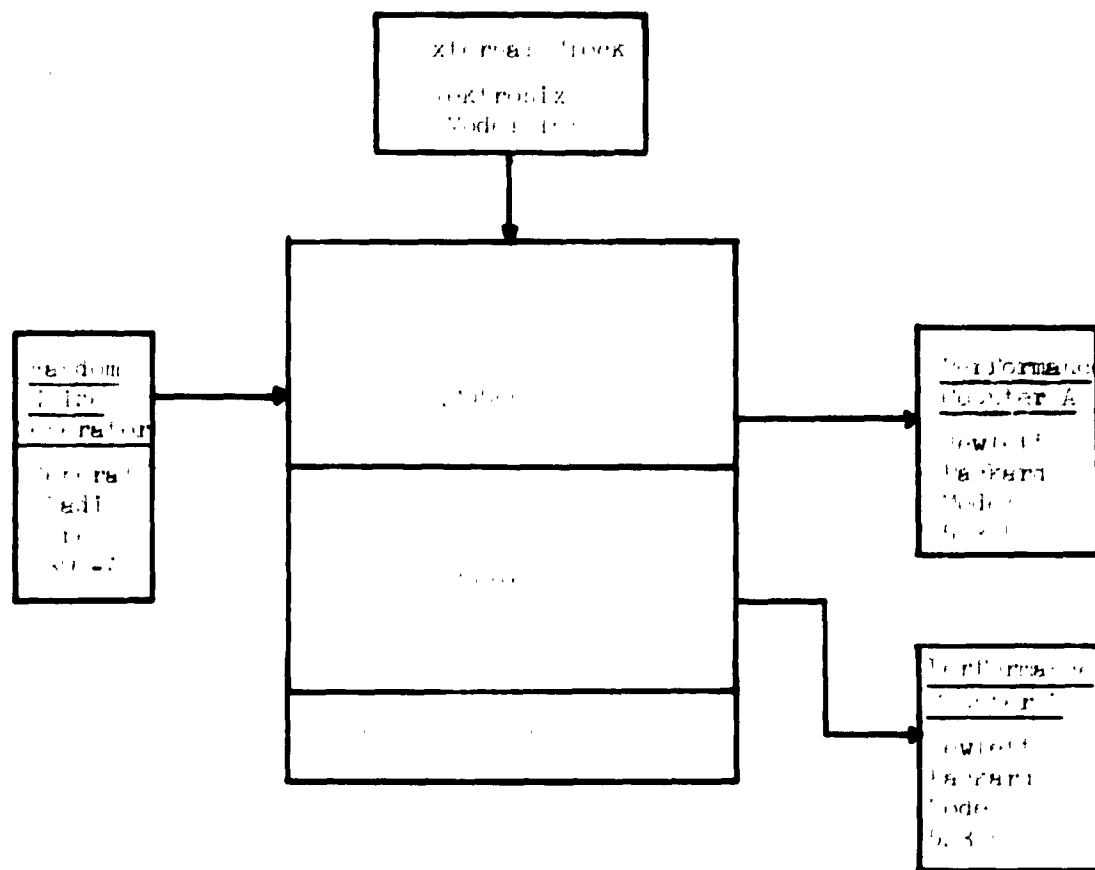
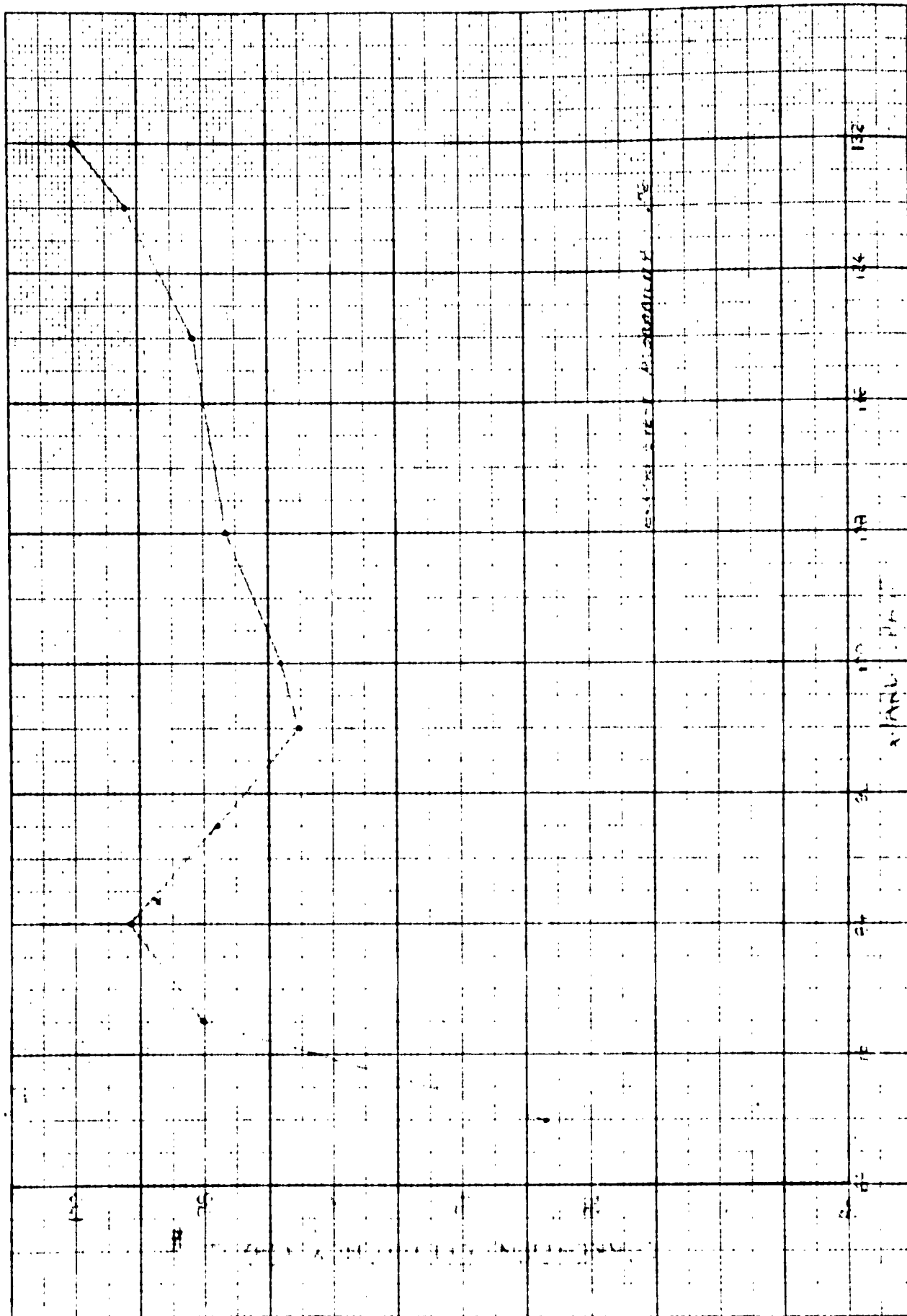


Figure 1

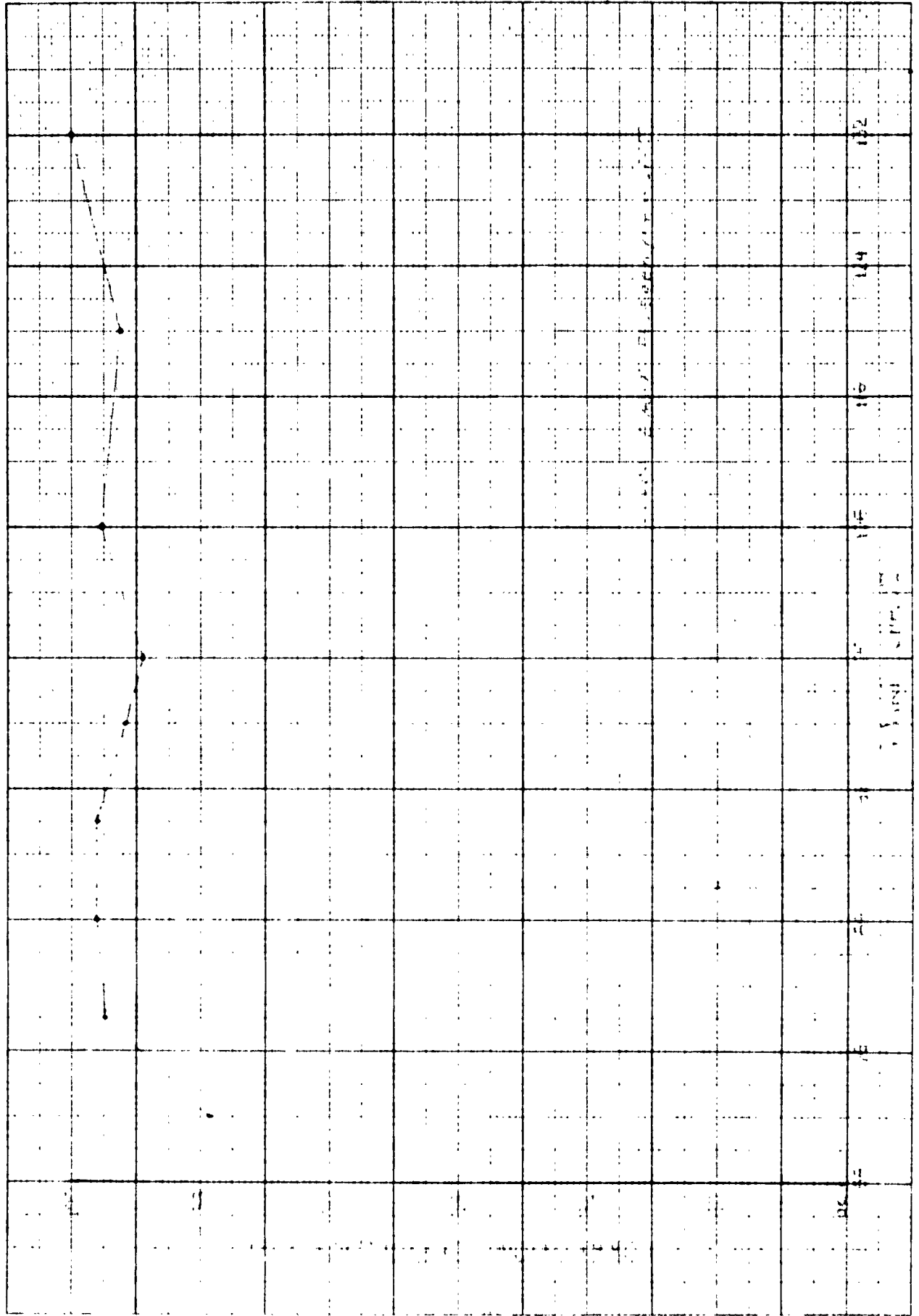
EUGENE DIETZGEN CO.
MADE IN U.S.A.

NO 340 20 DIETZGEN GRAPH PAPER
20x20 PER INCH



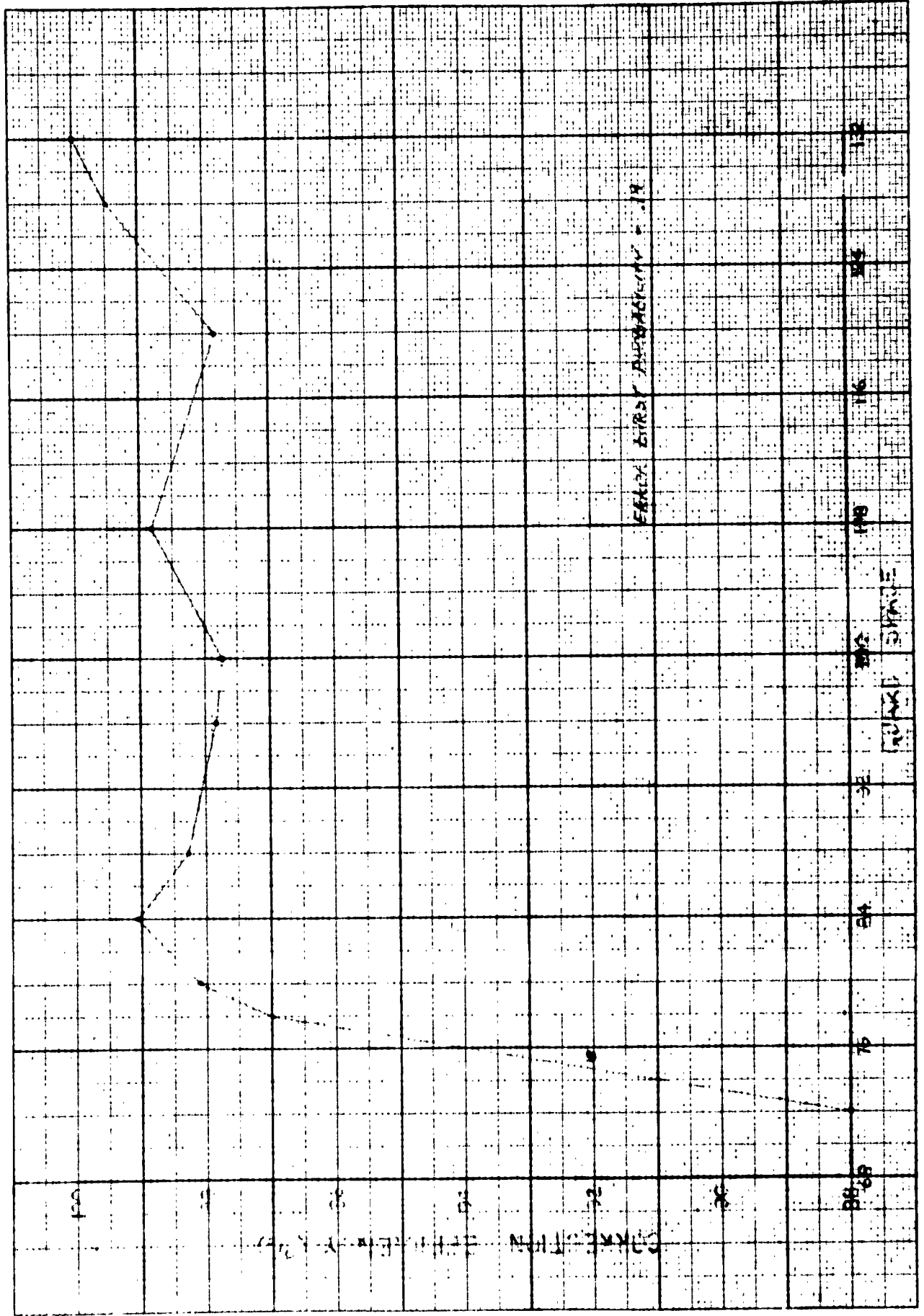
N^o 340 20 DIETZGEN GRAPH PAPER
20 X 20 PER INCH

EUGENE DIETZGEN CO
MADE IN U. S. A.



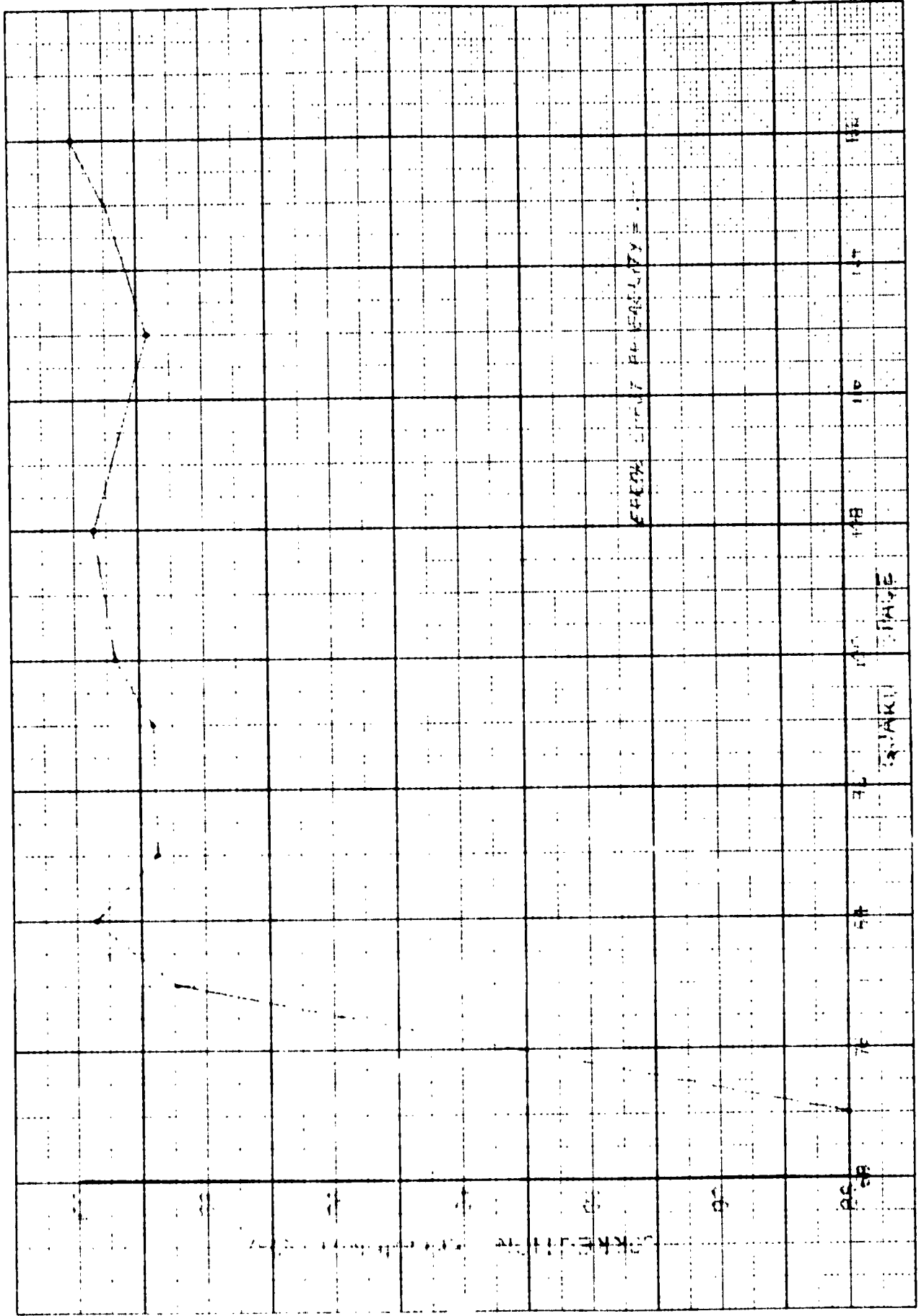
EUGENE DIETZGEN CO.
MADE IN U. S. A.

N. 340 20 DIETZGEN GRAPH PAPER
20 X 20 PER INCH



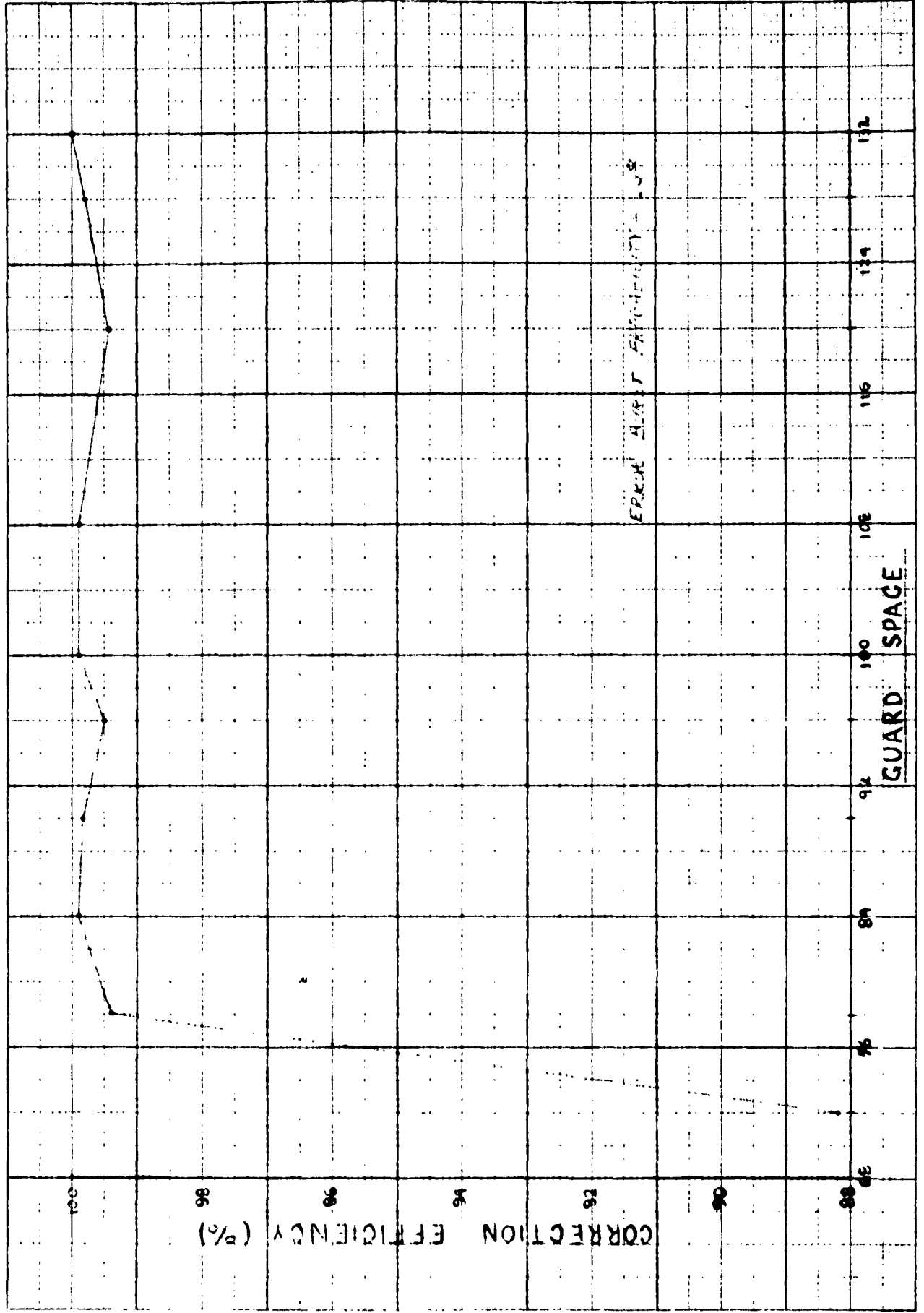
NO 340 40 DIETZEN GRAPH PAPER
20 X 20 PER INCH

EUGENE DIETZEN CO
MADE IN U.S.A



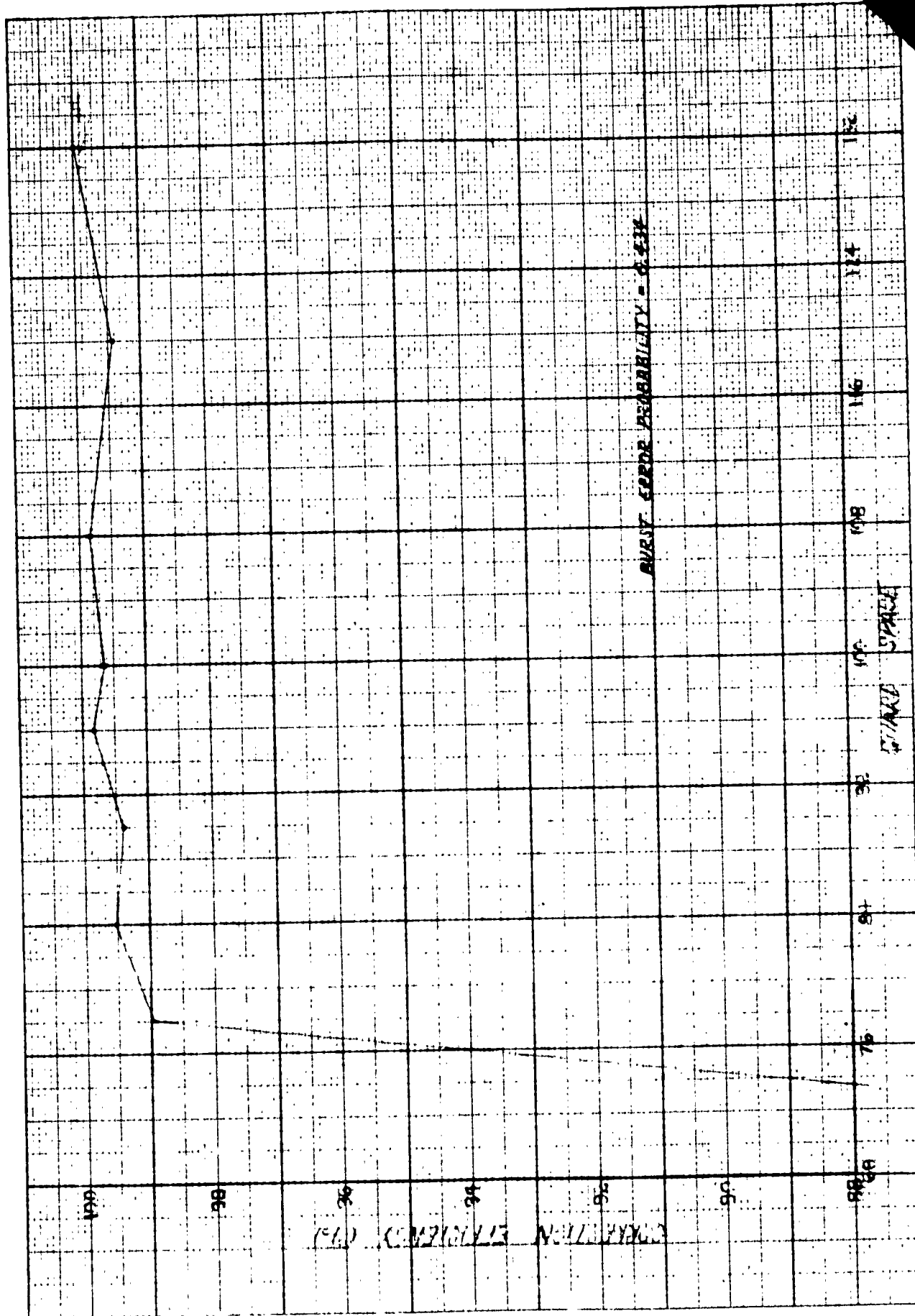
NO 340 20 DIETZGEN GRAPH PAPER
20 X 20 PER INCH

EUGENE DIETZGEN CO
MADE IN U. S. A.



EUGENE DIETZEN CO.
MADE IN U. S. A.

NO 340 20 DIETZEN GRAPH PAPER
20 X 20 PER INCH



NO 34C 20 DIETZGEN GRAPH PAPER
20 X 20 PER INCH

EUGENE DIETZGEN CO.
MADE IN U. S. A.

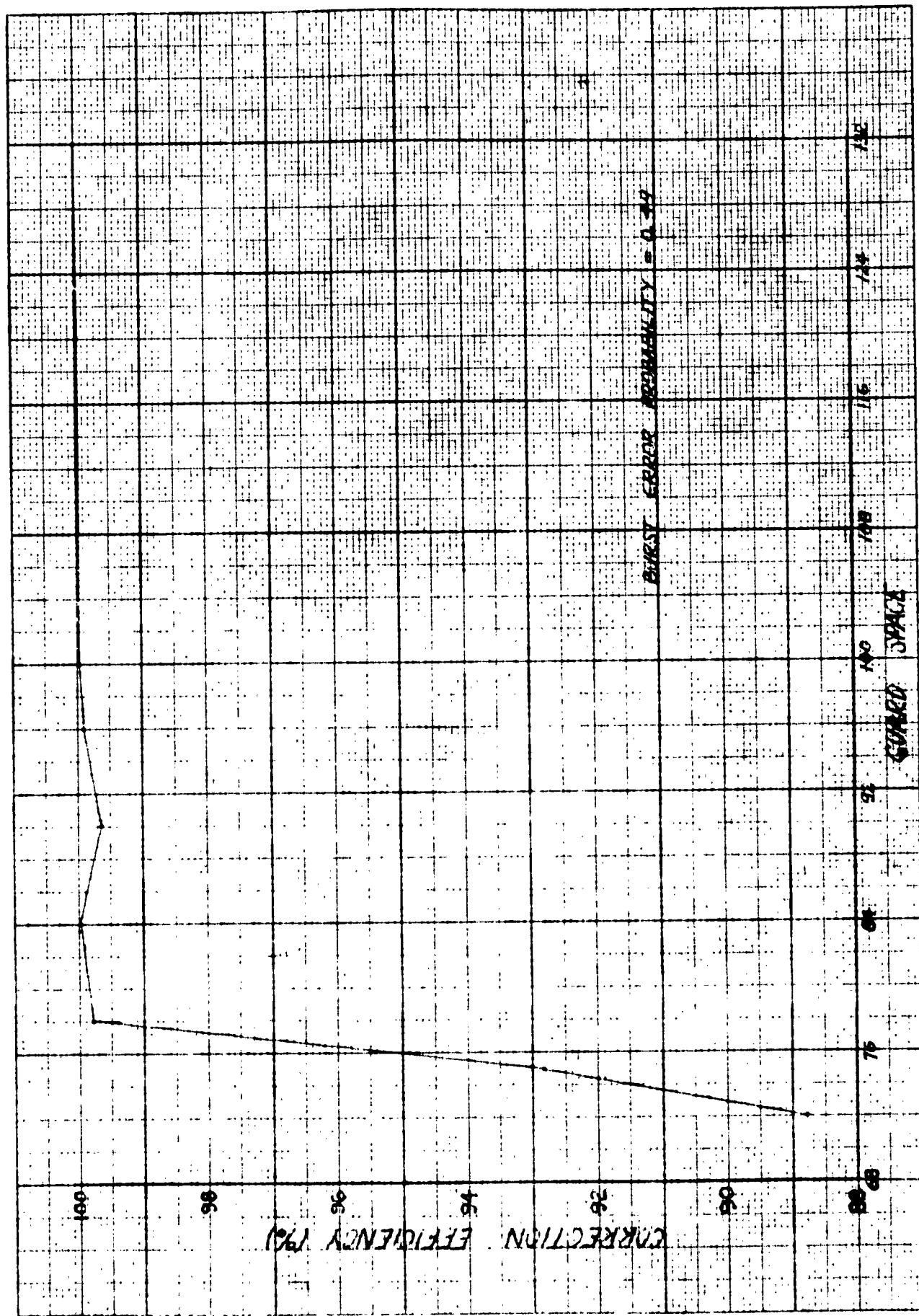
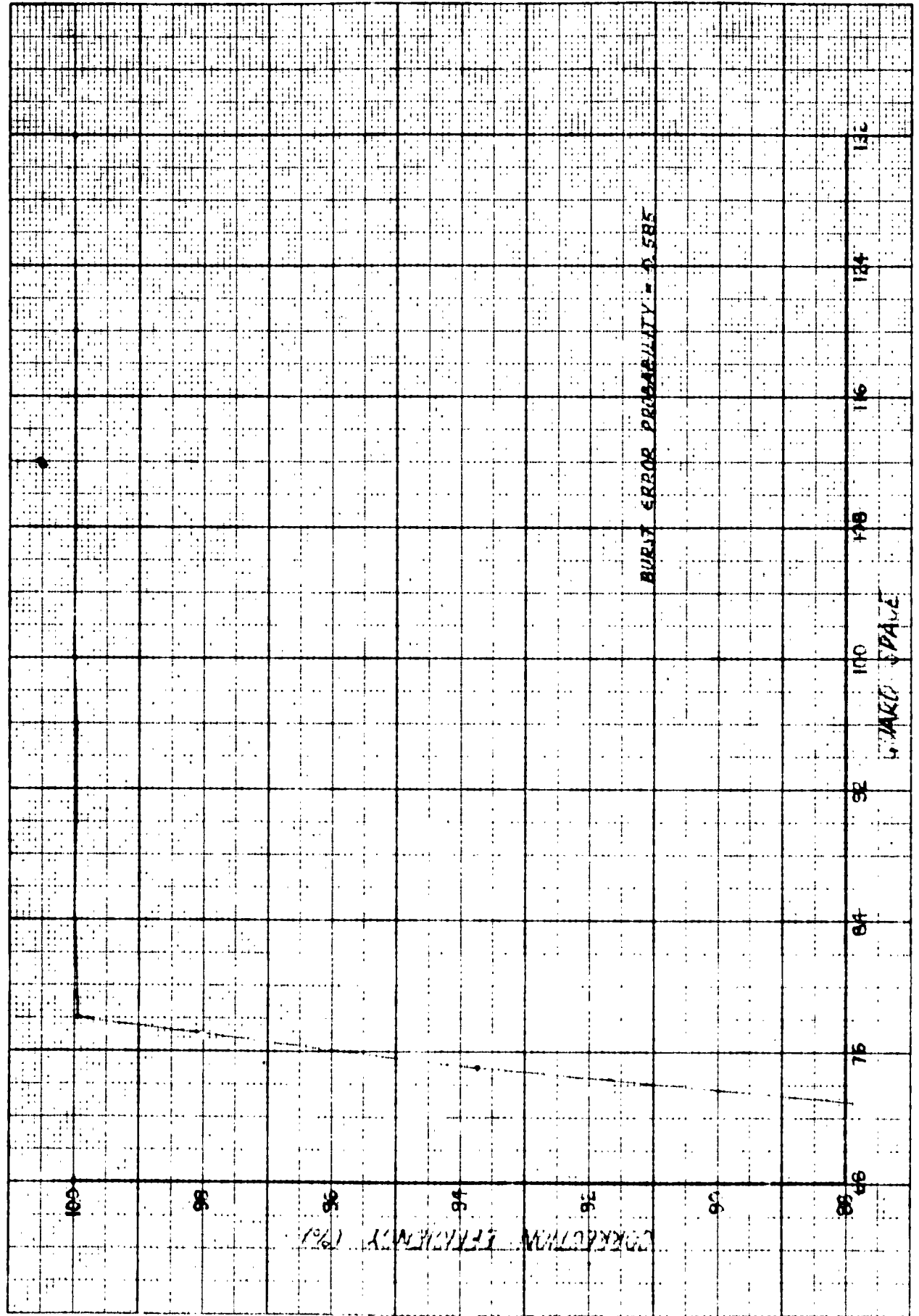


Fig. 6-7

NO 340 -20 DIETZEN GRAPH PAPER
20X30 PER INCH

EUGENE DIETZEN CO.
MADE IN U. S. A.



NO 34L -20 DIETZGEN GRAFT PAPER
20 X 20 PPR INCH

EUGENE DIETZGEN CO.
MADE IN U. S. A.

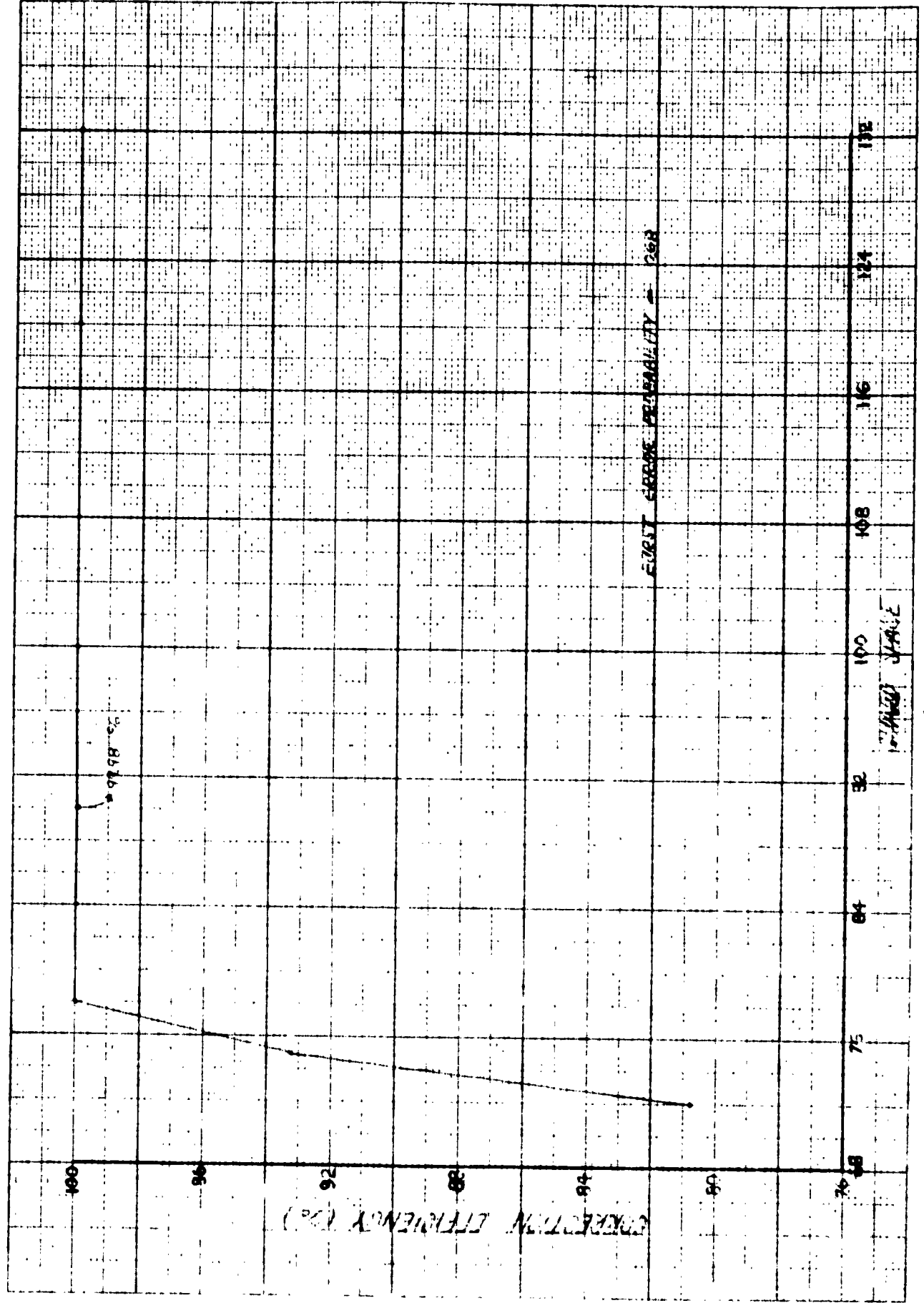


Fig. C-9

EUGENE DIETZGEN CO.
MADE IN U. S. A.

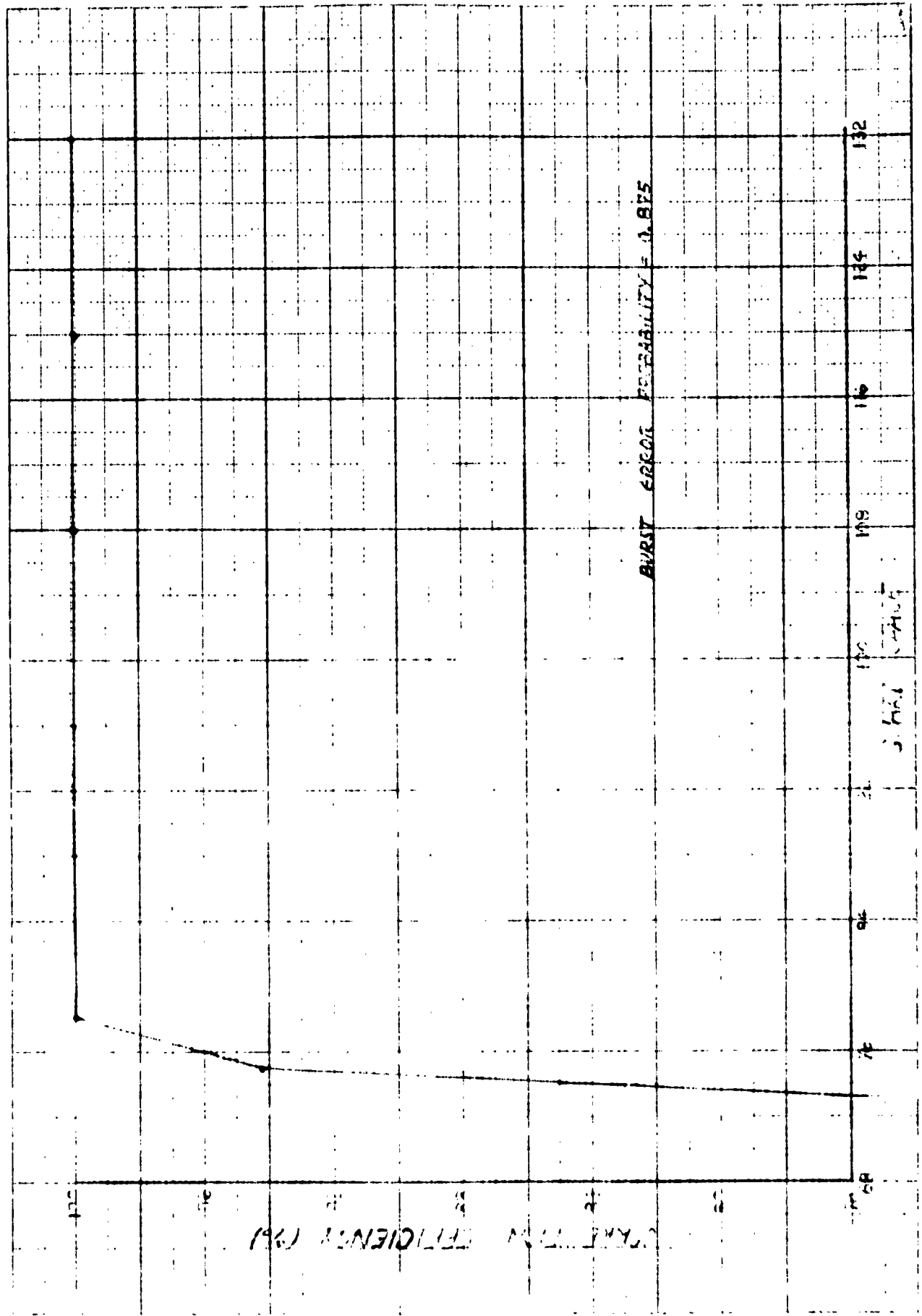
NO. 340 20 DIETZGEN GRAPH PAPER
20 X 20 PER INCH



Fig. G-10.

EUGENE DIETZGEN CO.
MADE IN U. S. A.

20 X 20 PER INCH



EUGENE DIETZEN CO.
MADE IN U. S. A.

N° 340 20 DIETZEN GRAPH PAPER
20 X 20 PER INCH

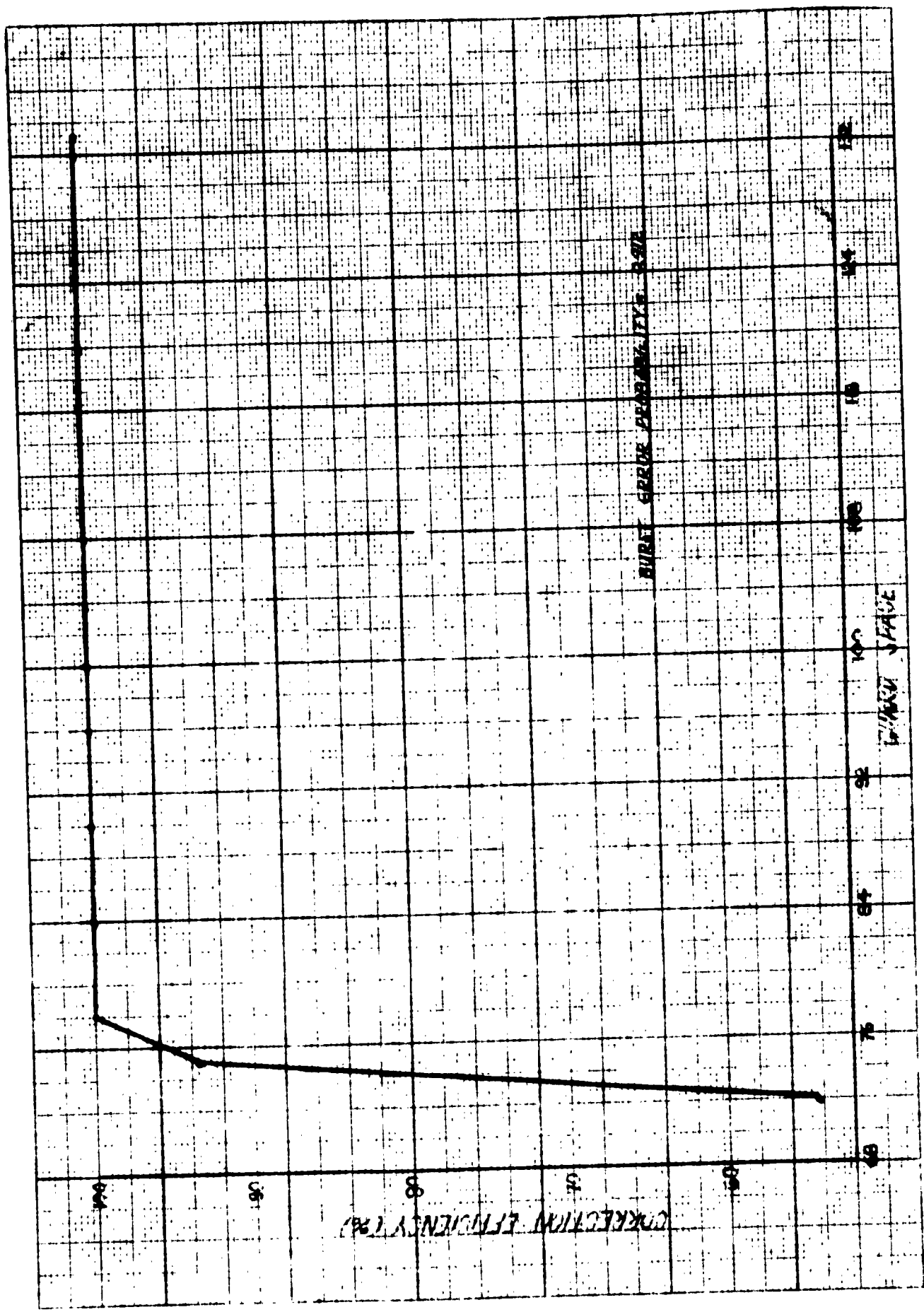


Fig. C-12

VI. INTERPRETATION OF TEST DATA

This section presents and explains the results of all tests run on the decoder. It will refer to the graphs in Figs. G-1 to G-12 as simply G-1 to G-12 respectively.

These graphs plot correction efficiency (i.e., the percentage of data-digit errors present in the decoder input that were corrected by the decoder) versus numbers of digits in the corresponding guard space. According to the theory in Report No. RADC-TN-607, and the construction of the decoder, all errors should have been corrected with guards of 132 or more digits. This was verified by test. Also, said report conjectured that a guard of 72 digits was absolute minimal for 1/4 redundant codes capable of correcting bursts of length up to 12 (depending on phase) and having all symmetric parity check sequence patterns. The symmetry condition was for easy instrumentability. Thus, G-1 to G-12 clearly show the efficiency and great practicality of the code for guards down close to the conjectured best bound of 72. It should be noted that uncorrected errors did not come through the decoder in big bunches except at guards near 72 and with high probabilities of error. Indeed, in such cases more errors seemed to be created by the decoder than corrected by it at times when it became confused. But this is not surprising.

For the most part, the test results are self-explanatory, but a few points deserve comment. First of all, G-1 to G-12 were formed from data lying well within confidence ranges establishable by theoretical statistics. In runs having a relatively large number of uncorrected errors, decoder input sequences of from 20 to 30 thousand digits were tested. In runs with fewer uncorrected errors, up to 100 thousand digits were tested. Hence, the graphs are, in fact, statistical.

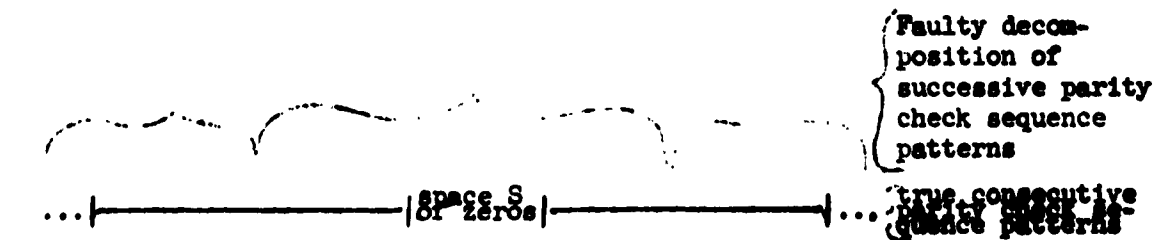
G-2, G-3, and G-4 exhibit most markedly an interesting "phasing" phenomenon common to linear recurrent codes having all symmetric parity check sequence patterns. Note that the curves peak at guards of 84 and 108, and minimize at guards of about 6 and 120. Now, at a guard of 72

successive parity check sequence patterns can juxtapose but not overlap. Hence, at guards of 84, 96, 108 and 120, these patterns must separate by at least 3, 6, 9, and 12 digits respectively.

Recall that the M matrix for this code (cf., the above mentioned report, Theorem 4 and Corollary, pp. 24, 25)*, given by

M =	Symmetry axis	Phase 1		
		. . 1	. . .	. . .
		. . .	. . 1	. . .
		. . .	. . .	. . 1
		Phase 2		
		. 1 .	. . .	. . .
		. . .	. 1 .	. . .
		. . .	. . .	. 1 .
		Phase 3		
		1 . .	. . .	. . .
		. . .	1 . .	. . .
		. . .	. . .	1 . .
		Phase 2		
		. 1 .	. . .	. . .
		. . .	. 1 .	. . .
		. . .	. . .	. 1 .
		Phase 1		
		. . 1	. . .	. . .
		. . .	. . 1	. . .
		. . .	. . .	. . 1

implies two blocks of these consecutive parity check sequence digits for each error phase. Hence, if consecutive columnar sums of M -- i.e., consecutive parity check sequence patterns -- are to be analyzed incorrectly in the decoder as shown below



(cf., the above mentioned report, Section VII)

the space S must most probably contain an even multiple of 3 zeros. Hence, guards of $72 + 2(12) = 96$ and $72 + 4(12) = 120$ tend to minimize decoder efficiency while those of $72 + 1(12) = 84$ and $72 + 3(12) = 108$ tend to maximize it. The series of graphs reflect this argument.

The second comment is that since all tested guard numbers are divisible by 4 (the basic block length of the code), it is important to be sure that the curves in G-1 to G-12 are not special cases of the code's operation. The curves are general since with a burst error probability less than about $1/3$, the first digit that is capable of starting a burst most likely won't be in error (i.e. its probability of being in error is less than $1/3$). Hence, in such cases, there is a fairly even statistical distribution among the phases of when the first digit error occurs. The actual error burst begins at the first corrupted digit. For higher probabilities (see G-5 to G-12), correction efficiency is so good that true fluctuations in the efficiency curve remain small regardless of the phase most bursts begin at. Thus, the segmented curves G-1 to G-12 are in fact good approximations.

VII. EQUIPMENT CHARACTERISTICS

Most of the logic modules used in the construction of the encoder-decoder were manufactured by the Navigation Computer Corporation of 1621 Snyder Ave., Philadelphia, Pa. The 300 Series Modules are 5" x 6" glass epoxy printed circuit cards, which in general, contain five individual stages. Supply voltages required are plus and minus 12 volts, and + 100 for neon indicators. All of the system modules utilize negative logic.

A common reset buss is present on all multi-stage flip-flop units. Reset requirements are a negative square wave (0v to -10v) at least 5 microseconds wide with a negative rise time of 0.1 microsec/volt. In general, this is the purpose of the delay amplifiers; to drive the guard space and burst length counter resets.

Several of the modules have been modified slightly to change circuit operating regions, drive capabilities, etc. However, these changes are minor, (usually resistance values) so modified circuit diagrams are not shown. If the modification involved a change in the basic circuit configuration, the modified schematic is shown.

Since the upper frequency range of operation is approximately 20 kilocycles per second, the 100kc system clock could not be used without dividing it down by a factor of 5. Unfortunately, budget limitations prohibited us from obtaining the necessary division circuitry. Therefore, an external clock (TEKTRONIX 161 pulse generator) was used. The required clock pulse must be at least -4 volts in amplitude and two microseconds wide for reliable operation.

In order to obtain statistically independent samples from the voltage noise source a General Radio Random Noise Generator Type No. 1390-A was operated with upper frequency components of 25kc when the equipment was operating at a 10kc rate.

VIII. OPERATION OF EQUIPMENT

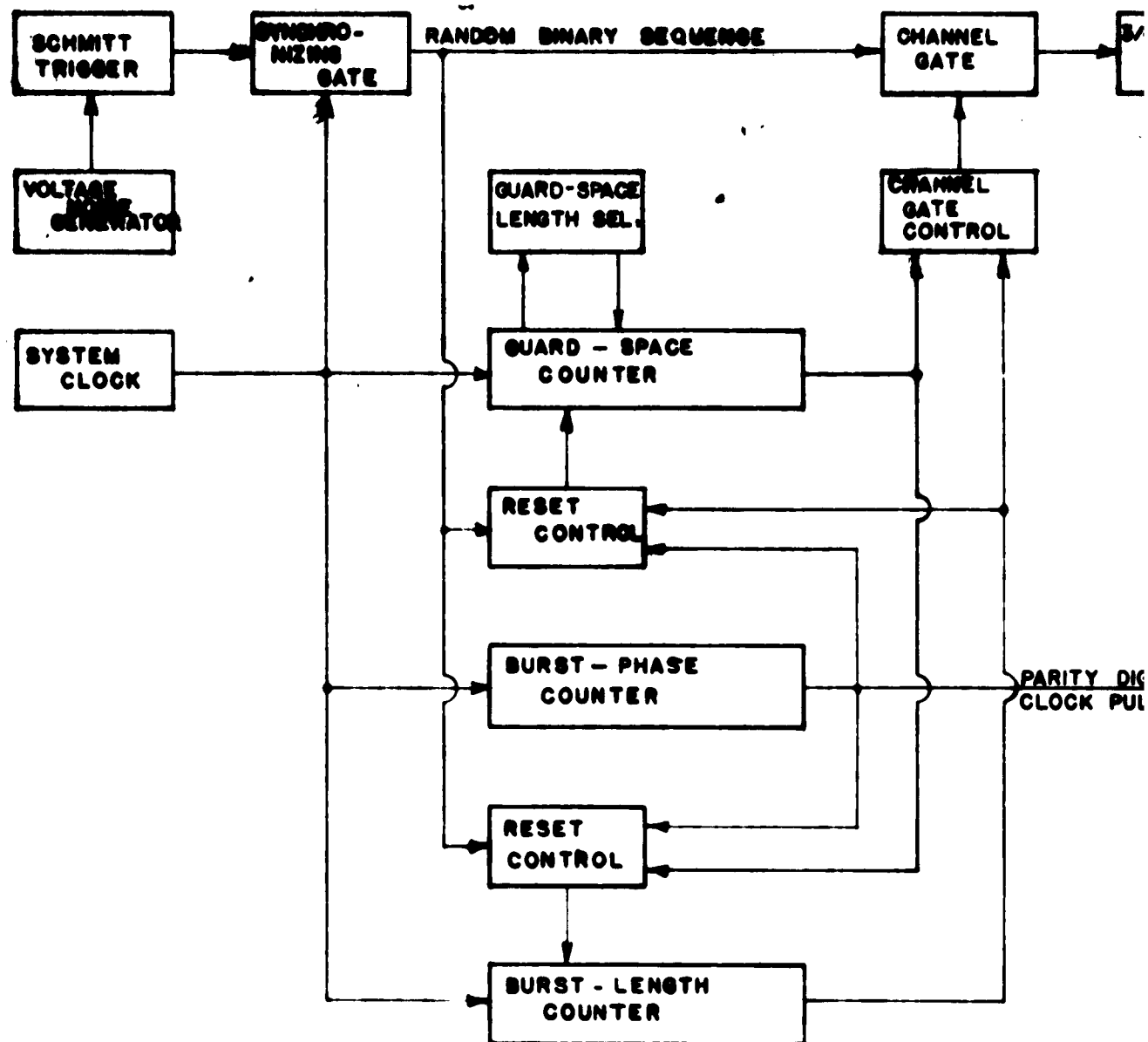
The following is a procedure for proper operation of the equipment:

1. Connect a random noise generator to the indicated terminals.
2. Connect a pulse generator (10kc on lower rep. rate, -4 volts amplitude, 2MS wide) to the input marked external clock.
3. Connect an electronic counter to each of the terminals marked channel gate out and decoder out. (If counters are set to totalize for data collecting a divider may be required to reduce the input level to the counter).
4. Turn on power supply switch.
5. Adjust the delays in the following order:
 - a. D_6 to 5.3 μ s
 - b. D_3 to 27.0 μ s
 - c. D_7 to 32.0 μ s
 - d. D_5 to 5.8 μ s
 - e. D_2 to 17.0 μ s
 - f. D_4 to 7.0 μ s
 - g. D_1 to 17.0 μ s
6. Turn on random noise generator.

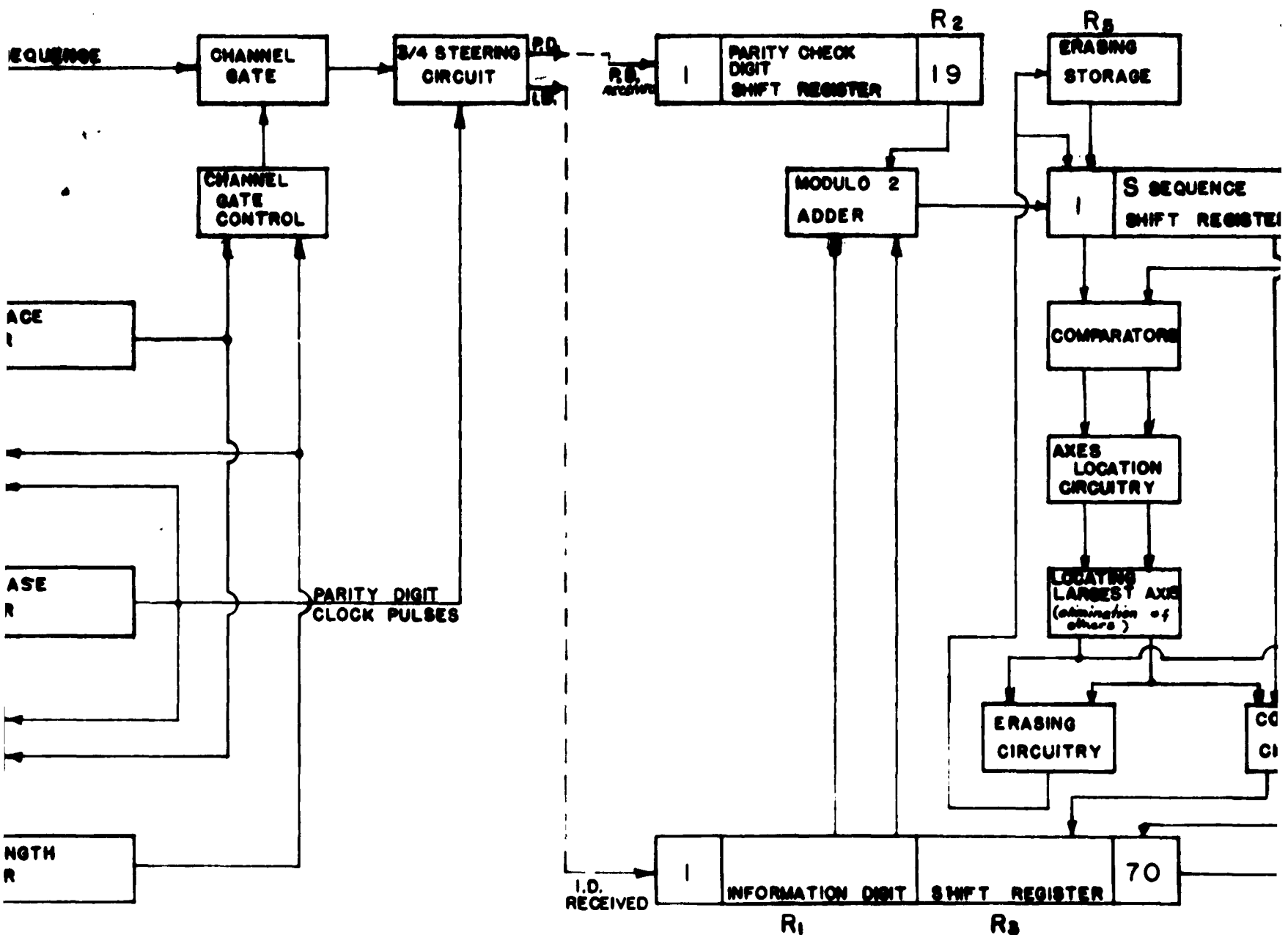
7. Immediately, lights indicating errors should progress across the shift register R_3 in bunches. All of these should be extinguished, after an axis is found, before reaching the end of the register. Occasionally, the smaller axes are selected and a digit will be corrected in the last cell of the register, so it may appear not to have been corrected.
8. A switch labeled probability must be used in conjunction with the random noise generator level control to obtain error probabilities ranging from zero to one.
9. Guard length may be adjusted by the following procedure. Each binary decimal counter module has within it an AND gate connected to all four cell outputs. If the outputs can be taken from either the ONE or ZERO side of each cell allowing a four celled counter to scale by any number from one to sixteen. The guard space counter uses a cascade of two of these modules. Thus, each guard space can be taken as the product of the scale factor of one module times that of the second. To scale each module, connect cells to AND gate as shown.

<u>Divide by</u>	<u>Cell</u>	<u>Divide by</u>	<u>Cell</u>
5	$\bar{1} \bar{2} 3 \bar{4}$	13	$\bar{1} \bar{2} 3 4$
6	$\bar{1} 2 3 \bar{4}$	14	$\bar{1} 2 3 4$
7	$\bar{1} 2 3 \bar{4}$	15	$\bar{1} 2 3 4$
8	$\bar{1} 2 3 \bar{4}$	16	$\bar{1} 2 3 4$
9	$\bar{1} \bar{2} \bar{3} 4$		
10	$\bar{1} 2 \bar{3} 4$	Note: $\bar{1}$ refers to zero side	
11	$\bar{1} 2 3 4$	of cell 1.	
12	$1 2 3 4$		

ENCODER



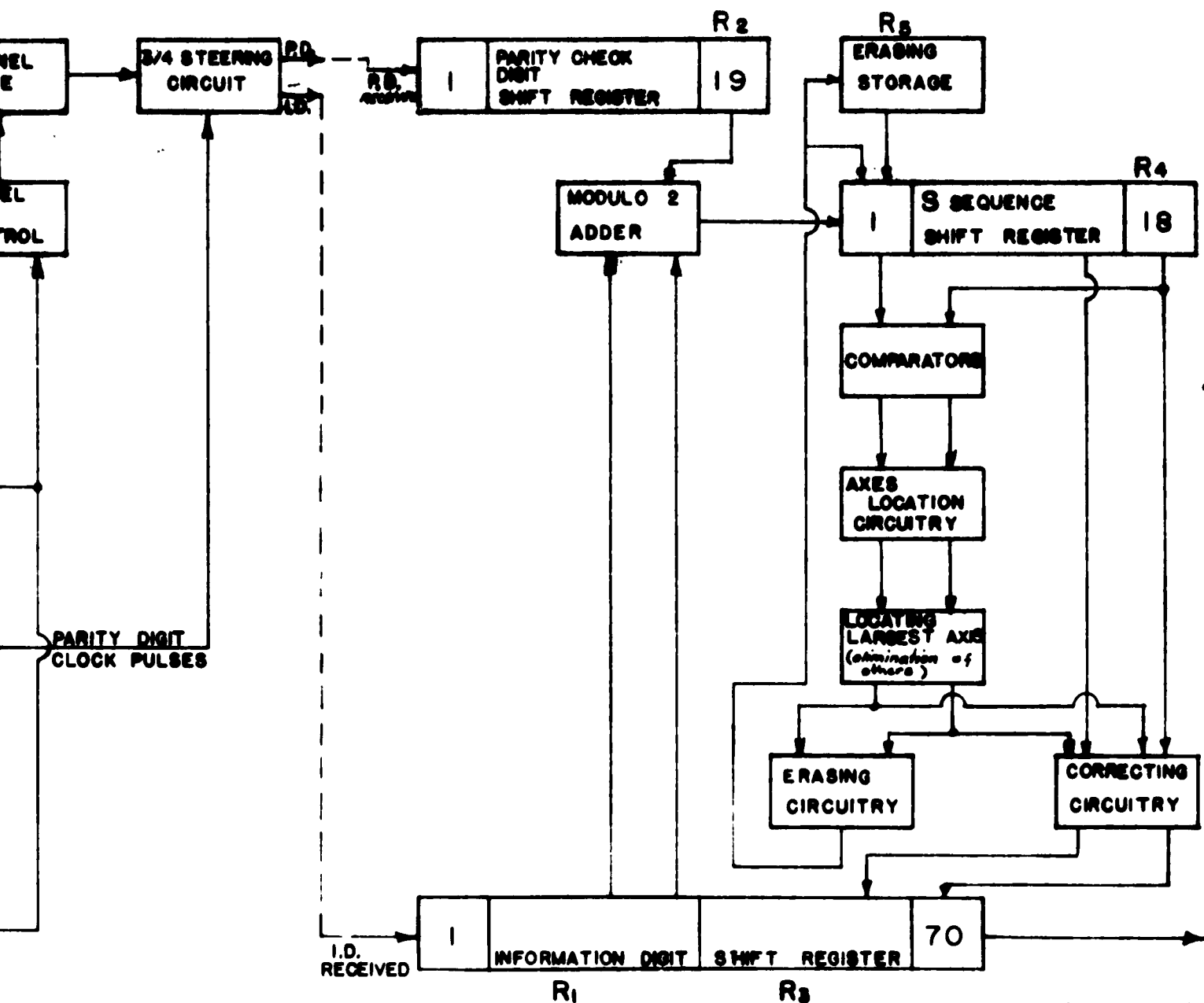
DECODER



2

ELECTRONIC R
MONTANA STA
ENCODER -
BLOCK D
FOR: ROME
CENTER GR
BASE, NEW
DRAWN BY: /

DECODER



DRWG. 192-1

ELECTRONIC RESEARCH LAB.

MONTANA STATE COLLEGE

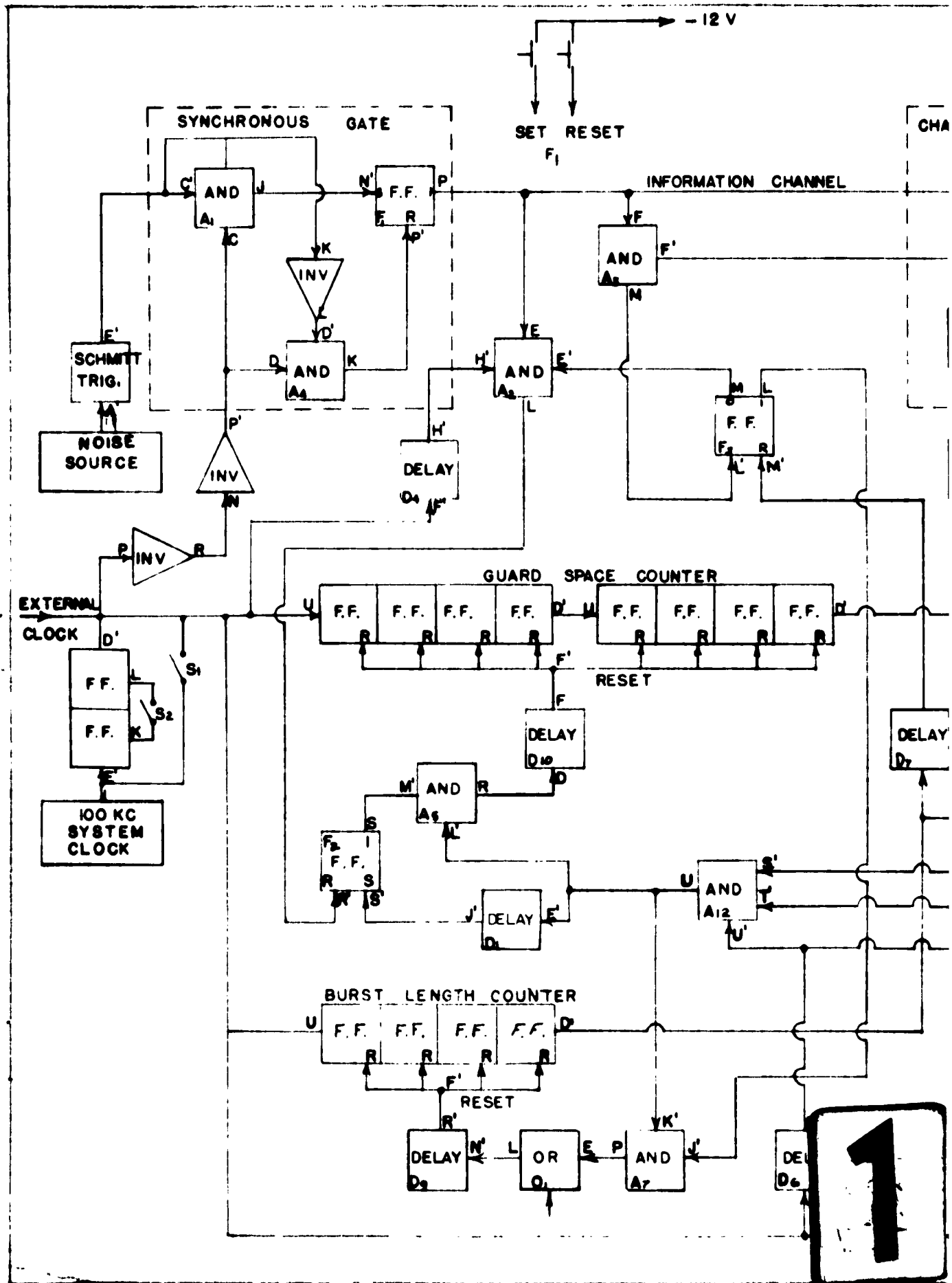
ENCODER - DECODER

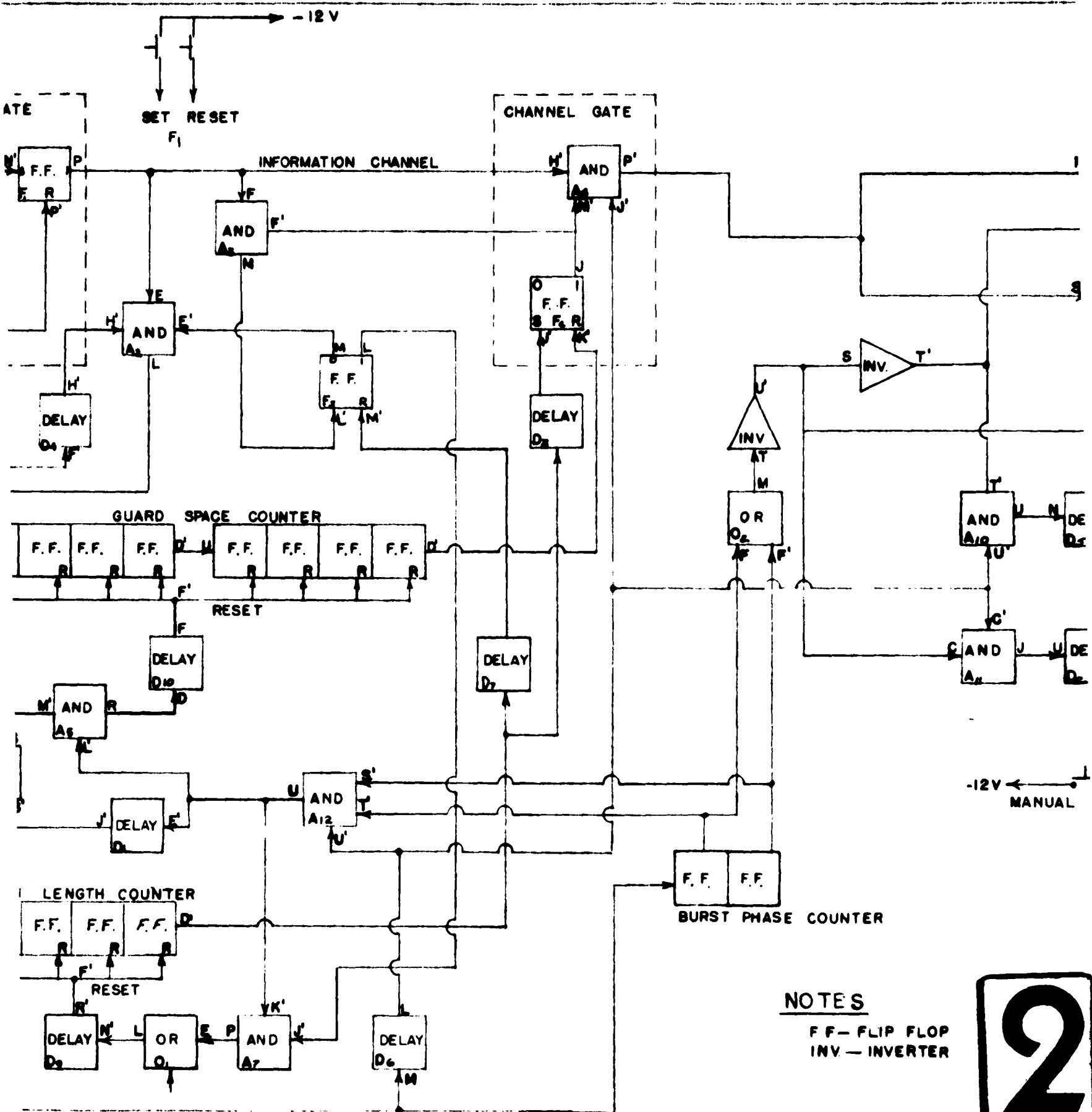
BLOCK DIAGRAM

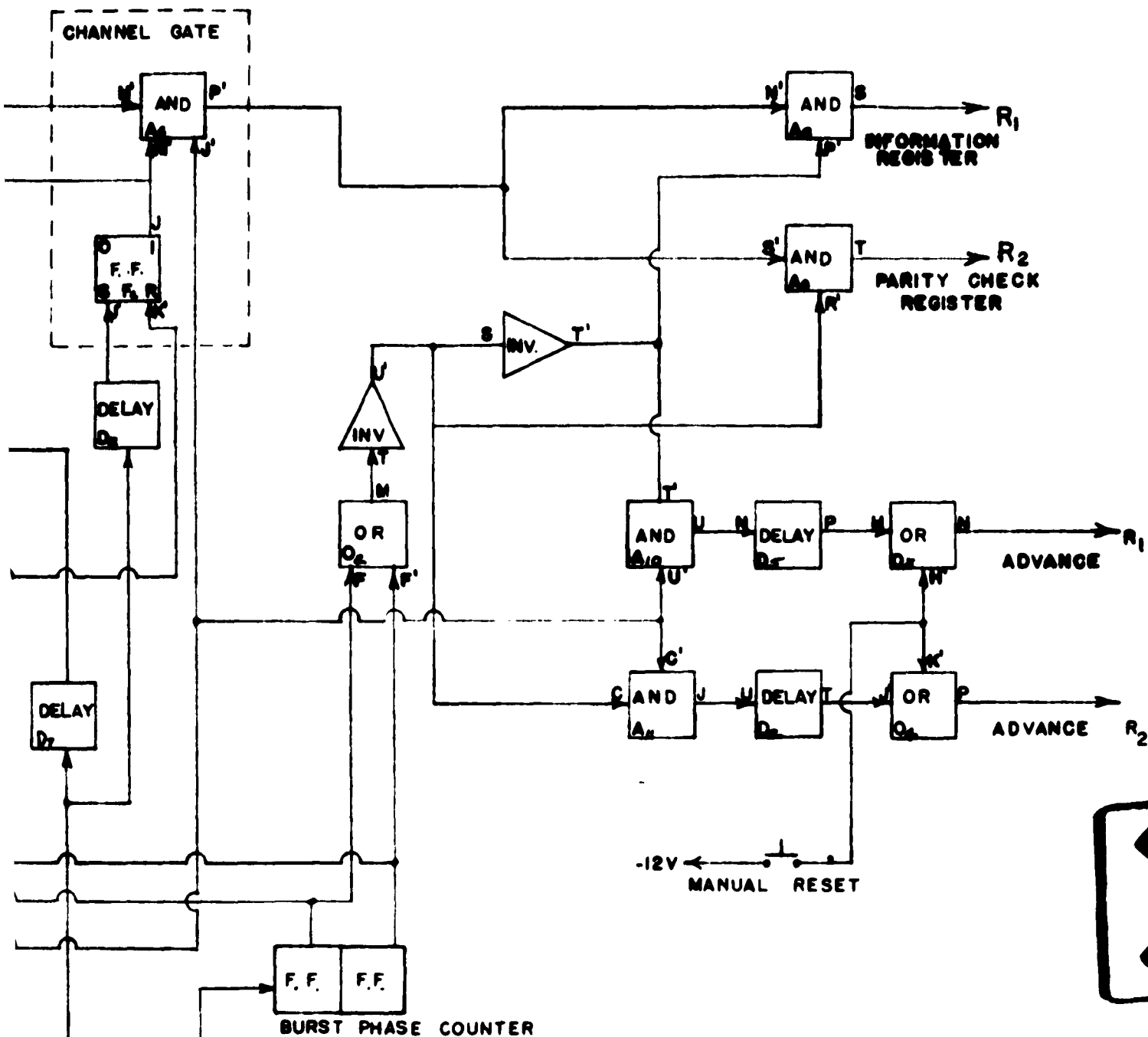
FOR: ROME AIR DEVELOPMENT
CENTER GRIFFIS AIR FORCE
BASE, NEW YORK

DRAWN BY: *RMB* DATE: 2/13/52

3



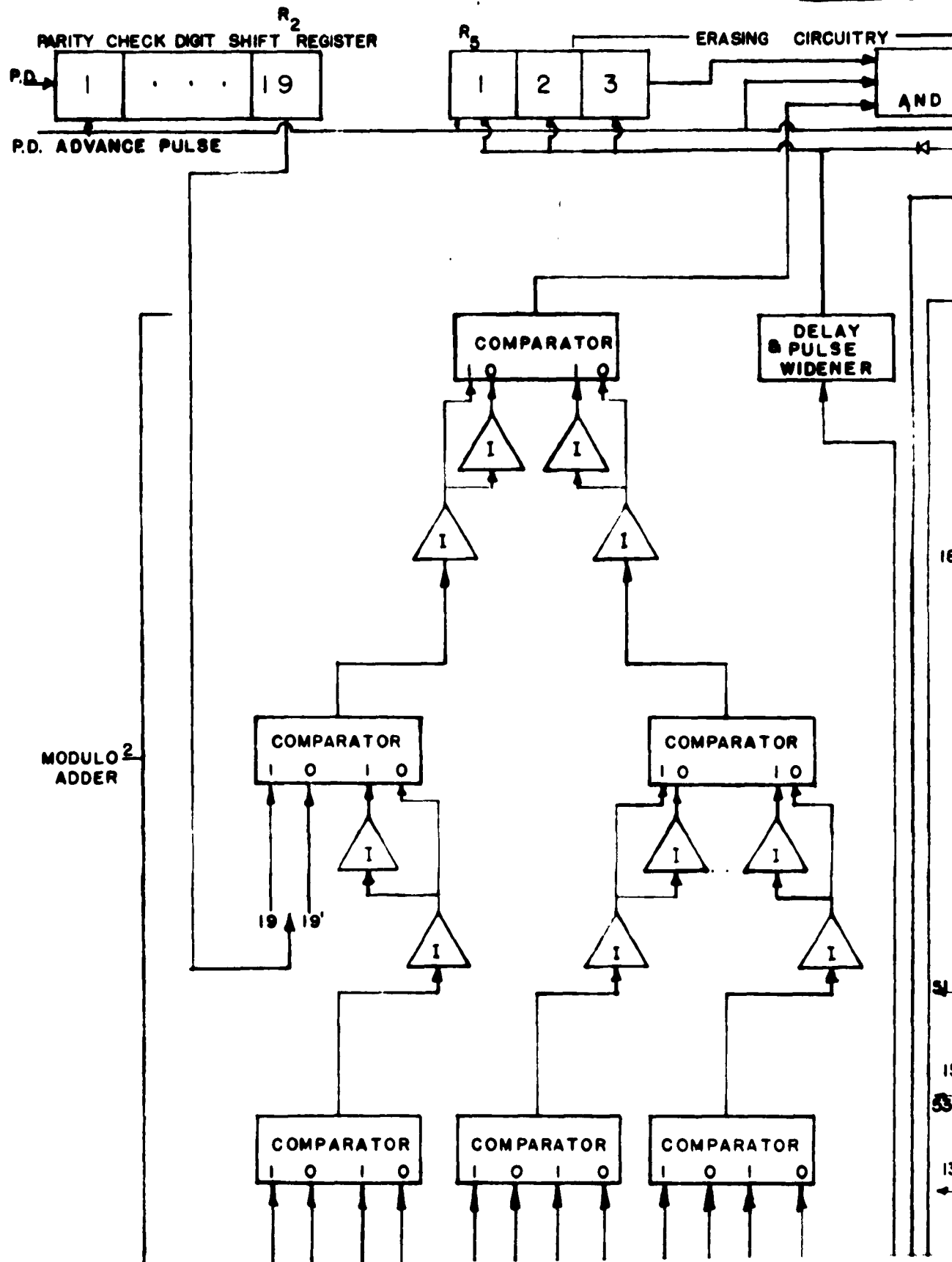


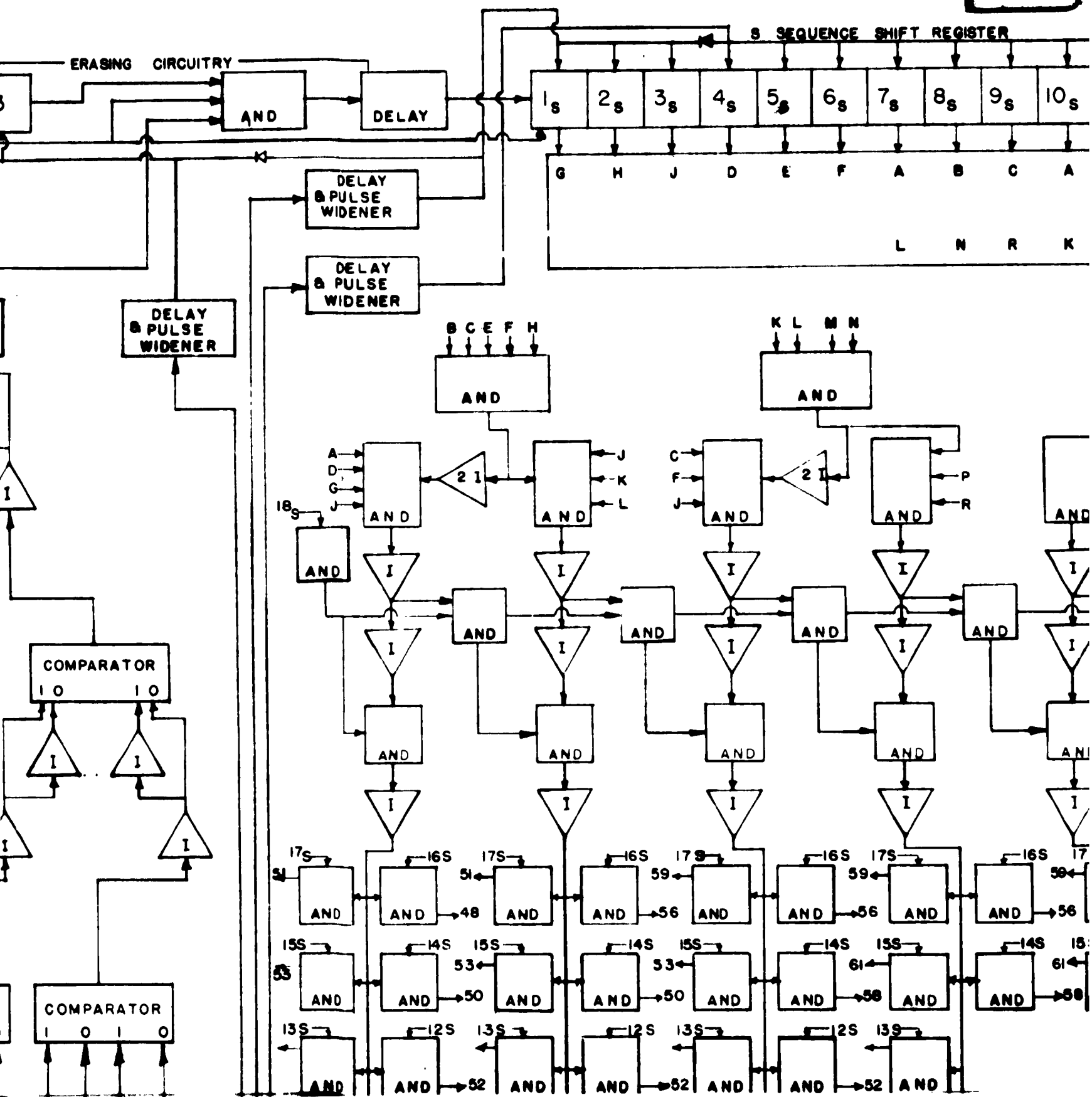


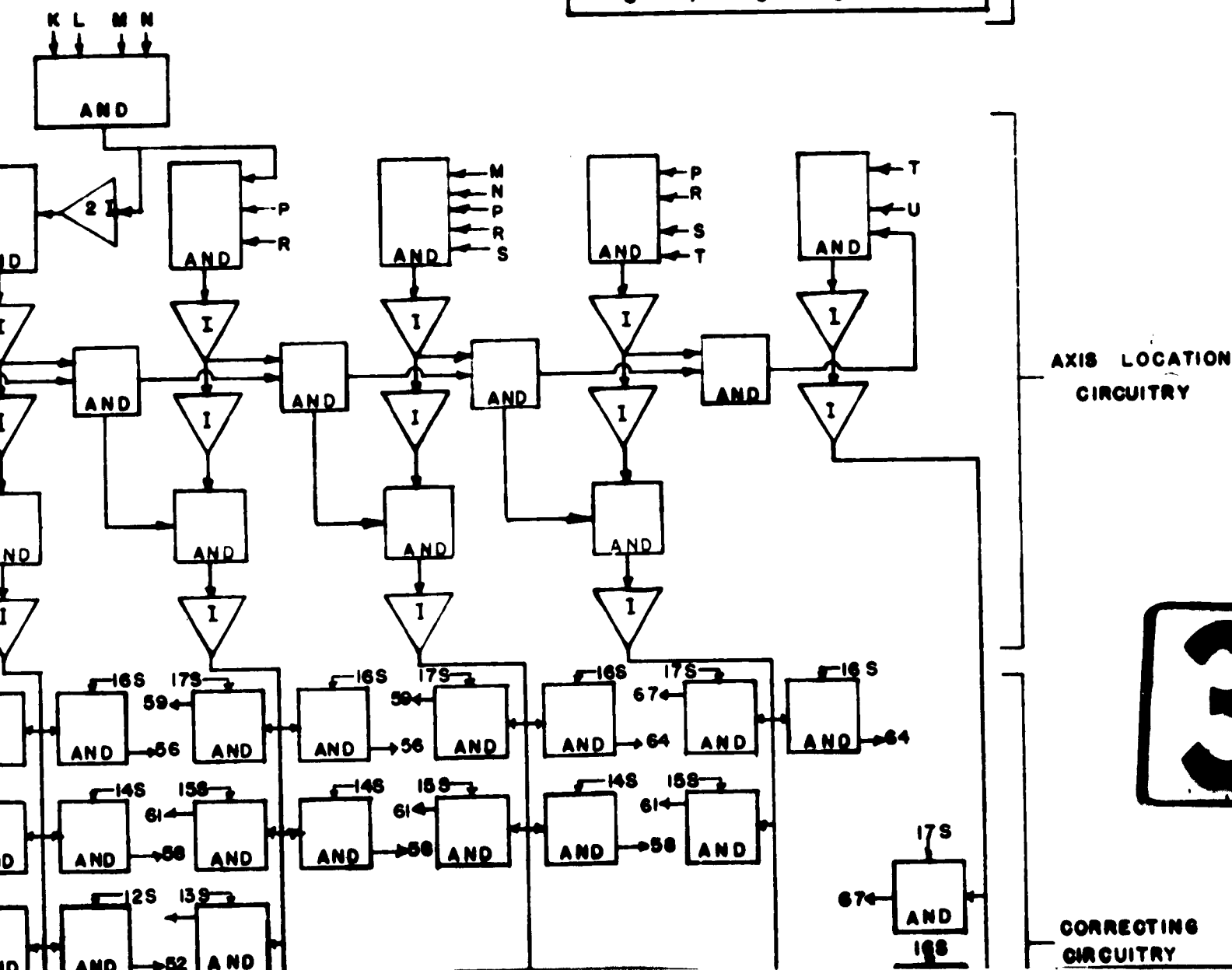
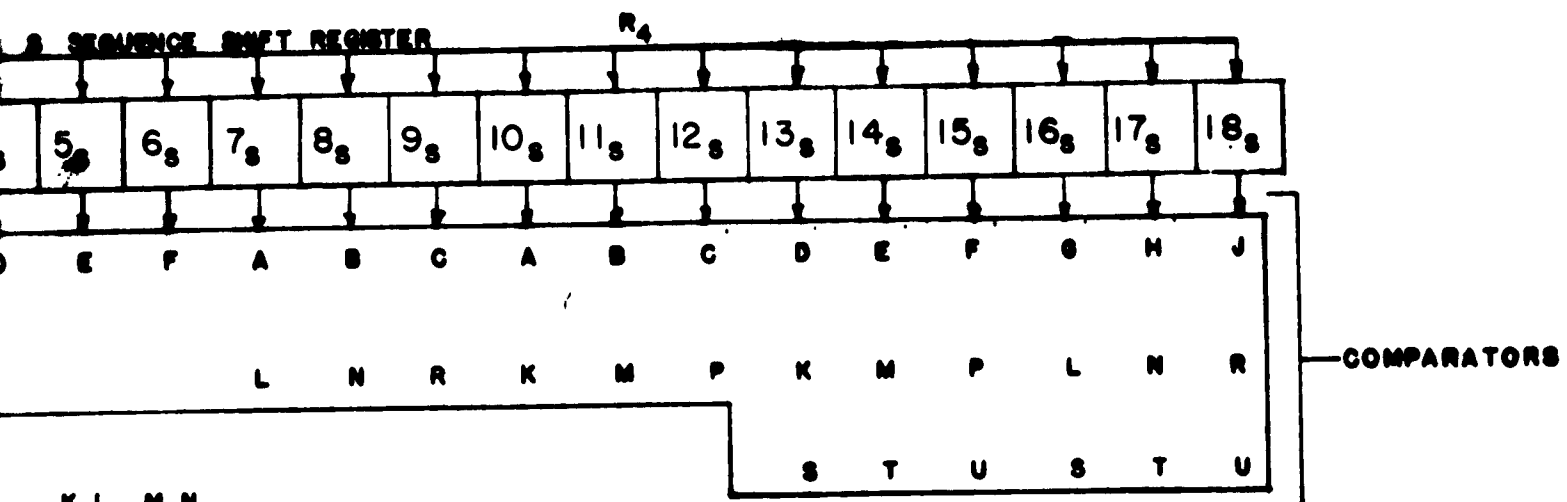
NOTES

FF — FLIP FLOP
INV — INVERTER

MONTANA STATE COLLEGE	
ELECTRONICS RESEARCH LAB.	
ENCODER	
LOGIC DIAGRAM	
FOR ROME AIR DEVELOPMENT	
CENTER GRIFFISS AIR FORCE BASE	
DESIGNED: CB. SYKES	DATE: 2/11/60
DRAWN: RMB	DATE: 2/23/62

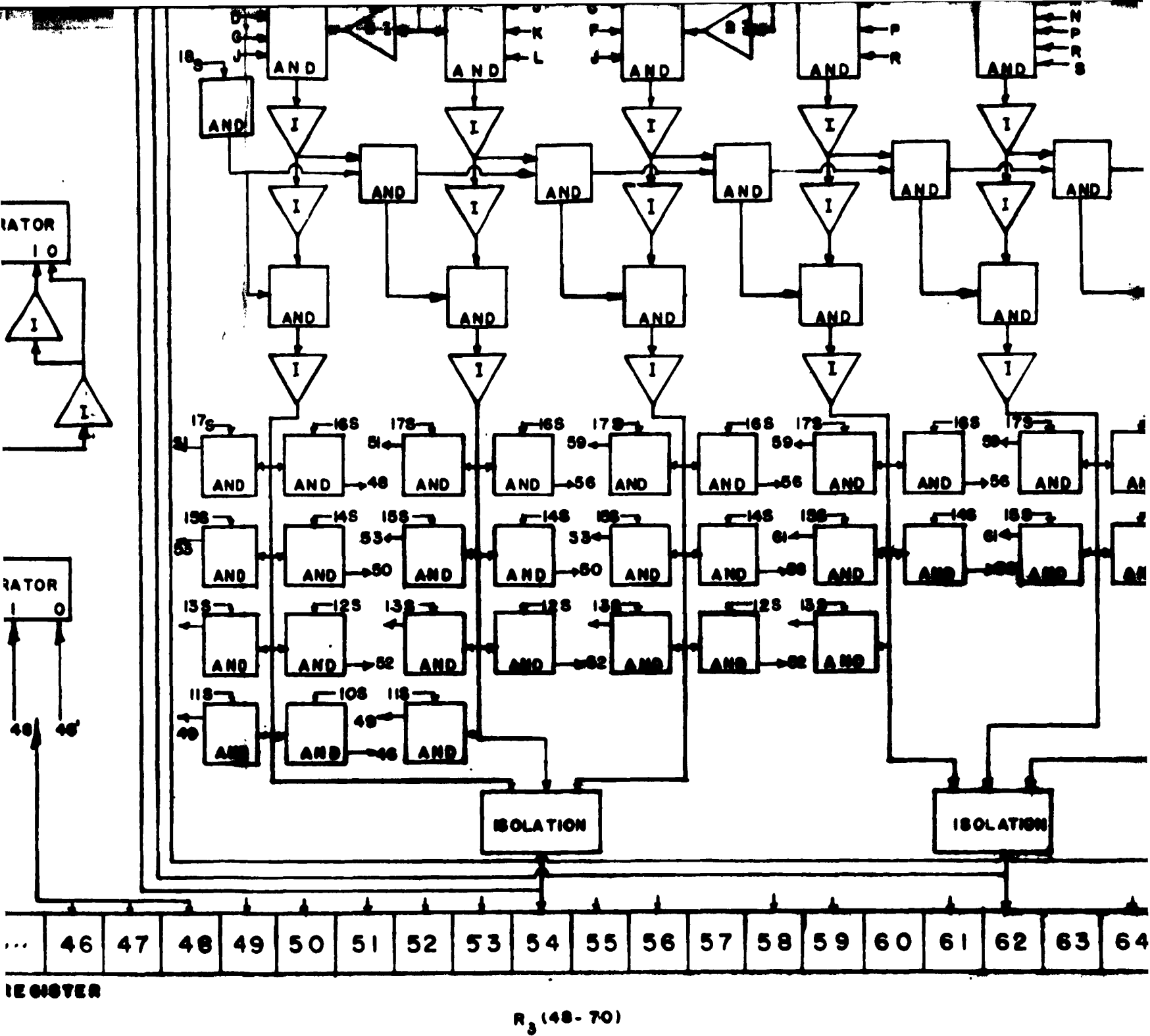




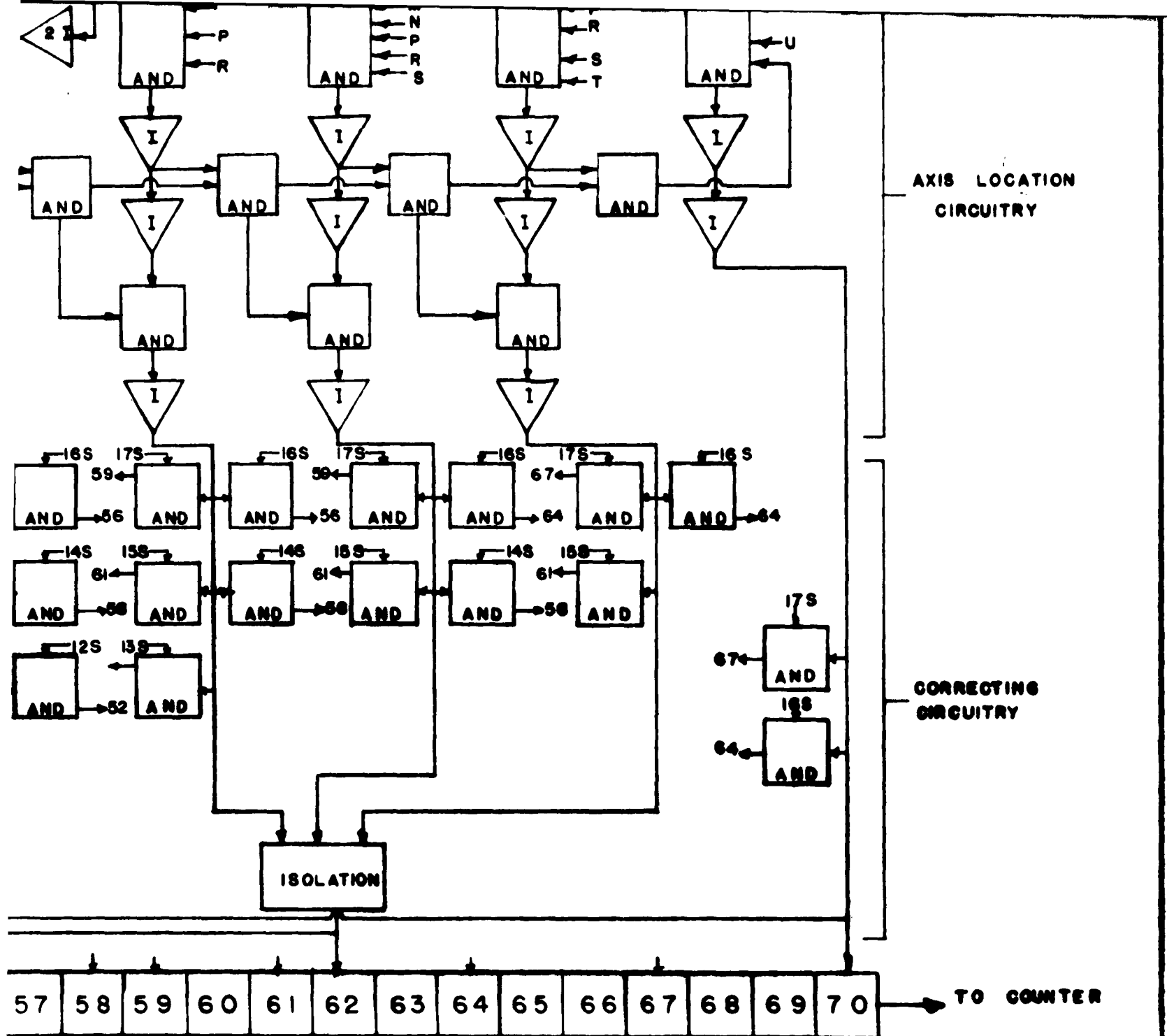


3





5



6

MONTANA STATE COLLEGE	
ELECTRONICS RESEARCH LAB	
DECODER	
LOGIC DIAGRAM	
FOR: ROME AIR DEVELOPMENT CN	
GRIFFIS AIR FORCE BASE N.Y.	
DESIGN:	DATE:
DRAWN: <i>RMB</i>	DATE: <i>2/23/62</i>