Understanding Power Management Design Issues for Handheld Wireless Devices

Andrew Girson
InHand Electronics, Inc.
agirson@inhandelectronics.com

10-19-04

Presentation Overview

- "System" Power Management Defined
  - How Does It Differ from Component Level Power Management?
- Hardware Design for Power Management
  - Design-Time and Run-Time Features
- Software-Based Run-Time Power Management
  - Performance Adjustment, Energy Sharing, and Event Reduction
- Summary

General Definition of Power Management

- Definition:
  - The intelligent supervision and manipulation of energy sources and energy loads in wireless devices, so as to extend usable life (without affecting perceived performance)
- Why?
  - Because improvements in batteries (energy sources) are not keeping up with the requirements of devices (energy loads)

Power Management – The Good News

- Power management is a growing business
- Major semiconductor and system software vendors are making a significant push to reduce the energy consumption of their products and are making their products more energy-aware
- Major hardware/software development tools suppliers are incorporating features to allow designers to better manage and limit energy consumption
- Results at the component level:
  - Handling of leakage current in silicon
  - Use of power islands to make energy control more granular
  - Provision for power modes in CPU hardware and OS software

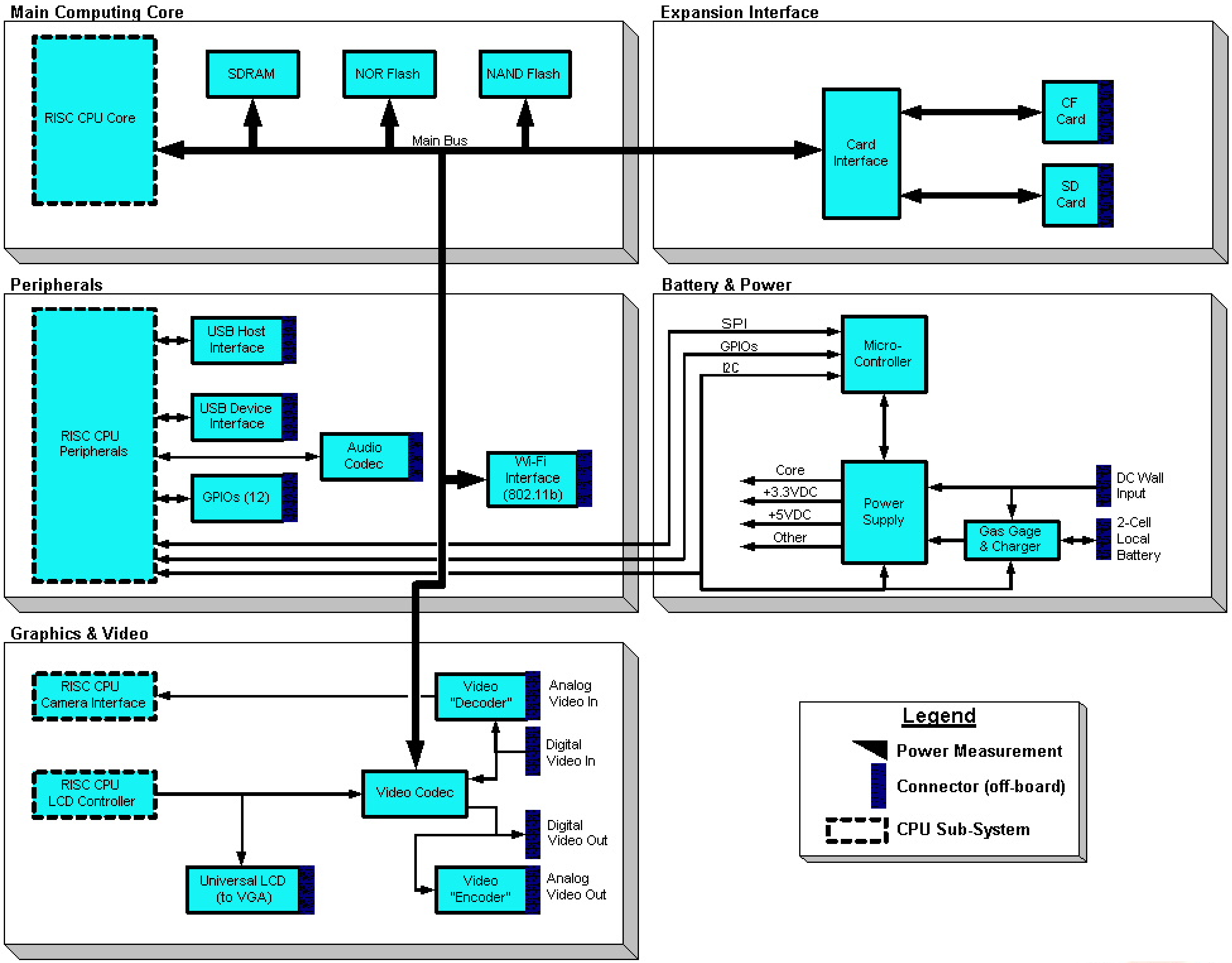
The “System” Power Management Challenge

- These innovations at the semiconductor and system software level must be orchestrated into device designs
  - How a component’s power management features are used is highly dependent on the system design of the device itself
  - Many items impact how components are combined into a complete device design
    - Feature requirements
    - Usage scenarios

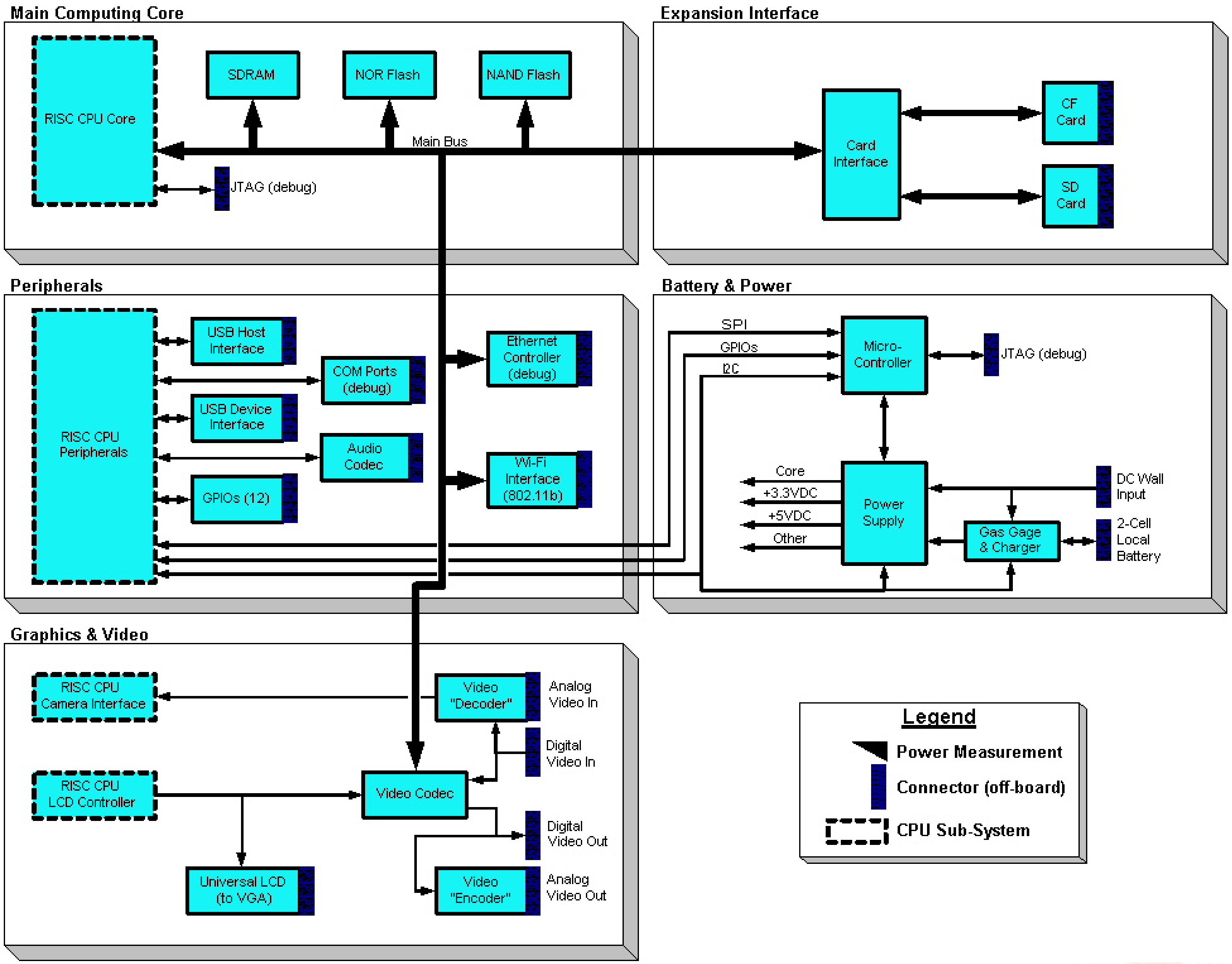
Hardware Design for Power Management

- High-efficiency power supplies
- Support for multiple low-power modes
- Partitioned power control for peripherals
- Battery management
- Voltage adjustment
- Peripheral performance enhancers
- Peripheral power measurement
- Peripheral performance measurement

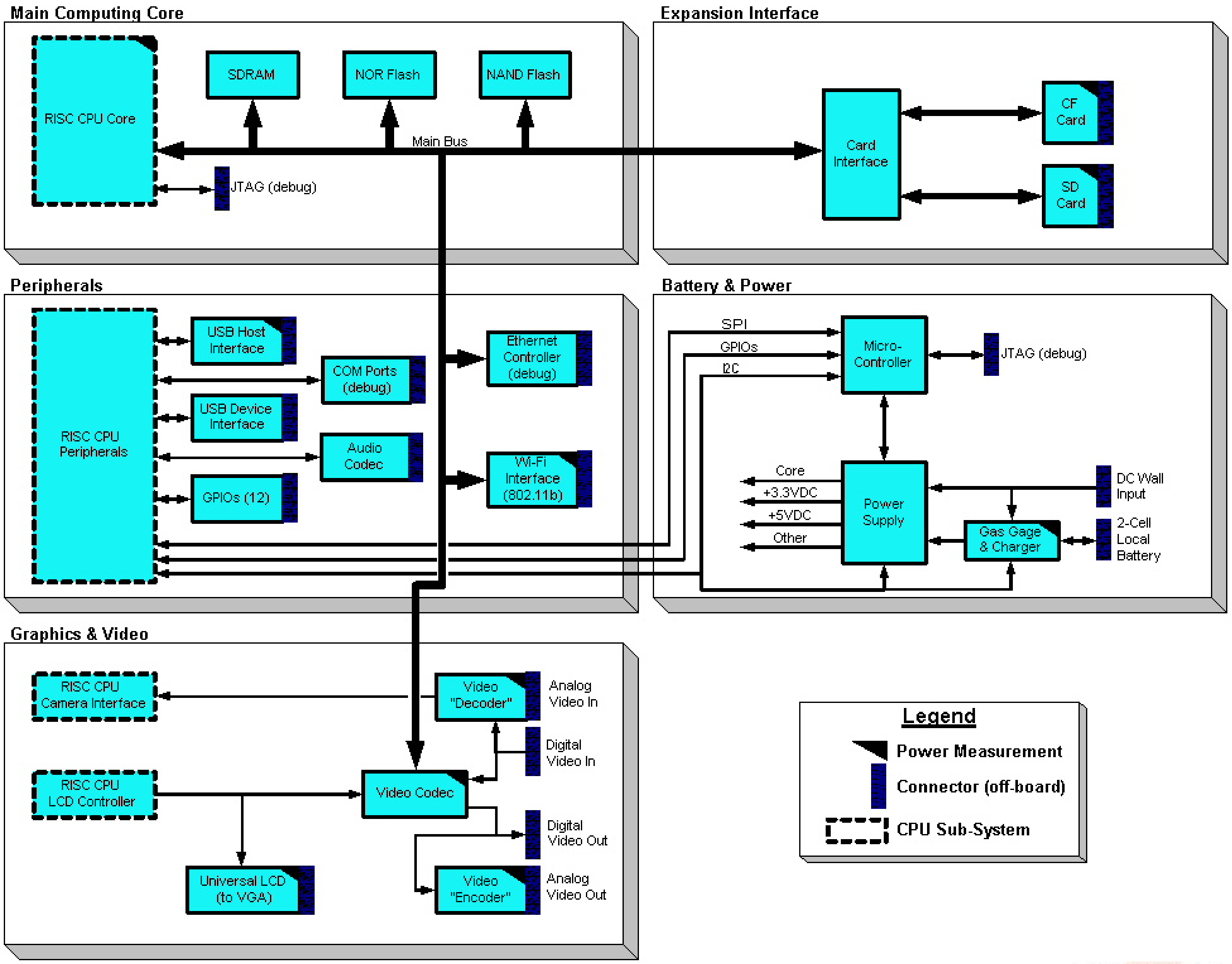
Hardware Design – System Block Diagram



Hardware Design – Debug



Hardware Design – Power Measurement



Hardware Design Summary

- High-efficiency power supplies
- Support for multiple low-power modes
- Partitioned power control for peripherals
- Battery management
- Voltage adjustment
- Peripheral performance enhancers
- Peripheral power measurement
- Peripheral performance measurement

- *To design a truly low-power device, you must expand on "functional" debug hardware to include "power" debug hardware*

Software-Based Run-Time Power Management

- Performance Adjustment
- Energy Sharing
- Event Reduction

Performance Adjustment

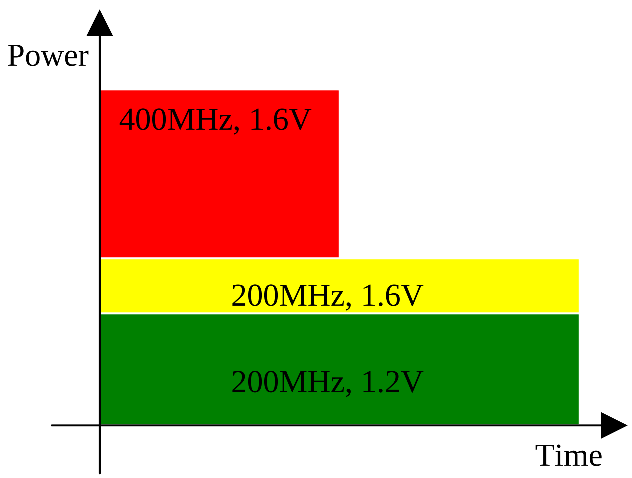
- A device's software tasks have varying performance requirements
- Most of today's CPUs for wireless devices can operate at different clock frequencies
- Proper adjustment of clock frequency can greatly impact battery life
- This is an area of significant business interest right now

Clock & Voltage Scaling

- The internal performance level of a CPU core is directly proportional to its clock frequency
- The power consumed by a CPU core is directly proportional to its clock frequency
- The power consumed by a CPU core is directly proportional to the square of its supply voltage

  - $P = K * F * V^2$

Clock & Voltage Scaling

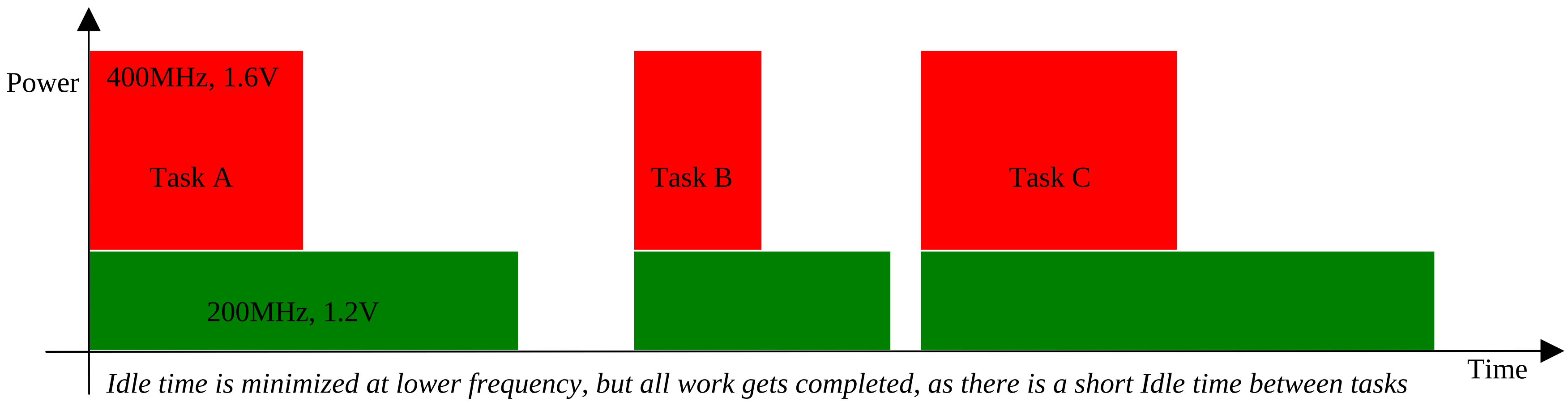


Energy is the area under the curve

- Reducing a CPU's clock frequency by 50% without reducing voltage cuts power consumption for the core by about 50%
- However, because internal performance is reduced by 50%, a task can take twice as long
- The result is that energy consumption is approximately the same
- However, if voltage is dropped too, overall energy consumption is reduced

Clock & Voltage Scaling

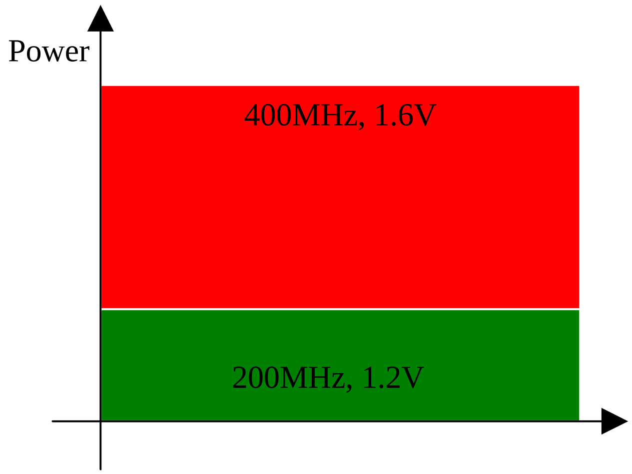
- Reducing a CPU's clock frequency by 50% without reducing voltage cuts power consumption for the core by about 50%
- However, because internal performance is reduced by 50%, a task can take twice as long
- The result is that energy consumption is about the same
- However, if voltage is dropped too, overall energy consumption is reduced



Idle time is minimized at lower frequency, but all work gets completed, as there is a short Idle time between tasks

We call this a "just-in-time" clock/voltage adjustment algorithm

Clock & Voltage Scaling

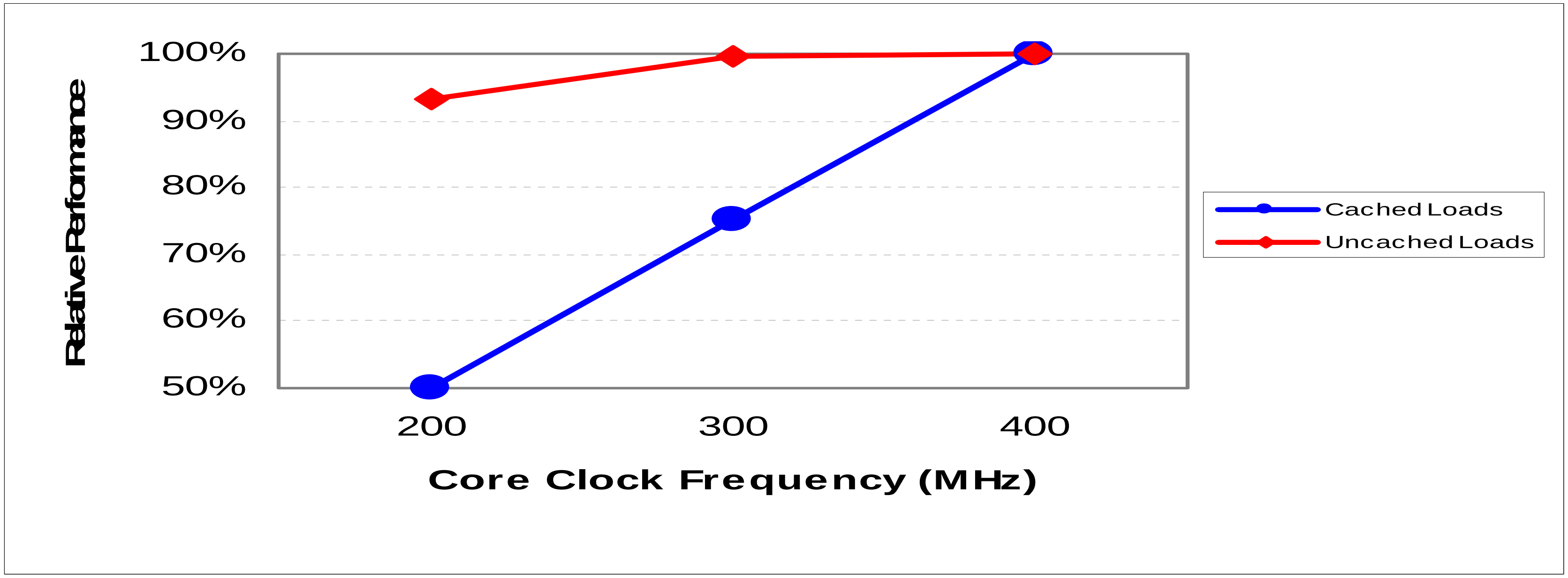


Task completes in same amount of time, regardless of clock frequency

- "Just-in-time" algorithms do not analyze Run mode; rather they just minimize Idle
- "Saturation" algorithms analze the system and adjust clock frequency and voltage, even when the CPU is not Idling
  - Saturation algorithms search for periods when performance is independent of clock frequency
    - Example scenarios include peripheral activities such as wireless or disk I/O
    - Energy consumed during these scenarios can drop by 50% or more, depending on the combination of peripheral activity
    - The challenge is determining when performance is independent of clock frequency
- "Saturation" algorithms can also incorporate other system factors such as battery performance issues

New Slide

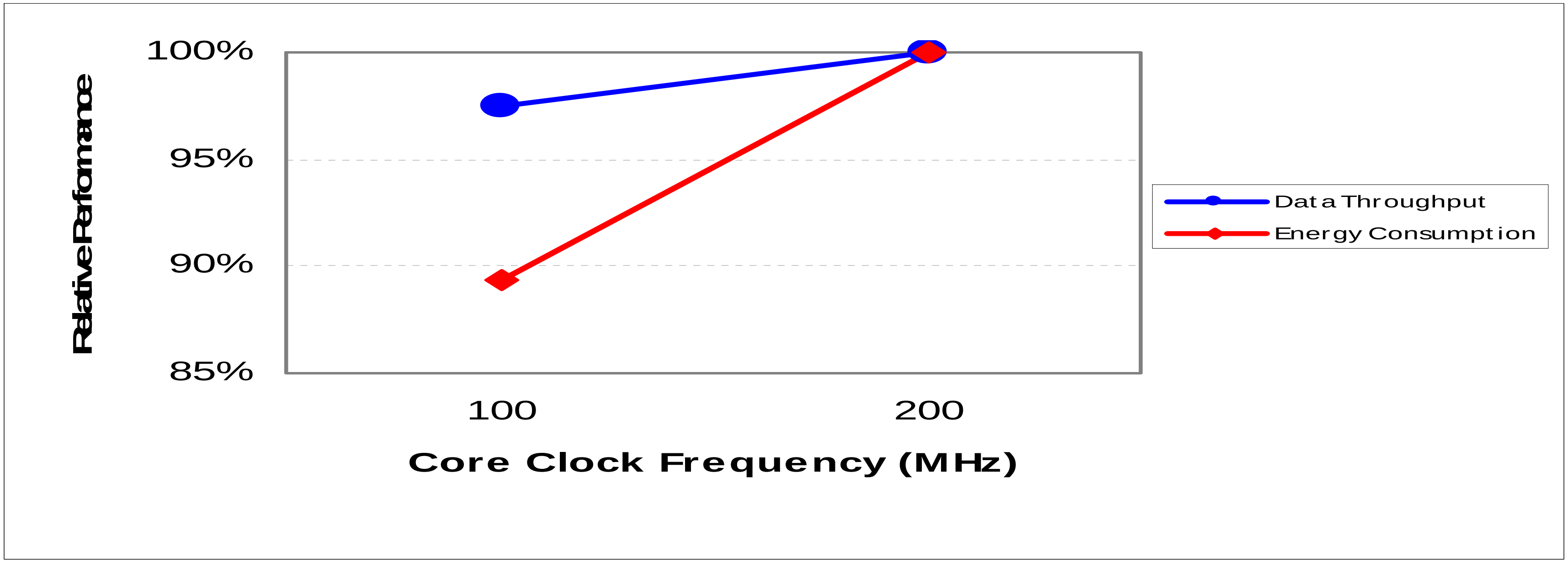
Clock & Voltage Scaling



- The graph shows low-level microprocessor "relative" throughput for loading of data from cached and uncached locations
- Throughput of cached loads is directly proportional to clock frequency
- Throughput of uncached loads is not proportional to clock frequency, and if detected unobtrusively, can be used to reduce energy consumption with minimal performance effects

New Slide

Clock & Voltage Scaling

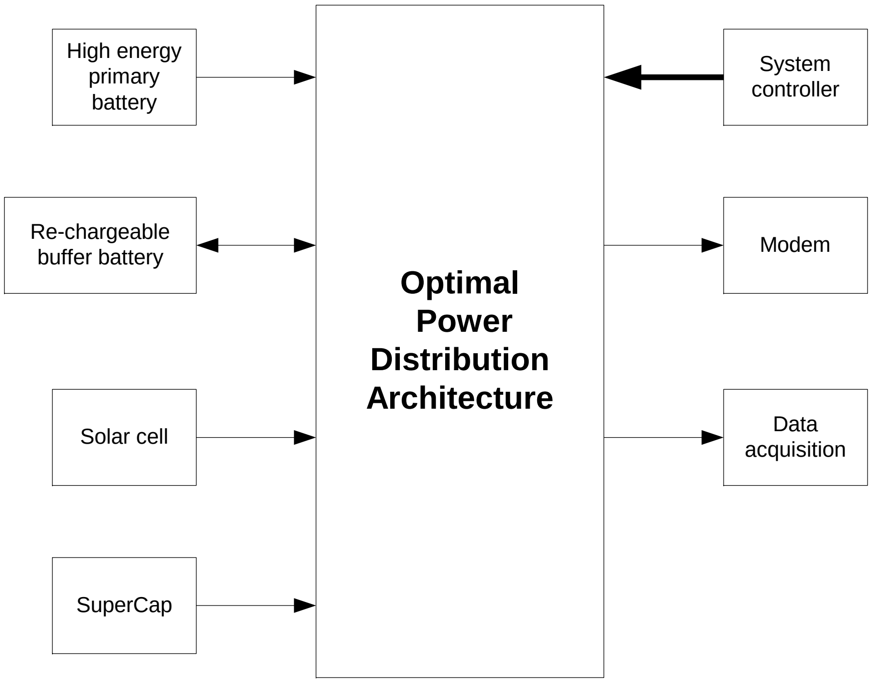


- The graph shows high-level throughput and energy consumption for file access to a Compact Flash storage device
- A halving of clock frequency drops throughput only 2.5%
- A halving of clock frequency drops energy consumption 11%
  - Total "system" energy consumption
  - No voltage adjustment

Energy Sharing

- Devices have multiple internal loads with differing characteristics
- Different batteries have operational features that define how well they handle different types of loads
- Correct matching of batteries to loads can increase a device's operational life and lengthen a rechargeable battery's life

Energy Sharing – Load Distribution

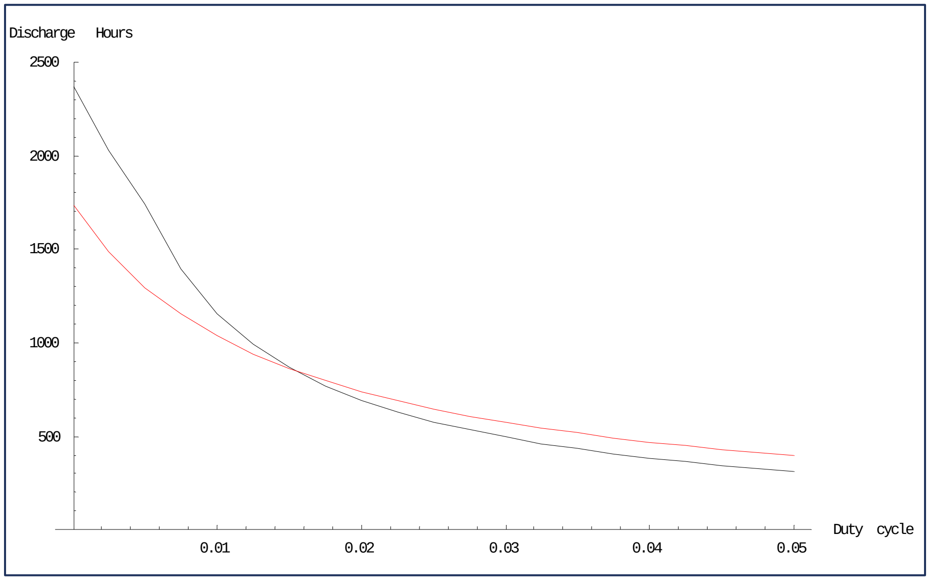


- Load distribution networks direct the flow of energy from multiple sources to multiple loads
- The goal is to "match" source and load characteristics

Energy Sharing – Hybrid Sources

- To achieve the most efficient release of energy, an energy source must operate as close to it's optimal regime as possible
  - Batteries are non-linear and provide optimum energy release only in a very narrow range of operation (supply current, temperature, etc.)
  - Run/Idle modes create highly variable peak-to-average power and energy consumption
  - A single energy source is not efficient for all modes
- A hybrid energy source incorporates multiple sources and hardware/software to distribute energy from the sources in real-time as a function of load changes
  - Hybrid gasoline/electric automobiles have a similar philosophy
  - Devices with widely variable power modes – such as cell phones and smart sensors – can benefit greatly from properly managed hybrid sources

Energy Sharing – Summary



- The graph shows a standard battery source (red) and a hybrid source (black)
- At a 1% run/idle duty cycle, the hybrid source increases battery life by 20% (without altering volume or weight)

Event Reduction – Defined

- Reducing the number of events (interrupts) to allow the CPU to spend more time in lower-power modes, without necessarily affecting performance

Event Reduction Concepts

- Direct memory access (DMA) is a well-known feature that takes on new meaning in wireless devices, because it allows the number of events to be reduced, thereby increasing Idle time and consequently battery life
  - Software developers should incorporate hardware-assisted DMA whenever possible, avoiding polling and interrupt-driven I/O
  - The result is a win-win: reduced CPU core performance requirements and lower power consumption
- Many embedded operating systems now implement intelligent time-slicing, eliminating the standard 1ms thread-switching interrupt and replacing it with variable time-slices
  - Software developers need to understand their software's process/thread structure and adjust accordingly

Summary

- Battery technology is improving at a linear rate and new energy source technology (e.g., fuel cells, radio isotopes) is not yet mass-producible, while energy requirements for devices are increasing at an exponential rate
- Many silicon, software, and tools providers are now making their products energy-aware
- Innovations in system-wide power management technologies are now taking advantage of the energy-aware features of new components, thereby reducing the gap between energy sources and energy loads