# SoC-level design

larssv@chalmers.se

DAT093

# Why?

- Course focus (especially lab series) on rather small, self-contained designs
  - Necessary skills regardless of design scope
- Actual industry/research designs larger, with many constraints and dependencies
  - Some consequences addressed in other courses
  - Discuss some other consequences today

# Why not just “system design”?

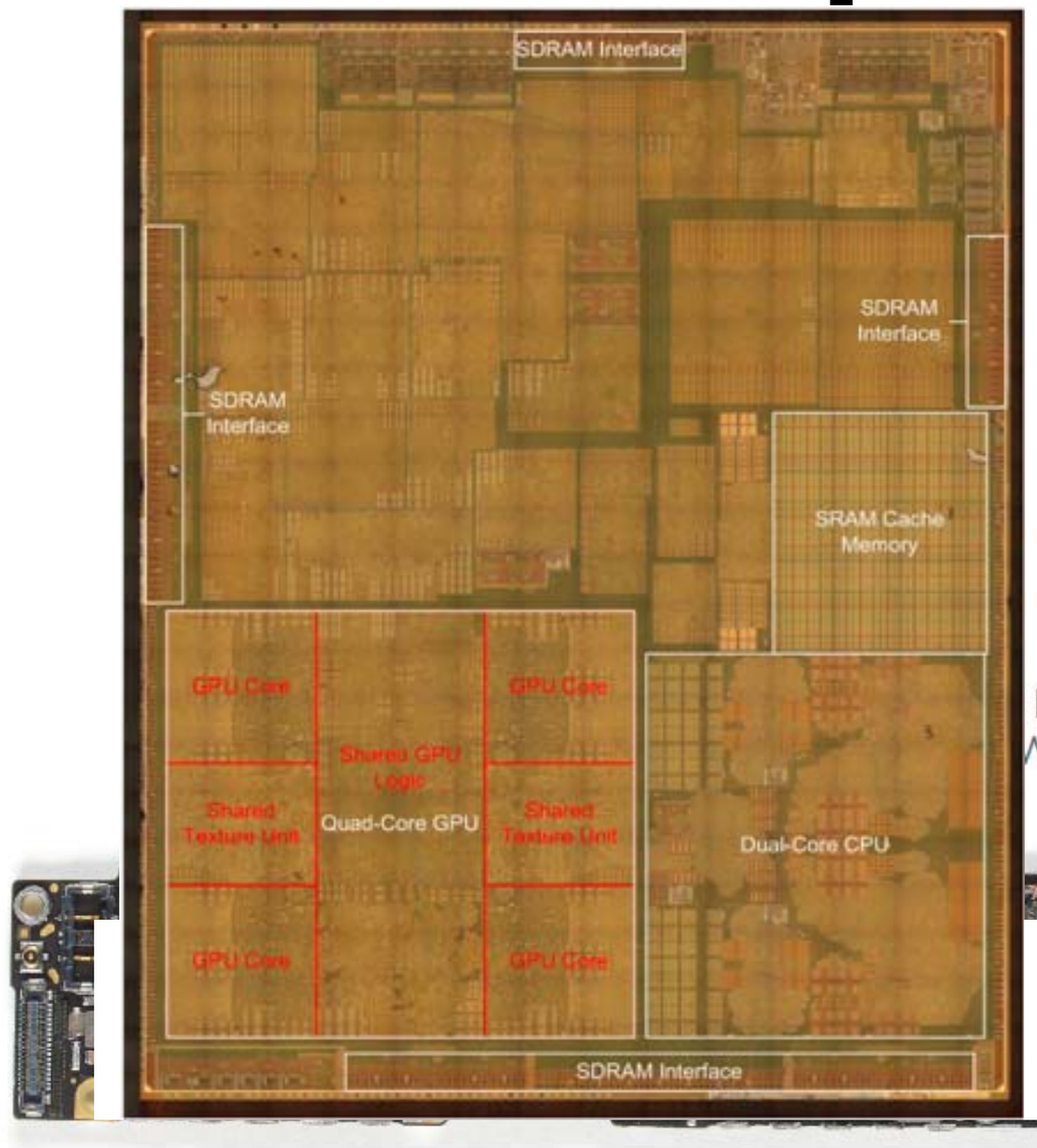
- Common aspects of system design apply also for SoC design
  - Size, complexity, team size, evolving specifications, …
- Implementation medium adds emphasis to some
  - Will exemplify in this lecture

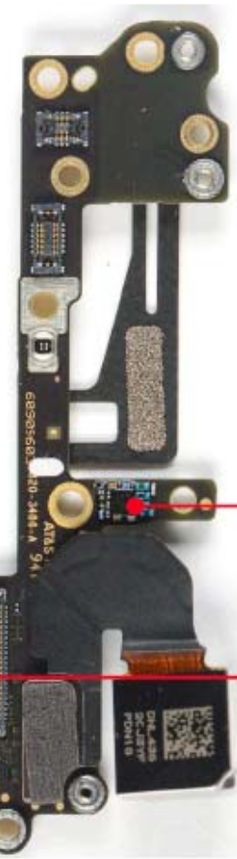
# What?

- SoC: System on Chip
  - “an integrated circuit that integrates all components of a computer or other electronic system” [Wikipedia]
    - Not only a many-core microprocessor
- “Next step” in trend towards higher integration
  - … but term was introduced already in the 1990s
- Recall: a microcontroller is a processor with some peripherals (hard vs soft lecture, 181004)
  - SoC: A microcontroller on steroids?

# SoC example 1

Consumer electronics




iPhone 6
PCB

Is this really an SoC?

- Apple A8 processor A8 (2014)
  - 20nm (TSMC), 2B transistors, 89 mm$^2$

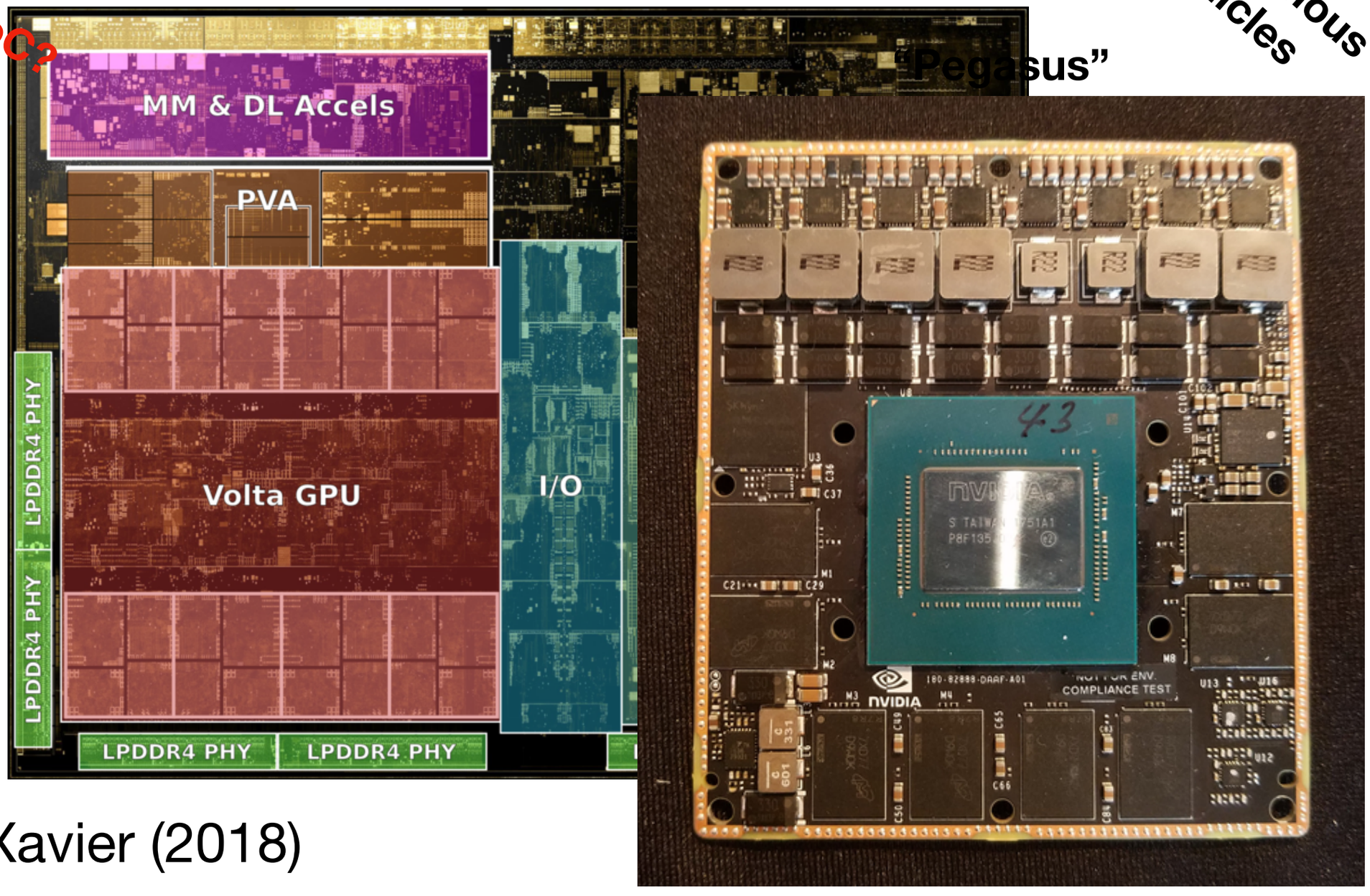
[Chipworks]

# SoC example 2

Is this really an SoC?

Autonomous vehicles

"Pegasus"




- Nvidia Xavier (2018)
  - 9B transistors, 350 mm², TSMC 12FFN process

[wikichip.org]

This page intentionally left blank

# Common observations

- Several different programmable processors
  - A8: 2 ARM cores, 4 GPU cores
  - Xavier: 8 ARM cores, many GPU cores
  - Lund chip: RISCV + vector processor
- Memory (cache or other)
- Misc. peripherals (memories, JTAG, USB, etc)
- Interconnect for all this

# Challenges

- Complexity
- Performance
- Power
- NRE cost
- Manufacturing cost
- …

Cf. lecture slides 180903

Slight changes

# 1. Complexity

- Not only very many transistors, but also many very disparate parts and sub-designs
  - Several “processors” with very different architectures
  - “Accelerators” (special-purpose blocks, very efficient at certain well-defined tasks)
  - Interfaces (memory, networking)

# Complexity, cont.

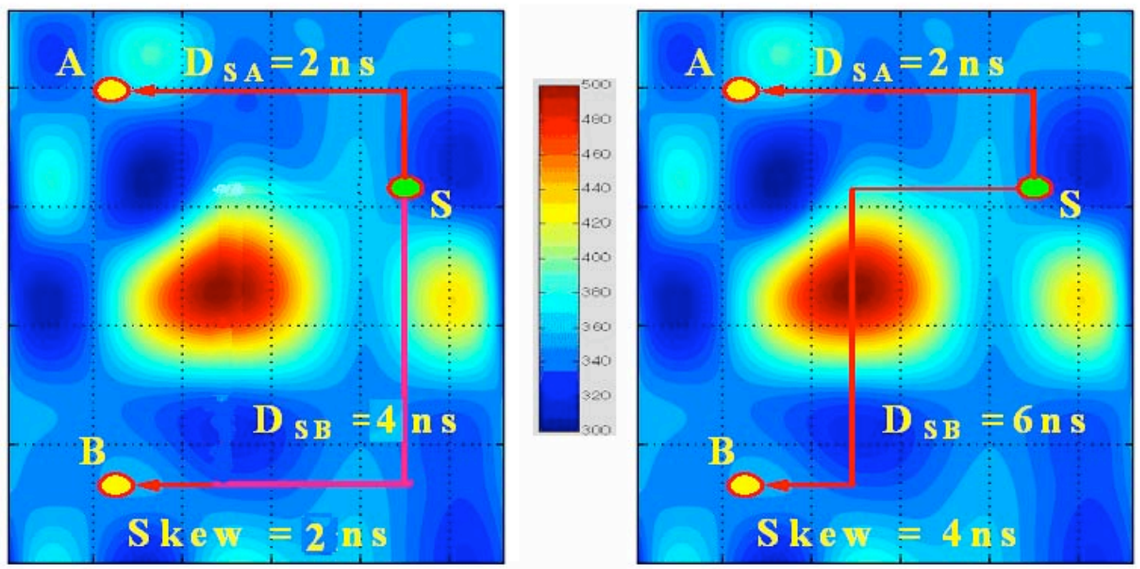
- Each part must be programmed, configured, debugged, etc
  - Multiple instruction sets
    - Multiple programming languages/paradigms
      - Signal processing, graphics, machine learning, etc in addition to “standard” imperative programming
- Design team for 1B-transistor SoC needs much broader skill set than for 1B-transistor memory chip

# 2. Performance

- Not only performance requirements on blocks/cores, but also on all interconnects, data transfers, etc (involves interactions of several blocks)
  - Bus interconnects such as in the Lund chip
    - Contention limits total bandwidth
  - Network-on-Chip (NoC)
    - Similar abstractions as in data networks
    - Scalability, modularity
    - Guest lecture by Ioannis Sourdis tomorrow

# 3. Power dissipation

- Not only need to cope with total power, but also with distribution in time and space
  - Include blockwise power-down mechanisms (unused processors, etc) and adaption (DVS; cf power lecture)
  - Hot-spots may affect performance in many ways!
    - Ex: clock skew may depend on heat profile!




[Liu et al. ICCD 2008, pp. 107–113.]

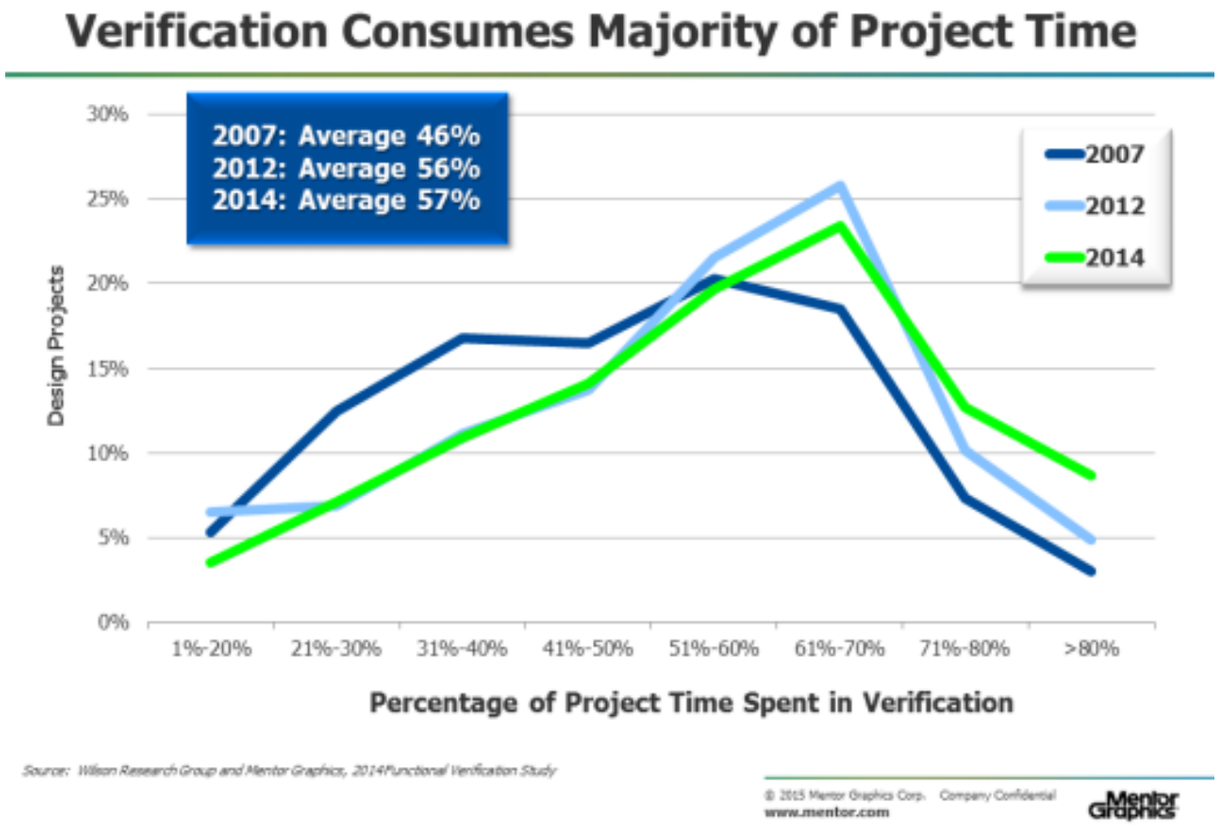
# “Dark silicon”

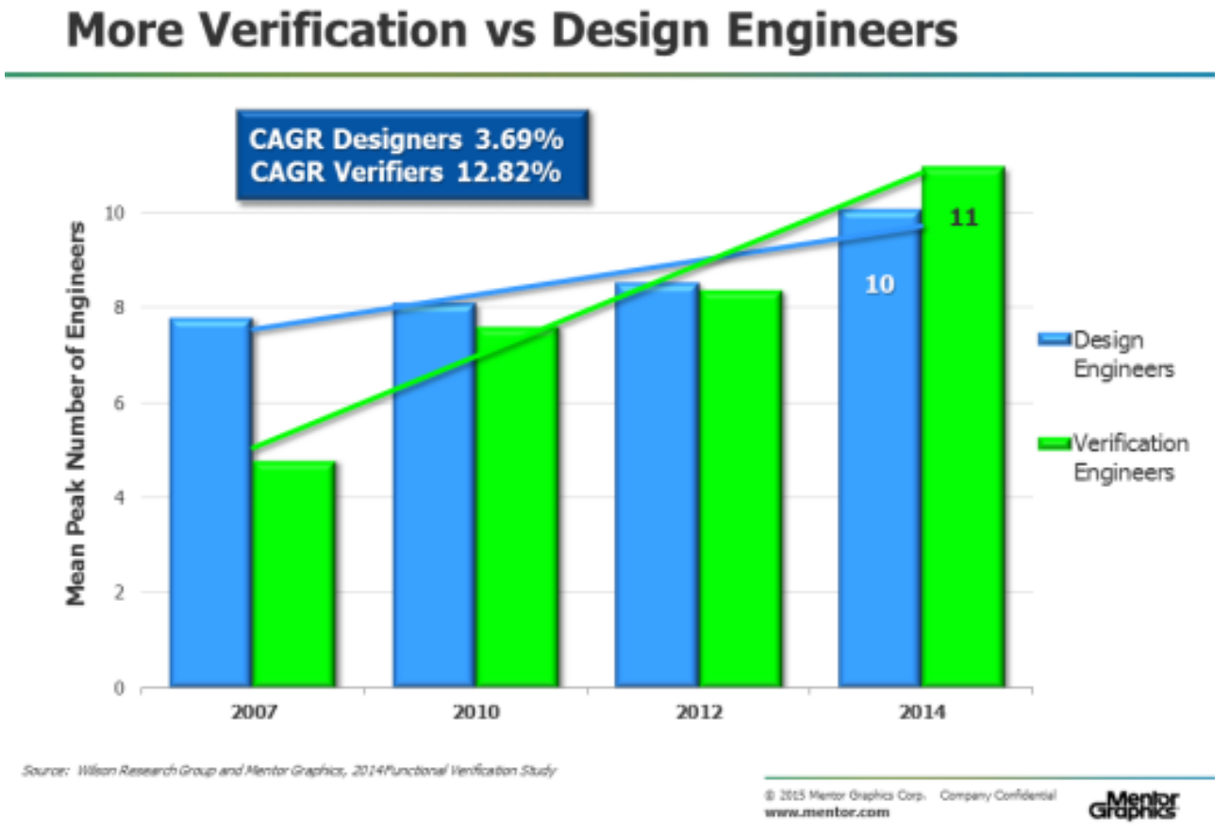
- Maximum allowable overall power may be lower than sum of powers of all blocks
  - Then all blocks cannot be active simultaneously!
    - Dark silicon: it’s there, but you cannot see/use/detect it
- Viable idea e.g. if power is more important than chip area
  - Ex: Include several versions of a filter with different power/performance, use the best one for each use case

# 4. NRE cost

- Not only design and verify each block, but also verify the whole system
  - Complex systems have more ways to malfunction
    - Simulation coverage shrinks with growing system size
    - How select simulation cases?
- Malfunction very expensive for large chip
  - Direct cost of re-spin
  - Lost market opportunity

# Verification dominates



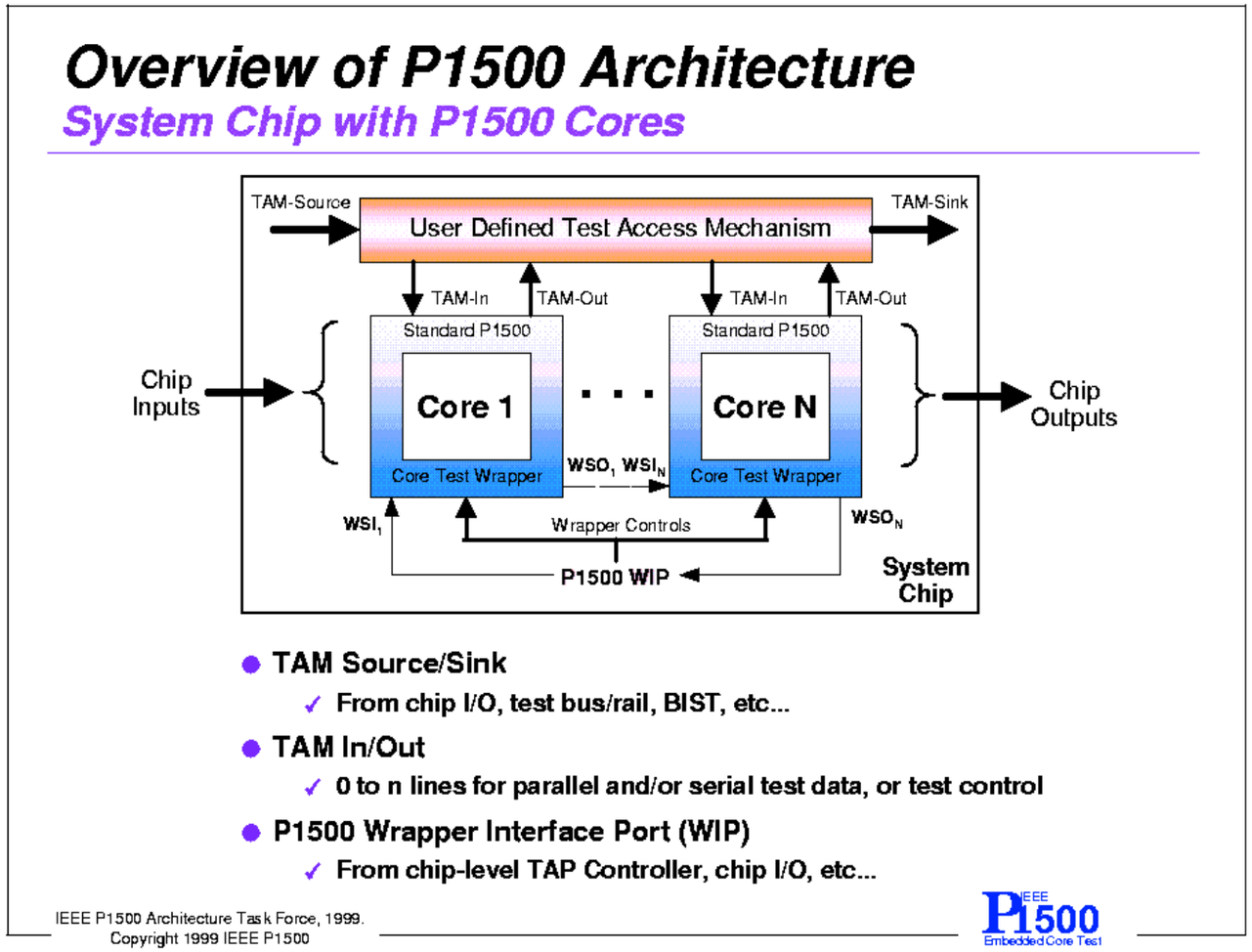

- These numbers are across 1886 chip designs of all sizes

[Wilson Research Group + Mentor Graphics, 2014 Functional Verification study]

# Verification success


- Larger, more expensive designs are taken more seriously?
- Attempted mostly by more mature organizations?

# 5. Production cost



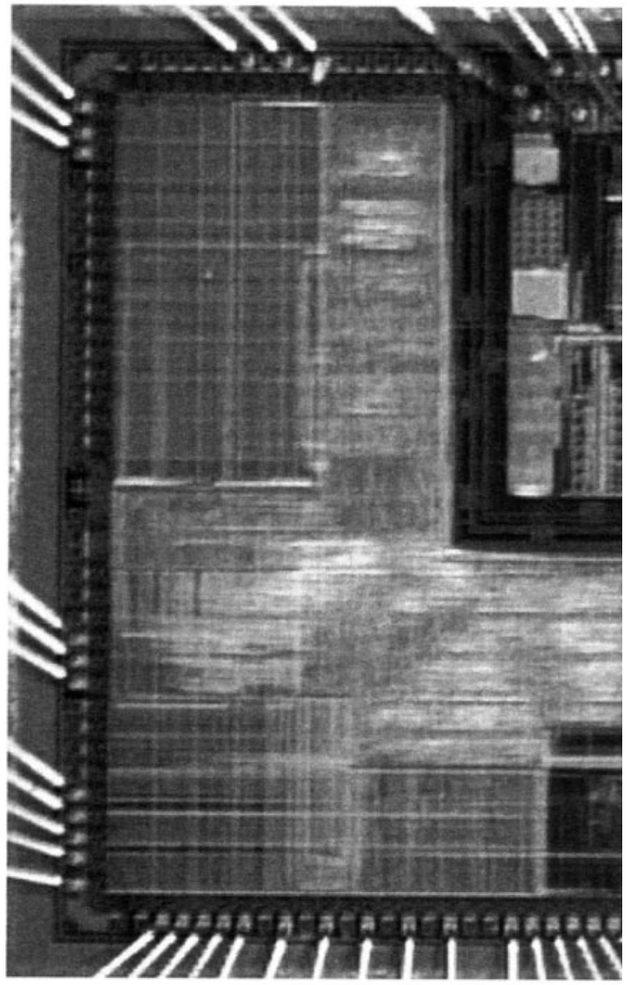

- Not only production testing of each block, but also validation of their interactions
  - Cannot cover all possible interactions, so select significant use cases!
  - Test parallelism may be limited by power dissipation (“dark silicon” again)

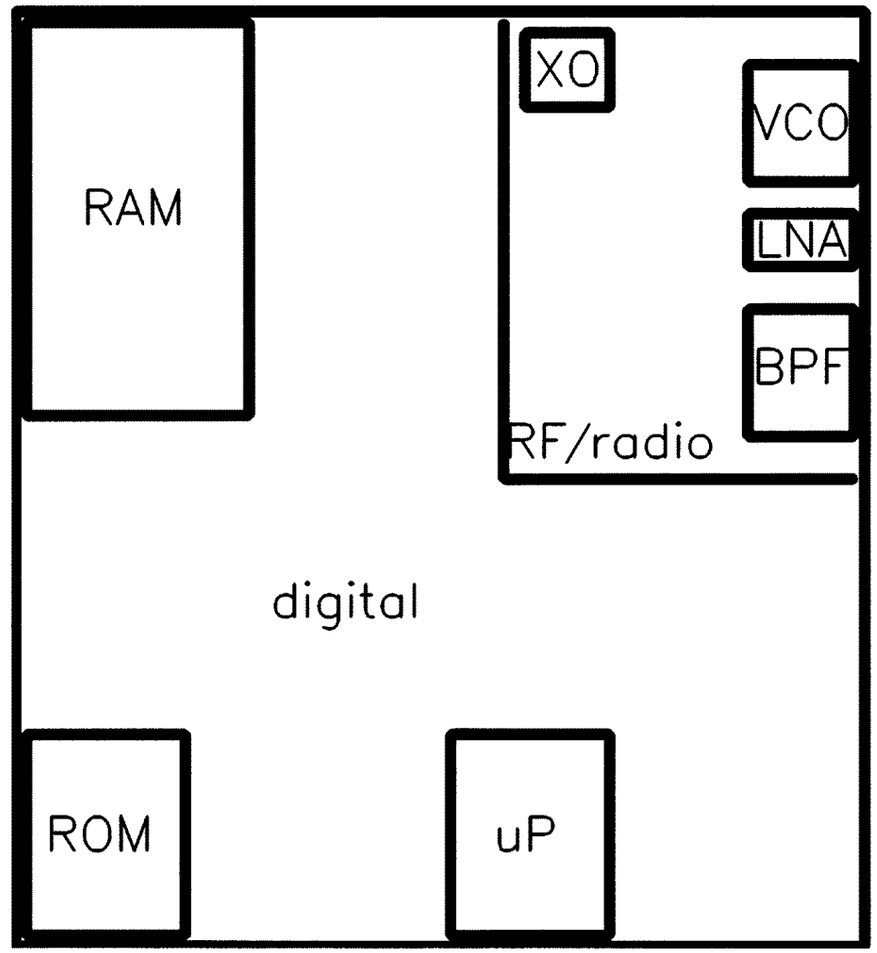
[IEEE P1500 Architecture Task Force, 1999]

# Are these “SoCs” really full systems?

What about the analog parts?

# Bluetooth transceiver



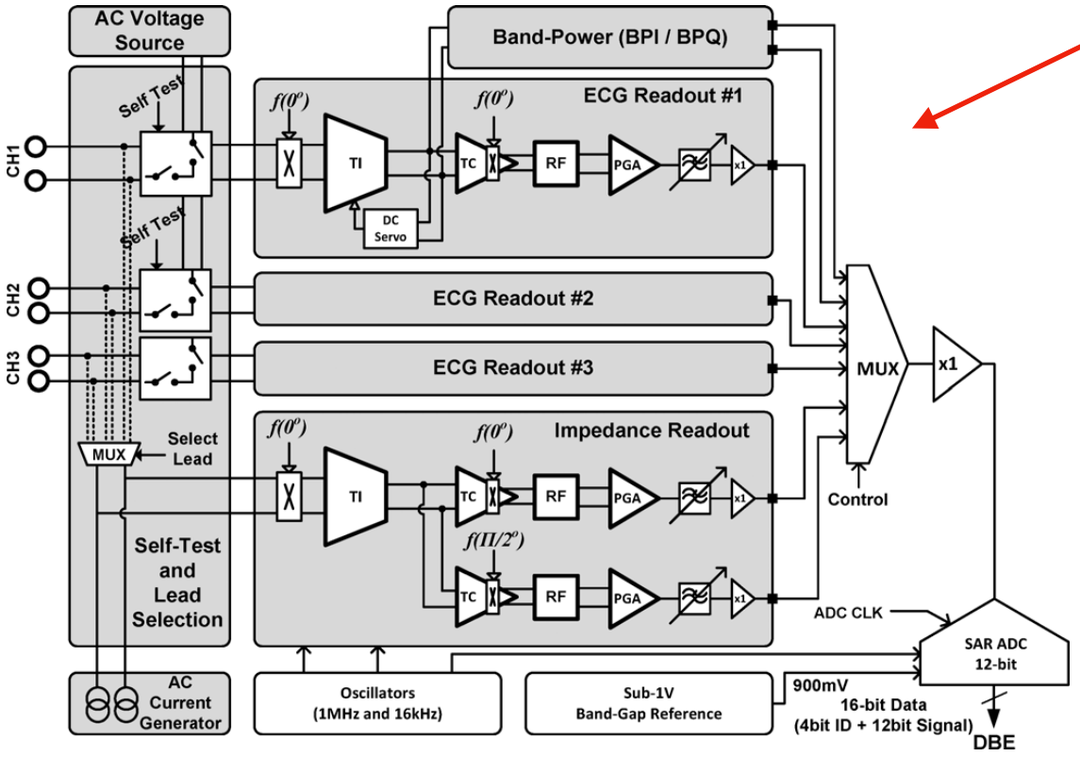


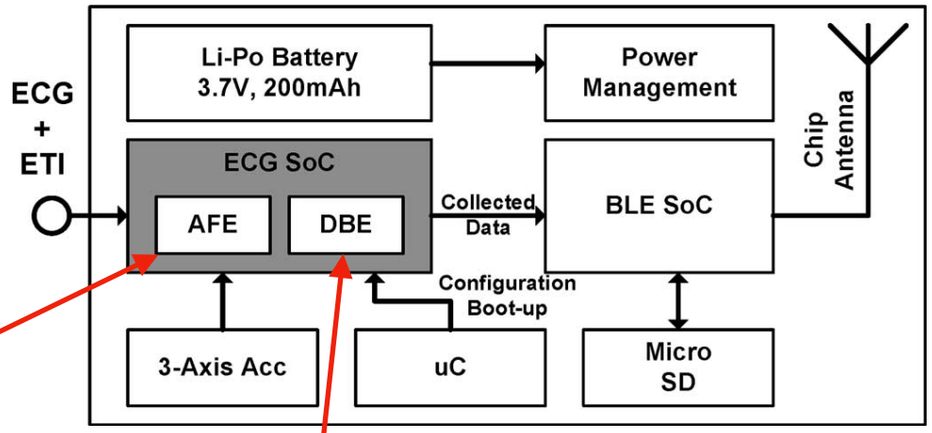

- 180nm CMOS, 16 mm², >1M digital gates, 2.4 GHz radio front end
- Digital and analog parts designed separately and combined at layout
  - 300um “moat” around the RF to attenuate switching noise

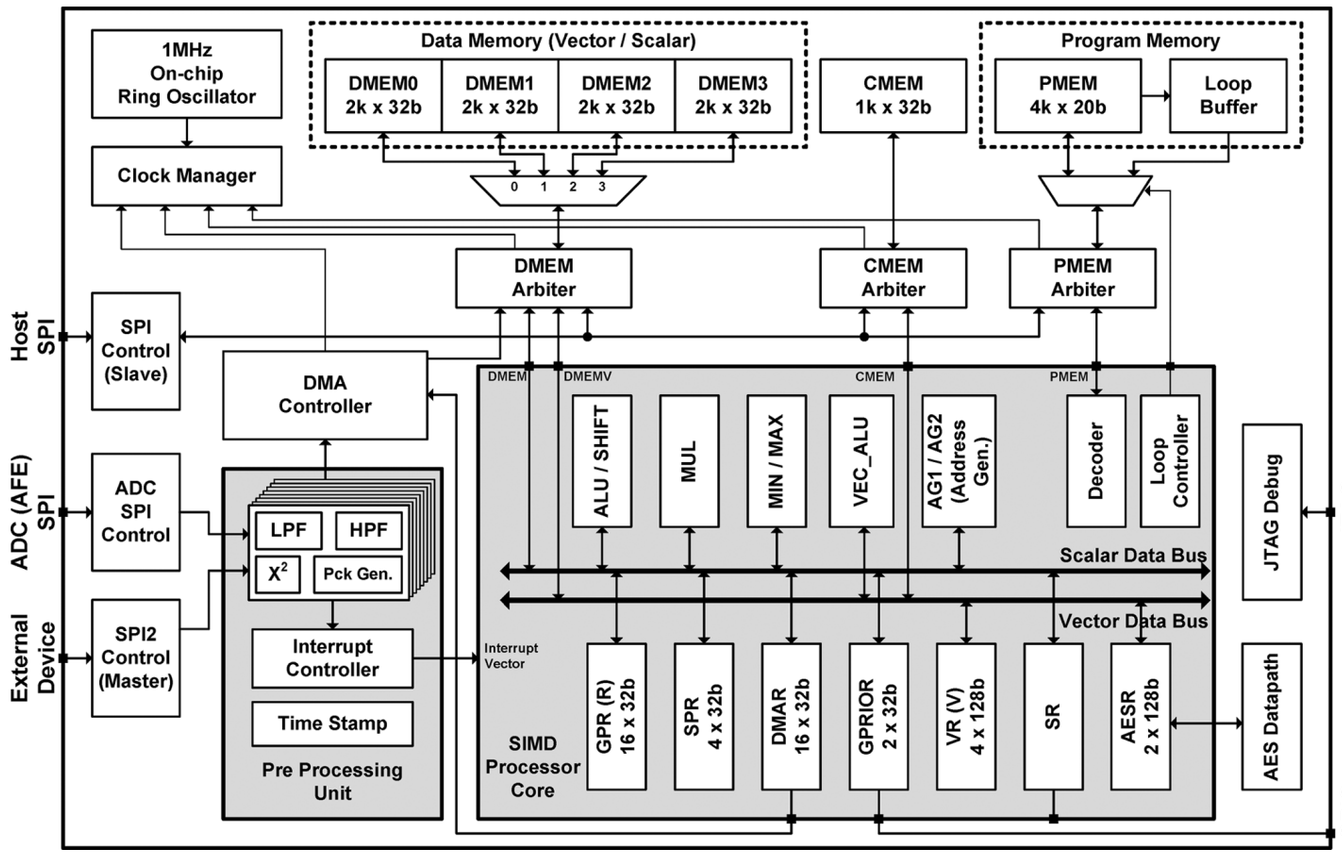


[P. van Zeijl et al, JSSC Dec 2002]

# ECG monitor



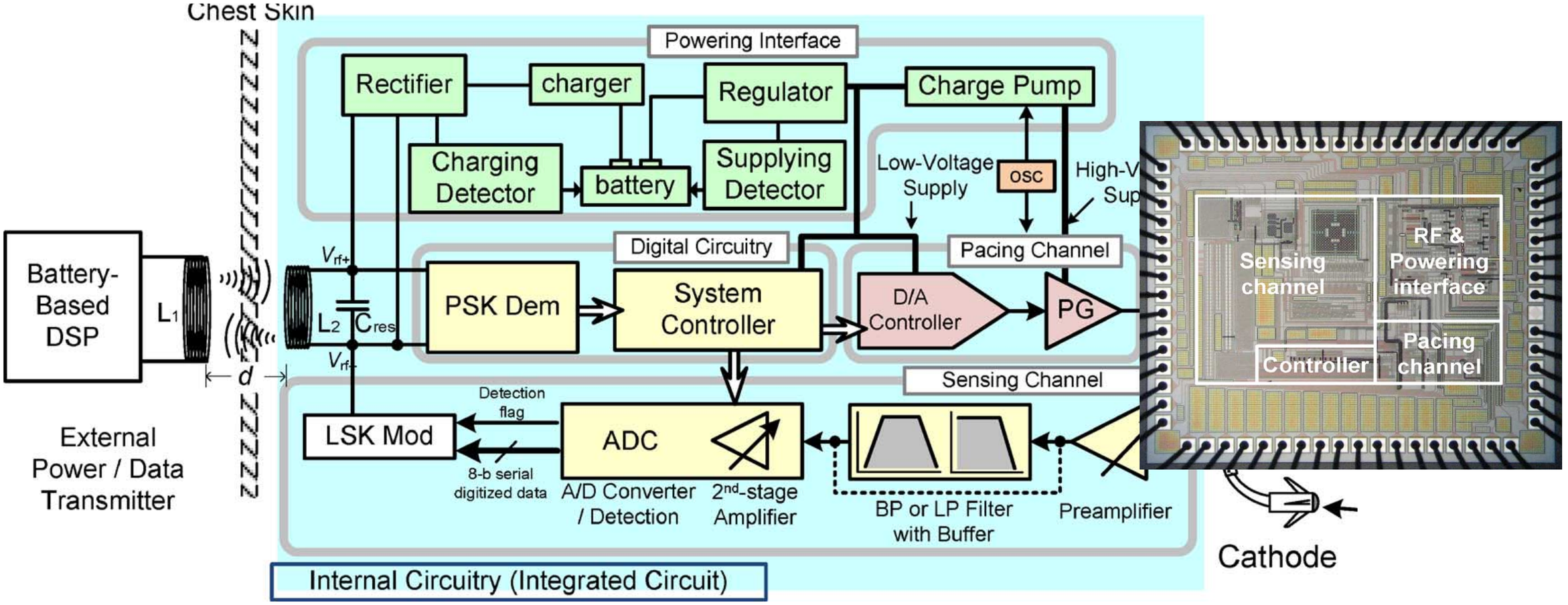




- 180nm CMOS, 12-bit ADC @ 512 S/s, 32 uW

[Kim et al. IEEE Trans. on Biomedical Circuits and Systems, Apr. 2014]

# Pacemaker



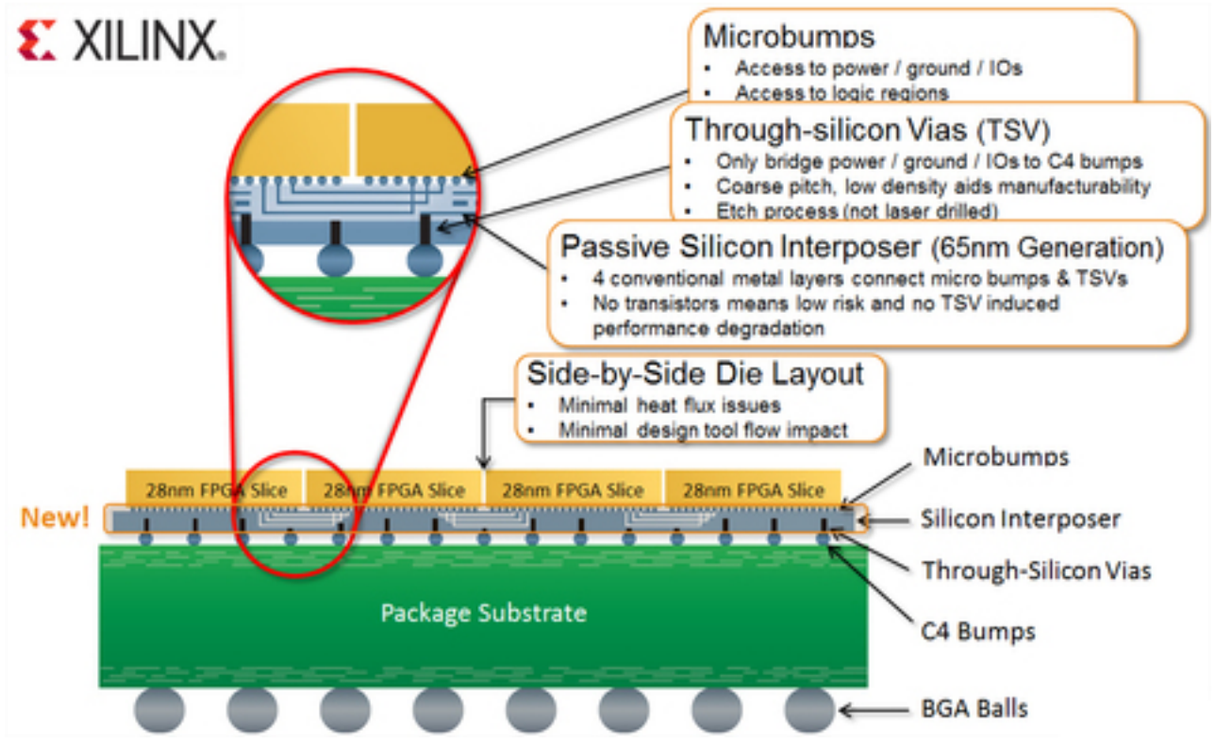

- Cardiac sensing and pacing, transcutaneous power supply
  - 350nm, 2.4mm$^2$, 48uW

[Lee et al. IEEE Trans. on Biomedical Circuits and Systems, Dec. 2011]

# Summary

- Systems-on-Chip offer unique integration benefits
  - Size, performance, power, reliability, cost (provided large volumes)
  - Design process includes extra complications
  - In practice similar to “Structured ASICs” of early 2000’s
    - Processors, memory, interfaces, interconnects
    - Adaptable to several similar applications

# Summary, cont.




- Don’t forget alternatives such as System-in-Package!
  - Several chips sharing physical enclosure
  - May be built in different technologies (CMOS, memory techs, GaAs, etc)
  - Chips may be tested separately
- Used today for large FPGAs, telecom applications, etc

[xilinx.com]