



VLSI Research Group

University of Windsor

*Digital Arithmetic Using Analog
Cellular Neural Networks*

Saeid Sadeghi-Emamchaie, G.A. Jullien, V.S. Dimitrov
and W.C. Miller

Digital Arithmetic Using Analog Cellular Neural Networks

Saeid Sadeghi-Emamchaie, G.A. Jullien, V.S. Dimitrov and W.C. Miller
VLSI Research Group, University of Windsor
Windsor, Ontario, Canada N9B 3P4

Abstract

We discuss the realization of digital arithmetic using analog arrays in the form of Cellular Neural Networks (CNNs). These networks replace the fast switching nodes of logic gates with slewing nodes using current sources driving into capacitors; this provides both low current spikes and low voltage slewing rates, reducing system noise and cross-talk in low-voltage mixed-signal applications. In this paper we generalize the design methodology using a Symbolic Substitution (SS) technique, and we use a recently developed Double-Base Number System (DBNS) to illustrate our design technique. This choice is predicated on the fact that the DBNS representation is naturally 2-dimensional and excites more degrees of freedom in the design space. Spatial configurations of the recognition/replacement patterns used in SS are defined based on the properties of the DBNS arithmetic operation. The SS recognition phases are implemented by dynamic evaluation of simple conditions defined based on an analysis of the cell dynamic routes. The replacement phases are automatically executed through switching current sources which force the transition of cell state voltages between logic levels. In effect, we build self-timed logic arrays with all nodes in the system under controlled slew. Simulation results from schematic level designs are provided to demonstrate the effectiveness of the technique.

Keywords: *Cellular Neural Networks; Symbolic Substitution; Double-Base Number System; Analog VLSI Arrays; Low-noise Arithmetic Implementation.*

1 Introduction

This paper presents general methods for implementing digital arithmetic using a class of analog Cellular Neural Networks (CNNs). CNNs are basically large scale nonlinear circuits, composed of locally connected cells on a 2D rectangular grid; a structure well suited to VLSI analog implementations [8]. All cells in the CNN operate in parallel and in continuous time. Although the cells are only locally connected, the network is able to perform global operations on the 2D inputs. This is possible by the propagation of information through the network from cell to cell [9]. We have two strong reasons for using CNNs as digital processing arrays in deep sub-micron VLSI technologies.

1. Since the nodes in the array change voltage in a controlled slew, there is the potential for both reduced cross-talk (low $\frac{\partial v}{\partial t}$) and reduced system noise (low $\frac{\partial i}{\partial t}$) compared to equivalent arrays of logic gates.
2. Using our new technique, CNNs provide the equivalent of self-timed logic arrays in terms of clock-free processing, but with the attendant advantages of low noise analog processing.

In our method a given arithmetic operation is reformulated as processing a binary 2-D signal (image) using a Symbolic Substitution (SS) technique [3], where *recognition patterns* are successively replaced by *replacement patterns*. The spatial configuration of these patterns are determined based on the properties of the given arithmetic operation. The arithmetic task is finished when there are no recognition patterns left for substitution.

In this paper we present an analog CNN structure suited for Symbolic Substitution, and we introduce methodologies to design these CNN arrays for low-noise arithmetic operations in a recently developed Double-Base number system (DBNS) [6]-[7]. In Section 2 we briefly introduce the DBNS and the addition operation in this number system. Then we discuss the pattern substitutions required for reducing the representation to an addition-ready form. In Section 3 we present a restricted definition of the SS method that is required for performing digital arithmetic. In Section 4, a class of CNNs is introduced for realizing the Symbolic Substitution paradigm. Design methods are presented in Section 5 and, based on these methods, the design of a CNN-based DBNS adder is presented in Section 6. We conclude the work in Section 7.

2 The Double-Base Number System (DBNS)

The *double-base number system* (DBNS) has the following representation [6][7]:

$$x = \sum_{i,j} d_{i,j} 2^j 3^i \quad d_{i,j} \in (0, 1). \quad (1)$$

Clearly the binary number system is a special case of the above representation and it is clear that the representation is redundant (unlike Signed-Digit binary, there is no guaranteed unique canonic form). There is a natural 2-dimensional nature to the representation. Consider the ‘‘DBNS Map’’ shown in Figure 1 for numbers 79 and 30; the nonzero digits $d_{i,j} = 1$ are shown with black pixels (cells) which we also refer to as active cells.

The DBNS is very sparse (i.e. an integer can be represented with a very small number of non-zero digits) and this sparsity promises efficient arithmetic for applications such as modular exponentiation in cryptography [14]. A particular application targeted for this CNN implementation is a processor for a mixed-signal hearing instrument, where low noise rather than high processor speed is a desirable feature.

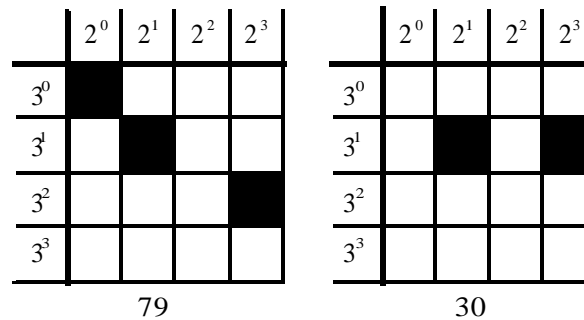


Figure 1. Number representation in the DBNS map.

We require the following 2 definitions:

Definition 1: Representation of numbers using the minimal number of active cells is defined as a *canonic double-base number representation* (CDBNR). This canonic form is not unique.

Definition 2: A DBNR that has no consecutive active cells in a row in its map is defined as an addition ready DBNR (ARDBNR). It can be shown that a CDBNR is always an ARDBNR [7].

2.1 DBNS Addition

Let $I_x(i, j)$ and $I_y(i, j)$ be the DBNS maps of the integers x and y , represented in the ARDBNR. The image $I_z(i, j)$ of the DBNS map of the number $z = x + y$ can be obtained by simply overlaying the $I_x(i, j)$ and $I_y(i, j)$ and applying the following steps:

1. $I_z(i, j) = I_x(i, j) \text{ XOR } I_y(i, j)$
2. $I_z(i, j + 1) = I_x(i, j) \text{ AND } I_y(i, j)$

Step 1 represents the simple identity $2 \times 2^j 3^i = 2^{j+1} 3^i$. If both $I_x(i, j)$ and $I_y(i, j)$ have an active cell at an arbitrary position (i, j) , then in the DBNS map of the result, $I_z(i, j)$, these overlapped active cells are replaced by an active cell at position $(i, j + 1)$. Note that in using the ARDBNR of x and y , if $I_x(i, j) = I_y(i, j) = 1$ then $I_x(i, j + 1) = I_y(i, j + 1) = 0$ therefore the operation contained in Step 2 is feasible.

Example 1: The DBNS-maps shown in Figure 1, are ARDBNR (i.e. no consecutive active cells in a row). The result of the addition of these two numbers using the above steps is shown in Figure 2. The resulting image $I_z(i, j)$, however, is not necessarily ARDBNR since it might have consecutive active cells in a row.

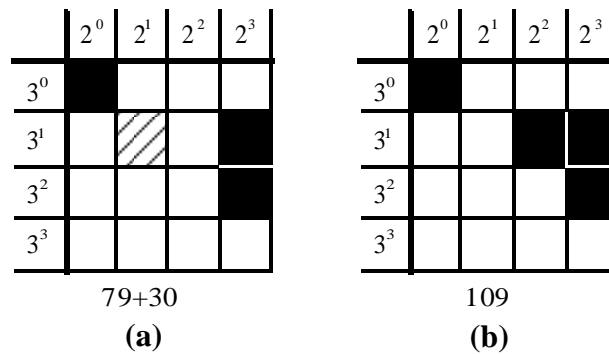


Figure 2. DBNS addition of 79 and 30 using the maps given in Figure 1. (a) result of step 1 and (b) final result.

To convert the resulting image to an ARDBNR the following simple identities can be used to replace adjacent cells.

Reduction rule 1:

Using the simple identity:

$$2^j 3^i + 2^{j+1} 3^i = 2^j 3^{i+1} \tag{2}$$

the two adjacent active cells at position (i, j) and $(i, j + 1)$ can be replaced by a cell at $(i + 1, j)$ (see Figure 3).

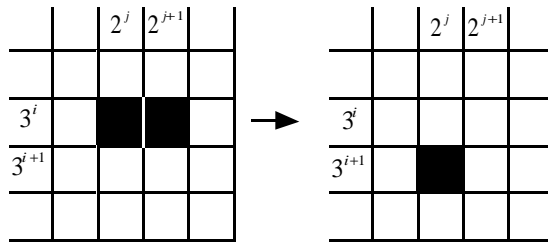


Figure 3. Reduction rule 1

Reduction rule 2:

In cases where the position $(i + 1, j)$ is occupied, the identity given in eqn. (3) provides reduction rule 2 as shown in Figure 4.

$$2^j 3^i + 2^{j+1} 3^i + 2^j 3^{i+1} = 2^{j+1} 3^{i+1} . \tag{3}$$

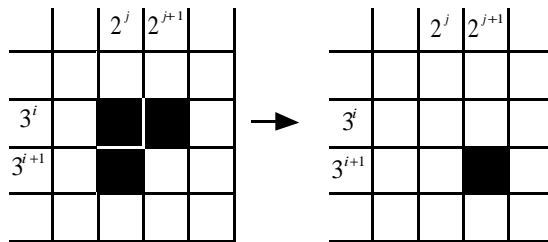


Figure 4. Reduction rule 2.

Obviously, as a result of the application of rules 1 and 2 to a number I_z , new consecutive active cells in a row might be created (corresponding to ‘carry propagation’ in binary addition). This requires that the two rules be applied successively until no adjacent active cells in a

row are left. If we assume that the binary representation of z requires n bits, it can be shown [7] that in the worst case after $\frac{n}{\lg_2 3} \approx 0.630925n$ applications of these reduction rules, the resulting map will have no adjacent active cells. This result shows a 37% improvement over ‘carry propagation’ in binary addition.

We can extend addition to provide multiplication by applying $2D$ shifts and DBNS additions. The promise here is that the number of operations is considerably reduced based on the sparseness of the representation [7].

3 Symbolic Substitution

The DBNS reduction rules are representative of the Symbolic Substitution process. Symbolic Substitution is a superset of Boolean logic in the respect that it is sensitive to both the value of the bits and their spatial location [4]. SS differs from traditional Boolean logic in the respect that a Boolean operator, for example NAND and NOR, recognizes a combination of bits and generates one bit at the output (see Figure 5 (a)), while SS recognizes not only a combination of bits but also the relative location of these bits and generates not just a single bit as in the case of Boolean logic but rather a combination of bits positioned in a particular manner at the output (see Figure 5 (b)).

SS can be decomposed into two phases:

- I. Recognition phase: Recognizing all the occurrences of a given pattern (*recognition pattern*).
- II. Replacement phase: Replacing each of the patterns detected in phase I with another pattern (*replacement pattern*) at a defined position with respect to the recognition pattern.

In this paper we employ a slightly different definition of symbolic substitution.

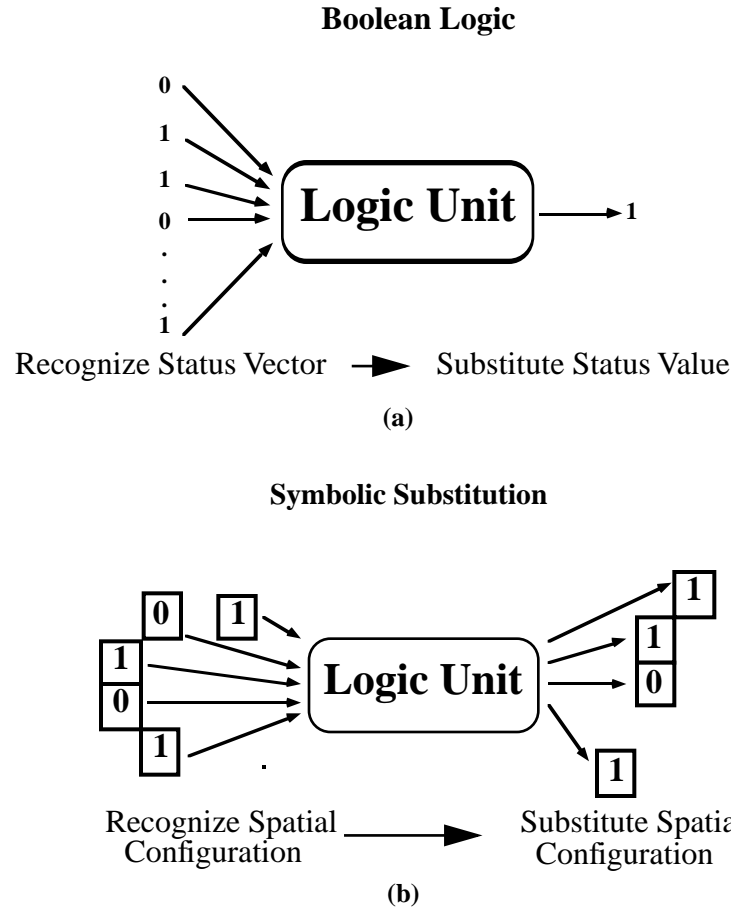


Figure 5. Comparison of Boolean logic (a) with symbolic substitution (b).

3.1 Symbolic Substitution Definitions

For convenience we refer to the recognition and replacement binary patterns as pattern **A** and pattern **B** respectively.

- I. Patterns **A** and **B** consist of the same number of pixels, N , with the same spatial configuration. We define this property as a one to one correspondence between pixels of two patterns and, as will be seen later, this condition is crucial in the implementation of digital arithmetic using CNNs¹.
- II. In the substitution process each pixel of pattern **A** is replaced by the corresponding pixel of pattern **B** as follows:
 - P white pixels of pattern **A** are replaced by P black pixels of pattern **B** at the same positions.

1. In the general definition [4], two recognition and replacement patterns can have different numbers of pixels and different spatial configuration.

- Q black pixels of pattern **A** are replaced by Q white pixels of pattern **B** at the same positions.
- R white pixels of pattern **A** are replaced by R white pixels of pattern **B** at the same positions.
- S black pixels of pattern **A** are replaced by S black pixels of pattern **B** at the same positions.

where P , Q , R and S are arbitrary numbers and $P + Q + R + S = N$.

III. A pixel is not allowed to participate in more than one SS simultaneously¹. By this we refer to the fact that this pixel may be part of more than one recognition pattern and we wish to use only one of the recognition patterns in a replacement phase.

In order to make these points clear, we present the following two examples².

Example 2: A pair of recognition and replacement patterns are shown in Figure 6. The one-to-one correspondence between the pixels of recognition and replacement patterns is shown by two-sided arrows. In this example $P = 1$, $Q = 2$, $R = 1$ and $S = 1$.

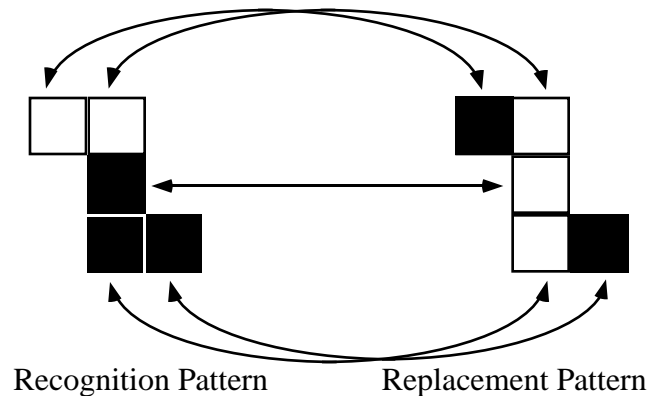


Figure 6. An example of a pair of recognition and replacement patterns and one-to-one correspondence between their pixels.

Example 3: In applying SS to the image shown in Figure 7(a), using the recognition and replacement patterns of example 1, two occurrences of the recognition pattern with two common cells can be identified. Based on our definition only one of these recognition patterns is replaced by a replacement pattern.

1. A pixel can participate in more than one symbolic substitution in the general definition [4]-[5].
 2. We have used more general examples than those provided by the DBNS reduction rules.

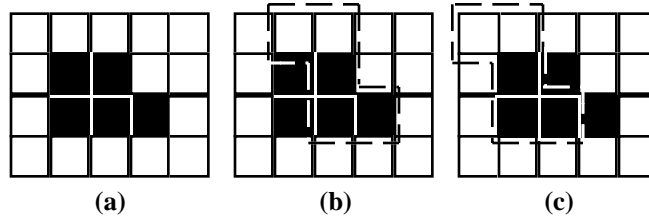


Figure 7. Given image (a) and two occurrences of the recognition pattern with two common cells (shown with dashed line (a) and (b)).

4 Symbolic Substitution using CNNs

In this section we present a class of CNNs designed to process a given binary image using the Symbolic Substitution paradigm.

4.1 A class of CNNs for Symbolic Substitution

Figure 8 shows the circuit diagram of a cell in the CNN used to process an $M \times N$ image. The structure of the cell is similar to the cell defined in [8]. The cell located at (i, j) in a $M \times N$ array is denoted by C_{ij} and its r -neighborhood N_{ij} is defined by:

$$N_{ij} = \{C_{kl} | \max(|k - i|, |l - j|)\} \leq r; \quad 1 \leq k \leq M, \quad 1 \leq l \leq N. \quad (4)$$

where r is a positive integer. The state of the cell C_{ij} at time t is denoted by $x_{ij}(t)$ (with initial condition $x_{ij}(0) \leq 1$) and the output by $y_{ij}(t)$. The output voltage is given by:

$$y_{ij}(t) = f(x_{ij}(t)) = \frac{1}{2}(|x_{ij}(t) + 1| - |x_{ij}(t) - 1|). \quad (5)$$

This nonlinear characteristic is shown in Figure 9.

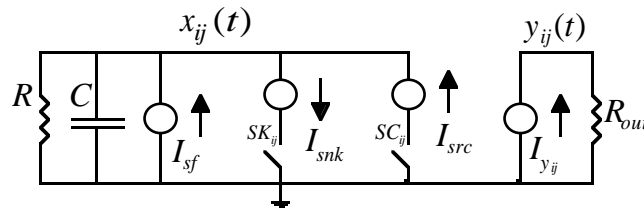


Figure 8. Circuit diagram of a cell in our CNN.

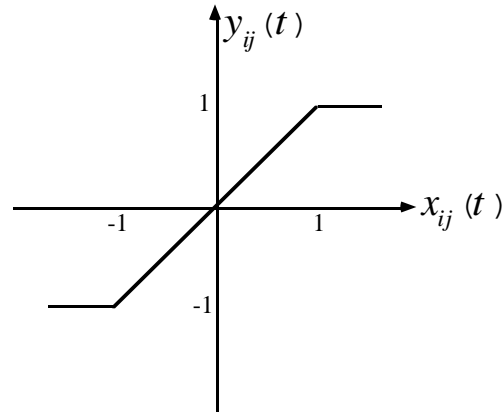


Figure 9. Cell nonlinear output characteristic.

The self-feedback current represented by I_{sf} in Figure 8., is a linear function of the cell output voltage and is given by:

$$I_{sf} = k \cdot y_{ij}(t). \quad (6)$$

The cell has two switchable current sources: $I_{snk} = -mk$ which sinks current, and $I_{src} = mk$ which sources current. m is a constant greater than 1. These two current sources are switched by SK_{ij} and SC_{ij} respectively. The binary status (*on* or *off*) of these switches are nonlinear functions of the state voltages of the cells in the neighborhood N_{ij} .

The state equation of a cell C_{ij} ($1 \leq i \leq M$, $1 \leq j \leq N$) is given by eqn. (7), where, for convenience and without loss of generality, we assume that $R = C = 1$:

$$\begin{aligned} \dot{x}_{ij}(t) = & -x_{ij}(t) + I_{sf}(y_{ij}(t)) + SK_{ij}(x_{k,l}(t))I_{snk} + SC_{ij}(x_{k,l}(t))I_{src} \\ & 1 \leq k \leq M; \quad 1 \leq l \leq N \end{aligned} \quad (7)$$

Bipolar values of the input image pixels, -1 for ‘white’ and 1 for ‘black’, are loaded as initial values of the state voltages. For binary outputs it can be shown that $k > 1$ [8].

Based on the analysis of the cell’s *dynamic route* (variation of $\dot{x}_{ij}(t)$ versus $x_{ij}(t)$) [8]-[10], when the switches SK_{ij} and SC_{ij} are open the state voltage has two *stable equilibrium points*, one at $x_{ij} = -k$ and the other at $x_{ij} = k$. Consequently, for $x_{ij}(0) > 0$ the state voltage set-

ties at k and the output voltage settles at 1. For $x_{ij}(0) < 0$ we have a final state voltage of $-k$ and an output voltage of -1 .

The transition of the state voltage from 1 to -1 , or vice-versa, is forced by switching either I_{snk} or I_{src} 'on'. The conditions under which these current sources switch will be developed in the next section.

The speed of the transition of the state voltage from one stable point to the other stable point is determined by $\dot{x}(t)$ which in turn is a function of the self-feedback current k . Although an increase in k speeds up the transition, it undesirably shifts the stable *equilibrium points* away from the origin. As a result the state voltage of the cell settles at a stable point considerably higher than the corresponding output voltage (1 or -1). Therefore, in each transition between stable points, an extra transition time is required for the state voltage to settle from $\pm k$ to ± 1 . We can reduce this transition time by reducing k from a high value to a value of $1 + \epsilon$ as soon as the state voltage reaches ± 1 . Therefore, when the switchable current sources are 'off', the cell has two stable equilibrium points, $1 + \epsilon$ and $-(1 + \epsilon)$. We approximate these two stable equilibrium points by 1 and -1 respectively.

In the remainder of the paper the term CNN refers to the structure introduced above.

5 Symbolic Substitution using CNNs

In this section we explain our method by designing a CNN to implement a given pair of recognition/replacement patterns.

Example 4: Suppose that an $M \times N$ binary image has been loaded onto an $M \times N$ CNN array. This image is to be processed using the recognition and replacement patterns shown in Figure 10. One of the pixels of the recognition pattern is arbitrarily chosen as the reference pixel. Recalling our definition of SS (Section 3), one white pixel in the recognition pattern is replaced one black pixel ($P = 1$), and two black pixels are replaced with two white pixels ($Q = 2$); finally ($R = S = 0$).

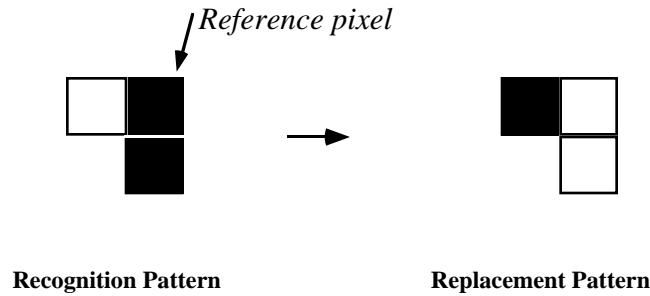


Figure 10. Recognition and replacement patterns employed in Example 3.

Recognition Phase

The recognition phase compares the state voltage of each cell with the state voltage of other selected cells in its neighborhood. In this example, the recognition pattern with its reference pixel at an arbitrary position (i, j) is detected when two black pixels at positions (i, j) , $(i + 1, j)$ and a white pixel at position $(i, j - 1)$ are detected. Therefore the state voltage of each cell at an arbitrary position (i, j) , must be compared with the state voltage of two other cells at positions $(i, j - 1)$ and $(i + 1, j)$.

We find that the following three conditions are to be satisfied:

- I.** The state voltage of the cells at positions (i, j) and $(i + 1, j)$ must have equal magnitudes and equal signs. This condition is satisfied if $V_d = |x_{i, j}(t) - x_{i+1, j}(t)| = 0$.
- II.** The state voltage of the cells at positions (i, j) and $(i, j - 1)$ must have equal magnitudes and opposite signs. This condition is met if $V_s = |x_{i, j}(t) + x_{i, j-1}(t)| = 0$.
- III.** $x_{i, j}(t) = 1$ ¹.

For conditions **I** and **II** we define, H , a function of the control voltage (V_d or V_s) as shown in Figure 11.

1. Two other equivalent conditions are $x_{i+1, j}(t) = 1$, and $x_{i, j-1}(t) = -1$.

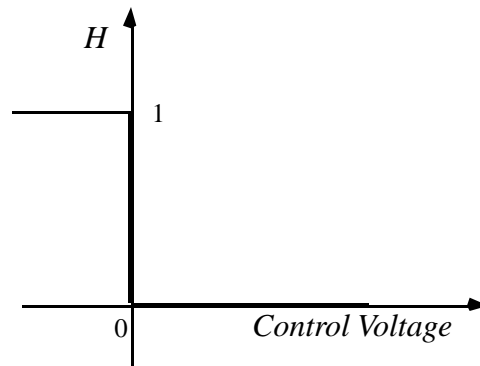


Figure 11. H as a function of V_d or V_s .

Replacement Phase

Once the recognition pattern is detected, the replacement phase starts. For this example white pixels have to be replaced with black pixels and vice-versa. This replacement is performed by forcing the state voltages of the targeted cells from their current equilibrium state to the other equilibrium state ($-1 \Rightarrow 1$ and vice-versa), by appropriately switching the current sources, I_{snk} and I_{src} in Figure 8, using the conditions defined in the recognition phase. The current sources must remain 'on' (H stable) until the state voltages pass through zero, the transition to the new stable points is then guaranteed [8].

In a VLSI implementation the time constants of the cells will not be exactly matched, which leads to different state voltage transition times and potential violation of conditions **I** and **II**. Our solution to this problem is to replace the function in Figure 11 with the hysteresis characteristic of Figure 12.

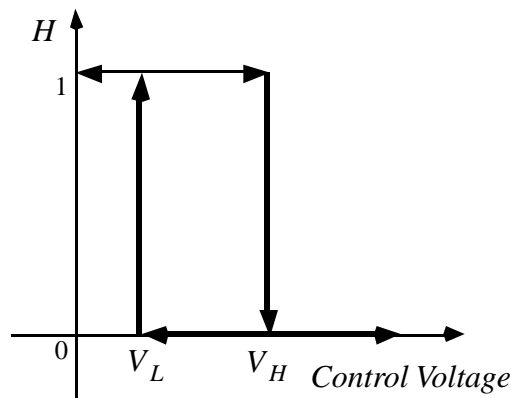


Figure 12. Hysteresis characteristic of the function H .

The hysteresis allows the switchable current sources to remain ‘on’ even if the unmatched transient speeds may cause the control voltage to reverse its direction. This greatly improves the stability of the design. In our simulations we have used values of $V_L = 0.2$ and $V_L = 0.4$ ¹.

Condition **III** is also violated as soon as the transitions start. Therefore this condition has to be modified such that, in combination with conditions **I** and **II**, the recognition patterns can be detected and held ‘TRUE’ during the replacement phase. The following considerations are used to determine the best times for turning the current sources ‘off’:

1. $I_{snk}(I_{src})$ must remain ‘on’ at least until the cell state voltage passes through zero.
2. $I_{snk}(I_{src})$ must be turned ‘off’ before the state voltage reaches the target stable point; this allows the cell to participate in any subsequent SS process.

Based on these considerations, and also to have highest noise immunity, the switchable current sources have to be turned ‘off’ when the state voltage reaches the mid point between ‘zero’ and the new stable point. This implies turning the current sources off when the state voltage reaches $\pm 0.5V$. These requirements are met by replacing the previous condition **III** with $x_{i,j}(t) > -0.5$ ².

It is noteworthy that we require only one stable point to satisfy this condition: $x_{i,j}(t) = 1$.

We summarize these three conditions for recognition and replacement of the patterns shown in Figure 10, with the reference point at (i, j) , by defining the Boolean function $A_{b;i,j}$ as follows³:

-
1. Note that the V_d and V_s (which are the absolute value of the sum and difference of two cells’ state voltage respectively) can vary in the range 0 to 2.
 2. Two other equivalent conditions are $x_{i+1,j}(t) > -0.5$, and $x_{i,j-1}(t) < 0.5$.
 3. We have used letter A to be consistent with the literature in definition of feedback template, letter ‘b’ refers to the fact that these conditions are basic conditions and will be completed later and finally i, j subscript refers to the position of the reference pixel.

$$A_{b;i,j} = \begin{cases} H(|x_{i,j}(t) - x_{i+1,j}(t)|) = 1 \\ \quad \& \\ H(|x_{i,j}(t) + x_{i,j-1}(t)|) = 1. \\ \quad \& \\ x_{i,j}(t) > -0.5 \end{cases} \quad (8)$$

Preventing the simultaneous participation of a cell in more than one SS.

Recalling our definition of SS (Section 3) a cell (pixel) is not allowed to participate in more than one SS simultaneously. To realize this requirement, we first need to consider all the possible ways a cell is able to simultaneously participate in more than one SS. In this example there is only one possible way as shown in Figure 13.

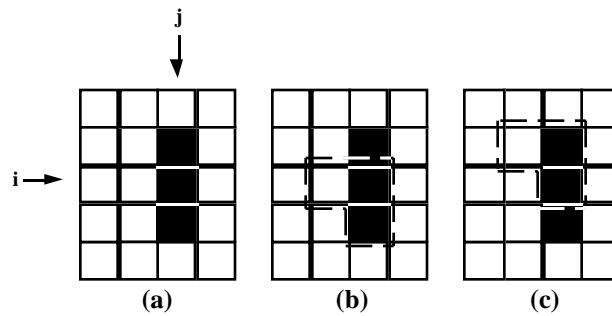


Figure 13. Possible ways a cell can simultaneously be part of more than one SS recognition pattern.

We see that a cell at position (i, j) can simultaneously participate in two different recognition patterns as follows:

1. As a reference pixel with two other pixels at positions $(i, j - 1)$ and $(i + 1, j)$ (Figure 13.(b)).
2. With two pixels at positions $(i - 1, j)$ (reference pixel) and $(i - 1, j - 1)$ (Figure 13 (c)).

For our example suppose that we consider a higher priority for the replacement pattern shown in Figure 13 (b) over the pattern shown in Figure 13 (c). In other words we only allow the recognition pattern consisting of cells at positions (i, j) , $(i + 1, j)$ and $(i, j - 1)$ to be replaced. This can be implemented by adding another condition to $A_{b,ij}$, by defining a new Boolean function A_{ij} as follows:

$$A_{ij} = \left\{ \begin{array}{l} H(|x_{i,j}(t) - x_{i+1,j}(t)|) = H(|x_{i,j}(t) + x_{i,j-1}(t)|) = 1 \\ \quad \& \\ \quad x_{i,j}(t) > -0.5 \\ \quad \& \\ \quad A_{b;i+1,j} = 0 \end{array} \right. . \quad (9)$$

Note that this condition is not unique, e.g. this condition could be replaced with $|x_{i+1,j-1}(t)| > 0.9$. The preference of one condition over any other depends on the application, complexity of the hardware implementation, etc.

Once the above conditions are satisfied for a reference pixel (cell) at an arbitrary position (i, j) , the *SC* switch of the cell at position $(i, j - 1)$ and *SK* switches of the cells at positions (i, j) and $(i + 1, j)$ are closed (see Figure 8) and the transitions start. Assuming matched dynamics for all of the cells, these switches will open when the state voltage of the cell at position $(i, j - 1)$ reaches $0.5V$ and the state voltage of the cells at positions (i, j) and $(i + 1, j)$ reach $-0.5V$ (see eqn. (7) and eqn. (9)). As soon as the state voltage of a cell reaches the new stable point, it is ready to participate in another SS providing that the corresponding conditions are met. The recognition and replacement phases continue until no recognition patterns remain.

In the situation where one or more pixels in the recognition pattern remain unchanged during the SS (see our definition in Section 3), some simple conditions are added to the definition of $A_{b;i,j}$. e.g. if we have a black pixel at position (k, l) in both the recognition and replacement patterns, the following condition is added:

$$\text{IV. } x_{k,l}(t) > 0.5$$

The techniques developed for this example, can be easily generalized to the case of arbitrary recognition and replacement patterns [13].

6 DBNS Arithmetic using CNN Arrays

In this section we illustrate the Symbolic Substitution technique by considering the implementation of DBNS arithmetic (which is naturally defined over a 2-dimensional grid). We note, in passing, that the SS technique has also been used to build binary number system adders [1].

6.1 A CNN DBNS adder

Steps 1 and 2 of DBNS addition involve evaluation of simple local Boolean functions that can be implemented using simple logic circuits or linear CNNs [6]; however, the resulting DBNS map is not ARDBNR and can not be used for another DBNS arithmetic operation (addition or multiplication). We reduce the representation to an ARDBNR form by successive application of the two reduction rules of Figure 3 and Figure 4.

Suppose that an $M \times N$ DBNS map, $I_z(i, j)$, is the result of step 1 and 2 of an addition (see Section 2). The initial value of the state voltages of the $M \times N$ CNN array are set to the corresponding pixel value in $I_z(i, j)$ (-1 for white and 1 for black pixels), and rules 1 and 2 are successively applied until no consecutive active cells remain in a row. We apply these rules with the two pairs of patterns shown in Figure 14 and Figure 15.

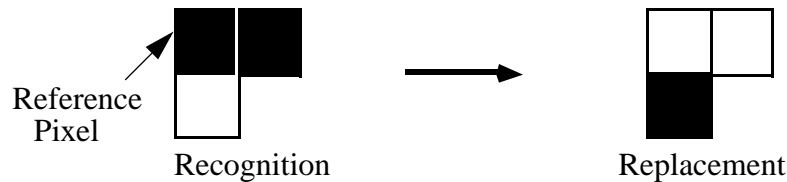


Figure 14. Recognition and replacement patterns of rule 1.

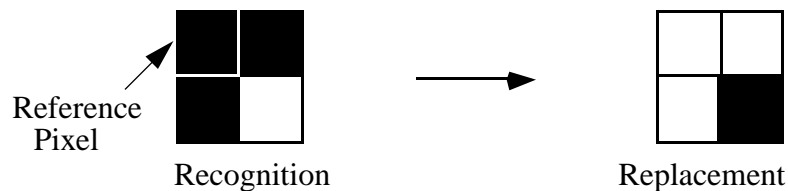


Figure 15. Recognition and replacement patterns of rule 2.

Following the general design steps presented in Section 5, to apply these two rules the conditions of eqn. (10) must be satisfied:

$$A_{b1;i,j} = \begin{cases} H(|x_{i,j}(t) + x_{i+1,j}(t)|) = 1 \\ \quad \& \\ H(|x_{i,j}(t) - x_{i,j+1}(t)|) = 1, \\ \quad \& \\ x_{i,j}(t) > -0.5 \end{cases} \quad (10)$$

To detect the recognition pattern of rule 2 the conditions of eqn. (11) must be satisfied:

$$A_{b2;i,j} = \begin{cases} H(|x_{i,j}(t) - x_{i+1,j}(t)|) = 1 \\ \quad \& \\ H(|x_{i,j}(t) - x_{i,j+1}(t)|) = 1 \\ \quad \& \\ H(|x_{i,j+1}(t) + x_{i+1,j+1}(t)|) = 1 \\ \quad \& \\ x_{i,j}(t) > -0.5 \end{cases} \quad (11)$$

where the function, H , has the hysteresis characteristic shown in Figure 12.

Preventing the simultaneous participation of a cell in more than one SS.

In the final step of the design, the possibility of simultaneous participation of a cell in more than one SS must be eliminated. Five different such cases can be identified:

1. Participation of a cell in two applications of rule 1 at the same time.

Figure 16. shows both possibilities of a cell participating in two applications of rule 1. In the initial pattern (Figure 16(a)), two occurrences of the recognition pattern 1 with the common cell at an arbitrary position (i, j) can be detected as shown by the broken line in Figure 16(b) and Figure 16(c) respectively.

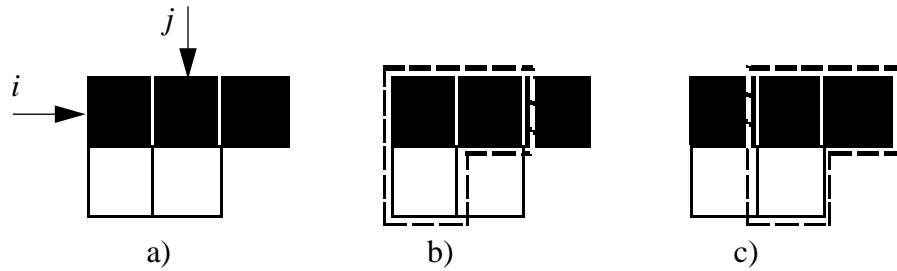


Figure 16. Participation of a cell, at position (i, j) , in two recognition patterns. Initial pattern (a); rule 1 applications (b) and (c) (shown as a broken line).

2. Participation of a cell in two applications of rule 2 at the same time.

Figure 17. shows both possibilities of the participation of a cell in two applications of rule 2.

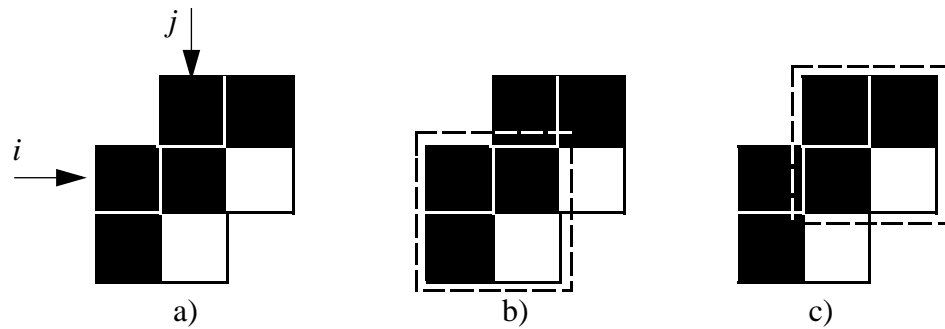


Figure 17. Initial pattern (a); rule 2 applications (b) and (c)

3. Participation of a cell in applications of rules 1 and 2 at the same time.

There are three possibilities, 3i), 3ii) and 3iii) as shown in Figure 18 to Figure 20, respectively:

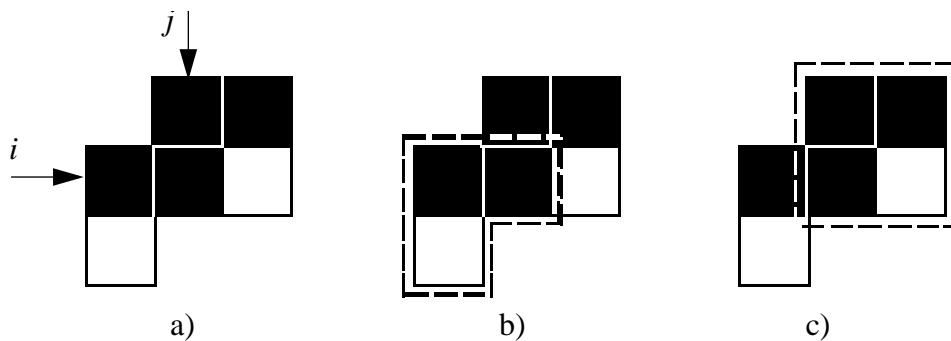


Figure 18. Initial pattern (a), first possible participation of a cell at an arbitrary position (i, j) in applications of rule 1 (b) and rule 2 (c) at the same time.

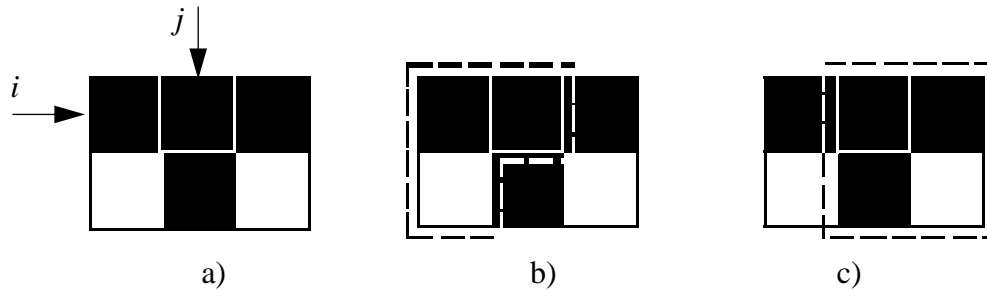


Figure 19. Initial pattern (a), second possible participation in applications of rule 1 (b) and rule 2 (c) at the same time.

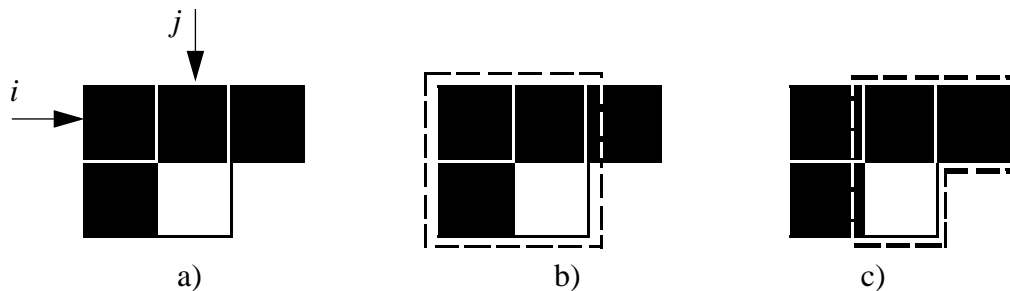


Figure 20. Initial pattern (a), third possibility of participation in applications of rule 2 (b) and rule 1 (c) at the same time.

Obviously in each of the five cases above only one of the two recognition patterns is allowed to be replaced by the corresponding replacement pattern. Choosing one possible SS over the other in these cases can be determined based on a trade off between the complexity of the resulting design and the acceptable delay.

Based on the analysis of the five cases, we can apply the following greedy algorithm to find a reduced number of extra conditions. In this algorithm we successively find additional conditions that eliminate one of the two possible SS choices *in as many cases as possible*. We repeat this process on the remaining cases until all have been covered.

1. Add the condition $|x_{i+1, j-1}(t)| > 0.8$ to the set of conditions of rule 2, given in eqn. (11), this leads to the selections in Table 1.

Table 1. SS pattern selections with addition of $|x_{i+1, j-1}(t)| > 0.8$

Case	Selected SS pattern
2	Figure 17(b)
3 i)	Figure 18(b)
3 ii)	Figure 19(b)

By adding this condition a recognition pattern of rule 2 with its reference pixel at an arbitrary position (i, j) is replaced by the corresponding replacement pattern if the cell at position $(i + 1, j - 1)$ is not participating in any SS.

2. Add the same condition, $|x_{i+1, j-1}(t)| > 0.8$, to the set of conditions of rule 1, given in eqn. (10). This will eliminate the possibility of the participation of a cell in two SS operations at the same time in the remaining cases, with the selections shown in Table 2.

Table 2. SS pattern selections by adding $|x_{i+1, j-1}(t)| > 0.8$

Case	Selected SS pattern
1	Figure 16(b)
3 iii)	Figure 20(b)

Based on the results from the previous 2 steps, the final set of conditions required for implementing the recognition and replacement phases of rules 1 and 2 are given below:

$$A_{1;i,j} = \left\{ \begin{array}{l} H(|x_{i,j}(t) + x_{i+1,j}(t)|) = 1 \\ \quad \& \\ H(|x_{i,j}(t) - x_{i,j+1}(t)|) = 1 \\ \quad \& \\ x_{i,j}(t) > -0.5 \\ \quad \& \\ |x_{i+1,j-1}(t)| > 0.8 \end{array} \right. . \quad (12)$$

$$A_{2;i,j} = \left\{ \begin{array}{l} H(|x_{i,j}(t) - x_{i+1,j}(t)|) = 1 \\ \quad \& \\ H(|x_{i,j}(t) - x_{i,j+1}(t)|) = 1 \\ \quad \& \\ H(|x_{i,j+1}(t) + x_{i+1,j+1}(t)|) = 1. \\ \quad \& \\ x_{i,j}(t) > -0.5 \\ \quad \& \\ |x_{i+1,j-1}(t)| > 0.8 \end{array} \right. \quad (13)$$

Based on the satisfaction of the rule 2 conditions given in eqn. (13), let us suppose that recognition pattern 2, with the reference pixel at position (i, j) , is being replaced with the corresponding replacement pattern (see Figure 15). If we assume matched time constants for the participating cells during the replacement process, their state voltages reach 0 at the same time (cells at positions (i, j) , $(i, j + 1)$ and $(i + 1, j)$ with falling transition and the cell at position $(i + 1, j + 1)$ with a rising transition). Unfortunately, the conditions for application of rule 1 on the pattern consisting of cells at positions (i, j) , $(i, j + 1)$ and $(i + 1, j)$ is also satisfied (see eqn. (12)). As a result, while rule 2 forces the state voltage of the cell at location $(i + 1, j)$ to fall to -1 through activating I_{snk} , rule 1 forces the cell state voltage to rise to 1 by turning on I_{src} . We refer to this as the *zero crossing problem*; it is easily avoided by preventing simultaneous activation of the cell's switchable current sources.

6.2 Conditions for switching the current sources

Based on the results presented in the previous section, the status of the cell's switchable current sources are determined as follows:

$$SK_{i,j} = A_{1;i,j} \vee A_{1;i,j-1} \vee A_{2;i,j} \vee A_{2;i,j-1} \vee A_{2;i-1,j} \quad (14)$$

Consider the recognition pattern of rule 1 with its reference pixel at position (i, j) (see Figure 14). The first and second terms on the right side of eqn. (14), provide the conditions to be satisfied for participation of a cell at positions (i, j) and $(i, j + 1)$. Recalling the recognition pattern of rule 2 (see Figure 15) the third to fifth terms provide the conditions to be satisfied for participation of a cell at positions (i, j) , $(i, j + 1)$ and $(i + 1, j)$.

$$SC_{i,j} = (A_{1;i+1,j} \vee A_{2;i+1,j+1}) \wedge SK_{i,j} \quad (15)$$

The first and second terms on the right hand side of eqn. (15) provide the conditions to be satisfied for participation of a white pixel in applications of rule 1 and rule 2, respectively. The final term on the right prevents the zero crossing problem.

6.3 Simulation results:

In this example simulation, we have used $R=32.5K\Omega$ and $C=0.4pF$ to provide a time-constant of 13nS for each cell. Figure 21 shows the complete cell structure that will be replicated in the network¹. This structure contains the basic cell ($C_{i,j}$), a set of current mirrors that are used to output the state variable to all of the neighbouring cells, and blocks that realize the recognition/replacement conditions. We demonstrate our technique on a non-canonic representation of the number 54 represented in a 4×3 map. Figure 22 shows the initial values (left) and final values of the state voltages, and demonstrates the correct operation. The simulations were performed using HSPICE. Figure 23 shows the state voltages of the cells during processing.

7 Conclusions

This paper has discussed techniques for implementing arithmetic operations using a class of Cellular Neural Networks. Our method is based on converting a given arithmetic operation task to a problem of processing a binary image using Symbolic Substitution (SS). The SS recognition phase is performed by evaluating simple conditions. For any given cell in the CNN array, these conditions are controlled by the sum or difference of the state voltage of the cell with the state voltages of certain neighborhood cells (control voltage). The location of these cells is determined by the spatial configuration of the recognition pattern(s). Stability of the SS recognition/replacement mechanism is guaranteed by the use of a hysteresis characteristic associated with the control voltages. We have demonstrated the practicality of the method by simulating an array of cells built from a standard analog cell structure. The resulting arithmetic array employing this method shows a better noise performance and to static-logic-based structures because of the reduction of switching current 'spikes' and the reduction of the slewing rate of the node voltages [11][12].

1. Note that since the swing of the state voltage of the cells is kept in the range of $[-1, 1]$, therefore

$x_{ij}(t) = y_{ij}(t)$ and the conditions can be realized using $y_{ij}(t)$ instead of $x_{ij}(t)$.

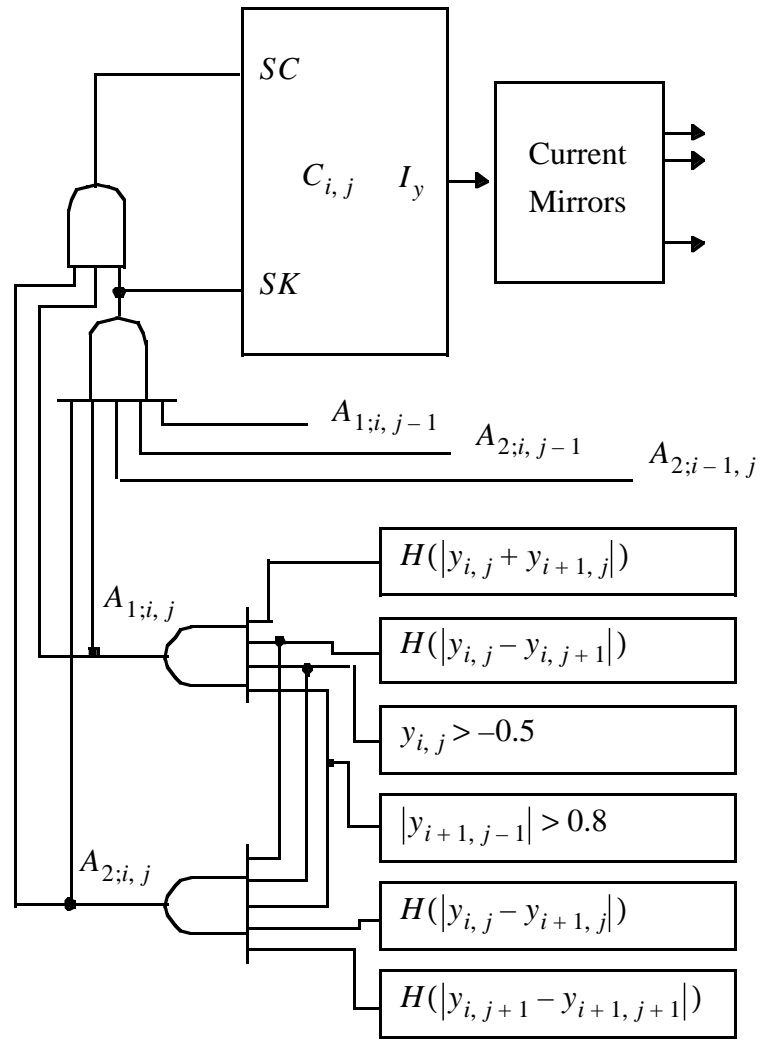


Figure 21. Block diagram of a cell at an arbitrary position (i, j) with surrounding modules in the designed CNN

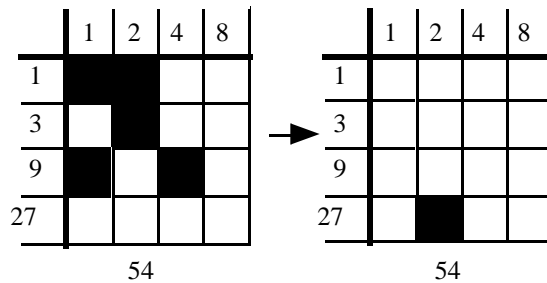


Figure 22. Initial and final values of the state voltages

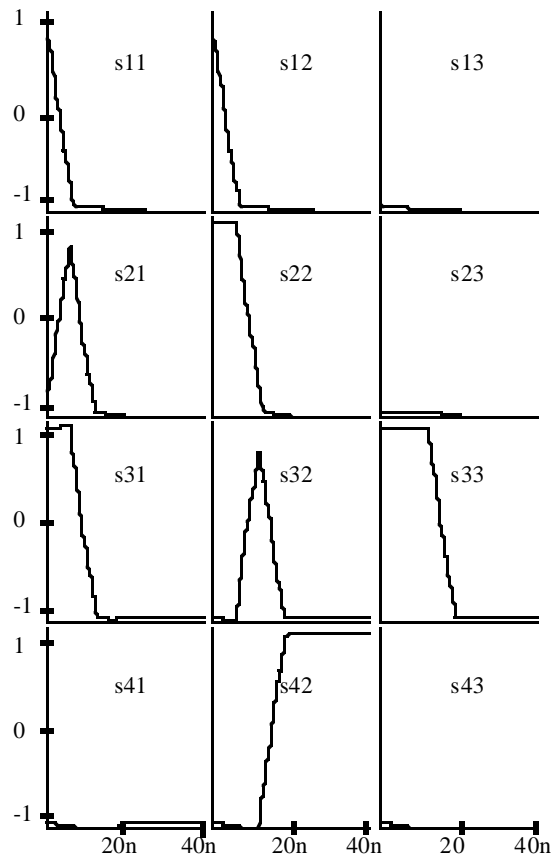


Figure 23. Cell state voltages

8 Acknowledgments

The authors acknowledge financial support from the Natural Sciences and Engineering Research Council of Canada, the Micronet Network of Centres of Excellence, Gennum Corporation, and design tools, workstations and fabrication services from the Canadian Microelectronics Corporation.

9 References

- [1] S. Sadeghi-Emamchaie, G.A. Jullien, and W.C. Miller, "Arithmetic Arrays using Cellular Neural Network" *Proceedings of 31st Asilomar conference on Signals, Systems and Computer*, Asilomar, California, vol. 2, pp.1192-1196, Nov. 1997
- [2] S. Sadeghi-Emamchaie, G.A. Jullien, V. Dimitrov, and W.C. Miller, "Digital Arithmetic Using Analog Arrays" *Proceedings of the 8th Great Lakes Symposium on VLSI*, Lafayette, Louisiana, Feb. 1998.
- [3] A. Huang, "Parallel Algorithms for Optical Digital Computers," in *Technical Digest, IEEE Tenth International Optical Computing Conference*, pp. 13-17, 1983.

- [4] K. Brenner, A. Huang, and N. Streibl, "Digital optical computing with symbolic substitution," *Applied Optics*, vol. 25, No. 18, pp. 3054-3060, 15 Sep. 1986
- [5] S. D. Goodman and W. T. Rhodes, "Symbolic substitution applications to image processing," *Applied Optics*, vol. 27, No. 9, pp. 1708-1714, 1 May 1988
- [6] Vassil Dimitrov, S. Sadeghi-Emamchaie, G.A. Jullien, and W.C. Miller, "A near Canonic Double-Based Number System (DBNS) with applications in Digital Signal Processing", *Proceeding of SPIE conference on Advanced Signal Processing Algorithm, Architecture, and Implementations VI*, Aug. 1996, Denver, Colorado, pp. 14-25.
- [7] V. S. Dimitrov, G.A. Jullien, and W.C. Miller, "Theory and Applications of the Double-Base Number System," submitted to *IEEE Transactions on Computers*.
- [8] L. O. Chua and L. Yang, "Cellular neural networks: Theory & applications," *IEEE Trans. Circuits and Syst.*, vol. 35, pp. 1257-1290, Oct. 1988.
- [9] P. Kinget, M. Steyaert, *Analog VLSI Integration of Massive Parallel Processing Systems*. Kluwer Academic Publishers 1997.
- [10] L. O. Chua, C. A. Desoer, and E. S. Kuh, *Linear and Nonlinear Circuits*. New York: McGraw-Hill, 1987.
- [11] D. J. Allstot, S. Chee, S. Kiaei, and M. Shrivastawa, "Folded Source-Coupled Logic vs. CMOS Static Logic for Low-Noise Mixed-Signal ICs," *IEEE Transaction on Circuit and Systems-I: Fundamental Theory and Applications*, vol. 40, no. 9, Sep. 1993, pp. 553-563.
- [12] G. A. Katopis, "Delta-I noise specification for a high performance computing machine," *Proceedings of the IEEE*, vol. 73, no. 9, pp.1405-1415, Sept. 1985.
- [13] S. Sadeghi-Emamchaie, "Novel Cellular Neural Network Arrays for low noise DSP Applications", Ph.D. thesis, University of Windsor, 1999.
- [14] V. S. Dimitrov, G.A. Jullien, and W.C. Miller, "Complexity and fast algorithms for multi-exponentiations," *IEEE Transactions on Computers*, 1999, in print.