DAT093:
Summary and conclusion

lars.svensson@chalmers.se

What did you learn?

Course aims

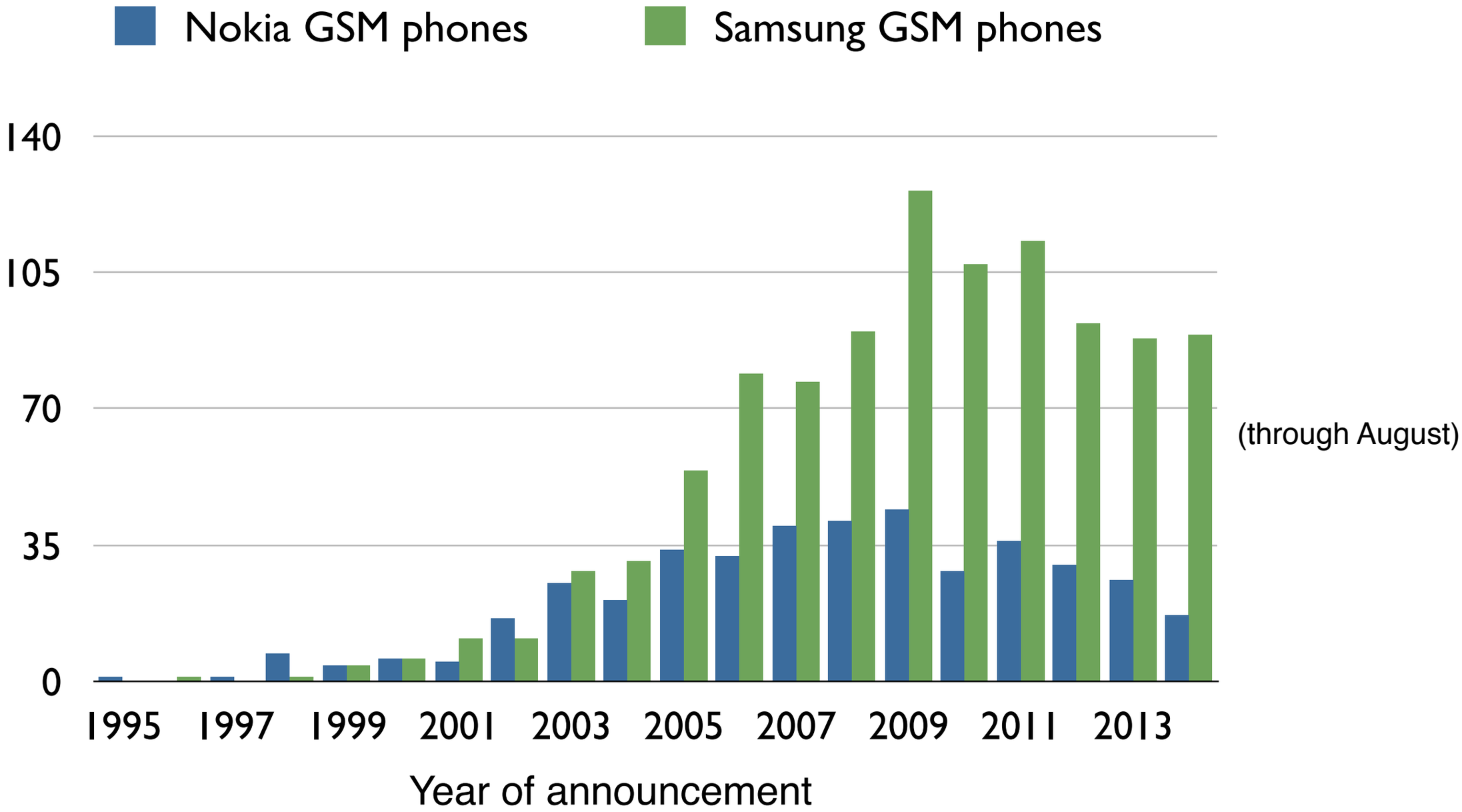
- Introduction / overview of electronic system design
- VHDL primer / brush-up
- Signal processing sub-system implementation
- Focus today on first point above

Modern electronic system design

- Ever-increasing complexity

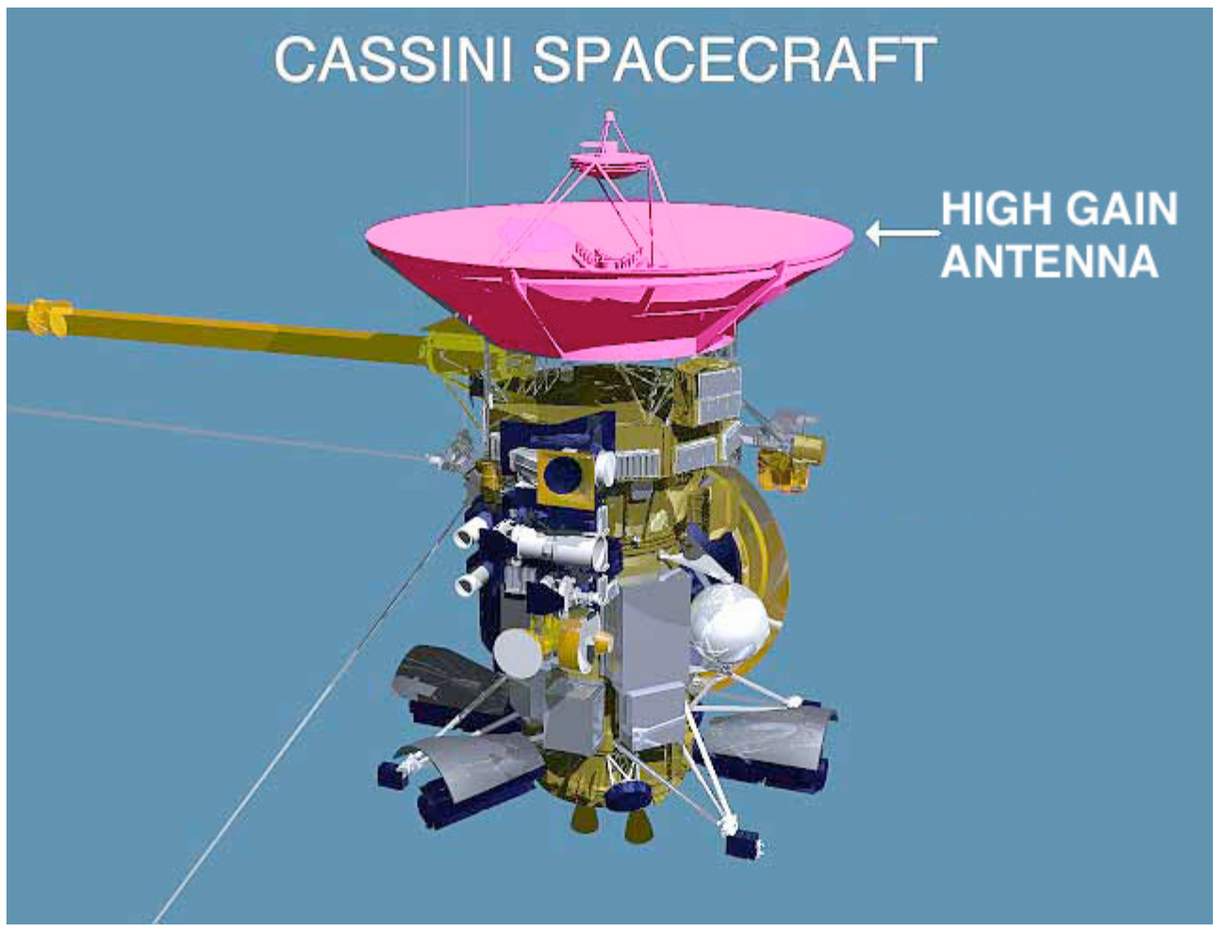
1. Market requirements
2. Range of end products
3. Subsystems
4. Toolsets
5. Implementation technologies

1. Market requirements



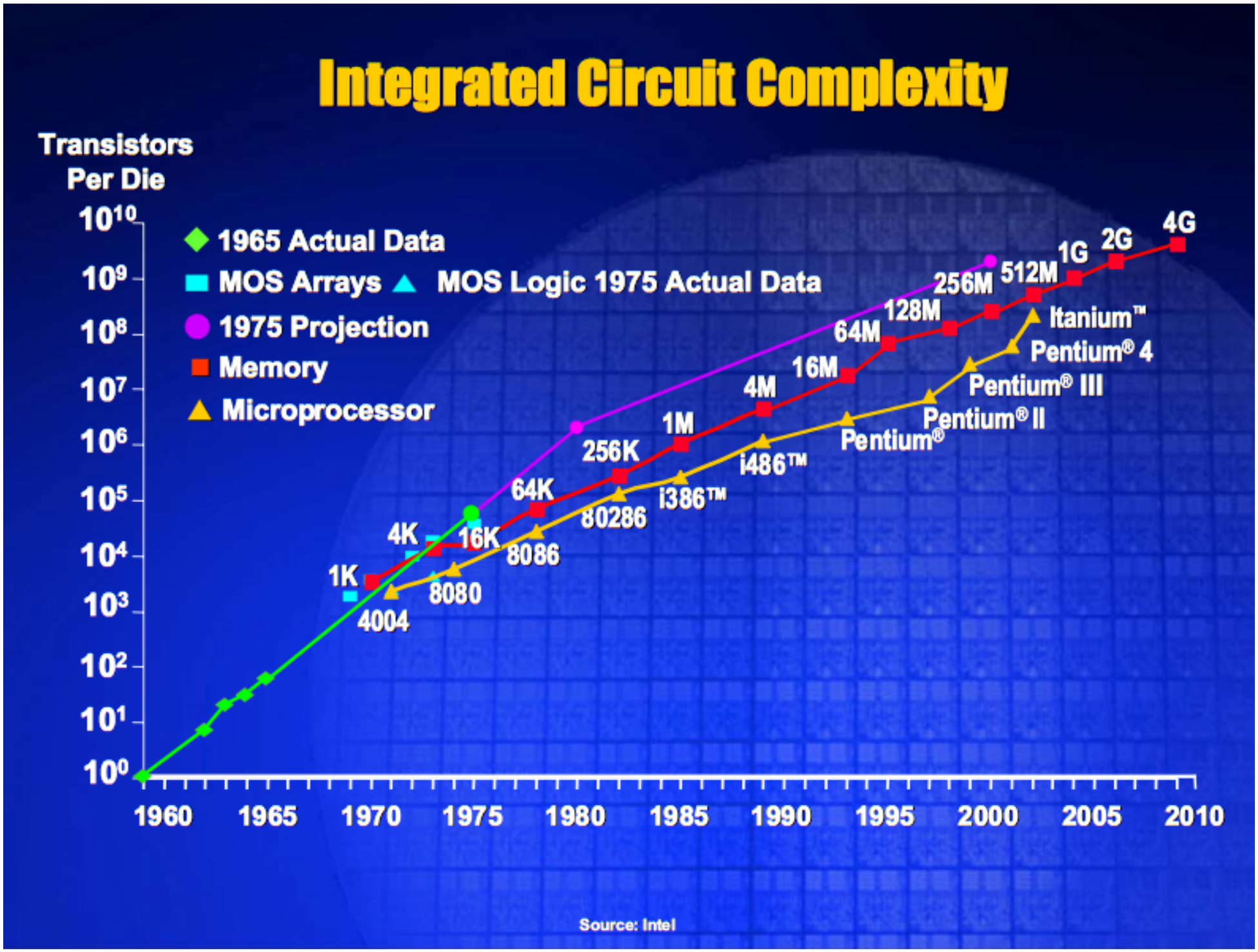
www.gsmarena.com

2. Range of end products



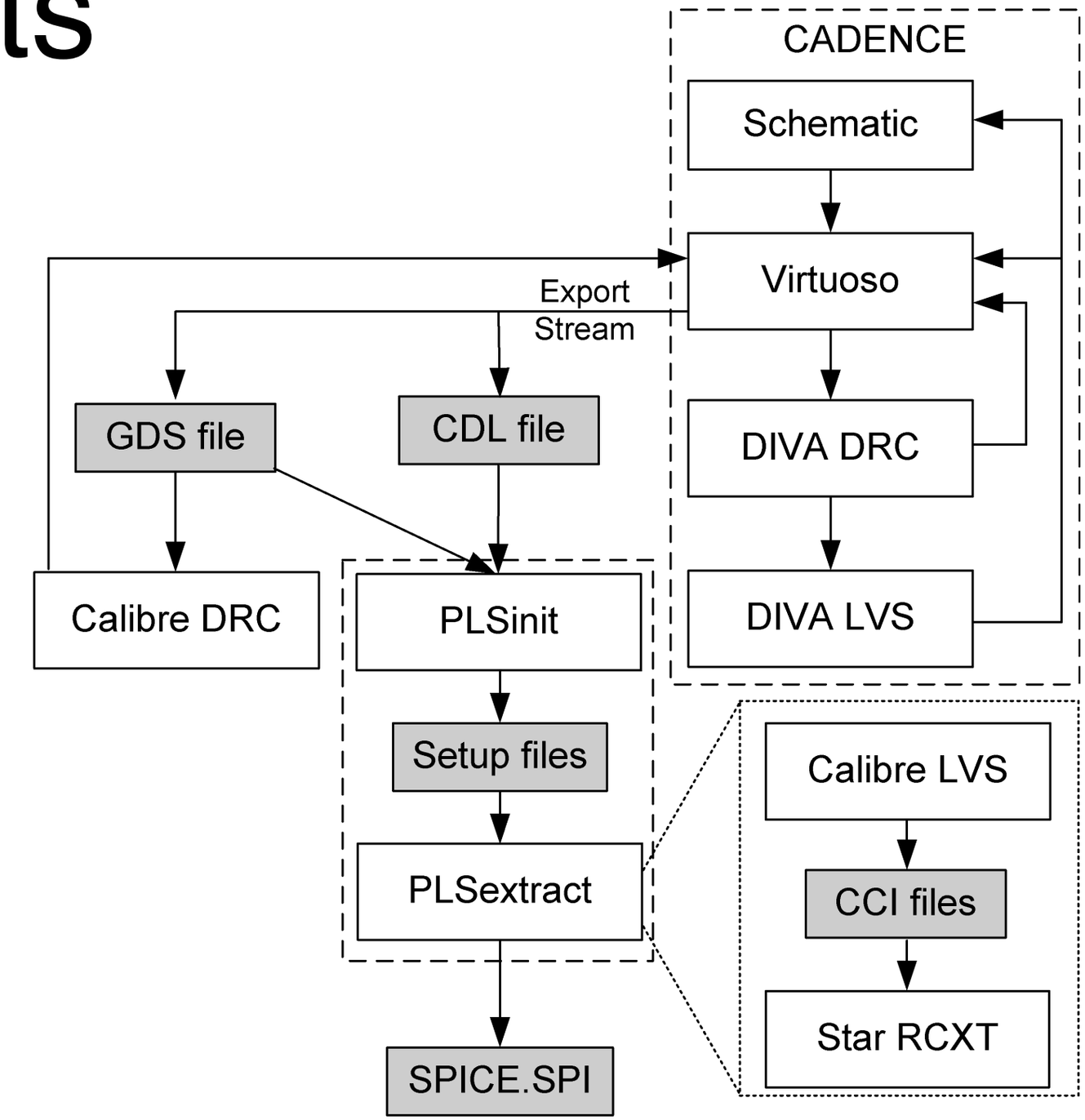
Different challenges!

3. Subsystems

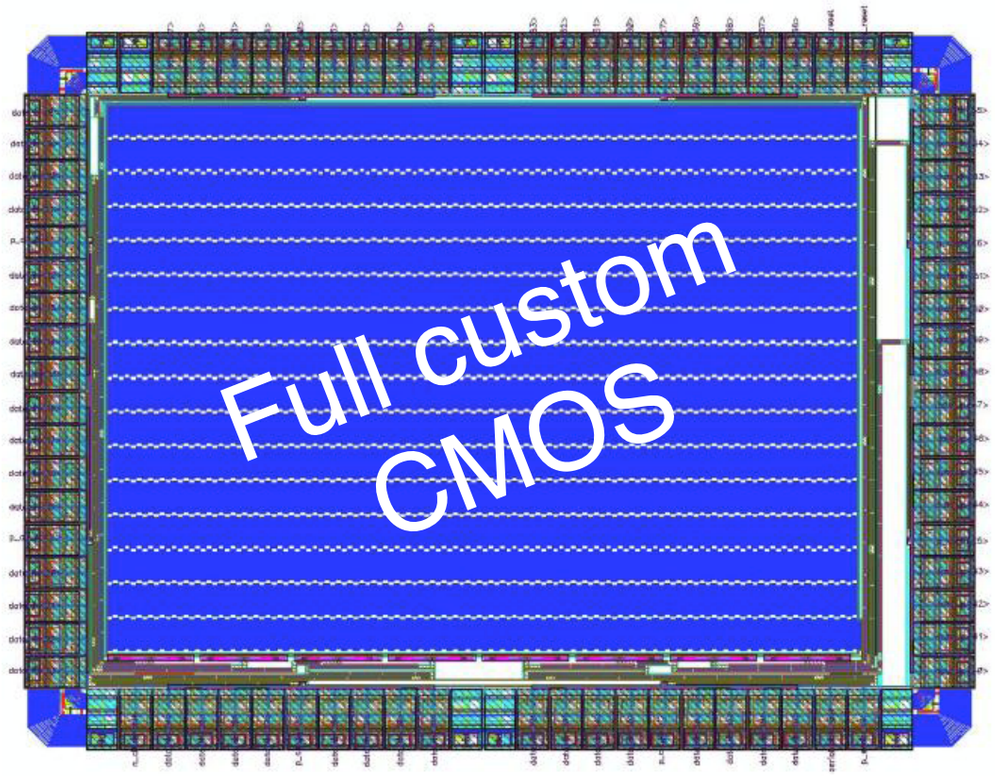


4. Toolsets

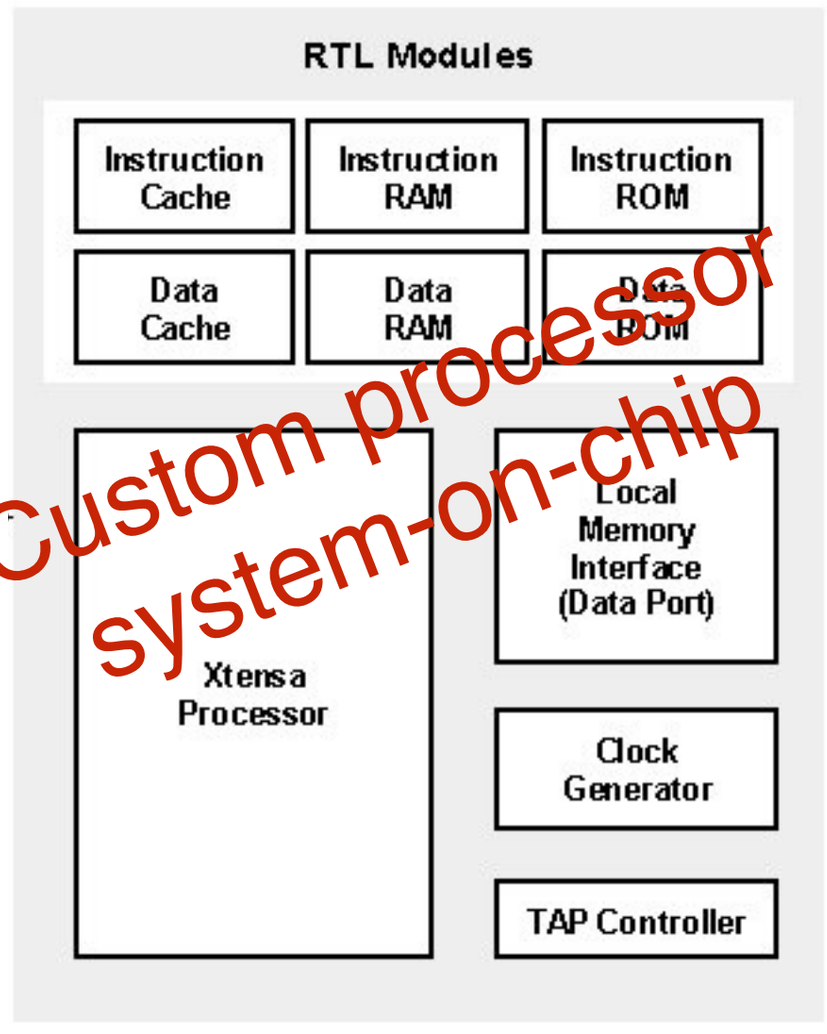
- ST 0.12 um, full-custom only
- (Much) increased complexity for more advanced processes
- Industrial projects: dedicated teams for toolset maintenance



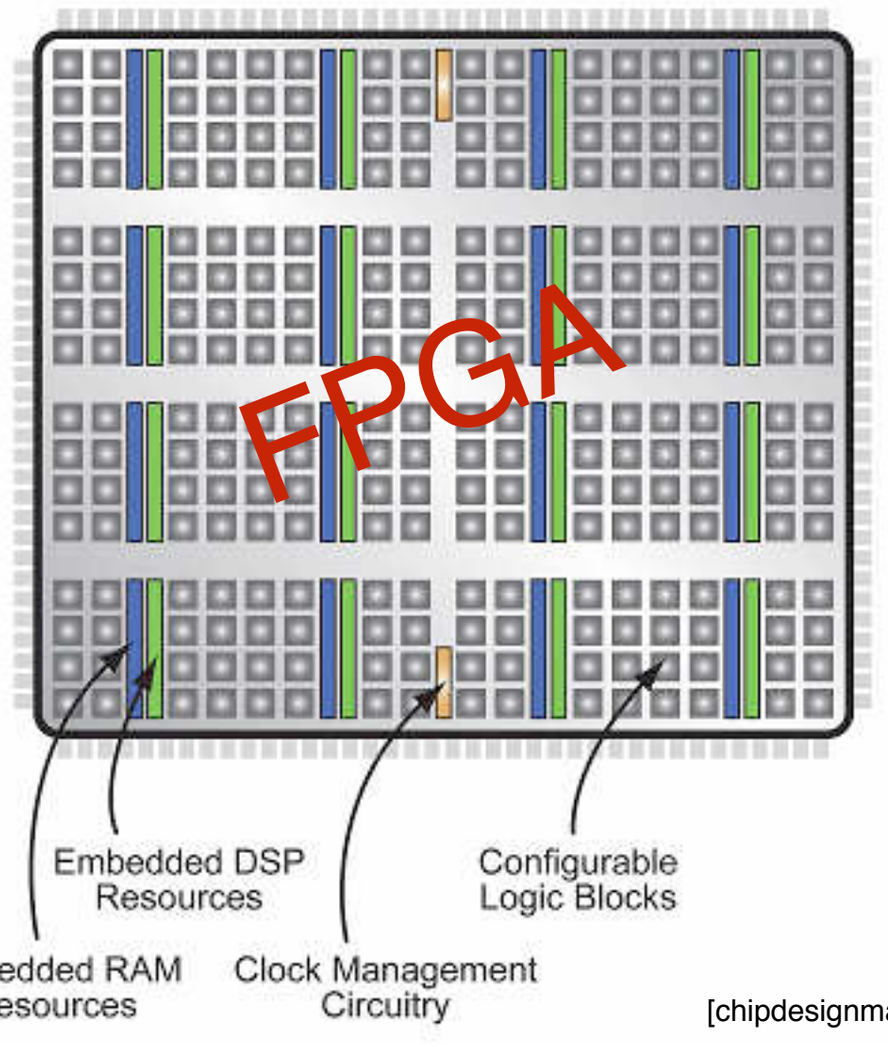
5. Implementation technologies



[Erik Ryman]



[eetimes.com]



[chipdesignmag.com]

How will the designer survive?

Survival skills:

- Vertical competence, broad insight
  - Specialize in one or a few areas
    - Embedded software
      - …
      - Logic and circuit design
- Team interactions
- Development process

Manage your work!

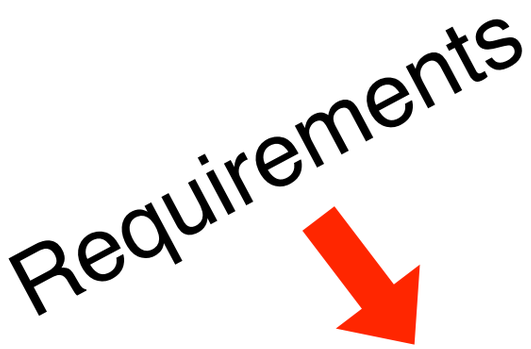
- Development process
  - Team process
  - Personal process

Internal view

External view

- Project structure
  - Predictability
    - Time
    - Money

Evolved version



Write specification — Full detailed specification!

Design architecture — Hardware + software

Decompose — Distribute requirements

"Design" means *recurse*

Select or design

Verify that reqs are met

Integrate

Remaining issues?

$$:-)

Deliver

Decompose + select/design

- Distribute requirements
  - What goes where? (HW/SW, etc)
    - May be steered by benchmarking
- Select or design; if the latter,
  - choose technology platform, and
  - do detailed design

Technology platform alternatives

- Analog / digital?
- Complex behavior?
  - ➡ Software on processor
- High performance, bad fit to processor? Low volume?
  - ➡ FPGA
- High volume, strict performance / power requirements?
  - ➡ Cell-library-based ASIC
- Toughest performance / power requirements?
  - ➡ Custom-design chip

Non-functional requirements

- Power, real-time response, fault tolerance
- Often “cross-cutting concerns”
  - Cannot be assigned to single sub-design!
  - Considered at all abstraction levels
- Tool support is less developed today
  - …but improving

Select or design?

- Buy what you can… (afford)
- Previous generation of system may be available
  - Re-use (most) parts
  - Old mistakes may be perpetuated… :-(
- Platform-based / platform-centric design
  - Plan for re-use across product/technology range/generations

Detailed design

- Analog + mixed: other courses
- Digital hardware development
  - Language-based
    - Heavy use of CAD toolchain
    - Similar for FPGAs, cell-based ASICs
- Software development
  - (Many) other courses…

Verify / integrate

- Verify against specification
  - Test suites; may revisit benchmark suites
- Test benches
  - Same HDL as for design, or special-purpose language; and/or
  - Formal verification techniques
- Debugging
  - Deviations from intended behavior cause Engineering Change Orders (ECOs)

Abstraction levels

- System-level design should focus on system-level questions
  - Big decisions, not the detailed ones
- VHDL less than ideal at system level
  - Detailed decisions too early
    - Example: clocking
- MATLAB; C, C++; UML; others

Hardware + software

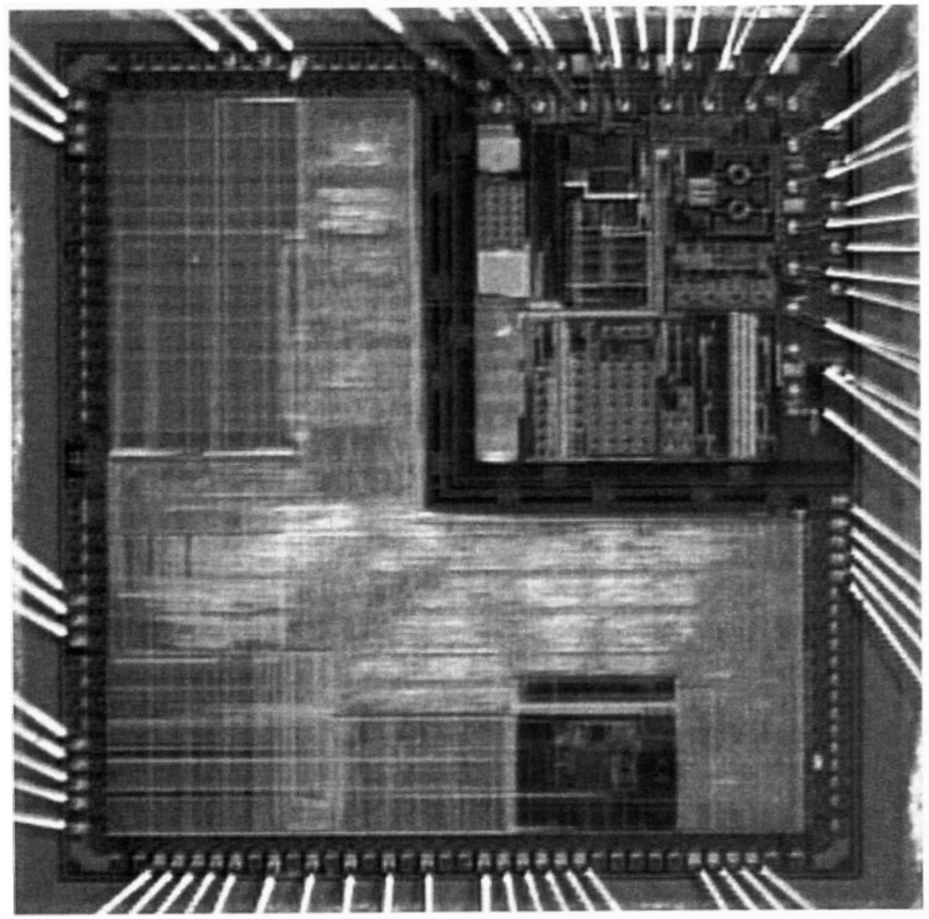
- What should come first: chicken or egg?
  - Write software assuming certain hardware
  - Design hardware assuming certain software
    - Design each towards interface
    - Co-design hardware and software
    - …but rarely a pure case of either
- Benchmarks, simulation

Advanced / Future tech

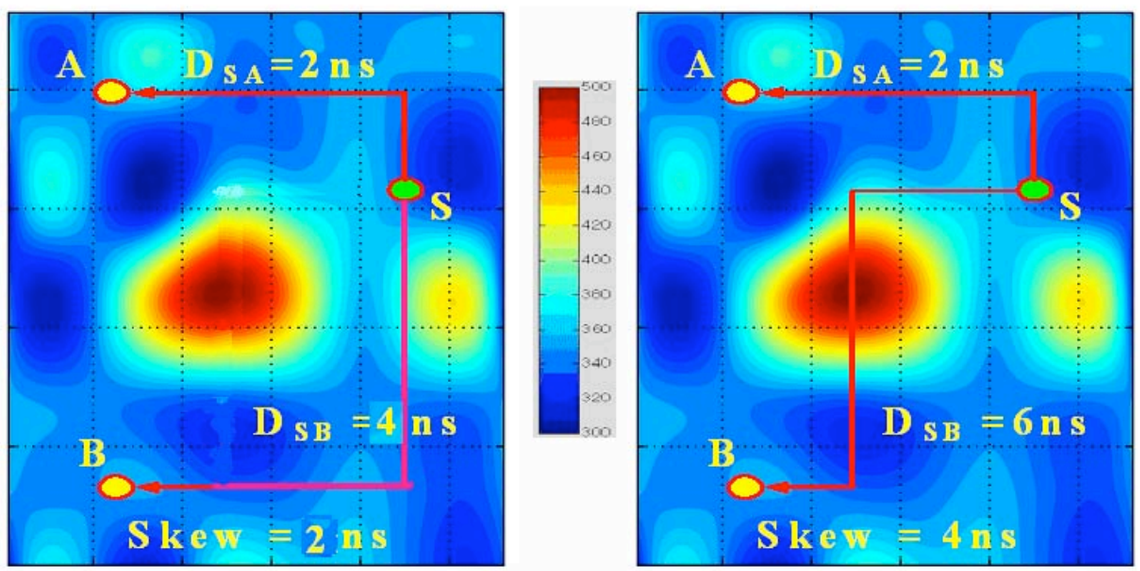
- Increasing complexity forces design on higher abstraction levels
- End of Moore; what next?
  - 3D integration
    - Power dissipation / heat removal
  - “More than Moore”: silicon++
    - Graphene? Carbon nanotubes? (What else? Who knows?)
- Major new design concern (+ all the old ones):
  - Variability

Future: more cross-cutting issues!

- Examples:
  - Supply noise
    - Digital radio transceivers
  - Power and heating
    - Routing example



van Zeijl et al. JSSC Dec 2002, pp. 1679 – 1687.



Liu et al. ICCD 2008, pp. 107–113.

Wrap-up

Labs

- Extra session Fri Oct 26 (8–12) + Mon Oct 29 (8–12)
  - TAs available to pass you on remaining labs
  - Hardware labs reviewed until Oct 29 session
    - Earlier labs until Nov 1 @ 16:00
- O/w wait until after period 2
  - May arrange unattended lab time if required
- Lab 6 report deadline: Nov 5
  - Submit in PingPong

Lab 6 report?

- Show that you have understood what you have done during lab 6 (and 5)
- Focus on the “Why” and the “How”, not on the “What”
- Individual submission
- Suggested length: 3–4 pages
- “Informal” document

Sit-down exam

- Written exam, no literature allowed; digital…
- Covers lecture content, incl. guest lecture, and reading material
- No problem solving or VHDL coding
- 48 grading points max; 24/48 to pass
  - Up to 12 bonus grading points earned during course
  - >= 40 for grade 4; >= 50 for grade 5
- Passing rate over 80% last few years

Exam prep

- Exam on Saturday, Oct 27, 8:30 – 12:30
  - Exam halls in SB building
  - Be there in time to set up computers etc!
- Old exams are up on PingPong, w/ solutions
  - Note: older exams had other grade limits (30/60)
  - Note: Some problems may refer to guest lectures with no correspondence this year!
- Lars: office hours tomorrow 13:15 – 14:00
  - Also email (recall tag!)

Questions?