Adaptive Computing

Augustus K. Uht
Microarchitecture Research Institute
University of Rhode Island

December 15, 2004

Abstract

Recently, Adaptive Systems have been proposed to realize better-than-worst-case performance or power consumption. At the same time, such systems adapt to dynamically-changing environmental and operating conditions, as well as statically-varying manufacturing conditions. Adaptive systems take advantage of whatever conditions exist and optimize for the best performance, power, etc.

Overview of Presentation

Many systems must operate under trying conditions, ideally with good performance. However, classical design requires that worst-case scenarios be assumed when setting final system operating specifications. This eliminates higher performance or lower power consumption that could be obtained under typical conditions ***if*** one could ensure correct operation in all cases, e.g., stressful conditions.

In the past several years two approaches have been proposed to realize such systems: *Timing Error Avoidance (TEA)*[3, 4] and *Timing Error Toleration*[1, 2]. Using performance-enhancement as an example, in the latter approaches the clock frequency is increased until a timing error occurs; the system then corrects or recovers from the error, slows down the clock, and the process repeats.

In the TEA approach, timing errors never occur in the regular system logic. Instead, a one-bit wide copy of the worst-case path through the system (plus a safety margin) is monitored. When an error occurs in the copy, we know that an error might occur in the regular logic if the frequency is further increased; therefore, the frequency is reduced, and the process repeats.

At the University of Rhode Island we have investigated both approaches, but have mainly focused on the TEA approach. We have built two prototypes to investigate different aspects of adaptive systems.

The *TEAtime* prototype (see Figure 1) demonstrated the basic feasibility of the TEA approach. Performance could almost be doubled (on a simple 32-bit 5-stage pipelined CPU), while the system readily adapted to changes in temperature and operating voltage, executing almost as fast as possible under the existing conditions; see Figure 3.

We have also investigated an approximation of the TEA approach on a real PC. The prototype is called: TEAPC; see Figure 2; [5]. In this situation the key problem was not enhancement of performance per se, but rather devising a suitable control system to change its operating voltage and frequency so as to adjust to varying loads and conditions. In the work cited above, with Prof. Rick Vaccaro, a formally-designed feedback control system was designed and realized completely in software. The complete program runs as a regular application program on a Windows 2000 –based PC. It uses less than 1% of the CPU time. With this system, we are able to maintain a constant CPU core temperature, adapting to varying conditions. In an extreme case, we have also demonstrated disaster tolerance: on a 3.0 GHz Pentium 4 under full load, we turned off the CPU's fan. The control system kicked in, and reduced the core frequency and voltage to a point where the CPU could continue to operate safely: its temperature stabilized at a safe value. See Figure 4.

The TEAPC approach is applicable to a wide range of systems, e.g., embedded systems and server farms. It is also applicable to power control, etc.

In this presentation we will review the TEA approach, data, and ongoing work. Time and resources permitting, we will also give a live demo of TEAtime and/or TEAPC.

References

[1] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," in Proceedings of the 2003 International Symposium on Microarchitecture. San Diego, Calif., USA: IEEE, ACM, December 2003.

[2] A. K. Uht, "Achieving Typical Delays in Synchronous Systems via Timing Error Toleration," Department of Electrical and Computer Engineering, University of Rhode Island, Kingston, RI, Technical Report 032000-0100, March 10, 2000.

[3] A. K. Uht, "Uniprocessor Performance Enhancement Through Adaptive Clock Frequency Control," in Proceedings of the SSGRR-2003w International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, e-Medicine, and Mobile Technologies on the Internet. L'Aquila, Italy: Telecom Italia, January 6-12, 2003.

[4] A. K. Uht, "Going Beyond Worst-Case Specs with TEAtime," Computer, vol. 37, no. 3, pp. 51-56, March 2004.

[5] A. K. Uht and R. J. Vaccaro, "TEAPC: Adaptive Computing and Underclocking in a Real PC," in Proceedings of the First IBM P=ac2 Conference. Yorktown Heights, NY, USA: IBM T.J. Watson Research Center, October 6-8, 2004, pp. 45-54. URL: http://www.ele.uri.edu/~uht/papers/IBM-PAC2-F2004-44-Uht-Fnl-PostPrint.pdf.

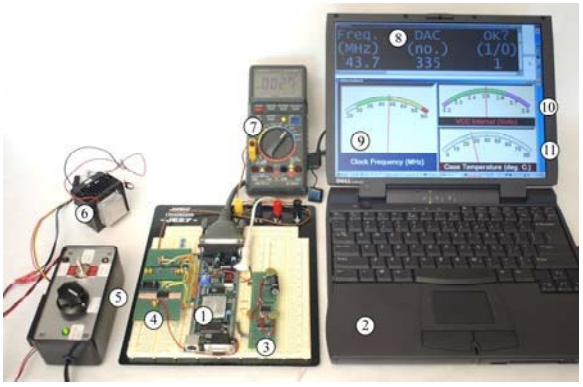

Figure 1. TEAtime prototype with experimental and demonstration setup. See [4] for details.

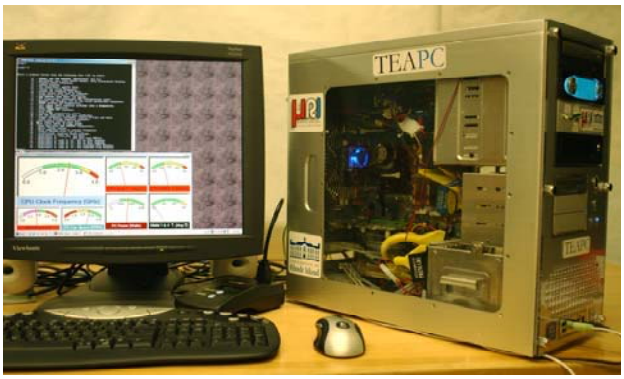

Figure 2. TEAPC prototype, with experiment instrumentation shown on the display.

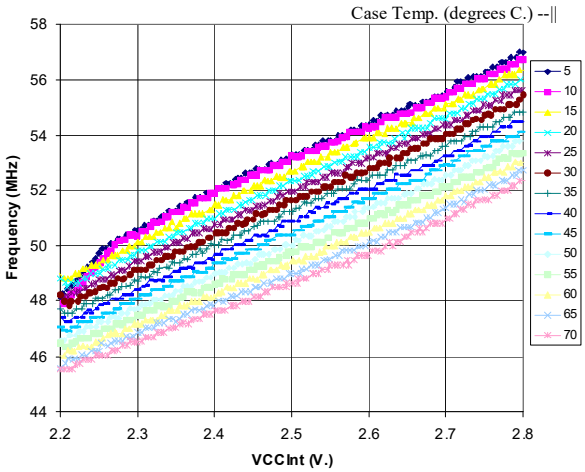

Figure 3. TEAtime frequency as a function of case temperature and chip VDD. The baseline frequency is 30 MHz.

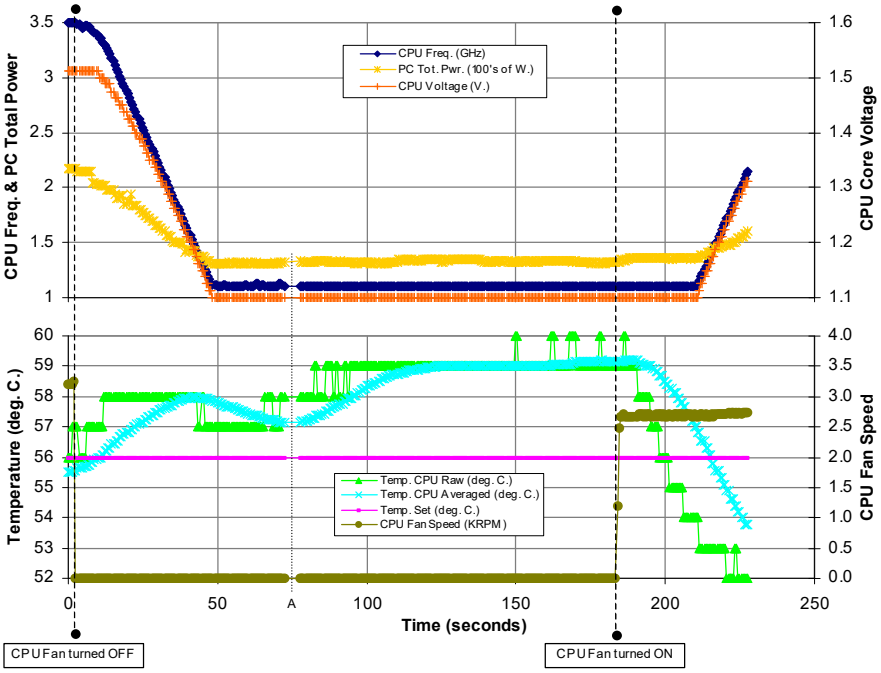

Figure 4. Example of disaster tolerance and recovery: CPU fan turned off then back on; system under full load. TEAPC remains functional at the low frequency and core voltage, even with the fan off. TEAPC continuously adapts to take the best advantage of existing conditions.