Zuverlässigkeit und Fehlertoleranz / Dependability and Fault Tolerance

Chapter 2

Faults and Fault Models

H. T. Vierhaus

Winter Semester 2018 / 2019

Faults- Errors-Defects

Specifically for hardware in electronic systems

Defekt – Defect: Something wrong with the physical structure

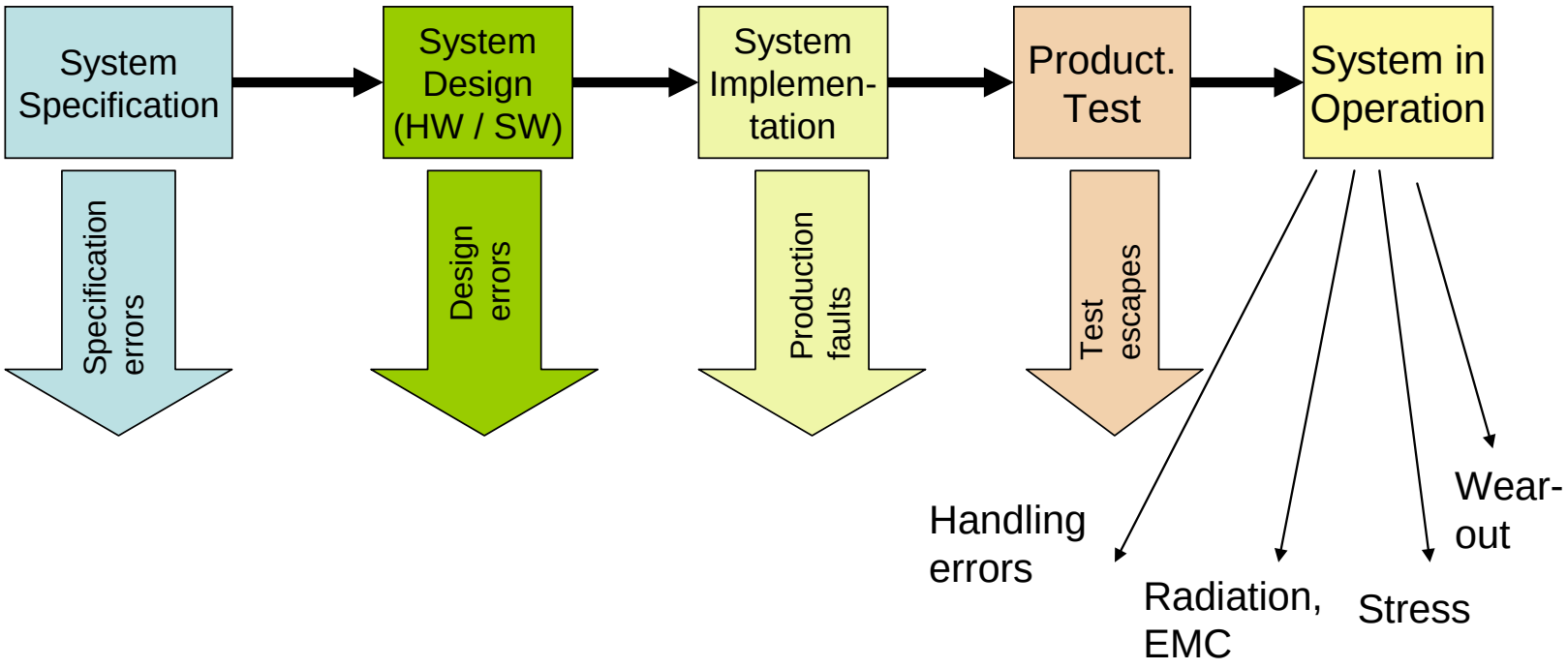
Fehler – Fault: a faulty signal, for example a fault bit, possibly never to appear

Fehler – Error: a faulty signal which becomes visible and may have an effect

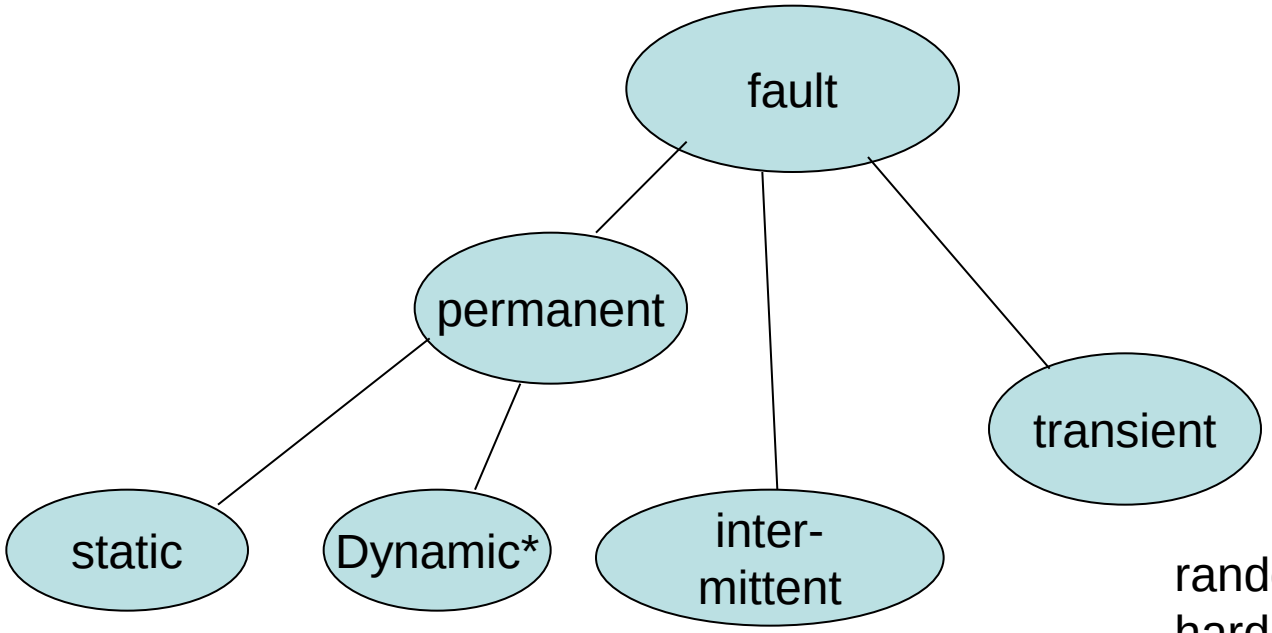
Failure – Versagen: the systems produces an erroneous output with effects

Breakdown - Ausfall: the total function of a system collapses

Hierarchy of Faults / Errors

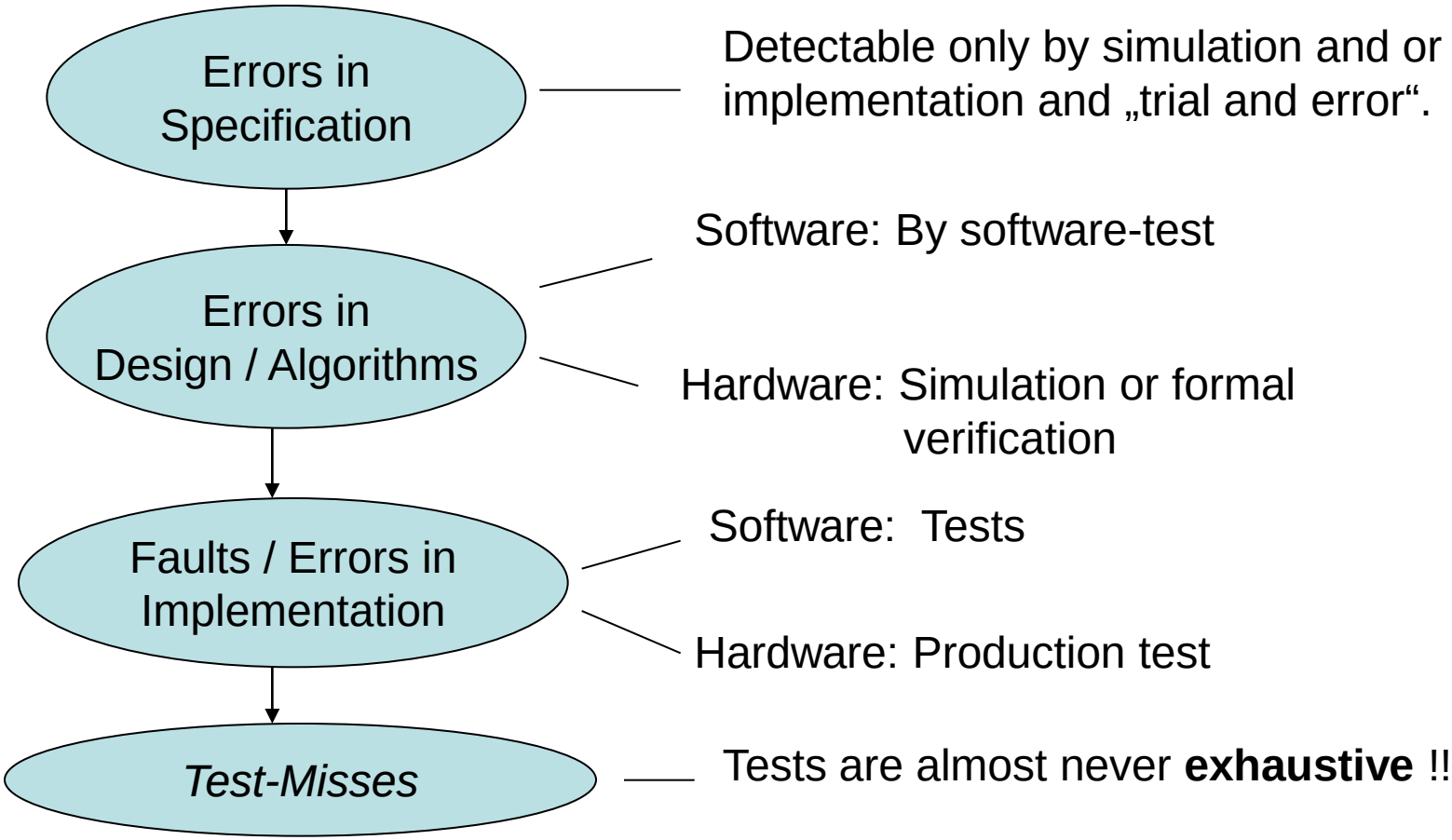


Types of Faults in Hardware



*** most important: delay faults**

Error Mechanisms and Validation



Fault Behaviour

Errors in specificaton, in hard- and software design / implementation are **permanent,** unless there is a repair event**.**

They will not alter themselves during the life time of a system.

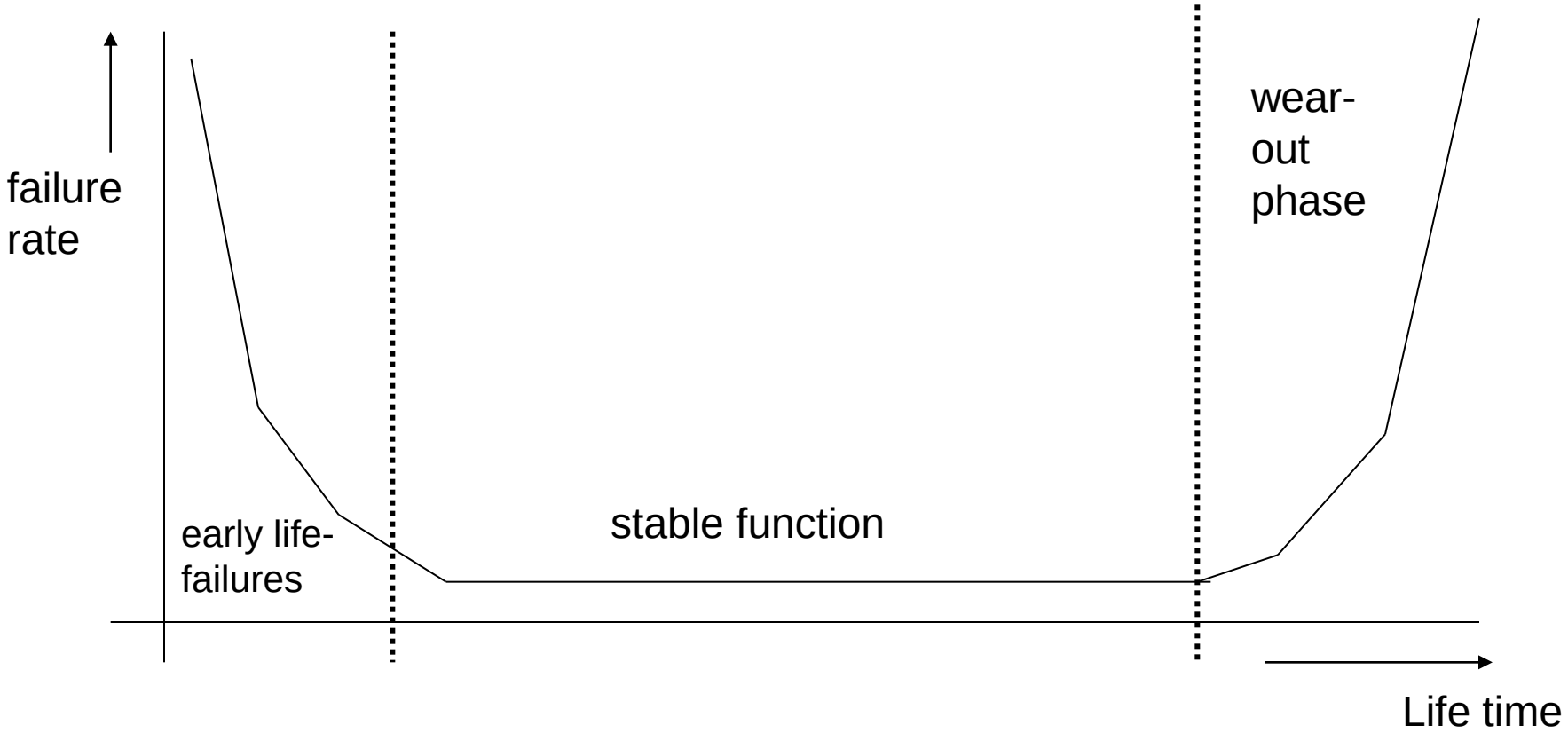
Faults in hardware can be **permanent, transient** or **intermittent**.

Throughout system life time, hardware faults can:

- newly appear (due to stress, aging, radiation, EM coupling)
- disapper (self healing)
- get worse (change from intermittent to permanent).

Errors / Failures over Time

„bathtub-curve“



Lifetime / Failure Curve

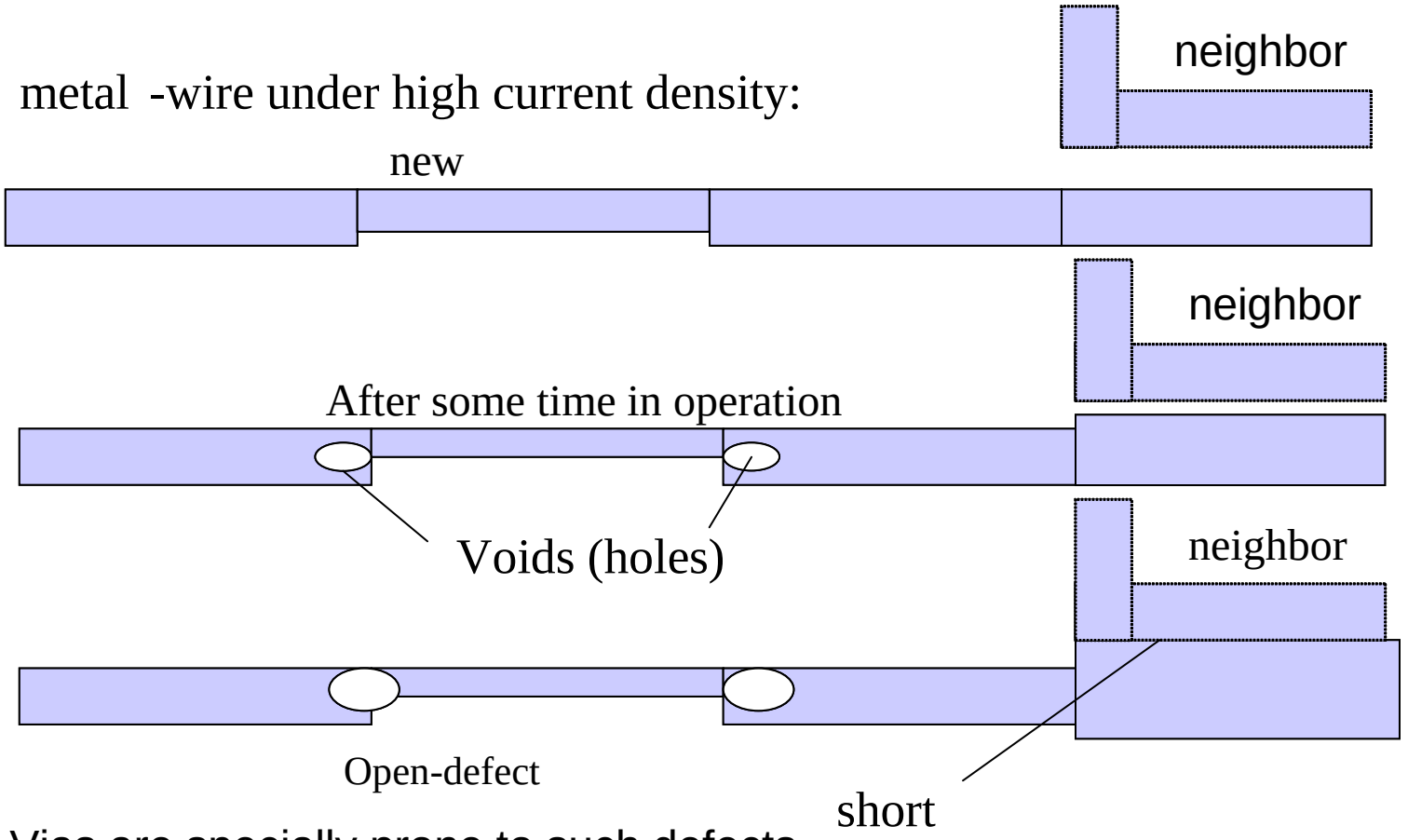
Early life failures: Hidden defects after production, will produce faults and errors only after receiving some „stress“. Also due to test misses !

Wear-out failures: All mechanical, electrical, electronic devices undergo mechanisms of aging and wear-out.

Wear-out : Only in hardware. But software needs processors to run. Strongly depending on temperature !!! Typically, stress grows about exponentially with temperature.

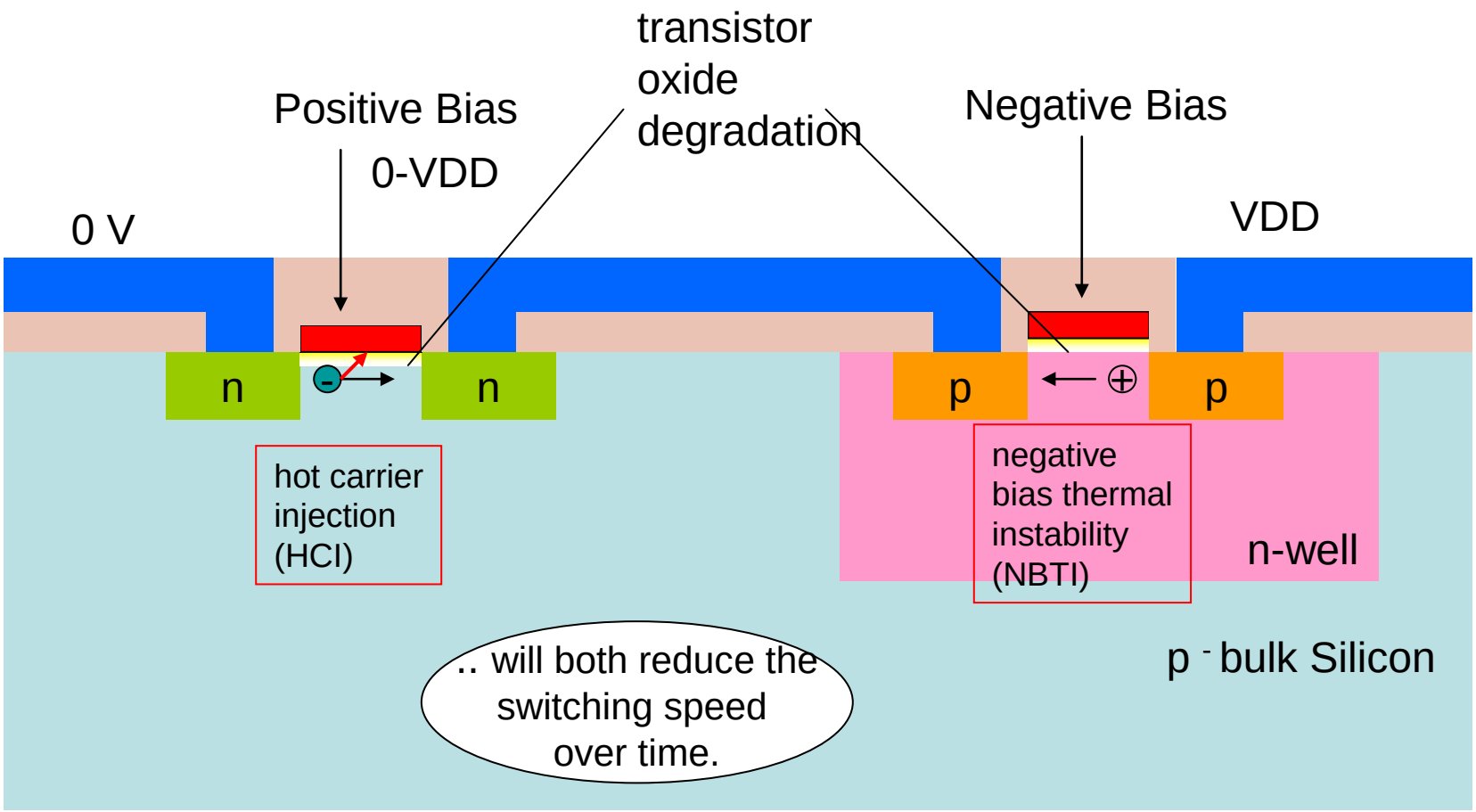
Transient faults: Error density after early life and before wear-out is mainly caused by radiation effects and EM-coupling..

Metal Migration



- Vias are specially prone to such defects
- The effect is reversible by reversing the direction of current flow !

IC-Fault Mechanisms



Fault Mechanisms in CMOS ICs

- Hot Carrier Injection (HCI), mainly in n-channel transistors:

  High-energy electrons „tunnel“ trough the gate-oxide, assemble on the gate and cause modifications of the transistor parameters.
  A well-know effects technically used for EEPROMs and for FLASH memory !

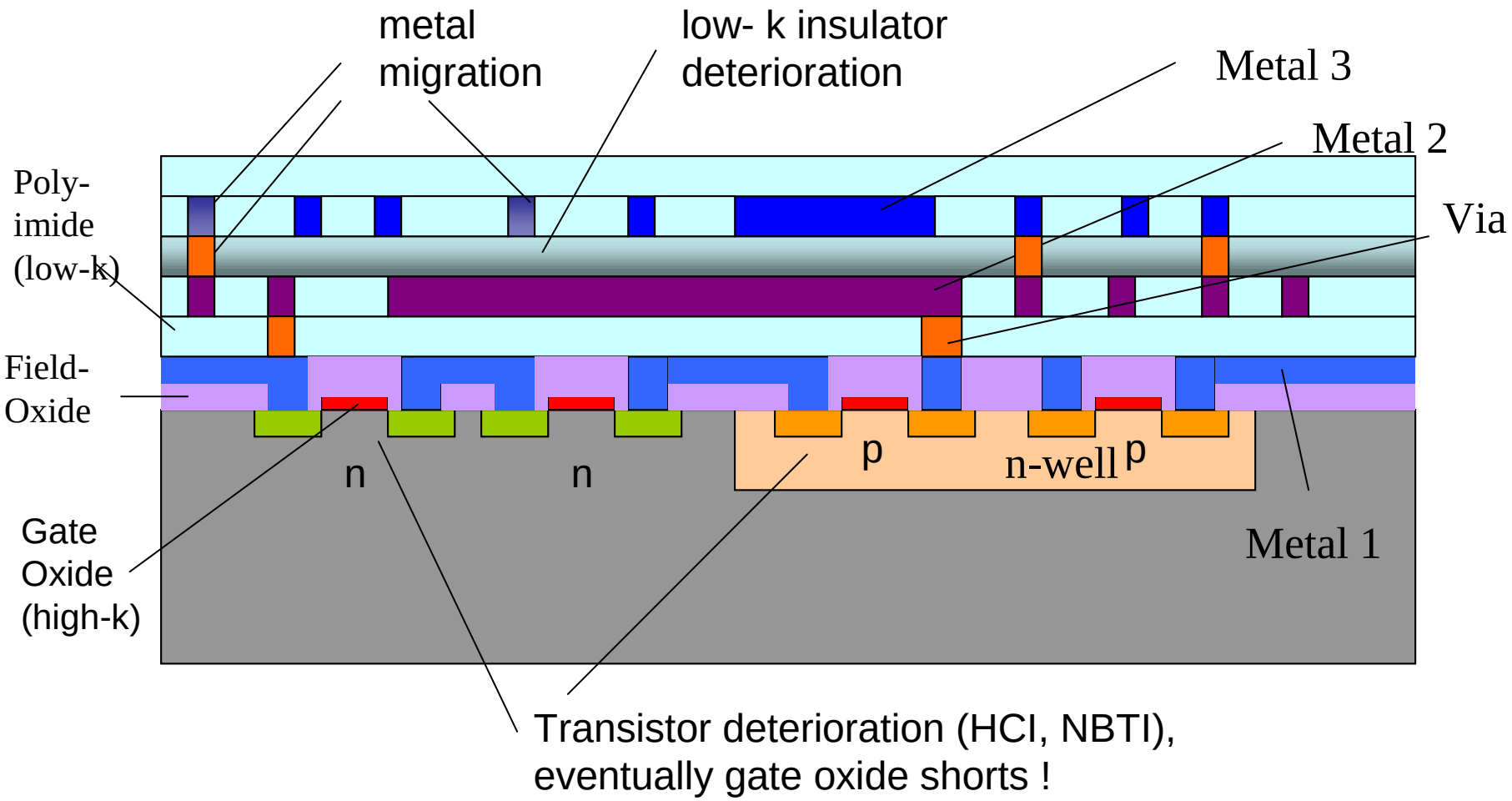
- Negative (positive) bias thermal instability NBTI (PBTI):

  With a constant positive / negative bias applied over a longer time, hydrogen ions at the semiconductor / oxide interface migrate in the gate-oxide. This causes variations in the transistor threshold voltage (the voltage between gate-bulk from which it starts conducting). The transistor becomes slower, gate delays increase. The effect is partly reversible. It is much stronger on p-channel (NBTI) than on n-channel transistors (PBTI).

- Dielectric gate rupture:

  Thin oxide layers (typically only 3 atoms thick on CMOS gates !) may actually break and cause a non-reversible short between gate and channel of an MOS transistor !

Wear-Out and Aging

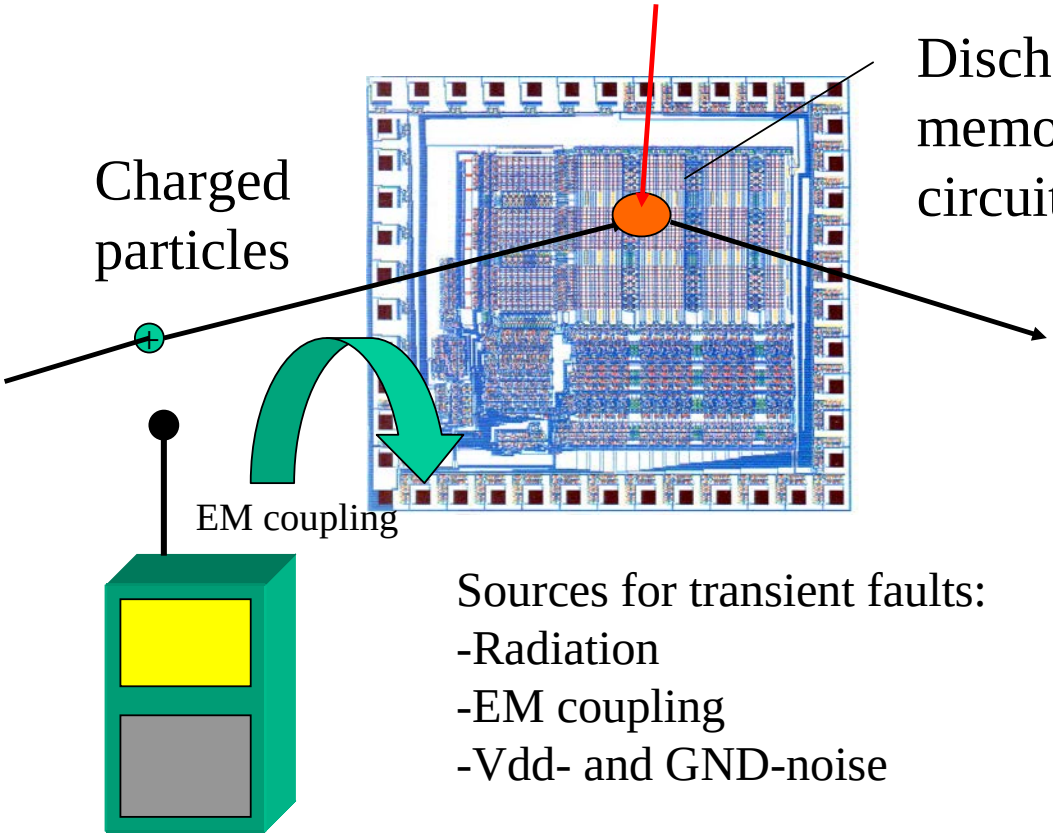


Important Aging Effects on ICs

- Parameter shifting, mainly increase of V_{th} (threshold voltage of MOS-transistors) by negative / positive bias thermal instability (NBTI / PBTI) and by hot carrier injection (HIC), tunnelling of electrons through gate oxide (mainly in n-MOS).
- Thinning of lines, creation of voids through metal migration, may also create bridges.
- Break of oxide layers between metal wires (insulators)
- Break of lines due to mechanial stress, also caused by heating
- Break of gate oxide in MOS transistors

.. with higher specific stress for nano-circuits !! Due to thinner lines and thinner oxide layers !

Transient Faults



Sources for transient faults:
-Radiation
-EM coupling
-Vdd- and GND-noise

Sources of Transient Faults

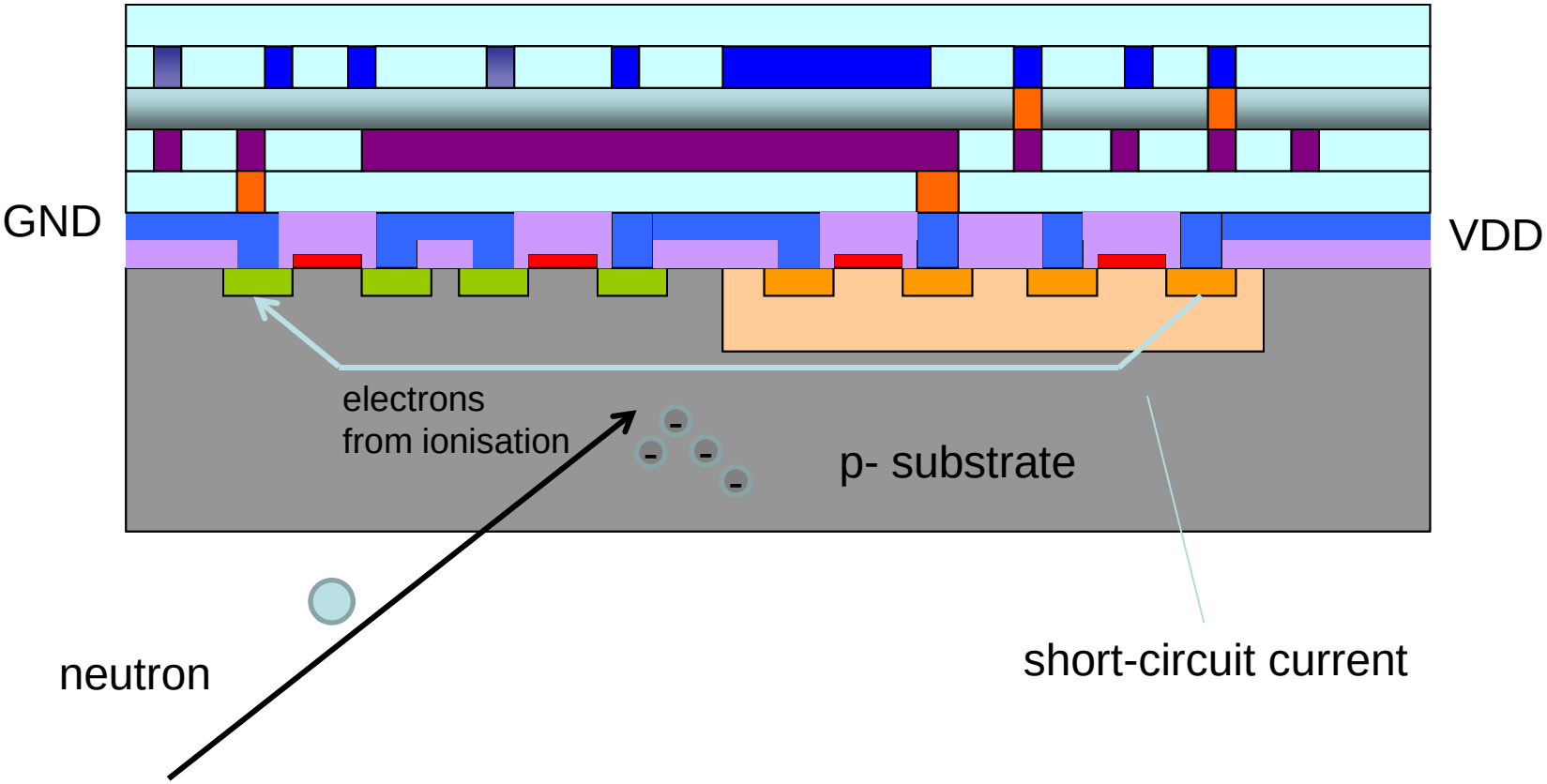
1. **Particle Radiation:** Neutrons, alpha-particles from radioactive decay cause „faulty“ bits als „single even upsets(SEUs) or single event transients (SETs) also as „multi event upsets“ (MEUs) / multi-event transients (METs).

   Particle radiation can trigger a break-through of reverse biases p-n-junctions in MOS transistors. „Latch-Up“ is a well known failure process in CMOS-logic which creates a direct VDD- GND-connection, high currents, thermal damage.

2. **Electromagnetic Coupling**: Lines on ICs may act as antennas and „catch“ wireless EM-radiation. Lines on ICs suffer from capacitive coupling of lines, the more the smaller.

3. **VDD / GND Noise:** VDD / GND voltage levels are not stable, but will change with circuit activity. More switching events at the same time cause stress on the VDD-grid and lower voltage locally. Currents on ground nets can also „raise“ GND-voltage !

Latch-Up


GND
VDD
electrons
from ionisation
p- substrate
neutron
short-circuit current

Error Density in Real Life

Cray XT5, Oak Ridge, Tennessee, „Jaguar“

Memory size: 360 terabytes, all ECC*-protected

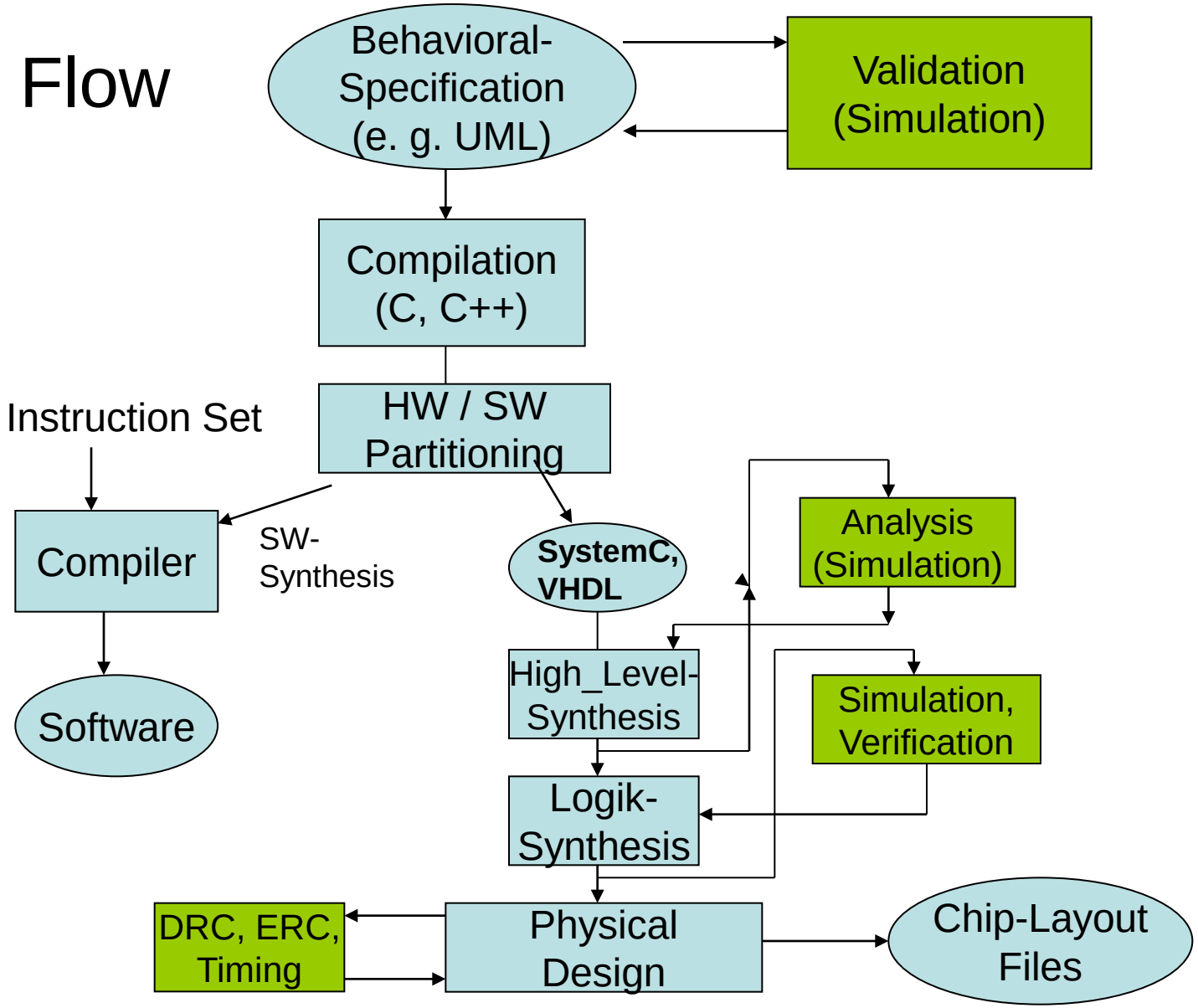
ECC error rate: 350 per minute (single bit errors)
1 per 24 hours (double bit errors)

This corresponds to one single bit error on a 1 GByte-memory about every 16 hours !

Radiation- induced „hits“ causing permanent damages were also measured, but much less frequently !

* error correction code

Design Flow


Behavioral-Specification (e. g. UML)
Validation (Simulation)
Compilation (C, C++)
HW / SW Partitioning
Instruction Set
Compiler
SW-Synthesis
Software
SystemC, VHDL
Analysis (Simulation)
High_Level-Synthesis
Simulation, Verification
Logik-Synthesis
DRC, ERC, Timing
Physical Design
Chip-Layout Files

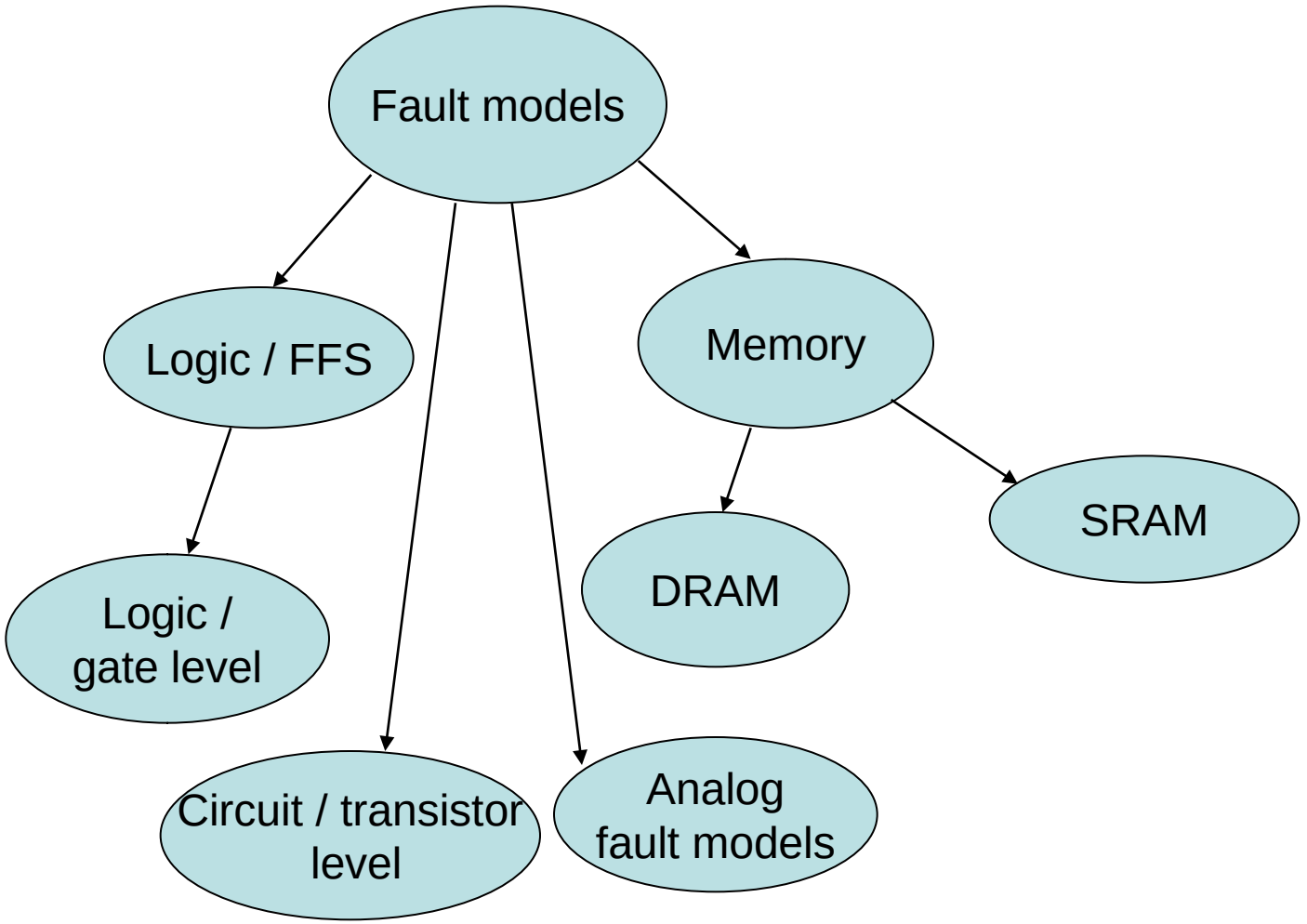
Simulation auf Fault Behavior

Fault Simulation: The fault is „modeled“, mainly at the logic gate level. Fault simulations can then effectively simulate the faulty (and the good) circuits, often in parallel.

Fault Injection: We take a normal simulator at the RT- or gate-level or even an FPGA. Then we introduce a fault, e. g. by setting a node to „always 0“ and observe the outcome.

Fault injection is the more universal approach, since it does not require special models or special tools, but it is much slower.

Fault Models for Digital Systems

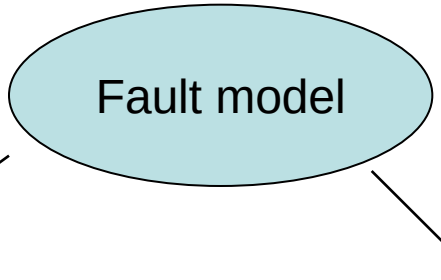


Hardware Fault Models

The variety of physical defects is large and almost inpossible to cover by models and simulation tools. Complexity problem !!!

One common solution is to „simplify“ real faults towards specific types of „abstracted“ faults at the circuit- or logic level.

This simplification works only for digital circuits / systems.



Simple an inaccurate

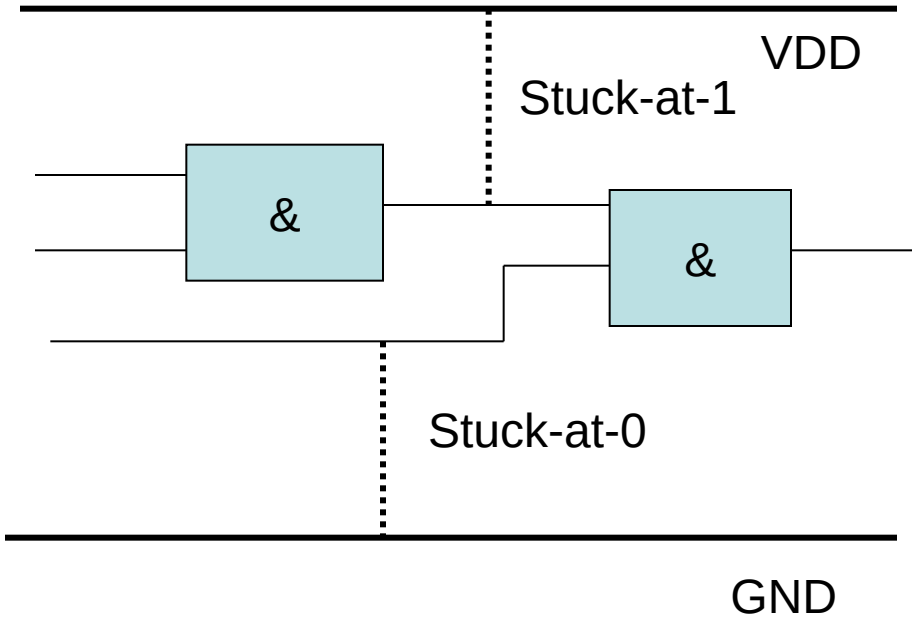
Pro: Can manage large Complexity in simulation
Con: May miss real faults

Complex and accurate

Pro: May catch complex fault behavior

Con: Complexity of simumation for larger circuits explodes.

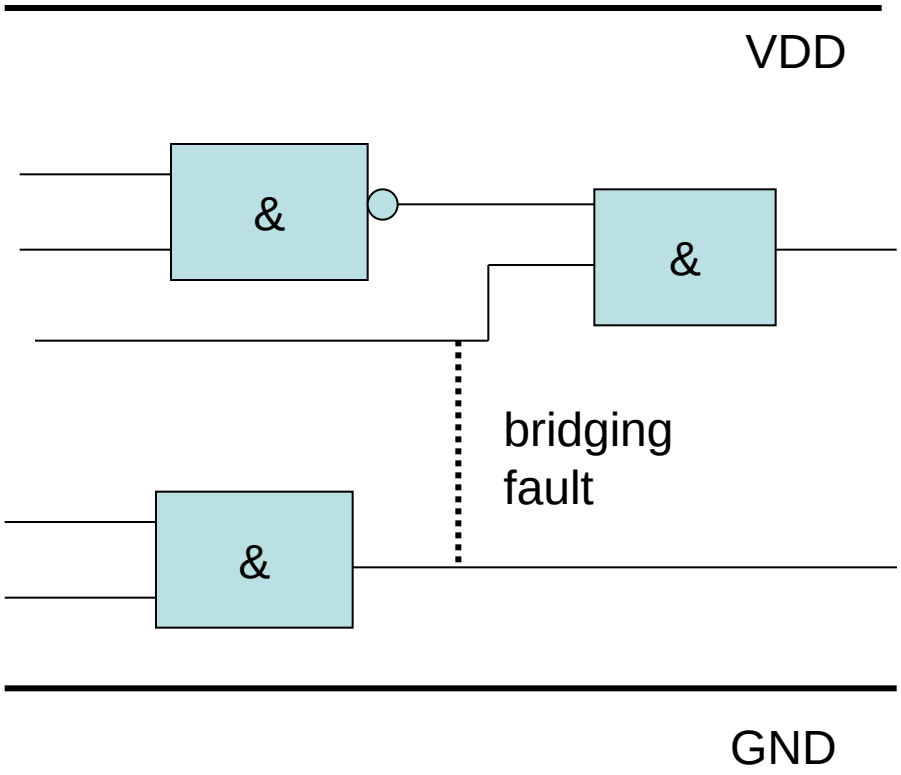
Fault Models: Stuck-at-Fault



A node in a logic circuit is erroneously fixed at the „VDD“ (1) or the ground (0) potential.

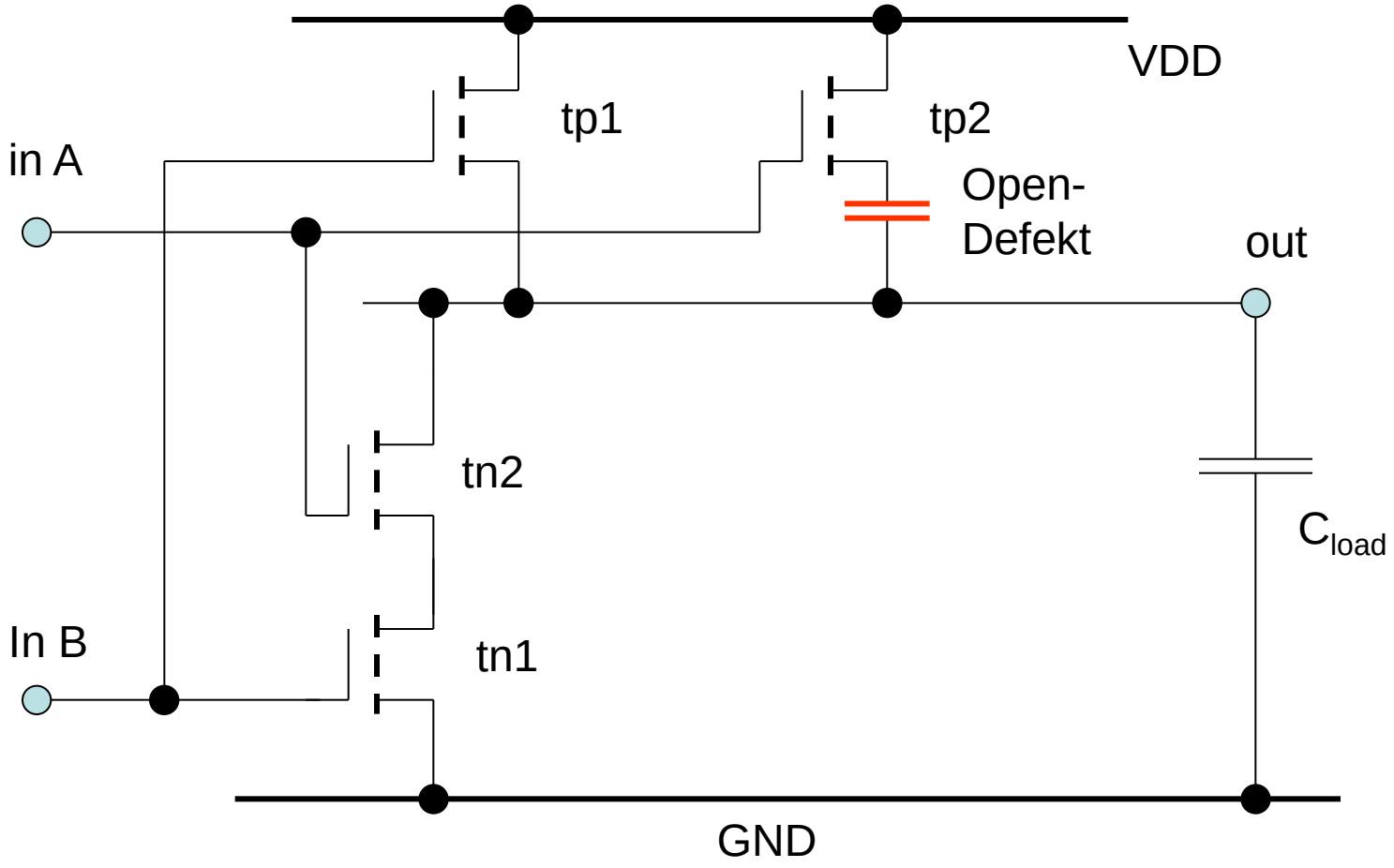
Eldred, 1959

Bridge Fault



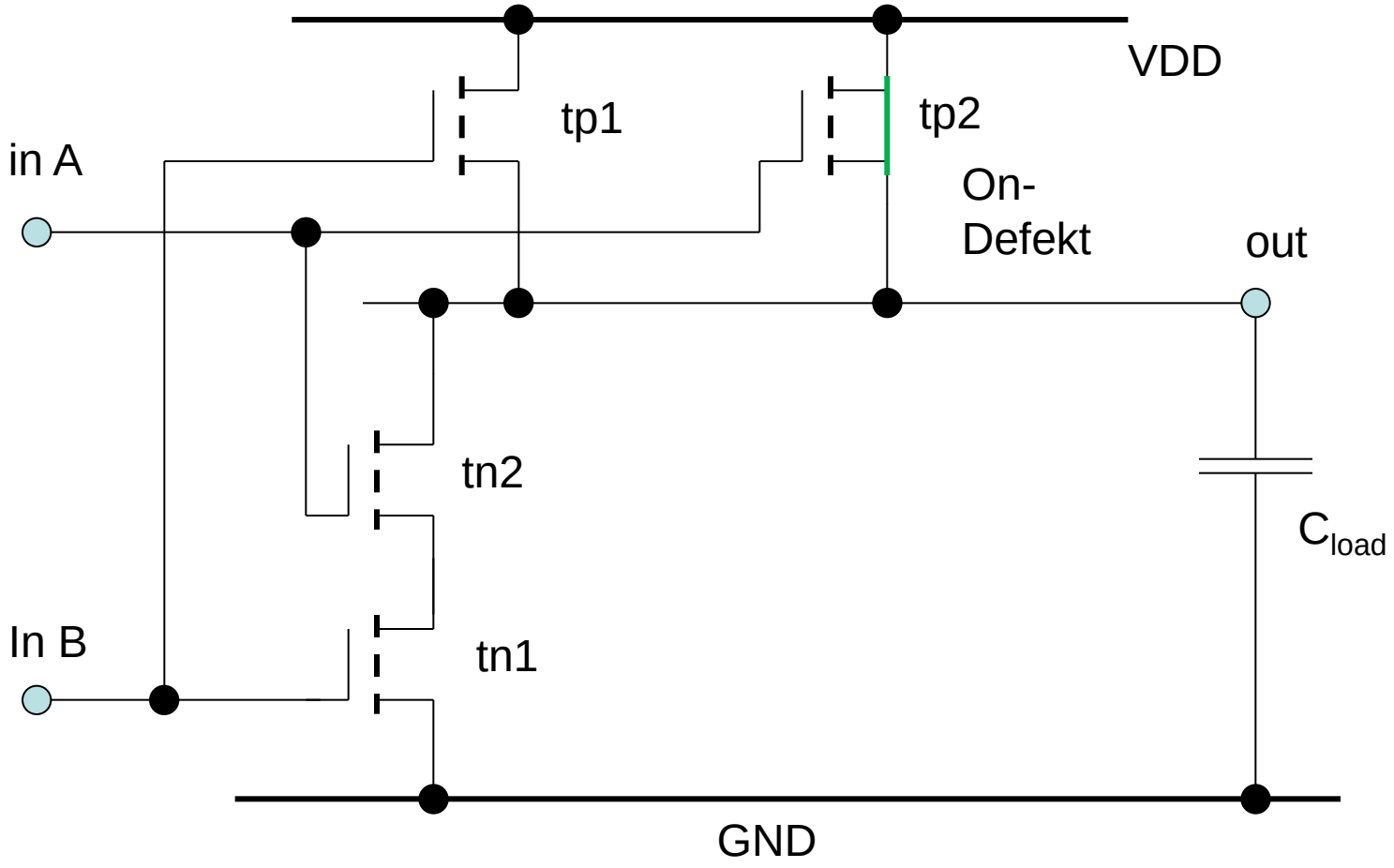
Bridging a signal line to VDD / GND results in a stuck-at 1 / 0 fault. Bridges between signal lines are much more difficult to model at the logic gate level. They may result in voltage levels between VDD / GND or even in feedback in combinational logic. An intermediate voltage level at around VDD/2 may even be taken as „1“ by one fed gate and as „0“ by another one. Byzantine General`s Problem !

Stuck-Open-Fault



One transistor is always „non conducting“. Leads to parasitic storage effects !

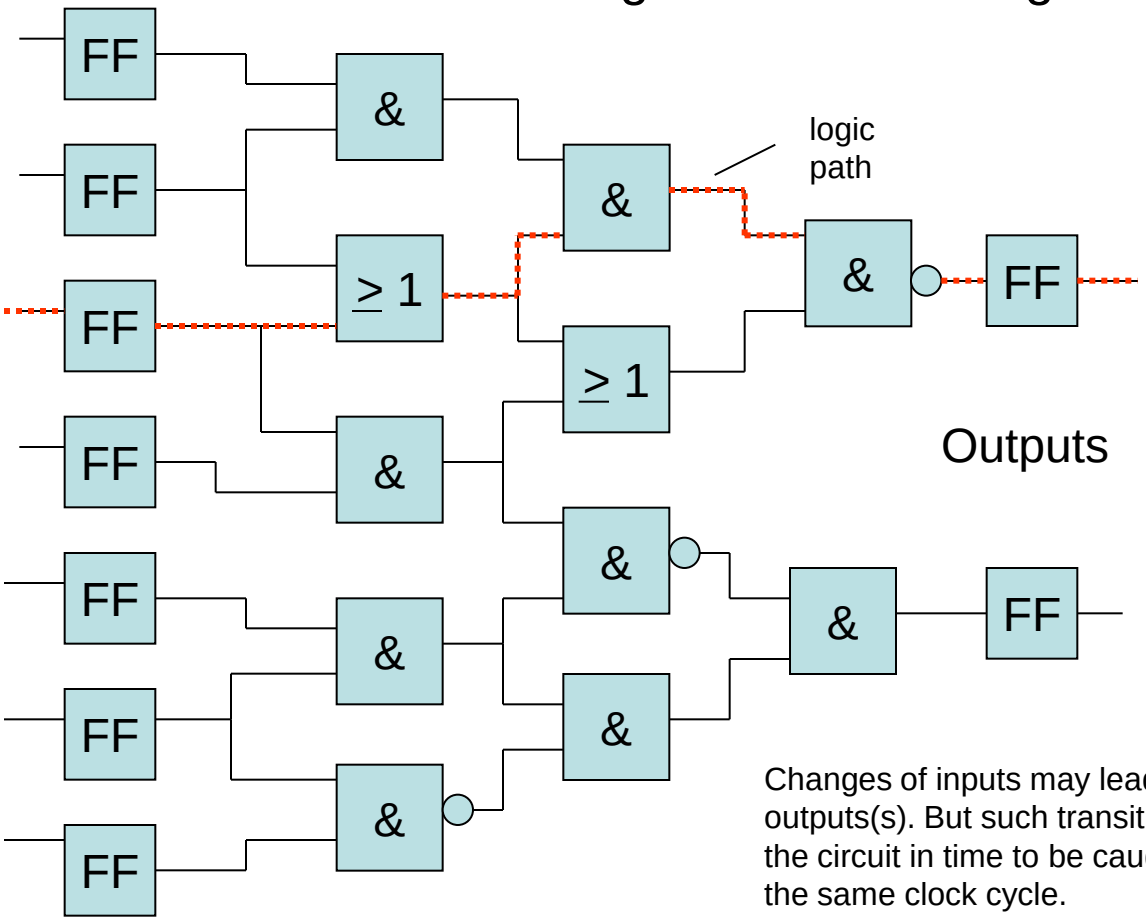
Stuck-On- / Short -Fault



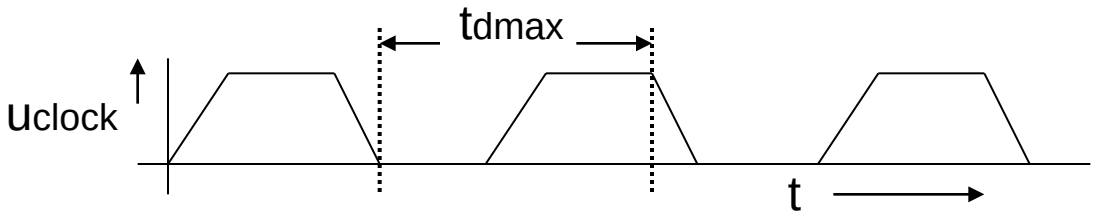
One transistor is always conducting. Leads to intermediate voltage levels.

Logic-Structure / Logic Paths

Inputs



Changes of inputs may lead to a change at outputs(s). But such transitions have to migrate through the circuit in time to be caught by output FFS within the same clock cycle.



Delay Faults

- Gate Delay Faults:

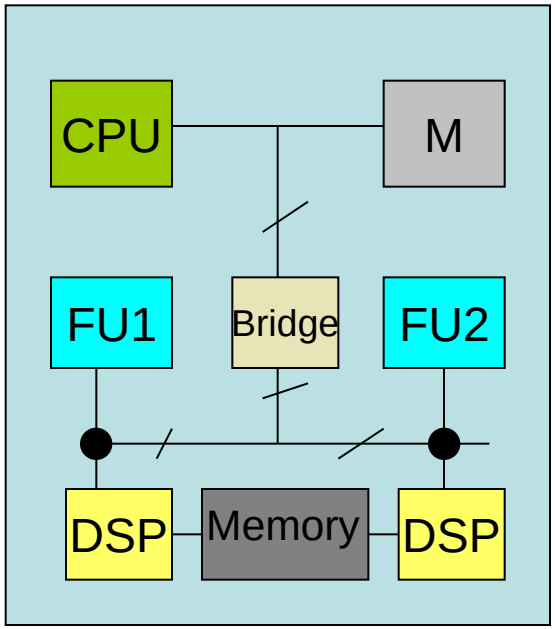
  Gates switch outputs too slowly. Reasons can be:
  Defects from manufacturing, aging, lower VDD because of „noise“.
  Almost untestable, since single gates can hardly be contacted for testing.

- Path Delay Faults:
  Switching of transitions along logic paths is too slow. Reasons can be gate-delay faults, but also EM-coupling and VDD-noise.

- Transition Faults: Logic gate outputs cannot be switched to opposite values via specific inputs. For example due to stuck-open or stuck-on transistors.

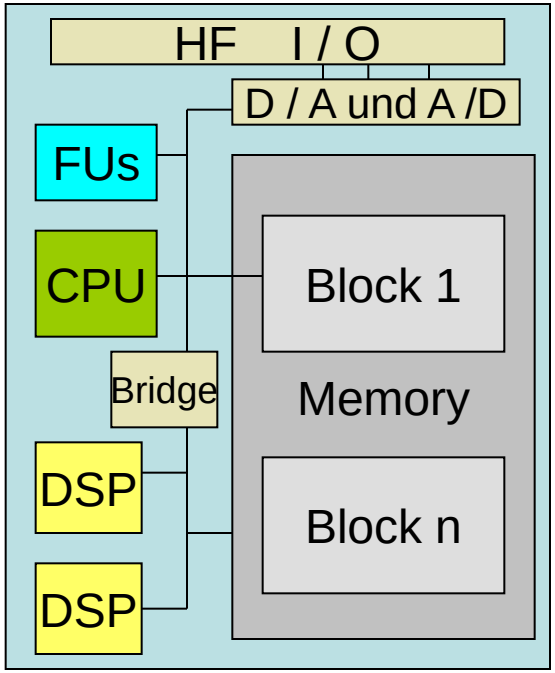
Logic and Memory

SoC 2001



Memory-Share 10 – 30%

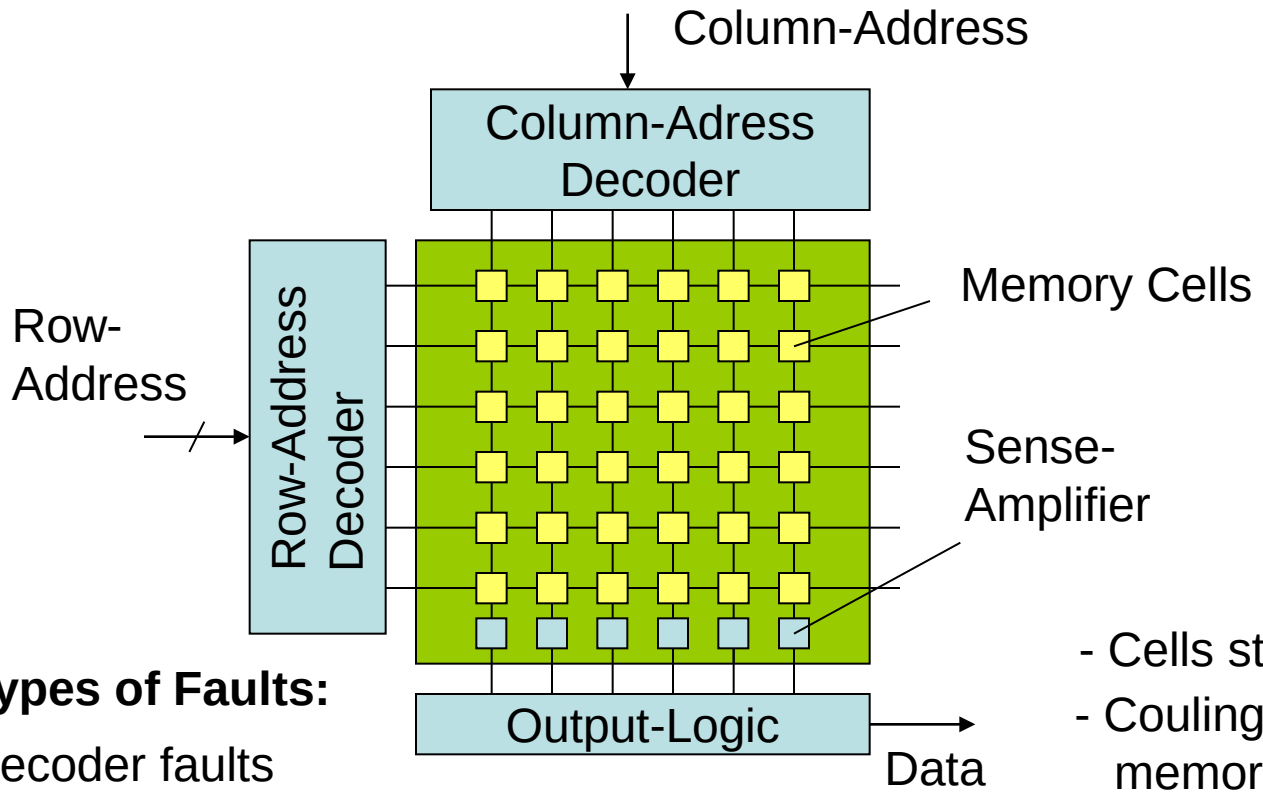
SoC 2015



Memory-Share 50 bis 90 %

Memory faults get a dominating importance !

Memory Faults



Types of Faults:

- Decoder faults
- Faults of the sense-amplifiers
- Faults on multiplexors and registers

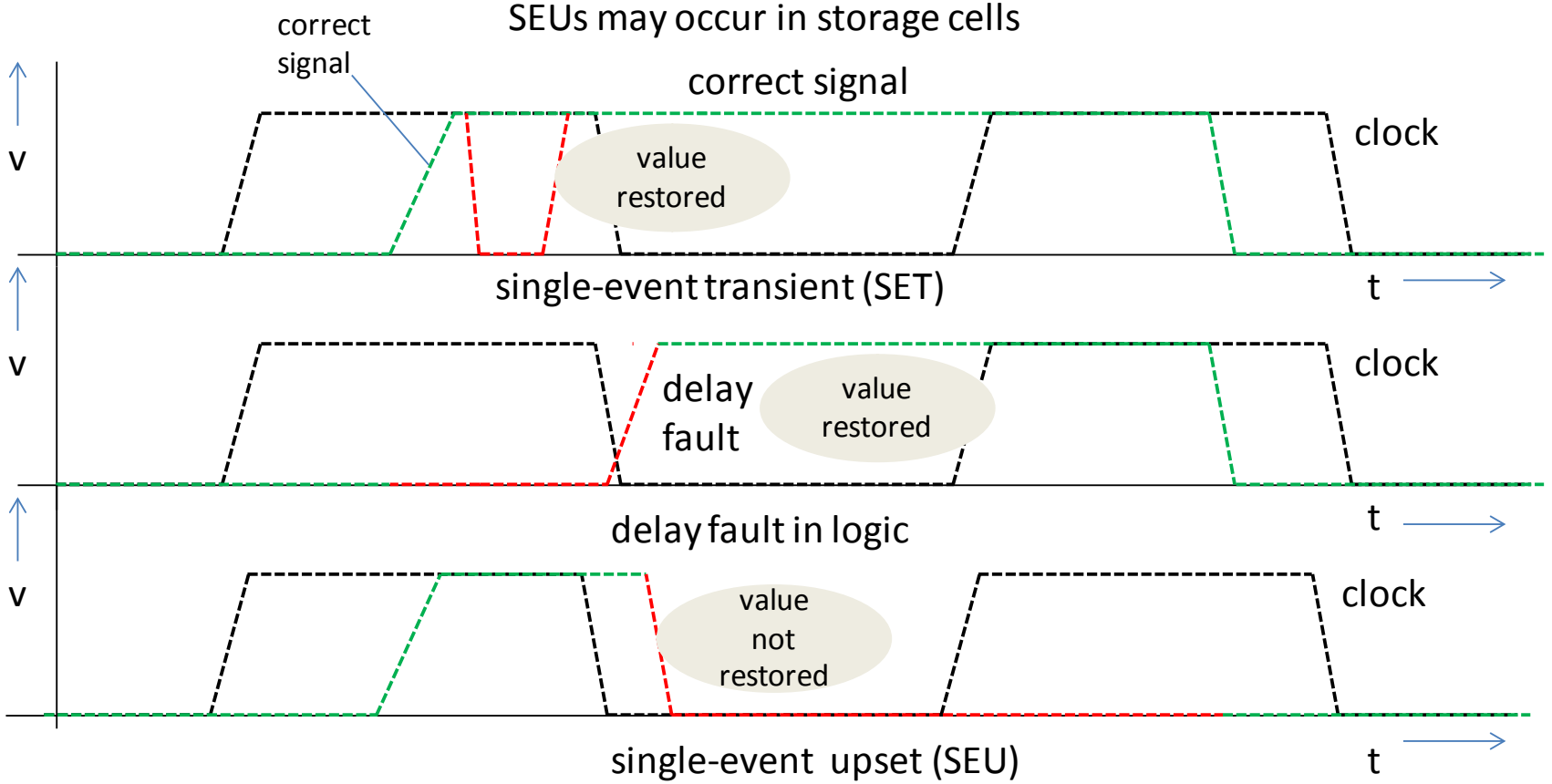
Plus: Transient Memory- Errors !

- Cells stuck-at 0 / 1
- Couling between memory cells
- Data-Retention-Fault (storage too short / loss of information, only DRAMs)

Transient Faults and Delay Faults

SETs typically occur in logic

SEUs may occur in storage cells



Transient faults or SEUs that last longer than a clock cycle T are not detectable by most methods targeting SETs or delay faults !

Transient Faults

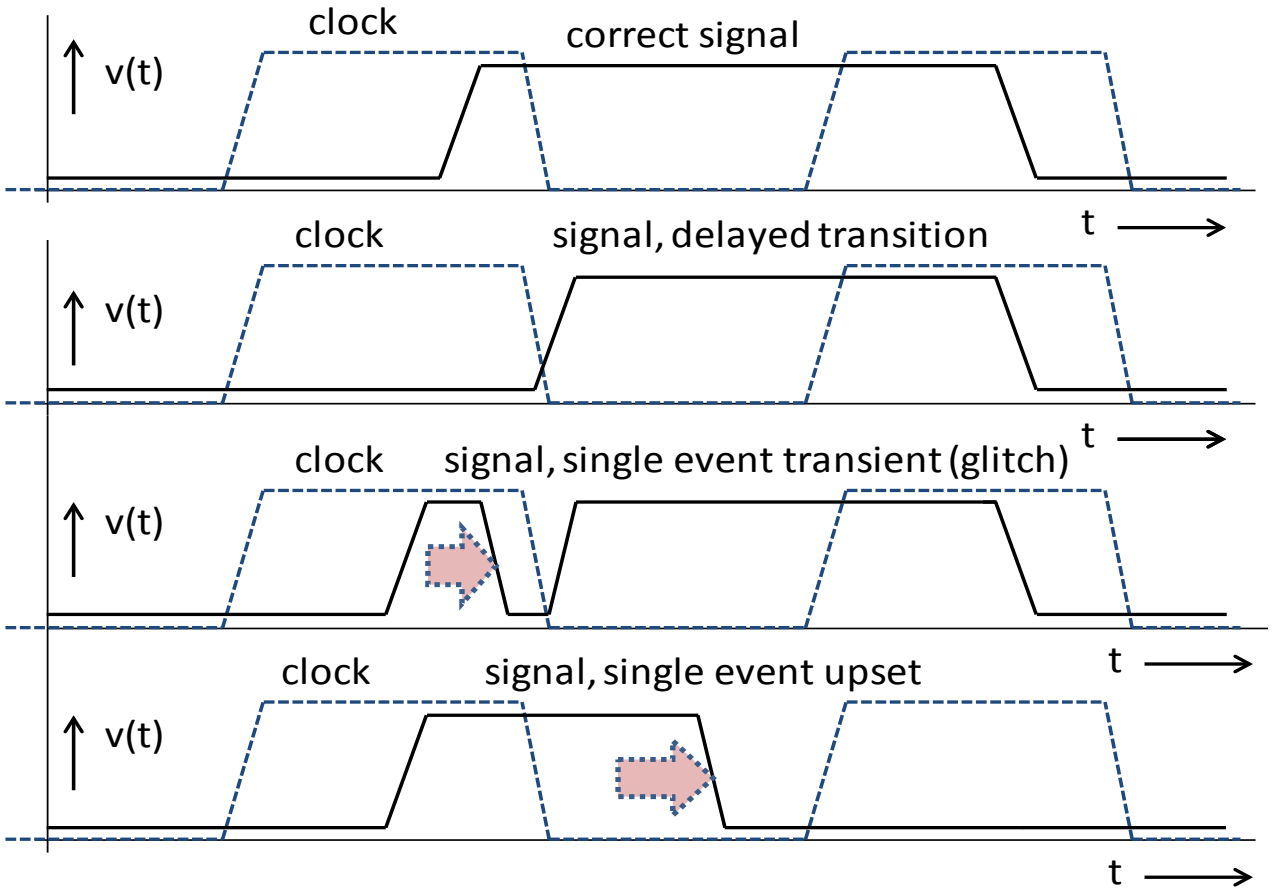
- Single even upset (SEU): A circuit node or storage cell changes status due to radiation.
- Multi-Event Upset (MEU): The same, but for several nodes / cells.

… typically happens in storage nodes /flip-flops / memory cells.

- Single event transients (SET): A circuit node gets the wrong opposite value, but returns to the correct value by itself.
- Multi-event transient (MET): The same, but for several nodes / cells.

… this typically happens to nodes in CMOS logic. Also to flip-flops with „active“ inputs.

SEUs, SETs, Delay Faults



SET: Single Event Transient, returns to correct value. Typical in CMOS logic.

SEU: Single Event Upset, does not returns to correct value. Typical in flip-flops and memory.

Also Multi-Event Transients (METs) and Multi-Event Upsets (MEUs) affecting more than one circuit node are common, but much less frequent !

Software Faults

Software does not age by itself. Therefore software faults always come from faults in algorithms or in implementation.

Except for faulty specifications, faults in software may be found by formal methods. But this is practically possible only for relatively small (embedded!) pieces of software. Formal verification fails for large software units.

Pracitically, software is „validated“ by testing. This is not exhaustive.

Software-Funktion that are often used are well tested. However, in systems like cars, planes, trains etc. there is also software that is scarcely used, e. g. in order to handle exceptions and emergencies. This software is not that well-tested and typically exhibits errors just by the wrong time !

Of course, faults in memory cells can affect software !! But here error correction coding (ECC) seems to work reasonably well !