Taking evolutionary circuit design from experimentation to implementation: some useful techniques and a silicon demonstration

A. Stoica, R.S. Zebulum, X. Guo, D. Keymeulen, M.I. Ferguson and V. Duong

Abstract: Current techniques in evolutionary synthesis of analogue and digital circuits designed at transistor level have focused on achieving the desired functional response, without paying sufficient attention to issues needed for a practical implementation of the resulting solution. No silicon fabrication of circuits with topologies designed by evolution has been done before, leaving open questions on the feasibility of the evolutionary circuit design approach, as well as on how high-performance, robust, or portable such designs could be when implemented in hardware. It is argued that moving from evolutionary 'design-for experimentation' to 'design-for-implementation' requires, beyond inclusion in the fitness function of measures indicative of circuit evaluation factors such as power consumption and robustness to temperature variations, the addition of certain evaluation techniques that are not common in conventional design. Several such techniques that were found to be useful in evolving designs for implementation are presented; some are general, and some are particular to the problem domain of transistor-level logic design, used here as a target application. The example used here is a multifunction NAND/NOR logic gate circuit, for which evolution obtained a creative circuit topology more compact than what has been achieved by multiplexing a NAND and a NOR gate. The circuit was fabricated in a 0.5 μm CMOS technology and silicon tests showed good correspondence with the simulations.

1 Introduction

Conventional circuit design techniques explore only a small fraction of the design space, relying on validated subcircuit topologies, assembling them as building blocks for achieving overall functionality, and optimising their parameters for process portability and increased performance.

Unconventional design techniques, which are still in the 'proof-of-concept' *design-for-experimentation* mode, have recently been attempted for exploring larger design spaces by lowering the granularity of the building-block components. This challenges the very foundation of modern *design reuse* based on building blocks (subcircuits), and often achieves a unique full *design on-demand*, often with topologies never used before. In principle, more optimal solutions may be obtainable in this larger search space; validating the solutions is, however, nontrivial. Breaking the building blocks for which good library models from silicon characterisation are available exposes the designer to risks, reducing his/her capability to accurately predict how the circuits would behave in silicon. Limitations of simulators or simulation models and simplifications to render unconventional design techniques practical, such as testing only on certain operational points, have been the challenges for silicon fabrication of unconventional circuits.

IEE Proceedings online no. 20040503

doi: 10.1049/ip-cdt:20040503

Paper first received 15th May and in revised form 8th December 2003

A. Stoica, R.S. Zebulum, D. Keymeulen, M.I. Ferguson and V. Duong are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA

X. Guo is with Chromatech, Alameda CA 94501, USA

Evolutionary circuit design [1–3], enters this category, allowing the exploration of a larger fraction of the design space compared with conventional tools [2]. The power of evolutionary algorithms to do complete topological synthesis of small circuits has been proven before [3–5]. However, there are great challenges for the scalability of this approach for addressing large circuits. One of the challenges relates to the combinatorial explosion of search space with the increasing number of components (if all connections are allowed), in which case more individuals need to be evaluated. Worse, the time for SPICE simulations (required for every candidate solution) scales badly with the number of nodes in the circuit. Most of the evolved designs reported in the literature have less than 20 components; circuits with a larger number of components were also obtained, yet it is doubtful that a more efficient solution could not be derived if parsimony was strictly enforced, or if pruning was performed at the end of evolution. Designs that are more complex could still be addressed if one increases the complexity of the components. Other solutions to scalability are also under investigation [6].

However, in the context of this paper, the focus is on another potential limitation of evolutionary circuit design is of interest, which is true for all unconventional design techniques, i.e. how can one evolve solutions with a high confidence that these would actually work in silicon as predicted by simulations. The design exploration work focused on proof-of-concept, functionality and creativity, and not on design-for-implementation. Resulting circuits were never fabricated and tested; hence there is no information about their performance in silicon. The set of evolved circuits that are solutions to satisfying a function subsumes a smaller set which is also efficient and works under implementation constraints. Evolved circuits were notorious for being hard to explain by humans; it may be

possible that the closer these are for implementation/fabrication, the easier would it be to explain and predict their behaviour. Lack of prediction outside the test points makes these designs risky to accept. One of the outcomes of these explorations of nonconventional techniques may be an extension of the human-designed class of circuits currently in use, with new machine-designed circuits, which would be used in the future as common building blocks for larger circuits.

No silicon implementation of a circuit with topology designed by evolution has been reported to date (we refer here to an ASIC solution rather than evolved solution mapped on a programmable device as in our FPTA experiments [7], or a configuration/routing solution using conventional FPGA cells [8]). Perhaps the lesser performance of unconventional evolved designs compared to conventional designs, or perhaps the lack of trust in their performance after fabrication deterred researchers from risking fabrication costs.

Indeed, we found that applying only tests conventionally used for testing classic circuits was insufficient when targeting design-for-implementation. Thus, the objective of the paper is twofold: to present techniques that were useful in evolving circuits that are sufficiently robust and suited for VLSI implementation, and to present silicon results, providing for the first time a silicon 'proof' that evolved circuits can be made to operate in silicon at predictable performance.

The problem domain is the design of multifunction logic circuits, with a specific example of seeking new, unconventional, topologies for a configurable NAND/NOR logic element. Such NAND/NOR designs are used for example in programmable logic cells [9], or in processor design in maximising the flexibility of using spare gates for correcting design errors through changes in the interconnection of logic elements [10].

2 Methods used in evolutionary design-for-implementation

Evolutionary design has been mainly design for experimentation. Our observation by analysing over time evolved designs that we and other researchers obtained was that evolved designs do not normally work outside the specific model and tight conditions in which they were evolved. This may also indicate that factors about the fabrication that may not have been included in simulations may prevent the circuit from working when actually fabricated into silicon. As an example, such factors may not have been included in order to accelerate evolution by simulating with a simple model; however, there is also the possibility that some information on the process may be lacking, and this may be something that evolved circuits may prove sensitive to.

Testing is usually done for a narrow domain since testing large domains is at least impractical, if not impossible. The solution may work for the targeted process used in simulations and fail on another. Obviously if a targeted solution was sought and good models available, a high-performance solution may not be directly portable to another fabrication process. In this paper we are interested in deriving circuit solutions that are relatively general, such as, for example, the classic transistor-level design for a NAND gate, which would work robustely when fabricated in various processes. This 'portability' problem is similar to that first noted by Thompson in early evolutionary experiments [11], where the solution evolved in one FPGA (with the hardware in the loop) failed to work in the same way when tried in another similar FPGA, or even in different region of the same FPGA. The cause is related to differences of the chip characteristics that evolution exploited in one FPGA chip/region and could not exploit in the other. Particularly, for evolutions with the chip in the loop, evolution can explore subtle properties of the silicon, and parasitic effects, which vary even between chips from the same fabrication lot.

The same has been noted when evolving in simulations and then attempting to map the result to programmable hardware. This worked in some situations but it did not work in others. The reverse was also observed: circuits evolved on a programmable device did not always work the same way in a simulation of the topology (see [7] for more details). Horrocks [12] was the first to address the issues of evolving under many process variations, with the fitness measure being an average of individual fitness measures, thus rewarding circuits robust to process variations. Thompson [13] also used the same idea in evolving with individuals evaluated on several chips (and also at different temperatures) to provide robustness to these variations.

The solution in [7] was named mixtrinsic evolution, evaluating candidate solutions both in software (SW), also referred to as extrinsic, and in hardware (HW), also referred to as intrinsic. The evaluation can be done for both situations as described in [12] (here each individual both in SW and in HW), and receiving a combined measure of individual fitness measures. Another way, proposed in [7], is to alternate evaluation in SW and in HW over generations. Thus an individual will be assigned to be evaluated in SW or in HW with a certain probability; over generations the individuals will be alternatively assigned to SW or HW; solutions that are poor in any of the two would be eliminated. The solution in [7] works only for reconfigurable devices (since requires HW evaluation), and was extended in [14] to include mixtures of software-only models, such as models of different resolution of various fabrication processes.

Comprehensive testing is needed to ensure that evolved solutions cover the intended operational space; no assumptions on their performance outside the points tested during evolution can be reliably made. The methods to be described proved useful in obtaining circuits that satisfied the requirements and functioned as predicted in silicon. Some are general, others are specific to the particular target domain of transistor-level design of logic circuits.

2.1 Ensuring correct transient response for all input combinations

Candidate logic circuits should be tested in transient analysis for all possible transitions of sequence of input levels, as opposed to all possible levels in only one order. For example, a circuit may respond well as an AND-gate to input sequences of levels 0–0, 0–1, 1–0, 1–1. However, it may turn out to have a too long discharge time when tested with the sequence of inputs 1–1 following 0–0 and not 1–0 as above, which is not tested in the simple scheme. The price to pay for testing all sequences is an increased duration of the transient analysis. Thus, even for four different input combinations for the operating point analysis in a two-input gate, transient analysis used seven input configuration cases (for the gates studied here) to include all the sequences. For example, the OR-gate published in [15] which was tested during evolution only for input combinations 0–0, 0–1, 1–0, 1–1, provided inadequate response when later tested for the complete set of sequences of input transitions. Figure 1 shows the circuit response for the two cases. It can be seen that, during evolution, the circuit was only tested for

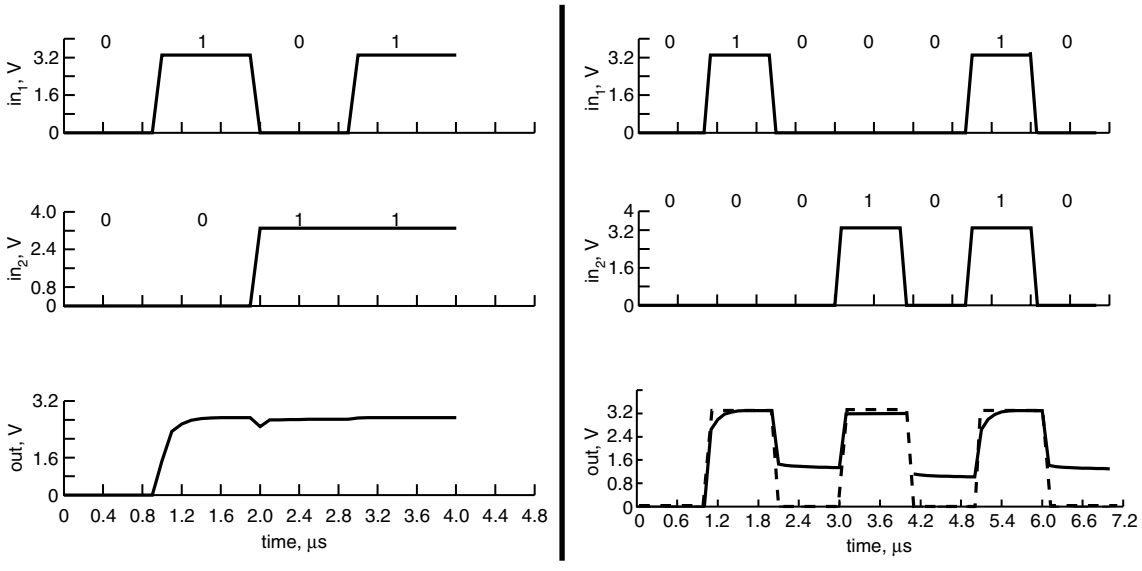

Fig. 1 *OR-gate evolved for correct logic levels but only for selected transitions in order shown left, performing well for those transitions. Response is suboptimal when tested against all possible input transitions, as seen right*

– – target response
—— circuit response

one positive transition when the inputs change from '00' to '10' (left of the Figure). When the circuit is tested for the complete set of input transitions (right of the Figure), the negative output transitions, not tested during evolution, are very slow.

2.2 Ensuring driving capability by loading with identical circuit

Circuits should be tested on various loads including loads consisting of copies of itself (identical circuits) to guarantee that it will be able to drive similar gates. This is a way to ensure that the circuit would drive others like it, and that it can be driven by others (both input and output impedance aspects are addressed this way). Driving a fixed load may not be optimal since one may not know anything in advance about input or output impedance of the circuit to be designed, unless it is a design requirement. This avoids problems observed in our earlier experiments in which evolved circuits were not able to drive similar circuits.

2.3 Ensuring driving capability (cascadability) by using domain knowledge

Domain knowledge can be used to speed-up evolution of circuits with good loading capability. For example, this can be done by constraining evolution into using only the transistor gate terminal to connect circuit inputs (preventing input connections to transistor source or drain), thereby forcing high the input impedance of the evolving circuit gate. In our experiments this shortened the time for evolving cascaded circuits by almost an order of magnitude.

Figure 2 shows an example of an AND-gate evolved in an earlier experiment in which no precaution was taken to ensure drive capability. The circuit schematic shows that the inputs are connected to the drain/source terminals of the transistors. As a consequence we observe a degradation of the output DC level when the gate is tested in a cascaded configuration as opposed to driving a fixed capacitor.

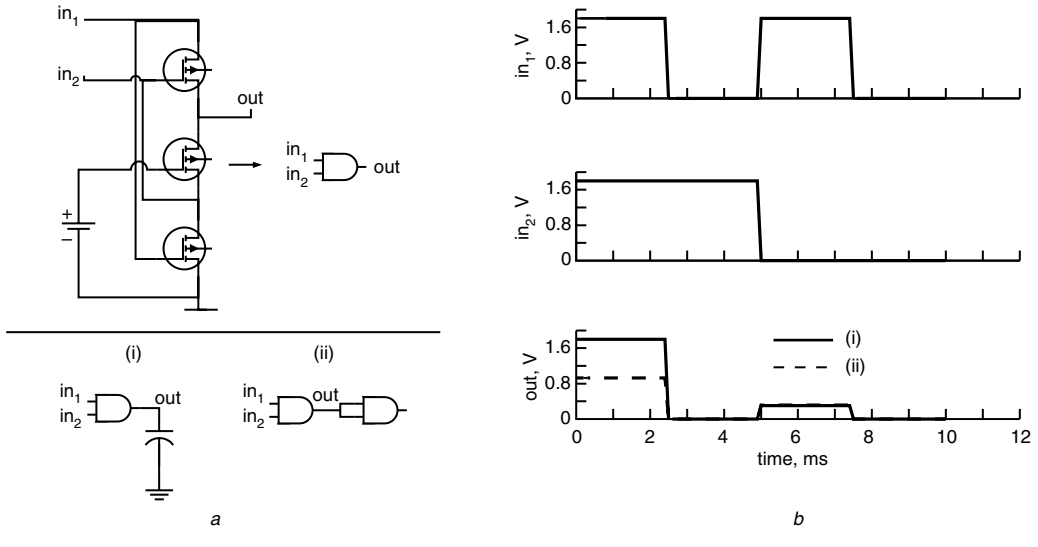

Fig. 2 *Earlier evolved AND-gate without constraining input application points*

a Circuit schematic is shown (top) together with two testing configurations (bottom), with fixed capacitor (i) and loading a copy of itself (ii)
b Inputs In1 and In2 and output shown in time domain

2.4 Guaranteeing operation at different timescales

Testing at several timescales should be used instead of testing only at one timescale. The implicit assumption of human designers that a circuit should function at various frequencies may be missing from explicitly formulated requirements and thus from the fitness function of an evolutionary design. An example is the implicit assumption that a logic gate should have the same behaviour over a 'frequency range' i.e. function with slow/DC signals as well as faster input signals. Simplistic testing would use an input stimulus with a SPICE transient analysis with changes in the microsecond range, and correct behaviour for this timescale would be quickly achieved by evolution. However, this circuit may have a totally different behaviour at a different timescale. Circuits required to be fast may not work on DC levels (we evolved several of these circuits which work if the inputs switch faster than charge is eliminated). An example of an evolved NAND-gate presenting this behaviour is shown in Fig. 3. Similarly, circuits evaluated at a slow-timescale evolution will result in slow gates: so evaluations at both domains are needed. This is an example of mixtrinsic evolution, in which same circuit is evaluated not on two or more models, but with two or more analysis types. This approach may lead to extensive simulation time for evaluations. One way to address this is to extend the transient analysis duration to avoid transient solutions (with wrong behaviour at large timescales) while keeping the transient analysis step small enough to assess the gate speed.

2.5 Speeding-up evolution

Accelerated evolution benefits from mixtrinsic evolution [7], with biasing of the population to more individuals evaluated with a simplified (faster to simulate) SPICE model. For example, we used a level-three transistor model for the HP 0.5 μm process to simulate faster than the BSIM model given by the manufacturer. At one extreme, the population consists only of circuits evaluated with the simplified model, however, in such cases we simulated again all solution circuits using the complete BSIM models and their design corners (using both slow and fast versions of the HP model). In our experiments the simplified transistor model delivered sufficiently accurate results in the case of logic gates when compared with the silicon measurements. This will certainly prove insufficient for designs pushing the limits of performance (e.g. very high frequencies).

2.6 Robustness to parameter variation

Mixtrinsic evolution can also be used to speed-up evolution for robustness to changes in temperature and power supply (V_{dd}). Again, we have the choice of skewing the distribution of population in mixtrinsic search, from populations in which all individuals go through full testing (at the cost of an increase in the evolution time), to populations in which few or none go to all testing and most or all go to simplified testing and only the final evolved circuits are tested to all design corners. In our experiments most (but not all) of the solution circuits achieved through partial testing worked for $\pm 10\%$ variations of V_{dd} and a wide range of temperatures (-20 to 200°C). In this case it was convenient to evolve the circuits for nominal conditions and test the final solution for the design corners, but again this may not be necessarily the fastest way.

3 Evolved circuits and silicon validation results

Several evolutionary designs were performed using the techniques. The example used here for illustration is a multifunction NAND/NOR logic gate circuit, for which evolution obtained a creative novel topology. Multifunction

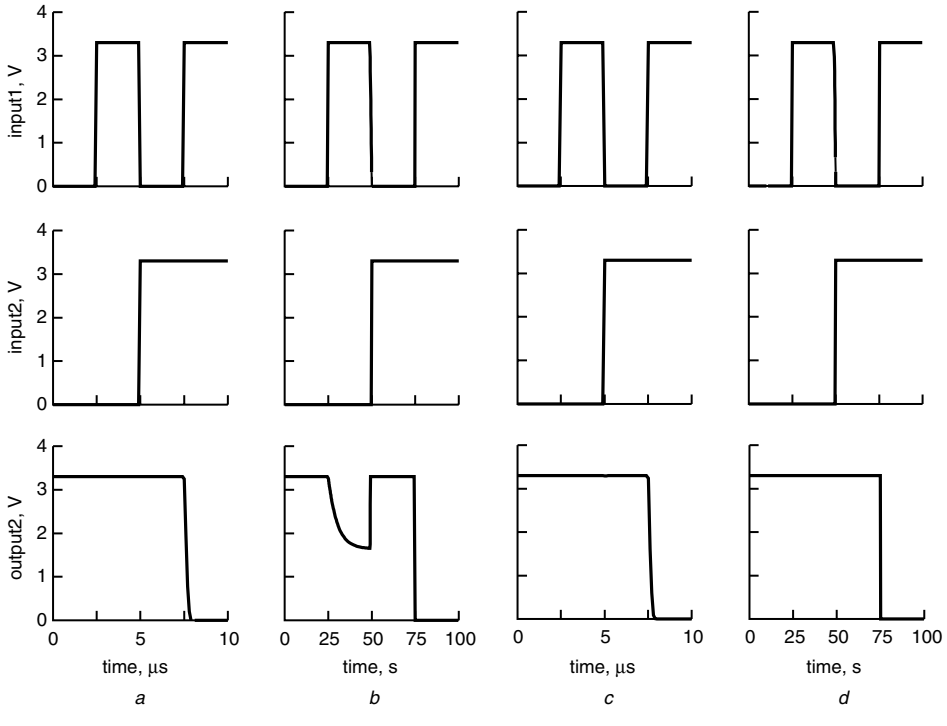

Fig. 3 *Response of evolved NAND-gate*

a Timescale in microseconds
b Incorrect behaviour when simulated in timescale of seconds
c Correct behaviour, timescale in microseconds
d Correct behaviour, timescale in seconds

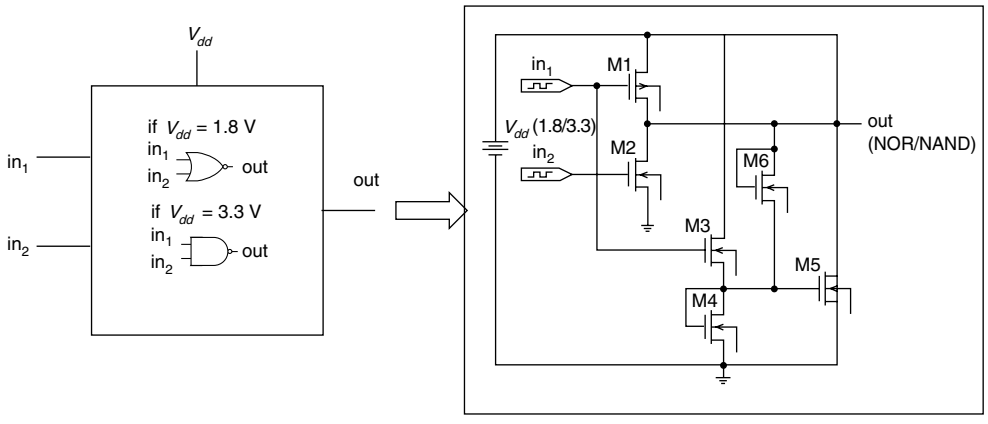

Fig. 4 *Schematic of evolved NAND/NOR circuit*

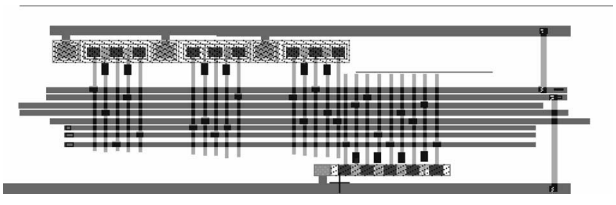

Fig. 5 *Layout of multifunctional logic gate manufactured using HP 0.5 μm technology*

NAND/NOR gates are useful in programmable logic cells or in processor design to maximise the flexibility of using spare gates for correcting design errors through changes in the interconnection of logic elements. A particular innovative characteristic of this design is that the function change is controlled by changes in V_{dd} level. No conventional design was available for such a requirement.

The following techniques introduced in the previous Section were used during the evolutionary experiment.

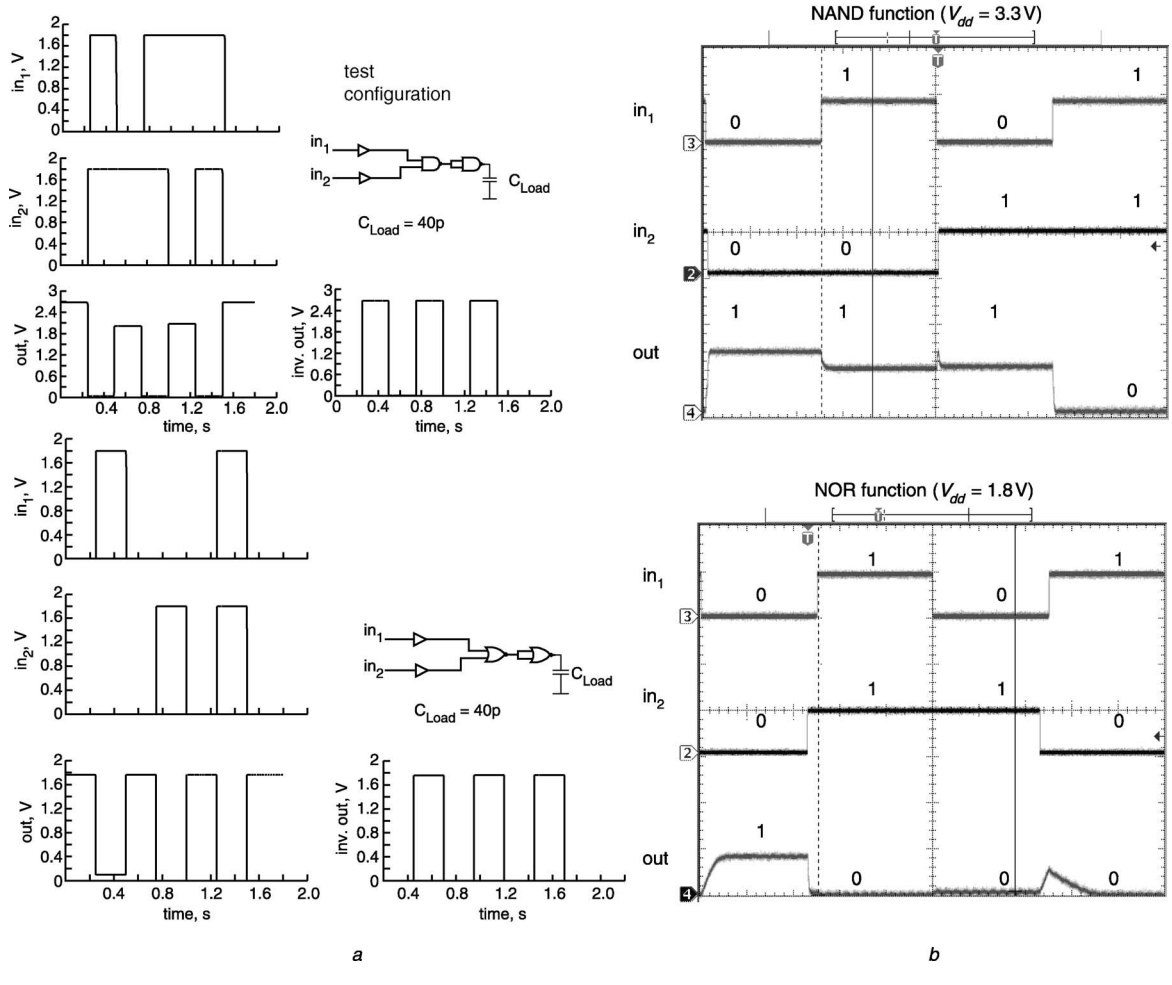

Fig. 6 *Results of transient analysis for NAND/NOR logic gate*

Gate is tested in cascaded configuration in simulation and two outputs are shown (output of gate used as load is complement of output of first gate)
a Simulation results
b Silicon results

• Technique 1 (Section 2.1). The NAND and NOR functions were evaluated for all the possible input transitions causing output changes (shown in the left of Fig. 6).
• Technique 3 (Section 2.3). Circuit inputs were applied only to transistor gate terminals, ensuring that the gate can be cascaded.
• Technique 4 (Section 2.4). The transient analysis duration time was extended to 1.5 s, large enough to avoid transient solutions (with incorrect behaviour at large timescales). The transient analysis timestep was kept at 1 ms and the capacitive load was 40 pF. Although the circuit was evolved for 1 kHz, the final solution was able to work at 1 MHz for a 1 pF load (equivalent to a fan-out of about four).

Technique 2, using an identical gate as load, was used after evolution to confirm that the solution circuit can be cascaded. This technique was used only for testing the final circuit because it is more time consuming than technique 3. Techniques 5 and 6 were not used in this example, but the robustness to V_{dd} and temperature variations was tested after evolution. In this particular case, all circuits in the population were simulated with a simplified set of HP 0.5 µm technology parameters. The final solution was correctly tested using the complete HP 0.5 BSIM model. In addition, the circuit design solution worked for other technologies (TSL 0.5 µm).

In this experiment the circuits in the population were only evaluated for nominal values of V_{dd} and temperature. The solution circuit was successfully tested for $\pm 10\%$ variations of V_{dd} and for temperatures -20 and 200°C.

An integer representation was used in which the chromosomes consisted of a collection genes, each of which encoding a particular component (MOS transistor). The gene determines the type (n or p), connecting points and size (width and length) of the MOS transistor. This representation was unconstrained since each transistor terminal could be connected to any circuit node during the chromosome decoding process. This was a variable-length representation where the chromosomes were allowed to grow in size. The circuits encoded by the initial population had a small number of components, typically three or four and could grow up to 30 components along the evolutionary process. The growth operator worked together with the chromosome correction operation: whenever there was a floating node found during the decoding process of one individual, a new component was added connected to this node thereby correcting the circuit. The maximum number of circuit nodes was initially assigned to the initial number of circuit components; the value of this parameter was increased as the circuit sizes increased during the evolutionary process. Most experiments used populations of 40 individuals and a number of 400 generations.

The evolved circuit is shown in Fig. 4. The circuits were fabricated on a prototype ASIC on a HP 0.5 µm process. Figure 5 depicts the circuit layout. Figure 6 shows waveforms, which prove the correct functionally in simulation and in silicon of the evolved design. For clarity, in the silicon tests, we show the gate response only for the four possible input configurations in one sequence. Further tests were performed in silicon, applying all possible input sequences (as shown in simulation in Fig. 6), with the results agreeing very well with the simulations.

4 Conclusions

This paper presented some techniques that may be useful in moving from evolutionary 'design-for experimentation' to evolutionary 'design-for-implementation'. Some techniques address problems that are also found in conventional design. Others address problems that are particular to evolutionary design. The techniques were used in evolving a novel multifunction NAND/NOR logic gate circuit, which was successfully tested in a prototype chip, this being the first silicon validation of an evolved circuit (ASIC solution).

5 Acknowledgments

The research described was performed at the Jet Propulsion Laboratory, California Institute of Technology, and sponsored by the Defense Advanced Research Projects Agency and the National Aeronautics and Space Administration.

6 References

1 Stoica, A., Keymeulen, D., Zebulum, R.S., Thakoor, A., Daud, T., Klimech, G., Jin, Y., Tawel, R., and Duong, V.: 'Evolution of Analog Circuits on field Programmable Transistor Arrays'. Proc. of 2nd NASA/DoD Workshop on Evolvable Hardware (EH2000), Palo Alto, CA, 13–15 July 2000, pp. 99-108
2 Thompson, A., Layzell, P., and Zebulum, R.S.: 'Explorations in design space: unconventional electronics design through artificial evolution', *IEEE Trans. Evol. Comput.*, 1999, pp. 167–196
3 Koza, J.R., Keane, M.A., and Streeter, M.J.: 'Evolving inventions', *Sci. Am.*, 2003, pp. 52–59
4 Zebulum, R.S., Pacheco, M., and Vellasco, M.: 'Artificial evolution of active filters'. Proc. 1st NASA/DoD Workshop on Evolvable Hardware (EH), Pasadena, 19–21 June 1999, pp. 66-75
5 Lohn, J.D., and Colombano, S.P.: 'Automated analog circuit synthesis using a linear representation', *Lecture Notes Comput. Sci.*, 1998, **1478**, pp. 125–133
6 Stoica, A., Zebulum, R.S., Keymeulen, D., Ferguson, M.I., and Guo, X.: 'Scalability issues in evolutionary synthesis of electronic circuits: lessons learned and challenges ahead'. Proc. AAAI Workshop on Computational Synthesis, Stanford University, CA, 24–27 March 2003, pp. 233–238
7 Stoica, A., Zebulum, R.S., and Keymeulen, D.: 'Mixtrinsic evolution'. Proc. Int. Conf. on Evolvable Systems (ICES), Edinburgh, UK, 17–19 April 2000, pp. 208–217
8 Levi, D., and Guccione, S.: 'GeneticFPGA: Evolving stable circuits on mainstream FPGA devices'. Proc. 1st NASA/DoD Workshop on Evolvable Hardware, Computer Society, July 1999, pp. 12–17
9 Bertolet, A.R., and Clinton, K.P.N., Fuller, C.M., Gould, S.W., Hartman, S.P., Iadanza, J.A., Keyser, F.R., Millham, E.E., Reny, T.S., Worth, B.A., Yasar, G., and Zittritsch, T.J.: 'Programmable logic cell'. United States Patent 5,748,009, May 1998
10 McDermott, M.W., and Turner, J.E.: 'Configurable NAND/NOR element'. United States Patent 5,592,107, January 1997
11 Thompson, A.: 'Silicon evolution'. Proc. Conf. on Genetic Programming (GP), Stanford University, 29–31 July 1996, pp. 444–452
12 Horrocks, D.H., and Khalifa, Y.M.A.: 'Genetic Algorithm Design of Electronic Analogue Circuits Including Parasitic Effects'. Proc. 1st Online Workshop on Soft Computing, Nagoya University, 19–30 August 1996, vol. 1, pp. 274–278
13 Thompson, A.: 'On the automatic design of robust electronics through artificial evolution', *Lecture Notes Comput. Sci.*, 1998, **1478**, pp. 13–24
14 Guo, X., Stoica, A., Zebulum, R.S., and Keymeulen, D.: 'Development of consistent equivalent models by mixed-mode search'. Proc. IASTED Int. Conf. on Modelling and Simulation (MS), Palm Springs, CA, 24–26 February 2003, http://www.iasted.org/conferences/2003/palm/ms.htm
15 Stoica, A., Zebulum, R., and Keymeulen, D.: 'Polymorphic electronics'. Proc. Int. Conf. on Evolvable Systems, Tokyo, Japan, October 2001, pp. 291–302