Proceedings of the 2007 IEEE Workshop on
Evolvable and Adaptive Hardware (WEAH 2007)

On the “Evolvable Hardware” Approach to Electronic Design Invention

Varun Aggarwal, Karl Berggren and Una-May O’Reilly
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, USA

Abstract

Our interest is in evolutionary electronics and its exploitation for invention. In this paper, we survey how the community currently tries to exploit the invention potential of evolvable hardware. We assess whether the current state-of-art in evolvable hardware has resulted in successful inventions or invention tools. We discuss the challenges faced by the community in this regard and how the community can overcome these challenges by exploiting nascent technologies, which include new design substrates, new computational paradigms and recently discovered design principles.

1 Introduction

The term “Evolvable hardware” broadly refers to the set of diverse activities at the intersection of evolutionary algorithms and hardware design. Our particular interest is in evolutionary electronics: evolutionary algorithms applied to circuit design. Since its inception, evolvable hardware for circuitry has offered a means of:

1. single instance tuning of a circuit for correcting fabrication drifts.
2. evolving circuits that are self-repairing and fault tolerant
3. evolving circuits that *must* self-adapt and tune
4. *inventing* novel circuit designs that:
   (a) take advantage of substrate computation below the human level of abstraction, or
   (b) exhibit robustness to device failures in ways that humans would not design, or
   (c) evolved within the realm of human designed device models.

In this submission we would like to focus on evaluating how well EHW has lived up to its promise as a means of inventing novel circuit designs. We ask whether our community’s collective progress toward fulfilling this potential is steering us in promising directions which we should continue to follow, and, whether there are other means of tapping the inventive potential of EHW.

2 Assessment Criteria

As with any assessment it is best to first clearly lay out our success criteria. Our standard shall be that an evolutionary circuit invention system is successful if its invented designs are respected and approved by the circuits community. This implies that the circuits community plausibly could use the invented designs in either an academic or industrial context. It might employ the algorithms that generate inventive designs as general purpose tools. And, it will accept for publication, evolved designs for the sake of their functionality and performance rather than the novelty of the technique that derived them. Collectively, the evolutionary invention system must engender trust in its methods and acceptance of its designs by the circuits community.

There have been examples where evolutionary hardware systems outside electronics have been able to successfully invent unpredictable and complex designs which have been of use to the community of application. One such compelling example is the NASA X-band antenna [19] that may actually fly on a spacecraft. The antenna design is flight worthy and entirely derived by an evolutionary algorithm. It complies with its performance requirements and sports an unusual organic-looking structure which seems unlikely to have been designed by hand. What about evolvable electronics?

3 Evolutionary Circuit Invention Today

We can address our question by considering each of the three means, a) to c), being pursued:

1-4244-0699-4/07/$20.00 ©2007 IEEE

a) Inventing Circuits that exploit substrate behavior ignored by humans The EHW research community has tried to exploit the inventive power of the EHW paradigm by delving below the human abstraction of the lowest level computational building block. The human-imposed abstraction of the perfect digital gate, lumped-resistor/capacitor assumption or transistor models exists because humans find it difficult to cope with the actual complexity that emerges from the substrate. In effect, the abstraction bundles up and hides a certain amount of what is going on and even assumes that it behaves not quite exactly as it truly does. The abstraction, though necessary to enable humans to design, comes at the price of efficiency. The unique aspects of individual behavior of a component are considered as non-idealities, rather than exploited, and complex behavior in potentially efficient power regimes is sacrificed for simplicity. The promise of working at the substrate level is that circuits which are not composed of abstracted elements could be more efficient in terms of use of chip power or area resources (or both) yet display the same functionality and performance. Efficiency would arise either from the liberation from design rules that work within the constraints imposed by abstraction or the *use* of biases (non-idealities for circuit designer) of the substrate in the evolutionary design process.

Adrian Thompson's work [31] on evolving circuits below the gate level (using an FPGA as hardware) was the first invention exploration in this vein. Thompson used a frequency discriminator as a case-study for much of his experimental invention [31]. Though evolution could come up with compact solutions for the problem on a digital substrate by exploiting its inherent analog nature, its circuits were not robust. A circuit would not work appropriately when implemented on other parts of the FPGA or a different FPGA. This indicated the large extent to which the circuit was intertwined with the physics of the chip section that it was evolved on. This might have been acceptable had the circuit at least been robust to other variations such as voltage, temperature and time-invariance. However, this was not the case either. In later experiments, Thompson addressed robustness by allocating fitness with tests on multiple substrates and temperatures [32]. The resulting evolved circuit though not constrained to respect the digital abstraction, succumbed to it! The evolved robust circuit worked in simulation using gate level models. Thompson's series of experiments seem to indicate that evolvable hardware, in fact, may not gain by exploiting substrate behavior when robustness is required. To yield even a modicum of robustness (the designed circuit had glitches and was still not industrial strength), it must resort to employing higher level abstractions just like humans do.

There were other attempts in the more general vein of evolving an unconventional design through freedom from conventional abstraction. Gwaltney and Ferguson [10] tried to generate controllers with a smaller space budget than a conventional PID controller. Again, lack of robustness and efficiency were roadblocks to the controllers meeting industrial functionality. Miller [23] studied the evolution of digital filters using only simple logic gates such as AND, OR, XOR, etc. Conventional filter design is far more complex and elaborate. It uses many floating-point multipliers and summers in the design. Miller's results showed success by evolution of filters in a much smaller resource budget than the conventional but the filters could not handle the challenge of behaving linearly, which is in general a design requirement.

It is probably still premature to completely write off exploiting substrate properties as a means of evolving inventive circuits. There are scenarios in which the apparent shortcomings are not deal breakers. Consider the need for industry strength circuits on an extremely small area and power budget. In a production setting, the circuit for a functionality could be evolved separately for each chip. Or, we could study the design trends in our evolved unconventional circuits and observe from them heretofore ignored but exploitable (and probably quite complex) substrate behavior. This would enable us to enumerate new design principles and develop new simulation models that express the newly observed behavior. Then, the entire body of existing optimization and design techniques, combined with the efforts of human designers, could work with these new principles and models to generate as yet unimagined designs that would be efficient and robust.

b) Inventing Circuits that are Robust in a Novel Way: One property observed in some invented evolved circuits is their intrinsic robustness. For example, sometimes parts of an evolved circuit can be excised without harming its functionality. This intrinsic robustness has arisen purely from evolutionary dynamics. It can be interpreted as stemming from a principle akin to evolutionary "survival of the flattest" rather than the typically recognized evolutionary principle of 'survival of fittest'. In nature the environment is relatively noisy, the replication process of inheritance is subject to copy errors and the variation processes of mutation and crossover introduce random, unanticipated change. Any organism that through generations has survived this set of disruptive influences has a genetic design that is inherently robust for survival. In artificial circuit evolution this same robustness of design can be fostered by setting up the algorithm to use high mutation rates, high selective pressure and no elitism. These mechanisms guide the final evolved circuit to be robust to the changes created by the mutation operator. For example, if the mutation operator removed one or two signal links between components, the evolved circuit will end up being robust to removal of one or two of its signal links. This does not carte-blanche imply that the circuit will never fail if one or two signal links are removed. But, there is nonetheless a lower likelihood of failure in such circumstances. This probabilistic and innovative way of achieving robustness is unique to evolution. In contrast, humans resort to deterministic methods like redundancy or majority-voting. The literature reveals an initial set of papers by Thompson and Layzell[30, 16] in this vein. The initial results are a small, marginally encouraging step but the level of circuit robustness did not equal or surpass the quality of human methods. Thus it remains an open question as to whether evolution can invent new design principles for achieving robustness that are better than those *presently* invented by humans.

c) Inventing Circuits that Use Human Design Principles in Novel Ways: In the case of exploiting substrate behavior (see a) above), invention is being focused below human abstraction. The evolutionary algorithm is being used to explore the possibilities of a new *language* of design that humans are uncomfortable with. On a different tact, the community often provides the evolutionary algorithm with *human level* abstractions as its building block vocabulary (e.g. transistors or logic gates). When this is done, evolution can still invent novel circuits.

To identify the ways in which invention occurs, it is helpful to examine the design process of humans. Human designers use first-order computational models of devices and an intuition of the higher order effects. For verification, designers move to circuit simulators that more accurately capture device behavior. The more accurate (and detailed) the simulator, the more expensive the simulation. Designers finally depend on the fact that a circuit that works well on simulation will also work well when actually fabricated as an integrated circuit. In what ways does evolutionary invention contrast to this framework?

One way is to avoid using models or simulation at all. It appears that success with this approach depends on the extent to which human design principles in building block specification and interconnection are respected. For example, researchers have used FPTAs [15, 34, 26] with the goal of inventive circuits starting from the transistor while not enforcing assumptions of transistor saturation, zero gate current, etc. The custom-made devices of these projects also allow varying degrees of transistor interconnect. These designs obviously do not suffer from modeling inaccuracies. They are, however, susceptible to over-specializing to the device physics that transistor models sidestep. This has been addressed by what is called mixtrinsic evolution [27]. Mixtrinsic evolution uses both simulators and an FPTA to balance model inadequacy with hardware brittleness. To our best knowledge, no circuit produced on the FPTAs has shown to be an alternative to an existing design for a particular functionality, which has been published in an electronic journal for the merit of the result and not the technique. The limitations imposed by the FPTA's architecture, the non-idealities of the switches and lack of robustness of the design have been persistent challenges. Another set of approaches use FPAAs. They have been limited in inventive success by the relatively small scale of the devices, though [29] invented a controller. No published designs or application of designs has occurred.

Another way in which invention occurs is when the evolutionary algorithm works with circuit simulators and may exploit aspects of the simulation models that humans ignore or miss. This exploitation, unfortunately can run in two directions: it may be profitable for more efficient designs or it may just be a capitalization of a shortcoming of the model that a human never normally encounters. Some models are not sufficiently developed to capture all regions of operation equally well and are biased to be accurate in regions where humans use them. Another challenge of exploiting models is that they are not the entire source of domain knowledge that goes into the design. In fact, humans implicitly address issues like robustness by *choosing* to work in only certain regions of the model. By not respecting this human knowledge, evolution must take the added burden of incorporating robustness explicitly in its design. Examples are Koza [14] and Mesquita [35]. Many circuits produced by Koza have seemingly been inventions in this vein. However, it is legitimate to question their actual realizability and robustness. Though the invented circuits have been granted patents, they have not been published in any analog journals or conferences.

A final way to invent is by completely signing on to human methodology but leveraging the efficiency of well balanced exploitation and exploration by the EA to identify solutions that humans have simply not had time or resources to discover. This restricts some of the previously identified innovation possibilities while circumventing their problems. This restricted form of invention has been strongly criticized, often to the point of arguing there is no invention. However, rejection would extend to all the innovations analog engineers have created in the last several years which is clearly not insignificant! Examples are Grimbleby [9], Horrocks [12], Lohn [18] and Aggarwal [1] among others.[1] The separate works of Grimbleby, Horrocks and Lohn mostly concentrated on design of filters and amplifiers. For filters, only frequency-domain characteristics were considered, leaving phase and time-domain characteristics unattended. Without this analysis, the circuit is not useful to the analog design community. The work of Horrocks did find space in analog-design conferences with its interesting approach to address preferred valued components and robustness. On the other hand, Grimbleby and Lohn's work often did not transfer to analog journals. In the cases it did, it re-

[1] Though Horrocks worked with simulators since most of his work used passive components for synthesis the work is mentioned here.

lied for publication not only on the result, but also on the technique. The work of Stoica with polymorphic circuits [28] demonstrated the invention of a new class of circuits, however search for a good application for such functionality is still on [6]. Aggarwal's work was concentrated on evolution of sine-wave oscillators. This is so far the only work with results published on their own merit in analog design journals. The authors have invented 18 new oscillators, which have been spread over 3 publications [2, 3, 13] in analog journals.

4 Assessment Today

Our success criteria demand closure from the evolution of a lab proof of concept to a real world "product" accepted by hardware engineers. Has closure been achieved? The brief survey above shows that though there has been lot of exciting ideas and research in evolvable hardware, the promise of a successful invention system has been sparsely delivered.

One response is to look to our relative successes and see what weaknesses they expose or opportunities they highlight. We have exploited huge computational resources (e.g. running spice for many distributed simulations). Can we accurately state that this resource usage is not the source of the success and evolution is instead? If evolution is the source, this indicates we need to pursue an understanding of distributed, very large scale evolutionary algorithm dynamics that seem to be different from those at smaller scale. We have also capitalized on domain knowledge. For example, oscillators evolved by Aggarwal used knowledge from the circuit domain to scale down the size and complexity of the search space. This yielded relatively small size oscillators successfully. There must be other small circuit design problems where evolutionary invention could be exploited.

Another response is to ask why this is the case. In general, part of the reason is that we have framed our invention goals alongside a very advanced state of the art. The state-of-art analog circuits use as many as 50 to 100 transistors and the number of performance goals each satisfies is anywhere between 5 to 20. We are trying to re-create the invention process of human circuit design from scratch at a time when humans are very far ahead! While our current attempts are not futile, the large gap we need to close indicates where further attention might be focused. Do current approaches scale to large, complex designs that must meet a large number of performance specifications? For instance, a simple op-amp can have as many as 10 different specifications including gain, phase margin, unity gain frequency, noise figure, slew rate, output swing, offset, etc. To invent circuits that are actually useful to the analog community, all of these criteria must be considered by our algorithms. We need to solve large multi-objective optimization problems that translate to the variety of important performance goals. In [37], multi-objective evolutionary algorithms were used for circuits, but it was just for setting sizes of transistors and not inventing topology. We believe that it is appropriate that our community now addresses the same problems the broader evolutionary algorithms community is tackling. We all need to develop a good understanding of how to evolve modular and complex systems. Whereas earlier it seems that our special identification within the evolutionary algorithms community has been helpful, in this circumstance, it may be better to join forces.

We can (and surely will) play catch-up with the analog circuit design community. Proceeding in this manner will allow us to improve our algorithms toward the design of evolutionary invention systems. We might derive new and useful substrate characteristics. The pressure will be to quickly use these characteristics in an analog solution of current scale and show that the payoff is sufficiently large. We will plausibly succeed in addressing how to scale up our algorithms. The pressure will be to do so fast enough to eventually catch and address the challenges of that domain's state of the art. The circuit design community by their own evaluation has already squeezed performance to the maximum at small transistor level circuits. They have moved up the hierarchy of design to in novate at the system level which gives them a higher payoff in efficiency. Therefore, there is pressure to progress our state of art quickly. The lessons we learn along the way should be immediately transferable to other domains: for e.g., they will extend *Artificial Intelligence*. Thus this route to success is legitimate on the scientific front. However, it has to be somewhat called into question because, while we catch up, we will not be able to claim we solve open problems.

5 New Options for Evolutionary Invention

As we paint such an uphill struggle, is there any hope? Our rejoinder is "yes!". We believe there to be plenty of room for optimism. The direction lies in pursuing untapped opportunities where invention is necessary. In observing how human circuit design advanced, we found our clues. When the field was young, four transistor circuits such as the wilson-current mirror, cascoded-amplifier, and Gilbert's cell [25] were considered useful inventions. The situation was very open ended with no obvious waypoints. At that point, efficient abstraction layers (i.e. the intermediate waypoints) had not been "discovered" and it was not fully possible to even envision the kinds of applications that current solutions might address. Imagine trying to foresee the personal computer when the IBM 360 was designed. Reflecting this insight on our current approach reveals it to appear derivative and backward looking. How can we, instead, proceed in an open ended way? What circumstances present themselves today that are as open ended as circuit design was at its outset?

The answer translates to taking the framework of an evolutionary invention system to nascent technologies. Nascent technologies are ripe for development but the efficient abstraction layers that will allow them to culminate into efficient application solutions have not yet been found. There is a gap between technology knowledge and application development. In these kinds of circumstances, an evolutionary invention system can partner with humans or even preempt them. The advantage of getting in early manifests itself in numerous ways. The open problems will be small. There will be no risk of results being interpreted as re-invention. There will be real needs and, among them, the most important one will be the invention of fundamental advances in an emerging field. In such a situation, our community can feel confident. Our community has documented successes of solving smaller problems well. While we have not totally scaled up our algorithms, they are mature enough for the smaller problems that will exist and we could scale them hand in hand with advancement. Instead of playing catchup, we will be positioned on the open frontier of advancement. Open ended problem landscapes are intimidating yet compellingly valuable. It is always harder to advance algorithms when any solution is unknown. The security and learning value of the circuit design domain will have to be sacrificed.

A few brief examples of open ended invention opportunities are:

New design regimes with analog elements: The community might consider addressing the design of circuits that work in the sub-threshold region of MOSFET transistor [4]. This area has gained considerable interest in the last 10-15 years due to its low power operation. There still seems to be opportunities for circuit level innovation. Another possible area is that of chaotic circuits, where the field took off around 20 years ago with the publication of the first chaotic circuit [22]. Another possibility is the domain of current-mode signal processing [33]. These fields are also not completely new, but relatively newer than conventional analog design. Moreover, some of them, for instance sub-threshold circuits, attract industry attention.

System design with novel computational paradigms: New computation paradigms arise on occasion. For example, digital abstraction emerged from analog systems. Our community should partner with the inventors of these paradigms. An evolutionary algorithm can be used for evaluating their versatility and facilitate invention using them.

In one example, in 1997, Brockett [7] demonstrated a sorting computation for discrete numbers that is purely built from analog circuits. Evolutionary invention could be used to determine what sort of higher level functions could be built from the sorting computation. It could be exploited to integrate the sorting computation with other computation for novel applications. It could identify other similar discrete operations. Additionally, biological cells are regarded as computation elements. So are neurons. Is there an opportunity to invent from cellular or neurological computation models?

Quantum computation (QC) is another example where evolutionary invention might be exploited in a new computational paradigm [21]. The two main questions and issues that EHW addresses for quantum computation are: (1) quantum algorithm discovery; and (2) control of QC operation. The basic approach to both of these problems is shown schematically in figure 1.

Quantum Algorithm Discovery: The government roadmap for quantum computing states that "The search for new quantum algorithms is one of the biggest challenges in quantum computation today... The exploration of quantum algorithms is therefore of fundamental importance"[11]. While traditional mathematical investigation may result in development of new algorithms, given the importance of algorithms to the field, alternative approaches are worth considering.

A reconfigurable (or reprogrammable) QC could be used with an evolutionary algorithm to search for new quantum algorithms. Existing machine-based search methods for quantum algorithms rely on simulations run on classical computers [20], and therefore cannot develop algorithms for QCs with more than 25 qubits (the maximum number that can currently be simulated on a classical computer). However, if it could be built (and it may not be for a long time yet), a reconfigurable QC with more than 25 qubits could itself be used to search for algorithms. Because many QC instantiations (in particular superconductive approaches) are reconfigurable (the gates and qubits can be switched on or off, or tuned in frequency or coupling strength dynamically either in between calculations or during a calculation), it is possible to control a QC with an ancillary classical computer which reconfigures or reprograms the QC, and then tests the configuration against the problem of interest.

Control of Computer Operation: Quantum computing control at the level of the physical qubit is a complex analog control problem: a QC has been compared to a symphony [17], but a symphony played in a noisy room where the instruments (the qubits and gates) must all be tuned to perfection, played (controlled) flawlessly (at least on the timescale over which error-correction may be performed), and the audience (the source of decoherence) must be kept quiet (again, on the timescale over which error-correction may be performed). This analogy clarifies why control problems are hard.

Despite being an important and hard problem, control has not come to dominate QC-related literature (for the few exceptions, see [36] and [24]). This is because at present the control problems are generally masked by decoherence (using the symphony analogy, we cannot tell how the instruments are played because the audience will not keep quiet). So far, only nuclear-magnetic-resonance (NMR)-based QCs, because of their long decoherence times, have had to grapple seriously with the problem of control. And of all technologies, NMR-based QCs are probably the easiest to control for three reasons: (1) the qubits are all identical; (2) five decades of academic and industrial development have gone into development of NMR technology; and (3) the qubit evolution occurs on the microsecond timescale. But even with these advantages, NMR has recently turned to numerical optimization [8] to achieve adequate gate fidelity. By comparison, a solid-state qubit with nanosecond decoherence times and huge device-to-device variance has no hope of control based on solely numerical optimization; an adaptive approached based on in situ control optimization is the only viable approach for such systems.

New technology substrates: Materials technologies are experiencing a rapid growth spurt on the wave of better fabrication techniques. One such technology is nanoscale photonics[5]. To date, photonic circuits (on conventional silicon-based substrate) are in development for custom communications and signal processing applications. As signal processors, they have great potential to be more energy efficient than conventional DSP technology. The research is sufficiently mature to conceptualize a programmable photonic processing array as a technology platform for next generation signal processors and computers. The array would be a lattice of optical waveguides, steering photons through channels and optical couplers that would be used to route and mix photons of different channels. The control of the routing is the programmable aspect of the array. This control lies at the optical couplers. The overall challenge is to develop techniques for mapping algorithms onto the lattice filter structure, but there is a wealth of smaller problems that must be solved first. As illustrated in Figure 2.1-3, there are three possible evolutionary invention scenarios for a programmable, photonic processing array:

1. Humans will choose the lowest level abstraction that is appropriate as a foundation for formulating layered solutions for the application. For example, digital gates may be selected. Evolutionary invention will be used to discover the implementation of this lowest level abstraction in terms of basic photonic processing elements.
2. The appropriate computational primitives of the substrate will be identified by humans. These will be transferred to the evolutionary algorithm as building blocks. One possibility is to regard the coupler as if it is a tunable low pass filter and use these filters as building blocks. Optionally appropriate design principles will also be selected by humans. They will be transferred as genetic operators or some means of constraining and structuring the ways in which subsolutions are combined. The evolutionary invention system will use this abstraction and structural knowledge from the domain to efficiently search the space of possible designs. It may venture where humans would not due to cognitive limits of complexity and the awkwardness of intuitive design with the building blocks. It may exploit design opportunities unnoticed by human designers.
3. Neither the appropriate computational primitives nor design principles will be known when the invention system is employed. Nor will the appropriate level of abstraction for formulating layered solutions be known. In a much longer running context, the evolutionary invention system will evolve different solutions from different combinations of primitives. Over a long evolutionary time scale, the more general, modular, reusable primitives will be identified by arising multiple times in multiple circumstances. Evolution will invent (or discover) the appropriate language of the substrate and reveal the appropriate abstractions that facilitate solutions of upward complexity. The result is an invented design methodology. This framework may operate on a startlingly large scale compared to evolutionary invention on silicon. Potentially, photonic arrays offer reconfigurability and testing on the nanoscale. So population size and generations could be as large scale as the best invention context known - nature on Earth.

6 Summary

To summarize, our interest is in evolutionary electronics and its exploitation for invention. We have considered the ways in which the community currently tries to exploit invention potential. Against the yardstick of passing industry

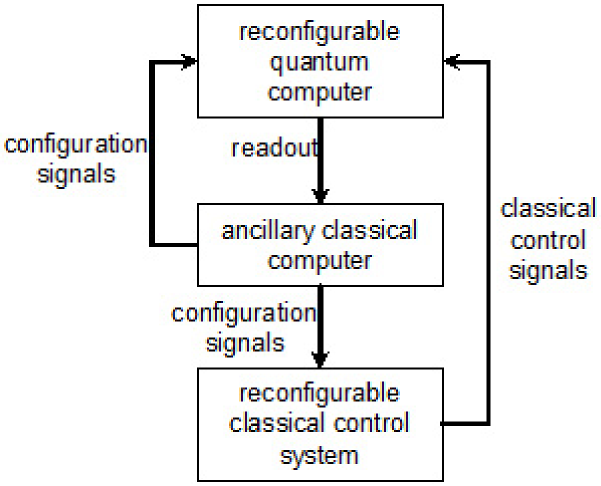
reconfigurable quantum computer
configuration signals
readout
ancillary classical computer
classical control signals
configuration signals
reconfigurable classical control system

Figure 1. Schematic diagram showing quantum-computer control system that includes quantum hardware in the feedback loop.

muster, our current results are immature. They have encountered some sobering, major roadblocks: substrate exploitation is challenged by robustness, circuits that are robust to failure are not *presently* better than those invented by humans, and the novel designs that have been invented are not sufficiently complex, nor do our algorithms handle large scale design specifications. We can continue to advance on these routes to invention with scholarly benefit. However, to make our research pragmatically useful and better appreciated, we should also look to more open ended problem domains. In lieu of solely pursuing a derivative process with circuit design we should consider generative opportunities. Some examples are new design regimes with analog elements, system design with novel computation paradigms and new technology substrates.

Acknowledgements

We would like to thank Rajeev Ram, MIT, for suggesting to us the idea of applying evolvable hardware methods to the arrayed photonic signal processing architecture.

References

[1] V. Aggarwal. Evolving sinusoidal oscillators using genetic algorithms. *The 2003 NASA/DoD Conference on Evolvable Hardware*, 00:67, 2003.

[2] V. Aggarwal. Novel canonic current mode DDCC based SRCO synthesized using a genetic algorithm. *Analog Integr. Circuits Signal Process.*, 40(1):83–85, 2004.

[3] V. Aggarwal, S. Kilinc, and U. Cam. Minimum component SRCO and VFO using a single DVCCC. *Analog Integr. Circuits Signal Process.*, 49(2):181–185, 2006.

[4] A. Andreou, K. Boahen, P. Pouliquen, A. Pavasovic, R. Jenkins, and K. Strohbehn. Current-mode subthreshold MOS circuits for analog VLSI neural systems. *Neural Networks, IEEE Transactions on*, 2(2):205–213, 1991.

[5] T. Barwicz, M. A. Popovic, P. T. Rakich, M. R. Watts, H. A. Haus, E. P. Ippe, and H. I. Smith. Microring-resonator-based add-drop filters in SiN: fabrication and analysis. *Optics Express*, 12:1437–1442, 2004.

[6] M. Bidlo and L. Sekanina. Providing information from the environment for growing electronic circuits through polymorphic gates. In *GECCO '05: Proceedings of the 2005 workshops on Genetic and evolutionary computation*, pages 242–248, New York, NY, USA, 2005. ACM Press.

[7] R. Brockett. A rational flow for the toda lattice equations. *Operators, Systems, and Linear Algebra (U. Helmke et al. eds.), B.G. Teubner, Stuggart*, 2(2):205–213, 1997.

[8] E. M. Fortunato, M. A. Pravia, N. Boulant, G. Teklemariam, T. F. Havel, and D. G. Cory. Design of strongly modulating pulses to implement precise effective hamiltonians for quantum information processing. *Journal of Chemical Physics*, 116:7599–7606, 2002.

[9] B. Grimbleby. An automatic analogue network synthesis using genetic algorithms. In A. M. S. Zalzala, editor, *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA*, volume 414, pages 53–58, Sheffield, UK, 12-14 Sept. 1995. IEE.

[10] D. A. Gwaltney and M. I. Ferguson. Intrinsic hardware evolution for the design and reconfiguration of analog speed controllers for a DC motor. In *Evolvable Hardware*, pages 81–90. IEEE Computer Society, 2003.

[11] T. Heinrichs and R. Hughes. Quantum information science and technology rodmap, part 1: Quantum computation, version 2.0, 2004.

[12] D. Horrocks and A. Y. Khalifa. Genetically derived filters circuits using preferred value components. In *Proc. IEE Colloquium on Analogue Signal Processing*, pages 4/1–4/5. Oxford, UK,, 1994.

[13] S. Kilinc, V. Jain, V. Aggarwal, and U. Cam. Catalogue of variable frequency and single-resistance-controlled oscillators employing a single differential difference complementary current conveyor. *Frequenz: Journal of RF-Engineering and Telecommunications*, 60(7-8):142–146, 2006.

[14] J. R. Koza, M. A. Kean, M. J. Streeter, W. Mydlowec, J. Yu, and G. Lanza. *Genetic Programming IV:Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers, 2003.

[15] J. Langeheine, M. Trefzer, D. Brüderle, K. Meier, and J. Schemmel. On the evolution of analog electronic circuits using building blocks on a CMOS FPTA. In K. Deb, R. Poli, W. Banzhaf, H.-G. Beyer, E. Burke, P. Darwen, D. Dasgupta, D. Floreano, J. Foster, M. Harman, O. Holland, P. L. Lanzi, L. Spector, A. Tettamanzi, D. Thierens, and A. Tyrrell, editors, *Genetic and Evolutionary Computation – GECCO-2004, Part I*, volume 3102 of *Lecture Notes in Computer Science*, pages 1316–1327, Seattle, WA, USA, 26-30 June 2004. Springer-Verlag.

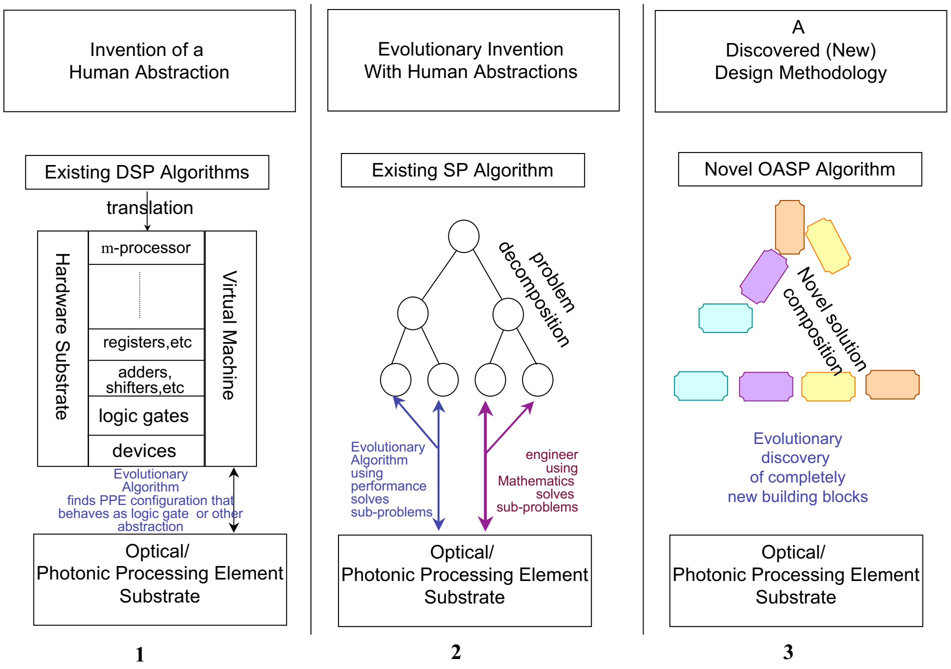
Invention of a Human Abstraction
Existing DSP Algorithms
translation
m-processor
registers,etc
adders, shifters,etc
logic gates
devices
Hardware Substrate
Virtual Machine
Evolutionary Algorithm finds PPE configuration that behaves as logic gate or other abstraction
Optical/ Photonic Processing Element Substrate
1
Evolutionary Invention With Human Abstractions
Existing SP Algorithm
problem decomposition
Evolutionary Algorithm using performance solves sub-problems
engineer using Mathematics solves sub-problems
Optical/ Photonic Processing Element Substrate
2
A Discovered (New) Design Methodology
Novel OASP Algorithm
Novel solution composition
Evolutionary discovery of completely new building blocks
Optical/ Photonic Processing Element Substrate
3

Figure 2. Evolutionary invention approaches for programmable photonic processing arrays.

[16] P. J. Layzell and A. Thompson. Understanding inherent qualities of evolved circuits: Evolutionary history as a predictor of fault tolerance. In *ICES '00: Proceedings of the Third International Conference on Evolvable Systems*, pages 133–144, London, UK, 2000. Springer-Verlag.

[17] S. Lloyd. Quantum-mechanical computers. *Scientific American*, 273:44–48, 1995.

[18] J. D. Lohn and S. P. Colombano. A circuit representation technique for automated circuit design. *IEEE-EC*, 3(3):205, September 1999.

[19] J. D. Lohn, G. S. Hornby, and D. S. Linden. Rapid reevolution of an X-band antenna for NASA's space technology 5 mission. In T. Yu, R. L. Riolo, and B. Worzel, editors, *Genetic Programming Theory and Practice III*, chapter 5, pages 65–78. Springer, Ann Arbor, 12-14 May 2005.

[20] P. Massey, J. A. Clark, and S. Stepney. Evolving quantum circuits and programs through genetic programming. In K. Deb, R. Poli, W. Banzhaf, H.-G. Beyer, E. Burke, P. Darwen, D. Dasgupta, D. Floreano, J. Foster, M. Harman, O. Holland, P. L. Lanzi, L. Spector, A. Tettamanzi, D. Thierens, and A. Tyrrell, editors, *Genetic and Evolutionary Computation – GECCO-2004, Part II*, volume 3103 of *Lecture Notes in Computer Science*, pages 569–580, Seattle, WA, USA, 26-30 June 2004. Springer-Verlag.

[21] P. Massey, J. A. Clark, and S. Stepney. Evolution of a human-competitive quantum fourier transform algorithm using genetic programming. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1657–1663, New York, NY, USA, 2005. ACM Press.

[22] T. Matsumoto, L. Chua, and S. Tanaka. Simplest chaotic non-autonomous circuit. *Physical Review A (General Physics)*, 30(2):1155–1157, 1984.

[23] J. F. Miller. Digital filter design at gate-level using evolutionary algorithms. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1127–1134, Orlando, Florida, USA, 13-17 1999. Morgan Kaufmann.

[24] K. M. Obenland and A. M. Despain. A parallel quantum computer simulator, 1998.

[25] A. S. Sedra and K. C. Smith. *Microelectronic circuits, 2nd ed.* Holt, Rinehart & Winston, Austin, TX, USA, 1987.

[26] A. Stoica, R. Zebulum, D. Keymeulen, R. Tawel, T. Daud, and A. Thakoor. Reconfigurable VLSI architectures for evolvable hardware: from experimental field programmable transistor arrays to evolution-oriented chips. *IEEE Transactions on VLSI Systems, Special Issue on Reconfigurable and Adaptive VLSI Systems*, 9(1):227–232, 2001.

[27] A. Stoica, R. S. Zebulum, and D. Keymeulen. Mixtrinsic evolution. In *ICES '00: Proceedings of the Third International Conference on Evolvable Systems*, pages 208–217, London, UK, 2000. Springer-Verlag.

[28] A. Stoica, R. S. Zebulum, and D. Keymeulen. Polymorphic electronics. In Y. Liu, K. Tanaka, M. Iwata, T. Higuchi, and M. Yasunaga, editors, *ICES*, volume 2210 of *Lecture Notes in Computer Science*, pages 291–302. Springer, 2001.

[29] M. A. Terry, J. Marcus, M. Farrell, V. Aggarwal, and U.-M. O'Reilly. Grace: Generative robust analog circuit exploration. In F. Rothlauf, J. Branke, S. Cagnoni, E. Costa, C. Cotta, R. Drechsler, E. Lutton, P. Machado, J. H. Moore, J. Romero, G. D. Smith, G. Squillero, and H. Takagi, editors,

EvoWorkshops, volume 3907 of *Lecture Notes in Computer Science*, pages 332–343. Springer, 2006.

[30] A. Thompson. Evolutionary techniques for fault tolerance. In *Proceedings of the UKACC International Conference on Control, pp. 693–698. IEE.*

[31] A. Thompson. An evolved circuit, intrinsic in silicon, entwined with physics. In *ICES*, pages 390–405, 1996.

[32] A. Thompson and P. J. Layzell. Evolution of robustness in an electronics design. In *ICES*, pages 218–228, 2000.

[33] C. Toumazou, F. Lidjey, and D. Haigh. *Analog IC Design: The Current-Mode Approach*. Peter Peregrinus, UK, 1990.

[34] M. Trefzer, J. Langeheine, K. Meier, and J. Schemmel. A modular framework for the evolution of circuits on configurable transistor array architectures. *First NASA/ESA Conference on Adaptive Hardware and Systems*, 0:32–42, 2006.

[35] P. F. Vieira, L. B. Sa, J. P. B. Botelho, and A. Mesquita. Evolutionary synthesis of analog circuits using only MOS transistors. In R. Zebulum, D. Gwaltney, G. Hornby, D. Keymeulen, J. Lohn, and A. Stoica, editors, *Proceedings of the NASA/DoD Conference on Evolvable Hardware*, pages 38–45, Seattle, Washington, 2004. IEEE Computer Society.

[36] L. Viola, S. Lloyd, and E. Knill. Universal control of decoupled quantum systems. *Physical Review Letters*, 83:4888–4891, 1999.

[37] R. S. Zebulum, M. A. Pacheco, and M. Vellasco. A multiobjective optimisation methodology applied to the synthesis of low-power operational amplifiers. In I. J. Cheuri and C. A. dos Reis Filho, editors, *Proceedings of the XIII International Conference in Microelectronics and Packaging*, volume 1, pages 264–271, Curitiba, Brazil, 1998.