Robust Self-Calibrated Dynamic Voltage Scaling in FPGAs with Thermal and IR-Drop Compensation

Shuze Zhao[1], *Student Member, IEEE*, Ibrahim Ahmed[1], *Student Member, IEEE*, Carl Lamoureux[1], Ashraf Lotfi[2], *Senior Member, IEEE* Vaughn Betz[1], *Senior Member, IEEE*, and Olivier Trescases[1], *Senior Member, IEEE*
[1]University of Toronto, 10 King's College Road, Toronto, ON, M5S 3G4, Canada
[2]Intel Corp., Hampton, NJ, 08827, USA

***Abstract*— Field Programmable Gate Arrays (FPGAs) are widely used in telecom, medical, military, cloud computing and other high-performance computing applications, thanks to their unique combination of parallel hardware execution and reprogrammability. During compilation, the computer-aided design (CAD) tool estimates the maximum operating frequency of the user application based on the worst-case timing analysis of the critical path at a fixed nominal supply voltage, which usually results in significant voltage or frequency margin. Hence Dynamic Voltage Scaling (DVS) has great potential to reduce the power overhead in FPGAs; however the reprogrammability of FPGAs make a safe implementation of DVS for any application that could be programmed into the FPGA challenging. This work presents a robust universal DVS scheme for FPGAs intended to run on a system production line, or regularly during each FPGA power-up. The proposed scheme requires the FPGA to be programmed twice: offline self-calibration and online DVS. During the offline self-calibration, the FPGA frequency and core voltage operating limits at different self-imposed temperatures are automatically found and stored in a Calibration Table (CT). During online operation, the power-stage refers to the CT and dynamically adjusts the core voltage according to the FPGA temperature and the resistive voltage drop in the power delivery path. The proposed DVS scheme is demonstrated on a 60-nm Intel Cyclone IV FPGA, with a digitally-controlled dc-dc converter, leading to 40% power savings in two typical applications.**

***Index Terms*— Field-Programmable Gate Array (FPGA), dc-dc Converter, Dynamic Voltage Scaling (DVS), Self-calibration, Thermal Compensation, IR-Drop Compensation.**

I. Introduction

Field Programmable Gate Arrays (FPGAs) can be reprogrammed to accommodate new designs and evolving standards, and therefore they eliminate the need for custom manufacturing of an Application Specific Integrated Circuit (ASIC) and the attendant, high Non-Recurring Engineering (NRE) costs and development time. In addition, FPGAs can outperform general purposed processors such as Central Processing Units (CPUs), microprocessors and Digital Signal Processors (DSPs) in many applications, thanks to their ability to implement massively parallel algorithms [1]–[3]. Thus FPGAs are widely used in telecom, medical, military and cloud computing applications. However, they typically consume 14 times the dynamic power of an ASIC performing the same task [4], making power reduction techniques crucial for FPGAs.

Unlike ASICs, which have a fixed on-chip configuration, FPGAs have a unique programmable fabric. As shown in Fig. 1, the basic building block of an FPGA is a Logic Element (LE) composed of a programmable Look-up Table (LUT) and a register stage. FPGAs are built from thousands of LEs that are interconnected through programmable routing. During configuration, these LEs are programmed to perform the functions defined by the designer, which results in an application-dependent on-chip configuration.

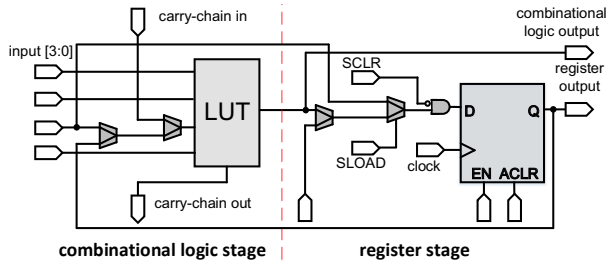


Fig. 1. Simplified LE cell of Intel Cyclone IV FPGA.

The Intel Chip Planner view of three different user applications is shown in Fig. 2. Clearly the on-chip location and the routing of their critical paths are quite different in each application.

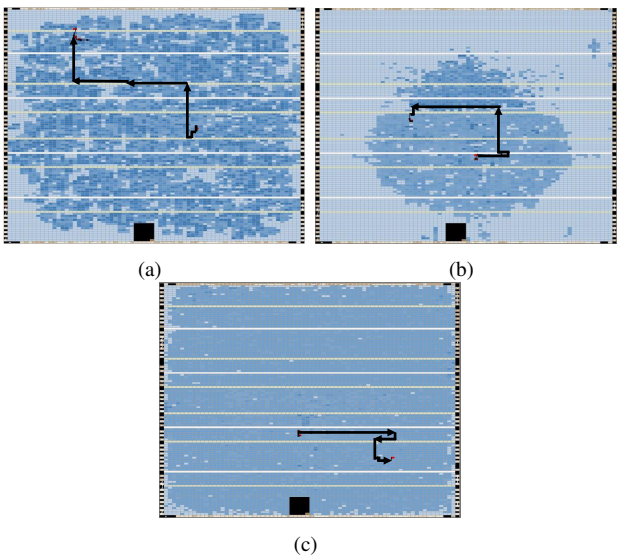


Fig. 2. On-chip configuration of three different applications and their most critical path, as reported by the Quartus Prime CAD tool. The shade of blue represents the relative utilization within each LE group. The black lines represent the routing of the most critical path. Note that while only one critical path is shown, there are usually multiple near-critical paths with similar delay.

Currently, most FPGA users operate each Integrated Circuit (IC) at its rated nominal voltage, and must choose a clock frequency at or below the limit predicted by the Computer-Aided Design (CAD) tool's timing analysis. Since this timing analysis assumes worst-case process corner, temperature, and on-chip voltage drop, the predicted clock frequency limit is quite conservative. In the vast majority of chips and systems, however, the supply voltage can be reduced significantly below nominal, while still meeting the required clock frequency. This reduction results in significant energy savings without any performance loss. Operating the IC at a lower voltage also reduces the impact of aging effects such as Bias-Threshold Instability (BTI), and improves the Mean Time Before Failure (MTBF) [5], [6].

Some vendor technologies provide users with limited flexility to operate certain advanced FPGAs at lower voltages for energy savings [7]–[9]. The Voltage Identification (VID) technique from Xilinx [7] allows chips with sufficient margin on every resource in the device can be operated at 0.9 V (nominal at 1 V) for all designs. As only a chip considerably faster than the speed bin limit on every resource can be run at 0.9 V in this scheme, its use is limited to the slowest speed grade/timing model. The Intel Smart VID technique for Arria 10 [8] similarly tests every device resource and stores a lower voltage rating in non-volatile on-chip storage if every resource can meet the rated speed at that lower voltage; this technique has finer voltage granularity than the Xilinx scheme. Arria 10 Smart VID also has a limited form of temperature adaptation in which the device is run at its nominal (higher) voltage at lower temperatures. Note that neither of these techniques adapt to the application by determining the voltage needed by the design-specific critical paths, nor do they adapt to the current drawn (and resulting IR-drop) of an application; instead they assume worst-case values for both of these effects. The approach proposed in this work therefore can outperform these non-design-adaptive approaches.

The programmable power technique of [9] uses programmable back bias to adjust the threshold voltages of pairs of logic blocks and their associated routing to reduce leakage in parts of an application that have sufficient slack. This approach does adapt to the design being implemented, but only targets static power and as the programmable back bias settings are the same for all devices, it cannot adapt to local variation in each device. Our approach is complementary to the approach in [9], as our calibration designs can simply use the same programmable back bias setting for each FPGA tile as was used in the original design compilation to ensure the calibration design captures the impact of programmable back bias on timing. Using the proposed DVS scheme in addition to programmable back bias will produce both dynamic power savings and additional static power savings over using programmable back bias alone.

A. Dynamic Voltage Scaling

Dynamic Voltage Scaling (DVS) has been widely deployed in microprocessor applications over the past two decades [10]–[16]. The dynamic power of a digital IC, $P_{dynamic}$, can be described by:

$$P_{dynamic} = \alpha C_{eq} {V_{core}}^2 f_{sys}, \quad (1)$$

where α is activity-factor, i.e., the fraction of the transistors that are switching; C_{eq} is equivalent total switched capacitance; V_{core} is the IC core voltage and f_{sys} is the operating frequency of the application. By scaling down V_{core}, $P_{dynamic}$ can be reduced quadratically. Further savings can be realized by the reduction in static power, which is increasingly important, especially at high operating temperatures, in modern CMOS technologies.

The minimum core voltage is determined by the critical path; therefore tracking the delay of the critical path is vital for DVS. A general DVS system for a microprocessor is shown in Fig. 3, where f_{ref} represents the reference/target frequency of the Voltage Controlled Oscillator (VCO), and the VCO acts as a replica of the critical path in the microprocessor. The VCO output frequency, f_{fb}, tracks the delay of the microprocessor critical path over Process, supply Voltage and Temperature (PVT) variations. The control loop regulates the Voltage Regulator Module (VRM) output voltage, V_{out}, to match the two frequencies, such that the microprocessor operates at the minimum core voltage.

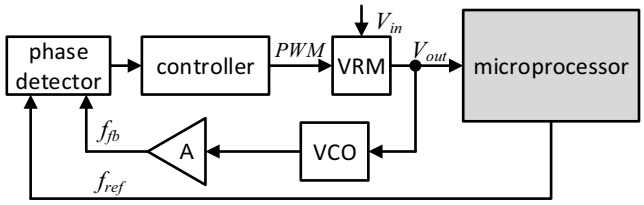


Fig. 3. A general DVS system.

Just as in microprocessors and ASICs, DVS in FPGAs has the potential to reduce both power and aging effects; however the fact that an FPGA can be reprogrammed to perform any digital function gives rise to some unique challenges in designing a DVS control system:

- Since the critical path in an FPGA is application dependent, it is impossible to achieve an accurate delay *vs.* PVT tracking using a pre-fabricated replica path scheme, unlike the microprocessor DVS system in Fig. 3.
- In a typical digital IC, there are multiple near-critical paths with nearly identical path delays. The actual critical path is then dependent on PVT conditions [17]. DVS systems for ASICs deal with this issue by adding guardbands to the single path based DVS, since the guardbands can be calculated or measured by the ASIC manufacturer through simulation and tests. This approach is not suitable for FPGAs, due to their application-dependent on-chip configuration.

B. Prior Work

In [18], a Logic Delay Measurement Circuit (LDMC) is used to determine the voltage at which the application circuit has a timing failure, and adjusts the supply voltage accordingly. It assumes that the critical path in the application circuit can

be exercised by randomly generated inputs during calibration, which is not a valid assumption for most modern FPGA applications. The approaches in [19] and [18] also assume that the VCO/LDMC delay perfectly tracks the delay variation in the application circuit critical paths with temperature and aging; this assumption is not met by modern processes.

In [20]–[22], online timing slack measurement is achieved by using a phase-shifted clock and one shadow register for each timing-critical register in the design to determine the timing headroom in the design during operation. In [23], a new Dynamic Voltage and Frequency Scaling (DVFS) tool, *Elongate*, is proposed, which inserts slow flipflops as logic monitor circuits in the original design netlist, which similarly allows for measurement of timing headroom in a running design. The strength of this type of work [20]–[23] is that it can monitor the timing of many near-critical paths, but it also has several significant shortcomings:

1) The timing slack measurement is dependent on the input data, which cannot be controlled during normal operation.
2) The technique is limited to FPGA components where a second capture register can be added at the end of a critical path, which is not feasible for important 'hard' blocks such as the on-chip RAM.
3) The scheme requires extra LEs and clock resources, increasing circuit power and reducing the usable capacity of the FPGA.
4) 100% coverage of all near-critical paths is not guaranteed due to routing limitations.

Similar methods have also been used for DVS in microprocessors and ASICs [24], [25]; they use an additional path near each critical path for error checking or correction. The exact transistor-level design cannot be directly used in FPGAs, and they still have the same aforementioned issues as [20]–[23].

In addition, the past works [18], [20]–[22], [25] do not employ a high-frequency digitally-controlled dc-dc converter to generate the variable core voltage. Many important practical issues, such as the voltage ripple, converter response time and quantization issues are therefore not considered.

The DVS technique demonstrated in this work has several advantages over the prior art:

1) It uses a combination of offline calibration and runtime monitoring to cover many near-critical paths to ensure the technique is reliable, while simultaneously minimizing the on-chip resources added to the running application to avoid area and power overhead.
2) It monitors the load current and chip temperature, and adapts the voltage to compensate for the IR voltage drop and temperature-dependent path delay in real time.
3) It can be extended to ensure reliable operation of all FPGA resources, including hard blocks such as RAM and Digital Signal Processing (DSP) blocks that have less programmability than the logic fabric.

In addition, the proposed DVS technique is demonstrated on an FPGA powered with a Switched-Mode Power Supply (SMPS) in real-time.

A portion of this work was published in [26]; that work proposed an offline universal self-calibration-based DVS scheme, which takes advantage of the digitally-controlled dc-dc converter to automatically characterize the exact relationship between the maximum operating frequency for each core voltage and temperature corner. This information is saved in a Calibration Table (CT) that is used for DVS during normal operation. This paper expands upon [26] in several aspects:

1) Several hundred critical and near-critical paths are monitored during the characterization phase, whereas only the most critical path was considered in [26].
2) Imperfections in the characterization due to the fan-out of the critical paths not being replicated are calculated via static timing analysis, and used to determine safe guardbands for the DVS control circuit.
3) The IR-drop in the power-delivering path is estimated and accounted for to produce an accurate core voltage.

All experimental results presented here incorporate the above improvements and are new compared to [26]. In addition, the proposed DVS scheme is demonstrated on two common applications, a digital Finite Impulse Response (FIR) filter and a CrossBar switch. Note that the IR-drop self-measurement method used in this work is presented in [27], and the results are directly used in the IR-drop compensation section; a detailed discussion of the IR-drop methodology can be found in [27].

The paper is organized as follows. The self-calibration scheme for DVS is described in Section II. The detailed system architecture and ideal waveforms for self-calibration are shown in Section III. Experimental setup and measurements are reported in Section IV and finally, the conclusions are summarized in Section V.

II. Self-Calibration based DVS

The proposed universal self-calibration process is intended to run on a system production line, or during each power-up sequence of a system, in order to generate the CT for the DVS operation of the application. It is therefore important that this process (1) be reasonably fast, (2) require the minimum possible FPGA resource overhead, as well as be (3) highly automated and (4) robust enough to guarantee correct operation in the worst-case conditions. The self-calibration-based DVS has three steps and requires the FPGA to be programmed twice, as shown in Fig. 4:

Step 1) The user's application design is automatically analyzed by *FRoC* - the augmented CAD tool to extract the logic paths having the most critical timing. An application-specific self-calibration bit-stream is then created together with the application bit-stream. The FPGA is subsequently programmed twice with these two bit-streams, as discussed in steps 2) and 3) respectively. In our past work [17], an automated tool, *FRoC*, is presented which is capable of automating the generation of the calibration bit-stream. The critical paths used in self-calibration are *exact* replicas of the critical paths in the application; they are placed and routed using identical resources (routing paths, LEs, etc.). All inputs along the critical paths are set to non-controlling values to guarantee that the paths are sensitized. The output of each LUT

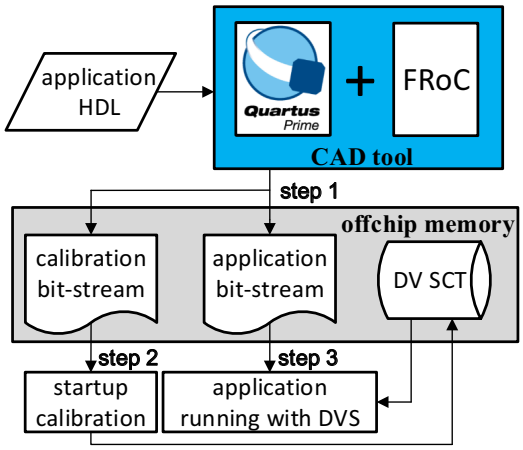


Fig. 4. Self-Calibration-based DVS scheme.

in any critical path can either be inverting or non-inverting with respect to the tested input. This means that for a path with n LUTs, there are 2^n possible inverting/non-inverting combinations. For each critical path, the control signals are set to mimic the worst-case inverting/non-inverting behaviour that was determined by the static timing analyzer. Note that the path fan-out is not precisely modeled in *FRoC*; therefore, a fan-out compensation factor is introduced in the CT which is calculated based on the difference between the f_{max} reported by Quartus Prime for the application and the calibration HDL [17].

Step 2) The FPGA is programmed once with the self-calibration bit-stream, as shown in Fig. 5(a). The on-chip configuration contains:

- the application-specific critical paths with stimulus, sensitization control and error checking circuits,
- flip-flop chain based logic blocks configured as programmable heaters for temperature control,
- a temperature sensing circuit,
- a frequency synthesizer,
- a digital dc-dc controller,
- a calibration controller.

Each application-specific critical path is exercised by toggling the source register and the error checking circuit monitors the output node to check for timing failures. The detailed implementation of the error checking circuit is described in [17].

The "heaters" consist of arrays of generic logic elements clocked by clk_heat at a fixed high-frequency; when all the heaters are activated they can draw substantial current on-chip. Heater cells are distributed across the entire chip, and are used to actively control the junction temperature. The FPGA proceeds to run the calibration, using self-heating and automatic timing failure checking, and populates the DVS CT. An ideal CT is shown in Fig. 6 and the detailed calibration scheme is described in Session III.

Step 3) Finally, when the self-calibration is complete, the FPGA is automatically programmed a second time with the user's application, as well as the DVS controller, as shown in Fig. 5(b). Based on the clock frequency requirements and chip temperature, the DVS control block refers to the CT to

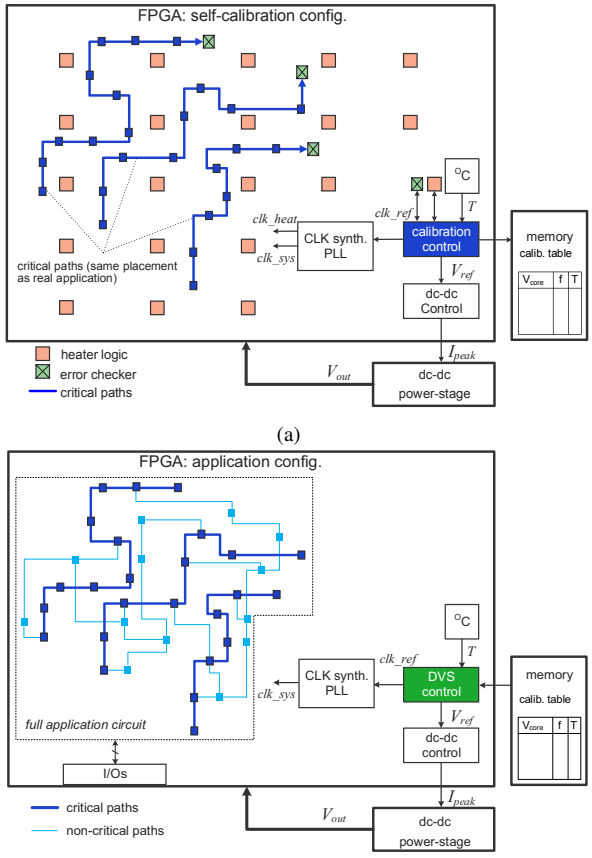


Fig. 5. FPGA with (a) self-calibration configuration and (b) application configuration for DVS system.

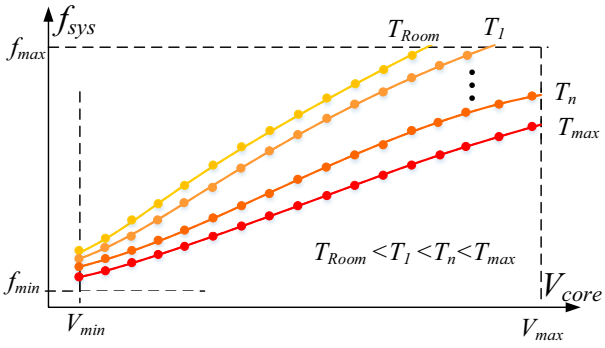


Fig. 6. An ideal CT as the result of the self-calibration process.

set the core voltage, V_{core}, accordingly in real time. Real-time IR-drop compensation is also added to the converter output voltage to address the resistive voltage drop in the Power Delivery Network (PDN), as described in Session IV-*B*. The IR-drop compensation method is described in detail in [27] and not repeated here. Note that unlike ASICs or microprocessors which perform DVS (or DVFS) based on their operating mode or task scheduling [28]–[30], the proposed DVS scheme only responds to the varying chip temperature and load current.

The proposed DVS scheme only requires the general resources available on any commercial FPGAs and can therefore be applied to any FPGA family. The design flow does not require any extra manual steps from the user; the CAD tool

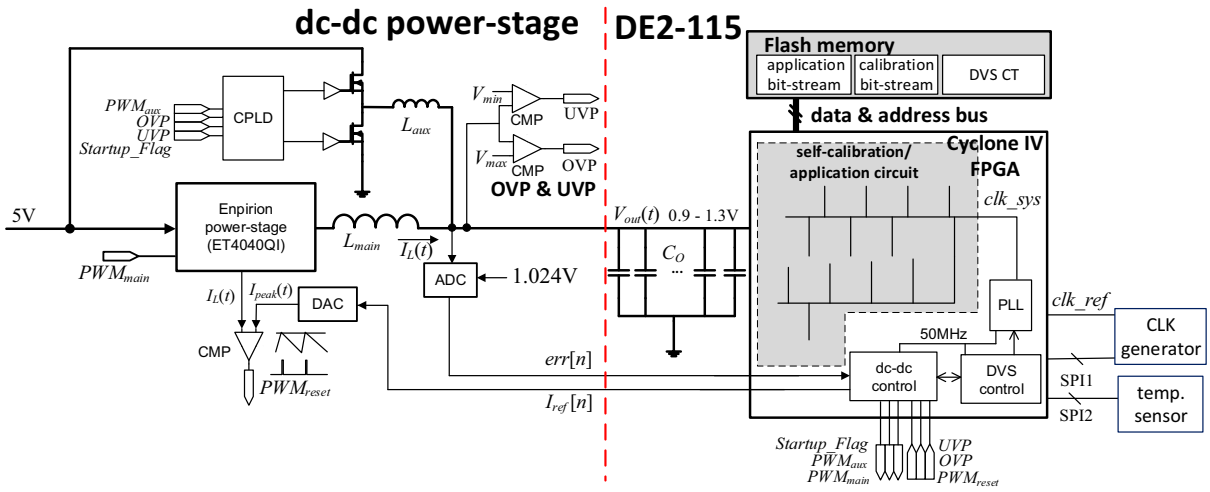


Fig. 7. Detailed system-level implementation.

works as a black box to generate two bit-streams. Once the bit-streams are downloaded in the Flash memory on the FPGA board, the FPGA can perform self-calibration and normal operation automatically. Since the scheme is application-dependent, the self-calibration should be performed every time the FPGA is programmed with a new design. Ideally, it only needs to be done once for each application. However, to account for aging of the FPGA, variation in passive components and other peripheral IC components, the FPGA should be re-calibrated every few months to get an updated CT.

Though the proposed scheme is robust over a variety of conditions, certain aspects of DVS do not lend themselves to accurate characterization under this model. Consequently, sufficient guardband should be added to the CT to ensure reliable operation under these aspects, which include:

- **Crosstalk**. Crosstalk is the coupling of two or more signals on the FPGA die. It can cause a slowdown or a speed-up in a signal transition. Crosstalk effects are extremely difficult to replicate in the calibration circuit, and therefore are best addressed by adding a guardband to the calibration table to account for worst-case crosstalk not being exercised during calibration. Ideally this would be the same guardband that is used in the commercial FPGA timing analyzer [31], but as that data is not public, we could use an experimentally determined guardband that is typically a few percent of the path delay.
- **Clock uncertainty and jitter**. This may exist on external clock sources, internal clock networks and other on-chip clock management blocks. By testing the same paths over many clock cycles, the proposed scheme can capture high-frequency clock jitter in the system and reduce the guardband required for clock jitter.
- **Power supply transients and PDN**. The static timing analysis has guardbands for power supply transients and PDN and it should be included in the CT as well. The proposed method considers the IR voltage drop in the PDN, which allows us to remove part of this guardband.

III. System Level Architecture

The system architecture is shown in Fig. 7 and includes a 60-nm CMOS Intel Cyclone IV FPGA (EP4CE115F29C7N). The two-phase Buck converter has an input voltage of 5 V and regulates the FPGA core voltage, V_{core}. The main phase, which delivers the FPGA power most of the time, is implemented using an Intel Enpirion power module, ET4040QI. The main phase is rated at 10 W and operates in digital peak current mode control. The digital controller is implemented in the load FPGA and is carefully optimized to operate down to the minimum FPGA core voltage, V_{min}. While the latest-generation FPGAs include both on-chip temperature and core voltage sensing, this project considers the general case where only off-chip measurements are possible.

The auxiliary phase, which has a lower power rating of 3 W, is controlled by a non-volatile CPLD to assist with the startup process when the main-phase controller in the FPGA is not programmed. After the power up from the auxiliary phase, the FPGA is programmed with target bit-stream that contains the main phase controller and the main phase takes over the control. The auxiliary phase is then disabled. The CPLD can be omitted in future implementations in which the startup control can be integrated into the power management IC.

The fully automated calibration process is shown in Fig. 8 and can be explained as follows. The heater cells are first enabled to cause the die temperature to ramp up. Each heater cell is programmable and consists of N_{heater} = 8 groups of inverter chains switching at 250 MHz. Each heater group can be individually enabled/disabled to draw different level of current on-chip for better temperature control. With heater cells enabled and V_{out} = 1.2 V, the FPGA package reaches 80°C. At every 5°C, CT entries are obtained and stored in the on-board Flash memory. During each sweep, the dc-dc controller drops V_{out} to V_{min} = 0.896 V and starts to increase the clock frequency, f_{sys}, from the lowest operating frequency. V_{min} is set at 0.896 V as it is close to the chip reset voltage. An increasing clock frequency, f_{sys}, is applied to the critical paths until a logic error is detected (err_flag = 1, when f_{sys} =

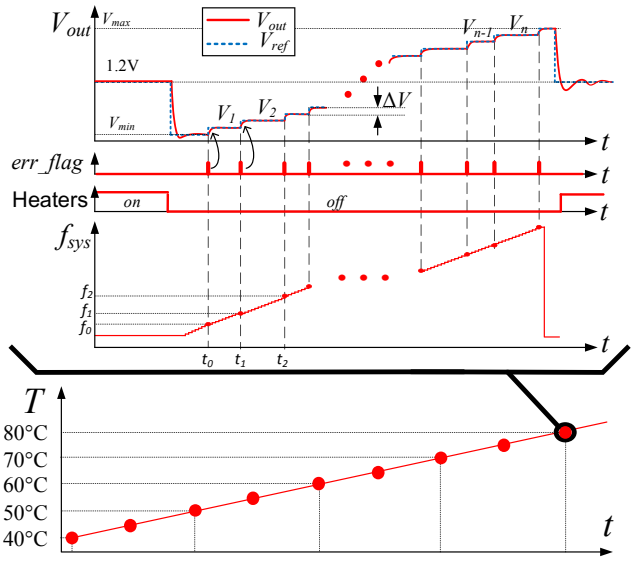


Fig. 8. Ideal waveform for the entire self-calibration process and detailed waveform at each temperature point.

f_{max}) by the error-checking blocks. Once an error is detected, V_{out} is increased by ΔV = 8 mV until V_{out} = V_{max} = 1.304 V at the end of the sweep. Since a higher voltage always allows for a higher frequency, the frequency range only needs to be swept once with this method. Note that usually the FPGA should not be running higher than nominal voltage; we include results for $V_{out} > 1.2V$ here only to demonstrate the full potential of the FPGA. In between sweeps, V_{out} is set to 1.2 V. To minimize the on-chip voltage drop during timing measurement, the calibration control circuitry turns the heaters off briefly as it determines the maximum safe frequency for each voltage and temperature pair. Hence the FPGA has a very low current consumption as the failing frequencies are measured (ie: below 100 mA for the Cyclone IV FPGA used in this research) and the FPGA core voltage, V_{core} is considered the same as the converter output voltage V_{out}.

The IR-drop is not negligible when a full application is running when the current consumption is high. The set-point of the dc-dc converter must therefore be dynamically adjusted during operation to account for the voltage drop from V_{out} to V_{core}. The converter output voltage, $V_{out}(t)$, should be set as the sum of the core voltage, $V_{core}(t)$, from the CT, and the compensating voltage, $V_{comp}(t)$, to account for the IR-drop, such that the FPGA core is powered with the desired voltage from the CT:

$$V_{out}(t) = V_{core}(t) + V_{comp}(t). \quad (2)$$

The compensating voltage, $V_{comp}(t)$, can be calculated as:

$$V_{comp}(t) = R_{V2D} \times I_{peak}(t), \quad (3)$$

while the resistance from the VRM feedback node to the FPGA die, R_{V2D}, can be characterized by the technique of [27]. The peak current command, $I_{peak}(t)$, which is directly available in the digital controller, is used as an estimate for the load current to calculate $V_{comp}(t)$. The use of $I_{peak}(t)$ adds some additional margin, compared to using the average load current.

The duration of the full calibration process is limited by the system's thermal response time, which is considerably longer than the dc-dc converter dynamics. For each calibration, one sweep of frequency and voltage takes less than 150 ms, while the entire temperature sweep takes approximately 1 minute. The calibration time can be greatly reduced by limiting the temperature, voltage range and other calibration parameters depending on application needs.

IV. Experimental Results

The automated self-calibration process is demonstrated using two common applications: a dual-channel 51-tap low-pass digital FIR filter representative of signal processing applications, and a CrossBar switch, which is commonly used in telecom applications for routing digital signals. The FIR filter occupies 72,802 out of 114,480 (64%) LEs, while the CrossBar occupies 30,208 out of 114,480 (26%) LEs. The dc-dc and DVS control blocks added by our approach occupy less than 1% of the total LEs. The package temperature ranges from 35°C - 80°C for V_{out} from 0.896-1.304 V. The experiment setup is shown in Fig. 9. The power-stage supplying V_{core} on the DE2-115 is disconnected, and the customized dc-dc converter is mounted on top of the FPGA, while its output, V_{out}, is connected to the decoupling capacitors on the DE2-115 board through vias with a short path to supply V_{core}. Since the environmental conditions are unlikely to change rapidly, case temperature is acceptable as an estimate of the die temperature in this case. Note that more advanced FPGAs have an on-die Temperature-Sensing Diode (TSD), which eliminates this issue. High-end FPGAs cost a few thousands dollars each, so considering the risk in the debugging and testing, only the 60-nm FPGA is tested in this paper.

To capture the actual failing frequency of an application running on a specific chip, we need to measure the delay of the speed-limiting path on that chip. Due to process variation and changes in operating conditions, the speed-liming path could be different on different chips and could also change due to operating conditions. For this reason, it is necessary to extract and test all paths whose delay is within a specified margin of the most critical path. In this work, the top 500 critical paths are extracted to prove that our offline calibration scheme can handle characterization of many paths, and 500 paths was more than sufficient for robust calibration of these designs. A formal approach to selecting a certain set of paths to optimize the probability of critical path coverage, while minimizing the number of required configuration bit-streams is outside the scope of this paper, but is part of our future work.

The on-chip configuration of the self-calibration and the FIR filter application are shown in Fig. 10(a) and (b), respectively. The 500 most critical paths are extracted and highlighted in red and black. Many paths have overlap with other critical paths, so the 500 paths are separated into 26 groups to perform error checking group by group, so each path can be sensitized and tested individually. To account for the influences of the converter output voltage ripple and switching noise, at least

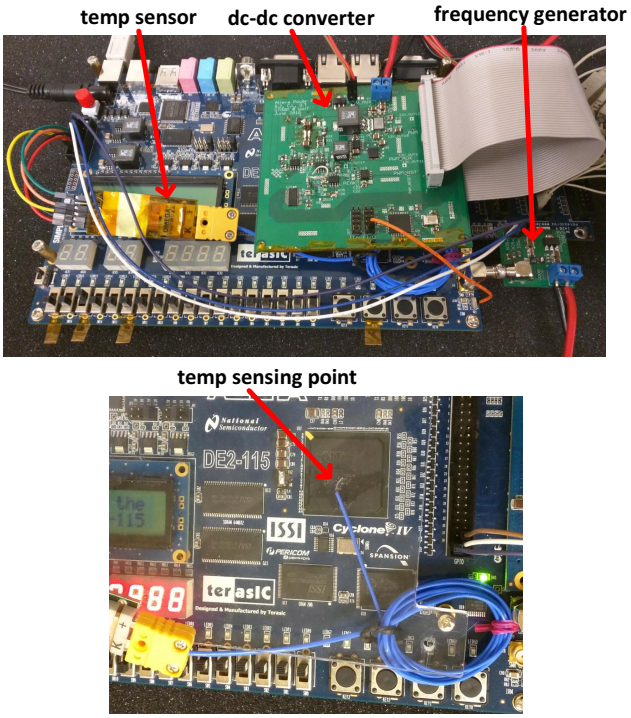
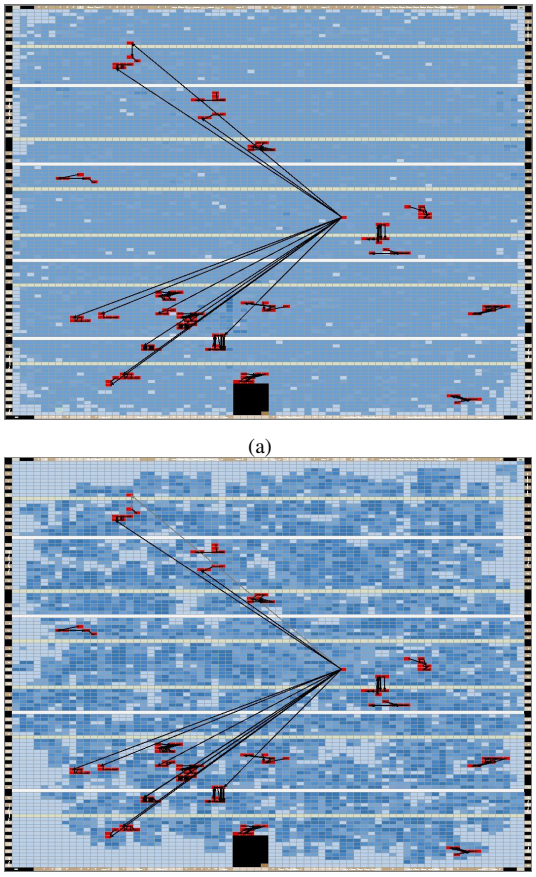


Fig. 9. Experimental setup.



(a)

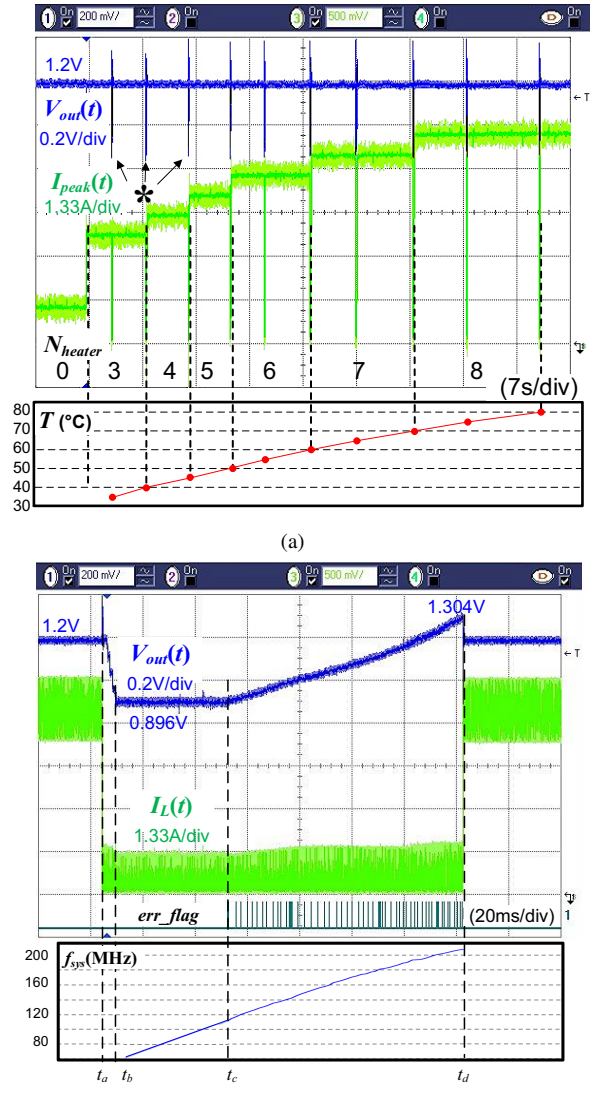
(b)

Fig. 10. The FPGA Chip Planner view of (a) self-calibration configuration and (b) application configuration. The red boxes are the used LEs in the critical paths, and the black line connects the relevant LEs in a path.

one converter switching period is monitored for the error checking of each group.

A. Self-calibration for CT

The entire calibration process of the FIR filter application is shown in Fig. 11(a): each voltage spike (as noted by "⋆") corresponds to one full sweep of f_{max} versus V_{out} at the given temperature. One such sweep is shown in Fig. 11(b), which reveals the converter dynamics.



(a)

(b)

Fig. 11. (a) An entire self-calibration process from 35°C to 80°C. N_{heater} is dynamically controlled to achieve an approximately linear rise in temperature, T. (b) One sweep of frequency, f_{sys}, and output voltage, V_{out}, at 75°C.

During self-heating, V_{out} is held at 1.2 V and then ramped down to 896 mV when the target temperature is reached. All the heater circuits are turned *off* at this point to guarantee minimum current draw on chip, which minimizes the internal voltage drop and therefore V_{core} is considered equal to V_{out}. f_{sys} is increased until the failing indicator, err_flag, goes high at which time the frequency is stored with the corresponding core voltage V_{core} in the CT, $V_{ref}[n]$ is then increased by 8 mV and the process repeats.

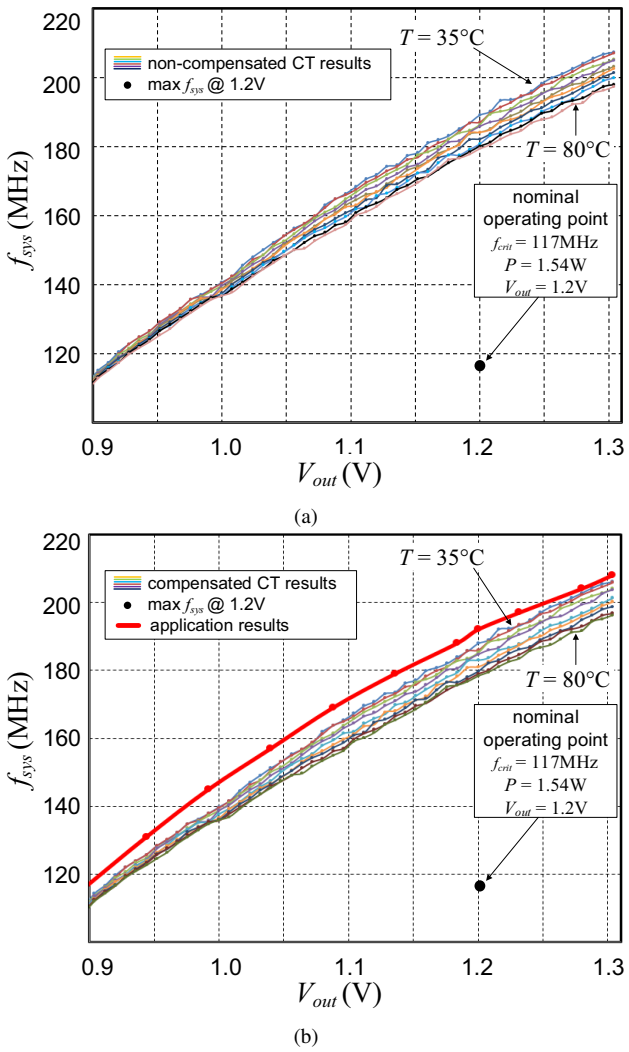


Fig. 12. (a) Extracted calibration data of FIR filter. (b) Calibration data with 0.6% fan-out compensation.

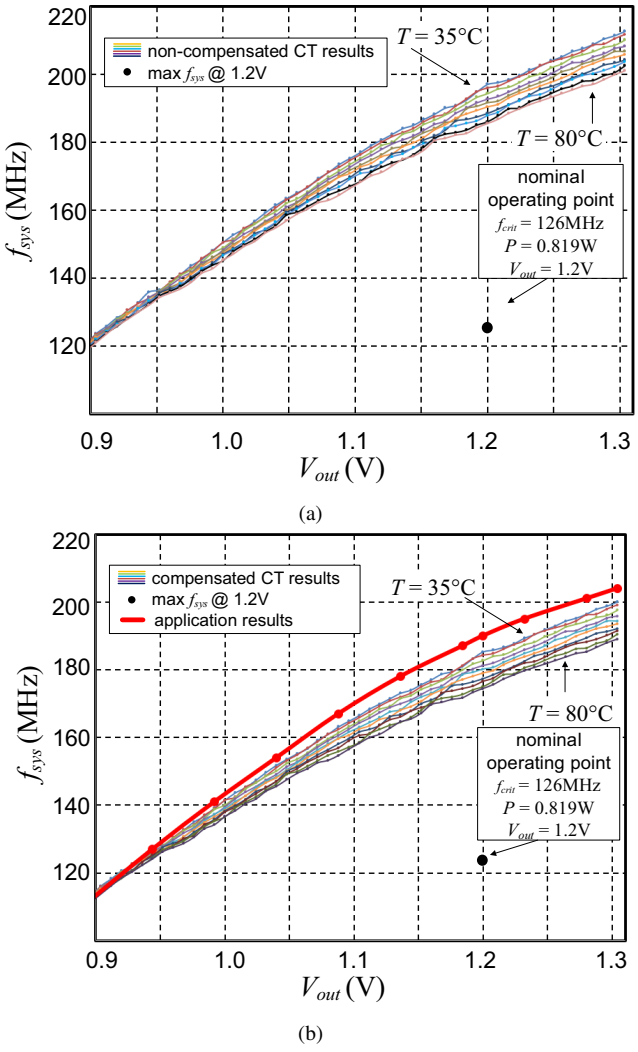


Fig. 13. (a) Extracted calibration data of CrossBar switch. (b) Calibration data with 6% fan-out compensation.

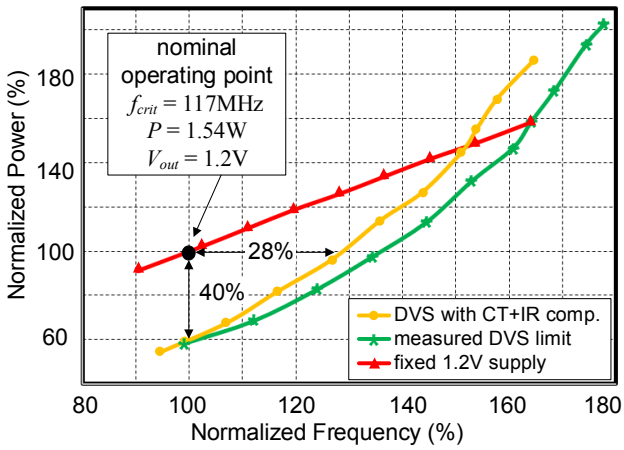
The raw CT data of f_{max} versus V_{core} versus T for the FIR filter and CrossBar switch are plotted in Fig. 12(a) and Fig. 13(a), respectively, with one curve per 5°C temperature increment. The effect of temperature is more noticeable at higher V_{core}. For example, at V_{core} = 1.304 V, f_{max} drops by 4.81% over a temperature range of 45°C.

As mentioned in Section II, *FRoC* does not model the complete fan-outs of replicated paths at this stage; therefore a correction factor is added to compensate for the missing fan-outs as their load does impact the path delay somewhat. For the FIR filter and the CrossBar switch, 0.6% and 6% compensation factors were added to the extracted CT values respectively, as shown in Fig. 12(b) and Fig. 13(b). The compensated CT is used for the DVS, and the CT refers to compensated CT in this paper unless noted specially.

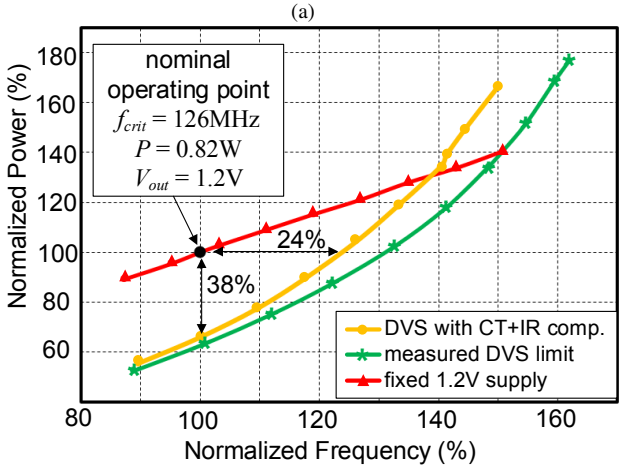
In order to check the accuracy of the CT data, which is generated from the calibration configuration (ie: Fig. 5(a)), the maximum clock frequency of the full FIR/CrossBar application (ie: in Fig. 5(b)) was independently checked using a random data generator and error checking. The result is shown as red line in Fig. 12(b) and Fig. 13(b). The calibration results are more conservative than the tested application, since the random input data is not guaranteed to exercise the worst-case critical paths.

The measured power consumption of the application is shown in Fig. 14(a) and (b), with constant voltage (thick red line). The frequency axis is normalized to the maximum value specified by the timing analysis of the CAD tool, f_{crit} (ie: the best available data for designers currently). The green line corresponds to the measured minimum DVS operation power with the given input data; it is manually measured from the two applications at their thermal steady state.

Several key points can be drawn from the data in Figs. 12-14: (1) even with V_{core} fixed at 1.2 V, the application circuit can operate up to 65% above f_{crit}. This shows that the CAD tool timing is necessarily conservative as expected, since it must account for worst-case temperature and process variations; (2) using DVS enables ~40% power savings at f_{crit} in the two tested applications; (3) for the same power consumption, DVS enables ~25% increase in the clock frequency; (4) the application failing frequencies are higher than the calibration results, which is a desired behavior as there are no guarantees that the random input stimulus exercises the applications' critical paths.



(a)



(b)

Fig. 14. Power saving comparison between the proposed method and the measured limit for (a) FIR filter, and (b) CrossBar switch.

B. IR-Drop Compensation

As described at the end of Section III, the output voltage, V_{out}, is assumed to be equal to core voltage, V_{core}, during self-calibration. However the IR-drop is not negligible when a full application is running when the current consumption is high. The compensating voltage, $V_{comp}(t)$, can be calculated as: $V_{comp}(t) = R_{V2D} \times I_{peak}(t)$. The measured resistance from the VRM feedback node to the FPGA die, R_{V2D}, is 16 mΩ according to [27]. The compensation voltage, $V_{comp}(t)$, is incorporated in the dc-dc controller to account for the IR-drop. Note that some advanced FPGAs have remote voltage sensing pins, which directly gives the core voltage. If the remote voltage sensing pins are used for voltage feedback, the IR-drop compensation should be revised accordingly.

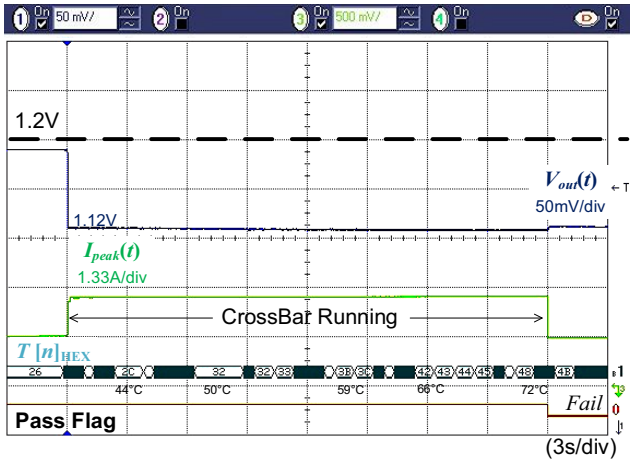
C. Online DVS for Two FPGA Applications

The two tested applications operate at 180 MHz, which corresponds to around 1.2 V core voltage in the CT, as shown in Table. I and Table. II.

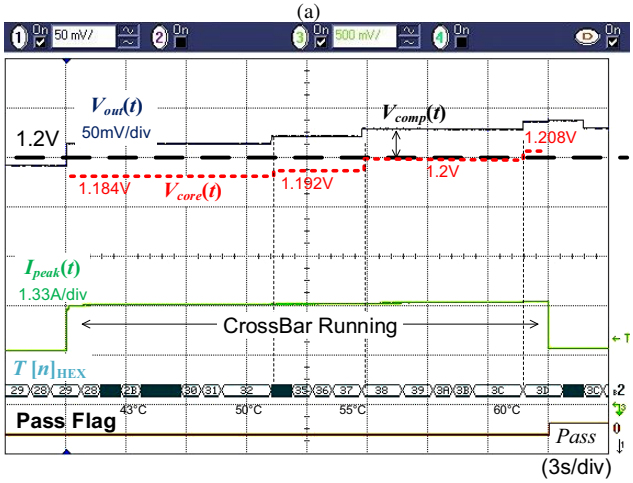
The CT data for the CrossBar application running at 180 MHz is listed in Table. I. In Fig. 15(a), the CrossBar application operates at 35°C CT without any compensation. In Fig. 15(b), the CrossBar application operates with the proposed DVS scheme. The application's error verification test only passes if the DVS follows the CT with thermal and IR-drop compensation.

TABLE I
CROSSBAR APPLICATION OPERATING CT @ 180 MHZ

Temperature, T (°C)	Core Voltage, V_{core} (V)	Failing Frequency, f_{crit} (MHz)
35	1.184	180.93
40	1.184	181.18
45	1.192	181.68
50	1.192	180.19
55	1.2	180.19
60	1.208	181.46
65	1.216	180.76
70	1.224	180.76
75	1.232	181.22
80	1.232	180.52



(a)

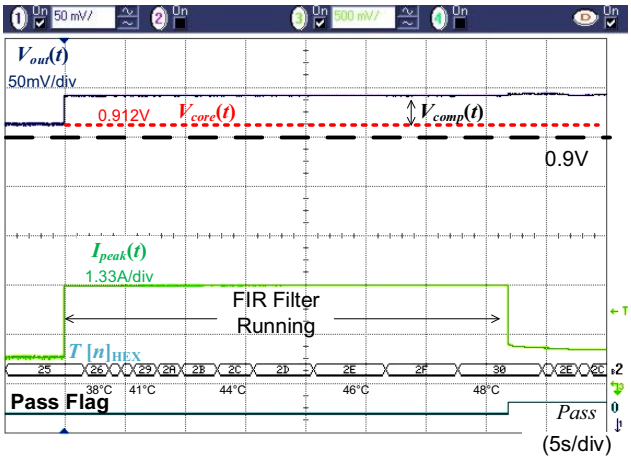


(b)

Fig. 15. (a) The CrossBar switch fails without IR-drop compensation and thermal feedback (V_{out} = 1.12V, f_{sys} = 180 MHz from non-compensated CT @ 35°C). (b) The CrossBar passes with thermal feedback and IR-drop compensated DVS at f_{sys} = 180 MHz. The inferred core voltage is shown in the dashed red line.

TABLE II
FIR FILTER OPERATING CT @ 180 MHZ

Temperature, T (°C)	Core Voltage, V_{core} (V)	Failing Frequency, f_{crit} (MHz)
35	1.16	180.99
40	1.168	180.99
45	1.176	182.17
50	1.176	181.22
55	1.184	181.93
60	1.184	180.52
65	1.192	181.93
70	1.2	180.76
75	1.2	180.06
80	1.208	180.52



(a)

(b)

Fig. 16. FIR Filter application running with thermal feedback and IR-drop compensated DVS at (a) nominal operating frequency, f_{crit} = 117 MHz and (b) f_{crit} = 180 MHz. In both cases the application runs successfully without logic errors, due to the real-time temperature and IR-drop compensation.

The CT data for the FIR filter application running at 180 MHz is listed in Table. II. In Fig. 16(a), the FIR filter runs at nominal operating frequency f_{sys} = 117 MHz, and in Fig. 16(b), the application operates at 180 MHz with DVS and IR-drop compensation. The results indicates that the application can either operate at nominal frequency with much lower core voltage, or over-clocked at 180 MHz around nominal core voltage. The tests start at FPGA idle temperature allowing temperature rise to demonstrate the dynamic voltage scaling at different temperature.

V. CONCLUSION

While DVS is widely adopted in microprocessors, it remains elusive in FPGAs mainly due to the fundamental challenge of application-dependent critical paths. The proposed three-step DVS technique can be universally applied to any user application, has a very low resource overhead and gives a very low probability of logic errors during operation. This procedure is fast enough to be applied automatically at board burn-in/test time, or possibly even at each board power-up. The main overhead is 1) additional start-up time, which depends on the thermal response of the system and 2) additional Flash memory to store multiple configuration bit-streams. The technique allows FPGA designers to safely operate each FPGA at its optimal performance point, reaching power savings on the order of 40%. The experiment demonstrates that real-time IR-drop and thermal compensation are required to achieve error-free DVS operation.

VI. ACKNOWLEDGEMENT

This work was supported by Intel Corporation, the Ontario Centres of Excellence, the Natural Sciences and Engineering Research Council of Canada, the Canadian Foundation for Innovation and the Ontario Research Fund. The authors thank Mark Bourgeault and Gurvinder Tiwana for their valuable insights on DVS in FPGA applications.

REFERENCES

[1] B. Sukhwani, H. Min, M. Thoennes, P. Dube, B. Brezzo, S. Asaad, and D. Dillenberger, "Database Analytics: A Reconfigurable-Computing Approach," *Micro, IEEE*, vol. 34, no. 1, pp. 19–29, Jan 2014.

[2] M. Lavasani, H. Angepat, and D. Chiou, "An FPGA-based In-Line Accelerator for Memcached," *Computer Architecture Letters*, vol. 13, no. 2, pp. 57–60, July 2014.

[3] A. Putnam, A. Caulfield, E. Chung, D. Chiou, K. Constantinides, J. Demme, H. Esmaeilzadeh, J. Fowers, G. Gopal, J. Gray, M. Haselman, S. Hauck, S. Heil, A. Hormati, J.-Y. Kim, S. Lanka, J. Larus, E. Peterson, S. Pope, A. Smith, J. Thong, P. Xiao, and D. Burger, "A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services," *Micro, IEEE*, vol. 35, no. 3, pp. 10–22, May 2015.

[4] I. Kuon and J. Rose, "Measuring the Gap Between FPGAs and ASICs," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 2, pp. 203–215, Feb 2007.

[5] E. Mintarno, J. Skaf, R. Zheng, J. Velamala, Y. Cao, S. Boyd, R. Dutton, and S. Mitra, "Self-Tuning for Maximized Lifetime Energy-Efficiency in the Presence of Circuit Aging," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 30, no. 5, pp. 760–773, May 2011.

[6] N. Gong, J. Wang, S. Jiang, and R. Sridhar, "TM-RF: Aging-Aware Power-Efficient Register File Design for Modern Microprocessors," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 23, no. 7, pp. 1196–1209, July 2015.

[7] K. Chapman and J. Hussein, "Lowering Power using the Voltage Identification Bit," 2012, available at http-s://www.xilinx.com/support/documentation /application_notes/xapp555-Lowering-Power-Using-VID-Bit.pdf.

[8] "AN 711: Power Reduction Features in Intel Arria 10 Devices," 2017, available at https://www.altera.com/content/dam/altera-www/global/en_US/pdfs /literature/an/an711.pdf.

[9] "White Paper: 40-nm FPGA Power Management and Advantages," 2008, available at https://www.altera.com/content/dam/altera-www/global/en_US/pdfs /literature/wp/wp-01059-stratix-iv-40nm-power-management.pdf.

[10] A. Kahng, S. Kang, R. Kumar, and J. Sartori, "Enhancing the Efficiency of Energy-Constrained DVFS Designs," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 21, no. 10, pp. 1769–1782, Oct 2013.

[11] M. Yadav, M. Casu, and M. Zamboni, "LAURA-NoC: Local Automatic Rate Adjustment in Network-on-Chips With a Simple DVFS," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 60, no. 10, pp. 647–651, Oct 2013.

[12] M. Gerards, J. Hurink, and J. Kuper, "On the Interplay between Global DVFS and Scheduling Tasks with Precedence Constraints," *Computers, IEEE Transactions on*, vol. 64, no. 6, pp. 1742–1754, June 2015.

[13] M. Al-Mothafar and K. Hammad, "Small-signal Modelling of Peak Current-Mode Controlled Buck-derived Circuits," *Electric Power Applications, IEE Proceedings* -, vol. 146, no. 6, pp. 607–619, Nov 1999.

[14] B. Das and H. Onodera, "Frequency-Independent Warning Detection Sequential for Dynamic Voltage and Frequency Scaling in ASICs," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 22, no. 12, pp. 2535–2548, Dec 2014.

[15] T. Lin, K.-S. Chong, J. Chang, and B.-H. Gwee, "An Ultra-Low Power Asynchronous-Logic In-Situ Self-Adaptive V_{DD} System for Wireless Sensor Networks," *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 2, pp. 573–586, Feb 2013.

[16] Y.-W. Ma, J.-L. Chen, C.-H. Chou, and S.-K. Lu, "A Power Saving Mechanism for Multimedia Streaming Services in Cloud Computing," *Systems Journal, IEEE*, vol. 8, no. 1, pp. 219–224, March 2014.

[17] I. Ahmed, S. Zhao, O. Trescases, and V. Betz, "Measure Twice and Cut Once: Robust Dynamic Voltage Scaling for FPGAs," in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, Aug 2016, pp. 1–11.

[18] C. Chow, L. Tsui, P. Leong, W. Luk, and S. Wilton, "Dynamic Voltage Scaling for Commercial FPGAs," in *Field-Programmable Technology, 2005. Proceedings. 2005 IEEE International Conference on*, Dec 2005, pp. 173–180.

[19] O. Trescases and J. Ng, "Variable Output, Soft-switching DC/DC Converter for VLSI Dynamic Voltage Scaling Power Supply Applications," in *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*, vol. 6, June 2004, pp. 4149–4155 Vol.6.

[20] J. Levine, E. Stott, G. Constantinides, and P. Cheung, "Online Measurement of Timing in Circuits: For Health Monitoring and Dynamic Voltage & Frequency Scaling," in *Field-Programmable Custom Computing Machines (FCCM), 2012 IEEE 20th Annual International Symposium on*, April 2012, pp. 109–116.

[21] J. M. Levine, E. Stott, G. Constantinides, and P. Y. Cheung, "SMI: Slack Measurement Insertion for Online Timing Monitoring in FPGAs," *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on*, pp. 1–4, Sept 2013.

[22] J. M. Levine, E. Stott, and P. Y. Cheung, "Dynamic Voltage & Frequency Scaling with Online Slack Measurement," in *Proceedings of the 2014 ACM/SIGDA International Symposium on Field-programmable Gate Arrays*, ser. FPGA '14. New York, NY, USA: ACM, 2014, pp. 65–74. [Online]. Available: http://doi.acm.org/10.1145/2554688.2554784

[23] J. L. Nunez-Yanez, "Adaptive Voltage Scaling with In-Situ Detectors in Commercial FPGAs," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 45–53, Jan 2015.

[24] S. Das, C. Tokunaga, S. Pant, W. H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, "RazorII: In Situ Error Detection and Correction for PVT and SER Tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, Jan 2009.

[25] M. Wirnshofer, L. H. A. N. Kakade, N. P. Aryan, G. Georgakos, and D. Schmitt-Landsiedel, "Adaptive voltage scaling by in-situ delay monitoring for an image processing circuit," in *2012 IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)*, April 2012, pp. 205–208.

[26] S. Zhao, I. Ahmed, C. Lamoureux, A. Lotfi, V. Betz, and O. Trescases, "A universal self-calibrating Dynamic Voltage and Frequency Scaling (DVFS) scheme with thermal compensation for energy savings in FPGAs," in *2016 IEEE Applied Power Electronics Conference and Exposition (APEC)*, March 2016, pp. 1882–1887.

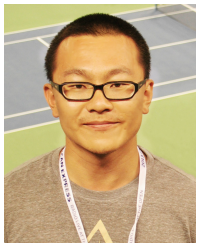
[27] S. Zhao, I. Ahmed, A. Khakpour, V. Betz, and O. Trescases, "A Robust Dynamic Voltage Scaling Scheme for FPGAs with IR Drop Compensation," in *2017 IEEE Applied Power Electronics Conference and Exposition (APEC)*, March 2017, pp. 2939–2944.

[28] D. You and K. S. Chung, "Quality of Service-Aware Dynamic Voltage and Frequency Scaling for Embedded GPUs," *IEEE Computer Architecture Letters*, vol. 14, no. 1, pp. 66–69, Jan 2015.

[29] X. Lin, Y. Wang, N. Chang, and M. Pedram, "Concurrent Task Scheduling and Dynamic Voltage and Frequency Scaling in a Real-Time Embedded System With Energy Harvesting," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 11, pp. 1890–1902, Nov 2016.

[30] X. Lin, Y. Wang, Q. Xie, and M. Pedram, "Task Scheduling with Dynamic Voltage and Frequency Scaling for Energy Minimization in the Mobile Cloud Computing Environment," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 175–186, March 2015.

[31] "White Paper: Guaranteeing Silicon Performance with FPGA Timing Models," 2010, available at http-s://www.altera.com/en_US/pdfs/literature/wp/wp- 01139-timing-model.pdf.



Shuze Zhao received the B.S. degree in electrical engineering from Zhejiang University, Hangzhou, China, and M.A.Sc. degree in electrical engineering from University of Toronto, Ontario, Canada, in 2011 and 2013 respectively. He is currently a Ph.D. candidate at University of Toronto. His current research interests include resonant dc-dc converters, digital control of power electronics and design of mixed-signal integrated power converters.

Ibrahim Ahmed received his Bachelor of Science in electronics engineering from the German University in Cairo in 2012. He received his Masters of Science in information technology and embedded systems from the University of Stuttgart in 2014. Since 2015, he has been a Ph.D. student at the University of Toronto.

Carl Lamoureux received the B.S. degree in Electrical Engineering from the University of Toronto, Ontario, Canada, in 2017. He is currently working toward the M.A.Sc degree in Electrical Engineering in the Energy Systems Group at the University of Toronto. His current research interests include power converters for electric vehicles and photovoltaic systems.

Ashraf Lotfi is VP / Fellow at Intel serving as CTO of the power products business organization. Prior to Intel he was at Altera as Power CTO serving its Enpirion power business as well as the broader FPGA R&D community. Prior to that he served as President & CEO of Enpirion, which he founded in 2002. In 2013 he drove Enpirions merger into Altera to realize his vision of highly integrated power management closely coupled to leading-edge digital loads. Since Enpirions inception, he led its strategic direction with a unique industry-first vision to create the ultimate power converter-on-chip to create ubiquitous DC-DC conversion at the silicon level. He created

Enpirion from the ground-up, raising all its capital needs, developing its innovative technology, deploying its products, winning its customers and growing its profitable sales. Prior to founding Enpirion, he was Director of Advanced Power Research at Bell Laboratories, Murray Hill, NJ. He received the BSEE from Cairo University in 1984, MSEE & PhD from Virginia Tech in 1993 and executive MBA from Northeastern University in 2001 and is a Senior Member of the IEEE.

Vaughn Betz (S'88–M'91–SM'17) co-founded Right Track CAD to develop new FPGA architectures and CAD tools and was its VP of Engineering until its acquisition by Altera in 2000. He was at Altera from 2000 to 2011, ultimately as Senior Director of Software Engineering, and is one of the architects of both the Quartus II CAD system and the Stratix I - V and Cyclone I - V FPGAs. He is a Professor and the NSERC/Intel Industrial Research Chair in Programmable Silicon at the University of Toronto, where his research covers FPGA architecture, CAD, and acceleration of computation using FPGAs.

Olivier Trescases received his Ph.D. from the University of Toronto in 2007. From 2007 to 2008, he worked as a concept engineer and mixed-signal IC designer at Infineon Technologies in Austria, focussing on safety-critical automotive applications. Dr. Trescases is currently an associate professor in the Edward S. Rogers Sr. Department of Electrical and Computer Engineering at the University of Toronto. His group conducts research on advanced power electronic converters. Since 2015 he is an Associate Editor of the IEEE Transactions on Power Electronics. While on sabbatical in 2016 Prof. Trescases worked at the Texas Instruments Kilby Labs in Santa Clara and then at NXP Semiconductor. Since 2016 he is the Vice Chair of the IEEE Toronto Section and has served on various IEEE conference technical committees, including the Applied Power Electronics Conference (APEC), the Custom Integrated Circuits Conference (CICC), and the International Symposium on Power Semiconductors and ICs (ISPSD).