EEM412 Course

Introduction to IC Design – II

Instructor

Dr. İsmail Enis Ungan

Design Description

- An IC design may be described in three domains;
  - Behavioral Domain
  - Structural Domain
  - Physical Domain

- Design flows from behaviour to structure then finally to physical implementation, manually or by automated tools.
  - Behaviour descriptions are transformed to structural descriptions
  - Structural descriptions are transformed to physical descriptions

- At each transformation, correctness of the transformation is tested by comparing the pre- and post-transformation design.

Design Description

- Each domain has design options. A design option is selected to solve a particular problem during the design. The choice should be entirely **economic**.
  - In behavioral domain, selection of either parallel or sequential algorithm.
  - In structural domain, possible selections may be;
    - one of logic family (CMOS, LVTTL, LVDS, GTL+, etc.)
    - one of clocking strategy
    - one of logic structures (complementary, dynamic, transmission gate etc.)
  - In physical domain, possible selections may be;
    - set of chips, boards, and units
    - one of processing technology
    - one of IC packages
    - one of cell libraries

Design Description

- Each domain may be hierarchically divided into levels of design abstractions;
  - Architectural or Functional abstraction level
  - Logic or Register-Transfer abstraction level (RTL)
  - Circuit abstraction level

Design Description

- Examples to Behavioral Domain and its Abstraction Levels
  - Architectural Abstraction Level
    - Systems, Operating systems, Algorithms
  - RTL and Logic Abstraction Level
    - Applications, Register transfer, Programs, Logic, Procedures, Logic functions, C++ or HDL Statements
  - Circuit Abstraction Level
    - Subroutines, Instructions, Machine codes, C++ or HDL Statements

Design Description

- Examples to Structural Domain and its Abstraction Levels
  - Architectural Abstraction Level
    - Processors, SOCs, Hardware Modules, Parallel, Sequential, Serial, Array architectures
  - RTL and Logic Abstraction Level
    - ALUs, Registers, Adders, Multipliers, Gates, FlipFlops
  - Circuit Abstraction Level
    - Transistors, Logic family, Logic structures

Design Description

- Examples to Physical Domain and its Abstraction Levels
  - Architectural Abstraction Level
    - Boards, ICs, Packages
  - RTL and Logic Abstraction Level
    - Multi-chip modules, Cells
  - Circuit Abstraction Level
    - Process technology, Cell layouts, Transistor Layouts, Device Layout, Rectangles

Structured Design Strategies

- In an IC design, a specified **concept** is converted into;
  - Architecture,
  - Logic and Memory,
  - Circuit,
  - Physical Layout.
- The design description domains and design abstraction levels in each domain must have **consistent** descriptions.
- Domain and abstraction level descriptions have the following parameters to measure and consider;
  - Performance (speed, power, function, flexibility)
  - Die size (cost of die)
  - Design time (cost of engineering and schedule)
  - Ease of verification, testability (cost of engineering and schedule)

Structured Design Strategies

- Design is a continuous tradeoff to achieve **adequate** results for **all** of the design parameters.
- Methodology and tools aim to reduce IC design complexity and to provide team-work.
- A good method of simplifying the approach to a complex design is by use of **constraints** and **abstractions**.

Structured Design Strategies

- Structured Design Techniques;
  - Hierarchy
    - That is divide and conquer. Reduce complexity by dividing modules into sub-modules and then repeat division until the complexity of sub-modules is at an appropriately comprehensible level of detail.
    - Allows the use **virtual components** (reusable previous designs or **intellectual property** designs)
  - Regularity
    - Attempt to divide the hierarchy into a set of similar building blocks.
    - "Correct by construction" is possible.
    - Less effort in verifications.

Structured Design Strategies

- ... Structured Design Techniques;
  - Modularity
    - Modules should have “Well-defined” functions and interfaces.
    - Interaction among other modules can be well characterized.
    - Well-defined:
      - Name, Function, Signal type, Electrical and Timing constraints of the ports, port locations, port wiring layer and width.
    - Helps the designer to clarify and document an approach to a problem.
    - Well-defined modules become IP sources that aid System-on-Chip designs where many of them have to be interfaced.
  - Locality
    - A form of “information hiding” of a module by specification of external interfacing only. So that, module’s apparent complexity is reduced.

Design Options

- There are a number of design options to choose in order to implement an IC design.
  - Microprocessor or DSP
    - General purpose processors.
    - Great flexibility by updating system in the field.
    - Costs reduce by embedding the microprocessor into a SoC.
    - Often, cost, speed, or power dissipation may not meet the system goals.
  - Programmable Logic
    - More efficient than microprocessors.
    - A particular programmable logic device can be selected to implement different IC designs.
    - Field Programmable Gate Arrays
      - One-time programmable (non-volatile)
      - Re-programmable (volatile or non-volatile types)

Design Options

- Gate Array and Sea-of-Gates Design
  - Base wafers contain array of transistors ready to be connected.
  - Interconnects among transistors are programmed during chip fabrication using 2-5 metal masks only (less mask costs and process time).
- Standard Cell Design
  - Logic or function is found in a library. IC designer uses cells from the library. Design entry is done by schematic or a hardware description language (HDL). Layout is automatically placed and routed by CAD software.
- Full-Custom Design
  - Logic function and layout of every transistor/device is optimized. Provides the best design parameters of speed, power, function, and die area. But, has the worst design parameters of flexibility and design time.

Design Tools

- A design starts with Behavioral Level then progress with RTL Level, and then Layout Level.
- CAD tools may be employed at any design description domain and at any design abstraction level.
  - Behavioral Synthesis Tool
    - Design description in behavioral domain can be synthesized and converted into a layout in the physical domain.
    - Example at architectural abstraction level: Y = FILTER(parameters);
    - Example at RTL abstraction level: a = a + b x c
  - RTL Synthesis Tool
    - Design descriptions at RTL abstraction level are usually done by an HDL (like Verilog HDL, VHDL). An HDL design entry is synthesized by a CAD tool. The output is a set of registers and combinational logic.

Design Tools

- Logic Optimization Tool
  - RTL level descriptions are converted to an other optimized set of RTL level descriptions using the predefined library.
- Structural-to-Layout Synthesis Tool
  - RTL level descriptions are converted to a layout. Programmable logic, including the Gate Array, and Standard Cell design options use this synthesis. Two main synthesis phases are placement and routing.
- Layout Synthesis Tool
  - RTL level descriptions are converted to layout.
  - Examples: RAM, ROM, Multipliers, Adders, Register arrays.

Design Tools

- Design Capture Tools
  - HDL design
  - Schematic design
  - Layout design
  - Floor Planning: Arrangement of layout blocks in order to minimize delay and die area.
  - Chip Composition: Floor planning and inter-block routing.
- Design Verification Tools
  - The design descriptions in all domains and at all abstraction levels are needed to be verified in order to fulfill the design specifications.
  - **Simulation**: Circuit Level, Timing, Logic Level, Switch Level, Mixed-Mode
  - **Timing Verification**: Propagation delays for each signal path are analyzed and the "critical paths" are emphasized.
  - **Netlist Comparison**: Circuit level design abstractions in structural domain and physical domain are compared with each other.

Design Tools

  - **Design Rule Verification**: Geometric patterns for mask generation in the layout design are verified for conformance to given geometric design rules.
- Supplementary Tools
  - **Layout Extraction**: Physical domain design description at circuit level is converted to description in structural domain at circuit level.
  - **Back-Annotation**: Layout design adds parasitics to circuit design. The tool extracts and moves the parasitics in physical domain to the circuit design in structural domain for further design verifications.
  - **Pattern Generation**: Mask data are generated form layout design for IC production. A common data format is Electron Beam Exposure System.

Design Flow

- Design flow is a set of procedures that allows designers to progress from a specification to the final chip implementation in an error-free way.
- Generalized Design Flow
  - Product Requirement
  - Behavioral/Functional Specification
    - Simulate and Compare / Modify
  - Behavioral (RTL) Synthesis
  - Structural Specification
    - Simulate and Compare / Modify
  - Physical Synthesis
  - Physical Specification
    - Simulate and Compare / Modify
  - Fabrication

Design Economics

- IC production costs are;
  - Nonrecurring Engineering Costs (NRE)
    - They are spent once during the IC design and manufacturing.
    - Include;
      - Engineering design costs
        - » Personnel costs due to design work
        - » Support costs due to computer, CAD tools, education.
      - Prototype manufacturing costs
        - » Mask cost
        - » Test fixture cost
        - » Packaging cost

Design Economics

- Recurring Costs
  - The manufacturing price for a specific IC. Includes a recuring cost that recurs every time an IC is sold.
    - Packaging cost
    - Testing cost
    - Wafer cost + Process cost
      - » Die yield, test yield, package yield are considered.
- Fixed Costs
  - IC support cost
    - Writing data sheets
    - Writing application notes
    - Marketing and overhead costs

Design Economics

- Design Schedule
  - A design schedule aims to bring an IC project to success by managing the available resources.
  - Estimating the schedule is essential to be able to select a **strategy** by which the ICs will be available in the **right time** and at the **right price**.
  - To estimate the schedule, amount of effort need to be guessed.
  - Although productivity can be formulated, the best predictor of design schedule for a team is the previous performance.

Design Economics

- Project Management
  - It is the overall supervision of the project.
  - Many tasks are listed and resources (designers) are assigned to the tasks.
  - Resources become available at the appropriate time.
  - Communication between design groups are ensured.
  - Project progress and risks are summarized regularly to the management.
  - Rapid prototyping management approach
    - Prototype basically works. Detail is added later on. It is risky.
  - Preplan every thing management approach
    - Estimate task times
    - Use project planning tool
    - Products do not come out soon, but delivers within budget and on time.

Data Sheets

- Data sheet for an IC design describes what it does and outlines the specifications to be used in a system.
- The data sheet includes;
  - Introduction / Summary
  - Pinout
  - Operation description
  - DC specifications
  - AC specifications
  - Package diagram / dimensions