High Speed Field Programmable Analog Array Architecture Design

Edmund Pierzchala, Student Member, IEEE †‡
Marek A. Perkowski, Member, IEEE ‡

†Analogix Corporation
2515 S.E. Harrison St., Milwaukie, OR 97222
(503) 653–6827

‡Portland State University
Department of Electrical Engineering
P.O. Box 751, Portland, Oregon, 97207–0751
tel: (503) 725–3823
e–mail: edmundp@ee.pdx.edu
mperkows@ee.pdx.edu

In this paper we address the problem of the best architecture for high speed (high-frequency) field programmable analog array (FPAA). An architecture for current-mode FPAA is presented, which is suitable for a wide class of circuits, including continuous-time filters, differential and algebraic equation solvers, dynamic systems, and fuzzy and multi-valued logic circuits. Examples demonstrate an implementation of a ladder continuous-time filter, an analog rank filter, a circuit for tracking a product of two matrices, a linear equations solver, and a circuit for solving a linear programming problem by the method of steepest descent. The device achieves high performance through the reduced use of global signal interconnections, in favor of local ones. The FPAA is being implemented in a bipolar transistor array technology.

1. Introduction

Analog circuits are a necessary component of many modern signal and information processing systems. In a number of applications, such as anti-aliasing and smoothing (reconstructing) filters, or pulse-slimming circuits in computer disk memories [42], they cannot be replaced by digital circuits, either for the reason of speed, or because of their unique ability to work in continuous-time mode, or – as in some signal processing applications – they are preferred over digital circuits for their relative simplicity.

The unquestioned success of digital programmable devices of various kinds [2, 6–10] is perhaps one of the main stimuli for the development of analog circuits with similar degree of programmability. A number of programmable circuits for various applications were reported. However, *full programmability*, i.e. one of parameters and structure, is still a research issue in analog circuits domain. Whereas several interesting approaches were published, leading to various degrees of programmability, many problems still remain unsolved [1, 3, 7, 13, 16, 17, 19–21, 25–29, 33, 34, 36, 37, 44].

Programmable designs for analog circuits presented in the literature to this day seem to favor flexibility (universality) of the topology[1] (pattern of connections in the programmable device) rather than high performance [22–24, 38]. This is perhaps due to the wide spectrum of topologies of analog circuits designed currently, which do not seem to form any "typical" pattern of interconnection schemes, or at least a pattern suitable for "most" analog circuits. As a result, architectures with many long (global) signal interconnections are created. Whereas they achieve greater flexibility of interconnection patterns, sometimes allowing almost every cell in the programmable device to be connected with any other cell, this approach seriously jeopardizes high frequency performance of the programmable device. Effects caused by parasitics associated with long signal lines and crosstalk between such lines, crosstalk between long analog lines and digital lines on the same chip, etc. are the source of serious noise and stability problems

1. In this paper, the words: architecture, topology, and structure will be used interchangeably where it does not lead to confusion.

in analog and mixed-signal design. Such problems are most acute in the high-frequency domain, where analog circuits have the most interesting and desired applications. Although some of those problems can be partially counteracted by appropriate circuit and layout design techniques, it seems that for a successful development of analog programmable circuits the basic architectural questions need to be addressed.

In this paper we focus on the design of topologies suitable for high-speed, high-performance fully programmable analog circuits. The highest performance can be achieved by reducing the length of signal interconnection lines, if possible – by using **locally-only interconnected structures**. Such an approach – of course – limits the class of circuits realizable in a given structure. However, interesting circuits, such as ladder continuous-time filters, can be realized in locally-only interconnected topologies. We present an architecture of a fully, field programmable analog array, and in a series of examples demonstrate how by extending an initially locally-only interconnected topology one can accommodate more and more complex analog designs, never compromising the high performance for the sake of functionality.

2. Local or global interconnections ?

Many analog circuits are characterized by predominantly local signal interconnections. For instance, in multi-stage amplifiers, there is usually a local coupling between the stages, and local feedback loops around individual stages. Rarely does a feedback loop go around more than two – three stages, and almost never around the whole amplifier. This is dictated by stability considerations. Also, rarely is a feedback loop around many stages necessary [14, 15].

Another example of topologies containing only local interconnections are some of the multiple-feedback-loop topologies for continuous-time filters, particularly one of the most useful, namely the ladder [35]. Further in this paper we present an example of a ladder filter.

Other classes of circuits, such as cellular neural networks (CNNs) [5, 6, 17, 32, 33, 41], cellular automata [30, 43], and certain considerations in artificial neural networks (ANNs) suggesting the advantages of locally interconnected architectures over fully interconnected ones, provide strong incentive to address the question: how useful are the local interconnection structures for the design of programmable analog devices ?

We further present a field programmable analog array, based on local signal interconnections. To make the array more general, a carefully designed global interconnection structure is superimposed on the local interconnections. The array provides flexibility for implementing circuits with local connectivity, in a way maximizing the circuit's performance. Global interconnections can be used when absolutely necessary.

3. The device

Any cell-based programmable device design must address at least two issues: (i) what is the functionality of an individual cell, and (ii) how are the cells interconnected to allow implementation of the desired class of circuits. The first question can be addressed more or less systematically, by considering functions performed by the circuits under consideration. In analog designs, a cell usually will contain a gain element of some kind, and perhaps a time-domain operation such as integration. Some nonlinear functions may be added to enhance functionality. The design of the presented cell and of the interconnection structure is oriented on dynamic-system type of applications, and fuzzy and multi-valued logic.

The device is based on a regular, square array of processing cells, interconnected on two levels: local, and global one. Fig. 1.A, B show the local and the global interconnection patterns, respectively. Each cell is connected to its four nearest neighbors by a two-way current-mode signal interconnection, and thus is able to receive four different signals produced by those neighbors, whether all of them, or just the selected ones. The cell's own output signal is programmably distributed to the same four neighbors (Fig. 1.A). The global interconnection pattern is superimposed on the local one, but it is shown separately to avoid clutter (Fig. 1.B). Each cell can broadcast its output signal (there is just one output signal produced by the cell, the same one which is sent to its local neighbors) to any of the four global lines to which the cell is connected (possibly to more than one line at a time). Since the signals are in current mode, they are summed on the global lines. If more than one cell receives the signal from a given global line, the signal will be divided evenly by the receiving cells. Each cell than can send and receive signals to and from any of it's four nearest neighbors and any of the four global lines to which it is connected. Each cell thus has eight input ports and one output port (or, in fact, eight output ports with copies of the same output signal).

The design of the cell is a result of a compromise between the circuit's power and its simplicity. A block diagram of the cell is shown in Fig. 2. The cell processes current-mode, differential signals. Its core comprises the Gilbert

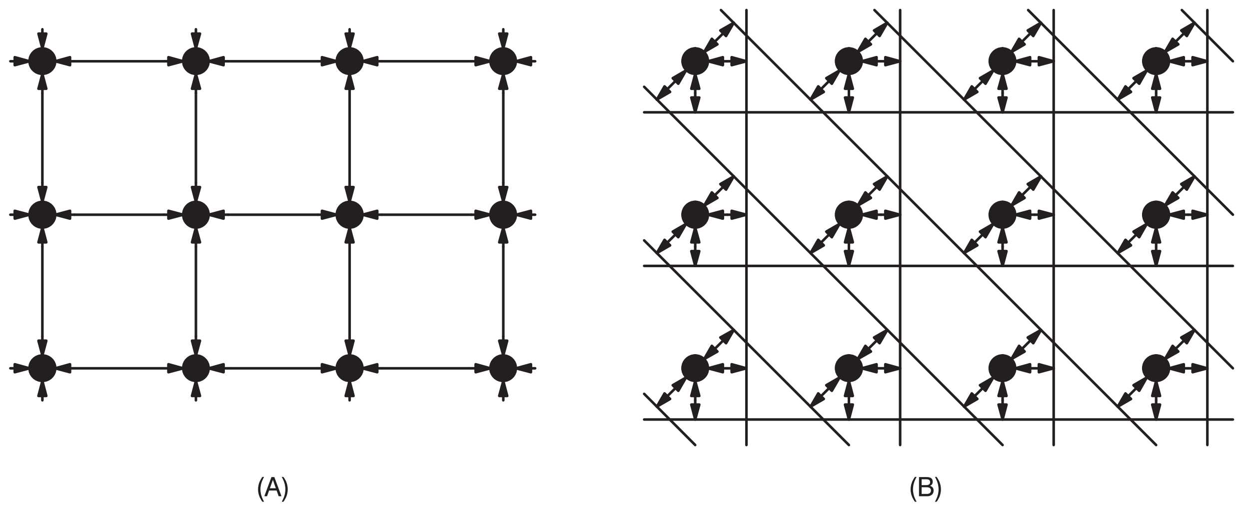

FIGURE 1. Local and global signal interconnections of the FPAA.

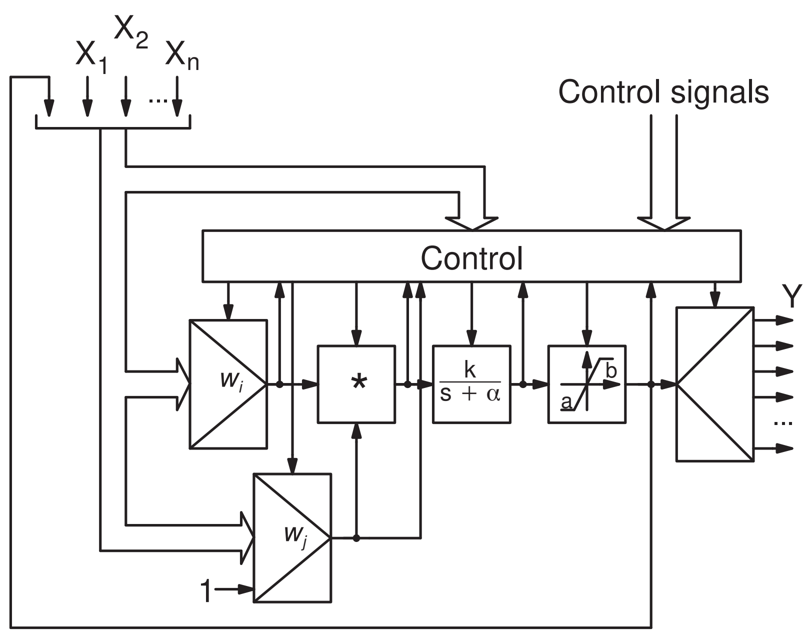

FIGURE 2. Functional diagram of the programmable cell.

multiplier circuit and a wide-band, current-mode amplifier, tunable in the range of 0 – 80 dB [11, 12]. One way to implement the integrator is to have a fixed capacitor programmably connected to the amplifier's output, providing tunable integration (by means of programming the gain of the amplifier). Some additional circuitry is used for programming losses represented by α. α can be turned down almost to 0, if lossless integration is desired. Likewise, the capacitor can be disconnected from the integrator, which results in an "infinite" gain (i.e. very large one, like in an open loop op amp). These extreme cases are required by certain circuits. Finally, the integration function can be bypassed (internally in the integrator, it is not shown in the figure), if a finite, programmable gain is required. The *clipping (saturation) block* provides limiting of the output signal of the cell between two independently programmable values a and b, $-max \leq a \leq 0$, $0 \leq b \leq max$, where $\pm max$ represents the dynamic range of the output signal. The nonlinear characteristic of the clipping block can also be shifted along the horizontal axis. The *input* and *output ports* assure proper communication of the cell with the rest of the programmable device through the local and global signal lines. The input ports (arranged in two groups) pass selectively the input signals, multiplying them by programmable *weights* w_i and w_j, which can be positive, negative, or zero, and then summing. The two input sums (one from each group of input ports) are then passed to the multiplier. Thus, a product of two independently selected sums of input signals is created. If no multiplication is desired, a constant signal (represented by "1") is selected and fed to the second input of the multiplier. The *output ports* produce the desired number of copies of the output signal. Each of the output ports can be turned off. Through the feedback connection the output signal can be connected to the inputs of the cell, which is required in certain circuits. The *control block* stores programming information (loaded to it via *control signals* connection), and sends programming signals to the analog processing blocks of the cell. Since the control block has access to input and internal signals of the cell as well, it can produce control signals being a function of instantaneous input and internal signal values. This feature is used in programming operations such as the *minimum (maximum) follower*, and for other applications, which will be presented in forthcoming papers. To realize the minimum (maximum) follower, the control block simply detects the smallest (largest) signal and selects this signal on the inputs of the cell.

Some operations that can be performed by the cell are summarized in Tab. 1. below. X_i denote input signals, Y denotes output signal. No distinction is made between local and global signals, since the cell processes them in the same manner.

Table 1.

1	$Y = k\,(\sum_{w_i \in W_1} w_i X_i)\,(\sum_{w_j \in W_2} w_j X_j)$	W_1 and W_2 are independent sets of input weights. k is tuned in the range 0 – 80 dB, or k = ∞.
2	$Y = k\,(\sum w_i X_i)$	
3	$Y = k\,X_i\,X_j$	
4	$Y = k\,X_i^2$	
5	$Y = k\,\min(X_1, ..., X_n)$	The control block “watches” input signals and selects the smallest one.
6	$Y = k\,\max(X_1, ..., X_n)$	
7	$Y = k\,Y_{1-6}\frac{1}{s + \alpha}$	$Y_{1–6}$ is any of the functions presented in rows 1–6 above. $\alpha \geq 0$.
8	$Y = k\,sign(Y_{1-7})$	$a = b$, k = ∞. $Y_{1–7}$ is any of the functions presented in rows 1–7. above. This is an ideal diode with clipping.
9	$Y = k\,U(Y_{1-7})$	U is a step function. $a = 0$, $b = max$, $k = \infty$. $Y_{1–7}$ is any of the functions presented in rows 1–7 above.
10	$Y = X_i$	

As seen from the table, the cell performs summing of selected input signals, multiplication of two signals (squaring of one signal), or multiplication of two independently derived sums of input signals. Further processing includes lossless or lossy integration, and clipping[2].

The above shown functions are important for the implementation of continuous-time dynamic-systems [18], multi-valued and continuous (Lukasiewicz, fuzzy) logic circuits, which are the expected applications of the device.

The cell is being implemented in a bipolar transistor array technology. The detailed, circuit-level design of the cell will be presented in forthcoming papers.

4. Examples

4.1. Local connections only

4.1.1. Continuous-time ladder filter

Fig. 3. shows an electrical schematic of an eight-order, elliptic band-pass filter realized as an OTA–C[3] ladder (this schematic results from the so-called OTA–C simulation [35] of an RLC prototype based on the design presented in

2. In a physical realization of the cell the output signal is always limited, i.e. clipping is always present.

[40]). This is a voltage-mode circuit, since each OTA takes a voltage signal as input, and although it produces a current signal, this current is always turned into voltage, either by the integrating operation of a capacitor, or by another OTA with a feedback connection, which is equivalent to a resistor. This is so because eventually each signal created in this circuit is going to be fed to some OTA (which can accept only voltage-mode signals at the input), or connected to the output terminals of the circuit, which also require a voltage-mode signal. This circuit, and other voltage-mode circuits, can be realized in an equivalent current-mode form in the structure of the presented device.

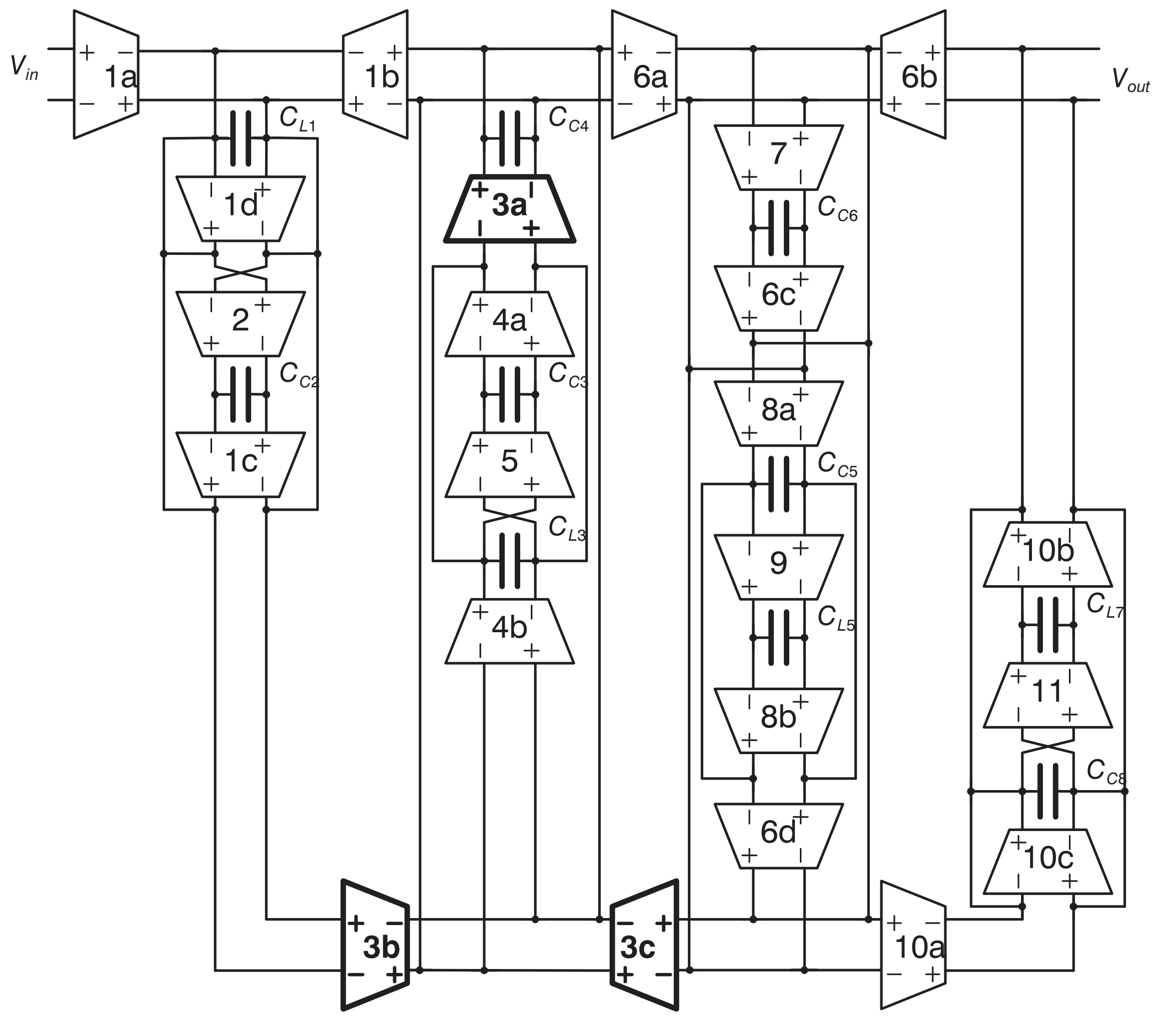

FIGURE 3. Eighth-order elliptic band-pass, fully-differential, ladder OTA–C filter.

At first, the network of the filter does not exhibit much regularity, nor locality of connections. The easiest way to see both is by drawing the graph of connections of the circuit. Since each pair of wires carries one differential signal, it can be represented as a single node of the graph, and each OTA can be represented as a directed edge of the graph (Fig. 4.). The graph reveals regularity which can lead to a number of different realizations based on regular, locally-only interconnected structures. One particular way of deriving a regular structure for the circuit is by grouping all edges coming into a given node as a single unit. As an example, let us consider edges 3a–c, representing OTAs with the same labels (as it is easy to notice, OTAs with outputs connected to the same pair of wires are labeled with the same

3. Operational transconductance amplifier and capacitor.

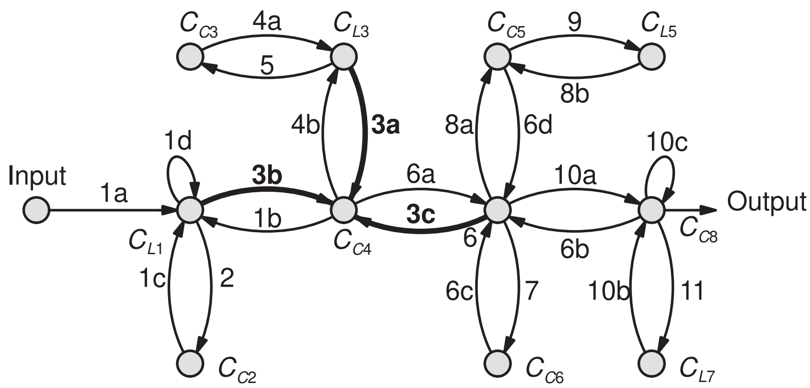

FIGURE 4. Graph of connections of the ladder filter.

numbers). OTAs 3a–c can be collected together as in Fig. 5.A, forming a "cell" with four inputs and one output[4]. Eleven such "cells" can be connected locally only to comprise the whole filter, as it is shown in Fig. 5.B. Dashed lines

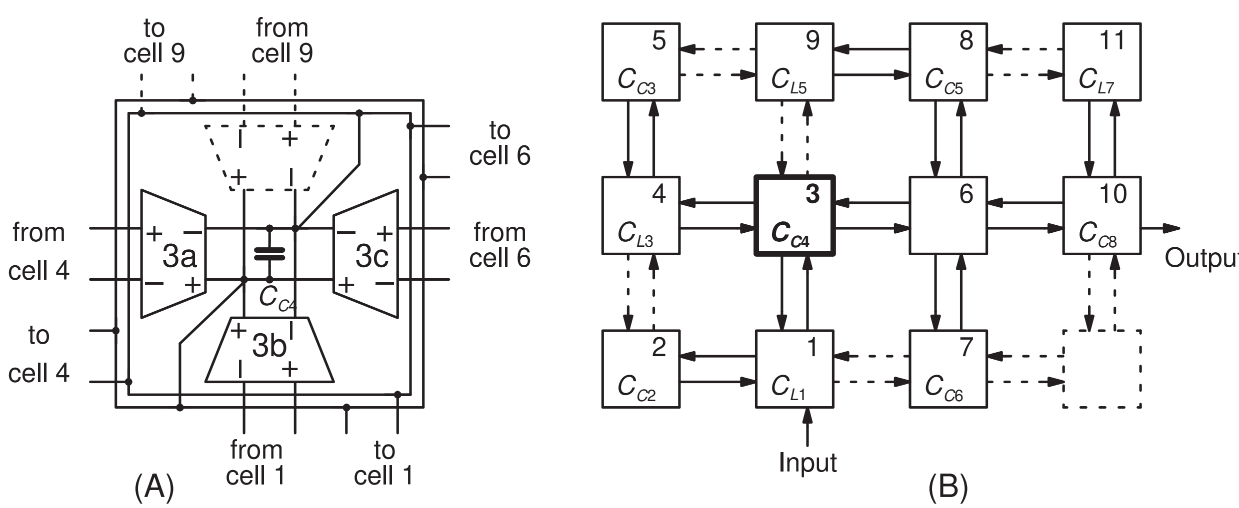

FIGURE 5. Locally-only interconnected topology of the ladder filter.

indicate unused parts of the structure.

Instead of voltage-mode cells (OTA and C) current mode cells of the programmable device are used. The structure of connections is independent of the mode of signals, therefore the cells in the programmable device are arranged the same way as shown in Fig. 5.B. Each cell works in integrating mode, except the cell 6, which realizes "infinite" gain. Cells 1 and 10 realize lossy integrators, all others – lossless. All the signal interconnections are local, within the structure provided by the device. Since the input and output terminals are on the sides of the rectangular collection of cells in Fig. 5.B, no extra global connections are necessary for this circuit, which can be simply placed in the corner of the array.

Most of the ladder filters of practical importance can be mapped into the structure of the presented device in a similar way. Second-order (biquad) filters can be easily mapped, too. Since every transfer function can be realized as a cascade of biquads, which can be then put one next to each other in the array, the device provides the way of realizing continuous-time filters by means of local signal interconnections only.

4. As far as the realization of the whole filter circuit, nodes 1 and 10 would additionally require a feedback connection in one of the OTAs to realize lossy integration. All four OTAs are required only in cell 6.

4.1.2. Analog rank filter

As an example of a nonlinear circuit that can be realized with local signal interconnections only, let us consider a rank filter, described in [31]. Fig. 6.A shows a block diagram of a single cell of the rank filter, and Fig. 6.B demonstrates how it can be mapped into the structure of the presented device. Two cells of the device are necessary to implement one cell of the rank filter. The left-hand cell in Fig. 6.B implements the left-hand part of the rank filter cell, the right-hand cell – the right-hand part. The reader should have no trouble identifying functions performed by each cell in Fig. 6.B. A required number of such cells can be easily placed next to each other to realize the rank filter circuit.

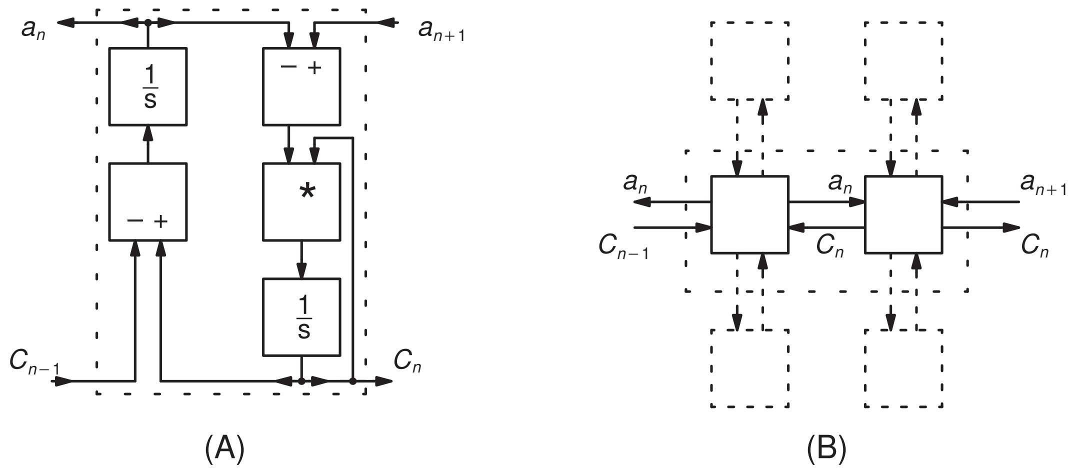

FIGURE 6. Analog rank filter cell.

4.2. Circuits with global connections

4.2.1. Tracking a matrix product

Fig. 7. shows the structure of a matrix product tracking circuit. It takes two time-varying matrices $\mathbf{A}(t) = [a_{ij}]$ and $\mathbf{B}(t) = [b_{ij}]$, both 3×3, and creates their product $\mathbf{C}(t) = \mathbf{A}(t) \cdot \mathbf{B}(t)$ (a factor of 3 is required to account for the distribution of each input signal to 3 cells; alternatively the gain k of each cell could be increased by the same factor). The circuit can be easily generalized for any rectangular conformable matrices. Each element $c_{ij}(t)$ of the product matrix is produced by a "local" group of cells along a diagonal global signal line. However, to distribute the input signals and to collect the result signals, global connections are necessary. Each diagonal output line is used to sum elementary products $a_{ij} \cdot b_{jk}$, $j = 1, ..., n$, comprising the product element c_{ik}.

It is instructive to note that the "globality" of connections results primarily from the need to distribute input signals, and collect the output signals. Creation of each matrix product is done "locally" (although using global signal lines). What is important also, is that global lines are used here only at the "terminals" of the circuit, i.e. for the input and output signals. Global lines are not involved in transmitting internal signals of the circuit.

4.2.2. Tracking the solution of a system of linear equations

Modifying slightly the matrix product tracking circuit, we can build a circuit tracking the solution of a system of linear equations. The solution $\mathbf{x}(t)$ of the system $\mathbf{A}(t) \cdot \mathbf{x}(t) = \mathbf{b}(t)$ can be found by solving a system $\dot{\mathbf{x}}(t) + \mathbf{A}(t) \cdot \mathbf{x}(t) - \mathbf{b}(t) = \mathbf{0}$ of differential equations provided that matrix $\mathbf{A}$(t) is always positive stable [18]. In many practical cases matrix $\mathbf{A}$ will be time-invariant, but it is instructive to see the solution of a more general problem, i.e. with a time-varying matrix $\mathbf{A}(t)$. Fig. 8. shows a circuit solving a system of 3 equations with 3 unknowns $x_1(t)$, ..., $x_3(t)$. The global connections in this circuit carry internal feedback signals, although the distance travelled by these signals is small.

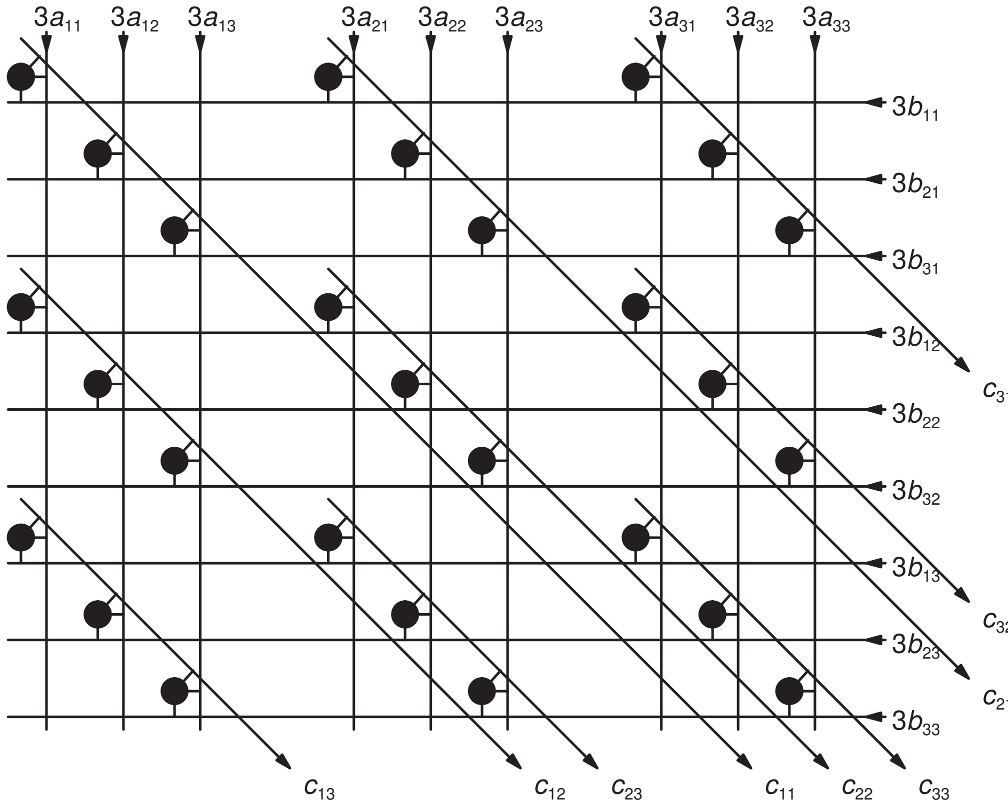

FIGURE 7. Continuous-time matrix product tracking circuit.

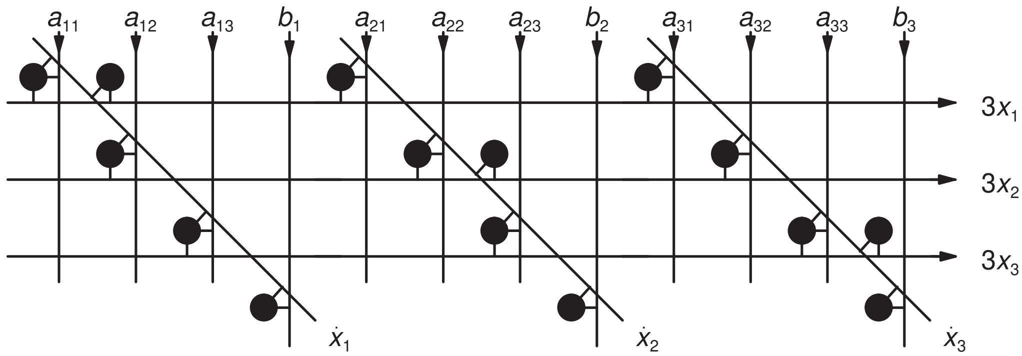

FIGURE 8. Continuous-time circuit for solving a system of linear algebraic equations.

4.2.3. Tracking the solution of a linear programming problem

A linear programming problem can be stated as follows. Given a set of constraints $\mathbf{g}(t) = \mathbf{F}(t) \cdot \mathbf{x}(t) = [g_1(t), ..., g_m(t)]' \leq \mathbf{0}$ (the inequality is supposed to hold for every element of the vector; **F** is a rectangular matrix of constraints coefficients, **g** is a vector representing individual constraints), minimize the objective function $\rho(x_1, ..., x_n) = \boldsymbol{\rho} \cdot \mathbf{x} = \rho_1 x_1 + ... + \rho_n x_n$, where $\boldsymbol{\rho} = [\rho_1, ..., \rho_n]$. Application of the method of steepest descent leads to the system of equations $\dot{\mathbf{x}} = -\mu \cdot \boldsymbol{\rho}' - 2a \cdot \mathbf{A} \cdot \mathbf{diag}(\mathbf{g}) \cdot U(\mathbf{g})$, where $U(\mathbf{g})$ denotes the step function, $\mathbf{diag}(\mathbf{g})$ denotes a diagonal matrix with elements of vector **g** on the main diagonal, and μ and a are constants ($\mu \to 0$, $a \to \infty$) [18]. This system can be solved by the circuit shown in Fig. 9. In the case of linear constraints matrix **A** will be identical to matrix **F**, nevertheless a more general circuit not assuming this equality is shown as an illustration of the versatility of the device. A simplified circuit, with only matrix **F** input, can be easily derived.

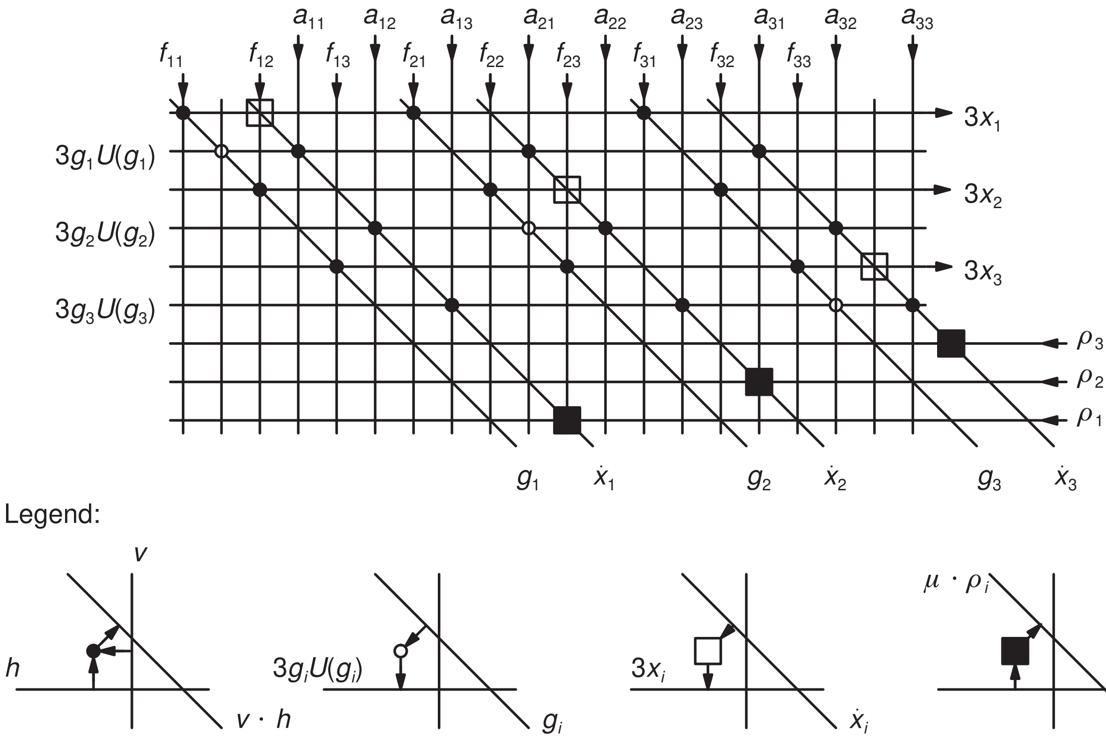

FIGURE 9. Continuous-time circuit for solving a linear programming problem.

5. Conclusions

We have presented an architecture suitable for the realization of a wide class of analog circuits. This specific architecture results from the general premise to use local signal interconnections whenever possible, and global signal interconnections only when absolutely necessary. The design of the individual cells, and specific details of the architecture, were determined upon consideration of the perceived applications of the device, i.e. fast dynamic systems and fuzzy and multi-valued logic circuits. Although only continuous-time examples were shown, the device is capable of discrete-time operation as well, which will be demonstrated in forthcoming papers. The fuzzy and multi-valued logic applications of the device, as well as the electrical design of the individual cells, will also be presented in future papers.

Bibliography

1. *80170NX, Electrically Trainable Analog Neural Network*, Intel Corp., Santa Clara, CA, June 1991.
2. *The Programmable Gate Array Data Book,* Xilinx, San Jose, CA, 1989.
3. Allstot, D. J., Brodersen, R. W., and P. R. Gray, “An Electrically–Programmable Switched Capacitor Filter”, *IEEE J. Solid–State Circ.*, Vol. 14, pp. 1034–1041, Dec. 1979.
4. Chang, P. R., Hwang, K. S., and H. M. Gong, “A High-Speed Neural Analog Circuit for Computing the Bit-Level Transform Image-Coding”, *IEEE Trans. on Consumer Electronics*, Vol. 37, No. 3, pp. 337–342, 1991.
5. Chua, L. O., “Cellular Neural Networks: Theory”, *IEEE Trans. on Circuits and Systems*, Vol. 35, No. 10, pp. 1257–1272, Oct. 1988.
6. Chua, L. O., “Cellular Neural Networks: Applications”, *IEEE Trans. on Circuits and Systems*, Vol. 35, No. 10, pp. 1273–1290, Oct. 1988.
7. Fisher, W. A., Fujimoto, R. J., and R. C. Smithson, “A Programmable Analog Neural Network Processor”, *IEEE Trans. on Neural Networks*, Vol. 2, No. 2, pp. 222–229, Mar. 1991.
8. Furtek, F. C., “Programmable Logic Cell and Array”, U.S. Pat. No 4,918,440, Apr. 17, 1990.
9. El Gamal, A., “Programmable Interconnect Architecture”, U.S. Pat. No 4,873,459, Oct. 10, 1989.
10. El Gamal, A., Greene, J., Reyneri, J., Rogoyski, E., El–Ayat, K. A. , and A. Mohsen, “An Architecture for Electrically Configurable Gate Arrays”, *IEEE J. Solid–State Circ.*, Vol. 24, No. 2, pp. 394–398, Apr. 1989.
11. Gilbert, B., “A New Wide–Band Amplifier Technique”, *IEEE Journal of Solid State Circuits*, Vol. SC–3, No. 4, pp. 353–365, Dec. 1968.
12. Gilbert, B., “Current–mode Circuits From a Translinear Viewpoint: A Tutorial”, in: *Analogue IC Design: the current–mode approach*, ed. C. Toumazou, F. J. Lidgey, D. G. Haigh, pp. 11–91, Peter Peregrinus Ltd., 1990.

13. Graf, H. P., and D. Henderson, “A Reconfigurable CMOS Neural Network”, *ISSCC, IEEE*, San Francisco, CA, Feb. 1990.
14. Grebene, A. B., *Bipolar and MOS Analog Integrated Circuit Design*, J. Wiley, 1984.
15. Grey, P. R., and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 3rd ed., J. Wiley, 1993.
16. Halonen, K., Porra, V., Roska, T., and L. Chua, “Programmable Analog VLSI CNN Chip with Local Digital Logic”, *ISCAS, IEEE*, pp. 1291–1294, Singapore, 1991.
17. Harrer, H., Nossek, J. A., and R. Stelzl, “An Analog Implementation of Discrete–Time Cellular Neural Networks”, *IEEE Trans. on Neural Networks*, Vol. 3, No. 3, pp. 466–476, May 1992.
18. Hausner, A., *Analog and Analog/Hybrid Computer Programming*, chap. 11, Prentice–Hall, Inc., Englewood Cliffs, N.J., 1971.
19. Johns, D. A., Snelgrove, W. M., and A. S. Sedra, “Continuous-Time Analog Adaptive Recursive Filters”, *ISCAS, IEEE*, Portland, OR, 1989.
20. Kub, F. J., Moon, K. K., Mack, I. A., and F. M. Long, “Programmable Analog Vector–Matrix Multipliers”, *IEEE Journal of Solid-State Circ.*, Vol. 25, No. 1, pp. 207–214, Feb. 1990.
21. Kub, F. J., Mack, I. A., and K. K. Moon, “Programmable Analog Voltage Multiplier Circuit Means”, U.S. Pat. No 4,931,674, June 5, 1990.
22. Lee, E. K. F., and P G. Gulak, “A CMOS Field–Programmable Analog Array”*, 1991 IEEE ISSCC Dig. Technical Papers*, Vol. 34, pp.186–187, Feb. 1991.
23. Lee, E. K. F., and P G. Gulak, “A CMOS Field–Programmable Analog Array”, *IEEE J. Solid–State Circ.*, Vol. 26, No. 12, pp. 1860–1867, Dec. 1991.
24. Lee, E. K. F., and P G. Gulak, “Field Programmable Analogue Array Based on MOSFET Transconductors”, *Electronics Letters*, Vol. 28, No. 1, pp. 28–29, IEE, Jan. 2 1992.
25. Loh, K–H., Hiser, D. L., Adams, W. J., and R. L. Geiger, “A Versatile Digitally Controlled Continuous–Time Filter Structure, with Wide–Range and Fine Resolution Capability”, *IEEE Trans. on Circuits and Systems*, Vol. 39, No. 5, pp. 265–276, May 1992.
26. Manetti, S., and M. C. Piccirilli, “A Fully Programmable Structure for Continuous–Time MOS Filters”, *6th Mediterranean Electrotechnical Conference, (MELECON ’91), IEEE*, pp. 355–358, Ljubljana, Yugoslavia, May 22–24, 1991.
27. Mann, J., Lippmann, R., Berger, B., and J. Raffel, “A Self-Organizing Neural Net Chip”, *CICC, IEEE*, pp. 10.3.1–10.3.5, 1988.
28. Marchesi, M., Orlandi, G., Piazza, F., and A. Uncini, “Neural Networks with Self–Adaptive Topology”, *ISCAS, IEEE*, pp. 2498–2501, Singapore, 1991.
29. Melton, M. S., Phan, T., Reeves, D. S., and D. E. Van den Bout, “The TInMANN VLSI Chip”, *IEEE Trans. on Neural Networks*, Vol. 3, No. 3, pp. 375–384, May 1992.
30. Von Neumann, J., *Theory of Self–Reproducing Automata* (edited and completed by Artur Burks), Univ. of Illinois Press, 1966.
31. Paul, S., Hümper, K., and J. A. Nossek, “A Simple Analog Rank Filter”, *ISCAS, IEEE*, pp. 121–124, San Diego, CA, 1992.
32. Rodriguez-Vazquez, A., Espejo, S., Dominguez-Castro, R., Huertas, J. L., and E. Sanchez-Sinencio, “Current-Mode Techniques for the Implementation of Continuous- and Discrete-Time Cellular Neural Networks”, *IEEE Trans. on Circuits and Syst., II: Analog and Digital Sign. Proc.*, Vol. 40, No. 3, March 1993.
33. Roska, T., and L. O. Chua, “The CNN Universal Machine: An Analogic Array Computer”, *IEEE Trans. on Circuits and Syst., II: Analog and Digital Sign. Proc.*, Vol. 40, No. 3, March 1993.
34. Satyanarayana, S., Tsividis, Y. P., and H. P. Graf, “A Reconfigurable VLSI Neural Network Chip”, *IEEE J. Solid–State Circ.*, Vol. 27, No. 1, pp. 67–81, Jan. 1992.
35. Schaumann, R., Ghausi, M. S., and K. R. Laker, *Design of Analog Filters*, Prentice Hall, Englewood Cliffs, NJ, 1990.
36. Schwartz, D. B., Howard, R. E., and W. E. Hubbard, “A Programmable Analog Neural Network Chip”, *IEEE J. Solid–State Circ.*, Vol. 24, No. 2, pp. 313–319, Apr. 1989.
37. Sheu, B. J., “VLSI Neurocomputing with Analog Programmable Chips and Digital, Systolic Array Chips”, *ISCAS, IEEE*, pp. 1267–1270, Singapore, 1991.
38. Sivilotti, M. A., “A Dynamically Configurable Architecture for Prototyping Analog Circuits”, in *Advanced Research in VLSI, Fifth MIT Conf.*, ed. J. Allen, F. Thomson Leighton, pp. 237–258, The MIT Press, Cambridge, MA, Mar. 1988.
39. Van der Spiegel, J., Mueller, P., Blackman, D., Chance, P., Donham, Ch., Etienne–Cummings, R., and P. Kinget, “An Analog Neural Computer with Modular Architecture for Real–Time, Dynamic Computations”, *IEEE J. Solid–State Circ.*, Vol. 27, No. 1, pp. 82–92, Jan. 1992.
40. Tan, M. A., *Design and Automatic Tuning of Fully Integrated, Transconductance–Grounded Capacitor Filters*, Ph.D. Thesis, Univ. of Minnesota, 1988.
41. Varrientos, J. E., Sanchez-Sinencio, E., and J. Ramirez-Angulo, “A Current-Mode Cellular Neural Network Implementation”, *IEEE Trans. on Circuits and Syst., II: Analog and Digital Sign. Proc.*, Vol. 40, No. 3, March 1993.
42. De Veirman, G. A., and R. G. Yamasaki, “Monolithic 10 – 30 MHz Tunable Bipolar Bessel Lowpass Filter”, *ISCAS, IEEE*, Singapore, pp. 1444–1447, 1991.
43. Ulam, S., “Random Processes and Transformations”, *Int. Congr. Mathem.* (held in 1950), Vol. 2, pp. 264–275, 1952.
44. Zarabadi, S. R., Larsen, F., and M. Ismail, “A Reconfigurable Op-Amp/DDA CMOS Amplifier Architecture”, *IEEE Trans. on Circuits and Syst., II: Analog and Digital Sign. Proc.*, Vol. 39, No. 6, June 1992.