

Architectural Issues in Dependable Embedded Systems

H. Kopetz
March 2006

Today, in 2006,
computer technology is in the middle of a major
paradigm shift, similar to the transition from the
Mainframe to the *Personal Computer*
twenty-five years ago.

Outline

3

- ◆ Introduction
- ◆ The Technology Landscape
- ◆ The Effects of Ambient Cosmic Radiation
- ◆ Determinism and Order
- ◆ The DECOS Approach
- ◆ Conclusion

© H. Kopetz 3/30/2006

The Technology Landscape--Hardware

4

- ◆ The limits of *Moore's* law are becoming visible: power dissipation, physical feature size, reliability.
- ◆ The *memory wall* gets higher and higher (about 80 % of the transistors in the *Itanium* chip are for caches).
- ◆ The performance increase of a single processor from one generation to the next is proportional only to the square root of the increase in silicon area (*Pollack's* rule).
- ◆ The transient failure rate of sub-micron devices is increasing [both single-event upset (SEU) and single-event transient (SET)].
- ◆ Multi-computer chips (SoC) are appearing. The development cost of such a chip can pass the 100 Mio \$ wall--mass markets are needed to justify this level of investment.

© H. Kopetz 3/30/2006

Example: Cell Processor

5

Joint development of IBM, Sony and Toshiba, 90 nm process 250 Mio Transistors, 221mm², 4 Ghz, 250 GFLOPS, Development Cost > 400 Mio \$

QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

© H. Kopetz 3/30/2006

The Cell contains eight SPE Computers

6

Specialized Processor for SIMD-type data streams

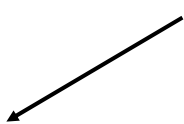
256 Kbytes of private memory (LS0-LS3), access latency is 6 cycles

32 bit instructions, 128 registers

Ca 15 mm²

QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

BIU less than 1 mm²



© H. Kopetz 3/30/2006

The Technology Landscape--Communication

7

- ◆ The widespread availability of a wireless communication infrastructure enables the ad-hoc detection and integration of services without any physical action--the coming of *situation aware* systems.
- ◆ Seamless integration of *Radio Frequency Identification* (RFID) technology with embedded devices (e.g., cell phones).
- ◆ Flexible transmission technologies (e.g., *spread spectrum, ultra wide band, frequency hopping*) that allow multiple users to share a given frequency band with minimal interference and reduce the power required per bit transmitted.
- ◆ Waveform-agile transmission methodologies (*cognitive radio*) that enable a wireless device to create ad hoc different types of communication links under software control.

© H. Kopetz 3/30/2006

The Technology Landscape--Embedded Software

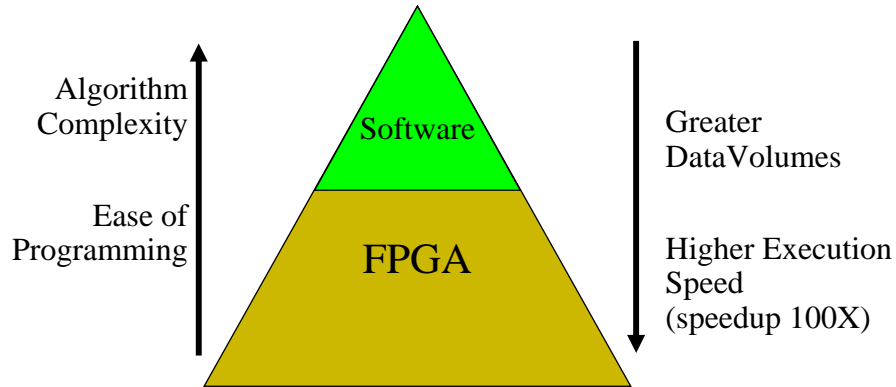
8

- ◆ *Uncontrolled system complexity*: The costs of design, verification, integration and maintenance are getting prohibitive.
- ◆ *Component-based design* elevates the design process to a higher level of abstraction--but many key issues are still open, e.g., the precise specification of component interfaces, the identification of *fault-containment units*.
- ◆ *The investment in software* is more long-lasting than the investment in hardware.
- ◆ *Security becomes a key issue*-- as embedded devices are integrated into the Internet, particular in wireless systems.
- ◆ The clear distinction between software and hardware is disappearing, e.g., *power-dissipation* is becoming also a software issue, not only in battery-operated devices.

© H. Kopetz 3/30/2006

New Implementation Choices

9



Example: Look for keywords in a set of documents:
(A and B) or (C and D)

Search for the occurrence of A,B,C,D in FPGA, connect the results in software

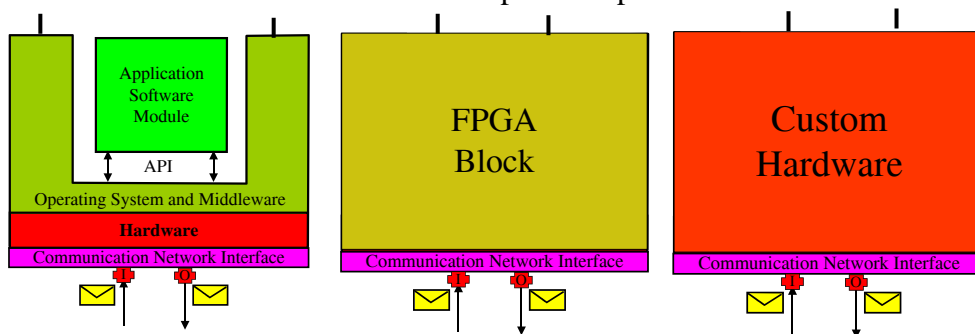
From: R. Chamberlain, Embedding Applications within a Storage Appliance Proc. HPEC 2005, p. 2

© H. Kopetz 3/30/2006

Different Types of Components

10

Local Interfaces--Open Components



The Communication Network Interfaces (CNI) of all three different types of system components should have the same *syntax*, *timing* and *semantics*. For a user, it should not be discernible which type of system component is behind the CNI.

© H. Kopetz 3/30/2006

User Expectations

11

- ◆ In a large embedded system that consists of a vast assembly of networked components that must operate 24 hours per day for 365 days per year the *occurrence of transient and permanent failures* of components and interconnects must be *considered the norm*, not the exception. Future system must thus include strategies and mechanisms that assure that the reliability of the *user-perceived system services* remains at an acceptable level despite the occurrence of these failures.
- ◆ In an ambient intelligence scenario, where a multitude of diverse embedded devices is fielded in a home, it cannot be expected that the end-user is willing to spend her/his time and effort to troubleshoot a misbehaving distributed embedded system. A system must thus be capable to diagnose its own faults and guide an untrained user to repair the system with minimal effort.

© H. Kopetz 3/30/2006

The Effects of Ambient Cosmic Radiation

12

The neutrons of the ambient cosmic radiation interact with the atoms of the semiconductor devices, giving rise to *soft errors*:

- ◆ Create *electron-hole pairs* that interfere with the electric charge that denotes the information contents of cell (*bit-flip*). Since this electric charge decreases with smaller feature size, disturbances become more probable when the feature size shrinks.
- ◆ Affected area in the order of a few μm^2 . Duration in the nano-second range (i.e., the duration of cycle time of a modern CPU).
- ◆ Single bit-flip most probable, double-bit failures possible, both storage (SEU) and logic (SET). No permanent failure of the device.
- ◆ At present, the sea-level failure rate is about 1000 FITs/megabit (1 FIT = 1 Failure in 10^9 hours, about 100 000 years)
- ◆ SoC failure rate of 1 000 000 FIT: one event per month, but not every event causes a severe failure (e.g., pixel on a screen).
- ◆ Increase: 3-5x at 1500m, 10x at 3000m, 100x at 10 000m

© H. Kopetz 3/30/2006

Mitigation Strategies w.r.t. Soft Errors

13

It is hardly possible to shield a device from the ambient cosmic radiation, however the effects of this radiation can be mitigated on different levels:

- ◆ Material selection to reduce the neutron interaction coefficient (Example: SoI *Silicon in Insulator*)
- ◆ Layout of electronic devices: larger devices
- ◆ Radiation-hardened circuit design
- ◆ Error detection and correction for SEU (e.g., 64 bit words requires 8 additional bits). Mitigation of SETs more difficult.
- ◆ High-level architectural means: triple modular redundancy (TMR) in space and/or time.

© H. Kopetz 3/30/2006

Integrity-Level of Application Domains

14

| Application | System MTF w.r.t. permanent failures (in years) | System MTF w.r.t. transient failures (in years) | Data-integrity requirement | Market volume | Examples |
|--------------------|---|---|----------------------------|---------------|------------------------|
| Low-Integrity | > 10 | > 1 | low | huge | Consumer Electronics |
| Moderate-Integrity | > 100 | > 10 | moderate | large | Present-day automotive |
| High-Integrity | > 1000 | > 100 | very high | moderate | Enterprise server |
| Safety-Critical | > 100 000 | > 100 000 | very high | small | Flight control |

© H. Kopetz 3/30/2006

The Dilemma

15

- ◆ The consumer electronics (CE) domain has the size to support the large development costs needed to build powerful SoCs.
- ◆ Since in the near future there is no need to harden CE chips to mitigate the consequences of ambient cosmic radiation, the CE industry will not pay extra for hardening their chips.
- ◆ Architectural mitigation strategies have to be developed such that replicated mass-market chips can be used to build high-integrity embedded systems.

© H. Kopetz 3/30/2006

Mitigation of Soft Errors by Architectural Means

16

Architectural means to mitigate the consequences of component failures might become a necessity when using the upcoming submicron devices, as stipulated in the latest *2005 International Roadmap of Semiconductors* p.6:

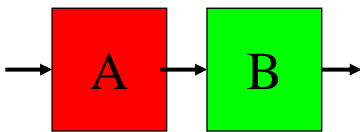
*Relaxing the requirement of 100% correctness for devices and interconnects may dramatically reduce the costs of manufacturing, verification and test. Such a paradigm shift is likely **forced in any case by technology scaling**, which leads to more transient and permanent failures of signals, logic values, devices and interconnects.*

© H. Kopetz 3/30/2006

Mitigation at the Architecture Level: TMR

17

Triple Modular Redundancy (TMR) is the generally accepted technique for the mitigation of component failures at the system level:

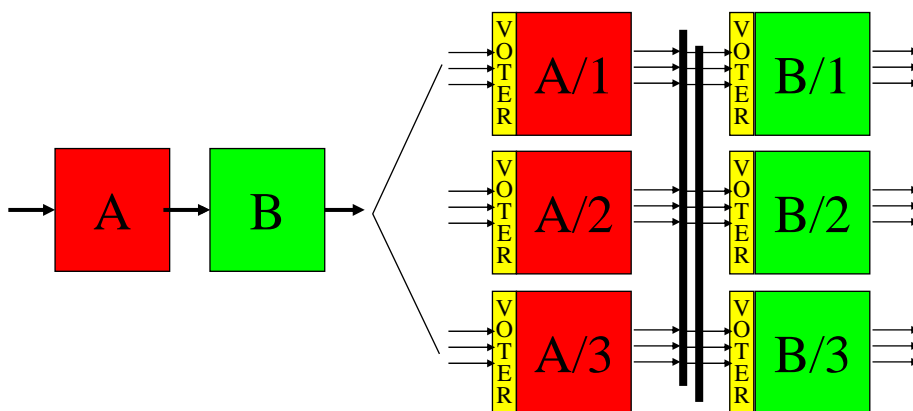


© H. Kopetz 3/30/2006

Mitigation at the Architecture Level: TMR

18

Triple Modular Redundancy (TMR) is the generally accepted technique for the mitigation of component failures at the system level:



© H. Kopetz 3/30/2006

What is Needed to Implement TMR?

19

What architectural services are needed to implement Triple Modular Redundancy (TMR) at the architecture level?

- ◆ Provision of an Independent Fault-Containment Region for each one of the replicas
- ◆ Synchronization Infrastructure
- ◆ Multicast communication
- ◆ Replicated Communication Channels
- ◆ Support for Voting
- ◆ Timely and Deterministic Operation

© H. Kopetz 3/30/2006

Independence of FCRs

20

The independence of failures of different FCRs is the most critical issue in the design of an ultra-dependable system.

There are three mechanisms that compromise the independence of FCRs

- ◆ Missing fault isolation among the FCRs
- ◆ Error propagation--the consequences of a fault, the *ensuing error*, propagates to a healthy FCR by an erroneous message.
- ◆ Consensus protocols among the FCRs that are needed if the architecture is *non-deterministic*.

© H. Kopetz 3/30/2006

Example: Determinism of a Communication Channel²¹

A communication channel is called *deterministic* if (as seen from an omniscient external observer):

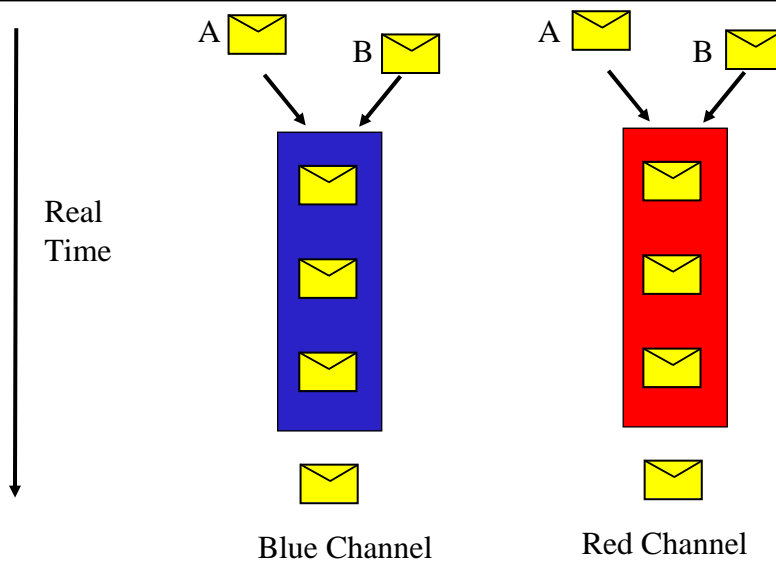
- ◆ The *receive order* of the messages is the same as the *send order*. The *send order* among all messages is established by the *temporal order* of the *send instants* of the messages as observed by an omniscient observer.
- ◆ If the *send instants* of n ($n > 1$) messages are the *same*, then an order of the n messages will be established in an *a priori* known manner.

Two correctly operating *independent* deterministic communication channels will deliver messages *always* in the same order.

© H. Kopetz 3/30/2006

Determinism--Temporal Order is Obvious

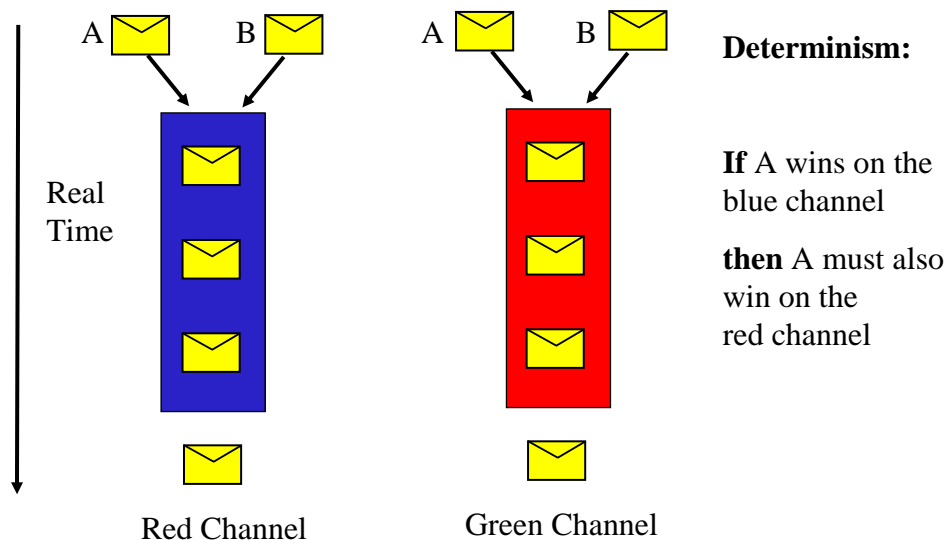
22



© H. Kopetz 3/30/2006

Determinism: Simultaneity--Who Wins?

23



© H. Kopetz 3/30/2006

Simultaneity: A Fundamental Problem

24

The ordering of simultaneous events is a fundamental problem of computer science:

- Hardware level: metastability
- Node level: semaphore operation
- Distributed system: ordering of messages

There are two solutions *within* a distributed system to solve the simultaneity problem:

- Distributed consensus--takes real-time and requires bandwidth (atomic broadcast)
- Sparse time

© H. Kopetz 3/30/2006

Determinism Requires Consistent Order

25

Accesses to common resources must be temporally ordered in order to avoid *race conditions*. Any race-condition is *in conflict* with determinism:

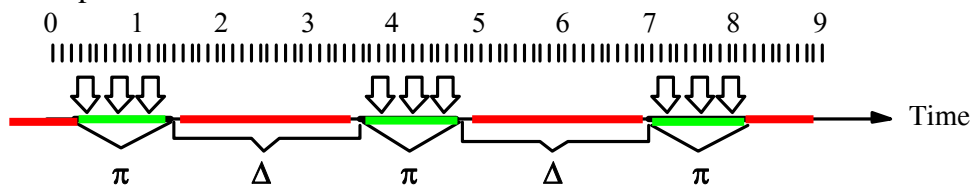
- ◆ **Establishing temporal order is much more expensive than maintaining temporal order.**
- ◆ Once a *proper global time* has been established, it can be used to order events consistently in the temporal domain.
- ◆ In a system with global time, the difficult ordering problem has to be solved only once, at system startup, to achieve the initial synchronization of the clocks. Maintaining the clock synchronization is relatively easy.
- ◆ In a system without global time, the difficult problem has to be solved every time temporal order is required (atomic broadcast).


© H. Kopetz 3/30/2006

Fault Tolerant Sparse Time Base in the TTA

26

If the occurrence of events is restricted to some active intervals with duration π with an interval of silence of duration Δ between any two active intervals, then we call the timebase π/Δ -sparse, or sparse for short.



Events  are only allowed to occur at subintervals of the timeline

In a sparse time base, instants can be represented by integers.

© H. Kopetz 3/30/2006

The Goal of the EU DECOS Integrated Project

27

It is the goal of DECOS (Dependable Components and Systems) to facilitate the systematic design of large dependable embedded systems out of components. The interactions of the components are realized by the exchange of time-triggered and event-triggered messages across interfaces to a real-time communication system.

The driving forces for an integrated architecture are:

- ◆ System services must be more reliable than component services
- ◆ Cognitive complexity reduction in order to reduce the design and development effort
- ◆ Reuse of components: The components may be newly designed according to a given *architectural style* or may be already existing systems (legacy systems).
- ◆ Simplified diagnostics and repair.

© H. Kopetz 3/30/2006

DECOS Data

28

Partners (19):

- ◆ *Res.Centers*: ARCS (project manager), SP
- ◆ *Industry*:
Audi, Fiat, Hella
Airbus, EADS, Thales, Liebherr Aerospace
Infineon, TTTech,
Profactor, Esterel
- ◆ *Universities*:
TU Vienna, TU Darmstadt, TU Hamburg-Harburg,
Budapest University,
Univ. Kassel, Univ. Kiel

Overall Budget: 15 M€

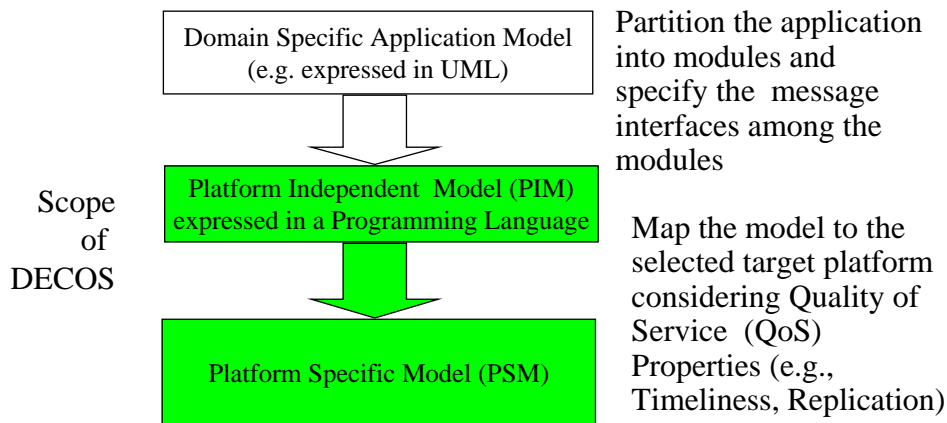
Funding: ca. 9 M€

Duration: 3 years, from July 2004-2007

© H. Kopetz 3/30/2006

DECOS supports a *Model-Driven Design Style*

29



© H. Kopetz 3/30/2006

DECOS is based on the TTA

30

The Time-triggered Architecture (TTA) provides an execution environment for real-time applications. It is

- ◆ a *distributed architecture* that provides a ***fault-tolerant sparse global time-base*** of high precision at every node.
- ◆ a ***deterministic architecture*** that support *fault tolerance by replication*, where a node can be a single-chip computer (SoC).
- ◆ a ***generic architecture***, which can be deployed in different application domains (e.g., automotive, aerospace, train signaling, process control, multimedia).
- ◆ an ***integrated architecture***, where different application subsystems (DAS) up to the *highest criticality class* can be integrated into a single framework.

Kopetz, H, Bauer, G., The Time-Triggered Architecture, Proc. of the IEEE, Jan 2003, Vol 91 p. 112-126

© H. Kopetz 3/30/2006

Distributed Application Subsystem (DAS)

31

The services of a large RT control system (e.g., the computer system onboard a car or an airplane) can be partitioned into a set of nearly autonomous subsystems, we call them *Distributed Application Subsystems (DAS)*.

Examples of a DAS onboard a car are

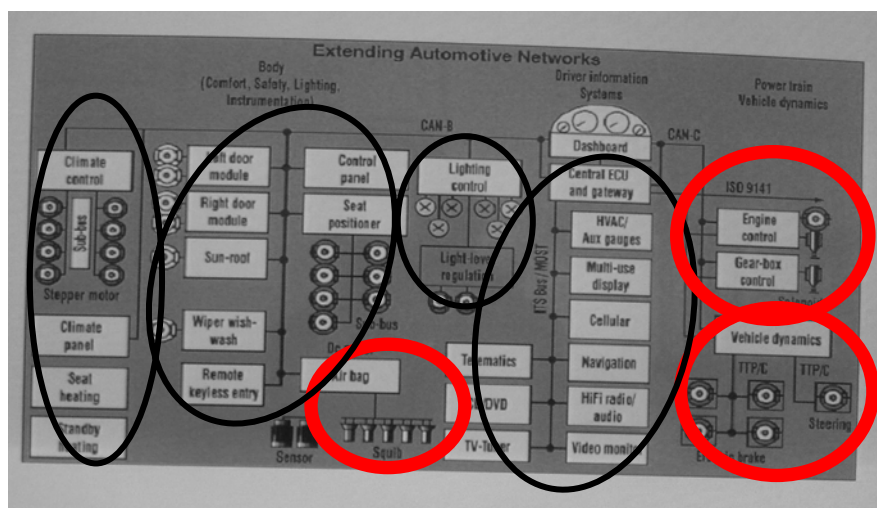
- ◆ The body electronics DAS (doors, lights, clima control etc.)
- ◆ The power train control DAS
- ◆ The multi-media DAS

Systems of systems do not have a single top.

© H. Kopetz 3/30/2006

Example of DASes onboard a Car

32



DAS-Distributed Application Subsystem

© H. Kopetz 3/30/2006

Integrated Architecture

33

A number of technical and economic advantages could be realized if the different DASes were integrated into a single architecture

- ◆ Cost savings by the reduction of the number of ECUs, sensors and wiring points (results also in an increase in hardware reliability).
- ◆ Better integration of functions--more flexibility
- ◆ Implementation of fault tolerance simplified

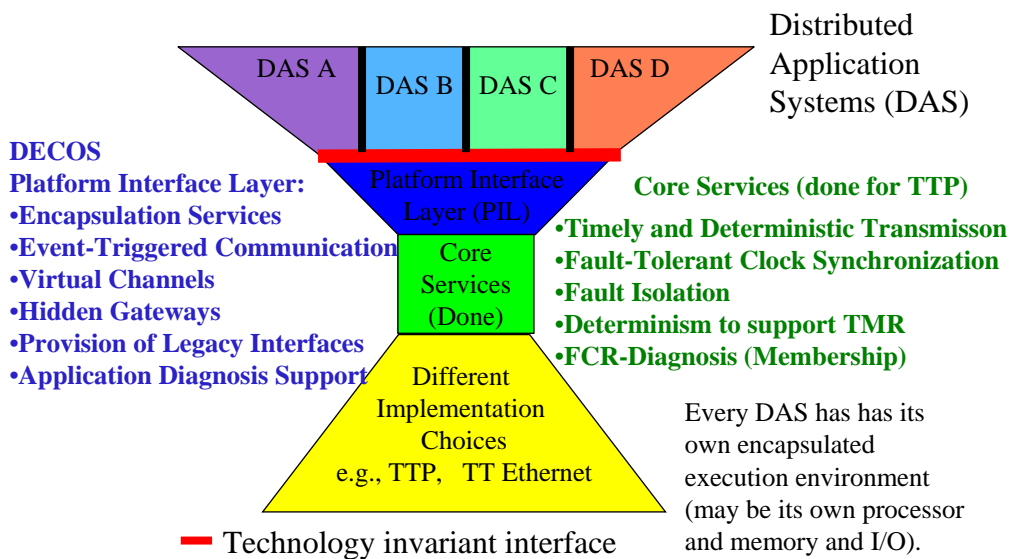
But

- ◆ Independence of individual DAS compromised--increased potential of error propagation from one DAS to another DAS
- ◆ Integration increases complexity and diagnostics
- ◆ Allocation of responsibility more difficult.

© H. Kopetz 3/30/2006

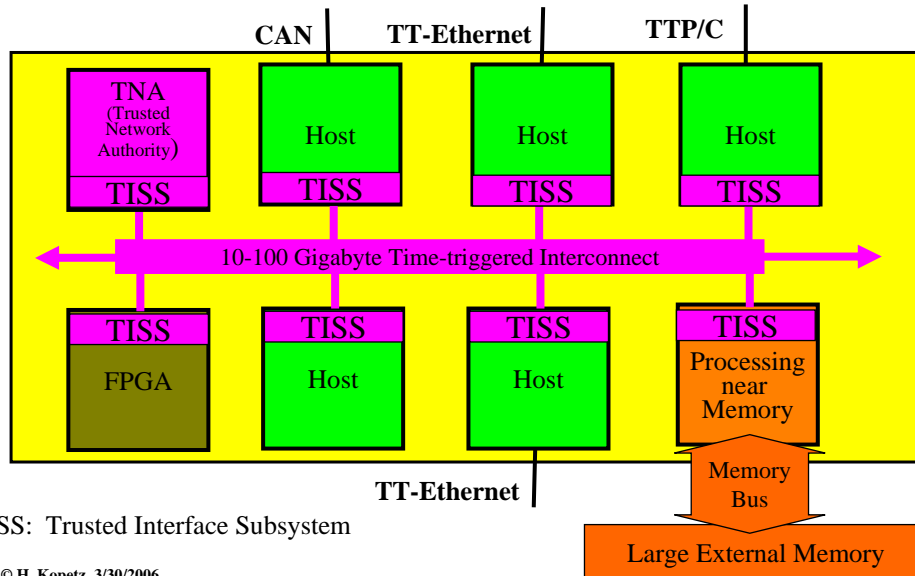
The TTA is an *Integrated Platform Architecture*

34



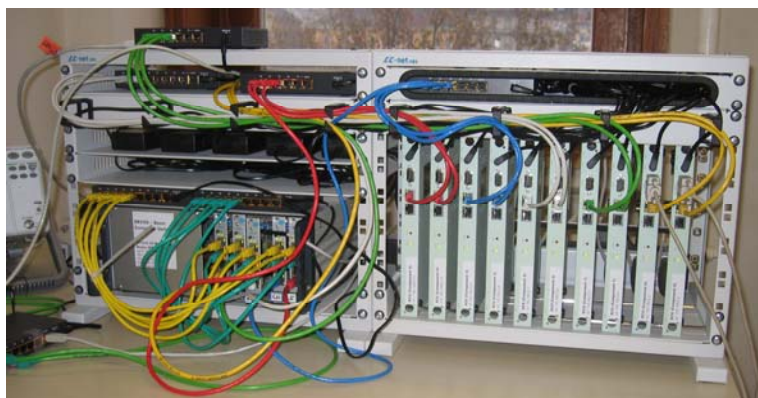
TTA Vision: Node on a single Chip (SoC)

35



Prototype of a DECOS TTA Node at TU Vienna

36

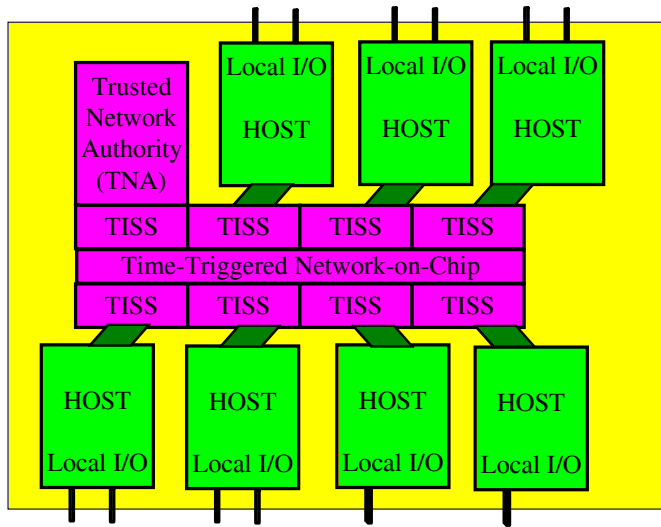


A set of application computers (*host*) and a communication network (TTP or TT Ethernet) are forming a single node.

© H. Kopetz 3/30/2006

Next Step: Integrate the Hosts by an NoC FPGA

37



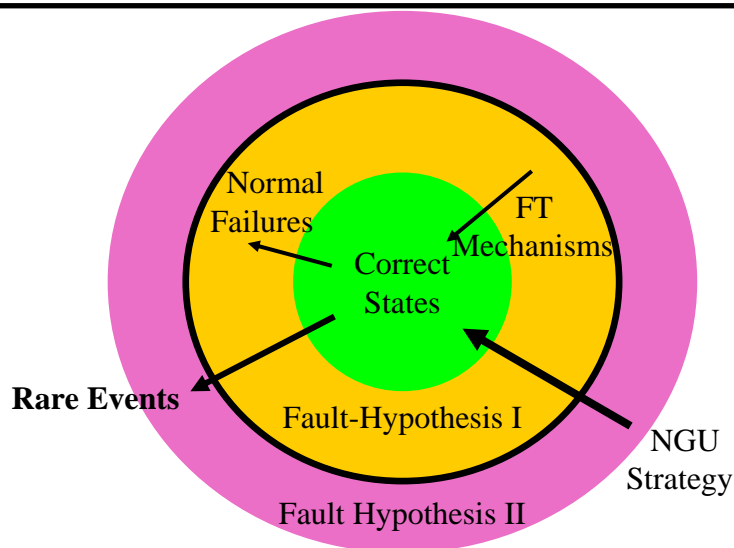
Single PCB Board with heterogeneous COTS Micro-computers connected by a Network-on-Chip (NoC) in a single FPGA.

Standard Interface between Host and TISS (Trusted Interface Subsystem)

© H. Kopetz 3/30/2006

Fault Hypothesis in the TTA

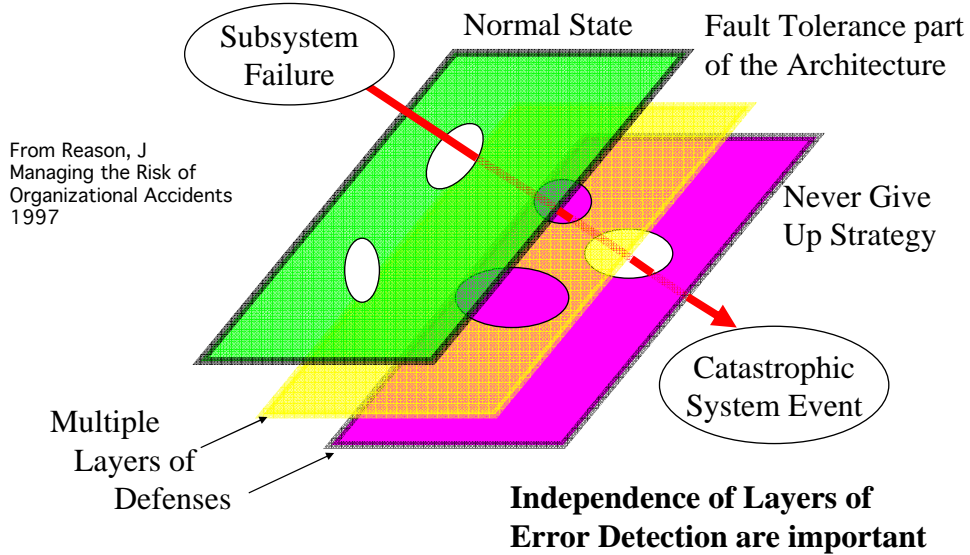
38



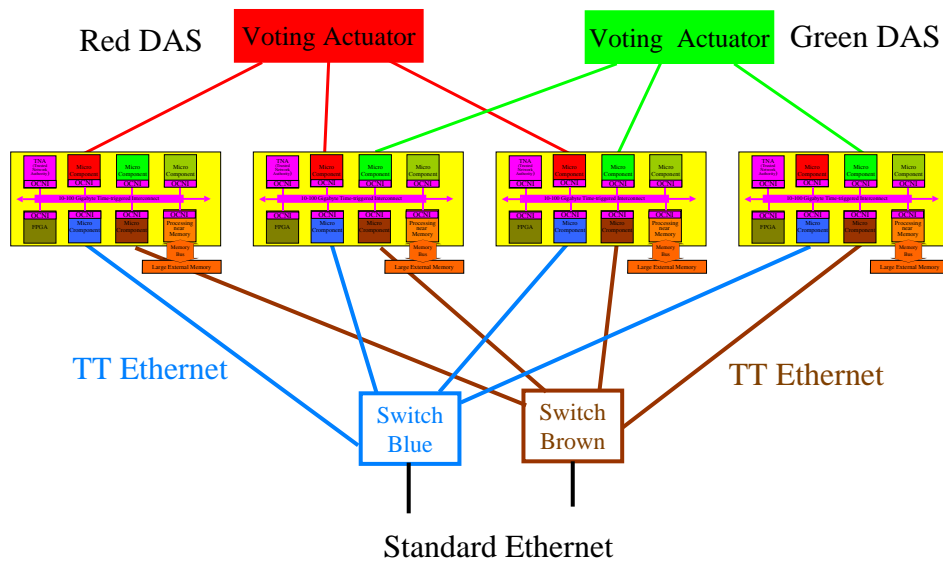
© H. Kopetz 3/30/2006

Approach to Safety: The *Swiss-Cheese Model*

39

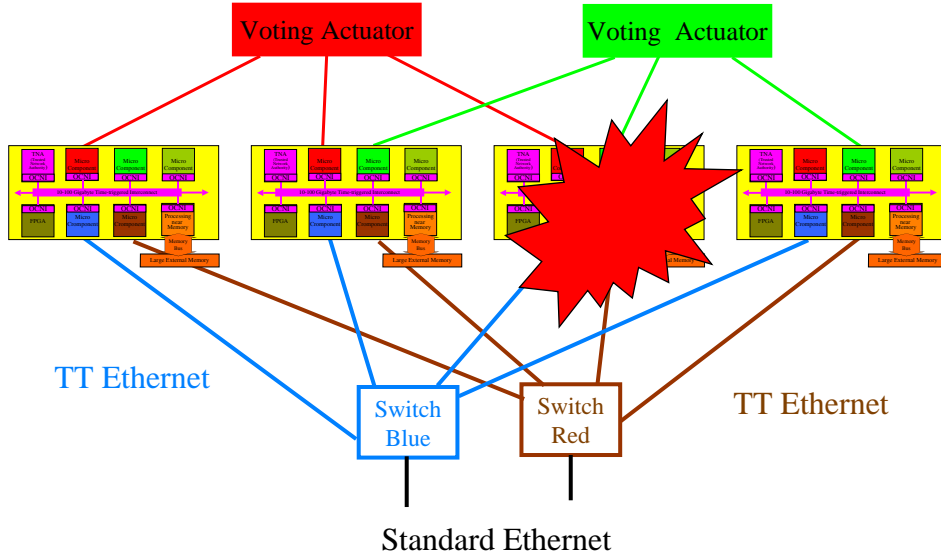


Example: TMR Configuration with the DECOS SoC⁴⁰



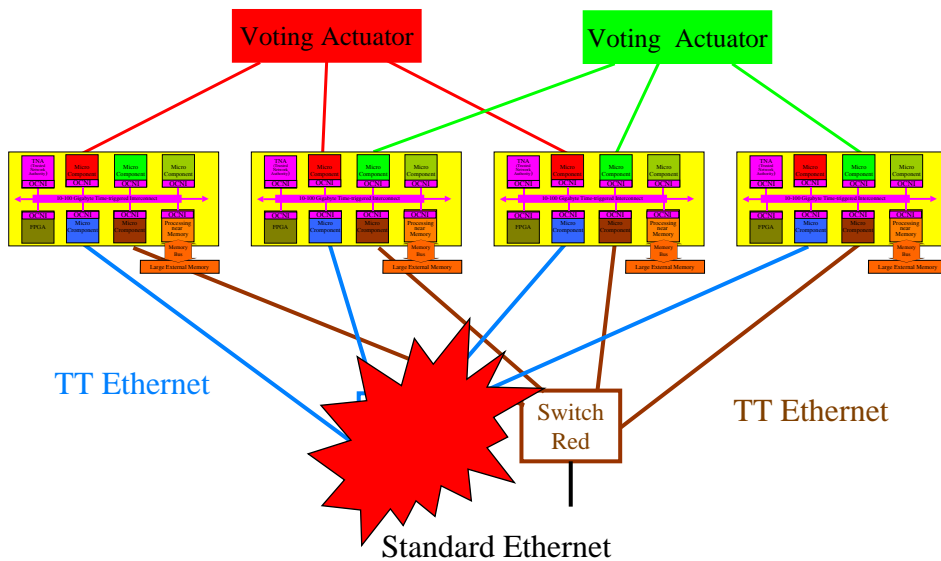
Example: TMR Configuration

41



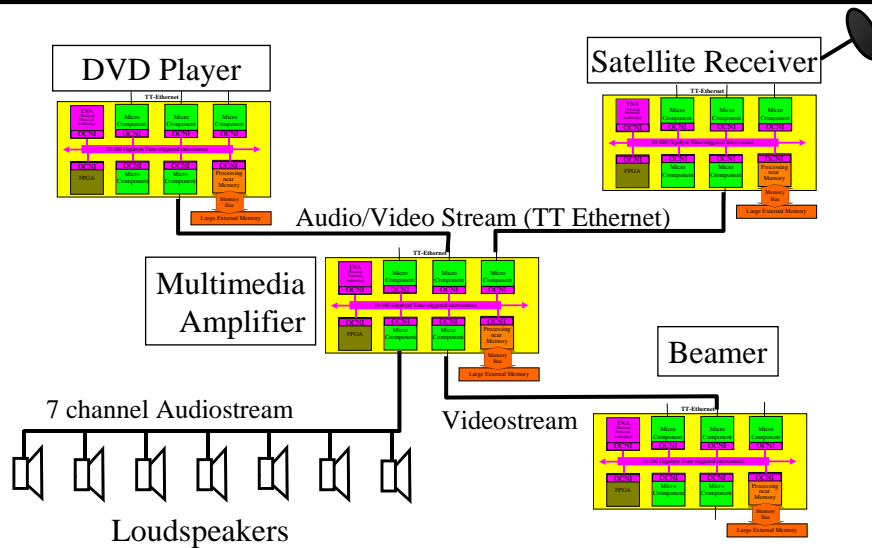
Example: TMR Configuration

42



Example: Streaming Multimedia System

43



© H. Kopetz 3/30/2006

Conclusion

44

- ◆ We are in a period of dramatic change--such a period offers many opportunities, but many of the present views and approaches have to be reconsidered.
- ◆ It is a great challenge to build *robust systems* out of imperfect components.
- ◆ Determinism at the level of the architecture is required if component faults are mitigated by Triple-Modular Redundancy (TMR).
- ◆ The DECOS project, based on the Time-Triggered Architecture, is intended to provide a framework for the implementation of robust embedded systems.

© H. Kopetz 3/30/2006