A RANDOM NOISE GENERATOR FOR A DIGITAL COMPUTER

by

John Edward Belt

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

1969

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: John Edward Belt

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

G. A. Korn

G. A. KORN
Professor of Electrical Engineering

5/12/63
Date

ACKNOWLEDGEMENTS

This project was part of a continuing study of hybrid analog-digital computer techniques directed by Professor G. A. Korn. The writer is grateful to the National Aeronautics and Space Administration Ames Research Center for support under NASA Grant NsG-646.

TABLE OF CONTENTS

		Page
LIST OF ILLUSTRATIONS		v
LIST OF TABLES		vii
ABSTRACT		viii
Chapter 1	INTRODUCTION AND BASIC DESIGN	1
1.1	Introductory Remarks	1
1.2	Design Requirements	2
1.3	Basic Organization	5
Chapter 2	NOISE SOURCE DESIGN	7
2.1	A Sampling-Type Noise Source	7
2.2	Noise Device Circuit	8
2.3	Schmitt Trigger and Clock Circuit	11
Chapter 3	THE COMPLETE NOISE GENERATOR	13
3.1	Circuit Design	13
3.2	Logic Design	15
Chapter 4	TEST PROGRAMS AND RESULTS	19
4.1	Contingency Table Test for Statistical Independence of Adjacent Bits	19
4.2	Adjacent-Bit Correlation Test	22
4.3	A Correlation Display Test	23
4.4	User Results	28
4.5	Conclusions	29
Appendix I	MECL TO DEC LOGIC LEVEL CONVERSION CIRCUIT WORST-CASE DESIGN ANALYSIS	30
Appendix II	PDP-9 PROGRAM LISTINGS STATIN, C1OUT, C2OUT, C3OUT	40
Appendix III	PDP-9 PROGRAM LISTINGS BITCOR AND CMPOUT	49
Appendix IV	PDP-9 PROGRAM LISTING CORDIS	53
REFERENCES		58

LIST OF ILLUSTRATIONS

Figure		Page
1.1	Noise Generator Block Diagram	3
2.1	Noise Source Block Diagram	9
2.2	Typical Noise Source Waveforms	9
2.3	Noise Source Circuit Diagram	10
3.1	MECL to DEC Logic Level Conversion Circuit	14
3.2	MECL Power Supply Circuit Diagram	14
3.3	Noise Generator Printed Circuit Board Locations	16
3.4	Logic Diagram of Noise Generator Shift Registers	17
3.5	I/O Interface Logic Diagram	18
4.1	Contingency Table	20
4.2	Graph of $\bar{r}$ Plotted as a Function of T	24
4.3a	Photograph of Display with Generator 1 versus Generator 2 with 0 Bit Delay	26
4.3b	Photograph of Display with Generator 2 versus Generator 2 with 0 Bit Delay	26
4.4a	Photograph of Display with Generator 2 versus Generator 2 with 5 Bit Delay	27
4.4b	Photograph of Display with Generator 1 versus Generator 1 with 1 Word Delay. Noise diode is not properly biased.	27
A.1	Differential-Amplifier Circuit	31
A.2	Differential-Amplifier Circuit with Voltage Divider	31
A.3	Equivalent Circuit with Q_2 OFF	34

List of Illustrations (Continued)

Page

A.4 Equivalent Circuit with Q_2 ON 34

A.5 Graphs of Circuit Equations Showing Solution Space 36

A.6 Equivalent Circuit with Q_2 ON 37

A.7 Circuit for Computing Emitter Resistor Value 37

LIST OF TABLES

Table		Page
1.1	Table of Noise Generator Specifications	4
4.1	Five Sets of Computed Test Statistics	21

ABSTRACT

This thesis describes the design and test results of a "true" random noise generator. Although designed for use with the PDP-9 digital computer, the design is easily adapted to any other small computer that utilizes a parallel I/O Bus.

Zener diodes provide the analog noise that is sampled, and then accumulated in an integrated circuit shift register. The contents of the shift register are then available to the computer upon execution of the appropriate I/O instruction.

Test results do not disprove the hypothesis that the noise produced is uniformly distributed with adjacent bits being statistically independent and uncorrelated. The generator has been used successfully in optimization programs using the creeping random search technique. Gaussian words can be produced at high speed by using the generator output to drive a table look-up program.

CHAPTER 1

INTRODUCTION AND BASIC DESIGN

1.1 Introductory Remarks

A digital computer or a hybrid analog/digital installation involved with Monte Carlo computations, systems simulation, or statistical computations requires a high-quality source of random numbers, i.e., a device capable of producing multibit, statistically independent, random or pseudorandom computer words at high rates. Pseudorandom numbers generated either by software (programmed algorithm) or hardware (feedback shift register) have been used, but are not always satisfactory. When generated by software, considerable effort is required to test the pseudorandom numbers for randomness properties; the computer time required for the software generation of pseudorandom numbers may interfere with or preclude real-time and hybrid computations; and, last but not least, the use of some types of pseudorandom numbers may lead to incorrect results in simulating systems with time constants that are long with respect to the cycle time of the shift register or in non-linear systems because higher-order distributions do not have desirable statistical properties (Korn 1966, and White 1967).

To avoid the problems associated with pseudorandom number generators, a "true" random-noise generator producing uniformly distributed 18 bit words derived from analog noise (noisy diodes) was developed. This report describes the design of an all solid-state

noise generator designed as an accessory for the Digital Equipment Corporation PDP-9 digital computer; essentially similar multibit noise generators would work with other digital computers having parallel I/O busses.

1.2 Design Requirements

The basic design requirement was for random words on 18 parallel data lines (see Fig. 1.1) into a digital computer, with a new word being available every 30 - 40 μsec. The 30 - 40 μsec. word rate was assumed to be adequate since the computation time associated with each noise word exceeds this 30 - 40 μsec. for most applications. The noise generator was designed for the PDP-9 computer but the design is easily adapted for use on other computers. MECL (Motorola Emitter Coupled Logic) integrated-circuit logic elements were selected for the noise generator logic because of their low cost and ready availability, and because of the low level of digital noise that they produce. However, Digital Equipment Corporation (DEC) series R logic cards were selected for implementing the I/O interface logic (i.e., I/O bus gates and device selector logic shown in Fig. 1.1), so that the necessary compatibility at the PDP-9 I/O Bus could be maintained. The noise generator was designed to operate in a typical computer environment and, since the noise generator was to be housed in a cabinet with other devices using DEC logic elements and sharing the same power supplies, the design was limited to using only the standard DEC voltages of +10 and -15 v.

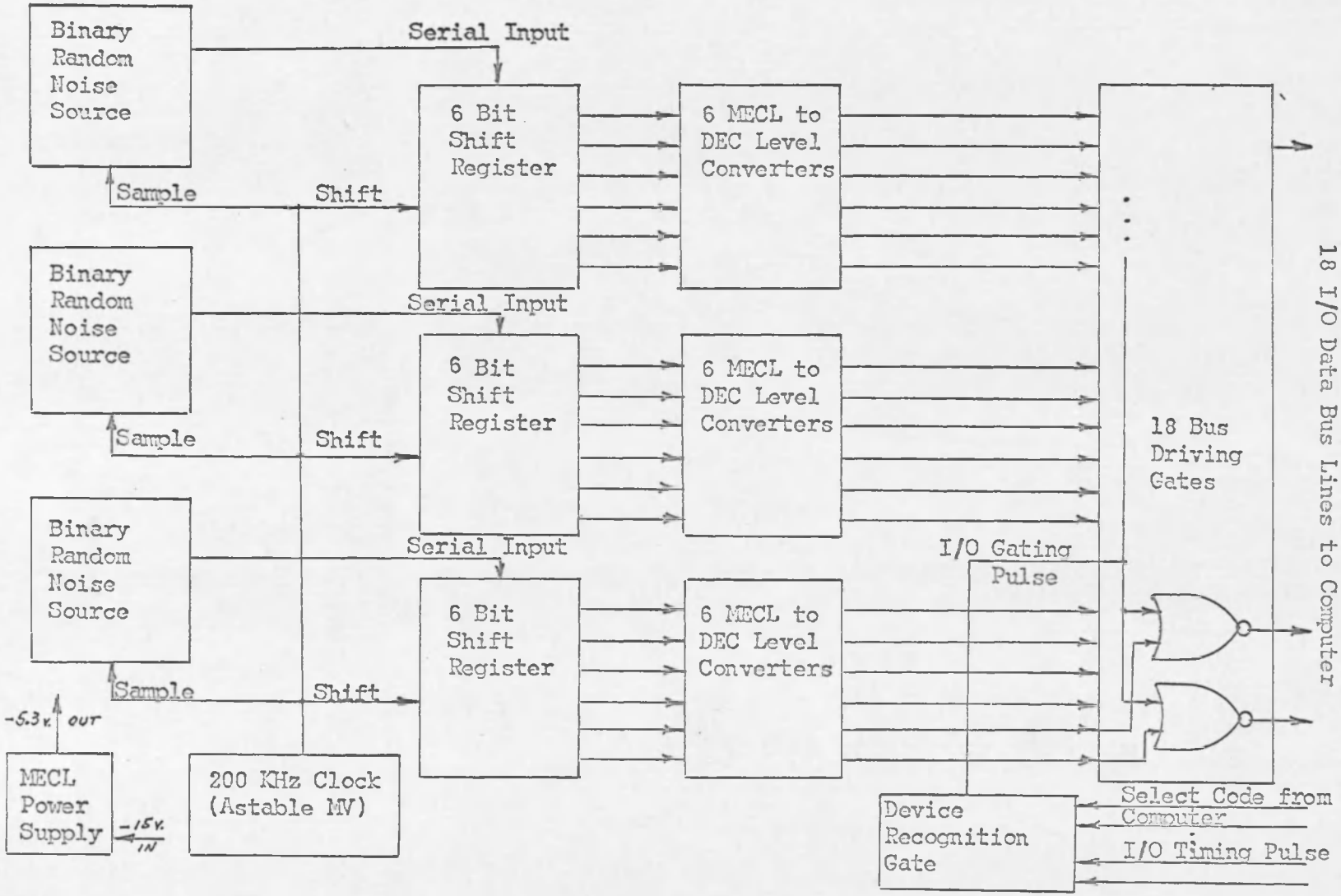

Fig. 1.1 Noise Generator Block Diagram

Table 1.1

Table of Noise Generator Specifications

General Requirements	Solid-state devices only; Plug-compatible with DEC PDP-9 I/O Bus
Word Rate	Statistically independent, uniformly distributed words available every 30 - 40 μsec.
Word Length	18 bits
Temperature	70° - 85° F. { Typical computer operating environment
Humidity	30% - 80% { Typical computer operating environment
Power Supply Voltages	-15 v. (+0.5 v., -1.5 v.) +10 v. (+1.5 v., -0.5 v.) DEC standard voltages and tolerances

1.3 Basic Organization

Preliminary tests on the inexpensive transistor-junction noise source indicated a sampling period of 5 μsec. would yield essentially uncorrelated binary noise. An 18-bit shift register will accumulate a noise word for transmission on the I/O Bus. To accumulate a full word with a single serial-input shift register would then require 90 μsec., which is a long period for the PDP-9 which has a 1 μsec. memory cycle time and a 4 μsec. I/O instruction execution time. A noise circuit for each bit in the word would be required for optimal computer performance, but this was considered to be unnecessary for most of the intended applications; this is true because the program usually requires some 30 to 50 μsec. before new noise words are needed. Three noise circuits were chosen as a reasonable compromise. Each noise circuit feeds a 6 bit MECL shift register, so that a complete new noise word is available in 30 μsec. Figure 1.1 is a block diagram of the noise generator, which consists primarily of the three noise sources and associated shift registers, 18 gates for driving the I/O Bus and a recognition gate used by the computer to select the noise generator and to time the I/O transfer. The programmer using the noise generator is responsible for assuring that the access period is greater than 30 μsec. or at least realizing in his use of the noise that the 3 shift registers shift at 200 kHz: there is no logic interlock that would prevent faster computer access to the noise generator.

The use of "Program Controlled Transfer" mode for data transfers into the PDP-9 requires that each I/O device have assigned to it

a device code (Digital Equipment Corporation, 1967b). The device code assigned to the noise generator is 64. Therefore, the machine language instruction that reads a noise word into the accumulator is 706412 and the corresponding MACRO-9 mnemonic is RAND.

CHAPTER 2

NOISE-SOURCE DESIGN

2.1 A Sampling-Type Noise Source

The noise source described in this paper is of the "sampling" type which can employ a noise-generating device whose statistical characteristics are not known in detail, except that large bandwidth is desirable. The polarity of the noise source is sampled at a sampling frequency small ($\leq 1/10$) compared to the noise -3 db bandwidth. The statistical properties of the binary noise thus generated will depend on the sampling frequency, but not on the statistical properties of the original analog noise. This "sampling" type noise generator has been described by Kohne, Little and Soudak (1965) and has been further developed for use in ASTRAC II (Handler 1967) and LOCUST (Conant 1968). H. Kohne et al have shown that the statistics of the binary noise (being accumulated in the shift register) are essentially independent of changes in the statistics of the noise source and drift in the triggering level of the Schmitt trigger. The key to this stability is the fact that the binary noise is taken from the output of a trigger flip-flop, FF_T, which has probability one-half of being in either state 0 or state 1, regardless of input probability variations, when the flip-flop input is sampled periodically with a sufficiently large number of intervening state changes. Spectral bandwidth of the original noise source has been sacrificed in this arrangement, because the flip-flop divides the mean zero-crossing rate by two.

Consider the block diagram of the noise source in Fig. 2.1. The noise device circuit drives a Schmitt trigger that follows the polarity changes (with small hysteresis, less than 100 mv.). The Schmitt trigger feeds a trigger flip-flop which is then sampled by the first shift register flip-flop under control of the clock. Figure 2.2 shows typical waveforms corresponding to points labeled in the block diagram. Figure 2.2a shows the output voltage of the noise device with the Schmitt trigger threshold voltage, S_a, superimposed. Figure 2.2b shows the output of the Schmitt trigger, and Fig. 2.2c shows the output of the trigger flip-flop that changes state for each positive transition of the Schmitt trigger. The sampling-clock waveform is presented in Fig. 2.2d, and the output of the first shift register flip-flop is shown in Fig. 2.2e.

2.2 Noise Device Circuit

A semi-conductor diode operating in avalanche mode near the knee of the characteristic curve was selected as the basic noise element, since the design requirement was for an all solid-state system using only the DEC logic voltage of -15 v. and +10 v. Other devices such as Geiger tubes, gas discharge tubes, high frequency oscillators, etc. (Shreider et al 1964) have been used as noise sources in other designs. An inexpensive silicon transistor (PSI 2N1410) was found to have the highest noise bandwidth of the semiconductors tried, and was adopted as the basic noise element.

The circuit diagram of the noise source is shown in Fig. 2.3. D_1 is the base-emitter junction of the PSI 2N1410 transistor biased

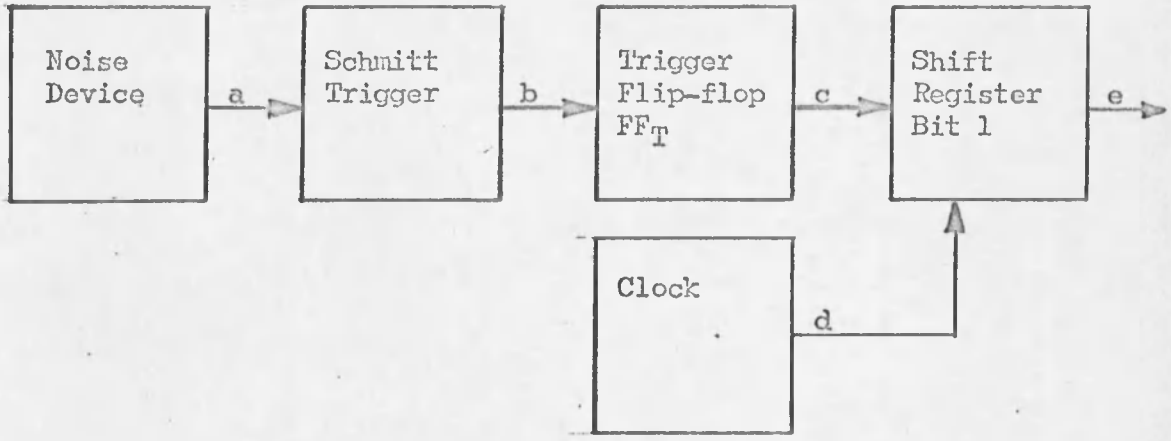

Fig. 2.1 Noise Source Block Diagram

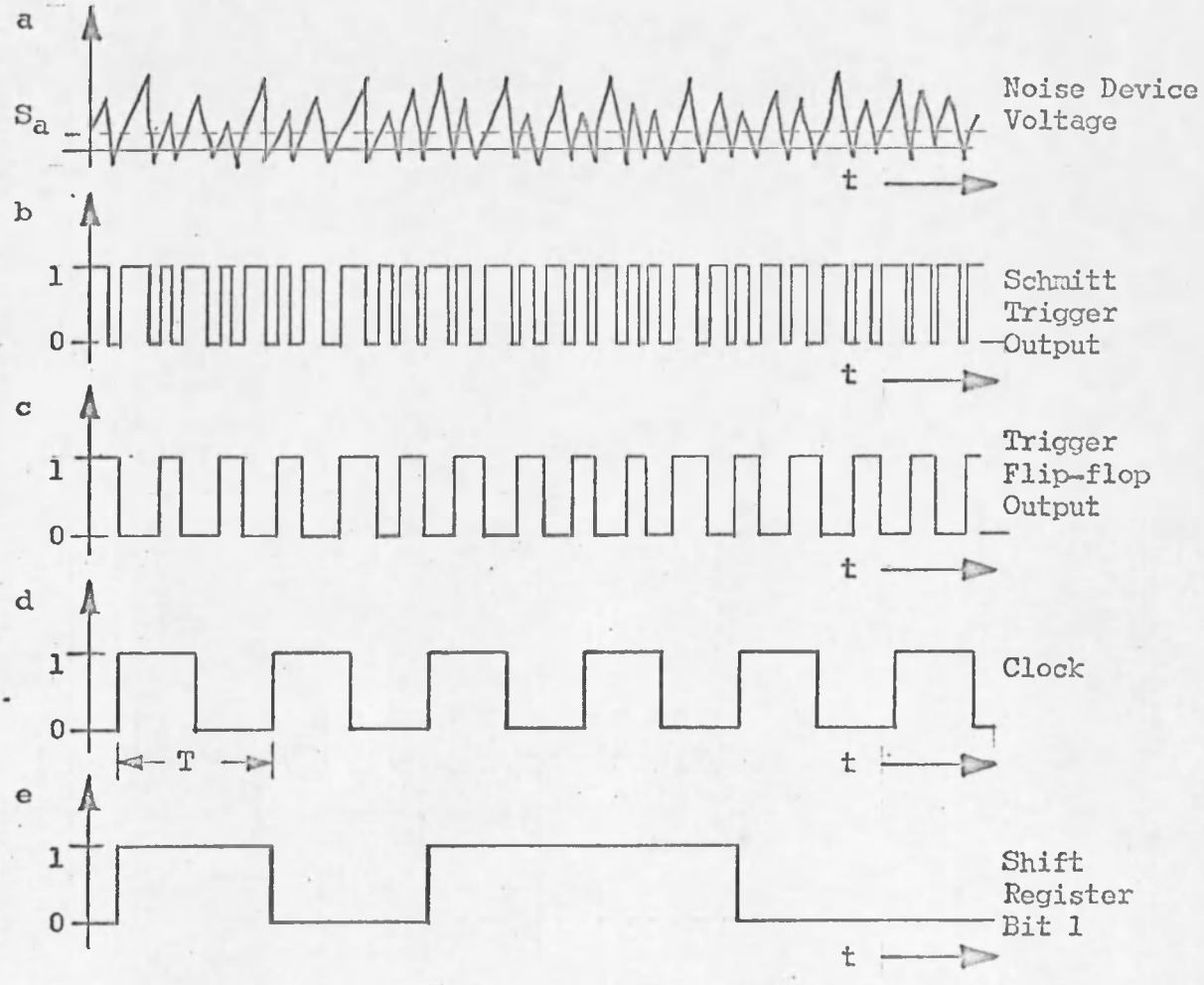

Fig. 2.2 Typical Noise Source Waveforms

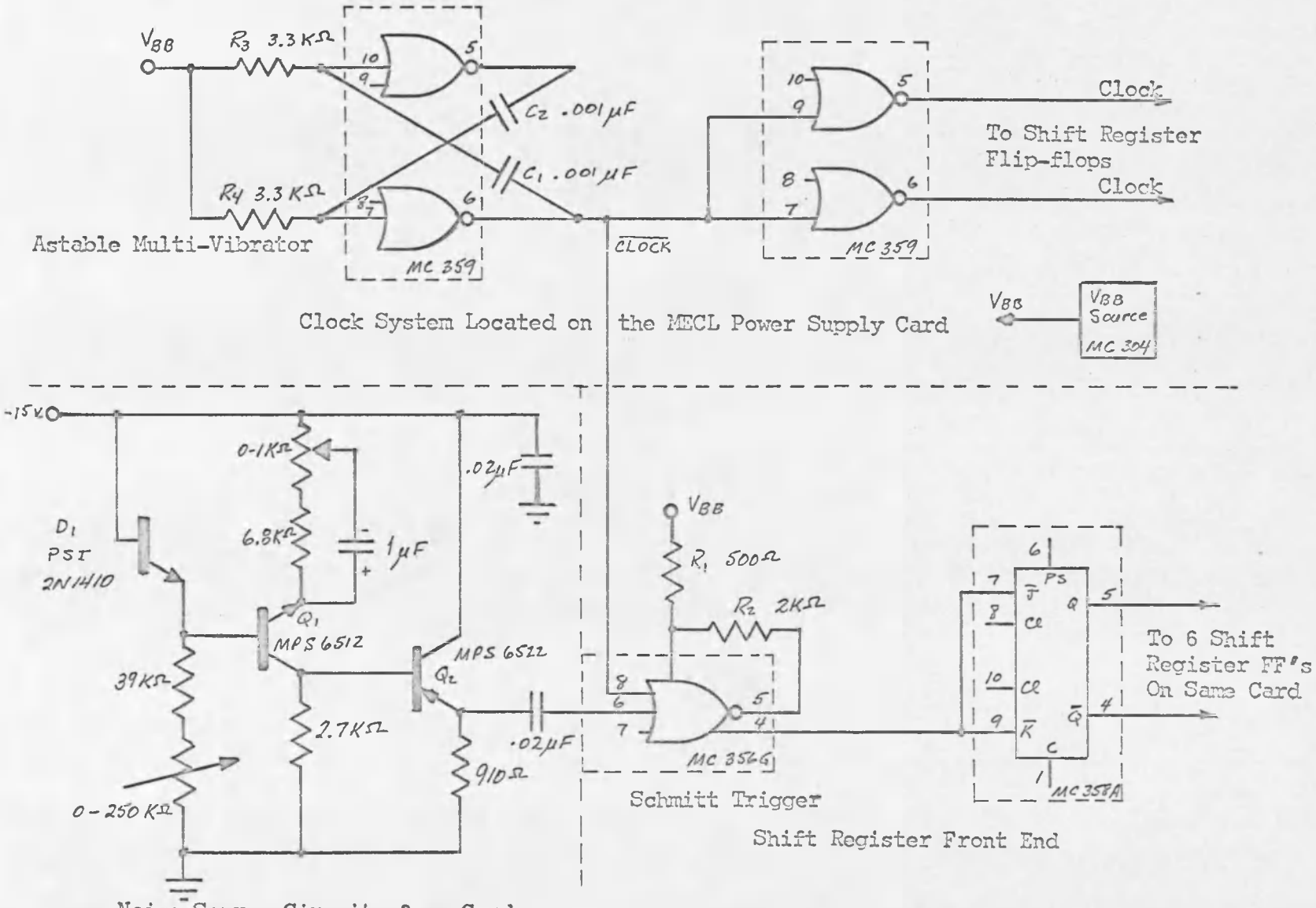

Fig. 2.3 Noise Source Circuit Diagram

into avalanche mode to produce noise. The 0-250K ohms variable resistor adjusts avalanche current for maximum noise. The noise voltage is amplified in the next stage (Q_1), which then drives an emitter follower circuit (Q_2). The emitter follower provides a low-impedance source for the Schmitt trigger, thus minimizing pick-up of signals which could affect noise statistics.

2.3 Schmitt Trigger and Clock Circuit

The logic elements chosen to implement the noise generator were MECL elements (Motorola Emitter Coupled Logic). It is particularly easy to build the Schmitt trigger and clock circuits using standard MECL gates with only a few additional external components. A circuit diagram of the Schmitt trigger is shown in Fig. 2.3. The two resistors, R_1 and R_2, form a feedback network which gives the NOR gate its latching action. When any input is a logic "1", the circuit latches to a "1" output, and when all inputs are logic "0", the circuit output is an "0". However, when all inputs are logic "0" except one which connected to the noise circuit, then a small noise voltage excursion can cause the Schmitt trigger to toggle becuase the hysteresis is small due to the feedback network.

A circuit diagram of the clock circuit is also shown in Fig. 2.3. The circuit consists of two NOR gates cross-coupled with capacitors. The two resistors bias the gate input transistors into their active regions, so that the circuit has high loop gain. One of the gates then switches and turns the other one OFF through the coupling capacitor. The capacitor voltage decays with time constant RC, and

then the other gate switches ON. Thus the circuit is an astable flip-flop, whose period is controlled by the RC time constant.

CHAPTER 3

THE COMPLETE NOISE GENERATOR

3.1 Circuit Design

The Schmitt trigger, trigger flip-flop, shift registers, and shift clock were implemented with MECL integrated circuit logic elements. For circuit compatibility at the I/O Bus, DEC series R logic cards were used to implement the I/O interface logic. This arrangement requires a logic level conversion circuit from the MECL shift registers to the DEC I/O Bus drivers. MECL logic levels are nominally -0.75 v., and -1.55 v. while DEC levels are nominally 0 v. and -3.0 v. Figure 3.1 is a circuit diagram of the conversion circuit. The circuit uses a differential amplifier input because of the small MECL voltage swing and this amplifier drives the output through a voltage divider network in order to get 0 v. when Q_2 is ON. A worst-case d-c design analysis is found in Appendix I. The conversion circuits are located on the same printed wiring boards with the MECL shift registers.

The MECL circuits require a power supply of approximately -5 v., so a power supply operating on the available -15 v. DEC logic voltage supply was designed. Figure 3.2 is a circuit diagram of the power supply which is simply an emitter follower circuit where the base voltage and thus the output voltage is set by a zener diode.

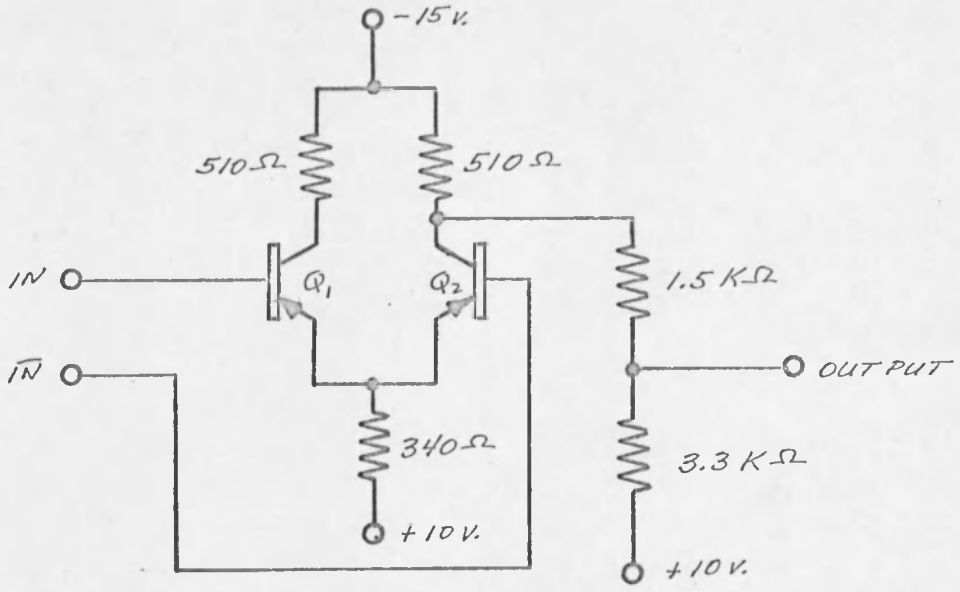

Fig. 3.1 MECL to DEC Logic Level Conversion Circuit

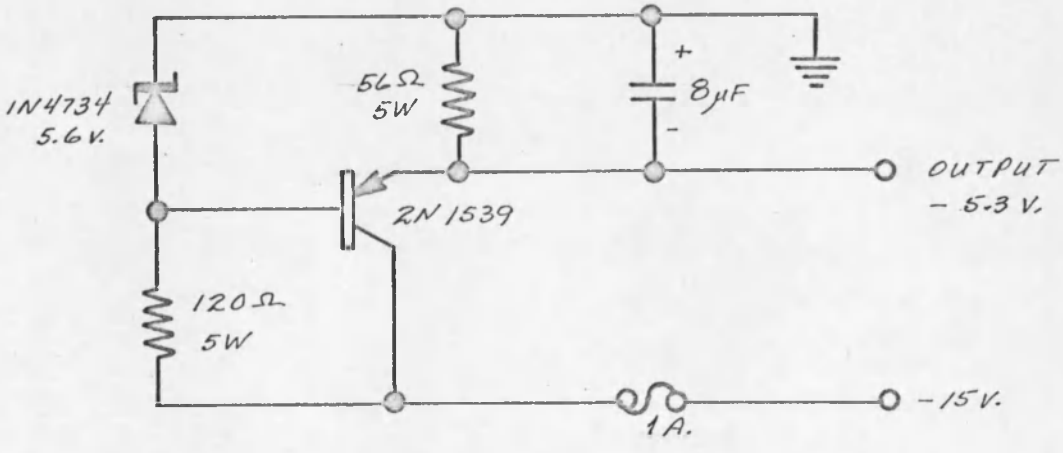

Fig. 3.2 MECL Power Supply Circuit Diagram

3.2 Logic Design

The logic design of the noise generator was straightforward and consisted primarily of meeting the application specifications for the MECL and DEC logic elements (Digital Equipment Corporation, 1967a, and Motorola Corporation, 1966). Figure 3.3 shows the printed circuit board locations in the noise generator and the board and pin number identification system that is used in the logic diagrams. Figure 3.4 is the logic diagram of the noise sources and shift registers and Fig. 3.5 is the logic diagram of the I/O interface logic. This interface logic can easily be changed to adapt the noise generator to other computer systems since it consists simply of a set of I/O Bus drivers and a recognition gate (W-103 device selector).

Fig. 3.3 Noise Generator Printed Circuit Board Locations

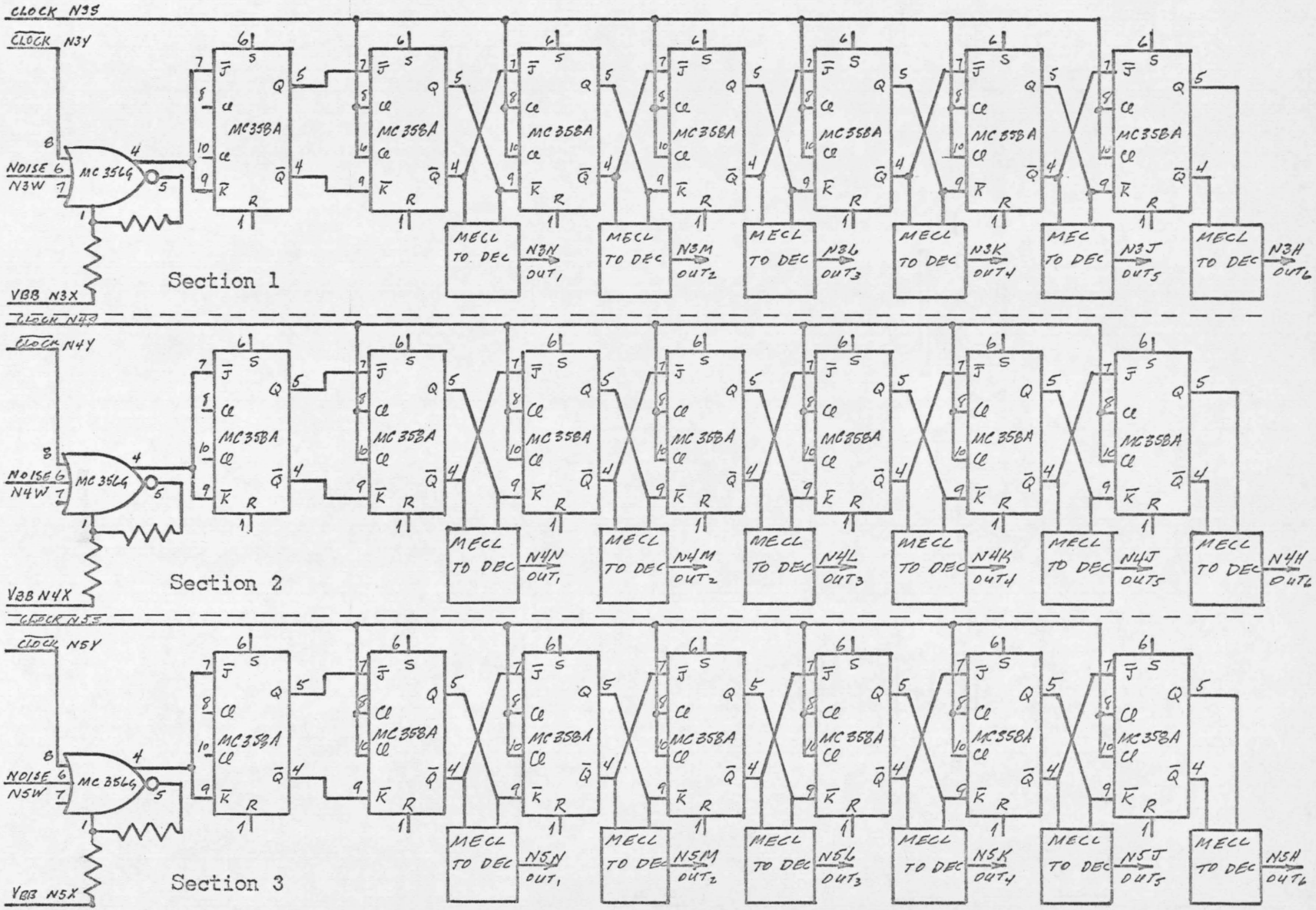

Fig. 3.4 Logic Diagram of Noise Generator Shift Registers

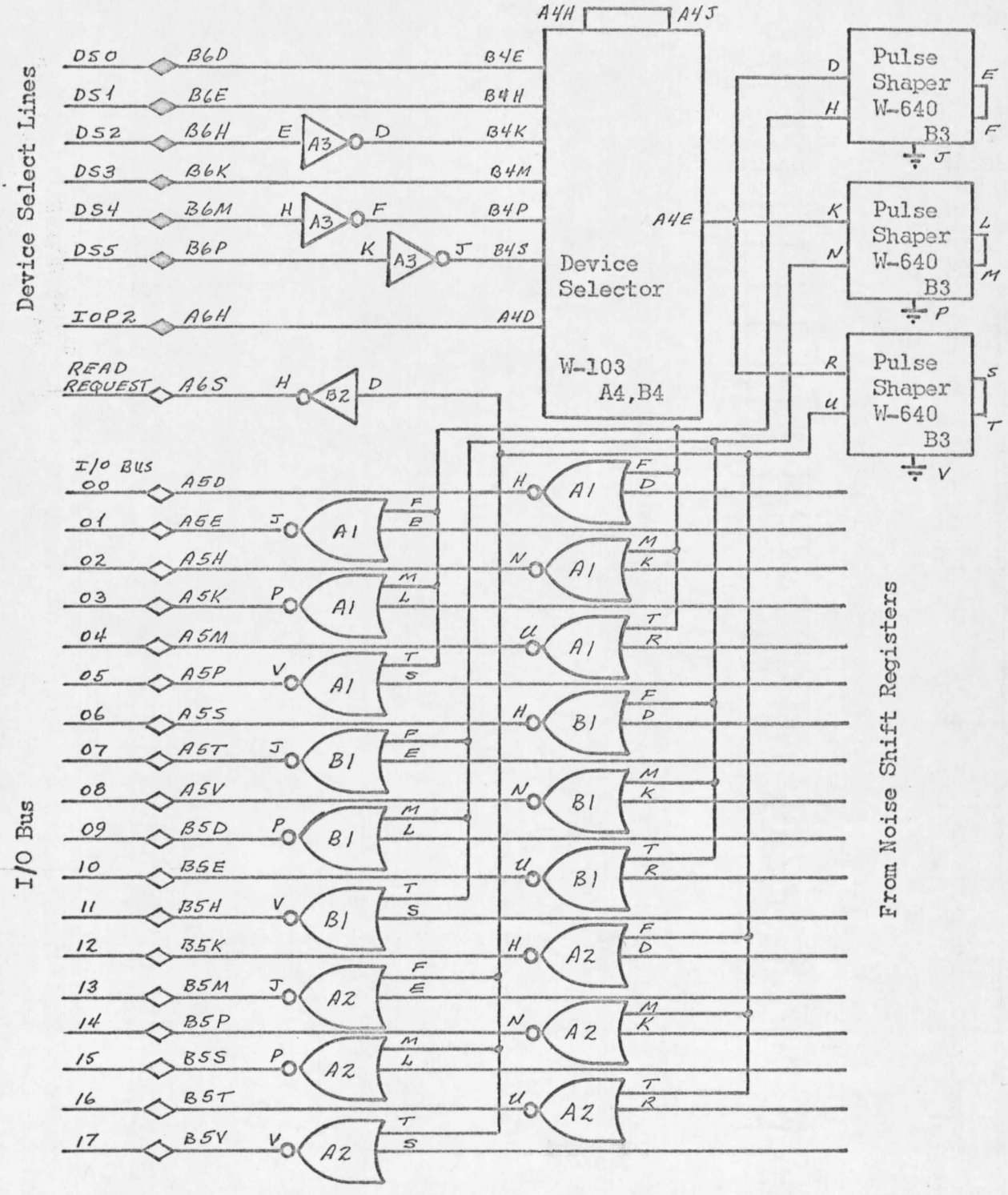

Fig. 3.5 I/O Interface Logic Diagram

CHAPTER 4

TEST PROGRAMS AND RESULTS

4.1 Contingency Table Test for Statistical Independence of Adjacent Bits

Adjacent bits accumulated in the noise generator shift registers should be statistically independent. A PDP-9 program that uses a contingency-table technique (Cramer 1946, and Korn and Korn 1968). for testing the noise generator for statistical independence was written. Let x be the value (0 or 1) of a noise generator bit, and let y be the value of the next higher order bit. Then the 2 x 2 contingency table is shown in Fig. 4.1. The corresponding statistic, f^2 (mean square contingency), is given by the following equation:

$$f^2 = \frac{(n_{11}n_{22} - n_{12}n_{21})^2}{n_{1\cdot}n_{2\cdot}n_{\cdot 1}n_{\cdot 2}} \qquad (1)$$

The test statistic, nf^2, is computed separately for each of the three noise generator sections using an n of 10,000 bit samples. Bits 12 - 17 of the noise word correspond to the first shift register section, bits 6 - 11 correspond to the second, and bits 0 - 5 correspond to the third. Nf^2 is then compared to χ_1^2 (chi-squared distribution with one degree of freedom), and if nf^2 is less than χ_1^2 at the required level of significance, then the hypothesis of statistical independence is accepted. Table 4.1 contains five runs of the computer program from which it is seen that the largest nf^2 is approximately 2.61. Comparing this

y \ x	0	1	
0	n_{11}	n_{12}	$n_{1.}$
1	n_{21}	n_{22}	$n_{2.}$
	$n_{.1}$	$n_{.2}$	n

Fig. 4.1 Contingency Table

Table 4.1

Five Sets of Computed Test Statistics

1. GEN 1 STATISTIC; NF21= Ø.925Ø2
   GEN 2 STATISTIC; NF22= Ø.95113
   GEN 3 STATISTIC; NF23= Ø.19181

2. GEN 1 STATISTIC; NF21= 1.6382Ø
   GEN 2 STATISTIC; NF22= Ø.242Ø5
   GEN 3 STATISTIC; NF23= 2.6Ø998

3. GEN 1 STATISTIC; NF21= Ø.15Ø54
   GEN 2 STATISTIC; NF22= 1.13151
   GEN 3 STATISTIC; NF23= Ø.19321

4. GEN 1 STATISTIC; NF21= 1.Ø3784
   GEN 2 STATISTIC; NF22= Ø.719Ø4
   GEN 3 STATISTIC; NF23= 2.43494

5. GEN 1 STATISTIC; NF21= Ø.19622
   GEN 2 STATISTIC; NF22= Ø.ØØ963
   GEN 3 STATISTIC; NF23= Ø.Ø6441

statistic to a χ_1^2 table (Korn 1966), the hypothesis of statistical independence would be accepted at 10% significance level.

Appendix II contains a listing of the PDP-9 MACRO-9 statistic gathering program, STATIN, and three FORTRAN-4 subroutines, C1OUT, C2OUT and C3OUT, which compute and print the test statistic, nf^2, for each of the three noise generator sections. All four programs should be loaded together prior to execution.

4.2 Adjacent-Bit Correlation Test

Adjacent bits accumulated in the noise-generator shift registers should be statistically independent and thus uncorrelated if the noise generator is functioning properly. A program that computes the sample correlation coefficient of adjacent bits for each of three shift register sections was written to test the noise generator. The sample correlation coefficient, $\bar{r}$, is given by:

$$\bar{r} = \frac{\overline{x_1 x_2}}{\sqrt{\overline{x_1}^2 \, \overline{x_2}^2}}$$

where

$$\bar{x}_1 = \frac{1}{n} \sum_{i=1}^{n} x_{1i} \qquad \text{and} \qquad \bar{x}_1^{\,2} = \frac{1}{n} \sum_{i=1}^{n} x_{1i}^2$$

For a sample size of n, let n_c be number of coincidences of adjacent bits and then $n-n_c$ = number of anti-coincidences of adjacent bits. Then, for binary noise

$$\bar{r} = \frac{\frac{n_c}{n} - \left(\frac{n-n_c}{n} \right)}{1} = 2 \frac{n_c}{n} - 1$$

$\bar{r}$ is computed for each of the three noise generator sections using a sample size of 10,000 bits. Bits 12 - 17 of the noise word correspond to the first shift register section, bits 6 - 11 correspond to the second, and bits 0 - 5 correspond to the third.

$\bar{r}$ gives some qualitative measure of the noise generator performance although it is not used in a hypothesis test. Figure 4.2 is a graph of $\bar{r}$ versus a variable sampling period, T, for one of the three noise generator sections. From this data it is seen that T should certainly be set greater than 2 μ sec., or the noise bits will be significantly correlated. This program was also used to set the noise-diode current which is adjustable on the noise source circuit card. The diode current is adjusted, while running the program, until minimum $\bar{r}$ is obtained.

Appendix III contains a listing of the PDP-9 MACRO-9 statistic gathering program, BITCOR, and a FORTRAN-4 subroutine, CMPOUT, which computes and prints $\bar{r}$ for each of the three noise generator sections.

4.3 A Correlation Display Test

A computer installation using a noise source needs a simple operating test on the noise generator to determine whether the generator is operational before proceeding with more elaborate tests (if required), or with its use as input to a program. With an X-Y cathode-ray tube display in the computer system, a noise display scheme was tried in an attempt to give the operator a quick visual indication of the noise generator operation. The program reads 1024 words from the noise generator into a buffer area in memory. Then one of the three

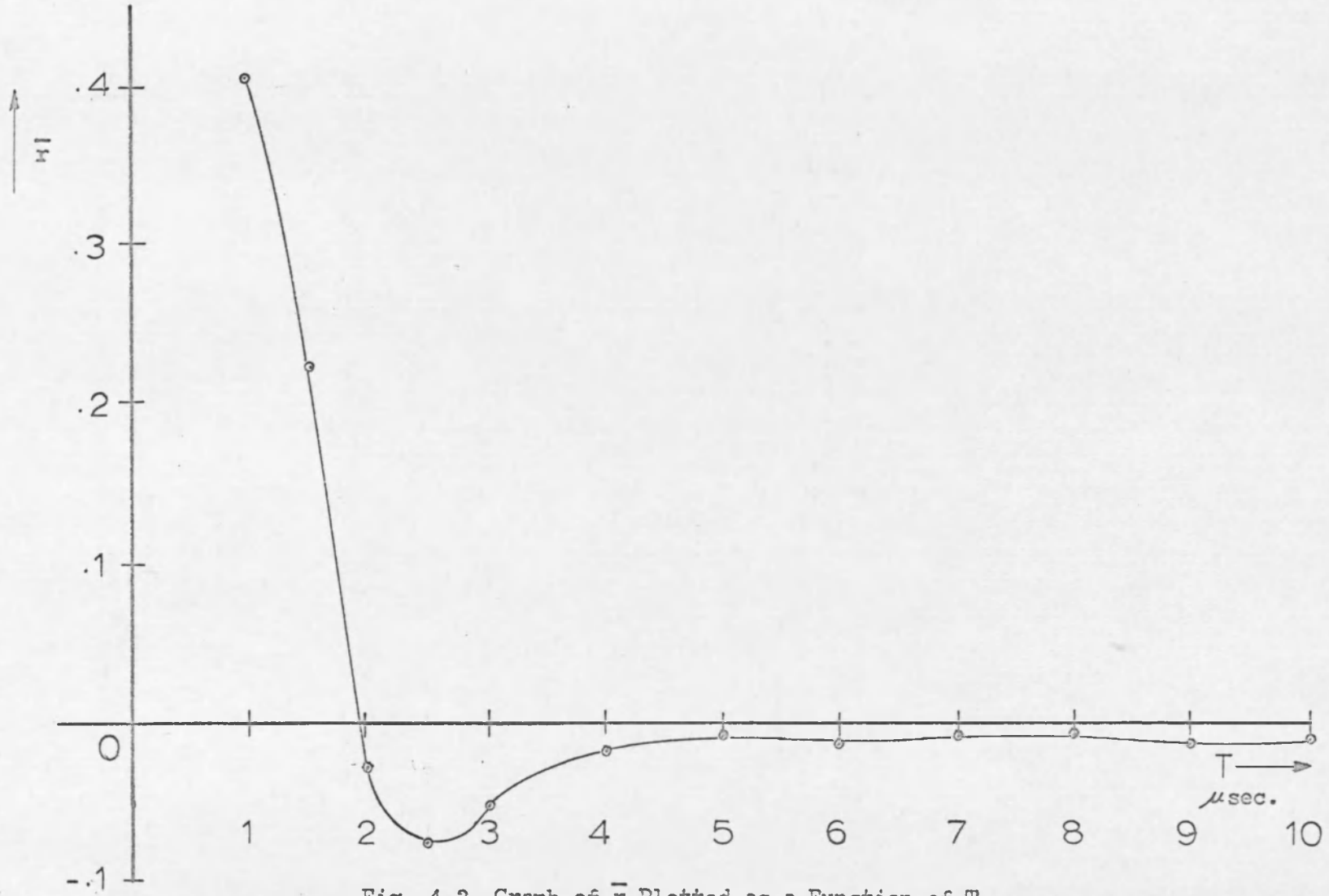

Fig. 4.2 Graph of $\bar{r}$ Plotted as a Function of T

6 bit sections is used as the X-coordinate and another selected 6 bit section is used as the Y-coordinate of the cathode-ray tube display. Thus 1024 dots are displayed on the screen. Provisions are made to provide arbitrary bit and word delays between X selected and Y selected noise generator sections by using the accumulator switches on the PDP-9 console. This selected delay corresponds to the $(t_2 - t_1)$ delay between the X and Y samples. Switches 15 - 17 select noise generator section (1-3) to be used as X-coordinate. Switches 12 - 14 select noise generator section (1-3) to be used as Y-coordinate. Switches 9 - 11 select bit delay (0-6) between X and Y data. Switches 0 - 8 select word delays (0 - 512 words of 6 bits each) between X and Y data. A purely random pattern on the screen would indicate no correlation and any clustering of dots indicates correlation. Figures 4.3 and 4.4 show photographs of the display information. Figure 4.3a shows an apparently uncorrelated pattern; namely, section 2 as X-coordinate, section 3 as Y-coordinate with a 0 bit delay. Figure 4.3b shows a completely correlated pattern; namely, section 2 as both X and Y-coordinate with 0 bit delay. Figure 4.4a shows a correlated display with only 1 bit in common; section 2 is used for X and Y-coordinate with 5 bits delay. The correlation can still be observed. Figure 4.4b shows a correlated display resulting from a noise diode not biased properly to produce sufficiently fast noise for the 5 μ sec. sampling period. Note the clustering of display dots. The $\bar{r}$ computed for this case is 0.345.

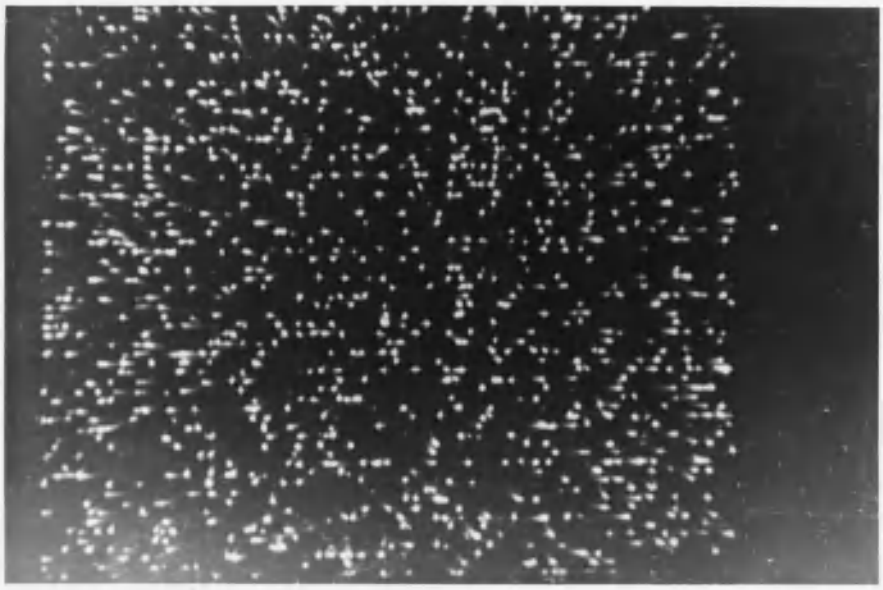

Fig. 4.3a Photograph of Display with Generator 1 versus Generator 2 with 0 Bit Delay

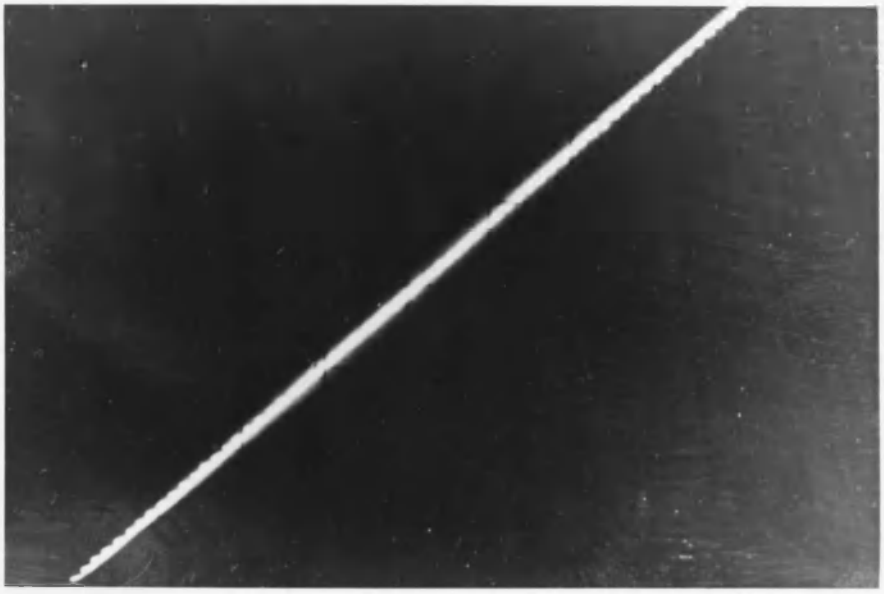

Fig. 4.3b Photograph of Display with Generator 2 versus Generator 2 with 0 Bit Delay

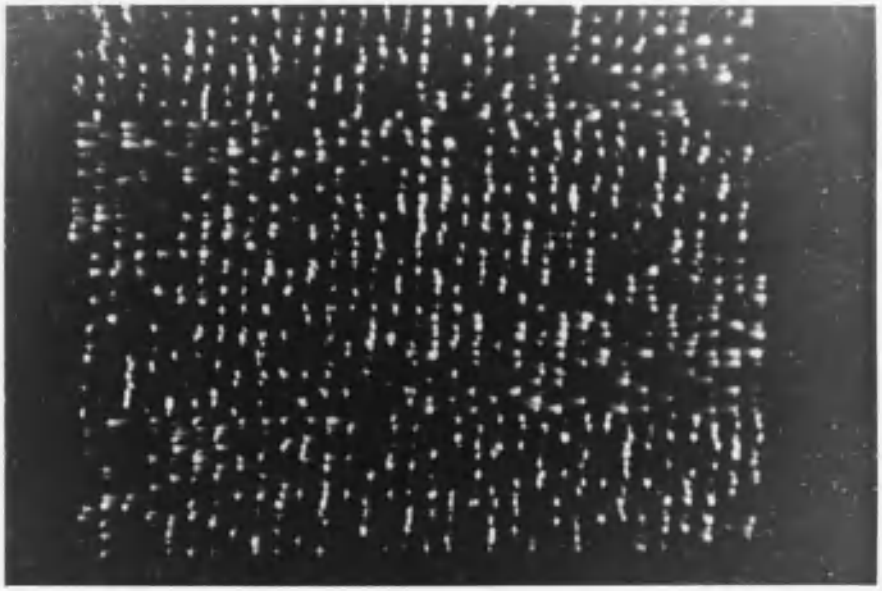

Fig. 4.4a Photograph of Display with Generator 2 versus Generator 2 with 5 Bit Delay

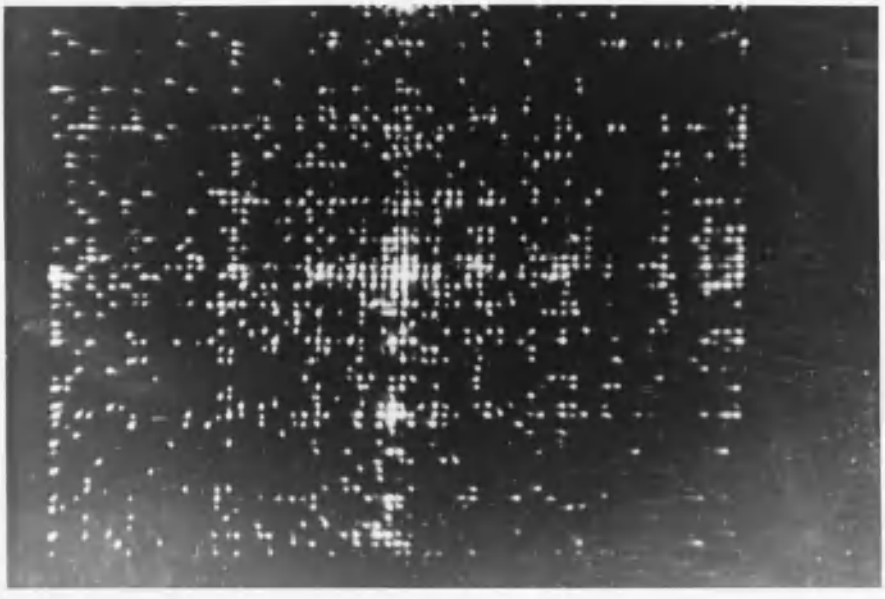

Fig. 4.4b Photograph of Display with Generator 1 versus Generator 1 with 1 Word Delay. Noise diode is not properly biased.

This test does not give quantitative information, but it is surprising how good a human observation is at determining problems with the use of a visual display. Appendix IV contains a listing of the PDP-9 MACRO-9 Correlation Display Program, CORDIS.

4.4 User Results

R. McClellan and T. Liebert (McClellan and Liebert 1969) developed a computer program which measures the mutual information content of two sections of the noise generator. They measured a very low mutual information of approximately 3×10^{-3} bits. This is an indication of statistical independence of different 6 bit sections in the noise generator.

R. C. White, Jr., developed a computer program which is a chi-squared goodness-of-fit test for a uniform distribution. If adjacent bits are statistically independent, then the noise generator should produce a uniform distribution of binary words. The chi-squared test results of R. C. White, Jr., give no evidence to disprove the hypothesis that the noise generator output is not uniformly distributed at a confidence level of 95% using a sample size of 2,000 words.

Gaussian noise is easily generated from the uniformly distributed noise by using the output of the generator as a table-lookup address of a Gaussian distribution (Korn and Aus 1969). This requires about 50 μ sec. per Gaussian word. Interpolation may or may not be required depending upon the application.

The noise generator has been successfully used as a random number source for two different creeping-random-search optimization programs (Gonzales 1969) and (White 1969).

4.5 Conclusions

The noise generator design presented here performs well as a random noise generator for the Analog/Hybrid Computer Laboratory. The generator produces a uniform distribution of binary words where the adjacent bits are essentially statistically independent and uncorrelated. Expanding laboratory applications and requirements may dictate the need for a faster noise generator. In this case, either a higher bandwidth noise source must be developed to permit faster sampling rates or the noise generator will have to be re-designed to include a separate noise source for each bit in the generated noise word.

APPENDIX I

MECL TO DEC LOGIC LEVEL CONVERSION CIRCUIT

WORST-CASE DESIGN ANALYSIS

Design Problem

The noise generator shift registers use MECL logic elements with nominal logic voltages of -0.75 v. and -1.55 v. The noise generator must work with the PDP-9 I/O interface, where DEC series R logic circuits are used. These series R circuits use nominal logic voltages of 0 v. and -3.0 v. Thus, a logic-level conversion circuit is required to convert from MECL to DEC logic levels.

The general circuit design objectives were: 1.) simplicity; 2.) reliability; 3.) operating speed compatible with DEC logic to assure proper I/O interface operation; and, 4.) to use standard DEC logic voltages only to minimize power supply requirements.

Design

A differential-amplifier input scheme (Fig. A.1) was chosen for two reasons. First, the input voltage swing is so small (nominally 800 mv.) that it precludes the selection of a grounded emitter switch. Second, both the true and complement logic signals that are needed by the differential amplifier are available from the MECL flip-flops in the noise generator shift register.

The circuit output must provide a logic signal with nominal voltage levels of -3.0 v. and 0 v. In the circuit of Fig. A.1, the

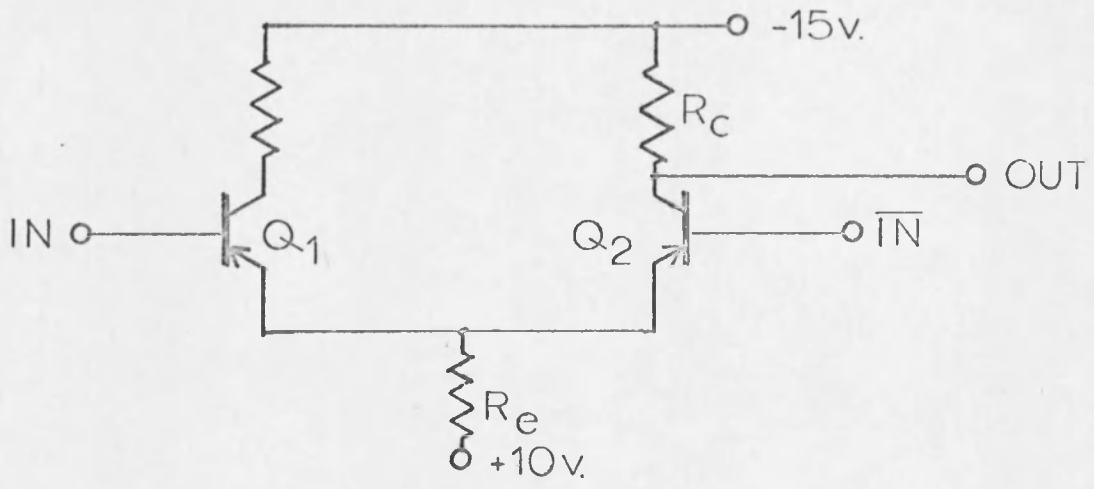

Fig. A.1 Differential-Amplifier Circuit

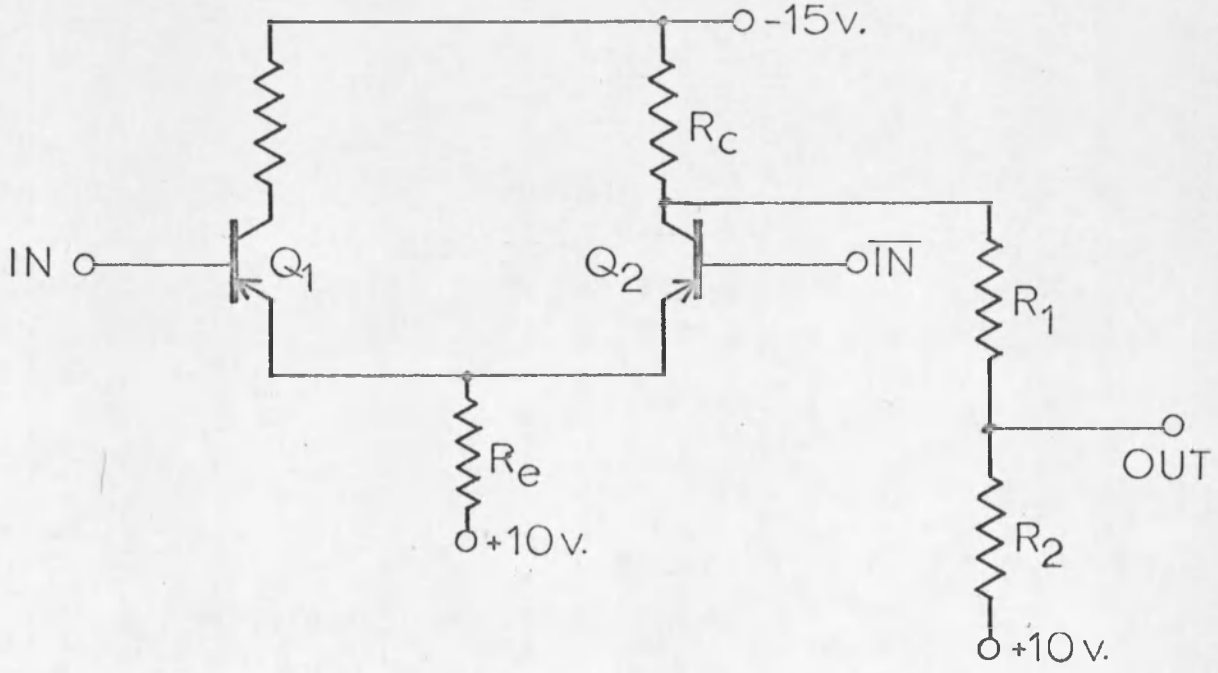

Fig. A.2 Differential-Amplifier Circuit with Voltage Divider

-3.0 v. is easily obtained with Q_2 OFF, since the collector resistor, R_c, is returned to -15 v. However, obtaining 0 v. out when Q_2 is ON is not possible since the base of Q_2 is at -1.55 v. and even with Q_2 saturated, its collector cannot reach 0 v. To eliminate this problem in the simplest way, a voltage divider was added to the output (Fig. A.2).

This simple approach is not generally good, since the divider represents fractional current gain, thus reducing fan-out capability of the circuit. However, in this case, the output drives only a single logic load, and the current requirements of that load are quite small, namely 0 mA at -3.0 v. (diode leakage current) and 1 mA at 0 v., nominally. Also, the noise generator is physically compact, so that the stray capacitance that has to be driven by the output is small.

Worst-Case Design

The following notation is used for a worst-case d-c design analysis of the circuit proposed in Fig. A.2.

$\overline{R}$ indicates maximum value of variable R.

$\underline{R}$ indicates minimum value of variable R.

For example, if R is a 510 ohms 5% resistor, then $\overline{R}$ = 535 ohms and $\underline{R}$ = 485 ohms.

From the Digital Equipment Corporation Small Computer Handbook, (1967c), the -15 v. supply can have worst-case values of -14.5 to -16.5 v. and the +10 v. supply has output of +11.5 to +9.5 v. The use of 5% resistors was then assumed.

With Q_2 OFF, the worst-case equivalent circuit, relative to the constraint that V_{out} is less than - 3.0 v., is shown in Fig. A.3. This circuit assumes that Q_2 base-emitter junction is reverse biased and that Q_2 is a silicon transistor so that leakage current in collector is negligible compared to divider currents. A constraint equation is then developed as follows:

$$i = \frac{11.5 + 14.5 \text{ v.}}{\bar{R}_1 + \bar{R}_c + \underline{R}_2} \qquad (1)$$

$$V_{out} = 11.5 \text{ v.} - i\,\underline{R}_2 \leq 3 \text{ v.} \qquad (2)$$

Combining equations (1) and (2) gives

$$11.5\,\underline{R}_2 \geq 14.5\,(\bar{R}_1 + \bar{R}_c) \qquad (3)$$

With Q_2 ON, the worst-case equivalent circuit relative to the constraint that $V_{out} \geq 0$ v. while $i_{out} = 1$ mA is shown in Fig. A.4, where V_c is the collector voltage of Q_2 with Q_2 ON.

A second constraint equation is then developed from the equivalent circuit of Fig. A.4.

$$9.5 - i_2\bar{R}_2 - (i_2 - 1)\,\underline{R}_1 - \underline{V}_c = 0 \qquad (4)$$

$$-V_{out} = -9.5 - i_2\bar{R}_2 \geq 0 \qquad (5)$$

Combining equation (4) with the constraint inequality (5) yields:

$$\bar{R}_2 \leq \frac{9.5R_1}{\underline{R}_1 - \underline{V}_c} \qquad (6)$$

At this point, two inequalities, (3) and (6) have been developed that contain four variables, so the design will proceed by successive trials while considering limitations due to transistor specifications.

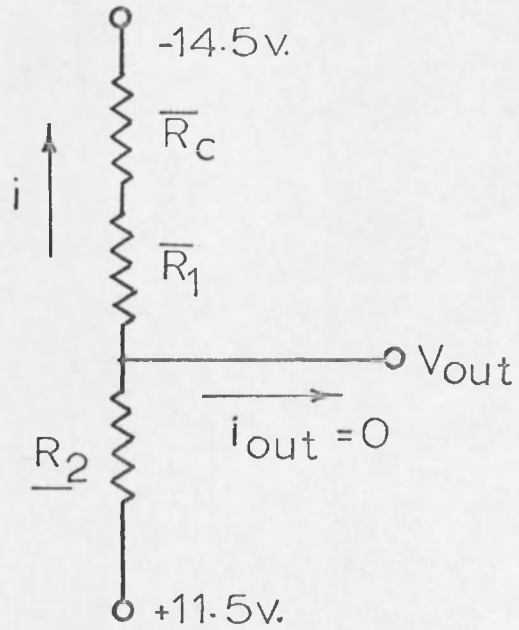

Fig. A.3 Equivalent Circuit with Q_2 OFF

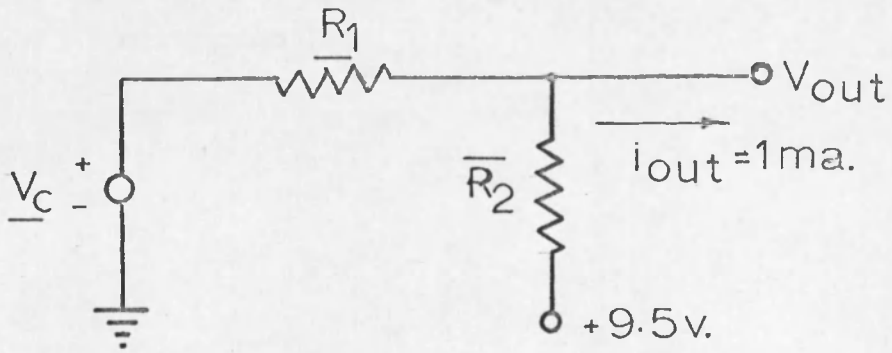

Fig. A.4 Equivalent Circuit with Q_2 ON

In Fig. A.5, graphs of equations (3) and (6) have been plotted with R_2 as a function of R_1 with R_c and V_c as parameters. Nominal values of R_1 and R_2 must be chosen so that rectangular area representing possible values of R_1 and R_2 due to total tolerances lies below curve of equation (6) and above curve of equation (3). Based on this criterion, the following values were chosen:

1) $R_c = 0.51\ K \Rightarrow \underline{R}_c = .48\ K;\ \overline{R}_c = .54\ K$

2) $V_c \cong -2\ v.$

3) $R_1 = 1.5\ K \Rightarrow \underline{R}_1 = 1.58\ K$

4) $R_2 = 3.3\ K \Rightarrow \underline{R}_2 = 3.13\ K;\ \overline{R}_2 = 3.47\ K$

Next, the emitter resistor, R_E, must be selected based on the above circuit values and then the transistor specifications must be checked against the trial design. Figure A.6 is an equivalent circuit with Q_2 ON from which $\underline{i}_c$, the collector current, can be calculated.

$$9.5 - i_2\overline{R}_2 - (i_2 - 1)\,\underline{R}_1 - (\underline{i}_c + i_2 - 1)\,\underline{R}_c + 16.5 = 0 \qquad (7)$$

$$\underline{V}_c - (\underline{i}_c + i_2 - 1)\,\underline{R}_c + 16.5 = 0 \qquad (8)$$

Solving simultaneous equations (7) and (8) with selected values from above yields:

$$i_2 = \frac{9.5 - V_c + \underline{R}_1}{\overline{R}_2 + \underline{R}_1} = \frac{9.5 + 2 + 1.42}{3.47 + 1.42} = 2.64\ mA$$

now solving (8) for $\underline{i}_c$ yields:

$$i_c = \frac{V_c + 16.5 - (i_2 - 1)\,\underline{R}_c}{\underline{R}_c} = \frac{-2 + 16.5 - (1.64)(.48)}{0.48} = 28.6\ mA$$

thus $i_c \geq 28.6\ mA$ (9)

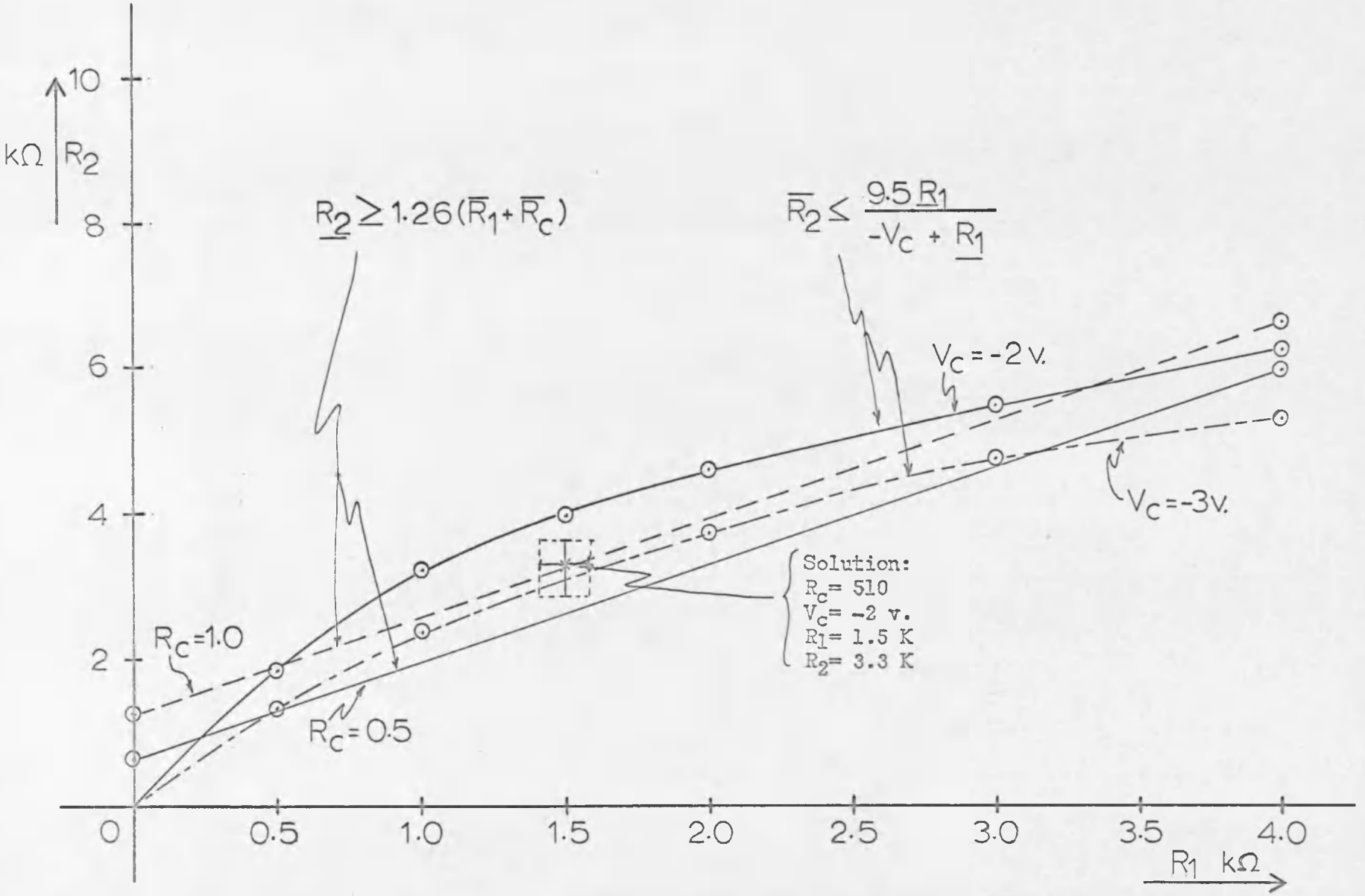

Fig. A.5 Graphs of Circuit Equations Showing Solution Space

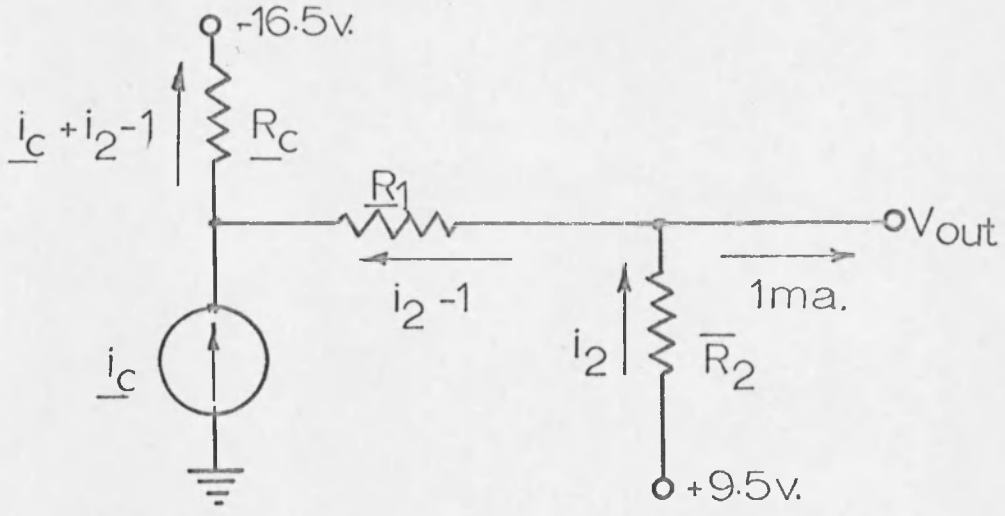

Fig. A.6 Equivalent Circuit with Q_2 ON

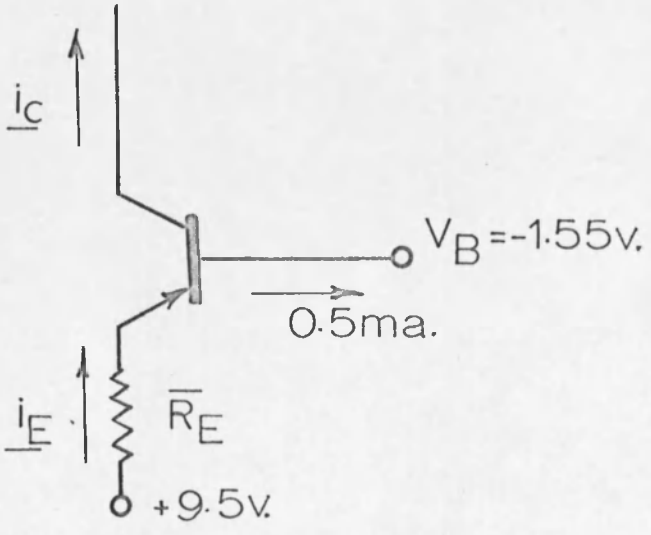

Fig. A.7 Circuit for Computing Emitter Resistor Value

Assuming base-emitter drop of Q_2 ON is approximately 0.5 v. and that minimum $h_{FE} \approx 60$, then $\underline{i}_E \approx \underline{i}_c + 0.5$ mA or $\underline{i}_c \approx 29.1$ mA and using equivalent circuit of Fig. A.7:

$$9.5 - \underline{i}_E \bar{R}_E - 0.50 + 1.55 = 0$$

or

$$\bar{R}_E = \frac{10.5}{\underline{i}_E} \approx \frac{10.5}{29.1} \approx .361 \text{ K ohms}$$

Thus R_E = 340 ohms was picked.

The transistor selected for this circuit is the Motorola MPS 6522 silicon PNP. It was selected because of its availability, low cost, and because its specifications satisfy the circuit requirements, namely:

Collector Emitter Breakdown Voltage (I_c = -0.5 mA d-c, I_B = 0)	BV_{CEO} -25 v. min.
D-C Current Gain (I_c = -2mA d-c, V_{CE} = -10 v.)	h_{FE} 200 min.
(I_c = -100mA d-c, V_{CE} = -10 v.)	90 min.
Current Gain Bandwidth Product (I_c = -10mA d-c, V_{CE} = -10 v.)	f_T 420 MH_z typ
Device dissipation @ T_A = 60°C	P_D 210 mW.

Transistor power dissipation estimation is $P_W \approx i_c V_{CE}$ $P_W \approx (28.8\text{mA})(-2 \text{ v.} + 1 \text{ v.}) \approx 30$ mW. which is well below 210 mW capability @ 60° C. Since duty cycle is low (maximum 200 K_c) the average dissipation due to switching should not cause any problem.

No transient analysis was made on the circuit because the speed requirements were not severe and because the selected transistor was quite fast. The circuit was constructed and found to function correctly with a rise time plus delay time of less than 50 nanosec.

APPENDIX II

PDP-9 PROGRAM LISTINGS

STATIN, C1OUT, C2OUT, C3OUT

Contingency Table Test Programs

This appendix contains computer listings of the MACRO-9 program, STATIN, and the three FORTRAN-4 programs, C1OUT, C2OUT and C3OUT, which are used to compute the contingency table test statistic, nf^2. The statistic is computed on a sample size of 10,000 bits for each of the three noise generator sections. All four programs are loaded prior to execution.

```
        /J.E.BELT 3-27-69
        /PROGRAM TO TEST STATISTICAL
        /INDEPENDENCE OF ADJACENT
        /BITS (CONTINGENCY TABLE TEST)
        .TITLE STATIN
        .GLOBL C1OUT
        .GLOBL C2OUT
        .GLOBL C3OUT
START   LAC (N110-1     /CLEAR STATISTICS
        DAC* (10        /ACCUMULATORS
        .DEC
        LAC (-18
        .OCT
        DAC STRCTR#
        CLA
        DAC* 10
        ISZ STRCTR
        JMP .-2
        .DEC
        LAC (-2000
        DAC WDCTR#
        .OCT
REWD    LAC (-5
        DAC SHFTCR#
        706412          /READ NOISE WORD
        CLL
        RAR
RECYC1  SZL             /COMPUTE GEN1 STAT
        JMP X1EQ1
        RAR             /X1 IS ZERO
        SZL
        JMP OX0Y1
        DAC NTEMP#
        LAC N110        /X1 IS 0,Y1 IS 0
        ADD (1
        DAC N110
        LAC NTEMP
BEGN2   ISZ SHFTCR
        JMP RECYC1
        DAC NTEMP
        LAC (-5
        DAC SHFTCR
        LAC NTEMP
        RAR             /COMPUTE GEN2 STAT
RECYC2  SZL
        JMP X2EQ1
        RAR             /X2 IS 0
        SZL
```

```
        JMP  TXØY1
        DAC  NTEMP
        LAC  N11T       /X2 IS Ø,Y2 IS Ø
        ADD  (1
        DAC  N11T
        LAC  NTEMP
BEGN3   ISZ  SHFTCR
        JMP  RECYC2
        DAC  NTEMP
        LAC  (-5
        DAC  SHFTCR
        LAC  NTEMP
        RAR
RECYC3  SZL             /COMPUTE GEN3 STAT
        JMP  X3EQ1
        RAR             /X3 IS Ø
        SZL
        JMP  DXØY1
        DAC  NTEMP
        LAC  N11D       /X3 IS Ø,Y3 IS Ø
        ADD  (1
        DAC  N11D
        LAC  NTEMP
BEGN4   ISZ  SHFTCR
        JMP  RECYC3
        ISZ  WDCTR
        JMP  REWD
        LAC  N110       /COMPUTE MARGINAL
        ADD  N120       /STATISTICS FOR GEN
        DAC  N1.0       /NO. 1
        LAC  N110
        ADD  N210
        DAC  N.10
        LAC  N11T       /COMPUTE MARGINAL
        ADD  N12T       /STATISTICS FOR GEN
        DAC  N1.T       /NO. 2
        LAC  N11T
        ADD  N21T
        DAC  N.1T
        LAC  N11D       /COMPUTE MARGINAL
        ADD  N12D       /STATISTICS FOR GEN
        DAC  N1.D       /NO. 3
        LAC  N11D
        ADD  N21D
        DAC  N.1D
        JMS* C1OUT      /PRINT SUBROUTINE
        JMP  .+7
        .DSA N110
```

```
        .DSA  N120
        .DSA  N210
        .DSA  N220
        .DSA  N.10
        .DSA  N1.0
        JMS*  C20UT
        JMP   .+7
        .DSA  N11T
        .DSA  N12T
        .DSA  N21T
        .DSA  N22T
        .DSA  N.1T
        .DSA  N1.T
        JMS*  C30UT
        JMP   .+7
        .DSA  N11D
        .DSA  N12D
        .DSA  N21D
        .DSA  N22D
        .DSA  N.1D
        .DSA  N1.D
        HLT
        JMP   START
0X0Y1   DAC   NTEMP
        LAC   N210        /X1 IS 0, Y1 IS 1
        ADD   (1
        DAC   N210
        LAC   NTEMP
        JMP   BEGN2
X1EQ1   RAR
        SZL
        JMP   0X1Y1
        DAC   NTEMP
        LAC   N120        /X1 IS 1,Y1 IS 0
        ADD   (1
        DAC   N120
        LAC   NTEMP
        JMP   BEGN2
0X1Y1   DAC   NTEMP
        LAC   N220        /X1 IS 1,Y1 IS 1
        ADD   (1
        DAC   N220
        LAC   NTEMP
        JMP   BEGN2
TX0Y1   DAC   NTEMP
        LAC   N21T        /X2 IS 0,Y2 IS 1
        ADD   (1
        DAC   N21T
```

```
        LAC  NTEMP
        JMP  BEGN3
X2EQ1   RAR
        SZL
        JMP  TX1Y1
        DAC  NTEMP
        LAC  N12T       /X2 IS 1,Y2 IS Ø
        ADD  (1
        DAC  N12T
        LAC  NTEMP
        JMP  BEGN3
TX1Y1   DAC  NTEMP
        LAC  N22T       /X2 IS 1,Y2 IS 1
        ADD  (1
        DAC  N22T
        LAC  NTEMP
        JMP  BEGN3
DXØY1   DAC  NTEMP
        LAC  N21D       /X3 IS Ø,Y3 IS 1
        ADD  (1
        DAC  N21D
        LAC  NTEMP
        JMP  BEGN4
X3EQ1   RAR
        SZL
        JMP  DX1Y1
        DAC  NTEMP
        LAC  N12D       /X3 IS 1,Y3 IS Ø
        ADD  (1
        DAC  N12D
        LAC  NTEMP
        JMP  BEGN4
DX1Y1   DAC  NTEMP
        LAC  N22D       /X3 IS 1,Y3 IS 1
        ADD  (1
        DAC  N22D
        LAC  NTEMP
        JMP  BEGN4
N110
N120
N210
N220
N.10
N1.0
N11T
N12T
N21T
N22T
```

STATIN


```
N.1T
N1.T
N11D
N12D
N21D
N22D
N.1D
N1.D
        .END START
```

```
C       SUBROUTINE FOR COMPUTING
C       NFSQ FOR CONTINGENCY TABLE
C       TEST FOR STATIN AND PRINTING
C       RESULTS.  J.E.BELT 3-27-69
        SUBROUTINE C1OUT(N110,N120,
      1 N210,N220,NP10,N1P0)
        X110=N110
        X120=N120
        X210=N210
        X220=N220
        XP10=NP10
        X1P0=N1P0
        XNF1SQ=(10000.*((X110*X220-X120
      1 *X210)**2))/(X1P0*(10000.-X1P0)
      1 *XP10*(10000.-XP10))
        WRITE (5,10) XNF1SQ
10      FORMAT (* GEN 1 STATISTIC;NF21=
      1 *F8.5)
        RETURN
        END
```

```
C       SUBROUTINE FOR COMPUTING
C       NFSQ FOR CONTINGENCY TABLE
C       TEST FOR STATIN AND PRINTING
C       RESULTS.  J.E.BELT 3-27-69
        SUBROUTINE C2OUT(N11T,N12T,
     1  N21T,N22T,NP1T,N1PT)
        X11T=N11T
        X12T=N12T
        X21T=N21T
        X22T=N22T
        XP1T=NP1T
        X1PT=N1PT
        XNF2SQ=(10000.*((X11T*X22T-X12T
     1  *X21T)**2))/(X1PT*(10000.-X1PT)
     1  *XP1T*(10000.-XP1T))
        WRITE (5,11) XNF2SQ
11      FORMAT (* GEN 2 STATISTIC; NF22=
     1  *F8.5)
        RETURN
        END
```

```
C       SUBROUTINE FOR COMPUTING
C       NFSQ FOR CONTINGENCY TABLE
C       TEST FOR STATIN AND PRINTING
C       RESULTS.  J.E.BELT 3-27-69
        SUBROUTINE C3OUT(N11D,N12D,
     1  N21D,N22D,NP1D,N1PD)
        X11D=N11D
        X12D=N12D
        X21D=N21D
        X22D=N22D
        XP1D=NP1D
        X1PD=N1PD
        XNF3SQ=(10000.*((X11D*X22D-X12D
     1  *X21D)**2))/(X1PD*(10000.-X1PD)
     1  *XP1D*(10000.-XP1D))
        WRITE (5,12) XNF3SQ
12      FORMAT (* GEN 3 STATISTIC; NF23=
     1  *F8.5)
        RETURN
        END
```

APPENDIX III

PDP-9 PROGRAM LISTINGS

BITCOR & CMPOUT

Correlation Coefficient Test Programs

This appendix contains computer listings of the MACRO-9 program, BIRCOR, and the FORTRAN-4 subroutine, CMPOUT, which are used to compute the sample correlation coefficient, $\bar{r}$. This test statistic is computed on a sample size of 10,000 bits for each of the three noise generator sections. Both programs are loaded prior to execution.

```
/J. E. BELT  3-10-69
/BIT CORRELATION PROGRAM
        .TITLE BITCOR
        .GLOBL CMPOUT
START   DZM NCOR1#
        DZM NCOR2#
        DZM NCOR3#
        .DEC
        LAC (-2000
        DAC WDCTR#
        .OCT
REWD    LAC (-5
        DAC SHFTCR#
        706412
        CLL
        RAR
RECYC1  SZL
        JMP BIT11
        RAR
        SZL
        JMP NOMAT1
MATCH1  DAC NTEMP#
        LAC NCOR1
        ADD (1
        DAC NCOR1
        LAC NTEMP
NOMAT1  ISZ SHFTCR
        JMP RECYC1
        DAC NTEMP
        LAC (-5
        DAC SHFTCR      /START OF GEN 2
        LAC NTEMP
        RAR
RECYC2  SZL
        JMP BIT12
        RAR
        SZL
        JMP NOMAT2
MATCH2  DAC NTEMP
        LAC NCOR2
        ADD (1
        DAC NCOR2
        LAC NTEMP
NOMAT2  ISZ SHFTCR
        JMP RECYC2
        DAC NTEMP
        LAC (-5
        DAC SHFTCR
```

```
        LAC NTEMP
        RAR
RECYC3  SZL
        JMP BIT13
        RAR
        SZL
        JMP NOMAT3
MATCH3  DAC NTEMP
        LAC NCOR3
        ADD (1
        DAC NCOR3
        LAC NTEMP
NOMAT3  ISZ SHFTCR
        JMP RECYC3
        ISZ WDCTR
        JMP REWD
        JMS* CMPOUT    /COMPUTE PRINT SUBR
        JMP .+4
        .DSA NCOR1
        .DSA NCOR2
        .DSA NCOR3
        HLT
        JMP START
BIT11   RAR
        SNL
        JMP NOMAT1
        JMP MATCH1
BIT12   RAR
        SNL
        JMP NOMAT2
        JMP MATCH2
BIT13   RAR
        SNL
        JMP NOMAT3
        JMP MATCH3
        .END START
```

```
C       JEB 3-9-69
C       SUBROUTINE FOR COMPUTING
C       CORRELATION COEFFICIENTS
C       AND PRINTING RESULTS
        SUBROUTINE CMPOUT(NCOR1,
     1  NCOR2,NCOR3)
        XCOR1=NCOR1
        XCOR2=NCOR2
        XCOR3=NCOR3
        R1=((2.*XCOR1)/10000.)-1.
        R2=((2.*XCOR2)/10000.)-1.
        R3=((2.*XCOR3)/10000.)-1.
        WRITE (5,10) R1
10      FORMAT (* GENERATOR 1 BIT
     2  CORRELATION COEFFICIENT*F6.3)
        WRITE (5,11) R2
11      FORMAT (* GENERATOR 2 BIT
     3  CORRELATION COEFFICIENT*F6.3)
        WRITE (5,12) R3
12      FORMAT (* GENERATOR 3 BIT
     4  CORRELATION COEFFICIENT*F6.3)
        RETURN
        END
```

APPENDIX IV

PDP-9 PROGRAM LISTING

CORDIS

Correlation Display Test Program

This appendix contains a computer listing of the MACRO-9 program, CORDIS, which is used to display the outputs of two of the noise generator sections as the X- and Y-coordinate values on the X-Y cathode-ray tube display. The accumulator switches on the PDP-9 console are used to control the operation of this program. Switches 15-17 select the noise generator section, 1-3, to be used as the X-coordinate, and switches 12-14 select the noise generator section, 1-3, to be used as the Y-coordinate. Switches 9-11 select the bit delay, 0-6, between the X and Y data, while switches 0-8 select the word delays, 0-512 words of 6 bits each, between the X and Y data.

```
      .TITLE      CORRELATION DISPLAY
        /OF RANDOM NOISE GENERATOR
        /J.E.BELT      2-10-69
        .DEC
START   LAC (1
        DAC STFLG#
        JMP INPAR       /SET WDTAU,TAU,&
REPT    LAC (-1536      /CCFLG FROM SWT'S
        DAC WDCTR#
        .OCT
        LAC (N1ADR-1    /SET AUTO INDEX
        DAC* (10        /REG FOR NOISE WD
RECYC   706412          /DISTRIBUTION
        DAC* 10
        LAC (-6
        DAC DELCTR#
        ISZ DELCTR
        JMP .-1
        ISZ WDCTR
        JMP RECYC
        LAC (N1ADR-1
        DAC* (10
        ADD WDTAU#
        DAC* (11
        DAC* (12
        LAC* 12
        .DEC
        LAC (-1024
        .OCT
        DAC WDCTR
        LAC (DABADR+2
        DAC* (13
NEWORD  CLL
        CLQ
        LAC CCFLG#
        AND (000030
        LRS 3
        TAD (-1
        SNA
        JMP EXT1
        TAD (-1
        SNA
        JMP EXT2
        JMP EXT3
EXT1    LAC* 10
        AND (000077
        DAC TEMP1#
ENTR1   TAD (-400
```

```
        SNA
        JMP .+5
        LAC TEMP1
        LLSS 14
        DAC NTEMP#
        JMP CONVRG
        LAC TEMP1
        ADD (1
        LLSS 14
        DAC NTEMP
        JMP CONVRG
EXT2    LAC* 10
        AND (007700
        LRSS 6
        DAC TEMP1
        JMP ENTR1
EXT3    LAC* 10
        AND (770000
        LRSS 14
        DAC TEMP1
        JMP ENTR1
CONVRG  CLQ
        CLL
        LAC CCFLG
        AND (000003
        TAD (-1
        SNA
        JMP EXTY1
        TAD (-1
        SNA
        JMP EXTY2
        JMP EXTY3
EXTY1   LAC* 11
        AND (000077
        DAC TEMP1
        LAC* 12
        AND (000077
        DAC TEMP2#
ENTRY1  LAC TAU#
        AND (000007
        ADD (LRSS
        DAC SHFT1
        LAC TEMP2
        LLSS 6
        ADD TEMP1
SHFT1   XX      /LRSS TAU
        AND (000077
        LLSS 3
```

```
        ADD NTEMP
        JMP YCNVRG
EXTY2   LAC* 11
        AND (007700
        LRSS 6
        DAC TEMP1
        LAC* 12
        AND (007700
        LRSS 6
        DAC TEMP2
        JMP ENTRY1
EXTY3   LAC* 11
        AND (770000
        CLL
        LRS 14
        DAC TEMP1
        LAC* 12
        AND (770000
        LRS 14
        DAC TEMP2
        JMP ENTRY1
YCNVRG  DAC* 13
        ISZ WDCTR       /WORD INTO BUFFER
        JMP NEWORD
        LAC DABADR+3
        DAC DABADR+1
        LAC STFLG
        SZA
        JMP SERVST
INPAR   CLA
        LAS
        DAC TEMP1
        AND (000033
        DAC CCFLG
        LAC TEMP1
        CLL
        LRS 6
        AND (000007
        DAC TAU
        LAC TEMP1
        LRS 11
        AND (000777
        DAC WDTAU
        JMP REPT
SERVST  DZM STFLG
        JMS DCH
        JMP INPAR
DCH
```

```
        CAL
        16
        700621
        INT
        DBK
        .DEC
DCHIN   LAC (-1027
        .OCT
        DAC* (36
        LAC (DABADR-1
        DAC* (37
        LAC (400
        700622          /START DATA CHANNEL
        LAC SAV#
        ION
        DBR
        JMP* DCH
INT     DAC SAV
        700501
        LAC* (0
        DAC DCH
        JMP DCHIN
DABADR 400000
        XX
        400200
        .DEC
        .BLOCK 1023
        .OCT
        400000
        .DEC
N1ADR   .BLOCK 1536
        .OCT
        .END START
```

REFERENCES

Conant, B. "The Design of a New Solid State Electronic Iterative Differential Analyzer Making Maximum Use of Integrated Circuits", Ph.D. Dissertation, Department of Electrical Engineering, University of Arizona, 1968.

Cramer, H. Mathematical Methods of Statistics, Princeton University Press, 1946.

Digital Equipment Corporation. Logic Handbook, 1967a.

Digital Equipment Corporation. PDP-9 Users Handbook, 1967b.

Digital Equipment Corporation. Small Computer Handbook, 1967c.

Gonzales, R. "An Optimization Study on a Hybrid Computer", M.S. Thesis, Department of Electrical Engineering, University of Arizona, 1969.

Handler, H. "High-Speed Monte Carlo Technique for Hybrid Computer Solution of Partial Differential Equations", Ph.D. Dissertation, Department of Electrical Engineering, University of Arizona, 1967.

Kohne, H., W. D. Little, and A. C. Soudak. "An Economical Multichannel Noise Source", Simulation, September, 1965.

Korn, G. A. Random-Process Simulation and Measurements, McGraw-Hill Book Company, New York, 1966.

Korn, G. A., and T. M. Korn. Mathematical Handbook for Scientists and Engineers, McGraw-Hill Book Company, New York, 2nd edition, 1968.

Korn, G. A., and H. M. Aus. "Table-Lookup/Interpolation Function Generation for Fixed Point Digital Computations", IEEE TEC, August, 1969.

McClellan, R., and T. Liebert. "Estimation of Entropy and Mutual Information Using the PDP-9", Term Project Report, Department of Electrical Engineering, University of Arizona, January, 1969.

Motorola Corporation. "The MECL Line of Digital Integrated Circuits", 1966.

Papoulis, A. Probability, Random Variables and Stochastic Processes, McGraw-Hill Book Company, New York, 1965.

Shreider, Y. A., et al. Methods of Statistical Testing, Elsevier Publishing Company, New York, 1964.

White, R. C. Jr. "Experiments with Digital Computer Simulations of Pseudo-Random Noise Generators", IEEE TEC, June, 1967.

_____. "Hybrid Computer Optimization of Systems with Random Parameters", Ph.D. Dissertation, Department of Electrical Engineering, University of Arizona, 1969.