Delay Modeling and Timing of Bipolar Digital Circuits †

D. G. Saab, A. T. Yang and I. N. Hajj

Coordinated Science Laboratory
and Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
1101 W. Springfield Avenue,
Urbana, Illinois 61801.

ABSTRACT

In this paper an approach for timing simulation of bipolar ECL digital circuits is descibed. The approach is based on the development of a switch-level model of the transistor and on the representation of the circuit by a switch-graph. The circuit is partitioned into subcircuits, and symbolic logic expressions are then generated, which represent the logic states of the nodes in terms of subcircuit inputs and initial conditions. Timing information is computed using an analytical delay model which relates outputs of a subcircuit to its inputs waveforms. The model includes the effects of the transistor SPICE parameter model as well as the circuit parameters. The combination of the switch-level graph model and the delay model provides fast and accurate timing simulation of ECL circuits. In addition, the switch-graph model provides a unified way for simulating BIMOS circuits.

1.1. Introduction

Almost all reported work in the area of logic and timing simulation of VLSI circuits has centered on MOS circuits. For high performance applications, however, bipolar digital circuits are still in wide use today [1-5]. The design of these circuits depends on relatively accurate circuit modeling and simulation to verify the performance of the design. Circuit level simulation, such as SPICE [6], provides an accurate performance evaluation for given input sequences. But, such simulation is expensive for large-scale circuits and is only practical for relatively small circuits. A need exists for verifying the logic and timing behavior of the design directly from transistor interconnections and characteristics without having to pay the price of detailed circuit simulations.

The approach followed in this paper is based on developing a switch-level model of the bipolar transistor and on representing the transistor circuit by a labeled weighted graph, similar in some respects to the graph representation of MOS digital circuits [7-11]. Path finding algorithms are then used to automatically partition the graph into subgraphs, which correspond to subcircuits or gates, and to extract logic expressions that define the states of the nodes in the subcircuits in terms of subcircuit inputs and initial conditions [13]. In most cases, the logic expressions are reducible to standard Boolean expressions, and thus can be used, if needed, to automatically generate an equivalent logic block diagram description of the circuit; but, such a diagram representation is not essential for timing simulation. An analytical delay model of bipolar ECL circuits is developed. This delay model is used in conjunction with the logic expressions to perform the timing analysis. The major advantage of using explicit expressions is that they can be executed efficiently and occupy much less memory space than table lookup models [12]. This delay model handles complex digital ECL circuits with multiple inputs and/or multiple levels. The important effects of input slew rate and loading are incorporated as well. This analytical delay model is shown to predict the delay time with less than 15 percent error as compared to SPICE[6] with over two orders of magnitude speed improvement.

In the next section we briefly describe the switch-level bipolar transistor model and the graph representation and logic expression extraction of ECL circuits. A more complete description of the method is given in [13]. The derivation of the delay function is explained in section 3. Implementation of the approach is described in Section 4 along with examples and comparison with SPICE.

1.2. Symbolic logic representation of ECL circuits

The method for generating logic expressions from transistor circuit description has been explained in [13]. For the sake of completeness a brief overview is given.

We consider circuits formed by interconnections of bipolar (NPN) transistors. The transistor has three terminals: base, emitter and collector; and for digital applications is assumed to act as a switch. Distinction is made between the emitter and the collector in order to differentiate between the logic expressed by the current (flowing or not flowing) and that expressed by the voltage (low or high). Diodes act as voltage level shifters, and thus will be assumed not to affect the logic levels at their terminal nodes. Resistors act as pull-up or pull-down devices.

The transistor network is represented by constructing a corresponding switch-graph model similar to the switch-level model used in the logic simulation of MOS circuits [7-11]. The switch-graph G(V,E), where V is the set of vertices and E the set of edges, is a labeled, edge-weighted, vertex-weighted, undirected graph, which is constructed from the circuit according to simple rules derived from the circuit topology. As an example, Figure 1.1 (a) shows an exclusive or/nor gate circuit and Figure 1.1 (b) shows its corresponding labeled switch-graph model, where each transistor is represented by an edge with a logic label and strength γ_2, which indicates the relative conductance when ON.

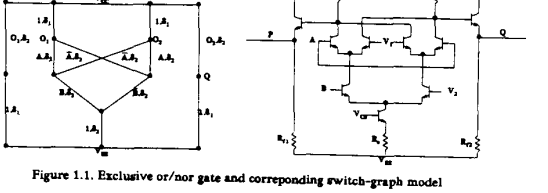

Figure 1.1. Exclusive or/nor gate and correponding switch-graph model

† This work was supported in part by the Semiconductor Research Corporation contract 84-06-049, and in part by the Joint Services Electronics Program (U.S.Army, Navy, and Air Force) contract N00014-84-C-0149.

The load resistor R_{C1} is represented by an edge with label 1 and strength γ_1, where $\gamma_1 < \gamma_2$.

Similar to what is done in MOS switch-level modeling [7-11], the nodes are classified into two sets: input nodes and 'storage' nodes. Input nodes include primary inputs, ground and supply voltages, such as V_{CC} or V_{EE}. All other nodes are considered storage nodes. Input nodes have known states and are considered stronger than storage nodes; i.e., their states are not influenced by any other nodes in the circuit. Storage nodes, on the other hand, can have their states influenced by input nodes as well as other storage nodes. Storage nodes are assigned different strength levels depending on their relative capacitance. such modeling provides a unified model when simulating BIMOS circuits.

As is seen in the example in Figure 1.1, the switch-level graph representation automatically partitions the circuit into subcircuits connected only at V_{CC} and V_{EE}. These subcircuits are equivalent to the channel-connected subcircuits in MOS circuit designs. Because of the nature of digital bipolar designs and the absence of 'transmission gates', it is possible to extract Boolean expressions at subcircuit outputs (nodes O_1, O_2, P, Q in Figure 1.1) in terms of subcircuit inputs (branch labels)[13]. In Figure 1.1, for example,

$$O_1 = AB \vee \overline{A}\,\overline{B}$$

$$O_2 = \overline{A}B \vee A\overline{B}$$

$$Q = \overline{O_2}$$

$$P = \overline{O_1}$$

1.3. Delay modeling

1.3.1. Delay definitions and dependency

In this section we describe the method for deriving a set of analytical expressions describing the delay at the output nodes of a subcircuit obtained by the partitioning described in Section 2 including the emitter followers. These expressions are used in conjuction with the symbolic expressions to check the circuit limitation and critical performance.

Definition: The delay τ_d (Figure 1.2) associated with an output node is the time interval between the midpoints of the input transition and the output transition.

To identify the parameters that affect τ_d, detailed SPICE [6] simulations were performed on the basic ECL inverter circuit shown in Figure 1.3. The results were examined and τ_d was found to depend strongly on parameters given in Table 3.1.

τ_f	Forward transit time
r_b	Transistor base resistace
R_c	Pullup resistance
C_{jc}	Base-collector depletion capacitance
C_{cs}	Collector to substrate depletion capacitance
C_{root}	Capacitance of pull-down transistor current source
T_{slew}	Input slew rate
$C_{je}f$	Depletion capacitace of the emitter follower
R_f	Pulldown resistance of the emitter follower
C_{load}	Loading capacitance

Table 3.1. Delay-sensitive parameters.

The switching mechanisms for ECL circuits are very complex for several reasons. First, multiple nonlinear capacitances make switching approximation intractable. Second, overall switching must be described in terms of several switching events. Each event may not neccessarily be completed before another begins. Thirdly, there are many **delay—sensitive**

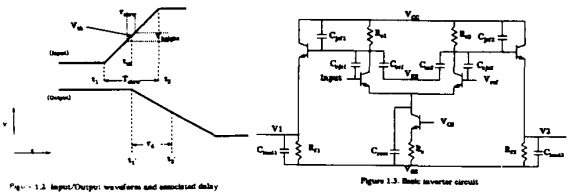

Figure 1.3. Basic inverter circuit

parameters. Finally, associated with each input switching, there are two complementary output switchings and they may have different delays.

We have made two important approximations which enable us to derive adequate delay expressions. First, since the voltage change across each capacitance is known, we replace each nonlinear capacitance by an average capacitor value. Second, we identify and decouple the switching events and approximate the overall delay, τ_d by.

$$\tau_d \approx \tau_{in} + \tau_{slew} + T_{ext} \qquad (1)$$

where τ_{in} is denoted as the *intrinsic delay*, τ_{slew} is the *slew rate delay* and T_{ext} is the *extrinsic delay*.

The justification for Eq. (1) is in fact quite physical for a basic inverter. The extension of our model to more complex circuits consisting of multiple inputs and multiple levels are discussed in Section 3.5. For a basic inverter shown in Figure 1.3, when an input is applied, time is needed to switch the steering current from the "ON" differential transistor to the other "OFF" differential transistor. Once the current is steered, it charges or discharges the associated extrinsic capacitances. Since the switching threshold for the differential transistors is quite small, different input slew rates incur linear output shifts.

1.3.2. Intrinsic Delay

Definition: Intrinsic delay τ_{in} is the time for the mechanism of extracting (injecting) *half* of the total stored excess minority charges, Q_f, from (to) the base of the ON (OFF) differential pair transistor.

Based on the forward active model given in [14] a transient circuit model for an ECL inverter is shown in Figure 1.4. This model is used in the derivation of the analytical expressions for τ_{in}.

To simplify the derivation, τ_{in} is expressed as the sum of $\tau_{in(ideal)}$, denoting the delay without the effect of the C_{root}, and $\tau_{C_{root}}$, denoting the added delay due to C_{root}.

1.3.2.1. Intrinsic Delay for falling input step

Consider the circuit model of the basic inverter shown in Figure 1.4. The current through a capacitor is given by $i(t) = \frac{dq}{dt}$. If $i(t)$ is approximated by a constant i_{avg}, then $\Delta T = \frac{\Delta q}{i_{avg}}$, where ΔT is the time it takes to change the charge across the capacitor by an amount of Δq, which results in corresponding change of capacitpor voltage, ΔV. From the definition of τ_{in}, and because ΔT does not include the effect of

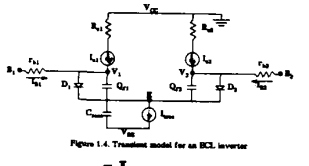

Figure 1.4. Transient model for an ECL inverter

C_{root}, then $\tau_{in(ideal)} = \frac{1}{2}\frac{\tau_f I_{tree}}{i_{avg}}$, where $\Delta q = \tau_f I_{tree}$. Referring to

Figure 1.4, for large β, $I_{tree} = I_{C1}+I_{C2}$ and $I_{B1}+I_{B2} = 0$. If these conditions are satisfied, $V_1 = V_2 = -1.75\phi$, where $\phi \approx 0.8V$, and $i_{avg} = I_{B1} = \frac{V_{B1} - V_1}{r_b} = \frac{1}{4}\frac{\phi}{r_b}$. Substituting i_{avg} into $\tau_{in(ideal)}$ expression and replacing I_{tree} by $\frac{\phi}{R_c}$ yields: $\tau_{in(ideal)} = 2\tau_f\frac{r_b}{R_c}$. For ideal intrinsic delay, $\tau_{rise(ideal)}$ is equal to $\tau_{fall(ideal)}$ where the subscript, *rise* and *fall* stand for rising and falling output, respectively.

To compute the effects of C_{root} on the intrinsic delay, note that due to the clamping effects of both D1 and D2, V_E will have a step drop of $\frac{3}{4}\phi$. Therefore, C_{root} needs to be discharged for an amount of $Q_{root} = C_{root}\frac{3}{4}\phi$. As soon as the input falls, V_1 and V_E cannot change instantly. As a result, a *peak* current, $I_{peak} = \frac{\phi}{r_b}$, flows out of B1 which will discharge both Q_{f1} and Q_{root}. While Q_{f1} and Q_{root} are being discharged, V_1 and V_E drop slowly. This, in turn, causes a drop of I_{peak}. Taking half of I_{peak} is a valid average for this falling current, $I_{qf} = \frac{1}{2}I_{peak} = \frac{\phi}{2r_b}$. The expression for τ_{fall} can be written as

$$\tau_{fall} = \tau_{in(ideal)} + \tau_{root} \qquad (2)$$

where τ_{root} is the extra time it takes to discharge C_{root}. The discharging current of C_{root} is the sum of I_{qf} and I_{tree}. Thus $\tau_{root} = \frac{Q_{root}}{\frac{\phi}{R_c} + \frac{\phi}{2r_b}}$. Substituting appropriate values into (2) yields:

$$\tau_{fall} = 2\tau_f\frac{r_b}{R_c} + \frac{3C_{root}r_b R_c}{4r_b + 2R_c}. \qquad (3)$$

To derive τ_{rise}, note that as C_{root} is being discharged for a swing of $\frac{3}{4}\phi$, Q_{f1} is discharged by an amount of $\tau_{root} I_{qf}$. After C_{root} has completed its discharge, Q_{f1}, will then be discharged by $\frac{\phi}{4r_b} = I_{avg}$. Therefore, $\tau_{rise} = \tau_{root} + \frac{\frac{1}{2}\tau_f I_{tree} - \tau_{root} I_{qf}}{\frac{\phi}{4r_b}}$ which when τ_{root} and I_{qf} are replaced by there respective value yields:

$$\tau_{rise} = 2\tau_f\frac{r_b}{R_c} - \frac{3C_{root}r_b R_c}{4r_b + 2R_c} \qquad (4)$$

1.3.2.2. Intrinsic delay for rising input step

Considering Figure 1.3 and an input with a rising step from -2ϕ up to $-\phi$, a similar analysis to that of the falling steps results in exactly the same τin(ideal). However, C_{root} now has negligible effect on τ_{in} since C_{root} needs to be charged for a swing of only $\frac{1}{4}\phi$ (as opposed to $\frac{3}{4}\phi$ from the falling input step). Hence,

$$\tau_{rise} \approx \tau_{fall} \approx \tau_{in(ideal)}. \qquad (5)$$

Equations (3), (4) and (5) constitute a set of analytical expressions for the intrinsic delay of a single-input, single-level basic gate.

1.3.3. Input slew rate

In Figure 1.1, T_{slew} $(t_2 - t_1)$ is defined as a measure of the slew rate of the input signal. V_{th} and t_{th} are the input threshold voltage and its corresponding threshold crossing time. The output transition begins when the input voltage reaches the threshold voltage, V_{th}. Consequently, t_1 equals t_{th}. Given t_1, t_2 can be evaluated from the delay model. The output waveform can then be represented by a straight line with slope defined by t_1 and t_2.

The delay operator derived so far assumes an ideal input step. Strictly speaking, time is needed for the input to cross the threshold voltage, V_{height}, which is denoted by τ_{slew} and equals to $\frac{V_{height}}{\phi}T_{slew}$. For a differential pair, it is well known that V_{height} is approximately 120 mV. Thus,

$$\tau_{slew} = 0.15T_{slew} \qquad (6)$$

1.3.4. Extrinsic delay for single-input single-level circuit

Definition: The extrinsic delay, T_{ext}, is the time for the mechanism of charging or discharging all the associated capacitances through half a logic swing, $\frac{1}{2}\phi$, at the output nodes.

1.3.4.1. Emitter follower charge-up

The forward biased model [14] for the emitter follower results in the equivalent circuit shown in Figure 1.5. where C_t is a simple sum of all the parasitic capacitances at the base of the

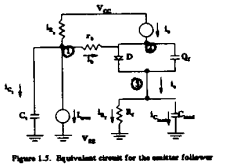

Figure 1.5. Equivalent circuit for the emitter follower

emitter follower (node 1 in Figure 1.5.) and is given by $C_t = C_{cs} + 2C_{jc} + C_{jcf}$, where the factor of two is introduced to account for the *Miller effect* at the inverting output. When the current is steered away, an excess (transient) base current, $i_b(t)$, flows in to charge Q_f of the emitter follower. The resultant excess charge, Q_{excess}, is reflected as excess emitter current used to charge up the load and $Q_{excess} = \frac{i_b T_{rise}}{2}$, where i_b is the average excess base current for the period of interest, T_{rise}. From basic charge-control equations of a bipolar transistor [14], we can then write $i_c = \frac{i_b T_{rise}}{2\tau_f}$ and $i_e = i_b(\frac{T_{rise}}{2\tau_f} + 1)$. Note the following, as the output rises, the current through the loading resistor, R_f, also increases. However, to simplify the analysis, this increase is assumed to be negligible thus $i_e \approx i_{C_{load}} \approx \frac{C_{load}\psi_e}{T_{rise}}$, where ψ_e stands for the total voltage change at the output. Equating the two expression for i_e, results in a quadratic equation of $T_{rise}\frac{i_b}{2\tau_f}T_{rise}^2 + i_b T_{rise} - C_{load}\psi_e = 0$. From Fig. 1.5, for emitter follower charge-up, $i_b = i_{R_c} - i_{C_t}$ and $i_{C_t} = \frac{C_t\psi_b}{T_{rise}}$. Note that ψ_b stands for the voltage change at the base of the emitter follower (node 1 in Figure 1.5). substituting in the last quadratic equation yields $\frac{i_{R_c}}{2\tau_f}T_{rise}^2 - (\frac{C_t\psi_b}{2\tau_f} - i_{R_c})T_{rise} - (C_t\psi_b + C_{load}\psi_e) = 0$. From the definition of T_{ext}, ψ_e is equal to $\frac{1}{2}\phi$ and ψ_e is set to $\frac{5}{8}\phi$ to

account for the extra transient base voltage drop. On the other hand, i_{R_e} can be computed as an average value $i_{R_e} = \frac{i_{peak} + i_{mid}}{2}$, where i_{peak} and i_{mid} are the boundary current values during the period of interest, T_{rise}. Therefore, $i_{R_e} = \frac{\frac{\phi}{R_c} + \frac{3\phi}{8R_c}}{2}$. Substituting this expression for i_{R_e} and the values of ψ_e and ψ_b into the previous quadratic equation, we can solve the extrinsic delay, T_{rise}:

$$T_{rise} = 0.455R_cC_t - \tau_f + \sqrt{0.207(R_cC_t)^2 + 0.911\tau_fR_cC_t + 1.457\tau_fR_cC_{load} + \tau_f^2}. \quad (7)$$

The effects of the loading resistance are incorporated by replacing i_e by $i_{C_{load}} + \frac{1}{4}\frac{\psi_e}{R_f}$, where the second term is an approximation for the average increase of current through R_f. A more accurate expression for T_{rise} is then derived as

$$T_{rise} = 0.455R_cC_t - \tau_f + f_1 + \sqrt{0.207(R_cC_t)^2 + 0.911\tau_fR_cC_t + 1.457\tau_fR_cC_{load} + \tau_f^2 + f_2}$$

$$f_1 = \frac{0.368\tau_fR_c}{R_f}. \quad (8)$$

$$f_2 = \frac{\tau_fR_c}{R_f}(0.331C_tR_c - 0.727\tau_f + \frac{0.132\tau_fR_c}{R_f}).$$

1.3.4.2. Emitter follower discharge

Consider the equivalent circuit in Fig. 1.5 with I_{tree} is steered to node 1. The mechanisms governing the emitter follower discharge can be explained qualitatively from the following three cases. First case is when $C_t << C_{load}$: Q_f is quickly discharged by I_{tree}. The emitter follower is quickly turned off. If τ_{load} is the time constant at the output, then $\tau_{fall} \approx \tau_{load}$. Second case is when $C_t >> C_{load}$: The emitter follower stays on throughout the transition. If $\tau_{R_eC_t}$ is the time constant at the base, then $\tau_{fall} \approx \tau_{RtCt}$. Finally when $C_t \approx C_{load}$: τ_{fall} depends strongly on both time constants, $\tau_{R_eC_t}$ and τ_{load}.

An expression for T_{fall} is formulated and verified by extensive SPICE simulation. The discharge mechanism is then found out to be

$$T_{fall} \approx \frac{\tau_{R_eC_t}(1 + 0.9\alpha), + \tau_{load}\alpha^2}{1 + \alpha^2} \quad (9)$$

where $\tau_{R_eC_t} = 0.693R_cC_t$ is the discharging time constant seen at the base of the emitter follower, and $\tau_{load} = R_fC_{load}\ln(\frac{-V_{EE} - \phi}{-V_{EE} - 1.5\phi}) = 0.1R_fC_{load}$ is the discharging time constant seen at the output of the ECL gate with V_{EE} equals -5 V, and $\alpha = \frac{\tau_{load}}{\tau_{R_eC_t}}$. When $C_t << C_{load}$, equation (9) yields $T_{fall} \approx \tau_{load}$. When $C_t >> C_{load}$, $T_{fall} \approx \tau_{R_eC_t}$.

1.3.5. Extensions to Multi-Input/Level ECL Structures

Figure 1.6 shows a logic realization of $(A \vee B \vee C) \wedge (D \vee E)$ and its complement. Two transistors have their emitters tied together to form an "OR" logic and a stack logic realization is used to generate two levels of functional complexity (AND) per bias current. The timing behaviors for these complex ECL circuits can be obtained by extending our basic delay model derived above.

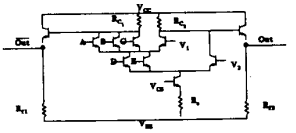

Figure 1.6 An example of multi-level logic

A simple analogy can be used to illustrate the delay through a complex circuit. Consider the *electrons* moving through *transistors* as *water molecules* flowing through *valves*. Associated with each valve there is an opening time and a closing time which is denoted as τ_{valve}. At a junction of valves, molecules naturally seek passage through the opened valves. At an opened valve, molecules must wait in line with other fellow molecules to pass through the narrow passage provided. We denote this waiting delay as $\tau_{transit}$. A path from the bottom valve to the top valve is called a *branch*. A junction of valves is called a *node*.

1.3.5.1. Delay for single-input, multi-level circuit

Definition: A transistor t_j is in level i if the proper mid voltage swing at the base of t_j is equal to $-(i+\frac{1}{2})\phi$ (level 1 is denoted by top level).

From the above definition and the valve anology $\tau_{valve}(i)$ and $\tau_{transit}(i)$ are given $\tau_{valve}(i) = \tau_{in}(i)$ and $\tau_{transit} = \gamma\tau_f$, where $\tau_{in}(i)$ is the intrinsic delay associated with the i^{th} level, and where γ is an empirical number and is determined by SPICE to be 0.70.

1.3.5.1.1. Single input switching

In this section we consider that only one input (all other inputs remain HIGH) is switching from HIGH (LOW) to LOW (HIGH) in a stacked circuit. The total intrinsic delay is indicated as τ_{IN} and can be found from $\tau_{in}(i)$ and $\tau_{transit}$. If the switching input is at level i, then $\tau_{IN} = \tau_{valve}(i) + (i-1)\tau_{transit}$ and is equal to

$$\tau_{IN} = \tau_{in}(i) + \gamma(i-1)\tau_f. \quad (10)$$

For a stacked circuit, since the extrinsic delay ,T_{ext}, is associated only with the top level, the output threshold crossing time is given by $t_2' = t_1 + \tau_{IN} + T_{ext}$, and the transistion at the output starts at $t_1' = t_{th}(i)$, where t_1, t_2 and t_{th} are defined in Fig. 1.1.

1.3.5.1.2. Multiple inputs from low to high

Consider the case when multiple inputs change from LOW to HIGH, while remaining inputs stay HIGH. Let $t_2'(i)$ denote the output threshold crossing time for an *intrinsic* circuit due to an input change at *level* i. From (10) we define

$$t_2'(i) = t_{th}(i) + \tau_{IN}(i) = t_{th}(i) + \tau_{in}(i) + \gamma(i-1)\tau_f. \quad (11)$$

τ_{IN} for the stacked circuit is mainly limited by the *slowest* $t_2'(k)$ at the k^{th} level. Hence,

$$t_2' = T_{ext} + \max(t_2'(i)). \quad (12)$$

$$t_1' = \max(t_{th}(i)) \quad (13)$$

1.3.5.1.3. Multiple inputs from high to low

Consider the case when more than one input are switching from HIGH to LOW. From the *valve analogy*, molecules can be redirected to other branches by turning off *any* valve associated with the current branch. Thus, an estimate of the delay is obtained by choosing the *fastest* $t_2'(i)$; Therefore,

$$t_2' = T_{ext} + \min(t_2'(i)). \quad (14)$$

and the output will begin to switch as soon as any input begins to switch at

$$t_1' = \min(t_{th}(i)). \tag{15}$$

Equation (14); however, provides the worst-case timing prediction for multiple inputs that are switching simultaneously. It assumes that during switching *all* the molecules are redirected to the other branch from valve(**k**), at the k^{th} level, which turns off faster than any other valve. Strictly speaking, there can be an *overlapping time* between each of the other valves and valve(**k**). Therefore, the molecules can be redirected to other branches through other valves as well. Let us denote $\tau_{ovlap}(i)$, i=1,2,···,L−1, as the overlap time between each individual valve along the branch to valve(**k**). L is the total number of levels. Note that $\tau_{ovlap}(i)$ ranges from 0 to $\tau_{in}(k)$. To account for this reduction of delay, we rewrite (40) as:

$$t_2' = T_{ext} + \min(t_2'(i)) - \sum_{i=1}^{L-1} \frac{\tau_{ovlap}(i)}{i(i+1)}. \tag{16}$$

Note that (16) is valid only when $\tau_{ovlap}(i)$ is decreasing in value for each increment of i. This can be done by simply sorting $\tau_{ovlap}(i)$. Equation (16) has the following characteristics:

(1) The number of level, L, is arbitrary.

(2) Any combination of simultaneously switching inputs from HIGH to LOW is allowed.

(3) Inputs can switch at different times with different slew rates.

1.3.6. Delay for multi-input, single-level circuit

1.3.6.1. Single input switching

Consider the case where only one input is switching while all other inputs are LOW. The timing model is identical to the one derived earlier for a basic gate except that multiples of C_{jc} (base-collector feedback capacitor) must be added to compute T_{ext}.

1.3.6.2. Multiple inputs from low to high

Consider the case when more than one input are switching from LOW to HIGH. Let $t_2'(n)$ denotes the output threshold crossing time for an intrinsic circuit due to an input change at *input* n. And

$$t_2'(n) = t_{th}(n) + \tau_{in}(n). \tag{17}$$

From the *valve analogy*, molecules can be directed to a branch by turning on *any* valve associated with the current branch. Thus, an estimate of the timing constraint is obtained by choosing the *smallest* $t_2'(m)$ for valve(**m**). However, such estimate is readjusted for transitions that overlapped. Let **N** equal the total number of parallel inputs associated with the node in question, then t_2' is given by:

$$t_2' = T_{ext} + \min(t_2'(n)) - \sum_{n=1}^{N-1} \frac{\tau_{ovlap}(n)}{n(n+1)} \tag{18}$$

Similarly, (18) is valid only when $\tau_{ovlap}(n)$ is decreasing in value for each increment of n. The output will begin to switch as soon as any input begins to switch. Therefore,

$$t_1' = \min(t_{th}(n)) \tag{19}$$

1.3.6.3. Multiple inputs from high to low

Consider the case when multiple inputs change simultaneously from HIGH to LOW. τ_{IN} for the multi-input circuit is mainly limited by the *slowest* intrinsic midpoint crossing time. Therefore,

$$t_2' = T_{ext} + \max(t_2'(n)) \tag{20}$$

The output begins its transition only after the latest input begins to switch, or

$$t_1' = \max(t_1'(n)) \tag{21}$$

1.3.7. Loading effect due to fanouts

Calculation of delays for a gate driving signal fanouts requires the total capacitance at its output nodes. In the previous sections, we have assumed this capacitance, C_{load} is known. Strictly speaking, C_{load} can be expressed as $C_{load} = C_{wire} + F_1 \cdot \sum_{fanout} C_{jc} + \sum_{fanout} C_{leakage}$, where C_{wire} is the wiring capacitance, C_{jc} is the base-collector depletion capacitance of the fanout differential transistor. F_1 is an empirical factor and set to 0.5 from SPICE. $C_{leakage}$ is the dynamic capacitance for the base leakage current of each fanout. For each fanout whose I_{tree} is switched from one transistor to the other due to the driving signal, we can express $C_{leakage}$ simply by

$$C_{leakage} = F_2 \cdot \frac{T_{ext} \cdot I_{leakage}}{\phi/2} \tag{22}$$

In the above equation, $I_{leakage}$ is the average base leakage current which starts to flow when the output is approaching its midpoint threshold. Therefore, throughout the entire extrinsic delay of the driver gate , we can assume $I_{leakage}$ only flows for a certain portion of time approximated by a constant percentage factor, F_2. Using SPICE, we obtain the best results if F_2 equals 0.3. From the intrinsic delay calculation we also know $I_{leakage} = i_{avg} \cdot \frac{\tau_{in} - 0.15\, T_{slew}}{\tau_{in}}$, where i_{avg} is the average current.

In our timing simulator, C_{load} is evaluated from the following algorithm:

```
procedure C_load (driver parameter, fanouts parameter)
begin
    C_load = C_wire
    Evaluate T_ext of the driver with C_load = 0
    for Count = 1 to (Total number of fanouts)
                if Current tree is steered then
                begin
                Evaluate τ_in, i_avg of fanout (Count)
                Evaluate C_leakage(Count) (eq. 22)
                C_load = C_load + C_leakage(Count)
            end if
            C_load = C_load + F_1 · C_jc(Count)
            end
end
```

1.4. Implementation

The above approach has been implemented in a computer program, EXPRESS-B, for logic and timing simulation of ECL bipolar circuits. The program accepts SPICE-like [6] input data in the form of transistor, resistor and capacitor interconnections. It then constructs a switch-graph model of the circuit. The program then performs circuit partitioning, identifies strongly-connected components using a depth-first search algorithm [15], and sets up an analysis sequencing procedure which implicitly exploits the latency properties of the subcircuits during the simulation. Symbolic logic expressions are then generated at the outputs of the subcircuits in terms of subcircuit inputs and initial conditions. Logic and timing simulation is then performed using the logic expressions and the delay model which consists of analytical expressions. The effect of parasitic capacitances, the loading capacitances and the effects of input slew rates are incorporated in the delay model.

Table I shows the CPU-time vesus the number of transistors. These circuit are based on an ECL implementation of a one

Circuit	#trans	EXPRESS-B Time(sec)	Spice Time (sec)
1-Bit Full adder	16	0.16	106.93
2-Bit Full adder	32	1.16	219.33
3-Bit Full adder	48	1.56	356.25
4-Bit Full adder	64	1.78	472.42
3 stage Ring Oscillator	16	0.42	60.17
5 stage Ring Oscillator	26	0.62	133.45
4-bit Shift Register	32	0.84	92.07

TABLE I

bit full adder shown in Figure 1.7. This circuit was cascaded to produce full-adders from 1 to 4 bits. This table shows that the time taken by EXPRESS-B is linear with the circuit size, and is more than 200 times faster than SPICE. This speed improvement will increase as the size of the circuit grows. The resulting waveform resulting from EXPRESS-B and from SPICE are shown in Figure 1.8 for the one bit adder. The dotted line is the SPICE waveform and the solid line is the EXPRESS-B waveform. Note the propagation of a glitch from node z to the output. If a glitch does not cross a threshold (or the midpoint crossing), it will not be propagated. Note that the glitch shown in Figure 1.8 is due to improper input waveforms. These logic glitches are detected and their effect propagated to the rest of the circuit. On the other hand, if the glitch is an overshoot or undershoot caused by feedback capacitor (such as Bootstrap), then these glitches are not detected. The full-adder example illustrates the correctness of the timing simulation without feedback. To illustrate the performance and accuracy in circuits containing feedback, a five stage ring oscillator circuit is simulated and the resulting waveform is shown in Figure 1.9.

To compare the accuracy of the result with SPICE , τ_d is computed by both EXPRESS-B and SPICE. The result was compared for many circuits. It was found that τ_d is in general within 15% when compared with that of SPICE.

The above circuits were run on a SUN 3/75 running Berkeley 4.2 UNIX; the program is written in the C programming language.

1.5. References

[1] M. I. Elmasry, *Digital Bipolar Integrated Circuits*. New York: John Wiley and Sons, 1983.

[2] S. Muroga, *VLSI System Design*. New York: John Wiley and Sons, 1982.

[3] P. Kozak, A. K. Bose , and A. Gupta, "Design aids for simulation of bipolar gate arrays," in *Proceedings of the 20th ACM Design Automation Workshop*, Miami Beach, FL, pp. 286-292, June 1983.

[4] B. Yuan, "A circuit compiler for CML stack synthesis," *VLSI Design*, pp. 68-76., January 1985 .

[5] The Second Topical Reasearch Conference; organized by the Semiconductor Research Corporation (SRC), *University of Florida at Gainsville*, May 28-29, 1987.

[6] L. W. Nagel , "SPICE2: a computer program to simulate semiconductor circuits," in *ERL Memo ERL-M520*,, University of California, Berkeley, May 1975.

[7] R. E. Bryant, "An algorithm for MOS logic simulation," *LAMBDA (now VLSI) Magazine*, vol. 1, pp. 46-53, 1980.

[8] M. R. Lightner and G. D. Hachtel, "Implication algorithms for MOS switch-level function macromodeling, implication and testing," *Proceedings of the 19th ACM Design Automation Workshop*, pp. 691-698, June 1982.

[9] I. N. Hajj and D. G. Saab, "Symbolic logic simulation of MOS circuits," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Newport Beach, CA, pp. 246-249, May 1983.

[10] J. P. Hayes, "A unified switching theory with application to VLSI design," *Proceeding IEEE*, vol. 70, pp. 1140-1151, October 1982.

[11] R. E. Bryant, "A switch-Level model and simulator for MOS digital systems," *IEEE Transactions on Computers*, vol. C-33, pp. 160-177, 1984.

[12] V. B. Rao, T. N. Trick, and I. N. Hajj, "A table-driven delay operator approach to timing simulation of MOS VLSI circuits," in *Proceedings of the IEEE International Conference on Computer Design*, New York, pp. 445-44, November 1983.

[13] I.N.Hajj and D.G.Saab, "Switch-level logic simulation of digital bipolar circuit," *IEEE, Transactions on Computer-Aided Design.*, vol. CAD-6 , pp. 251-258., March 1987.

[14] D. A. Hodges and H. G. Jackson, , *Analysis and Design of Digital Integrated Circuits*.. New York: McGraw-Hill, 1983.

[15] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, pp. 146-160, 1972.