

An Evolutionary Approach to Runtime Variability Mapping and Mitigation on a Multi-Reconfigurable Architecture

Simon J. Bale, Pedro B. Campos, Martin A. Trefzer, James A. Walker* and Andy M. Tyrrell
Intelligent Systems and Nano Science Research Group, Department of Electronics, University of York, UK
simon.bale@york.ac.uk

Abstract—Intrinsic device variability has become a significant problem in deep sub-micron technology nodes. The stochastic variations in device performance, which are a result of structural irregularities at the atomic scale, can impact both the yield and reliability of a circuit design. In this paper we describe a novel multi-reconfigurable FPGA architecture, the programmable analogue and digital array (PAnDA), which can tackle this problem by allowing post-fabrication reconfiguration of the effective transistor gate widths in a circuit. We demonstrate the advantages of this architecture by creating a frequency variability map of the array using ring oscillators in order to ascertain the location of any frequency outliers. We then show that it is possible, using an evolutionary algorithm, to select alternative transistor configurations which minimise the difference in frequency between one of these outliers and the chips median frequency of operation. Such methods can be used to increase system performance and reliability by presenting an array with more uniform performance characteristics.

I. INTRODUCTION

The continuous scaling of transistors has significantly increased the function density and performance of modern digital systems and this has resulted in a reduction in the cost per logic function in an integrated circuit. Scaling the physical dimensions of a transistor can reduce the propagation delay, power consumption and area of the device but at the cost of increased intrinsic variability which is a result of structural irregularities at the atomic scale [1] [2]. As transistor scaling has continued beyond 14 nm, the performance and reliability limitations introduced by these stochastic variations have become difficult to ignore.

In addition to this challenge, the increasing number of transistors in electronic systems may result in a proportional increase in device failure rates. Transistor ageing further contributes to the increasingly small margins seen by a designer, and therefore device reliability is ultimately affected by these low-level effects. In most ASICs a significant performance degradation or the failure of a single transistor can therefore potentially result in the irreversible failure of the entire device. Without specific recovery mechanisms incorporated into the device, a single point-of-failure can render an entire circuit unusable. Designing high speed, high yield digital systems

*J.A. Walker is now with University of Hull, Hull, UK. Email: j.a.walker@hull.ac.uk

with low supply voltages and low power dissipation in ultra-deep sub-micron CMOS technology (<100 nm feature size) poses a critical challenge to the future of CMOS design.

Reconfigurable devices, however, have the potential to mitigate some of the challenges which are a result of device scaling and increased transistor counts. The reconfiguration process can be used to select a more optimal placement for a circuit or circumvent faulty devices and subsequently recover functionality. Some devices have been developed which take reconfiguration beyond functionality and provide mechanisms for changing the geometry of individual transistors [3].

In this paper we describe a novel multi-reconfigurable FPGA architecture, the programmable analogue and digital array (PAnDA) [4], which allows variability-aware design, rapid prototyping and post-fabrication optimisation of a digital system. The PAnDA architecture includes mechanisms to overcome the effects of intrinsic device variability by allowing post-fabrication reconfiguration of the effective transistor gate widths.

The performance of a circuit which is mapped onto PAnDA is dependant on both the size of the transistors used as well as any local variations in the performance characteristics of the devices. A set of 15 stage ring oscillators are mapped onto the PAnDA fabric and used to generate a frequency variability map and locate any frequency outliers. We then show that it is possible to find alternative transistor configurations which minimise the difference in oscillation frequency between a frequency outlier and the chips median frequency of operation. Finding the optimal configurations can be a challenging due to the large search space — particularly in cases where multiple oscillators are optimised simultaneously — therefore a Genetic Algorithm is implemented and used to find the required bit-stream configurations.

This paper is ordered as follows: Section II provides a detailed description of the PAnDA architecture, section III describes the variability mapping procedure, section IV describes the implementation of the genetic algorithm used to locate the optimal bit-stream configurations and applies this algorithm to the hardware. Finally, section V provides a summary and conclusions.

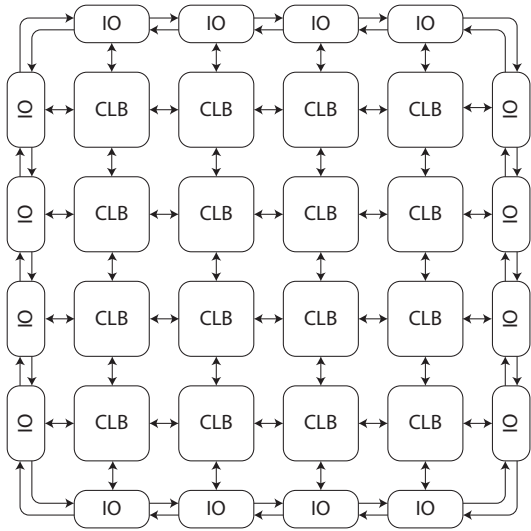


Fig. 1. A top level view of the PANDA architecture is shown. Two test chips with array sizes of 4×4 and 20×20 CLBs have been fabricated. The bi-directional arrows signify programmable routing resources.

II. ARCHITECTURE DESCRIPTION

The Programmable Analogue and Digital Array is a multi-reconfigurable architecture [4] which consists of an array of configurable logic blocks (CLBs) interconnected using a programmable routing structure (Fig. 1). The routing architecture consists of a switch matrix and cross-bar network. Additional input/output (IO) blocks surround the CLB array and allow buffering of external signals.

A novel feature of PANDA is the ability to access reconfiguration facilities at multiples levels of abstraction. The first level of abstraction is equivalent to that of a conventional FPGA where logic functions can be mapped onto configurable logic blocks. At the lowest level, configuration of the effective transistor gate widths is possible which enables a designer to modify aspects of the analogue circuitry underlying the digital function level. This allows the digital circuits implemented to be more closely optimised for the characteristics of the silicon substrate on which their transistors are fabricated. We have previously demonstrated, in simulation, that this technique can be used to recover and improve circuit performance that is degraded as a result of stochastic variability [4] [5] [6].

These additional levels of reconfiguration make it possible to apply novel optimisation and design techniques which are not possible using a conventional ASIC or FPGA architecture. For example it is possible to modify the power/delay ratio of a design post fabrication and potentially at runtime. Further, the performance variations caused by the intrinsic device variability, found in sub-micron technologies can be reduced by selecting more optimal devices. Increased fault tolerance is also possible by exploiting the symmetries of the architecture which allow a ‘fault-blind’ repair capability by considering functionally equivalent and structurally different alternative mappings [7].

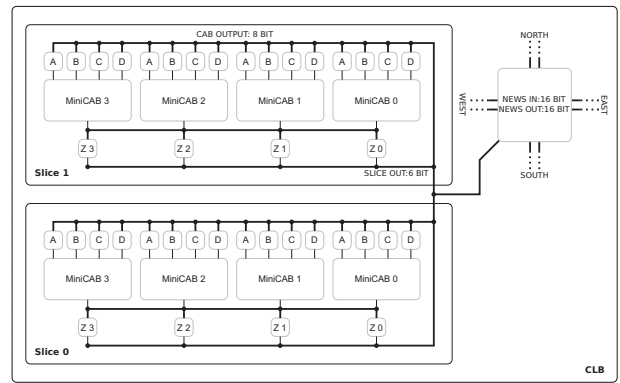


Fig. 2. A PANDA configurable logic block (CLB) is shown. Each CLB consists of two slices and a North-East-South-West (NEWS) routing block. Each slice consists of four configurable analogue blocks which when combined using the output mergers (Z0-Z3) can implement any 4-input logic function.

A. Configurable Logic Block

The CLB (Fig. 2) is a hierarchical structure consisting of two slices and a routing switch block. Each slice contains four configurable analogue blocks (CABs), input multiplexers and an output merger. A logic function is formed by combining CABs using the set of four output mergers. The routing switch block then provides external connectivity to the CLB array and allows up to four bi-directional buses to be simultaneously routed north, east, west and south. In addition any of the twenty four incoming signals can be routed to either of the two slices and the slice outputs can be routed to any of the twenty four outgoing signals. It is also possible to bypass the slice logic entirely and route directly between NEWS blocks.

Two test chips with array sizes of 4×4 and 20×20 CLBs have been fabricated using a 65 nm process. The CLB block is a fully tillable structure so it’s possible to make larger array sizes.

B. Configurable Analogue Block

Each CAB (Fig. 3) is constructed using eight NMOS and eight PMOS configurable transistors (CTs) arranged as two branches. This branch structure closely matches the circuit topology seen in CMOS logic design and an arrangement of four CABs combined with an output merger allows up to any four input logic function to be implemented.

C. Configurable Transistor

Fig. 4 shows a schematic of a configurable transistor (CT). Each CT consists of six transistors, arranged in a parallel configuration. The transistor gate widths range from 135 nm to 230 nm on slice 1 and on slice 0 are all the same minimum-size gate width of 135 nm. Configuration circuitry allows the input to the gates of each of these transistors to be enabled, disabled or clamped to a logic level. This allows any parallel arrangement of the six input transistors to be programmed resulting in an effective single transistor equivalent device with 64 possible gate width combinations ranging from 135 nm to 1045 nm (or 135 nm to 810 nm).

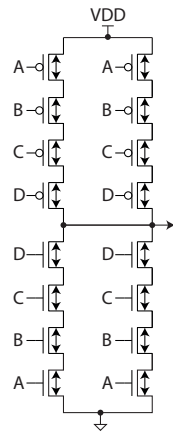


Fig. 3. A schematic of the configurable analogue block (CAB), showing the 8 NMOS and 8 PMOS CTs. One CAB can implement any 2 input logic function.

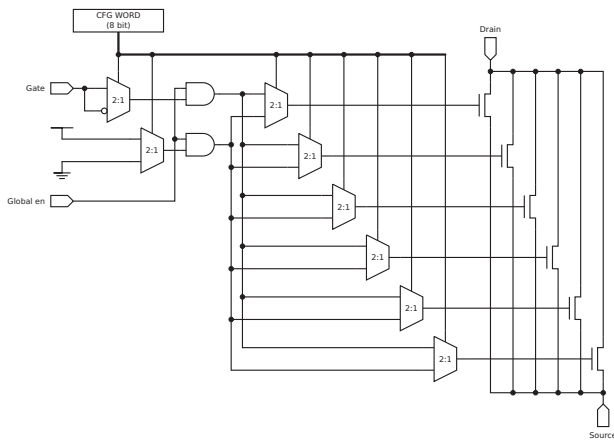


Fig. 4. A schematic showing a configurable transistor (CT). Each CT consists of six transistors, arranged in a parallel configuration, with gate widths range from 135 nm to 230 nm.

Each type of CT serves a different purpose. A greater number of width combinations and greater maximum width is possible using the CTs which contain the devices with differing widths (slice 1) and this allows the operating point of a circuit to be modified. The CTs which contain devices of the same size (slice 0) offer greater variability tolerance and allow the shape of the performance distribution to be modified.

III. VARIABILITY MAPPING

To validate the operation of the ASIC and to demonstrate the presence of variability in the reconfigurable architecture, a set of ring oscillators (ROs) were configured on the 20×20 CLB chip and their frequency of operation measured. Each 15 stage RO was constructed using a square, 2×2 , arrangement of CLBs where each CLB contained 4 (or 3) of the inverters configured on a single slice. The width of the NMOS and PMOS CTs in each inverter was fixed at 135 nm and 190 nm respectively. The in-phase feedback was created by connecting the output of the final inverter to the input of the first using

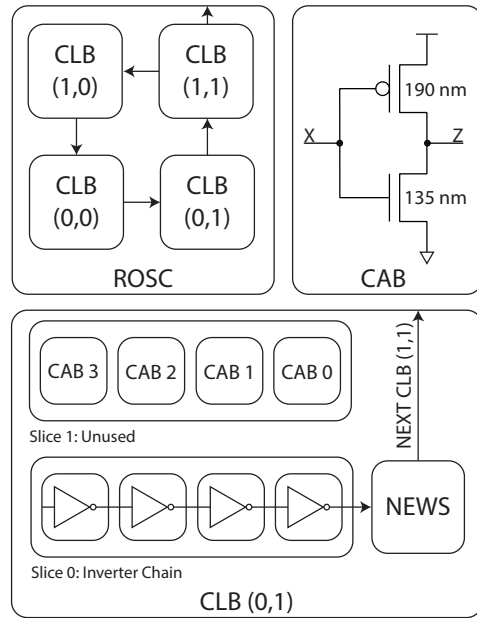


Fig. 5. Top Left: CLB arrangement for one of the 15 stage ring oscillators. Top Right: CAB configuration, showing the device sizes for one of the inverters. Bottom: A single CLB configuration showing a chain of four inverters, only one slice is used in each CLB.

the NEWS routing blocks. Fig. 5 shows the CLB arrangement for a single ring oscillator.

The output of the RO is then routed through a set of four edge-triggered D-Type flip-flops, each configured as a frequency divider, and then finally to a fixed output pin on the chip. Fig. 6 shows an illustration of the internal chip configuration for these experiments. The four red cells contain the 15 stage ring oscillator, the blue cells are used to route the RO output and only the NEWS blocks are configured in these CLBs. The green cells contain the 4 frequency dividers and route the output to the required pin on the chip. A total of ninety independent ring oscillators can be configured and measured using this arrangement.

Configuration and control of the PAnDA chip was achieved using an external FPGA. The embedded system on this chip allows the user to issue ASCII commands over an Ethernet connection to load configuration data, read/write the PAnDA IO pins and control the supply voltages. A simple C based XY router and bit-stream generator were developed in order to map and route the ring oscillators onto the hardware. Given the sensitivity of the ring oscillator centre frequency to external temperature fluctuations the frequency measurements were performed in a temperature controlled environment at 21 °C. The frequency of each RO was allowed to stabilise before a measurement was started and only one RO was active during each measurement.

The measurements were performed using a high speed oscilloscope to capture the time domain data which was then transferred to a PC for further signal processing. A simple software based spectrum analyser was developed in

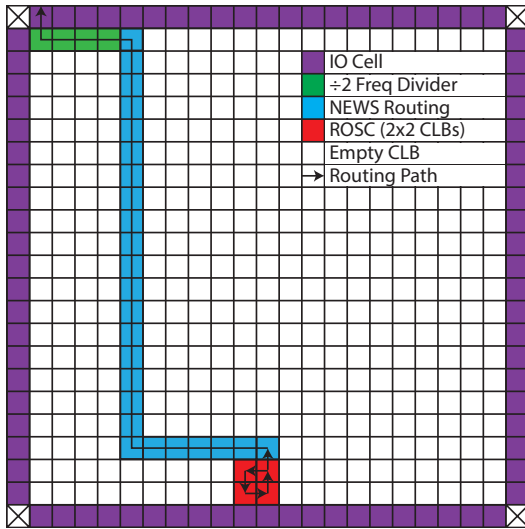


Fig. 6. Internal configuration of the PAnDA chip showing the routing arrangement for one ring oscillator.

MATLAB[®] and used to translate the time domain samples into the frequency domain. Due to the availability of anti-aliasing filters a sampling rate of 200 MSamples/Second was used and each data frame was 2×10^6 samples in length resulting in a resolution bandwidth of 100 Hz. Due to the jitter present in the output signal the centre frequency was calculated from an average of 1000 FFTs. The total measurement time for a single ring oscillator was approximately 11 minutes, inclusive of the initial time required for the centre frequency to stabilise. Mapping all ninety oscillators takes approximately 17 hours.

Fig. 7 & 8 show oscillator frequency/variability maps for two of the PAnDA chips. The median frequency of operation for chip 1 is 1.68535 MHz with a total range of 30.5 kHz. For chip 2 the median and range of the data are 1.64935 MHz and 26.9 kHz respectively. It should be noted that due to the presence of the frequency dividers the internal oscillator frequency is $16\times$ larger.

IV. VARIABILITY MITIGATION

In order to demonstrate the benefits of giving the designer access to underlying analogue properties of a circuit a frequency matching experiment has been performed. In this section we show that it is possible to find transistor configurations which minimise the difference in oscillation frequency between a chips median frequency of oscillation and one of the frequency outliers.

Finding the optimal configurations can be a challenging due to the large search space therefore a genetic algorithm has been used. Genetic algorithms (GAs) are a type of stochastic and adaptive search algorithm based on ideas taken from natural selection and genetics. Instead of starting from a single solution an evolutionary algorithm creates a random population of initial solutions spread throughout the search space. The algorithm then uses three genetic operators selection, recombination and mutation to evolve the fitness of

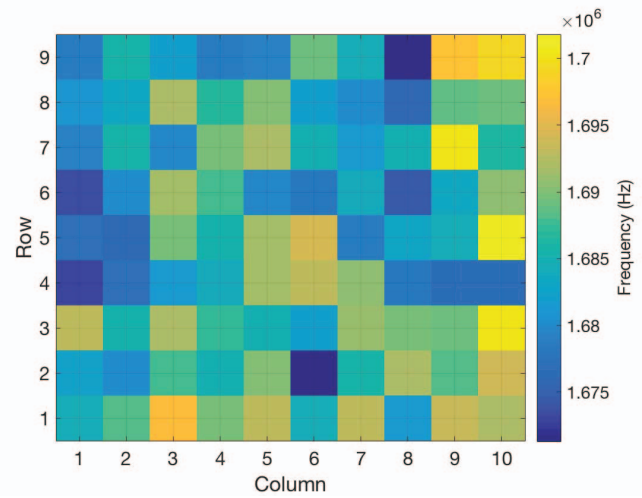


Fig. 7. An oscillator frequency map for PAnDA chip one. The median frequency of the oscillators, after frequency division, was 1.68535 MHz with a total range of 30.5 kHz.

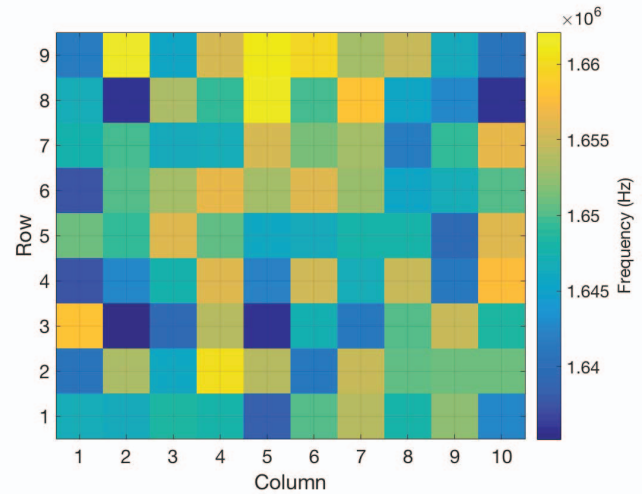


Fig. 8. An oscillator frequency map for PAnDA chip two. The median frequency of the oscillators, after frequency division, was 1.64935 MHz with a total range of 26.9 kHz.

the individuals within the population until the global optimum is reached. These operators are analogous to the process of evolution that occurs in the natural world. Genetic algorithms were established by John Holland [8] in 1975 and are well suited to solving problems with a high dimensionality search space.

A. Microbial Genetic Algorithm

In a traditional GA the optimisation loop operates on a generational basis. Specifically, the fitness of the entire population is evaluated and then the genetic operators are applied creating a complete new population which replaces the previous generation. In this work, we use a simplified, minimal GA originally proposed by Harvey [9]. This algorithm

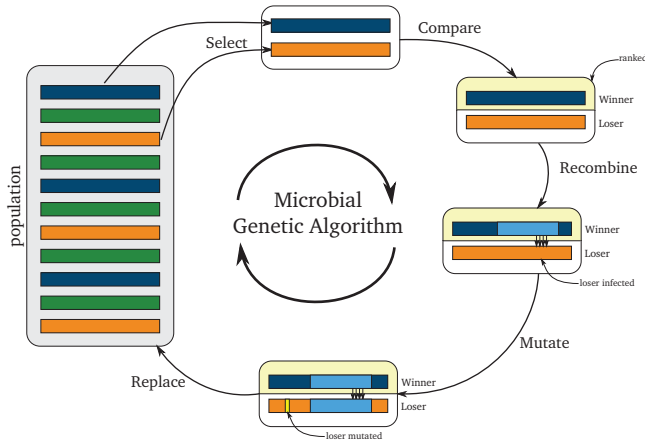


Fig. 9. A flowchart showing the operation of the Microbial Genetic Algorithm [9]

is an example of a steady-state GA which in its simplest form replaces just one individual in the population with a single new one during each iteration. In a steady-state GA the modified offspring individual immediately forms part of the mating pool and this may allow a shift towards a more optimal solution earlier in the optimisation process [10]. Given the long measurement time of each oscillator this is an advantage in our application.

The algorithm starts by generating an initial population of random individuals where each individual is represented as an array of integers (genes). Two individuals are then picked at random from this population and ranked according to their fitness to establish a winner and loser. Uniform recombination is then applied which replaces 50% of the genetic material in the losing individual with that of the winner. The purpose of this process is to attempt to guide the population towards a more optimal solution. To stop the population becoming genetically uniform a process of mutation is also used. This is implemented by randomly selecting one gene in the losing individual and replacing its value with a randomly generated value. The original loser is then replaced with this modified individual. The processes of selection, recombination and mutation are then applied iteratively until a stopping criterion is reached. A graphical illustration of the algorithm is shown in Fig. 9.

In this work a population size of 10, recombination rate of 0.5 and mutation rate of 1 gene are used. Each ring oscillator is represented using thirty integers in the range 0 to 5 where the value of each integer denotes which transistor in the CT should be enabled. Thirty integers are required as there are 15 N and 15 P transistors in each ring oscillator. The fitness is calculated by taking the magnitude of the difference between the median frequency of operation for the chip and the frequency of the oscillator to be optimised, as shown in equation 1.

$$fitness = |median(F) - f_{xy}| \quad (1)$$

where $median(F)$ is the median frequency of the whole set of oscillators and f_{xy} is the frequency of the oscillator in row

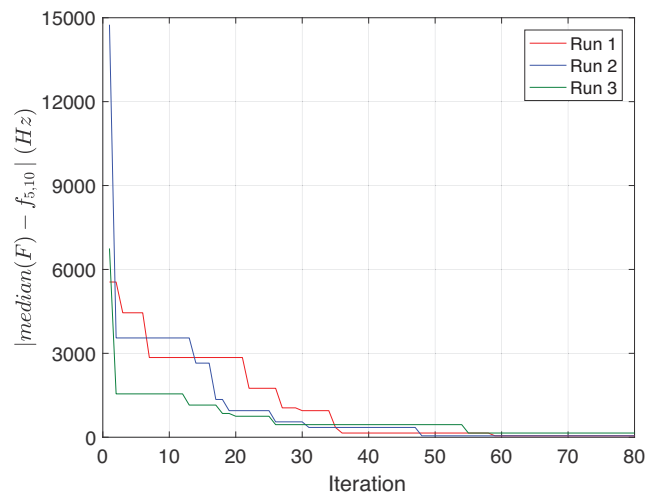


Fig. 10. A plot showing the absolute frequency shift of ring oscillator (5, 10) from the median oscillation frequency of PAnDA chip 1. Only the best individual in the population is plotted and three independent runs are shown.

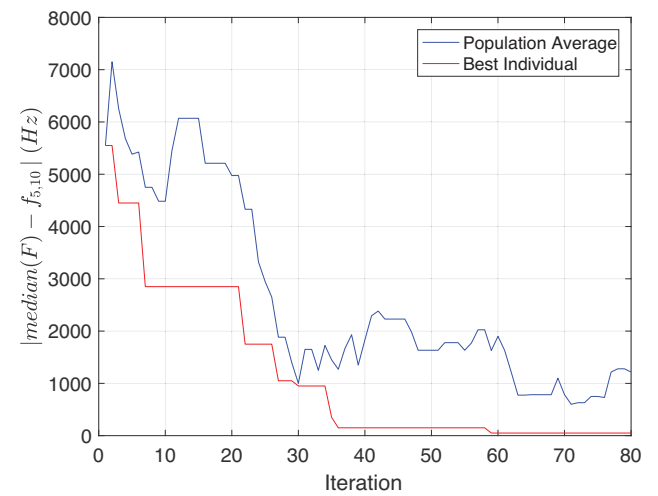


Fig. 11. A plot showing the absolute frequency shift for best individual in optimisation run 1 (lower) and the average absolute frequency shift for all individuals in the population during this run (upper).

column y . A lower value represents a more optimal fitness.

B. Experimental Results

The oscillator with the highest frequency relative to the chip median was selected for optimisation using this algorithm. With reference to Fig. 7 this oscillator is located at row 5 column 10 and has a nominal frequency of 1.7018 MHz and is 16.45 kHz from the median frequency of 1.68535 MHz. The chip configuration and experimental apparatus remained as described in III. The optimisation loop was left to run for 80 iterations and three independent runs were performed, each run took approximately 14 hours.

Figure 10 shows plots of the three independent optimisation runs. The frequency offset of the best individual in the population (lowest absolute distance from the median frequency of

the chip) is plotted. After 35 iterations the offset of all runs has reduced to a maximum of 450 Hz. This is a significant improvement relative to the initial offset of 16.45 kHz. After 80 iterations the maximum and minimum offsets across all runs are 150 Hz and 50 Hz respectively, although future work will be required to ensure the measurement uncertainty is sufficiently low to measure offsets this small.

Figure 11 shows a plot of the mean absolute frequency offset for all individuals in the population during run 1. An improvement in the average fitness of the population is clearly visible as the algorithm progresses, and this demonstrates that the population contains several individuals with a reduced frequency offset. It should be noted that due to the specifics of the GA used the population may contain less than 10 evaluated individuals and therefore the mean is calculated only from members with a valid fitness.

V. SUMMARY AND CONCLUSIONS

In this work we have demonstrated the effects of variability on the frequency of a simple ring oscillator circuit. Oscillator frequency variability maps are generated for a pair of PANDA chips by independently measuring the frequency of 90 ring oscillators. The median frequency of operation for each of these chips is 1.68535 MHz and 1.64935 MHz with total ranges of 30.5 kHz and 26.9 kHz respectively.

We have also show that, when combined with an evolutionary algorithm, the low-level configuration facilities provided in the PANDA architecture make it possible to mitigate some of this variation. Specifically we have demonstrated that by selecting alternative transistor configurations it is possible to minimise the difference in oscillation frequency between a frequency outlier and the chips median frequency of operation.

Future research will focus on two aspect of this work: Firstly, we will establish how much of the variation is a direct result of the intrinsic device variability in the CTs vs the surrounding digital routing and configuration fabric. Secondly, we will show that it is possible to optimise the offset frequency of multiple oscillators simultaneously and fully homogenise the PANDA fabric. Under the assumption that the relationship between the NMOS and PMOS transistors as well as the ordering of the ring oscillator stages matters, the combinatorial complexity of the search space is 6^{30} . If the offset frequency of multiple oscillators is optimised simultaneously then the search space increases further as a multiple of the number of oscillators. Evolutionary algorithms are well suited to exploring such large search spaces.

In summary, the increased configuration granularity provided by the PANDA architecture allows static and dynamic optimisation of a circuit's performance at the digital function and transistor level, as well as providing increased fault tolerance [7] [11] in the case of component degradation and failure during a circuit's lifetime. This is a novel approach to synthesizing and optimizing designs on programmable logic devices, which is not possible with any currently existing commercial FPGA.

ACKNOWLEDGEMENT

This work was part funded by EPSRC projects EP/I005838/1, EP/K040820/1 and EP/L000563/1.

REFERENCES

- [1] A. Asenov, S. Kaya, and A. R. Brown, "Intrinsic parameter fluctuations in decananometer mosfets introduced by gate line edge roughness," *IEEE Transactions on Electron Devices*, vol. 50, no. 5, pp. 1254–1260, May 2003.
- [2] A. Asenov, "Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 μm mosfet's: A 3-d ldquo;atomistic rdquo; simulation study," *IEEE Transactions on Electron Devices*, vol. 45, no. 12, pp. 2505–2513, Dec 1998.
- [3] P. Wilson and R. Wilcock, "Yield improvement using configurable analogue transistors," *Electronics Letters*, vol. 44, pp. 1132–1134(2), September 2008.
- [4] J. A. Walker, M. A. Trefzer, S. J. Bale, and A. M. Tyrrell, "Panda: A reconfigurable architecture that adapts to physical substrate variations," *IEEE Transactions on Computers*, vol. 62, no. 8, pp. 1584–1596, Aug 2013.
- [5] P. B. Campos, M. A. Trefzer, J. A. Walker, S. J. Bale, and A. M. Tyrrell, "Optimising ring oscillator frequency on a novel fpga device via partial reconfiguration," in *2014 IEEE International Conference on Evolvable Systems*, Dec 2014, pp. 93–100.
- [6] M. A. Trefzer, J. A. Walker, S. J. Bale, and A. M. Tyrrell, "Fighting stochastic variability in a d-type flip-flop with transistor-level reconfiguration," *IET Computers Digital Techniques*, vol. 9, no. 4, pp. 190–196, 2015.
- [7] M. A. Trefzer, D. M. R. Lawson, S. J. Bale, J. Walker, and A. M. Tyrrell, "Hierarchical strategies for efficient fault recovery on the reconfigurable panda device," *IEEE Transactions on Computers*, vol. PP, no. 99, pp. 1–1, 2016.
- [8] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, 2nd ed. Cambridge, MA, USA: MIT Press, 1992.
- [9] I. Harvey, "The microbial genetic algorithm," in *Proceedings of the 10th European Conference on Advances in Artificial Life: Darwin Meets Von Neumann - Volume Part II*, ser. ECAL'09. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 126–133. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2017762.2017781>
- [10] F. Vavak and T. C. Fogarty, "Comparison of steady state and generational genetic algorithms for use in nonstationary environments," in *Proceedings of IEEE International Conference on Evolutionary Computation*, May 1996, pp. 192–195.
- [11] D. M. R. Lawson, J. A. Walker, M. A. Trefzer, S. J. Bale, and A. M. Tyrrell, "Evolving hierarchical low disruption fault tolerance strategies for a novel programmable device," in *2014 IEEE International Conference on Evolvable Systems*, Dec 2014, pp. 77–84.