Technical Report No. 377
Probabilistic Error Correction in Arbitrarily Large Łukasiewicz Logic Arrays

R. A. Montante, J. W. Mills
Computer Science Department
Indiana University
Bloomington, Indiana

April 21, 1993

Abstract

Lukasiewicz logic arrays (LLAs) are analog VLSI circuits structured as trees of continuous-valued implications. First described in 1990, LLAs are now finding uses as application- specific array processors for image processing, fuzzy inference, and robotic control.

Yet an early question persists: Why do LLAs work?

Intuitively, a small error should build up through successive implications of an arbitrarily deep Lukasiewicz logic array (LLA) and render the output unusable, yet LLAs fabricated in normal MOSIS runs are typically precise to 10 bits. Analog neural nets are comparably robust.

We compared the information capacity of a 31-cell LLA and a baseline circuit consisting of simple wires. The results indicate that the LLA preserves approximately 3/4 bit less information per input than the baseline circuit.

The magnitude of the error is not a function of the number of implications in the LLA. Functional and electrical simulations suggest that information loss in an LLA increases less than linearly with circuit size. Our hypothesis, based on information theory, is that random noise in one node partially cancels the error introduced by previous nodes. In arbitrarily large LLAs random noise acts to correct errors in analog values without having to encode them. The LLA is inherently and probabilistically error-correcting.

We close by discussing circuits that we are fabricating to verify the hypothesis by measuring information loss at multiple points within an LLA.

1 Introduction

Algorithms and computations based on alternatives to Boolean logic have been explored in robotic and process control. One such alternative is the continuous-valued Łukasiewicz logic $Ł_\infty$. The Łukasiewicz truth-value domain runs from zero or **False** to one or **True**. This is the natural domain of analog circuits, which can simply and rapidly compute continuous-valued logical expressions. Their size and complexity advantages are compounded by their ability to operate on "real-world" inputs and outputs without needing analog-digital conversions.

Analog VLSI logic circuits offer significant advantages over digital logic circuits in terms of circuit size and communication requirements with their external environment. [1] However, questions remain about the precision, or equivalently the information content, of the values being computed.

A digital circuit represents values by a specific number of bits, each with its own datapath. This determines the number of distinct possible values and maximum information content of them. An analog circuit represents the entire value on a single path, with a precision limited by the noise level in the circuit. Random noise might be expected to accumulate through successive stages of an array of circuits until the information content of the value is completely gone. The analog logic circuits discussed here are surprisingly resistant to this cumulative information loss. Analog circuits with feedback can even reach a point of no further loss to computations, as in an electronic Analog Computer producing a resonant solution to a differential equation.

In this paper we examine the information capacity and information loss in a binary tree of analog circuits which compute Łukasiewicz implication. Claude Shannon's information theory ([SW63], [Ing71]) provides the basis for calculating the information content of a collection of transmitted values. Calculations of information loss show that circuit-induced errors tend to partially cancel each other out between successive stages, so cascades of circuits retain the information content of their signals better than the single-circuit performance would predict. We also discuss measurements of information transmission and loss in a binary tree of analog circuits fabricated on a MOSIS chip.

[1] roughly measured by the number of pinouts required.

2 Łukasiewicz Logic and Circuit Trees

Łukasiewicz ([ŁT83]) described a family of multivalued logics, denoted $Ł_3, Ł_4 \ldots Ł_\infty$ according to the number of distinct logical values. These logics are characterized by their logical functions. Implication is a primitive operation defined by

$$(\alpha \rightarrow \beta) \equiv \min(1, 1 - \alpha + \beta).$$

Implication combined with the logical constants 0 and 1 can express any other function in the logic. In general, any compound logical expression in Łukasiewicz logic can be expanded into a binary tree of implications. An example of this is shown in Figure 1, where the four-term notch() function is defined and partially expanded.

Simple regular circuit structures exist which compute the implication function for variables represented as analog current flows; these can be efficiently packed into binary trees. Mills et al [MBD90] describe the ŁL9 VLSI chip, a 31-node analog logic circuit. It is capable of evaluating a variety of fuzzy membership functions, including the fully expanded notch() function.

3 Cascaded Information Loss

Values lose information as they propagate through a circuit, so that at each stage in a cascade of computing circuits the entropy is less. If a stage loses half a bit of entropy it would seem that the information content in a 12-bit value would be lost after 24 stages of computation. This doesn't happen because the information loss in each stage is dependent on losses in previous stages. This can be seen by considering the way information is defined and manipulated, and how the entropy of each stage is determined.

3.1 Measuring information

Information theory provides a way to measure the information content of a set of message symbols or numeric values. A symbol or value m, selected from a set M, conveys an amount of information given by the (negative) base-2 logarithm of the probability p_m of selecting of that particular value from the set. If the set M is discrete, as it is when the set of messages contains only

$$
\begin{aligned}
(\textbf{True})\ 1 &\equiv 0 \to \alpha \qquad (0 \le \alpha \le 1) \\
(\textbf{False})\ 0 &\equiv 1 \to 0 \\
\text{Identity}(\alpha) &\equiv 1 \to \alpha \\
\neg\alpha &\equiv \alpha \to 0
\end{aligned}
$$

$$
\begin{aligned}
\text{notch}(\alpha) &\equiv (\neg\alpha \to \alpha) \to \neg\,(\alpha \to \neg\alpha) \\
&= ((\alpha \to 0) \to (1 \to \alpha)) \to \neg\,((1 \to \alpha) \to (\alpha \to 0)) \\
&= ((\alpha \to 0) \to (1 \to \alpha)) \to (((1 \to \alpha) \to (\alpha \to 0)) \to 0)
\end{aligned}
$$

$$\vdots$$

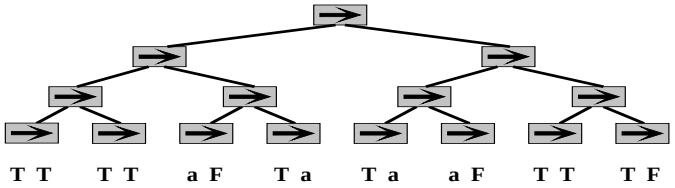

Figure 1: Expanding the notch() function into a binary tree of implications. The input α and the constants can be further expanded to produce a 4-deep balanced tree.

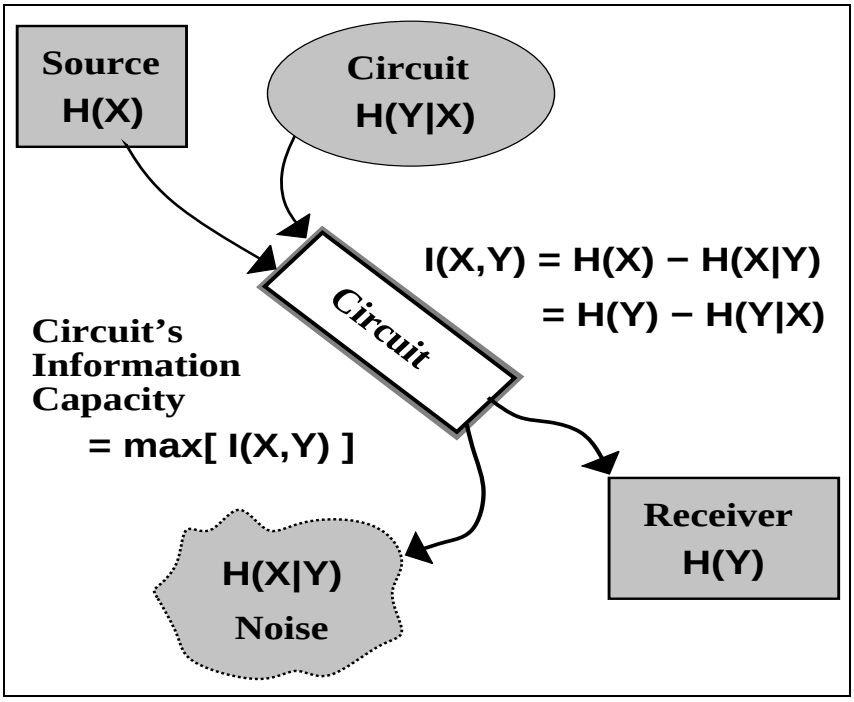

Figure 2: Entropy in a transmission channel or circuit. The source may consist of one or more distinct signals; the total received information may also be divided among multiple pathways (as for example, a full adder with a carry-out bit).

the Boolean values **True** and **False**, then its average information per value or *entropy* is given by

$$H_M \equiv -\sum_{m \in M} p_m \lg(p_m)$$

Values from a continuous set are distributed according to a probability density function $p(m)$, and the entropy is given by the integral over the range of possible values:

$$H_M \equiv -\int_{m \in M} p(m)\ \lg(p(m))\, dm$$

An information-transmitting system is diagrammed in Figure 2. A source emits values from some set X; a communication channel or circuit maps the values into a set Y at a receiver. Noise may distort the source, so that for a given received y there is uncertainty about the original value of x. The circuit itself may be subject to error, mapping a given source value x to y values according to some probability distribution $p(y|x)$.

An entropy may be calculated for each of these elements, based on the associated probabilities. The average information sent per value is $H(X)$; receiver entropy $H(Y)$ measures the collected information. In an ideal channel all the source information is transmitted to the receiver and $H(Y) = H(X)$.

In practice some of the source entropy is lost as noise entropy $H(X|Y)$, and some of the received entropy comes from the circuit $H(Y|X)$ rather than the source. The net information transmitted by the system is the source entropy less the noise entropy, $H(X) - H(X|Y)$; or equivalently the received entropy less the circuit's contribution, $H(Y) - H(Y|X)$. Information theory then defines the circuit capacity to be the maximum amount (or rate) of net information transmission, over all probability distributions of input values (the other probability distributions being fixed properties of the circuit).

Information and computation. A computing circuit loses information in the process of computation: its outputs contain less entropy than its inputs.[2] The term $H(Y)$ in Figure 2 becomes $H(Y) + H_{\text{computation}}$ if the circuit does computation on the source inputs. For many computations this effect is implicit in the (usually more interesting) values of the inputs and

[2]In reversible circuits, the "excess" entropy is diverted to another output so that the output total equals the input total.

Boolean Implication:

input α	0	0	1	1
input β	0	1	0	1
$i = \alpha \rightarrow \beta$	1	1	0	1

$$
\begin{aligned}
H(\alpha) = H(\beta) &= -\tfrac{1}{2} \cdot \lg(\tfrac{1}{2}) - \tfrac{1}{2} \cdot \lg(\tfrac{1}{2}) = 1 \text{ bit} \\
H(\text{inputs}) &= H(\alpha) + H(\beta) = 2 \text{ bits} \\
H(i) &= -\textstyle\sum_i p_i \lg(p_i) \\
&= -\tfrac{1}{4}\lg(\tfrac{1}{4}) - \tfrac{3}{4}\lg(\tfrac{3}{4}) \approx 0.915 \text{ bits} \\
H_{\alpha\rightarrow\beta} &= H(\text{inputs}) - H(i) \approx 1.085 \text{ bits}
\end{aligned}
$$

Łukasiewicz Implication: $i = \min(1, 1 - \alpha + \beta)$

$$
\begin{aligned}
H(\alpha) = H(\beta) &= -\textstyle\int_0^1 \frac{1}{1} dx = 1 \text{ bit} \\
H(\text{inputs}) &= H(\alpha) + H(\beta) = 2 \text{ bits} \\
H(i) &= -\textstyle\int_0^1 p(i) \lg(p(i))\, di \\
&= -\textstyle\int_0^1 \int_0^\alpha p(i) \lg(p(i))\, d\beta\, d\alpha - \int_0^1 \int_\alpha^1 p(i) \lg(p(i))\, d\beta\, d\alpha \\
&= \tfrac{1}{2} + \tfrac{1}{3} \approx 0.833 \text{ bits} \\
H_{\alpha\rightarrow\beta} &= H(\text{inputs}) - H(i) \approx 1.167 \text{ bits}
\end{aligned}
$$

Figure 3: Entropies in the implication function.

outputs themselves. Neural nets represent computations whose inputs and outputs are not so accessible. Recent neural processing conferences have aired papers on the amounts of information required as inputs and as weights for threshold functions ([NIPS91]).

Figure 3 illustrates the entropy decrease in the implication function. The uniformly distributed Boolean inputs contain 1 bit of entropy apiece, while the output carries just $\frac{9}{10}$ bit. Thus the computation itself has removed an entropy $H_{\rightarrow}$ of almost $1\frac{1}{10}$ bits from the inputs to produce the output. Łukasiewicz implication consumes even more entropy as the precision of the inputs increases. The computation reduces an input entropy of roughly 3.17 bits, or three truth values per input, to an output entropy of 1.22 bits; with 12 bits of input entropy (4096 truth values) the output entropy is reduced to 6.86 bits.

3.2 Successive circuits

Intuitively, the rate of information loss in a cascade of less-than-ideal circuits is cause for concern. However, previous simulations of cascades of analog Łukasiewicz-implication circuits have indicated that the precision does not diminish linearly with the the length of the cascade. Calculations of intermediate entropies in a cascade of circuits bear this out. The circuit model assumes a finite discrete set of inputs (from a Digital-to-Analog Converter, for example) to a trivial circuit computing the identity function (e.g., a wire). The circuit tends to disperse its input according to a gaussian probability distribution,

$$p(y|x) = \frac{1}{\sqrt{(2\pi)}\sigma} e^{-\frac{(y-x)^2}{2\sigma^2}}.$$

This produces a somewhat rippled, continuous output distribution which is the input distribution for the next stage, another copy of the same circuit. σ is somewhat arbitrarily set at $\frac{1}{40}$ of the full range, so that 97% of the distribution is spread over no more than $\frac{1}{10}$ of the full range.

A sequence of output distributions was simulated, and the entropies were numerically integrated at each stage to model this effect. Plots of the distributions and the entropy losses are shown in Figure 4. The entropy loss at each stage is less than at the previous one. This is because the probabilistic dispersion at each stage is equally likely to correct a prior-stage deviation

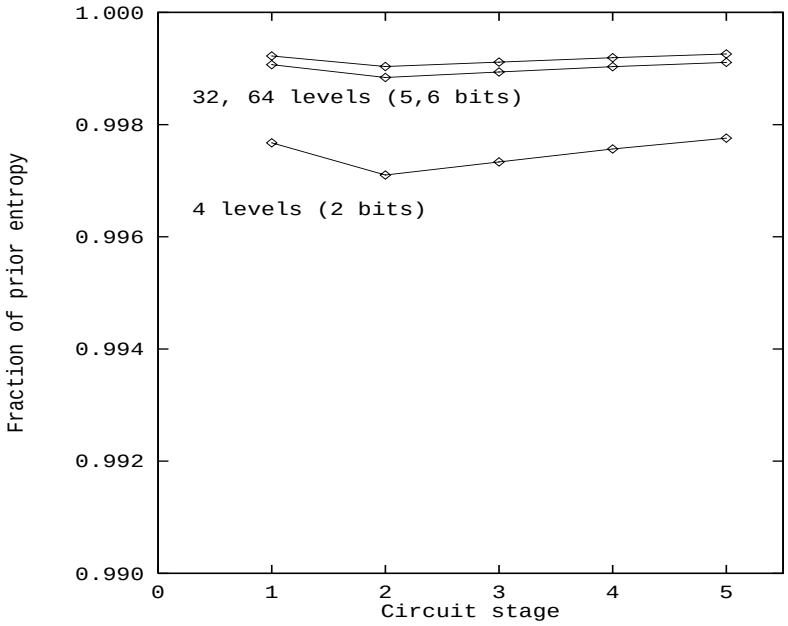

Figure 4: Entropy at each stage of a circuit cascade, as a fraction of the prior-stage entropy. Each stage loses less entropy than the previous one, relative to the entropy of its inputs and also in absolute terms. Three different starting entropies are shown. (Continuous densities are approximated by discrete distributions at three levels; a glitch in the initial-stage/1st-stage calculation diminishes with fineness of approximation.)

or increase it. In effect, the circuit partially corrects prior errors despite adding its own. The overall result is that larger circuits are relatively better at maintaining precision than smaller ones.

4 Testing an Analog Circuit

We measured the information capacity and entropy loss in a VLSI analog logic chip. ŁL9 is an array of 31 Łukasiewicz-implication circuits connected as a five-deep H-tree. It can evaluate the notch() function, a five-stage cascade of identity functions, or other functions by appropriate selections of inputs and constants. The chip's programming and early testing are described in Mills et al (*ibid.*)

4.1 The test device

ŁL9 is a current-mode device, with inputs on the order of magnitude of $100\mu A$. A testing device called the Analog Test Board or ATB generates the required 30 current-mode inputs and captures the output currents. It incorporates 12-bit Digital-to-Analog converters driving op amps to generate 36 separate input currents plus some reference currents and voltages, and converts up to 5 analog-current outputs through 12-bit Analog-to-Digital converters. A desktop computer supplies 12-bit input values, reads 12-bit outputs, and controls all board functions through a parallel interface.

4.2 The test

The input-value and output-value sets for an analog circuit tested on the ATB are discretized with 12 bits of resolution. Assuming a uniform distribution of inputs, this supports a source entropy of 12 bits. It also allows entropy calculations in the form of summations over the discrete distributions; the alternative would be to determine probability densities empirically and integrate over those. The total current range for each input and output is from 0 to $100\mu A$, so the values are spaced in steps of $0.0244\mu A$.

The entropy test for the ŁL9 circuit involves supplying every distinct input value and recording the resulting output value, many times. In the most general case this would require $(2^{12})^{30}$ distinct inputs, with data quantities

rather exceeding the desktop computer's storage capacity. Programming the ŁL9 to compute the single-variable notch() function pares this down to 26 constant inputs and 4 inputs which receive the same value, or merely 4095 distinct input values. Typically each input value x is tested 600 times and the distribution of output values is recorded for each input. [3] The source entropy is dictated by the input-sampling scheme; for uniform sampling it is precisely 12 bits. The receiver or output entropy is calculated from the overall distribution of recorded outputs Y, while the circuit entropy is calculated as the average over all inputs of the entropy in the output distributions for each input. Noise entropy is calculated analogously to circuit entropy, by permuting the recorded data into distributions of inputs for each output.

The "circuit" in a test like this includes more than just the analog circuit being studied. It includes the DACs and the ADC, the coaxial cables which carry the input and output currents, the chip carrier and the plugboard it's plugged into, and the standard I/O pads used in the circuit design as well as the active analog circuit itself. These extraneous circuit components are compensated for by repeating the test on a "null" chip: a VLSI chip fabricated in the same manner as ŁL9, but with a simple wire connection between the input and output pads. Logically such a circuit computes the Identity() function, since the output of the wire should be exactly its input. The null chip used includes a short 27λ wire formed in the M2 aluminum layer, with a pin-to-pin resistance of 64Ω. Any information loss due specifically to this wire is uncompensated for, although it could be estimated by comparing circuit entropies of null chips with varying wire dimensions.

5 Measured Results

The 600-repetition entropy test described above was run on an ŁL9 chip programmed for the notch() function and subsequently on a null chip using the "m27" wire. In the result data, the most frequently-occurring output value for each input was identified as the Expected (or "correct") value, and other output values expressed relative to the expected value. The resulting relative (conditional) distributions were pooled across all inputs, and plotted

[3] Even this test size exceeds the PC's capacity; the input range is cut into four subranges and each subrange is tested separately. Testing the total range requires about 26 hours of running time.

in Figure 5 to show a subjective comparison of the repeatability of ŁL9 compared to the null circuit. The null circuit does rather better than the ŁL9 circuit: the null-circuit output falls more than 1 count away from the expected value, i.e. $\pm\frac{1}{4096}$ of full-scale, less than 5% of the time. In contrast, ŁL9 outputs fall more than 2 counts away ($\pm\frac{2}{4096}$ of full-scale) more than 6% of the time.

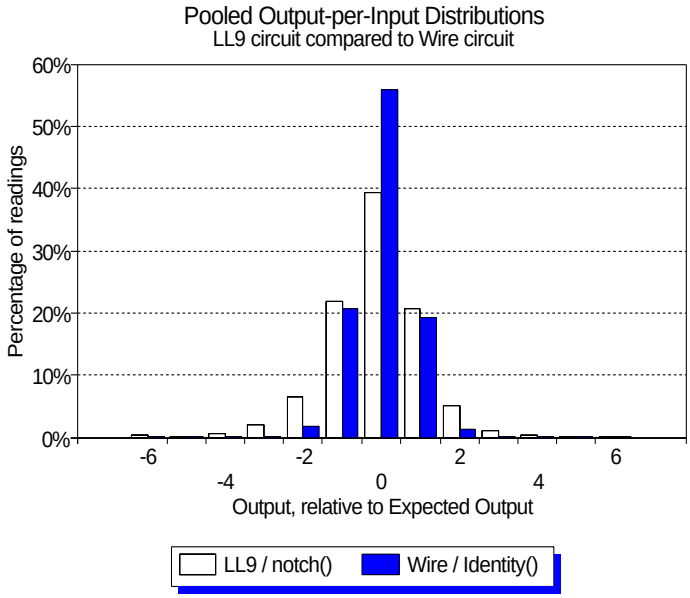

Figure 5: Distributions of output values relative to statistically expected output values, cumulative over all inputs. ŁL9test on the left, m27 test on the right.

Plots of the expected output versus input (Figure 6) illustrate the function that each circuit computes. Some of the ATB's reference signals were imperfectly tuned to the ŁL9 circuit, so that output levels were elevated by some 700 counts and outputs saturated for extreme input values. The saturation reduced the circuit entropy as well as receiver entropy. To compensate for this error in the testing procedure the saturated points were excluded from entropy calculations.

Figure 7 compares the entropy terms for the two circuits. The source entropies reflect the fact that four points in the input range were sampled

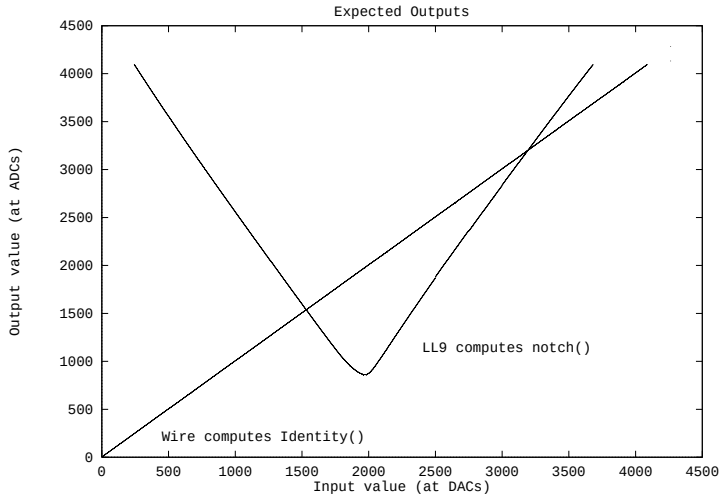

Figure 6: Expected outputs of ŁL9computing the notch() function, and m27 wire computing the Identity function.

	ŁL9/ notch()	Wire (m27)
Number of Input Samples	$2,071,200$	$2,461,200$
H_{inputs} $H(X)$	11.745	11.995
average H_{circuit} $H(Y\|x)$	2.093	1.406
average H_{noise} $H(X\|y)$	2.224	1.444
H_{outputs} $H(Y)$	11.613	11.956
average information per input value $I(X,Y)$	9.520	10.550

Figure 7: Entropies calculated from measured output distributions.

twice as often as the rest. The source entropy also shows the effect of excluding the saturated test points. The average information carried by an input in the m27 test was about $10\frac{1}{2}$ bits, and in the ŁL9 test was about $9\frac{1}{2}$ bits; allowing for the cascade effect, somewhat more than one bit of entropy was lost in the implication-tree circuitry.

6 Conclusions

These circuit measurements demonstrate that entropy measurements can be applied to analog logic circuits. A second-generation analog logic circuit has been fabricated, which incorporates a significantly simpler architecture for the $(\alpha \rightarrow \beta)$ function and thus is somewhat smaller. It will permit entropy determinations for circuit cascades of varying lengths, to fully verify the calculated effects of arrays of analog circuits.

The measurements described here are "steady-state" in that the circuit was given ample time to stabilize its output values. The next application of this approach is to determine the circuit's performance over a range of operating speeds. It is anticipated that as an analog circuit computes its outputs, they approach the "correct" values for given inputs smoothly in time. The information capacity will increase steadily from none at excessive operating speeds to the levels measured here at sufficiently low speeds. The product (information capacity/value) $\times$ (values/second) is the bandwidth, which will reach a maximum at some point in this range of operating speeds.

7 Acknowledgements

J. Mills and C. Daffinger designed the ŁL9 chip for fabrication by the MOSIS facility. C. Daffinger designed and built the Analog Test Board, and performed the original testing of ŁL9.

References

[Ing71] Franklin M. Ingels. *Information and Coding Theory.* Intext Educational Publishers, Scranton, PA, August 1971.

[ŁT83] Jan Łukasiewicz and Alfred Tarski. Investigations into the sentential calculus. *Logic, Semantics, Metamathematics*, chapter IV, pages 38–59. John Corcoran, editor. Hackett Publishing, Indianapolis, IN, second edition, 1983.

[MBD90] Jonathan W. Mills, Gordon Beavers, and Charles A. Daffinger. Łukasiewicz Logic Arrays. Technical Report 296, Indiana University, Bloomington, IN, March 1990.

[SW63] Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, Urbana and Chicago, 1963.

[NIPS91] David S. Touretsky, Richard P. Lippmann, John E. Moody, and Steve J. Hanson, editors. *Neural Information Processing Systems 1-4*, San Mateo, CA, 1988-1991. Morgan Kaufmann. (Papers in each of the conferences discuss neural-net views of the information requirements of analog circuits. Noteworthy titles include 'The VC-Dimension versus the Statistical Capacity of Multilayer Networks', 'Analog Neural Networks of Limited Precision I: Computing with Multilinear Threshold Functions', 'ε-Entropy and the Complexity of Feedforward Neural Networks'.)