Technical Articles

Stochastic computer thrives on noise
page 72

Many of today's computer control applications cannot be handled by conventional analog or digital machines, or even hybrid machines. A new approach, in which probability as an analog quantity switches a digital circuit, offers interesting promise. In this kind of machine, digital integrated circuits are randomly switched to simulate analog computer elements. Although slower than an analog machine and not as accurate as a digital one, the stochastic computer has a speed-size-economy combination that cannot be matched by either.

Computer-aided design: part 10, Making a video amplifier to measure
page 85

Even the performance of circuits that an engineer is very familiar with can be improved by using a computer to help in the design. For one thing, the computer can tailor the design to meet a wide range of operational requirements. Then too, a circuit can be refined for optimal operation plus easy and inexpensive construction. Here computer-aided design is applied to a well-known circuit, the video amplifier.

Special report Medical electronics: part I
page 95

The start of a comprehensive examination of medical electronics.

1. Prescription for medical instrumentation
A doctor finds the electronics industry so intrigued by gadgets that it isn't developing the kind of equipment doctors want. He urges engineers to help doctors find practical solutions to diagnostic and therapeutic problems.

2. Collecting body signals
What doctors most need are measurements of key body parameters. Present technology can produce equipment extracting far more information from such traditional diagnostic aids as electrocardiogram traces, electroencephalograms and electromyograms if engineers aim their designs at specific body signals. For the cover, Vincent Pollizzotto took his camera inside a big metropolitan hospital to show electronics in an advanced operating room.

Coming July 24

- **Replacing inductors with digital filters**
- **Glass devices for computer memories**
- **Special report: Medical electronics, part II, Computers in medicine**

Stochastic computer thrives on noise

Standard integrated circuits in a computer that applies probability as an analog quantity will economically increase speeds of computing for complex control problems

By Brian R. Gaines
Standard Telecommunications Laboratories Ltd., Harlow, Essex, England

A large segment of today's computer control applications—on-line control of chemical plants, aerospace navigation controls, and other large, complex systems—cannot be handled by conventional digital, analog, or hybrid computing systems. Even an order of magnitude increase in the capability of these systems would leave many computing needs unfilled. Therefore, instead of chipping away at the limitations of present systems, a new approach was taken to determine how today's computing elements—in particular, large-scale integrated digital circuits—could be used to fill these needs. The stochastic computer is the result of this new approach.

The stochastic computer differs from other computers in that it uses probability as an analog quantity—the probability of switching a digital circuit. It thus uses digital IC's that are randomly switched to simulate analog computer elements—multipliers, summers, inverters, integrators, and analog memories. Still in an early development stage, it is not quite as accurate as a digital computer, and it is not quite as fast as an analog computer, but because it uses low-cost elements in an analog configuration, it gives a speed-size-economy combination that cannot be matched by either conventional computer type.

Although the stochastic computer uses digital IC's, it is programed as an analog computer. Initial conditions are set with digital counters, which also serve as stochastic integrators. The use of digital IC's means that the analog-type circuits are made smaller and cost less and, with large-scale integration techniques, many functions will be combined in a single package.

The bandwidth of a digital computer is limited because it computes sequentially. Thus, on problems with many equations and variables, the total time per computation quickly becomes excessive. Analog computers are not suitable for large problems because many operational amplifiers are required and because accurate computing elements are high priced.

Attempts to capitalize on the best features of digital and analog computers have only resulted in new systems which are just as complex and as economically unfeasible as the individual computers.

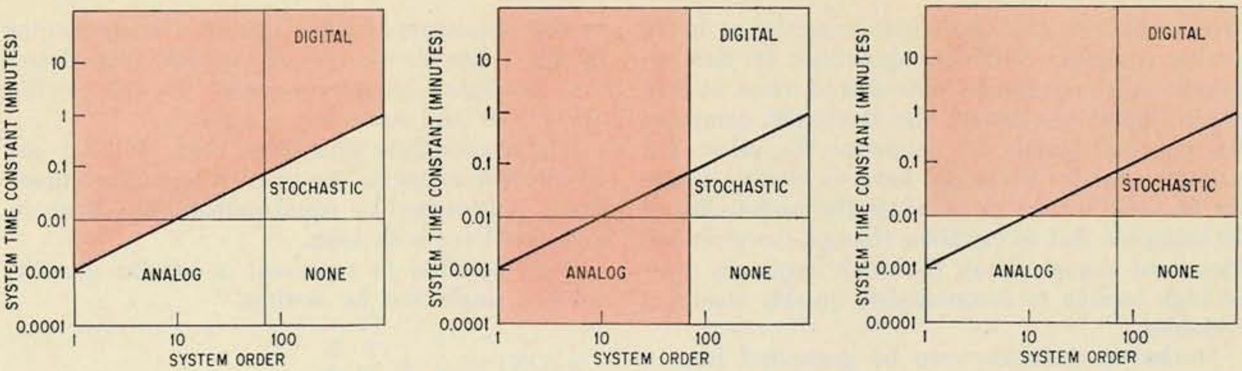
Stochastic computers are most useful where the availability of low-cost computing elements is more important than the speed and accuracy of computation. In many computer-controlled processes, a 10-hz bandwidth and an over-all accuracy of 1% is adequate, and where feedback is applied, even 10% accuracy may be acceptable.

Stochastic counting

The concept of representing a quantity by a probability, a stochastic representation, is basically simple. Consider a man counting the number of bales of straw coming along a conveyor belt by incrementing a counter one step as each bale passes him. What does he do if only half a bale passes by? If he neglects it and several other half-bales pass by, then he will underestimate the total amount of straw; if instead he counts it as one bale and increments the counter, then the more half-bales that come along, the more he overestimates the quantity of straw.

One solution would be to note that there was a half-bale and, when another comes along, increment the counter by unity. But this defeats the object of having a counter in the first place since the man now has to remember the data.

Another alternative is to count in units of half a bale (if it is a binary counter, add another flip-



Approximate regions of application for digital (left), analog (center), and stochastic (right) computers. Analog computers are limited mainly by system order (number of differential equations), while digital computers are limited in both size and speed. Stochastic computers are limited by speed but not by size. Advances in digital computers are steadily shifting the diagonal line toward the right, but a large area remains uncovered.

flop). This method is fine, (although it involves a slightly larger counter) until the bales begin arriving in less convenient fractions: a quarter-bale—two extra bits; an eighth-bale—three extra bits; and so it goes. Any form of rounding off is liable to lead to cumulative errors which may become appreciable in the long run.

Suppose now the man adopts an alternative technique. When half a bale comes along he tosses a coin and if it comes down heads, he increments the counter; if tails, then he does not. There is then a 50% probability that when half a bale comes along he will increment the counter.

This trick has some of the advantages gained by making the counter count in half units—in the long run there will be no bias towards underestimation or overestimation; the expected count will be exactly the number of bales that have passed. This trick can be extended to other fractions of a bale—if three-eighths of a bale passed, the man could toss three coins and increment the counter if two, and only two, of them were face up.

But note that although probability is a continuous variable not subject to round-off and cumulative errors, it is subject to another form of error—random variance.

To have correctly recorded the number of bales that passed, the man's counter should display a number equal to the mean value of all the counts that could have arisen. But he tossed the coin only a finite number of times, so the counter will display a count that varies from the mean. It will be one of the many probable counts clustering around the mean. Thus, with the coin-tossing scheme, the man has eliminated round-off error but has incurred a random variance error instead. But he has eliminated the need for memory and this basic advantage in hardware carries over to the stochastic computer.

Stochastic computing

A quantity within the stochastic computer is represented by the probability that a logic level in a clocked sequence will be ON. If many clock pulse positions are taken, then the quantity can be considered to be the fraction of ON logic levels in a clocked sequence of binary ON and OFF logic levels, with no pattern in the sequences of logic levels.

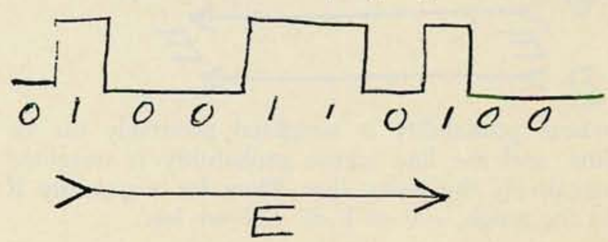
The probability that a logic level will be ON is an analog variable whose value ranges continuously from 0 to 1. In some computations, this is a convenient range to use, but generally the physical variables in a problem (length, etc.) have to be scaled into the allowable range of computer variables. This is similar to the scaling operation in the analog computer, where the physical variables are related to voltages. The actual mapping will determine the hardware required to perform a particular computation, and although many mapping schemes are possible, three examples will be considered: unipolar, two-line bipolar, and single-line bipolar.

In the unipolar representation, if the quantities are always positive (or always negative), simple scaling is all that is required to bring them within range. Given a quantity E in the range $0 \leq E \leq V$, it can be represented by the probability:

$$p\,(ON) = \frac{E}{V}$$

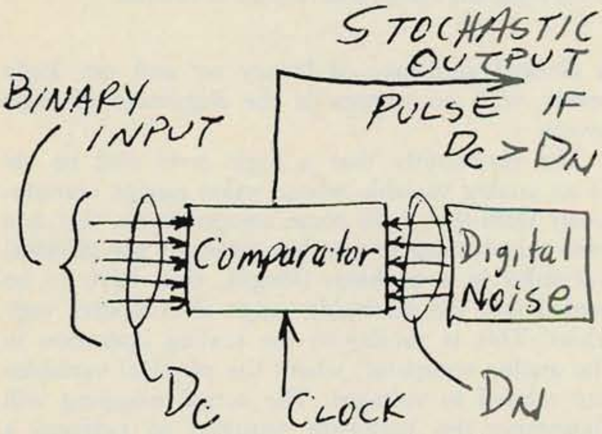
so that the maximum value of the range, $E = V$, is represented by a logic level always ON, $p(ON) = 1$, zero value by its being always OFF, $p\,(ON), = 0$, and an intermediate value by its fluctuating randomly, but with a certain probability that it will be ON at any particular instant.

For example, though the pulse sequence below is not infinite, we can grossly say that it has $p(ON) = 4/10$. If it represented a physical variable whose maximum value was, say, 17, then the value



represented here would be 17 x 4/10 or 6.8. Note that the wire that carries these pulses is always associated with this particular variable, as in an analog computer. Different quantities do not appear serially on common, time-shared wires, as they do in digital computers; the stochastic computer has no word length. To determine the value of a variable exactly, we would have to monitor a line for an infinite time to calculate the probability of its being ON. But as variables change, the probabilities must change. Thus, the clock frequency must be high enough to accommodate quickly changing variables.

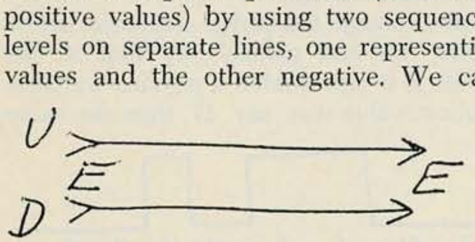
Stochastic quantities can be generated from a binary quantity with a comparator and a digital noise source. The comparator will provide an output pulse if, at a particular clock-pulse time, the



binary input is a larger number than the number represented by the digital noise pulses. For example, if all binary digits are 0, then the digital noise source will always be greater and no pulses will appear on the output line, so that p(ON) = 0. Similarly, if all binary digits are 1, then the comparator will deliver a continuous stream of output pulses. Thus, for a high value of the binary number, the probability is that it will often be greater than the randomly generated digital noise number and many ON output pulses will be generated. The probability of an output pulse thus is directly proportional to the magnitude of the binary number.

Besides serving as a means of inserting data in the computer, this scheme also is used to produce a stochastic output from a stochastic integrator.

The unipolar representation can be simply extended to bipolar quantities (both negative and positive values) by using two sequences of logic levels on separate lines, one representing positive values and the other negative. We call the line



whose probability is weighted positively the UP line, and the line whose probability is weighted negatively the DOWN line. Then for a quantity E in the range, $-V \leq E \leq +V$, we let:

$$p\,(UP = ON) - p\,(DOWN = ON) = \frac{E}{V}$$

so that maximum positive quantity is represented by the UP line always ON and the DOWN line always OFF, maximum negative quantity by the UP line always OFF and DOWN line always ON.

For intermediate quantities there will be stochastic sequences on one or both lines. Zero quantity is represented by equal probabilities of an ON logic level for both lines.

It is possible to represent a bipolar quantity using a single line by setting:

$$p\,(ON) = \frac{1}{2} + \frac{1}{2}\frac{E}{V}$$

so that maximum positive quantity is represented by a logic level always ON, maximum negative quantity by a logic level always OFF, and zero by a logic level fluctuating randomly with equal probability of being ON and OFF.

This is the stochastic representation most studied to date since it allows the simplest logical elements to carry out all the normal analog computing operations with both positive and negative quantities.

Comparison of various computers

Although the stochastic representation of quantity leads to economy in computer hardware, the price paid for this economy is a low computing efficiency. Since probability cannot be measured exactly, it must be estimated with an error that decreases with the number of samples taken. Hence, high accuracy means low bandwidth.

The effect of this variance on the efficiency of representation may be seen by comparing the number of levels of voltages or states required to carry analog data with a precision of one part in N:

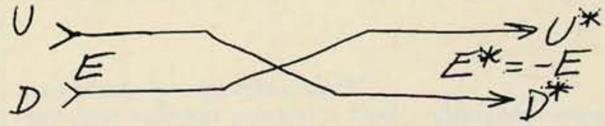
- The analog computer requires one continuous level.
- The digital computer requires $\log_2 mN$ ordered binary levels.
- The digital differential analyzer requires mN unordered binary levels.
- The stochastic computer requires mN^2 unordered binary levels.

The constant m is taken large enough to minimize the effects of round-off error or variance, say m = 10. The N^2 term is a result of the error being inversely proportional to the square root of the length of pulse sequence.

This comparison is a little unfair to the stochastic computer where operations such as integration are concerned, since the digital computer requires complex routines and additional precision to maintain this accuracy, whereas the redundancy of the stochastic representation enables the use of simple counting techniques for integration. However, this by no means compensates for the loss of efficiency in stochastic computing, which may be regained only in computations requiring parallel data-processing unsuited to the digital computer.

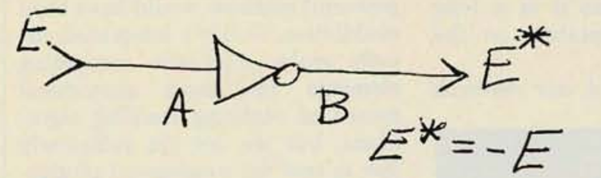
Stochastic computing elements

Inverters: to multiply a two-line bipolar quantity by −1 requires only the interchange of UP and



DOWN lines. The unipolar representation, of course, has no inversion operations.

In the single-line bipolar case, a conventional logical inverter, with an output that is the complement of its input, performs the same function when used as a stochastic element. Consider the



relationship between the probability that its output (E^*) will be ON, p(B), and the probability that its input (E) will be ON, p(A); since the two cases are mutually exclusive, the sum of the probabilities is 1, or:

$$p(B) = 1 - p(A)$$

Thus the probabilities and the quantities they represent are:

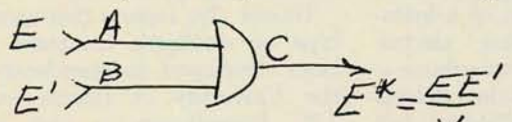
$$p(A) = \frac{1}{2}\frac{E}{V} + \frac{1}{2}$$

$$p(B) = \frac{1}{2}\frac{E^*}{V} + \frac{1}{2}$$

hence

$$E^* = -E$$

Multipliers: a simple two-input AND gate, whose output is ON if, and only if, both its inputs are ON, acts as a multiplier for quantities in the unipolar



representation. The relationship between the probability that its output will be ON, p(C), and the probability that its two inputs will be ON, p(A) and p(B), is:

$$p(C) = p(A)\,p(B)$$

These probabilities and the quantities they represent are:

$$p(A) = \frac{E}{V}$$

$$p(B) = \frac{E'}{V}$$

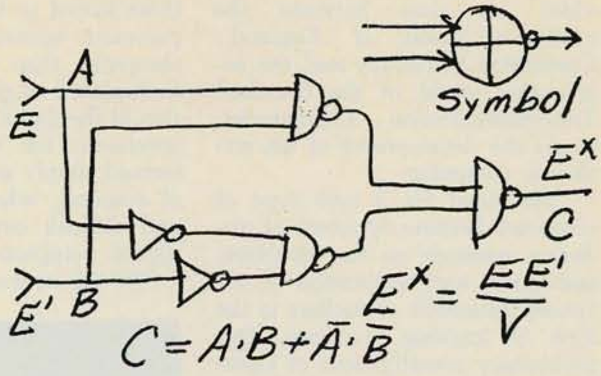
$$p(C) = \frac{E^x}{V}$$

Hence:

$$E^x = \frac{EE'}{V}$$

The product thus is normalized to lie within the range $0 \leq E^x \leq V$.

Multiplication in single-line bipolar representation is performed by an inverted exclusive-OR gate.



Its output is ON when its two inputs are the same, so that two positive quantities at the inputs, or two negative quantities at the inputs, represent a positive quantity at the output.

That multiplication does occur may be confirmed by examining the relationship between input and output probabilities for the gates shown.

$$p(C) = p(A)\,p(B) + [1 - p(A)]\,[1 - p(B)]$$

and

$$p(A) = \frac{1}{2} + \frac{1}{2}\frac{E}{V}$$

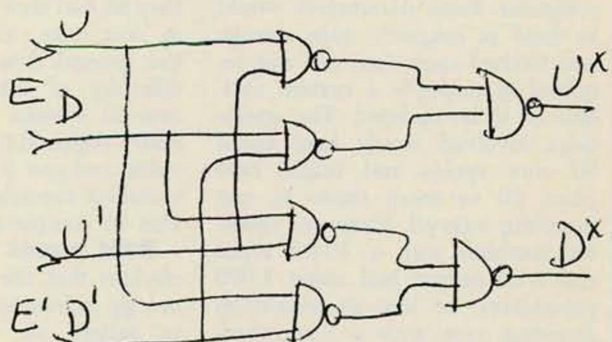
$$p(B) = \frac{1}{2} + \frac{1}{2}\frac{E'}{V}$$

so that:

$$p(C) = \frac{1}{2} + \frac{1}{2}\frac{EE'}{V^2}$$

which is normalized multiplication of E by E'.

Two similar gates are required for multiplication



in two-line bipolar representation; these stochastic multipliers can be realized with NAND logic.

Squarers: an important phenomenon is illustrated by the use of a stochastic multiplier as a squarer. Unlike conventional analog multipliers, it is not sufficient to short-circuit the inputs of the gate in a stochastic multiplier, for its output will then be the same as its input. This difficulty arises because the input sequences must be statistically independent. Fortunately, because the sequences are

Learning machines sparked the project

Author Brian R. Gaines, who divides his time between the academic world of England's Cambridge University and the engineering world of the Standard Telecommunication Laboratories, traces the development of his stochastic computer:

"The need for a new type of computer became apparent at STL during research on the structure, realization, and application of advanced automatic controllers in the form of learning machines. Our preliminary investigations of algorithms for machine learning were hampered by the time and expense of simulating even a small learning system on a conventional digital computer. Although we were successful in demonstrating that certain general algorithms could form the basis for machine learning in a wide variety of control and problem-solving environments, it was obvious that no conventional computing system was capable of realizing these algorithms in real time at an economic price.

"An adaptive controller is, by definition, a variable parameter system. This implies a minimum hardware complement of the following: stores to hold the parameters; multipliers to enable the paramaters to weigh other variables in the system; and parameter-adjustment logic to allow learning to take place.

Digital too slow. "In the digital computer these parameters would be held in magnetic core storage and fetched each time one was required to multiply a system variable or to be updated. The operations involved would take about 50 core cycles and might take place 20 or more times in one sampling interval. Even our smallest machine with a 10-bit input and 2-bit output had about 1,000 parameters, so that its maximum sampling rate with a 1-microsecond cycle time would be about 1 per second—10 times slower than a human operator.

"These figures can only be rough guides to the problems involved, but they indicate the difficulties in realizing a machine with even a fraction of the speed and adaptability of man—and eventually we must aim to surpass him.

"The computations involved in machine-learning were obviously more suited to the parallel, multiprocessor operation of the analog computer, than to the sequential, multiplexed single-processor operation of the digital computer. Matrix inversion, for example, is performed simply and rapidly by nets of resistors, whereas it is a long and difficult computation on the digital computer.

"In the course of our research

Brian R. Gaines

we investigated many forms of analog storage which might be used to hold the parameters of a learning machine. Capacitor, electroplating, electrolytic, transfluxor—they all had their individual defects in cost, size, and reliability, but the general deficiency was in the difficulty of integration into the over-all system. The external circuitry required to adjust the stored value and use it to multiply other variables exceeded the original device in complexity, size and cost.

Build around IC's. "It became obvious that the only device technology advancing rapidly enough to satisfy our requirements for large numbers of computing elements at low cost was that of integrated circuits. And it was decided to concentrate on new methods of computing with standard digital devices—gates and flip-flops —rather than develop new forms of devices. In particular we wish to be in the position to take full advantage of large-scale integration.

"Eight years ago we would have built a learning machine with vacuum tubes and relays, and it would have been massive, unreliable and useless. Four years ago we would have built it with discrete semiconductor devices and it would have served as a demonstration of principle—but the cost and size of a powerful machine would have been prohibitive. Today's integrated circuits make stochastic computing elements the most economical means of realizing learning algorithms, but are not yet sufficiently low in cost for commercial production of a learning system. In a few years time we shall have complex arrays containing hundreds of gates and flip-flops interconnected as a complete system."

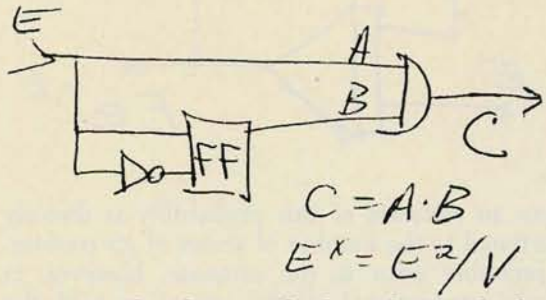
At STL, Gaines and others developed a family of stochastic computing elements and studied applications to problems of data-processing, automatic control, radar and pattern recognition. He points out that the only commercial data-processing device that now uses a stochastic representation of analog data is the Enhancetron made by Nuclear Data Inc. of Palatine, Ill. This unit is used in the study of biological responses to stimuli and replaces an analog-to-digital converter and a digital adder with a stochastic comparator and digital counter.

Gaines also reports that another type of stochastic computer has been developed independently at the University of Illinois under J.W. Poppelbaum as part of a program on optical data processing. The report of that work in Electronics, Dec. 12, 1966, spurred Gaines to write his article.

As a consultant to STL's New Systems division, Gaines, who recently completed his doctoral research on the human adaptive controller at the Cambridge psychological laboratory, developed the stochastic computer described in this article in close collaboration with J.H. Andreae, who is now at the University of Canterbury, Christchurch, New Zealand. It was constructed by P.L. Joyce of STL. It is the first computer to be developed as a result of this research, and is particularly fascinating because it uses what is normally regarded as a nuisance or waste product—random noise.

not patterned, a statistically independent sequence is obtained by delaying the input for one clock pulse.

A unipolar multiplier may be used as a squarer by interposing a delay flip-flop in one input. The



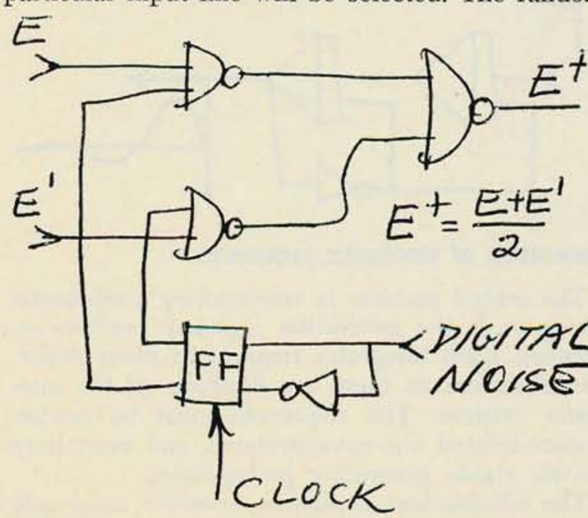
same may be done with the other multipliers, a delay in both UP and DOWN lines being required for the two-line representation.

Flip-flops used in this way perform as stochastic isolators, acting to remove the correlation between two sequences with a similar pattern.

Summers: having seen how readily inversion and multiplication is performed by simple gates, one is tempted to assume that similar gates are used to perform addition. However this is not so—stochastic logic elements must be introduced to sum the quantities represented by two stochastic sequences.

For example, consider two sequences in single-line bipolar representation, one representing maximum positive quantity and hence always ON, the other representing maximum negative quantity and hence always OFF. The sum of these quantities is zero and should be represented by the stochastic sequence with equal probabilities of being ON or OFF. But a probabilistic output cannot be obtained from a deterministic gate with constant inputs; stochastic behavior therefore must be built into the summing gates of the computer.

Stochastic summers may be regarded as switches which, at a clock pulse, randomly select one of the input lines and connect it to the output. The output line denotes the sum of the quantities represented by the input lines. The sum is weighted according to the probability (½ in the schematic below) that a particular input line will be selected. The random



selection is performed by internally generated stochastic sequences, obtained either by sampling flip-flops triggered by a high bandwidth noise source, or from a delayed sequence of a central pseudo-random shift register; these sequences are called digital noise.

Two-input stochastic summers with equal weighting are shown. Identical circuits are used for unipolar and single-line bipolar quantities while for two-line bipolar representations, an additional inhibitory gate reduces the variance of the output.

That addition does occur may be confirmed by examining the relationship between input and output probabilities for the gates below. Assuming symmetrically distributed digital noise, we have:

$$p\,(C) = \frac{1}{2}\,p\,(A) + \frac{1}{2}\,p\,(B)$$

and hence

$$\frac{E^+}{V} = \frac{1}{2}\frac{E}{V} + \frac{1}{2}\frac{E'}{V}$$

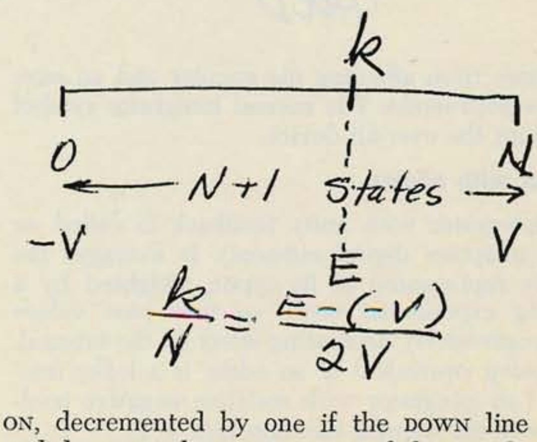
so that

$$E^+ = \frac{1}{2}\,(E + E')$$

which is normalized addition in the unipolar case and for the single-line bipolar case, too.

Integrators: The basic integrator in a stochastic computer is a digital counter. In the unipolar representation the counter is incremented by one if the input line representing the quantity to be integrated is ON and not incremented if it is OFF. If the counter has N +1 states, corresponding to an output range of 0 to +V, then the value of the integral when it is in its k'th state is k/N.

In the bipolar cases, a reversible counter is required since both positive and negative quantities occur. In the bipolar single-line representation, the counter is incremented by one if the input line is ON and decremented by one if it is OFF. In the bipolar two-line representation, the counter is incremented by one if the UP line at the input is



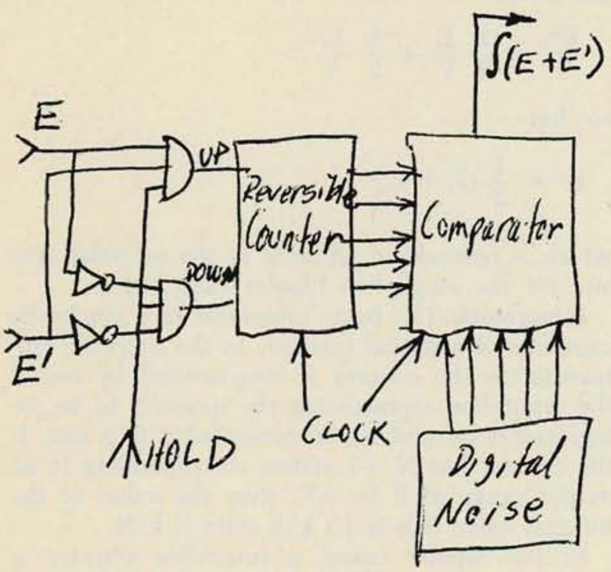
ON, decremented by one if the DOWN line is ON and does not change its count if the two lines are both ON or both OFF. If the counter has N +1 states, corresponding to an output range of $-V$ to $+V$, then the value of the integral when it is in its k'th state is

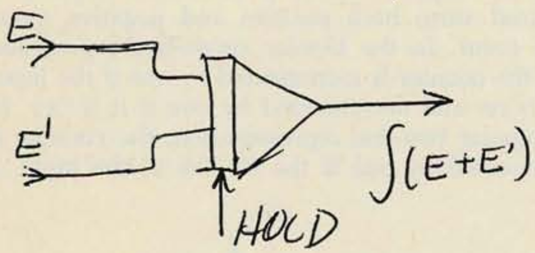
$$\int E d\tau = \left(\frac{2k}{N} - 1\right) V$$

This quantity, to be used in further computations, must be made available as a stochastic sequence. The sequence is generated by comparing the binary number of the counter with a uniformly distributed, binary random number obtained from a central pseudo-random shift register or a sampled cycling counter.

In the unipolar and single-line bipolar representations, the comparator output line is ON if the stored count is greater than the random number.

Two-input stochastic integrators with gating at the input of the counters, form the integral of the sum of the input lines. A HOLD line prevents the



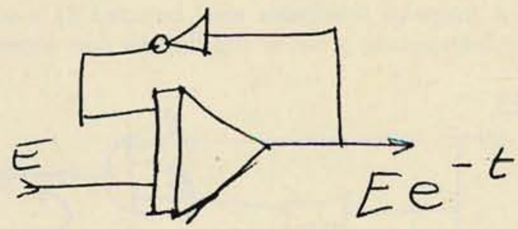


input lines from affecting the counter and an integrate or hold mode. The normal integrator symbol is used for the over-all device.

Readout with addies

An integrator with unity feedback is called an addie (adaptive digital element). It averages the quantity represented at its input, weighted by a decaying exponential term, so that past values have progressively decreasing effect on the integral. The analog equivalent of an addie is a leaky integrator (an integrator with resistive negative feedback) with the transfer function $1/(s + 1)$.

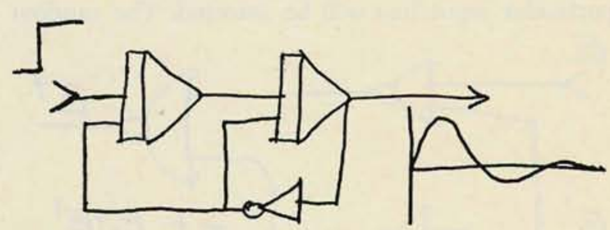
In terms of the stochastic sequences the fractional count in the addie tends to an unbiased estimate of the probability that the input line will be ON at a clock pulse. The time taken by an addie



to form an estimate of this probability is directly proportional to the number of states of its counter. The probable error in the estimate, however, is inversely proportional to the square root of the number of states. Thus any quantity represented by a probability in the stochastic computer may be read out with an addie but the more states, the longer the time constant of smoothing and the lower the bandwidth of the computer.

Integrators or addies have binary representation of quantity in their counters and form the natural output interface of the stochastic computer. Integrators with HOLD lines OFF also form the input interface for digital or analog data, since binary numbers may be transferred directly into the counter to generate a stochastic output sequence, and analog quantities may be easily converted to binary form.

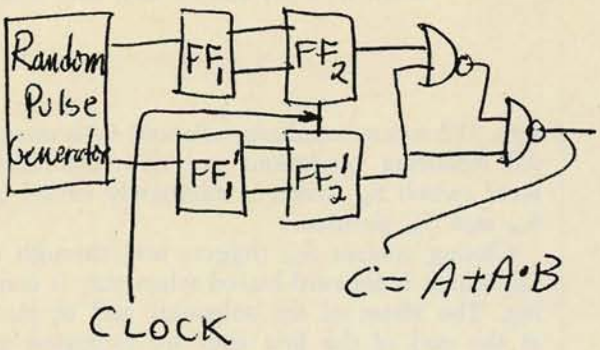
Similarly an integrator may be used to hold a multiplying constant, and thus act as an analog potentiometer if coupled to a multiplier. Arbitrary functional relationships may be realized by imposing a suitable nonlinear relationship between the stored count and the stochastic output; for example, to represent a switching function in the single-line bipolar case, use an integrater whose output is on when the count is equal to or above mid-value, and off when it is below mid-value. A pair of integrators coupled with appropriate stabilization can be used to generate sine and cosine functions. The generation of damped harmonic waveforms with two stochastic integrators is below.



Generation of stochastic sequences

The central problem in constructing a stochastic computer is the generation of many random sequences. Each integrator requires as many separate sequences as there are flip-flops in the integrator counter. The sequences must be neither crosscorrelated nor autocorrelated, and must have known, stable generating probabilities.

The independent sequences, however, need only have a probability of ½, since any other probability may be obtained by appropriate gating. One technique for generating stochastic sequences with a probability of ½ uses sampled flip-flops toggling rapidly from a noise source. NAND gates may be



$$p(C) = [1-p(A)] + p(A)\,p(B) = \frac{3}{4}$$

used to convert a number of these sequences to one sequence with any required generating probability. Generating digital noise in this way is attractive since radioactive or photon emitting sources may be coupled directly to semiconductor devices to form random pulse generators.

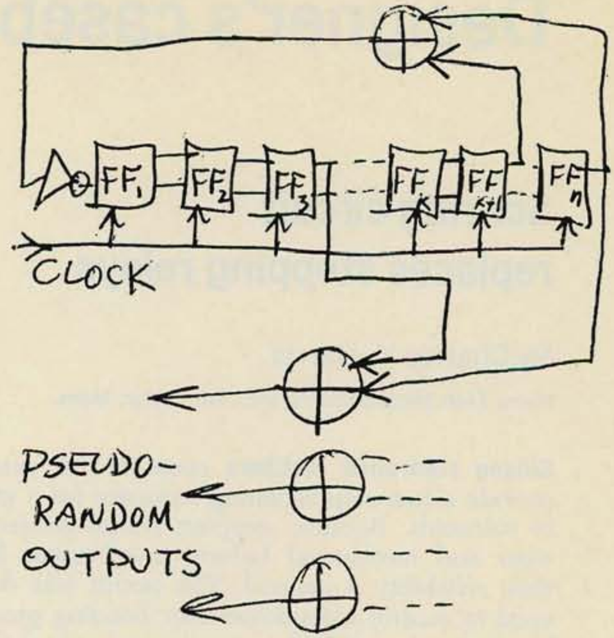
The Mark I stochastic computer built at Standard Telecommunications Laboratories had six 10-bit integrators, each with its own internal digital noise source consisting of 10-bit counters cycling at a high clock frequency. These counters were sampled at a much lower anharmonic clock frequency to give an effectively random output. This was not a practical arrangement, however, since the sampling frequency had to be so low, 500 hz, that the overall bandwidth of the computer was only 0.1 hz. As an experimental tool, however, it did allow a check out of the configurations, such as the stochastic integrator, whose behavior is difficult to determine theoretically.

Pseudo-random shift registers

In the Mark II, now being constructed, entirely different techniques are used to reduce its size and cost, while increasing the clock frequency to 1 Mhz. A single pseudo-random shift register generates stochastic sequences for all computing elements. Different sequences for each element are obtained by appropriate exclusive-OR gating of the shift register outputs, giving delayed replicas of the sequence in the shift register itself. Such a generator, with 43 flip-flops, is capable of delivering random sequences to 100 16-bit integrators for one hour without cross-duplication.

Serial arithmetic is used in the integrators of the Mark II so that the counters may be built with shift registers and fewer gates may be used in the comparators. In this way a 16-bit stochastic integrator may now be fabricated from only six dual in-line packages. A clock frequency of 16 Mhz in the shift registers produces a clock frequency of 1 Mhz in the computer, and respectable bandwidths of 100 hz or so are attained.

The use of pseudo-random noise is experimental because little is known about its high-order dis-



tributions that may cause bias in computations. It is possible that conventional gates and flip-flops will not ultimately be used to implement the stochastic computer.

At particle level, random behavior is generally the rule, and stochastic computing may be the most direct means of utilizing the high-speed interactions between photon, electrons, alpha particle, and so on. OR-gates and attenuators are available for these particles, and only some form of coincidence gate and storage element is required for computing.

Application of stochastic computers

The stochastic computer is at a stage of rapid development where prediction of its future applications is difficult. Apart from designing the basic hardware, we have also carried out analytical and simulation studies of various learning-machine subsystems built with stochastic computing elements. The purpose has not been so much to design hardware for these subsystems, but rather to discover universal complexes of computing elements which will be suitable for large-scale integration.

The only immediate commercial application of stochastic computing techniques has been in a pseudo-stochastic machine for radar and aircraft navigation systems developed at STL. The conversion of range and bearing to x and y coordinates for the generation of the radar sweep on the plane-position indicator for instance, is normally performed by analog computing elements, which are subject to drift and some degree of unreliability. The all-digital stochastic computing elements may be used in a similar configuration to perform the same function with higher reliability and increased accuracy.

while erase time is a second and a half.

The first order of business at the Kingston labs is to solve the tipping problem.

Computers

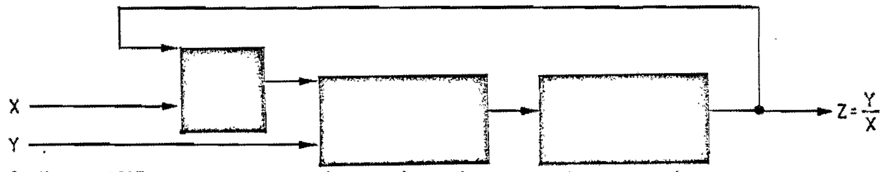
Dropping the guard

When a computer must work in an area cluttered with electromagnetic pollution, the traditional step is to protect its sensitive signals by shielding the entire machine with noise-absorbing screens. That approach is fine when the computer is being used in a factory, where space and weight are not at a premium. But in a missile or an airplane, where a few pounds and a few cubic inches count heavily, screening may be a bit of a problem. Now at the University of Illinois, a group of computer researchers has designed a series of analog computer circuits that depend on noise and therefore needn't be protected from it.

In the circuits, developed under the supervision of W.J. Poppelbaum, an electrical engineering professor, an analog quantity is represented by the average value of a sequence of randomly spaced pulses. The pulses are generated from white noise and a trigger level proportional to the analog signal.

Shaping the noise. The pulses are shaped and clipped so that all have the same duration and amplitude. Two or more such pulse trains applied to an ordinary diode AND gate, shown below, produce a random pulse sequence at the output; the average value of this sequence, after again shaping and clipping, is proportional to the product of the original analog quantity.

The quotient of two analog quantities can be obtained with a feedback network, shown below. In this network a random sequence generator produces a pulse train representing a quotient. An AND gate forms the product of the quotient and the divisor; the product is then compared with the dividend. If the two are unequal, the random sequence generator is automatically adjusted to make them equal.

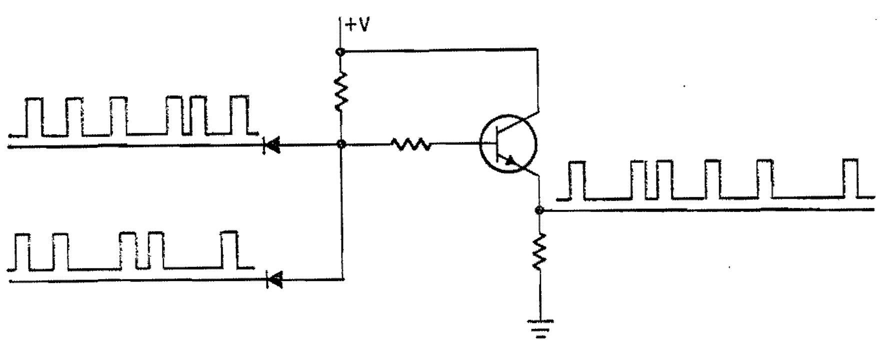


Ordinary AND gate, a comparator, and a pulse generator compute the quotient of two analog quantities.

The sum can be obtained just like the product, by using a diode OR gate, but a conventional adder network (a number of equal resistors with one common connection) is cheaper and works just as well. Likewise, the difference of two analog quantities is the sum of one and the logical inverse of the other; if the basic pulse train has positive-going pulses from a negative reference, the inverse would have the same voltage swing but negative-going pulses from a positive reference.

Problem-solving. With a sufficiently large number of units for performing the four basic arithmetic operations, theoretically almost any problem can be solved. The speed is limited only by the width and average frequency of the noise pulses; fractional nanosecond noise pulses can be obtained from a diode reverse-biased close to the avalanche point, providing a frequency response in the neighborhood of 500 megahertz.

The principal limitation to the circuits, as with any analog system, is in their precision. For example, a problem might be solved to a precision of 1% by taking the average value of 1,000 noise pulses; but 10 times the precision, or 0.1%, would require an average of perhaps 100 times the number of pulses. A very complex analog computer with a limited degree of precision can be designed at a fraction of the cost of ordinary analog machines that use operational amplifiers; but the circuits cannot economically approach the precision of digital computers.



Analog multiplication, using noise, can be performed with an AND gate.

AIAA meeting

Closer link

The dinosaur is extinct because it couldn't respond fast enough to attacks from the rear. The communications network between brain and tail was too lossy.

The Avco Corp.'s Electronics division in Cincinnati has developed a new approach to airborne high-frequency communication in order to prevent lossy communications aboard flying behemoths. The system will go into the C-5A, a heavy logistic transport aircraft being built for the Air Force by the Lockheed Aircraft Corp.

To avoid stringing a lossy coaxial cable 260 feet from the cock-