Error Resilience in Digital Integrated Circuits

Heinrich T. Vierhaus

BTU Cottbus-Senftenberg

Computer Engineering Group

Outline

1. Introduction
2. Faults and errors in nano-electronic circuits
3. Classical fault tolerant computing
4. Transient faults and delays
5. Detection and correction of permanent faults
6. Software-based self repair
7. Error prevention at minimum cost
8. Summary and Conclusions

Systems Dependabilty

Cyber-physical system

Software-free system

German ICE Train (1990-2005):
Had serious temperature
problems in winter and summer.
Sometimes breaks down in operation
after less than 20 years life time.

Prussian class P8 locomotive (1906-1923):
Gave dependable service in Germany
and Poland for more than 50 years.
Record breaking 60 years of operational
life in Poland !

What does Resilience Mean ?

You can handle fault and errors or mis-use without „killing“ the systems function.

Aspects:

Detect and correct faults

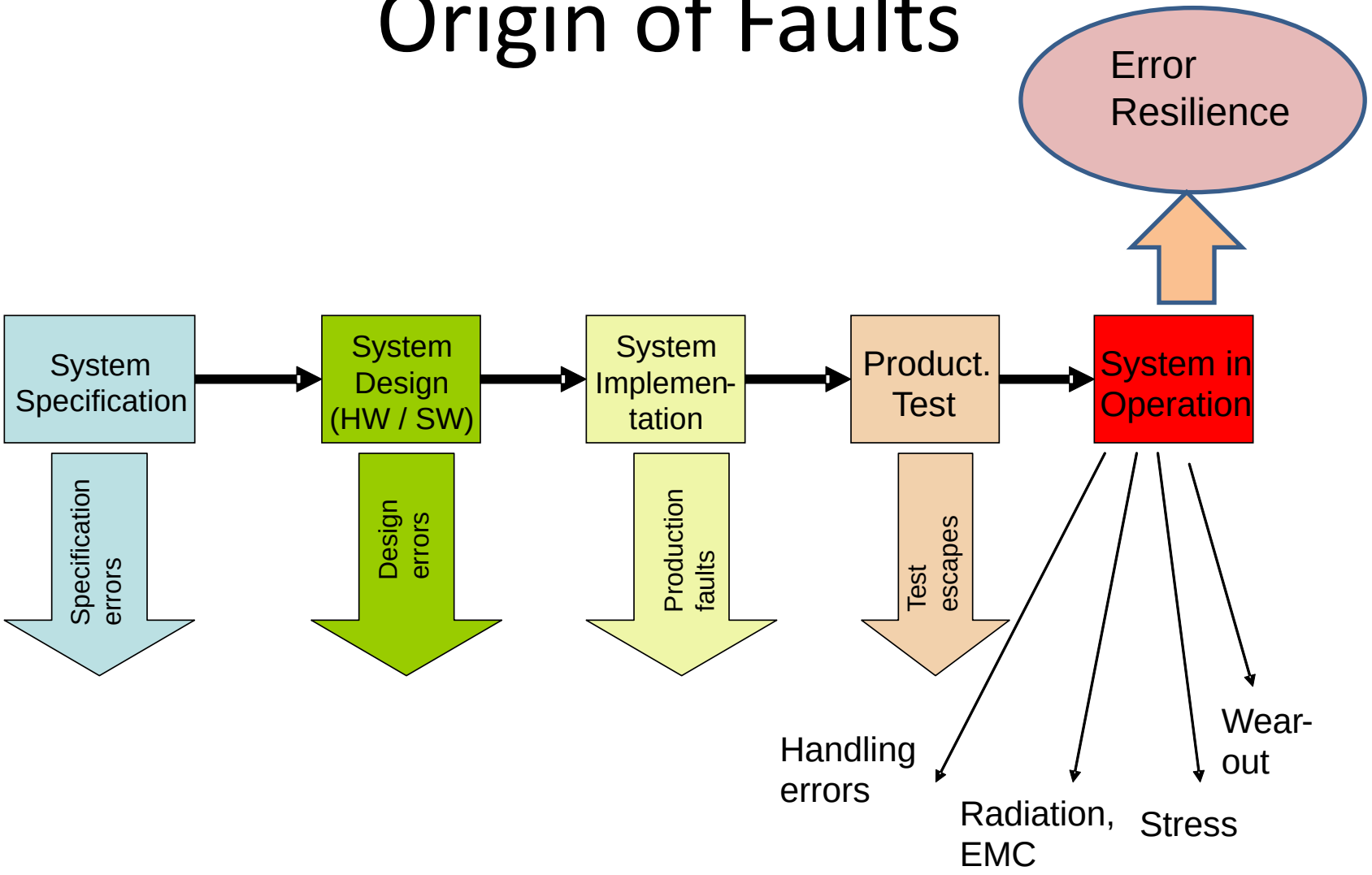
Detect and tolerate faults

Detect security problems and handle them in an appropriate manner

Handle errors by re-organisation of the systems to exclude faulty units, often with reduced performance

Observe critical parameters (heat, power, leakage) to predict faults, failures etc.

Origin of Faults



Changing Sources of Faults

- **Software bugs continue playing a role, but in general software dependability has improved a lot, despite ever increasing complexity.**

- **Digital hardware (processors) was highly reliable in the past.**
  This seems to be changing with nano-technologies:
  - Increasing vulnerability to transient fault effects (from radiation and / or noise),
  - Much stronger aging effects limiting the „operational“ life time of microelectronic circuits and systems to a few years rather than decades !

Thermal stress plays an important role !!

Furthermore:
Recent reports from conferences indicate that aging processes starting with „intermittent“ faults which may become dominant over transient faults !

Lowering supply voltages below 1 V is beneficial for power savings, but may introduce new and enhanced sources of errors !

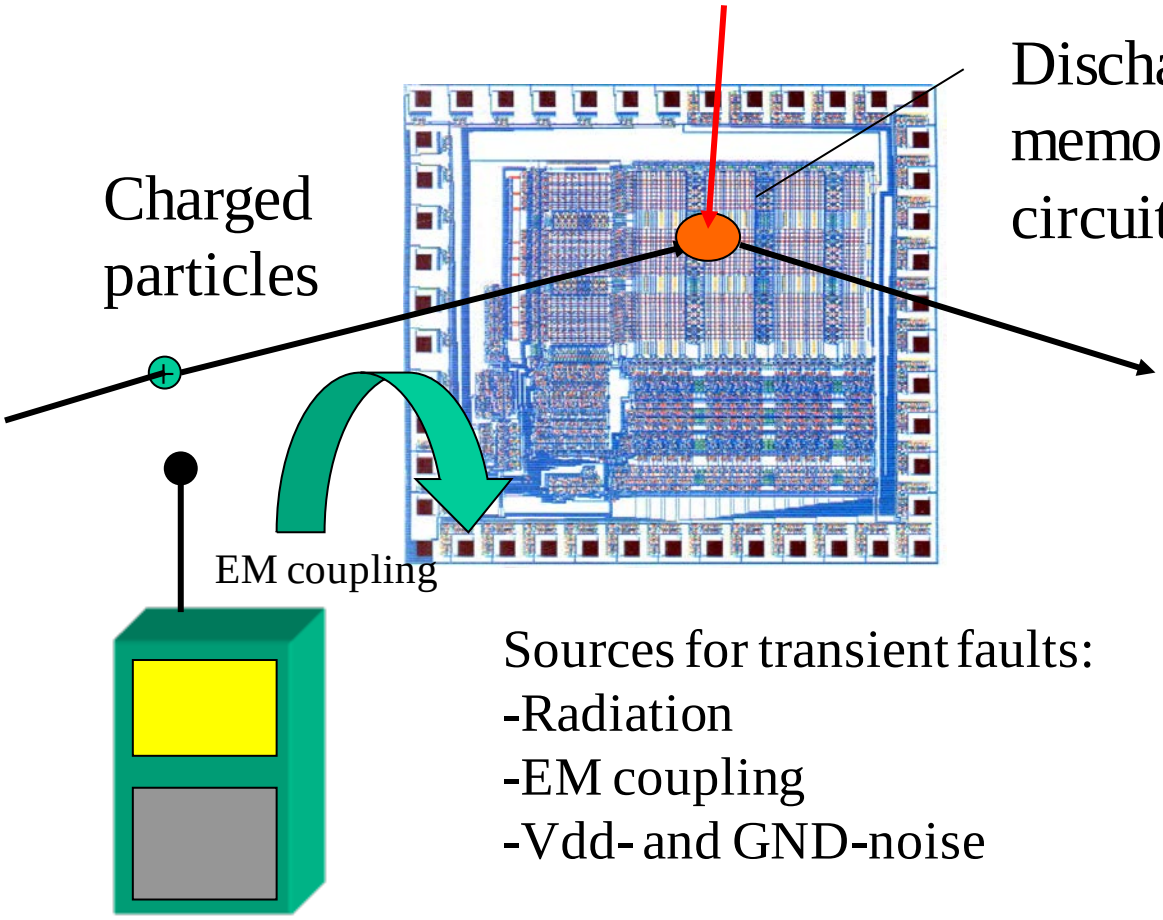
2. Faults in Nano-Electronic Circuits

The technology of ultra large-scale integration is driven by mobile communications. Smart phones have to be smart and cheap, but not ultra-reliable, limited lifetime of 2-3 years is well acceptable !!

ICs fabricated in advanced technologie with dimensions down to 14 nm are:

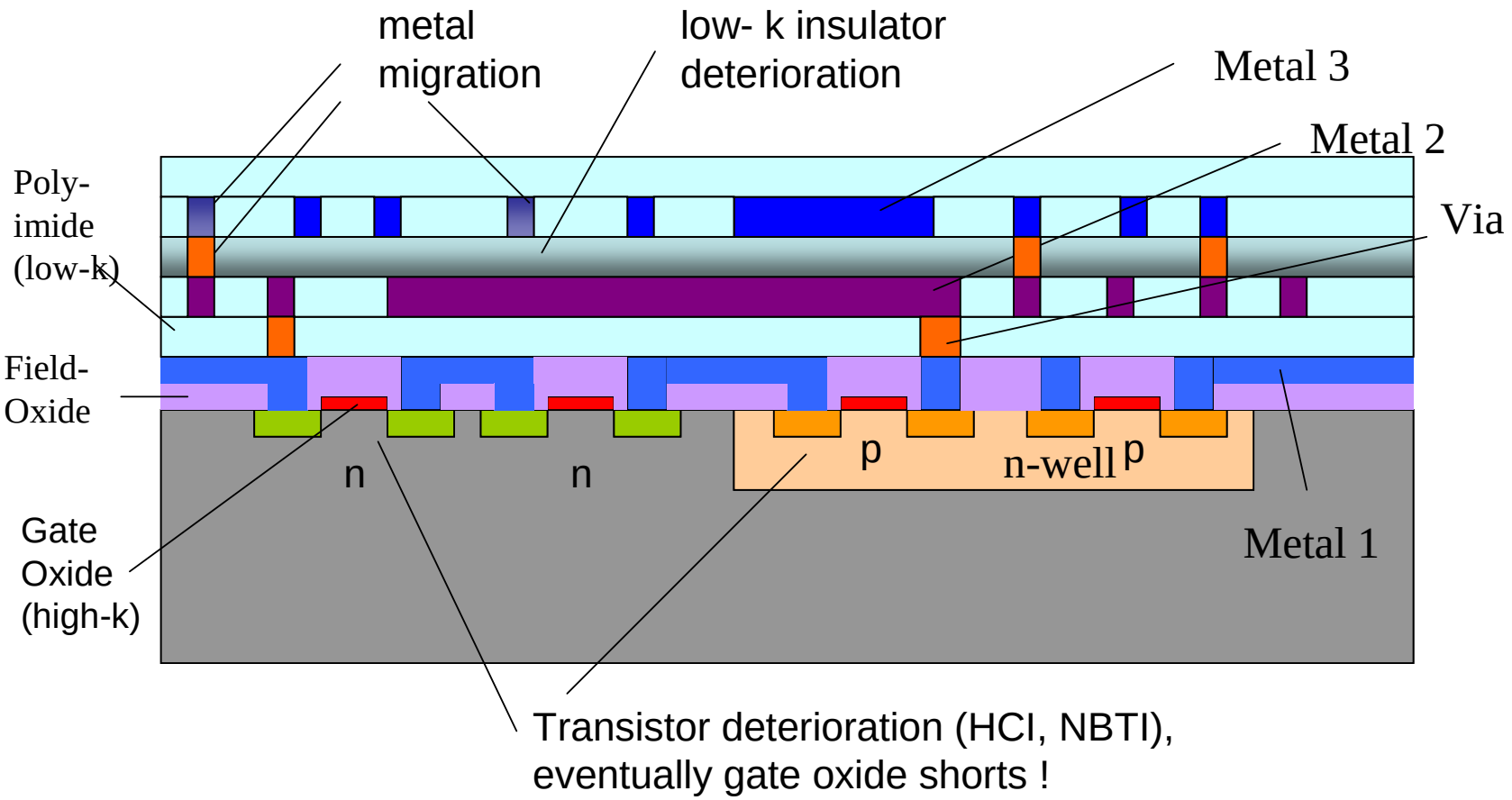
- more volatile to external influences like **particle radiation** and electro-magnetic radiation,
- subject to **aging mechanisms** that modify circuit parameters such as transistor threshold voltages,
- subject to **sudden „deadly“ fault effects** such as „dielectric gate rupture“, which cause permanent faults.

Transient Fault Effects

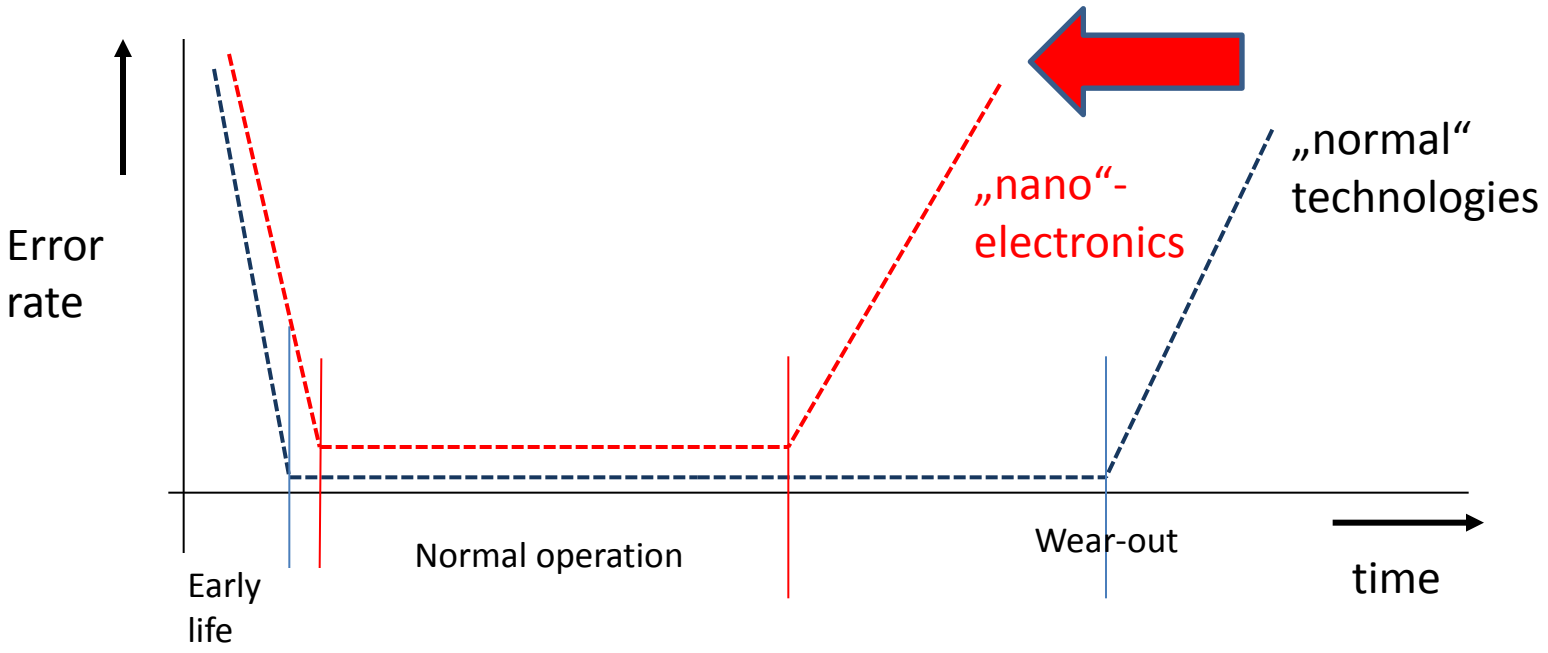


Sources for transient faults:
-Radiation
-EM coupling
-Vdd- and GND-noise

Permanent Faults



Life Time „Bathtub“ Curve



Devices in nano-technologies show stress-induced aging and then deterioration of critical parameters (threshold voltage, switching speed).

3. Classical Fault Tolerant Computing

Fault tolerant computing is an old technology. Traditionally ist was developed for:

- ***Avionics (electronics in airplanes)***
- ***Space flight vehicles***
- ***Nuclear power stations***

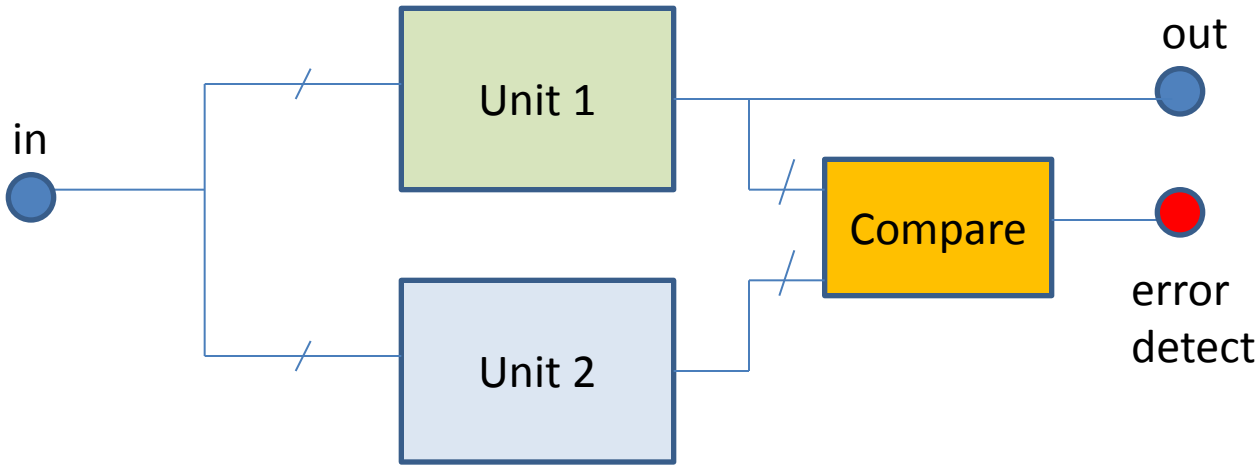
because of high-radiation-environments and

- ***automotive electronics***
- ***medical electronics***

possibly because of electromagnetic coupling problems.

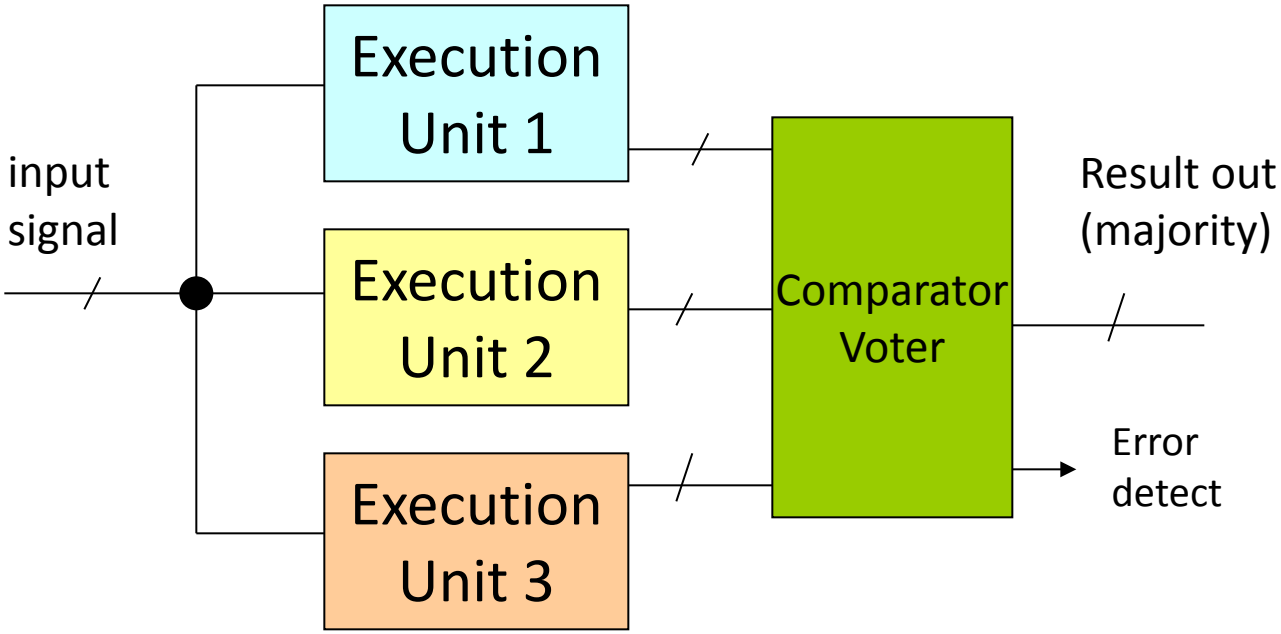
Emphasis has been on the detection and compensation of transient fault effects.

Duplication and Comparison



- Applicability to any type of fault (permanent, intermittent, transition)
- Fast error detection within the same clock cycle
- More than double overhead in hardware
- More than double power !
- Back-up in case of de-funct units !

Triple Modular Redundancy (TMR)

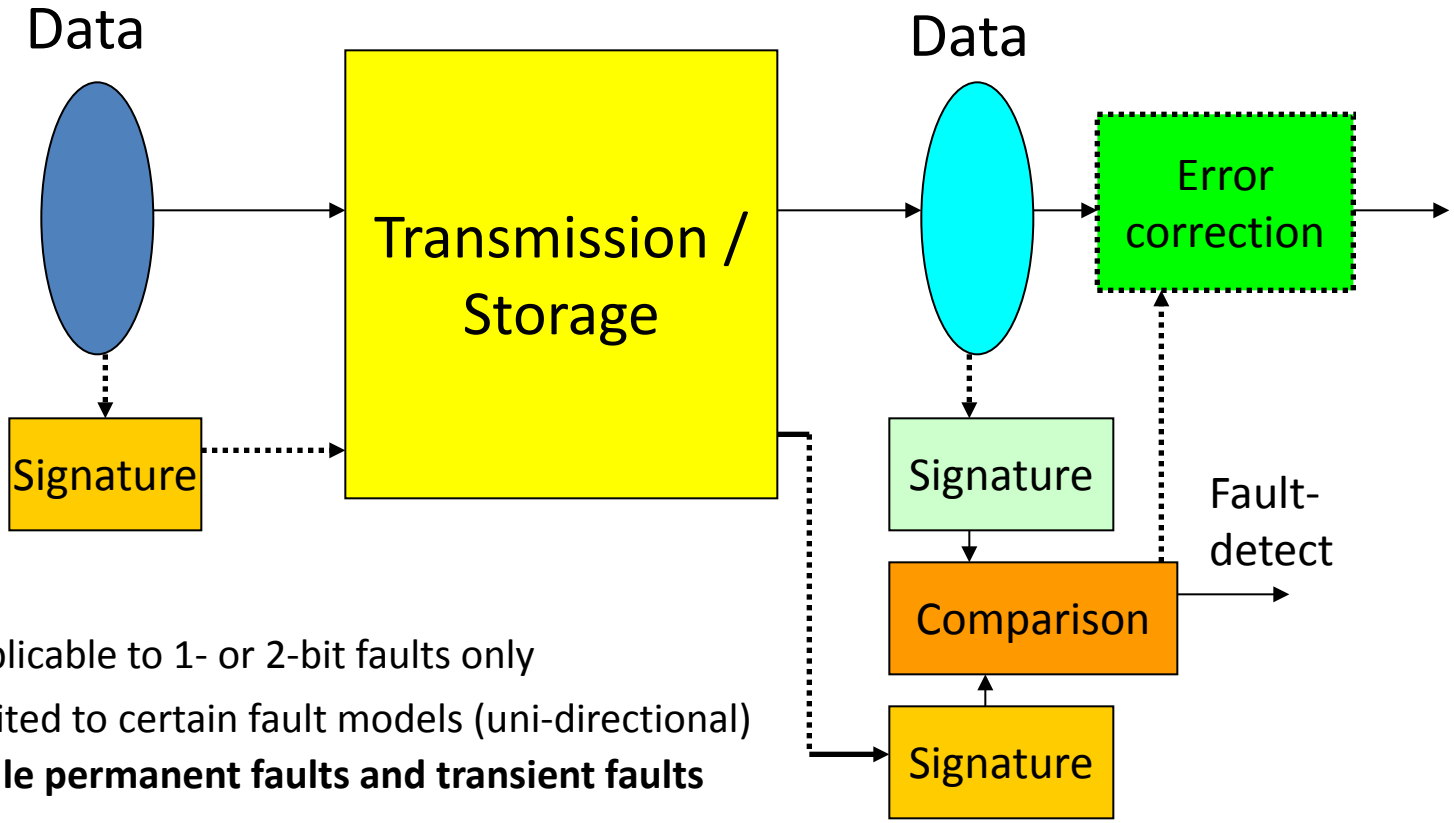


- Can detect and compensate almost any type of fault
- Overhead about 200-300 %, additional signal delays
- The voter itself is not covered but must be a „self checking checker“

Standard (by law) in avionics applications!

- Back-up in case of de-funct units !

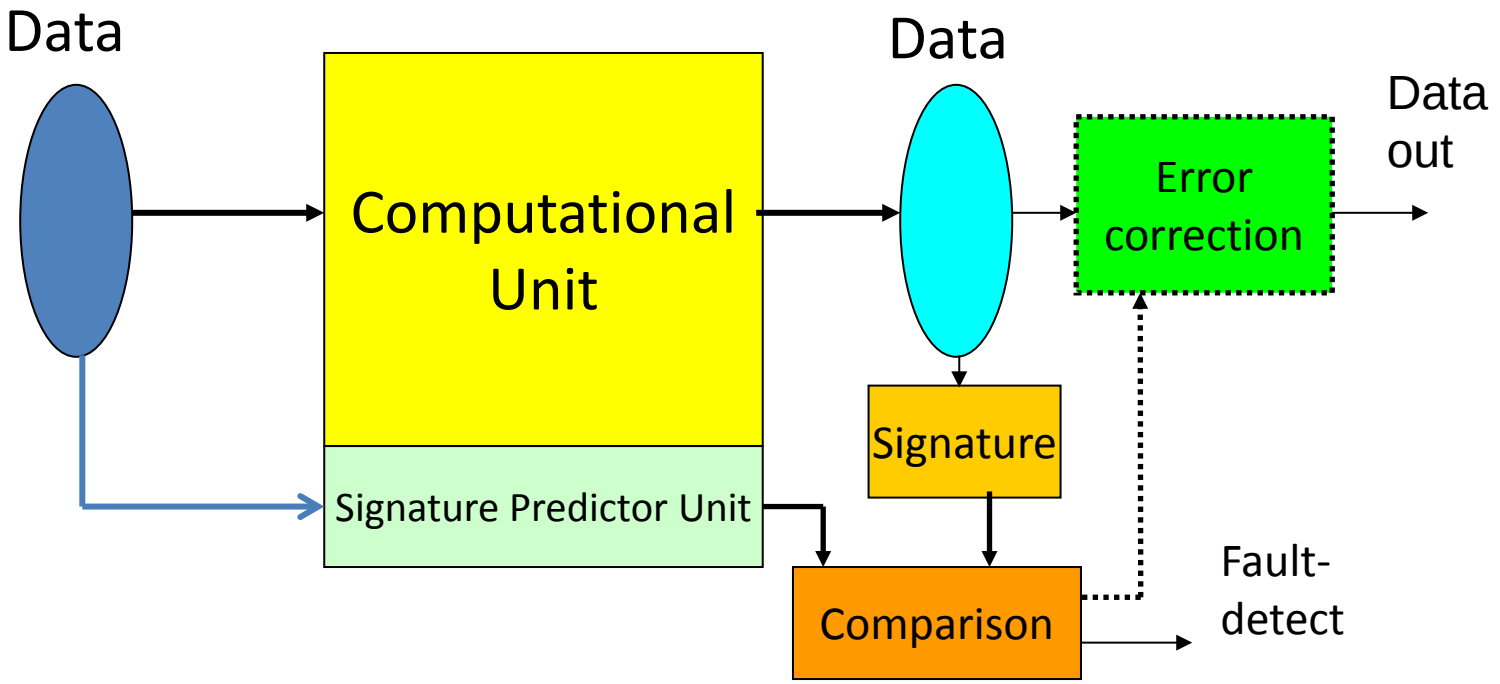
Code-Based Error Detection



- Often applicable to 1- or 2-bit faults only
- Often limited to certain fault models (uni-directional)
- **Can handle permanent faults and transient faults**

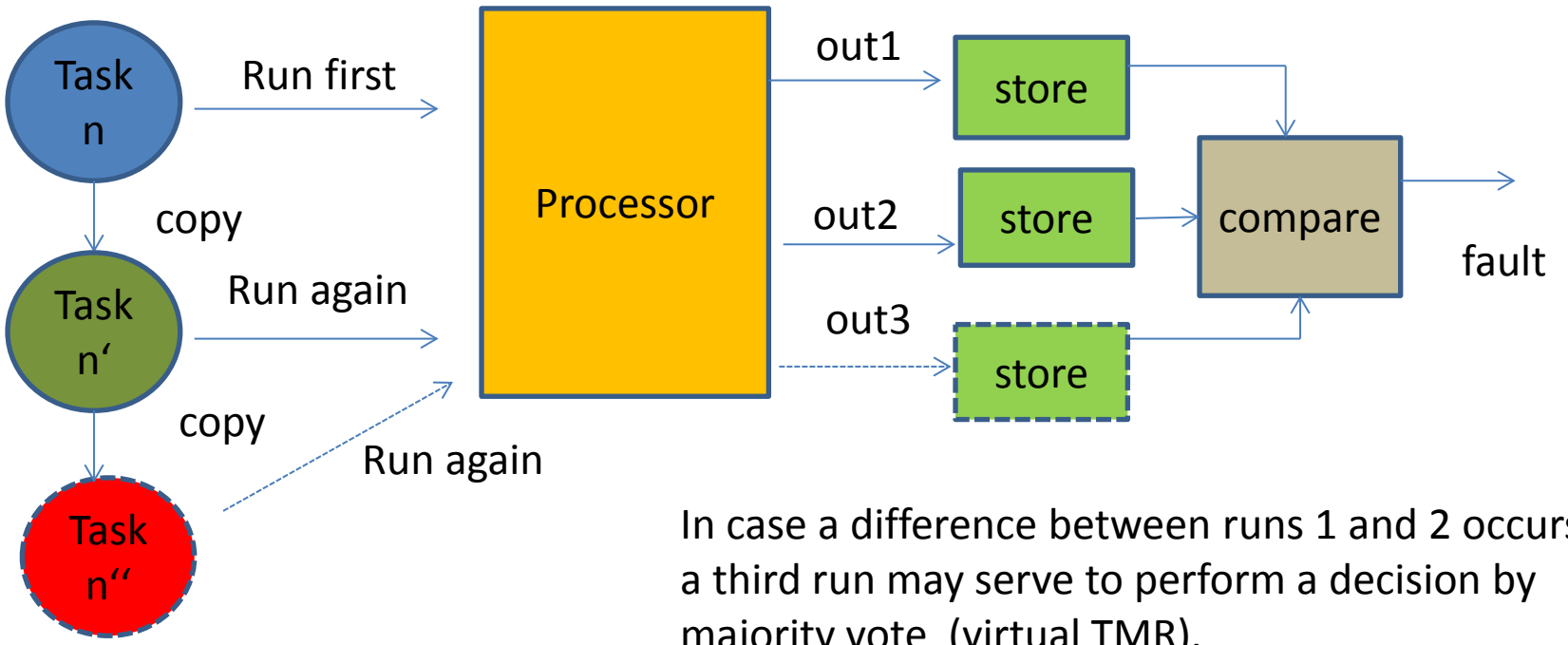
- No back-up in case of de-funct units !

Coding with Predictor Unit



Problem: The signature predictor unit may be almost as expensive as the computational unit itself, but cannot replace it in case of a permanent fault !

Software-Based Error Detection & Correction by „virtual“ TMR in Time



In case a difference between runs 1 and 2 occurs, a third run may serve to perform a decision by majority vote (virtual TMR).

- Little extra hardware
- Applicable to transitional faults only, does not work for permanent faults !
- Double / triple time, double / triple energy !!

Reliability and Life Time

During the normal life time of the system, duplication or triplication can enhance reliability significantly. But also area and power consumption are about triplicated.

And by the end of normal operating time (out of fuel / steam) all three units in TMR - systems will fail shortly one after the other !!

Reliability enhancement is not equal to life time extension !!

4. Transient Faults and Delays

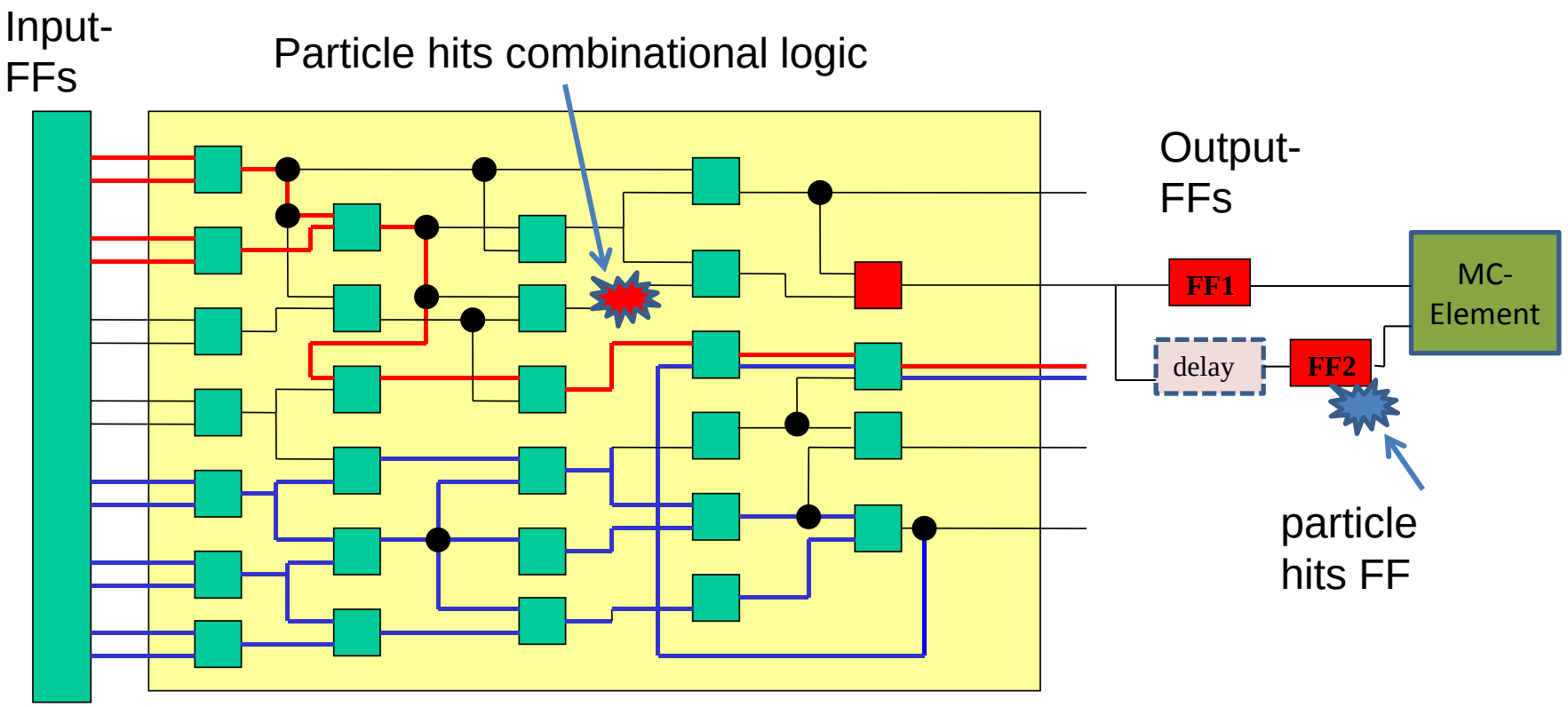
In processors, combinational logic is primarily used in two ways:

1. ..in control logic, which is mostly not extremely time-critical,

2. ..in arithmetic or logic blocks which are part of processor pipelines.

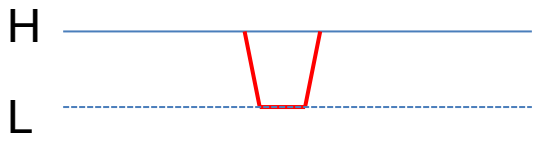
With power dissipation problems in processors getting worse and worse with structural down-scaling, processor designs that allow for down-scaling the supply voltage down to 0.6 V are badly needed.

Catching Transient Faults



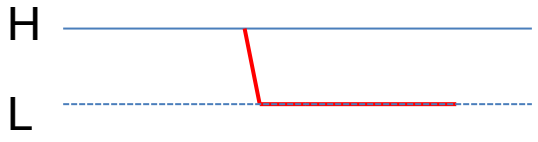
Events Triggered by Transient Faults

Single event transits:



In combinational logic, a short „glitch“ is triggered by radiation, after which the circuit node affected „returns“ to the correct value

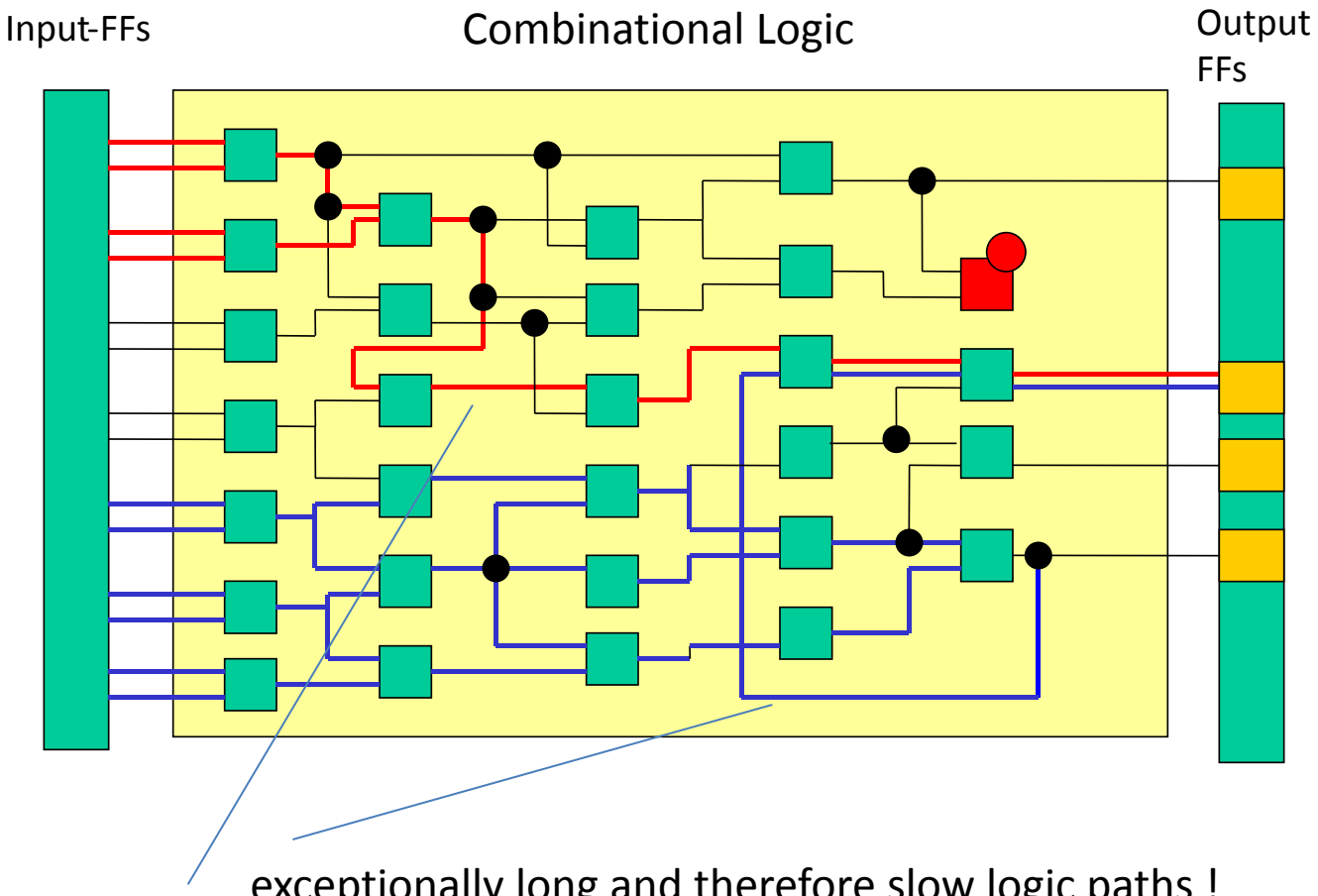
Single event upsets:



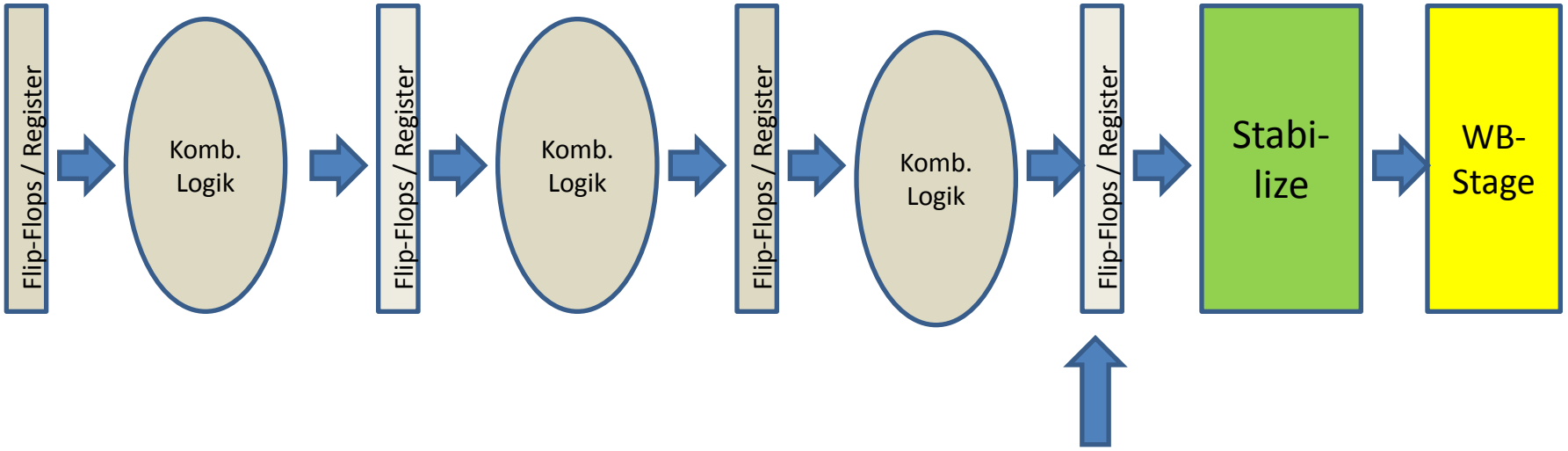
In storage elements such as flip-flops, a particle may set the node to a false value until the next „write“ event or the next clock cycle.

… and a single particle may even trigger multiple upsets !

Delay Problems



Pipeline Organisation

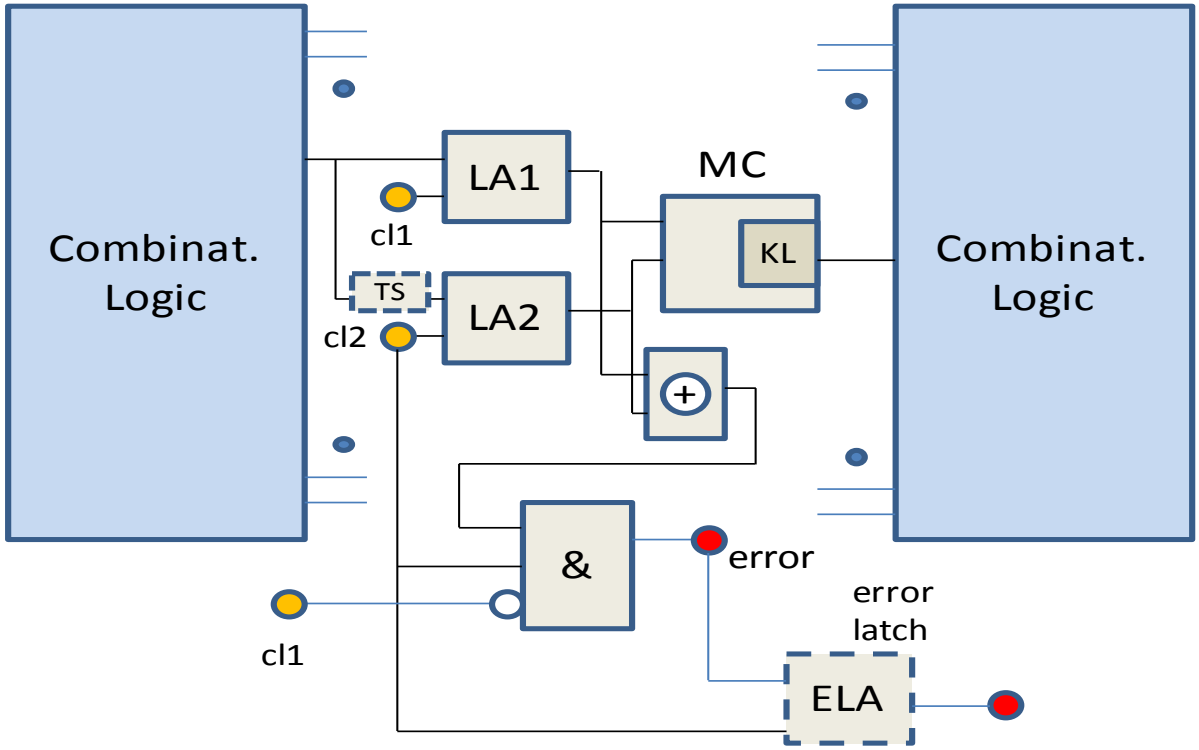


„RAZOR“ technology: Detect and repair late-arrivals of transitions, induced by statistical variations of path delays for low VDD (Univ. of Illinois).

Delays and Transient Faults

- Units designed for delay fault correction will not automatically detect or even correct transient faults.
- So far, delay fault detection / correction has been applied to known „longest“ paths only because of relatively high cost.
- Advanced „Bubble Razor“ concepts are known which can detect / correct permanent and transient faults by different machanisms.
- … but if applied to all outputs for complete coverage the overhead may be above 100 % !!

Muller-C plus Comparator



The Muller C-element acts as a „glitch“ filter. Delayed transitions are detected by the XOR comparator. But the MC-element passes the wrong value in case of a delayed transition.

A pipeline „stall“ mechanism is needed !

5. Detection and Compensation of Permanent Faults

- **Code-based methods: (e. g. Hamming Code)**

They cover only relatively few faults bits, typically one or two. Then fail if transient faults come on top of permanent faults.

- **Triple modular redundancy (TMR)**

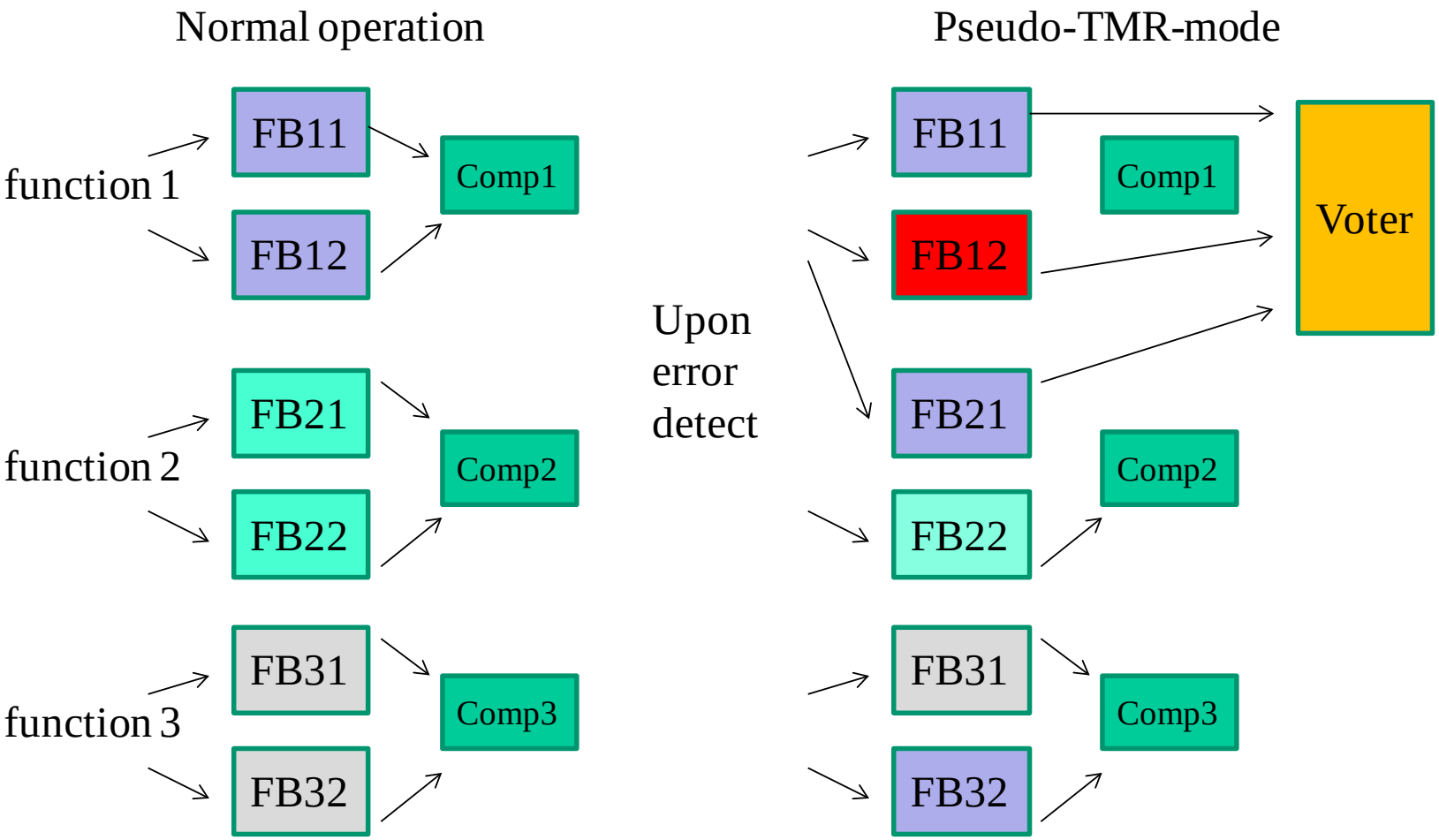
They will produce the correct result with only moderate delays, often within the same clock cycle. But more than 200 % extra hardware and extra power !

- **Virtual triple modular redundancy**

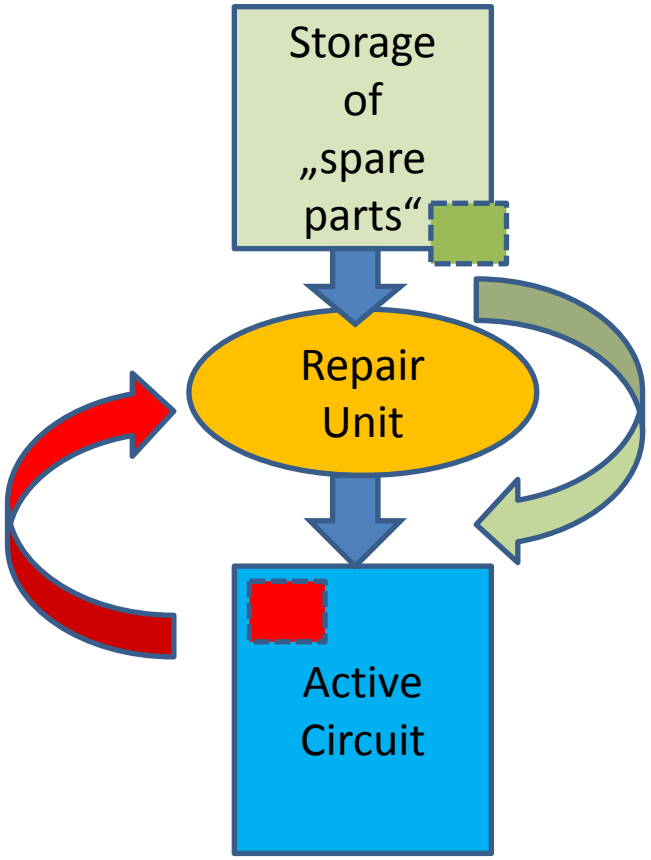
Either re-uses the same hardware several times (then not suitable for permanent faults !) or borrows extra hardware from „next door“ units.

Reduces peak power and hardware at the cost of extra time. But typically needs a duplication of power for fast on-line fault detection !

The Pseudo-TMR-Scheme



Self Repair or „Self Healing“

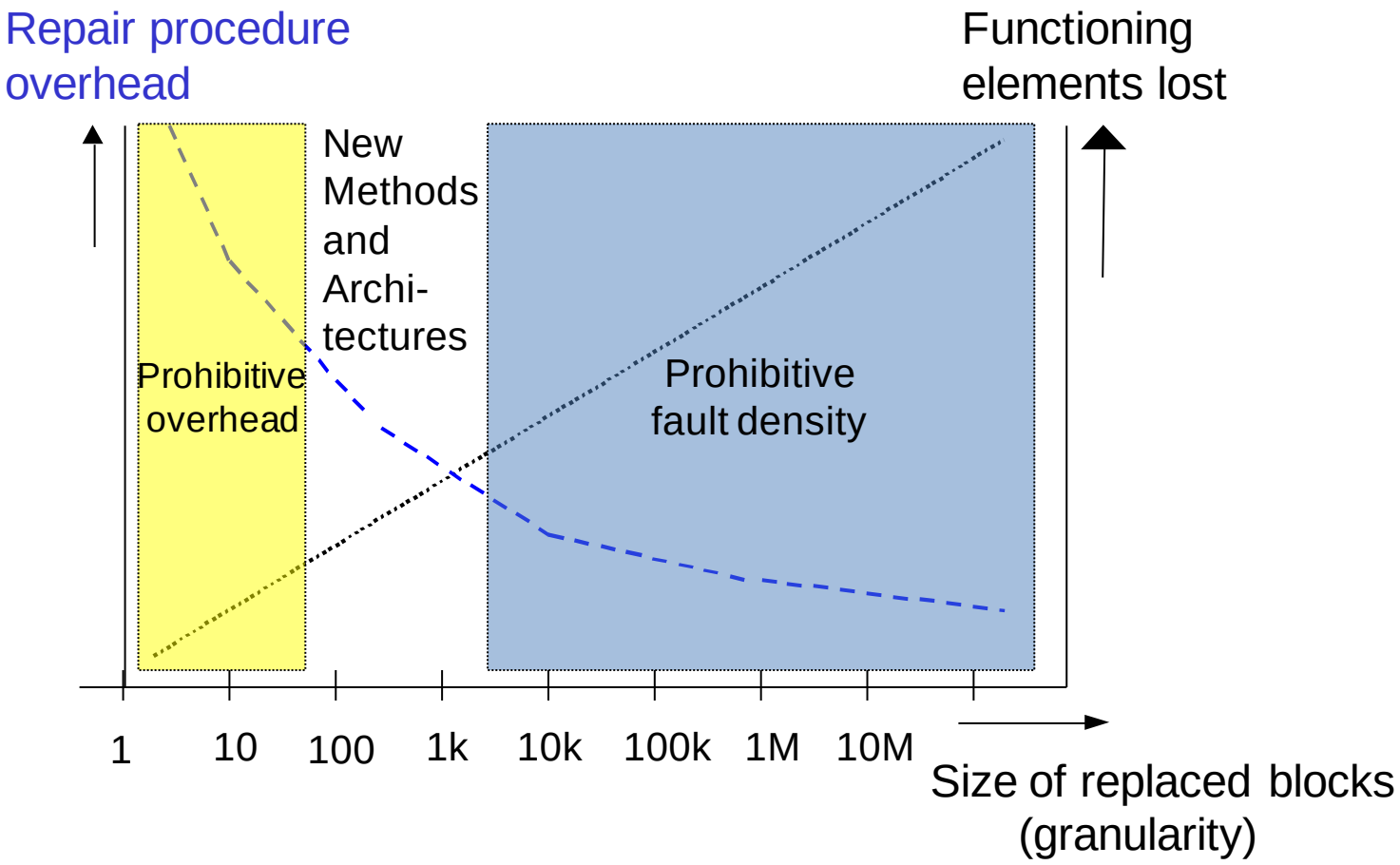


Upon registration of a permanent fault, the circuit undergoes an „off-line“ process of re-configuration for replacement of (permanently) faulty units.

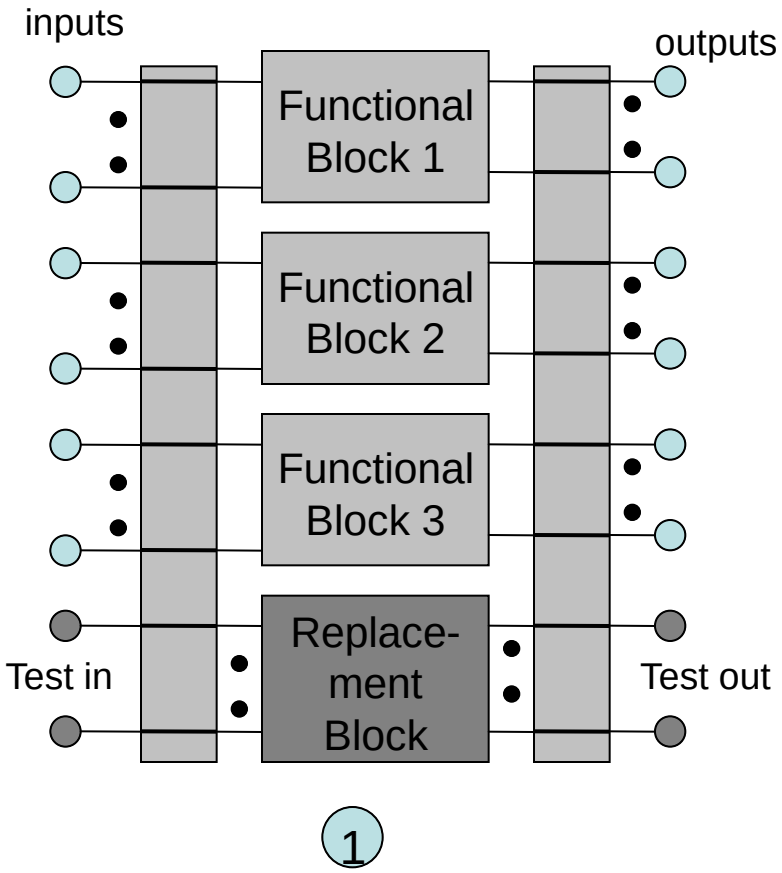
- Less extra power than with TMR, therefore longer system life time expected,
- The repair process is complex and slow, requiring in-system diagnostic testing.

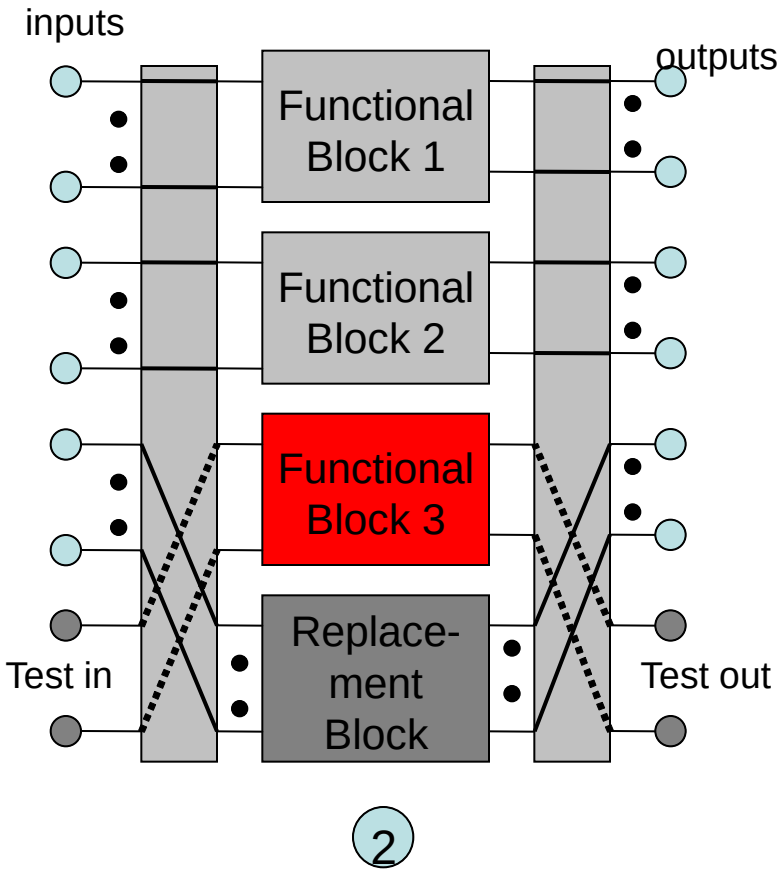
The repair process must be done off-line, while the system is „at rest“ or at „start-up“.

Granularity of Self Repair

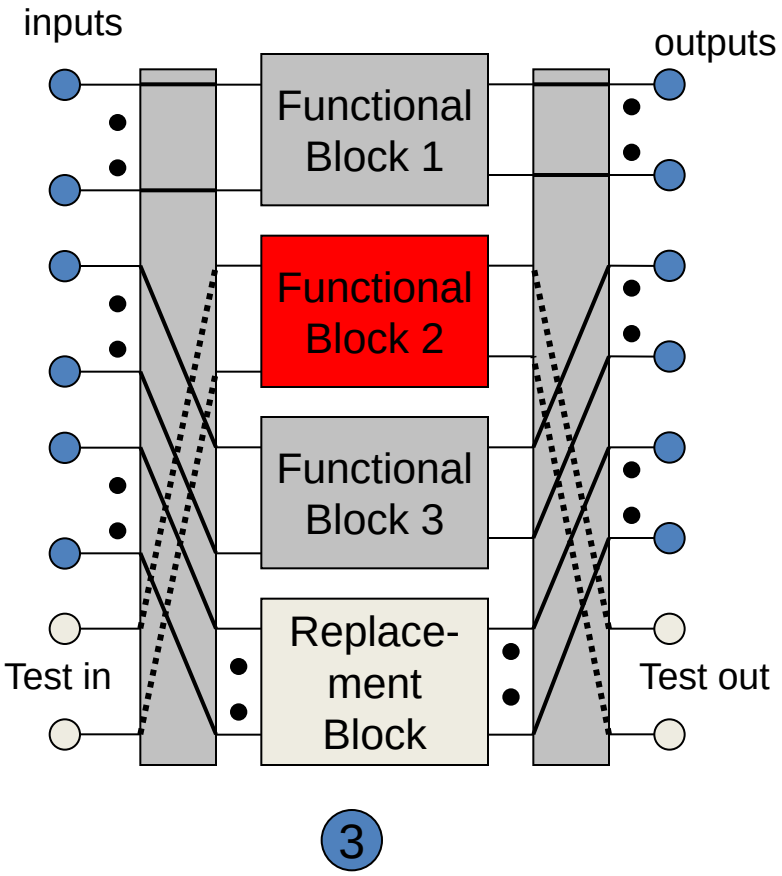


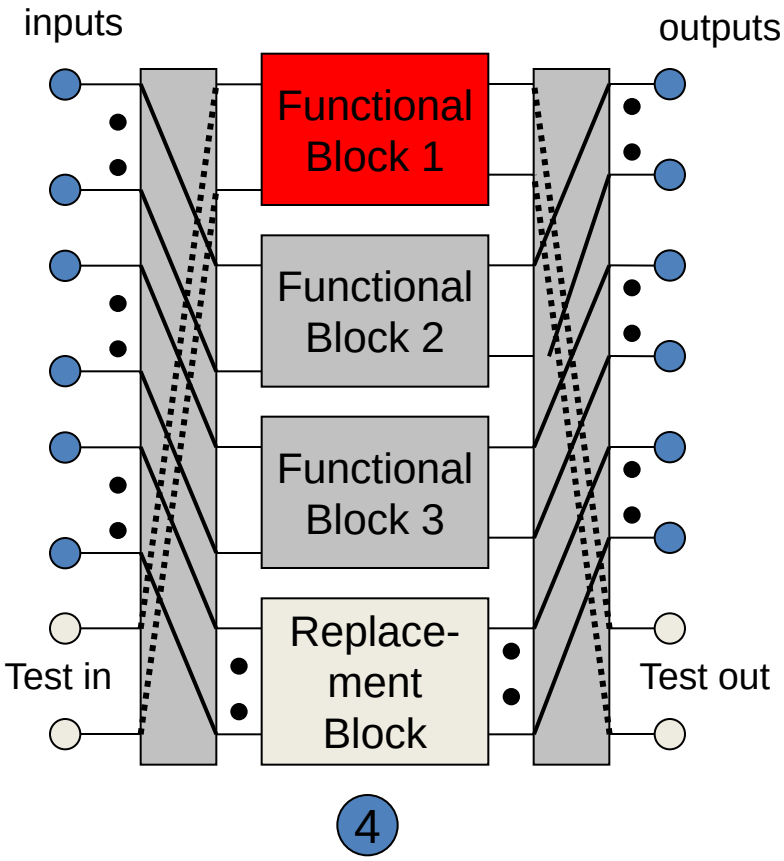
Regular Blocks and Backup





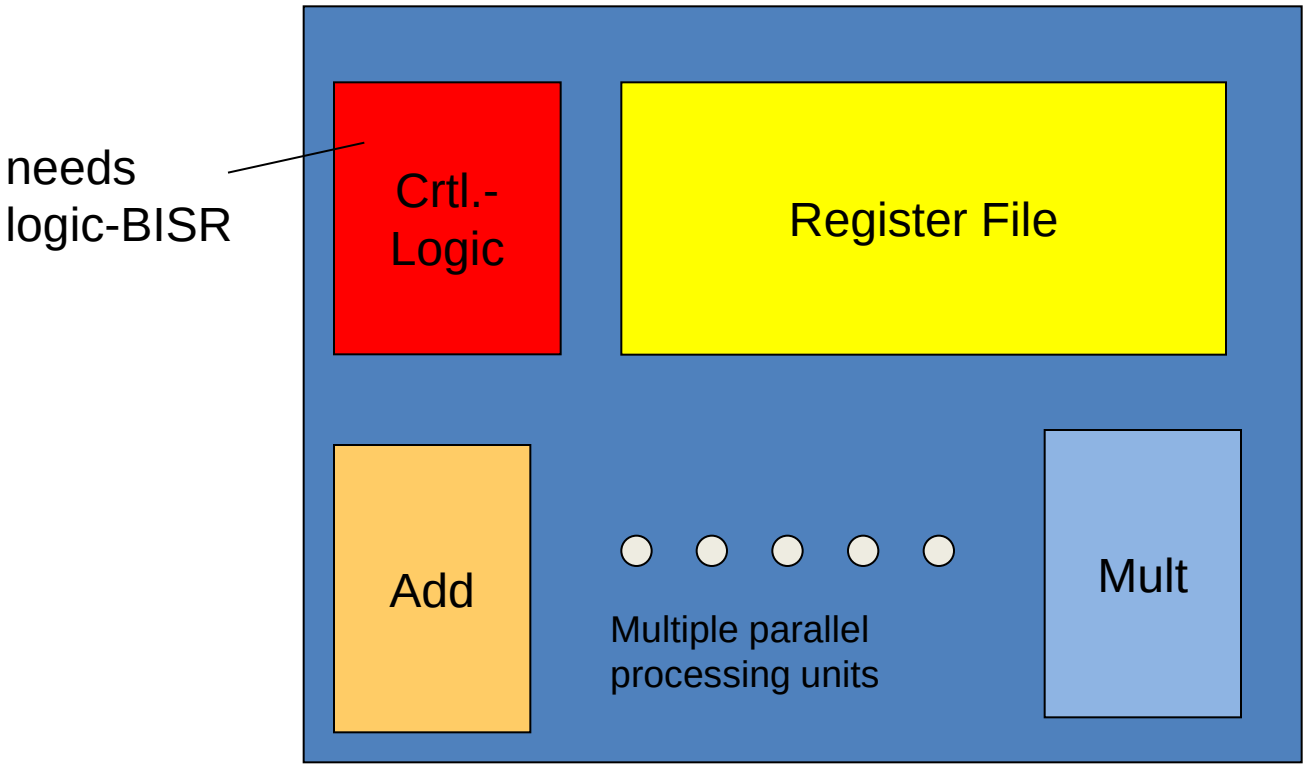
Regular Blocks and Backup (2)





6. Software-Based Self Repair

Structure of a regular multi-slot-processor
(e. g. Very Long Instruction Word Processor, VLIW)

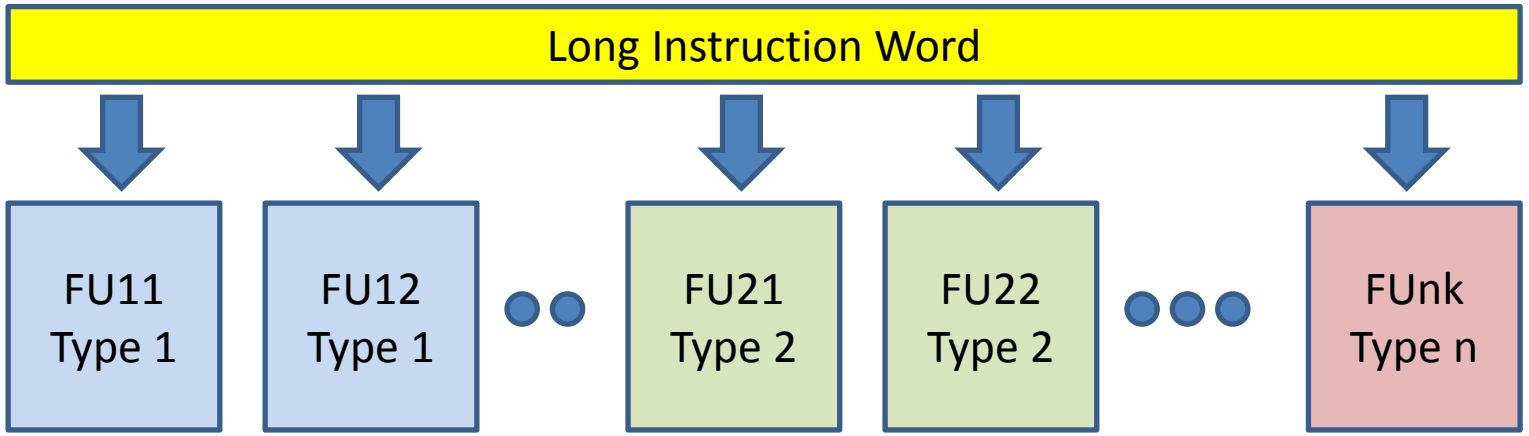


The processor can perform a software-based self repair process by re-allocating functional units to tasks according to the fault condition !

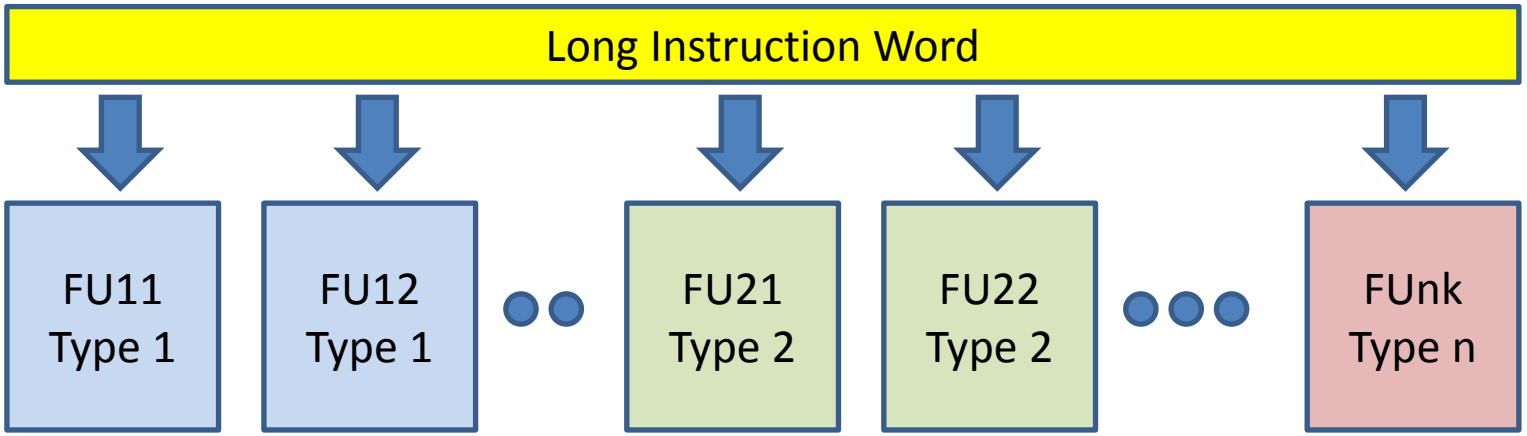
Regular Processors

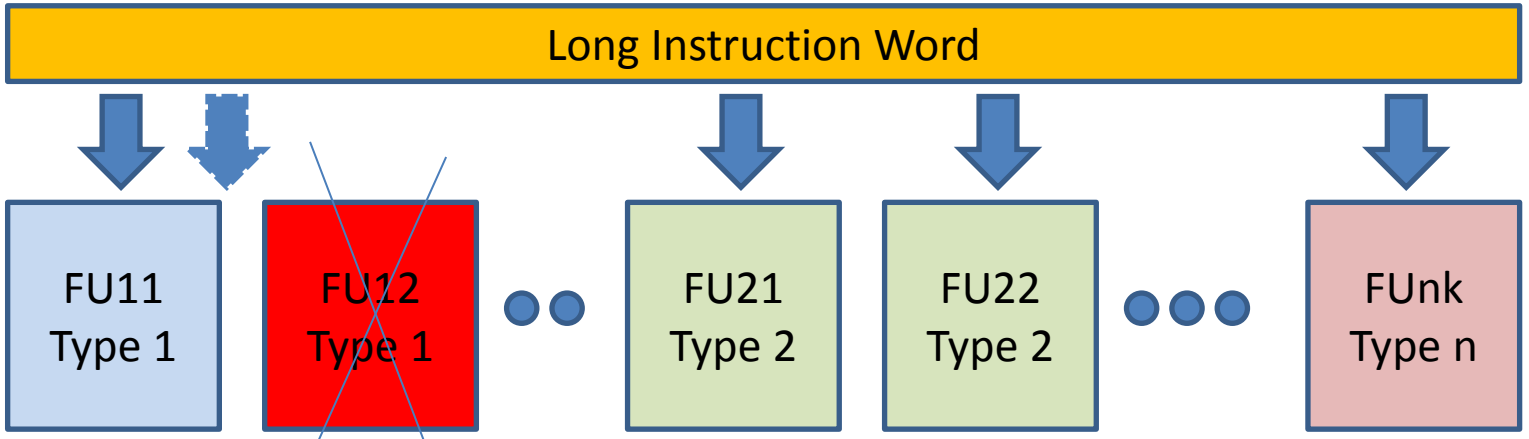
They contain several building blocks (slots) of equal type and structure such as adders, shifters, registers, etc.

In case of a permanent fault in one unit, the processor can be „re-organized“ to continue working with fewer units and reduced performance. Therefore the schedule and allocation of operations must be re-organized. This can be done in software by „re-allocation“.

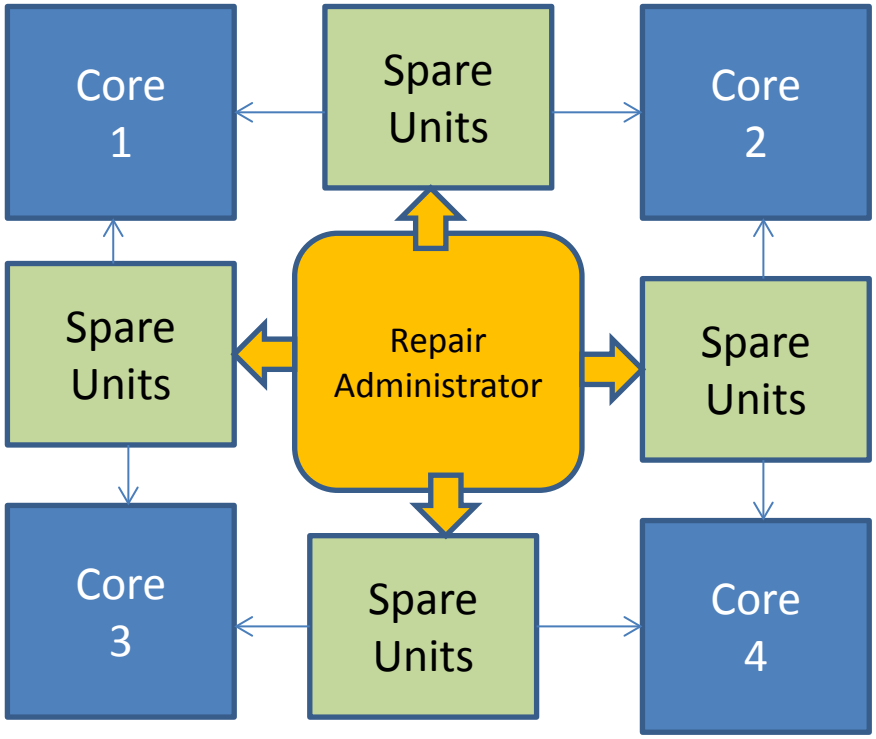


Processor Repair





Multi-Core- Units

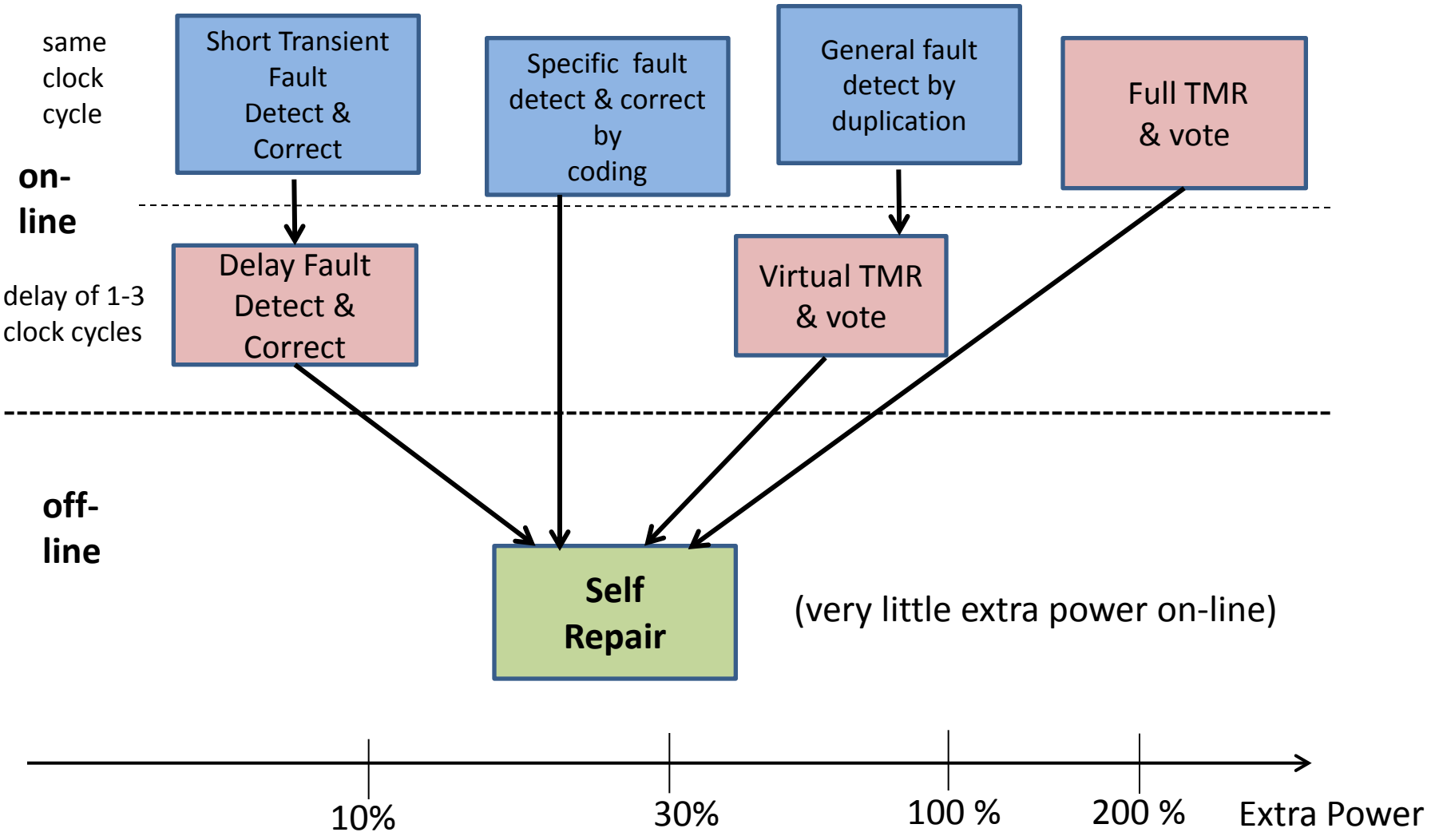


Spare units for self-repair can be allocated to different processor units in a processor cluster.

7. Error Prevention at Minimum Cost

What we need is a system-specific methodology that trades degrees of feedom in timing against requirements of dependability.

Strategies


same clock cycle
on-line
delay of 1-3 clock cycles
off-line
Short Transient Fault Detect & Correct
Specific fault detect & correct by coding
General fault detect by duplication
Full TMR & vote
Delay Fault Detect & Correct
Virtual TMR & vote
Self Repair
(very little extra power on-line)
10%
30%
100 %
200 %
Extra Power

Total Error Resilience

- Monitoring of critical parameters of ICs which indicate stress and aging (switching speed, currents).
- Detection and compensation of non-permanent faults at minimum cost in extra power using reserves in system timing parameters.
- Built-in self test with diagnostic features to identify faulty units.
- Built-in self repair (off-line) to replace defect parts and units that show wear-out-effects.

8. Summary and Conclusions

- **Error resilience** is a system- and application specific technology to „cope with faults in real operation“.
- Fault detection & correction that works for **any** type of fault is expensive in power and hardware.
- Monitoring circuits & systems to identify „weak spots“ becomes a must.
- Tolerating the fault for seconds or minutes before off-line repair may save a lot of efforts.
- Correction within the same clock cycle is expensive, 1-3 extra clock cycles may save half the overhead in power.
- Self repair based on „sleeping“ redundancy is cheap in terms of extra power, but is slow and may be used primarily for life time extension in case of „early life failures“.
- *We may need some extra functions in the operating system that serve error management, scheduling of self repair, and monitoring the inventory of spare parts !*

Thank You for Your Attention !

Questions ??