TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Functional testing for cellular neural networks

Citation for published version (APA):
Willis, J., & Pineda de Gyvez, J. (1993). Functional testing for cellular neural networks. *Electronics Letters*, *29*(25), 2206-2208. https://doi.org/10.1049/el:19931482

DOI:
10.1049/el:19931482

Document status and date:
Published: 01/01/1993

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 16. Nov. 2023

$$s_{11} = \frac{s_{11,b}s_{21,a} - s_{11,a}s_{21,b}s_{21,L}}{s_{21,a} - s_{21,b}s_{21,L}}$$

$$s_{22} = \frac{s_{11,a} - s_{11,b}}{s_{21,a} - s_{21,b}s_{21,L}}$$

can be calculated. $s_{21,L} = \exp(-j\beta l)$ is the forward transmission coefficient of the connecting line of length l and propagation coefficient β. Losses are neglected.

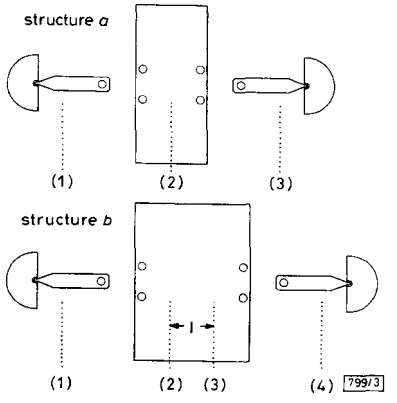

Fig. 3 *Measurement structures consisting of two different double transistions*

Results: The operating frequency of our radio module is 38GHz and the transition is matched to 50Ω. A substrate thickness of 170μm was chosen which needs a line width of 363μm for the microstrip and coplanar line. The slot width is 31μm. The via holes are processed with a CO_2 laser and have diameters of 260μm on the top side and 180–230μm on the back side. These via holes have to be filled either in a soldering process or by means of an organic material (glop top filling).

The resulting excellent return loss of ~20dB is shown in Fig. 4. Although we performed the measurement only from 36 to 40GHz we mention here that this structure can be used as a broadband transition because no frequency selective elements are incorporated.

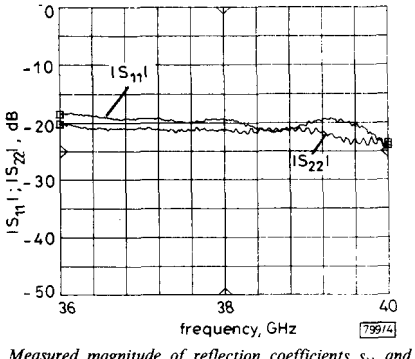

Fig. 4 *Measured magnitude of reflection coefficients s_{11} and s_{22} of transition shown in Fig. 2*

Conclusions: A novel microstrip-to-microstrip transition with side changed metallisation patterns has been experimentally demonstrated. A measurement procedure has been presented which extracts the transition properties by two different double transition measurements with electrical separation in the frequency domain. Although only measured around 38GHz the structure is suitable for a larger operating bandwidth.

Acknowledgments: The authors wish to express their thanks to Dr. Greving for helpful support and discussions and to Mr. Wagner for excellent technological realisation, both from Alcatel SEL AG, Stuttgart. This work is partially funded by the ministry of research and development.

© IEE 1993
15 October 1993
Electronics Letters Online No: 19931439

U. Meier and H. Müller (*Alcatel SEL AG, FS/ENNA, Lorenzstraße 10, D-70435 Stuttgart, Germany*)

U. Meier is now with: Fachhochschule Lippe, Liebigstraße 87, D-32657 Lemgo, Germany

References

1 IZADIAN, J.S., and IZADIAN, S.M.: 'Microwave transition design' (Artech House, Inc., 1988)
2 WILLIAMS, D.F., and MIERS, T.H.: 'A coplanar probe to microstrip transition', *IEEE Trans.*, 1988, **MTT-36**, (7), pp. 1219–1223

Functional testing for cellular neural networks

J. Willis and J. Pineda de Gyvez

Indexing terms: Cellular neural networks, Testing

A novel approach to test the functional behaviour of cellular neural networks (CNNs) is proposed. The method attains 100% stuck-at fault coverage regardless of the array size without any extra hardware for its implementation. The Letter discusses the new fault model, presents the algorithmic procedures and shows simulated testing results.

Introduction: The testing of CNNs has been scarcely addressed [1]. The approach described in this Letter overcomes the testing constraints of [1] and achieves 100% fault detection without any extra hardware. Our testing method is based on the concept of C-testability [2]. Using this approach it is possible to determine the functionality of a processing array by applying a constant number of predetermined vectors independent of the array size and then comparing the actual output values to the predicted output values.

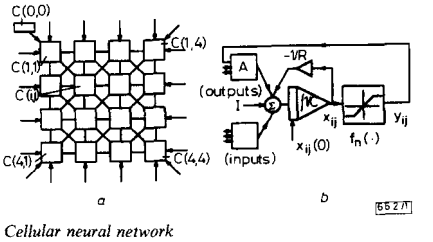

Fig. 1 *Cellular neural network*

a 4 × 4 processing array with border cell inputs shown
b Individual processing cell

A CNN is an analogue cellular nonlinear dynamic processor array (see Fig. 1*a*). The basic circuit unit is called the 'cell' [3,4] (see Fig.1*b*). The first order nonlinear differential equation defining the dynamics of a cellular neural network can be written as follows:

$$C\frac{dx_{ij}(t)}{dt} = -\frac{1}{R}x_{ij}(t) + \sum_{C(k,l)\in N(i,j)} A(i,j;k,l)y_{kj}(t) + \sum_{C(k,l)\in N(i,j)} B(i,j;k,l)u_{kl} + I$$

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

A and B are called 'templates' and are used to control the interaction between the cells $C(k,l)$ in the neighbourhood $N(i,j)$ of a reference cell $C(i,j)$. The variable $x(t)$ represents the state of the cell, u_{ij} represents the input image to the cell and y_{ij} represents the output equation.

A set of inputs is necessary to simulate interaction with

imaginary cells outside the processing array to ensure that the cells on the perimeter of the processing array achieve proper convergence. These imaginary cells are called border cells and form a ring around the processing array. The border cells are treated as members of the array for initialisation purposes and template implementation, but are not considered in the final state analysis.

Fault models: A CNN processor has only two output states. Commonly, in image processing applications these states appear as white or black pixels. A white pixel is associated to a cell whose output voltage is normalised to −1 V. A black pixel is associated to a cell whose normalised output voltage is +1 V. If a cell is unable to change from one state to the other, it is defined to be 'stuck-at-white' or 'stuck-at-black', depending on its current value. With the proposed test method it is possible to detect 100% of the stuck-at faults in the processor.

Test methods: The test procedure has two separate methods to detect faulty cells, a local method using the B template and a propagation method using the A template. The entire array can be tested using either of these methods regardless of its size. The advantage of the A template method is that it verifies that each cell is responding correctly to its neighbour's output.

The local method uses the input image and the B template to predict the final output state y_{ij} of each cell in the array. The algorithm for the local test procedure is shown below.

(i) let

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad I = 0$$

(ii) set the input image and border cells to white

$$\forall_{ij}\ u_{ij} = -1 \quad i = 0, 1, \ldots n+1 \quad j = 0, 1, \ldots m+1$$

(iii) set the initial conditions to white

$$\forall_{ij}\ x_{ij}(0) = -1 \quad i = 1, 2, \ldots n \quad j = 1, 2, \ldots m$$

(iv) allow the CNN to converge

(v) all cells, C_{ij} remaining at white are considered faulty, e.g. 'stuck-at-white' cells

$$\forall_{ij}\ y_{ij} \equiv -1 \rightarrow C_{ij} \text{ is faulty} \quad i = 1, 2, \ldots n \quad j = 1, 2, \ldots m$$

In step 2 the normalised value corresponding to a white input is −1. Notice that the B template values are also −1. The positive product of these two values results in current being injected into cell $C(i,j)$ from each of its eight neighbours. As the current is injected into the cell, the integrator voltage rises and the cell output reaches the normalised value of 1, which corresponds to black. If cell $C(i,j)$ fails to change under such overwhelming circumstances the conclusion that must be drawn is that the cell is 'stuck-at-white'. The same algorithm can be applied to find 'stuck-at-black' cells by changing all instances of white to black and −1 to 1 in steps 2 – 5 of the local test algorithm.

The test of the CNN array using the A template uses the idea of propagation of information across the network. The propagation ability of CNNs has been described before [5]. Here we use the same concept although the templates are different because we only want propagation and not 'full dragging' as described in [5]. In this case the input image does not matter and the border cells and initial conditions of the network are black. The A template causes each cell, $C(i,j)$, to look at the cell behind it, $C(i\text{-}1,j)$, and change to the colour of that cell. The algorithm for the propagation test is as follows:

(i) let

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 5 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad I = 0 \quad k = 1$$

(ii) set the border cells to black

$$\forall_{ij}\ u_{ij} = 1 \quad i = 0, 1, \ldots n+1 \quad j = 0, 1, \ldots m+1$$
$$\forall_{ij}\ u_{ij} = 1 \quad i = 0, 1, 2, \ldots n+1 \quad j = 0, m+1$$

(iii) set the initial conditions to black

$$\forall_{ij}\ x_{ij}(0) = 1 \quad i = 1, 2, \ldots n \quad j = 1, 2, \ldots m$$

(iv) allow the CNN to converge and save the results of rotation k

$$\forall_{ij}\ x_{ij}(t) \rightarrow C_{ij}^{(k)} \quad i = 1, 2, \ldots n \quad j = 1, 2, \ldots m$$

(v) add 1 to k and rotate A template clockwise 45°

(vi) if $k > 8$ continue to step 7, otherwise go to step 2

(vii) perform the logical OR of the results

$$\bigvee_{k=1}^{k=8} \forall_{ij}\ C_{ij}^{(k)} \rightarrow C_{ij} \quad i = 1, 2, \ldots n \quad j = 1, 2, \ldots m$$

(viii) all cells C_{ij} remaining at white are considered faulty, e.g. 'stuck-at-white' cells

$$\forall_{ij}\ y_{ij} \equiv -1 \rightarrow C_{ij} \text{ is faulty} \quad i = 1, 2, \ldots n \quad j = 1, 2, \ldots m$$

The process starts at the left edge of the array and propagates across the network to the right side. Because the border cells, $C(i,0)$, are black, the predicted result should be an all black image. If a 'stuck-at-white' fault is detected, then all properly functioning cells to the right of the stuck cell should also remain white. The faulty cell in effect casts its 'shadow' across the array. The situation where two or more faulty cells lie on the same row can be detected by rotating the A template clockwise 45° and repeating the test. The template should be rotated in this manner 360° to ensure complete coverage of the array. The 'stuck-at-white' cells can be determined by performing the logical OR of the resulting eight output images. The same procedure can be used to detect 'stuck-at-black' cells by changing all occurrences of white to black and −1 to 1 in steps 2 – 8 in the propagation test algorithm and performing a logical AND in step 7.

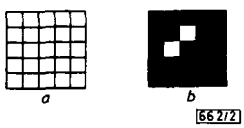

Fig. 2 *Local test method*

a Input image
b Final result

Simulated results: Using a CNN simulation program the above tests were applied to a 5 × 5 CNN network. In both the local and propagation tests, cells $C(3,2)$ and $C(2,3)$ were intentionally forced into the 'stuck-at-white' state. Fig. 2*a* shows the input image used for both tests. Fig. 2*b* shows the resulting final image after the simulation of the local test.

Fig. 3 shows the results of the simulation after the propagation test. Fig. 3*a* shows the 'shadow' effect discussed earlier. It is safe to assume by viewing Fig. 3*a* that cells $C(3,2)$ and $C(2,3)$ are faulty. Fig. 3*b–h* show the result after each rotation of the template by 45° and the new direction of propagation. The final results of the test are shown in Fig. 3*i*.

Conclusion: A testing method for CNNs has been presented which provides 100% fault detection with no additional hardware required. Any size array can be tested using a constant size set of input vectors. The local testing method provides 100% fault isolation. There are some fault location configurations that could impede fault isolation using the propagation test, i.e. if the faults form a complete rectangle, the status of the cells inside the rectangle would be unknown due to the shadow effect. This fact lowers the fault isolation capabilities of the propagation method but does

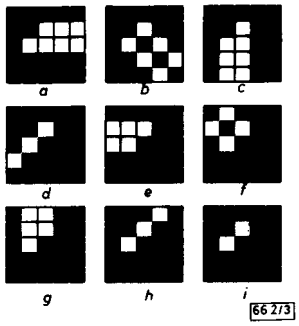

Fig. 3 *Propagation test clockwise template rotation*

a – h: 0 – 315°, 45° steps
i: Final result

not change the fault detection percentage.

© IEE 1993 *4 October 1993*
Electronics Letters Online No: 19931401

J. Willis and J. Pineda de Gyvez (*Department of Electrical Engineering, Texas A & M University, College Station, TX 77843-3128, USA*)

References

1 RUEDA, A., and HUERTAS, J.L.: 'Testability in analogue cellular neural networks', *Int. J. Circuit Theory and Appl.*, 1992, **20**, pp. 583–587
2 HUANG, W., and LOMBARDI, F.: 'On an improved design approach for C-testable orthogonal iterative arrays', *IEEE Trans.*, 1988, **CAD-5**, pp. 609–615
3 CHUA, L.O., and YANG, L.: 'Cellular neural networks: Theory', *IEEE Trans.*, 1988, **CAS-35**, pp. 1257–1272
4 CHUA, L.O., and YANG, L.: 'Cellular neural networks: Applications', *IEEE Trans.*, 1988, **CAS-35**, pp. 1273–1290
5 MATSUMOTO, T., CHUA, L.O., and SUZUKI, H.: 'CNN cloning template: Connected component detector', *IEEE Trans.*, 1990, **CAS-37**, (5), pp. 633–635

Relation between template spectrum and convergence of discrete-time cellular neural networks

R. Perfetti

Indexing terms: Neural networks, Cellular arrays

A convergence criterion is proposed for reciprocal, discrete-time cellular neural networks, that exploits the peculiar interconnection structure of such networks. It is based on the Fourier spectrum of the discrete sequence representing the cloning template, and can be applied in the usual case where the neighbourhood is much smaller than the network. The corresponding design constraint, involving the template values, results in being less restrictive than those existing in the literature.

Introduction: Discrete-time cellular neural networks (CNNs) have been recently introduced as promising architectures for nonlinear and real-time image processing [1]. The stability analysis of CNNs is a main topic that has been investigated in recent contributions (see [2] for a survey). In [3] the convergence is proven for reciprocal discrete-time CNNS with continuous, monotonically increasing nonlinearities, and feedback operator which comply with

$$A(i,j;i,j) \geq \sum_{k\neq i}\sum_{l\neq j} |A(i,j;k,l)| \qquad (1)$$

for every i and j.

This dominance constraint on the feedback operator is overly restrictive. Indeed, the convergence proof outlined in [3], that applies to every discrete-time feedback neural network, requires only that the interconnection matrix be symmetrical and positive semidefinite (PSD). It is therefore possible to exploit the peculiar interconnecting structure of CNNs to derive a design constraint which is less restrictive than that in eqn. 1. To simplify the notation, this improved constraint will be derived in the case of one-dimensional CNNs. However, the generalisation to two-dimensional networks is straightforward.

Assumptions and definitions: Assume a one-dimensional reciprocal discrete-time CNN described by the following state equation:

$$\mathbf{x}(n+1) = \mathbf{A}\,\mathbf{y}(n) + \mathbf{B}\,\mathbf{u} + \mathbf{d} \qquad n = 0,1,2,\ldots \qquad (2)$$

In eqn. 2 $\mathbf{x} = [x_1 \ldots x_N]^T$, where x_i is the state of the ith cell. $\mathbf{y} = [y_1 \ldots y_N]^T$, where $y_i = f(x_i)$ denotes the output of ith cell. N is the number of cells. $\mathbf{A}(N \times N)$ represents the symmetrical connection matrix. $\mathbf{B}$ is the matrix of control parameters. $\mathbf{u}$ is the input vector and $\mathbf{d}$ is a threshold vector. The nonlinear function $f(\,)$ is a continuous, monotonically increasing function satisfying the following conditions: $f(x) = +1$, if $x \geq +1$; $f(x) = -1$, if $x \leq -1$; $df/dx > 0$, if $-1 < x < +1$; $df/dx = 0$ if $x = \pm 1$.

Let $\{a(0)\ a(1) \ldots a(r)\}$ denote the symmetrical cloning template, where r is the dimension of the cell neighbourhood; then, the connection matrix is given by

$$\begin{aligned} A(i,j) &= a(|i-j|) \quad \text{if } |i-j| \leq r \\ A(i,j) &= 0 \qquad\qquad \text{otherwise} \end{aligned} \qquad i,j = 1,\ldots,N \qquad (3)$$

We define the template spectrum $S(\lambda)$ as follows:

$$S(\lambda) = \sum_{k=-r}^{r} a(k)\exp(j\,k\,\lambda)\,d\lambda \qquad (4a)$$

so that

$$a(k) = \frac{1}{2\pi}\int_0^{2\pi} S(\lambda)\exp(-j\,k\,\lambda)\,d\lambda \qquad (4b)$$

Taking into account the symmetry of the template we can rewrite:

$$S(\lambda) = a(0) + 2\sum_{k=1}^{r} a(k)\cos(k\,\lambda) \qquad (4c)$$

Improved convergence criterion: As shown in [3], convergence of the model eqn. 2 can be proven under the assumption that matrix $\mathbf{A}$ is PSD. This condition can be formulated in terms of the (real) eigenvalues of $\mathbf{A}$; namely it is required that all the eigenvalues of $\mathbf{A}$ be non-negative. Matrix $\mathbf{A}$ is a finite-order Toeplitz matrix, having the form

$$\mathbf{A} = \begin{bmatrix} a(0) & a(1) & \cdots & a(r) & & & \\ a(1) & a(0) & a(1) & \cdots & a(r) & & \mathbf{0} \\ \vdots & a(1) & a(0) & \ddots & & \ddots & \\ a(r) & & \ddots & \ddots & \ddots & & a(r) \\ & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & & \ddots & & \ddots & \ddots & a(1) \\ & & & a(r) & \cdots & a(1) & a(0) \end{bmatrix} \qquad (5)$$

No analytic expression is known for the eigenvalues of matrix eqn. 5. However, in the case of CNNs we are interested in networks where the number of cells is far larger than the template dimension, i.e. $N>>r$. It is therefore possible to exploit an asymptotic property of finite-order Toeplitz matrices, to derive a design constraint which is satisfied by a broader class of cloning templates, with respect to that complying with eqn. 1. To this end, we introduce matrix $\mathbf{C}(N \times N)$ defined by

$$\begin{aligned} C(i,j) &= a(|i-j|) && \text{if } |i-j| \leq r \\ C(i,j) &= a(N-|i-j|) && \text{if } |i-j| \geq N-r \\ C(i,j) &= 0 && \text{otherwise} \end{aligned} \qquad i,j = 1,\ldots,N \qquad (6)$$

where it is assumed $N \geq 2r+1$. Matrix $\mathbf{C}$ is a circulant matrix, whose circulant elements are

$$a(0)\,a(1)\ldots a(r)\,0\ldots 0\,a(r)\ldots a(1)$$

It is well known that the eigenvalues of a circulant matrix can be