Adaptive Computing

Gus Uht

(w/ Rick Vaccaro)

Dept. of Electrical and Computer Engineering

UNIVERSITY OF
Rhode Island

@URI
μRI
Microarchitecture Research Institute
University of Rhode Island

BARC: January 21, 2005

Background

- Three typical goals (alone or combined):
  1. ~Maximize performance.
  2. ~Minimize power consumption.
  3. **Adapt** to current (or prior) conditions:
     a. Environmental, e.g., temperature. (current)
     b. Operational, e.g., power supply voltage. (current)
     c. Manufacturing, e.g., slower or faster chips. (prior)
- ~All adaptive methods applicable to all.
- “Better-than-worst-case design”

Outline

1. Adaptive Methods
2. TEAtime – Prototype – Picture

3. Adaptive Control System
4. TEAPC & TEA42 – Prototypes and Data
5. Demo – TEA42
6. Summary

Adaptive System Methods

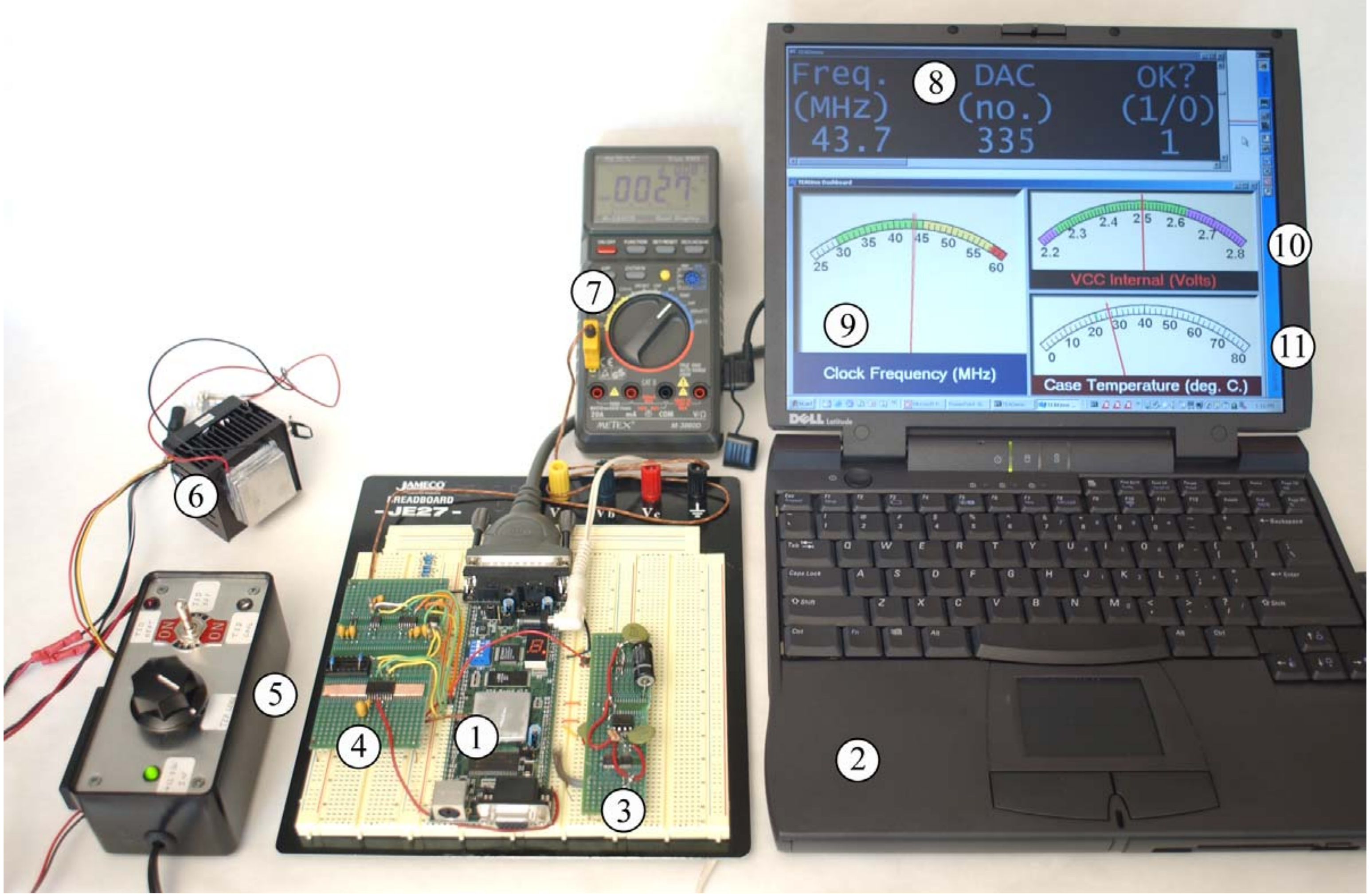
- Timing Error *Toleration:*
  1. Change something (freq., volt.), then:
  2. ***Let error occur, detect it, then recover.***
  3. Change the thing back, repeat: GOTO 1.

  ➢ Complex, hard to design; only ~optimal within cycle.

- Timing Error *Avoidance:*
  1. Change something (freq., volt.), then:
  2. ***Stop just before error would occur.***
  3. Change the thing back, repeat: GOTO 1.

  ➢ Simple, easy to design & build; constant cycles.

TEAtime Prototype

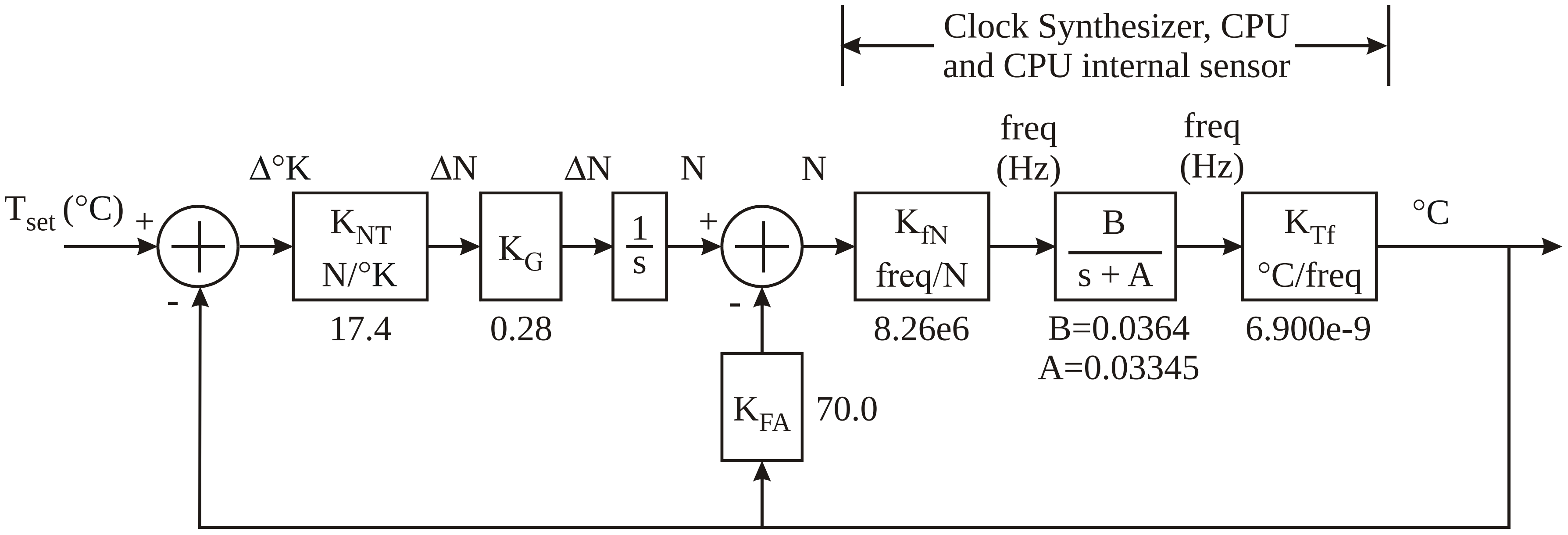


TEAPC, TEA42 Goals

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

Current Control System

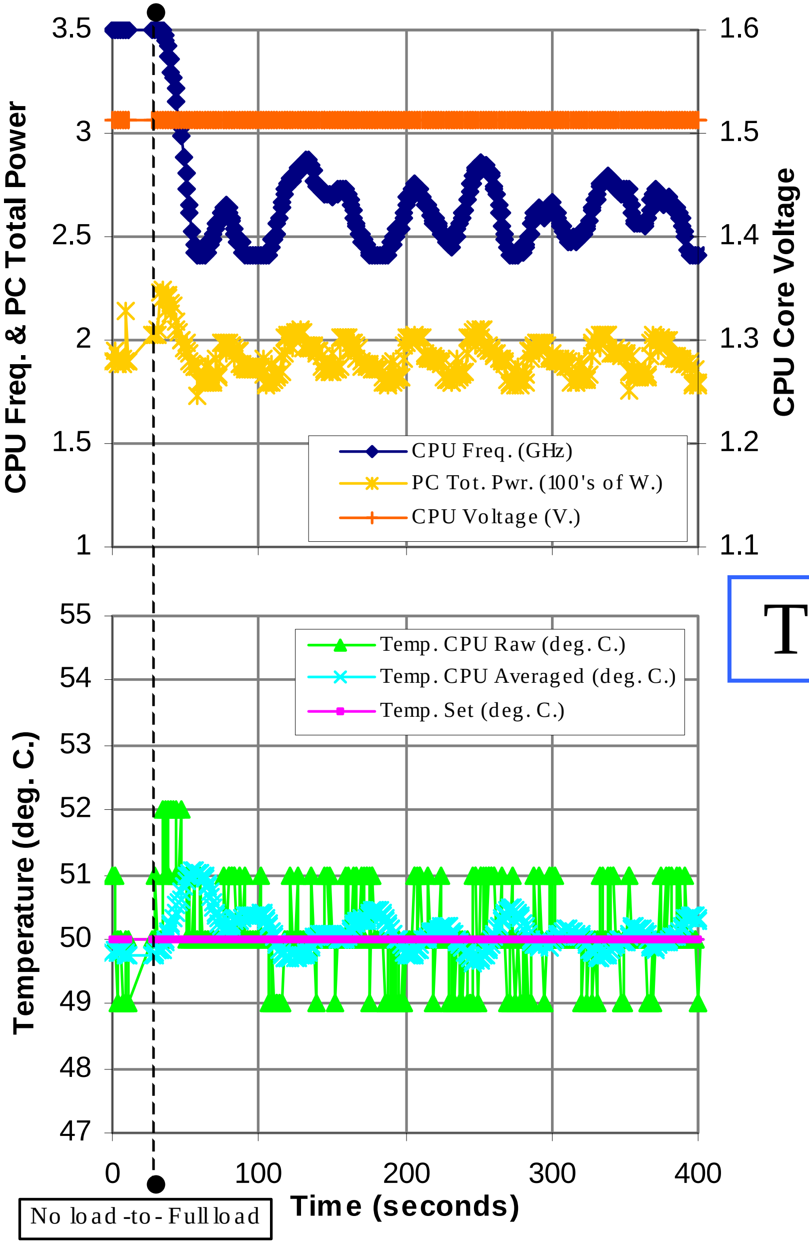
- Only input is CPU temperature (feedback line).
- Primary output is CPU frequency (N).
  [sometimes: Vcore = *f* (N)]



TEAxx Prototypes' Features

- Hardware:
  - No modifications → all COTS parts.
  - System: ~3.0 GHz Pentium 4;
    Intel chipset; 1 GB RAM.
- Software (`teapcwin` program):
  - Realizes feedback control system.
  - Standard MS Windows application.
  - (Standard OS: W2K SP4; no modifications [of course].)
  - Small: 1.1 megabytes.
  - Fast: low CPU utilization.

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



No load-to-Full load

Freq. (GHz)

Vcore (V)

Power (100 W)

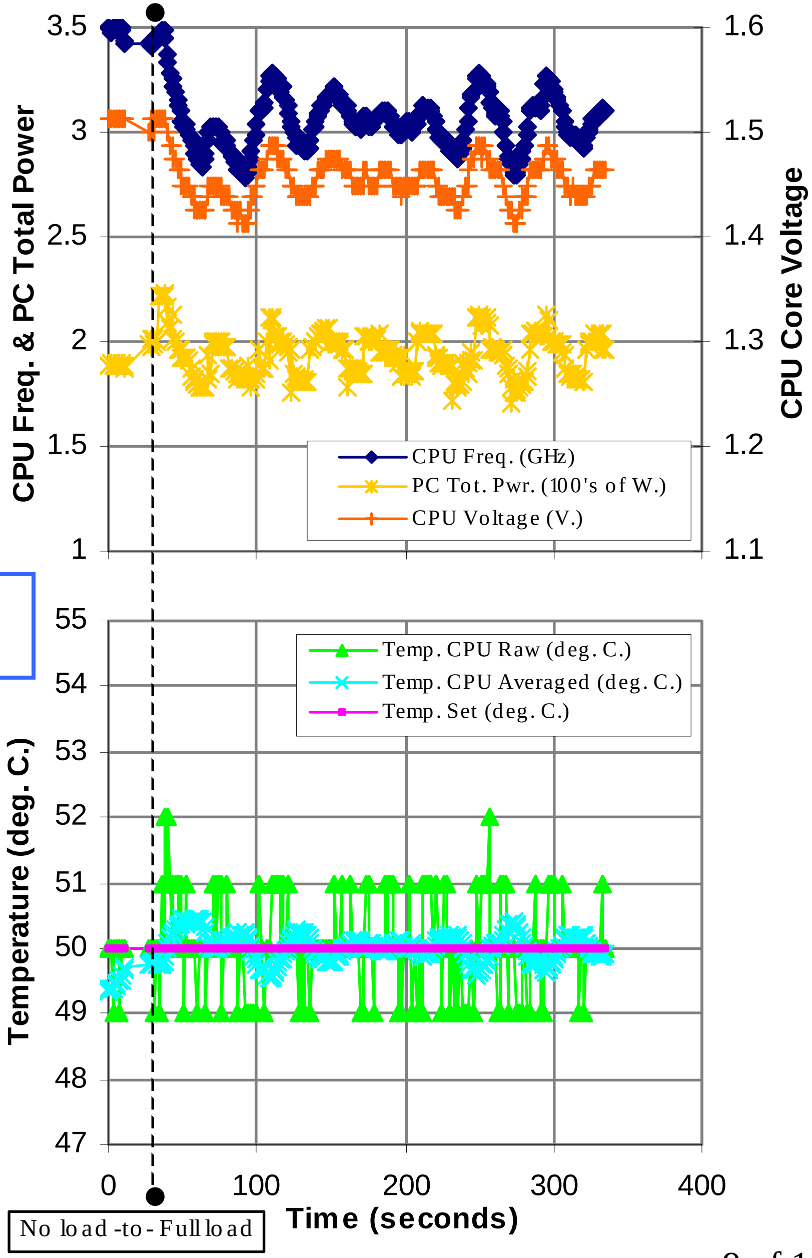
TEAPC

Tset

Traw

Tavg

(all deg. C)



No load-to-Full load

…and now it's time for the:

DEMO

Overall Summary

- TEAtime, TEAPC & TEA42 realize:
  1. Better-than-worst-case performance.
  2. ***Adaptive*** operation to both environment and/or loading.
  3. Low-power, high-reliability operation.
  4. Disaster tolerance.
- Feedback-control great for a system, too.
- ***Adaptive systems*** are the way to go.
- ***They Work!!***

Pointers…

- Appendix.
- *Computer*, March, 2004 Special Issue.
- *IEEE Trans. Computers*, Feb. 2005.
- My website: www.ele.uri.edu/~uht
- Or µRI website: www.ele.uri.edu/muri

Adaptive Computing

Gus Uht

(w/ Rick Vaccaro)

Dept. of Electrical and Computer Engineering

UNIVERSITY OF
Rhode Island

@URI
μRI
Microarchitecture Research Institute
University of Rhode Island

BARC: January 21, 2005

Appendix

15. Timing Error Toleration.
16. Timing Error Avoidance.
17. Adaptive Control Systems.
18. TEAPC Block Diagram.
19. TEAPC Components.
20. [TEAPC] Experiment Setup.
21. TEAtime Block Diagram.
22. [TEAtime Data.]
23. TEAtime Summary.
24. TEA42 Disaster Tolerance.

Timing Error Toleration

- Operation:
  1. *Speed up clock (resp. reduce voltage) until error occurs.*
  2. *Slow down clock (resp. increase voltage) for no error.*
  3. *Backtrack / repair error.*
  4. *Repeat: GOTO 1.*
- Examples:
  - Uht 2000: *TIMERRTOL*: performance ~maximized; not pursued.
  - Austin *et al* 2003: *Razor:* power ~minimized; chip fabbed.
- Plusses:
  - True (or better than) perf. max., resp. power min. [**within** cycle]
- Minuses:
  - Can be costly (TIMERRTOL: 2x cost; Razor: little extra cost)
  - Is complex and hard to design.
  - Can lead to **increase** in cycles → perf. may NOT be true max.

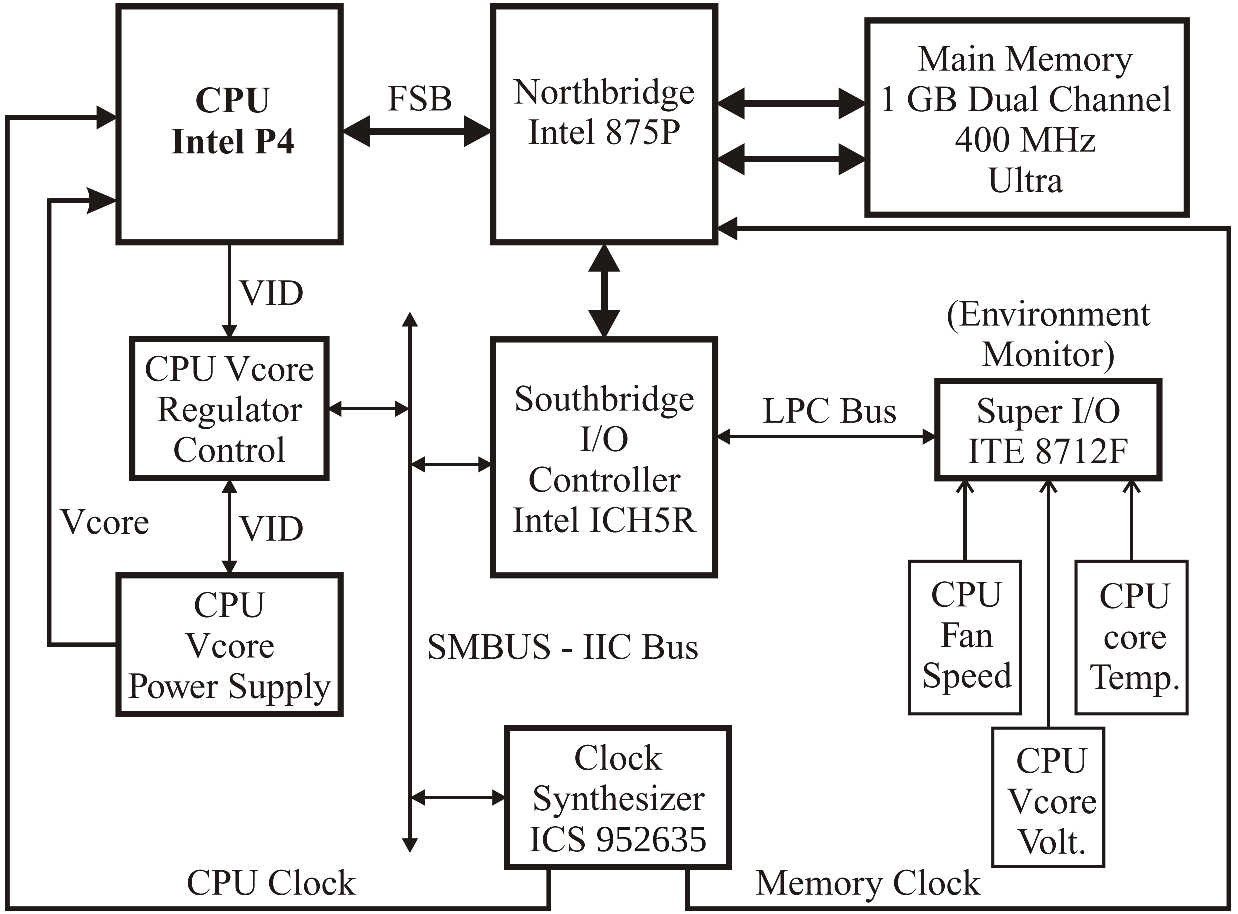
Timing Error Avoidance

- Operation:
  1. *Speed up clock (resp. reduce voltage) ‘til* ***just before*** *error occurs.*
  2. *Slow down clock (resp. increase voltage) for no error.*
  3. *Repeat: GOTO 1.* *[No backtracking or repair needed.]*
- Examples:
  - Olivieri, et al, 1999 – used microcontroller.
  - Uht 2003: *TEAtime*: ~performance maximization; prototype built.
    - *One-bit wide slowest-path test logic always errs before real logic does.*
- Plusses:
  - Close to true performance maximization, resp. power minimization.
  - Simple, easy to design and build. (TEAtime)
  - No increase in cycles.
- Minuses:
  - Shorter papers. *(no, wait, that’s a plus…)*

Adaptive Control Systems

- TEAtime:
  - – Simple up/down control system – 0 delay.
- Skadron, Bahar:
  - – Classic feedback control system.
  - – In hardware.
- TEAPC, TEA42:
  - – Bigger delays.
  - – State-space feedback control system.
  - – In software; little overhead.

TEAPC Block Diagram



(FSB - Front Side Bus)

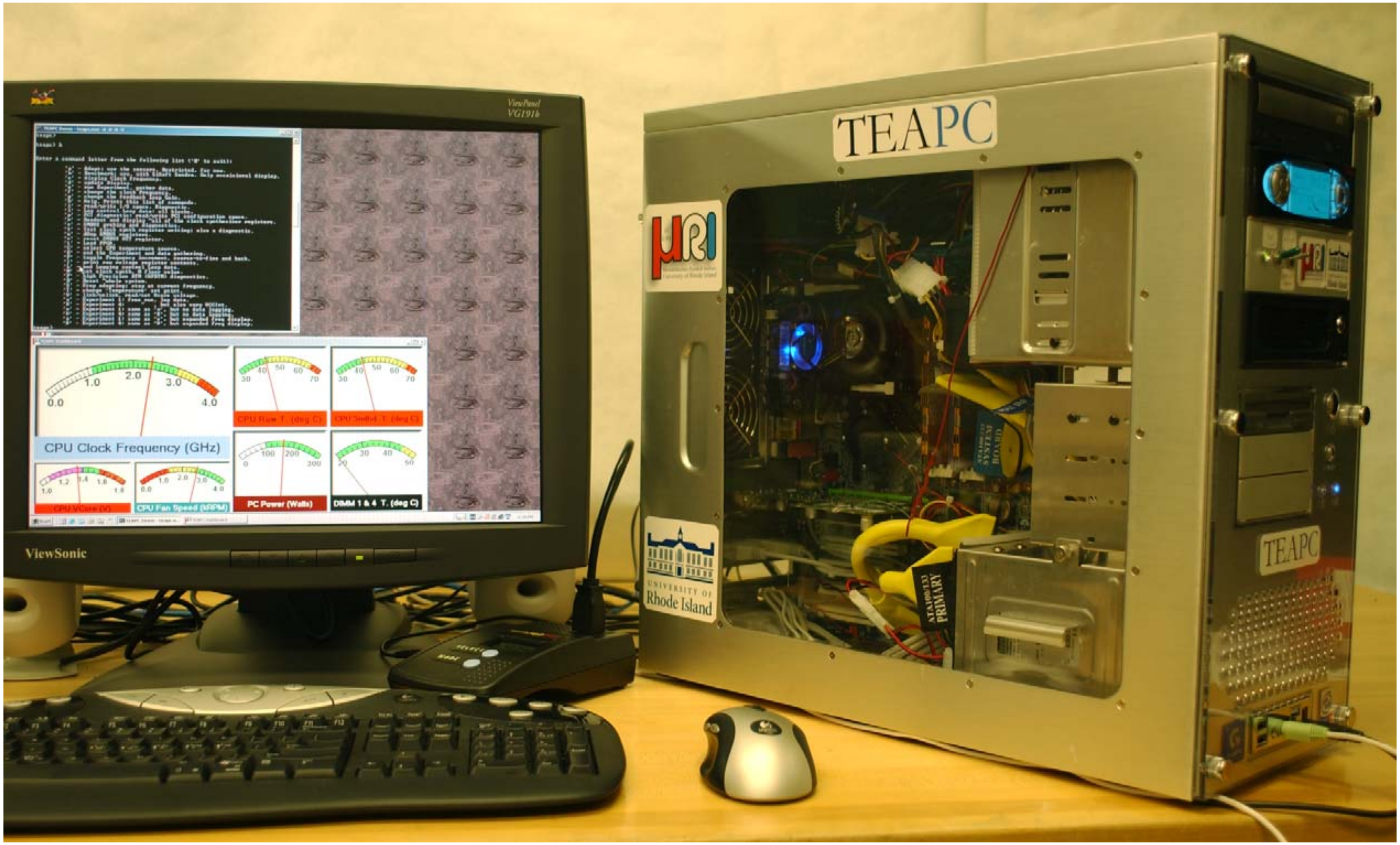
Only directly relevant components and connections are shown.

(LPC - Low Pin Count)

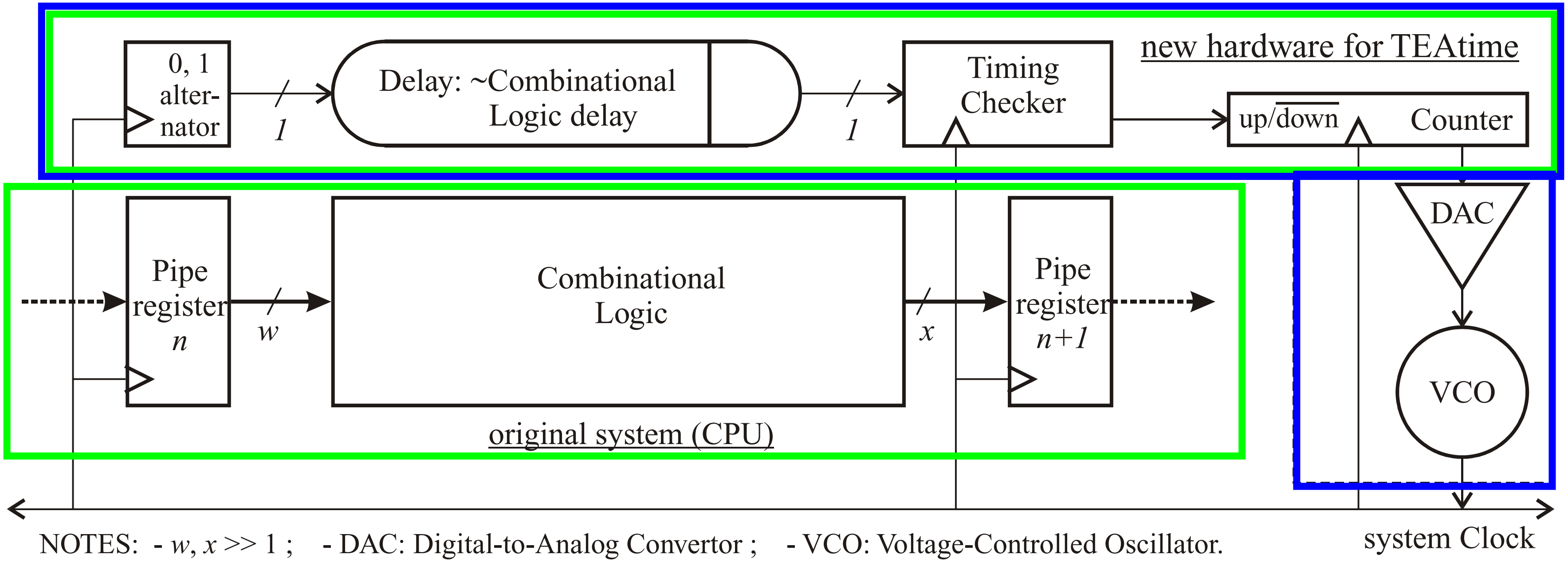
TEAPC Components

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)

Experiment Setup



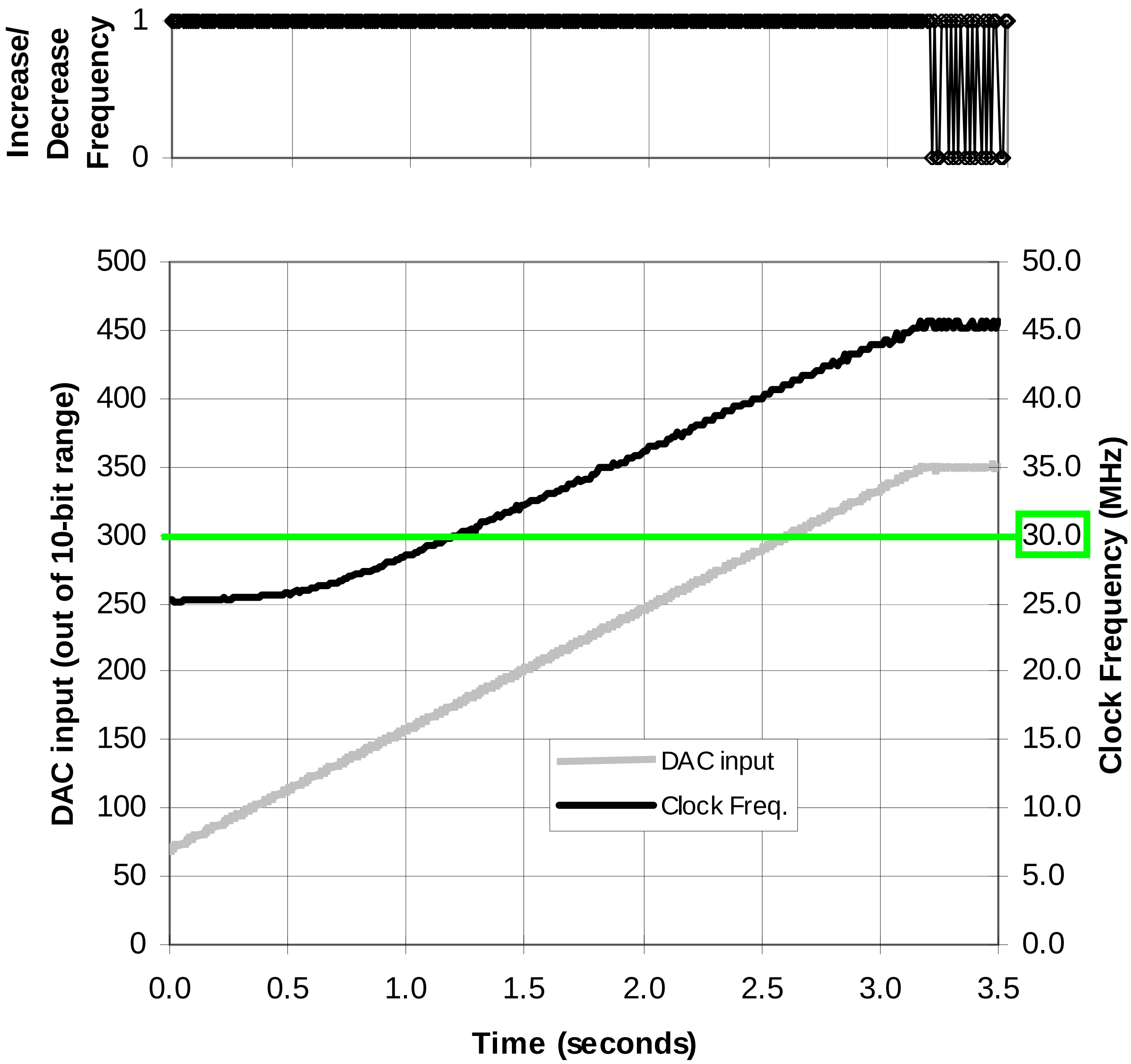
TEAtime Block Diagram



NOTES: - *w*, *x* >> 1 ; - DAC: Digital-to-Analog Convertor ; - VCO: Voltage-Controlled Oscillator.

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

Frequency & DAC setting vs. time (Temp. = 22° C.)



TEAtime Summary

- Performance of prototype improves by >= 34%
- TEAtime provides adaptable frequency control of any synchronous digital system
- Always ~maximizes performance
- Very cheap
- Very easy to add to existing or future designs
- ~Can be adapted to physically existing systems
- ***It Works!!***

TEA42 Disaster Tolerance

- Example: CPU Fan dies….
- Changes (automatic, via feedback system):
  - Freq: 3.15 GHz → 1.5 GHz
  - Vcore: ~1.5 V. → ~1.1 V.

  → Power: ~160 W. → ~100 W. (~37% savings)
- CPU temperature stabilizes at safe value (with this CPU).
- System still works.