An Area Efficient Implementation of a Cellular Neural Network

K.K. Lai and P.H.W. Leong
Department of Electrical Engineering
Sydney University Australia

Abstract

A time multiplexing scheme for implementing cellular neural networks (CNN) is described. This scheme makes it possible to realise much higher density implementations of CNNs in VLSI circuits. A circuit implementation of this technique is presented along with simulation results.

Introduction

Cellular neural networks are generalised, locally connected architecture which are particularly suitable for image processing. The circuit model equation in [1] can be written as (assume single layer and unity neighbourhood size):

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

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 a standard CNN (SCNN), a programmable basic cell would require 18 multipliers (9 each for a_m and b_m 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 [2]. Another programmable CNN chip by Lim et al. had an area of 0.4 mm^2 per cell [4]. A recent chip by Kinget et al. [3] had a cell size of 0.26 mm^2. 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 a large net saving in the area required to implement a fully programmable CNN.

Mathematical Model of TMCNN and Software Simulation Results

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

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

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 equals 9 in the present discussion. The maximum value of M is 9 for a single layer CNN with unity neigbourhood size. Contributory components in the same direction m for all cells are obtained by performing summation over a periodic pulse of width T through small incremental time steps.

The differential equation in eqn. 2 can be solved using Euler's method. A C-program was written to solve eqn. 2. The software was applied to perform edge detection with the conditions in equations 3-5.

$$\begin{aligned} A &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, & (3) \\ B &= \begin{pmatrix} -0.25 & -0.25 & -0.25 \\ -0.25 & 2.0 & -0.25 \\ -0.25 & -0.25 & -0.25 \end{pmatrix}, & (4) \\ I &= 0.2. & (5) \end{aligned}$$

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

0-8186-7174-2/95 $04.00 © 1995 IEEE

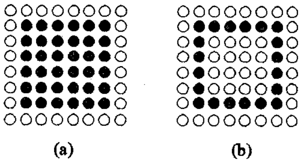

Figure 1: Edge detection of a square pattern of array size 8 x 8. A dot denotes a black pixel and a circle a white pixel. (a) Original pattern. (b) Final pattern after detection.

Circuit Architecture of TMCNN

The circuit architecture of the basic cell is shown in fig. 2. The time multiplexing scheme consists in replacing some multipliers with transmission gates (tgates). Only 2 time multiplexed core multipliers are used for the A and B templates. The timing diagram of the drivng pulses for the transmission gates and the gain control voltage waveforms for the time multiplexed multipliers (A and B multiplier) is 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 inputted to the A and B multipliers respectively (as shown in fig. 2). The timing sequence of the drivng pulses will 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. Since 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.

The Multiplier Circuit

The multiplier is a Gilbert four quadrant multiplier circuit as described by Mead [5]. The gain and sign of the multipliers are determined by the magnitude and sign of a_m and b_m values. These are globally programmed by the common $V_A(t)$ and $V_B(t)$ voltage sources.

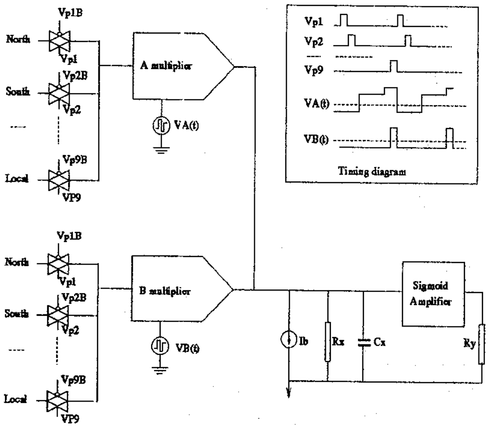

Figure 2: The architecture of time multiplexed cellular neural network. The inset shows the timing diagram of the driving pulses and the multipliers gain control voltage waveforms $V_A(t)$ and $V_B(t)$. In general, $V_A(t)$ and $V_B(t)$ are multi-level voltage waveforms.

The Active Resistor

We make use of a MOS linear resistor as the active resistor. The configuration is similar to that of Wang [6]. 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 separte 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:

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

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

Spice Simulation Results

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:

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

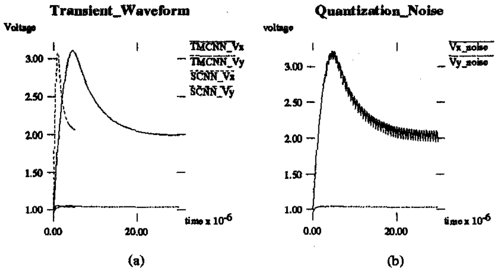

Figure 3: The transient voltage waveform of a cell circuit C(2,2) in a 4 x 4 array of a TMCNN and SCNN. (a) The transient response of the TMCNN is delayed as shown. (b) Noise ripples are associated with the time multiplexing scheme when the multiplexing pulses do not have sufficiently short pulse width.

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 quantization noise ripples disappeared when the multiplexing pulse width T was small enough (≤ 40 ns, in the present case).

Edge Detection

A complete 8 x 8 array TMCNN was simulated using SPICE. Four-quadrant Gilbert multipliers were used for the template multipliers, and minimum geometry transmission gates were employed to perform the multiplexing. The technological parameters were those of Orbit Semiconductor's 1.2 μm double metal, double polysilicon, n-well CMOS technology. The multiplexing pulse width T was 120 ns. The simulation time for a Sun Sparc 10 workstation took about 5.2 hours. 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.

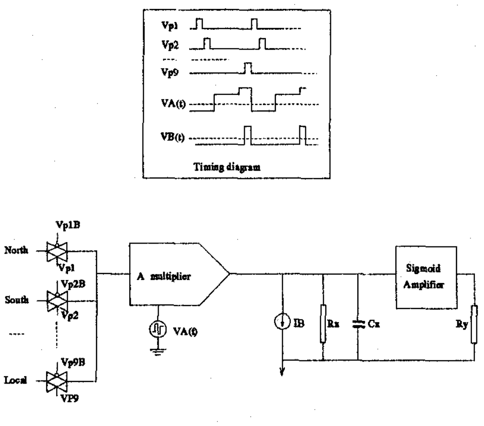

Figure 4: Simplified circuit architecture of time multiplexed CNN. IB is the substitute current source in place of B-template multipliers and the original bias Ib.

Limitation of the Time Multiplexing scheme

The settling time τ to reach steady state response is longer for a TMCNN than a SCNN because of the time multiplexing scheme. In our simulion result of Edge Detection, 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 cylce of the multiplexing pulses. The scheme becomes complicated for CNNs of multiple layers. However it is noted that most image processing problems can be solved with single layer CNN and the maximum value of M is 9.

Further Reduction in Area

It is noted that the terms of $(b_m V_{uij}^m + I/M)$ in eqn. 2 is a constant sum since the input sources and the template coefficients are invariant with time. Hence these terms can be pre-computed and programmed into the circuit as a fixed bias (IB). The original bias term Ib can also be lumped here. All the B-template multipliers can be dispensed with and replaced with DAC's or simple dynamic current mirrors, resulting in

Table 1: Comparison of hardware requirement in SCNN and TMCNN.

Type of CNN		*Hardware requirement*
SCNN		18 multipliers
TMCNN	Scheme AB	2 multipliers and 8 tgates
	Scheme A	1 multiplier, 9 tgates and 1 dynamic current mirror

further reduction of area (at the expense of additional off-chip computation). The simplified architecture is shown in fig. 4. Table 1 lists the comparison of neural processing hardware requirement in the SCNN and our proposed schemes of TMCNN. Scheme "AB" employs A- and B-template multipliers and Scheme "A" A- template multiplier only.

Conclusions

We proposed a simple time multiplexing method which significantly reduces the area of a fully programmable cellular neural network. The scheme was verified through mathematical modelling and SPICE simulations, and it was successfully applied to the problem of edge detection. The greatly reduced area offered by the time multiplexing scheme will make it feasible to implement single chip VLSI CNNs with much higher densities than previous approaches.

References

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

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

[3] 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.

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

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

[6] 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.