Analysing Evolvable Cell Design for Optimisation of Routing Options

Daniel M. Ellin
Royal Holloway University of London
Egham, Surrey
TW20 0EK, UK
d.ellin@rhul.ac.uk

Stuart J. Flockton
Royal Holloway University of London
Egham, Surrey
TW20 0EK, UK
s.flockton@rhul.ac.uk

ABSTRACT

Evolution of analogue circuitry on a cell based system can benefit from the use of a generic cell design. A comparative analysis of the re-routing ability of two different cell topologies is used to highlight the difficulty of assessing a generic cell design's fitness for purpose. An analytical method is then proposed for the production of suitable comparable data.

Categories and Subject Descriptors

B.8 [**PERFORMANCE AND RELIABILITY**]: Performance Analysis and Design Aid; B.8 [**PERFORMANCE AND RELIABILITY**]: Miscellaneous

General Terms

Design, Measurement

Keywords

Cell design, Analogue circuitry, Analogue evolvable hardware, Evolvable cells, Cell signal routing

1. INTRODUCTION

The design of analogue electronic circuits is more difficult than digital circuits. The continuous input-to-output response of analogue components, as opposed to the discrete input to output response of digital components, results in a continuous design space for analogue versus a countable number of design options for digital. The continuous analogue design space has been found to be a suitable region for the application of a genetic design process, where both new and traditional solutions to problems have been evolved [5]. There are many methods for the application of Genetic Algorithms (GA) to the design process, component libraries, variable components [3, 10] and the cell based applications described later have all been devised with varying success. Generally the design process consists of a population of candidate circuits, the required input-output combinations are applied to test the suitability of each circuit as a solution to the problem, the most suitable providing the basis of the next population.

Cell based analogue circuit evolution has been found successful for both generic [11, 8] and more specialised [9, 12] cell designs.

When designing a cell it is important to strike a balance between adaptability and functionality. An overtly functional design may provide a platform for the rapid and easy evolution of known designs, and in some applications this may be what is needed. A more generic design of cell has a search space released from some of the traditional design constraints introduced by a design tailored for a particular purpose. By enhancing the freedom of design the opportunities for GA-led innovation are increased.

The removal of constraints when designing a general cell is not as simple as it seems. Some design constraints must still be adhered to (e.g. prevention of live-earth shorts etc...) in order that the cell remains realisable.

Analysis of a general cell to determine if it is a 'good design' is complicated by contradicting conditional arguments over what properties are considered good. What is a good cell, with many design advantages for circuits of type A, may be a poor cell design for circuit type B, where the design advantages become disadvantages. Therefore it is fair to propose a good generic cell does not exist[1].

The cell designer, having no universal cell design to use, must consider the fitness for purpose of his cell, whilst not reverting to the design conventions that would break the generality of any design.

Analysing the abilities of a cell, purposefully designed to be as general as possible, without resorting to conventional circuits as testing agents is a difficult task. Even a simple design modification can easily have many cascading effects that may have beneficial, detrimental and even ambiguous results(see Figure 1). We propose that by isolating a key feature beneficial for a particular subtype of circuit, a comparative analysis of this feature would provide information about the suitability of a circuit, without resorting to comparison to conventional designs.

A probable further benefit of this restrictive analysis, is the identification of key design limits particular to a feature.

We take two cell designs (NEWS and HOLNES) based on our definition for a general cell, and attempt to analyse their ability to re-route signals. A cell design which has good re-routing ability is beneficial to systems where the circuit is expected to be subject to breakage and repair, thus the

GECCO'07, July 7-11, 2007, London, England, United Kingdom
Copyright 2007 ACM 978-1-59593-698-1/07/0007 ...$5.00.

[1] This is not to say that a generic cell does not exist

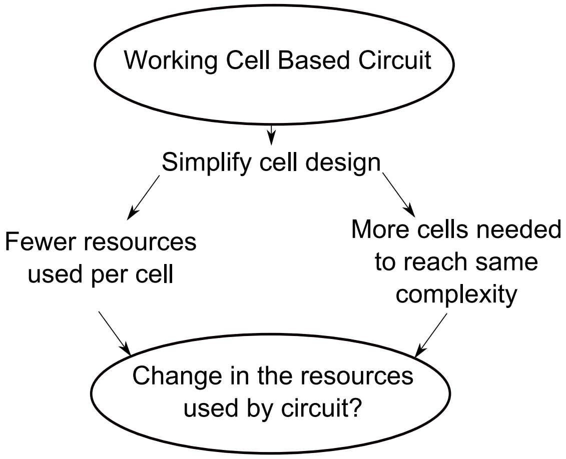
Working Cell Based Circuit
Simplify cell design
Fewer resources used per cell
More cells needed to reach same complexity
Change in the resources used by circuit?

Figure 1: An example of an improvement having a conditional benefit. The resource use is conditional on the circuit in question, the smaller cell may benefit one circuit, whilst harming another.

approach taken is to asses the ability to repair a break via re-routing. The task is shown to be difficult, and an analytical approach for a more complete analysis is proposed. Further, from early results it is proposed that the number of unused cell IO's at maximum cell function is the defining factor for this ability.

2. GENERAL CELL DESIGN

The design of a general cell can be approached from different perspectives. Apart from prescribing some rules that are needed to ensure a basic level of generality, our approach at this stage also adds a few extra rules to aid in analysis.

The cell needs to be approach invariant. This means the options available within an unused cell must be identical for all points of access to the cell (cell-IO's), defined as nodes on surface B in Figure 2. This is not an absolute correspondence, but a relative relationship. i.e. if node α does not have a direct path to the node one to the left, node β must not have a direct path to the node on it's left. This relationship can be interpreted as nodes on surface B having rotational symmetry.

The functional-routing (f-r) nodes (defined on surface A in Figure 2) must all have access to all cell-IO nodes as a result of the approach invariance rule above.

The maximum number of f-r nodes that can be usefully connected at any one time, defined as $Max(frIO)$ must follow the relation:

$$Max(frIO) \leq cIO$$

where cIO is the number of cell-IO's. This relationship ensures that a functional unit will have enough IO's available to perform all designed functions.

Other than these rules, the designer is free to choose the contents of the functional block, the number of IO's, and the arrangement of routing.

2.1 The NEWS Cell

The NEWS (North, East, West, South) Cell is a relatively simple design which utilises a 'square' design for easy interconnection (see Figure 3).

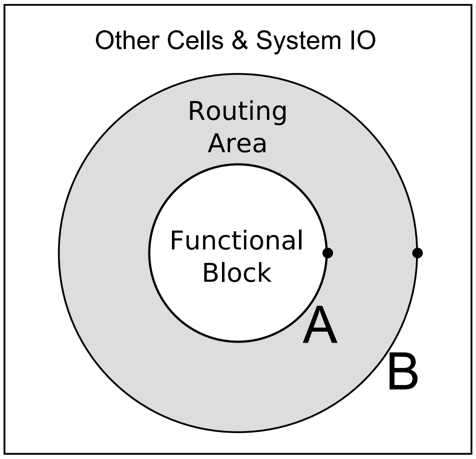

Figure 2: A General Cell Overview. A general cell consists of two concentric areas. The innermost area, the functional block, contains the work elements of the cell (transistors, diodes, capacitors etc. . .) and power/earth lines. The surface A contains nodes for interface between the functional block and the routing area. The routing area contains lines linking nodes on surface A, and nodes on surface B, which are interfaces between the routing area and the exterior (other cells and system IO's).

It is composed of 4 cell-IO nodes, and 3 f-r nodes. Any cell-IO node can connect to any other cell-IO node, or multiple cell-IO nodes. Any f-r node can connect to any non-f-r node that is not connected to another f-r node. Thus the NEWS design conforms to the General Cell Design Rules set out in Section 2.

In this design, an active node link is not compulsory, thus a setup where all nodes are mutually isolated is valid, and considered to be the 'cell off' state. All node connections are based on a single switch, thus a node cannot refuse any legal connection[2]. This cell design provides a total of 2624 possible connection setups. This figure discounts the contents of the functional block, any possible variation within which would result in an increase in the number of configurations.

Each cell-IO node also acts as the link to an adjoining cell if such a cell exists, by connecting to a corresponding cell-IO node in that cell. If there is no cell to connect to, a system IO[3] line is created instead. There is no switch on an interconnect node, thus two connected T1 nodes from two different cells behave as one node, and an active system IO line is fixed to the connected node. A group of connected cells is called an array, the design of which is dictated by the cell design, in this case a square, so the NEWS array is a simple x–y grid pattern.

Even with a relatively simple building block such as the NEWS cell, the combination of even a few cells possesses a large number of different possible setups. For example, an array of 9 NEWS cells with fixed system IO's produces a total of $\sim 6 \times 10^{30}$ individual system setups. More setup

[2] f-r node links can be considered to be multiplexer based, thus activation of one line implicitly deactivates another.

[3] The system being defined as the area around the cell array which provides & receives signals to & from the circuit.

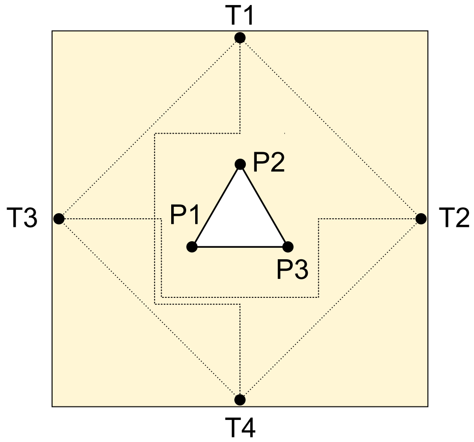

Figure 3: The design of the NEWS cell. Cell-IO nodes are labelled T1–4, and f-r nodes are labelled P1–3. Available cell-IO to cell-IO routes are shown by the dotted line.

options translates to a larger search space for a GA based circuit design.

The cells developed by the University of Heidelberg [4] can be considered a fully defined NEWS cell, with the functional block containing a variable system of transistors. This analogue chip has been produced and shown to be successful in the evolution of a variety of circuits over the last 6 years, including logic circuits [6] and digital to analogue converters[7].

2.2 The HOLNES Cell

The HOLNES Cell is more complicated than the NEWS cell, but maintains many similarities in its design (see Figure 4). The HOLNES cell has 6 cell-IO nodes compared to 4 for the NEWS cell, but both cells contain an identical functional unit. The node interconnects are designed in accordance with the same design specifications used by the NEWS cell, giving total cell-IO node connectivity[4], and 1:1 only f-r to cell-IO connections. Having 6 cell-IO nodes, the cell is best represented as a hexagon, as it tessellates easily for cell interconnects.

By the increase in cell-IO node count by 2, the resources used by a single cell are significantly increased. Correspondingly though the number of possible configurations is raised to $\sim 5.2 \times 10^6$ compared with NEWS's 2624. By following the same interconnect rules as the NEWS cell, to achieve the same number of possible setups found in an array of 5 HOLNES cells, you would need an array of 10 NEWS cells.

2.3 Comparative Figures

Though the cells are similar in many ways, there are a few significant differences that are worth highlighting.

At full activation, where the greatest number of lines are active simultaneously, both cells demonstrate the same behaviour; all cell-IO nodes are connected, and at least one f-r node is connected to a cell-IO node. This setup though is not particulary useful. If routing only is considered, the HOLNES cell has many more options, as shown in Table 1, the most important feature being the ability to route the extra independent signal. Any of these signals could also be routed via the functional unit without interfering with other purely routing signals. The extra ability stretches to situations of maximum f-r node activation, where the news cell has at most a single unused cell-IO node, whereas the HOLNES cell has 3, leaving it with the ability to also route an independent signal.

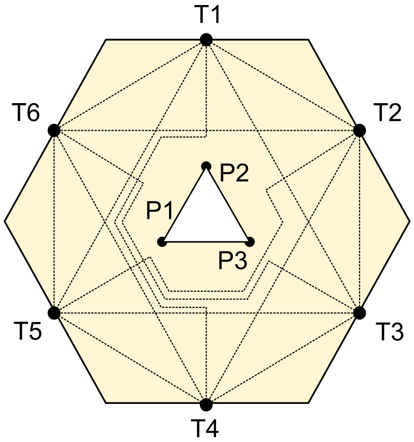

Figure 4: The design of the HOLNES cell.

Table 1: Cell Routing Ability

	NEWS	HOLNES
Maximum Independent Signals Routable	2	3
Single Signal Routing Options	1:1,2,3	1:1,2,3,4,5
Double Signal Routing Options	1:1 & 1:1	1:1 & 1:1,2,3,4, 1:2 & 1:2
Triple Signal Routing Options		1:1 & 1:1 & 1:1

3. RE-ROUTING

A cell design can be self defeating. A cell structure for evolution of circuit types α may be very efficient at such a task, and in doing so waste very little resources. However, when asked to repair a break in such a circuit, the cell design may make the process exceptionally complex, outweighing the benefits of the efficient earlier development. Some have approached this problem by developing GAs that take into account the need for later adaptation. Alternatively, the problem can be approached at the cell design level.

If a link between two nodes[5] that is in use by the circuit is broken, the arrays routing map is broken[6]. For circuit function to continue, this break has to be repaired. In order to measure the cell design's ability, the designer needs to find out: is it possible to repair the break? and, if so, what is the best way to repair the break with the least resource expenditure?

On an infinite array of cells, any break not at a system-IO point is fixable. But what is in consideration is a strictly

[4] any cell-IO node has a direct connection to any other.

[5] the nodes need not be next to each other. A chain of linked nodes has a link between any two nodes.

[6] For the purpose of this argument, instances of parallel routing are not allowed.

1. *Within cell signal re-route*
2. *Local region signal re-route*
3. *Local region all signal re-route*
4. *Local region all feature re-design*
5. *Circuit wide all feature re-design*

Figure 5: Five level scheme for testing re-routing ability. Signal refers to the signal that was travelling down the broken route, all signal refers to all signals, broken or not.

```
A. Isolate region of array suitable for testing level.
B. Test within required bounds for fixed time period.
C. If successfully repaired, goto E. Else goto D.
D. If higher testing level available, progress to next
   testing level and goto A. Else goto E.
E. Record level reached, whether successful, time
   taken. Fetch new test circuit and goto A.
```

Figure 6: Pseudo-code for the testing regime. Levels refer to those found in Figure 5. The Algorithm is initialised on A with level set to 1 and a pre-broken circuit provided.

limited array, and as such there will be breaks that are impossible to fix. Knowing if a break is fixable or not is a very difficult question to answer. There are so many possible array setups that an answer for each and every break is impossible. Consider a 6x6 NEWS array, analysing 1 setup per second (difficult enough due to there being the possibility of 1000's of different breaks on an average setup), it would take in the region of 10^{115} years to test all possible breaks. Even with the massive time savings through pattern matching and intelligent searching, the task is still infeasible for the more complicated cells. Thus a complete answer to 'is it possible to repair the break?' is not forthcoming.

Due to the vast number of possible breaks, for reasons similar to the repair possibility question above no specific fix is going to be the best way to repair all breaks. Work has been produced that demonstrates generic fixes for broken cells; mostly these take the form of a cell shift, either routing the signal via dedicated spare cells, or disabling the affected cell and re-mapping a section of the circuit in an area past the break [1, 2]. These methods require suitable re-routing lines (either dedicated or not), and/or have a reliance upon availability of a suitable axis to shift cells along, neither of which is a requirement of the general system discussed here.

With no easily accessible complete solution, a different analytical system is needed. Maintaining our general approach, we break down the task of re-routing into a hierarchy of measurable goals. The actual re-route remains the primary task, but ideally the task is achievable in a limited timescale, and without great difficulty. A framework for measuring the goals is set out in Figure 5, along with the related pseudo-code in Figure 6. The algorithm shown is applicable to any routing fault, but a complete analysis of a cell can not be based on a single fault. Total analysis, or general case analysis based on reasonable approximations is a feasible approach for the early levels, but production of comparable figures, as shown in Sections 3.1 & 3.2 is highly subjective. Analysis of higher levels is much more complex and requires a different approach as is proposed in Section 4.

3.1 Level 1 - Within-cell signal re-route

The simplest of all possible re-routes is a cell internal re-route. Due to the strictly limited number of variables in this problem for most cell designs, and certainly the two in use here, it falls below the level of being suitable for GA application. Any routing errors within the functional block are not considered as the general designs used here do not specify the functional block contents.

The total connectivity[7] of the routing area in the two designs under consideration simplifies this level of analysis significantly. The ability to re-route around any broken line between any two nodes (either f-r or cell-IO) is determined by the number of free cell-IO nodes. As long as there is one free cell-IO node, re-routing is possible. As cell-IO nodes have a paired existence with any adjoining cell, the availability of cell-IO nodes is not solely determined by the cell's internal configuration. Thus any internally free cell-IO node may be rendered unavailable by the configuration of an adjoining cell.

The particular solution for either of the cells under consideration, with this type of break, is simple but a general answer for either cell is not. Without data of the probability of each situation occurring, the number of assumptions needed to produce figures is prohibitive.

A secondary result of the free cell-IO node dependence is that a local region signal re-route (re-routing without affecting other lines, Level 2 in Figure 5) is also impossible without available cell-IO nodes, and thus an irrelevant consideration for this type of breakage.

3.2 Level 2 - Local region signal re-route

As already stated, simple re-routing outside the cell for a line break is redundant, as, if a node is available for such routing, it is also available for internal re-routing. What must be considered is a different type of signal break, a node failure. In particular, the failure of a cell-IO node as a f-r node failure cannot be addressed at this level of repair, as it requires re-assignment of the functional block.

As found for Level 1 (Section 3.1), a complete analysis would require many assumptions without statistical data being provided, but this time a general analysis has some benefit.

The shortest re-route is the optimal re-route, and there are significant differences between the two cell designs considered here. From Figure 7 it is obvious that the HOLNES cell has a re-routing advantage over NEWS, needing only 1 extra cell and 2 extra nodes, compared to NEWS needing 2 cells and 3 nodes.

[7] All cell-IO nodes can directly connect to any other cell-IO node

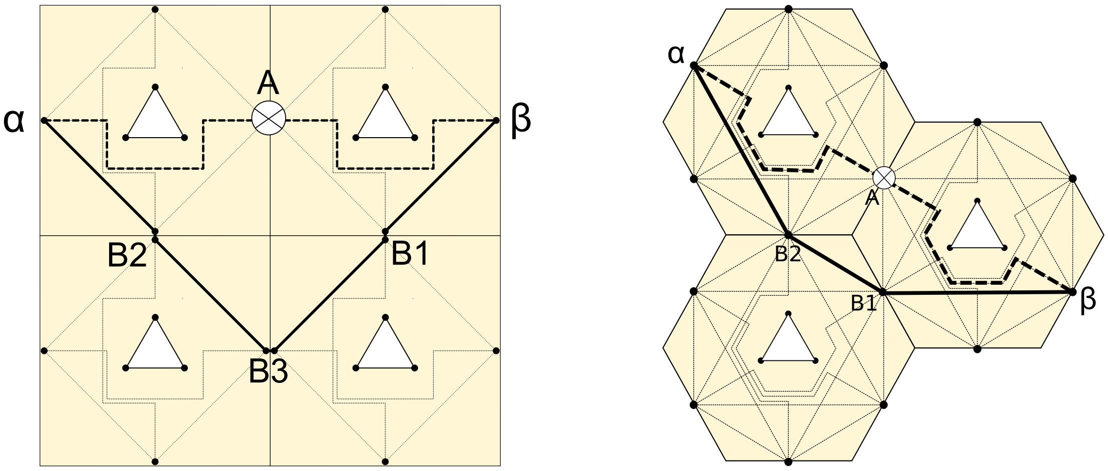

Figure 7: The Shortest Re-route available for NEWS (left) and HOLNES (right) cells. The signal needs transport between nodes α and β in both systems. The dashed line represents the original route, with the cross on the corresponding defective node A. The heavy solid line through the nodes labelled B represents a replacement route that uses the least extra nodes possible.

Table 2: Route Node Count

	NEWS	HOLNES
Shortest	3 Nodes	2 Nodes
1 Ideal Node Unavailable	5 Nodes	3 Nodes
2 Ideal Nodes Unavailable	7 Nodes	4 Nodes

If it is assumed that the cell-IO node in-use count is 50% and is distributed evenly across all available nodes, the probability of the shortest route being available can be calculated. Based on Node path length, the probability of the shortest route being available for the NEWS cell is half that of the HOLNES cell, as it needs one more node. If it is taken into account that for both cell designs the shortest path can be duplicated to the other side of a break (i.e. clockwise re-route instead of ant-clockwise), the probability of at least one of the two shortest routes being available is 0.4375 for HOLNES, and 0.2344 for NEWS.

Examining how the second and third shortest path(s) (see Table 2) grow, the number of nodes needed by the NEWS cell system will generally increase by two for each unavailable node, where the HOLNES cells will only need an extra one[8]. Extending the probability assumptions from previous calculations, the likelihood of a route being available decreases at a much greater rate for NEWS cells than HOLNES cells for every extra unavailable node, particulary when the number of extra cell needed with each extension is taken into account.

Though providing some comparative information about the two cells, the above analysis relies upon the assumption of 50% cell-IO node activity, and an even node use distribution, which is an exceptionally unlikely scenario. What is much more probable is a high density of cell-IO node use concentrated in areas where the functional block is in use, low density where routing is the primary action, and some cells completely unused. For more useful information, and analysis of higher re-routing stages, a different approach is needed.

4. ANALYSIS IMPLICATIONS

For levels greater than 2 in the testing hierarchy (Figure 5) a general analysis is not feasible. Even at levels 1 and 2 (Sections 3.1 & 3.2) the general analysis is unsatisfactory.

The testing scheme set out in Figures 5 & 6 works on a break by break basis but, when the general case is considered, the lack of statistical data forces the use of assumptions, the only other option being a complete analysis of all possible setups[9]. With a non-general system, suitable statistical data could be acquired by analysing a series of known circuit designs; but as we demand generality, the use of known circuits is forbidden. Thus what we propose is the generation of pseudo-circuits that adhere to the basics of circuit design (i.e. all used wires are connected), but do not conform to a predetermined design or function. Combining these circuits with an automated implementation of the testing regime (Figures 5 & 6), will produce test results that have a significantly higher level of analytical strength. By maintaining common variables across the analysis of multiple candidate cells, the new test system (The Pseudo-code of which is outlined in Figure 8) will alow direct comparison of cell designs.

A secondary feature of the suggested test platform is it can be used for developing and matching suitable routing GAs to individual cell designs. By moving the test regime over to pseudo-circuity, a candidate routing GA can be tested over a large problem-set without having to go through the time consuming process of designing a series of cell-design specific real-circuit examples. Additionally, as the cell-design specific routing GA would probably be implemented within the GA design process of real-circuits for that cell-design, the pseudo-circuity removes the chicken-egg design problem.

[8] There exist systems where if the unavailable nodes are arrange in certain patterns, further unavailable nodes will not have such an effect, but this will only occur for unavailable node counts greater than 3

[9] by intelligent complete analysis this can be reduced, but it equates to a complete analysis

A. Generate pseudo-circuit.

B. Select cell containing active circuit, and break suitable element.

C. Run test algorithm on broken circuit and record results.

D. If test circuit considered large enough, Goto B.

Else goto A.

Figure 8: Automated Testing Algorithm for generation of comparative data for circuit analysis. The test algorithm referred to is found in Figures 5 & 6.

5. SUMMARY AND FURTHER WORK

We have shown that analysis of a cell specifically designed to be general is difficult, particulary when restricted from using non-general test circuits. The example of re-routing used highlights specifically the problem of analysis when the number of circuit setups can be so large. The solution of an automated pseudo-circuit analysis for the production of comparative results is suggested, with the further advantage of generalising the process of the specialisation of a GA design for a particular platform.

The plan now is to develop this idea further through implementing a suitable test environment for cell design, and then designing a cross cell comparable re-routing algorithm for the test cells described (Sections 2.1 & 2.2) and some other cell designs not shown. It is hoped that, with the testing of a range of cell designs, task based comparative analysis will become feasible and have a workable degree of rigour.

In parallel it is hoped that the specialisation of GAs to cell-designs can be explored through the same test architecture, with the long term possibility of the development of a shortened GA design process.

And further from the use in re-routing testing, it is hoped to apply the method to the analysis of cells designed for the evolution of multi-objective analogue circuitry. This area is of particular interest, as it is hoped that a GA will be able to find solutions to problems where no good comparison circuit is known. Thus the process of designing the GA to be fit for the task could benefit significantly from test data produced by suitable pseudo-circuitry.

6. REFERENCES

[1] R. Canham and A. Tyrrell. An embryonic array with improved efficiency and fault tolerance. In *Evolvable Hardware, 2003. Proceedings. NASA/DoD Conference on*, pages 265–272, 2003.

[2] S. Dutt and F. Hanchek. Remod: a new methodology for designing fault-tolerant arithmetic circuits. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 5(1):34–56, March 1997.

[3] S. J. Flockton and K. Sheehan. Behavior of a Building Block for Intrinsic Evolution of Analogue Signal Shaping and Filtering Circuits. In *2000 NASA/DoD Workshop on Evolvable Hardware (EH'00)*, pages 117–123. IEEE, 2000.

[4] J. Langeheine, J. Becker, S. Folling, K. Meier, and J. Schemmel. A CMOS FPTA chip for intrinsic hardware evolution of analog electronic circuits. In *2001 NASA/DoD Workshop on Evolvable Hardware*, pages 172–175, 2001.

[5] 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 A. Stoica, J. Lohn, R. Katz, D. Keymeulen, and R. Zebulum, editors, *2002 NASA/DoD Conference on Evolvable Hardware*, pages 76–85. IEEE Computer Society, 2002.

[6] J. Langeheine, K. Meier, and J. Schemmel. Intrinsic Evolution of Analog Electronic Circuits Using a CMOS FPTA Chip. In *EUROGEN 2003 - Proc. of the Fifth Conf. on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems (EUROGEN 2003)*, 2003. ISBN: 84-95999-33-1.

[7] J. Langeheine, K. Meier, J. Schemmel, and M. Trefzer. Intrinsic Evolution of Digital-to-Analog Converters Using a CMOS FPTA Chip. In *2004 NASA/DoD Conference on Evolvable Hardware (EH2004)*, 2004.

[8] J. Langeheine, M. Trefzer, D. Brderle, K. Meier, and J. Schemmel. On the Evolution of Analog Electronic Circuits Using Building Blocks on a CMOS FPTA. In *GECCO 2004 - Genetic and Evolutionary Computation Conference*, 2004.

[9] A. Stoica, D. Keymeulen, R. Zebulum, A. Thakoor, T. Daud, G. Klimeck, Y. Jin, R. Tawel, and V. Duong. Evolution of Analog Circuits on Field Programmable Transistor Arrays. In *2000 NASA/DoD Workshop on Evolvable Hardware (EH'00)*, pages 99–108. IEEE, 2000.

[10] M. Trefzer, J. Langeheine, K. Meier, and J. Schemmel. New Genetic Operators to Facilitate Understanding of Evolved Transistor Circuits. In *2004 NASA/DoD Conference on Evolvable Hardware (EH2004)*, 2004.

[11] M. Trefzer, J. Langeheine, K. Meier, and J. Schemmel. Operational Amplifiers: An Example for Multi-Objective Optimization on an Analog Evolvable Hardware Platform. In *6th International Conference on Evolvable Systems*, pages 86–97. Springer-Verlag, 2005.

[12] R. S. Zebulum, D. Keymeulen, V. Duong, X. Guo, M. I. Ferguson, and A. Stoica. Experimental Results in Evolutionary Fault-Recovery for Field Programmable Analog Devices. In *2003 NASA/DoD Evolvable Hardware Conference*, 2003.

APPENDIX

A. STATISTICAL FORMULAE

The number of different linking arrangements within the two cells (NEWS & HOLNES) used in this paper is found by

$$2^X \sum_{k=0}^{k=N(T_f)} \frac{N(T_c)!}{(N(T_c)-k)!} \qquad (1)$$

Where $N(x)$ = Number of node type x, T_c = cell-IO node type, T_f = f-r node type, and X is the number of cell-IO to cell-IO links found from;

$$X = \frac{\left(N(T_c)^2 - N(T_c)\right)}{2}$$

Discounting system IO's, for a square grid of NEWS cells, side length m, the number of different system setups is given by.

$$\left(2^X \sum_{k=0}^{k=N(T_f)} \frac{N(T_c)!}{(N(T_c)-k)!}\right)^{m^2} \tag{2}$$