Gate-level Morphogenetic Evolvable Hardware for Scalability and Adaptation on FPGAs

Justin Lee and Joaquin Sitte
Smart Devices Laboratory
School of Software Engineering and Data Communications
Queensland University of Technology
GPO Box 2434, Brisbane, Qld 4001, Australia
Email: jm.lee@qut.edu.au, j.sitte@qut.edu.au

Abstract

Traditional approaches to evolvable hardware (EHW), in which the field programmable gate array (FPGA) configuration is directly encoded, have not scaled well with increasing circuit and FPGA complexity. To overcome this there have been moves towards encoding a growth process, known as morphogenesis, however existing approaches have tended to abstract away the underlying FPGA architecture.

Although currently commercially available FPGAs are not the most evolution-friendly platforms, having complex architectures and issues with potentially damaging configurations, evolving circuits on commercially available devices without requiring a move to high-level building blocks is a necessary prerequisite for the adoption of EHW to solving real problems in electronic design, repair and adaptation.

In this paper we present a morphogenetic EHW model where growth is directed by the gate-level state of the FPGA. We demonstrate that this approach consistently outperforms a traditional EHW approach using a direct encoding, in the number of generations required to find an optimal solution, and in its ability to scale to increases in circuit size and complexity.

Issues in EHW problem solvability are also identified, and preliminary work is presented showing that a morphogenetic approach to EHW may be well suited to correcting damaged circuits.

1 Introduction

Evolvable hardware (EHW) has proven its worth in the generation of small novel circuits on reconfigurable hardware, particularly field programmable gate arrays (FPGAs). However, the ability of EHW to generate larger and more complex circuits has been limited due to the use of direct encodings in which the chromosome directly represents the device's configuration.

A practical approach to solving this problem for specific application domains has been function-level evolution, involving the use of higher-level primitives or modules. Although this scales EHW to more complex problems, it comes at the price of higher gate counts and designer bias [26], as well as the loss of potential novelty in solutions, thus countering some of the original motivations for EHW.

Another approach is through decomposing the problem into components or subtasks which are evolved first and then combined. However, these approaches are limited to applications with straightforward decompositions, without interdependencies.

A separation between genotype (the chromosome) and phenotype (the generated circuit), and a way of generating the phenotype from the genotype (a growth process), is the approach taken by nature to evolve complex organisms, and has increasingly been seen as a means of scaling EHW to more complex problems without losing its ability to generate novelty.

Morphogenetic approaches have been successfully applied to generating tessellating patterns [1], neural networks [11, 4, 22], and adders and parity generators on a simple virtual FPGA model [7], but have had little success in generating circuits at the gate level on modern FPGAs [6, 9]. However, recent work by the authors [15] has shown that morphogenesis can be successfully applied to EHW at the gate level on a Xilinx Virtex FPGA.

Gate-level Morphogenetic EHW

In our morphogenetic EHW model, biological cells correspond to functionally independent logic elements, comprised of a LUT-register pair (LUTs are Boolean function

Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06)
0-7695-2614-4/06 $20.00 © 2006 IEEE

generators) and associated routing. There are four logic elements within a configurable logic block (CLB). Multiplexors (gates) are interpreted as proteins within a cell, with each protein representing that resource and its configuration. Inter-cellular signaling is implemented via shared routing lines, such that for each line connecting logic elements, both cells have a signalling protein that corresponds to the configuration of the multiplexor (mux) in the other logic element.

Genes are encoded on a variable length base-4 chromosome and identified by signature sequences (specifying the gene promoter, coding region and regulatory bind sites). The promoter, to which polymerase binds for initiating transcription of the gene's coding region, is a fixed 4-character sequence upstream of the gene coding region. Gene transcription is regulated by the binding of proteins to bind sites adjacent to the promoter: those upstream enhance polymerase binding, while those downstream repress polymerase binding. The gene coding region is delimited by start and stop codons (triplets) and encodes one or more proteins with a codon-based genetic code.

Each cell receives a copy of the (decoded) chromosome, and implements a transcription level gene expression model, whereby gene transcription is initiated by the binding of proteins (the configuration state of the muxes) in the cell, and in turn generate proteins (reconfiguring the muxes in this logic element) which are able to bind to gene bind sites to further regulate gene transcription. Through the interaction of genes with the FPGA configuration and inter-cell signalling, a gate-level morphogenesis process emerges that is able to generate circuits within the constraints of an FPGA. Details of the morphogenetic EHW system can be found in [15], along with the results of exploratory experiments.

Outline of Paper

In this paper we extend this work, to clearly show the ability of the morphogenetic approach to scale to both increases in circuit size and complexity. This is presented in Section 2.

Then we look at some issues that were encountered when evolving one bit full adders. The result of this work was the development of a means of identifying whether or not a given experiment is likely to succeed, based on measures of problem difficulty and feedback from the fitness function. This is presented in Section 3.

Furthermore, while decompositional, and arguably function-level, approaches to EHW are limited in their applicability to circuit adaptation and repair, morphogenetic approaches seem ideally suited to solving these problems. Circuit adaptation and repair with EHW are covered in detail in Section 4, and some exploratory work with our morphogenetic EHW is presented, before concluding with a discussion of future work in Section 5.

2 Scaling with Morphogenesis

To test the performance and scalability of the morphogenetic and traditional EHW approaches, signal routing circuits were evolved with severely constrained routing that disallowed simple connection rules (see [15] for details). These experiments allow a comparison of the two EHW approaches relative to increases in circuit size, as measured by the size of the CLB matrix evolved, and circuit complexity, as measured by the number of signals routed across the matrix.

All experiments in this section use the same evolutionary and morphogenetic parameters. A steady state genetic algorithm using tournament selection without replacement was used with a population size of 100. The crossover rate was set at 80%, mutation at 2%, inversion at 5%, and for the variable length chromosomes used with the morphogenetic approach, a base insert/delete rate of 0.1% was used with 50-50 chance of insertion or deletion. Note that while the traditional EHW approach uses a direct encoding on a fixed-length binary chromosome, the morphogenetic approach encodes a gene expression model on a variable length base-4 chromosome, with unbounded chromosome growth limited through the use of a soft gene ceiling of 8 genes (over which chromosomes with fitnesses less than the maximum attained so far are penalised).

For the morphogenetic approach, a transcription rate of 4 bases per growth step is used, with a polymerase to gene ratio of 30% and a 20% probability of polymerase binding to the promoter of an activated gene (at any given growth step). Growth is done for a minimum of 30 iterations, with fitness evaluated at each step. Growth is continued if the maximum phenotype fitness for this genotype increased in the last 15 iterations, or if phenotype fitness is increasing. The genotype's fitness is given by the maximum phenotype fitness achieved during growth.

2.1 Experiments in Scaling Circuit Size

To determine the performance of both morphogenetic and direct encoding approaches to increases in circuit size, experiments were conducted on four CLB matrix sizes, these being 5x5 (containing 100 cells), 9x9 (324 cells), 13x13 (676 cells), and 17x17 (1156 cells). This involved an increase of 11.56 times the number of logic elements from the smallest to largest evolved CLB regions.

In all cases the circuit function was fixed to routing a single signal entering the center of the West side of the CLB matrix to exit the center of the East side, with signal input and output points remaining the same (single lines $West11$ and $East11$ respectively) to ensure that the only difference

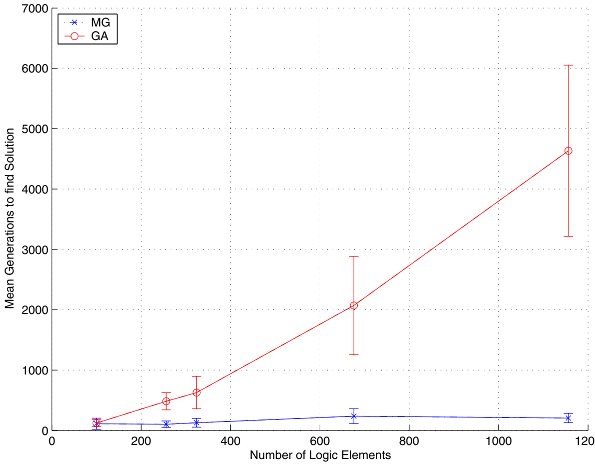

Figure 1. Mean & S.D. Generations Required

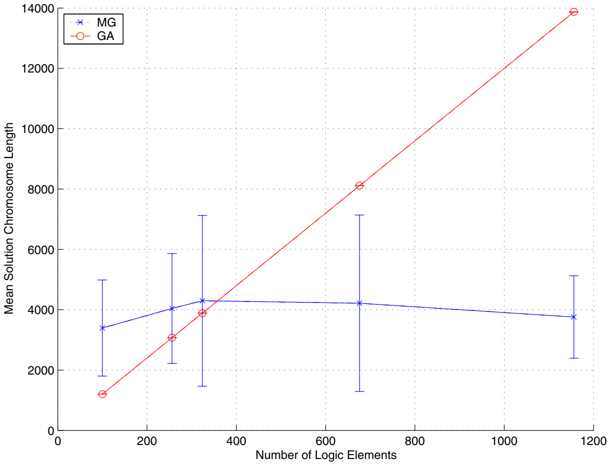

Figure 2. Mean & S.D. Chromosome Length

in problem difficulty lies in the scaling of CLB matrix size. A reduced set of routing resources was used that disallows a direct West to East connection across the CLB matrix.

For each set of experiments 10 runs were conducted for both morphogenetic (abbreviated to MG in figures and tables) and direct encoding (abbreviated to GA) approaches. Figures 1 and 2 show the mean and standard deviation (S.D.) of each of these sets of experiments, in terms of, respectively, generations required and chromosome length versus the number of logic elements (the number of CLBs times four). For illustrative purposes, these figures also include the results from the 8x8 single signal routing experiment (which lies between the 5x5 and 9x9 experiments, with 256 cells, but has different IO points) from Section 2.2.

Note, however, that for the 17x17 CLB problem, one of the direct encoding EHW runs didn't complete in 10,000 generations, having reached a maximum fitness of around 82.35% at generation 5636. This run was not included in the mean or standard deviation of the direct encoding approach.

Comparison of Results

From Fig. 1 it appears that for the morphogenetic approach, the number of generations required to find a solution is not dependent on the size of the evolved region, in contrast to the direct encoding approach, which is strongly affected.

For the smallest experiment, the direct encoding approach took a mean of 121 generations (S.D. of 58.9) to find a 100% solution, while the morphogenetic approach took a mean of 109.4 generations (S.D. of 93.9). Thus the morphogenetic approach generally showed superior performance, in terms of generations required in this experiment set, and the performance of the MG approach over the GA became increasingly pronounced with increases in the evolved CLB area.

For the largest experiment, the direct encoding approach (discounting the failed run mentioned above) required on average 4635.1 generations to find the solution (S.D. of 1417.7), a 38 fold increase for the 11.56 increase in logic elements. The morphogenetic approach, however, required only 204.9 generations on average (S.D. of 75.8) to find a 100% solution, a mere 87% increase in generations required over the 5x5 problem, and furthermore the mean of the 17x17 set almost lies within 1 SD of the mean of the 5x5.

Another measure of scalability is the length of chromosome required to encode the solution. While the chromosome length of a direct encoding approach to EHW has a direct correspondence to the size of the evolvable region, it is obvious from Fig. 2 that this is not the case for a morphogenetic approach. Whereas the direct encoding approach had a chromosome length that increased from 1200 bits for the smallest problem to 13,872 bits for the largest, the morphogenetic approach varied from around 3500 bases for the 5x5 and 17x17 problems to around 4250 bases for the 9x9 and 13x13 problems.

Comparing the number of genes required to solve the problem, for the morphogenetic approach, also showed no correspondence with circuit size, with the 5x5 and 17x17 CLB problems both requiring on average 7.5 genes, while the 9x9 and 13x13 problems required 8.8 and 8.4 genes, respectively. In all cases the difference between experiment set means is less than the standard deviation within sets.

Likewise, for the number of growth steps required to construct the solution, there was no direct correspondence between circuit size and growth steps required. The 5x5 problem required on average 32.6 growth steps, the 8x8 problem (from the next section) required slightly less at 32.4, the 9x9 problem required 36 steps and the 13x13 re-

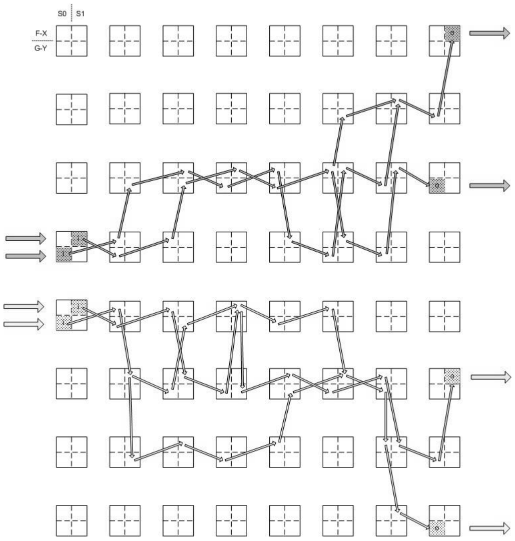

Figure 3. E.g. 4-Signal Routing Solution

quired 38.7, but then the 17x17 problem required less than both the 9x9 and 13x13 at an average of 35.9. As with the number of genes required, the difference between experiment set means was less than the standard deviation within sets.

2.2 Experiments in Scaling Circuit Complexity

In this set of experiments the scalability of morphogenetic and direct encoding approaches to EHW is evaluated and compared in terms of circuit complexity, by fixing the CLB matrix size to 8x8 CLBs (containing 256 cells) while increasing the number of signals routed across the matrix from 1 to 4, and hence also the number of IO points from 2 to 8, requiring evolution to learn not just how to connect horizontally across the matrix, but also how to spread vertically from the middle outwards.

In the case of single signal routing, the entry point is LUT input single $West14$ in the lower of the two middle row CLBs on the West edge of the CLB matrix, while the output point is single $East3$ output from the upper of the two middle row CLBs on the East edge of the CLB matrix.

For the multi-signal routing experiments, two sets of two inputs (single lines $West11$ and $West14$) enter the center two CLBs of the West edge of the CLB matrix, while outputs are spread evenly across the East edge of the CLB matrix, with exit lines being $East3$ and $East11$ (two each). The layout of the multi-signal routing is illustrated with a handcrafted shortest-path solution in Figure 3.

Here, as with the previous experiments, the aim is to route signals, possibly inverted, from inputs to outputs, but in this case it was only necessary that each input and output is fully connected. In other words, the relationship between the different inputs and outputs is not important, it is only necessary that all inputs are connected and one or more of these drive the outputs.

Multi Signal Experiment Results

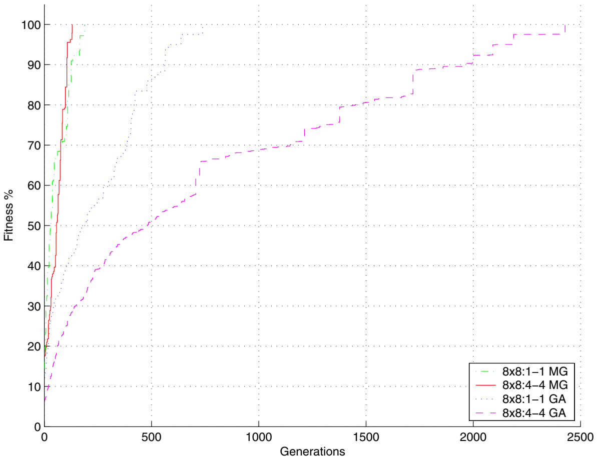

Figure 4. Mean Max Fitness for 8x8 Routing

Again, for each set of experiments 10 runs were conducted for both morphogenetic and direct encoding approaches. Figure 4 shows the mean maximum for all runs at each generation, for all four sets of experiments (1-1 and 4-4 indicate the number of signals, while MG and GA denote the morphogenetic and direct encoding approaches, respectively). From this it is obvious that the morphogenetic approach outperformed the direct encoding approach for both runs at all stages of evolution (generations). Both single and multi signal runs of the morphogenetic approach outperformed even the single signal runs of the direct encoding approach. Furthermore, there was little difference in performance between the single and multi signal runs for the morphogenetic approach at all stages of evolution (101.9 generations mean with S.D. of 53.8 versus 89.7 generations with S.D. of 23.1, for the single and multi signal routing sets respectively), while in comparison, the multi signal runs of the direct encoding approach took close to four times as long to solve (the single signal set had a mean of 482.5 generations and S.D. of 142.7 versus 1616.7 generations and S.D. of 598.7 for the multi signal routing set).

In terms of chromosome length, the direct encoding approach required 3072 bits for both sets, while the morphogenetic approach varied between a mean of 4043.3 bases (S.D. of 1821.9) for the single signal set and 3841.5 bases (S.D. of 1463.7) for the multi signal set. The mean gene count for solutions in both sets of experiments was 7.7 (with S.D. of 2.21 and 3.34 for the single and multi signal experiments, respectively), showing that increasing the complexity incurred no recognisable cost in terms of genes.

The growth count for solutions, however, increased modestly from a mean of 32.4 (S.D. of 6.24) for the single signal experiments to a mean of 37.9 (S.D. of 8.28).

2.3 Summarising Scalability Results

The results of these experiments clearly shows that the morphogenetic approach is able to scale to increases in both circuit size and complexity, with little effect on generations required, chromosome length, and gene count. The number of growth steps required also appeared not to be overly dependent on circuit size. Also, the morphogenetic approach, on average, outperformed the direct encoding for all problems, which became increasingly obvious with increases in the size of the evolved CLB region, or circuit complexity.

Unlike the morphogenetic approach, the direct encoding approach, increasingly struggles as the circuit size is increased, and eventually at the largest problem failed to successfully complete one of its experiment runs. It is likely that this is an indication that at even larger circuit sizes it would begin to fail more often.

3 EHW Problem Solvability

The previous section demonstrated the capabilities of a morphogenetic approach to scaling circuit size and complexity, within the constraints of a limited number of FPGA resources. Experiments were then undertaken to investigate the evolution of more substantial circuits, for which a one bit full adder on a 2x2 CLB matrix was chosen as the target circuit.

As with the previous experiments, each cell correspond to a logic element, however here, the evolutionary and morphogenetic processes are able to utilise all four of the LUT's inputs, the full 16 bits of the LUT's functionality, the unregistered logic element output, two dedicated CLB out bus lines (from the 8 available) and six single lines per out bus line, with connections available to all of the neighbouring CLBs.

The evolvable region is setup so that the x and y input signals are provided to single lines that can be fed to the LUT inputs in the South West CLB, while the cin signals are fed towards the North West CLB's LUT inputs, and the sum and $cout$ output signals are sampled from one logic element each on the two East CLBs. This layout is shown in Figure 5.

Several lines are available to the circuit inputs, while (as with the previous experiments) the outputs require signals to be routed to specific lines. Signals are routed directly from IO blocks on the perimeter of the FPGA to the CLBs adjacent to the evolvable region. This approach is an artefact of the original design aims, which were to allow the evolution of asynchronous circuits for robot control.

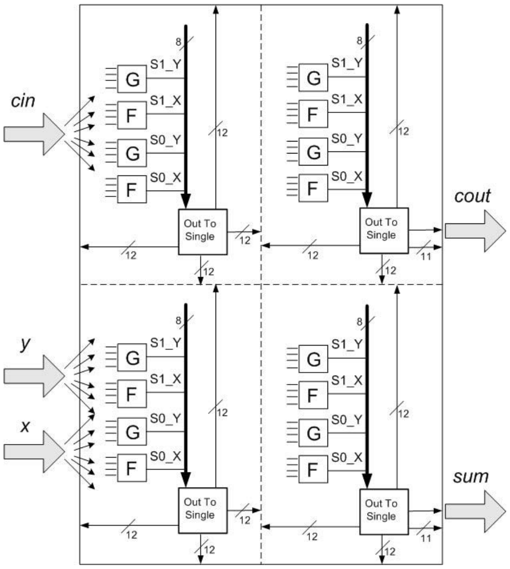

Figure 5. Layout for Adder Experiments

The results of experiments showed that neither morphogenetic or traditional EHW approaches were able to solve this problem (within the small number of runs performed), without constraining the resources available. Further experiments were undertaken with portions of the circuit fixed to components of a known solution. By testing with LUT's only evolution (fixed routing), LUTs and LUT inputs only, and routing muxes only (fixed LUT) it was found that the problem lay in the evolution of the routing between LUTs, while LUT functions are able to be evolved quite rapidly.

Table 1 provides the results, in percentage of successful runs, for all experiments (signal routing experiments are denoted by CLB rows x columns and number of input-output signals, while adder experiments denote the resources manipulated – with LUTin being LUTs plus input muxes) for both morphogenetic (MG) and traditional (GA) approaches. The experiment state space sizes (S), probability of a random FPGA configuration being a solution (P) and the measure of *effective* feedback provided to the evolutionary and morphogenetic processes (E) are also given in base-2 logs. More detailed results are provided in [16].

From the calculated values provided in Table 1, it is apparent that the measure $|P|/E$ provides a good indicator of problem solvability (especially for the morphogenetic approach, while the traditional approach is also affected by the size of the state space). In other words, EHW problems need to take into account both the difficulty of the problem (as measured by the probability of a random FPGA configuration being a solution) and the amount of information provided by the fitness function to the morphogenetic and

Table 1. Experiment Calculation Summaries

Experiment	S	P	E	$\vert P\vert/E$	MG%	GA%
5x5:1-1	999	-36.9	144	0.26	100	100
9x9:1-1	3237	-67.2	590	0.11	100	100
13x13:1-1	6754	-98.5	1333	0.07	100	100
17x17:1-1	11550	-129.8	2373	0.05	100	90
8x8:1-1	2558	-58.2	450	0.13	100	100
8x8:4-4	2558	-208.6	464	0.45	100	100
adder LUT	256	-25.4	21	1.21	100	100
adder LUTin	496	-73.3	21	3.49	50	0
adder Mux	464	-127.0	21	6.05	0	0
adder All	720	-144.5	21	6.88	0	0

evolutionary processes for guiding the system to a solution.

From the experiments conducted to date, it appears that for the problem to be consistently solvable, the value of $|P|/E$ needs to be less than or equal to one, with less than one providing more feedback than necessary, and as this value increases the experiment's likelihood of being solved decreases. From the limited experiments conducted, values of $|P|/E$ just below 3 had a 50% chance of being solved, those around 3.5 had around a 25% chance of being solved, and those around 4.5 or more, were unlikely to be solved (in a limited number of runs).

As can be seen, the routing experiments, while they are harder to solve, according to P, remain solvable due to the large amount of effective feedback to guide evolution. while, on the other hand, the unfixed adder problem and the adder problem in which multiplexor settings only need to be evolved both lack sufficient feedback to guide evolution to a full solution. It can also be seen that evolving the adder LUTs was successful due to the larger proportional coverage of feedback to the problem.

The motivation for calculating $|P|/E$ is that $|P|$ can be seen as a way of reducing the size of the state space (S), by removing the redundancies (for example logic elements that don't contribute towards the solution and are hence able to be configured in any manner without effecting the solution) that dominate the state space of solutions, to a size where there is only a single solution. Therefore, $|P|/E$ shows the amount of effective fitness feedback coverage of this smaller state space, which obviously maps the entire non-redundant space for $|P|/E \leq 1$, while increasingly leaving gaps for $|P|/E > 1$.

This approach makes the assumption that all the solutions in the original state space are uniformly distributed, however, given that $|log(P)|$ is derived largely through the redundancy of unused logic elements, this seems a reasonable assumption (i.e. the redundancy that creates the majority of the alternate solutions doesn't clump them all together in the same area of state space).

Importance of Feedback

This identifies a significant issue in EHW: while problem difficulty and state space size may influence the number of generations required to solve a given problem, it is the amount of useful fitness feedback information supplied to the EHW system that determines whether or not a given problem can be solved by evolutionary and morphogenetic methods.

4 EHW Adaptation and Repair

While most approaches to EHW usually evolve a static circuit design, optimised for certain situations but unable to change behaviour once evolved, EHW has the capacity, unlike conventional hardware, to reconfigure its hardware structure dynamically and autonomously.

This ability allows EHW to achieve adaptation and fault tolerance in several manners. The most obvious of these is that when a given circuit ceases to function as required, due to unexpected environmental effects for example, evolution can be continued until a new circuit is found that produces the desired behaviour under the current circumstances. In actual fact, for circuits that cannot be fully specified a priori, but where the desired behaviour is known, evolutionary approaches may be the only viable option. In conjunction with this, circuits can be evolved specifically to be adaptive and resilient to faults. Lastly, evolution is able to incorporate faulty components into the circuit, according to their actual behaviour, as opposed to their expected behaviour.

An example of where adaptation to changes in the environment might be useful in EHW was given in [14]. An EHW controller for a legged robot may be able to change its gait if a leg joint seized, right itself if placed upside-down, or find a way of walking over or around a previously unencountered obstacle.

Other examples of applying EHW to adapting robot controllers to changes in the environment, includes work by the Electrotechnical Laboratory in Japan, where it was used for learning to navigate an autonomous wheeled robot for tracking a coloured ball while avoiding obstacles and adapting to sensor failures [10]; and the Complete Hardware Evolution (CHE) approach of [8] used to develop their GERC robot controller, in which evolution can be restarted when the current design is no longer able to perform satisfactorily.

Another area where hardware adaptation is attractive is in space exploration (for example adaptable hardware for space instrument controllers [3]), and especially for repairing faulty circuits (see [27] for an example aimed at space missions). An early example of work in adaptive EHW for self-repairing circuits includes the FIPSOC (Field Programmable System on a Chip) architecture [19].

In EHW approaches robustness of evolved circuits to component faults and failures has been shown to be able to be achieved without explicitly needing to be pre-designed, through the evolutionary process [24, 25]. Extensive work has been done in applying EHW to problems in fault tolerance (for example [13, 18, 21, 23, 5, 17]).

There are also some non-EHW approaches to circuit repair, such as Embryonics [20] and Immunotronics [2], which are able to modify their configuration in response to failures in the device's cells.

Issues with Online Adaptation in EHW

While EHW is able to adapt to changes in its environment by restarting evolution, it is not certain that online adaptation is necessary for EHW in all cases. Designs can be evolved explicitly for their ability to generalise to unforseen circumstances, as is achieved in neural network approaches. This was shown to be the case in [12] where an evolved controller was able to generalise to situations that it hadn't encountered in evolution.

A comparison between the fault tolerance of a population-evolution approach, as done in online adaptive EHW (where a population is kept and if a fault occurs another member may be used) and a fitness-evolution approach (where fault tolerance is tested for during evolution resulting in an individual that has this property) was made in [13]. It was found that the fitness-based approach has the advantage of obtaining a single circuit resilient to multiple faults, however the population approach is able to find circuits in the population with better performance and requires 20% to 50% less computation than the fitness-based approach. However, although the population approach offers a self-repair mechanism, it must be done off-line in 50-80% of the time needed to obtain a solution from scratch.

Morphogenesis for Adaptation and Repair

In the course of the adder experiments presented briefly in Section 3, and more thoroughly in [16], we found that in spite of the insufficient amount of feedback provided to the morphogenetic EHW system when evolving an adder without constraints, morphogenesis was readily able to find an adder circuit when started at a point where the LUT functions were unconfigured but routing muxes were preconfigured on the FPGA (but with both able to be modified by the morphogenetic process).

This seemed to indicate that a morphogenesis process could be used as a dynamic circuit repair mechanism. To quantify this, morphogenesis was able to complete the circuit configuration in as few as 15 growth steps with 6 genes (or 19 steps with 3 genes) when started at a Hamming distance of 64 bits from the desired solution's configuration. Although this result is very preliminary, it looks to be a promising avenue for continued research in the near future.

Morphogenesis offers real potential for providing online autonomous and robust circuit adaptation and self-repair. By evolving genotypes for both performance and adaptivity, when a circuit ceases to fulfil its task morphogenesis can be continued until a new circuit forms that performs as required. Alternatively, if the morphogenesis process is kept running at all times, it could provide a dynamic self-regulating mechanism that keeps the circuit functioning within operational bounds, akin to homeostasis in biological organisms. These approaches could be much faster than having to restart evolution to find a new circuit, as is the most common approach taken currently.

In a morphogenetic approach, evolution only needs to be restarted when environmental changes are beyond the adaptive capabilities of the individual. This could provide both rapid adaptation through morphogenesis, and long term and extreme adaptation through evolution.

5 Conclusion

In this paper we have shown that a morphogenetic approach to EHW scales extremely well to increases in circuit size and complexity. Our morphogenetic EHW approach has proven to be largely unaffected by the size of the state space size, reflecting FPGA platform complexity and circuit size, in direct contrast to traditional direct encoding approaches to EHW. This offers great promise to EHW, as it provides scalability without having to compromise the advantages of gate-level evolution.

Furthermore, we have introduced a quantitative measure of problem difficulty, in terms of the probability of a random FPGA configuration being a solution, and a heuristic indicating whether an EHW problem is solvable or not according to its difficulty and the amount of effective information provided by the fitness function for guiding evolution. This highlights the importance of fitness feedback in EHW – without sufficient fitness feedback relative to the problem's difficulty, it is likely that evolutionary approaches to circuit generation will fail.

Lastly, we have provided a brief overview of research into the ability of EHW to adapt to changes in its environment, and to respond to and repair circuit faults. These abilities are particularly important as an enabling technology for exploration of harsh environments, and for future long term space missions. In particular, we have identified morphogenesis as an attractive approach to adaptation and self-repair for EHW. This approach appears to be a promising avenue for future research.

References

[1] P. Bentley and S. Kumar. Three ways to grow designs: A comparison of evolved embryogenies for a design prob-

lem. In *Proc. of the Genetic and Evolutionary Conference (GECCO '99)*, pages 35–43, 1999.

[2] D. Bradley and A. Tyrell. Immunotronics: Hardware fault tolerance inspired by the immune system. In *Proceedings of the Third International Conference on Evolvable Systems (ICES2000)*, Lecture Notes in Computer Science, 1801, pages 11–20. Springer-Verlag, 2000.

[3] A. G. Darrin, R. Conde, B. Chern, P. Luers, S. Jurczyk, and C. Mills. Adaptive instrument module: Space instrument controller "brain" through programmable logic devices. In *Proceedings of the Third NASA/DoD Workshop on Evolvable Hardware*, pages 256–260, 2001.

[4] P. Eggenberger. Cell interactions as a control tool of developmental processes for evolutionary robotics. In *Proceedings of SAB '96*, pages 440–448, 1996.

[5] M. Garvie and A. Thompson. Scrubbing away transients and jiggling around the permanent: Long survival of FPGA systems through evolutionary self-repair. In *Proc. 10th IEEE Intl. On-Line Testing Symposium*, pages 155–160. IEEE Computer Society, 2004.

[6] T. G. Gordon and P. J. Bentley. Towards development in evolvable hardware. In *Proceedings of the 2002 NASA/DoD Conference on Evolvable Hardware*, 2002.

[7] T. G. Gordon and P. J. Bentley. Development brings scalability to hardware evolution. In *Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware*, pages 272–279. IEEE Computer Society, 2005.

[8] P. C. Haddow and G. Tufte. An evolvable hardware FPGA for adaptive hardware. In *Proceedings of the 2000 Congress on Evolutionary Computation (CEC '00)*, pages 553–560, 2000.

[9] P. C. Haddow, G. Tufte, and P. van Remortel. Shrinking the genotype: L-systems for ehw? In *4th International Conference on Evolvable Systems (ICES01)*, Lecture Notes in Computer Science, pages 128–139. Springer, 2001.

[10] T. Higuchi, M. Iwata, D. Keymeulen, H. Sakanashi, M. Murakawa, I. Kajitani, E. Takahashi, K. Toda, M. Salami, N. Kajihara, and N. Otsu. Real-world applications of analog and digital evolvable hardware. *IEEE Transactions on Evolutionary Computation*, 3(3):220–235, 1999.

[11] N. Jakobi. Harnessing morphogenesis. Technical Report CSRP 423, School of Cognitive and Computer Science, University of Sussex, 1995.

[12] D. Keymeulen, K. Konaka, M. Iwata, Y. Kuniyoshi, and T. Higuchi. Robot learning using gate-level evolvable hardware. In *Proceedings of the Sixth European Workshop on Learning Robots, Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1998.

[13] D. Keymeulen, R. S. Zebulem, Y. Jin, and A. Stoica. Fault-tolerant evolvable hardware using field-programmable transistor arrays. *IEEE Transactions on Reliability*, 49(3):305–316, 2000.

[14] P. Layzell. *Hardware Evolution: On the Nature of Artificially Evolved Electronic Circuits*. PhD thesis, School of Cognitive and Computing Science, University of Sussex, Sussex, 2001.

[15] J. Lee and J. Sitte. A gate-level model for morphogenetic evolvable hardware. In *Proc. of the 2004 IEEE International Conference on Field-Programmable Technology (FPT'04)*, pages 113–119, 2004.

[16] J. Lee and J. Sitte. Issues in the scalability of gate-level morphogenetic evolvable hardware. In *Recent Advances in Artificial Life*, Advances in Natural Computation, Vol. 3, pages 145–158. World Scientific, 2005.

[17] H. Liu, J. Miller, and A. M. Tyrrell. Intrinsic evolvable hardware implementation of a robust biological development model for digital systems. In *Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware*, pages 87–92. IEEE Computer Society, 2005.

[18] J. F. Miller and M. Hartmann. Untidy evolution: Evolving messy gates for fault tolerance. In *Proceedings of the Fourth International Conference on Evolvable Systems (ICES2001), Lecture Notes in Computer Science*, volume 2210, pages 100–111. Springer-Verlag, 2001.

[19] J. Moreno, J. Madrenas, J. Cabestany, E. Canto, R. Kielbik, J. Faura, and J. Insenser. Realization of self-repairing and evolvable hardware structures by means of implicit self-configuration. In *Proceedings of the First NASA/DoD Workshop on Evolvable Hardware*. IEEE Computer Society, 1999.

[20] C. Ortega-Sánchez and A. Tyrell. Fault-tolerant systems: The way biology does it! In *Proccedings of Euromicro 97*, pages 146–151. IEEE Computer Society Press, 1997.

[21] L. Prodan, G. Tempesti, D. Mange, and A. Stauffer. Embryonics: Artifical cells driven by artificial DNA. In *Proceedings of the Fourth International Conference on Evolvable Systems (ICES2001), Lecture Notes in Computer Science*, volume 2210, pages 100–111. Springer-Verlag, 2001.

[22] D. Roggen, D. Floreano, and C. Mattiussi. A morphogenetic evolutionary system: Phylogenesis of the POEtic circuit. In *Proc. of the 5th International Conference on Evolvable Systems: From Biology to Hardware ICES 2003, LNCS 2606*, pages 153–164. Springer, 2003.

[23] A. Stoica, D. Keymeulen, T. Arslan, V. Duong, R. Zebulum, I. Ferguson, and X. Guo. Self-recovery experiments in extreme environments using a field programmable transistor array. In *Proceedings of the 2004 IEEE International Conference on Field-Programmable Technology (FPT'04)*, pages 9–15, 2004.

[24] A. Thompson, I. Harvey, and P. Husbands. Unconstrained evolution and hard consequences. In *Towards Evolvable Hardware: The Evolutionary Engineering Approach, Lecture Notes in Computer Science*, volume 1062, pages 136–165. Springer-Verlag, 1996.

[25] A. Tyrrell, G. Hollingworth, and S. Smith. Evolutionary strategies and intrinsic fault tolerance. In *Proceedings of the Third NASA/DoD Workshop on Evolvable Hardware*, pages 98–106, 2001.

[26] V. K. Vassilev and J. F. Miller. Scalability proplems of digital circuit evolution: Evolvability and efficient designs. In *Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware*. IEEE Computer Society, 2000.

[27] S. Vigander. Evolutionary fault repair of electronics. Master's thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, 2001.