Operating Beyond FPGA Tool Limitations: Nervous Systems for Embedded Runtime Management

Matthew Rowlings, Andy M. Tyrrell, Martin A. Trefzer
Department of Electronic Engineering
University of York
York, United Kingdom
matthew.rowlings@york.ac.uk, martin.trefzer.york.ac.uk

***Abstract*—Fabrication issues throttle VLSI designs with pessimistic design constraints and speed-grade device binning necessary to avoid failure of devices. We propose that a on chip monitoring system (a Nervous System) can reduce this margin by automatically sensing and reacting to failures and environmental changes at runtime. We demonstrate that pessimistic margins in the FPGA tools allow our test circuit to be overclocked by twice the maximum design tool frequency and run at 50 °C above its maximum operating temperature without error. The Configurable Intelligence Array is introduced as a low-overhead intelligence platform and used for a prototype neural circuit that can close the loop between a timing-fault detector and a programmable Phase Locked Loop (PLL) oscillator.**

I. Introduction

Modern integrated circuit performance and cost has revolutionised the use and capabilities, as well as the number, of electronic devices that are now forming an integral part of our day-to-day lives. As we start to meet fundamental physical device limits, fabrication issues throttle a VLSI design with pessimistic design constraints that must be met to avoid faults, guarantee specific minimum lifetime of a device and mitigate inevitable mass-production variability. This results in conservative design approaches where worst-case corners in Electronic Design Automation (EDA) tools and post-manufacture processes such as speed-grade device binning lead to a waste of unused VLSI resources and prevents systems from running at their maximum possible performance.

Future design approaches that can handle faults are required to mitigate these constraints. The ability to adapt and perform reliably in the presence of these failures requires low-level monitoring of a device's operating state at runtime, and continuously managing key parameters, including clock frequencies, voltage rails and enabling sub-circuits.

However, in a modern SoC consisting of millions of transistors there are a huge number of potential monitoring and actuator points. This makes the autonomous management task difficult to achieve and inherently non-scalable, because of the centralised nature of monitoring points and the intelligence required for actuator decisions. To date, information redundancy localized fault detection architectures [1] and traditional redundancy strategies [2] result in prohibitively high system overheads in large-scale architectures and cannot currently locate or predict faults.

In this paper we approach this challenge by taking inspiration from an intelligent, scalable and power efficient fault detection and adaptive mitigation system: the Nervous System of biological organisms. The novel contributions outlined in this paper are: 1. A study that highlights the restrictions of FPGA binning, 2. The design of hardware-efficient neuromorphic decision units, 3. A prototype adaptive system based on this platform.

II. An On-Chip Nervous System

Neural pathways of Nervous Systems combine a large number of monitoring elements into closed loop, threshold-based decision units to actuate, e.g. muscular effector cells. It is seen that many sensory cells in insects and other simple creatures create a series of impulses as their output [3], a model that may translate well to monitoring digital hardware. It is envisaged that such a network could be overlaid on a complex SoC where "nerve endings" probe test points within the computing system and connected peripherals and examine temporal patterns, both locally and globally, to identify anomalies and use strategies to adapt the CPU and ancillary hardware (e.g. clocks, resets) to mitigate them. Off-the shelf system monitoring circuitry such as RAZOR timing fault detectors [4] and ring oscillators are used as shown in Fig. 1.

These embedded monitors connect to small spiking pathways, which aggregate signals in a way that is inspired by the peripheral nervous system, to enable intelligent sensing capabilities. Neurons from this aggregation layer feed into local small-scale Spiking Neural Netowrks (SNNs), which are capable of detecting and enacting local changes in circuit behaviour through actuator neurons. Such actuators are configurable design elements that operate at a low level such as, e.g. configurable clock tree delay taps, fractional clock frequency modification and fine-grained gate voltage control of local power islands. These feedback loops provide local runtime management to keep circuits within their safe operating envelope.

Due to the temporal nature of SNN control and the longer-term adaptation of our system, the nervous system can run at a much slower speed than the system it protects; resulting in a very low additional power consumption of the nervous system. These local control pathways can also be aggregated into larger

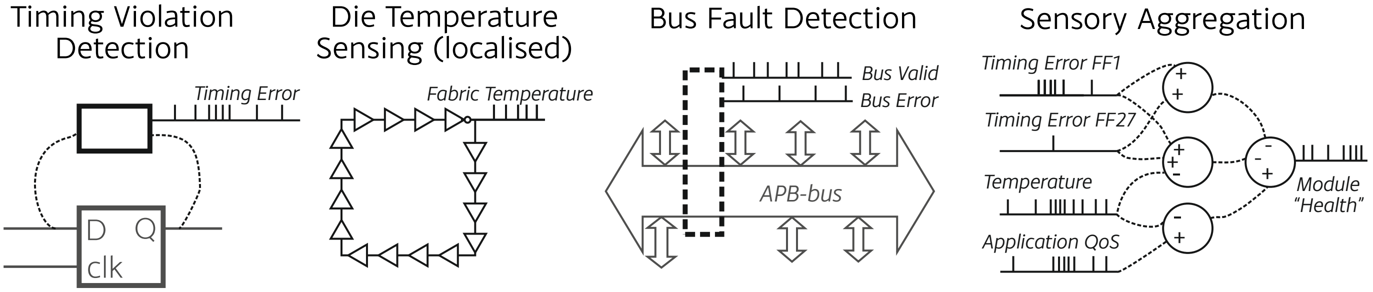

Fig. 1. Some of the embedded sensing modules that can easily be designed to output information in a digital spike train that is compatible with spiking neural network intelligence units. The fourth item is an example of an intermediary aggregation intelligence block that takes signals from several sensing module and outputs a spike train representing a higher-level property of the monitored hardware to be used with other intelligence units.

clusters of neurons capable of detecting and predicting system-wide anomalies and errors.

III. Runtime Characteristics of FPGA Designs

Adapting to both design requirements and operating environment is a key role of runtime management nervous systems. To explore what opportunities can be exploited, a series of experiments are performed to characterise the performance of a design produced by the Xilinx Vivado 2018.3 toolchain against the operating limit of the circuit when implemented on the device, an Artix-7 100T-1 FPGA. A 32-bit multiplier (as an illustrative example) has been implemented (using LUTs, no hard macros) and integrated with a soft-processor, which can load test data into the multiplier and verify the data calculated by the multiplier. This multiplier has then been implemented in eight different physical locations on the FPGA to account for variability in logic resources and routing used for each multiplier implementation. To asses the effectiveness of a potential fault detection monitor, the 32 output registers of each multiplier are instrumented with timing violation detectors (based on the RAZOR concept [4]). All multipliers have been synthesised and implemented for a 50MHz performance.

A. Overclocking

The first set of characterisation runs sweeps the clock frequency in increments of 1MHz for each of the eight 32-bit multipliers. 1,000,000 random multiplications are performed, checked and the number of errors is recorded to establish at which operating frequency the multiplier starts to suffer from timing errors. The results in Table I highlight the pessimistic approach taken by the FPGA tools when calculating the critical path, relative to the actual performance of the fabric. There is a small degree of variation between the multipliers at the eight different physical locations on the chip, which is likely due to the different routing and placement solutions found by the FPGA design tools, rather than variability in the performance of the LUTs themselves.

B. Overheating

The second experiment considers the operating environment of the device. A Peltier heater is mounted on top of the FPGA chip and used to heat or cool the device. The on-die temperature sensor of the FPGA is used to read out the temperature of the silicon die. The device datasheet [5] claims that the device under test is rated at 85°C. The temperature is swept in 10°C increments up to 125°C (the maximum temperature of the highest grade Artix-7 100T). For each temperature setting, a frequency sweep is performed to find the operating frequency that first produces errors. The results in Table I show that, at 125°C, not only can the device operate at this temperature but it can still be overclocked but with a slight decrease (up to 5MHz) in the maximum error-free frequency when compared to room temperature (Table I).

TABLE I
Results of A. Overclocking the FPGA and B. overheating and overclocking the FPGA. All circuits perform better than the tools suggest that their operating limit is.

DUT ID	**Design Tools Fmax**	**First Error Frequency at 30°C**	**First Error Frequency at 125°C**
1	*53.2MHz*	101MHz	99MHz
2	*51.3MHz*	101MHz	100MHz
3	*50.3MHz*	98MHz	93MHz
4	*51.8MHz*	100MHz	95MHz
5	*50.0MHz*	98MHz	94MHz
6	*51.3MHz*	98MHz	98MHz
7	*51.8MHz*	101MHz	98MHz
8	*50.0MHz*	103MHz	99MHz

C. Evaluation

It should be noted that the experiments presented here are not entirely fair on the FPGA design tools: the test designs have empty (unconfigured) space around the multipliers, which may dissipate heat, and the multipliers do not operate concurrently, which would be the worst case the design tools must consider. The device datasheet [5] suggests that there is approximately a 20% performance difference in CLB delay between a -1 (device used here) and -3 speed grade. This may explain some of the performance gap measured, as the device may be designed for -3 performance and then rated at -1 as a result of manufacturing yield.

This is also likely to be the case for the temperature performance, as the device may be designed for expanded or military range but is then derated to be sold as a lower-cost commercial grade device; most of the extra cost for higher-grade devices result from testing and qualifying processes. Therefore, it is likely that our -1 speed, commercial grade device is a binned higher-rated device for commercial reasons or due to faults in the non-evaluated parts of the die. In either case, it is clear our device has unused operating capacity that the standard design rules and pessimistic constraints would not allow us to use.

IV. Configurable Intelligence Array

To get the full performance from the multipliers, the design will need to move beyond the performance bounds provided by the tools. The real operating bound will need to be discovered and managed at runtime to ensure the system does not become faulty, indeed this point may change as the device ages or enters extreme operating environments. Use of techniques such as canary circuits allow faults to be anticipated before they manifest in the actual data path.

These experiments show the potential for Nervous System control of the clock frequency for each multiplier instantiation. This control must be provided by some decision making circuitry that must be area and power efficient enough to be embedded in the design with the application. Other studies [6] [7] have showed the efficiency of stimulus response-threshold models for making simple decisions whilst embedded in hardware. These models can also be used for aspects of the nervous system control. They take impulses as inputs and transform them into an output spike-train based on the threshold value and the input frequency of the excitatory and inhibitory inputs of the response-threshold unit. To allow rapid prototyping of such decision making pathways, our *Configurable Intelligence Array (CIA)* (Fig. 3) was developed. This is a hierarchical intelligence block with configurable decisions threshold units that have programmable routing between them. It is implemented as an FPGA macro and only requires 4 FPGA slices per configurable decision unit. The CIA consists of the following subsystems:

Configurable Intelligence Unit (CIU)

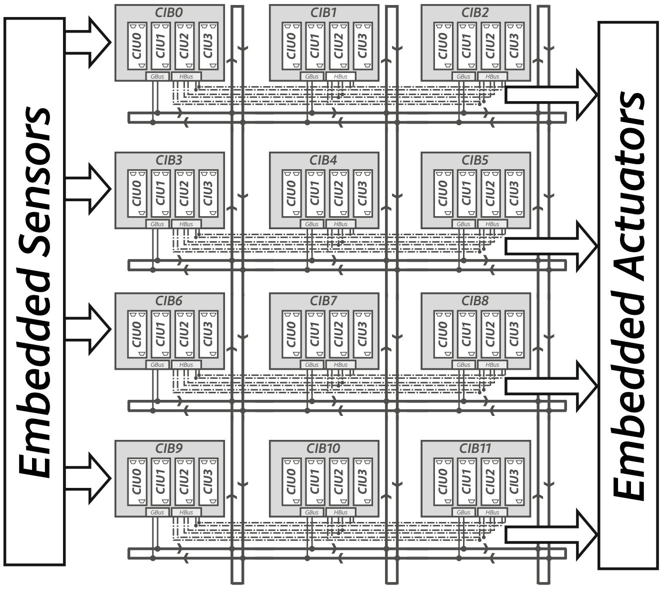

Fig. 3. The *Configurable Intelligence Array (CIA)*. This is a 4x3 array of CIBs and therefore provides 48 CIU thresholders.

Shown in Fig. 2, this is the fundamental decision making unit of the CIA. It implements a stimulus-response-threshold decision unit by taking excitatory and inhibitory spiking inputs. These inputs control an internal counter, incrementing and decrementing its value respectively. When this counter exceeds the threshold then an impulse is output and is used to drive an actuator or feed into further CIUs. The threshold value, inhibitory behaviour and output impulse behaviour can all be configured individually for each CIU.

Configurable Intelligence Block (CIB)

The CIB connects four CIUs together and provides programmable routing between them. It also provides global routing structures that allow input and output impulses to be shared between CIBs. Each CIB (including the four CIUs) requires 20 Artix-7 slices.

Configurable Intelligence Array (CIA)

The CIA tiles CIBs to form the larger array of decision threshold units (Fig. 3). It also provides input and output

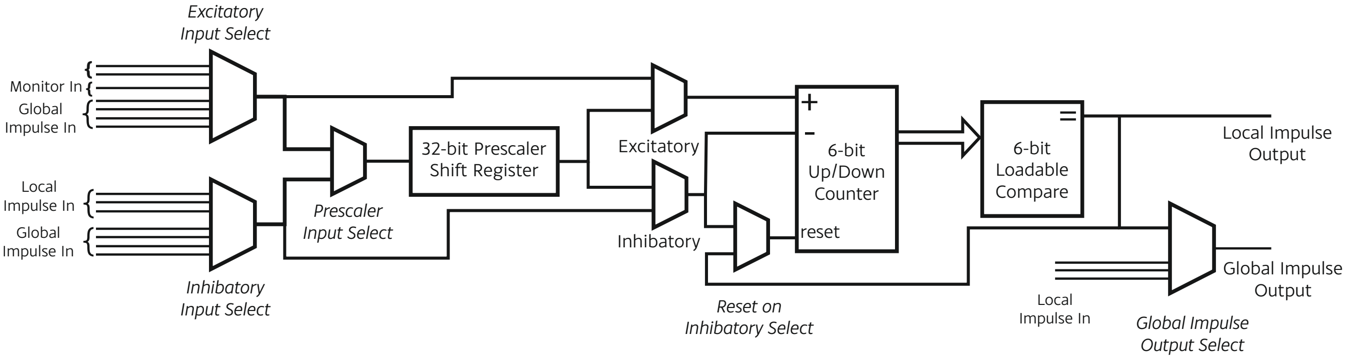

Fig. 2. A *Configurable Intelligence Unit (CIU)*. The CIU first selects the excitatory and inhibitory inputs and routes one of them through the prescaler shift-register. The excitatory and inhibitory signals are then fed into a 6-bit up/down counter. The output of this counter is compared to the value stored in the comparison unit and an impulse issued if the counter matches this value. The issue of the impulse will also reset the counter, as can an inhibitory impulse if the relevant configuration bit is set.

TABLE II
RESULTS OF THE PROOF-OF-CONCEPT EXPERIMENT.

DUT Frequency Range	Average CIA Spike Output Frequency	Average Number of Timing Errors
75 - 84 MHz	54.8 KHz	0
84 - 94 MHz	46.2 KHz	0
95 - 104 MHz	42.4 KHz	0.5
105 - 114 MHz	6.6 KHz	184
115 - 124 MHz	5.1 KHz	818

crossbars and conversion for monitor and actuator signals, as well as global configurations such as the clock-enable scaling of the array to allow the CIA to be run at a significantly slower speed than the application circuit it is protecting.

V. PROOF OF CONCEPT EXPERIMENT

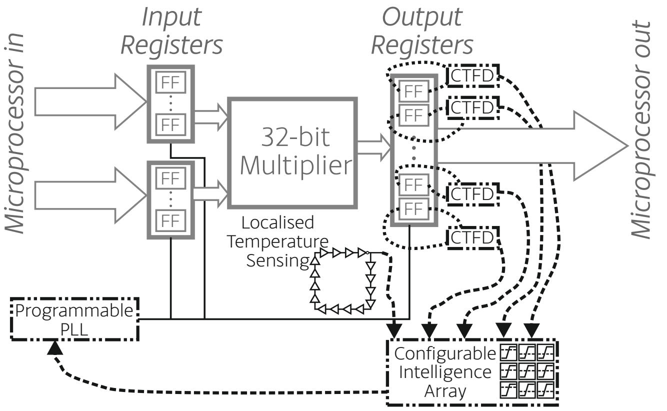

Fig. 4. Setup of our proof of concept experiment. *Configurable Timing Fault Detector (CTFD)* blocks allow optimal delay windows for complete timing violation detection to be achieved with minimal false-positives. The intelligence can use this information with the ring-oscillator to decide on the running frequency of the instrumented circuit (a 32-bit multiplier).

One of the multipliers from the test set-up is instrumented with the CIA with the goal of adapting the clock frequency of the multiplier autonomously to gain maximum throughput. Fig. 4 shows the test arrangement implemented. The 32-bit multiplier is instrumented with timing violation detectors on each of its 32 output registers. These output a spike when a timing fault is detected on the register. We also embed a ring-oscillator temperature sensor next to the multiplier, this will output a spike train frequency depending on the temperature of the silicon fabric. These sensing modules are our embedded instrumentation.

The intelligence outputs a spike train that controls the frequency of the clock driving the multiplier registers and uses the timing violation detectors to detect when the circuit is too fast. We configure the structure and thresholds of the intelligence array to maximise the clock frequency of the multiplier.

Table II shows the output spike frequency of the CIA as the operating frequency of the circuit is increased. The first errors occur at 103MHz. As the speed is increased beyond this point and errors occur, the output frequency of the CIA drops to indicate the clock should be slowed down. For closed-loop control, this output frequency would then be translated to the value that is passed into the PLL configuration interface.

VI. CONCLUSIONS AND FURTHER WORK

We have shown an example highlighting the significant effects that binning and product derating have on designing for digital devices. Whilst this is now standard practice for the industry, this derating will only get more significant as extra margin for ageing and fabrication with smaller technologies is accounted for. By modifying key system parameters at runtime, the system can be tuned for the actual performance of the hardware, in the actual operating environment and for the actual age of that device. EDA tool support will need to highlight what areas can be relaxed given the parameters that can be controlled. For example, runtime management of system clock frequency will allow a system to recover correct operation of a design element that has become slower due to ageing, but there is little point doing this if a critical system requirement is no longer met.

We have demonstrated how parts of a circuit can be transformed to be closed-loop, self-managing for maximum performance, using canary circuits to ensure that errors are detected and reacted to before they manifest. By studying the architecture of Nervous Systems, we aim to instrument more complex designs with more complex parameter sets using these self-regulatory feedback loops. A further challenge is how such self-regulatory aspects can be managed as part of the design process. Not all parts of the design are suitable candidates for self-regulation and not all monitoring and actuator points will be required to instrument a design. It is envisaged that inputs from both the designer and the implementation tools will be required when generating, inserting and training the control nervous system for a specific design. We are currently exploring how this can be integrated into design tool flows to produce a fully automated overlay architecture process for generic digital designs.

The authors thank EPSRC for funding under Platform Grant EP/K040820/1

REFERENCES

[1] A. Charif, N. Zergainoh, and M. Nixolaidis, “Addressing transient routing errors in fault-tolerant networks-on-chips,” in *IEEE European Test Symposium*. IEEE, 2016.

[2] A. Milluzzi and A. George, “Exploration of tmr fault masking with persistent threads on tegra gpu socs,” in *IEEE Aerospace Conference*. IEEE, 2017.

[3] P. Simmons and D. Young, *Nerve cells and animal behaviour*, 3rd ed. Cambridge University Press, 2010.

[4] D. Ernst, S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N. S. Kim, and K. Flautner, “Razor: circuit-level correction of timing errors for low-power operation,” *IEEE Micro*, vol. 24, no. 6, pp. 10–20, Nov 2004.

[5] Xilinx Inc, “Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics,” Xilinx Inc, Tech. Rep., 2018. [Online]. Available: www.xilinx.com

[6] M. Rowlings, A. M. Tyrrell, and M. A. Trefzer, “Hardware Implementation of Social-Insect-Inspired Adaptive Many-Core Task Allocation,” in *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*. Institute of Electrical and Electronics Engineers Inc., Feb 2017.

[7] M. Rowlings, A. Tyrrell, and M. Trefzer, “Embedded social insect-inspired intelligence networks for system-level runtime management,” in *Design, Automation and Test in Europe Conference*, Mar. 2020.