TEAPC:
Adaptive Computing and Underclocking in a Real PC

Augustus K. Uht and *Richard J. Vaccaro*
Dept. of Electrical and Computer Engineering

IBM PAC2: October 6, 2004

Background

Prior work: *TEAtime* (Timing Error Avoidance)

- "Maximized" performance. All hardware. Method:
  - Use one-bit copy of worst-case delay path in system; add a small safety-margin delay.
  1. *Speed up clock 'til just before real error would occur*
  2. *Slow down clock*
  3. *Repeat: GOTO 1.*
- TEAtime adapts to:
  - Current environmental conditions (e.g., temperature)
  - Current operating conditions (e.g., voltage)
  - Prior manufacturing conditions (quality of a prod. Run)
- Prototype almost doubled performance

Outline

1. Motivation and Goals
2. Related Work
3. TEAPC's Features
4. Feedback-Control System
5. TEAPC Software and Hardware Details
6. Experiments
7. Summary
8. Demo

Motivation and Goals

- **Motivating Goals:**
  - Realize TEAtime characteristics in a real computer
    - Adaptive computing
    - Improved performance – "Better-than-Worst-Case"
- Additional Goals:
  1. ***Workload adaptation***
  2. ***Reduced power consumption***
  3. ***Improved reliability***
  4. ***Disaster tolerance (always enabled)***
  5. ***…and all in a real machine***
- BUT: can't redesign or build Pentium 4's
- SO: use real IBM/Intel-standard PC

Related Work

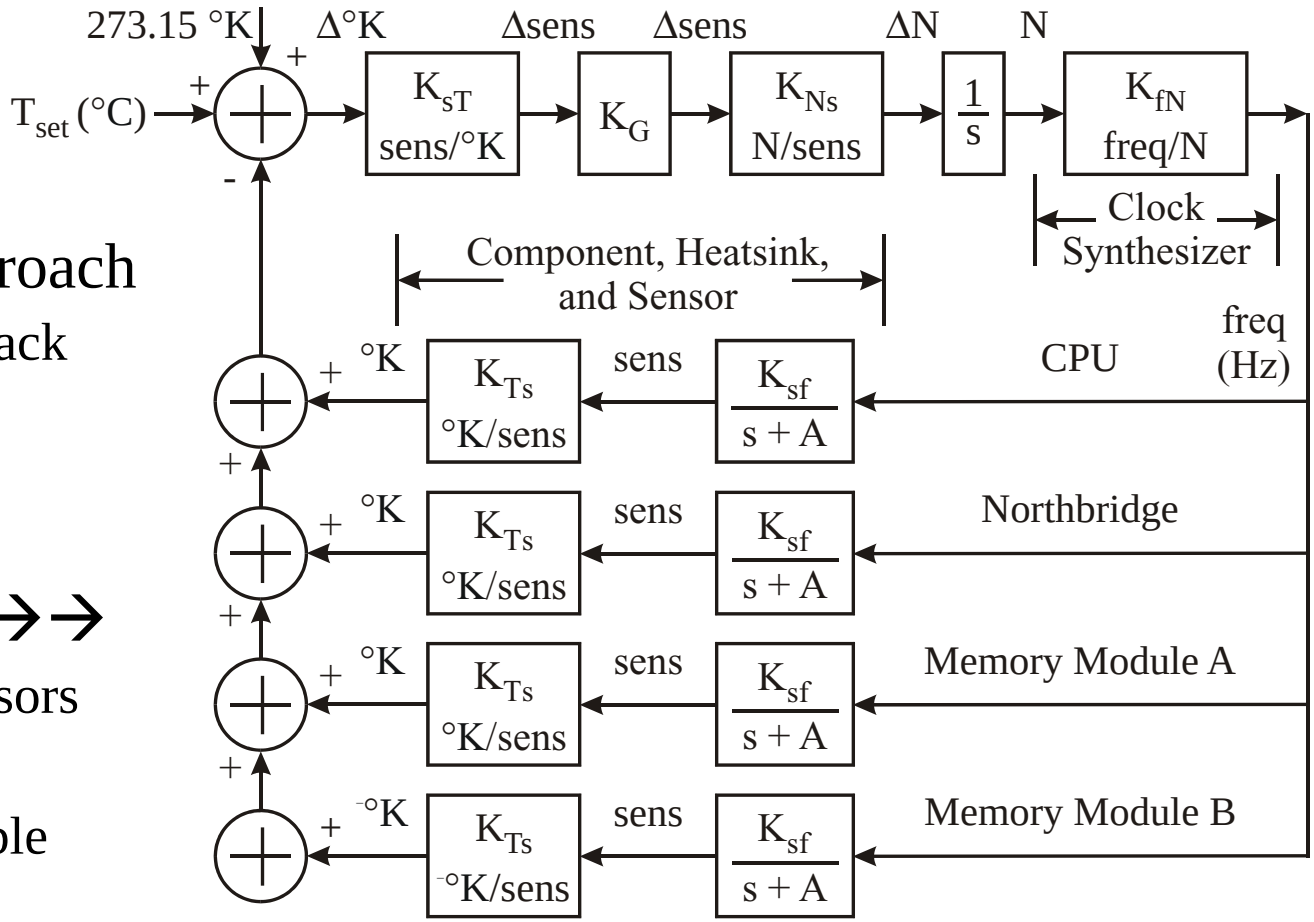
- Rohou & Smith, 1999 – temperature adaptive system
  - Adjusted temperature with frequency changes
  - BUT: required modifying OS (Linux)
  - Performance not enhanced
- Skadron *et al*, 2002 – temperature adaptive system
  - Used classical feedback control theory
  - Modeled and controlled temperatures of parts of a chip
  - Instruction-fetch toggling controlled temperature
  - Performance reduced
- (Note: ~all adaptive methods useful for either or both:
  - Performance improvement
  - Power reduction)

TEAPC's Key Features

- High thermal capacitances and delays, hence:
  Modern feedback-control theory and system used
  - Input: CPU's internal temperature (from embedded thermal diode)
  - Outputs: CPU's clock frequency (I/O clocks unmodified), Vcore
- User-control provided by control system's Tset point
- Entire system realized in Windows application
  - No OS modifications – Windows 2000 used
  - No hardware modifications – all COTS parts
- Applicable to many kinds of PC's
  - New designs of commercial PC's
  - Possibly existing motherboards (MOBO)

Control System History

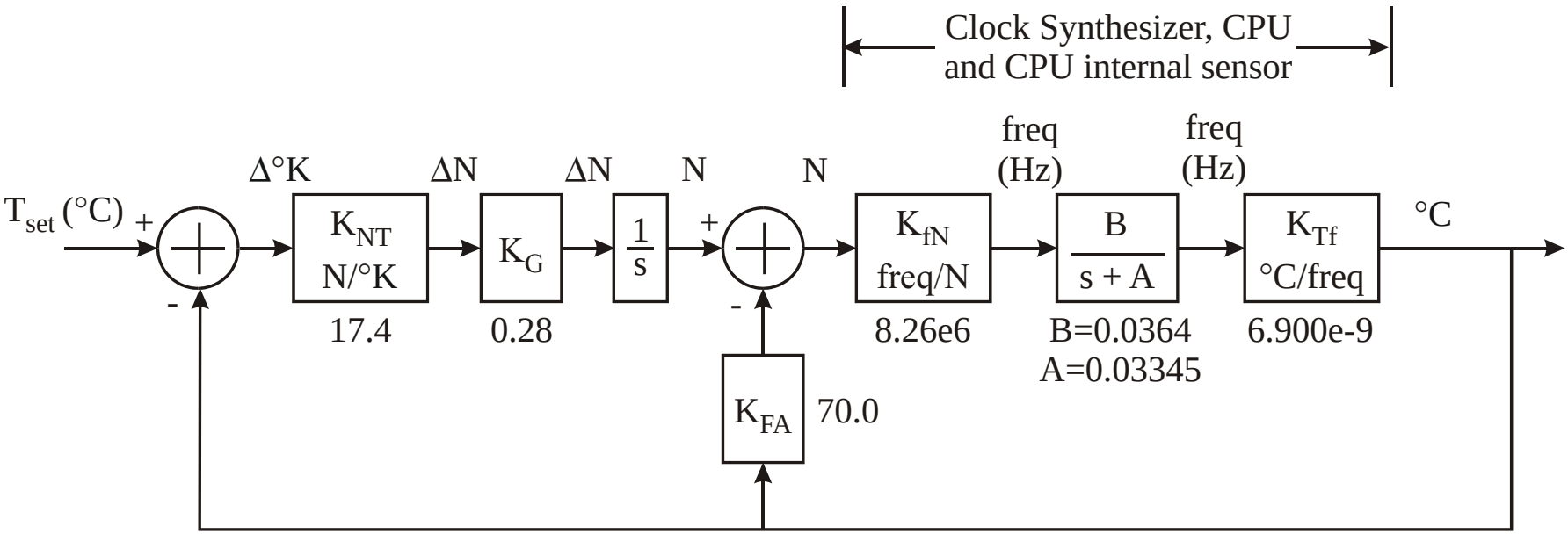
1. TEAtime approach
   - Direct feedback
   - In TEAPC: Oscillated
2. TEAPC-0 →→→
   - External sensors
   - Too slow
   - Uncontrollable



Note: all of the constants of the four components differ.

Final Control System

- Only input is CPU temperature (feedback line)
- Primary output is CPU frequency (N) [sometimes: Vcore = *f* (N)]
- State-space discrete control system design (modern)
- Quick response



teapc Control Program

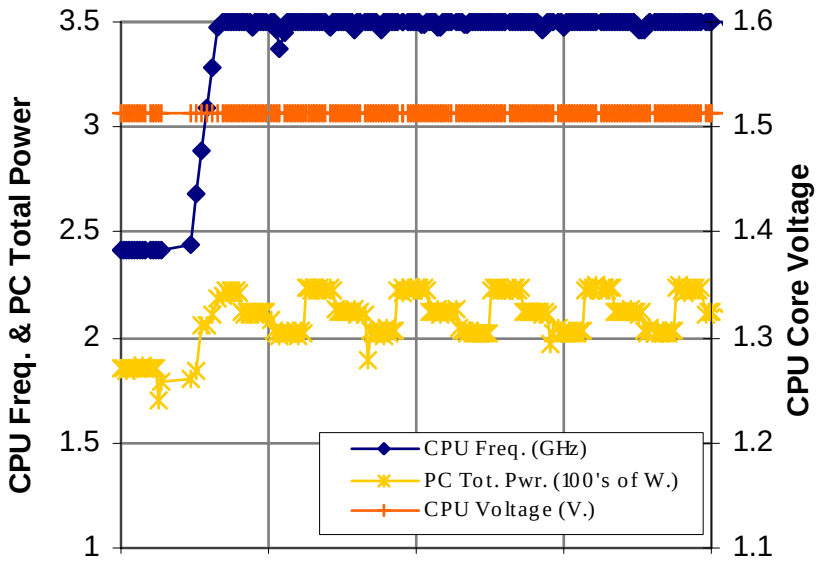
- Windows application – no changes to OS
- Uses x86 I/O address space to access hardware
- Small: 800 kilobytes
- Fast: < 5% CPU utilization
- Control loop updated every second
- Hard limits on max./min. frequency

TEAPC Components

PC Component	Manufacturer	Part Number/Description
Motherboard	Gigabyte	GA-8KNXP (Rev. 2); w/DPS regulator
CPU	Intel	P4 3.0 GHz, 800 MHz bus
Chipset	Intel	875P, ICH5R
Clock Synthesizer	ICS	ICS952635
Super I/O (Environment Mon.)	ITE	IT8712F V0.6
CPU Volt. Regulator Control	ITE	IT8206R V0.1
Main Memory	Ultra	U10-5903R; 2 x 512 MB; 400 MHz DDR, Dual Channel (Operated at 320 MHz.)
Operating System	Microsoft	Windows 2000 SP4, HT disabled
Disk System – RAID 0+1	ITE	GigaRAID IT8212F
Disks	Maxtor	4 x 6E040L0, 40 GB, 133MHz IDE
Equipment for experiments only		
Fan Controller & Temp. Mon.	Thermaltake	Hardcano 12; for 4 fans, 4 thermocouples
Power Meter	Electronic Educational Devices	watts up? PRO (Note: this is the unit's model name.)
CPU Fan Controller	custom	On/Off, control sel. (MOBO or Hardcano)

Underclocking & Performance "Maximization"

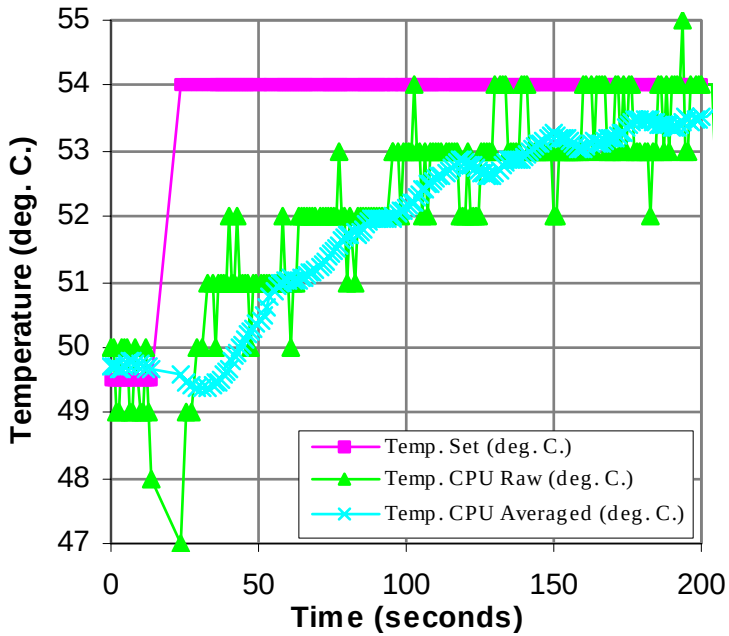
CPU nominal freq. = 3.0 GHz →



Freq. (GHz)

Vcore (V)

Power (100 W)



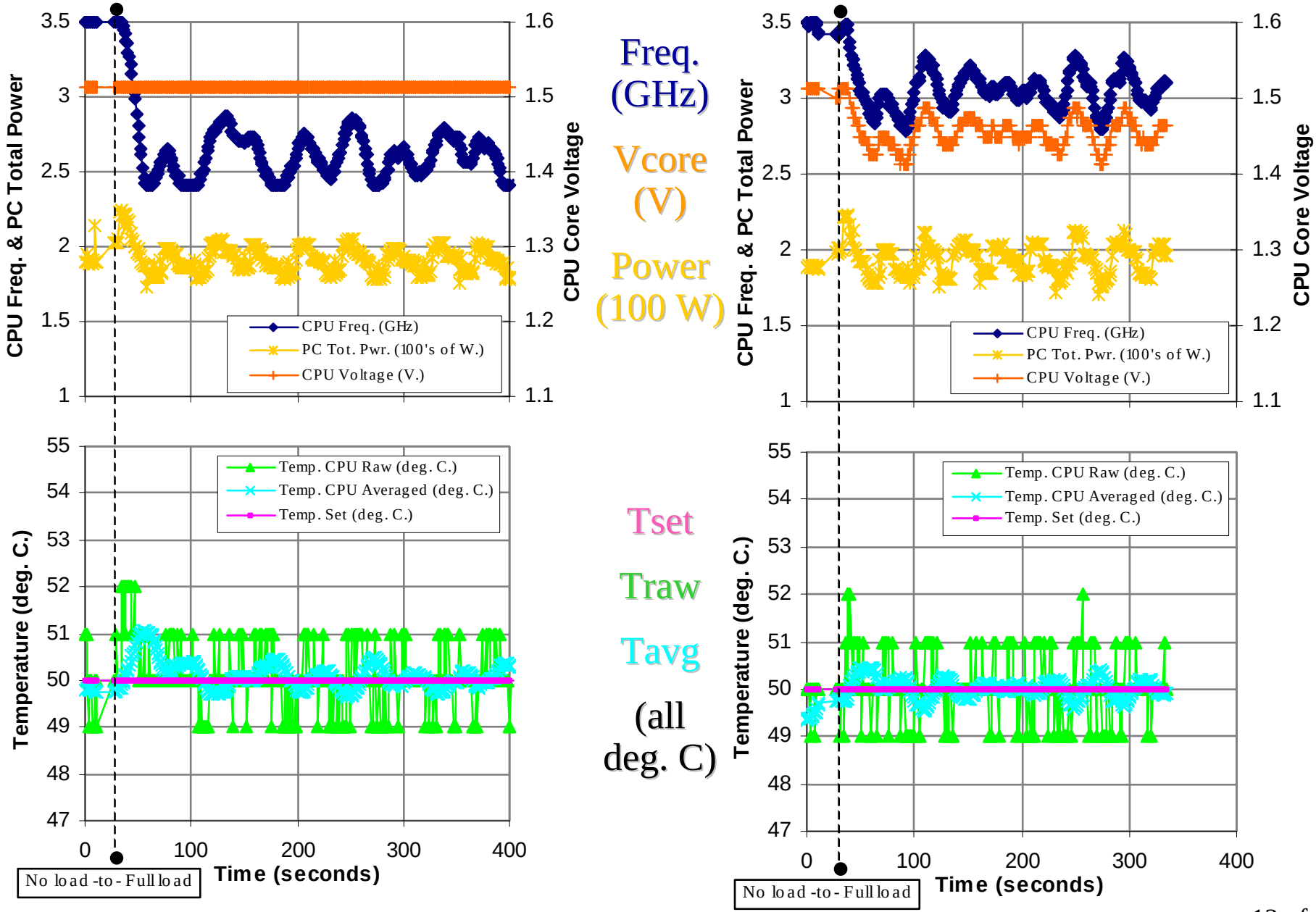
Tset

Traw

Tavg

(all deg. C)

freq., Vcore **un**linked < Load Adaptation > freq., Vcore linked



Disaster Tolerance

- Example: CPU Fan dies….
- Changes (automatic, via feedback system):
  - Freq: 3.5 GHz -> 1.1 GHz
  - Vcore: 1.5125 V. -> 1.0875 V.
  - → Power: 222 W. -> 140 W. (37% savings)
- CPU temperature stabilizes at safe value (with this CPU)
- System still functional

Summary

- TEAPC realizes:
  1. Better-than-worst-case performance
  2. ***Adaptive*** operation to both environment and loading
  3. Low-power, high-reliability operation
  4. Disaster tolerance
- Feedback-control great for a system, too
- ***Underclocking*** is a great tool
- ***It Works!***

…and now it's time for the:

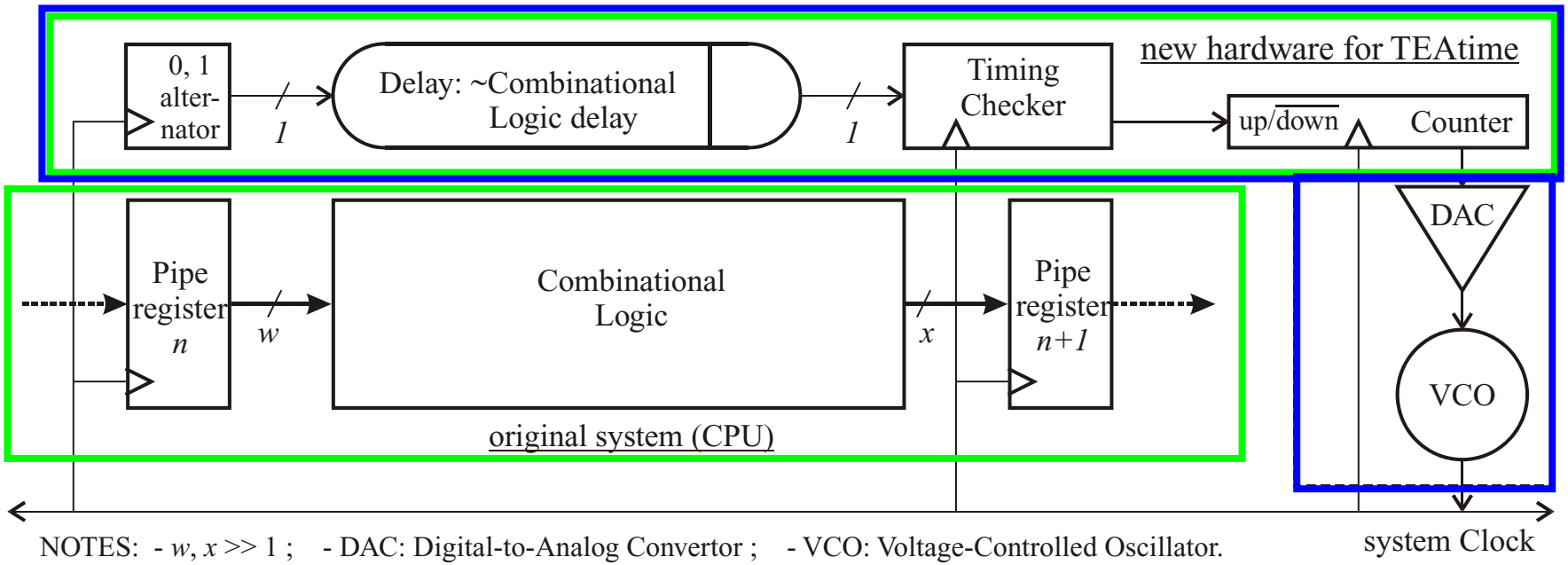
DEMO

TEAPC:
Adaptive Computing and Underclocking in a Real PC

Augustus K. Uht and *Richard J. Vaccaro*
Dept. of Electrical and Computer Engineering

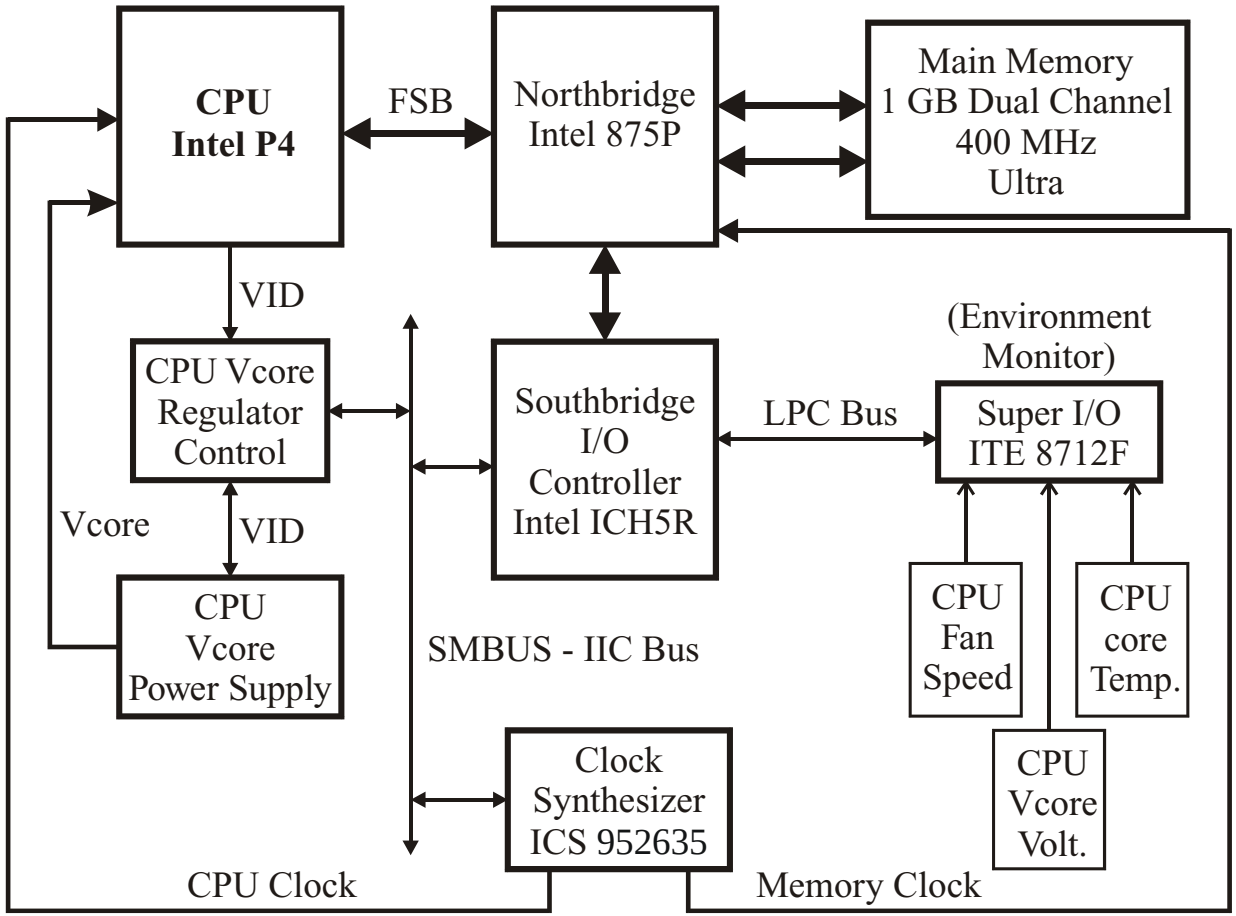
IBM PAC2: October 6, 2004

TEAtime Block Diagram



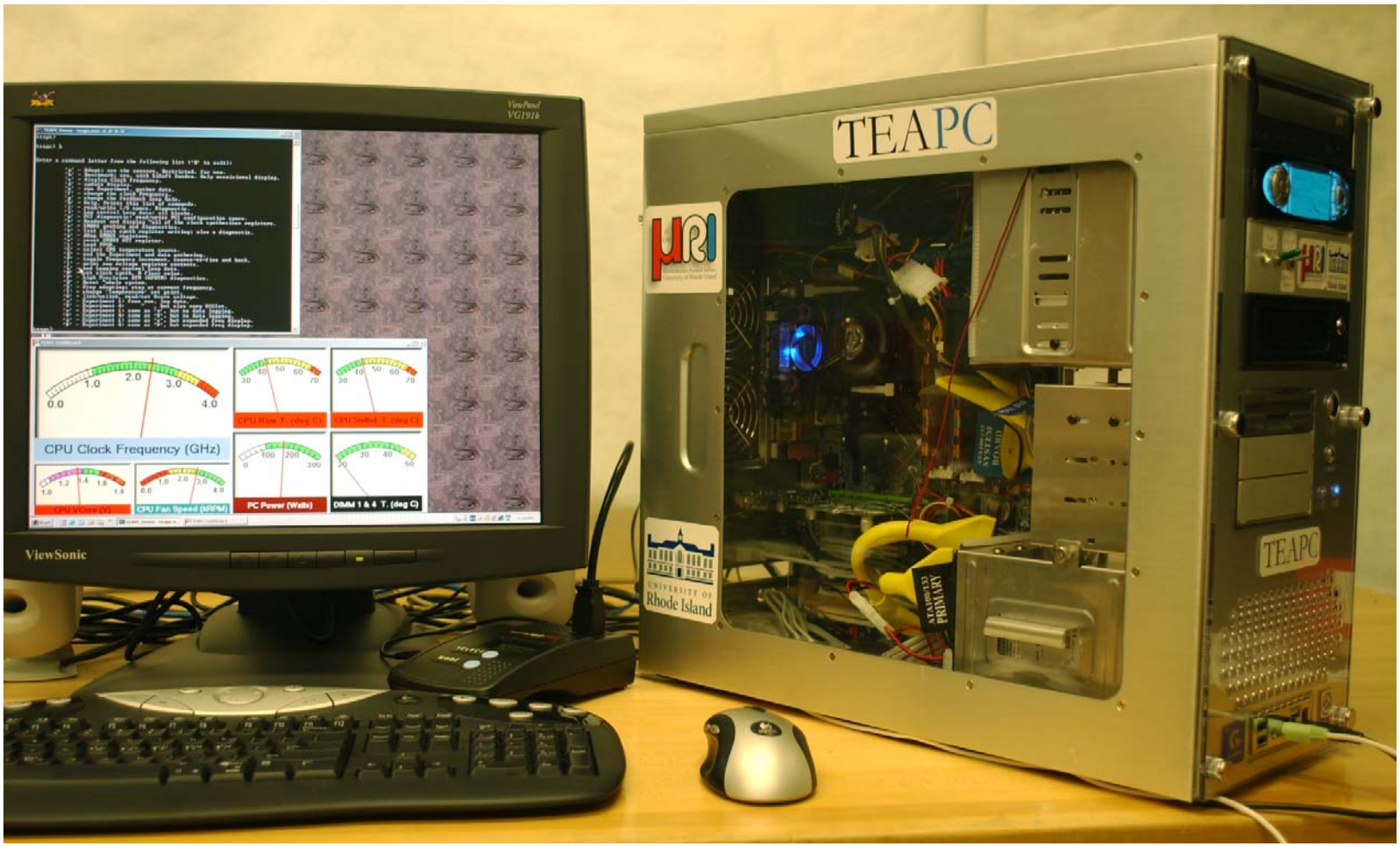
- Timing Error Avoidance system
  - Blue: TEAtime hardware
  - Green: on FPGA

TEAPC Block Diagram



(FSB - Front Side Bus) Only directly relevant components and connections are shown. (LPC - Low Pin Count)

Experiment Setup



IBM PAC2: October 6, 2004

TEAPC

Operating Scenarios

Goal	Tset	Goal	Tset
Comments		**Comments**	
Low power, high reliability	Low	Disaster tolerance	Any
Low CPU freq. & Vcore. Un-intensive apps., e.g.: web-browsing. Still works.		TEAPC always enabled for disaster tolerance.	
Mid- power & reliab.	Mid-range	Environment adaptivity	Any
CPU freq. & Vcore vary. Ex.: Moderately-intensive apps. such as low-end animation.		E.g.: High temps.: CPU temp. kept to safe level. Still works. E.g.: Low temps.: High perf.	
High performance	High: freq. pegged		
High CPU freq. & Vcore. Ex.: FPGA net routing; 3-D games.			