Administrative Matters

- **Time/Location:** Tuesday 2:20-5:20pm; EE-II 104
- **Instructor:** Yao-Wen Chang, Chung-Yang Huang, Chien-Mo Li
- **E-mail:** {ywchang, ric, cmlj}@cc.ee.ntu.edu.tw
- **URL:** http://cc.ee.ntu.edu.tw/~eda/Course/IntroEDA06
- **Office:** BL-428; EE-II 444; EE-II 339
- **Office Hours:** Contact Instructors
- **Teaching Assistants**
  - TBD
- **Prerequisites:** Computer Programming & logic design.
  - Other supplementary reading materials will be provided.
Course Objectives

- Study techniques for electronic design automation (EDA), a.k.a. computer-aided design (CAD).
- Study IC technology evolution and their impacts on the development of EDA tools
- **Study problem-solving (-finding) techniques!!!**

\[
\begin{array}{cccccc}
S1 & S2 & S3 & S4 & S5 \\
P1 & P2 & P3 & P4 & P5 & P6
\end{array}
\]

Course Contents

- Introduction to VLSI design flow/styles/automation, technology roadmap, and CMOS Technology
- Algorithmic graph theory
- Computational Complexity and Optimization
- Physical design: partitioning, floorplanning, placement, routing, compaction, deep submicron effects
- Logic synthesis and verification
- (High Level Synthesis)
- Simulation
- Testing
Grading Policy

- Grading Policy:
  - Homework assignments: 25%
  - Midterm Exam: 20%
  - Programming assignment: 25%
  - Final Exam: 30%

- Homework: 50% per day late penalty
  - Due dates on web

- Academic Honesty: Avoiding cheating at all cost.

Unit 1: Introduction

- Course contents:
  - Introduction to VLSI design flow/methodologies/styles
  - Introduction to VLSI design automation tools
  - Semiconductor technology roadmap
  - CMOS technology

- Readings
  - Chapters 1-2
  - Appendix A
Milestones for IC Industry

- **1947**: Bardeen, Brattain & Shockly invented the transistor, foundation of the IC industry.
- **1952**: SONY introduced the first transistor-based radio.
- **1958**: Kilby invented integrated circuits (ICs).
- **1965**: Moore’s law.
- **1968**: Noyce and Moore founded Intel.
- **1970**: Intel introduced 1 K DRAM.

- **1971**: Intel announced 4-bit 4004 microprocessors (2250 transistors).
- **1976/81**: Apple II/IBM PC (technology driver).
- **1984**: Xilinx invented FPGA’s.
- **1985**: Intel began focusing on microprocessor products.
- **1987**: TSMC was founded (fabless IC design).
- **1991**: ARM introduced its first embeddable RISC IP core (chipless IC design).
Milestones for IC Industry (Cont’d)

- **1996**: Samsung introduced IG DRAM.
- **1998**: IBM announces 1GHz experimental microprocessor.
- **1999/earlier**: System-on-Chip (SoC) applications.
- **2002/earlier**: System-in-Package (SiP) technology.
- An Intel P4 processor contains 42 million transistors (1 billion in 2005)
- Today, we produce > 30 million transistors per person (1 billion/person by 2008).

---

SoC Architecture

- An SoC system typically consists of a collection of components/subsystems that are appropriately interconnected to perform specified functions for users.
IC Design & Manufacturing Process

From Wafer to Chip

Wafer: Place of Making Dies

Figure 1.1: Removed wafer stock to expose a single slice of silicon. This exposed region is then chemically etched to form an IC circuit.
Traditional VLSI Design Cycles

1. System specification
2. Functional design
3. Logic synthesis
4. Circuit design
5. Physical design and verification
6. Fabrication
7. Packaging

• Other tasks involved: testing, simulation, etc.
• Design metrics: area, speed, power dissipation, noise, design time, testability, etc.
• Design revolution: interconnect (not gate) delay dominates circuit performance in deep submicron era.
  – Interconnects are determined in physical design.
  – Shall consider interconnections in early design stages.
Design Actions

- **Synthesis**: increasing information about the design by providing more detail (e.g., logic synthesis, physical synthesis).
- **Analysis**: collecting information on the quality of the design (e.g., timing analysis).
- **Verification**: checking whether a synthesis step has left the specification intact (e.g., layout verification).
- **Optimization**: increasing the quality of the design by rearrangements in a given description (e.g., logic optimizer, timing optimizer).
- **Design Management**: storage of design data, cooperation between tools, design flow, etc. (e.g., database).
Design Issues and Tools

- **System-level design**
  - Partitioning into hardware and software, co-design, co-simulation, etc.
  - Cost estimation, design-space exploration

- **Algorithmic-level design**
  - Behavioral descriptions (e.g., in Verilog, VHDL)
  - High-level simulation

- **From algorithms to hardware modules**
  - High-level (or architectural) synthesis

- **Logic design:**
  - Schematic entry
  - Register-transfer level and logic synthesis
  - Gate-level simulation (functionality, power, etc.)
  - Timing analysis
  - Formal verification

Logic Design/Synthesis

- **Logic synthesis** programs transform Boolean expressions into logic gate networks in a particular library.
- Optimization goals: minimize area, delay, power, etc
- **Technology-independent** optimization: logic optimization
  - Optimizes Boolean expression equivalent.
- **Technology-dependent** optimization: technology mapping/library binding
  - Maps Boolean expressions into a particular cell library.
Logic Optimization Examples

- **Two-level**: minimize the # of product terms.
  \[ F = \bar{x_1}x_2x_3 + \bar{x_1}x_2x_3 + x_1\bar{x_2}x_3 + x_1x_2x_3 + x_1x_2x_3 \Rightarrow F = \bar{x_2} + x_1x_3. \]

- **Multi-level**: minimize the #'s of literals, variables.
  - E.g., equations are optimized using a smaller number of literals.

  - E.g., equations are optimized using a smaller number of literals.

  \[ t1 = a + b + c; \]
  \[ t2 = d + e; \]
  \[ t3 = a + b + d; \]
  \[ t4 = t1 \cdot t2 + f \cdot g; \]
  \[ t5 = t4 + h + t2 \cdot t3; \]
  \[ F = t5; \]

- Methods/CAD tools: Quine-McCluskey method (exponential-time exact algorithm), Espresso (heuristics for two-level logic), MIS (heuristics for multi-level logic), Synopsys, etc.

Design Issues and Tools (Cont’d)

- **Transistor-level design**
  - Switch-level simulation
  - Circuit simulation

- **Physical (layout) design**
  - Partitioning
  - Floorplanning and Placement
  - Routing
  - Layout editing and compaction
  - Design-rule checking
  - Layout extraction

- **Design management**
  - Data bases, frameworks, etc.

- **Silicon compilation**: from algorithm to mask patterns
  - The *idea* is approached more and more, but still far away from a single *push-bottom* operation
Circuit Simulation of a CMOS Inverter (0.6 μm)

M1 3 2 0 0 nch W=1.2u L=0.6u AS=2.16p PS=4.8u AD=2.16p PD=4.8u
M2 3 2 1 1 pkg W=1.8u L=0.6u AS=3.24p PS=5.4u AD=3.24p PD=5.4u
CL 3 0 0.2pF

VDD 1 0 3.3
VIN 2 0 DC 0 PULSE (0 3.3 0ns 100ps 100ps 2.4ns 5ns)

.LIB '../mod.06' typical

.OPTION NOMOD POST INGOLD=2 NUMDGT=6 BRIEF
.DC VIN OV 3.3V 0.001V
.PRINT DC V(3)
.TRAN 0.001V 5N
.PRINT TRAN V(2) V(3)
.END

Physical Design

- Physical design converts a circuit description into a geometric description.
- The description is used to manufacture a chip.
- Physical design cycle:
  1. Logic partitioning
  2. Floorplanning and placement
  3. Routing
  4. Compaction
- Others: circuit extraction, timing verification and design rule checking
Physical Design Flow

**Physical Design**

- Partitioning
- Floorplanning & Placement
- Routing
- Compaction
- Extraction & Verification

**B*-tree based floorplanning system**

A routing system

Floorplan Examples

- PowerPC 604
- Intel Pentium 4

A floorplan with interconnections
Routing Example

• 0.18um technology, two layers, pitch = 1 um, 8109 nets.

Testing

• Goal of testing is to ensure defect-free products.
• Need high quality tests that can detect realistic defects
• Varieties of testing: functional testing, performance testing
IC Design Considerations

- Several conflicting considerations:
  - **Design Complexity**: large number of devices/transistors
  - **Performance**: optimization requirements for high performance
  - **Time-to-market**: about a 15% gain for early birds
  - **Cost**: die area, packaging, testing, etc.
  - **Others**: power, signal integrity (noise, etc.), testability, reliability, manufacturability, etc.

“Moore’s” Law: Driving Technology Advances

- Logic capacity doubles per IC at a regular interval.
- Moore: Logic capacity doubles per IC every two years (1975).
- D. House: Computer performance doubles every 18 months (1975)
Technology Roadmap for Semiconductors

- Deep submicron technology: node (feature size) < 0.25 μm.
- Nanometer Technology: node < 0.1 μm.

### ITRS 2004 Technology Roadmap

<table>
<thead>
<tr>
<th>Year</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology Node (nm)</td>
<td>100</td>
<td>80</td>
<td>70</td>
<td>65</td>
<td>67</td>
<td>70</td>
<td>50</td>
</tr>
<tr>
<td>MPU 4-transistor Area factor **</td>
<td>120</td>
<td>117.8</td>
<td>115.8</td>
<td>113.7</td>
<td>111.9</td>
<td>116.4</td>
<td>109</td>
</tr>
<tr>
<td>Logic Gate (4-transistor) Area factor **</td>
<td>320</td>
<td>320</td>
<td>320</td>
<td>320</td>
<td>320</td>
<td>320</td>
<td>320</td>
</tr>
<tr>
<td>SEAM Cell (n-transistor) Area efficiency **</td>
<td>0.63</td>
<td>0.63</td>
<td>0.63</td>
<td>0.63</td>
<td>0.63</td>
<td>0.63</td>
<td>0.63</td>
</tr>
<tr>
<td>Logic Gate (4-transistor) Area efficiency **</td>
<td>0.50</td>
<td>0.60</td>
<td>0.60</td>
<td>0.60</td>
<td>0.60</td>
<td>0.60</td>
<td>0.60</td>
</tr>
</tbody>
</table>

**NEW**
- SEAM Cell (n-transistor) Area ** | 1.27 | 2.59 | 2.06 | 1.63 | 1.30 | 1.63 | 2.83 |
- Logic Gate (4-transistor) Area ** | 2.5 | 2.7 | 2.0 | 1.63 | 1.30 | 1.63 | 2.83 |

**WAS**
- Functions per chip at introduction (million transistors [International]) | 515 | 571 | 719 |
- Functions per chip at introduction (million transistors [International]) | 287 | 287 | 287 |
- Functions per chip at introduction (million transistors [International]) | 400 | 400 | 400 |

**IS**
- Chip size at introduction (mm²) | 0.268 | 0.268 | 0.268 |
- Cost performance MPU (micrometers² or introduction) including on-chip SRAM | 0.110 | 0.138 | 0.174 | 0.219 | 0.276 | 0.346 | 0.438 |
- Generation or production ** | 0.464 | 0.86 | 0.86 |
- Functions per chip at introduction (million transistors [International]) | 140 | 140 | 140 | 140 | 140 | 140 | 140 |
- Chip size at production (mm²) | 0.268 | 0.268 | 0.268 |
- Cost performance MPU (micrometers² or production) including on-chip SRAM | 0.110 | 0.138 | 0.174 | 0.219 | 0.276 | 0.346 | 0.438 |
**ITRS 2004 Technology Roadmap (cont’d)**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WAS DDRAM ½ Pitch (nm)</td>
<td>25</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS DDRAM ½ Pitch (nm)</td>
<td>25</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAS Generation at introduction *</td>
<td>p1.3c</td>
<td>—</td>
<td>p1.6c</td>
<td>—</td>
<td>p1.6c</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS Generation at introduction *</td>
<td>p1.3c</td>
<td>—</td>
<td>p1.6c</td>
<td>—</td>
<td>p1.6c</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAS Functions per chip at introduction (million transistors)</td>
<td>1,546</td>
<td>1,546</td>
<td>2,454</td>
<td>3,092</td>
<td>4,038</td>
<td>6,184</td>
<td>8,816</td>
<td>5,816</td>
<td>5,816</td>
</tr>
<tr>
<td>IS Functions per chip at introduction (million transistors)</td>
<td>1,546</td>
<td>1,546</td>
<td>2,454</td>
<td>3,092</td>
<td>4,038</td>
<td>6,184</td>
<td>8,816</td>
<td>5,816</td>
<td>5,816</td>
</tr>
<tr>
<td>WAS Chip size at introduction (mm²)</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>140</td>
<td>140</td>
</tr>
<tr>
<td>IS Chip size at introduction (mm²)</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>140</td>
<td>140</td>
</tr>
<tr>
<td>WAS Cost performance MPU (Mtransistors/mm² at introduction) (including on-chip SRAM)</td>
<td>652</td>
<td>114</td>
<td>1,104</td>
<td>1,104</td>
<td>1,104</td>
<td>1,104</td>
<td>1,104</td>
<td>1,104</td>
<td>1,104</td>
</tr>
<tr>
<td>IS Cost performance MPU (Mtransistors/mm² at introduction) (including on-chip SRAM)</td>
<td>652</td>
<td>114</td>
<td>1,104</td>
<td>1,104</td>
<td>1,104</td>
<td>1,104</td>
<td>1,104</td>
<td>1,104</td>
<td>1,104</td>
</tr>
<tr>
<td>WAS Generation at production *</td>
<td>p1.3c</td>
<td>—</td>
<td>p1.6c</td>
<td>—</td>
<td>p1.6c</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS Generation at production *</td>
<td>p1.3c</td>
<td>—</td>
<td>p1.6c</td>
<td>—</td>
<td>p1.6c</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAS Functions per chip at production (million transistors)</td>
<td>773</td>
<td>974</td>
<td>1,227</td>
<td>1,546</td>
<td>2,454</td>
<td>3,092</td>
<td>3,996</td>
<td>4,038</td>
<td>4,038</td>
</tr>
<tr>
<td>IS Functions per chip at production (million transistors)</td>
<td>773</td>
<td>974</td>
<td>1,227</td>
<td>1,546</td>
<td>2,454</td>
<td>3,092</td>
<td>3,996</td>
<td>4,038</td>
<td>4,038</td>
</tr>
<tr>
<td>WAS Chip size at production (mm²)</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>140</td>
<td>140</td>
</tr>
<tr>
<td>IS Chip size at production (mm²)</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>140</td>
<td>140</td>
</tr>
<tr>
<td>WAS Cost performance MPU (Mtransistors/mm² at production, including on-chip SRAM)</td>
<td>652</td>
<td>686</td>
<td>816</td>
<td>1,104</td>
<td>1,304</td>
<td>1,753</td>
<td>2,209</td>
<td>2,783</td>
<td>5,816</td>
</tr>
<tr>
<td>IS Cost performance MPU (Mtransistors/mm² at production, including on-chip SRAM)</td>
<td>652</td>
<td>686</td>
<td>816</td>
<td>1,104</td>
<td>1,304</td>
<td>1,753</td>
<td>2,209</td>
<td>2,783</td>
<td>5,816</td>
</tr>
</tbody>
</table>

---

**Design Complexity Increases Dramatically!!**
Power Is Another Big Problem!!

- Power density increases exponentially!


Interconnect Dominates Circuit Performance!!

In \( \leq 0.18 \mu m \) wire-to-wire capacitance dominates \((C_W \gg C_S)\)

\[ P=VI: \ 75W @ 1.5V = 50A! \]

Worst-case interconnect delay due to crosstalk

Interconnect delay

Gate delay

\[ 650 \ 500 \ 350 \ 250 \ 180 \ 150 \ 100 \ 70 \ (nm) \]

Technology Node
**Lithography Process**

**Sub-wavelength Lithography Causes Problems!!**

- **Mask patterns**
- **Printed layout**
- **Drawn layout**
- **Printed wafer**
- **Proximity corrected layout**
- **Printed wafer**
Design Productivity Crisis

- Human factors may limit design more than technology.
- Keys to solve the productivity crisis: CAD (tool & methodology), hierarchical design, abstraction, IP reuse, platform-based design, etc.

Hierarchical Design

- **Hierarchy**: something is composed of simpler things.
- Design cannot be done in one step ⇒ partition the design hierarchically.
Abstraction

- **Abstraction**: when looking at a certain level, you don’t need to know all details of the lower levels.

Design domains:
- **Behavioral**: black box view
- **Structural**: interconnection of subblocks
- **Physical**: layout properties

Each design domain has its own hierarchy.

---

Three Design Views

Behavior

```verilog
module add4 (s, c4, ci, a, b);
  input [3:0] a, b;
  input ci;
  output [3:0] s;
  output c4;
  wire [2:0] co;
  add f0 (co[0], s[0], a[0], b[0], ci);
  add f1 (co[1], s[1], a[1], b[1], co[0]);
  add f2 (co[2], s[2], a[2], b[2], co[1]);
  add f3 (c4, s[3], a[3], b[3], co[2]);
endmodule
```

Structural

Physical

```text
(400, 400)  (100, 100)  (0, 0)
```

---
Gajski’s Y-Chart

BEHAVIORAL DOMAIN
- Systems
- Algorithms
- Register transfers
- Logic
- Transfer functions

STRUCTURAL DOMAIN
- Processors
- ALU’s, RAM, etc.
- Gates, flip-flops, etc.
- Transistors
- Transistor layout
- Cell layout
- Module layout
- Floor plans
- Physical partitions

PHYSICAL DOMAIN

Top-Down Structural Design

BEHAVIORAL DOMAIN
- Systems
- Algorithms
- Register transfers
- Logic
- Transfer functions

STRUCTURAL DOMAIN
- Processors
- ALU’s, RAM, etc.
- Gates, flip-flops, etc.
- Transistors
- Transistor layout
- Cell layout
- Module layout
- Floor plans
- Physical partitions

PHYSICAL DOMAIN
Design Styles

- Specific design styles shall require specific CAD tools

**Issues of VLSI circuits**

- Performance
- Area
- Cost
- Time-to-market

**Different design styles**

- Full custom
- Standard cell
- Gate array
- FPGA
- CPLD
- SPLD
- SSI

Performance, Area efficiency, Cost, Flexibility

---

**SSI/SPLD Design Style**

(a) 4-bit comparator.

(b) SSI implementation.

(c) SPLD (PLA) implementation.
Full Custom Design Style

- Designers can control the shape of all mask patterns.
- Designers can specify the design up to the level of individual transistors.

Standard Cell Design Style

- Selects pre-designed cells (of the same height) to implement logic
- Over-the-cell routing is pervasive in modern designs
**Standard Cell Example**

![Diagram of Standard Cell Example](image)

*Courtesy of Newton/Pister, UC-Berkeley*

---

**Gate Array Design Style**

- Prefabricates a transistor array
- Needs wiring customization to implement logic

![Diagram of Gate Array Design Style](image)
FPGA Design Style

- Logic and interconnects are both prefabricated.
- Illustrated by a symmetric array-based FPGA

Array-Based FPGA Example

Lucent Technologies 15K ORCA FPGA, 1995
- 0.5 um 3LM CMOS
- 2.45 M Transistors
- 1600 Flip-flops
- 25K bit user RAM
- 320 I/Os

Fujitsu’s non-volatile Dynamically Programmable Gate Array (DPGA), 2002
**FPGA Design Process**

- Illustrated by a symmetric array-based FPGA
- No fabrication is needed

---

**Comparisons of Design Styles**

<table>
<thead>
<tr>
<th></th>
<th>Full custom</th>
<th>Standard cell</th>
<th>Gate array</th>
<th>FPGA</th>
<th>SPLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell size</td>
<td>variable</td>
<td>fixed height*</td>
<td>fixed</td>
<td>fixed</td>
<td>fixed</td>
</tr>
<tr>
<td>Cell type</td>
<td>variable</td>
<td>variable</td>
<td>fixed</td>
<td>programmable</td>
<td>programmable</td>
</tr>
<tr>
<td>Cell placement</td>
<td>variable</td>
<td>in row</td>
<td>fixed</td>
<td>fixed</td>
<td>fixed</td>
</tr>
<tr>
<td>Interconnections</td>
<td>variable</td>
<td>variable</td>
<td>variable</td>
<td>programmable</td>
<td>programmable</td>
</tr>
</tbody>
</table>

* Uneven height cells are also used.

<table>
<thead>
<tr>
<th></th>
<th>Full custom</th>
<th>Standard cell</th>
<th>Gate array</th>
<th>FPGA</th>
<th>SPLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fabrication time</td>
<td></td>
<td></td>
<td></td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Packing density</td>
<td>+++</td>
<td>++</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit cost in large quantity</td>
<td>+++</td>
<td>++</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit cost in small quantity</td>
<td>--</td>
<td>--</td>
<td>+</td>
<td>+++</td>
<td>+</td>
</tr>
<tr>
<td>Easy design and simulation</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>+++</td>
<td>++</td>
</tr>
<tr>
<td>Easy design change</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>+++</td>
<td>++</td>
</tr>
<tr>
<td>Accuracy of timing simulation</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Chip speed</td>
<td>+++</td>
<td>++</td>
<td>+</td>
<td>-</td>
<td>--</td>
</tr>
</tbody>
</table>

+ desirable; – not desirable
The Structured ASIC Is Coming!

- A structured ASIC consists of predefined metal and via layers, as well as a few of them for customization.
- The predefined layers support power distribution and local communications among the building blocks of the device.
- Advantages: fewer masks (lower cost); easier physical extraction and analysis.

A structured ASIC (M5 & M6 can be customized)

Faraday’s 3MPCA structured ASIC (M4–M6 can be customized)
Complementary MOS (CMOS)

- The most popular VLSI technology (vs. BiCMOS, nMOS).
- CMOS uses both $n$-channel and $p$-channel transistors.
- Advantages: lower power dissipation, higher regularity, more reliable performance, higher noise margin, larger fanout, etc.
- Each type of transistor must sit in a material of the complementary type (the reverse-biased diodes prevent unwanted current flow).
A CMOS Inverter

A CMOS inverter.

CMOS Inverter Structure

A CMOS inverter.
A CMOS NAND Gate

A CMOS NOR Gate
### Basic CMOS Logic Library

<table>
<thead>
<tr>
<th>Name</th>
<th>Ductive shape</th>
<th>Algebraic equation</th>
<th>Cost (# of transistors)</th>
<th>Scaled gate delay (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>![AND symbol]</td>
<td>( F = XY )</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>OR</td>
<td>![OR symbol]</td>
<td>( F = X + Y )</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>NOT (inverter)</td>
<td>![NOT symbol]</td>
<td>( F = \overline{X} )</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Buffer (driver/repeater)</td>
<td>![Buffer symbol]</td>
<td>( F = X )</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>NAND</td>
<td>![NAND symbol]</td>
<td>( F = XY )</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>NOR</td>
<td>![NOR symbol]</td>
<td>( F = \overline{X+Y} )</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>Exclusive-OR (XOR)</td>
<td>![XOR symbol]</td>
<td>( F = \overline{X+Y} + XY )</td>
<td>14</td>
<td>42</td>
</tr>
</tbody>
</table>

### Construction of Compound Gates

- **Example:** \( F = A \cdot B + C \cdot D \).
- **Step 1** (**n-network**): **Invert** \( F \) to derive **n-network** \( \overline{F} = A \cdot B + C \cdot D \)
- **Step 2** (**n-network**): Make connections of transistors:
  - **AND** \& Series connection
  - **OR** \& Parallel connection

![CMOS compound gate diagram]
**Construction of Compound Gates (cont’d)**

- **Step 3 (p-network):** Expand $F$ to derive $p$-network
  - $(F = \overline{AB} + CD = \overline{AB} \cdot \overline{CD} = (A + \overline{B}) \cdot (\overline{C} + D))$
  - each input is inverted
- **Step 4 (p-network):** Make connections of transistors (same as Step 2).
- **Step 5:** Connect the $n$-network to GND (typically, 0V) and the $p$-network to VDD (5V, 3.3V, or 2.5V, etc).

---

**A Complex CMOS Gate**

- The functions realized by the $n$ and $p$ networks must be complementary, and one of the networks must conduct for every input combination.
- Duality is not necessary.
**CMOS Properties**

- There is always a path from one supply (VDD or GND) to the output.
- There is never a path from one supply to the other. (This is the basis for the low power dissipation in CMOS—virtually no static power dissipation.)
- There is a momentary drain of current (and thus power consumption) when the gate switches from one state to another.
  - Thus, CMOS circuits have dynamic power dissipation.
  - The amount of power depends on the switching frequency.

**Stick Diagram**

- Intermediate representation between the transistor level and the mask (layout) level.
- Gives topological information (identifies different layers and their relationship)
- Assumes that wires have no width.
- Possible to translate stick diagram automatically to layout with correct design rules.
Stick Diagram (cont'd)

- When the same materials (on the same layer) touch or cross, they are connected and belong to the same electrical node.

- When **polysilicon** crosses N or P **diffusion**, an N or P transistor is formed.
  - Polysilicon is drawn on top of diffusion.
  - Diffusion must be drawn connecting the source and the drain.
  - Gate is automatically self-aligned during fabrication.

- When a metal line needs to be connected to one of the other three conductors, a **contact cut (via)** is required.

CMOS Inverter Stick Diagrams

- **Basic layout**

- **More area efficient layout**
Design Rules

- Layout rules are used for preparing the masks for fabrication.
- Fabrication processes have inherent limitations in accuracy.
- Design rules specify geometry of masks to optimize yield and reliability (trade-offs: area, yield, reliability).
- Three major rules:
  - **Wire width**: Minimum dimension associated with a given feature.
  - **Wire separation**: Allowable separation.
  - **Contact**: overlap rules.
- Two major approaches:
  - **“Micron” rules**: stated at micron resolution.
  - **λ rules**: simplified micron rules with limited **scaling** attributes.
- λ may be viewed as the size of minimum feature.
- Design rules represent a tolerance which insures very high probability of correct fabrication (not a hard boundary between correct and incorrect fabrication).
- Design rules are determined by experience.
Example CMOS Design Rules

MOSIS Layout Design Rules

- MOSIS design rules (SCMOS rules) are available at http://www.mosis.org.
- 3 basic design rules: wire width, wire separation, contact rule.
- MOSIS design rule examples

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Min active area width</td>
<td>3 λ</td>
</tr>
<tr>
<td>R3</td>
<td>Min poly width</td>
<td>2 λ</td>
</tr>
<tr>
<td>R4</td>
<td>Min poly spacing</td>
<td>2 λ</td>
</tr>
<tr>
<td>R5</td>
<td>Min gate extension of poly over active</td>
<td>2 λ</td>
</tr>
<tr>
<td>R8</td>
<td>Min metal width</td>
<td>3 λ</td>
</tr>
<tr>
<td>R9</td>
<td>Min metal spacing</td>
<td>3 λ</td>
</tr>
<tr>
<td>R10</td>
<td>Poly contact size</td>
<td>2 λ</td>
</tr>
<tr>
<td>R11</td>
<td>Min poly contact spacing</td>
<td>2 λ</td>
</tr>
</tbody>
</table>