Changing Output Quality for Thermal Management

Phillip H. Jones, James Moscola, Young H. Cho, John W. Lockwood
Washington University in St. Louis
Department of Computer Science and Engineering
St. Louis, Missouri 63130
{phjones,jmm5,young,lockwood}@arl.wustl.edu

1 Introduction

A growing number of embedded computing systems are used outside of environmentally controlled locations. In locations such as remote parts of deserts, deep ocean floors, and outer space, it is not only difficult to predict environmental effects on a system, they also allow very limited accessibility once a system is deployed. Therefore, it is often necessary for system parameters to be over provisioned to guarantee correct functionality under worst case environmental conditions. This often leads to an end system that is suboptimal for typical conditions.

Our work attempts to overcome the performance loss in such systems due to over-provisioning. In this paper, we present an adaptive mechanism that automatically adjusts system operating parameters to yield the best performance for the given environmental conditions. Real-time feedback from sensors is used to tune our circuits to run at near-optimal performance. Our previous work in this area is published in [1, 2, 3]. Other work in this area has been published in [4, 5].

Developers of high performance computers constantly strive to build efficient systems that obtain the highest performance per area. As transistors shrink in size and are more densely packed, one of the biggest concerns for system developers today is system temperature management. As the ambient temperature rises, the heat generated by circuits can cause devices to become unstable or damaged due to internal temperatures rising above maximum operating thresholds. Thus, our research focuses on the regulation of circuit temperature.

The adaptive circuit illustrated in this paper allows the system to make application specific tradeoffs between output quality and generated heat. A performance evaluation was conducted to quantify the difference between our adaptive system and a static system under different environmental conditions. For our case study, we use an image processing application that offers a high degree of parallelism.

2 Image Recognition Application

Image recognition is an application that is well-suited for hardware implementation. It is a highly parallelizable application that allows multiple image processing cores to run simultaneously on the same image data. This decreases the amount of work each processing engine must do, thereby decreasing the total time required to process an image. We utilize an image recognition application with multiple processing engines to evaluate the effectiveness of our thermally adaptive techniques.

We model our image recognition application as a feature-based image processor with support for four different features. Up to two masks for each feature can be scanned for in four parallel image processing cores for a total of eight masks. A block diagram of the basic image recognition application is shown in figure 1.

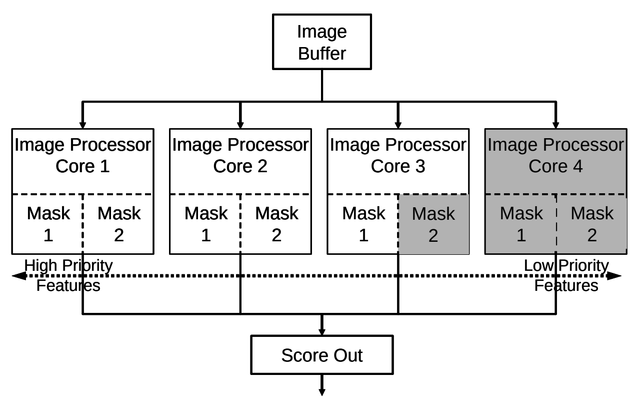

Figure 1. Application Adaption Example

2.1 Adaption Policy

Under optimal thermal conditions the image recognition application processes images at the maximum frequency using both masks for all four image processing cores. As the

FPGA heats up and thermal conditions become more adverse, it is necessary to modify the application parameters to prevent the FPGA from overheating.

The image recognition application parameters can be modified in two ways. The first method used to decrease the temperature of the FPGA is to decrease the clock frequency. Decreasing the clock frequency decreases the amount of work done by the chip, thus lowering its temperature. Since some image recognition applications may require that some minimum number of frames be processed per second, there is a minimum frequency that the circuit must operate above. If the circuit is already operating at its minimum frequency and the thermal conditions continue to degrade other parameters of the application must be modified.

In feature-based image recognition applications, some features may have higher priority than others. If this is the case, then features of the image recognition application can be put in order of their priority and selectively disabled to decrease the amount of work done by the FPGA and hence its temperature. Figure 1 illustrates the priority ordering of features. Lower priority features are disabled first and then the higher priority features. In figure 1, both mask 1 and mask 2 of image processing core 4 are disable, effectively disabling this image processing core. Mask 2 of image processing core 3 is also disabled.

3 Test Setup & Results

The image recognition application was deployed on the Virtex-4 FX100 FPGA of our development platform. This platform was installed into a 3U rackmount case equipped with 2 fans that each supply approximately 250 Linear Feet per Minute of air flow. During the execution of an experiment the top of the case is put into place and a thermally isolating cover is placed over the case to help isolate the thermal conditions inside the case from the laboratory conditions. An external heat source was used to generate ambient temperatures inside the case greater than that of the laboratory. A temperature probe was installed in the case to monitor the ambient case temperature during experiments.

Evaluation experiments were conducted to compare the performance of the adaptive image processing application to a static version. These experiments were executed under the four different thermal conditions, shown in figure 2. The clock frequency and number of processing cores used for the static application was determined by finding the frequency and number of cores, under worst case thermal conditions (i.e. scenario S1), for which a thermal budget of 65 C was maintained. This frequency was found to be 50 MHz, and the number of cores was found to be two. The adaptive version of the application uses a thermally controlled frequency that can switch between 50 MHz and 200 MHz. The adaptive application can also use up to four cores.

Figure 3 gives a summary of the experimental results for scenarios S1-S4. First it should be noted that the adaptive version of the application can reduce its frequency and number of cores in order to operated safely under worst case thermal conditions (S1). As thermal conditions improve the adaptive application takes advantage of these conditions by first increasing the number of cores (i.e. quality) of the application, then by increasing the effective frequency. Under best case thermal conditions (S4) the adaptive application can operate at 200 MHz, 4 times the frequency of the static application, and has 2 times the quality.

	Ambient Temperature	# of Fans
S1	37.0 C (98.6 F)	0
S2	28.0 C (82.4 F)	0
S3	24.8 C (76.6 F)	1
S4	24.0 C (75.2 F)	2

Figure 2. Thermal conditions used for evaluation

Effective Frequency (MHz): Scenarios S1-S4

	S1	S2	S3	S4
Adaptive 50-200 MHz	50.0	66.0	160.0	200.0
Fixed 50 MHz	50	50	50	50

Quality Level (# of features scanned): Scenarios S1-S4

	S1	S2	S3	S4
Adaptive 50-200 MHz	4	8	8	8
Fixed 50 MHz	4	4	4	4

Figure 3. a.) Effective Frequency, b.) Quality Level

References

[1] P. H. Jones, Y. H. Cho, and J. W. Lockwood. An adaptive frequency control method using thermal feedback for reconfigurable hardware applications. In *IEEE International Conference on Field Programmable Technology (FPT)*, Bangkok, Thailand, Dec. 2006.

[2] P. H. Jones, Y. H. Cho, and J. W. Lockwood. Dynamically optimizing fpga applications by monitoring temperature and workloads. In *20th IEEE/ACM International Conference on VLSI Design*, Bangalore, India, Jan. 2007.

[3] 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.

[4] J. Peddersen and S. Parameswaran. Energy driven application self-adaptation at runtime. In *20th IEEE/ACM International Conference on VLSI Design*, Bangalore, India, Jan. 2007.

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