

Panel: Challenges & Industrial Practice to Improve System Reliability

Moderator

Xinli Gu, Huawei Technologies, USA

Panelists

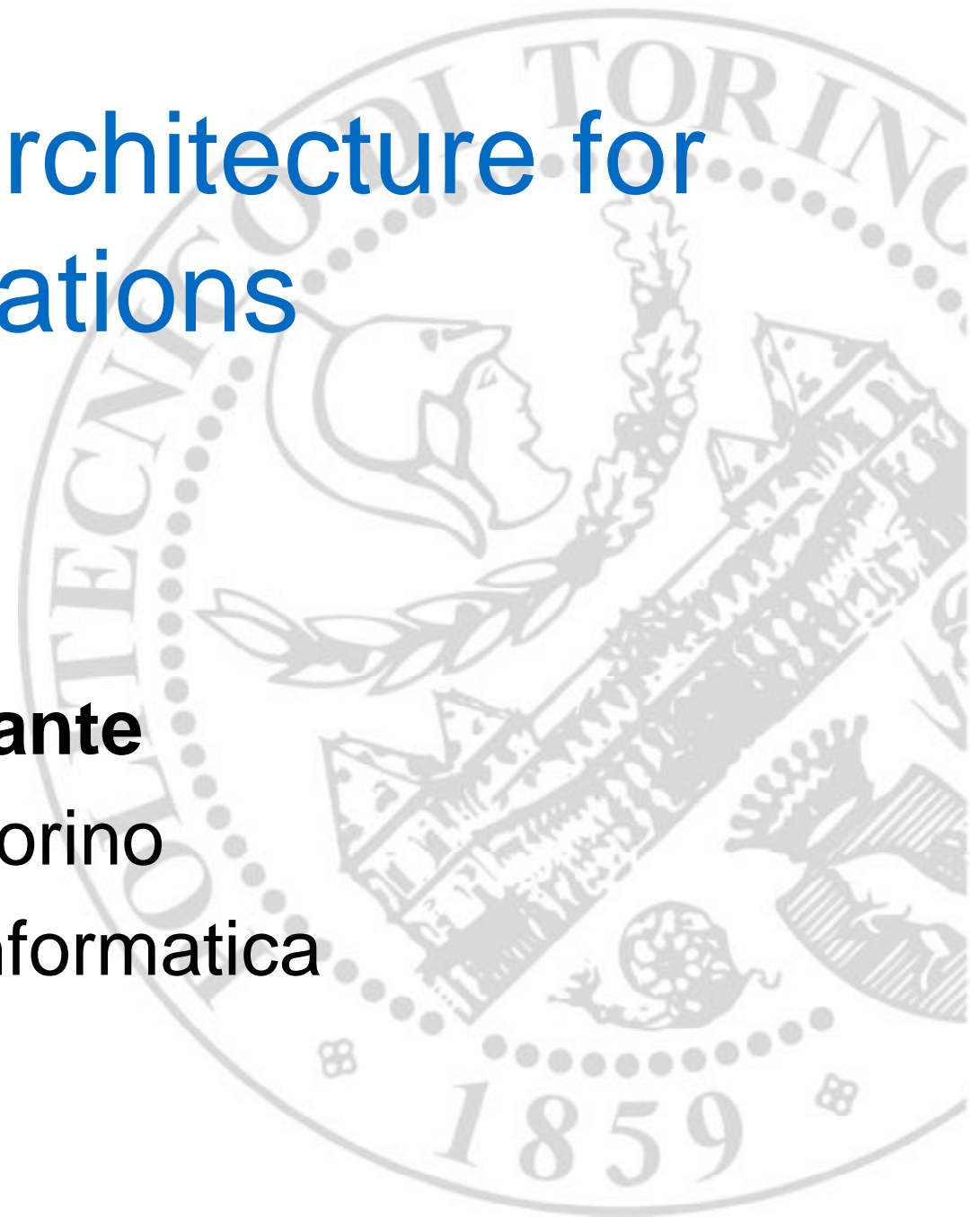
- Jun Qian, AMD, Shanghai
- Bill Eklow, ex- Cisco Systems, Inc., USA
- Jaan Raik, Tallinn Univ. of Technology, Estonia
- Massimo Violante, Politecnico di Torino, Italy

Resilient multi-core architecture for critical applications

Massimo Violante

Politecnico di Torino

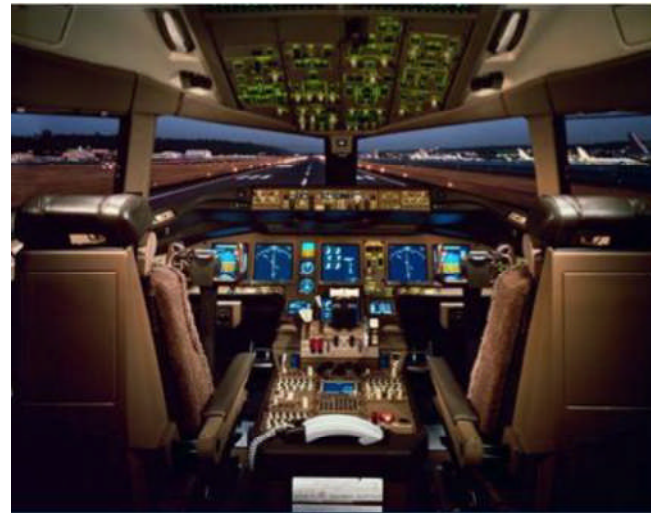
Dip. Automatica e Informatica



Massive amount of embedded software in many applications



Space Shuttle
~500.000 LOCs

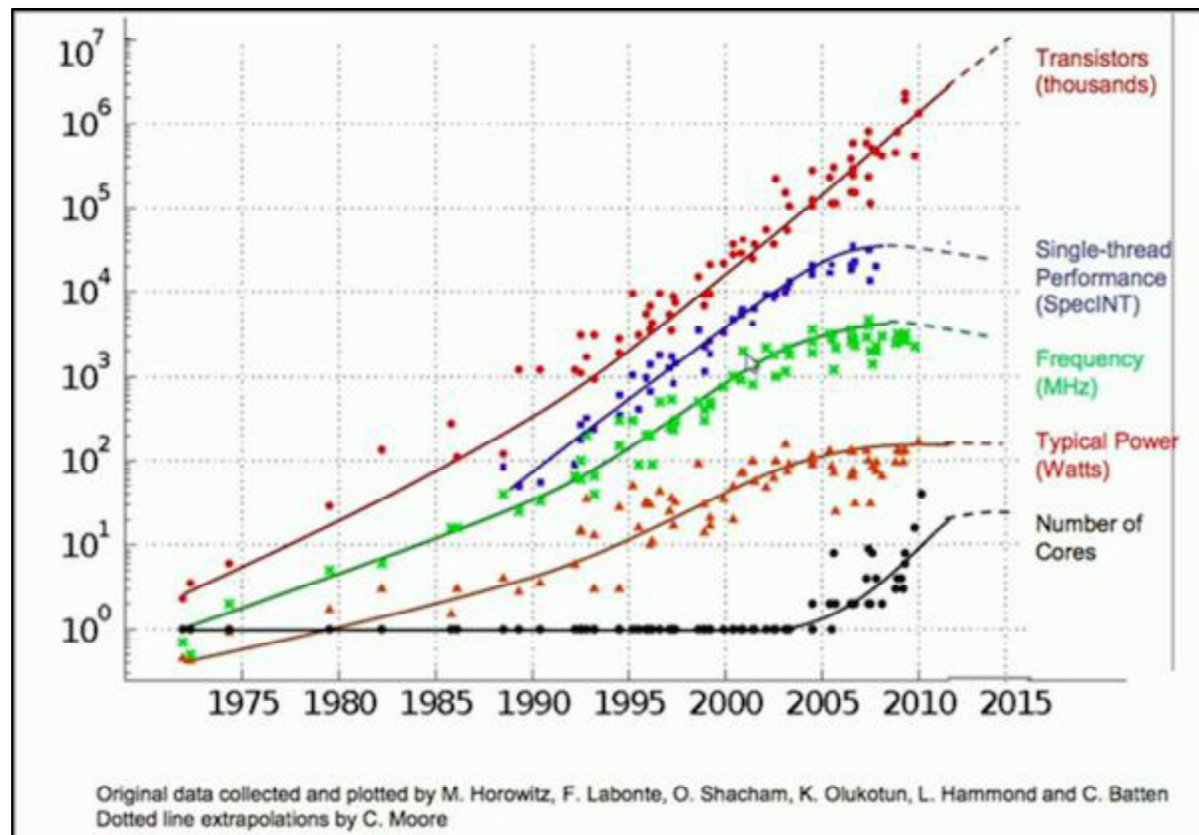


Boeing 777
~3 Millions LOCs

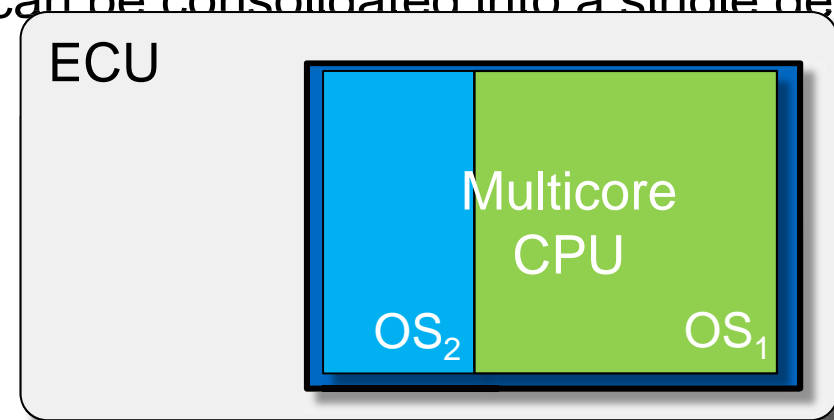


High-end vehicle
~15 Millions LOCs

Multi-/many-core scenario – the possibilities

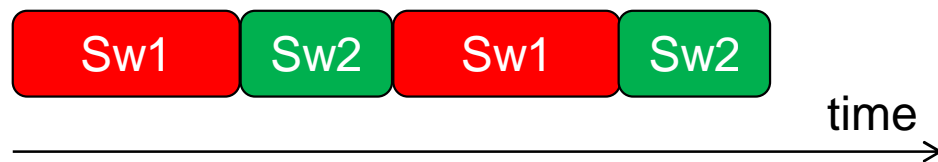


- Multi-/many-core architectures offer huge amount of computing capabilities
 - Distributed sw can be integrated into a single device
 - Multiple-independent computing devices can be consolidated into a single device

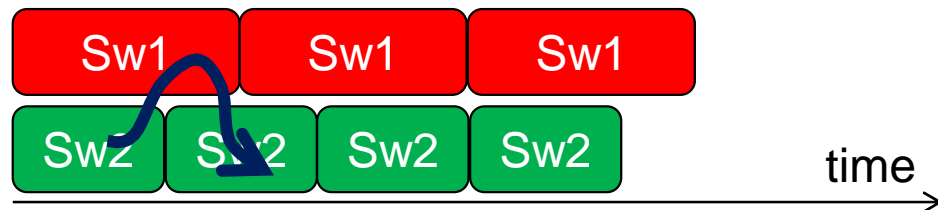


Multi-/many-core scenario – the challenges

- Functional interference

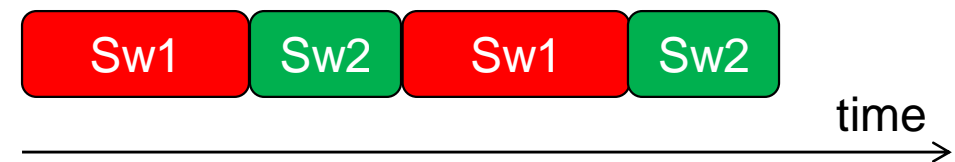


The sw running on one core alters the memory allocated to the sw running on the other core

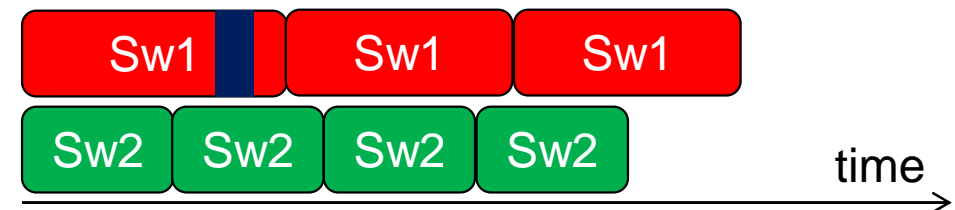


The functional interference can be controlled using virtualization solutions, e.g., hypervisors

- Temporal interference



The sw running on one core alters the execution time of the sw running on the other core



The time interference requires novel hw features to guarantee QoS

Conclusions

- Multi-/many-core architecture will dominate high-performance embedded system landscape
- Novel solutions are needed to enable their usage in mission-/safety-critical applications to cope with functional/temporal interference
- No silver bullet available today!





1918



TALLINNA TEHNIKAÜLIKOO

TALLINN UNIVERSITY OF TECHNOLOGY

Challenges and Common Industrial Practice to Improve Systems Reliability

Jaan Raik



Personal Background

- 20+ years in HW test
- 10+ years in verification
- Last years: coordinating/participating in EU collaborative research projects
 - FP7 DIAMOND (2010-12)
 - FP7 BASTION (2014-16)
 - H2020 IMMORTAL (2015-18)
 - Partners: IBM, Ericsson, Infineon, DLR, Recore, Testonica, ...
 - Resiliency in many-core systems



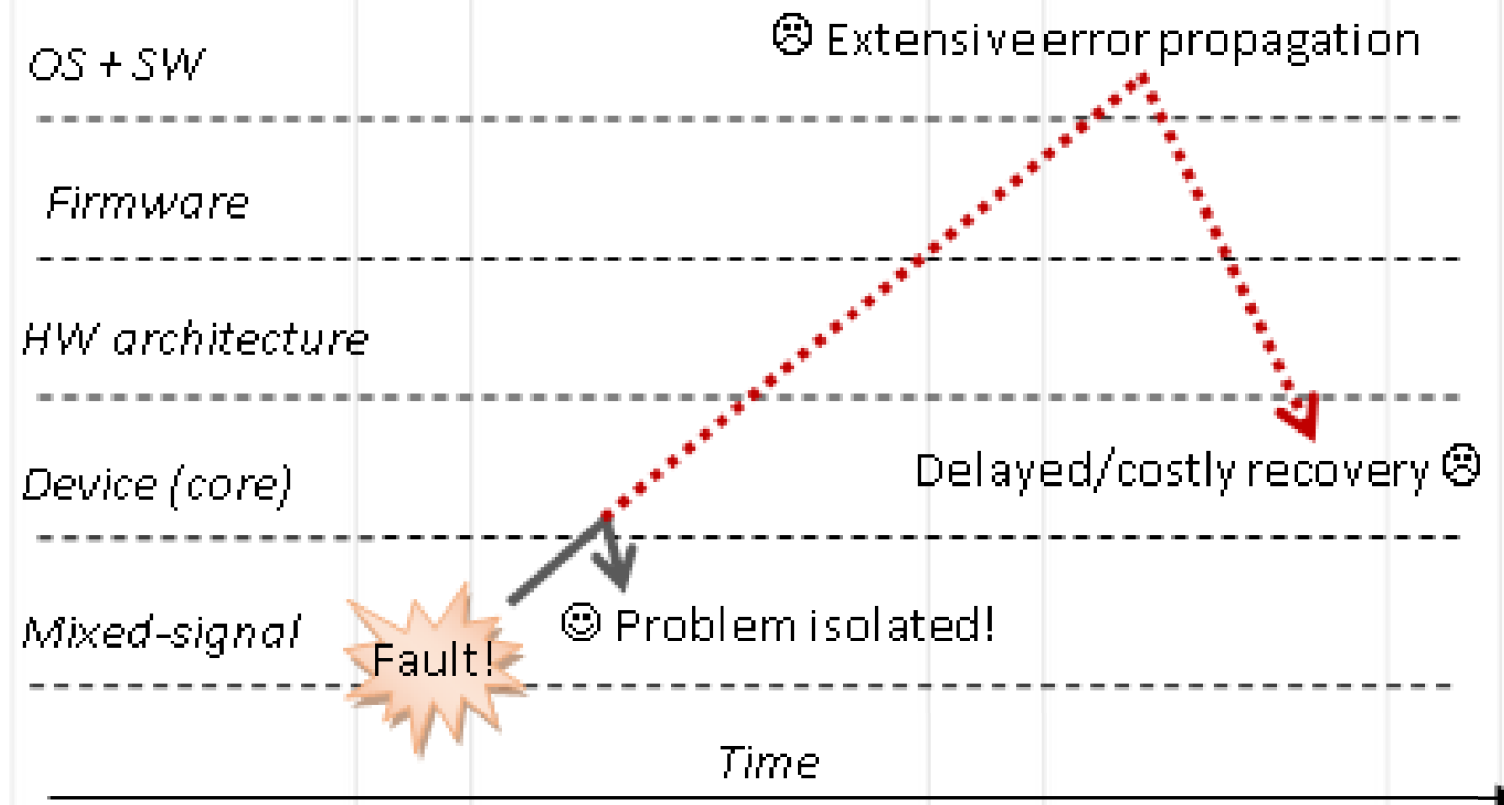
Why System Resilience?

- Resilience vs “Good Old” Tolerance
- A trade-off (depending on the application!)
- Fault tolerance:
 - Simple to implement/no downtime
 - Redundancy/hardware overhead
 - No information about system health status
- Resilience:
 - Complex/time penalty involved
 - Potentially less redundancy(?)
 - System health status/management ->
relevant concerning degradation issues



Improving System Reliability Challenges/Industry Practices

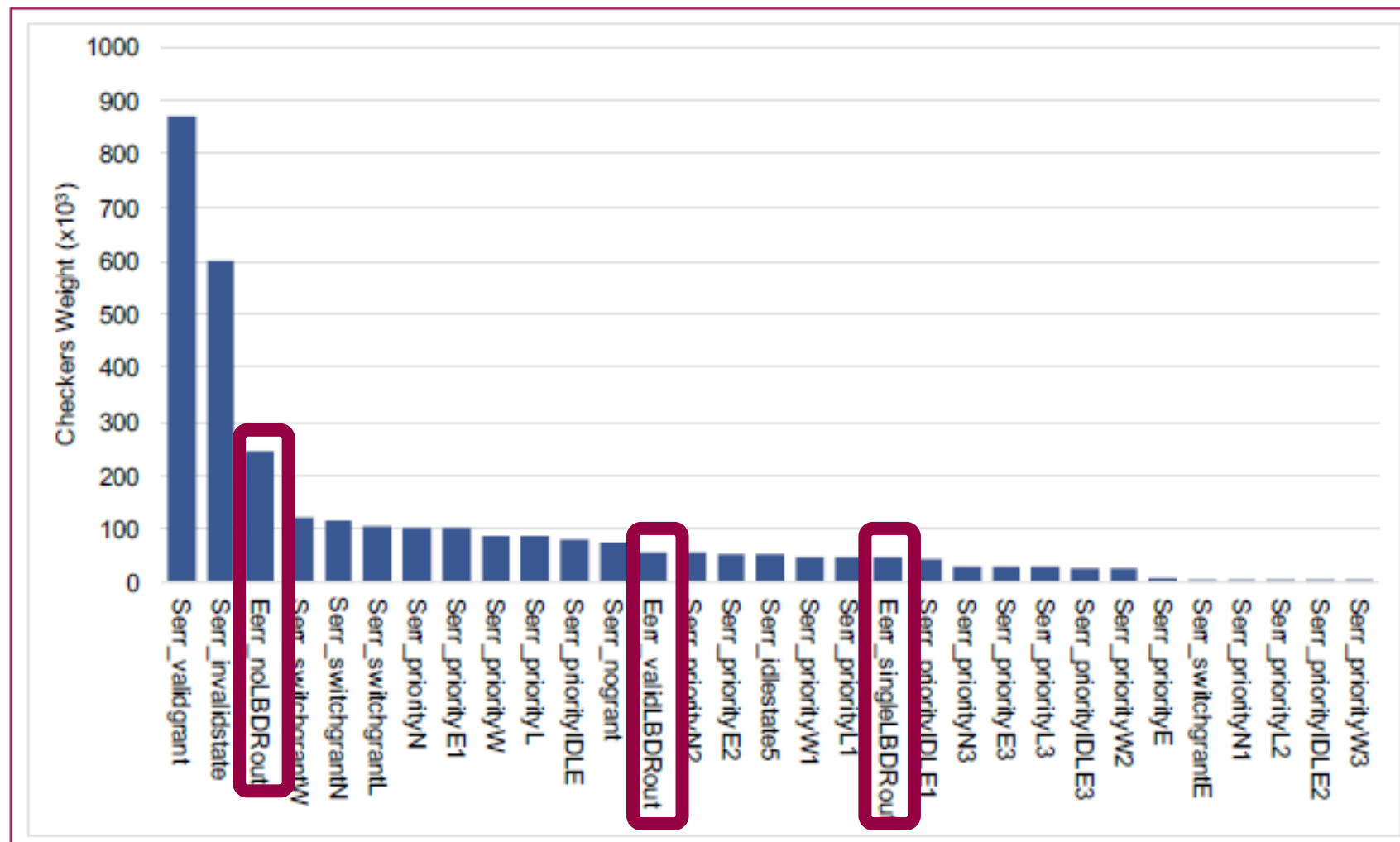
- Achieving system resilience a challenge
- Not about the dots... but the right combination of dots and about connecting them
- A cross layer approach needed
- Latency of FDIR is crucial!





Fast detection

- NoC router control part (routing + arbitration)
- 5 checkers, F.C. 100%, area overhead ~60%
- However, checking flow control (hand-shaking) costly **(temporal checkers, more area or time needed!)**





Research on Systems' Resilience

- Fast fault detection (partner: IBM)
- Fault management infrastructure, health map (Testonica, Ericsson)
- Resilience in many-core systems (SW runtime): (Recore)
- FDIR at system level (DLR)
- See also www.h2020-immortal.eu

IMMORTAL
CYBER-PHYSICAL SYSTEMS



Thank You!

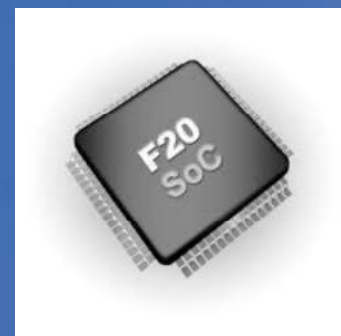
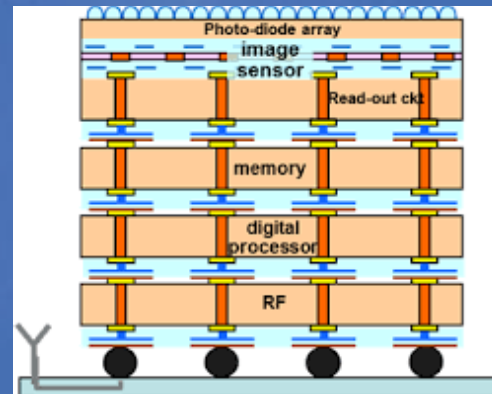
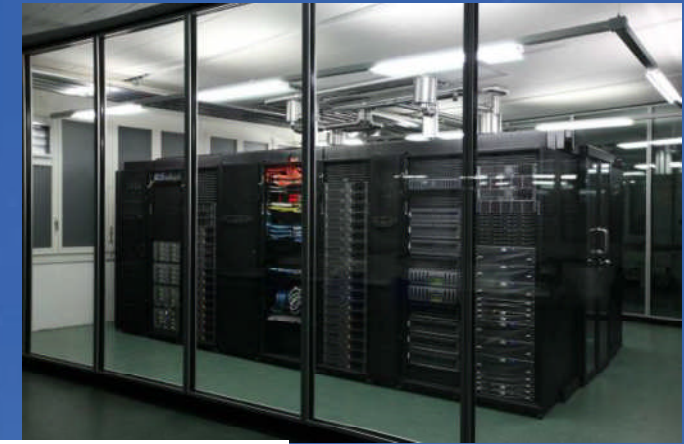


SYSTEM-LEVEL, RANKED
RELIABILITY BASED ON
COST/CONSEQUENCE METRIC

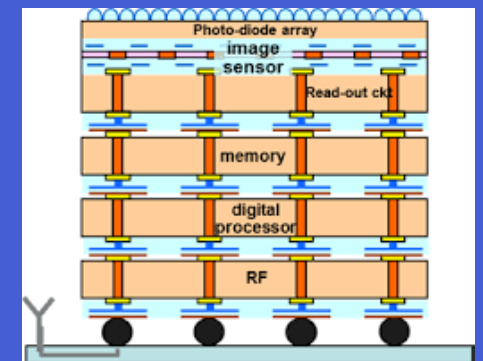
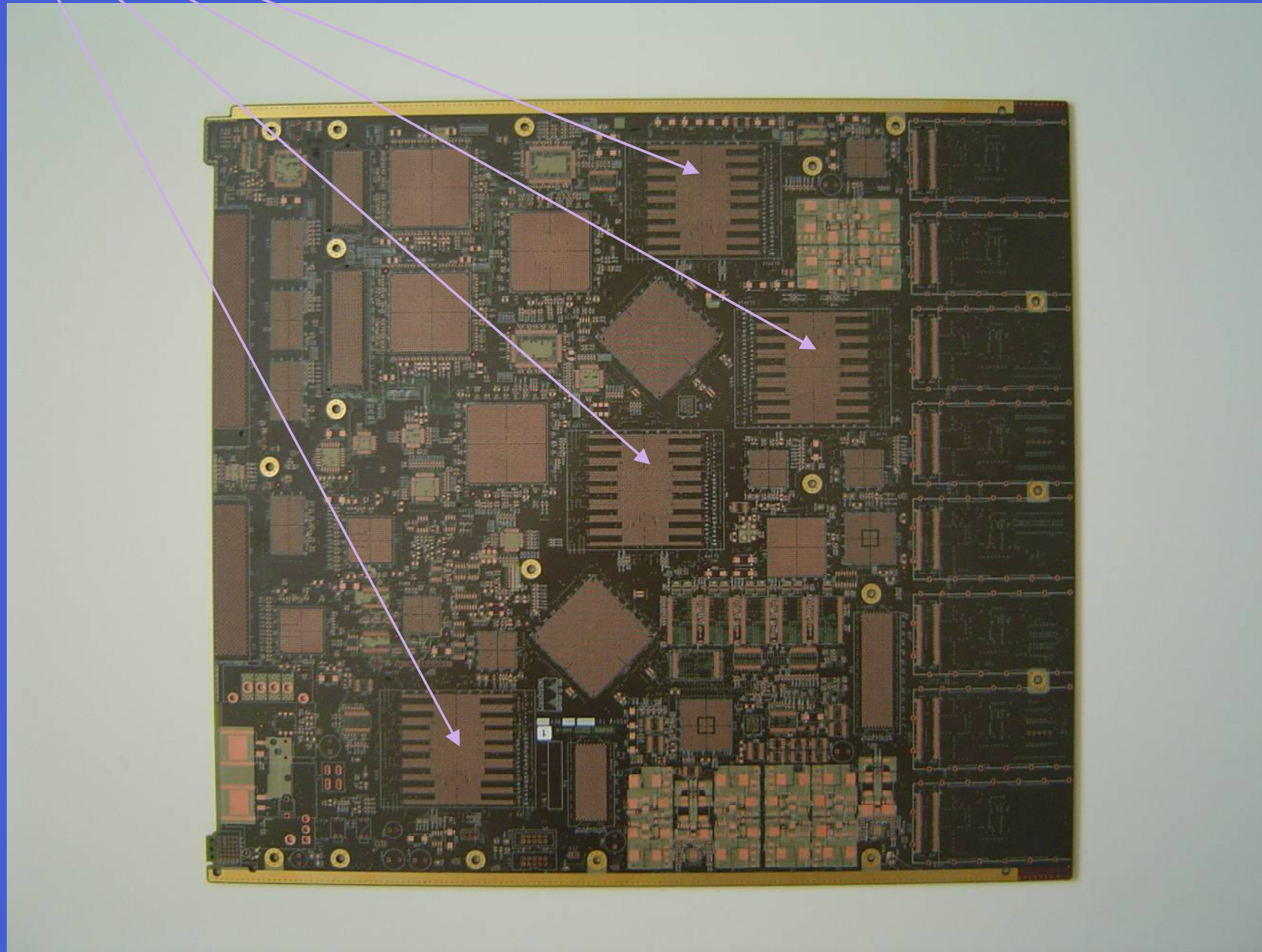
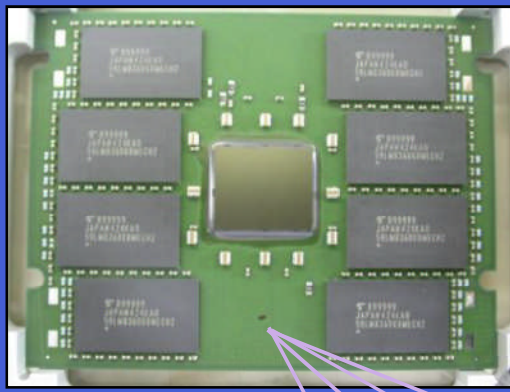
Bill Eklow

Retired from Cisco

WHAT COMPRISES A SYSTEM?



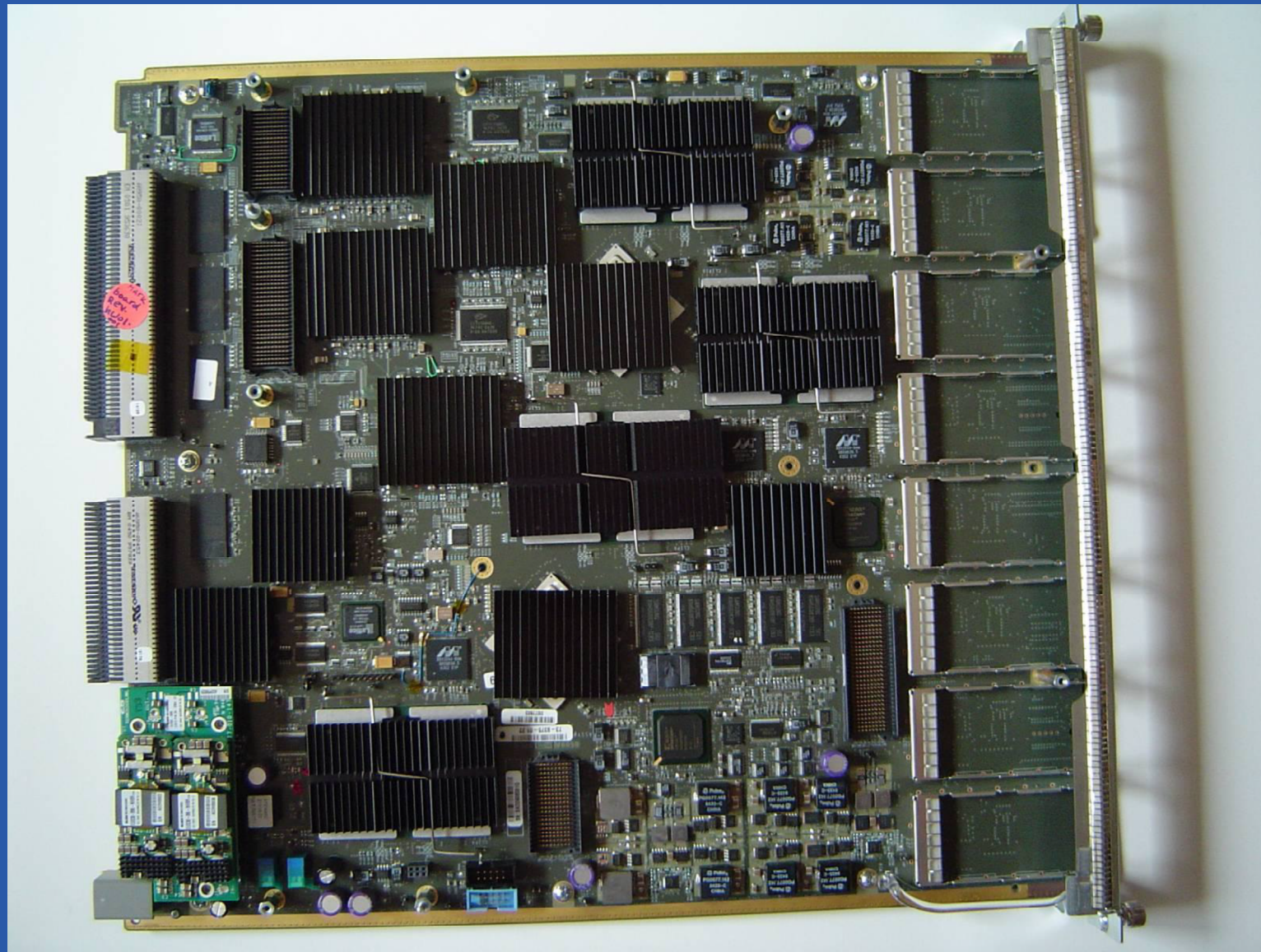
FROM COMPONENT COMPLEXITY TO SYSTEM COMPLEXITY



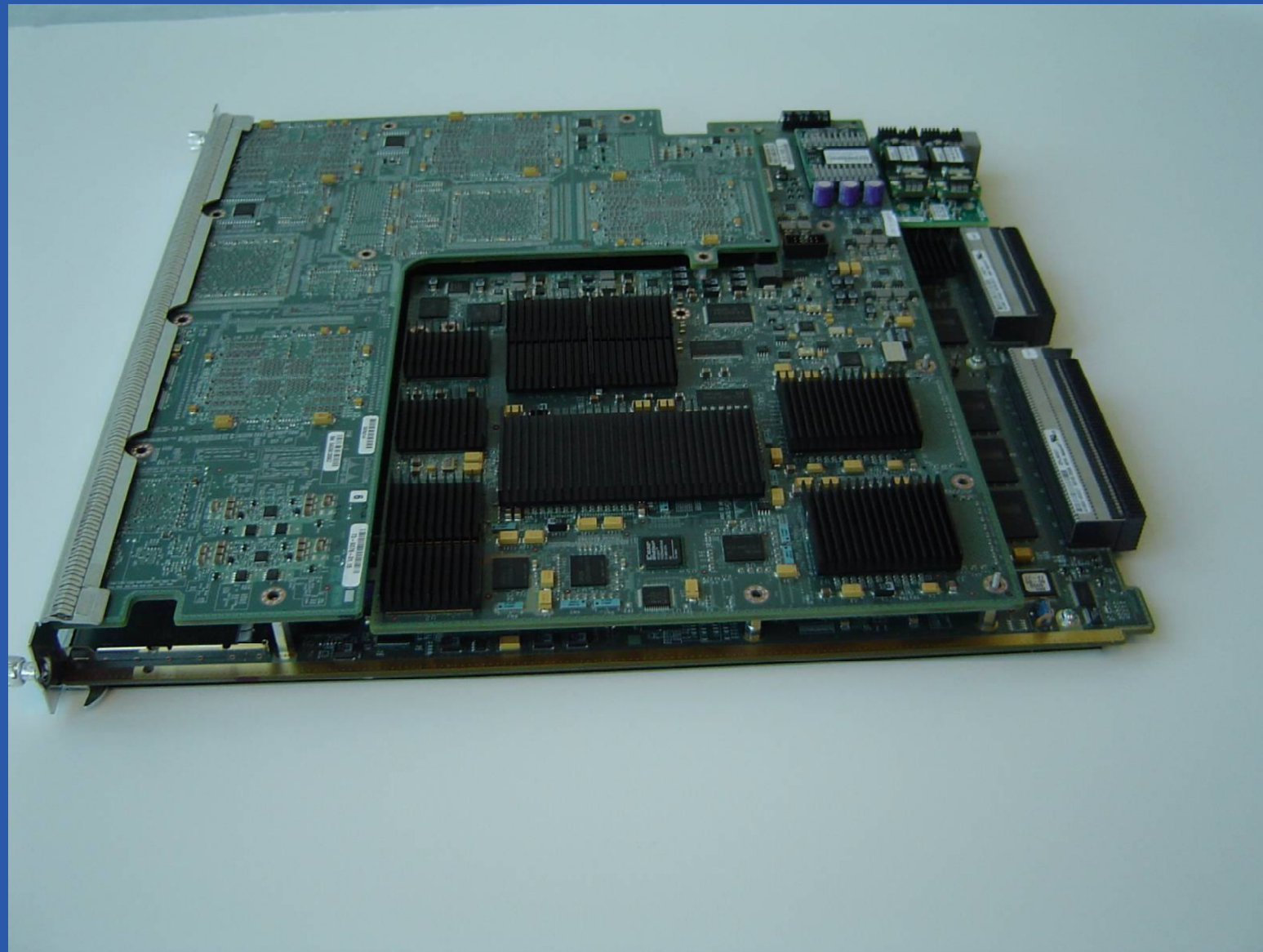
THE COMPONENT COMPLEXITY CHALLENGE

- Functional Specs over 1000 pages
- Millions -> Billions of flops
- Giga to Terabytes of memory
- Extremely high speeds
- Margins are very tight
- Zero access to any signals
- Very long test times
- Add 3D-SIC and Heterogeneous Integration

SYSTEM ON SYSTEM HIERARCHY



SYSTEM ON SYSTEM HIERARCHY



SYSTEM ON SYSTEM HIERARCHY

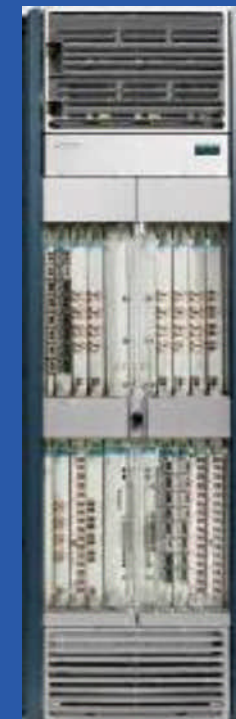


Integrating Switches into the Network



Diagnosing through the System Hierarchy

Diagnose and Isolate – Extremely Difficult but Critical



Localize Misbehavior

Detection and Mass Replacement no clue here

SYSTEM/APPLICATION LEVEL TEST

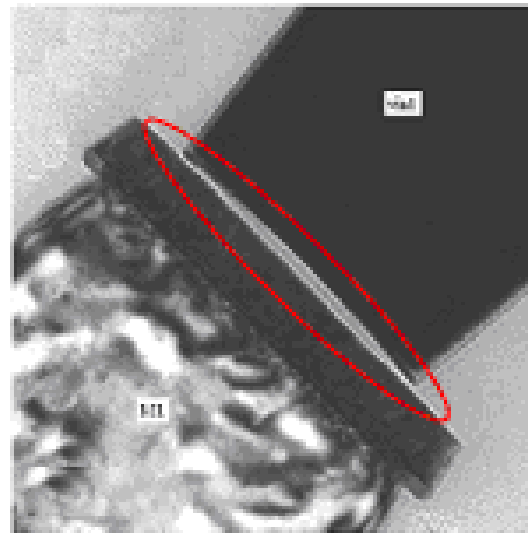


SYSTEM TESTING METHODOLOGY

- Functional Test
 - Function based – based on functional spec
 - Usually driven by on board processor
 - Onion skin approach – simple logic first
 - Tests sequenced to aid diagnosis
 - Some level of “manual” debug required (painful)
- Application Test
 - Customer based/like applications (Network traffic)
 - Use operating system software
 - Very little visibility into what’s going on
 - Some monitors at system level

POSSIBLE CAUSES OF LATENT SYSTEM FAILURES

(a) Resistive via



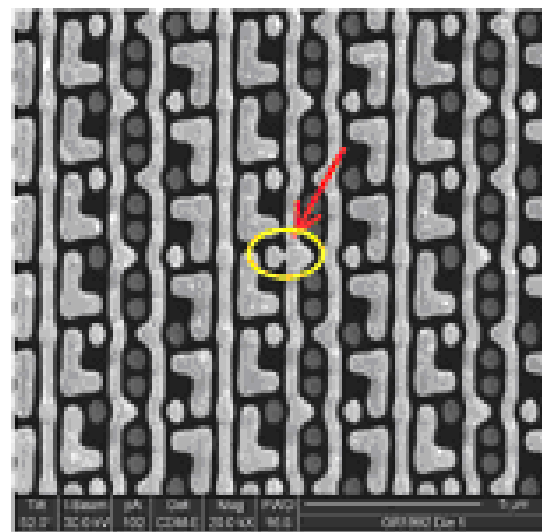
Source : [Sachdev 2007]

(b) Resistive open



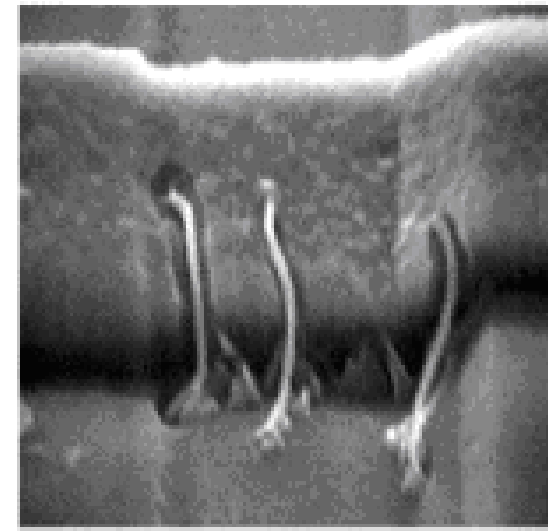
Source : IBM

(c) Resistive bridge

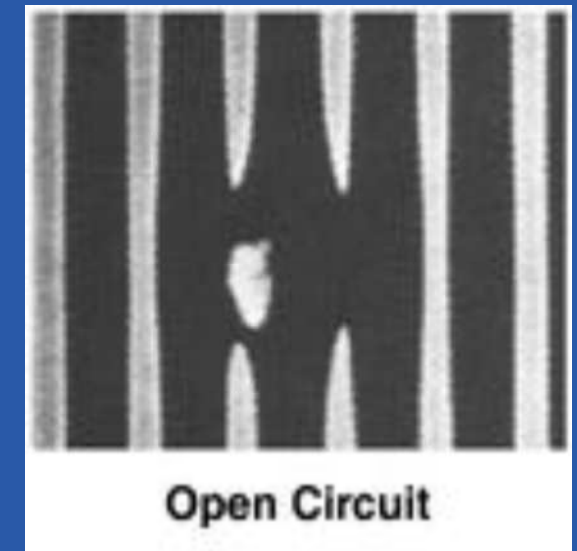


Source : LSI Logic

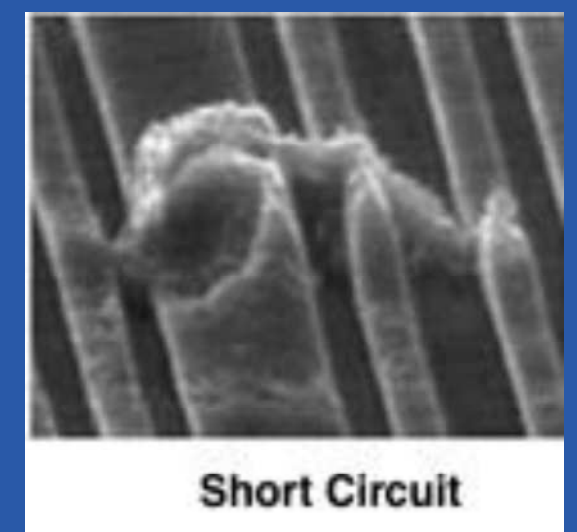
(d) Gate oxide short



Source : LSI Logic

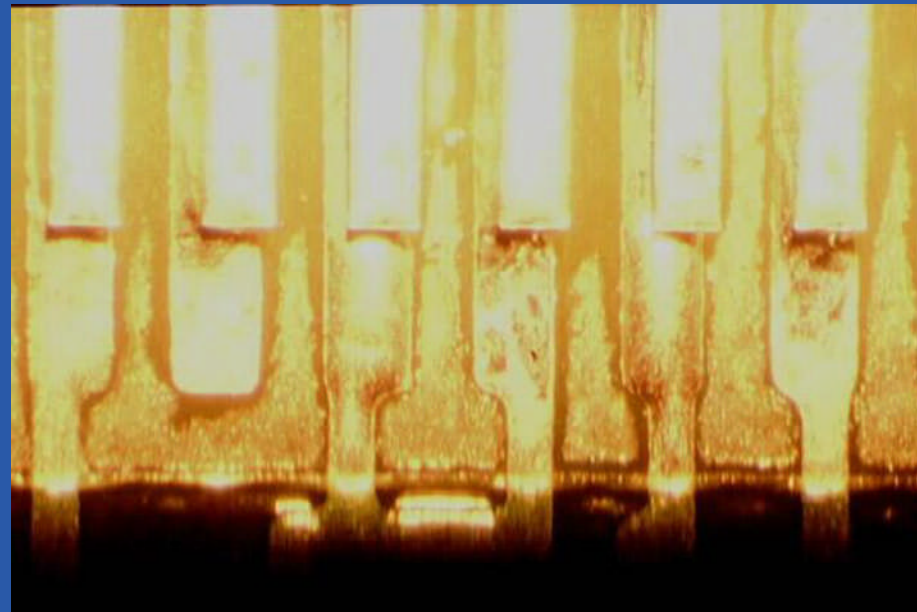
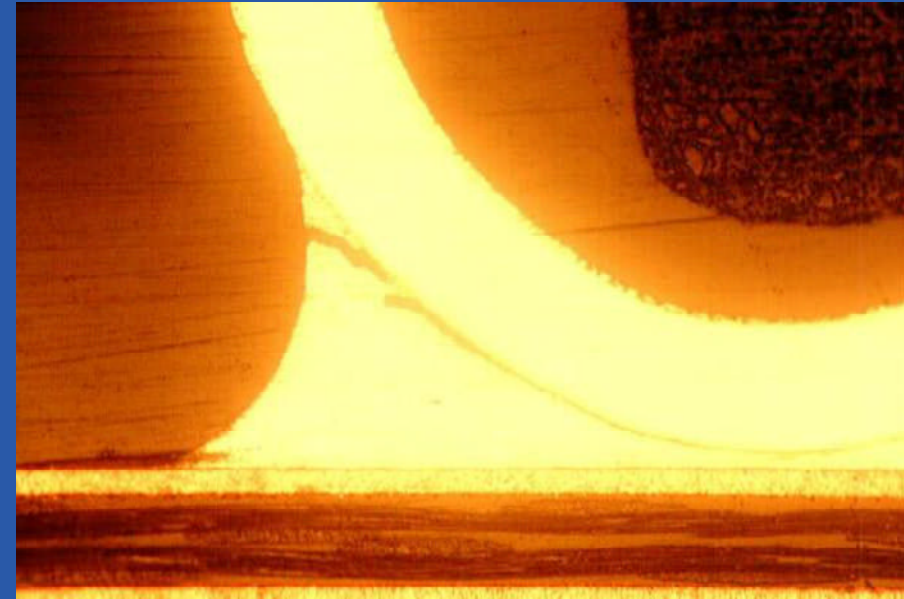
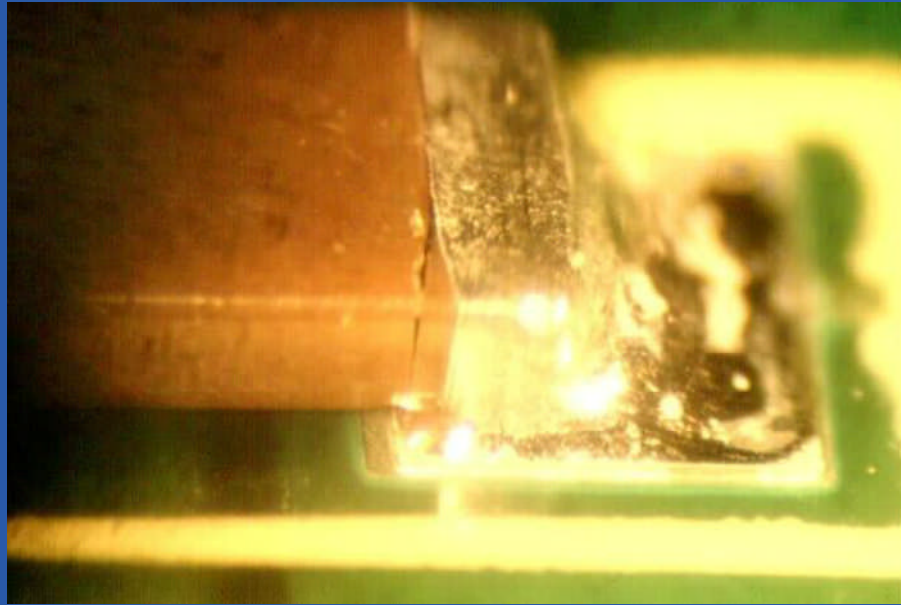


Open Circuit



Short Circuit

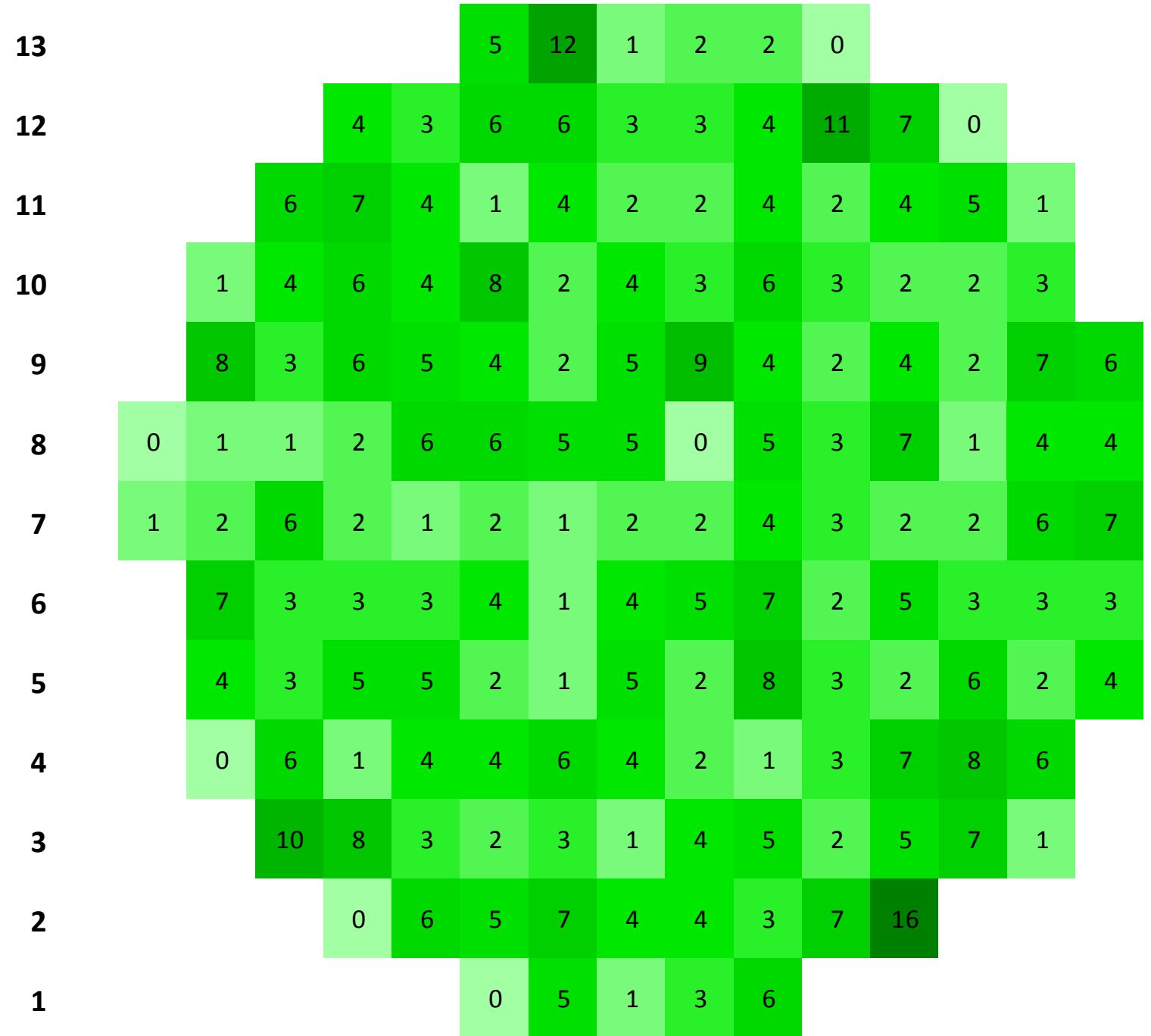
POSSIBLE CAUSES OF LATENT SYSTEM FAILURES



CHALLENGES DIAGNOSING “FUNCTIONAL” FAILURES

- Complexity (# gates, # LoC, # Transactions)
 - Lack of Access (Density, SI)
 - Subtlety of Defects (Timing, SI, PI)
 - Intermittent Failures
 - Fault Containment (Block level)
 - Failure Propagation (Several Clock Cycles)
- * Requires Dedicated Expertise
- ** Application Level Diagnosis More Difficult

TRAFFIC

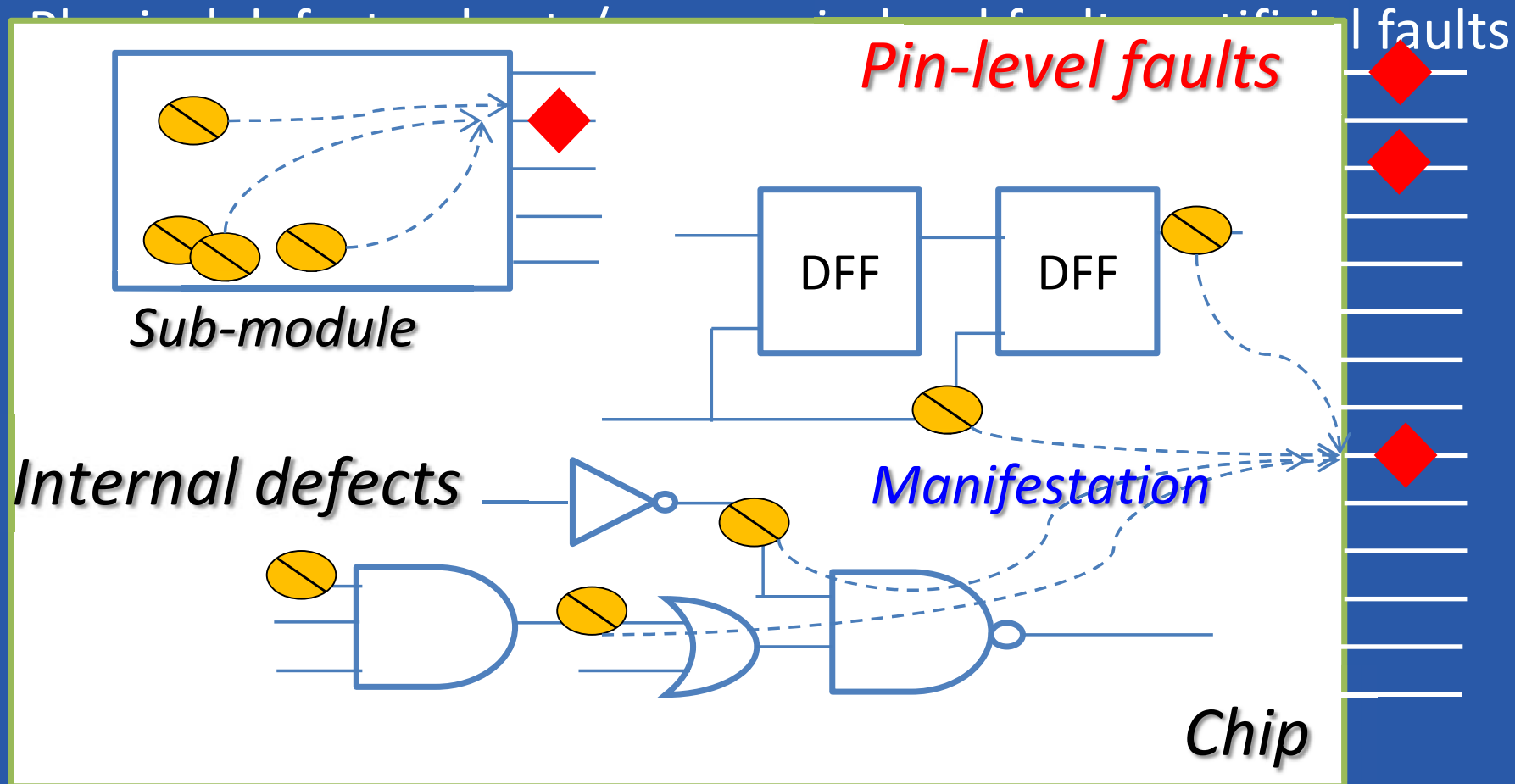


594

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

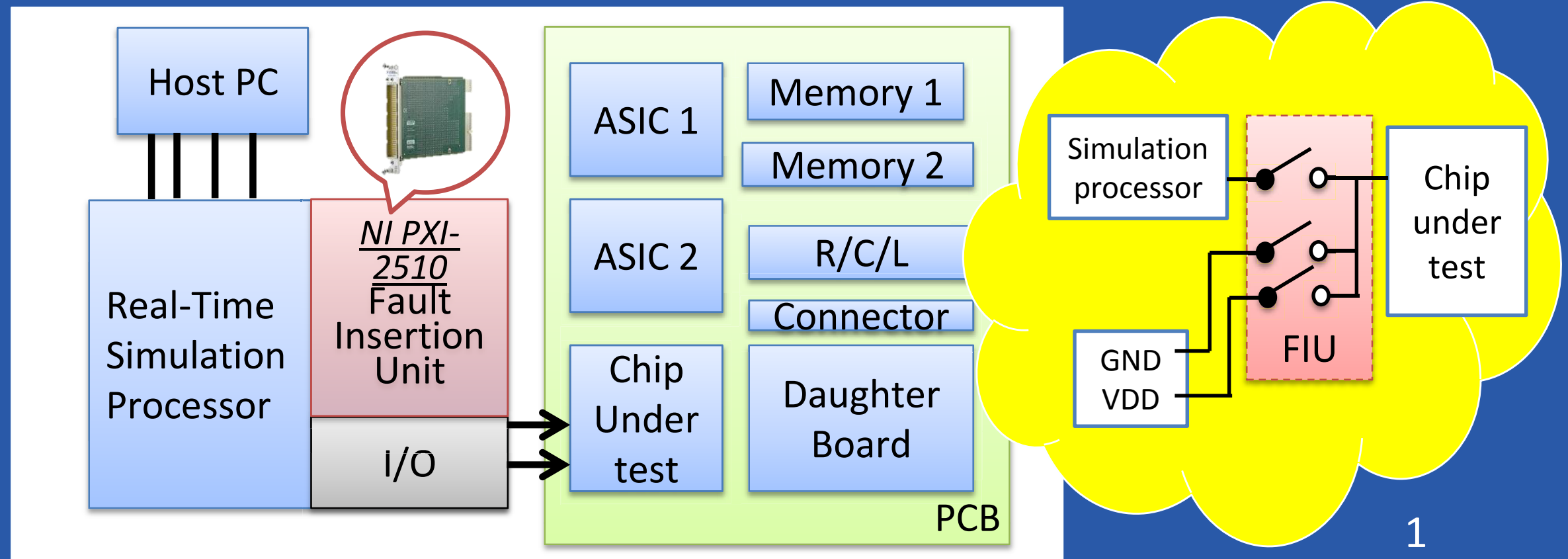
PHYSICAL DEFECT MODELING

- Construct pin-level fault model to mimic the errors caused by internal physical defects
 - One pin-level fault represents multiple internal defects



FAULT-INSERTION TEST (FIT)

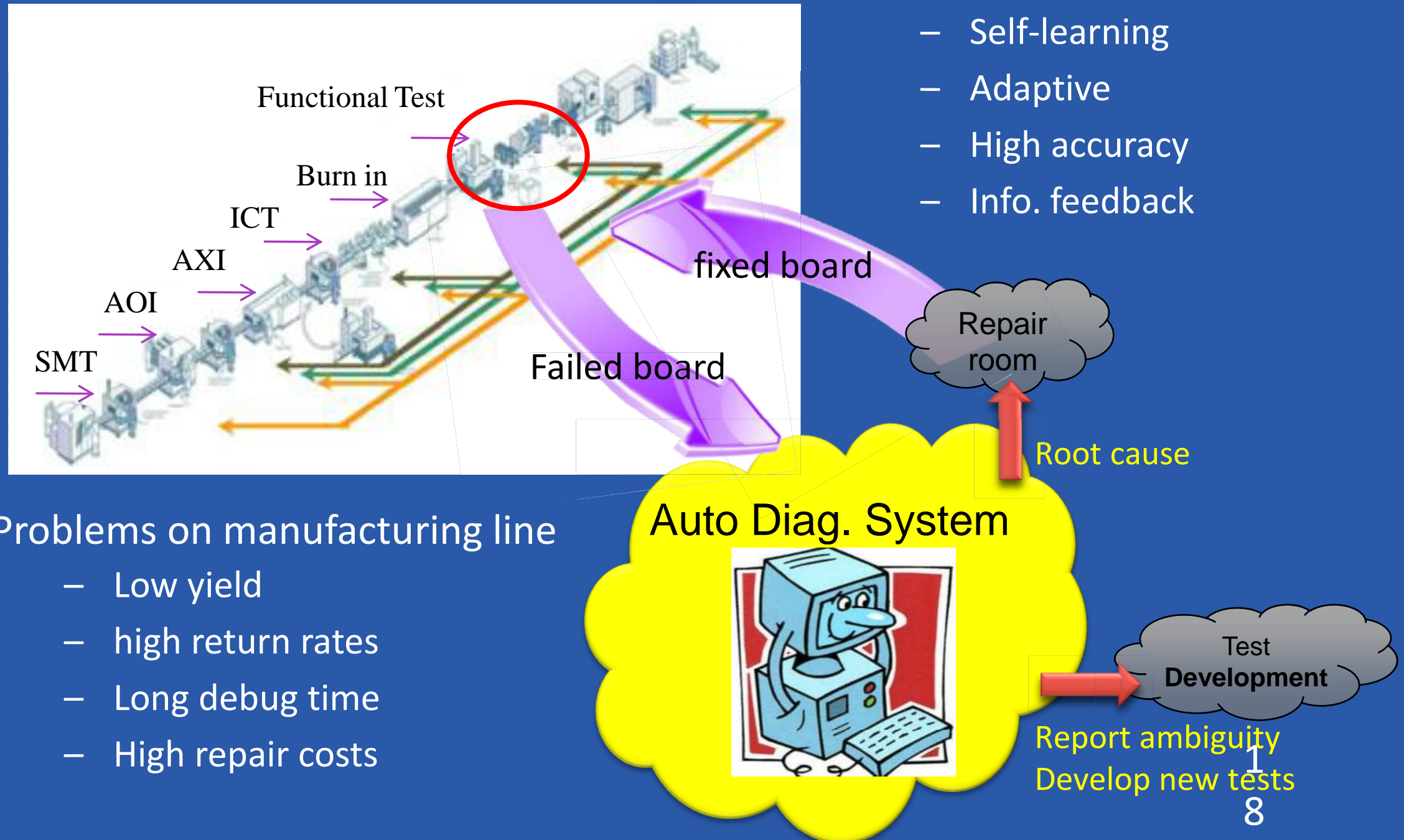
- Intentionally insert faults to evaluate system reliability and diagnostic programs
 - Verify error detection, error handling, recovery
 - Create faulty scenarios



DIAGNOSIS SYSTEM USING MACHINE LEARNING

- Proposed diag. system

- Self-learning
- Adaptive
- High accuracy
- Info. feedback



- Problems on manufacturing line

- Low yield
- high return rates
- Long debug time
- High repair costs

Report ambiguity
Develop new tests

RELIABILITY ECONOMICS

- Cost of Proactive vs Reactive
- Potential Consequences (cost, reputation)
- Defect free system = \$\$\$\$, and lead times
- Data Driven Fault Diagnosis/Containment
- Nordstroms/Lexus model – service trumps perfect
- Manufacturing and repair depot efficiency
- Reliability vs. Warranty

SMALL SYSTEM COMPLEXITY

SST
SST25VF080B
1 MB Serial Flash

SAMSUNG
Application
Processor and
DDR SDRAM

ST MICROELECTRONICS
LIS331 DL
Accelerometer

INFINEON
SMP3i
SMARTi Power
Management IC

SKYWORKS
SKY77340
Power Amp. Module

INFINEON
UMTS Transceiver

**NATIONAL
SEMICONDUCTOR**
LM2512AA
Display Interface

TRIQUINT
TQM666032
WCDMA/HSUPA
Power Amp.

BROADCOM
BCM5974
Touchscreen
Controller

TRIQUINT
TQM676031
WCDMA/HSUPA
Power Amp.

WOLFSON
WM6180C
Audio Codec

TRIQUINT
TQM616035
WCDMA/HSUPA
Power Amp.

INFINEON
PMB2525
Hammerhead II GPS

LINEAR TECHNOLOGY
LTC4088-2
Battery Charger/
USB Controller

NXP
Power Management

INFINEON
Digital Baseband
Processor

NUMONYX
PF38F3050M0Y0CE
16 MB NOR + 8 MB
Pseudo - SRAM



THANK YOU!