Intrinsic Evolution of Digital-to-Analog Converters Using a CMOS FPTA Chip

Jörg Langeheine, Karlheinz Meier, Johannes Schemmel, Martin Trefzer
Kirchhoff Institute for Physics, University of Heidelberg,
INF 227, D-69120 Heidelberg, Germany
Email: langehei@kip.uni-heidelberg.de

Abstract

The work presented here tackles the problem of designing a unipolar 6-bit digital-to-analog converter (DAC) with a voltage mode output by hardware evolution. Thereby a Field Programmable Transistor Array (FPTA) is used as the analog substrate for testing the candidate solutions. The FPTA features 256 programmable transistors, whose channel geometry and routing can be configured to form a large variety of transistor level analog circuits. A series of experiments reveals that variations of the output voltage range influence evolution's success more severely than varying the amount of available electronic resources or the geometrical setup. Although a considerable number of runs yield converters with a nonlinearity of less than 1 bit, no DAC is found to maintain a nonlinearity of less than 0.5 bits under worst case conditions, as required for a true 6-bit resolution. While the evolved circuits work comparably well at different time scales as well as on different dice, they lack the ability to abstract from the analog voltage levels of the digital input signals. It is experimentally verified that this can be remedied by inserting digital buffers at the circuits' inputs.

1 Introduction

During the last decades, many signal processing tasks have been shifted from the analog to the digital domain. However, in order to interface electronic systems with the real world, digital signals have to be translated into physical signals, which usually requires a conversion into analog signals. Digital-to-analog converters (DACs) thus have become key elements in many of today's electronic systems. As a matter of fact, they are used in a large variety of applications ranging from CD players to graphic cards, from wireless communication devices to automotive applications. Accordingly, if evolvable hardware is ever to be useful for building up complex electronic systems, it will have to be able to interface to digital signals.

The DACs found by means of hardware evolution reported in the literature so far are restricted to 3 ([1]) and 4 bits ([2], [3]). The former experiments are based on simulations using a generic SPICE 3 model called from a genetic programming algorithm. It took approx. $4.5 \cdot 10^7$ evaluations to find the best-of-run solution, which uses bipolar transistors as well as resistors and capacitors. Since some of these possess values down to $1\,\Omega$ and up to $100\,\mu$F, a direct implementation of the circuit on one piece of silicon would be impractical. The latter work by Zebulum et. al. presents different divide and conquer approaches yielding 3- and 4-bit DACs. While the 4-bit DAC obtained in [2] possesses a current mode output and was tested using SPICE simulations, the circuit proposed in [3] was evolved using the FPTA2 chip described in [4] and produces an output voltage. In addition to facilitating artifical evolution by using a hierarchical approach, a total of four human designed operational amplifiers are included in the circuit.

The work presented in this paper focuses on designing unipolar digital-to-analog converters with a voltage mode output and a target resolution of 6 bits. All evolution runs were allowed to freely explore the used analog substrate – a Field Programmable Transistor Array (FPTA) dedicated to the evolution of transistor level circuits (for details refer to [5]) – without any form of human guidance.

In order to be useful in real world applications a digital-to-analog converter must not rely on the exact voltage levels of its inputs. Hence, a number of experiments are devised to the problem of evolving circuits that are robust against those input voltage variations. Since this task turns out to be too difficult to be solved with the given setup, another series of experiments investigates if this obstacle can be overcome by human intervention, i.e., by inserting digital buffers at each of the six digital inputs.

2 Evolution System

The evolution system, illustrated in Fig. 1, 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 via 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.

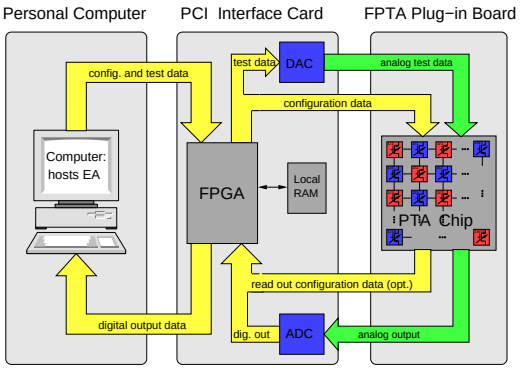

Figure 1. Overview of the evolution system.

2.1 FPTA Chip

The FPTA consists of 16×16 programmable transistor cells. As CMOS transistors come in two flavors, namely N- and PMOS, 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 in Fig. 2.

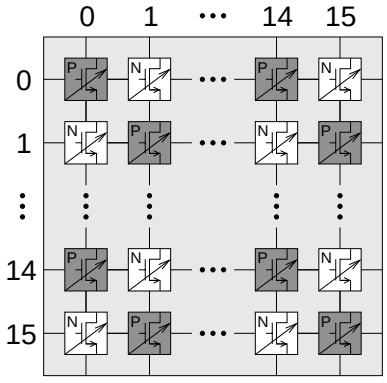

Figure 2. Schematic diagram of the FPTA.

Each cell contains the programmable transistor itself, three decoders that allow to connect the three transistor terminals to one of the four cell boundaries, *vdd* or *gnd*, and six routing switches. A block diagram of the transistor cell is shown in Fig. 3. Width W and Length L of the programmable transistor can be chosen to be $1, 2, \ldots, 15\,\mu\mathrm{m}$ and $0.6, 1, 2, 4, 8\,\mu\mathrm{m}$ respectively. The three terminals *drain*, *gate* and *source* of the programmable transistor can be connected to either of the four cell edges named after the four cardinal points, as well as to *vdd* or *gnd*. The only means of routing signals through the 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 [5].

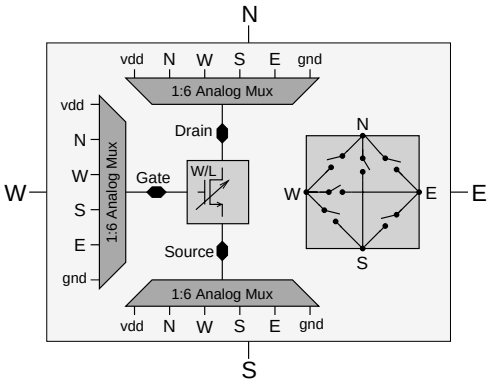

Figure 3. Simplified schematic of one transistor cell.

2.2 Evolutionary Algorithm

Throughout all experiments a simple genetic algorithm was used in conjunction with truncation selection. As can be seen from Table 1, a large fraction of 20 % was directly promoted to the next generation in order to prevent the algorithm from loosing an already good solution due to noise in the measuring process. The individuals taking part in crossover were chosen from the best 60 % and the ones undergoing only mutations from the best 40 % of the current generation respectively. As indicated in Table 1, the mutation rates for changing routing bits, transistor terminal connections and channel geometry can all be set individually. Please note that the mutation rates define the probability of changing the according feature of one cell; they are not scaled with the different array sizes used throughout the paper. Since the crossover block size – the maximum edge length of rectangular blocks that can be exchanged within a crossover operation – is limited to 2, the genetic differences between two successive generations are fairly small and mutation was probably the driving force in the evolution process. For further details of the used GA refer to [6].

GA Parameter	Value
generation size	50
number of generations	10,000
selection scheme	truncation selection
reproduction fraction	0.2
mutation fraction	0.4
mutation rate Term. Connection	3 %
mutation rate Width, Length	2 %
mutation rate Routing	3 %
crossover fraction	0.6
crossover rate	30 %
crossover block size	2

Table 1. GA parameters used.

3 Experimental Setup

3.1 Experiment series

Altogether five series of eight experiments, each featuring 20 runs, were carried out, as summarized in Table 2. For the series FW1, FW4 and FWB4 the task was to map the digital words to analog voltages in an unsigned binary encoding, where the lowest word (all inputs low) corresponds to the lowest and the highest word (all inputs high) to the highest output voltage. In the remaining two series INV1 and INV4 the encoding is inverted, that is, the output voltage should be at its maximum for the lowest input word and vice versa.

Since in initial experiments the output of the evolved DAC circuits was found to strongly depend on the input voltage levels, series FW4 and INV4 were designed to evolve circuits that rely only on the digital information present at the inputs. This is achieved by testing the response of each candidate circuit to all input codes at four different input voltage levels: $0/5, 0.5/4.5, 1/4$ and $1.5/3.5$ V (cf. Table 2). As a result, each DAC is characterized by four curves.

Supposed that there is a correlation between the input voltage level and the output of the DACs evolved in series FW1, it would be interesting to investigate whether this could be avoided by using a reversed encoding scheme: The reverse encoding might bias artificial evolution to use inverters at the inputs, thereby gaining robustness against the input voltage variations. This should be observable by a comparison of series INV4 and FW4. Finally, in the experiments of series FWB4 digital buffers are inserted at the inputs of the circuit under test to restore the analog voltage level of the input signals (cf. Fig. 4). Thereby, the evolution of DACs robust against input voltage level variations should be significantly facilitated.

Series	Input Encoding	Curves	Inp. Voltages [V]
FW1	Forward	1	0; 5
FW4	Forward	4	0...1.5 ; 3.5...5
INV1	Inverse	1	0; 5
INV4	Inverse	4	0...1.5 ; 3.5...5
FWB4	Forw. Buffered	4	0...1.5 ; 3.5...5

Table 2. The five different experiment series.

For each series of experiments, three parameters of the setup are varied as shown in Table 3. First, the desired output voltage range is varied between the intervals 0 to 5 V and 1 to 4 V, where the former one corresponds to the power supply range of the programmable transistor array. Second, two differently sized areas were made available to the GA. The according locations used for inputs and output are depicted in Fig. 4: The upper row contains the geometric setups for all series of experiments except for those of series FWB4, which are depicted in the lower row. The setups for experiments 1,2,5 and 6 are shown in the left column of Fig. 4, whereas those for experiments 3,4,7 and 8 are illustrated on the right hand side of the figure. Assuming the GA uses a resistive network to solve the DAC design problem, the task intuitively appears easier for a setup that places the more significant bits close to the circuit's output, because they are expected to influence it more directly. Accordingly, this *reversed* input order is used for experiments 5 to 8 to test the above hypothesis.

Exp.	Output range	Array Size	Input Order
1	0...5 V	14×14	forward
2	1...4 V	14×14	forward
3	0...5 V	10×10	forward
4	1...4 V	10×10	forward
5	0...5 V	14×14	reverse
6	1...4 V	14×14	reverse
7	0...5 V	10×10	reverse
8	1...4 V	10×10	reverse

Table 3. Experiments for each series.

3.2 Fitness Function

The fitness function used throughout all experiments is simply the sum of squared errors

$$\mathrm{SSE} = \sum_{j=0}^{63} (V_{\mathrm{out}}(j) - V_{\mathrm{tar}}(j))^2 \quad , \tag{1}$$

with regard to the target function

$$V_{\mathrm{tar}}(j) = \begin{cases} V_{\mathrm{low}} + (V_{\mathrm{high}} - V_{\mathrm{low}}) \frac{j}{63} & \text{(2a)} \\ V_{\mathrm{high}} - (V_{\mathrm{high}} - V_{\mathrm{low}}) \frac{j}{63} \quad , & \text{(2b)} \end{cases}$$

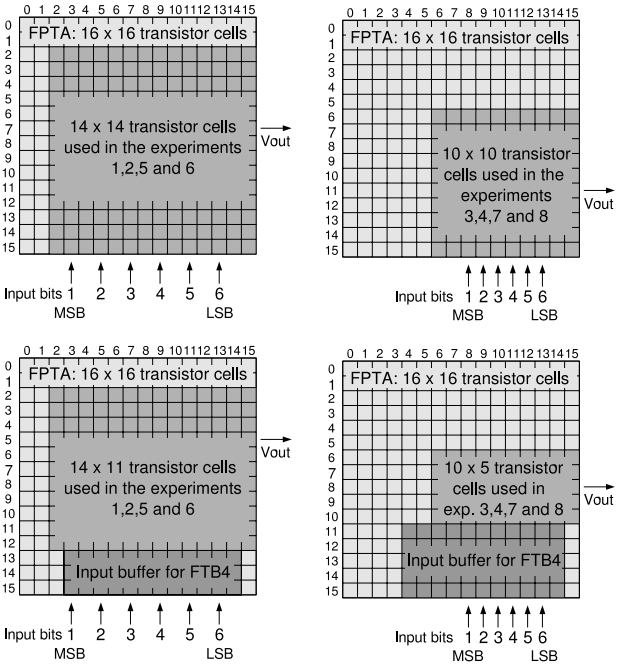

Figure 4. Geometrical setups for the different experiments.

where the integer input code j is calculated from the inputs V_{I_i} by

$$j = \sum_{i=0}^{5} I_i \cdot 2^i \quad \text{with} \quad I_i = \begin{cases} 0 & \text{if} \quad V_{I_i} < 2.5\,\text{V} \\ 1 & \text{if} \quad V_{I_i} > 2.5\,\text{V} \end{cases}, \quad (3)$$

and V_{low} and V_{high} are the boundaries of the output ranges listed in Table 3. While (2a) describes the target function V_{tar} for the series FW1, FW4 and FWB4, (2b) is used for INV1 and INV4. Accordingly, the sum of squared errors has to be minimized by means of the used algorithm. This choice of fitness function aggregates the different objectives high linearity, exact gain and minimal offset, but does not allow to control the weight of their contributions to the total fitness.

3.3 Test Pattern

For series FW1 and INV1 all of the 64 input codes are tested exactly once resulting in one output curve. In case of the other series (FW4, INV4 and FWB4) each input code was tested for all different input voltage levels yielding a total of four output curves. In order to prevent artificial evolution from abusing information from the timing/order of the test pattern, one out of ten different random orders is chosen randomly for each fitness test. In addition, this ensures that varying input code transitions are used for the fitness evaluations in the course of the evolution process. Due to the fact that the FPTA has only one single analog input, the input voltages have to be written sequentially to the chip, where they are stored in sample and hold cells. During evolution the time between the application of two successive input voltages is 167 ns. The output voltage is sampled approximately 1.27 μs after the first input and 0.47 μs after the last input voltage is applied to the transistor array. Thus, the sample frequency with which the different input codes are tested amounts to 750 kHz. These *normal* values of the test pattern timing are summarized in the second column of Table 4.

Time Parameter	Normal	Slow
settling time for last input	0.47 μs	13.4 μs
settling time for first input	1.27 μs	80.6 μs
sample frequency f_S	750 kHz	12.4 kHz
time per run: FW1, INV1	≈ 15 min	-
same for remaining series	≈ 20 min	-

Table 4. Time and Timing considerations for the DAC experiments.

In order to test whether the evolved converter circuits are also working at a different time scale, verification tests were done at the sample rate of 750 kHz used during evolution as well as at 12.4 kHz, where the latter timing is referred to as *slow*. A complete run featuring 10,000 generations and a generation size of 50 took between 15 and 20 minutes depending on the number of different input voltage levels.

4 Results

For each of the 40 experiments 20 evolution runs were carried out. The best genotypes of the last generation of all evolution runs are taken as the result of the experiment. After all runs had been finished, the phenotypical behavior of all these genotypes was verified by measuring the according circuit response 100 times with the same test patterns as used during the evolution process.

Since (1), the sum of squared errors, which is used for the fitness evaluation during evolution, is not an intuitive quality measure, it is converted to the root mean square error per data point in lsb by

$$f = \text{RMSE} = \frac{\sqrt{\frac{\text{SSE}}{N_{\text{IC}}}}}{1\,\text{lsb}} \qquad \text{with}$$

$$N_{\text{IC}} = \begin{cases} 64 & \text{for} \quad \text{FW1, INV1} \\ 256 & \text{for} \quad \text{FW4, INV4, FWB4} \end{cases}, \quad (4)$$

which is used throughout the remainder of this paper. In this context it is worthwhile noting that 1 lsb (least significant

bit) corresponds to 79.4 mV for an output range of 0 to 5 V and 47.6 mV for one of 1 to 4 V, respectively.

4.1 Results for Series FW1

The influence of the eight different experimental setups listed in Table 3 is studied exemplary for series FW1. In Fig. 5 the results of all experiments are plotted as eight histograms. For each run the worst fitness value out of 100

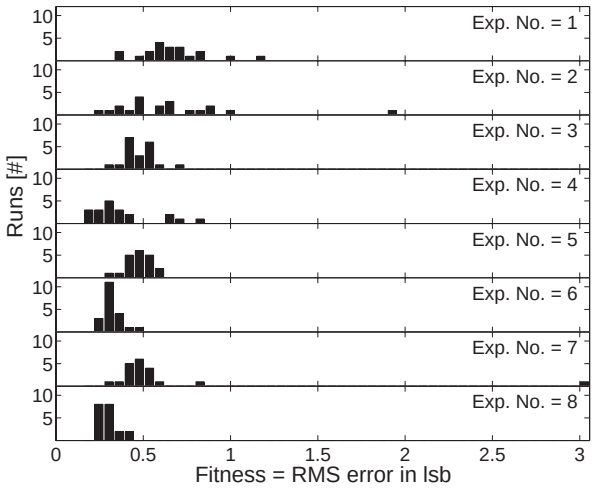

Figure 5. Fitness Histograms for all experiments of series FW1.

verification measurements is used for the plot. Apparently, the runs targeted at an output range of 1 to 4 V performed significantly better than their counterparts required to cover the full power supply range with their outputs. A more detailed analysis reveals that the worse results are mostly due to larger offsets and gain errors and not as much to larger nonlinearities. Neither the geometrical setup nor the size of the transistor array available to the EA influences the evolution results significantly, except for the combinations chosen for experiments 1 and 2: Evolving on the large array of 14×14 cells (see Fig. 4) together with having the less significant bits closer to the output edge yields worse results than all other combinations, independent of the output voltage range.

One of the most important measures to evaluate the quality of digital-to-analog converters are their differential and integral nonlinearity (INL, DNL). They are defined as

$$\mathrm{DNL}(j) = \frac{V_{\mathrm{out}}(j) - V_{\mathrm{out}}(j-1)}{V_{\mathrm{lsb}}} - 1$$

$$\text{for} \quad j = 1, 2, \ldots, 63 \quad \text{and} \tag{5}$$

$$\mathrm{INL}(j) = \frac{V_{\mathrm{out}}(j) - (V_{\mathrm{out}}(0) + V_{\mathrm{lsb}} \cdot j)}{V_{\mathrm{lsb}}}$$

$$\text{for} \quad j = 0, 1, \ldots, 63 \quad \text{, with} \tag{6}$$

$$V_{\mathrm{lsb}} = \frac{V_{\mathrm{out}}(63) - V_{\mathrm{out}}(0)}{63} \quad .$$

Differential as well as integral nonlinearity are plotted in Fig. 6 for the *best* circuit of series FW1, where *best* refers to the lowest RMS error achieved. This circuit was found among the runs of experiment 4. The INL and DNL values

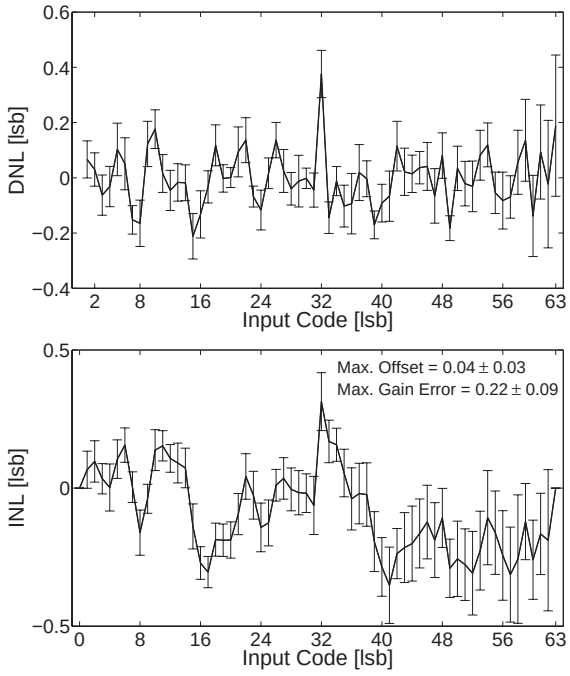

Figure 6. DNL (top) and INL (bottom) for the best evolved DAC of series FW1 (experiment 4).

are averaged over 100 offline tests and the error bars indicate the according standard deviations. As can be seen from Fig. 6, both, INL as well as DNL amount to less than ±0.5 lsb including error bars. It is thus save to say, that the linearity of this DAC, on average, complies with the full target resolution of 6 bits.

However, the histograms of Fig. 7 illustrate that this does not hold for worst case conditions: For each of the 100 verification tests the absolute maximum DNL/INL value is determined. The maximum of the resulting 100 values is taken as the result for one run and appears in the according histogram. The bin size was set to 0.5 lsb for both x-axes. While a considerable amount of evolved DACs manage to achieve maximum nonlinearities of less than 1 lsb for experiments 4,6 and 8, no single circuit was found to have a nonlinearity of less than 0.5 lsb.

Using the definition of the DNL given in (5), it can be deduced that a DAC's output is bound to be monotonic if

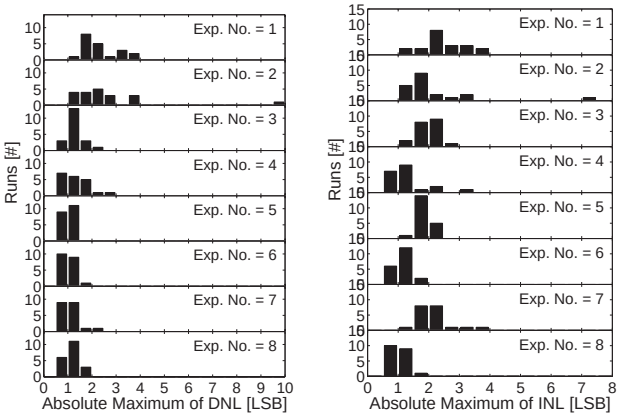

Figure 7. Histograms for the DNL (left) and INL (right) of all experiments of series FW1.

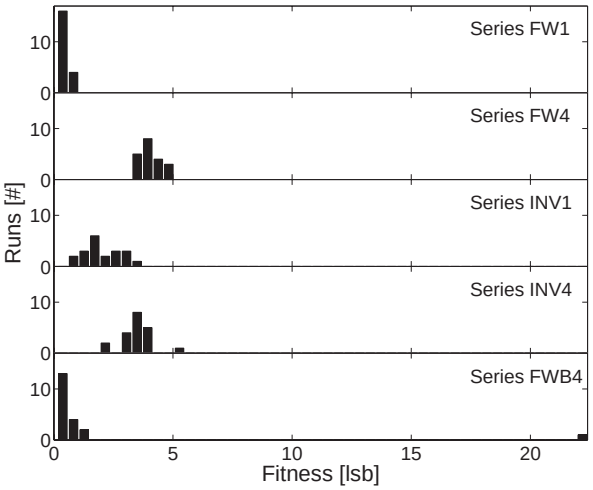

Figure 8. Fitness histograms: Experiment 4, all five series.

$|\mathrm{DNL}| < 1$ is satisfied. Hence, the histograms in Fig. 7 indicate that for experiments 3 to 8 in the order of five to ten evolved DACs possess a monotonic output characteristic.

4.2 Comparison of the Different Series of Experiments

As described in section 4.1 the best results or series FW1 were evolved in experiments 4 and 8 (described in Table 3). Since this also holds for the remaining four series of experiments, the runs performed under the conditions of experiment 4 are used to compare the results of all five different series. Applying (4) to this data yields the histograms shown in Fig. 8, where again the fitness is taken as the maximum value measured in 100 verification tests.

The histograms of Fig. 8 show that it is significantly harder to find digital-to-analog converters that use an inverse encoding as required in series INV1 (cf. section 3.1). The results for the two series FW4 and INV4, in which the output characteristic is tested for four different input voltage levels, are even worse. This indicates that the EA is strongly relying on the analog voltage level of the digital inputs instead of extracting the digital information included. As was expected, the circuits produced in series INV4 behave – on average – slightly better than their counterparts of series FW4. The necessary inversion of the input signals seems to be helpful in abstracting the digital information from the analog input signals. However, as can be inferred from the histogram for series INV1, the algorithm did never choose to place inverters at the inputs, because this would have resulted in circuits with fitness values similar to those of the runs in series FW1. It is worth noting though, that the gain of one stage inverters realizable with the FPTA's transistor cells is not sufficient to restore all four different input voltage levels to exactly 0 and 5 V. Hence, inverting the input signals once would not solve the problem entirely.

Finally, the histogram at the bottom of Fig. 8 proves that the desired robustness against variations of the input voltage levels can be achieved by inserting buffers (two inverters in series) at the inputs of the prospective DAC circuits. Thereby, the total number of used transistor cells was almost preserved, as illustrated in Fig. 4. Thus, the resources available to the EA for implementing the D/A converter are reduced accordingly; in fact, for the setup using the smaller array size, they are actually halved. This and/or the harder timing constraints caused by the additional two gate delays of the input buffers may be responsible for the fact, that the circuits evolved in series FWB4 are slightly less performant than those obtained from series FW1.

To further illuminate the differences between the five different series of experiments, the output characteristics of the best of series DACs are plotted in Fig. 9. From left to right and top to bottom the graphs belong to series FW1, INV1, FW4, INV4 and FWB4. Each plot shows the mean voltage characteristic averaged over 100 consecutive measurements. For series FW1 and INV1 the error bars indicate the according standard deviation; this is omitted for the remaining three graphs for clarity, since they contain four curves each. The graphs contain information about the best, mean and worst fitness value calculated from the 100 verification tests as well as the fitness achieved during evolution. The proximity of these four values proves the underlying circuits to be stable.

While the output characteristic of the best circuit of series FW1 looks almost perfect, the corresponding curve for the best DAC of series INV1 does not form a perfectly straight line. Moreover, both ends resemble the character-

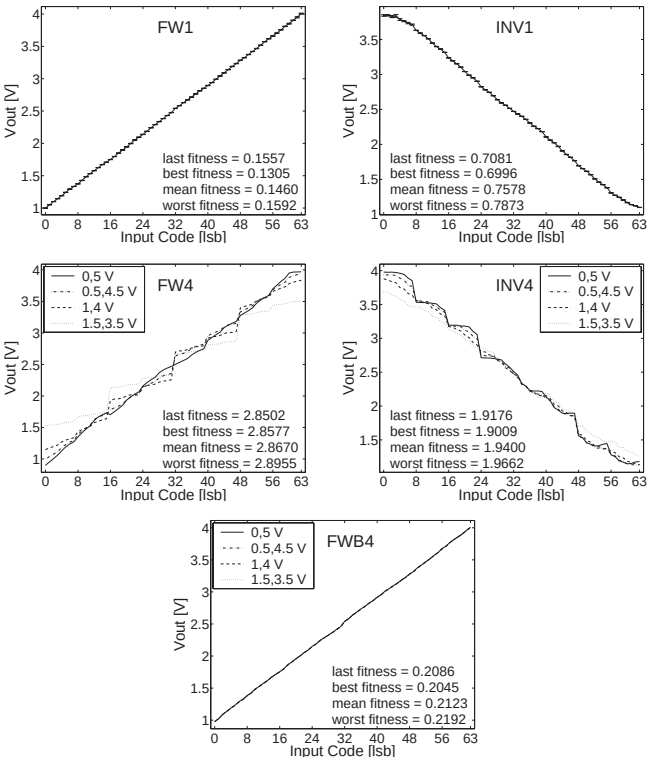

Figure 9. Output characteristics of the best evolved DACs for all five series.

istic curve of an inverter. The situation is worse for series FW4 and INV4: The four characteristic curves exhibit large nonlinearities and differ significantly in offset and gain. Finally, the four curves shown for the best individual found for series FWB4 compare well with that belonging to series FW1 *and* perfectly coincide.

4.3 Verification at a Second Time Scale

As already explained in section 3.3, special precautions were taken to prevent the algorithm from abusing temporal correlations in the test pattern: For each fitness test, the input codes were applied in fixed random orders. Since the exploitation of temporal information was observed in prestudies for other experiments as well as in the work reported in [2], the functionality of the evolved digital-to-analog converters of series FW1 was nevertheless tested on a different time scale. Table 4 sums up the larger settling times and lower sample frequencies of the crosscheck as well as those used for all other verification tests and during evolution; they are referred to as *slow* and *normal* respectively. The settling times differ by a factor of 63 for the first and 28 for the last input.

The fitness values achieved under the two different timing conditions are plotted in Fig. 10: For each experiment

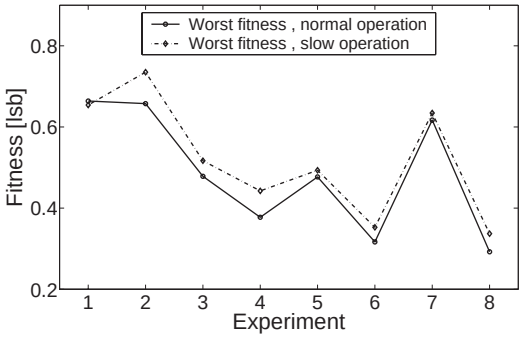

Figure 10. Comparison of the performance of the evolved DACs operated at two different time scales.

of series FW1 the worst fitness values measured in 100 verification tests are used to calculate the mean worst fitness of all 20 runs, once for the sample rate of 750 kHz used during evolution and once for the sampling rate of 12.4 kHz used for crosschecking. Since the resulting curves do not differ significantly (less than a tenth of an lsb) the evolved converters can be said to work well on both time scales and can be expected to do so for the whole frequency range in between.

4.4 Verification on a Second Chip

An important issue in the field of hardware evolution is whether the evolved solutions can be generalized to work under realistic conditions, or if they are bound to the particularities of the very special substrate/model they are evolved on: While simulation based approaches may produce circuits that rely on the special models and parameters of the used simulator, circuits found on one particular die may rely on its exact electrical qualities and fail to work on another die. Therefore the performance of the circuits evolved in series FW1 was tested on a second chip.

The results are plotted in Fig. 11: Again, the worst fitness values obtained in 100 verification tests are used to calculate the mean fitness averaged over all 20 runs belonging to one experiment. On average, the evolved circuits perform slightly worse on the second die; the effect is stronger for the experiments using the smaller output range (2,4,6 and 8). Supposedly, most circuits still work properly on the second die, but their analog performance may be slightly degraded.

The observed discrepancies may be explained in different ways: First, the output of some of the evolved circuits may be strongly deteriorated when measured on the second

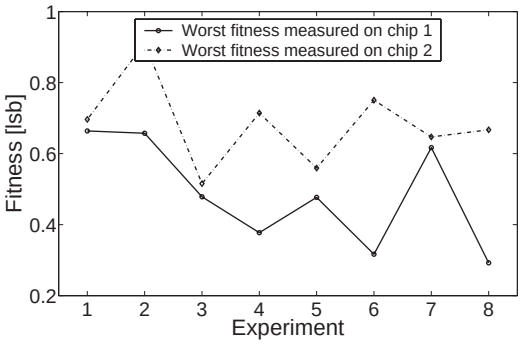

Figure 11. Performance comparison on two different dice for the DACs of series FW1 (chip 1 = chip used for evolution).

chip, while other circuits may exhibit almost the same output characteristic. Second, it is conceivable that the performance of all circuits is evenly degraded, which may be partially due to differences in the analog processing of the in- and output signals of the FPTA and/or to a different power supply voltage, because the second chip was plugged into a different PCI card hosted by a different computer. This hypothesis is also sustained by a more detailed analysis, which reveals that the performance differences are rather caused by deviations of gain and offset than by an increase in nonlinearity.

5 Conclusion and Outlook

The analysis of different series of experiments targeted at finding 6-bit digital-to-analog converters revealed the following: Choosing an output range of 1 to 4 V in conjunction with a suited geometrical setup allows to evolve DAC circuits with an effective resolution of 5 bits. This raises the question whether it is possible to increase the effective resolution by using more sophisticated fitness functions and optimized algorithms.

Further analysis yields that the evolved DACs fail to provide a digital interface, i.e., strongly rely on the analog voltage level of their inputs. It is demonstrated that this flaw can be remedied by inserting buffers at the circuit's inputs. Future experiments will thus provide a pair of reference voltages to the candidate solution, which define the output voltage range. On one hand, this may aid the EA in abstracting from the analog voltage of the input signals, on the other hand it supports the evolution of multiplying DACs.

Moreover, the evolving DACs have not been exposed to a resistive load, which will have to be included to find circuits useful in real world applications. A randomly varied resistive load however, will further constrain the design space to solutions that do not rely on the analog voltage level of the inputs. Since these additional constraints increase the problem difficulty, they may raise the need for more elaborate methodologies as, for example, hierarchical approaches.

The average performance of the evolved circuits gracefully degrades when they are tested on a second chip. In order to get circuits working well on different dice, they could either be fine tuned to the specific electrical properties of the particular die, or be evolved to work equally well on different dice. The latter goal could be achieved by aggregating the fitness values achieved on different dice during the process of artificial evolution, as e.g. done in [7].

6 Acknowledgment

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

References

[1] F. H. Bennett III, J. R. Koza, M. A. Keane, J. Yu, W. Mydlowec, and O. Stiffelman. Evolution by means of genetic programming of analog circuits that perform digital functions. In *Proc. of the Genetic and Evolutionary Computation Conference*, pages 1477–1483, Orlando, Florida, USA, July 1999. Morgan Kaufmann.

[2] R. S. Zebulum, A. Stoica, and D. Keymeulen. Experiments on the evolution of digital to analog converters. In *Proc. of the IEEE Aerospace Conference*, Montana, USA, Mar. 2001. ISBN: 0-78-3-6600-X (Published in CD).

[3] R. S. Zebulum, D. Keymeulen, V. Duong, G. Xin, M. Ferguson, and A. Stoica. Experimental results in evolutionary fault-recovery for field programmable analog devices. In J. Lohn, R. Zebulum, J. Steincamp, D. Keymeulen, and M. I. Stoica, Adrian an Ferguson, editors, *Proc. of the Fifth NASA/DOD Workshop on Evolvable Hardware*, pages 182–186, Chicago, IL, USA, July 2003. IEEE Press. ISBN 0-7695-1977-6.

[4] A. Stoica, R. S. Zebulum, M. I. Ferguson, D. Keymeulen, and V. Duong. Evolving circuits in seconds: Experiments with a stand-alone board-level evolvable system. In *Proc. of the Fourth NASA/DOD Workshop on Evolvable Hardware*, pages 67–74, Alexandria, VA, USA, July 2002. IEEE Press.

[5] J. Langeheine, J. Becker, S. Fölling, K. Meier, and J. Schemmel. Initial studies of a new VLSI field programmable transistor array. In *Proc. 4th Int. Conf. on Evolvable Systems: From Biology to Hardware*, pages 62–73, Tokio, Japan, Oct. 2001. Springer Verlag.

[6] J. Langeheine, K. Meier, and J. Schemmel. 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*, pages 76–85, Alexandria, VA, USA, July 2002. IEEE Press.

[7] A. Thompson. On the automatic design of robust electronics through artificial evolution. In *Proc. 2nd Int. Conf. on Evolvable Systems: From Biology to Hardware*, pages 13–24, Lausanne, Switzerland, Sept. 1998. Springer Verlag.