

# Optimal manufacturable CNN array size for time multiplexing schemes

**Citation for published version (APA):**

Han, G., Pineda de Gyvez, J., & Sanchez-Sinencio, E. (1996). Optimal manufacturable CNN array size for time multiplexing schemes. In *Proceedings of the Fourth IEEE International Workshop on Cellular Neural Networks and their Applications, CNNA-96, 24-26 June 1996, Sevilla, Spain* (pp. 387-392). Institute of Electrical and Electronics Engineers. <https://doi.org/10.1109/CNNA.1996.566605>

**DOI:**

[10.1109/CNNA.1996.566605](https://doi.org/10.1109/CNNA.1996.566605)

**Document status and date:**

Published: 01/01/1996

**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](http://www.tue.nl/taverne)

**Take down policy**

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

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.



## Optimal Manufacturable CNN Array Size for Time Multiplexing Schemes

Gunhee Han, Jose Pineda de Gyvez and Edgar Sánchez-Sinencio

Department of Electrical Engineering  
Texas A&M University, College Station, TX 77843-3128  
Tel.: (409)862-1837, Fax: (409)845-7161, E-mail: han8613@amesp01.tamu.edu

**ABSTRACT:** This paper presents a feasibility analysis to predict the optimal size of VLSI CNN implementations. A 3x3 CNN IC test prototype was designed and fabricated for this purpose. The study considers both the manufacturability and computing performance power of hypothetical large CNN arrays. The manufacturability analysis has been geared towards IC yield prediction using our actual IC layout along with some realistic parameters representing the "cleanliness" of the manufacturing line. Additionally, from experimental results we have found that offset effects are dominant and if they are not properly canceled they can produce incorrect processing results. As a one-on-one mapping between image pixels and CNN cells is practically impossible, the computing performance analysis concentrates on the optimal array size needed to efficiently implement a multiplexing scheme versus the hypothetical fully parallel CNN architecture. Our results indicate that a 50x50 array is feasible for a time multiplexing scheme. This array will consume around 4W. The predicted yield of such array is about 70%. The implementation cost is around 30% of a 100x100 array, or alternatively only 2% of a 200x200 array, and only 0.04% slower than a hypothetical fully parallel processing architecture.

### 1. Introduction

After the CNN was introduced by Chua[1][2], many potential applications were reported in the literature. Even though the CNN architecture has great potential, the hardware implementation has a lot of problems that are mainly due to the large required array size. The main advantage of CNN is the fact that the function of the array is determined by the template. This means that the CNN can be considered as a general purpose computing architecture. Therefore, the templates including the resistor should be fully programmable. The difficulty in CNN designs is the low accuracy of the analog circuits especially when they are designed with small silicon area and low power consumption constraints. For time multiplexing image processing operations[3] an optimal array size should be sought based on the accuracy, computational speed and IC processing cost because the cost increases rapidly as the array size increases.

### 2. 3x3 CNN Test-chip

A fully programmable 3x3 CNN test chip has been fabricated using a 2 $\mu$ m double-poly double-metal CMOS process through MOSIS to investigate the possible problems that will be encountered when a larger array is fabricated. The designed CNN circuit has not only programmable templates but a programmable resistor as well. The size of a cell is 0.16mm<sup>2</sup> and the static power consumption is 1.5mW/cell.

The main circuits in our proposed CNN are multipliers and integrators. Figure 1(a) shows the circuit diagram of the limiter and the multiplier. The limiter provides some tolerance against the charge leakage through the non-ideal switches during the initialization stage. The same circuit topology is used to implement a programmable resistor. In the case of the resistor, the differential pair in the limiter is replaced with a linearized differential pair[4] for wider dynamic range. The implemented integrator uses an opamp to reduce the loading effect of the multipliers. A transistor (not shown in the figure) is provided at the summing node of each cell. Even though the single-ended implementation has higher offset than the fully-differential implementation, the multiplier and the integrator are chosen to be single-ended to avoid complex interconnections and common-mode feedback circuits. A current mirror is provided for each multiplier instead of implementing a large one at the summing node. This

scheme distributes the devices over the silicon and helps to make the offset mean to be around zero. Several control switches are provided around the integrator (see Figure 1(b)). The switches operate as follows:

1. Turn off the *start* and reset all the cells at the same time.
2. Set address to turn on the *select<sub>j</sub>*, Turn on the  $\phi_1$ , and apply  $x_j(0)$  to the input node.
3. Turn on the  $\phi_2$  and apply  $u_j$  to the input node.
4. Repeat 2 and 3 for each cell.
5. Turn on the *start*.
6. Read out the result cell by cell.

The *start* switch should be kept on until all the results are read out because the turning off the switch causes the feedthrough.

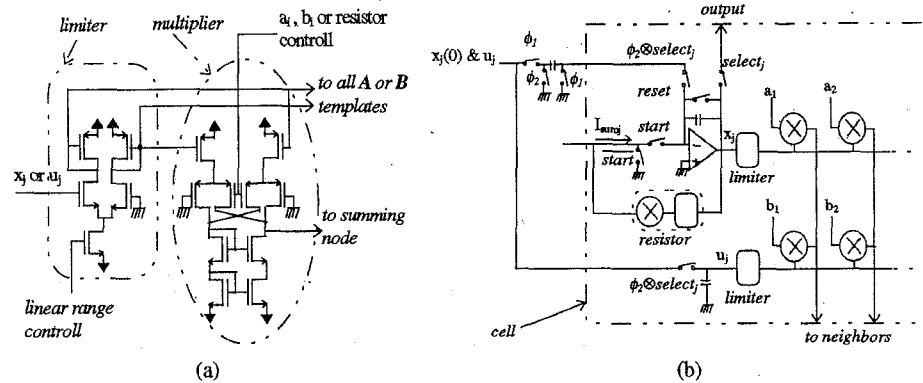


Figure 1: Circuit diagram of fabricated test chip: a) Limiter and multiplier, b) Block diagram of a cell

### 3. Yield Analysis

A functional yield analysis was carried out based on the layout of our design. This analysis allows us to study the manufacturing feasibility of large CNN arrays taking into account the environmental conditions of the silicon fab. Parametric faults are not considered and only the functional behavior is inspected.

In mature manufacturing lines spot defects are the main detractors in the successful outcome of an IC. Their manifestation is as local disturbances of silicon layer structures mainly caused by dust particles, process variabilities, and contamination of the fabrication equipment. Spot defects are in essence random phenomena occurring on the wafer with certain stochastic size and frequency per unit area (defect density). In order to verify the robustness of the design when it is exposed to defects in a real manufacturing environment, it is necessary to extract its critical areas. The so called critical areas are the places in the layout where a defect can induce a catastrophic behavior in the IC. For instance, an extra material spot defect can cause a short and a missing material spot defect can in turn induce an open circuit. A figure of merit that measures the design's vulnerability is obtained as the ratio of the total critical area to the total layout area. This figure of merit is known as defect sensitivity. Figure 2(a) shows the critical areas of layer metal-1 for an extra material spot defect of size of  $6\mu m$ . Metal is shown in gray and critical area in black. Figure 2(b) presents the layout's defect sensitivity for both bridges and opens for a range of defect sizes.

By combining the defect sensitivity with the actual defect size distribution existing in the manufacturing line one can predict the design's average probability of failure. This is evaluated as the integral of the sensitivity times the defect size distribution. Finally, yield is the probability of manufacturing ICs without faults taking into account the previously discussed issues. In our projections we are using the negative binomial yield formula

$$Y = \left(1 + \phi n A_c D / \alpha\right)^{-\alpha} \quad (1)$$

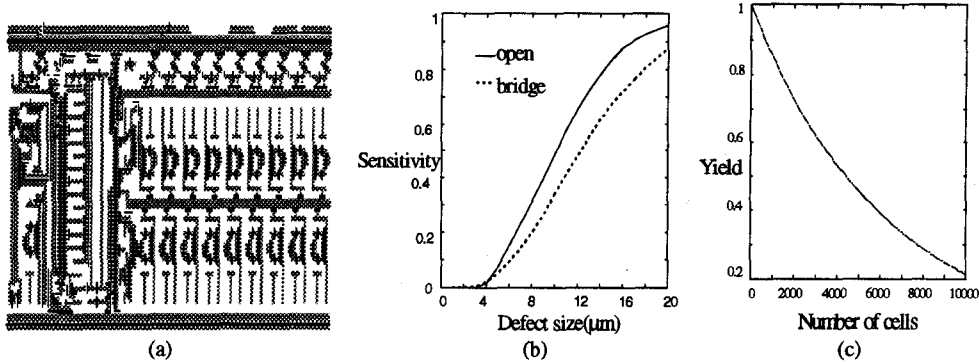


Figure 2: Yield analysis of designed chip: a) Critical area in a cell for a  $6\mu\text{m}$  spot defect, b) Sensitivity of metal-1 layer, c) Expected yield

where  $A_c$  is cell area,  $n$  is the number of cells,  $D$  is the defect density,  $\alpha$  is a defect clustering parameter, and  $\phi$  ( $\phi = 0.15$ ) is the probability of failure taking into account design and environmental conditions. For our simulations we assumed a defect size distribution that follows the  $1/x^3$  law[5] a peak defect size of  $1\mu\text{m}$ , a defect density of  $1\text{ def/cm}^2$ , and  $\alpha = 2$  which is a reasonable clustering. These parameters are representative of "clean" manufacturing. Figure 2(C) presents the corresponding yield projections up to an array size of 10,000 cells ( $100 \times 100$  array). These results show that with the current technology and present design a  $100 \times 100$  array is unfeasible since only a 21% yield can be attained. Other approaches such as MCMs or multiplexing with smaller chip sizes should be sought.

#### 4. Optimal Multiplexing Level

Originally, the CNN is intended to be a fully parallel processing architecture. The hardware implementation of a large array that can process a practical image size is extremely difficult and of high cost with the current integrating technology. It is true that fully parallel processing realizes high processing speed. However, the speedup over the multiplexing pseudo-parallel processing is not significant when the image size is large unless any fully parallel input and output scheme is provided. In a fully parallel architecture with sequential I/O data, the processing time of a fully parallel implementation,  $t_p$ , for an image of  $n$  pixels can be approximated as

$$t_p = nt_c + t_d \quad (2)$$

where  $t_c$  is the time to read and write a pixel, and  $t_d$  is the time spent for the dynamics until all the cells are converged. The  $n$ -pixel image can be divided into  $k$  sub-images and then be processed by an  $n/k$  array of cells by sweeping over the whole image once. If the feature size is much smaller than the image size and the degree of multiplexing is not high then I/O data time for overlaps is negligible. Therefore, the total processing time of a multiplexing implementation,  $t_m$ , can be roughly evaluated as

$$t_m = k \left( \frac{n}{k} t_c + t_d \right) = nt_c + kt_d \quad (3)$$

The efficiency,  $s$ , of multiplexing processing over fully parallel processing can be found as a "factor" that measures how slower is one approach versus the other. We call this factor "slow-down" and defined it as

$$s = \frac{t_m - t_p}{t_p} = \frac{k-1}{n+1} \quad (4)$$

under the assumption that  $t_c = t_d$ . Clearly, the slow-down factor due to multiplexing gets smaller as the image size increases.

The manufacturing cost of integrated circuits is roughly inversely proportional to its yield. Based on the yield model in (1), the cost of the  $n/k$  cells array can be normalized by the cost for  $k=1$  as

$$C = \frac{c(k)}{c(1)} = \frac{\left(1 + \phi \frac{nA_o}{2k}\right)^2}{\left(1 + \phi \frac{nA_o}{2}\right)^2} = \left(\frac{2 + \phi \frac{n}{k} A_o}{2 + \phi n A_o}\right)^2 \quad (5)$$

The optimal multiplexing level,  $k$ , for given image size,  $n$ , can be found by the L-curve optimization method[6] considering processing time and implementation cost. Figure 3 shows L-curves for  $n=100 \times 100$  and  $n=200 \times 200$ . The corners of the curve represent points where the speed up starts to saturate as the cost increases. In other words, it is the point where significant cost savings are obtained without significant high processing time penalty.

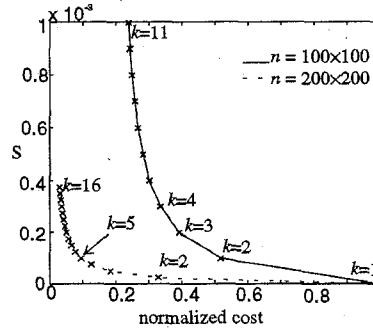


Figure 3: L-curves to find the optimal multiplexing level

From Figure 3, the optimal multiplexing levels are obtained at  $k = 3$  for a  $100 \times 100$  image size, and at  $k = 5$  for  $200 \times 200$  image size. The cost of a  $50 \times 50$  array (corresponding to  $k = 4$  for  $n = 100 \times 100$  and  $k = 16$  for  $n = 200 \times 200$ ) is around 30% of a  $100 \times 100$  array and around 2% of a  $200 \times 200$  array while keeping the slow-down factor less than 0.04%. The advantage of the multiplexing is clearer for large images. Of course this is a crude approximation. The time evaluation is dependent on the architecture of the circuits and required overlap size. The cost evaluation is dependent on the process. If a detailed time evaluation and a cost function are obtained then this method can determine the optimal degree of multiplexing. The one advantage of this L-curve method is that the time evaluation and cost are not necessary to be an analytical function. They can be numerical data with respect to the array size.

### 5. Offset Cancellation

The CNN is a system of nonlinear differential equations with sparse coefficient matrix expressed as

$$\left\{ C \frac{dx_j}{dt} = -\frac{1}{R} x_j + \sum_{i \in N_j} [a_i f(x_i) + b_i u_i] + I \right\}_{j \in \Omega} \quad (6)$$

where  $C, R, a_i, b_i, I$  are constants,  $f$  is a nonlinear function,  $u_i$  is an external excitation,  $\Omega$  is an index set and  $N_j$  is a subset of  $\Omega$ . A large number of scientific and engineering problems belong to this family of a system. Even though the equation itself has very simple form, its hardware has some random process variations. Considering these variations, equation (6) becomes a system of stochastic nonlinear differential equations. If a building block is modeled by  $(a + \tilde{a})(s) + a_o$ , where  $s$  is input,  $a$  is desired gain,  $\tilde{a}$  is variation of gain and  $a_o$  is output referred offset. Then (6) for the  $j$ th cell can be modified to

$$(C + \tilde{C}_j) \frac{dx_j}{dt} = -\left(\frac{1}{R} + \frac{1}{R_j}\right) x_j + I_{R_j} + \sum_{i \in N_j} [(a_i + \tilde{a}_{ij})(f(x_i) + \tilde{f}_i(x_i) + f_{io}) + a_{ij_o} + (b_i + \tilde{b}_{ij})(u_i + \tilde{u}_i + u_{io}) + b_{ij_o}] + (I + I_{j_o}) \quad (7)$$

Some manipulation of (7) leads to

$$\begin{aligned}
 (C + C_j) \frac{dx_j}{dt} = & -\frac{1}{R} x_j + \sum_{i \in N_j} [a_i f(x_i) + b_i u_i] + I \\
 & - \frac{1}{R_j} x_j + \sum_{i \in N_j} [a_i \tilde{f}_i(x_i) + \tilde{a}_{ij} (f(x_i) + \tilde{f}_i(x_i)) + b_i \tilde{u}_i + \tilde{b}_{ij} (u_i + \tilde{u}_i)] \\
 & + \sum_{i \in N_j} [(a_i + \tilde{a}_{ij}) f_{i_0} + (b_i + \tilde{b}_{ij}) u_{i_0} + b_{i_0}] + I_{R_i} + I_{j_0}
 \end{aligned} \quad (8)$$

The first part in the right half side of (8) is identical to (6). The second part is the dynamic perturbation that is dependent on the trajectory of state and input. This is mainly caused by the transconductance variation of the devices. The third part is the static offset that is caused mainly by device mismatches. It can be canceled out by providing additional compensation current for each cell.

Once the array is fabricated, the offset is not directly observable. If the input of the hard limiter can be disconnected from the state node then the following indirect cancellation scheme will cancel out the static offset.

1. Ground the input of the hard limiter. Set  $u_i$ ,  $R$  and  $I$  to zero. Set the desired templates,  $a_i$  and  $b_i$ , for all cells.

This eliminates the first two parts in (8).

2. Start the dynamics and tune the compensation current,  $I_{compj}$ , so the  $x_j$  stays equal to zero.

Even though this scheme cannot cancel out the dynamic perturbation, it will cancel out all offset effect. This scheme requires tuning circuits in every cell. This additional circuitry may increase the silicon area and power consumption and makes the implementation of large array more difficult. A global compensation method was discussed in [7] but as the area size becomes very large the global compensation becomes less effective.

## 6. Optimal Template Scaling with Programmable Resistor

The allowable absolute state voltage,  $x_{allowed}$ , is determined by the power supply. As a result, the allowable absolute current,  $I_{allowed}$ , that can be injected to the lossy integrator is limited by

$$I_{allowed} = \frac{x_{allowed}}{R} \quad (9)$$

for proper operation with fixed  $R$ . While the maximum absolute current that is injected into the integrator is obtained from

$$I_{max} = K \sum |t| + I \quad (10)$$

where  $K$  is a fixed constant related to the conductance of the multiplier. The term  $t$  is a signal corresponding to  $a_i$  and  $b_i$ . The template values,  $t$ , should be chosen within the following two constrains.

$$\begin{cases} I_{max} \leq I_{allowed} \\ |t| \leq t_{max} \end{cases} \quad (11)$$

where  $t_{max}$  is multiplier's absolute input range.

If  $R$  is too high then  $t$  cannot use the maximum allowed input range due to the limited  $I_{allowed}$ . On the other hand, if  $R$  is too small then the state,  $x$ , utilizes only a small portion of the maximum allowed dynamic range,  $x_{allowed}$ . Since the actual  $I_{max}$  is dependent on the templates, the linear range of the hard limiter should be small enough to secure the saturation for various templates. These situations are not desirable because the maximum possible signal swing should be used to minimize the offset effect. There are two solutions for this problem. One is to have a variable power supply for the integrator and the other is to have a variable resistor. The variable resistor may be an easier solution. With a programmable resistor, the template scaling becomes straight forward as follows.

1. Select  $t_{max}$  for the largest template value from  $a_i$  and  $b_i$ .
2. Scale other template values keeping the ratios.

3. Calculate  $I_{max}$  by (10).
4. Choose  $R$  by (9).

## 7. Experimental Results

Four 3x3 array chips were fabricated through a 2 $\mu$ m double-poly double-metal CMOS process. In our design, only the state of the center cell is observable in continuous time. Four chip had random offset and it significantly degrades the performance of the chip.

Fig 4 shows the states of the center cell measured from four chips with different conditions. Figure 4(a) shows the random offset effects of the cells in different chips when all parameters are set to zero. Figure 4(b) shows the offset effect with  $u_{22}=1$ ,  $b_{22}=1$  and all other parameters set to zero. One out of four cells converges to the wrong direction. Figure 4(c) shows the result of offset compensation with  $u_{22}=1$ ,  $b_{22}=1$  and all other parameters set to zero. The behavior of all four chips becomes very close to each other after offset cancellation.

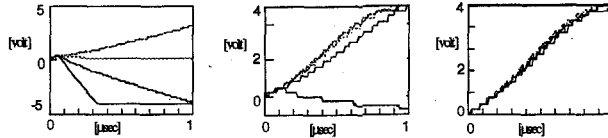


Figure 4: Offset effect and its cancellation: a) state of a cell when all weights are zero b) state of a cell with  $u_{22}=1$ ,  $b_{22}=1$ , c) state of a cell with  $u_{22}=1$ ,  $b_{22}=1$  after cancellation

## 8. Conclusions

The fabricated test chip shows that only a few chips from the non-defected chips will work properly. This means that if the rate of pass in the testing and the testing cost are included in the cost, then the actual cost may be higher than the cost model used in this paper.

Considering a square image, the optimization analysis suggests an optimal 50x50 array for a 100x100 image size. This array size is a sub-optimal array size for a larger image. The yield results are around 70% yield for this 50x50 array. The implementation cost is only 2% of that of a 200x200 array while the slow-down factor is less than 0.04%. Considering all these points of view, a 50x50 array is deemed as an optimal array size with the current technology. Since the multiplexing architecture is suitable to adopt a pipelining scheme, real-time image processing may be achieved at low cost. If a new integration technology is introduced the optimal array size will be increased and the possible image resolution will be increased.

## 9. References

- [1] L. O. Chua and L. Yang: "Cellular Neural Networks: Theory," *IEEE Trans. Circuits Syst*, Vol. 35, pp. 1257-1272, Oct. 1988.
- [2] L. O. Chua and L. Yang: "Cellular Neural Networks: Applications," *IEEE Trans. Circuits Syst*, Vol. 35, pp. 1273-1290, Oct. 1988.
- [3] C. C. Lee and J. Pineda de Gyvez: "Time Multiplexing CNN Simulator," *1994-International Symposium on Circuits and Systems*, pp. 407-410, 1994.
- [4] F. Krummenacher and N. Joehl: "A 4-Mhz CMOS Continuous-time Filter with On-Chip Automatic Tuning," *IEEE J. Solid-state Circuits*, Vol. 23, pp. 750-758, June 1988.
- [5] C. H. Stapper: "Modeling of Defects in Integrated Circuit Photolithographic Patterns," *IBM J. Res. Develop.*, vol. 28, no. 4, pp. 461-475, Jul. 1984.
- [6] P. C. Hansen: "Analysis of Discrete Ill-Posed Problems by means of the L-Curves," *SIAM Review*, Vol. 34, no. 4, pp. 561-580, Dec. 1992.
- [7] M. Grimaila and J. Pineda de Gyvez: "Parametric Fault Identification and Dynamic Compensation Techniques for Cellular Neural Network Hardware," accepted for publication in *IEE Proc. Circuits Devices and System*.