COMPUTER TECHNOLOGY

A hardware random number generator for use with computer control of probabilistic contingencies[1]

J. R. MILLENSON *and* G. D. SULLIVAN *INSTITUTE OF EXPERIMENTAL PSYCHOLOGY, OXFORD UNIVERSITY, Oxford, England*

A solid-state random number generator is described, which uses the leakage across a reversed biased diode as the random source. Statistical tests carried out on sequences of numbers failed to show any biases from randomness at sampling rates up to 3 kHz.

Probabilistic contingencies are frequently required in psychological research. Perhaps the most obvious cases occur in probabilistic learning experiments and in the various random schedules of reinforcement for operant behavior. But more generally, production of haphazard stimulus sequences, randomization of time intervals, and stochastic selection of particular stimuli from a given stimulus population all rely upon probabilistic sampling.

A variety of methods have been used by experimenters to produce probabilistic events, including electronic roulette wheel principles (e.g., Schoenfeld, Cumming, Snapper, & Haas, 1960; Clark & Hull, 1965; Hendry, 1965), such naturally occurring random processes as radioactive particle emission (Millenson, 1965), and, of course, random number tables.

When experiments are computer-controlled it is more usual to generate randomization by the stored program itself. Program control of randomization permits a far greater degree of flexibility in probabilistic contingencies than any of the special-purpose devices available. Under computer control, for instance, probabilities may be automatically set, incremented, and decremented according to a present program, or by moment-to-moment properties of the S's responding. Moreover, with only minor additions to the program, independent probability values may be concurrently produced when more than one experiment is running simultaneously.

At present, such program control of probabilistic procedures is generally achieved by software (e.g., McLean, 1968; Grason-Stadler, 1968), that is, by the generation of a sequence of pseudorandom numbers via additive or multiplicative subroutines (cf. Green, 1963). Aside from the fact that such sequences deviate slightly but significantly from perfect randomness, they have two other defects that, under certain conditions, can prove troublesome in their practical use. First, when an experimental procedure using software-generated random numbers is repeated day after day, the *same* sequence of numbers is invariably generated. Since an experimental S might eventually come to discriminate the sequence as such, the programmer-experimenter is forced to intervene manually in order to insure that the sequence starts at a different point each time it is required. (Of course, the ability to generate the same sequence over and over is often considered an advantage since repetition is possible but can be easily avoided by starting with a different initial entry.)

The sheer time required to generate a random number by software may constitute a second problem in real time processing. The multiplicative generator used with the Digital Equipment Corporation PDP 5/8 family (Brady, 1965) takes 4 msec with the PDP 8/S. In the control language presently in use with that machine in our laboratory (Millenson, 1968), this time constitutes an appreciable proportion (over 20%) of the total time available for updating experiments.

The difficulties inherent in software generators are avoided in the hardware generator described below, in which the random source is the "shot noise" across a reversed biased diode (Van der Ziel, 1959; Bassett, 1968).

Circuit Description

The noise generator and amplifier circuit is shown in Fig. 1. Diode D1 is a low-quality 8-V Zener diode, which was found by trial and error to have high noise properties. Individual diodes vary considerably, so the biasing resistor (R1) should be selected for maximum output. We found values in the range 10-50 k ohm satisfactory. The voltage fluctuations from the diode are amplified by two BC109 stages before driving a BCY33 emitter-follower output stage. A 100 k ohm Minipot adjustable resistor permits fine adjustment of the mean output voltage. The entire circuit is compact enough to be constructed on a single 5 x 2½ in. plug-in module board.

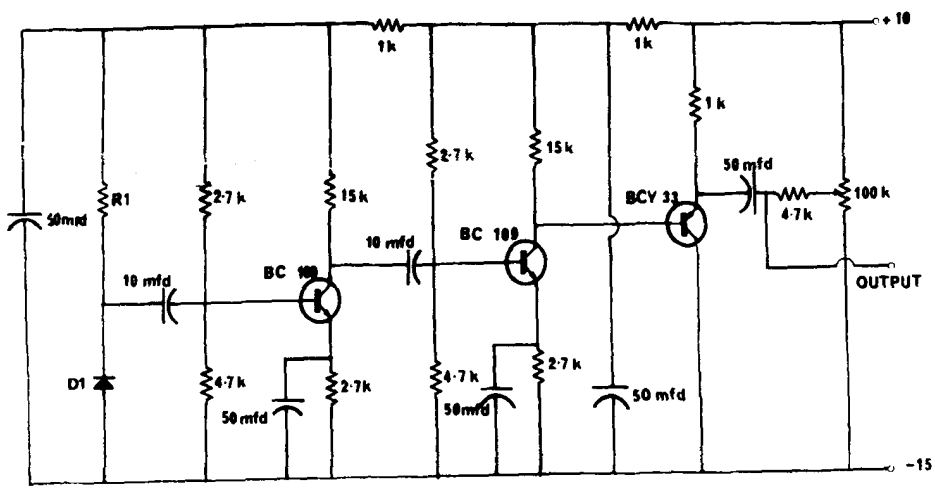

Fig. 1. Schematic of the noise generator and amplifier. D1 and R1 are explained in the text.

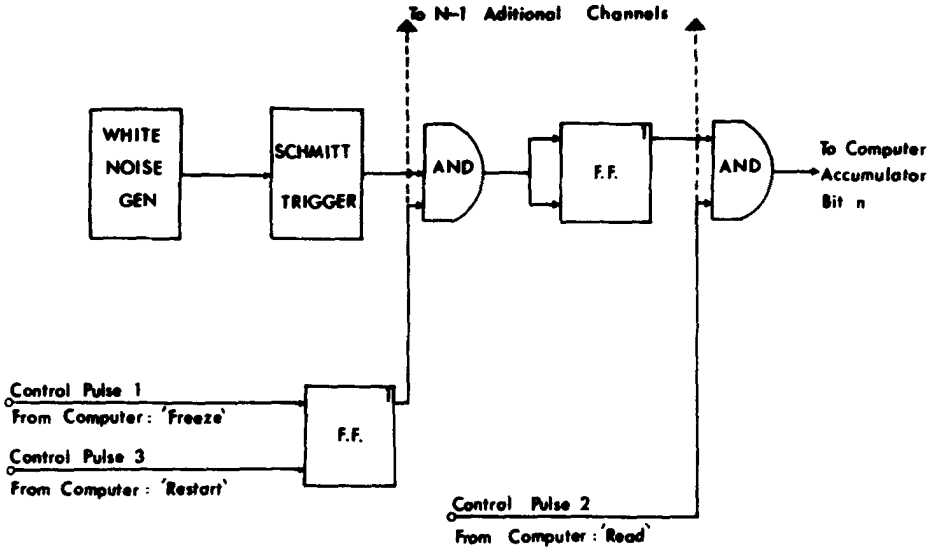

Fig. 2. Block diagram of one channel of binary random number generator and associated logic for reading the state of the output FF to the PDP /S accumulator.

The output from the circuit of Fig. 1 consists of randomly varying voltage fluctuations. These fluctuations are transformed to a time-dependent binary (0 and 1) random variable accessible to the accumulator of a computer by the circuit of Fig. 2. First the signal is applied to a Schmitt trigger, which passes only negative going noise peaks below a given threshold and standardizes them as sharp pulses varying randomly in time. (A typical sample of this transformation is shown in Fig. 3.) At times determined by the state of the interposed AND gate this pulse stream from the Schmitt is used to drive a flip-flop (FF) operating in J-K (toggle) mode, so that at any given moment the FF will be set to either 1 or 0 randomly.

Expanding the circuit of Fig. 2 to N such channels produces an N bit random number, whose bits are independent. In the PDP /S, a 12-bit buffer made up of 12 channels may be strobed directly to the accumulator in 38 μsec. Thus the time improvement on the software routine is better than a factor of 100. The method of reading used in our laboratory employs standard Digital Equipment Corporation flipchips and proceeds in three steps (see Fig. 2). First a computer-generated command pulse (1) is sent down to disenable the AND gate. This prevents random pulses from reaching the FF buffer, thus effectively "freezing" the current random number. One microsecond later the FF buffer is strobed into a single bit of the accumulator of the PDP 8/S by a second command pulse (2). Finally, yet another microsecond later, the gate to the random FF is reenabled by computer pulse (3). It is necessary momentarily to "freeze" the random number, otherwise the number might change while being read, and since changes from 0 to 1, or from 1 to 0, would both set the corresponding accumulator bit to 1, a systematic deviation from randomness would be introduced.

It should be emphasized that an entire N-bit system does not go on one card but only the noise generator for a single bit. One of everything in Fig. 2 is required for each bit of random noise.

Statistical Tests

An attempt was made to determine at what rate of sampling a channel would reliably give 1's and 0's randomly. The variable resistor of Fig. 1 was adjusted to produce an average Schmitt pulse output of about 6 kHz. Figure 4 shows the observed probability that two successive readings of the output FF produce the same number as a function of time between readings. It is apparent that above 300-μsec intersampling intervals, successive samples are independent. In further tests with average pulse rates raised to 16 kHz and above, successive samples were found independent down to intersampling intervals as short as 150 μsec.

The presence of any systematic bias in a 6-bit prototype generator towards particular numbers was tested. The generator was sampled 262,144 times at an intersampling time of approximately 350 μsec, and occurrences of each of the possible numbers (0 to 63) were counted. Figure 5 (top) shows the observed probability of occurrence, $P_{(n)}$, against the value (n). The dotted lines delimit the band in which 98% of the points are expected to fall, as predicted from chi square tables. A chi square test across all data points was consistent with the hypothesis that these points come from the distribution of equally likely values ($\chi^2 = 74.2$; df = 63).

First-order sequential dependencies in successively drawn 6-bit numbers were tested by determining the distribution of numbers occurring immediately after an arbitrarily chosen number (010101) was read. Figure 5 (bottom) shows the probability ($P_{(n/25)}$) that a number of value n will follow an occurrence of the arbitrary number. A chi square test over all the data points showed that the results were well within the expectation of randomness ($\chi^2 = 70.0$; df = 63).

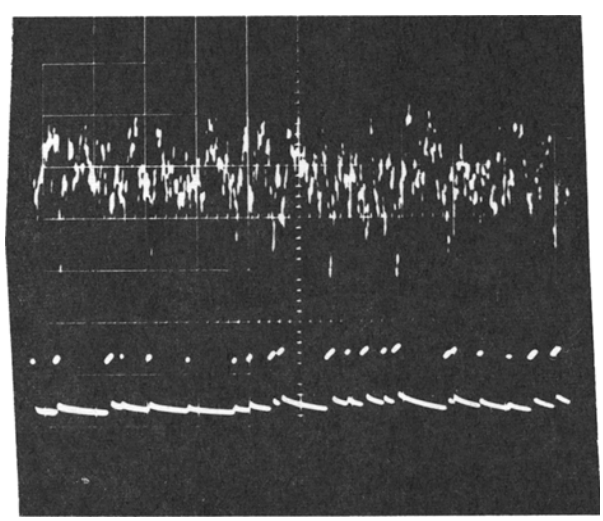

Fig. 3. (top) Output fluctuations from random noise circuit of Fig. 1. (bottom) Simultaneous Schmitt trigger output of Fig. 2. Time scale = 100 μsec per division.

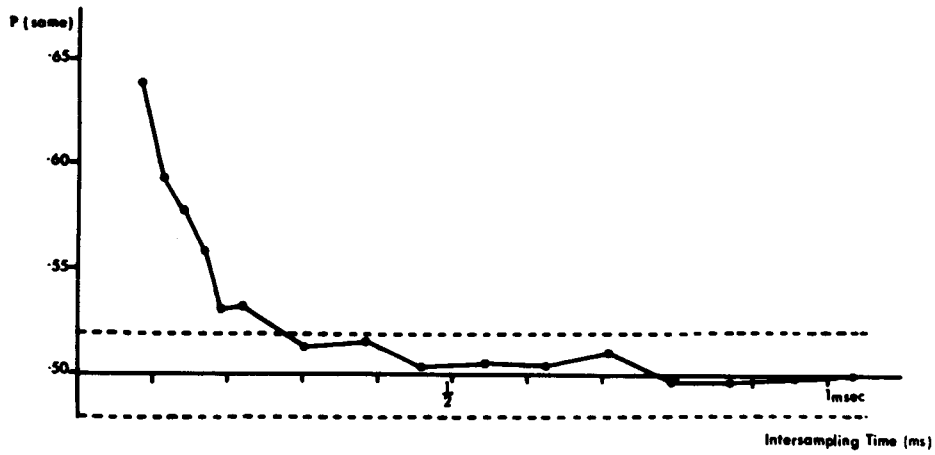

Fig. 4. Probability that successively sampled binary random digits are the same, as a function of intersampling time. The dashed lines indicate the 98% chi square confidence limits. Each plotted point is the mean of three trials of 4096 samples.

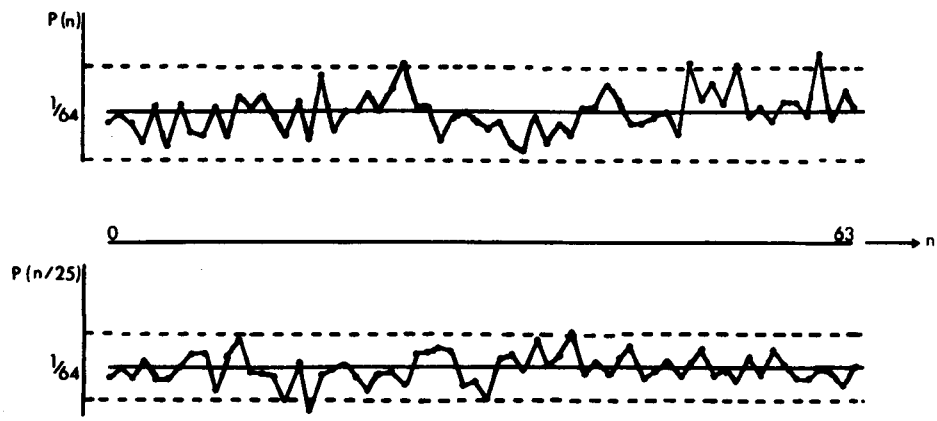

Fig. 5. (top) Observed probabilities of given numbers from 0 to 63 produced by 262,144 successive samples of a 6-bit version of the generator. (bottom) Observed probabilities of given numbers from 0 to 63 produced by 131,072 samples following the arbitrarily chosen number, 010101. Dashed lines in both portions are the 98% confidence intervals.

A weak test of channel independence was performed by counting the number of times that k bits were set in a 6-bit number over 4096 samples (where k = 0, 1, . . . 6). The means of 24 trials were plotted and were found to be visually indistinguishable from the predicted binomial distribution.

The 6-bit prototype described here has successfully been used in the PDP /S system in this laboratory to control probabilistic reinforcement schedules, with considerable time saving over software generators. In principal, the generator is adaptable to any computing system and can generate numbers of any desired length.

REFERENCES

BASSETT, A. J. White noise generator. Practical Electronics, January 1968, 44-45.

BRADY, P. T. A pseudo random number generator for the PDP 5 computer. DECUS Program Library, Digital Equipment Corporation, Maynard, Mass., July 27, 1965.

CLARK, F. C., & HULL, L. D. The generation of random interval schedules. Journal of the Experimental Analysis of Behavior, 1965, 8, 131-133.

GRASON-STADLER. *The SCAT primer* (preliminary). West Concord, Mass.: Grason-Stadler Co., 1968.

GREEN, B. F., JR. *Digital computers in research.* New York: McGraw-Hill, 1963.

HENDRY, D. P. A central probability-generator station. Journal of the Experimental Analysis of Behavior, 1965, 8, 447-449.

McLEAN, R. S. PSYCHOL: A formal language for the control of psychological experiments. Unpublished PhD thesis, Carnegie-Mellon University, 1968.

MILLENSON, J. R. An inexpensive geiger gate for controlling probabilities of events. Journal of the Experimental Analysis of Behavior, 1965, 8, 345-346.

MILLENSON, J. R. A general language for on-line control of psychological experimentation. DECUS Proceedings, Spring 1968, 137-144.

SCHOENFELD, W. N., CUMMING, W. W., SNAPPER, A. G., & HAAS, P. Some electronic control units for operant behavior studies: II. A random ratio generator. Journal of the Experimental Analysis of Behavior, 1960, 3, 107-108.

VAN DER ZIEL, A. *Fluctuation phenomena in semiconductors.* London: Butterworth Scientific Publications, 1959.

NOTE

1. Development of this apparatus was supported by Grant B/SR/5528 from the Science Research Council (UK).