On the Evolution of Analog Electronic Circuits Using Building Blocks on a CMOS FPTA

Jörg Langeheine, Martin Trefzer, Daniel Brüderle,
Karlheinz Meier, and Johannes Schemmel

University of Heidelberg, Kirchhoff-Institute for Physics,
INF 227, D-69120 Heidelberg, Germany,
ph.: ++49 6221 54 9838 `langehei@kip.uni-heidelberg.de`
`http://www.kip.uni-heidelberg.de/vision/projects/eh/index.html`

Abstract. This article summarizes two experiments utilizing building blocks to find analog electronic circuits on a CMOS Field Programmable Transistor Array (FPTA). The FPTA features 256 programmable transistors whose channel geometry and routing can be configured to form a large variety of transistor level analog circuits. The transistor cells are either of type PMOS or NMOS and are arranged in a checkerboard pattern. Two case studies focus on improving artificial evolution by using a building block library of four digital gates consisting of a NOR, a NAND, a buffer and an inverter. The methodology is applied to the design of the more complex logic gates XOR and XNOR as well as to the evolution of circuits discriminating between square waves of different frequencies.

1   Introduction

The design of complex competitive analog electronics is a difficult task. In fact, to date existing technologies fail to automatically synthesize new transistor level circuit topologies for problems of medium or high complexity ([1] contains an overview of recent efforts). In engineering science, if a problem is hard to solve, usually a divide and conquer approach is used to simplify it. This leads to hierarchical approaches as e.g. described in [2]. Unfortunately, the division into subproblems often is a nontrivial task itself. Another approach, corresponding to the bottom up design principle, is to use functional subunits (building blocks) and assemble them to form solutions to more complex problems as for example done in [3].

The Field Programmable Transistor Array (FPTA) utilized in this work is a fine grained analog substrate dedicated to hardware evolution that offers a fairly high degree of complexity (cf. [4] for an overview of existing hardware). Hence, it is well suited to host hardware-in-the-loop experiments that can take advantage of both worlds: Find new circuit solutions exploiting transistor physics as well as accelerate the evolution process by using (predefined) building blocks and thereby relieving the evolutionary algorithm of reinventing substructures that have been proven useful in analog circuit design.

In order to test the proposed building block concept, a small library of well known building blocks that are well suited to the posed problem is sought. In this regard, the evolution of the analog dc behavior of the more complex gates XOR and XNOR by means of a building block library comprising the four simple logic gates NOR, NAND, inverter and buffer is considered a good test case. On one hand, the used building blocks are known to be useful for the design of the XOR/XNOR gates. On the other hand, the evolution of the analog behavior of XOR/XNOR gates can be easily stated in terms of the fitness function and is considered to be nontrivial, because it is not linearly separable. In a second case study, the same building block library is used to enhance the evolution of circuits that distinguish between square waves of different frequencies referred to as tone discriminators (TDs) (cf. [5]). In contrast to the first test problem, the usefulness of the used building blocks is not obvious at all in this case, because tone discrimination (in the absence of an external clock) is an inherently analog problem.

2   Evolution System

The used evolution system can be divided into three main parts: The actual FPTA chip serving as the silicon substrate to host the candidate circuits, the software that contains the search algorithm running on a standard PC and a PCI interface card that connects the PC to the FPTA chip. The software uploads the configuration bit strings to be tested to the FPTA chip via the PCI card. In order to generate an analog test pattern at the inputs of the FPTA chip, the input data is written to the FPGA on the PCI interface card. There it is converted into an analog signal by a 16 bit DAC. After applying the analog signal to the FPTA, the output of the FPTA is sampled and converted into a digital signal by means of a 12 bit ADC. The digital output is then fed back to the search algorithm, which in turn generates the new individuals for the next generation.

2.1   FPTA Chip

The FPTA consists of 16×16 programmable transistor cells. As CMOS transistors come in two flavors, namely N- and PMOS transistors, half of the transistor cells are designed as programmable NMOS transistors and half as programmable PMOS transistors. P- and NMOS transistor cells are arranged in a checkerboard pattern as depicted on the left hand side of Fig. 1.

Each cell contains the programmable transistor itself, three decoders that allow to connect the three transistor terminals to one of the four cell borders, *vdd* or *gnd*, and six routing switches. A simplified block diagram of the transistor cell is shown on the right hand side of Fig. 1. Width W and Length L of the programmable transistor can be chosen to be $1, 2, \ldots, 15\,\mu$m and $0.6, 1, 2, 4, 8\,\mu$m respectively. The three terminals *drain, gate* and *source* of the programmable transistor can be connected to either of the four cell borders named after the four cardinal points, *vdd* or *gnd*. The only means of routing signals through the
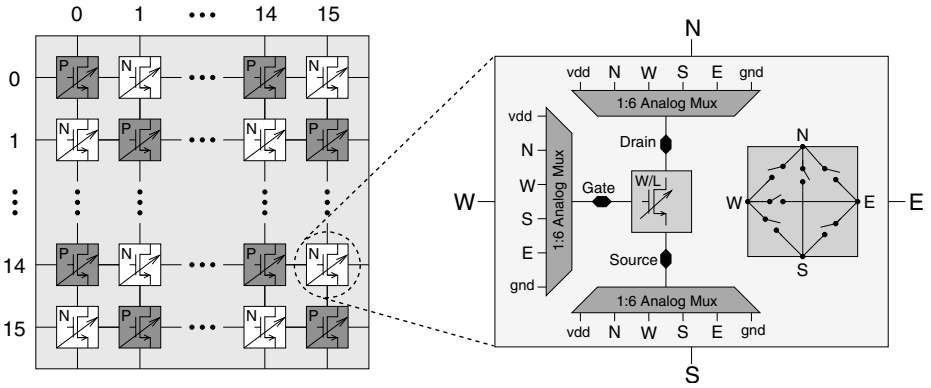
Fig. 1. Left: Schematic diagram of the 16 x 16 programmable transistor cell array. **Right:** Close-up on one NMOS transistor cell.

chip is given by the six routing switches that connect the four cell borders with each other. Thus, in some cases it is not possible to use a transistor cell for routing *and* as a transistor. More details on the FPTA can be found in [6].

2.2   Evolutionary Algorithm

The experiments of all three case studies were performed employing a straight forward genetic algorithm implementation in conjunction with a truncation selection scheme. In order to keep the algorithm stable in case of noisy and/or unreliable fitness measurements, relatively large values for the *reproduction fraction* (the fraction of the population that is moved to the new generation unchanged) are used, as can be seen from Table 1.

**Table 1.** Genetic algorithm parameters used throughout the presented experiments.

GA Parameter	TD:BB	TD: Cell	X(N)OR: BB	X(N)OR: Cell
generation size	50	50	50	50
reprod. fraction	0.3	0.2	0.3	0.2
mutation fraction	0.4	0.4	0.4	0.4
crossover fraction	–	0.5	–	0.5
crossover rate	0	1 %	0	1 %
mut. rate routing	4 %	1 %	4 %	1 %
mut. rate W/L	3 %	1 %	3 %	1 %
mut. rate term. con.	–	1 %	–	1 %
mut. rate BB	2 %	–	1 %	–
no. of used blocks	16	–	16	–
no. of used cells	112	64	112	64
no. of generations	10000	10000	5000	5000

3   Building Block Concept

The standard genotype representation reflects the structure of the FPTA chip: For each cell the transistor geometry, its terminal connections as well as the state of the routing switches can be mutated individually. The crossover operation is cell based: A rectangular array of cells is copied from one individual to the same location of another individual producing one offspring. A more detailed description of the underlying representation can be found in [7].

This representation is extended for the usage of building blocks (BBs) by introducing genetic access rights for any of the transistor cells and by new crossover and mutation operators: The access rights define the genetic operations the EA is allowed to apply to the according cell. The new crossover operation preserves the building block structure, i.e., the chosen crossover blocks are extended such that they embed all partially covered BBs. The new mutation operator replaces a randomly selected BB of the genotype with one randomly chosen from the building block library. The first generation is initialized with BBs chosen randomly from the used library and randomly configured transistor cells (according to the used genetic access rights). As a result, the genotype can be freely divided into BB sites and simple transistor cells that can be altered in only exactly the ways defined in the particular experimental setup.

For the case studies a library of four simple logic building blocks implemented using 3×3 transistor cells is utilized. Both experiments use the complete chip and a total of 16 building block sites, as depicted in Fig. 2. While genetic operations
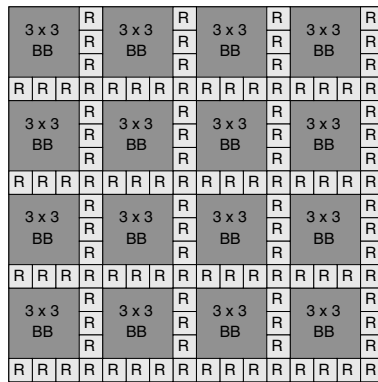

Fig. 2. Geometrical setup for case studies I,II. The R denotes routing cells.

for the cells denoted with an R are restricted to changes of their routing, the GA is allowed to change the channel dimensions W and L for the cells reserved for the BBs. On insertion, all transistors of all BBs possess an aspect ration of W/L = 4/2. The input signals are applied to the left hand side, the circuit's output is measured on its right hand side. The crossover operation was omitted

in the experiments using BBs. However, the reference experiments using the pure transistor cell implementation did use crossover, but were restricted to 8×8 cells to constrain the design space to a size comparable to that of the building block experiments.

3.1 Logic Gate Library

Fig. 3 illustrates the four logic gates making up the used building block library. Each block possesses two inputs A and B at its western edge, which are short
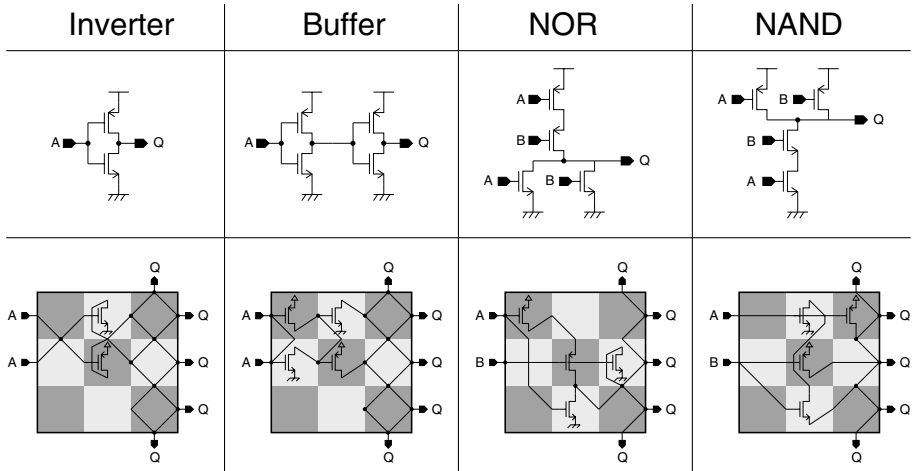
Fig. 3. Building block library used for case studies I and II. The second row shows the schematics of the used circuits and the third one displays their implementation as a block of 3×3 transistor cells. PMOS transistor cells are shaded in darker gray than their NMOS counterparts.

circuited in case of the inverter and buffer implementations. The output Q is available at five terminals at the eastern side. Thus, the proposed building blocks support the aforementioned signal flow from left to right that is used throughout all experiments.

4   Case Study I: XOR and XNOR

As a first test of the building block concept the BB library shown in Fig. 3 is used to evolve the more complex logic gates XOR and XNOR. A total of four experiments each featuring 30 runs were carried out, two using the described building block setup of Fig. 2 and two using the plain cell genotype respectively. For the experiments using plain transistor cells the array provided to the GA was restricted to 8×8 cells. Both input voltages, V_{in1} and V_{in2}, are applied to the western side of the array while the output is measured at the opposite side.

4.1 Experimental Setup

All experiments are run at a fixed generation size of 50 individuals and a number of generations of 5000. During evolution, the used test pattern consists of a set of eight curves with $V_{in1} = 0 \ldots 2\,V, 3 \ldots 5\,V$ each in 4 steps and $V_{in2} = 0 \ldots 2\,V, 3 \ldots 5\,V$ each in 16 steps. A target voltage of $V_{tar} = 0\,V$ corresponds to the logic zero and $V_{tar} = 5\,V$ to the logic one. The input voltage range between 2 and 3 V, where the gate switches its output, is not of interest for the application of logic gates and therefore not covered in the test pattern. Moreover, it would constrain any possible solution more severely than necessary. The sample voltages are applied in randomly chosen random orders with a sample frequency of 244 kHz. Hence, the settling time must be less than 4.1 μs. For measuring the voltage characteristics of the evolved logic gates, a modified test pattern is used that covers the full range of $V_{in2} = 0 \ldots 5\,V$, thus including the transition region.

4.2 Fitness Calculation

Throughout all experiments the sum of squared errors is used as the fitness criterion:

$$\text{Fitness} = \sum_{i=1}^{256} \left(V_{tar}(i) - V_{out}(i)\right)^2 \quad . \tag{1}$$

Hence, the GA has to minimize this fitness. However, in order to add a physical meaning to the fitness measure, the fitness is converted to the root mean square error per data point in mV for all results presented in the remainder of this section:

$$\text{RMS Error } [\text{mV}] = \sqrt{\frac{\text{Fitness}}{256}} \cdot 1000 \quad . \tag{2}$$

4.3 Evolution Results

The fitness values cover a theoretical range of $0 \ldots 5000\,\text{mV}$. Practically, typical random individuals that are used for the initialization of the population obtain a fitness of about $2500 \pm 500\,\text{mV}$; a circuit that exhibits the exact inverse of the target behavior is as improbable as the desired one.

Comparison of the Results of the Different Experiments. The *RMS error* values of all experiments are shown in the histograms in Fig. 4. The results confirm that, as expected, the building blocks extensively help the GA in finding good solutions for more complex logic gates. While the results of the experiments using the standard representation are comparable to those presented in [7], the use of building blocks boosts the rate of runs finishing with the desired output behavior from 0 to more than 80%.
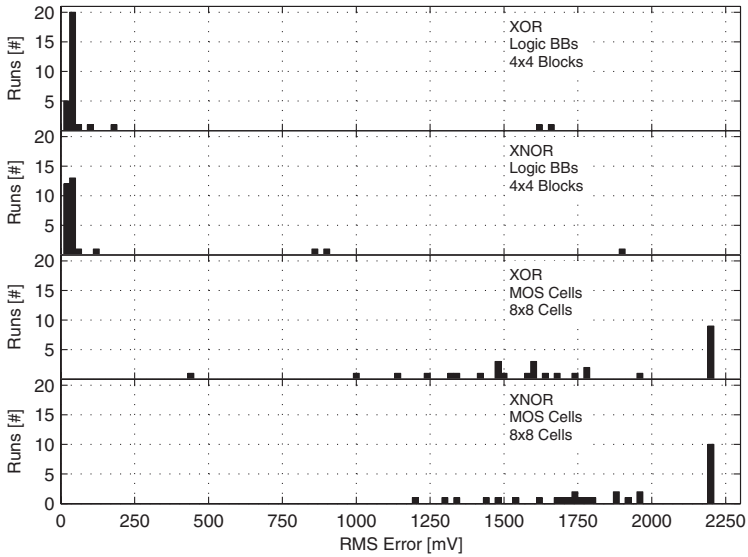
Fig. 4. Comparison of the achieved fitness values of 30 runs per XOR/XNOR experiment.

Voltage Characteristics of the Evolved Logic Gates. While the transition region was not considered during evolution, it is measured and plotted for the best circuits of each experiment in Fig. 5 to obtain information about the complete voltage characteristic of the evolved gates. Both, the best XOR as well as the best XNOR gate evolved with building blocks exhibit an output voltage characteristic that perfectly matches the fitness criterion. Conversely, the XOR and XNOR evolved using the plain transistor cell genotype both fail to reach the voltage rails for at least one of the four logic input combinations. However, assuming a threshold of 2.5 V, they would manage to produce the correct logical result.

The fact that none of the circuits evolved without building blocks perfectly meets the target specifications can be explained as follows: For one, the fitness criterion may be too ambitious in that the region the gate is allowed to switch in is very narrow. In fact, an XOR/XNOR circuit from the standard cell library provided by the manufacturer of the used process technology would be evaluated with a non-zero fitness as shown in [7],[1]. For the other, the difficulty of the task is increased by the used representation that is closely related to the structure of the FPTA chip. Therefore, the design of the circuit and its physical layout on the FPTA chip have to be processed in one single step.
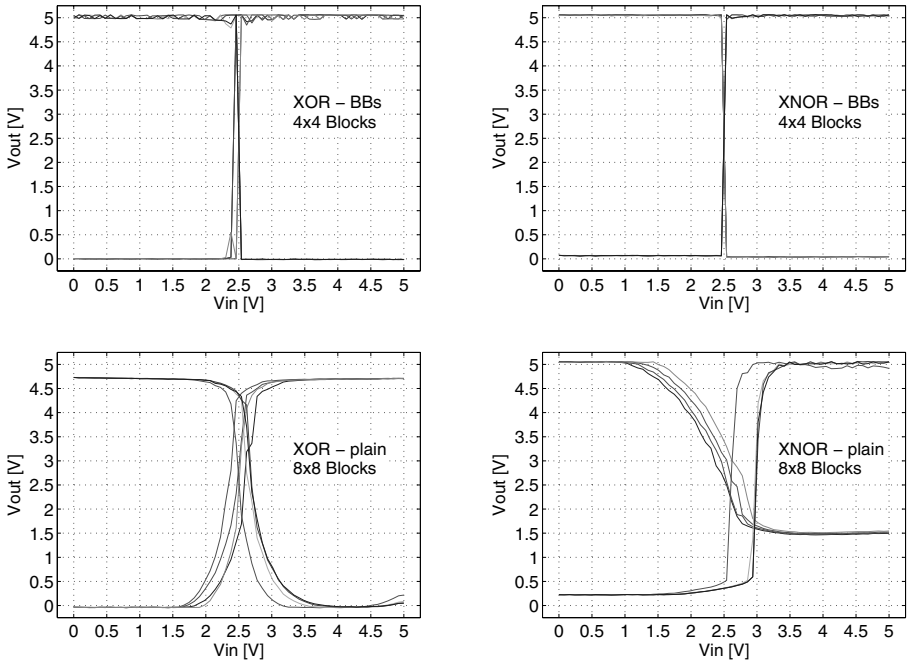
Fig. 5. Measured voltage characteristics of the evolved XORs and XNORs using Building Blocks and standard cells.

5   Case Study II: Tone Discrimination

The problem of discriminating square waves of different frequencies suits hardware evolution well: On one hand, the problem definition in terms of test patterns and fitness function is relatively simple; on the other hand, the design of an analog tone discriminator is a nontrivial task. Within the field of evolvable hardware the problem was first tackled by Adrian Thompson (see e.g. [5]) who used an FPGA to discriminate tones of 1 and 10 kHz.

5.1   Problem Definition

Test Pattern. In contrast to the original experiment, the frequencies to be distinguished were shifted to 40 and 200 kHz, in order to decrease the time necessary for one fitness evaluation. As can be seen from Fig. 6 the test pattern consists of 20 periods of the 200 kHz square wave followed by 4 periods of the 40 kHz one. This pattern is applied twice for each fitness test. The output is sampled with a frequency of 2 MHz, resulting in 800 test points and a total time of 400 μs. In order to prevent successful candidate solutions from exploiting the charge distribution left from the test of its predecessor, a randomly created gene
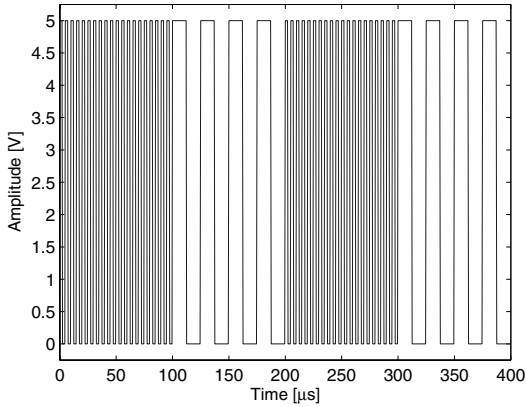
Fig. 6. The input pattern used for the evolution of tone discriminators. $2 \times$ (20 periods of 200 kHz + 4 periods of 40 kHz).

was written to the chip before the next candidate solution was downloaded and tested.

Fitness Function. During the evolution process the fitness is evaluated by

$$\text{Fitness} = \sum_{i=1}^{800}(V_{\text{tar}}(i) - V_{\text{out}}(i))^2 + 3\sum_{i=2}^{800}(V_{\text{out}}(i) - V_{\text{out}}(i-1))^2 \ , \qquad (3)$$

with the target voltage defined as

$$V_{\text{tar}} = \begin{cases} 0,5 & \text{for} \quad f = 200\,\text{kHz} \\ 5,0 & \text{for} \quad f = 40\,\text{kHz} \end{cases} \ . \qquad (4)$$

The actual $V_{\text{tar}}(f)$ is chosen to minimize the fitness value; thereby the GA is relieved of the constraint of finding a solution with a prescribed output polarity. While the left term of (3) yields the sum of squared deviations from the target voltage (4), the right sum penalizes unwanted glitches and oscillations of the output. The weighting factor of 3 was chosen based on the experience gathered in preliminary studies.

However, for the analysis within this section the fitness is calculated as the root mean square error per data point given in mV,

$$\text{RMS Error [mV]} = \sqrt{\frac{\sum_{i=1}^{800}(V_{\text{tar}}(i) - V_{\text{out}}(i))^2}{800}} \cdot 1000 \ , \qquad (5)$$

which adds a physical meaning to the fitness measure.

5.2   Results

The geometrical setup is identical to the one described in case study I and the structure of the two experiments – one using the building block representation described in Fig. 3, the other one the plain cell genotype – is similar to those of section 4. In order to acquire information about the reliability of the evolved circuits, the best individuals of all evolution runs were tested 100 times. Fig. 7 compares the results of both series using the worst fitness values measured during the verification tests. During the course of the experiments it was observed that
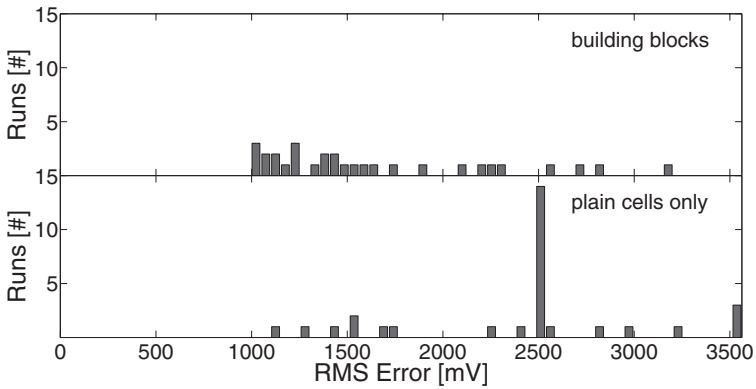

Fig. 7. Comparison of the worst fitness from 100 verification tests for the experiment with and without building blocks respectively.

the algorithm frequently chooses to clamp the output to 2.5 V. On one hand, this realizes the minimum RMS error without having to discriminate between the two frequencies. On the other hand, a circuit producing such a constant output voltage can easily be realized on the FPTA. Accordingly, all runs should finish with a fitness smaller than or equal to 2500 mV – the value resulting from applying (5) to the situation described above. In the histograms of Fig. 7 however, some circuits manage to behave even worse, which indicates that these solutions were performing better during evolution, but fail to work reliably under the verification test conditions.

The results suggest that the GA was more successful in finding tone discriminators of moderate quality when it was allowed to use BBs. The large peak in the histogram for the runs utilizing only plain transistor cells indicates that a large fraction of them got stuck in the local minimum described above.

The circuit responses of the best individual with and without BB usage are plotted in Fig. 8. The left half of the figure captures the circuit response to the test pattern used during evolution. In the right half, the output is plotted versus frequency, where output is defined as the output voltage averaged over one period of a square wave. As can be seen from Fig. 8, the best solutions found
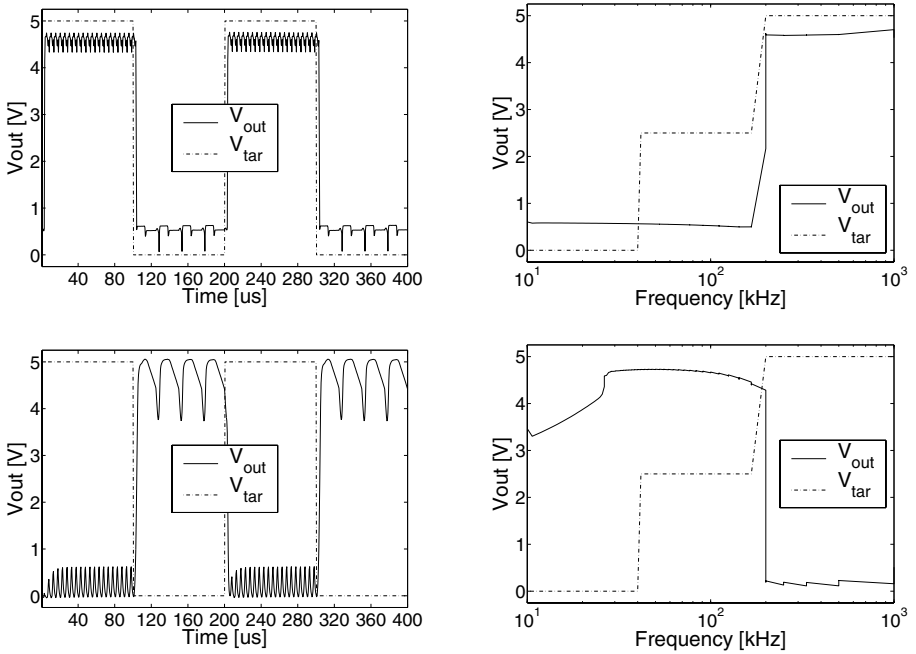
Fig. 8. Measured response of the best evolved tone discriminators: **Top**: using building blocks, **Bottom**: using transistor cells only, **Left**: Original fitness criterion. **Right**: Frequency Sweep. The outputs have different polarities with reference to the input frequency, which is allowed by the fitness criterion described above.

with and without BB usage do not differ significantly. Both solutions clearly distinguish between the two input frequencies but fail to reach the rails of the power supply range and carry a considerable amount of ripple. Considering the frequency sweep tests, it can be observed that both tone discriminators correctly distinguish between frequencies lower than approximately 200 kHz and those above 200 kHz in the measured frequency range of 10 kHz to 1 MHz. Since the best circuits obtained in this work are not as good as the results achieved in the original experiments documented in [5], it should be noted that both experiments do differ in a variety of ways. Most prominently, the FPGA used by Thompson was able to use a larger amount of resources to fulfill the task.

6   Discussion

The use of building blocks was introduced and tested in two case studies, namely the evolution of XOR/XNOR gates and the evolution of tone discriminating circuits. While the success rate as well as the performance of the best evolved circuits could be greatly enhanced in case of the gates, the building blocks mainly support the GA in finding solutions of moderate quality more frequently for the

evolution of tone discriminators. The latter result is remarkable insofar as the used building block library is far from being especially devised to the task of tone discrimination.

The proposed building block library of simple logic gates is not expected to be a particularly good choice to solve analog problems, but on the contrary, the choice of good building blocks is a key to efficiently solve a particular problem. Besides the actual functionality of the blocks the geometry of their in- and outputs as well as the geometrical setup they are embedded in are expected to play an important role. To find answers to these questions, future experiments will have to apply different building block libraries and topologies to a wider range of test problems. From the resulting data valuable information can be gathered about better FPTA cells and architectures that will eventually lead to a second generation FPTA.

References

1. Langeheine, J., Meier, K., Schemmel, J.: Intrinsic evolution of analog electronic circuits using a CMOS FPTA chip. In: Proc. of the 5th Conf. on Evolutionary Methods for Design, Optimization and Control (EUROGEN 2003), Barcelona, Spain, IEEE Press (2003) 87–88 Published on CD: ISBN: 84-95999-33-1.
2. Zebulum, R.S., Stoica, A., Keymeulen, D.: Experiments on the evolution of digital to analog converters. In: Proc. of the IEEE Aerospace Conference, Montana, USA (2001) ISBN: 0-78-3-6600-X (Published on CD).
3. Shibata, H.: Computer-Aided Design of analog Circuits Based on Genetic Algorithm. PhD thesis, Tokyo Institute of Technology (2001)
4. Zebulum, R.S., Stoica, A., Keymeulen, D.: A flexible model of a CMOS field programmable transistor array targeted for hardware evolution. In Miller, J., Thompson, A., Thomson, P., Fogarty, T.C., eds.: Proc. of the Third Int. Conference on Evolvable Systems: From Biology to Hardware (ICES2000), Edinburgh, UK, Springer (2000) 274–283 LNCS 1801.
5. Thompson, A., Layzell, P., Zebulum, R.S.: Explorations in design space: Unconventional electronics design through artificial evolution. IEEE Trans. Evol. Comp. **3** (1999) 167–196
6. Langeheine, J., Becker, J., Fölling, S., Meier, K., Schemmel, J.: Initial studies of a new VLSI field programmable transistor array. In: Proc. 4th Int. Conf. on Evolvable Systems: From Biology to Hardware, Tokio, Japan, Springer Verlag (2001) 62–73
7. Langeheine, J., Meier, K., Schemmel, J.: Intrinsic evolution of quasi dc solutions for transistor level analog electronic circuits using a CMOS FPTA chip. In: Proc. of the Fourth NASA/DOD Workshop on Evolvable Hardware, Alexandria, VA, USA, IEEE Press (2002) 76–85