Physical Security

Embedded, Smart Card, Post-Quantum & Biometrics

Global outline

- **Part I - Embedded systems**
  - Paolo MAISTRI (paolo.maistri@imag.fr)
- **Part II – Biometric Systems**
  - Jean-Francois MAINGUET (Jean-francois.mainguet@cea.fr)
- **Part III – Quantum Crypto**
  - Mehdi MHALLA (mehdi.mhalla@imag.fr)
- **Part IV – Smart Cards**
  - Charles GUILLEMET (Charles.GUILLEMET@cea.fr)

Embedded Systems – Global Outline

- Introduction – Contents and objectives
- Part I.1 - Embedded system design and architectures
  - Basic concepts and hardware description language (~2h)
  - VHDL tutorial (~1h)
- Part I.2 - Design constraints, qualification, common criteria
- Part I.3 - Design and implementation of secured circuits and crypto processors

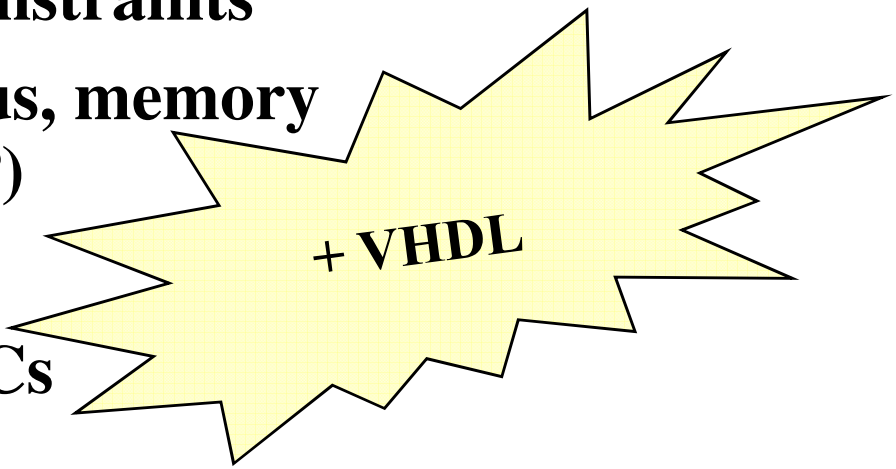
Warning !

- This is not a lecture on cryptography …
- …but an introduction to implementation-related issues …
- … with a special focus on hardware architecture and integrated circuit issues

  (including some basic reminders on hardware circuit design)
- Examples given on well known algorithms
  - AES for symmetric
  - RSA/ECC for asymmetric
  - …
- Prerequisites: digital circuit design, computer architecture (M1 courses).

Contents and Objectives (part 1 and 2)

- Embedded systems: definition and constraints
- Hardware architecture (integrated bus, memory architecture, advanced processors, IP)
- (Software architecture (RTOS, API))
- Integrated circuit basics – SoCs, SoPCs



- Understand global architecture of (integrated) embedded systems
- Understand basic blocks integrated in SoCs/SoPCs – Focus on hardware parts

- Secure circuits: design constraints, qualification, common criteria.

Contents and Objectives (part 3)

- Types of attacks, exploitation examples.
- Examples of secure architectures and protections (counter-measures).
- Fault/error models.
- Impact on test techniques.

- Understand the context and the security threats
- Understand the basics and limitations of protection techniques
- Prerequisites: basics in cryptography (M1 courses).

Evaluation (part I)

- Continuous control 1 (mini)
  - Simulation of Poster/Rump session
  - Document analysis and presentation
  - 5' talks max for each student, (mainly) no questions
  - Preparation W2-W3, presentation Oct 7th
- Continuous control 2 (full)
  - Simulation of full paper presentation
  - Document analysis and presentation
  - 15-20' talks by 2-student groups + questions
  - Preparation W4-W6, presentation Nov 4th
- Lab works
  - 15 hours
  - Basically two types of simulated hardware attacks

Part I
Embedded system design and architecture
Basic concepts

Outline – Part I

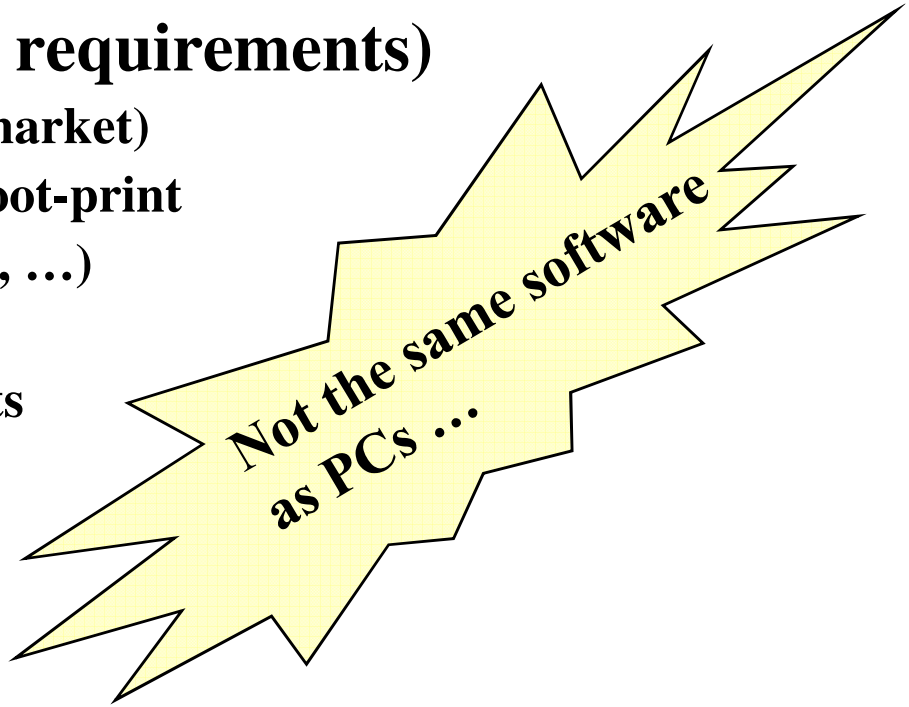
- Embedded systems – general notions
  - Definition and applications
  - Typical constraints
  - Overview of implementation technologies
- Basic blocks
  - Processors
  - Communication resources
  - Programmable arrays

Embedded systems?

- There is no formal definition of an embedded system, but it is generally accepted to be a type of computer designed to solve a specific (industrial) problem or task
- This is in contrast to a general-purpose computer such as a PC or workstation
- Embedded systems typically use one or several microprocessor(s) combined with other hardware and software
- Embedded system software ranges from a small executive to a large real-time operating system (RTOS) with a graphical user interface (GUI)
- Typically, the embedded system software must respond to events in a deterministic way and should be guaranteed not to crash

Embedded systems: general properties

- **Single-functioned**
  - Typically, is designed to perform predefined function
- **Tightly constrained (non functional requirements)**
  - Tuned for low cost (+ predictable time to market)
  - Single-to-fewer components based, small foot-print
    - Limited amount of resources (memory, …)
  - Performs functions fast enough
    - Differs from high performance markets
  - Consumes minimum power (battery …)
  - High expectations: reliability, security, etc.



- **Reactive and real-time**
  - Must continually monitor the desired environment and react to changes
  - Worst Case Execution Time (WCET) evaluation constraints (determinism !)
- **Hardware and software co-existence (+ HW/SW interface !)**

Embedded systems: application areas

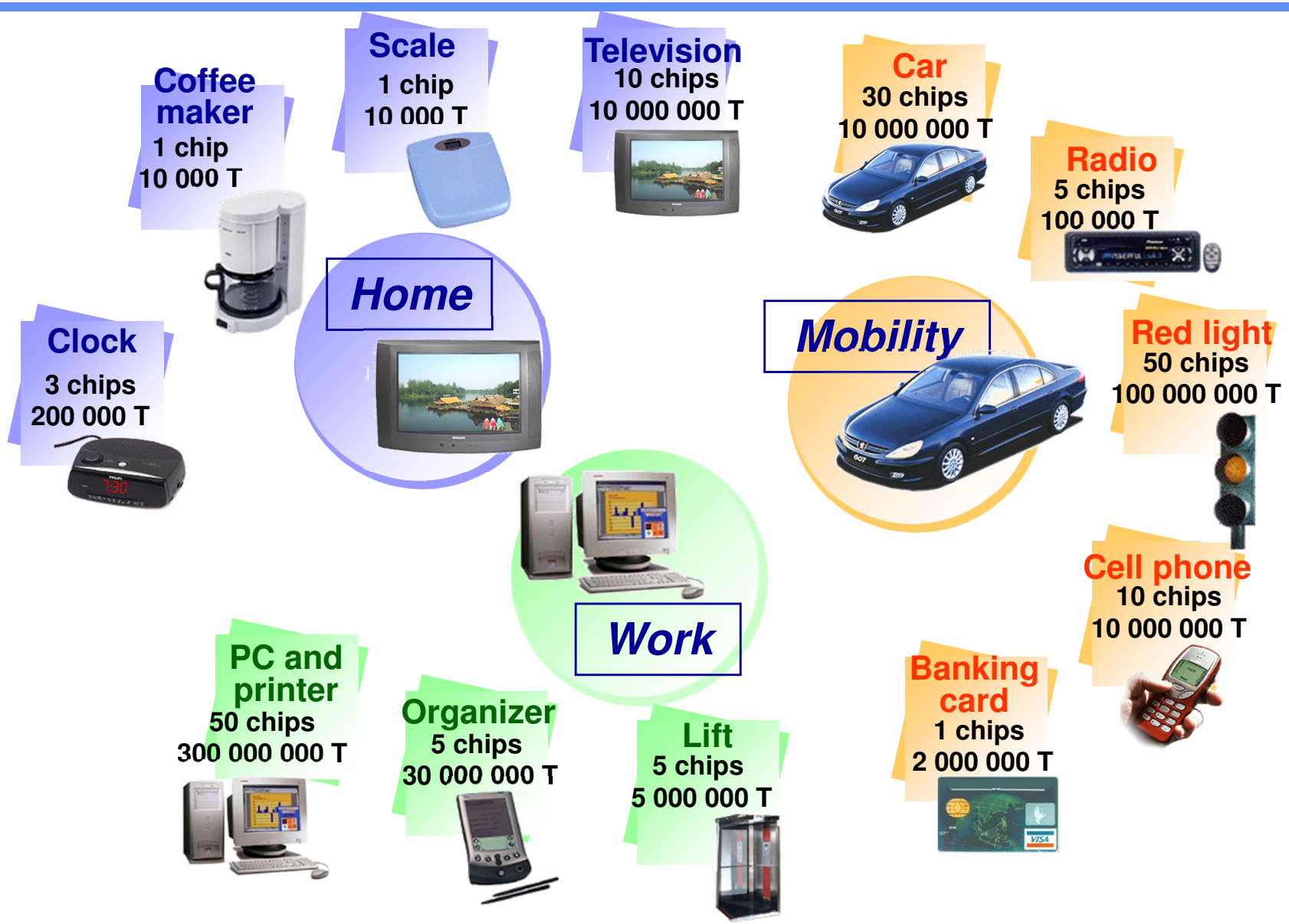
- The embedded system landscape is as diverse as the world's population:

  … no two systems are the same.

- Embedded systems range from large computers such as an air traffic control system to small computers such as a handheld device that fits into any pocket.

- Examples:
  - Communication devices: wired and wireless routers and switches
  - Automotive applications: braking systems, traction control, airbag release systems, and cruise-control applications
  - Aerospace applications: flight-control systems, engine controllers, auto-pilots and passenger in-flight entertainment systems
  - Defense systems: radar systems, fighter aircraft flight-control systems, radio systems, and missile guidance systems

Embedded systems => Integrated circuits



Embedded systems and dependability

- Increasing number of application fields ranging from consumer electronics to very critical application fields:
  - Transports
  - Energy (e.g. nuclear plants)
  - Human security/protection devices (e.g. factory secured areas)
  - Access control, identity
  - Pay-per-view TV or similar services, banking
  - …
- Note: criticality comes from several points of view – from human injuries to financial losses

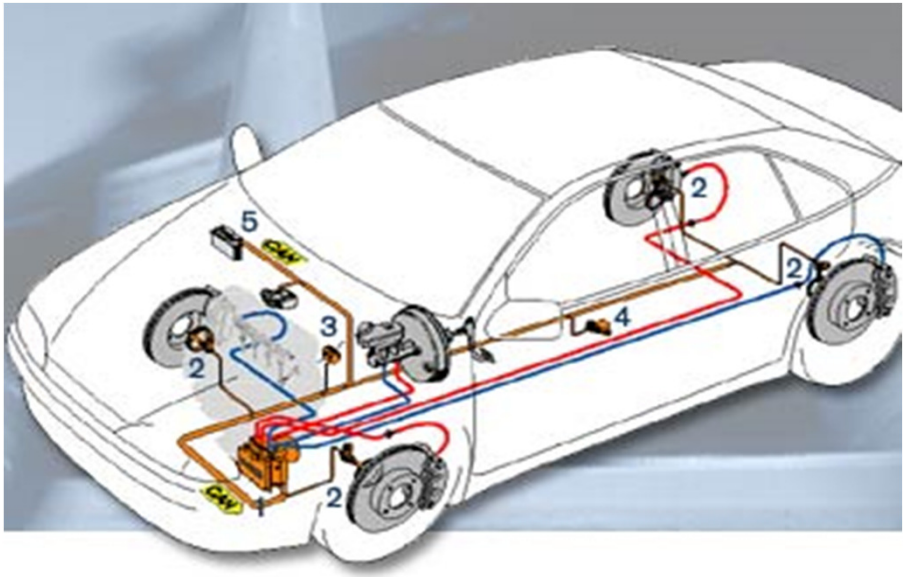
Automotive embedded systems: overview

- Today's vehicle networks are truly distributed electronic systems (70+ nodes (=ECUs) [1]).

- Cars contain numerous (10+) heterogeneous time or event driven bus systems
  - CAN, LIN, FlexRay, MOST

Most are critical

- x-by-wire
- steering aids, ABS, ESP(DSC)
- remote window and lock control
- engine control
- airbag control
- navigation systems
- entertainment systems



[1] P. Hansen. New s-class mercedes: Pioneering electronics. *The Hansen Report on Automotive Electronics*, 18(8):1–2, October 2005.

Automotive embedded systems: typical properties

- Software is mission critical
  - Highly dependable
  - Hard real-time
  - Typically statically scheduled and bound
- Lifetime is rather long (10-14 years)
  - Modular design
  - Exchangeable components (modules)
- Systems are produced in high quantities (56.3 million cars in 2005)
  - Costs have to be small
  - Bug fixes are extremely expensive

Note: "sécurité" (security) is a concern … but mainly means "harmlessness" (safety), not "confidentiality" … until today …

Car Hijacking


WIRED

Hackers Remotely Kill a Jeep on the Highway—With Me in It

BUSINESS | CULTURE | DESIGN | GEAR | SCIENCE | SECURITY

ANDY GREENBERG SECURITY 07.21.15 6:00 AM

HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT

- The vents in the Jeep Cherokee started blasting cold air
- the radio switched to the local hip hop station
- the windshield wipers turned on
- a picture of the two hackers performing these stunts appeared on the car's digital display
- They cut the transmission. … my accelerator stopped working. … they cut the brakes, leaving the SUV slid into a ditch
- Practically all carmakers are turning the automobile into a smartphone […] whose cellular connection also lets anyone gain access from anywhere

Another type of embedded system …

Note: "sécurité" is again a concern … but here it means "confidentiality"

Societal context

Internet & Computer Networks, trusted infrastructures, trusted communications

Electronic payments

Electronic voting

Electronic passport, access control

International competition, counterfeiting, intellectual property

Ubiquitous computing and privacy

Etc …

Other example … implanted devices

From …

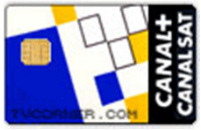
… to …

Today, a pace-maker is a small computer able to analyze the heart electrical signals that can be tested and reprogrammed by telemetry (contactless communications)

Dependability constraints (with minimum power consumption !):
Reliability: difficult replacement !
Safety: EM perturbations (e.g. iPod …)
http://www.techshout.com/ipod/2007/11/ipods-can-cause-pacemakers-to-malfunction-study/
Security: malicious activation/deprogrammation by a hacker
http://www.newscientist.com/blog/technology/2008/03/death-by-radio-waves-hacking-pacemaker.html

Historical targets: systems to clone (or modify)

- Unauthorized access to services (Transport, Pay-TV, Communications, …)



- Illegal duplications (e.g. duplication of games, DVDs, …)

- Identity (Building access, Identity documents, …)

- Credit cards (but also electronic purses … and smartphones)

- …

Targets are increasing with evolutions

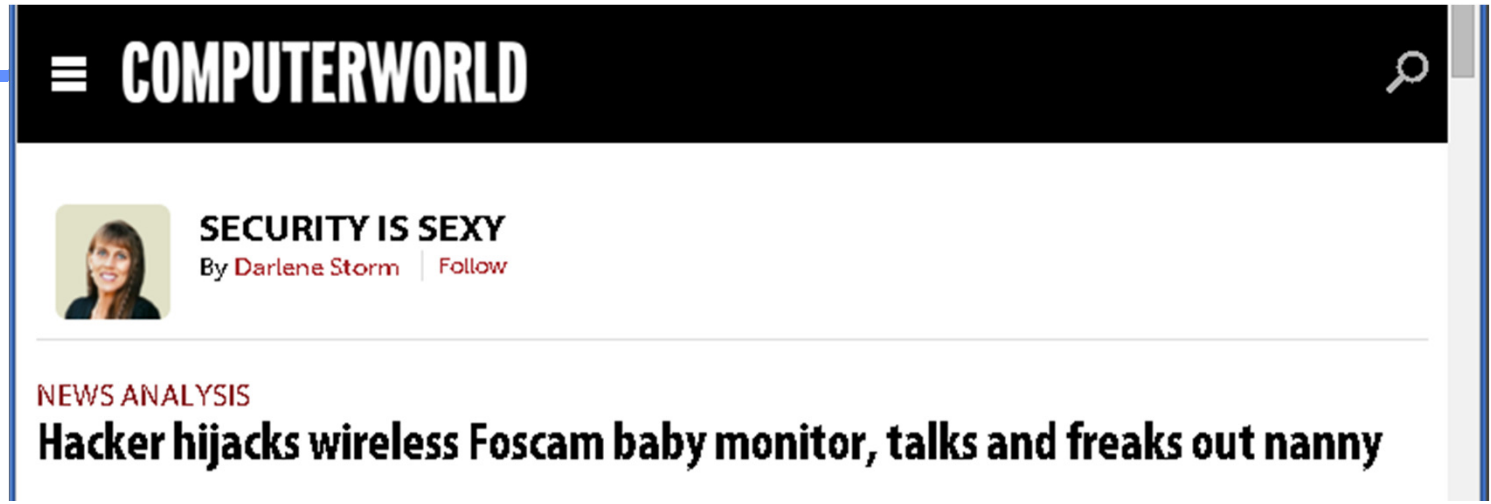
- Technological evolutions combined with social evolutions
  - Counterfeiting fighting: luxury products, but also automotive spare parts …

  - More devices related to health (available on the market): communicating pacemakers (tomorrow: brain control), insuline pumps, …

  - More remote (electronic) control on energy-related devices ("smart grids"): a new target for potential hacking, must be secured

  - Future automotive systems ("X-by-wire"): brakes, but also all other controls (e.g. steering) => not only safety-related aspects have to be taken into account during design

Foscam Hack


COMPUTERWORLD

SECURITY IS SEXY

By Darlene Storm | Follow

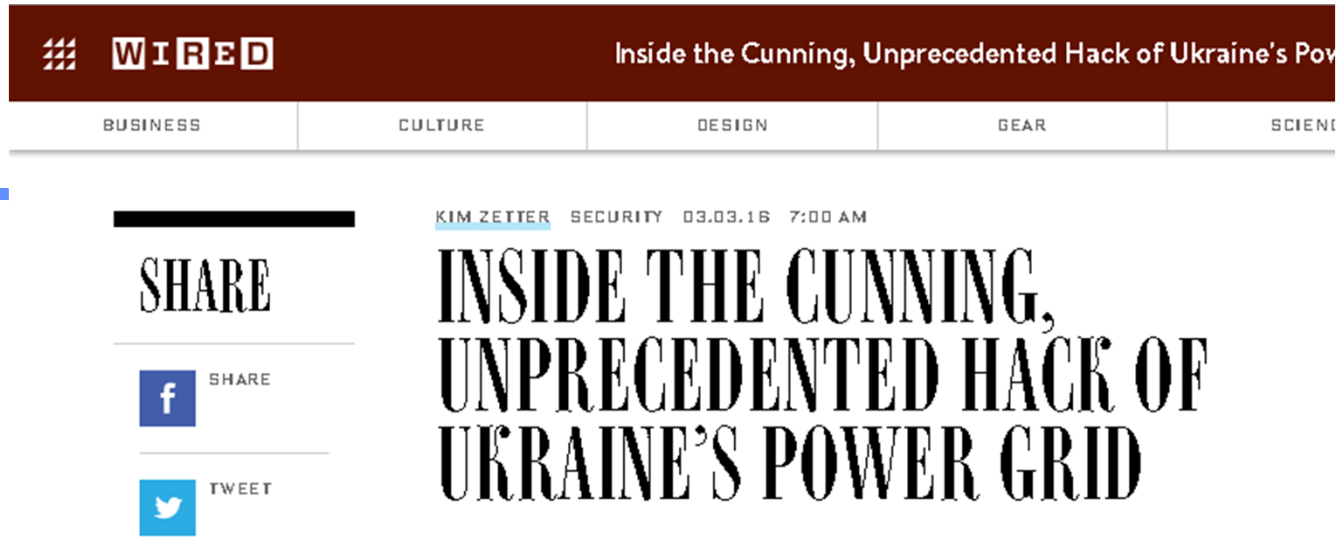
NEWS ANALYSIS

Hacker hijacks wireless Foscam baby monitor, talks and freaks out nanny

- 2013: spy on and hurl curse words at a two-year-old girl
- 2014: screaming at a 10-month-old girl in Ohio before screaming obscenities at her dad
- 2015: third time hijacking a wireless camera/baby monitor made a virtual intrusion known by talking

- What about all the other silent intrusions?

Power Grids


WIRED

Inside the Cunning, Unprecedented Hack of Ukraine's Pov

BUSINESS | CULTURE | DESIGN | GEAR | SCIEN

SHARE

SHARE

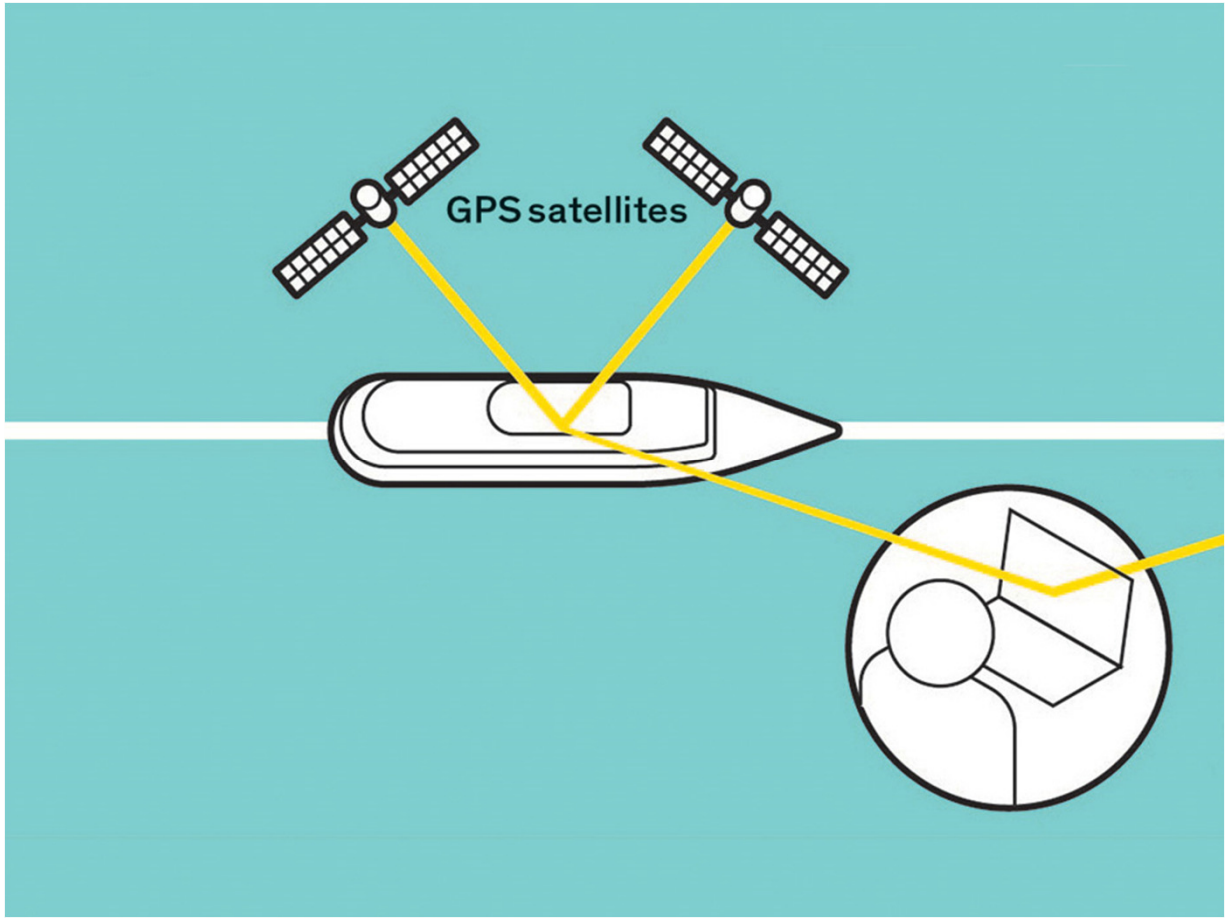
TWEET

KIM ZETTER SECURITY 03.03.16 7:00 AM

INSIDE THE CUNNING, UNPRECEDENTED HACK OF UKRAINE'S POWER GRID

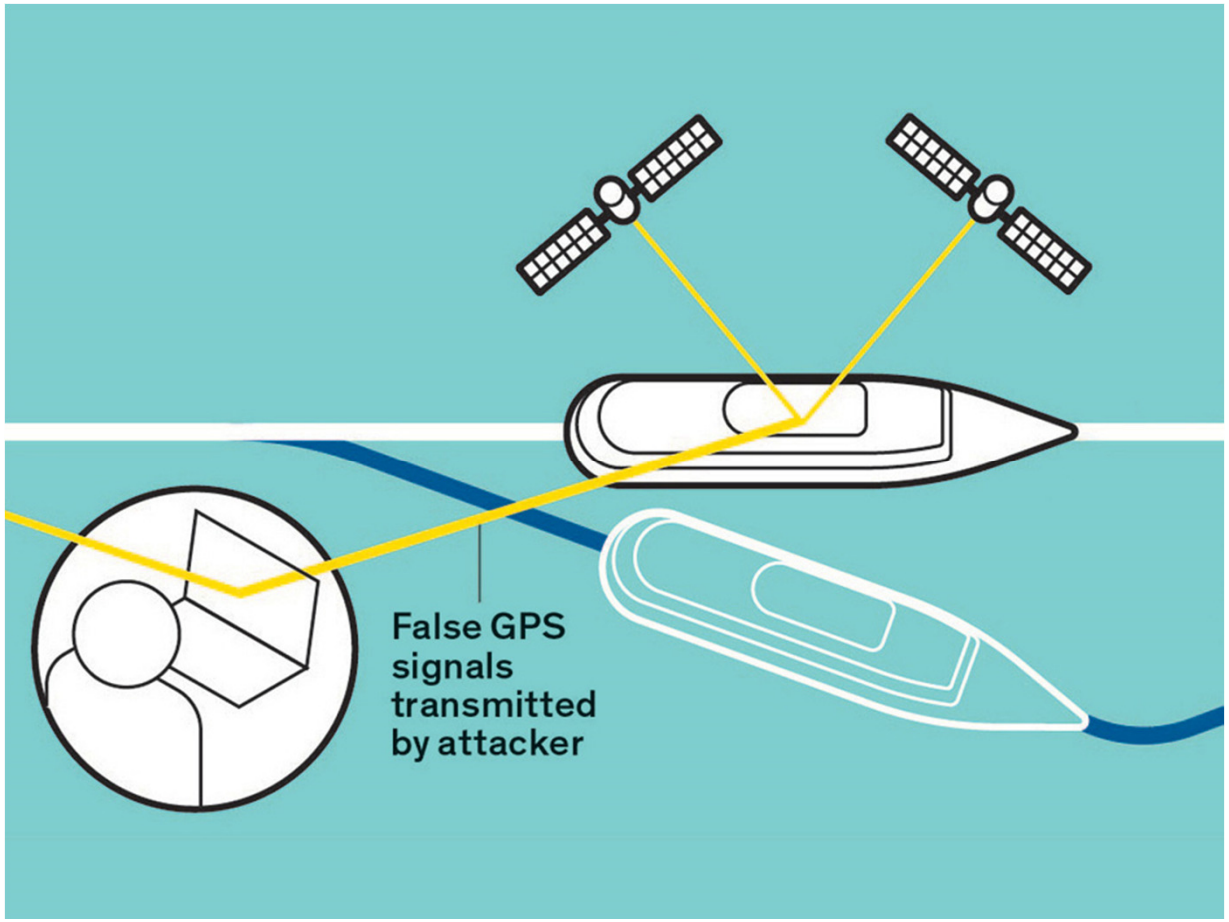
- Three attacks. Thirty minutes apart. Against three electrical substations serving Ukraine's power grid.
  - First known cyber-attack of its kind
  - Confirmed by US Cyber Emergency Response Team
- Timeline
  - Phishing campaign opening backdoors
  - Explore and map the networks
  - Harvest VPN credentials
  - Access the SCADA network

GPS



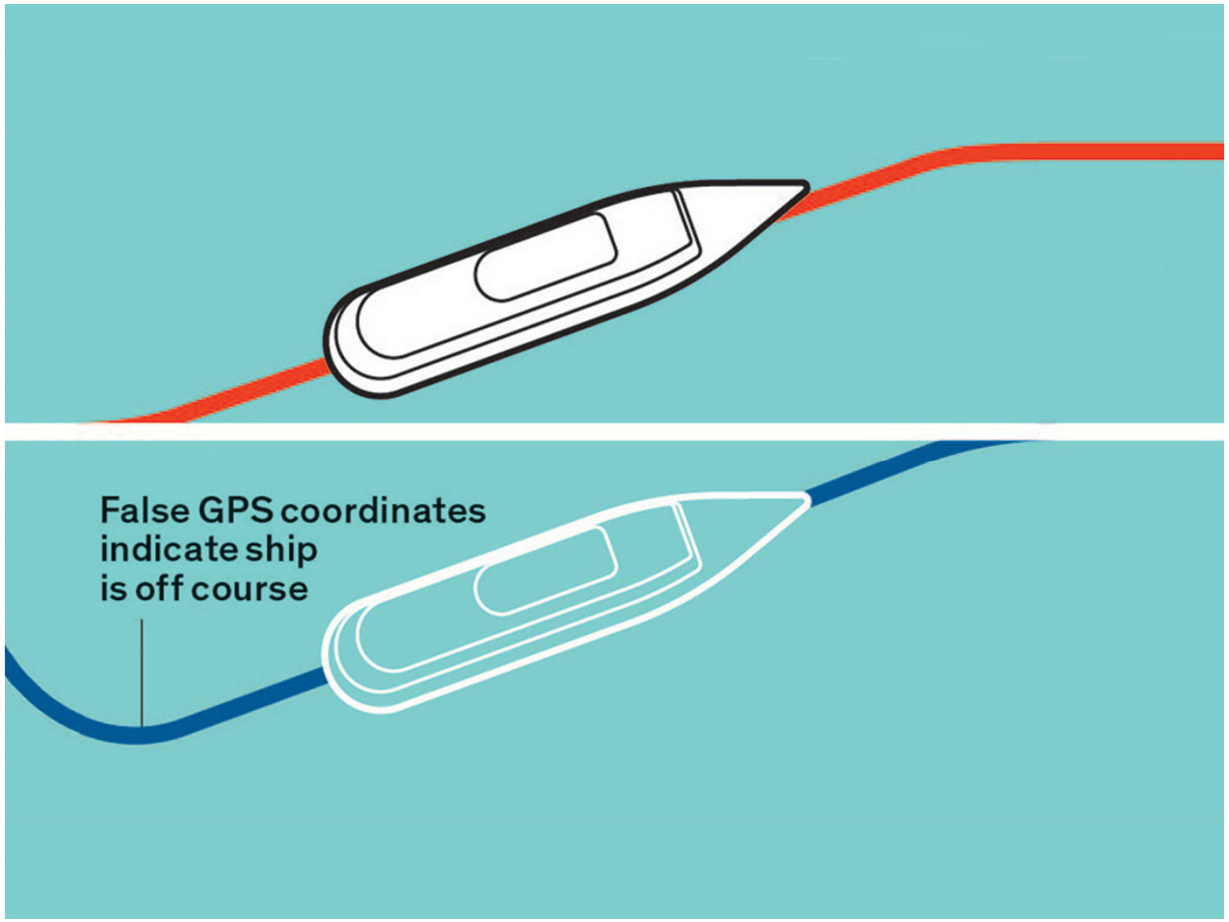
A ship's crew relies on GPS signals emitted from a constellation of satellites to safely navigate the seas [IEEE Spectrum, 07/2016]

GPS



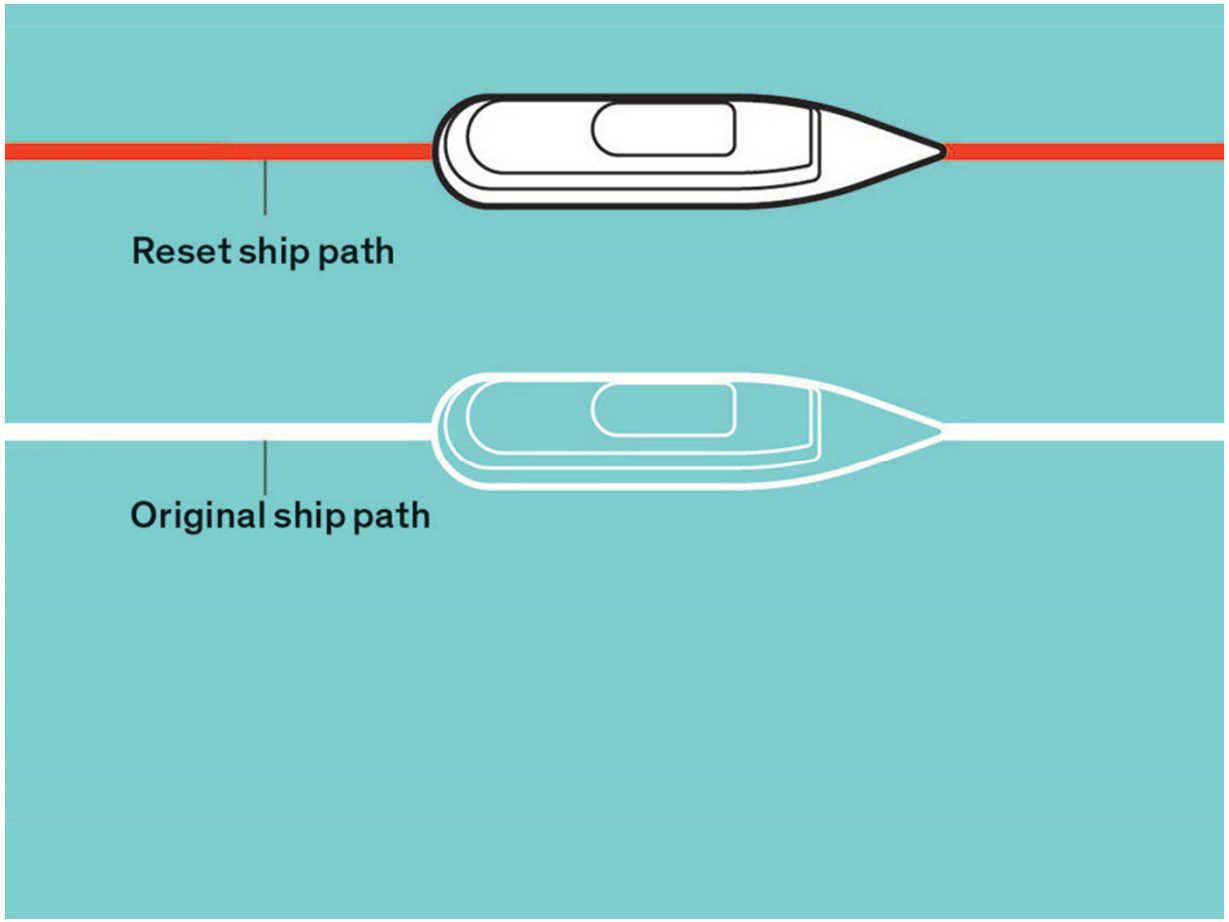
To spoof a vessel, an attacker transmits false GPS signals to override the signals the ship receives from these satellites.

GPS



Then, the attacker adjusts the coordinates so that the crew believes the ship has blown off course (blue line)

GPS



When the crew resets the ship's path, they unwittingly guide it onto a new route (red line)

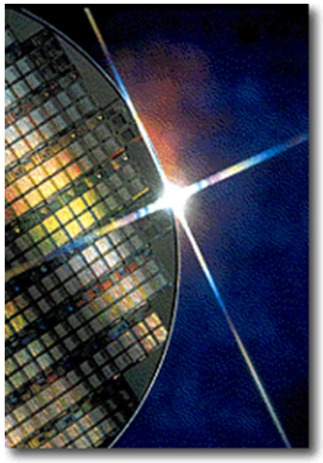
Implementation technologies

- Microcontroller-based systems
- DSP-based systems

} Electronics board or System on Chip (SoC)

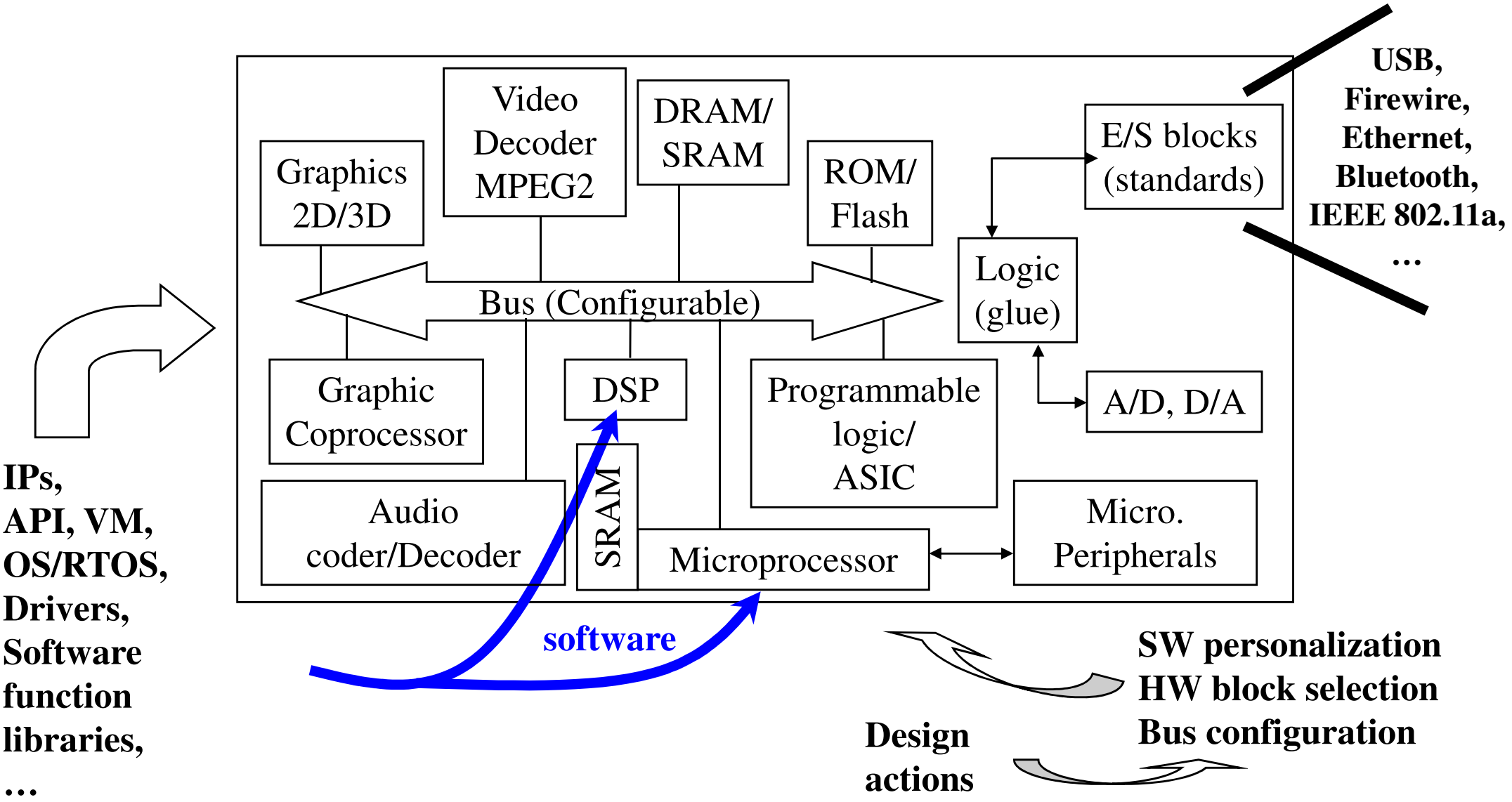
- ASIC technology
- Programmable array (FPGA) technology

  => Reconfigurable architectures

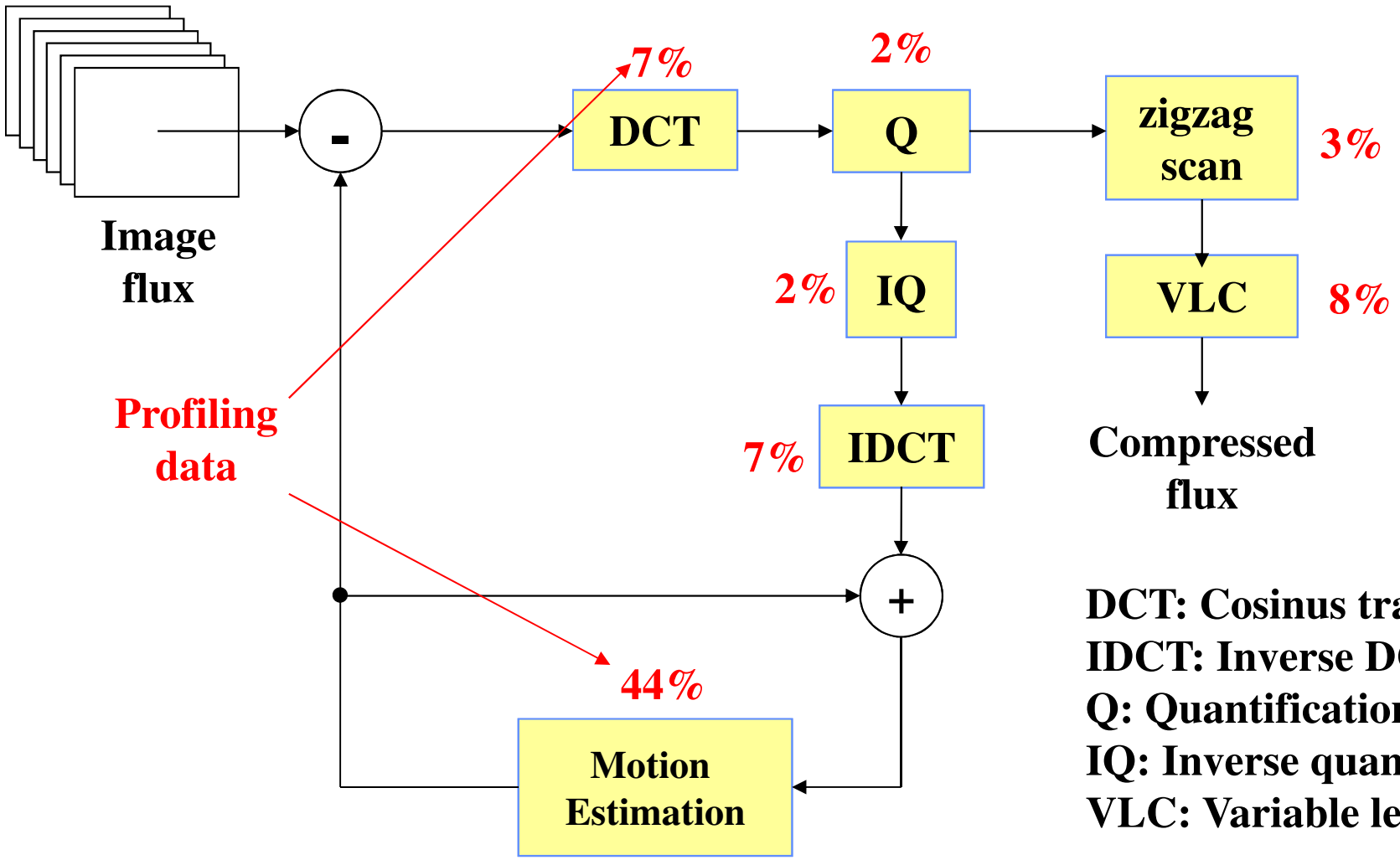


- Embedded system design involves:
  - Hardware (Board/ASIC/FPGA) design
  - Drivers for hardware (C …)
  - Software development

An (integrated) electronic system



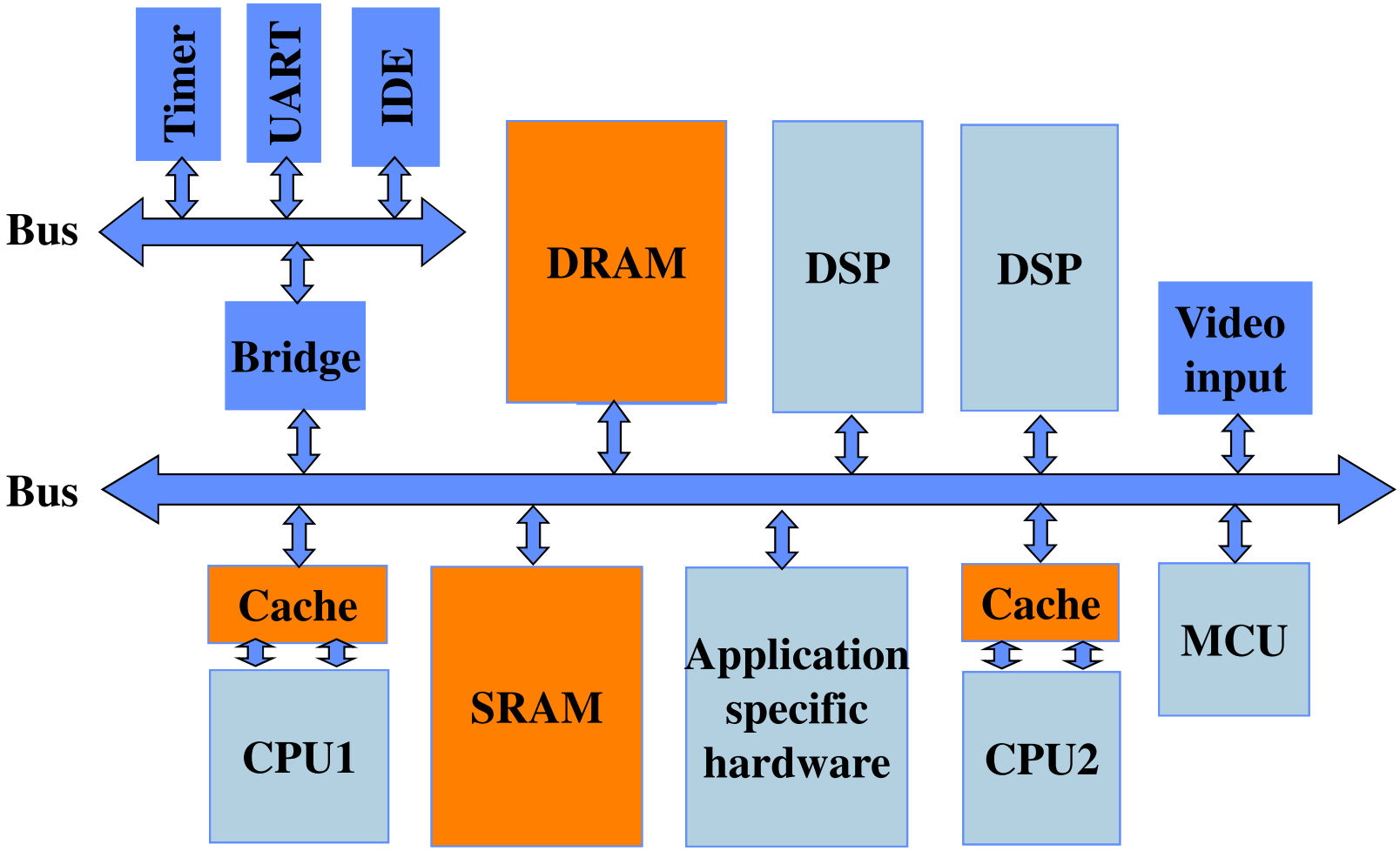
Co-Design: MPEG example



DCT: Cosinus transform
IDCT: Inverse DCT
Q: Quantification
IQ: Inverse quantification
VLC: Variable length coding

Source: Steven Derrien, IRISA, Vannes pedagogic days, 2006

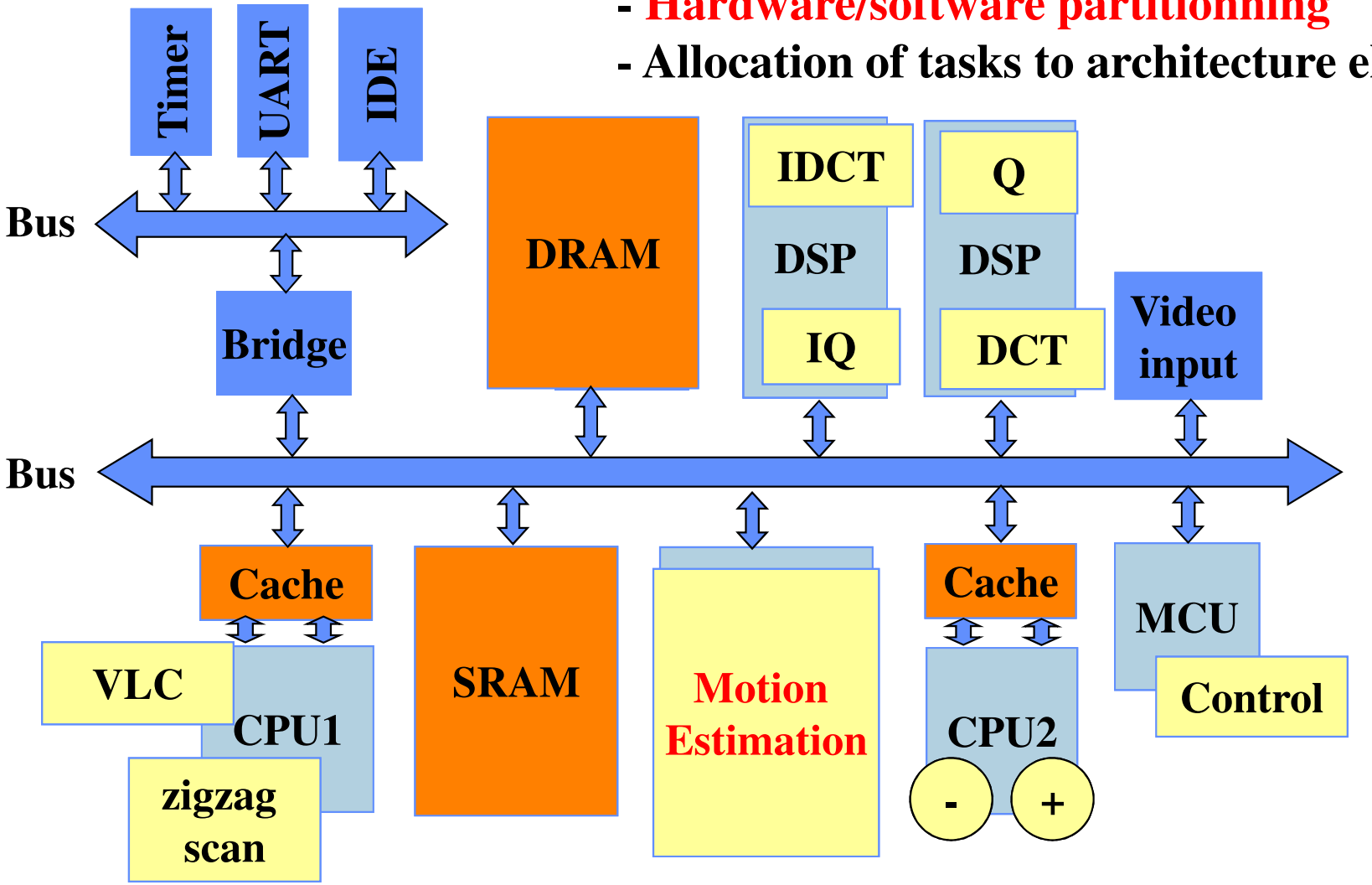
MPEG example – architecture definition



Source: Steven Derrien, IRISA, Vannes pedagogic days, 2006

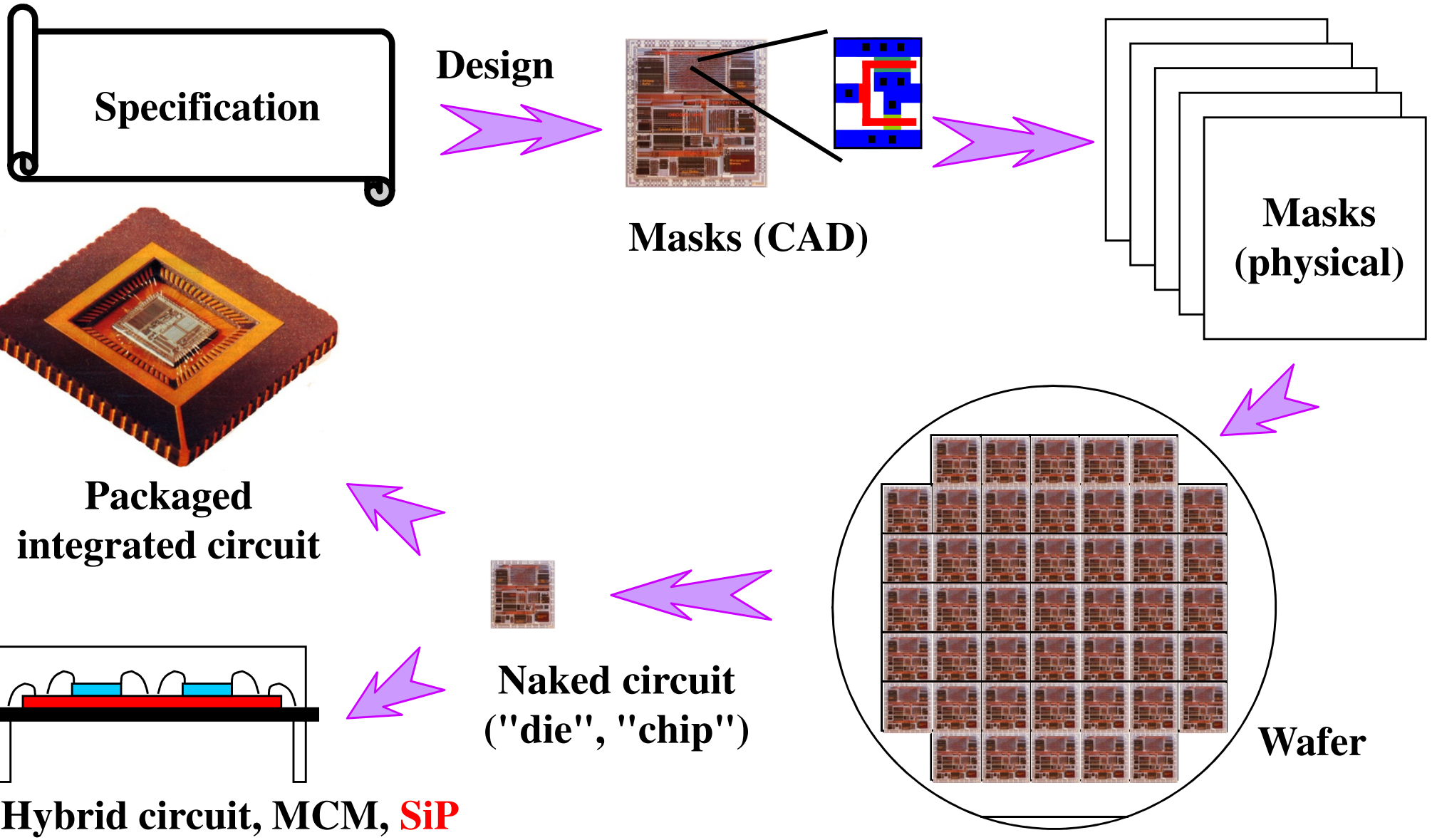
MPEG example – mapping of functions

- Hardware/software partitionning
- Allocation of tasks to architecture elements

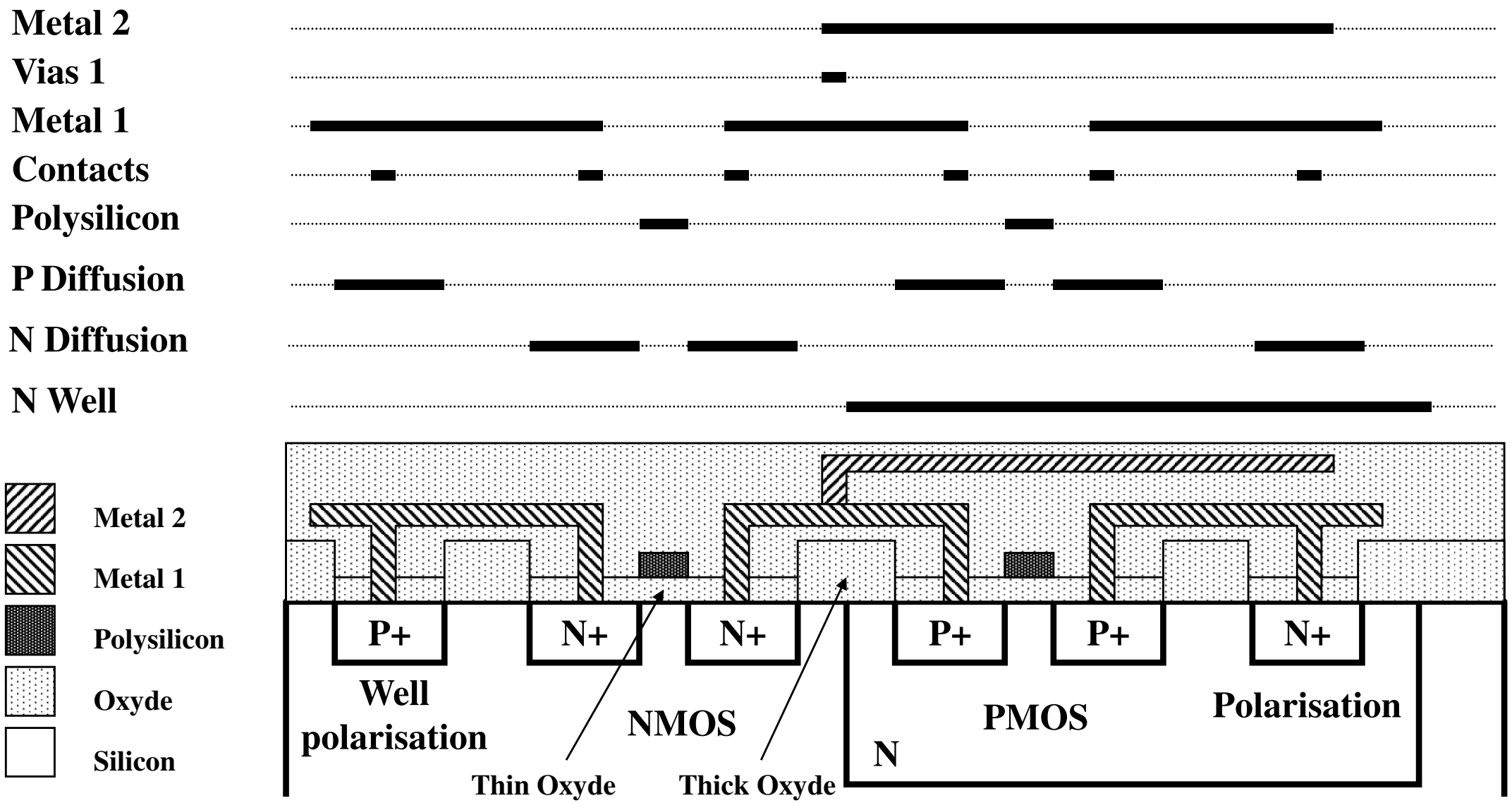


Source: Steven Derrien, IRISA, Vannes pedagogic days, 2006

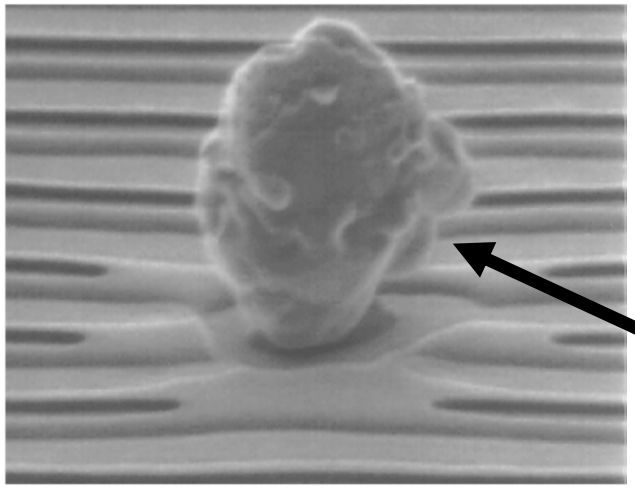
From design to physical component



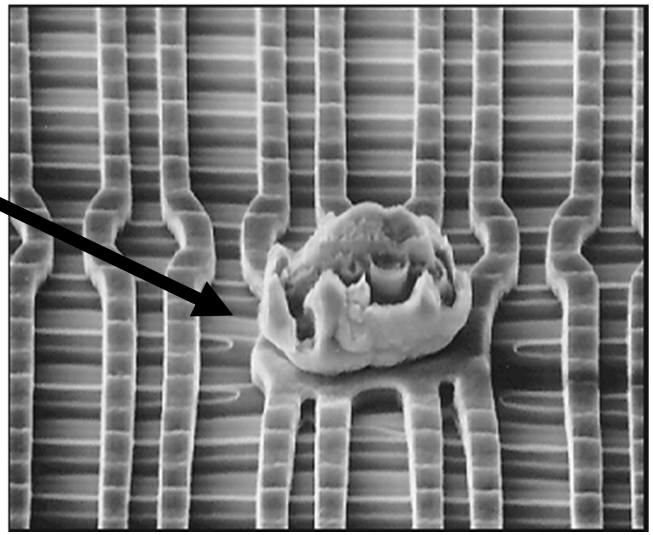
Simplified masks (CMOS inverter)



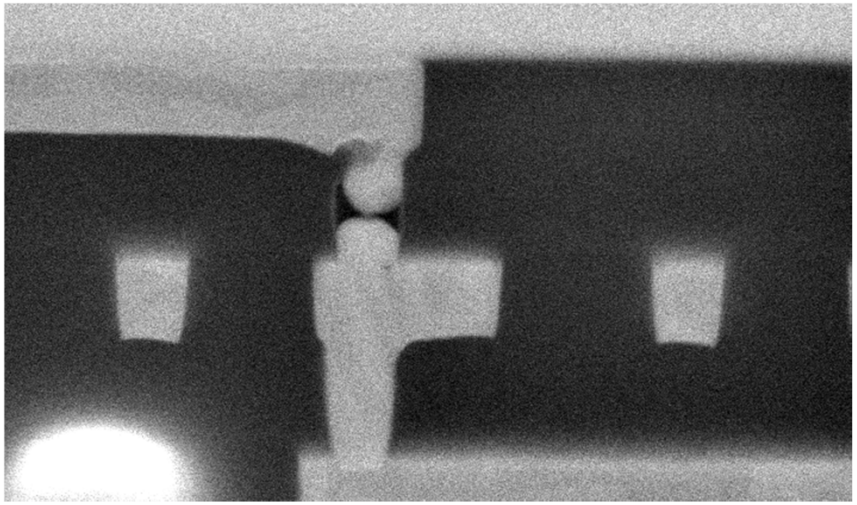
Manufacturing defects: examples

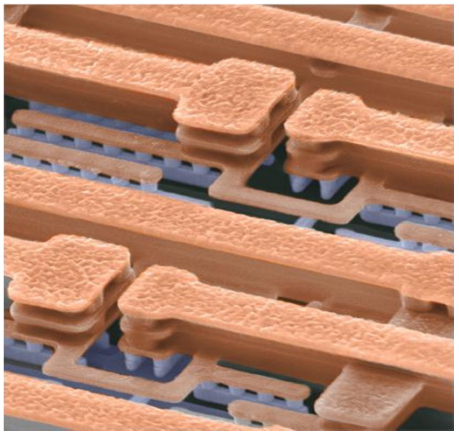


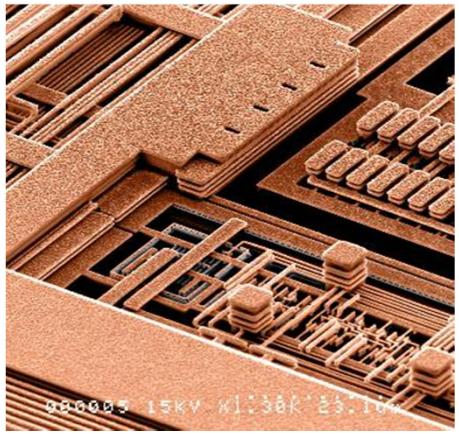
Unexpected connection
=> short circuit



Interconnection break
=> open circuit

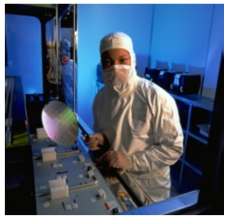




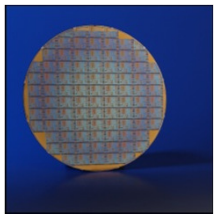


Source: IBM

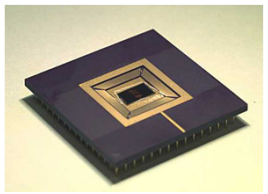
After manufacturing: Testing!



Manufacturing (process)



Processed wafers

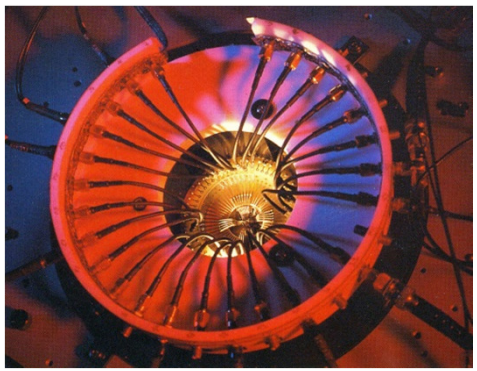


Die slicing/ assembling

Accelerated aging

Visual controls (options)

Probing-based tests



Test in package
- **parametric**
- **consumption**
- **functional (nominal and extrem conditions)**
- **dynamic (performances)**

Burn-in (option)

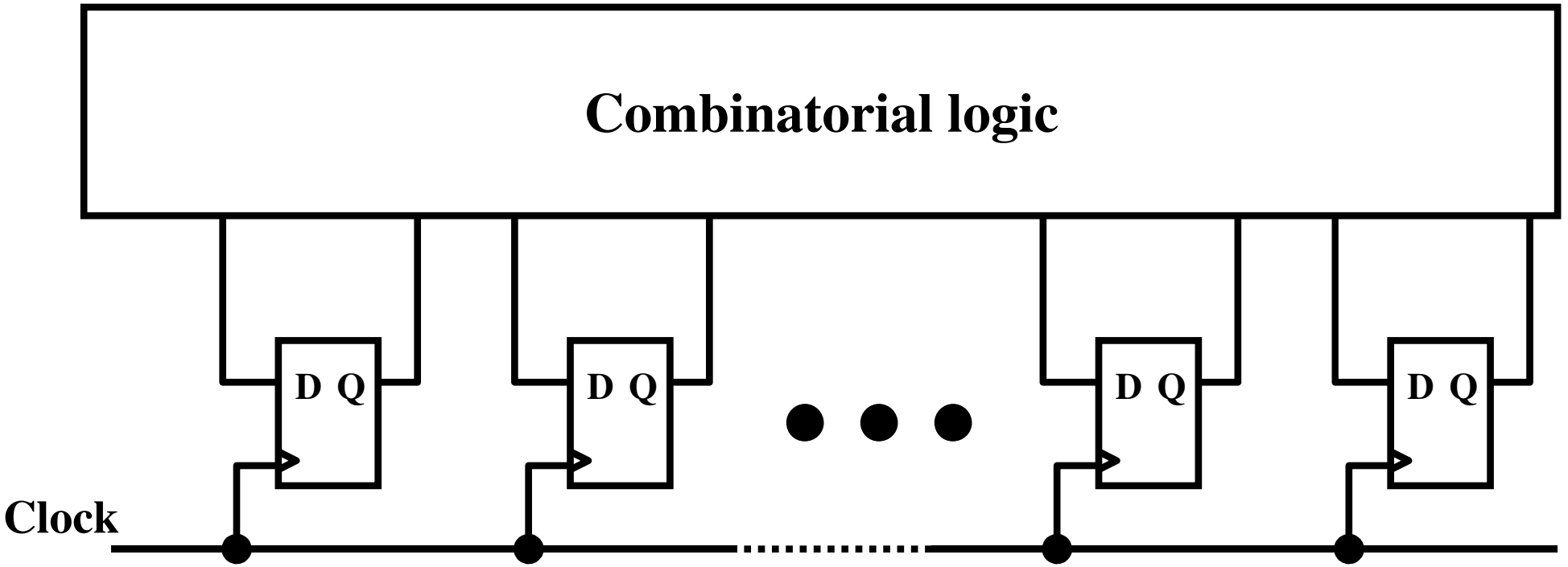
Distribution

Tests: need for DfT (Design for Testability)

- Test must be prepared at each design step
- Various techniques
  - Helping external testing with ATE
  - or performing (partial) self-test
- Classical approach: scan test (Boundary scan at board level)
  - Internal memory elements (flip-flops) connected in serial chain(s) or shift register(s) + I/Os if boundary scan
  - Simple external control signals for complete or partial memory controllability and observability
- Security constraints opposed to testability contraints !!
  - Scan chain = backdoor for hackers
  - Maximum (test) vs. minimum (security) controllability/observability

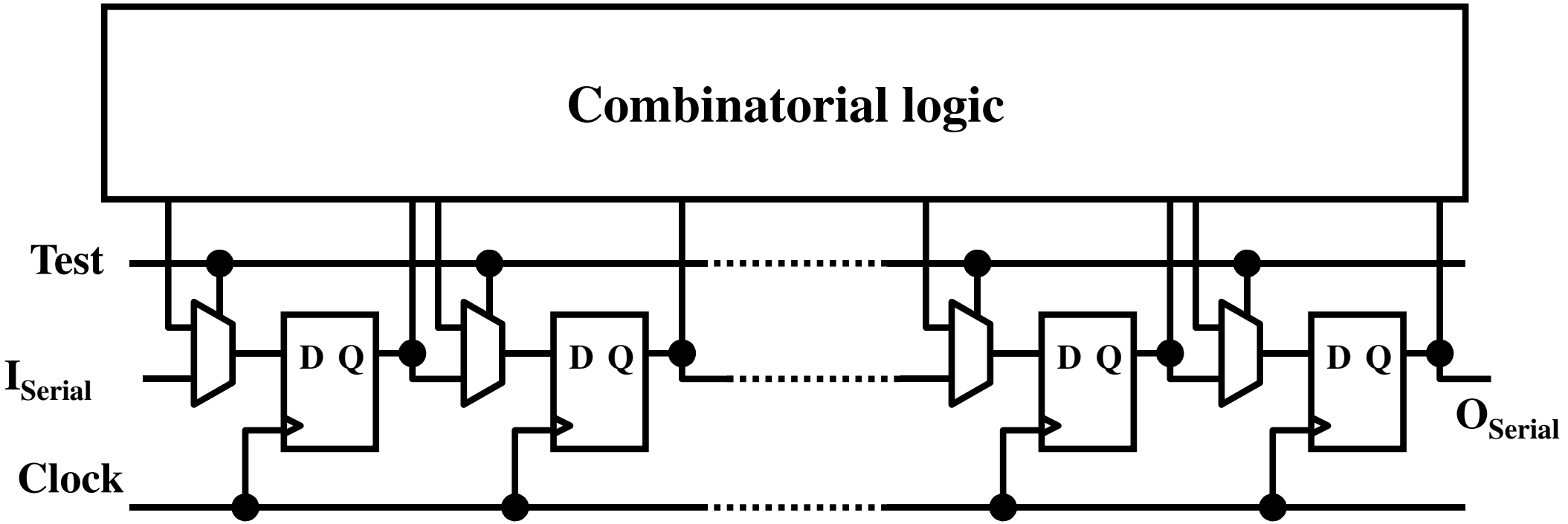
"Scanpath" implementation – basics (1)

Initial circuit:



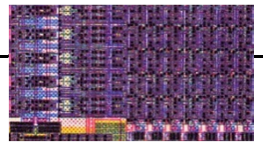
"Scanpath" implementation – basics (2)

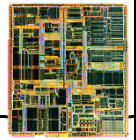
Circuit with a single scan chain:



Integrated circuits: styles

Design style	Full custom	Custom	Semi-custom	Programmable arrays
Specific Masks	All*	All*, after P&R (concatenation)	Interco.	None
Optimization	Maximum	Good: optimized cells, IPs, …	Medium	Medium
Problems/ risk	Maximum	Increasing with recent techno.	Medium (e.g. power lines)	Minimum
Pre-processing	No	No	Yes	On the shelf
Flexibility	Minimum	Low	Low	Maximum
Selected for	Critical blocks, basic cells	Large volumes, decreasing market	Smaller market	Quick grows
Remarks	Analog, MEMS, …		"Structured ASICs"	May be re-programmable





* 90 nm: 1 M$ per mask set … (40 to 50 M$ for a custom circuit design)

"Specific" Circuits

- Dedicated to a given application, or a small application category
  - ASIC: Application Specific Integrated Circuit
  - Opposed to "general usage" circuits, available "on the shelf" (COTS)
- Specificity possible at different levels
  - Specific manufacturing (mask-level configuration)
  - Generic manufacturing, user configuration
    => "programmable arrays": PLD, CPLD, FPGA
- Many architectural/design common notions
- Focus on Programmable Arrays in the sequel (due to increasing ASIC NRE costs + lab work feasibility), in spite of limitations

Main circuit design constraints …

- Area and yield
- Speed (clock frequency / computing power)
- Power/energy consumption and heat dissipation
- Pin number (off-chip interconnections)
- On-chip interconnections
- Testability
- Electromagnetic characteristics (emission, susceptibility)
- Robustness / dependability (including security)

- + Time-to-market, cost, …

Constraints: link with security ??

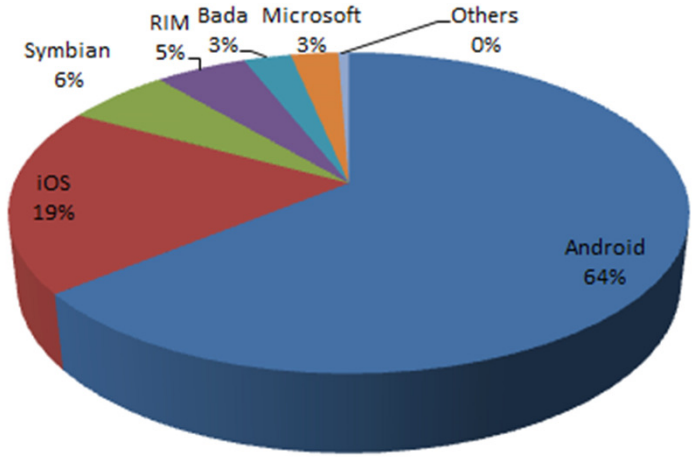
- Area and yield ? Yes, limits protections
- Speed (clock frequency / computing power) ? Yes, limits protections and/or increases susceptibility to faults
- Power/energy consumption and heat dissipation ?
  - Dynamic consumption: Yes, side channel
  - Temperature: Partially, temperature-induced faults
- Pin number (off-chip interconnections) ? Not directly
- On-chip interconnections ? Yes, in particular dummies
- Testability ? Yes, potential backdoor
- Electromagnetic characteristics (emission, susceptibility) ? Yes, side channel
- Robustness / dependability ? Yes, fault-based attack detection

Outline – Part I

- Embedded systems – general notions
  - Definition and applications
  - Typical constraints
  - Overview of implementation technologies
- Basic blocks
  - Processors
  - Communication resources
  - Programmable arrays

Basic blocks: hardware + software

- Hardware: focus of the sequel
- Operating systems (OS/RTOS/eRTOS)
  - Linux, µLinux
  - Windows
  - Android
  - iOS
  - …
- Support for application software: function libraries (e.g. generic signal processing or arithmetic functions)

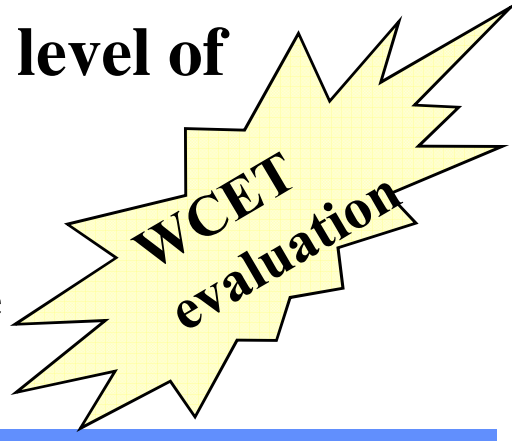


Smartphone share by operating system
Source: Gartner (Aug 2012)

Embedded OS

- Should be or have:
  - Modular
  - Scalable
  - Configurable
  - Small footprint
  - Wide CPU support
  - Large number of device drivers
  - …

- + real-time (ability of the OS to provide a required level of service in a bounded response time)
  - Soft real time: usual Worst-Case Execution Time
  - Hard real time: guaranteed Worst-Case Execution Time



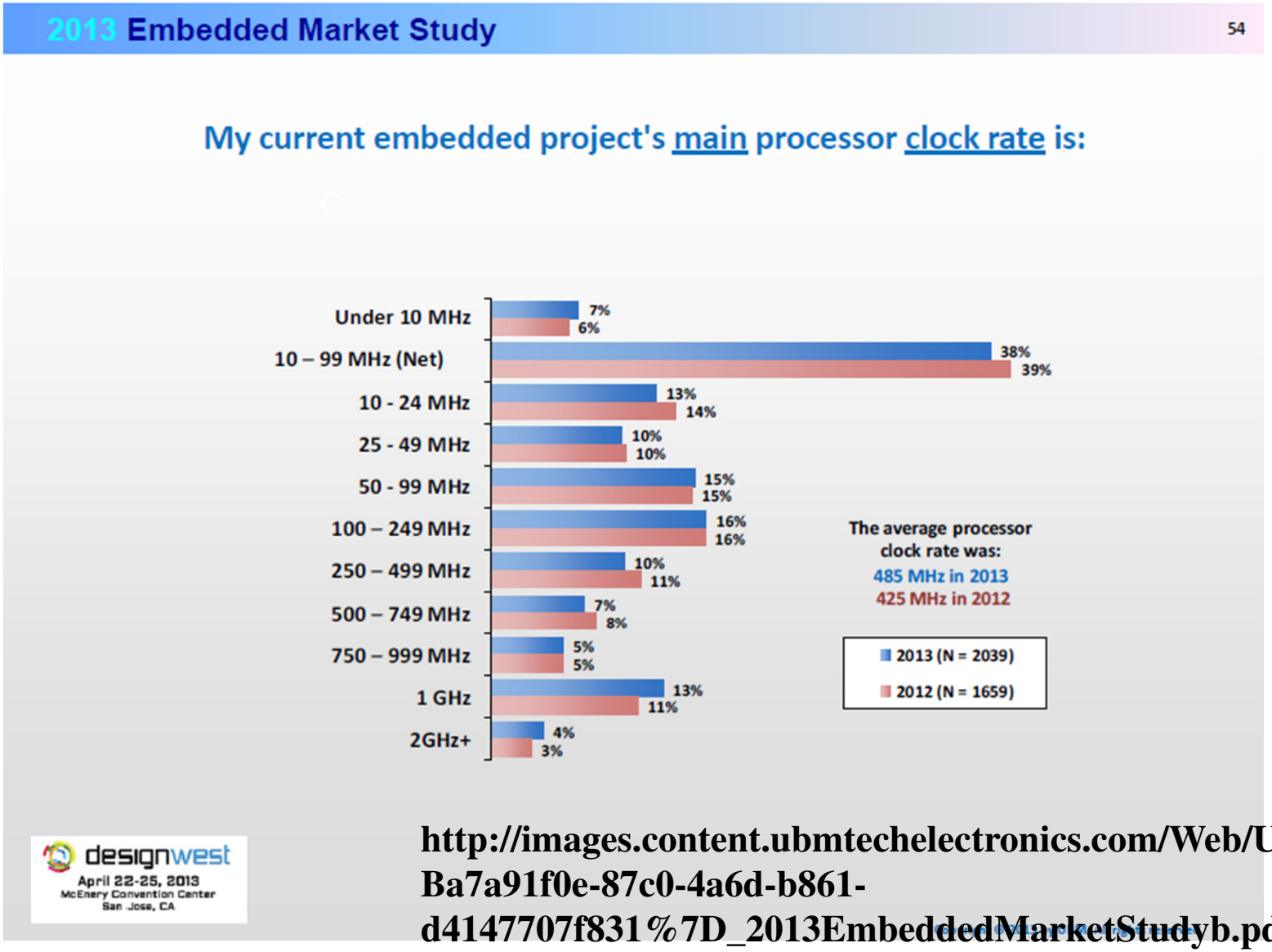
Main (hardware) cores

- Processors / DSPs
  - A brief section on microprocessors (MPUs) / microcontrollers (MCUs)
  - DSP specifics out of the scope of this course
- Other processing elements and digital peripherals
  - More specific (often one function, no software)
  - Many common micro-architecture optimizations (pipeline, etc.)
  - Crypto-processors addressed in Part II of this lecture
- Memories, analog peripherals
  - Details out of the scope of this course
- Communication resources - bus and system interconnections
  - Examples

Processor/MCU cores (global market)

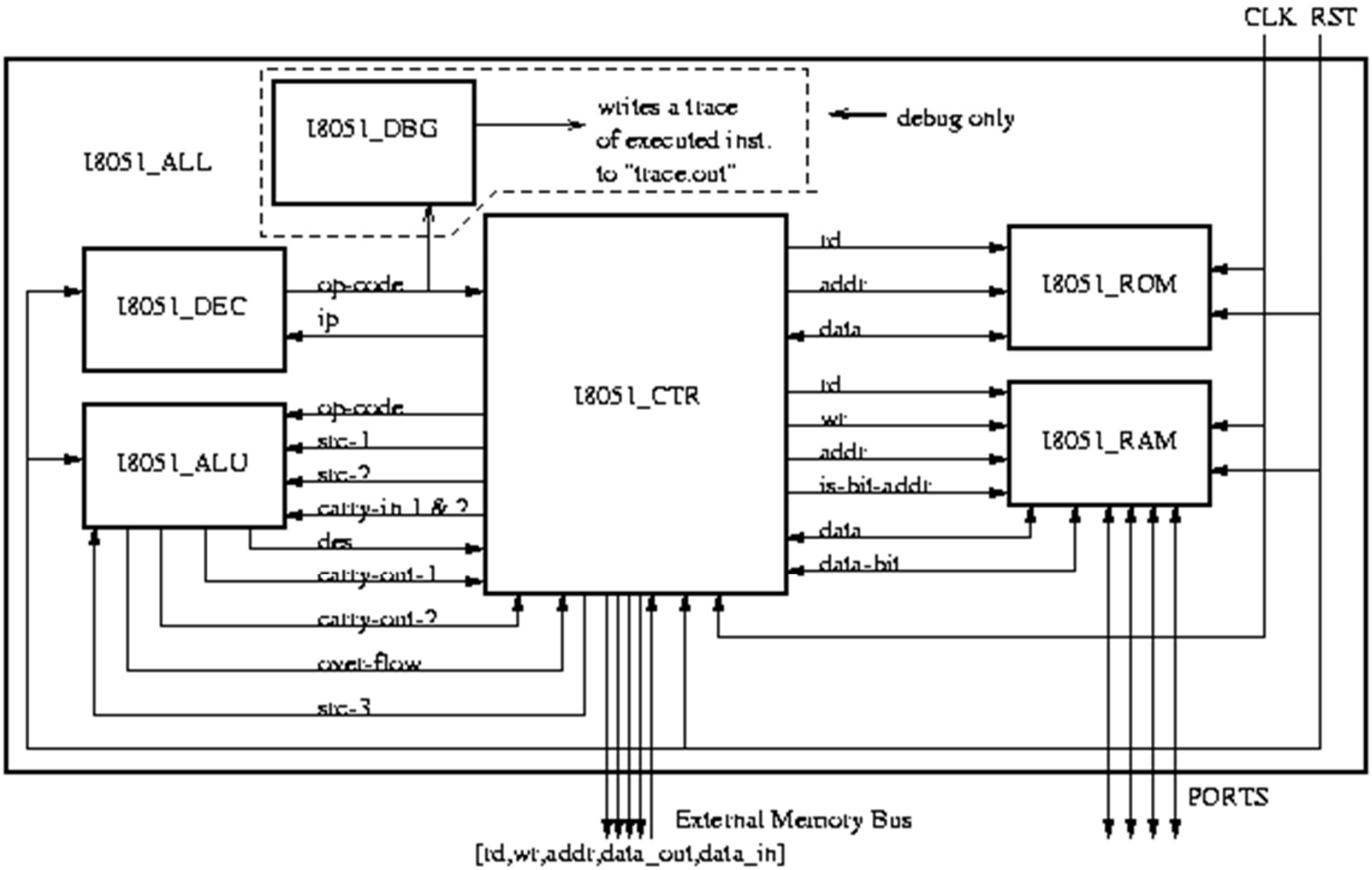
- Trading Area/Power vs. Performance
  - 8051 vs. PowerPC architecture - 8-bit vs. 64-bit - 100 MHz vs. 4 GHz
  - How many MIPS does a coffee maker or a toaster oven require ??
- Type of cores
  - Hard cores
  - Soft (synthesizable) cores => ASIC, FPGA …
  - Configurable cores (flexible ISA, specific hardware accelerators, …)
- I/Os and peripherals are at least as important
  - Memories (capacity, type)
  - Application related standards (e.g. CAN or FlexRay for automotive market)
  - External communications (e.g. RS232, PCI, …)
  - Watchdogs (or "counter-measures" for secure applications)
  - …

Clock frequency: 3 GHz or more??



http://images.content.ubmtechelectronics.com/Web/UBMTechElectronics/%7Ba7a91f0e-87c0-4a6d-b861-d4147707f831%7D_2013EmbeddedMarketStudyb.pdf

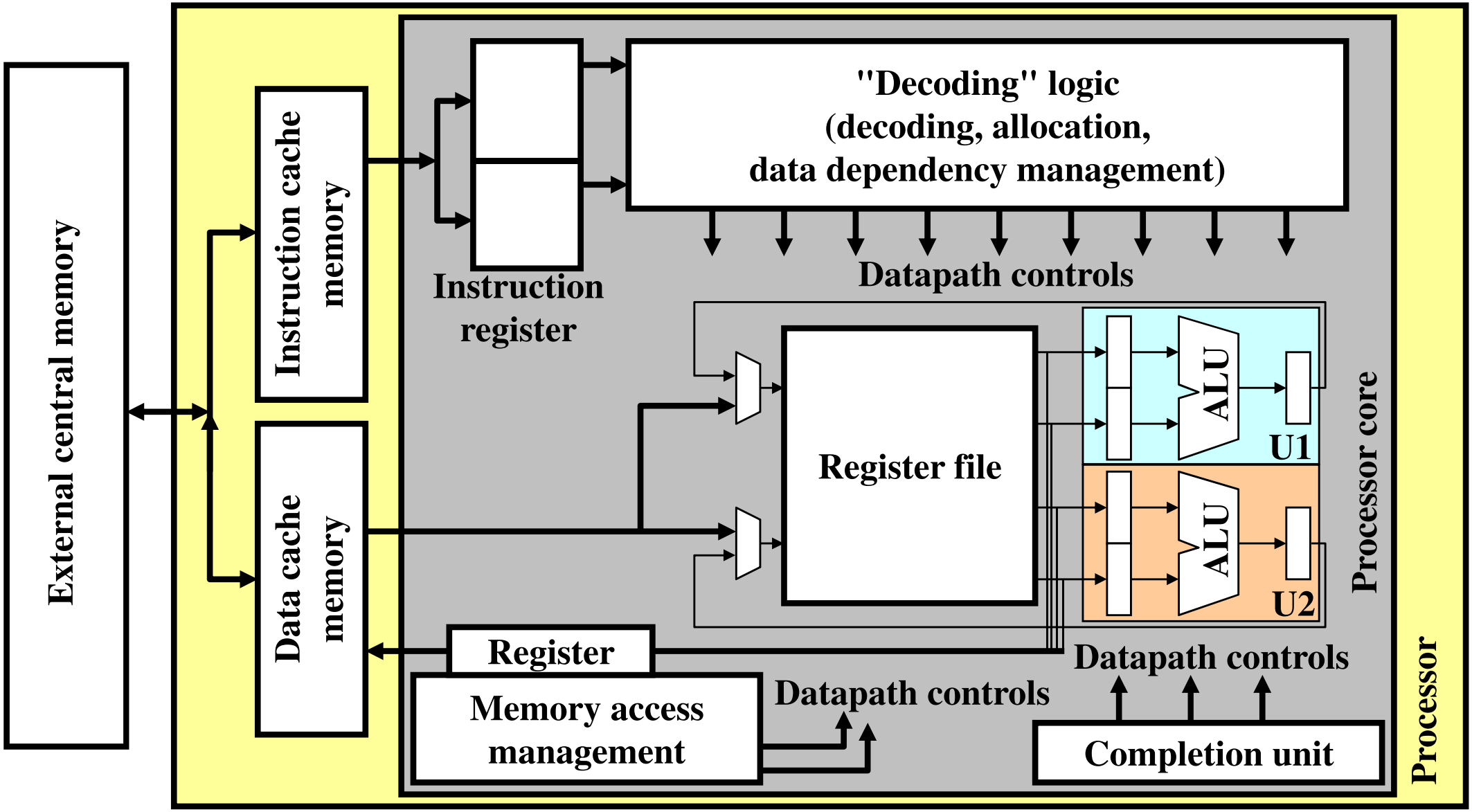
8051 microcontroller



VHDL RTL models

(http://www.cs.ucr.edu/~dalton/i8051 or http://www.8051.free.fr/)

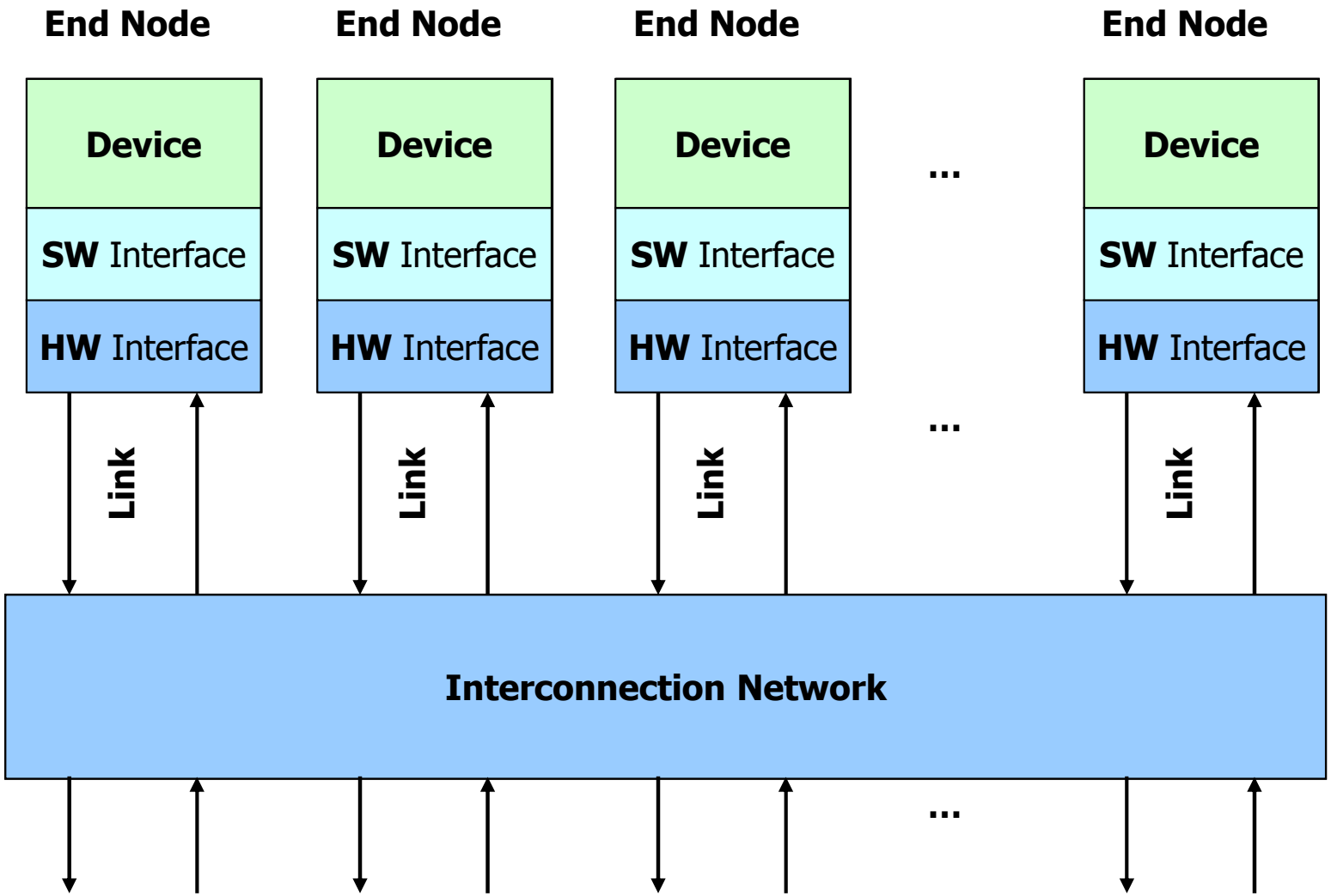
Elementary superscalar processor



Micro-architectural optimizations

- Out-of-order execution, not deterministic
- Predictions (speculative executions)
- Example: branch prediction => performance increase, but potential security threats (exploited by attacks)
- Cache memories also concerned

Interconnection = hardware + software



Main SoC interconnection resources

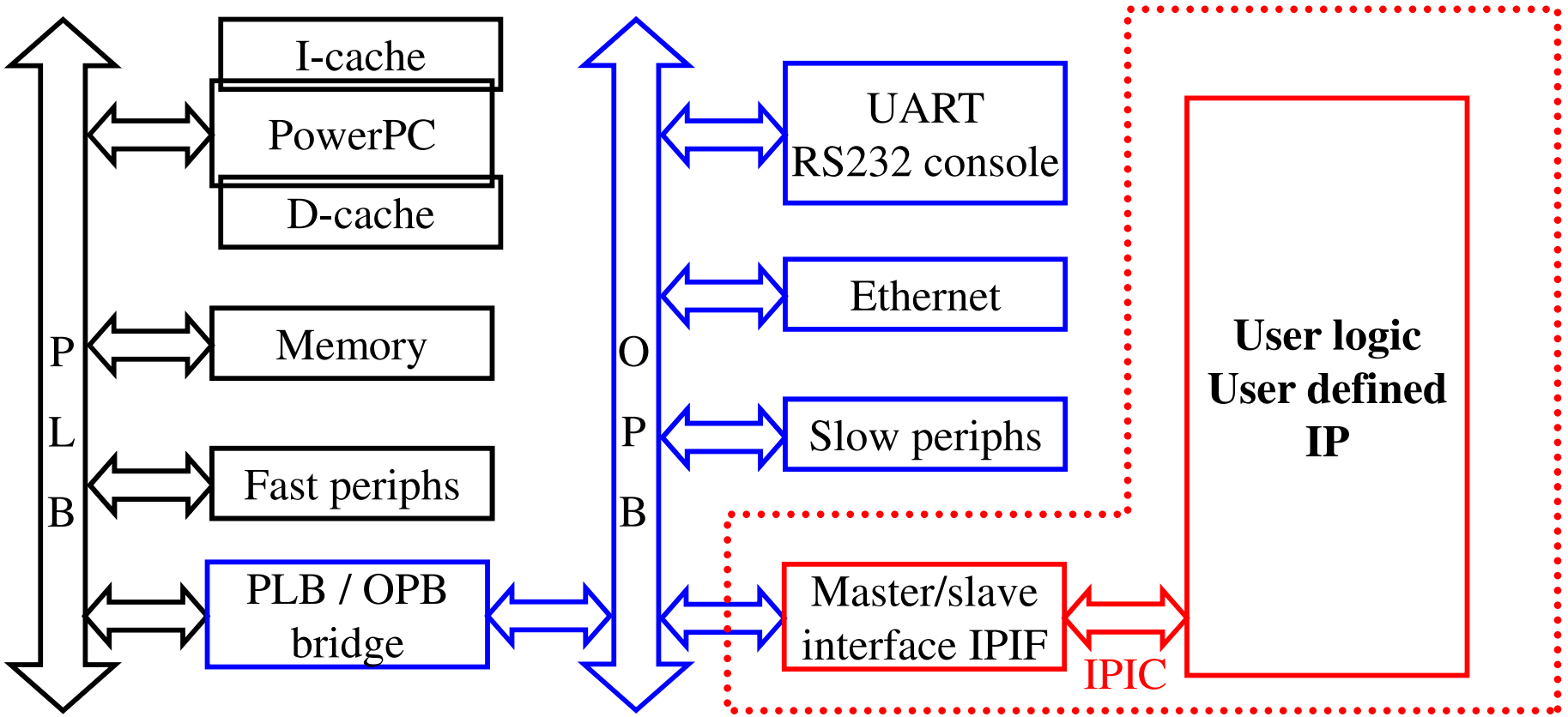
- System buses
  - AMBA (ARM)
  - CoreConnect (IBM)
  - STBus (ST)
  - Avalon (Altera - SOPCBuilder)
  - AXI
  - …

Characteristics:
- complex protocols, several masters
- high performance goals (burst transfers)
- not limited by pin count (internal buses)
- simplified protocol for low performance transfers

- From buses to NoCs …
  … or from interconnection to communication !
  - Increased scalability and hierarchical composition needs (complexity of SoCs: number of cores/processors)
  - Increased flexibility needs (dynamic transaction definitions, data-driven algorithms)
  - Compatible with future technologies (noise, interferences, IR drop, variability, soft errors …) + multiple clock domains and GALS (globally asynchronous, locally synchronous) schemes
  - Technology improvements: more transistors available for interconnection/communication resources !

=> On-chip packet switching networks, or NoCs

CoreConnect (IBM)



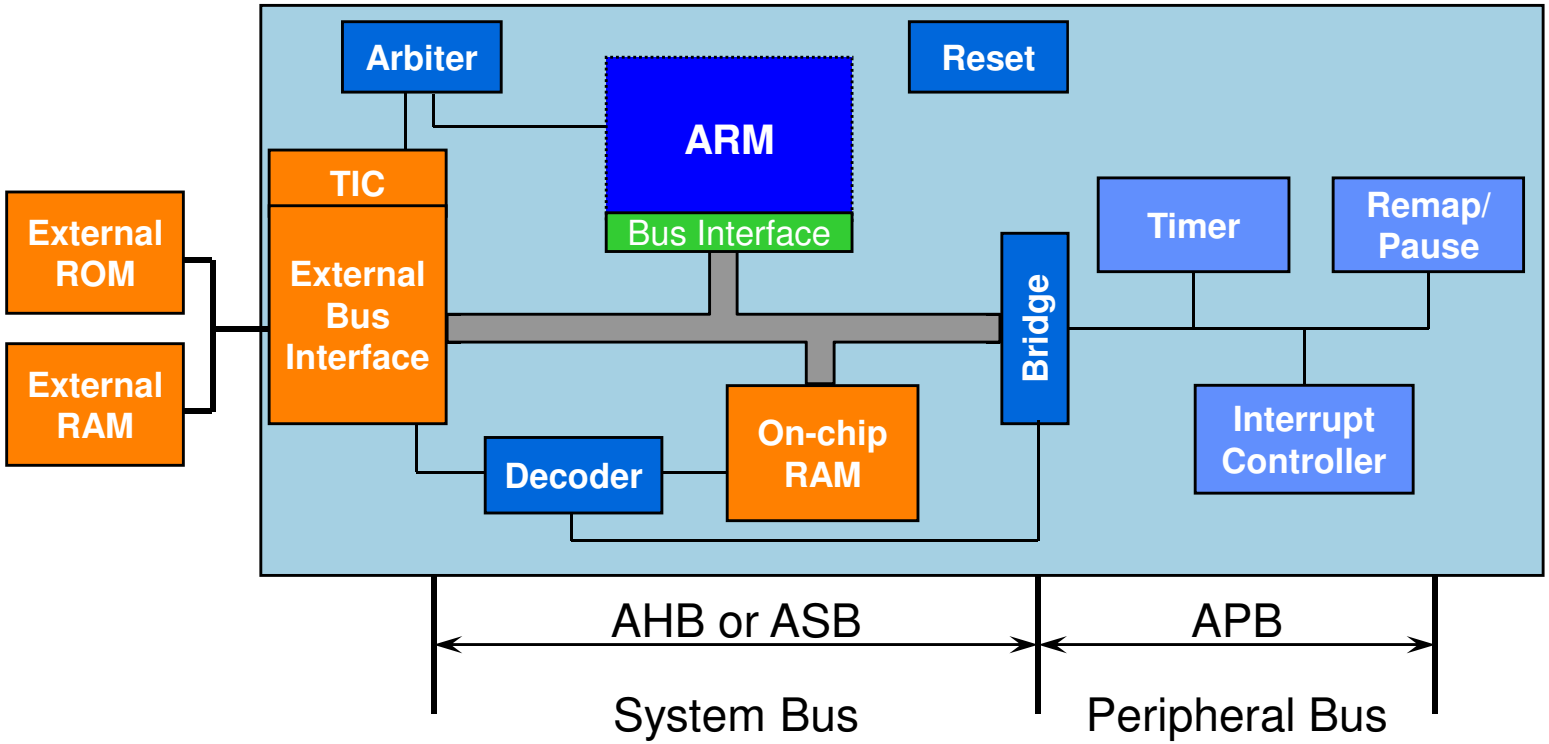
- PLB: high performance synchronous bus designed for connection of processors to high-performance peripheral devices
- OPB: general-purpose synchronous bus for on-chip slower peripherals
- User cores: attached to one of the embedded processor buses (several possible configurations, in general connected to OPB)

IPIF – IP interface

IPIC – IP interconnect

} Xilinx

AMBA: Advanced Microcontroller Bus Architecture

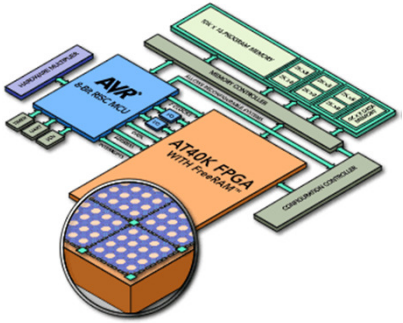


- **AHB: Advanced High-performance Bus (introduced to provide improved support for high performance, synthesis and timing verification)**
- **ASB: Advanced System Bus (older form)**
- **APB: Advanced Peripheral Bus (Slaves / low performance)**

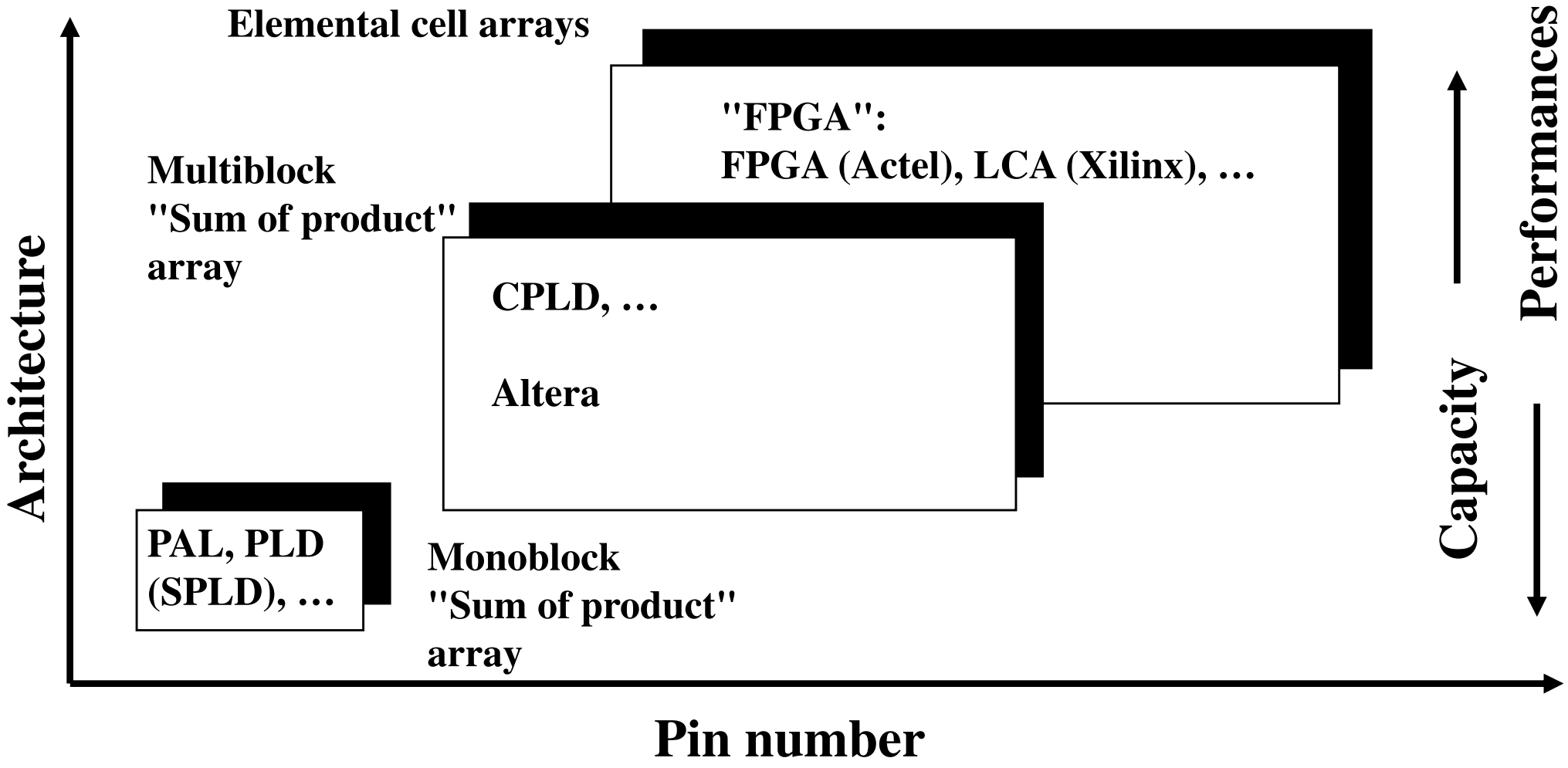
Programmable arrays

- A few thousands … (www.ednmag.com)

- Several classification criteria
  - Elementary cell architecture type
    - AND/OR arrays: sum of products => PLD
    - Elemental functions (Mux-based or LUT-based) => FPGA
  - Logic complexity
    - SPLD (one block: PAL, PLA, PLD, …) vs. CPLD (multiblocks)
    - FPGA (elemental cell array) vs. SoPC or "FPSLIC" (FPGA+optimized hard processor core)
  - Programmation technology
    - Fuses / anti-fuses
    - PROM, EPROM, EEPROM (PLD, EPLD, EEPLD, …)
    - SRAM, Flash



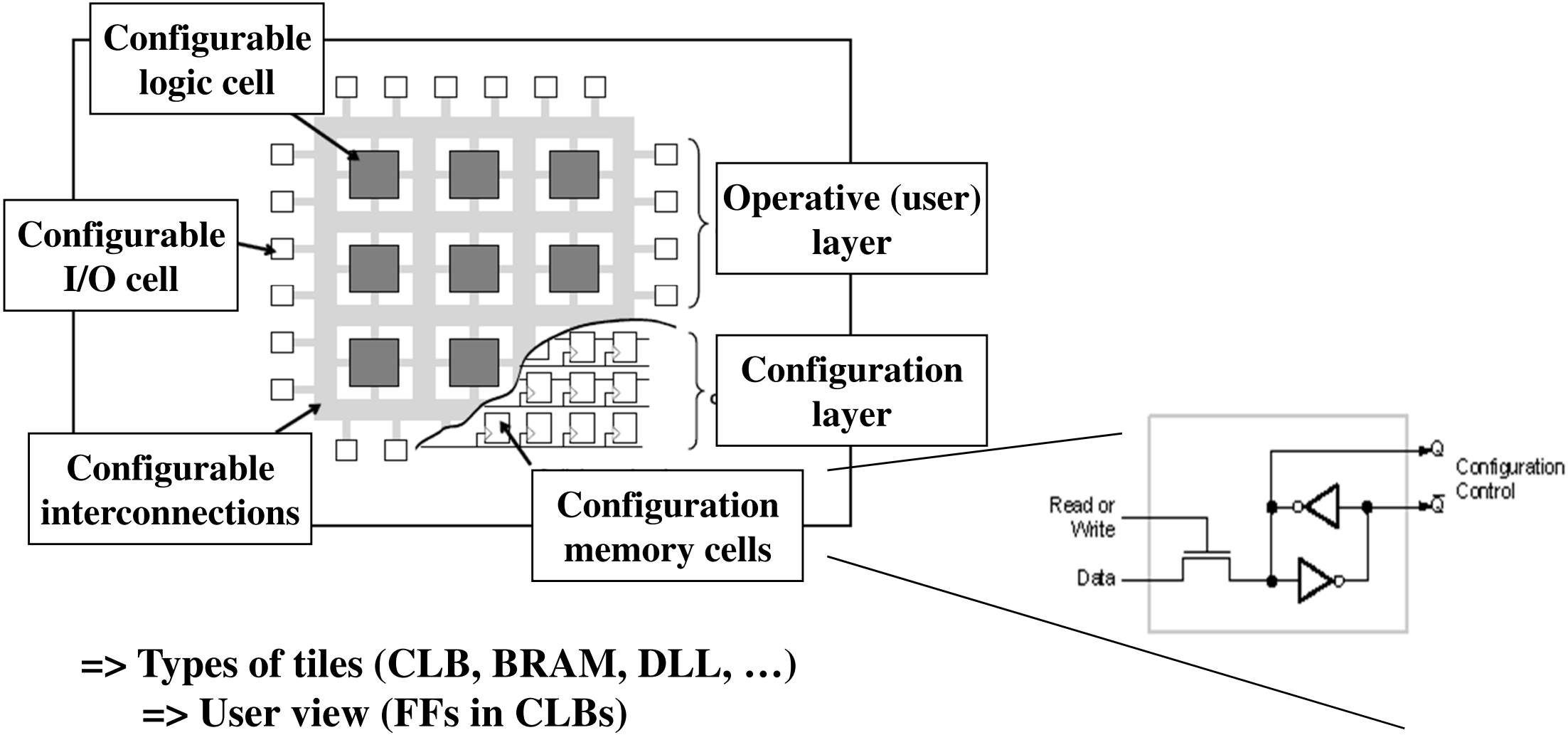
Complexity and performances of architectures



Main characteristics of a device

- Digital, analog or mixte …
- In most cases: digital
  - Number of User I/Os
  - Number of flip-flops (+ memory blocks)
  - Array capacity (equivalent gates) and usage rate (depends on P&R and implemented function => influence of the basic architecture, based on sum of products or elemental cell)
  - Predictive evaluation and control of critical paths
  - Programmation type (retention, confidentiality, programmation speed, total or partial re-configurability)
- Security: SRAM-based is a concern
  - With respect to cloning (=> encrypted bitstreams)
  - With respect to configuration errors (modified function and/or structure)

Generic organization of a SRAM-based FPGA



=> Types of tiles (CLB, BRAM, DLL, …)

=> User view (FFs in CLBs)

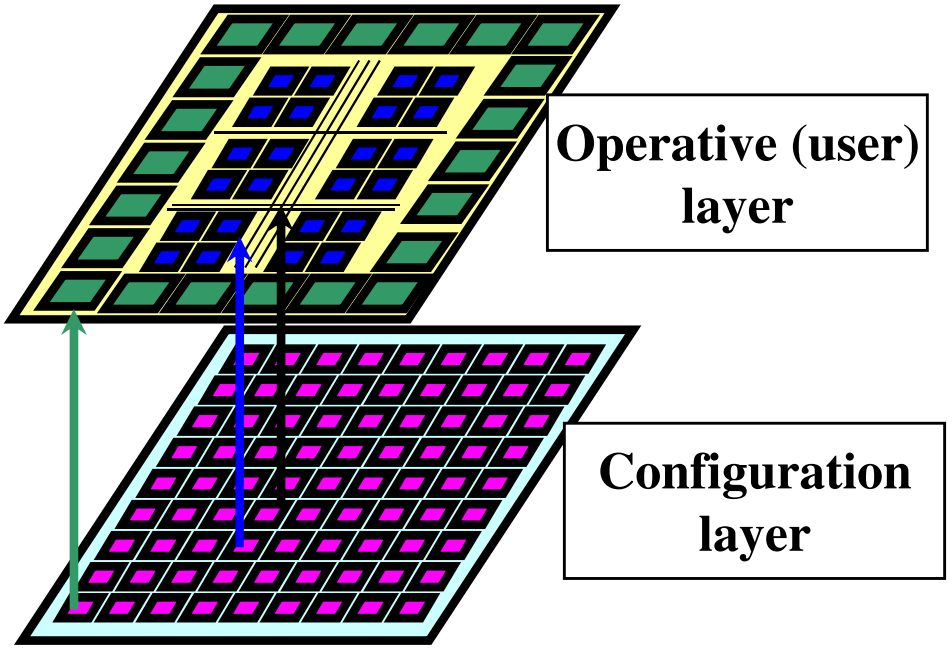
=> Configuration view (configuration memory)

=> May also include hardwired processors, multipliers, …

FPGAs vs. ASICs

- **Permanent configuration: basically equivalent to ASICs (user layer only)**

- **SRAM-based configuration: huge number of memory cells, errors in the configuration layer cannot be neglected**



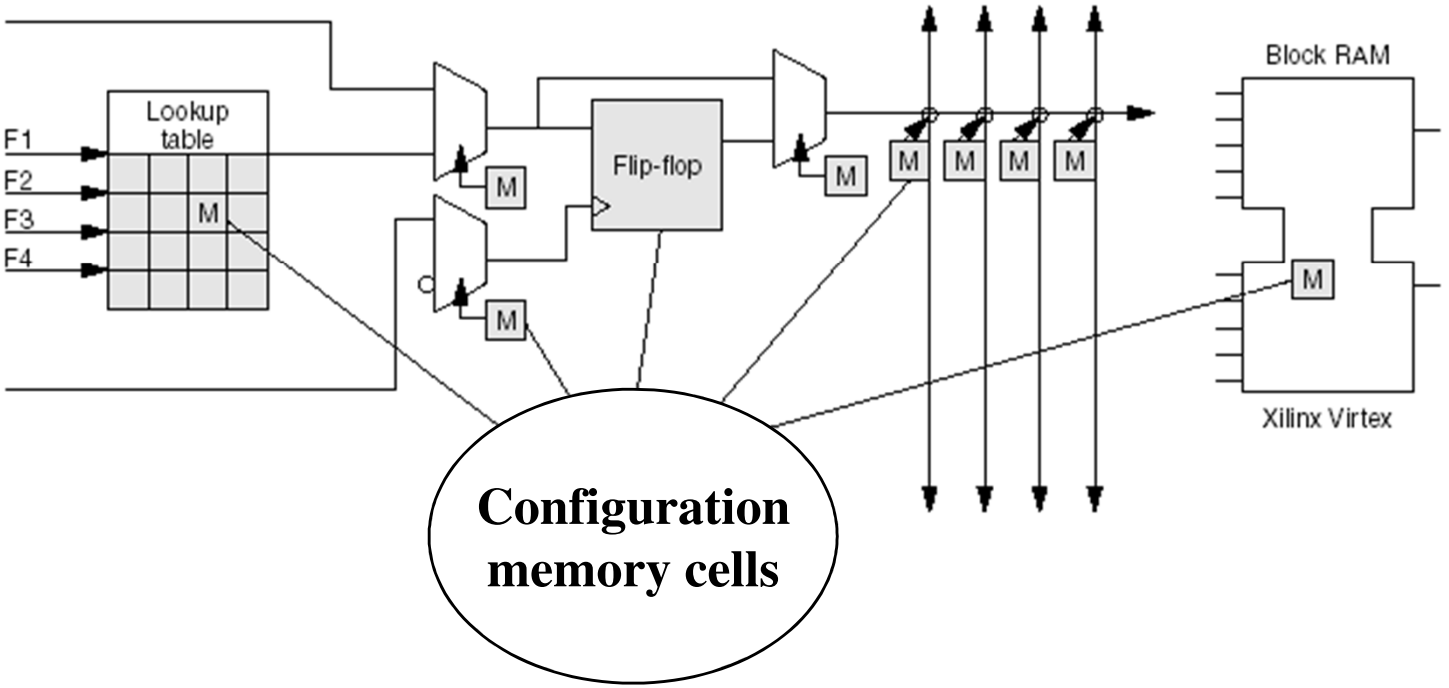
I/O blocks

Logic blocks

Interconnections and switches

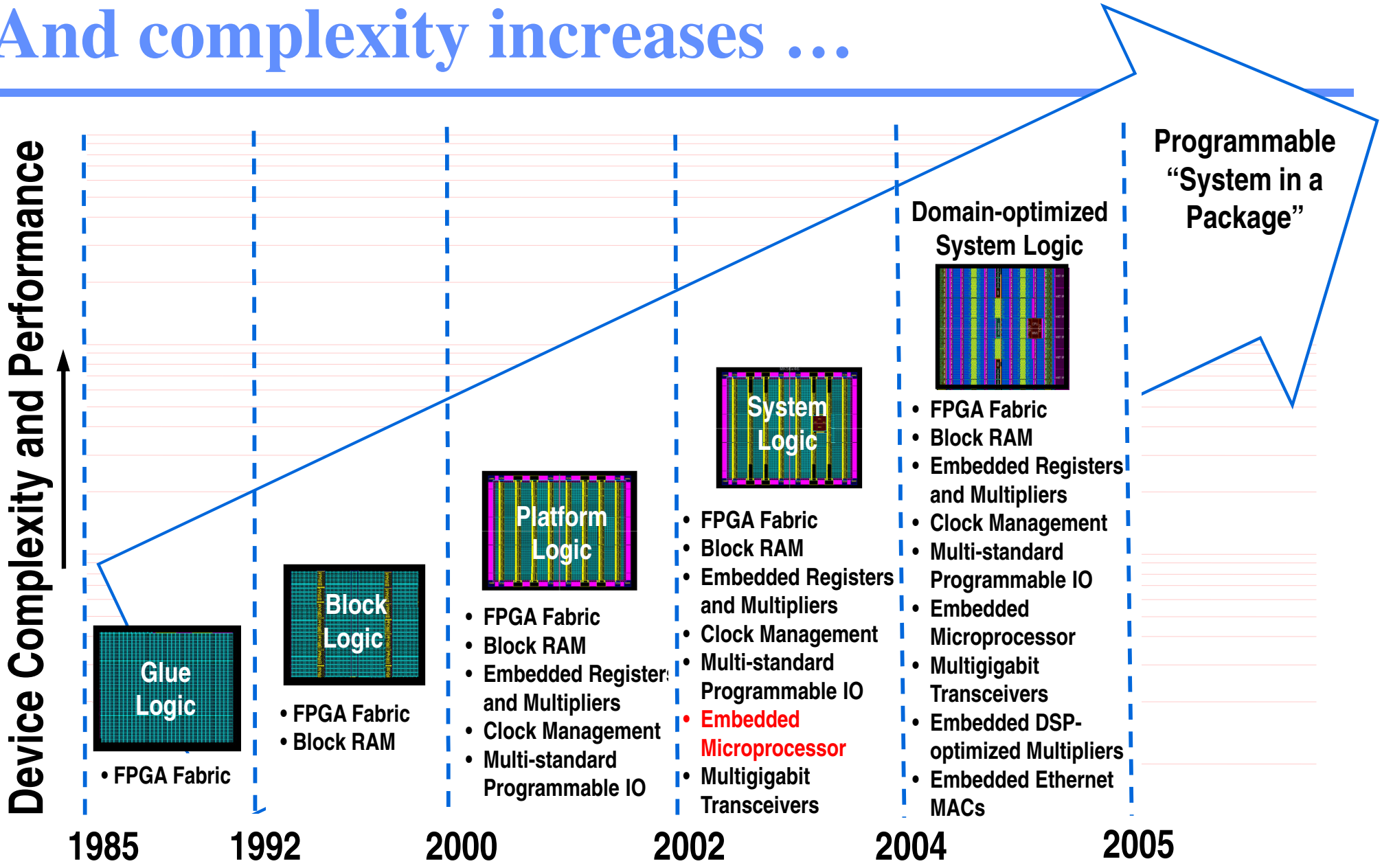
Configuration memory points

SRAM-based FPGA CLB: simplified view



From: F. Gusmao de Lima Kastensmidt, G. Neuberger, R. F. Hentschke, L. Carro, R. Reis
Designing fault tolerant techniques for SRAM-based FPGAs
IEEE Design & Test of Computers, vol. 21, no. 6, November-Décember 2004, pp. 552-562

And complexity increases …



From http://klabs.org/mapld05/presento/245_padovani_p.ppt

Part II

Secure circuits: qualification, common criteria

Security definition

- Security = one attribute of dependability
- Security is the concurrent existence of several pillars
  - Availability (but for authorized users only => Confidentiality),
  - Integrity = absence of improper system alterations; with "improper" meaning "unauthorized",
  - Authenticity (information comes from trusted source)
  - Resilience (or robustness w.r.t. attacks against previous pillars + limiting the potential damage in case of successful attack)
- Often linked to the use of cryptography (or steganography) => most examples will be given on crypto-processors or cryptographic accelerators
- System-level security policies will not be discussed here

Chip level security: Why ? Because security is:

- … An increasing (and already major) societal and economics concern in the "information society",
  - Mastering some kind of information is now the key in most situations
  - Need for availability, integrity, authenticity of services
  - Ensuring privacy is required in spite of all the information flows
  - …
- … More and more deeply inserted at the heart of many of our everyday-life applications,
- … Implemented by "embedded systems", that are in most cases integrated systems (with on-chip cryptography),
- … Thus very dependent on potential flaws at the chip level.

Tampering secure circuits

- A secure circuit contains secret data (e.g. a secret cryptographic key)
- Knowing the secret grants unauthorized privileges (access, message interception, …)
- May allow in some cases to clone a device (e.g. pay-per-view TV decoder … or a bus/subway card …)

Security flaws at chip level ??

- Yes, many examples ! Not only Internet viruses or big infrastructure weaknesses …

- A recent one ?
  "T card has security flaw, says researcher", by Hiawatha Bray
  The Boston Globe (boston.com), March 6, 2008

  - A computer science student at the University of Virginia asserts that he has found a security flaw in the technology behind the Massachusetts Bay Transportation Authority's CharlieCard system (based on the MiFare Classic RFID chip by NXP Semiconductors). The company spent $192 million to introduce the CharlieCard in 2006. The system replaced cash and tokens.
  - Such a breakthrough could be used to make counterfeit copies of the cards (sold in the underworld), allowing commuters to pay for bus and subway rides.

  - It is also the chip used in London's subway system.

This should be very complex and expensive !!

Complex ?

- "A press release issued by the University of Virginia said the research team obtained the same kind of chip, then used abrasives to scrape away the chip layer by layer. By examining the chip circuitry, they were able to figure out the encryption algorithm it uses and found weaknesses that made it easy to break. Next, the team was able to use commercially available RFID readers to capture data from any RFID-equipped cards that came within range. They could then decrypt the data on those cards and copy them."
- Not for everybody … but far from impossible …

Expensive ?

- "Nohl said that his team needed only about $1,000 worth of equipment to dismantle the chip and crack the code."
- Techniques are in some cases still cheaper …

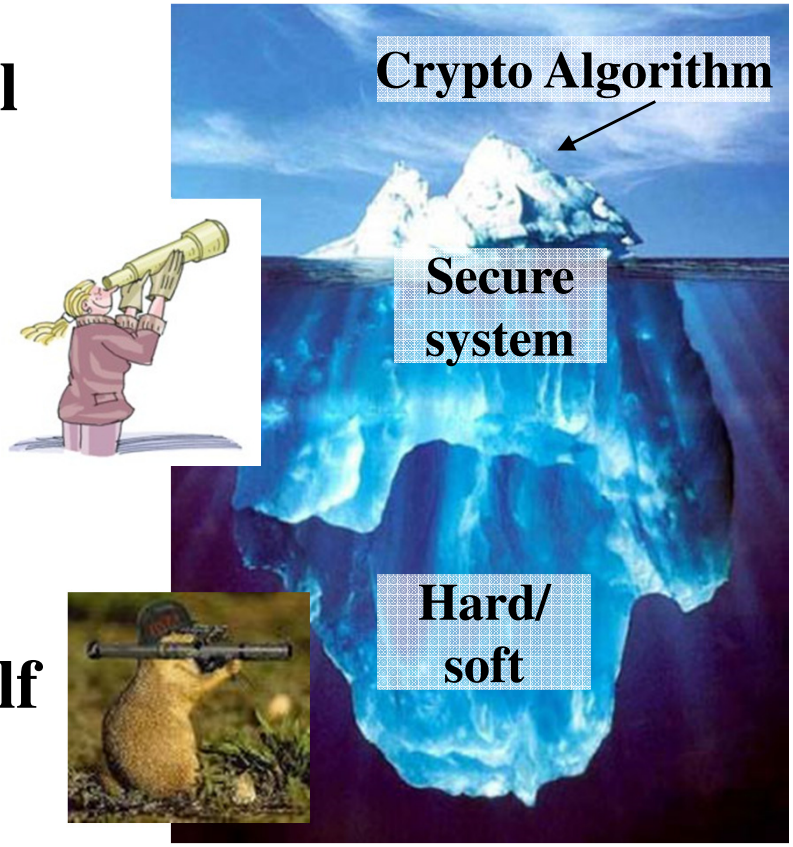
Few comments on this case

- Several security layers => One breach does not imply the full system failure
- Risk of (large) financial losses in this case for the provider – Many similar cases (cash from ATMs, pay-per-view TV access, …)
- Consequences may be more dangerous in other cases (unauthorized access by a terrorist or a spy in a critical area …)
- Chip-level security is a concern !
  - Attacks – need to understand possible chip-level weaknesses (even without design error)
  - Counter-measures – need to develop protections w.r.t. those attacks

Attacks vs. classical cryptanalysis

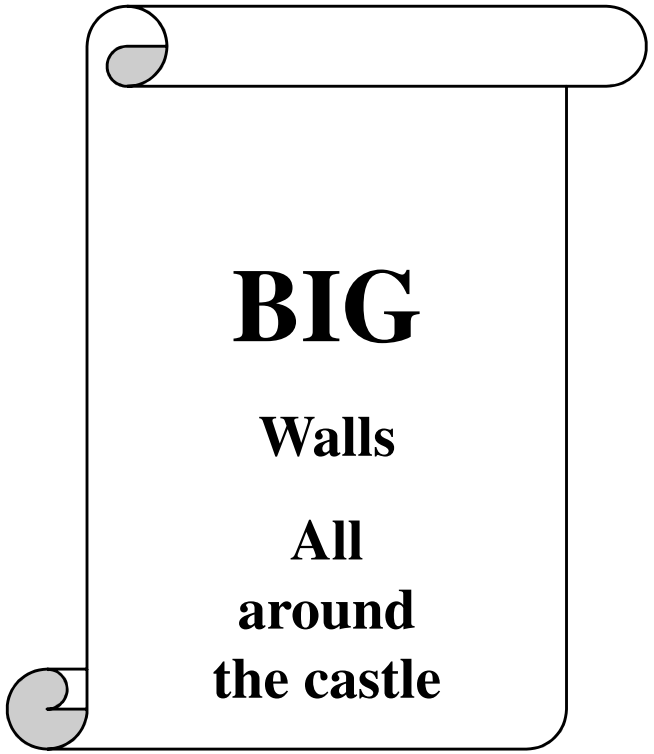
- Classical cryptanalysis mainly aims at finding
  - Algorithm flaws (mathematical analysis)
  - Practical limits in exhaustive search (brute-force attacks)

- Careful study of the mathematical model describing a cryptosystem, yielding properties revealing information about the secret
- General, sometimes very theoretical



- Attacks target the implementation characteristics, not the algorithm by itself
- Specific, much more practical

Algorithm vs. implementation



Algorithm **Implementation**

Hardware-level security is at the heart of system security

Secure product validation and qualification

- Several steps
  - Design time (simulations, proofs, …)
  - Final product samples (official qualification – national certified laboratories or evaluation centers)
- Multi-aspect evaluation
  - Product robustness – attacks
  - Source code analysis (hardware and software)
  - Design/development process (methods, tools, controlled environment, …)
- International reference: Common Criteria (among other standards)
  - With respect to specified threats (not an absolute qualification)
  - White box evaluation: not representative of real hacker attacks

Common Criteria (CC)

- ISO 15408
  - Not a security framework
  - Framework for specification of evaluation
- Target Of Evaluation (TOE)
  - The product or system that is the subject of the evaluation of the security features
- Protection Profile (PP)
  - A document identifying security requirements for a class of security devices
- Security Target (ST)
  - The document identifying the security properties of the TOE
  - Usually published, so that potential customers may determine the specific security features that have been certified

Product Evaluation

- Aims at evaluating:
  - Product conformance (w.r.t. claims/technical datasheet)
  - Robustness of implemented mechanisms
  - Confidence in the development process
- Oriented to purchaser/user of system
- Assurance that system operates as advertised
- Three top level documents
  - Common Criteria Documents - describes requirements, defines EALs
  - CC Evaluation Methodology (CEM) - details on the evaluation
  - Evaluation (certification) Scheme - national specific rules for how CC evaluations are performed (NIST in US, DCSSI in France)
- Product evaluated with respect to
  - Security Target
  - A given type of attacker (individual vs. governmental department)

Attack levels (Robustness levels)

- CC version 3.1 : 5 levels
  - Public vulnerability (AVA_VAN.1)
  - Elementary attack (AVA_VAN.2)
  - Re-enforced elementary attack (AVA_VAN.3)
  - Medium attack level (AVA_VAN.4)
  - High attack level (AVA_VAN.5)
- Assumptions about the type of attacker (knowledge, equipment)

Evaluation grades

- Estimation of:
  - Required equipment
  - Required experimental duration
  - Required knowledge on the product (e.g. black vs. white box attack)
  - Required expertise
  - Required opportunity
- Sum of grades => intervals compatible with the 5 robustness levels

Evaluation Assurance Levels (2016)

69	EAL1 – Functionally Tested
309	EAL2 – Structurally Tested
337	EAL3 – Methodically Tested and Checked
784	EAL4 – Methodically Designed, Tested, and Reviewed
431	EAL5 – Semiformally Designed and Tested
42	EAL6 – Semiformally Verified Design and Tested
6	EAL7 – Formally Verified Design and Tested

High levels

Evaluation limitations …

- Few products evaluated above EAL4 (about 50 EAL5 worldwide in Q1 2008, mainly smartcards – many others on-going – and 2 products qualified EAL6, EAL7)
- Most common products not evaluated (especially for software products)
  - Evaluation takes a long time (even with incremental process)
  - Modern patch/upgrade process hardly compatible with evaluation
  - E.g. Windows versions
    - Do you use EAL4 certified "Microsoft Windows XP, Professional; SP 2 (hotfixes 896423, 899587, 899588, 896422, 890859, 873333, 885250, 888302, 885835, and 907865)"
    - Or a more recent release, with fewer known vulnerabilities ?
- Similar problem increasing for hardware products (reconfigurable chips)

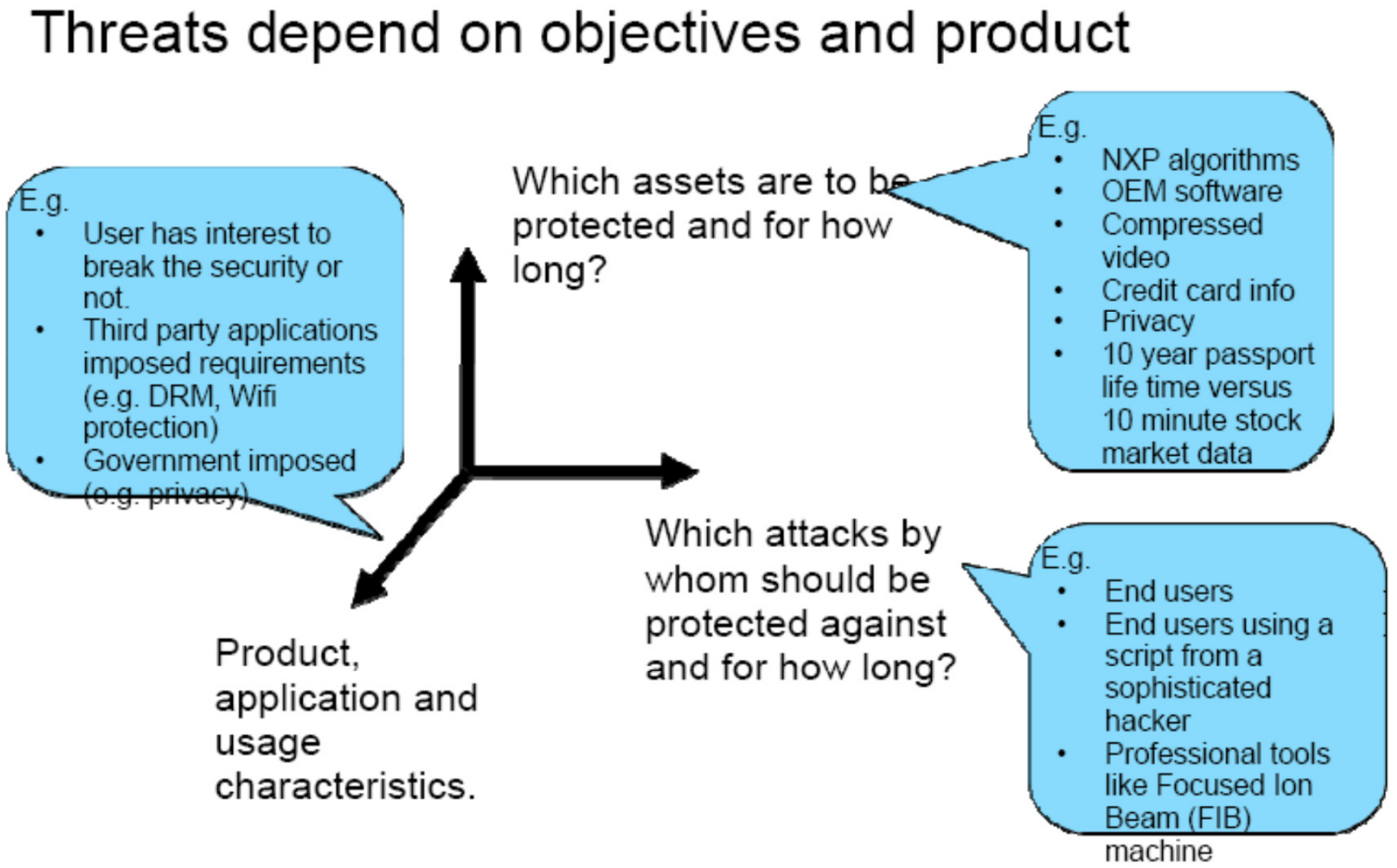
Case of SoCs and consumer market

- Increasing need for security in (complex) consumer devices
- Historically few concerns about security (open systems) but
  - Automotive
  - Set-top-boxes, mobile (smart)phones
  - Digital camera, digital TV, multimedia, DVD player, game stations
  - …
- Dedicated security modules are often not an adequate solution
- Need for SoC (and MPSoC) robustness against HW & SW attacks

[Security challenges in complex SoCs
MINATEC Security WG, June 4th 2010
B. Kasser, Advanced System Technology - Security R&D - STMicroelectronics]

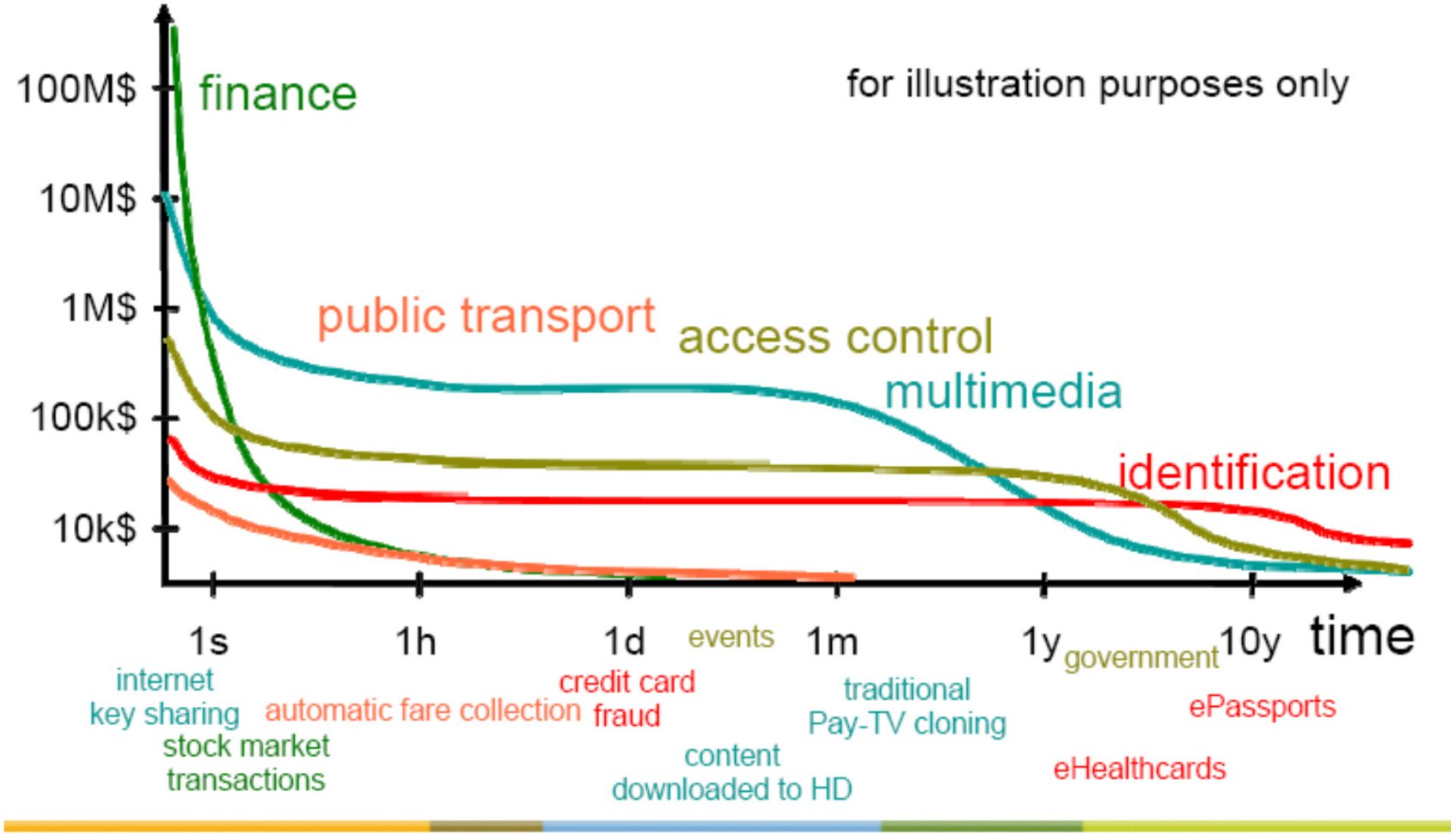
Risk analysis depends on product

"A system is vulnerable if the effort to break is less than the expected gain"



Hugues de Perthuis, Presentation to the GdR SoC-SiP Security working day, Nov. 16, 2009

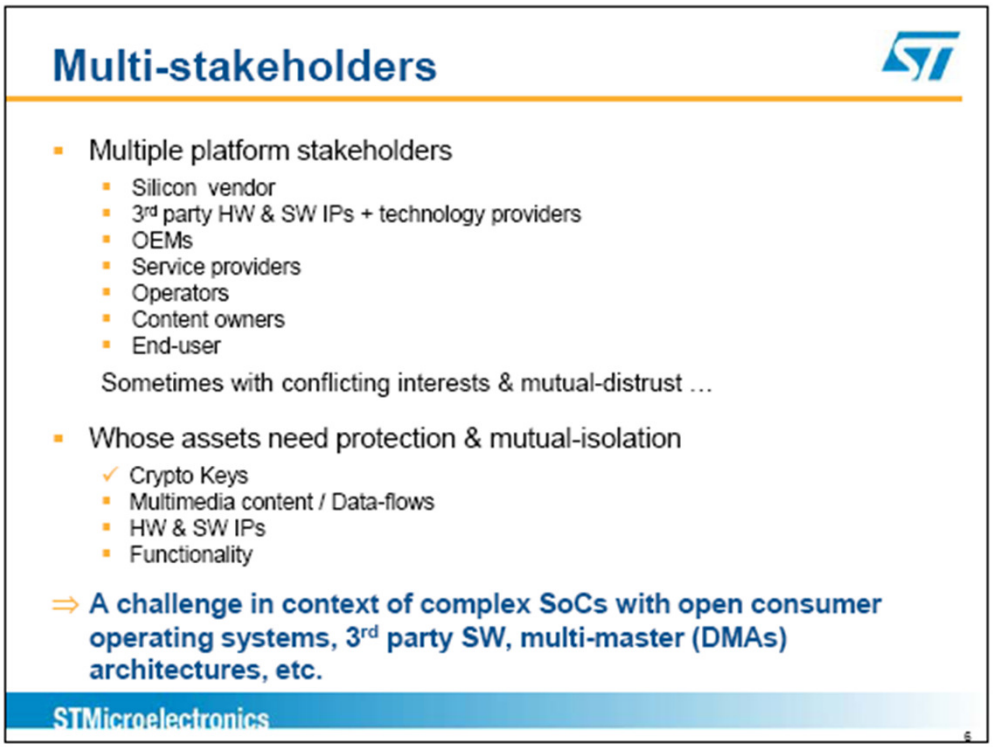
Attack value vs. time and product



Hugues de Perthuis, Presentation to the GdR SoC-SiP Security working day, Nov. 16, 2009

Many security challenges (at "low" cost)

- **SW integrity challenges**
  - **Avoiding PC-like threats (Virus, Trojans, DoS, etc.) exploiting SW vulnerabilities in connected embedded systems => doing better than in the PC world !**
  - **Digital Rights Management (HW & SW IPs)**
- **HW attacks**
  - **Finding best split between system-level countermeasures, dedicated security ICs, and SoC-level security measures**
  - **Minimizing impact on SoC design&verification flow + right SoC security cost / performance trade-off for given applications' robustness needs**



[Security challenges in complex SoCs MINATEC Security WG, June 4th 2010 B. Kasser, Advanced System Technology Security R&D – STMicroelectronics]

Goals depend on attacker profile + stakeholder

- **Professional** — Attacker
  - Content stealing
  - IP reverse-engineering
  - Device cloning /unlicensed usage
  - Reputation damage
- **Hacker/academia**
  - Technical challenge
  - Peer recognition
- **User**
  - Feature addition/upgrade
  - Alternative usage
  - Content abuse

Stakeholder

Who is interested in security features ?

e.g. DVD player:

Manufacturer ? **No !**

User ? **NO !**

DVD editor ? **YES !**

=> Risks of User attack

Hugues de Perthuis, Presentation to the GdR SoC-SiP Security working day, Nov. 16, 2009

But consumer MPSoCs are not smartcards!

- Embedded security research traditionally driven by smartcard world
  - Extremely cost sensitive
  - Low/medium complexity ICs
  - Few IOs
  - Not in most aggressive (45/32/22nm) CMOS technologies (eNVM + not economical)
- Complex SoC have different constraints
  - 45 / 32 => 22nm
    - Gate-count becoming irrelevant
    - Advanced power management techniques
    - New attack paths / security challenges ?
  - Packaging-level security could become practical / **economical**

⇒ **Need & opportunity to think wider …**

Security evaluation & certification

- CC certification usually not used in SoCs:
  - Too complex / costly / impacting
- Proving & comparing => selling security robustness can be a challenge
  - Compliance to robustness rules vs. having to demonstrate actual robustness achievements
  - Selling good/better security robustness can be challenge in increasingly horizontal contexts
- **An applicable, lighter weight security evaluation / certification process remains to be found ?**

STMicroelectronics

10

**[Security challenges in complex SoCs
MINATEC Security WG, June 4th 2010
B. Kasser, Advanced System Technology
Security R&D – STMicroelectronics]**