An Adapative Frequency Control Method using Thermal Feedback for Reconfigurable Hardware Applications

Phillip H. Jones #1, Young H. Cho #2, John W. Lockwood #3

Applied Research Laboratory
Washington University
St. Louis, MO, USA
[1]phjones@arl.wustl.edu
[2]younghc@gmail.com [3]lockwood@arl.wustl.edu

***Abstract*— Reconfigurable circuits running in Field Programmable Gate Arrays (FPGAs) can be dynamically optimized for power based on computational requirements and thermal conditions of the environment. In the past, FPGA circuits were typically small and operated at a low frequency. Few users were concerned about high-power consumption and the heat generated by FPGA devices. The current generation of FPGAs, however, use extensive pipelining techniques to achieve high data processing rates and dense layouts that can generate significant amounts of heat. FPGA circuits can be synthesized that can generate more heat than the package can dissipate. For FPGAs that operate in controlled environments, heatsinks and fans can be mounted to the device to extract heat from the device. When FPGA devices do not operate in a controlled environment, however, changes to ambient temperature due to factors such as the failure of a fan or a reconfiguration of bitfile running on the device can drastically change the operating conditions. A protection mechanism is needed to ensure the proper operation of the FPGA circuits when such a change occurs. To address these issues, we have devised a reconfigurable temperature monitoring system that gives feedback to the FPGA circuit using the measured junction temperature of the device. Using this feedback, we designed a novel dual frequency switching system that allows the FPGA circuits to maintain the highest level of performance for a given maximum junction temperature. Our working system has been implemented and deployed on the Field Programmable Port Extender (FPX) platform at Washington University in St. Louis. Our experimental results with a scalable image correlation circuit show up to a 2.4x factor increase in performance as compared to a system without thermal feedback. Our circuit ensures that the device performs the maximum required computation while always operating within a safe temperature range.**

I. Introduction

Many applications are exposed to multiple thermal conditions during their operational lifetime. Mobile systems, such as military and space applications, require high performance computation in embedded systems that move rapidly between different environments. Stationary systems, such as outdoor surveillance systems, must adapt to variable ambient temperatures. Even systems that operate in tightly controlled environments, such as rack-mounted FPGA computational blades in a machine room, must adapt to variable thermal environments so that they will not completely fail due to a fault in a fan or obstruction of air flow. In general, all reconfigurable devices can find themselves exposed to conditions much different then their typical operating conditions. In these cases, it is desirable to allow the circuit to adapt to the environment.

Most existing FPGA circuits operate at a fixed operating frequency. At this frequency, the heat dissipation mechanisms are built to handle worst-case operating conditions. When there is a significant gap between the worst-case operating condition and the typical operating condition, the system must be over-engineered and/or the performance realized by the system may be significantly less than optimal during typical operating conditions.

This work presents an adaptive frequency control method that uses thermal feedback to adjust the operating speed of a reconfigurable system. We quantify how much computation performance can be gained by using this method instead of operating the device only at a fixed frequency.

A. Motivation

While testing high performance circuits on the FPX development platform, we experienced an incident that overheated an FPGA [1]. Given unfavorable environmental conditions, an FPX platform was damaged because the bitfile generated more heat than the package could dissipate given the amount of airflow available in an open chassis. In order to prevent such an event from occurring in the future, we designed a temperature monitoring circuit that runs on another FPGA on the FPX platform that acts like a thermal circuit breaker [1]. The FPX platform now provides a mechanism to monitor the temperature of the reconfigurable device over the network and provides a mechanism which can dynamically adjust the operation of the reconfigurable logic device.

During our characterization of the FPGA thermal behavior, we discovered that we had an opportunity to make use of the relatively fast measurements of junction temperature changes verses the relatively slow rate of change of temperature of the system due to thermal mass of the package and heatsink. A relatively large amount of time is available to operate a circuit at a high frequency while the package slowly warms as

compared to the period at which the FPX platform performs computation on data packets. Seeing this as an opportunity to improve the performance of our reconfigurable hardware platform in transient conditions, we devise a novel scheme that dynamically adjusts the operation of the reconfigurable logic device between two clock frequencies using temperature thresholds. This mechanism generates a thermally-adaptive frequency that maximizes the computational throughput for a specified maximum application temperature which we refer to in this paper as the application's *thermal budget*. As stated previously, there are several types of applications that benefit from using dynamic adjustment of frequency as compared to a fixed operating frequency.

B. Contribution

In the following section, we discuss related academic work and industry solutions related to thermal management. Section III gives a summary of the previous work that we used to build upon in this paper. The two main contributions of the previous work was (1) the implementation of a thermal shutdown circuit for applications implemented on FPGAs, and (2) a systematic approach for thermal profiling reconfigurable hardware. The contributions of this paper are detailed in Sections IV and V. In Section IV a novel implementation for temperature driven frequency control of reconfigurable hardware is described. In Section V, we quantify the results of the circuit implemented for a case study of a a high-power image-correlation application.

II. Related Work

Research on Dynamic Thermal Management (DTM) systems is related to this work. Microprocessors have been built that allow their voltage and frequency to be scaled to extend battery life of mobile computers. Companies like Intel and AMD extend this concept to manage heat dissipation on servers [2]. By introducing power management features, software running on the CPU can scale voltage and frequency to lower power usage before the device overheats. Such technology is critical for servers located in large data centers that house hundreds or thousands of computation nodes.

Low-power embedded processors like Xscale [3] have hooks that allow voltage and frequency scaling to increase power and thermal efficiency. Work presented by [4] makes use of these features to present a dynamic thermal management (DTM) system that would scale the processor frequency in response to temperature readings from an external thermal-couple.

The Advanced Configuration and Power Interface (ACPI) standard codeveloped by Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba, defines an interface for software to perform power management on devices within a system [5]. ACPI targets systems that have an OS or significant management firmware, not reconfigurable applications that run using little or no software.

There has also been work for FPGAs that use feedback to implement DTM mechanisms. In [6] a dynamic voltage scaling (DVS) mechanism was presented that used gate delay feedback to minimize the voltage supplied to internal FPGA logic.

Typically DTM strategies are implemented for the purpose of conserving power. In the case of mobile computers this extends the battery life. For servers in data centers, lower power consumption saves energy and provides a lower cost of ownership for the system. When excessive heat is produced, systems with temperature-controlled fans increase the airflow to cool the device. The work presented in this paper looks at DTM from a different point of view. The goal of our approach is to run reconfigurable circuits so as to achieve maximum possible performance for a given temperature. Our approach adapts the frequency of the application to maintain a specified temperature even as the ambient environment changes.

III. Reconfigurable Hardware Temperature Monitor

We start this section with an overview of the development platform used for this work. We then summarize our previous work on which this work is built upon.

A. Development Platform

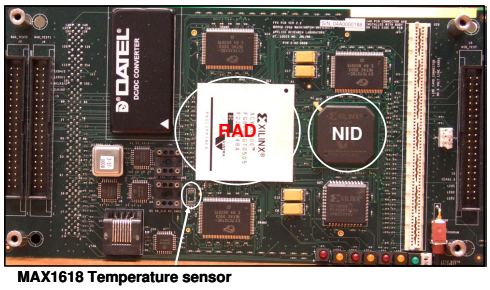

Fig. 1. FPX Development Platform

The circuits described in this paper were implemented on the FPX platform, shown in Figure 1. The FPX platform contains two FPGAs: (1) a small Xilinx Virtex FPGA called the *Network Interface Device (NID)* is configured with a static bitfile, and (2) a large Xilinx Virtex FPGA called the *Reconfigurable Application Device (RAD)* is reconfigured with bitfiles loaded dynamically over a network [7]. New bitfiles that implement modular data processing functions are sent to the NID over the network within a bitfile that is used to reconfigure the RAD [8]. The FPX uses an on-board Maxim temperature measurement device (MAX1618) to digitally sample the temperature of the RAD.

B. Thermal Shutdown Circuit

Figure 2 shows the side-view of an FPX platform that was damaged by a bitfile running on the RAD that consumed more power than the platform could dissipate in a chassis with insufficient airflow to cool the system. The circuit board warped and caused a short-circuit between power planes. The excess current through the power pins burned the connector, as can be seen in the photo. Motivated by the need to prevent such a high-powered application from damaging another FPX,

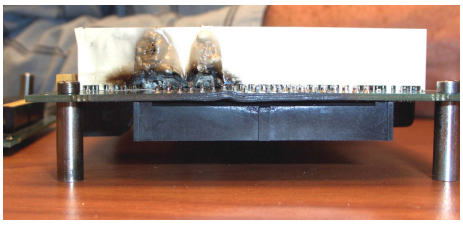

Fig. 2. FPX side view: Note how layers of the board were warped because of heat

a thermal monitor and shutdown circuit was implemented. The circuit allows the NID to monitor the junction temperature of the RAD. If an application causes the junction temperature of the RAD to surpasses a programmable maximum threshold, then the NID acts as a circuit breaker to unload and reconfigure the high-power bitfile from the device.

C. Temperature Monitor and Thermal Shutdown Circuit Implementation on the FPX

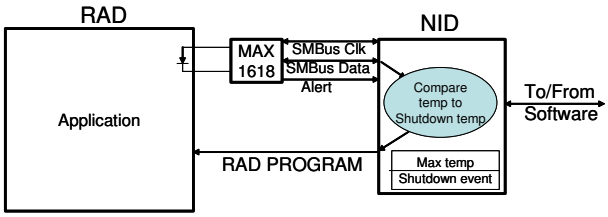

Fig. 3. Shutdown Circuit Architecture

Figure 3 illustrates how the temperature monitor and shutdown circuit is mapped onto the FPX. The thermal shutdown circuit was implemented using logic on the NID to prevent an applications deployed on the RAD from exceeding a safe operating temperature. The NID interfaced to a MAX1618, a Maxim temperature monitor chip that measures the junction temperature using a sense diode embedded in the silicon of the RAD. The NID samples the MAX1618 and compares the temperature received from this device to a user-programmable maximum temperature threshold. If the preset threshold is surpassed, the NID shuts down the application deployed on the RAD by sending a command through the SelectMAP interface of the RAD to clear the configuration memory.

The temperature of the RAD can also be monitored externally by sending a query message over the network to the NID. The NID responds with a status message that reports the temperature of the RAD. We wrote software to log the temperature of the RAD while running custom-designed thermal benchmark circuits. [1].

Figure 4 shows a plot of temperature over time for a circuit that is shutdown due to exceeding a set thermal threshold of 70 C.[1]. Section IV discusses how this temperature monitor and shutdown circuit was extended to implement adaptive frequency control of applications deployed on the RAD.

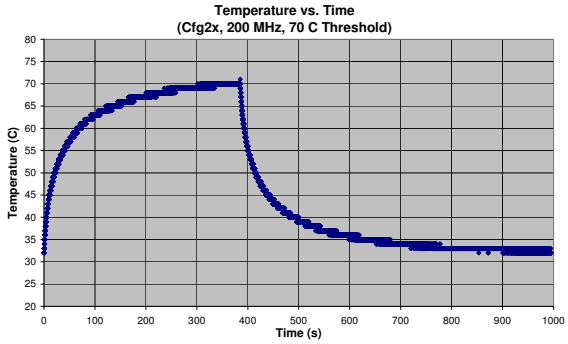

Fig. 4. FPGA being reprogrammed by the Shutdown Circuit upon reaching a thermal threshold of 70 C

IV. Temperature Driven Frequency Control

This section begins with a discussion of the types of applications that benefit from a thermally-adaptive frequency management circuit. Next, we describe the operation of the frequency controller that uses thermal feedback. This section concludes with a description of how the adaptive frequency mechanism operates on the FPX platform.

A. Target Applications

Reconfigurable systems with certain characteristics benefit most from the use of adaptive frequency control using thermal feedback. First, systems deployed in environments where the temperature changes benefit by allowing the circuit to adapt their performance. Second, systems that have multiple modes of operation that impact their thermal output benefit from adaptive thermal control. Third, systems that have bursty computation with demands for low latency benefit by allowing the device to temporarily operate at frequencies faster than would be allowed in steady-state.

The more variation that exists in the range of possible thermal conditions over which a circuit must operate, the greater the benefit of using adaptive control. A design that uses a fixed frequency must operate at a speed that will not cause the application to exceed a preset thermal budget even when operating under worst-case conditions. If there exists a large difference between the worst-case and best-case thermal conditions, a conservative frequency must be used to prevent damage to the device or a shutdown mechanism must be used to safeguard the system. The use of a thermally-adaptive frequency allows the system to operate close the optimal frequency for all thermal conditions.

Dynamic adjustment of frequency is especially useful for systems that have bursty modes of operation. During idle periods the system replenishes its available thermal budget. When a burst of data arrives the device can operate faster for a short amount of time, allowing applications such as an Internet routing module to achieve lower latencies.

B. Adaptive Dual Frequency Switch

FPGAs available today from vendors such as Xilinx and Altera have Delay Lock Loops (DLLs) that can multiply

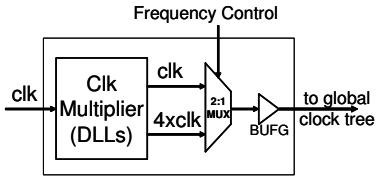

Fig. 5. Frequency Multiplexing circuit

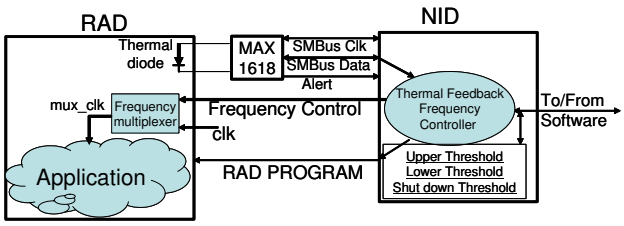

Fig. 6. Temperature Measurement and Threshold Frequency control mechanism

and divide a clock input signal. We use DLLs combined with a 2:1 multiplexor to switch between the base input clock and a clock that operates at 4x the base frequency. The select line of the 2:1 multiplexor is controlled by logic that monitors the application's temperature and implements a high and low temperature threshold control strategy. Figure 5 shows the architecture of the Frequency Multiplexing circuit. More elaborate techniques can and should be used to avoid clock glitches. For example a glitch free version of the 2:1 mux component can be implemented with the BUFGMUX component available for the Virtex-II [9] and later generations of Xilinx FPGAs.

C. Thermal Feedback Frequency Controller

A basic temperature control mechanism is implemented by controlling the input to the frequency selector using control logic based on thermal feedback. Application logic on the reconfigurable device operates using the 4x clock while the temperature remains below the upper threshold. Once the upper threshold is reached, the application circuit is given the base clock and allowed to cool down until the lower threshold is reached. At this point, the cycle repeats.

The main idea of this approach is to modulate the duty cycle at which the application runs with the faster (4x) clock. As the external thermal environment changes, the duty cycle will automatically adjust keeping the application temperature between the upper and lower bounds. By selecting thresholds appropriately and switching quickly between modes, the application can maintain a target average temperature within tight bounds.

The upper temperature threshold is the application thermal budget. The objective is to achieve maximum computational performance for a given thermal budget by adaptively adjusting the duty cycle as the thermal operating environment changes.

D. System Integration

The implementation of our thermally controlled adaptive frequency mechanism on the the FPX platform is shown in Figure 6. Our approach has two components: the frequency multiplexing circuit and the frequency control circuit. The functional description of these two parts is given in section IV-B and IV-C.

The frequency multiplexor resides in the RAD. This circuit uses the 4x clock multiplier circuit described in Xilinx XAPP174 [10]. We modified the circuit by replacing the 4x clock output with the output of a 2:1 multiplexor. The multiplexor uses the 4x clock as one input and the base clock as the other. The selector of the mux is managed by the frequency control circuit.

The frequency control circuit resides on the NID and is responsible for controlling the frequency multiplexor circuit. This circuit extends the thermal shutdown circuit described in section III-B. A state machine was developed to implement a temperature threshold controller. Configuration commands sent to the NID over the network set the upper and lower temperature threshold values. The thermal budget of the application is the value contained by the upper threshold.

NID status messages were augmented with fields that return the frequency at which the application is being driven in addition to the application temperature. This information was used in performance evaluation experiments, section V-C, to compute the effective frequency of the application under several thermal operating conditions.

V. Implementation

This section describes a computationally intensive FPGA application that is capable of exceeding the safe thermal limits of the FPGA package of 85 C. In section V-C, we use this application circuit in a case study to evaluate the effectiveness of our adaptive frequency approach.

A. Image Correlation Application

Image correlation is an application well-suited for hardware implementation. It is highly parallelizeable [11], [12]. The specific image correlation application we describe in this paper scans an input image for up to four different patterns. The circuit is inherently high-powered and cannot run at its maximum clock rate without thermal management or it overheats the FPGA.

Figure 7 shows the high-level architecture of the image correlation application. Note the core of the circuit is separate from the input and output module. The core uses the adaptive frequency technique to adapt to changes in the operating thermal environment, while the input and output modules use a fixed clock. This simplifies the interface to the external network interface components and memory modules. Asynchronous FIFOs are used to transfer data between the two different clock domains. The core logic of this application was used to evaluate the effectiveness of thermal frequency control. Instead of reading image data from external memory, signals from a block RAM and a Linear Feedback Shift Register

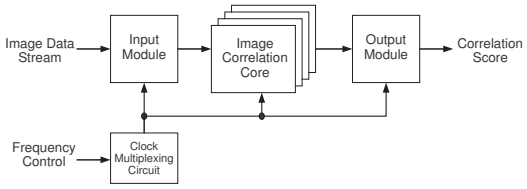

Fig. 7. High Level System Architecture

(LFSR) were used to produce pseudo-random data for the core to process. Results of synthesis and characteristics of the application are given in Figure 8.

VirtexE 2000 Resource Utilization

Lookup Tables (LUTs)	D Flip Flops (DFFs)	Occupied Slices	Block RAM	Max Frequency
72% (27,788)	64% (24,832)	82% (15,808)	26% (43)	125 MHz

a.)

Image Correlation Characteristics

Image Size (# pixels)	Pixel Resolution	# of Mask Patterns	# of Templates	Image Processing Rate
640x480	8-bit (grey scale)	1 - 4	10 (in parallel)	12.7/second (at 125 MHz)

b.)

Fig. 8. a.) FPGA Utilization, b.) Application Details

1) Algorithm: A 64x64 bit-mask pattern is scanned over incoming images. A score is computed for each possible offset of the mask. This score is the sum of the product of each bit of the mask with a corresponding pixel value. The algorithm for scanning an image can be represented by Equation 1:

$$\begin{aligned} U &= 0 \ to \ M \\ V &= 0 \ to \ N \\ S_{U,V} &= \sum_{R=0,C=0}^{R=y,C=x} I_{(U+C,V+R)} T_{(C,R)} \end{aligned} \tag{1}$$

Figure 9 helps to illustrate this equation. Template T scans image I from left to right and from top to bottom.

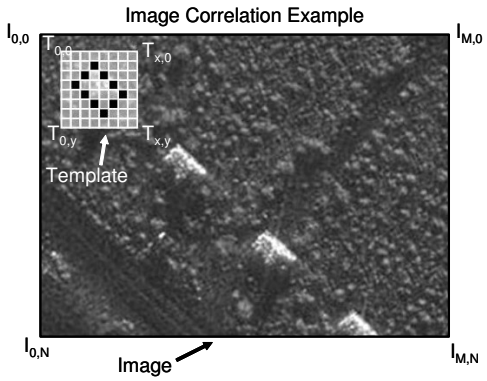

Fig. 9. Image Correlation Algorithm Example

2) Logic Layout: We found that the backend CAD tools had trouble placing and routing this high resource utilization application without manually controlled layout. To overcome this problem, we manually placed our circuit to compactly fit into the logic that implements the RAD (a VirtexE 2000E) which allowed us to operate the circuit at a high clock rate. Logic is placed on the FPGA as shown in Figure 10. Precise layout of the circuit elements on the FPGA was accomplished using the Relative Location (RLOC) attribute. Figure 10 illustrates how data flows through the application circuit. Data is shifted through the circuit in such a way that pixels move only short distances between Flip/Flops, thereby enabling the data path to run at a high clock frequency. Control logic (not shown) is pipelined to control the operation of the data path at a high clock rate. The layout helped to localize computation around local resources. This, in turn, allowed the backend tools to efficiently route components together.

Without the manual placement constraints, the Xilinx tools were not able to fit the circuit into the FPGA. With automatic placement, the MAP tool required 106% of the available slice resources, while with manual placement constraints, only uses 82% of the available slices.

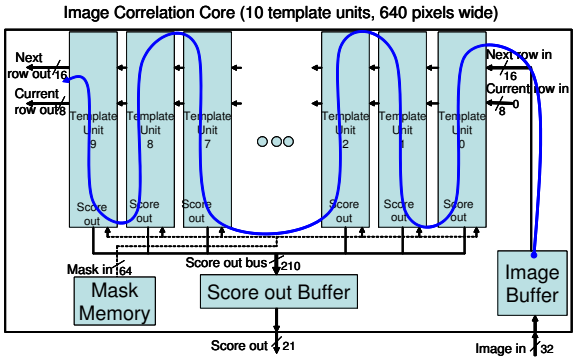

Fig. 10. Layout and Pixel data flow for Image Correlation core

B. Experimental Setup

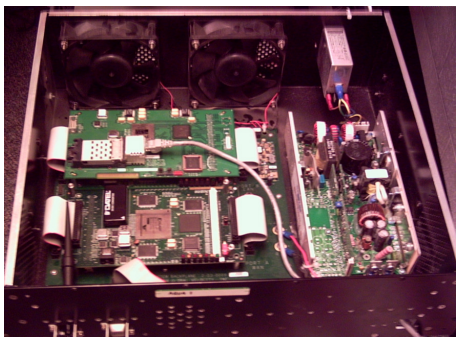

Fig. 11. Experimentation Platform

The image correlation application is deployed on the RAD of the FPX platform. The FPX was placed into a 3U rackmount case, as shown in Figure 11. The case was equipped with

2 fans that each supply approximately 250 Linear Feet per Minute (LFM) of air flow. The system has a removable case cover, which is not installed for the photo shown in Figure 11.

Figure 12 describes the five thermal conditions used to evaluate our adaptive frequency approach. First, for the worst case thermal operating condition (Scenario S1) a fixed frequency of 50 MHz was found to use up a 70 C thermal budget. Once this fixed frequency was determined, the application was run under the other four operating conditions to measure the unused portion of the thermal budget. Next, the thermal feedback adaptive frequency mechanism was deployed using the same five scenarios. Temperature thresholds were set to force the circuit to fully utilize the allocated thermal budget under all operating conditions. For each scenario, the effective operating frequency achieved by the adaptive frequency approach was measured and compared to the circuit that used a fixed frequency.

Scenario S1 – S5

	Ambient Temperature	# of Fans	Case Cover Used
S1	35 C (95 F)	0	No
S2	26 C (79 F)	0	No
S3	27 C (81 F)	0	Yes
S4	25.2 C (77 F)	1	Yes
S5	24.5 C (76 F)	2	Yes

Fig. 12. Description of Experimental Scenarios S1-S5

C. Results and Analysis

Figure 13 and Figure 14 give a summary of the results obtained from conducting experiments for scenarios S1-S5. As can be seen the use of this thermal feedback adaptive frequency approach gives about a 2.4x improvement for the best case scenario (S5) verses using the fixed frequency needed for safe operation during worst case conditions. Figure 16 and 17 show more details of the results obtained.

Effective Frequency (MHz): Scenarios S1-S5

	S1	S2	S3	S4	S5
Adaptive 30-120 MHz	48.3	67.7	55.9	95	119.5
Fixed 50 MHz	50	50	50	50	50

Fig. 13. Effect Frequency for Scenarios S1-S5

Average Temperature (C): Scenarios S1-S5

	S1	S2	S3	S4	S5
Adaptive30-120 MHz	68.4	68.4	68.5	68.5	69
Fixed 50 MHz	68.5	60	63	46	43

Fig. 14. Average Temperature for Scenarios S1-S5

Starting with the worst-case scenario S1, the base fixed frequency was determined to be 50 MHz. For this scenario, the adaptive frequency circuit achieves an effective frequency of 48 MHz. Under theses conditions, the fixed-frequency outperforms the adaptive frequency by about 4%. It is understandable that the fixed-frequency circuit performs better in this case because the fixed frequency is tailored for the worst-case scenario. For all other thermal experiments, however, the adaptive frequency obtains better performance as compared to the fixed frequency circuit. The amount of thermal budget regained by using a adaptive frequency is illustrated by Figure 15 a-e.

The main difference between scenario S2 and S3 is that S2 operated the rack-mount system without the cover installed. The presence of a cover without operating fans significantly degrades the thermal operating conditions. This is because the case cover not only traps heat generated by the application, but it also traps heat generated by the power supply unit located in the case. Having the cover in place without operating fans is nearly as bad of a scenario as S1, which used an external heat source applied to a system without the case cover. When the case cover is removed, heat can freely escape.

Scenarios S4 and S5 differ by the number of fans active. As can be seen from Figure 12 while the ambient temperature differs by less then 1 C, the difference in effective frequency is 25%. The addition of the second fan does not drop the case temperature significantly, but the increase in air follow does significantly increase the systems ability to pull heat away form the FPGA, therefore allowing the application to run at a significantly increased frequency.

Figure 16 gives a side by side comparison of the duty cycles observed for scenarios S1-S5. The duty cycle is shown over the measured junction temperature. These plots are ordered from top to bottom in increasing effective frequency. It can been seen that the time spent at the high-end frequency of 120 MHz increases as thermal operating conditions improve.

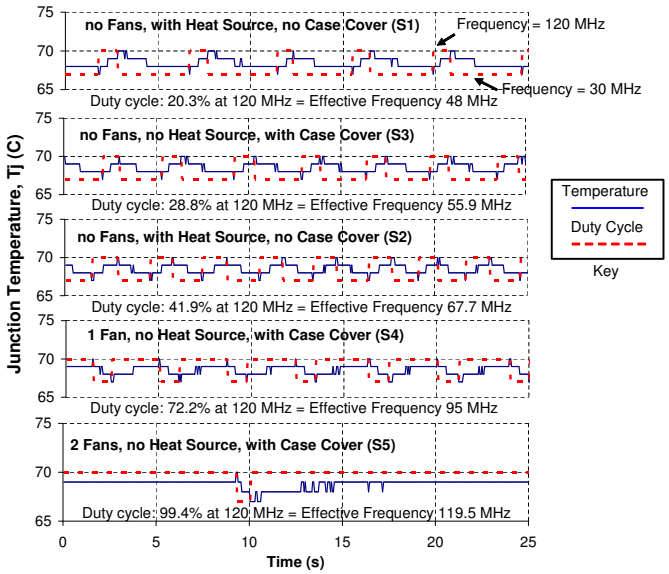

Fig. 16. Duty cycle comparisons of scenarios S1-S5

VI. Conclusion

An adaptive frequency control method using thermal feedback for reconfigurable hardware applications was presented. The thermal control method was implemented on the FPX platform and measurements were obtained to quantify the effectiveness of adaptive frequency control. Results show that for large variations in thermal operating environments, an application using a thermally adaptive frequency obtains much better performance than for the same circuit with a fixed frequency. For our image-processing case study, we achieve a factor of 2.4x in increased computational throughput.

Acknowledgment

The authors would like to thank the National Science Foundation for the funding of this research under grant ITR 0313203.

References

[1] P. H. Jones, J. W. Lockwood, and Y. H. Cho, "A thermal management and profiling method for reconfigurable hardware applications," in *16th International Conference on Field Programmable Logic and Applications (FPL)*, Madrid, Spain, Aug. 2006.

[2] Intel Corporation, "Addressing power and thermal challenges in the datacenter," 2005.

[3] *Intel 80200 Processor based on Intel XScale Microarchitecture Developer's Manual*, 2003.

[4] E. Wirth, "Thermal management in embedded systems," Master's thesis, University of Virginia, 2004.

[5] ACPI, "Acpi specification," http://www.acpi.info/spec.htm, Sept. 2004.

[6] C. T. Chow, L. S. M. Tsui, P. H. W. Leong, W. Luk, and S. J. E. Wilton, "Dynamic voltage scaling for commercial fpgas," in *ICFPT*, 2005, pp. 173–180.

[7] J. W. Lockwood, J. S. Turner, and D. E. Taylor, "Field programmable port extender (FPX) for distributed routing and queuing," in *ACM International Symposium on Field Programmable Gate Arrays (FPGA'2000)*, Monterey, CA, USA, Feb. 2000, pp. 137–144.

[8] J. W. Lockwood, N. Naufel, J. S. Turner, and D. E. Taylor, "Reprogrammable Network Packet Processing on the Field Programmable Port Extender (FPX)," in *ACM International Symposium on Field Programmable Gate Arrays (FPGA'2001)*, Monterey, CA, USA, Feb. 2001, pp. 87–93.

[9] *Virtex-II Platform FPGA User Guide*, Xilinx, 2005.

[10] Xilinx Inc., "Using delay-locked loops in spartan-ii fpgas," Xilinx XAPP174, Jan. 2000.

[11] Y. H. Cho, "Optimized automatic target recognition algorithm on scalable myrinet-field programmable array nodes," in *34th IEEE Asilomar Conference on Signals, Systems, and Computers*, Monterey, CA, Oct. 2000.

[12] K. Chia, H. J. Kim, S. Lansing, W. H. Mangione-Smith, and J. Villasenor, "High-performance automatic target recognition through data-specific vlsi," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 6, no. 3, pp. 364–371, Sept. 1998.

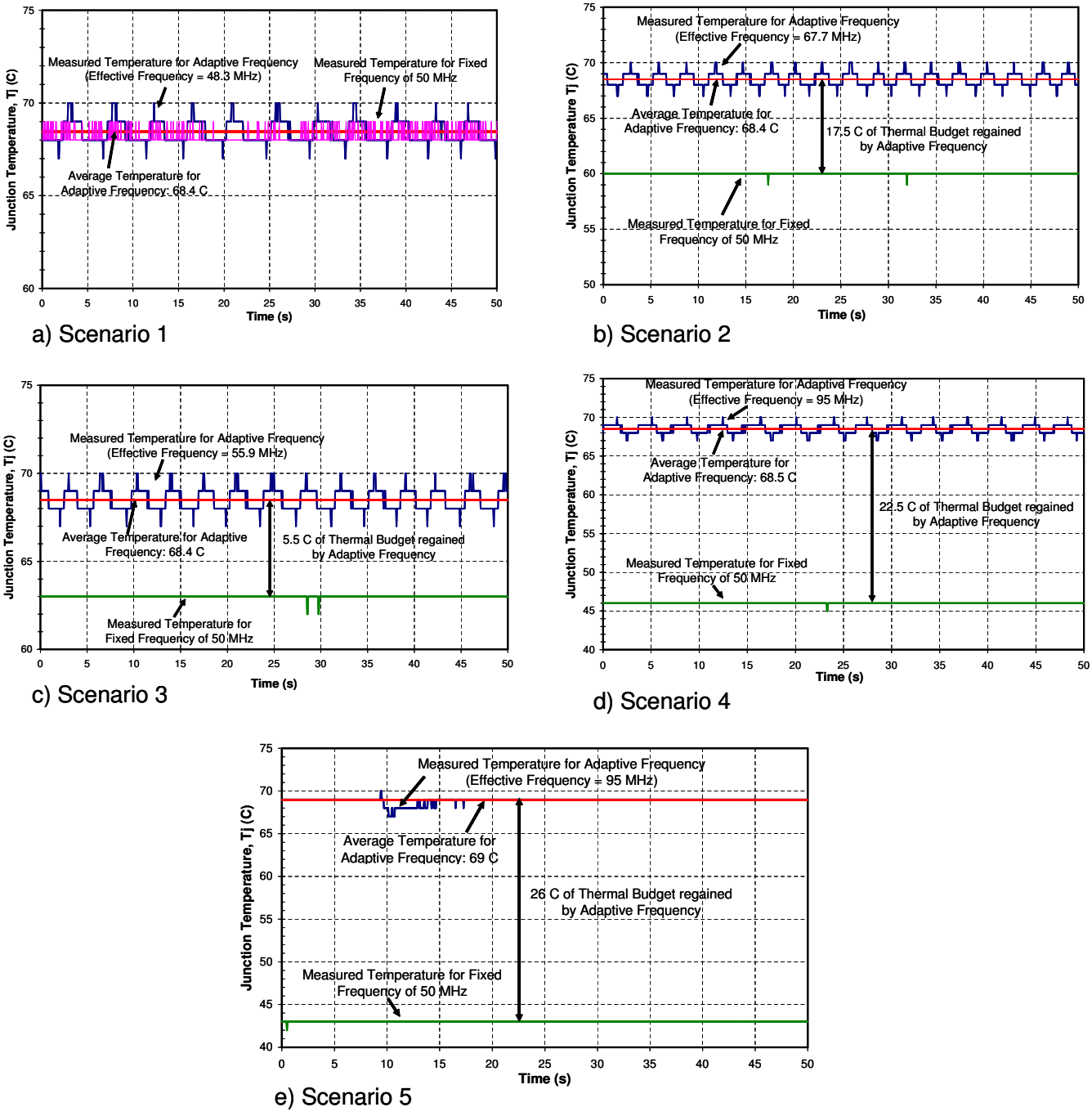

Fig. 15. Thermal budget gap for scenarios S1-S5, upper temperature threshold=70 C, lower temperature threshold=67 C