Variability-Aware Duty Cycle Scheduling in Long Running Embedded Sensing Systems

Lucas Wanner, Rahul Balani, Sadaf Zahedi, Charwak Apte, Puneet Gupta, and Mani Srivastava

wanner@ucla.edu

www.variability.org
Outline

• Duty Cycle in Sensing Systems
  • Variability in Contemporary Hardware
  • Duty Cycle Scheduling Methods
  • Software Architecture
• Results
• Conclusions
Duty Cycle in Sensing Systems

\[ \uparrow \text{active time} = \uparrow \text{quality of sensing} \]
Maximum Allowable Duty Cycle

Active Power ($P_A$)  Sleep Power ($P_S$)  Energy ($E$)

Lifetime ($L$)

$$DC = f(P_A, P_S, L, E)$$
Maximum Allowable Duty Cycle

\[ DC = f (P_A, P_S, L, E) \]

Datasheet spec:
- Active Power
- Sleep Power

Our objective: OS mechanisms to adapt duty cycle to measured hardware signatures

DC: Duty Cycle

- \( P_A \): Active Power (W)
- \( P_S \): Sleep Power (W)
- \( L \): Lifetime (s)
- \( E \): Energy (J)
Outline

• Duty Cycle in Sensing Systems
• Variability in Contemporary Hardware
• Duty Cycle Scheduling Methods
• Software Architecture
• Results
• Conclusions
Causes of Variability

• Difficulties in Semiconductor Manufacturing with scaling of physical dimensions
  • WYS is not WYG
• Vendor Differences
  • Multi-sourcing of parts
• Ambient Conditions
  • Extreme voltage/temperature environments especially for sensors
• Aging
  • Temporal stress, e.g., due to NBTI
Active Power Variability

ARM Cortex M3 Active Current (Room Temperature)

Processor Instance

Current (mA)

Atmel SAM3U4E Cortex M3
Active Mode, 4MHz Internal Oscillator
Room Temperature
Active Power Variability

ARM Cortex M3 Active Current (Room Temperature)

< 10% Variation

Atmel SAM3U4E Cortex M3 Active Mode, 4MHz Internal Oscillator Room Temperature
Sleep Power Variability

ARM Cortex M3 Sleep Current (Room Temperature)

Atmel SAM3U4E Cortex M3
Sleep Mode, 32KHz Slow Oscillator
Room Temperature
Sleep Power Variability

ARM Cortex M3 Sleep Current (Room Temperature)

9x Variation

Atmel SAM3U4E Cortex M3
Sleep Mode, 32KHz Slow Oscillator
Room Temperature
Sleep Power Variability Across Temperature

![Graph showing the relationship between temperature and power variability for different participants.](image-url)
Sleep Power Variability Across Temperature

- 14x Variation
- 30% Variation
- 4x Variation

Current (μA)

Temperature
Outline

• Duty Cycle in Sensing Systems
• Variability in Contemporary Hardware
• Duty Cycle Scheduling Methods
• Software Architecture
• Results
• Conclusions
Duty Cycle Scheduling

- Duty cycle is a function of
  - Active Power
  - Sleep Power
  - Application Lifetime
  - Available Energy

$$DC = f (P_A, P_S, L, E)$$

How to determine maximum duty cycle when $P_A$, $P_S$ vary with instance and temperature?
Duty Cycle based on Datasheet Spec

- Use $P_A$, $P_S$ from datasheet

The graph compares the actual current (21°C) with the datasheet spec for different processor instances. The actual current is lower than the datasheet spec for most instances, indicating that the device will not meet lifetime requirements and will leave energy untapped.
Duty Cycle based on Worst-Case Power

- Use worst case $P_A$, $P_S$ across all instances and target temperature

![Graph showing current vs. temperature with all nodes leaving energy untapped at high temperatures]
Optimum Duty Cycle

- Determine $P_A$, $P_S$ as a function of instance and temperature

Temperature Profile: Death Valley, CA, 2009
Optimum Duty Cycle

- Determine $P_A$, $P_S$ as a function of instance and temperature
- Duty cycle is determined for each instance, time unit
- Linear program that maximizes active time subject to energy constraint
Optimum Duty Cycle: Temperature Profile

- Exact temperature profile is hard to predict
  - Temperature distribution is simpler
  - Duty cycle is determined for each instance, temperature
Variability-Aware Uniform Duty Cycle

- Determine $P_A$, $P_S$ for each instance across a temperature profile
  - “Weighted mean” of power across temperature for an instance

- When the difference ($P_A - P_S$) is constant across temperature, this method yields the optimum duty cycle
  - True in practice for current generation embedded processors like SAM3U
Reactive Duty Cycle

- Determine duty cycle adaptively, based on energy delta
  - Project past energy consumption at current duty cycle to remaining lifetime
  - Adapt duty cycle based on energy deficit / surplus
  - Requires no explicit understanding/modeling of variability

- Problems with this approach
  - Power measurement
    - Accurately measuring power online is “expensive”
    - Estimating remaining battery charge is difficult
  - With variability, past power consumption is not necessarily a good indicative of future power consumption
Outline

• Duty Cycle in Sensing Systems
• Variability in Contemporary Hardware
• Duty Cycle Scheduling Methods
• Software Architecture
• Results
• Conclusions
Duty Cycling in TinyOS

- Typical duty-cycled application in TinyOS

p: timer period
c: task computation time

duty cycle for the task: \( \frac{c}{c+p} \)

to adapt duty cycle, change c or p

p: timer period
c: task computation time
Adaptable Tasks for TinyOS

- Tasks with adaptable period or adaptable number of iterations

\[
\text{Application} \quad \rightarrow \quad \text{Adaptable Task}
\]

- System activates tasks at maximum allowable rate

\[
p_{\text{min}}, p_{\text{max}} \quad i_{\text{min}}, i_{\text{max}}
\]

System activates tasks at maximum allowable rate

\[p_{\text{max}}, p_{\text{min}}: \text{minimum and maximum period}\]

\[i_{\text{min}}, i_{\text{max}}: \text{minimum and maximum number of iterations per period}\]
Software Architecture

Duty Cycle Scheduler: \[ DC = f(P_A, P_S, ...) \]

- Adaptable Task
- Adaptable Task
- Traditional Task

Hardware Signature

allowable DC

\[ P_A, P_S, ... \]
Outline

• Duty Cycle in Sensing Systems
• Variability in Contemporary Hardware
• Duty Cycle Scheduling Methods
• Software Architecture
• Results
• Conclusions
Comparison of DC Scheduling Methods

Lifetime: 1 year
Processor Instance #7
Battery Capacity: 850 mAh
Temperature Profile: Death Valley, CA, 2009
Comparison of DC Scheduling Methods

Lifetime: 1 year
Processor Instance #7
Battery Capacity: 850 mAh
Temperature Profile: Death Valley, CA, 2009
Comparison of DC Scheduling Methods

Lifetime: 1 year
Processor Instance #7
Battery Capacity: 850 mAh
Temperature Profile: Death Valley, CA, 2009
Comparison of DC Scheduling Methods

Lifetime: 1 year
Processor Instance #7
Battery Capacity: 850 mAh
Temperature Profile: Death Valley, CA, 2009
Improvement over Worst-Case Duty Cycle

average: 7x improvement

Instance

P1  P2  P3  P4  P5  P6  P7  P8  P9  P10
Energy Left Untapped by Worst-Case DC

average: 60% energy left untapped
Lifetime reduction with Datasheet Spec DC

average: 50 days short for one year’s lifetime
Benefits are greater for smaller duty cycles
Benefits are greater with newer technology

significant benefits even with high duty cycles

Worst-Case Duty Cycle: 10%
Temperature range: 60°C
Outline

• Duty Cycle in Sensing Systems
• Variability in Contemporary Hardware
• Duty Cycle Scheduling Methods
• Software Architecture
• Results
• Conclusions
Conclusions

• Growing variability → Need for a software stack that opportunistically adapts to “as measured” hardware characteristics
• Duty Cycled Sensing Systems
  • conservative specifications → untapped energy
  • “optimistic specifications” → unmet lifetime requirements
• Variability-aware duty cycle scheduler
  • Optimizes duty cycle with lifetime constraints
  • 7x improvement over conservative specification
• Duty Cycle Scheduler for TinyOS
  • Adaptable tasks with small overhead
• Ongoing work
  • Alternative methods for exposing variation to software layers
  • Cheap variation monitoring strategies
  • Implications for hardware design
  • See the NSF Variability Expedition website (http://variability.org)
Variability-Aware Duty Cycle Scheduling in Long Running Embedded Sensing Systems

Lucas Wanner, Rahul Balani, Sadaf Zahedi, Charwak Apte, Puneet Gupta, and Mani Srivastava

wanner@ucla.edu

www.variability.org
Comparison of DC Scheduling Methods

![Graph showing duty cycle over time with different schedules: Datasheet Spec, Worst-Case, Reactive.]

- **Lifetime**: 1 year
- **Processor Instance #7**
- **Battery Capacity**: 850 mAh
- **Temperature Profile**: Death Valley, CA, 2009
Comparison of DC Scheduling Methods

Lifetime: 1 year
Processor Instance #7
Battery Capacity: 850 mAh
Temperature Profile: Death Valley, CA, 2009
Analytical Modeling of Sleep Power

• Sources of static power
  • **Sub-threshold Leakage current** that flows between source and drain of a MOSFET for gate-to-source voltages below the threshold
  • Gate Leakage current due to tunneling of carriers through the gate oxide to the substrate
  • Reverse Biased Junction Leakage current which flows from the source/drain regions to the substrate through the reverse biased p-n junctions due to band-to-band tunneling and diffusion
  • Gate Induced Drain Leakage current due to band to band tunneling in the region of overlap between the gate and drain.
  • Sub-threshold Leakage current exhibits strong variation with temperature

\[ P_{sleep} = V_{dd}(AT^2e^{B/T} + I_{gl}) \]