A Programmable Analogue and Digital Array for Bio-inspired Electronic Design Optimization at Nano-scale Silicon Technology Nodes

Martin A. Trefzer, James A. Walker, Andy M. Tyrrell
Department of Electronics
University of York
York, YO10 5DD
Email: {mt540,jaw500,amt}@ohm.york.ac.uk

***Abstract*—Field programmable gate arrays (FPGAs) are widely used in applications where on-line reconfigurable signal processing is required. Speed and function density of FPGAs are increasing when shrinking transistor sizes to the nano-scale. Unfortunately, in order to reliably create electronic designs according to specification time-consuming statistical simulations become necessary due to effects of intrinsic variability. This paper describes an adaptive, evolvable architecture that allows for correction and optimisation of circuits directly in hardware using bio-inspired techniques. Like FPGAs, the programmable analogue and digital array (PAnDA) architecture introduced provides a digital configuration layer for circuit design. Accessing additional configuration options of the underlying analogue layer enables continuous adjustment of circuit characteristics.**

I. Introduction

A variety of reconfigurable hardware platforms exist, which serve a wide range of purposes. For example, digital commercial substrates have originally been designed merely as glue logic with the purpose of facilitating printed circuit board (PCB) design. However, over the past 20 years, these devices have rapidly improved. Current devices can be configured to implement entire digital systems including microprocessors and peripherals, which places them between processors and application specific integrated circuits (ASICs).

This development was founded on the feasibility to scale device sizes in perfect accord with Moores Law [1] (“happy scaling”). As a consequence, the mind set of both users and manufacturers of such devices have settled at a point where it is almost taken for granted that higher and higher densities of reliably operating logic can be achieved (see Figure 1). Great efforts are made to integrate more programmable general-purpose logic, but whenever there are tight constraints on bandwidth, power or speed this is currently resolved by incorporating full-custom optimised building blocks into the reconfigurable architecture rather than developing radically novel FPGA fabric structures. From an economical point of view this strategy makes sense, since industry cannot afford making drastic architectural changes to well-established approaches, as this would pose too much of a financial risk.

However, as device sizes are now approaching the atomistic level, intrinsic variations are becoming more abundant, leading to a lower production yield and higher failure rates [2]–[4]. This will also threaten the very building blocks that are used at the moment to alleviate the increasing demands for speed and low power. This is likely to put pressure on the industry to change their strategy. For example Intel were forced to make the biggest change in transistor technology since the 1960s in order to reach the 45nm CMOS technology node [5]. These predictions and issues were originally focussed on large-scale integration, mainly connected with microprocessor design. However, in the last 10 years the rise of field programmable devices, such as FPGAs, both in terms of technology advances and application domains has meant that these issues are now relevant to these devices as well.

Until now the design most affected by intrinsic variability has been SRAM [6]. Since SRAM is mainly used for storing configuration data and Look-up Tables (LUTs) in reconfigurable devices, which can be operated at relatively low speeds compared with the actual designs, FPGA fabric has not been as severely affected as other designs like memory and processors. However, it is projected that the next victim of variability after SRAM will be latches. This will have a direct impact on FPGA architectures, which comprise of a large number of Flip-Flops of which latches are an essential part.

We believe that in order to accommodate the increased variability of individual device characteristics there is a need for novel device architectures and circuit design methodologies. In order to further advance reconfigurable architectures, it is essential to identify and review current and future challenges both from an electronic design, and from an emergent technologies point of view. In order to overcome those challenges it is necessary to take a more holistic approach to electronic design problems, i.e. understanding and tackling problems on multiple levels rather than considering the device, analogue, digital or system levels separately. Particularly in the case of reconfigurable architectures there may be great opportunities to include mechanisms that allow for inherent correction and optimisation of shortcomings due to variability or other parameters.

In order to address those issues, this paper introduces the PAnDA (Programmable Analogue and Digital Array) ar-

978-1-4673-0323-1/11/$26.00 ©2011 IEEE

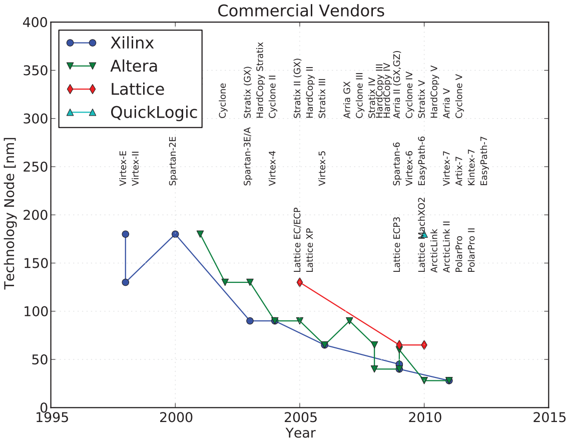

Fig. 1. The figure shows the adoption of smaller semiconductor technology nodes by major FPGA vendors.

chitecture, which proposes a novel reconfigurable variability tolerant architecture (illustrated in Figure 2), which allows variability aware design and rapid prototyping by exploiting the configuration options of the architecture. It is believed that these are vital steps towards the next generation of FPGA architectures.

A prototype of the PAnDA architecture currently exists as a simulation model, which is due to be fabricated in a 40nm process the first quarter of 2012.

II. PAnDA Architecture

It has been discussed in Section I how stochastic variability in CMOS transistors impedes reliably designing logic standard cells, which are the fundamental building blocks of FPGAs. Most severely affected designs are SRAMs and latches, which are fundamental elements of any current programmable logic architecture. The results in [7], which have been obtained from statistical SPICE simulations, suggest that optimising the widths of transistors in standard cells may improve their variability tolerance, speed and power consumption. It is also shown that it is possible to design and optimise analogue CMOS circuits in hardware using field programmable transistor arrays (FPTAs) [8], [9]. Therefore, if FPTA based mechanisms to alter device sizes according to [7] could be incorporated in hardware, it would be possible to optimise designs post fabrication. This would not only have the advantage of being able to enhance variability tolerance and performance for a specific design, but could also account for variations between different devices. In addition, due to the large numbers of statistical measurements necessary it will be orders of magnitude faster to perform optimisation directly in hardware, rather than in SPICE simulation.

With the programmable analogue and digital array (PAnDA) we propose a novel FPGA architecture, which aims to overcome challenges arising when shrinking device sizes to the nano-scale. In order to achieve this, the PAnDA architecture

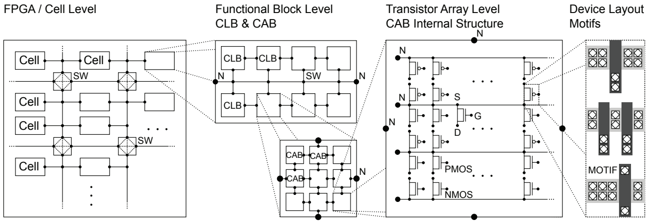

Fig. 2. The figure shows the hierarchical design structure of the PAnDA architecture. Cell, interconnect and CLB level are similar to current commercial FPGAs, whereas CT (transistor array level) and CAB are unique to PAnDA. While conventional FPGAs can only be reconfigured on the CLB and interconnect level, PAnDA offers additional configuration options on the analogue level.

features configurable transistors (CTs), the device sizes of which can be altered in wide ranges by configuring them in different ways. The hierarchical design of PAnDA is illustrated in Figure 2. As can be seen from the figure, the most important difference to a conventional FPGA is that the PAnDA architecture extends to a finer granularity of building blocks. The reason for introducing these additional levels is to both create and provide access to a lower design level, i.e. the transistor and function level, which enables tweaking the properties of the architecture on the analogue level in addition to the digital level.

The PAnDA architecture comprises configurable transistors (CTs), configurable analogue blocks (CABs), configurable logic blocks (CLBs), logic cells and interconnect. CLBs, logic cells and interconnect are also present—and have similar structure—in current commercial FPGA architectures, whereas CTs and CABs are unique to PAnDA. The designs of the the PAnDA CTs and CABs are introduced in this paper and are described in the following sections.

A. *Configurable Transistor (CT)*

The CT is the smallest reconfigurable element of the PAnDA architecture. As can be seen from Figure 3, the CT is a device formed from 7 PMOS (or 7 NMOS) transistors (M0…M6) that are connected in parallel. Each of the 7 transistors can be individually turned on or off via opening or closing a switch (S0…S6) that connects their gate to a common gate connection (CG). The states of the switches are controlled via configuration bits stored in a configuration static random access memory (SRAM).

This design exploits the fact that a number of CMOS transistors of the same gate length (L) can be connected in parallel in order to form a device that is equivalent to a single transistor of the same length and of the sum of the individual widths (W). For example, connecting two transistors in parallel where the size of one is $\frac{W}{L} = \frac{120\ \text{nm}}{40\ \text{nm}}$ and the size of the other is $\frac{W}{L} = \frac{180\ \text{nm}}{40\ \text{nm}}$, would result in a device that is equivalent to a transistor with $\frac{W}{L} = \frac{300\ \text{nm}}{40\ \text{nm}}$.

Providing more reconfiguration capabilities, however, comes at the cost of adding additional configuration circuitry, which introduces parasitic effects that may degrade the performance

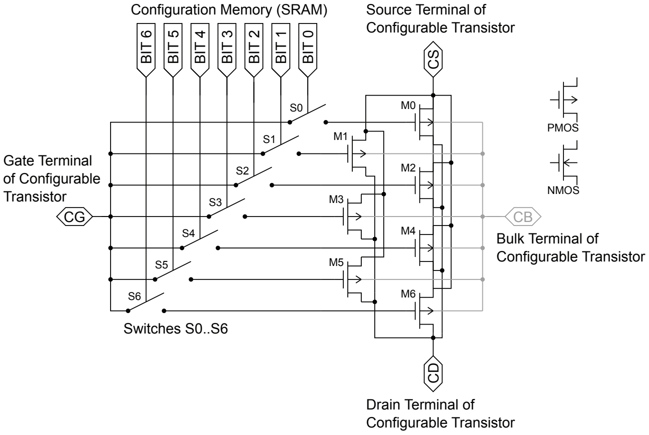
Configuration Memory (SRAM)
BIT 6
BIT 5
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0
Source Terminal of Configurable Transistor
CS
S0
S1
S2
S3
S4
S5
S6
M0
M1
M2
M3
M4
M5
M6
PMOS
NMOS
Gate Terminal of Configurable Transistor
CG
CB
Bulk Terminal of Configurable Transistor
Switches S0..S6
CD
Drain Terminal of Configurable Transistor

Fig. 3. The figure shows a configurable transistor (CT) of the PAnDA architecture. The transistor sizes used in the PAnDA CT are $L_{0\ldots6} = 40$ nm and $W_{0\ldots6} = [120, 120, 140, 160, 180, 200, 220]$ nm, which allows for configuring the CT a range of 120 nm and 1140 nm

of the CT. Hence, there is always a trade-off to be made between lowering the performance and increasing variability and fault tolerance of the design. In this case, the ratio between transistors that process user signals and those used for configuration purposes is 1 : 11. This number is calculated by comparing the 7 ”user” transistors with the number of transistors required for implementing switches (5T) and configuration SRAM (6T). Note that this number will slightly increase when also taking memory controllers and interconnect into account at the CAB and CLB level. The relevant numbers are given in Sections II-B and II-C.

The transistor sizes used in the PAnDA CT are $L_{0\ldots6} = 40$ nm and $W_{0\ldots6} = [120, 120, 140, 160, 180, 200, 220]$ nm. Minimum width and length are constrained by the smallest device sizes that are allowed according to the design rules of the 40 nm technology in which the PAnDA prototype will be fabricated. The increment of 20 nm is chosen as half the minimum feature size of the technology used, which should provide a suitable value range when optimising for effects of stochastic variability. Maximum sizes are based on maximum drive strengths in the order of 10 X. . . 30 X (a drive strength of 10 X means that the output of a circuit can produce enough output current to drive up to 10 logic cells attached), which are expected to be required when mapping designs. The different sizes allow the CT to be configured with 47 different widths between 120 nm and 1140 nm, and there is generally more than one possible combination of transistors to achieve a certain width of the CT, as illustrated in Figure 4. This redundancy may help when optimising for stochastic variability, since it is expected that different combinations for the same width of the CT will exhibit different behaviour in terms of variability. Note that the special case where $W = 0$ nm corresponds to turning the CT off. This is an essential feature, since not all CT are always used to implement higher level functions.

Note that since there are two types of transistors, NMOS and PMOS, there are also two types of CTs, CT-NMOS and CT-PMOS, depending on which type is used for the 7 ”user” transistors.

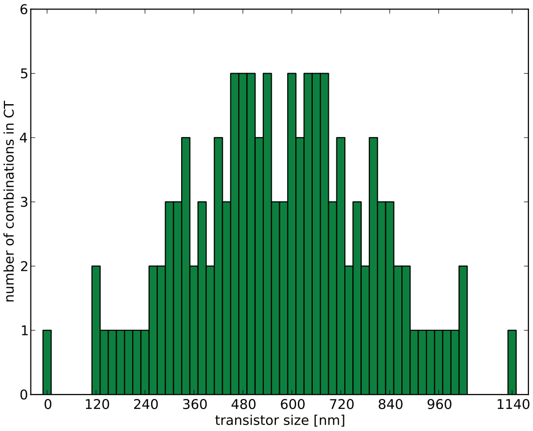

Fig. 4. The figure shows a configurable transistor (CT) of the PAnDA architecture. Different combinations of the 7 parallel transistors M0. . . M6 allow to configure the CT with 47 different widths between 120 nm and 1140 nm.

B. Configurable Analogue Block (CAB)

Configurable analogue blocks (CABs) represent the next higher level of entities to the CTs. With conventional CMOS design in mind, a PAnDA CAB is formed from 4 CT-NMOSs, 4 CT-PMOSs and configurable interconnect, as shown in Figure 5. A CAB features 7 analogue input/output (I/O) pins. The purpose of a CAB within the PAnDA architecture is to represent basic logic blocks that may implement FPGA fabric. In this case, it is possible to realise either 3 inverters, a buffer, a transmission gate, a logic gate (NOR, NAND, OR, AND), a latch, or a SRAM with a single CAB. The number of IO pins is the minimum number required to implement these logic functions. An example where the CAB is configured as a logic AND gate is shown in Figure 6.

In order to keep the ratio between transistors that process user signals and those used for configuration purposes low also on CAB level, the configuration options of the CAB have been constrained in such a way that the basic functions, which can be realised, are restricted to the ones mentioned before. Although this saves configuration circuitry and improves the performance of the design, the ratio for the CAB is 1 : 20, which is slightly higher than in the case of the CT due to the additional configurable routing required to map the basic logic functions.

For instance, it is possible to create a 3-bit look-up table (LUT) using 17 CABs, where 8 are configured as SRAMs and 9 are configured as 3-to-8 decoder. Another example would be using 6 CABs configured as logic NAND gates to implement a positive-edge-triggered D-type flip-flop. The latter two examples represent the two fundamental building

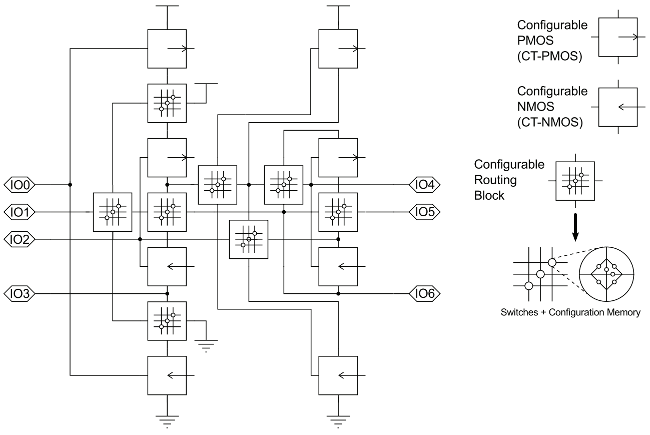

Fig. 5. The figure shows a configurable analogue block (CAB) of the PAnDA architecture. A CAB features 7 analogue input/output (I/O) pins and it is possible to realise either 3 inverters, a buffer, a transmission gate, a logic gate (NOR, NAND, OR, AND), a latch, or a SRAM with a single CAB.

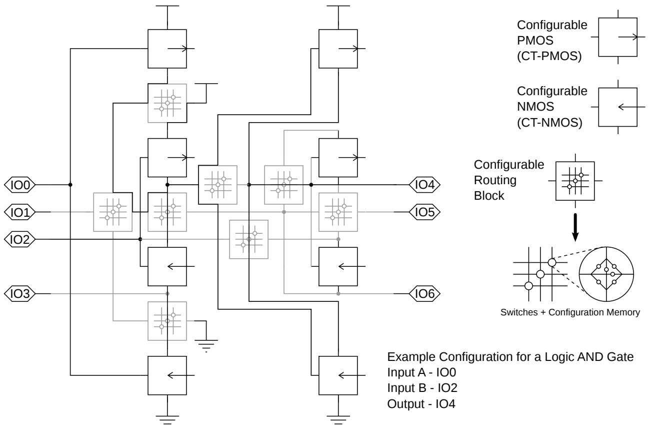

Fig. 6. The figure shows a CAB of the PAnDA architecture that is configured as an AND gate. In this case, the usage of the CAB is 75%.

blocks of current commercial FPGA architectures.

C. Configurable Logic Block (CLB)

This section follows on from the CABs to the next higher hierarchy level: the configurable logic blocks (CLBs). As described in Section II-B, a number of CABs can form LUTs, MUXs and flip-flops, which are the building blocks of current commercial FPGA architectures. For example, a CLB of a Xilinx Virtex-6 FPGA (www.xilinx.com), which is currently one of the top-of-range devices, comprises eight 6-bit LUTs, 16 flip-flops, eight 8-to-1 multiplexers (MUXs) and some carry/shift logic. Hence, one Virtex-6 CLB could be implemented using of the order of 700 CABs locally connected together.

Although this may sound like a large number of CABs required, the following must be considered: first, the Virtex-6 is a large, cutting-edge technology device and the resources incorporated in one CLB vary significantly between different models. For example, a CLB of the Virtex-4 could be implemented using of the order of 60 PAnDA CLBs. The reason the resource consumption increases dramatically in newer models is the transition from 4-bit LUTs to 6-bit LUTs that has been made after the Virtex-4 model range, which has lead to a significant increase in SRAMs per CLB. It is worth noting that the scenario looks similar in the case of Altera FPGAs (www.altera.com), such as the Cyclone V and the Stratix V (Altera is the second largest FPGA vendor). Second, the PAnDA CLB provides the additional feature of optimising for intrinsic variability, reducing propagation delay and minimising power consumption due to the CTs used (see Section II-A). This feature is not present in any of the FPGA architectures currently commercially available. Third, the reconfigurable nature of the PAnDA CAB, which also allows alteration of their function, may facilitate mapping of designs and improve the resource usage of the mapped designs due to the fact that functionality can be made available as required by the mapped design. For instance, a simple logic function could be represented in a more compact fashion using logic gates, rather than LUTs. Accordingly, a PAnDA CLB could be configured as either a larger 7-bit LUT, a large number of basic logic gates, or an array of 12 flip-flops by simply changing the function of the CABs. Again, current commercial FPGAs do not have this capability, which may lead to a less resource efficient mapping.

The first generation PAnDA chip is currently being implemented in a 40 nm CMOS process. Therefore, the number of PAnDA CABs per CLB is not yet determined. However, it will be at least in the order of 60, which would correspond to the complexity of a Xilinx Virtex-4. In addition to the logic function, CLBs also incorporate switch matrices that link them to local and global interconnect. The first generation of PAnDA is likely to follow a similar strategy as commercial FPGAs, which is a fixed crossbar architecture providing connections to the nearest neighbour CLBs as well as direct connections to resources that are further away within the FPGA fabric. Following a similar approach with the routing architecture should facilitate comparison of the PAnDA architecture with commercially available ones in the future.

III. Bio-inspired Design Optimisation on PAnDA

As dicussed in Sections II-A, II-B and II-C, PAnDA's facilities to optimise mapped designs at different design levels brings about a number of advantages, particularly when fabricating chips at nano-scale technology nodes (65nm and below) where the effects of stochastic variability and increased current leakage are prevalent. At the CLB level, it is possible to map logic designs in the same way as is done in commercial FPGAs. This makes both architectures compatible (and comparable) at this level. In addition, PAnDA offers the possibility to alter the CLB function set at the CAB level, which may lead to more efficient and compact mappings. Furthermore, at the PAnDA CT level, it is possible to alter the sizes of the transistors that make up the logic design.

This allows for on-line optimisation of the mapped design for variability tolerance, speed and power consumption without affecting overall functionality.

Future work will focus on developing electronic design automation methods for the PAnDA architecture. Based on previous work in the area of electronic design optimisation in hardware [9]–[13], it is envisioned to perform the mapping of logic designs on the CLB level using commonly used approaches that are based on metrics, principle component analysis, graph partitioning, and simulated annealing. Extending the mapping with the ability to automatically allocate CAB functions may be achieved via parametrising the latter approaches and optimising the parameters using global optimisation techniques from the field of evolutionary computation (EC). In addition, EC techniques may also be suitable for device size optimisation on the CT level. Examples of EC optimisation algorithms are genetic algorithms (GAs), evolutionary strategies (ES) and multi-objective optimisation (MO) [11], [12].

IV. Conclusion

This paper introduces a novel FPGA architecture, namely the PAnDA (programmable analogue and digital array) architecture. PAnDA is a hierarchical architecture, which comprises configurable transistors (CTs), configurable analogue blocks (CABs), configurable logic blocks (CLBs), and interconnect. At the highest CLB and interconnect level, PAnDA is similar to current commercial FPGA architectures. However, at lower levels it is the presence of CTs and CABs, which are unique to PAnDA. These additional levels of granularity provide access to a lower level of electronic design, i.e. the transistor (CT) and function (CAB) level, which enables tweaking the properties of the architecture at the analogue level as well as at the digital level (CLBs).

It has been shown in simulation [7] that altering transistor sizes can improve variability tolerance, speed and power consumption of digital designs. Therefore, providing the same features in reconfigurable hardware should result in an architecture that is armed with the facilities to overcome challenges arising when shrinking device sizes to the nano-scale, where effects of stochastic variability and increased leakage impede reliable electronic design. Furthermore, PAnDA's CAB architecture allows adaptation of the functionality of FPGA resources available to the demands of a design that is being mapped. This may facilitate mapping of designs and improve their compactness, thereby utilise FPGA resources more efficiently.

It is intended that the PAnDA architecture will close the gap between the analogue design of standard cells and the design of reconfigurable digital systems based on standard cell libraries, by providing a design platform that is reconfigurable on both the analogue and digital levels. The focus is to configure PAnDA with digital designs and optimise them in multiple stages. Firstly, by changing the location and topology of the digital components and secondly, by manipulating the properties and improving the intrinsic variability of parts of the circuit by changing the underlying analogue and device layers. The latter is a novel approach to synthesizing designs on an FPGA, and is not possible with any currently existing commercial FPGA. This will enable us to investigate the optimisation of digital circuits on multiple layers of abstraction using novel bio-inspired approaches to produce fault-tolerant and variability tolerant electronic designs.

The compatibility with commercial FPGAs on the CLB level, together with the additional configuration features on CAB and CT level and the possibility of applying post-mapping design optimisation techniques on those levels indicates significant potential of PAnDA to be a next-generation FPGA architecture.

Acknowledgment

This work is part of the PAnDA project that is funded by EPSRC (EP/I005838/1) and is the subject of a UK patent application (GB1119099.8).

References

[1] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, 1965.

[2] A. Asenov, "Variability in the Next Generation CMOS Technologies and Impact on Design," in *Proceedings of the 1st International Conference of CMOS Variability*, 2007.

[3] G. Declerck, "A look into the future of nanoelectronics," in *Symposium on VLSI Technology Digest of Technical Papers*. IEEE Press, 2005, pp. 6–10.

[4] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Proceedings of the 40th annual Design Automation Conference (DAC)*. ACM Press, 2003, pp. 338–342.

[5] M. T. Bohr, R. S. Chau, T. Ghani, and K. Mistry, "The High-k Solution," *IEEE Spectrum*, vol. 10, pp. 1–6, 2007.

[6] A. Asenov, "Statistical Nano CMOS Variability and Its Impact on SRAM," in *Extreme Statistics in Nanoscale Memory Design*. Springer Berlin Heidelberg, 2010, pp. 17–50.

[7] J. A. Walker, J. A. Hilder, D. Reid, A. Asenov, S. Roy, C. Millar, and A. M. Tyrrell, "The evolution of standard cell libraries for future technology nodes," *Genetic Programming and Evolvable Machines*, vol. 12, no. 3, pp. 235–256, Apr. 2011.

[8] J. Langeheine, M. A. Trefzer, J. Schemmel, and K. Meier, "Intrinsic Evolution of Analog Electronic Circuits Using a CMOS FPTA Chip," in *Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems (EUROGEN)*, 2003.

[9] A. Stoica, R. Zebulum, D. Keymeulen, R. Tawel, T. Daud, and A. Thakoor, "Reconfigurable VLSI architectures for evolvable hardware: from experimental field programmable transistor arrays to evolution-oriented chips," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 1, pp. 227–232, 2001.

[10] P. F. Vieira, L. B. Botelho, and A. Mesquita, "Evolutionary Synthesis of Analog Circuits Using Only MOS Transistors," in *Proc. of the 2004 NASA/DoD Conf. on Evolvable Hardware*, A. Zebulum, Ricardo S., Gwaltney, David, Hornby, Gregory, Keymeulen, Didier Lohn, Jason, and Stoica, Ed. Los Alamitos: IEEE Computer Society Press, 2004, pp. 38–45.

[11] A. M. Tyrrell, E. Sanchez, D. Floreano, G. Tempesti, D. Mange, J.-m. Moreno, J. Rosenberg, and R. Ep, "POEtic Tissue: An Integrated Architecture for Bio-inspired Hardware," pp. 129–140, 2003.

[12] M. Trefzer, J. Langeheine, J. Schemmel, and K. Meier, "New Genetic Operators to Facilitate Understanding of Evolved Transistor Circuits," in *Proc. of the 2004 NASA/DoD Conf. on Evolvable Hardware*, R. S. Zebulum, D. Gwaltney, G. Hornby, D. Keymeulen, J. Lohn, and A. Stoica, Eds. IEEE Computer Society Press, 2004, pp. 217–224.

[13] A. Greensted and A. Tyrrell, "RISA: A hardware platform for evolutionary design," in *IEEE Workshop on Evolvable and Adaptive Hardware, 2007. WEAH 2007*, 2007, pp. 1–7.