TIMING MODELS FOR MOS CIRCUITS

BY
Mark Alan Horowitz

January 1984

Prepared under
US Army Research Office
Contract No. DAAG-29-80-K-0046

Integrated Circuits Laboratory
Stanford Electronics Laboratories
Stanford University, Stanford, California

Abstract

Performance is an important aspect of integrated circuit design, and depends in part on the speed of the underlying circuits. This thesis presents a new method of analyzing MOS circuit delay, based on a single-time-constant approximation. The timing models characterize the circuit by a single parameter, which depends on the resistance and capacitance of the circuit elements. To ensure the single-time-constant approximation is valid for a particular circuit, the timing models provide both an estimate and bounds for the output waveform. For circuits where the bounds are poor, an improved timing model is derived. These simple models provide insight about circuit performance issues, as well as determining the circuit delay.

The timing models are first developed for linear networks and then are extended to model MOS circuits driven by a step input. By using the single-time-constant approximation, the output waveform of a complex MOS circuit can be modelled by the output of a circuit consisting of a single MOS transistor and a single capacitor.

Finally, a new circuit model of a gate is used to derive the output waveform of a circuit driven by an arbitrary input. The resulting timing model does not depend strongly on the shape of the input: the output waveform only depends on the input's slope at the gate's switching voltage.

Table of Contents

List of Figures

List of Symbols

α The rise time of the input normalized to the time constant of the gate; the input takes $\alpha\tau_f$ to ramp from 0 to full scale.

α_{ke} A lower bound on V_k/V_e for two nodes in an RC tree.

β_{ke} An upper bound on V_k/V_e for two nodes in an RC tree.

C_k The capacitance at node k in an RC tree. For two-tree circuits, a second subscript indicates which tree the capacitor is in: C_{k_1} is in tree 1.

C_T Total capacitance of the output network of a logic gate.

e A node in an RC tree, usually an output node.

$f(V)$ A transformation of variables that converts a nonlinear resistor into one that is pseudo-linear.

Se τ_{De} minus the integral of U_e (V_e for a linear network).

g_m The forward transconductance of a logic gate evaluated at the gate's switching voltage.

k A node in an RC tree, usually used as a summing index.

R_{ke} In an RC tree, the resistance from the root to the last node on the path to both node e and node k. Thus, R_{kk} is simply the total resistance from node k to the root. In two-tree circuits, a second subscript indicates which tree is being referenced.

$R_r(R_f)$ The effective resistance of a logic gate in the low-gain region for a rising (falling) transient.

t_d The time required for the output of a transistor cluster to reach the switching point of the next gate after the gate's input crosses its switching voltage.

t_s The time when a logic gate enters the low-gain region of its drive curves. For bounds, a second subscript is used to indicate whether this time is an upper (u) or lower (1) bound.

$T(t)$ The output waveform for a single- time-constant nonlinear circuit.

$\tau_{\alpha e}$ A lower bound on the time constant of output e's slow mode in a nonlinear circuit.

$\tau_{\beta e}$ An upper bound on the time constant of output e's slow mode in a nonlinear circuit.

τ_{De} An estimate of the output's time constant. A $'$ superscript indicates the time constant is for a two-tree circuit.

τ_f The time constant of a logic gate driven by a step input.

τ_{in} Time constant of the input waveform. For a ramp input, τ_{in} is the inverse slope; for an exponential input, it is the exponential's time constant.

τ_{Me} An estimate of the coefficient of the s^2 term in node e's frequency response divided by τ_P.

τ_P An upper bound on the lowest frequency time constant of any output in a linear network, and is equal to the sum of the open circuit time constants. A $'$ indicates the time constant is for a two-tree circuit.

τ_{Re} A lower bound on the lowest frequency time constant of output e in a linear network. A $\hat{}$ superscript indicates an improved bound; $'$ indicates the time constant is for a two-tree circuit.

τ_1 An estimate of the time constant caused by the first resistor in a mixed nonlinear circuit.

τ_1, τ_2 An estimate of the lowest two poles of node e's frequency response.

τ_z An estimate of the lowest frequency zero of node e's frequency response.

V_e The voltage at node e, usually an output voltage of the tree.

V_e^* An estimate of the voltage at node e.

V_k The voltage at node k.

V_s The switching voltage of a logic gate.

U_e The transformed voltage at node e, $f(V_e)$.

U_e^* An estimate of the transformed voltage at node e.

Acknowledgments

Without the help of many people, this thesis would not have been possible; even with their help, there were times when I still had my doubts. First and foremost, I would like to thank my parents and my friends Tom and Jeannie Blank, and Barbara Lee for keeping me going even when I thought the situation was dismal.

This thesis has benefitted from many helpful discussions I have had during my stay at Stanford. Tom Blank, Robert White, Paul Penfield, Chuck Seitz, Robert Mathews, John Newkirk, and Robert Dutton deserve special mention for reading drafts of this work and providing valuable feedback. I would like to thank John Newkirk and Robert Mathews for asking me a question about MOS circuit delay that eventually lead to the work presented in this thesis, and for their helpful discussions throughout. I am also grateful for the opportunity to work with Paul Penfield. He is still the only person I know who, after a 2 minute conversation, can point out the mistakes in my arguments. Finally, I would like to thank my advisor, Robert Dutton, who has patiently watched, supported, and guided me on my trek to find a thesis topic, and then helped refine the material into its present form.

This work was supported in part by the ARO under research contract DAAG-29-80-K-0046. The author's support by an IBM Fellowship during the 1981-1982 school year is also gratefully acknowledged.

Chapter 1

INTRODUCTION

A million-transistor integrated circuit (IC) may sound impressive, but if it cannot out-perform a thousand-transistor integrated circuit, what use is it? Performance is an important aspect of an IC and depends on two factors: the chip's micro-architecture and the speed of the underlying circuits. To develop a successful chip, the designers must consider both factors. The best micro-architecture can be made ineffectual by slow circuits, and fast circuits are wasted in a poor architecture.

Integrated circuit designers have many tools at their disposal to help them estimate the performance of a chip. At an architectural level, the tools determine how many primitive operations (clock cycles) the IC requires to complete a desired task. At a circuit level, the tools numerically estimate the delay through the internal gates. Unfortunately these circuit-level tools only can analyze small designs. They cannot simulate the entire chip to determine the time needed to perform a primitive operation at an architectural level. This thesis bridges the gap between architectural tools and circuit tools by providing a conceptually and computationally simple method of modelling the delay through Metal Oxide Semiconductor (MOS) integrated circuits.

1.1 Delay Estimation

Determining a chip's performance directly is difficult because it is a large, nonlinear circuit; an IC can contain tens of thousands of signals and hundreds of thousands of devices. However, the limited interactions in a digital system allow the IC to be partitioned into many smaller subcircuits. The chip delay then can be determined by estimating the delay through the subcircuits. Partitioning the circuit converts the chip performance estimation problem into many simpler subcircuit problems.

Currently, designers use empirical models to estimate subcircuit delays. They determine the delay's dependence on the circuit parameters by numerically simulating many circuits and performing curve fitting on the results. The disadvantages of this technique are lack of error control in the resulting models and difficulty in relating the delay back to the circuit elements.

This thesis describes a single- time-constant approximation for generating subcircuit timing models. This approximation allows the output waveform to be characterized by a simple sum of resistances and capacitances. The timing model is computationally simple, making it attractive for large MOS circuits. More importantly, the close relationship between device parameters and the timing model makes it easier for a designer to determine a component's effect on the total delay. Thus, the timing models not only estimate how fast a circuit will operate, when necessary they also can help designers determine how to speed up the circuit.

Most subcircuit outputs can be approximated by single-time-constant estimates. To ensure the validity of this approximation, bounds on the output waveform also are derived. When the bounds are poor, the estimate is not a good model of the output, and an improved estimate and bounds are derived by using a more complex model of the output waveform. The use of bounds removes the biggest limitation of simple timing models — the uncertainty in the overall accuracy.

1.2 Organization

The next chapter describes earlier work in delay modelling. To apply system-analysis techniques to MOS integrated circuits, the chip must be partitioned into transistor *clustcrs*, sub-circuits that can be viewed as digital logic blocks. From transistor cluster delays, the chip delay can be easily detcrmined.

Chapter 3 introduces a timing model for transistor clusters based on a linear transistor model. By approximating transistors by linear resistors, transistor clusters become linear RC trees. A timing model using the single-time-constant approximation, and following the derivation of Rubinstein, Penfield, and Horowitz [RP83] yields an estimate and bounds on the output waveform. These models are then extended to include systems without a single, dominant, time constant.

Chapter 4 removes the restriction that MOS transistors be modelled as linear resistors. By looking at MOS transistors in a new way, the response of the nonlinear network can be found using techniques analogous to the linear derivation. Again, a single-time-constant model (both estimate and bounds) for MOS transistor clusters is derived, which is similar, but not identical, to those for linear networks. The similarity explains why the linear models work well; the differences show where they will fail to be accurate. For circuits with multiple time constants, an improved timing model is generated, again using the same basic technique as was used to improve the linear models.

Chapter 5 describes the effect the input waveform has on the output of a transistor cluster. When the input changes gradually with time, to determine the output voltage requires modelling the output current of a logic gate versus input and output voltages. The resulting model is quite simple and can be used to show why certain gates are more sensitive to input slope than others. It leads to improved timing models for transistor clusters. For fast input waveforms, these models reduce to those derived in Chapter 3 and 4. For slow inputs, the delay through the transistor cluster increases but it is only weakly coupled to the shape of the input waveform. The input's slope at the gate's switching voltage is sufficient to predict the output waveform.

Finally, Chapter 6 presents a synopsis of delay modelling and summarizes the contributions of this thesis. Areas for further investigations are also described.

Chapter 2

DELAY ESTIMATION TECHNIQUES

Historically, tools for designing integrated circuit logic components **have been** very different from the tools for constructing systems from these integrated circuits. The design of a logic component is an analog design problem. Tools must model the electrical elements used to determine the circuit's digital characteristics: its delay, noise margins, and power dissipation. On the other hand, in large system design, the models of the underlying building blocks are digital. The system design tools use a simplified model of logic components. The digital model hides the analog aspects of the problem from the designer and the tools, allowing larger systems to be designed.

As the complexity of integrated circuits increases, the line between component and system design becomes increasingly fuzzy. Delay analysis tools for these complex circuits must merge component-analysis techniques, to determine the subcircuit delays, with system-analysis techniques, to compose subcircuit delays to yield the chip delay.

2.1 Linear System Analysis

In the 1940s, systems were neither digital nor integrated; they were multi-stage, analog, tube amplifiers. Although design techniques for these analog amplifiers might seem outdated compared to digital MOS VLSI design, the basis of the old analog analysis techniques — the single-time-constant approximation — can be used to generate MOS timing models.

Prior to the late 1960s, performance estimation techniques were computationally simple, since all calculations were done by hand. To simplify the analysis problem, all nonlinear elements were approximated by linear models. By looking at small voltage excursions around an operating point (small-signal analysis), each nonlinear element could be replaced by an effective linear element [GS69]. The response of this linearized circuit was obtained using frequency-domain analysis, since the circuit's frequency response H(s), the Laplace transform of the system's impulse response $h(t)$, could be determined directly from the circuit schematic [TA65].

The performance of an amplifier can be estimated from the low frequency terms of H(s), since they dominate the output waveform. The single-time-constant approximation models the response of the amplifier by a system with a single pole [TA65]. In 1948, Elmore reported that, for a step input, the delay through a linear amplifier was roughly equal to the first moment of its impulse response and that the output rise time was approximately $\sqrt{2\pi}$ times the second moment of the impulse response minus the first moment squared [El48]. Based on the relationship between the moments of $h(t)$ and the derivatives of H(s), the delay and rise time can be found from the frequency response. More sophisticated estimates for the output waveform have subsequently been developed, including output bounds;[†] however, the basic approach, using the low frequency terms to estimate the output, has remained the same.

Unfortunately, frequency domain analysis is valid only for linear networks. When nonlinear elements are present, superposition, and therefore Laplace transform techniques, do not apply. The growth of digital integrated circuits provided both the impetus for nonlinear analysis techniques — the circuits are intrinsically nonlinear — and an inexpensive method of performing the required computation

[†] A good review of this work is in [TA65], especially Chapter 8.

— digital computers. Circuit simulation programs developed during the late 1960s provided a method to estimate the performance of nonlinear circuits.

2.2 Circuit Simulation

A circuit simulation program computes output waveforms from a description of the circuit and its input waveforms. Second-generation programs, like SPICE2 [Na75] and ASTAP-II [WJ73], have become an essential tool for integrated circuit designers. The simulator makes finding circuit delays easy.[†] The delay is simply the amount of time between an input change and the corresponding change in the simulated output voltage. For circuits with multiple inputs and outputs, the only additional difficulty is choosing the input combination that gives the longest delay.

A circuit simulator uses numerical methods to solve the set of coupled nonlinear differential equations that define the time dependence of the nodal voltages [CL75]. The program generates the equations by first using device models to relate device currents to terminal voltages and then applying Kirchhoff's current law to obtain each capacitor current in terms of the other device currents. The net result is a set of equations relating the change in nodal voltages to the nodal voltages. Using vector notation, this set of equations can be written as

$$\mathbf{v}' = f(\mathbf{v}), \tag{2.1}$$

where v is the vector of dependent nodal voltages — nodes not driven by a voltage source.

Numerically integrating these equations provides an estimate of the output waveform. Explicit integration methods are computational simple, but cannot be

[†] Assuming that the circuit and devices are modelled accurately, and that the simulation converges [Pe82]. Simulators must solve a large set of coupled nonlinear equations to generate the initial dc solution. This task is very difficult, especially for complex circuits.

used in a general program because they have poor numerical stability. Implicit integration methods are more complex because finding the output voltages at each time step requires solving a set of nonlinear algebraic equations:

$$\mathbf{v}_{n+1} = \mathbf{v}_n + f(\mathbf{v}_{n+1})(t_{n+1} - t_n), \qquad (2.2)$$

Circuit simulators solve Eq. (2.2) by using Newton's method. The m^{th} **Newton-**Raphson interation involves finding the Jacobian of $f(\mathbf{v}^{m}_{n+1})$ and then solving the resulting set of linear equations to generate $\mathbf{v}^{m+1}_{n+1}$, the new estimate. When the difference between the new estimate and the old estimate is smaller than a set tolerance, the simulator increments the time point and the process is repeated.

Since simulators use an implicit integration method, the maximum error in the voltage estimates can be controlled by the user. Obtaining this error control, however, requires evaluating every device model at each time step and then solving a large set of nonlinear equations. As a result, circuit simulators work best with relatively small circuits.

Macro-modelling is a technique developed in the mid 1970s to help simulators analyze large circuits [RR78]. For MOS designs, the strategy was to reduce the number of nodes required to represent a circuit by eliminating all nodes internal to a logic gate [Ra73]. This simplification is possible because the internal dynamics of a gate are normally not important. Although simulators could analyze larger circuits using macro-models, simulation of the entire chip was still impossible.

To estimate the performance of MOS circuits too large for circuit simulators, a faster technique, timing simulation, was developed. The programs MOTIS [CG75] and MOTIS-C [FH77] are similar to circuit simulators in that they numerically integrate nonlinear differential equations, but they use a simplified set of equations and a simplified solution method to speed program execution. An undesirable effect of these simplifications is an increase in the uncertainty of the result; the output error can be bounded only for a limited class of circuits.

Timing simulation is based on a gate-level, rather than a transistor-level, description of the circuit. Gate macro-models lower the number of nodal voltages which must be determined. To reduce program execution time further, timing simulation only evaluates changing nodes. It updates gates with changing inputs; gates with stable inputs are ignored. Since at any particular time, most of the nodes have a stable voltage, i.e. are latent, only a small fraction of the circuit needs to be evaluated at each time step. Exploiting circuit latency together with the other simplifications make timing simulation about two orders of magnitude faster than circuit simulation.

Recently there has been work on third generation circuit simulators [HS81, LS82, SK83]. These programs are roughly the same speed as timing simulation, but maintain the accuracy and error control of circuit simulation. Although these new programs hold great promise as circuit analysis tools, they do not solve the delay modelling problem. The new programs still suffer from two problems fundamental to all numerical simulators: slow execution speed — the programs are still orders of magnitude too slow to use to analyze an entire MOS IC — and an inability to provide information on what causes the circuit delay.

Although a simulation program can determine the delay, it cannot diagnose why the circuit is slower than expected or indicate how the delay can be reduced. Solving the numerical equation yields the correct. answer but does not find the right question. This information can be obtained only when the analysis tool understands the circuits being evaluated. The result of this limitation is that experienced MOS designers use SPICE to get a feel for the technology — to calibrate their internal models — in addition to using it to analyze a particular circuit.

As the next section will show, for digital systems, the problem of estimating the delay through a large circuit can be transformed into a problem of estimating the delay through thousands of subcircuits. However, this transformation only is useful if a good subcircuit timing model exists.

2.3 System Timing Analysis

Even in the 1960s, digital systems were large and, as a result, complexity was a major issue. The controlled interactions in such a system was used to limit the complexity of simulation. Because these systems are constructed from digital circuit blocks (logic gates) the analog nature of the circuits are hidden by the external digital model. A simulator for logic need only model the delay and logical function of each block.

The controlled interaction of digital gates also enables logic simulation programs to exploit circuit latency. Since a gate is unidirectional, its output can change only if one of its inputs change. Evaluating only gates that have changing inputs (selective trace), rather than all possible gates, greatly reduces the number of evaluations at each time step.

Unfortunately, logic simulation only gives the delay for the input changes that are tested. Unless all possible machine states are tested, there is no guarantee that the longest delay found during logic simulation is the longest delay for the system. To remove this limitation, value-independent timing analysis or timing verification was developed in 1966 [KC66], butwas not applied to system design until the early 1980s, for example [McW80, Mo82]. This timing analysis uses only the timing portion of the logic specification. Without the logical description, the timing specification becomes a signal flow graph; signals flow into gates, experience some delay, and then leave. The delay through any path from an input to an output is simply the sum of the delays of the gates on that path. More important, the worst-case delay through the logic can be determined by using a PERT scheduling algorithm, whose time complexity is linear in the number of gates.

In the early 1970s IC designers began putting large systems onto a single MOS chip. Since these circuits were too complex for circuit simulation tools, the designers turned to logic simulation and timing analysis to estimate the circuit delay. Before these tools could be applied to MOS designs, two questions needed to be answered: what are the logic blocks for MOS circuits, and what are the delays through these logic blocks? When systems were built from bipolar SSI and MSI integrated circuits, the answers to these questions were obvious; the logic blocks were the ICs, and the delays were published as part of the IC specification. For large integrated circuits, the answers were no longer as clear.

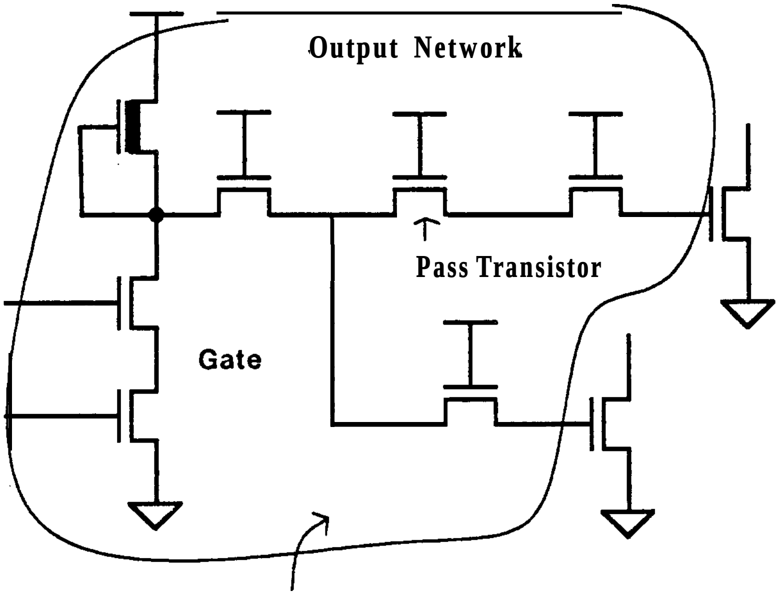
2.4 MOS Gate Delay Models

The digital model is an abstraction that suppresses the analog nature of the input and output waveforms, and represents the circuit by a boolean function. Another constraint of digital circuits is unidirectionality: a block's input is not affected by its output. Both logic simulation and timing analysis use this constraint when they assume changes only propagate from the inputs of a block to its output.

A logic block is any subcircuit that can be accurately represented by a digital model. In MOS circuits, MOS transistors provide unidirectional coupling. The transistor's gate voltage affects its source and drain voltages, but the reverse coupling is small and can be ignored.[†] The MOS transistors also provide gain, so the details of their gate voltages has a minor affect on their outputs. Thus a MOS logic block is a subcircuit whose inputs and outputs are all connected to the gates of MOS transistors. Logic blocks that cannot be subdivided into smaller logic blocks are referred to as transistor clusters; see Figure 2.1.

A transistor cluster can be viewed as an MOS logic gate and its associated output network. The output network includes any pass transistor network connected to the gate's output as well as the the parasitic resistance and capacitance of the

[†] The reverse coupling is capacitive coupling between the source and drain and the gate. For most timing questions this coupling is small enough that it can be ignored; an effective grounded capacitor can be used instead.



Transistor Cluster

Figure 2.1 An nMOS transistor cluster

output wires. Both the gate and output network can be quite complex. The gate can be a large AND-OR-Invert structure and the output network can include a large pass network, or a complex wire tree. Wire parasitics, especially the wire capacitance, are an important component of the transistor cluster, and must be included for an accurate timing model. Fortunately, programs are available to extract parasitic capacitance [SA78] and resistance [HD83] from a layout description.

Initial attempts to create delay models for transistor clusters used circuit simulators to generate empirical timing models. Such empirical models limit the kinds of transistor clusters that can be modelled — a simple gate with a capacitor load is typical [PS72]. Although recent timing models can accommodate more complex transistor clusters [AD82, OM83], the models are still very limited.

These empirical MOS timing models are more complex than the bipolar gate models. In addition to having different rise and fall delays, the delay through a MOS transistor cluster depends on the slope of the input waveform. For accurate timing analysis, this dependence means the shape of the signal during the transition is important; the timing models must determine both the delay and the slope of the output waveform.

In 1981, Penfield and Rubinstein presented a technique to bound the output waveform of a linear RC tree, based on a single-time-constant approximation [PR81]. This method can be applied to generate delay models for MOS transistor clusters by making two approximations: (1) modelling the input of the clusters by step waveforms, and (2) modelling conducting transistors by linear resistors. This technique has two advantages over empirical models. First, it can be applied to any type of transistor cluster, so a separate model for each type of cluster is no longer needed. In addition, it can relate the delay back to the circuit, showing which portions need improvement. As a result of its generality, this model was quickly incorporated into many MOS timing-analysis programs, for example, TV [Jo83] and Auto-Delay[Pu82].

The linear RC tree model has many limitations. Since transistors are not linear devices, their effective resistance values must be determined empirically. Although the timing model produces bounds, these waveforms only bound the output of the ideal linear model, not the output of the nonlinear circuit. The relationship between the model and the actual MOS circuit remains unchecked. The error in the estimated delay can be large even when the bounds are good because of the approximations used to derive the linear circuit model. Since the model provides no method to estimate or bound its error, the accuracy of results cannot be quantified.

This thesis generalizes the concepts used in finding bounds for RC trees — using the single-time-constant approximation — to generate improved timing models for MOS transistor clusters. In particular, the new models remove the need to model transistors as linear resistors and inputs as step waveforms. Thus, the waveform bounds of the new model provide a valid accuracy check on the timing estimate.

The next chapter begins by describing linear timing models for transistor clusters, since they form the foundation of the more advanced timing models.

2.5 Summary

The complexity of current MOS integrated circuits makes it infeasible to estimate a chip's delay directly using a circuit simulator. **A** chip contains too many nodes, even considering circuit latency, to numerically solve for the voltage at each node. Instead, the circuit must be viewed as a large digital system. The delay can then be estimated from the delays of its transistor clusters, the logic blocks for MOS circuits. Currently, designers use simple empirical timing models to estimate a transistor cluster's delay. The uncertainty resulting from the inaccuracy of the models is the main limitation of this technique.

Chapter 3

LINEAR NETWORKS

3.1 Overview

This chapter derives timing models for linear networks. To apply these models to a transistor cluster, MOS transistors must be approximated by linear resistors. Although this is a crude approximation, the resulting model provides a first-order estimate to the cluster's output waveform. The advantage of this linear approximation is that linear network theory can be used to help derive the timing models. The insight gained from the linear derivation is used in later chapters to derive more accurate timing models.

When MOS transistors are approximated by linear resistors, a transistor cluster becomes a linear RC tree. Because the model is linear, a qualitative description of the output waveforms can be found using frequency domain analysis. The insight gained from this analysis will guide the development of a more formal timing model. The derivation follows that of Rubinstein et al.[†] The resulting model uses three easily computed time constants to produce an estimate of and bounds on the output waveform of a linear RC tree. Estimating the output waveform avoids the difficult question of defining the delay for a system with slow rise and fall times. From the output estimate, the approximate time required for the voltage to reach any level can be determined.

[†] Penfield and Rubinstein [PR81] developed the initial method for bounding the delay in RC trees. After reading their work, this author developed a simpler derivation, which yielded slightly better bounds. This improved derivation is presented in [RP83] and is used in this chapter.

The bounds serve to check the single-time-constant approximation used to generate the estimate. For some outputs, the estimate matches the real output, yet the bounds are poor. The bounds for these circuits can be improved by simply improving τ_{Re}, one of the time constants used to generate the bounds. For other outputs, both the estimate and bounds poorly match the real output, because the real output does not have a single dominant time constant. To better represent these outputs, a two-time-constant estimate is derived.

Finally, the timing models are extended to estimate the output of an RC tree driven by another RC tree. This situation occurs in MOS circuits when a pass transistor turns on, connecting a new network to the output of a previously settled gate. The extension also provides a method to estimate the output of a logic gate whose internal capacitance is not negligible.

3.2 Modelling Transistor Clusters

A transistor cluster represents a logic gate and its associated output net; see Figure **3.1.** Although all the outputs of this cluster are logically equivalent, the voltages at the outputs need not be the same because of the resistance of the output net. Hence, a unique delay is needed for each physical output of the cluster. Both the gate and the output net must be characterized to generate a timing model for this structure. Two approximations simplify this task: the inputs to the cluster are modelled as step waveforms and conducting MOS transistors are modelled as linear resistors.

Transistors have only two possible states, if step inputs are assumed: fully conducting and not conducting. A transistor can be modelled as a non-linear resistor in series with a switch. The transistor's gate voltage controls the state of the switch. Using this model of a transistor, a resistor connected to ground models the logic gate for a low output; a resistor connected to the power supply models a high output.

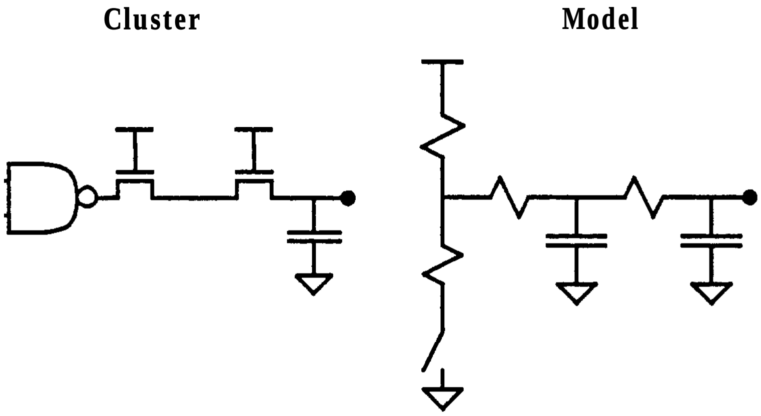


Figure **3.1** A nMOS transistor cluster.

The value of the resistor is equal to the resistance through the pullup transistor(s) in the high state and is equal to the resistance through the pulldown transistor(s) in the low state. In nMOS, the pullup transistor is ignored in the low state since the resistance of the pulldown transistors is much less than the resistance of the pullup transistors. When the gate's output changes, the resistive path in the gate drives the output net, changing all the output capacitors from the old output value to the new value.

By using a linear resistor to model a conducting transistor, the circuit model of a transistor cluster becomes a linear *RC* tree: a network of floating resistors (possibly distributed RC lines) and grounded capacitors driven by a voltage source, where a unique resistive path exists from each capacitor to the voltage source.† Figure 3.2 shows an RC tree model of a transistor cluster. The node connected to the voltage source is called the root of the tree.

† This assumes there is a unique signal path from each capacitor in the output net to the gate's output. Although it is possible to have output nets with loops, these circuits are unusual. Rarely does a signal net split off only to recombine with itself. In the situation where loops are present, the model of a transistor cluster becomes an RC mesh: The timing models for these structures are very similar to models for RC trees; see Appendix B.

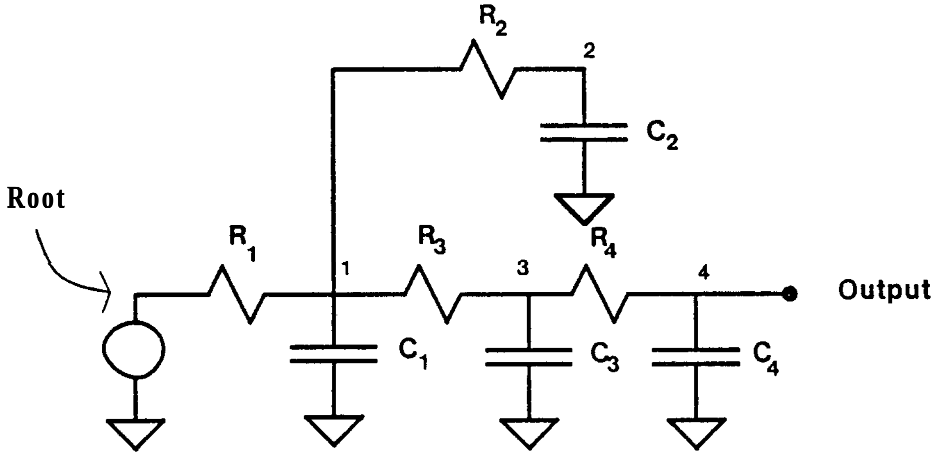


Figure 3.2 An RC tree.

The voltage source and the first resistor (R_1)model the logic gate; the rest of the RC tree models the output network. When the logic gate changes state (because its inputs change), both the value of the voltage source and the first resistor (R_1) change. During a falling transition, the output of the voltage source is ground and the first resistor is equal to the resistance through the pulldown transistor; during a rising transition, the output of the voltage source is V_{power} and the first resistor is equal to the resistance through the pullup transistor. For determining the delay, the output network is assumed to have settled to the previous state before the new change occurs: output changes do not interact. Using this assumption, the output of a transistor cluster is equal to the output of an RC tree driven by a step voltage source. This assumption is quite good for digital circuits, but breaks down for circuits that use positive feedback in an attempt to improve circuit performance. This limitation is discussed further in Chapter 6.

3.3 Qualitative Analysis

Using the linear transistor model, a MOS logic gate driving a capacitor load is represented by a single-resistor, single-capacitor circuit; its output is a simple exponential. Surprisingly, the output of most complex transistor clusters can be accurately approximated by an exponential waveform. To understand why the outputs are so simple requires looking at the frequency response of RC trees.

If an output can be well modelled by a single-resistor, single-capacitor circuit, then its frequency response must be dominated by a single pole. A dominant pole occurs when one pole is located at a much lower frequency than all the other poles and zeros.[†] In general, the output at the end of a series of identical elements has a single pole response. For example, the voltage at the end of a long polysilicon wire (modelled as a series of small RC sections) has a large number of poles, but is nicely approximated by an exponential; see Figure 3.3. The high frequency poles are most important during the initial transient, and here the approximation has its largest error. But even at its worst, the error is still small. Using a linear transistor model, the voltage at the end of a series of identical pass transistors resembles the output of an RC line and also can be approximated by an exponential. Adding a capacitive load at the end of an RC line (to model input or wire capacitance) lowers the frequency of the dominant pole, which means the single-pole estimate is an even better approximation to the real output.

There are two classes of linear networks that do not have a single-pole response. Circuits with coincident poles may have a group of low-frequency poles that dominate the output; circuits with a low-frequency pole-zero pair have a low-frequency zero that partially cancels the dominant pole, causing the output to have a two-time-constant behavior. Although these types of linear networks are easy to construct, they rarely arise as a model for a transistor cluster.

[†] To understand why higher frequency poles are less important, consider a step traversing a series of filters. Each filter corresponds to a pole of the output's frequency response. If the lowest frequency pole is put first, then it will attenuate the input's high frequency components. Subsequent poles will have only a small effect, as all the high frequency components have already been attenuated. The larger the difference in pole frequencies the smaller the effect high frequency poles have on the output.

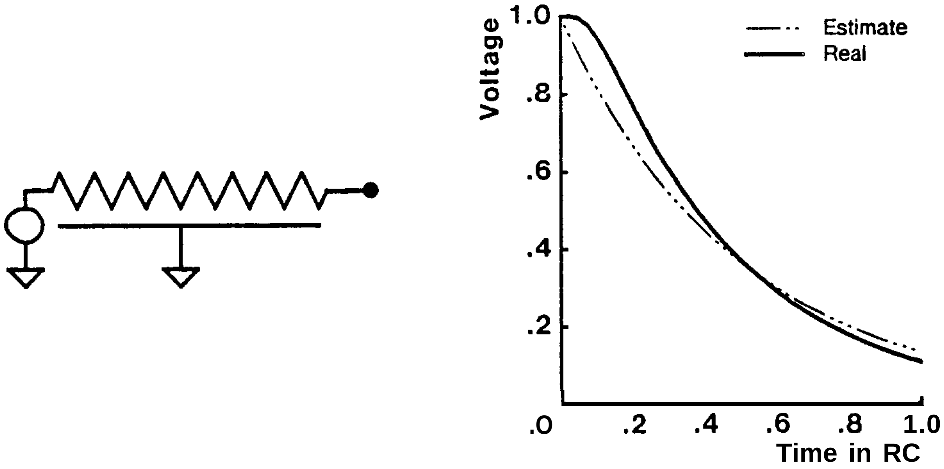


Figure 3.3 Output of a uniform RC line.

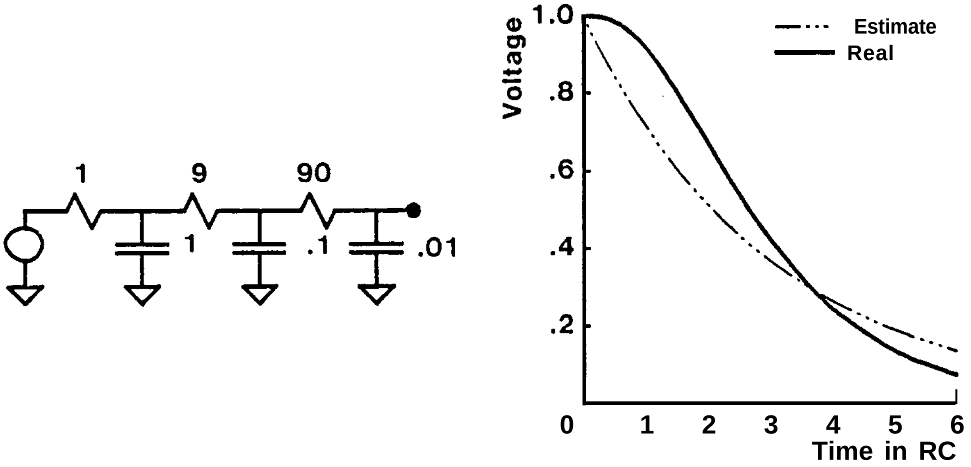


Figure 3.4 Output waveform of a circuit with three coincident poles.

Figure 3.4 shows a network with three closely spaced poles and its response. The large range in resistance and capacitance required to generate coincident poles means this type of circuit rarely occurs in MOS designs. The one exception is in modelling busses. Here the bus driver and the bus capacitance form the low impedance RC section, and the read circuitry form the high impedance section. If the time constants of the two sections are roughly equal, then the circuit will have two coincident poles.

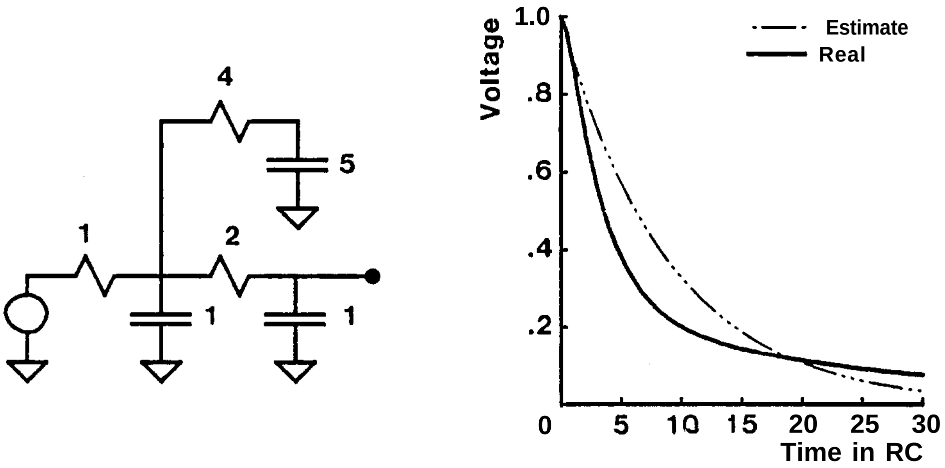


Figure 3.5 Output waveform of a circuit with a low-frequency pole-zero pair.

An example of a network with a low-frequency pole-zero pair and its output waveform are shown in Figure 3.5. The presence of a low-frequency zero partially cancels the dominant pole; causing the output waveform to have a slowly decaying tail. Physically this type of output occurs when the dominate time constant in the circuit is caused by capacitance that is located on a side branch of the tree and not directly on the path from the root to the output. The voltage at the output initially decays quickly with a time constant caused by the local output capacitance, but eventually the voltage on the distant capacitance controls the output through a voltage divider. This type of circuit also is rare as a transistor cluster model because it requires one output of a gate to be much slower than another output of the same gate. Usually a designer creates a circuit so all the outputs have roughly the same timing. The one exception to this rule is in regular structures, where control wires run in polysilicon and drive many circuits. The output closest to the control driver will be much faster than the output at the end of the poly line and will have a slowly decaying tail. But this fast output is usually not on the critical path. The output of interest is normally the slowest one, the one at the end of the poly wire, which has a single-pole response.

3.4 Single-Time-Constant Model

The following derivation provides an approximate solution and bounds for each output in a linear RC tree. Voltages have been normalized to range between 0 and 1. Att = 0 the logic inputs change causing the output net to change from a one to a zero. For $t < 0$, all the nodes in the tree are 1 and at $t = 0$, the root of the tree is grounded. The derivation for the rising waveform is similar. Nodal voltages V_n simply are replaced by $1 - V_n$.

The voltage at an output node e, V_e, is equal to the voltage drop through the resistors between e and ground — the root of the tree. This voltage can be found by replacing each capacitor by its equivalent current source, $i_n = -C_n dV_n/dt$, and then using superposition. The output voltage is the sum of the contributions from each individual current source. The voltage at node e caused by a current at node k is just the current times the resistance of the path to ground (the driven input) shared by the two nodes. Defining this resistance to be R_{ke},[†] the voltage drop caused by current i_k becomes $R_{ke}\, i_k = -R_{ke} C_k \frac{dV_k}{dt}$. Summing over all capacitor currents in the tree gives the output voltage:

$$V_e = -\sum_k R_{ke} C_k \frac{dV_k}{\mathrm{dt}}. \tag{3.1}$$

3.4.1 Waveform Estimate

Equation (3.1) is difficult to solve exactly because it involves a set of coupled differential equations. The capacitors in the tree lead to many time constants. Since most output waveforms are dominated by a single pole, a single-time-constant estimate, V_e^*, is used to model output voltage at node e. Replacing dV_k/dt by

[†] For example, R_{nn} is the total resistancc from node n to ground. In Figure 3.2, $R_{23} = R_1$ and $R_{34} = R_1 + R_3$.

$\alpha_k dV_e/dt$, where α_k is an arbitrary constant, converts Eq. (3.1) into a single-time-constant equation. The value of α_k that gives the best estimate is 1, since this value makes the integral of the error, $\int(V_e - V_e^*) = \sum_k R_{ke}C_k \int(\frac{dV_k}{dt} - \frac{dV_e}{dt})$, zero since V_k and V_e have the same starting and ending points. The estimated output waveform is a simple decaying exponential with a time constant τ_{De}:

$$V_e^*(t) = \exp(-t/\tau_{De}); \qquad \tau_{De} = \sum_k R_{ke}C_k.$$

The time constant, τ_{De}, is equal to the first moment of the circuit's impulse response, a quantity used to approximate the delay through linear amplifiers [El48]. The output estimates shown in Figures 3.3-3.5 were generated using this model.

In the frequency domain, the single-time-constant estimate is equivalent to modelling the output using a single-pole transfer function. The value of τ_{De} matches the frequency response of the output and the estimate at low frequencies: to first-order terms in s.

3.4.2 Waveform Bounds

As we saw earlier, most transistor cluster outputs can be approximated by a single-time-constant estimate. Unfortunately, there are also outputs where this estimate is poor. To make the estimate more useful, waveform bounds are derived to provide error control. If the bounds are close to the estimate, then the maximum possible timing error is small: the real output is roughly exponential in shape. If the bounds are very different from the estimate, then a more complex model is required.

To bound the voltage at output e requires a bound on Eq. (3.1). There is no simple way to bound dV_k/dt, but it is possible to bound V_k in terms of V_e. Hence, integrating Eq. (3.1) yields an equation for the integral of V_e, which can be bounded. The bounds on the integral of V_e then can be used to bound V_e.

Integrating Eq. (3.1) yields

$$\int_0^t V_e(\tau)d\tau = \sum_k R_{ke}C_k(1 - V_k(t)).$$

Defining $g_e(t)$ to be $\tau_{De} - \int V_e$[†] simplifies the above equation:

$$g_e = \sum_k R_{ke}C_kV_k. \tag{3.2}$$

Bounding V_k in terms of V_e provides a method to bound Eq. (3.2). Since all voltages in an RC tree decrease monotonically with time, the follow bounds on V_k hold:[‡]

$$\frac{R_{ke}}{R_{ee}}V_e \leq V_k \leq \frac{R_{kk}}{R_{ke}}V_e. \tag{3.3}$$

Substituting the bounds into Eq. (3.2) bounds g_e in terms of V_e:

$$\tau_{Re}V_e \leq g_e \leq \tau_P V_e \tag{3.4}$$

where

$$V_e = -\frac{dg_e}{dt}; \qquad \tau_P = \sum_k R_{kk}C_k; \qquad \tau_{Re} = \sum_k \frac{R_{ke}^2 C_k}{R_{ee}}.$$

Bounding V_k in terms of V_e causes the bounds on g_c to become single-time-constant equations. The time constant of the lower bound is τ_{Re}; the time constant of the upper bound is τ_P. Both bounds on g_e are equal to τ_{De} at t = 0, and decay to zero. Using the g_e bounds in Eq. (3.4) provides bounds on the output voltage, V_e:

$$\begin{array}{lcl} g_{e_{lower}} \leq g_e \leq \tau_P V_e, & \Rightarrow & \frac{\tau_{De}}{\tau_P}\exp(-t/\tau_{Re}) \leq V_e(t); \\ Seupper \geq g_e \geq \tau_{Re}V_e, & \Rightarrow & \frac{\tau_{De}}{\tau_{Re}}\exp(-t/\tau_P) \geq K(t). \end{array} \tag{3.5}$$

[†] Since voltages have been normalized, $\int$ V has the dimensions of time

[‡] For a complete derivation see Appendix A.

The bounds only depend on three time constants, τ_{De}, τ_{Re}, and τ_P. τ_{Re} is a lower bound on the output's time constant; τ_P is an upper bound. When all three time constants are similar in value, the estimate's maximum error is small. The error increases as the difference in the time constants increase.

The bounds on the output voltage in equation (3.5) can be improved by using the additional constraints that V_e decreases monotonically with time and $V_e \leq 1$. Using the monotonicity of V_e gives

$$g_e(t) + (t - t')V_e(t) \leq g_e(t').$$

Replacing $g_e(t)$ and $g_e(t')$ with bounds, $\tau_{Re}V_e$ and $\tau_{De}\exp(-t/\tau_P)$ respectively, leads to the following improved upper bound on the output voltage:

$$V_e(t) \leq \begin{cases} 1, & 0 \leq t \leq \tau_{De} - \tau_{Re}; \\ \dfrac{\tau_{De}}{t + \tau_{Re}}, & \tau_{De} - \tau_{Re} \leq t \leq \tau_P - \tau_{Re}; \\ \dfrac{\tau_{De}}{\tau_P} \exp\left(\dfrac{-t + \tau_P - \tau_{Re}}{\tau_P}\right), & \tau_P - \tau_{Re} \leq t. \end{cases} \tag{3.6}$$

Since g_e is $\tau_{De} - \int V_e$, using the constraint $V_e \leq 1$ gives a better lower bound on g_e, which leads to a better lower bound on V_e:

$$V_e(t) \geq \begin{cases} \dfrac{(\tau_{De} - t)}{\tau_P}, & t \leq \tau_{De} - \tau_{Re}; \\ \dfrac{\tau_{Re}}{\tau_P} \exp\left(\dfrac{-t + \tau_{De} - \tau_{Re}}{\tau_{Re}}\right), & t \geq \tau_{De} - \tau_{Re}. \end{cases} \tag{3.7}$$

Although these new bounds are tighter than the previous ones (Eq. (3.5)), the general dependence on the time constants remains the same. The difference between τ_{Re} and τ_P controls the uncertainty in the estimate. The bounds for a circuit with a low-frequency pole-zero pair are widely separated (see Figure 3.6), warning

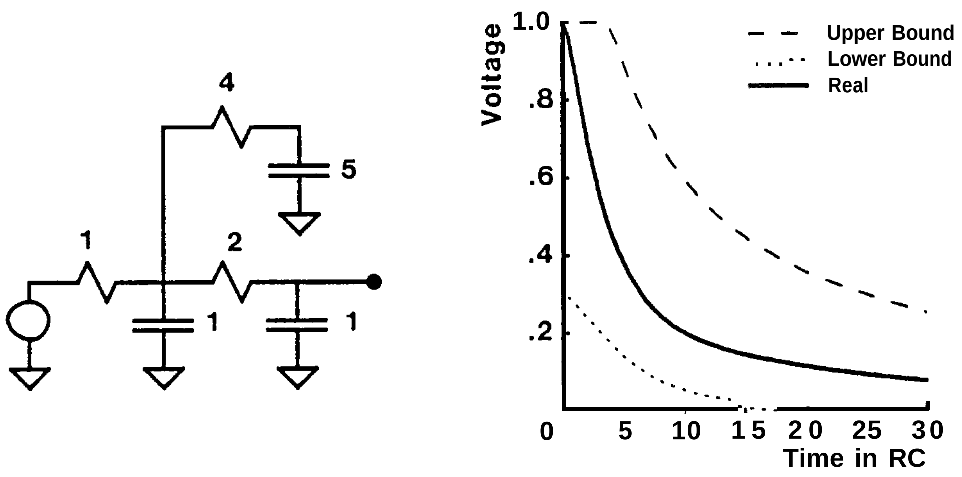


Figure 3.6 Output bounds for **a** circuit with a low-frequency pole-zero pair. τ_{De} = 9, τ_{Re} = 5, τ_P = 29

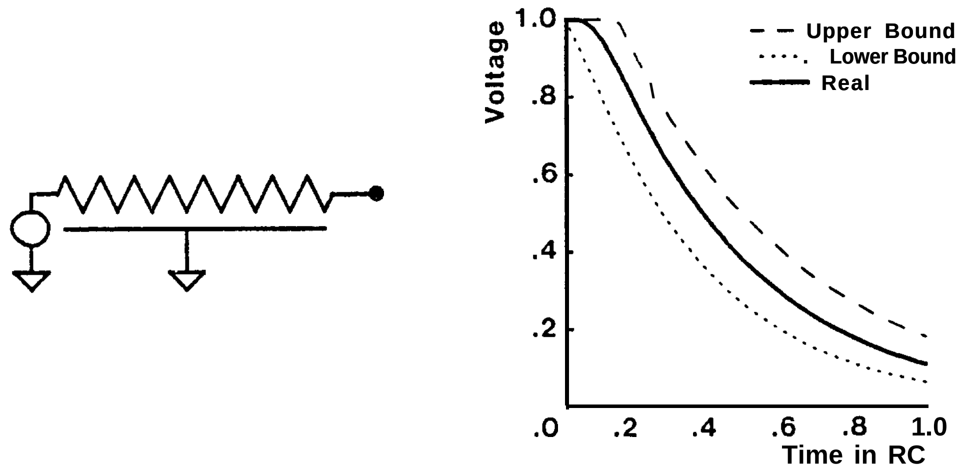


Figure 3.7 Output bounds for **a** distributed RC line. τ_{De} = .5, τ_{Re} = .33, τ_P = .5

the designer that the estimate is poor. For an output with a single-time-constant behavior, the bounds are close to the estimate, as can be seen in Figure 3.7.

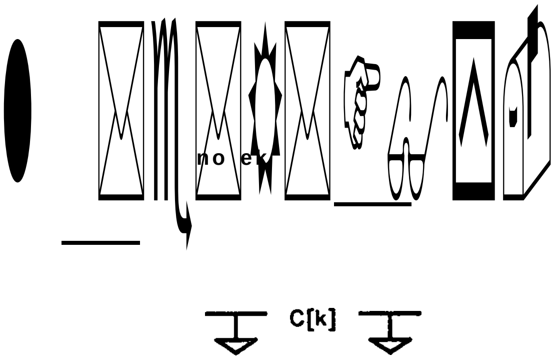
3.4.3 Computational Requirements

For the waveform estimate and bounds, three time constants characterize the output: τ_{De}, τ_{Re}, and τ_P. These time constants are needed for each physical output of a transistor cluster, since the outputs can have different waveforms. Each time constant is a sum over all the capacitors in the network (for distributed elements these sums become integrals), so for a network of n capacitors a time constant takes O(n) time to compute. Repeating this process for all outputs requires $O(n^2)$ time, assuming the number of outputs is proportional to the number of nodes (capacitors). The time complexity can be reduced to O(n) by using an algorithm that finds the time constants for all the outputs simultaneously. This reduction is possible because the time constants for different outputs share common terms; finding the time constants for the outermost nodes provides the inner time constants with little extra computational effort. An algorithm for finding τ_{De} for all nodes in a tree is given in Figure 3.8. The algorithm for τ_{Re} is analogous.

3.4.4 Time Constant Interpretation

In addition to providing output bounds, τ_{Re} and τ_P provide useful information about the network, especially for cases where the bounds are poor. For an output dominated by a cluster of n poles all at the same frequency, τ_{Re} is equal to τ_{De}/n, while τ_P remains approximately equal to τ_{De}. When low-frequency pole-zero pairs are present, τ_P is much larger than τ_{De} without affecting τ_{Re}. When τ_{Re} and τ_P are both close to τ_{De}, the output is dominated by a single pole.

A small τ_{Re} does not imply coincident poles. A single pole dominates the output of the RC tree shown in Figure 3.9, yet τ_{Re} is much less than τ_{De}. This type of circuit often occurs in modelling a bus. The bus capacitance (C_1) dominates the circuit, but is located between the driving gate (R_1) and the pass transistor reading the bus (R_2). The particular voltage bounds used to generate τ_{Re} (Eq. (3.3)) substantially underestimate the actual voltages present for this circuit. As will be shown in the next section, improving the internal voltage bounds leads to better single-time-constant bounds on the output waveform.



```
procedure find$C_T$( k:node );
    (* $C_T$[ k ] is the total cap. of the subtree with node k as its root. *)
    begin
        $C_T$[ k ] := C[ k ];
        for all children i do begin
            find$C_T$( i );
            $C_T$[ k ] := $C_T$[ k ] + $C_T$[ i ];
        end;
    end;

procedure find$\tau_D$(k:node);
    begin
        for all children i do begin
            $\tau_D$[ i ] := $\tau_D$[ k ] + R[ i ] $C_T$[ i ];
            find$\tau_D$( i );
        end;
    end;

begin
    $\tau_D$[root] := 0;
    find$C_T$(root);
    find$\tau_D$(root);
end;
```

Figure 3.8 Algorithm to find τ_{De} in linear time.

The bounds are poor for outputs where $\tau_P >> \tau_{De}$ because a single pole does not dominate the output waveform. An exponential is a crude model of an output with a slowly decaying tail. The upper bound over-estimates the output during the initial fast decay, since the output is modelled by a waveform that decays at the slow tail rate. The lower bound is also poor since it ignores the fast transient completely. It becomes a lower bound on the slow tail, which is a valid, but not useful, lower

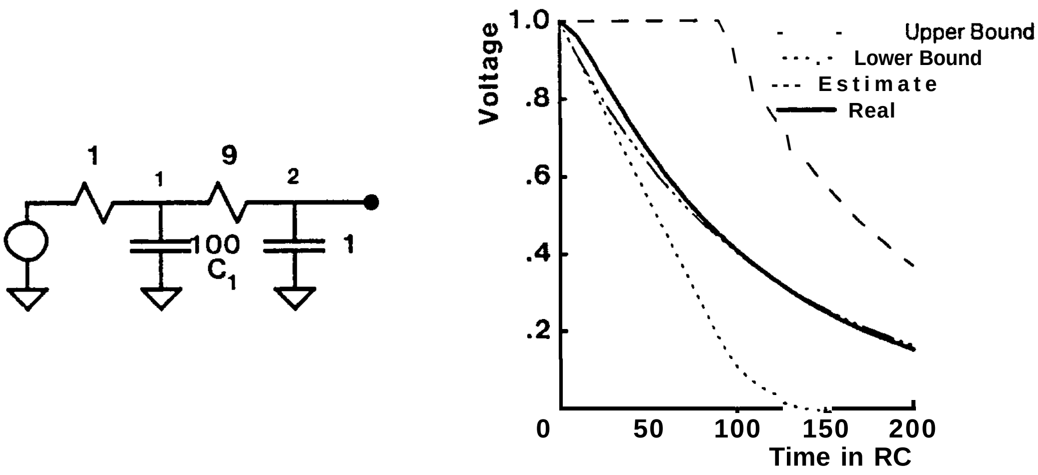


Figure 3.9 Circuit with a poor τ_{Re}. $\tau_{De} = 110, \tau_{Re} = 20, \tau_P = 110.$

bound on the actual output. To improve this output estimate a two-time-constant model is needed.

3.4.5 Improved τ_{Re}

For an RC tree with a poor τ_{Re}, a capacitor (or group of capacitors) in the middle of the network sets the circuit's dominant pole. All voltages further from the source, including the output node e, track this capacitor's voltage. Yet the lower bound on the dominant capacitor's voltage is $\frac{R_{ke}}{R_{ee}}V_e$ instead of V_e — A. Approximating a node's voltage by this lower bound causes the output bound to be poor if R_{ke} for the dominant capacitor is much less than R_{ee}. For the circuit shown in Figure 3.9, the capacitor at node 1 dominates the circuit; the output tracks this voltage. Yet, the lower bound on node 1 is $V_2/10$, a poor approximation to the actual voltage.

An improved bound on the output requires a better lower bound for V_k. The lower bound on V_k in Eq. (3.3) is the lowest voltage possible given V_e and positive capacitor currents; the bound does not use any information about the time behavior of the network. For an RC line, using this information leads to an improved voltage

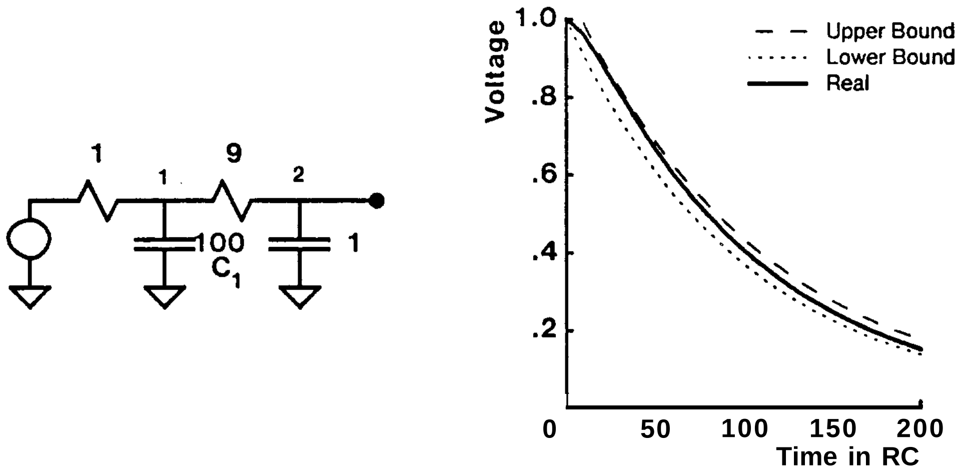


Figure 3.10 Improved output bounds using $\hat{\tau}_{Re}$. τ_{De} = 110, $\hat{\tau}_{Re}$ = **101,** τ_P = **110.**

bound, which in turn improves τ_{Re}.[†] The improved time constant, $\hat{\tau}_{Re}$, is

$$\hat{\tau}_{Re} = \sum_k \alpha_{ke} R_{ke} C_k; \qquad \alpha_{ke} = \max\left(\frac{R_{ke}}{R_{ee}}, 1 - \frac{\tau_{De} - \tau_{Dk}}{\tau_{Rk}}\right).$$

Figure 3.10 shows the improved bounds for the RC line given in Figure 3.9. The new τ_{Re} greatly improves the bounds when the dominant capacitor lies on the path from the output to driving gate. Outputs where $\hat{\tau}_{Re}$ is still much smaller than τ_{De} have coincident poles and can be better approximated by a two-time-constant estimate.

3.5 Two-Time-Constant Model

For outputs with multiple time constants, the problem with the single-time-constant model is the assumption that all voltages in the network are proportional to the output voltage. These networks have a low frequency pole caused by capacitors

[†] The improved bounds are derived using the constraint $-\frac{dV_n}{dt}/V_n \geq -\frac{dV_q}{dt}/V_q$ when node q is downstream of n. For a more complete derivation of the improved bounds see Appendix **A.**

off the path from the root to the output. As a result, the voltage on some nodes in the circuit are not tightly coupled to the output voltage. To improve the estimate and bounds for these outputs, the loosely coupled nodes are handled separately from the other nodes. The resulting timing model has two time constants. Outputs in this class are characterized by $\tau_P \gg \tau_{De}$.

3.5.1 Waveform Estimate

The improved estimate has two time constants — one to model the decay of the initial transient and the other to model the slow decay of the output tail. The improved estimate is the best two-pole, single-zero model of the output waveform. This is the simplest system that can have a slow tail in its output response.

The second order estimate has three parameters: the time constants of the two poles and the one zero. These time constants can be related to three characteristic time constants of the output: the sum of the open circuit time constants, τ_P; the first moment of the impulse response, τ_{De}; and the second moment of the impulse response, $2\tau_P(\tau_{De} - \tau_{Me})$. Setting the model’s characteristic time constants equal to the characteristic time constants of the output yields the best estimate of the output waveform. This choice of time constants matches the frequency response of the output with the frequency response of the estimate at low frequencies. Unlike the single-time-constant estimate, this model matches the output to second-order terms in s.

The model network transfer function can be written as

$$H_m(s) = \frac{1 + s\tau_z}{(1 + s\tau_1)(1 + s\tau_2)}. \tag{3.8}$$

The characteristic time constants for this system are[†]

[†] The n^{th} order moment of the impulse response is equal to -1” times the n^{th} derivative of the transfer function evaluated at s = 0.

$$\tau_P = \tau_1 + \tau_2, \qquad \tau_{De} = \tau_P - \tau_z, \qquad \tau_{Me} = \frac{\tau_1\tau_2}{\tau_1 + 72} \tag{3.9}$$

The first two time constants for the output are already known:

$$\tau_P = \sum_k R_{kk}C_k, \qquad \tau_{De} = \sum_k R_{ke}C_k.$$

τ_{Me} can be found by generating the second order moment of the impulse response, which is twice the first moment of the output voltage. Integrating by parts gives

$$\int tV_e\,dt = \sum_k R_{ke}C_k \int t\,dV_k = \sum_k R_{ke}C_k\,\tau_{Dk} = \tau_P(\tau_{De} - \tau_{Me})$$

or

$$\tau_{Me} = \sum_k R_{ke}C_k\left(1 - \frac{\tau_{Dk}}{\tau_P}\right).$$

Inverting the relations in Eq. (3.9) gives τ_1 and τ_2 in terms of the physical time constants:

$$\begin{aligned} \tau_z &= \tau_P - \tau_{De}; \\ \tau_2, \tau_1 &= \frac{\tau_P}{2}\left(1 \pm \sqrt{1 - 4\tau_{Me}/\tau_P}\right). \end{aligned}$$

The inverse Laplace transform of Eq. (3.8) gives the estimated output voltage:

$$V_e^* = \frac{(\tau_z - \tau_1)e^{-t/\tau_1} + (\tau_2 - \tau_z)e^{-t/\tau_2}}{\tau_2 - \tau_1}. \tag{3.10}$$

This equation provides an improved estimate for all outputs, even those where $\tau_P = \tau_{De}$. The improved estimate is only slightly more complex than the single-time-constant estimate, and requires two additional time constants, though τ_P is needcd for the bounds already.

Figure 3.11 shows the single-time-constant estimate, the second-order estimate, and the actual output for the networks shown in Figures 3.4 and 3.5. The improved estimate accurately reflects the actual output. The advantage of the two-time-constant estimate is its ability to model the different decay rates of the output. The estimate has the largest error when an output has many time constants, not just

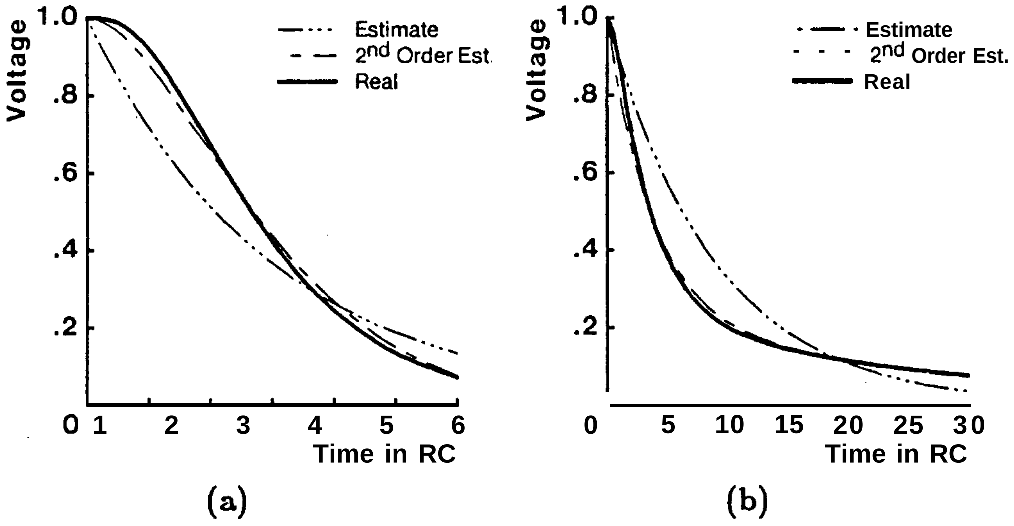


Figure 3.11 Two-time-constant Estimate for (a) the circuit with three coincident poles shown in Figure 3.4 — $\tau_{De} = 3$, $\tau_P = 3$, $\tau_{Me} = 1$; and for (b) the circuit with a low-frequency pole-zero pair shown in Figure 3.5 — $\tau_{De} = 9$, $\tau_P = 29$, $\tau_{Me} = 3.2$.

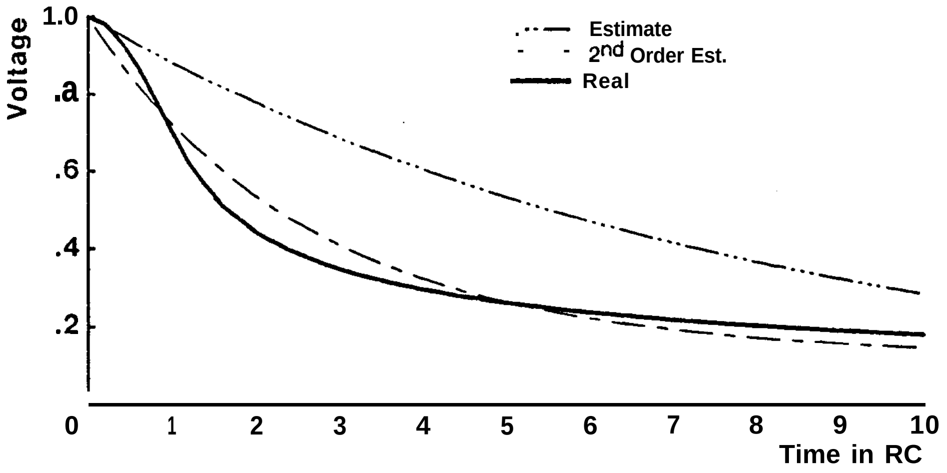


Figure 3.12 Output from the begining of a long RC line.

two. This situation occurs when the output is at the beginning of a long distributed RC line. The output and estimate in this case are shown in Figure 3.12.

3.5.2 Physical Interpretation

When $\tau_P \gg \tau_{De}$, the output estimate (Eq. (3.10)) can be viewed as the sum of the slow nodes'contribution and the fast nodes'contribution. Expanding Eq. (3.10) and only keeping terms of first order in $\frac{\tau_{Dk}}{\tau_P}$ will clarify this point.

To first order, $\tau_1 = \tau_{Me}$ and $\tau_2 = \tau_P - \tau_{Me}$, since τ_{Me} is less than τ_{De}. For an output with a low-frequency pole-zero pair, τ_{Me} represents the time constant of the initial fast transient; $\tau_P - \tau_{Me}$ represents the time constant in the tail region. τ_{Me} is roughly equal to $\sum R_{ke}C_k$, where the sum is over all nodes where $\tau_{Dk} << \tau_P$; it is the time constant of the output if the slow nodes in the circuit are removed. Thus τ_1 represents the time constant of the fast nodes, and τ_2 represents the time constant of the slow nodes.

The amplitude of the τ_2 term $(\frac{\tau_2-\tau_z}{\tau_2-\tau_1})$ to first order is $\frac{\tau_{De}-\tau_{Me}}{\tau_2}$, which is equal to $\sum_{slow} R_{ke}C_k/\tau_2$. The effect of the slow nodes is modelled by assuming they all decay exponentially with a time constant τ_2.[†] To first order in $\frac{\tau_{De}}{\tau_P}$, the estimate is simply the sum of the contributions of the fast and slow nodes:

$$V_e^*(t) \approx \left(1 - \frac{\tau_{De_2}}{\tau_2}\right)e^{-t/\tau_1} + \frac{\tau_{De_2}}{\tau_2}e^{-t/\tau_2},$$

where

$$\tau_1 = \sum_{\text{fast}} R_{ke}C_k; \qquad \tau_2 = \sum_{\text{slow}} R_{kk}C_k; \qquad \tau_{De_2} = \sum_{slow} R_{ke}C_k.$$

[†] If the voltage at the slow nodes is a simple exponential with a time constant τ_2, then their contribution to the output is $\sum_{slow} R_{ke}C_k\frac{dV_k}{dt} = \exp(-t/\tau_2)\sum_{slow}\frac{R_{ke}C_k}{\tau_P}$.

3.5.3 Bounds Improvement when $\tau_P \gg \tau_{De}$

When τ_P is much larger than τ_{De}, some capacitors decay slowly compared to the output. The relationship between the output and the voltage at these slow nodes changes as the output decays. Initially all the voltages start at the same voltage. When the output voltage is small, the slow nodes'voltage can still be quite large, since they decay more slowly than the output. Using a single constant to bound the voltage at these slow nodes in terms of the output voltage will be a poor approximation of the actual node voltage, at least for some range of output. The improved estimate overcomes this problem by grouping the slow nodes together and letting them decay at their own rate. The output is then written as the sum of two terms, one modelling the slow nodes, and the other modelling the rest of the tree. The same idea of decoupling the slow nodes from the output is used to improve the bounds.

To improve the lower bound on the output, τ_P is improved by using a better upper bound on the internal nodal voltages:

$$V_k \leq \min\left(1, \frac{R_{kk}}{R_{ke}} V_e\right).$$

This upper bound sets the voltage at the slow nodes to be 1 when the output voltage is large, thus decoupling the slow nodes during the initial transient. The improved bound on the output voltage becomes

$$V_e \geq \frac{g_{e_{lower}} - \tau_{De}^{\alpha}}{\tau_P^{\alpha}}, \tag{3.11}$$

where

$$\tau_{De}^{\alpha} = \sum_k \alpha_k R_{ke} C_k; \qquad \tau_P^{\alpha} = \sum_k (1 - \alpha_k) R_{kk} C_k; \qquad \alpha_k = \begin{cases} 1, & 2R_{ke} \leq R_{kk}; \\ 0, & 2R_{ke} > R_{kk}. \end{cases}$$

The upper bound on the output is more difficult to improve. Here the slow nodes are decoupled from the output by replacing them by a voltage source. The value of the source is chosen to give an upper bound on the actual output. Superposition is then used to write the bound as the sum of two terms, one caused by the slow nodes and the other from the network minus the slow nodes. Assuming the voltage source is placed at node v, the improved upper bound is

$$V_e \leq \frac{R_{ve}}{R_{VV}} + \frac{\tau^*_{De}}{\tau^*_{Re}} e^{-t/\tau^*_P}, \tag{3.12}$$

where the starred time constants represent time constants for the modified network, with the slow nodes shorted out by the added voltage source. Appendix C describes these bounds in more detail.

3.5.4 Computational Requirements

In both the improved bounds and estimate, a small number of time constants characterize the output waveform. Each time constant is a sum over all the nodes in the tree. For a network with n nodes, each time constant takes O(n) time to compute. However, unlike the original bounds, the terms being summed are not shared by different outputs. The bounds are improved by tailoring the time constants to a particular output. This difference means it takes $O(n^2)$ time to find improved bounds for all nodes in an RC tree, rather than the O(n) time required for the original bounds. If only a constant number of outputs need improved bounds, then the time complexity remains O(n).

The improved estimate only depends on three time constants, τ_{De}, τ_P, and τ_{Me}. Figure 3.8 has already illustrated how to find the first two time constants for all nodes in a tree in linear time. τ_{Me} also can be found for all nodes in linear time.[†] Thus, both an estimate and an improved estimate can be found for all outputs in O(n) time.

[†] Define $C^*_k = C_k(1 - \tau_{Dk}/\tau_P)$. C^*_k can be found for all outputs in O(n) time. Thus determining τ_{Me} can be reduced to finding $\sum C^*_k R_{ke}$ for every node, a task equivalent to finding τ_{De} for all nodes in a tree.

3.6 Connecting Two RC Trees

The delay models derived in the preceding sections have always assumed that a voltage source in series with a resistor drives the output network to its new value. This approximation breaks down when the capacitance within a gate is not negligible, or when a pass transistor turns on, connecting a new output net to an already settled output. For these situations, the output net is driven by an RC tree. This section derives timing models for these two-tree circuits. Like previous models, an initial single-time-constant estimate and bounds are derived, and then methods to improve these waveforms are discussed. Networks where the bounds poorly approximate the actual output closely resemble the $\tau_P >> \tau_{De}$ problem in a single RC tree, and the same techniques can be used to improve the timing model.

Figure 3.13 shows a model of a two-tree circuit. It consists of two RC trees, where an output, n, of the first tree connects to the input of the second tree. As before, the voltages are normalized to range between 0 and 1, and the output waveform for the falling edge is derived. For a falling transient, thc root of the first tree is grounded, and the second tree is precharged to 1. At t = 0 the switch between the two trees closes. The voltage at any node in tree 2 can be written as the sum of the voltages caused by each capacitor current:

$$V_e = -\sum_{k\in\text{tree } 1} R_{kn_1} C_{k_1} \frac{dV_{k_1}}{dt} - \sum_{k\in\text{tree } 2} (\mathrm{R}_{nn_1} + R_{ke_2}) C_{k_2} \frac{dV_{k_2}}{\mathrm{dt}}. \qquad (3.13)$$

3.6.1 Single-Time-Constant Model

Both the estimate and bounds use the same formula derived for a single RC tree; only the time constant definitions change. For a single-time-constant estimate, dV_k/dt is replaced by $\alpha_k dV_e/dt$. Matching the integral of the estimate with thc integral of the real output again gives the best value of α_k. Since nodes in tree 1 start and end at ground, their α_k is zero. For nodes in tree 2, α_k is one. The

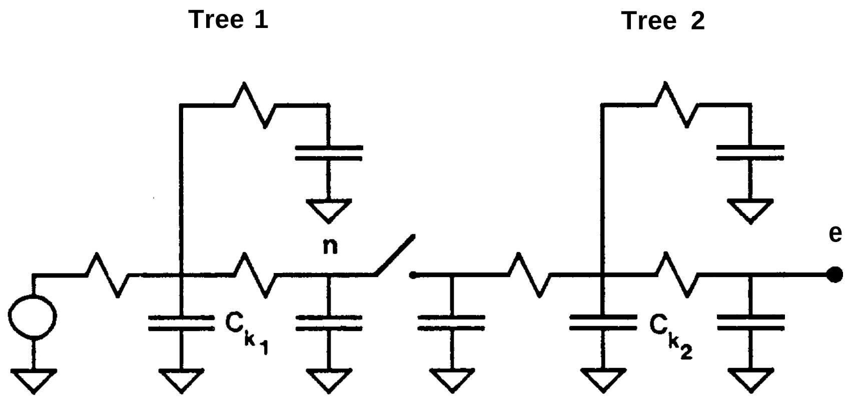


Figure 3.13 Two- tree model.

estimate is a decaying exponential with a time constant τ'_{De}:

$$V_e^*(t) = \exp(-t/\tau'_{De}); \qquad \tau'_{De} = \sum_{k\in 2}(R_{nn_1} + R_{ke_2})C_{k_2}.$$

For the single-time-constant estimate, tree 1 can be modelled by a resistor of **value** R_{nn_1}: the capacitors in tree 1 have no effect. Thus for the output estimate, internal gate capacitance can be ignored if its voltage at the beginning and end of the transient is the same; only capacitors that change voltage need to be modelled.

A bound on the output voltage can be found by integrating equation (3.13) as was done in Section 3.4.2. However, the internal voltage bounds for the two-tree network differs from the single-tree bounds. The single-tree bounds only apply when the current from each capacitor, $-CdV/dt$, is positive. When two trees are connected together, the nodes in tree 1 first rise and then fall. The only lower bound for these nodes is $V_{k_1} \geq 0$. Since the input of the second tree is always ≥ 0, a lower bound for these nodes when the input is grounded will also be a lower bound in this case: $V_{k_2} \geq \frac{R_{ke_2}}{R_{ee_2}} V_e$. Using these lower bounds gives τ_{Re} for a two-tree output:

$$\tau'_{Re} = \sum_{k\in 2} \frac{(R_{nn_1} + R_{ke_2})R_{ke_2}}{R_{ee_2}} C_{k_2}.$$

An upper bound for the internal voltages can be found by using the monotonicity of voltage along a path: $V_n \leq V_k$ if node k is downstream of node n. For

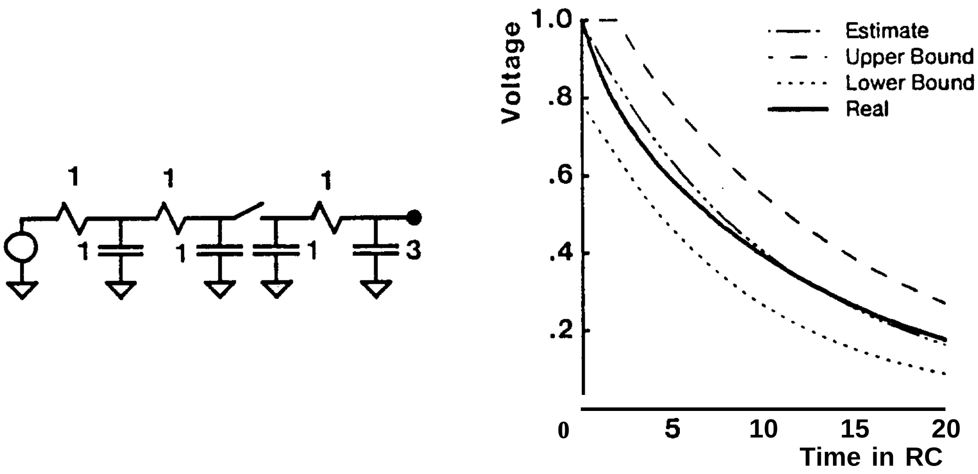


Figure 3.14 **single-time-constant two-tree output.** $\tau_{De}=11, \tau_P=14,$ τ_{Re} = 9.

tree 1 this constraint leads to $V_{k_1} \leq (R_{kk_1}/R_{ke_1})V_e$. For tree 2 the upper bound is $V_{k_2} \leq (R_{kk_2}/R_{ke_2})V_e$. Using these upper bounds gives τ_P for this network:

$$\tau'_{Pe} = \sum_{k\in 1} R_{kk_1}C_{k_1} + \sum_{k\in 2} \frac{(R_{nn_1} + R_{ke_2})R_{kk_2}}{R_{ke_2}} C_{k_2}.$$

These time constants together with Eq. (3.5) (or Eq. (3.6)-(3.7)) bound the output waveform. Figure 3.14 shows the estimate, bounds, and real output for a circuit where the capacitance of the second tree is large compared to the capacitance in tree 1. This situation occurs when a pass transistor turns on, connecting a large output net to a previous settled gate.

The bounds are poor when a tree with small nodal capacitance is attached to a slow, highly capacitive tree 1. This situation occurs when a pass transistor turns on to read the value of a long bus. The capacitance of the bus greatly exceeds the capacitance of the read network. The capacitance of the first tree only affects τ'_P. When this capacitance is large, τ'_P is much larger than τ'_{Re}. As for single-tree networks, $\tau'_P >> \tau'_{Re}$ indicates that the output has a slow tail. The bounds and estimate for this type of output are shown in Figure 3.15, along with the actual output.

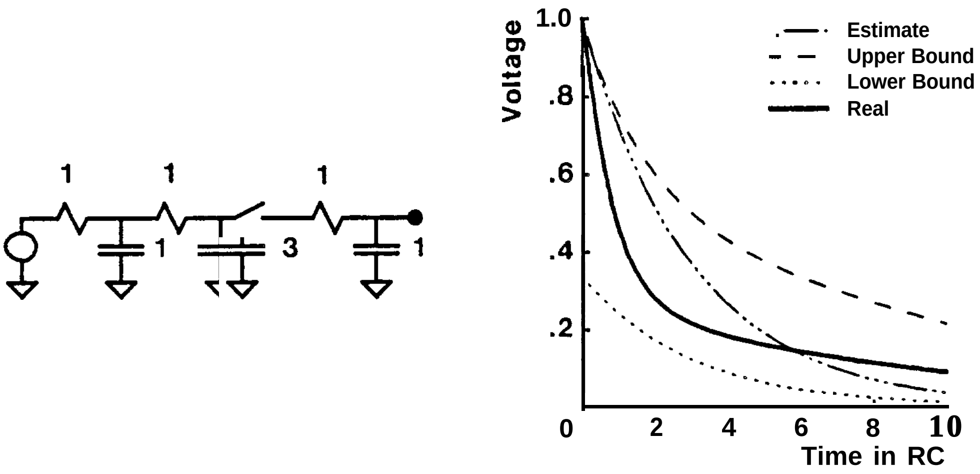


Figure 3.15 Two-tree output with a low-frequency pole-zero pair. τ_{De} = 3, τ_P = 9, τ_{Re} = 3.

3.6.2 Two-Time-Constant Model

For networks where the bounds are poor, the step in the initial voltage distribution adds **a** low-frequency zero to the output's frequency response. The zero partially cancels the dominant pole, leaving a low-frequency pole-zero pair and an output with two time constants. Physically the large capacitance in tree 1 is able to supply current during the initial transient and act like the driving source. This current shunt speeds up the output's initial transient, since the discharging capacitors in tree 2 see a smaller effective resistance to ground. During this phase of the output, the capacitors in tree 1 charge to some small, positive value. **As** the voltages in the two trees equilibrate, the output begins to track the voltage in tree 1. The slow voltage decay of the large tree 1 capacitors causes the slow tail on the output.

The estimate can be improved by following the same method used to improve the single-tree estimate (Section 3.5). Instead of modelling the output by a single time constant, a two-time-constant model is used. This model requires three time constants: τ'_{De}, τ'_{P}, and τ'_{Me}. For the network shown in Figure 3.13, the time constants are:

$$\begin{aligned}
\tau'_{De} &= \sum_{k\in 2}(\mathrm{R}_{nn_1} + R_{ke_2})C_{k_2}; \\
\tau'_P &= \sum_{k\in 1} R_{kk_1}C_{k_1} + \sum_{k\in 2}(R_{nn_1} + R_{kk_2})C_{k_2}; \\
\tau'_{Me} &= \sum_{k\in 2}(R_{ke_2} + R_{nn_1})C_k\left[1 - \frac{\tau'_{Dk}}{\tau'_P}\right] - C_{T_2}\sum_{k\in 1} R_{kn_1}\frac{R_{kn_1}C_{k_1}}{\tau'_P}.
\end{aligned} \tag{3.14}$$

Simplifying the solution provides some physical insight into this estimate. Assuming $\tau'_P \gg \tau'_{Re}$, the two time constants of the estimate become τ'_{Me}, to model the initial decay, and $\tau'_P - \tau'_{Me}$, to model the slow tail. Using this same approximation, τ'_{Me} can be simplified. Since τ'_{Dk}/τ'_P is much less than 1, the first sum in the τ'_{Me} definition is just τ'_{De}. The second sum can be written as $C_{T_2}R_{qn}$, where node q is in the middle of tree 1's dominant capacitance. Thus τ'_{Me} is equal to $\tau_{De_2} + C_{T_2}(R_{nn} - R_{qn})$: for the initial transient, tree 1 appears to be shorted to ground at node q. The amplitude of the slow tail to first order is $C_{T_2}/(C_{T_2} + C_{T_1})$, which is the simple result one would get using charge sharing. Again using a bus as an example, the capacitance of the bus acts as the driving source during a read. If the bus capacitance is large enough, the amplitude of the slow tail is small enough that it can be ignored.

3.7 Summary

When transistors are modelled as linear resistors, the model of a transistor cluster becomes a linear RC tree. A surprising and useful property of RC trees is that a single pole usually dominates the output waveform, allowing most outputs to be modelled by a simple exponential. The first moment of the output's impulse response, τ_{De}, is equal to the integral of the step response and is the time constant used in the exponential model, since for a one-time-constant system, $\tau_{De} = \tau$.

Single-time-constant bounds provide a method to check whether a single exponential is a good estimate for the output. These bounds depend on three time constants: the first moment of the output's impulse response, τ_{De}; the sum of all the open-circuit time constants in the network, τ_P; and a weighted sum of open-circuit time constants, τ_{Re}. Not only do these time constants provide bounds, they also give useful information about the output. When all three time constants are close in value, a simple exponential estimate is a good approximation to the output's wave shape. Unfortunately, the estimate may be adequate even when $\tau_{Re} \ll \tau_{De}$. An improved definition of τ_{Re}, $\hat{\tau}_{Re}$, has been presented to eliminate this problem. Using the improved time constant, the bounds are poor only when the actual output does not have a single dominant time constant. $\hat{\tau}_{Re} << \tau_{De}$ indicates the output has many closely spaced time constants; $\tau_P >> \tau_{De}$ indicates the output has a low frequency pole-zero pair.

A two-time-constant model provides a better estimate of an output when $\tau_P >> \hat{\tau}_{Re}$. This model has three parameters, which can be related to three physical time constants of the output: τ_{De}, τ_P, and τ_{Me}; the latter is derived from the second moment of the output's impulse response (the first moment of the step response). Methods to improve the bounds ensure this two-time-constant model adequatcly approximates the output's waveform.

Finally, a timing model has been derived for an output of a tree driven by another tree. This models the situation when the inputs to a pass transistor network switch. The initial estimate again uses a single-time-constant model of the output, and the time constant is set equal to the first moment of the output's impulse response, τ'_{De}. The definition of τ'_{De} for this type of network is different from τ_{De} for a single tree since the initial conditions differ. Deriving τ'_P and τ'_{Re} provides bounds for this type of network. The bounds again provide a method to determine when this simple estimate can be used. For a better estimate, τ'_{De}, τ'_P, and τ'_{Me} are determined and then used in the two-time-constant model.

Chapter 4

NONLINEAR NETWORKS

4.1 Overview

Chapter 3 has presented techniques to both estimate and bound the output waveform of a linear RC network. To apply these results to a MOS transistor cluster requires modelling transistors by linear resistors. The accuracy of the timing model depends on two factors: the accuracy of the linear model and the accuracy of the estimate. The bound on a linear network only checks the error between the estimate and the linear model. It provides no information about the relation between the estimate and the output of the actual MOS circuit; the accuracy of the linear approximation remains unchecked. To improve the timing model, this chapter describes a method to estimate and bound the output of MOS circuits using a nonlinear transistor model.

First, the problems that arise when transistors are modelled by linear resistors are discussed. Although linear models work well in many applications, there is always an underlying uncertainty about the accuracy of the result. How can a linear model accurately represent a nonlinear MOS transistor? To eliminate this concern, timing models for nonlinear circuits are derived. These models estimate and bound the output of circuits where the i-V curve of all resistors and resistor compositions have the same shape — a property of most MOS networks. With elements of this form, a simple transformation of variables converts the nonlinear problem into one that is pseudo-linear. The transformation makes it possible to use the techniques developed for linear networks on nonlinear networks.

The linear waveform estimate and bounds derivations developed in Chapter 3 are extended to handle this limited set of nonlinear networks. The resulting bounds again depend on three time constants, but the shape of the bounding waveforms is no longer exponential. The shape depends on the type of nonlinear resistor present in the circuit. For a simple quadratic model of an nMOS transistor, the output of a pass network is $t/(t + \tau)$ for the rising transient, and $1 - \tanh(t/\tau)$ for the falling transient.

The timing model is then extended so it can estimate and bound the output of a limited class of networks with two types of nonlinear resistors. This extension is important, since it provides a method for modelling the output of a pass transistor network driven by a depletion load. This simple circuit has two types of nonlinear resistors, since the i-V curve of a depletion load is different from the i-V curve of an enhancement transistor. The resulting model is surprisingly simple: the delay for the series combination of two nonlinear devices is the sum of the delays for each device considered individually.

4.2 Problems with Linear Models

Since its introduction in 1981, the linear transistor model has been used in a wide range of MOS timing analysis programs, for example [Pu82, Jo83, Ou83]. These papers report good correlation between the model predictions and more accurate simulation results. The success of these programs, while encouraging, does not mean that the model is trouble-free.

Determining the effective resistance of a MOS transistor is one of the main problems of the linear model. Most often, the value of the model resistor is chosen empirically. The delay through simple MOS circuits are simulated and then the resistance is chosen to match these delays. The effect of parameter variations can only be indirectly modelled. Each new parameter set requires a set of simulations to determine the new effective resistance. Moreover, two different values of resistance are required to match the simulator's results: one to model falling transients, and another to model rising transients. The difference between the two resistance **values** can be quite large; the rising resistor is about twice as large as the falling resistor. The difference in resistance values is outside the scope of the linear model; the model provides no information on why two different resistance values are needed.

A more fundamental problem with linear transistor models is the overall uncertainty about the accuracy of the results. Why does the linear model work? And more important, when does it fail? To answer these questions requires a method of analyzing circuit timing that does not model transistors as linear resistors.

4.3 Qualitative Analysis

For a general nonlinear circuit, estimating the output waveform is difficult. The simplifications typically used in linear network analysis — frequency-domain analysis and single-time-constant estimates — indirectly rely on superposition, **a** property that nonlinear circuits lack. Directly solving for the output waveform is even more difficult; it involves solving a set of coupled, nonlinear, differential equations. Yet, the output of MOS circuits is quite simple: MOS transistors are not general nonlinear elements.

Estimating the output of a general nonlinear circuit is difficult because the shape of the output waveform depends on the components used, and the particular configuration of the circuit. Two circuits using identical components can have very different outputs. For MOS circuits, the output shape. depends on the type of transistors used (nMOS or pMOS, enhancement or depletion mode) and the type of transient (rising or falling). It does not strongly depend on the configuration of the elements. In this sense MOS networks are similar to linear networks. Their output is usually dominated by a single slow mode, and this slow mode is the same for all

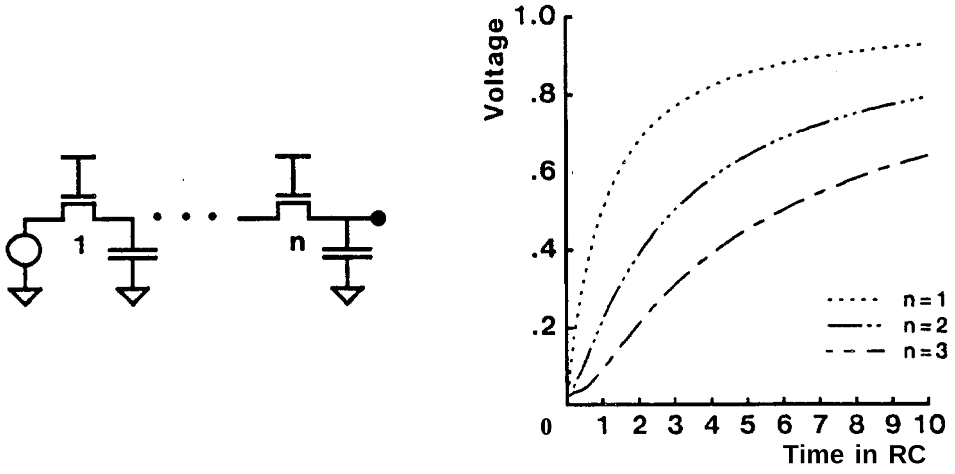


Figure 4.1 MOS pass net output waveforms.

circuits with the same type of elements. The outputs of all pass transistor networks have the same basic shape, as do the outputs of all simple logic gates; see Figure *4.1.*

MOS circuits have simple outputs because the ***i-V*** curve of all MOS transistors and transistor combinations have the same general shape. Mathematically, this means that there is a solution for the time-dependent voltages in a MOS circuit that is separable: all nodes in the circuit have the same time dependence. This solution is the slowest decaying mode in the circuit and usually dominates the output waveform. The time dependence of the slow mode depends on the circuit only through a time constant. The shape of the output is set by the shape of the common ***i-V*** curve.

Since the output of all simple logic gates have the same shape, the gate delay can be related to a simple time constant, which sets the time scale for the output waveform. That is, the output voltage can be written as $T(t/\tau)$ where T is the same for all gates and τ is proportional to the gate delay. Tau rules [MC80] make use of this time scaling to provide a first-order timing model. The rules simply relate the time scale factors for different gates, without ever determining the output waveform.

4.4 Single-Time-Constant Model

With circuits composed of voltage-dependent nonlinear resistors, where a composition of the resistors yields a resistor with the same voltage dependence, a simple transformation of variables can be used to make the nonlinear network look similar to a linear network. The same steps used to derived timing models for linear RC trees are then applicable to these nonlinear RC trees. Thus, it is possible to generate directly a single-time-constant estimate for a MOS circuit, without modelling the transistor as a linear resistor.

4.4.1 Nonlinear Circuit Transformation

The current through a voltage-dependent resistor can be written as

$$i = \frac{V_{\mathbf{max}}}{R} g(V_1, V_2),$$

where V_1 and V_2 are the normalized voltages at the two terminals of the resistor, and $g(V_1, V_2)$ is normalized to range between 0 and 1. The effective resistance, R, is equal to the maximum voltage divided by the maximum current through the resistor. In general, the series combination of two voltage dependent resistors yields a resistor with a new voltage dependence, $g^*(V_1, V_2)$, even if the original resistors are identical. For a nonlinear circuit to be transformed into a pseudo-linear circuit, the following conditions must be satisfied:

- All the resistors in the network have the same form of i-V curve: $g(V_1, V_2)$ is the same for all resistors.
- The incremental resistance of each element is positive for all possible bias conditions.
- Two series resistors can be represented by a single resistor with the same i-V curve, i.e., the same $g(V_1, V_2)$, and the effective resistance of the result

is equal to the sum of the resistances of the two components. This condition requires that the function g be of the form $g(V_1, V_2) = f(V_2) - f(V_1)$.

For circuits obeying these constraints, the current through each resistor can be written as

$$i = \frac{V_{max}}{R}[f(V_2) - f(V_1)],$$

where f is a monotonic function of V. The current depends linearly on the difference of $f(V)$. Using a transformation of variables, $U = f(V)$, the current through a resistor can be expressed as a constant times the difference in the U values at its terminals. In the i-U plane the resistors appear linear. This transformation allows a limited form of superposition to be used, and allows the techniques used in Chapter 3 to be used for this type of network as well.

4.4.2 Waveform Estimate

The following derivation provides an estimate for each output in a nonlinear RC tree; see Figure 4.2. Voltages have been normalized to range between 0 and 1. The waveform for the falling transient will be discussed; nodes are initially at 1 and decay to 0. The voltage at output node e, V_e, is determined.

The U-drop across a resistor (or series combination) can be found by summing the U-drop caused by each current source in the circuit. In analogy with Eq. (3.1), the transformed voltage at node e, U_e, can be written as

$$U_e = \sum_k \frac{R_{ke} i_k}{V_{max}} = -\sum_k R_{ke} C_k \frac{dV_k}{dt}, \tag{4.1}$$

where R_{ke} is the effective resistance from the input to the last node on the path to both nodes e and k; C_k is the capacitance at node k; and V_k is the normalized voltage at node k. This equation is not equivalent to Eq. (3.1), since U_e depends on dV/dt not dU/dt. The transformation does not remove the nonlinearity; it simply allows one to use a limited form of superposition.

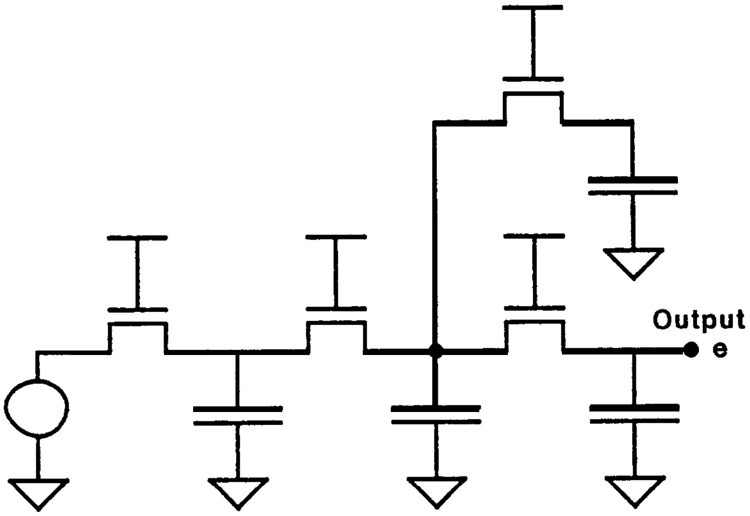


Figure 4.2 A nonlinear RC tree.

A single-time-constant estimate, V_e^*, models the output waveform of a nonlinear RC tree by the output of a single-nonlinear-resistor, single-linear-capacitor circuit. Setting dV_k/dt equal to dV_e/dt in Eq. (4.1) yields a single-time-constant estimate of the output at node e:

$$V_e^* = T(t/\tau_{De}); \qquad f(T(t)) = -\frac{dT(t)}{dt}; \qquad \tau_{De} = \sum_k R_{ke}C_k. \tag{4.2}$$

The shape of the output voltage waveform is $T(t)$, the shape of a single nonlinear RC circuit's output waveform, and the time scale factor is τ_{De}. This derivation minimizes the error in U_e not V_e, since the integral of $U_e - U_e^*$ is zero. This difference does not cause a large error because U controls dV/dt, so minimizing the error in U^* also keeps the error in V^* small. The bounds can be used ensure this error is small.

4.4.3 Waveform Bounds

To find the bounds on V_e, the same procedure used to End bounds for a linear RC tree is applied. First, Eq. (4.1) is integrated, and then V_k is bounded in terms of V_e. These bounds are used to generate bounds on the integral of U_e, which in turn are used to bound the output voltage. Defining $g_e(t)$ to be $\tau_{De} - \int U_e$, where $\tau_{De} = \sum R_{ke}C_k$, gives

$$g_e(t) = \tau_{De} - \int_0^t U_e(\tau)d\tau = \sum_k R_{ke}C_kV_k(t). \qquad (4.3)$$

The required bounds on V_k are found by first determining bounds on U_k. For networks that can be transformed, U is analogous to V in a linear network. Since the incremental resistance is always positive, once again all the voltages in the circuit will de&ease monotonically with time (for a falling transient) [Wy82]. Thus, using the analogy between V and U leads to the following bounds on U (see Appendix A):

$$\frac{R_{ke}}{R_{ee}}U_e \leq U_k \leq \frac{R_{kk}}{R_{ke}}U_e.$$

The bounds on U can be used to generate bounds on V, though there is no simple transformation that will convert the bound into the required form:

$$\alpha_{ke}V_e \leq V_k \leq \beta_{ke}V_e. \qquad (4.4)$$

Using the bounds for V_k (Eq. 4.4) in Eq. (4.3) bounds g_e:

$$\tau_{\alpha e}V_e \leq g_e \leq \tau_{\beta e}V_e, \qquad (4.5)$$

where

$$\tau_{\alpha e} = \sum_k \alpha_{ke}R_{ke}C_k; \qquad \tau_{\beta e} = \sum_k \beta_{ke}R_{ke}C_k; \qquad \tau_{De} = \sum_k R_{ke}C_k.$$

The bounding equations are single-time-constant differential equations. The time constant $\tau_{\alpha e}$ is analogous to τ_{Re} for a linear network: it is a lower bound on the output's time constant. $\tau_{\beta e}$ is analogous to τ_P: it is an upper bound on the output's time constant. The time scale for the lower bound is $\tau_{\alpha e}$, and the time scale for the upper bound is $\tau_{\beta e}$. These equations can be converted into a differential equation of the same form as a single-time-constant nonlinear circuit by applying the function f to both sides, and substituting $-dg_e/dt$ for $f(V_e)$:

$$f(g_e/\tau_{\beta e}) \leq -\frac{dg_e}{dt} \leq f(g_e/\tau_{\alpha e}). \tag{4.6}$$

The shape of the output bounds is set by the type of nonlinear resistor, and not by the circuit. The bounds on g_e can then be substituted back into Eq. (4.5) to generate bounds for V_e. Like the bounds for a linear network, these bounds can be improved by using additional constraints on U_e: $U_e \leq 1$ and U_e decreases monotonically with time.

The techniques used to improve the timing model for linear circuits relied on the linearity between i and V. These same methods can be used to improve the timing models for nonlinear circuits. Estimate and bounds for the output‘ of a nonlinear tree driven by another nonlinear tree can be derived by following the linear two-tree derivation. For circuits where $\tau_{\alpha e}$ << τ_{De}, an improved $\tau_{\alpha e}$ can be generated by using better internal voltage bounds. As for linear networks, when $\tau_{\beta e}$ >> τ_{De} the output has multiple time constants. Better estimate and bounds for these outputs are provided by a two-time-constant model. The derivation of the two-time-constant model is slightly more difficult for nonlinear circuits since frequency domain analysis no longer holds, and is described in Section 4.6.

4.4.4 Computational Requirements

The output waveform estimate and bounds depend on three time constants, $\tau_{\alpha e}$, $\tau_{\beta e}$, and τ_{De}, and the shape of the output waveform. Since the nonlinear resistor sets the output wave shape, it can be precomputed: only the time constants are calculated for each output.

As in the linear-circuit case, the time constants are simple sums over all capacitors in the network. For a network with n capacitors, each time constant takes O(n) time to compute. Unlike linear networks, time constants for different outputs will not in general share terms due to the increased complexity of the internal voltage bounds for nonlinear circuits. Thus, to find bounds for all the outputs in a nonlinear RC tree requires $O(n^2)$ time, versus O(n) time for a linear RC tree.

The estimated output is simpler to compute. Since it only depends on τ_{De}, and τ_{De} for all the outputs can be found in O(n) time, an estimate of all outputs in a nonlinear RC tree can be found in linear time.

4.5 MOS Circuits

To generate timing models for MOS circuits requires determining *f(V)* for MOS transistors. First the *i-V* curve of a MOS transistor will be discussed. Somewhat surprisingly, a simple quadratic model of a transistor is adequate to obtain a good match between the response of the MOS circuit and the model circuit. The quadratic model is used to generate bounds and an estimate of the output waveform. The estimate and bounds for an nMOS pass transistor network have a (1 — l/t) shape for the rising transient, and a $(1 - \tanh(t))$ dependence for the falling transient.

4.5.1 MOS Transistors

The current through an MOS transistor under low electric fields can be written as the difference of a quadratic function of the terminal voltages.7 Therefore the transformation of variables described in the previous section can be applied to these devices. Unfortunately, since voltage levels have remained constant while device dimensions have been scaled down, the electric field in the transistor has increased. The simple, quadratic, model of a transistor's *i-V* curve must be supplemented to

[†] Actually, the resulting function is a quadratic plus a function to the 3/2 power, but the contribution of the latter is small.[MK77]

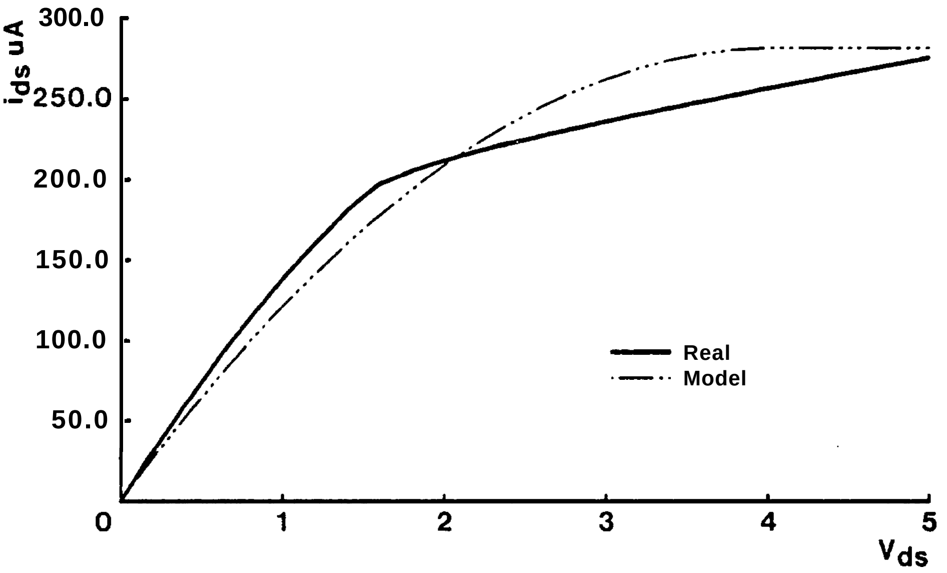


Figure 4.3 4μ nMOS transistor *i-V* curve.

account for high-field effects such as velocity saturation, mobility reduction, drain induced barrier lowering, etc. Although strictly this high-field device cannot be transformed, it can be approximated by one that can be transformed. This level of approximation is equivalent to characterizing transistors by an effective W/L ratio, rather than their W and L values.

Figure 4.3 shows a set of current-voltage curves for a real 4μ transistor and a quadratic model of that transistor. Although the high-field effects are clearly visible in the real transistor's i-V curve, a good model for the transistor can be found. The key question is not the fit of the model's *i-V* curve to the device, but rather the fit of the model's transient response to the actual output. In fact, this matching is better than the matching of the *i-V* curves. Figure 4.4 compares the output of a real transistor with the output of the quadratic model. The fit is excellent.

Using a simple quadratic model for an nMOS transistor gives

$$i = \mu C_{ox}\frac{W}{L}(V_{dd} - V_{th} - V_{ds}/2)V_{ds}.$$

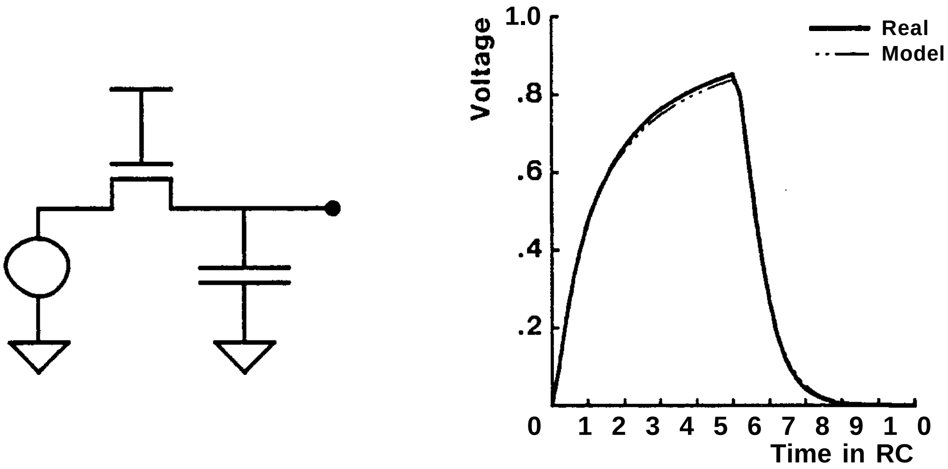


Figure 4.4 Transient output waveform for a simple pass network.

Rewriting this equation into the correct form defines *f(V)* and R_{eff}:

$$i = \frac{(V_{dd} - V_{th})}{R_{eff}}[f(V'_d) - f(V'_s)],$$

where

$$V' = \frac{V}{V_{dd} - V_{th}}, \qquad f(V') = 1 - (1 - V')^2; \qquad R_{eff} = \frac{2L}{W\mu C_{ox}(V_{dd} - V_{th})}.$$

In the following derivation, the normalized voltage will be written as *V*, rather than *V'*; and R_{ke} will represent the effective resistance of the common path to nodes *k* and e. Finally, the rising and falling waveforms are not the same in a nonlinear network, so both need to be determined.

4.5.2 Falling Transient

The output estimate, V_e^*, is found by simply substituting the correct value of τ_{De} and *f(V)* into Eq. (4.2):†

$$V_e^* = 1 - \tanh(t/\tau_{De}) = \frac{2e^{-t/\tau_{De}}}{e^{+t/\tau_{De}} + e^{-t/\tau_{De}}} \tag{4.7}$$

†To keep the equations simple, the output is assumed to be less than $V_{dd} - V_{th}$, so the transistor never enters the saturation region.

The shape of the output waveform is the same as the output of a single nMOS transistor driving a capacitor load [Cr67].

To find bounds on the output waveform, bounds on the internal node voltages must first be determined. The bounds on U_k can be approximated to give bounds on V_k. These bounds then define $\tau_{\alpha e}$ and $\tau_{\beta e}$:

$$\tau_{\alpha e} = \sum_k R_{ke}C_k\left(1 - \sqrt{1 - R_{ke}/R_{ee}}\right); \qquad \tau_{\beta e} = \sum_k \frac{R_{ke}C_k}{\left(1 - \sqrt{1 - R_{ke}/R_{kk}}\right)} \tag{4.8}$$

Using these time constants, the bounds on the output become:[†]

$$V_e(t) \le \begin{cases} 1, & t \le \tau_{De} - \tau_{\alpha e} \\ \left(1 + \frac{\tau_{\alpha e}}{2t}\right)\left(1 - \sqrt{1 - \frac{4t\tau_{De}}{(2t + \tau_{\alpha e})^2}}\right), & \tau_{De} - \tau_{\alpha e} \le t \le \tau_{\beta e}\left[\frac{\tau_{\beta e} - \tau_{\alpha e}}{2\tau_{\beta e} - \tau_{De}}\right] \\ \left[1 - \tanh\left(\frac{t}{\tau_{\beta e}} - \frac{\tau_{\beta e} - \tau_{\alpha e}}{2\tau_{\beta e} - \tau_{De}} + \frac{1}{2}\ln\left(\frac{2\tau_{\beta e} - \tau_{De}}{\tau_{De}}\right)\right)\right], & t \ge \tau_{\beta e}\left[\frac{\tau_{De} - \tau_{\alpha e}}{2\tau_{\beta e} - \tau_{De}}\right] \end{cases} \tag{4.9}$$

$$V_e(t) \ge \begin{cases} \frac{(\tau_{De} - t)}{\tau_{\beta e}}, & t \le \tau_{De} - \tau_{\alpha e}; \\ \frac{\tau_{\alpha e}}{\tau_{\beta e}}\left[1 - \tanh\left(\frac{t - \tau_{De} + \tau_{\alpha e}}{\tau_{ae}}\right)\right], & t \ge \tau_{De} - \tau_{\alpha e}. \end{cases} \tag{4.10}$$

4.5.3 Rising Transient

To determine an estimate and bounds for an output's rising transient, $1 - U_e$ replaces U_e, and $1 - V_e$ replaces V_e in equations (4.1) through (4.5). Thus, the output of a nMOS transistor tree is estimated by

$$1 - U_e^* = (1 - V_e^*)^2 = -\tau_{De}\frac{d(1 - V_e^*)}{dt},$$

[†] Appendix D derives bounds for MOS networks in more detail.

or solving for V_e^*,

$$V_e^* = \frac{t}{t + \tau_{De}}. \tag{4.11}$$

The shape of the output estimate is the same as the shape of a rising transient of a single nMOS transistor driving a capacitor load [Cr67].

To generate bounds on the output voltage, bounds on the internal voltages must first be found. Since the square root of $1 - U$ is $1 - V$, the bounds on $1 - V$ are just the square roots of the bounds on $1 - U$. The two bounding time constants are

$$\tau_{\alpha e} = \sum_k \frac{R_{ke}^{\frac{3}{2}} C_k}{R_{ee}^{\frac{1}{2}}}; \qquad \tau_{\beta e} = \sum_k \sqrt{R_{kk} R_{ke}} C_k. \tag{4.12}$$

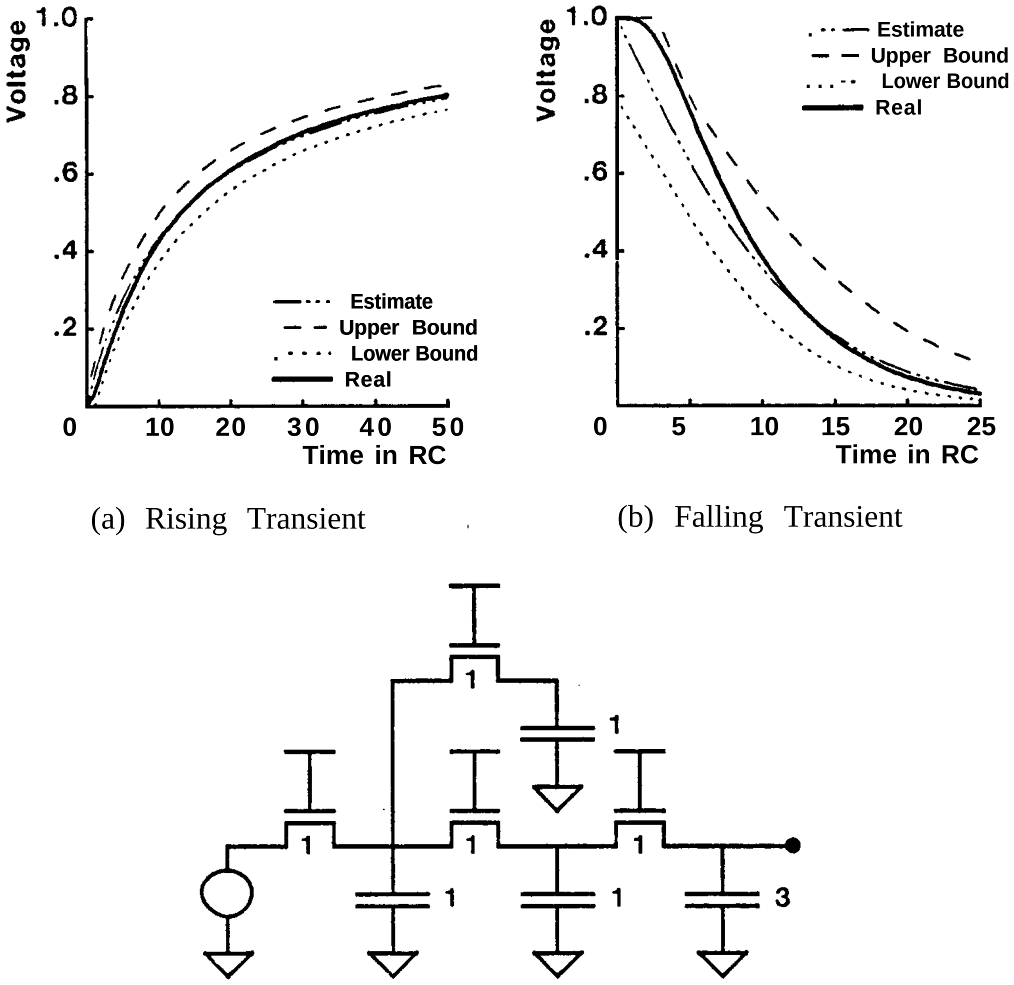
The resulting bounds on the output voltage are[†]

$$V_e(t) \geq \begin{cases} 0, & t \leq \tau_{De} - \tau_{\alpha e}; \\ 1 - \frac{\tau_{\alpha e}}{2t}\left(\sqrt{1 + 4t\tau_{De}/\tau_{\alpha e}^2} - 1\right), & \tau_{De} - \tau_{\alpha e} \leq t \leq \frac{(\tau_{\beta e} - \tau_{\alpha e})\tau_{\beta e}}{\tau_{De}}; \\ 1 - \frac{2\tau_{\beta e} - \tau_{\alpha e}}{t + \tau_{\beta e}^2/\tau_{De}}, & \frac{(\tau_{\beta e} - \tau_{\alpha e})\tau_{\beta e}}{\tau_{De}} \leq t. \end{cases} \tag{4.13}$$

$$V_e(t) \leq \begin{cases} 1 - \frac{\tau_{De} - t}{\tau_{\beta e}}, & t \leq \tau_{De} - \tau_{\alpha e}; \\ 1 - \frac{\tau_{\alpha e}^2}{\tau_{\beta e}(t - \tau_{De} + 2\tau_{\alpha e})}, & t \geq \tau_{De} - \tau_{\alpha e}. \end{cases} \tag{4.14}$$

Figures 4.5a and b show the estimated output waveform, the bounding waveform, and the results from a simulation of the circuit shown in Figure 4.5c for both

[†] Appendix D derives these bounds in more detail.



(a) Rising Transient

(b) Falling Transient

(c) Circuit

Figure 4.5 Pass transistor estimate and bounds.

the rising and falling edges. By using a nonlinear model of MOS transistors, both the rising and falling waveforms are accurately modelled.

4.5.4 Comparison with Timing Models for Linear Networks

The timing models for MOS circuits closely resemble linear timing models. This similarity helps to explain why the linear models work well. Both models generate a single-time-constant estimate, which depends on a single time scale factor, τ_{De}.

The form of the scale factor is the same, but the values may differ because the resistance of the linear model may be different than the value used in the nonlinear timing model. The crucial difference in the timing models is the shape of the output waveforms. For nonlinear resistors, the outputs are no longer exponentials. Instead the output waveform has a $T(t)$ dependence, where $T(t)$ is the output of a single nonlinear RC circuit. For an nMOS pass network, $T(t)$ corresponds to a $1 - 1/t$ rising waveform, and a $1 - \tanh(t)$ falling waveform.

This difference in the output wave shape points out the major limitation of linear circuit models: a linear waveform estimate cannot match the actual output over the entire transient. However, the resistance of the linear model can be chosen so the linear model predicts the correct delay for a particular switching point. The delay to reach voltage V_o is

$$t = -\tau_{De} \ln(V_o)$$

for a linear model, and

$$t = \tau_{De} \tanh^{-1}(1 - V_o)$$

for a nonlinear model of a MOS transistor (falling transient). The two times can be made equal by making τ_{De} in the linear model smaller than τ_{De} in the nonlinear model. Since τ_{De} for a circuit is proportional to the resistance of the transistors, the value of τ_{De} is changed by making the effective linear resistance of each transistor less than the resistance used in the nonlinear derivation. To match the delay for the rising transient, the effective resistance must be made larger than the resistance used in the nonlinear model, since $1 - 1/t$ is slower than $1 - \exp(-t)$. Thus two different linear resistance values are required to model an nMOS transistor, since the shape of the rising and falling waveforms are different.

4.6 Two-Time-Constant Model

Since most outputs are dominated by a single slow mode, the single-time-constant estimate is usually a good approximation to the output waveform. The bounds check whether the circuit being modelled has a single dominant time constant. When all three time constants have a similar value, the bounds will be close to the estimate, and the estimate accurately models the output waveform. When $\tau_{\beta e} >> \tau_{De}$, the bounds are poor because the output has multiple time constants. The estimate of these output waveforms is improved by using a two-time-constant model.

It is possible to improve the bounds as well, which provides tighter error bounds on the estimate. The bounds improvement follow the methods used to improved the linear timing models. Since these have already been discussed in detail (see Appendix C), the nonlinear results only will be briefly reviewed. The improved estimate will be discussed in more detail to try and illustrate the differences between linear and nonlinear circuits. The following derivation is for falling transients in an nMOS circuit. Timing models for rising transients can be improved using a analogous method.

4.6.1 Waveform Estimate

It was possible to use frequency-domain analysis to provide a two-time-constant timing model for outputs of linear circuits. This technique does not have a direct extension for nonlinear circuits. However, it is possible to generate an improved model of the output voltage that is similar to the simplified, improved estimate for linear circuits.

The problem with the initial estimate is that all nodes are forced to decay at the same rate: all dV_k/dt terms are set equal. A two-time-constant output is generated by partitioning the nodes into two groups, and only assuming nodes in a group to decay at the same rate. Nodes in group 1 are allowed to decay rapidly compared to nodes in group 2. For circuits with two time constants, the distribution of τ_{Dk} is bimodal. Therefore, the choice of dividing line between group 1 and 2 is not critical. Using the mean of the output time constant and the largest time constant in the circuit as the break point, all nodes where $\tau_{Dk} \leq \frac{\tau_{De}+\max(\tau_{Dk})}{2}$ are placed in group 1; all other nodes are placed in group 2. Three time constants characterize the output waveform:

$$\tau_{De_1} = \sum_{k\in 1} R_{ke}C_k; \qquad \tau_{P_2} = \sum_{k\in 2} R_{kk}C_k; \qquad \tau_{De_2} = \sum_{k\in 2} R_{ke}C_k.$$

The transformed output voltage can then be represented as the sum of the contributions of the slow group 2 nodes, U_{e_2}, plus the contributions of the faster group 1 nodes, U_{e_1}. The slow nodes will decay at a rate approximately equal to τ_{P_2}, so their contribution to the output can be estimated by[†]

$$U_{e_2} = -\sum_{k\in 2} R_{ke}C_k\frac{dV_k}{dt} \approx \frac{\tau_{De_2}}{\tau_{P_2}}\operatorname{sech}^2(t/\tau_{P_2}).$$

The contribution of the faster nodes is found by assuming all these nodes decay at the same rate as the output:

$$U_{e_1} \approx -\sum_{k\in 1} R_{ke}C_k\frac{dV_e}{dt} \quad = \quad -\tau_{De_1}\frac{dV_e}{dt}. \tag{4.15}$$

To solve this equation, U_{e_1} must be related to U_e. For outputs with two time *Constants,* $\tau_{P_2} >> \tau_{De_1}$, *so* U_{e_2} will be roughly constant during the decay of U_{e_1}. Therefore U_{e_1} can be approximated by $U_e - \tau_{De_2}/\tau_{P_2}$. It is important to realize that Eq. (4.15)isd ifferent than the equation for a single-transistor driving a capacitance load. The relationship between U_e and V_e depends on their values: the presence of

[†]For the falling transient, $V = 1 - \tanh(t)$. dV/dt is proportional to U, which is $\operatorname{sech}^2(t)$.

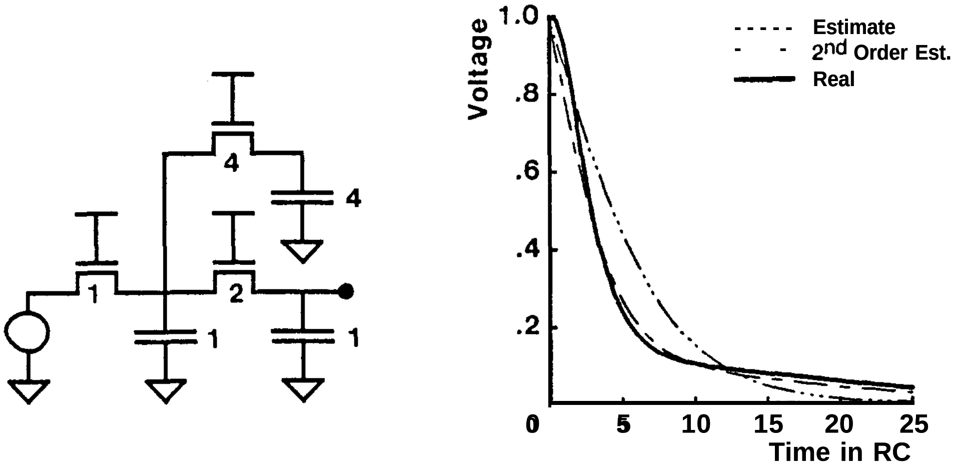


Figure 4.6 Two-time-constant output waveform.

a dc voltage at the nodes changes the nature of the transient. Solving Eq. (4.15) yields an estimate of U_{e_1}:

$$U^*_{e_1} = \left(1 - \frac{\tau_{De_2}}{\tau_{P_2}}\right)\mathrm{sech}^2(\alpha t/\tau_{De_1}); \qquad \alpha = \sqrt{\left(1 - \frac{\tau_{De_2}}{\tau_{P_2}}\right)}.$$

For the falling edge, the effect of the dc voltage is minor: the shape of the output remains unchanged; only the time constant is changed.+ Combining the two contributions to U yields the two-time-constant estimate:

$$U^*_e = \left(1 - \frac{\tau_{De_2}}{\tau_{P_2}}\right)\mathrm{sech}^2(\alpha t/\tau_{De_1})\ \text{i- ?}\ _{2}\ \mathrm{sech}^2(t/\tau_{P_2}) \tag{4.16}$$

where

$$V^*_e = 1 - \sqrt{1 - U^*_e}.$$

Figure 4.6 shows a two-time-constant output along with the original estimate and the improved estimate. The improved estimate is a much better model of the output waveform.

†For the rising edge, the shape of the faster transient changes form. The slow mode is **1** — l/t while the faster mode is coth(t).

4.6.2 Bounds Improvement

For outputs where $\tau_{\alpha e} << \tau_{De}$, the bounds can be improved by using better internal voltage bounds to generate $\tau_{\alpha e}$. For nonlinear circuits, the bound improvement is a two-step procedure: first bound U_k in terms of U_e, and then from this bound generate bounds on the actual voltage. A better lower bound on U_k can be determined using a method analogous to the linear circuit technique described in Section 3.4.5. The improved bound on U improves the bounds on V, which leads to an improved $\tau_{\alpha e}$.

For outputs where $\tau_{\beta e} >> \tau_{De}$, tighter bounds can be generated by following the linear two-time-constant technique. The improvement of the lower bound is identical to the method described in Section 3.5.3. This method simply uses the constraint that $V_k \leq 1$ to improve the upper bound on the internal voltages used to generate $\tau_{\beta e}$. The result is a set of time constants that provide a better lower bound on the voltage.

Generating a better upper bound proves to be more difficult. Again the same basic approach used to improve the linear bound can be applied to the nonlinear network: a voltage is added to isolate the slow nodes from the output. By adding the voltage source, the transformed output voltage can be written as the sum of two terms. The voltage source causes one term, and the capacitor currents cause the other. The effect the voltage source has on the output is easily determined; the effect of the capacitor currents is more difficult to ascertain.

Like the fast transient in the improved nonlinear estimate, the output waveform depends on the voltages present in the circuit (which result from the voltage source). For a linear network these voltages are not an issue: by superposition the sources can be considered one at a time. These dc voltages also affect the bounds that relate the maximum change in V_k to the change in V_e. The bounds on the transformed voltage at each node are identical to the bound on the voltages for a linear network:

$$\frac{R'_{ke}}{R'_{ee}}U_k \leq U_e \leq \frac{R'_{kk}}{R'_{ke}}U_k.$$

Unfortunately the relation between U and V depends on the voltage. Bounds on the voltages can still be obtained, but the dc voltages present at the node need to be taken into account. Once a bound for the transient is obtained, the bound on the transformed output voltage is the sum of the bounds on its two components. The inverse transform of this sum yields an improved bound on the output.

4.7 Mixed Nonlinear Elements

One limitation of the timing models that have been presented in this chapter is the requirement that all resistors in the circuit must have the same form of *i*-*V* curve. To generate a timing model for the rising output of an nMOS pass transistor network driven by a gate (see Figure 4.7), either the depletion transistor must be modelled as a pass transistor or the pass transistors must be modelled as depletion transistors. This approximation causes a problem when the dominant resistance in the circuit changes with time. For example, in Figure 4.7, if the resistance of the depletion load is small compared to the pass transistor resistance, then the depletion transistor does not have a large effect on the output waveform. A crude model of this transistor can be used (for example, modelling it as a short) without causing a large error in the output estimate.

A problem arises when the resistance of the depletion transistor is initially large compared to the pass transistor's resistance. When the output is low, the dominant resistor is the depletion transistor. As the output rises, the resistance of the pass transistors increases and they eventually control the output waveform. The shape of the output cannot be modelled by either thc depletion transistor or the pass transistors alone. Each device affects a portion of the output. Timing models for this type of circuit are generated by extending the basic timing model to handle circuits where a resistor of one form drives an RC tree composed of resistors

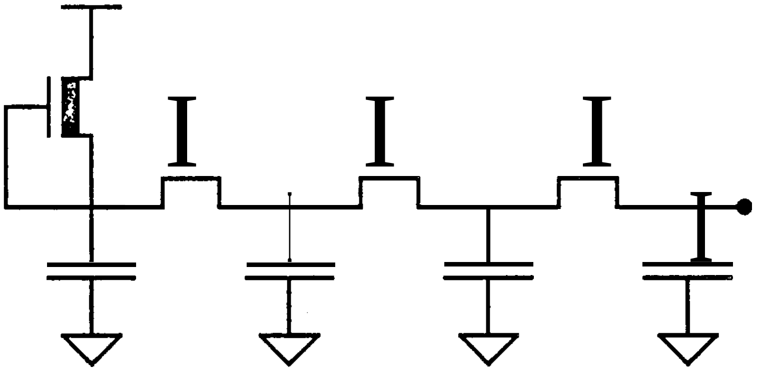


Figure 4.7 Circuit for a rising transient in nMOS.

of another form. The resulting waveform estimate is quite simple. The total delay is the sum of the delays obtained by considering each type of resistor separately.

4.7.1 Waveform Estimate

Figure 4.8 shows a model of a mixed nonlinear tree. R_1 is the first resistor in the tree, and has a different type of *i-V* curve than all the other resistors in the tree. The current through R_1 is proportional to the function f_1 applied to its terminal voltages, and the current through each of the other resistors is proportional to the function f_2 applied to its terminal voltages. R_{ke} is the effective resistance of the path through the RC tree common to nodes e and *k*, but does not include the resistance of R_1. For example in figure 4.8, $R_{3\ 4}$ is $R_2 + R_3$. Voltages have been normalized to range between 0 and 1, and all nodes in the circuit are initially 1. The output, node e, may be any node in the RC tree.

The transformed output voltage can be written as the drop in the RC tree plus the drop across the initial resistor (R_1):

$$f_2(V_e) = f_2(V_1) - \sum_k R_{ke} C_k \frac{dV_k}{dt}. \tag{4.17}$$

The U-drop across the first resistor can be found by summing the currents that flow through it:

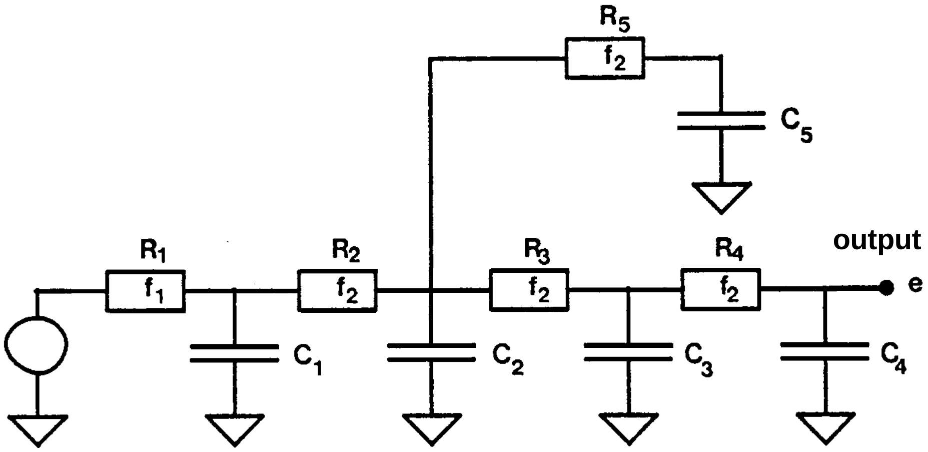


Figure 4.8 A mixed nonlinear circuit.

$$f_1(V_1) = -R_1 \sum_k C_k \frac{dV_k}{dt}. \tag{4.18}$$

Substituting Eq. (4.18) in Eq. (4.17) gives

$$f_2(V_e) = -R_1 \frac{f_2(V_1)}{f_1(V_1)} \sum_k C_k \frac{dV_k}{dt} - \sum_k R_{ke} C_k \frac{dV_k}{dt}. \tag{4.19}$$

To generate a single-time-constant estimate, dV_k/dt must be approximated by a term proportional to dV_e/dt. Setting V_k equal to V_e in Eq. (4.19) and rearranging terms yields an estimate of the output waveform, V_e^* :

$$\text{At} = -\tau_1 \int_1^{V_e^*} \frac{dV_e}{f_1(V_e)} - \tau_{De} \int_1^{V_e^*} \frac{dV_e}{f_2(V_e)}, \tag{4.20}$$

where

$$\tau_1 = \sum_k R_1 C_k; \qquad \tau_{De} = \sum_k R_{ke} C_k.$$

The estimated delay is the sum of two terms. The first term is the delay caused by R_1, assuming all the other resistors are replaced by a short circuit; the second term is the delay through the RC tree assumirg the first resistor is replaced by a short circuit.

The integral of the difference between the transformed (using the function f_2) output voltage and the estimate is

$$\int f_2(V_e) - \int f_2(V_e^*) = \sum_k R_1 C_k \int_0^1 \left[\frac{f_2(V_1)}{f_1(V_1)} - \frac{f_2(V_k)}{f_1(V_k)}\right] dV_k$$

$$+ \sum_k R_{ke} C_k \left[\int_0^1 dV_k - \int_0^1 dV_e\right]$$

The difference in the second sum is zero since V_e and V_k start and end at the same voltages. The difference in the first sum is harder to quantify. The two terms in the sum will cancel only when V_1 and V_k are similar in value. However, the contribution of R_1 to the integral of the transformed output is only significant when R_1's resistance is large compared to the resistance of the RC tree. In this case, most of the voltage is dropped across R_1 and therefore V_1 and V_k will be roughly equal. When the resistance of R_1 is small compared to the resistance of the RC tree, the difference in V_1 and V_k can be large, but the percentage error will still be small, since the first term is a small fraction of the total output voltage.

As an example, consider a set of pass transistors driven by a linear resistor as shown in Figure 4.9. The waveform estimate for this circuit is

$$At = -\tau_1 \ln(1 - V_e^*) + \tau_{De} \frac{1}{1 - v:},$$

The response is initially dominated by the resistor, but for large voltages the pass transistors dominate. Since the estimate models both regions, it is a good approximation of the actual output.

4.7.2 Bounds

To bound the output waveform, Eq. (4.19) must be integrated, since bounds for the differential voltages are not known. This integral cannot be evaluated directly since the relationship between V_k and V_1 is not known. Instead, a bound for V_1 can

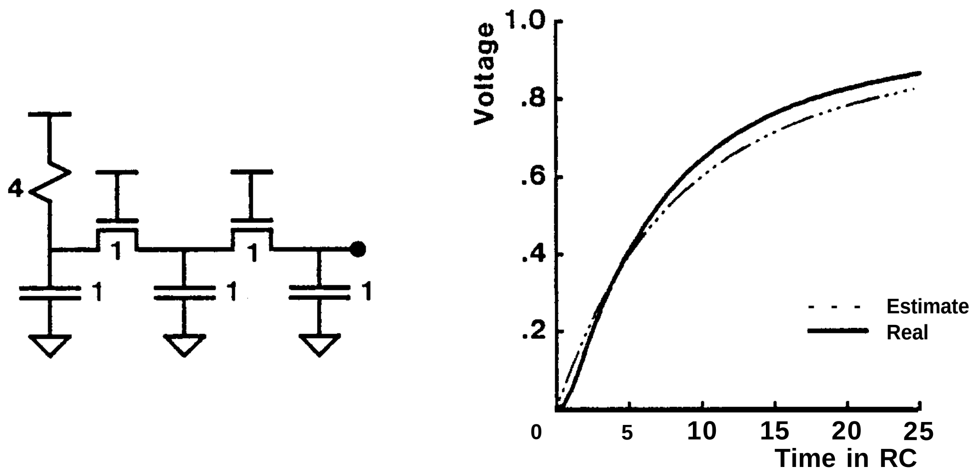


Figure 4.9 Output for a mixed nonlinear circuit.

be used to generate a bound on the integral. This bounded integral can then be used to generate bounds on the output waveform. These waveforms bound both the single-time-constant error and the error resulting from approximating the value of V_1 in Eq. **4.19** by V_e. Simply finding $\tau_{\alpha e}$ and $\tau_{\beta e}$ for the nonlinear tree is sufficient to check the single-time-constant approximation: the presence of R_1 (a series resistor driving the tree) only makes these time constants closer to each other and to τ_{De}. If the bounds for the tree alone (assuming R_1 is 0) are good, then the single-time-constant estimate will be a good approximation for the output.

4.8 Summary

Timing models based on linear RC trees have one fundamental drawback: the uncertainty in the error caused by linearizing the transistors. To eliminate this type of error, timing models have been derived for MOS resistor trees. Although the output of general nonlinear circuits is difficult to estimate, the output of a MOS circuit can be'modelled more easily, since all transistors and transistor combinations have roughly the same shape of *i*-*V* curve.

For circuits where the *i-V* curves of all resistor and resistor combinations have the same shape, a simple transformation of variables converts the nonlinear circuit into a pseudo-linear circuit. The techniques used to derive an estimate and bounds for linear networks can be applied to the transformed nonlinear network. Like the linear models, the resulting timing models depend on three simple time constants.

The timing model can be applied to MOS circuits by using a simple quadratic model for a MOS transistor. The crucial difference between the nonlinear timing model and the linear model is in the output waveform. An output estimate using a linear transistor model is exponential, whereas an output estimate using a quadratic transistor model has a $(1 - l/t)$ dependence for a rising transient, and a $1 - \tanh(t)$ dependence for a falling transient. Linear models minimize the timing error caused by the incorrect waveform by allowing a transistor's effective resistance to depend on the type of transient.

Outputs that are not dominated by a single slow mode have poor bounds. A two-time-constant model provides a better estimate of these outputs. The key difference between the nonlinear and linear improved estimate is that the faster decaying component of the output in a nonlinear circuit may have a different time dependence than the slow mode. In a linear circuit both components to the output waveform are exponentials.

The timing model was extended to handle circuits that contain two different types of nonlinear resistors; the delay through a mixed circuit is approximately the sum of the delays obtained by looking at each type of resistor individually. For a depletion load driving a pass network, the delay is the sum of the time it takes the depletion transistor to charge the total capacitance plus the delay through the pass transistor network.

Chapter 5

SLOW INPUTS

5.1 Introduction

Chapter 3 has presented a method for generating timing models for linear RC circuits. To apply this model to MOS transistor clusters, two approximations must be made: transistors must be modelled as linear resistors and the inputs to transistor clusters must be modelled as step waveforms. Chapter 4 has eliminated the need for the first approximation by presenting a method to derive timing models for nonlinear networks. The goal of this chapter is to eliminate the need for the second approximation: to generate an estimate of the output waveform for an arbitrary input waveform. The need for this extension is shown in Figure 5.1; the shape of the output waveform clearly depends on the shape of the input driving the gate. Approximating all inputs by a step waveform will significantly underestimate the gate delay for slow inputs.

For step inputs, a very simple model of a transistor can be used; the transistor is either an open circuit or a fixed nonlinear resistor. When the input changes gradually, the states between on and off must be modelled. Logic-gate drive curves are one way to represent this information. Drive curves depict a contour map of the gate output current versus input and output voltage. The shapes of all drive curves are roughly the same and lead to a very simple circuit model for a gate. This gate model reduces to the model used in Chapters 3 and 4, a resistor in series with a switch, for fast inputs.

Using the gate model: the output waveform of a transistor cluster can be determined as a function of the input waveform. The output of a simple gate is

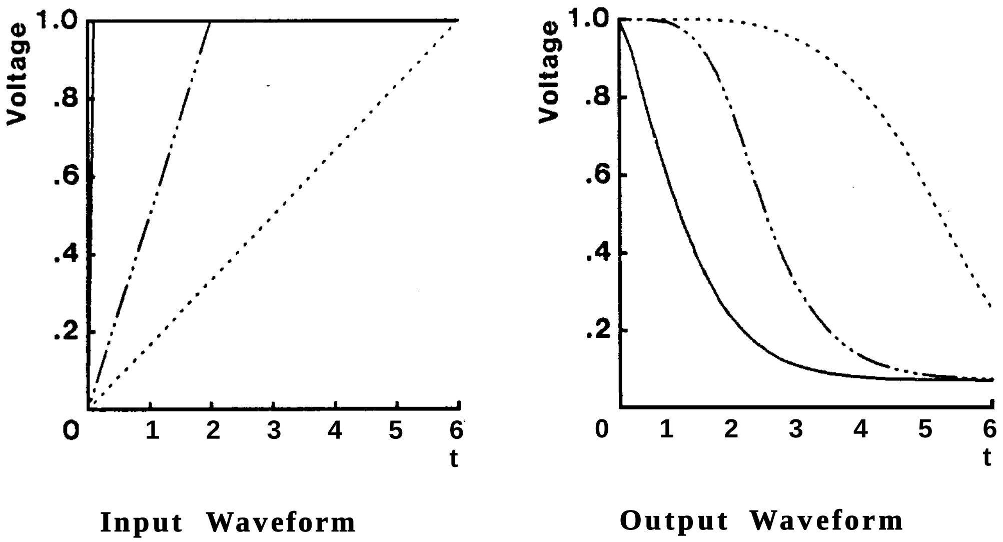


Figure 5.1 Gate output waveform for different input waveforms.

derived first, since this circuit only has a single time constant. The results show that the output waveform is not very sensitive to the shape of the input: a simple ramp model of the input is sufficient to determine the output waveform.

Next, the output of a general transistor cluster is estimated. For these circuits, the output network is an RC tree, and the output cannot be determined exactly. However, the single-time-constant approximation converts a complex transistor cluster into a simple gate, which allows the output waveform to be estimated. To insure the single-time-constant approximation is valid, single-time-constant bounds are also derived. The bounds are worst for quickly changing inputs and improve as the input becomes slower. Thus, any output with good bounds for a step input will have even better bounds when driven with a slower input.

5.2 Gate Models

Before a gate model can be constructed, the important characteristics of a gate must be determined. A gate's transfer curve, which shows the dc output voltage for any input voltage, is only part of the information needed. The time behavior of the output voltage depends on the gate output current, since the current is what drives the output to its new state. To find the output waveform, the output current must be known as a function of input and output voltages. Drive curves are one way to represent this information.

5.2.1 Drive Curves

The drive curves for a gate give a contour map of the gate output current versus input and output voltages. Instead of plotting the input-output pairs where the output current is zero, which gives the transfer curve, the output voltage vs. input voltage is plotted for different output currents. The transfer curve is only one curve in a set of curves.

Figure 5.2 shows the drive curves for CMOS, nMOS and bipolar gates. The transfer curves for the three gates are similar. Comparing the drive curves shows a more striking difference. The bipolar gate's drive curves are spaced closer together than those for the MOS gates, indicating a higher current gain. The output waveform of a bipolar gate will not depend strongly on the input waveform since the output current only depends on the input over a small range of input voltage. For both nMOS and CMOS gates, the input can affect the output current over the entire input voltage range. Consequently these gate types are more sensitive to the shape of the input waveform than the bipolar gate.

The drive curves for all gates have a high gain region (magnitude of the slope >> 1) and a low gain region, giving the curves an 'L' or inverted 'L' shape. This shape is fundamental to all simple, level restoring logic gates. If a gate is able to restore a degraded input signal, then the transfer curve of the gate must have a gain much less than one near the dc output levels. These low-gain regions force the remaining region to have a gain much larger than one because a gate's input and output voltage ranges are equal. The constraints on the transfer curve fix the shape

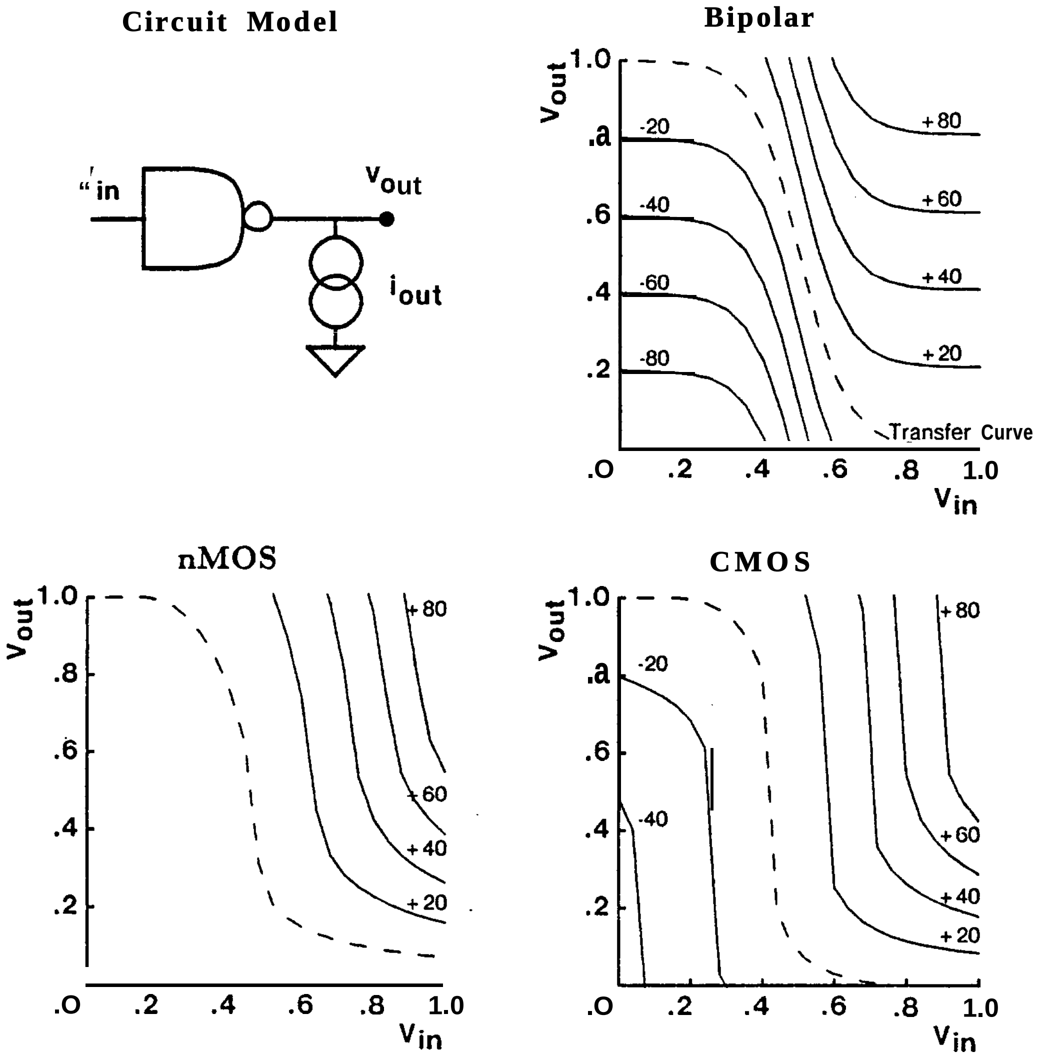


Figure 5.2 Drive Curves for different types of gates.

of all the drive curves, since a drive curve is just the transfer curve under a current load.

5.2.2 Circuit Model

A simple gate model is generated by idealizing the actual drive curves; see Figure 5.3. The slope of the drive curves is set to zero in the low-gain region, and is set to infinity in the high gain region. Thus, at any time, the output current is

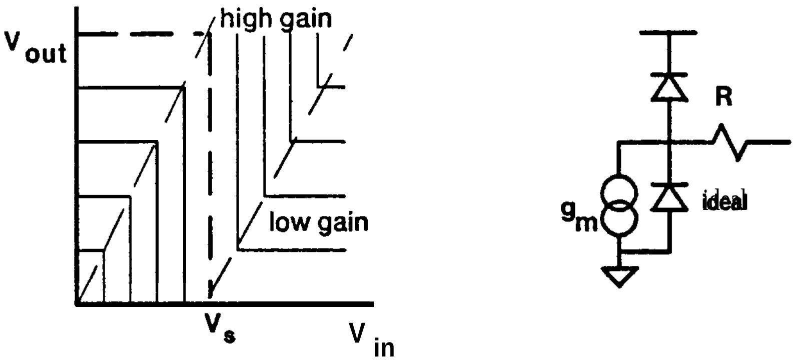
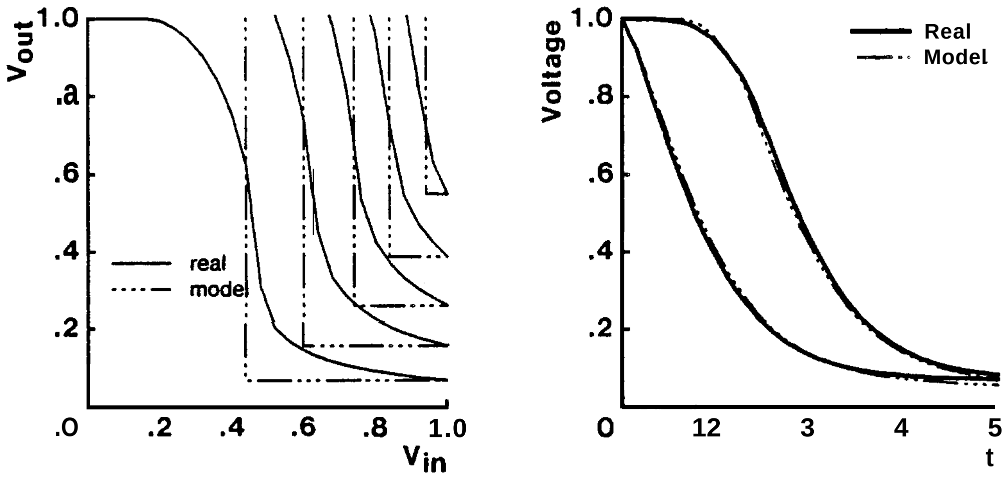


Figure 5.3 Idealized drive curve and resulting gate model.

controlled by either the input or the output voltage, but never both. The circuit model for a gate becomes a voltage-controlled current source when the gate is in the high-gain region, and a resistor, when the gate is in the low-gain region. The low-gain region is entered when the resistor current is lower than the output of the current source. Both regions can be modelled by a resistor in series with a current source, and two ideal diodes. The ideal diodes model the transitions from the high-gain region to the low-gain regions. The current source pulls the output toward ground when the input is greater than V_s, the switching voltage of the gate, and it pulls the output high when the input is lower than V_s.

For a falling transient, the output remains clamped to the high output level until the input rises to V,. After the input reaches V_s, the output current changes sign, and the top clamp diode turns off. The output enters the high-gain region and the output current is controlled by the current source. When the output of the current source becomes larger than V_{out}/R, the bottom diode shorts out the current source and the current is controlled by the resistor. This circuit model is not of necessity piece-wise linear: the current source and/or the resistor may be nonlinear. This model of a gate makes it possible to determine the output for an arbitrary input waveform.



(a) Drive Curves (b) Output Waveforms

Figure 5.4 An nMOS gate model.

By idealizing the drive curves, the effect of the output voltage on the output current is ignored in the high-gain region. This approximation is valid when the input and output time constants are comparable. For the same change in voltage, the output's effect on the output current is A,, times smaller than the input's effect, where *A,,* is the slope of the drive curves. Ignoring the output's contribution completely does not cause a large error.[†] The drive curves for a nMOS gate along the model drive curves are shown in Figure 5.4a; the output waveforms for two different inputs are shown in Figure 5.4b. The fit to the output waveform is excellent, indicating the approximations made to generate the gate model do not cause a large error in the output estimate.

Using this model, a gate is characterized by four parameters: the gate's switching voltage, V_s; the gate's forward transconductance in the high-gain region, g_m;

[†] For very slow inputs, this approximation breaks down. The output voltage now changes much faster than the input, and its contribution to the current can no longer be ignored. Although the error in the gate delay increases as the input slows down, if the error is measured relative to the total delay, including the delay of the gate driving the input, then the percentage error actually decreases for very slow inputs. Stated simply, the input delay swamps the total.

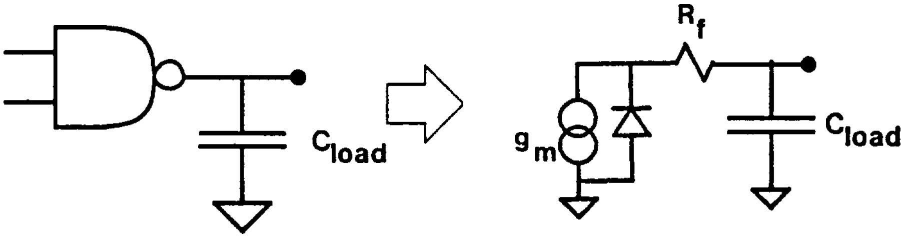


Figure 5.5 A simple gate.

and the output resistance in the low-gain region for the rising (falling) waveform, R_r (R_f). In general both g_m and R_r (R_f) can be nonlinear.

Finally, this simple gate model reduces to the gate model used in Chapters 3 and 4 for a step input. When a gate input changes from a 0 to a 1, the output current instantly changes to its maximum value. This maximum current is greater than or equal to the largest current through the output resistance, so the gate immediately enters the low-gain region of the drive curve. The gate model becomes a simple (nonlinear) resistor, R_f, which is the model used in the previous chapters.

5.3 Simple Gates

A simple gate is one where its output network is a single capacitor; see Figure 5.5. These gates have only a single time constant, and their output for an arbitrary input can be found exactly. The model shows that the output waveform is only weakly coupled to the shape of the input waveform. A qualitative description of the output is given first, followed by a quantitative description of a linear-gate output. Using the linear gate as a guide, the output waveform of a nonlinear gate is derived. The nonlinear model is then applied to MOS gates, and their characteristics are discussed.

5.3.1 Qualitative Analysis

For a rising input, the output remains high until the input reaches the switching voltage of that gate. The initial shape of the input waveform has no affect on the output. After the switching point is reached, the input voltage controls the output current. The output voltage in this region is 1 minus the integral of the output current. The output current, and therefore the slope of the output voltage, increases monotonically until the output enters the low-gain region of the gate. At this point, the output voltage again becomes independent of the input, and decays to 0 with a time constant determined by the gate's resistance (R_f) and the output capacitance (C_{load}). The shape of the initial and final sections of the input waveform do not have any affect on the shape of the output. This decoupling means that a minimum of information about the input waveform is needed to determine the output of a gate. Usually, the slope of the input when it crosses the gate's switching voltage is sufficient.

5.3.2 Linear Gates

Linear gates are the simplest to analyze, since both the output resistance and the voltage-controlled current source are linear. First, the output waveform for a linear gate driven by a ramp input is derived. From this analysis, the gate delay can be found as an explicit function of the input slope and the gate parameters. To determine if a ramp can be used to model the actual exponential input waveforms, the output waveform for an exponential input is determined. The resulting output is quite similar to the output for a ramp input, as long as the slopes of the two input waveforms are the same at the switching voltage of the gate.

In the following derivation, the input and output voltages have been normalized to range between 0 and 1, and the switching point of the gate is V_s. Since the rising and falling transients are similar, only thc falling waveform is discussed. The input transconductance has been normalized by the output resistance to yield the factor $\beta = (g_m R_f)^{-1}$. The value of β ranges between 0 and $1 - V_s$, and represents the range of input voltages where the output current is determined by the input. The quantity $V_s + \beta$ is the highest input voltage that can affect the output current. Gates with smaller β have higher current gain, and are less sensitive to the input waveform than gates with larger β. Finally, the intrinsic time constant of the gate is $\tau_f = R_f C_{load}$. For a step input, the output is a decaying exponential with a time constant τ_f.

A ramp input will be represented by

$$V_{in} = V_s + \frac{t}{\alpha\tau_f},$$

so the input is equal the switching point of the gate at $t = 0$. The slope of the input is controlled by α; it takes $\alpha\tau_f$ for the input to rise from 0 to 1. For this input voltage, the output current is a ramp, and the resulting output voltage is

$$V_{out} = \frac{t^2}{2\alpha\beta\tau_f^2}. \tag{5.1}$$

The output voltage is given by Eq. 5.1 until the output enters the low-gain region of the drive curves. This transition occurs when the voltage drop across the series resistor, $i_{out}R_f$, is equal to V_{out}, and the time when this occurs will be called t_s:

$$t_s = \tau_f\left[\sqrt{1 + 2\alpha\beta} - 1\right].$$

After t_s the output current is determined by the resistor, and the output exponentially decays to 0. The resulting output waveform begins as a quadratic, and ends as a exponential:

$$V_{out}(t) = \begin{cases} \frac{1}{2\alpha\beta\tau_f^2}t^2, & 0 \le t \le t_s \\ \left(1 - \frac{t_s^2}{2\alpha\beta\tau_f^2}\right)\exp\left(\frac{t_s - t}{\tau_f}\right), & t_s \le t \end{cases} \tag{5.2}$$

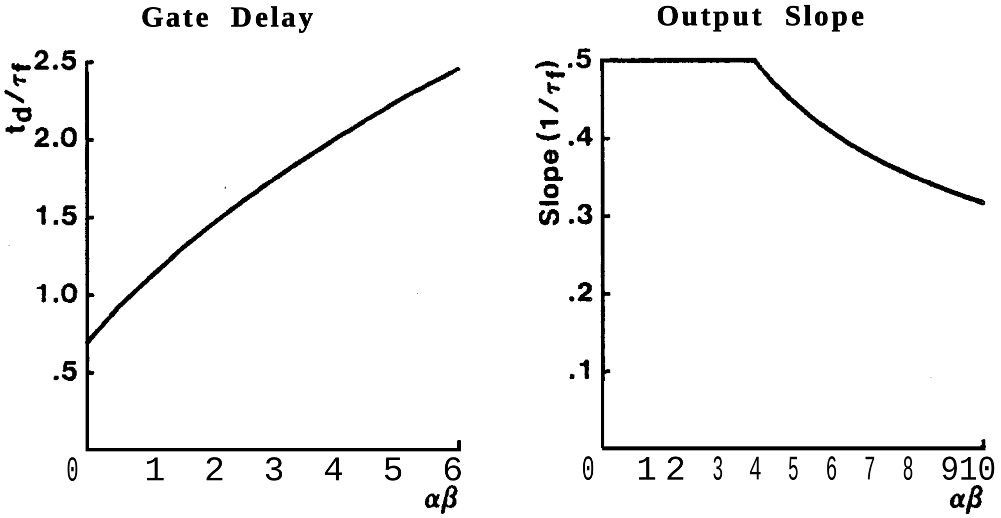


Figure 5.6 Gate delay and output slope versus input ramp rise time.

The gate delay, t_d, is defined to be the time required for the output to reach V_s after its input reaches V_s, so the delay through a series of gates is simply the sum of the individual gate delays. From the output waveform, both the gate delay and the output's slope at the switching voltage can be found:

$$t_d = \begin{cases} t_s + \tau_f \ln\left(\dfrac{t_s}{\alpha\beta\tau_f V_s}\right), & \alpha\beta \le \alpha\beta_{crit}; \\ \tau_f\sqrt{2\alpha\beta(1-V_s)}, & \alpha\beta \ge \alpha\beta_{crit}; \end{cases} \tag{5.3}$$

$$\left.\frac{dV_{out}}{dt}\right|_{V=V_s} = \begin{cases} V_s/\tau_f, & \alpha\beta \le \alpha\beta_{crit}; \\ \sqrt{\dfrac{2(1-V_s)}{\alpha\beta\tau_f^2}}, & \alpha\beta \ge \alpha\beta_{crit}. \end{cases} \tag{5.4}$$

where $\alpha\beta_{crit} = 2(1 - V_s)/V_s^2$. Figure 5.6 shows a plot of the gate delay and output slope as a function of $\alpha\beta$ for $V_s = .5$. The gate delay can be approximated by the simple formula:

$$t_d = \tau_f\sqrt{(\ln V_s)^2 + 2\alpha\beta(1 - V_s)} \tag{5.5}$$

It is not surprising that the importance of the input slope is determined by the size of $\alpha\beta$. This product represents the amount of time (measured in units of τ_f, the intrinsic gate delay) that it takes the output current to change from zero to full scale, since α is the number of time constants the input takes to change from 0 to 1, and β is the voltage range that must be crossed to turn the output current fully on. When $\alpha\beta$ is much less than one, the current changes rapidly compared to the output, and the output waveform is a simple decaying exponential. When $\alpha\beta$ is much larger than one, the output current is limited by the input, and the gate delay increases. The increase in delay is caused by slowing down the initial fall of the output waveform. The shape of the output waveform for low voltages is unaffected by the input slope, except for very slow inputs.

Another way to view the gate delay is as the sum of two terms: one is the intrinsic delay of the gate, the other is the delay caused by the input. Rewriting Eq. *(5.5)* in terms of the inverse of the input slope, setting $\tau_{in} = \alpha\tau_f$ gives

$$t_d = \sqrt{(\tau_f \ln V_s)^2 + 2\tau_{in}\tau_g(1 - V_s)}, \tag{5.6}$$

where $\tau_g = C_{load}/g_m$ is the time constant associated with the current source.

Since the output waveform for a falling transient looks exponential at low voltages, the rising output of a gate will be exponential at high voltages. Therefore, to find the output waveform at the end of a series of gates, the output for an exponential input must be determined. The derivation for an exponential input is similar to the derivation for a ramp.[†] Figure 5.7 shows the delay versus τ_{in}, the time constant of the exponential input, for a gate with $\beta = V_s = .5$.

The delay for an exponential input given in Figure 5.7 is almost identical to the delay for a ramp input, Eq. (5.5), if the slope of the ramp is set to the slope of the exponential at the switching point of the gate. This result is not surprising, since

[†] For an exponential input the time when the gate enters the low-gain region cannot be found explicitly. Instead an implicit relation must be used: both t_s and τ_{in} (the input time constant) are found as a function of the output voltage at t_s, $V_c(t_s)$.

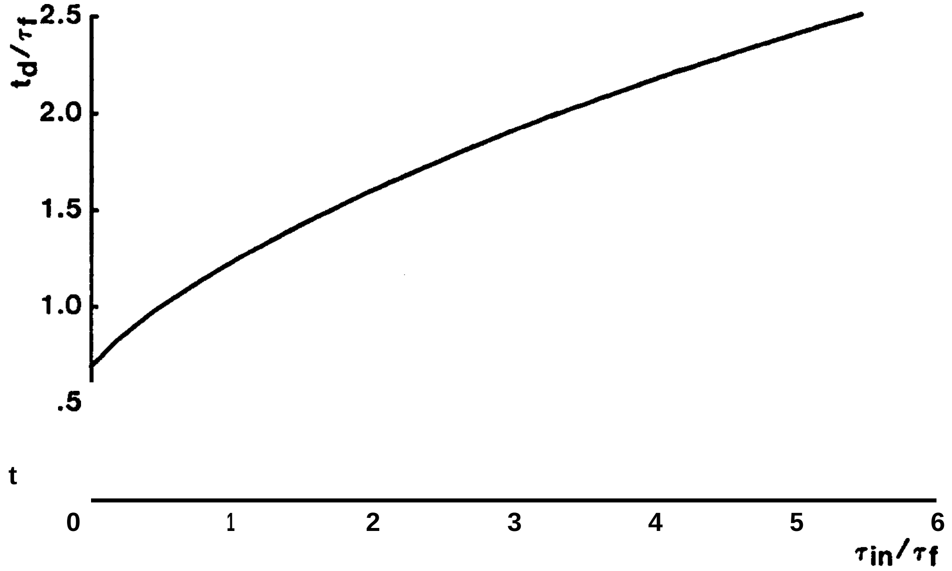


Figure 5.7 Gate delay for an exponential input.

the output is only affected by a narrow region of the input voltage. For slow input transitions, an exponential differs significantly from a ramp only after the output enters the low-gain region of the drive curve. When the input affects the output, the two waveform are nearly identical. The input shape has the largest affect on the output voltage when the input and output time constants are roughly equal; even in this case the difference in gate delay in Figure 5.7 and the delay for a ramp is less than **10** percent. Thus, the gate delay given in Eq. (5.5) is valid for exponential inputs when α is set equal to $\tau_{in}/(\tau_f(1 - V_s))$, the inverse slope of the exponential input.

Since Eq. (5.5) (or Eq. (5.6)) is valid for exponential inputs, it provides a delay model that can be used to determine the delay through a series of gates: the total delay is the sum of the gate delays, when each delay is found using the slope of its input waveform. The next section extends this timing model to handle nonlinear gates, and Section 5.4 shows how to apply this timing model to gates with complex output networks.

5.3.3 Nonlinear Gates

The technique used to derive the output for a linear gate can be used on nonlinear gates; the nonlinearity just makes it more difficult to find t_s, the time when the output enters the low-gain region. As was done for linear gates, all voltages have been normalized, and the falling output waveform is derived. The output current of a nonlinear gate can be written as the minimum of the nonlinear current source,

$$i_{out} = g_m g(V_{in} - V_s),$$

and the nonlinear resistor current,

$$i_{out} = \frac{V_{max}}{R_f} f(V_{out}).$$

R_f is defined to be $\frac{V_{max}}{i_{max}}$, so $f(1) = 1$, and g_m is defined to be $\frac{di_{out}}{dV_{in}}\Big|_{V_s}$, so the slope of g(0) is 1. Using these definitions, $\beta = (g_m R_f)^{-1}$ is the maximum **range** of *g(V)* where the input affects the output, and $\tau_f = R_f C_{load}$ is the intrinsic time constant for the falling waveform.

While the output is in the high-gain region, the output voltage is simply 1 minus the integral of the output current:

$$V_{out} = 1 - \frac{1}{\beta \tau_f} \int_0^t g(V_{in}(\tau) - V_s)\, d\tau$$

The gate enters the low-gain region when the resistor current equals the input current, i.e., t_s is the solution to

$$t_s \quad : \quad \beta f(V_{out}(t_s)) = g(V_{in}(t_s) - V_s).$$

The output waveform after t_s is the solution to the single-time-constant nonlinear RC problem, $T(t/\tau_f)$.

The effect of the nonlinear transconductance can be visualized by creating another circuit with a linear g_m and distorting its input so the output currents of the two systems are equal. To maintain equal currents, the distorted input is $V_s + g(V_{in} - V_s)$. Since the slope of g(0) is one, the distorted input looks similar to the actual input around the gate’s switching point. As was previously shown, the shape of the input is only important near the switching point. Thus, gentle nonlinearities in transconductance do not have a significant effect on the output waveform.

For an input waveform where $g(V_{in} - V_s)$ is a ramp,

$$V_{in} = V_s + g^{-1}(t/\alpha\tau_f),$$

the initial output is identical to the output of a linear gate driven by a ramp:

$$V_{out} \doteq 1 - \frac{t^2}{2\alpha\beta\tau_f^2}.$$

The resistor begins to control the output current when the voltage drop across the resistor becomes equal to the output voltage. Although it is difficult to determine this transition time (t_s) as a function of the input time constant, t_s and α can be found as a function of $V_{out}(t_s)$, the output voltage at the transition time’:

$$t_s = 2\tau_f\left[\frac{1 - V_{out}(t_s)}{f(V_{out}(t_s))}\right]$$

$$\alpha = \frac{2}{\beta}\left[\frac{1 - V_{out}(t_s)}{[f(V_{out}(t_s))]^2}\right]$$

The waveform after t_s is

$$V_{out} = T(t/\tau_f - t_o); \qquad t_o = \frac{t_s}{\tau_f} - T^{-1}(V_{out}(t_s)).$$

This gives a gate delay of

$$t_d = \begin{cases} \tau_f[T^{-1}(V_s) + t_o], & \alpha\beta \leq 2\dfrac{1-V_s}{[f(V_s)]^2}; \\ \tau_f\sqrt{2\alpha\beta(1-V_s)}, & \alpha\beta \geq 2\dfrac{1-V_s}{[f(V_s)]^2}. \end{cases} \tag{5.7}$$

Like the delay through a linear gate, this delay can also be approximated by a simple formula:

$$t_d = \tau_f\sqrt{[T^{-1}(V_s)]^2 + 2\alpha\beta(1-V_s)}. \tag{5.8}$$

5.3.4 MOS Gates

A MOS gate is nonlinear since both the input transconductance and the output resistance are nonlinear. Surprisingly, these nonlinearities have only a small effect on the output waveform: the output of a MOS gate is very similar to the output of a linear gate.

Figure 5.8 shows the pulldown current versus input voltage for an nMOS inverter; the curve for CMOS is similar. The transconductance increases slightly with increasing gate voltage. This nonlinearity results from the quadratic relation between gate voltage and drain current for a MOS transistor. The output current is not proportional to the input voltage squared, since the gate's output current is the difference between the pullup and pulldown transistor currents. The nonlinearity is reduced even further when high-field effects on the transistor are included. The result is that the output current has only a small deviation from a linear g_m. This nonlinearity has a small effect on the output waveform and will be ignored.

The nonlinear output resistance sets the shape of the intrinsic output waveform. For nMOS and CMOS gates, the nonlinear resistor leads to falling waveforms that

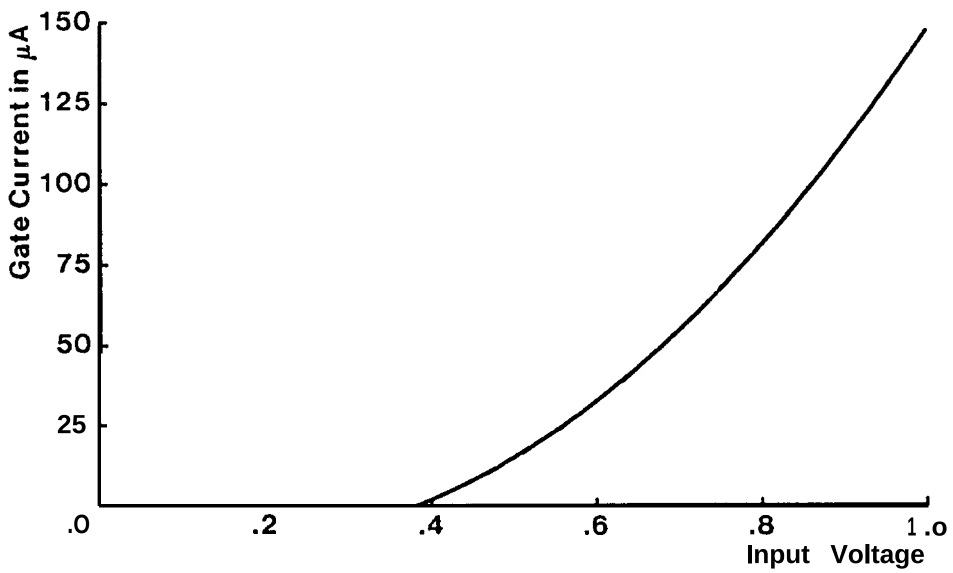


Figure 5.8 Output current vs. input voltage for an nMOS gate.

are roughly $1-\tanh(t)$ and rising waveforms that are $\tanh(t)$.[†] The change in output waveform has only a small effect on the gate delay. Using a simple quadratic model for a transistor, with $V_{dd} = 5V$, and $V_{th} = 1V$, $f(V)$ for a MOS gate is

$$f(V) = \begin{cases} 1 & .8 \leq V; \\ 1-(1-V/.8)^2 & 0 \leq v \leq .8. \end{cases}$$

For this $f(V)$,

$$T(t) = \begin{cases} 1 - t & 0 \leq t \leq .2; \\ .8(1-\tanh(t/.8-.25) & .2 \leq t. \end{cases}$$

Using this function of $T(t)$, in Eq. (5.5) gives the falling gate delay:

$$t_d = \tau_f\sqrt{\left[.2+.8\tanh^{-1}(1-V_s/.8)\right]^2 + 2\alpha\beta(1-V_s)}, \qquad V_s \leq 0.8. \qquad (5.9)$$

[†] The rising waveform is for a depletion load in nMOS and for a p-channel transistor in CMOS. Actually the output waveforms are not $\tanh(t)$ for the entire output range. For a step input, the output initially ramps down untii the transistor leaves the saturation region. After this initial ramp, the output decays with a $1 - \tanh(t)$ time dependence.

Since an nMOS inverter operates with a pulldown device overpowering the pullup, its drive curves are asymmetric; see Figure 5.2. The switching voltage of the gate, V_s, is located below half scale and the series resistance for the rising transition, R_r, is much larger than the resistance for the falling transition, R_f. This difference in series resistance affects both the gate's input sensitivity and its delay. Since the resistance determines the intrinsic output time constant, the rising output will, in general, be slower than the falling output. In addition, the large series resistance, R_r, and relatively large g_m limit the range of voltages where the input affects the output current for a falling input. Together, the low sensitivity to falling inputs and the generally slow rising outputs in nMOS cause the input's effect on the rising output waveform to be small, but its effect on the falling output to be important.

The drive curves for an nMOS NOR gate are similar to the drive curves for aninverter. If only one of the inputs is on, the presence of the other inputs has no effect. The drive curves for an NAND gate differ from those of an inverter. The g_m of the gate is similar to the inverter; however, the series resistances, R_r and R_f, are n times bigger, where n is equal to the number of inputs. This increase in series resistance increases the intrinsic gate delay, $\tau_f = R_f C_{load}$, as one might expect, but it also decreases the gate's sensitivity to input waveforms. The increase in series resistance decreases β, the range of voltages where the input affects the output, by n. The result is the input must be n times slower relative to the output to cause the same relative effect on the output, or n^2 times slower than the input to an equivalent inverter.

5.4 Complex Gates

When the output network of a gate is an RC tree, directly calculating the output waveform becomes difficult. To find the output waveform involves solving a set of differential equations. The single-time-constant approximation — assuming

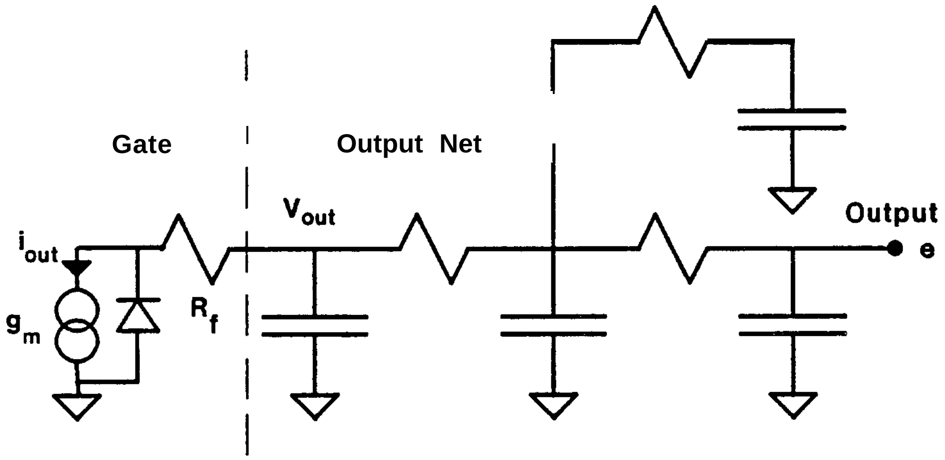


Figure 5.9 Circuit model for a complex gate.

all nodes decay at the same rate — reduces this complex problem to one of finding the output of an equivalent simple gate. To insure the errors in the estimate are small, bounds are also derived. When the bounds are close to the estimate, the errors in the single-time-constant approximation are small. When the upper and lower bounds are very different from each other, a multiple-time-constant estimate may be required.

Figure 5.9 shows an example transistor cluster. It consists of a gate driving an output net, which is modelled by an RC tree. The output waveform for node e in the RC tree is derived. The problem of finding the output waveform is similar to finding the output waveform for a grounded RC tree, which has been discussed in Chapters 3 and 4. However, the problems are not identical, since the RC tree is driven by a current source during part of the output. First an estimate of the output waveform for a linear tree is derived, and then the results are generalized to include nonlinear RC trees. Bounds on the output are a!so derived.

5.4.1 Waveform Estimate

For a current source drive, the sum of the capacitor currents in the tree must be equal to the drive current:

$$i(t) = -\sum_k C_k \frac{dV_k}{dt}. \tag{5.10}$$

A single-time-constant estimate for the output voltage is found by assuming all nodes decay at the same rate: $\frac{dV_k}{dt}$ is equal to $\frac{dV_e}{dt}$. This gives the estimated output voltage, V_e^*, while the gate's output is in the high gain part of the drive curve:

$$V_e^* = 1 - \frac{\int_0^t i(\tau)\,d\tau}{C_T}, \tag{5.11}$$

where C_T is equal to the sum of the capacitors in the RC tree. To determine when the gate's current becomes limited by the resistor, the voltage at the output of the gate must also be found. The voltage drop between V_e and the gate's output in a linear tree is

$$V_e - V_{out} = -\sum_k R_{ke} C_k \frac{dV_k}{dt}.$$

Again using the single-time-constant approximation, this voltage difference can be related to $\frac{dV_e}{dt}$, which is i_{out}/C_T:

$$V_{out}^* = V_e^* - R_\mu i_{out};\ R_\mu = \frac{\sum_k R_{ke} C_k}{\sum_k C_k}.$$

The output enters the low-gain region when the voltage at the output of the current source equals 0. This transition occurs when $(R_f + R_\mu) i_{out} = V_e^*$. Once the gate current is determined by the series resistance, the problem becomes identical to estimating the output of a grounded RC tree. The resulting output is a decaying exponential, with a time constant τ_{De}:

$$\tau_{De} = \sum_k (R_f + R_{ke}) C_k = (R_f + R_\mu) C_T.$$

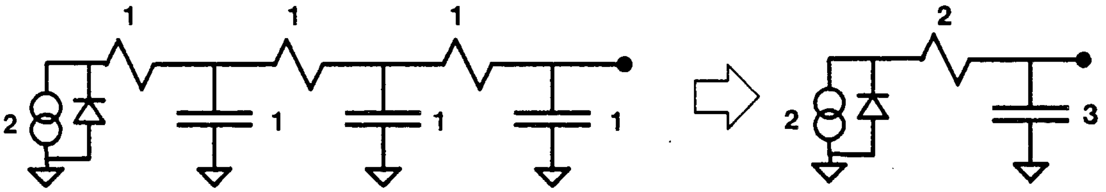


Figure 5.10 Conversion from a complex gate into a simple gate.

The estimated output voltage, V_e^*, is identical to the output of a simple linear gate, where the load capacitance of the simple gate is equal to the sum of all capacitors in the RC tree, and where the series resistance of the simple gate is equal to R_f + R,. The transconductance of the simple gate is g_m, the value of the original gate. Using the single-time-constant approximation converts a general transistor cluster into an equivalent simple gate; see Figure 5.10. The delay through a complex gate is approximately

$$t_d = \sqrt{[\tau_{De} \ln(V_s)]^2 \cdot_t 2\tau_{in}\tau_g(1-V_s)} \tag{5.12}$$

where the input slope at V_s is $1/\tau_{in}$. The delay is composed of two terms: one represents the intrinsic delay of the gate and the other represents the delay caused solely from the input. The value $\tau_{De} \ln(V_s)$ is the cluster delay for a step input. This is the delay predicted by the models in the previous chapters. The other term is the cluster delay if all the resistors in the circuit are set to zero. Then, the time required for the current source to discharge all the capacitors to V_s, $\sqrt{2\tau_{in}\tau_g(1-V_s)}$, sets the delay.

The timing model for complex linear gates can be extended to cover MOS transistor clusters because these circuits can be transformed into pseudo-linear networks. Since the derivation is analogous to the complex linear gate derivation, it is only reviewed here. The single- time-constant approximation is again used to convert a complex transistor cluster into a simple gate. The nonlinearity just makes the resulting simple gate nonlinear. All resistors, including the gate's output resistor, are assumed to have the same type of nonlinearity. When the gate's output resistor has a different type of nonlinearity, then the estimate becomes a mixed nonlinear problem and can be solved using the method described in Section 4.7.

The current through a nonlinear resistor will be written as

$$i = \frac{V_{max}}{R_{eff}}[f(V_1) - f(V_2)],$$

where R_{eff} is the effective resistance of the device. As was described in Chapter 4, this device appears linear if a transformed voltage, U = ***f(V),*** is used instead of ***V.***

For a current source drive, the constraint on the output tree is only on its current; the estimate for a nonlinear circuit is the **same** as the estimate for a linear gate (Eq. (5.11)). For a nonlinear circuit, the gate output voltage must be found by estimating the U-drop **in** the output net:

$$U_e - U_{out} = \sum_k R_{ke}C_k\frac{dV_k}{dt}.$$

Again using the single-time-constant approximation, this U difference can be found as a function of current:

$$U^*_{out} = U^*_e - R_\mu i_{out}; \qquad R_\mu = \frac{\sum_k R_{ke}C_k}{\sum_k C_k}.$$

The input loses control of the output current when the voltage drop across the current source becomes 0. This transition occurs when $(R_f + R_\mu)i_{out} = f(V^*_e)$. After the output resistance of the gate takes over, the problem becomes identical to finding an output for a grounded nonlinear tree. The resulting output is $T(t/\tau_{De})$, where $T(t)$ is the solution to the nonlinear RC network, and τ_{De} is $\sum(R_f + R_{ke})C_k$.

Like linear complex gates, the estimated output voltage is identical to the output of a simple gate, but for nonlinear output networks, the simple gate is also nonlinear. The load capacitance of the simple gate is C_T, and the output resistor has an effective resistance of $R_\mu + R_f$.

5.4.2 Waveform Bounds

To generate bounds on the output waveform, Eq. (5.10) is integrated and then the V_k are replaced by bounds in terms of V_e. This substitution yields bounds on the output voltage. First bounds for a linear network are derived and then they are extended to model nonlinear circuits. Integrating Eq. (5.10) yields

$$\sum_k C_k V_k = C_T - \int_0^t i_{out}(\tau)\, d\tau \qquad (5.13)$$

Bounds on V_k are found by assuming all the current flows to node k (giving on upper bound on V_k) or node e (giving a lower bound on V_k). For a linear network this gives:

$$V_e - (R_{ee} - R_{ke}) i_{out} \leq V_k \leq V_e + (R_{kk} - R_{ke}) i_{out}.$$

Using these bounds in Eq. (5.13) gives bounds on the output voltage:

$$\begin{aligned} C_T V_e &\geq C_T - \int_0^t i_{out}(\tau)\, d\tau - i_{out}(\tau_P - \tau_{De}); \\ C_T V_e &\leq C_T - \int_0^t i_{out}(\tau)\, d\tau + i_{out}(C_T(R_{ee} + R_f) - \tau_{De}); \end{aligned} \qquad (5.14)$$

where

$$\tau_P = \sum_k (R_f + R_{kk}) C_k; \qquad \tau_{De} = \sum_k (R_f + R_{ke}) C_k.$$

These bounds on V_e hold until the gate enters the low-gain region of the drive curves. The transition time depends on the voltage at the gate's output, V_{out}. Since this voltage is not known exactly, the transition time must also be bounded. For the lower bound on the tree's output, the gate current should be maximized, so the gate's output voltage is set at its upper bound, $V_{out} = V$,. The latest time the gate could enter the low-gain region is when

$$V_e(t_{s_l}) = R_f i_{out}(t_{s_l}).$$

An upper bound on V_e requires the gate current be minimized, which in turn requires that the gate's output voltage be as low as possible: $V_{out} = V_e - (R_{ee} + R_f)i_{out}$. The earliest the gate could enter the low-gain region is

$$V_e(t_{s_u}) = (R_f + R_{ee})i_{out}(t_{s_u})$$

After the transition has occurred, the waveform bounding problem reduces to finding bounds for a grounded RC tree. This problem is similar to the one solved in Chapter 3, the primary difference being that the initial voltages in the tree are not equal. The resulting bounds are

$$V_e \le V_e(t_{s_u}) \exp((t_{s_u} - t)/\tau_P), \qquad t \ge t_{s_u}$$

$$V_e \ge V_e(t_{s_l}) \exp((t_{s_l} - t)/\tau_{Re}), \qquad t \ge t_{s_l}$$

where

$$\tau_{Re} = \sum_k \frac{(R_f + R_{ke})^2}{R_{ee}} C_k; \qquad \tau_P = \sum_k (R_f + R_{kk})C_k.$$

The bounds are continuous at the transition time.

For a ramp input, $V_{in} = V_s + t/\tau_{in}$, the bounds on the output are initially

$$\left[1 - \frac{t^2}{2\tau_{in}\tau_g} - \frac{(\tau_P - \tau_{De})t}{\tau_{in}\tau_g}\right] \le V_e \le \left[1 - \frac{t^2}{2\tau_{in}\tau_g} + \frac{((R_{ee} + R_f)C_T - \tau_{De})t}{\tau_{in}\tau_g}\right] \qquad (5.15)$$

After t_s, the output bounds are simple decaying exponentials:

$$\frac{(R_f C_T)t_{s_l}}{\tau_{in}\tau_g} \exp\left(\frac{t_{s_l} - t}{\tau_{Re}}\right) \le V_e \le \frac{(R_{ee} + R_f)C_T t_{s_u}}{\tau_{in}\tau_g} \exp\left(\frac{t_{s_u} - t}{\tau_P}\right), \qquad (5.16)$$

where

$$t_{s_u} = \tau_{De}\left[\sqrt{1 + \frac{2\tau_{in}\tau_g}{\tau_{De}^2}} - 1\right];$$

$$t_{s_l} = (\tau_P - \tau_{De} + \tau_f)\left[\sqrt{1 + \frac{2\tau_{in}\tau_g}{(\tau_P - \tau_{De} + \tau_f)^2}} - 1\right].$$

The bounds, the estimate and the actual output for the circuit shown in Figure **5.11a** are shown in Figure **5.11b** $(\tau_{in} = 2\tau_{De})$ and Figure 5.11~ $(\tau_{in} = 10\tau_{De})$.

The bounds on the output waveform depend on the same three time constants that define the bounds for a step input: τ_{De}, τ_{Re}, and τ_P. If the output has good bounds for a step input, the case where τ_{Re} roughly equals τ_P, then the bounds for a continuous input also will be good. It is not difficult to show that the bounds are worst for a step input. Slowing down the input makes the bounds closer together, since with a slow input the output is controlled by the input current source, which is known, and not by its intrinsic time constants, which must be bounded.

The derivation for a nonlinear circuit follows the derivation used for linear circuits. The output bounds are found by bounding V_k by V_e in Eq. (5.13). For a nonlinear circuit, the bounds on V_k are generated from the bounds on U_k. The latter can be determined by looking at the current flow in the tree:

$$U_e - (R_{ee} - R_{ke})i_{out} \leq U_k \leq U_e + (R_{kk} - R_{ke})i_{out}.$$

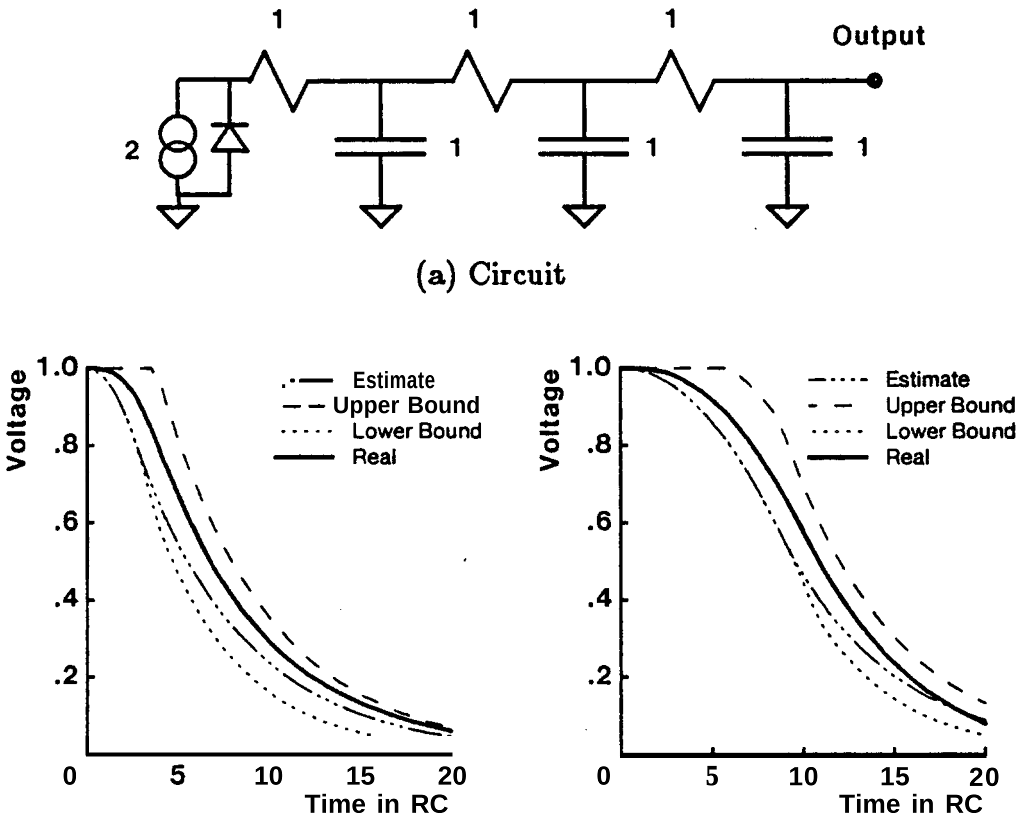
These bounds can be converted to yield bounds on the nodal voltages. Using the voltage bounds in Eq. (5.13) yields bounds on the output voltage. The output bounds have the same form as the linear bound, but the definitions of the time constants differ.

The gate's U output can be bounded in terms of U_e, providing a method of bounding when the gate enters the low-gain region:

$$f(V_e(t_{s_l})) = R_f i_{out}(t_{s_l});$$

$$f(V_e(t_{s_u})) = (R_f + R_{ee})i_{out}(t_{s_u}).$$

After this transition occurs, bounding the output reduces to finding bounds for a grounded, (possibly mixed) nonlinear RC tree. The resulting bounds have a $T(t)$



(b) Normal input, $\tau_{in} = 2\tau_{De}$ (C) Slow input, $\tau_{in} = 10\tau_{De}$

Figure 5.11 Output of a complex gate for different input slopes. $\tau_{De} = 6$, $\tau_g = 1.5$.

shape, the output of the single nonlinear RC circuit. Like linear networks, the bounds on the output waveform are worst when the input to the gate is a step.

5.5 Summary

The timing models presented in Chapters 3 and 4 were derived assuming the input to a transistor cluster was a step waveform. This approximation allowed one to determine the intrinsic delay of the cluster. Unfortunately, these timing models cannot accurately determine the output of a transistor cluster, since the input waveform is rarely a step.

To remove this limitation, this chapter has used a more complete gate model to determine the effect of input shape on the output waveform. The drive curves of a gate, a contour map of the gate output current versus input and output voltage, provide the basis for this improved gate model. The output range of a gate can be divided into two regions. In the high-gain region, the output current is mainly controlled by the input voltage; in the low-gain regions the output current is mainly controlled by the output voltage. By neglecting the effect of the output voltage in the high-gain region and the effect of the input voltage in the low-gain region, a simple gate model has been derived. This simple gate model reduces to the resistor model used in Chapters 3 and 4 for step inputs.

Using this gate model, the output waveform dependence on input waveform has been determined. For a simple gate, one with a purely capacitive load, the output can be determined exactly. The output waveform is not strongly coupled to the input waveform. For fast inputs, the input shape is irrelevant: the intrinsic gate time constant controls the output. waveform. For slow inputs, the input changes only slightly during the output transient, and again the shape of the input is not very important. The output waveform can be determined using only a first order approximation to the input, $V_{in} \approx V_s + t/\tau_{in}$. The gate delay can be approximated by the square root of the sum of the squares of two terms: the intrinsic delay of the gate (for a step input) and the delay caused by the input waveform assuming the intrinsic delay is zero.

The single- time-constant approximation has been used to convert complex gates into an equivalent simple gate. Thus, the timing models derived for simple gates apply to complex transistor clusters as well. To check the accuracy of the single-time-constant approximation, bounds on the output, waveform have been derived. The bounds are tightest for very slow inputs and become worse as the input becomes faster. The worst bounds are for a step input, where they are equivalent to the simple bounds derived in Chapters 3 and 4. If the bounds for a step input are good, the bounds for a continuous input will also be good.

Using this timing model, it is possible to find the delay through a complex transistor clusters. Thus, these timing models can be used to solve the problem posed in Chapter 1: they can be used to find the delay through a complex MOS circuit by determining the delay through its transistor clusters.

Chapter 6

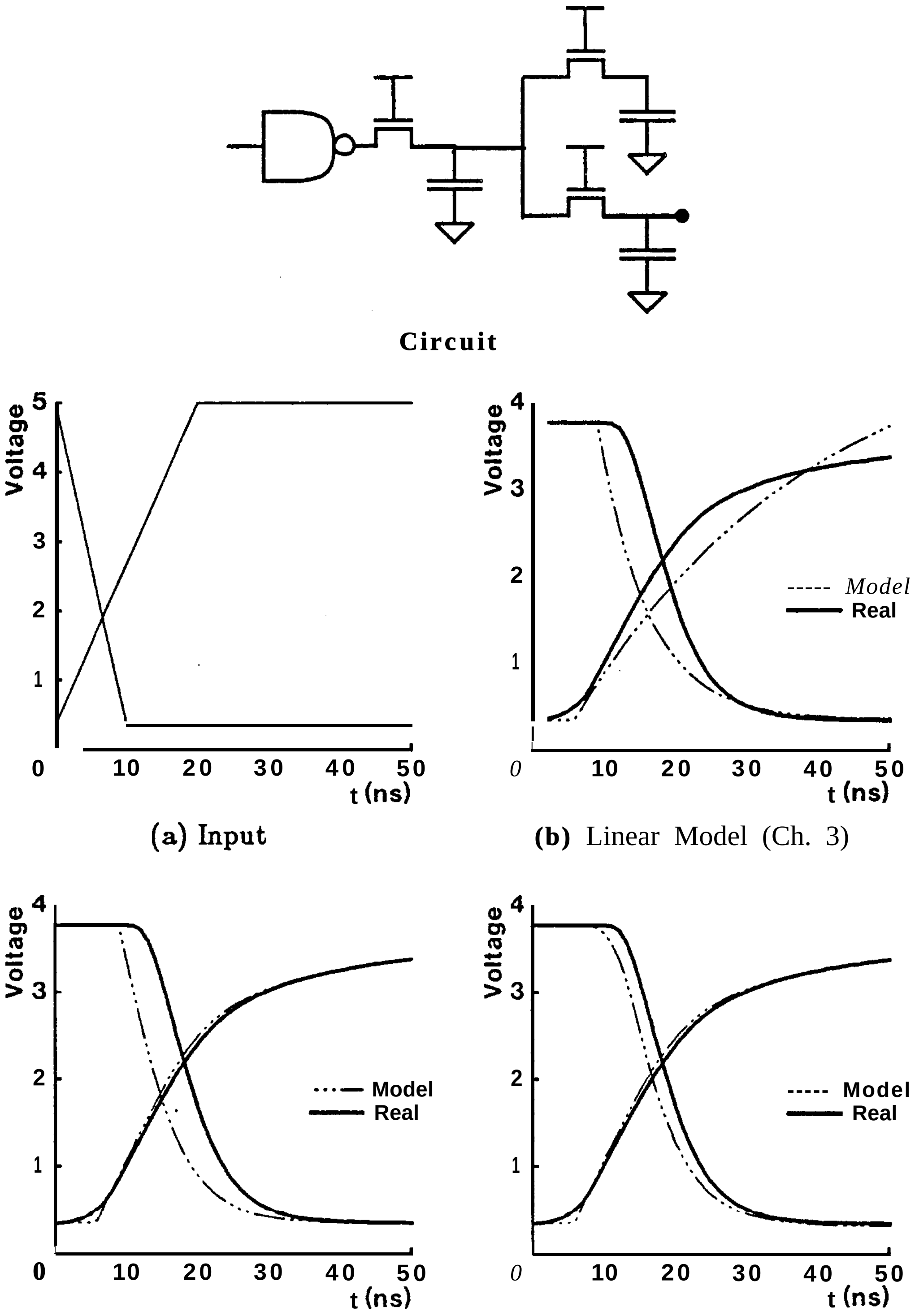
CONCLUSIONS

To determine the delay through a large MOS circuit, the delay through the transistor clusters that compose the circuit must be found. The timing models used to determine the cluster speed must be simple, since an integrated circuit can contain tens of thousands of clusters. Currently, timing analysis tools use empirical models to estimate the cluster delay.

This thesis has presented a new timing model that retains the simplicity of previous models, but is firmly grounded in the governing physics of the circuit. This coupling between the model and the actual circuit eliminates the need to base the output estimate on empirical results. Since the approximations used to derive an estimate of the output waveform are explicit, the maximum error caused by these approximations also can be determined. The coupling between the estimate and the circuit physics provides more than just error control; it provides a way to view MOS circuits so performance questions are easy to answer.

The timing model was developed in three stages, which is illustrated in Figure 6.1. First came a linear timing model. Here transistors were approximated by linear resistors, and the inputs to a transistor cluster were approximated by step waveforms. The model of a transistor cluster became an RC tree, and its output waveform could be estimated using the single-time-constant approximation. The accuracy of the single-time-constant approximation could be checked by deriving waveform bounds, but the bounds did not check the validity of the other two approximations. Figure 6.1b shows the output of this model.

The timing model was improved by applying the single-time-constant approximation directly to a nonlinear MOS RC tree, eliminating the need to model each



Circuit

(a) Input

(b) Linear Model (Ch. 3)

(c) Nonlinear Model (Ch. 4)

(d) Full Model (Ch. 5)

Figure 6.1 Output estimates of the timing models.

transistor as a linear resistor. Again bounds were derived to check the validity of the single-time-constant approximation, leaving only the step input approximation unchecked. Figure 6.1c shows the output of this model.

Finally, a new gate model was developed based on the drive current of a gate. This model together with the single-time-constant approximation provided a simple method to estimate the output waveform of a transistor cluster for an arbitrary input waveform. Since the approximations used to derive the output estimate are explicit, the range of circuits where this model will be valid is also known. Figure 6.1d shows the output of this model.

6.1 Future Work

The models presented in this thesis show how to analyze most MOS circuits, but suffer from two limitations: 1) the models assume the dc voltage levels of the signals are known, since the model output is in normalized voltage, and 2) the models assume the output of a transistor cluster does not affect its input.

The dc voltage range for the timing models can be found by first performing a static analysis of a circuit. The static analysis must be able to find the voltage range for every signal in the circuit, as well as determine the switching point of every gate. This analysis is useful in its own right, since it provides a method to check other aspects of circuit design, for example, noise margins and power dissipation. Once the voltage swings are known, the timing models presented in this thesis can be applied to a broad class of circuits,' including cascode amplifiers and voltage clamped busses.

The other limitation of the timing models is the requirement that the input waveform be independent of the output. The output is determined as a function of the input, but there is no way to determine the output when the input also depends on the output voltage. This situation occurs in circuits that use positive feedback to increase circuit performance, such as bootstrap drivers and sense amplifiers. To analyze these types of circuits, the single time constant approximation must be extended to handle circuits that contain multiple transistor clusters.

6.2 Final Thoughts

If the two limitations of the timing models are removed, the models should be powerful enough to analyze *all* MOS circuit forms. The performance of a MOS circuit only seems to depend on a few factors: the voltage swing, the type of non-linear devices present, and the effective single-time-constant — a number which can be found easily from the circuit. This improved understanding of the factors that affect circuit performance is the most important result of this work.

Appendix A

VOLTAGE BOUNDS IN RC TREES

A.1 Simple Bounds

For a falling transition, all nodes in an RC tree monotonically decrease with time [Wy82] and therefore current flows from every capacitor to the root of the tree (ground). Because the current sourced from every capacitor is positive, the current flowing along a path is guaranteed to increase monotonically and the voltage along a path will decrease monotonically as one gets closer to the root. Using these results, it is possible to bound one voltage in an RC tree in terms of another.

To bound the voltage at node k by the voltage at node e, define node n to be the last node on the path to both nodes e and k. Then

$$V_k \geq V_n; \qquad \frac{V_n}{R_{nn}} \geq \frac{V_e}{R_{ee}}; \qquad R_{nn} = R_{ke};$$

and

$$V_e \geq V_n; \qquad \frac{V_n}{R_{nn}} \geq \frac{V_k}{R_{kk}}.$$

Combining the first set of inequalities gives a lower bound on V_k; the second set gives an upper bound:

$$V_k \geq \frac{R_{ke}}{R_{ee}} V_e; \qquad V_k \leq \frac{R_{kk}}{R_{ke}} V_e. \tag{A.1}$$

In the general case, if some function of the voltage, $U = f(V)$, is proportional to the current, then the bounds become

$$U_k \geq \frac{R_{ke}}{R_{ee}} U_e; \qquad U_k \leq \frac{R_{kk}}{R_{ke}} U_e. \tag{A.2}$$

A.2 Improved Bounds

The above bounds were generated using only information about the sign of the capacitor currents; the bounds did not use any information about the time behavior of the currents. Using this information leads to improved voltage bounds.

For the falling transition in an RC line,

$$\frac{-\frac{dV_n}{dt}}{V_n} \geq \frac{-\frac{dV_q}{dt}}{V_q}, \tag{A.3}$$

when ‘node q is downstream of node n. At $t = 0^+$, dV_k/dt is 0 except at the first node, where it is negative, so the inequality holds. Since the equality condition of Eq. (A.3) forms a boundary in the solution space, if the inequality holds for the initial conditions, it holds for all time. This bound on dV_k/dt is used to improve the bounds for an RC line.

In an RC line, the voltage drop between the output at the end of the line, e, and node k is

$$V_e - V_k = -\sum_n (R_{ne} - R_{nk}) C_n \frac{dV_n}{dt}.$$

The only nonzero terms in the sum are for nodes downstream of k. Using Eq. (A.3), an upper bound on the voltage drop can be found:

$$V_e - V_k \leq -\sum_n (R_{ne} - R_{nk}) C_n \frac{V_n}{V_k}\frac{dV_k}{dt} \leq -(\tau_{De} - \tau_{Dk}) \frac{V_e}{V_k}\frac{dV_k}{dt}. \tag{A.4}$$

Eq. (A.3) also provides a upper bound on $-dV_k/dt$:

$$V_k = -\sum_n R_{nk} C_n \frac{dV_n}{dt} \geq -\frac{dV_k}{dt} \sum_{n \leq k} R_{nk} C_n \frac{V_n}{V_k} - \frac{dV_e}{dt} \sum_{n > k} R_{nk} C_n \frac{V_n}{V_e}.$$

Although setting dV_e/dt to 0 yields an upper bound on dV_k/dt, this approximation is inconsistent with the approximation used in Eq. (A.4) to find an upper bound on $V_e - V_k$. Considering both equations together requires dV_e/dt to be maximized to

minimize V_k:

$$V_k \geq -\frac{dV_k}{dt} \sum_n R_{nk} C_n \frac{V_n}{V_k} \geq -\frac{dV_k}{dt} \tau_{Rk}. \tag{A.5}$$

Combining Eq. (A.4) and Eq. (A.5) yields a lower bound on V_k:

$$V_k \geq V_e \left(1 - \frac{\tau_{De} - \tau_{Dk}}{\tau_{Rk}} \right) \tag{A.6}$$

This improved lower bound on V_k only changes the definition of τ_{Re}, which becomes

$$\hat{\tau}_{Re} = \sum_k \alpha_{ke} R_{ke} C_k; \qquad \alpha_{ke} = \max\left(\frac{R_{ke}}{R_{ee}}, 1 - \frac{\tau_{De} - \tau_{Dk}}{\tau_{Rk}} \right). \tag{A.7}$$

Appendix B

RC MESHES

Not all transistor clusters can be represented by a nonlinear RC tree. In some circuits more than one resistive path exists from a node to a voltage source. Examples of nontree circuits are output networks with loops and outputs driven by two voltage sources. These circuits can be modelled by an RC mesh: a resistor mesh, where every node in the mesh may have a capacitor to ground. Figure B.l shows two simple RC meshes.

If all the voltage sources driving the RC mesh are not at ground, then the output voltage will have a dc component in addition to the transient output. The transient problem can be separated from the dc problem by using superposition. First, the dc response is determined, and then this component of the nodal voltages is removed. The remaining component of each nodal voltage is positive, and decays monotonically to ground. The transient output waveform can be estimated using the timing models for RC trees; only the definition of R_{ke} needs to change.

For an RC tree, R_{ke} is defined to be the resistance of the path to ground common to both nodes e and k. This, resistance is useful, since i_k R_{ke} represents the voltage induced at node e from a current at node k. If R_{ke} for a RC mesh is defined to be V_e/i_k when $i_{n\neq k}$ = 0, then the transient output voltage again can

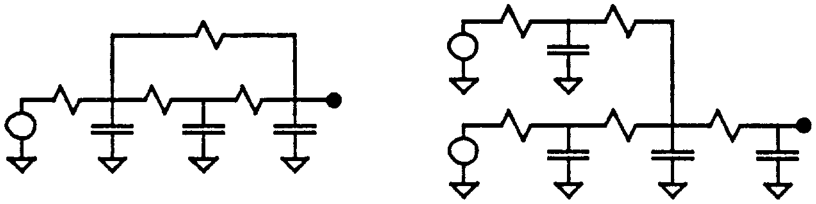


Figure B.l Two simple RC meshes.

be written as a sum of the voltages caused by each capacitor current:

$$V_e = -\sum_k R_{ke} C_k \frac{V_k}{dt}$$

Moreover, using this definition of R_{ke}, the bounds on the nodal voltages derived in Appendix A,

$$\frac{R_{ke}}{R_{ee}} V_e \leq V_k \leq \frac{R_{kk}}{R_{ke}} V_e,$$

are valid for an RC mesh as well. Thus, both the estimate and bounds derived for RC trees can be applied to an RC mesh.

Although extending the timing models to cover RC meshes is conceptually easy, it is computationally difficult. In general, determining R_{ke} for an RC mesh requires solving a set of nodal equations, a task which is superlinear in the number of nodes. For circuits that are mostly a tree, ad hoc techniques can be applied to efficiently determine R_{ke}.

Appendix C

BOUNDS IMPROVEMENT

When τ_P is much larger than τ_{De}, some capacitors on a side branch of the tree decay slowly compared to the output. These slow nodes are not strongly coupled to the output voltage. The improved estimate overcame this problem by grouping these slow nodes together and letting them decay at their own rate. This same technique also can be used to improve the bounds.

When the output is low and tracking the slow nodes, the upper bound on the voltage at these nodes is a good approximation to the actual voltage. The problem with the bounds occurs during the initial transient, when all the internal nodes are close to 1, yet V_k is approximated by $\frac{R_{kk}}{R_{ke}}$, a voltage much larger than one. Bounds improvement for this case is quite easy, since V_k is always less than or equal to 1:

$$V_k \leq \min\left(1, \frac{R_{kk}}{R_{ke}} V_e\right). \qquad (C.1)$$

The best upper bound on a nodal voltage depends on the output voltage. With out losing generality, let α_{ke} = 1 if $V_e \leq R_{ke}/R_{kk}$ and be zero otherwise. Then the improved bound on g_e is[†]

$$g_e = \sum_k R_{ke} C_k V_k \leq \tau_{De}^{\alpha} + \tau_P^{\alpha} V_e,$$

where

$$\tau_{De}^{\alpha} = \sum_k (1 - \alpha_{ke}) C_k R_{ke}; \qquad \tau_P^{\alpha} = \sum_k \alpha_{ke} C_k R_{kk}.$$

[†] Notice that approximating the nodal voltage as 1 has the same effect as setting the capacitance at that node to zero. An alternative way of looking at the bounds improvement is that it constructs a faster network, by removing capacitors, that has a better lower bound.

The improved lower bound on the voltage becomes

$$V_e \geq \frac{g_{e_{lower}} - \tau_{De}^{\alpha}}{\tau_P^{\alpha}}. \tag{C.2}$$

Although using the best τ_P^{α} and τ_{De}^{α} pair for every voltage would require too many time constants, most of the time constants are not needed. Since every τ_P^{α} and τ_{De}^{α} pair form a valid lower bound, one may use as many or as few as desired. Usually, one pair with α_{ke} set for $V_e = .5$ in addition to the original τ_P is sufficient to generate a good lower bound.

An improved lower bound on V_e also can be used to improve the upper bound by generating a better upper bound on g_e:

$$g_e \leq \begin{cases} \tau_{De}^{\alpha} + (\tau_{De} - \tau_{De}^{\alpha}) \exp(-t/\tau_P^{\alpha}), & \text{for } t \leq t_o; \\ [\tau_{De}^{\alpha} + (\tau_{De} - \tau_{De}^{\alpha}) \exp(-t_o/\tau_P^{\alpha})] \exp((t_o - t)/\tau_P), & \text{for } t > t_o, \end{cases} \tag{C.3}$$

where

$$t_o = \tau_P^{\alpha} \ln \left[\frac{(\tau_{De} - \tau_{De}^{\alpha})(\tau_P - \tau_P^{\alpha})}{\tau_P^{\alpha} \tau_{De}^{\alpha}} \right].$$

This improvement is significant only if τ_{De}^{α} for the modified circuit is small compared to τ_{De}, and τ_P^{α} is much less than τ_P. RC trees with the dominant capacitance (but not necessarily the dominant time constant) located near the path from the output to ground can be improved in this manner.

When the dominant capacitance in a RC tree lies far from the path between the output and ground, the upper bound for the output voltage will not significantly improve when the upper bound on g_e improves. An alternative method is needed. Again,' the cause of the problem is that the bounds poorly approximate the real voltages present at the slow nodes in the circuit. To improve the upper bound on the output voltage requires improving the lower bound on the internal nodal voltages. The lower bound for a node on a side branch is equal to the voltage at the root of that side branch. For the slow nodes, this bound poorly approximates

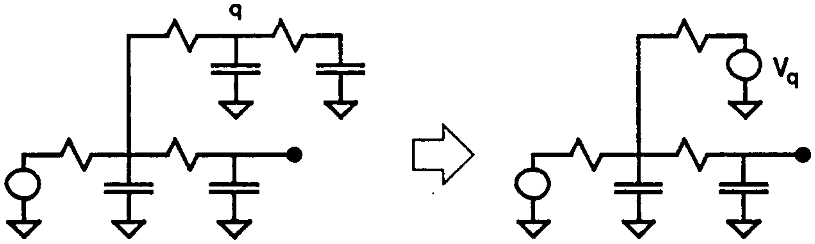


Figure C.l Replacing a capacitor with **a** voltage source.

the nodal voltage when the output is small.' The actual voltage of a slow node k is larger than the output by **a** factor of roughly R_{kk}/R_{ke}, while the bound is less than the output by **a** factor of R_{ke}/R_{ee}.

Since no simple method to improve the lower bound on V_k exists,+ an alternative approach that does not require this information is used. Instead of writing the output voltage as the sum of all capacitor currents, one (or possibly more) capacitor in the tree is replaced by a voltage source. The output voltage of this source has the same time dependence as the voltage on the capacitor it replaces; see Figure C.l. Thus, the output of the tree remains unchanged. Using this model of the network, the output voltage can be written as the sum of two terms, one caused by the voltage source, and the other caused by the remaining capacitor currents. An upper bound on these two terms will give an upper bound on the output voltage. Adding the voltage source decouples the slow nodes from the rest of the circuit.

An upper bound on the output voltage caused by the voltage source is simple to find. Since all voltages are less than or equal to 1, the voltage of the added source will always be $\leq$ 1. Assuming the source is placed at node v, the voltage it will cause at output node e is less than or equal to $\frac{R_{ve}}{R_{vv}}$, the dc response to a unit voltage source at node v.

[†] At least none that this author could find.

The voltage caused by the capacitor current is found by subtracting the dc voltage caused by the added source from each node, and setting the source voltage to zero. This modified network then has two ground connections, and an initial voltage that is not uniform. The dc voltage at node n caused by the source at node v is R_{kv}/R_{vv}. By superposition, the voltage involved in transient is therefore $1 - R_{kv}/R_{vv}$.

Output bounds for nodes in a network with multiple grounds are found using the same derivation used to bound voltages in an RC tree; only the definition of R_{ke} needs to change. For an network with multiple grounds, R_{ke}^* is defined to be V_e/i_k when all the other currents are zero; see Appendix B. The three time constants that determine the bounds are

$$\tau_P^* = \sum_k R_{kk}^* C_k, \qquad \tau_{Re}^* = \sum_k \frac{R_{ke}^{*\,2}}{R_{ee}^*} C_k, \qquad \tau_{De} = \sum_k \frac{R_{vv} - R_{vk}}{R_{vv}} R_{kc}^* C_k.$$

Using these time constants in Eq. (3.5) gives an upper bound on the voltage caused by the capacitor currents. Adding this voltage to the upper bound on the output voltage caused by the voltage source gives the desired upper bound on the output:

$$V_e \le \frac{R_{ve}}{R_{vv}} + \frac{\tau_{De}^*}{\tau_{Re}^*} e^{-t/\tau_P^*}. \tag{C.4}$$

The only remaining issue is the location of the voltage source. The source should be placed so nodes where $\tau_{Dk} >> \tau_{De}$ are removed from the network. The best placement depends on the output voltage range that is most important. Moving the voltage source closer to ground improves the bound on the initial transient, but makes the bounds for large t worse. If node j ‘has the largest τ_{Dj} in the network, then the voltage source should be placed at a node on the path from j to ground. Placing the source at a node with a resistance about τ_{Dj}/C_{total} is a good compromise between the conflicting requirements.

As an example, consider the RC tree shown in Figure ***C.2.*** Node 3 is the slowest node in the circuit; τ_{Dk} = 27. Replacing C_3 with a voltage source provides

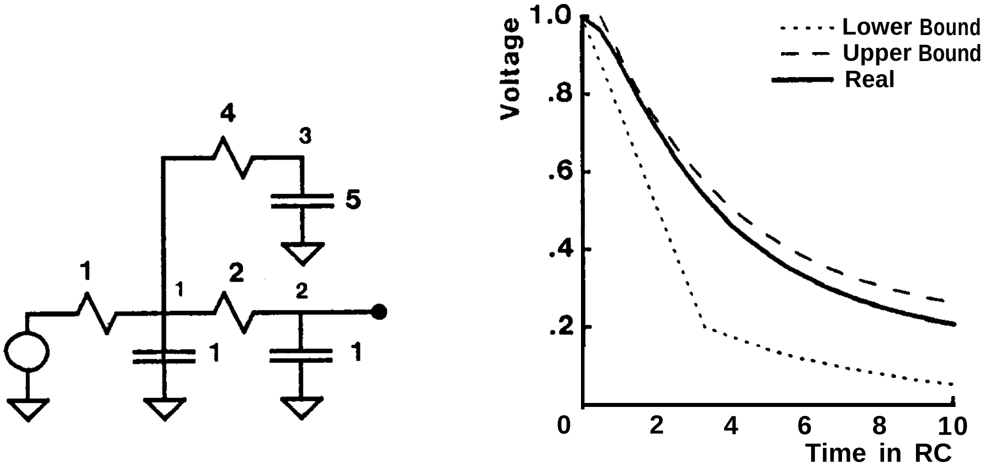


Figure C.2 Improved bounds for an RC tree with a pole-zero pair.

a method to improve the upper bound. An upper bound **on** the output voltage caused by the voltage source is R_{23}/R_{33}, or .2. The three time constants of the modified network are all close in value to each other, indicating the bound on the initial transient will be good. The improved upper bound is

$$V_2 \leq .2 + \frac{2.9}{3.1} \exp(--t/3.6).$$

and is shown in Figure C.2. Also shown in this figure is the improved lower bound for this network. The lower bound was improved by computing one additional set of time constants optimized for an output voltage of .5 of fullscale.

Appendix D

MOS BOUNDS DERIVATION

D.l Falling Transient

The bounds on U_k are

$$\frac{R_{ke}}{R_{ee}}(2V_e - V_e^2) \leq 2V_k - V_k^2 \leq \frac{R_{kk}}{R_{ke}}(2V_e - V_e^2),$$

and can be further approximated by

$$\left(1 - \sqrt{1 - R_{ke}/R_{ee}}\right)V_e \leq V_k \leq V_e/\left(1 - \sqrt{1 - R_{ke}/R_{kk}}\right).$$

These bounds provide the two additional time constants needed to generate the output bounds:

$$\tau_{\alpha e} = \sum_k R_{ke}C_k\left(1 - \sqrt{1 - R_{ke}/R_{ee}}\right); \qquad \tau_{\beta e} = \sum_k \frac{R_{ke}C_k}{\left(1 - \sqrt{1 - R_{ke}/R_{kk}}\right)}.$$

The differential inequality for g_e (Eq. (4.6)) can be solved, giving bounds on g_e. Using these bounds in Eq.(4.5) provides bounds on the output:

$$V_e(t) \geq \begin{cases} \dfrac{(\tau_{De} - t)}{\tau_{\beta e}}, & t \leq \tau_{De} - \tau_{\alpha e}; \\[2ex] \dfrac{\tau_{\alpha e}}{\tau_{\beta e}}\left[1 - \tanh\left(\dfrac{t - \tau_{De} + \tau_{\alpha e}}{\tau_{\alpha e}}\right)\right], & t \geq \tau_{De} - \tau_{\alpha e}; \end{cases} \qquad (D.1)$$

$$V_e(t) \leq \frac{\tau_{\beta e}}{\tau_{\alpha e}}\left[1 - \tanh\left(\frac{t}{\tau_{\beta e}} + \frac{1}{2}\ln\left(\frac{2\tau_{\beta e} - \tau_{De}}{\tau_{De}}\right)\right)\right].$$

The upper bound on the output voltage can be improved by observing that V is less than or equal to 1, and U decreases monotonically with time. The monotonicity of U_e gives

$$g_e(t) + (t - t')U_e \leq g_e(t').$$

Replacing $g_e($ t), and $g_e(t')$ with bounds, $\tau_{\alpha e}V_e$ and $g_{e_{max}}(t')$ respectively, gives an upper bound on V_e as a function of t':

$$\tau_{\alpha e}V_e + (t - t')f(V_e) \leq g_{e_{max}}(t'),$$

or

$$\tau_{\alpha e}V_e + (t - t')(2V_e - V_e^2) \leq \tau_{\beta e}\left[1 - \tanh\left(\frac{t'}{\tau_{\beta e}} + \frac{1}{2}\ln\left(\frac{2\tau_{\beta e} - \tau_{De}}{\tau_{De}}\right)\right)\right]. \qquad (0.2)$$

Setting the derivative of V_e with respect to t' equal to zero yields the optimal bound. Since $\frac{dg}{dt}$ is equal to --f(g), for the optimal solution $f(V_e)$ = $f(g_{e_{max}}(t')/\tau_{\beta e})$ **or** $\tau_{\beta e}V_e$ = $g_{e_{max}}$(t'). Until $V_e \leq \tau_{De}/\tau_{\beta e}$, the best value of t' is 0, since $g_e \leq \tau_{De}$. When V_e is below the cutoff, the optimal t' becomes difficult to determine exactly. Setting t' in this region to $t - t_o$, where $V_e(t_o)$ = $\tau_{De}/\tau_{\beta e}$, approximates the exact solution and yields the following improved upper bound on V_e:

$$V_e(t) \leq \begin{cases} 1, & t \leq \tau_{De} - \tau_{\alpha e} \\ \left(1 + \frac{\tau_{\alpha e}}{2t}\right)\left(1 - \sqrt{1 - \frac{4t\tau_{De}}{(2t + \tau_{\alpha e})^2}}\right), & \tau_{De} - \tau_{\alpha e} \leq t \leq \tau_{\beta e}\left[\frac{\tau_{\beta e} - \tau_{\alpha e}}{2\tau_{\beta e} - \tau_{De}}\right] \\ \left[1 - \tanh\left(\frac{t}{\tau_{\beta e}} - \frac{\tau_{\beta e} - \tau_{\alpha e}}{2\tau_{\beta e} - \tau_{De}} + \frac{1}{2}\ln\left(\frac{2\tau_{\beta e} - \tau_{De}}{\tau_{De}}\right)\right)\right], & t \geq \tau_{\beta e}\left[\frac{\tau_{De} - \tau_{\alpha e}}{2\tau_{\beta e} - \tau_{De}}\right] \end{cases} \qquad (D.3)$$

D.2 Rising Transient

For a rising transient, $1 - U$ replaces U, and $1 - V$ replaces V in the bounds derivation. The bounds on $1 - U_e$ are

$$\frac{R_{ke}}{R_{ee}}(1 - V_e)^2 \leq (1 - V_k)^2 \leq \frac{R_{kk}}{R_{ke}}(1 - V_e)^2.$$

Taking the square root of these bounds gives the desired bounds on the nodal voltages:

$$\sqrt{\frac{R_{ke}}{R_{ee}}}(1-V_e) \leq (1-V_k) \leq \sqrt{\frac{R_{kk}}{R_{ke}}}(1-V_e).$$

These bounds set the values of the two bounding time constants,

$$\tau_{\alpha e} = \sum_k \frac{R_{ke}^{\frac{3}{2}} C_k}{R_{ee}^{\frac{1}{2}}}; \qquad \tau_{\beta e} = \sum_k \sqrt{R_{kk} R_{ke}} C_k$$

The resulting bounds on g_e are

$$\frac{\tau_{\alpha e}^2}{t + \tau_{\alpha e}^2/\tau_{De}} \leq g_e \leq \frac{\tau_{\beta e}^2}{t + \tau_{\beta e}^2/\tau_{De}},$$

which give the following bounds on the output voltage:

$$1 - \left(\frac{\tau_{\beta e}}{\tau_{\alpha e}}\right)\frac{\tau_{\beta e}}{t + \tau_{\beta e}^2/\tau_{De}} \leq V_e \leq 1 - \left(\frac{\tau_{\alpha e}}{\tau_{\beta e}}\right)\frac{\tau_{\alpha e}}{t + \tau_{\alpha e}^2/\tau_{De}}. \qquad (D.4)$$

Again these simple bounds can be improved by using additional constraints on U_e. In analogy with Eq. (D.2), using monotonicity gives

$$\tau_{\alpha e}(1 - V_e) + (t - t')(1 - V_e)^2 \leq \frac{\tau_{\beta e}^2}{t + \tau_{\beta e}^2/\tau_{De}}. \qquad (D.5)$$

This inequality can be rearranged to provide an upper bound on $1 - V_e$ as a function of t'. Choosing t' to minimize the bound improves the lower bound on V_e:†

$$V_e(t) \geq \begin{cases} 0, & t \leq \tau_{De} - \tau_{\alpha e}; \\ 1 - \frac{\tau_{\alpha e}}{2t}\left(\sqrt{1 + 4t\tau_{De}/\tau_{\alpha e}^2} - 1\right) \\ 2t, & \tau_{De} - \tau_{\alpha e} \leq t \leq \frac{(\tau_{\beta e} - \tau_{\alpha e})\tau_{\beta e}}{\tau_{De}}; \\ 1 - \frac{2\tau_{\beta e} - \tau_{\alpha e}}{t + \tau_{\beta e}^2/\tau_{De}}, & \frac{\text{b@} - \tau_{\alpha e})\tau_{\beta e}}{\tau_{De}} \leq t. \end{cases} \qquad (D.6)$$

† This minimization is easier than it looks. The trick is to define τ' as $\tau_{\alpha e}/(1 - V_e)$, and find the optimal t' in terms of τ'. Although the resulting t' will depend on V_e ($t' = [\tau' - \tau_{\beta e}^2/\tau_{De} + t]/2$), the output voltage can still be found explicitly.

The lower bound on $1 - V_e$ is improved by using the constraint $1 - U_e \leq 1$ to improve the lower bound on g_e. The improved bound on the output voltage is

$$V_e(t) \leq \begin{cases} 1 - \dfrac{\tau_{De} - t}{\tau_{\beta e}}, & t \leq \tau_{De} - \tau_{\alpha e}; \\ 1 - \dfrac{\tau_{\alpha e}^2}{\tau_{\beta e}(t - \tau_{De} + 2\tau_{\alpha e})}, & t \geq \tau_{De} - \tau_{\alpha e}. \end{cases} \qquad (D.7)$$

REFERENCES

[AD82], H. Al-Hussein and R. Dutton, "Path Delay Computation for Integrated Systems," ***Internal*** Conference on Circuits and Computers, ***pp. 426-430,*** Sept. 1982.

[CG75], B. Chawla, H. Gummel, and P. Kozak, "MOTIS — An MOS Timing Simulator," *IEEE* Transactions on Circuits and Systems, Vol. CAS-22 No. ***12, pp. 901-910,*** Dec. 1975.

[CL75], L. Chua and P. Lin, Computer Aided Analysis of Electronic Circuits: Algorithms ***and*** Computational Techniques. Prentice-Hall, 1975.

[CR67], R. Crawford, MOSFET ***in*** Circuit Design, McGraw-Hill, Section 4.2, 1967.

[El48], W. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers," Journal of Applied Physics, Vol. ***19, pp. 55-63, Jan. 1948.***

[FH77], S. Fan et al., "MOTIS-C: A New Circuit Simulator for MOS LSI Circuits," International Symposium on Circuits and Systems, pp. ***700-703, 1977.***

[GS69], P. Grayand C. Searle, Electronic Principles, John Wiley & Sons, Inc., Section 1.4, 1969.

[HD83], M. Horowitz and R. Dutton, "Resistance Extraction from Mask Layout Data," IEEE Transactions on Computer-Aided Design of *ICs,* Vol. CAD-2, No. 3, pp.1 45-150, July 1983.

[HS81], G. Hachtel and A. Sangiovanni-Vincentelli, "A Survey of Third-Generation Simulation Techniques," Proceedings of ***the IEEE, Vol.*** 69 No. 10, pp. 1264-1280, Oct. 1981.

[Jo83], N. Jouppi, "TV: An nMOS Timing Analyzer," Third CalTech Conference on VLSI, pp. 71-85, March 1983.

[KC66], T. Kirkpatrick and N. Clark, "PERT as an Aid to Logic Design," IBM Journal *of* Research *and* Development, *pp.* 135-141, March 1966.

[LS82], E. Lel arasmee and A. Sangiovanni-Vincentelli, "Relax: A New Circuit Simulator for Large Scale MOS Integrated Circuits," Proceeding *of the 19th* Design Automation Conference, *pp.* 682-690, June 1982.

[McW80], T. McWilliams, "The SCALD Timing Verifier: A New Approach to Timing Constraints in Large Digital Systems," International Symposium *on* Circuits *and Systems, pp. 415-423,* May 1980.

[MC80], C. Mead and L. Conway, Introduction to *VLSI* Systems, Addison-Wesley, 1980

[MK77], R. Mu1ler and T. Kamins, *Device Electronics for Integrated Circuit* Design, Wiley & Sons, Inc., Section 8.1, 1977.

[Mo82], M. Monachino, "Design Verification System for Large Scale LSI Designs," *IBM Journal of* Research and *Development, Vol. 26 No.* 1, *pp.* 89-99, Jan. *1982.*

[Na75], L. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits," University of California, Berkeley, ERL-520, May 1975.

[OM83], K. Okazaki, T. Moriya, and T. Yahara, "A Multiple Media Delay Simulator for MOS LSI Circuits," ***Proceeding of the 20th Design Automation Conference, pp. 279-285, June 1983.***

[Ou83], J. Ousterhout, "Crystal: A Timing Analyzer for nMOS VLSI Circuits," Third CalTech Conference on VLSI, pp. 57-69, March 1983.

[Pe82], D. Peterson, "IC CAD / Simulation," IEEE Custom Integrated Circuit Conference Proceeding, pp. 248-251, 1982.

[PR81], P. Penfield and J. Rubinstein, "Signal Delay in RC Tree Networks," Proceeding ***of the*** 18th Design Automation Conference, ***pp. 613-617, June 1981.***

[PS72], D. Pillingand J. Skainik, "A Circuit Model for Predicting Transient Delays in LSI Logic Systems," 6th ***Asilomar Conference on Circuits Systems*** and Computers, ***pp. 424-428, 1972.***

[PS73], D. Pillingand H. Sun, 'Computer-Aided Prediction of Delays in LSI Logic Systems,', ***Proceeding of the*** 10th Design Automation ***Conference, pp. 182-186, June 1973.***

[Pu82], R. Putatunda, "Auto-Delay: A program for Automatic Calculation of Delays in LSI/VLSI Chips, ***Proceeding of the 19th Design Automation Conference, pp. 616-621, June 1982.***

[Ra73], N. Rabbat, "A Computer-Aided Approach for the Prediction of LSI Transient Circuit Performance," 7th ***Asilomar Conference on Circuits Systems and Computers, pp. 136-142, 1973.***

[RP83], J. Rubinstein, P. Penfield, and M. Horowitz, "Signal Delay in RC Tree Networks," IEEE ***Transactions on Computer-Aided Design of ICs, Vol. CAD-2, No. 3, pp. 202-211, July 1983.***

[RR78], A. Ruehli, R. Rabbat, and H. Hsieh, "Macromodelling — an Approach for Analysing Large-Scale Circuits," Computer-Aided Design, Vol. 10 No. 2, pp. 121-129, March 1978.

[SA78], L. Scheffer and R. Apte, "LSI Design Verification Using Topology Extraction," 12th ***Asilomar Conference on Circuits Systems*** and ***Computers, pp. 149-153, Nov. 1978.***

[SK83], R. Saleh, J. Kleckner, R. Newton, "Interated Timing Analysis in SPLICE1," ***International Conference on Computer Aided Design 83, pp. 139-140, Sept. 1983.***

[TA65], R. Thornton et al, Multistage ***Transistor Circuits,*** John Wiley & Sons, Inc., 1965.

[WJ73], W. Weeks et al., "Algorithms for ASTAP — A Network Analysis Program," ***IEEE Transactions on Circuit Theory, Vol. CT-20, pp. 628-634, Nov. 1973.***

[Wy82], J. Wyatt, "Monotone Behavior of Nonlinear RC Meshes," MIT VLSI Memo 82-128, Nov. 1982, unpublished.