Implementation of Time-Multiplexed CNN Building Block Cell

K.K. Lai and P.H.W. Leong
Systems Engineering Design Automation Laboratory
Department of Electrical Engineering, Sydney University, NSW, Australia.
E-mail: kamlai@sedal.su.oz.au

Abstract

We have proposed an area efficient implementation of Cellular Neural Network by using the time-multiplexed method [6]. This paper describes the underlying theory, method, and the circuit architecture of a VLSI implementation. Spice simulation results have been obtained to illustrate the circuit operation. A building block cell of a time-multiplexed cellular neural network has been completed and is currently being fabricated. We expect to deliver test results at the Conference.

1: Introduction

Cellular neural networks are a parallel computing paradigm characterised by locally connected architecture. They are particularly suitable for image processing. Although the computing elements are localised, the dynamics of transient evolution propagates throughout the network and global processing of information is achieved. The local nature of interconnection amongst the computing elements is particularly suited to VLSI implementation for the obvious reason of reduced area spent on routing of connections.

The design of general purpose, programmable cellular neural network (CNN) chips with dimensions required for practical applications still remains a challenge. In a standard CNN (SCNN), a programmable basic cell would require 18 multipliers (9 each for feedback (a_m) and feedforward (b_m) template coefficient values). Thus, the chip area is proportional to $18N^2$, where N is the array size. For example, a programmable CNN chip by Halonen et al. required 1 mm^2 per cell [4]. Another programmable CNN chip by Lim et al. had an area of 0.4 mm^2 per cell [7]. A recent chip by Kinget et al. [5] had a cell size of 0.26 mm^2 (see Table 1). For practical image processing applications, a simple implementation of a programmable CNN requires a prohibitively large area.

To reduce the number of multipliers, instead of using 9 multipliers, we propose using one multiplier 9 times in a multiplexed fashion. With this scheme, a time multiplexed CNN (TMCNN) cell requires only 2 multipliers, one each for a_m and b_m values respectively. This technique results in large area saving in implementing a fully programmable CNN.

2: Mathematical model of time-multiplexed CNN (TMCNN)

The circuit model equation of a CNN (assuming single layer and unity neighbourhood size) can be written as: [1]

$$\frac{d}{dt}V_{xij} = \frac{1}{C}(-\frac{V_{xij}}{R}+\sum_{m=1}^{9} a_m V_{yij}^m+\sum_{m=1}^{9} b_m V_{uij}^m+I) \quad (1)$$

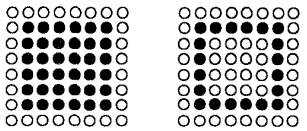
where V_{xij} is the cell state, m an index, being a direction indicator (*NE, N, NW, ..., S, SW*) around a cell *C(i,j)*, V_{yij}^m the cell output corresponding to the neighbour of cell *C(i,j)* in direction m, V_{uij}^m the external neighbour input, I the bias and a_m and b_m the coefficients in the feedback A-template and feedforward B-template respectively.

In order to perform the time multiplexing, eqn. 1 is rewritten as:

$$\frac{d}{dt}V_{xij} = \frac{1}{C}(-\frac{V_{xij}}{MR} + a_m V_{yij}^m + b_m V_{uij}^m + I/M) \quad (2)$$

where m is varying from 1 to 9 (ad infinitum). A factor M is included in the equation owing to the multiplexing process and M varies according to the number of active template coefficients. The maximum value of M is 9 for a single layer CNN with unity neighbourhood size. Contributory components in the same direction m for all cells are obtained by performing summation over

1086-1947/96 $5.00 © 1996 IEEE
Proceedings of MicroNeuro '96



(a) (b)

Figure 1. Edge detection of a square pattern of array size 8 x 8. A dot denotes a black pixel and a circle a white pixel. (a) Original pattern. (b) Final pattern after detection.

a periodic pulse of width T through small incremental time steps.

A C-program was written to solve the differential equation (2) using the Runge Kutta Method. The software was successfully applied to perform edge detection with the conditions in equations(3)-(5) and the results are shown in fig. 1.

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (3)$$

$$B = \begin{pmatrix} -0.25 & -0.25 & -0.25 \\ -0.25 & 2.0 & -0.25 \\ -0.25 & -0.25 & -0.25 \end{pmatrix}, \quad (4)$$

$$I = -0.2. \quad (5)$$

The edge detection result for an 8 x 8 array CNN was successful and the results shown in fig. 1.

3: TMCNN cell

In a TMCNN building block cell, some multipliers are replaced with transmission gates (see fig. 2). Only 2 time multiplexed core multipliers are used for the A and B templates. The timing diagram of the driving pulses for the transmission gates and the gain control voltage waveforms for the time multiplexed multipliers (A and B multiplier) are shown in the inset of fig. 2. The a_m and b_m coefficients are implemented by applying their values in a time multiplexed manner on $V_A(t)$ and $V_B(t)$ (see fig. 2). Each template coefficient will be active for one-ninth of the time since there are 9 coefficients each in A- and B-templates respectively. Corresponding coefficients in the two templates are concurrently active. While the *mth* coefficients in the A and B templates are active, the *mth* neighbouring output V_{ym}^m and the *mth* input source V_{um}^m will be sent to the A and B multipliers respectively (as shown in fig. 2). The timing sequence of the driving pulses control which and when V_{ym}^m and V_{um}^m will be connected to the inputs of the A- or B-multiplier respectively by the transmission gates. Assuming that the A- and B-templates are space invariant, the same multiplexing pulse V_{pm} will drive all the cells in the CNN with the corresponding V_{ym}^m and V_{um}^m. The appropriate gain of the multipliers are set by the gain control voltage waveforms $V_A(t)$ and $V_B(t)$. The dynamic evolution of the CNN occurs as a result of the continuous integration process across the state capacitors.

3.1: The multiplier circuit

The multiplier is a Gilbert four quadrant multiplier circuit as described by Mead [8]. The gain and sign of the multipliers are determined by the magnitude and sign of a_m and b_m values. These are globally programmable by the common $V_A(t)$ and $V_B(t)$ voltage sources. In general, $V_A(t)$ and $V_B(t)$ are multi-level voltage waveforms.

We have adopted a differential mode operation for the multiplier. The differential mode operation will suppress common mode noise in the circuit. Transmission gates are used to implement the multiplexing scheme. The switching process gives rise to quantisation noise and clock feedthrough error owing to the charge injection effect. The differential mode operation alleviates some of these problems.

3.2: The active resistor

We make use of a MOS linear resistor as the active resistor (a strict linearity may not be required but our active resistor exhibits a large linearity throughout the signal range in our operation). The configuration is similar to that of Wang [10]. The active resistor consists of 2 p-transistors connected in series. The lower device has its substrate node attached to the source in order to eliminate the body effect. For an n-well process, a separate n-well is required for this transistor. The active resistor requires less silicon area than a passive counterpart and the value is given by the expression:

$$R_{in} = \frac{1}{k_p} \frac{1}{V_{dd} - |V_T|} \frac{L}{W} \quad (6)$$

The active resistor is used as a load in the summing and squashing circuit in fig. 2.

3.3: The MOS gate capacitor

We use MOS gate capacitors for the state node capacitors. These capacitors have the advantage of higher

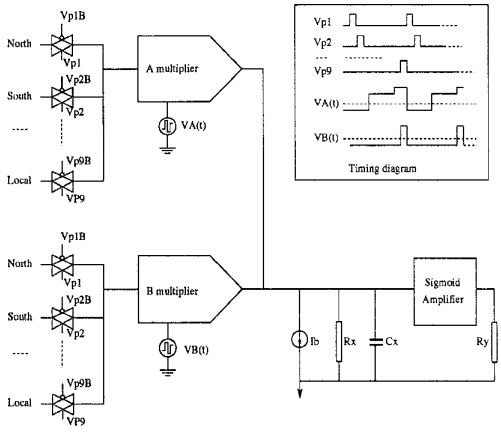


Figure 2. The TMCNN building block basic cell. The inset shows the timing diagram of the driving pulses and the multipliers gain control voltage waveforms $V_A(t)$ and $V_B(t)$.

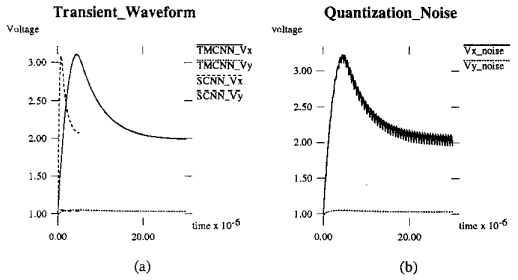


Figure 3. The transient voltage waveform of a cell circuit C(2,2) in a 4 x 4 array of a TMCNN and SCNN. (a) The transient response of the TMCNN is delayed as shown. (b) Noise ripples are associated with the time multiplexing scheme when the multiplexing pulses do not have sufficiently short pulse width.

value and less stray components compared with those fabricated using double polysilicon structure. The latter feature is considered especially important since the stray capacitance will affect the circuit dynamics. The signal swing across the state nodes have bipolar polarities and it is necessary to use a pair of nfet and pfet transistors connected back to back.

3.4: The transmission gates

Transmission gates are designed with minimum geometry to reduce clock feedthrough error and hence charge injection.

4: Spice simulation results

We interconnect a number of basic cells to form the TMCNN array. The cells are fully programmable to have different feedback and feedforward template values by programming the voltages values of the voltage sources $V_A(t)$ and $V_B(t)$. The technological parameters were those of Orbit Semiconductor's 2 μm double metal, double polysilicon, n-well CMOS process. The inputs, state variables and the bias must be scaled appropriately [9] before the circuit can operate properly according to the state equation in eqn.(2).

4.1: Trajectory of Chua's 4 x 4 array

The multiplexing technique was applied to the CNN described in Chua and Yang [1] with the same initial conditions as in fig. 9a of [1]. We used the same templates and bias conditions in [1], namely:

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad B = 0, \quad I = 0. \qquad (7)$$

Similar circuits as in [1] were used except that time multiplexing was used instead of a parallel implementation. The trajectory of the cell $C(2,2)$ (for the cell location, see [1]) was monitored both for a standard CNN (SCNN) and our TMCNN. The transient waveform is shown in fig. 3. When time multiplexing, it is not necessary to compute the zero a_m values and so in this case, $M = 5$. Our results were exactly the same for both types of CNN except that the multiplexing process delayed the circuit response by $M(= 5)$ times. Noise owing to switching normally occurred in the TMCNN waveform, as shown in (b) of fig. 3. The quantisation noise ripples disappeared when the multiplexing pulse width T was small enough (≤ 40 ns, in the present case).

4.2: Horizontal line detection

The TMCNN cell was also successfully simulated as a horizontal line detector. The input condition was that of Chua's fig.1 [1]. The input vector I_{in} is a 4 x 4 array as shown in eqn.(8). Using a template of A (eqn. 10), we obtain the output image of the output vector OUT shown in eqn. (9). Fig. 4 shows the trajectories of the state variables V_x and outputs V_y of two cells $C_{0,1}$ and $C_{2,1}$ which changed their signs during the processing period of the horizontal line detection.

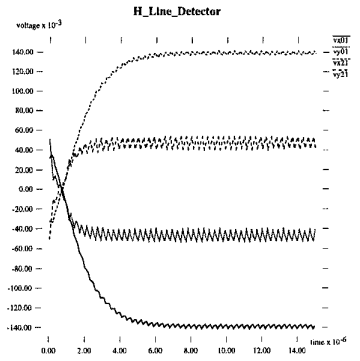


Figure 4. Trajectories of state nodes V_x and outputs V_y of cells $C_{0,1}$ and $C_{2,1}$

The waveforms exhibited some noise ripples which are typical of the time-multiplexing method.

$$I_{in} = \begin{pmatrix} -1.0 & 0.4 & -0.8 & -1.0 \\ -0.4 & -1.0 & -0.8 & -0.6 \\ 0.8 & -0.4 & 0.8 & 1.0 \\ -0.8 & -0.6 & -0.8 & -1.0 \end{pmatrix} \quad (8)$$

$$OUT = \begin{pmatrix} -1.0 & -1.0 & -1.0 & -1.0 \\ -1.0 & -1.0 & -1.0 & -1.0 \\ 1.0 & 1.0 & 1.0 & 1.0 \\ -1.0 & -1.0 & -1.0 & -1.0 \end{pmatrix} \quad (9)$$

$$A = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 1.0 & 2.0 & 1.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \quad (10)$$

4.3: Edge detection

We performed two simulation tests, first with the usual template conditions of equations(3)-(5), and second with equations(4) and (5) being lumped together.

4.3.1 Usual template conditions

A complete 8 x 8 array TMCNN was simulated. The multiplexing pulse width T was 120 ns. The conditions of equations (3)-(5) were used to successfully perform edge detection (see fig. 1). It is of interest to note that only 1 pass of each coefficient was needed to reach the steady state response.

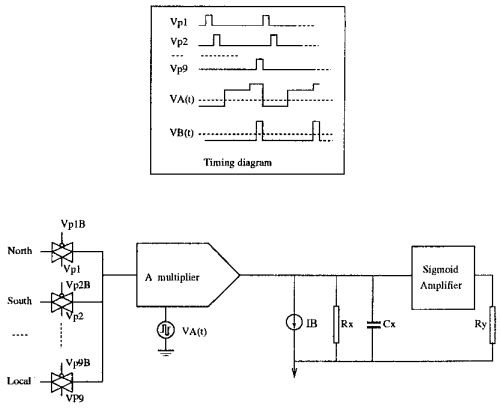


Figure 5. Simplified circuit architecture of time multiplexed CNN. IB is the substitute current source in place of B-template multipliers and the original bias Ib.

4.3.2 Lumped circuit conditions

It is noted that the input sources and the template coefficients are invariant with time during the processing period. The terms of $(\sum_{m=1}^{9} b^m u^m + I/M)$ add up to a constant sum. Hence these terms can be pre-computed and programmed into the circuit as a fixed bias (IB). Note that the original bias term Ib has been included here.

A 6 x 6 TMCNN array was simulated using the lumped circuit conditions and the result was successful. A typical threshold bias expression θ computed off chip for a 6 x 6 array has the form shown below (eqn. 11).

$$\theta = \begin{vmatrix} -2.95 & -2.95 & -3.45 & -3.45 & -2.95 & -2.95 \\ -2.95 & 1.3 & 0.3 & 0.3 & 1.3 & -2.95 \\ -3.45 & 0.3 & -1.2 & -1.2 & 0.3 & -3.45 \\ -3.45 & 0.3 & -1.2 & -1.2 & 0.3 & -3.45 \\ -2.95 & 1.3 & 0.3 & 0.3 & 1.3 & -2.95 \\ -2.95 & -2.95 & -3.45 & -3.45 & -2.95 & -2.95 \end{vmatrix} \quad (11)$$

In this circuit mode, all the B-template multipliers have been dispensed with and replaced with simple dynamic current mirrors. This mode of operation achieves a further reduction in chip area (at the expense of additional off-chip computation).

The simplified architecture is shown in fig. 5.

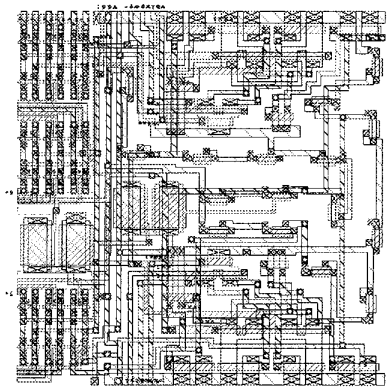


Figure 6. Layout design of the TMCNN building block cell

Author	Templates	Technology	Cell Size	Array Size
Cruz et al. 1992 [2]	Fixed	2 μm	0.032 mm^2	6 x 6
Halonen et al. 1992 [4]	Programmable	2 μm	1.0 mm^2	4 x 4
Espejo et al. 1994 [3]	Fixed	1.6 μm	0.015 mm^2	16 x 16
Kinget et al. 1995 [5]	Programmable	2.4 μm	0.26 mm^2	4 x 4
TMCNN	*Programmable*	*2 μm*	*0.065 mm^2*	*test cell*

Table 1. Recent Analog VLSI CNN Implementation

5: VLSI layout design of the TMCNN building block cell

We performed the layout of a cell to implement the TMCNN circuit architecture of Fig. 5. We made use of linear MOS resistor and MOS gate capacitors described in section 3. The active resistor is of 1.6MΩ and has a high linearity throughout the range of signal swing in our circuit. The MOS gate capacitance has a value of about 0.5pf. From our simulation result, the average cell current is about 100nA at a supply voltage of $\pm$2.5V. The test chip will be sent out for fabrication soon and we expect to deliver test results at the Conference.

Our cell occupies an area of 0.065 mm^2 which is about a quarter of the size of the smallest reported programmable CNN cell (see Table 1).

6: Delayed circuit response of the time multiplexing scheme

The settling time τ to reach steady state response is inherently longer for a TMCNN than a SCNN. This is due to the sequential operation of the time multiplexing scheme. In our simulation result of 4.3.1, τ of SCNN is 120 ns and that of TMCNN 1.1 μs. Our result showed that τ of a TMCNN was about M times that of a SCNN when the neighbourhood size is 1 (M is 9 in this case.) Larger neighbourhood size and the greater number of non-zero template coefficients will increase τ. The multiplexing scheme is a function of these two factors since the active periods of different template coefficients are the same and will share the time within a cycle of the multiplexing pulses. The scheme becomes complicated for CNNs of multiple layers. However most image processing problems can be solved with single layer CNN and the maximum value of M is 9. Also, the extra processing time of TMCNN may not be a problem since the processing bottle-neck may lie with the throughput of the input and output processing.

7: Conclusions

The time-multiplexed CNN was proposed and the operation verified through mathematical modelling and SPICE simulations. We have designed a building block cell to implement the TMCNN. We expect to deliver test results of the cell by the time of the Conference. A fully programmable CNN which is implemented through this simple time multiplexing scheme required one quarter of the chip area of a conventional SCNN, making it feasible to implement single chip VLSI CNNs with much higher densities than previous approaches.

References

[1] L. Chua et al. Cellular neural network: theory. *IEEE Transactions on Circuits and Systems*, 35(10):1257–1272, 1988.

[2] J. Cruz et al. Design of high-speed, high-density cnns in cmos technology. *International Journal of Circuit Theory and Applications*, 20:555–572, 1992.

[3] S. Espejo et al. Switched-current techniques for image processing cellular neural network in mos vlsi. *IEEE International Symposium on Circuits and Systems*, pages 1537–1540, 1992.

[4] K. Halonen et al. Programmable analogue vlsi cnn chip with local digital logic. *International Journal of Circuit Theory and Applications*, 20:573–582, 1992.

[5] P. Kinget et al. A programmable analog cellular neural network cmos chip for high speed image processing. *IEEE Journal of Solid-State Circuits*, 30(3):235–243, March 1995.

[6] K. K. Lai et al. An area efficient implementation of a cellular neural network. *To be published in The Second New Zealand International Two-Sream Conference on Artificial Neural Networks and Expert Systems - ANNES' 95*, November 1995.

[7] D. Lim et al. A programmable, modular cnn cell. *IEEE International Workshop on Cellular Neural Networks and their Applications*, pages 79–84, 1994.

[8] C. Mead. *Analog VLSI and Neural Systems*. Addison-Wesley Publishing Company, 1989.

[9] J. Nossek et al. Cellular neural networks: Theory and circuit design. *International Journal of Circuit Theory and Applications*, 20:533–554, 1992.

[10] Z. Wang. A cmos four-quadrant analog multiplier with single-ended voltage output and improved temperature performance. *IEEE Journal of Solid-State Circuits*, 26(9):1293–1301, September 1991.