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 Bult

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

G. A. Korn
Professor of Electrical Engineering

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

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

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Noise Generator Block Diagram</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Noise Source Block Diagram</td>
<td>9</td>
</tr>
<tr>
<td>2.2 Typical Noise Source Waveforms</td>
<td>9</td>
</tr>
<tr>
<td>2.3 Noise Source Circuit Diagram</td>
<td>10</td>
</tr>
<tr>
<td>3.1 MECL to DEC Logic Level Conversion Circuit</td>
<td>14</td>
</tr>
<tr>
<td>3.2 MECL Power Supply Circuit Diagram</td>
<td>14</td>
</tr>
<tr>
<td>3.3 Noise Generator Printed Circuit Board Locations</td>
<td>16</td>
</tr>
<tr>
<td>3.4 Logic Diagram of Noise Generator Shift Registers</td>
<td>17</td>
</tr>
<tr>
<td>3.5 I/O Interface Logic Diagram</td>
<td>18</td>
</tr>
<tr>
<td>4.1 Contingency Table</td>
<td>20</td>
</tr>
<tr>
<td>4.2 Graph of ( r ) Plotted as a Function of ( T )</td>
<td>24</td>
</tr>
<tr>
<td>4.3a Photograph of Display with Generator 1 versus Generator 2 with 0 Bit Delay</td>
<td>26</td>
</tr>
<tr>
<td>4.3b Photograph of Display with Generator 2 versus Generator 2 with 0 Bit Delay</td>
<td>26</td>
</tr>
<tr>
<td>4.4a Photograph of Display with Generator 2 versus Generator 2 with 5 Bit Delay</td>
<td>27</td>
</tr>
<tr>
<td>4.4b Photograph of Display with Generator 1 versus Generator 1 with 1 Word Delay. Noise diode is not properly biased.</td>
<td>27</td>
</tr>
<tr>
<td>A.1 Differential-Amplifier Circuit</td>
<td>31</td>
</tr>
<tr>
<td>A.2 Differential-Amplifier Circuit with Voltage Divider</td>
<td>31</td>
</tr>
<tr>
<td>A.3 Equivalent Circuit with Q2 OFF</td>
<td>34</td>
</tr>
</tbody>
</table>
List of Illustrations (Continued)

A.4 Equivalent Circuit with Q₂ ON .............................................. 34
A.5 Graphs of Circuit Equations Showing Solution Space .................. 36
A.6 Equivalent Circuit with Q₂ ON .............................................. 37
A.7 Circuit for Computing Emitter Resistor Value ........................... 37
<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>4</td>
</tr>
<tr>
<td>4.1</td>
<td>21</td>
</tr>
</tbody>
</table>

**Table of Noise Generator Specifications**

**Five Sets of Computed Test Statistics**
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.
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>
<thead>
<tr>
<th><strong>General Requirements</strong></th>
<th>Solid-state devices only; Plug-compatible with DEC PDP-9 I/O Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Word Rate</strong></td>
<td>Statistically independent, uniformly distributed words available every 30 – 40 $\mu$ sec.</td>
</tr>
<tr>
<td><strong>Word Length</strong></td>
<td>18 bits</td>
</tr>
</tbody>
</table>
| **Temperature**          | $70^\circ$ – $85^\circ$ F. $\{ \begin{array}{ll} 
\text{Typical computer} \\
\text{operating environment}
\end{array} \}$ |
| **Humidity**             | $30\%$ – $80\%$                                              |
| **Power Supply Voltages**| $-15$ v. (+0.5 v., -1.5 v.) $\begin{array}{ll}
+10$ v. (+1.5 v., -0.5 v.) \\
DEC standard voltages and tolerances
\end{array}$ |
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.
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 (\(< 1/10\)) 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
A stable Multi-Vibrator

Clock System Located on the MECL Power Supply Card

Noise Source Circuit; 3 on Card

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 (Q₁), which then drives an emitter follower circuit (Q₂). 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, R₁ and R₂, 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 because 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 Q2 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. 3.4 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 $2 \times 2$ contingency table is shown in Fig. 4.1. The corresponding statistic, $f^2$ (mean square contingency), is given by the following equation:

$$f^2 = \frac{(n_{11}n_{22} - n_{12}n_{21})^2}{n_1n_2n_1n_2}$$

(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 $\chi^2_1$ (chi-squared distribution with one degree of freedom), and if $nf^2$ is less than $\chi^2_1$ 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
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$n_{11}$</td>
<td>$n_{12}$</td>
<td>$n_1$</td>
</tr>
<tr>
<td>1</td>
<td>$n_{21}$</td>
<td>$n_{22}$</td>
<td>$n_2$</td>
</tr>
<tr>
<td></td>
<td>$n_{.1}$</td>
<td>$n_{.2}$</td>
<td>$n$</td>
</tr>
</tbody>
</table>

Fig. 4.1 Contingency Table
Table 4.1

Five Sets of Computed Test Statistics

<table>
<thead>
<tr>
<th></th>
<th>GEN 1 STATISTIC; NF21=0.92502</th>
<th>GEN 2 STATISTIC; NF22=0.95113</th>
<th>GEN 3 STATISTIC; NF23=0.19161</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GEN 1 STATISTIC; NF21=1.63820</td>
<td>GEN 2 STATISTIC; NF22=0.24205</td>
<td>GEN 3 STATISTIC; NF23=2.6998</td>
</tr>
<tr>
<td>2</td>
<td>GEN 1 STATISTIC; NF21=0.15054</td>
<td>GEN 2 STATISTIC; NF22=1.13151</td>
<td>GEN 3 STATISTIC; NF23=0.19321</td>
</tr>
<tr>
<td>3</td>
<td>GEN 1 STATISTIC; NF21=0.19622</td>
<td>GEN 2 STATISTIC; NF22=0.00963</td>
<td>GEN 3 STATISTIC; NF23=0.06441</td>
</tr>
<tr>
<td>4</td>
<td>GEN 1 STATISTIC; NF21=0.19622</td>
<td>GEN 2 STATISTIC; NF22=0.00963</td>
<td>GEN 3 STATISTIC; NF23=0.06441</td>
</tr>
</tbody>
</table>
statistic to a $\chi^2$ 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, $n_f^2$, 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:

$$\bar{r} = \frac{x_1x_2}{\sqrt{x_1^2x_2^2}}$$

where

$$\bar{x}_1 = \frac{1}{n} \sum_{i=1}^{n} x_{1i} \quad \text{and} \quad \bar{x}_2 = \frac{1}{n} \sum_{i=1}^{n} x_{1i}^2$$

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

$$\bar{r} = \frac{n_c - (n-n_c)}{n} = 2 \frac{n_c}{n} - 1$$
\( \bar{r} \) 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 \( \mu \) 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 \(\mu\) sec. sampling
period. Note the clustering of display dots. The \(\bar{F}\) 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, COEDIS.

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 $3 \times 10^{-3}$ 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 $50 \, \mu$ 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 (nominal 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 Q2 OFF, since the collector resistor, 
Rc, is returned to -15 v. However, obtaining 0 v. out when Q2 is ON 
is not possible since the base of Q2 is at -1.55 v. and even with Q2 
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., nomi-
inally. 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:

\begin{equation}
 i = \frac{11.5 + 14.5 \text{ v.}}{R_1 + R_c + R_2} \tag{1}
\end{equation}

\begin{equation}
 V_{out} = 11.5 \text{ v.} - i R_2 \leq 3 \text{ v.} \tag{2}
\end{equation}

Combining equations (1) and (2) gives

\begin{equation}
 11.5 R_2 \geq 14.5 (R_1 + R_c) \tag{3}
\end{equation}

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.

\begin{equation}
 9.5 - i_2 R_2 - (i_2 - 1) R_1 - V_c = 0 \tag{4}
\end{equation}

\begin{equation}
 V_{out} = 9.5 - i_2 R_2 \geq 0 \tag{5}
\end{equation}

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

\begin{equation}
 R_2 \leq \frac{9.5 R_1}{R_1 - V_c} \tag{6}
\end{equation}

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) \( R_C = 0.51 \ \text{K} \Rightarrow R_C = 0.48 \ \text{K}, \quad R_C = 0.54 \ \text{K} \)
2) \( V_C = -2 \ \text{v} \)
3) \( R_1 = 1.5 \ \text{K} \Rightarrow R_1 = 1.58 \ \text{K} \)
4) \( R_2 = 3.3 \ \text{K} \Rightarrow R_2 = 3.13 \ \text{K}, \quad R_2 = 3.47 \ \text{K} \)

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 \( i_C \), the collector current, can be calculated.

\[
9.5 - i_2 R_2 - (i_2 - 1) R_1 - (i_C + i_2 - 1) R_C + 16.5 = 0 \quad (7)
\]
\[
V_C - (i_C + i_2 - 1) R_C + 16.5 = 0 \quad (8)
\]

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

\[
i_2 = \frac{9.5 - V_C + R_1}{R_2 + R_1} = \frac{9.5 + 2 + 1.42}{3.47 + 1.42} = 2.64 \ \text{mA}
\]

now solving (8) for \( i_C \) yields:

\[
i_C = \frac{V_C + 16.5 - (i_2 - 1) R_C}{R_C} = \frac{-2 + 16.5 - (1.64)(.48)}{0.48} = 28.6 \ \text{mA}
\]

thus \( i_C \geq 28.6 \ \text{mA} \)
Solution:
R_C = 510
V_C = -2 V.
R_I = 1.5 K
R_2 = 3.3 K

\[ \bar{R}_2 \geq 1.26(\bar{R}_I + \bar{R}_C) \]
\[ \bar{R}_2 \leq \frac{9.5 \bar{R}_I}{-V_C + R_I} \]

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 Q2 ON is approximately 0.5 v. and
that minimum h\textsubscript{FE} \approx 60, then \( i_{E} \approx i_{C} + 0.5 \text{ mA} \) or \( i_{C} \approx 29.1 \text{ mA} \) and using
equivalent circuit of Fig. A.7:

\[
9.5 - \frac{i_{E}R_{E}}{i_{E}} - 0.50 + 1.55 = 0
\]

or

\[
R_{E} = \frac{10.5}{i_{E}} \approx \frac{10.5}{29.1} \approx 0.361 \text{ K ohms}
\]

Thus \( R_{E} = 340 \text{ ohms} \) 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 \( BV_{CEO} \geq 25 \text{ v. min.} \) \( (I_{C} = -0.5 \text{ mA d-c, } I_{B} = 0) \)
- D-C Current Gain
  \( (I_{C} = -2 \text{mA d-c, } V_{CE} = -10 \text{ v.}) \) \( h_{FE} \geq 200 \text{ min.} \)
  \( (I_{C} = -100 \text{mA d-c, } V_{CE} = -10 \text{ v.}) \) \( 90 \text{ min.} \)
- Current Gain Bandwidth Product
  \( (I_{C} = -10 \text{mA d-c, } V_{CE} = -10 \text{ v.}) \) \( f_{T} \geq 420 \text{ MHz typ} \)
- Device dissipation @ \( T_{A} = 60^\circ \text{C} \) \( P_{D} \geq 210 \text{ mW} \)

Transistor power dissipation estimation is \( P_{W} \approx i_{C} V_{CE} \)

\( P_{W} \approx (28.8 \text{mA})(-2 \text{ v.} + 1 \text{ v.}) \approx 30 \text{ mW} \). which is well below 210 mW capa-
bility @ 60\textdegree C. Since duty cycle is low (maximum 200 K\textsubscript{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, CIOUT, C2OUT, C3OUT

Contingency Table Test Programs

This appendix contains computer listings of the MACRO-9 program, STATIN, and the three FORTRAN-4 programs, CIOUT, C2OUT and C3OUT, which are used to compute the contingency table test statistic, $n^2f^2$. 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 OXØY1
DAC NTEMP#
LAC NI10 /X1 IS 0,Y1 IS 0
ADD (1
DAC NI10
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 TX0Y1
DAC NTEMP
LAC N11T /X2 IS 0,Y2 IS 0
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 0
SZL
JMP DX0Y1
DAC NTEMP
LAC N11D /X3 IS 0,Y3 IS 0
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
S T A T I N ' 4 3

® D S A N I . 2 0

. D S A . N 1 2 . 0
. D S A . N 2 1 . 0
. D S A . N 2 2 . 0
. D S A . N . 1 . 0
. D S A . N 1 - . 0
J M S * C 2 O U T
J M P + 7

. D S A . N 1 1 . T
. D S A . N 1 2 . T
. D S A . N 2 1 . T
. D S A . N 2 2 . T
. D S A . N . 1 . T
. D S A . N 1 . T
J M S * C 3 O U T
J M P + 7

. D S A . N 1 1 . D
. D S A . N 1 2 . D
. D S A . N 2 1 . D
. D S A . N 2 2 . D
. D S A . N . 1 . D
. D S A . N 1 . D
H L T

O X 0 Y 1
D A C N T E M P
L A C N 2 1 0
/X 1 I S 0 , Y 1 I S 1
A D D 1
D A C N 2 1 0
L A C N T E M P
J M P B E G N 2

X I E Q 1
R A R
S Z L
J M P O X 1 Y 1
D A C N T E M P
L A C N 1 2 0
/X 1 I S 1 , Y 1 I S 0
A D D 1
D A C N 1 2 0
L A C N T E M P
J M P B E G N 2

O X 1 Y 1
D A C N T E M P
L A C N 2 2 0
/X 1 I S 1 , Y 1 I S 1
A D D 1
D A C N 2 2 0
L A C N T E M P
J M P B E G N 2

T X 0 Y 1
D A C N T E M P
L A C N 2 1 T
/X 2 I S 0 , Y 2 I S 1
A D D 1
D A C N 2 1 T
LAC NTEMP
JMP BEGN3

X2 EQ 1
RAR
SZL
JMP TX1 Y1
DAC NTEMP
LAC N12 T /X2 IS 1,Y2 IS 0
ADD (1
DAC N12 T
LAC NTEMP
JMP BEGN3

TX1 Y1
DAC NTEMP
LAC N22 T /X2 IS 1,Y2 IS 1
ADD (1
DAC N22 T
LAC NTEMP
JMP BEGN3

DX0 Y1
DAC NTEMP
LAC N21 D /X3 IS 0,Y3 IS 1
ADD (1
DAC N21 D
LAC NTEMP
JMP BEGN4

X3 EQ 1
RAR
SZL
JMP DX1 Y1
DAC NTEMP
LAC N12 D /X3 IS 1,Y3 IS 0
ADD (1
DAC N12 D
LAC NTEMP
JMP BEGN4

DX1 Y1
DAC NTEMP
LAC N22 D /X3 IS 1,Y3 IS 1
ADD (1
DAC N22 D
LAC NTEMP
JMP BEGN4

N1.10
N1.20
N2.10
N2.20
N.1.10
N1.0
N1.1T
N1.2T
N2.1T
N2.2T
STATIN

N.1T
N1.1T
N1 1D
N1 2D
N2 1D
N2 2D
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, NIPO)
X110=N110
X120=N120
X210=N210
X220=N220
XP10=NP10
X1P0=NIPO
XNFISQ=(10000.*((X110*X220-X120
1 *X210)**2))/(X1P0*(10000.-X1P0)
1 *XP10*(10000.-XP10))
WRITE (5,10) XNFISQ
10 FORMAT (* GEN I STATISTIC; NF21=
1 *FB.5)
RETURN
END
SUBROUTINE FOR COMPUTING
NFSQ FOR CONTINGENCY TABLE
TEST FOR STATIN AND PRINTING
RESULTS. J. E. BELL 3-27-69
SUBROUTINE C2OUT(N11T, N12T, N21T, N22T, NP1T, N1PT)

X11T = N11T
X12T = N12T
X21T = N21T
X22T = N22T
XP1T = NP1T
X1PT = N1PT

XNF2SQ = (10000.0 * (X11T * X22T - X12T * X21T)**2) / (X1PT * (10000.0 - X1PT) * XP1T * (10000.0 - XP1T))

WRITE (5, 11) XNF2SQ

FORMAT (* GEN 2 STATISTIC; NF22 = *F8.5)

RETURN
END
SUBROUTINE FOR COMPUTING
SUBROUTINE FOR CONTINGENCY TABLE
SUBROUTINE FOR STATIN AND PRINTING
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=(10000.0*(X11D*X22D-X12D
1 *X21D)**2)/(X1PD*(10000.0-X1PD)
1 *XP1D*(10000.0-XP1D))
WRITE (5,12) XNF3SQ
12 FORMAT (* GEN 3 STATISTIC; N F3 =
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, BITCOR, and the FORTRAN-4 subroutine, CMPOUT, which are used to compute the sample correlation coefficient, $r$. 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
BITCOR

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 1+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
SUBROUTINE CMPOUT(NCOR1, NCOR2, NCOR3)
  XCOR1 = NCOR1
  XCOR2 = NCOR2
  XCOR3 = NCOR3
  R1 = ((2.0 * XCOR1) / 10000.) - 1.
  R2 = ((2.0 * XCOR2) / 10000.) - 1.
  R3 = ((2.0 * XCOR3) / 10000.) - 1.
  WRITE (5, 10) R1
  WRITE (5, 11) R2
  WRITE (5, 12) R3
RETURN
END
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

.START
LAC (1
DAC STFLG#
JMP INPAR /SET WDATA,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 WDATA#
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 EXIT1
TAD (-1
SNA
JMP EXIT2
JMP EXIT3

.EXIT1
LAC* 10
AND (000077
DAC TEMP1#

.ENTRY
TAD (-400
CORDIS

SNA
JMP *+5
LAC TEMP1
LLSS 14
DAC NTMP#
JMP CONVRG
LAC TEMP1
ADD (1
LLSS 14
DAC NTMP
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 (00007
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
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


