NC STATE UNIVERSITY

TEACHING OPTIMIZATION OF TIME AND ENERGY IN EMBEDDED SYSTEMS

Alex Dean
Dept. of ECE
NC State University

NC STATE UNIVERSITY

Overview

- Motivation
- Course Basics
- Time Optimization
- Power and Energy Optimization
- Conclusions

NC STATE UNIVERSITY

Motivation

- Teach methods for optimizing deeply embedded ***systems***
  - Look at entire system, not just a fragment
- Compelling examples for students
  - Make it run faster
  - Make the battery last longer
- My motivation for this talk
  - Give you ideas on how to structure the programming projects so they work

NC STATE UNIVERSITY

Course Administrivia

- Piggy-backed senior (20%) and grad (80%) courses
  - Most grad students didn't do undergrad at NCSU
- Assume students have some background
  - MCU and peripheral programming in C
    - GPIO, timer, ADC, DAC, UART,
    - Interrupts
- Need to teach hands-on material practically, given 102 students and two teaching assistants

NC STATE UNIVERSITY

Course Contents

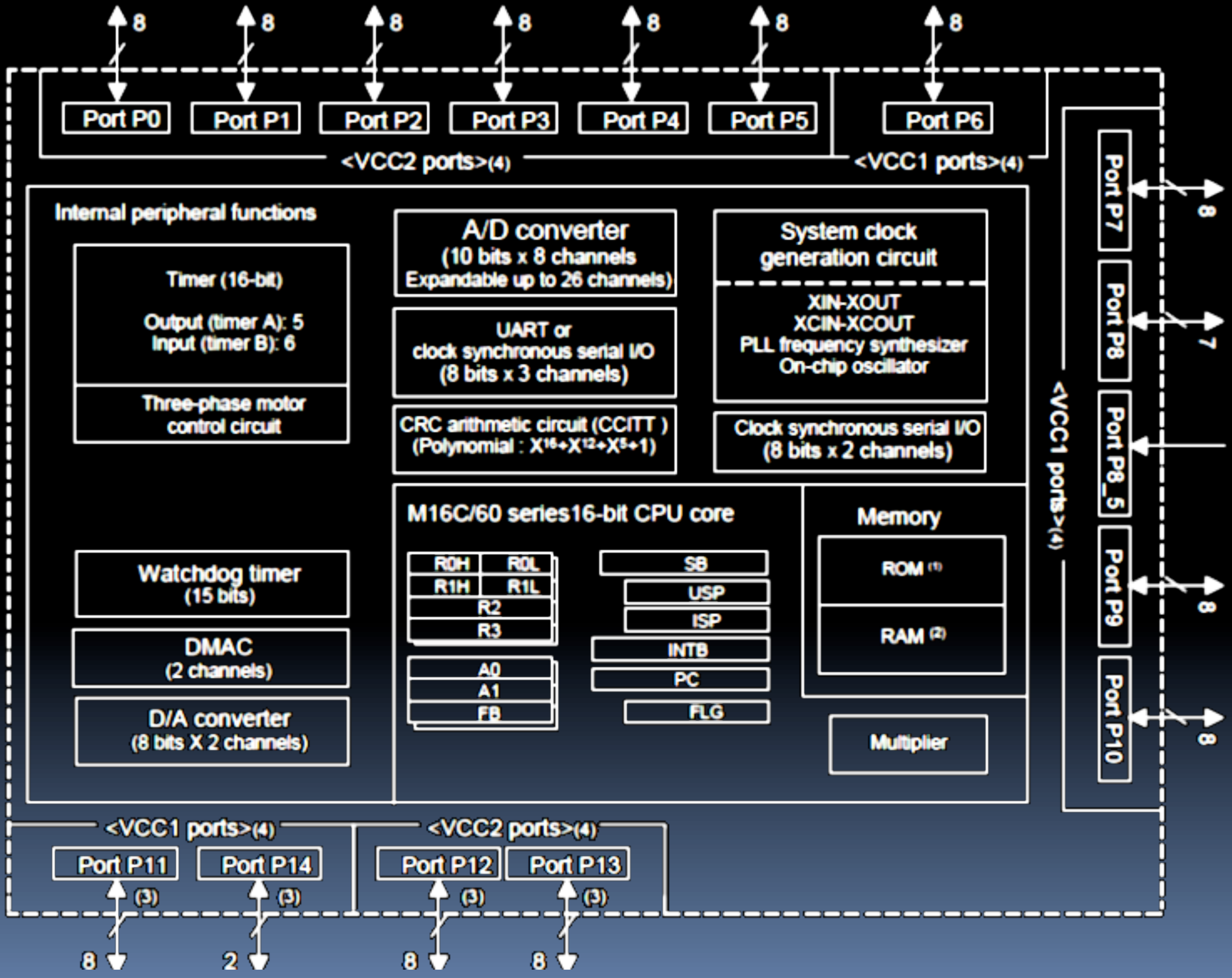
- MCU, ISA and development board (QSK) introduction
- Interfacing projects: analog and digital, timers, serial communications
- Run-time Performance Analysis and Optimization + project
- Energy Optimization + project
- Schedulers (task ordering, preemption)
- Real-time Systems
- Application design with an RTOS
- Lightweight Software Engineering for Embedded Systems
- Embedded code porting project (TFT LCD, FAT on microSD card)
- Final project

NC STATE UNIVERSITY

Development Platform

- Renesas M16C architecture MCU
  - 16-bit CISC, 24 MHz
  - 64 KB on-chip flash, 10 KB SRAM
  - Rich set of peripherals
- QSK62P+ board
  - USB for power, debugger
  - LCD, thermistor, pot, switches, LEDs, RS232, many expansion headers
- Good tool support
  - Source-level debugging
  - Much easier to use than GNU toolchain
- Cheap enough - $70 at Digikey
  - Students purchase the board, keep it
  - Benefits of ownership

MCU Core and Peripherals


8
Port P0
Port P1
Port P2
Port P3
Port P4
Port P5
Port P6
<VCC2 ports>(4)
<VCC1 ports>(4)
Internal peripheral functions
Timer (16-bit)
Output (timer A): 5
Input (timer B): 6
Three-phase motor control circuit
A/D converter (10 bits x 8 channels Expandable up to 26 channels)
UART or clock synchronous serial I/O (8 bits x 3 channels)
CRC arithmetic circuit (CCITT) (Polynomial : X16+X12+X5+1)
System clock generation circuit
XIN-XOUT
XCIN-XCOUT
PLL frequency synthesizer
On-chip oscillator
Clock synchronous serial I/O (8 bits x 2 channels)
Watchdog timer (15 bits)
DMAC (2 channels)
D/A converter (8 bits X 2 channels)
M16C/60 series16-bit CPU core
R0H R0L
R1H R1L
R2
R3
A0
A1
FB
SB
USP
ISP
INTB
PC
FLG
Memory
ROM (1)
RAM (2)
Multiplier
Port P7
Port P8
Port P8_5
Port P9
Port P10
<VCC1 ports>(4)
7
<VCC1 ports>(4)
<VCC2 ports>(4)
Port P11
Port P14
Port P12
Port P13
(3)
2

NC STATE UNIVERSITY

Optimization Lessons

- Be a slacker - Avoid all run-time work possible
  - Pre-compute when possible
  - Reuse computation results
  - Avoid unnecessary conversions
- Do whatever remains efficiently
  - Start with efficient algorithm and problem representation, then work down into implementation details
  - Avoid double-precision math
  - Use low-complexity algorithms
- Be scientific: let the data drive your activities
  - Identify functions which dominate time
  - Evaluate impact of optimization attempts

Time Optimization

- Motivation
  - Students often blindly optimize what they think is slow, rather than what really is slow
  - Students often skip ahead to low-level optimizations, forgetting about the big picture
    - Algorithm -> implementation -> tweaks (convincing the compiler to be good)
    - Our other classes focus on the detailed instruction-level optimizations
- Methods
  - Profile-driven optimization
  - Program rich with algorithmic optimization opportunities

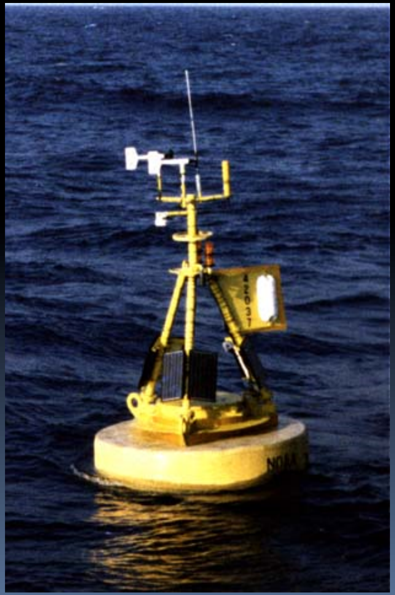
Profile-Driven Optimization

- Sample program counter periodically with timer interrupt, increment corresponding function's count in statistics table
- ISR examines return address on stack
- Students use gawk script to create region address look-up table based on map file
- Does not require instrumentation of application code (more like oprofile than gprof)

NC STATE UNIVERSITY

Program for Speed Optimization

- Program requirements
  - Not a kernel (e.g. FFT)
  - Not a ginormous server or desktop application (SPEC)
- Find distance and bearing to nearest of 163 weather/sea state monitoring buoys
  - Spherical surface (tunnels and submarines are not allowed)
  - distance = acos(sin(lat1)*sin(lat2)+ cos(lat1)*cos(lat2)*cos(Δlon))*R
  - bearing = atan2(sin(Δlong)*cos(lat2), cos(lat1)*sin (lat2) – sin(lat1)*cos(lat2)*cos(Δlon))
- Locations in list of lat/lon coordinates
- Test cases: two “undisclosed locations”



Optimization Opportunities

- Graded on a curve
  - Some of score depends on performance relative to class
- Intentionally provided a program rich with opportunities for optimization - not written badly, but a reasonable, structured program designed for correctness
  - Double-precision math (through math.h)
  - Progress updates on LCD
  - Unnecessary calculations of bearing
  - Repeated conversion of degrees to radians
  - Repeated calculation of sine and cosine of fixed values
  - Repeated calculation of PI/180
  - Alphabetically sorted data (not geographically sorted)

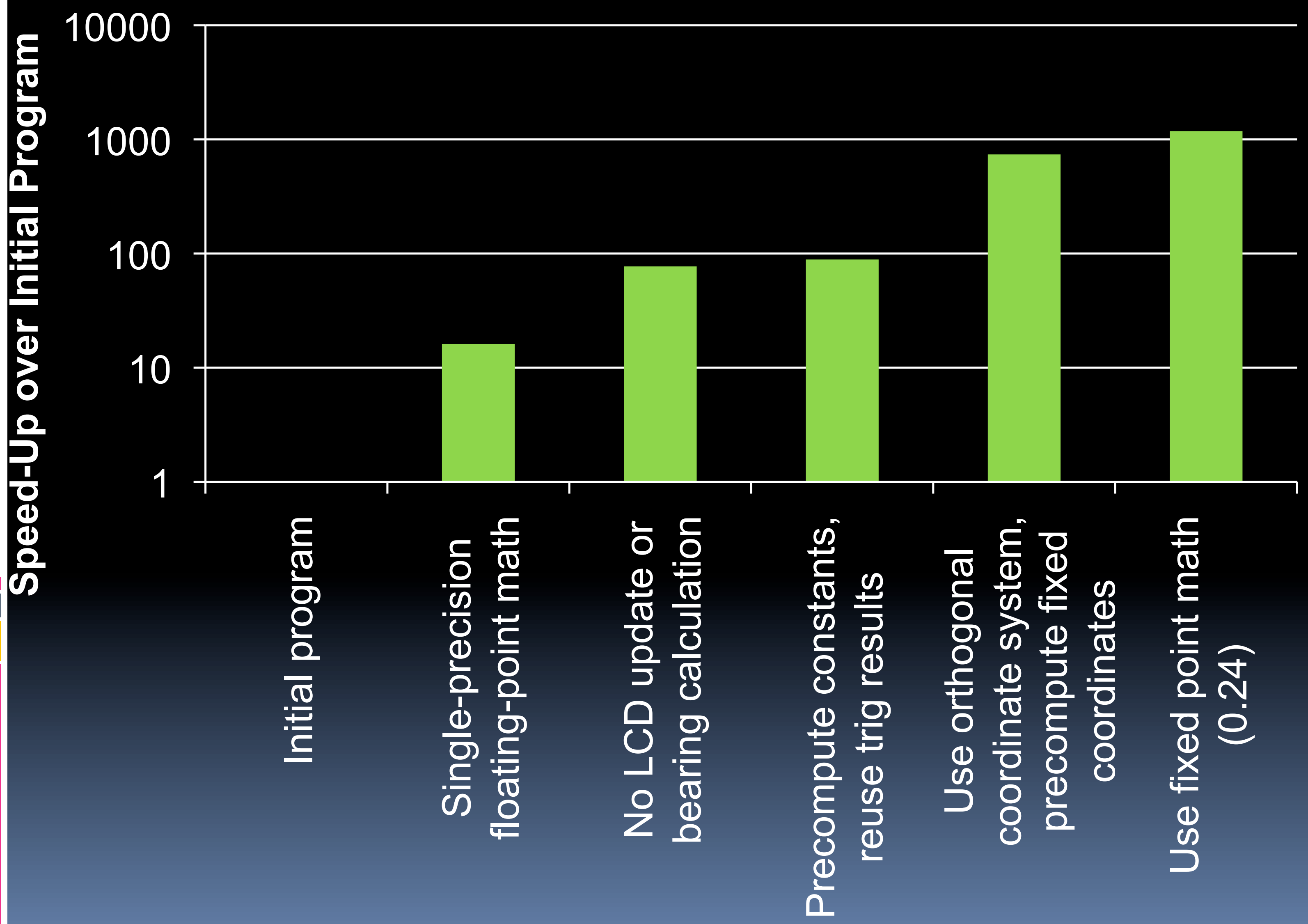
NC STATE UNIVERSITY

Project Post-Mortems

- Always hold these after projects
  - Great way to get many different ways of looking at a problem (crowdsourcing)
- Informal class discussion
  - How did you get such good performance?
  - What worked best?
  - What didn't work well?
  - How would you change what you did, if you could repeat it?
  - How would you change the project assignment for next time?
- Lab reports include lessons learned section

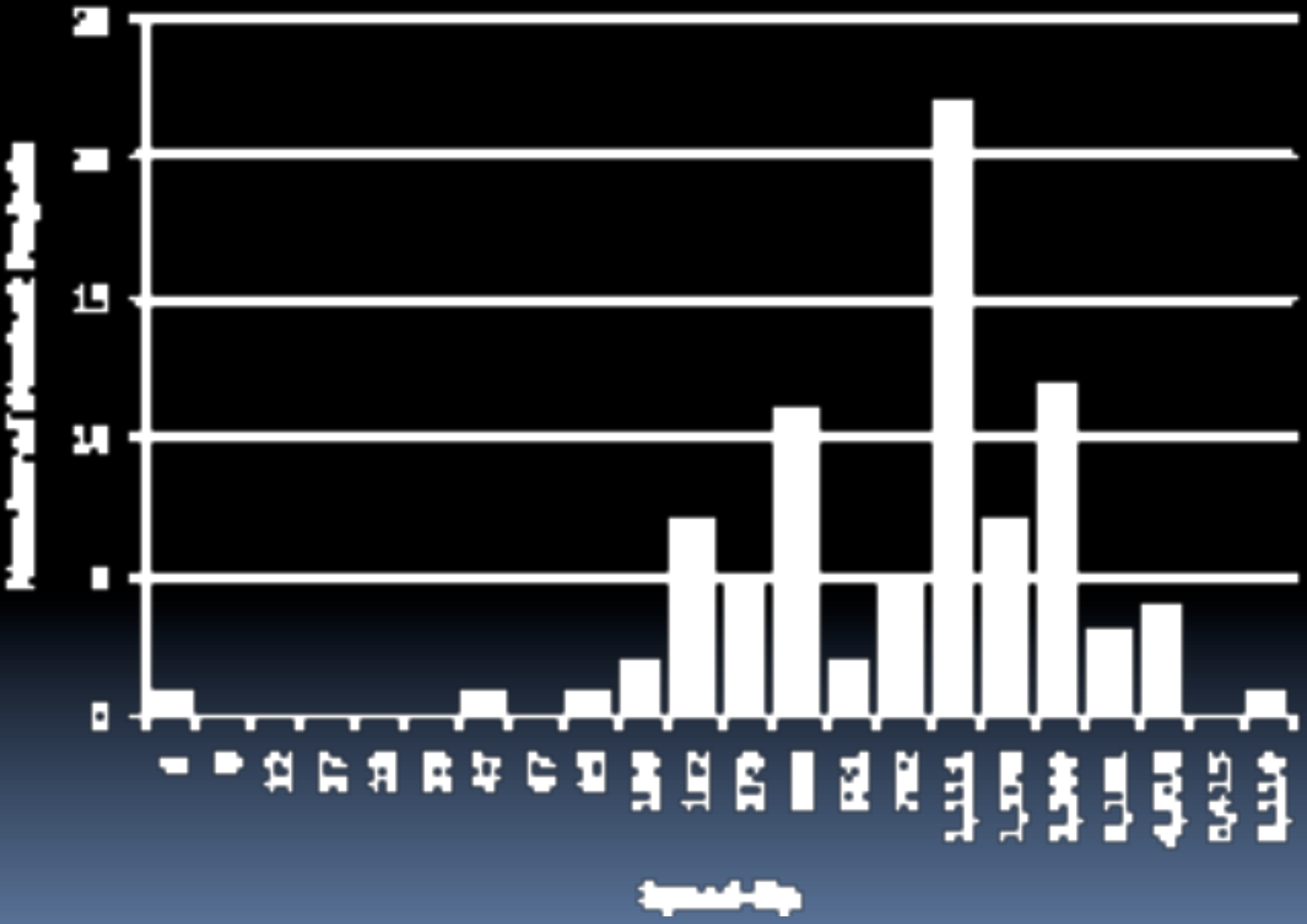
NC STATE UNIVERSITY

One Student's Optimizations



NC STATE UNIVERSITY

Overall Speed-Up Results



NC STATE UNIVERSITY

Power and Energy Optimization

- Challenge: how to measure power or energy consumption?
  - Should be inexpensive, fast, repeatable, sensitive
- Initial idea - build a cheap power-meter/integrator
  - Use op-amp circuits to multiply current by voltage, then integrate over time, then sample with MCU ADC.
  - Need to consider integrator drift, integrator reset, range, software support
- Solution - how long can a supercap power the circuit?
  - Cheap and easy, breadboardable
  - Elegant energy calculation:

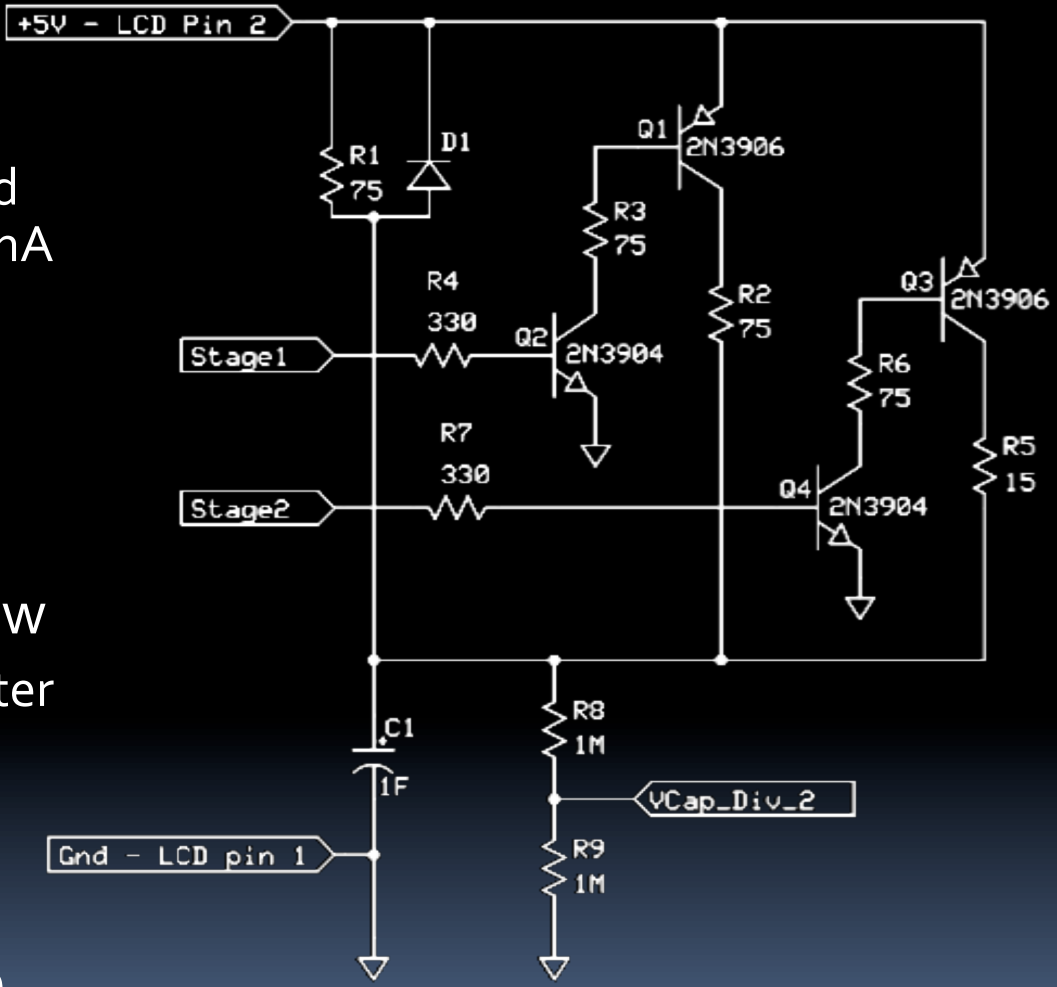
$$E = \frac{C}{2}\left(V_1^2 - V_0^2\right)$$

  - Provide a supercap to each student, but use one reference supercap for grading purposes

NC STATE UNIVERSITY

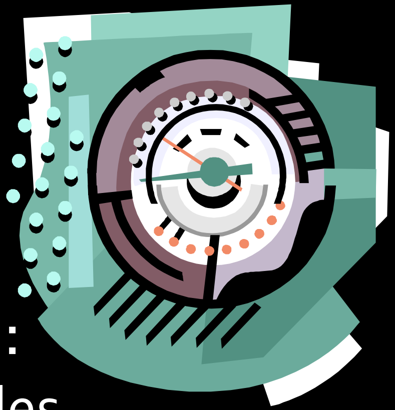
Supercapacitor Circuit

- Use 1F 5 V supercap
- Current limiting required
  - QSK is USB-powered -> need to keep current under ~125 mA
  - R1 provides initial current limiting
- D1 bypasses R1 when I > 8 mA
- Exponential charging = slow
  - R3 and R6 get switched in later (under software control) to reduce charge time
- Monitor supercap voltage with resistive divider



NC STATE UNIVERSITY

Program for Energy Optimization



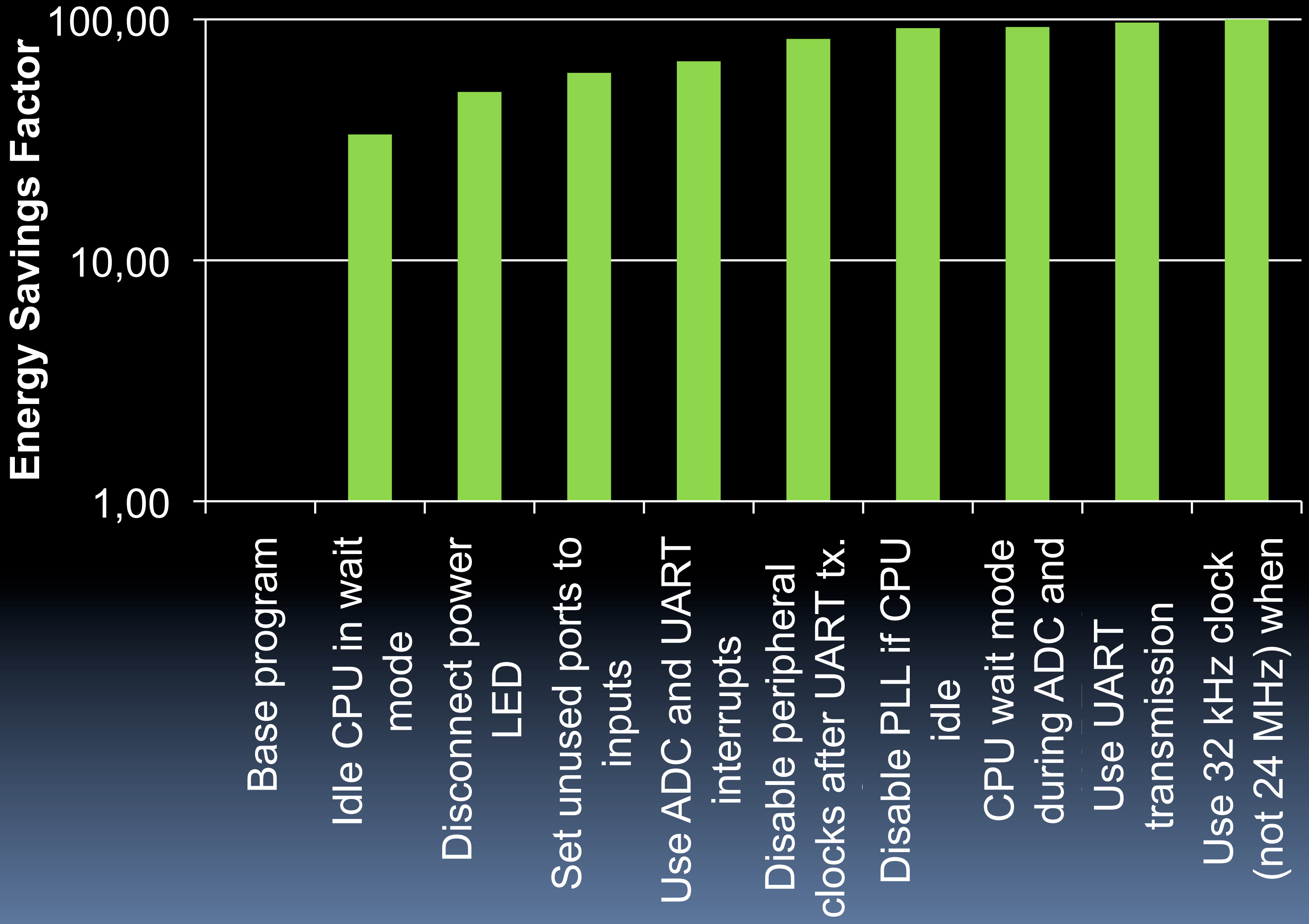
- Program requirements
  - Idle time for CPU to sleep
  - Easy testing and measurement
- Device reports temperature every second:
  - Read thermistor with ADC in burst of 10 samples
  - Average the samples
  - Convert to formatted text string ("77.3F")
  - Send string out UART at 57600 baud (e.g. to Bluetooth)
- Base program runs 3 seconds before supercap gives out
- And as the power fails…
  - LCD stops working below 4.3 V
  - Board forced into reset below 2.7 V by brown-out detector

Energy Optimization Techniques

- All the speed optimizations discussed previously
- Processor
  - Operating modes - covered in application note
  - Clock speeds - from 32 kHz to 24 MHz
- Peripheral devices
  - Can slow or disable clock, remove supply voltage for subsystems
  - Oscillator drive modes
  - GPIOs
- Board devices
  - LEDs
  - LCD

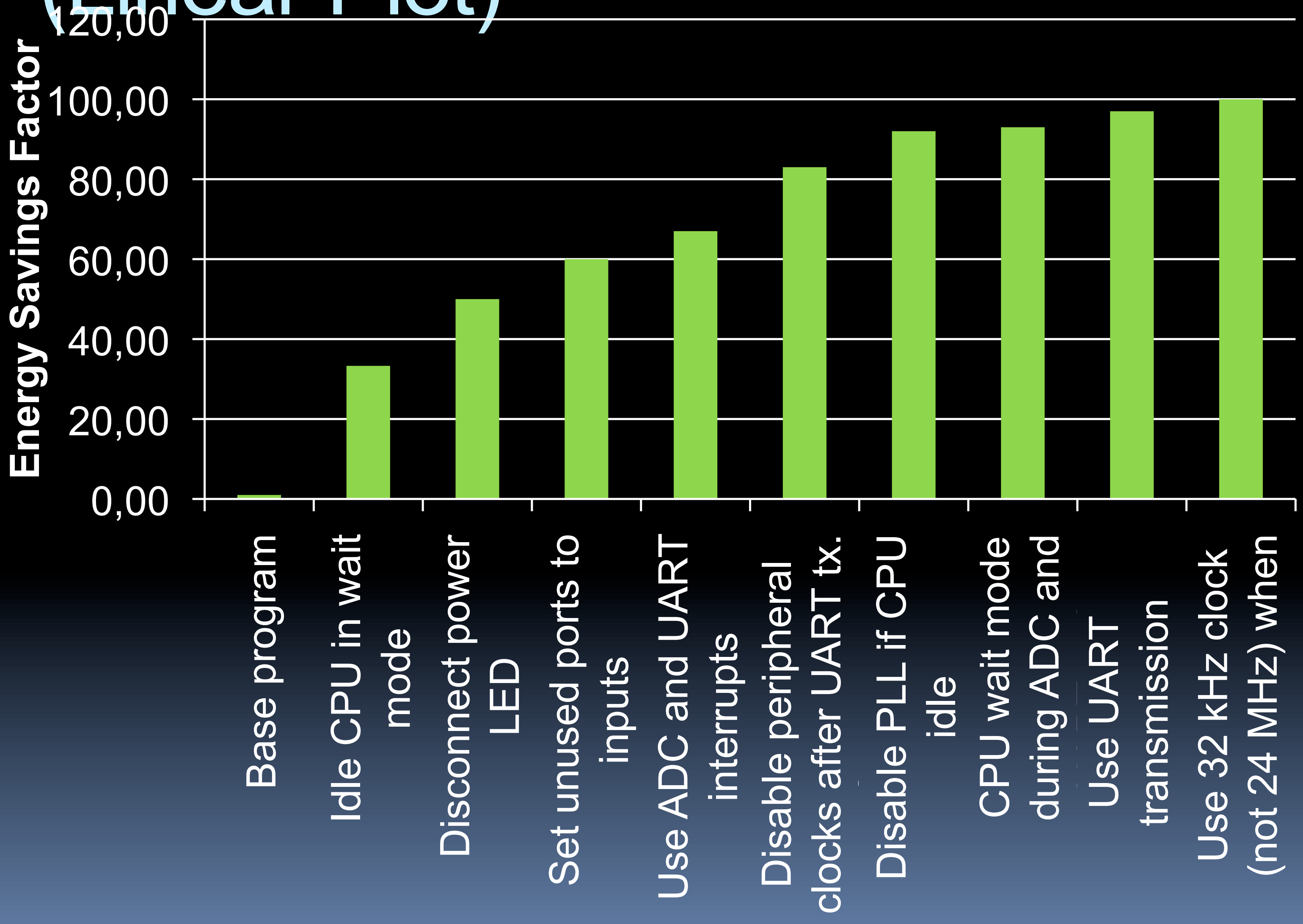
NC STATE UNIVERSITY

One Student's Optimizations



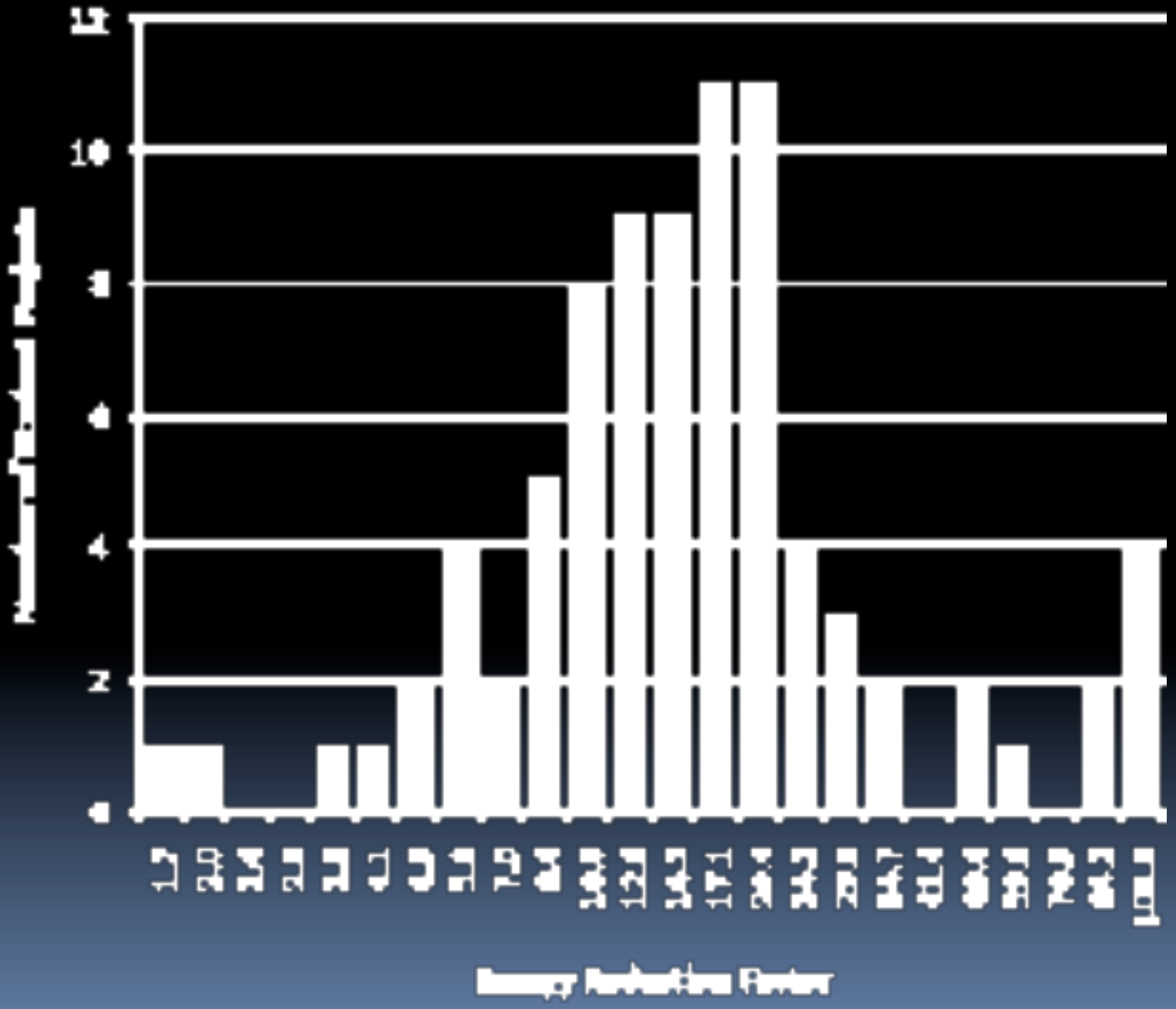
NC STATE UNIVERSITY

One Student's Optimizations (Linear Plot)



NC STATE UNIVERSITY

Overall Energy Savings



NC STATE UNIVERSITY

Student Feedback

- Likes
  - Projects!
  - Whiteboard rather than powerpoint
  - Open-ended nature of projects
  - In-class programming examples
- Dislikes
  - Too many A+'s!
  - Embedded coding style should be taught
  - Too much pre-written code
  - Cover less introductory material, assume students will be able to read manual
  - In-class programming examples

Conclusions and Changes

- Very effective - students were excited by competitive aspect of optimization
- Programs which offer a range of optimizations encourage outside the box thinking

- Changes for next time
  - Switch from supercap to supply rail when discharged, keeping board running
  - Reduce introductory projects
  - Add ARM Cortex-A8 for breadth and comparison
    - $50 processor vs. $5 processor

- Course materials available from me at agdean@ncsu.edu or at Renesas University program at http://renesasrulz.com

NC STATE UNIVERSITY

NCSU's ECE Dept. is Hiring!

- http://jobs.ncsu.edu - position #581
- Dr. Veena Misra ecefacultyhire@ncsu.edu
- Department-wide, junior and senior, multiple slots