INTERNATIONAL JOURNAL OF CIRCUIT THEORY AND APPLICATIONS
Int. J. Circ. Theor. Appl. 2007; **35**:149–164
Published online 25 October 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cta.385

Function-in-layout: A demonstration with bio-inspired hyperacuity chip

András Mozsáry[1,*,†], Jen-Feng Chung[2] and Tamás Roska[1,3]

[1]*Jedlik Laboratories of the Faculty of Information Technology, Peter Pázmány Catholic University, Práter u. 50/a, Budapest H-1086, Hungary*
[2]*Department of Electrical and Control Engineering, National Chiao Tung University, Ta-Shueh road 1001, Hsinchu 300, Taiwan*
[3]*Computer and Automation Research Institute of the Hungarian Academy of Sciences, Kende u. 13-17, Budapest H-1111, Hungary*

SUMMARY

Below 100 nm a new scenario is emerging in VLSI design: floorplanning and function are inherently interrelated. Using mainly local connectivity, wire delay and crosstalk problems are eliminated. A new design methodology is proposed, called function-in-layout, that possesses: regular layout, mainly local connectivity, functional 'parasitics'. A bio-inspired demonstration is presented, a hyperacuity chip, with 30 ps time difference detection using 0.35 mm complementary metal-oxide semiconductor (CMOS) technology. Copyright © 2006 John Wiley & Sons, Ltd.

Received 22 June 2006; Revised 31 August 2006

KEY WORDS: function-in-layout; delay-domain computing; non-Boolean logic; hyperacuity; cellular neural network; cellular nonlinear network; cellular wave computer; time-to-digital converter

1. INTRODUCTION

Scaling down in silicon technology, as well as stored programmability of a prototype architecture with open interfaces, resulted in an unprecedented boom in electronics during the last 50 years. Moore's law for scaling down was the protagonist phenomenon until very recently. The roadmap for the development of integrated circuits, however, starts to show a slowing down. Below 180 nm feature size, especially below 100 nm, qualitatively new hurdles are emerging. New techniques are used, for example, to *decrease power dissipation*, by switching off temporarily the parts not used,

[*]Correspondence to: András Mozsáry, Jedlik Laboratories of the Faculty of Information Technology, Peter Pázmány Catholic University, Práter u. 50/a, Budapest H-1086, Hungary.
[†]E-mail: mozsary@itk.ppke.hu

Contract/grant sponsor: OTKA (Hungarian National Research Foundation) Doctoral School Framework; contract/grant number: TS40858

predicted by the software running on the chip. Moreover, *wire delay becoming more important than gate delay* and the *synchronization of two distant parts becomes a major problem*. Wires are no longer equipotential surfaces. Designers are also looking for bio-inspired solutions. One example is the cellular neural/nonlinear network (CNN) [1] based cellular wave computer architecture [2, 3]. The connections are mainly local and the many identical cell processors, of speed defined by the parasitic capacitances, interact locally. The strengths and layout of these interactions define the basic functionality: a spatial-temporal wave. This deep-submicron-friendly elementary instruction is embedded in the stored programmable cellular wave computer architecture, called the CNN universal machine [4]. The cells may be working in an analog mode or in an emulated digital way. The extended cells have local memory and programmability, again fully distributed. The emergent spatial-temporal dynamics is defining the function of the array. Moreover, sensing and elementary computing might be integrated. This is just one feature heralding the new era of the function-in-layout design. In view of the non-Boolean processing, we should not forget that a transistor is more than just a switching device. This new world of processing is a complete departure from classical number crunching or binary logic. Interestingly, this is the way and language, for example, of the mammalian retina [5]. Here, the arithmetic depth, a notion introduced by von-Neumann, is very low, since the elementary instruction, the wave action, what is the most difficult task for number crunching computers, is the simplest one, performed by our programmed physics in one single step [4].

Constantly diminishing CMOS feature size has made no major impact on circuit design until 0.18 μm. At this level, propagation delay of logic signals predominantly comes from wiring [6]. At 0.13 μm power dissipation also becomes a limiting factor. At 0.35 μm and above, metal-oxide semiconductor (MOS) gate capacitances played the main role in circuit speed. The turning point was at 0.25 μm, where signal delay resulting from gate capacitances became equal to routing delay caused by aluminium wire parasitics. With further growing integration density, 0.18 and 0.13 μm, gate delay tends to disappear compared to chip delay, while signal routing is the crucial design issue for digital systems. Historically, this was the first problem of deep submicron technologies.

Whenever the abstract algorithm-level description requires data that physically reside in the other end of a chip, a long piece of aluminium or copper wire is to be brought across the whole circuit. This is then a source of huge capacitive load, delay and crosstalk. Function-in-layout approach is a tool for designing systems where function, circuitry and floorplanning are inextricably linked. The idea is to extend hardware–software co-design. Hence, we do not want to force a preconceived architecture; on the contrary, we start from deep-submicron-friendly units and interconnections. Biological examples motivate to formulate problem solving on locally connected array basis: visual, auditory, tactile pathways remain topographic, mainly locally connected all the way up to higher level brain centres.

Function-in-layout concept has emerged when the CNN paradigm [3] started to emphasize local connectivity on hardware level. The success of the series of analogic visual microprocessor chips [7–9] means triumph of the function-in-floorplanning idea. There is only one more step to dive deeper in microelectronic design. Further shortening the wires will bring us to the realm of function-in-layout. In very deep submicron, two-digit nanometer MOS technology, the effect of the parasitics will be significant.

The CNN algorithm, called hyperacuity in time [10] is a candidate to implement the first true layout-engineered function. This is made possible, because the hyperacuity operation runs on a cellular architecture. Each grid point has a simple role that is possible to be realized by as few as six transistors. (More details are in Section 4.) For this small number of transistors and small

DOI: 10.1002/cta

piece of silicon area, with only local connectivity, no wire is longer than a few microns. At future technologies below 0.18 mm, parasitic crosstalk will have a range that it is able to span cell-to-cell distance. This will enable to implement interaction to neighbours using the parasitics themselves. Capacitively connected cell-to-cell communication will be available. Local communication will be made without wire and design optimization will be done with respect to the layout rather than circuit schematic. It is going to be feasible to perform analysis with full parasitic extraction and optimize the cell function with respect to the layout. In other words, it allows us to think a little bit beyond the transistor. This direct implementation of the function in the chip layout gave the name to the concept: *function-in-layout*. The basic ideas are as follows:

- *Processing cells*, components have minimal features depending on the design goal (minimum size, minimum dissipation, maximal speed, etc.). The other parameters are becoming ‘properties’ and not ‘parasitics’.
- *Wires* are introducing significant delay and wave actions, in our example it is used as a construction element.
- *Function* is introduced by wiring; including ‘parasitics’ hence the layout plays a crucial role.

2. BASIC ARCHITECTURE

2.1. What is a TDC?

The function of the presented hyperacuity chip is actually going to be a clock: a time measurement device. The operation is as simple as a stopwatch. There is going to be a start and a stop signal, and the electronics will tell us how much time has elapsed between them. Nowadays, microelectronic technology places the ultimate value in the order of tens of picoseconds. Electronic equipment measuring at this time range is called time-to-digital converter (TDC). Their applications range from reflected laser-light distance measurement to elementary particle lifetime measurement in nuclear physics experiments [11, 12]. TDC is basically performing an analog to digital conversion. However, the continuous-valued input is not expressed by a voltage or current, but a time delay. The timing of an electrical signal represents the analog input information. This is quantized, and logic information is given on the output. The present design study is a TDC that is based on a hyperacuity core function. The circuit receives rising edge transitions on two input pins, and the timing order and amount of delay between these two signals are coded into a binary number as a result.

2.2. Biology motivation

The integrated circuit layout has been inspired by the brain anatomy of the barn owl [13]. It has excellent sound localization skill catching a mouse even in total darkness, relying on acoustic signals only. This owl can hear the direction of the sound, calculating the azimuth based on the time difference that is between its two ears. The two ears have different path length to the sound source, and this produces difference in propagation delay between the two signals. The owl’s brain measures interaural time difference (ITD) [14] by a resolution of 10 μs. Formal sketch of neuron structure is given in Figure 1. One can find a strictly regular set-up, and this is translated onto silicon. The time measurement function is kept, and the electrical circuit is used as a TDC that measures and digitizes time difference between two input electrical pulses.

DOI: 10.1002/cta

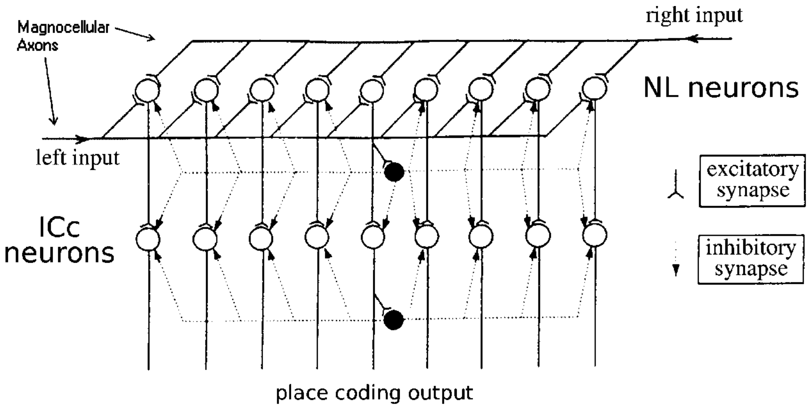

Figure 1. Nerve system for detecting interaural time difference (ITD). Observe the interplay between the local and global interconnections. This part of the barn owl's brain anatomy serves as a starting point for the hyperacuity chip design.

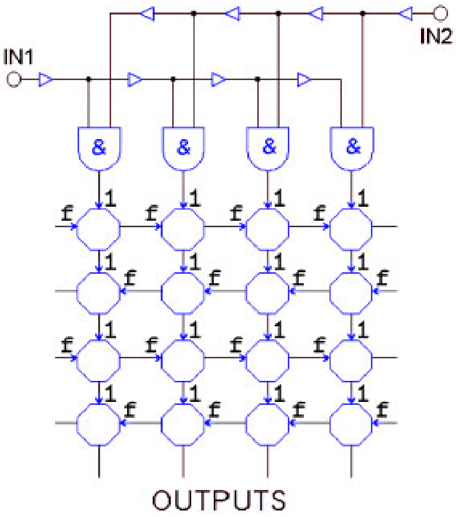

Figure 2. Functional block diagram of the hyperacuity time-to-digital converter chip.

2.3. *Translating anatomy into electrical circuit*

In the following, the circuit architecture is introduced through a comparison between anatomy in Figure 1 and electrical circuit block diagram in Figure 2. Input signals are received from the left and right ear (Figure 1), correspondingly there are IN1 and IN2 inputs on the circuit (Figure 2). Magnocellular axons from left and right deliver action potentials for the row of nucleus laminaris (NL) neurons. As the action potential has finite propagation speed, the signal gets delayed along the magnocellular axon. A delay path in the circuit, including buffers, which are needed to recondition and preserve the signal, implements this. Figure 1 shows that NL neurons receive synapses from

DOI: 10.1002/cta

both magnocellular axons. NL neurons have input from the left magnocellular axon and the right magnocellular axon, but they fire only when both inputs are active simultaneously. This coincidence detection function is implemented by logical AND gate in the electronic hardware. NL neurons are also synapsing each other using inhibitory interneurons. Figure 1 shows only one inhibitory interneuron, that is a filled black dot connected to the middle neuron in NL row. In reality, every neuron has its own inhibitory interneuron, through which it can send inhibition signal to its neighbour NL neurons. For simplicity, only one is drawn in Figure 1. The role of this lateral inhibition is to penalize competitor NL neurons, and implement a kind of winner-takes-all algorithm. This will be explained in Section 3. The CNN cells maintain this lateral interconnection in the electrical circuit in Figure 2. The CNN cell is a custom-designed part of the circuitry, and its details are given in Circuit Implementation Section 4. For circuit design reasons, each CNN cell has just one lateral input and lateral output. Therefore, CNN cells are able to synapse either to the left side neighbour or the right side neighbour. On the contrary, the lateral inhibition network of NL neurons is bi-directional. The electrical circuit can mimic this using two CNN cells to implement one living neuron. One CNN cell synapses to the right and another synapses to the left. This way the total functionality of a NL neuron and belonging inhibitory interneuron is implemented by three blocks in the circuit: one logical AND gate, plus two pieces of CNN cell. The outputs from the NL neuron layer goes to a higher brain centre, named the inferior colliculus (ICc). There the topographic representation is kept and the lateral processing structure is repeated. In the electronic circuit, two more rows of CNN cells correspond to this.

3. FUNCTIONAL DESCRIPTION

Three different types of elements build up the hyperacuity chip: delay path (Figure 3(a)), logic AND gate (Figure 3(b)) and CNN cell (Figure 3(c)). The delay path role is to generate many delayed replica of the input. The delay is uniformly increasing along the line, therefore, each stage implements the single unit delay:

$$\mathrm{Out}(t) = \mathrm{In}(t - \tau) \tag{1}$$

In our 0.35 μm chip implementation $\tau = 30$ ps. Equation (1) is put into a simulator. Simulation result (Figure 4) is given in the form of still image sequence, which are shot at regular time intervals: 150, 300, 450, 600 ps, etc. Subfigure 'Time 1' is taken at 150 ps, 'Time 2' at 300 ps, and 'Time 3' at 450 ps. Altogether 24 subfigures were generated this way.

The operation of the delay path (Equation (1)) is shown the best on subfigures 'Time 1–6'. These illustrate signal propagation from both left and right. The two signal sources IN1 and IN2 (Figure 2) send their step-input to propagate toward the centre.

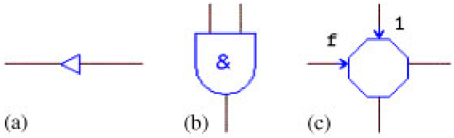

Figure 3. Circuit blocks that build up the hyperacuity chip: (a) delay element; (b) AND gate; and (c) CNN cell hardwired with a 4-connected special template.

DOI: 10.1002/cta

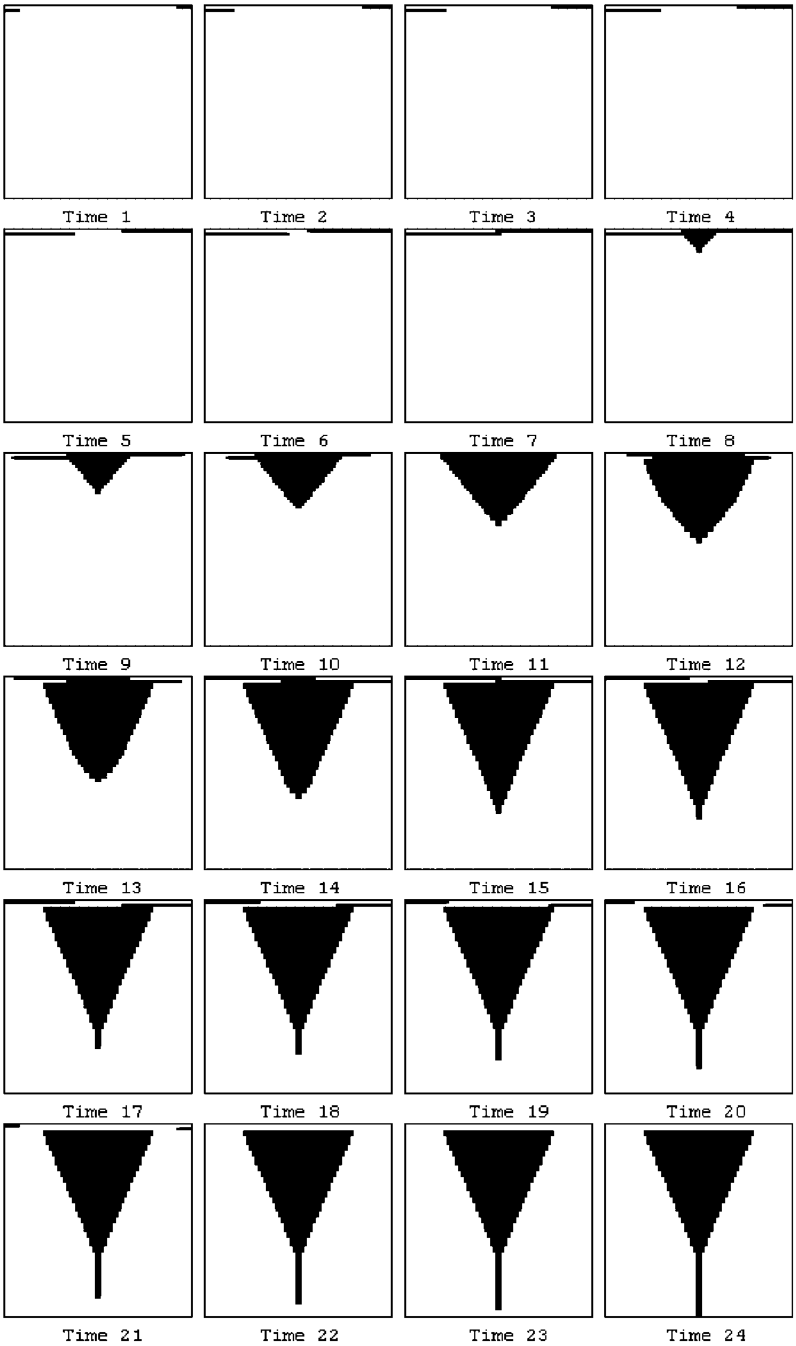

Figure 4. Functional simulation of the hyperacuity chip block diagram. Snapshots at regular 150 ps time intervals are presented to visualize how time is elapsing. Delay path state is represented on the top two pixel rows. CNN wave processing is illustrated on the rest area.

Logic AND gate element is given in Figure 3(b). Its two inputs are connected to the two delay paths. The conventional Boolean definition of AND gate behaviour is $Y = A \cap B$, however, for time domain processing purposes, a time domain equation serves as model for the standard AND gate circuitry:

$$\mathrm{Out}(t) = \min(\mathrm{In}_1(t), \mathrm{In}_2(t)) \tag{2}$$

The outcome of Equation (2) is that a CNN wave is initiated at those column positions, where and when, both delay path signals coexist. This is the role of using the logic AND gate. Please refer to Figure 4; subfigures 'Time 8–10' are examples for that.

Signals come through AND gates, and propagate columnwise downward the CNN array. Each feedforward channel in the anatomy (Figure 1) corresponds to a column of the CNN grid (Figure 2) and also represented by a pixel column in Figure 4. While propagating vertically, the signals interact with their neighbours via lateral interconnection. The vertical propagation speed is controlled by the delay conditions of the neighbours. The individual CNN cell dynamic behaviour is governed by the general CNN state equation:

$$C\frac{\mathrm{d}}{\mathrm{d}t}v_{x_{ij}}(t) = -R^{-1}v_{x_{ij}}(t) + \sum_{kl\in N_r} A_{ij,kl}v_{y_{kl}}(t) + \sum_{kl\in N_r} B_{ij,kl}v_{u_{kl}}(t) + I_{ij} \tag{3}$$

The cell array function is determined by local template matrices A and B that define the connectivity. The hyperacuity CNN template looks rather sparse:

$$A_{\mathrm{odd}} = \begin{bmatrix} 0 & 1 & 0 \\ \mathbf{f} & R^{-1} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = 0, \quad I = 0 \tag{4}$$

$$A_{\mathrm{even}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & R^{-1} & \mathbf{f} \\ 0 & 0 & 0 \end{bmatrix}, \quad B = 0, \quad I = 0 \tag{5}$$

The CNN template has two different forms: A_{odd} (4) and A_{even} (5). The template matrix is horizontally flipped, the matrix element **f** is either on left or right side: $\mathbf{f} = A_{\mathrm{odd}-10} = A_{\mathrm{even}+10}$. Observe that the lateral neighbour interconnection has only one direction on Figure 3(c). The two versions A_{odd} and A_{even} are used simultaneously in the CNN grid. Cells placed in even-numbered rows are governed by A_{even} (5), and cells that are in odd-numbered rows obey A_{odd} (4). In Figure 2 every second row drives to the left, while odd-numbered rows drive to the right. On the contrary, there is symmetric and bi-directional lateral connection in the anatomy (Figure 1). The vertical symmetry of the lateral inhibitory function in owl brain is retrieved in implementation. Consider two CNN rows together, first row A_{odd} then A_{even}. While wavefront passes both rows, it undergoes first A_{odd}, then A_{even} operation. Hence, the lateral connection to both left and right direction is provided.

There is one more difference between anatomy and CNN: namely that inhibitory interneurons synapse up to the third neighbourhood (Figure 1), while CNN cells are connected to their first neighbour only (Figure 2). This makes a CNN cell less effective than a living neuron. This

DOI: 10.1002/cta

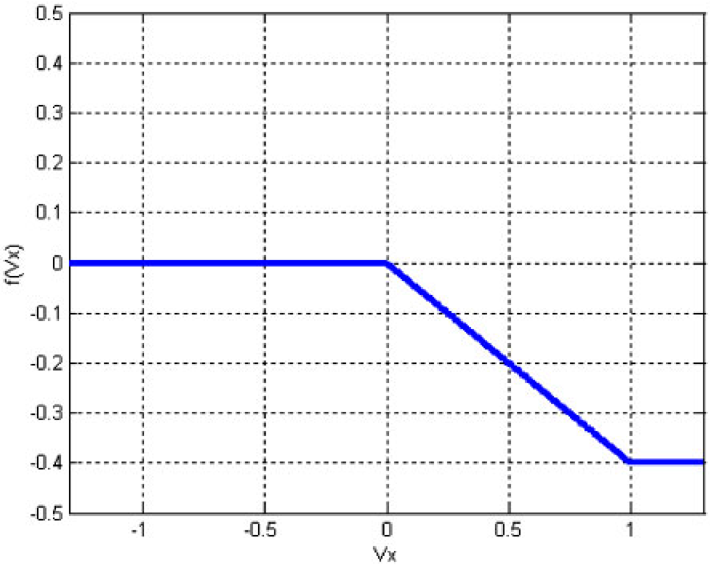

Figure 5. Nonlinear 'synapse' function for the lateral interaction of CNN cells. This function 'f' is used as a template element in template A.

shortcoming is compensated for, using much more CNN rows than the number of neuronal layers. More details on CNN sizing are given in Section 5.2.

The reason behind using one direction and first neighbourhood CNN template in CNN against bi-directional and third neighbourhood lateral inhibition in anatomy is the reduced number of transistors in the CNN cell.

The top-middle constant $A_{0-1} = 1$ is responsible for vertical signal propagation. In Figure 4 the wavefront travels row-by-row, see subfigures: Time 9–11.

The middle-left element $A_{\text{odd}-10} = A_{\text{even}+10} = \mathbf{f}$ is a nonlinear function [15, 16] that is causing the lateral interaction. The piecewise-linear function 'f' is given by diagram, in Figure 5.

The effect of synapse function 'f' is depicted in Figure 4 subfigures: Time 12–15. Signals, which are at the side of the wavefront, are penalized. They get marooned and finally lost. Propagation stops everywhere, except for the tip. The leading edge of the wavefront has no signal ahead itself, no opponent to compete. Therefore, it reaches the bottom row eventually. This is given in subfigures Time 16–24. The position of the centre is read out than, because this is coding the initial input timing.

For the sake of completeness, we have to emphasize that CNN model Equation (3) also implies v_y. This is generated via the conventional sigmoid-type CNN output nonlinearity:

$$v_{y_{ij}}(t) = \text{sigm}(v_{x_{i-1j}}(t)) = 0.5(|v_{x_{ij}}(t) + 1| - |v_{x_{ij}}(t) - 1|) \tag{6}$$

We will substitute Equations (6) and (4) into Equation (3), so that the state equation of the CNN cell will yield in closed form (7). The central element $A_{00} = R^{-1}$ (Equations (4) and (5)) cancels the dissipative term $-R^{-1}$ in Equation (3). From Figure 5 it is observed that function **f** already implies sigmoid-type saturation, therefore, applying Equation (6) to **f** is invariant, and therefore it

DOI: 10.1002/cta

is omitted. The hyperacuity CNN cell state equation is given as

$$C\frac{\mathrm{d}}{\mathrm{d}t}v_{x_{ij}}(t) = f(v_{x_{ij-1}}(t)) + \mathrm{sigm}(v_{x_{i-1j}}(t)) \tag{7}$$

4. THE CIRCUIT IMPLEMENTATION DETAILS

Three different types of elements build up the hyperacuity chip: wire buffers, logic AND gates, and CNN cells. The transistor-level implementation of the buffers and logic AND gates follow the standard source-coupled logic (SCL) style. The CNN cell is a custom design, though the starting point for the CNN cell circuitry was also an SCL-type buffer. There is a little modification added to it. Figure 6 shows the circuit schematic. The CNN cell consists of six transistors. Five of them belong to an SCL buffer: one current source transistor, two compose a differential amplifier pair, and two diode-connected transistors serve as load resistor. The differential input +IN and −IN, and the differential output +OUT and −OUT connect rows/layers to each other, while Lin and Lout implement the intra-layer lateral interaction. The mechanism of this is the following: the Lin input controls a transistor that opens an alternative current path, other than the differential pair. This way, Lin signal can reduce the bias current injected in the differential pair. In this case, the transconductance of the buffer decreases, subsequently the propagation speed from IN to OUT drops.

4.1. Simulation of the CNN cell

CNN cells are arranged in rows, where each cell receives a signal into its IN terminal. Those signals are then processed according to the neighbour's state. Communication from the neighbour is made through the Lout–Lin connections. The processed result is then presented at the OUT terminals that are attached to the next row input. The operation is done using delay-domain computing style. This means that the information in these CNN cells is represented by the timing of the pulses throughout the whole process. Therefore, the relation that is characterizing the CNN cell behaviour is the IN–OUT propagation delay as a function of the delay between neighbouring IN terminals. IN–OUT propagation delay is defined beginning when +IN and its inverse, −IN become equal, elapsed to the time when +OUT and its inverse, −OUT become equal. This is the definition for

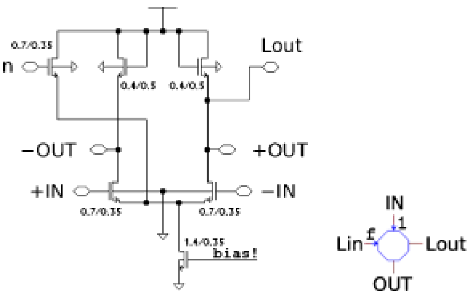

Figure 6. Transistor circuit schematic and symbol of the CNN cell. Five transistors out of six compose a differential SCL-type buffer and one transistor with input Lin synapses to the neighbouring cell.

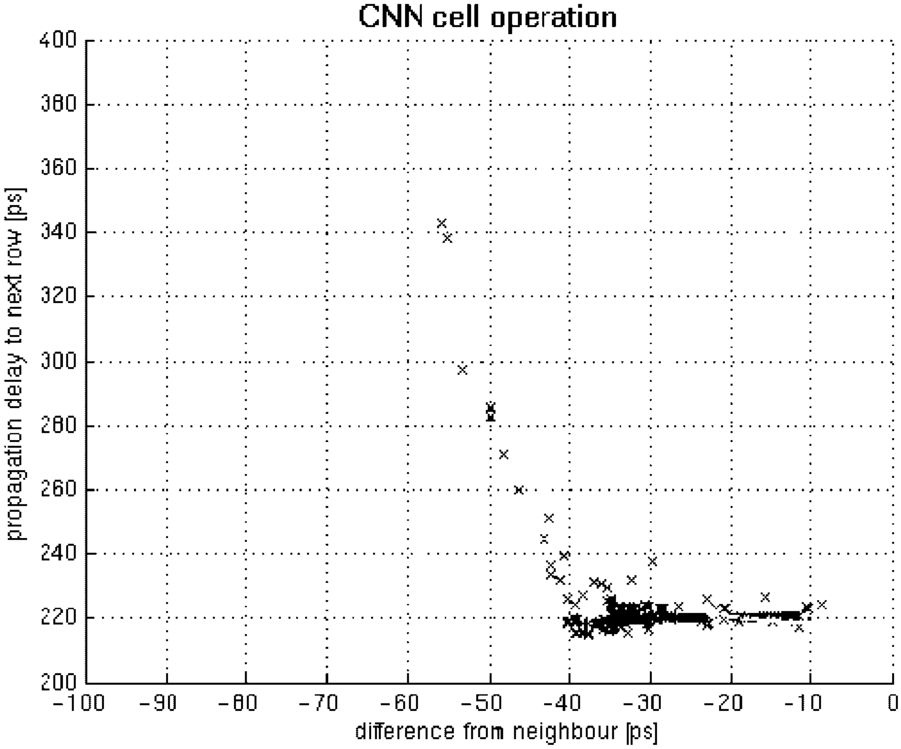

Figure 7. Behaviour of the CNN cell lateral connection. Data points were recorded through post-layout SPICE simulation.

zero crossing of the differential signal. For labels +IN, −IN, +OUT, and −OUT refer to Figure 6 circuit schematic. Figure 7 depicts the SPICE simulation result. Figure 7 yields that the IN–OUT propagation speed in normal conditions is around 220 ps. However, when a certain cell receives IN signal much later than the neighbouring cell, the output will get delayed. According to the diagram, the effect is put into action at around 20 ps. Conclusively, a cell that is more than 20 ps ahead can penalize others who are late. Over 60 ps lag, the lateral inhibition will stop propagation. This effect eventually cancels activity in all columns that are not at the edge. The resulting synapse function in Figure 7 is not a reproduction of synapse function 'f' in Figure 5. Instead, it is a functional implementation. The implementation goal was the hardware simplicity. The developed single transistor synapse saves much silicon area. On the other hand, it is able to substitute 'f' (Figure 5). This is justified by comparing Figures 4 and 8. More details on simulation (Figure 8) are presented at the end of the next section.

5. OPERATION PRINCIPLE

The hyperacuity TDC circuit is functionally divided into two main parts: the delay path and a CNN network. The role of the delay part is to perform the topographic mapping of the

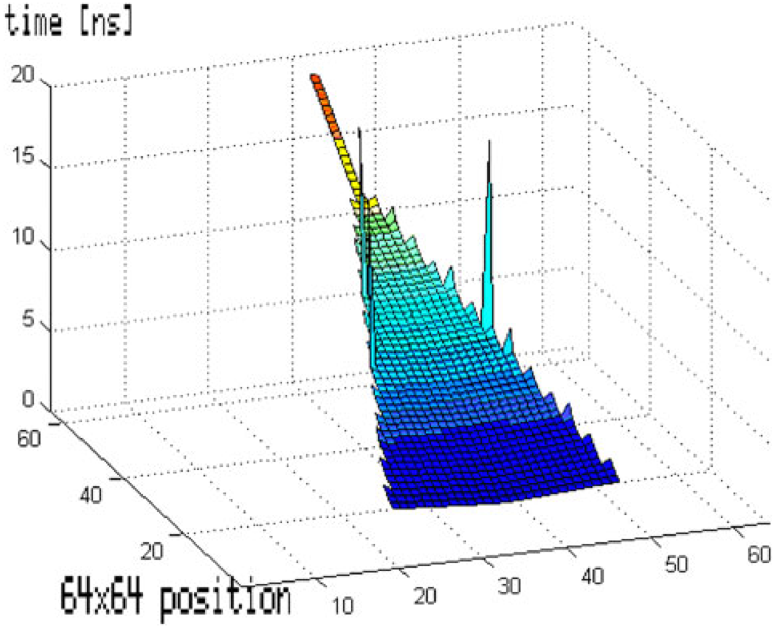

Figure 8. Three-dimensional diagram illustrating the wavefront propagation on the cellular neural/nonlinear network (CNN). In SPICE transient simulator the time was recorded for each cell when pulse leading edge arrived. This is plotted over the 64 × 64 grid.

input information (details in the next section). The CNN network is responsible for the amplification of the stimuli map and decision-making according to a winner-takes-all strategy. Of course, there are some additional glue logic, integrated on-chip: a register and a multiplexer were needed to interface the hyperacuity chip to the outside world. These make result data readout possible.

5.1. Topographic mapping: the spatio-temporal convolution

The operation principle (illustrated in Figure 9) is as follows: the two input channels IN1 and IN2 are fed into the two identical delay paths, running against each other. After a while, the propagating signals meet each other. In case where input $\Delta T = 0$, this coincidence occurs at the middle position. On the other hand, whenever there is a delay at one input with respect to the other, this will cause the coincidence point to shift, become off-centre. Figure 2 illustrated a four column CNN network, but the real chip implementation has 6-bit resolution. Therefore, it has 64 columns in the CNN array. After the first coincidence, propagation starts in that particular column of CNN network. This is going to be the winner column. Of course, signals still continue to run in the delay path further, causing other CNN columns to become active as well. First, the coincidence point, then the neighbouring columns, and so on. This forms a kind of triangle-shaped wavefront propagating downwards on the CNN (Figure 4, Time 10). At system-level approach, one could say this triangle is a direct outcome of the convolution of the input step functions. The delay paths

DOI: 10.1002/cta

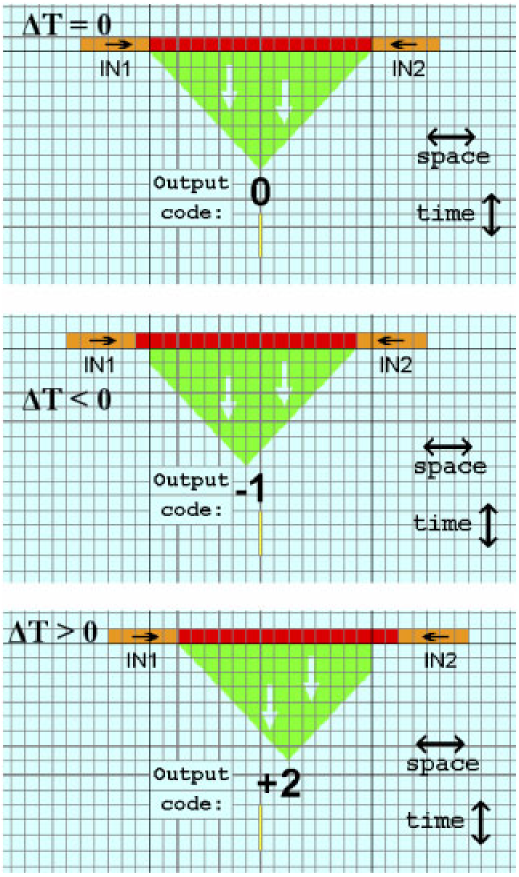

Figure 9. Method of generating output code in the hyperacuity chip. Triangle is the outcome of spatio-temporal convolution of rectangular pulses that are received through IN1 and IN2. The position of the triangle tip depends on the timing difference between the two inputs.

running in opposite direction together with the AND gates formulate a spatial realization of the convolution integral. So we could say that there is a convolution integral implementation in the owl's brain. It is programmed by the physical placement of neurons in NL: Magnocellular axons are running in opposite direction along the regularly positioned row of identical NL neurons. After the NL stage, the auditory information is represented as the location of the neuronal activity along the row. The topographic representation of the sensory field is preserved up to higher brain centres. Therefore, in the artificial realization there is also a spatio-temporal processing engine employed for decision-making: a CNN.

5.2. The CNN cell array

The function of the CNN array is to enhance the timing conditions and evaluate the result. The circuit realization of system Equation (7) was given in Figure 6. That differential amplifier stands

DOI: 10.1002/cta

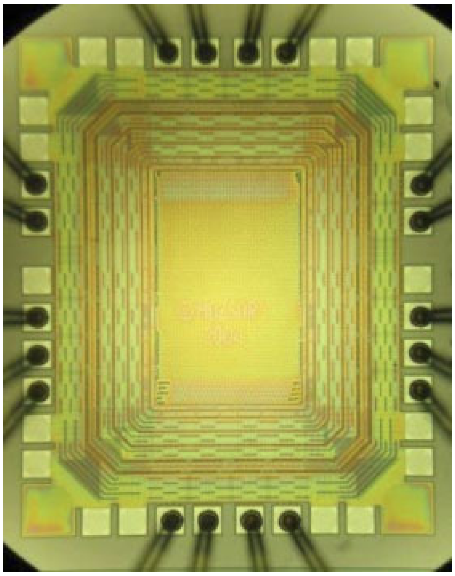

Figure 10. Micrograph of the hyperacuity prototype chip.

for the integrator, with the parasitic input capacitance taken as the state variable. The sigmoid function corresponds to the saturation of the circuit at supply rails. This circuit contains single transistor implementation for the lateral interaction between cells (denoted by function 'f' in Equations (4) and (5)). The lateral interaction function actual shaping (shown in Figure 7) was driven by hardware simplicity. Of course, the function's domain contains only negative numbers, as this is a function defined over delay, and as such, ought to be causal. When a certain signal is too much late than its neighbour, (>20 ps) it will be slowed down, having more delay. This is a self-steering effect. The slow signal will be slower and slower, and finally disappears. Only the fastest signal remains, and the leading edge will indicate the amount of detected input delay finally. Figure 8 is a SPICE simulation result of the whole hyperacuity circuit. The single column that manages to reach the other side of the CNN grid codes the result: its lateral position is the digital output code, a co-ordinate between 1 and 64. The CNN grid is 64 columns wide, so that the resolution of the output is 6 bit. The CNN is 64 rows deep, so that the wave has enough space to run and formulate its final tip. The necessary length of the runway was determined and the 64×64 size for the CNN was fine-tailored through simulation.

6. CHIP MEASUREMENTS

The hyperacuity chip was developed using 0.35 CMOS prototype fabrication runs. There have been three iterations, and the last really successful chip has the name 'Hypthree'. See chip micrograph in Figure 10, and chip technical data in Table I. The main measurement result is the time-to-digital conversion characteristics, given in Figure 11. The circuit possesses perfect linearity that is the outcome of the regular architecture. The low yield of 33% and the high noise of 78 ps RMS

DOI: 10.1002/cta

Table I. Technical data of the hyperacuity prototype chip.

Hypthree chip	9th March 2006
Function	Time measurement (TDC)
Sensitivity	LSB = 29.3 ps (single shot)
Resolution	6 bit
Full scale	1.9 ns
Accuracy	INL = 77.9 ps = 2.7 LSB (RMS)
Technology	0.35 μm CMOS
Manufactured	TSMC, year 2005
Yield	32.6%
Missing codes	67.4%
Chip dimensions	1139 μm × 1230 μm
Power consumption	675 mW at 3Vdd

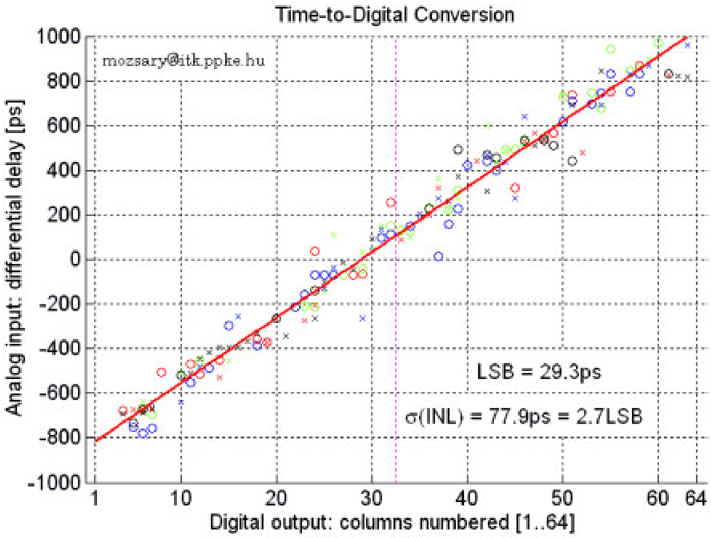

Figure 11. Measurement of time-to-digital conversion (TDC) characteristics. The regression fitting line steepness is 29.3 ps per column, this is the conversion unit.

systematic jitter address the problem of robustness, therefore, the analog part of the circuit still needs to be optimized. The main design goal is the fine resolution in timing detection: first-order fitting yielded 29.3 ps to be the average conversion step. This qualifies the hyperacuity chip to find its place among state-of-the-art TDC chips. Table II summarizes key parameters of the present day TDC designs. Compared to different existing TDC architectures, CNN-based hyperacuity offers a good alternative solution in terms of single-shot resolution and conversion speed.

Table II. State-of-the-art in time-to-digital converter design. Hyperacuity-based TDC outperforms existing techniques in terms of single-shot time resolution and conversion speed.

Design/architecture	Channel width (LSB)	Conversion speed (latency time) word length	Chip size, technology
Nested DLLs [17]	92 ps 780 ps	7 bit interpolator + counter	0.8 μm CMOS 3.1 × 2.2 mm
Linear pulse-shrinking [18]		6 bit interpolator + counter	1.2 μm CMOS 2.9 × 2.5 mm
Cyclic pulse-shrinking [19]	68 ps	100 ksps 8 bit	0.35 μm CMOS ($L = 1$ μm) core size: 0.35 × 0.09 mm
Cyclic pulse-shrinking [20]	20 ps, (std = 3.8 channels)	50 ksps 10 bit	0.8 μm CMOS 1 × 2 mm
Cyclic delay line [21]	156 ps	4 bit interpolator + counter	AMS 0.35 μm CMOS 1.81 × 1.81 mm
FPGA differential delay lines [22]	200 ps	6 bit	0.65 μm CMOS FPGA
Differential delay lines + DLL [23]	60 ps	7 bit 130 Msps	0.7 μm CMOS 3.2 × 3.1 mm
Hyperacuity chip	30 ps (std = 2.6 chan.)	6 bit>200 Msps	0.35 μm CMOS 0.3 × 0.3 mm

7. CONCLUSIONS

A new design principle, function-in-layout, is briefly presented. It is efficient in the design example shown in the paper. The hyperacuity-in-time chip performs a 30 ps accuracy, equivalent of 10 mm path of a light beam, using 0.35 μm CMOS technology.

It is foreseen that this design principle will be more and more important for sub-100 nm CMOS design, and a mainly locally coupled architecture with sparse bus-like wires would prevail in many circuits and system designs in the future.

ACKNOWLEDGEMENTS

The authors wish to thank the continuous support at the Jedlik Laboratories of the Faculty of Information Technology of the Peter Pázmány Catholic University (PPCU), Budapest, Hungary. Also recalling the substantial help on electronic circuit problems and implementation issues from Prof. Angel Rodríguez-Vázquez at Centro Nacional de Microelectrónica, Instituto Microelectrónica de Sevilla (CNM-IMSE) Seville, Spain. Special thanks to Prof. Chin-Teng Lin and his laboratory at National Chiao Tung University, Department of Electrical and Control Engineering (NCTU-ECE), Hsinchu, Taiwan, for hosting the implementation and making the hyperacuity chip into a physical reality.

REFERENCES

1. Chua LO, Yang L. Cellular neural networks: theory. *IEEE Transactions on Circuits and Systems* 1988; **35**: 1257–1272.
2. Roska T. Cellular wave computers for brain-like spatial-temporal sensory computing. *IEEE Circuits and Systems Magazine* 2005; **5**(2):5–19.
3. Chua LO, Roska T. The CNN paradigm. *IEEE Transactions on Circuits and Systems I* 1993; **40**(3):147–156.
4. Roska T, Chua LO. The CNN universal machine—an analogic array computer. *IEEE Transactions on Circuits and Systems—II* 1993; **40**:163–173.
5. Werblin F, Roska T, Chua LO. The analogic cellular neural network as a bionic eye. *International Journal of Circuit Theory and Applications* 1995; **23**:541–569.
6. Lev L, Chao P. Down to the wire—requirements for nanometer design implementation. *White Paper*, Cadence Design Systems Inc., 2003.
7. Dominguez-Castro R, Espejo S, Rodríguez-Vázquez A, Carmona R, Foldesy P, Zarandy A, Szolgay P, Sziranyi T, Roska T. A 0.8-μm CMOS two-dimensional programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage. *IEEE Journal of Solid-State Circuits* 1997; **32**(7):1013–1026.
8. Liñan G, Földesy P, Espejo S, Domínguez-Castro R, Rodríguez-Vázquez A. A 0.5μm CMOS 10^6 transistors analog programmable array processor for real time image processing. *Proceedings of the Fifth European Solid-State Circuits Conference*, Duisburg, Germany, September 1999; 358–361.
9. Rodríguez-Vázquez A, Liñan-Cembrano G, Carranza L, Roca-Moreno E, Carmona-Galan R, Jimenez-Garrido F, Dominguez-Castro R, Meana SE. ACE16k: the third generation of mixed-signal SIMD-CNN ACE chips toward VsoCs. *IEEE Transactions on Circuits and Systems I* 2004; **51**(5):851–863.
10. Lotz K, Bölöni L, Roska T, Hámori J. Hyperacuity in time: a CNN model of a time-coding pathway of sound localization. *IEEE Transactions on Circuits and Systems—I*: *Fundamental Theory and Applications* 1999; **46**(8):994–1002.
11. Christiansen J. An integrated CMOS 0.15 ns digital timing generator for TDCs and clock distribution systems. *IEEE Transactions on Nuclear Science* 1995; **42**(4):753–757.
12. Gorbics MS, Kelly J, Roberts KM, Sumner RL. A high resolution multihit time to digital converter integrated circuit. *IEEE Nuclear Science Symposium*, *Conference Record*, vol. 1. 2–9 November 1996; 421–425.
13. Carr CE, Konishi M. A circuit for detection of interaural time differences in the brain stem of the barn owl. *Journal of Neuroscience* 1990; **10**:3227–3246.
14. Fujita I, Konishi M. The role of GABAergic inhibition in processing of interaural time difference in the owl's auditory system. *Journal of Neuroscience* 1991; **11**:722–739.
15. Jacobs A, Roska T, Werblin F. Methods for constructing physiologically motivated neuromorphic models in CNNs. *International Journal of Circuit Theory and Applications* 1996; **24**(3):315–339.
16. Lotz K, Jacobs A, Vandewalle J, Werblin F, Roska T, Vidnyánszky Z, Hámori J. Cellular neural network realizations of neuron models with diverse spiking patterns. *International Journal of Circuit Theory and Applications* 1996; **24**(3):301–314.
17. Mantyniemmi A, Rahkonen T, Kostamovaara J. A high resolution digital CMOS time-to-digital converter based on nested delay locked loops. *IEEE International Symposium on Circuits and Systems*, *ISCAS'99 Proceedings*, vol. 2. 1999; 537–540.
18. Raisanen-Ruotsalainen E, Rahkonen T, Kostamovaara J. A low power CMOS time-to-digital converter. *IEEE Journal of Solid-State Circuits* 1995; **30**(9):984–990.
19. Chen P, Liu S, Wu J. A CMOS pulse-shrinking delay element for time interval measurement. *IEEE Transactions on Circuits and Systems—II* 2000; **47**(9):954–958.
20. Tisa S, Lotito A, Giudice A, Zappa F. Monolithic time-to-digital converter with 20 ps resolution. *European Solid-State Circuits Conference ESSCIRC'03*, 2003; 465–468.
21. Nissinen I, Mantyniemi A, Kostamovaara J. A CMOS time-to-digital converter based on a ring oscillator for a laser radar. *European Solid-State Circuits Conference ESSCIRC'03*, 2003; 469–470.
22. Kalisz J *et al.* Field-programmable-gate-array-based time-to-digital converter with 200 ps resolution. *IEEE Transactions on Instrumentation and Measurement* 1997; **46**:51–55.
23. Dudek P, Szczepanski S, Hatfield JV. A high-resolution CMOS time-to-digital converter utilizing a vernier delay line. *IEEE Journal of Solid-State Circuits* (JSSC) 2000; **35**:240–247.

DOI: 10.1002/cta