64th meeting of IFIP WG 10.4

Workshop on Dependability and Fault Tolerance

Report on Session 3 — Jean Arlat

Al Avižienis — The Architecture of a Resilience Infrastructure for Computing and Communication Systems

Jacob Abraham — Defects and Faults in Emerging Circuit Technologies

Hermann Kopetz — A Conceptual Model for the Information Transfer in System of Systems

Visegrád, Hungary — June 27-30, 2013

Al Avižienis Talk 1/2

Design a generic, FT, SW-free Resilience Infrastructure (RI)

- **About Resilience**
  - Elaboration on J.-C. Laprie Definition (Dependability when facing changes)
"Changes" -> "Harmful changes"
    - Exceed limits of expected threats
    - Unexpected threats
  - How is resilience created?
    - Implicit — Exceed specifications requirements: a) inadvertently, b) deliberately
    - Explicit — Add new features to system architecture to provide Resilience

Comment on related assumptions

- **Resilience Infrastructure to provide Resilience to a Client**
  - Physically separate (failure independence) from client
  - Generic to be able to serve any client
  - HW/firmware implemented
  - Fully self-protecting via HW FT techniques

Will this HW orientation still allow for adaptation?

Programmable HW

Al Avižienis Talk 2/2

- Installation of the RI
  - Client formed of N subsystems (C-Nodes) —> Error-confinement region
  - Monitor node (M-node)
    - ROM, non volatile status register
    - S3 = Startup-Shutdown-Survival = multiple pairs of self-checking pairs
  - M-node cluster the M-Cluster (patented): TMR + 2 spares
- Possible target for investigation/deployement of the RI ?
  - Human Exploration of Mars project
  - Very demanding level of resilience (1000 day manned mission)
  - RI Compatible with other FT features; it will "guard the guardians"
  - Absence of SW a major feature

- Further comments and questions

Importance of interfaces

Role of simplicity in design

Status messages protection?
I am alive messages protected by fail-safe coding

Probably complementary actions at SW level neded?

Jacob Abraham Talk 1/2

- Historical pesrpective
  - IC origin (late 50's),
  - Original Moore's Graph (mid 60's)
- IC complexity and computing power
  - For past 3 decades: Transistor # x 2 every 26 months
  - 32 nm in full production, 7 probably doable…
  - Exponential rate of Emerging Technology for 110 years (Kurzweil)
- Challenges
  - Defects (development faults) = Manufacturing, wearout; Design bugs
  - Faults (in operation) = HW: process-related, environmental; SW: bugs; System: external attacks
- Defect effects are "dynamic" -> new test methods (beyond stuck-at)
- (Manufacturing) Fault-Tolerant ICs:
  - Memories, FPGAs
  - Carbon nanotube circuits
    - Self-assembled, so defects are to be expected "by design"
    - Defect tolerant designs — not in production, actually

Jacob Abraham Talk 2/2

- **How to achieve printed features smaller than Lithography wavelength?** Anticipate the distorsion ☺
- **Line edge roughness and line width —>** delays, power leakage
- **Examples of significant fluctuations**
  - Dopant -> soon only few 10th of atoms in Channel -> quantum physics effects!
  - Gate oxide thickness —> MOS with actual Metal (intead of polysilicon)
  - Temperature
  - Dynamic voltage and power variations

What about 3D structures?

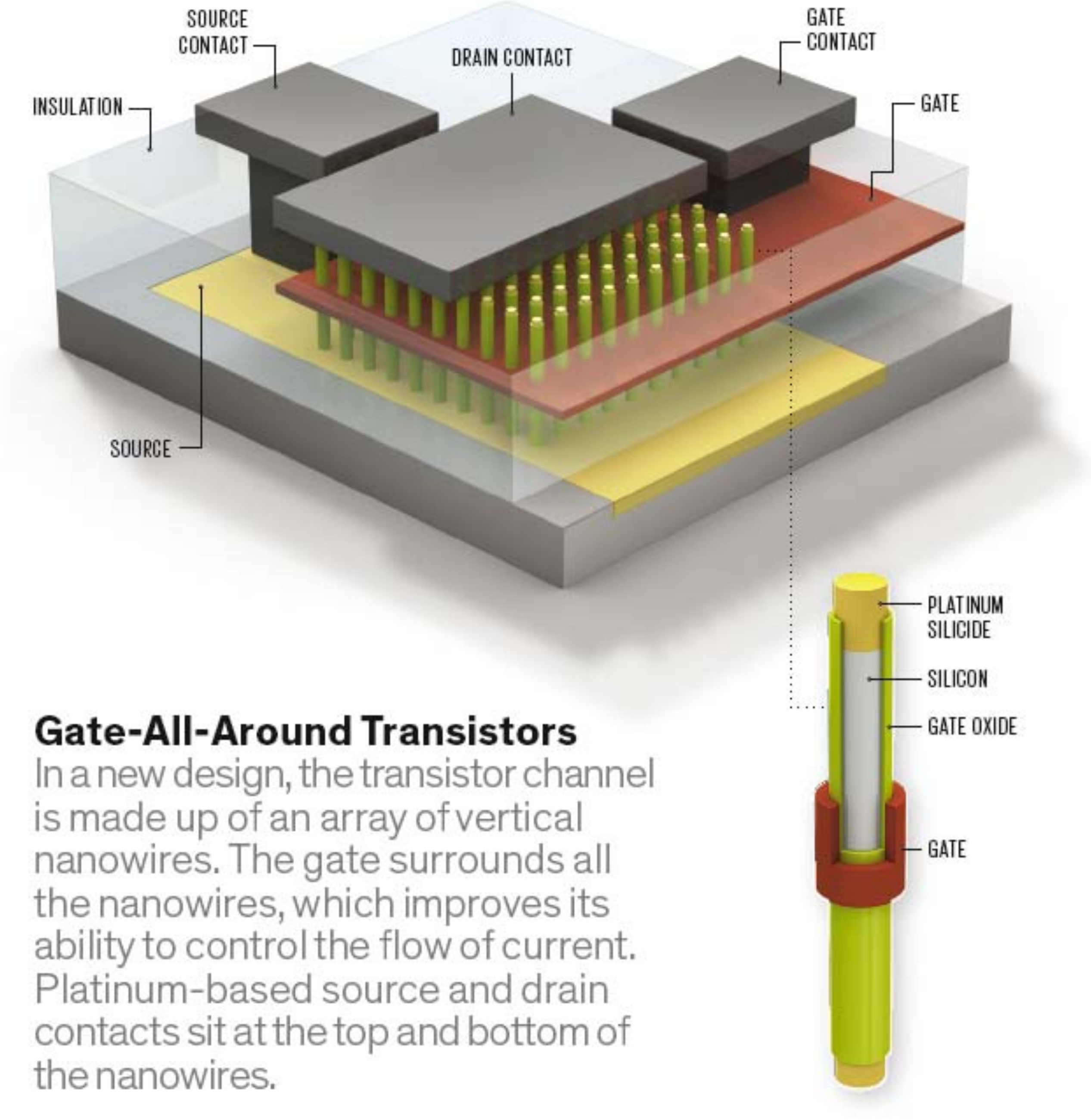
Communication delays

- Fault processing in operation
  - Circuit level: Transient error detection via delayed signal latching (shadow latch)
  - Application level: Checksum (example JPG picture)

Do we really care about the increase in HW-level faults

Manufacturers care about defects more than fault in operation

Technology Trend: Nanowire FETs'



Gate-All-Around Transistors

In a new design, the transistor channel is made up of an array of vertical nanowires. The gate surrounds all the nanowires, which improves its ability to control the flow of current. Platinum-based source and drain contacts sit at the top and bottom of the nanowires.

Ring Around the Nanowire

News Section, IEEE Spectrum, May 2013, pp.14-16

Hermann Kopetz Talk 1/2

Systems of Systems — Focus on Information representation

- Semantic vs representation of information
- Impact of inadequacies at the semantic level

- Itom = Information Atom (data + explanation of the data)
  - Data = artifact
  - Explanation : *Gives meaning* to the data
- Afferent (input) *vs* Efferent (output) Data
  - Example: Electronic Toll Collection
- Explanation of the Data
  - Identification Purpose, Meaning, Time, Ownership
  - Cultural issues involved, Receiver: Human or Machine
- Representation of an Itom
  - *Markup languages*, such as *XML*
- Itoms properties
  - Name, Purpose, Thruthfulness (no assumption made), Temporal, Neutrality, Phycalism (storage)

Hermann Kopetz Talk 2/2

- Itoms for Humans
  - Understandability = Patterns, Symbols, … to represent the Itom are compatible with *conceptual landscape* in the human mind of receiver
  - Utility = User dependent, difficult to quantify
- Itoms for Machines
  - Data: Bit strings; Explanations: Computer instructions & explanation of purpose
  - Digital object data and Digital metadata
  - Recursion -> Data processable by Machine — Design of computer serves as an explanation for the meaning of the data
- Communication: Itoms exchange using Gateways

Comments and questions

Connection to Ontologies?

Emerging behaviors ?

Connection with OSI/ISO layers?

Timing issues not properly involved

Open systems vs. SoS?

General Discussion

- Bottleneck due to HW implementations
- On-chip monitors more observability OK; Security issues?
- Predictions based upon Analog aging monitors?
  Getting close to margins provides a possible trigger?
- Low level errors do no matter any longer?
  Much cheaper recovery mechanisms at application level
- HW manufacturers do not develop applications; they mostly care about the yield issue
  See ITRS recommendation for Reliability and Resilience

  2011 Edition/2012 Update: Design for Reliability and Resilience confirmed as "New long-term *Grand Challenge*" (together with design of concurrent software)
  "Design Technology for Resilience: A Fundamental Portion of DFM"
- Embeded systems more FT mech. needed at processor level
- Computation cores can be including extra nodes
- Strong dependence on Application wrt these statements