TEAtime:
Timing Error Avoidance for Performance Enhancement and Environment Adaptation

Augustus K. Uht
Dept. of Electrical and Computer Engineering

µRI Road Show, Spring 2003

Note, Acknowledgements

TEAtime = Timing Error Avoidance:

– *Speed up clock 'til just before error would occur*

Work supported by:

- U.S. National Science Foundation
- URI Office of the Provost
- Mentor Graphics
- Xilinx

Outline

1. Motivation
2. Related Work
3. TEAtime's Essence
4. TEAtime Features
5. TEAtime Block Diagram
6. Experimental Setup
7. Experimental Results
8. Demo
9. Summary

Motivation

- GOALS: improve performance, and make digital systems adapt to varying conditions
  - Example of latter: adapt to temperature changes from one mobile Internet location to another, always getting 'best' performance
- Digital systems are designed to accommodate worst-case conditions, environmental and manufacturing
- Under typical conditions, could run much faster if one could guarantee no errors would occur
- TEAtime does this; that is, if specified highest allowable clock frequency is ***f***, TEAtime allows operation at frequencies >> ***f***

Related Work

- Olivieri, et al, 1999 – used microcontroller
  - Increased ***f***, every so often checked sample calculation of adder, adjusted clock if necessary
  - Lost time
  - Not readily adaptable to non-adder circuits
- Asynchronous systems – subsumed by TEAtime
- TIMERRTOL – our prior work in the area:
  - Actually detected regular logic's errors and replaced them with correct results
  - Truly maximized performance, but:
  - Very costly (cost more than doubles), hard to design with
- Other systems: open loop (no error detection), not built, metastability issues

TEAtime's Essence

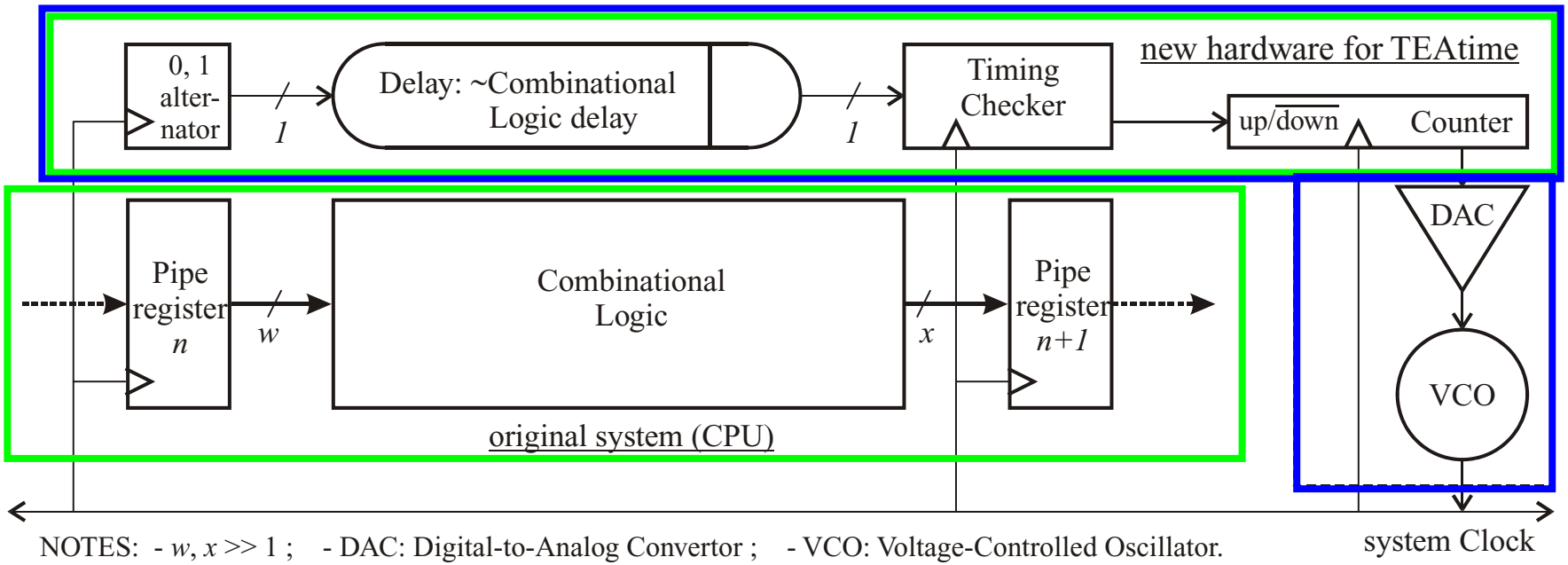
- Processor controls system clock frequency
- Every cycle, clock frequency is adjusted up or down by a 'notch'
- Dummy, or *Tracking*, logic mimics worst-case delay through slowest part of processor, + safety margin

1. Clock frequency is increased until error occurs at output of Tracking logic
2. While error condition exists, decrease clock frequency, a notch at a time
3. GOTO 1.

TEAtime Features

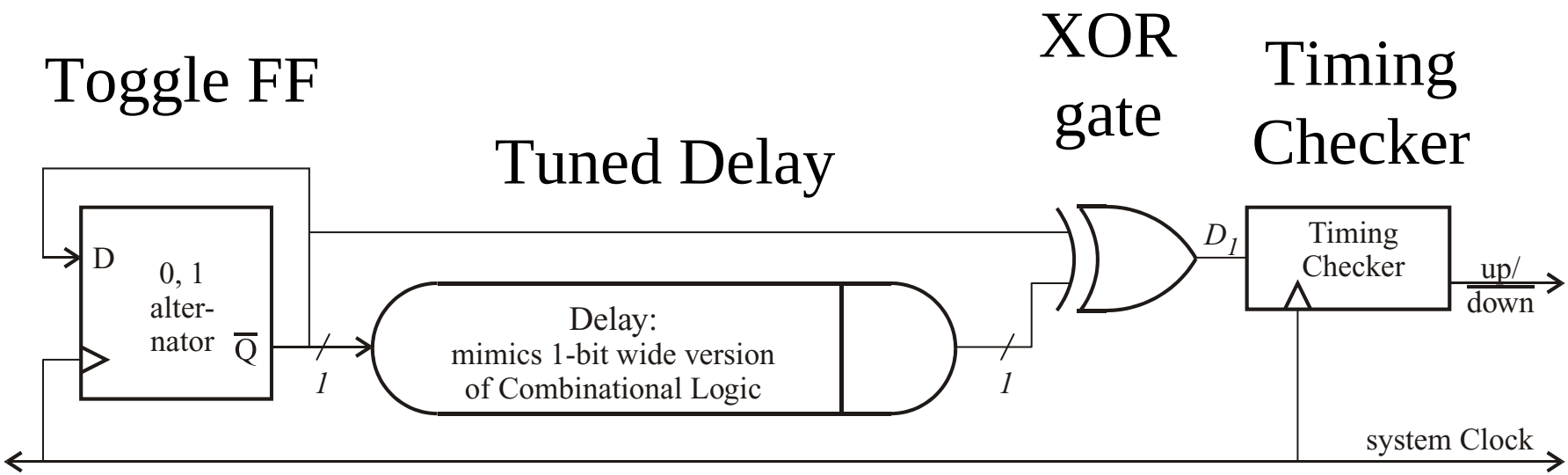
- Tracking logic's attributes mirror real logic's attributes
- Adapts to current environmental conditions, always giving 'best' performance possible
  - Temperature goes up → frequency goes down
  - Temperature goes down → frequency goes up
    - Can artificially cool device to improve performance
- Also adapts to manufacturing conditions extent at processor chip build time, e.g., differing chip qualities
- Dirt cheap – just a few gates and a variable oscillator
- Readily adaptable to virtually all synchronous digital systems (possibly even some of those already built)

TEAtime Block Diagram



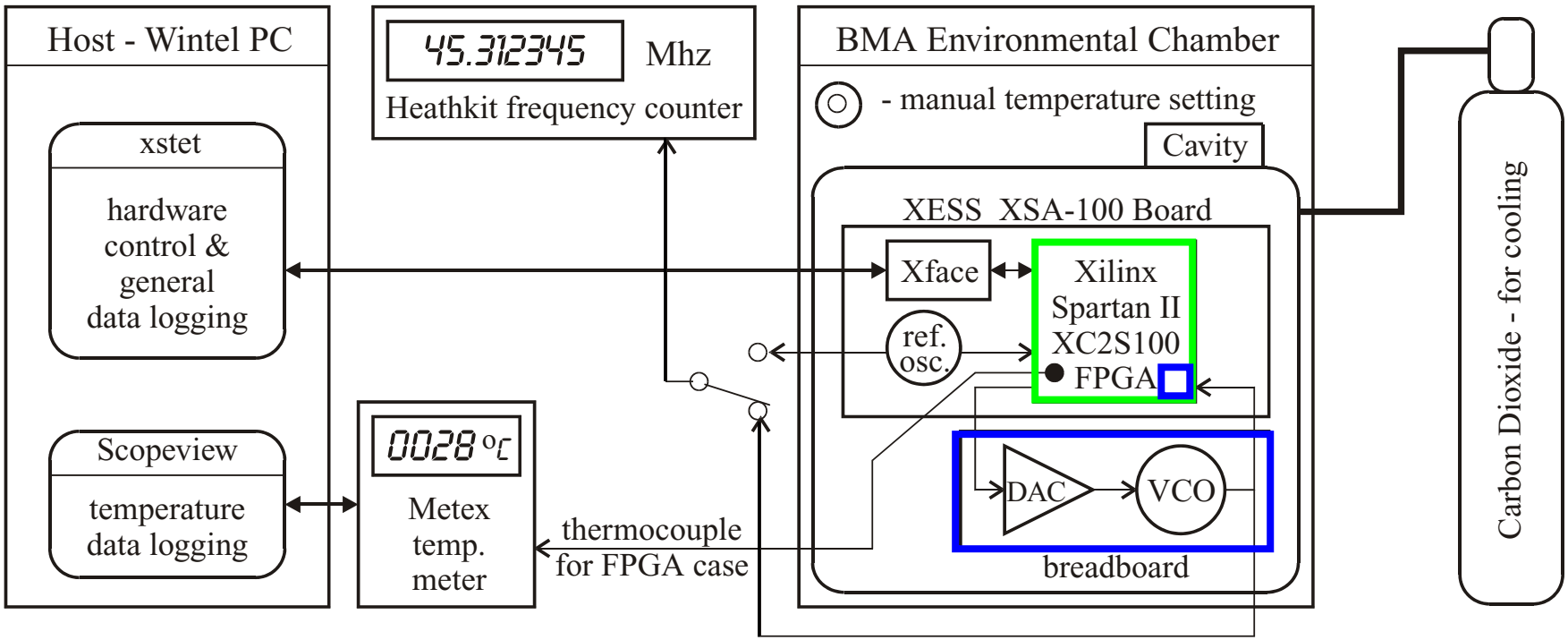
- Timing Error Avoidance system
  - Blue: TEAtime hardware
  - Green: on FPGA

TEAtime Control



- Toggle FF exercises 0→1 and 1→0 transitions
- XOR gate → constant logic polarity to D_1

Experimental Setup



- Blue: TEAtime hardware; - Green: FPGA

Experimental Protocol

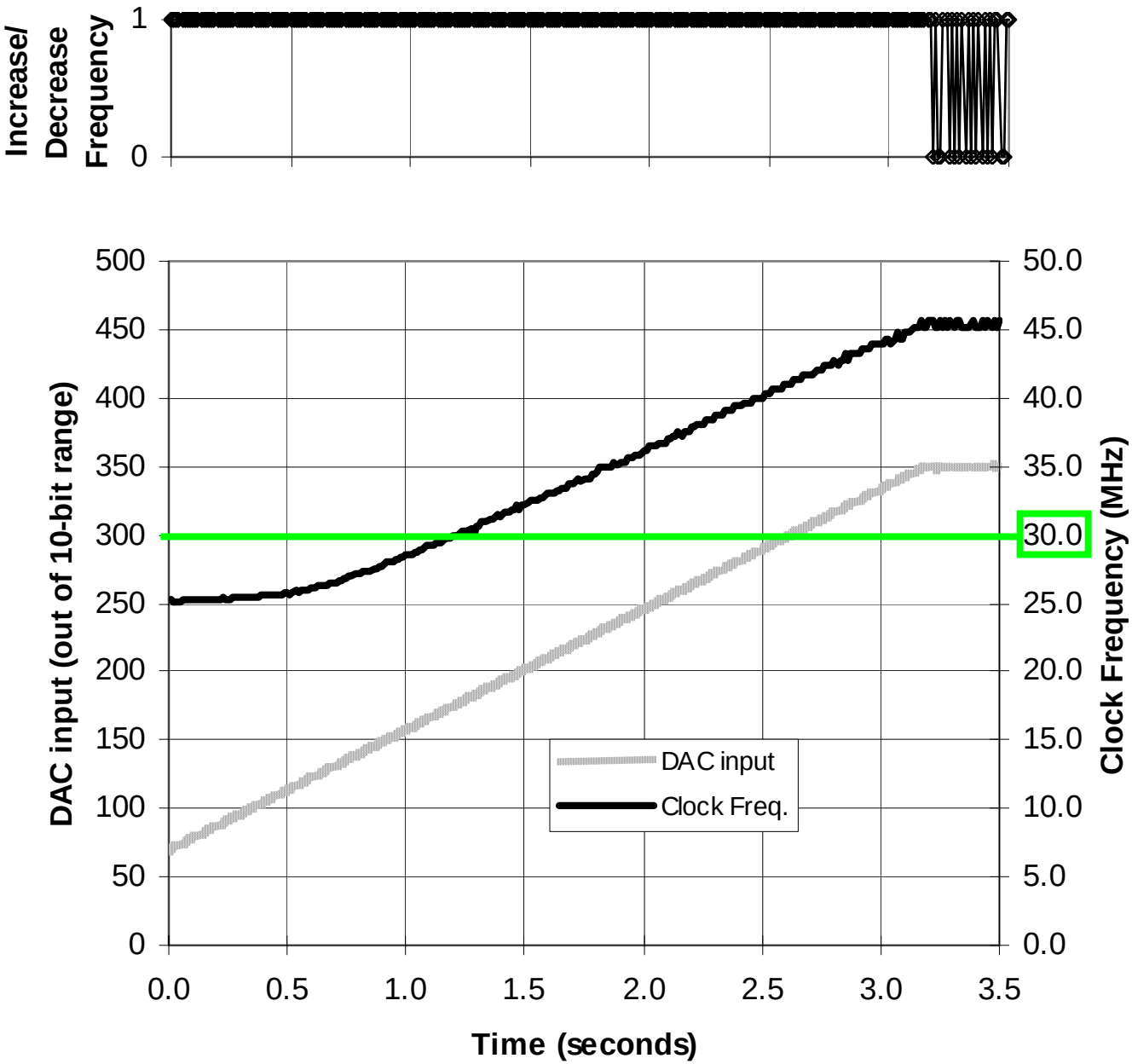
1. Load and configure FPGA
2. Load program into CPU's memory
3. Initialize clock frequency
4. CPU runs program, alters clock frequency
5. Host checks program results
6. Host resets program results (to bad data)
7. Frequency, temperature, etc. data logged
8. (GOTO 4.)

Experiments

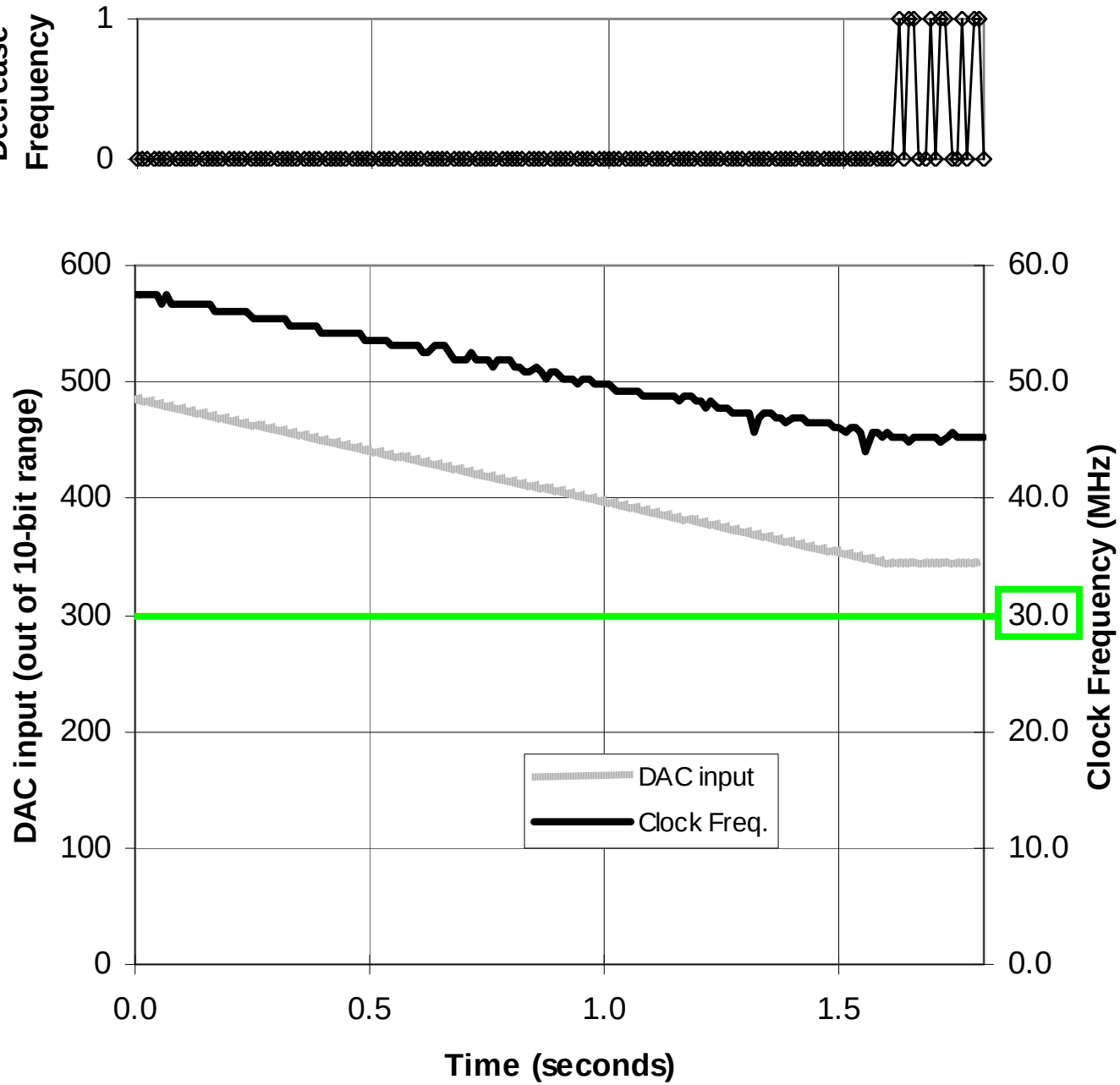
- Normal maximum ***f*** ~ 30 MHz (baseline)

1. Basic operation – ***f*** up and ***f*** down
   - higher frequency, stabilization
2. Frequency as a function of temperature
   - frequency adapts to temperature
3. Frequency as a function of supply voltage
   - frequency adapts to VCCInt

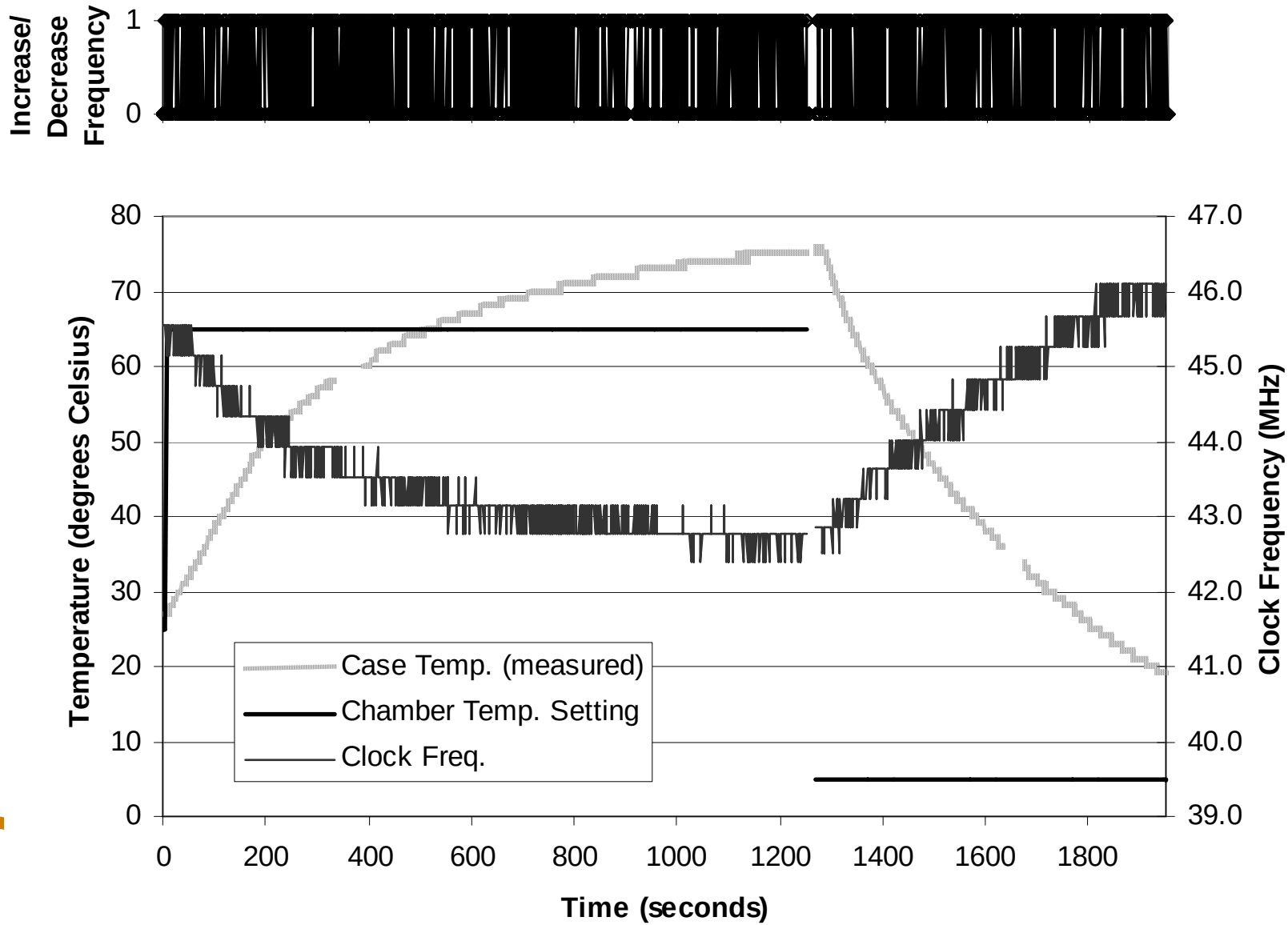
Frequency & DAC setting vs. time (Temp. = 22° C.)



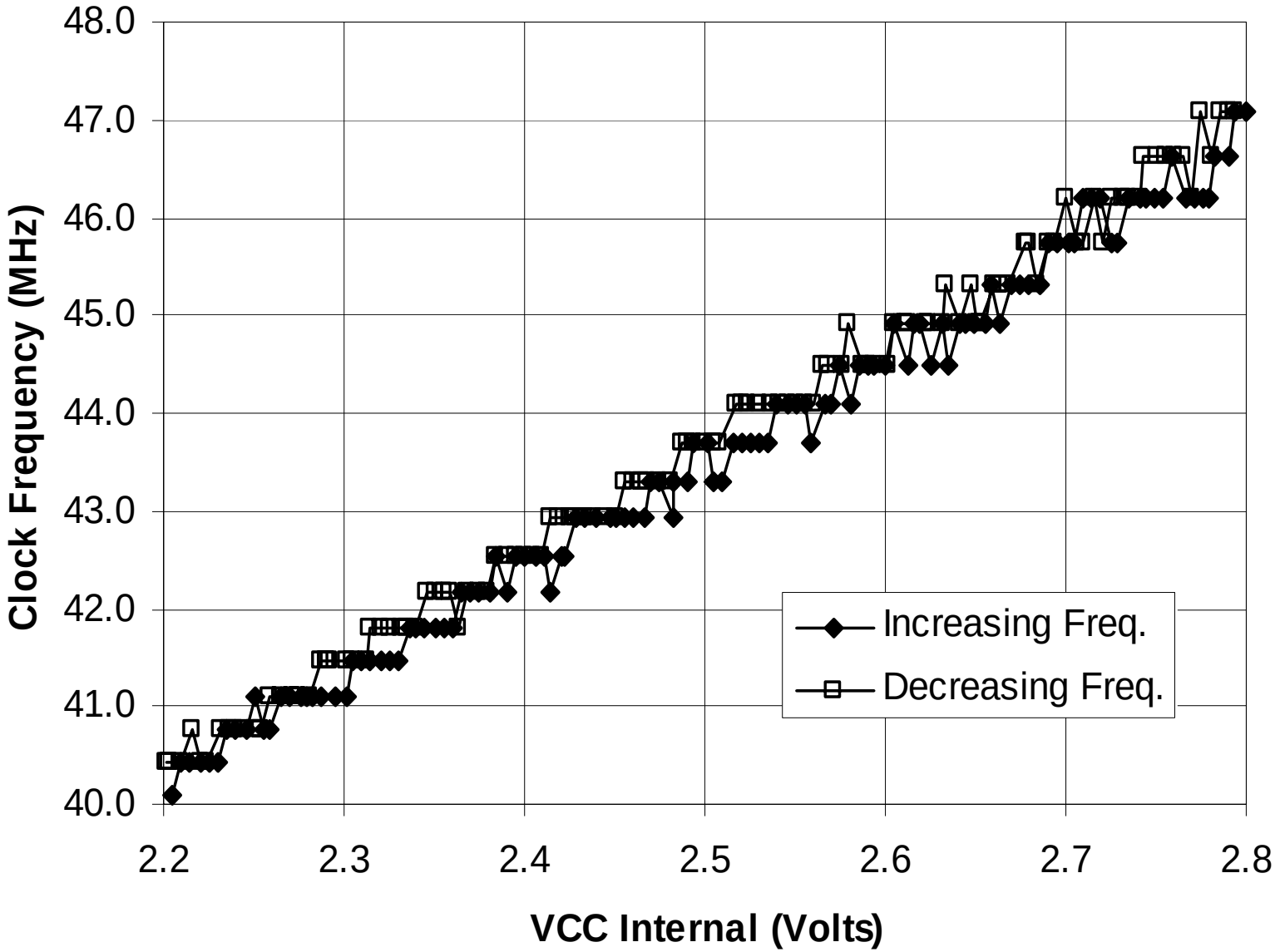
Frequency & DAC setting vs. time (Temp. = 22° C.)



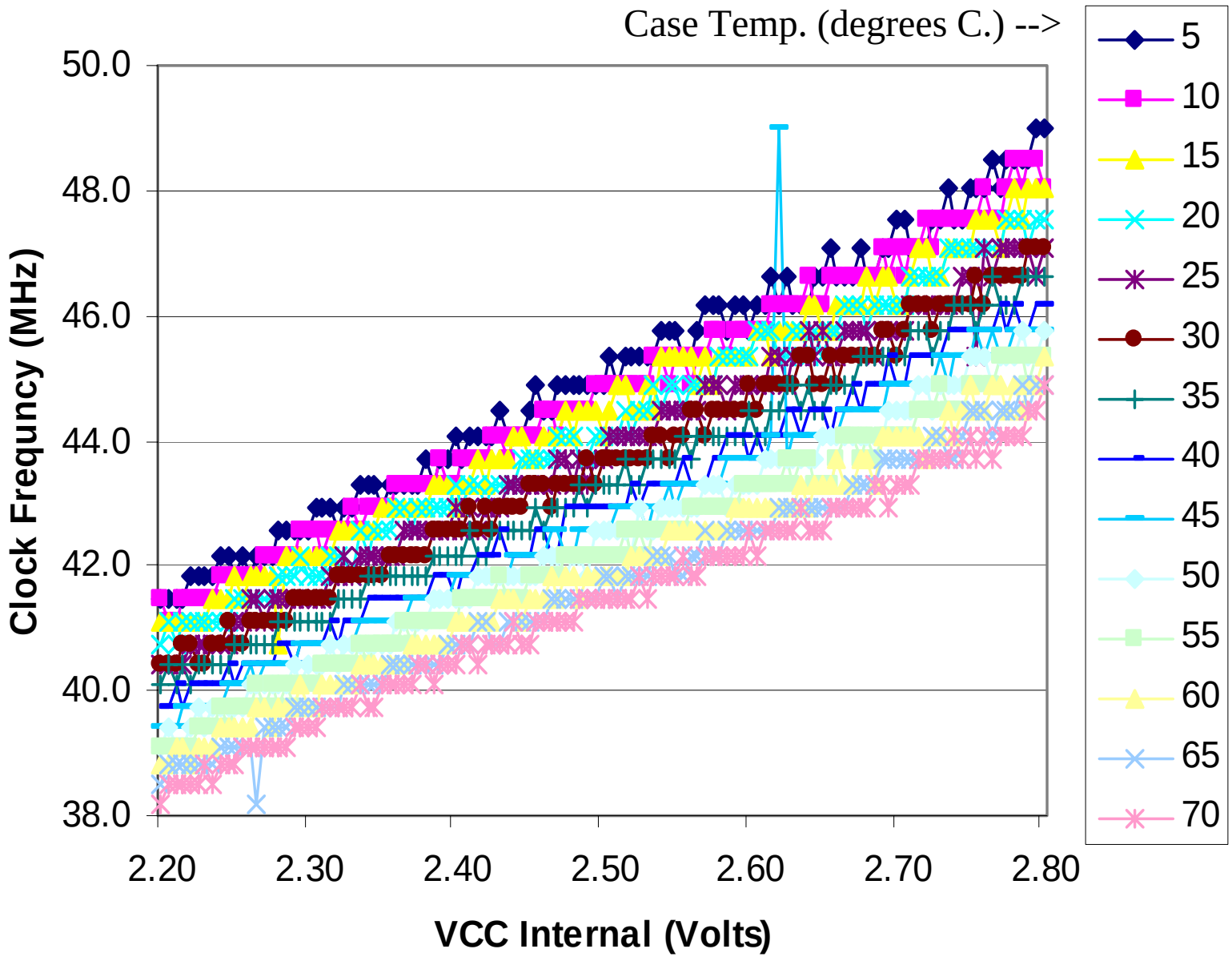
f & Case T vs. time



Frequency vs. VCCInt



Freq. vs. VCCInt vs. Temp.



(Requisite) Low Power Note

- Measured average power: PCCInt
- (Power proportional to $V^2 f$)
- → reducing VCCInt from 2.5 V. to 2.2 V. –
  - Reduces performance (time) by: 7.7%
  - For a Power (norm. to f) reduction of: **26.8%**
  - Or an absolute Power reduction of: **31.9%**
- Pretty good tradeoff!
- Opportunity here for safe, dynamic, and performance-adaptive power reduction

Tuning the Tracking Logic

- Recall: TEAtime gives ~30% perf. improve.
- IF: we tune the tracking logic:
  - More closely match the real logic's delay
  - (cost: reduced safety margin)
- Reduce by ~2 gate delay →
  ~43% performance improvement
  - Still tracks
- Almost twice the work done in same time!

Demo

- Recall: normal maximum ***f*** ~ 30 MHz
- Start demo
  (reset scale for Temp., VCCInt change)

Summary

- Performance of test system 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
- May be adaptable to physically existing systems
- ***It Works!!***

TEAtime:
Timing Error Avoidance for Performance Enhancement and Environment Adaptation

Augustus K. Uht
Dept. of Electrical and Computer Engineering

µRI Road Show, Spring 2003