technische universität
dortmund

fakultät für informatik
informatik 12

Evaluation and Validation

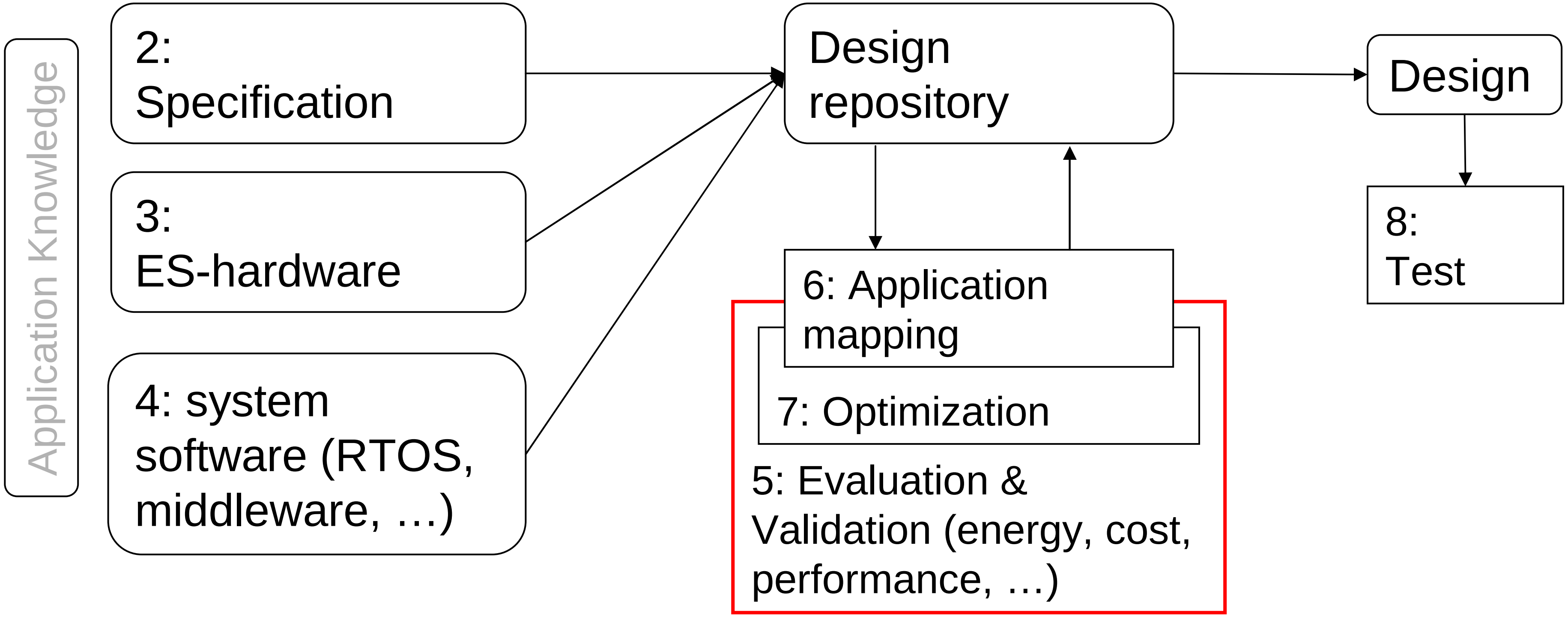
Peter Marwedel
TU Dortmund, Informatik 12
Germany

Graphics: © Alexandra Nolte, Gesine Marwedel, 2003

2010年 12 月 05 日

These slides use Microsoft clip arts.
Microsoft copyright restrictions apply.

Structure of this course



Numbers denote sequence of chapters

Evaluation of designs according to multiple objectives

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 models

$$E = \int P dt$$

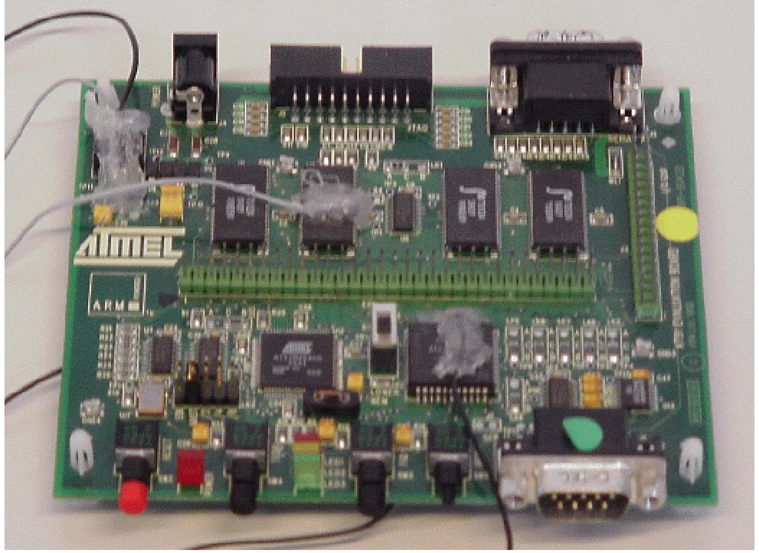
Power/energy models becoming increasingly important

- due to mobile computing,
- since energy availability becomes more relevant due to increased performance, and
- due to environmental issues.

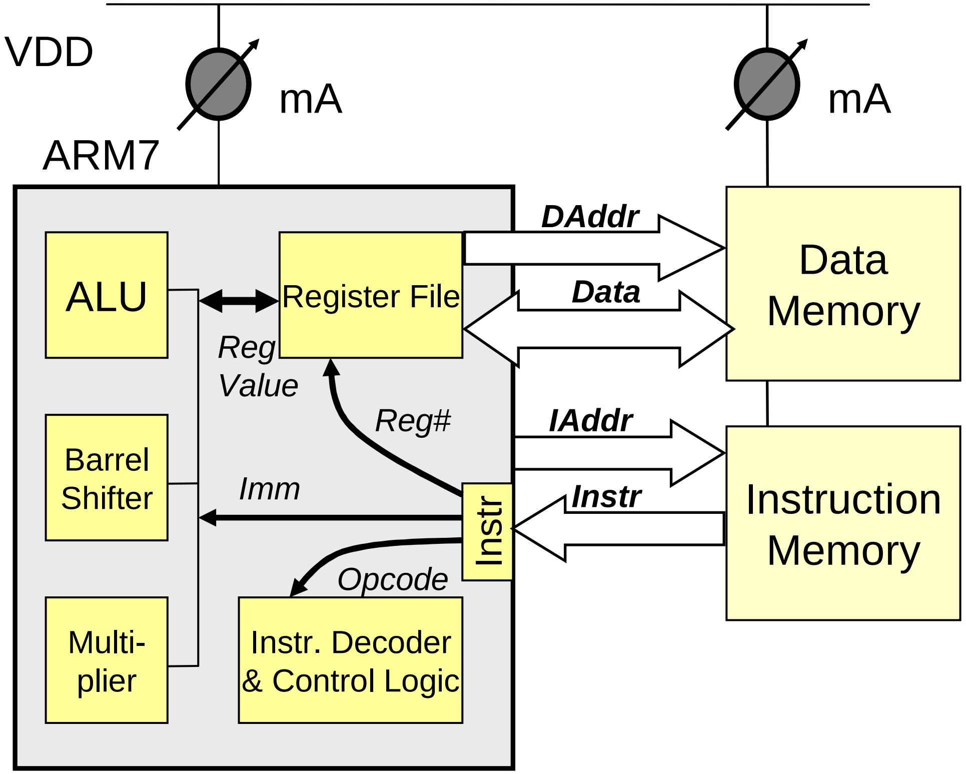
Energy models

- 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): Measurements for 2 fixed configurations
- Steinke et al., UniDo (2001): mixed model using measurements and prediction
- CACTI [Jouppi, 1996]: Predicted energy consumption of caches
- Wattch [Brooks, 2000]: Power estimation at the architectural level, without curcuit or layout

Steinke’s & 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: Instruction dependent costs in the CPU

Cost for a sequence of *m* instructions

$$
\begin{aligned}
E_{cpu\_instr} = {} & \sum \mathrm{MinCostCPU}(\mathit{Opcode}_i) + \mathrm{FUCost}(\mathit{Instr}_{i-1},\mathit{Instr}_i) \\
& \alpha_1 * \sum \mathrm{w}(\mathit{Imm}_{i,j}) + \text{ß}_1 * \sum \mathrm{h}(\mathit{Imm}_{i-1,j}, \mathit{Imm}_{i,j}) + \\
& \alpha_2 * \sum \mathrm{w}(\mathit{Reg}_{i,k}) + \text{ß}_2 * \sum \mathrm{h}(\mathit{Reg}_{i-1,k}, \mathit{Reg}_{i,k}) + \\
& \alpha_3 * \sum \mathrm{w}(\mathit{RegVal}_{i,k}) + \text{ß}_3 * \sum \mathrm{h}(\mathit{RegVal}_{i-1,k}, \mathit{RegVal}_{i,k}) + \\
& \alpha_4 * \sum \mathrm{w}(\mathit{IAddr}_i) + \text{ß}_4 * \sum \mathrm{h}(\mathit{IAddr}_{i-1}, \mathit{IAddr}_i)
\end{aligned}
$$

w: number of ones;

h: Hamming distance;

FUCost: cost of switching functional units

α, ß: determined through experiments

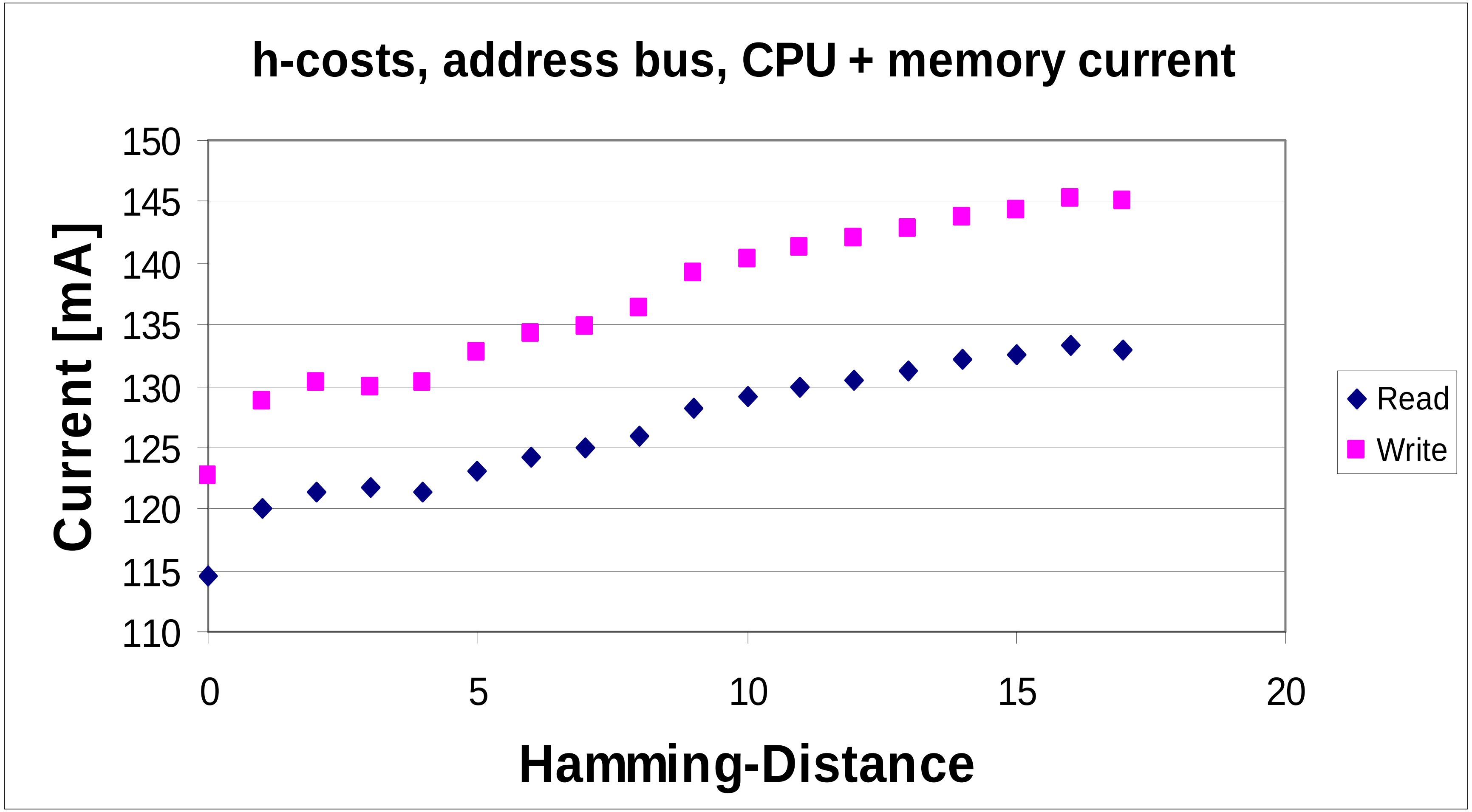
Other costs

$$E_{cpu\_data} = \sum \alpha_5 * w(\mathit{DAddr}_i) + ß_5 * h(\mathit{DAddr}_{i-1}, \mathit{DAddr}_i) + \alpha_6 * w(\mathit{Data}_i) + ß_6 * h(\mathit{Data}_{i-1}, \mathit{Data}_i)$$

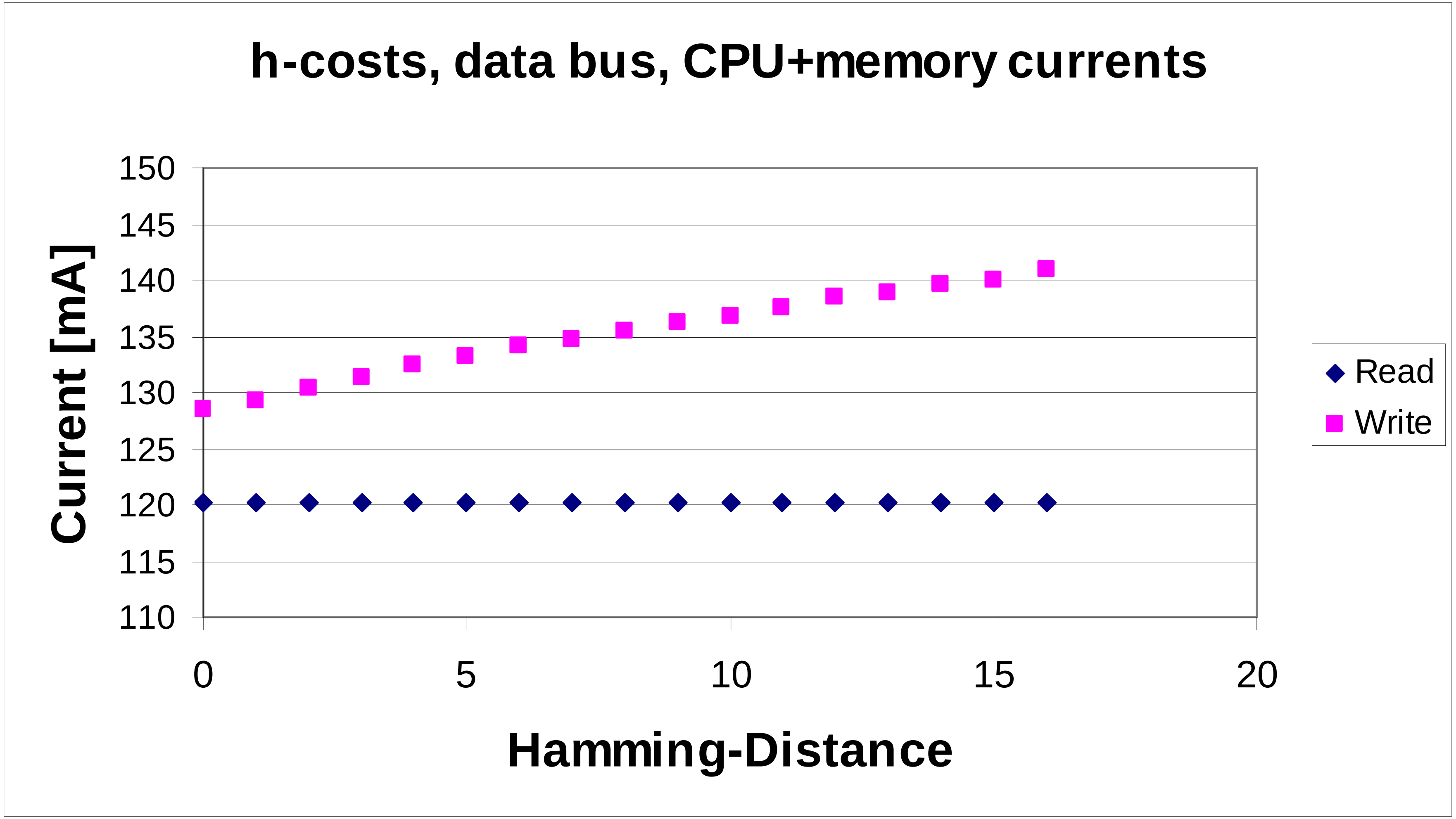
$$E_{mem\_instr} = \sum \mathrm{MinCostMem}(\mathit{InstrMem}, \mathit{Word\_width}_i) + \alpha_7 * w(\mathit{IAddr}_i) + ß_7 * h(\mathit{IAddr}_{i-1}, \mathit{IAddr}_i) + \alpha_8 * w(\mathit{IData}_i) + ß_8 * h(\mathit{IData}_{i-1}, \mathit{IData}_i)$$

$$E_{mem\_data} = \sum \mathrm{MinCostMem}\ (\mathit{DataMem}, \mathit{Direction}, \mathit{Word\_width}_i) + \alpha_9 * w(\mathit{DAddr}_i) + ß_9 * h(\mathit{DAddr}_{i-1}, \mathit{DAddr}_i) + \alpha_{10} * w(\mathit{Data}_i) + ß_{10} * h(\mathit{Data}_{i-1}, \mathit{Data}_i)$$

The Hamming Distance between adjacent addresses is playing a major role



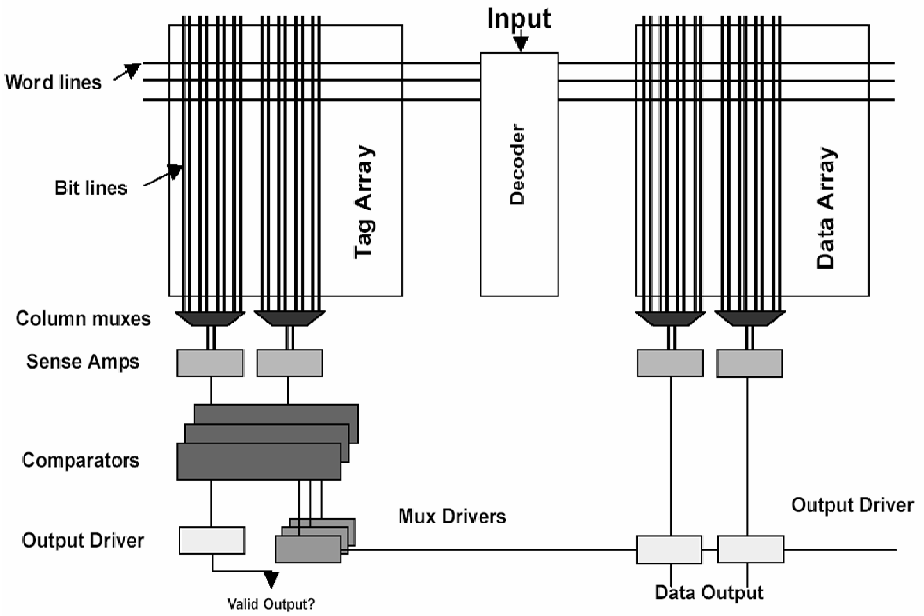
The Hamming Distance between adjacent values on the data bus is playing a major role



Results

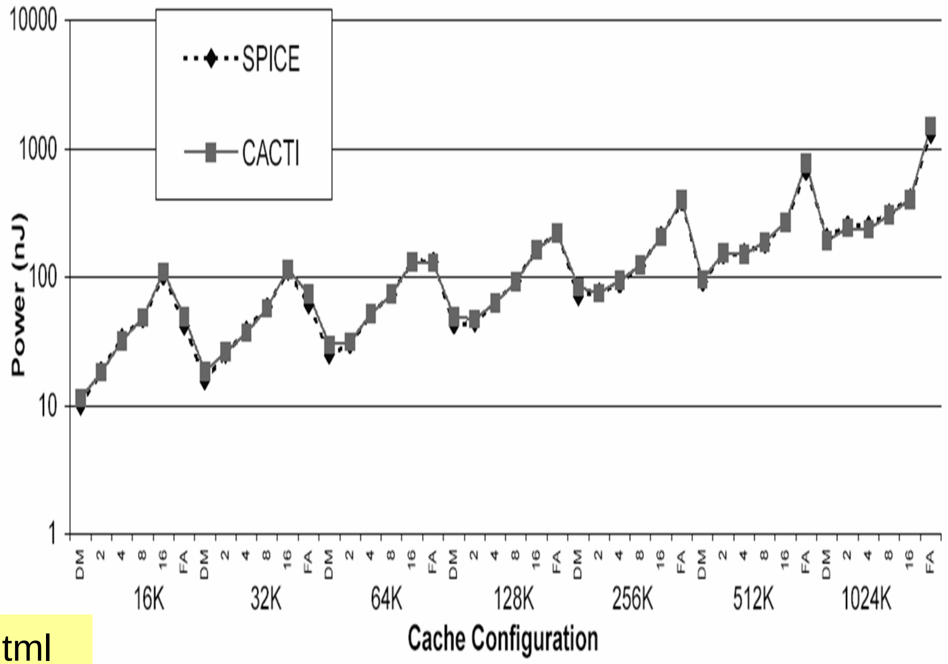
- It is not important, which address bit is set to ‘1‘
- The number of ‘1‘s in the address bus is irrelevant
- The cost of flipping a bit on the address bus is independent of the bit position.
- It is not important, which data bit is set to ‘1‘
- The number of ‘1‘s on the data bus has a minor effect (3%)
- The cost of flipping a bit on the data bus is independent of the bit position.

CACTI model



Cache model used

Comparison with SPICE



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

Evaluation of designs according to multiple objectives

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 becoming increasingly important

- since temperatures become more relevant due to increased performance, and
- since temperatures affect
  - usability and
  - dependability.

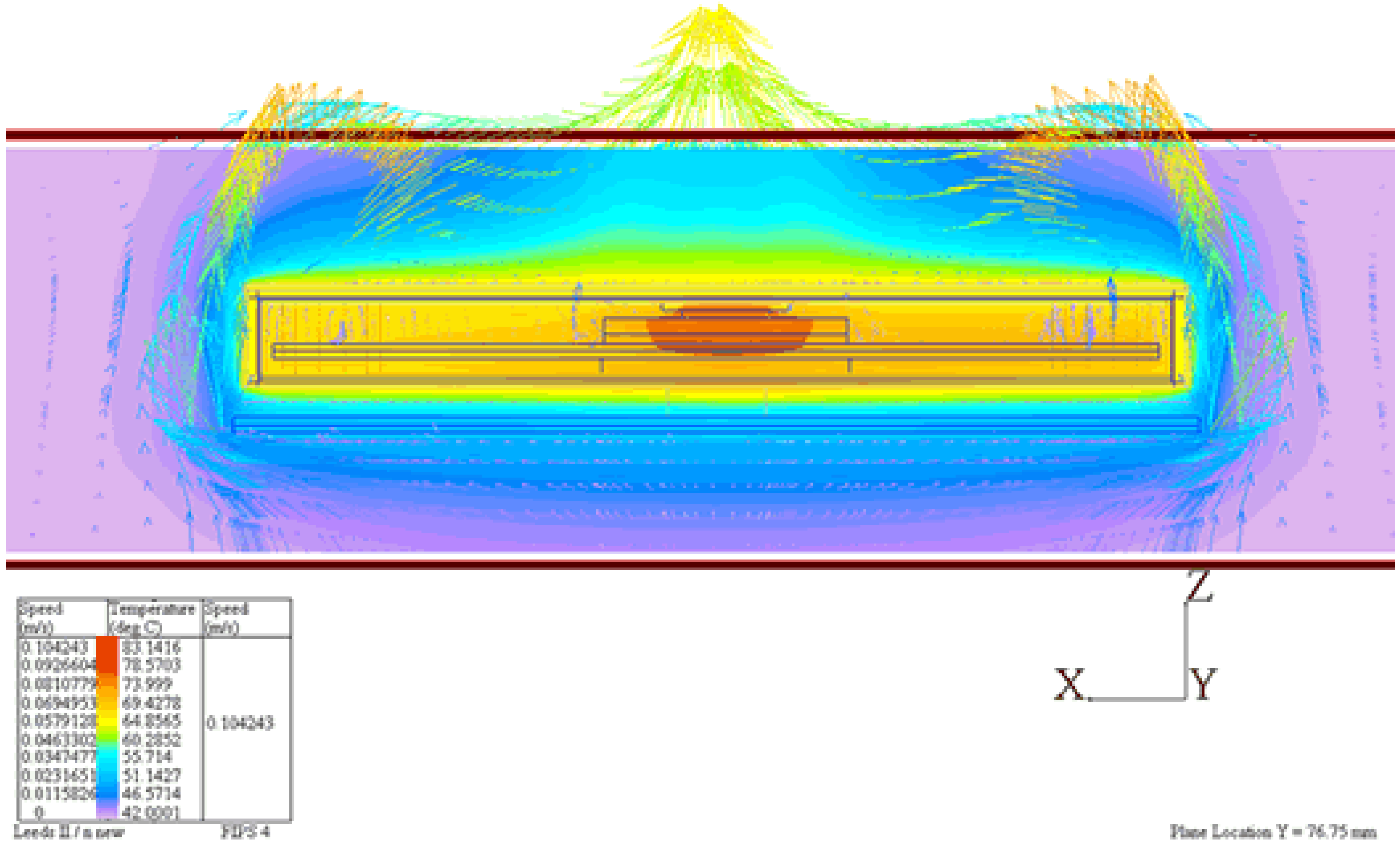
Model components

- **Thermal conductance** reflects the amount of heat transferred through a plate (made from that material) of area A and thickness L when the temperatures at the opposite sides differ by one Kelvin.
- The reciprocal of thermal conductance is called **thermal resistance**.
- **Thermal resistances add up** like electrical resistances
- Masses storing heat correspond to **capacitors**

☞ Thermal modeling typically uses **equivalent electrical models** and employs well-known techniques for solving electrical network equations

Results of simulations based on thermal models (1)

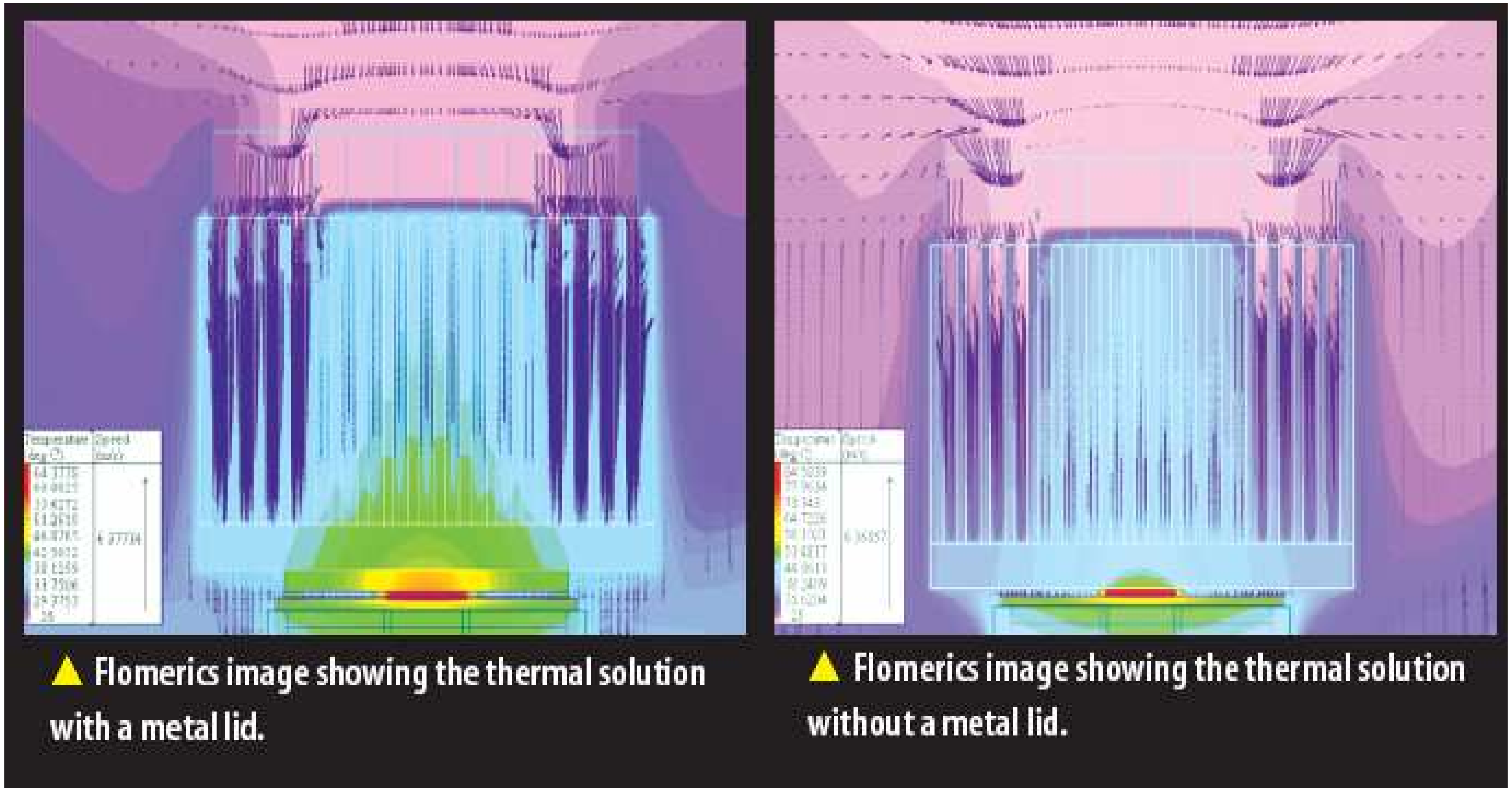
Encapsulated cryptographic coprocessor:



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

Results of simulations based on thermal models (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

Evaluation of designs according to multiple objectives

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

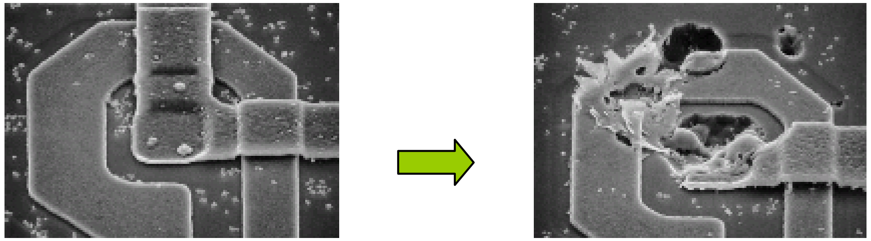
Impact of shrinking feature sizes

- Reduced reliability due to smaller patterns within semiconductor chips [ITRS, 2009]
- Transient & permanent faults
- Rate of faults expected to increase such that designs need to become fault-tolerant

Types of faults: Example: Electro-migration

Example : metal migration @ Pentium 4

www.jrwhipple.com/computer_hangs.html



FIT & “10^{-9}“

“10^{-9}“: For many systems, probability of a catastrophe has to be less than 10^{-9} per hour ≡ one case per 100,000 systems for 10,000 hours.

FIT: failure-in-time unit for failure rate (=1/MTTF≈1/MTBF);

1 FIT: 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 & **fault** models.

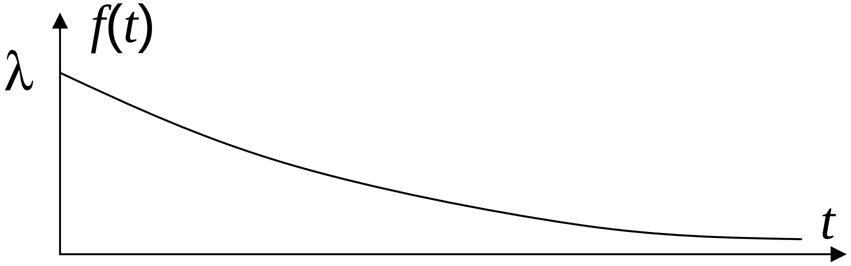
[Laprie et al., 1992, 2004]

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

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



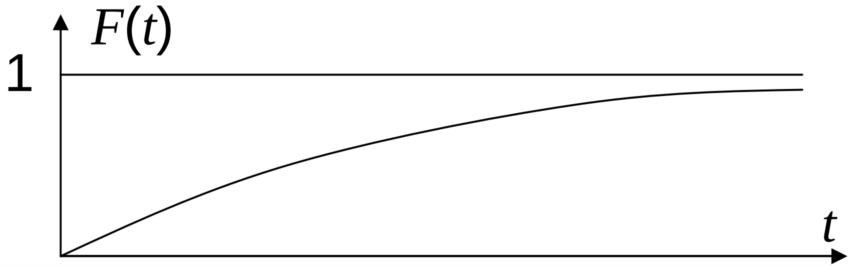
- $F(t)$ = probability of the system being faulty at time t:

$F(t)$ = $\Pr(T \le t)$

$$F(t) = \int_0^t f(x)\,dx$$

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:

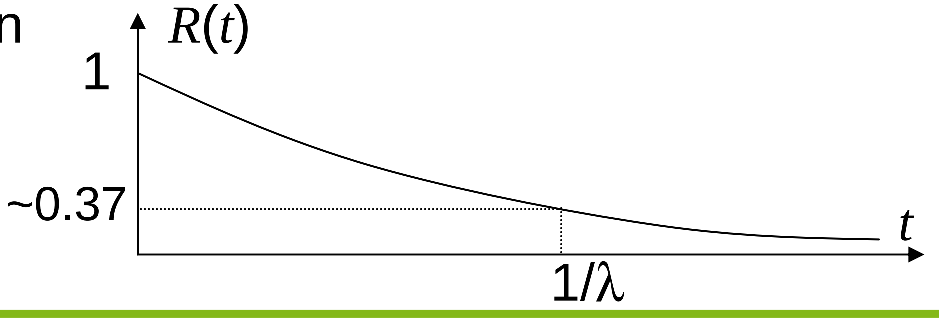
$$R(t)=\Pr(T>t),\ t\geq 0 \qquad R(t)=\int_t^{\infty} f(x)dx$$

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

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

<u>Example</u>: Exponential distribution

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



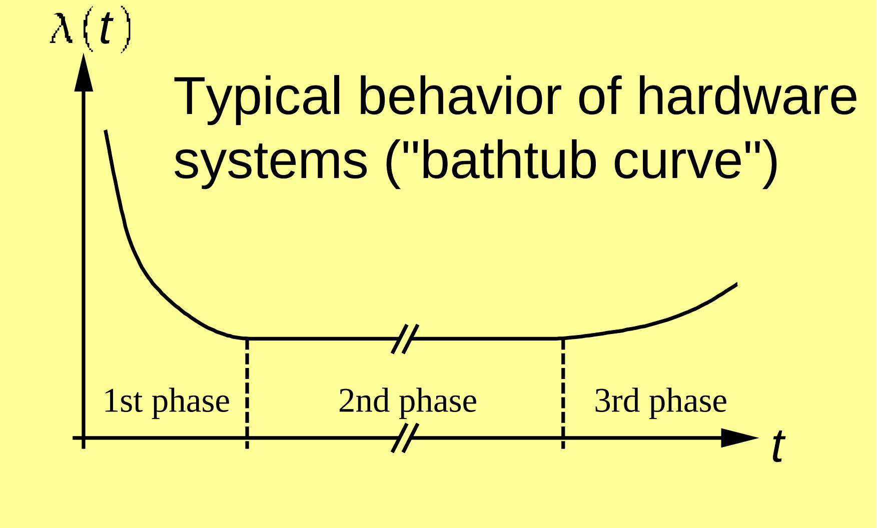
Failure rate

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

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

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* value 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

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

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

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

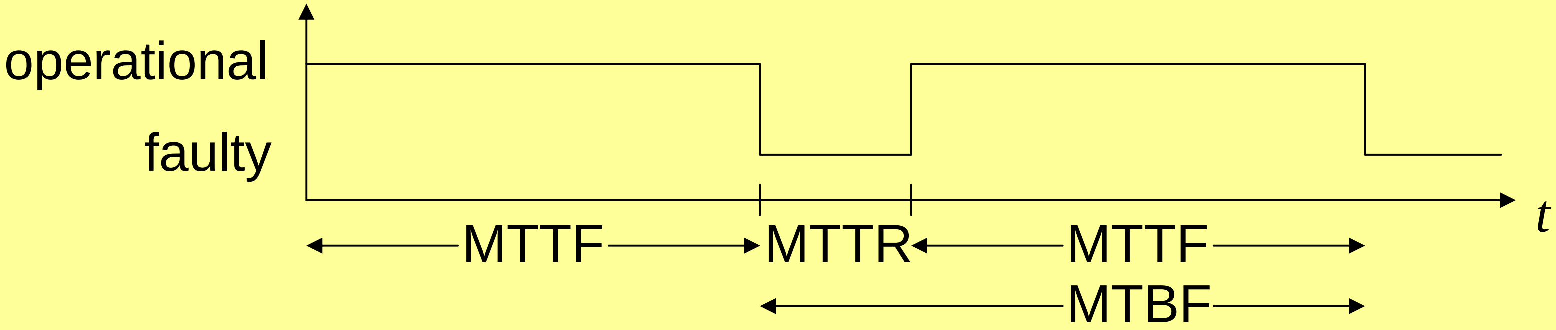
MTTF is the reciprocal value of failure rate.

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 …

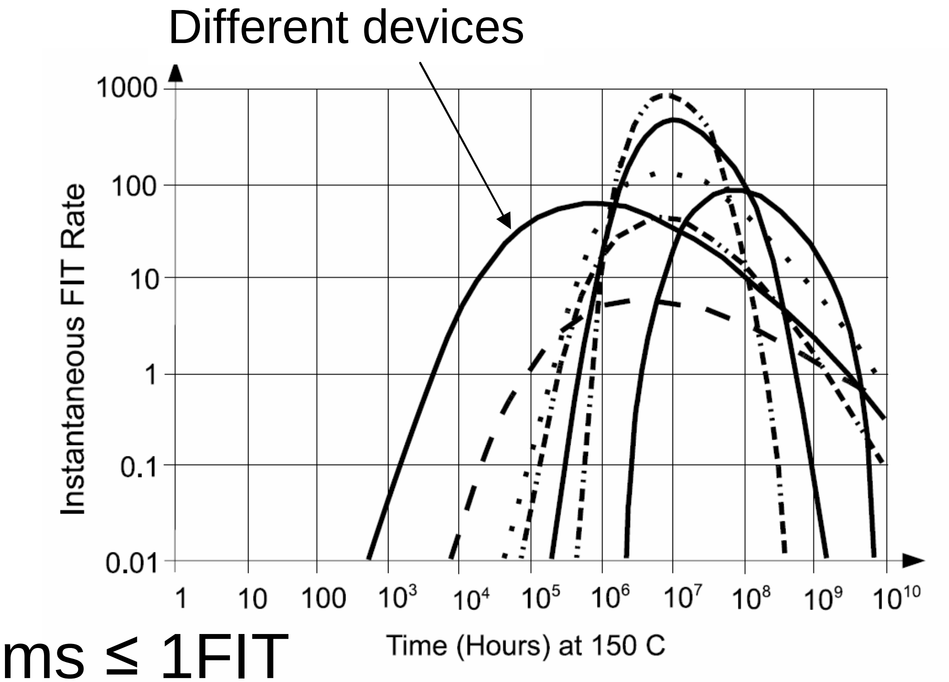


* Mixed up with MTTF, if starting in operational state is implicitly assumed

Actual failure rates

Example: failure rates less than 100 FIT for the first 20 years (175,300 hrs) of life at 150℃ @ TriQuint (GaAs)

[www.triquint.com/company/quality/faqs/faq_11.cfm]



Target: Failures rates of systems ≤ 1FIT

Reality: Failures rates of circuits ≤ 100 FIT

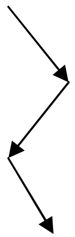
☞redundancy is required to make a system more reliable than its components

∃ non-constant failure rates!

Fault tree Analysis (FTA)

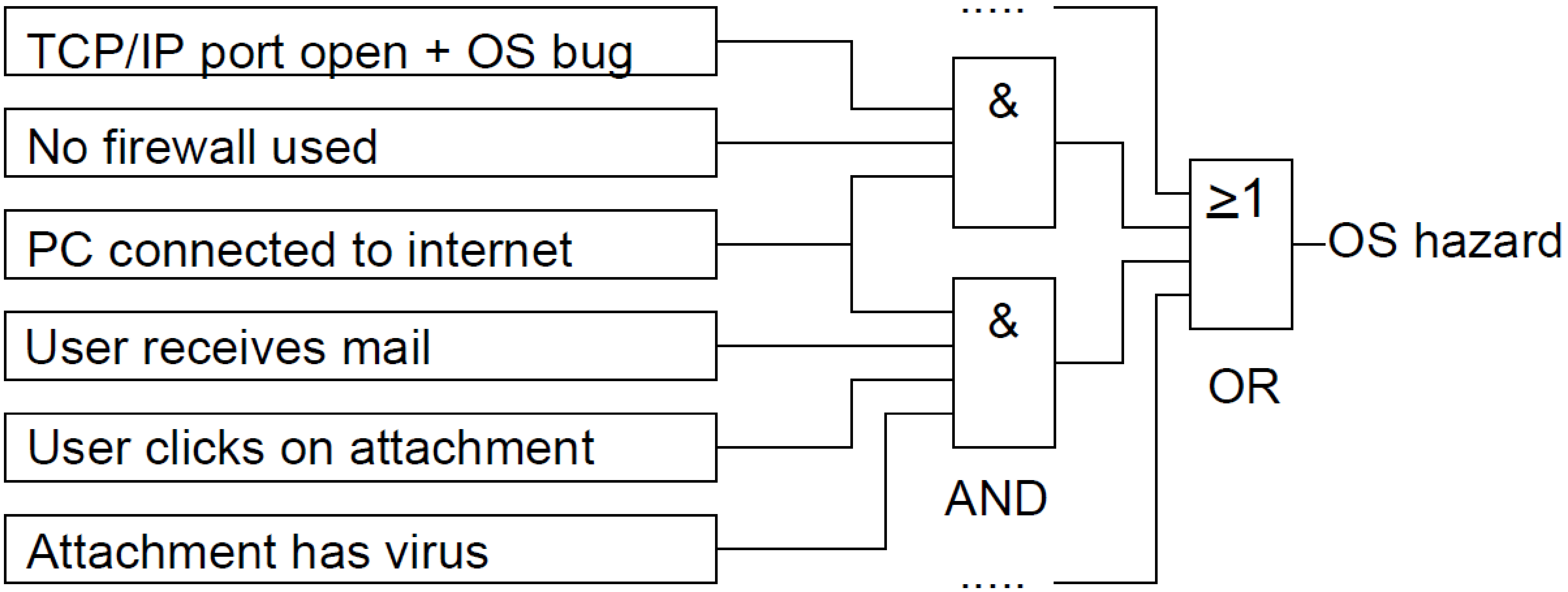
Damages are resulting from hazards/risks.
For every damage there is a severity and a probability.
Several techniques for analyzing risks.

- FTA is a top-down method of analyzing risks. Analysis starts with possible damage, tries to come up with possible scenarios that lead to that damage.



- FTA typically uses a graphical representation of possible damages, including symbols for AND- and OR-gates.
- OR-gates are used if a single event could result in a hazard.
- AND-gates are used when several events or conditions are required for that hazard to exist.

Example


TCP/IP port open + OS bug
No firewall used
PC connected to internet
User receives mail
User clicks on attachment
Attachment has virus
.....
&
&
AND
.....
≥1
OR
OS hazard

Limitations

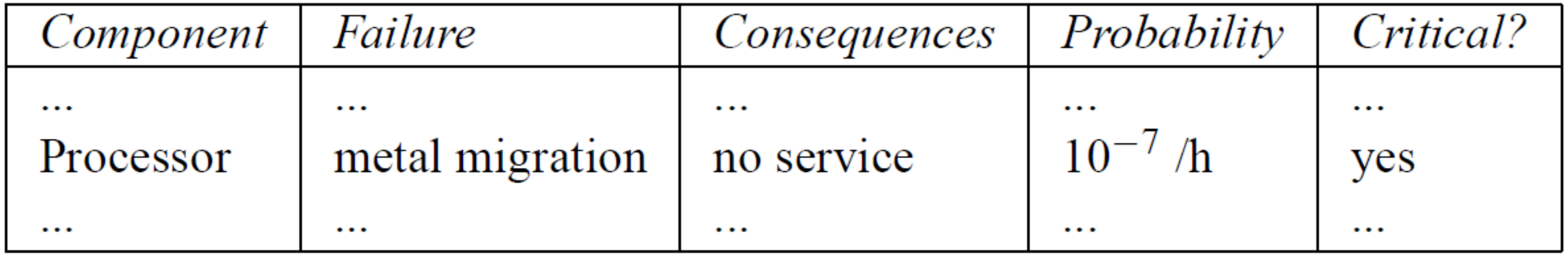
The simple AND- and OR-gates cannot model all situations.

For example, their modeling power is exceeded if shared resources of some limited amount (like energy or storage locations) exist.

Markov models may have to be used to cover such cases.

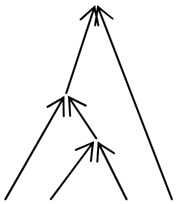
Failure mode and effect analysis (FMEA)

- FMEA starts at the components and tries to estimate their reliability. The first step is to create a table containing components, possible faults, probability of faults and consequences on the system behavior.



Component	*Failure*	*Consequences*	*Probability*	*Critical?*
...	...	...	...	...
Processor	metal migration	no service	10^{-7} /h	yes
...	...	...	...	...

- Using this information, the reliability of the system is computed from the reliability of its parts (corresponding to a bottom-up analysis).



Safety cases

Both approaches may be used in “safety cases”.

In such cases, an independent authority has to be convinced that certain technical equipment is indeed safe.

One of the commonly requested properties of technical systems is that no single failing component should potentially cause a catastrophe.

Fault injection

Fault simulation may be too time-consuming

☞ If real systems are available, faults can be injected.

Two types of fault injection:

1. local faults within the system, and
2. faults in the environment (behaviors which do not correspond to the specification). For example, we can check how the system behaves if it is operated outside the specified temperature or radiation ranges.

Physical fault injection

Hardware fault injection requires major effort, but generates precise information about the behavior of the real system.
3 techniques compared in the PDCS project on the MARS hardware [Kopetz]:

Injection Technique	**Heavy-ion**	**Pin-level**	**EMI**
Controllability, space	Low	High	Low
Controllability, time	None	High/medium	Low
Flexibility	Low	Medium	High
Reproducibility	Medium	High	Low
Physical reachability	High	Medium	Medium
Timing measurement	Medium	high	Low

Software fault injection

Errors are injected into the memories.

Advantages:

- **Predictability:** it is possible to reproduce every injected fault in time and space.
- **Reachability:** possible to reach storage locations within chips instead of just pins.
- **Less effort** than physical fault injection: no modified hardware.

Same quality of results?

Dependability requirements

Allowed failures may be in the order of 1 failure per 10 ^{9} h.

~ 1000 times less than typical failure rates of chips.

☞ For safety-critical systems, the system as a whole must be more dependable than any of its parts.

☞ fault-tolerance mechanisms must be used.

Low acceptable failure rate → systems 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.

Summary

Evaluation and Validation: Objectives

- Energy and power consumption
- Thermal behavior
- Reliability
  - Definitions
  - Failure rates
  - MTBF, MTTF, MTTR
  - Fault tree analysis, FMEA
  - Fault injection
    - Software and
    - hardware-based techniques