# Chapter 1

# Introduction

# The Story Begins

| Integration Scale | Number of Components | Examples |
|---|---|---|
| SSI | < 10 transistors | Logic gates |
| MSI | 10 - 1,000 transistors | Adders, counters |
| LSI | 1,000 - 10,000 transistors | Multipliers |
| VLSI | > 10,000 transistors | Microprocessors |

Fig. 1.1 Integration scales.

# Moore's Law

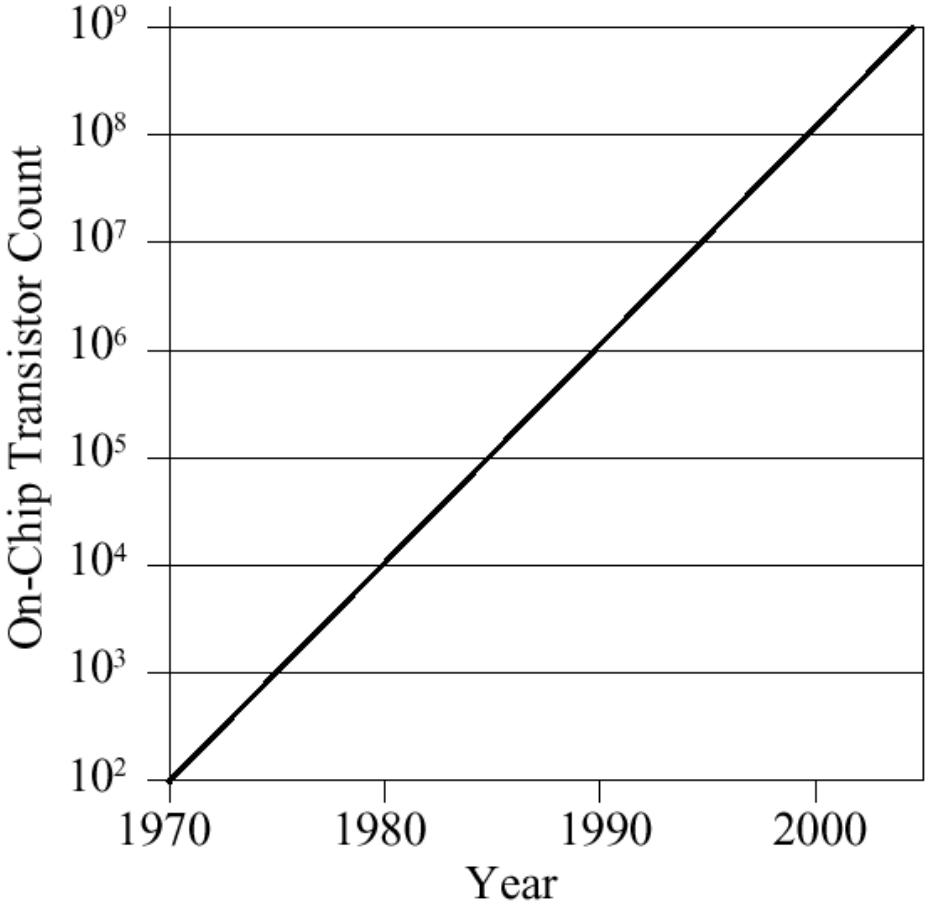



Fig. 1.2 Moore's law.

# VLSI Design Methodology

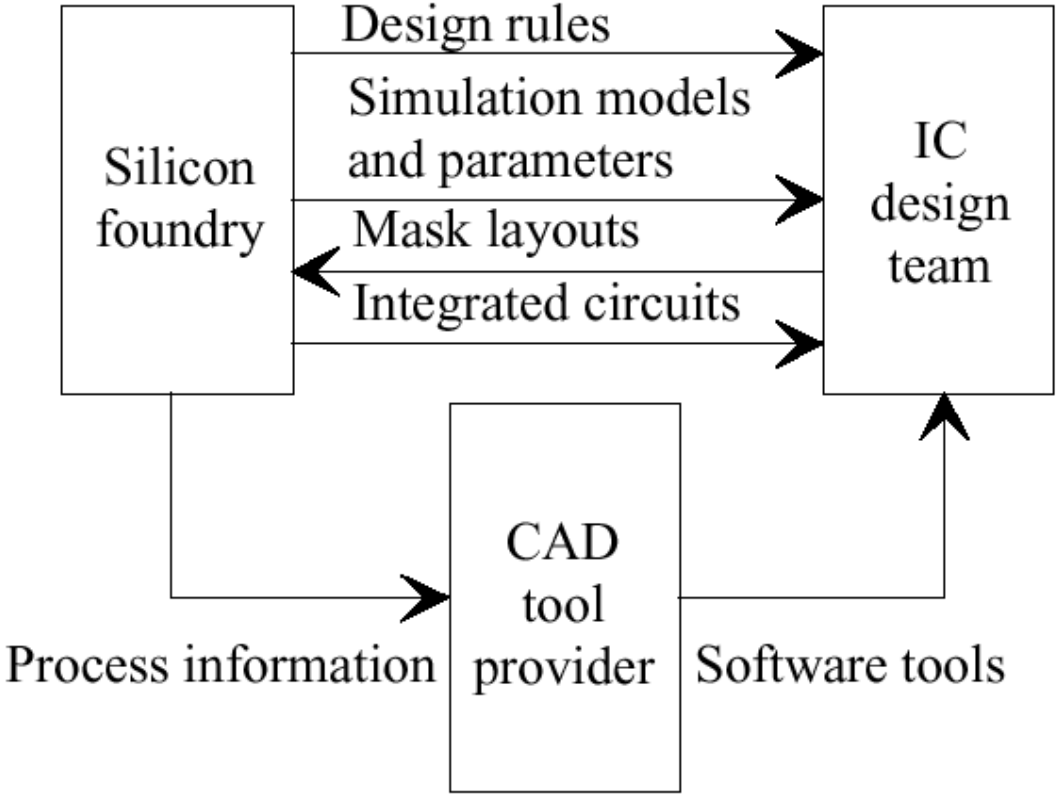



Fig. 1.3 Relationship between a silicon foundry, an IC design team, and a CAD tool provider.

# Behavioral (Functional) Design

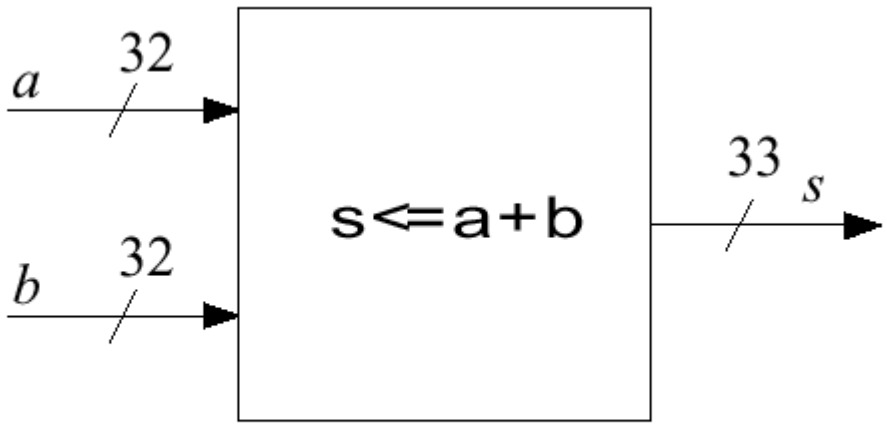



Fig. 1.4 Graphical behavioral representation of an adder.

# Behavioral (Functional) Design

```
-- Interface
ENTITY adder IS
  PORT (a, b: IN unsigned(0 to 31);
    s: OUT unsigned(0 to 32));
END adder;

-- Behavioral representation
ARCHITECTURE behavioral OF adder IS
BEGIN
  s <= a + b;
END behavioral;
```

# Structural (Architectural) Design

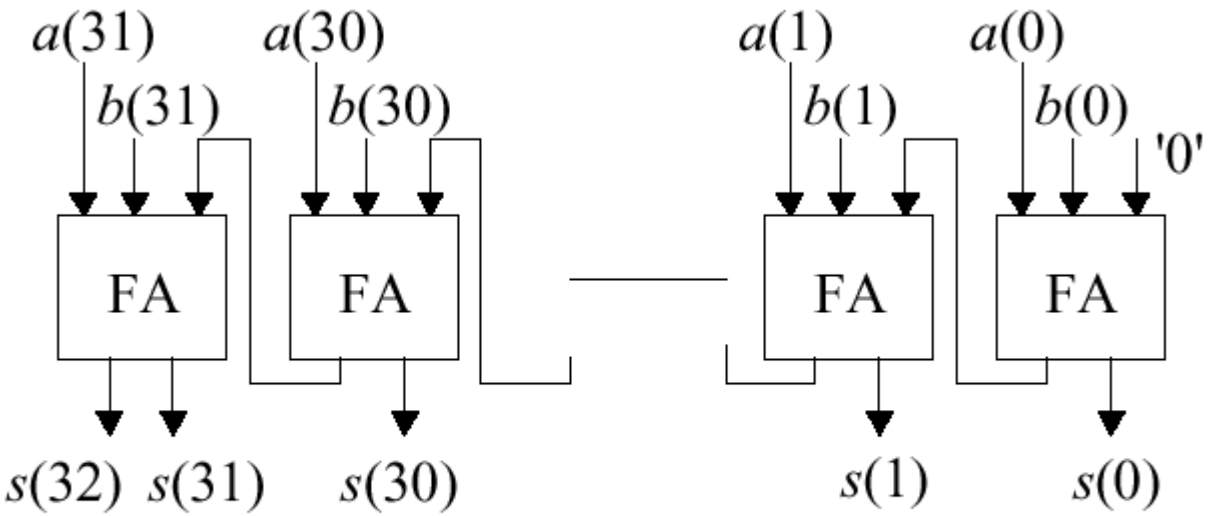



Fig. 1.5 Ripple-carry adder.

# Logic Design

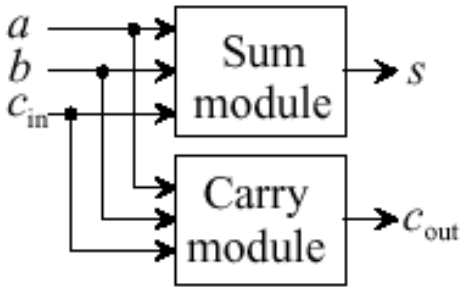



Fig. 1.6 I/O interface of a 1-bit full adder.

$$s = a \oplus b \oplus c_{in}$$

| $a$ | $b$ | $c_{in}$ | $c_{out}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

# VHDL Design

```
-- Interface of a full adder (Part 1)
ENTITY full_adder IS
  PORT (a, b, cin: IN BIT; s, cout: OUT BIT);
END full_adder;

-- Behavioral representation of a full adder (Part 2)

ARCHITECTURE behavioral OF full_adder IS
-- carry() and sum()
  FUNCTION carry(a, b, cin: BIT) RETURN BIT IS
    BEGIN
      RETURN (a AND b) OR (b AND cin) OR (a AND cin);
    END carry;
  FUNCTION sum(a, b, cin: BIT) RETURN BIT IS
    BEGIN
      RETURN a XOR b XOR cin;
    END sum;
-- Relationship between the outputs and inputs
-- with propagation delay information
BEGIN
  s <= sum(a, b, cin) AFTER 1 ns;
  cout <= carry(a, b, cin) AFTER 0.8 ns;
END behavioral;
```

# VHDL Design

```
-- Structural representation of a full adder (Part 3)
ARCHITECTURE structural OF full_adder IS
-- Component list
COMPONENT xor_cell
  PORT (a, b: IN BIT; z: OUT BIT);
END COMPONENT;
COMPONENT or_cell
  PORT (a, b: IN BIT; z: OUT BIT);
END COMPONENT;
COMPONENT and_cell
  PORT (a, b: IN BIT; z: OUT BIT);
END COMPONENT;
```

# VHDL Design

```
-- Sources of components
FOR ALL:xor_cell USE ENTITY WORK.lib_xor(structural);
FOR ALL:or_cell USE ENTITY WORK.lib_or(structural);
FOR ALL:and_cell USE ENTITY WORK.lib_and(structural);

-- Internal wires
SIGNAL wire1, wire2, wire3, wire4, wire5: BIT;

-- Connections
BEGIN
  a1: xor_cell PORT MAP(a, b, wire1);
  a2: xor_cell PORT MAP(wire1, cin, s);
  a3: and_cell PORT MAP(a, b, wire2);
  a4: and_cell PORT MAP(a, cin, wire3);
  a5: and_cell PORT MAP(b, cin, wire4);
  a6: or_cell PORT MAP(wire2, wire3, wire5);
  a7: or_cell PORT MAP(wire4, wire5, cout);
END structural
```

# Physical Design

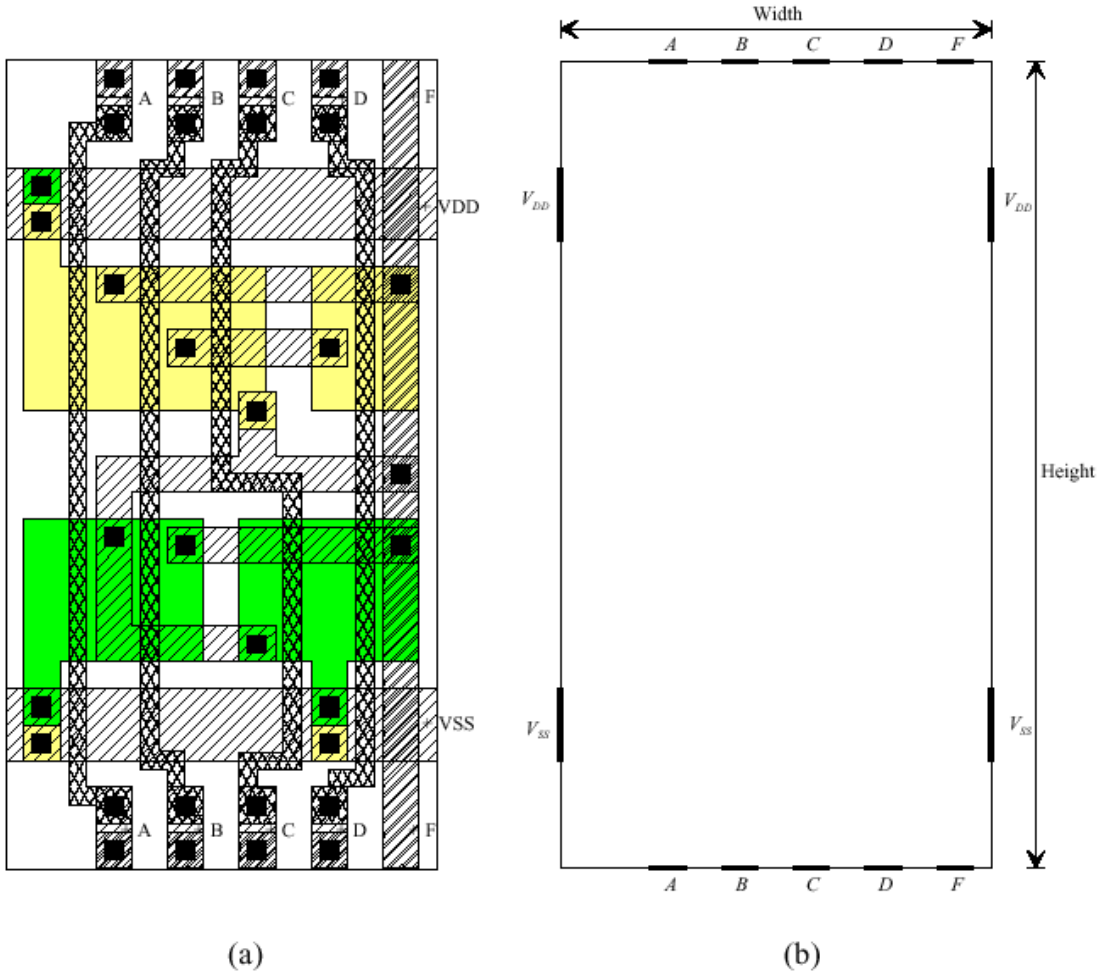



Fig. 1.7 (a) Physical design; (b) Designer's view of a standard cell.

# Standard Cell Design

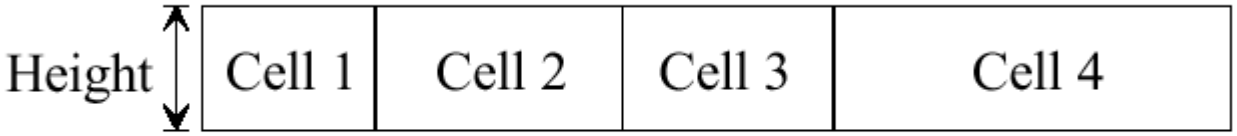



Fig. 1.8 Arrangement of standard-cells in a row.

# Standard Cell Design

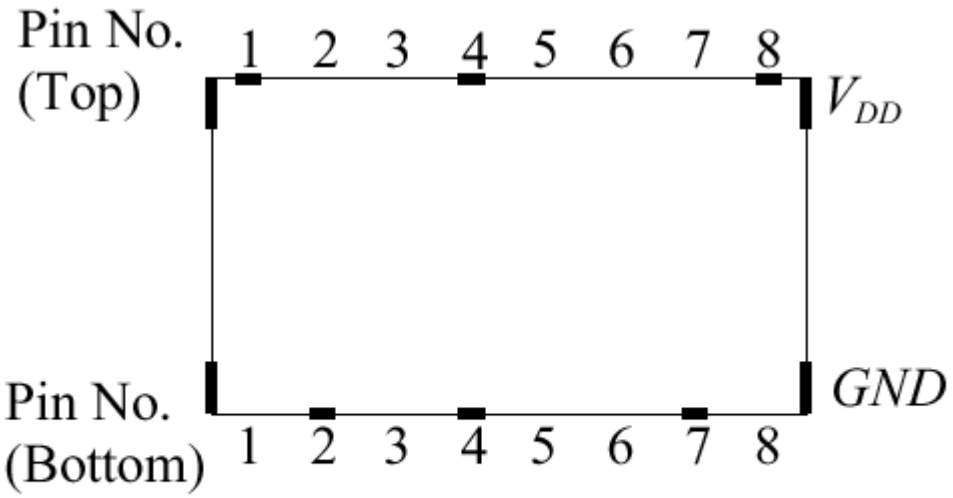
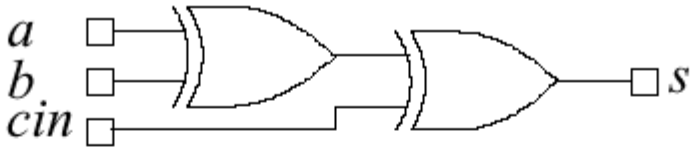



Fig. 1.9 Pin numbers in a standard-cell.

# Netlist




```
(a, XOR[1].IN1)
(b, XOR[1].IN2)
(XOR[1].OUT, XOR[2].IN1)
(cin, XOR[2].IN2)
(s, XOR[2].OUT)
(XOR[1].Vdd, XOR[2].Vdd)
(XOR[1].GND, XOR[2].GND)
```

Fig. 1.10 Logic diagram of a circuit and its netlist.

# Placement and Routing

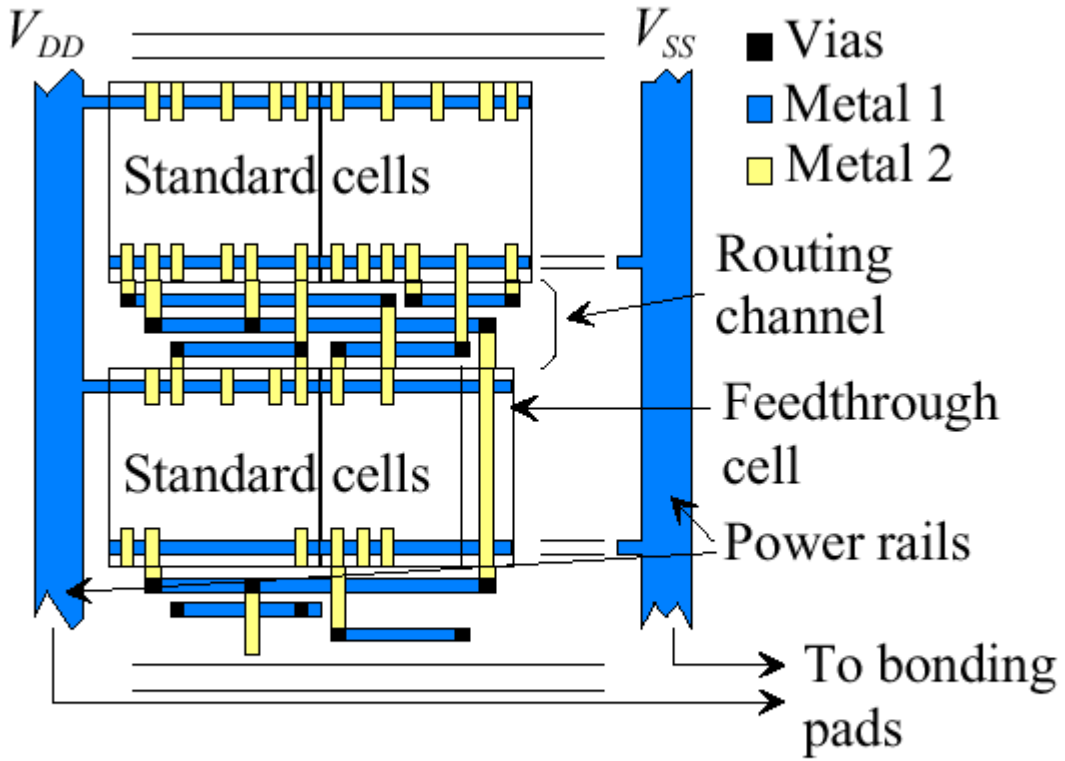



Fig. 1.11 Placement and routing of standard-cells.

# Floorplan

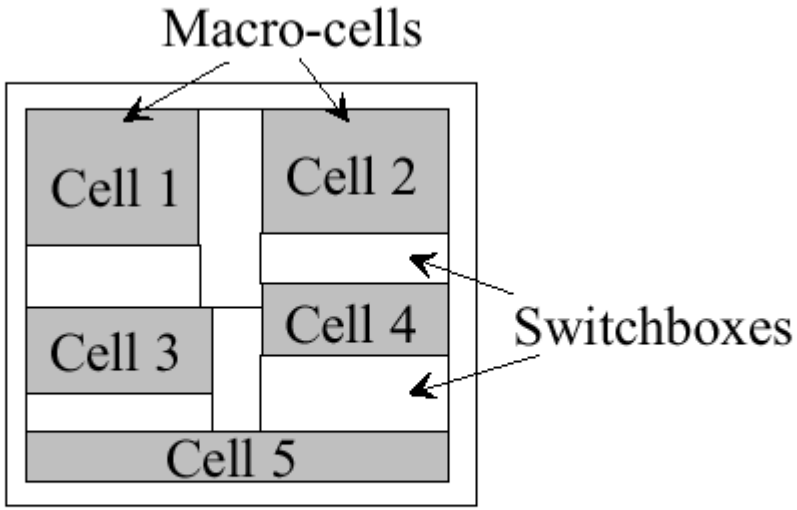



Fig. 1.12 Floorplan of a layout using macro-cells.

# Switchbox Routing

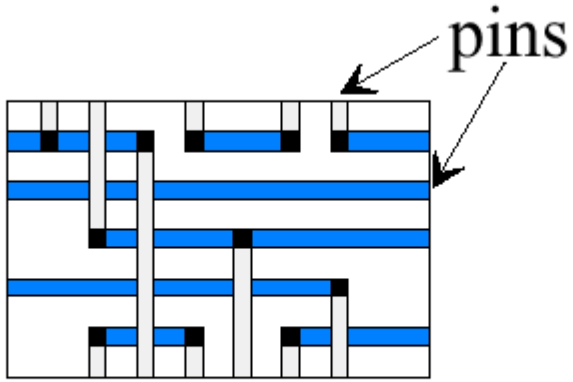



Fig. 1.13 Switchbox routing example.

# Fabrication Process

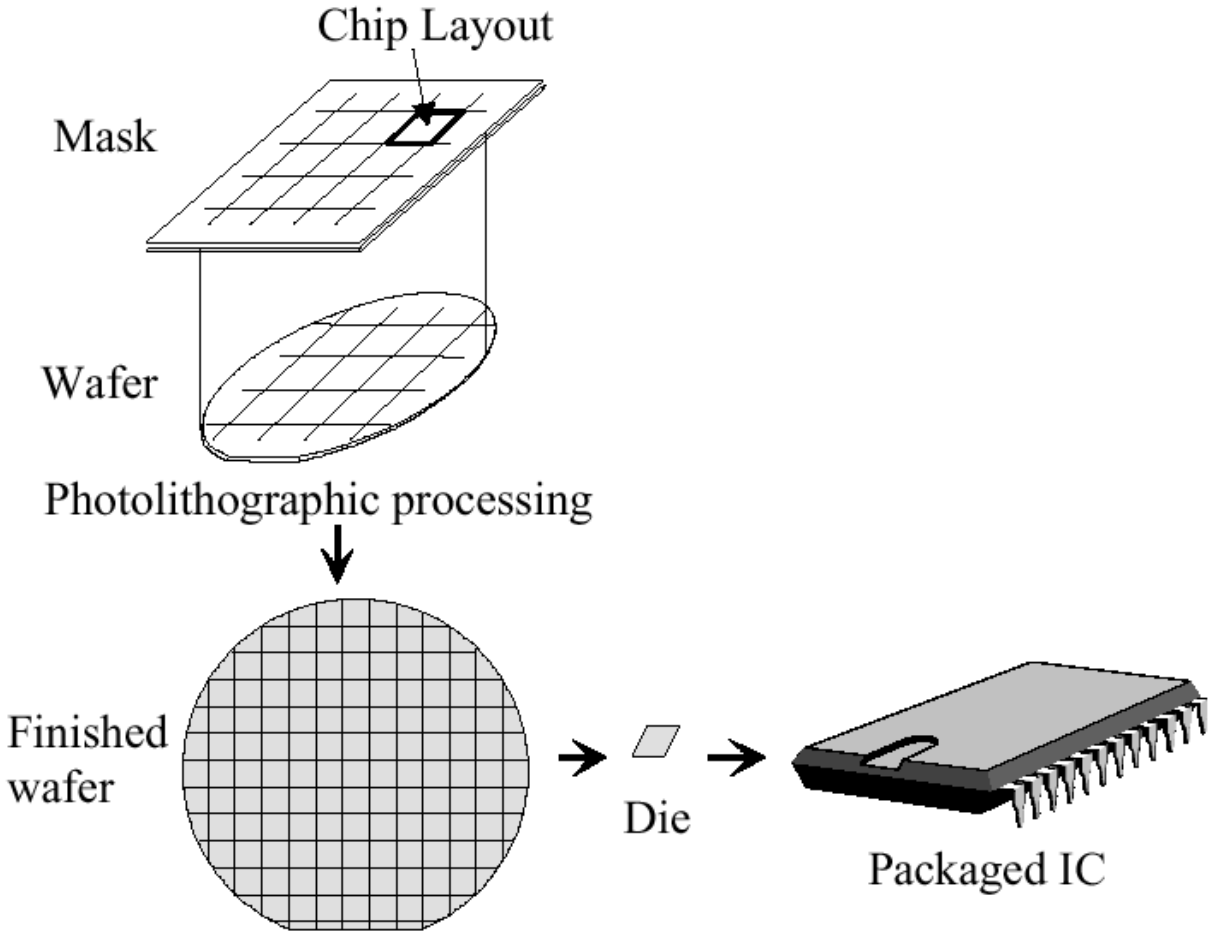



Fig. 1.14 Simplified fabrication process of an integrated circuit.

# Mask Programmable Gate Array

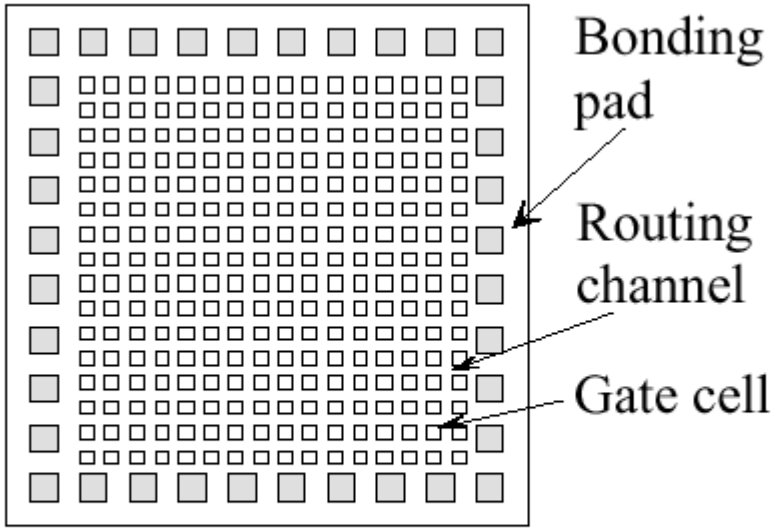



Fig. 1.15 Floorplan of an MPGA.

# Gate Array Cell

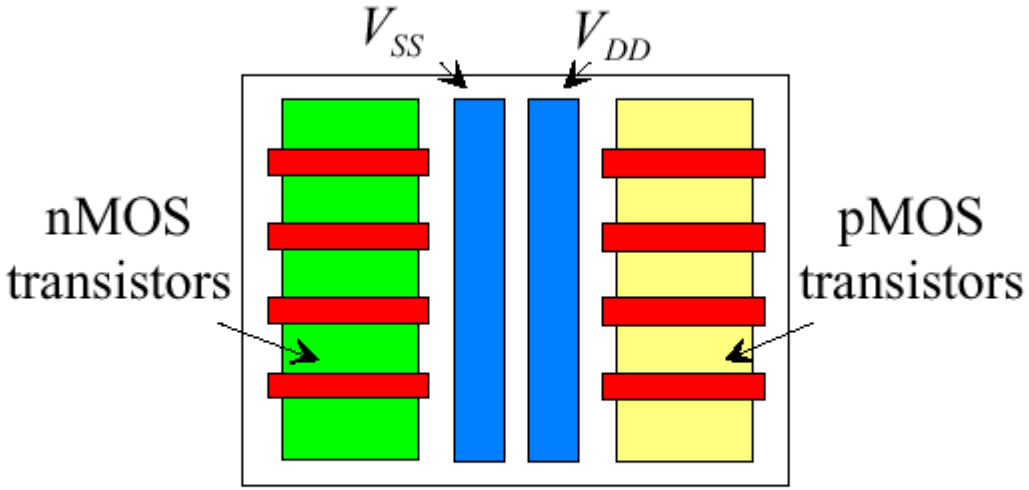



Fig. 1.16 Example of a gate cell in a gate array.

# Gate Array Customization

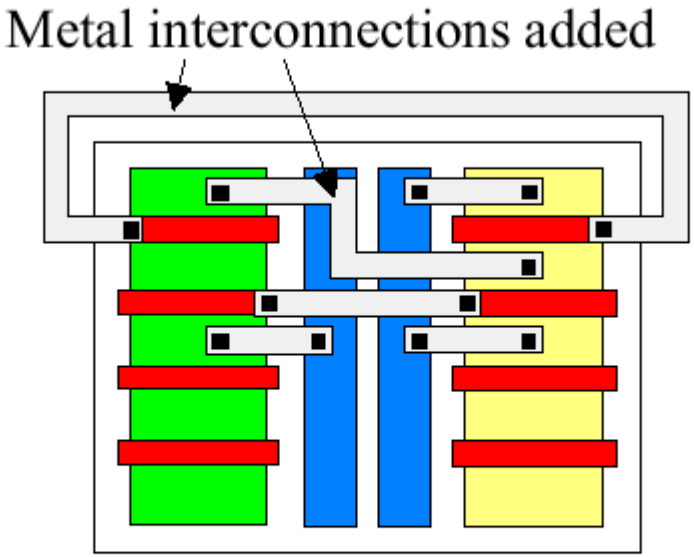



Fig. 1.17 Customization of a gate-cell.

# Field Programmable Gate Array

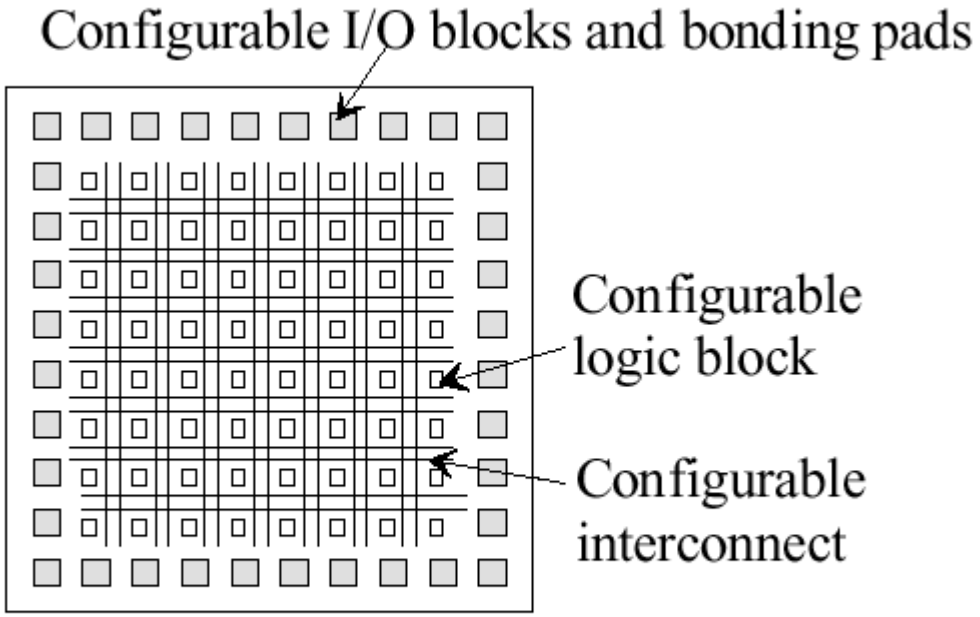



Fig. 1.18 Floorplan of a typical FPGA.

# Pad Frame

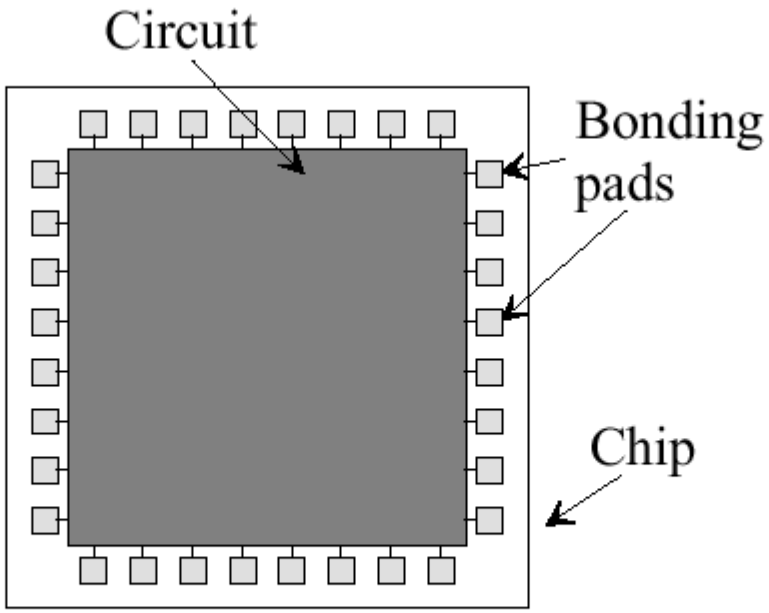



Fig. 1.19 Chip layout including a circuit and a ring of bonding pads (pad frame).

# IC Package

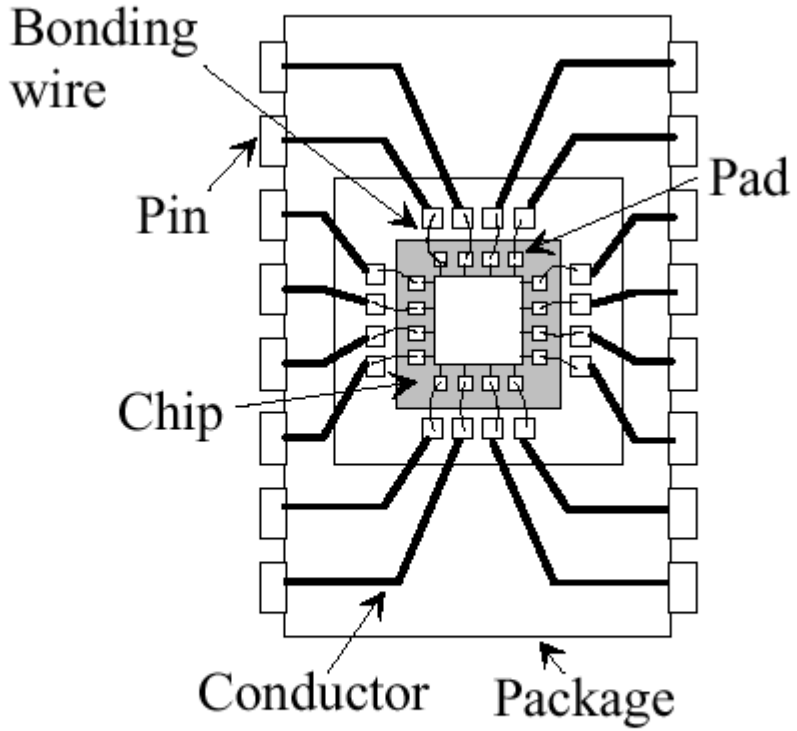



Fig. 1.20 Chip mounted in an IC package.

# Pin Grid Array

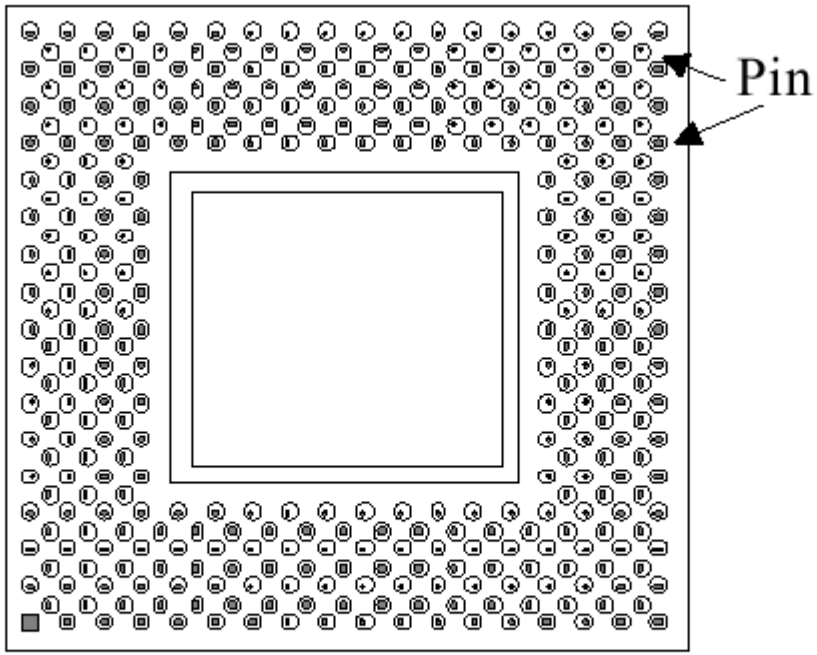



Fig. 1.21 Bottom view of a pin grid array (PGA) package.

# Design Flow

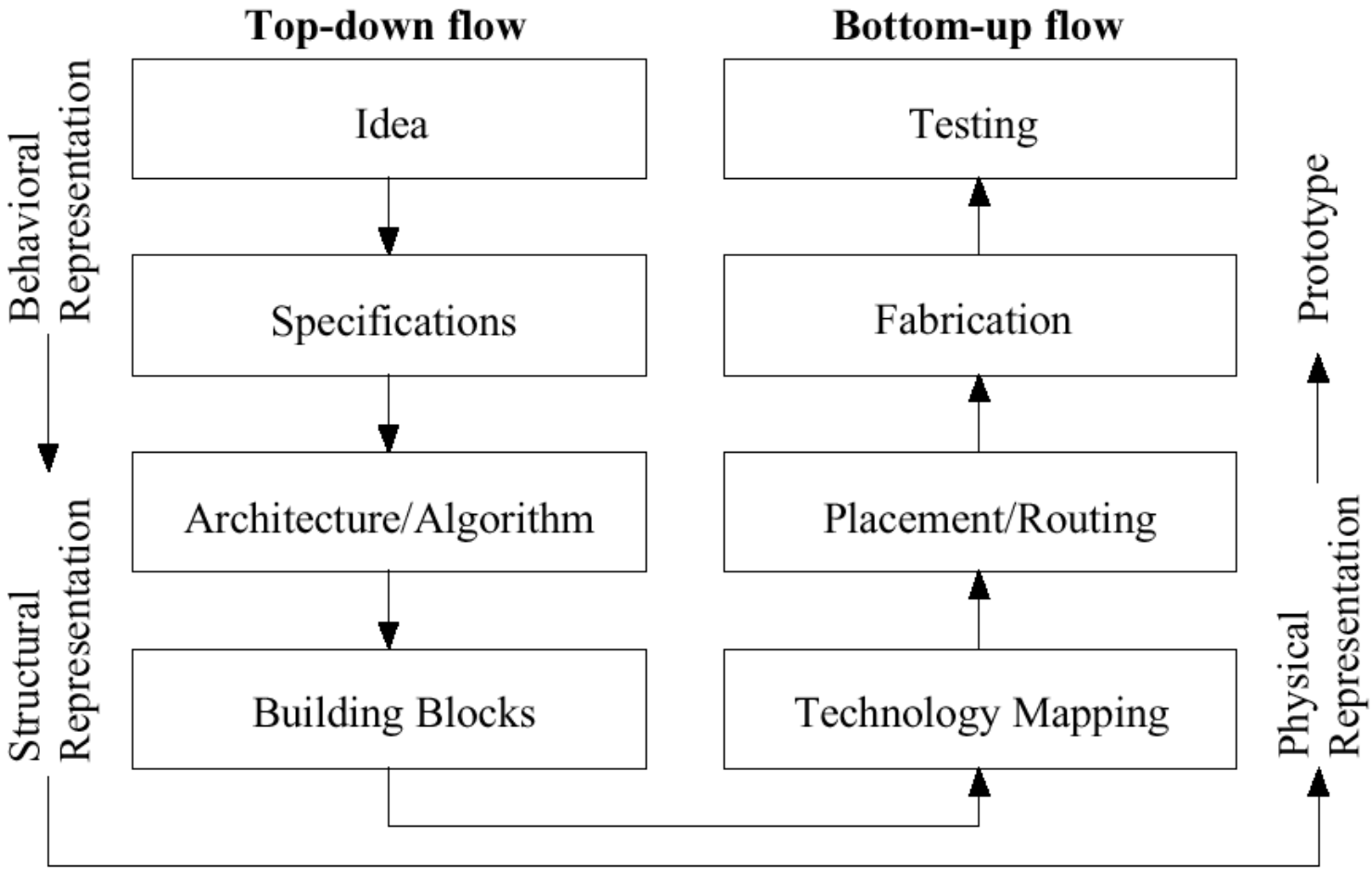



Fig. 1.22 VLSI circuit design flow.