Very Fast Programmable CNN Based on FG-Inverter

Jesus E. Molinar-Solis
Mexico-State Autonomous University
(UAEM-UAP Ecatepec), Electronics Laboratory
Ecatepec de Morelos, Estado de Mexico, MEXICO
Email: jmolinar@sees.cinvestav.mx

Felipe Gomez-Castaneda, Jose A. Moreno-Cadenas
and Victor H. Ponce-Ponce
Center for Research and Advanced Studies (CINVESTAV-IPN), Electrical Engineering Department
*(UPIITA-IPN)
Mexico City, MEXICO

Abstract— In this paper, a different CNN implementation based on floating-gate inverters is presented. The methodology adjusts the original CNN model in order to incorporate a floating-gate inverter and reduce hardware complexity. The approach brings a fast response compatible with standard digital circuits. Eight different processing tasks were tested on a 4x4 CNN prototype fabricated on AMI 1.2 micron through MOSIS.

I. INTRODUCTION

The Cellular Neural Network (CNN), is a powerful architecture with a high parallel computing capability to perform image processing tasks in real-time. Recent CNN implementations have demonstrated a high computational power in comparison with conventional image processing techniques.

From its inception, many VLSI designers have developed CNN chips with different properties, and there is a constant search in order to improve their electrical features such as power consumption, settling time and cell density. In this work a different hardware methodology is proposed, instead of the typical technique based on the current-mode representation of analogue variables for the arithmetic operations [1], a floating-gate CMOS (FG-CMOS) inverter is used allowing the sum operation in voltage mode and reducing static power consumption.

Since the piece-wise linear function in the original CNN model can be replaced by a hard-limiting (sign-type) output function for those tasks with a local regularity condition [2], the same FG-CMOS inverter output can be used for this purpose. These two facts give the idea to use a FG-CMOS inverter to execute the addition and the output function of the CNN equations at the same device.

II. ORIGINAL CNN MODEL

Following the CNN concept by L.O. Chua and L. Yang [3], there is a state equation for each cell in the network, given by:

\[
\frac{dx_i}{dt} = -x_i + \sum_{k} A_{i,k} y_k + \sum_{k} B_{i,k} \phi_G + z_i
\]

There is also an output function, which is expressed by:

\[
y_i = f(x_i) = \frac{1}{2} (|x_i| + 1) - |x_i - 1|
\]

Where the variables \(u, x, y\) are the input, internal state and output signals of the reference cell \(C_{r}\), respectively. A, B and \(z\) are the feedback template, the control template and the threshold, respectively. The notation \(C(k,l) \in N_{r(i,j)}\) includes all the cells in the neighbourhood \(N_{r(i,j)}\) of cell \(C_{r}\).

III. FLOATING GATE CMOS INVERTER

A floating-gate inverter is a typical CMOS inverter with two or more input capacitances coupled to the floating gate (FG), which is common to both N and P channel transistors. The potential induced in the FG can be controlled as a weighted sum, in voltage-mode, of all input signals. The potential of the FG, which is common to the complementary MOSFET transistors, establishes the on-off state of the CMOS inverter [4, 5]. In contrast to the common current wired summation, the coupled capacitive inputs to the FG do not dissipate static power.

The clocked NeuMOS concept is a particular case where a switch transistor not only allows to discharge the FG, but gives the possibility to pre-charge the FG to a given potential [6, 7] as well. The FG potential \(\phi_G\) with \(n\) inputs can be calculated using the following expression:

\[
\phi_G(t) = \phi_G(0) + \sum_{i=0}^{n} C_i V_i(t) - \sum_{i=0}^{n} C_i V_i(0)
\]

Where \(\phi_G(0)\) and \(V_i(0)\) are the FG potential and \(i\)-th input voltage respectively in the reset period. And \(\phi_G(t), V_i(t)\) are the same signals but at the evaluation period. The potential...
\( f_{FG} (0) \) may be zero, but is useful for applications where an
offset in the FG is necessary, avoiding the use of an extra
input capacitance. Fig. 1 is an example of a FG-CMOS
inverter with \( n \) input capacitances. Due to leakage currents,
the main drawback of the "clocked" concept is that a
periodical refreshing signal is necessary. In this work, we
use the clocked NeuMOS concept to develop a compact
CNN cell.

![Circuit diagram of \( n \) input Clockented NeuMOS Inverter.](image)

### IV. CNN Model For VLSI

In the case of binary image processing by a CNN, it has
demonstrated that the typical piecewise-linear (PWL)
function (2), can be replaced by a hard-limiting or sign-type
(sgn) output function for those tasks with locally regular
(LR) condition [2]. This exclusion of the output linear
region allows simplifying the CNN dynamics. Now, the
state equation can be expressed as:

\[
\frac{dx}{dt} = -x + a_{0,0} y + k_i
\]

Where \( a_{0,0} \) is the template central element and,

\[
k_i = \sum_{(k,j) \in C} A(k,j) y_u + \sum_{(k,j) \in C} B(k,j) y_u + z_q
\]

The notation \( N_k(i,j) / C \) represent all the neighbourhood
cells, except the cell \( C \) itself. Since the output for a sign-
type function will always be saturated, \( y = \pm 1 \), the solution
of (4) is in the following form:

\[
x_y(t) = k_i \pm a_{0,0} + C e^{-t}
\]

Where \( C \) is a constant.

An LR task implies "consistency" [2], this condition
makes possible to consider the local neighbors of \( C \) cell
with a constant value, therefore, as \( t \to \infty \) the state will
decrease exponentially to the value \( k_i \pm a_{0,0} \). This suggests
that the element \( a_{0,0} \) may not be useful for those tasks where
the actual cell output is not relevant, and this fact will not
affect the system dynamics. The stability condition \( a_{0,0} \to 0 \)
will be necessary only for those tasks that make use of such
element. From (6), for the (black) output, \( y = \pm 1 \), the
condition \( k_i \pm a_{0,0} \to 0 \) must hold, and similarly for the (white)
output \( y = -1, k_i \pm a_{0,0} \to 0 \). Following a similar notation used
in [4], for the case when \( t \to \infty \), we can define a template
vector \( p \) containing \( m \) elements of matrices \( A, B \) and \( z \).
Similarly, a vector \( q = \{-1,1\} \) can be defined for all the
neighbouring cells inputs \( u \), and outputs \( y \), respectively.
The state for this case can easily be found as:

\[
x_y = pq^T
\]

And the output, is expressed by:

\[
y_y = \text{sgn}(pq^T)
\]

These equations, suggest that for the (black) output \( y_0 =
+1 \), the condition \( pq \geq 0 \) must be fulfilled and for the
(white) output \( y = -1, pq < 0 \) is required. One important
feature related to the lack of a linear output region is that the
analogue multipliers commonly required for the original
CNN model can be replaced by simple analogue
multipliers. This fact makes possible a considerable transistor
count reduction per cell. In the original model, the
input \( u, e \{-1,1\} \), and the output \( y, e \{-1,1\} \). The template
elements of matrices \( A, B \) and \( z \) belong to real numbers.
Therefore, the corresponding products of such matrix
elements and the inputs or outputs, i.e., \( a_{0,0} y \) and \( b_{i,j} y_{i,j} \)
require a four quadrant multiplier. In a binary CNN with a
sgn function output, the input and the output can only take
one of two possible values, \( u, e \{-1,1\} \), and \( y, e \{-1,1\} \).
The case for which a simple analogue multiplexer is used
instead of a multiplier, the input and output cell convention
can be redefined for an easier mathematical representation.
The vector \( u = \{1 \ 0 \} \) will represent the (black) input, \( u = -1 \)
and, \( y, e \{-1,1\} \) the (black) output, \( y, e \{-1,1\} \). The electrical
representation of this idea is shown in Fig. 2, where the
output vector makes turn on one transistor and makes the
other cut-off, therefore, the output corresponds to an
associated value. Similarly, a vector \( u_{\text{black}} \) \( y, \{-1,1\} \) will
represent a white "-1" input, output.

![Arithmetic product using NMOS switches](image)

---

3727
With this representation, each template element, e.g. \( a \), must be transformed to a vector representation such as \( \mathbf{a} \). Therefore, the vector \( \mathbf{p} \) changes to a matrix: \( \mathbf{P} \). In this way, we can rewrite the CNN equations in a matrix notation, where analogous to \( \mathbf{P} \), the matrix \( \mathbf{Q} \) contains one of all possible constellations [2]. For the state, (7) this is:

\[
x_y = \mathbf{P} \cdot \mathbf{Q}^T = \mathbf{a}_{ni,j+1} \mathbf{a}_{nj,j} \cdots \mathbf{b}_{k,j} \cdots \mathbf{z}_n
\]

(9)

And for the new output function:

\[
y_y = \begin{bmatrix} y_{y1} & y_{y2} \end{bmatrix}
\]

(10)

Where the \( \text{sgn} \) function is replaced by:

\[
y_y = \text{step}(x_y) = \begin{cases} 1, & x_y \geq 0 \\ 0, & x_y < 0 \end{cases}
\]

(11)

\[ A. \quad \text{CNN Model with positive values} \]

The obtained CNN equations should be adapted to the electrical features of a floating-gate inverter. First, the inverter transfer function resembles to a hard-limiting output with a transition point, shifted to a positive value commonly at \( \frac{1}{2}V_{DD} \). Second, it is desirable to use a single voltage supply for the whole integrated circuit. This implies the use of only positive voltage signals in the system. Following a similar approach [8], it is possible to obtain a positive numerical solution for a given processing task, a detailed example of the followed methodology is shown in [9]. Therefore (9) is still valid but \( \mathbf{P} \) and \( \mathbf{Q} \) elements are all positive. The output function (11) is modified as:

\[
y_y = \text{step}(\mathbf{P} \cdot \mathbf{Q}^T \cdot t_p)
\]

(12)

Where the element \( t_p \) is the FG-CMOS inverter transition point, commonly at \( \frac{1}{2}V_{DD} \). As an example, the template used for the "Shadow Detector" is shown below.

\[
\begin{bmatrix}
A & B
\end{bmatrix} = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

and \( z \)

The matrix \( \mathbf{P} \) generated from this template:

\[
\mathbf{P} = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

The following notation in [2], we can define a matrix \( \mathbf{K} = [Q_1, Q_2, Q_3, Q_4] \) including all the possible constellations. For the "Shadow Detector" task [10], with \( t_p = \frac{1}{2}V_{DD} \), the following inequalities must be fulfilled:

\[
\mathbf{P} \cdot Q_1 > t_p, \quad \mathbf{P} \cdot Q_2 > t_p, \quad \mathbf{P} \cdot Q_3 > t_p, \quad \mathbf{P} \cdot Q_4 > t_p
\]

(13)

V. \quad A 4x4 CNN ARRAY USING A FG-CMOS INVERTER

An important observation relative to CNN templates, is that most binary tasks only use one template, \( A \) or \( B \), rarely both of them. This fact; leads to an architecture with only one template including a switching capacity to behave like an \( A \) or \( B \) template. Equation (9) is implemented through transmission gates as analogue multiplexers and the addition through the capacitive coupled inputs to the FG-CMOS inverter. Equation (10) is implemented by means of the FG-CMOS inverter transfer function and an auxiliary conventional inverter, in this case, the transistors width relation is \( w_p/w_N \approx 3 \), corresponding to a transition point near to \( \frac{1}{2}V_{DD} \). The electrical diagram of a CNN cell is shown in Fig. 3. The key element, is the floating-gate inverter with six input capacitors with a floating gate gain factor, \( y = 0.89 \). Four input capacitors are connected to four neighbours, \( \text{(north, south, east and west)} \) one for the central template element and another for the threshold \( z \). Through a logic level in the terminal \( Wh \), the central template element is chosen corresponding to an \( A \) or \( B \) template. Similarly, through the terminal \( B/A \), the non-central template elements are defined. A microphotograph of the 4x4 CNN array is shown in Fig. 4. An P+P photodiode, with an amplifier was included on each cell, allowing the introduction of input signals optically. An output buffer was also considered in order to overcome the charge effects of the oscilloscope probe.

![Figure 3 Electrical cell diagram.](image-url)
VI. CNN Prototype Measurements

Eight different processing tasks were tested on the CNN prototype, these are: Shadow Detector, Hole Filling, Edge Extraction, Noise Removal, Erosion, Dilatation, Global Component Detection and Connected Component Detection. The input patterns were projected on using a microscope. For a given measurement, clock signal R1 sets the initial condition and clock R2 turns on the CNN array. For the Shadow Detection task, Fig. 5, shows the input, output patterns, and the transition of a cell to a “black” output (logic high level). Fig. 6 shows for this task the time response between cells.

![Figure 4. CNN chip microphotograph.](image)

![Figure 5. Cell transition in the "Shadow Detection" task](image)

![Figure 6. Response between cells for the "Shadow Detection".](image)

VII. Conclusion

A low complexity CNN implementation through the use of floating-gate inverters, was presented. The CNN array provides a good settling time, around 50ns, as compared with other binary CNN networks [10, 11]. The power consumption, ~700μW, is comparable to other designs with a similar technology [12]. The cell dimension is approximately 162x227μm without global wiring.

REFERENCES


