Computer-Aided VLSI System Design: An Introduction

Jie-Hong R. Jiang

National Taiwan University
Outline

- Review of the relationship between layout and IC manufacturing
- The design productivity challenge
- Design flow overview
- Full custom design flow
- Cell-based design flow
Outline

- Review of the relationship between layout and IC manufacturing
  - The design productivity challenge
  - Design flow overview
  - Full custom design flow
  - Cell-based design flow
Integrated Circuits (ICs)
Invention of Integrated Circuits

Robert Noyce, 1927—1990

Jack St. Clair Kilby, 1923—2005
A Brief History

- 1947: First transistor
  - Invented by W. Schockley, J. Bardeen, and W. Brattain in Bell Labs, New Jersey

- 1958: First integrated circuit
  - Flip-flop using two transistors
  - Invented by Jack Kilby at Texas Instruments, Texas

- 2003
  - Intel Pentium 4 processor (55 million transistors)
  - 512 Mbit DRAM (> 0.5 billion transistors)

- 53% compound annual growth rate over 45 years
  - No other technology has grown so fast so long

- Driven by miniaturization of transistors
  - Smaller is cheaper, faster, lower in power!
  - Revolutionary effects on society
Moore’s Law and Technology Scaling

“...the performance of an IC, including the number components on it, doubles every 18-24 month with the chip price...” —Gordon Moore 1965
Moore’s Law in Action

- Example:

  Intel Pentium 4 microprocessor evolution

  - 0.18 micron technology
    - 42M transistors
    - Die size: 217 mm$^2$

  - 0.13 micron technology
    - 55M transistors
    - Die size: 146 mm$^2$

Source: Intel
The Relationship between Layout and IC Manufacturing

- We start from the simplest logic gate: Inverter
Inverter Cross-Section

- Typically use p-type substrate for nMOS transistors
- Requires n-well for body of pMOS transistors
Layout of Inverter

- Transistors and wires are defined by *masks*
- Cross-section taken along dashed line
Detailed Mask Views

- Six masks
  - n-well
  - Polysilicon
  - n+ diffusion
  - p+ diffusion
  - Contact
  - Metal
Transistor & Interconnection Layers

Source: Application-Specific Integrated Circuits

C. M. Huang / 2000.01 / CBDC
Process Design Rules

Source: Application-Specific Integrated Circuits
Outline

- Review of the relationship between layout and IC manufacturing
- Design productivity challenge
- Design flow overview
- Full custom design flow
- Cell-based design flow
Design Productivity Challenge

A growing gap between design complexity and design productivity

Source: sematech97
How to Fill the Gap?

- New design methodology
  - Computer-aided design (CAD) tools
  - Module-based design

- New design concept
  - Different levels of design reuse
Implementation Choices

Digital Circuit Implementation Approaches

- Custom
- Semicustom
  - Cell-based
    - Standard Cells
    - Compiled Cells
  - Array-based
    - Macro Cells
    - Pre-diffused (Gate Arrays)
    - Pre-wired (FPGA's)
Types of ASICs

- Full-custom ASICs
- Semi-custom ASICs
  - Cell-based ASICs
  - Gate Array-based ASICs
- Programmable ASICs
  - Programmable Logic Devices
  - Field Programmable Gate Array

<table>
<thead>
<tr>
<th></th>
<th>Building block</th>
<th>Interconnection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-custom</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Semi-custom</td>
<td>Δ</td>
<td>O</td>
</tr>
<tr>
<td>Programmable</td>
<td>Δ</td>
<td>Δ</td>
</tr>
</tbody>
</table>

O = custom design
Δ = NOT custom design
Full-custom ASICs

Circuit Design

Layout Design
Standard Cell-based ASICs

Standard cell

[Diagram of standard cell layout]
Standard Cell-based ASICs (cont'd)
Gate Array-based ASICs

Channeled Gate Array

Channelless Gate Array

Embedded Gate Array

Source: Application-Specific Integrated Circuits
Programmable ASICs

Programmable Logic Devices (PLD)  Field Programmable Gate Array (FPGA)
Break-even Analysis of ASICs

FPGA cheaper

ASIC cheaper
Outline

- Review of the relationship between layout and IC manufacturing
- The design productivity challenge
- Design flow overview
- Full custom design flow
- Cell-based design flow
VLSI Design Flow
VLSI Design Flow (cont’d)

- Extraction & verification
- Structural representation
- Physical design
- Fabrication
- Packaging

Diagram showing the flow from extraction and verification to structural representation, then to physical design, fabrication, and packaging.
Design Objectives

- Effectively translate and implement design ideas into foundry required contents and formats, which satisfy the following design requirements:
  - Function correctness
  - Performance (timing)
  - Die size (area)
  - Power consumption
  - Design effort
  - Reliability

- The way to do it is the so-called design methodologies
Design Challenges

- How to describe (model) the design?
  - Possible description models
  - How to choose the most suitable model for your design

- How to verify the model?
  - Function verification
  - Circuit analysis
  - Topological/geometrical constraint verification

- How to realize the model into a more detailed model?
  - Manual design
  - Design automation

- Iterative refinement until design constraints satisfied
How to Describe the Design

- Three design domains
  - Behavioral domain
  - Structural domain
  - Physical domain

- Several levels of design abstraction
  - Behavioral/functional level
  - Register-transfer level (RTL)
  - Logic level
  - Circuit level
  - Layout level
Design Domains and Levels

Y-Chart
(Daniel D. Gajski, 1988)
Possible Representations

if $A = 0$ then
  $Z = 1$;
else
  $Z = 0$;

Behavioral Domain  Structural Domain  Physical Domain
How to Choose Proper Domain and Level

- Depend on your application
  - Regular structure module →
    use full custom design flow
    - Structural domain (lower level)
    - Datapath, memories, etc.

- Irregular structure module →
  use cell-based design flow
  - Behavioral domain (higher level)
  - FSMs, random logics, controllers, etc.
Function Verification

- Design / implementation verification
  - Informal verification
    - Simulation
  - Formal verification
    - Property checking, equivalence checking

- Manufacture verification
  - Testability analysis
    - Fault coverage
Circuit Analysis

- **Timing**
  - Delay, setup/hold time

- **Power**
  - Static/dynamic power consumption

- **Signal integrity**
  - Noise immunity, tolerance to process variation
Topological / Geometrical Verification

- Layout vs. schematic (LVS)
  - Netlist comparison
  - Electrical parasitics extraction

- Design rule checks
Mappings among Domains

Design space exploration

- Behavioral
- Structural
- Physical

one-to-many
Many-to-one
How to Choose Mapping Target

- Tradeoffs among design effort, speed, and area

```
    Speed
     ↑
    /  \
  Regularity
     ↓
  Area Design Effort
```
Manual Implementation

- Manual logic minimization, circuit layout

  - Pros
    - Potential to achieve optimal solutions under certain design scale
  
  - Cons
    - Quality heavily depends on designers
    - Error prone
    - Relative long turn around time
    - Relative low productivity per designer
Design Automation Using CAD Tools

- Automatic behavioral to structural synthesis, structural to physical synthesis

  □ Pros
  - Relative high productivity per designer
  - Relative short turn around time
  - Potential for re-targetable design

□ Cons
  - Can achieve sub-optimal (yet reasonable) solutions for most cases
Chip Integration

- Behavioral/functional domain
  - Simulation in the same or different levels of design abstraction
  - Property verification

- Physical domain
  - Abstract model for P&R tools
Outline

- Review of the relationship between layout and IC manufacturing
- The design productivity challenge
- Design flow overview
- Full custom design flow
- Cell-based design flow
IC Design Flow: Full-Custom Design

1. Specification
2. System Architecture
3. Logic Design
4. Circuit Design
5. Layout
Full-Custom IC Design

Full Custom 設計流程

- Behavioral Level
- RTL Level
- Logic Synthesis
- Logic Level Design
- Circuit Level Design
- Layout Level Design
- Post Verification

Composer → SPICE

Virtuoso/Olive, Dracula

RC Extraction (Star-RC, Dracula)

GDSII
The Custom Approach

Intel 4004

Courtesy Intel
Outline

- Review of the relationship between layout and IC manufacturing
- The design productivity challenge
- Design flow overview
- Full custom design flow
- Cell-based design flow
Transition to Automation and Regular Structures

Intel 4004 ('71)

Intel 8080

Intel 8085

Intel 8286

Intel 8486

Courtesy Intel
IC Design Flow: Cell-Based Design (1)

Specification

System Architecture

Verilog HDL (Hardware Description Language) Code

Simulation to Ensure the Correctness
IC Design Flow: Cell-Based Design (2)

Synthesis
(RTL to Gate)

Netlist
(Verify the timing, area, power, ...)

Automatic Place and Route
(APR, physical design)

Layout
(Verify the timing, area, power, ...)

IC Design Flow:
Cell-Based Design (2)
Cell-based Design

Routing channel requirements are reduced by presence of more interconnect layers.

Routing channel

Feedthrough cell

Logic cell

Rows of cells

Functional module (RAM, multiplier, …)
Standard Cell — Example

[Brodersen92]
Standard Cell – The New Generation

Cell-structure hidden under interconnect layers
Standard Cell - Example

3-input NAND cell
(from ST Microelectronics):
C = Load capacitance
T = input rise/fall time

<table>
<thead>
<tr>
<th>Path</th>
<th>1.2V - 125°C</th>
<th>1.6V - 40°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>In1→t_{pLH}</td>
<td>0.073+7.98C+0.317T</td>
<td>0.020+2.73C+0.253T</td>
</tr>
<tr>
<td>In1→t_{pHL}</td>
<td>0.069+8.43C+0.364T</td>
<td>0.018+2.14C+0.292T</td>
</tr>
<tr>
<td>In2→t_{pLH}</td>
<td>0.101+7.97C+0.318T</td>
<td>0.026+2.38C+0.255T</td>
</tr>
<tr>
<td>In2→t_{pHL}</td>
<td>0.097+8.42C+0.325T</td>
<td>0.023+2.14C+0.269T</td>
</tr>
<tr>
<td>In3→t_{pLH}</td>
<td>0.120+8.00C+0.318T</td>
<td>0.031+2.37C+0.258T</td>
</tr>
<tr>
<td>In3→t_{pHL}</td>
<td>0.110+8.41C+0.280T</td>
<td>0.027+2.15C+0.223T</td>
</tr>
</tbody>
</table>
Standard Cell Library

- Core cells (for random logic)
  - Combinational elements, sequential elements
- I/O cells
  - Input, output, inout, power
- Hard macro generators (for regular logic)
  - RAM, ROM, PLA, datapath
Design Flow for Standard Cells

- Cell specification
- Circuit design
- Layout design
- Circuit and RC extraction
- Parameter extraction
- Models (Views) Generation
- Test key design and fabrication
- Parameter measurement and calibration
CIC Provided Cell Libraries

- TSMC 0.35um 1P4M CMOS process
- TSMC 0.25um 1P5M CMOS process
- TSMC 0.18um 1P6M CMOS process
- TSMC 0.13um CMOS process
- UMC 90nm CMOS process
- .... Keep shrinking
- CIC virtual process
Where are Standard Cell Libraries?

- CIC libraries are all installed on NTUEE cad machines
- You can see the path when logged in
Full-Custom Design v.s. Cell-Based Design

- Full-custom design: for memory, analog blocks, and physical layer interfaces for high-speed communication protocols

- Compared with full-custom design, cell-based design (data from an experiment)
  - Maximum clock frequency was the same
  - Power was about 2x higher without any power synthesis tool
  - With power synthesis tool, the power overhead is within few percent of the original full-custom version
  - Area overhead: 7%

[Ref: Reuse Methodology Manual]
Full-Custom Design v.s. Cell-Based Design

- The most aggressive processor designers are using full-custom techniques only for small portions of their design.

- Processor designers tend to use synthesis for control logic and full custom only for data paths.

[Ref: Reuse Methodology Manual]
Cell-Based IC Design
A Typical Cell-Based Design Flow

1. Specifications
2. RTL Coding
3. Synthesis
4. Dft Insertion
5. Place and route
6. Tape out

Front-End

- ATPG
  - Test patterns
  - Fault coverage

Back-End

- DRC LVS LPE
  - layout

behavior description
RTL codes
Gate-level netlist w/o scan chain
Gate-level netlist /w scan chain
Tools and Files

- **specifications**
- **RTL Coding**
  - Your brain
- **Synthesis**
  - Design Compiler
- **Dft Insertion**
  - dft compiler
- **ATPG**
  - Tetramax
- **Place and route**
  - SoC Encounter/ IC Compiler
- **Tape out**
  - Calibre Dracula Time/Powermill

Files:
- `behavior.v`
- `rtl.v`
- `gate.v`
- `gate_scan.v`
- `spf`
- `stil`
- `gds2`
- `wgl`
RTL Coding and Debugging

1. **specifications**

2. **RTL Coding**
   - Your brain!

3. **Synthesis**
   - Design Compiler

4. **Dft Insertion**
   - dft compiler

5. **Place and route**
   - SoC Encounter/IC Compiler

6. **Tape out**

7. **Waveform Viewer**
   - Verdi
   - .fsdb

8. **Simulator**
   - Verilog-XL
   - NC-Verilog

9. **Verilag-XL**
   - behavior.v
   - rtl.v
   - gate.v
   - gate_scan.v

10. **NC-Verilog**
    - .gds2
Synthesis

specifications

RTL Coding
Your brain

Synthesis
Design Compiler

Dft Insertion
dft compiler

ATPG
Tetramax

.gate
.gate_scan.v
.spf

Place and route
SoC Encounter/ IC Compiler

Tape out

behavior.v

rtl.v

gate.v

gate_scan.v

.stil
.wgl

.gds2
Design for Testability

specifications

RTL Coding
Your Brain

Synthesis
Design Compiler

dft compiler

Dft Insertion

ATPG
Tetramax

Place and route
SoC Encounter/ IC Compiler

Tape out

behavior.v

rtl.v

gate.v

gate_scan.v

.spf

.stil

.wgl

.gds2
Automatic Test Pattern Generation

1. **specifications**
2. **RTL Coding**
   - **Your Brain**
3. **Synthesis**
   - **Design Compiler**
4. **Dft Insertion**
   - **dft compiler**
5. **Place and route**
   - **SoC Encounter/ IC Compiler**
6. **Tape out**
   - **.gds2**
7. **ATPG**
   - **Tetramax**
   - **.stil**
   - **.wgl**
8. **behavior.v**
9. **rtl.v**
10. **gate.v**
11. **gate_scan.v**
12. **.spf**
Place and Route

specifications

RTL Coding
Your brain

Synthesis
Design Compiler

Dft Insertion
dft compiler

ATPG
Tetramax

Place and route
SoC Encounter/ IC Compiler

Tape out

behavior.v

ttl.v

gate.v

gate_scan.v

.stil
.wgl

gate_scan.v

.spf

.gds2
Design Rule Check (DRC)

specifications

RTL Coding
Your Brain

Synthesis
Design Compiler

Dft Insertion
dft compiler

Place and route
SoC Encounter/ IC Compiler

Tape out

DRC
Dracula Calibre
Layout Versus Schematics (LVS)

- specifications
- RTL Coding
  - Your Brain
- Synthesis
  - Design Compiler
- Dft Insertion
  - dft compiler
- Place and route
  - SoC Encounter/IC Compiler
- Tape out
- LVS
  - Dracula Calibre
Layout Parameter Extraction (LPE) with Post Layout Simulation

- Specifications
- RTL Coding
  - Your Brain
- Synthesis
  - Design Compiler
- Dft Insertion
  - dft compiler
- Place and route
  - SoC Encounter/ IC Compiler
- Tape out
- LPE Calibre
- Post layout Sim.
  - powermill/timemill
- behavior.v
- rtl.v
- gate.v
- tectbench
  - tb.v
- .gds2
- .spice
Verification, Verification, Verification...

specifications

RTL Coding
Your Brain

Synthesis
Design Compiler

Dft Insertion
dft compiler

Place and route
Silicon Ensemble/ IC Compiler

Tape out

behavior.v

rtl.v

gate.v

.gate_scan.v

.sdf

Simulation
NC-Verilog

Simulation
NC-Verilog

Simulation
NC-Verilog

Simulation
NC-Verilog

Pre-layout
simulation

post-layout
simulation

RC
Dracula

.sdf
Static Timing Analysis

- **specifications**
- **RTL Coding**
  - Your Brian
- **Synthesis**
  - Design Compiler
- **Dft Insertion**
  - dft compiler
- **Place and route**
  - SoC Encounter/IC Compiler
- **Tape out**
  - .v
  - .sdf
  - .gds2

- **STA**
  - Prime Time
  - gate.v
  - gate_scan.v
  - .sdf

- **STA**
  - Prime Time
  - behavior.v
  - rtl.v

- **STA**
  - Prime Time
  - .sdf
FPGA

FPGA Compiler
Altera Xilinx

FPGA Implementation

specifications

RTL Coding
Verilog-XL

behavior.v

rtl.v

gate.v

gate_scan.v

Synthesis
Design Compiler

Dft Insertion
dft compiler

Place and route
Silicon Ensemble/ IC Compiler

Tape out

.gds2
## Popular Commercial Tools

<table>
<thead>
<tr>
<th>Category</th>
<th>Vendor 1</th>
<th>Vendor 2</th>
<th>Other Vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verilog</td>
<td>Verilog-XL</td>
<td>VCS</td>
<td>Mentor Graphics, ModelSim</td>
</tr>
<tr>
<td></td>
<td>NC Verilog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synthesis</td>
<td>Encounter RTL</td>
<td>Design</td>
<td></td>
</tr>
<tr>
<td></td>
<td>compiler</td>
<td>compiler</td>
<td></td>
</tr>
<tr>
<td>DFT</td>
<td></td>
<td>Dft</td>
<td>Mentor Graphics, DFT</td>
</tr>
<tr>
<td>ATPG</td>
<td></td>
<td>Tetramax</td>
<td>Syntest: TurboScan</td>
</tr>
<tr>
<td>STA</td>
<td></td>
<td>Prime</td>
<td></td>
</tr>
<tr>
<td>Formal Verification</td>
<td>Conformal LEC</td>
<td>Formality</td>
<td></td>
</tr>
<tr>
<td>P&amp;R</td>
<td>SoC Encounter</td>
<td>IC Compiler</td>
<td>Magama</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRC LVS LPE</td>
<td>Dracula</td>
<td>Star RC</td>
<td>Mentor Graphics, Calibre</td>
</tr>
<tr>
<td>FPGA</td>
<td></td>
<td></td>
<td>Xilinx; Altera</td>
</tr>
</tbody>
</table>