

Initial Studies of a new VLSI Field Programmable Transistor Array

Jörg Langeheine, Joachim Becker, Simon Fölling, Karlheinz Meier, Johannes Schemmel

Address of principle author: Heidelberg University, Kirchhoff-Institute for Physics,
Schröderstr. 90, D-69120 Heidelberg, Germany,

langehei@kip.uni-heidelberg.de

WWW home page: <http://www.kip.uni-heidelberg.de/vision.html>

Abstract. A system for intrinsic hardware evolution of analog electronic circuits is presented. It consists of a VLSI chip featuring 16×16 programmable transistor cells, an FPGA based PCI card and a software package for setup and control of the experiment. The PCI card serves as a link between the chip and the computer that runs the genetic algorithm to produce the configurations for the Field Programmable Transistor Array (FPTA). First measurement results prove chip and system to be working as well as they indicate the tradeoff between performance and configurability. The system is now ready to host a wide variety of evolution experiments.

1 Introduction

While digital hardware is becoming more and more powerful, there are a lot of problems requiring analog electronic circuits. Examples are sensors (e.g. [1]), that will always use some analog front end to measure a physical quantity in an analog manner, analog filters or sometimes (massive parallel) signal processing circuits. For the latter example the use of analog circuitry can result in a better ratio of performance and area and/or power consumption (cf. e.g. [2], [3]). Unlike its digital counterpart the domain of analog design is not blessed with powerful tools simplifying the design process. This is, at least to some extent, due to the tight relationship between the used technology, the chosen layout and the performance of the resulting circuit, which makes the simple reuse of standard building blocks without any adaption virtually impossible. Moreover great care has to be taken in how the specific process parameters can be used to achieve the desired behavior because of the device variations on the actual dice. As evolutionary algorithms are assumed to yield good results on complex problems without explicit knowledge of the detailed interdependencies involved, they seem to be a tempting choice. Accordingly the project described in this paper tries to make a step towards the design automation of analog electronics by means of evolvable hardware.

From the variety of different approaches intrinsic evolution on a fine grained FPAA, namely a Field Programmable Transistor Array (FPTA) designed in

CMOS technology, is chosen for the following reasons: First, the use of hardware in the loop is expected to be advantageous because it faces the algorithm with the full complexity of the problem including device mismatching as well as any kind of electronic noise inherent to the chip. There is evidence that the presence of different environmental conditions during the evolution process is helpful to evolve circuits that work on different dice under different conditions (cf. [4], [5]). Second, intrinsic evolution is expected to be faster than evolution using software models for the hardware. Third, the use of large scale integration techniques facilitates the design of complex systems. CMOS nowadays is the most widely used and therefore cheapest technology for the design of integrated electronic circuits.

The final goal can be twofold: On one hand, it would be desirable to have a system that can be fed with an abstract problem description, such as a sort of fitness function, and that after some time produces a solution to the problem. Without caring about the details of the implementation the designer merely has to ensure that the circuit is working correctly under all expected conditions. On the other hand it may be useful to analyze the circuits obtained by the hardware evolution process and understand them to such an extent that it is possible to use the extracted circuits or design principles in a different chip, thus using the system as a design tool.

The paper is organized as follows: Section 2 gives an overview over the evolution system. In section 3 the implementation of the Field Programmable Transistor Array is discussed. Finally in section 4 experimental results are given and the expected performance of the chip is discussed, before the paper closes with a summary.

2 The Hardware Evolution System

Figure 1 shows the setup of the evolution system. A commercial PC is used to control the system and as a user interface. The software allows to create and edit circuit configurations for the FPTA chip. A PCI-card serves as the link between the FPTA board that can be plugged into the PCI-card and the computer. A state machine run on the FPGA generates all the necessary digital signals: It creates the signals used to write the configuration to the SRAM of the FPTA and performs the read out of the SRAM. Furthermore the state machine provides the DAC with the necessary data and timing signals to produce the analog input patterns for the FPTA and controls the data conversion of the analog outputs of the FPTA carried out by the ADC. The RAM module on the PCI-card can be used for example to cache the data for the analog input patterns, the output of the ADC and the next individuals to be loaded into the FPTA.

In figure 2 a screenshot of the user interface of the software is displayed. The right window contains 6×4 cells of the lower right corner of the transistor array, consisting of a total of 128 P- and 128 NMOS transistors arranged in a checkerboard pattern as denoted by the letters P and N in each of the cells. From this window any circuit can be downloaded to the chip in order to test it. The

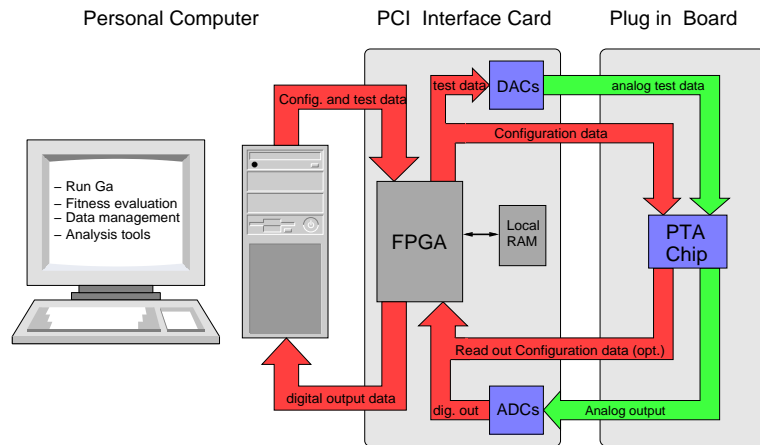


Fig. 1. Schematic diagram of the evolution system.

left window reflects the configuration of the cell 15/15 (cells are identified by their x/y coordinates): Each of the three terminals gate, source and drain of the MOS transistor can be connected to either the supply voltage, ground, or any of the four edges of the cell. Furthermore to enable signals to be routed through the chip any of the four cell edges can be connected to any of the remaining three edges.

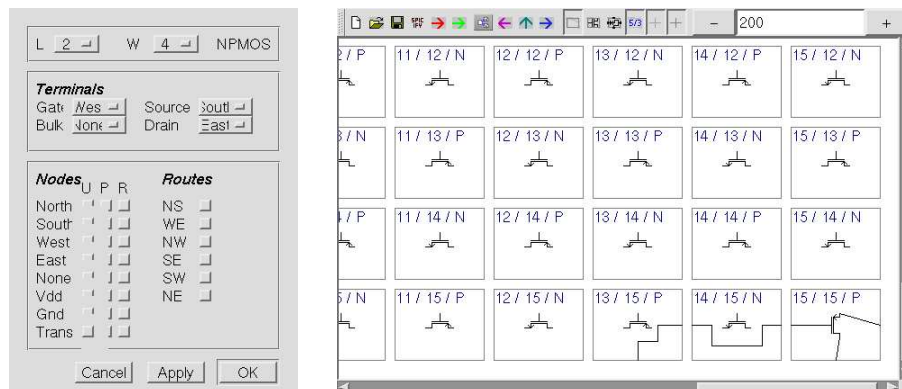


Fig. 2. Screenshot of the circuit editor window of the software: **Left:** Editor to set the connections and W/L values for one cell (here 15/15). **Right:** Editor showing the setup for the measurement of one PMOS transistor in the lower right corner of the chip.

3 Implementation of the FPTA

In order to provide some *primordial soup*, i.e. a configurable hardware device, for the intrinsic evolution a Field Programmable Transistor Array (FPTA) has been designed and manufactured in a $0.6\ \mu\text{m}$ CMOS process (More information can be found in [6]). Figure 3 shows a micro photograph of the chip whose die size is about $33\ \text{mm}^2$.

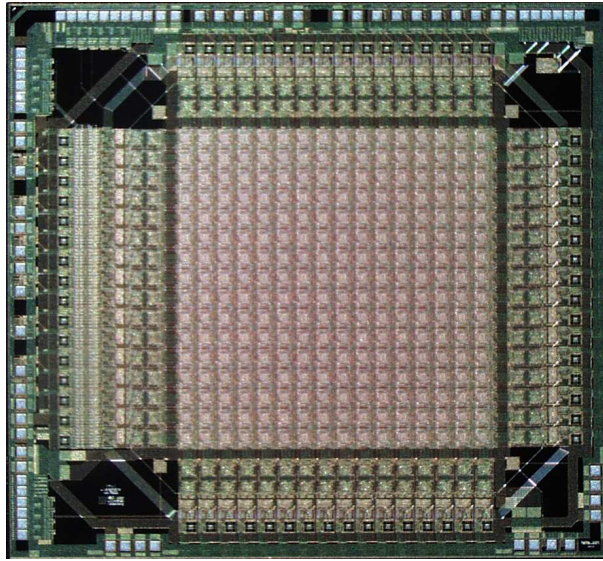


Fig. 3. Micro photograph of the FPTA chip.

The core of the chip consists of an array of 16×16 programmable transistor cells. These cells contain either a programmable P- or NMOS transistor, whose channel geometry can be tuned. The terminals of these transistors can be connected to the four neighboring cells. The signals from the adjacent cells can be routed through the cells.

The choice for this implementation is motivated as follows: First, it was desired to have distinct transistors, that contain the circuit functionality as transistors do in usual designs, in order to simplify the analysis of evolved circuits. Second, the array was designed as homogeneous and symmetric as possible to keep the implementation details of the evolutionary algorithms simple and to enable it to reuse parts of the genome by copying and translating it. However, a single cell was reserved for P- and NMOS transistors respectively to save die area. Third, the transistor geometry can vary in 5 logarithmically graded lengths and 15 linearly graded widths resulting in 75 different aspect ratios in order to obtain a smooth fitness landscape at least for choosing the transistor dimensions.

3.1 Architecture of the Complete Chip

The transistor cell array is surrounded by 64 IO-cells that are connected to the 64 terminals of the 60 transistor cells forming the edges of the array (see fig. 4). The functionality of the IO-cells can be selected by setting their registers. Possible settings are to connect the terminal of the according border cell directly to the analog input or output, directly to the according *array border pad*, leave it unconnected, or to access it via a sample and hold stage. The direct access granted by the array border pads serves two purposes: First, it simplifies debugging and allows direct measurement of the transistor cells. Second, the transistor array can be expanded by bonding together the array border pads of two or more chips. The array border pads are smaller than the standard pads used for analog signals to reduce their capacity and lack the ESD protection circuitry in order to contribute as little distortion as possible to the signals crossing a die border.

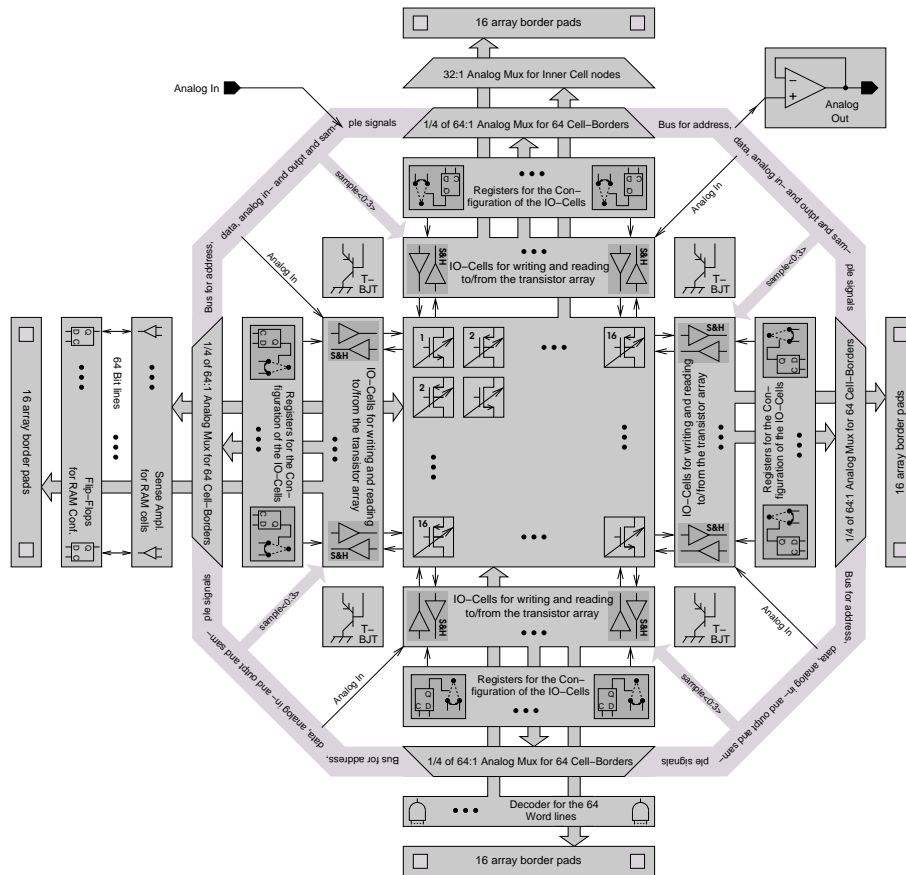


Fig. 4. Block diagram of the PTA chip. Note that the address and data buses are used for all multiplexers and demultiplexers as well as for the programming of the IO-cells.

Each sample and hold stage can be configured to either buffer an input voltage applied to the border terminal or to sample and hold the voltage present at the border cell. The cells configured in the former manner can be used to create complex input patterns from the single analog input. Therefore the sample signals can be taken from four external sample lines. Used as output buffers the sample and hold stages can be utilized to multiplex more than one border cell voltage to the analog output. Moreover they allow the successive read out of different outputs sampled at the same time.

The configuration of the transistor cell array is stored in static RAM cells that are integrated in the transistor cells. Both, read and write access to the SRAM and the configuration of the IO-cells use a 10 bit wide address and a 6 bit wide data bus, that are looped around the chip as shown in figure 4. Each transistor cell contains an operational amplifier that can buffer one out of four possible nodes in the according cell (cf. figure 5). These signals are used to determine voltages and currents inside the transistor cells and can also be multiplexed to the analog output line, which is buffered again before the output signal leaves the chip.

3.2 Architecture of the Transistor Cell

Figure 5 shows the setup of an NMOS cell. At each corner some of the configuration information is stored in a block of static RAM containing 6 bits each. Of the 22 bits used, 6 bits directly control the routing switches that route signals through the cell. Each terminal of the programmable transistor, whose channel geometry is set by 7 bits, can be connected to either power (vdd), ground (gnd) or any of the four edges of the cell, named after the four cardinal points. The remaining two codes of the multiplexers for drain and source are used to leave the terminals floating. For the gate the same code ties the gate terminal to power or ground for P- and NMOS transistors respectively, thus disabling the transistor.

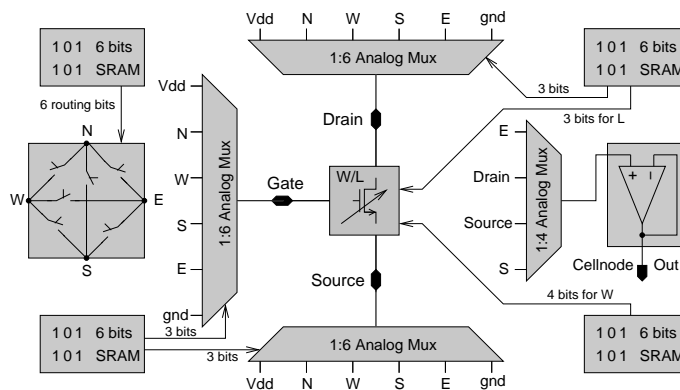


Fig. 5. Block diagram of one NMOS transistor cell.

In order to be able to analyze the behavior of successfully evolved circuits, the voltage at nodes *east*, *south*, *drain* and *source* can be read out by means of a unity gain buffer. Thereby all the nodes between adjacent cells can be read out and all currents flowing through the active transistors can be estimated via the voltage drop across the transmission gates connecting them to the cell borders. The layout of a complete transistor cell is shown in figure 6. It occupies an area of about $200\ \mu\text{m} \times 200\ \mu\text{m}$.

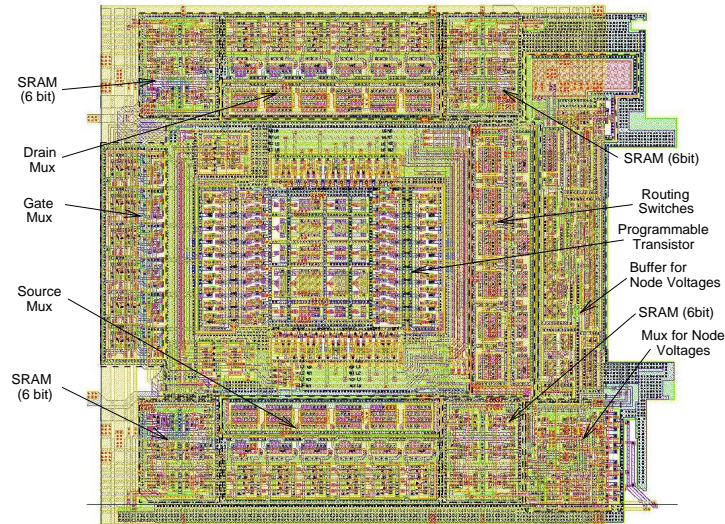


Fig. 6. Layout of one complete NMOS-Cell.

4 Experimental Results

First measurements of the FPTA chip show the full functionality of the transistor cell array: The SRAM can be written to and read out and the programmable transistors behave as expected, which is demonstrated by some transistor characteristics.

4.1 Time needed for the Configuration of the Transistor Array

For the configuration of the complete chip $256 \times 24 = 6144$ bits have to be written. For a write access the 96 bits for one column have to be written to a row of registers in the chip in 16 steps, each time writing 6 bits. Then one complete column is loaded down into the SRAM. In the current implementation of the state machine controlling the RAM access, which is not optimized for

speed, the time for a complete configuration amounts to about 2 ms. From timing measurements however a configuration time of about $70 \mu\text{s}$ with a more optimized FPGA configuration can be inferred. As far as the chip is concerned simulation results suggest that even this time can at least be halved. Compared to the expected evaluation times per individual of 1 to 10 ms, this is almost negligible.

4.2 Transistor Characteristics

In order to measure the output characteristic of some of the PMOS transistor cells the configuration shown in figure 2 has been loaded into the chip. The connected border cell terminals are directly routed to the according array border pads such that the transistor cell can be controlled and measured by an HP 4155A semiconductor parameter analyzer.

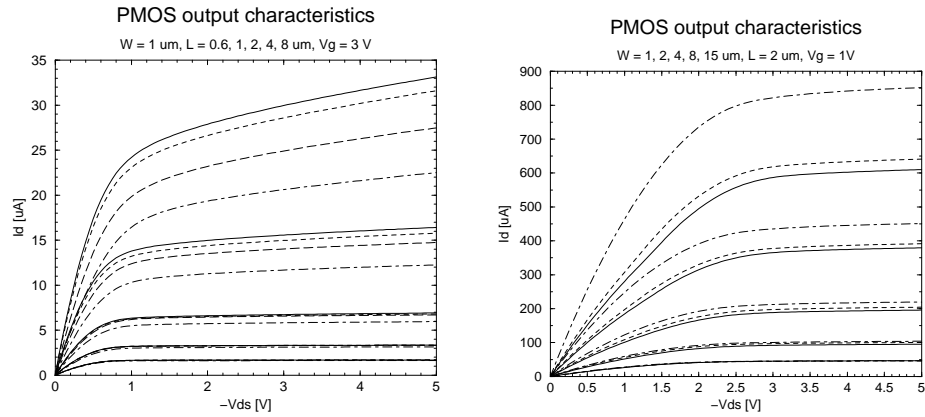


Fig. 7. Output characteristics of programmable PMOS transistors: **Left:** Comparison of PMOS transistors placed at different locations on the chip: Solid: 15/15, dashed: 14/14, long dashed: 9/9 dot-dashed 1/1. **Right:** Comparison of the measured cell 15/15 (solid line), a simulation including all transmission gates (dashed) and one of plain PMOS transistors (dot-dashed).

To compare the output characteristics of different transistors, PMOS transistor cells at different locations on the chip have been measured. For that purpose the terminals of the programmable transistor are always connected to the same pads using the routing capability of the transistor cell array. The results for five different lengths are shown on the left side of figure 7. Apart from looking like transistor output characteristics the curves belonging to the same L value do look similar, but vary in their drain current values. In fact, the output current is the smaller the longer the routing path to the connected border cells. While the relative difference of the saturated drain currents for $L = 0.6 \mu\text{m}$ amounts to approximately 32 %, it decreases to about 4 % for $L = 8 \mu\text{m}$. This is due to the

finite resistance of the transmission gates providing the routing, which explains why the effect is more severe for larger currents (i.e. smaller transistor lengths). In the right half of figure 7 the output characteristic of the PMOS transistor cell in the lower right corner of the chip is compared to the simulation of a plain PMOS transistor as well as to one including the transmission gates used in the measurement. Results are shown for five different transistor widths. While the more precise model of the transistor cell matches the measured curve quite well, the output currents of the transistor cell are always smaller than the ones from the simulation of the plain transistor. Again this is due to the finite resistance of the transmission gates and the discrepancy worsens for higher currents.

4.3 Ring Oscillators

As was already discussed in [6] the bandwidth of any possible circuit in the FPTA is reduced in comparison to the corresponding *direct* implementation in the same process due to the parasitic resistance and capacitance of the transmission gates. In order to get a measure for the maximum frequencies possible in the FPTA the gate delay of an inverter chain has been measured using a ring oscillator consisting of 9 inverters as shown in figure 8. The rightmost inverter buffers the oscillating signal of the circuit, such that it can be measured without changing the oscillator frequency.

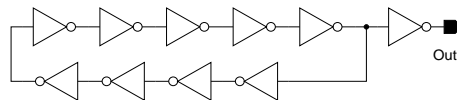


Fig. 8. Implementation of a ring oscillator with 9 inverters.

The circuit was implemented in the FPTA (cf. figure 9) in five different locations, namely all four array corners and the middle of the array. For comparison it was also simulated for different process parameter sets denoting the slowest and fastest as well as the typical behavior of the devices fabricated in the used process.

Used location	upper left	upper right	lower right	lower left	middle	average	119 inverters	lower right slowest W/L
Period	148.5 ns	147.5 ns	150 ns	150.5 ns	148.5 ns	149 ns	1.8 μ s	6.78 μ s
Gate delay	8.25 ns	8.14 ns	8.35 ns	8.36 ns	8.25 ns	8.28 ns	7.56 ns	376.7 ns

Table 1. Measured period and gate delay of the 9 inverter ring oscillator placed in 5 different locations on the chip and of the 119 inverter ring oscillator. The gate delays are calculated by dividing the period by 18 (238 in case of the 119 inverter ring).

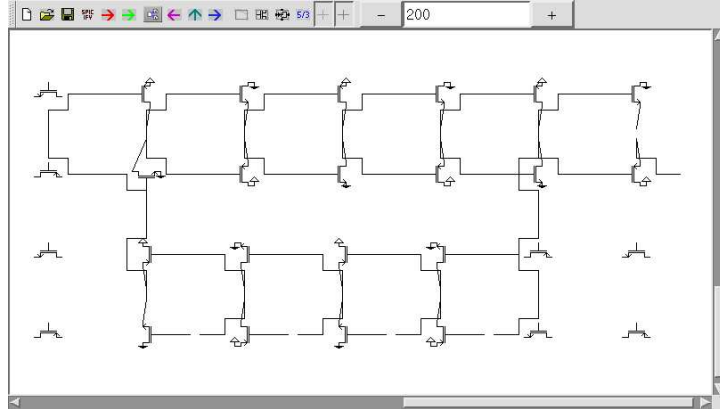


Fig. 9. Implementation of the ring oscillator in the lower right corner of the FPTA.

The aspect ratios used were $14\ \mu\text{m}/0.6\ \mu\text{m}$ and $8\ \mu\text{m}/0.6\ \mu\text{m}$ for the P- and NMOS transistors respectively. Furthermore the oscillator in the lower right corner was measured for an aspect ratio of $2\ \mu\text{m}/8\ \mu\text{m}$ (PMOS) and $1\ \mu\text{m}/8\ \mu\text{m}$ (NMOS) resulting in a lower oscillation frequency. In addition an oscillator containing 119 inverters occupying the complete transistor array was implemented. The results are listed in table 1. A screenshot of the output signal recorded by an oscilloscope is shown in figure 10.

The ring oscillator was simulated using the exact architecture of the FPTA implementation and an implementation using standard cell inverters. Both simulations were carried out with and without the back-annotated parasitic capacitances of the layout and for three sets of process parameters. While *typical mean* (*tm*) denotes the average set of process parameters, *worst case power* (*wp*) and *worst case speed* (*ws*) refer to the parameter sets marking an upper and a lower bound to the speed of the manufactured devices guaranteed by the manufacturer. The results are listed in tables 2 and 3.

Transistor cell simulation	back-annotated simulation			simulation without parasitics		
	tm	wp	ws	tm	wp	ws
Period	219.6 ns	148.2 ns	365.23 ns	84.5 ns	47.4 ns	161.6 ns
Gate delay	12.2 ns	8.23 ns	20.29 ns	4.69 ns	2.64 ns	8.98 ns

Table 2. Simulation results for the ring oscillator with 9 inverters. The left part of the table displays the periods and calculated gate delays for simulations with all parasitic capacitances back-annotated from the layout. On the right hand side the simulation results for the pure schematic (without any parasitic capacitances) are given. The abbreviations *tm*, *wp*, *ws*, refer to different parameter sets for the simulations (further explanations see text).

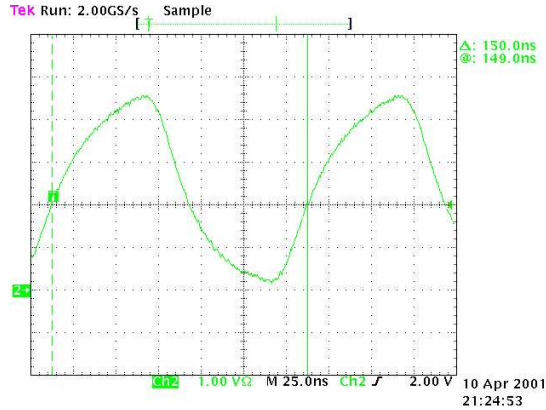


Fig. 10. Screenshot of the output signal of the ring oscillator implemented in the FPTA. One square corresponds to 25 ns and 1 V for the x- and y-axis respectively.

Taking the measured and simulated gate arrays as a measure for the speed of the technology it can be inferred that the loss of speed caused by the overhead for the configurability is about a factor of 100, limiting possible application for the FPTA to frequencies of the order of MHz. Furthermore the fact, that the variation of the observed frequencies is quite small indicates a high level of homogeneity of the array cells. The smaller gate delay extracted from the measurement of the 119 inverters is probably due to the better ratio of the number of cells used as inverter parts to the number of routing cells. Finally the comparison to the implementation with the small aspect ratios shows the range of possible frequency adjustments that can be obtained by simply changing the transistor geometries.

The comparison of measurement and simulation results for the transistor cells yields the following: First, the measured gate delay is significantly smaller than the gate delay extracted from the back-annotated typical mean simulation, although the process parameters accessible from the vendor are closely matching the typical mean parameters. This may be due to the fact that the extraction of the parasitic capacitances yields worst case values. Second, the difference between the gate delay of the simulation without parasitic capacitances and the

Standard cell simulation	back-annotated simulation			simulation without parasitics		
	tm	wp	ws	tm	wp	ws
Period	1.46 ns	942.2 ps	2.47 ns	1.352 ns	853.42 ps	2.34 ns
Gate delay	81.11 ps	52.34 ps	137.3 ps	75.11 ps	47.41 ps	130 ps

Table 3. Simulation results for a ring oscillator with 9 inverters designed out of digital standard cells. As in Table 2 results are shown for the simulation with and without parasitic capacitances.

measured gate delay indicates, that the capacitances introduced by the metal lines are of the same order as the parasitic capacitances introduced by the transmission gates used for connecting the programmed transistors (cf. [6]).

5 Summary and Future Plans

A Field Programmable Transistor Array has been fabricated in a $0.6\ \mu\text{m}$ CMOS process. The chip is embedded in a hardware evolution system designed for the intrinsic evolution of analog electronic circuits. First measurements have proven the chip to work. The time for a configuration of the whole chip is extrapolated to be less than $70\ \mu\text{s}$ allowing for testing rates of up to 1000 individuals per second. Time domain measurements suggest that the chip can be used for frequencies in the order of MHz. The evolution system is almost ready to be programmed for first evolution experiments. The next steps are to optimize the system for high throughput rates and extend it to monitor the die temperature and the current used by the transistor cell array itself.

6 Acknowledgment

This work is supported by the Ministerium für Wissenschaft, Forschung und Kunst, Baden-Württemberg, Stuttgart, Germany.

References

1. M. Loose, K. Meier, J. Schemmel: Self-calibrating logarithmic CMOS image sensor with single chip camera functionality, *IEEE Workshop on CCDs and Advanced Image Sensors, Karuizawa, 1999, R27*
2. J. Schemmel, M. Loose, K. Meier: A 66×66 pixels analog edge detection array with digital readout, In: *Proceedings of the 25th European Solid-state Circuits Conference (ESSCIRC'99)*, B.J. Hosticka, G. Zimmer, H. Grünbacher, Eds., pp 298-301, Edition Frontières, 1999.
3. M. Murakawa, S. Yoshizawa, T. Adachi, S. Suzuki, K. Takasuka, M. Iwata, T. Higuchi: Analogue EHW Chip for Intermediate Frequency Filters, In: Proc. 2nd Int. Conf. on Evolvable Systems: *From biology to hardware (ICES98)*, M. Sipper et al., Eds., pp 134-143, Springer-Verlag, 1998.
4. Thompson, A, Layzell, P.: Evolution of Robustness in an Electronics Design, In: Proc. 3rd Int. Conf. on Evolvable Systems: *From biology to hardware (ICES2000)*, T. Fogarty, J. Miller, A. Thompson and P. Thompson, Eds., pp 218-228, April 17-19, 2000, Edinburgh, UK. New York, USA, Springer Verlag.
5. A. Stoica, R. Zebulum and D. Keymeulen: Mixtrinsic Evolution, In Proceedings of the Third International Conference on Evolvable systems: *From Biology to Hardware ICES2000*, T. Fogarty, J. Miller, A. Thompson and P. Thompson, Eds., pp 208-217, April 17-19, 2000, Edinburgh, UK. New York, USA, Springer Verlag.
6. J. Langeheine, S. Fölling, K. Meier, J. Schemmel: Towards a silicon primordial soup: A fast approach to hardware evolution with a VLSI transistor array, In: Proc. 3rd Int. Conf. on Evolvable Systems: *From Biology to Hardware (ICES2000)*, J. Miller et al., Eds., pp 123-132, Springer-Verlag, 2000.