technische universität
dortmund

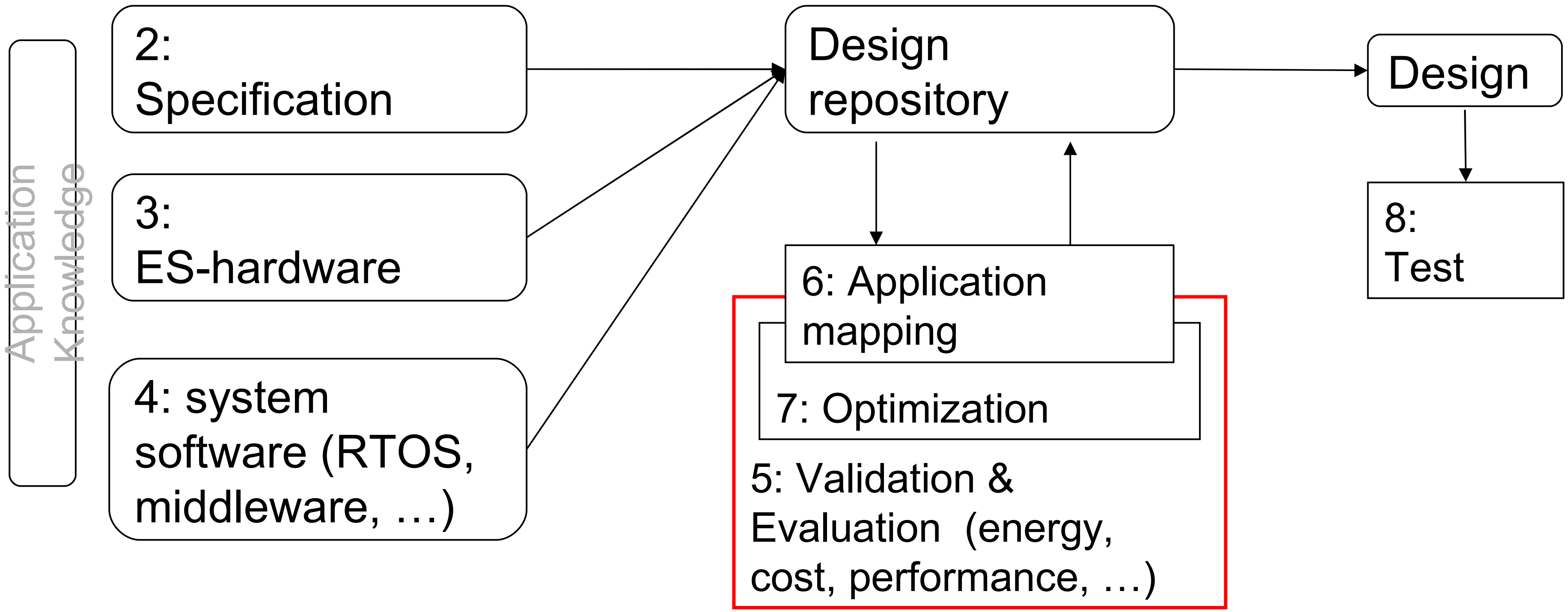
fakultät für informatik
informatik 12

Evaluation and Validation

Peter Marwedel
TU Dortmund, Informatik 12
Germany

2009/11/27

Structure of this course



Numbers denote sequence of chapters

Validation and Evaluation

Definition: *Validation* is the process of checking whether or not a certain (possibly partial) design is appropriate for its purpose, meets all constraints and will perform as expected (yes/no decision).

Definition: Validation with mathematical rigor is called *(formal) verification*.

Definition: *Evaluation* is the process of computing quantitative information of some key characteristics of a certain (possibly partial) design.

How to evaluate designs according to multiple criteria?

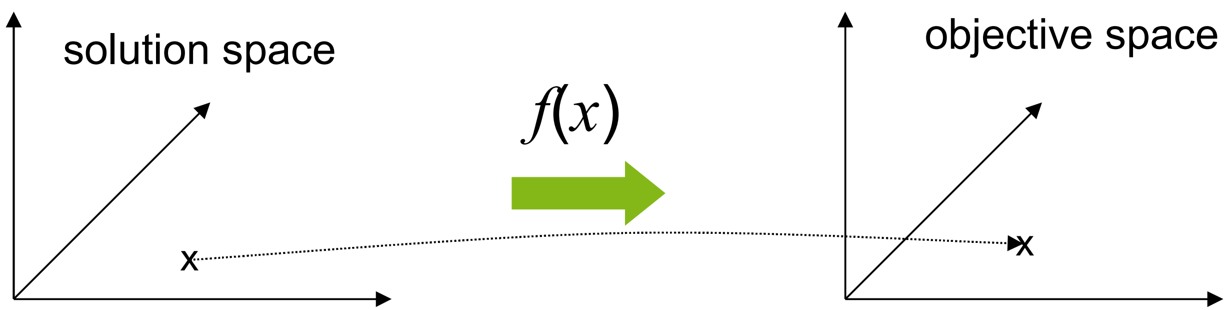
In practice, many different criteria are relevant for evaluating designs:

- (average) speed
- worst case speed
- power consumption
- cost
- size
- weight
- radiation hardness
- environmental friendliness ….

How to compare different designs?
(Some designs are “better” than others)

Definitions

- Let X: m-dimensional **solution space** for the design problem. Example: dimensions correspond to # of processors, size of memories, type and width of busses etc.
- Let F: n-dimensional **objective space** for the design problem. Example: dimensions correspond to speed, cost, power consumption, size, weight, reliability, …
- Let $f(x)=(f_1(x),\ldots,f_n(x))$ where $x \in X$ be an **objective function.** We assume that we are using $f(x)$ for evaluating designs.



Pareto points

- We assume that, for each objective, a total order < and the corresponding order ≤ are defined.
- **Definition**:
  Vector $u=(u_1,\ldots,u_n)\in F$ **dominates** vector $v=(v_1,\ldots,v_n)\in F$

  $\Leftrightarrow$

  u is "better" than v with respect to one objective and not worse than v with respect to all other objectives:

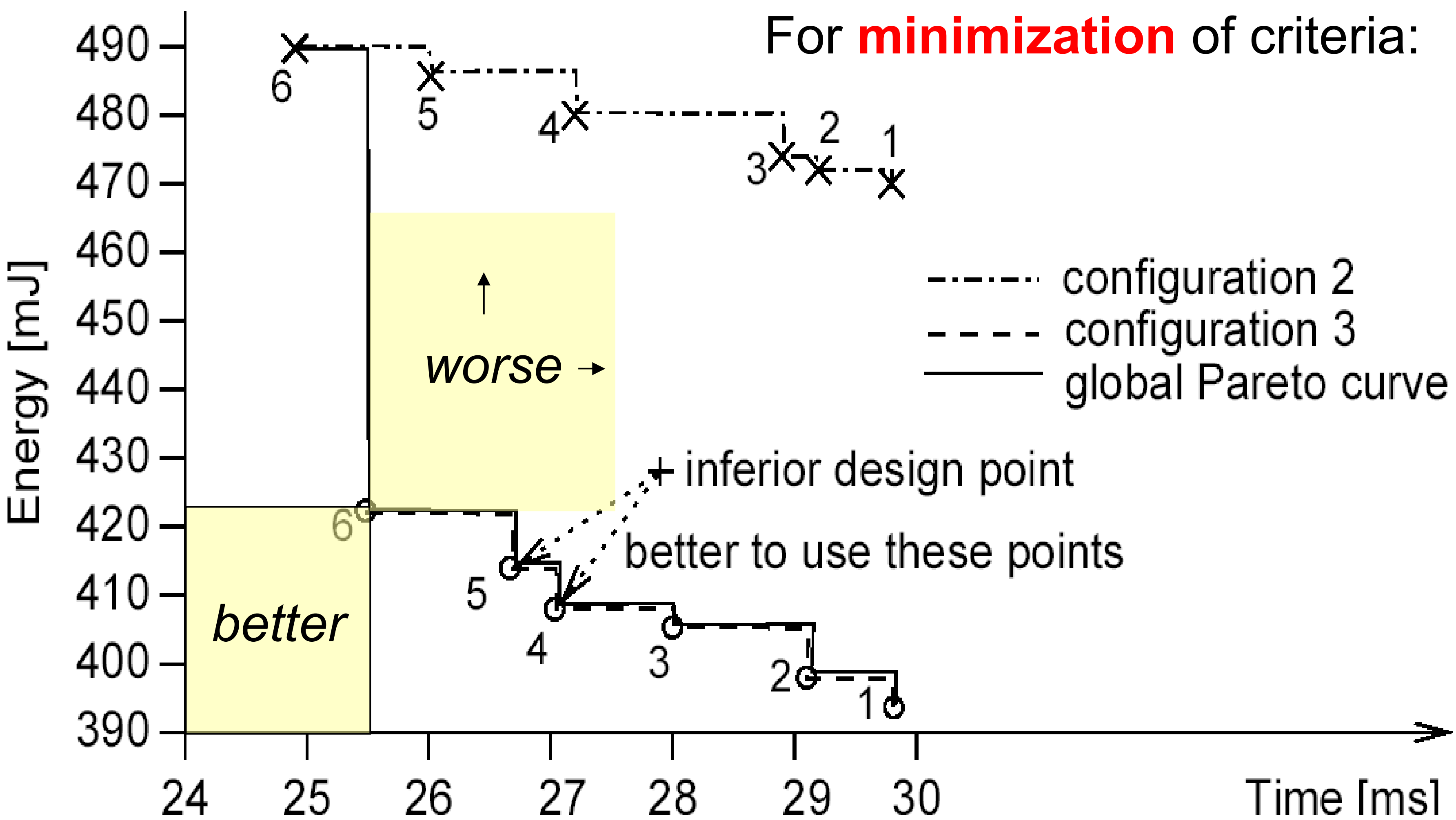
  $$\forall i\in\{1,\ldots,n\}: u_i \leq v_i \wedge$$
  $$\exists i\in\{1,\ldots,n\}: u_i < v_i$$

- **Definition:**
  Vector $u\in F$ is **indifferent** with respect to vector $v\in F$
  $\Leftrightarrow$ neither u dominates v nor v dominates u

Pareto points

- A solution $x \in X$ is called **Pareto-optimal** with respect to X
  $\Leftrightarrow$
  there is no solution $y \in X$ such that $u=f(x)$ is dominated by $v=f(y)$
- **Definition**: Let $S \subseteq F$ be a subset of solutions.
  v is called a **non-dominated solution** with respect to S
  $\Leftrightarrow$ v is not dominated by any element $\in S$.
- v is called **Pareto-optimal**
  $\Leftrightarrow$ v is non-dominated with respect to all solutions F.

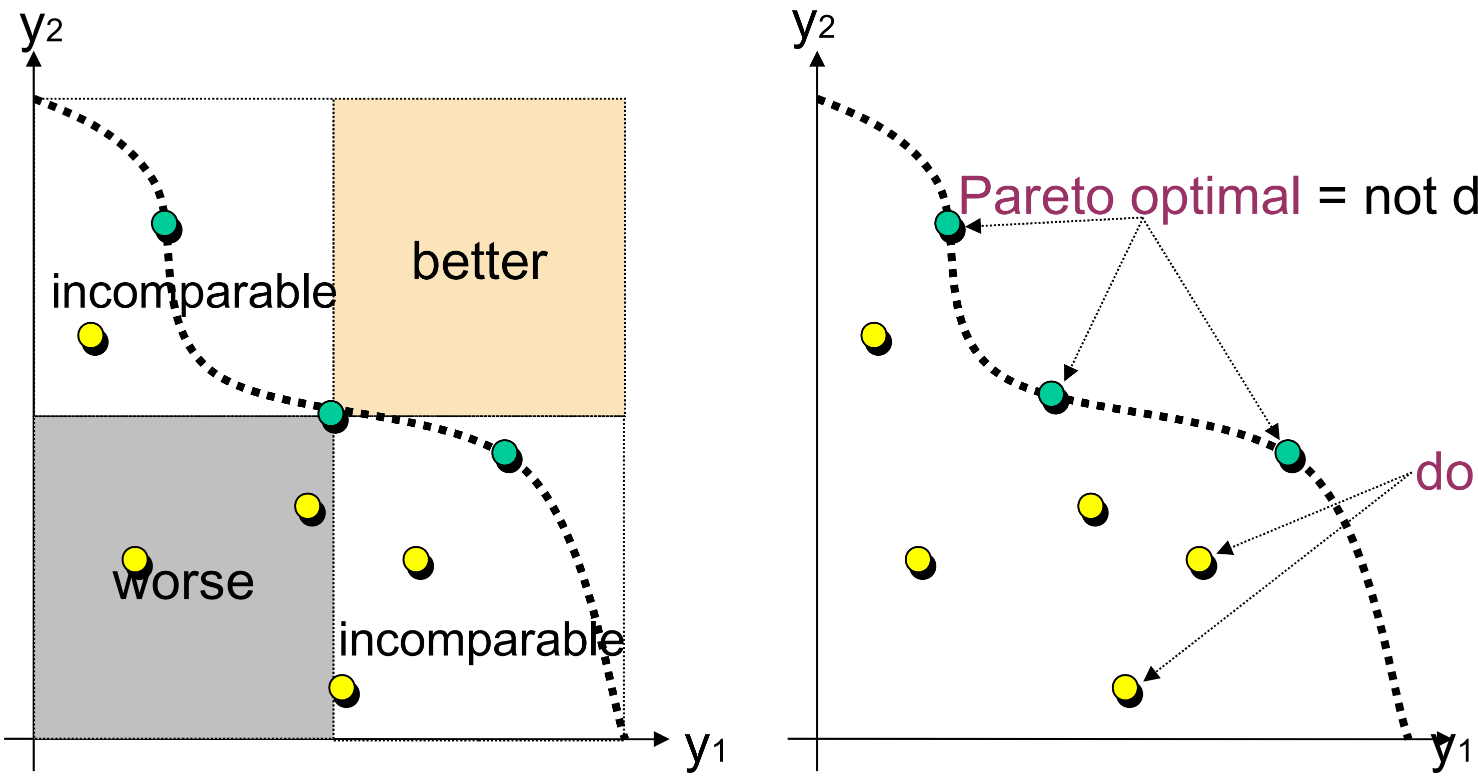
Pareto Points



Multiobjective Optimization

For **maximization** of criteria:

Maximize $(y_1, y_2, \ldots, y_k) = f(x_1, x_2, \ldots, x_n)$



Pareto set = set of all Pareto-optimal solutions

Design space evaluation

Design space evaluation (DSE) based on Pareto-points is the process of finding and returning a set of Pareto-optimal designs to the user, enabling the user to select the most appropriate design.

Simulations

- Simulations try to imitate the behavior of the real system on a (typically digital) computer.
- Simulation of the functional behavior requires executable models.
- Simulations can be performed at various levels.
- Some non-functional properties (e.g. temperatures, EMC) can also be simulated.
- Simulations can be used to **evaluate** and to **validate** a design

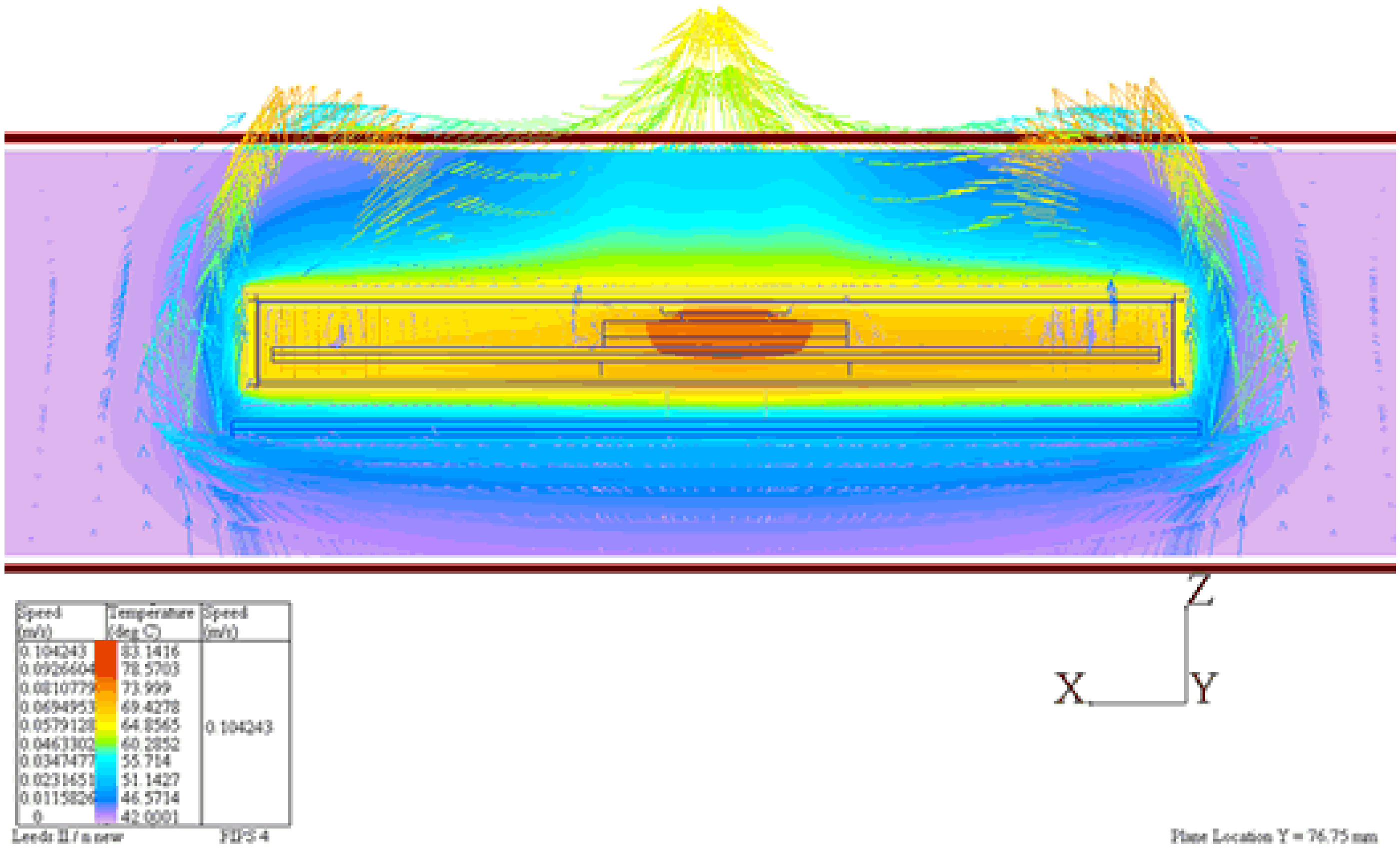
Validating functional behavior by simulation

Various levels of abstractions used for simulations:

- High-level of abstraction: fast, but sometimes not accurate
- Lower level of abstraction: slow and typically accurate
- Choosing a level is always a compromise

Non-functional behavior: Examples of thermal simulations (1)

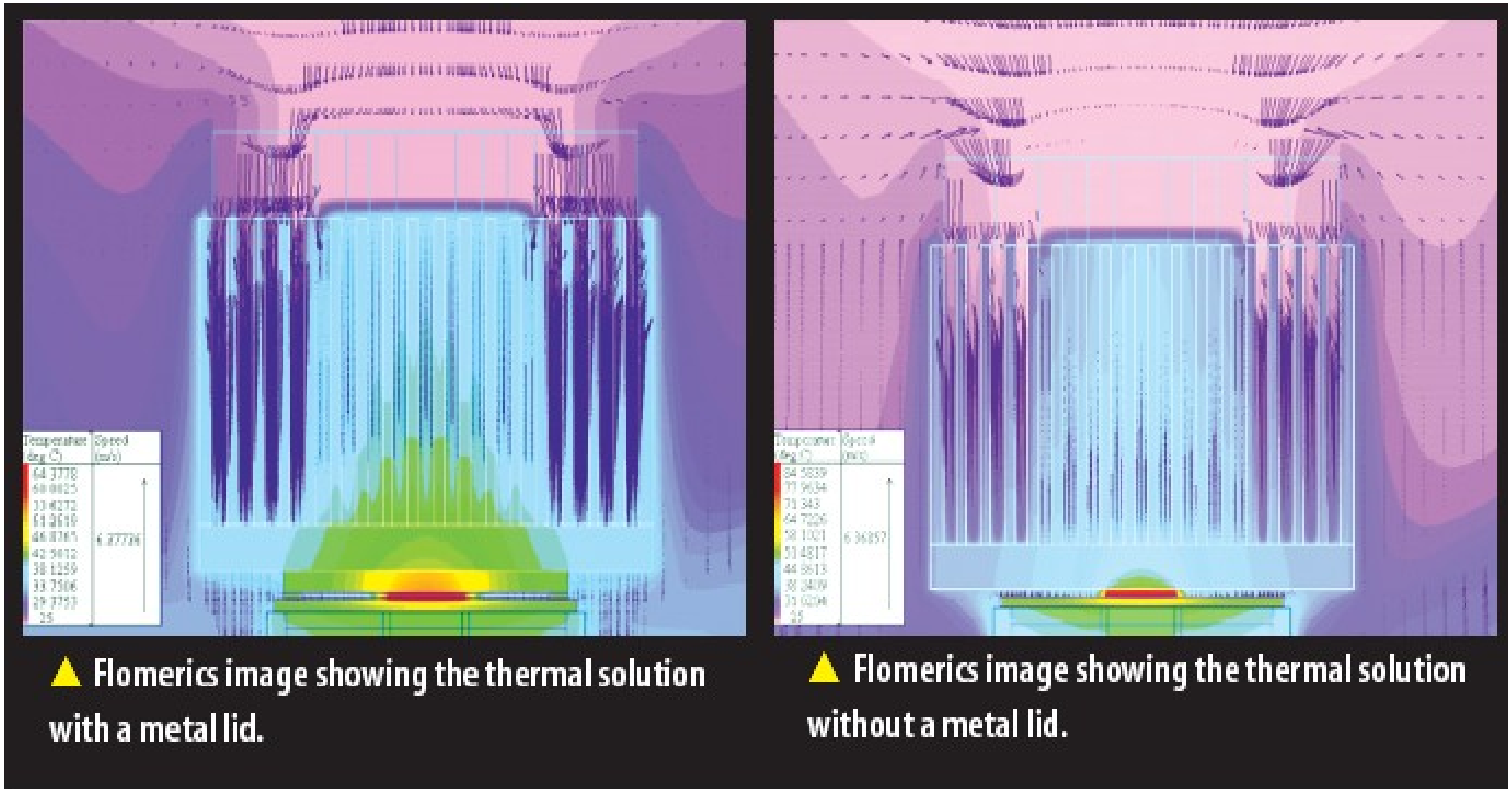
Encapsulated cryptographic coprocessor:



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

Examples of thermal simulations (2)

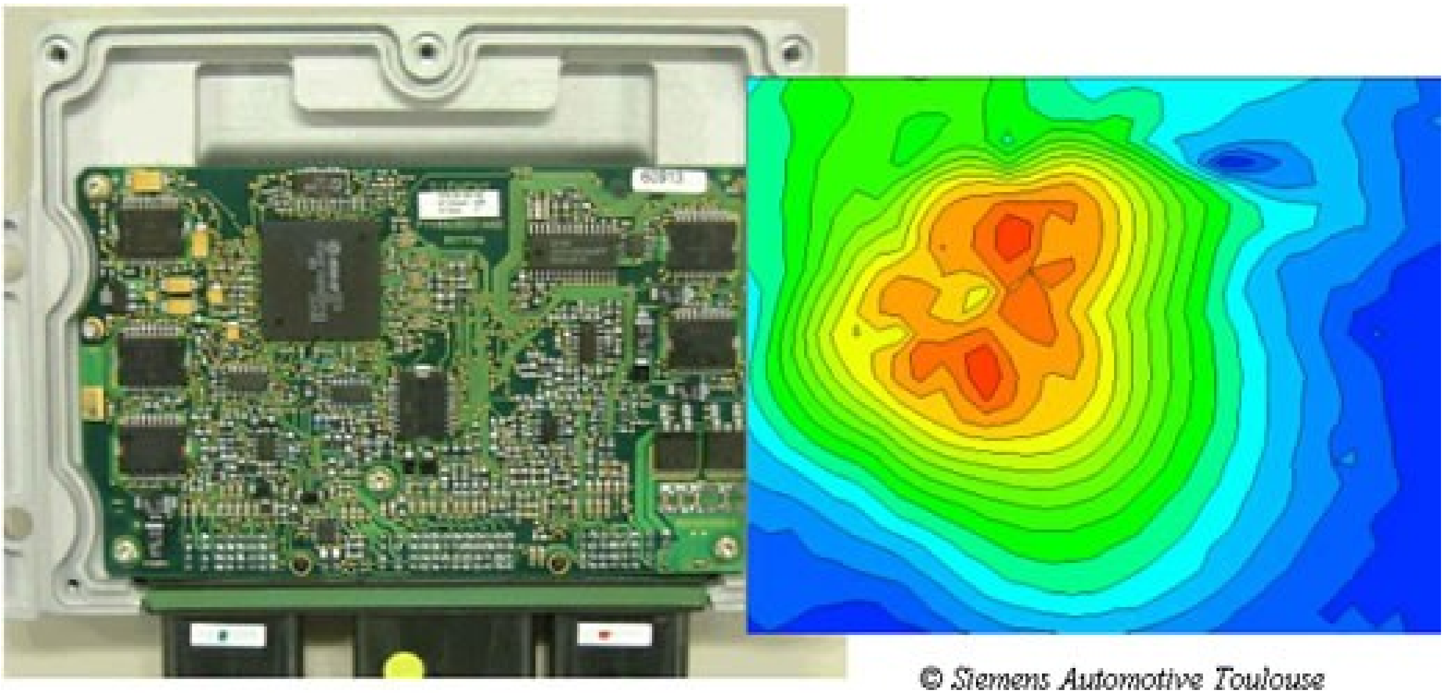
Microprocessor



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

EMC simulation

Example: car engine controller



© Siemens Automotive Toulouse

Red: high emission

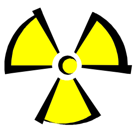
Validation of EMC properties often done at the end of the design phase.

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

Simulations
Limitations

- Typically slower than the actual design.
  ☞ **Violations of timing constraints** likely if simulator is connected to the actual environment

- Simulations in the real environment may be **dangerous**



- There may be huge amounts of data and it may be impossible to simulate enough data in the available time.

- Most actual systems are too complex to allow simulating all possible cases (inputs).
  Simulations can help finding errors in designs, but they cannot guarantee the absence of errors.

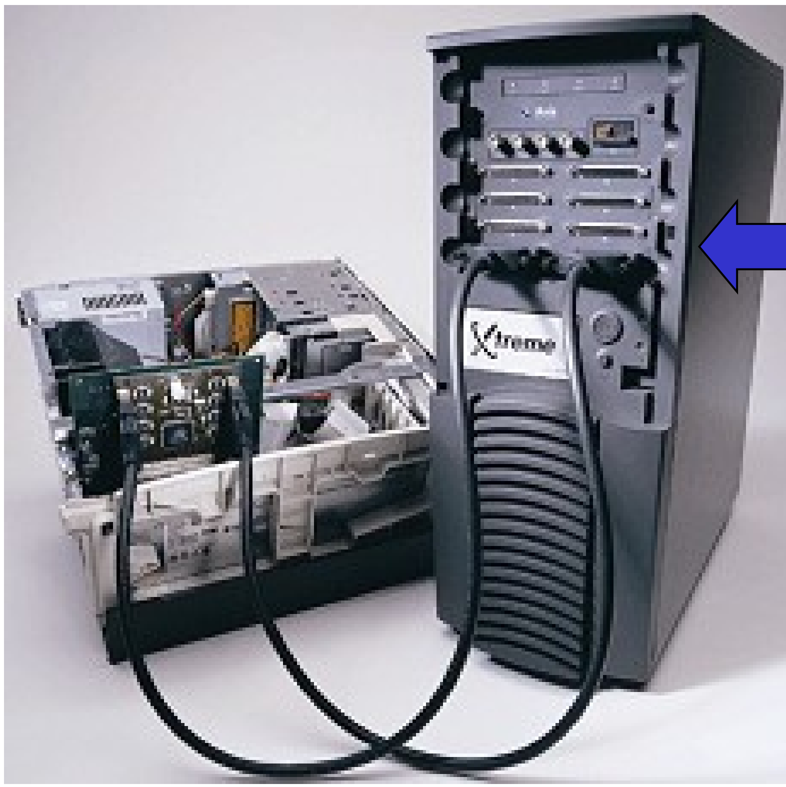
Rapid prototyping/Emulation

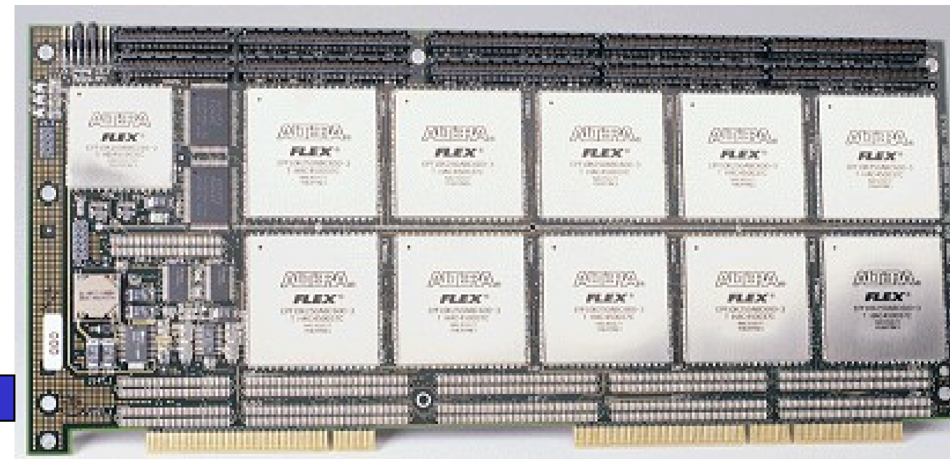
- Prototype: Embedded system that can be generated quickly and behaves very similar to the final product.
- May be larger, more power consuming and have other properties that can be accepted in the validation phase
- Can be built, for example, using FPGAs.

Example: Quickturn Cobalt System (1997), ~0.5M$ for 500kgate entry level system

Source & ©: http://www. eedesign. com/editorial/1997/ toolsandtech9703.html

Example of a more recent commercial emulator





[www.verisity.com/images/products/xtremep{1|3}.gif]

Summary

Evaluation and Validation

- In general, multiple objectives
- Pareto optimality
- Design space evaluation (DSE)
- Simulations
- Rapid prototyping