Embedded Systems: Introduction

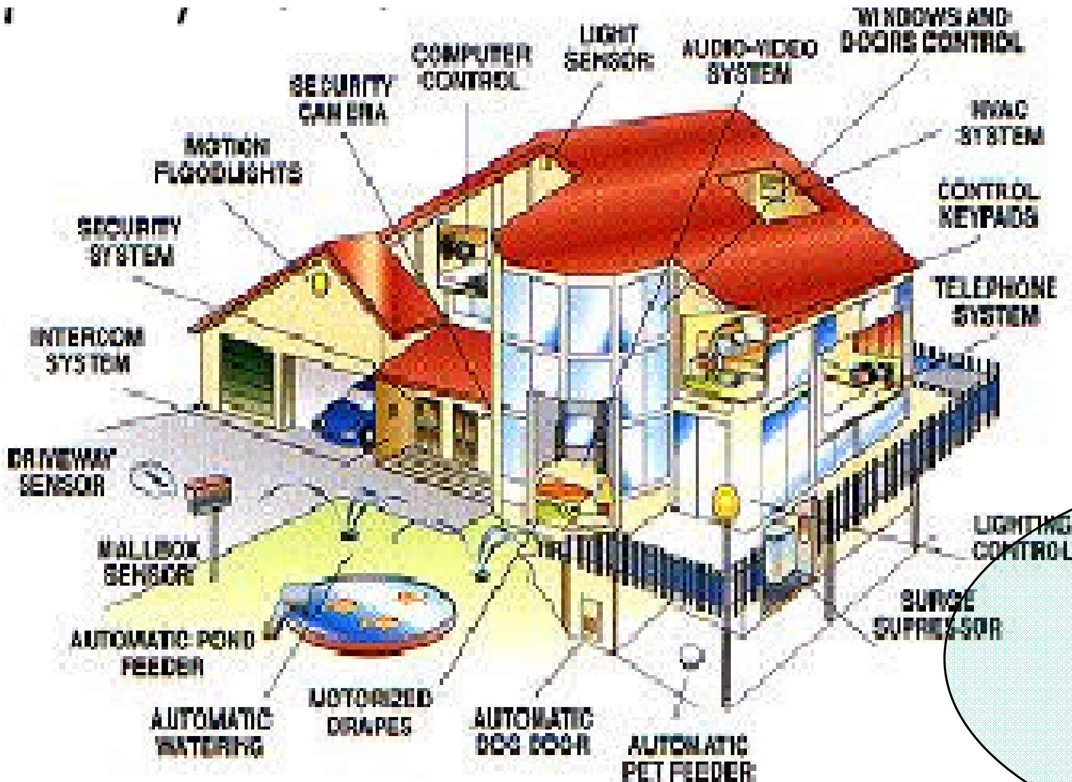
Anno Accademico 2004-2005

Lecturer:
Prof. William Fornaciari
Politecnico di Milano, DEI
fornacia@elet.polimi.it
www.elet.polimi.it/people/fornacia

Embedded Systems Everywhere

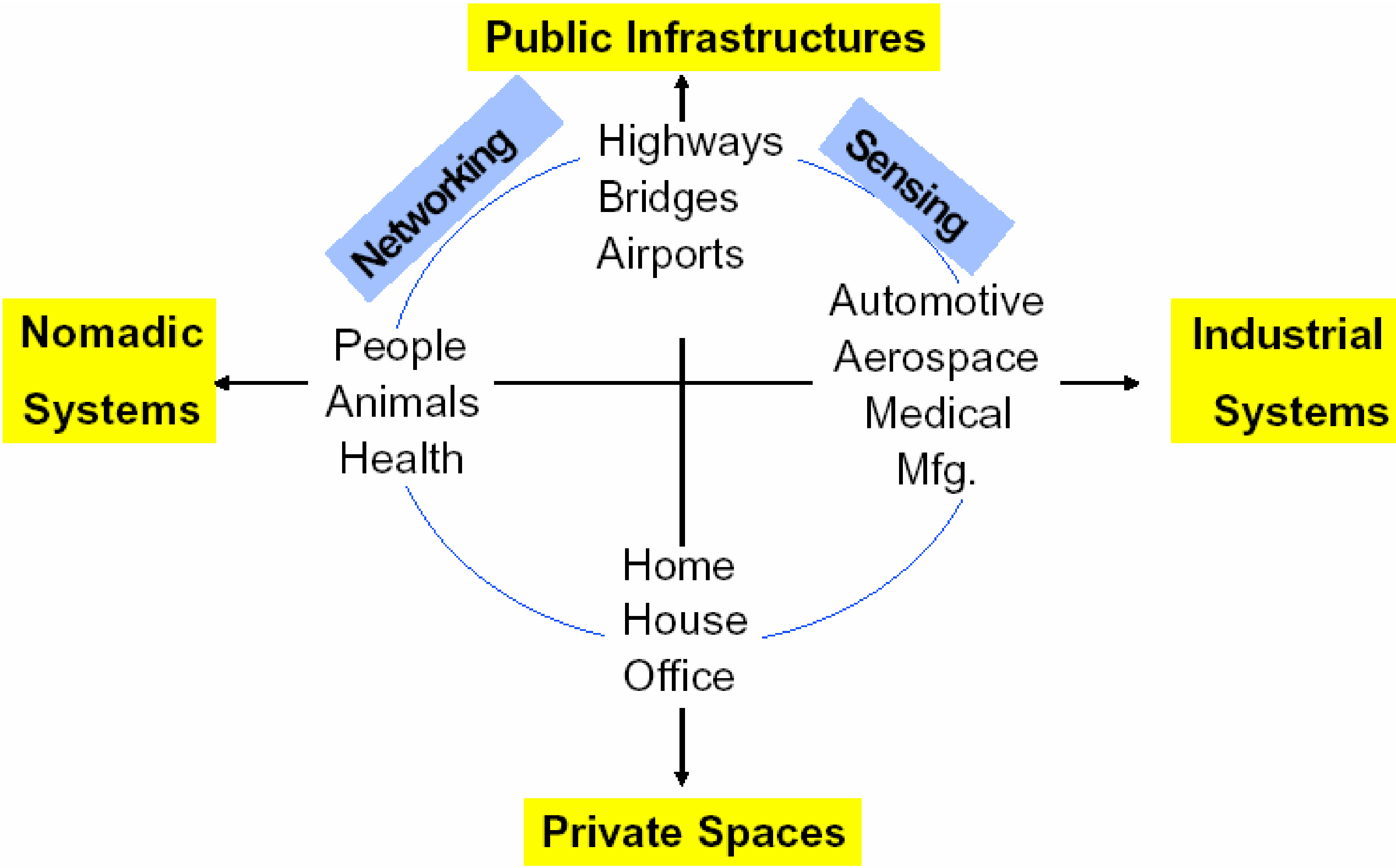
Networked Embedded Intelligence

- Enabling transportation, infrastructure industries
- Leading to revolutions like the digital home
- Turning ambient dreams into reality
- Enabling sensor networks improving our quality of life



Ubiquitous
Low Power
High performance
Interconnected

Four main application contexts



Embedded Systems 10 years from now

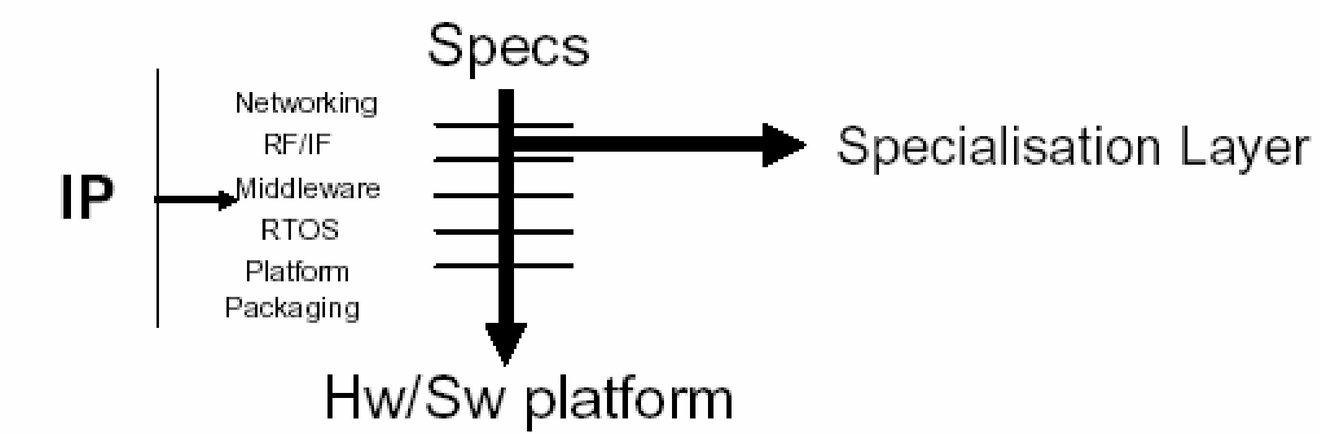
- Networked: from working in isolation towards communicating, networked, distributed solutions
- Secure: threatened by enormous security issues, challenging its technical and economical viability
- Complex:
  - Giga-complexity enabled by nano-technology
  - Complex through heterogeneity
  - Transducer devices
    - Sensors: Biosensors, MEMS, NEMS
    - Actuators/Interactive Screens/Displays
    - Speech input device/Handwriting input devices
  - Computing devices: more software than hardware, application domain specific, reconfigurable
  - Communication: protocols, standards, RF
- Low power: scavenging power

Business as usual…

- Embedded Systems require a holistic approach to design, manufacturing, and skill creation in a distributed industrial context (eco-alliances)
- Embedded systems are complex by nature:
  - GLOBAL TECHNICAL SYSTEMS: networked-sensingintelligence
  - (hw/sw)-actuation = MULTI-DISCIPLINARY
  - OPERATING on an EMBEDDING ENVIRONMENT IN AN APPLICATION DOMAIN with its own requirements and expertise

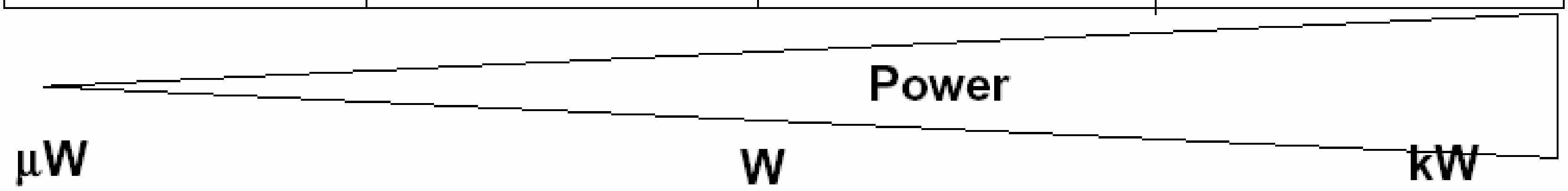
Embedded System Characteristics

- Multi-disciplinary by nature: EE + CS + DOMAIN
- Hard constraints: real-time, low cost, low energy yet complex software on dedicated distributed platforms, short t.t.m., security, ease of use …
- Requires global system approach based on application domain expertise
- Products result from eco-alliances in domain Traverses many layers of abstraction (vertical)



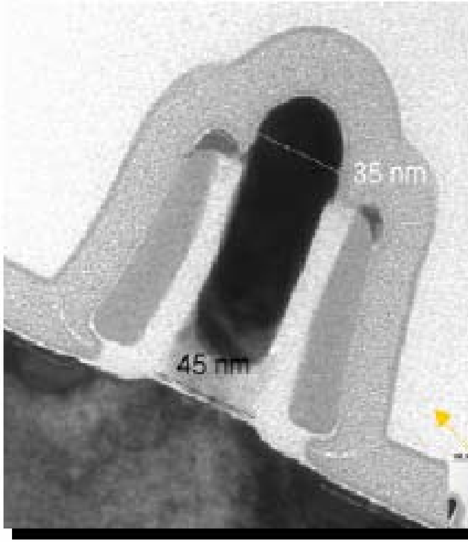
requirements and technologies…

Nomadic	Private Space	Public Infrastr.	Industrial
Sw. Radio Auto-Reconf Low Energy Secure netw	Sensor Netw Multimedia Context sens. Self-x	Sensor netw Longevity Data mining Language proc. Adaptivity	Safety critical Adv. Control Reliability Self-dia-repair
SoC-SiP	SoC	MPU-FPGA-SoC SiP	MPU-FPGA



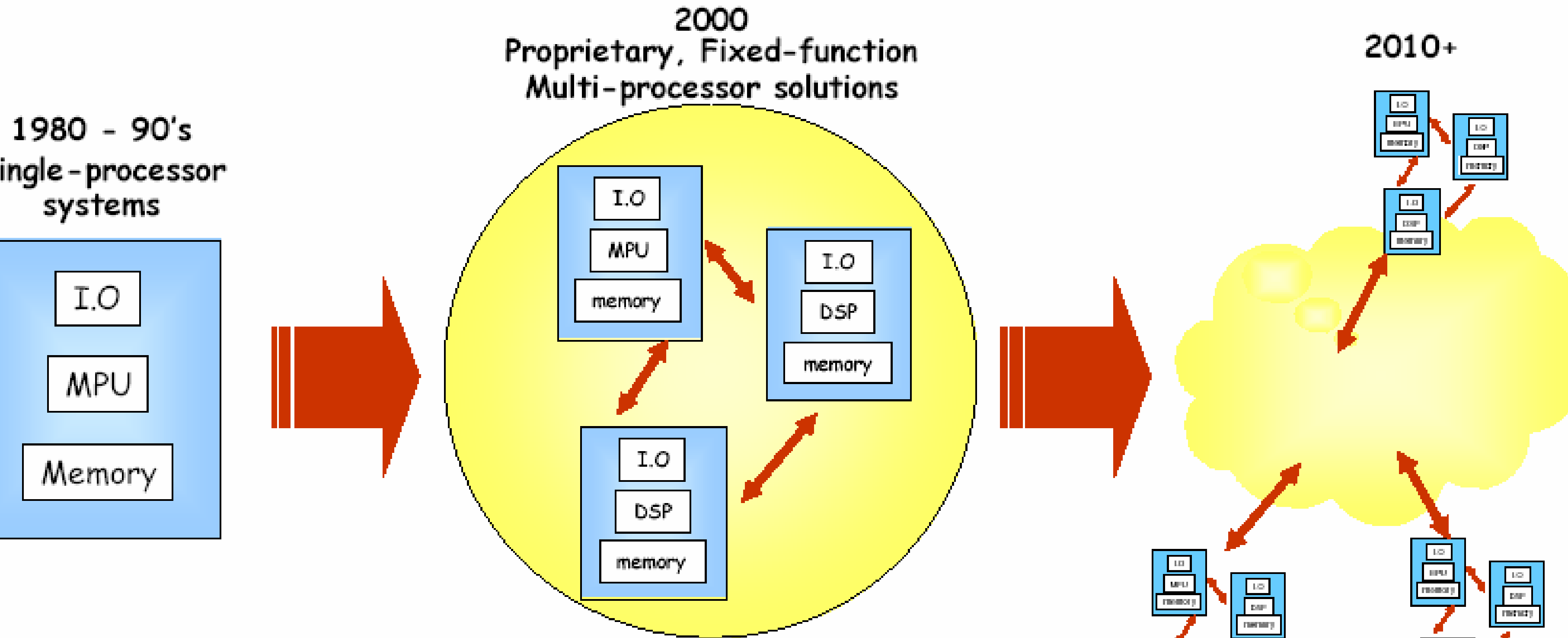
Where has Moore's law brought us?

- Moore's law has had almost 40 years of validity
- The Semiconductors Industry Association (SIA) roadmap for process technology predicts the progress to go on for at least the next 10 – 15 years
- We are presently
  - Developing products in 90 nm
  - 65 nm is in development
  - 45 nm in research



- Physical limits (like e.g. the wavelength of light) have been predicted to stall the progress…….but over time all issues were resolved
- So we may be able to realize 10 nm and smaller geometries, which would produce another factor of 81 more transistors on the same area

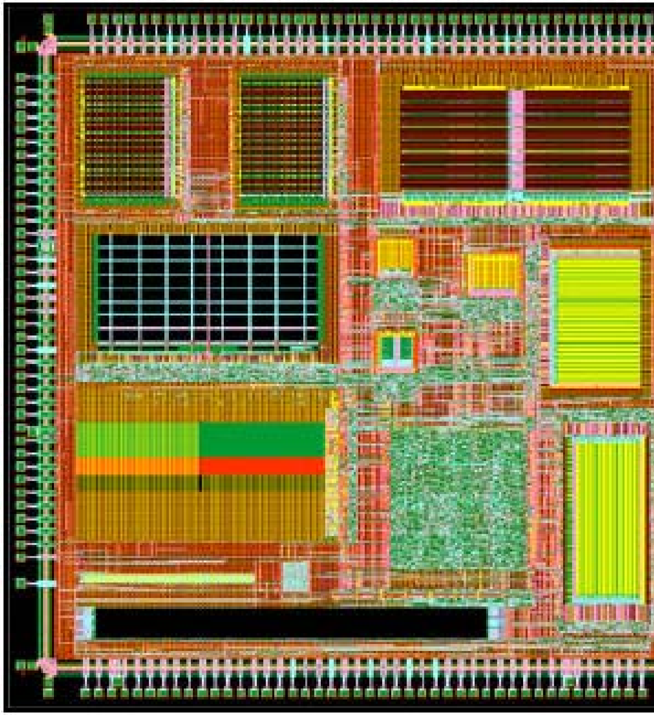
Chip: Architectural evolution



- Driven by Moore's law and low power
- Domain specific processors, going from simple to massive parallelism
- Software methods should cope with multi-processor networked architectures

Design complexity and costs

- IC complexity has grown faster than design efficiency: design crisis
  - The answer was reuse
  - Three generations of re-use
    - Standard cell and automatic synthesis: sea of gates
  - IP block reuse: sea of IP
  - Architecture reuse: platforms
- Design costs are increasing (for 90 nm: up to 20M€ / design)
  - Platform based design (SoC)
  - System in Package (SiP)

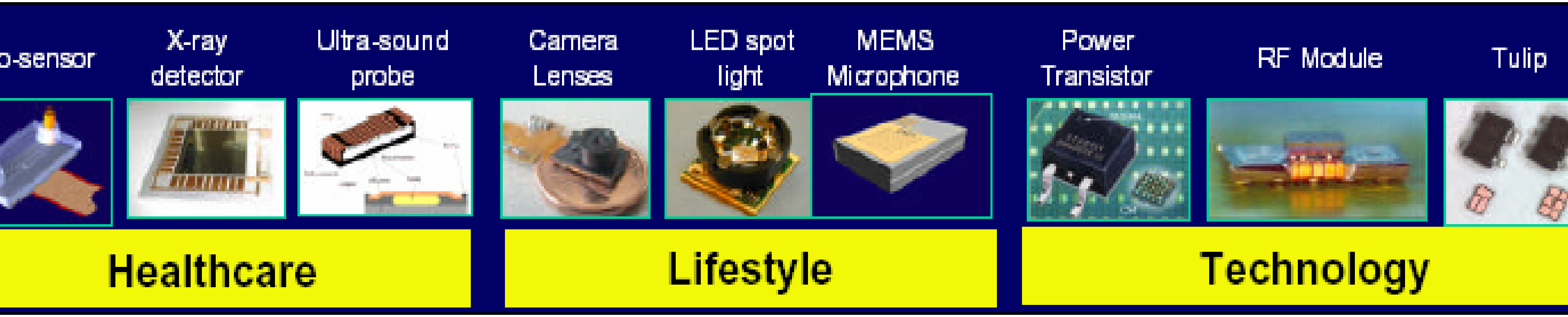


The critical discontinuity

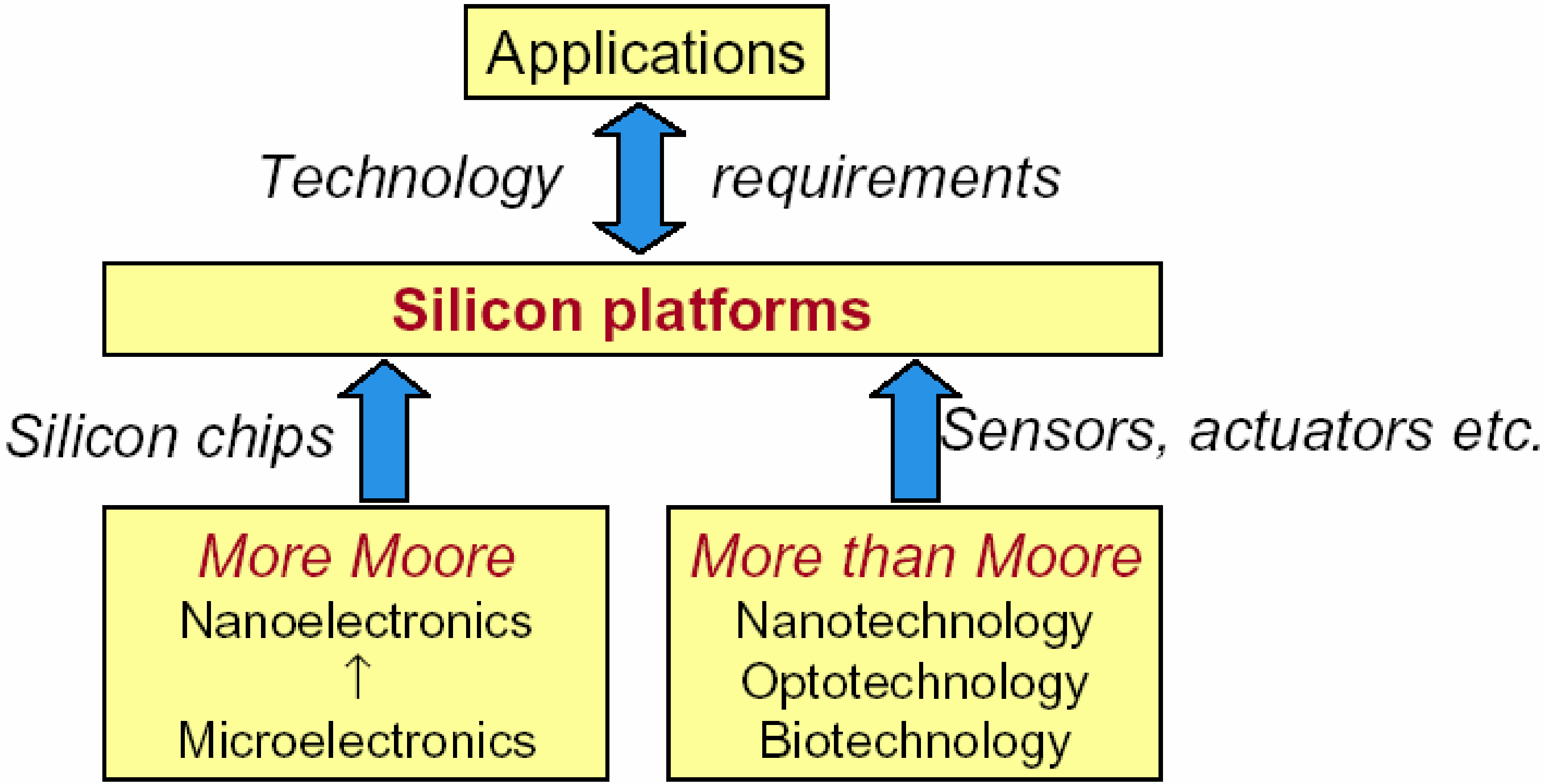
- Moving from closed to interconnected systems
- Moving from closed to open application platforms
- Moving from telecom standards [ATM/WAP etc...] to IP networks
- As a result we shall need
  - huge general purpose low power computing systems with standardized software platforms
  - Also, internet broadband connections will redefine the balance between locally run and web hosted applications

Long term technology trends

- System-on-Chip (SoC)
  - Focus on full integration and lowest cost per transistor
- System-in-Package (SiP)
  - Focus on lowest cost per function and for total system
- Complementing, not competing architectures
- Each requiring a different industrial approach
  - Advanced R&D / knowledge needed
  - Different manufacturing competences

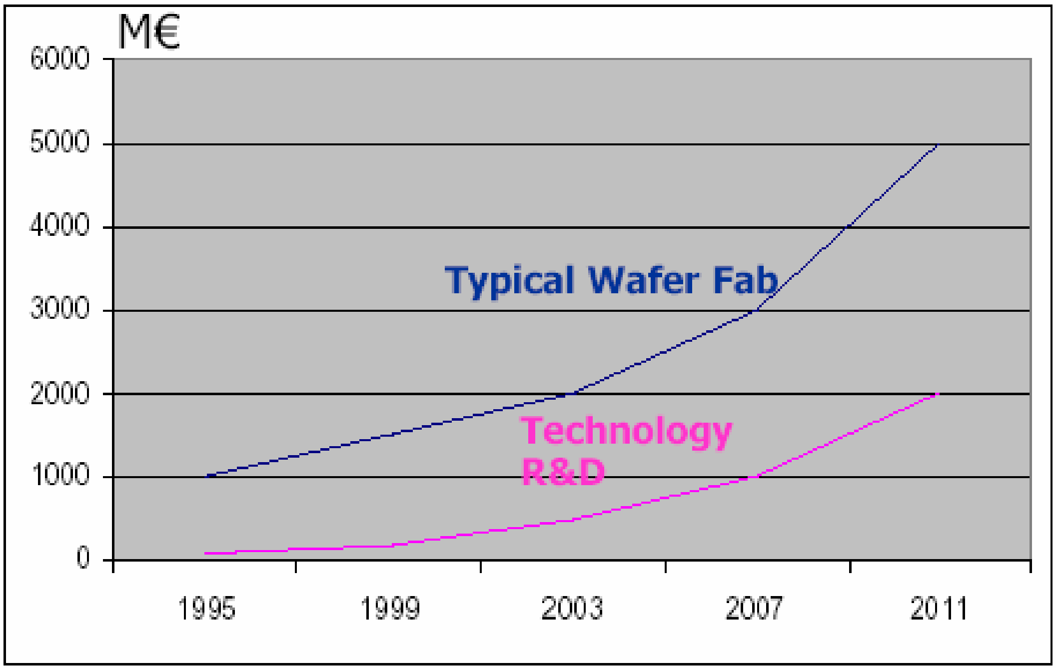


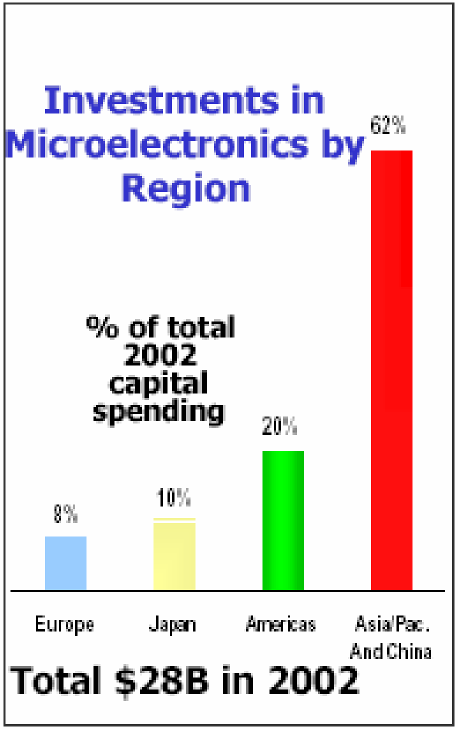
Technology requirements



Leadership and Competitiveness

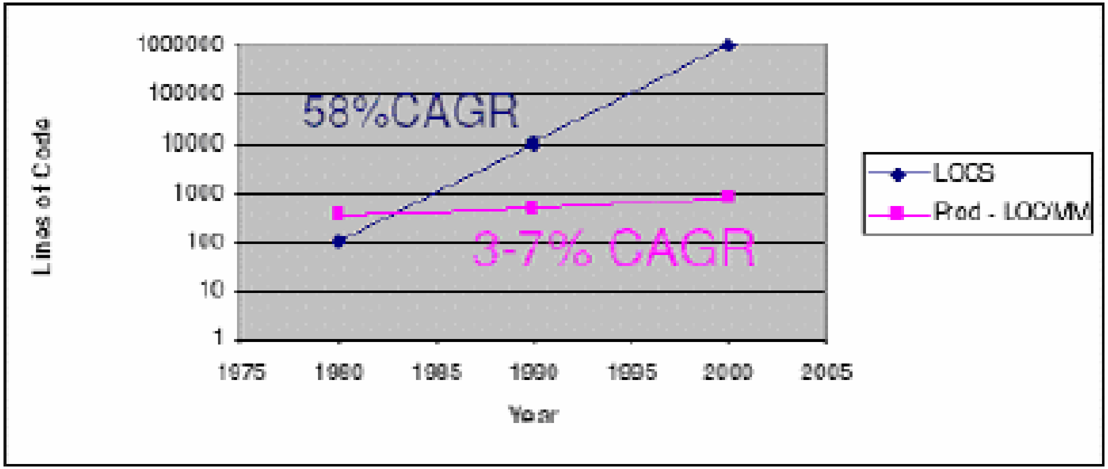
- Europe needs to quickly fill the gap on IP architectures and Computer Science
- Polimi courses?



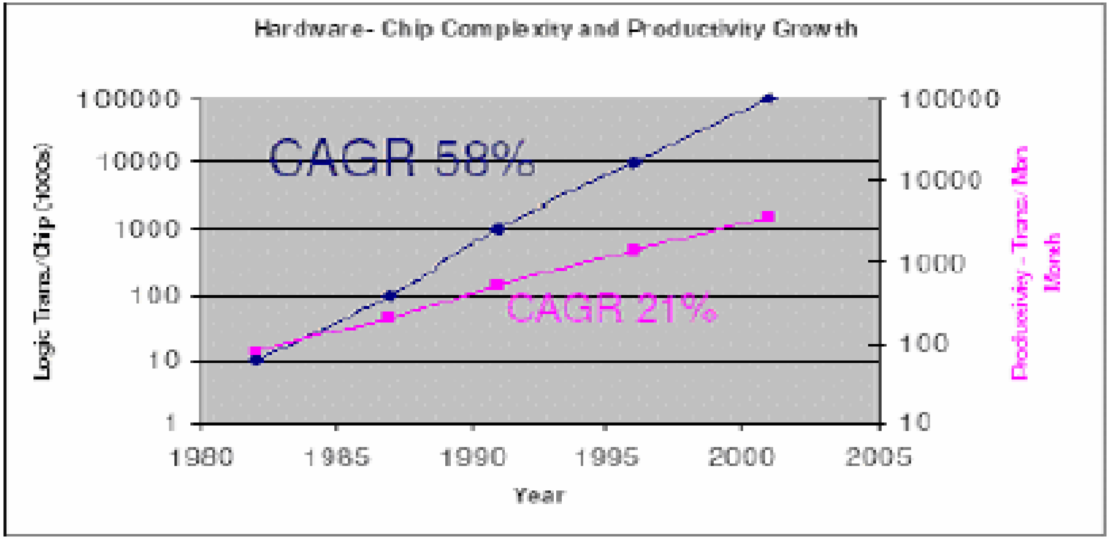


Complexity vs productivity growth

SW complexity & productivity growth

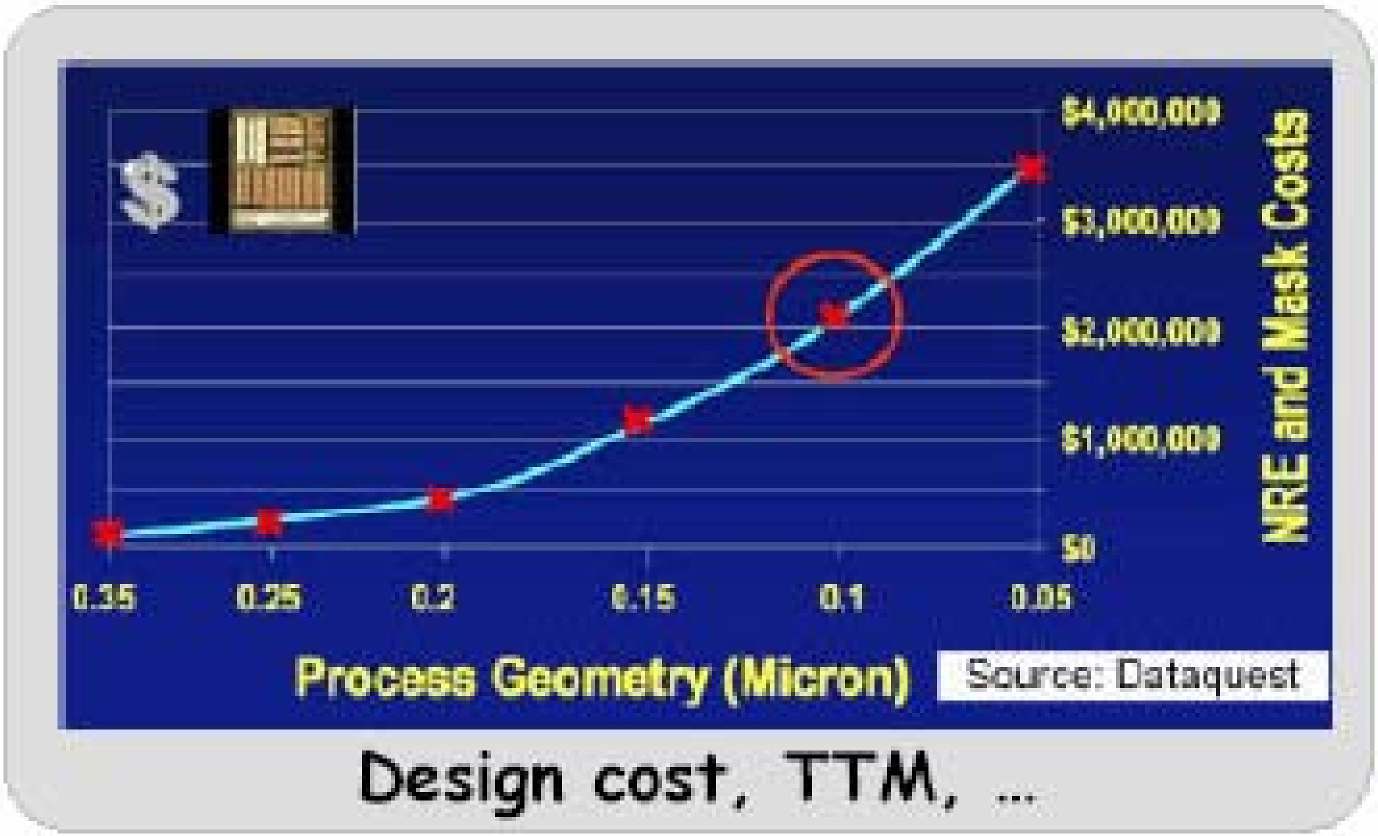


HW complexity & productivity growth



Nano-systems with Giga-complexity

- If design productivity increases in the next 5 years at the pace of the last 5 years, we will need 10-20 times bigger teams to design our future chips



Some research directions

Software and compute architecture

Communication

Pheriferals

Design methodology: addressing complexity

Security

Compute architecture research

- Embedded systems are becoming computing networks: NOCs
  - major challenge for the engineering community, especially for SW developers: traditional SW programming methods do not work well for distributed highly concurrent platforms
- Focus should be the development of a network definition that is
  - agnostic to the processing element
  - independent of the application domain
  - unconcerned about location in the network
- At building block level: focus on domain specific processing units
  - Widely different programming models, from reconfigurable hardware to massively parallel processors
- The whole system architecture to be optimized for critical non-functional qualities, such as energy consumption

Software research

- Real time and embedded software technologies
  - Real time OS
  - Lightweight Middleware with QOS
- Platform independence
  - Address Software portability
  - Address the multiprocessor platform challenge
- Complexity
  - Model driven development allowing development at a higher level of abstraction
  - Verification and validation is increasingly becoming the bottleneck, topics include formal verification, modelling
- Address Standardization
  - Proven in the general purpose world: Linux, UML, XML
  - Embedded world is still a green field

Communication

- ES: Evolving from working in isolation towards communicating, networked, distributed solutions
- IP protocol demonstrated the power of a universal protocol
  - Reduces complexity in development, in validation
  - Ubiquitous, from data telecom network towards the phone [VoIP], the home network, enterprise etc…
  - Hossein Esambolchi (ATT CTO) says “IP is like a Pacman, it will eat everything by the end of the decade”.
- For Embedded systems we should:
  - Define and standardize a universal communication protocol
  - Address heterogeneous communication: Car environment talking to Mobile environment, talking to home environment, talking to www.
  - Address ad-hoc networks: communication established on an ad-hoc or opportunistic basis. Self discovering, self diagnosis, self organizing
  - Move towards several communicating objects forming one functionality

Peripherals

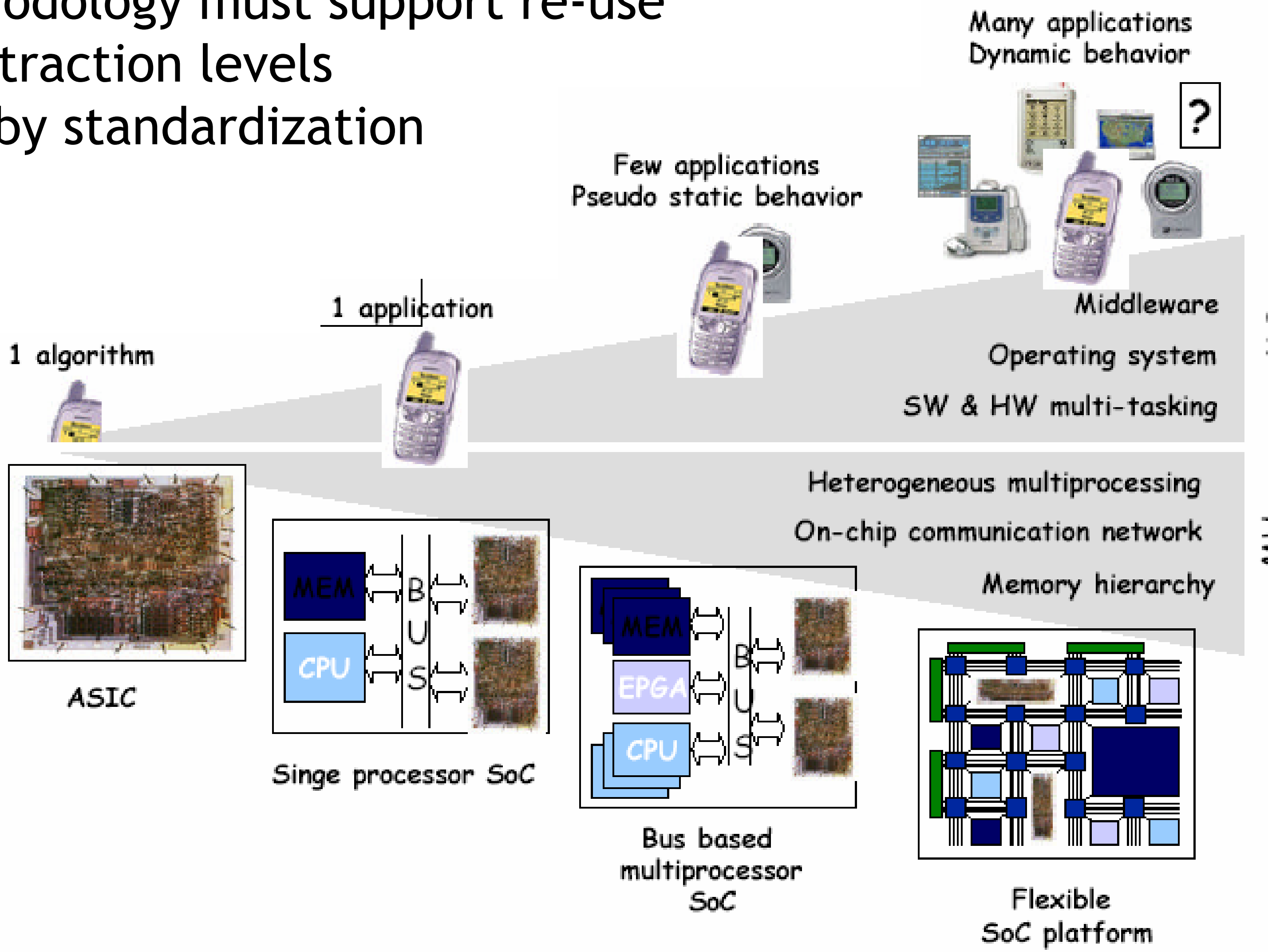
- Transducers, sensors and actuators are essential enabling technologies for embedded systems,
  - interface to/shield from a sometimes harsh, rugged environment,
  - requiring various types of technologies
- Research agenda includes
  - Cost effective, integrated sensors and actuators, based on a wide range of new technologies: MEMS, NEMS, BIOsensors,…
  - Output & Power devices in nm CMOS
  - Communication interfacing technology: wired, RF, optical…
- Should be
  - Capable of living in the harsh environment: mechanically and thermally robust
  - Low power: autonomous, power scavenging
  - Fail safe, degrading gracefully, reliable

complexity

- Quick design exploration: translate quickly applications into architectures
- Formal approach to capture user requirements
- Automation: link to final implementation should be fast:
  - Model based design -> design activity will be done at higher abstraction level where models can be verified and manipulated
- Formal synthesis and formal verification,
  - supported by the definition of clear layers of abstraction, masking implementation details
- Design methodologies must support heterogeneous systems,
  - abstracting the hell of physics,
  - including exotic technologies like MEMS, Biosensors, all in one design flow

Platform based design

Design methodology must support re-use
- at high abstraction levels
- supported by standardization



Security

- Management of rights (DRM) for a connected device will be fundamental to almost all future devices and SOCs
  - "Cisco VP of strategic Technology believes the DRM is THE fundamental factor that will drive their business (internet traffic) to the next level."
- Individual devices are easy targets for disruptive attacks in open, ad hoc wireless networks
- Widespred diffusion of intelligence/data (e.g., into smartcards) can become a new source of attacks (like DPA)
- Challenges:
  - The basis for any security solution is a trusted infrastructure, putting challenges towards all components of an embedded system
  - Standardization
  - Global end to end security management