Application-Specific Integrated Circuits (ASICS)

S. K. Tewksbury
Microelectronic Systems Research Center
Dept. of Electrical and Computer Engineering
West Virginia University
Morgantown, WV 26506
(304)293-6371
May 15, 1996

Contents

1 Introduction 1
2 The Primary Steps of VLSI ASIC Design 3
3 The Increasing Impact of Interconnection Delays on Design 5
4 General Transistor Level Design of CMOS Circuits 6
5 ASIC Technologies 8
   5.1 Full Custom Design 8
   5.2 Standard Cell ASIC Technology 9
   5.3 Gate Array ASIC Technology 10
   5.4 Sea-of-Gates ASIC Technology 10
   5.5 CMOS Circuits using Megacell Elements 10
   5.6 Field Programmable Gate Arrays: Evolving to an ASIC Technology 11
6 Interconnection Performance Modeling 12
7 Clock Distribution 14
8 Power Distribution 15
9 Analog and Mixed-Signal ASICS 16
10 Summary 16

1 Introduction

Today’s VLSI CMOS technologies can place and interconnect several million transistors (representing over a million gates) on a single integrated circuit (IC) approximately 1 cm square. Provided with such a vast number of gates, a digital system designer can implement very sophisticated and complex system functions (including full systems) on a single IC. However, efficient design (including optimized performance) of such functions using all these gates is a complex puzzle of immense complexity. If this technology were to have been provided to the world overnight, it is doubtful that designers could in fact make use of this vast amount of logic on an IC.
However, this technology instead has evolved over a long period of time (about three decades), starting with only a few gates per IC, with the number of gates per IC doubling consistently about every 18 months (a progression referred to as Moore’s Law), and evolving to the present high gate densities per IC. Projections of this evolution [1] over the next 15 years, as shown in Table 1, promise continued dramatic advances in the amount of logic and memory which will be provided on individual integrated circuits. Paralleling the technology evolution, computer-aided design (CAD) tools [2, 3, 4, 5] have evolved to assist designers of the increasingly complex ICs. With these CAD tools, today’s design teams effectively have an army of very “experienced” designers embedded in the tools, capable of applying the knowledge gained over the long and steady history of IC evolution. This prior experience captured in CAD tools includes the ability to convert a high level description [6, 7] of a specific function (e.g., ALU, register, control unit, microcontroller, etc.) into an efficient physical implementation of that function.

Figure 1 is a photomicrograph of a contemporary, high performance VLSI ASIC circuit, the SHARC digital signal processor from Analog Devices, Inc.

Figure 1: Photomicrograph of the SHARC digital signal processor of Analog Devices, Inc., provided for this chapter by Douglas Garde, Analog Devices, Inc., Norwood, MA.

This chapter summarizes the design process, the gate-level physical design, and several issues which have become particularly important with today’s VLSI technologies. An important and growing issue is that of testing, including built-in testing, design for testability, and related topics (e.g., [20, 21]). This is a broad

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature size (µm)</td>
<td>0.35</td>
<td>0.25</td>
<td>0.18</td>
<td>0.13</td>
<td>0.10</td>
<td>0.07</td>
</tr>
<tr>
<td>DRAM bits/chip</td>
<td>64M</td>
<td>256M</td>
<td>1G</td>
<td>4G</td>
<td>16G</td>
<td>64G</td>
</tr>
<tr>
<td>ASIC gates/chip</td>
<td>5M</td>
<td>14M</td>
<td>26M</td>
<td>50M</td>
<td>210M</td>
<td>430M</td>
</tr>
<tr>
<td>Chip size (ASIC) (mm²)</td>
<td>450</td>
<td>660</td>
<td>750</td>
<td>900</td>
<td>1100</td>
<td>1400</td>
</tr>
<tr>
<td>Maximum number of wiring levels (logic)</td>
<td>4-5</td>
<td>5</td>
<td>5-6</td>
<td>6</td>
<td>6-7</td>
<td>7-8</td>
</tr>
<tr>
<td>On-chip speed (MHz)</td>
<td>300</td>
<td>450</td>
<td>600</td>
<td>800</td>
<td>1000</td>
<td>1100</td>
</tr>
<tr>
<td>Chip-to-board speed (MHz)</td>
<td>150</td>
<td>200</td>
<td>250</td>
<td>300</td>
<td>375</td>
<td>475</td>
</tr>
<tr>
<td>Desktop supply voltage (V)</td>
<td>3.3</td>
<td>2.5</td>
<td>1.8</td>
<td>1.5</td>
<td>1.2</td>
<td>0.9</td>
</tr>
<tr>
<td>Maximum power (Watts), heatsink</td>
<td>80</td>
<td>100</td>
<td>120</td>
<td>140</td>
<td>160</td>
<td>180</td>
</tr>
<tr>
<td>Maximum power (Watts), no heatsink</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Power (Watts), battery systems</td>
<td>2.5</td>
<td>2.5</td>
<td>3.0</td>
<td>3.5</td>
<td>4.0</td>
<td>4.5</td>
</tr>
<tr>
<td>Number of I/O</td>
<td>900</td>
<td>1350</td>
<td>2000</td>
<td>2600</td>
<td>3600</td>
<td>4800</td>
</tr>
</tbody>
</table>
2 The Primary Steps of VLSI ASIC Design

The VLSI IC design process consists of a sequence of well-defined steps [8, 9, 10, 11] related to the definition of the functions to be designed; organization of the circuit blocks implementing these logic functions within the area of the IC; verification and simulation at several stages of design (e.g., behavioral simulation, gate-level simulation, circuit simulation [12, 13], etc.); routing of physical interconnections among the blocks, and final detailed placement and transistor-level layout of the VLSI circuit. This process can also be used hierarchically to design one of the blocks comprising the overall IC, representing that circuit block in terms of simpler blocks. This establishes the “top-down” hierarchical approach, extendible to successively lower level elements of the overall design.

These general steps are illustrated in Figure 2a, roughly showing the basic steps taken. A representative example [14] of a contemporary design approach is illustrated in Figure 2b. Design approaches are continually changing and Figure 2b is merely one of several current design sequences. Below, we summarize the general steps highlighted in Figure 2a.

Figure 2: Representative VLSI design sequences. (a) Simplified but representative sequence. (b) Example design approach from recent trade journal [14]
Figure 3: Circuit stages of design. (a) Initial specification (e.g., HDL, schematic, etc.) of ASIC function in terms of functions, with the next lower level description of function C illustrated. (b) Estimated size of physical blocks implementing functions. (c) Floorplanning to organize blocks on IC. (d) Placement and routing of interconnections among blocks.

A. Behavioral Specification of Function: The behavioral specification is essentially a description of the function expected to be performed by the IC. The design can be represented by schematic capture, with the designer representing the design using block diagrams. High-level description languages (HDLs) such as VHDL [15, 16, 17] and Verilog [18] are increasingly used to provide a detailed specification of the function in a manner which is largely independent of the physical design of the function. VHDL and Verilog are “hardware description languages,” representing designs from a variety of viewpoints including behavioral descriptions, structural descriptions, and logical descriptions. Figure 3a illustrates the specification of the overall function in terms of subfunctions (A(s), B(s), ..., E(s)) as well as expansion of one subfunction (C(s)) into still simpler functions (c1, c2, c3, ..., c6).

B. Verification of Function’s Behavior: It is important to verify that the behavior specification of today’s complex ICs properly represents the behavior desired. In the case of HDL languages, there may be software “debugging” of the “program” until the desired behavior is obtained, much as programming languages need to be debugged until correct operation is obtained. Early verification is important since any errors in the specification of the function will lead to an IC which is faulty due to the design, rather than to physical defects.

C. Mapping of Logical Function into Physical Blocks: Next, the logical functions (e.g., A(s), B(s), ..., E(s) in Figure 3a) are converted into physical circuit blocks (e.g., blocks A(b), B(b), ..., E(b) in Figure 3b). Each physical block represents a logic function as a set of interconnected gates. Although details of the physical layout are not known at this point, the estimated area and aspect ratio (ratio of height to width) of each circuit block is needed to organize these blocks within the area of the IC.

D. Floorplanning: Next, the individual circuit blocks must be compactly arranged to fit within a minimum area. Floorplanning establishes this organization, as illustrated in Figure 3c. At this stage, routing of interconnections among blocks has not been performed, perhaps requiring modifications to the floorplan after routing. During floorplanning, the design of a logic function in terms of a block function can be modified to achieve shapes which better match the available IC area. For example, the block E(b) in Figure 3b has been redesigned to provide a different geometrical shape for block E(f) in the floorplan in Figure 3c.

E. Verification/Simulation of Function’s Performance: Given the floorplan, it is possible to estimate the average length of interconnections among the blocks (with the actual length not known until after
interconnection routing in step F below. Signal timing throughout the IC are estimated, allowing verification that the various circuit blocks interact within timing margins.

F. Placement and Routing: When an acceptable floorplan has been established, the next step is to complete the routing of interconnections among the various blocks of the design. As interconnections are routed, the overall IC area expands to provide the area for the wiring, with the possibility that the original floorplan (which ignored interconnections) is not optimal. Placement [19] considers various rearrangements of the circuit blocks, without changing their internal design but allowing rotations, etc. For example, the arrangement of some of the blocks in Figure 3c have been changed in the arrangement in Figure 3d.

G. Verification/Simulation of Performance: Following placement and routing, the detailed layout of the circuit blocks and interconnections has been established and more accurate simulations of signal timing and circuit behavior can be performed, verifying that the circuit behaves as desired with the inter-block interconnections in place.

H. Physical Design at Transistor/Cell Level: As the above steps are completed, the design is moving closer toward a full definition of the ASIC circuit in terms of a set of physical masks precisely specifying placement of all the transistors and interconnections. In this step, that process is completed.

I. Verification/Simulation of Performance: Before fabricating the masks and proceeding with manufacture of the ASIC circuit, a final verification of the ASIC is normally performed. Figure 2b represents this step as “golden” simulation, a process based on detailed and accurate simulation tools, tuned to the foundry’s process and providing the final verification of desired performance.

3 The Increasing Impact of Interconnection Delays on Design

In earlier generations of VLSI technology (with larger transistors and wider interconnection lines/spacings), delays through the low-level logic gates greatly dominated delays along interconnection lines. This was largely the result of the lower resistance of the larger cross section interconnections. Under these conditions, a single pass through a design sequence such as shown in Figure 2 was often adequate. In particular, the placement of blocks on the ICs, though impacting the lengths of interconnections among blocks, did not have a major impact on performance since the gate delays within the blocks were substantially larger than interconnection delays between blocks. Under such conditions, the steps of floorplanning and of placement & routing focused on such objectives as minimum overall area and minimum interconnection area.

However, as feature sizes have decreased below about 0.5µm, this condition has changed and today’s VLSI technology has interconnection delays substantially larger than logic delays. The increasing importance of interconnection delays is driven by several effects. The smaller feature size leads to interconnections with a higher resistance $R_l$ per unit length and with a higher capacitance $C_l$ per unit area (the capacitance increase also reflecting additional metal layers and coupling capacitances). For interconnection lines among high-level blocks (spanning the IC), the result is a larger RC time constant ($R_lC_lL_l^2$), where $C_l$ is the line's capacitance per unit length and $L_l$ is the line length. While interconnect delays are increasing, gate delays are decreasing. Figure 4 illustrates the general behavior on technology scaling to smaller features. A logic function $F$ in a previous generation technology requires a smaller physical area and has a higher speed in later, scaled technology (i.e., a technology with feature sizes decreased). Although the intra-block line lengths decrease (relaxing the impact within the block of higher $R_lC_l$), the inter-block lines continue to have lengths proportional to the overall IC size (which is increasing), with the larger $R_lC_l$ leading to increased RC delays on such interconnections.

As the interconnection delays have become increasingly dominant, the design process has evolved into an iterative process through the design steps, as illustrated by the back and forward annotation arrows in Figure 2a. Initial estimates of delays in step B need to be refined through back-annotation of interconnection
delay parameters obtained after floorplanning and/or after placement and routing to reflect the actual interconnection characteristics, perhaps requiring changes in the initial specification of the desired function in terms of logical and physical blocks. This iterative process moving between the logical design and the physical design of an ASIC has been problematic since often the logical design is performed by the company developing the ASIC whereas the physical design is performed by the company (the “foundry”) fabricating the ASIC. CAD tools are an important vehicle for coordination of the interface between the designer and the foundry.

4 General Transistor Level Design of CMOS Circuits

The previous section has emphasized the CAD tools and general design steps involved in designing an ASIC. A top-down approach was emphasized, with the designer addressing successively more detailed portions of the overall design through a hierarchical organization of the overall function’s description. However, the design process also presumes a considerable understanding of the bottom-up principles through which the overall IC function will eventually appear as a fully detailed specification of the transistor and interconnection structures throughout the overall IC [22, 23, 24, 25, 26].

Figure 5 illustrates the transistor-level description of a simple three-input, NAND gate. VLSI ASIC logic circuits are dominated by this general structure, with the PMOS transistors (comprising the pull-up section) connected to the supply voltage ($V_{dd}$) and the NMOS transistors (comprising the pull-down section) connected to the ground return (GND). When the logic function generates a logic “1” output, the pull-up section is shorted through its PMOS transistors to $V_{dd}$ leading to a high output voltage while the pull-down section is open, with no connection of GND to the output. When generating a logic “0” output, the pull-down section is shorted to GND while the pull-up section is open (no path to $V_{dd}$). Since the output is either a “1” or a “0”, only one of the sections (pull-up or pull-down) is shorted, with no DC current flowing directly from $V_{dd}$ to GND through the logic circuit’s pull-up and pull-down sections.

The PMOS transistors used in the pull-up section are fabricated with P-type source and drain regions on N-type substrates. The NMOS transistors used in the pull-down section, on the other hand, are fabricated...
Figure 5: Transistor representation of 3-input NAND gate. (a) Transistor representation without regard to layout. (b) Transistor representation using parallel rows of PMOS and NMOS transistors, with interconnections connected from a wiring channel.

with N-type source and drain regions on P-type substrates. Since a given silicon wafer is either N-type or P-type, a deep, opposite doping type region must be placed in the silicon wafer for those transistors needing a substrate of the opposite type. The shaded regions in Figure 5a represent the “substrate type” within which the transistors are fabricated. A deep doping of the desired “substrate” type is provided, with transistors fabricated within such deep, substrate doping “wells.” To allow tight packing of transistors, large substrate doping wells are used, which a large number of transistors placed in each well.

Each logic cell must be connected to power (Vdd) and ground, requiring that external power and ground connections to the IC be routed (on continuous metal lines to avoid resistive voltage drops) to each logic cell on the IC. Early IC technologies provided only a single level of metalization on which to route power and ground and an interdigitated layout, illustrated in Figure 6a, was adopted. Given this power and ground layout approach, channels of pull-up sections and channels of pull-down sections were placed between the power and ground interconnections, as illustrated in Figure 6b.

Bottom-up IC design is hierarchical, with the designer completing detailed layout of a specific logic function (e.g., a binary adder), placing that detailed layout (a cell) in a library of physical designs, and then reusing that library cell when other instantiations of the cell are required elsewhere in the IC. Figure 7 illustrates this cell-based approach, with cells of common height but varying width placed in rows between the power and ground lines. The straight power and ground lines shown in Figure 7 allow tight packing of adjacent rows.

To achieve tight packing of cells within a row, adjacent cells (Figure 7) are abutted against each other. Since metal interconnections are used within cells (and most basic cells are reused throughout the design) it is generally not possible to route inter-cell metal interconnections over cells. This constraint can be relaxed, as shown in Figure 8b, when additional metal layers are available, restricting a subset of the layers for intra-cell use and allowing over-the-cell routing [27] with the other layers.

When metal inter-cell interconnections can not be safely routed over cells, interconnection channels must be provided between rows of logic cells, leading to the wiring channels above and/or below rows of logic cells as shown in Figure 7. In this approach, all connections to and from a logic cell are fed from the top and/or bottom wiring channel. The width of the wiring channel is adjusted to provide space for the number of inter-cell interconnections required in the channel. Special cells providing through-cell routing can be used to support short interconnections between adjacent rows of cells. Given this layout-style at the lowest
5 ASIC Technologies

Drawing on the discussion above, the primary ASIC technologies (gate arrays, sea-of-gate arrays, standard cell ASICS, ASICS with “megacells,” and field-programmable gate arrays) can be easily summarized. For comparison, full custom VLSI is briefly described first.

5.1 Full Custom Design

In full custom design, custom logic cells are designed, starting at the lowest level (i.e., transistor-based cell design) and extending to higher levels (e.g., combinations of cells for higher level functions) to create the overall IC function. Figure 9a illustrates the general layout at the cell level. The designer can exploit new cell

level of cells, larger functions can be readily constructed, as illustrated in Figure 8a. Figure 7b illustrates interconnections (provided in the polysilicon layer under the metal layers) to the logic cells from from the wiring channel. For classical CMOS logic cells, the same set of input signals are applied to both the pull-up and pull-down sections, as in the example in Figure 5b. By organizing the sequence of transistors along the pull-up and pull-down sections properly, inputs can extend vertically between a PMOS transistor in the pull-up section and a corresponding NMOS transistor in the pull-down section. Algorithms to determine the appropriate ordering of transistors evolved early in CAD tools.

Figure 6: Power and ground distribution (interdigitated lines) with rows of logic cells and rows of wiring channels. (a) Overall power distribution and organization of logic cells and wiring channels. (b) Local region of power distribution network.

Figure 7: Cell-based logic design, with cells organized between power and ground lines and with inter-cell wiring in channels above (and/or below, also) the cell row.
designs which improve the performance of the specific function being designed, can provide interconnections through wiring areas between logic cells to create compact functions, can use the full variety of CMOS circuit designs (e.g., dynamic logic, pass-transistor logic, etc), can use previously cells developed privately for earlier ICs, and can use cells from a standard library provided by the foundry.

## 5.2 Standard Cell ASIC Technology

Consider a full custom circuit design completed using only predefined logic cells from a foundry’s specific library and physical design processes provided by standard EDA/CAD tools. This approach, \textit{standard cell design} [28, 23, 29], is one of the primary ASIC technologies. A critical issue impacting standard cell ASIC design is the quality of the standard cell library provided by the foundry. By providing a rich set of library cells, the coordination between the logical design and the physical design is substantially more effective. Figure 9a also illustrates the general standard cell approach, using standard library cells of design-specified height (according to cells used), design-specified width logic rows and varying the width of the wiring channel to accommodate the number of interconnection wires determined during place and route.
5.3 Gate Array ASIC Technology

The gate array technology [30] is based on partially prefabricated (up to but not including the final metalization layer) wafers with simple gate cells. Such non-customized wafers are stockpiled and the ASIC designer specifies the final metalization layer added to customize the gate array. Gate array cells draw on the general cell design shown earlier (Figure 5b). Figure 10 illustrates representative non-customized transistor-level cells (though different foundries use different physical layouts) with the dashed lines representing metal layers (including power and ground) added during the final metalization process.

The ASIC designer’s task is to translate the desired VLSI logic function into a physical design using the basic gate cells provided on the non-customized IC. To avoiding the routing of intercell interconnections around long rows of logic cells, some of the “cells” along the row are feedthrough cells, allowing routing of interconnections to other logic cell rows. Included in the designer’s toolset are library functions predefining the construction of more complex logic functions (e.g., adders, registers, multipliers, etc.) from the gate cells on the non-customized gate array IC.

The gate array technology shares the costs of masks among all ASIC customers and exploit high volume production of the non-customized wafers. On the other hand, construction of higher level functions must adhere to the predefined positions and type of gate cells, leading to less efficient and lower performance designs than in the standard cell approach. In addition the width of the wiring channel is fixed, limiting the number of parallel interconnections in the channel and imposing wasted area if all available wires are not used.

5.4 Sea-of-Gates ASIC Technology

The sea-of-gates technology also uses pre-manufactured, non-customized gate arrays. However, additional metalization layers are provided, with lower level metalization layer(s) used to program the internal function of the cells and the upper level layer(s) used for over-the-cell routing [27] of signals among cells (such as illustrated earlier in Figure 8b). This eliminates the need for wiring channels and feedthrough cells, leading to denser arrays of transistors.

5.5 CMOS Circuits using Megacell Elements

The examples above have focussed on low-level logic functions. However, as the complexity of VLSI ICs has increased, it has become increasingly important to include standard, high-level functions (e.g., microprocessors, digital signal processors, PCI interfaces, MPEG coders, RAM arrays, etc.) within an ASIC. For example, an earlier generation of microprocessor may offer the necessary performance and would occupy only a small portion of a present generation ASIC’s area. Including such a standard microprocessor has the advantages of allowing the ASIC design to be completed more quickly as well as providing users with the microprocessor’s standard instruction set and software development tools. Such large cells are called
5.6 Field Programmable Gate Arrays: Evolving to an ASIC Technology

The **Field Programmable Gate Array** (FPGA), like the gate array, places fixed cells on the wafer and the FPGA designer constructs more complex functions from these cells. However, the cells provided on the FPGA can be substantially more complex than the simple gates provided on the gate array. In addition, the term “field programmable” highlights the customizing of the ASIC by the user, rather than by the foundry manufacturing the FPGA. The **mask-programmable gate array** (MPGA) is similar to the FPGA (using more complex cells than the gate array) but the programming is performed by addition of the metal layer by the FPGA manufacturer.

Figure 11 shows an example of cells and programmable interconnections for a representative FPGA technology [31] (Actel, Inc.). The array of cells is constructed from two types of cell, which alternate along the logic cell rows of the FPGA. The combinational logic cell (“C-module” in Figure 11a) provides a ROM-based lookup table (LUT) able to efficiently implement a complex logic function (with four data inputs and two control signals). The sequential cell (“S-module”) adds a flip-flop to the combinational module, allowing efficient realization of sequential circuits. The interconnection approach illustrated in Figure 11c is based on (i) short vertical interconnections directly connecting adjacent modules, (ii) long vertical interconnections extending through the overall array, (iii) long horizontal interconnections extending across the overall array, (iv) points at which the long vertical interconnections can be connected to cells, and (v) points at which the long vertical and horizontal lines can be used for general routing. The long horizontal and vertical lines are broken into segments, with programmable links between successive segments. The programmer can then connect a set of adjacent line segments to create the desired interconnection line. In addition, programmable connection points allow the programmer to make transitions between the long vertical lines and the long horizontal lines. By connecting various inputs to an FPGA’s cell to either $V_{dd}$ or $GND$, the cell can be...
“programmed” to perform one of its possible functions. The basic array is complemented by additional driver and other circuitry around the perimeter of the FPGA for interfacing to the “external world.”

Different FPGA manufacturers have developed different basic cells, seeking to provide the most useful functionality in the cells for generation of overall FPGA functions. Cells range from fine-grained cells consisting of basic gates through medium-grained cells providing more complex programmable functions to large-grained cells. Different FPGA manufacturers also provide different approaches to the programming step. FPGA programming technologies include one-time programming or multiple-time programming capabilities with the programming either non-volatile (i.e., programming is retained when power is turned off) or volatile (i.e., programming is lost when power is turned off). Physical programming includes antifuse (fuse) approaches in which the programming nodes are normally off (on) and are “blown” into a permanently on (off) state, providing one-time, non-volatile programming. Electrical switches also can be used for programming, with an electrical control signal setting the state of the switch. The state control signal can be provided by an EPROM (one-time, non-volatile), an EEPROM (multiple-time, non-volatile), or an SRAM (multiple-time, volatile), with different approaches having different advantages and disadvantages.

6 Interconnection Performance Modeling

Accurate estimation of signal timing is increasingly important in contemporary VLSI ASICs and will become even more important as feature sizes decrease further. Higher clock rates impose tighter timing margins, requiring more accurate modeling of the signal delays. In addition, more sophisticated models are required for smaller feature size VLSI.

Perhaps of greatest impact is the rapid increase in the importance of interconnect delay relative to gate delay. In the earlier 1.0 µm VLSI technologies, typical gate delays were about six times the average interconnection delays. For the 0.5 µm technologies, gate delays had decreased while interconnect delays had increased, becoming approximately equal. At 0.3 µm, the decreasing gate delay and increasing interconnect delays have led to average interconnect delays about six times greater than typical gate delays. Accurate estimation of signal delays early in the design is therefore increasingly difficult, since the designer does not have a detailed knowledge of interconnection lengths and nearby lines, which can cause coupling noise, until much of the design has been completed. As the design proceeds and the interconnection lengths become better specified, parameters related to the signal performance can be fed back (back-annotated) to the earlier design steps, allowing the design to be adapted to reflect necessary changes to achieve desired performance.

In earlier VLSI technologies, a linear delay model was adequate, representing the overall delay $\tau$ from the input to one cell (cell A in Figure 12) to the input to the connected cell (cell B in Figure 12) by an analytic form such as

$$\tau = \tau_0 + k_c \cdot C_{out} + k_s \cdot \tau_s,$$

where $\tau_0$ is the intrinsic (internal) delay of the cell with no loading, $C_{out}$ is the output capacitance seen by the output driver of the cell, $\tau_s$ is the rise/fall time of the signal, and the parameters $k_c$ and $k_s$ are constants. In the case of deep submicron CMOS technologies, the overall delay must be divided into the intrinsic delay
of the gate and the delay of the interconnect, each having substantially more complex models than the linear model.

Factors impacting the intrinsic delay of gates include the following, with the input and output signals referring to logic cell A in Figure 12. (i) A 0-to-1 change in an input to cell A may cause a different delay to the output of cell A than a 1-to-0 change in that input. (ii) Starting from the time when the input starts to change, slower transition times lead to a longer delays before the threshold voltage is reached, leading to longer delays to the output. (iii) Once the input passes the threshold voltage, a slower changing input may lead to a longer delay to the output transition. (iv) The delay from a change in a given input to a change in the output may depend on the state of other inputs to the circuit. These are merely representative examples of the more complex behavior seen in the gates as the feature sizes decrease. Together, those complexities lead to a nonlinear delay model, which is typically implemented as a look-up table, rather than with an analytic expression.

The models used for interconnections [32] have also changed, reflecting the changing interconnection parameters and increasing clock rates. The three primary models are as follows.

**Lumped RC Model:** If the rise/fall times of the signal are substantially greater than the round trip propagation delay of the signal, then the voltage and current are approximately constant across the length of the interconnection. The interconnection is modeled as a single lumped resistance and a single lumped capacitance. The signal does not incur a propagation delay and at all points along the line the signal has the same rise/fall times.

**Distributed RC Model:** If the line is sufficiently long, the signal sees a decreasing resistance and capacitance toward the destination as the signal traverses the line. To represent this changing RC, the distributed RC model divides the overall line into shorter segments, each of which can be represented by the lumped RC model above. The transfer function of the overall line is then the product of the transfer functions of the sections. The propagation delay is negligible, though the rise/fall times increase as the signal propagates toward the far-end gates (significant if the line is tapped along its length to drive multiple gates).

**Distributed RLC Model:** As the rise/fall times become shorter, the relative contributions of capacitance and inductance change. In particular, the impedance of the capacitance is inversely proportional to frequency while that of the inductance is proportional to frequency. At sufficiently high data rates, the inductance effects become significant. In this case, the signal is delayed as it propagates toward the far-end gates, with the rise/fall times increasing along the line. Different terminal points of a net will see the signal at different times with different rise/fall times.

Given the wide range of lengths of signal interconnections, all three models above are relevant; the lumped RC model suitable for short interconnections, the distributed RC model for moderate speed signals on longer length interconnections, and the distributed RLC model for high speed signals on longer interconnections.

Accurate modeling of signals propagating on an interconnection requires detailed knowledge of the capacitance $C_l$ and inductance $L_l$ per unit length along the line’s length. As additional metal layers have been provided, capacitance to neighboring lines (on different or the same metal layer) has become increasingly important, even exceeding the capacitance to ground in some cases. Extraction of accurate interconnect delay parameters may require the use of 3-D field solvers, with 2-D analysis used for less accurate modeling of signal behavior.

In addition to the effects noted above, crosstalk (increasingly problematic for buses whose parallel lines run long distances) and reflections (of increasing importance as the signal frequencies increase) degrade signals. This broad range of effects impacting signal delay, distortion, and noise have made *signal integrity* an increasingly important issue in VLSI design. Signal integrity effects also appear on the “DC” power and ground lines, due to large transient currents caused by switching gates and switching drivers of output lines.
7 Clock Distribution

Signals are increasingly distorted not only by long line lengths but also by the higher clock frequency in today’s VLSI circuits, with the combination of long lines and high clock rates of particular concern. Today’s VLSI circuits include a vast number of flip-flops (often as registers) distributed across the area of the VLSI circuit. Synchronous ASICs use a common clock, distributed to each of these flip-flops. With the clock signal being the longest interconnection on the VLSI circuit and the highest frequency signal, design of the clock distribution network is critical for highest performance.

Complex synchronous ASICs are designed assuming that all flip-flops are clocked simultaneously. Clock skew is the maximum difference between the times of clock transitions at any two flip-flops of the overall VLSI circuit. The clock network must deliver clock signals to each of the flip-flops within the margins set by the allowed clock skew, margins which are substantially less than the clock period. For example, part of the 2 ns clock period of a high speed VLSI circuit operating with a 500 MHz clock is consumed by the rise/fall times of the signals appearing at the input to the flip-flop and by the specified setup and hold times of the flip-flop. The result is that the clock must be applied to the flip-flop within a time interval small compared to the clock period.

The distance over which the clock signal can travel before incurring a delay greater than the clock skew defines isochronous regions (illustrated in Figure 13a as shaded regions) within the IC. If the external clock can be provided to such regions with zero clock skew, then clock routing within the isochronous region is not critical. Figure 13a illustrates the H-tree approach, whose clock paths have equal lengths to terminal points, ideally delivering clock pulses to each of the tree’s terminal points (leaf nodes) simultaneously (zero skew). In a real circuit, precisely zero clock skew is not achieved since different network segments encounter different environments of data lines coupled electrically to the clock line segment.

In Figure 13a, a single buffer drives the entire H-tree network, requiring a large area buffer and wide clock lines toward the connection of the clock line to the external clock signal. Such a large buffer can account for up to 30% or more of the total VLSI circuit power dissipation. Figure 13b illustrates a distributed buffer approach, with a given buffer only having to drive those clock line segments to the next level of buffers. In this case, the buffers can be smaller and the clock lines can be more narrow. The 300 MHz DEC Alpha microprocessor, for example, uses an H-tree clock distribution network with multiple stages of buffering extending to the final legs of the H-tree network. Another approach to relax the clock distribution problem uses multiple I/O pins for the clock. In this case, a number of smaller H-trees can be driven separately, one
starting at each clock I/O pin.

The constraint on clock timing is a bound on clock skew, not a requirement for zero clock skew. In Figure 13c, the clock network uses multiple buffers but allows different path lengths consistent with clock skew margins. For tight margins, an H-tree can be used to deliver clock pulses to local regions in which distribution proceeds using a different buffered network approach such as that in Figure 13c.

Other approaches for clock distribution are gaining in importance. For example, a lower frequency clock can be distributed across the VLSI circuit, with phase-locked loops (PLLs) used to multiply the clock rate at various sites on the circuit. In addition to multiplying the low clock rate, the PLL can also adjust the phase of the high rate clock, correcting clock skew which may have been introduced in the routing of the low rate clock to that PLL. Another approach is to generate a local clock at a local register when the register’s input data changes. This self-timed circuit approach leads to asynchronous circuits but can be quite effective for register-transfer logic (generating a local clock for a large register).

8 Power Distribution

Today’s VLSI ASICs consume considerable power, unless specifically designed for battery-operated, low-power portable electronics [33]. A 40W IC operating at 3.3V requires a current of 12 amps, with currents increasing in future generations of high power VLSI (due not only to the higher power dissipation but also to lower $V_{dd}$).

Voltage and ground lines must be properly sized to prevent the peak current density exceeding the level at which the power lines will be physically “blown out,” leading to rapid and catastrophic failure of the circuit. An equally serious problem is gradual deterioration of a voltage or ground line, eventually leading to failure, due to electromigration. Electromigration failure affects both signal and power lines, but is particularly important in power lines due to the constant direction of the current. As current flows through an aluminum interconnection, the average force exerted on the metal atoms by the electrons leads to a slow migration of those atoms in the direction of electron flow, causing the line to migrate in the direction of the electrons. In regions of the metal line where discontinuities occur (e.g., at the naturally occurring grain boundaries), a void can develop, creating an open in the line. Fortunately, there is a current density threshold level (about 1 ma/$\mu$m) below which electromigration is insignificant. Notably, copper, in addition to having a lower resistivity than aluminum, has greater resistance to electromigration. Accurate estimates of power dissipation due to logic switching within logic blocks of the ASIC are also necessary to assess thermal heating within the IC.

Another issue in power distribution concerns ground bounce (or simultaneous switching noise), which is increasingly problematic as the number of ASIC I/O data pins increases. Consider $M$ output lines switching simultaneously to the “1” state, each of those lines outputting a current transient $I_{out}$ within a time $t_{out}$. (If $M$ output lines switch simultaneously to the “0” state, then a corresponding input current transient is produced.) The net output current ($MI_{out}$) is fed through the $V_{dd}$ pin (returned to ground in the case of outputs switching to “0”). With an inductance $L$ associated with the $V_{dd}$ pin, a transient voltage $\delta V \approx LM I_{out} / t_{out}$ is imposed on $V_{dd}$. A similar effect occurs on the ground connection for outputs switching to “0”. With a total output current of 200ma/nsec and a power pin inductance of 5 nH, the voltage transient is about 1 V. The voltage transient propagates through the IC, potentially causing logic blocks to fail to produce the correct outputs. The transient voltage can be reduced by reducing the power line inductance $L$, for example by replacing the single $V_{dd}$ and $Gnd$ pins by multiple $V_{dd}$ and $Gnd$ pins, with $K$ voltage pins reducing the inductance by a factor of $K$.

With power distribution and power line noise problems growing in importance, EDA/CAD tools are rapidly evolving to provide the designer with early estimates and final accurate assessments of various measures of current and of power dissipation.
9 Analog and Mixed-Signal ASICs

One of the exciting ASIC areas undergoing rapid development is the addition of analog integrated circuits [34, 35, 36] to the standard digital VLSI ASIC, corresponding to mixed-signal VLSI. Mixed-signal ICs allow the IC to interact directly with the real physical (and analog) world. Library cells representing various analog circuit functions supplement the usual digital circuit cells of the library, allowing the ASIC designer to add needed analog circuits within the same general framework as the addition of digital circuit cells. Automotive electronics is a representative example, with many sensors providing analog information which is converted into a digital format and analyzed using microcomputers or other digital circuits. Mixed-signal library cells include A/D and D/A converters, comparators, analog switches, sample-and-hold circuits, etc., while analog library cells include op amps, precision voltage sources, and phase-locked loops.

As such mixed-signal VLSI ASICs evolve, EDA/CAD tools will also evolve to address the performance and design issues related to analog circuits and their behavior in a digital circuit environment. In addition, analog high-level description languages (AHDLs) are being developed to support high level specifications of mixed-signal circuits.

10 Summary

For about three decades, microelectronics technologies have been evolving, starting with primitive digital logic functions and evolving to the extraordinary capabilities available in today’s VLSI ASICs. This evolution promises to continue for at least another decade, leading to VLSI ICs containing complex systems and vast memory on a single IC. ASIC technologies (including the EDA/CAD tools which guide design to a final IC) deliver this complex technology to the systems designers, including those not associated with a company having a microfabrication facility. This delivery of a highly complex technology to the average electronic systems designer is the result of a steady migration of specialized skill to very powerful EDA/CAD tools which control the complexity of the design process and the result of a need to provide a wide variety of electronics designers with access to technologies which earlier had been available only within large, vertically integrated companies.

References


