Nunez-Yanez, J. L. (2015). Adaptive Voltage Scaling with In-Situ Detectors in Commercial FPGAs. IEEE Transactions on Computers, 64(1), 45-53. 10.1109/TC.2014.2365963

Adaptive Voltage Scaling with in-situ Detectors in Commercial FPGAs

Jose Luis Nunez-Yanez
Department of Electrical and Electronic Engineering
University of Bristol, UK.
E_mail : j.l.nunez-yanez@bristol.ac.uk

Abstract— **This paper investigates the limits of adaptive voltage scaling (AVS) applied to commercial FPGAs which do not specifically support voltage adaptation. An adaptive power architecture based on a modified design flow is created with in-situ detectors and dynamic reconfiguration of clock management resources. AVS is a power-saving technique that enables a device to regulate its own voltage and frequency based on workload, process and operating conditions in a closed-loop configuration. It results in significant improved energy profiles compared with DVFS (Dynamic Voltage Frequency Scaling) in which the device uses a number of pre-calculated valid working points. The results of deploying AVS in FPGAs with in-situ detectors shows power and energy savings exceeding 85% compared with nominal voltage operation at the same frequency. The in-situ detector approach compares favorably with critical path replication based on delay lines since it avoids the need of cumbersome and error-prone delay line calibration.**

***Index Terms*— FPGA, energy efficiency, DVFS, AVS.**

I. INTRODUCTION

Energy and power efficiency in Field Programmable Gate Arrays (FPGAs) has been estimated to be up to one order of magnitude worse than in ASICs [1] and this limits their applicability in energy constraint applications. Since FPGAs are fabricated using CMOS transistors power can be divided into two main categories, dynamic power and static power. Lowering the supply voltage in CMOS circuits reduces dynamic and static power at the cost of increased circuit delay. As a result, voltage scaling is often combined with frequency scaling in order to compensate for the variation of circuit delay. Essentially, voltage and frequency scaling attempts to exploit performance margins so that tasks complete just in time obtaining power and energy savings. An example of this is Dynamic Voltage and Frequency Scaling (DVFS) which is a technique that uses a number of pre-evaluated voltage and frequency operational points to scale power, energy and performance. With DVFS, margins for worst case process and environmental variability are still maintained since it operates in an open-loop configuration. However, worst case variability is rarely the case. For this reason, in Adaptive Voltage Scaling (AVS) run-time monitoring of performance variability in the silicon is used together with system characterization to influence the voltage and the frequency on the fly in a closed-loop configuration. The importance of this technology for future microprocessor design has been discussed in [2] that advocates for the need to consider also hardware customization at run-time to deliver the performance increases and the low power required over the next 20 years. Addressing these issues the contributions of this work can be summarised as follows:

1. We present a power adaptive architecture based on in-situ detectors and adaptive voltage scaling suitable for commercially available FPGAs.
2. We present a novel design flow that creates the adaptive power architecture starting from an user design in RTL format.
3. We demonstrate the power and energy savings possible using as a test case an ARM Cortex-M0 processor obtained as an obfuscated netlist from the vendor.

The rest of the paper is structured as follows. Section 2 describes related work. Section 3 presents the hardware platform used in this research while section 4 introduces the design flow that embeds the AVS capabilities in the user design. Section 5 presents the power adaptive architecture based on the novel in-situ detectors. Section 6 presents and discusses the results focusing on power and energy measurements. Section 7 presents the final conclusions and future work.

II. RELATED WORK

In order to identify ways of reducing the power consumption in FPGAs, some research has focused on developing new FPGA architectures implementing multi-threshold voltage techniques, multi-Vdd techniques and power gating techniques [3-7]. Other strategies have proposed modifying the map and place&route algorithms to provide power aware implementations [8-10]. This related work is targeted towards FPGA manufacturers and tool designers to adopt in new platforms and design environments. On the other hand, a user level approach is proposed in [11]. A dynamic voltage scaling strategy for commercial FPGAs that aims to minimise power consumption for a giving task is presented in their work. In this methodology, the voltage of the FPGA is controlled by a power supply that can vary the internal voltage of the FPGA. For a given task, the lowest supply voltage of operation is experimentally derived and at run-time, voltage is adjusted to operate at this critical point. A logic delay measurement circuit

is used with an external computer as a feedback control input to adjust the internal voltage of the FPGA (VCCINT) at intervals of 200ms. With this approach, the authors demonstrate power savings from 4% to 54% from the VCCINT supply. The experiments are performed on the Xilinx Virtex 300E-8 device fabricated on a 180nm process technology. The logic delay measurement circuit (LDCM) is an essential part of the system because it is used to measure the device and environmental variation of the critical path of the functionality implemented in the FPGA and it is therefore used to characterise the effects of voltage scaling and provide feedback to the control system. This work is mainly presented as a proof of concept of the power saving capabilities of dynamic voltage scaling on readily available commercial FPGAs and therefore does not focus on efficient implementation strategies to deliver energy and overheads minimisation. A similar approach is also demonstrated, by the authors in [12]. A dynamic voltage scaling strategy is proposed to minimise energy consumption of an FPGA based processing element, by adjusting first the voltage, then searching for a suitable frequency at which to operate. Again, in this approach, first the critical path of the task under test is identified, then a logic delay measurement circuit is used to track the critical point of operation as voltage and frequency are scaled. Significant savings in power and energy are measured as voltage is scaled from its nominal value of 1.2V down to its limit of 0.9V. Beyond this point, the system fails. The experiments were carried out on a Xilinx ML402 evaluation board with a XC4VSX35-FF668-10C FPGA fabricated in a 90 nm process and energy savings of up to 60% are presented.

The previously presented efforts are based on the deployment of delay lines calibrated according to the critical path of the main circuit. This calibration is cumbersome and it could lead to miss tracking due for, for example, the different locations of the delay line and the critical paths of the circuit having different temperature profiles. In-situ detectors located at the end of the critical paths remove the need for calibration. This technology has been demonstrated in custom processor designs such as those based around ARM Razor [13]. Razor allows timing errors to occur in the main circuit which are detected and corrected re-executing failed instructions. The latest incarnation of Razor uses an optimized flip-flop structure able to detect late transitions that could lead to errors in the flip-flops located in the critical paths. The voltage supply is lower from a nominal voltage of 1.2V (0.13µm CMOS) for a processor design based on the Alpha microarchitecture observing approximately 33% reduction in energy dissipation with a constant error rate of 0.04%. The Razor technology requires changes in the microarchitecture of the processor and it cannot be easily applied to other non-processor based designs. It also uses utilizes a specialized flip-flop. In this paper we present the application of in-situ detectors to commercial FPGAs that deploy arbitrary user designs. The presented approach uses the technology primitives and elements already available in the FPGAs and therefore does not require chip fabrication or redesign in order to be used. The novelty of this approach is the design flow formed by the tools and IP blocks that embed the in-situ detectors in the original netlist and the design of the in-situ detectors that need to maintain very strict timing constraints for the circuits to operate reliably. This is different from literature that considers the fabrication of new devices to enable the introduction of performance and power adaptation. In [14] the authors propose to use pulse logic as an alternative to static logic to enable a wider tuning range. Both Vdd and threshold voltage tuning are considered. The pulse mode architecture is evaluated in a custom design FPGA device fabricated using 90nm CMOS process. The design changes the effective threshold voltage as well as Vdd and their results obtain a 20% higher performance at the same energy consumption or 35% lower power at the same performance. In-situ detectors are also proposed in [15] that uses a transistor-level circuit to implement a pre-error flip-flop to detect late data transitions but with the possibility that some errors might affect the main circuit to obtain further savings. The authors proposed to trade power with a possible error rate and observed that with a very low error rate of 1E-15 active energy is reduced by 13.5%. The authors of [16] present a fine-grain supply-voltage-control scheme for FPGAs based on asynchronous principles. Logic blocks in the sub-critical paths are autonomously switched to a reduce voltage level. The dual-rail encoding used in the asynchronous device enables the system to select which blocks can be switch to the low power supply. The scheme proposes to use power domains as small as a single four-input logic block avoiding the need of level shifters to get small overheads. Two voltage levels 1.2 V and 0.9 V are considered in the fabricated prototype based on an e-shuttle FPGA.TI PowerWise [17] energy management uses hardware performance monitors that are embedded in the newly fabricated silicon chip to adjust the operational point of the device. TI makes a distinction between AVS and DVFS which is also used in this paper. DVFS rely on pre-characterized frequency-to- voltage pair look-up tables that the processor uses to optimize power. It is open-loop and it does not compensate for process and temperature variations. Adaptive voltage scaling (AVS) uses real-time feedback on process and temperature variations. This voltage is adjusted to compensate the monitored variations in process, temperature and power supply. This reduces the actual operating voltage margin and the overall energy consumed. Xilinx has also investigated the possibility of using lower voltage levels to save power in their latest family implementing a type of static voltage scaling in [18]. The voltage identification bit available in Virtex-7 allows some devices to operate at 0.9 V instead of the nominal 1 V maintaining nominal performance. During testing, devices that can maintain nominal performance at 0.9 V are programmed with the voltage identification bit set to 1. A board capable of using this feature can read the voltage identification bit and if active can lower the supply to 0.9 V reducing power by around 30%. This is a static configuration that maintains the original level of performance and takes place during boot time. Our approach adapts the operational point over a wide range of voltage (0.65 V to 1 V) and frequency levels at run-time adapting to temperature, process and workload changes automatically.

III. Platform Description

The research platform used is the Xilinx XUPV5-LX110T evaluation board (XUPV5) with a Virtex-5 XC5VLX110T FPGA manufactured in a 65 nm process technology. The XC5VLX110T is conventionally powered by DC-to-DC power supplies that ensure fixed, stable and noise free supplies to three main voltage sources; VCCAUX, VCCO and VCCINT. VCCAUX provides power to the clock resources and clock primitives in the FPGA. VCCO provides power to the input and output banks of the device. VCCINT provides power to the logic resources of the device such as flip-flops, LUTs, configuration memory etc and as a result, heavily influences static and dynamic power. Static and dynamic power have approximately a quadratic and cubic dependency on voltage respectively. To vary the power consumption of the FPGA, voltage scaling is applied to the VCCINT voltage source. To achieve this, the DC-to-DC module that supplies the VCCINT voltage to the FPGA was redesigned to provide variable voltage without affecting the other voltage sources to the device. This was accomplished by first designing a voltage scaling module on a printed circuit board (PCB), then the original DC-to-DC module that provides a fixed voltage to the VCCINT terminal of the FPGA was replaced by the voltage scaling PCB as shown in Fig.1. The control signals that vary the voltage of the DC-to-DC module are then fed back to the I/O interface of the FPGA to form a closed-loop system. With this architectural layout, a power management solution implemented in the FPGA is able to control its internal voltage (VCCINT). The voltage scaling PCB is implemented using a PTH08T220WAZ DC-to-DC module from Texas Instruments. This module delivers up to 16 A output current and operates at efficiencies of up to 96%. The output voltage of this module is controlled by varying the resistance between a pin of the the DC-to-DC module - known as the the voltage adjust terminal - and ground. Conventionally, designs using the PTH08T220W utilise a fixed resistance of very low tolerance (1% and below) and very low temperature coefficient (ppm/K) to set a static and stable output voltage. As an example, a resistance of 20.5 kΩ sets the output to 1 V and a resistance 4.75 kΩ sets the voltage to 1.8 V. In our design however, the fixed resistance is substituted for a digital potentiometer to be able to vary the output voltage at run-time. The AD5282 from Analogue Devices is the digital potentiometer used in the voltage regulator PCB.

The maximum voltage rating of the Virtex-5 XC5VLX110T FPGA as quoted in the data-sheet is 1.05 V. To prevent damage to the FPGA, the voltage scaling module is constrained to this maximum voltage. This is achieved by adding an offset resistance in series with the resistance of the digital potentiometer. In practice, the offset resistance is implemented with another potentiometer to finely calibrate the maximum output voltage. Because the output voltage of the DC-to-DC module is inversely proportional to the resistance between the voltage adjust terminal and ground, when the digital potentiometer is set to 0 Ω, the voltage is limited to a maximum of 1.05 V by the offset resistance. When the resistance of the digital potentiometer is increased, the output voltage decreases. With this approach, the output voltage of the voltage scaling module never rises above the limit set by the offset resistance.

To permit the FPGA to control its own voltage, the control interface to the voltage scaling module - which uses the Serial Peripheral Interface (SPI) protocol - was connected to the general purpose I/O interface of the FPGA. Within the FPGA, a SPI slave controller was implemented. With this approach, a system configured in the FPGA can control its own voltage by adjusting the value of the digital potentiometer in the voltage scaling module through the SPI controller. It is also possible to bypass the voltage regulator and input an external voltage to the device directly. This was done during the verification and testing phase in which a Keithley sourcemeter was used to measure input currents and obtained power consumption values.

Figure 1. Voltage scaling PCB

IV. Power Adaptive Design Flow and In-situ Detection Logic

A. *Power adaptive design flow*

The power adaptive design flow introduces the in-situ detectors in the design netlist guided by post place&route timing information. The core of the flow is the novel *Elongate* tool that transforms the original design netlist into a new netlist with identical functionality and added power management core and in-situ detectors. Fig.2 shows the overall flow that can be decomposed into three distinct phases. During the first phase the original netlist goes through a full implementation run to obtain post place&route timing data in the form of a TWR text file. In the second stage the *Elongate* tool takes as input the obtained timing data, the original netlist and Elongate component library that describes the power management core and in-situ detectors and produces the new power adaptive netlist. The third stage consists of a final implementation run of the power adaptive netlist to obtain the device bitstream ready to be downloaded in the device.

The input into the flow is a netlist in either VHDL or Verilog format based on the implementation primitives available in the target technology. This means that initial synthesis is required to obtain the netlist that will be processed by the *Elongate* tool as shown in Fig.2 with the SYN block. The need for this initial pre-processing is because the *Elongate* transformation does not take place at source level directly. The reason is that slight changes in the source can have a large effect on timing and also

because it is possible to annotate the critical paths found after static timing analysis with the physical flip-flops in the netlist. The timing information contained in the TWR file is critical to allow *Elongate* to replace the end-point flip-flops in the critical paths with new soft-macro flip-flops that incorporate the in-situ detection logic. Each primitive flip-flop component in the technology library has a corresponding soft-macro flip-flop stored in the *Elongate* component library with identical functionality. Part of the user constraints input to *Elongate* indicate the level of coverage requested for the critical paths in the design. The coverage must be sufficient so that the critical paths of the final design have as endpoints the newly inserted soft macro flip-flops. If there is not enough coverage then the final implementation netlist could have critical paths not protected by the soft macros and the design could not operate reliably across the range of frequencies and voltages considered. To detect this situation the tool analyzes the final timing data to verify that the critical paths end in soft-macro flip-flops and that the slowest main flip-flop is located inside a soft-macro. If these constraints are not met the designer is informed so that a new run can be launched using a different path coverage value. As a rule of thumb our experiments have indicated that a coverage level of 10% of the total number of flip-flops is sufficient but this is ultimately dependent on how balanced the signal paths in the design are.

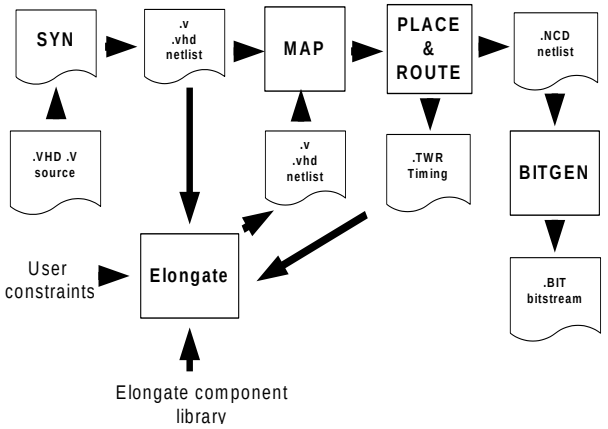

Figure 2. Elongate design flow

B. Detection logic

Fig.3 shows how the Virtex-5 logic slice is configured to create the soft-macro flip-flops. Three of the four logic cells available in the slice are configured to obtain the main original flip-flop (MFF), the slow flip-flop (SFF) and the XOR gate responsible for detecting different values between SFF and MFF. MFF performs the equivalent function as the flip-flop it replaces in the original netlist. SFF is configured exactly as MFF and adds an additional internal delay path to the logic cell as shown in Fig. 3 with the darker wires connected to the SFF flip-flop. Since only paths internal to the logic cell are used to create this circuit, the SFF is forced to be slower than MFF independently of the place&routing of the rest of the design. As long as the values of MFF and SFF are equivalent the XOR gate output remains at zero and no timing violations are detected. The circuit ensures that the SFF fails first and this is detected by the XOR gate and communicated to the power management control logic. The XOR gate output is input into a flip-flop as shown in the figure to remove any possible metastability resulting from timing failures in SFF. The difference in timing between the SFF and the MFF in the soft macro forms the speculative window which for the considered Virtex-5 device is approximately 0.181 ns. This value has been obtained using the vendor timing analysis and FPGA editor tools. This value determines the maximum instantaneous timing variation allowed following a change of frequency or voltage. As long as this constraint is kept SFF fails timing before MFF and this will result in a discrepancy that can be detected by the detection logic.

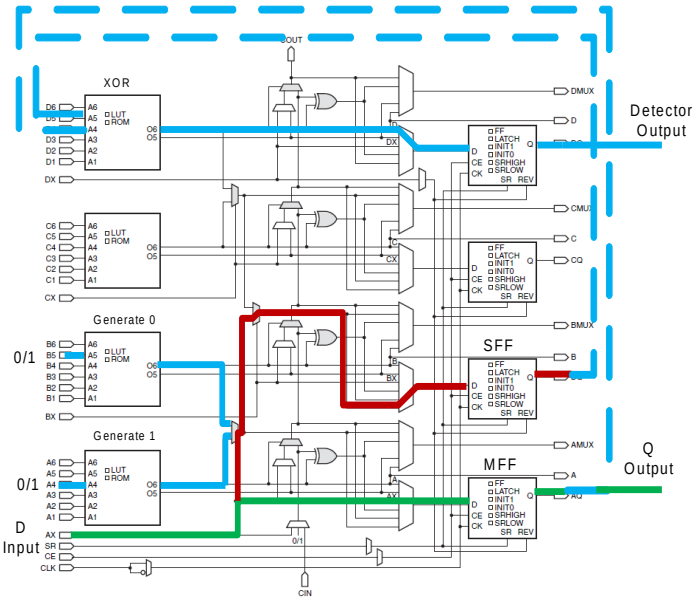

Figure 3. Elongate soft-macro flip-flop

To create this basic soft-macro flip-flop that can then be used to obtain functionally equivalent flip-flops in the component library a number of placement and routing constraints are necessary. These constraints must ensure that the positions of the SFF and MFF flip-flops and the internal routing inside the slice are locked. This is achieved using placement constraints associated to the soft-macro components and also by the two additional LUT's labeled as generate 0 and generate 1. These two LUT's control one of the internal multiplexors and create the internal propagation path. It is important to note that the vendor implementation tools will see the MFF and SFF as functionally equivalent and try to optimize and remove the circuit. To avoid this problem the values input into the LUT's are not hardwired to 0 or 1 and they are allowed to change when the device is reset. All these additional constraints are contained in the component library and are automatically used by the *Elongate* tool so the designer can use the technology transparently to these low level implementation details.

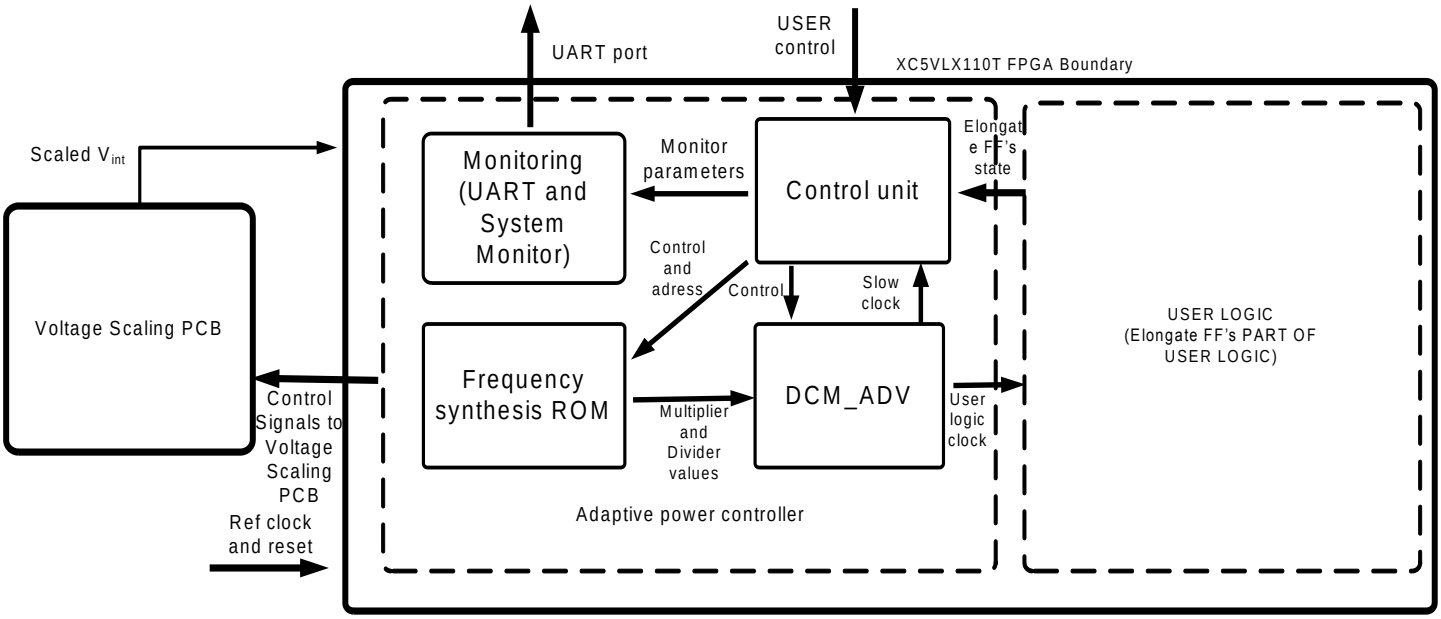

Figure 4. Power adaptive architecture

V. POWER ADAPTIVE SYSTEM ARCHITECTURE AND ROBUSTNESS ANALYSIS

A. Power adaptive architecture

Fig 4 shows the power adaptive architecture based on the in-situ detectors which are part of the *Elongate* soft-macro flip-flops embedded in the user logic. The control unit receives the outputs from the XOR gates part of the *Elongate* flip-flops and proceeds to ORed all these outputs to detect any timing violations. In an energy driven configuration once the voltage level is assigned the control unit finds the highest frequency that can be supported with that voltage. A frequency generation ROM memory forms part of the adaptive power controller. This ROM contains values for the Digital Clock Managers (DCM_ADV) used to generate the clock for the user logic. The outputs obtained from this memory are written by the control unit using the reconfiguration port available in the DCM_ADV block and new frequencies are generated at run-time. Once the DCM_ADV's have locked the clock is driven into the user logic. Once the frequency reaches a value that causes timing violations these are reported by the detectors and the state machine stops increasing the frequency until a new higher voltage is configured in the system. The power adaptive controller instantiates the system monitor IP block available in the FPGA device to monitor internal variables such as temperature and voltage.

It also includes a UART that is used to output these parameters and the state of the soft-macro flip-flops to a PC-based monitoring software where system state values are displayed. A screen-shot of the monitoring software is shown in Fig. 5. The upper line in the figure corresponds to the chip temperature. The internal sensors seem to introduce noise in the temperature measurement with low values display sporadically in the Fig.5. A possible explanation could be that this noise is due to the lower voltage being used in these experiments. In any case although the internal temperature varies as the operating point changes it remains within the valid range. Power was measured externally using a Keithley source meter and no power values are shown in the figure. The frequency parameter can be seen increasing monotonically until the detectors start firing. Fig.5 shows four distinct firing periods. In this test, voltage has been increased from 0.6 volts to 0.75 volts in four steps and the detectors can be seen firing at each of these steps. The firings correspond to timing errors in the SFF flip-flops and are collected during a small amount of time before being displayed in the tool with values ranging from zero to some non-deterministic value. In this test the user logic has been configured with the ARM Cortex M0 processor which runs an application that includes a software kernel for motion estimation used in video coding. The figure shows that although the task that the processor is executing is constant timing errors are affected by timing variability and the number of timing errors is not constant. In all the cases the errors do not affect the protected MFF's and the system computes correctly 100% of the time.

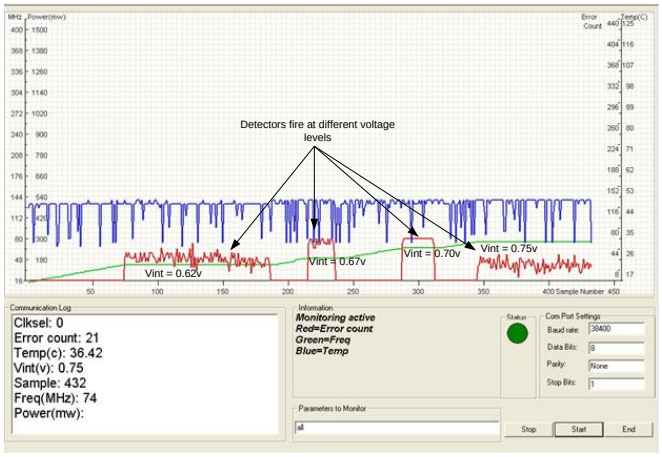

Figure 5. Elongate monitoring software.

B. Robustness analysis

The power adaptive architecture is designed to search for an optimal frequency for a given voltage value. In the test system the valid range of voltages extend from 0.6 V to 1 V. Frequencies are internally generated using the available DCM_ADV (Digital Clock Manager Advanced) and its capability to reconfigure at run-time. A problem exists if the instantaneous frequency change (in one single step) is such that both SFF and MFF failed timing and the signal does not land inside the speculative window. If this is the case then the system will stop working. This is illustrated in Fig.6 that shows the timing relations that must hold for the circuit to work. The first equation is the general timing equation and establishes that the clock period has to be large enough to accommodate the logic delay of the main circuit (T_c), the speculation window (T_w), the clock skew (T_s) and the clock uncertainty (T_u). The second equation is specific to elongate and establishes that the change in the clock period between two successive frequencies has to be smaller than $T_w - T_u$ since the clock uncertainty, which is 0.035 ns for the considered device, could potentially reduce the speculation window size. In this section we analyse the robustness of the proposed platform to this condition given the limited granularity of the frequency generation module.

As described in the previous section a ROM memory based on internal BRAMs is programmed with a set of multiplier and divider values for the DCM_ADV frequency synthesizer. The values in this ROM have been calculated so the granularity of the frequency generation is as high as possible. The current board uses an external reference clock at 100 MHz that drives the DCM_ADV. The range of possible frequencies extends from a minimum of 21.8 MHz to a maximum of 397.5 Mhz. Fig 7 shows the time difference in the period of one frequency and the next for the range of possible frequencies. The figure considers three reference clocks at 33, 100 and 200 MHz .

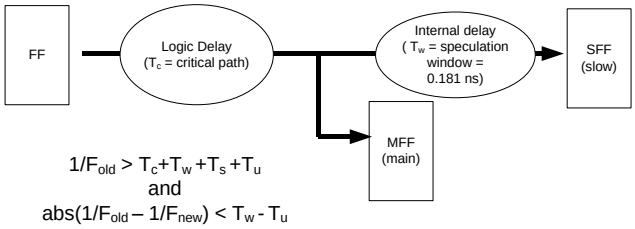

Figure 6. Timing constraints

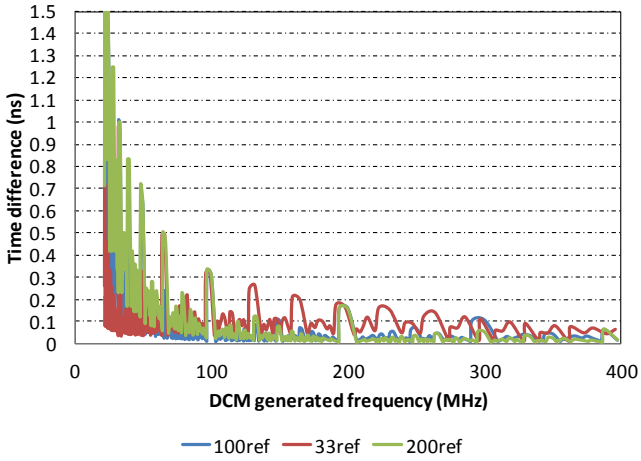

Figure 7. Frequency synthesis granularity

Looking at Fig. 7 is possible to observe that for DCM frequencies higher than 100 MHz the time difference is smaller than the speculative window (0.181 ns) for the 100ref and 200ref input clocks. The conclusion is that the design should have a working frequency higher than 100 MHz at nominal voltage of 1 V. It is not possible to determine the speculation window size at voltages lower than nominal using the standard static timing analysis tools and libraries provided by the FPGA vendor since characterization data at these lower voltages is not available but some reasonable assumptions can be made. As the voltage decreases the size of the speculative window should grow since signal propagation slows down so it is reasonable to expect that if this constraint holds at 1 V it will hold at a lower voltage. The testing conducted as part of the research has been used to verify this assumption. The constraint that the design should have a working frequency higher than 100 MHz at nominal voltage is reasonable since this working frequency is higher than the frequency reported by the tools. For example although the Cortex M0 has a reported frequency of 80 MHz the real working frequency exceeds 145 Mhz at nominal voltage. Naturally, this analysis is technology dependent and should be recalculated if a different device technology is being considered.

VI. Power and Performance Analysis

A. Test system

A test system based around the Cortex M0 processor from ARM has been built to test the capabilities and limitations of the system. The Cortex M0 is a popular processor from ARM with a three stage pipeline and optimized for low power and low cost applications. It executes the Thumb instruction set and can be obtained through the ARM university program as a Verilog obfuscated netlist. In this test system the Cortex M0 is connected through the AHBlite bus to 32 Kbytes of internal BRAM memory used to store program binaries and data. A number of processing kernels have been extracted from popular communication and video processing applications. The kernels consider include FFT, fast motion estimation and convolution. A particular benchmark set (e.g. EEMBC) has not been used in these experiments since the objective of these kernels is to exercise the Cortex M0 data and control paths in a bare metal configuration and this can be achieved without using a more complex benchmark set that will not fit in the limited internal memories. An example of the motion estimation sum-of-absolute-differences loop which runs for each point of a hexagon search pattern over a variable number of search steps is shown in Fig. 8.

```
int i_sum = 0;
    int x, y;
    for( y = 0; y < ly; y++ )
    {
        for( x = 0; x < lx; x++ )
        {
            i_sum += abs( pix1[x] - pix2[x] );
        }
        pix1 += i_stride_pix1;
        pix2 += i_stride_pix2;
    }
    return i_sum;
```

Figure 8. Sample kernel for the Cortex M0 processor.

Elongate can work with any general design and the ARM M0 is being used as a test case. The system composed for the M0 netlist and memories is then processed with the *Elongate* toolset and several coverage levels are considered from 100 to 300 paths. This is equivalent to approximately 10 to 30% of the total number of flip-flops in the design. Fig. 9 shows the complexity and performance of the resulting implementations. Logic utilization increases as the number of protected paths increases compared with the original design also shown in Fig.8. It is important to note that although the total number of flip-flops and LUT's increases considerably the number of slices where these LUT's are mapped increases more slowly. The explanation is that the soft-macros are designed to map the logic tightly in the slice so the number of additional slices required is moderate. The additional logic also adds some delay to the final implementation. Although it is reasonable to expect an increase due to the presence of the slower SFF flip-flops this increase is actually mainly due to the place&routing algorithms and it could change with different tool versions, constraints or designs to the point in which the *Elongate* delay is lower than the original delay. The circuit works correctly for all the configurations so in this case the best choice is to select the circuit with 100 paths to maintain the additional logic and delay to a minimum. Fig. 10 shows the valid frequency and voltage points for the protected circuits. The minimum stable voltage is 0.62 volts. Lower voltages create problems in the lock signal of the DCM_ADV blocks so they have not been used. For this voltage the circuit auto-detects a valid working frequency at around 40 MHz. The original design frequency, as reported by the tools after static timing analysis is 80 MHz, but the maximum valid working frequency generated by *Elongate* at nominal voltage is 163 MHz. This suggests that the margins needed to compensate for voltage, temperature and process variations can be effectively exploited to obtain both higher performance designs and lower power/energy profiles in commercial FPGA's.

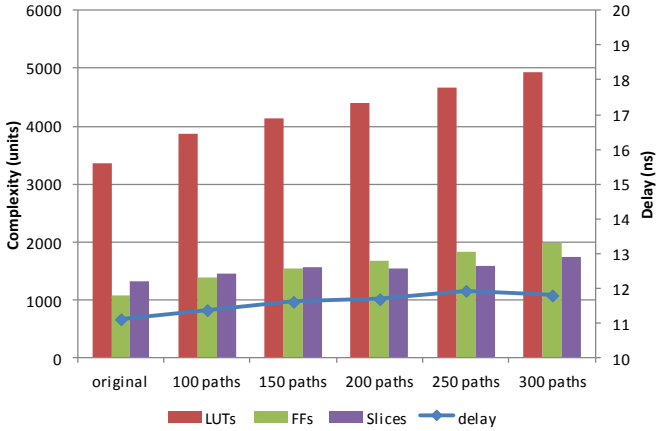

Figure 9. Cortex M0 implementation results.

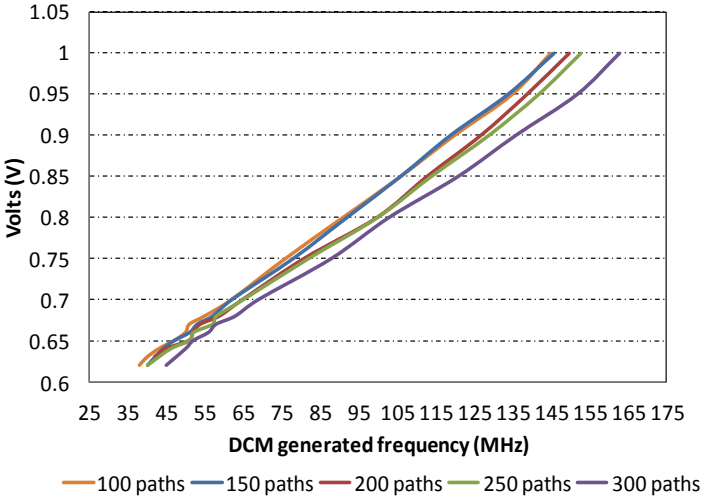

Figure 10. Frequency and voltage analysis

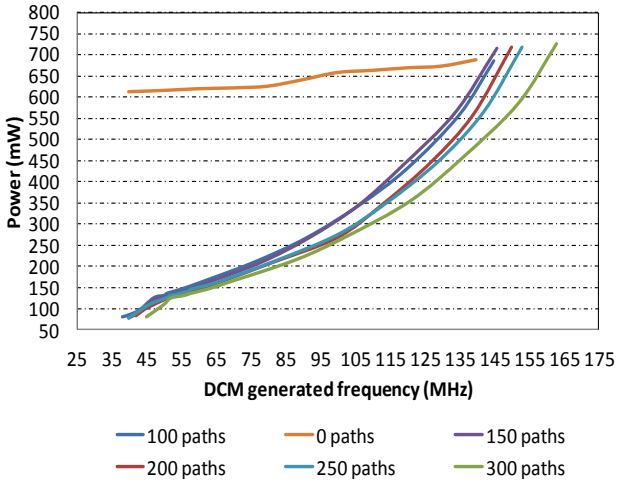

Figure 11. Static and dynamic power analysis.

B. Power and performance analysis

Fig. 11 analyzes the power consumption for the different configurations and compares them with power at nominal voltage for the original design. All these values correspond to power measured in the board which is the case for all the experiments reported in this paper. The minimum power at a working point corresponding to 0.62 V and 40 MHz is approximately 80 mW while the power at nominal voltage and the same frequency is 615 mW. This implies a power reduction of up to 87% compared with working at the same frequency and nominal voltage. At a nominal voltage of 80 MHz the system can operate at a reduce voltage of 0.75 V and power of 210 mW while power at 80 MHz and nominal voltage is 625 mW which implies a power reduction of 66% while maintain the same level of performance. Finally, at maximum performance of 145 MHz power consumption is equivalent but it is important to note that for the unprotected original design all working points with a frequency higher than the reported 80 MHz mean overclocking with no means of detecting if the new points are still functionally correct. The design with the protected paths auto-detects if the new operational points are valid to deliver safe and reliable operation adapting to process and temperature variations. These results are summarized in Table 1 for the 100 paths case.

Fig. 12 investigates the distribution of dynamic and static power as voltage is reduced. Static power has been obtained by stopping the reference clock input into the board. As expected a large proportion of total power is static due to the leakage of the fast transistors used by FPGA manufacturers to obtain circuits as fast as possible. The dependency of power with voltage is approximately quadratic for dynamic power and cubic for static power so reducing the voltage reduces both values significantly as it can be seen in the Fig. 12.

Freq	Original power	Elongate power	Description
40 MHz	615 mW	80 mW	Highest energy efficient point
80 MHz	625 mW	210 mW	Nominal performance point
145MHz	N/A	686 mW	Maximum performance

Table 1. Power consumption summary

An additional consideration is the possible overheads in terms of energy and performance introduced by the voltage and frequency changes. This test system operates with a single clock frequency manager which means that the system stops computing while the clock manager circuits lock a new frequency. The time overhead has been evaluated in Fig. 13. To alleviate this problem it is possible to follow an alternative approach in which additional clock managers work in parallel so one is ready with a new frequency while the other is re-locking and this optimization will be investigated in future work. The clock generation circuit is not stopped during voltage changes which take place in small steps of 1 mV. Although this can potentially introduce additional voltage noise in the power rails the tests conducted show tolerance to this noise and the system works correctly.

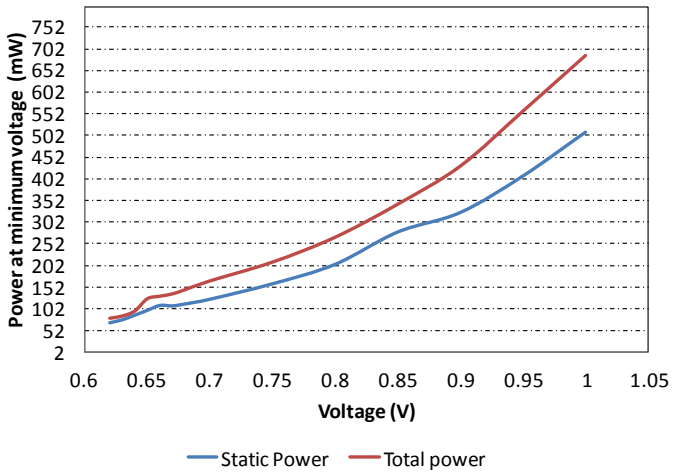

Figure 12. Power consumption analysis

C. Energy analysis

So far the paper has shown the important reduction of power that can be achieved with the power adaptive flow. An important consideration is how this relates to energy savings. Energy reductions will not be achieved if power reduction implies an equivalent increase in computational time. It is energy that limits battery run time or increases the running costs of a high performance computing centre so energy analysis is required to validate the potential of the proposed

techniques. For this experiment we have assumed a task that needs 10^6 cycles to complete and which at the minimum valid frequency of 38 MHz will need 26.3 ms to complete. This value is used to define T_{total} as shown in Fig. 14. As the frequency and voltage increases the active computation time defined by T_{active} in Fig. 14 decreases. Notice that we need to consider this division between T_{active} and T_{total} because if we simply consider that energy is the multiplication of total time by power we will be assuming that a system using a faster clock will use zero power once the computation completes and static power or leakage is reduced to zero. This will only be the case if the FPGA device could be power gated which is not a viable option in the considered SRAM FPGA since it will need to be fully configured with a considerable cost in terms of time and energy, potentially every few milliseconds.

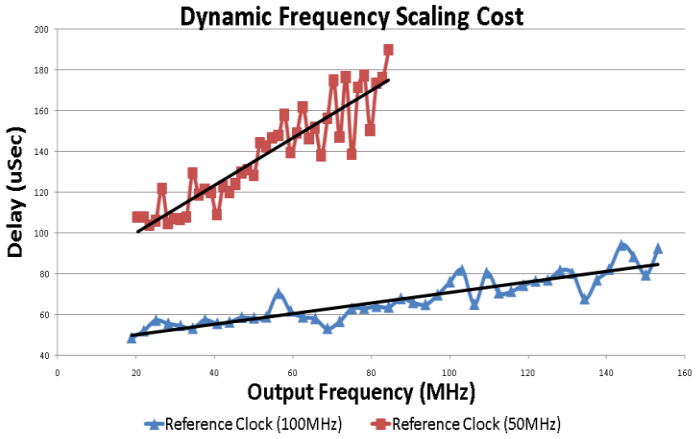

Figure 13. DCM locking time in function of input and output frequencies.

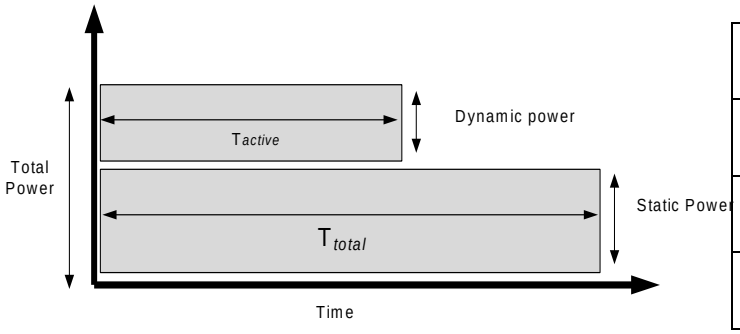

Figure 14. Power distribution for energy analysis

Consequently, the experiments consider that when the circuit is not active only static power remains until the end of T_{total}. For example a working frequency of 91 MHz requires 0.8 volts and T_{active} is 10.9 ms at which point only static power remains until the total time of 26.3 ms is reached. Power gating could become a valid alternative in future devices that include different voltage domains for the configuration memory and logic at which point these results will need to be re-examined.

Fig. 15 depicts the energy savings obtained using the Elongate technology in the Cortex M0 test case. Nominal energy corresponds to the original circuit working at nominal voltage for the range of frequencies considered. It can be seen that for the nominal case a reduction of frequency increases the total computation time in the same proportion and the required energy remains constant as expected. The optimal energy corresponds to the *Elongate* circuit that tracks the lowest voltage possible for the requested frequency. The savings in energy are comparable to those observed in power with 86% less energy at the highest energy efficient point and 68% less energy at nominal frequency working point. These results are summarized in Table 2.

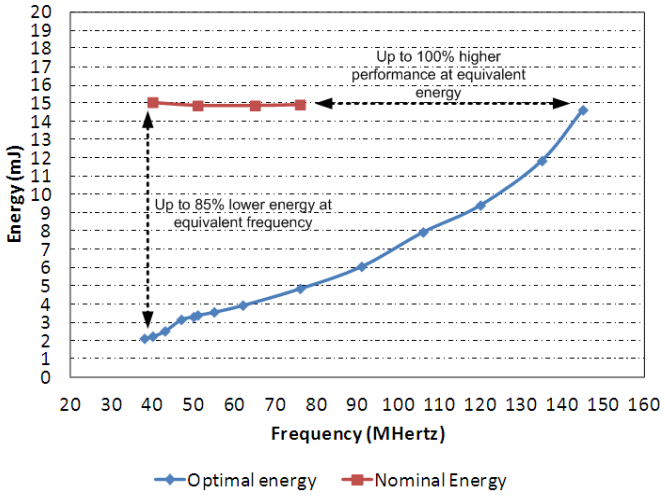

Figure 15. Energy analysis

Freq	Original Energy	Elongate Energy	Description
40 MHz	15.05mJ	2.24 mJ	Highest energy efficient point
80 MHz	14.92mJ	4.85 mJ	Nominal performance point
145 MHz	N/A	14.63 mJ	Highest performance

Table 2. Energy analysis summary.

VII. CONCLUSION AND FUTURE WORK

This paper has presented a novel design flow and IP library that enable the integration of closed-loop variation-aware adaptive voltage scaling in commercial FPGAs. The integration of in-situ detectors coupled to the critical paths of the design creates a robust architecture that removes the need of delay line

calibration and correction as done in previous work [12]. Although the FPGA devices employed have not been validated by the manufacturer at below nominal voltage operational points, the investigation shows that savings approaching one order of magnitude are possible by exploiting the margins and overheads available in the devices. Future work involves using the technology with other FPGA devices manufacture in different process nodes (e.g 40 nm and 28 nm) to investigate the margins that exist at lower feature sizes. We also plan using the technology with cores that include adaptive logic scaling so that multiple configurations with different levels of complexity, performance and power are possible. This should generate a new design paradigm in the form of Adaptive Voltage and Logic Scaling (AVLS) that can help address the energy and power challenges that current and future chips face.

REFERENCES

1. Kuon, I. and Rose, J. 2007. Measuring the gap between fpgas and asics. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 26, 2, 203 – 215.
2. Shekhar Borkar and Andrew A. Chien. 2011. The future of microprocessors. Commun. ACM 54, 5 (May 2011), pp. 67-77.
3. Rahman, A., Das., Tuan T., and Rahut, A. 2005. Heterogeneous routing architecture for low-power FPGA fabric. In Custom Integrated Circuits Conference, 2005. Proceedings of the IEEE 2005. pp. 183 – 186.
4. Ryan, J. and Calhoun, B. 2010. A sub-threshold fpga with low-swing dual-vdd interconnect in 90nm cmos. In Custom Integrated Circuits Conference (CICC), 2010 IEEE. pp. 1 –4.
5. Li, F., Lin, Y., and He, L. 2004. Vdd programmability to reduce fpga interconnect power. In Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on. pp. 760 – 765.
6. Li, F., Lin, Y., He, L., and Cong, J. 2004. Low-power fpga using pre-defined dual-vdd/dual-vt fabrics. In Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays. FPGA '04. ACM, New York, NY, USA, 42–50.
7. Raham A. and Polavarapuv, V. 2004. Evaluation of low-leakage design techniques for field programmable gate arrays. In Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays. FPGA '04. ACM, New York, NY, USA, 23–30.
8. Lamoureux, J. and Wilton, S. . On the interaction between power-aware fpga cad algorithms. In Computer Aided Design, 2003. ICCAD-2003. International Conference on. 701 – 708.
9. Lamoureux, J. and Wilton, S. 2007. Clock-aware placement for FPGAs. In Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on. 124 –131.
10. Gayasen, A., Tsai, Y., Vijaykrishnan, N., Kandemir, M., Irwin, M. J., and Tuan, T. 2004. Reducing leakage energy in fpgas using region constrained placement. In Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays. FPGA '04. ACM, New York, NY, USA, 51–58.
11. Chow, C., Tsui, L., Leong, P., Luk, W., and Wilton, S. 2005. Dynamic voltage scaling for commercial FPGAs. In Field-Programmable Technology, 2005. Proceedings. 2005 IEEE International Conference on. 173 –180.
12. Nunez-Yanez, J., Chouliaras,, V., and Gaisler, J. 2007. Dynamic voltage scaling in a FPGA-based system-on-chip. In Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on. pp. 459 –462.
13. S. Das, et al., Razor II, IEEE J. Solid-State Circuits, pp. 32--48, Jan. 2009.
14. Bita Nezamfar; Energy-performance Tunable Digital Circuits, Phd thesis, Dept. of Electrical Engineering, Stanford University, 2008.
15. Wirnshofer, M.; Heiss, L.; Georgakos, G.; Schmitt-Landsiedel, D.; "A variation-aware adaptive voltage scaling technique based on in-situ delay monitoring," Design and Diagnostics of Electronic Circuits & Systems (DDECS), pp.261-266, 13-15 April 2011
16. Shota Ishihara, Zhengfan Xia, Masanori Hariyama, Michitaka Kameyama,"Evaluation of a Self-Adaptive Voltage Control Scheme for Low-Power FPGAs", Journal of Semiconductor Technology and Science (JSTS), Vol. 10, No. 3, pp.165-175, 2010.
17. Information available at http://www.ti.com/ww/en/analogpower_management/powerwise-avs.shtml
18. Information available at http://www.xilinx.com/support/documentation/application_notes/xapp555-Lowering-Power-Using-VID-Bit.pdf

Jose Luis Nunez-Yanez received the B.S. degree from Universidad de La Coruna, La Coruna, Spain, and the M.S. degree from Universidad Politécnica de Cataluña, Barcelona, Spain, in 1993 and 1997, respectively, both in electronics engineering, and the Ph.D. degree in the area of hardware architectures for high-speed data compression from Loughborough University, Loughborough, U.K., in 2001. During 2005 he worked at STMicroelectronics, Italy after receiving a Marie Curie fellowship in the area vector architectures for video processing and in 2010 he worked ARM Ltd, Cambridge with a Royal Society fellowship in the area of system-level energy estimation and modelling. He is currently a Senior Lecturer with the Department of Electronic Engineering, University of Bristol, U.K. His current research interests include the areas of data and video compression, reconfigurable computing, energy efficient processors and brain modelling.