A novel hardware implementation of IDS method

Mahdi Tarkhan[1a)], **Saeed Bagheri Shouraki**[1], **and Seyed Hossein Khasteh**[2]

[1] *Electrical Engineering Department, Sharif university of Technology*
Azadi Ave., Tehran, Iran

[2] *ComputerEengineering Department, Sharif university of Technology*
Azadi Ave., Tehran, Iran

a) *takhan@ee.sharif.edu*

Abstract: ALM is an adaptive recursive algorithm which tries to express a multi-input multi-output system as a fuzzy combination of some single-input single-output systems. It uses a fuzzy curve fitting technique for behavior extraction or finding the input-output transformation of each of the single-input single-output systems, which called ink drop spread (IDS). In this paper we present a new implementation of a hardware unit implementing the ink drop spread (IDS) method.

Keywords: IDS method, ALM, hardware design, analog memory
Classification: Integrated circuits

References

[1] M. Murakami, *Practicality of Modeling Systems Using the IDS Method: Performance Investigation and Hardware Implementation*, Ph.D dissertation, The University of Electro-Communications, March 2008.

[2] T. Yamakawa and T. Miki, "The Current Mode Fuzzy logic Integrated Circuits Fabricated by the Standard CMOS Process," *IEEE Trans. Comput.*, vol. c-35, no. 2, pp. 161–167, Feb. 1986.

[3] T. Temel, "High-performance current-mode multi-input loser-take-all minimum circuit," *IEEE Electron. Lett.*, vol. 44, no. 12, pp. 718–719, June 2008.

[4] I. Baturone, J. L. Huertas, A. Barriga, and S. Sanchez-Solano, "Current-Mode Multiple-Input Max CIrcuit," *Electron. Lett.*, vol. 30, no. 9, pp. 678–679, April 1994.

[5] C. Sawigun and W. A. Serdijn, "Low-voltage, low-power, low switching error, class-AB switched current memory cell," *IEEE Electron. Lett.*, vol. 44, no. 12, pp. 706–708, June 2008.

[6] C. Sawigun and W. A. Serdijn, "0.75 V micro-power SI memory cell with feedthrough error reduction," *IEEE Electron. Lett.*, vol. 44, no. 9, pp. 561–561, April 2008.

[7] L. Ravezzi, D. Stoppa, and G. F. Della Betta, "Simple high-speed CMOS current comparator," *IEEE Electron. Lett.*, vol. 33, no. 22, pp. 1829–1830, Oct. 1997.

[8] S. B. Shouraki and N. Honda, "Hardware Simulation of Brain Learning Process," *15th Symp. of FUZZY Syst.*, oosaka, Japan, pp. 523–526, June 1999.

© IEICE 2009
DOI: 10.1587/elex.6.1626
Received July 27, 2009
Accepted October 03, 2009
Published December 10, 2009

1 Introduction

IDS is a method that expresses the behavior of a single-input single-output system by two functions, a narrow path function and a spread function. ALM uses these functions to explain the behavior of a multi-input single-output system. For computing the narrow path and spread functions, IDS uses planes that each describing as SISO system. In this method each new data point in every plane is diffused like ink, then narrow path and spread functions are evaluated. SHOURAKI proposed methods for the hardware implementation of IDS [8]; MURAKAMI designed a digital hardware for this purpose [1]. In this paper we propose a new hardware solution and compare it with MURAKAMI's design with respect to the speed constraint.

2 Proposed method

For calculating the narrow path and spread functions, the IDS method divide the plane into a 256 × 256 grid; we call each intersection of grid lines a cell.

The pattern of data diffusion is illustrated in Fig. 1 (a). This process is called "data spread". As individual data spreads overlap, the overlapping portions become darker, finally resulting in an image on the surface of the plane.

In this hardware, information is processed as current signals.

In order to compute the ink amount in each point of the ink spread pattern, we only need to calculate this amount for one point; the amount of ink is same for a square, and it reduces when the squares become larger. Fig. 1 (b). demonstrates this.

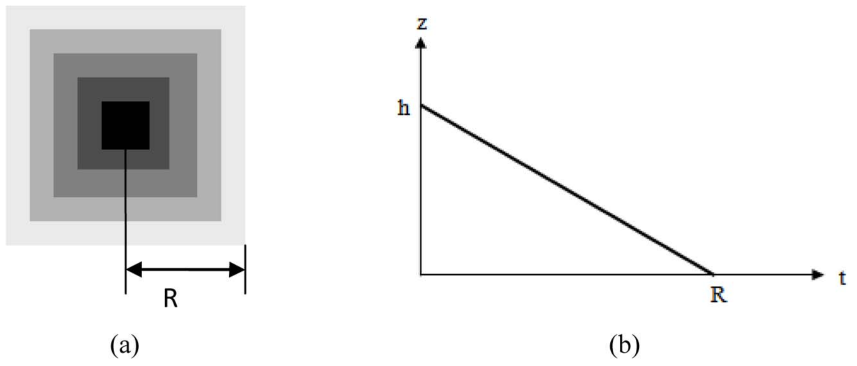

Fig. 1. (a) Ink spread pattern, and (b) Relation between ink amount and distance

This relationship can be formulated as presented below:

$$z = -\frac{h}{R}t + h$$

In this formula z is the amount of ink in a square with length t, and R is the maximum possible length of the squares. If we assume that the length of maximum effective t (which is equal to R) is equal to maximum possible ink

© IEICE 2009
DOI: 10.1587/elex.6.1626
Received July 27, 2009
Accepted October 03, 2009
Published December 10, 2009

for each data (h), then the above formula becomes:

$$z = R - t$$

The hardware implementation of this formula is easier than previous formula because we don't need divider and multiplier modules. This formula can be implemented by a simple bounded difference circuit.

For a new input data located in (x, y) location, the ink pattern is created and it influences those cells that are located in this ink effect region.

In order to measure the amount of ink in a specific cell that is located in (xi, yi) two special squares must be drawn; these two squares sides lengths are $|x-xi|$ and $|y-yi|$, in the previous formula t is equal to $Max(|x-xi|, |y-yi|)$ and the amount of ink from this input data is $z = R - max(|x - xi|, |y - yi|) \quad if \ R \geq max(|x - xi|, |y - yi|)$.

This means this cell is located in an affected region of new data.

For implementing the formula above several circuits are used, including the bounded difference [2], the absolute difference [2] and the max circuit [4].

In order to save some amount of ink in cells we assign an analog memory for each cell. These memories are designed to save current by means of the switched current technique.

We use switched current memory cells like those introduced in [5, 6]. In the previous section, z is calculated and must be added to the previous amount of ink in the cell. For this reason we need 2 memory cells as illustrated in Fig. 2. (a).

clk 1 and clk 2 pattern are depicted in Fig. 2. (b).

When clk1 is high and clk2 is low $\{i2 = i1 = z + current\ amount\ of\ ink\}$, and ink is constant because clk2 is low.

When clk2 is high and clk1 is low $\{new\ amount\ of\ ink = i2\}$, and therefore z is added to the previous amount of ink and the amount of ink is updated.

A narrow path is a path in the plane that connect cells with following property: the sum of the amount of inks in the cells that are located above that cell in that column, is approximately equal to the sum of the amount of ink in the cells that are located below that cell in that column.

Lemma1: if we have some memory cells in a column and want to find the narrow path cell, we use the following formulas, with these notations:
$\psi(x) =$ A point in the Narrow path.
$b =$ the narrow path cell is located at (x, b).
$d(x, y) =$ The amount of ink in the cell that is located in (x, y).
$m =$ The half of the sum of the amount of inks in all of the cells in same column.

$$\psi(x) = \left\{ b \middle| \sum_{y=1}^{b} d(x, y) \approx \sum_{y=b}^{y_{max}} d(x, y), \ b \in Y \right.$$

$$\sum_{y=1}^{b} d(x, y) \approx \sum_{y=b}^{y_{max}} d(x, y) \Rightarrow$$

© IEICE 2009
DOI: 10.1587/elex.6.1626
Received July 27, 2009
Accepted October 03, 2009
Published December 10, 2009

$$\sum_{y=1}^{b-1} d(x,y) + d(x,b) \approx \sum_{y=b+1}^{y_{max}} d(x,y) + d(x,b) \Rightarrow$$

$$\sum_{y=1}^{b-1} d(x,y) \approx \sum_{y=b+1}^{y_{max}} d(x,y)$$

If $m = \dfrac{\sum_{y=1}^{ymax} d(x,y)}{2}$ then

$$2m = \sum_{y=1}^{b} d(x,y) = \sum_{y=1}^{b-1} d(x,y) + d(x,b) + \sum_{y=b+1}^{ymax} d(x,y)$$

$$2m = 2\sum_{y=1}^{b-1} d(x,y) + d(x,b) \Rightarrow \sum_{y=1}^{b-1} d(x,y) = m - \frac{d(x,b)}{2}$$

$$\Rightarrow \sum_{y=1}^{b-1} d(x,y) \geq m$$

$$\sum_{y=1}^{b} d(x,y) = m + \frac{d(x,b)}{2} \Rightarrow \sum_{y=1}^{b} d(x,y) \leq m$$

We can compute m and start from the first cell comparing the amount of ink in this cell with m; if $ink > m$, then this cell is the desired point in the narrow path otherwise we add the amount of ink in the second cell to the first cell and compare this amount with m; if this amount become larger than m, the second cell is the desired point, otherwise we add the third cell ink to the second and the first cell ink and compare with m and so on.

To implement the formula above we design the circuit shown in fig. 2. (c) In this circuit CCs are current comparator circuits [7] and $d(x,i)$ is the amount of ink in the cell that is located in the (x,i) location.

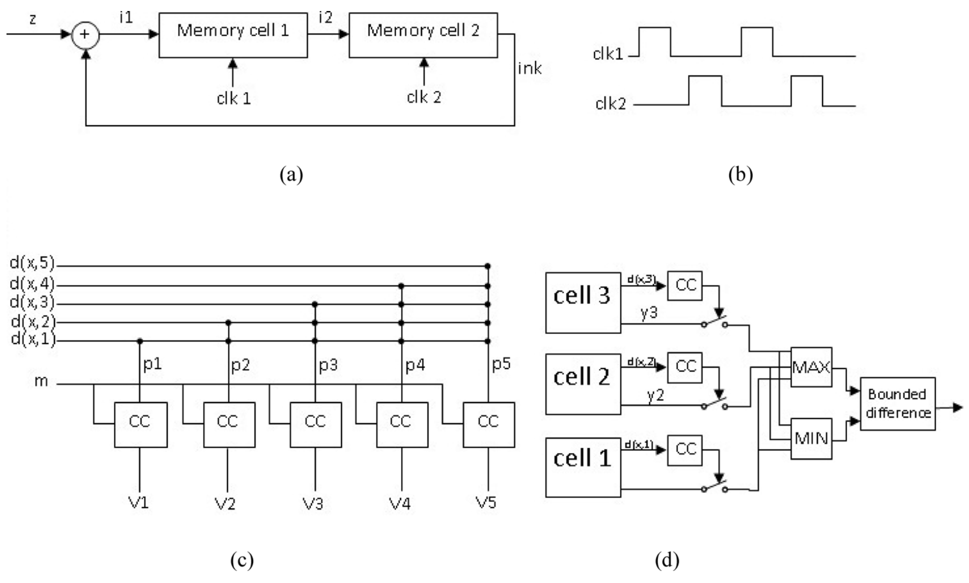

Fig. 2. (a) Analog memory cells, and (b) Clock patterns, and (c) Narrow path computing circuit, and (d) Spread computing circuit

© IEICE 2009
DOI: 10.1587/elex.6.1626
Received July 27, 2009
Accepted October 03, 2009
Published December 10, 2009

The output voltages are 0 or 1 and therefore we have a binary code in output. If $Pi < m$ then Vi is 0 and vice versa. Therefore we have a code such as 00111. The place of the first 1 in the code determines the desired cell in the narrow path. For example in the code above, the third cell is the desired cell in narrow path.

For computing the spread function we must find the highest and lowest cells in each column that contain ink. The place of each cell is determined with two currents, one for xi and another for yi. Spread computing has several steps. In the first step, the amount of ink in each cell in a column is compared with 0 by means of current comparator circuit [5] to determine whether this cell has ink or not. If the cell has ink in itself, the output voltage of current comparator is a logic 1, otherwise 0. These voltages are applied to switches to connect yi to MAX [4] and MIN [3] circuits; this implementation is illustrated in Fig. 2. (d).

3 Simulation results

The proposed circuit is simulated with a 0.25 um CMOS technology in the HSpice Simulation environment. The input signals are depicted in Fig. 3. (a) and the output signals in Fig. 3. (b). As it can be seen the speed of this circuit is 0.5 us in comparison to 5 us of MURAKAMI's design [1]. The amount of time to complete a computation is almost 0.5 us in both empty and filled x-y plane in contrast with MURAKAMI's design which the result is generated after 29.5 us in case of filled plane.

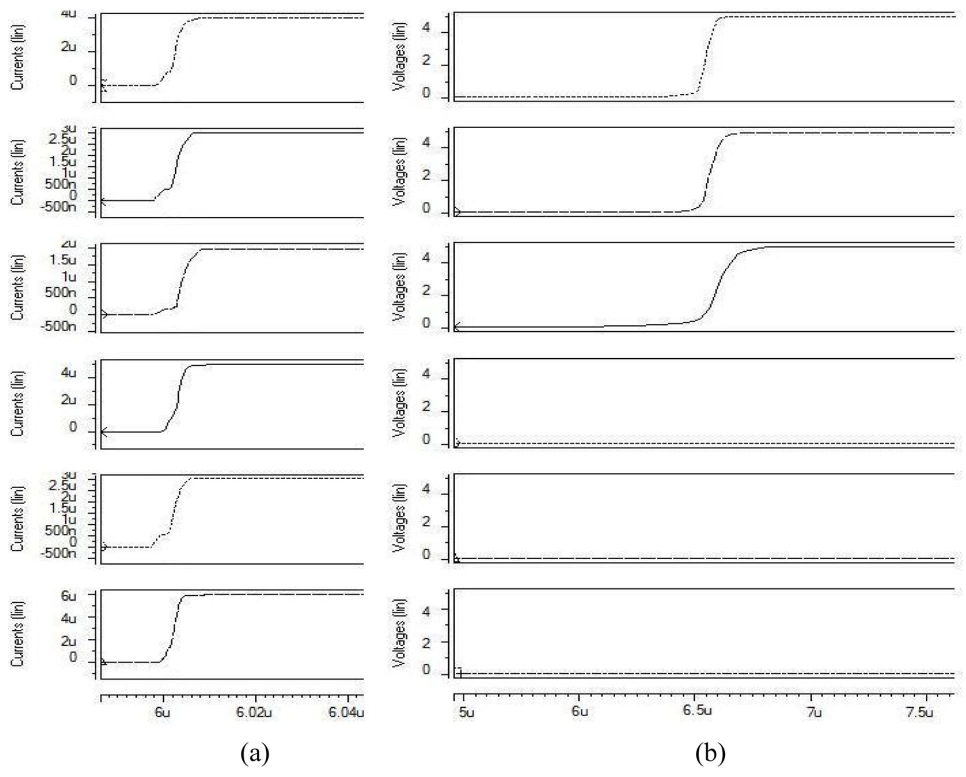

Fig. 3. (a) Input signals, and (b) output signals

DOI: 10.1587/elex.6.1626
Received July 27, 2009
Accepted October 03, 2009
Published December 10, 2009