McGill

ECSE 421 Lecture 14: More Evaluation and Validation

ESD Chapter 5

Last Time

- Evaluation and Validation
  - In general, multiple objectives
  - Pareto optimality
  - Design space evaluation (DSE)
- Performance Analysis
  - Always a trade-off between x and y
  - Worst-case execution time
  - Real-time calculus

Where Are We?

4	W	3-Feb-2016	L07	Virtual Memory	B.4			Guest: Y. Akaveeti
	F	5-Feb-2016	L08	Introduction to Pipelining	C.1	PD3	PD2	
5	W	10-Feb-2016	L09	Pipeline Hazards	C.2			
	F	12-Feb-2016	L10	Implementing Pipelining	C.3-C.4			
6	W	17-Feb-2016		***Midterm exam: in-class, closed book***				Ch 1, B, and C
	F	19-Feb-2016	L11	Introduction to ILP	3.1			
7	W	24-Feb-2016	L12	Pipeline Scheduling	3.2	PD4	PD3	
	F	26-Feb-2016	L13	Branch Prediction	3.3			
	W	2-Mar-2016		***No Class***				Winter break
	F	4-Mar-2016		***No Class***				Winter break
8	W	9-Mar-2016	L14	Dynamic Scheduling	3.4-3.5			
	F	11-Mar-2016	L15	Hardware Speculation	3.6			
9	W	16-Mar-2016	L16	Superscalar Execution	3.7-3.9			
	F	18-Mar-2016	L17	Limits of ILP and Multi-threading	3.10, 3.12			
10	W	23-Mar-2016	L18	Vector Architecture	4.1			
	F	25-Mar-2016		***No Class***				Good Friday
11	W	30-Mar-2016		Vector Architecture	4.2	PD5	PD4	
	F	1-Apr-2016	L19	SIMD Extensions	4.3			
12	W	6-Apr-2016		GPUs	4.4			
	F	8-Apr-2016	L20	Detecting and Enhancing LLP	4.5			
13	W	13-Apr-2016	L21	Intel Processor Architecture				
	F	15-Apr-2016		***Final Exam Review***			PD5	
15	M	25-Apr-2016		***Final Exam: closed book, cumulative***				2:00 PM

Today

- More Evaluation
  - Energy and power
  - Temperature
  - Reliability
- Validation

Multi-Objective Design

- Different design objectives/criteria are relevant:
  - Average performance
  - Worst case performance
  - Energy/power consumption 
  - Thermal behavior
  - Reliability
  - Electromagnetic compatibility
  - Numeric precision
  - Testability
  - Cost
  - Weight, robustness, usability, extendibility, security, safety, environmental friendliness

Energy- and Power

$$E = \int P dt$$

- Power/energy models are increasingly important
  - Mobile computing
  - Power and energy rise with performance
  - Environmental issues

Energy Models

- Tiwari (1994): Energy consumption within processors
- Simunic (1999): Using values from data sheets
  - Allows modeling of all components, but not very precise
- Russell, Jacome (1998): Based on precise measurements from two fixed configurations
- Steinke et al., UniDo (2001): Mixed model using measurements and prediction
- CACTI [Jouppi, 1996]: Energy consumption of caches
- Wattch [Brooks, 2000]: Power estimation at the architectural level, without circuit or layout

McPAT [Li *et al.*, 2009]

- State-of-the-art model
- Interfaces with performance tools
- Models systems at multiple levels of abstraction
- Extensible

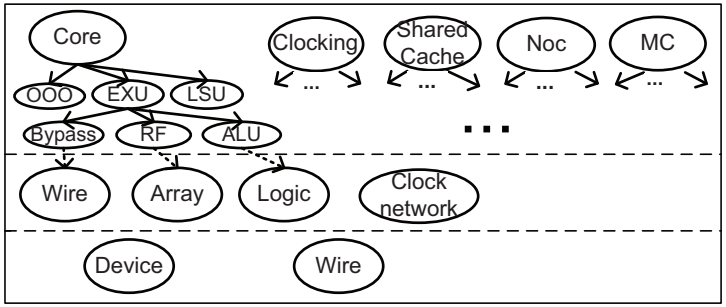


Figure 2: Modeling methodology of McPAT.

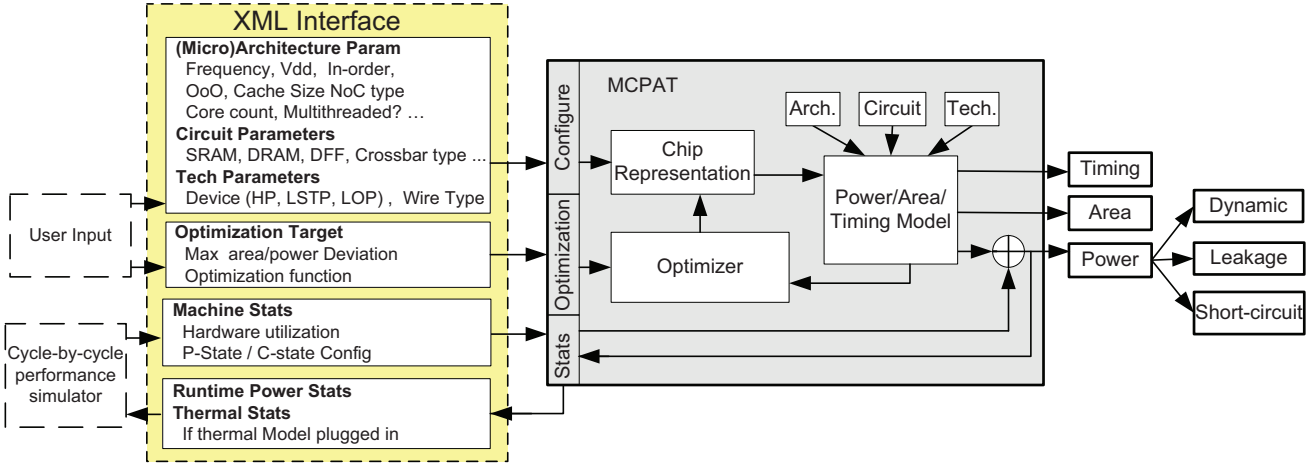
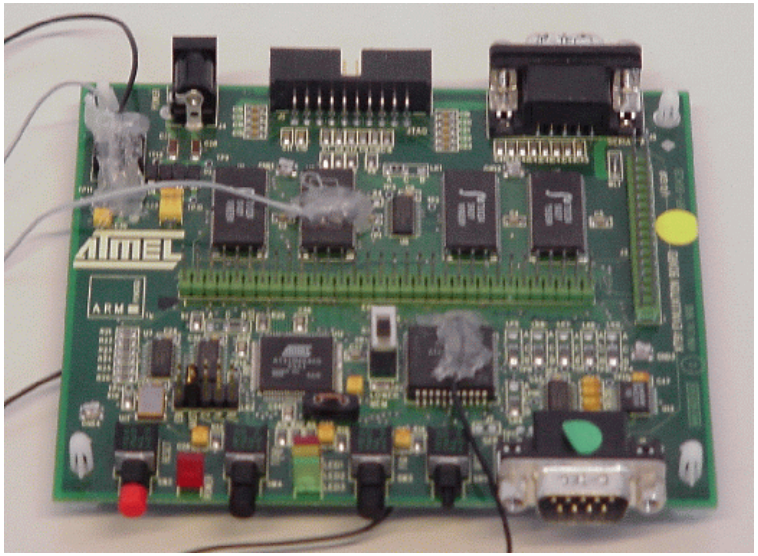
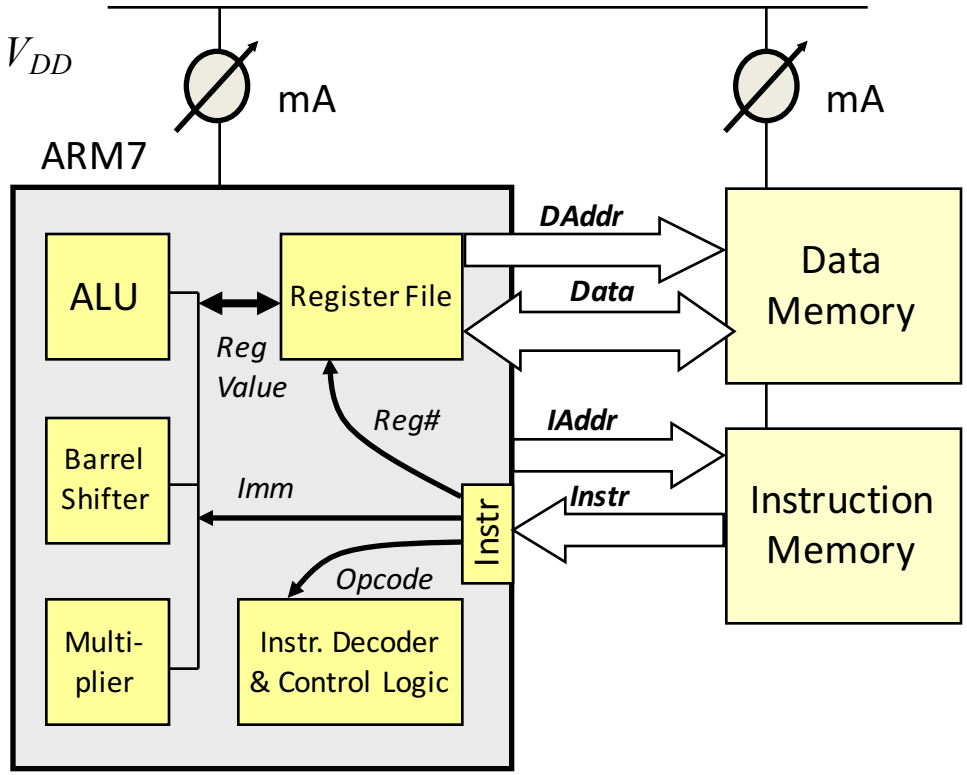


Figure 1: Block diagram of the McPAT framework.

Steinke & Knauer's Model



E.g.: ATMEL board with ARM7TDMI and ext. SRAM



$$E_{total} = E_{cpu\_instr} + E_{cpu\_data} + E_{mem\_instr} + E_{mem\_data}$$

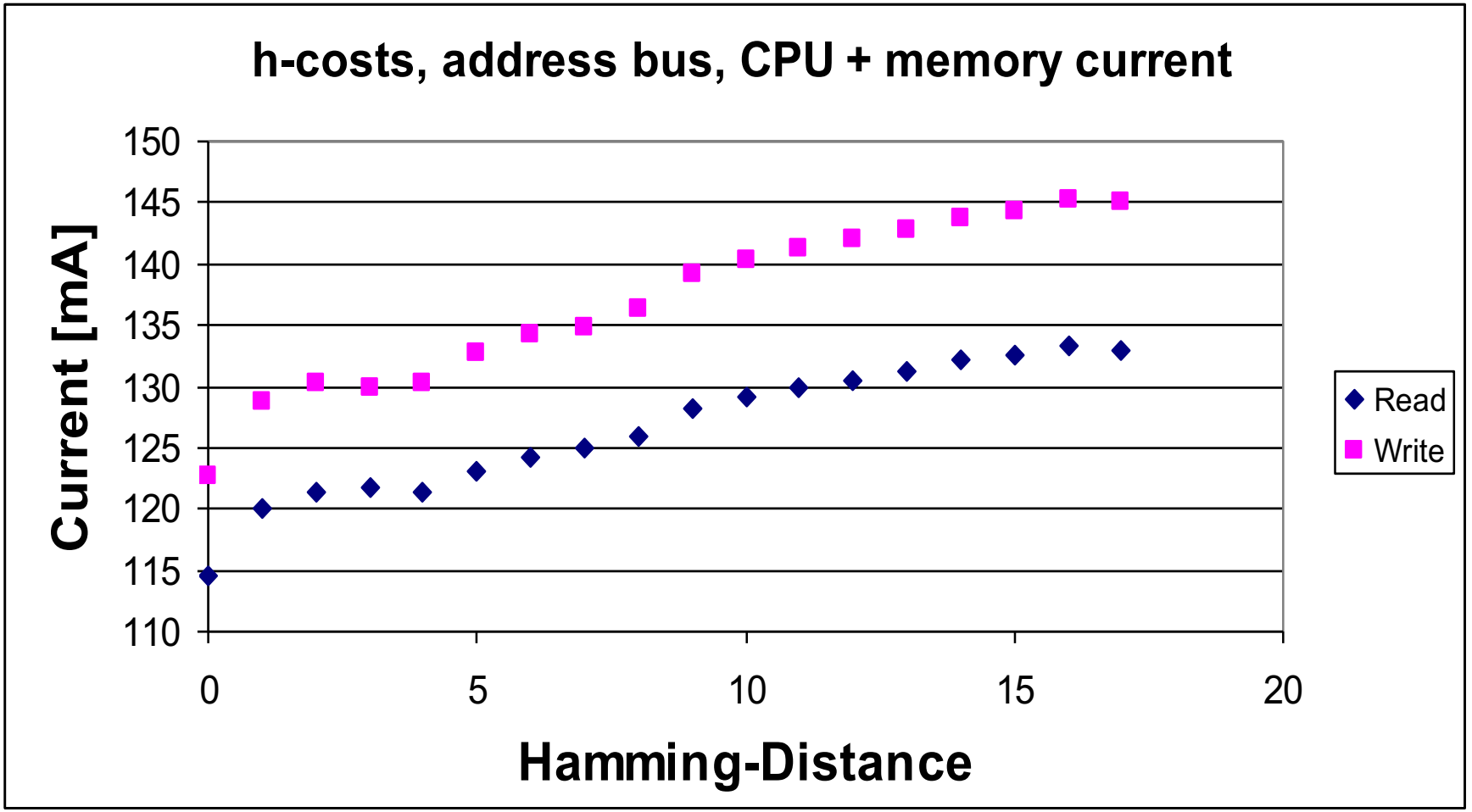
Example: Per Instruction Energy

Cost for a sequence of m instructions

$$E_{cpu\_instr} = \sum \text{MinCostCPU}(Opcode_i) + \text{FUCost}(Instr_{i-1}, Instr_i) +$$
$$\alpha_1 * \sum w(Imm_{i,j}) + \beta_1 * \sum h(Imm_{i-1,j}, Imm_{i,j}) +$$
$$\alpha_2 * \sum w(Reg_{i,k}) + \beta_2 * \sum h(Reg_{i-1,k}, Reg_{i,k}) +$$
$$\alpha_3 * \sum w(RegVal_{i,k}) + \beta_3 * \sum h(RegVal_{i-1,k}, RegVal_{i,k}) +$$
$$\alpha_4 * \sum w(IAddr_i) + \beta_4 * \sum h(IAddr_{i-1}, IAddr_i)$$

w:	number of ones;
h:	Hamming distance;
FUCost:	cost of switching functional units
α, ß:	determined through experiments

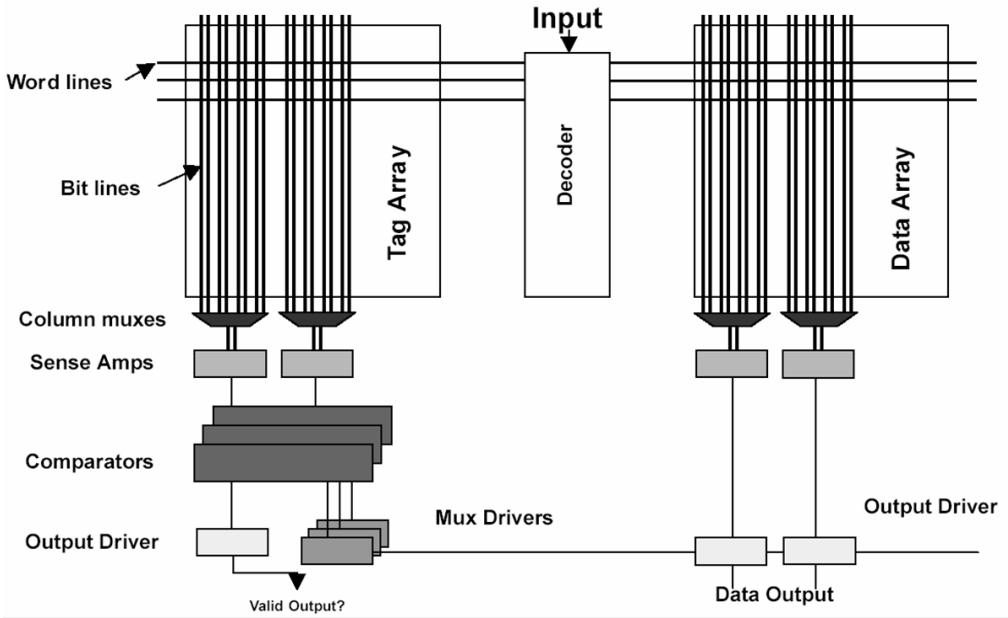
Address Bus Hamming Distance

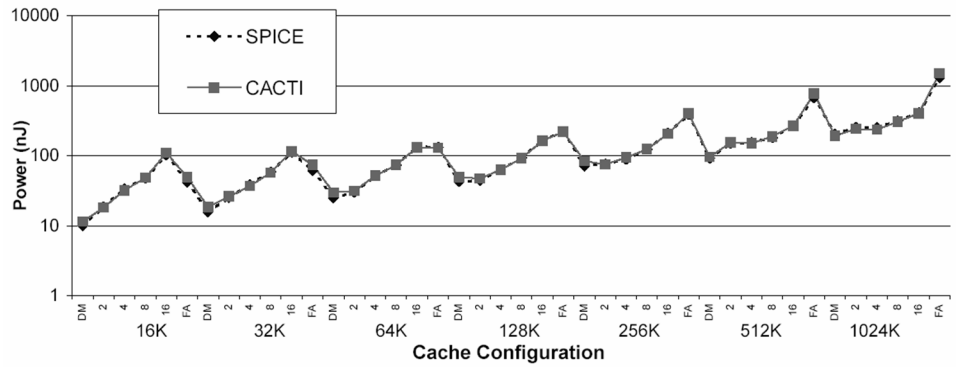


Results

- Memory address bus
  - The bit position of ‘1’s is irrelevant
  - The number of ‘1’s is irrelevant
  - The cost of flipping a bit on the address bus is independent of the bit position
- Memory data bus
  - The bit position of ‘1’s is irrelevant
  - The number of ‘1’s has a minor effect (3%)
  - The cost of flipping a bit on the data bus is independent of the bit position

CACTI



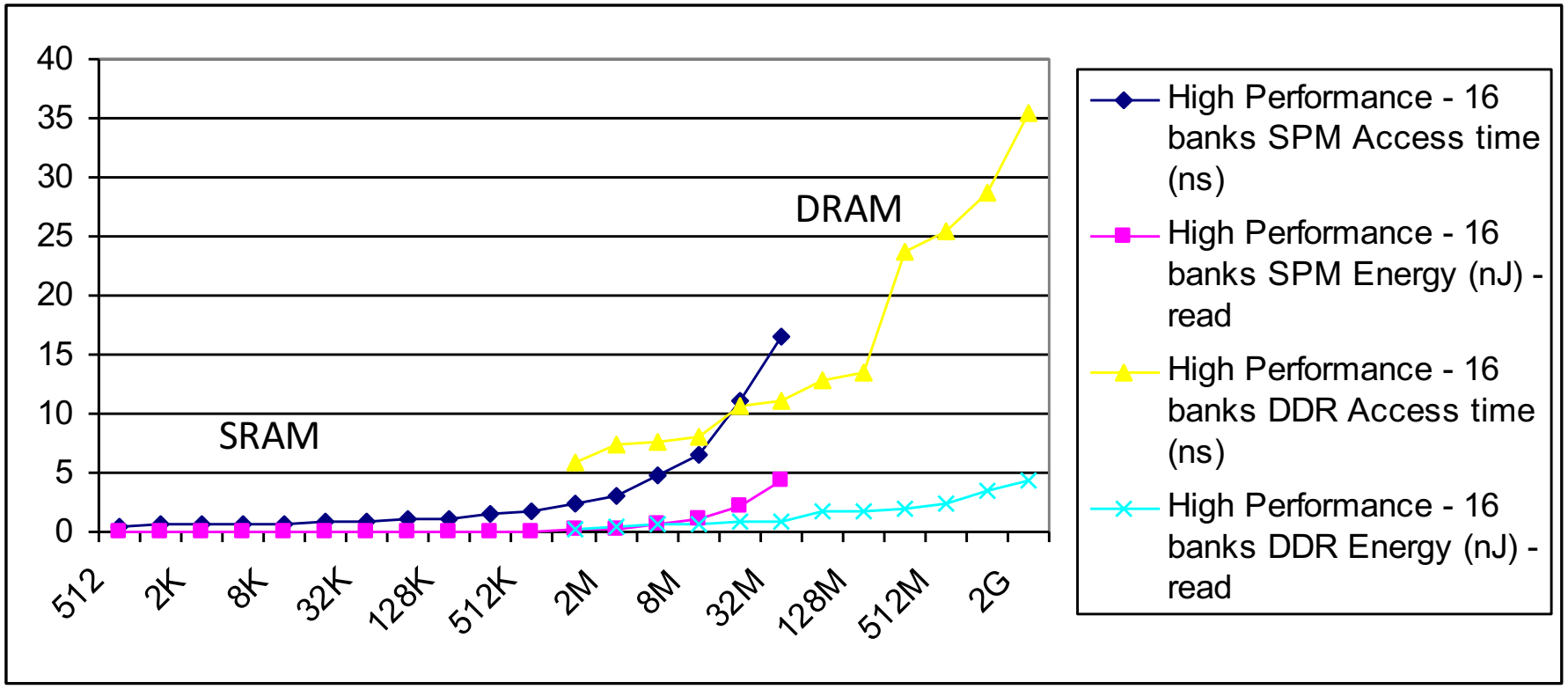


- Parameterized model
- You give
  - Size, ports
  - Technology
  - Optimization target
- You get
  - Energy
  - Area
  - Delay

http://research.compaq.com/wrl/people/jouppi/CACTI.html

Energy Consumption and Access Time

- Scratchpad SRAM vs. DDR2 DRAM -



- http://www.hpl.hp.com/research/cacti/
- 16 bit read; size in bytes;
- 65 nm for SRAM, 80 nm for DRAM

Source: Olivera Jovanovic, TU Dortmund, 2011

Multi-Objective Design

- Different design objectives/criteria are relevant:
  - Average performance
  - Worst case performance
  - Energy/power consumption
  - Thermal behavior 
  - Reliability
  - Electromagnetic compatibility
  - Numeric precision
  - Testability
  - Cost
  - Weight, robustness, usability, extendibility, security, safety, environmental friendliness

Thermal Models

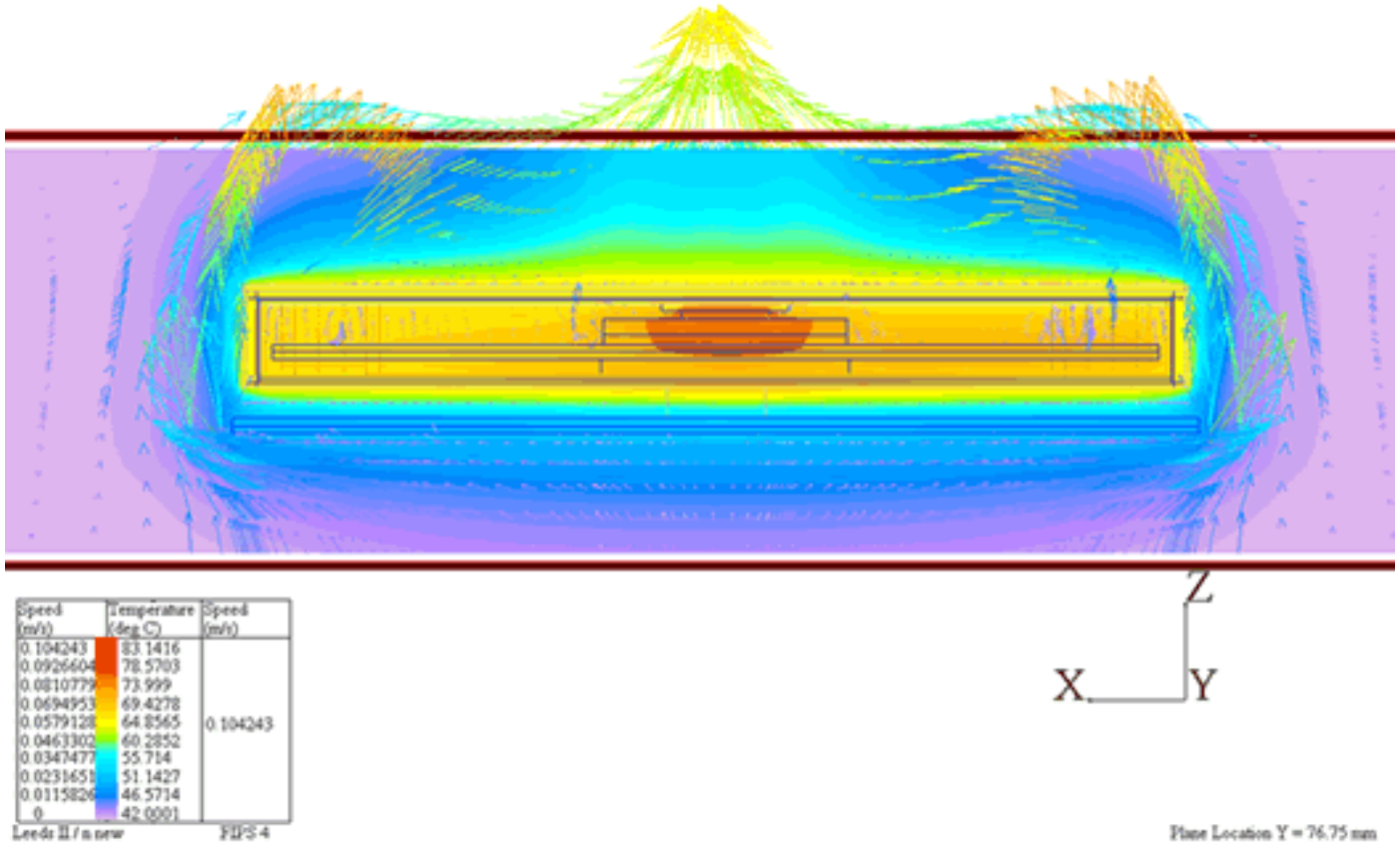
- Thermal models are increasingly important
  - Temperature rises with performance
  - Temperature affects
    - Usability
    - Dependability

Model Components

- **Thermal conductance**
  - The amount of heat transferred through a plate of area A and thickness L when the temperatures at the opposite sides differ by one Kelvin
- **Thermal resistance**
  - The reciprocal of thermal conductance
- Thermal modeling uses equivalent electrical models
  - Thermal resistances add up like electrical resistances
  - Masses storing heat correspond to capacitors
  - Well-known techniques for solving electrical network equations can then be used to determine temperature

Thermal Model Simulation Results (1)

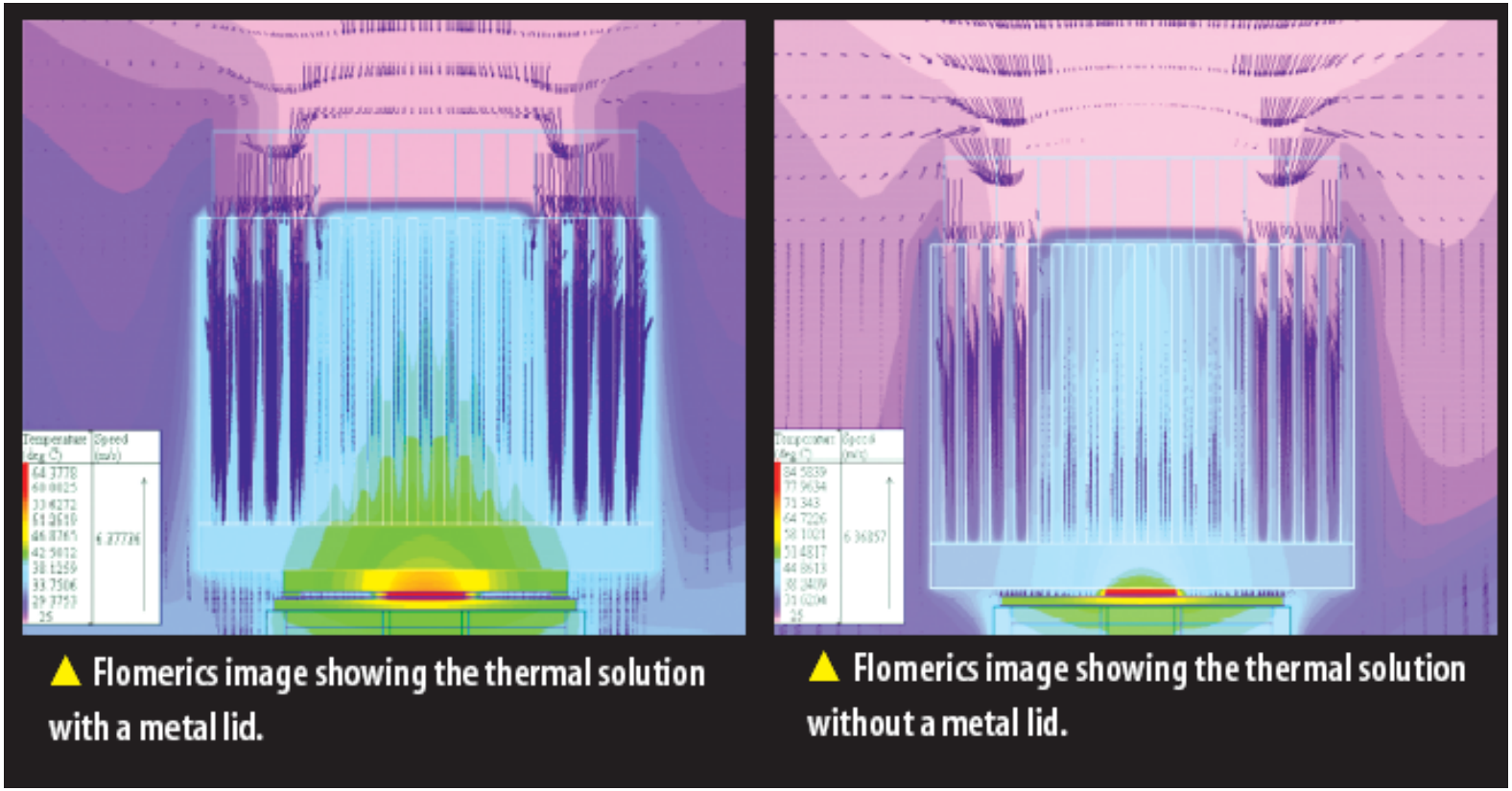
Packaged cryptographic coprocessor:



Source: http://www.coolingzone.com/Guest/News/NL_JUN_2001/Campi/Jun_Campi_2001.html

Thermal Model Simulation Results (2)

Microprocessor



▲ Flomerics image showing the thermal solution with a metal lid.

▲ Flomerics image showing the thermal solution without a metal lid.

Source: http://www.flotherm.com/applications/app141/hot_chip.pdf

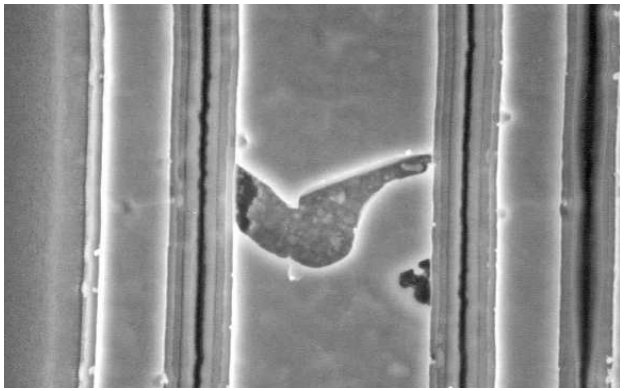
Multi-Objective Design

- Different design objectives/criteria are relevant:
  - Average performance
  - Worst case performance
  - Energy/power consumption
  - Thermal behavior
  - Reliability 
  - Electromagnetic compatibility
  - Numeric precision
  - Testability
  - Cost
  - Weight, robustness, usability, extendibility, security, safety, environmental friendliness

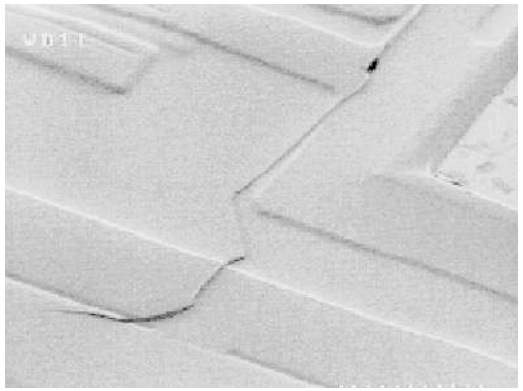
Impact of Shrinking Feature Sizes

- Reduced reliability due to smaller transistors and wires [ITRS, 2009]
  - Transient, intermittent, permanent faults
- Rate of faults expected to increase such that *all* designs need to become fault-tolerant

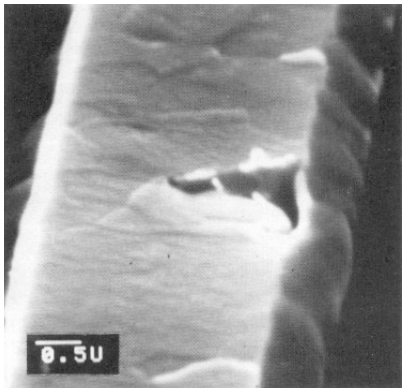
Electromigration



Thermal Cycling



Stress migration



[Source: JEDEC]

FIT and "10^{-9}"

- Reliability target: < 10^{-9} failures per hour
  - Equivalent to one failure per 10,000 hours per 100,000 systems
- FIT: failures-in-time, a rate of failure
  - =1/MTTF≈1/MTBF
- 1 FIT: a rate of 10^{-9} failures per hour

Terms

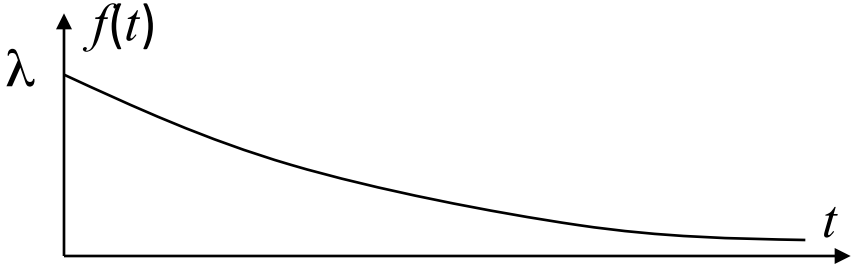
- *"A **service failure**, often abbreviated here to **failure**, is an event that occurs when the delivered service of a system deviates from the correct service."*
- *"The definition of an **error** is the part of the total state of the system that may lead to its subsequent service failure."*
- *"The adjudged or hypothesized cause of an error is called a **fault**. Faults can be internal or external of a system."*
- Example:
  - Transient **fault** flipping a bit in memory
  - After this bit flip, the memory cell will be in **error**
  - **Failure**: if the system service is affected by this error
- We will consider **failure** rates and **fault** models

[Laprie et al., 1992, 2004]

Reliability: $f(t)$, $F(t)$

- Let T: time until first failure (random variable)
- Let $f(t)$ be the density function of T
- Example: Exponential distribution

$$f(t)=\lambda e^{-\lambda t}$$

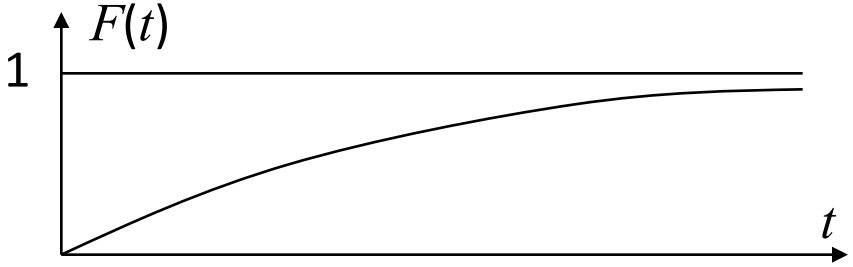


- $F(t)$ = probability of the system being faulty at time t:

$$F(t) = \Pr(T \leq t) \qquad F(t) = \int_0^t f(x)dx$$

- Example: Exponential distribution

$$F(t) = \int_0^t \lambda e^{-\lambda x} dx = -[e^{-\lambda x}]_0^t = 1 - e^{-\lambda t}$$



Reliability: $R(t)$

- **Reliability** $R(t)$ = probability that the time until the first failure is larger than some time t:

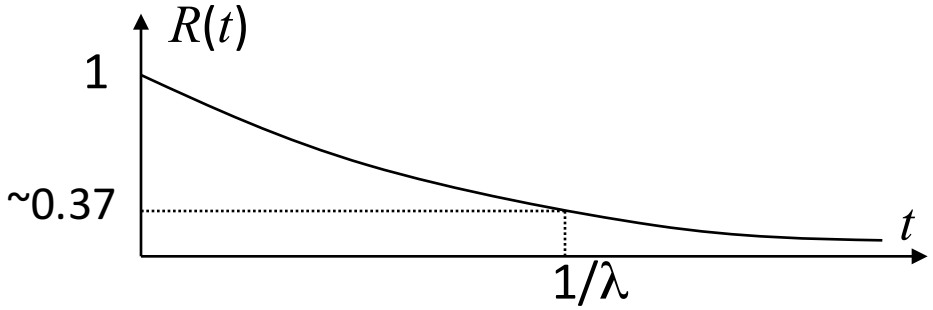
$R(t)=\Pr(T>t),\ t\geq 0$ $\qquad R(t)=\int_t^{\infty} f(x)dx$

$$F(t)+R(t)=\int_0^t f(x)dx+\int_t^{\infty} f(x)dx=1$$

$$R(t)=1-F(t) \qquad f(t)=-\frac{dR(t)}{dt}$$

Example: Exponential distribution

$R(t)=e^{-\lambda t}$



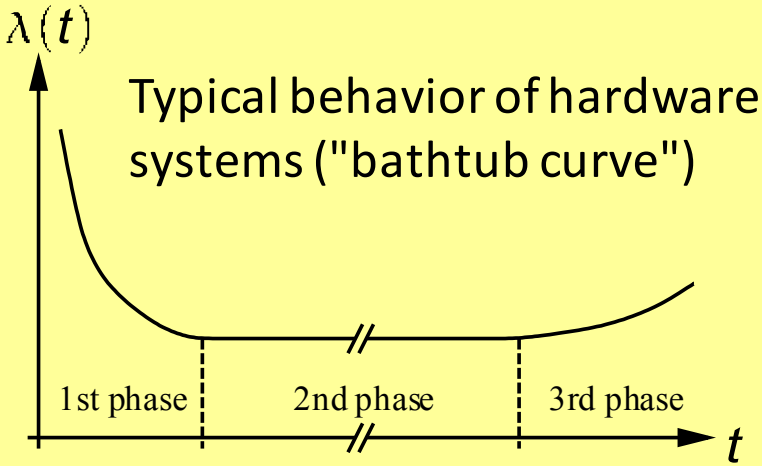
Failure Rate

The failure rate at time t is the probability of the system failing between time t and time $t+\Delta t$:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{\Pr(t < T \le t + \Delta t \mid T > t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{F(t+\Delta t) - F(t)}{\Delta t R(t)} = \frac{f(t)}{R(t)}$$

Conditional probability ("provided that the system works at t");

$\Pr(A|B) = \Pr(AB)/\Pr(B)$



For exponential distribution:

$$\frac{f(t)}{R(t)} = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \lambda$$

FIT = expected number of failures in 10^9 hrs.

MTTF = $E\{T\}$, the *statistical mean* of T

$$\text{MTTF} = E\{T\} = \int_0^\infty t \cdot f(t)\, dt$$

According to the definition of the statistical mean value

Example: Exponential distribution

$$\text{MTTF}_{\text{exp}} = \int_0^\infty t \cdot \lambda e^{-\lambda t} dt = -\left[t \cdot e^{-\lambda t}\right]_0^\infty + \int_0^\infty e^{-\lambda t} dt$$

$$\int u \cdot v' = u \cdot v - \int u' \cdot v$$

$$\text{MTTF}_{\text{exp}} = -\frac{1}{\lambda}\left[e^{-\lambda t}\right]_0^\infty = -\frac{1}{\lambda}[0 - 1] = \frac{1}{\lambda}$$

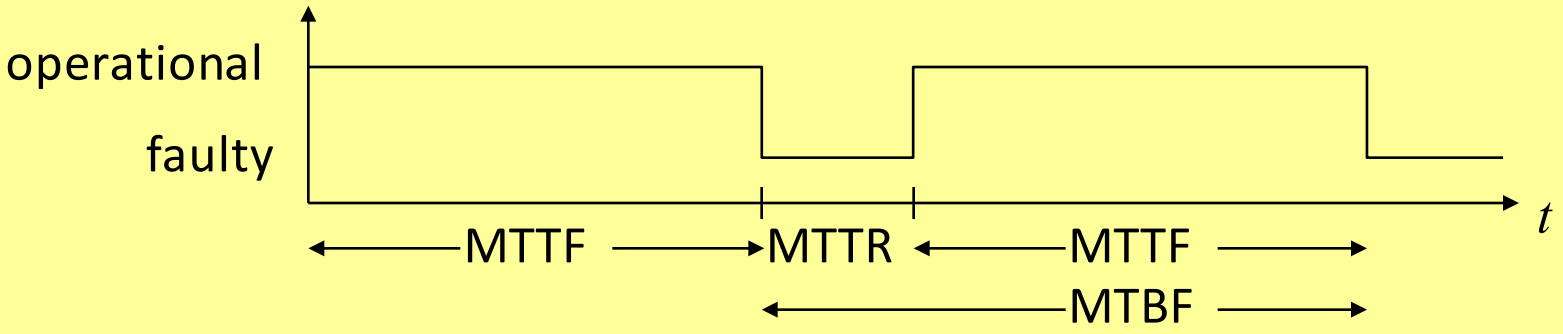
MTTF is the reciprocal of the failure rate

MTTF, MTTR and MTBF

MTTR = mean time to repair
(average over repair times using distribution $M(d)$)
MTBF* = mean time between failures = MTTF + MTTR

$$\text{Availability } A = \lim_{t \to \infty} A(t) = \frac{\text{MTTF}}{\text{MTBF}}$$

Ignoring the statistical nature of failures …



* Mixed up with MTTF, if starting in operational state is implicitly assumed

Dependability Requirements

- Allowed failures may be in the order of one per 10^9 h
  - ~ 1000 times less than typical failure rates of chips

⇒ For safety-critical systems, the system must be more dependable than any of its parts

⇒ Fault-tolerance mechanisms must be used!

- Low acceptable failure rate → systems are not 100% testable

⇒ Safety must be shown by a combination of testing and reasoning

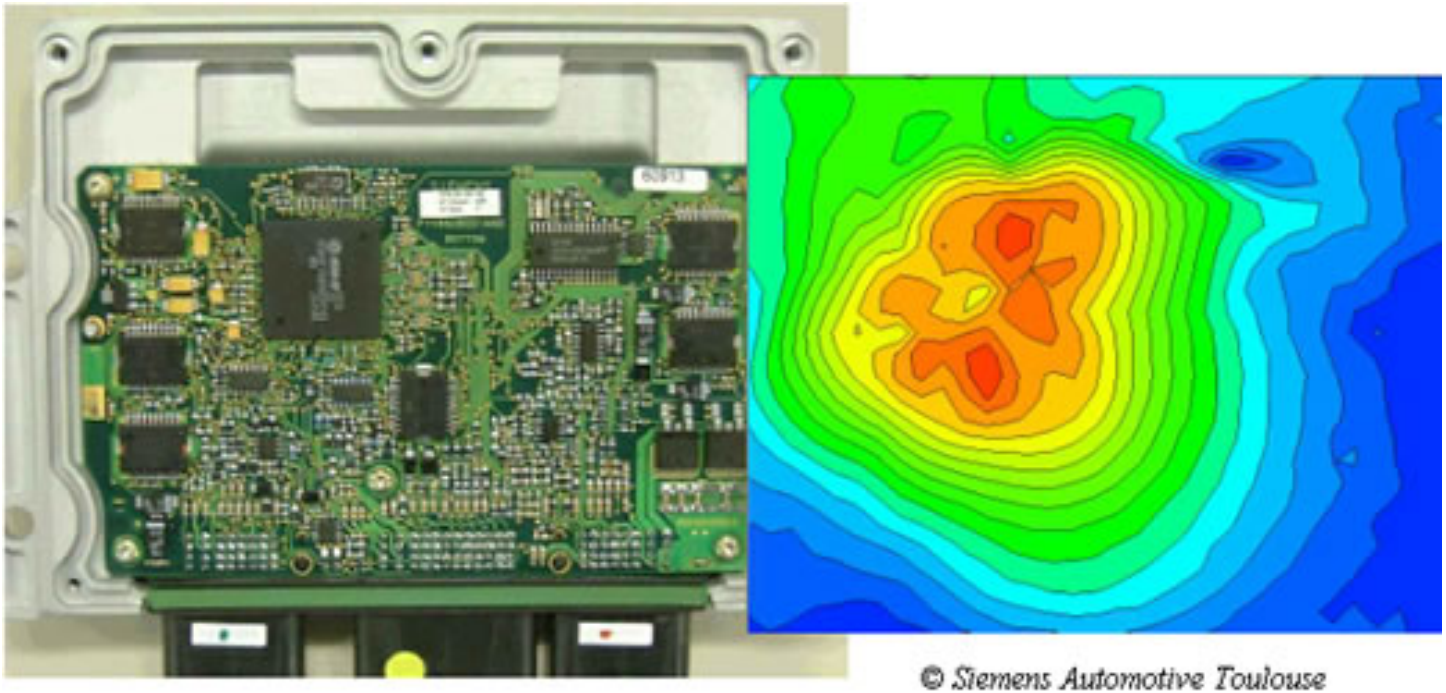
  - Abstraction must be used to make the system explainable using a hierarchical set of behavioral models
  - Design faults and human failures must be taken into account

Multi-Objective Design

- Different design objectives/criteria are relevant:
  - Average performance
  - Worst case performance
  - Energy/power consumption
  - Thermal behavior
  - Reliability
  - Electromagnetic compatibility 
  - Numeric precision
  - Testability
  - Cost
  - Weight, robustness, usability, extendibility, security, safety, environmental friendliness

Electro-magnetic compatibility (EMC)

Example: car engine controller



© Siemens Automotive Toulouse

Red: high emission; Validation of EMC properties often done at the end of the design phase.

Source: http://intrage.insa-tlse.fr/~etienne/emccourse/what_for.html

Design Simulation

- Simulations try to imitate the behavior of the real system on a (typically digital) computer
- Simulating functional behavior requires executable models (*e.g.*, Chapter 2)
- Simulations can be performed at various levels
  - System, architecture, microarchitecture, circuit ...
- Non-functional properties can also be simulated
  - *E.g.* temperature, EMC
- Simulations can be used for evaluation, validation

Validating Functionality by Simulation

- Various levels of abstractions are used:
  - High-level of abstraction
    - Fast, but sometimes not accurate
  - Lower level of abstraction
    - Slow and typically accurate
- Choosing a level is always a compromise
  - Can perform many high-level simulations
  - Can only perform a few low-level simulations

Limits of Simulation

- Typically slower than the actual design
  - Timing constraints can be violated if simulator is connected to the actual environment
- Simulations in the real environment may be dangerous
  - Would you want to "simulate" anti-lock braking?
- Huge input data sets may make simulation impossible
- Most actual systems are too complex to allow simulating all possible cases (inputs)
  - Simulations can help find errors in designs,
  - But they cannot guarantee the absence of errors!

Rapid Prototyping/Emulation

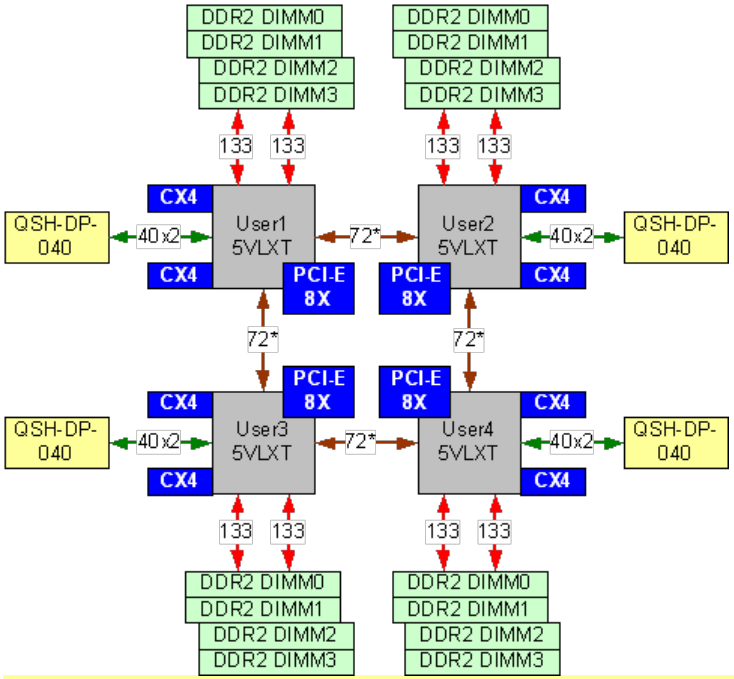
- *Prototype*: An embedded system that
  - Can be generated quickly, and
  - Behaves very similarly to the final product
- May be
  - Larger,
  - Consume more power,
  - Have other undesirable properties that are acceptable at the validation phase

Emulation

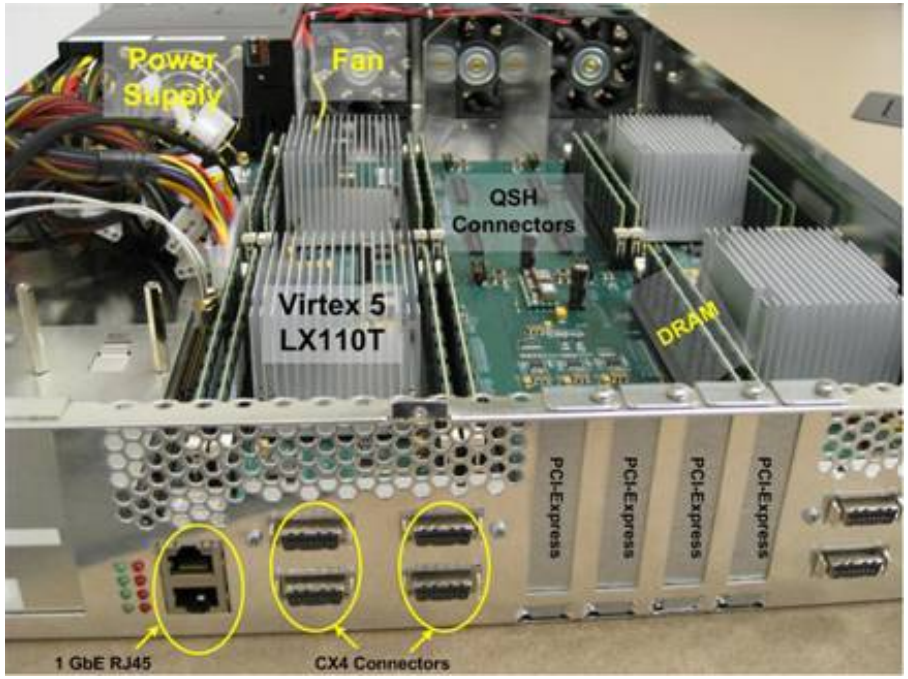
- Simulations are based on models that are at best approximations of real systems
  - In general: ∃ difference between the real system and the model of the real system
  - Goal: Reduce gap by implementing parts of SUD more precisely!
- **Definition**: *Emulation* is the process of executing a model of the SUD where at least one component is not represented by simulation on some kind of host computer

State-of-the-art Emulation (1)

- Berkeley Emulation Engine 3
- Expandable FPGA system
  - 1-64 4-FPGA 2U blades

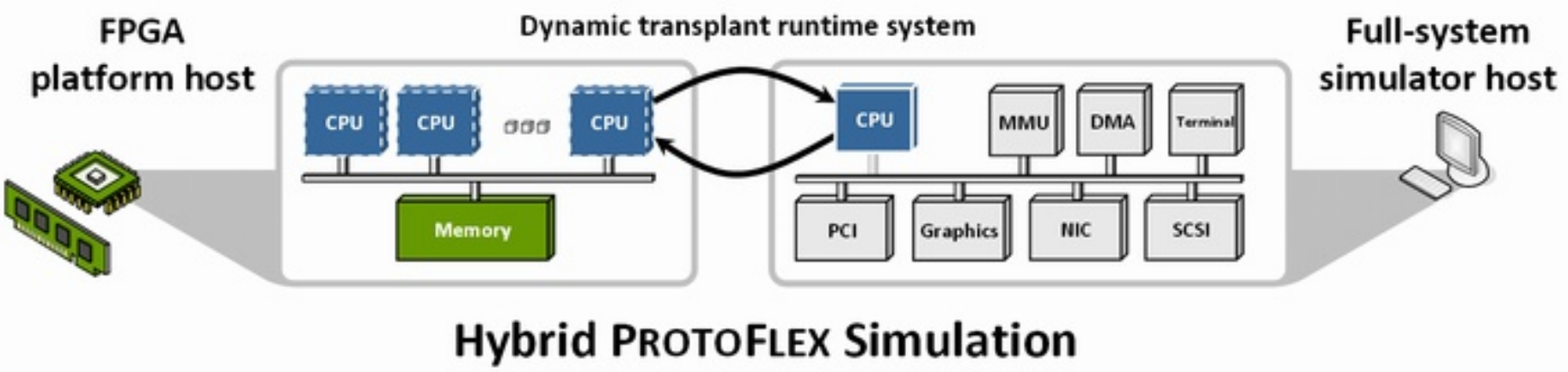


Source: http://research.microsoft.com/en-us/projects/bee3/



State-of-the-art Emulation (2)

- ProtoFlex
- Hybrid emulation platform
  - Common-case code is “transplanted” to the FPGAs
  - Uncommon behavior (I/O) executes on the host



Formal Verification

- Formal verification = formally proving a system correct
  - Using mathematics
- Requires a formal model
  - Cannot be automatically generated from another model
- Given a formal model, try to prove properties
- Even a formally verified system can fail
  - *E.g.* if assumptions are not satisfied

- Formal techniques are classified by the type of logic employed
- **Ideally**: Formally verified tools transforming specifications into implementations (*correctness by construction*)
- **In practice**: Unverified tools and manual design steps
  - Validation of each and every design is required
  - Has to be done at intermediate steps, too; not just the final design
  - Therefore requires significant effort!

Propositional Logic (1)

- Consisting of Boolean formulas comprising Boolean variables and connectives such as ∨ and ∧
- Gate-level logic networks can be described
- *Typical aim*: checking if two models are equivalent
  - Called tautology checkers or equivalence checkers
- Propositional logic is *decidable*, so it is
  - Decidable whether two representations are equivalent
- Tautology checkers
  - Can cope with large designs
  - An alternative to validation by exhaustive simulation

Propositional Logic (2)

- Source of the power of tautology checkers:
  - Binary Decision Diagrams (BDDs)
  - A tree representation of a logic function
- Complexity of equivalence checks of Boolean functions represented with BDDs: O(number of BDD-nodes)
  - Whereas equivalence check for sums of products is NP-hard
- Many functions can be efficiently represented with BDDs. In general, however, the #(nodes) of BDDs grows exponentially with the number of variables.
- Simulators frequently replaced by equivalence checkers if functions can be efficiently represented with BDDs.
- Limited ability to verify FSMs

First Order Logic (FOL)

- FOL includes quantification, using ∃ and ∀
- Some automation for verifying FOL is feasible
- However, since FOL is undecidable in general
  - There may therefore be cases of doubt

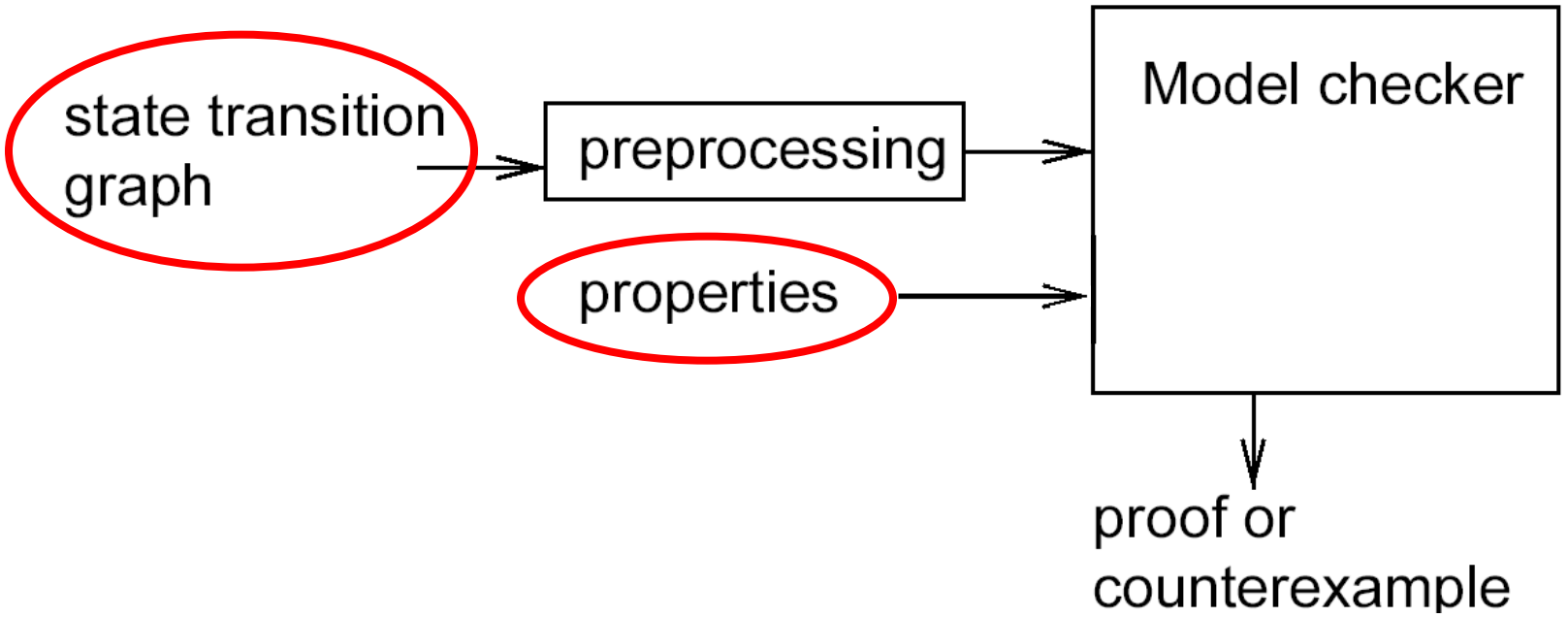
Higher Order Logic (HOL)

- Higher order logic allows functions to be manipulated like other objects
- Proofs can hardly ever be automated
  - Instead, proofs are done manually
  - Proof-support may be available

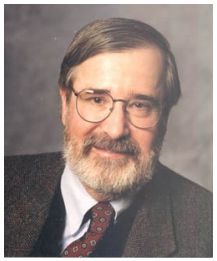
Model Checking

- Goal: verification of finite state systems
- Analyzes the state space of the system
- Verification in three stages:
  - Model generation
  - Property definition
  - Model checking (the actual verification step)

Two Types of Input



Verification tools can prove or disprove the properties.
In the latter case, they can provide a counter-example.
Example: Clarke's EMC-system



Examples

1. $M,s \models AGg$
   means:
   in the transition graph M, property g holds for all paths (denoted by A) and all states (denoted by G).
2. For the Thalys example (Petri Nets), we could prove that the number of trains is indeed constant.

Computational Properties

- Model checking is easier to automate than FOL
- History
  - In 1987, model checking was implemented using *binary decision diagrams* (BDDs)
  - It was possible to locate several errors in the specification of the *future bus* protocol
  - Model checking since has become very popular
- Extensions are needed in order to also cover real-time behavior and data values

Summary

- Design Objectives
  - Energy and Power
  - Temperature
  - Reliability
  - Electromagnetic Compatibility
- Validation
  - Simulation
  - Emulation
  - Formal Verification

Next Time

- Application Mapping
  - Introduction
  - Aperiodic task scheduling
  - Chapters 6.1-2