A NOVEL ARCHITECTURE OF CELLULAR NEURAL NETWORK USING DSP CORE

Dharavath Madhavi[1] **V. Lavanya** [2] **P. Sharmila Rani**[3]
dharavathmadhavi@gmail.com[1] lavanya.veerella@gmail.com[2] sharmilakanchi@gmail.com[3]
[1]PG Scholar,VLSI, Teegala Krishna Reddy Engineering College, Hyderabad.
[2]Assistant Professor, Dept of ECE, Teegala Krishna Reddy Engineering College, Hyderabad.
[3]Associate Professor, Dept of ECE, Teegala Krishna Reddy Engineering College, Hyderabad.

Abstract: Emulations of cellular nonlinear networks on digital reconfigurable hardware are renowned for an efficient computation of massive data, exceeding the accuracy and flexibility of full-custom designs. In this contribution, a digital implementation with polynomial coupling weight functions is proposed for the first time, establishing novel fields of application, e.g., in the medical signal processing and in the solution of partial differential equations. We present an architecture that is capable of processing large-scale networks with a high degree of parallelism, implemented on state-of-the-art field-programmable gate arrays.

Index Terms— Cellular neural networks, field-programmable gate arrays (FPGAs), image processing, partial differential equations (PDEs), system-on-chip.

I. INTRODUCTION

CELLULAR neural networks (CNN) are nonlinear, continuous computing-array structures well suited for nonlinear signal processing. Due to their structural simplicity, CNNs are ideal for VLSI (very large scale integration) implementation. The CNN Universal Machine architecture transforms the CNN array into a stored-program computer. The first CNNUM (Cellular Neural network Universal machine) chip has 32 x 32 cells in standard CMOS technology, and its performance is close to logarithmic operations per second. Solving simulating partial differential equations (PDE's) and systems of locally interconnected ordinary differential equations (ODE's) are two new challenging areas of application for CNN.

The cellular nonlinear networks (CNNs) have proved to be suitable for the image processing, medical signal processing, robot control, and solution of partial differential equations (PDEs), among others. Analog and mixed-signal implementations of the CNN universal machine provide the exceptional computational performance of thousands of processing units on a single chip. However, the precision of analog implementations is usually not sufficient for numerically sophisticated applications.

Thus, the emulation of CNNs on reconfigurable digital devices, especially on field-programmable gate arrays (FPGAs), becomes attractive for prototyping and applications where flexibility and/or higher precision is required. It has been shown that networks with nonlinear couplings are inevitable for numerous biologically motivated applications and especially for solving PDEs.

Polynomial cellular neural networks

Cellular Neural Networks (CNNs) are complex nonlinear dynamical systems. CNN is simply an analogue dynamic processor array, made of cells, which contain linear capacitors, linear resistors, linear and nonlinear controlled sources. Let us consider a two-dimensional grid with 3 ×3 neighborhood system as it is shown on Fig. 1.

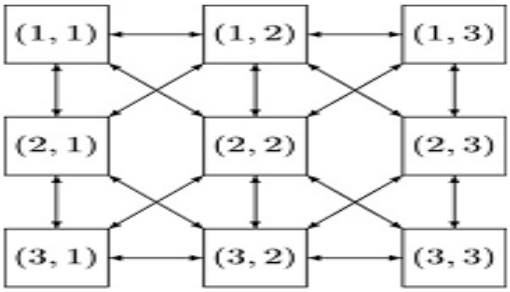


Fig. 1.One-layer two-dimensional CNN.

The squares are the circuit units-cells, and the links between the cells indicate that there are interactions between linked cells. One of the key features of a CNN is that the individual cells are nonlinear dynamical systems, but that the coupling between them is linear. Roughly speaking, one could say that these arrays are nonlinear but have a linear spatial structure, which makes the use of techniques for their investigation common in engineering or physics attractive. We will give the general definition of a CNN which follows the original one.

$X(t) = \widetilde{L1}\ \widetilde{F1}$(x,n,t,u),

$X(t) = \widetilde{L2}\ \widetilde{F2}$(x,n,t,u)

It is easy to see that the operators $\widetilde{L}_1$ and $\widetilde{L}_2$ are linear, bounded and therefore continuous.

II. LITERATURE REVIEW

Cellular Neural Networks With Non-Linear And Delay-Type Template Elements And Non-Uniform Grids

The cellular neural network (CNN) paradigm is a powerful framework for analogue non-linear processing arrays placed on a regular grid. In this paper we extend the current repertoire of CNN cloning template elements (atoms) by introducing additional non-linear and delay-type characteristics. In addition, architectures with non-uniform processors and neighbor hoods (grid sizes) are introduced. With this generalization, several well-known and powerful analogue array-computing structures can be interpreted as special cases of the CNN. Moreover, we show that the CNN with these generalized cloning templates has a general programmable circuit structure (a prototype machine) with analogue macros and algorithms. The relations with the cellular automaton (CA) and the systolic array (SA) are analyzed. Finally, some robust stability results and the state space structure of the dynamics are presented.

A CNN-Based Chip for Robot Locomotion Control

In this paper, the paradigm of emergent computation is applied to locomotion control in legged robots: the locomotion gait is the result of self-organization of a network of locally coupled nonlinear oscillators. This means to adopt the biological paradigm of central pattern generator (CPG), implemented by using cellular neural networks (CNNs). The whole control strategy is hybrid in the sense that the gait generation is accomplished by a fully analog CNN, while a simple logic unit modulates the behavior of the CNN-based CPG, so that the strategy is suitable to eventually include sensory feedback. The design of a VLSI chip implementing the CNN-based CPG and some experimental results on the chip are presented. The chip is designed using a switched-capacitor technique, fundamental to obtain in a simple and direct way some key features of the hybrid control discussed. The experimental results confirm the suitability of the approach.

III. PROPOSED SYSTEM

PTCNN MODEL

A CNN is a regular system of processing elements (PEs) (cells) that are coupled to its neighbors in lateral and diagonal directions In a common 1-neighborhood each cell is hence coupled to eight neighbors and to itself (also called 3 ×3 neighborhood). In a PTCNN, the couplings between the neighboring cells are represented by polynomial weight functions. Since a standard model has not yet been defined, we exclusively refer to feedback and feed forward terms of polynomial order P∈N, which leads to a cell state defined by the differential equation

$$x_{ij}(t) = -x_{ij}(t) \dot{+} \sum_{\substack{|k|<1 \\ |l|<1}} \sum_{p=1}^{p} akl^{p} . yi + k, j + l^{p}(t) + \sum_{\substack{|k|<1 \\ |l|<1}} \sum_{p=1}^{p} bkl^{p} . ui + k, j + l^{p} + z \ldots\ldots(1)$$

To avoid confusion,y^p denotes the pth power of y, whereas in $a^{(p)}$, p is an index. Using the forward Euler

integration method, we obtain the discrete time state equation of a PTCNN

$$xij(n+1) = N\left(xij(n) + \sum_{\substack{|k|<1 \\ |l|<1}} \sum_{p=1}^{p} \hat{akl}(p) x_{i+k,j+l}^{p}(n) + \hat{w}ij\right) \quad (2)$$

with the nonlinear output function

$$N(x) = \begin{cases} -1, x < -1 \\ x, -1 \le x \le 1 \\ 1, x > 1 \end{cases} \quad \ldots\ldots\ldots (3)$$

and with the modified weights

$$\hat{a}_{0,0}^{(1)} = h \cdot (a_{0,0}^{(1)} - 1);\ \hat{a}_{kl}^{(p)} = h \cdot a_{kl}^{(p)},$$

$$\forall (k,l,p) \neq (0,0,1), |k| \le 1, |l| \le 1, p \ge 1;\ \hat{b}_{kl}^{(p)} = h \cdot b_{kl}^{(p)},$$

$$\forall (k,l,p), |k| \le 1, |l| \le 1, p \ge 1 \text{ and } \hat{z} = h \cdot z.$$

The modified offset term of the PTCNN is given as

$$w^{ij} = \sum_{\substack{|k|<1 \\ |l|<1}} \sum_{p=1}^{p} \hat{}_{bkl}^{(p)} u_{i+k,j+l}^{p} + z\hat{} \quad \ldots (4)$$

replacing the lower term in (1). Applying the Horner scheme, the double sums in (2) can be rewritten as

$$\sum_{\substack{|k|<1 \\ |l|<1}} \sum_{p=1}^{p} {}_{bkl}^{p} u_{i+k,j+l}^{p}(n)$$

$$= \sum_{\substack{|k|<1 \\ |l|<1}} x_{i+k,j+l}(n).\left[{}_{akl}^{1} + x_{i+k,j+l}(n) + \cdots x_{i+k,j+l}(n)\right.$$

The double sum in (4) is written down similarly.

HARDWARE ARCHITECTURE

A. NERO Architecture

In this architecture, the network is divided into vertical segments that are treated line-by-line and segment-by-segment by a row of PEs. Each PE is connected to its left and right neighbors, or to a dummy PE at either ends of the row, respectively. Fig. 1 shows the structure of the 1-D processor array with the main control and data paths. For an optimal hardware utilization, two PEs share a common memory unit (mem) and a PE control unit. Both the PEs and the mems are coupled only locally to moderate the routing complexity. The left and right PE dummies (D) have no calculation core and are required only to provide data from the borders to the neighboring segments, or to ensure the network boundary conditions, respectively. The template values (weight coefficients) of the operation are stored in a template first-in-first-out that is designed as a ring buffer. The structure of the PEs, mems, and controlling modules are explained.

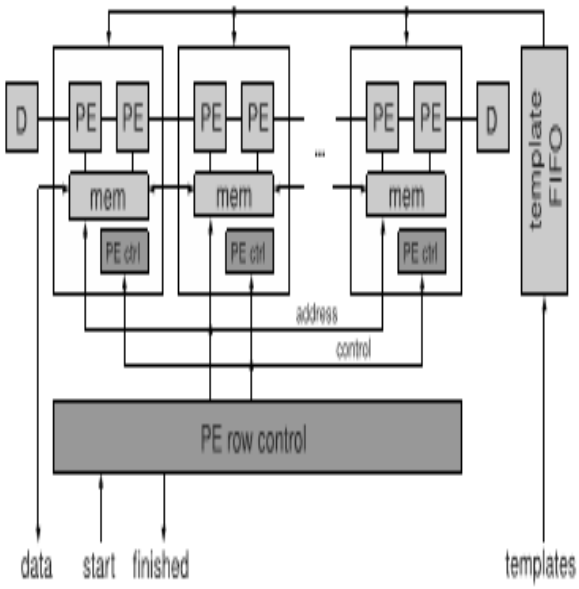


Fig.1. Array architecture with a row of locally coupled PEs and additional PE dummies (D)

B. TITUS

The modifications to NERO to process the PTCNN state equation (2) comprise the calculation core of the PE, the local controller unit, and the template FIFO.

The novel architecture of the calculation core will be explained subsequently. The utilization of the Horner scheme (5) exhibits some advantages over the conventional form of the PTCNN state equation (2)the powers of the state variables do not have to be stored explicitly in registers, and the interconnections within the PE can be retained. With an increasing polynomial order, each cell state computation easily requires dozens of MAC operations. Therefore, a further parallelization inside the PE becomes reasonable. . To avoid an unbalanced workload, the utilization of one, three, or nine DSP elements is favorable for the given 1-neighborhood. For the targeted FPGA platform, the usage of three DSPs rendered the best choice since the on-chip DSP slices feature a three-stage pipeline that can be utilized efficiently to store three intermediate results. This configuration allows each DSP to calculate the three polynomials in a pipeline with single clock cycle latency and without any interruption. The structure of the extended calculation core is shown in Fig. 2.

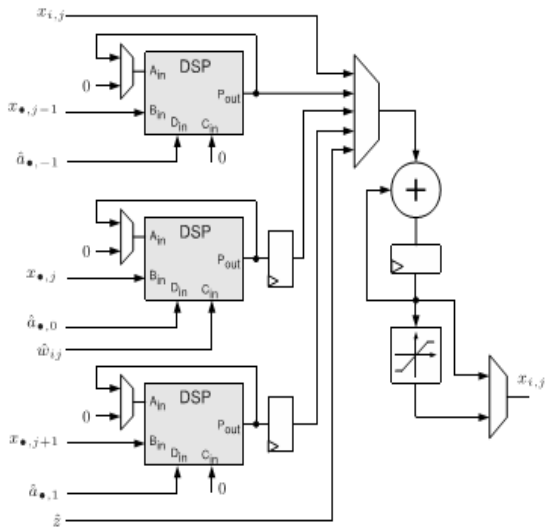


Fig. 2. TITUS calculation core for the emulation of a PTCNN the place-holder (•) refers to all rows of the neighborhood.

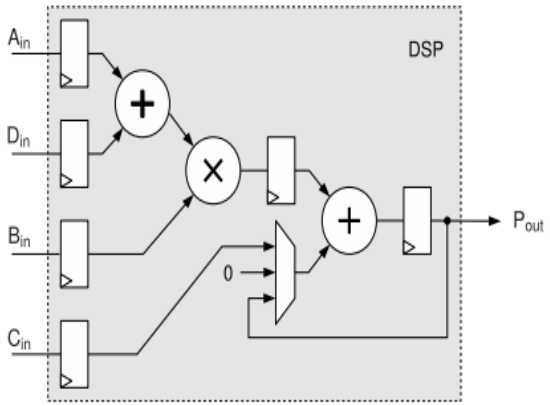


Fig. 3. Structure of the DSP core.

It comprises three DSPs for the left, central, and right column, each processing the contribution of three neighboring cells to the state equation. Each DSP (shown in Fig. 3) calculates three elements of the Horner scheme

$$m_{k+1} = x \cdot (a + m_k) \qquad (6)$$

withm0=0.

The number of iterations of (6) depends on the selected polynomial order. The results are accumulated, added toˆ wij, and finally constrained to the range[−1;1]with the operatorN.

MULTI-BIT FLIP-FLOPS

Multi-Bit Flip-Flops are capable of reducing the power consumption because they have shared inverter inside the flipflop. Clock skew is also minimized at the same at the same time. Sngle and multi-bit flip-flop have the same clock condition. set and reset condition is also same. the example of multi-bit flip-flops is shown in fig 1.1 . 2-bit flip-flop is formed by merging of single one bit flip flop. It share the clock buffer based and power reduction can be achieved.

Advantages of flip-flop:

1. Avoid the duplicate inverters.
2. Total area contributing to flip-flop can be reduced.

3. Power optimization through shared invertors.

ALGORITHM.

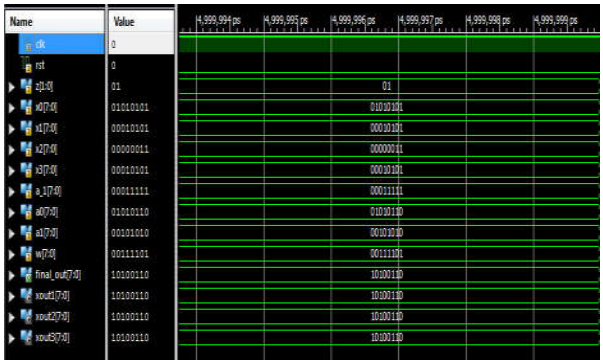
Algorithm is split into three steps as, First is to identify the merged flip-flops. In second we can build the combinational table according to the overlapped region in the first step. We can build the combinational table in binary tree representation for easy representation. In the third step, based on the combinational table, merging flip flops is formed.

VI. RESULTS

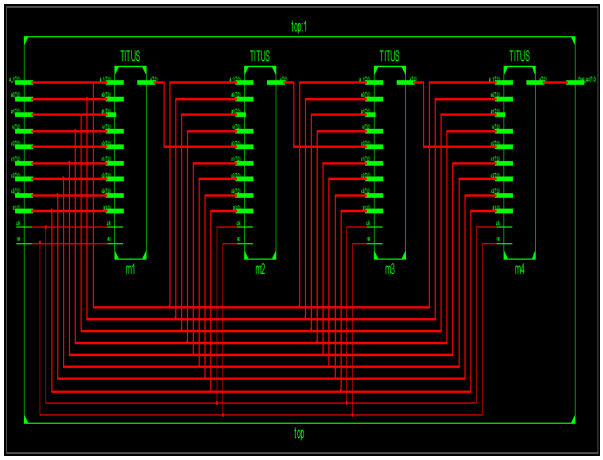
The Verilog HDL Modules have successfully simulate, verified and synthesized using Xilinxise13.2.

Existing Result:

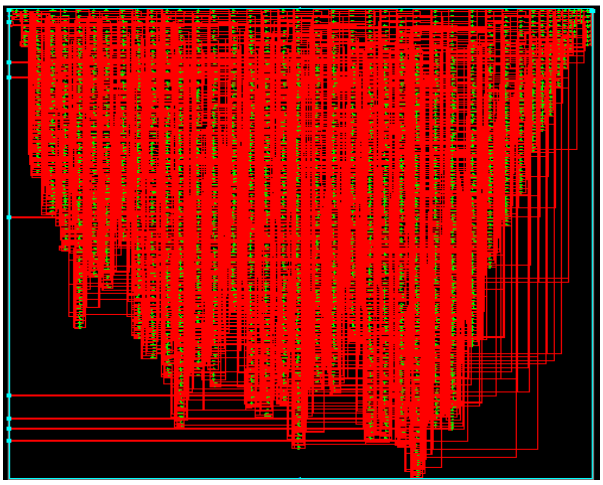
SIMULATION:



RTL SCHEMATIC:



TECHNOLOGY SCHEMATIC:



DESIGN SUMMARY:

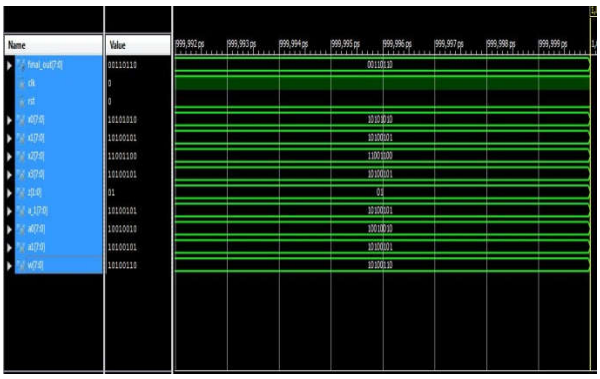
Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	766	4656	16%
Number of Slice Flip Flops	480	9312	5%
Number of 4 input LUTs	1421	9312	15%
Number of bonded IOBs	68	232	29%
Number of GCLKs	1	24	4%

TIMING REPORT:

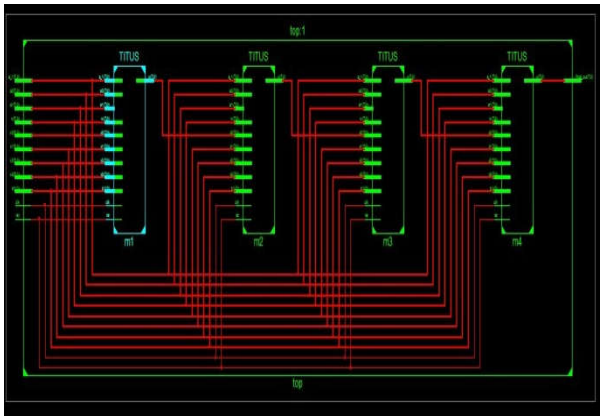
```
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
  Total number of paths / destination ports: 23 / 8
-------------------------------------------------------------------------
Offset:              5.501ns (Levels of Logic = 2)
  Source:            m4/m8/q_0 (FF)
  Destination:       final_out<7> (PAD)
  Source Clock:      clk rising

  Data Path: m4/m8/q_0 to final_out<7>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q             9   0.514   0.849  m4/m8/q_0 (m4/m8/q_0)
     LUT2:I0->O           1   0.612   0.357  m4/x<0>1 (final_out_0_OBUF)
     OBUF:I->O                3.169          final_out_0_OBUF (final_out<0>)
    ----------------------------------------
    Total                     5.501ns (4.295ns logic, 1.206ns route)
                                      (78.1% logic, 21.9% route)

=========================================================================
```

Proposed Result.

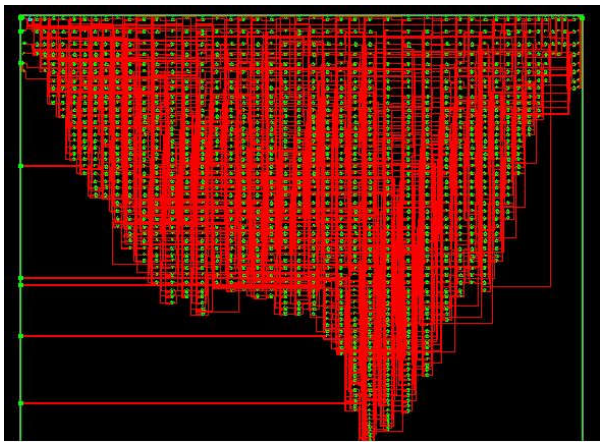
SIMULATION:



RTL SCHEMATIC:



TECHNOLOGY SCHEMATIC:



DESIGN SUMMARY:

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	360	4656	7%
Number of Slice Flip Flops	623	9312	6%
Number of 4 input LUTs	348	9312	3%
Number of bonded IOBs	54	232	23%
Number of GCLKs	1	24	4%

TIMING REPORT:

```
================================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
  Total number of paths / destination ports: 23 / 8
--------------------------------------------------------------------------------
Offset:              5.501ns (Levels of Logic = 2)
  Source:            m4/m8/m2/q_0 (FF)
  Destination:       final_out<7> (PAD)
  Source Clock:      clk rising

  Data Path: m4/m8/m2/q_0 to final_out<7>
                                Gate     Net
    Cell:in->out      fanout   Delay    Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q             9    0.514    0.849  m4/m8/m2/q_0 (m4/m8/m2/q_0)
     LUT2:I0->O           1    0.612    0.357  m4/x<0>1 (final_out_0_OBUF)
     OBUF:I->O                 3.169          final_out_0_OBUF (final_out<0>)
    ----------------------------------------
    Total                      5.501ns (4.295ns logic, 1.206ns route)
                                       (78.1% logic, 21.9% route)

================================================================================
```

VII.CONCLUSION

General purpose architecture for a digital emulation of CNNs with polynomial couplings has been presented. Implemented on a state-of-the-art FPGA, the system is capable of high-speed computation of PTCNN operations on large-scale networks. The proposed system is considered the first digital hardware implementation of a PTCNN so far. Applications for image processing and the simulation of PDEs have been discussed, some of which could not be realized in a CNN hardware before. We are currently implementing an extension of the polynomial weight architecture supporting on-chip optimizations of network parameters, and thus paving the way to a very efficient determination of problem-specific templates.

REFERENCES

[1] L. O. Chua and L. Yang, “Cellular neural networks: Theory,” IEEE Trans. Circuits Syst., vol. 35, no. 10, pp. 1257–1272, Oct. 1988.

[2] L. Nicolosi, F. Abt, A. Blug, A. Heider, R. Tetzlaff, and H. Höfler, “A novel spatter detection algorithm based on typical cellular neural network operations for laser beam welding processes,” Meas. Sci. Technol., vol. 23, no. 1, p. 015401, 2012.

[3] F. Gollas, C. Niederhöfer, and R. Tetzlaff, “Toward an autonomous platform for spatio-temporal EEG-signal analysis based on cellular nonlinear networks,” Int. J. Circuit Theory Appl., vol. 36, nos. 5–6, pp. 623–639, Sep. 2008.

[4] P. Arena, L. Fortuna, M. Frasca, and L. Patané, “A CNN-based chip for robot locomotion control,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 9, pp. 1862–1871, Sep. 2005.

[5] T. Roska, L. O. Chua, D. Wolf, T. Kozek, R. Tetzlaff, and F. Puffer, “Simulating nonlinear waves and partial differential equations via CNN. I. Basic techniques,” IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 42, no. 10, pp. 807–815, Oct. 1995.

[6] F. Puffer, R. Tetzlaff, and D. Wolf, “Modeling nonlinear systems with cellular neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), vol. 6. May 1996, pp. 3513–3516.

[7] T. Roska and L. O. Chua, “The CNN universal machine: An analogic array computer,” IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 40, no. 3, pp. 163–173, Mar. 1993.

[8] A. Rodriguez-Vazquez et al., “ACE16k: The third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 5, pp. 851–863, May 2004.