Evolution and Trends of VLSI Design
Methodologies and CAD Tools

Presenter: Raj Singh

IC Design Group
CEERI
Pilani – 333 031

Tel: 01596-242359
Fax: 01596-242294
Email: raj@ceeri.ernet.in
Design Methodologies

A View of IC Design

Architecture

Algorithm

CAD

Technology
Complexity of VLSI Design Task

- Problem Domain Complexity:
  - Competing or contradictory requirements (speed, power, area).
  - “Impedance mismatch” between users and designers.
  - Application area specialization and knowledge.
  - Changing/evolving specifications.
  - Specification capture issues.
  - Cost of design/development.
Design Methodologies

**Complexity of VLSI Design Task**

- **Design/Development Process Complexity:**
  - Rapidly changing technology.
  - Large task requiring multi-disciplinary team.
  - Geographically separate design teams.
  - Large design space.
  - Short design cycle.
Complexity of VLSI Design Task

- **Design/Development Process Complexity:**
  - First-time success requirement.
  - Assuring unity and integrity of design.
  - Multiple views/representations each with different characterization.
  - Inadequate documentation.
Design Methodologies

Complexity of VLSI Design Task

- Complexity Because of Choices:
  - Many technologies/implementation choices (NMOS, CMOS, ...)
  - Many methodologies (FPGA, Semi-custom, Full custom, ASIC, ...)
  - Many logic forms (Dynamic, 2-phase, 4-phase, Static, ...)
  - Combinatorial explosion as one goes down in abstraction level.
  - Many possible partitions at each level.
Complexity of VLSI Design Task

• Other Complexities:
  – Clocking and Timing issues.
  – Testing-related issues.
  – Packaging-related issues.
  – Concurrency of hardware operations.
VLSI/ASIC/FPGA design activity is

*Synthesis* and then *Analysis*

at several successive levels of design abstraction.

**Synthesis Phase** : Proposing a solution of the design problem at a certain level of abstraction.

**Analysis Phase** : Checking that solution for its validity as well as its consistency with some other design representation (usually at a different level of abstraction) and the characterization/evaluation of the design solution.
**Synthesis and Analysis**

At each level of design abstraction, *synthesis* and *analysis* may be used in a loop (iterative procedure) to arrive at an optimal solution.

*Synthesis* involves *creativity* and *new concepts*.

*Analysis* (except when developing new methods of analysis) involves applying *known methods* for checking the design and processing of large data using *known/standard methods*.

Thus, *synthesis* has traditionally been the prerogative of humans, while *analysis* has traditionally been the first task to have been handed over to the computers via creation of CAD tools.

When *synthesis* can be expressed as a method, it can also be carried out by computers via CAD tools created for that purpose.
More recently, synthesis tools have been created which draw upon the results of human creativity and exploration via systematized methods and choices of approaches to produce “good” synthesized solutions for the design problems (or parts of it) at different levels of design abstractions.

Analysis tools also improve as the results of human creativity and exploration in the form of new methods get incorporated into them.

Therefore, the division of labour in VLSI/ASIC/FPGA design is clear:

Humans do the creative parts and computers (CAD tools) do the cumbersome detailed work.
Design Methodologies

Top Level of Abstraction

Lower Level of Abstraction

Lowest Level of Abstraction

More Abstract Level
(Consistency)

Synthesis
Analysis
(Check)

Less Abstract Level
Design Methodologies

Traditional VLSI Design Flow

- **Design Entry / Design Synthesis**
  - **Behavioral Level**
    - OK
    - No
    - Design Output
  - **Register-Transfer Level**
    - OK
    - No
    - Design Output
  - **Logical/ Register Level**
    - OK
    - No
    - Design Output
  - **Transistor/Circuit Level**
    - OK
    - No
    - Design Output
  - **Layout/Physical Level**
    - OK
    - No
    - Design Output

- **Design Verification Through Simulation**
  - Specification From User
Design Methodologies

HDL-Based Design Flow

Specifications + Constraints

Code Organization Choice

Behavioral Description in HDL

Architectural Choice

RT-Level Description in HDL

Logic Implementation Choice

Gate-Level Netlist

Placement and Routing Choice

Physical Design
Design Methodologies

VLSI Design Methodologies

Systematic design methods (called design methodologies) are necessary for successfully designing complex digital hardware.

Different design methodologies differ in their choice of number and levels of design abstractions used during the design process and the manner of constraints on the translations between the abstraction levels.

These constraints are usually in the form of use of a particular structure type at the lower level of design abstraction while translating the design description that exists at a higher level of abstraction.
For example, while translating from the logic level abstraction to physical level, popular design methodologies are:

1. FPGA/CPLD.
2. Gate-Array.
4. Full-Custom.
The popular methodologies for implementing the control part are:

1. Hardwired Control.
3. PLA-based Control.
Design Methodologies

**Design-Foundry-CAD Interface**

- Proprietary Design Tools and Libraries
  - Design for Fabrication
    - ASIC Vendor or Foundry
    - ASIC Devices
    - ASIC Design Center
  - ASIC Users
- Back-end CAD Tools
  - CAD Tool Vendor
- Front-end CAD Tools
Typical Phases of a Product

- Awareness of need and its Identification.
- Design and Development.
- Implementation/Manufacture.
- Testing/Verification/Validation.
- Feed-back and Correction.
- Release/Maintenance.
CAD : Computer Aided Design

What is CAD?

Any kind of activity which uses a computer to assist in the creation, modification and analysis of a design.

ECAD : Electronic CAD.

EDA : Electronic Design Automation.
Major Benefits of CAD

- Improved performance and quality of product. *(better engineered products).*

- Higher productivity of designers.

- More flexibility to react to market changes and product modifications.

- Provide time to experiment and explore alternatives. *(design space exploration).*
CAD Environment: Factors Affecting Design

- Market requirements.
- Manufacturing technology status.
- CAD tool(s) status.
- Human talent available.
CAD Environment: Reducing Risk of Failure

- Simulate performance before fabrication. (*simulation*)

- Explore various alternatives and characterize them in terms of cost, performance, ... (*design space exploration*)

- Check against all possible known fabrication process violations before data is given to manufacturing. (*design sign-off*)

CAD tools help discover problems so that they can be corrected at minimal cost, both in terms time and resources (*catch errors as early as possible*).
CAD Tool Components

- Input Handler (input from keyboard/mouse).
- Data Structure + Algorithm (in-memory).
- Output Handler (output to display).
- File Input/Output Handler (to/from disk).
Design Methodologies and CAD Tools

CAD Tools at Various Levels of Hierarchy

- **Design Entry (HDL)**
- **Design Entry (C, C++, ...)**
- **Design Entry (Schematic)**
- **Specification From User**
- **Behavioural Simulation**
- **Behavioural Synthesis**
- ** RTL Simulation**
- **Analysis**
  - **Logic Simulation, LVL, Static Timing Analysis, ERC**
  - **Design Entry (HDL)**
  - **Design Entry (Schematic)**
  - **DFT Insertion, ATPG**
- **Logic Synthesis**
- **Register-Transfer Level**
- **Logic/Gate Level**
- **Circuit Synthesis**
- **Circuit Simulation, Power Analysis, Delay Estimation**
- **Transistor/Circuit Level**
- **Analysis Synthesis**
- **Layout/Physical Level**
  - **LVS, DRC, Circuit Extraction**
  - **Design Entry (Layout), Floor Plan, Place-and-Route**
  - **Flash Analysis**
  - **PG File Creation**
  - **Fracturing, Sorting, Process Simulation, Lithography Simulation**
  - **Yield Prediction, Tester File Creation**
  - **Mask Level**
  - **Process Simulation, Lithography Simulation, Validation, Test (File Creation, Foundry)**

©CEERI, Pilani
Design Methodologies and CAD Tools

Design and Tool Levels

**SYSTEM STRUCTURAL DOMAIN**
- System Structure
- Processors, Buses
- RAM, Registers, ALUs
- Gates, Flip-flops
- Transistors

**BEHAVIORAL DOMAIN**
- System Behavior
- Flow charts, Algorithms
- Register Transfers
- Boolean Expressions
- System Design and Behavioral Design

**TRANISTOR DESIGN**
- Transistor Functions
- Transistor Layouts
- Cell Designs

**CELL DESIGN**
- Cell Layouts

**LAYOUT DESIGN**
- Block Layouts
- Chips, Floorplans
- Boards, MCM, System Partitions

**PHYSICAL DOMAIN**

©CEERI, Pilani
IC Design Group
CAD Tools Classification

Front-end Tools:

Design Entry, Editors, Simulation, Synthesis, Timing Analysis, DFT Insertion, Test Generation, ...  

Back-end Tools:

Floor Planning, Place-and-Route, Extraction, LVS (Layout vs. Schematic), LVL (Layout vs. Logic), ERC, DRC, Pattern Generators, Format Converters, Mask Graphics, ...
Design Methodologies and CAD Tools

Another CAD Tools Classification

Design Capture Tools:

Editors, VHDL, Verilog, State Charts, FSM Capture, ...

Synthesis Tools:

Behavioral Synthesis, RTL Synthesis, FPGA Synthesis, Logic Synthesis, Physical Synthesis, Module/Cell Generators (ROM, PLA, RAM), Data-path Compiler, Adder/Multiplier Generators, DSP Synthesis, ...
Design Methodologies and CAD Tools

Another CAD Tools Classification

Analysis Tools:

- **Checkers**: DRC, ERC, Net Compare, Ratio Checker, Short-circuit Checker, Fan-in / Fan-out Checker, Power Checker, ...

- **Verifiers**: Timing Verifier, Simulators, ICE/Hardware Simulators, Formal Verifier, ...

Testing Related Tools:

ATPG, DFT Tools, ...
Evolution of CAD Tools and Methodologies: First Epoch (1959-1979)

Layout level tools, Circuit level tools. Full-Custom design methodology.

Feedback on the design available to designer at layout level only, either after fabrication or after circuit extraction.

Performance (speed, area, power) assessment not part of overall design process.

Technology at SSI, MSI, LSI levels.

Designer, User and Tools Developer – all at a single company (monolithic).

Productivity of designer (notionally) = 10 transistors/day.

Logic level tools, Macro generators, Module compilers, ASICs, FPGAs.

Shorter synthesis/analysis loop. Standard-Cell / Gate-Array / ASIC design methodology.

Basic capability of performance feedback and design exploration becomes available. However, it is not fully incorporated in the design flow.

Technology at LSI, VLSI levels.

User/Designer separated from Tools Developer i.e. separate companies selling tools come into being.

Productivity of designer (notionally) = 10 gates/day.
Evolution of CAD Tools and Methodologies: Third Epoch (1990-1999)

RT-level synthesis, Preliminary Behavioural synthesis tools. HDL-based design entry, HDL Code analyzers/advisers.

Design exploration easier, Estimators for performance become available. HDL-based design methodology.

Technology at VLSI levels. Deep-submicron (DSM) issues.

User, Designer and Tools Developer become separate groups/companies. Fabless companies.

CAD tools move towards PC platforms.

Productivity of designer (notionally) = 10 lines of VHDL code/day.
Evolution of CAD Tools and Methodologies: Third Epoch (1990-1999)

Low-power gains importance. Mixed-signal design issues.

Tools for designing MEMS and Embedded systems appear.

Intellectual Property, Design Re-use, Reconfigurable Computing, Cores, ASIP, ASSP, DSP, ...
Evolution of CAD Tools and Methodologies: Fourth Epoch (2000 …)

System-level synthesis tools, Systems-on-a-Chip (SoC), MEMS/MEOS.

Block-level chip design (using cores, IP blocks). SoC design methodology.

Consolidation among EDA companies.

Technology at VLSI, ULSI levels. Deep-submicron (DSM) issues.

Productivity of designer (notionally) = 10 lines of specification code/day.

RF IC Design, Hardware-Software Codesign, Web-based CAD tools and design environment.

Methodology and Tools for Non-Silicon based Designs? Nano-technology?
What is Hardware-Software Codesign?

Integrated design of electronic systems implemented using hardware and software components developed *concurrently* and *cooperatively*.

It is a part of the system-level design which may consist of mechanical, electrical or chemical parts in addition to electronics.
Advantages of Hardware-Software Systems

- Family of products on a common hardware platform.

- Upgradation and efficient evolution path of product through updating software.

- Chip/circuit’s high cost reduced by providing functionality in software.

- Range of system costs and performances – from (high cost + high performance) to (low cost + low performance).

  Embedded core, ASIC/FPGA, Microprocessor + Software, …
Application Areas

- Large systems *e.g.* Aircraft, Telecommunication.

- Computing systems *e.g.* Supercomputer, Workstation, PC.

- Strategic/Defence systems *e.g.* Radar, Missile.

- Embedded systems.
  - Control *e.g.* automobile, medical, industrial.
  - Hand-held *e.g.* cellular phone, PDA.
Application Areas

- Consumer *e.g.* microwave oven, washing machine.
- Music systems *e.g.* MP3 players, Ogg players.
- Sound recording *e.g.* phone-answering machine.
- Speech processing *e.g.* voice synthesis.
- Graphics processing *e.g.* laser printer, X-terminal.
- Video processing *e.g.* VCR, VCD, DVD, Digital TV, HDTV.
- Robotics and Mechatronics.
Hardware-Software Design Steps

- System-level Modeling and Simulation.

- Hardware-Software Partitioning.

- Concurrently,
  - Hardware Synthesis.
  - Interface Synthesis.
  - Software Code Generation.
Trends: Codesign and Embedded Systems

Codesign Environment

- System Model
- Optimization
- Partitioning

HARDWARE
- Synthesis
- Netlist
- FPGA, ASIC, ...

SOFTWARE
- Synthesis ?
- Netlist ?
- ROM, Glue Logic
- uP, uC, DSP, ...

CO-SIMULATION

©CEERI, Pilani
IC Design Group
What is Reconfigurable Computing?

Reconfigurable Computing is an approach that allows reconfigurable aspects of hardware (e.g. FPGAs) to be as flexible as software.

It can be implemented either statically (configure and then repeatedly execute) or dynamically (repeatedly configure-and-execute).

It requires extensive knowledge base and quantitative approach for evaluating different system architectural options vis-a-vis the system requirements (in terms of speed, power, cost, user-interface, configurability, . . .).
Why Reconfigurable Computing?

- “Virtual” hardware.
- Reduced time-to-market.
- Cheaper than ASICs or processors.
- Better re-use of Silicon than ASICs.
- More application specific adaptation than processors.
- Lower system life-time cost. Also, upgradeable in the field.
Types of Reconfigurable Computing

- Rapid Prototyping.
- Hardware Accelerator; Hardware Emulator.
- Processor + FPGA (Embedded Systems).
- Reconfigurable Compiler.
Trends: Reconfigurable Computing

Types of Reconfigurable Computing

- Part-reconfiguration and on-the-fly Reconfiguration.
- Multi-context Architectures.
- Structured ASIC.
Conclusion

VLSI Design is a complex task.

CAD tools and design methodologies are important partners of the designer in overcoming this complexity.

CAD tools free the designer from easy-to-do tasks and allow the designer to concentrate on creative tasks.

Design methodologies and CAD tools evolve based on designer’s needs.