VHDL and System-on-Chip

IAX8165 - 5.0 AP 5 3-2-0 E

http://mini.li.ttu.ee/~lrv/IAX8165/

Peeter Ellervee

IT226 620 2258 511 3631 LRV@cc.ttu.ee
http://www.ttu.ee/users/lrv/ http://mini.li.ttu.ee/~lrv/

Course plan

- **Introduction, myths. [3h]**
- **Hardware description languages [8 lectures]**
  - **VHDL. [2*3h]**
  - **Modelling of discrete systems. [3h]**
  - **VHDL and synthesis. [3h]**
  - **Verilog HDL. [2*3h]**
  - **Other HDLs. SystemC, VHDL-AMS - overview. [2*3h]**
- **Digital hardware synthesis [6 lectures]**
  - **Design methodologies. Physical, logic, and register-transfer level syntheses. [3h]**
  - **High-level synthesis - scheduling, allocation, and binding. [2*3h]**
  - **Local timing transformations. System level synthesis. [3h]**
  - **System-on-Chip design. [2*3h]**
- **Hands-on exercises [7*4h]**
  - **Timeslots at every Wednesday**

Textbooks

- Michael John Sebastian Smith, “Application-Specific Integrated Circuits.” [1997]
  - http://www.edacafe.com/books/ASIC/ASICs.php
- Dirk Jansen et al. (editors), “The electronic design automation handbook.” [2003]
- Ben Cohen, “VHDL coding styles and methodologies: [... an in-depth tutorial].” [1995]
- Stefan Sjoholm and Lennart Lindh, “VHDL for designers.” [1997]
- Ken Coffman, “Real world FPGA design with Verilog.” [2000]
- Donald E. Thomas, Philip R. Moorby, “The Verilog hardware description language.” [1996]
- Douglas J. Smith, “HDL chip design: A practical guide for designing, synthesizing and simulating ASICs and FPGAs using VHDL or Verilog.” [1997]
- John F. Wakerly, “Digital Design: Principles and Practices.” [2006]
- M. Morris Mano, “Digital Design.” [2002]
- Daniel D. Gajski, “Principles of Digital Design.” [1997]
- Giovanni De Micheli, “Synthesis and Optimization of Digital Circuits,” [1994]

Motivation

- System-on-Chip (SoC) requires new design methodologies
  - to increase designers productivity
  - to get products faster into market
- There exists a demand for efficient design methodologies at higher abstraction levels
- A different thinking needed from the designers
- At higher abstraction levels
  - a designer has much wider selection of possible decisions
  - each of these decisions has also a stronger impact onto the quality of the final design

Optimizations

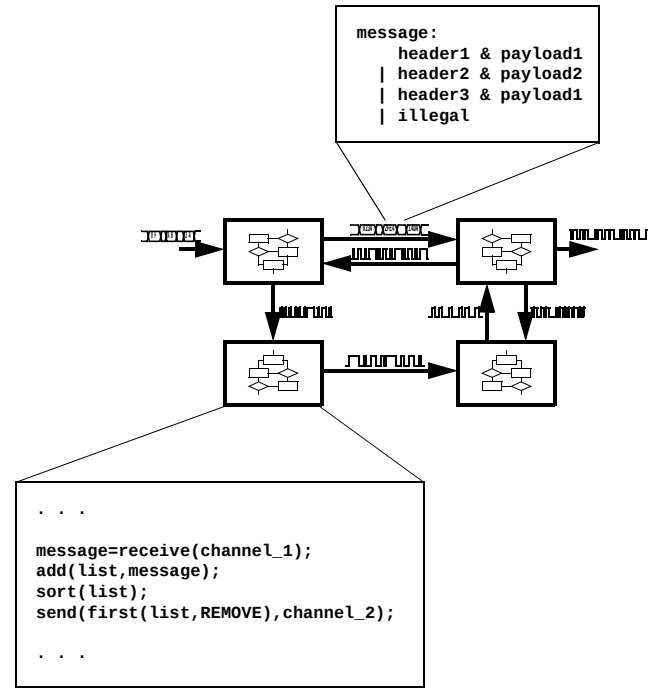
- Optimizations at logic level
  - thousands of nodes (gates) can exist
  - only few possible ways exist how to map an abstract gate onto physical gate from target library
  - optimization algorithms can take into account only few of the neighbors
- Optimizations at register transfer level (RTL)
  - handle hundreds of nodes exist (adders, registers, etc.)
  - there are tens of possibilities how to implement a single module
- At higher levels, e.g. at system level
  - there are only tens of nodes to handle (to optimize)
  - there may exist hundreds of ways how to implement a single node
  - every possible decision affects much stronger the constraints put onto neighboring nodes thus significantly affecting the quality of the whole design

Decisions at higher abstraction levels

- Two major groups of decisions
  - selection of the right algorithm to solve a subtask
    - making transformations inside the algorithm, e.g. parallel versus sequential execution
    - affect primarily the final architecture of the chip
  - decisions about the data representation
    - e.g. floating point versus fixed point arithmetic, bit-width, precision.
- Selection of a certain algorithm puts additional constraints also onto the data representation
- Selecting a data representation narrows also the number of algorithms available

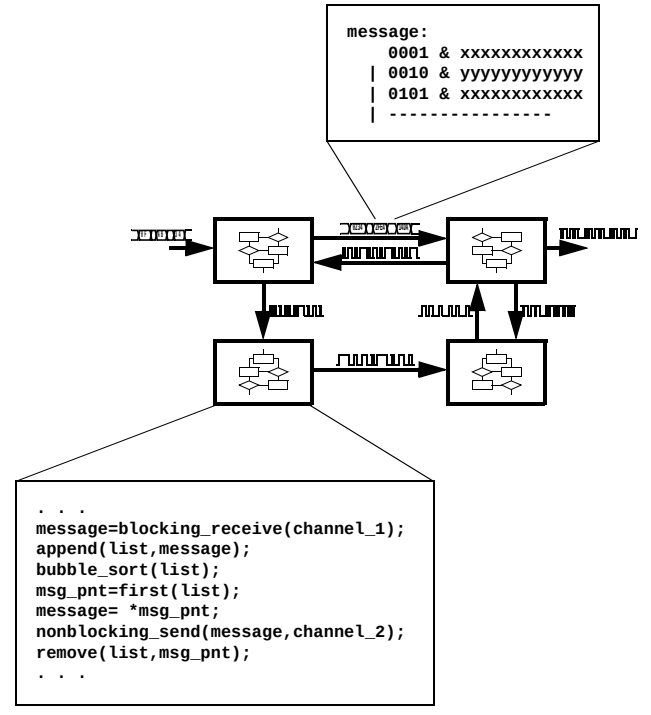
Abstraction levels

- System level
  - modules / methods
  - channels / protocols



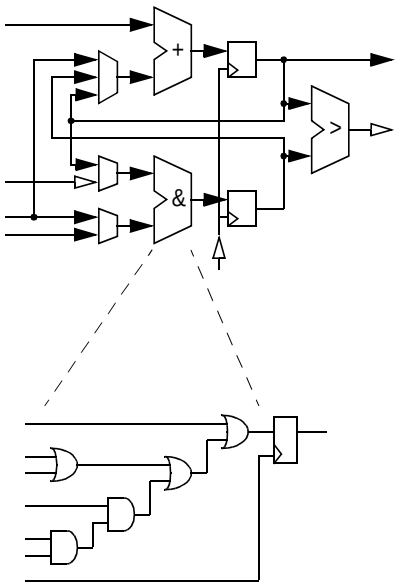
Abstraction levels

- Algorithmic level
  - (sub)modules / algorithms
  - buses / protocols



Abstraction levels

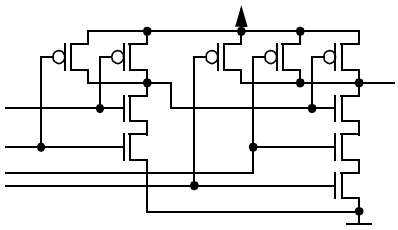
- **Register transfer (RT) level**
  - **blocks / logic expressions**
  - **buses / words**



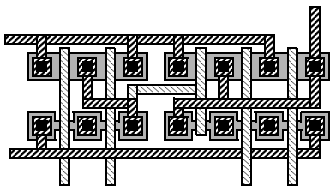
- **Logic level**
  - **logic gates / logic expressions**
  - **nets / bits**

Abstraction levels

- **Physical level**
  - **transistors / wires**

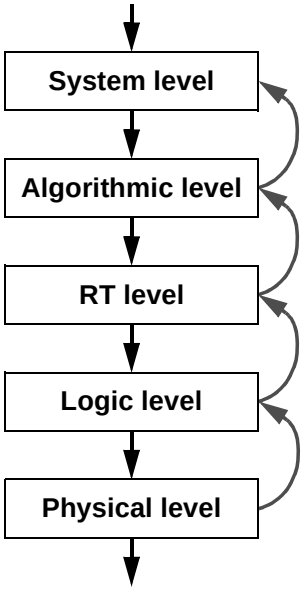


  - **polygons**

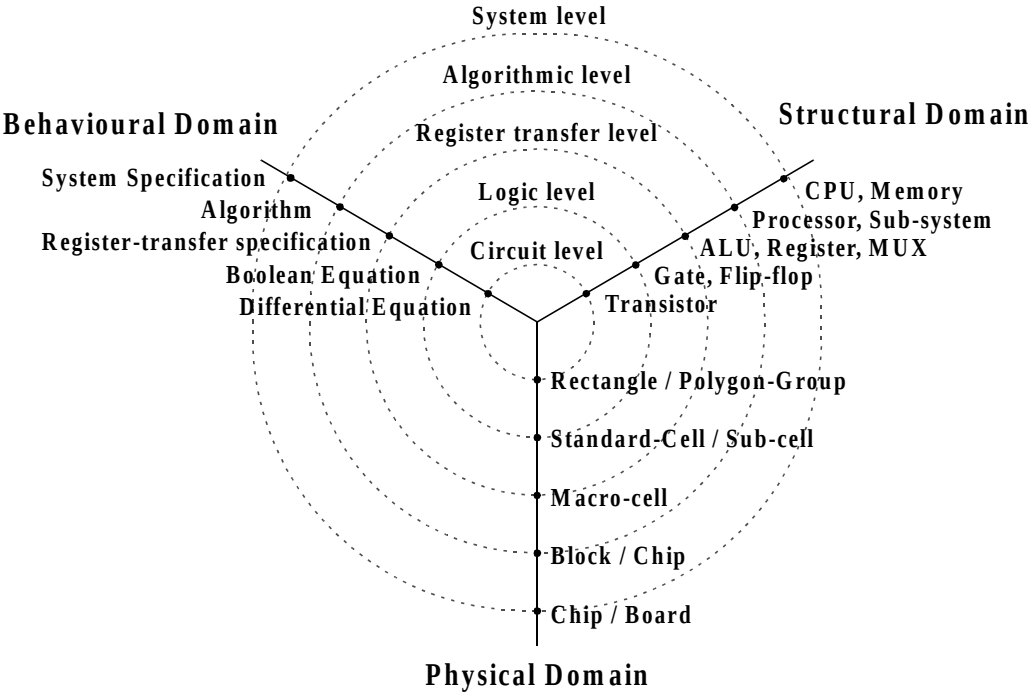


Design flow

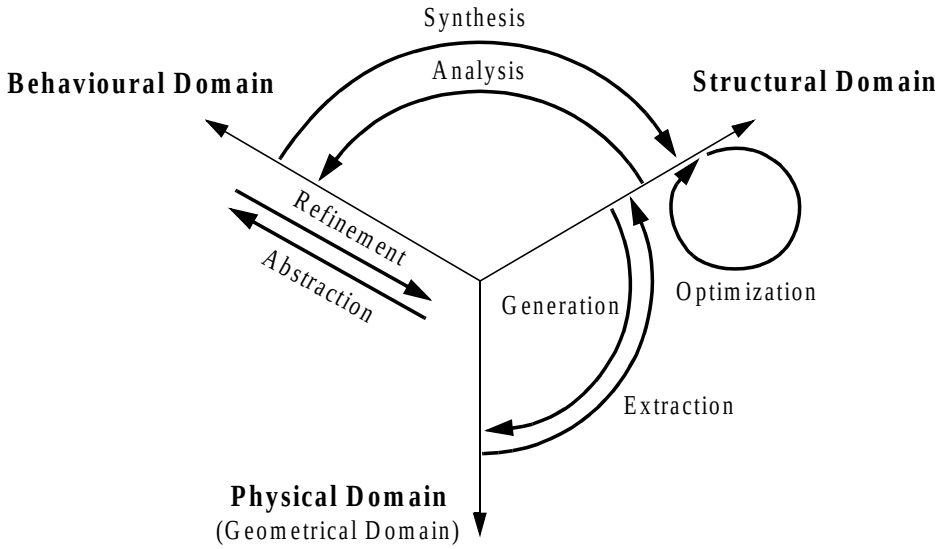
- **Specification refinement**
  - from higher to lower abstraction levels
  - refinement = transformations
- **Algorithm selection**
  - universal vs. specific
  - speed vs. memory consumption
- **Partitioning**
  - introducing structure
- **Technology mapping**
  - replacing Boolean equations with gates



Y-chart



Y-transformations



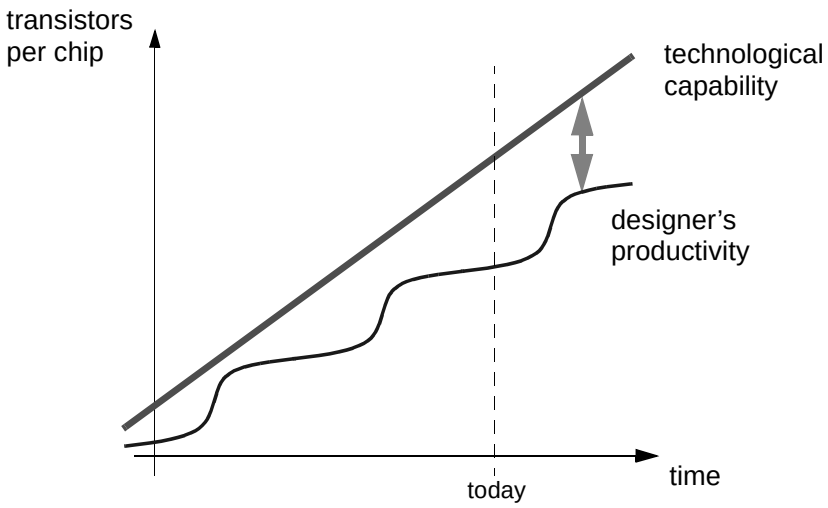
Design steps

- **System design**
  a.k.a. **Architectural-level synthesis**
  a.k.a. **High-level synthesis**
  a.k.a. **Structural synthesis**
  - **description / specification --> block diagram**
  - **determining the macroscopic structure, *i.e.*, interconnection of the main modules (blocks) and their functionality**
- **Logic design**
  - **block diagram --> logic gates**
  - **determining the microscopic structure, *i.e.*, interconnection of logic gates**
- **Physical design**
  a.k.a. **Geometrical-level synthesis**
  - **logic gates --> transistors, wires**

Synthesis -- levels and tasks

- System Level Synthesis
  - Clustering. Communication synthesis.
- High-Level Synthesis
  - Resource or time constrained scheduling
  - Resource allocation. Binding
- Register-Transfer Level Synthesis
  - Data-path synthesis. Controller synthesis
- Logic Level Synthesis
  - Logic minimization. Optimization, overhead removal
- Physical Level Synthesis
  - Library mapping. Placement. Routing

Synthesis -- design automatization



Design automation

- 1990 -- 4 Kgates / year / designer
- 1993 -- inhouse place and route -- 5.6K
- 1995 -- engineer (RTL-->GDSII) -- 9.1K
- 1997 -- small blocks reuse (2.5K-75K) -- 40K
- 1999 -- large blocks reuse (75K-1M) -- 56K
- 2001 -- synthesis (RTL-->GDSII) -- 91K
- 2003 -- intelligent testbench -- 125K
- 2005 -- behavioral and architectural levels, HW/SW (co)design -- 200K
- 2007 -- very large blocks reuse (>1M, IP cores) -- 600K

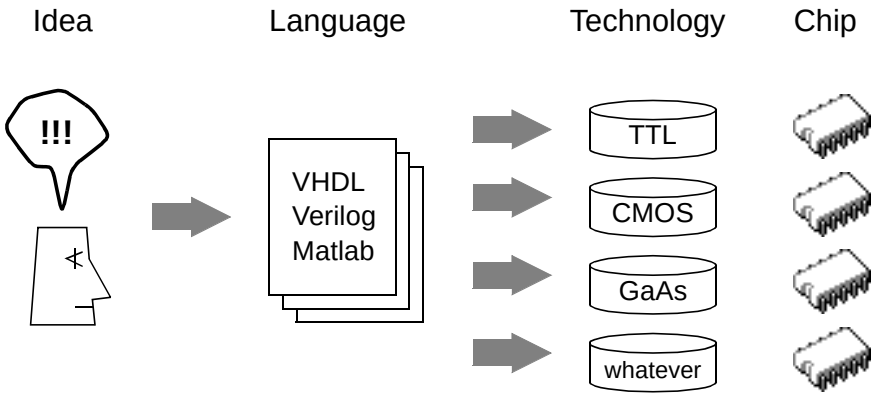
Market = $$$

- **Design cost**
  - ***design time*** **&** **chips** ***production*** **cost**
  - **huge** ***investments*** **(G$)**
  - **almost impossible to** ***correct***
- **High cost of modifications**
  - **large** ***production volumes*** **are more cost effective**
  - ***zero-defect*** **is very important**
  - **following** ***market trends*** **is important**
- **Price is inversely proportional to production volume**
  - **common purpose processors - cheap but not always usable**
  - **ASIC - application specific tuning (e.g. telecommunication)**
  - **prototypes - flexibility is extremely important in the development phase**
  - **special purpose chips (e.g. satellites)**
- **Reconfigurability**
  - **flexible products, possibility to modify working circuits**

Design criteria

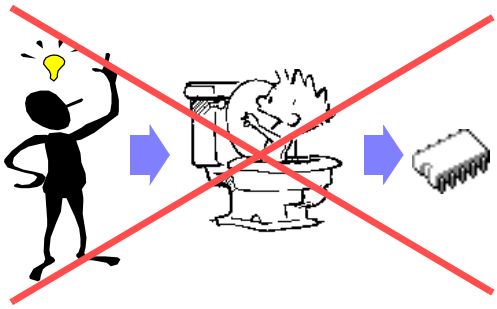
- Three dimensions - area, delay, power
  - size, speed, energy consumption
  - four dimensions - plus testability (reliability)
- Area
  - gates, wires, buses, etc.
- Delay
  - inside a module, between modules, etc.
- Power consumption
  - average, peak and total
- Optimizations
  - transferring from one dimension to another
  - design quality is measured by combined parameters, e.g., energy consumption per input sample

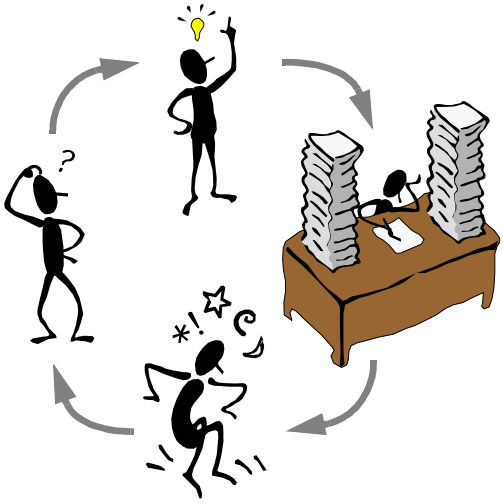
HDL - designing Systems-on-a-Chip (SoC)



MYTH #1

- High level design is a single pass

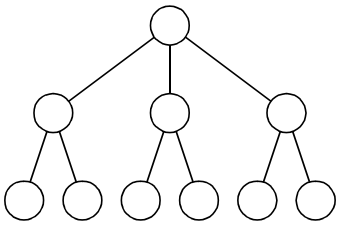


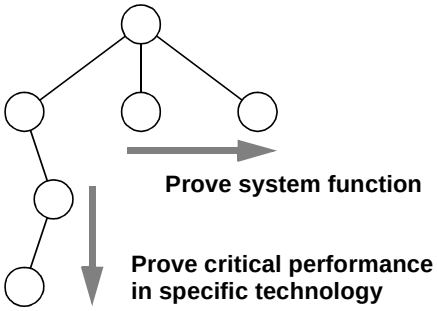


- Iterations needed
  - Functionality
  - Design goals

MYTH #2

- Top down design, in its purest form, works





- The pure breadth-first approach never actually works in practice
  - Bottom-up technology information must be considered early and often
  - Go depth-first for critical parts
  - Mix breadth-first with depth-first

MYTH #3

- You don't need to understand digital design anymore

- One must know hardware to get a good hardware
- Hope
  - Intimate knowledge of hardware is not necessary to design digital systems
- Fear
  - Using HDL based design methodology will turn them into software hackers
- Reality
  - High performance designs require a good deal of understanding about hardware
  - Designers must seed the synthesis tools with good starting points
  - Understanding the synthesis process is necessary to get good quality designs

MYTH #4

- Designer's job is just the functional specification now

- Specification = Functionality +
  Design goals +
  Operating conditions
- Schematic capture
  - Design goals and operating conditions were implicit (in designer's mind)
  - Implementation was chosen (modified) to meet goals and conditions
- HDL
  - Design goals (area, speed, power, etc.) are explicitly specified
  - Operating conditions (variations, loads, drives) are also explicitly specified

MYTH #5

- **Behavioral level is better than RTL**

- **Behavioral (a.k.a. algorithmic, high) level synthesis is not as mature as RTL (register-transfer level)**
- **If your specification includes enough timing information use RTL synthesis**
- **Behavioral constructs like *while*, *if-then-else* does not necessarily mean behavioral specification**
  - **... and/or can be misinterpreted**

Use of HDL --> Simulation

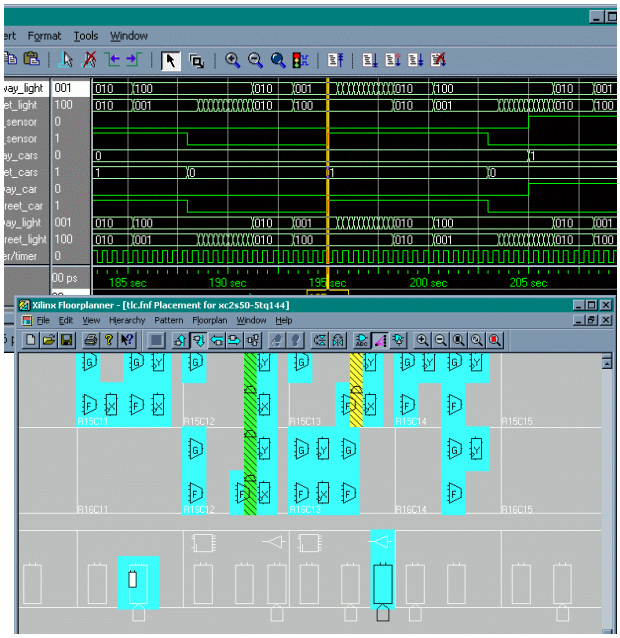
Simulation = modelling + analysis

- **Logic level simulation**
- **RT-level simulation**
- **Functional/behavioral level simulation**
- **System level simulation**

Design process today

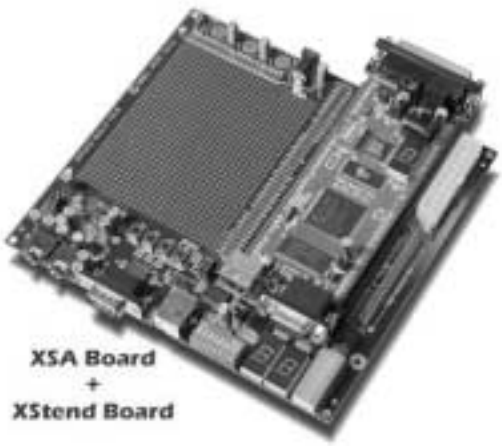
- **Hardware Description Language**

```
--
-- Highway is green, sidestreet is red.
--
if  sidestreet_car = NoCar  then
  wait until  sidestreet_car = Car;
end if;
-- Waiting for no more than 25 seconds ...
if  highway_car = Car  then
  wait until  highway_car = NoCar  for 25 sec;
end if;
-- ... and changing lights
highway_light <= GreenBlink;
wait for 3 sec;
highway_light <= Yellow;
sidestreet_light <= Yellow;
wait for 2 sec;
highway_light <= Red;
sidestreet_light <= Green;
```

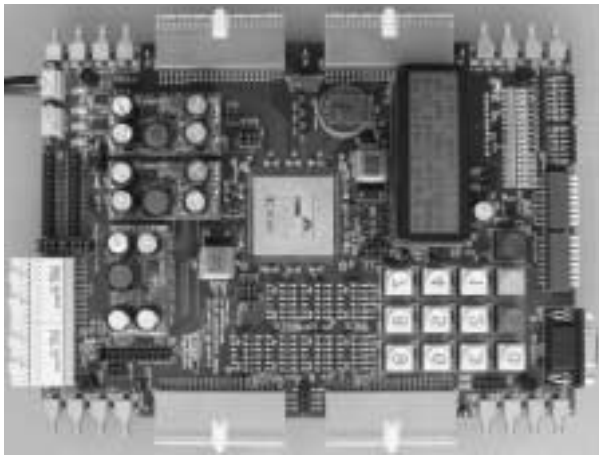


Prototyping

- **Possibility to check how a system works at conditions very close to the operating environment without the need to create expensive chips**

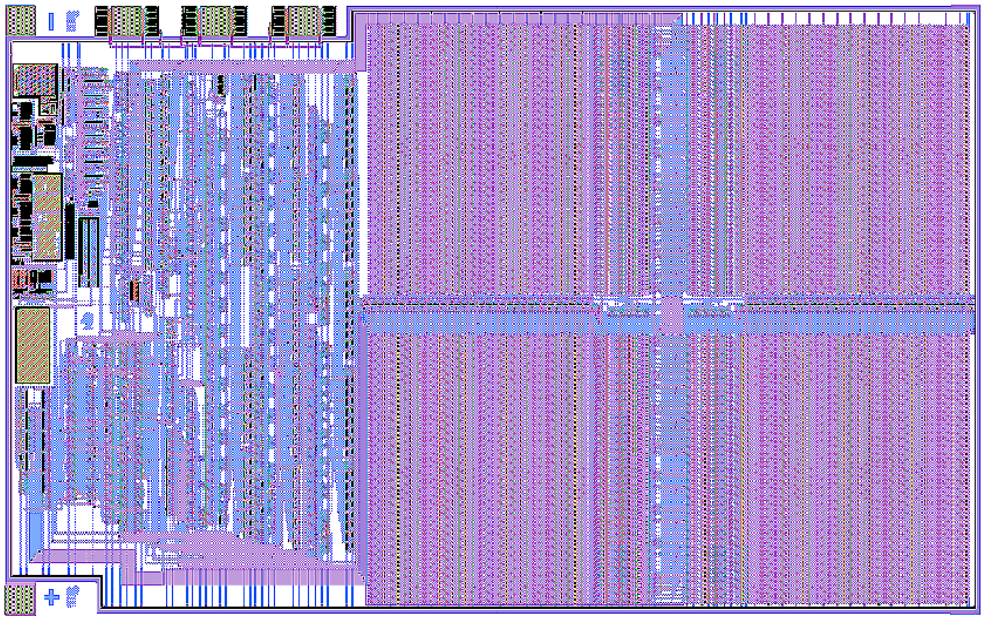


XESS Corp., www.xess.com, $400
[Xilinx Inc., FPGA XC2S100]



ErST Electronics, www.erst.ch, $3380
[Xilinx Inc., FPGA X2V2000]

Chip - the final result



Reality

- **Found On First Spin ICs/ASICs**
  - **Functional Logic Error - 43%**
  - **Analog Tuning Issue - 20%**
  - **Signal Integrity Issue - 17%**
  - **Clock Scheme Error - 14%**
  - **Reliability Issue - 12%**
  - **Mixed Signal Problem - 11%**
  - **Uses Too Much Power - 11%**
  - **Has Path(s) Too Slow - 10%**
  - **Has Path(s) Too Fast - 10%**
  - **IR Drop Issues - 7%**
  - **Firmware Error - 4%**
  - **Other Problem - 3%**
- **Overall 61% of New ICs/ASICs Require At Least One Re-Spin.**

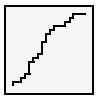
  - **Aart de Geus, Chairman & CEO of Synopsys, Boston SNUG keynote address, 9.09.2003**

Future?

2000		2010
2 Gbit	memory size	256 Gbit
$8 \cdot 10^6$	transistors per cm^2	$160 \cdot 10^6$
1.5 GHz	internal clock frequency	10 GHz
0.5 GHz	external / bus clock frequency	1.5 GHz
2000	pin count	6000
800 mm^2	chip area	1300 mm^2
140 nm	wire width	40 nm
1.5 V	supply voltage	0.6 V
100 W	power consumption	170 W
0.5 W	power consumption (batteries)	1.5 W

Problems

physical level	**quantum effects**
	noise
logic level	**crosstalk**
	speed of light
system level	**# of transistors**



10 mm --> (10^8 m/s) --> 10^{-10} s --> (10%) --> 1 GHz !?

- **GALS - globally asynchronous locally synchronous**
- **mixed signal - digital and analog circuits on the same chip**