CR 73469

AVAILABLE TO THE PUBLIC

STATISTICAL PROPERTIES OF WEIGHTED RANDOM AND PSEUDO-RANDOM SEQUENCES

N70-36008 (ACCESSION NUMBER) (THRU)
98 (PAGES) 1 (CODE)
CR 73469 (NASA CR OR TMX OR AD NUMBER) 29 (CATEGORY)
FACILITY FORM 602

by

AUG 1970
RECEIVED
NASA STI FACILITY
INPUT BRANCH

Timothy J. Healy, Assistant Professor
Randolph L. Cramer, Research Assistant
Pierre Loisel, Research Assistant

Final Report Submitted to the Ames Research Center
National Aeronautics and Space Administration
NASA-Ames Agreement T1-C/USC Inst.

June 1, 1970

The University of Santa Clara • California

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
Springfield, Va. 22151

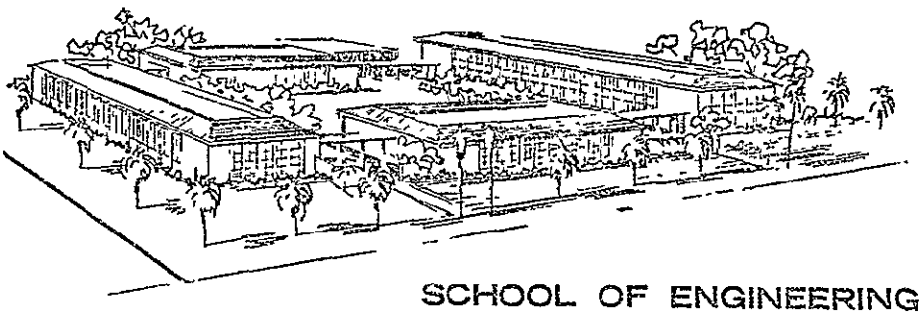
STATISTICAL PROPERTIES OF WEIGHTED RANDOM AND PSEUDO-RANDOM SEQUENCES

by

Timothy J. Healy, Assistant Professor
Randolph L. Cramer, Research Assistant
Pierre Loisel, Research Assistant

Final Report Submitted to the Ames Research Center
National Aeronautics and Space Administration
NASA-Ames Agreement T1-C/USC Inst.

June 1, 1970



SCHOOL OF ENGINEERING
ENGINEERING AND APPLIED SCIENCE RESEARCH
UNIVERSITY OF SANTA CLARA
SANTA CLARA, CALIFORNIA AREA CODE 408-246-3200

TABLE OF CONTENTS

		Page
0.0	Introduction	
	0.1 Areas of Application	3
1.0	Forms of Probability Density Functions	5
	1.1 The Probability Density Function	5
	1.2 Moments	10
2.0	Autocorrelation and Spectral Properties	12
	2.1 Discrete Autocorrelation of Pseudo-Random Processes	12
	2.2 Convolution Approach	16
	2.3 Synthesis of Autocorrelation Functions	21
	2.4 Continuous Autocorrelation Functions	22
	2.5 Appendix	27
3.0	The Probability Distributions of Certain Sums of Random Variables	32
	3.1 The Uniform Cases	33
	3.2 A Non-uniform Case	36
	3.3 Conclusion	41
	3.4 Appendix	42
4.0	Pseudo-Random Noise Generation and Digital Filter Implementation	45
	4.1 Shift Register	45
	4.2 Pseudo-Random Sequence Generator	47
	4.3 Digital Filter	50
	4.4 A Hardware Realization of the Random Sequence Generator	59
	4.5 A Hardware Realization of a Nonrecursive Digital Filter	63
	4.6 Experimental Measurements on the Linearity and Frequency Response of the Digital Filter	63
	4.7 Design and Experimental Results of a Lowpass Digital Filter	67
	4.8 Software Simulation of a Lowpass Digital Filter	78
5.0	Generation of Partition Numbers	86
	5.1 Partition Numbers and Convolution	86
	5.2 Digital Filter Generation	89
	5.3 Generation of Other Sequences	89

BIBLIOGRAPHY

Statistical Properties of Weighted Binary
Random and Pseudo-Random Sequences

0.0 Introduction

The kind of problem discussed in this final report is illustrated in figure 0-1:

An n-stage binary shift register accepts the input binary random or pseudo-random sequence. The sequence is shifted one stage to the right at each clocking time. The values in the stages of the register are multiplied or weighted by the numbers $\{a_1, a_2, \cdots a_i \cdots, a_n\}$. The resulting products are summed to provide the output from the system. If the input is a pseudo-random sequence the shift register may also serve as the sequence generator by connecting it as a feedback shift register.

This problem was discussed in detail in an earlier report [1], which includes an extensive bibliography. The present report presents a number of extensions of [1].

First, the input is assumed to be pseudo-random, and the weights are 0 or 1. We consider the possible forms of the probability density function of the output, and some of its moments.

Second, the input is assumed to be either random or pseudo-random, and we consider the autocorrelation function and the power spectral density of the output.

Third, we assume that $n \to \infty$, and consider some fundamental properties of infinite sums of random variables.

Fourth, the present status and initial experiments in a new communications laboratory are discussed. It is shown how this laboratory will be used to study many of the concepts considered above. The design and testing of a

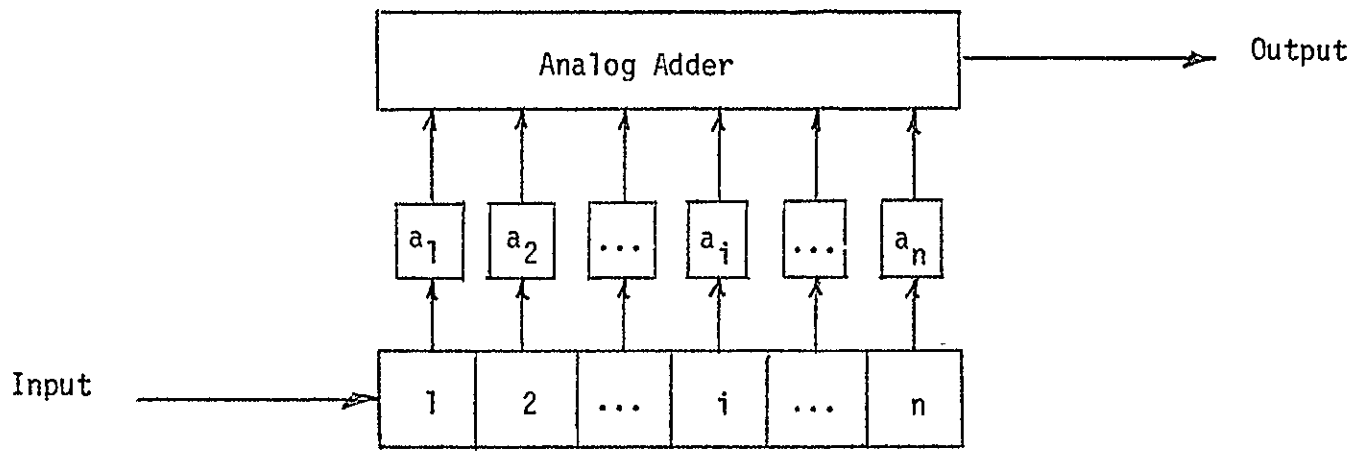


Figure 0-1. Basic Circuit Being Studied

binary nonrecursive digital filter is discussed in detail.

Fifth, some unexpected relations to some important basic mathematical relations are obtained. It is shown that such sequences as successive partition numbers, and the Fibonacci sequence, can be generated by digital filter circuits.

0.1 Applications

The problems studied here have a wide range of applications. First, we are concerned with the generation of random number sequences with different statistical properties. Such sequences are used in communications systems, system identification, equipment testing, and potentially in a great many other areas where random signals are required. These applications will continue to grow in number and importance as the trend towards digitalization of networks and systems continues.

Second, we are concerned with the effect of digital filters on random sequences. In some sense this is related to the first problem. In this case, however, the emphasis is on filtering. Digital filtering is rapidly becoming a very important approach to filtering. The effect which such filters have on random sequences is of great importance. The techniques developed here can be used to design filter to have desired effects on sequence autocorrelation functions.

Third, we have done some hardware design and development work on binary nonrecursive filters, which will be useful in implementing such filters.

Finally, a number of the ideas and hardware which have evolved from this study have led directly to applications in education. Material emanating from our work is introduced in undergraduate courses in probability

theory, (AM 108) and communications theory (EE 141), as well as in graduate courses in the same areas. A paper [17] has recently been prepared describing many of the ideas discussed here in an educational context. In addition a pair of courses in digital filtering, one at the undergraduate and one at the graduate level, based largely on our contract work, are being prepared for the winter of 1971. Thus the stimulus to our academic program has been both clear and direct.

1.0 Forms of Probability Density Functions

Some recent papers [2] - [6] have discussed the problems of the sum of weighted digits in an n-stage feedback shift register connected to generate maximal length sequences. In particular the papers by Lindholm [2] and Davies [5] have been concerned with the sum over m unity-weighted stages. They consider the problem of the statistics of the pseudo-random variable

$$w_i = \sum_{k=0}^{m-1} a_{k-i} \tag{1-1}$$

where a_i is the ith value (0 or 1) of the m-sequence having period $L = 2^n - 1$. n is the number of stages in the shift register feedback loop. The primary interest of these two papers is in the first few moments of w_i. Davies also presents graphical results of the actual distributions of w_i for $n = 5$ and $1 \leq m \leq 31$.

In this study the sum (1-1) is considered over any m stages that need not necessarily be successive. The probability density functions (PDF) of w_i are obtained in a straight-forward manner for different combinations of m stages. These PDF's are compared and found to be consistent with the results of reference [5]. The problem is considered for a particular case but the approach is applicable to other cases.

1.1 The Probability Density Function

Consider the 5-stage feedback shift register with generating polynomial in the delay operator D of the form $D^5 \oplus D^2 = D^0$, where D^i stands for i units of delay and $\oplus$ stands for modulo-2 addition. A shift register with five stages in the feedback loop and 26 additional stages available for summation is

shown in Figure 1-1. We seek the distribution of the analog sum:

$$S_k = D^1 + D^2 + D^3 + D^4 + D^5 + D^k \ , \ 1 \leq k \leq 31 \qquad (1\text{-}2)$$

This is the sum of the first five stages plus any other one of 31 stages. (It is not of interest to consider more than 31 stages since the sequence has period 31). Consider first the case where k = 6 in equation (1-2).

$$S_6 = D^1 + D^2 + D^3 + D^4 + D^5 + D^6$$

$$= D^1 + D^2 + D^3 + D^4 + D^5 + (D^1 \oplus D^3) \qquad (1\text{-}3)$$

The second equality follows from the generating polynomial. ($D^5 \oplus D^2 = D^0$ implies $D^6 = D^1 \oplus D^3$)

We now apply a method introduced in reference [6]. Group the terms in equation (1-3) according to the order of the delay.

$$S_6 = [D^1 + D^3 + (D^1 \oplus D^3)\ ] + D^2 + D^4 + D^5 \qquad (1\text{-}4)$$

The last three terms and the bracketed term in (1-4) are independent if we assume the all-zero state in the five-stage feedback shift register for computational purposes. Hence their PDF's can be convolved to find the PDF of S_6 . The PDF of the bracketed term is found from the following table:

D^1	D^3	$D^1 \oplus D^3$	$D^1 + D^3 + (D^1 \oplus D^3)$
0	0	0	0
0	1	1	2
1	0	1	2
1	1	0	2

From the above table the PDF of the bracketed term can be written as the sequence

$$\{1/4 \quad 0 \quad 3/4\}$$

where the order of the probability terms refers to sum values of 0, 1 and 2 from left to right. Again assuming the all-zero state for computational

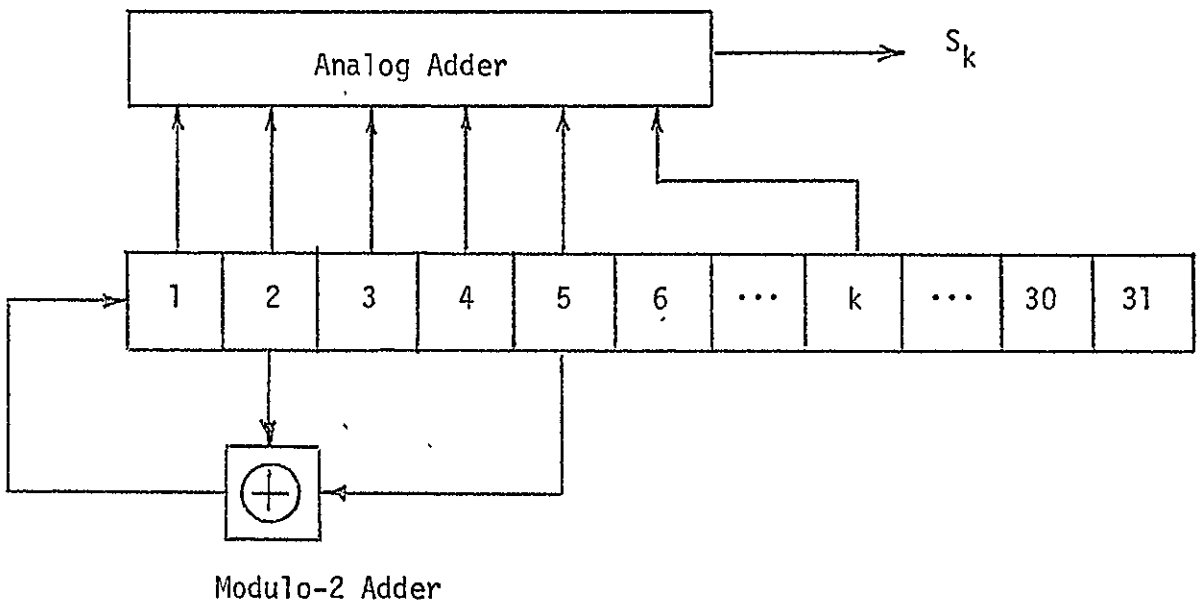


Figure 1-1. Sum of 6 Pseudo-Random Sequences

purposes the PDF for each of D^2, D^4 and D^5 is {1/2 1/2} :

The PDF of the sum S_6 is thus the convolution:

$$\{1/4 \quad 0 \quad 3/4\} * \{1/2 \quad 1/2\} * \{1/2 \quad 1/2\} * \{1/2 \quad 1/2\} =$$

$$1/32 \ \{ 1 \quad 3 \quad 6 \quad 10 \quad 9 \quad 3 \}$$

and, neglecting the all-zero state, this becomes:

$$1/31 \ \{ 0 \quad 3 \quad 6 \quad 10 \quad 9 \quad 3 \}$$

If we repeat this procedure for S_7 and S_8 we get the same PDF as for S_6 . If, however, we consider S_9 we obtain:

$$\begin{aligned} S_9 &= D^1 + D^2 + D^3 + D^4 + D^5 + D^9 \\ &= D^1 + D^2 + D^3 + D^4 + D^5 + D^6 \oplus 0^4 \\ &= [\ D^1 + D^3 + D^4 + (D^1 \oplus D^3 \oplus D^4)] + D^4 + D^5 \end{aligned} \tag{1-5}$$

with PDF

$$1/31 \ \{ 0 \quad 2 \quad 7 \quad 12 \quad 7 \quad 2 \quad 1 \}$$

The important difference between equations (1-4) and (1-5) is that in equation (1-4) there are two interdependent stages (1 and 3) whereas in equation (1-5) there are three interdependent stages (1, 3 and 4). We obtain a different PDF for different numbers of interdependent stages. The five possible types of PDF are given in Table 1-1. The first moment and the second through fifth central moment are also given.

There are a total of 31 ways to obtain all the PDF's. The number of ways of obtaining each type is simply the combination of the 5 basic stages taken r at a time where r is the number of interdependent stages. $r = 1$ corresponds to the case where $k = 1, 2, 3, 4$ or 5. In this case the weighting is essentially 2 rather than 1 for one of the 5 stages.

This basic approach can be applied to any combination of m stages. If $m = 6$ it is only necessary to group the terms appropriately and count the number of interdependent terms . The distribution is given in Table 1-1.

Number of Interdependent Stages	Ways to Obtain PDF	PDF (x 31) 0	1	2	3	4	5	6	m_1	μ_2	μ_3	μ_4	μ_5
1	5	0	4	7	8	7	4	1	3.097	1.764	0.360	6.956	4.423
2	10	0	3	6	10	9	3	0	3.097	1.248	-0.267	3.614	-1.638
3	10	0	2	7	12	7	2	1	3.097	1.248	0.508	4.862	5.427
4	5	0	1	10	10	5	5	0	3.097	1.248	0.508	3.314	2.305
5	1	0	0	15	0	15	0	1	3.097	1.248	0.508	3.314	6.176

$$m_1 = E(S_k)$$

$$\mu_r = E\{[S_k - E(S_k)]^r\}, \quad r = 2,3,4,5$$

TABLE 1-1

Also, of course, tables equivalentto 1-1 can be obtained for any n and m following the basic procedure outlined above.

1.2 Moments

It is shown in reference [5] that the first moment of a sum over m stages is:

$$E_1 = \frac{m\ (L + 1)}{2\ L} \tag{1-6}$$

which is called m_1 in Table 1-1.

E_1 is not dependent on which m stages are selected. Hence it is not surprising that, in agreement with (1-6), all the distributions in Table 1-1 are found to have $E_1 = 96/31 = 3.097$.

The variance, as obtained from equations (2) and (3) in reference [5] is:

$$\sigma^2 = \mu_2 = \frac{m(L+1)\ (L-m)}{4L^2} \tag{1-7}$$

This result is based on an assumption that the summed terms are from successive stages. For $m = 6$ and $L = 31$, equation (1-7) gives $\sigma^2 = 1.248$. As Table 1-1 indicates, this is in agreement with our results for all values of k except $k = 1, 2, 3, 4, 5$ (one interdependent stage). Hence the variance depends only on the number of stages summed, as long as they are separate stages, and not on which particular stages are selected. That is, they need not be successive. A study of the mathematics in reference [5], which leads to the expression for the second moment, indicates that this result is reasonable. The summations involved are essentially dependent only on the number of terms in the sums and not on the order of summation.

Similarly, the third moment μ_3 is the same for 3, 4, and 5 interdependent stages, and the fourth moment μ_4 is the same for 4 and 5 interdependent stages. In general, for this case, the rth moment (or central moment) is the same for all those cases where there are at least r interdependent stages. A number of other cases of n (for $4 \leq n \leq 10$) were partially checked; it was found that the rule in the preceding sentence holds for all cases investigated.

THe results indicated in this chapter were published by Healy [7]. This publication was responded to by Davies [8], who indicates an alternative approach to obtaining the PDF, through the use of transforms.

2.0 Autocorrelation and Spectral Properties

This section is concerned with the autocorrelation function of weighted and summed pseudo-random and random sequences. Consider the circuit shown in Figure 0-1. The outputs of the register stages are weighted by $(a_1, a_2, \cdots)$ and added to yield an output from the analog adder which is the pseudo-random or random sequence of interest here.

The PDF pseudo-random outputs has been studied in some recent papers ([2] - [6]). In this section we restrict our interest to the autocorrelation function and power spectral density of the output.

The autocorrelation function of a sequence can be considered from two viewpoints. First, it can be considered as discrete if we center our interest on an entire clocking period at a time. This is the viewpoint which is of interest, for example, to the computer user who is generating pseudo-random or random numbers. Alternatively, we may consider the clocking time to be unknown. This viewpoint is commonly of interest to communications engineers. We consider first the discrete viewpoint.

2.1 Discrete Autocorrelation of Pseudo-Random Processes

Let $a_1 = 1$ and $a_i = 0$ for all $i \neq 1$. Then the output is a binary sequence that we write as:

$$[b_1, b_2, b_3, \cdots, b_k \cdots, b_{2^n-1}, b_1, b_2, \cdots]$$

For this correlation study it is convenient to assume that the b_k take values +1 and -1. Since the sequence is periodic, we have $b_{k+2^n-1} = b_k$.

The discrete autocorrelation function is defined as:

$$R_1(m) = \frac{1}{2^n-1} \sum_{k=1}^{2^n-1} b_k b_{k+m}, \quad m = 0, 1, 2 \cdots \tag{2-1}$$

Consider first the pseudo-random case. It is well known [9] that the autocorrelation function of a maximal-length pseudo-random sequence is:

$$R_1(m) = \begin{cases} 1 & , m = 0 \\ \dfrac{1}{2^n-1} & , m \neq 0 \end{cases} \qquad (2\text{-}2)$$

for $|m| < 2^n-1$ and that $R_1(m)$ is periodic with period 2^n-1.

Consider now the autocorrelation of the output of the system shown in Figure 0-1. We assume the device is connected as a feedback shift register of length n stages for pseudo-random number generation, and there are L successive non-zero weights. That is, $a_i = 0$ for $i > 1$. Also assume $L \leq n$. Then an output value, which we might call the "k^{th}" output, is:

$$(a_L b_k + a_{L-1} b_{k+1} + \ldots + a_1 b_{k+L-1})$$

and a value m clock periods later is

$$(a_L b_{k+m} + a_{L-1}\, b_{k+1+m} + \ldots + a_1 b_{k+L-1+m}).$$

Hence, the autocorrelation function is:

$$R_L(m) = \frac{1}{2^n-1} \sum_{k=1}^{2^n-1} (a_L b_k + a_{L-1} b_{k+1} + \ldots + a_1 b_{k+L-1}) \cdot (a_L b_{k+m} + a_{L-1} b_{k+1+m} + \ldots + a_1 b_{k+L-1-m}) \qquad (2\text{-}3)$$

In forming the product indicated in (2-3), we collect terms having the same difference in the indices (subscripts) of b. Equation (2-3) thus becomes:

$$\begin{aligned} R_L(m) = \frac{1}{2^n-1} \sum_{k=1}^{2^n-1} (& a_L^2 b_k b_{k+m} + a_{L-1}^2\, b_{k+1} b_{k+1+m} + \ldots + a_1^2 b_{k+L-1}\, b_{k+L-12+m} \\ & + a_L a_{L-1} b_{k+1} b_{k+m} + \ldots + a_2 a_1 b_{k+L-1} b_{k+L-2+m} \\ & + \ldots . \\ & + a_L a_1 b_{k+L-1} b_{k+m}) \end{aligned} \qquad (2\text{-}4)$$

We assume the process is stationary with respect to clock intervals. A comparison of equations (2-1) and (2-4) then tells us that each term in the first row in (2-4) is proportional to $R_1(m)$, each term in the second row is proportional to $R_1(m-1)$, etc. Hence equation (2-4) reduces to:

$$\begin{aligned} R_L(m) = {} & (a_L^2 + a_{L-1}^2 + \dots + a_1^2)\, R_1(m) + \\ & (a_L a_{L-1} + a_{L-1} a_{L-2} + \dots + a_2 a_1)\, R_1(m-1) + \\ & \dots + (a_1 a_L)\, R_1(m-L+1) \\ = {} & \sum_{r=0}^{L-1} \sum_{i=1}^{L-r} a_i a_{i+r}\, R_1(m \pm r)\ , \quad r < L \end{aligned} \tag{2-5}$$

This relation holds for $|m| < 2^n-1$. For other values of m we need only note that $R(m)$ is periodic with period 2^n-1. This arises from the fact that the original pseudo-random sequence has this same period.

Although the relation (2-5) has been derived for a specific kind of input, it can be easily generalized for any input autorcorrelation $R_x(m)$ and output autocorrelation $R_y(m)$.

$$R_y(m) = \sum_{r=0}^{L-1} \sum_{i=1}^{L-r} a_i a_{i+r}\, R_x(m \pm r) \tag{2-6}$$

where L again stands for the number of weights.

Let us consider two examples. First, let $a_i=1$ for $1 \leq i \leq L$, $a_i=0$ for $i > L$. That is, we assume unit weights for all L stages. Equation (2-5) then becomes:

$$R_L(m) = \sum_{r=0}^{L-1} (L-r)\, R_1\ (m \pm r)\ , \ r < L \tag{2-7}$$

This result is plotted in Figure 2-1 for $n = 4$ and $L = 1$, 3 and 6.

As a second example let us consider binary weighting; that is, let $a_i = 2^{i-1}$ for $1 \leq i \leq L$ and $a_i = 0$ for $i > L$. (It was shown by Davies [3] and

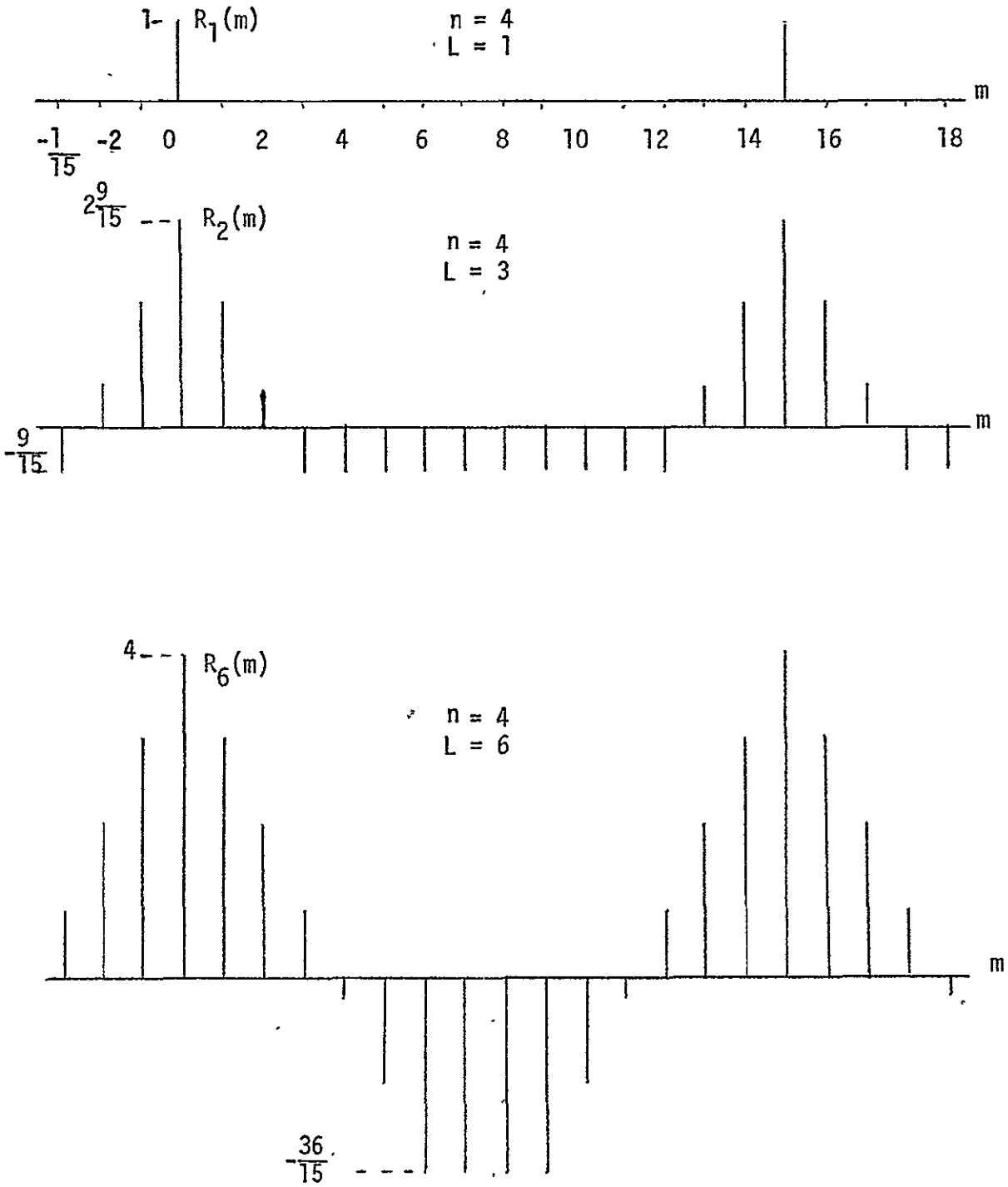


Figure 2-1. Discrete Autocorrelation Function of Unity Weighted Pseudo-random Sequence Sums

Healy [4] that this weighting leads to a uniform probability distribution). Substitution of $a_i = 2^{i-1}$ into equation (2-5) leads to:

$$R_L(m) = \sum_{r=0}^{L-1} \sum_{i=1}^{L-r} 2^{i-1} 2^{i-1+r} R_1(m \pm r)$$

$$= \sum_{r=0}^{L-1} 2^r \sum_{i=1}^{L-r} 4^{i-1} R_1(m \pm r),$$

$$= \sum_{r=0}^{L-1} 2^r \frac{4^{L-r}-1}{3} R_1(m \pm r) , \quad r < L \qquad (2\text{-}8)$$

The last form follows from the fact that the inner sum on the next to the last line is just a truncated geometric series. Equation (2-8) is plotted in Figure 2-2 for $n = 4$ and $L = 4$.

The results obtained above apply to random as well as pseudo-random inputs. If the process is binary (taking values 1 and -1), the autocorrelation function can be defined as:

$$R_1(m) = \begin{cases} 1, & m = 0 \\ 0, & m \neq 0 \end{cases} \qquad (2\text{-}9)$$

Other non-periodic (purely random) autocorrelations may be used in equation (2-5).

2.2 A Convolution Approach

In the analysis above we were interested in the relation of the output autocorrelation given the autocorrelation of a known input to a specified system. The system actually is a non-recursive digital filter, that is, a digital filter that does not use past output values to obtain subsequent outputs (outputs do not "recur".). Douce [10] has pointed out the analogy

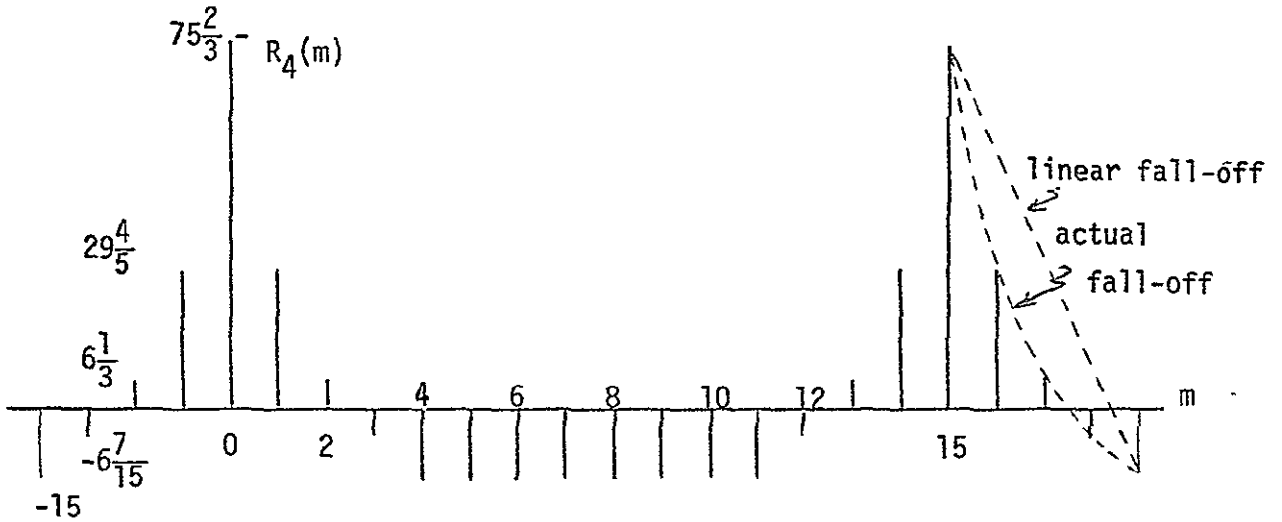


Figure 2-2. Discrete Autocorrelation Function of Binary Weighted Pseudo-random Sequence Sums ($n = 4$, $L = 4$)

with the problem of autocorrelation functions of signals into and out of continuous filters. As Papoulis [11] shows:

$$R_{yy}(\tau) = R_{xx}(\tau) * h^{*}(-\tau) * h(\tau) \tag{2-10}$$

where $R_{xx}(\tau)$ is the input autocorrelation function to a filter, $R_{yy}(\tau)$ is the output autocorrelation function, $h(\tau)$ is the impulse response, the asterisk operators denote convolution, and the asterisk superscript complex conjugation.

In analogy with the $R_L(m)$ above, the output autocorrelation of non-recursive digital filter shown in Figure 1, can be obtained from:

$$R_L(m) = R_1(m) * h(m) * h(-m) \tag{2-11}$$

where h(m) is the discrete system analog to an impulse response, It is the response of the system to a sequence { 1 0 0 0 ··· }, and it is simply equal to the sequence of weights { a_1 a_2 ... a_i ... a_L } . It is not necessary that the input autocorrelation $R_1(m)$ in equation (2-11) have the same form as that given in equation (2-2). In fact it may have any form. If the form of $R_1(m)$ is not simple, it may be more convenient to apply equation (2-11) rather than equation (2-5).

Equation (2-11) suggests the hardware implementation shown in Figure 2-3. This device may be attractive as a pedagogical tool to show the effect of filters on autocorrelation functions. The shift register on the left is easy to build if the autocorrelation is binary, such as { 1 0 0 0 ... } Then the shift register on the left will also be binary. But the shift register on the right must be m-ary where m may be quite large depending on the bounds put on the input and the weights. If the input is not binary, neither shift register can be binary. The major difficulty in building a device such as that shown in Figure 2-3 appears to be in the m-ary shift register.

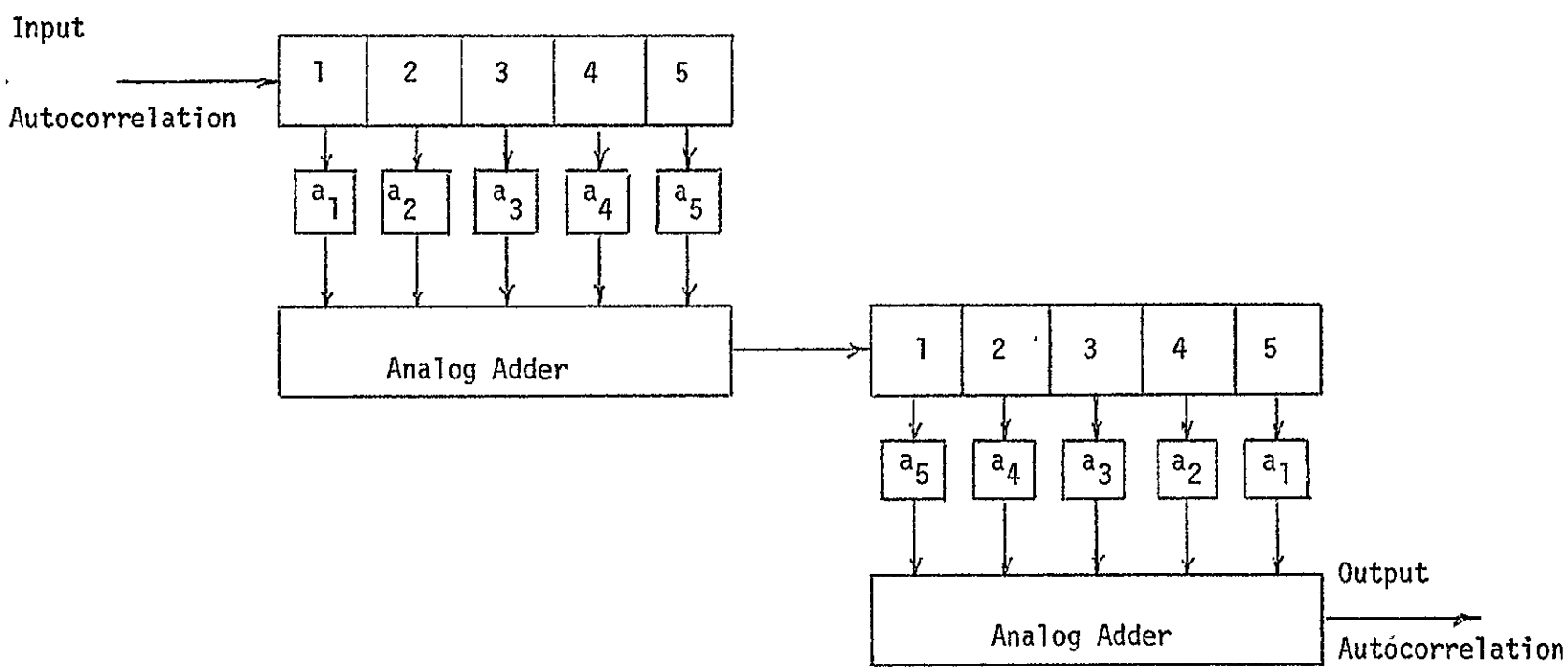


Figure 2-3. Hardware Implementation of an Autocorrelation Function Generator

2.3 Synthesis of Autocorrelation Function

Consider the basic circuit of Figure 0-1. Given an input autocorrelation $R_x(m)$, specify the weights a_i necessary to obtain a desired output autocorrelation $R_y(m)$. This is the synthesis problem.

This problem is solved through use of equation (2-6). Substitution of the first L integers $\{ 0, 1, 2, \ldots, L \}$ into (2-6) yields L independent simultaneous algebraic equations in the L unknowns a_i where $1 \leq i \leq L$. Other equations which might be written, for negative m, are not independent because the autocorrelation is an even function. Hence we have L equations in L unknowns, and we can therefore find the required a_i .

Of course not all autocorrelations can be synthesized for a given input autocorrelation. For example, if we let the input autocorrelation be

$$R_x(m) = \begin{cases} 1, & m = 0 \\ 0, & m \neq 0 , \end{cases} \tag{2-12}$$

then it is not possible to select a_1 and a_2 to obtain an autocorrelation of the form:

$$R_y(m) = \begin{cases} 1, & m = 0 \text{ or } \pm 1 \\ 0, & m \neq 0 \text{ and } m \neq \pm 1 \end{cases} \tag{2-13}$$

To see this let us carry out the synthesis operation. Equation (2-6), for $L = 2$, becomes

$$R_y(m) = \sum_{r=0}^{1} \sum_{i=1}^{2-r} a_i a_{i+r} R_x(m \pm r)$$

$$= a_1^2 R_x(m) + a_2^2 R_x(m) + a_1 a_2 R_x(m \pm 1) \tag{2-14}$$

$$R_y(0) = a_1^2 + a_2^2$$

$$R_y(1) = a_1 a_2 \tag{2-15}$$

If we substitute $R_y(0) = 1$ and $R_y(1) = 1$ into equations (2-15) we find that this pair of equations has no real solution. Solving (2-15) for a_2 yields:

$$a_2 = \frac{\left(R_y(0) + \sqrt{R_y^2(0) - 4R_y^2(1)} \right)^{1/2}}{2} \tag{2-16}$$

It is clear that a necessary condition for a real a_2 is

$$\frac{R_y(1)}{R_y(0)} \leq \frac{1}{2} \tag{2-17}$$

The equality in (2-17) gives the largest possible ratio of $R_y(1)$ to $R_y(0)$. As (2-16) suggests, this corresponds to uniform weighting ($a_1 = a_2 = \frac{1}{\sqrt{2}}$). It seems reasonable, though it has not been proven here, that uniform weighting should lead to the autocorrelation function with the largest possible relative values for $m \neq 0$. We also note that uniform weights results in an autocorrelation which falls off linearly with m. (See equation (2-7) and Figure 2-1.) If the above conclusion about relative values is correct, then any weighting which is not uniform should result in an autocorrelation fall-off which is more rapid than a linear function of m. This phenomenon is illustrated on the right side of Figure 2-2 for binary weighting.

2.4 Continuous Autocorrelation Functions

In this section we consider the case where the clocking reference is not known. Then the autocorrelation function of a pseudo-random signal is [9]:

$$R(\tau) = \begin{cases} \frac{P+1}{P}\left(1 - \frac{|\tau|}{T}\right) - \frac{1}{P}, & |\tau| \leq T \\ -\frac{1}{P}, & T \leq \tau < (P-1)T \end{cases} \tag{2-18}$$

where $p = 2^n-1$, and T is the time between clock pulses. Equation (2-18) specifies $R(\tau)$ over a time pT, which is one period. $R(\tau)$ then repeats with this period.

The power spectral density of $R(\tau)$ in (2-18) can be obtained through the Wiener-Khintchine theorem. As shown in the appendix (section 2-5) (See also [9] for result, but not derivation):

$$S(w) = \frac{1}{p^2}\ \delta\ (w) + \frac{p+1}{p^2}\ \left(\frac{\sin w\ T/2}{wT/2}\right) \sum_{\substack{n=\infty \\ n\neq 0}}^{2\ \infty} \delta\ (w-n\ w_o), \qquad (2\text{-}19)$$

where $w_o = \frac{2\pi}{pT}$.

This spectrum is plotted in figure 2-4 for p=15. The first zero in the (sinx)/x envelope occurs at $f = 1/T$.That is, the zero of $\frac{\sin x}{x}$ is dictated by the pulse width or time between clocking pulses. The spacing between frequency components depends on the period. There are just p lines in the spectrum from the origin to the first zero of $S(w)$. This result suggests that the pseudo-random signal can be used as an excellent source of repeatable "white" noise by decreasing T far enough so that the spectrum is essentially flat over the range of interest.

To summarize:

a) Decreasing T increases the frequency range over which the spectrum is "flat."

b) Increasing p decreases the spacing between lines or increases the number of lines in a given band.

For example, suppose we require a spectrum which is flat to 5%. What are the constraints on f and T ?

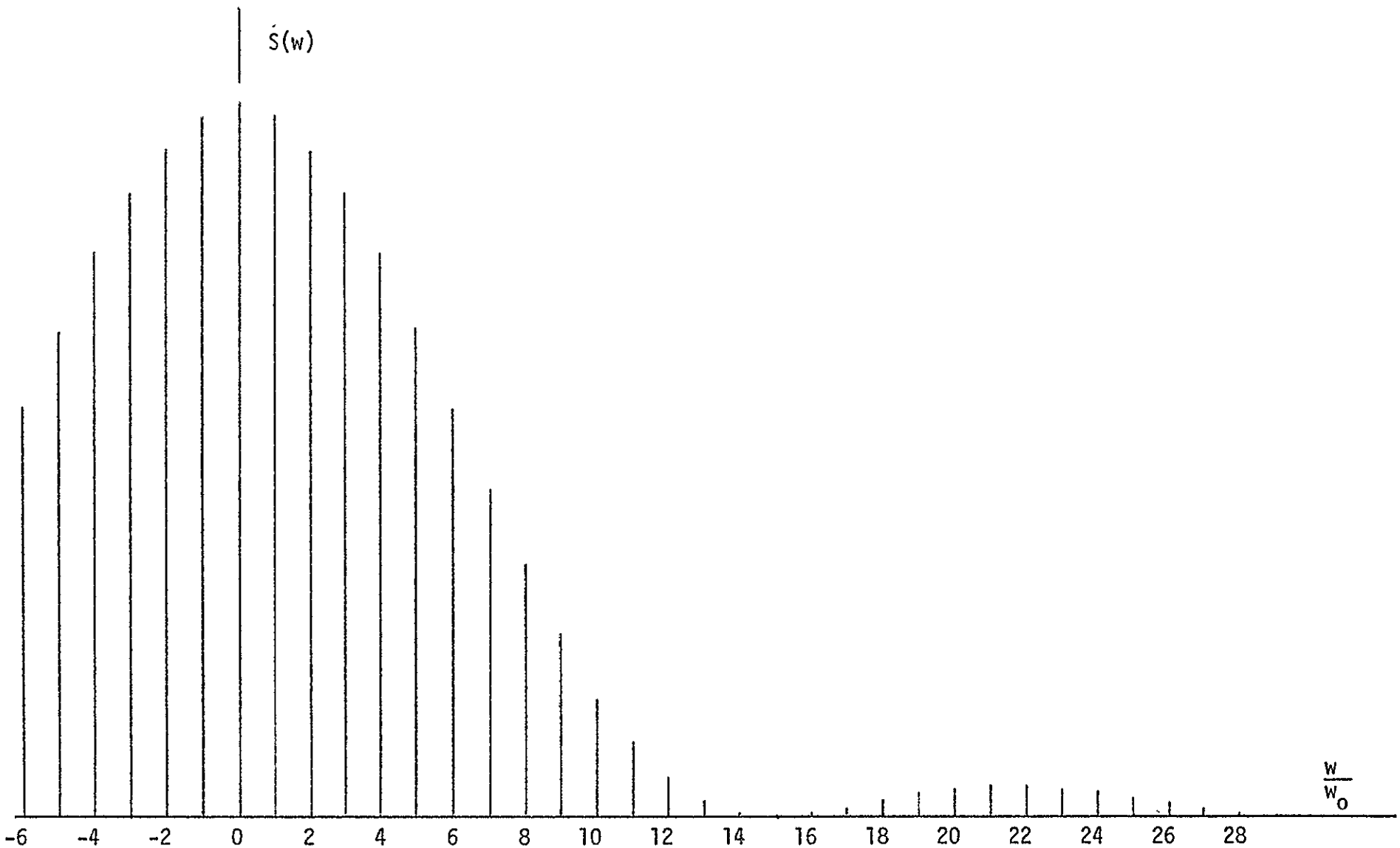


Figure 2-4. Power Spectral Density of a Pseudo-Random Signal
Clock Frequency = $\frac{2\pi}{w_o}$, n = 4 (p = 15)

$$\left(\frac{\sin \frac{wT}{2}}{\frac{wT}{2}}\right)^2 = 0.95$$

$$\frac{\sin \frac{wT}{2}}{\frac{wT}{2}} = 0.975$$

so that

$$fT \approx 0.125$$

$$fmax \approx \frac{1}{8T}$$

We now turn to the problem of the power spectral density and autocorrelation function of the output signal from the weighting system (or filter) of figure 0-1.. The corresponding input-output relations are [11]:

$$R_y(\tau) = R_x(\tau) * h(\tau) * h(-\tau) \tag{2-20}$$

$$S_y(w) = |H(jw)|^2 S_x(w) \tag{2-21}$$

where H(jw) is the transfer function (or Fourier transform of the impulse response)of the system.

It is probably easier in most cases to use equation (2-21). For the circuit of figure 0-1 the transfer function is:

$$H(jw) = \sum_{i=1}^{L} a_i e^{-ijwT} \tag{2-22}$$

where T is again the time between clock pulses. Equation (2-22) is simply a series of shift terms obtain from the time-shift theorem of transform theory. Then:

$$S_y(w) = \left|\sum_{i=1}^{L} a_i e^{-ijwT}\right|^2 S_x(w) \tag{2-23}$$

At this point we alter the basic problem slightly, adding the input (with a weight a_0) before it enters the register, to the other terms. $S_y(w)$ is

then:

$$S_y(w) = \left|\sum_{i=0}^{L} a_i e^{-jiwT}\right|^2 S_x(w) \tag{2-24}$$

Expansion of the $|\cdot|^2$ term leads to:

$$S_y(w) = \left[\sum_{i=0}^{L} a_i^2 + \sum_{r=1}^{L}\sum_{i=0}^{L-r} a_i a_{i+r} \cos rwT\right] S_x(w) \tag{2-25}$$

If we let the input be a pseudo-random sequence of period p and clocking time T, then $S_x(w)$ is given by (2-19) and $S_y(w)$ becomes:

$$S_y(w) = \frac{1}{p^2}\sum_{i=0}^{L} a_i^2 \delta(w) + \frac{p+1}{p^2}\left|\frac{\sin\frac{wT}{2}}{\frac{wT}{2}}\right|^2 \sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} \delta(w-nw_0)$$

$$\times \left(\sum_{r=1}^{L}\sum_{i=0}^{L-r} a_i a_{i+r} \cos rwT\right) \tag{2-26}$$

For a particular set of $\{a_i\}$ equation (2-26) may be reduced and eventually transformed to obtain $R_y(\tau)$. Let us consider a special case where the input is pseudo-random and the weighting is uniform. A similar case where the weighting was uniform and the input purely random was solved by Wolf [12], and the development here closely parallels that work.

If $a_i=1$ for $0 \leq i \leq L$ equation (2-22) becomes:

$$H(jw) = e^{-jwLT/2}\,\frac{\sin\frac{L+1}{2}wT}{\sin\frac{wT}{2}} \tag{2-27}$$

$$|H(jw)|^2 = \frac{(L+1)^2}{\sin^2\frac{wT}{2}}\;\frac{\sin^2\frac{L+1}{2}wT}{(L+1)^2} \tag{2-28}$$

Substitution of (2-28) and (2-19) into (2-21) yields:

$$S_y(w) = \frac{(L+1)^2}{p^2}\,\delta(w) + \frac{p+1}{p^2}(L+1)^2 \cdot \left|\frac{\sin(L+1)\frac{wT}{2}}{(L+1)\frac{wT}{2}}\right|^2 \sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} \delta(w-nw_0) \tag{2-29}$$

A comparison of equations (2-29), (2-19) and (2-18) suggests that (2-29) must be the Fourier Transform of:

$$R(\tau) = \begin{cases} \frac{P+1}{P}(L+1)\left(1 - \frac{|\tau|}{(L+1)T}\right) - \frac{(L+1)^2}{P}, & |\tau| \leq (L+1)T \\ \frac{-(L+1)^2}{P} & (L+1)T \leq |\tau| \leq (P-L-1)\,T \end{cases} \quad (2\text{-}30)$$

Again, $R(\tau)$ is periodic with period pT. The result is simply another triangular form as shown in Figure 2-5.

2.5 Appendix

The purpose of this appendix is to show the derivation of equation (2-19) from (2-18). We start by writing equation (2-18) as:

$$R(\tau) = R\,(\tau) - \frac{1}{P} \quad \text{where} \quad (2\text{-}31)$$

$$R\,(\tau) = \begin{cases} \frac{P+1}{P}\left(1 - \frac{|\tau|}{T}\right), & |\tau| \leq T \\ 0, & T \leq \tau < (P-1)T \end{cases} \quad (2\text{-}32)$$

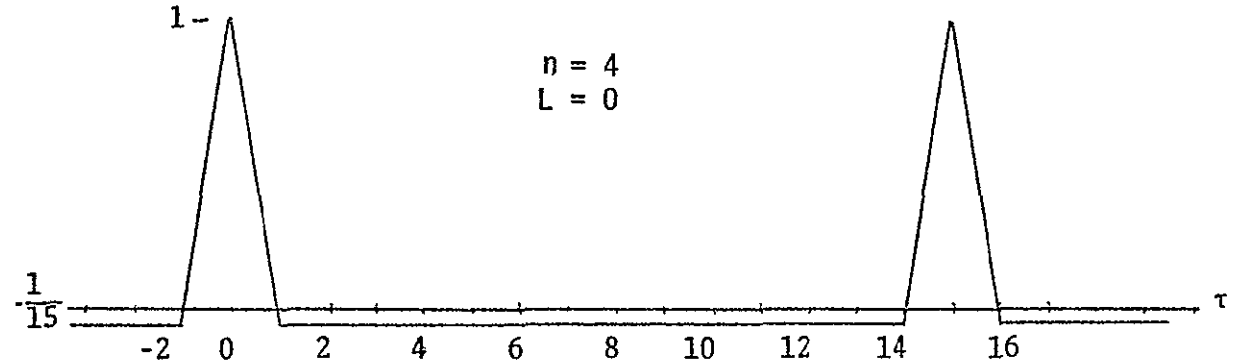
Since $R(\tau)$ is periodic we seek its Fourier series representation.

$$R(\tau) = \sum_{n=-\infty}^{\infty} C_n\, e^{jnw_0\tau}, \quad \left(w_0 = \frac{2\pi}{pT}\right) \quad (2\text{-}33)$$

The "DC" term, C_0, is:

$$C_0 = \frac{1}{pT}\int_{-T}^{(P-1)T} R(\tau)d\tau$$

$$= \frac{1}{pT}\int_{-T}^{T} \frac{P+1}{P}\left(1 - \frac{|\tau|}{T}\right) d\tau - \frac{1}{pT}\int_{-T}^{(P-1)T} \frac{1}{P}\, d\tau$$



Continuous Autocorrelation Function of Each Sequence

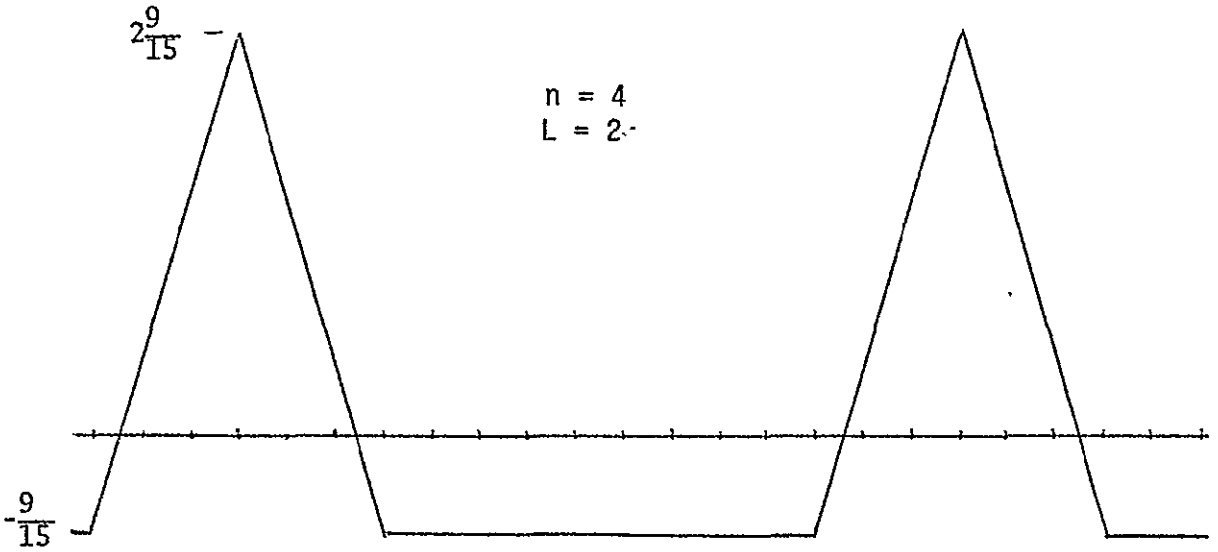


Figure 2-5. Continuous Autocorrelation Function of Unity Weighted Pseudorandom Sequence Sums

$$= \frac{p+1}{p^2T}\ (2T - T) - \frac{1}{p^2T}\ pT$$

$$= \frac{1}{p^2} \tag{2-34}$$

The constant term $-\frac{1}{p}$ in $R(\tau)$ will not contribute to the "AC" coefficients in the Fourier series, so we have:

$$C_n = \frac{1}{pT} \int_{-T}^{(p-1)T} R'(\tau)d\tau$$

$$= \frac{1}{pT} \int_{-T}^{T} \frac{p+1}{p} \left(1 - \frac{|\tau|}{T}\right) e^{jnw_o\tau}d\tau$$

$$= \frac{2}{pT} \int_{o}^{T} \frac{p+1}{p} \left(1 - \frac{\tau}{T}\right) \cos nw_o\tau d\tau$$

since $R'(\tau)$ is an even function.

$$C_n = \frac{2(p+1)}{p^2T} \left(\frac{\sin nw_oT}{n\ w_o} + \frac{1-\cos nw_oT}{n^2w_o^2T} - \frac{\sin nw_oT}{n\ w_o}\right)$$

$$= \frac{2(p+1)}{p^2T}\ \frac{2\sin^2 nw_oT/2}{n^2w_o^2T}$$

$$= \frac{p+1}{p^2} \left| \frac{\sin \frac{nw_oT}{2}}{\frac{nw_oT}{2}} \right|^2 \tag{2-35}$$

We now substitute (2-34) and (2-35) into (2-33) to obtain:

$$R(\tau) = \frac{1}{p^2} + \sum_{\substack{n=-\infty \\ n\neq o}}^{\infty} \frac{p+1}{p^2} \left| \frac{\sin \frac{nw_oT}{2}}{\frac{nw_oT}{2}} \right|^2 e^{jnw_o\tau} \tag{2-36}$$

To find the power spectral density we take the Fourier transform of equation (2-36).

$$\delta(w) = \int_{-\infty}^{\infty} R(\tau)e^{-jw\tau}d\tau \tag{2-37}$$

This transform is obtained quite easily through use of the transform:

$$[e^{jnw_o\tau}] = \delta(w - nw_o) \tag{2-38}$$

where $\delta(\cdot)$ is the impulse or Dirac function. Substitution of (2-36) into (2-37) and the application of (2-38) leads to:

$$\delta(w) = \frac{1}{p^2}\,\delta(w) + \frac{p+1}{p^2}\left|\frac{\sin\frac{wT}{2}}{\frac{wT}{2}}\right|^2 \sum_{\substack{n=-\infty \\ n\neq o}}^{\infty} \delta(w - nw_o) \tag{2-39}$$

3.0 The Probability Distributions of Certain Sums of Random Variables

This section deals with sums of random variables of the form

$$S \equiv \sum_{k=1}^{\infty} d^k s_k \quad . \qquad (3\text{-}1)^*$$

In this expression d is an arbitrary fixed rational number of the form $\frac{1}{g}$ with g a natural number $\{ 1, 2, 3, 4, \ldots \}$; and $< s_k >$ is a sequence of independent, identically distributed random variables taking certain non-negative integer values $\{ 0, 1, 2, 3, \ldots \}$ with equal probabilities.

We shall determine the distribution of the random variable S in the following four cases:

Case (i) : $S \equiv Y$; $d = \frac{1}{2}$; $s_k \equiv y_k = 0, 1$.

Case (ii) : $S \equiv Z$; $d = \frac{1}{3}$; $s_k \equiv z_k = 0, 1, 2$.

Case (iii): S ; $d = \frac{1}{g}$, g a number from the set $\{ 1, 2, 3, \ldots \}$;

$s_k = 0, 1, 2, \ldots, g - 1$.

Case (iv) : $S \equiv X$; $d = \frac{1}{3}$; $s_k \equiv x_k = 0, 2$.

Cases (i) and (ii) are specializations of case (iii) . We shall use case (i) to illustrate the application of elementary probability theory to the determination of the (cumulative) distribution function of the sum S in case (iii) . We will see that the random variable S in cases (i),

---

* The symbol $\equiv$ means "is defined to be."

(ii), and (iii) is uniformly distributed on the closed unit interval [0, 1]* . The main emphasis is on case (iv) in which $S \equiv X$ has the Cantor distribution. Figure 3-1 gives an idea of the appearance of the Cantor function. We believe that some of our results in case (iv), especially the discrete approximations to the cumulative distribution function of the random variable X, are new and may be useful for computer study of sums of random variables of the form (3-1).

3.1 The Uniform Cases

Case (i) deals with the random variable

$$Y \equiv \sum_{k=1}^{\infty} \frac{1}{2^k} y_k \tag{3-2}$$

where $y_k = 0$ or 1, each with probability $\frac{1}{2}$, and the $< y_k >$ are mutually independent. There are at least three methods for obtaining the (cumulative) distribution function $F(v) \equiv \text{Prob} \{ Y \leq v \}$. The first method draws on combinatorial analysis, and it applies to all sums of random variables of the form (3-1). The second method uses convolution and leads to a functional equation for the cumulative distribution function F of the random variable Y. The third method employs the familiar transform technique (characteristic functions). We prefer the first method over the other two methods because of its intuitive appeal, and we will present the first method in this section. The other two methods appear in Cramer [13] .

Let us determine F by the first method. We note that, for any $m = 1, 2, 3, \ldots , 2^n$ and $n = 1, 2, \ldots$, the event $\{ \frac{m-1}{2^n} \leq Y < \frac{m}{2^n} \}$

---

* This notation is standard in mathematical literature. Thus it means here that the values of the random variable S lie in the interval from 0 to 1..

occurs if and only if the first n random variables $< y_k : k = 1, 2, \ldots, n >$ take a unique sequence of values $< a_k : k = 1, 2, \ldots, n >$, namely the first n digits in the binary expansion* of $\frac{m-1}{2^n}$. Since the $< y_k >$ take the two values 0 and 1, each with probability $\frac{1}{2}$, and since the $< y_k >$ are independent, the probability that the $< y_k : k = 1, 2, \ldots, n >$ take the particular values $< a_k : k = 1, 2, \ldots, n >$ is $\frac{1}{2^n}$. Thus the probability that Y takes a value in an interval of the form $\left[\frac{m-1}{2^n}, \frac{m}{2^n}\right)$ equals $\frac{1}{2^n}$, which is the length of that interval. Now for each m and n, the events $\{ \frac{m-1}{2^n} \leq Y < \frac{m}{2^n} \}$ are mutually exclusive. This implies that, for $r = 1, 2, 3, \ldots, 2^n$ and $n = 1, 2, \ldots,$ we have**

$$F\left(\frac{r}{2^n} - \right) = \text{Prob} \left\{ \bigcup_{m=1}^{r} \left\{ \frac{m-1}{2^n} \leq Y < \frac{m}{2^n} \right\} \right\}$$

$$= \sum_{m=1}^{r} \text{Prob} \left\{ \frac{m-1}{2^n} \leq Y < \frac{m}{2^n} \right\}$$

$$= \frac{r}{2^n} .$$

---

* If the number $\frac{m-1}{2^n} \in [0, 1)$ admits a binary expansion that terminates after a finite number of digits, then there are actually two different sequences $< a_k >$ of digits representing that number $\frac{m-1}{2^n}$. One sequence terminates after a finite number K of non-zero a_k's. The other sequence has repeating 1's, i.e., $a_{K+1} = a_{K+2} = \ldots = 1$, and it is usually excluded to ensure uniqueness of expansion (Kac [19]). In our problem, however, sequences containing repeating 1's form a set of probability zero and so can be ignored.

** The strict inequality $Y < \frac{r}{2^n}$ necessitates the minus sign in the argument of F because we defined $F(v) \equiv \text{Prob} \{ Y \leq v \}$ rather than $F(v) = \text{Prob} \{Y < v\}$.

Since the function G defined by $G(v) \equiv F(v-)$ is left-continuous on $(0, 1]$*, and since for any $v \in (0, 1]$ there is a sequence of numbers of the form $\langle \frac{r}{2^n} : r = 1, 2, 3, \ldots, 2^n ; n = 1, 2, \ldots \rangle$ converging to v from below, it follows that $F(v-) = v$ for every $v \in (0, 1]$. Thus the distribution of Y is the uniform distribution on $[0, 1]$ with (probability) density function $F'(v) \equiv f(v) = 1, 0 \leq v \leq 1$.

Case (ii) deals with the random variable

$$Z \equiv \sum_{k=1}^{\infty} \frac{1}{3^k} z_k \tag{3-3}$$

where $z_k = 0, 1,$ or 2, each with probability $\frac{1}{3}$, and the $\langle z_k \rangle$ are mutually independent. To determine the cumulative distribution function $F(v) \equiv \text{Prob}\{Z \leq v\}$, we can proceed as in case (i). Letting $\frac{1}{3^n}$ replace $\frac{1}{2^n}$ throughout, we can show that the probability of Z taking a value in an interval of the form $[\frac{m-1}{3^n}, \frac{m}{3^n})$ equals $\frac{1}{3^n}$, which is the length of the interval. The conclusion of the indicated procedure would be that the random variable Z has the cumulative distribution function $F(v) = v$ for $v \in [0, 1]$. We remark in passing that expression (3-3) corresponds to a <u>ternary</u> expansion in which the digits $\langle z_k \rangle$ have been made random variables.

The preceding idea is easily generalized to case (iii) in which we have the random variable

$$S \equiv \sum_{k=1}^{\infty} d^k s_k . \tag{3-4}$$

---

* Read: The half-open unit interval that is open on the left, $v=0$, and closed on the right, $v = 1$.

The constant d may be any rational number of the form $\frac{1}{g}$ with g a natural number $\{1, 2, 3, 4, \ldots\}$; and $\langle s_k \rangle$ is a sequence of independent, identically distributed random variables taking the values $\{0, 1, 2, \ldots, g-1\}$ with equal probabilities $\frac{1}{g}$. Considerations similar to those preceding make it evident that expression (3-4) can be regarded as the expansion of any number in the unit interval in the base $g = \frac{1}{d}$. Thus the random variable S of expression (3-4) is uniformly distributed on $[0, 1]$. In summary, the random variable S of expressions (3-1) or (3-4) is uniformly distributed when the following two conditions hold simultaneously:

1. $\langle s_k \rangle$ mutually independent.
2. $\text{Prob}\ \{s_k = 0\} = \text{Prob}\ \{s_k = 1\} = \ldots = \text{Prob}\ \{s_k = \frac{1}{d} - 1\} = d$.

3.2 A Non-uniform Case

Case (iv) deals with the random variable

$$X \equiv \sum_{k=1}^{\infty} \frac{1}{3^k} x_k \tag{3-5}$$

where $x_k = 0$ or 2 , each with probability $\frac{1}{2}$, and the $\langle x_k \rangle$ are mutually independent. There are at least two methods for arriving at a formula for the (cumulative) distribution function $F(v) \equiv \text{Prob}\ \{X \leq v\}$ of the random variable X. The first method uses a great deal of intuition. Motivated by Figure 3.1 that shows the cumulative distribution function $F_{(4)}$ of the partial sum $\sum_{k=1}^{4} \frac{1}{3^k} x_k$, we speculate that the Cantor function is the cumu-

lative distribution function of the random variable X. Then we prove that this is indeed correct by using a theorem that we have formulated expressly for this purpose. The second method is independent of the first method, and it uses a combinatorial argument. We determine a formula for the cumulative distribution functions $F_{(K)}$ of the partial sums $\sum_{k=1}^{K} \frac{1}{3^k} x_k$, and then we obtain F as the limit of $F_{(K)}$ as K approaches infinity. The first method is quicker, and we will present it in this section. The second method appears in Cramer [13]; here we will give only the key results obtained from the second method.

The Cantor function G, which P. Halmos [14] gives, has the following form:

$$G(v) = \begin{cases} \sum_{k=1}^{L-1} \frac{c_k}{2^k} + \frac{1}{2^L}, & L(v) < \infty \\ \\ \sum_{k=1}^{\infty} \frac{c_k}{2^k}, & L(v) = \infty \end{cases} \tag{3-6}$$

where

$$v = \sum_{k=1}^{\infty} \frac{b_k}{3^k}$$

$$b_k = 0, 1, \text{ or } 2; \quad k = 1, 2, \ldots,$$

$$c_k \equiv \frac{b_k}{2}, \quad k = 1, 2, \ldots, L - 1,$$

and

$$L(v) \equiv \begin{cases} \min \{ k: b_k = 1, \quad k = 1, 2, \ldots \} \\ \infty \quad \text{if} \quad b_k \neq 1, \quad k = 1, 2, \ldots . \end{cases}$$

To show that G of equation (3-6) is the cumulative distribution function F, restricted to $[0, 1]$, of the random variable X, we have formulated the following:

Theorem:

If B is any non-decreasing function defined on the real line, taking values in the interval $[0, 1]$, if X is any random variable, and if the random variable $Y \equiv B(X)$ has a uniform distribution on $[0, 1]$, then B is continuous and is the (cumulative) distribution function of X.

The proof of this theorem appears in the appendix. We now apply this theorem to the function G of equation (3-6). P. Halmos [14] informs us that the Cantor function G has the following properties: G is non-decreasing, takes values between 0 and 1, and is continuous. To see that $G(X)$ has a uniform distribution on $[0, 1]$, we consider

$$G(X) \equiv G\left(\sum_{k=1}^{\infty} \frac{1}{3^k} x_k\right) = \sum_{k=1}^{\infty} \frac{1}{2^k} \frac{x_k}{2} .$$

Setting $\frac{x_k}{2} \equiv y_k$, it is evident that the above expression is identical with expression (3-2) for the random variable Y which we have already shown to be uniform on $[0, 1]$ in section 3.1. We therefore conclude that the random variable X of expression (3-5) has the Cantor function (3-6) as its cumulative distribution function.

Let us now outline the second method of obtaining the cumulative distribution function F of the random variable X defined in expression (3-5) where we use a combinatorial argument. We write expression (3-5) in the form

$$X = \lim_{K \to \infty} \sum_{k=1}^{K} \frac{1}{3^k} x_k$$

$$= \lim_{K \to \infty} X_{(K)}$$

where we define the partial sums

$$X_{(K)} \equiv \sum_{k=1}^{K} \frac{1}{3^k} x_k \quad . \tag{3-7}$$

To determine formulas for the cumulative distribution function $F_{(K)}(v) \equiv \text{Prob}\{ X_{(K)} \leq v \}$, where K is finite, we first note that $F_{(K)}$ has 2^K jump discontinuities, each of size $\frac{1}{2^K}$. (Figure 3.1 shows $F_{(K)}(v)$ for $K = 4$.) If we count the number $N_{(K)}(v)$ of jump discontinuities that $F_{(K)}$ has in the interval $[0, v]$, to the left of some given point v, then we have immediately

$$F_{(K)}(v) = \frac{1}{2^K} N_{(K)}(v) \quad . \tag{3-8}$$

Thus the determination of $F_{(K)}(v)$ reduces to the combinatorial problem of finding $N_{(K)}(v)$. Figure 3.1 suggests that it would be advantageous to express any given real number v in the domain of $F_{(K)}$ in its <u>ternary</u> expansion

$$v = \sum_{k=1}^{\infty} \frac{b_k}{3^k} \quad .$$

The elements of the sequence $\langle b_k \rangle$ are numbers (not random variables), and they can be 0, 1, or 2. This representation of points v allows us to analyze the effect of each digit b_k on the number $N_{(K)}(v)$ of jump discontinuities in $[0, v]$. The work which we are leaving out here can be found in Cramer [13] ; let us state only the key results. The number $N_{(K)}(v -)$ of jump discontinuities which $F_{(K)}$ has in the half-open interval $[0, v)$ depends only on the digits $\langle b_k : k = 1, 2, \ldots, \min(L, K) \rangle$, i.e.,

$$N_{(K)}(v -) = N_{(K)}\left(\sum_{k=1}^{\min(L,K)} \frac{b_k}{3^k} -\right)$$

The digits following $b_{\min(L,K)}$ merely place v somewhat to the right of the point $\sum_{k=1}^{\min(L,K)} \frac{b_k}{3^k}$, but still within an interval on which $F_{(K)}$ is constant. The results of the combinatorial argument are the formulas

$$N_{(K)}(v) = \begin{cases} \sum_{k=1}^{K} 2^{K-k} c_k + 1, & L(v) > K \\ \sum_{k=1}^{L-1} 2^{K-k} c_k + 2^{K-L}, & L(v) \leq K \end{cases} \tag{3-9}$$

and the recursion formulas

$$N_{(K+1)}(v) = \begin{cases} 2\, N_{(K)}(v), & L(v) < K+1 \\ 2\, N_{(K)}(v) - 1, & L(v) = K+1 \\ 2\, N_{(K)}(v) - 1 + c_{K+1}, & L(v) > K+1 \end{cases}$$

where $c_{K+1} \equiv \frac{b_{K+1}}{2} = 0$ or 1 when $b_{K+1} = 0$ or 2 . Equation (3-9) can be written in the equivalent form

$$N_{(K)}(v) = \sum_{k=1}^{\min(L-1,K)} 2^{K-k} c_k + 2^{K-\min(L,K)}$$

Substituting this expression into equation (3-8) yields the cumulative distribution function

$$F_{(K)}(v) = \sum_{k=1}^{\min(L-1,K)} \frac{c_k}{2^k} + \frac{1}{2^{\min(L,K)}} ,$$

K finite, of the random variables $X_{(K)}$ defined in expression (3-7).

Let us now obtain the cumulative distribution function F of the random variable X defined in expression (3-5) as the limit of $F_{(K)}$ as K approaches infinity. Since X is the pointwise limit of $X_{(K)}$ as K approaches infinity, $X_{(K)}$ also converges to X in distribution so that we can write $F(v) = \lim_{K \to \infty} F_{(K)}(v)$ at all points v where F is continuous, i.e.,everywhere as shown in Cramer [13]. Performing the limiting operation yields

$$F(v) = \begin{cases} \sum_{k=1}^{L-1} \frac{c_k}{2^k} + \frac{1}{2^L} , & L(v) < \infty \\ \sum_{k=1}^{\infty} \frac{c_k}{2^k} , & L(v) = \infty . \end{cases}$$

This equation is exactly the Cantor function (3-6) that we arrived at in the earlier part of this section.

3.3 Conclusion

This section has dealt with the distributions of the sums $\sum_{k=1}^{\infty} d^k s_k$ where the random variables $\langle s_k \rangle$ are independent and identically distributed, each taking certain non-negative integer values {0, 1, 2, 3, ...} with equal

probabilities. Using elementary probability theory, it was shown that these sums are uniformly distributed on [0, 1] when $\frac{1}{d}$ is a natural number {1, 2, 3, 4, ...} and the $\langle s_k \rangle$ take the values $\{0, 1, 2, \ldots, \frac{1}{d} - 1\}$ for $k = 1, 2, \ldots$, each with probability d. The main emphasis was on the sum $X \equiv \sum_{k=1}^{\infty} \frac{1}{3^k} x_k$ where $x_k = 0$ or 2, each with probability $\frac{1}{2}$. Two methods were presented to show that X has the Cantor function as its cumulative distribution function F. The first method employed a theorem that was formulated expressly to prove this. The second method used combinatorial analysis to arrive at formulas for the cumulative distribution functions $F_{(K)}$ of the partial sums $\sum_{k=1}^{K} \frac{1}{3^k} x_k$. Then F was obtained as the limit of $F_{(K)}$ as K approaches infinity.

The authors feel that this work may furnish a theoretical basis for further studies in the following areas of application:

1. Output distributions of digital filters with known input processes.
2. Distributions of the analogue sums of the weighted outputs obtained from feedback shift registers.
3. Discrete systems identification using known discrete input processes.
4. Singular detection and estimation problems.
5. Determination of the distributions of the sums $\sum_{k=1}^{\infty} d^k s_k$ where $\frac{1}{2} < d < 1$.

3.4 Appendix

Let us prove the theorem that we stated in section 3.2. Since B is non-decreasing, $X \leq v$ implies $B(X) \leq B(v)$. This means $\{X : X \leq v\} \subseteq \{X : B(X) \leq B(v)\}$. Define $B(X) \equiv Y$. Since Y is assumed to be uniform,

we have Prob $\{X \leq v\} \leq$ Prob $\{Y \leq B(v)\} = B(v)$, Similarly, $X \geq v$ implies $B(X) \geq B(v)$, so that Prob $\{X \geq v\} \leq$ Prob $\{Y \geq B(v)\} = 1 - B(v)$; then $1 -$ Prob $\{X < v\} \leq 1 - B(v)$, and Prob $\{X < v\} \geq B(v)$. We have thus obtained the inequalities

$$B(v) \leq \text{Prob } \{X < v\} \leq \text{Prob } \{X \leq v\} \leq B(v)$$

which lead to the equalities

$$\text{Prob } \{X < v\} = \text{Prob } \{X \leq v\} = B(v) .$$

These equalities allow two conclusions. From the equality Prob $\{ X \leq v \} = B(v)$ we conclude that B is the cumulative distribution function of the random variable X . And from the equality Prob $\{X < v\} =$ Prob $\{X \leq v\}$ we conclude that X has no mass points, i.e., B is continuous. This completes the proof of the theorem.

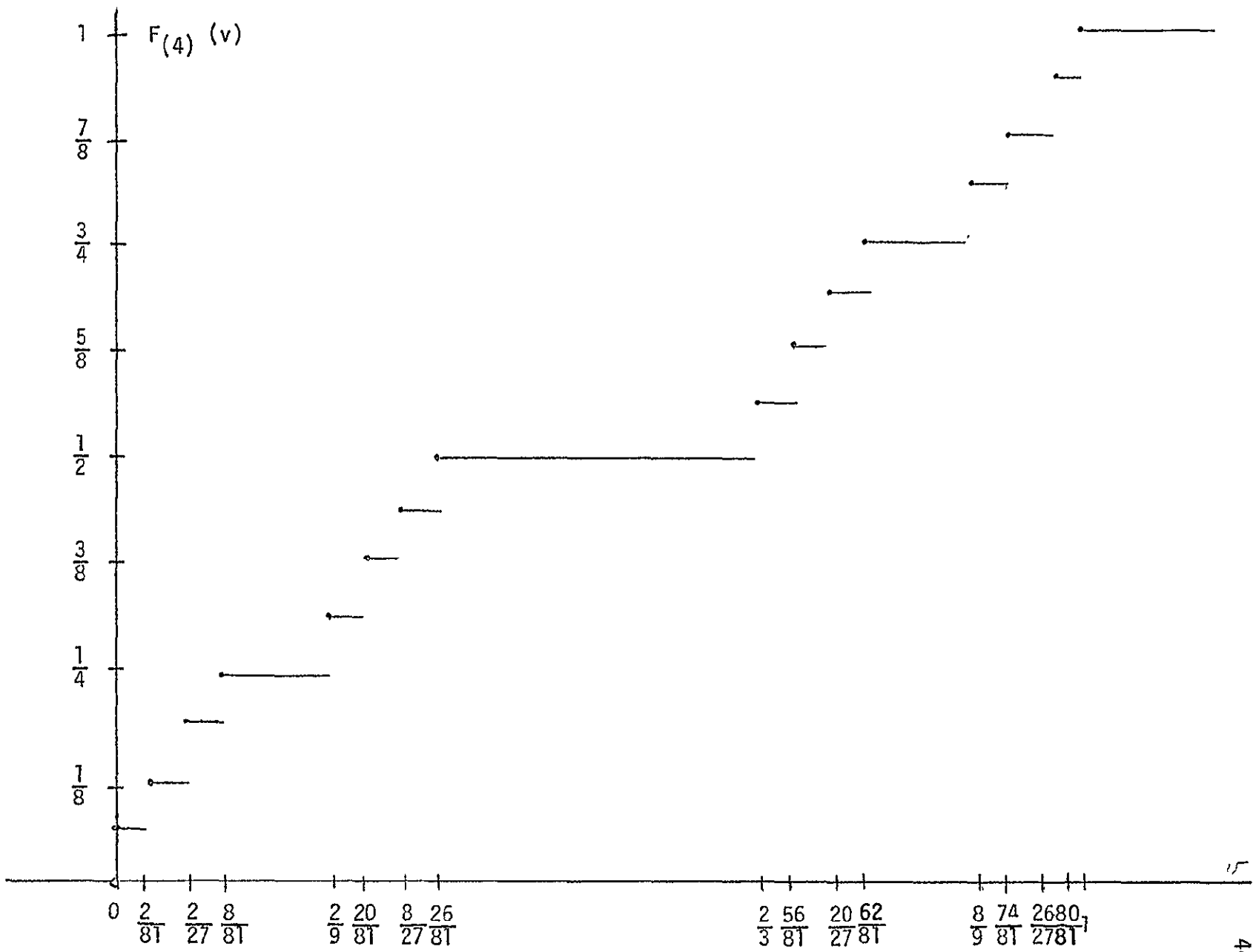


Figure 3-1.Cumulative Distribution Function $F_{(4)}$ (v)

4.0 Pseudo-Random Noise Generation And Digital Filter Implementation

The purpose of this chapter is to discuss the hardware implementation of some of the devices used in previous chapters, and to describe some actual circuits built and some of the experimental results.

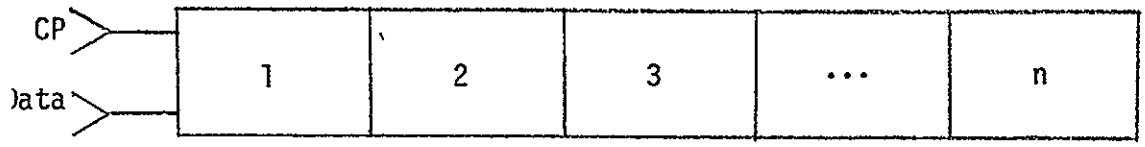
A pseudo-random sequence of length L can be generated from a shift register containing n stages, where each stage can assume M different levels. With proper feedback connections, the length L can reach a maximum of M^n-1 before repeating itself. In this chapter we describe the shift register, its use in generating pseudo-random sequences, the nonrecursive digital filter into which the sequence is fed, and the implementation of the noise generator and filter.

4.1 Shift Register

Let us consider the n-stage shift register shown in Figure 4-1. Each stage can assume the values 0 or 1. Two inputs are provided to the register: a clock input (CP) and a data input.

When the clock pulse input is activated, each stage assumes the state of the stage on its left. The first stage assumes the state of the data input.

A hardware realization of a binary shift register uses flip-flops as its constituent stages. The output of a flip-flop can assume one of two levels, the logical 0 and 1. We will assume that the hardware realization uses J-K flip-flops, whose characteristic table and logic diagram are given in Figure 4-2. Q^k represents the output at the kth clock pulse, Q^{k+1} the output at the (k+1)th clock pulse, and $\bar{Q}^k$ the complement of Q^k. The logic diagram representing Figure 4-1 will then be as shown in Figure 4-3, where the sumbol $-\triangleright\!\!\circ-$ represents an inverter.

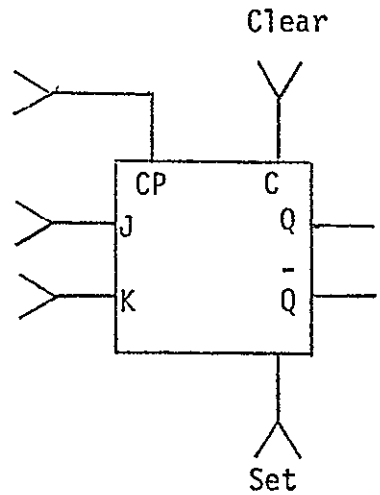


Representation of a Shift-Register

Figure 4-1

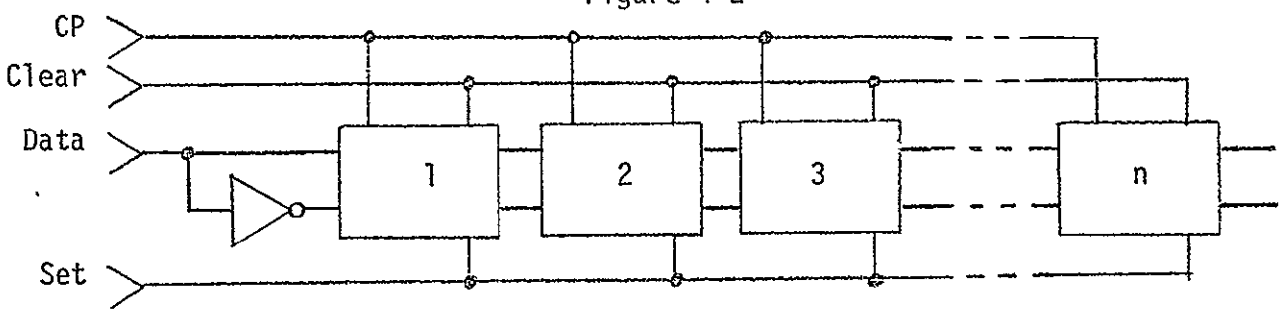
inputs		output
J	K	Q^{k+1}
0	0	Q^k
0	1	0
1	0	1
1	1	$\overline{Q}^k$

a) Characteristic table



b) Logic diagram

Figure 4-2



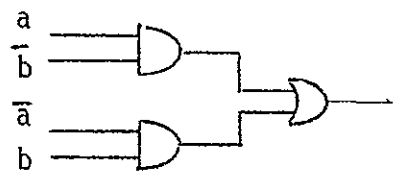
Logic Diagram of a Shift-Register

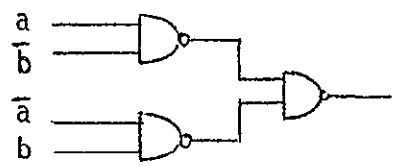
Figure 4-3

4.2 Pseudo-Random Sequence Generator

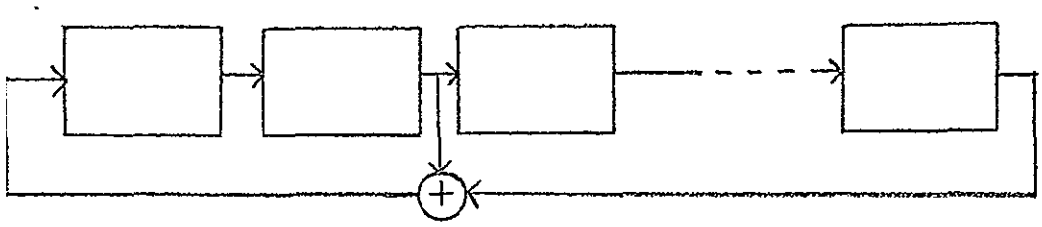
The sequence of states of any of the flip-flops of the register shown in Figure 4-3 is a maximal-length pseudo-random sequence if the proper data are fed into the first flip-flop. These data can be generated by a feedback configuration, involving two or more connections as shown in Figure 4-4, where $\overset{a}{\leftarrow\oplus\leftarrow}$ b denotes a modulo-2 adder, with truth table

b \ a	0	1
0	0	1
1	1	0



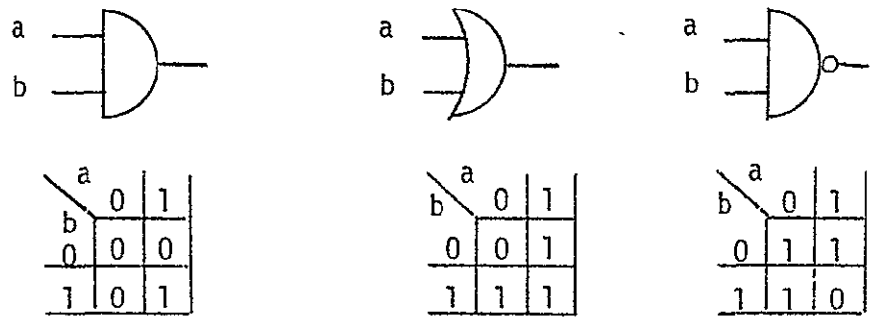


The symbols used here are defined by their truth table given in Figure 4-5. The maximal length sequence $L = 2^n - 1$ will be achieved before repetition of the sequence given the proper feedback connections. For certain lengths of the register, feedback from the output of only two stages will not give the maximal length, and more than two feedback connections are required. Table 4-1 gives the possible feedback connections for a maximal length sequence when the number n of stages goes from 4 to 15. A maximal-length four-stage pseudo-random sequence can then be described by Figure 4-6. The states of the flip-flops of Figure 4-6



Representation of a Pseudo-Random Sequence Generator

Figure 4-4



b \ a	0	1
0	0	0
1	0	1

a) AND gate

b \ a	0	1
0	0	1
1	1	1

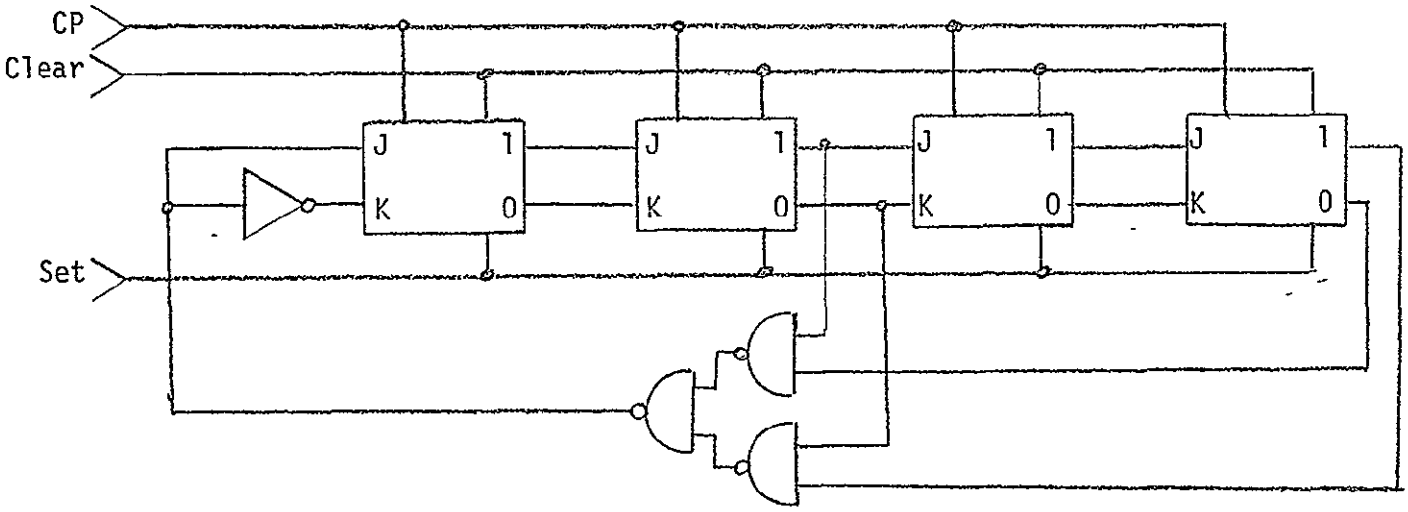
b) OR gate

b \ a	0	1
0	1	1
1	1	0

c) NAND gate

Symbols and Truth Table of Logic Functions

Figure 4-5



Logic Diagram of a 4-stage Pseudo-Random Sequence Generator

Figure 4-6

n	feedback connections for maximal length 2^n-1
4	1 ⊕ 4 or 3 ⊕ 4
5	2 ⊕ 5 or 3 ⊕ 5
6	1 ⊕ 6 or 5 ⊕ 6
7	1 ⊕ 7 or 3 ⊕ 7 or 4 ⊕ 7 or 6 ⊕ 7
8	3 ⊕ 5 ⊕ 7 ⊕ 8
9	4 ⊕ 9 or 4 ⊕ 9
10	3 ⊕ 10 or 7 ⊕ 10
11	2 ⊕ 11 or 9 ⊕ 11
12	6 ⊕ 8 ⊕ 11 ⊕ 12
13	4 ⊕ 6 ⊕ 10 ⊕ 13
14	4 ⊕ 8 ⊕ 13 ⊕ 14
15	4 ⊕ 15 or 7 ⊕ 15 or 8 ⊕ 15 or 14 ⊕ 15

Feedback Connections

Table 4-1

are shown in the timing chart of Figure 4-7, assuming that all the flip-flops have been set to 1 at t=0. Any one of the columns is a pseudo-random sequence of 0 and 1. It should be noted that the all-zero state of the register never occurs. If it did, the register would be locked in that state.

4.3 Digital Filter

Consider the preceding shift register with n stages, and a clock of frequency f_c Hertz. A shift will occur every T seconds ($T = 1/f_c$). At time kT, the last stage of the register contains the state of the first stage at time $(kT - (n - 1)T)$, or $(k - n + 1)T$. At any given time the states of the first stage at times kT, kT-T, kT-2T, kT-3T, up to kT-(n-1)T are present in the register. This suggests the possibility of "filtering" the sequence using a nonrecursive digital filter defined by the equation:

$$y(kT) = \sum_{i=0}^{n-1} a_i \, x\,(kT-iT) \qquad (4\text{-}1)$$

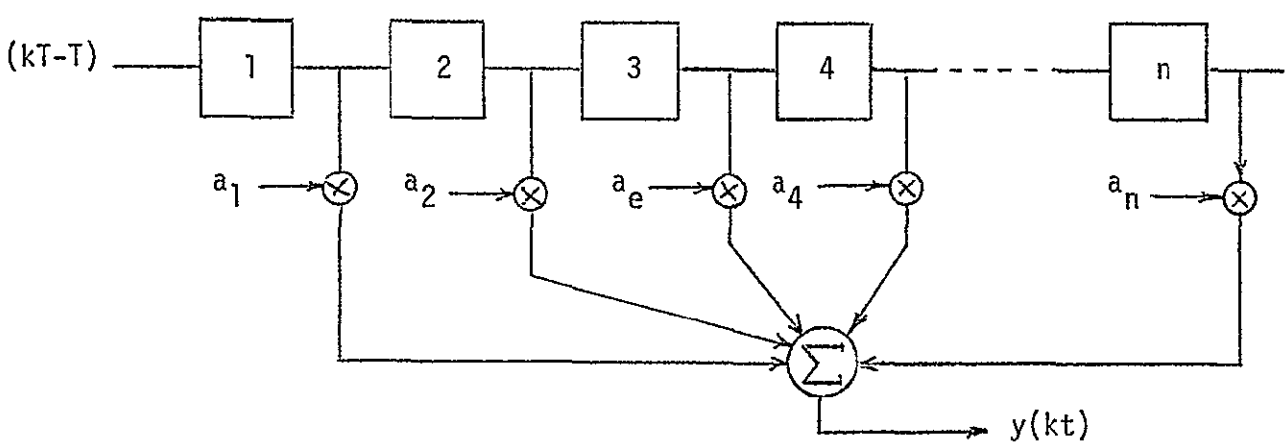
where y(kT) and x(kT) are the output and input of the filter at time kT respectively, and a_i are the weights given to the present input and n-1 previous inputs. With the representation of Figure 4-1, equation (4-1) can be realized by Figure 4-8. A number of hardware realizations of Figure 4-8 can easily be imagined. The simplest one implements the weights by resistors, as shown in Figure 4-9. If we wish to make the contribution of the output of a stage to the total sum y(kT) independent of the state of the other stages (condition that has to be met to assure the linearity of the output summer), then this configuration limits us greatly in the choice of acceptable values for the Rj's. Any Rj should always be much larger than R, such that, looking from the output of one stage, R looks much smaller than the parallel combination of all the other Rj's. Given $R_E \gg R$, R_E can be neglected, and, for all

Clock Flip-flops

	A	B	C	D	
0	1	1	1	1	
1	0	1	1	1	
2	0	0	1	1	
3	0	0	0	1	
4	1	0	0	0	
5	0	1	0	0	
6	0	0	1	0	
7	1	0	0	1	
8	1	1	0	0	
9	0	1	1	0	
10	1	0	1	1	
11	0	1	0	1	
12	1	0	1	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	starts repeating itself

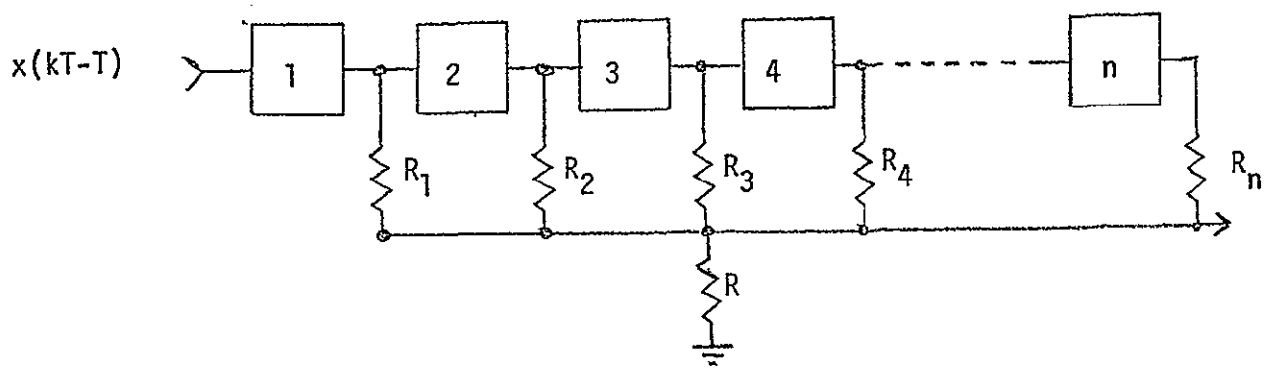
Timing Chart for the Register of Figure 4-6

Figure 4-7



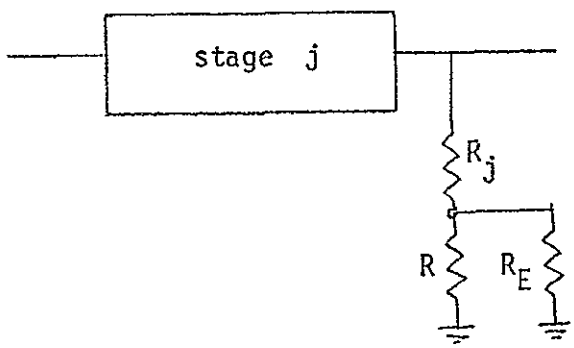
Representation of Equation (1)

Figure 4-8



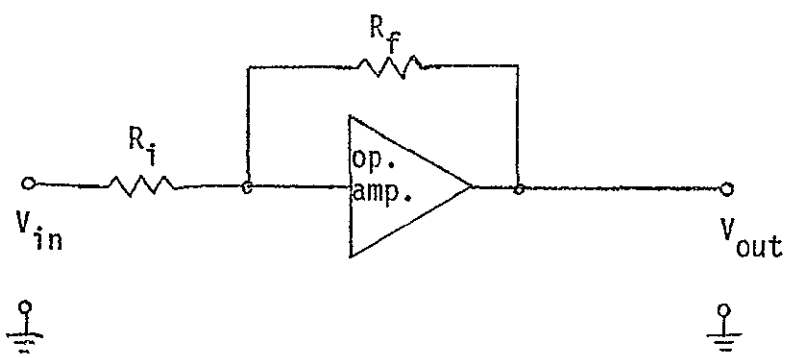
A Simple Hardware Realization of Figure 4-8

Figure 4-9



$$R_E = \sum_{i=1}^{n} \frac{1}{R_i} - R_j$$

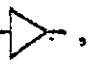
Figure 4-10



Multiplier Using an Operational Amplifier

Figure 4-11

practical purposes, the current through R is the sum of the currents through each Rj. This is shown in Figure 4-10. Ideally, we would not like to be limited in the range of available weights. Unless we use active devices, we cannot expect the weights to exceed 1, and the above realization adds a further limitation on the lowest acceptable weight.

Multipliers can be implemented using operational amplifiers, represented by the symbol  , as shown in Figure 4-11 where $V_{out} = Vin \frac{R_f}{R_i}$.

This last factor is the weight a_j of the filter, having a range that is limited only by the operating characteristics of the amplifier. Figure 4-8 would then have the realization of Figure 4-12. The condition $R_1 >> R_2$ still holds, but does not have any effect on the weights defined by the input and feedback resistors of the operational amplifier.

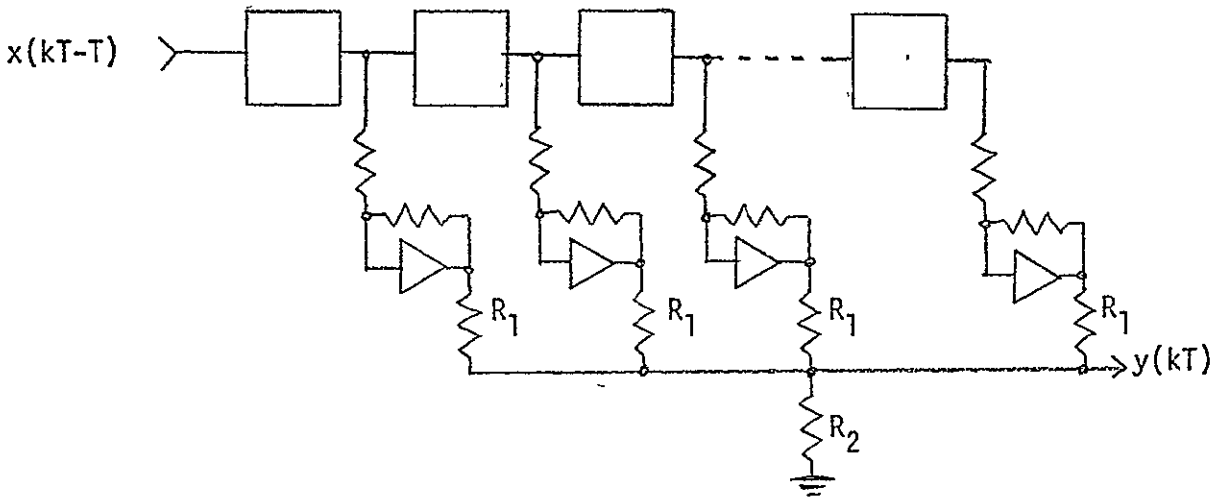
A simpler realization of Figure 4-8 can be implemented using a single operational amplifier, in a summer configuration. Consider the operational amplifier of Figure 4-13 with two inputs (inverted and non-inverted). All resistors have the same value. Due to the non-inverted input held at ground level, point 0 can be considered very close to ground level, independent of the input and feedback currents. Since the input impedance is very high (of the order of megaohms), the current into the amplifier can be neglected, and thus

$$I_{R_f} = I_{R_1} + I_{R_2} + I_{R_3} \tag{4-2}$$

Since all resistors have the same value, the output voltage is

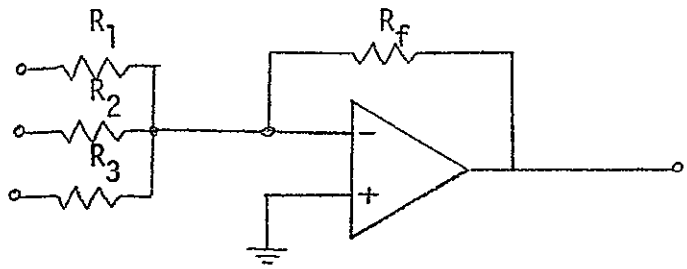
$$V_o = V_1 + V_2 + V_3 \tag{4-3}$$

If we want to add weighting factors to the different inputs, the values of the input resistors can be varied to give



Realization of Figure 4-8 Using Multipliers

Figure 4-12



Summer Configuration of an Operational Amplifier

Figure 4-13

$$\frac{V_o}{R_f} = \frac{V_1}{R_1} + \frac{V_2}{R_2} + \frac{V_3}{R_3}$$

$$V_o = V_1\left(\frac{R_f}{R_1}\right) + V_2\left(\frac{R_f}{R_2}\right) + V_3\left(\frac{R_f}{R_3}\right) \qquad (4\text{-}4)$$

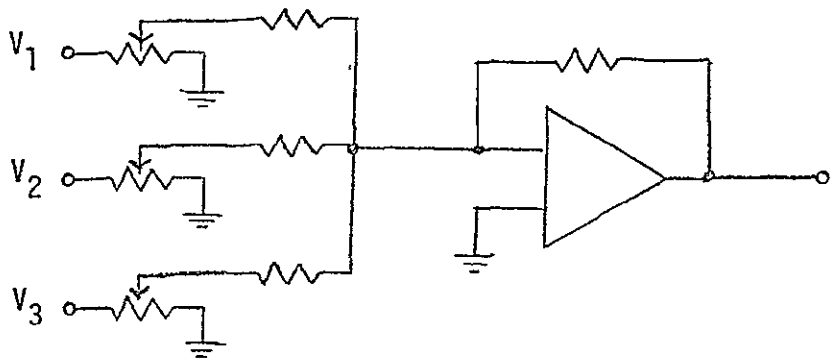
where the factors R_f/R_j are the weighting factors. An alternate way of weighting the input currents is to weight the input voltages before sending them into the summing circuit as shown in Figure 4-14. The current through the input resistor is negligible in comparison to the current through the variable resistor.

This last configuration has been chosen in our implementation. Its shortcoming compared with the previous configuration is the limitation in the range of weighting factors (0 to 1), but its simplicity (1 operational amplifier against k) offsets the shortcoming.

All essential elements for the realization of the pseudo-random sequence generator and the digital filter have been presented. A few more details have to be added.

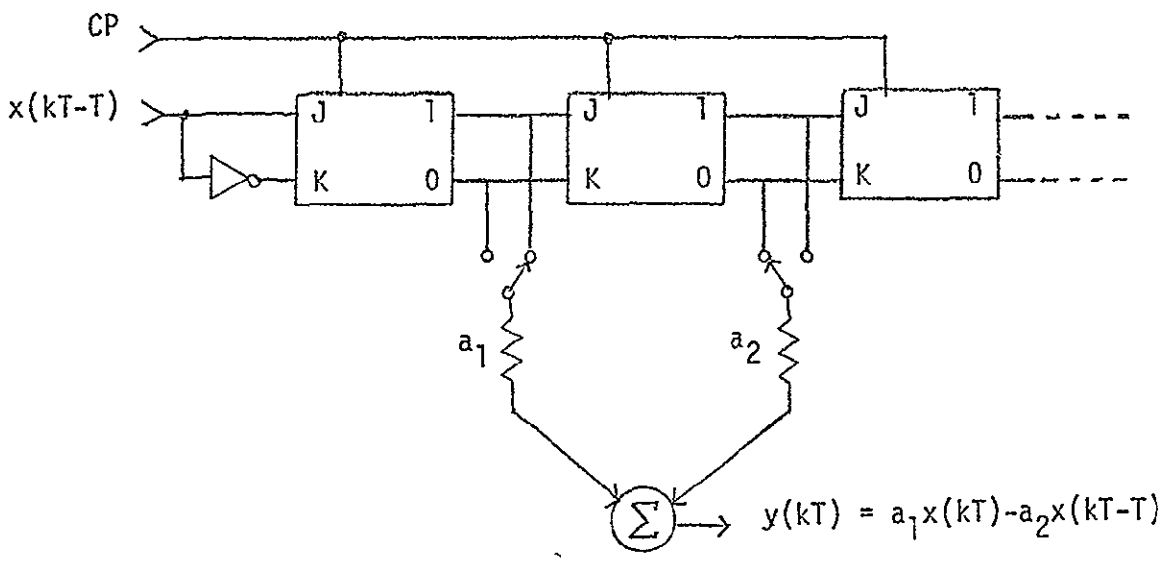
In the implementation of some nonrecursive digital filters, some weights assume a negative value. A resistor cannot have a negative value, but the voltage applied to it can be inverted, giving the same effect. In the implementation of a shift register using flip-flops, the output of any of the stages always has its complement available from the other output of the flip-flop. This is shown in Figure 4-15.

One of the methods of finding the weights of the filter is to realize the inverse Fourier transform of the required frequency spectrum. The result is the impulse response of the filter. For final calibration of the weights, it is useful to see this impulse response on the screen of an



Weighting of the Inputs to a Summer

Figure 4-14



Implementation of Positive and Negative Weights

Figure 4-15

oscilloscope. A way of feeding an impulse to the filter must be provided. The sequence {1,0,0,0,0,0,0,...,} containing m terms (m larger than n, the number of stages used by the filter), is fed into the filter from the shift register and repeated to provide a continuous display on the oscilloscope. The reset line for the register sets the first stage, and clears all the other stages (at the same time providing for the initial conditions appropriate for the generation of the random sequence). The resistor should provide the option of a circular configuration, where the first stage assumes the state of the last stage when the CP input is activated. (This method is used in section 4-7. See Figure 4-32).

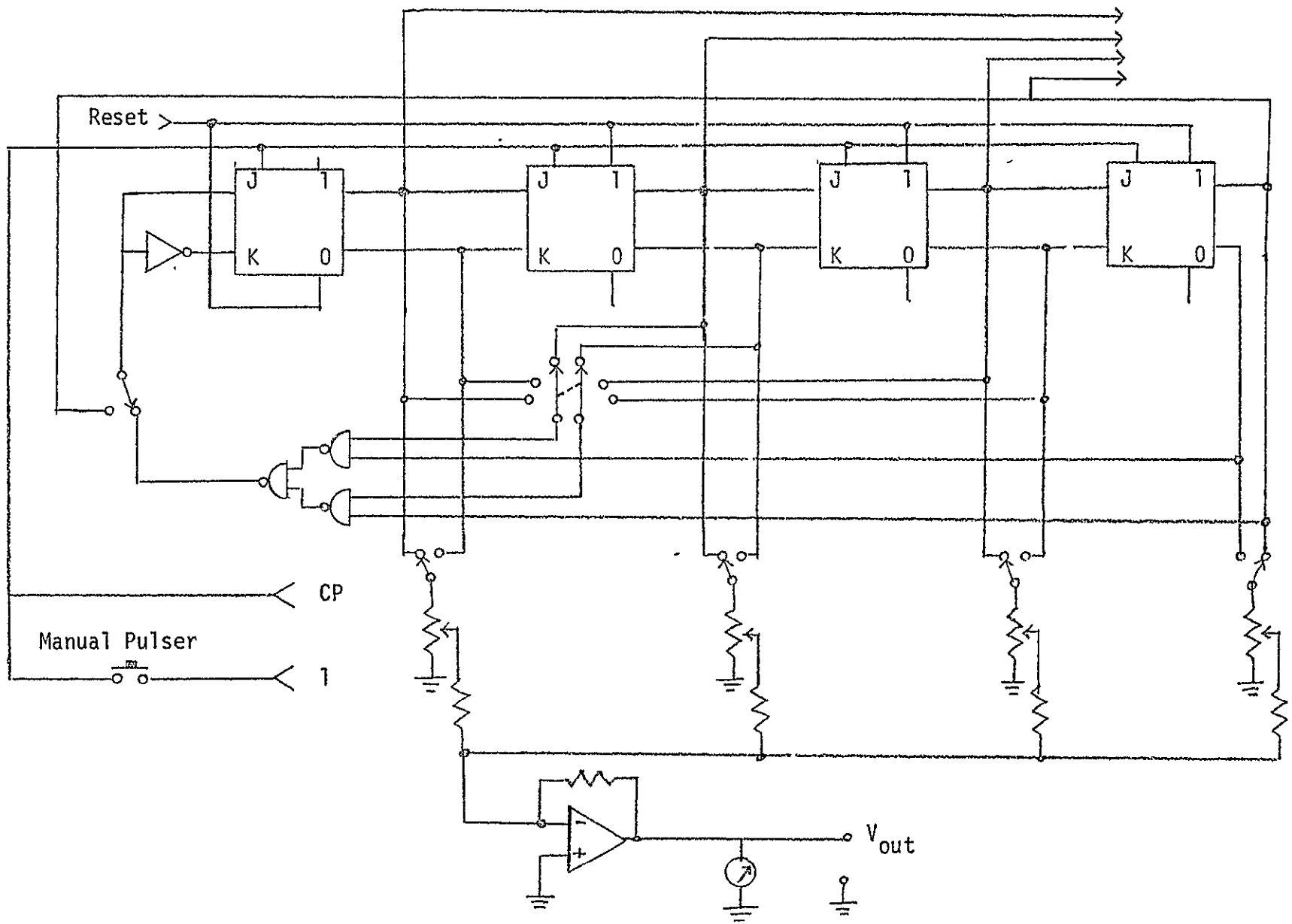
The variable resistors of Figure 4-14 are calibrated to give the required weights. If the stage associated with the resistor to be calibrated is in the 1 state, with all the other stages in the 0 state, the output voltage of the filter will be a function of the setting of that particular variable resistor, and the weight will be given by

$$a = \frac{V_{out}}{V_{max}} \tag{4-5}$$

where V_{out} is the measured output voltage, and V_{max} is the voltage chosen to represent the weight of 1. A manual clock and a D.C. voltmeter at the output are provided to facilitate the calibration.

When testing the operation of the shift register and when calibrating the resistors, it is useful to have a visual display of the state of the stages used by the filter. The output of the stages can be amplified and sent to a light bulb. The clock should have a frequency low enough to allow time to check the feedback operations and the shifting.

Figure 4-16 shows a logic diagram of a four stages shift register together with a digital filter using all four stages.

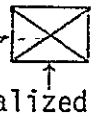


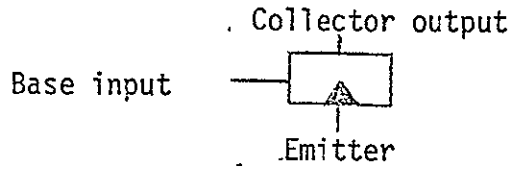
Logic Diagram of a Four-stage Shift Register and a Nonrecursive Digital Filter

Figure 4-16

4.4 A Hardware Realization of the Random Sequence Generator

Implementation of the register of Figure 4-16 uses Digital Equipment Corporation (DEC) flip chip modules. Reference [15] gives a detailed description of the modules. We will present here only the parts of the modules that are used in the implementation of the register. The modules are mounted on a DEC H901 mounting panel, with a type DEC 700D power supply and input panel that provides for the push button pulsers and a clock.

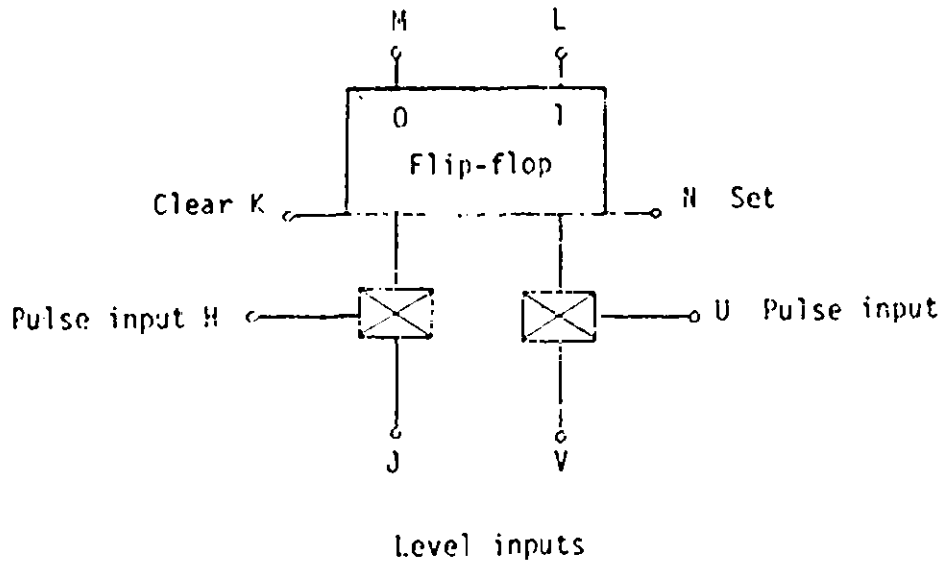
Figure 4-17 shows the logic diagram for the DEC R201 flip-flop. Fifteen of these were used (implementation of a 15-stage shift register). The symbol  denotes a diode-capacitor-diode (DCD) gate. The feedback logic is realized with the DEC R111 NAND/NOR gates, shown in Figure 4-18, with the following symbol to represent a common emitter transistor:



The DEC R107, shown in Figure 4-19, is used to provide for the complement of some of the outputs.

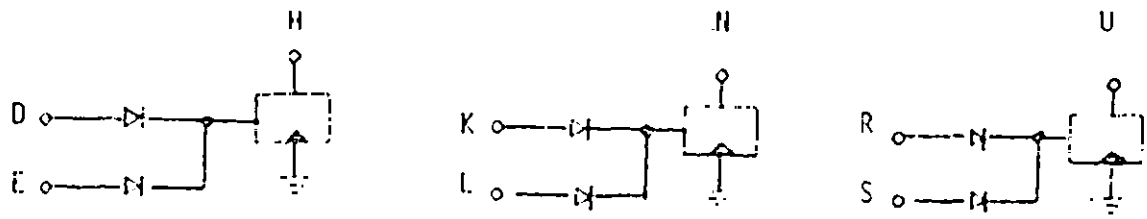
The DEC W520 comparator and DEC W501 Schmitt trigger were used to provide some means for applying an external clock signal and for feeding an external binary sequence. They are shown in Figure 4-20. The symbol stands for a difference amplifier.

The complete wiring diagram for a 15-stage pseudo-random noise generator is given in Figure 4-21. THe outputs of stages 14 and 15 are used as feedback. The dotted lines coming from the DEC R107 module indicate alternate connections when an external sequence is fed into the register. In this case, the feedback connections have to be disconnected (disconnect the wire going into S of R107 coming from R111).

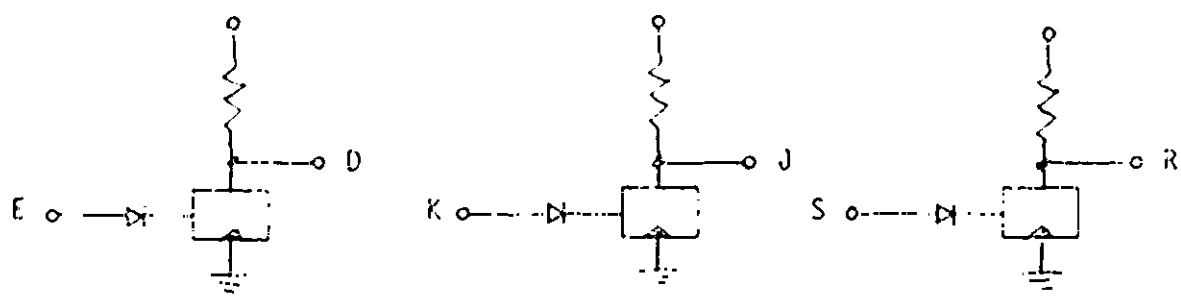


R201 flip-flop

Figure 4-17

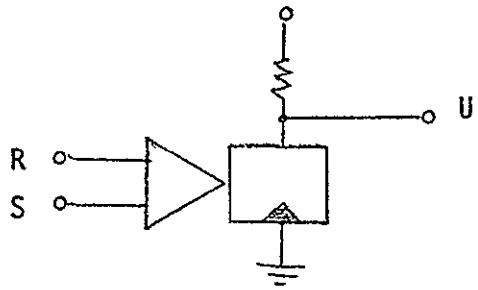


R111 NAND/NOR Gates

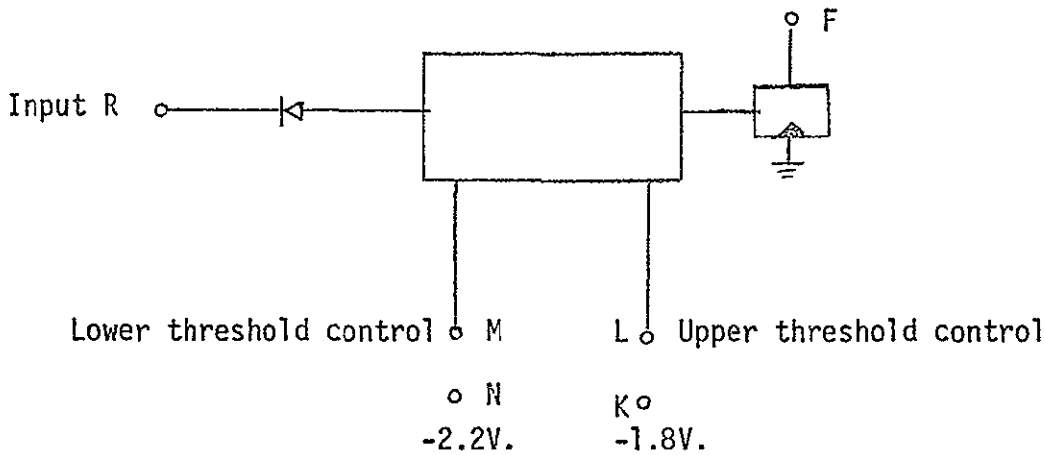


R107 Inverter

Figure 4-19

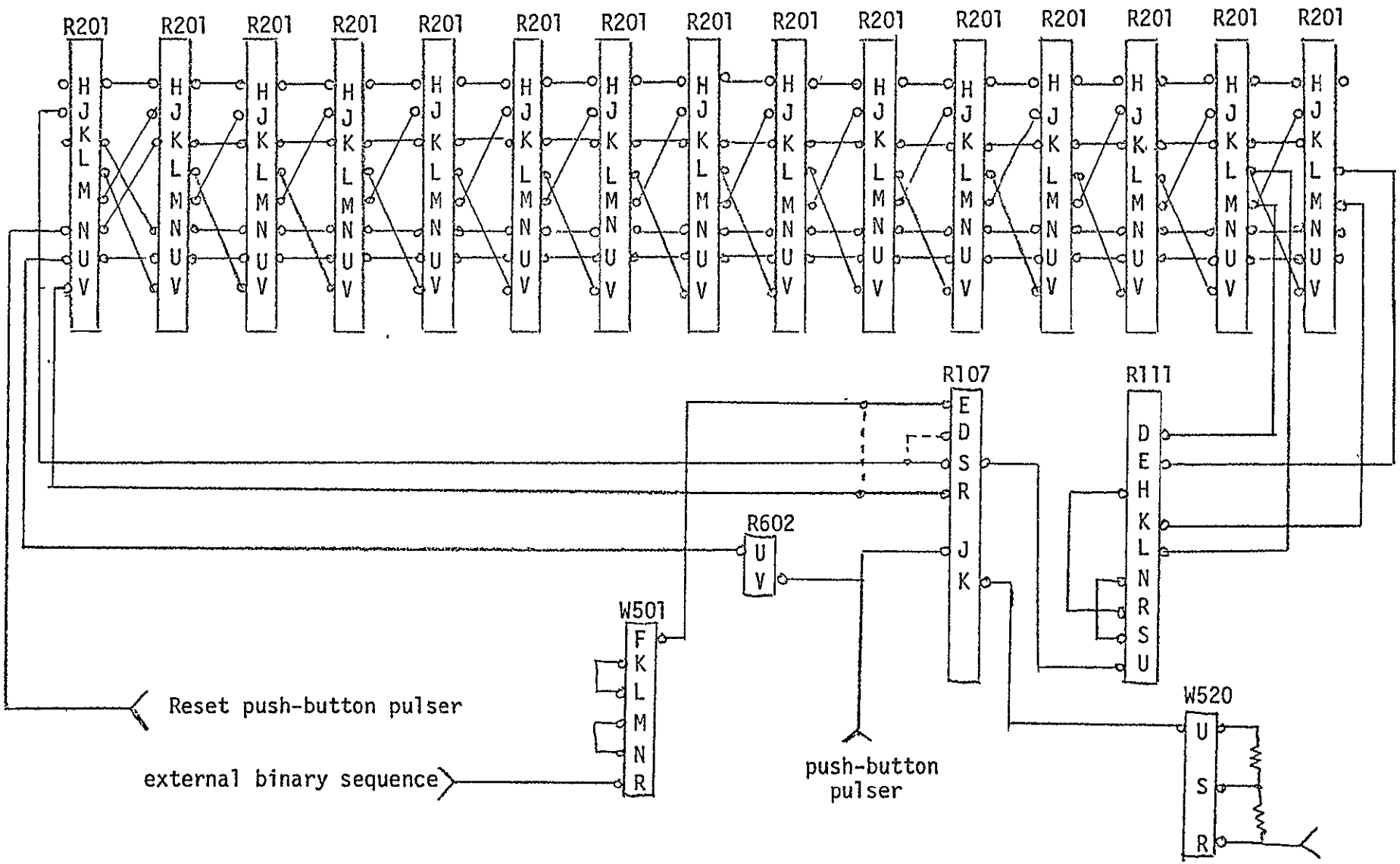


a) W520



b) W501 Schmitt trigger

Figure 4-20



Hardware Realization of a Pseudo-Random Sequence Generator

Figure 4-21

4.5 A Hardware Realization of a Nonrecursive Digital Filter

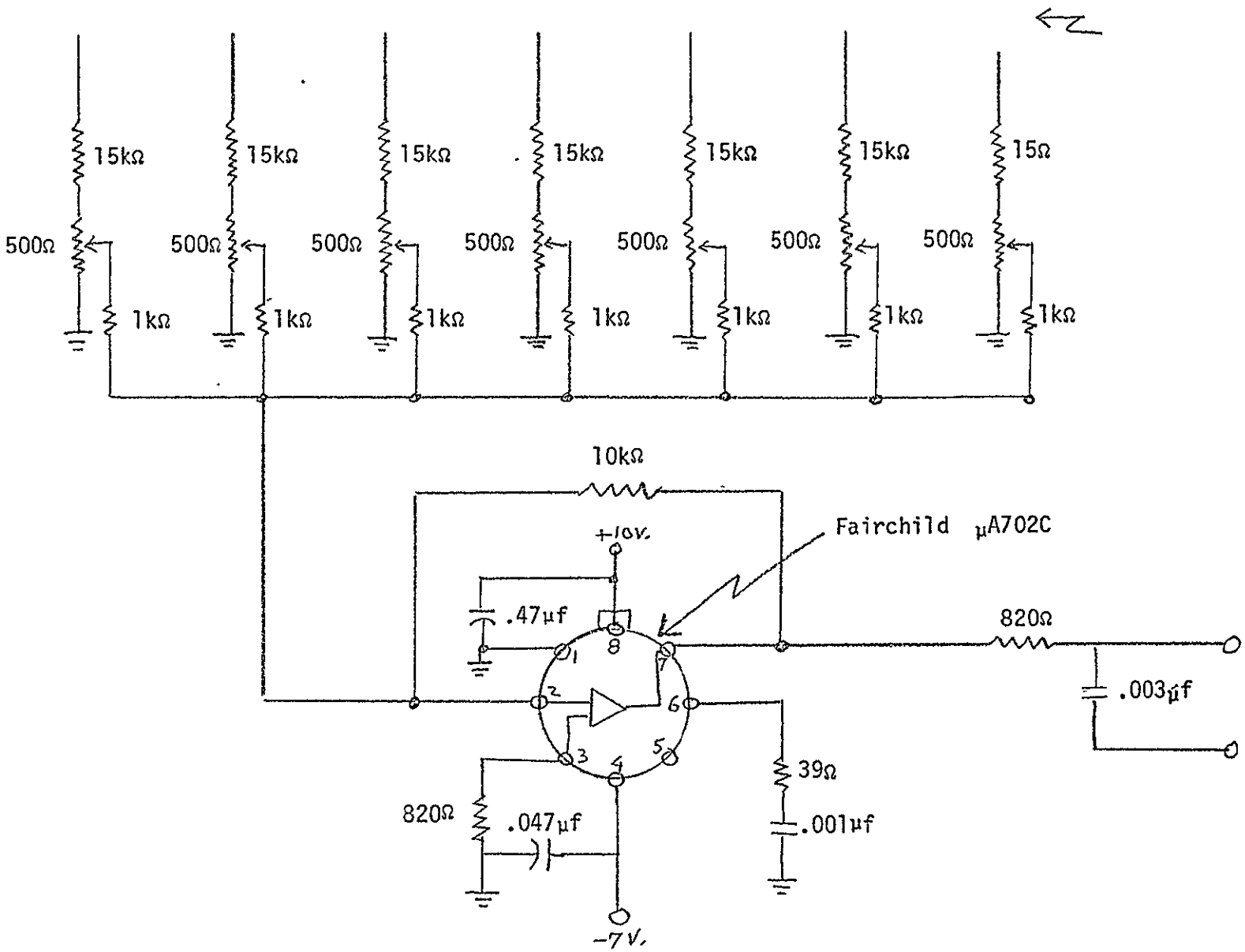
The main difficulty encountered when implementing the diagram of Figure 4-14 was finding an operational amplifier with a good response at a clock frequency around 1 MHz. The Fairchild integrated circuit μA709 and its self-compensated version, the μA741, were first used with a voltage gain of 10. With a slew rate of .3 V/sec. at unity gain, the output waveform was greatly distorted, making it difficult to operate at a clock frequency faster than 100KHz. (The slew rate is one of the factors describing the operation of an operational amplifier: it is defined as the rate of change in the output voltage when a step input voltage saturates one of the inputs).

The Fairchild μA715 is designed for high-frequency applications, with a slew rate of 65 V/sec. at a voltage gain of 100, and 20 V/sec. at unity gain. Efforts were made to use this operational amplifier, but major difficulties were encountered when trying to compensate it. After repeated trials, the "ringing" at the output without an applied input signal could still not be eliminated.

The Fairchild μA702C High Gain, Wideband DC Amplifier was chosen. It has a slew rate six times faster than the μA741, giving satisfactory operating characteristics at a clock frequency of 10 MHz.

In Figure 4-22 the input comes from the output (direct or complemented) of the first seven stages of the shift register of Figure 4-20. The positive voltage applied at pin 8 of the amplifier is provided by the 700D power supply, and a negative voltage of 7 volts coming from an external power supply is applied at pin 4. The characteristics of the μA702C are given in the appendix.

4.6 Experimental Measurements on the Linearity and Frequency Response of the Digital Filter



Hardware Realization of a Nonrecursive Digital Filter

Figure 4-22

A nonrecursive digital filter is defined by equation (4-1), repeated here for convenience

$$y(kT) = \sum_{i=0}^{m-1} a_i \, x(kT - iT) \qquad (4\text{-}1)$$

The output y(kT) is a linear function of the actual input and the (m-1) previous inputs. The circuit of Figure 4-22 will realize equation (4-1) only if the operational amplifier has a linear characteristic in its voltage range of operation. The maximum output voltage is reached under the conditions

$$a_i = 1$$

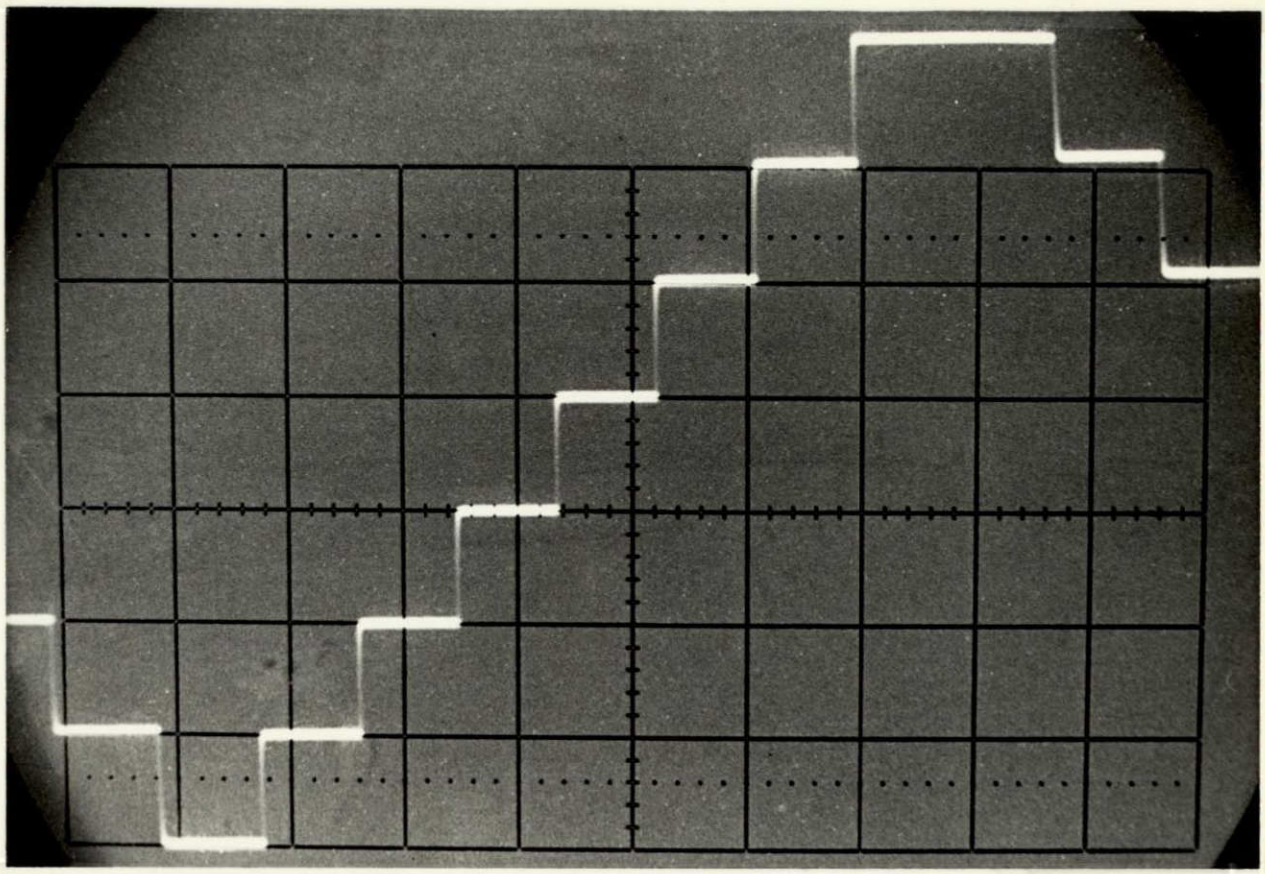
$$x(kT - iT) = 1 \text{ (logical)}$$

for all i's. It is a function of the voltage gain of the feedback amplifier, the voltage level associated with the logical state 1, and the number of stages used by the filter.

To check the linearity of the feedback amplifier in its range of operation, all the weights can be set to 1, and a sequence of m 0's followed by m 1's can be fed into the shift register connected in a circular configuration, m stands for the number of shift-register stages used by the filter. With a linear characteristic of the summer, the oscilloscope display should look like a staircase with a constant increase between each step. The experimental results are shown in Figure 4-23. The results indicate excellent amplifier linearity.

Another important characteristic of the summer of Figure 4-22 is a good frequency response at the clock frequency used when filtering input signals. Ideally, for an applied step input, we would like the output to rise instantly without any overshoot. Practically we are limited by a definite slew rate and a certain amount of overshoot, generally larger as the

NOT REPRODUCIBLE



Linear characteristics of the summer of figure 4-22

Figure 4-23

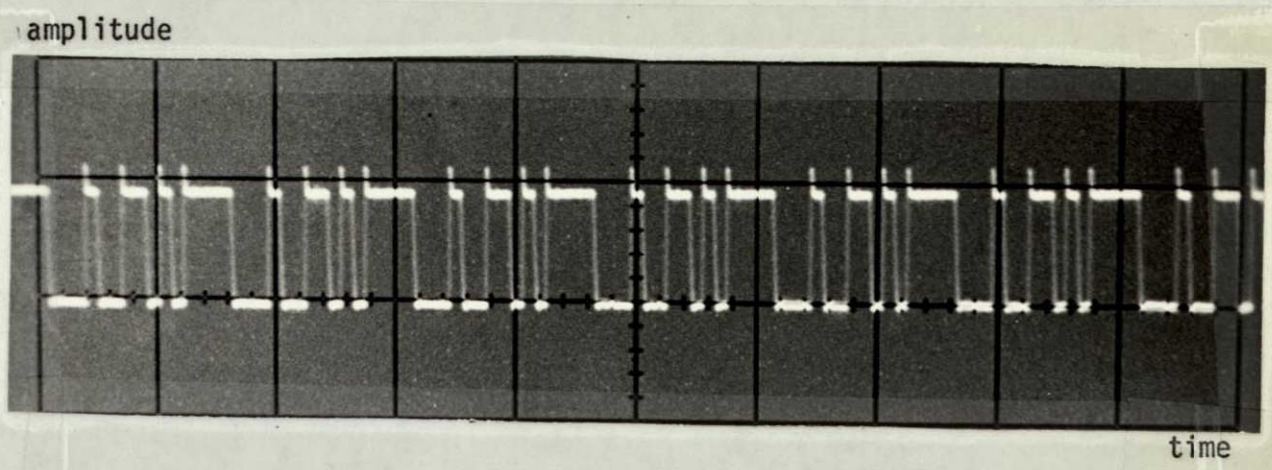
Clock period	333μS
Oscilloscope sweep time	.2mS/cm
Sensitivity	.2V/cm
Weight setting a_j=1 for all j's	

slew rate increases. The overshoot is eliminated by the output low-pass filter of Figure 4-22. Figure 4-24 shows the output of the four-stage pseudo-random sequence generator with clock periods of 3.33 μs. and 1 μs., without output filter. Experiments have shown that such overshoot adds a few low frequency components to the power spectrum of the pseudo-random sequence, and high frequency components outside the range of interest. The time constant of the output filter has been found by trial and error. Varying the time constant by regular steps; the overshoot was reduced to a point where the low frequency components due to overshoot reached a minimum. The final shape of a four-stage pseudo-random sequence is shown in Figure 4-25, at three different clock frequencies. From these results, we chose the clock frequency of 300 KHz. to run the experiments presented in the next section. It is the fastest clock frequency without serious distortion of the output.

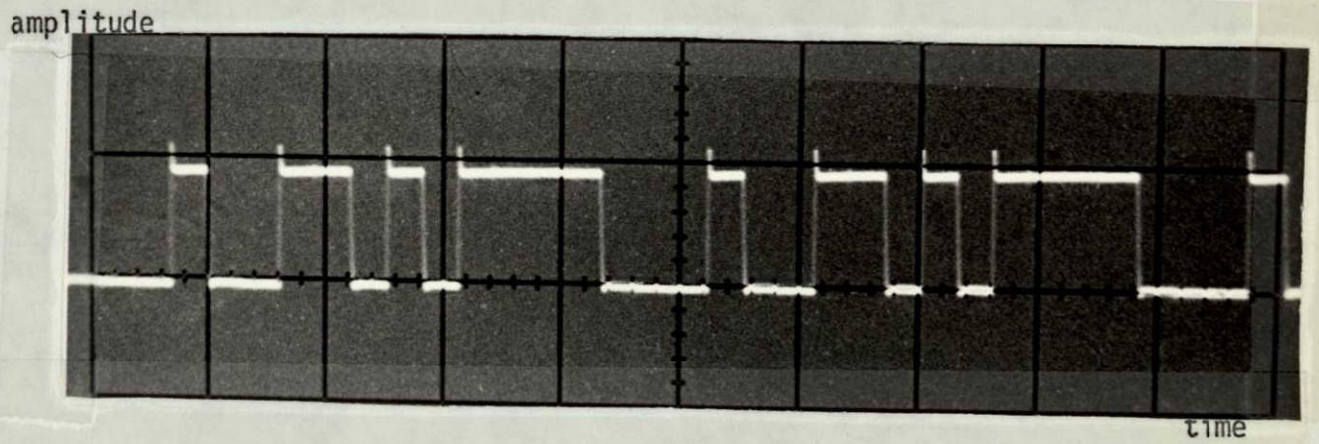
A close look at Figure 4-24 a will show a small fluctuation at each clock pulse when the output stays at the same level between pulses. This is due to a very high frequency oscillation of the output of the stages of the shift register at the clock pulse, when the output should not change level. This could be eliminated by the use of a better flip-flop to realize the shift register, or by a lowpass filter inserted between the output of the stages and the weights of the filter.

4.7 Design and Experimental Results of a Lowpass Digital Filter

In this section, we will present a method for finding the weights of a nonrecursive digital filter with m delays, design a lowpass filter, describe the statistical characteristics of the input to the filter (a 15-



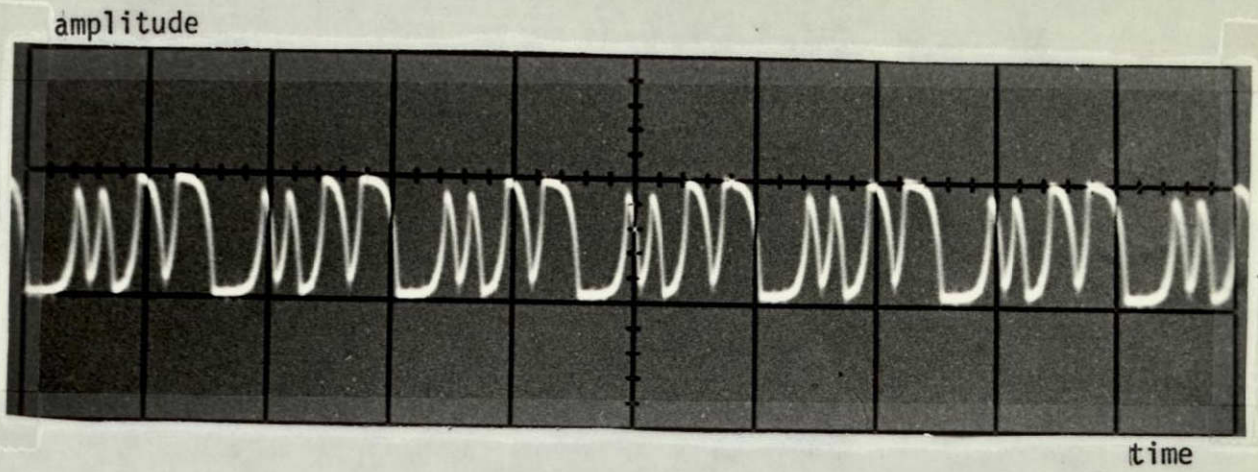
b) clock period of 1μS



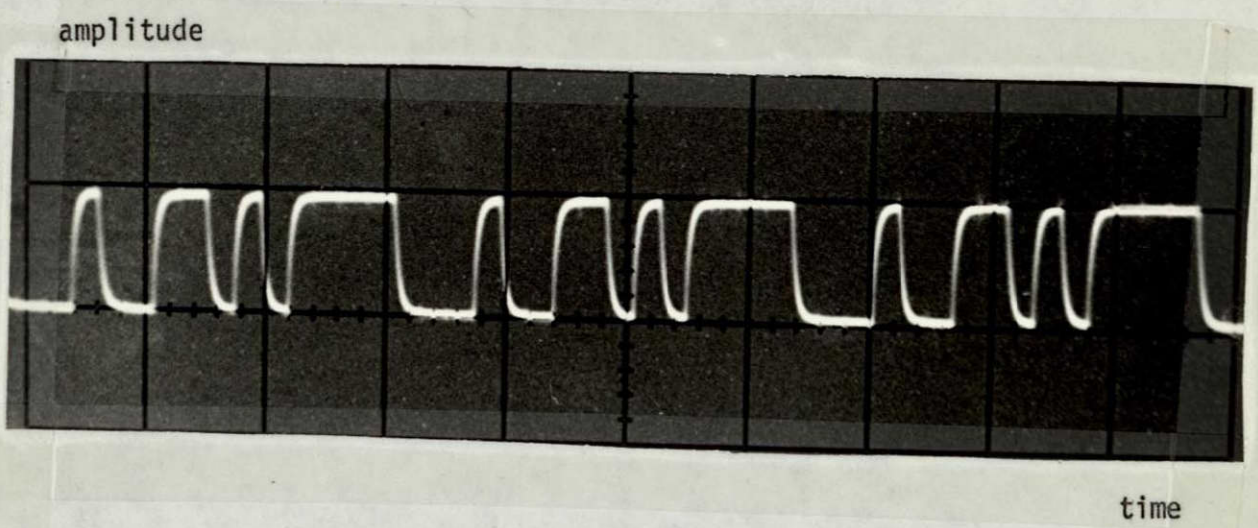
a) clock period of 3.33μS

Output of a 4-Stage Pseudo-Random Sequence Generator Without Output Filter

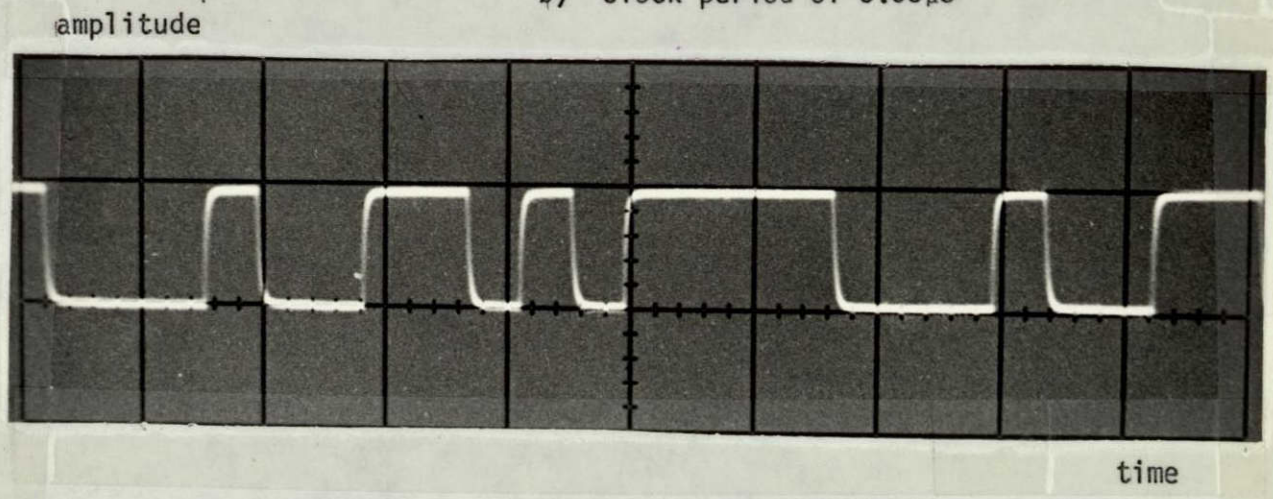
Figure 4-24



c) clock period of 1μS



b) clock period of 3.33μS



a) clock period of 10μS

Output of a 4-Stage Pseudo-Random Sequence Generator With Output Lowpass Filter

Figure 4-25

stage pseudo-random sequence), and finally present the experimental results of the filter implementation.

The frequency response S(wT) of a nonrecursive digital filter is given by

$$S(wT) = \sum_{n=-\infty}^{\infty} \beta_n \, e^{-jnwT} \tag{4-6}$$

Equation (4-6) is assumed periodic in wT and defined for $-\pi \leq wT \leq \pi$. The β_n's are the Fourier coefficients of the periodic function S(wT), and are given by

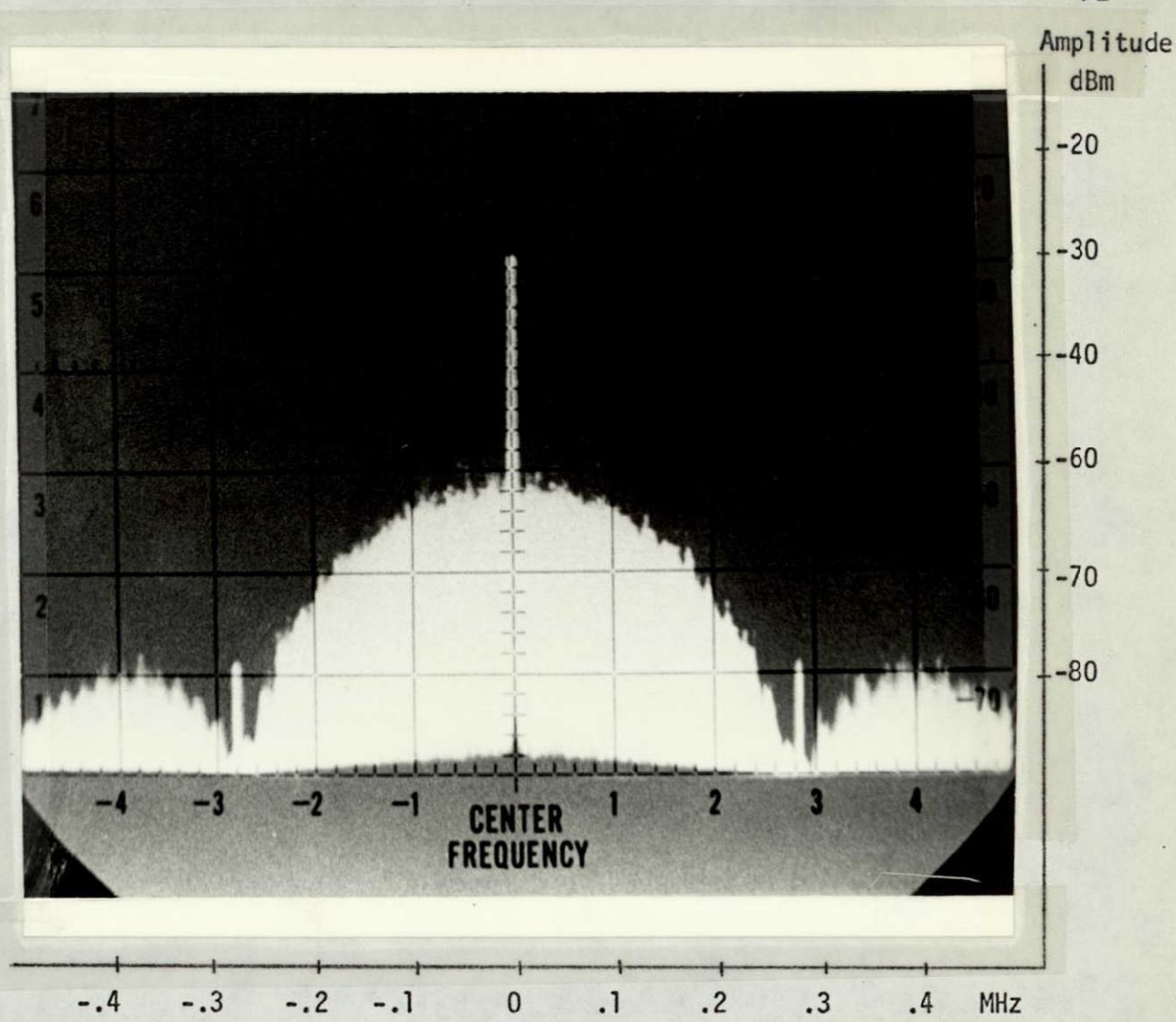
$$\beta_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(wT) \, e^{jwTn} \, d(wT) \tag{4-7}$$

Given a filter specified in the frequency domain by S(wT), the inverse Fourier transform of S(wT) will give the β_n's that describe the impulse response of the filter. As most periodic functions are exactly described only by an infinite number of Fourier harmonics, there will be,in most cases, an infinite number of β_n's. If we want to realize the desired frequency response by a nonrecursive digital filter which has a finite impulse response (it has only a finite number of delays), the infinite series of β_n's will have to be truncated, some β_n's made zero outside a given aperture. The implementation of a desired S(wT) on a nonrecursive digital filter will then be only an approximation of an ideal frequency response. The larger the number of delays, the more accurate will be the implementation.

To find the number of delays required for a desired accuracy, successive trials have to be made: first obtain the β_n's by taking the inverse Fourier transform of S(wT); truncate the series of β_n's according to a chosen aperture, and transform the truncated series to get a modified S'(w T), the

approximation to the ideal S(wT). If the modified S'(wT) is not accurate enough, try a different number of β_n's (go from m delays to m' delays), or change the position of the aperture, and repeat the process until results are obtained. (A weighting function, called a "window", can be used to modify the β_n's, improving the shape of S'(wT). Examples of these are the Hanning window, the Hamming window, the Blackman window, etc. The use of a weighting function is neither discussed nor applied here.)

The Fast Fourier Transform (FFT) is of great help in determining the number of delays and the weights of the filter. The function S(wT) is put into sampled form, using M samples. For use with the FFT, the number M should be a power of 2. We give an example of the method by realizing a lowpass filter with m = 7 (the number of delays in Figure 4-22). The input to the filter will be a pseudo-random sequence of length $2^{15} - 1$, that has a power spectrum given by equation 2-19, and shown in Figure 4-26. (Figures 4-27 and 4-28 are further illustrations of equation 2-19. They are the power spectra of the output of a four-stage generator, with two different scales.) We would like to filter out all the frequency components of the first lobe higher than $f_c/3$, with f_c the clock frequency of 300 KHz. As S(wT) is periodic for a digital filter, the desired frequency characteristics S(wT) of the filter will be as shown in Figure 4-29. Part of the power spectrum of the first lobe of the shape Sin x / x input is shown in Figure 4-30. The cutoff frequency of the desired filter is indicated by f_o. The frequency response of the ideal lowpass filter for $-\tau \leq wT \leq \tau$ is shown in Figure 4-31a) in sampled form; its Inverse Fast Fourier Transform (IFFT) is given by b) of the same figure. As the hardware implementation has only 7 stages (Figure 4-22), we chose an aperture of $7\Delta T$ (when $\Delta T = 1/f_c$), centered about β_o . Making all the other β_n's zero (Figure 4-31 c)) and taking the FFT

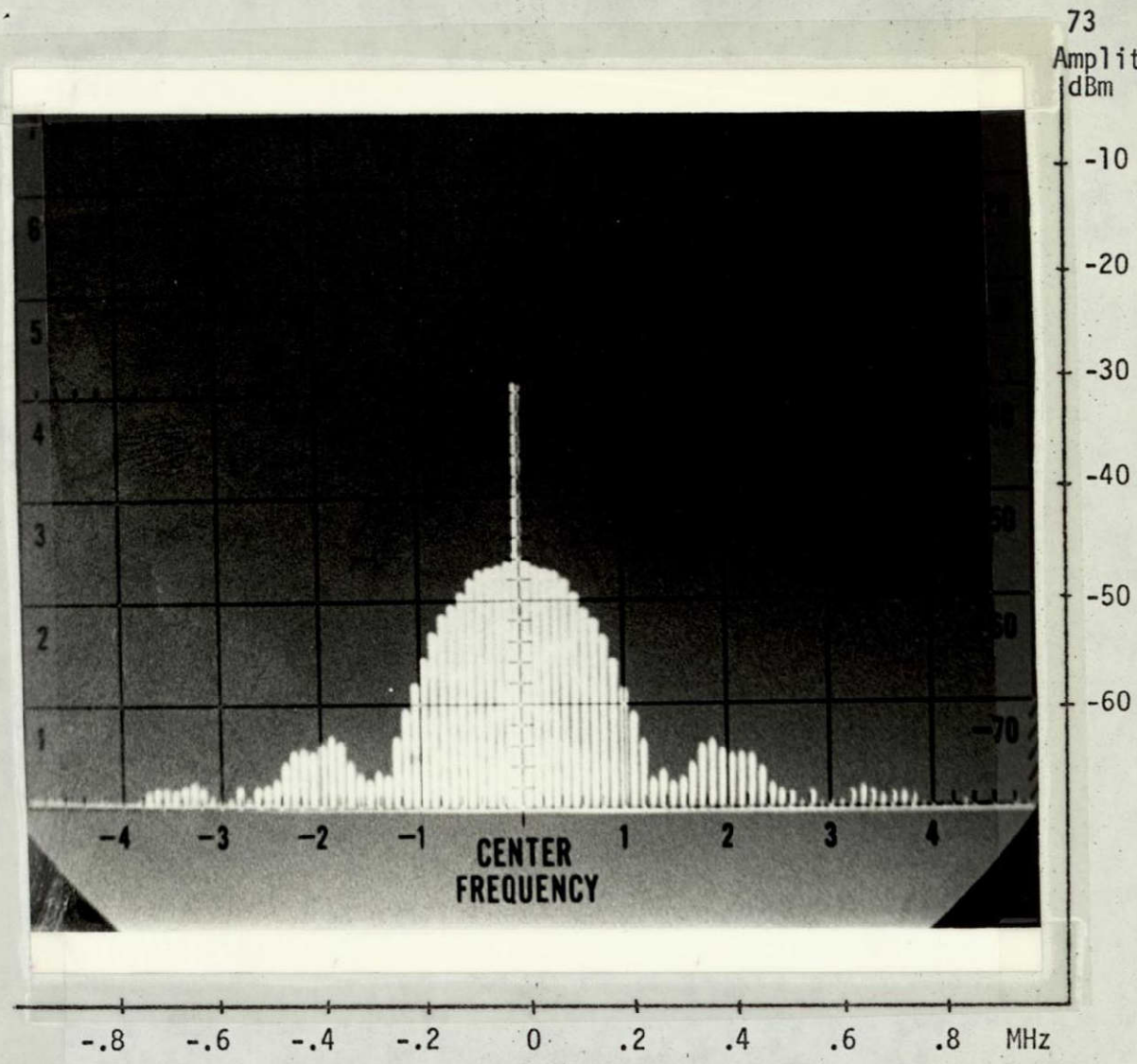


Power Spectrum of the Output of a 15-Stage Pseudo-Random Sequence Generator

Figure 4-26

Clock period	3.3μS
Frequency scale	.1 MHz/div.
Bandwidth	1 KHz
Scan time	2 S/div
Log scale	
Log reference level	0 db
Atenuation	0

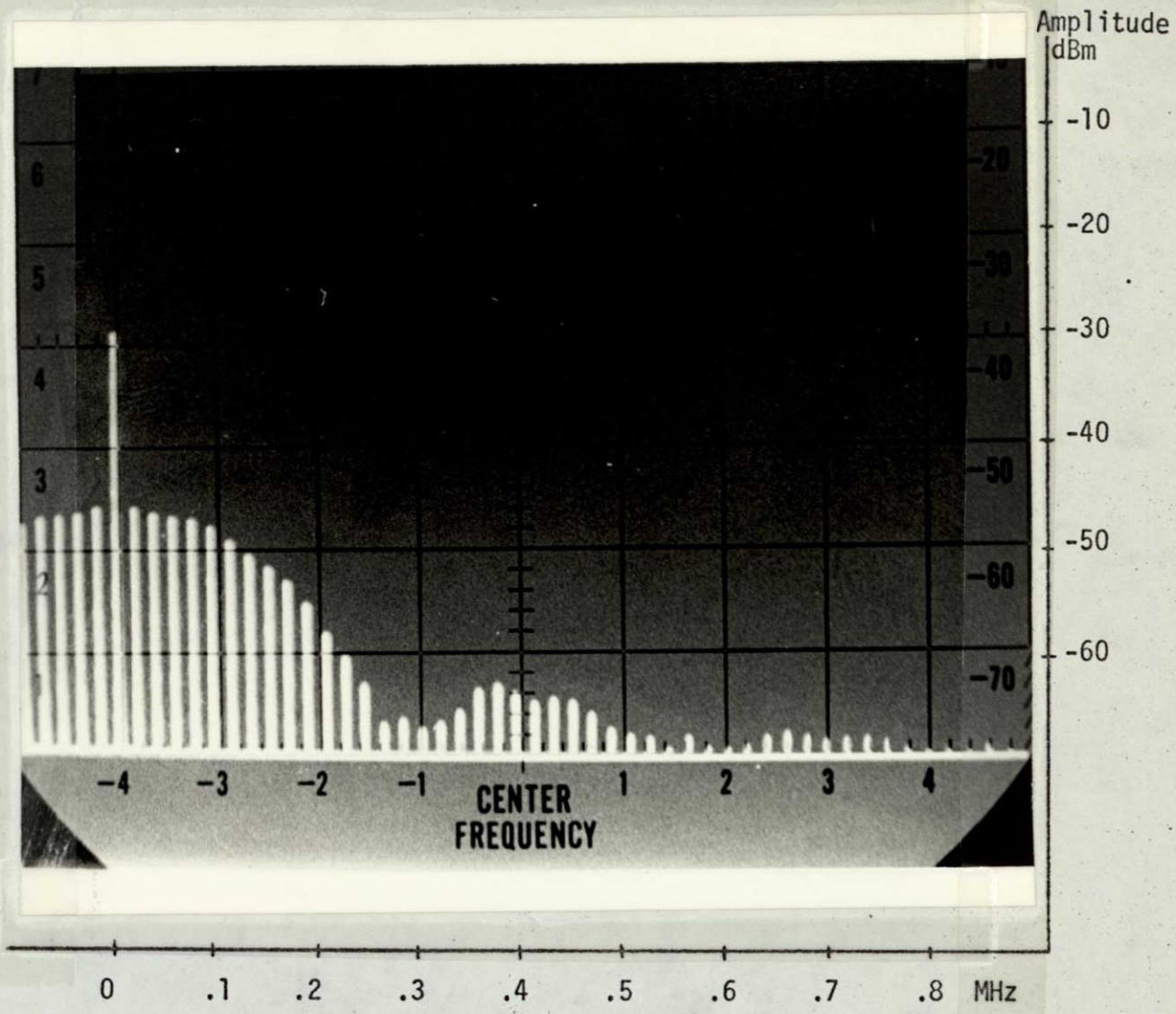
NOT REPRODUCIBLE



Power Spectrum of a Four-Stage Pseudo-Random Sequence Generator

Figure 4-27

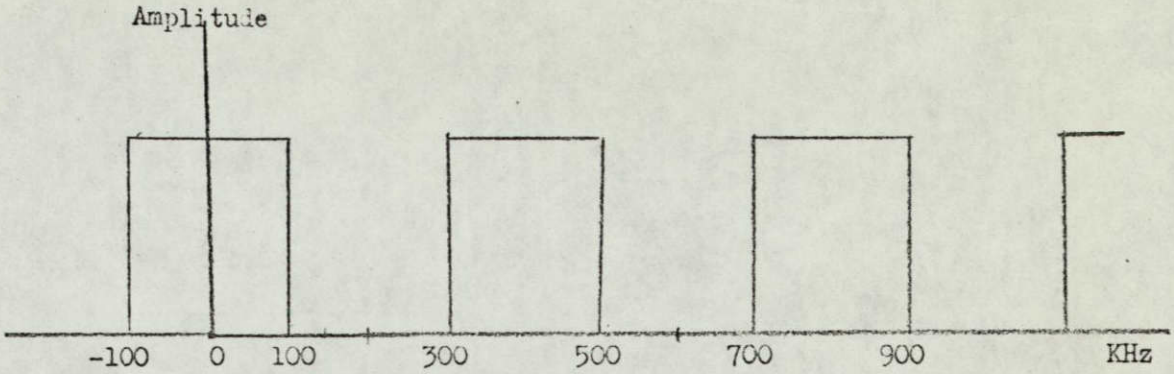
Clock period	3.3μS
Frequency scale	.2 MHz/div.
Bandwidth	.3 KHz
Scan time	.5 S/div.
Log scale	
Log reference level	10 dBm
Atenuation	0



Power Spectrum of a 4-Stage Pseudo-Random Sequence Generator

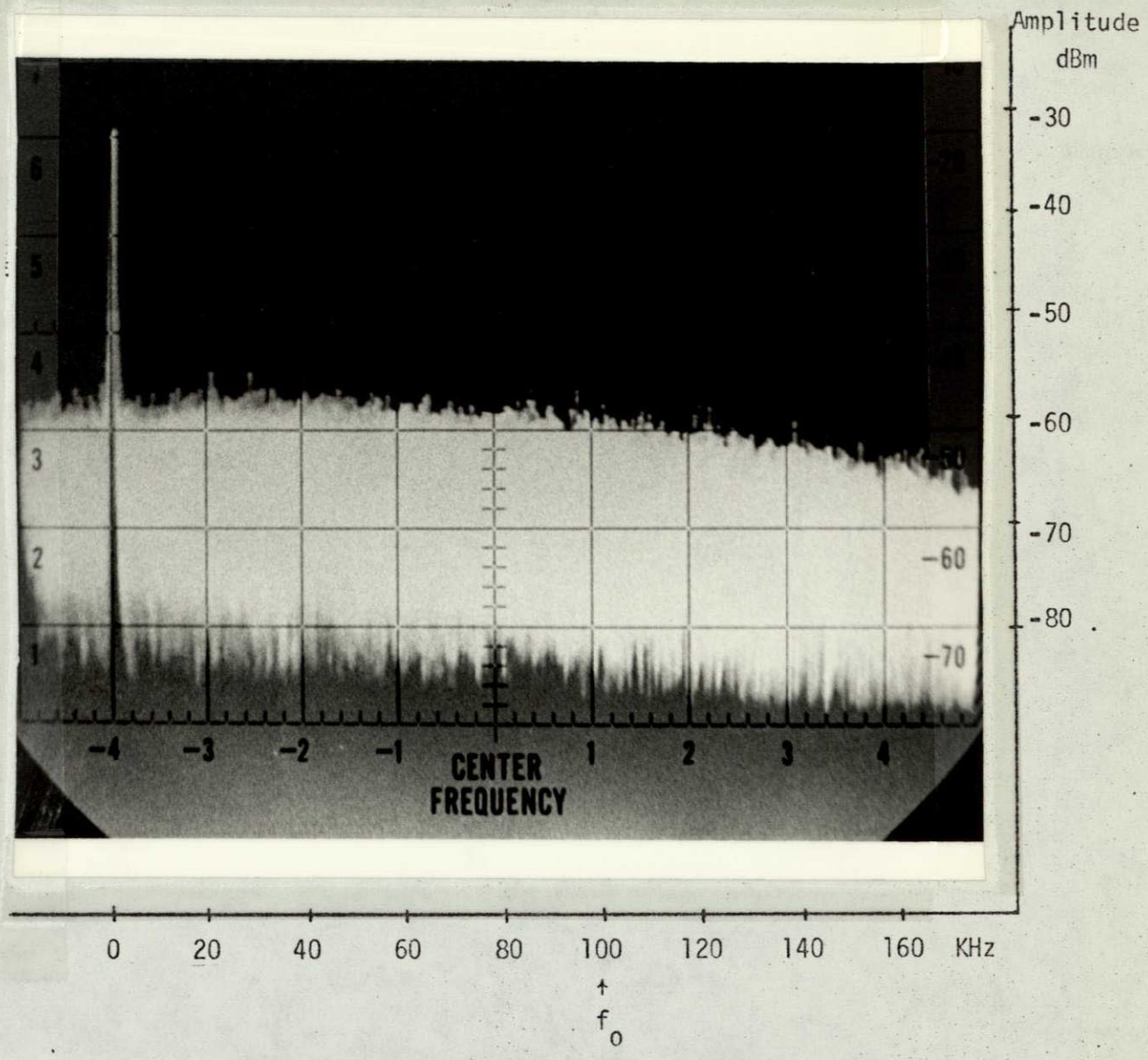
Figure 4-28

Clock time	3.3μS
Frequency scale	.1 MHz/div.
Bandwidth	.3 KHz
Scan time	.5 S/div.
Log Scale	
Log reference level	10 dBm
Atenuation	0



Desired frequency response of the lowpass filter

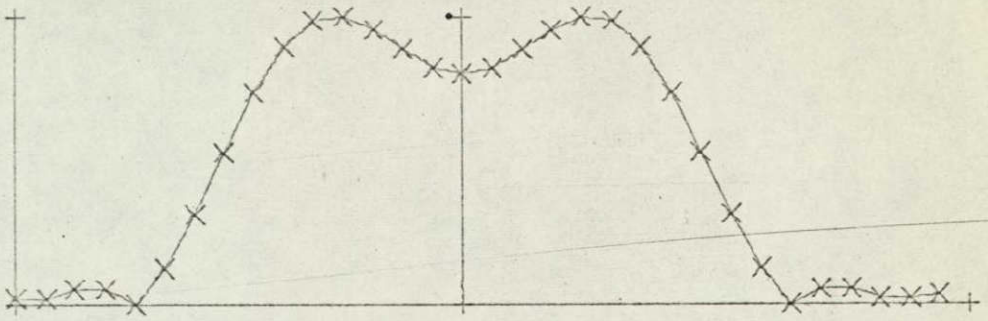
Figure 4-29



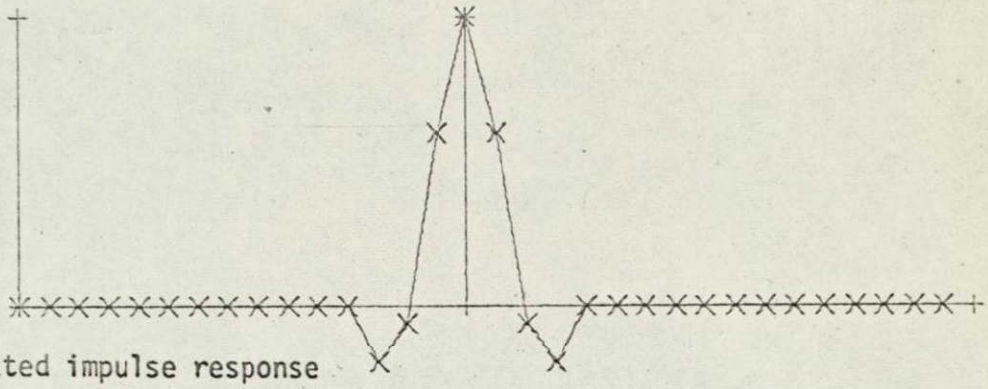
Power spectrum of the input to the lowpass filter

Figure 4-30

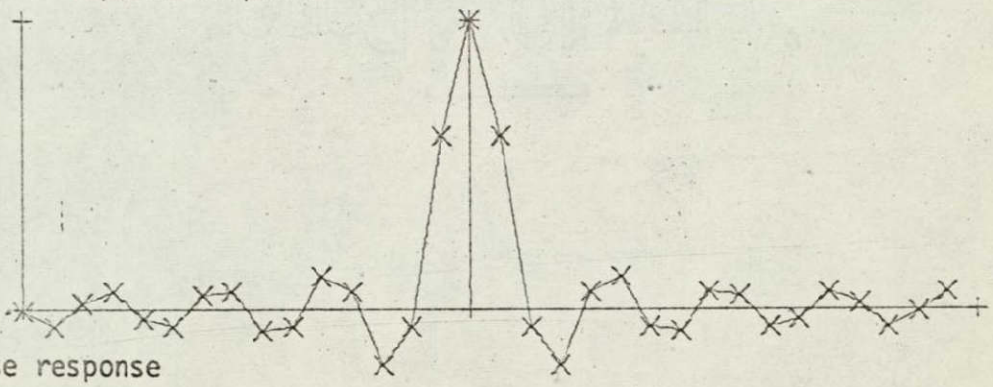
Clock period	3.3μS
Frequency scale	20 KHz/div
Bandwidth	.3 KHz
Scan time	.2 sec/div
Log scale	
Log reference level	-10 dBm
Atenuation	0



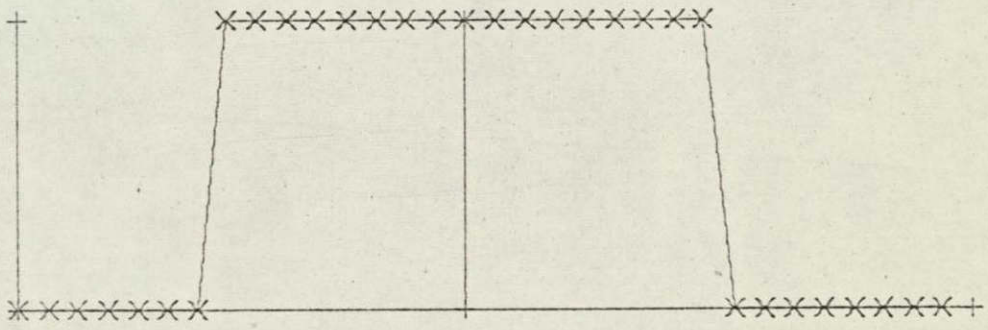
d) Approximated frequence response



c) Truncated impulse response



b) Impulse response



a) Desired frequency response

Steps for finding the β_n's of the filter

Figure 4-31

of the truncated series, we get the frequency characteristics of the filter shown in Figure 4-31 d), which is an approximation of the ideal characteristics of a) in the same figure. Assuming that this approximation is accurate enough for our purpose, we should realize this filter by implementing the following weights:

$a_1 = -.194$
$a_2 = -.059$
$a_3 = .597$
$a_4 = 1.0$
$a_5 = .597$
$a_6 = -.059$
$a_7 = -.194$

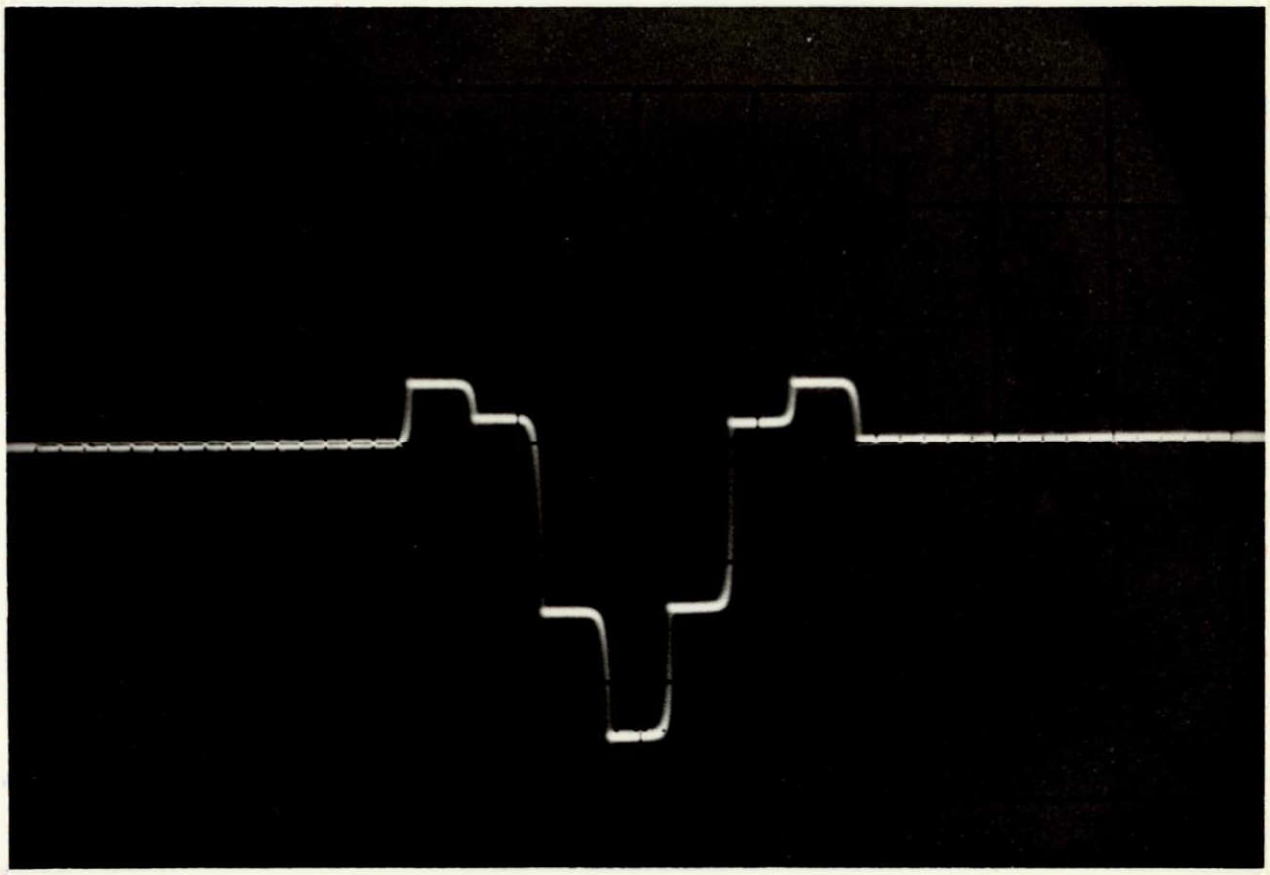
After setting of the variable resistors of Figure 4-22, the impulse response of the filter can be checked by circulating a 1 in the shift register, all other stages being in the state 0. This impulse response is shown in Figure 4-32. The approximation to a $\frac{\sin x}{x}$ form is apparent.

The power spectrum of the digitally-filtered pseudo-random sequence is shown in Figure 4-33. The scaling is identical to the one in Figure 4-28, which is the input to the filter. Figure 4-34 gives a better idea of the periodicity of a digital filter. It shows the output of the filter for about 5 cycles. The input to the filter corresponding to the same is shown in Figure 4-26. For curiosity, we have shown in Figure 4-35 the output of the filter in the time domain for a short part of the long periodic sequence.

4.8 Software Simulation of a Lowpass Digital Filter

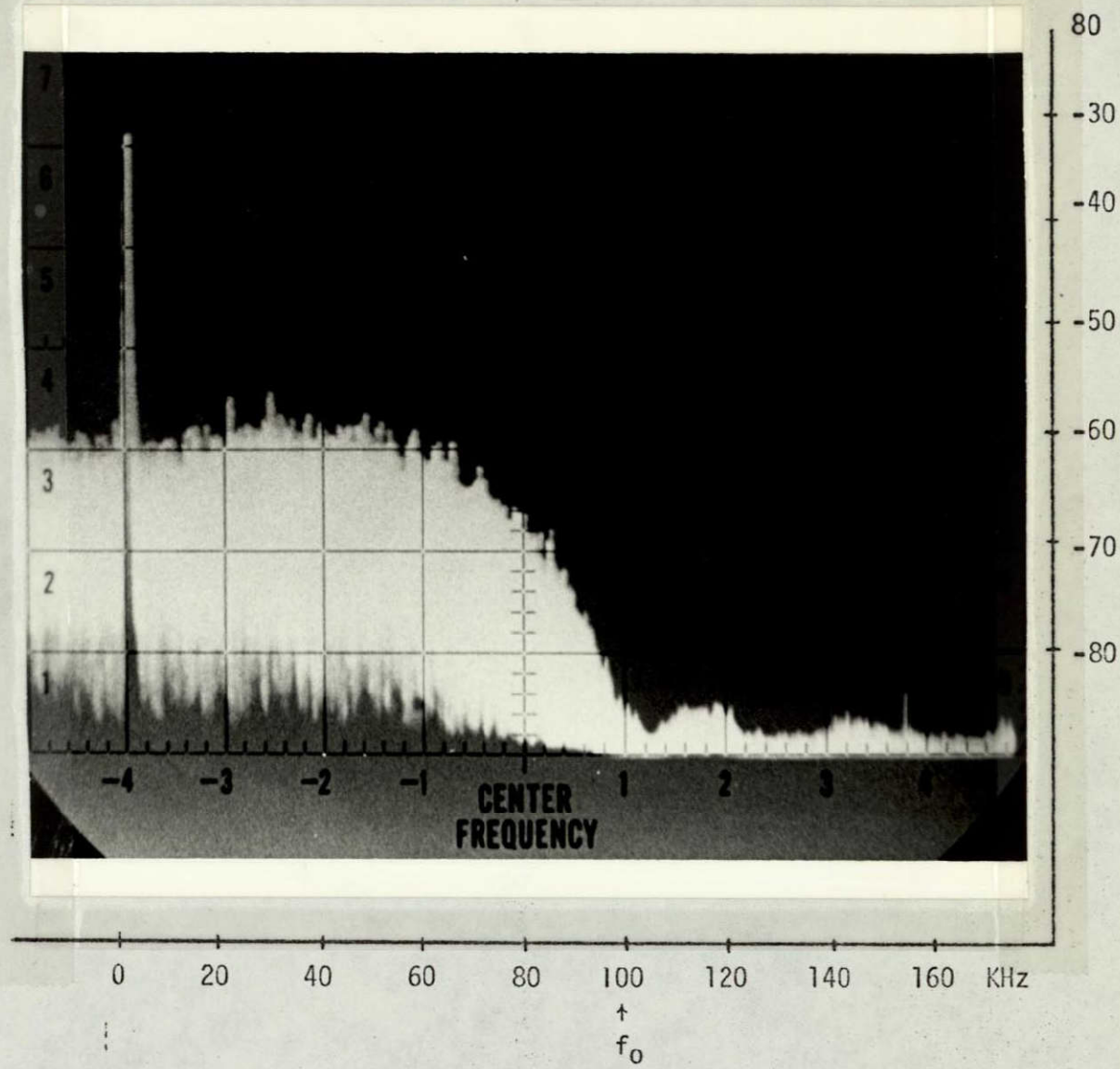
Finally, we present a simulation on the IBM 1130 of the lowpass nonrecursive digital filter presented above.

NOT REPRODUCIBLE



Impulse response of the lowpass nonrecursive digital filter

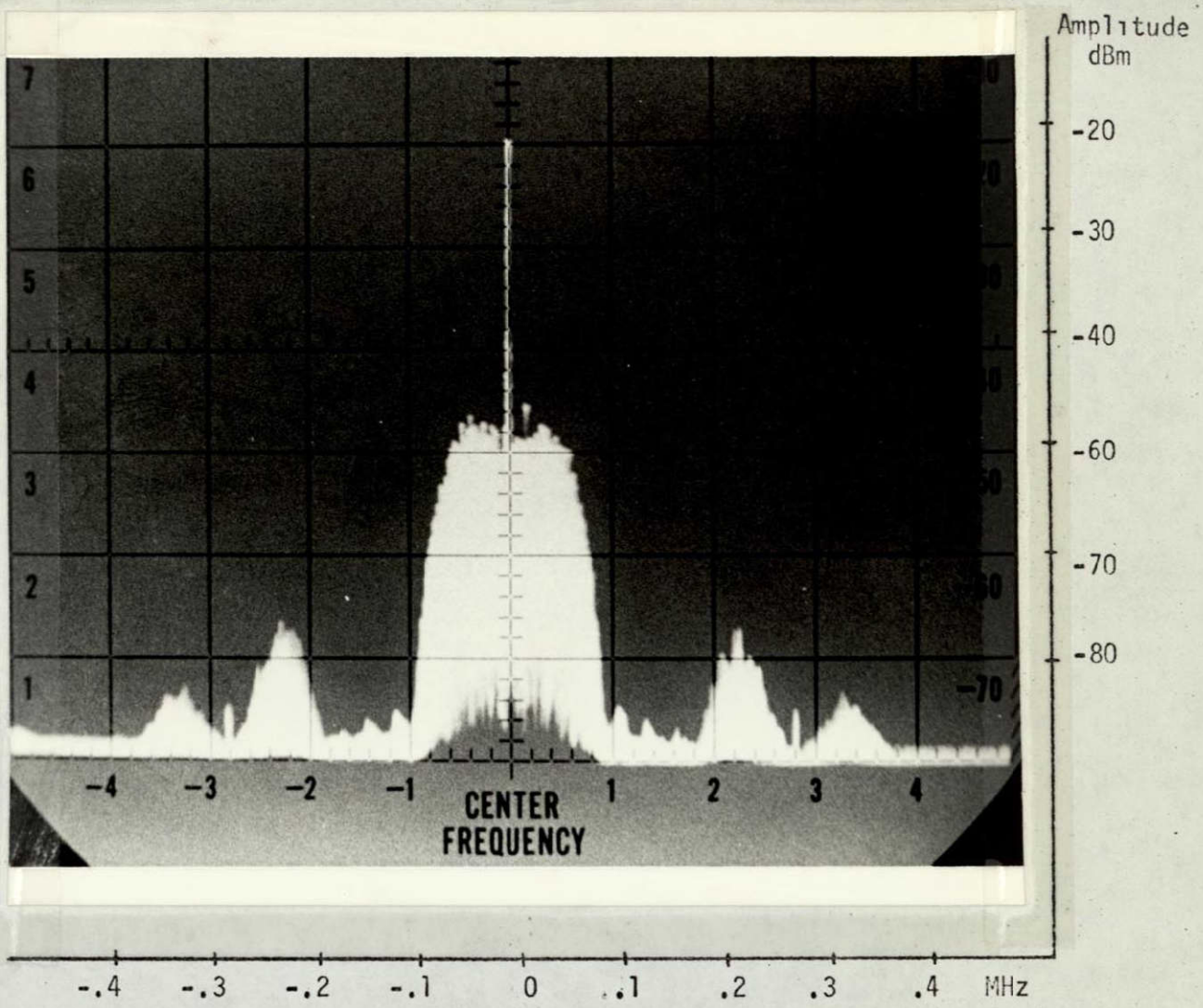
Figure 4-32



Power spectrum of the output to the lowpass filter

Figure 4-33

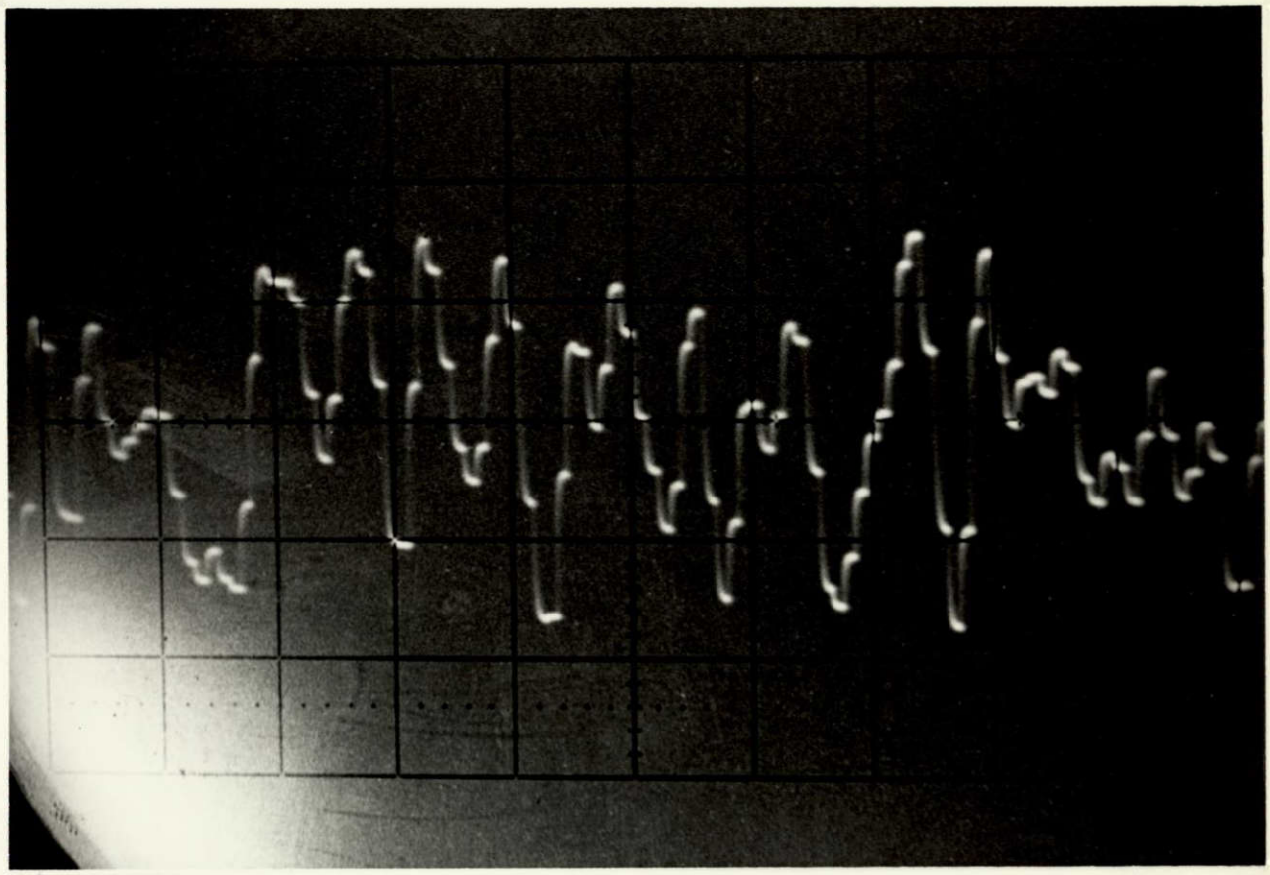
Clock period	3.3μS
Frequency scale	20 KHz/div
Bandwidth	.3 KHz
Scan time	.2 sec/div
Log scale	
Log reference level	-10 dBm
Atenuation	0



Power spectrum of the filtered output of a 15-stage pseudo-random sequence generator

Figure 4-34

Clock period	3.3μS
Frequency scale	.1 MHz/div.
Bandwidth	1 KHz
Scan time	2 S/div.
Log scale	
Log reference level	0db
Atenuation	0



Time response of the lowpass nonrecursive digital filter

Figure 4-35

A Fortran program has been written containing three main parts: generation of the input pseudo-random sequence, simulation of the filter, and Fast Fourier Transform of the output. Because of the limitation in the computer memory size available (16,000 words of 16 bits), we could not simulate the complete output of a 15-stage noise generator. Instead we used a 9-stage shift register (with states .5 or -.5 to avoid D.C. component in the power spectrum), with a sequence of 512 terms. The change in the number of stages used has the effect of increasing the distance between the power spectral lines, without affecting the sin x/x envelope. The power spectrum of the input is shown in Figure 4-36.

The output of the filtered sequence is shown in Figure 4-37. This simulated result agrees very closely with the experimental results.

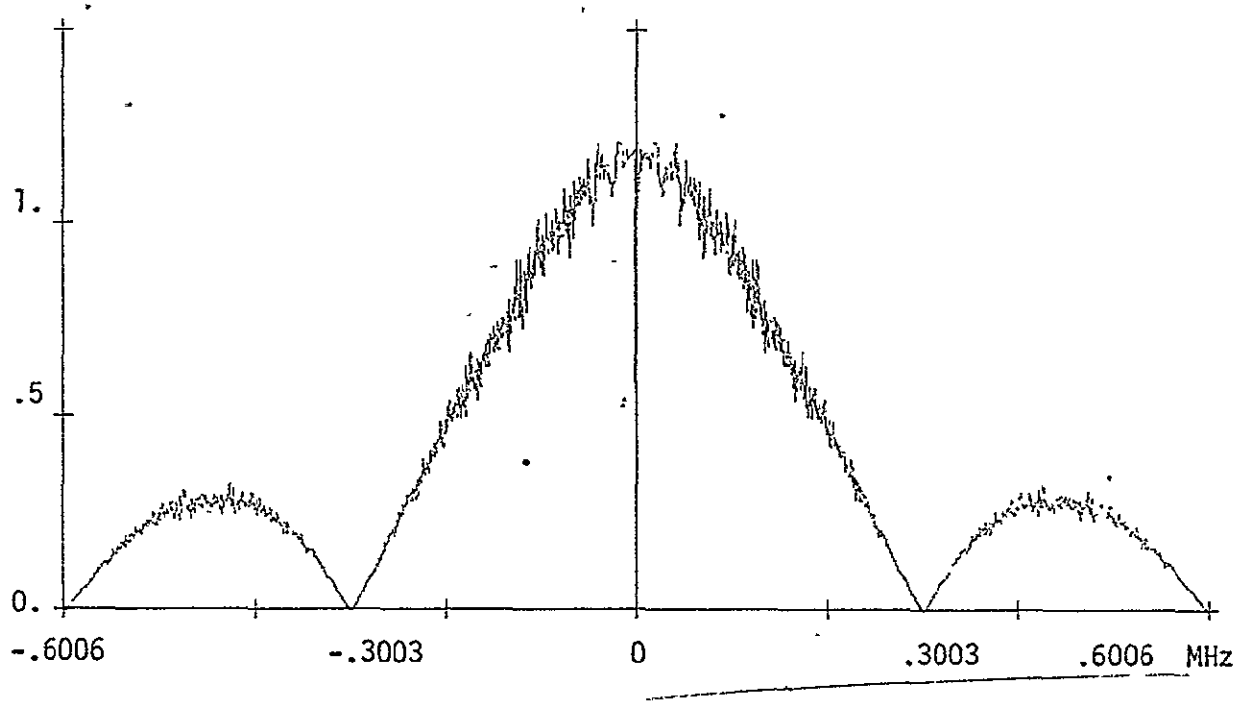


Figure 4-36

Power spectrum of the output of a 9-stage generator

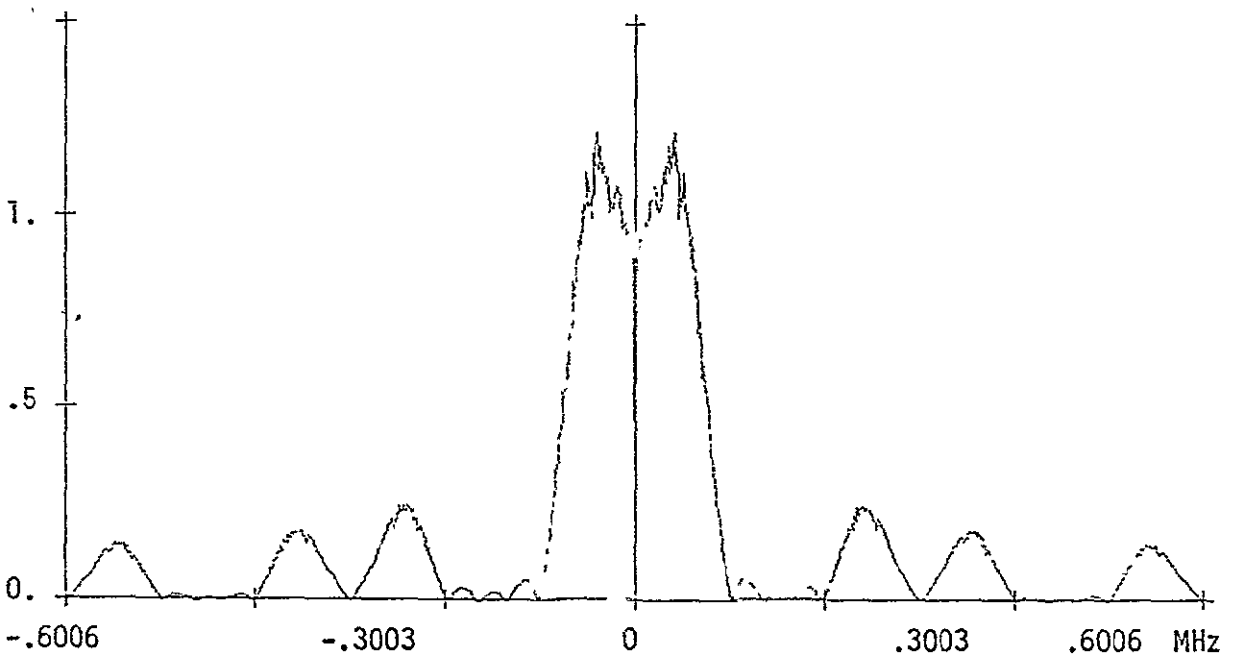


Figure 4-37

Power Spectrum of the output of the filter

5.0 Generation of Partition Numbers

This section describes an interesting and unexpected approach to the generation of partition numbers and some other sequences through the use of convolution, and digital filtering techniqes.

5.1 Partition Numbers and Convolution

A partition of a positive integer is the expression of the integer as a sum of positive integers. For example, the integer 4 has 5 partitions.

$4 = 4$

$4 = 3 + 1$

$4 = 2 + 2$

$4 = 2 + 1 + 1$

$4 = 1 + 1 + 1 + 1$

A change in order is not considered to lead to a new partition. The number of partitions of the integer n shall be called the "partition number" here, and denoted $p(n)$. From the above example, $p(4) = 5$. It is well known (see for example, Alder [16]) that a generating function for $p(n)$ is given by

$$f(x) = \frac{1}{\prod_{\nu=1}^{\infty} (1 - x^{\nu})}, \quad |x| < 1 \tag{5-1}$$

An alternative to this multiplication procedure for obtaining $p(n)$ is suggested by the convolution theorem of operational calculus.

Consider the following expansion of equation (5-1)

$$f(x) = \frac{1}{1-x} \cdot \frac{1}{1-x^2} \cdot \frac{1}{1-x^3} \cdots$$

$$= (1 + x + x^2 + \cdots) \cdot (1 + x^2 + x^4 + \cdots)$$

$$\cdot (1 + x^3 + x^6 + \cdots) \cdots \qquad (5\text{-}2)$$

The p(n) are the coefficients of the x^n terms in the infinite product. The sums in parentheses in equation (5-2) are each in the form of a z-transform of a sequence. The sequences corresponding to the first three sums are:

$$\{ 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; \cdots \}$$

$$\{ 1 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 0 \;\; \cdots \}$$

$$\{ 1 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; \cdots \}$$

$$\cdots\cdots\cdots$$

Further terms have associated sequences with progressively more zeros between ones. Since equation (5-2) is essentially a product of z-transforms, we know that f(x) is the z-transform of a sequence which is the convolution of the sequences associated with the sums on the right in (5-2). This follows from the convolution theorem. That is:

$$F(\omega) = \{ 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; \cdots \} * \{ 1 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; \cdots \} *$$

$$\{ 1 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; \cdots \} * \cdots$$

where $F(\omega)$ is the z-transform of f(x) and * indicates discrete convolution (See Healy [18]) . ω is the set of positive integers associated with the terms in the sequences, in ascending order from 0. At first glance it might seem that an infinite numbers of convolutions need be carried out to obtain values of P(n), which are just the terms in $F(\omega)$. But actually, since the kth sequence has the form:

$$\{\ 1\ \underbrace{0\ 0\ \cdots\ 0\ 0}_{k-1}\ 1\ 0\ 0\ \cdots\ \}$$

convolution by this and higher order sequences will reproduce the first k terms in $F(\omega)$ without change. Hence to obtain the first k terms in $F(\omega)$, that is the first k numbers of partitions $p(n)$, it is only necessary to convolve the first k sequences. Fortunately, this operation is easily done in tabular form as shown in table 5-1. The table is carried out far enough to generate the first 9 values of $p(n)$, which appear as the first nine numbers in the 9th (bottom) row.

The table is generated in the following way. Write a sequence of ones as the top line. To get a number in the next row, add the number above the desired number to the second number to the left plus the fourth number to the left, etc. until you reach the left. To get any number in the jth row, add the number above it in the j-1 row, to the number j units to the left in the j-1 row, to the number 2j units to the left, etc. until you reach the left. (e. g. in the third row, 12 = 5 + 4 + 2 + 1). The above procedure simply carries out the process of successive discrete convolution.

Table-5-1 has some additional information. Consider the diagonal rows (+ 45°) . For example, the 6th diagonal row from the upper left is { 1 3 3 2 1 1 } , reading from upper right to lower left. The sequence gives the number of partitions of 6 into μ parts where μ is the order of the numbers in the sequence. Similarly, the mth diagonal row gives the number of partitions of m into μ parts as a sequence of values corresponding to $\mu = 1, 2, 3, \cdots$. The number of partitions of m into parts the largest of which is μ is of course given by the inverse sequence as suggested by Alder's [16] theorem 2. A similar procedure can be used to find numbers of partitions into even parts or odd parts or a number of other possible forms.

5.2 Digital Filter Generation

The above convolution of sequences suggests another viewpoint, or method of generating $p(n)$. The sequences of ones, separated by k zeros, is simply the discrete impulse [$\{ 1 \quad 0 \quad 0 \quad 0 \quad \cdot \quad \cdot \quad \cdot \}$ for discrete systems] response of a one feedback-stage recursive digital filter with $k + 1$ units of delay. This is illustrated in figure 5-1. Hence the necessary convolution of sequences with increasing zeros-spacing (value of k) can be obtained as the discrete impulse response of a cascade of filters with increasing k, as shown in figure 5-2.

The output of the first stage in figure 5-2 is the first line in table 5-1, the output of the second stage is the second line, etc. It is apparent from the argument in the preceding section that to obtain the first k $p(n)$ it is necessary to cascade k filters.

5.3 Generation of Other Sequences

It should be apparent from the above that there are many other sequences which can be generated by digital filters. One example is the Fibonacci sequence in which each number is the sum of the previous two numbers in the sequence. The sequence begins as:

$$1 \quad 1 \quad 2 \quad 3 \quad 5 \quad 8 \quad 13 \quad 21 \quad 34 \quad 55 \quad \cdot \quad \cdot \quad \cdot$$

This sequence has a generating function:

$$G(x) = \frac{1}{1 - x - x^2} \tag{5-3}$$

From the fundamental property of the sequence, as expressed above, or from the generating function, we deduce that the sequence can be generated by a two stage recursive digital filter as shown in figure 5-3. This

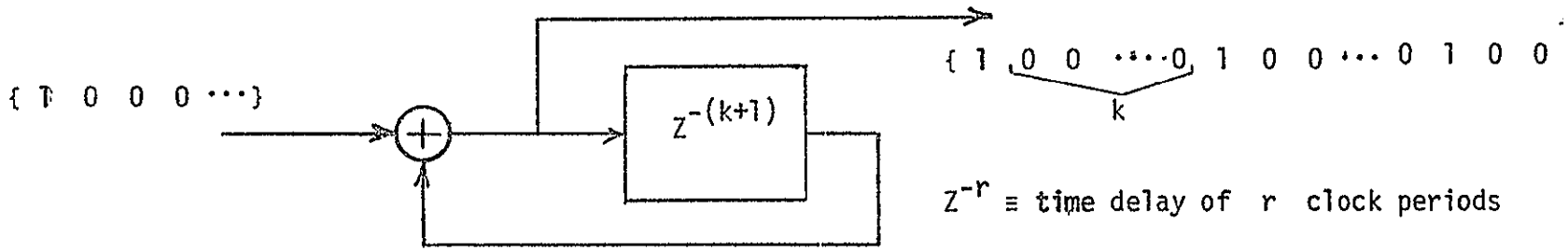


Figure 5-1. Recursive Digital Filter Stage (RDFS)

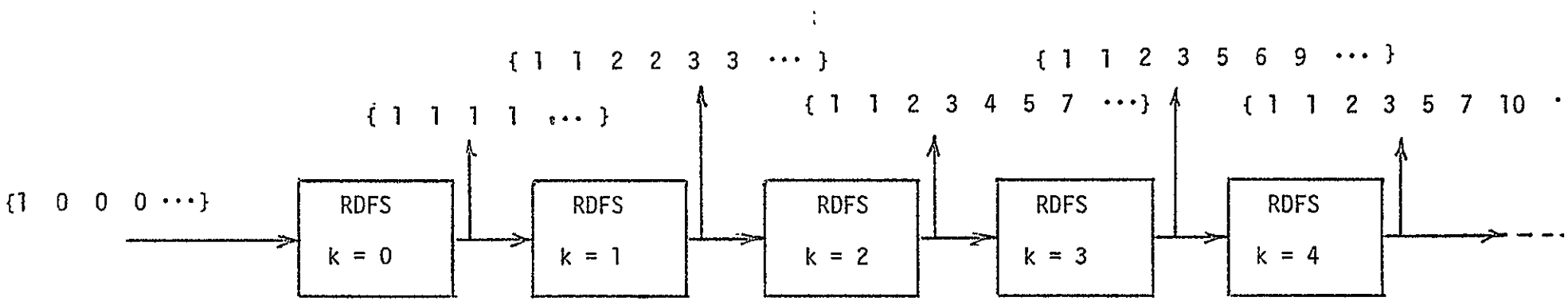


Figure 5-2. Digital Filter Partition Number Generator

1	1	1	1	1	1	1	1	1	1	.	.	.
1	1	2	2	3	3	4	4	5	5	.	.	.
1	1	2	3	4	5	7	8	10	12	.	.	.
1	1	2	3	5	6	9	11	15	18	.	.	.
1	1	2	3	5	7	10	13	18	23	.	.	.
1	1	2	3	5	7	11	14	20	26	.	.	.
1	1	2	3	5	7	11	15	21	28	.	.	.
1	1	2	3	5	7	11	15	22	29	.	.	.
1	1	2	3	5	7	11	15	22	30	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.

Table 5-1

circuit adds the number delayed by one time period (Z^{-1}) to the number delayed by two time periods (Z^{-2}) to form the new output.

It is apparent that many other sequences can be generated by other feedback schemes.

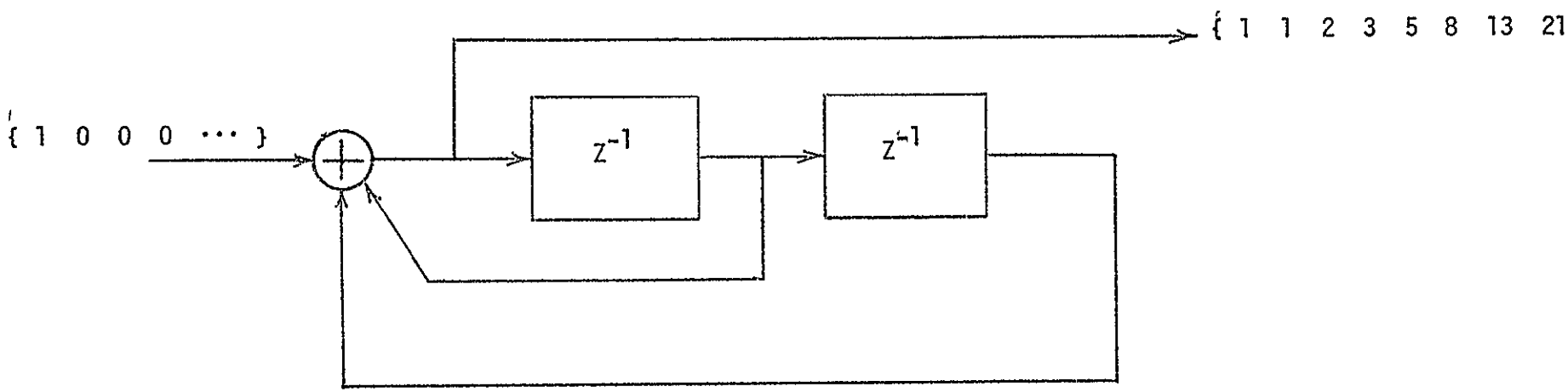


Figure 5-3. Fibonacci Sequence Generator

BIBLIOGRAPHY

1. Healy, T. J.: "Properties of Sums of Pseudo-Random Variables in Feedback Shift Registers," Final Report on NASA-Ames-Santa Clara Project 008, June 1, 1969, CR-73356.

2. Lindholm, J. H.: "An Analysis of the Pseudo-Randomness Properties of Subsequences of Long m-Sequences", IEEE Trans. on Inf. Th., Vol IT-14, No. 4, July, 1968, pp. 569-577

3. Davies, A. C.: "Probability Distributions of Noiselike Waveforms Generated by a Digital Technique," Electronics Letters, Sept. 20, 1968, pp. 421-423

4. Healy, T.J.: "The Synthesis of Distributions of Pseudo-Random Variables," Second Asilomar Conference on Circuits and Systems, Pacific Grove, California, October 30-November 1, 1968

5. Davies, A. C.: "Probability-Density Functions of Digitally Filtered m-Sequences." Electronic Letters, May 15, 1969, pp. 222-224

6. Douce, J. L., and Healy, T. J.: "Evaluation of the Amplitude Distribution of Quasi-Gaussian Signals Obtained from Pseudo-Random Noise," IEEE Transactions on Computers, Vol. C-18, No. 8, August, 1969, pp. 749-752

7. Healy, T. J.: "Forms of Probability-Density Functions Obtainable from Summed m-Sequences," Electronic Letters, Nov. 27, 1969, pp. 624-625

8. Davies, A. C.: "Probability-Density-Functions of Summed m-Sequences," Electronic Letters, Feb. 19, 1970, pp. 89-90

9. Golomb, S. W. et al: Digital Communications with Space Applications, Prentice-Hall, Englewood Cliffs, New Jersey, 1964

10. Douce, J. L., Private communication

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

12. Wolf, J. K.: "On the Application of Some Digital Sequences to Communication," IEEE Trans. on Comm. Sys., Dec., 1963, pp. 422-427

13. Cramer, R. "The Probability Distributionsof Certain Sums of Random Variables," MSEE Thesis, University of Santa Clara, Santa Clara, Calif., February 1970

14. Halmos, P. "Measure Theory," D. Van Nostrad, Princeton, 1961

15. The Digital Logic Handbook, The Digital Electronics Corporation, Maynard, Mass., 1967

16. Alder, H. L.: "Partition Identities-From Euler to Present", The American Mathematical Monthly, Vol. 76, No. 7, Aug-Sept. 1969, pp. 733-746

17. Healy, T. J.: "The Feedback Shift Register as an Educational Tool," Submitted to the IEEE Proceedings - Special Issue on Education

18. Healy, T. J.: "Convolution Revisited," IEEE Spectrum, April, 1969, pp. 87-93

19. Kac, M. : Statistical Independence in Probability, Analysis and Number Theory, John Wiley and Sons, Inc., New York, 1959