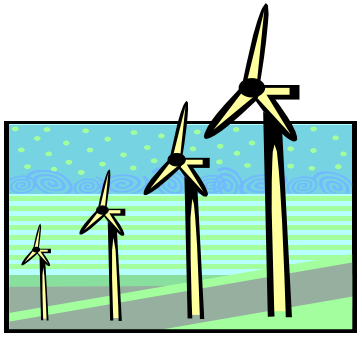
Embedded Systems

Processing units

- Need for efficiency (power + energy):

Why worry about energy and power?

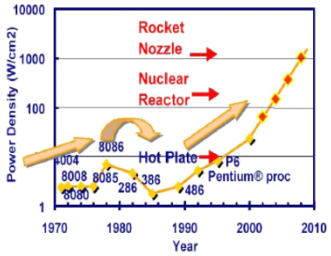


"Power is considered as the most important constraint in embedded systems"
[in: L. Eggermont (ed): Embedded Systems Roadmap 2002, STW]

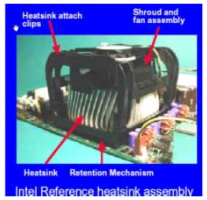
Energy consumption by IT is the key concern of green computing initiatives (embedded computing leading the way)

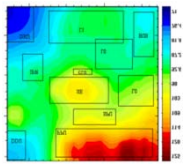
http://www.esa.int/images/earth,4.jpg

Low Power vs. Low Energy Consumption



- Minimizing **power consumption** important for
  - the design of the power supply
  - the design of voltage regulators
  - the dimensioning of interconnect
  - short term cooling
- Minimizing **energy consumption** important due to
  - restricted availability of energy (mobile systems)
    - limited battery capacities (only slowly improving)
    - very high costs of energy (solar panels, in space)
    - RF-powered devices
  - cooling
    - high costs
    - limited space
  - dependability
  - long lifetimes, low temperatures





Introduction

$$P = P_{SC} + P_{SW} + P_{LK}$$

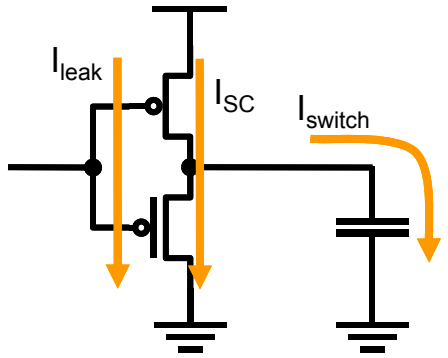
P_{SC} ...Short Circuit Power

P_{SW} ...Switching Power

P_{LK} ...Leakage Power

Minimize I_{leak} by:

- Reducing operating voltage
- Fewer leaking transistors



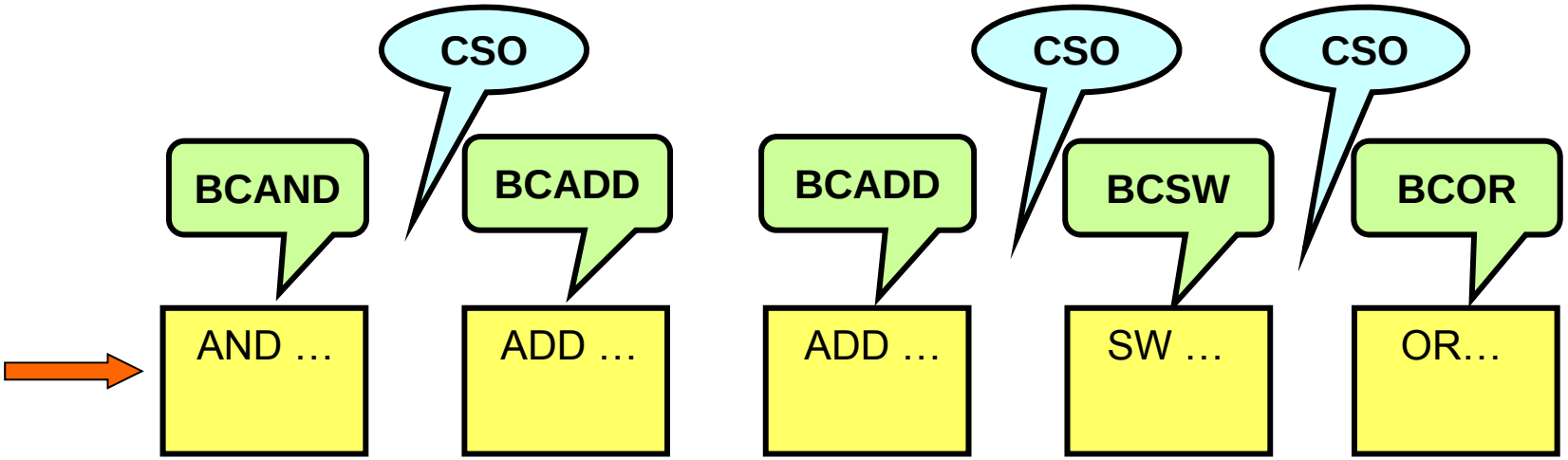
Generic Energy Model

$$E_{total} = \sum_{n=0}^{n_{total}} [E_i(n) + E_d(n) + E_c(n) + E_p(n)]$$

- The overall energy consumption is split into 4 parameters
  - E_i instruction dependent energy dissipation
    - Independent on source and target operands and operand values
    - Estimation based on base cost and CSO (Tiwari et all.)
  - Ed data dependent energy dissipation
    - Energy consumption of each instruction depends on operands and operand values
    - Hamming distance and hamming weight
  - Ec energy dissipation of the cache system
    - Cash hit / miss
  - EP memories and peripherals
    - Power state models
- Huge number of parameters, which have to be characterized

Instruction Path Energy Dissipation

- Considers only instruction flow in pipeline
  - Base Costs (BC)
  - Circuit State Overhead (CSO)



$$E_i = \sum_{k} BC(i) + \sum_{k=0}^{n-1} CSO(instr[k], instr[k+1], k)$$

REVIEW

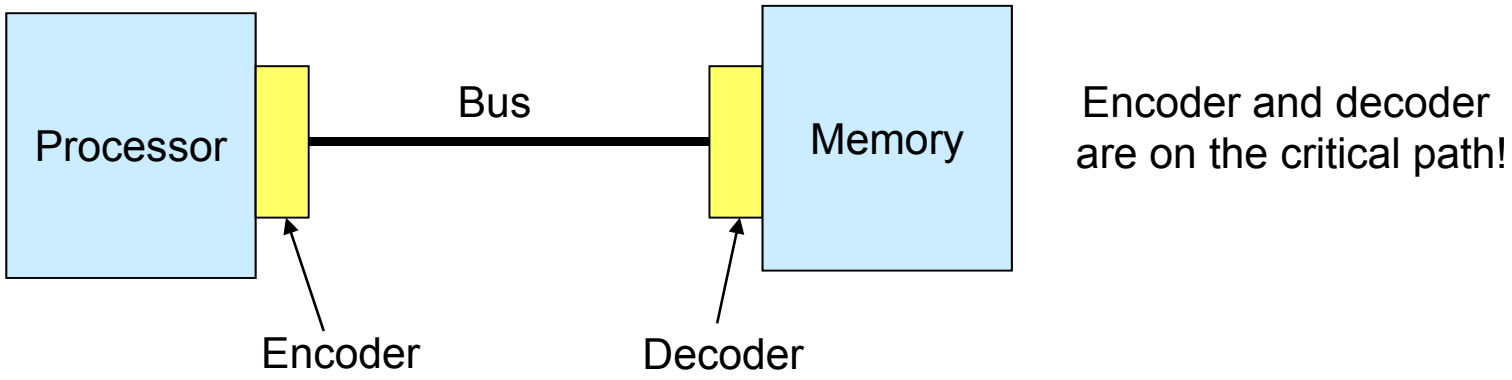
Power/Energy Optimization Levels

- HW level
  - Low power design (transistors, gates, clock gating, …)
- Machine code optimization
  - Operand switching
  - Instruction reordering: minimize circuit state overhead
  - Instruction replacing: use low power instructions
- Source level optimization
  - Algorithmic transformations: simplify computation by reducing quality of service
  - Loop optimization – parallel loads
- HW-System level Power Optimization
  - Data Representation (bus encoding)
  - Memory Design Optimization (access, architecture, partitioning)
- System Level
  - Dynamic Power Management
  - Dynamic voltage scaling / dynamic frequency scaling
  - Remote processing

Bus and Memory Design Optimizations

Data Representation (1)

- System bus lines have a high capacitive load
- Bus system has a high impact on total power
- Use data encoding to reduce switching activity!
  - Choose a representation for the information being transferred such that bus activity is minimized



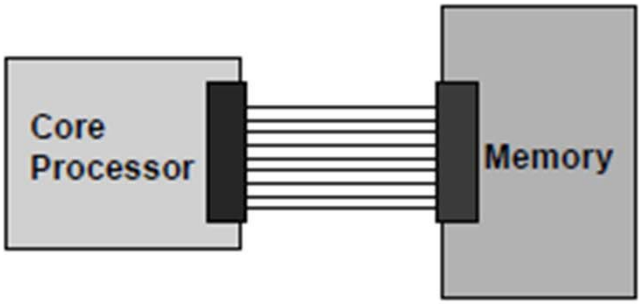
Data Representation (2)

- Bus encoding techniques
  - Bit encoding: Indicates the way 1's and 0's are represented
  - Word encoding

- Codes:
  - Bus-Invert Code: Invert current pattern if hamming distance to previous pattern is larger than 0.5. this technique requires additional redundant bus line.

  - Address-Bus Encoding: Gray code, T0 code has improved performance compared to Gray code for in-sequence addresses by using redundant information

Bus-Invert Code - example



00101010
00111011 2
11010100 7
11110100 1
00001101 6
01110110 6
00010001 5
10000100 4
Original stream:
31 Transitions

00101010 0
00111011 0 2
00101011 1 2
00001011 1 1
00001101 0 3
10001001 1 3
00010001 0 4
10000100 0 4
Encoded stream:
19 Transitions

Overhead of codec must be taken into account when doing the power balance.

- Look at two consecutive patterns, A and B.
- If H(A,B) <= N/2, then transmit B.
- If H(A,B) > N/2, then transmit B'.

N = Bus width.
H(A,B) = Hamming distance between A and B.

Power dissipation on the address bus

- Differences to the data bus:
  - Data not randomly distributed
  - Often sequential addresses (eg. FIFO implemented by a RAM and a counter)
  - Bus Invert coding brings no benefit (just overhead)

  - Gray Coding is the better solution (Hamming distance always 1)
    - Binary code for continuous numbering

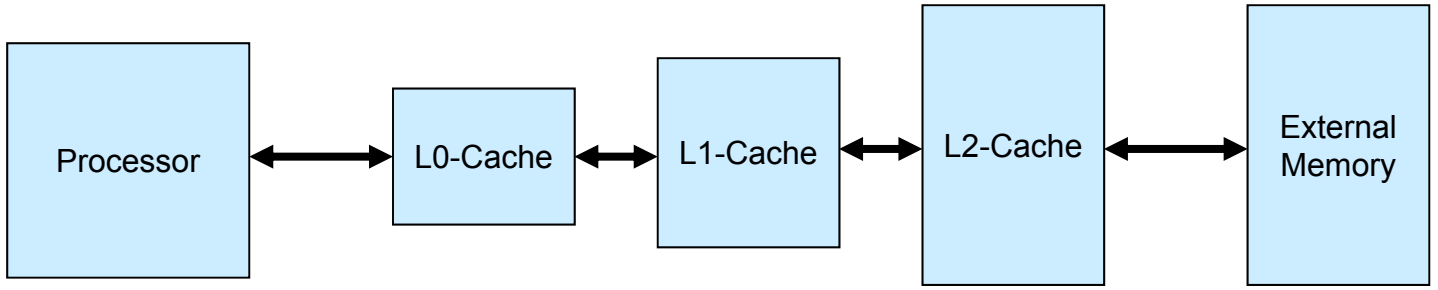
  - sometimes a combination of GRAY coding and Bus Invert Coding is the solution

Memory Design Optimization

- Minimization of memory access power
  - Fixed memory access patterns
    - Optimize memory hierarchy
  - Fixed memory architecture
    - Optimize memory access patterns
  - Concurrent optimization of memory architecture and access patterns
- Minimization of information transfer power
  - Code density optimization
  - Data density optimization

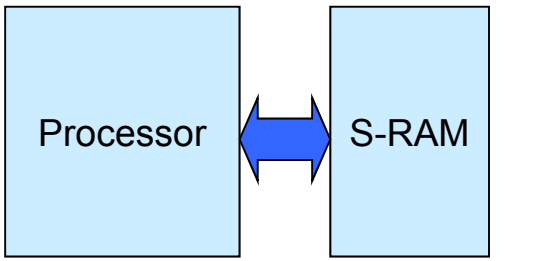
Memory Architecture

- Enforce locality in the cache and memory subsystem
  - Data replication
  - Alternatives to caches (scratch pad buffers)
  - Memory partitioning
- General purpose memory hierarchy

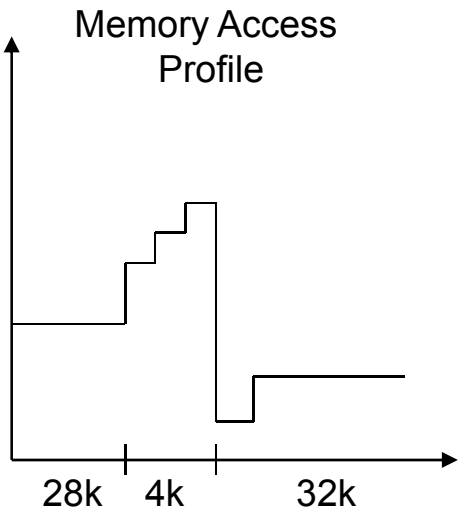


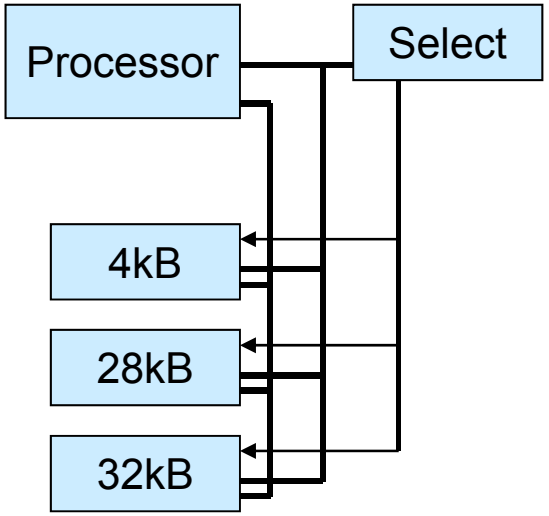
Memory Partitioning

- Power consumption depends on memory block size
- Consider memory access profile and split monolithic memory blocks into several blocks



Monolithic
S-RAM Block





System Level

- Dynamic voltage scaling / dynamic frequency scaling

- Dynamic Power Management

- Remote processing

Fundamentals of dynamic voltage scaling (DVS)

Power consumption of CMOS circuits (ignoring leakage):

$P = \alpha \; C_L \, V_{dd}^2 \; f$ with

α : switching activity

C_L : load capacitance

V_{dd} : supply voltage

f : clock frequency

Delay for CMOS circuits:

$$\tau = k \, C_L \frac{V_{dd}}{(V_{dd} - V_t)^2}$$ with

V_t : threshhold voltage

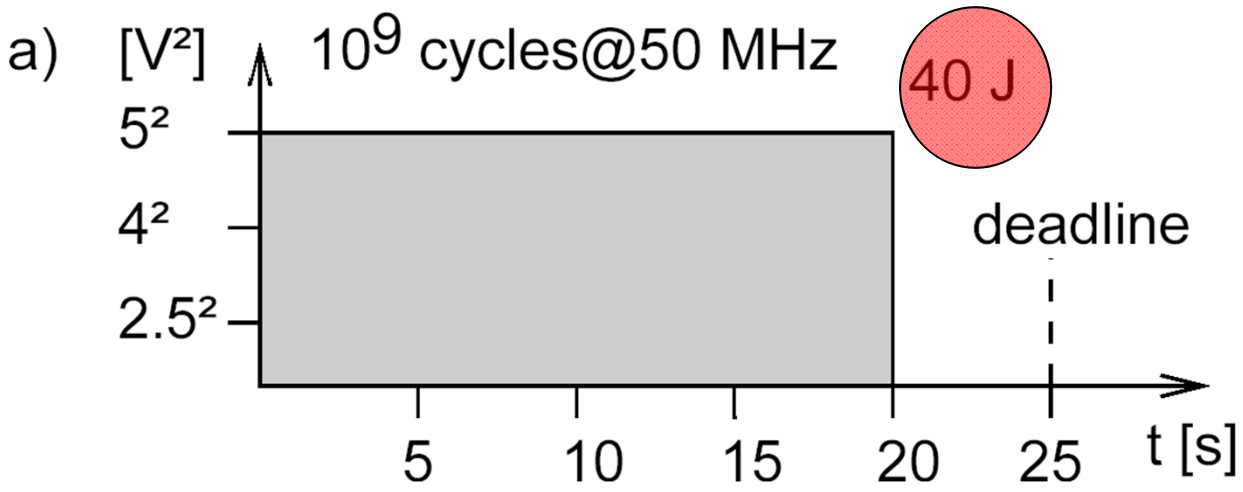
($V_t <$ than V_{dd})

☞ Decreasing V_{dd} reduces P quadratically, while the run-time of algorithms is only linearly increased

Example: Processor with 3 voltages
Case a): Complete task ASAP

- Task that needs to execute 10^9 cycles within 25 seconds.

V_{dd} [V]	5.0	4.0	2.5
Energy per cycle [nJ]	40	25	10
f_{max} [MHz]	50	40	25
cycle time [ns]	20	25	40

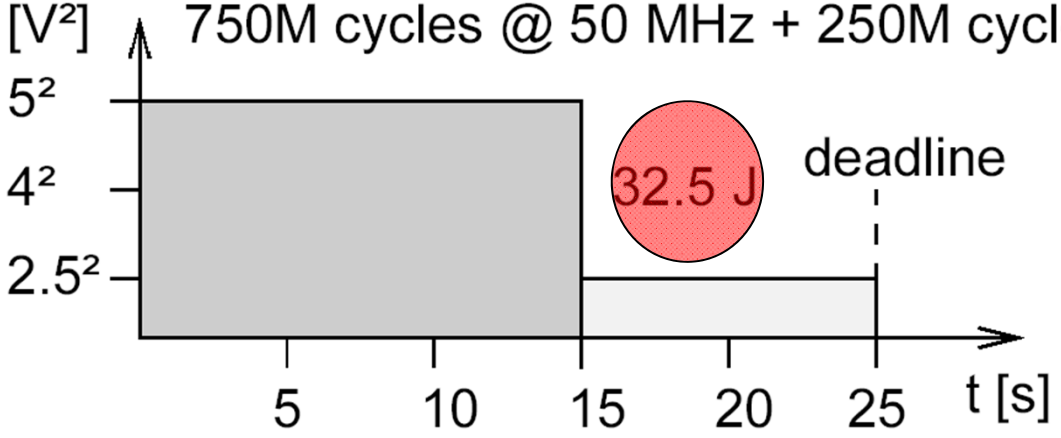


$E_a = 10^9 \times 40 \times 10^{-9}$
$= 40$ [J]

Case b): Two voltages

V_{dd} [V]	5.0	4.0	2.5
Energy per cycle [nJ]	40	25	10
f_{max} [MHz]	50	40	25
cycle time [ns]	20	25	40

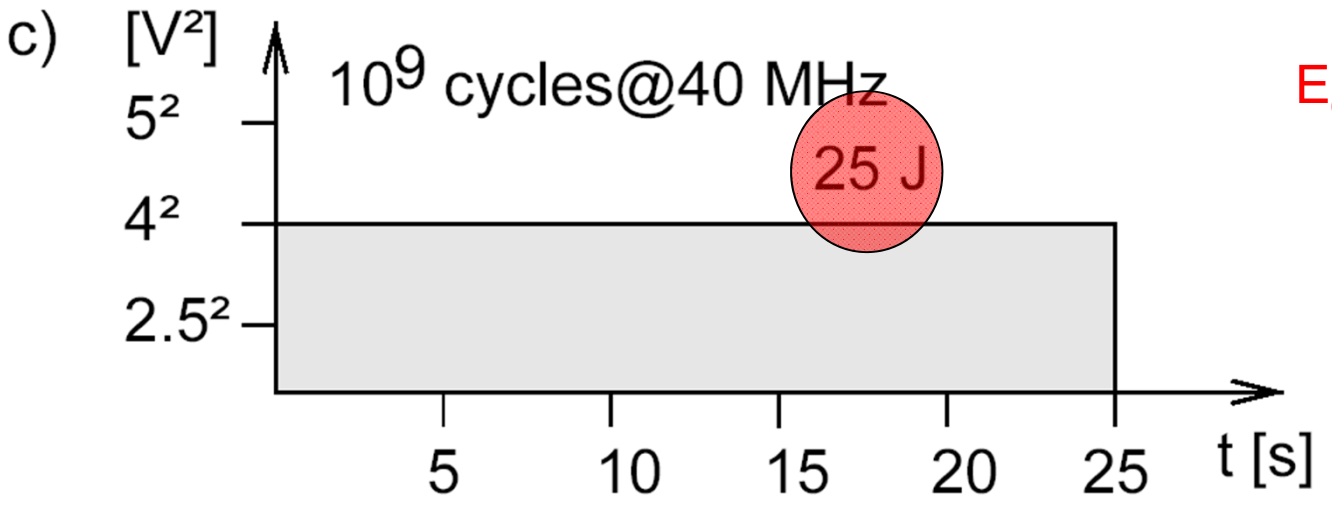
b) [V²] 750M cycles @ 50 MHz + 250M cycles @ 25



E_b= 750 10^6 x 40 10^{-9} +
250 10^6 x 10 10^{-9}
= 32.5 [J]

Case c): Optimal voltage

V_{dd} [V]	5.0	4.0	2.5
Energy per cycle [nJ]	40	25	10
f_{max} [MHz]	50	40	25
cycle time [ns]	20	25	40



$E_c = 10^9 \times 25 \times 10^{-9}$
$= 25$ [J]

Observations

- A minimum energy consumption is achieved for the ideal supply voltage of 4 Volts.
- In the following: variable voltage processor = processor that allows any supply voltage up to a certain maximum.
- It is expensive to support truly variable voltages, and therefore, actual processors support only a few fixed voltages.

Low voltage, parallel operation more efficient than high voltage, sequential operation

Basic equations

Power:	$P \sim V_{DD}^2$,
Maximum clock frequency:	$f \sim V_{DD}$,
Energy to run a program:	$E = P \times t$, with: t = runtime (fixed)
Time to run a program:	$t \sim 1/f$

Changes due to parallel processing, with α operations per clock:

Clock frequency reduced to:	$f' = f / \alpha$,
Voltage can be reduced to:	$V_{DD}' = V_{DD} / \alpha$,
Power for parallel processing:	$P° = P / \alpha^2$ per operation,
Power for α operations per clock:	$P' = \alpha \times P° = P / \alpha$,
Time to run a program is still:	$t' = t$,
Energy required to run program:	$E' = P' \times t = E / \alpha$

☞Argument in favour of voltage scaling, VLIW processors, and multi-cores

Rough approxi-mations!

Application: VLIW procesing and voltage scaling in the Crusoe processor

- VDD: 32 levels (1.1V - 1.6V)
- Clock: 200MHz - 700MHz in increments of 33MHz

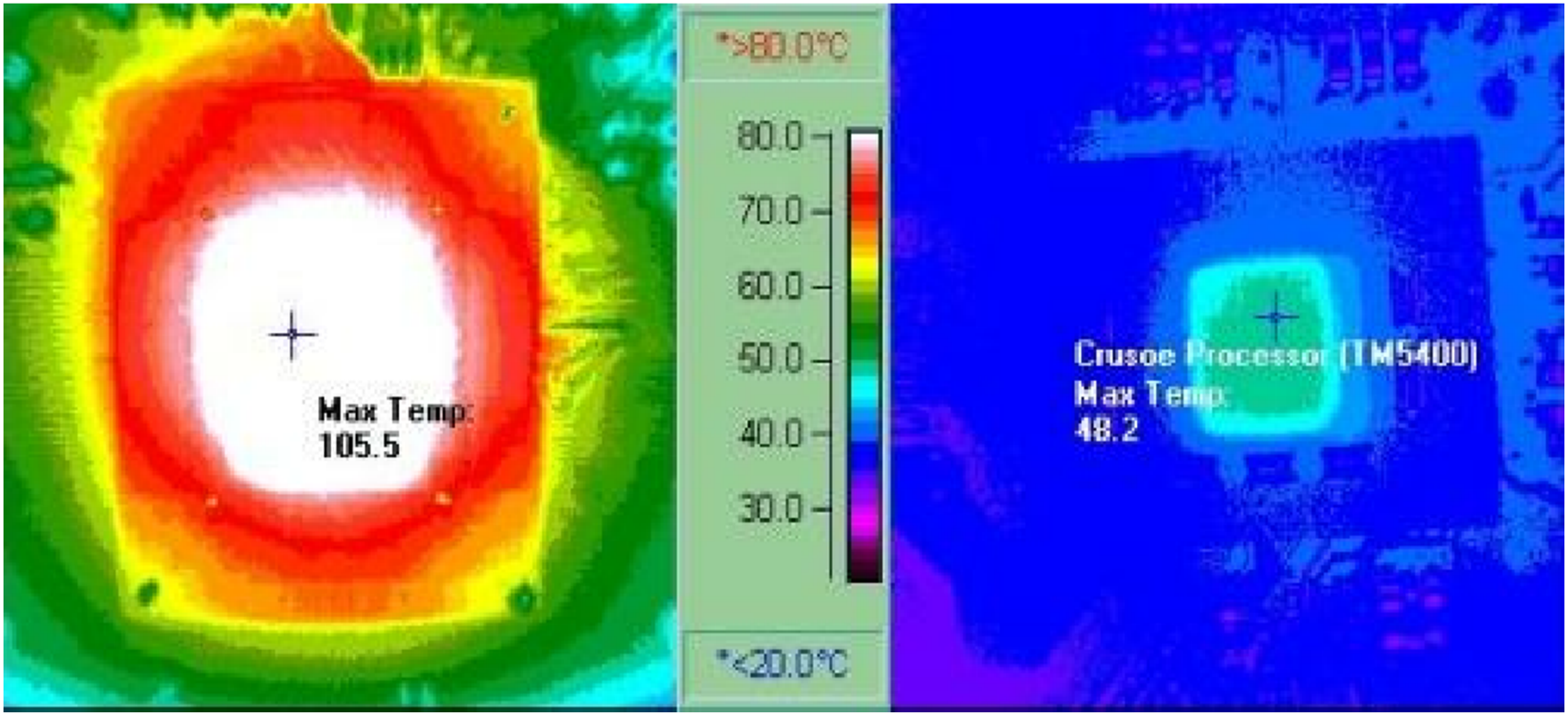
- Scaling is triggered when CPU load change is detected by software
  - More load: Increase of supply voltage (~20 ms/step), followed by scaling clock frequency
  - Less load: reduction of clock frequency, followed by reduction of supply voltage

- Worst case (1.1V to 1.6V VDD, 200MHz to 700MHz) takes 280 ms

Result (as published by transmeta)

Pentium

Crusoe



Running the same multimedia application.

[www.transmeta.com]

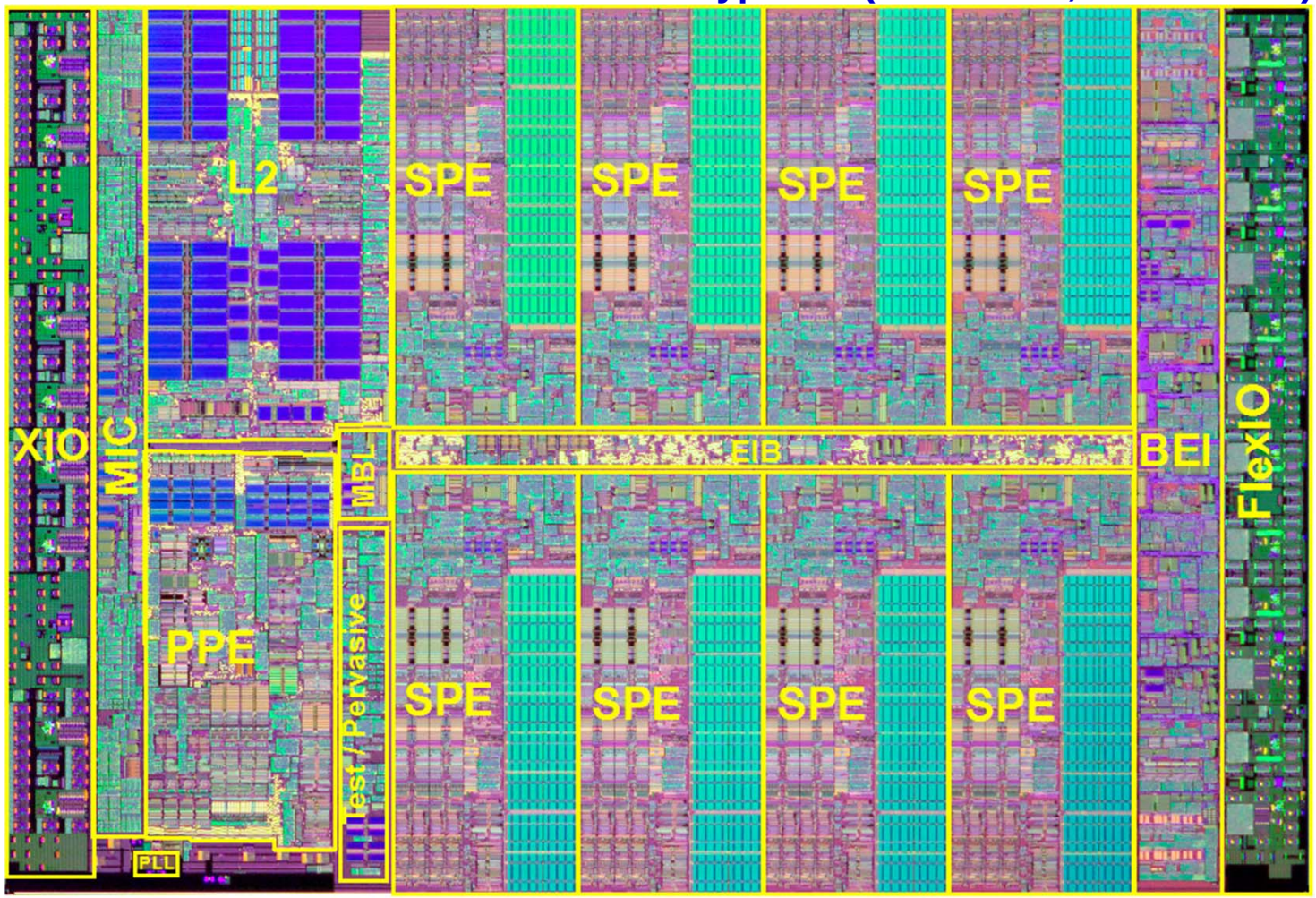
More parallelism

- As long as enough parallelism exists, it is more efficient to achieve the same performance by doubling the number of cores rather than doubling the frequency.

- There are at least three camps in the computer architects community
  - Multi-cores - Systems will continue to contain a small number of "big cores" – Intel, AMD, IBM
  - Many-cores – Systems will contain a large number of "small cores" – Sun T1 (Niagara)
  - Asymmetric-cores – combination of a small number of big cores and a large number of small cores – IBM *Cell architecture (Playstation 3)*

Cell Overview

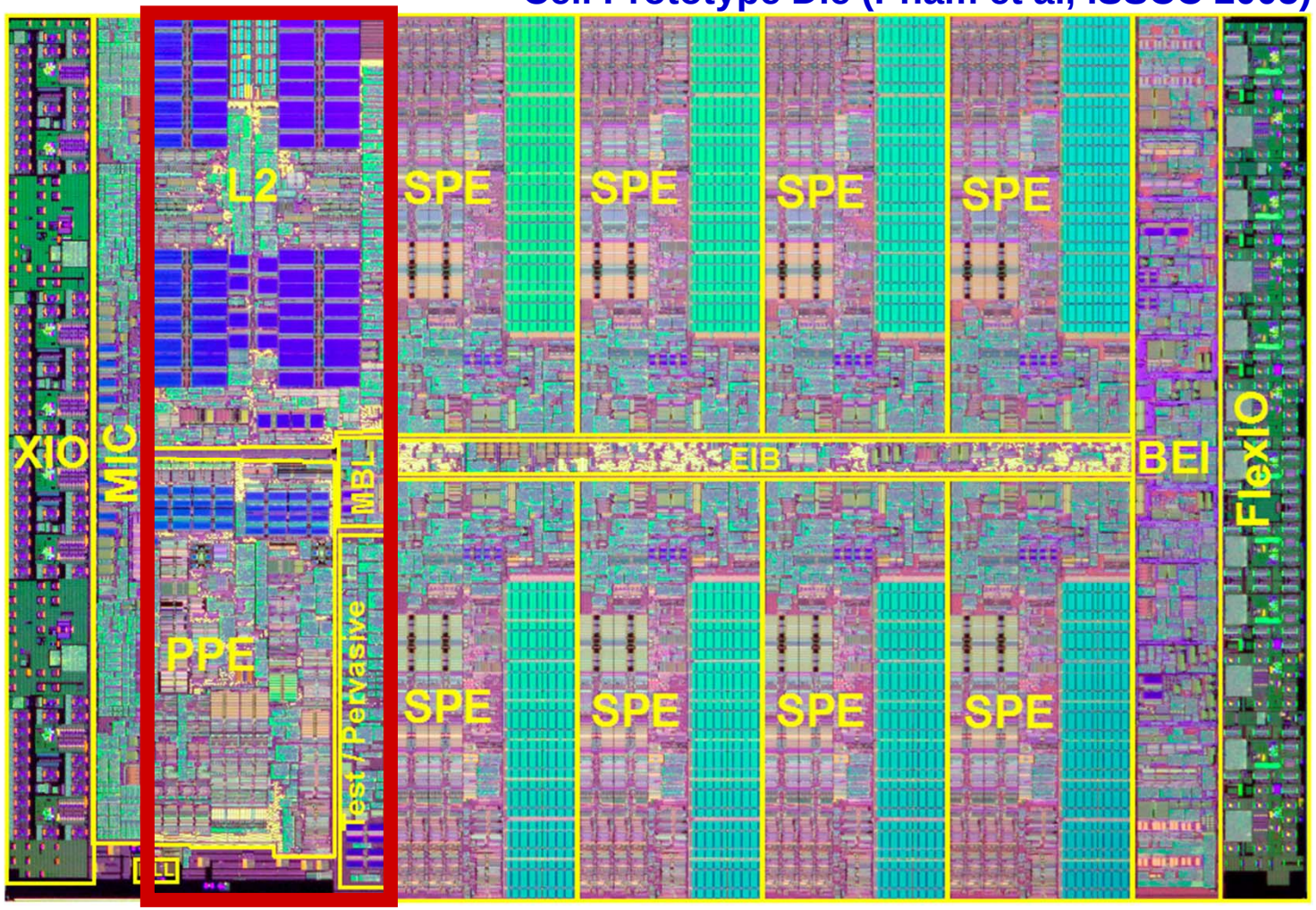
Cell Prototype Die (Pham et al, ISSCC 2005)



- IBM/Toshiba/Sony joint project - 4-5 years, 400 designers
  - 234 million transistors, 4+ Ghz
  - 256 Gflops (billions of floating pointer operations per second)

Cell Overview - Main Processor

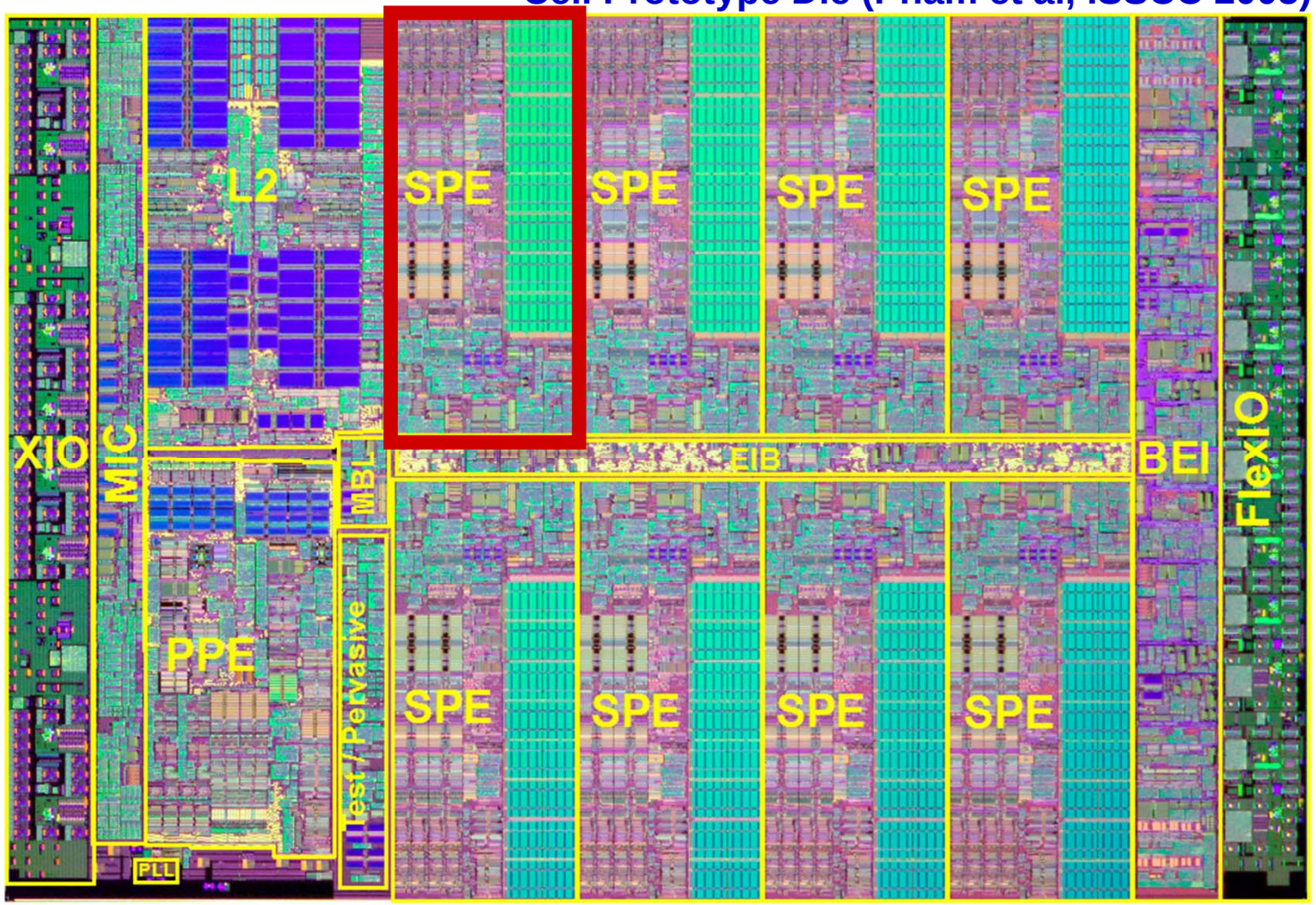
Cell Prototype Die (Pham et al, ISSCC 2005)



- One 64-bit PowerPC processor
  - 4+ Ghz, dual issue, two threads
  - 512 kB of second-level cache

Cell Overview - SPE

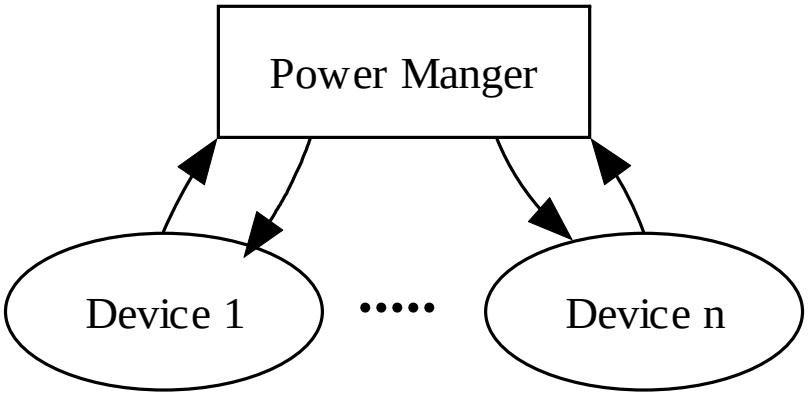
Cell Prototype Die (Pham et al, ISSCC 2005)



- Eight Synergistic Processor Elements
  - Or “Streaming Processor Elements”
  - Co-processors with dedicated 256kB of memory (not cache)

Dynamic power management (DPM)

- The power manager (PM) implements a control procedure based on observations and assumptions about the workload.
- The control procedure is called a policy.
- “Oracle” power manager

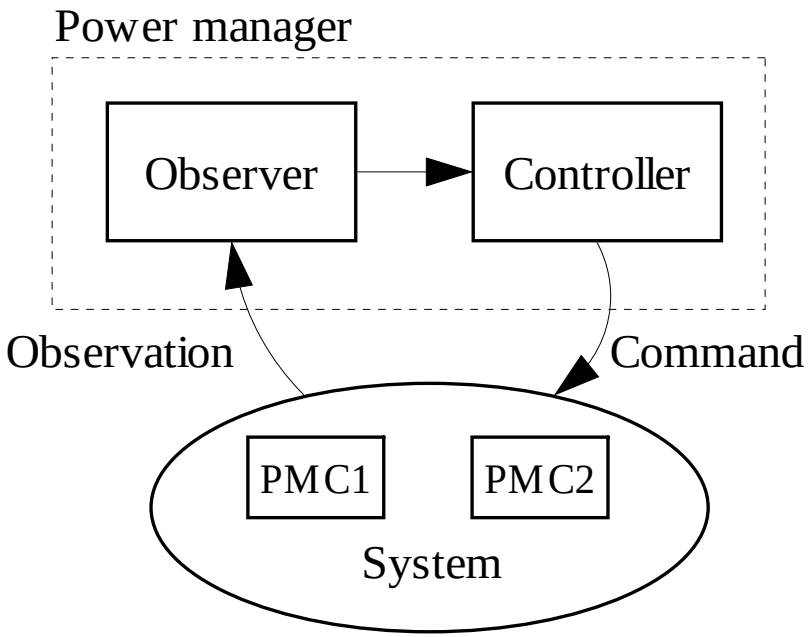


Implementation

- Hardware
  - Frequency reduction
  - Supply voltage
  - Power shutdown
- Software
  - Mostly used
  - Most flexible
- Operative system power manager (OSPM)
  - Microsoft's OnNow
  - ACPI (Advanced Configuration and Power Interface)

Modeling

- View the system as a set of interacting power-manageable components (PMCs), controlled by the power manager (PM).



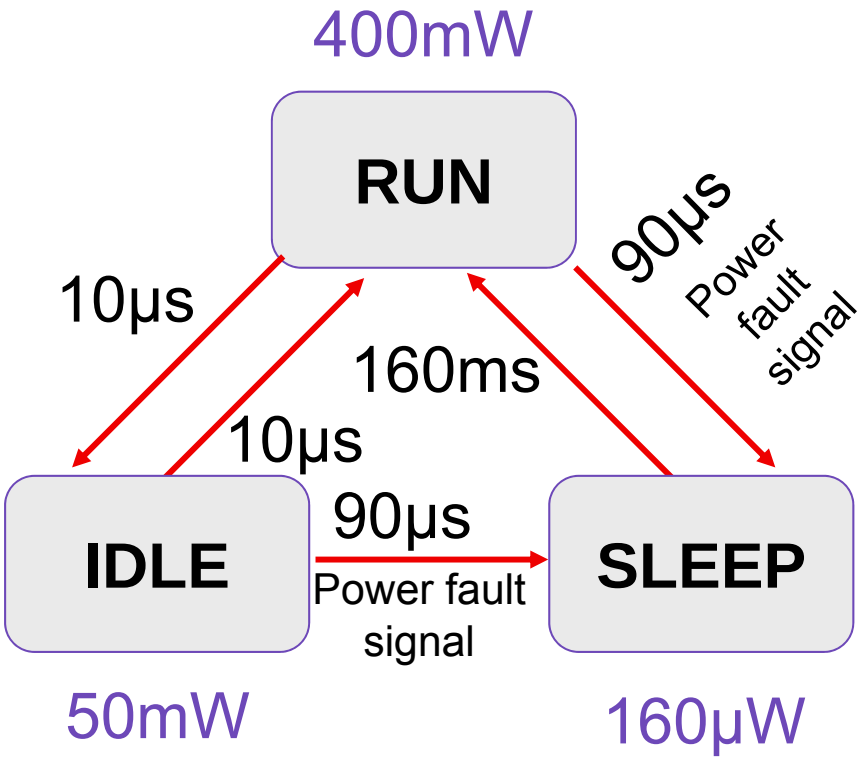
Modeling

- Independent PMCs.
- Model PMCs as FSMs
- Transition between states have a cost.
- The cost is associated with delay, performance and power loss.
- Service providers and service requesters.

Dynamic power management (DPM)

Example: STRONGARM SA1100

- RUN: operational
- IDLE: a sw routine may stop the CPU when not in use, while monitoring interrupts
- SLEEP: Shutdown of on-chip activity



power states

Power and performance issues..

- Break-even time T_{be} - minimum length of an idle period to save power. Move to sleep state if $T_{idle} > T_{be}$
  - T_0 : Transition delay (shutdown and wakeup)
  - E_0 : Transition energy
  - P_s , P_w : Power in sleeping and working states

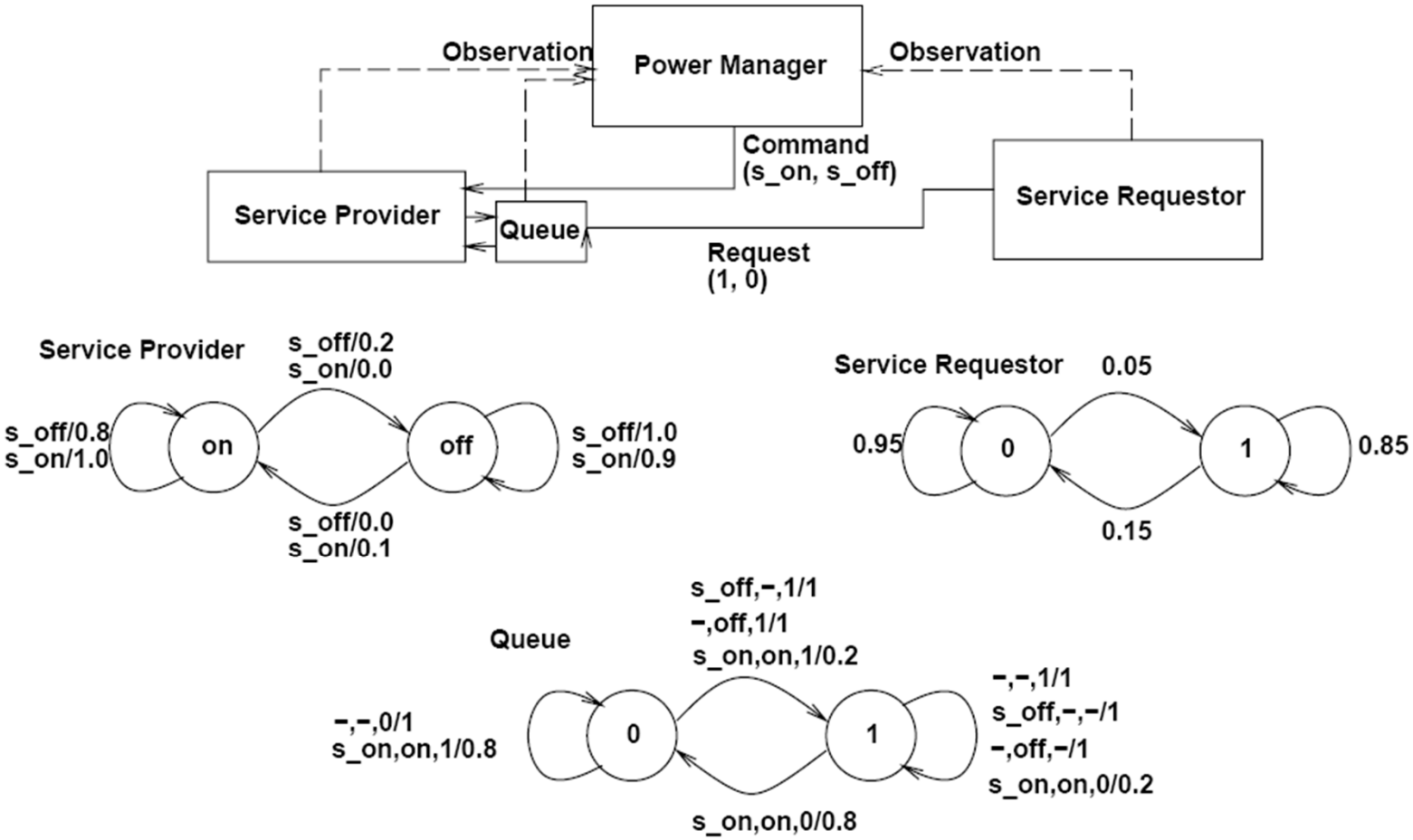
$$P_w \times T_{be} = E_0 + P_s(T_{be} - T_0) \Rightarrow T_{be} = \frac{(E_0 - P_s \times T_0)}{P_w - P_s}$$

$$T_{be} = \max\left[\frac{E_0 - P_s \times T_0}{P_w - P_s}, T_0\right]$$

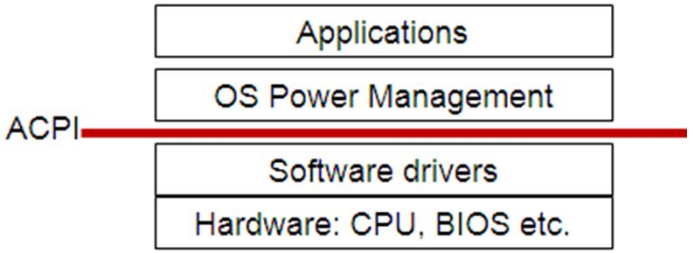
Policies

- Different categories:
  - Predictive
  - Adaptive
  - Stochastic
- Application dependent
- Statistical properties
- Resource requirements

System modeling



ACPI (Advanced Configuration and Power Interface)

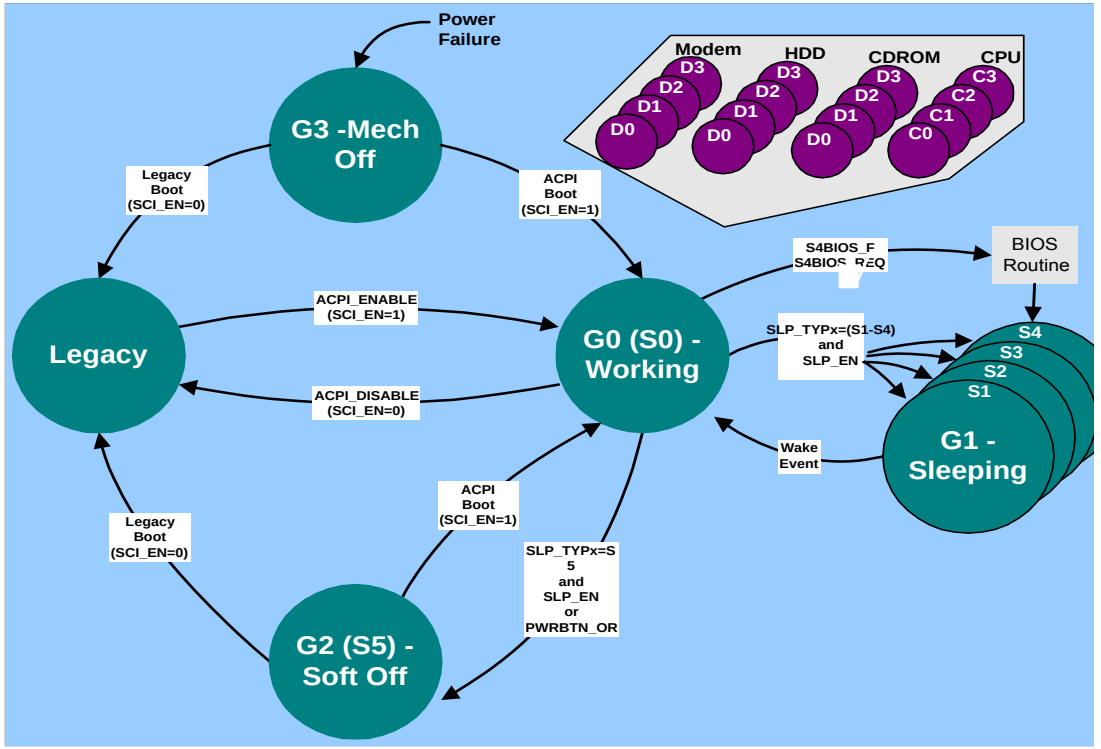


- ACPI is platform independent general specification
  - Integrate power management features in low level routines
  - Communicating directly with hardware

- Defines:
  - Interfaces between OS and Hardware
  - Applications interact with OS using *APIs*
  - A module in OS communicate with hardware
  - Power management module interacts with hardware
    - Kernel services (system calls)

ACPI

- States:
  - Working (G0) → Sleeping (G1)
  - Idle devices → Sleep states (D0-D3)
  - CPU put to sleep (C0-C3)
  - Sleep Substates(S1-S4),
    - Differ on wake events
  - Soft off state (G2)
  - Mechanical off-state(G3)
  - Legacy state



ACPI State Hierarchy (2/3)

- Global system states (g-state)
- **G0** : Working
  - **Processor power states (C-state)**
  - **C0** : normal execution
  - **C1** : idle
  - **C2** : lower power but longer resume latency than C1
  - **C3** : lower power but longer resume latency than C2
- **G1** : Sleeping (e.g., suspend, hibernate)
  - **Sleep State (S-state)**
  - **S0-S4**
- **G2** : Soft off (**S5**)
- **G3** : Mechanical off

ACPI State Hierarchy (3/3)

Frequency	Voltage	P-State
1.6 GHz	1.484 V	P0
1.4 GHz	1.420 V	P1
1.2 GHz	1.276 V	P2
1.0 GHz	1.164 V	P3
800 MHz	1.036 V	P4
600 MHz	0.956 V	P5

Intel Pentium M at 1.6GHz

- **G0** : Working
  - **Processor power states (C-state)**
  - **C0** : normal execution
    - **Performance state (P-State)**
    - **P0:** highest performance, highest power
    - **P1**
    - **Pn**
  - **C1, C2, C3**
- **G1** : Sleeping (e.g., suspend, hibernate)
  - **Sleep State (S-state): S0, S1, S2, S3, S4**
- **G2** : Soft off (**S5**)
- **G3** : Mechanical off

Framework for Power Aware Remote Processing

Gerald Käfer, Ph.D thesis @ ITI, TU Graz

In order to reduce the power consumption of mobile distributed systems, modern devices have to use available wireless networks for remote processing

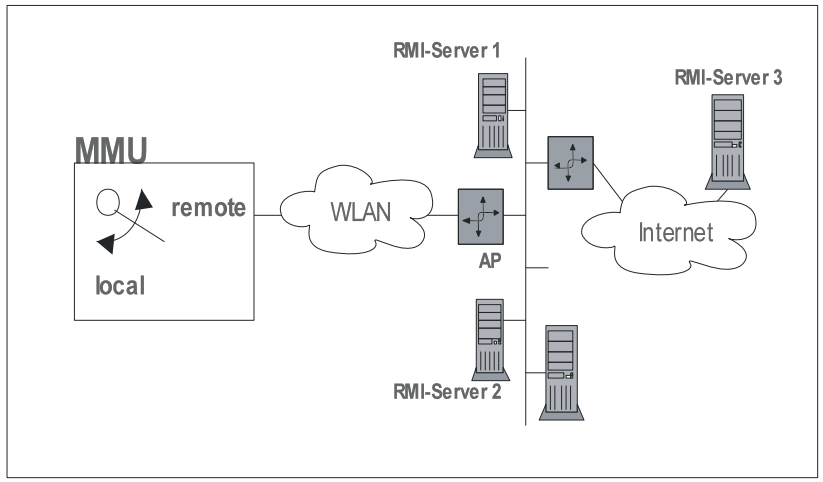
Partner: www.xybernaut.com (Fairfax, US)

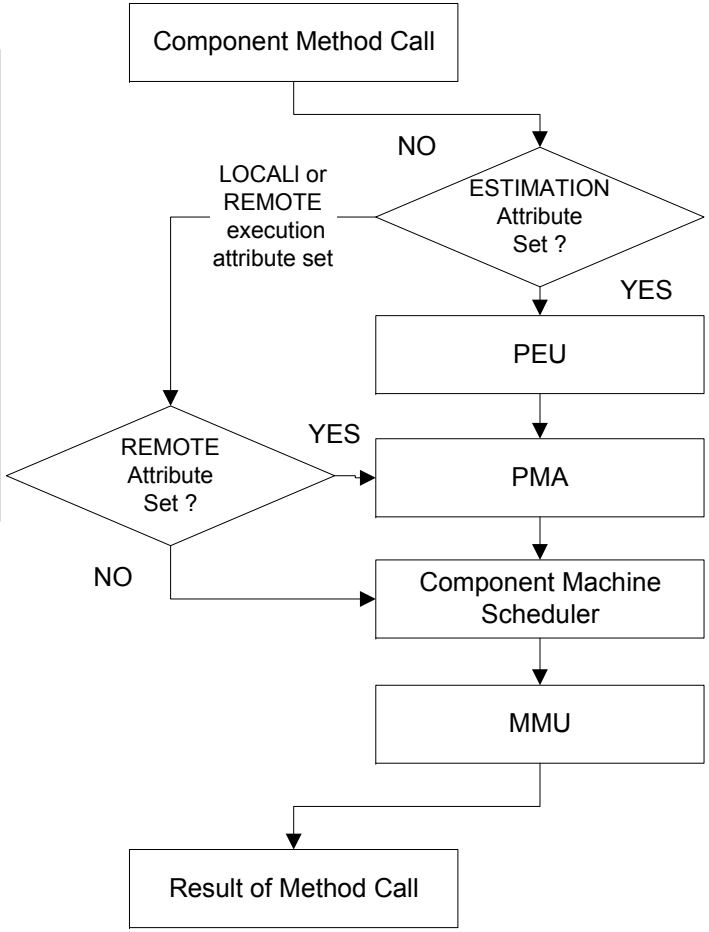
Framework for Power Aware Remote Processing (1)

AIM	Reduction of mobile device‘s energy consumption by selective task migration to remote network servers.
NEED	Relation between software and energy consumption and transparent code migration!
IDEA	Framework for Power Aware Remote Processing

Framework for Power Aware Remote Processing (2)

Framework (Working Cycle)

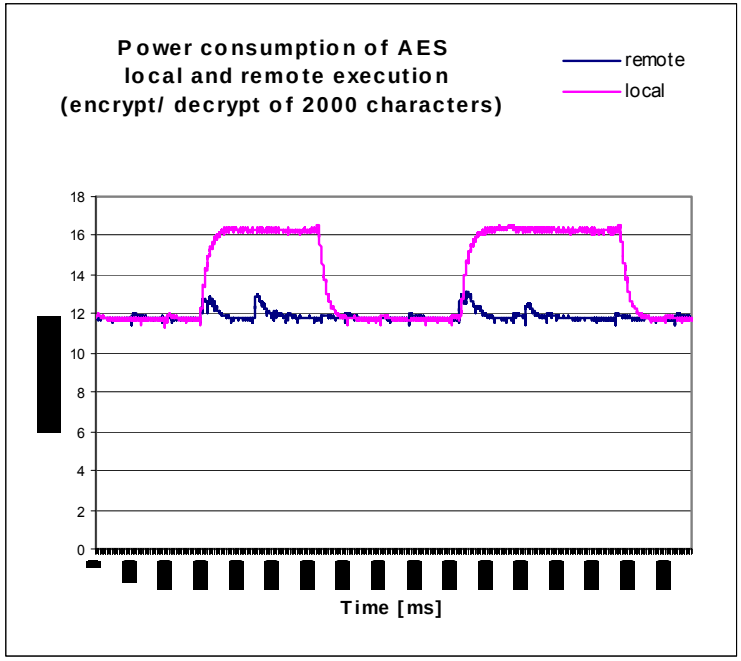


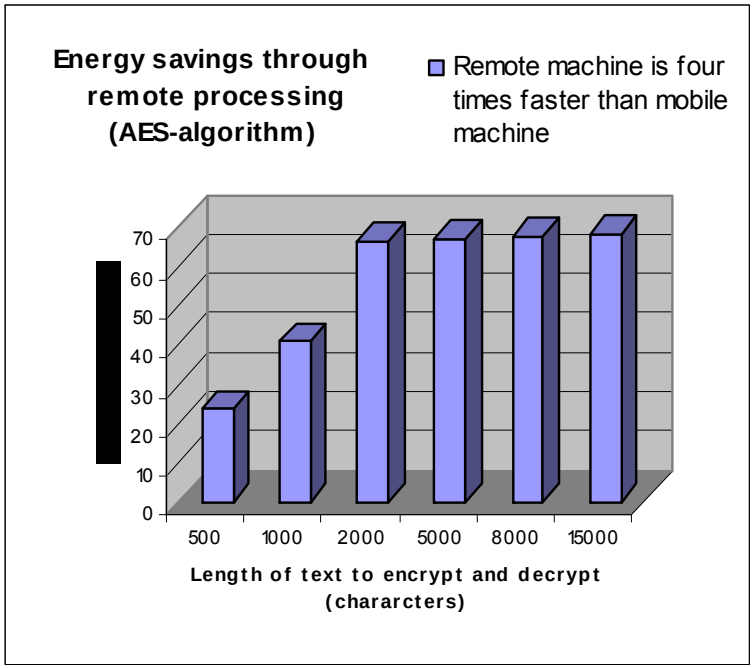


PEU... Power Estimation Unit
PMA... Power Management Agent
MMU... Multi Machine Unit

Framework for Power Aware Remote Processing (3)

Evaluation of AES (Advanced Encryption Standard)



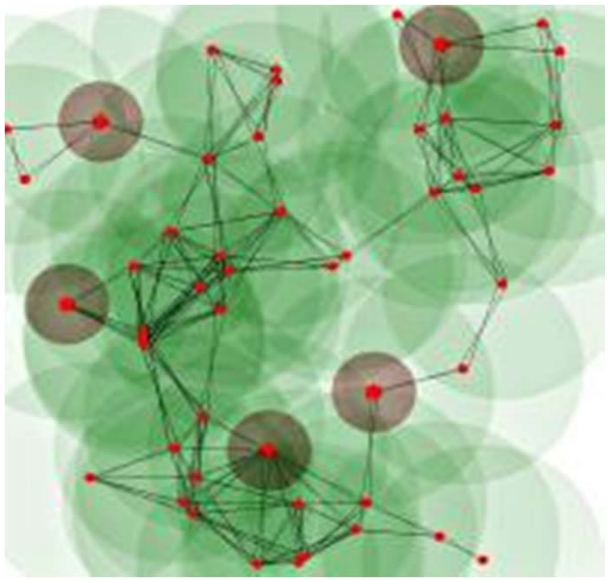


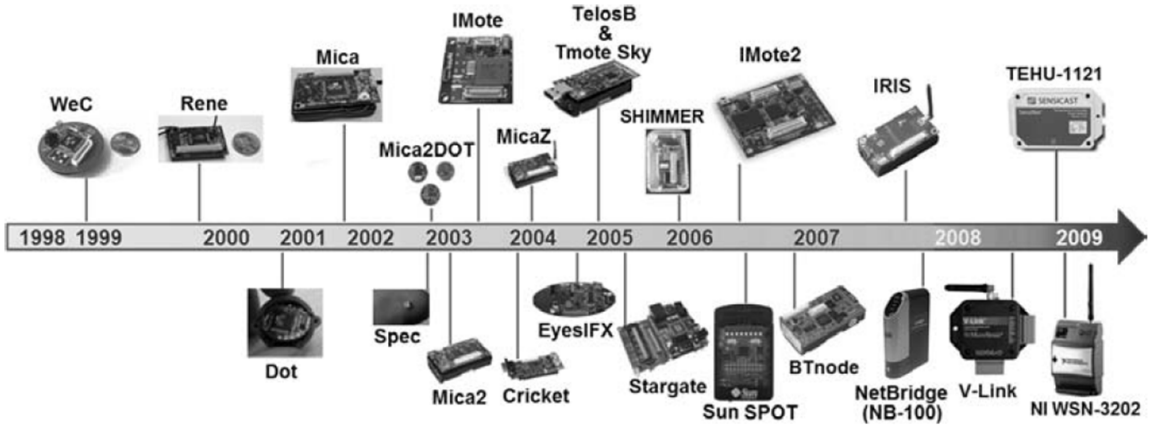
70% savings of energy consumption possible for AES!

Wireless ad hoc Sensor Networks and Power Awareness

Wireless Sensor Network

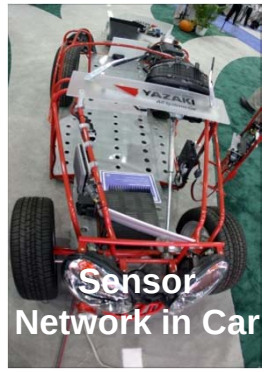
- Collection of small, locally powered, intelligent sensor nodes
- Communicate detected events over a wireless channel (typically through multi-hop routing).
- WSNs are continuing to receive an escalating research interest, due in part to the considerable range of applications that they are suited to.



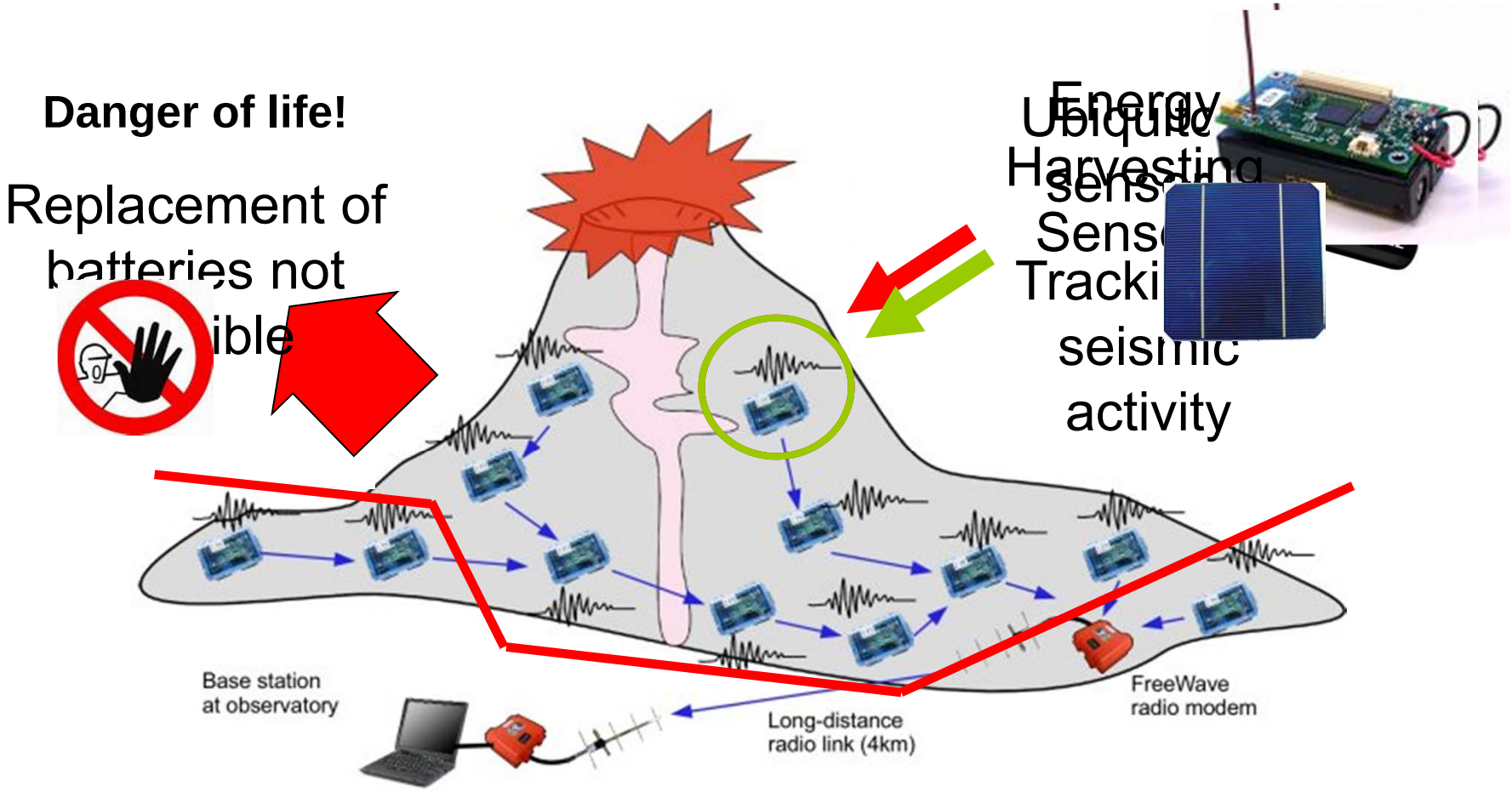


Applications

- Environmental monitoring
  - Habitat monitoring
  - Precision agriculture
  - Security, surveillance
- Structure and equipment monitoring
  - Structural dynamics
  - Condition-based maintenance
  - Emergency response
- Supply chain monitoring
  - Manufacturing flows, asset tracking
- Context aware computing
  - Information beacons



Ubiquitous sensors + Energy Harvesting



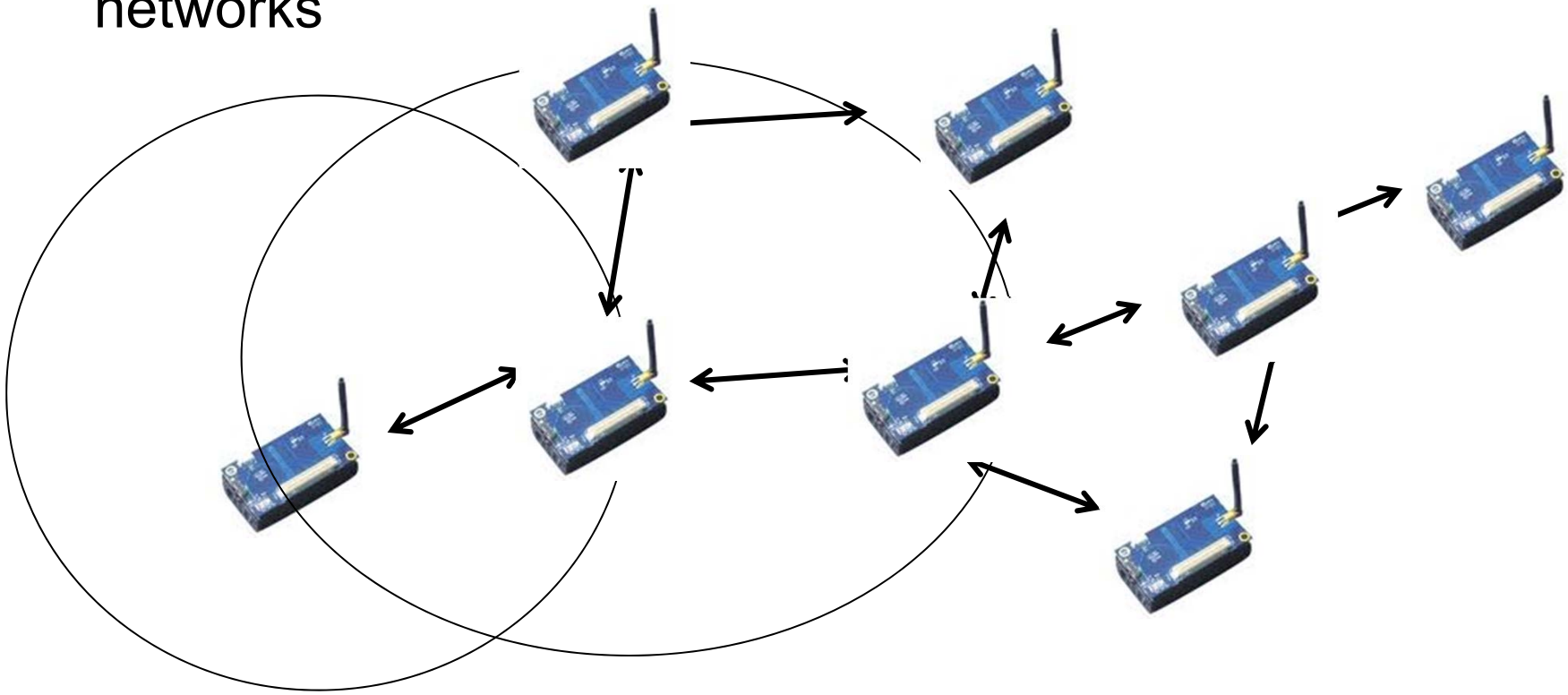
Energy harvesting: operation almost indefinitely

Introduction Ad-Hoc Networks

- Ad Hoc is a Latin phrase and means “for this purpose”
- Ad-Hoc Networks
  - Wireless Networks with two or more subscribers
  - No fix infrastructure
  - The connection is established for the duration of one session
  - Devices discover others within range to form a network
  - To reach devices out of the range, devices flood the network with broadcast.
  - Each node forwards every broadcast.

Introduction Ad-Hoc Networks

- Limited Range of the nodes
- Communication with every node needs multi-hop networks



Wireless Devices and Sensor Networks

- Low-end platforms: Mica family, Telos/Tmote, EYES
  - Mica, Mica2, MicaZ, IRIS (Crossbow)
  - 8 bit Atmel AVR MCU, 4-16 MHz, 128-256 kB flash
  - Mica/2: 433/868/916 MHz, 40 kbps, -Z/IRIS: IEEE 802.15.4, 2.4 GHz 250 kbps
  - 4-8 kB RAM, 512 kB data memory
  - 51-pin connector

Wireless Devices and Sensor Networks

- High-end platforms: Stargate, Imote, Sun SPOT
  - Sun SPOT: uses a Sun Java Micro Edition; 180 MHz, 32 bit ARM920T; 512k RAM, 4M flash
  - 2.4 GHz IEEE 802.15.4-enable transceiver

Processor board state	Radio	Sensor Board	Current draw
Deep sleep mode	Off	Any	~33 microamperes
Shallow sleep[1]	Off	Not present	~24 milliamperes
Shallow sleep	On	Not present	~40 milliamperes
Awake, actively calculating	Off	Not present	~80 milliamperes
Awake, actively calculating	On	Not present	~98 milliamperes
Shallow sleep	Off	Present	~31 milliamperes
Shallow sleep	On	Present	~46 milliamperes
Awake, actively calculating	Off	Present	~86 milliamperes
Awake, actively calculating	On	Present	~104 milliamperes

1. Shallow sleep means devices active, but no active threads.

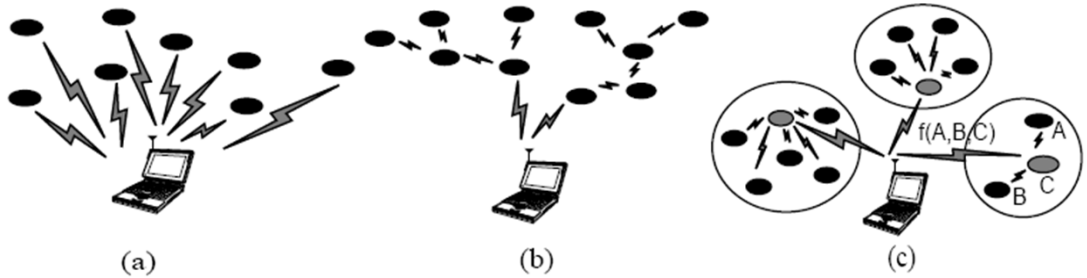
Power Awareness

Hardware Level:
- Micro Controller Unit (MCU)
- Radio
- Sensors
- Battery Design

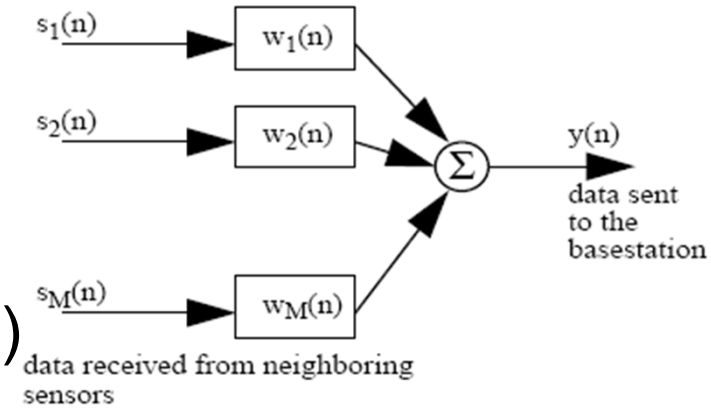
Software Level:
- Energy Aware Software
- Power Aware Computing
- Power Management of Radios
- Power Management of MCU

Communication Techniques:
- Modulation Schemes
- Link Layer Optimizations
- Energy Aware Packet Routing/Forwarding
- Traffic Distribution
- Topology Management
- Computation/Communication Tradeoff

Signal Processing in the Network



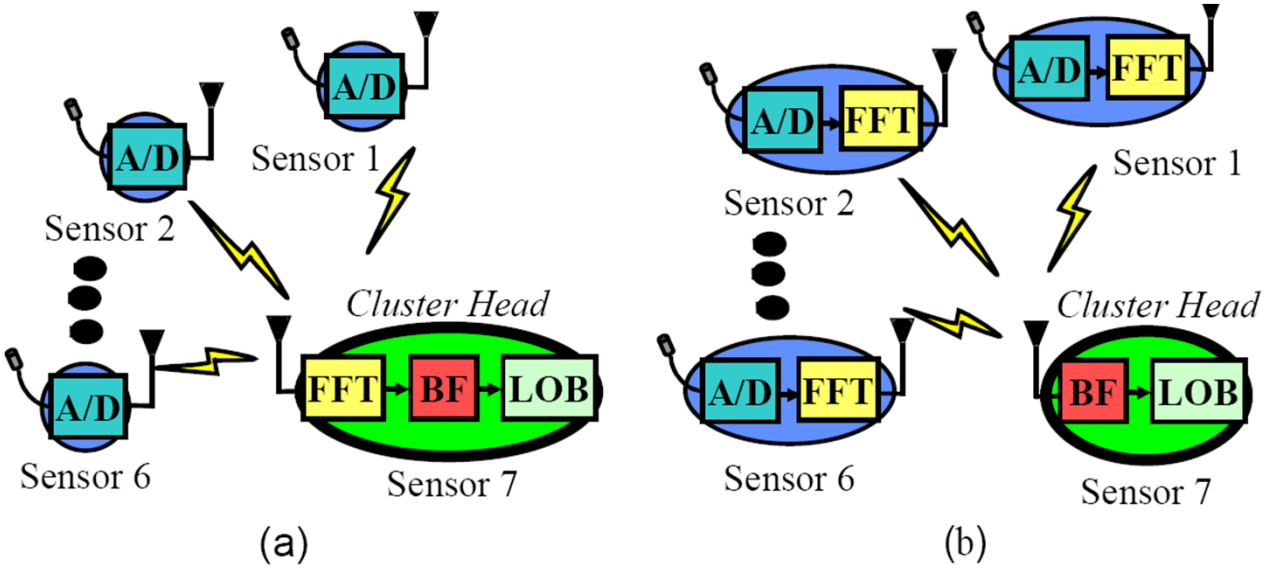
a) Direct Communication
b) Multi-hop with the basestation
c) Clustering algorithm
   - Rotating *cluster-head*
   - Data aggregation (e.g. beamforming)
   - Reduces data to the basestation
   - Energy efficience



System Partitioning

a) All computation is done at the cluster-head
   - 1024-point FFTs

b) Computating in parallel
   - Greater latency per computation
   - Energy Savings through f and V scaling
   - 44% improvement in energy dissipation



Components for mobile PA ES

- Energy storage structures

- Energy harvesting devices

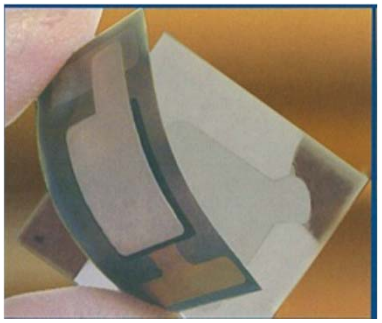
Energy storage structures

- Primary (not rechargeable)
  - Batteries
  - Nuclear microbatteries
  - Fuel cells

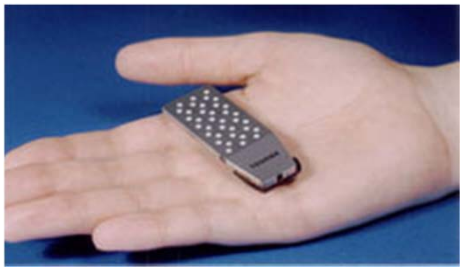
- Secondary (rechargeable)
  - Accumulators
  - Ultracapacitors

Energy storage structures

- Batteries
  - + common
  - - battery effects
- Microbatteries
  - + very small size
  - - very low capacity
- Fuel cells
  - + high energy density
  - - low efficiency at ambient temperature
  - - low voltage
- Electrochemical capacitors
  - + no battery effects
  - + high cycle life



© Infinite Power Solutions, Inc.

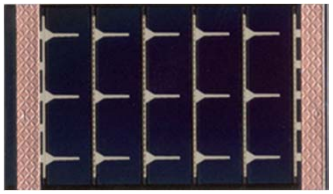


© TOSHIBA CORPORATION

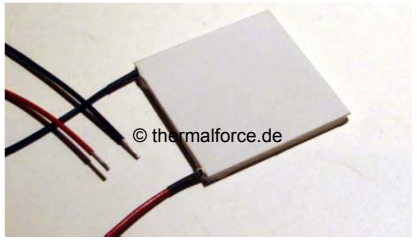
© Maxwell Technologies, Inc.

Energy harvesting devices

- Solar cells
  - + Stable voltage output
  - - Low efficiency

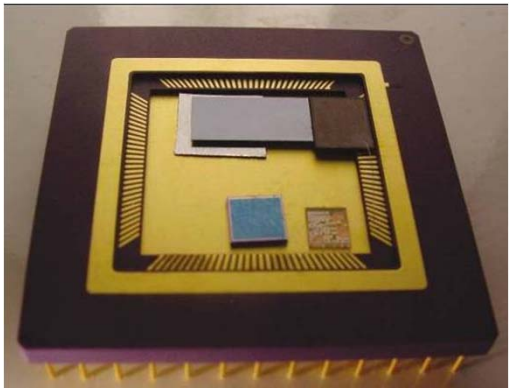


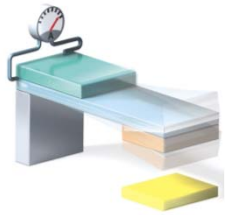
- Thermoelectric generators
  - - High temperature difference required



- Piezoelectric generators
  - - Vibration source required

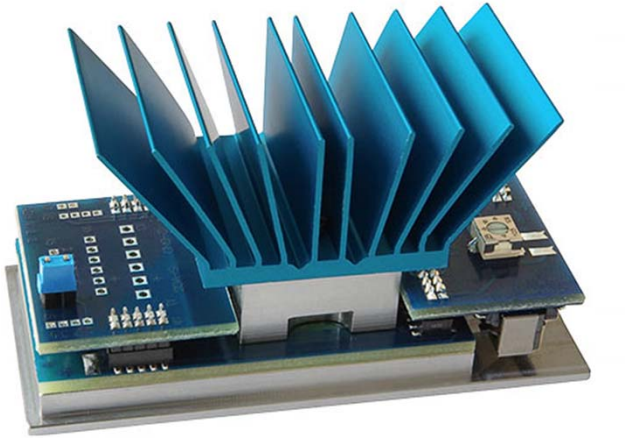
- Nuclear microbatteries
  - + Extremely long lifetime
  - - Low power output
  - - Difficult to obtain





Source: [Lal-2004]

Energy harvesting devices – www.micropelt.com



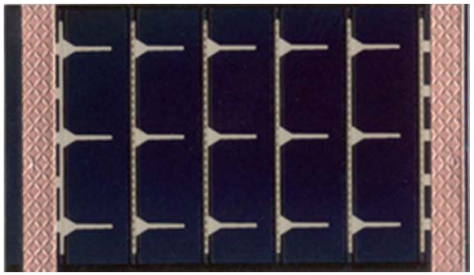
» TE-Power PROBE (» Buy at Mouser)

Screw or plug-in thermoharvester for dry or direct liquid attach.
Optimized for natural convection

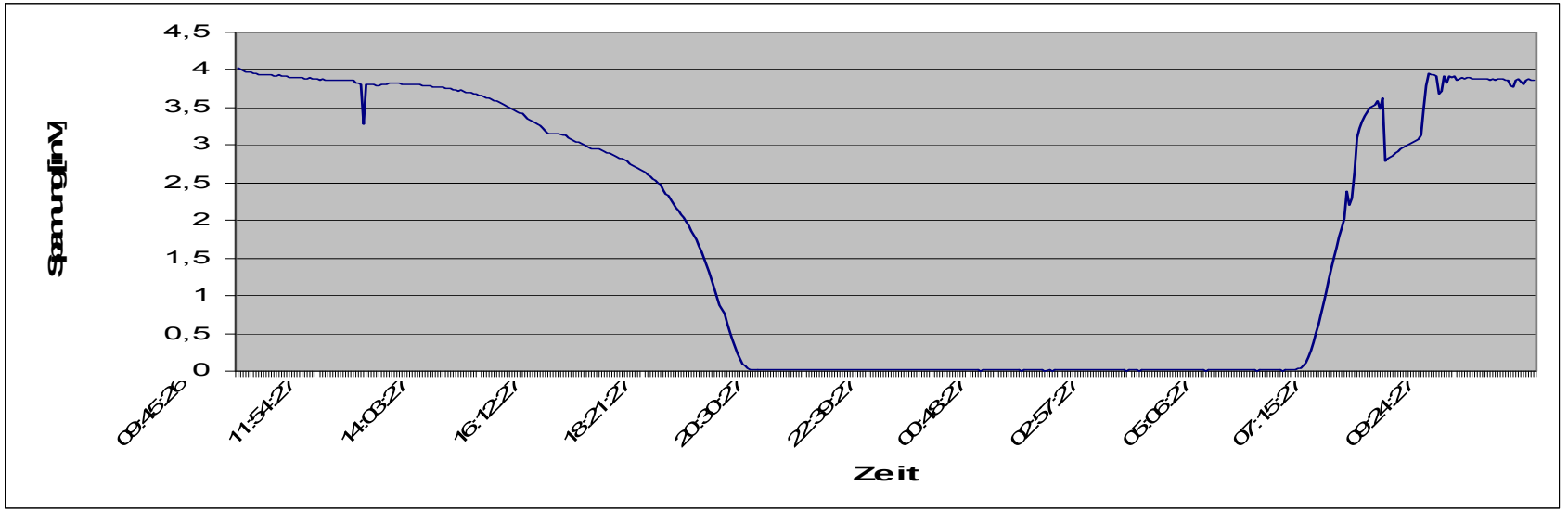
- High-performance heatsink
- Voltage and energy storage preselect
- Threaded dry attach or T-joint to liquid

e.g. energy harvesting device – solar cell

- For general purpose applications solar cells are suited best
  - For certain specific areas thermogenerators and piezogenerators may be suitable as well
- PowerFilm Inc. SP3-37
  - Good results under various conditions
  - Small, thin and flexible



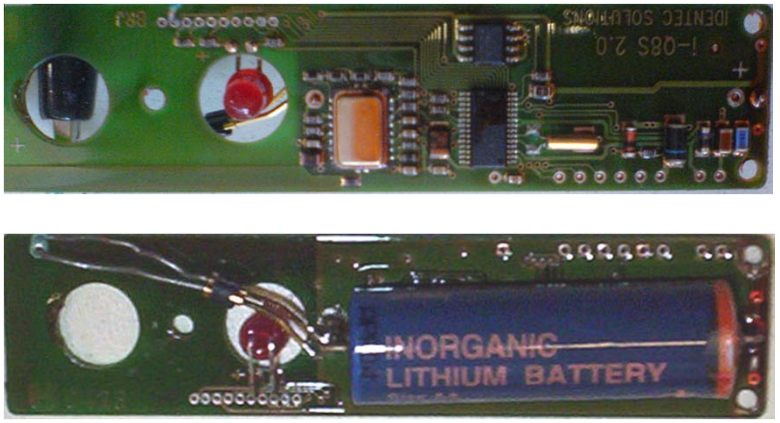
© PowerFilm Inc.

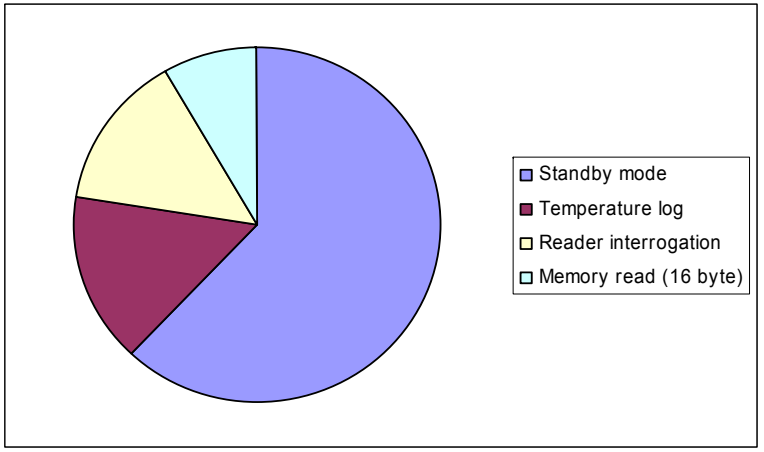


Sources: [Trummer-2006], [Janek-2007]

e.g. higher class RFID tag

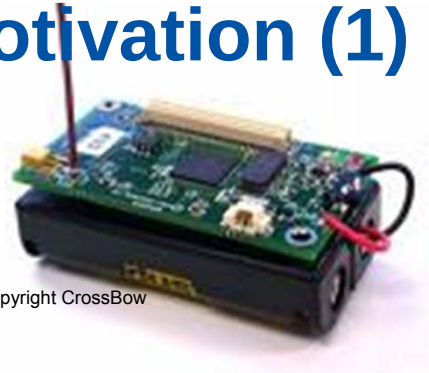
- Identec Solutions GmbH, i-Q tag
  - Measures temperature
  - 100-meter read/write range
  - Batteries last approx. 6 years
    - 600 x 16 byte read cycles / day
- Analysis of the working principle
- Evaluation of the energy dissipation
  - Standby mode: ~60%
  - Temperature logging: ~16%
  - Reader interrogation: ~14%
  - Memory Read (16 byte): ~10%
- Recording of the power profiles





Source: [Identec-2007]

Motivation (1)



Copyright CrossBow

Copyright Alien Technology

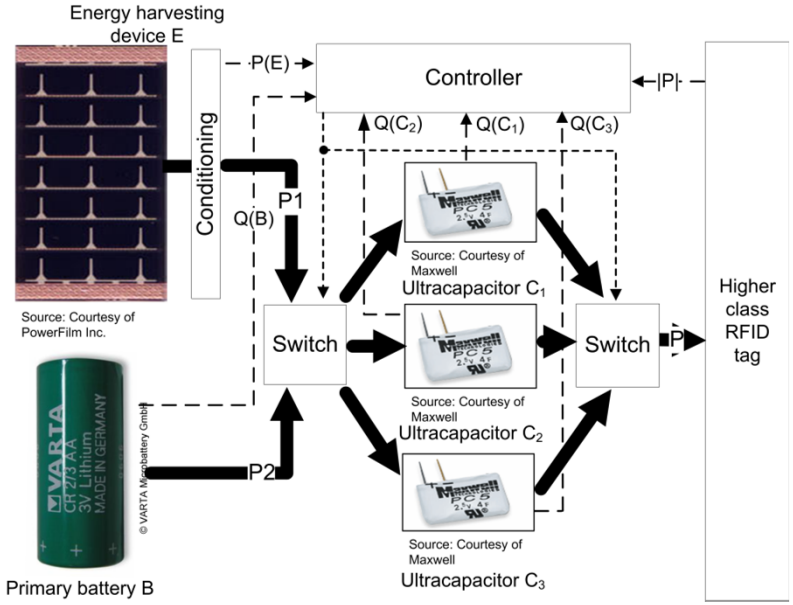
- Ubiquitous sensors
  - Elements
    - Higher class RFID tags[1]
    - Wireless Sensor Nodes
  - Goal: autonomous operation
    - State-of-the-art higher class RFID tag lifetime: 3-4 years
    - State-of-the-art Wireless Senor Node lifetime: < 1 year
  - Issue: limited lifetime

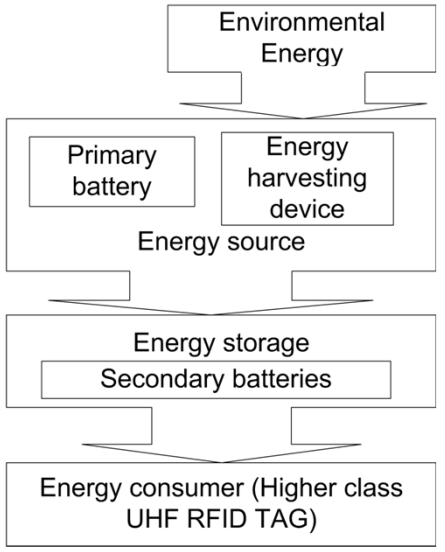
Copyright SAVI Technology

[1] RFID tag classification according EPCglobal Inc. Higher class RFID tag = RFID tag with sensors and energy source

Architecture design for Energy Harvesting Sensors

- Standby power reduction – energy harvesting
  - Integration: energy harvesting devices
  - Redesign: energy storage architecture
  - Benefit: doubled lifetime (non-optimized architecture) - 7,5 years vs. 4 years





Novel architecture design for Energy Harvesting Sensors: higher class RFID tag