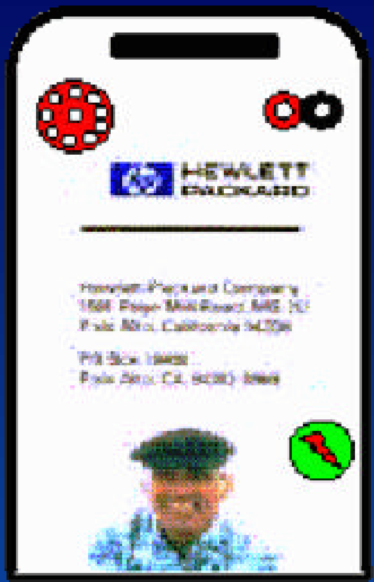
Cycle-Accurate Simulation of Energy Consumption in Embedded Systems

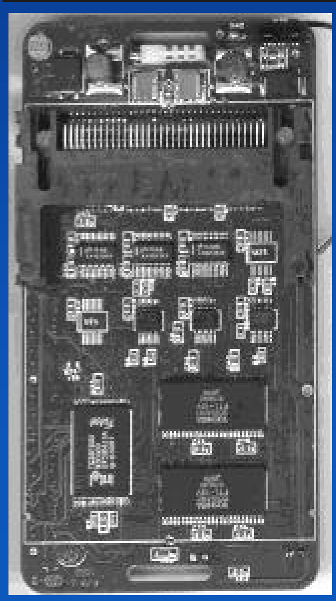
T. Simunic, L. Benini and G. De Micheli
Stanford University
Bologna University
Hewlett-Packard Laboratories

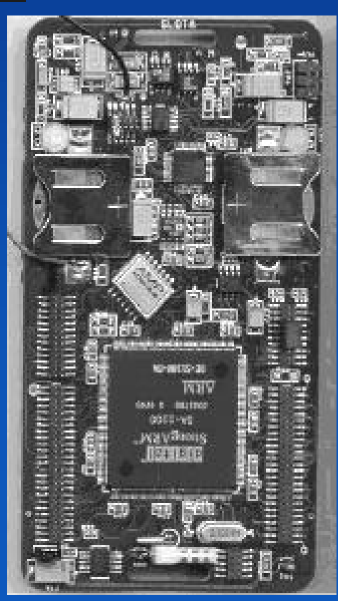
Motivation

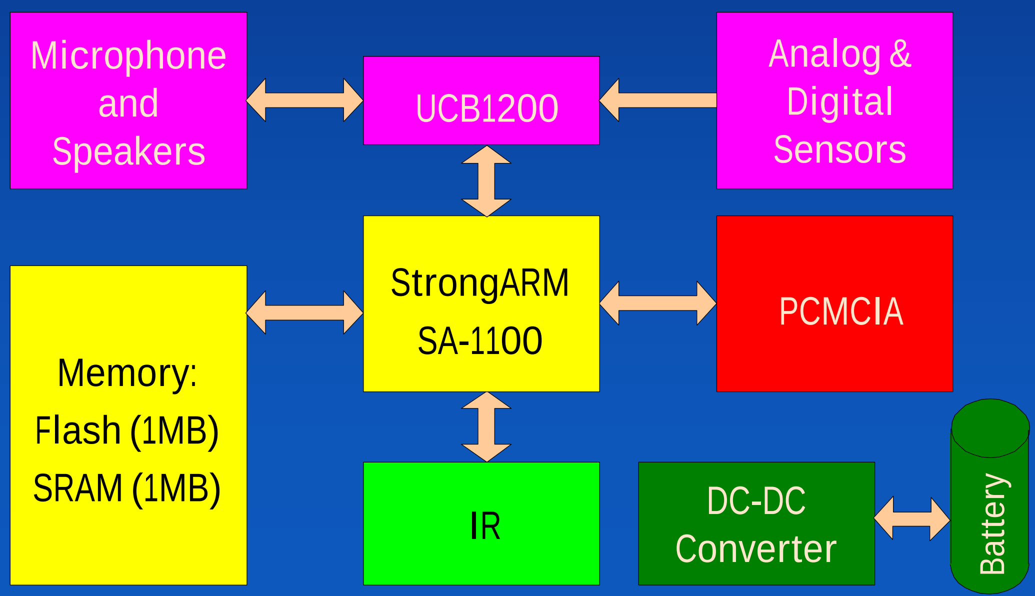


SmartBadge

- Small, portable design
- High performance
- Long battery life
- Discrete components







Problem

- Original SmartBadge running MPEG video decode
  - low performance: 6 frames/sec; need at least 20 frames/sec
  - high energy consumption → very low battery life
  - both hardware and software redesign needed

- Problem:
  - Hardware redesign requires evaluating multiple architecture options
  - Software design directly depends on the hardware architecture
  - Multiple iterations of board design are costly and slow
  - Verilog simulations are still too slow
  - Instruction-level simulator has only performance models

Contribution

Cycle-accurate energy consumption simulator

within 5% accuracy of hardware measurements

at speed comparable to the instruction-level simulator

Previous Work

- Architecture-level power modeling [eg. Landman et al., Liu]
  - analysis done at the netlist level
  - technology files required
- Energy and performance models of design components
  - cache [Kamble et al., Wilton and Jouppi]
    - capacitance and resistance values from technology files
    - run time statistics for hit/miss and read/write counts
  - RAM [Itoh et al.] requires technology parameters and netlist

Previous Work (cont.)

- Instruction-level power analysis [Tiwari et al., Wan]
  - measure energy consumption of each assembly instruction
  - measure energy consumption of non-ideal execution (eg. stall)
  - off-line analysis of assembly code gives total energy consumed
  - average power measured for instructions on StrongARM:
    - 200 mW at 170 MHz for most instructions
    - 260 mW at 170 MHz for loads and stores

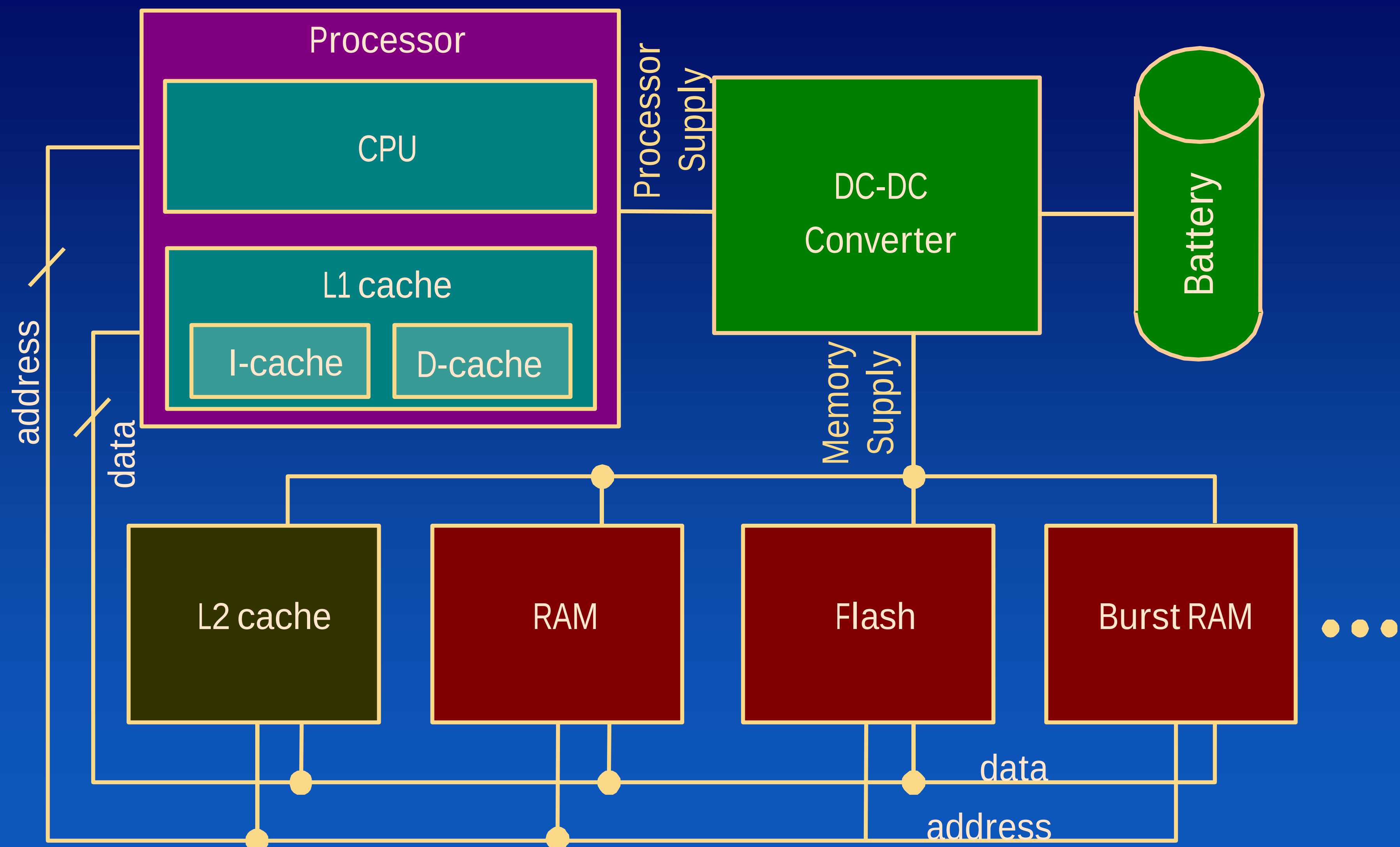
Previous Work (cont.)

- System-level energy simulation [Benini et al.]
  - system described as a state machine
  - each component has multiple power and performance states
  - very high level
- System-level SOC energy simulation [Li et al., Kapoor] for processor, cache and memory
  - each component analyzed separately
  - technology information needed for modeling

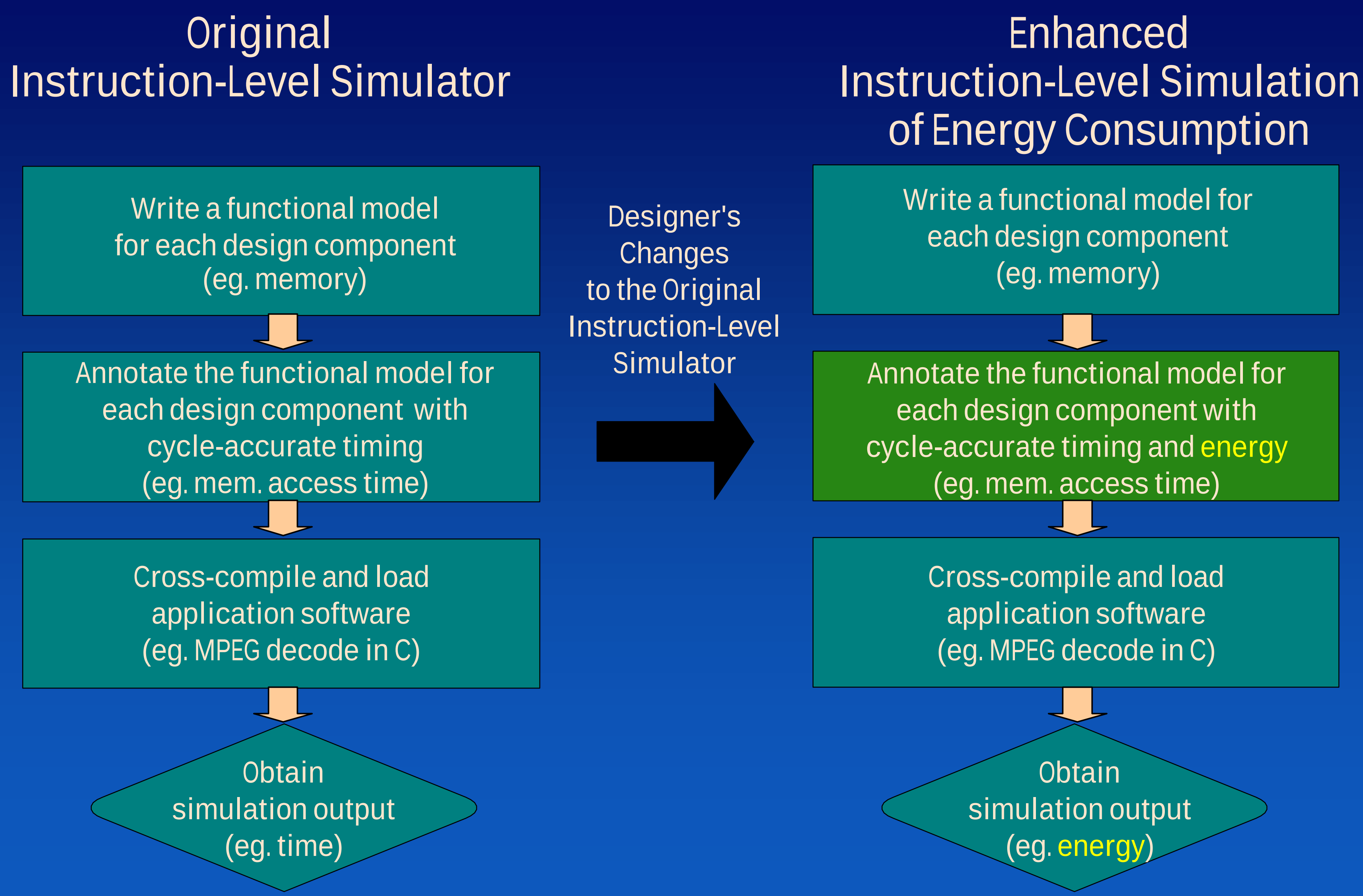
Solution

- Extend instruction-level simulator with energy models for all design components
  - component energy models are based on the data sheets
  - components are analyzed dynamically on cycle-accurate basis
    - very fast
    - many software and hardware architectures can be easily evaluated
  - plots of energy consumption vs. time are available
    - peak energy consumption analysis
    - detailed algorithm analysis

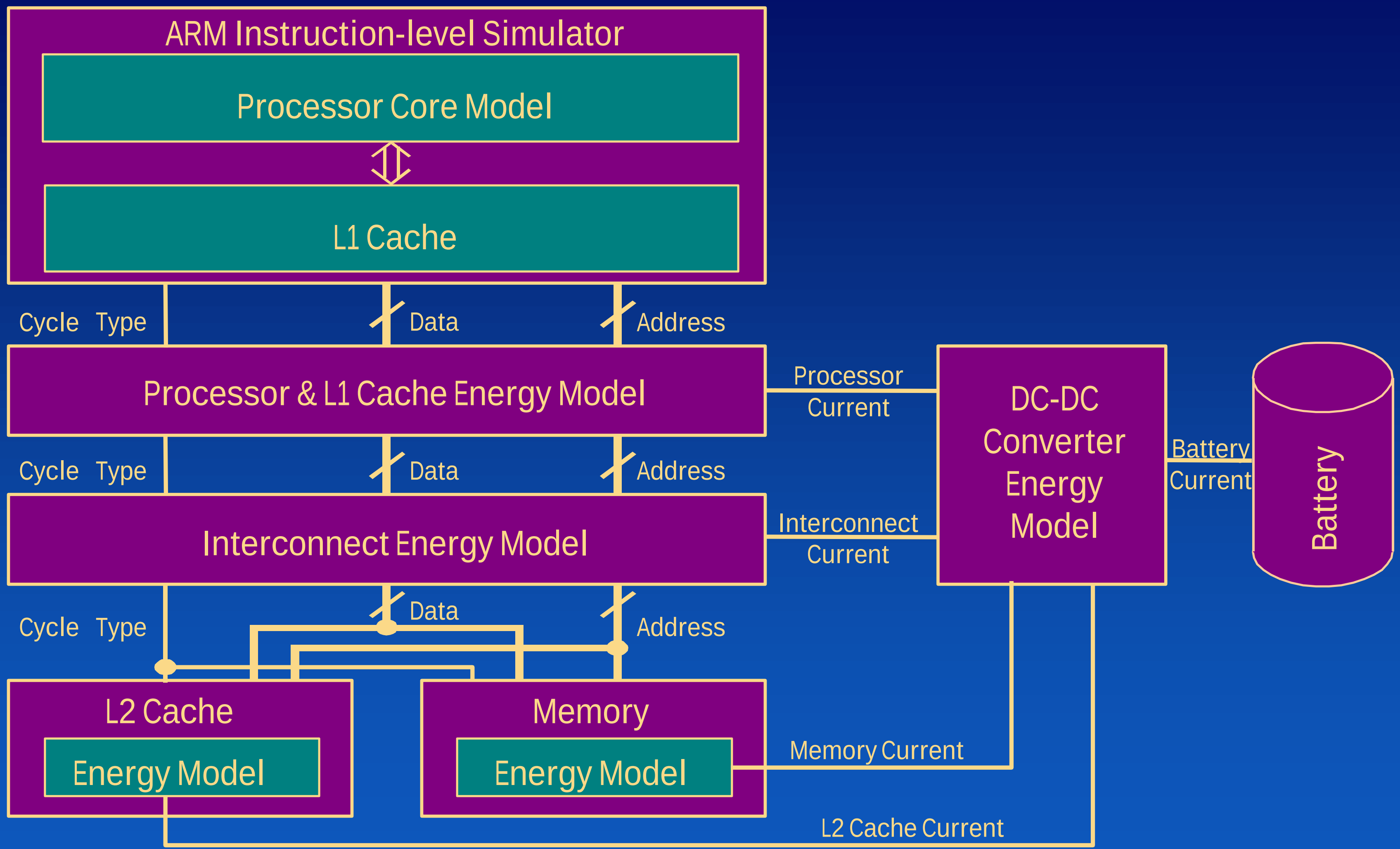
System Model



System Simulation Setup



Simulator Architecture



$$E_{Cycle} = E_{CPU} + E_{Memory} + E_{L2Cache} + E_{Interconnect} + E_{DC/DC}$$

Processor Energy Model

- Modeled ARM processor family
  - any processor family can be modeled with appropriate instruction-level simulator
- Estimated active and NOP processor energy per cycle from the values given in the data sheets:

Active Energy

$$E_{CPU,Active} = C_{CPU,Active} V_{cc}^2$$

$$C_{CPU,Active} = \frac{P_{Active}}{V_m^2 f_m}$$

NOP Energy

$$E_{CPU,NOP} = C_{CPU,NOP} V_{cc}^2$$

$$C_{CPU,NOP} = \frac{P_{NOP}}{V_m^2 f_m}$$

Memory Energy Model

- Model burst or normal RAM, FLASH and L2 Cache
- Number of wait cycles estimated by:

$$N_{wait} = \frac{T_{mem}}{T_{CPU}}$$

- Estimated active and idle memory energy per cycle from the values given in the data sheets:

Active Energy

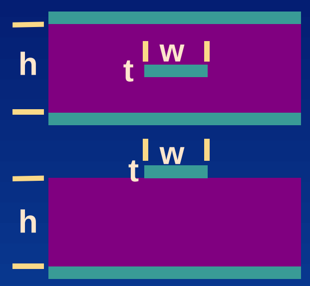
$$E_{Mem,Active} = \frac{C_{Mem,Active} V_{cc}^2}{N_{Wait} + 1} \qquad C_{Mem,Active} = \frac{P_{Mem,Active}}{V_m^2 f_m}$$

Idle Energy

$$E_{Mem,idle} = T_{Cycle} \sum_{idle=0}^{n} P_{idle} \rho_{idle}$$

Interconnect and Pins Energy Model

- Estimated total switched capacitance from the number of lines switching, interconnect cross-section, line length and pin capacitance



$$C_{Line} = \begin{cases} L_{Stripline} \; C_{Stripline} \\ L_{Microstrip} \; C_{Microstrip} \end{cases}$$

$$C_{Switch} = \sum_{Switch=0}^{n} (C_{Line} + C_{Pins})$$

- Total energy per cycle depends on the switched capacitance, frequency of access and voltage swing:

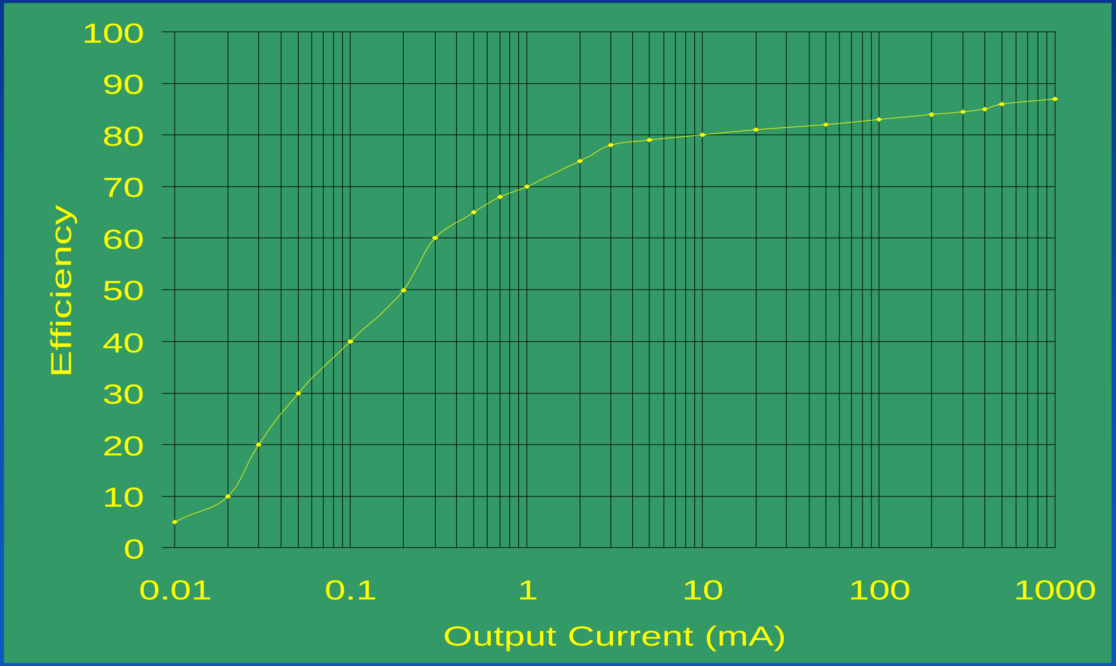
$$E_{Interconnect,Active} = \frac{C_{Switch} V_{cc}^{2}}{N_{Wait} + 1}$$

DC-DC Converter Energy Model

- Total energy per cycle is the difference between the energy supplied by the DC/DC converter to the portable system and energy supplied by the battery

$$E_{DC/DC} = I_{Bat} V_{Bat} T_{Cycle} - E_{DC/DCout}$$

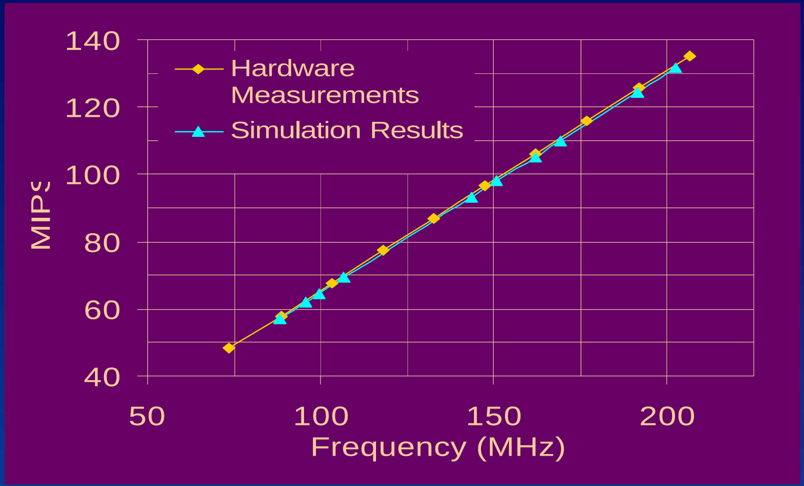
$$I_{Bat} = \frac{I_{DC/DCout}}{Efficiency}$$

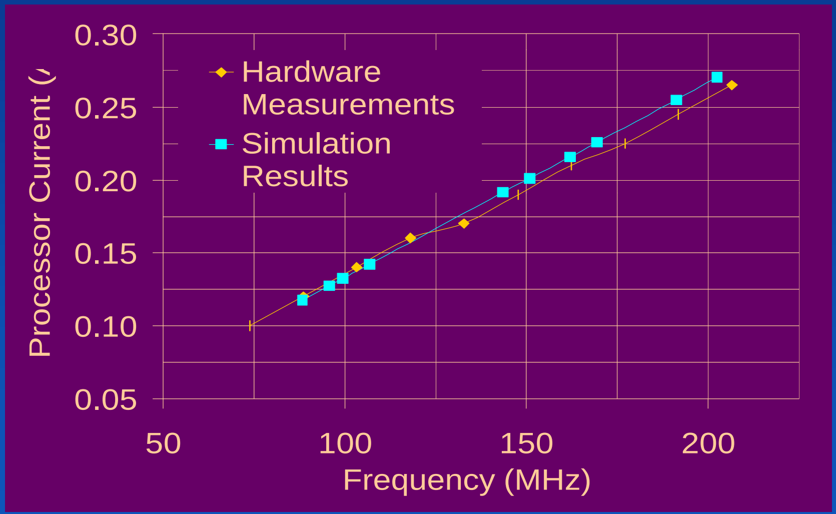


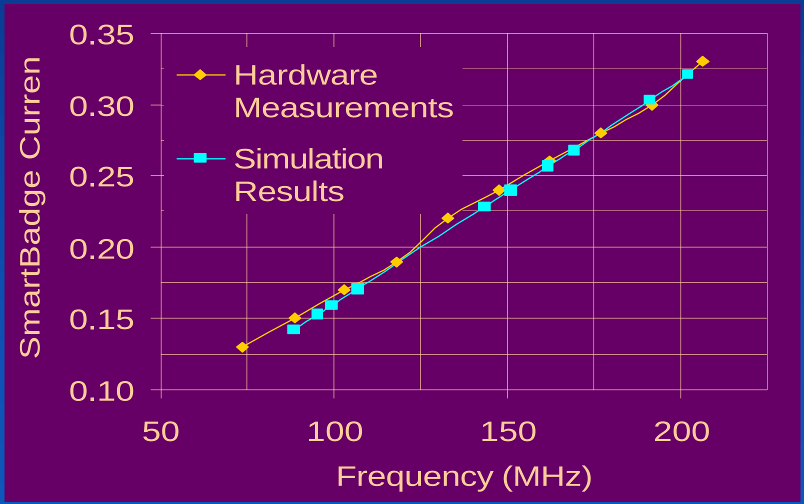
Validation of Simulation Methodology

SmartBadge Configuration

Component Units	Cycle Time (ns)	Active Power (mW)	Idle Power (mW)	Pin Cap. (pF)	Line Length (cm)
SA-1100	5-20	400	170	5	N/A
FLASH (1MB)	80	74	0.5	10	2
SRAM (1MB)	90	55	0.01	8	3







Dhrystone benchmark energy ***simulations and hardware measurements*** on SmartBadge are ***within 5% tolerance***

MPEG Decode Hardware Design Exploration

- ARM processor with 64KB L1 cache running at 200 MHz, 400 mW active and 170 mW idle power consumption

Hardware Configurations

Name	Instruction Memory	Data Memory	L2 Cache Present
Original	FLASH	SRAM	no
L2 Cache	FLASH	BSDRAM	yes
Burst SRAM	BFLASH	BSRAM	no
Burst SDRAM	BFLASH	BSDRAM	no

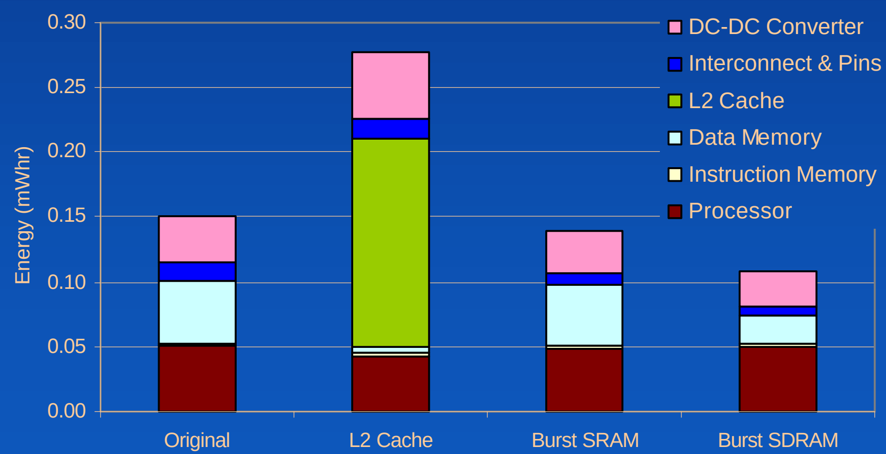
Memory Architectures

Name Units	Initial Access (ns)	Burst Access (ns)	Active Power (mW)	Idle Power (mW)	Interconnect Capacitance (pF/line)	I/O Pin Capacitance (pF/pin)	Manufacturer
FLASH	80	N/A	75	0.5	4.8	10	Intel
BFLASH	80	40.00	600	2.5	4.8	10	TI
SRAM	90	N/A	185	0.1	8	8	Toshiba
BSRAM	90	45.00	365	1.7	8	8	Micron
BSDRAM	30	15.00	430	10	8	8	Micron
L2 Cache	20.00	10	1985	330	3.2	5	Motorola

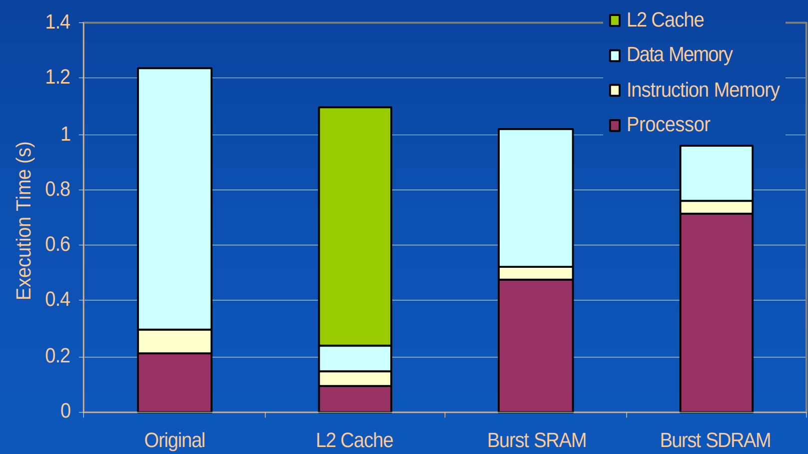
Hardware Design Exploration Results

- Data memory speed limits energy and performance efficiency, but instruction memory speed is not a limitation
- The most energy and performance efficient design uses fast and power hungry burst SDRAM
- L2 cache is neither energy nor performance efficient

Energy Consumption



Execution Time



MPEG Decode Software Design Exploration

- MPEG input data format
  - I-frame
    - jpeg encoded frame
  - P-frame
    - differences between the current and the previous frame
  - B-frame
    - differences between the current and both the previous and the future frames
  - combinations of I,P,B frames in a group of pictures (GOP)
- Decoding speed
  - faster decoding speed means less data

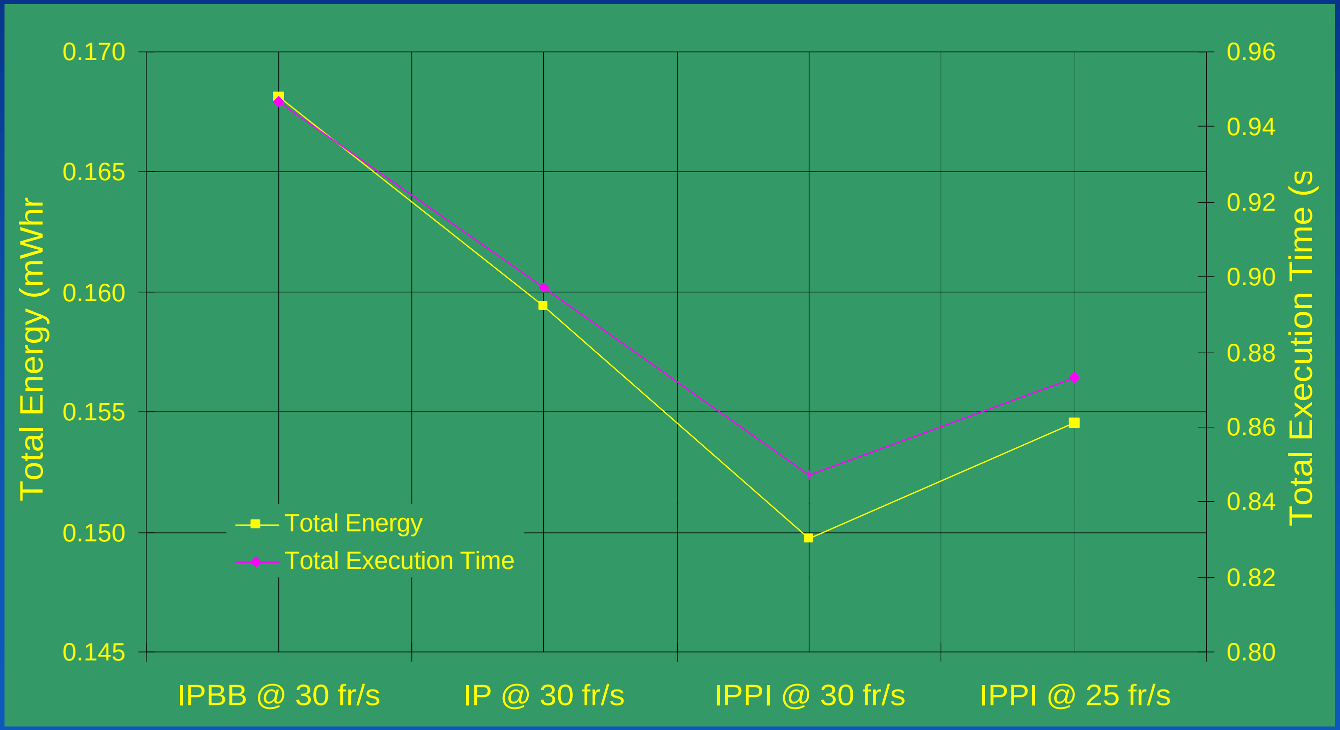
Software Configurations

Configuration	Speed (frames/s)	I-frames (number)	P-frames (number)	B-frames (number)
IPBB @	30	2	3	7
IP @	30	2	10	0
IPPI @	30	4	8	0
IPPI @	25	4	8	0

Software Design Exploration Results

- Combination of I and P-frames performs best for energy consumption and execution time
- B-frame decoding is not energy or time efficient
- Faster decoding speed gives energy savings

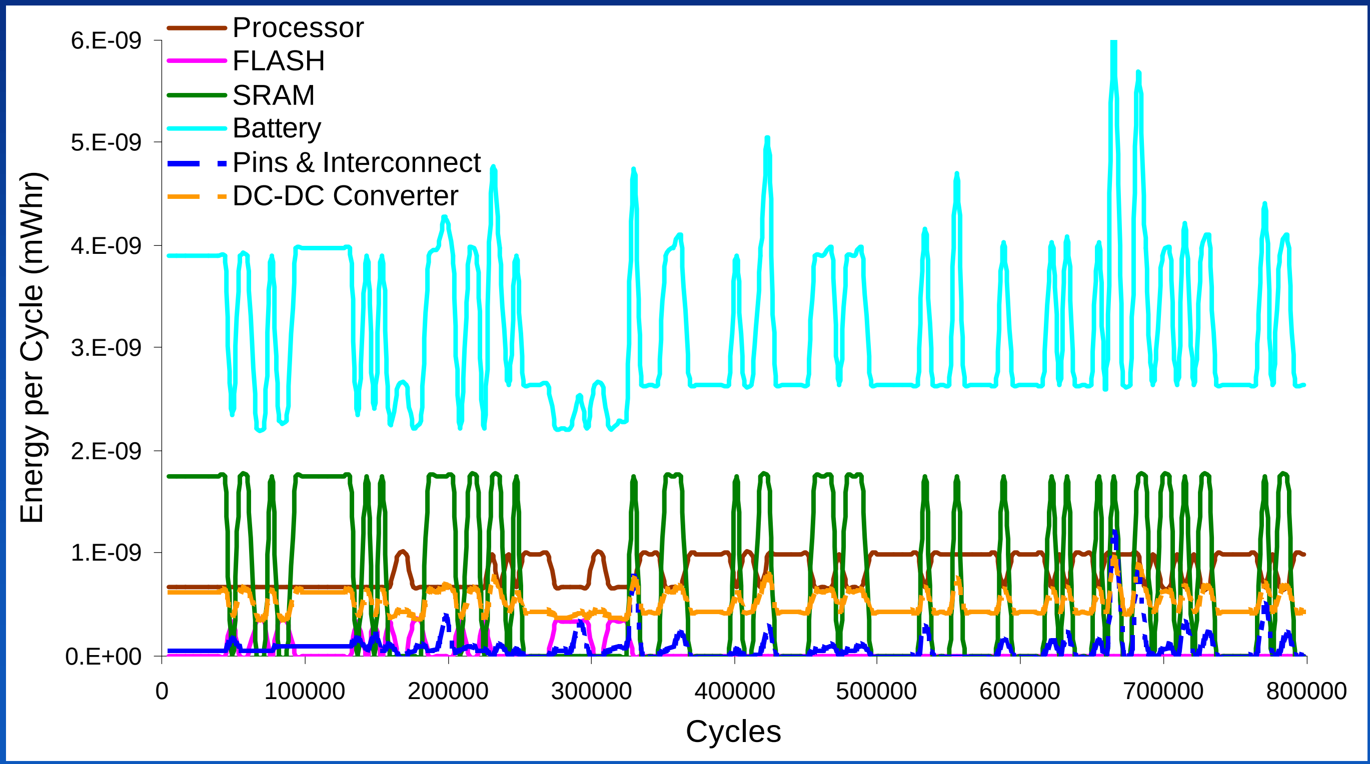
Energy Consumption and Execution Time



Peak Energy Consumption

- Peak energy consumption can be more than two times larger than the average so DC-DC converter, battery and thermal design have to be specified accordingly

Energy Consumption over Time



Conclusions and Future Work

- A methodology for cycle-accurate energy consumption simulation of discrete component designs has been presented
- Designer's extensions to the instruction-level simulator are minimal
- Simulation is within 5% accuracy of the hardware measurements
- Design exploration of both hardware and software architectures
  - MPEG decode design exploration example
- Plots of energy consumption over time are available
- Future extensions:
  - model of wireless link input
  - model the video output