Hardware Modeling

Giovanni De Micheli
Integrated Systems Centre
EPF Lausanne

Delft University of Technology

Module 1

- Objective
  - Electronic systems and their requirements
  - Integrated circuits
  - Design styles

Electronic systems



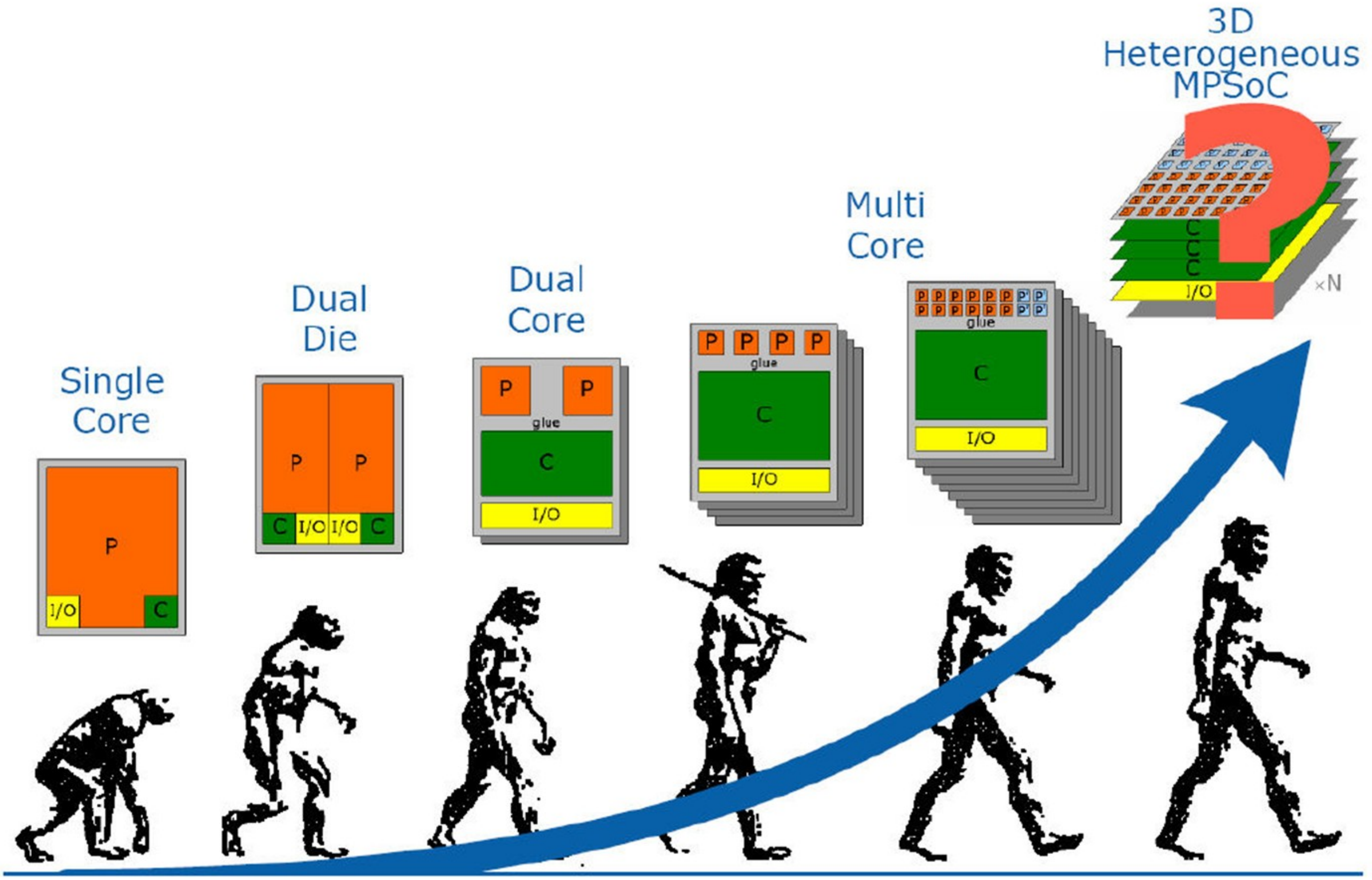

- Systems on chip are everywhere



Technology advances enable increasingly more complex designs

- Challenges:
  - Ride the technology wave
  - Cope with design complexity

Projecting the future



Trends

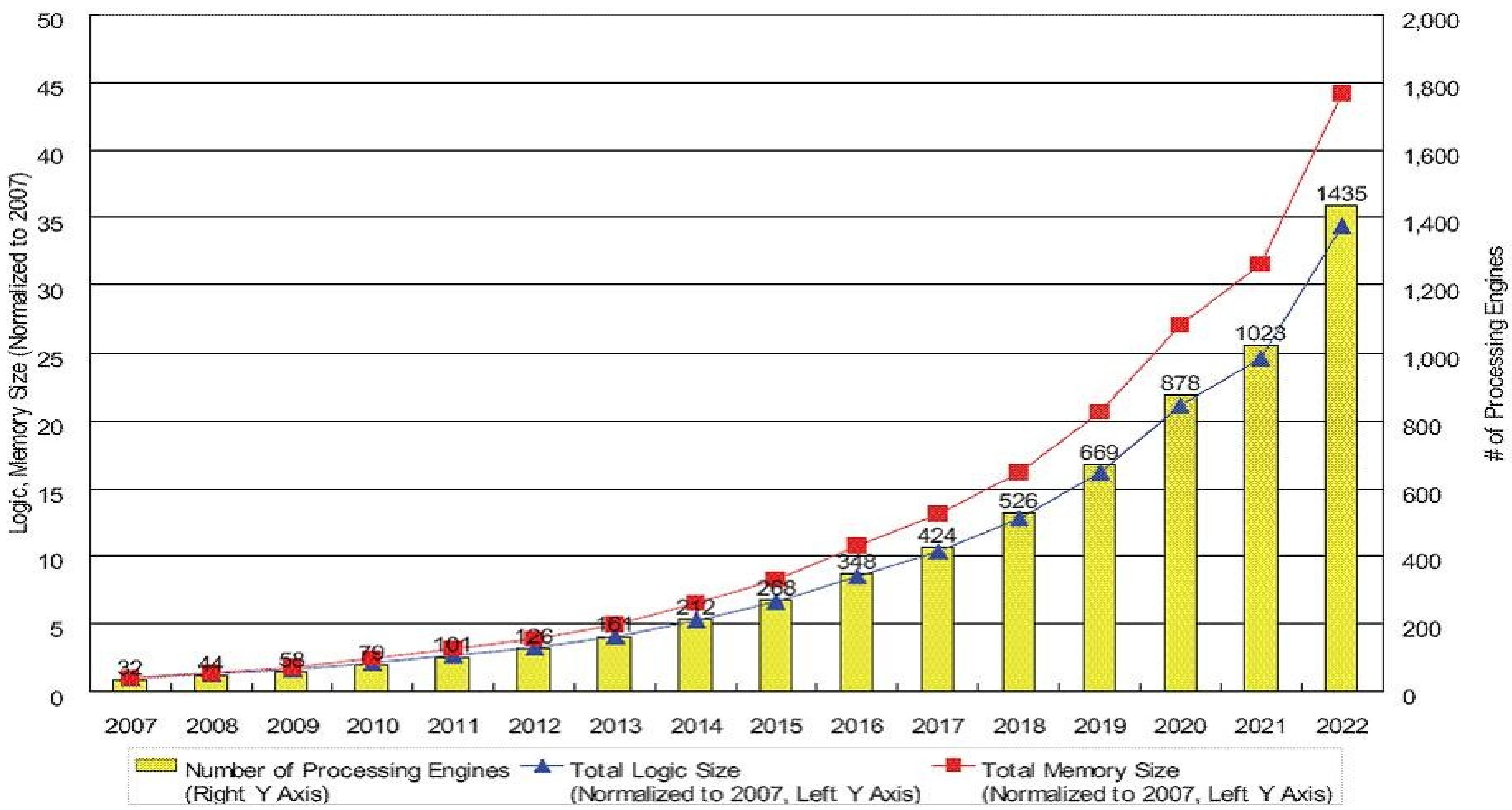
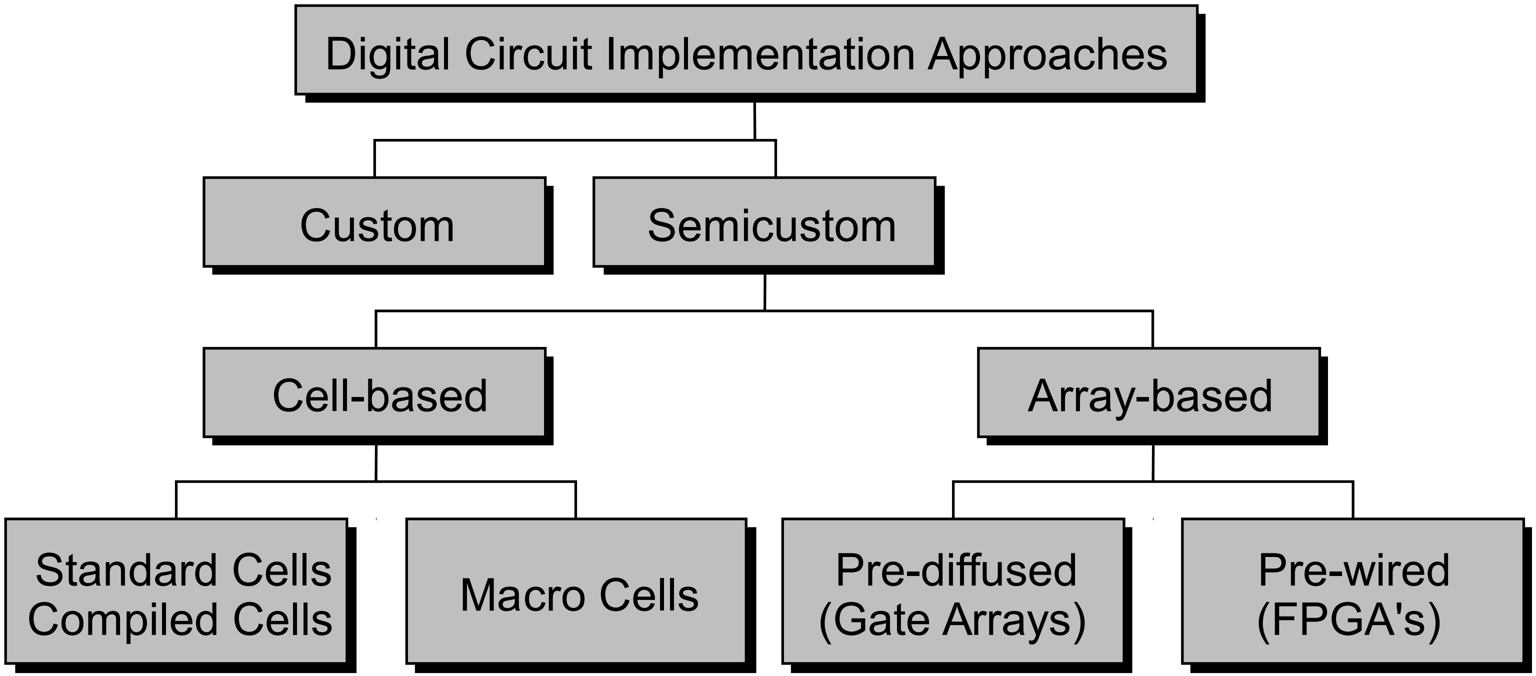


Figure SYSD5 SOC Consumer Portable Design Complexity Trends

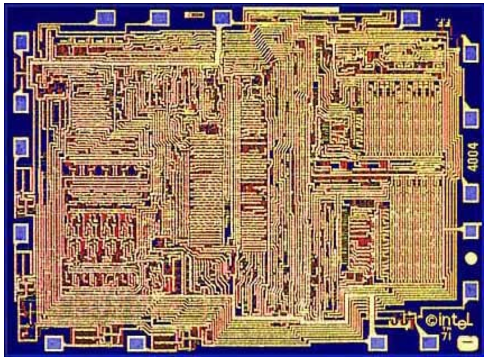
Integrated circuits

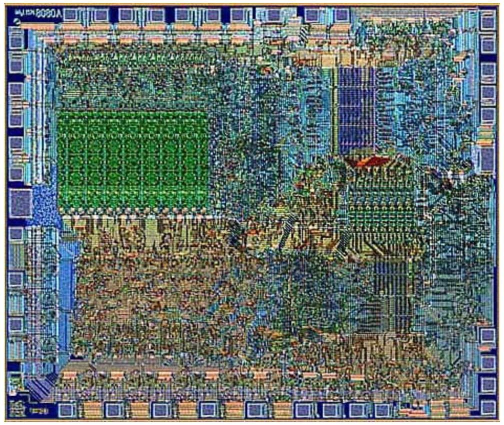
- Systems on Chip (SoC)
  - Multi-processing SoCs (MPSoCs)
- Systems in a package (SiP)
- Silicon technology (CMOS)
  - Down scaling of feature sizes
  - Nanotechnologies on the horizon …
- Different design styles
  - To address performance and cost issues

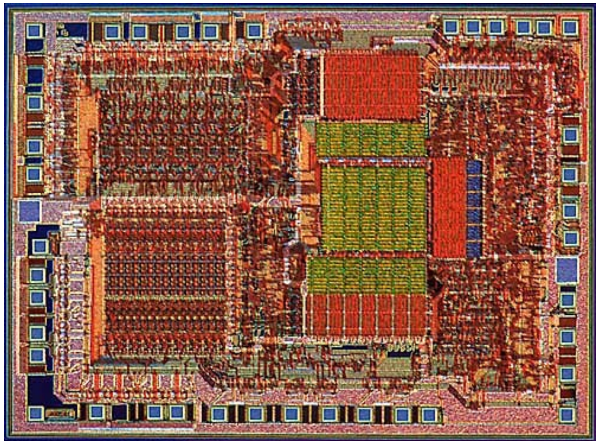
Integrated Circuit Design Styles

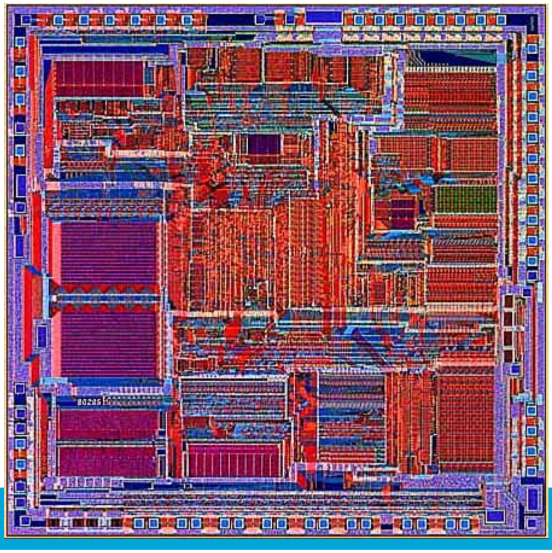


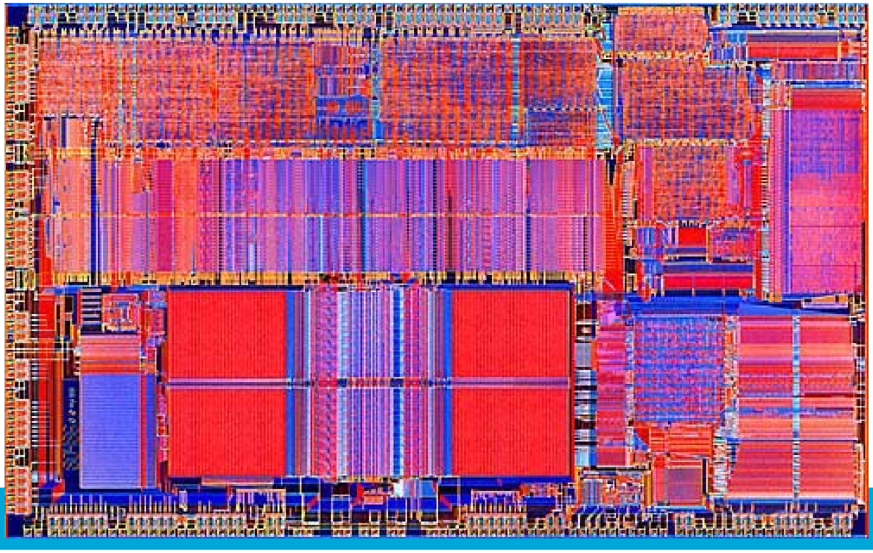
Transition to Automation and Regular Structures


Intel 4004 ('71)


Intel 8080


Intel 8085

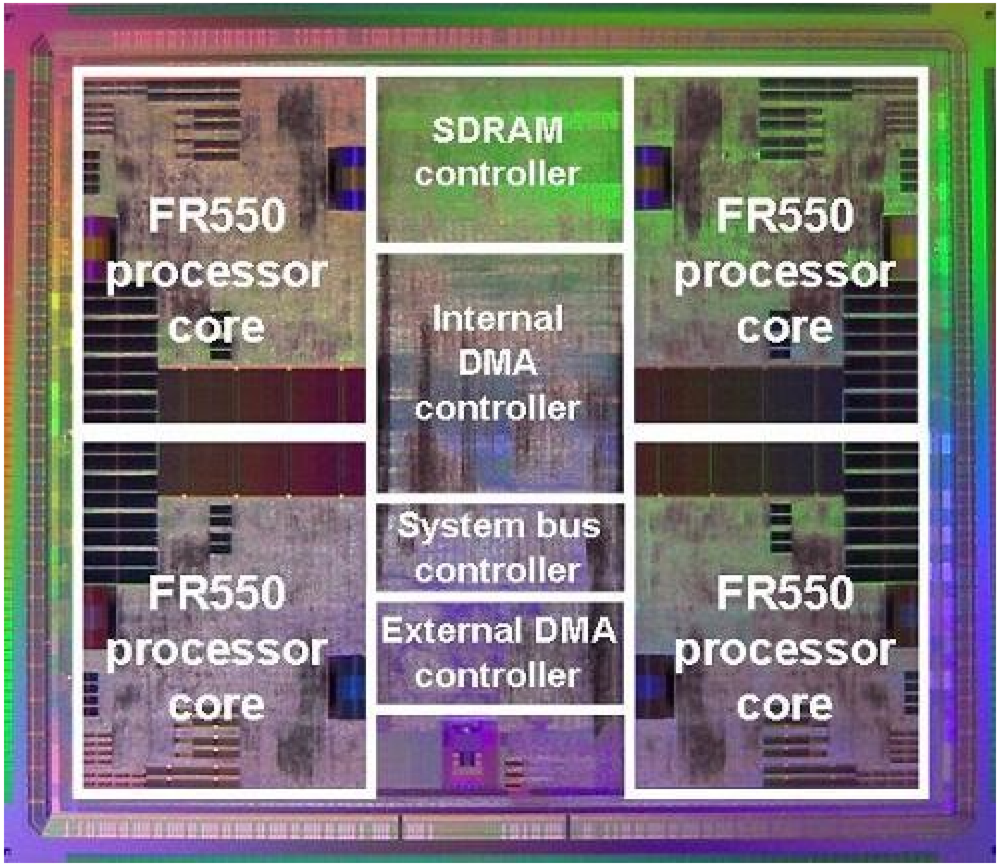

Intel 8286

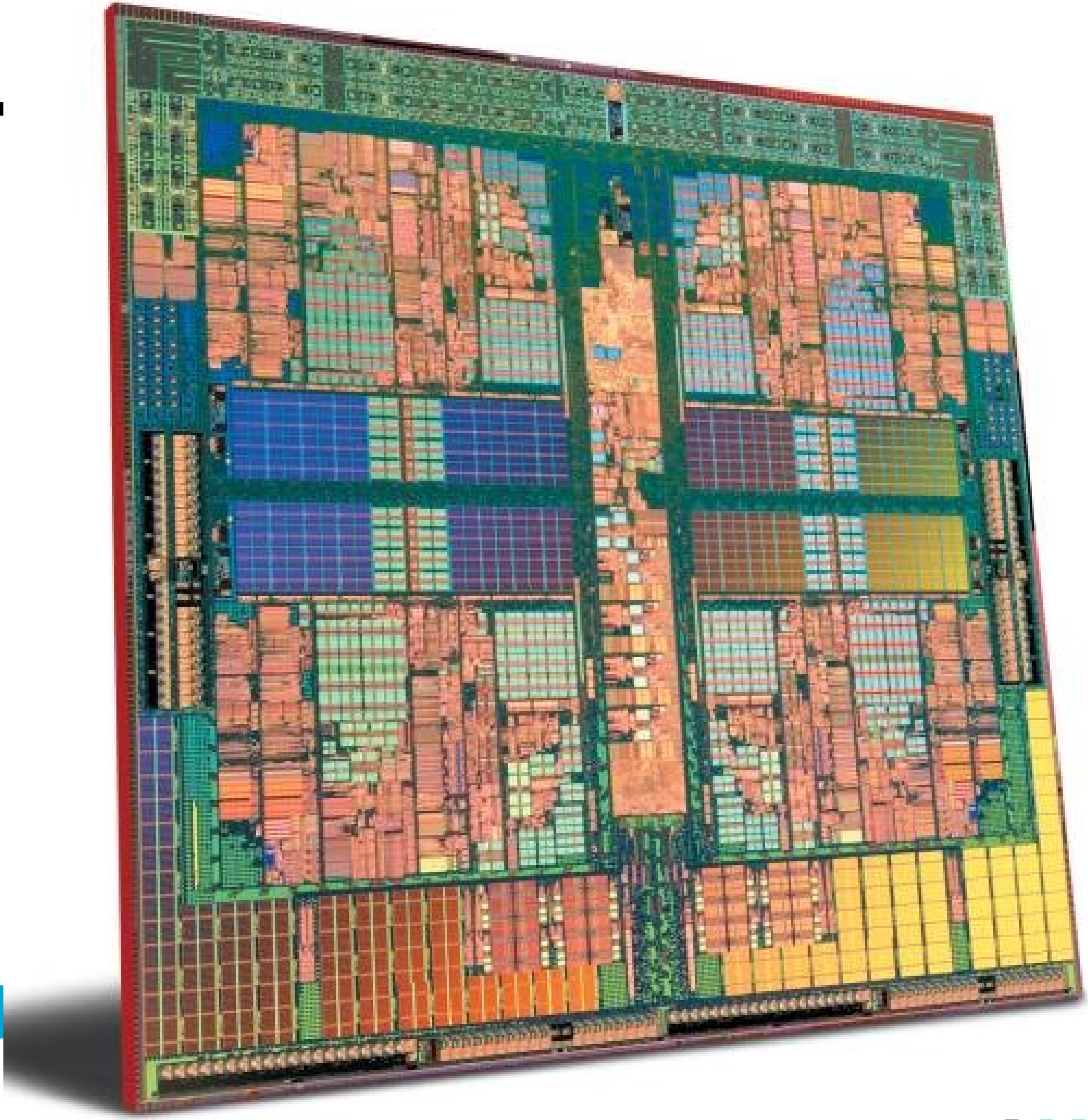

Intel 8486

Courtesy Intel

TU Delft

Multi Core designs





Module 2

- Objective
  - Electronic design automation
  - Synthesis and optimization
  - Multi-criteria optimization

Computer-aided design

- Enabling design methodology
  - Support large scale system design
  - Design optimization, trade-offs
  - Reduce design time and time to market

micro-Electronic System Level design

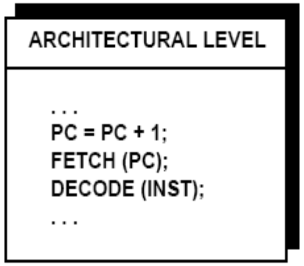
- Conceptualization and modeling
  - Hardware description languages
- Synthesis and optimization
  - Model refinement
- Validation
  - Check for correctness

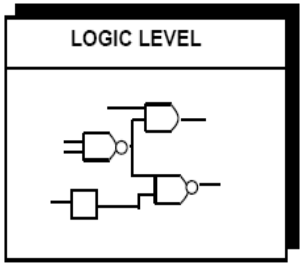
Synthesis history

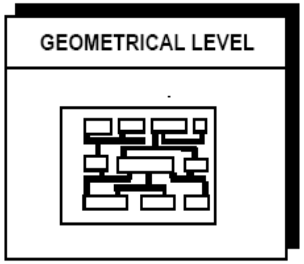
- Few logic synthesis algorithms and tools existed in the 70's
- Link to place and route for automatic design
  - Innovative methods at IBM, Bell Labs, Berkeley, Stanford
- First prototype synthesis tools in the early 80s
  - YLE [Brayton], MIS [Berkeley], Espresso
- First logic synthesis companies in the late 80's
  - Synopsys and others
- Today: Cadence, Mentor, Forte Design, Xilinx Vivado HLS

Modeling abstractions

- System level
  - Untimed specification
- Architectural level
  - Operations implemented by resources
- Logic level
  - Logic functions implemented by gates
- Geometrical level
  - Transistors and wires







TU Delft

System synthesis

- Architectural-level synthesis
  - Determine macroscopic structure
    - Interconnection of major building blocks
- Logic-level synthesis
  - Determine the microscopic structure
    - Interconnection of logic gates
- Physical design
  - Geometrical-level synthesis
  - Determine positions and connections

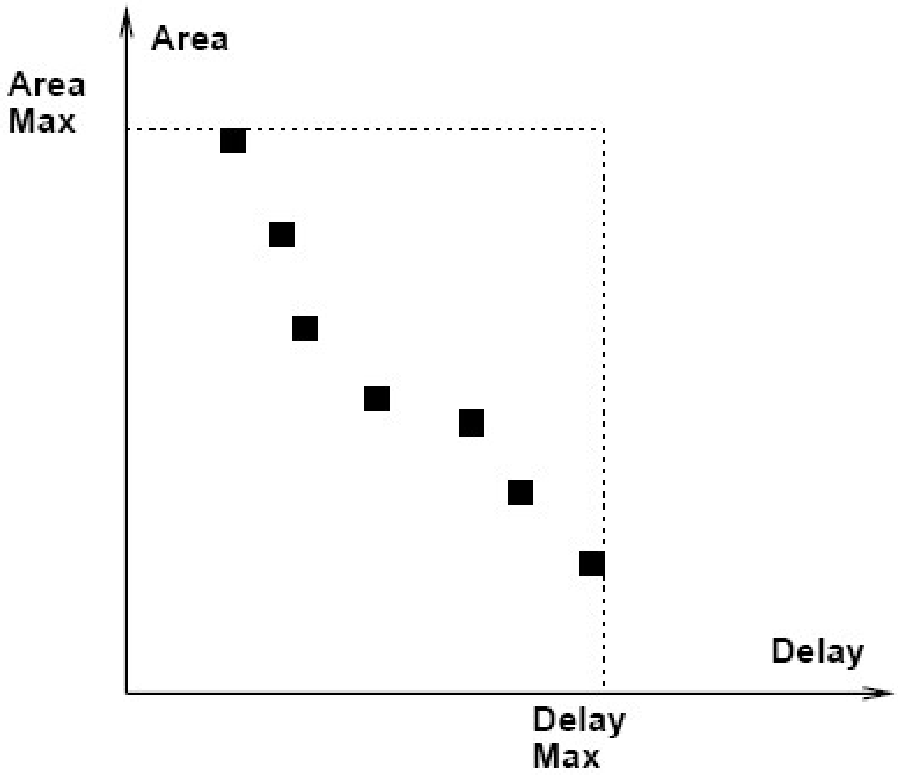
Synthesis and optimization

- Synthesis with no optimization has no value
- Optimization is the means to outperform manual design
- Objectives
  - Performance
    - Frequency, latency, throughput
  - Energy consumption
  - Area (yield and packaging cost)
  - Testability, dependability, …
- Optimization has multiple objectives
  - Trade off

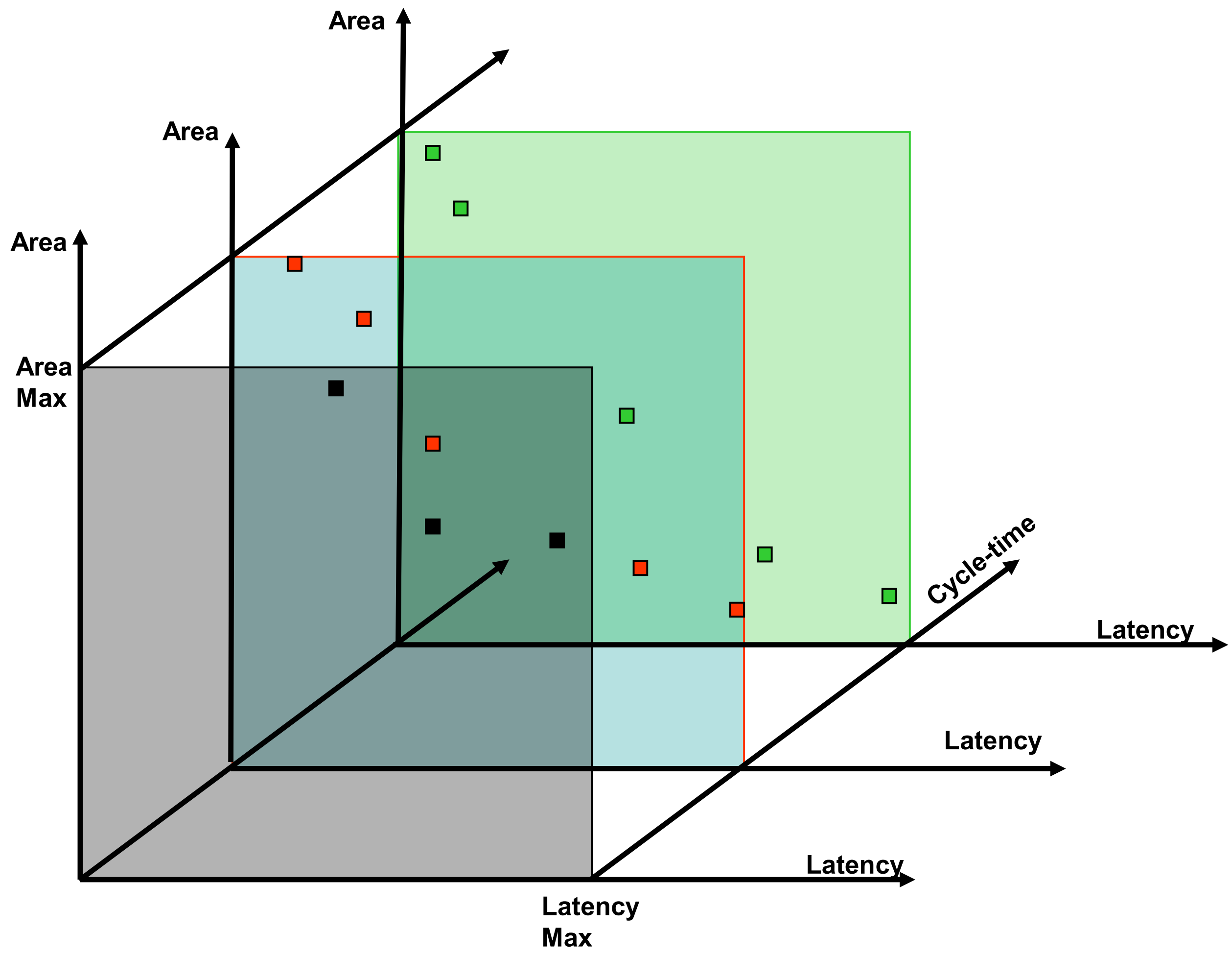
Pareto points

- Multi-criteria optimization
- Multiple objectives
- Pareto point:
  - A point of the design space is a Pareto point if there is no other point with:
    - At least one inferior objectives
    - All other objectives inferior or equal

Combinational circuit optimization



Optimization trade-off in sequential circuits

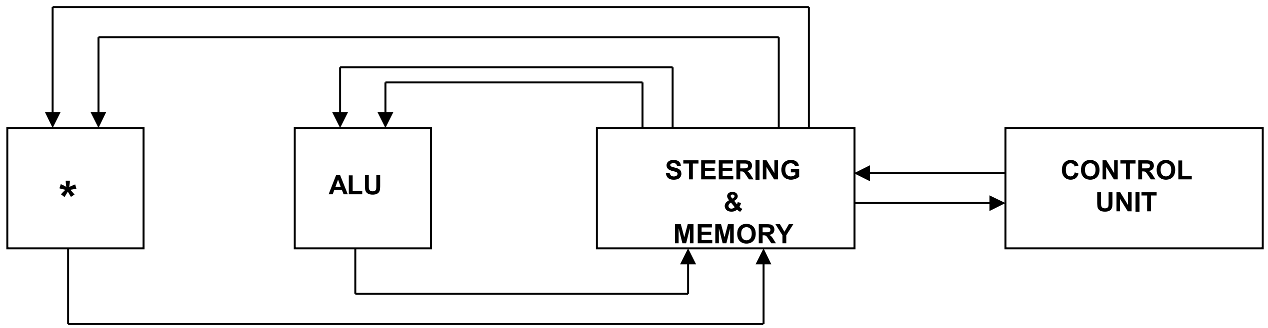


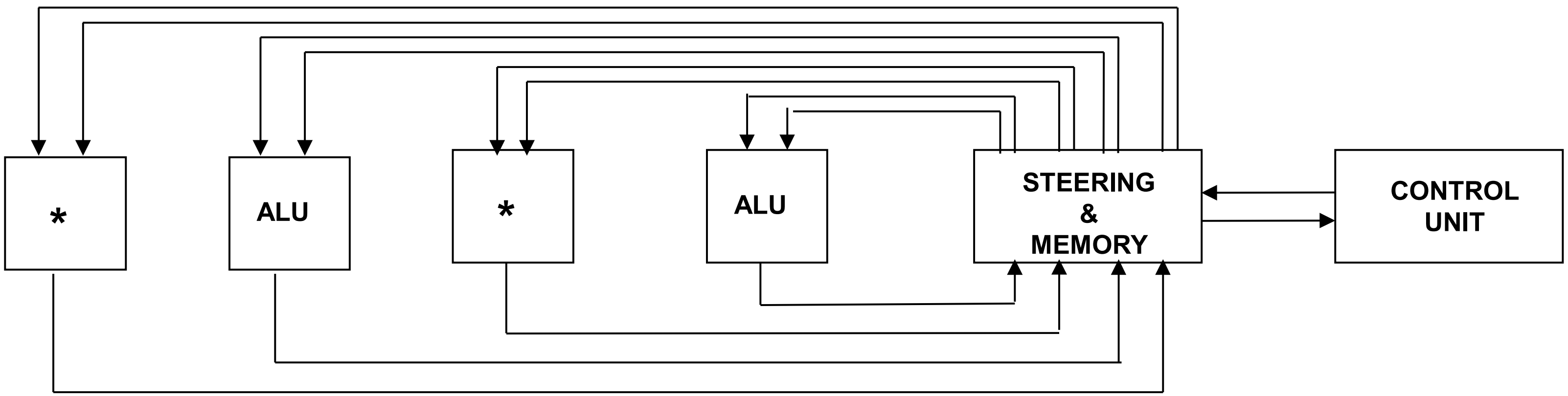
TU Delft

Example: Differential equation solver

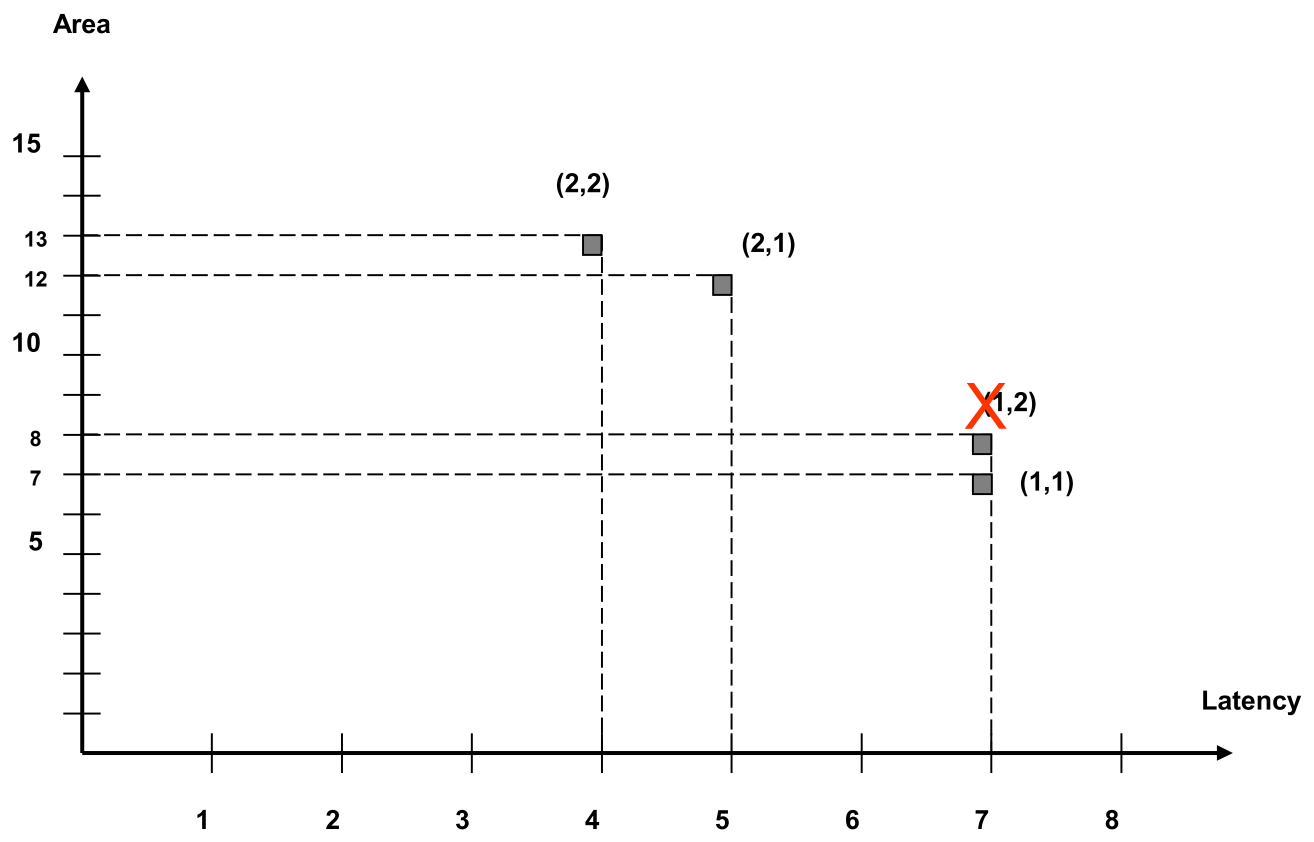
```
diffeq {
    read ( x, y, u, dx, a ) ;
    repeat {
        xl = x + dx;
        ul = u – ( 3 . x . u . dx ) – ( 3 . y . dx ) ;
        yl = y + u . dx ;
        c = x < a ;
        x = xl; u = ul; y = yl ;
    until ( c );
write ( y )
}
```

Example


*
ALU
STEERING
&
MEMORY
CONTROL
UNIT


*
ALU
*
ALU
STEERING
&
MEMORY
CONTROL
UNIT

Example



Summary

- Computer-aided IC design methodology:
  - Capture design by HDL models
  - Synthesize more detailed abstractions
  - Optimize critical parameters
- Computer-aided system design methodology:
  - Support for Hardware/Software co-design
  - Synthesis of hardware, software and interfaces
- Evolving scientific discipline

TU Delft