Analog Logic Automata

Kailiang Chen[1], Jonathan Leu[2], and Neil Gershenfeld[1]

[1] Center for Bits and Atoms
Massachusetts Institute of Technology
{name.surname}@cba.mit.edu

[2] Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
cyleu@mit.edu

***Abstract*—Analog Logic circuits work on digital problems using an analog representation of the digital variables, relaxing the state space of the digital system from the vertices of a hypercube to the interior. This lets us gain speed, power, and accuracy over digital implementations. Logic Automata are distributed, scalable and programmable digital computation media with local connections and logic operations. Here we propose Analog Logic Automata (ALA), which relax binary constraints on Logic Automata states and introduce programmability into Analog Logic circuits. The localized interaction and scalability of the ALA provide a new way to do neuromorphic engineering, enabling systematic designs in a digital work flow. Low-power, biomedical, decoding and communication applications are described and a 3X3 ALA chip is prototyped, which works at 50kHz, with a power consumption of 64μW. With the chip configured as a programmable Noise-Locked Loop (NLL), we obtain a Bit Error Rate (BER) of 1E-7 at an SNR of -1.13dB.**

***Index Terms*—Analog Logic, Logic Automata, programmable, low-power, biomedical, image processing, wireless communication, Noise-Locked Loop**

I. INTRODUCTION

Digital computation avoids and corrects errors by sacrificing continuous degrees of freedom. Analog Logic circuits recover this freedom by relaxing the digital states, with each device doing computation in the analog domain, and only quantizing at the output [1]. The analog representations come from either describing digital (binary) random variables with their probability distributions in a digital signal processing problem, or from relaxing binary constraints of an integer programming problem. The preserved information from this analog computation scheme for digital problems gives rise to robust, high-speed, low-power, and cost-effective hardware. Circuit realization examples include decoders [2] and the Noise-Locked Loop (NLL) for direct-sequence spread-spectrum acquisition and tracking, which promise order-of-magnitude improvement over digital realizations [3]. However, prior Analog Logic circuits were custom and special-purposed; no reconfigurable Analog Logic has been reported before.

Logic Automata, a subset of Cellular Automata (CA) [4], quantize space and time with distributed cells connected locally, each performing a basic logic operation [5]. Logic Automata are therefore scalable, universal for digital computation [6], and reflect the nature of many complex physical and biological systems [5], [7], [8]. Previous Logic Automata implementations were always digital, but our Analog Logic Automata (ALA) preserve the information contained between "0" and "1".

ALA are not field programmable analog arrays [9] in that ALA conceptually work in digital space with analog representations. This architecture is also different from FPGA because ALA interconnects are completely local, in contrast to global connections in FPGA. Thus, a systematic ALA design flow for distributed analog systems modeling can be setup, to improve neuromorphic application development, such as signal processing in the retina [10] or neural networks [8]. For example, we can integrate the parallel ALA into the image sensor array chip of the wireless capsule Gastrointestinal (GI) endoscopy [11]. This ALA pre-processing is low-power, real-time, and makes data compression more efficient, reducing the transmission overhead. Applications in telemetric communication and decoding have also been found because of ALA's programmability and power efficiency.

Section II and III discuss architectural and circuit design of ALA. Section IV continues to talk about applications. Section V is test results and section VI concludes the paper.

II. ARCHITECTURAL DESIGN

A. Overview

ALA circuits are realized by replacing digital processing circuits in Logic Automata with Analog Logic circuits, while preserving reconfigurable connectivity and functionality. To solve a signal processing problem, the target problem is formulated into a message-passing algorithm [12], and the ALA unit is programmed accordingly to fulfill the computation with Analog Logic circuits. Another class of problems that can be mapped onto ALA hardware is binary integer programming problems, in which the ALA unit is a relaxation over Logic Automata and all constraints on binary state variables are relaxed, turning integer programming into linear programming.

While we focus on clocked ALA in this work, the principles described here apply equally to un-clocked, i.e.

This work was supported by MIT Center for Bits and Atoms.

asynchronous automata. The asynchronous model keeps the same states and computation, but further localizes time by removing the global clock. Asynchronous Logic Automata have recently been reported in [7]. Future work will include making Analog Asynchronous Logic Automata with lower power consumption and faster speed.

B. Architecture

The ALA model is a continuous state, discrete time model. Each cell in the array stores an analog state Z and interacts with its Von Neumann neighborhood. The cell's X and Y inputs can be any combination of the outputs from its four neighbors, the current state of the cell itself, or external inputs, depending on its connection configuration. Every clock phase, each cell performs an Analog Logic computation according to its function configuration. The cell state is updated and accessible to neighboring cells in the next clock phase.

As specified in the model, all state variables, e.g. X, Y and Z, can be viewed as binary random variables. In current-mode circuits, the probability distributions are represented by

$$\begin{aligned} I_{Z1} &\propto P(Z=1) \equiv P_Z(1), \\ I_{Z0} &\propto P(Z=0) \equiv P_Z(0). \end{aligned} \tag{1}$$

With the above representation, the message-passing algorithms are reduced to a series of summations and multiplications. The summation over several variables is implemented by merging their respective currents, which effectively takes average on probability distributions of those random variables. The multiplication units are programmable soft gates implemented with Gilbert Multipliers [13]. Note that only 2-input soft gates are used in our ALA architecture because they suffice all computations.

For example, a 2-input soft XOR gate performs a statistical version of the XOR operation. The probability distribution of Z is derived from probability distributions of X and Y as in

$$\begin{bmatrix} P_Z(1) \\ P_Z(0) \end{bmatrix} = \begin{bmatrix} P_X(1)\cdot P_Y(0) + P_X(0)\cdot P_Y(1) \\ P_X(0)\cdot P_Y(0) + P_X(1)\cdot P_Y(1) \end{bmatrix}. \tag{2}$$

Similarly, many more soft gates, including 2-input soft AND, NAND, OR, NOR, and 1-input soft Inverter, can be derived from digital gates. But soft gates without digital counterparts also exist. For example, the 2-input EQUAL gate is defined as

$$\begin{bmatrix} P_Z(1) \\ P_Z(0) \end{bmatrix} = \gamma \begin{bmatrix} P_X(1)\cdot P_Y(1) \\ P_X(0)\cdot P_Y(0) \end{bmatrix}. \tag{3}$$

Its complementary, soft UNEQUAL is defined as

$$\begin{bmatrix} P_Z(1) \\ P_Z(0) \end{bmatrix} = \gamma \begin{bmatrix} P_X(1)\cdot P_Y(0) \\ P_X(0)\cdot P_Y(1) \end{bmatrix}. \tag{4}$$

Where γ in (3) and (4) is the normalization factor satisfying $P_Z(1) + P_Z(0) = 1$.

All the above equations indicate that a programmable soft gate can be made by selectively steering and merging the 4 Gilbert Multiplier output currents with 8 switches before the normalization and gate outputs, as shown in Fig. 1. This switching scheme guarantees full function programmability.

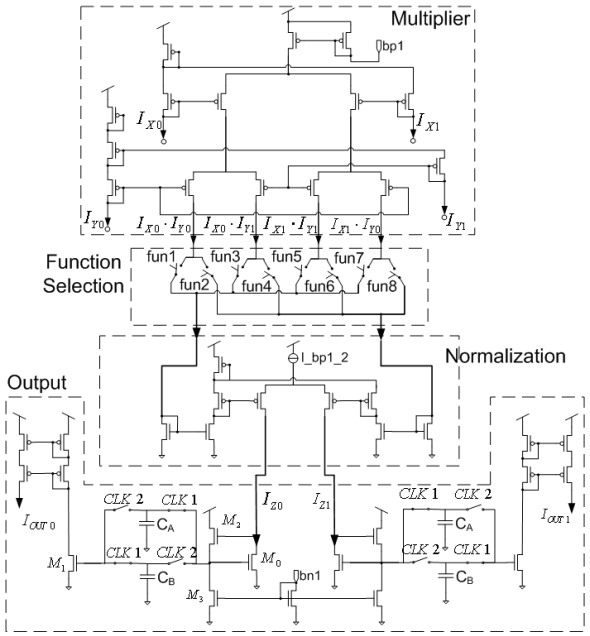

Figure 1. Core schematic of an ALA cell

III. Circuit Design

The core schematic of one cell is shown in Fig. 1. To implement the multiplier, we use sub-threshold MOS transistors in a translinear configuration, as shown in the dashed box labeled "Multiplier" in Fig. 1. This implies that the current density must be small, thus the tail current of the multiplier (I_{bp1}) and the transistor sizes are designed accordingly. Also, diode connected transistors are added to the sources of the input current mirrors, so that all transistors in the translinear circuit are saturated for accurate multiplication.

After passing through the 8 switches that determine the functionality, the two output current values representing the cell state need to be stored, and then driven to neighboring cells at the next clock phase. M_0 and M_1 must be well matched, and the capacitor that stores the gate voltage must be much greater than the gate parasitic capacitance of M_1. In order to charge and discharge the large capacitor within a clock phase, M_2 and M_3 are added, to form a buffer with low output impedance. When the current going into M_0 suddenly increases, the gate voltage of M_2 jumps up. Now M_2 puts more current into the capacitor than M_3 draws, charging it up. When the current going into M_0 suddenly decreases, the gate voltage of M_2 drops, which weakens M_2, so the capacitor discharges. In our test chip, the gate voltage of M_3 is adjustable to ensure the stability of the buffer.

In the first clock phase, capacitor A (C_A) is being written into, and capacitor B (C_B) is connected to the gate of M_1, which goes through cascode current mirrors to send the output currents to the neighbor cells. In the next clock phase, C_B is being written into, and C_A is connected to M_1. This results in the functionality described in the cell architecture.

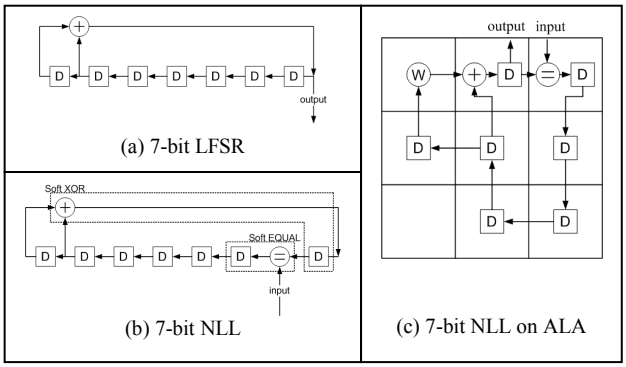

Figure 2. 7-bit NLL implemented on 3X3 ALA

IV. APPLICATIONS

A. Reconfigurable Noise-Locked Loop

The Noise-Locked Loop (NLL) is a generalization of the Phase-Locked Loop (PLL) [1]. Instead of synchronizing to a signal oscillating between two extreme values, an NLL relaxes this constraint and can synchronize to a more complex periodic pattern produced by a given Linear Feedback Shift Register (LFSR). By softening the components in the LFSR and adding a soft EQUAL gate, we obtain the corresponding NLL that synchronizes to a pseudorandom signal.

Within the 3X3 ALA framework, we can match different LFSR transmitters with corresponding NLL receivers up to 7-bit long by changing the array configuration. Fig. 2(a) and (b) show the 7-bit LFSR transmitter and NLL receiver pair. The dashed boxes in Fig. 2(b) indicate ALA cells performing soft XOR/EQUAL functions with a unit delay, which is denoted by "D". The actual implementation on the ALA is shown in Fig. 2(c), where the top left cell is configured as a WIRE gate, denoted by "W". The WIRE function bypasses the input directly to the output without any delay. It is introduced for more routing flexibility in the rectangular-connection-only array. In section V, the test result of this 7-bit NLL is given.

B. Forward-backward Algorithm for Decoding

Forward-backward Algorithm is used for bitwise Maximum APosteriori (MAP) Error Correcting Code (ECC) decoding. It is solved by passing and merging forward and backward messages on a trellis [14], which is easily mapped onto the ALA architecture. In principle, all ECC codes can be decoded by ALA circuits implementing trellis decoding. As an example, a (7, 4) Hamming Code decoder on 61X63 ALA is designed and simulated. Soft AND gates are used to calculate multiplying terms and 7 soft UNEQUAL gates are used for bitwise decoding decision. Simulation shows that the decoding process takes 56 clock cycles and the results agree with standard decoding algorithm.

Because both the ALA decoder and the NLL for pseudorandom signal synchronization, which is described in part A, can be more power efficient than their counterparts in a digital receiver, they can be applied in low-power portable biomedical devices or telemetric links for implantable sensors.

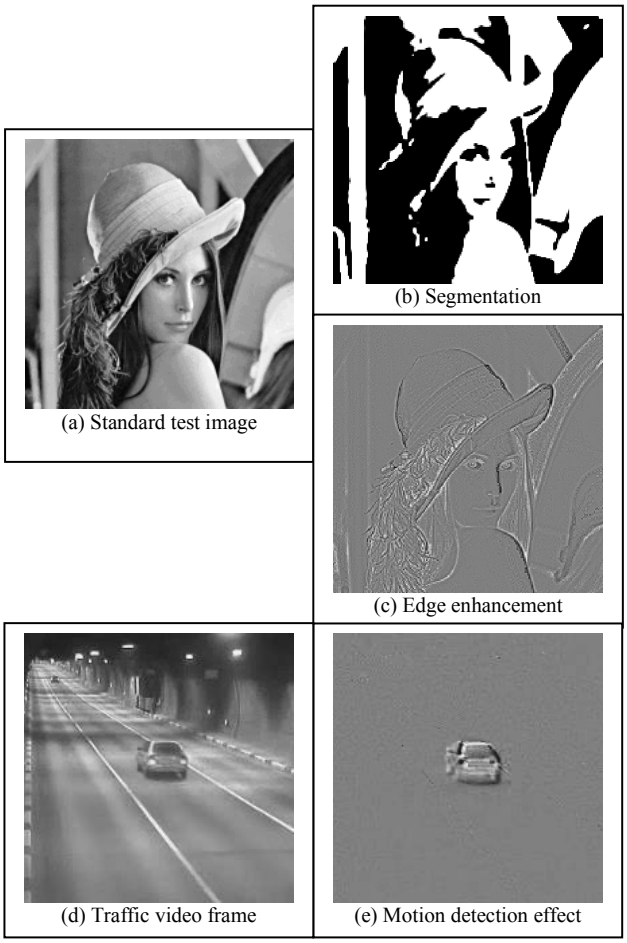

Figure 3. Image processing applications

C. Image Processing

With each cell representing a grey-level pixel, the ALA unit becomes a programmable image processor. Three simple examples are simulated, which might be used in wireless capsule GI endoscopy or other low-power biomedical image processing applications, as mentioned in the introduction.

Segmentation is achieved by the soft EQUAL operation, which depicts the similarity between a pixel and the average of its 4 neighbors. Similarly, edge enhancement is achieved by the soft UNEQUAL operation, depicting the difference between a pixel and its surroundings. Here we show the segmentation (Fig. 3(b)) and edge enhancement (Fig. 3(c)) of the standard test image (Fig. 3(a)) in full effect after 10 and 1 iterations, respectively. Note that with parallel computation, image edge enhancement needs only 1 clock cycle to complete.

Additionally, motion detection can be implemented by revealing the difference between two consecutive video frames. This is realized by performing the soft UNEQUAL function between two images, pixel by pixel. Fig. 3(e) shows the motion detection result of a traffic video (Fig. 3(d)).

V. Test Results

We fabricated a 3X3 ALA chip in the AMI 0.5µm CMOS process, with an area of 1.5X1.5mm^2 and 4V voltage supply. The array can work at 50kHz and the power consumption is 64µW, including both digital and analog circuits. Process scaling and less configuration registers (which take half of the chip area at present) can increase the cell density. The die photo is shown in Fig. 5.

The 3X3 ALA chip was tested for the 3-bit to 7-bit NLL applications, and worked correctly. We convert the current mode output to a voltage with an off-chip resistor, and then add a ±5V buffer to drive the oscilloscope probe. In Fig. 4, we can see that the NLL locks onto the noisy input signal after 43 clock phases. In this measurement, the input signal current swing is fixed at 47.4nA and by changing the noise power we derive the Bit-Error-Rate vs. SNR plot, as shown in Fig. 6.

VI. Conclusion

Here we propose Analog Logic Automata as a low-power, parallel, and programmable hardware, which is a relaxation over Logic Automata using Analog Logic gates, while preserving re-configurability. The new circuits provide a wide range of applications spanning biomedical image processing, digital communication, and decoding. We also demonstrate a 3X3 ALA chip for programmable NLL application.

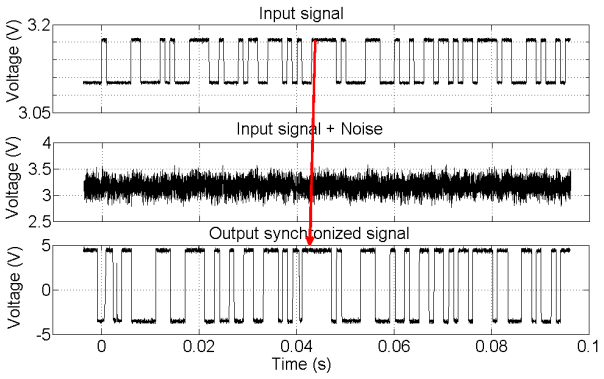

Figure 4. 7-bit NLL locking and tracking dynamics

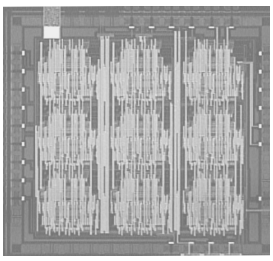

Figure 5. 3X3 ALA chip die photo

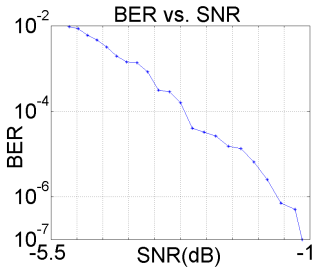

Figure 6. BER vs. SNR

The ALA circuits presented here promise great potential in many research fields. Firstly, this architecture offers a new approach to a better neuromorphic design. Secondly, more complex image processing algorithms can be developed and used in different biomedical applications, given this highly programmable hardware. Thirdly, the ALA architecture suggests a new way to realize Software Defined Radio (SDR). Instead of first digitizing the RF signal and then processing the signal digitally, ALA circuits can directly obtain the baseband signal through analog computation, only digitizing at the output. Finally, ALA can implement decoders for other types of ECC codes decodable in similar message-passing algorithms, such as turbo codes and low-density parity-check codes. Overall, the ALA architecture is promising as a versatile platform for fast algorithm/application development.

Acknowledgment

We thank Prof. Rahul Sarpeshkar and Soumyajit Mandal for their support in chip fabrication and testing; Luis Lafuente and David Dalrymple for discussion on theoretical analysis; Ara Knaian, David Kopp, and Forrest Green for help in test-bed setup; and Manu Prakash for general advice.

References

[1] B. Vigoda, Analog Logic: Continuous-Time Analog Circuits for Statistical Signal Processing, Ph.D. thesis, Massachusetts Institute of Technology, June 2003.

[2] H.-A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarkoy, "Probability propagation and decoding in analog VLSI," Information Theory, IEEE Transactions on , vol.47, no.2, pp.837-843, Feb 2001.

[3] B. Vigoda, J. Dauwels, M. Frey, N. Gershenfeld, T. Koch, H.-A. Loeliger, and P. Merkli, "Synchronization of Pseudorandom Signals by Forward-Only Message Passing With Application to Electronic Circuits," Information Theory, IEEE Transactions on , vol.52, no.8, pp.3843-3852, Aug 2006.

[4] N. Gershenfeld, The Nature of Mathematical Modeling, Cambridge University Press, 1999.

[5] N. Gershenfeld, "Programming Bits and Atoms," to be submitted.

[6] R.E. Banks, Information Processing and Transmission in Cellular Automata, Ph.D. thesis, Massachusetts Institute of Technology, 1971.

[7] D.A. Dalrymple, N. Gershenfeld, and K. Chen, "Asynchronous logic automata," Proceedings of AUTOMATA 2008 (14th International Workshop on Cellular Automata), pp.313-322, Jun 2008.

[8] L.O. Chua, "CA belongs to CNN," invited talk at AUTOMATA 2008 (14th International Workshop on Cellular Automata), Jun 2008.

[9] T.S. Hall, C.M. Twigg, J.D. Gray, P. Hasler, and D.V. Anderson, "Large-scale field-programmable analog arrays for analog signal processing," Circuits and Systems I: Regular Papers, IEEE Transactions on, vol.52, no.11, pp.2298-2307, Nov 2005.

[10] C. Mead, "Neuromorphic electronic systems," Proceedings of the IEEE , vol.78, no.10, pp.1629-1636, Oct 1990

[11] Meng-Chun Lin, Lan-Rong Dung, and Ping-Kuo Weng, "An ultra-low-power image compressor for capsule endoscope", BioMedical Engineering OnLine, vol.5, no.1, pp.14, 2006

[12] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," Information Theory, IEEE Transactions on , vol.47, no.2, pp.498-519, Feb 2001.

[13] F. Lustenberger, On the Design of Analog VLSI Iterative Decoders, Doctoral Dissertation, Swiss Federal Institute of Technology, Nov 2000.

[14] D. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.