

CONFETTI : A reconfigurable hardware platform for prototyping cellular architectures

Pierre-André Mudry, Fabien Vannel, Gianluca Tempesti, Daniel Mange

Cellular Architectures Research Group
École Polytechnique Fédérale de Lausanne (EPFL)
EPFL-IC-GRTEM, Station 14, 1015 Lausanne, Switzerland
pierre-andre.mudry@epfl.ch

Abstract

In this article, we describe a novel hardware platform aimed at the realization of cellular architectures. The system is built hierarchically from a very simple computing unit, called ECell. Several of these units can then be connected, using a high-speed serial communication protocol, to a more complex structure called the UltraStack. Consisting of four different kinds of interconnected boards (computational, routing, power supply, and display), these stacks can then be joined together to form an arbitrarily large parallel network of programmable circuits.

This structure, while theoretically universal in its operation, is however particularly suited for the implementation of cellular computing applications.

1 Introduction and motivations

Until recently, the ever-increasing demand of computing power have been met on one hand by increasing the operating frequency of processors and on the other hand by designing architectures capable of exploiting parallelism at the instruction level through hardware mechanisms such as super-scalar execution. However, both these approaches seem to have reached their practical limits, mainly due to issue related to design complexity and cost-effectiveness.

The current trend in computer design seems to favor a switch to coarser-grain parallelization, typically at the thread level. In other words, high computational power is achieved not by a single very fast and very complex processor, but through the parallel operation of several on-chip processors, each executing a single thread. This kind of approach is currently implemented commercially

through multi-core processors and in the research community through the *Multi-processors Systems On Chip* (MP-SoCs) term, which is itself largely based on the *Network On Chip* (NoC) paradigm ([6], [5]).

Extrapolating this trend to take into account the vast amount of on-chip hardware resources that will be available in the next few decades (either through further shrinkage of silicon fabrication processes or by the introduction of molecular-scale devices), together with the predicted features of such devices (e.g., the impossibility of global synchronization), this approach comes to resemble another computational paradigm, commonly known as *cellular computing*.

Loosely based on the observation that biological organisms are in fact highly complex structures realized by the parallel operation of vast numbers of relatively simple elements (the cells), this paradigm tries to draw an analogy between multi-cellular organisms and multi-processor systems. At the base of this analogy lies the observation that organisms, in addition to being completely asynchronous, are built through a bottom-up self-assembly process and do not require the specification of a complete layout.

The actual interpretations and implementations of this paradigm are extremely varied, ranging from theoretical studies [12] [13] to commercial realizations (notably, the *Cell* CPU [10] [11] jointly developed by IBM, Sony and Toshiba), through wetware-based systems [3], OS-based mechanisms [7] and amorphous computing approaches [1].

Depending on the authors, the *cells* may comprise different levels of complexity ranging from very simple, locally-connected, logic elements to high-performance computing units endowed with memory and complex network capabilities. However, in every case, the basic idea of two-dimensional systems composed of relatively simple connected elements, remains.

Our past research has tried to approach cellular comput-

ing by designing large arrays of custom processing elements and by analyzing how some of the mechanisms involved in the development of biological organisms can be effectively applied to these arrays in order to achieve useful properties such as fault tolerance or growth.

A key aspect of our research has traditionally been an attempt to physically realize, in hardware, the systems we have developed through the years in order to verify their properties and to analyze their efficiency. Considering the complexity of this kind of systems, their prototyping in hardware requires vast amounts of reconfigurable resources and has led us to the realization of a custom platform specifically designed to implement and test complex cellular computing systems. In this article, we present the salient features of this platform, which we have labeled *CONFETTI*, for *CONFigurable ElecTronic Tissue*.

The paper is organized as follows: in the next section, we will give a brief overview of the computational approach that motivated the hardware architecture and describe the state of the art in the domain by analyzing some previous work. The hardware platform will then be described in some detail in section 3 before discussing several general issues, such as power consumption and integrated test and monitoring, in section 4. Finally, section 5 concludes this article and introduces future work.

2 Background

Almost every living being, with the notable exceptions of viruses and bacteria, share the same basic principles for their organization. Based on cell differentiation, the incredible complexity present in organisms is based on multicellular organization where cells having a limited function achieve very complex behaviors by assembling into specific structures and operating in parallel. By analogy, in the context of thread-level parallelism in a computing machine, *cellular computing* consists of the replication of similar, relatively simple computing elements that execute in parallel the different parts of a given application. Containing memory, computational power and communication capabilities, each cell provides a complete environment for running a whole thread.

Our past research in this area [19], [15], has focused on developing a hierarchical approach to design digital hardware that can efficiently implement some specific aspects of this bio-inspired approach. The key aspect of our project in the context of this article is the need for extremely large prototyping platforms involving considerable amounts of programmable logic. This need, along with the non-standard features of our approach, lead us to design and build custom platforms that allow us to implement and test in hardware the mechanisms involved.

The first platform of this kind was realized a few years

ago thanks to a grant of the Villa Reuge foundation and was destined mainly to illustrate the features of our approach to the general public. The structure of the platform was centered around the need to clearly display the operation of the system and, as a consequence, the BioWall ([14], [16]) is a very large machine ($5.3m \times 0.6m \times 0.5m$). Intended as a giant reconfigurable computing tissue, the BioWall is composed of 4000 "molecules", each consisting of a 8 by 8 two color LED matrix, one transparent touch sensor and one Spartan[®] XCS10XL reconfigurable circuit.

This "electronic tissue" has been successfully used for prototyping bio-inspired computing machines [17], and has served as a basis for the development of a second bio-inspired architecture, the POEtic tissue ([19] and [18]). In both cases, the same idea of highly parallel interconnected simple cells has served as the background idea for the realization of the architecture.

Despite the fact the BioWall has fulfilled its role and has been successfully used during several years, it suffers from several limitations which hinder the development of new applications.

Firstly, the same FPGA configuration has to be used for each cell, which limits the functionality of every unit to the 10000 equivalent logic gates of the Spartan XCS10XL, while the considerable delays inherent in propagating a global signal over distances measured in meters limit the clock speed to about one megahertz. This latter fact confines the system to applications where the required computational speed is very low, such as those in which human interaction is required (the intended target of the platform).

Secondly, the entire system is controlled by an electronic board connected to a PC and aimed at configuring all the FPGAs and setting and distributing the clock signal to the 4000 FPGAs. Thus, the BioWall acts as a slave electronic system even if the application does not require any interaction with the host computer once configured. This limitation prevents the BioWall from being fully autonomous and introduces a functional bottleneck at the interface between the PC and the reconfigurable logic.

These drawbacks, along with the evolution of programmable logic devices, have led us to define a novel platform for the implementation of our systems. In the next sections, we will present the structure of the new platform and its salient features.

3 A novel hardware platform for cellular computing

The *CONFETTI* platform tries to avoid the Biowall's shortcomings by proposing an increased amount of versatility and interchangeability in the different constituting elements of the hardware system. Moreover, the system is built hierarchically by connecting elements of increasing com-

plexity which permits to handle more easily the complexity of the whole system.

The platform is composed of a set of stacks of printed-circuit boards (PCBs), called *UltraStacks* (Fig. 1), that can be connected together side by side to create two-dimensional arrays of arbitrary size.

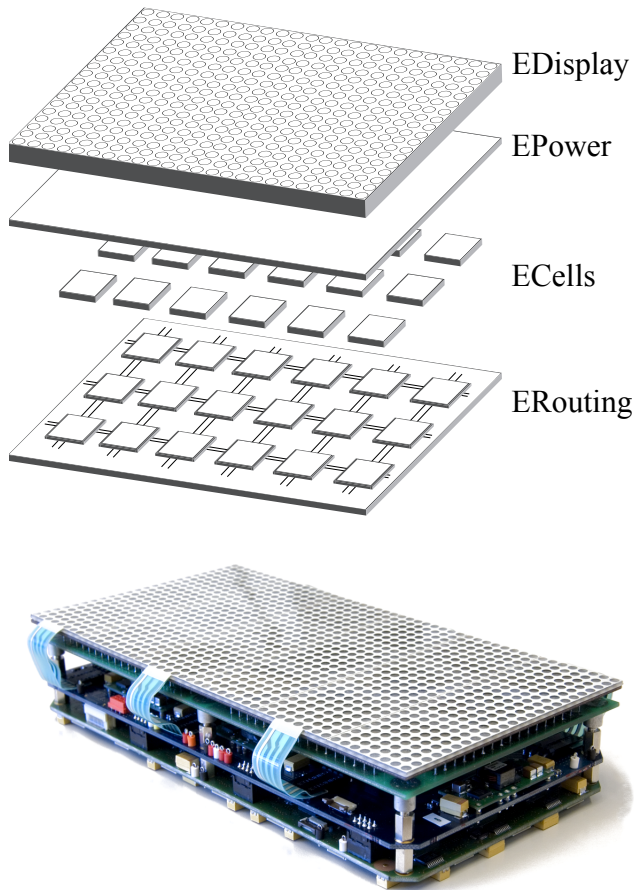


Figure 1. *UltraStack* - schematic and photography.

Each *UltraStack* is composed of four kinds of boards:

- The *ECell* boards (up to 18 per *UltraStack*) represent the computational part of the system and are composed of an FPGA and 8 MBytes static memory. Each *ECell* is directly connected to a corresponding routing FPGA in the subjacent *ERouting* board.
- The *ERouting* board (1 per *UltraStack*) implements the communication layer of the system. Articulated around 18 FPGAs, the board implements a routing network based on a mesh topology which provides

inter-FPGA communication but also communication to other routing boards.

- The topmost layer of the *UltraStack*, the *EDisplay* board, consists of a RGB LED display to which a touch sensitive matrix has been added.
- Above the routing layer lies a board called *EPower* that generates all the power supplies required by the system and handles functions such as startup and monitoring.

3.1 The *ECell* board

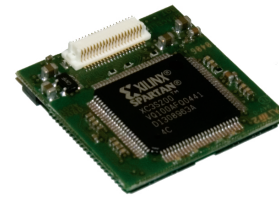


Figure 2. A picture of the *ECell* board.

The *ECell* (Fig. 2) constitutes the basic building block of our hardware platform. It is articulated around a SPARTAN[®] 3 FPGA¹ from Xilinx[®] coupled with 8 Mbits of 10 ns SRAM memory and a temperature measurement chip. Equivalent to 200000 logic gates, the core of this board possesses some interesting features such as hardware multipliers, 18 Kb of internal dual-port memory and four digital clock managers (DCM) that allow to obtain, from the 50 MHz local clock, working frequencies up to 300 MHz. All these components are soldered on a very small (26 × 26 mm) 8-layer PCB.

The *ECell* possesses various connections to the others components of the system that all pass through the connector visible on top of Fig. 2. The connectivity of the board is as follows:

- Differential high-speed connections lines with the subjacent FPGA on the *ERouting* board (3 pairs in each direction, 500 Mbits per pair).
- Configuration lines.
- Communication bus to the *EPower* board to carry the display signals.
- Power supply lines.

3.2 The *ERouting* board

One of the main challenges in today's hardware architectures resides in implementing versatile communication

¹The exact model is the XC3S200 FPGA

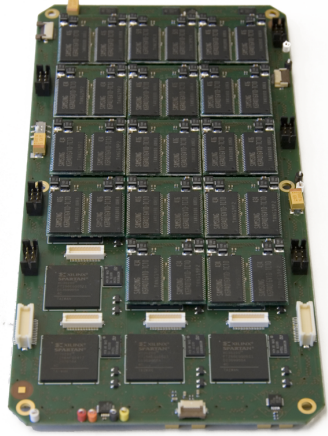


Figure 3. The *ERouting* board with four *ECCell* removed on the bottom left corner.

capabilities that are able to provide a sufficient bandwidth whilst remaining cost- and size-efficient, as evidenced in research for *Network-On-Chip* [4] and other [8] systems. For our platform, we opted for a solution based on high-speed serial connections able to sustain different kinds of routing algorithms.

To reduce as much as possible the load on the *ECCell* board, where the computational power resides, communication in our system is implemented within the *ERouting* board, which also handles tasks such as system configuration and display management.

Measuring 192×96 mm, this highly complex board (twelve layers) has components soldered on both sides and can host as many as eighteen *ECCell* boards. Based on a six-by-three regular grid topology (Fig. 3), this board is composed of a set of eighteen reconfigurable circuits (the *ERouting* FPGAs) and eighteen Flash memories, whose functions is outlined below.

3.2.1 High speed communication

As the main purpose of the board is the implementation of the routing network that connects the computational units of the system (the *ECCell* boards), one of the most crucial aspects of the *ERouting* board is the kind of connections that link the board's FPGAs together and with the *ECCell* boards.

Physically, every *ERouting* FPGA is linked to its four cardinal neighbors and to the *ECCell* board above it (Fig. 4). This setup was selected for modularity and scalability purposes (it avoids long and global communication lines that could cause bandwidth degradation in a big *CONFETTI* configuration) and because it is the kind of layout typically

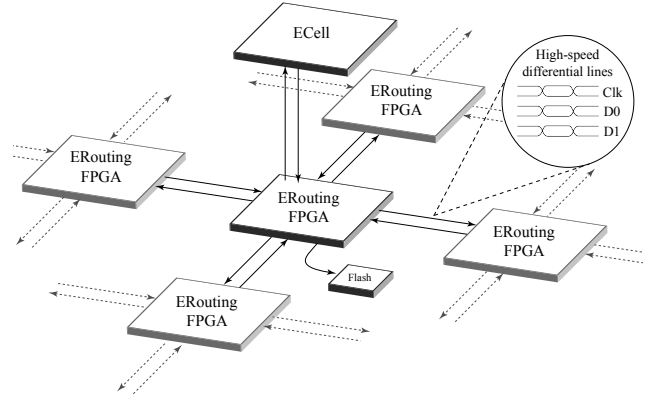


Figure 4. Detail of one *ERouting* FPGA and link with its *ECCell* module.

used in cellular computing applications.

The links between each FPGA are implemented using the built-in SPARTAN[®] 3 LVDS² I/O drivers that allow, in our case, data rates up to 500 Mbits/s. As depicted on Fig. 4, two communication buses (one for each direction, 3 bits per bus) are present for each neighboring pair. Because the SPARTAN[®] 3 family does not provide serial transceivers directly integrated on-chip, the transmitter and receiver blocks were designed by hand. Also, since there is no global clock in the system and no clock recovery possibility, a clock signal is transmitted on one differential pair to synchronize the data transmitted on the two others pairs. Thus, at *ERouting* level, a bandwidth of 1 Gbit/s is available on each *ERouting* FPGA for every direction. Moreover, the same type of bus exists between each *ERouting* FPGA and its corresponding *ECCell*.

As the *ERouting* boards constitute the communicating backplane of the whole *CONFETTI*, connections between the different *UltraStack* boards are also implemented here. External connectors are present on the four sides of the board and provide the same connectivity as the links between the FPGAs: two adjacent *ERouting* boards then represent effectively a single uniform surface of FPGAs. This setup allows the creation of systems consisting of several *UltraStacks* that behave as a single, larger *UltraStack*.

3.2.2 Routing

While the above-mentioned high-speed links provide only very local *physical* communication capabilities, the *ERouting* FPGAs obviously allow the implementation of more complex communication schemes such as broadcasting or point-to-point communication. Seen as a simple interface

²Low Voltage Differential Signaling

from the *ECell*, the routing network provided by the *ERouting* board provides the necessary substrate to be able to implement, at application level, complex data transfers between the *ECells*.

Of course, many different types of networking paradigms exist and could be implemented in our system (for example [9], [20] or [2]). As a first realization, we decided to use the *Hermes* framework [8], a powerful packet routing system, which provides many interesting functionalities for a relatively low hardware overhead. Available as a VHDL core, five *Hermes* switch are implemented in each *ERouting* FPGA, providing a bandwidth of 500 Mbits/s in every direction.

3.2.3 Configuration

The various reconfigurable circuits used in *CONFETTI* are all based on SRAM technology, which means they can be reconfigured an unlimited amount of times but also that the configuration requires a relatively short time (typically 20 ms). Because every *ECell* FPGA could have a different configuration and even be reconfigured dynamically, one of the problems that need to be addressed by the system is how to direct the correct configuration to each of the *ECell* FPGAs in the system. This task is performed within the *ERouting* board. To perform this task, each *ERouting* FPGA can access an adjacent 16 Mbits Flash memory, typically used to store as many as sixteen different configurations for the *ECell* FPGAs or serve as non-volatile memory available for applications.

The contents of this memory can be modified using an external interface connected to a computer (more on this in section 4.2). At the moment, this constitutes the only way to store new applications to be executed by the *ECells*. Of course, the configurability of the *ERouting* FPGAs allows almost unlimited versatility in the configuration scheme, allowing for example the implementation of applications that would update the Flash contents using external memories, Ethernet or WiFi connection, etc., or retrieve the *ECell* configurations from sources other than the local Flash memory.

3.3 The *EPower* board

The SPARTAN[®] 3 FPGAs that are used throughout the *CONFETTI* are very recent products, built on a 90 nm CMOS process. This gives them the advantage of being very fast and of having lot of embedded features but also has the disadvantage of needing several low-power voltages. Thus, the FPGA core is powered by a 1.2 V voltage but also needs 2.5 V for the LVDS interface and for configuration purposes. Finally, a 3.3 V voltage is needed to interface the Flash and SRAM memories. Moreover, all these voltages need to be very well stabilized, as Xilinx[®] FPGAs allow only $\pm 5\%$ tolerance.

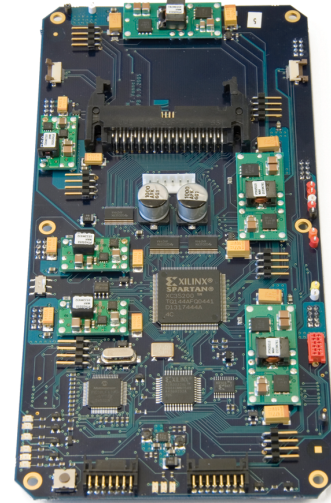


Figure 5. The *EPower* board.

To cope with all these requirements and the fact that the eighteen *ECells* and the *ERouting* board are not only very complex but also power-hungry, an *EPower* board was added on top of *ERouting* and *ECell* layers. This six-layer board, mainly responsible of supplying the correct voltage to all the components on the boards underneath, has the same size as the *ERouting* board. Articulated around six DC / DC converter, this board generates from a global 5 V the three mentioned voltages of 1.2 V, 2.5 V and 3.3 V that are then brought to the *ERouting* board using six 8-pins connectors.

Due to the high complexity of all the board in the *UltraStack*, a micro-controller on the *EPower* board acts as a supervisor and checks several factors (like power supplies stability) in the aim of preventing failures. This micro-controller is also responsible of supervising the start-up of the whole *UltraStack*, a rather complex sequence that involves switching on the DC / DC converters, controlling the stability of all voltages, configuring the *ERouting* FPGAs, and monitoring the temperatures of all *EPower*, *ERouting* and *ECell* boards. If any of these tests should fail, the entire system is switched off to prevent damages.

3.4 The *EDisplay* board

Unlike the above-mentioned BioWall, which was primarily a demonstrator, the main purpose of *CONFETTI* is the high-speed prototyping of complex multi-cell systems. Nevertheless, the success of the earlier machine led us to integrate in the new one a relatively simple display. On the very top of the *UltraStack* lies then a 24-bit RGB LED display capable of displaying 48×24 pixels that can be refreshed at a rate of 100 times per second (a dedicated

SPARTAN[®] 3 on the *EPower* board manages the display's framebuffer).

The purpose of this display is to provide a distributed overview of the operation of the system (for example, to illustrate its operation at reduced speed or to display long-term patterns such as network congestion or thermal buildup). Each *ECell* has access to only part of the screen, namely a square of eight by eight pixels directly above it. To provide a direct human interface to the system, a touch-sensitive surface was glued to each square.

Even if the resolution available for each *ECell* is very limited, the main advantage of this kind of screen resides in the fact that it is possible to put several screens border to border without any gap, a necessary feature in view of building large systems consisting of several *UltraStacks* side by side.

4 The CONFETTI system

The previous section has been devoted to the description of the *UltraStack* and, as we mentioned, a complete *CONFETTI* system consists of an arbitrary number of such stacks seamlessly joined together (through the border connectors in the *ERouting* board) in a two-dimensional array.

The current test configuration that has been built and tested, for example, consists of six *UltraStacks* in a 3 by 2 array (Fig. 6).

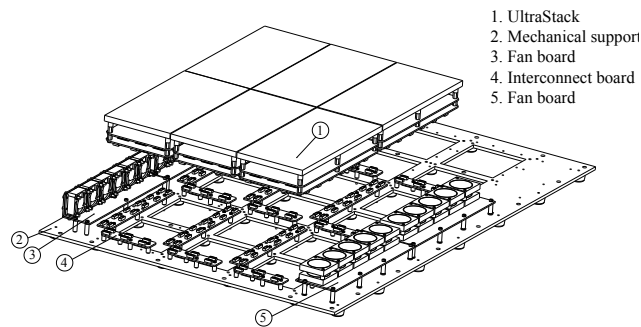


Figure 6. The *CONFETTI* system.

The connection of several boards together potentially allows the creation of arbitrarily large surfaces of programmable logic. Obviously, however, considerations of power consumption, thermal management, and system monitoring come into play for a system of this kind.

4.1 Thermal management

Despite the fact that the reconfigurable circuits in the platform use a state-of-the-art fabrication technology that makes them less power hungry than previous generation FPGAs, the number of circuits involved in the whole system

makes thermal management a real issue, each *UltraStack* consuming a maximum total of 100 Watts. To solve it, we had to implement an adequate cooling system in order to evacuate the generated heat. Thus, we integrated boards with fans on the top and bottom borders of the whole system. Two different types of fan boards have been used: on the bottom, fans are used for introducing cooler air on the FPGAs and on the top, fans extract the hot air away from the boards. Each board comprises between six and eight fans, each independently controllable.

To ensure a good thermal protection without having to turn on all the available fans, temperature is constantly monitored on several components:

- Every *ECell*
- Every FPGA on the routing board but also on three other locations of the *ERouting*
- Every *EPower* contains six temperature sensors.

It is then relatively easy to detect thermal hot spots and turn on the necessary fans, a process which is done by a dedicated component described in the next section.

4.2 Integrated test and monitoring

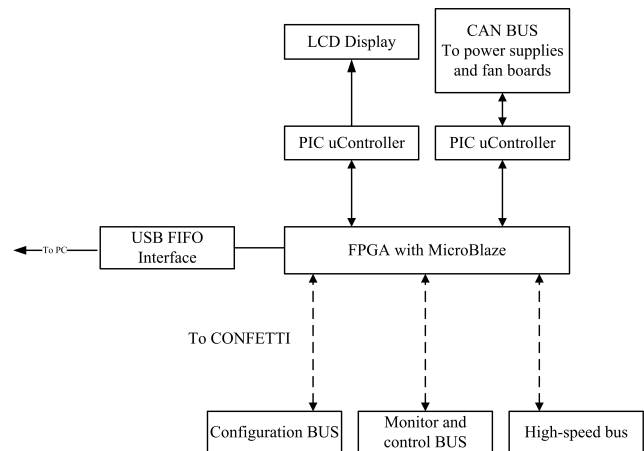


Figure 7. Schematic of the monitor and control board.

Because of the relatively high price of the components and the low number of *UltraStack* boards produced, we took precautions to minimize the risks of components failure due to short circuits or thermal stress.

Thus, each power supply works in limited current mode, in which 20 A under 5 V can be sourced. Moreover, despite the fact that the whole system can be used in a configuration where it is completely independent of any external control

mechanism, a supervisor board has been developed in order to monitor the whole system but also to help during the debugging phase of the system (Fig. 7). This board, which lies next to the *CONFETTI* system, comprises several elements:

- A SPARTAN[®] 3 FPGA in which a Microblaze CPU has been instantiated.
- An USB interface chip for the Microblaze CPU that permits transmission of configurations and data.
- A CAN protocol controller, managed by another micro-controller, and that manages the different power supplies and fans of the system.

5 Conclusions and Future Work

In this paper, we described a novel hardware platform aimed at the realization of cellular computing applications ranging from massively parallel computing through the exploration of various routing paradigms to bio-inspired computing. The versatility of the platform along with the potential computational power it can provide offer very interesting perspectives for future developments.

For example, should more processing power be needed, the *ECell* boards could easily be replaced by a bigger reconfigurable circuit or a different kind of circuit. Similarly, from the software perspective, the system's modularity implies that few changes would be required, for example, to allow different types of *ECell* boards on the same *ERouting* substrate. The exploration of this kind of modularity is currently under way, using the complete *CONFETTI* that has been built and tested.

Aside from the computational aspect, the system is also open to several improvements related to I/O aspects. For example, it is clear that the display available on each *UltraStack* will not be sufficient for many types of application and, in that case, it would also be relatively simple to add an external screen to display more complex *ECell* computations. And on a similar note, a planned improvement to the system is the introduction of high-speed I/O boards that, placed on the borders of the array, would allow the implementation of data-intensive applications (video streaming, for example).

In addition to hardware improvements, work is under way to endow the board with the necessary set of routing tools and to implement applications that can exploit the features of the system.

Acknowledgements

This project was funded by the Swiss National Science Foundation grant number PP002-68674 and by the Leenaards Foundation, Lausanne, Switzerland.

References

- [1] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, J. Thomas F. Knight, R. Nagpal, E. Rauch, G. J. Sussman, and R. Weiss. Amorphous computing. *Commun. ACM*, 43(5):74–82, 2000.
- [2] M. Amde, T. Felicijan, A. Efthymiou, D. Edwards, and L. Lavagno. Asynchronous On-Chip Networks. *IEE Proceedings Computers and Digital Techniques*, 152(02), March 2005.
- [3] M. Amos. *Cellular computing*. Oxford University Press, New York, 2004.
- [4] T. Bjerregaard and S. Mahadevan. A survey of research and practices of Network-on-chip. *ACM Comput. Surv.*, 38(1):1, 2006.
- [5] W. J. Dally and B. Towles. Route packets, net wires: on-chip interconnectoin networks. In *DAC '01: Proceedings of the 38th conference on Design automation*, pages 684–689, New York, NY, USA, 2001. ACM Press.
- [6] G. de Micheli and L. Benini. Networks on chip: A new paradigm for systems on chip design. In *DATE '02: Proceedings of the conference on Design, automation and test in Europe*, page 418, Washington, DC, USA, 2002. IEEE Computer Society.
- [7] K. Govil, D. Teodosiu, Y. Huang, and M. Rosenblum. Cellular Disco: resource management using virtual clusters on shared-memory multiprocessors. In *SOSP '99: Proceedings of the seventeenth ACM symposium on Operating systems principles*, pages 154–169, New York, NY, USA, 1999. ACM Press.
- [8] F. Moraes, N. Calazans, A. Mello, L. Möller, and L. Ost. HERMES: an infrastructure for low area overhead packet-switching networks on chip. *Integrated VLSI Journal*, 38(1):69–93, 2004.
- [9] A. Ngouanga, G. Sassatelli, L. Torres, T. Gil, A. Soares, and A. Susin. A contextual resources use: a proof of concept through the APACHES' platform. In *Proceedings of the 2006 IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, pages 44–49, April 2006.
- [10] D. Pham, T. Aipperspach, and D. B. et al. Overview of the architecture, circuit design, and physical implementation of a first-generation CELL processor. *IEEE Solid-State Circuits*, 41(1):179–196, 2006.
- [11] D. Pham, E. Behnen, M. Bolliger, H. Hostee, C. Johns, J. Kalhe, A. Kameyama, and J. Keaty. The design methodology and implementation of a first-generation CELL processor: a multi-core SoC. In *Proceedings of the Custom Integrated Circuits Conference*, pages 45–49. IEEE Computer Society, September 2005.
- [12] M. Sipper. The emergence of cellular computing. *Computer*, 32(7):18–26, July 1999.
- [13] M. Sipper and E. Sanchez. Configurable chips meld software and hardware. *Computer*, 33(1):120–121, January 2000.
- [14] G. Tempesti, D. Mange, A. Stauffer, and C. Teuscher. The BioWall: an electronic tissue for prototyping bio-inspired systems. In *Proceedings of the third Nasa/DoD Workshop*

- on *Evolvable Hardware*, pages 185–192, Long Beach, California, July 2001. IEEE Computer Society.
- [15] G. Tempesti, P.-A. Mudry, and G. Zufferey. Hardware/software coevolution of genome programs and cellular processors. In *AHS'06: Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06)*, pages 129–136, Washington, DC, USA, 2006. IEEE Computer Society.
 - [16] G. Tempesti and C. Teuscher. Biology Goes Digital: An array of 5,700 Spartan FPGAs brings the BioWall to "life". *XCell Journal*, pages 40–45, Fall 2003.
 - [17] C. Teuscher, D. Mange, A. Stauffer, and G. Tempesti. Bio-inspired computing tissues: Towards machines that evolve, grow, and learn. *BioSystems*, 68(2–3):235–244, February–March 2003.
 - [18] Y. Thoma. *Tissu Numérique Cellulaire à Routage et Configuration Dynamiques*. PhD thesis, EPFL, 1015 Lausanne, Apr. 2005. Thesis 3226.
 - [19] Y. Thoma, G. Tempesti, E. Sanchez, and J.-M. Moreno Arostegui. POetic: An electronic tissue for bio-inspired cellular applications. *BioSystems*, 74(1-3):191–200, Aug.-Oct. 2004.
 - [20] D. Wiklund and D. Liu. SoCBUS: Switched network on chip for hard real time embedded systems. In *IPDPS'03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, page 78.1, Washington, DC, USA, 2003. IEEE Computer Society.