

# Power-aware Embedded Computing

Margarida F. Jacome and Anand Ramachandran in University of Texas  
Information Technology Handbook, Editor Richard Zurawski, CRC Press, FL, 1-20, 2004

2<sup>nd</sup> Semester in Master's  
Wonhong Kwon

- 1. Embedded system에서 Power-awareness를 고려 할 시 가장 중요하게 생각 해 보아야 할 2가지 요소를 서술 하세요.
  - + (1) Power consumption과 Performance는 trade-off 관계 이므로 어느 한 쪽에 치우치지 않고 Target system의 Requirement에 맞게 Design 해야 한다.
  - + (2) Embedded system의 특성 상 general-purpose가 아닌 specific application을 위해 design 되기 때문에 Application에 적합한 techniques를 사용 하여야 한다.
- 2. Power dissipation에는 Dynamic/Static 두 가지 요소가 있는데 이 둘의 차이점을 서술 하시오.
  - + (1) Dynamic 은 Circuit이 그 design된 목적에 맞게 수행 될 때 소비되는 전력을 의미 한다.
    - Active 상태 에서의 소비 전력
  - + (2) Static은 Circuit switching activity사이에 state를 유지 하기 위해 사용 되는 전력으로 leakage power라고도 불리 운다.
    - Non-active상태 에서의 소비 전력

- Introduction
- Energy and Power Modeling
- System/Application Level Optimizations
- Energy Efficient Processing Subsystems
- Summary

- Recently, power consumption has emerged as a major concern in embedded systems.
  - + It has a direct impact on packaging, cooling costs, reliability, lifetime.
- Digital CMOS circuits have two main types of power dissipation.
  - + Dynamic
    - When the circuit performs the function which it was designed for.

$$P_{dynamic} = \sum_{k=1}^M C_k \cdot f_k \cdot V_{DD}^2$$

- + Static
  - Preserve the logic state between such switching activity.

- The power problem must be addressed at all levels of the design hierarchy from system to circuit.
- In this paper, providing on system and architecture level design techniques to reduce both static and dynamic power dissipation.
  - + At this high-level abstraction, the specifics of each particular embedded applications can be considered as a whole.

- Modeling and power estimation assist system and architecture-level design.
  - + Early design phase, high degree of accuracy on power estimation is not available.
  - + Highly accurate estimation would be too time consuming to need a reasonable degree of design.
  
- Power estimation during early design stage should aim to ensure a high degree of **fidelity** rather than accuracy.
  - + Evaluate the relative power efficiency of different candidate system architectures, alternative SW implementations and different power management techniques
  
- There are four models in this paper.
  - + Instruction- and Function-Level Models
  - + Micro-Architecture Models
  - + Memory and Bus Models
  - + Battery Models

## ■ Instruction-level power model

- + Estimate the relative power efficiency of different processors
  - Same application, possibly alternative memory configurations
- + Also evaluate the relative power efficiency of SW implementations
  - Same application, in the context of a specific embedded platform
- + It is **assigns a power cost to each assembly instruction** and estimates the overall energy consumed by summing up the instruction costs for a dynamic execution trace
- + But, this model can still be prohibitively time consuming during early design space
  - Collecting and analyzing large instruction traces for many processors

## ■ Function-level power model

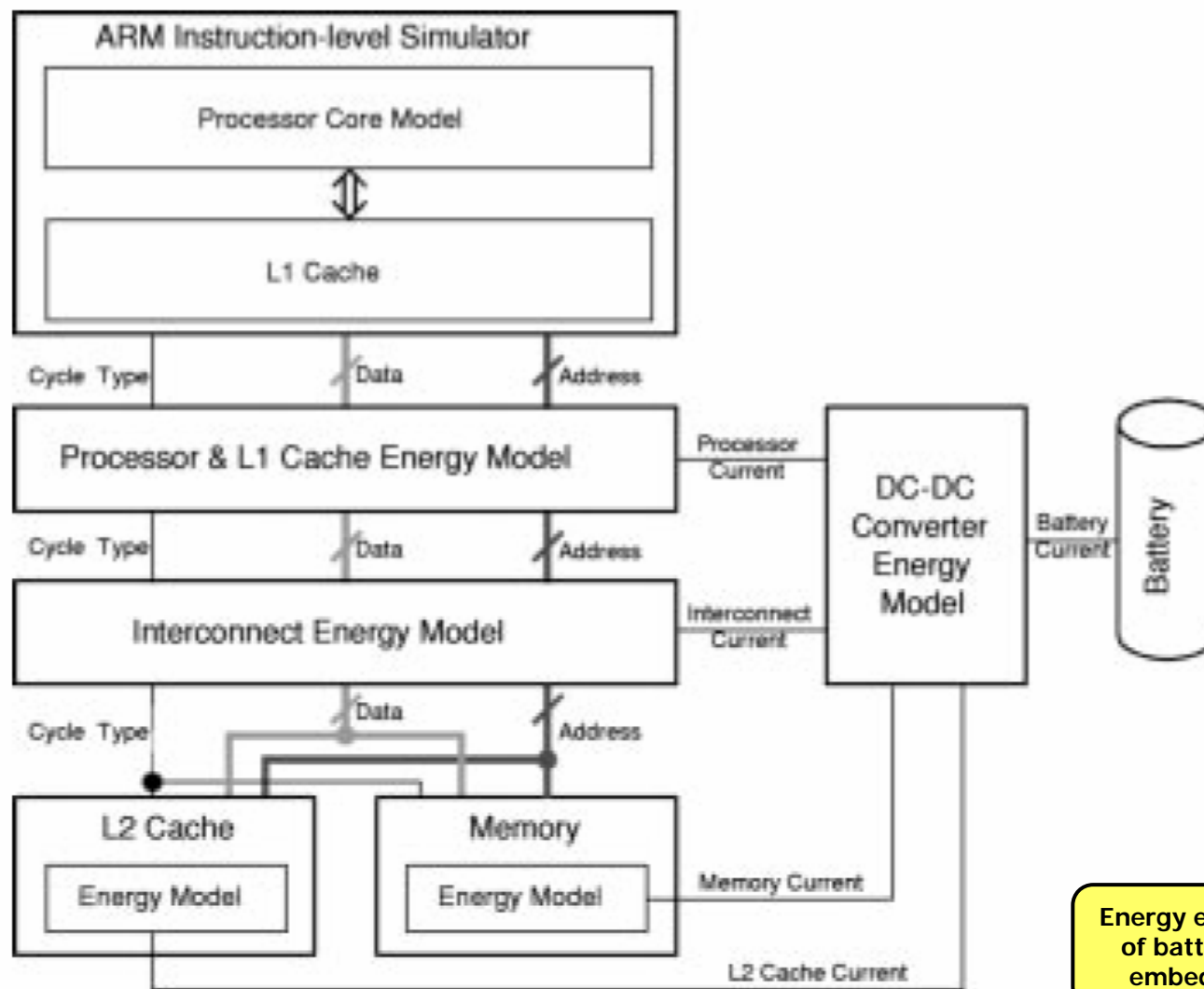
- + In order to **accelerate estimation**
- + It relies on the use of macromodels characterizing the average energy consumption of a library of subroutines executing on a target processor.
- + E.g.) Insertion sort  $\rightarrow an^2 + bn + c$ 
  - Actual power dissipation then needs to be measured for a large number of experiments, run with different values of  $n$ . (  $n$  denotes the number of elements )

- These models allow designers to quickly evaluate a large number candidate system architectures and alternative software implementation.
- This initial broad exploration is concluded, power models for each of the architecture's main subsystems and components are needed.
  - + In order to support the detailed architectural design phase that follows.



- This models are critical to evaluating the impact of different processing sub-system choices on power consumption.
  - + As well as the effectiveness of different power management techniques
  - + E.g.) Simplepower based on Simplescalar
- **Simulation-based power estimation** techniques use very simple empirical power models for data path and control logic, and slightly more sophisticated models for regular structures such as caches.
  - + E.g.) Equivalent capacitance
- A substantial percentage of the overall power budget of a processor is actually spent on the **global clock**.
  - + Power dissipation on global clock distribution is impacted by # of pipeline registers and by global and local wiring capacitances.
  - + Critical issue during processor core selection and configuration

# ARMulator power-extended version



Energy efficient design  
of battery-powered  
embedded system

- Storage elements constitute a substantial part of the power budget of embedded systems.
- CACTI
  - + Cache Access and Cycle Time
  - + Given a specific cache configuration and target requirements, It generates a structural design for such cache configuration.
- Buses are also a significant contributor to dynamic power dissipation.
  - +  $C \times V_{DD}^2 \times f_a$
  - + The average switching frequency of the bus (  $f_a$  ) is defined by the product of two terms.
    - The average number of bus transitions per word x bus frequency

- The capacity of a battery is a nonlinear function of the current drawn from it.
  - + E.g.) Increases current drawn from a battery by a factor of two, deliverable battery capacity and its lifetime is decreased by more than half.
- This is a trade-off between quality/performance and duration of service in a nonlinearity manner. ( rate-capacity effect )
- In order to properly evaluate the effectiveness of techniques during system level design, adequate battery models and metrics are needed.

## ■ Battery-discharge delay product

- + A new metric to emphasize the importance of accurately exploring key trade-offs between battery lifetime and system performance.
- + Rate-capacity effect is sensitive factor in this metric.
- + Detailed/precise battery model is required.
  - Predict the remaining capacity of a rechargeable lithium-ion battery in terms of several factors.
  - Discharge-rate, battery output voltage, battery temperature, cycle age

- To provide the system with the desired functionality while meeting cost, battery lifetime, and other requirements, it may be useful to explore trade-offs between power/energy and QoS.
  - + Optimizations is needed under the control of a system level power manager.
  - + E.g.) If the battery level drops below a certain threshold, the power manager may drop some services and/or swap some tasks to less power hungry version.
  - + Sometimes, power manager may also shutdown or slowdown subsystems.

# Voltage and Frequency Scaling

- Digital CMOS circuits have both dynamic and static power consumption.
- + Every transition of a digital circuit consumes power, because every charge or discharge of the digital circuit's capacitance drains power.

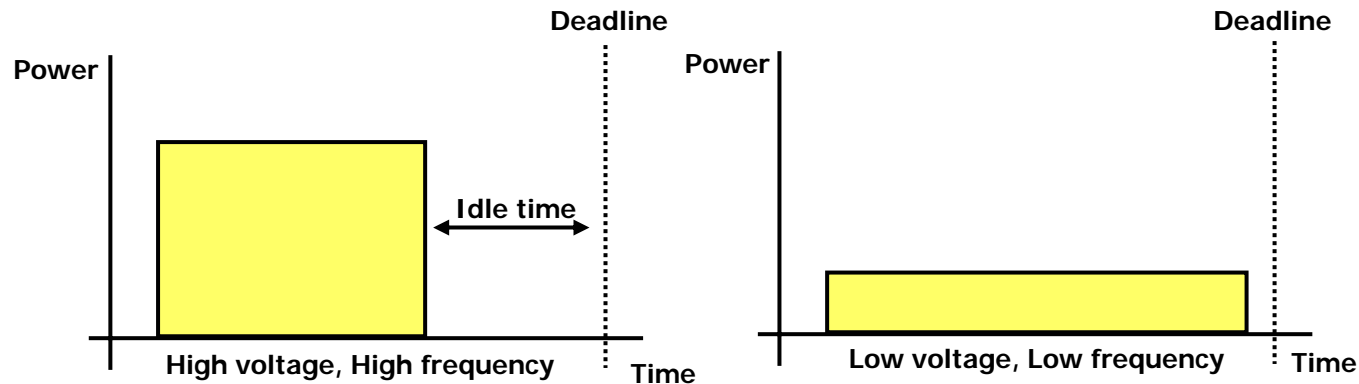
$$P_{dynamic} = \sum_{k=1}^M C_k \cdot f_k \cdot V_{DD}^2$$

$M$  : # of gates  
 $C_k$  : load capacitance of gate  $g_k$   
 $f_k$  : switching frequency of  $g_k$   
 $V_{DD}$  : supply voltage

- + Reduction of VDD is the most effective solution. However, Lowering VDD creates the problem of increased circuit delay.

$$\tau \propto \frac{V_{DD}}{(V_G - V_T)^2}$$

$\tau$  : propagation delay  
 $V_G$  : input gate voltage  
 $V_T$  : threshold voltage



- This technique must necessarily rely on adequate dynamic workload prediction and performance metrics.
  - + More complex
  - + But, needed RT systems



- Dynamic resource scaling refers to exploiting adaptive, fine-grained HW resource reconfiguration in order to improve power efficiency.
- Clock gating or  $V_{DD}$  gating
  - + Enhance the micro-architecture with the ability to selectively disable components, fully or partially.
- Thus, by dynamically scaling down micro-architecture components during low activity periods, substantial power savings can potentially be achieved.

- “Processor-memory performance gap” causes the memory access latency problem.
- As a consequence, power dissipation in the memory contributes to a substantial fraction of the total energy consumption.
  - + E.g.) StrongARM SA-110 revealed that more than 40% of the processor’s power budget. ( on-chip cache )

**Power-aware memory designs have received considerable attention in recent years.**

- The energy cost of accessing data/instructions from off-chip memories can be as much as two orders of magnitude higher than that of an access to on-chip memory.
- In the context of embedded systems, it is possible to match the bandwidth requirements and access patterns of the target embedded application by carefully tuning the configuration of the cache hierarchy.

- Energy efficiency is improved in these designs by taking direct advantage of specific characteristics of target classes of applications.
- Vertical partition schemes
  - + E.g.) Filter cache
    - Very small cache placed in front of the L1 data cache.
    - For applications with small working sets, this strategy can lead to considerable power savings.
  - + E.g.) Pre-decoded instruction buffers and loop buffers
    - Applied to instruction caches
    - Store recently used instructions on an instruction buffer, in a decoded form to reduce the fetch and decode. ( pre-decoded instruction buffers )
    - Hold time-critical loop bodies
- Horizontal partition schemes
  - + Place the additional buffers at the same level as the L1 cache
  - + E.g.) Region-based caches
    - Add two small 2KB L1 D-caches, one for stack and one for global data

- Dynamic Scaling of Memory Elements
  - + Reduce a leakage power
  - + E.g.) Cache Decay
    - Turned off cache lines after a fixed number of cycles ( decay interval )
- Special purpose memory subsystems for specific application
- Code Compression
- Interconnect Optimizations