Analog Implementation of a Kohonen Map with On-Chip Learning

Damien Macq, Michel Verleysen, Paul Jespers, *Fellow, IEEE,* and Jean-Didier Legat

***Abstract*—Kohonen maps are self-organizing neural networks that classify and quantify n-dimensional data into a one- or two-dimensional array of neurons [1]–[2]. Most applications of Kohonen maps use simulations on conventional computers, eventually coupled to hardware accelerators or dedicated neural computers. The small number of different operations involved in the combined learning and classification process makes however the Kohonen model particularly suited to a dedicated VLSI implementation, taking full advantage of the parallelism and speed that can be obtained on the chip. We propose here a fully analog implementation of a one-dimensional Kohonen map, with on-chip learning and refreshment of on-chip analog synaptic weights. The small number of transistors in each cell allows a high degree of parallelism in the operations, what greatly improves the computation speed compared to other implementations. This paper will emphasize on the storage of analog synaptic weights, based on the principle of current copiers; it will be shown that this technique can be used successfully for the realization of VLSI Kohonen maps.**

I. Kohonen Neural Network

A Kohonen map is a topological array of neurons where each element is connected to its nearest neighbors, and to common input vector (Fig. 1). The network evolves in the following way. The input vector is simultaneously compared to the synaptic vectors of all neurons in the map, with a predefined distance measure. The neuron whose synaptic weight vector is closest from the input vector is selected, and the weight vectors of the neurons topologically close from this winner in the map are moved in the direction of the input vector. A symmetric function centered around the winner is used as multiplying factor during the learning, to strengthen the adaptation of weights from neurons topologically close from the winner, and to reduce it as the distance increases. The adaptation law is

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)\Lambda_i(t)(x_j(t) - w_{ij}(t)) \quad \forall i = 1, \ldots, N \tag{1}$$

where $\eta(t)$ is the adaptation factor, and $\Lambda_i(t)$ a symmetric function centered around vector μ (winner) allowing a greater adaptation of neurons close from μ and a weaker adaptation of neurons far from μ. Rule (1) is known as the Kohonen rule, and is often used with constant $\Lambda(t)$ function in a finite neighborhood, or with a triangular function with constant slope [1]–[2].

Manuscript received July 15, 1992; revised August 24, 1992.
The authors are with the Université Catholique de Louvain—Microelectronics Laboratory, B-1348 Louvain-la-Neuve, Belgium.
IEEE Log Number 9206626.

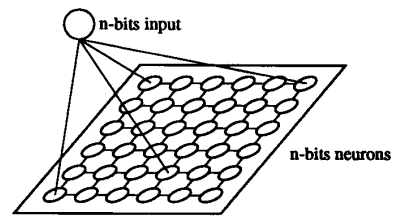

Fig. 1. Kohonen bi-dimensional map.

When a set of input vectors is successfully presented to the network, two processes simultaneously occur. First, because of the adaptation of weights in a neighborhood of the winning neuron, synaptic weights of topologically close neurons in the map will converge to close vectors with respect to a predefined distance measure. The network will thus establish a correspondence between close stimuli (input vectors) in the input space and close neurons in the map: it organizes the data. Secondly, because the number of neurons is generally much lower than the number of stimuli, the network quantifies the space, each neuron corresponding to a subset of input vectors. If the neighborhood function and the adaptation factor are adequately decreased with time, auto-organization mainly occurs during the first iterations of the algorithm, while precise quantification is preponderant during the last iterations.

Any distance measure may be used for the selection of the winner; this will of course influence the above described topological correspondence between the input space and the Kohonen map, but the global process remains similar. There is no *a priori* reason to prefer the Euclidean measure to other ones without information on the stimuli distribution; other distance measures are then considered, since the Euclidean one is difficult to implement with digital or analog VLSI. Dot product is more easy to implement, but is valid as a distance measure only for normalized patterns. Manhattan distance is then chosen, since it remains valid for nonorthogonal patterns and is easy to implement.

II. Analog Implementation

The principle of the proposed analog implementation of the Kohonen map is based on the use of currents to represent the values (components of the vectors and synaptic weights). To select the winner as described in Section I, each component

1045–9227/93$03.00 © 1993 IEEE

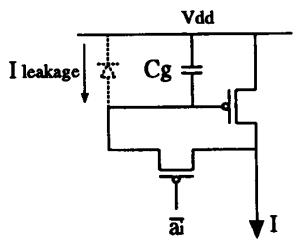

Fig. 2. Synaptic weight.

of the input vector is subtracted from the corresponding component of the synaptic weights stored in the network. Each component of the synaptic weights must thus be stored as a current in the circuit to be easily subtracted from an input current: this is detailed in the next section.

A. Storage of Weights

1) Principle The storage of the synaptic weights is based on the principle of the current copier [3] (see Fig. 2). A current copier cell contains one transistor with a switch connected between his gate and his drain. When the switch is on, the gate voltage adapts itself to the current to be memorized; when the switch cuts off, the gate voltage is stored on the gate capacitance of the copier-transistor and its drain current remains unchanged. Synaptic weights are so stored on the C_g capacitors of Fig. 2. However, leakage currents continuously modify the value stored on the capacitors. Refreshment must then be achieved, to keep a reasonable precision on these values.

Several errors affects the correct operation of the current copier cell. If these errors are deemed to have always the same direction, the principle of the refreshment systems is to replace the measured analog value on the capacitor by the next upper reference value in the n-bit grading.

The charge injection of the switch (channel and overlap capacitance) always discharges the gate capacitance C_g if the switch and the copier-transistor are both PMOS (or both NMOS) transistors. The error due to the switch channel charge injection can be expressed as

$$\Delta I = g_m \frac{Q_{\text{inj}}}{C_g} \tag{2}$$

where g_m and C_g are the transconductance and the gate capacitance of the copier-transistor, and Q_{inj} is the channel charge injected from the switch. Equation (2) can be expressed as function of the stored voltage:

$$\Delta I = -g_m k \frac{C_{sw}}{C_g} \left(-V_{\text{on}} + nV_{T0} + nV_g\right) \tag{3}$$

where V_{on} and V_{T0} are respectively the "on" and threshold voltage of the switch, V_g is the stored voltage, k represents the proportion of the channel charge injected on C_g, and C_{sw} is the switch capacitance. The SPICE simulation of this error is represented in Fig. 3 as function of the input current for a regulated cascode with a nonsaturated copier transistor.

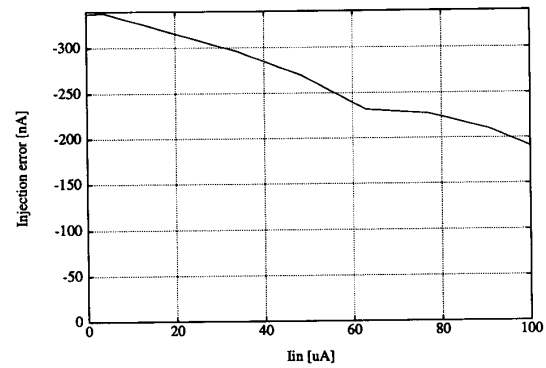

Fig. 3. Switch injection error on stored circuit.

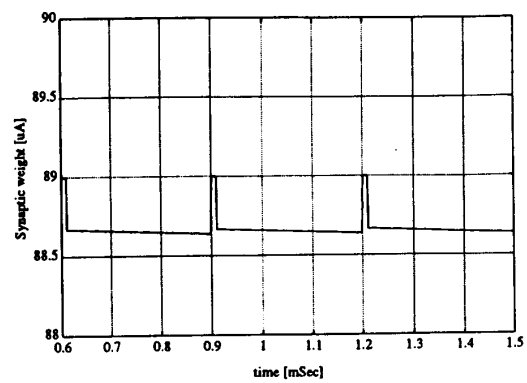

Fig. 4. Stored current with refreshments.

The leakage current is mainly due to the reverse-biased junction, represented by the diode in Fig. 2. It always discharges the gate capacitance of the copier-transistor. Other leakage currents (subthreshold channel conduction ...) can be neglected if the "off" voltage on the gate of the switch is low with respect to V_{T0}. At room temperature, leakage currents are very small ($< 1pA$). The stored current as function of time is represented at Fig. 4, with refreshment cycles to the next upper reference current.

Finally, the errors due to the finite output conductance and to the capacitive coupling between the gate and the drain of the copier-transistor must be minimized to be negligible with respect to the above-mentioned errors, by using long copier-transistors or, preferably, more elaborate memory cells (i.e., the regulated cascode current copier [4]).

The refreshment circuit acts as follows: all cells are periodically refreshed by a converter, which cancels the effect of leakage currents by reading the current memorized in a memory cell and writing the next upper reference current in the same cell.

A multilevel memory circuit has been fully tested [5], and a resolution of 7 bits per memory cell was measured. The cells are arranged in an array of 14 × 15 elements. Dimensions of the circuit, including the refreshment system,

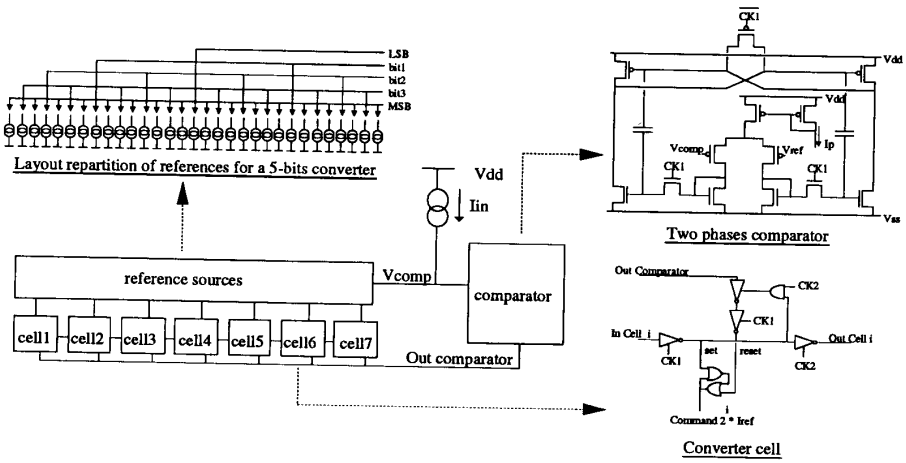

Fig. 5. Layout of the converter.

are 1.7 mm×1.7 mm in a conservative 2.4 μm CMOS process. With a more efficient process ($\leq 1\,\mu$ CMOS process), this corresponds to a density of ± 500 memory cells per square millimeter.

The memory cells only contain a simple current copier (size of the copier-transistor: $W = 90\,\mu$A, $L = 40\,\mu$A) and some transistors to address it. We saw that the memory cell is refreshed through lecture of the current and writing after adjustment to the next upper reference current: the stored current (I_{in} on Fig. 5) is compared to a set of reference currents according to the successive approximation scheme (comparison of the current I_{in} with $2^N/2$ references sources (I_{lsb}); if I_{in} is greater than the references, comparison of I_{in} with $2^N/2+2^N/4$ sources I_{lsb} and MSB $= 1$, else comparison with only $2^N/4$ sources I_{lsb} and MSB $= 0, \ldots$). After N comparisons, the N-bits digital value is obtained, and the converter keeps switched on or off the references sources corresponding to these N bits (ΣI_{lsb}). The sum of these reference currents will thus be the next lower reference current of the current coming from the copier cell ($\Sigma I_{\text{lsb}} < I_{\text{in}}$). A current equal to $(\Sigma I_{\text{lsb}}) + I_{\text{lsb}}$. is then written again in the copier cell.

The converter (see Fig. 5) uses two clock phases CK1 and CK2. During CK1, the 2^i reference sources corresponding to bit i are switched on, while the 2^i reference sources corresponding to the next upper bit $i+1$ are left switched on or off depending on the state of the comparator after the previous CK2 cycle.

During CK2, the set of switched on references sources ΣI_{lsb} is compared to the memorized current I_{in}; the result of this comparison is sampled on CK2 and will be used during CK1 of the next cycle.

Seven identical cells implemented in dynamic logic are connected. The cells are successively activated through a token ring architecture: The token comes in cell i when the input "In Cell i" is going to "0" (the reset mechanism injecting the token in the first cell and initializing correctly the others before each conversion is not represented here). These seven cells drive 127 matched cascode current sources (I_{ref}). Matching or sources I_{ref} is critical; a mismatching greater than 1 LSB on the i references (2^i reference sources in parallel) may cause nonmonotonicity. The implementation of these sources (linear distribution) is so that the influence of the variation of oxide thickness is maximally reduced (see Fig. 5: example of distribution for 31 sources).

The comparator is shown in Fig. 5.; a first phase CK1 allows to reset the comparator. The data are sampled at the end of CK1. During the reset (CK1), the flip-flop included in the comparator is in a metastable mode and the current difference on node V_{comp} increases or decreases the voltage on that node, fixing the state of the comparator at the end of CK1. V_{comp} and V_{ref} are connected during CK2, and the flip-flop switched to one of the two stable states.

2) Measurements The converter was first tested by connecting its input to an external current. The voltage at the input node is represented at Fig. 6. This node goes high (up to a limit of $V_{\text{ref}} + 0.7$ V to keep the current copier cell in saturated mode) or low depending on the result of the successive approximation algorithm. The dc characteristic and the conversion error of the converter are represented in Figs. 7 and 8. The reference current was fixed to 0.8 μA, VDD to 5 V and the reference voltage of the comparator (V_{ref} at Fig. 5) was 3 V.

The number of memory cells that can be refreshed with one converter can be approximated with the following equation

$$\#\,\text{cells} < \frac{1}{2^N}\,\frac{\nu}{K}\,\frac{V_g C_g}{I_{\text{leak}}} \tag{4}$$

where N is the resolution of the memory, ν is the clock frequency, K is the number of clock cycles needed for the refreshment of one cell ($K = 9$ in our design: one for addressing, seven for the A/D conversion and one for writing), V_g and C_g are the gate voltage and capacitance of the copier-transistor, and I_{leak} is the leakage current.

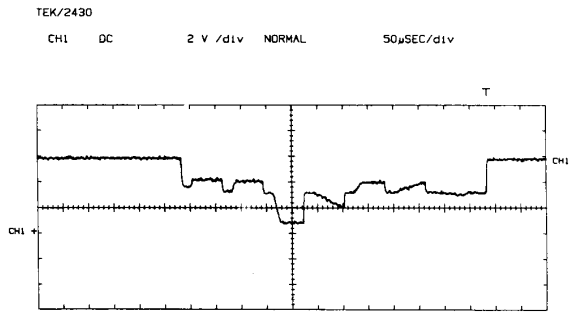

Fig. 6. Input node of the converter.

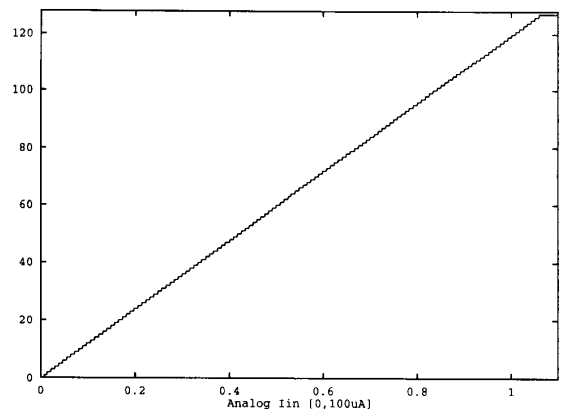

Fig. 7. DC characteristic of the converter.

The memory map was tested for the whole current dynamics (0–100 μA) with a clock frequency of 100 KHz, which gives a refreshment period of 18.9 ms for one memory cell. The stored voltage was digitally read at the output of the SAR converter. A resolution of seven bits was measured on the whole dynamics. Nevertheless, it must be mentioned that a more elaborate current copier [4] will present a better error immunity that the simple cell implemented here: a high sensitivity to the reference voltage of the comparator (V_{ref} at Fig. 5) was observed (if V_{ref} is less than 3 V, the memory does not work anymore). This sensitivity is due to the gate to drain coupling of the simple copier cell.

The measured leakage current is 0.97 fA at room temperature (25°C). This corresponds to a maximal number of 130 000 memory cells (see (4) with the numerical values given above); this number may be subject to changes because of the large variations of the leakage currents with the temperature [6] and with the process.

The chip performances are summarized in Table I.

A microphotograph of the chip fabricated by MIETEC ALCATEL is shown in Fig. 9; Fig. 10 gives the floor plan of the circuit. The memory cell occupies 78% of the total area.

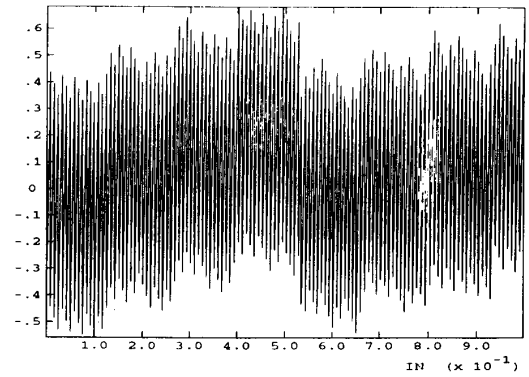

Fig. 8. Conversion error (INL + quantization) in LSBs.

TABLE I
PERFORMANCES OF THE REFRESHING SYSTEM

Power supply:	0–5 V
Refreshment cycle: reading-conversion-writing:	90 μs
Clock frequency:	100 KHz
Reference current I_{lsb}:	0.8 μA
Resolution:	7 bits
Time between 2 refreshment for one cell:	18.9 ms
Leakage currents at room temperature (25°C):	0.97 fA
Power dissipation (max current stored in all cells):	25 mW

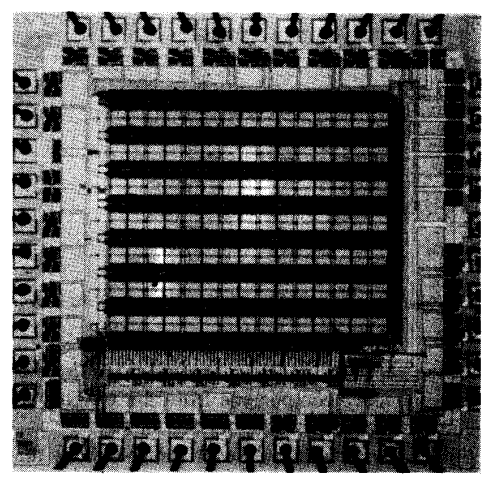

Fig. 9. Chip microphotograph.

B. Adaptation of Weights

Adaptation of the weights is also based on the principle of the current copier [3] (see Fig. 11 for the implementation of a Kohonen map with regulated cascode current copiers). When the P-type transistor T1 in each synapse is diode-connected, the value on its gate capacitor is adapted with a first-order time constant $T = C_g/g_m$, where C_g is the gate capacitance of the transistor and g_m its transconductance. Since the current

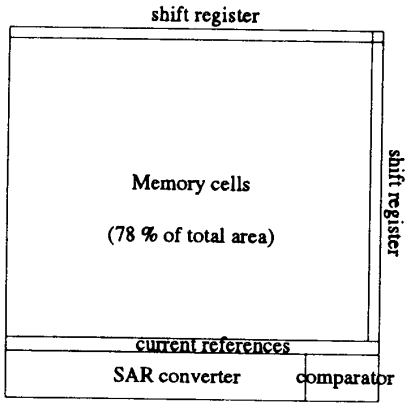

Fig. 10. Floor plan.

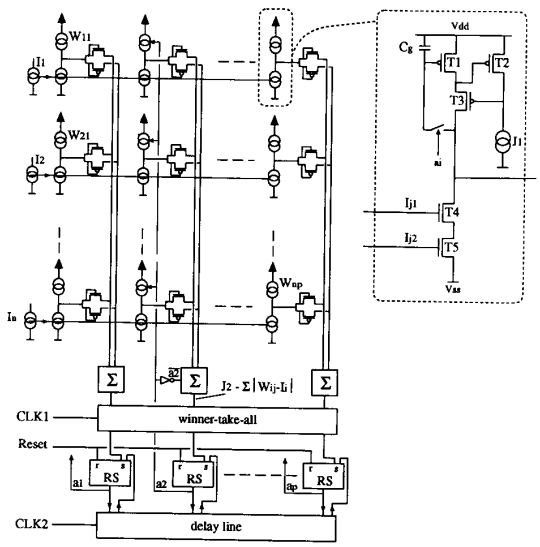

Fig. 11. Analog implementation of a one-dimensional Kohonen map.

present at the capacitor node is proportional to the difference of the corresponding components from the weight and input vectors, adaptation occurs as described in (1).

The time delay before the Reset signal defines the adaptation time of the synaptic weights connected to the neuron selected by the winner-take-all (see Fig. 12). Adaptation of the neighboring synaptic columns is delayed proportionately to the period of CLK2, each delay cell corresponding to one cycle of this clock. The learning rate factor η is thus fixed (at first order) by delay T_0 and time constant $T = C_g/g_m$ (for $T_0 \ll T$):

$$\eta \cong \frac{C_g}{g_m T_0}. \tag{5}$$

η can be reduced with time by adjusting the Reset signal. The neighborhood function $\Lambda_i(t)$ is a symmetric triangular function

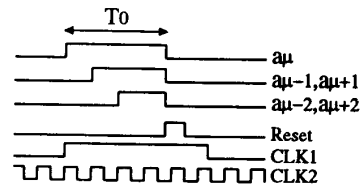

Fig. 12. Timing of the circuit.

whose value is 1 at the origin and whose slope is fixed by the frequency of CLK2:

$$\Lambda_i(t) = \left(1 - \frac{d(i, \mu)\, T_{\mathrm{CLK2}}}{T_0}\right) \tag{6}$$

where $d(i, \mu)$ is the distance (measured in number of cells) between neuron i and winner μ, and T_{CLK2} the period of CLK2. On-chip learning can be adjusted by means of these two parameters.

III. Conclusion

This paper described a fully analog implementation of a one-dimensional Kohonen map. Only 9 transistors are necessary in each synapse, plus a total of 51 transistors for each neuron (without the weight refreshing system). This small number of transistors allows the integration of about 35 7-bit neurons per square millimeter in a 2-μm standard technology, a sufficient density to integrate networks with hundreds of neurons on a single chip. The learning is fully implemented on the chip, the learning factor and the neighborhood function being externally controlled. Memorization of the synaptic weights by storage of analog values on capacitors and dynamic refreshment enhances this high density of integration. Although the presented chip is designed for a one-dimensional Kohonen map, the principle can be easily extended to two-dimensional maps, by adding supplementary connections to the delay line.

References

[1] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, no. 43, pp. 59–69, 1982.
[2] T. Kohonen, *Self-Organization and Associative Memory,* Berlin, Germany: Springer-Verlag, 1984.
[3] S. J. Daubert, D. Vallencourt, and Y. P. Tsividis, "Current Copier cells," *Electron. Lett.*, vol. 24, pp. 1560–1562, Dec. 1988.
[4] C. Toumazou, J. B. Hugues, and D. M. Pattullo, "Regulated cascode switched-current memory cell," *Electron. Lett.*, vol. 26, pp. 303–305, Mar. 1990.
[5] D. Macq, J. D. Legat, and P. G. A. Jespers, "Analog storage of adjustable weights," in *Proc. SPIE Conf. Applications of Neural Networks III,* Orlando FL, Apr. 20–24, 1992.
[6] B. Schwartz, E. Howard, and E. Hubbard, "Programmable analog neural network chip," *J. Solid-State Circuits,* vol. 24, no. 2, pp. 313–319, Apr. 1989.

Damien Macq was born in Leuven, Belgium on March 30, 1966. He received engineering degrees from the Université Catholique de Louvain, Louvain-La-Neuve, Belgium in 1989. He is currently working toward the Ph.D. degree at the Microelectronic Laboratory of Louvain-La Neuve.

His current interests are design and simulation of current-mode analogue custom integrated circuits in the area of artificial neural networks and A/D cyclic converters.

Michel Verleysen was born in Brussels on October 27, 1965. He received the engineering degree from the Université Catholique de Louvain, Louvain-La-Neuve, Belgium in 1987. From 1987 to 1990, he received a Belgian IRSIA grant to study towards the Ph.D. degree at the Microelectronics laboratory of the same university. He was awarded the Ph.D. degree in 1992, on the subject of VLSI design of neural networks.

Since 1987, his current interests have been analog and digital VLSI design, artificial neural networks (theory and VLSI implementations), signal processing, and pattern recognition. In the first half of 1992, he was invited professor at the Ecole Polytechnique Fédérale de Lausanne (Switzerland). He was local manager for the UCL of the ESPRIT-BRA project NERVES (1989–1991), and of the ESPRIT-BRA project ELENA (1992-1995). He is author of about 25 publications in the field.

Paul Jespers (M'60–SM'65–F'82) received the M.S. degree from the Université de Bruxelles (Belgium) in 1953, and the Ph.D. degree from the Université Catholique de Louvain (Belgium) in 1959.

He was member of board of Directors of the IEEE (1971–1972) and has been chairman of the Steering Committee of ESSCIRC-ESSDERC (1989–1990). In 1957–1958, he was guest at MIT (USA), where he worked on the statistical theory of communications. In 1967–1968, he was visiting professor at Stanford University, CA, (microelectronics). He is full professor at the microelectronics department of the Université Catholique de Louvain, and is teaching microelectronics, electronics, and systems. His current interests are design of MOS IC's (analog and digital), A/D converters, and neural networks. He is author or co-author of more than 100 publications in the field of microelectronics.

Jean-Didier Legat was born in Brussels, Belgium on April 15, 1957. He received engineering degrees from the Université Catholique de Louvain, Louvain-La-Neuve, Belgium in 1981. From 1981 to September 1984, he was granted a Belgian fellowship to undertake research in the field of document processing at the Microelectronics laboratory at the Université Catholique de Louvain.

At the end of 1984, he joined the Belgian Company Anfima with the proposal of creating a new company which would design and commercialize integrated devices for image processing. He was finally awarded his Ph.D. in February 1987 and in April of the same year a company called Image Recognition Integrated Systems (I.R.I.S.) was formed. Jean-Didier Legat was co-founder of I.R.I.S. and was Vice-President. In October 1990, he came back to the Microelectronics Laboratory of Louvain-La-Neuve. He is presently an invited professor. His current interests are design and simulation of analogue/digital custom integrated circuits in the areas of artificial neural networks, digital signal processing, computer vision and pattern recognition.