Mixed analogue–digital artificial-neural-network architecture with on-chip learning

A.Schmid, Y.Leblebici and D.Mlynek

Abstract: The authors present a novel artificial-neural-network architecture with on-chip learning capability. The issue of straightforward design-flow integration of an autonomous unit is addressed with a mixed analogue–digital approach, by implementing a charge-based artificial neural network which interacts with digital control and processing units. The circuit architecture and design-flow approach for the case of a Hamming network performing pixel-pattern recognition are described.

1 Introduction

The ability of artificial neural networks (ANNs) to acquire knowledge of their surrounding environment and adapt to it, as well as their use of a high degree of computing parallelism, makes them very efficient in many application fields including process and quality control, consumer products, optical character and speech recognition, and complex forecasting tasks, among many others [1].

Silicon implementation of ANNs as an integrated circuit (IC) [2] aims at providing a final product with desirable low-area, low-power and low-cost properties. Several purely analogue ICs, most of them belonging to the charge-based or current-based families, have been developed to meet the criteria of minimal area and fast throughput. However, the main drawbacks of analogue systems include sensitivity to ambient noise and to temperature, as well as the lack in efficient automated synthesis methods and tools. On the other hand, purely digital realisations have the advantage of a limited but well defined precision that is given by the quantification of all neuron parameters. One main characteristic of purely digital realisations is their straightforward design-flow; some realisations start from a high-level-hardware-language description such as VHDL, to be synthesised into a standard-cell-based architecture or an FPGA. The extensive reuse of precharacterised modules, whether these be VHDL-based descriptions or mask layouts, is yet another possible solution to speeding up the IC development process.

Combining the advantages of both analogue and digital realisations into a novel mixed-mode architecture is the purpose of the implementation described in this paper. It focuses on developing a simple design-flow aiming at the integration of artificial neural networks with on-chip learning into an autonomous and easily reconfigurable integrated-circuit architecture. The silicon integration of a Hamming network [3] of 20 charge-based neurons, interacting with a purely digital unit to which the on-chip-learning and circuit control tasks are dedicated is shown.

2 Circuit architecture and its design-flow

2.1 Main building blocks

The overall circuit architecture is divided into two main parts with regard to their operating modes, i.e. analogue and digital. The analogue ANN unit executes the neural-function processing based on a charge-based circuit structure; it is composed of a 20-neuron layer, each with 10-bit vector inputs. The winner-take-all (WTA) [4] unit is devoted to the task of selecting one neuron as the winner on the criterion of best degree of matching between the stored pixel pattern and the current input vector. On the other hand, the error-correction unit (OLU), the circuit-control (CCU) and clock-generator (CGU) units perform purely digital operations (seeFig. 1).

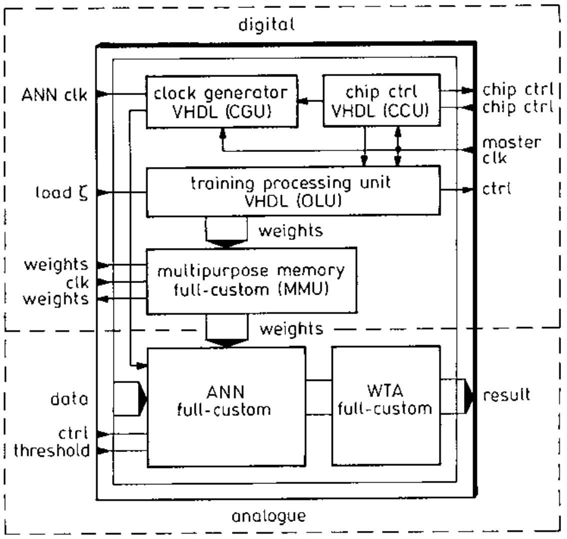

Fig. 1 *Block diagram of implemented integrated circuit*

This mixed analogue–digital architecture is consistent with the objectives of constructing a flexible, straightforward design-flow with reusability properties, and of

© IEE, 1999

IEE Proceedings online no. 19990685

DOI: 10.1049/ip-cds:19990685

Paper received 11th November 1998

A.Schmid and D.Mlynek are with the Integrated Systems Laboratory, Department of Electrical Engineering, Swiss Federal Institute of Technology (EPFL), CH-1015 Lausanne, Switzerland

Y.Leblebici is with the Analog/Digital Microelectronics Group, Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA, 01609, USA

addressing the issues of efficient and compact design. The ANN operates in the analogue domain and thus inherits most of the advantages associated with it, especially speed of neural function execution and compact design. All the digital parts, on the other hand, were designed, simulated and then synthesised from a VHDL high-level hardware language description. This design-flow significantly simplifies such issues as fast prototyping of a new algorithm into an IC, fast integration of the selected architecture, and easy layout floorplanning. A dedicated multipurpose memory unit (MMU) which has a scan-path architecture with parallel and serial read/write ability is devoted to the task of loading the initial weights and observing the new processed weights. This unit is an operating and test structure that together with others allow full testability and observability of the IC.

2.2 Control signals and data flow

The CCU is the master circuit controller; i.e. all other units are subordinated to this unit. Its tasks mainly include the synchronisation of all processing units among themselves and with the circuit supervisor, as well as the input/output protocol implementation. The control and dataflow are represented in Fig. 2. Note that the ANN and WTA are completely controlled by the CGU which has the ability of generating the clock signals ϕ_1 – ϕ_5, asynchronously from the circuit master clock.

The external dataflow consists of presenting new data on the START event; and sampling the result on the DONE event. The internal dataflow is simplified; no complex dataflow control structure is required as any new vector is processed immediately by the ANN. These data are lost at the end of a cycle since no internal framing is available.

2.3 Circuit environment

The developed architecture needs to interact with a supervisor to download several process-control and data signals in order to allow proper functioning. This global control system may either be a dedicated microcontroller, for an embedded microsystem, or a piece of control software driving a conventional microprocessor in a computer architecture. This requirement obviously reduces the autonomy of the overall circuit architecture and assumes that it has to be included in a complete system such as a computer board. Nevertheless, the ability to modify some algorithmic parameters and decision criteria in real time significantly improves the efficiency of the system.

For example, the learning rate parameter η has a significant influence on the ANN convergence and on its ability to properly acquire knowledge. As stated in Section 4, its value may be downloaded into the IC at any time. A signal indicating whether or not error correction was applied during the last cycle is sent to the supervisor in order to keep the decision of accepting or rejecting the convergence condition outside the chip. Following the same idea, the selection of the neuron to be trained is also carried out by the supervisor. The threshold value also has to be produced externally in the form of an analogue voltage V_θ.

All of these features could easily be integrated into a fully autonomous version of the developed architecture which, however, would result in loss of flexibility due to the impossibility of modifying any parameters.

3 Mixed analogue–digital ANN architecture and operation

3.1 ANN circuit architecture

A Hamming network is a two layer feedforward ANN with the ability to classify noise-corrupted patterns. Its internal architecture consists of a first layer of neurons performing in parallel the Hamming distance of a m-bit digital input vector with n previously stored exemplar patterns — this is the quantifier subnet; the second layer is devoted to the selection of the winner neuron which is that with the smallest Hamming distance to the input vector (see Fig. 3) — this is the discriminator subnet. This network performs efficient classification for relatively low complexity, and always converges to one of the previously stored combinations.

The number of independent neurons n corresponds to the number of patterns to be sorted out, and the number of synapses m associated with each neuron corresponds to the number of input-vector components.

For the realisation of the Hamming network, a modified version of the charge-based circuit architecture first presented in [5] is used; this was originally designed with fixed weights. In particular, the circuit architecture was modified to allow simple programming of the input weights. Since this paper is primarily focused on the overall system architecture, a detailed analysis of the charge-based quantifier and discriminator subnets is not presented here. The fundamental circuit architecture of the capacitive Hamming network is essentially identical to the fixed-weight classifier circuit published earlier, the operation and limitations of which were well documented in [5]. It has also been experimentally demonstrated earlier that charge-based circuit architectures offer the advantages of high integration density, high speed and low power dissipation, while sensitivity

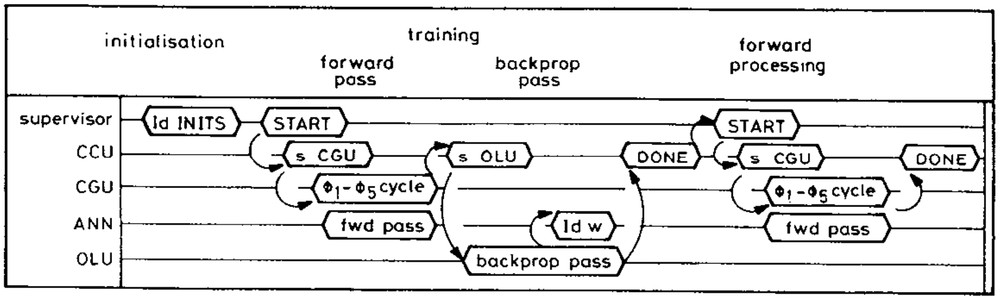

Fig.2 *Control signals and dataflow between main blocks*
Consquences of signals are as follows:
ld_INITS – IC initialisation sequence. IC in wait mode until START
START – a new vector is available for processing. One forward processing pass is allowed
s_CGU – makes the CGU produce one cycle of ϕ_1–ϕ_5 signal clocks
s_OLU – makes the OLU start one error-correction cycle
ld_w – loads the new processed weights into the ANN
DONE – the launched process has finished. When in forward-processing mode: a new result is available

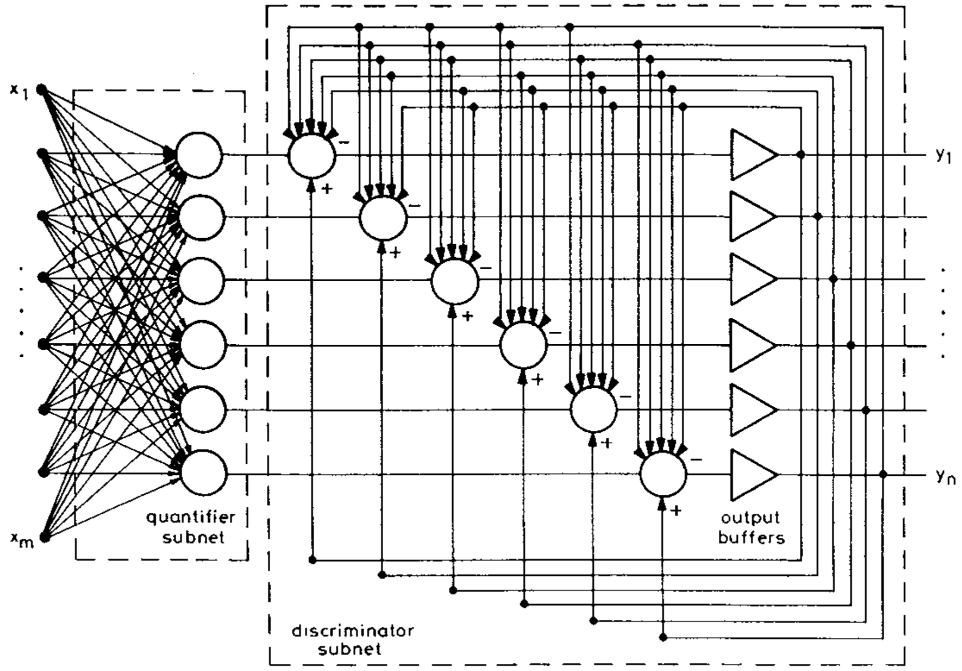

Fig. 3 *General structure and functional description of a Hamming network*

limitations (discriminator offset) that may stem from circuit/device mismatch still allow a relatively large input vector size [5, 6]. For a detailed description and electrical analysis of the charge-based capacitive Hamming network architecture, the reader is referred to [5].

Each charge-based synapse is composed of four binary weighted capacitors as well as four memory latches to support programmability of the device. The capacitor values associated with each synapse are chosen as $C_i = 2^n C_u$, where $n = 0, \ldots, 3$ and C_u is unit capacitance. Thus the modified configurable circuit architecture of the charge-based Hamming network allows four-bit weight programming.

3.2 ANN circuit operation

The circuit operates in two distinct modes (forward-processing mode and training mode) which can be selected by an external triggering signal. The circuit operates in the following sequence when in forward-processing mode (also called recall mode).

- Initialisation phase: the initial weights (or newly processed weights) are downloaded into the internal synaptic memory.

- Quantification phase: the input vector is applied. Depending on the programmed weight values, all dendritic voltages in the ANN structure assume their new level.

- Discrimination phase: the WTA processes to winner selection. The result may be sampled when convergence of the WTA is reached.

The circuit controller flags the availability of a new maximum-likelihood-classification result. All these steps repeat every time a new forward processing pass is required under the control of the circuit-supervisor unit.

The circuit has to be trained in order to acquire experience of the patterns to be sorted out. This happens during the training mode which is divided into two passes: one forward pass and one backpropagation pass (error correction pass).

- Forward pass: the training forward pass is identical to the normal forward-processing mode, with the exception that only one neuron is activated at a time (each neuron is trained separately). The neuron to be trained is selected by the supervisor and is activated through a forward-processing pass using the training pattern, while all other neurons are kept in idle mode to prevent undesired interaction.

- Back-propagation pass: given the current input vector, the current processing weights and the binary result of the forward pass, the circuit controller activates the OLU, the digital unit which computes the weight values the ANN will have in the next cycle, to process the learning algorithm. The OLU computes the new weights to be downloaded into the synaptic memory. The circuit controller flags the end of the cycle to the outside, and indicates whether or not error correction was to be applied during the current training pass. The circuit supervisor may then decide on the necessity of a refining training pass with the same or another threshold value, or to train another neuron because convergence was satisfactorily achieved.

4 Learning algorithm and algorithmic considerations

Hardware implementations of ANNs are typically subject to restrictions in terms of area, power and time which may complicate the realisation of a chosen learning algorithm. The so-called hardware-friendly algorithms [7] are intended to yield a simple hardware realisation, yet also achieve a high degree of efficiency despite limited precision of computation, approximation of the implied functions, and perturbing effects of quantisation.

The training algorithm was chosen as a hardware-friendly adaptation (eqn. 2) of the error-correction learning algorithm (eqn. 1) [8]:

$$w(n+1) = w(n) + \eta\{d(n) - y(n)\}x(n) \qquad (1)$$

Here $\boldsymbol{w}$ stands for the weight vector, $\boldsymbol{x}$ for the input vector, d is the expected output and y the actual neuron output result, η is the learning rate, n is the time increment.

$$w(n+1) = w(n) \pm \zeta$$

where

$$\zeta = \eta[d(n) - y(n)]x(n) \text{ if } x_j = 1$$

or

$$\zeta = 0 \text{ if } x_j = 0 \qquad (2)$$

The use of a WTA unit restricts the input vector x to be purely binary; thus all of its components belong to the binary set {0, 1}. This fact, together with the hard-limiting activation function in the ANN, produces a purely binary result to the $\{d(n) - y(n)\}x(n)$ operation. Hence, the influence of the η parameter is enhanced as it remains the only nonbinary parameter to be multiplied with one of the logical values {0, 1}. Thus the system was designed so as to allow the ζ value to be changed at any time by the supervisor controller.

Prior to the design of the unit, C simulations were run to validate the hardware-oriented algorithms. A specific simulation tool was developed in order to produce a realistic high-level characterisation, which is based on the model of a neuron that optimally reproduces the analogue behaviour of the real implementation in the integer domain. The simulations were run on a network consisting of nine neurons with 9-bit vectors to classify. The small size of the network does not in any way affect the quality of the results; expanding the network to a larger one would result in a longer delay to reach convergence (in a general case). The training set and simulation parameters can be seen in Fig. 4.

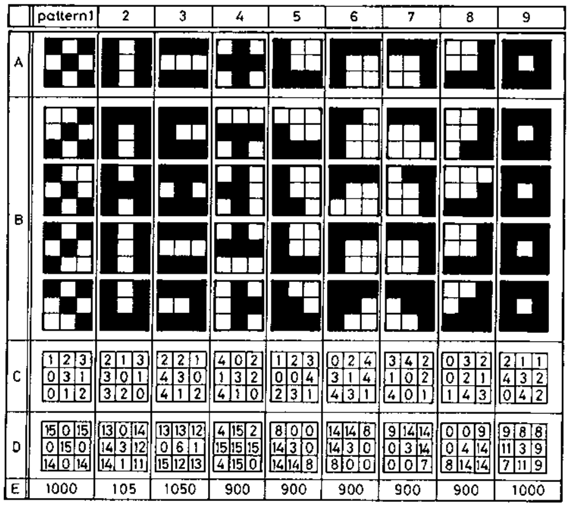

Fig. 4 *Simulation of the implemented algorithm for a pattern-recognition/classification example*
A – noiseless patterns to be recognised: pattern 1 to pattern 9
B – noisy patterns
C – initial weights
D – resulting weights
E – threshold θ

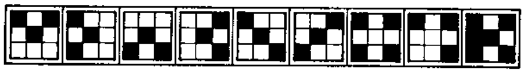

Fig. 5 *Pattern 1 perturbed by noise*
All these patterns were correctly classified

Some simulations were run to test the behaviour of the network when confronted with unknown patterns, which highlighted the efficiency of the network in generalising (see Fig. 5).

5 Realisation of the ANN integrated circuit

A test chip implementing the developed architecture was designed and realised in AMS (Austria Micro Systems) CMOS 0.8µ 2-poly technology [9, 10]. The layout can be seen in Fig. 6. The die size is less than 13mm^2; the functional modules (ANN, WTA, CCU, CGU and OLU) occupy less than 5mm^2. The number of pins is 100, several pins being attributed to additional test structures.

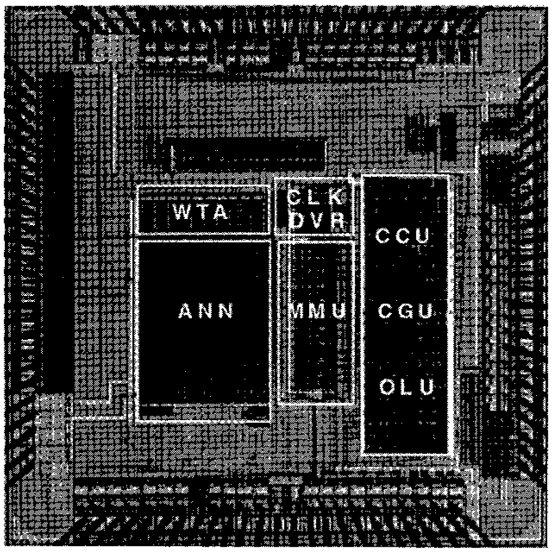

Fig. 6 *Microphotograph of realised integrated circuit*
All operative units occupy an area of less than 5mm^2; the overall die size is less than 13mm^2, with a 100 PGA package. Several test structures and test pins were implemented to allow easy testability of the chip

The test features were integrated into the design so as to make each main building element testable independently from all others. As mentioned above, the multipurpose memory unit is fully accessible in read/write mode, which allows the weights to be read to be loaded by the ANN or the OLU, or to be downloaded to check the computation of the OLU. The binary result of the WTA output in forward processing mode (is current presented pattern recognised as being that stored in current trained neuron?) is also fully accessible in read/write mode, which allows for testability of WTA and control of the OLU in test mode. The WTA outputs are all connected to output pins which ensures full testability over the ANN and WTA. The reason for observing all the WTA outputs lies in the internal operation of the WTA that may produce a multiple winner selection. The ANN and WTA can be tested independently; in this test mode all the driving-clock signals are provided by external means via the CLK_DVR unit. Finally, one single neuron with full external access was integrated to allow sensitivity and speed tests.

All of the major modules on chip were tested separately to confirm their functionality. The digital error-correction unit (OLU), circuit-control unit (CCU), multipurpose memory unit (MMU) and the clock-generator unit (CGU) were tested using the HP82000 testing environment and were found to be fully functional. Measurements were also performed to verify the operation of the analogue ANN and WTA modules. The WTA was found to operate correctly for all cases with a minimum Hamming distance of two bits or more. Discrimination of a winner neuron was found to become problematic in cases where the minimum Hamming distance was only one bit, which indicates that the unit weight capacitance of 17fF actually remains below the limit value dictated by the process-dependent quantifier offset voltage. Extensive measurements for a complete characterisation of the entire ANN architecture are continuing.

The main effort was not put on developing a high-speed architecture. Nevertheless, a speed of 4 × 10^6 inferences per second is expected in forward processing mode with an external control. Internal control processing is limited by the slowest clock signal to be produced by the CGU, by the circuit master clock and by the control path, which makes it difficult to evaluate. A realistic estimation gives an expected speed of 100 000 inferences per second with a master circuit clock reaching 10MHz and an ANN driving clock reaching 1MHz.

6 Conclusions

The integration of a novel artificial neural network architecture has been demonstrated. The proposed mixed analogue–digital realisation is based on an analogue ANN block which interacts with a purely digital learning unit, implementing the error-correction learning algorithm, as well as the circuit-control part. The ANN is a Hamming network including a first layer of charge-based neurons driving a WTA unit.

A test chip containing 20 neurons of 10 synapses each has been designed using an AMS CMOS 0.8 2-poly technology. It has an active area of less than 5mm^2 for a die size of 13mm^2.

The general idea in this development was to establish a valid design-flow for an ANN-based integrated circuit to be reusable in some other applications, rather than focus on integrating a high-throughput processing unit.

The mixed analogue–digital architecture presented in this work can be used in applications where the main focus is the on-chip learning ability of the ANN rather than a high processing/inference capability. This includes all autonomous systems with relatively slow time constants but a very long lifetime. Possible applications may be found in medical engineering, automotive engineering and consumer products.

7 References

1 KAPPEN, H.J.: 'An overview of neural network applications'. Proceedings of the 6th international congress for *Computer technology in agriculture*, Wageningen, The Netherlands, 1996, pp. 75–79

2 FAKHRAIE, S.M.: 'VLSI-compatible implementations for artificial neural networks' (Kluwer Academic Publishers, Dordrecht, 1997)

3 LIPPMANN, P.R.: 'An introduction to computing with neural nets', *IEEE ASSP Mag.*, April 1987, pp. 4–20

4 GÜNAY, Z.S., and SÁNCHEZ-SINENCIO, E.: 'CMOS winner-take-all circuits: a detail comparison'. Proceedings of 1997 international symposium on *Circuits and systems*, ISCAS'97, Hong Kong, June 1997, pp. 41–44

5 ÇILINGIROĞLU, U.: 'A charge-based neural Hamming classifier', *IEEE J. Solid-State Circuits*, 1993, **28**, (1), pp. 59–67

6 ÖZDEMIR, H., KEPKEP, A., PAMIR, B., LEBLEBICI, Y., and ÇILINGIROĞLU, U.: 'A capacitive threshold-logic gate', *IEEE J. Solid-State Circuits*, 1996, **31**, (8), pp. 1141–1150

7 MOERLAND, P., and FIESLER, E.: 'Neural network adaptations to hardware implementations' *in* FIESLER, E., and BEALE, R. (Eds.): 'Handbook of neural computation' (Institute of Physics Publishing/Oxford University Publishing, New York, 1996), E1.2–1.13

8 HAYKIN, S.: 'Neural networks: a comprehensive foundation' (Macmillan College Publishing Co., New York, 1994)

9 SCHMID, A., LEBLEBICI, Y., and MLYNEK, D.: 'A charge-based artificial neural network with on-chip learning ability'. Proceedings of 5th European congress on *Intelligent techniques and soft computing*, EUFIT'97, Aachen, 1997, pp. 250–254

10 SCHMID, A., LEBLEBICI, Y., and MLYNEK, D.: 'Hardware realization of a Hamming neural network with on-chip learning'. Proceedings of 1998 IEEE international symposium on *Circuits and systems*, ISCAS'98, Monterey, USA, 1998, pp. 191–194