

Energy proportional computing in Commercial FPGAs with Adaptive Voltage Scaling

Jose Nunez-Yanez
Department of Electrical and Electronic
Engineering
University of Bristol, UK
+44 117 3315128
j.l.nunez-yanez@bristol.ac.uk

ABSTRACT

Voltage and frequency adaptation can be used to create energy proportional systems in which energy usage adapts to the amount of work to be done in the available time. Closed-loop voltage and frequency scaling can also take into account process and temperature variations in addition to system load and this removes a significant proportion of the margins used by device manufacturers. This paper explores the capabilities of commercial FPGAs to use closed-loop adaptive voltage scaling to improve their energy and performance profiles beyond nominal. An adaptive power architecture based on a modified design flow is created with in-situ detectors and dynamic reconfiguration of clock management resources. The results of deploying AVS in FPGAs shows power and energy savings exceeding 85% compared with nominal voltage operation at the same frequency or 100% better performance at nominal energy. 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

1. 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, this paper investigates Adaptive Voltage Scaling (AVS) in which 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 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.

2. 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-V_{dd} techniques and power gating techniques [2-6]. Other strategies have proposed modifying the map and place&route algorithms to provide power aware implementations [7-9]. 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 [10]. A dynamic voltage scaling strategy for commercial FPGAs that aims to minimise power consumption for a given 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 comparable approach also based in

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

FPGAworld'13, September 10, 2013, Stockholm, Sweden.
Copyright 2013 ACM 978-1-4503-2496-0/13/09 ...\$15.00

delay lines is demonstrated, by the authors in [11]. 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 1.0V down to its limit of 0.6V. Beyond this point, the system fails. 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 [12]. 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 in contrast with the dynamic approach investigated in this paper.

In-situ detectors located at the end of the critical paths remove the need for delay lines. 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 (0.13 μ m 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 removes the need of delay lines as done previously by the authors in [11] increasing the system robustness and efficiency. Additionally, it only uses the technology primitives already available in the FPGA and it does not require chip fabrication or redesign.

3. RESEARCH PLATFORM

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. 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. 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.



Figure 1. Voltage scaling PCB

4. 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 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. This detection logic is built using the available logic in the CLBs (Configurable Logic Block) and reports back to the adaptive power controller when the functional flip-flop is about to fail timing. The design of the in-situ detector and internal paths are illustrated in Fig.3. It consist of main flip-flop (MFF), a shadow

flip-flop (SFF), a metastability remover flip-flop and detection logic in the form of a XOR gate. This structure is similar to the logic used in commercial technologies such as Razor but the challenge in the FPGA is to guarantee that all the elements will be placed in a single FPGA slice so that the relative timing is not altered and that the tools do not naturally removed the apparently functionless detection logic. Part of the user constraints input to Elongate indicate the level of coverage requested for the critical paths in the design.

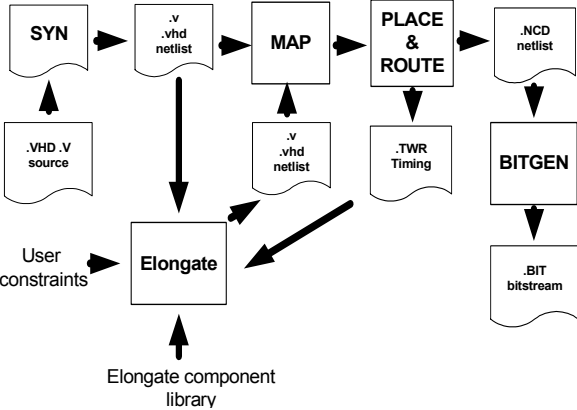


Figure 2. Elongate design flow

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.

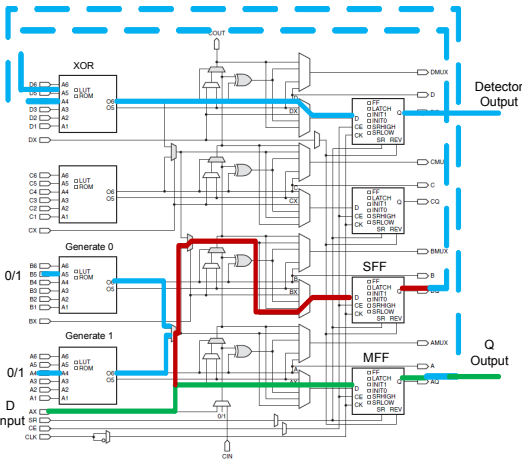


Figure 2. Power adaptive architecture

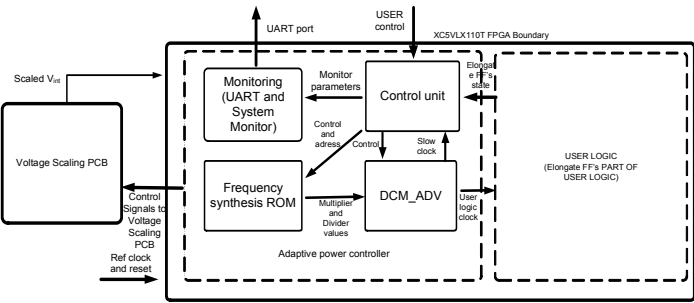


Figure 4. Power adaptive architecture

5. POWER ADAPTIVE ARCHITECTURE

Fig. 4 shows the power adaptive architecture. 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. The control unit achieved this by increasing the frequency level in small steps until timing violations are detected in the Elongate flip-flops. 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 in-situ detectors to a PC-based monitoring software where system state values are displayed. A screen-shot of the monitoring software is shown in Fig. 5.



Figure 5. Elongate monitoring software.

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. 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 figure shows that although the task that the processor is executing is constant critical paths are affected by timing variability and the number of timing violations is not constant. In all the cases the errors do not affect the protected flip-flops and the system computes correctly 100% of the time.

6. POWER AND PERFORMANCE ANALYSIS

6.1 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 implements 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 considered kernels include FFT, fast motion estimation and convolution.

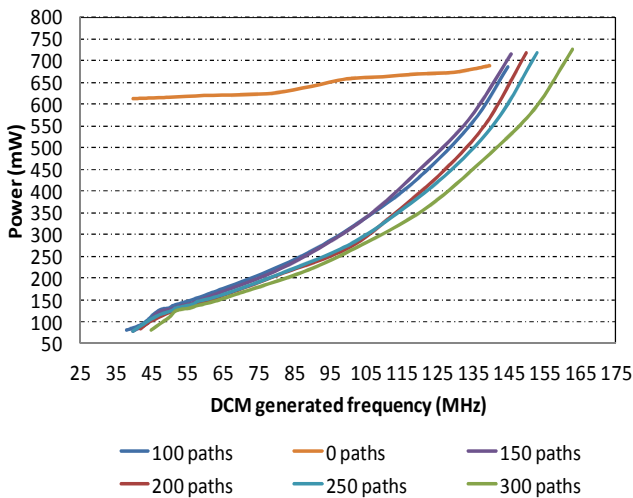


Figure 6. Power analysis

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. Logic utilization increases as the number of protected paths increases and ranges from additional 9% slices for the 100 paths case to 30% slices for the 300 paths case. 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 to a minimum. 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 indicates 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.

6.2 Power and performance analysis

Fig. 6 analyzes the power consumption for the different configurations and compares them with power at nominal voltage for the original design shown as 0 paths in the upper part of Fig. 5. 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 maintaining 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.

6.3 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. 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. As the clock frequency increases in different configurations active time decreases and only static power remains that has been measured stopping the reference clock inputs in the board.

Fig. 7 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.

7. 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 [11]. 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. 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.

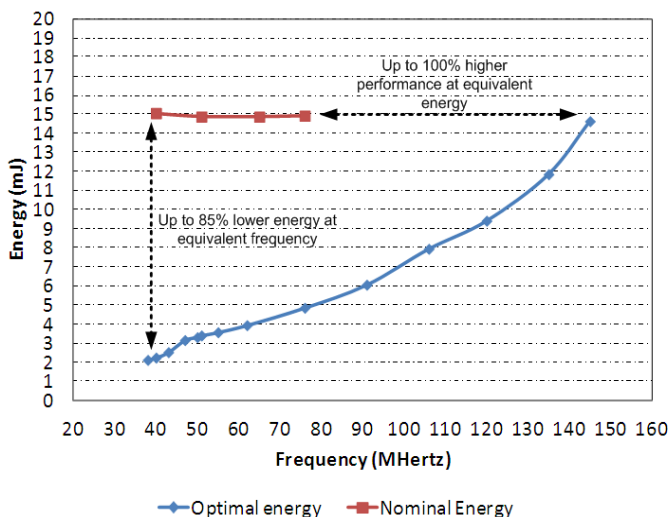


Figure 7. Energy analysis

8. 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] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] Lamoureux, J. and Wilton, S. 2007. Clock-aware placement for FPGAs. In *Field Programmable Logic and Applications*, 2007. FPL 2007. International Conference on. 124 –1
- [9] 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.
- [10] 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.
- [11] Atukem Nabina and Jose Luis Nunez-Yanez.. Adaptive Voltage Scaling in a Dynamically Reconfigurable FPGA-Based Platform. *ACM Trans. Reconfigurable Technol. Syst.* 5, 4, Article 20 (December 2012)
- [12] Information available at http://www.xilinx.com/support/documentation/application_notes/xapp555-Lowering-Power-Using-VID-Bit.pdf
- [13] S. Das, et al., Razor II, *IEEE J. Solid-State Circuits*, pp. 32--48, Jan. 2009. owman, M., Debray, S. K., and Peterson, L. L. 1993.