A VLSI Hamming Artificial Neural Network with k-Winner-Take-All and k-Loser-Take-All Capability

Stéphane Badel, Alexandre Schmid, Yusuf Leblebici

Swiss Federal Institute of Technology, Microelectronic Systems Laboratory
CH - 1015 Lausanne, Switzerland

alexandre.schmid@epfl.ch

***Abstract* - A novel circuit-level Hamming artificial neural network architecture based on the principle of analog charge-based computation of the neural function is proposed. k-winner-take-all and k-loser-take-all operations are performed in the time-domain, allowing for fast and compact realization of complex functions. The VLSI realization of a two-dimensional array arrangement of the Hamming network is presented, with the targeted precision alignment image processing application.**

I. INTRODUCTION

The complexity of artificial neural network (ANN) algorithms has opened the door to their hardware implementation with the main target of accelerating all computations involved. Silicon CMOS has proven to be an advantageous medium, allowing for high integration density and very fast operation. Hence, many realizations have been demonstrated both in the digital and analog domains [1]-[2].

However most of the proposed circuits face severe limitations in terms of circuit resolution, temporary memory size and their access schemes, realization of high interconnect density. Moreover, the increasing demand for low-power embedded systems remains a hurdle for computation intensive ANN related algorithms. Analog VLSI addresses these issues by the implementation of analog atomic elements, each processing some specific neural functions, and to be repeated into a regular structure forming a high-performance processing unit.

The hardware Hamming ANN implementation proposed in this paper follows this approach, with the goal of constructing a building block for high-speed and low-power image processing applications.

II. CAPACITIVE-BASED NEURAL HAMMING OPERATION

A Hamming ANN is a two layer feed-forward neural network which has the ability of classifying input patterns, based on the criterion of the Hamming distance between previously stored patterns, and the actual input vectors [3]. It always converges towards one of the stored patterns, as a benefit of its architecture. The first layer – called the quantifier network – is composed of a number of neurons which perform the Hamming distance computation. The second layer – called the discrimination layer – is traditionally composed of a feed-forward network which performs the winner-take-all (WTA) operation, i.e. selects the first-layer neuron of smallest Hamming distance as the winner.

Pattern classification applications based on the charge-based operation of the Hamming VLSI circuit have been previously demonstrated [4]-[6], where the quantifier networks is composed of capacitive-threshold logic (CTL) gates [7], and the discrimination network consists of an n-input version of a sense-amplifier, all processing being thus performed in the analog domain. CTL has been proven to accommodate a very large fan-in while performing the weighted sum of input vectors at high-speed and low power, thus qualifying for signal processing applications [8].

The design proposed in this paper consists of a modified CTL-based first layer network, driving decision circuits, which in turn are to be connected to analog or digital post-processing depending on the targeted application, to replace regular WTA units with increased functionality units performing in the time-domain.

The quantification network consisting of CTL gates is depicted in Fig. 1, where a number of capacitances make the bridge between a common node called row and the set of digital inputs and prechage circuitry. A perturbation

0-7803-7898-9/03/$17.00 ©2003 IEEE

column is added to the regular CTL gate as an analog input to modify the row voltage state [9]. The comparator and buffer stages build the decision network which provides the outside world with a binary decision.

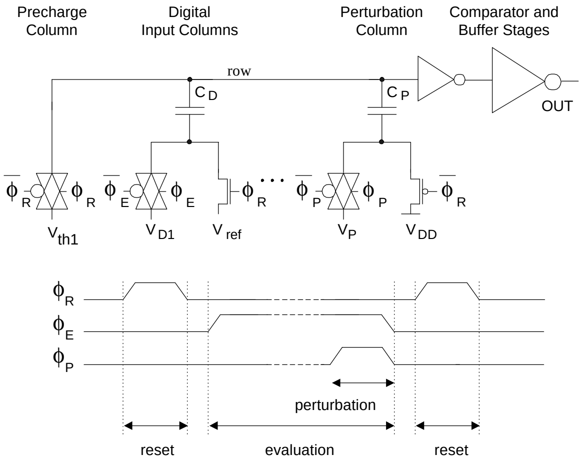


Fig. 1. CTL gate with additional analog perturbation column, and the driving signals.

The circuit operation is based on a three cycles scheme, throughout which charge conservation applies to the row node. The quantification operation is realized with a very simple two phase non-overlapping clock scheme consisting of a precharge (Φ_R) and evaluation phase (Φ_E). All nodes are imposed a voltage during the precharge phase. The row voltage is set to an externally imposed voltage V_{th1} while the capacitances other nodes' voltages are imposed as reference voltages set to V_{DD}, or GND with minor adaptation of the circuit. Note that the V_{th1} row precharge voltage can be conveniently synthesized as the comparator's threshold voltage, as used in a later phase, e.g. Fig. 6. The amount of charge transferred to the row is equal to:

$$Q_{reset} = (V_{th1} - V_{ref})C_D^{TOT} + (V_{th1} - V_{DD})C_P \quad \text{with} \quad C_D^{TOT} = \sum_n C_{D_n} \tag{1}$$

All nodes are set back into high impedance after completion of the precharge phase. The subsequent evaluation phase starts then, throughout which several vectors may be applied without performing any extra reset. The charge on the row node is considered as constant, the time constant of the leakage parasitic process being significantly larger than that of the system operation. However, the charge on the row capacitors nodes is affected:

$$Q_{eval} = \sum_n (V_{row} - V_{D_n})C_{D_n} + (V_{row} - V_P)C_P \tag{2}$$

Assuming the equality of these two charges, the row voltage is forced to vary to ΔV_{row} :

$$\Delta V_{row} = \frac{\sum_n (V_{D_n} - V_{ref})C_D^{TOT} + (V_P - V_{DD})C_P}{C_D^{TOT} + C_P} \tag{3}$$

Eventually, the comparator circuit restores a binary voltage depending on the sign of the row voltage variation (here V_{out} is considered at the output of the comparator stage):

$$\begin{cases} \Delta V_{row} < 0 \rightarrow V_{OUT} = V_{DD} \\ \Delta V_{row} > 0 \rightarrow V_{OUT} = GND \\ \Delta V_{row} = 0 \rightarrow \text{limit of the circuit precision} \end{cases} \tag{4}$$

SPICE simulations of the whole process can be seen on Fig. 2 where a first layer of twelve neurons was used.

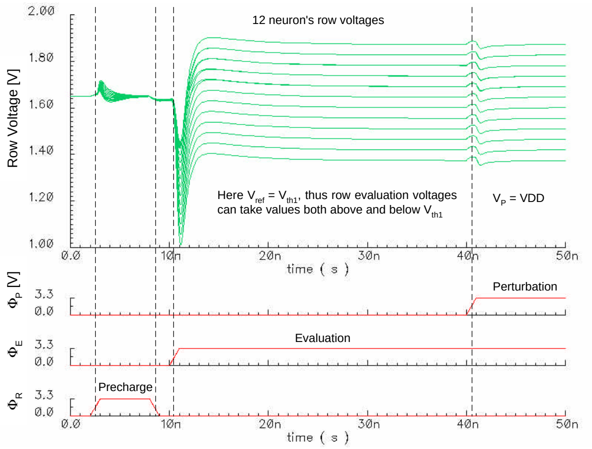


Fig. 2. SPICE simulation of several neurons in the precharge and evaluation phases, each having a different input vector, and thus a different evaluation row voltage.

The perturbation column remains set to its reset value until the discrimination process is to start driven by Φ_P. An analog signal is then applied to C_P forcing the row voltage to vary accordingly. Again, each row's comparator circuit switches at different moment in time, depending on the voltage values reached after evaluation. Fig. 3 shows the circuit reaction to a sinusoidal perturbation signal.

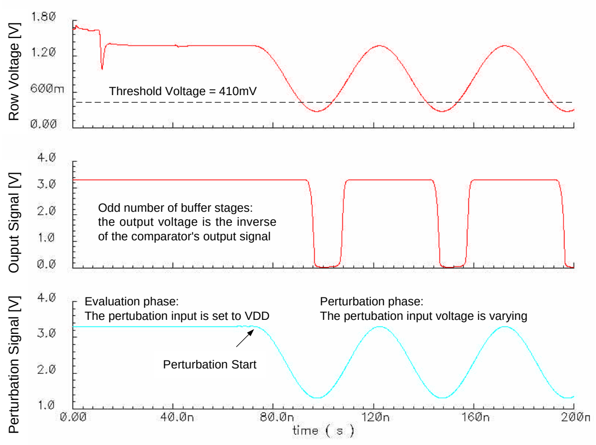


Fig. 3. SPICE simulation of a neuron with sinusoidal perturbation of its row voltage. Note that the polarity of the output signal is related to the logic depth of the buffer stages.

III. OPERATION MODES OF THE HAMMING ANN

A number of working modes can be derived from the basic architecture described previously. The selection must be made prior to design, as to each working mode corresponds some circuit operation characteristics which is depending on the targeted application, as well as their related hardware.

The circuit can be operated to detect relative or absolute Hamming distances. The neurons configurations for these two modes are depicted in Fig. 4.

In order to work on relative distances, only the capacitances representing a ‘Logic 1’ value to be stored for later comparison are integrated. Thus the operation actually implemented is an AND between the input and the pattern stored in the capacitances. Thus, the operation of this sort of neuron results in a distance computation that is not absolute, i.e. it must be compared to the respective results of other neurons computations in order to produce a meaningful information. Weighted computations are allowed in this mode, typically using different capacitance values, which allows for more advanced discriminations [6].

Using a unit capacitance for each input, while processing a digital Hamming distance operation prior to the capacitive stage allows the computation of the absolute Hamming distance. The logic operation to be performed is an XOR between the input and stored data. The row voltage after evaluation of this sort of neuron is proportional to the Hamming distance, thus it is possible to use it either way, with or without any WTA unit.

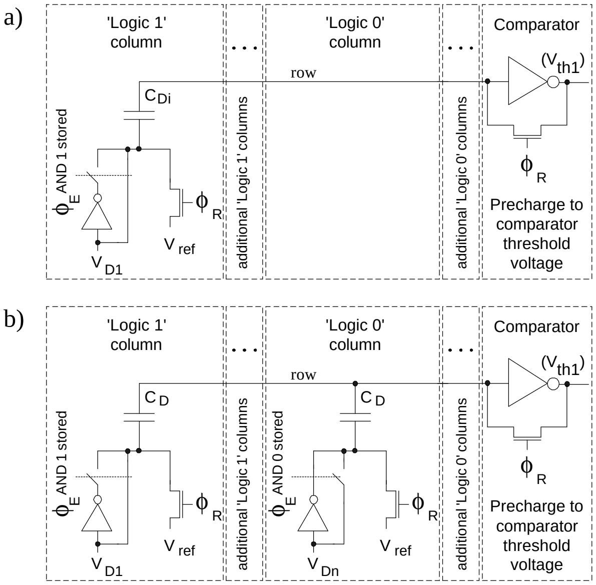


Fig. 4. Hamming network neurons configured for a) relative and b) absolute distance computations.

The second proposed working mode is related to the kind of perturbation signal that is applied to the perturbation columns, and hence affect the row voltages. In this paper we consider the perturbation signal to be equal for each neuron, which is not restrictive and can be easily modified for further applications. Both a ramp and a pulse perturbation signal prove to be interesting candidates, targeting at very different potential applications. Their respective effect on row voltage can be seen on Fig. 5.

Very fast system response can be achieved applying a pulse of calibrated amplitude. All row voltages amplitudes are affected according to the voltage reached at the end of the evaluation period. Thus, only the neurons having a limited row voltage variation at the end of evaluation will see their row voltage drop below the decision circuit threshold. Hence, all input vectors with a Hamming distance to a stored patter smaller than a value given by the perturbation amplitude are discriminated in one single shot as the only vector which have switched polarity after applying perturbation.

Using a ramp signal perturbation affects the row voltage to drop as ramp. Thus, depending their voltage value at the end of the evaluation period, each neuron with dissimilar inputs has its output switching with a different delay. This way, the Hamming distance computed during evaluation is transformed into delay, i.e. in the time domain.

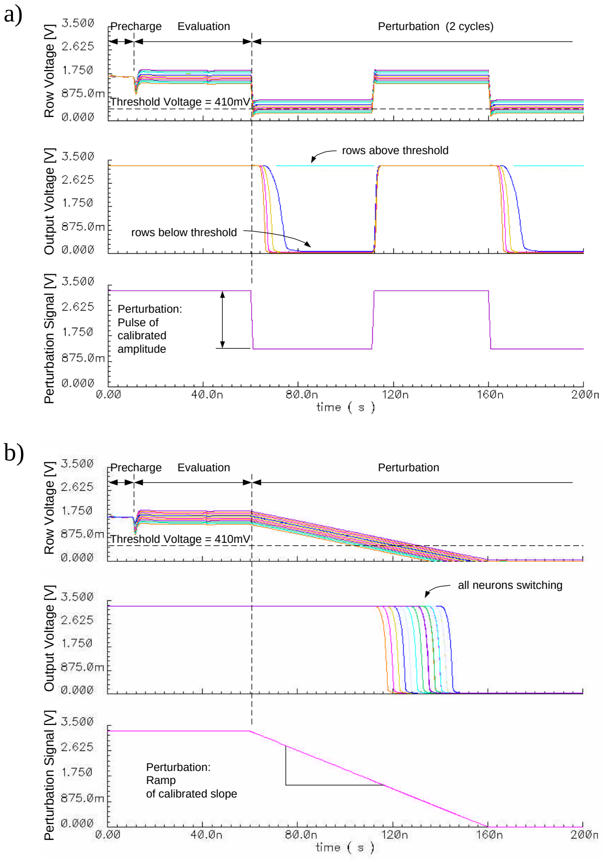


Fig. 5. a) Pulse signal and, b) ramp signal perturbation applied to the perturbation capacitance and their respective effect on the row nodes of several neurons with different input vectors.

Nevertheless, some very fast operation times can be achieved using this working mode. A limited amount of digital and mixed-mode circuitry has to be added as postprocessing elements. Regular WTA operation is achieved by detecting the neuron which switches at last as a result of ramp perturbation. Conversely, Loser-take-all (LTA) consists of detecting the neuron which switches the first. Moreover, the k-WTA and k-LTA operations simply consist of detecting the k neurons which switch their output polarity at last, respectively the first. The postprocessing circuits allowing these features can be developed from regular Boolean logic. Nevertheless, a better solution consists in using a CTL gate to perform the k-switches operation, which in turn triggers a bench of latches to capture the network output state as the computation result.

Similarly, using a limited amount of digital circuitry which can be easily synthesized, it is possible to achieve vector ranking based on their Hamming distances with the stored patterns. One possible post-processing circuit would consist of a $\log_2(n)$-bit counter, n $\log_2(n)$-bit memory cells, and some combinatory logic. Each switching neuron triggers both the counter, and the latching of the counter's value into the neuron's own related memory cell, allowing for proper ranking of all events including simultaneous switching.

Finally, some slight modifications of the hardware are shown to produce dramatically different row voltage responses during the evaluation process. The possible cases are shown in Fig. 6, which the following Paragraphs refer to. Cases 1 and 4 depict the maximal allowed row voltage amplitude, and Cases 2 and 3 depict the minimal forced row voltage. An offset capacitance with a value of one half of the unit capacitance is added on each neuron's row to guarantee this minimal voltage difference from the decision threshold of the comparator gate.

In the previous, it was admitted that the precharge state of the input column be GND, and the precharge state of the perturbation column be V_{DD}. Following these assumptions, the circuit state during evaluation is depicted by Cases 1 and 2, where the evaluation process causes a voltage increase above the decision threshold level, and the subsequent perturbation ramp has to start at V_{DD} and drop to GND. Alternatively, it is possible to switch the precharge value of input and perturbation columns, resulting in an evaluation state depicted by Cases 3 and 4. Here, the evaluation voltage is lower than the threshold level due the precharge at V_{DD} being higher than the total voltage induced by the input vector. The perturbation ramp, or pulse have to be driven from a lower to a higher voltage.

Considering one of these modes, for example precharge of input columns to GND, it has been shown that the row voltage response is limited to Cases 1 and 2. Moreover, a hardware driven selection of the affectation of WTA and LTA to either Cases 1 or 2 can be achieved by the proper selection of the logic function to be applied to the input and stored bits prior to application of the operation result to the capacitance. The selection of the XNOR causes a row voltage increase when both the input and the stored vector have bit-by-bit similarities, whereas the inverted case is true for the selection of the XOR. Consequently, using an XNOR, the Winner neuron is depicted by Case 1 and the Loser neuron by Case 2; using an XOR, the Winner neuron – the one with maximal similarity – is depicted by Case 2 and the Loser neuron by Case1. Noticing that the neuron depicted in Case 2 always switches the first during the perturbation phase, it becomes possible to adapt the hardware to the desired application in order to have the fastest circuit response time.

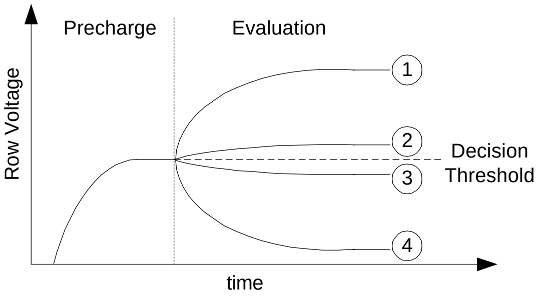


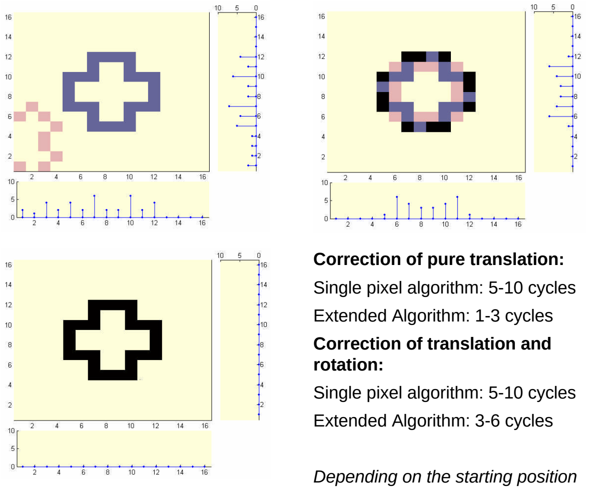
Fig. 6. Different row voltage behavior for a same input, depending on the Hardware configuration.

IV. AN IMAGE PROCESSING BUILDING BLOCK

A two-dimensional array arrangement of the Hamming network where each data is connected to a horizontal and a vertical row simultaneously, thus allowing direct mapping with a black-and-white image was demonstrated as a successful circuit for precision alignment systems. Closed-loop system simulations were run using Matlab software. The corrections algorithms were intentionally kept very simple in order to maintain a very low hardware overhead.

Absolute Hamming distance computation, ramp signal perturbation and regular XNOR Hamming operation were chosen as the working mode.

A simulation showing a pattern to be aligned on a centered symmetric reference can be seen on Fig. 7. More details on the implemented correction algorithms can be found in [10].



Correction of pure translation:

Single pixel algorithm: 5-10 cycles

Extended Algorithm: 1-3 cycles

Correction of translation and rotation:

Single pixel algorithm: 5-10 cycles

Extended Algorithm: 3-6 cycles

Depending on the starting position

Fig. 7. Matlab simulation of the precision alignment application. The Hamming distances are shown on the right and bottom side of the figures.

V. VLSI INTEGRATION OF THE HAMMING ANN

The system-level view of the two-dimensional Hamming distance comparator is depicted in Fig. 8. The Hamming core processing array is surrounded by several peripheral units which are devoted the tasks of generating working signals, and taking decisions on the base of the processing results. The offset cells located on each row are activated during evaluation in order to guarantee that the worse case row voltage be above the decision threshold level. The decision network consists of regular CMOS inverters and buffers.

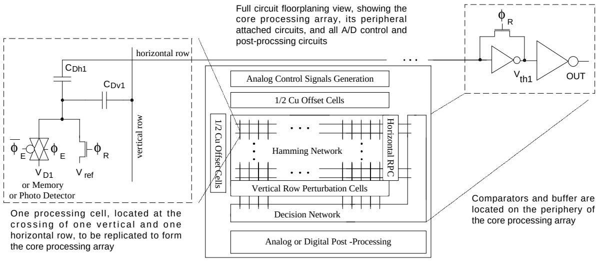
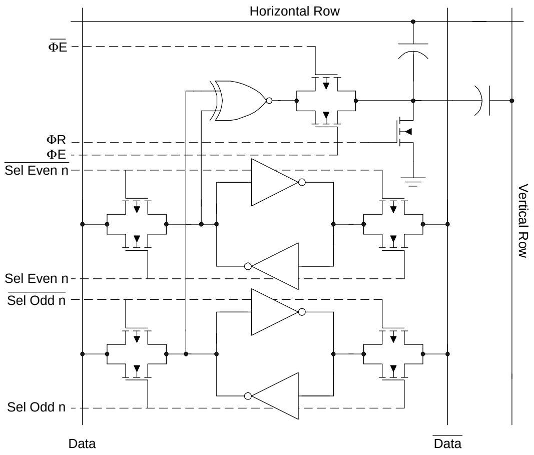

Full circuit floorplaning view, showing the core processing array, its peripheral attached circuits, and all A/D control and post-procssing circuits
horizontal row
C_{Dh1}
C_{Dv1}
vertical row
ϕ_E
ϕ_E
ϕ_R
V_{D1}
V_{ref}
or Memory
or Photo Detector
One processing cell, located at the crossing of one vertical and one horizontal row, to be replicated to form the core processing array
Analog Control Signals Generation
1/2 Cu Offset Cells
1/2 Cu Offset Cells
Hamming Network
Horizontal RPC
Vertical Row Perturbation Cells
Decision Network
Analog or Digital Post -Processing
ϕ_R
V_{th1}
OUT
Comparators and buffer are located on the periphery of the core processing array

Fig. 8. System-level view of the developed IC.

The integration of the proposed circuit architecture is produced using a 0.35µm double-polysilicon CMOS technology. The capacitances are realized as overlaps of two polysilicon layers. The Hamming array consists of a unit cell, to be abutted in the horizontal and vertical direction, thus creating the core Hamming network. The unit cell used to build a 16X16 array is depicted in Fig. 9. All control signals are routed in a single direction. The undesired coupling of the signal lines is envisioned to be cancelled in a future design using the third metal layer as a shield.

a)



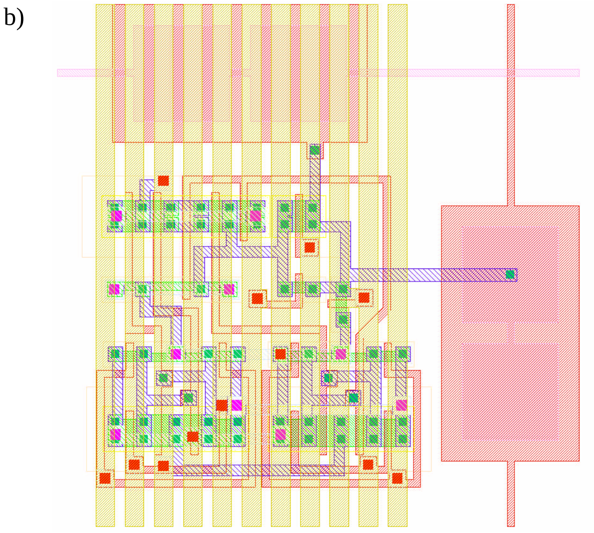


Fig. 9. Unit cell to be abutted horizontally and vertically in order to construct the Hamming array core; a) schematic and, b) layout views.

The size of the core array is 400 x 400μm2. The number of signals routed to I/O pads is 80, allowing easy direct and parallel access to most of the signal to be tested; the minimal number of pads allowing parallel access is in excess of five to the number of data I/0s. The core Hamming array is shown in Fig. 10.

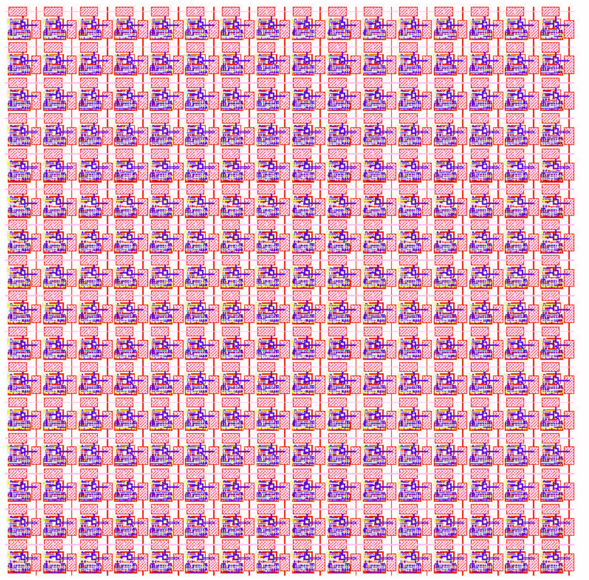


Fig. 10. Layout of the 16x16 array core.

CONCLUSION

The development of a very efficient and versatile Hamming ANN is shown in this paper. The main characteristics and originality stem from considering the WTA operation in the time domain which offers a wide range of new developments. This circuit finds its application as a dedicated hardware core for signal processing.

REFERENCES

[1] V. Beiu, Digital IC Implementations, *Handbook of Neural Computation*, E. Fiesler, R. Beale, Edts. Oxford University Publishing, 1997.

[2] E. A. Vittoz, Analog VLSI Implementation of Neural Networks, *Handbook of Neural Computation*, E. Fiesler, R. Beale, Edts. Oxford University Publishing, 1997.

[3] R. P. Lippmann, An Introduction to Computing with Neural Nets, IEEE ASSP Magazing, April 1987.

[4] U. Cilingiroglu, A Charge-Based Neural Hamming Classifier, IEEE Journal of Solid-State Circuits, Vol. 28, No. 1, January 1993.

[5] A. Schmid, Y. Leblebici and D. Mlynek, Hardware Realization of a Hamming Neural Network with On-Chip Learning, Proceedings of the 1998 IEEE International Symposium on Circuits and Systems ISCAS'98, Monterey, CA, 1998.

[6] A. Schmid, Y. Leblebici, D. Mlynek, Mixed analogue-digital artificial neural network architecture with on-chip learning, IEE Proc. Circ. Dev. and Syst., Vol. 146, No. 6, December 1999.

[7] H. Özdemir, A. Kepkep, B. Pamir, Y. Leblebici, U. Cilingiroglu, A Capacitive Threshold-Logic Gate, IEEE Journal of Solid-State Circuits, Vol. 31, No. 8, August 1996.

[8] Y. Leblebici, F. K. Gurkaynak, Modular Realization of Threshold Logic Gates for High-Performance Digital Signal Processing Applications, IEEE ASIC Conference, 1998.

[9] A. Schmid, Y. Leblebici, D. Mlynek, A Two-Stage Charge-Based Analog/Digital Neuron Circuit with Adjustable Weights, IJCNN, 1999.

[10] S. Badel, A. Schmid, Y. Leblebici, VLSI Realization of a Two-Dimensional Hamming Distance Comparator ANN for Image Processing Applications, ESANN, 2003.