INTEGRATED CIRCUITS

The invention of the transistor in 1947 by William Shockley and his colleagues John Bardeen and Walter Brattain at Bell Laboratories, Murray Hill, NJ, launched a new era of integrated circuits (IC). The transistor concept was based on the discovery that the flow of electric current through a solid semiconductor material like silicon can be controlled by adding impurities appropriately through the implantation processes. The transistor replaced the vacuum tube due to its better reliability, lesser power requirements, and, above all, its much smaller size. In the late 1950s, Jack Kilby of Texas Instruments developed the first integrated circuit. The ability to develop flat or planar ICs, which allowed the interconnection of circuits on a single substrate (due to Robert Noyce and Gordon Moore), began the microelectronics revolution. The substrate is the supporting semiconductor material on which the various devices that form the integrated circuit are attached. Researchers developed sophisticated photolithography techniques that helped in the reduction of the minimum feature size, leading to larger circuits being implemented on a chip. The miniaturization of the transistor led to the development of integrated circuit technology in which several hundreds and thousands of transistors could be integrated on a single silicon die. IC technology led to further developments, such as microprocessors, mainframe computers, and supercomputers.

Since the first integrated circuit was designed following the invention of the transistor, several generations of integrated circuits have come into existence: SSI (small-scale integration) in the early 1960s, MSI (medium-scale integration) in the latter half of the 1960s, and LSI (large-scale integration) in the 1970s. The VLSI (very large scale integration) era began in the 1980s. While the SSI components consisted on the order of 10 to 100 transistors or devices per integrated circuit package, the MSI chips consisted of anywhere from 100 to 1000 devices per chip. The LSI components ranged from roughly 1000 to 20,000 transistors per chip, while the VLSI chips contain on the order of up to 3 million devices. When the chip density increases beyond a few million, the Japanese refer to the technology as ULSI (ultra large scale integration), but many in the rest of the world continue to call it VLSI. The driving factor behind integrated circuit technology was the scaling factor, which in turn affected the circuit density within a single packaged chip. In 1965, Gordon Moore predicted that the density of components per integrated circuit would continue to double at regular intervals. Amazingly, this has proved true, with a fair amount of accuracy (1).

Another important factor used in measuring the advances in IC technology is the minimum feature size or the minimum line width within an integrated circuit (measured in microns). From about 8 μm in the early 1970s, the minimum feature size has decreased steadily, increasing the chip density or the number of devices that can be packed within a given die size. In the early 1990s, the minimum feature size decreased to about 0.5 μm, and currently 0.3, 0.25, and 0.1 μm technologies (also called as deep submicron technologies) are becoming increasingly common. IC complexity refers, in general, to the increase in chip area (die size), the decrease in minimum feature size, and the increase in chip density. With the increase in IC complexity, the design time and the design automation complexity increase significantly. The advances in IC technology are the result of many factors, such as high-resolution lithography techniques, better processing capabilities, reliability and yield characteristics, sophisticated design automation tools, and accumulated architecture, circuit, and layout design experience.

BASIC TECHNOLOGIES

The field of integrated circuits is broad. The various basic technologies commonly known are shown in Fig. 1. The inert substrate processes, further divided as thin and thick film processes, yield devices with good resistive and temperature characteristics. However, they are mostly used in low-volume circuits and in hybrid ICs. The two most popular active substrate materials are silicon and gallium arsenide (GaAs). The silicon processes can be separated into two classes: MOS (the basic device is a metal oxide semiconductor field effect transistor) and bipolar (the basic device is bipolar junction transistors). The bipolar process was commonly used in the 1960s and 1970s and yields high-speed circuits with the overhead of high-power dissipation and the disadvantage of low density. The transistor-transistor logic (TTL) family of circuits constitutes the most popular type of bipolar and is still used in many high-volume applications. The emitter-coupled logic

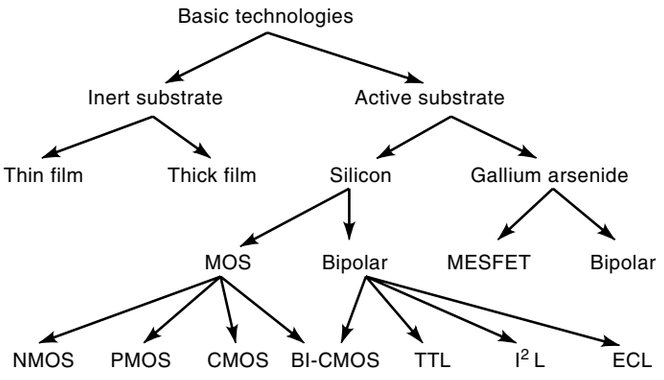

Figure 1. Overview of the basic technologies.

(ECL) devices are used for high-speed parts that form the critical path delay of the circuit. The MOS family of processes consists of PMOS, NMOS, CMOS, and BiCMOS. The term *PMOS* refers to a MOS process that uses only *p*-channel transistors, and *NMOS* refers to a MOS process that uses only *n*-channel transistors. PMOS is not used much due to its electrical characteristics, which are not as good as the *n*-channel field effect transistors (FET), primarily since the mobility of the *n*-channel material is almost twice compared to the mobility of the *p*-channel material. Also, the NMOS devices are smaller than the PMOS devices, and thus PMOS do not give good packing density.

CMOS was introduced in the early 1960s; however, it was only used in limited applications, such as watches and calculators. This was primarily due to the fact that CMOS had slower speed, less packing density, and latchup problems although it had a high noise margin and lower power requirements. Thus, NMOS was preferred over CMOS, in general, until the *p*-channel devices developed began to have similar characteristics as the *n*MOS and both the *p*-channel and *n*-channel transistors started delivering close to equal amounts of currents with similar transistor sizes. In the 1980s and the 1990s, the need for lower power consumption was the driving factor, and thus CMOS emerged as the leading IC technology (2). The BiCMOS technology combines both bipolar and CMOS devices in a single process. While CMOS is preferred for logic circuits, BiCMOS is preferred for input/output (I/O) and driver circuits due to its low input impedance and high current driving capability.

Since the 1980s, efforts have been directed toward designing digital ICs using GaAs devices. In many high-resolution radar systems, space systems, high-speed communication circuits, and microwave circuits, the integrated circuits need to operate at speeds beyond several gigahertz (GHz). In silicon technology, it is possible to obtain speeds on the order of up to 10 GHz using ECL circuits, which is almost pushing the limits of the silicon technology. In GaAs technology, the basic device is the metal semiconductor (Schottky gate) field effect transistor, called the GaAs MESFET. Given similar conditions, the electrons in *n*-type GaAs material travel twice faster than in silicon. Thus, the GaAs circuits could function at twice the speed than the silicon ECL circuits for the same minimum feature size. The GaAs mateial has a larger bandgap and does not need gate oxide material, as in silicon, which makes it immune to radiation effects. Also, the GaAs material has very high resistivity at room temperatures and lower parasitic capacitances, yielding high-quality transistor devices. However, the cost of fabricating large GaAs circuits is significantly high due to its low reliability and yield characteristics (primarily due to the presence of more defects in the material compared to silicon). The fabrication process is complex, expensive, and does not aid scaling. Also, the hole mobility is the same as in silicon, which means GaAs is not preferable for complementary circuits. Thus, the GaAs technology has not been as successful as initially promised. Since CMOS has been the most dominant technology for integrated circuits, we examine the MOS transistor and its characteristics as a switch in the next section.

THE MOS SWITCH

The MOSFET is the basic building block of contemporary CMOS circuits, such as microprocessors and memories. A MOSFET is an unipolar device; that is, current is transported by means of only one type of polarity (electrons in an *n* type and holes in a *p* type). In this section, we describe the basic structure of MOSFETS and their operation and provide examples of gates built using MOS devices.

Structure

The basic structure of a MOSFET (*n* and *p* type) is shown in Fig. 2. We describe the structure of an *n*-type MOSFET (3,4). It consists of four terminals with a *p*-type substrate into which two n^+ regions are implanted. The substrate is a silicon wafer that provides stability and support. The region between the two n^+ regions is covered by an insulator, typically polysilicon and a metal contact. This contact forms the gate of the

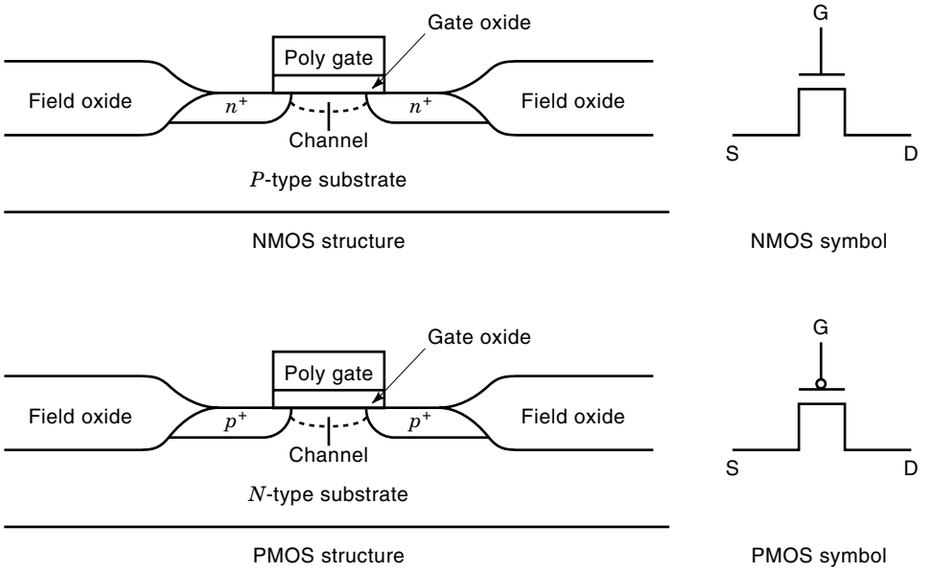

Figure 2. Structure of *n*- and *p*-type MOSFET.

transistor. The insulating layer is required to prevent the flow of current between the semiconductor and the gate. The two n^+ regions form the source and the drain. Due to the symmetry of the structure, the source and the drain are equivalent. The gate input controls the operation of the MOSFET. A bias voltage on the gate causes the formation of a channel between the n^+ regions. This channel causes a connection between the source and drain and is responsible for the flow of the current. The MOSFET is surrounded by a thick oxide, called the field oxide, which isolates it from neighboring devices. Reversal of n and p types in the discussion will result in a p-type MOSFET. Typical circuit symbols for n-type and p-type MOSFETS are also shown in Fig. 2.

Operation

When no gate bias is applied, the drain and the source behave as two pn junctions connected in series in the opposite direction. The only current that flows is the reverse leakage current from the source to the drain. When a positive voltage is applied to the gate, the electrons are attracted and the holes are repelled. This causes the formation of an inversion layer or a channel region. The source and the drain are connected by a conducting n channel through which the current can flow. This voltage-induced channel is formed only when the applied voltage is greater than the threshold voltage, V_t. MOS devices that do not conduct when no gate bias is applied are called *enhancement mode* or normally OFF transistors. In nMOS enhancement mode devices, a gate voltage greater than V_t should be applied for channel formation. In pMOS enhancement mode devices, a negative gate voltage whose magnitude is greater than V_t must be applied. MOS devices that conduct at zero gate bias are called normally ON or *depletion mode* devices. A gate voltage of appropriate polarity depletes the channel of majority carriers and hence turns it OFF.

Considering an enhancement mode n-channel transistor, when the bias voltage is above the predefined threshold voltage, the gate acts as a closed switch between the source and drain, the terminals of which become electrically connected. When the gate voltage is cut off, the channel becomes absent, the transistor stops conducting, and the source and the drain channels get electrically disconnected. Similarly, the p-channel transistor conducts when the gate voltage is beneath the threshold voltage and stops conducting when the bias voltage is increased above the threshold. The behavior of the MOS transistor as a switch forms the fundamental basis for implementing digital Boolean circuits using MOS devices.

Output Characteristics

We describe the basis output characteristics (5,6) of a MOS device in this subsection. There are three regions of operation for a MOS device:

1. Cutoff region
2. Linear region
3. Saturation region

In the cutoff region, no current flows and the device is said to be off. When a bias, V_{gs}, is applied to the gate such that $V_g > V_t$, the channel is formed. If a small drain voltage, V_{ds}, is applied, drain current, I_{ds}, flows from source to drain through the conducting channel. The channel acts like a resistance, and the drain current is proportional to the drain voltage. This is the linear region of operation. As the value of V_{ds} is increased, the channel charge near the drain decreases. The channel is pinched off when $V_{ds} = V_{gs} - V_t$. An increase in V_{ds} beyond the pinchoff value causes little change in the drain current. This is the saturation region of operation of the MOS device. The output characteristics of n- and p-type devices is shown in Fig. 3. The equations that describe the regions of operation can be summarized as follows:

$$I_{ds} = \begin{cases} 0 & \text{If } V_{gs} \le V_t \quad \text{(cutoff)} \\ k/2[2(V_{gs} - V_t)V_{ds} - V_{ds}^2] & \text{If } V_g > V_t, V_{ds} \le (V_{gs} - V_t) \quad \text{(linear)} \\ k/2(V_{gs} - V_t)^2 & \text{If } V_g > V_t, V_{ds} > (V_{gs} - V_t) \quad \text{(saturation)} \end{cases}$$

where k is the transconductance parameter of the transistor. A detailed analysis of the structure and operation of MOS devices is described in Refs. 3, 5, 7, and 8.

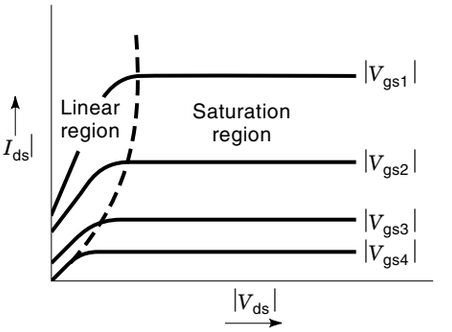

Figure 3. Output characteristics of a MOS transistor.

CMOS Inverter

The basic structure of an inverter is shown in Fig. 4, and the process cross section is shown in Fig. 5. The gates of both the NMOS and the PMOS transistors are connected. The PMOs transistor is connected to the supply voltage V_{dd}, and the NMOS transistor is connected to G_{nd}. When a logical 0 is applied at the input V_{in}, then the PMOS device is on and the output is pulled to V_{dd}. Hence the output is a logical 1. On the other hand, when a logical 1 is applied at the input, then the NMOS transistor is on and the output is pulled to the ground. Hence we have a logical 0. The operating regions of the tran-

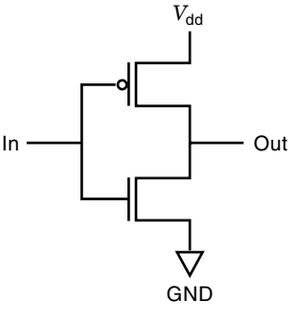

Figure 4. Circuit schematic of an inverter.

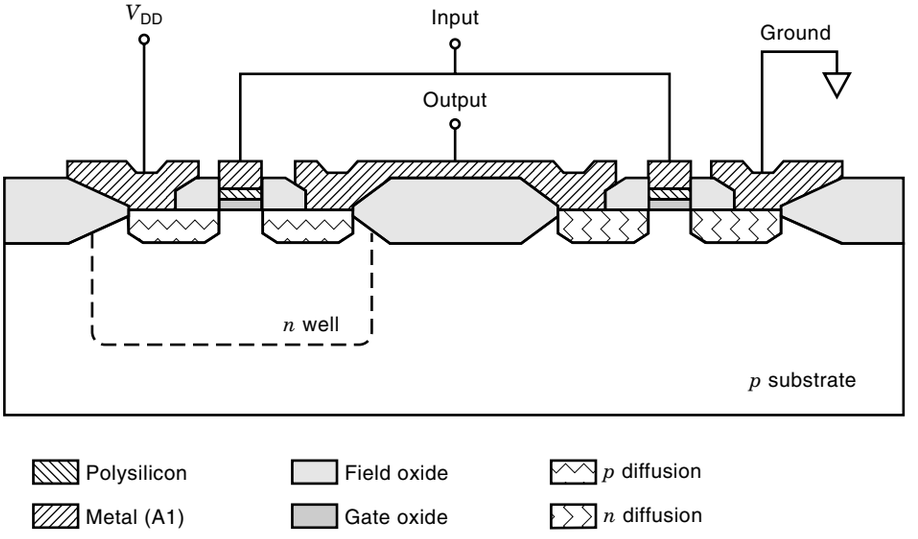

Figure 5. Process cross section of an n-well inverter.

sistor are shown in Fig. 6. In region I, the n device is off and the p device operates in the linear region. Hence the output is pulled to V_{dd}. In region II, the n and p devices operate in the linear and saturation region depending on the input voltage. In region III, the p device is cut off and the n device is operating in the linear region. The output is pulled to the ground. In region II, when both the transistors are on simultaneously, a short is produced between V_{dd} and G_{nd}. This accounts for the short circuit power dissipation in CMOS logic.

Transmission Gate

Consider the device shown in Fig. 7, which represents an NMOS or a PMOS device. By suitably controlling the gate bias, the device can be made to turn on or off. It behaves as an electrical switch that either connects or disconnects the points s and d. An NMOS device is a good switch when it passes a logical 0, and a PMOS is a good switch when it passes a logical 1. In CMOS logic, both the NMOS and PMOS devices operate together. In general, the NMOS transistor pulls down the output node to logical 0, and the PMOS device pulls up a node to logical 1. A transmission gate is obtained by connecting the two in parallel, as shown in Fig. 8. The control signal (say, g) applied to the n-type device is complemented and applied to the p-type device. When g is high, both the transistors are on and hence a good 1 or a 0 is passed. When g is low, both the devices are off. This is also called a complementary switch, or a C SWITCH (5).

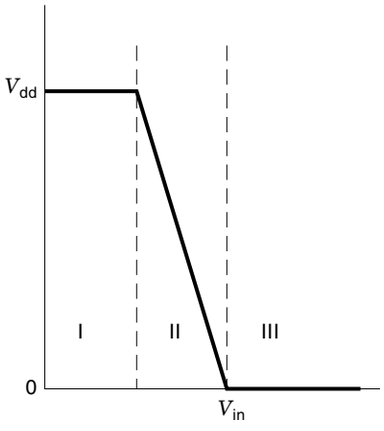

Figure 6. Operating regions of the transistor.

NAND and NOR Gates

CMOS combinational gates are constructed by connecting the PMOS and NMOS devices in either series or parallel to generate different logical functions. The structures for a two-input NAND and NOR gate are shown in Fig. 9.

NAND Gate. The p devices are connected in parallel, and the n devices are connected in series. When either of the inputs A or B is a logical 0, the output is pulled high to V_{dd}. When both A and B are high, then the output is pulled to the ground. Hence this structure implements the operation $f = (A \cdot B)'$.

NOR Gate. Similarly, in the NOR gate, the p devices are connected in series and the n devices are connected in parallel. When either of the inputs A or B is a logical 1, then the output is pulled to the ground. When both A and B are low, then the output is pulled to V_{dd}. Hence this structure implements the operation $f = (A + B)'$. The p structure is the logical dual of the n structure. An n input NAND and NOR gate can be constructed in a similar fashion.

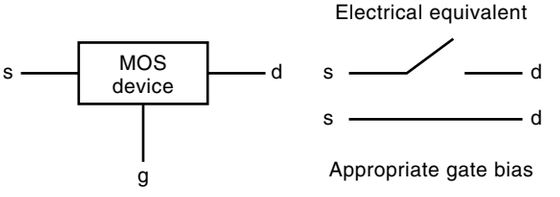

Figure 7. A MOS device as a switch.

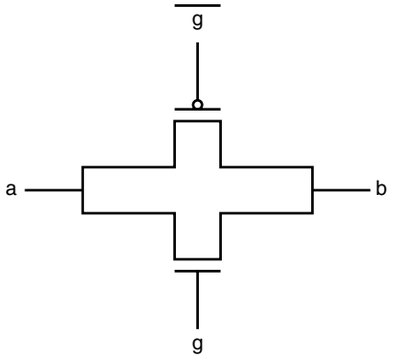

Figure 8. Transmission gate.

IC DESIGN METHODOLOGY

To design and realize VLSI circuits, several factors play key roles. The goal of an IC designer is to design a circuit that meets the given specifications and requirements while spending minimum design and development time avoiding design errors. The designed circuit should function correctly and meet the performance requirements, such as delay, timing, power, and size. A robust design methodology has been established over the years, and the design of complex integrated circuits has been made possible essentially due to advances in VLSI design automation. The various stages in the design flow are shown in Fig. 10. The design cycle ranges from the system-level specification and requirements to the end product of a fabricated, packaged, and tested integrated circuit. The basic design methodology is briefly described here, and the various stages are discussed in detail in the followiing sections using simple examples.

The first step is to determine the system-level specifications, such as the overall functionality, size, power, performance, cost, application environment, IC fabrication process, technology, and chip-level and board-level interfaces required. There are several tradeoffs to be considered. The next step is

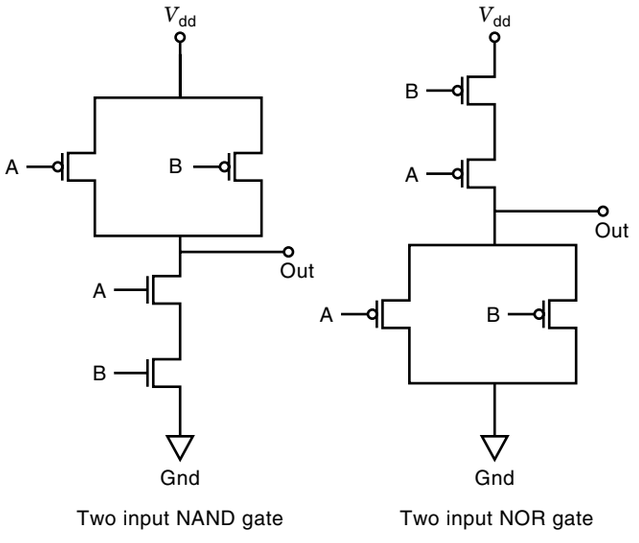

Figure 9. Two-input NAND and NOR gate.

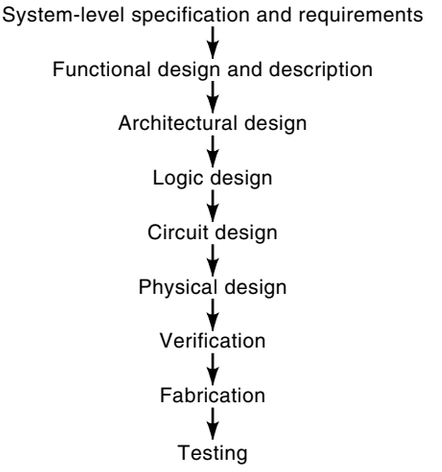

Figure 10. IC design methodology.

the functional design and description, in which the system is partitioned into functional modules and the functionality of the different modules and their interfaces to each other are considered. The issues to be considered are regularity and modularity of structures, subsystem design, data flow organization, hierarchical design approach, cell types, geometric placements, and communication between the different blocks.

Once the functionality of the various modules is determined, the architectural design of the modules is pursued. Many design alternatives are considered toward optimization. This stage also includes the design of any hardware algorithms to be mapped onto architectures. A behavioral-level description of the architecture is obtained and verified using extensive simulations, often with an iterative process. This stage is critical in obtaining an efficient circuit in the end and for simplifying the steps in some of the following stages. In the logic design stage, the architectural blocks are converted into corresponding gate-level logic designs, Boolean minimization is performed, and logic simulation is used to verify the design at this level. In some design flows, the circuit could be synthesized from the logic level by using gate-level libraries (this is referred to as logic synthesis). The logic design usually includes a conventioinal logic design approach and a nontraditional design, such as precharge logic. At this stage, gate delays are considered and timing diagrams are derived to verify the synchronization of the various logic modules. The next step is the circuit design stage, which essentially involves converting the logic design modules into a circuit representation. At this stage, the essential factors considered are clocking, switching speeds or delays, switching activity and power requirements, and other electrical characteristics (e.g., resistance, capacitance).

The most complex step in VLSI design automation is the physical design, which includes floorplanning, partitioning, placement, routing, layout, and compaction. This process converts the given circuit design or description into a physical layout that is a geometric representation of the entire circuit. Each step of the physical design by itself is complex and takes significant amounts of iterations and time. The various types of transistors, the interconnecting wires, and contacts between different wires and transistors are represented as different geometric patterns consisting of many layers placed ac-

cording to several design rules that govern a given fabrication technology and process. The floorplanning step involves higher-level planning of the various components on the layout. The partitioning step converts the overall circuit into smaller blocks to help the other steps. It is usually impractical to synthesize the entire circuit in one step. Thus, logic partitioning is used to divide the given circuit into a smaler number of blocks, which can be individually synthesized and compacted. This step considers the size of the blocks, the number of blocks, and the interconnections between the blocks and yields a netlist for each block that can be used in the further design steps.

During the next step, which is the placement of the blocks on the chip layout, the various blocks are placed such that the routing can be completed effectively and the blocks use minimum overall area, avoiding any white spaces. The placement task is iterative in that an initial placement is obtained first and evaluated for area minimization and effective routing possibility, and alternate arrangements are investigated until a good placement is obtained. The routing task completes the routing of the various interconnections, as specified by the netlists of the different blocks. The goal is to minimize the routing wire lengths and minimize the overall area needed for routing. The routing areas between the various blocks are referred to as channels or switchboxes. Initially, a global routing is performed in which a channel assignment is determined based on the routing requirements, and then a detailed routing step completes the actual point-to-point routing.

The last step in the physical design is the compaction step, which tries to compact the layout in all directions to minimize the layout area. A compact layout leads to less wire lengths, lower capacitances, and more chip density since the chip area is used effectively. The compaction step is usually an interactive and iterative process in which the user can specify certain parameters and check if the compaction can be achieved. The goal of compaction, in general, is to achieve minimum layout area. The entire physical design process is iterative and is performed several times until an efficient layout for the given circuit is obtained.

Once the layout is obtained, design verification needs to be done to ensure that the layout produced functions correctly and meets the specifications and requirements. In this stage, design rule checking is performed on the layout to make sure that the geometric placement and routing rules and the rules regarding the separation of the various layers, the dimensions of the transistors, and the width of the wires are followed correctly. Any design rule violations that occurred during the physical design steps are detected and removed. Then circuit extraction is performed to complete the functional verification of the layout. This step verifies the correctness of the layout produced by the physical design process. After layout verification, the circuit layout is ready to be submitted for fabrication, packaging, and testing. Usually, several dies are produced on a single wafer and the wafer is tested for faulty dies. The correct ones are diced out and packaged in the form of a pin grid array (PGA), dual in-line package (DIP), or any other packaging technology. The packaged chip is tested extensively for functionality, electrical and thermal characteristics, and performance. The process of designing and building an integrated circuit (9) that meets the performance requirements and functions perfectly depends on the efficiency of the design automation tools.

CIRCUIT DESIGN

To create performance optimized designs, two areas have to be addressed to achieve a prescribed behavior: (1) circuit or structural design, and (2) layout or physical design. While the layout design is discussed in a later section, this section focuses on the former.

A logic circuit must function correctly and meet the timing requirements. There are several factors that can result in the incorrect functioning of a CMOS logic gate: (1) incorrect or insufficient power supplies, (2) noise on gate inputs, (3) faulty transistors, (4) faulty connections to transistors, (5) incorrect ratios in ratioed logic, and (6) charge sharing or incorrect clocking in dynamic gates. In any design, there are a certain paths, called *critical paths,* that require attention to timing details since they determine the overall functional frequency. The critical paths are recognized and analyzed using timing analyzer tools and can be dealt with at four levels:

1. Architecture
2. RTL/logic level
3. Circuit level
4. Layout level

Designing an efficient overall functional architecture helps to achieve good performance. To design an efficient architecture, it is important to understand the characteristics of the algorithm being implemented as the architecture. At the register transfer logic (RTL)/logic level, pipelining, the type of gates, and the fan-in and the fan-out of the gates are to be considered. Fan-in is the number of inputs to a logic gate, and fan-out is the number of gate inputs that the output of a logic gate drives. Logic synthesis tools can be used to achieve the transformation of the RTL level. From the logic level, the circuit level can be designed to optimize a critical speed path. This is achieved by using different styles of CMOS logic, as explained later in this section. Finally, the speed of a set of logic can be affected by rearranging the physical layout. The following techniques can be used for specific design constraints.

The various CMOS logic structures that can be used to implement circuit designs are as follows:

1. *CMOS Complementary Logic.* The CMOS complementary logic gates are designed as ratioless circuits. In these circuits, the output voltage is not a fraction of the V_{dd} (supply) and the gates are sized to meet the required electrical characteristics of the circuits. The gate consists of two blocks, and n block and a p block, that determine the function of the gate. The p block is a dual of the n block. Thus, an n-input gate will consist of $2n$ transistors.
2. *Pseudo-NMOS Logic.* In this logic, the load device is a single p transistor with the gate connected to V_{dd} (5,10). This is equivalent to replacing the depletion NMOS load in a conventional NMOS gate by a p device. The design of this style of gate (11,12) involves ratioed transistor sizes to ensure proper operation and is shown in Fig.

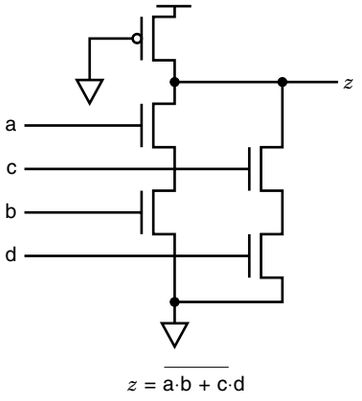

Figure 11. Pseudo-NMOS logic.

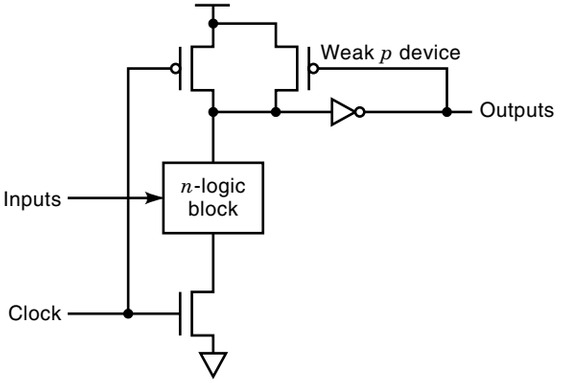

Figure 13. CMOS-domino logic.

11. The static power dissipation that occurs whenever the pull-down chain is turned on is a major drawback of this logic style.

3. *Dynamic CMOS Logic.* In the dynamic CMOS logic style, an n-transistor logic structure's output node is precharged to V_{dd} by a p transistor and conditionally discharged by an n transistor connected to V_{ss} (5). The input capacitance of the gate is the same as the pseudo-NMOS gate. Here, the pull-up time is improved by virtue of the active switch, but the pull-down time is increased due to the ground. The disadvantage of this logic structure is that the inputs can only change during the precharge phase and must be stable during the evaluate portion of the cycle Figure 12 depicts this logic style.
4. *Clocked CMOS Logic.* To build CMOS logic gates with low power dissipation (13), this logic structure was proposed. The reduced dynamic power dissipation is realized due to the metal gate CMOS layout considerations. The gates have larger rise and fall times because of the series clocking transistors, but the capacitance is similar to the CMOS complementary gates. This is a recommended strategy for "hot electron" effects, because it places an additional n transistor in series with the logic transistors (14).
5. *CMOS Domino Logic.* This is a modification of the clocked CMOS logic, in which a single clock is used to precharge and evaluate a cascaded set of dynamic logic blocks. This involves incorporating a static CMOS inverter into each logic gate (15), as shown in Fig. 13. During precharge, the output node is charged high and hence the output of the buffer is low. The transistors in the subsequent logic blocks will be turned off since they are fed by the buffer. When the gate is evaluated, the output will conditionally go low (1–0), causing the buffer to conditionally go high (0–1). Hence, in a cascaded set of logic blocks, each state evaluates and causes the next stage to evaluate, provided the entire sequence can be evaluated in one clock cycle. Therefore, a single clock is used to precharge and evaluate all logic gates in a block. The disadvantages of this logic are that (1) every gate needs to be buffered, and (2) only noninverting structures can be used.
6. *NP Domino Logic (Zipper CMOS).* This is a further refinement of the domino CMOS (16–18). Here, the domino buffer is removed, and the cascading of logic blocks is achieved by alternately composed p and n blocks, as is shown in Fig. 14. When the clock is low, all the n-block stages are precharged high while all the p-block stages are precharged low. Some of the advantages of

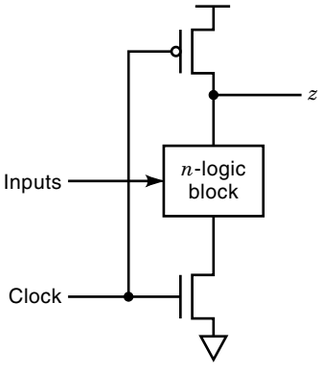

Figure 12. Dynamic CMOS logic.

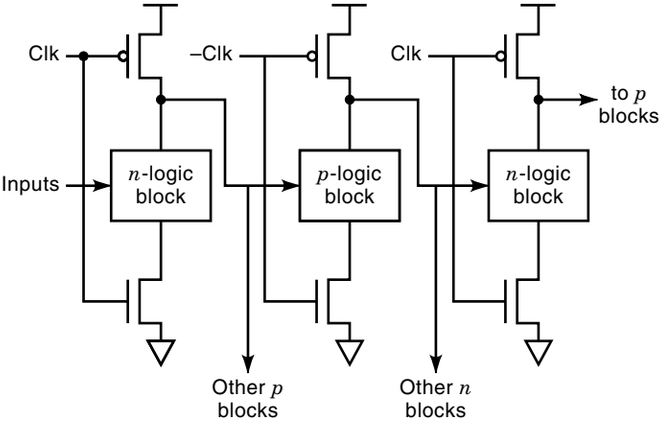

Figure 14. NP-domino logic.

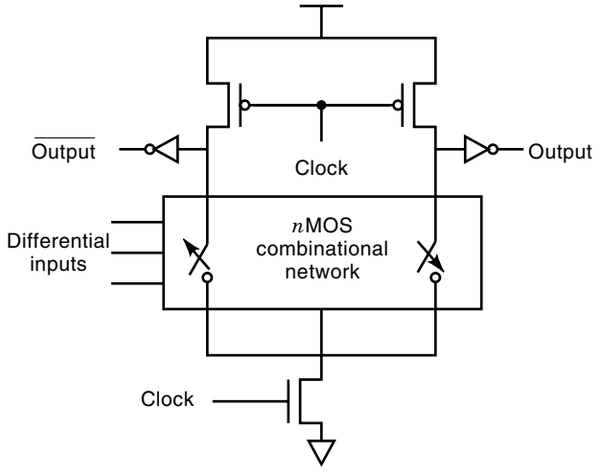

Figure 15. Cascade voltage switch logic.

the dynamic logic styles include (1) smaller area than fully static gates, (2) smaller parasitic capacitances, and (3) glitch-free operation if designed carefully.

7. *Cascade Voltage Switch Logic (CVSL).* The CVSL is a differential style of logic requiring both true and complement signals to be routed to gates (19). Two complementary NMOS structures are constructed and then connected to a pair of cross-coupled p pull-up transistors. The gates here function similarly to the domino logic, but the advantage of this style is the ability to generate any logic expression involving both inverting and noninverting structures. Figure 15 gives a sketch of the CVSL logic style.
8. *Pass Transistor Logic.* Pass transistor logic is popular in NMOS-rich circuits. Formal methods for deriving pass transistor logic for NMOS are presented in Ref. 20. Here, a set of control signals and a set of pass signals are applied to the gates and sources of the n transistor, correspondingly. From these signals, the truth table for any logic equation can be realized.
9. Source follower pull-up logic (SFPL): This is similar to the pseudo-NMOS gate except that the pull-up is controlled by the inputs (21). In turn, this leads to the use of smaller pull-down circuits. The SFPL gate style reduces the self-loading of the output and improves the speed of the gate. Therefore, it shows a marked advantage in high fan-in gates.

Using the various design styles, any circuit design can be built in a hierarchical fashion. The basic gates are first built, from which functional blocks like a multiplexer or an adder circuit can be realized. From these basic blocks, more complex circuits can be constructed. Once a design for a specific application has been designed, the functionality of the circuit needs to be verified. Also, other constraints, like the timing and electrical characteristics, have to be studied before the design can be manufactured. The techniques and tools to achieve this are the subject of the next section.

SIMULATION

Simulation is required to verify if a design works the way it should. Simulation can be performed at various levels of abstraction. A circuit can be simulated at the logic level, the switch level, or with reference to the timing. Simulation is a critical procedure before committing the design to silicon. The simulators themselves are available in a wide variety of types (22).

Logic-Level Simulation

Logic-level simulation occurs at the highest level of abstraction. It uses primitive models of NOT, OR, AND, NOR, and NAMD gates. Virtually all digital logic simulators are event driven (i.e., a component is evaluated based on when an event occurs on its inputs). Logic simulators are categorized according to the way the delays are modeled in the circuit: (1) unit delay simulators, in which each component is assumed to have a delay of one time unit, and (2) variable-delay simulators, which allow components to have arbitrary delays. While the former helps in simulating the functionality of the circuit the latter allows for more accurate modeling of the fast-changing nodes.

The timing is normally specified in terms of an inertial delay and a load-dependent delay, as follows:

$$T_{\text{gate}} = T_{\text{intrinsic}} + C_{\text{load}} \times T_{\text{load}}$$

where

T_{gate} = delay of the gate
$T_{\text{intrinsic}}$ = intrinsic gate delay
C_{load} = actual load in some units (pF)
T_{load} = delay per load in some units (ns/pF)

Earlier, logic simulators used preprogrammed models for the gates, which forced the user to describe the system in terms of these models. In modern simulators, programming primitives are provided that allow the user to write models for the components. The two most popular digital simulation systems in use today are VHDL and Verilog.

Circuit-Level Simulation

The most detailed and accurate simulation technique is referred to as *circuit analysis.* Circuit analysis simulators operate at the circuit level. Simulation programs typically solve a complex set of matrix equations that relate the circuit voltages, currents, and resistances. They provide accurate results but require long simulation times. If N is the number of nonlinear devices in the circuit, then the simulation time is typically proportional to N^m, where m is between 1 and 2. Simulation programs are useful in verifying small circuits in detail but are unrealistic for verifying complex VLSI designs. They are based on transistor models and hence should not be assumed to predict accurately the performance of designs. The basic sources of error include (1) inaccuracies in the MOS model parameters, (2) an inappropriate MOS model, and (3) inaccuracies in parasitic capacitances and resistances. The circuit analysis programs widely used are the SPICE program, developed at the University of California at Berekely (23), and the ASTAP program from IBM (24). HSPICE (25) is the commercial variant of these programs. The SPICE program provides various levels of modeling. The simple models are optimized for speed, while the complex ones are used to get accurate solutions. As the feature size of the processes is reduced, the models used for the transistors are no longer

valid and hence the simulators cannot predict the performance accurately unless new models are used.

Switch-Level Simulation

Switch-level simulation is simulation performed at the lowest level of abstraction. These simulators model transistors as switches to merge the logic-level and circuit-level simulation techniques. Although logic-level simulators also model transistors as switches, the unidirectional logic gate model cannot model charge sharing, which is a result of the bidirectionality of the MOS transistor. Hence, we assume that all wires have capacitance, since we need to locate charge-sharing bugs. RSIM (26) is an example of a switch-level simulator with timing. In RSIM, CMOS gates are modeled as either pull-up or pull-down structures, for which the program calculates a resistance to power or ground. The output capacitance of the gate is used with the resistance to predict the rise and the fall times of a gate.

Timing Simulators

Timing simulators allow simple nonmatrix calculations to be employed to solve for circuit behavior. This involves making approximations about the circuit, and hence accuracy is less than that of simulators like SPICE. The advantage is the execution time, which is over two orders of magnitude less than for SPICE. Timing simulator implementations typically use MOS-model equations or table look-up methods. Examples of these simulators are in Ref. 27.

Mixed-Mode Simulators

Mixed-mode simulators are available commercially today and merge the aforementioned different simulation techniques. Each circuit block can be simulated in the appropriate mode.

The results of the simulation analysis are fed back to the design stage, where the design is tuned to incorporate the deviations. Once the circuit is perfected and the simulation results are satisfactory, the design can be fabricated. To do this, we need to generate a geometric layout of the transistors and the electrical connections between them. This has been a subject of intense research over the last decade and continues to be so. The following section introduces this problem and presents some of the well-known techniques for solving it.

LAYOUT

The layout design is considered a prescription for preparing the photomasks used in the fabrication of ICs (5). There is a set of rules, called the design rules, used for the layout; these serve as the link between the circuit and the process engineer. The physical design engineer, in addition to knowledge of the components and the rules of the layout, needs strategies to fit the layouts together with other circuits and provide good electrical properties. The main objective is to obtain circuits with optimum yield in as small an area as possible without sacrificing reliability.

The starting point for the layout is a circuit schematic. Figure 2 depicts the schematic symbols for an n-type and p-type transistor. The circuit schematic is treated as a specification for which we must implement the transistors and connections between them in the layout. The circuit schematic of an inverter is shown in Fig. 4. We need to generate the exact layout of the transistors of this schematic, which can then be used to build the photomask for the fabrication of the inverter. Generating a complete layout in terms of rectangles for a complex system can be overwhelming, although at some point we need to generate it. Hence designers use an abstraction between the traditional transistor schematic and the full layout to help organize the layout for complex systems. This abstraction is called a *stick diagram.* Figure 16 shows the stick diagram for the inverter schematic. As can be seen, the stick diagram represents the rectangles in the layout as lines, which represent wires and component symbols. Stick diagrams are not exact models of the layouts but let us evaluate a candidate design with relatively little effort. Area and aspect ratio are difficult to estimate from stick diagrams.

Design Rules

Design rules for a layout (28) specify to the designer certain geometric constraints on the layout artwork so that the patterns on the processed wafer will preserve the topology and geometry of the designs. These help to prevent separate, isolated features from accidentally short circuiting, or thin features from opening, or contacts from slipping outside the area to be contacted. They represent a tolerance that ensures very high probability of correct fabrication and subsequent operation of the IC. The design rules address two issues primarily:

1. The geometrical reproduction of features that can be reproduced by the mask-making and lithographical process
2. The interaction among the different layers

Several approaches can be used to descibe the design rules. These include the micron rules, stated at some micron resolution, and the lambda (λ)-based rules. The former are given as a list of minimum feature sizes and spacings for all masks in a process, which is the usual style for the industry. Mead-Conway (29) popularized the λ-based approach, where λ is process dependent and is defined as the maximum distance by which a geometrical feature on any one layer can stray from another feature. The advantage of the λ-based approach is that by defining λ properly the design itself can be made independent of both the process and fabrication house, and the design can be rescaled. The goal is to devise rules that are simple, constant in time, applicable to many processes, standardized among many institutions, and have a small number of exceptions for specific processes. Figure 17 gives the layout of the inverter, with the design rules specified.

To design and verify layouts, different computer-aided design (CAD) tools can be used. The most important of these are the layout editors (30), design rule checkers, and circuit extractors. The editor is an interactive graphic program that allows us to create and delete layout elements. Most editors work on hierarchical layouts, but some editors, like Berekely's

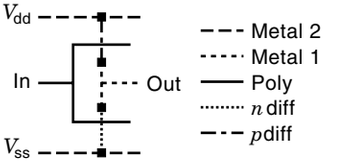

Figure 16. Stick diagram of the inverter.

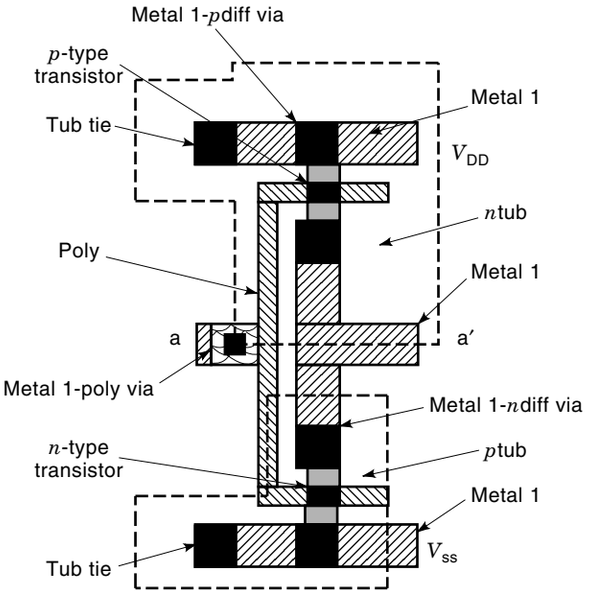

Figure 17. Transistor layout of the inverter.

Magic tool (31), work on a symbolic layout. The latter include somewhat more detail than the stick diagram but are still more abstract than the pure layout. The design rule checker, or DRC, programs look for design rule violations in the layouts. Magic has an on-line design rule checking. The circuit extractor is an extension of the DRC programs. While the DRC must identify transistors and vias to ensure proper checks, the circuit extractor performs a complete job of component and wire extraction. it produces a netlist, which lists the transistors in the layouts and the electrical nets that connect their terminals.

Physical Design

From the circuit design of a certain application and the design rules of a specific process, the physical design problem is to generate a geometric layout of the transistors of the circuit design conforming to the specified design rules. From this layout, photomasks can be generated that will be used in the fabrication process. To achieve this, the different modules of the design need to be placed first and then electrical connections between them realized through the metal layers. For instance, a two-layer metallization would allow the designer to lay out metal both vertically and horizontally on the floorplan. Whenever the wire changes direction, a via can be used to connect the two metal layers. Due to the complexity of this problem, most authors treat module placement and the routing between modules as two separate problems, although they are related critically. Also, in former days, when designs were less complex, design was done by hand. Now we require sophisticated tools for this process.

Placement. Placement is the task of placing modules adjacent to each other to minimize area or cycle time. The literature consists of a number of different placement algorithms that have been proposed (32–35). Most algorithms partition the problem into smaller parts and then combine them, or start with a random placement solution and then refine it to reach the optimal. The modules are usually considered as rectangular boxes with specified dimensions. The algorithms then use different approaches to fit these boxes in a minimal area or to optimize them to certain other constraints. For instance, consider a certain number of modules with specific dimensions and a given area in which to fit them. This is similar to the bin-packing algorithm. After the placement step, the different modules are placed in an optimal fashion and the electrical connections between them need to be realized.

Routing. Once the modules have been placed, we need to create space for the electrical connections between them. To keep the area of the floorplan minimal, the first consideration is to determine the shortest path between nodes, although a cost-based approach may also be used. The cost is defined to include an estimate of the congestion, number of available wire tracks in a local area, individual or overall wire length, and so on. Since the problem is a complex one, the strategy is to split the problem into global or loose routing and local or detailed routing. Global routing is a preliminary step, in which each net is assigned to a particular routing area, and the goal is to make 100% assignment of nets to routing regions while minimizing the total wire length. Detailed routing then determines the exact route for each net within the global route. There are a number of approaches to both of these problems.

Global Routing. Global routing (36) is performed using a wire-length criterion, because all timing critical nets must be routed with minimum wire length. The routing area itself can be divided into disjoint rectangular areas, which can be classified by their topology. A two-sided channel is a rectangular routing area with no obstruction inside and with pins on two parallel sides. A switch box is a rectangular routing area with no obstructions and signals entering and leaving through all four sides (37). The focus in this problem is only to create space between the modules for all the nets and not to determine the exact route. The algorithms proceed by routing one net at a time, choosing the shortest possible path. Since there is a lot of dependency among the nets, different heuristics are used to generate the least possible routing space in which to route the nets. Once space is created for all the nets, the exact route of each net can be determined.

Detailed Routing. Detailed routing is usually done by either a maze-search or a line-search algorithm. The maze-running algorithm (38,39) proposed by Lee-Moore finds the shortest path between any two points. For this, the layout is divided into a grid of nodes, in which each node is weighted by its distance from the source of the wire to be routed. The route that requires the smallest number of squares is then chosen. If a solution exists, this algorithm will find it, but an excessive amount of memory is required to achieve this. In the line-search algorithm, vertical and horizontal lines are drawn from the source and the target, followed by horizontal or vertical lines that intersect the original lines. This is repeated until the source and target meet. There are also a number of other heuristic algorithms that exploit different characteristics of the design to generate optimal routing solutions. Genetic algorithms and simulated annealing approaches to this problem have gained importance in recent years.

An introduction to the various algorithms that have been proposed for layouts can be found in Ref. 40. Once the layout

has been determined and the photomasks made, the circuit can go to the fabrication plant for processing.

FABRICATION

The section describes the approach used in building integrated circuits on monolithic pieces of silicon. The process involves the fabrication of successive layers of insulating, conducting, and semiconducting materials, which have to be patterned to perform specific functions. The fabrication therefore must be executed in a specific sequence, which constitutes an IC process flow. The manufacturing process itself is a complex interrelationship of chemistry, physics, material science, thermodynamics, and statistics.

Semiconducting materials, as the name suggests, are neither good conductors nor good insulators. While there are many semiconducting elements, silicon is primarily chosen for manufacturing ICs because it exhibits few useful properties. Silicon devices can be built with a maximum operating temperature of about 150°C due to the smaller leakage currents as a result of the large bandgap of silicon (1.1 eV). IC planar processing requires the capability to fabricate a passivation layer on the semiconductor surface. The oxide of silicon, SiO_2, which could act as such a layer, is easy to form and is chemically stable. The controlled addition of specific impurities makes it possible to alter the characteristics of silicon. For these reasons, silicon is almost exclusively used for fabricating microelectronic components.

Silicon Material Technology

Beach sand is first refined to obtain semiconductor-grade silicon. This is then reduced to obtain electronic-grade polysilicon in the form of long, slender rods. Single-crystal silicon is grown from this by the Czochralski (CZ) or float-zone (FZ) methods. In CZ growth, single crystal ingots are pulled from molten silicon contained in a crucible. For VLSI applications, CZ silicon is preferred because it can better withstand thermal stresses (41) and offers an internal gettering mechanism than can remove unwanted impurities from the device structures on wafer surfaces (42). FZ crystals are grown without any contact to a container or crucible and hence can attain higher purity and resistivity than CZ silicon. Most high-voltage, high-power devices are fabricated on FZ silicon. The single crystal ingot is then subjected to a complex sequence of shaping and polishing, known as wafering, to produce starting material suitable for fabricating semiconductor devices. This involves the following steps:

1. The single crystal ingot undergoes routine evaluation of resistivity, impurity content, crystal perfection size, and weight.
2. Since ingots are not perfectly round, they are shaped to the desired form and dimension.
3. The ingots are then sawed to produce silicon slices. The operation defines the surface orientation, thickness, taper, and bow of the slice.
4. To bring all the slices to within the specified thickness tolerance, lapping and grinding steps are employed.
5. The edges of the slices are then rounded to reduce substantially the incidence of chipping during normal wafer handling.
6. A chemical-mechanical polishing (43) step is then used to produce the highly reflective and scratch- and damage-free surface on one side of the wafer.
7. Most VLSI process technologies also require an epitaxial layer, which is grown by a chemical vapor deposition process.

The most obvious trend in silicon material technology is the increasing size of the silicon wafers. The use of these larger-diameter wafers presents major challengers to semiconductor manufacturers. Several procedures have been investigated to increase axial impurity uniformity. These include the use of double crucibles, continuous liquid feed (CLF) systems (44), magnetic Czochralski growth (MCZ) (44,45), and controlled evaporation from the melt.

Epitaxial Layer

The epitaxial growth process is a means of depositing a single crystal film with the same crystal orientation as the underlying substrate. This can be achieved from the vapor phase, liquid phase, or solid phase. Vapor phase epitaxy has the widest acceptance in silicon processing, since excellent control of the impurity concentration and crystalline perfection can be achieved. Epitaxial processes are used for the fabrication of advanced CMOS VLSI circuits, because epitaxial processes minimize latch-up effects. Also in the epitaxial layer, doping concentration can be accurately controlled, and the layer can be made oxygen and carbon free. Epitaxial deposition is a chemical vapor deposition process (46). The four major chemical sources of silicon used commercially for this deposition are (1) silicon tetrachloride ($SiCl_4$), (2) trichlorosilane ($SiHCl_3$), (3) dichlorosilane (SiH_2Cl_2), and (4) silane (SiH_4). Depending on particular deposition conditions and film requirements, one of these sources can be used.

Doping Silicon

The active circuit elements of the IC are formed within the silicon substrate. To construct these elements, we need to create localized *n*-type and *p*-type regions by adding the appropriate dopant atoms. The process of introducing controlled amounts of impurities into the lattice of the monocrystalline silicon is known as *doping*. Dopants can be introduced selectively into the silicon using two techniques: diffusion and ion implantation.

Diffusion. The process by which a species moves as a result of the presence of a chemical gradient is referred to as diffusion. Diffusion is a time- and temperature-dependent process. To achieve maximum control, most diffusions are performed in two steps. The first step is predeposition (47), which takes place at a furnace temperature and controls the amount of impurity that is introduced. The second step, the drive-in step (47), controls the desired depth of diffusion.

Predeposition. In predisposition, the impurity atoms are made available at the surface of the wafer. The atoms of the desired element in the form of a solution of controlled viscosity can be spun on the wafer, in the same manner as the photoresist. For these spin-on dopants, the viscosity and the spin

rate are used to control the desired dopant film thickness. The wafer is then subjected to a selected high temperature to complete the predeposition diffusion. The impurity atoms can also be made available by employing a low-temperature chemical vapor deposition process in which the dopant is introduced as a gaseous compound—usually in the presence of nitrogen as a diluent. The oxygen concentration must be carefully controlled in this operation to prevent oxidation of the silicon surface of the wafer.

Drive-In. After predeposition the dopant wafer is subjected to an elevated temperature. During this step, the atoms further diffuse into the silicon crystal lattice. The rate of diffusion is controlled by the temperature employed. The concentration of the dopant atoms is maximum at the wafer surface and reduces as the silicon substrate is penetrated further. As the atoms migrate during the diffusion, this concentration changes. Hence a specific dopant profile can be achieved by controlling the diffusion process. The dopant drive-in is usually performed in an oxidizing temperature to grow a protective layer of SiO_2 over the newly diffused area.

Ion Implantation. Ion implantation is a process in which energetic, charged atoms or molecules are directly introduced into the substrate. Ion implantation (48,49) is superior to the chemical doping methods discussed previously. The most important advantage of this process is its ability to control more precisely the number of implanted dopant atoms into substrates. Using this method, the lateral diffusion is reduced considerably compared to the chemical doping methods. Ion implantation is a low-temperature process, and the parameters that control the ion implantation are amenable to automatic control. After this process the wafer is subjected to annealing to activate the dopant electrically. There are some limitations to this process. Since the wafer is bombarded with dopant atoms, the material structure of the target is damaged. The throughput is typically lower than diffusion doping process. Additionally, the equipment used causes safety hazards to operating personnel.

Photolithography

Photolithography is the most critical step in the fabrication sequence. It determines the minimum feature size that can be realized on silicon and is a photoengraving process that accurately transfers the circuit patterns to the wafer. Lithography (50,51) involves the patterning of metals, dielectrics, and semiconductors. The photoresist material is first spin coated on the wafer substrate. It performs two important functions: (1) precise pattern formation and (2) protection of the substrate during etch. The most important property of the photoresist is that its solubility in certain solvents is greatly affected by exposure to ultraviolet radiation. The resist layer is then exposed to ultraviolet light. Patterns can be transferred to the wafer using either positive or negative masking techniques. The required pattern is formed when the wafer undergoes the development step. After development, the undesired material is removed by wet or dry etching.

Resolution of the lithography process is important to this process step. It specifies the ability to print minimum size images consistently under conditions of reasonable manufacturing variation. Therefore, lithographic processes with submicron resolution must be available to build devices with submicron features. The resolution is limited by a number of factors, including (1) hardware, (2) optical properties of the resist material, and (3) process characteristics (52).

Most IC processes require 5 to 10 patterns. Each one of them needs to be aligned precisely with those already on the wafer. Typically, the alignment distance between two patterns is less than 0.2 μm across the entire area of the wafer. The initial alignment is made with respect to the crystal lattice structure of the wafer, and subsequent patterns are aligned with the existing ones. Earlier, mask alignment was done using *contact printing* (53,54), in which the mask is held just off the wafer and visually aligned. The mask is then pressed into contact with the wafer and impinged with ultraviolet light. There is a variation of this technique called *proximity printing,* in which the mask is held slightly above the wafer during exposure. Hard contact printing was preferred because it reduced the diffraction of light, but it led to a number of yield and production problems. So the projection alignment and exposure system was developed, in which the mask and wafer never touch and an optical system projects and aligns the mask onto the wafer. Since there is no damage to the mask or photoresist, the mask life is virtually unlimited. VLSI devices use projection alignment as the standard production method.

Junction Isolation

When fabricating silicon ICs, it must be possible to isolate the devices from one another. These devices can then be connected through specific electrical paths to obtain the desired circuit configuration. From this perspective, the isolation technology is one of the critical aspects of IC fabrication. For different IC types, like NMOS, CMOS, and bipolar, a variety of techniques have been developed for device isolation. The most important technique developed was termed LOCOS, for LOCal Oxidation of Silicon. This involves the formation of semirecessed oxide in the nonactive or field areas of the substrate. With the advent of submicron-size device geometries, alternative approaches for isolation were needed. Modified LOCOS processes, trench isolation, and selective epitaxial isolation were among the newer approaches adopted.

LOCOS. To isolate MOS transistors, it is necessary to prevent the formation of channels in the field regions. This implies that a large value of V_T is required in the field region, in practice about 3 to 4 V above the supply voltage. Two ways to increase the field voltage are to increase the field oxide thickness and raise the doping beneath the field oxide. Thicker field oxide regions cause high enough threshold voltages but unfortunately lead to step coverage problems, and hence thinner field oxide regions are preferred. Therefore, the doping under the field oxide region is increased to realize higher threshold voltages. Nevertheless, the field oxide is made 7 to 10 times thicker than the gate oxide. Following this, in the channel-stop implant step, ion implantation is used to increase the doping under the field oxide. Until about 1970, the thick field oxide was grown using the *grow-oxide-and-etch* approach, in which the oxide is grown over the entire wafer and then etched over the active regions. Two disadvantages of this approach prevented it from being used for VLSI applications: (1) Field-oxide steps have sharp upper corners, which poses a problem to the subsequent metallization steps; and (2) chan-

nel-stop implant must be performed before the oxide is grown. In another approach, the oxide is selectively grown over the desired field regions. This process was introduced by Appels and Kooi in 1970 (55) and is widely used in the industry. This process is performed by preventing the oxidation of the active regions by covering them with a thin layer of silicon nitride. After etching the silicon nitride layer, the channel-stop dopant can be implanted selectively. The process, has a number of limitations for submicron devices. The most important of these is the formation of the "bird's beak," which is a lateral extension of the field oxide into the active areas of the device. The LOCOS bird's beak creates other problems as junctions become shallower in CMOS ICs. The LOCOS process was therefore modified in several ways to overcome these limitations: (1) etched-back LOCOS, (2) polybuffered LOCOS, and (3) sealed-interface local oxidation (SILO) (56).

Non-LOCOS Isolation Technologies. There have been non-LOCOS isolation technologies for VLSI and ultra-large-scale integration (ULSI) applications. The most prominent of these is *trench* isolation technology. Trench technologies are classified into three categories: (1) shallow trenches (<1 μm), (2) moderate depth trenches (1 to 3 μm), and (3) deep, narrow trenches (>3 μm deep, <2 μm wide). Shallow trenches are used primarily for isolated devices of the same type and hence are considered a replacement to LOCOS. The buried oxide (BOX) (57) isolation technology uses shallow trenches refilled with a silicon dioxide layer, which is etched back to yield a planar surface. This technique eliminates the bird's beak of LOCOS. The basic BOX technique has certain drawbacks for which the technique is modified.

Metallization

This subsection describes the contact technology to realize the connections between devices, and how the different metal layers are connected to realize the circuit structure.

Contact Technology. Isolated active-device regions in the single-crystal substrate are connected through high-conductivity, thin-film structures that are fabricated over the silicon dioxide that covers the surface. An opening in the SiO_2 must be provided to allows *contacts* between the conductor film and the Si substrate. The technology involved in etching these contacts is referred to as contact technology. These contacts affect the circuit behavior because of the parasitic resistances that exist in the path between the metal to Si substrate and the region where the actual transistor action begins. Conventional contact fabrication involves the fabrication of a contact to silicon at locations where the silicon dioxide has been etched to form a window. It involves the following steps:

1. In regions where contacts are to be made, the silicon substrate is heavily doped.
2. A window or *contact hole* is etched in the oxide that passivates the silicon surface.
3. The silicon surface is cleaned to remove the thin native-oxide layer that is formed rapidly when the surface is exposed to an oxygen-containing ambient.
4. The metal film is deposited on the wafer and makes contact with silicon wherever contact holes were created. Aluminum is the most commonly used metal film.
5. After depositing the metal, the contact structure is subjected to a thermal cycle known as *sintering* or *annealing*. This helps in bringing the Si and metal into intimate contact.

Al is desired as an interconnect material because its resistivity, 2.7 $\mu\Omega$-cm, is very low, and it offers excellent compatibility with SiO_2. Al reacts with SiO_2 to form Al_2O_3, through which the Al can diffuse to reach the Si, forming an intimate Al0Si contact. But using pure Al has certain disadvantages. Since Al is polycrystalline in nature, its grain boundaries offer very fast diffusion paths for the Si at temperatures above 400°C. Hence, if a large volume of Al is available, a significant quantity of the Si can diffuse into Al. As a result, the Al from the film moves rapidly to fill in the voids created by the migrating Si, which leads to large leakage currents or electrically shorting the circuit. This effect is referred to as *junction spiking* (58). To prevent junction spiking, different techniques are used:

1. Add approximately 1% of Si to Al.
2. Add a diffusion barrier to prevent Si from diffusing into Al.
3. Decrease sintering temperature, but this increases contact resistance.
4. Add a "barrier" metal to the contact hole (59).

Of these techniques, the most commonly used is the barrier metal. The idea is to block or hinder Al from intermixing with Si. There are three main types of contact barrier metallurgies: (1) sacrificial barrier, (2) passive barrier, and (3) stuffed barrier.

The use of Al has its own problems, the most important being its high resistivity and electromigration. There is also the problem with "hillock" formation. Hillocks are formed due to the thermal expansion mismatch among Al, SiO_2, and Si. As the wafer is cooled, thermal expansion mismatch forms stresses (usually compressive), which forms these hillocks. Therefore, copper metallization has been gaining importance. Copper is preferred over Al because it has a low resistivity (1.2 $\mu\Omega$-cm) and is resistant to electromigration. In fact, copper is added in small quantities to Al to reduce the electromigration problem of Al. However, there are some real problems with copper that need to be addressed before it can replace Al:

1. Cu is a terrible contaminant in Si. It has a very high diffusivity in Si and causes junction leakage, which degrades the gate oxide integrity (GOI).
2. Cu diffuses and drifts easily through SiO_2. Hence, Cu needs to be encapsulated for use in metallization.
3. Cu oxidizes to CuO easily.
4. The etch chemistry for Cu is highly contaminated, and the wafers need to be held at higher temperatures.

Typical process steps involved in the fabrication of a 0.8 μm LOCOS n-well inverter are as follows:

1. Wafer: 1×10^{15} to 1×10^{16} CZ(p) with $\langle 100 \rangle$ crystal orientation. Epitaxial layer required because of latch-up. The thickness is 2 μm to 16 μm with 5×10^{15}.

2. Grown screen oxide layer, with the thickness in the range 400 to 1000.
3. Expose the n-well photo on the wafer.
4. n-well ion implant. Use $1 \times 10^{13}/cm^2$ phosphorous. The voltage used is 60 keV to 2 MeV.
5. n-well drive-in. This step is carried out at 1050 to 1100°C for 2 to 6 h. This activates the dopant atoms. The drive-in depth is around 1 μm to 10 μm.
6. Perform LOCOS process.
   6.1. Strip wafer.
   6.2. Pad oxide. Thickness is 100 to 400.
   6.3. Pad nitride. Thickness is 1000 to 2000. LPCVD silicon nitride is used.
   6.4. Expose the diffusion photo on the wafer.
   6.5. Etch the nitride layer.
   6.6. Expose the block field (BF) photo. This is the inverse of the n-well photo and prevents the formation of the parasitic transistors between adjacent transistors.
   6.7. Field ion implantation. $1 \times 10^{13}/cm^2$ boron at 60 keV.
   6.8. Strip the BF and the resist layer.
   6.9. Grow the field oxide. The thickness is about 4000 to 6000 of SiO_2. The process used is a pyro process at 900° to 1050°C for 3 to 6 h.
   6.10. Strip the pad nitride layer by dipping the wafer in H_3PO_4.
   6.11. Strip the pad oxide layer by dipping the wafer in 50 : 1 HF.
7. Grow a sacrificial oxide layer and strip it. The thickness is about 600 to 1000. The sacrificial oxide layer eats into the bare silicon, thus exposing fresh silicon area on which the device can be grown.
8. Grow a sacrificial gate oxide layer. Here the thickness is about 80 to 130. This layer protects the gate when the V_T implant is done.
9. V_T implant. Two masks, one for the n region and one for the p region, are used. The concentration is 1×10^{11} to $1 \times 10^{12}/cm^2$ at 5 to 30 keV.
10. Strip the sacrificial gate oxide layer using a 50 : 1 HF solution.
11. Grow the gate oxide layer. Typical thickness is 80 to 130. The gate oxide layer is grown at 800 to 900°C for 20 min.
12. Polysilicon is deposited all over the wafer. LPCVD silane is used at 620°C for a blanket deposition. The typical thickness is 2500 to 4000.
13. Polysilicon doping is done by ion implantation using 5×10^{15} phosphorous.
14. The polysilicon etch is a very critical photo/etch process.
   14.1. Polysilicon photo is exposed on the wafer.
   14.2. Reactive ion etch (RIE) is used to etch the polysilicon.
   14.3. The resist is stripped.
15. Diffusion processing
   15.1. Mask the p^+ regions.
   15.2. Perform n^+ source/drain ion implantation using $5 \times 10^{15}/cm^2$ As^{75} at 40 keV. As is used because it is slow and does not diffuse deep into the silicon substrate.
   15.3. Perform n^+ anneal at 900°C for 15 min to activate the dopant.
   15.4. Strip the resist.
   15.5. Mask the n^+ regions.
   15.6. Perform p^+ source/drain ion implantation using $1 \times 10^{15}/cm^2$ BF_2/B^{11} at 5 to 20 keV.
   15.7. Source/drain anneal at 900°C for 30 min in an oxidizing atmosphere. This is a rapid thermal process.
   15.8. Strip the resist off.
16. Interlevel dielectric. Boro-phospho silicon glass (BPSG) is used because it flows well. Typical thickness is 5000 to 8000. A 900°C reflow anneal is also performed.
17. The contact photo is exposed on the wafer. This is critical to the layout density.
18. The contacts are etched using RIE.
19. After etching, the contact resist is stripped off.
20. Metallization. Ti barrier metallurgy is used. The actual contact is made with an alloy of Al/Cu/Si with percentages 98%, 1%, and 1%, respectively. The Al alloy is sputtered onto the wafer. The typical thickness is about 1.2 μm.
21. The Metal-1 layer photo is exposed.
22. Metal-1 etch.
23. Strip resist.
24. Foaming gas anneal is performed to improve the mobility of the electrons and relieve stress on the wafer.

The inverter is finally fabricated. Figure 5 describes the process cross section of this inverter after the various steps have been performed.

TESTING

Testing is a critical factor in the design of circuits. The purpose of testing is to verify conformance to the product definition. To understand the complexity of this problem, consider a combinational circuit with n inputs. A sequence of 2^n inputs must be applied and observed to test the circuit exhaustively. Since the number of inputs are high for VLSI circuits, testing the chip exhaustively is impossible. Hence, this becomes an area of importance to circuit design. There are three main areas that need to be addressed to solve this problem:

1. Test generation
2. Test verification
3. Design for testability

Test generation corresponds to the problem of generating a minimum number of tests to verify the behavior of a circuit. The problem of test verification, which is commonly gauged by performing fault simulations, involves evaluating measures of the effectiveness of a given set of test vectors. Circuits can also be designed for testability.

Test Generation

Test generation (60) involves the search for a sequence of input vectors that allow faults to be detected at the primary device outputs. VLSI circuits are typically characterized by buried flip-flops, asynchronous circuits, indeterminate states, complex clock conditions, multiple switching of inputs simultaneously, and nonfunctional inputs. Due to these factors, an intimate knowledge of the internal circuit details is essential to develop efficient test strategies. The goal of a test generation strategy (61,62) is multifold: (1) chip design verification in conjunction with the designer, (2) incorporation of the customer's specification and patterns into the manufacturing test program, and (3) fault detection by fault simulation methods.

Test Verification

Test verification (63) involves calculating measures for how efficient the test vectors for a given circuit are. This is often accomplished by using fault models (64). Fault simulation requires a good circuit simulator to be efficient and is hence closely related to logic simulation and timing analysis. While the logic simulator verifies the functionality of a design and ensures that the timing constraints are met, the fault simulator tells the designer if enough analysis has been performed to justify committing the design to silicon. In fault simulation, the true value of a circuit and its behavior under possible faults is simulated. The fault model is a hypothesis based on an abstract model of the circuit, conformed to some precise real physical defects. To begin with, the simulator generates a fault list that identifies all the faults to be modeled. Then a set of test vectors is simulated against the fault-free and faulty models. Those faults that cause an erroneous signal at an output pin are considered as detected faults. Now the fault coverage of the test vector set can be computed as the number of faults detected over the total number of faults modeled.

The most widely used fault model is the single stuck-at fault model. This model assumes that all faults occur due to the shorting of a signal node with the power rail. A number of faults can be detected by this model, but a major disadvantage of this model is its assumption that all faults appear as stuck-at faults. The limitations of this model have led to the increased use of other models, like the stuck-open (65) and bridging fault models (66). The former can occur in a CMOS transistor or at the connection to a transistor. The bridging faults are short circuits that occur between signal lines. These represent a frequent source of failure in CMOS ICs. A majority of the random defects are manifested as timing delay faults in static CMOS ICs. These are faults in which the increased propagation delay causes gates to exceed their rated specifications. The statically designed circuits have a transient power supply that peaks when the gates are switching and then settles to a low current value in the quiescent state. The quiescent power supply current (67), known as I_{DDQ}, can be used as an effective test to detect leakage paths due to defects in the processing. The measured I_{DDQ} of a defect-free CMOS IC is approximately 20 nA. Most physical defects will elevate I_{DDQ} by one to five orders of magnitude. Thus the I_{DDQ} testing approach can be used to detect shorts not detectable by the single stuck-at fault model.

There are several other ways of applying logic and fault simulation to testing:

1. *Toggle Testing.* This is the cheapest, simplest, and least time-consuming method of applying simulation to testing. Toggle testing provides a testability measure by tracking the activity of circuit nodes. From a set of vectors, the method identifies those parts of the network that exhibit no activity. Since the test vectors do not affect these nodes, faults occurring here cannot be detected by the fault simulator.
2. *Fault Simulation of Functional Tests.* The outputs of the functional simulator can be used in the design process as an effective design analysis tool. The lists of detectable and undetectable faults generated by the simulator can be used to locate problems in the design and correct them. This results in substantial savings in development and manufacturing.
3. *Fault Simulation after New Test Vector Generation.* High-quality testing in a reasonable timeframe would require an efficient test pattern generation system and a fault simulator. Test vectors are first generated to detect specific faults, and the fault simulator determines the quality of the vector set. In this scenario, it becomes important to fault simulate after every new test vector is generated in order to catch multiple faults. Accelerated fault simulation is faster than test pattern generation.

Design for Testability

Design for testability commonly refers to those design techniques that produce designs for which tests can be generated by known methods. The advantage of these techniques are (1) reduced test generation cost, (2) reduced testing cost, (3) high-quality product, and (4) effective use of computer-aided design tools. The key to designing circuits that are testable are two concepts, *controllability* and *observability.* Controllability is defined as the ability to set and reset every node that is internal to the circuit. Observability is defined as the ability to observe either directly or indirectly the state of any node in the circuit. There are programs like SCOAP (68) that, given a circuit structure, can calculate the ability to control or observe internal circuit nodes. The concepts involved in design for testability can be categorized as follows: (1) ad hoc testing, (2) scan-based test techniques, and (3) built-in self-test (BIST).

Ad Hoc Testing. Ad hoc testing comprises techniques that reduce the combinational explosion of testing. Common techniques partition the circuit structure and add test points. A long counter is an example of a circuit that can be partitioned and tested with fewer test vectors. Another technique is the use of a bus in a bus-oriented system for testing. The common approaches can be classified as (1) partitioning techniques, (2) adding test points, (3) using multiplexers, and (4) providing for easy state reset.

Scan-Based Test Techniques. Scan-based approaches stem from the basic tenets of controllability and observability. The most popular approach is the level sensitive scan design, or LSSD, approach, introduced by IBM (69). This technique is

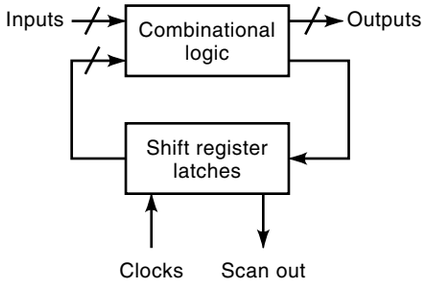

Figure 18. Level-sensitive scan design.

illustrated in Fig. 18. Circuits designed based on this approach operate in two modes—namely, normal mode and test mode. In the normal mode of operation, the shift register latches act as regular storage latches. In the test mode, these latches are connected sequentially and data can be shifted in or out of the circuit. Thus, a known sequence of data (controllability) can be input to the circuit and the results can be shifted out of the circuit using the registers (observability). The primary disadvantage of this approach is the increased complexity of the circuit design and the increased external pin count. The serial scan approach is similar to the LSSD, but the design of the shift register latch is simplified to obtain faster circuits. For most circuit designs, only the input and output register need be made scannable. This technique makes the designer responsible for deciding which registers need to be scanned and is called the partial serial scan technique (70). The parallel scan (71) approach is an extension of the serial scan in which the registers are arranged in a sort of a grid, where on a particular column all the registers have a common read/write signal. The registers that fall on a particular row have common data lines. Therefore, the output of a register can be observed by enabling the corresponding column and providing the appropriate address. Data can also be written into these registers in a similar fashion.

Built-In Self-Test. Signature analysis (72) or cyclic redundancy checking can be used to incorporate a built-in self-test module in a circuit. This involves the use of a linear feedback shift register, as depicted in Fig. 19. The value in the register will be a function of the value and number of latch inputs and the counting function of the signature analyzer. The good part of the circuit will have a particular number or signature in the register, while the faulty portion will have a different number. The signature analysis approach can be merged with the level-sensitive scan design approach to create a structure called a built-in logic block observation, or BILBO (73). Yet another approach to built-in test is called a design for autonomous test, in which the circuit is partitioned into smaller structures that are tested exhaustively. The partitioning method involves the use of multiplexers. The syndrome testing method, in which all possible inputs are applied and the number of 1's at the outputs is counted, is also a test method that requires exhaustive testing. The resultant value is compared to that of a known good machine.

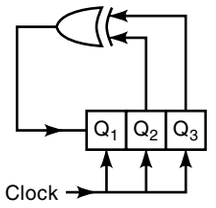

Figure 19. Linear feedback shift register.

Other Tests

So far we have discussed techniques for testing logic structures and gates. But we need testing approaches at the chip level and the system level also. Most approaches for testing chips rely on the aforementioned techniques. Memories, for instance, can use the built-in self-test techniques effectively. Random logic is usually tested by full serial scan or parallel scan methods. At the system level, traditionally the "bed-of-nails" testers have been used to probe points of interest. But with the increasing complexity of designs, system designers require a standard to test chips at the board level. This standard is the IEEE *1149 boundary scan* (74) architecture. ICs that are designed based on this standard enable complete testing of the board. The following types of tests can be performed in a unified framework: (1) connectivity test between components, (2) sampling and setting chip I/Os, and (3) distribution and collection of built-in self-test results.

BIBLIOGRAPHY

1. R. S. Schaller, Moore's Law: past, present and future, *IEEE Spectrum,* **34** (6): 52–59, June 1997.
2. J. Y. Chen, CMOS—The Emerging Technology, *IEEE Circuits Devices Mag.,* **2** (2): 16–31, 1986.
3. S. Wolf and R. N. Tauber, *Silicon Processing for the VLSI Era: Process Integration,* vol. 2, Sunset Beach, CA: Lattice Press, 1986.
4. D. Kahng and M. M. Atalla, Silicon-silicon dioxide field induced surface devices, *IRE Solid State Devices Res. Conf.,* Carnegie Inst. Technol., Pittsburgh, PA: 1960.
5. N. H. E. Weste and K. Esharaghian, *Principles of CMOS VLSI Design,* 2nd ed., Reading, MA: Addison-Wesley, 1993.
6. H. C. Pao and C. T. Shah, Effects of diffusion current on characteristics of metal-oxide (insulator)-semiconductor transistors (MOST), *Solid State Electron.,* **9**: 927–937, 1966.
7. S. M. Sze, *Physics of Semiconductor Devices,* New York: Wiley, 1981.
8. D. A. Hodges and H. G. Jackson, *Analysis and Design of Digital Integrated Circuits,* New York: McGraw-Hill, 1983.
9. P. K. Chaterjee, Gigabit age microelectronics and their manufacture, *IEEE Trans. VLSI Syst.,* **1**: 7–21, 1993.
10. Chung-Yu Wu, Jinn-Shyan Wang, and Ming-Kai Tsai, The analysis and design of CMOS multidrain logic and stacked multidrain logic, *IEEE JSSC,* **SC-22**: 47–56, 1987.
11. M. G. Johnson, A symmetric CMOS NOR gate for high speed applications, *IEEE JSSC,* **SC-23**: 1233–1236, 1988.
12. K. J. Schultz, R. J. Francis, and K. C. Smith, Ganged CMOS: trading standby power for speed, *IEEE JSSC,* **SC-25**: 870–873, 1990.
13. Y. Susuki, K. Odagawa, and T. Abe, Clocked CMOS calculator circuitry, *IEEE JSSC,* **SC-8**: 462–469, 1973.
14. T. Sakurai et al., Hot-carrier generation in submicrometer VLSI environment, *IEEE JSSC,* **SC-21**: 187–191, 1986.
15. R. H. Krambeck, C. M. Lee, and H. S. Law, High-speed compact circuits with CMOS, *IEEE JSSC,* **SC-17**: 614–619, 1982.
16. V. Friedman and S. Liu, Dynamic logic CMOS circuits, *IEEE JSSC,* **SC-19**: 263–266, 1984.
17. N. F. Gonclaves and H. J. DeMan, NORA: a racefree dynamic

CMOS technique for pipelined logic structures, *IEEE JSSC,* **SC-18**: 261–266, 1983.

18. C. M. Lee and E. W. Szeto, Zipper CMOS, *IEEE Circuits Devices,* **2** (3): 101–107, 1986.
19. L. G. Heller et al., Cascade voltage switch logic: a differential CMOS logic family, *Proc. IEEE Int. Solid State Circuits Conf.,* San Francisco, CA, Feb. 16–17, 1984.
20. T. D. Simon, A fast static CMOS NOR gate, in T. Knight and J. Savage (eds.), *Proc. 1992 Brown / MIT Conf. Advanced Res. VLSI Parallel Syst.,* Cambridge, MA: MIT Press, 1992, pp. 180–192.
21. D. Radhakrishnan, S. R. Whitaker, and G. K. Maki, Formal design procedures for pass transistor switching circuits, *IEEE JSSC,* **SC-20**: 531–536, 1985.
22. C. J. Terman, *Simulation Tools for VLSI, VLSI CAD Tools and Applications,* W. Fichtner and M. Morf (eds.), Norwell, MA: Kluwer, 1987, chapter 3.
23. L. W. Nagel, *SPICE2: a computer program to simulate semiconductor circuits,* Memo ERL-M520, Dept. Electr. Eng. Comput. Sci., Univ. California, Berkeley, May 9, 1975.
24. W. T. Weeks et al., Algorithms for ATSAP—a network analysis program, *IEEE Trans. Circuit Theory,* **CT-20**: 628–634, 1973.
25. *HSPICE User's Manual H9001,* Campbell, CA: Meta-Software, 1990.
26. C. Terman, Timing simulation for large digital MOS circuits, in A. Sangiovanni-Vincentelli (ed.), *Advances in Computer-aided Engineering Design,* vol. 1, Greenwich, CT: JAI Press, 1984, pp. 1–91.
27. J. White and A. Sangiovanni-Vincentelli, *Relaxation Techniques for the simulation of VLSI Circuits,* Hingham, MA: Kluwer, 1987.
28. R. F. Lyon, Simplified design rules for VLSI layouts, *LAMBDA,* **II** (1): 1981.
29. C. A. Mead and L. A. Conway, *Introduction to VLSI Systems,* Reading, MA: Addison-Wesley, 1980.
30. S. M. Rubin, *Computer Aids for VLSI Design,* Reading, MA: Addison-Wesley, 1987, chapter 11.
31. J. K. Ousterhout et al., Magic: A VLSI layout system, *Proc. 21st Design Autom. Conf.,* 1984, pp. 152–159.
32. U. Lauther, A min-cut placement algorithm for general cell assemblies based on a graph, *Proc. 16th Design Autom. Conf.,* 1979, pp. 1–10.
33. E. S. Kuh, Recent advances in VLSI layouts, *Proc. IEEE,* **78**: 237–263, 1990.
34. S. Kirkpatrick, C. Gelatt, and M. Vecchi, Optimization by simulated annealing, *Science,* **220** (4598): 671–680, 1983.
35. C. Sechen and A. Sangiovanni-Vincentelli,, TimberWolf 3.2: a new standard cell placement and global routing package, *Proc. 23rd Design Autom. Conf.,* Las Vegas, 1986, pp. 432–439.
36. G. W. Clow, A global routing algorithm for general cells, *Proc. 21st Design Autom. Conf.,* Albuquerque, NM, 1984, pp. 45–50.
37. G. Dupenloup, A wire routing scheme for double layer cell-layers, *Proc. 21st Design Autom. Conf.,* Albuquerque, NM, 1984, pp. 32–35.
38. E. F. Moore, The shortest path through a maze, *Proc. Int. Symp. Switching Theory,* vol. 1, Harvard University Press, 1959, pp. 285–292.
39. C. Y. Lee, An algorithm for path connection and its applications, *IRE Trans. Electron. Comput.,* 346–365, September 1961.
40. T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layouts,* New York: Wiley, 1981.
41. J. Doerschel and F. G. Kirscht, Differences in plastic deformation behavior of CZ and FZ grown Si crystals, *Phys. Status Solid,* **A64**: K85–K88, 1981.
42. W. Zuhlehner and D. Huber, *Czochralski Grown Silicon, Crystals 8,* Berlin: Springer-Verlag, 1982.
43. D. Biddle, Characterizing semiconductor wafer topography, *Microelectron. Manuf. Testing,* **15**: 15–25, 1985.
44. G. Fiegl, Recent advances and future directions in CZ-Silicon crystal growth technology, *Solid State Technol.,* **26** (8): 121–131, 1983.
45. T. Suzuki et al., CZ silicon growth in a transverse magnetic field, in *Semiconductor Silicon 1981,* Pennington, NJ: Electrochemical Society, 1981, pp. 90–93.
46. J. Bloem and L. J. Gilling, Epitaxial growth by chemical vapor deposition, in N. G. Einspruch and H. Huff (eds.), *VLSI Electronics,* vol. 12, Orlando, FL: Academic Press, 1985, chap. 3, 89–139.
47. S. Wolf and R. N. Tauber, *Silicon Processing for the VLSI Era: Process Technology,* Sunset Beach, CA: Lattice Press, 1986.
48. P. Burggraff, Ion implantation in wafer fabrication, *Semiconductor Int.,* **39**: 39–48, 1981.
49. H. Glawishnig and N. Noack, Ion implantation system concepts, in J. F. Ziegler (ed.), *Ion Implantation, Science and Technology,* Orlando, FL: Academic Press, 1984, pp. 313–373.
50. L. F. Thompson and M. J. Bowden, The lithographic process: The physics, in L. F. Thompson, C. G. Willson, M. S. Bowden (eds.), *Introduction to Microlithography,* American Chemical Society, Advances in Chemistry ser., vol. 219, 1983, pp. 15–85.
51. M. C. King, Principles of optical lithography, in N. G. Einspruch (ed.), *VLSI Electronics Micro Structure Science,* vol. 1, New York: Academic Press, 1981.
52. P. S. Gwozdz, Positive vs. negative: A photoresist analysis, *SPIE Proc., Semicond. Lithography VI,* **275**: 1981, pp. 156–182.
53. D. J. Elliot, *Integrated Circuit Fabrication Technology,* New York: McGraw-Hill, 1982, chapter 8.
54. R. C. Braken and S. A. Rizvi, Microlithography in semiconductor device processing, in N. G. Einspruch and G. B. Larabee (eds.), *VLSI Electronics—Microstructure Science,* vol. 6, Orlando, FL: Academic Press, 1983, pp. 256–291.
55. E. Kooi and J. A. Appels, Semiconductor Silicon 1973, in H. R. Huff and R. Burgess (eds.), *The Electrochem. Symp. Ser.,* Princeton, NJ, 1973, pp. 860–876.
56. P. Deroux-Dauphin and J. P. Gonchond, Physical and electrical characterization of a SILO isolation structure, *IEEE Trans. Electron Devices,* **ED-32** (11): 2392–2398, 1985.
57. M. Mikoshiba, *IEDM Tech. Dig.,* A new trench isolation technology as a replacement of LOCOS, 1984, pp. 578–581.
58. Y. Pauleau, Interconnect materials for VLSI circuits: Part II: Metal to Silicon Contacts, *Solid-State Technol.,* **30** (4): 155–162, 1987.
59. M. A. Nicolet and M. Bartur, Diffusion barriers in layered contact structures, *J. Vacuum Sci. Technol.,* **19** (3): 786–793, 1981.
60. V. D. Agrawal and S. C. Seth, *Tutorial: Test Generation for VLSI Chips,* Los Alamitos, CA: IEEE Computer Society Press, 1988.
61. S. T. Chakradhar, M. L. Bushnell, and V. D. Agrawal, Toward massively parallel automatic test generation, *IEEE Trans. CAD,* **9**: 981–994, 1990.
62. J. D. Calhoun and Franc Brglez, A framework and method for hierarchical test generation, *IEEE Trans. CAD,* **11**: 598–608, 1988.
63. H. K. Reghbati, *Tutorial: VLSI Testing and Validation Techniques,* Los Alamitos, CA: IEEE Computer Society Press, 1985.
64. W. Malay, Realistic fault modeling for VLSI testing, *IEEE / ACM Proc. 24th IEEE Design Autom. Conf.,* Miami Beach, FL, 1987, pp. 173–180.
65. A. P. Jayasumana, Y. K. Malaiya, and R. Rajsuman, Design of CMOS circuits for stuck-open fault testability, *IEEE JSSC,* **26** (1): 58–61, 1991.
66. J. M. Acken, Testing for bridging faults (shorts) in CMOS circuits, *Proc. 20th IEEE / ACM Design Autom. Conf.,* Miami Beach, FL, 1983, pp. 717–718.

67. K. Lee and M. A. Breuer, Design and test rules for CMOS circuits to facilitate IDDQ testing of bridging faults, *IEEE Trans CAD,* **11**: 659–670, 1992.

68. L. H. Goldstein and E. L. Thigpen, SCOAP: Sandia controllability/observability analysis program, *Proc. 17th Design Autom. Conf.,* 1980, pp. 190–196.

69. E. B. Eichelberger and T. W. Williams, A logic design structure for LSI testing, *J. Design Autom. Fault Tolerant Comput.,* **2** (2): 165–178, 1978.

70. R. Gupta, R. Gupta, and M. A. Breuer, An efficient implementation of the BALLAST partial scan architecture, *IFIP Proc. Int. VLSI'89 Conf.,* Munich, 1990, pp. 133–142.

71. H. Ando, Testing VLSI with random access scan, *IEEE/ACM Dig. Papers COMPCON 80,* February 1980, pp. 50–52.

72. R. A. Frohwerk, Signature analysis—a new digital field service method, *Hewlett Packard J.,* **28** (9): 2–8, 1977.

73. B. Koenemann, J. Mucha, and G. Zwiehoff, Built-in logic block observation techniques, *Dig. 1979 IEEE Test Conf.,* October 1979, pp. 37–41.

74. *IEEE Standard 1149.1: IEEE Standard Test Access Port and Boundary-Scan Architecture,* New York: IEEE Standards Board, 19.

Reading List

A. Mukherjee, *Introduction to nMOS and CMOS VLSI Systems Design,* Englewood Cliffs, NJ: Prentice-Hall, 1986.

L. A. Glasser and D. W. Dobberpuhl, *The Design and Analysis of VLSI Circuits,* Reading, MA: Addison-Wesley, 1985.

M. Annaratone, *Digital CMOS Circuit Design,* Norwell, MA: Kluwer, 1986.

M. Shoji, *CMOS Digital Circuit Technology,* Englewood Cliffs, NJ: Prentice-Hall, 1988.

D. A. Pucknell and K. Eshraghian, *Basic VLSI Design: Systems and Circuits,* Sydney: Prentice-Hall of Australia Lt., 1988.

W. Wolf, *Modern VLSI Design: A System Approach,* Englewood Cliffs, NJ: Prentice-Hall, 1994.

J. Schroeter, *Surviving the ASIC Experience,* Englewood Cliffs, NJ: Prentice-Hall, 1992.

J. M. Rabaey and M. Pedram, *Low Power Design Methodologies,* Boston: Kluwer, 1996.

S. M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits,* New York: McGraw-Hill, 1996.

C. Y. Chang and S. M. Sze, *ULSI Technology,* New York: McGraw-Hill, 1996.

N. Sherwani, *Algorithms for VLSI Physical Design Automation,* Boston: Kluwer, 1993.

R. L. Gieger, P. E. Allen, and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits,* New York: McGraw-Hill, 1990.

L. J. Herbst, *Integrated Circuit Engineering,* London, Oxford University Press, 1996.

M. S. Smith, *Application Specific Integrated Circuits,* Reading, MA: Addison-Wesley, 1997.

N. RANGANATHAN
RAJU D. VENKATARAMANA
University of South Florida