<table>
<thead>
<tr>
<th></th>
<th>Homework index</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lab six</td>
</tr>
</tbody>
</table>
Goals for lecture

• Lab four?

• Lab six

• Simulation of real-time operating systems

• Impact of modern architectural features
Lab four

• Please email or hand in the write-up for lab assignment four

• Problems? See me.
  
  – Will need everything from lab four working for lab six
Lab six

• Develop priority-based cooperative scheduler for TinyOS that keeps track of the percentage of idle time.

• Develop a tree routing algorithm for the sensor network.

• Send noise, light, and temperature data to a PPC, via the network root.

• Have motes respond to send audio samples and buzz commands.

• Play back or display this data on PPCs to verify the system functions.
Outline

• Introduction
• Role of real-time OS in embedded system
• Related work and contributions
• Examples of energy optimization
• Simulation infrastructure
• Results
• Conclusions
Introduction

• Real-Time Operating Systems are often used in embedded systems.

• They simplify use of hardware, ease management of multiple tasks, and adhere to real-time constraints.

• Power is important in many embedded systems with RTOSs.

• RTOSs can consume significant amount of power.

• They are re-used in many embedded systems.

• They impact power consumed by application software.

• RTOS power effects influence system-level design.
Introduction

• Real Time Operating Systems important part of embedded systems
  – Abstraction of HW
  – Resource management
  – Meet real-time constraints

• Used in several low-power embedded systems

• Need for RTOS power analysis
  – Significant power consumption
  – Impacts application software power
  – Re-used across several applications
Role of RTOS in embedded system

- Applications:
  - MPEG encoding
  - ABS
  - Communication etc.

- RTOS services:
  - IPC
  - Memory manager
  - Basic IO
  - Timer
  - Task manager
  - ISR

- Tasks:
  - Micro-browser
  - Organizer
  - Message composer
  - Database

- Hardware:
  - Processor
  - Memory
  - Timer
  - Other hardware
  - Network interface
  - Hardware
Related work and contributions

• **Instruction level power analysis**

• **System-level power simulation**
  Y. Li and J. Henkel, Design Automation Conf., 1998

• **MicroC/OS-II**: J.J. Labrosse, R & D Books, Lawrence, KS, 1998

• **Our work**
  – First step towards detailed power analysis of RTOS
  – Applications: low-power RTOS, energy-efficient software architecture, incorporate RTOS effects in system design
Simulated embedded system

- Easy to add new devices
- Cycle-accurate model
- Fujitsu board support library used in model
- $\mu$C/OS-II RTOS used
Singletask network interface

Get packet → Compute checksum → Procure Ethernet controller → Transfer packet → Release Ethernet controller

Checksum computation and output

Procuring Ethernet controller has high energy cost
TCP example

Straight-forward implementation

Multi-task implementation
Multi-tasking network interface

RTOS power analysis used for process re-organization to reduce energy

21% reduction in energy consumption. Similar power consumption.
ABS example

1. Sense speed and pedal conditions
2. Compute acceleration
3. Brake decision
4. Actuate brake
5. Sleep
6. Timer transition?
ABS example timing

- Timer
- Brake pedal
- ABS process
- Wheel sensor
- Brake action

Time
Straight-forward ABS implementation

Timer transition? →
- Sense speed and pedal conditions
- Compute acceleration
- Brake decision
- Actuate brake

Sleep

Diagram showing:
- Timer
- Brake pedal
- ABS process
- Wheel sensor
- Brake action

Time
Periodically triggered ABS

Timer transition?

Sense speed and pedal conditions

Compute acceleration

Brake decision

Actuate brake

Sleep

Y

N
Periodically triggered ABS timing

- Timer
- Brake pedal
- ABS process
- Wheel sensor
- Brake action

Time
Selectively triggered ABS

Flowchart:
- Pedal pressed?
  - Y: Sense speed and pedal conditions
    - N: Sleep
    - Y: Timer transition?
      - N: Sleep
      - Y: Actuate brake
  - N: Sleep
    - N: Timer transition?
      - Y: Sleep
      - N: Timer transition?
Selectively triggered ABS timing

63% reduction in energy and power consumption
Power-optimized ABS example

Pedal pressed?  
Yes → Sense speed and pedal conditions  
No → Sleep

Sleep  
Sleep → Timer transition?  
Yes → Actuate brake  
No → Brake decision

Brake decision  
Compute acceleration  
Brake pedal

ABS process  
Wheel sensor

Timer  
Brake action

Time
Infrastructure

Application code → SPARC lite compiler

OS code → SPARC lite compiler

External stimulus → SPARC lite compiler

Timer model

UART model

Models for other peripherals

SPARC lite cache simulator

SPARC lite ISS

Instruction-level energy model

Memory model

Memory energy model

Cache controller model

Bus interface unit model

Energy by call tree position for task A

Energy by call tree position for task B
Experimental results – time
Agent example

Key
- - - - - Broadcast
- - Price advertisement
- Sale

Agent 1

Agent 2

Agent 3

Agent 4

Agent 5

Agent 6

Money
Commodity 1
Commodity 2
Commodity 3
Commodity 4
Experimental results

(a) SleepSynchronization
(b) Task control

Energy (mJ)

Time (ms)

0 500 1000 1500 2000 2500 3000 3500 4000 4500 5000 5500 6000 6500 7000 7500 8000 8500 9000 9500 10000 10500

Application
Floating-point Initialization
Input/output Interrupt
Mailbox
Memory
Misc.
Scheduling
Semaphore
Sleep
Synchronization Task control
Experimental results

(a) Sleep synchronization task control
(b) Semaphore application floating-point initialization input/output interrupt memory

Energy (mJ)

- Application
- Floating-point
- Initialization
- Input/output
- Interrupt
- Mailbox
- Memory
- Misc.
- Scheduling
- Semaphore
- Sleep
- Synchronization
- Task control
Optimization effects

TCP example:
- 20.5% energy reduction
- 0.2% power reduction
- RTOS directly accounted for 1% of system energy

ABS example:
- 63% energy reduction
- 63% power reduction
- RTOS directly accounted for 50% of system energy

Mailbox example: RTOS directly accounted for 99% of system energy

Semaphore example: RTOS directly accounted for 98.7% of system energy
## Partial semaphore hierarchical results

<table>
<thead>
<tr>
<th>Function</th>
<th>Energy/invocation (uJ)</th>
<th>Energy (%)</th>
<th>Time (mS)</th>
<th>Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>init_lvecs</td>
<td>0.41</td>
<td>0.00</td>
<td>0.00</td>
<td>1</td>
</tr>
<tr>
<td>init_timer</td>
<td>1.31</td>
<td>0.00</td>
<td>0.00</td>
<td>1</td>
</tr>
<tr>
<td>do_main</td>
<td>887.44</td>
<td>0.28</td>
<td>2.18</td>
<td>1</td>
</tr>
<tr>
<td>save_data</td>
<td>1.56</td>
<td>0.00</td>
<td>0.00</td>
<td>1</td>
</tr>
<tr>
<td>init_data</td>
<td>1.31</td>
<td>0.00</td>
<td>0.00</td>
<td>1</td>
</tr>
<tr>
<td>init_bss</td>
<td>0.88</td>
<td>0.00</td>
<td>0.00</td>
<td>1</td>
</tr>
<tr>
<td>cache_on</td>
<td>2.72</td>
<td>0.00</td>
<td>0.00</td>
<td>1</td>
</tr>
<tr>
<td>win_unf_trap</td>
<td>1.90</td>
<td>1.20</td>
<td>9.73</td>
<td>1999</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSEnableInt</td>
<td>0.32</td>
<td>0.10</td>
<td>0.89</td>
<td>1000</td>
</tr>
<tr>
<td>OSSemPend</td>
<td>2.48</td>
<td>0.78</td>
<td>6.33</td>
<td>999</td>
</tr>
<tr>
<td>OSSemPost</td>
<td>0.29</td>
<td>0.18</td>
<td>1.59</td>
<td>1999</td>
</tr>
<tr>
<td>OSSched</td>
<td>19.07</td>
<td>6.00</td>
<td>47.97</td>
<td>999</td>
</tr>
<tr>
<td>OSSemPost</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>OSTimeGet</td>
<td>0.27</td>
<td>0.08</td>
<td>0.70</td>
<td>1000</td>
</tr>
<tr>
<td>OSPend</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.27</td>
<td>0.08</td>
<td>0.70</td>
<td>1000</td>
</tr>
<tr>
<td>OSTimeGet</td>
<td>1.09</td>
<td>0.00</td>
<td>0.00</td>
<td>1</td>
</tr>
<tr>
<td>OSPend</td>
<td>0.27</td>
<td>0.08</td>
<td>0.70</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>4.77</td>
<td>0.02</td>
<td>0.17</td>
<td>15</td>
</tr>
<tr>
<td>_OSEnableInt</td>
<td>108.89</td>
<td>34.30</td>
<td>258.53</td>
<td>1000</td>
</tr>
<tr>
<td>printf</td>
<td>2.05</td>
<td>0.65</td>
<td>5.06</td>
<td>1000</td>
</tr>
<tr>
<td>printf</td>
<td>1.09</td>
<td>0.00</td>
<td>0.00</td>
<td>1</td>
</tr>
<tr>
<td>exceptionHandler</td>
<td>4.77</td>
<td>0.02</td>
<td>0.17</td>
<td>15</td>
</tr>
<tr>
<td>printf</td>
<td>108.89</td>
<td>34.30</td>
<td>258.53</td>
<td>1000</td>
</tr>
<tr>
<td>printf</td>
<td>2.05</td>
<td>0.65</td>
<td>5.06</td>
<td>1000</td>
</tr>
<tr>
<td>printf</td>
<td>1.09</td>
<td>0.00</td>
<td>0.00</td>
<td>1</td>
</tr>
<tr>
<td>exceptionHandler</td>
<td>4.77</td>
<td>0.02</td>
<td>0.17</td>
<td>15</td>
</tr>
</tbody>
</table>

**Task1**

- **155.18 mJ total**
- **48.88 %**

<table>
<thead>
<tr>
<th>Function</th>
<th>Energy/invocation (uJ)</th>
<th>Energy (%)</th>
<th>Time (mS)</th>
<th>Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>realstart</td>
<td>6.41</td>
<td>2.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.51 mJ total</td>
<td></td>
<td>1.74</td>
<td></td>
<td></td>
</tr>
<tr>
<td>startup</td>
<td>0.90</td>
<td>0.28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.82 %</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>win_unf_trap</td>
<td>1.90</td>
<td>1.20</td>
<td>9.73</td>
<td>1999</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSEnableInt</td>
<td>0.32</td>
<td>0.10</td>
<td>0.89</td>
<td>1000</td>
</tr>
<tr>
<td>OSSemPend</td>
<td>2.48</td>
<td>0.78</td>
<td>6.33</td>
<td>999</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.18</td>
<td>1.59</td>
<td>1999</td>
</tr>
<tr>
<td>_OSEnableInt</td>
<td>0.29</td>
<td>0.18</td>
<td>1.59</td>
<td>1999</td>
</tr>
<tr>
<td>OSSched</td>
<td>19.07</td>
<td>6.00</td>
<td>47.97</td>
<td>999</td>
</tr>
<tr>
<td>OSSemPost</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>OSSched</td>
<td>19.07</td>
<td>6.00</td>
<td>47.97</td>
<td>999</td>
</tr>
<tr>
<td>OSSemPost</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>OSSched</td>
<td>19.07</td>
<td>6.00</td>
<td>47.97</td>
<td>999</td>
</tr>
<tr>
<td>OSSemPost</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>OSSched</td>
<td>19.07</td>
<td>6.00</td>
<td>47.97</td>
<td>999</td>
</tr>
<tr>
<td>OSSemPost</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>OSSched</td>
<td>19.07</td>
<td>6.00</td>
<td>47.97</td>
<td>999</td>
</tr>
<tr>
<td>OSSemPost</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>OSSched</td>
<td>19.07</td>
<td>6.00</td>
<td>47.97</td>
<td>999</td>
</tr>
<tr>
<td>OSSemPost</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>OSSched</td>
<td>19.07</td>
<td>6.00</td>
<td>47.97</td>
<td>999</td>
</tr>
<tr>
<td>OSSemPost</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>OSSched</td>
<td>19.07</td>
<td>6.00</td>
<td>47.97</td>
<td>999</td>
</tr>
<tr>
<td>OSSemPost</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>OSSched</td>
<td>19.07</td>
<td>6.00</td>
<td>47.97</td>
<td>999</td>
</tr>
<tr>
<td>OSSemPost</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>OSSched</td>
<td>19.07</td>
<td>6.00</td>
<td>47.97</td>
<td>999</td>
</tr>
<tr>
<td>OSSemPost</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>_OSDisableInt</td>
<td>0.29</td>
<td>0.09</td>
<td>0.78</td>
<td>1000</td>
</tr>
<tr>
<td>OSSched</td>
<td>19.07</td>
<td>6.00</td>
<td>47.97</td>
<td>999</td>
</tr>
</tbody>
</table>
## Energy per invocation for μC/OS-II services

<table>
<thead>
<tr>
<th>Service</th>
<th>Minimum energy (μJ)</th>
<th>Maximum energy (μJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSEventTaskRdy</td>
<td>18.02</td>
<td>20.03</td>
</tr>
<tr>
<td>OSEventTaskWait</td>
<td>7.98</td>
<td>9.05</td>
</tr>
<tr>
<td>OSEventWaitListInit</td>
<td>20.43</td>
<td>21.16</td>
</tr>
<tr>
<td>OSInit</td>
<td>1727.70</td>
<td>1823.26</td>
</tr>
<tr>
<td>OSMboxCreate</td>
<td>27.51</td>
<td>28.82</td>
</tr>
<tr>
<td>OSMboxPend</td>
<td>7.07</td>
<td>82.91</td>
</tr>
<tr>
<td>OSMboxPost</td>
<td>5.82</td>
<td>84.55</td>
</tr>
<tr>
<td>OSMemCreate</td>
<td>19.40</td>
<td>19.75</td>
</tr>
<tr>
<td>OSMemGet</td>
<td>6.64</td>
<td>8.22</td>
</tr>
<tr>
<td>OSMemInit</td>
<td>27.41</td>
<td>27.47</td>
</tr>
<tr>
<td>OSMemPut</td>
<td>6.38</td>
<td>7.91</td>
</tr>
<tr>
<td>OSQInit</td>
<td>20.10</td>
<td>20.93</td>
</tr>
<tr>
<td>OSSched</td>
<td>6.96</td>
<td>52.34</td>
</tr>
<tr>
<td>OSSemCreate</td>
<td>27.87</td>
<td>29.04</td>
</tr>
<tr>
<td>OSSemPend</td>
<td>6.54</td>
<td>73.64</td>
</tr>
<tr>
<td>etc.</td>
<td>etc.</td>
<td>etc.</td>
</tr>
</tbody>
</table>
Conclusions

• RTOS can significantly impact power
• RTOS power analysis can improve application software design
• Applications
  – Low-power RTOS design
  – Energy-efficient software architecture
  – Consider RTOS effects during system design
Impact of modern architectural features

• Memory hierarchy
• Bus protocols ISA vs. PCI
• Pipelining
• Superscalar execution
• SIMD
• VLIW
Summary

- Labs
- Simulation of real-time operating systems
- Impact of modern architectural features