TECHNICAL REPORT NO. 312

CMOS VLSI Łukasiewicz Logic Arrays

by

Jonathan W. Mills and Charles A. Daffinger

July 1990

COMPUTER SCIENCE DEPARTMENT
INDIANA UNIVERSITY

Bloomington, Indiana 47405-4101

CMOS VLSI ŁUKASIEWICZ LOGIC ARRAYS

JONATHAN WAYNE MILLS*
CHARLES A. DAFFINGER‡

Indiana University
Bloomington, Indiana 47405

Abstract

Łukasiewicz logic arrays (ŁLAs) are massively parallel analog computers organized as binary trees of identical processing elements. We have designed and performed preliminary tests on a series of CMOS VLSI ŁLAs whose cells perform Łukasiewicz implication (→). In this paper we describe the ŁLA architecture and its relationship to cellular automata, the CMOS VLSI implementation of versions LL9 and LL10 and their initial characterizations, and report on the results obtained by programming ŁL9 and ŁL10 as fuzzy function recognizers, the first step in designing more general function units such as expert systems and neural networks.

1. INTRODUCTION

Łukasiewicz logic arrays (ŁLAs) are massively parallel analog computers. They are organized as binary trees of identical processing elements (called PEs, or cells), each PE performing either Łukasiewicz implication (→), negated implication (↛) or both. We have designed and built two versions of a 31-cell CMOS VLSI ŁLA whose cells perform Łukasiewicz implication (→), ŁL9 and ŁL10. The success of the prototype has encouraged us to continue research in the design and application of ŁLAs. During this research we have observed that ŁLAs offer advantages as massively parallel analog computers.

1.1 Advantages

ŁLAs are regular VLSI architectures. VLSI implementations of ŁLAs are simple and area-efficient because they are derived from cellular automata, and implemented with analog rather than digital processing elements. Although ŁLAs are analog computers they can be

* 101 Lindley Hall, Department of Computer Science, (812) 855-7081, jwmills@iuvax.cs.indiana.edu

‡ 101 Lindley Hall, Department of Computer Science, (812) 855-6486, cdaf@iuvax.cs.indiana.edu

made precise (5 to 8 bits), due to the simplicity of their processing elements and the accuracy of VLSI process technology.

ŁLAs are inductive architectures, which means that they can be expanded by adding more processing elements without redesigning the interconnection network. While small ŁLAs can be used as circuit components, large ŁLAs can be used as massively parallel computers. Larger ŁLAs can be created by cascading individual ŁLAs.

The general-purpose nature of ŁLAs is theoretically well-founded. Multiple-valued logics used in computational networks are capable of both symbolic and algebraic computation. ŁLAs can implement fuzzy inference and expert systems [1], neural networks [2, 3], and algebraic functions [4, 5]. Viewed as circuit components, ŁLAs are one form of multiple-valued programmable logic arrays (PLAs).

The processing elements are simple, performing only Łukasiewicz implication ($\rightarrow$) to evaluate the sentences in Ł defined by the schema. Processing elements need only two input wires and one output wire because they use analog values. Thus, the bus structure of the ŁLA is also area-efficient.

However, one drawback to an area-efficient circuit is that it is limited by the number of pins available on existing VLSI circuit packages. Although an array of 1024 Łukasiewicz implication ($\rightarrow$) cells could easily fit onto a 4500μ x 2300μ chip, it would require 2048 input pins and 1 output pin. This is 1,921 more pins than are available on a 128 pin-grid array package. Our research has shown that many functions implemented with ŁLAs will have more than half of their inputs tied to *true* or *false*. For these functions ŁLAs can be built that use a programmable interconnection network to route internally replicated *true* and *false* inputs to the PE array. Data inputs also tend to be used more than once, so they could be internally replicated and routed also. This approach allows large ŁLAs to fit into existing VLSI packages.

1.2 Disadvantages

Of course, ŁLAs are not ideal analog processors, but we are working to reduce their drawbacks.

The prototype ŁLAs are programmed using sentences in the Łukasiewicz logic structured as binary trees. This introduces data inputs on the order of $O(2^n)$ for sentences in n implications, limits the size of the sentences that can be evaluated by a given ŁLA, and increases the number of pins needed on the VLSI package, far beyond the number available even in the foreseeable future. However, many data inputs are *true* or *false*, or are composed of a repeated number of variable inputs. Based on this observation we are designing ŁLAs that have external control inputs, and a restricted number of external data inputs. The data inputs are replicated and selected internally at each input of the processor array according to the externally applied control inputs.

Because ŁLAs are a new form of computational engine their applications are still being studied. We have only a basic understanding of the programming methodology for ŁLAs. For example, the theoretical applicability of ŁLAs as neural networks does not immediately lead to the construction of algorithms for back-propagation.

ŁLA programming is an instance of the more general problem of programming analog and hybrid digital-analog computer architectures. Research in this area stopped about 1970 due to the dominance of digital computers. Because ŁLA-based systems will be either analog or hybrid digital-analog computers we must develop programming languages for them. Mills and Faustini [6] have proposed a language for ŁLA-based systems, but its operational semantics is not fully defined. Completion of the semantics will require a better understanding of the dynamic behavior of ŁLAs, particularly ŁLAs with cyclic interconnections.

2. ARCHITECTURE

2.1 Design

Łukasiewicz logic arrays resulted from research into cellular automata as parallel architectures for logic programming. Cellular automata are of particular interest because they lead to area-efficient VLSI architectures. Such architectures are implemented as regular arrays of processing elements which communicate the results of their computation locally. They are derived by instantiating a portion of a cellular automaton as a VLSI circuit [7]. Cellular automata model a wide variety of parallel computational devices. Examples include the systolic architectures of Kung and Lieserson [8], the stochastic neural machines of Alspector et. al. [9, 10] and the analog VLSI computers of Mead [11].

Ideal Łukasiewicz logic arrays (ŁLAs) are cellular automata that implement a denumerably infinite sentence schema of some Łukasiewicz logic, Ł. The sentence schema of Ł and the cellular automaton C are related by requiring the logical variables of Ł to correspond to cells in the cellular space S, the structure of the sentence schema to correspond to the neighborhood function g, and the connectives of Ł to correspond to the transition function f of ϕ.

Real Łukasiewicz logic arrays are derived by restricting the denumerably infinite sentence schema of Ł to a finite sentence schema, and implementing the finite cellular automaton that results as a direct correspondence architecture. The structure of the resulting ŁLA is dependent on its interconnection network. The prototype ŁLA uses an H-tree network whose nodes are the processing elements corresponding to the connectives in the finite sentence schema. The H-tree network was selected for its efficient use of area on a VLSI circuit, as first proposed by Leiserson [12].

2.2 VLSI Implementation

Łukasiewicz logic arrays are implemented with analog processing elements. A cell in the ŁLA is implemented as an analog current-mode device performing addition, subtraction, *min* and *max* on currents. Early in our work we learned of a series of fuzzy functions implemented by a basic logic cell [1]. The circuits which implement these functions also implement the algebraic valuation functions for Ł. For our purposes the most useful of Yamakawa's circuits are implication ($\rightarrow$), which computes $min(1, 1 - \alpha + \beta)$, and bounded difference, which computes $max(0, \alpha - \beta)$. The original ŁLA cell used Yamakawa's components directly, but the more recent versions, ŁL9 and ŁL10, simplified the design by combining all components into a basic cell (Figure 1). ŁL10 adds two more

current mirrors to provide for an additional reference voltage, intended to increase the range of the circuit.

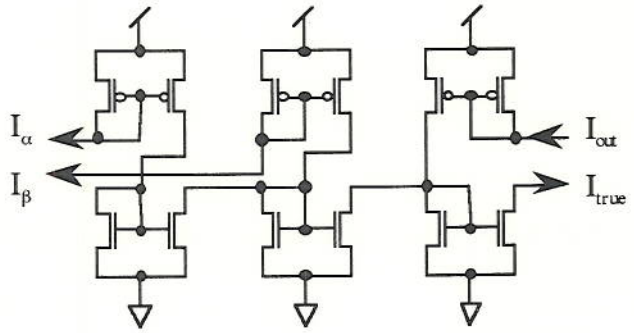

Figure 1. Schematic of ŁL9 implication-only cell (→)

The basic cell consists of six current mirrors, and performs Łukasiewicz implication (→). A cell has two inputs and a single output, and is designed to be tiled in an H-tree. The basic cell uses 11 transistors, and is 35μ by 114μ using the 2μ ORBIT SCPE technology provided by MOSIS. Basic cells are combined in an H-tree to form the ŁLA (Figure 2).

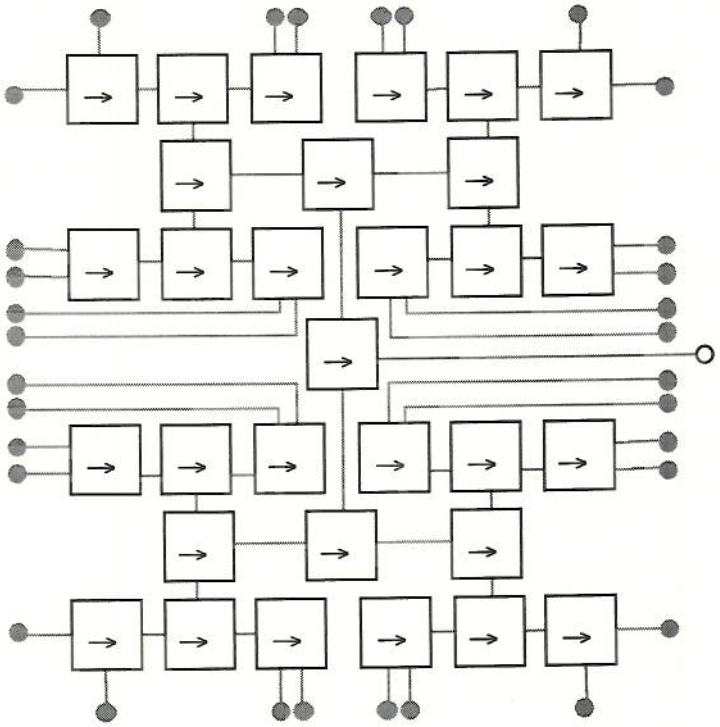

Figure 2. Architecture of ŁL9, a heterogeneous ŁLA in implication (→)

Both ŁL9 and ŁL10 have been tested, and found to be functional. The initial evaluation of these prototypes is described in the next section. A more detailed characterization is planned using an automated test fixture, which is currently being constructed.

2.3 Evaluation

ŁL9 has been tested with Vdd voltages ranging from 1.5 to 7 volts, and input and output currents varying from 0 to 200 microamperes (μA); ŁL10 has been tested with Vdd voltages ranging from 3 to 7 volts, Vss ranging from –3 to 0 volts (ground), and input and output currents varying from 0 to 400 μA.

Within this range the accuracy of the ŁLA is affected by four sources of error. The first is steady-state error, which is dependent on the final fabricated dimensions of the transistors and other process parameters for a particular MOSIS run. The second source of error is temperature dependent, and varies as the temperature changes over long periods of time. As long as the temperature of the system in which the ŁLA is placed varies uniformly this error can be ignored. The third source of error is transient error which arises when large current swings occur in the inputs of the ŁLA, and lasts until the cell has stabilized. A fourth source of error results from leakage currents in the CMOS devices, which have not been characterized at this point, but are typically small.

Initial tests of ŁL9 showed that the five-level ŁLA reproduced its input at the output linearly beyond 20μA. Below 20μA the error ranges from 4.5% at Vdd = 5 volts to 0.75% at Vdd = 1.5 volts (Figure 3).

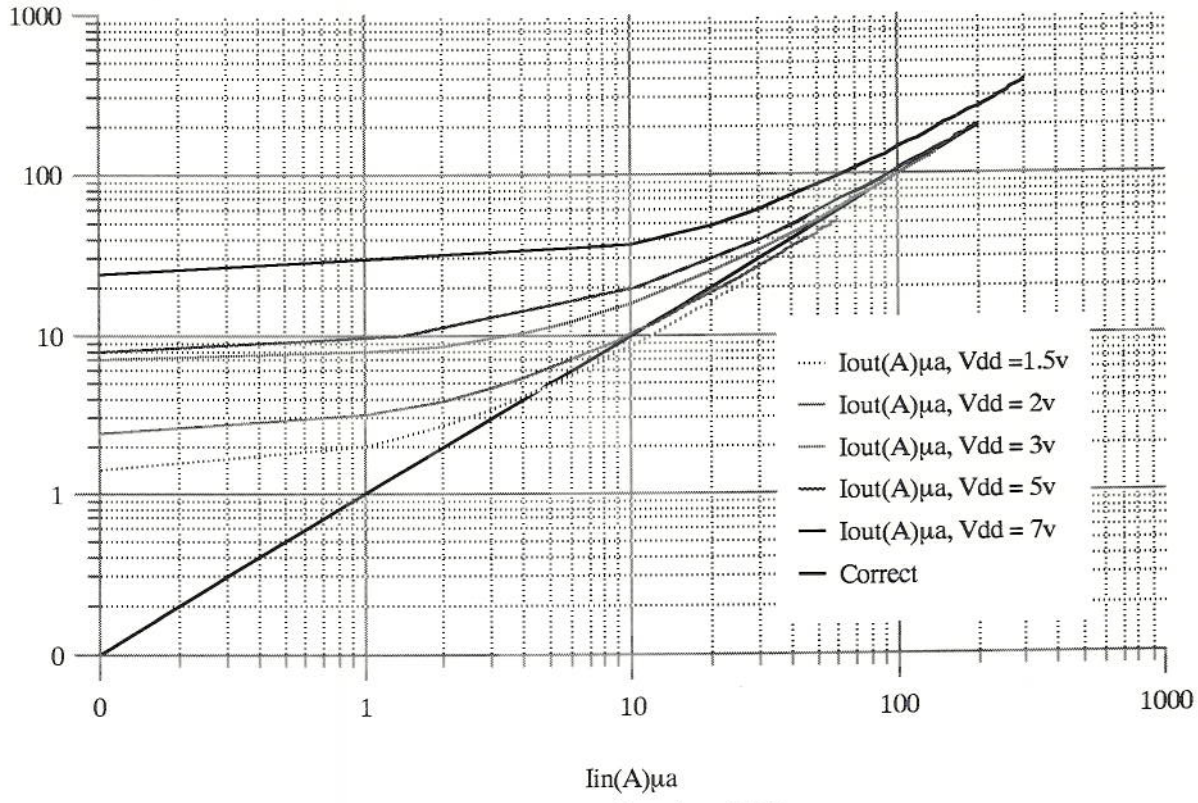

Figure 3. Output linearity of ŁL9

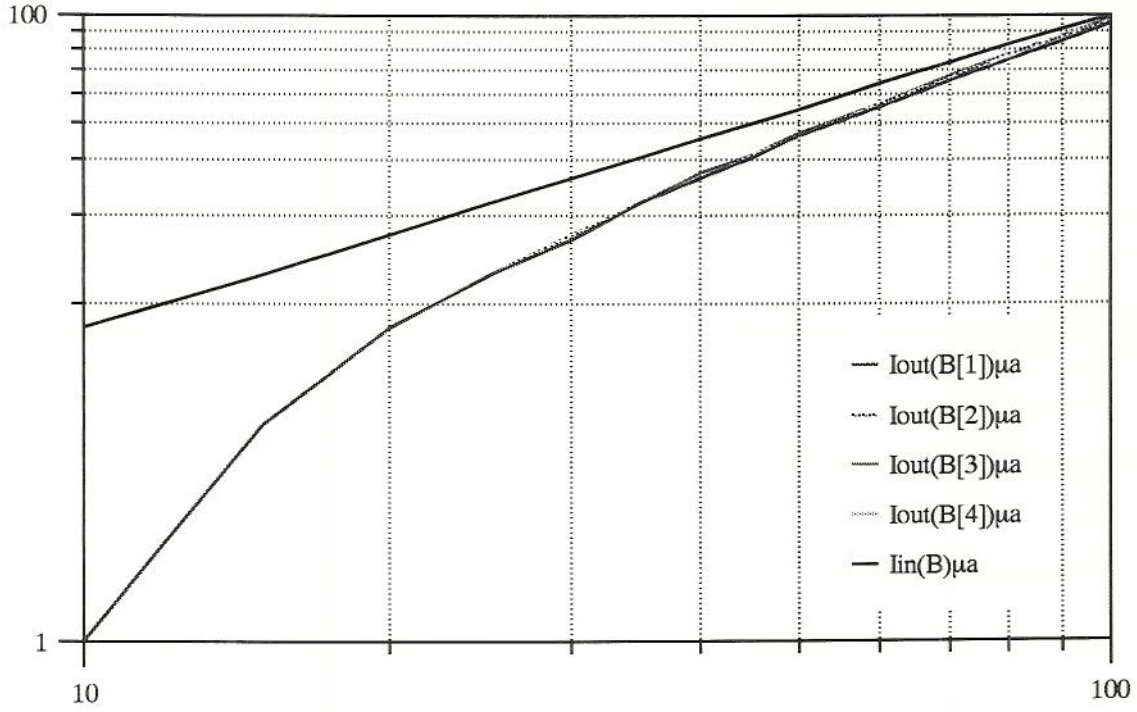

Figure 4. Variation in multiple current mirror outputs of ŁL10

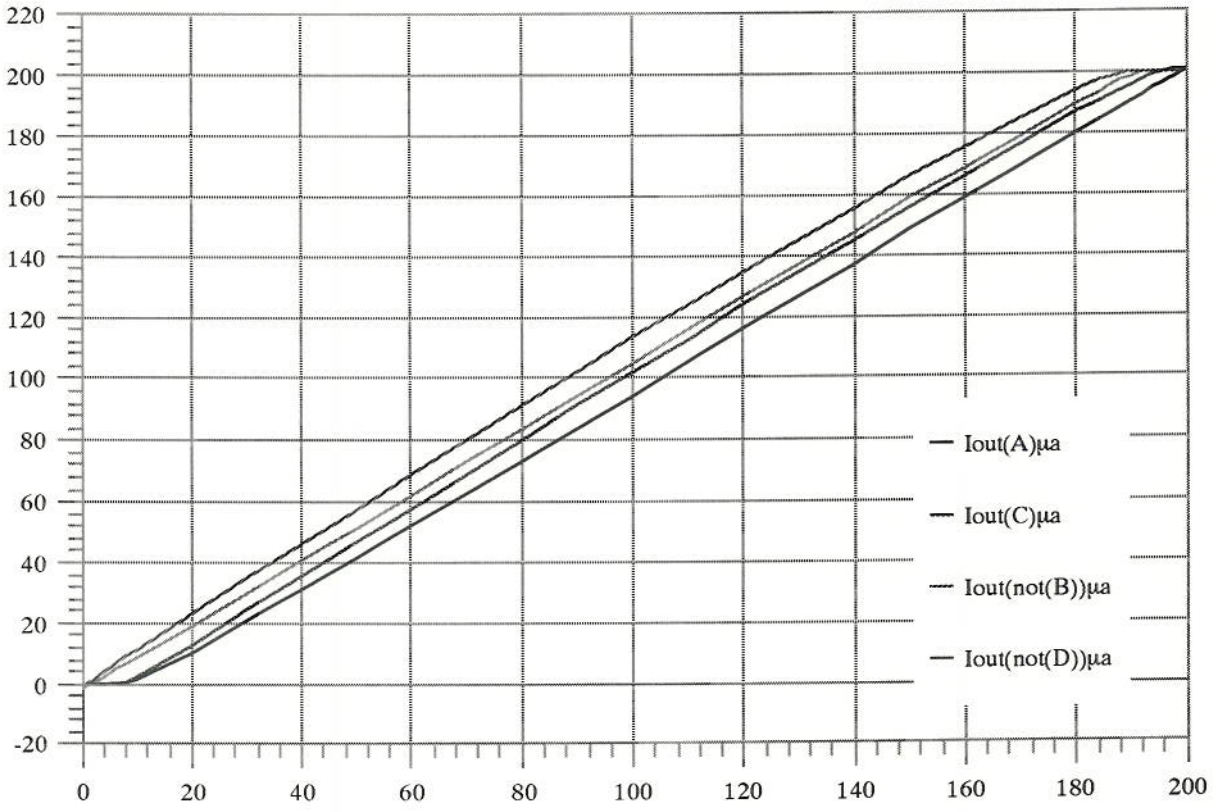

Figure 5. "Tail" non-linearity in normal and inverted outputs of ŁL9

To determine whether variation in the outputs of the current mirrors were affecting the outputs of the ŁLA cells in this range, ŁL10 was designed to provide multiple copies of its output through four current mirrors. For an output inverted with respect to Figure 3, there is no significant deviation between the four current mirrors' output (Figure 4). This suggests that the deviation seen results from a characteristic of the MOSFET used in the design of the ŁLA, rather than the current mirrors.

Further tests used the four-input two-level ŁLA, version ŁLT, to produce normal and inverted values for each of the four inputs. Here the non-linear behavior is emphasized at the different ends of related plots: for the non-inverted values a "tail" occurs between 0 and15μA, but for the inverted values the tail occurs between 190 and 200μA (Figure 5). This suggests that the same mechanism produces the tail, because inverting the value causes it to move to the opposite end of the range.

Next, a series of "notches," or membership functions in fuzzy logic, were constructed. Fuzzy functions were chosen because Łukasiewicz logic is closely related to fuzzy logic [13, 14, 15, 16, 17], and because Yamakawa showed designs for fuzzy inference engines and expert systems that could be embedded in Łukasiewicz logic arrays [1, 18, 19].

We have used the prototype ŁLA to compute a variety fuzzy membership functions, and present the results obtained for two such functions, along with observations on the error measured using the prototype ŁLA. The first function, an inverted "notch," is defined by the expression $(((\neg \alpha \rightarrow \alpha) \rightarrow \neg (\alpha \rightarrow \neg \alpha)) \rightarrow \beta)$. This membership function was programmed into the 31-cell ŁLA as the following 32-element vector:

(F,F,F,F,α,F, T, α, T, α, α, F,F,F,T,F,F, F, F, F, F, F, F, F, F, F, F, F, F, F, T, β).

The uncorrected output of the ŁLA was measured over the range of 0 to 200 μAmperes (μA) by varying α in 50μA increments. The ideal membership function was also calculated after adjusting the evaluation function for Łukasiewicz implication to the operating range of the circuit (Figure 6). As can be seen, the uncorrected output of ŁL9 varies from the ideal by as much as 14%. The error is constant, varying in the slope of the function, which led to the observation that it could be corrected by varying the output resistance of the ŁLA, or trimming it.

Other errors in the output of ŁL9 include a "step" at each end of the function, related to the "tail" observed earlier, and a displacement of the maximum output value above the ideal "true" reference. Referring to the ŁLA cels whose output ideally would fall in the non-linear range of <20μA, there appears to be no direct correlation between this non-linearity and the error in the notch function (Figure 6).

The exact cause of these errors is still under investigation, although one attempt to correct them by introducing a second power supply voltage in ŁL10 did reduce the displacement of the maximum output value above "true" (Figure 7).

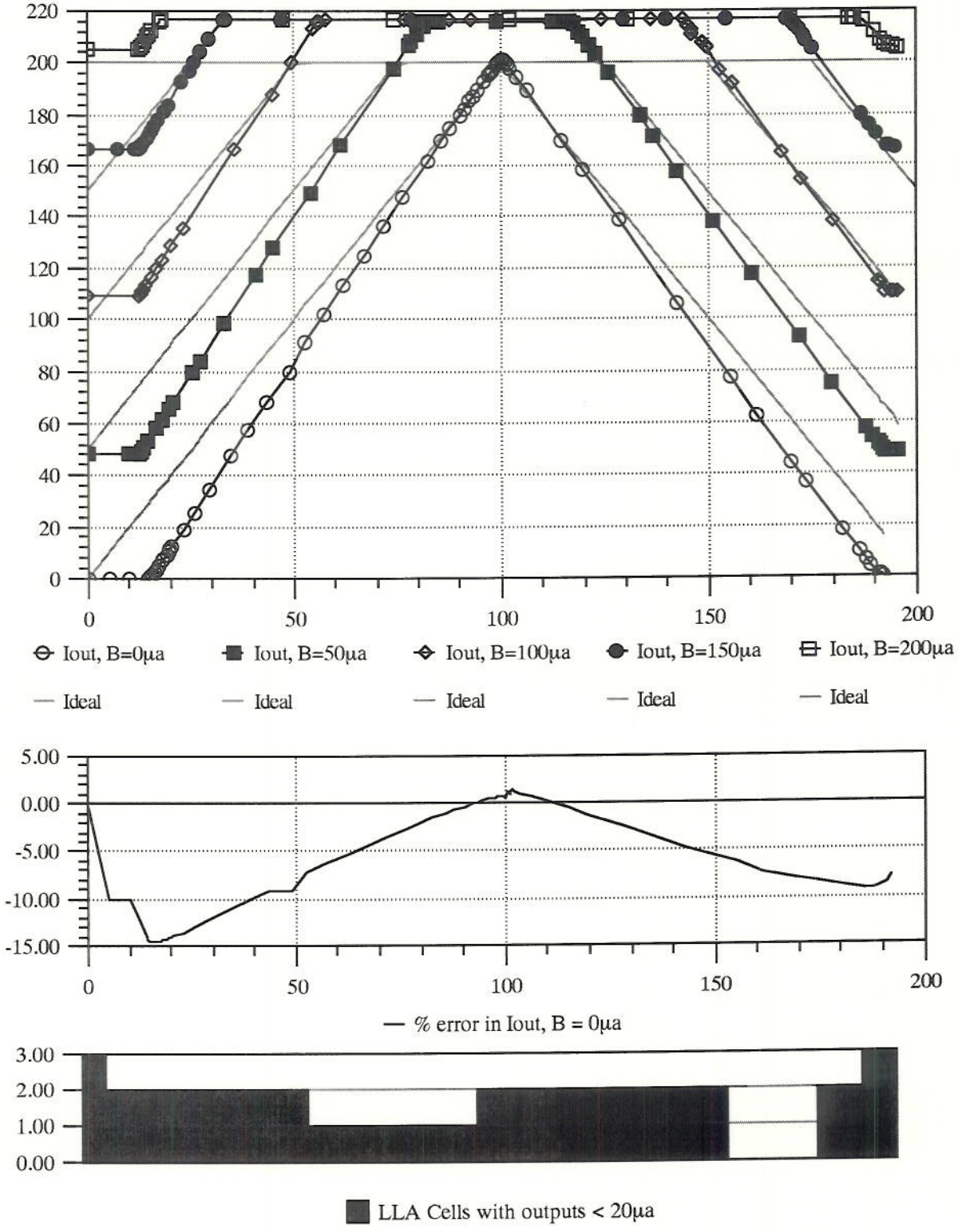

Figure 6. Evaluation of the "notch" using LL9

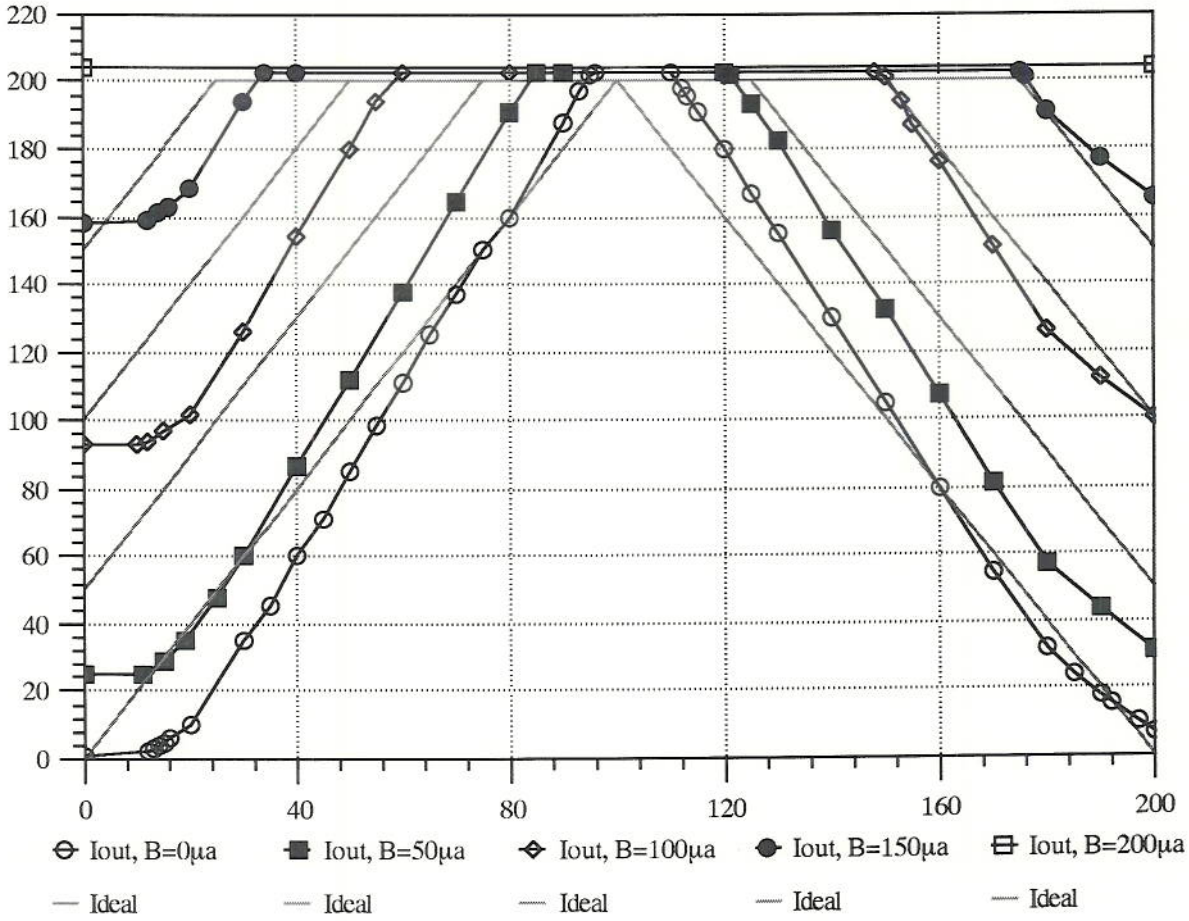

Figure 7. Evaluation of the "notch" using ŁL10

The result of the experiments with the "notch" shows that ŁL9 implemented the notch function linearly except at either end (where the tail appeared) but with a slope that varied from that of the calculated function. As stated before, this scaling is due to an arbitrary choice of a resistor in the measuring circuit. By changing this resistance ("trimming"), we have adjusted the output to produce a much closer fit. Using a simpler "notch" defined by the expression $(\neg \alpha \rightarrow \alpha) \rightarrow \neg (\alpha \rightarrow \neg \alpha)$ we have calculated that an optimally trimmed ŁLA would have a typical error of less than 2%, and a mean error less than 0.5% (Figure 8).

Choosing an analog processing element yields several advantages. Because the ŁLA is a current-mode circuit it has a precision which is not achievable with an equivalently-sized voltage-mode circuit. Although Ł is infinitely valued, in practice only $Ł_2$ through $Ł_{256}$ can be implemented due to device error and the resolution of our measuring devices. The output error measured for the prototype ŁLAs in the range of 0.25% to 2%. This gives an information density ranging from 5.6 to 8.6 bits, or approximately 50 to 400 discreet values per ŁLA. This is a useful precision for approximate reasoning systems.

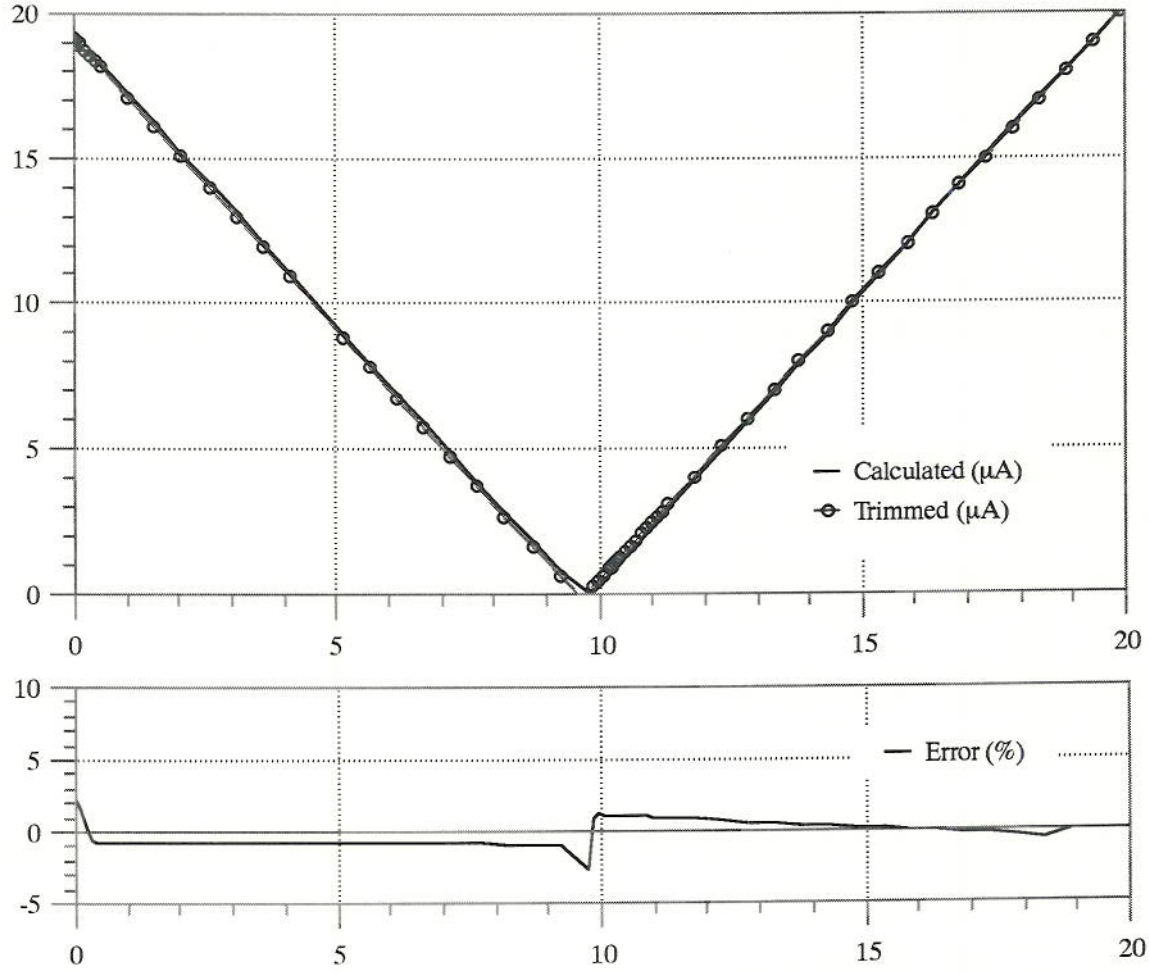

Figure 8. Optimally trimmed "notch" and error

3. APPLICATIONS OF ŁLAS

Łukasiewicz logic arrays were first proposed to evaluate sentences in multiple-valued logic, but because the Łukasiewicz logic Ł describes other forms of approximate reasoning, we propose to use ŁLAs in a variety of applications. The dual logical and algebraic semantics of Ł allow ŁLAs to implement expert systems, neural networks [10, 20], and fuzzy computers [13, 18]. We present schematic examples for these applications in a related paper [21].

4. CONCLUSIONS

We described the architecture of two operational 31-cell 32-input CMOS VLSI ŁLAs, which are implemented with analog rather than digital processing elements. Initial test results showed that the ŁLAs implemented the notch function linearly, but with a slope that varied from that of the ideal function. Trimming the ŁLA inputs and outputs is expected to

result in a typical error of less than 2%, and a mean error less than 0.5%. With further modifications to the circuit, and trimmed outputs, the error should drop to less than 1% for each cell as reported by Yamakawa [1].

ŁLAs present a new challenge in the design of massively parallel processors, as well as the design and programming of analog and hybrid computers. For those problems where precision may be traded for speed, ŁLAs provide an excellent solution.

ACKNOWLEDGMENTS

The referees' suggestions were extremely helpful. Thanks to William Hunt for technical assistance.

REFERENCES

[1] Yamakawa, T., and T. Miki. 1986. The current mode fuzzy logic integrated circuits fabricated by the standard CMOS process. *IEEE Transactions on on Computers* **C-35** (2): pp. 161-167.

[2] Rumelhart, D. E., and J. L. McClelland. 1986. *Parallel Distributed Processing*. Cambridge, Massachussetts: MIT Press.

[3] Grossberg, S. 1988. *Neural Networks and Natural Intelligence*. Cambridge, Massachussetts: MIT Press.

[4] Wilkinson, R. H. 1963. A Method of Generating Functions of Several Variables Using Analog Diode Logic. *IEEE Transactions on Electronic Computers* **EC-12** (2): pp. 112-128.

[5] McNaughton, R. 1951. A theorem about infinite-valued sentential logic. *Journal of Symbolic Logic* **16** pp. 1-13.

[6] Mills, J. W., and A. A. Faustini. *Łucid: An Intensional Programming Language for Łukasiewicz Logic Arrays*. (in preparation)

[7] Mills, J. W., M. G. Beavers, and C. A. Daffinger. 1990. Lukasiewicz Logic Arrays. *Proceedings of 20th International Symposium on Multiple-Valued Logic*. Charlotte, North Carolina: IEEE Computer Society Press. pp. 4-10.

[8] Kung, H. T., and C. E. Leiserson. 1978. Systolic arrays (for VLSI). *Proceedings of Symposium on Sparse Matrices Computations*. Knoxville, Tennessee: SIAM. pp. 245-282.

[9] Alspector, J., R. B. Allen, V. Hu, and S. Satyanarayana. 1987. Stochastic learning networks and their electronic implementation. *Proceedings of Neural Information Processing Systems—Natural and Synthetic*. Denver, Colorado, November 8-12:

[10] Alspector, J., and R. B. Allen. 1987. *A neuromorphic VLSI learning system*. In *Advanced Research in VLSI: Proceedings of the 1987 Stanford Conference*. Edited by P. Losleben. pp. 313-349. Cambridge, Massachusetts: MIT Press.

[11] Mead, C. 1989. *Analog VLSI and Neural Systems*. Edited by L. Conway, C. Seitz and C. Koch. VLSI System Series/Computation and Neural System Series. Reading, Massachusetts: Addison-Wesley.

[12] Leiserson, C. E. 1981. "Area-Efficient VLSI Computation." Ph.D., Dept. of Computer Science, Carnegie-Mellon University,

[13] Giles, R. 1976. Lukasiewicz logic and fuzzy set theory. *Int. J. Man-Machine Studies* **8** pp. 313-327.

[14] Giles, R. 1979. *A formal system for fuzzy reasoning*. In *Fuzzy Sets and Systems 2*. pp. 233-257. North-Holland.

[15] Giles, R. 1982. Semantics for fuzzy reasoning. *Int. J. Man-Machine Studies* **17** pp. 401-415.

[16] Giles, R. 1985. A resolution logic for fuzzy reasoning. *Proceedings of IEEE 17th International Symposium on Multiple-Valued Logic*. IEEE Computer Society Press. pp. 60-67.

[17] Zadeh, L. A. 1975. Fuzzy logic and approximate reasoning. *Synthese* **30** pp. 407-428.

[18] Yamakawa, T. 1988. High-speed fuzzy controller hardware system: The Mega-FIPS machine. *Information Sciences* **45** (2): pp. 113-128.

[19] Yamakawa, T., and H. Kabuo. 1988. A programmable fuzzifier integrated circuit—synthesis, design and fabrication. *Information Sciences* **45** (2): pp. 75-112.

[20] Mattrey, R. F., D. D. Givone, and C. M. Allen. 1973. Applying multiple-valued algebra concepts to neural modeling. *Proceedings of IEEE International Symposium on Multiple-Valued Logic*. Toronto, Canada:

[21] Mills, J., and C. Daffinger. 1990. An Analog VLSI Array Processor for Classical and Connectionist AI. *Proceedings of Application Specific Array Processors*. (accepted). Princeton, New Jersey: