Embedded Computing

Tajana Šimunić Rosing, UCSD, Charles R. Farrar,LANL

Source: Rabaey, Sasimi'04

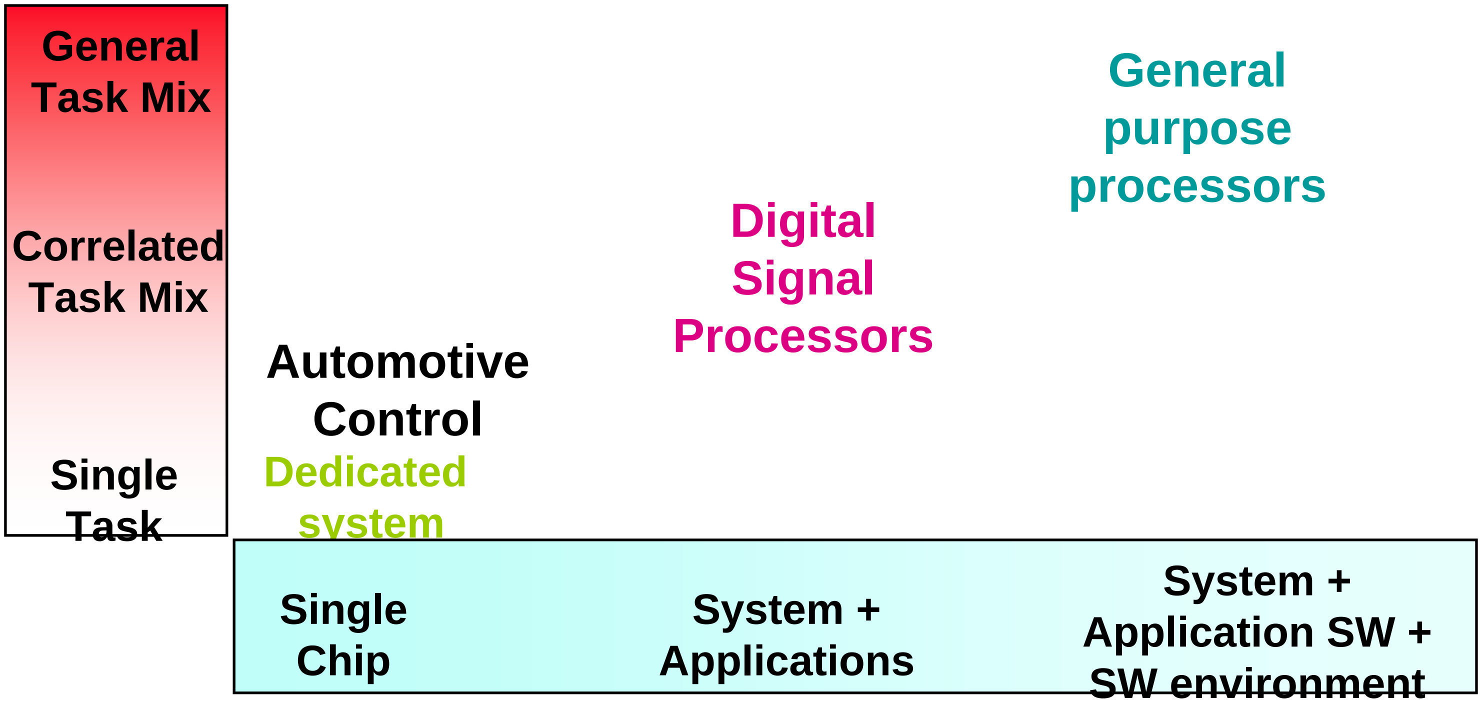
What is an embedded system?

- Systems which use computation to perform a *specific function*
- embedded within larger devices
- often completely unrecognized by the device's user
- Examples: home appliances, office automation, consumer electronics, smart buildings, sensor nets, wearable electronics, automobiles

Diversity in Embedded Computing

- Pocket remote control RF transmitter
  - 100 KIPS, crush-proof, long battery life
  - Software optimized for size
- Industrial equipment controller
  - 1 MIPS, safety-critical, 1 MB memory
  - Software control loops
- Military signal processing
  - 1 GFLOP, 1 GB/sec IO
  - Software for high performance.

System development



- Incremental SW programming
  - Start with a few procedures, debug, a few more, debug…
- HW development mostly by “prototyping”
  - in-circuit emulation
  - prototype hardware is debugged by application software

Diverse SW Development

- High-volume, low-cost, low-complexity: portable CD player
  - 8-bit microcontrollers with on-chip RAM, ROM, IO
  - hand-coded assembly, single task, 1K-8K bytes of code
- Medium-volume, moderate-cost, moderate-complexity: handheld remote terminal
  - 16-bit or 32-bit microcontrollers with memory and peripherals
  - Customizable SW developed as a mixture of C and assembly
  - 64 KB-1MB of code, more than one task
- Low-volume, high-cost, high-complexity: air-traffic control system
  - Security and reliability are primary measures of quality
  - Significant development and maintenance costs
  - very high performance
- Very high performance: wireless modem
  - HW enables design, SW must keep up with HW capabilities
  - Custom ASIP processor, complex development environment
  - Integration with hardware design tools is often desired
    - Co-simulation and co-synthesis tools

Trends in Embedded Systems

- Increasing code size
  - average code size: 16-64KB in 1992, 64K-512KB in 1996
  - migration from hand (assembly) coding to high-level languages
- Reuse of hardware and software components
  - processors (micro-controllers, DSPs)
  - software components (drivers)
- Increasing integration and system complexity
  - integration of RF, DSP, network interfaces
  - 32-bit processors, IO processors

Embedded system metrics

- Some metrics:
  - *performance*: MIPS, reads/sec etc.
  - *power*: Watts
  - *cost*: Dollars
  - Software and architecture:
    - Instruction set, code density, register organization, caches, addressing, data types etc.
- MIPS, Watts and cost are related
  - technology driven
  - to get more MIPS for fewer Watts
    - look at the sources of power consumption
    - use power management and voltage scaling

Example: PDA design

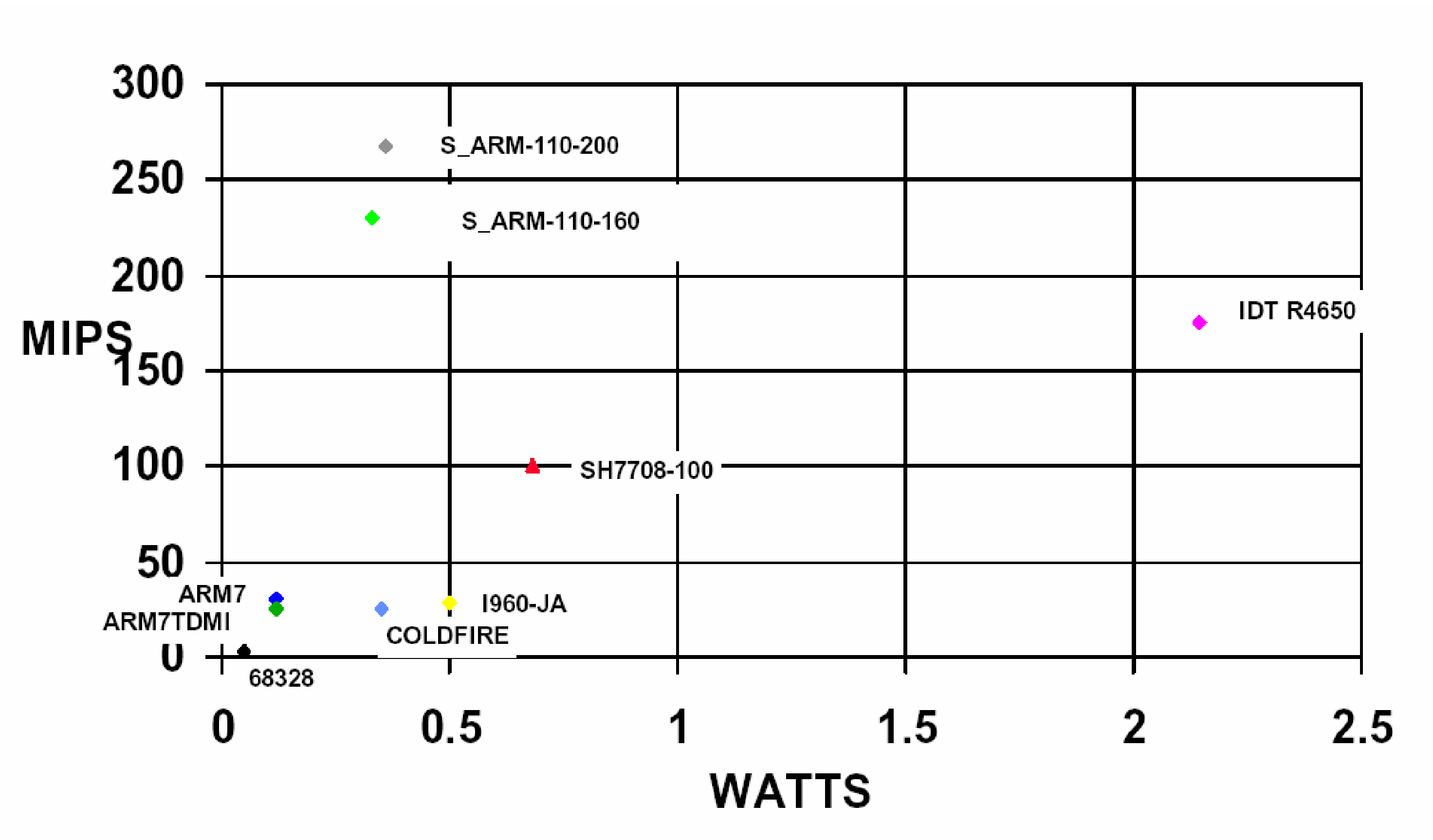
Why did they design it this way?

A 'Dragonball*' processor?

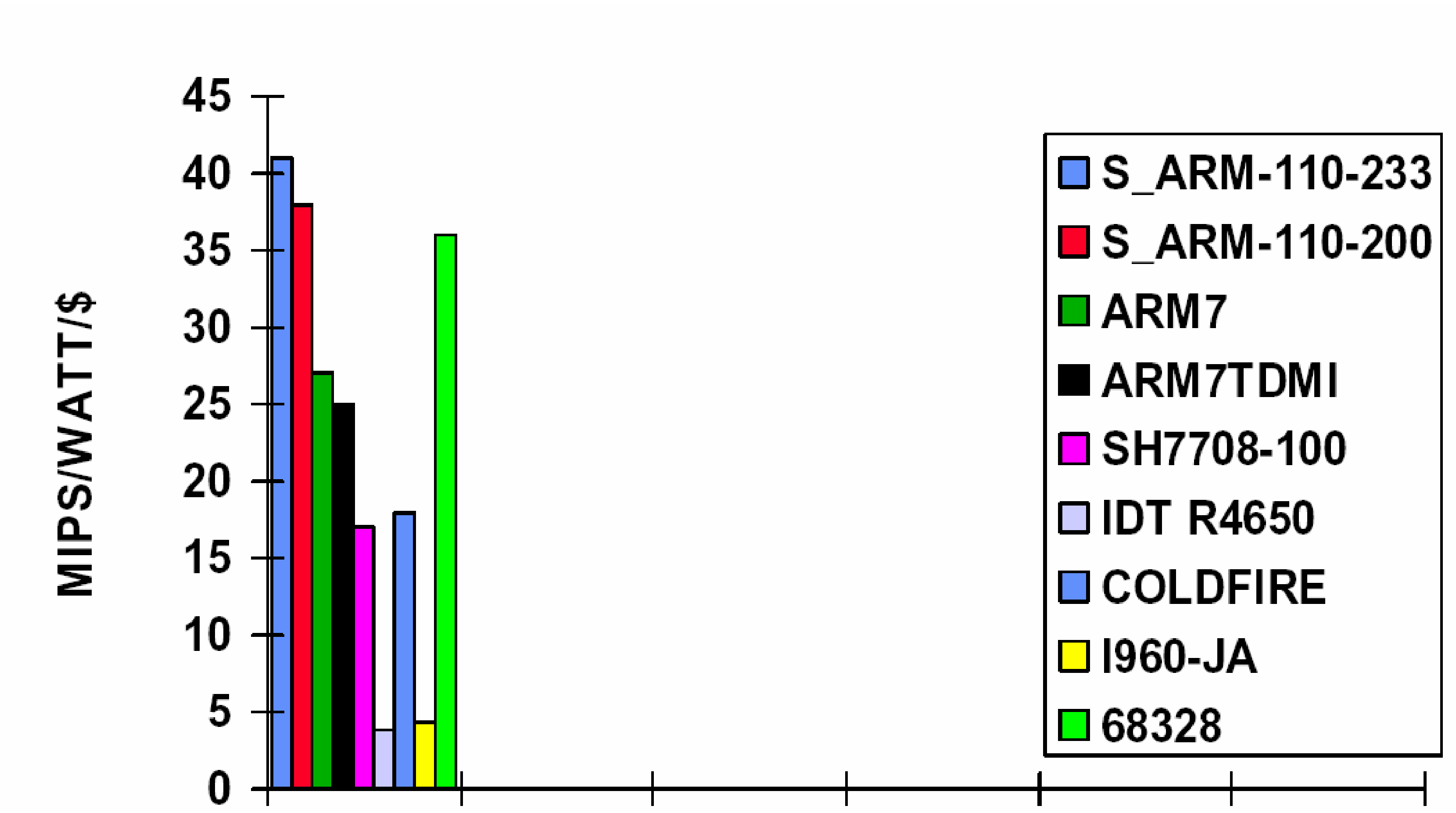
We all wanted StrongARMs

*The Dragonball used in the early Palm Pilots is a Motorola 68328

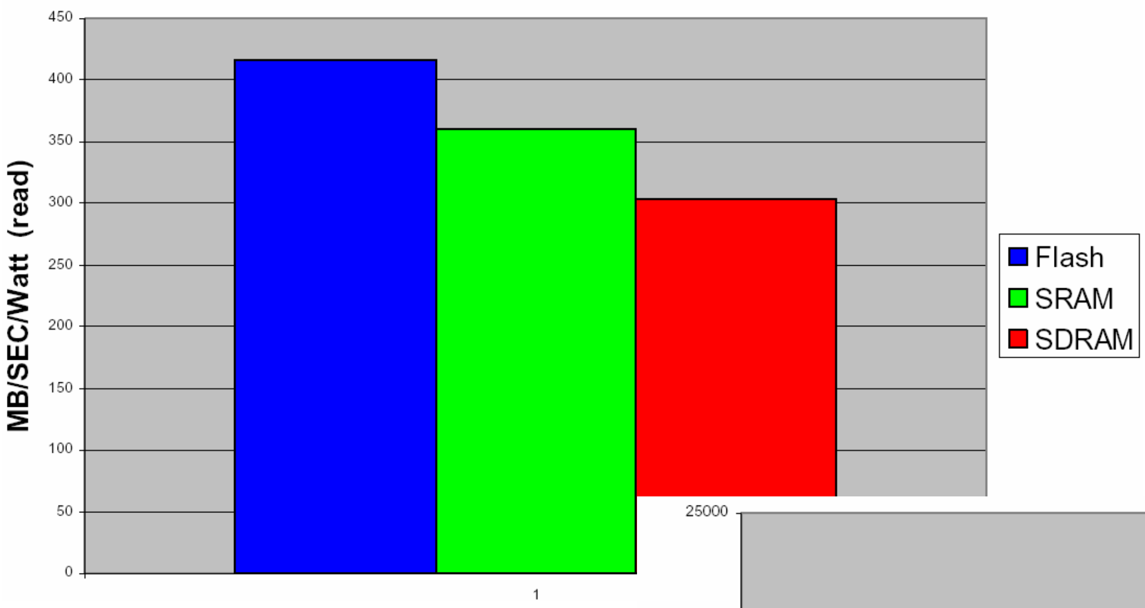
MIPS vs. Watts



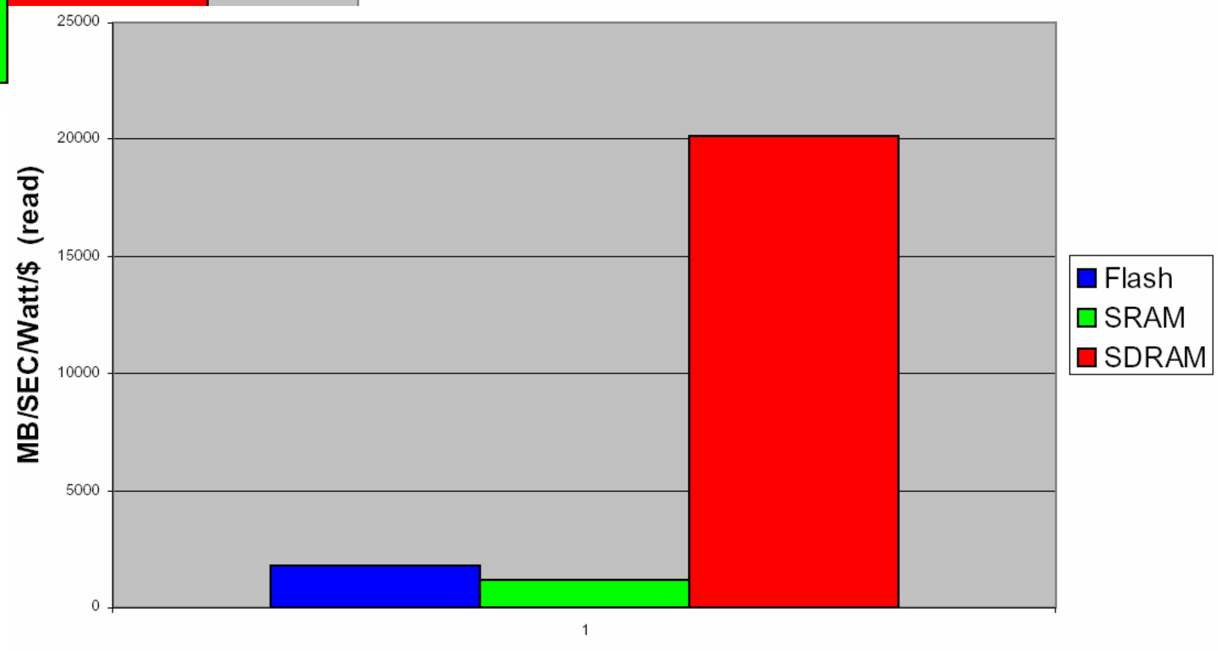
MIPS/W/$


MIPS/WATT/$
45
40
35
30
25
20
15
10
5
0
S_ARM-110-233
S_ARM-110-200
ARM7
ARM7TDMI
SH7708-100
IDT R4650
COLDFIRE
I960-JA
68328

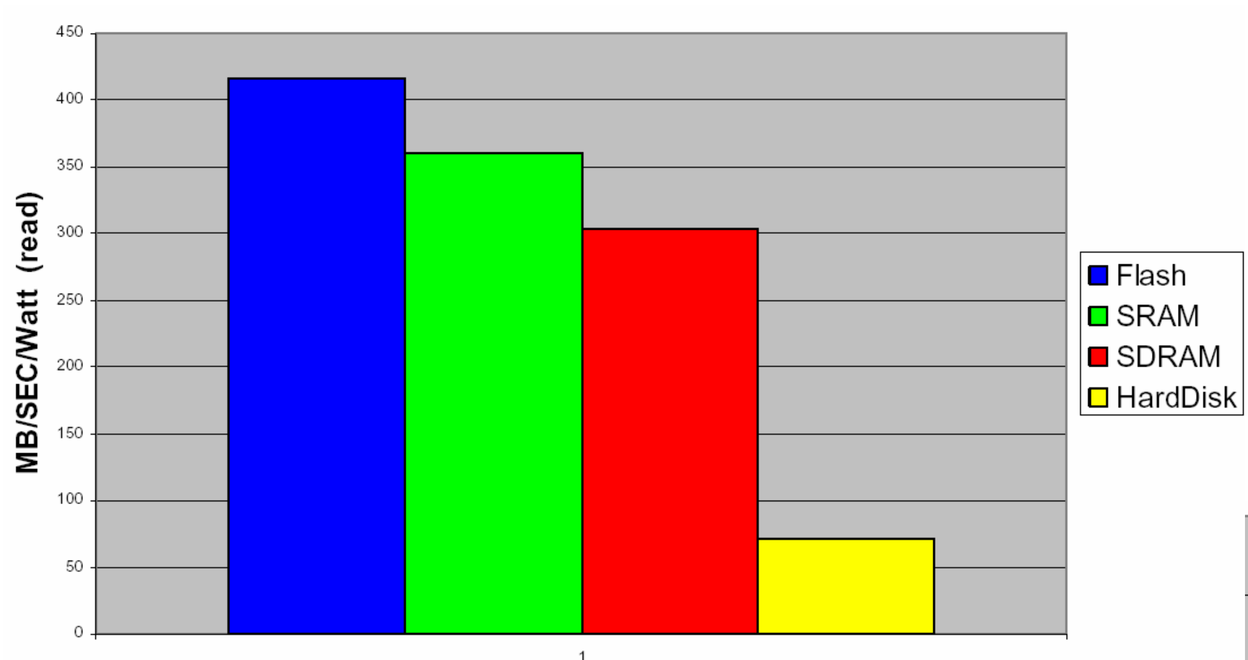
Bandwidth vs. Watt and $



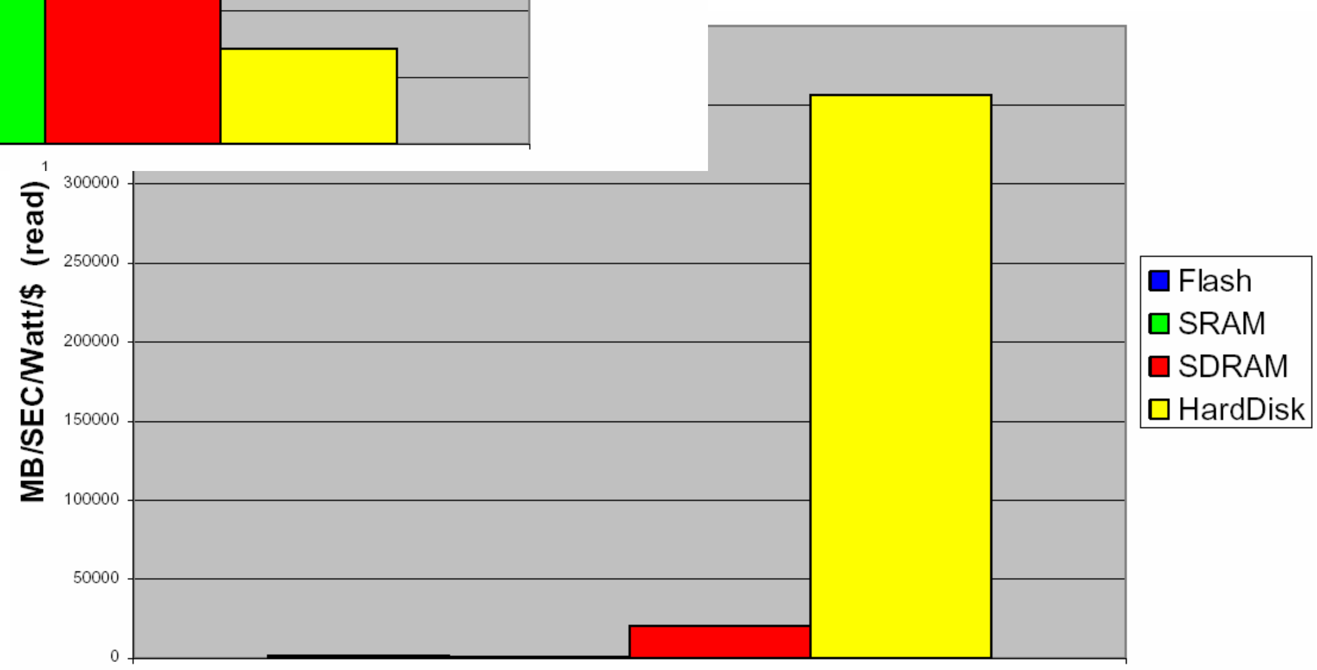
This is why PDAs use SDRAM



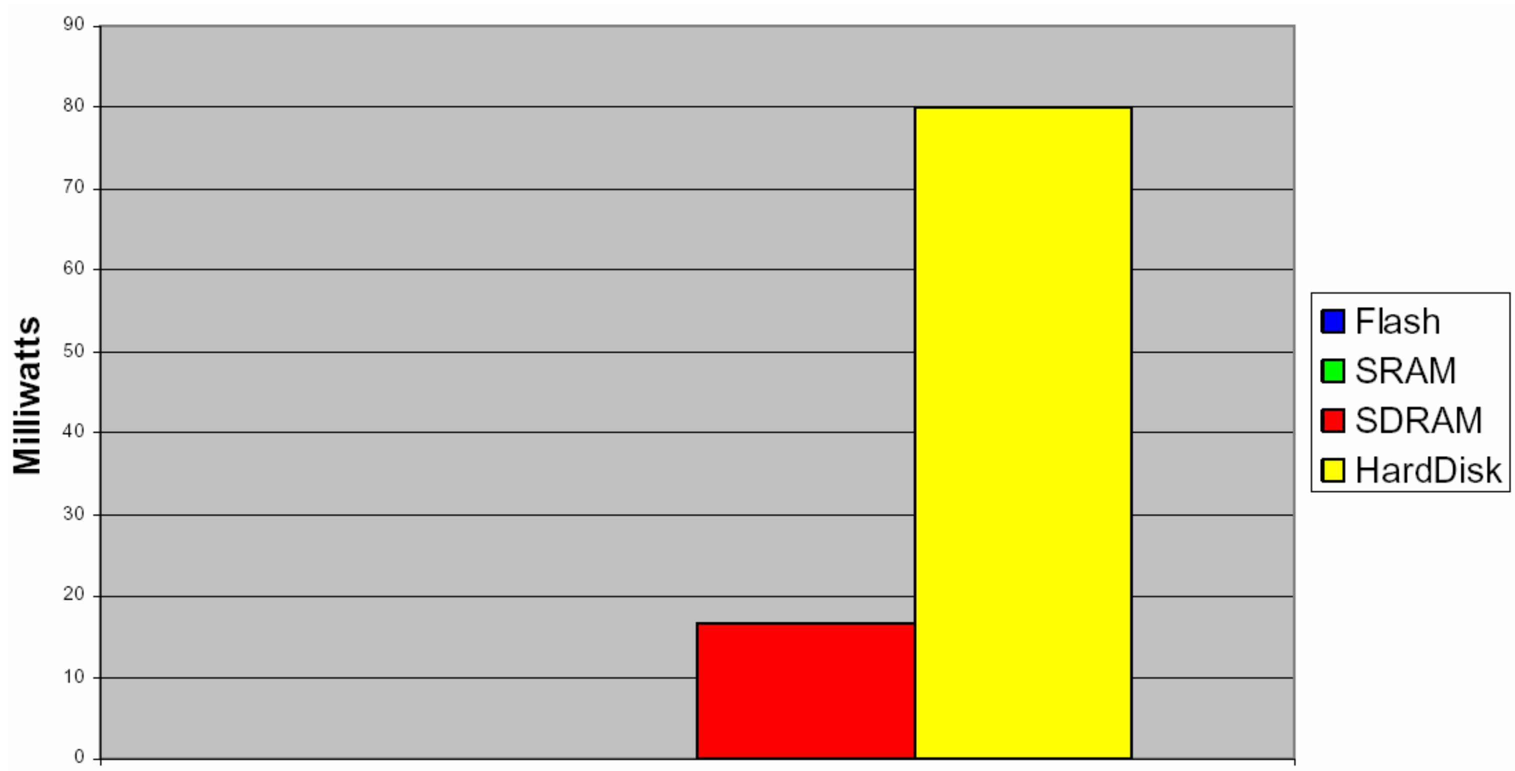
BW/W/$ with hard disk



Why IPOD used hard disk



Standby power



Here is why cell phone battery lasts longest, PDA shorter and IPOD only a few hours

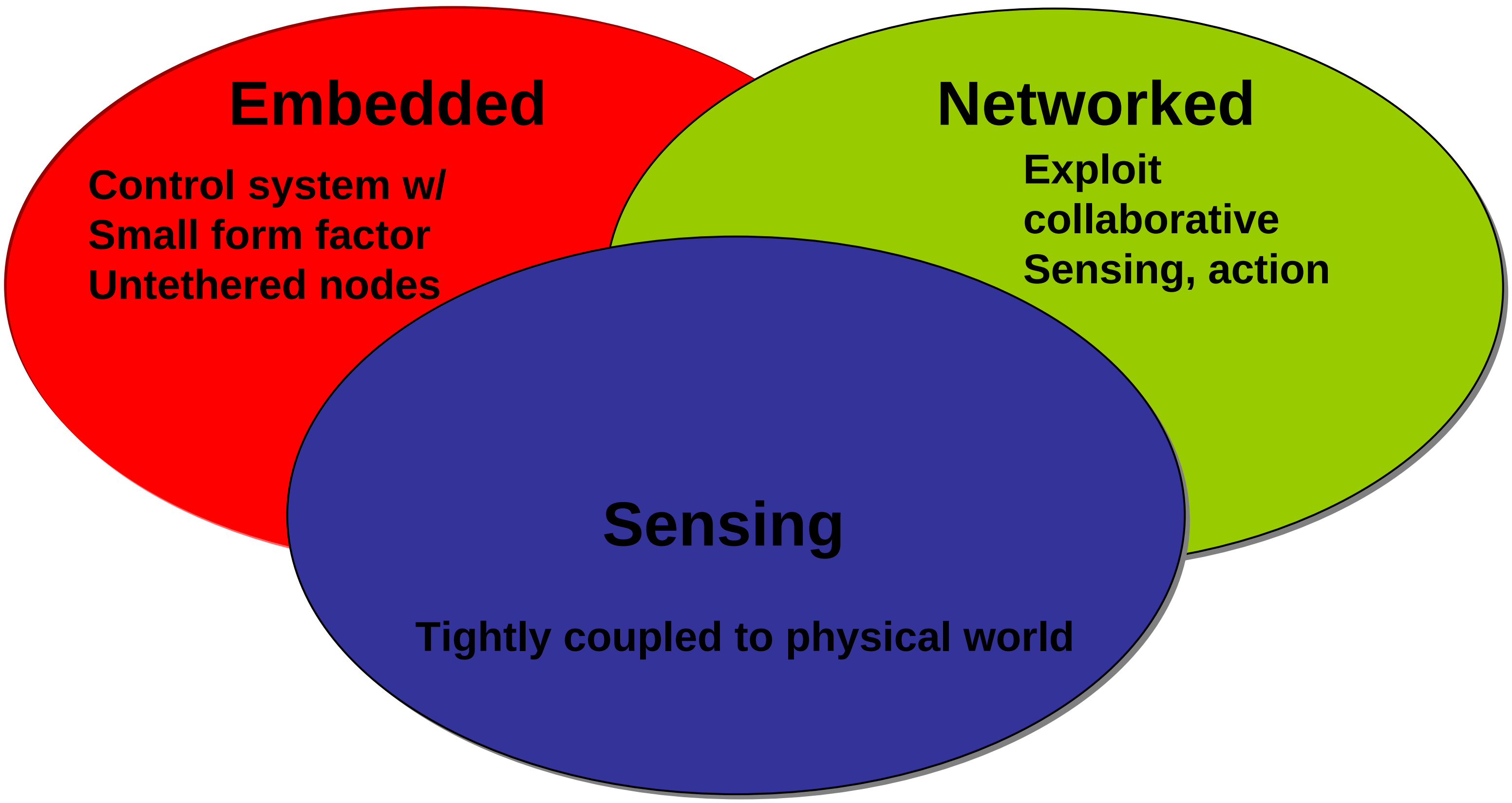
Enter Wireless: Computers + Radios

- Wireless
  - limited bandwidth, high latency (3ms-100ms)
  - variable link quality and link asymmetry due to noise, interference, disconnections
- Mobility
  - causes variability in system design parameters: connectivity, b/w, security domains, location awareness
- Portability
  - limited capacities (battery, CPU, I/O, storage, dimensions)

Enabling Technologies

Embed numerous distributed devices to monitor and interact with physical world

Network devices to coordinate and perform higher-level tasks

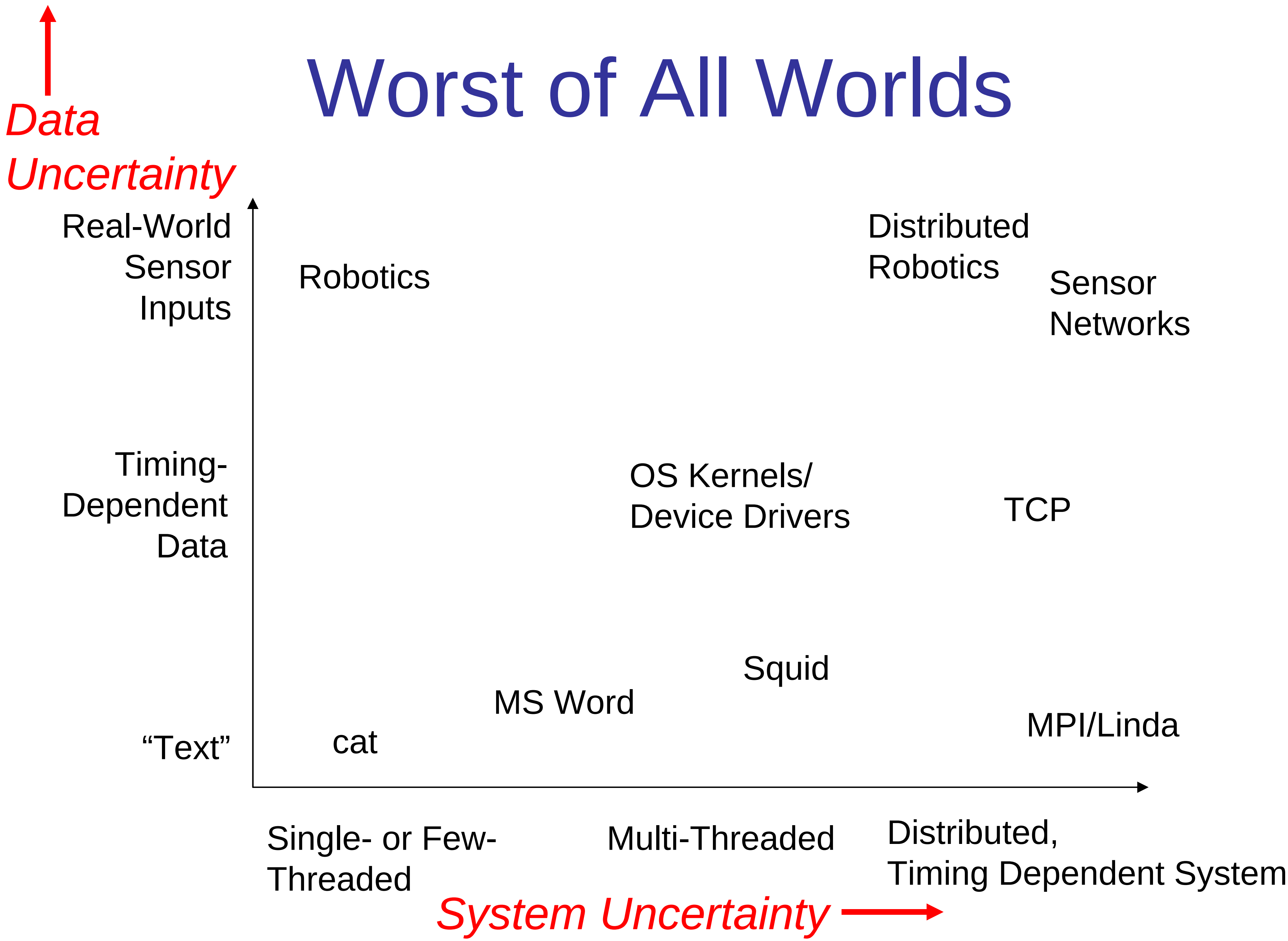


Exploit spatially and temporally dense, in situ, sensing and actuation

New Design Themes

- **Long-lived** systems that can be **untethered** and **unattended**
  - Low-duty cycle operation with bounded latency
  - Exploit redundancy and heterogeneous tiered systems
- **Leverage data processing inside the network**
  - Exploit computation near data to reduce communication
- **Self configuring** systems that can be deployed **ad hoc**
  - Measure and adapt to unpredictable environment
  - Exploit spatial diversity and density of sensor/actuator nodes
- Achieve desired global behavior with **adaptive localized algorithms**
  - Cannot afford to extract dynamic state information needed for centralized control

Worst of All Worlds

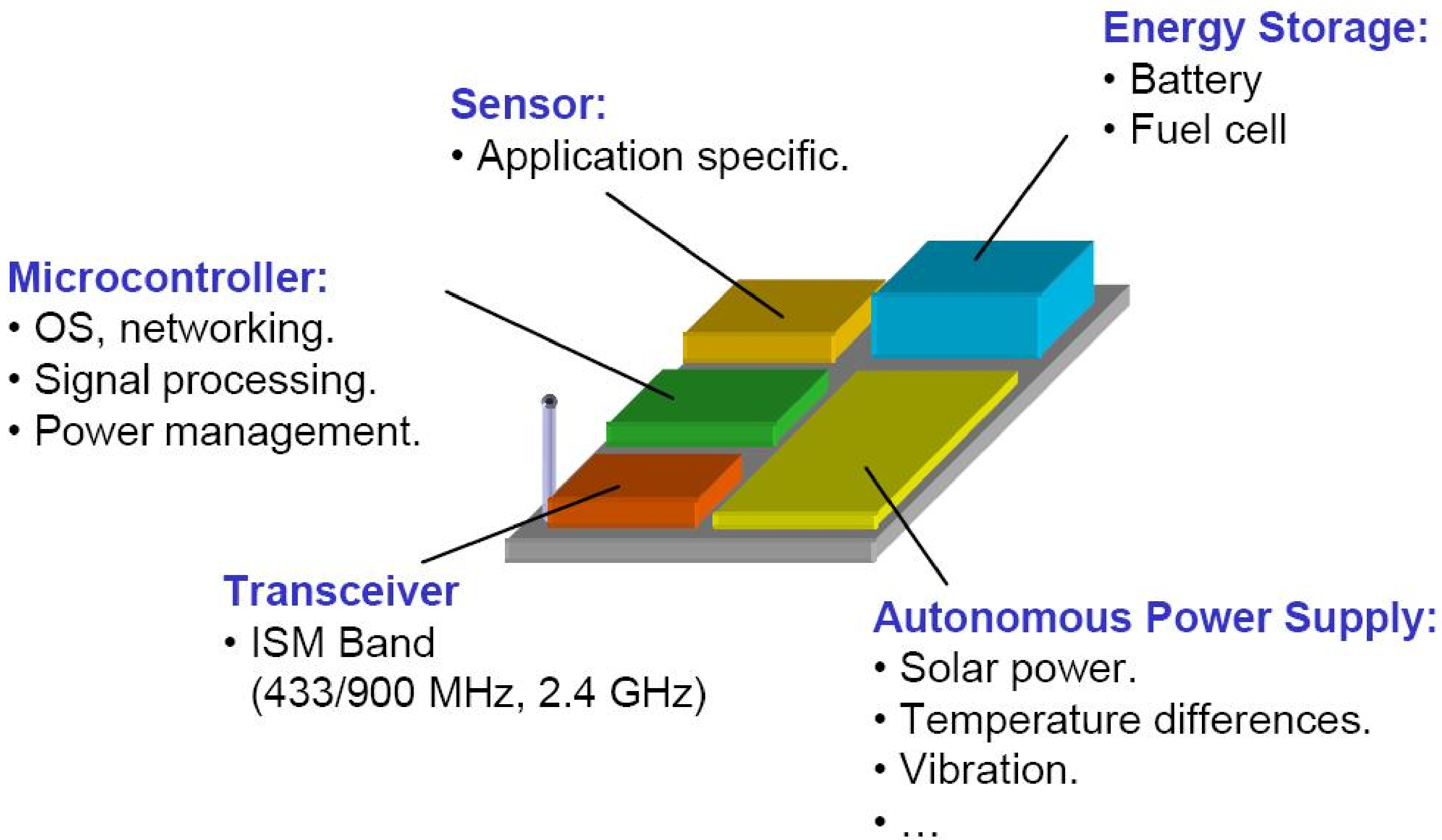


Jeremy Elson, Microsoft

Sensors

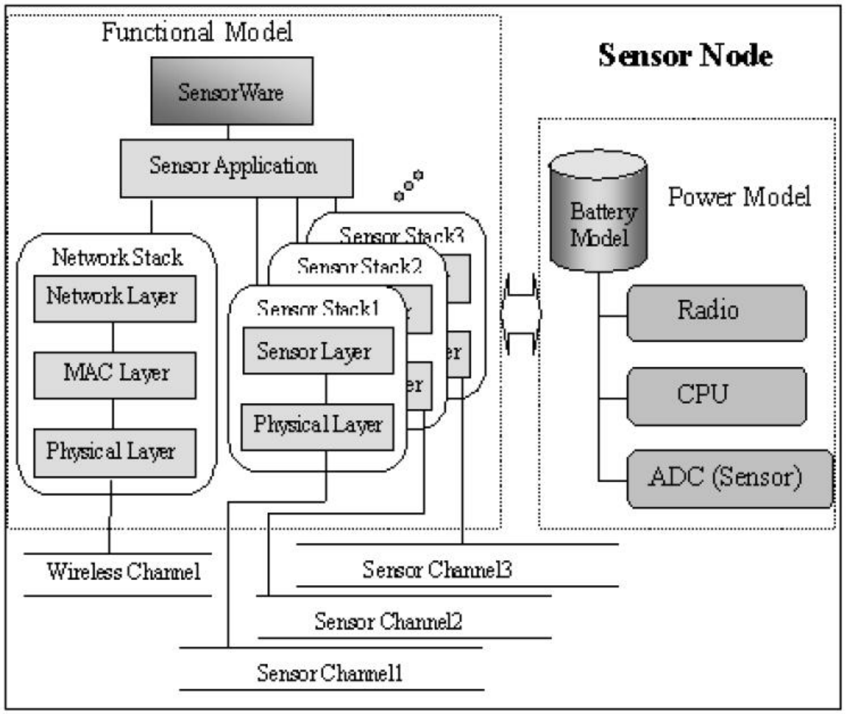
- Passive elements: seismic, acoustic, infrared, strain, salinity, humidity, temperature, etc.
- Passive Arrays: imagers (visible, IR), biochemical
- Active sensors: radar, sonar
  - High energy, in contrast to passive elements
- Technology trend:
  - use of IC technology for increased robustness, lower cost, smaller size
  - COTS adequate in many of these domains; work remains to be done in biochemical

Sensor Node Architecture



Source: Estrin, MOBICOM'03

Sensor Node Model



- Sensor can be divided into hardware and software
- Layering model can be used for the software
- Important to optimize both hardware and software – use simulation

Source: Estrin, MOBICOM'03

Generations of sensor nodes

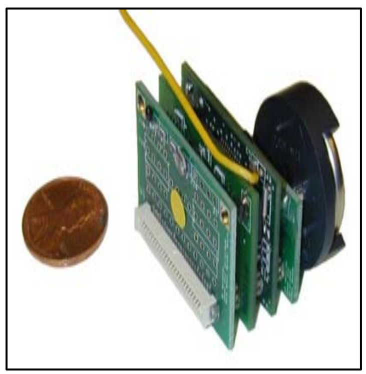
Characteristic	Generation1	Generation 2	Generation3
Timeline	1980-1990	2000-2003	2010
Size	Large shoe box or bigger	Pack of cards	Dust particle
Weight	Kilograms	Grams	Negligible
Node Architecture	Separate sensing, processing and communication	Integrated sensing, processing and communication	Integrated sensing, processing and communication
Topology	Pont to point, star	Client server, peer to peer	Peer to peer
Power supply lifetime	Large batteries, hours to days	AA batteries, days to weeks	Solar, months to years
Deployment	Vehicle placed or air drop single sensors	Hand placed	Embedded, “sprinkled’, left behind

Some Networked Sensor Nodes

LWIM III
UCLA, 1996
Geophone, RFM radio, PIC, star network

AWAIRS I
UCLA/RSC 1998
Geophone, DS/SS Radio, StrongARM, Multi-hop networks

UCB Mote, 2000
4 Mhz, 4K Ram
512K EEProm,
128K code,
CSMA
half-duplex RFM radio



WINS NG 2.0
Sensoria, 2001
Node development platform; multi-sensor, dual radio, Linux on SH4, Preprocessor, GPS

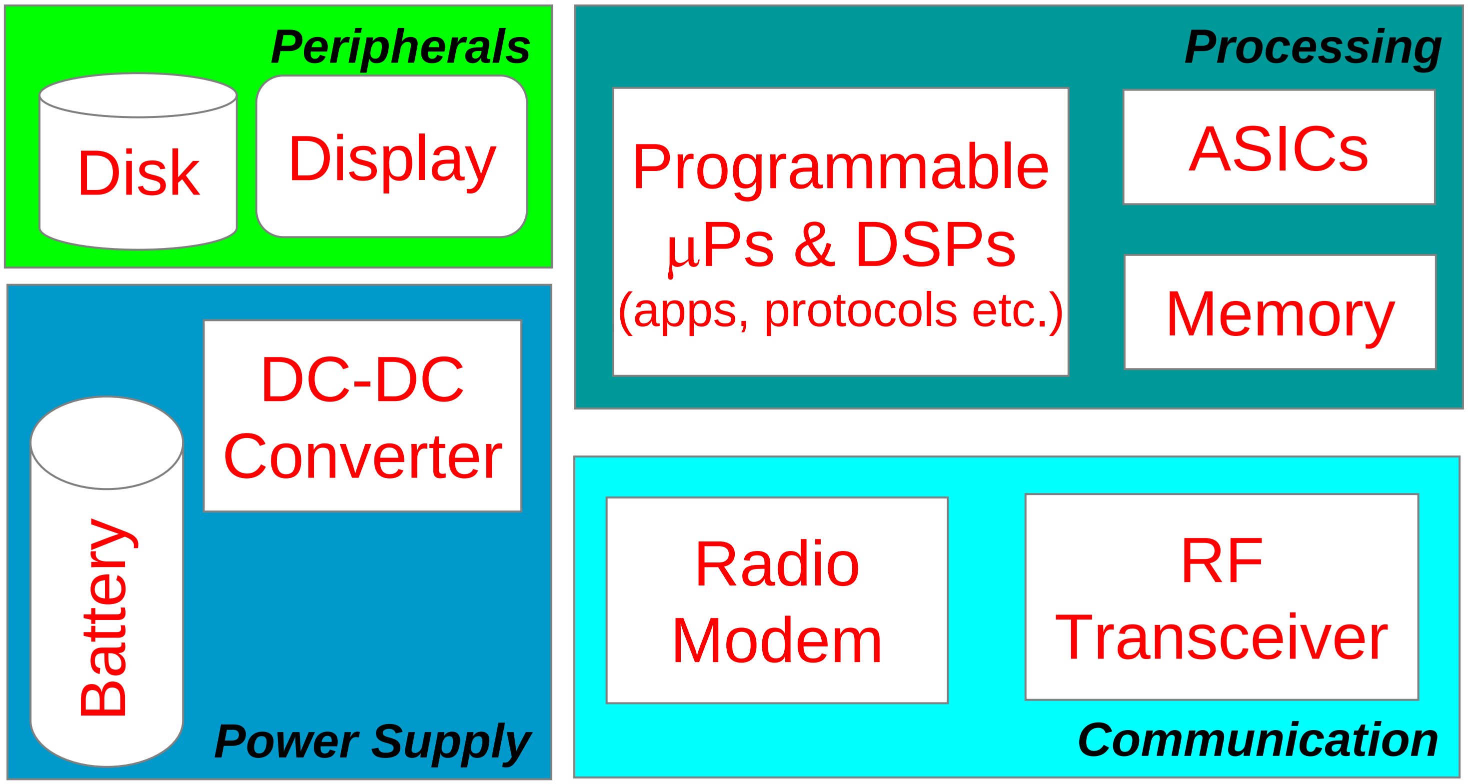
Processor

Source: Estrin, MOBICOM'02

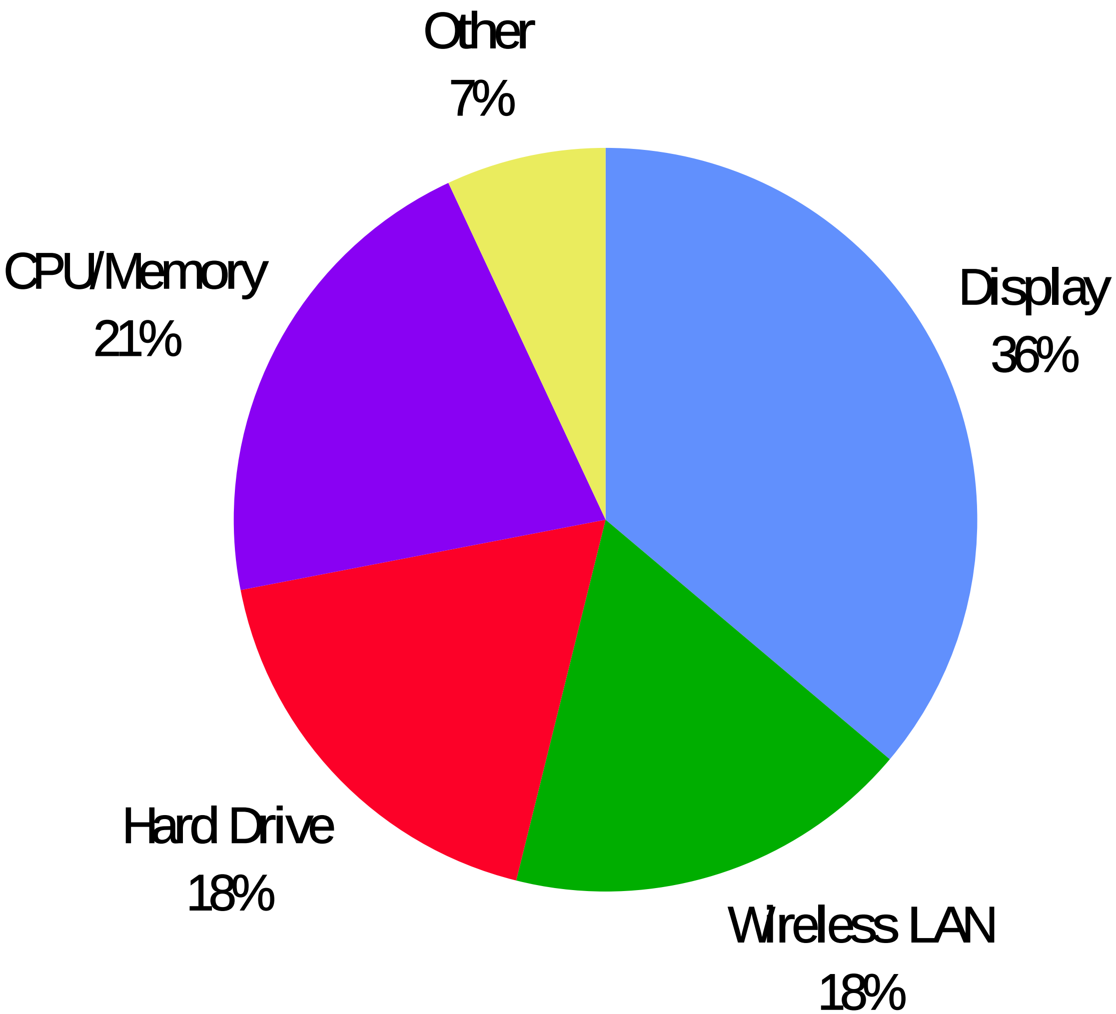
Micro sensor nodes

- Berkeley/Crossbow MICA2 Mote
  - $250 (MICA2 + sensor board)
  - www.xbow.com
  - ATMEGA microcontroller
    - 4 KB RAM, 4 KB EEPROM, 128 KB Flash
    - CPU 8 MHZ
  - Chipcon CC1000 radio, <20 kbps
    - Multifrequency
  - Two AA Batteries
    - 1% Duty Cycle can extend lifetime from 5 days to 3-6 months
  - Extension boards for sensors

Where does the Power Go?



Laptop Power Consumption



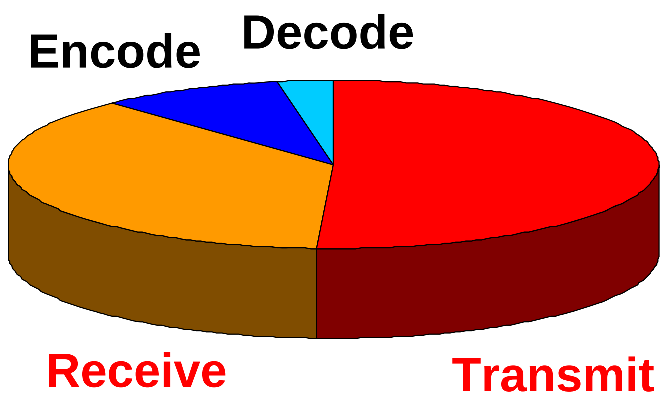
Power Consumption of Itsy

Itsy v1
StrongARM 1100
59–206 MHz (300 us to switch)
2 core voltages (1.5V, 1.23V)
64M DRAM / 32M FLASH
Touchscreen & 320x200 LCD
codec, microphone & speaker
serial, IrDA

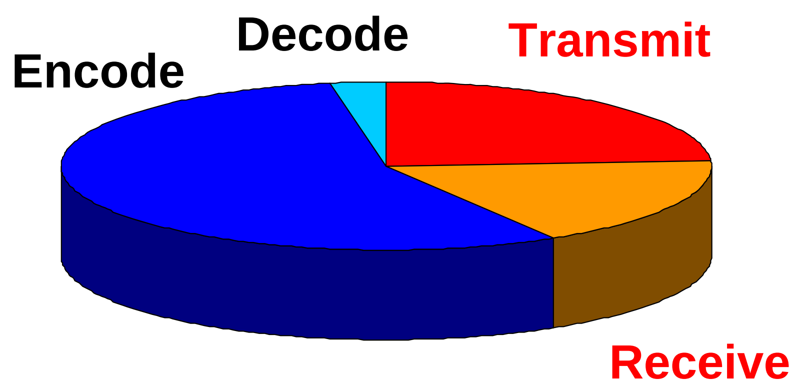
- System power < 1W
  - doing nothing (processor 95% idle)
    - 107 mW @ 206 MHz
    - 77 mW @ 59 MHz
    - 62 mW @ 59 MHz, low voltage
  - MPEG-1 with audio
    - 850 mW @ 206 MHz (16% idle)
  - Dictation
    - 775 mW @ 206 MHz (< 0.5% idle)
  - text-to-speech
    - 420 mW @ 206 MHz (53% idle)
    - 365 mW @ 74 MHz, low voltage (< 0.5% idle)
- Processor: 200 mW
  - 42-50% of typical total
- LCD: 30-38 mW
  - 15% of typical total
    - 30-40% in notebooks

Networked Embedded Systems & Sensor Nodes

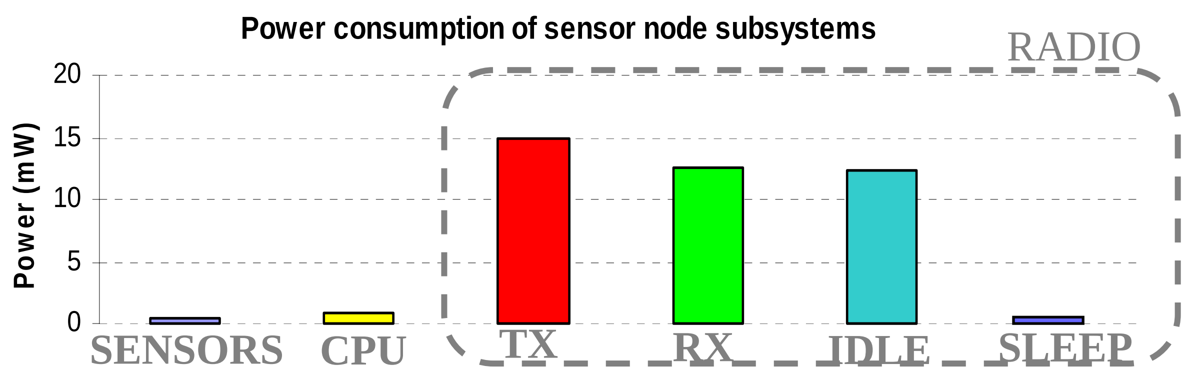
Energy breakdown for voice



Energy breakdown for MPEG video



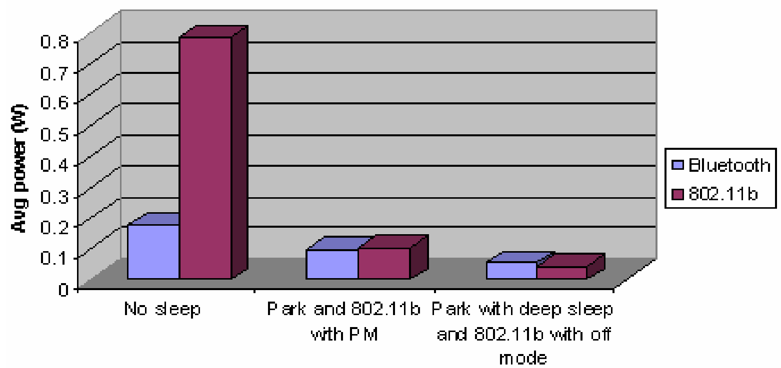
Lucent WaveLAN at 2 Mbps & SA-1100 CPU at 150 MPIS



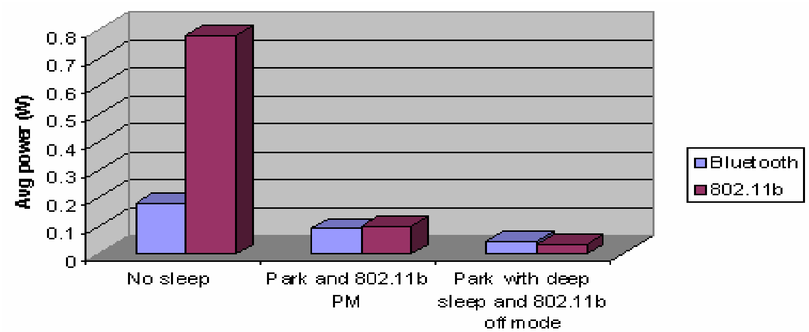
Source : Mobicom'01 SensorsTutorial

WLAN vs. Bluetooth

- Mp3 audio: MPEG4 video:

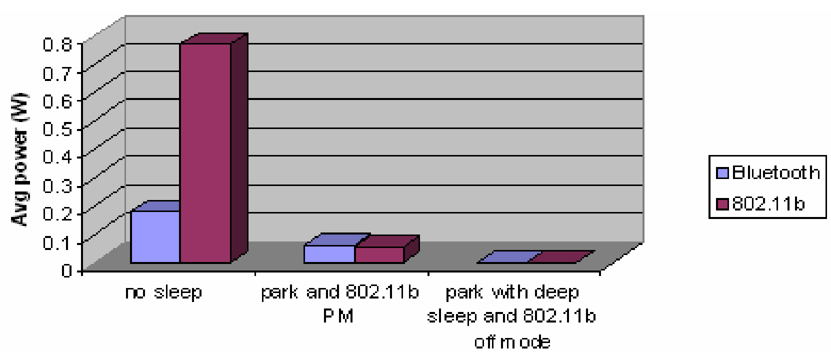


Average power consumption in Bluetooth Vs. 802.11b



Average power consumption in Bluetooth Vs. 802.11b

- For EMAIL



Average power consumption in Bluetooth Vs. 802.11b

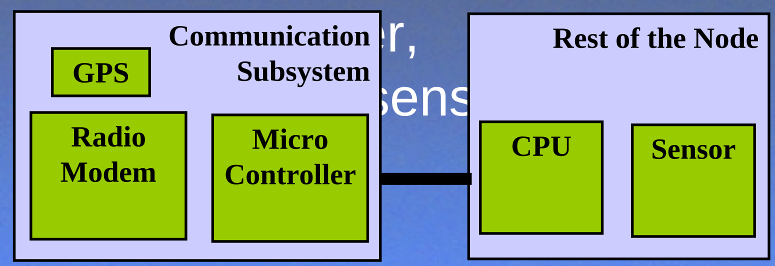
Power Measurements on WINS Node

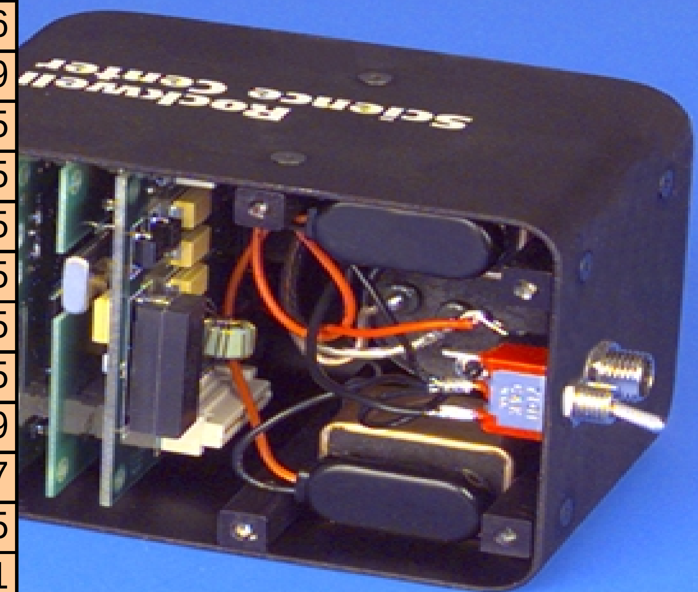
Processor	Seismic Sensor	Radio	Power (mW)
Active	On	Rx	751.6
Active	On	Idle	727.5
Active	On	Sleep	416.3
Active	On	Removed	383.3
Active	Removed	Removed	360.0
Active	On	Tx (36.3 mW)	1080.5
		Tx (27.5 mW)	1033.3
		Tx (19.1 mW)	986.0
		Tx (13.8 mW)	942.6
		Tx (10.0 mW)	910.9
		Tx (3.47 mW)	815.5
		Tx (2.51 mW)	807.5
		Tx (1.78 mW)	799.5
		Tx (1.32 mW)	791.5
		Tx (0.955 mW)	787.5
		Tx (0.437 mW)	775.5
		Tx (0.302 mW)	773.9
		Tx (0.229 mW)	772.7
		Tx (0.158 mW)	771.5
		Tx (0.117 mW)	771.1

Summary

- Processor = 360 mW
  - doing repeated transmit/receive
- Sensor = 23 mW
- Processor : Tx = 1 : 2
- Processor : Rx = 1 : 1
- Total Tx : Rx = 4 : 3 at maximum range

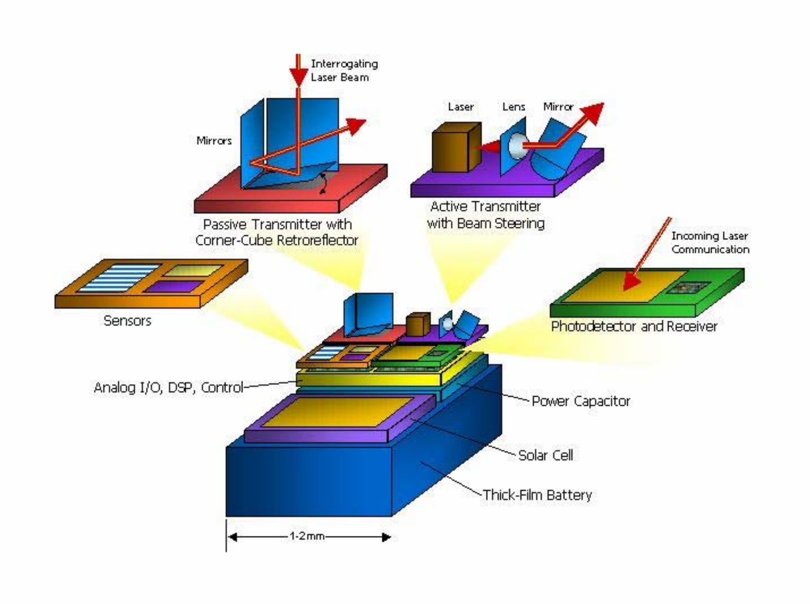
Capabilities: vibration, acoustic, accelerometer,



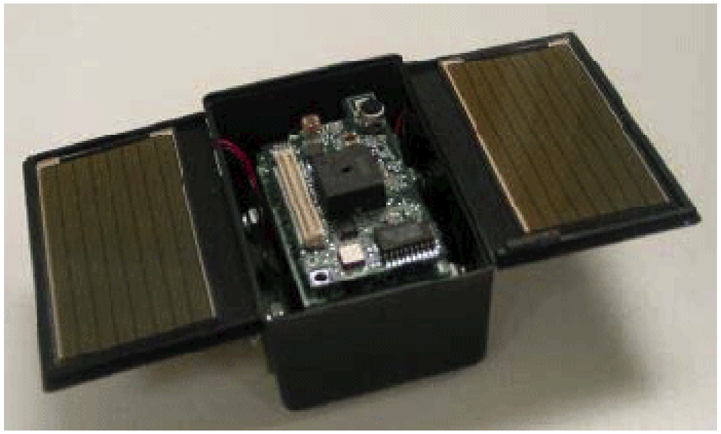


Solar Cell Sensor Nodes

- Smart dust (UCB)



Heliomote (UCLA)



- Zebra net
  - 5 day battery lifetime with no solar recharge (lithium-ion due to weight)
  - weight 1,4 to 2,3 kg
  - short (100m) and long range radios (8km)

System Design for Low Power

- Energy efficiency (has to) cut across all system layers
  - circuit, logic, software, protocols, algorithms, user interface, power supply...
  - Computation versus Communication; Node versus network
- Trade-off between energy consumption & QoS
  - optimize energy metric while meeting “quality” constraint

➲ When all low power tricks have been done, “duty cycling” remains the only available variable to reduce energy consumption
  - Must capture the “application intent”. Must be adaptive.
  - Pervasive power/energy awareness.

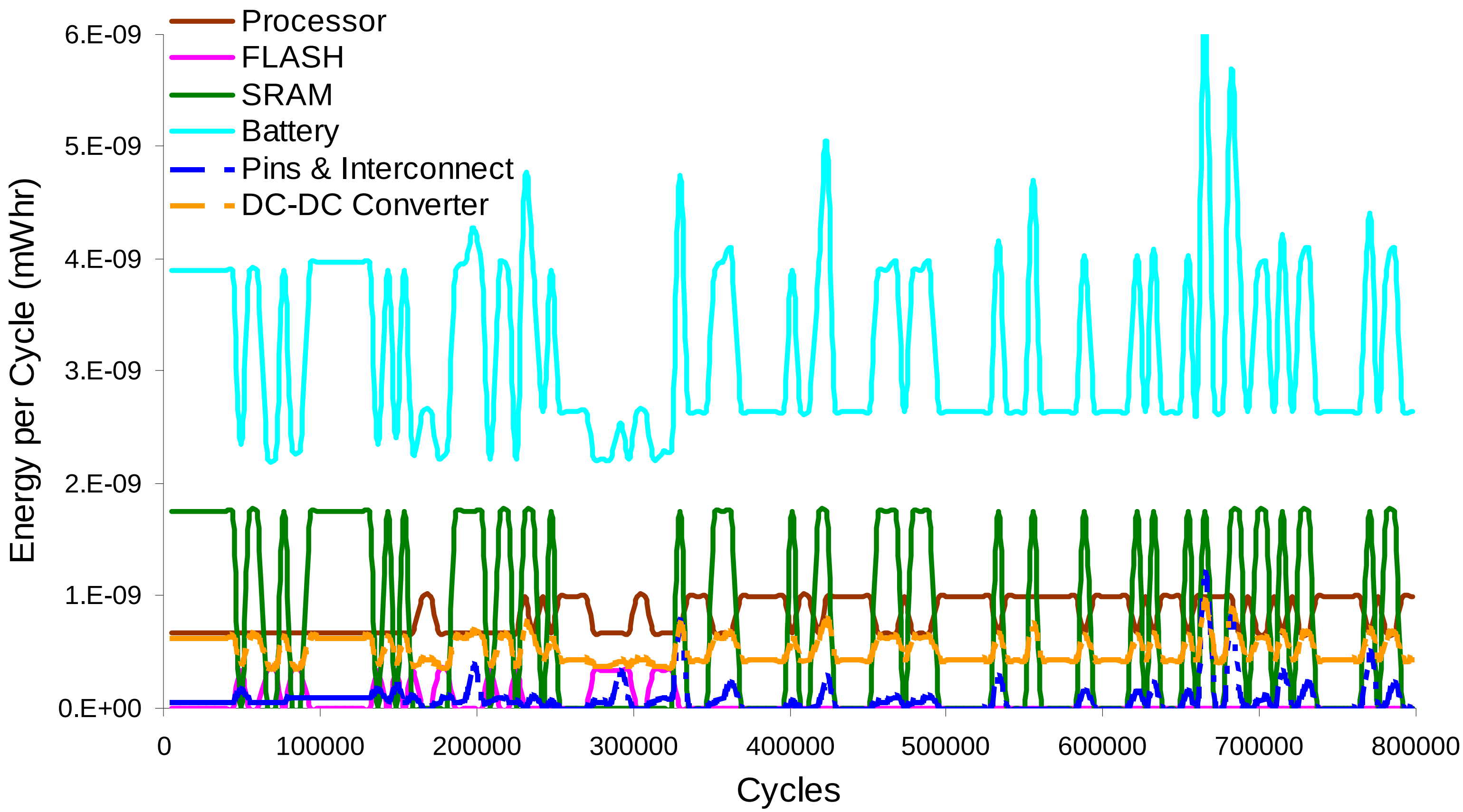
Energy Profiling

- Use cycle-accurate simulator and compiler get energy consumption
- Identifies critical procedures
  - Battery energy drain
  - By component analysis
- Fine grain analysis using detailed plots

Sample Profiler Output

Name	Cumulative (mWhr)	Self (mWhr)
main	3.20E-01	2.52E-02
...		
III_hybrid		6.71E-02
SubBandSynthesis		3.72E-02
III_stereo		2.75E-02
III_reorder		2.02E-02
III_antialias		1.45E-02
III_dequantize_sample		1.40E-02
III_hufman_decode		3.74E-03
III_get_scale_factor		1.28E-04
decode_info		3.20E-05
...		
III_hybrid	6.71E-02	6.36E-03
inv_mdctL		6.07E-02
SubBandSynthesis	3.72E-02	1.95E-02
chendct32_scaled		1.77E-02
III_stereo	2.75E-02	2.75E-02
III_reorder	2.02E-02	2.02E-02
III_antialias	1.45E-02	1.45E-02
III_dequantize_sample	1.40E-02	1.40E-02
III_hufman_decode	3.74E-03	1.53E-03
huffman_decoder		2.17E-03
initialize_huffman		1.03E-05
hsstell		3.20E-05

Energy consumption study



Software design example

Energy consumption by function

MP3 Code Rev.	1st	2nd	3rd
Original code	Floating Pt. 80.31%	SubBandSynthesis 10.31%	III_stereo 1.43%
Algorithmic Opts.	Floating Pt. 62.73%	III_stereo 6.12%	III_reorder 5.62%
Data & Instruction	SubBandSynthesis 34.32%	inv_mdctL 18.22%	III_stereo 7.32%
Combined Opts.	inv_mdctL 18.98%	III_stereo 8.61%	main 7.87%

Precision vs. Compliance

Precision # bits	Compliance
15	None
20	Partial
27	Full

Energy consumption HW components

MP3 Code Revision	Battery (mWhr)	CPU (mWhr)	Flash (mWhr)	RAM (mWhr)	DC-DC (mWhr)	Lines (mWhr)
Original code	0.446 0%	0.089 0%	0.005 0%	0.178 0%	0.045 0%	0.129 0%
Algorithmic Opts.	0.107 76%	0.020 77%	0.007 -44%	0.040 77%	0.011 76%	0.029 77%
Data & Instruction	0.130 71%	0.025 71%	0.004 27%	0.051 71%	0.013 71%	0.037 71%
Combined Opts.	0.105 77%	0.019 78%	0.007 -41%	0.040 78%	0.010 77%	0.028 78%

Performance

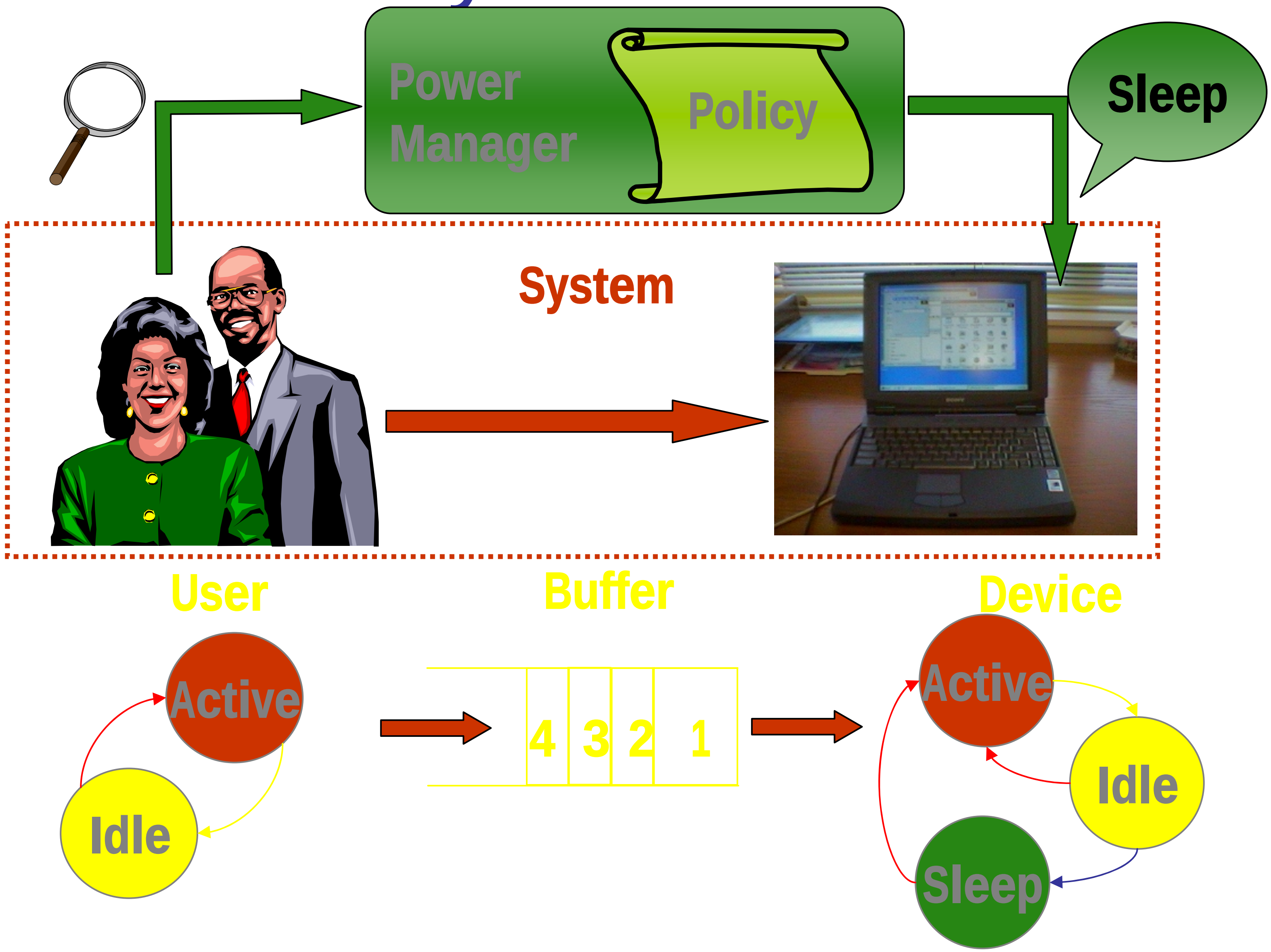
MP3 Code Revision	System (s)	Flash (s)	RAM (s)
Original code	68.490 0%	0.396 0%	6.309 0%
Algorithmic Opts.	34.562 50%	0.746 -88%	2.776 56%
Data & Instruction	9.185 87%	0.381 4%	4.186 34%
Combined Opts.	5.193 92%	0.718 -81%	2.093 67%

Energy savings of 77% and performance increase of 92%

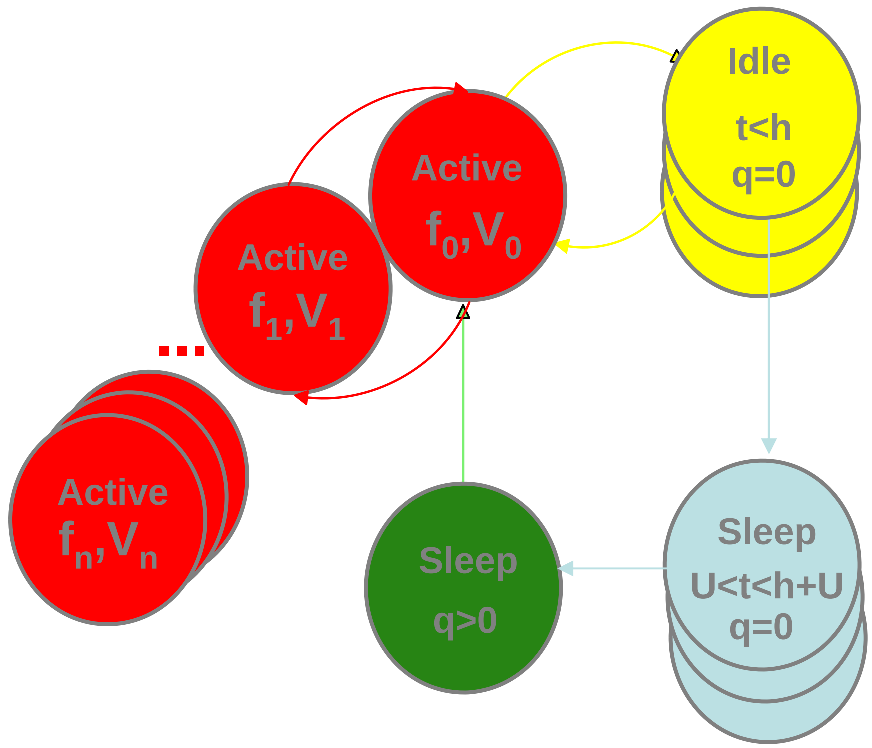
Saving Power at the System Level

- DPM (Dynamic Power Management)
  - "shutdown" through choice of right system & device states
    - Multiple sleep states
- DVS (Dynamic Voltage/Frequency Scaling)
  - "slowdown" through choice of right system & device states
    - Multiple active states
- DPM + DVS
  - Choice between amount of slowdown and shutdown

DPM System Model

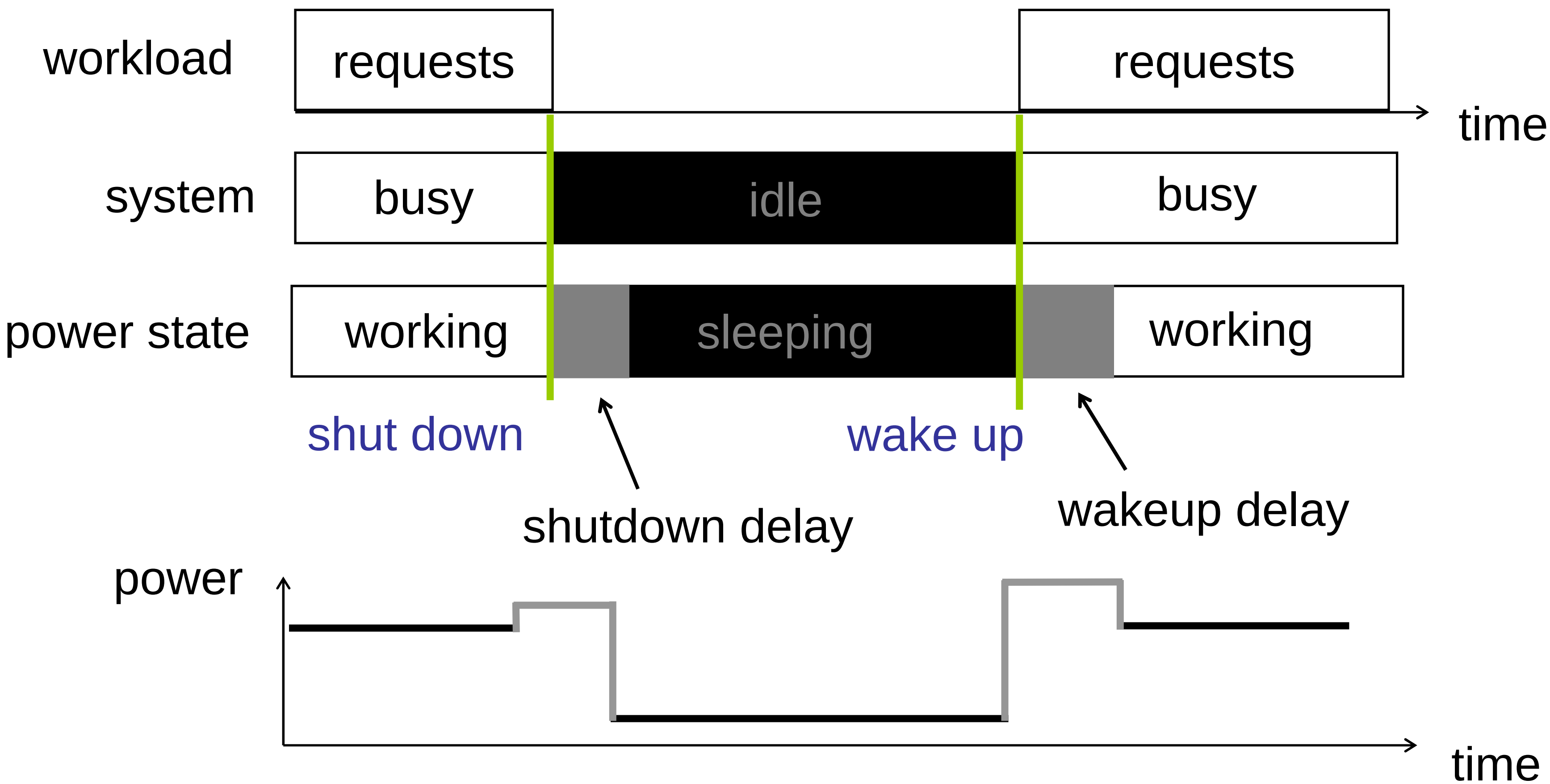


DPM with DVS Model

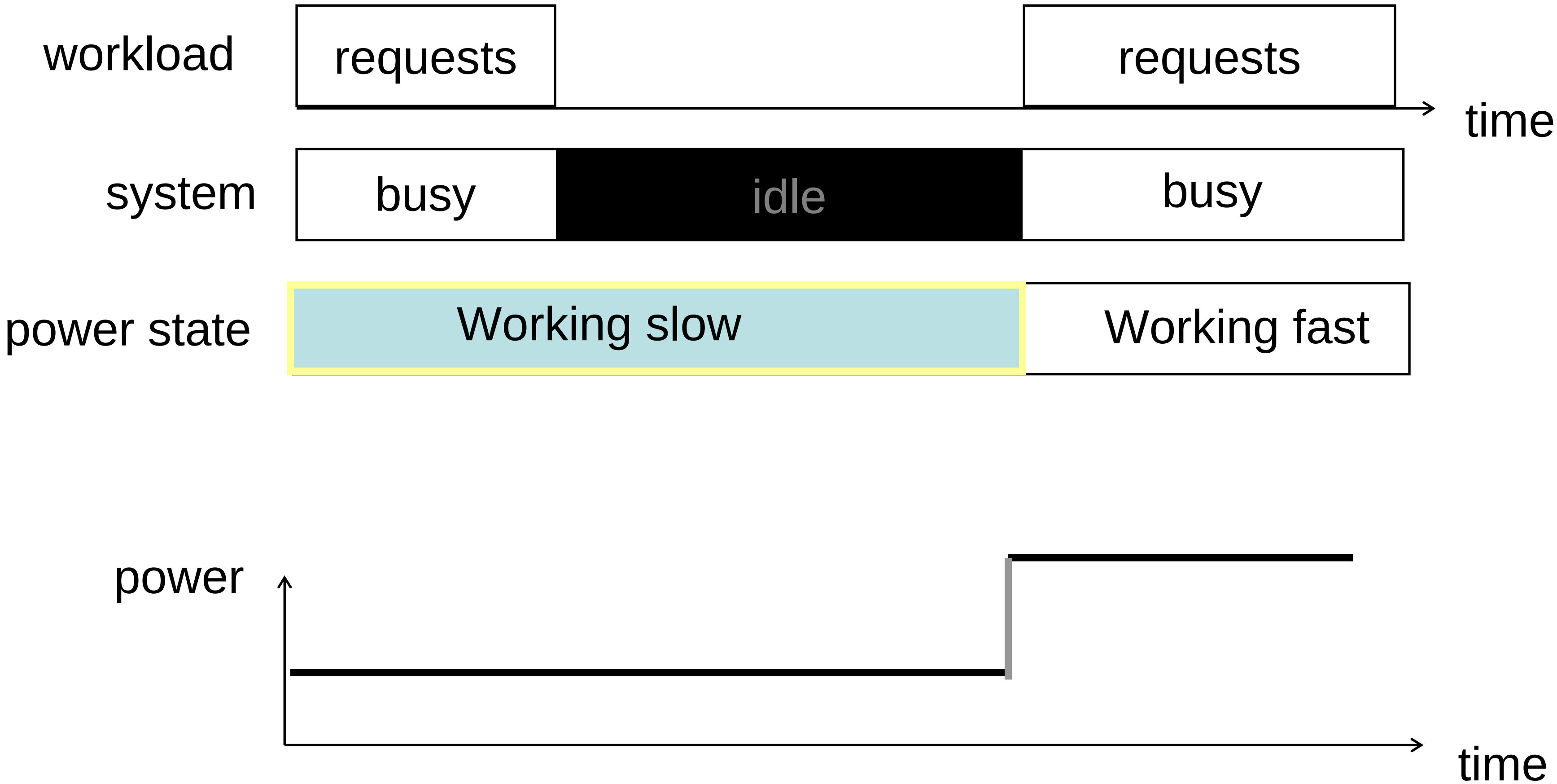


Dynamic Voltage Scaling with Dynamic Power Management

DPM example


workload
requests
requests
time
system
busy
idle
busy
power state
working
sleeping
working
shut down
wake up
shutdown delay
wakeup delay
power
time

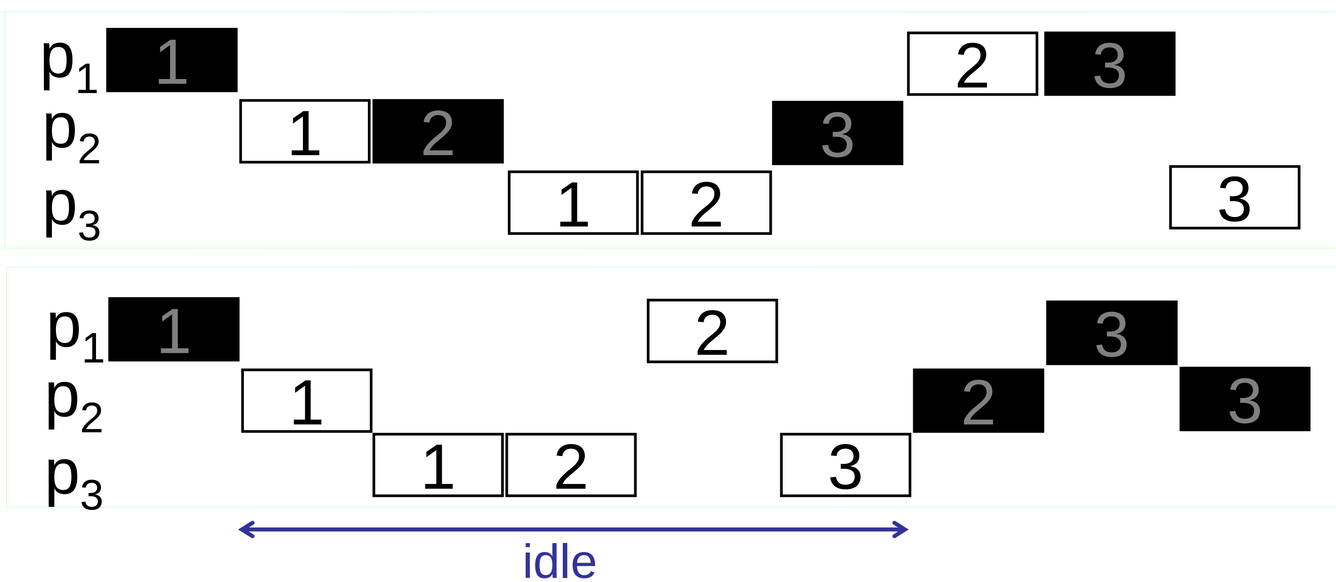
DVS example


workload
requests
requests
time
system
busy
idle
busy
power state
Working slow
Working fast
power
time

Low-Power Task Scheduling

tasks specify
- device requirements
- timing requirements

operating system
1. groups tasks with same device requirements
2. execute tasks in groups
3. wake up devices in advance to meet timing constraints



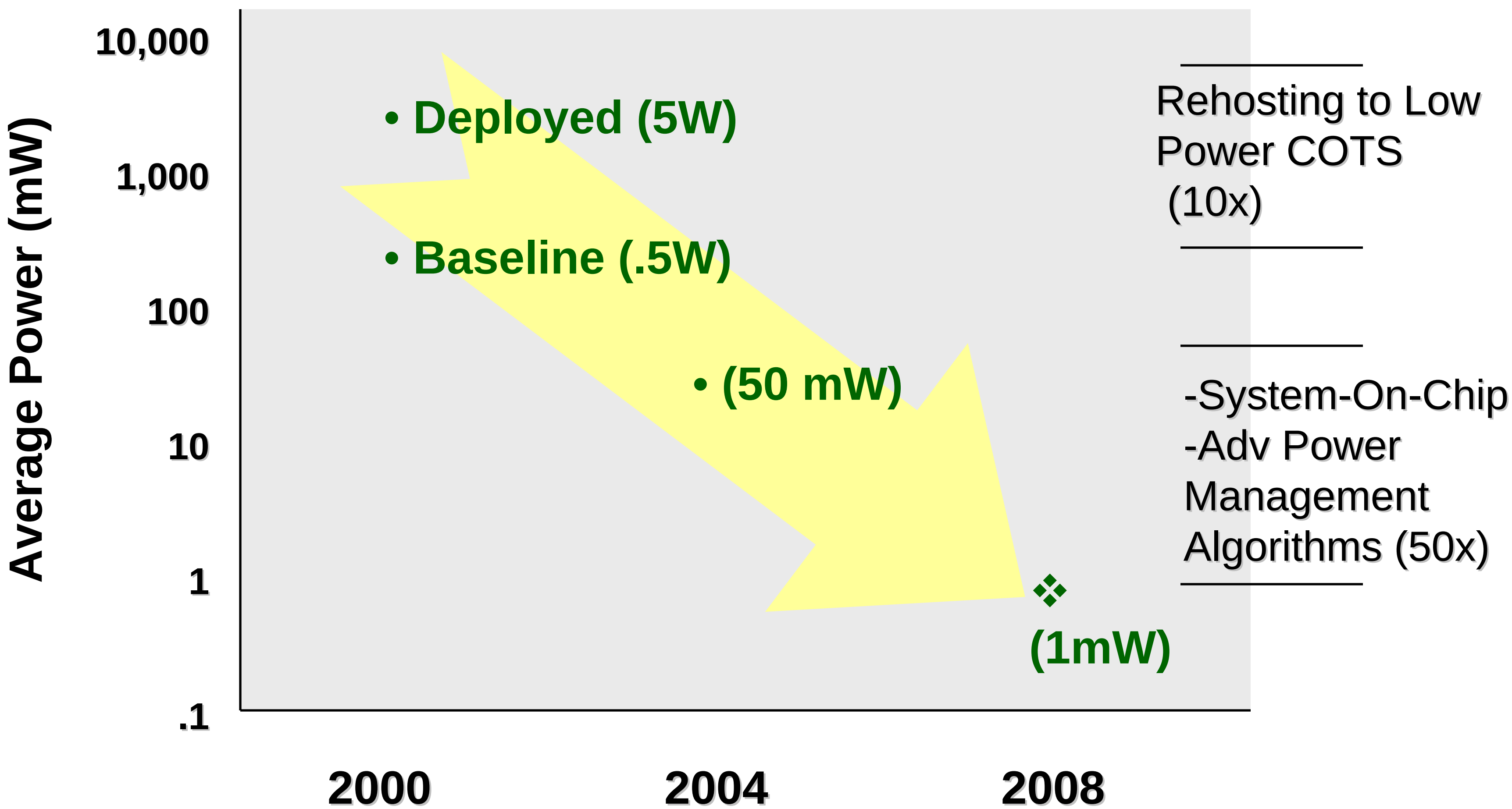
Low power is not always better!

- Energy really matters:
  - Bluetooth runs slower and thus has to stay on to finish transmission; 802.11b can transmit at max speed and then turn off – savings of 39%
  - StrongARM runs slower, so Xscale can run at top frequency and then go to sleep – savings of 26%

Wireless	802.11b	BT
Mbps	11	1
Power (W)	1.2	0.18
Time for 1Mb	0.09	1
Energy	0.11	0.18
Savings	39%	-

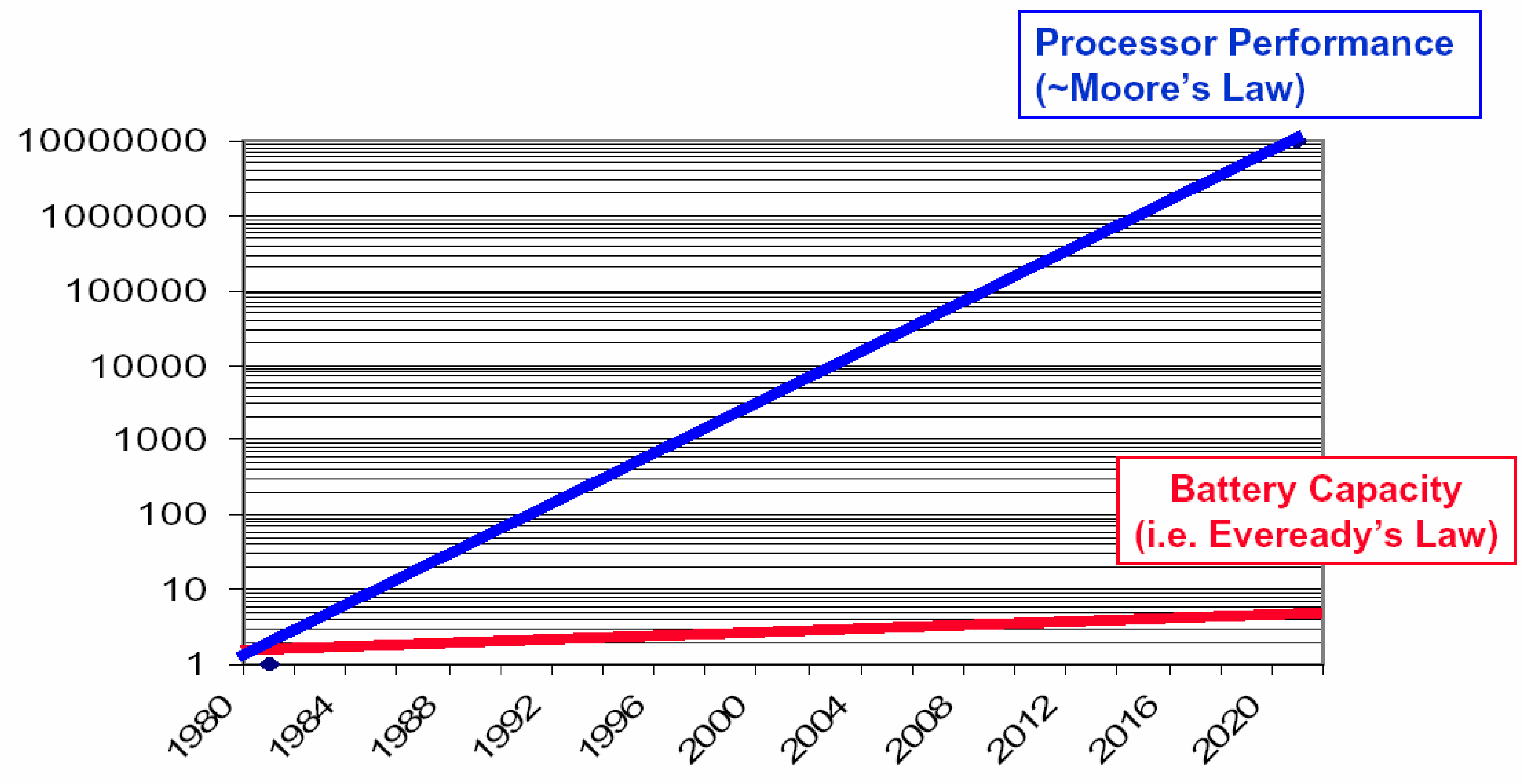
Processor	PXA270	SA-1100
Speed (MHz)	624	200
Power active (mW)	925	400
Power sleep (mW)	0.16	0.1
Ave. power (mW)	297	400
Savings	26%	-

Sensor Node Energy Roadmap



Source: ISI & DARPA PAC/C Program

Battery capacity vs. computing

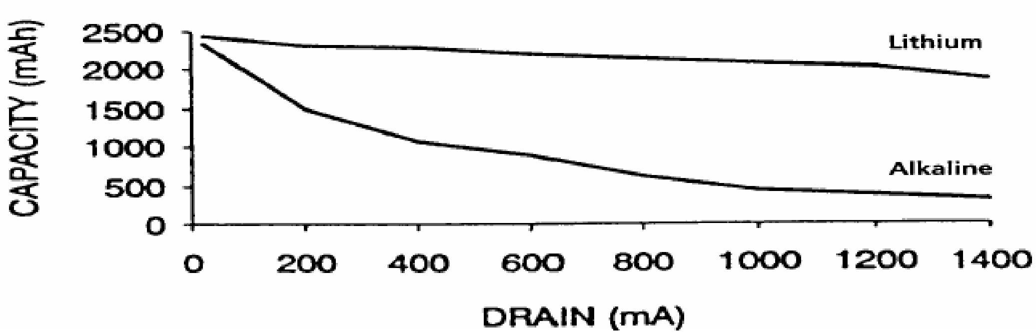


Battery characteristics

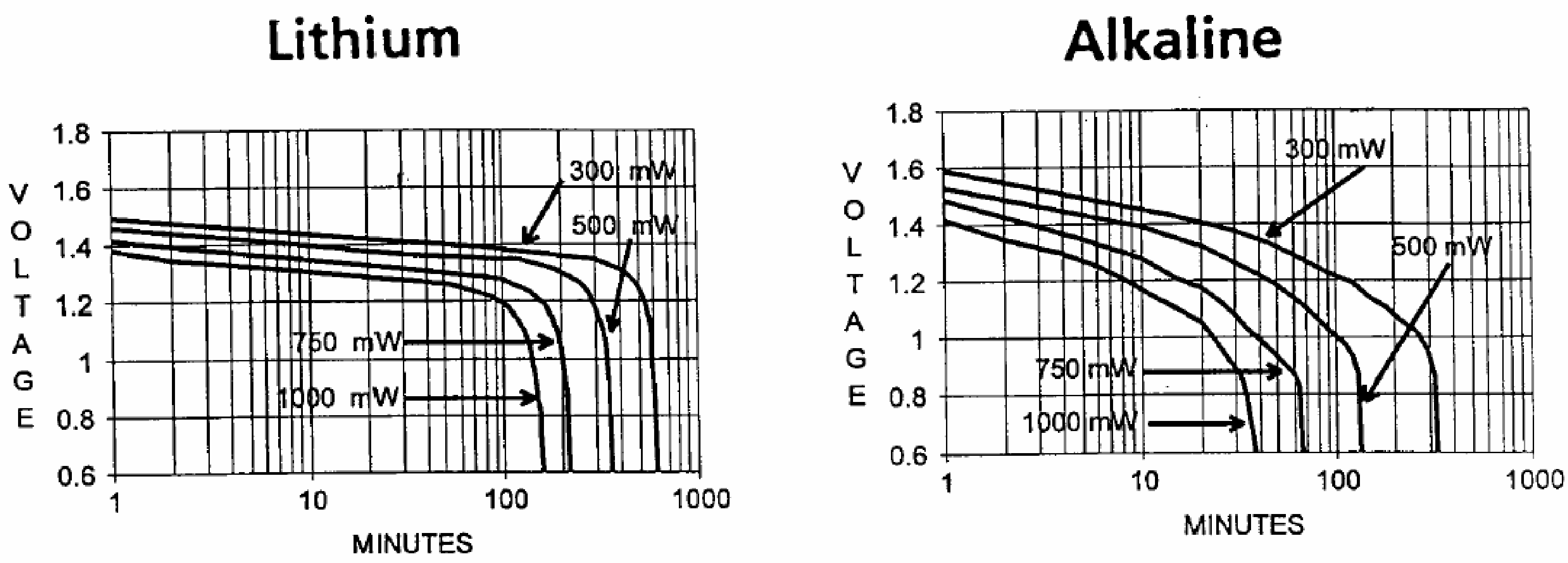
AA Primary Cylindrical Cells

System		E91 Alkaline Zn/MnO_2	L91 Lithium Li/FeS_2
Battery Weight (Grams)		23	14.5
Voltage	Nominal	1.5	1.5
	Open Circuit	1.6	1.8
Operating Time (Hours to 0.90 Volt)	1400 mA	0.2	1.3
	1000 mA	0.4	2.1
	400 mA	2.7	5.7
	20 mA	117	122
1kHz Impedance (Ohms)		0.17	0.18
Shelf Life (Years)		5	10

DRAIN vs. CAPACITY @ 21C



Battery efficiency



Constant power with continuous discharge at 21 degrees C

- Understand how system power is being used
  - Non-linear behavior, peak vs. average current
- Know the usage model
  - Camera: 1hr at 5 sec/pic = 720 pics!
  - Laptop: 1hr is not acceptable

Fuel cells

Product	Methanol fuel cell directly connected to the PC
Output	Average 12W Maximum 20W
Voltage	11V
Size	275 x 75 x 40mm (825cc)
Weight	900g
Operating hours	Approximately five hours with 50cc, and 10 hours with 100cc, concentration methanol fuel
Cartridge weight	120g (100cc), 72g (50cc) (Approximate)
Cartridge size	100cc: 50 x 65 x 35mm 50cc: 33x 65 x35mm
Fuel	Methanol

Toshiba Announces World's First Small Form Factor Direct Methanol Fuel Cell for Portable PCs"
http://www.toshiba.co.jp, March 5th, 2003

Micro Power Sources

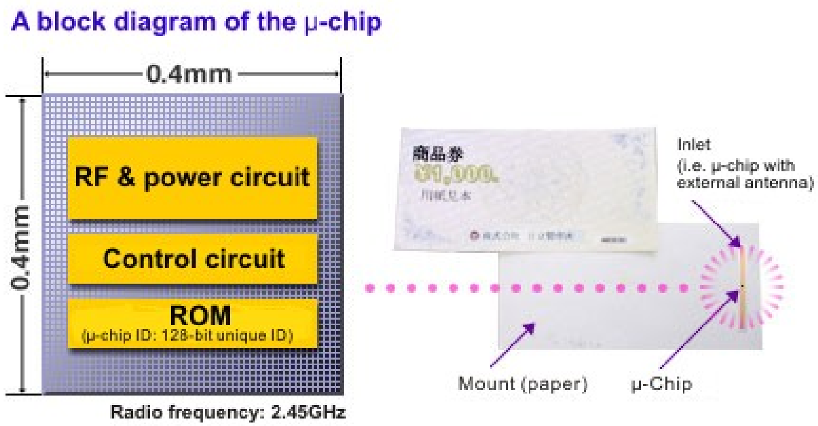
- Micro scale Battery
  - Similar to macro scale in source of energy
  - Problem is in power output due to surface area limitations
  - Challenge: To maintain performance while reducing size
- Micro-Fuel Cells
  - hydrocarbon based fuel cells have very high energy density compared to batteries
  - power electronics needed to stabilize voltage
  - very low efficiency (less than 1%)

Power Sources

- Micro heat engines
  - Miniaturization of heat engines and combustion based engines was not possible due to complexity and multitude of components involved
  - MEMS promises to revolutionize the field
  - Provides high power density but no efficient system has been developed yet
- Radio active sources
  - Provide extremely high energy density
  - Controversial technology
  - Low efficiency

Power Sources

- Electromagnetic (RF) power distribution
  - 2nd most common method of supplying power to embedded electronics
  - RF energy transmitted from nearby source to the device used to power the electronics
  - Example: µ- chip developed by Hitachi for RFID devices



•The µ-chip is a passive IC, that receives the microwave from the reader, generates electric power from the microwave, decodes its µ-chip ID and transmits it back to the reader
•4mm x 4mm size
•Operating frequency: 2.45 GHz

Power Scavenging

- Energy provided depends on how long the source is in operation
- Photovoltaic Cells
  - Fairly stable DC voltage provided
  - Used usually to charge secondary batteries
- Temperature gradient
  - Naturally occurring temperature variations can be used to scavenge energy
- Human Power
  - Average human body burns 10.5 MJ of energy per day
- Wind / Air flow
- Vibrations
  - PZTs etc.

Comparison of Energy Sources

Power Source	P/cm^3 (µW/cm^3)	E/cm^3 (J/cm^3)	P/cm^3/yr (µW/cm^3/Yr)	Secondary Storage Needed	Voltage Regulation	Comm. Available
Primary Battery	-	2880	90	No	No	Yes
Secondary Battery	-	1080	34	-	No	Yes
Micro-Fuel Cell	-	3500	110	Maybe	Maybe	No
Ultra-capacitor	-	50-100	1.6-3.2	No	Yes	Yes
Heat engine	-	3346	106	Yes	Yes	No
Radioactive(^{63}Ni)	0.52	1640	0.52	Yes	Yes	No
Solar (outside)	15000 *	-	-	Usually	Maybe	Yes
Solar (inside)	10 *	-	-	Usually	Maybe	Yes
Temperature	40 * †	-	-	Usually	Maybe	Soon
Human Power	330	-	-	Yes	Yes	No
Air flow	*380 ††*	-	-	Yes	Yes	No
Pressure Variation	*17 †††*	-	-	Yes	Yes	No
Vibrations	200	-	-	Yes	Yes	No

Roundy, EWSN '04

Summary

- Increasingly complex systems

  - Mixed hard & soft real-time requirements
  - Coordination of subsystem & multi-system control
- Demand for dynamic & adaptive response

  - Operation in unpredictably changing contexts
  - Variable performance demands

  - Management of resources (power, performance, availability, accessibility, throughput, security etc.)
- Faster hardware

  - Fast processors and networks
  - Integrated processing, common platforms

- Complex software development process