Computer Aided Design of Electronics
[Datorstödd Elektronikkonstruktion]

Zebo Peng, Petru Eles, and Nima Aghaee
Embedded Systems Laboratory
IDA, Linköping University

www.ida.liu.se/~TDTS01

Electronic Systems
Objectives

- Basic principles of computer-aided design for electronic systems (Electronic Design Automation).
- Electronic system design at high levels of abstraction.
- Synthesis and optimization algorithms.
- Test and design for testability techniques.
- The hardware description language VHDL and its use in the design/synthesis process.

Course Organization

- 10 Lectures (Petru and Zebo):
  - Introduction and basic terminology.
  - VHDL: overview and simulation semantics.
  - Behavioral and structural modeling with VHDL.
  - High-level synthesis of digital systems.
  - Heuristics and optimization algorithms.
  - Testing and design for testability.
- Invited industrial lecture (Björn Fjellborg, Ericsson)
- Laboratory part (Nima):
  - Three seminars on assignments and CAD systems.
  - Lab assignments, for groups of two students.
Recommended Literature

  - Other VHDL books can also be used.
  - A VHDL Cookbook is available at the course website.
- Articles to be distributed, including:
  - High-level synthesis of digital circuits.
  - Optimization heuristics.
  - Design for test and built-in self-test.
- Lecture notes (www.ida.liu.se/~TDTS01).

Lecture I

- Trend in microelectronics
- The design challenges
- Different design paradigms
- The test problems
Moore’s Law

Number of transistors per chip would double every 1.5 years.

Similar improvement in:
- Clock Frequency (every 2 years)
- Performance
- Memory capacity

Moore’s Law in Action

Source: Prof. K Yelick, U.C. Berkeley
Intel Microprocessor Evolution

Images courtesy of Intel Corporation

Intel 4004
2.3 Thousands Transistors
10000 nm

System on Chip (SoC)

Hardware
Software

Microprocessor
Digital ASIC
Analog circuit
Sensor
Embedded memory
DSP
Network
High-speed electronics

Source: Stratus Computers


1,000

10,000

Source:
S3

Computers
Lecture I

- Trend in microelectronics
  - The design challenges
    - Different design paradigms
    - The test problems
Many Design Tasks

- System specification (functionality and requirements)
- Hardware/software trade-offs
- Architecture selection and exploration
- Synthesis and optimization
- Implementation
- Testing and design for testability
- Analysis and simulation
- Verification and validation
- Design management: storage of design data, cooperation between tools, design flow, etc.

Design Objectives

- **Unit cost**: the cost of manufacturing each copy of the system, excluding NRE cost.
- **NRE cost** (Non-Recurring Engineering cost): The one-time cost of designing the system.
- **Size**: the physical space required by the system.
- **Performance**: the execution time or throughput.
- **Power**: the amount of power consumed by the system.
- **Testability**: the easiness of testing the system to make sure that it works correctly.
- **Flexibility**: the ability to change the functionality of the system without incurring heavy NRE cost.
- **Correctness, safety, etc.**
Mixed Technologies for Electronics

- Embed in a single chip: Logic, Analog, DRAM blocks

- Other advanced technology blocks on a chip:
  - FPGA, Flash memory, RF/Microwave

- Beyond Electronic
  - MEMS (Micro Electro Mechanical Systems)
  - Optical elements
The Main Challenges

- System complexity
  - Increasing functionality and diversity
  - Increasing performance
- Stringent design requirements
  - Low cost and low power
  - Dependability: reliability, safety and security
  - Testability and flexibility
- Technology challenges for cost-efficient implementation
  - Deep submicron effects (e.g., cross talk and soft errors)
  - Issues related to process variation

Possible Solutions

- Powerful design methodology and CAD tools.
- Advanced architecture (modularity).
- Extensive design reuse.

Design Paradigm Shift
Lecture I

- Trend in microelectronics
- The design challenges
- Different design paradigms
- The test problems

Capture and Simulate

- The detailed design is captured in a model.
- The model is simulated.
- The results are used to guide the improvement of the design.
- All design decisions are made by the designers.
Abstraction Hierarchy

- Layout/silicon level — The physical layout of the integrated circuits is described.

- Circuit level — The detailed circuits of transistors, resistors, and capacitors are described.

- Logic (gate) level — The design is given as gates and their interconnections.

Abstraction Hierarchy (Cont’d)

- Register-transfer level (RTL) — Operations are described as transfers of values between registers.

- Algorithmic level — A system is described as a set of usually concurrent algorithms.

- System level — A system is described as a set of processors and communication channels.
Gajski’s Y-Chart

The Three Domains

- Behavioral domain — A component is described by defining its input/output functional relationship.
- Structural domain — A component is described in terms on an interconnection of more primitive components.
- Physical/geometrical domain — A component is described in terms of its physical placement and characteristics (e.g., shape).
A Typical Top-Down Design Process

Informal Specification

Behavioural Domain

Algorithms, processes
Register-Transfer Spec.
Boolean Eqn.
Transistor functions.

Structural Domain

System level
CPU, Memory, Bus

RT-level
ALU, Reg., MUX

Logic level
Gate, Flip-Flop

Circuit level
Transistor

Physical Domain

Transistor layout
Standard-Cell/Subcell
Macro-Cells/chips
Board, MCMs

Describe and Synthesize Paradigm

- Description of a design in terms of behavioral specification.
- Refinement of the design towards an implementation by adding structural details.
- Evaluation of the design in terms of a cost function and the design is optimized w.r.t. the cost function.

\[ o_1 = (a + b) \cdot c + d \cdot c; \]
\[ o_2 = (d + f) \cdot c; \]
\[ o_3 = (a + b) \cdot d + d \cdot f; \]
High-Level Describe and Synthesize

- Description of a design in terms of behavioral specification.
- Refinement of the design towards an implementation by adding structural details.
- Evaluation of the design in terms of a cost function and the design is optimized for the cost function.

For \( I = 0 \) To 2 Loop
Wait until clk'event and clk'=1';
If (rgb[I] < 248) Then
    P=rgb[I] mod 8;
...

Core-based Design

- Utilization of pre-designed and pre-verified cores:
  - Reuse of large IP blocks, such as CPU, DSP, memory modules, communication infrastructure, and analog blocks.
- Divide-and-conquer design methodology.
- Flexibility based on different core description levels:
  - Soft: RT level (synthesizable VHDL/Verilog).
  - Firm: Gate-level netlist.
  - Hard: Layout.
- Legal issues:
  - IP right protection.
  - Liability in case of failures.
Platform-based Design

- A platform is a partial design:
  - for a particular application area;
  - includes embedded processor(s);
  - may include embedded software;
  - customizable to specific requirements.

- A platform captures the good solutions to the important design challenges in a given application area.

- A method for design re-use at all abstraction levels based on assembling and configuring platform components in a rapid and reliable fashion.
  - It reuses architectures.

Platform-based Design Steps

- Design the platform.
  - A highly configurable system architecture.
  - Optimize for performance, power, etc.
  - Useful for a set of applications.
  - Tools to explore the different configurations.

- Use the platform.
  - Modify hardware components for a particular customer’s needs.
  - Optimize the software.
  - HW/SW integration and test.
Lecture I

- Trend in microelectronics
- The design challenges
- Different design paradigms
- The test problems

Testing and its Current Practice

- Testing aims at the detection of physical faults (production errors/defects and physical failures).
- Different from the design task, testing is performed on each individual chip, after it is manufactured (volume sensitive).
- The common approach to perform testing is to utilize an Automatic Test Equipment (ATE).

Automatic Test Equipment
Testing of Mixed Technologies

How to test the mixed chip?

- Use multiple ATEs: Logic ATE, Memory ATE, Analog ATE, etc.
  - Usually time consuming, due to handling time.
- Employ a super ATE with combined capabilities.
  - Usually very expensive.

High Test Complexity

- # of transistors increases exponentially.
- # of access port remains stable.
- Implication:
  - Test Complexity Index (# of transistors per pin) increases rapidly.

Implications of SIA Roadmap: Testing

<table>
<thead>
<tr>
<th>Year</th>
<th>F. Size [µm]</th>
<th>Testing Complexity Index - [Tr. per Pin]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1992</td>
<td>0.5</td>
<td>1.0E+5</td>
</tr>
<tr>
<td>1995</td>
<td>0.35</td>
<td>1.40E+5</td>
</tr>
<tr>
<td>1998</td>
<td>0.25</td>
<td>1.60E+5</td>
</tr>
<tr>
<td>2001</td>
<td>0.12</td>
<td>1.80E+5</td>
</tr>
<tr>
<td>2004</td>
<td>0.1</td>
<td>2.00E+5</td>
</tr>
<tr>
<td>2007</td>
<td></td>
<td>4.00E+4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.00E+4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.00E+5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.40E+5</td>
</tr>
</tbody>
</table>

Source: W. Maly, 1996
Source: SIA Roadmap
Built-In Self Test (BIST)

- Solution: Dedicated built-in hardware for implementing test functions.
  - No need for expensive ATE.
  - At-speed testing.
  - Concurrent test possible.
  - Support O&M.
  - Support field test and diagnosis.

- Design for the best BIST mechanism = optimization.
- Testing => Design

Challenges still Remain

- System specification with very high-level languages.
- Modeling techniques for heterogeneous system.
- Testing issue to be considered during the design process.
- Design verifications => get the whole system right the first time!
- Very efficient power saving techniques.
- Design techniques to address process variation.
- Temperature aware design approaches.
- Powerful optimization algorithms.
- Parallel computers for design algorithms.
Conclusion Remarks

- Much of design of digital systems is managing complexity.
- What is needed: new techniques and tools to help the designers in the design process, taking into account different aspects.
- We need especially design tools working at the higher levels of abstraction, in order to deal with the complexity and have good design productivity.
  - This is what our course will focus on!