A Digital Hardware Realization of a Random Number Generator

JOHN L. PERRY, RONALD W. SCHAFER, and LAWRENCE R. RABINER

Abstract—A digital random number generator that generates number sequences with either uniform or Gaussian distributions has been constructed in TTL digital hardware for use as a peripheral device to a DDP-516 computer. The hardware design is based on an algorithm described by Rader, Rabiner, and Schafer [1]. Approximations to the Gaussian distribution are generated by summing $NA+1$ consecutive numbers from the uniform sequence, where NA (the number of additions) is specified by the user and is in the range $0 \leq NA \leq 213$. Measurements of its autocorrelation function, power spectrum, and amplitude histograms indicate that the generator produces uncorrelated samples with amplitude distributions that are very close to the desired uniform or Gaussian distributions. For generating Gaussian numbers, the hardware generator operates at least 600 times faster than an equivalent software realization.

Introduction

Random numbers are used extensively in many computer simulations. For example, a random signal is required for unvoiced excitation in speech synthesizers and as a stimulus in many psychoacoustic experiments. Random numbers are also essential for Monte Carlo simulations of perceptual processes and neural processing models. The pseudorandom number sequences that are used in such applications are typically generated by some iterative arithmetic process that is chosen to guarantee a large period of repetition and suitable statistical properties. A major disadvantage with such sequences is that the computations required for their generation are generally quite time consuming. For this reason, we have constructed a hardware digital random number generator that operates as a peripheral device to a DDP-516 laboratory computer. This device provides *reproducible* pseudorandom number sequences that approximate either uniform or Gaussian amplitude distributions.

Description of the Noise Generator

The basic subsystem in the hardware design is a uniform number generator based on an algorithm described by Radar *et al.* [1]. Each bit of an L-bit random number X_n is derived by an EXCLUSIVE-OR operation on a pair of corresponding bits of the two previous numbers X_{n-1} and X_{n-2}. These bits are not stored in the bit positions from which they were produced, but instead each new bit is rotated cyclically to the right by P-bit positions. The basic hardware configuration that realizes this algorithm is shown in Fig. 1. Three numbers exist in the circuit; the present number X_n and the two previous random numbers X_{n-1} and X_{n-2}. The next random number X_n is generated after the present X_n is clocked into the flip-flops storing X_{n-1} and the present X_{n-1} is clocked into the flip-flops storing X_{n-2}. In Fig. 1 the outputs of the EXCLUSIVE-OR gates are rotated one position to the right. In a more general case they may be rotated P positions to the right, where, for good statistics, P must be mutually prime to L [1].

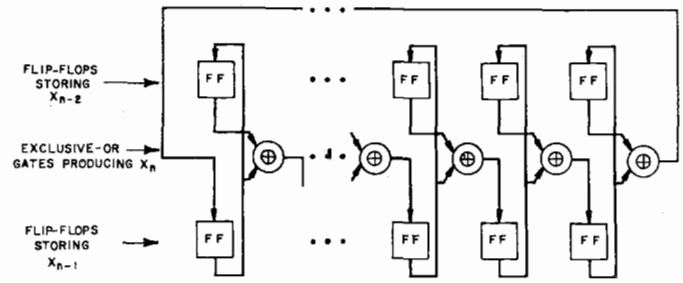

Fig. 1. Hardware configuration for uniform generator algorithm.

Such a configuration has obvious hardware advantages since the total time to compute a new number is the flip-flop settling time plus the delay through the EXCLUSIVE-OR gate. Furthermore, for L-bit random numbers, the only hardware required is $2L$ flip-flops for storage of X_{n-1} and X_{n-2}, L EXCLUSIVE-OR gates, and the necessary control logic.

The periods of sequences generated by this algorithm depend on L and P. P should be mutually prime to L and long periods are obtained only for certain values of L [1]. In the system that we have constructed, the word length is 19 bits and the outputs of the EXCLUSIVE-OR gates are rotated eight positions to the right. For these conditions, the period is 14 942 265. As shown in the next section, the sequence values obtained using the above algorithm are nearly uniformly distributed.

An approximation to a Gaussian distribution of amplitudes can be obtained by summing a finite number of uniformly distributed numbers. Using this principle, we have designed and built a flexible hardware random number generator that approximates pseudorandom number sequences with either uniform or Gaussian amplitude distributions. Fig. 2 shows a block diagram of the hardware, which operates as a peripheral device to a DDP-516 computer. The 19-bit uniform generator described previously is the basic subsystem. The uniform generator can be reset to its initial state[1] by an output instruction from the computer, thus enabling the user to obtain reproducible pseudorandom sequences. The computer commands the generator to pro-

Manuscript received May 4, 1972.
The authors are with Bell Telephone Laboratories, Inc., Murray Hill, N. J. 07974.

[1] The initial state of the system is (in binary form) X_{n-1} = 0000000000000000000 and X_{n-2} = 1000000000000000000.

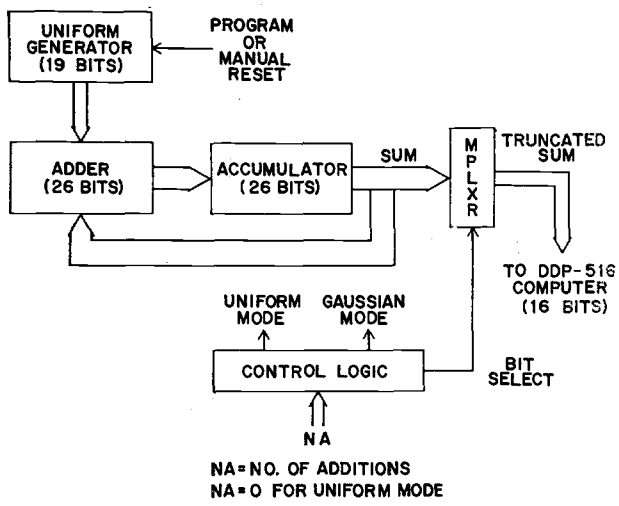

Fig. 2. Block diagram of hardware number generator.

duce either uniform or approximately Gaussian number sequences by outputting to the generator a constant NA, as shown at the bottom of Fig. 2. If NA is zero, a uniformly distributed sequence is produced, while if NA is in the range 1–213, an approximation to a Gaussian distribution is produced as a result of accumulating 2–214 samples of the basic 19-bit sequence.

This is accomplished in the hardware by connecting the output of the uniform generator and the output of an accumulator register to the inputs of a carry-save adder. The output of the adder is connected to the input of an accumulator register that accumulates from 2 to 214 uniformly distributed numbers. The size of the accumulator register is 26 bits, which is the maximum size of any result obtained by summing 214 consecutive 19-bit uniform numbers starting at the beginning of the uniform sequence. When fewer uniform numbers are summed, the size of the result held in the accumulator is equal to or less than 26 bits. For any size result the multiplexer selects the 16 most significant bits and transfers them from the accumulator to the computer upon computer request.

The transaction between the computer and the accumulator is handled through a computer data channel and is completed in 1.92 μs. However, the hardware, which is constructed out of high-speed TTL logic circuits and uses the carry-save adder technique, is capable of generating numbers at a much faster rate. The time required by the hardware to generate a number depends on the number of additions it has to perform. This time is approximated by

$$T = 0.075(5 + NA)\ \mu\text{s},$$

where NA is the number of additions. Thus, the generator can produce one uniform number every 0.375 μs and one Gaussian number, obtained with 15 additions, every 1.5 μs.

These times are significantly faster than software realizations of the same algorithm. Table I shows comparisons for the hardware and DDP-516 machine language programs that produce the same sequence.

TABLE I
Comparison of Run Time (μs)

	Hardware	Software	Ratio
Uniform	0.375	62.4	166
Gaussian ($NA = 15$)	1.5	920	613

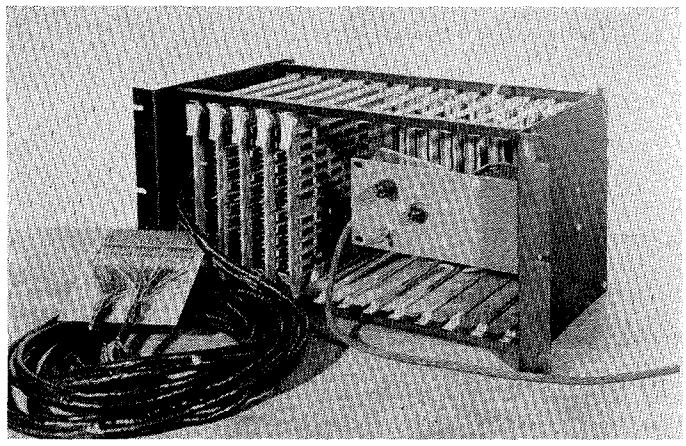

Fig. 3. Hardware random number generator.

With a slight modification the hardware could be made into a stand-alone device or could be used as a component of a larger system. As a stand-alone device, the initializing parameters could be entered manually and, with a D/A converter at the output, the generator could serve as an analog random noise generator.

The hardware random number generator is shown in Fig. 3. The device was built with the following basic components: four-wire wrap circuit boards that plug into a holding case; about 200 TTL dual-in-line integrated circuit packages; and a power supply. It should be noted, however, that this device was designed to operate at high speed and to have more flexibility than might normally be required. A generator designed for slower speeds and using a carry-ripple adder would require fewer integrated circuits. If the number of additions used to produce a Gaussian approximation is fixed, the multiplexer and some control logic could be eliminated. If a Gaussian approximation is not required, the basic hardware reduces to little more that what is shown in Fig. 1.

Statistical Properties

The period of the 19-bit uniform generator is satisfactory for speech and auditory simulations. However, a long period alone does not insure a useful digital random number generator. Given a sequence with a long period, the main concern is how well the statistical properties of the sequence match those of a desired random process. In particular, we are generally concerned with the autocorrelation function, the power spectrum, and the amplitude distribution of the numbers in the sequence. Fig. 4 shows an estimate of the autocorrelation function and the power spectrum for the uniform gen-

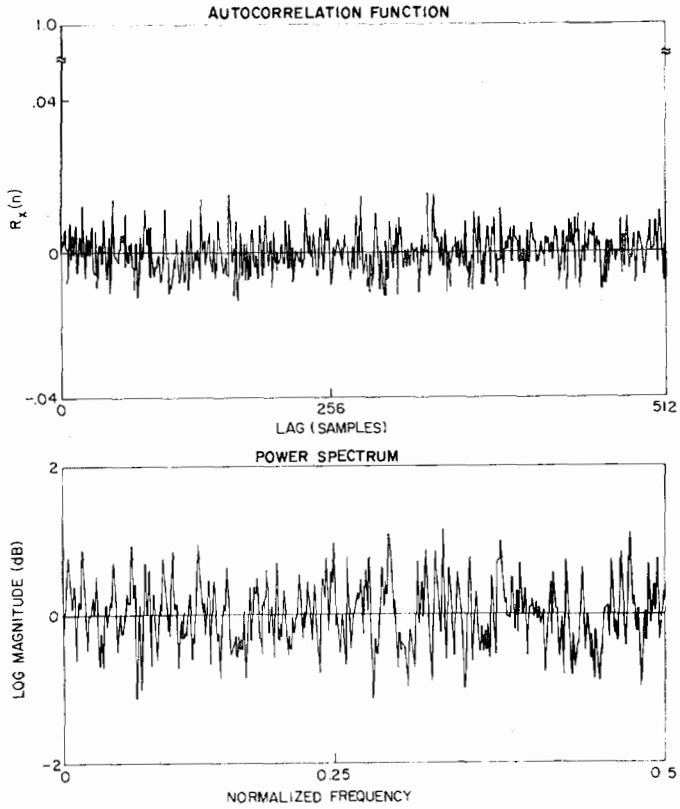

Fig. 4. Autocorrelation function and power spectrum for uniform generator.

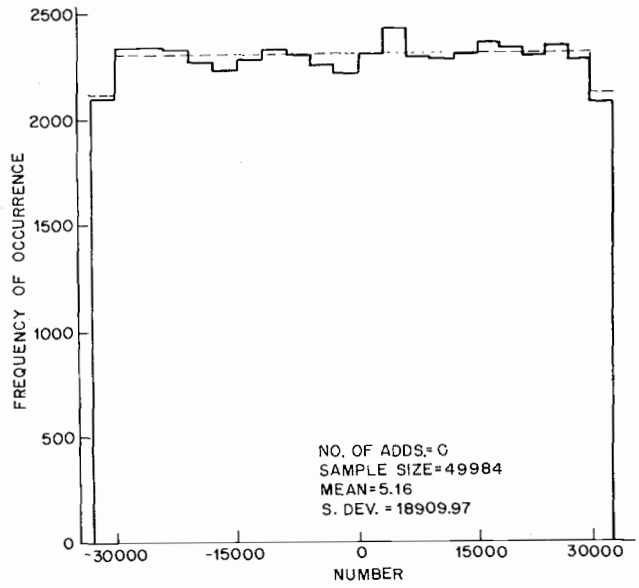

Fig. 5. Histogram of uniform generator.

erator [2]. The value of $R_x(0)$ is normalized to 1.0, while the maximum value obtained for any other lag is 0.016. It is clear from this figure that there are no irregularities present in the autocorrelation function. Statistical tests showed no reason to reject the hypothesis that the sequence approximates uncorrelated or white noise. The amplitude distribution of the uniform generator was measured and a typical result is shown in Fig. 5. The

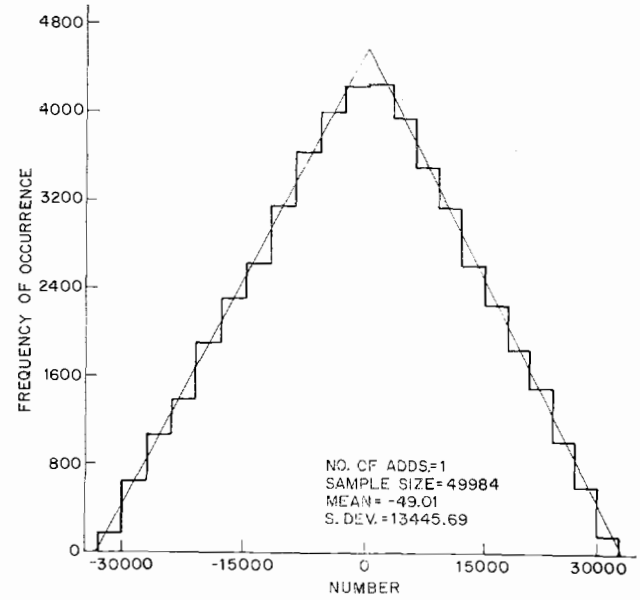

Fig. 6. Histogram of triangular generator ($NA = 1$).

most significant 16 bits of the 19-bit word were selected and the result interpreted as a two's complement number. Therefore, the numbers range from -32768 to $+32767$. The histogram in this figure was constructed from a sample of 49984 values of the sequence. The dashed line indicates the expected number of occurrences in each cell for the assumption that the numbers are uniformly distributed.[2] A χ^2 test was performed to test the goodness of fit of this uniform distribution, and at the 5 percent level of significance, there is no reason to reject the hypothesis that the distribution is uniform as shown. Thus, measurements of the autocorrelation function and the amplitude distribution indicate that good approximations to uniformly distributed white noise sequences are produced by the hardware operating in the uniform mode.

The central limit theorem provides a theoretical basis for obtaining a Gaussian distribution by summing a finite number of values of the uniformly distributed sequence. If N random variables are summed, the resulting amplitude distribution is the $(N-1)$-fold convolution of the original probability distribution with itself. As N increases, the distribution approaches Gaussian. For example, if two identically distributed uniform random variables are summed the resulting amplitude distribution is triangular. Fig. 6 shows the amplitude distribution measured for the sequence resulting from summing two uniformly distributed variables. The straight lines indicate the expected number of occurrences for the assumption that the numbers are triangularly distributed.

As the number of uniform variables that are summed increases, the distribution more nearly approximates a

[2] The cells on each end of the histogram are of width 2768 while the others have width 3000. Thus, the end cells have proportionally fewer occurrences.

Gaussian distribution. An example is shown in Fig. 7. In this example, 16 consecutive 19-bit numbers from the uniform generator were summed to produce each value of a new sequence. The distribution of sequence values shown in Fig. 7 was based on 49 984 samples of this sequence. The smooth curve is a Gaussian distribution with mean and variance equal to the sample mean and variance. Fig. 8 shows an estimate of the autocorrelation function and the power spectrum for this generator. The value of $R_x(0)$ is normalized to 1.0 while the maximum value obtained for any other lag is 0.016. A χ^2 goodness-of-fit test indicates that there is no reason to reject the hypothesis that the distribution is Gaussian at the 5 percent level of significance.

The amplitudes of the uniform generator are very close to being uniformly distributed with numbers between $-32\,768$ and $+32\,767$. Also, computed estimates of the mean and variance of the uniform number sequences are very close to the theoretically determined mean and variance of such a uniform distribution. When $NA+1$ numbers are added to obtain a Gaussian approximation, the theoretical mean and variance increase in proportion to $NA+1$. However, since only the most significant 16 bits are returned to the computer, each number is effectively divided by a power of 2, which depends on NA. The effect on the mean and variance is given by the equations

$$\mu_{NA} = \frac{(NA+1)\mu_0}{[D(NA)]^{\frac{1}{2}}} \tag{1}$$

and

$$\sigma_{NA}{}^2 = \frac{(NA+1)\sigma_0{}^2}{D(NA)}, \tag{2}$$

where μ_{NA} and $\sigma_{NA}{}^2$ are the mean and variance for NA additions, μ_0 and $\sigma_0{}^2$ the mean and variance of the uniform distribution, and $D(NA)$ is determined by hardware considerations. Table II shows $D(NA)$ as a function of NA.

The theoretical standard deviation for the uniform generator is $\sigma_0 = 18\,919$. A computed estimate of the standard deviation is 18 910 as seen in Fig. 5. Using Table II and (2), the theoretical standard deviations for the examples of Figs. 6 and 7 are $\sigma_1 = 13\,378$ and $\sigma_{15} = 4730$ as compared to the computed estimates of 13 446 and 4759, respectively. As seen from these examples, there is a close agreement between the theoretical and computed estimates of variance for the samples shown. In fact, we have found that (2) and Table II can be used to reliably predict the variance for any number of additions.[3]

The uniformly distributed sequence has a period $N_U = 14\,942\,265$. Since the Gaussian sequence is ob-

[3] Note that some control over the variance is offered by the choice of NA.

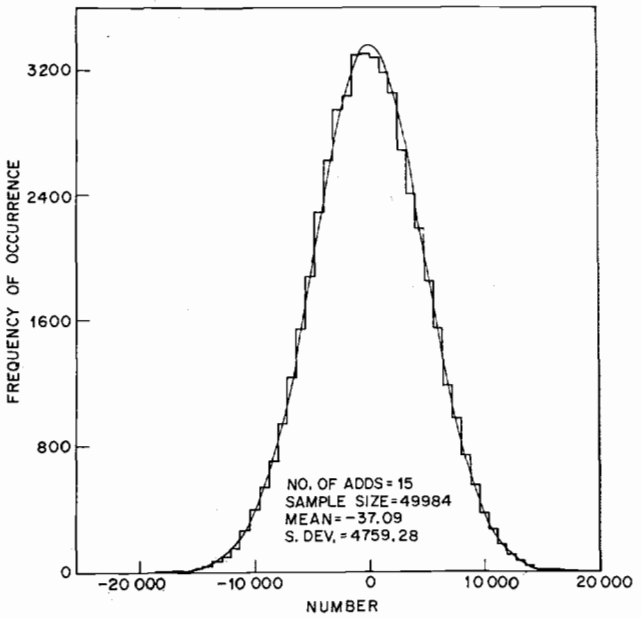

Fig. 7. Histogram of Gaussian generator ($NA = 12$).

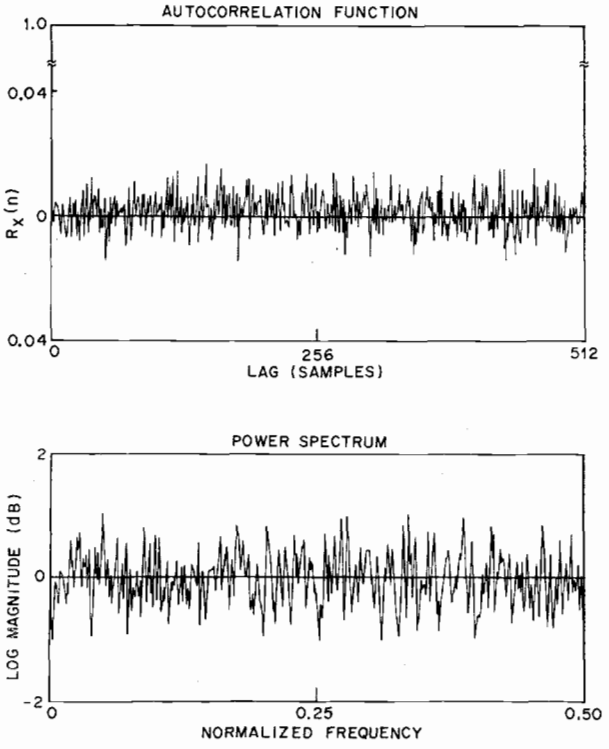

Fig. 8. Autocorrelation function and spectrum for Gaussian generator ($NA = 12$).

TABLE II

NA	$D(NA)$
0	1
1	4
2–3	16
4–7	64
8–19	256
20–44	1024
45–98	4096
99–213	16384

tained by summing $NA+1$ values of the uniform sequence, the period of the Gaussian sequence is

$$N_G = N_U/M$$

where M is the greatest common divisor of N_U and $NA+1$. The period of the uniform generator can be expressed as

$$N_U = (3)(5)(13)(19)(37)(109).$$

Thus, to insure that the greatest period is achieved, $NA+1$ must not have a factor of 3, 5, 13, 19, 37, or 109.[4] All other choices produce a period $N_G = 14\,942\,265$. Thus in our example, the Gaussian generator of $NA=15$ has the full period N_U.

Summary

A flexible pseudorandom number generator that provides reproducible uniform or Gaussian sequences has been constructed using TTL logic circuits. This device provides uniformly distributed numbers about 166 times faster than a machine language program of the identical generator and numbers with Gaussian distributions ($NA=15$) about 613 times faster than a machine language program. This increase in speed over software random number generators makes it possible to significantly reduce computation time in simulations that require many random numbers.

[4] Note that in all cases, the Gaussian samples will not be uncorrelated for lags greater than $N_U/(NA+1)$.

Acknowledgment

The authors wish to acknowledge the discussions with C. M. Rader that contributed to this work.

References

[1] C. M. Rader, L. R. Rabiner, and R. W. Schafer, "A fast method of generating digital random numbers," *Bell Syst. Tech. J.*, vol. 49, pp. 2303–2310, Nov. 1970.
[2] C. M. Rader, "An improved algorithm for high-speed autocorrelation with applications to spectral estimation," *IEEE Trans. Audio Electroacoust.*, vol. AU-18, pp. 439–441, Dec. 1970.

Digital Ladder Structures and Coefficient Sensitivity

RONALD E. CROCHIERE

Abstract—Recently, there has been a great deal of interest in the implementation of digital filter structures with low-coefficient word length. A conjecture has been made by Fettweis that if digital filter structures are modeled after analog ladder structures, which are known to have desirable coefficient sensitivity properties, then the digital ladder structures will also have these properties and could be implemented with low-coefficient word lengths.

To investigate this conjecture, a seventh-order Chebyshev low-pass filter was realized as a digital ladder structure and the coefficient sensitivity was analyzed experimentally under coefficient rounding in floating-point representation. To serve as a comparison similar examples of cascade structures of direct and coupled form sections were also analyzed in the same manner. The conclusions drawn are that, indeed, the digital ladder structures in many cases can be implemented with lower coefficient word lengths than the conventional structures.

Manuscript received April 6, 1972; revised May 2, 1972. This work was supported by the National Science Foundation Grant GK-31353. This paper was presented at the IEEE Workshop on Digital Filtering, Arden House, N. Y., January 1972.

The author is with the Department of Electrical Engineering and Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Mass. 02139.

I. Introduction

Recent studies (Fettweis [1, p. 79 ff], [2]) show that digital filter structures can be modeled after classical LC ladder structures. Since doubly terminated LC ladder structures are noted for the relative insensitivity of their frequency response to the element values, it has been conjectured that the digital structures should also have this desirable insensitivity.

In digital signal processing, recursive filters are often implemented as cascade or parallel structures of first- and second-order filters. One reason for this choice is to achieve relative coefficient insensitivity. In order to compare the coefficient-sensitivity properties of digital ladder structures with those of the conventional cascade structures, a seventh-order Chebyshev low-pass filter was realized as a ladder structure, a cascade structure of direct form filters, and a cascade structure of coupled form filters. The degeneration in the frequency responses of the three realizations are compared as the coefficients are quantized.

Essentially two alternatives are available for the design of the digital ladder structure. The first method utilizes an existing design of a lumped-element analog ladder structure and from this design a digital ladder structure is derived with the use of Richards' transformations, Kuroda's identities, and the "digitization" methods of Fettweis [1, p. 79 ff], [2]. The second alternative is to utilize the theory of unit-element filters directly in a synthesis procedure [3], [9] and synthesize a unit-element ladder structure, which can then be converted to a digital ladder structure, again by the "digitization"