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 described. 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 form 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 [15]. 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/or 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 y, which indicates the relative conducivity when ON.

Figure 1.1: Exclusive or/or gate and corresponding switch-graph model

25th ACM/IEEE Design Automation Conference®

Paper 21.2
288
The load resistor $R_{CL}$ is represented by an edge with label 1 and strength $y_1$, where $y_1 < y_2$.

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 the 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 $[l_3]$). In Figure 1.1, for example,

$$\begin{align*} 
O_1 &= A B \lor \overline{A} \overline{B} \\
O_2 &= \overline{A} B \lor A \overline{B} \\
Q &= \overline{O}_2 \\
P &= \overline{O}_1 
\end{align*}$$

### 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. We use the symbolic expressions to check the circuit limitation and critical performance.

**Definition:** The delay $\tau_u$ (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 $\tau_u$, detailed SPICE simulations were performed on the basic ECL inverter circuit shown in Figure 1.3. The results were examined and $\tau_u$ was found to depend strongly on parameters given in Table 3.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_{ff}$</td>
<td>Forward delay time</td>
</tr>
<tr>
<td>$R_b$</td>
<td>Transistor base resistance</td>
</tr>
<tr>
<td>$R_p$</td>
<td>Pullup resistance</td>
</tr>
<tr>
<td>$C_{BC}$</td>
<td>Base-collector depletion capacitance</td>
</tr>
<tr>
<td>$C_{CE}$</td>
<td>Collector to base capacitance</td>
</tr>
<tr>
<td>$C_{PQ}$</td>
<td>Capacitance of pull-up transistor current source</td>
</tr>
<tr>
<td>$T_{delay}$</td>
<td>Input slew rate</td>
</tr>
<tr>
<td>$C_{EF}$</td>
<td>Depletion capacitance of the emitter follower</td>
</tr>
<tr>
<td>$R_f$</td>
<td>Pulldown resistance of the emitter follower</td>
</tr>
<tr>
<td>$C_{load}$</td>
<td>Loading capacitance</td>
</tr>
</tbody>
</table>

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 necessarily be completed before another begins. Thirdly, there are many delay-sensitive parameters. Finally, associated with each input switching, there are two complementary output switching 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, $\tau_u$ by,

$$\tau_u \approx \tau_{in} + \tau_{slew} + T_{ext}$$

where $\tau_{in}$ is denoted as the intrinsic delay, $\tau_{slew}$ 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 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 $\tau_{in}$ is the time for the mechanism of extracting (injecting) half of the total stored excess minority charges, $Q_T$, 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 $\tau_{in}$.

To simplify the derivation, $\tau_{in}$ 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 steps

Consider the circuit model of the basic inverter shown in Figure 1.4. The current through a capacitive load is given by $i(t) = \frac{dq}{dt}$. If $i(t)$ is approximated by a constant $i_{slew}$, then

$$\Delta T = \int i_{slew} \, dt$$

across the capacitor by an amount of $\Delta q$, which results in corresponding change of capacitor voltage, $\Delta V$. From the definition of $\tau_{in}$, and because $\Delta T$ does not include the effect of $C_{root}$, then $\tau_{in}(ideal) = \frac{1}{2} \tau_{slew}$, where $\Delta q = \frac{C_{root}}{2}$. Referring to Figure 1.4, we see that $C_{root}$, then $\tau_{in}(ideal) = \frac{1}{2} \tau_{slew}$, where $\Delta q = \frac{C_{root}}{2}$. Referring to

---

**Paper 21.2**

289
Figure 1.4, for large $\beta$, $I_{\text{sat}} = I_{\text{FE}} + I_{\text{Qf}}$ and $I_{\text{Qf}} + V_{T} = 0$. If these conditions are satisfied, $V_{T} = -1.75\phi$, where $\phi = 0.8V$. and $\phi_{\text{source}} = \phi_{\text{ideal}} = \frac{1}{4} \phi$. Substituting $\phi_{\text{source}}$ into $\phi_{\text{source}}$ expression and replacing $I_{\text{source}}$ by $\phi_{\text{source}}$ yields:

$$\tau_{\text{fall(ideal)}} = \frac{r_{b}}{\phi_{\text{source}}}$$

$\tau_{\text{fall(ideal)}}$ expression and replacing $I_{\text{source}}$ by $\phi_{\text{source}}$ yields:

$$\tau_{\text{fall(ideal)}} = 2 \tau_{\text{rise}} \frac{r_{b}}{\phi_{\text{source}}}$$

For ideal intrinsic delay, $\tau_{\text{rise(ideal)}}$ is equal to $\tau_{\text{fall(ideal)}}$ where the subscript, rise and fall stand for rising and falling output, respectively.

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

$$\tau_{\text{fall}} = \tau_{\text{fall(ideal)}} + \tau_{\text{root}}$$

where $\tau_{\text{root}}$ is the extra time it takes to discharge $C_{\text{root}}$. The discharging current of $C_{\text{root}}$ is the sum of $I_{\text{f}}$ and $I_{\text{source}}$. Thus

$$\tau_{\text{root}} = \frac{\phi}{r_{b} + \frac{1}{2} \phi}$$

Substituting appropriate values into (2)

$$\tau_{\text{fall}} = 2 \tau_{\text{rise}} \frac{r_{b}}{\phi_{\text{source}}} + \frac{3 C_{\text{root}} r_{b} R_{E}}{4 r_{b} + 2 R_{E}}$$

To derive $\tau_{\text{rise}}$, note that as $C_{\text{root}}$ is being discharged for a swing of $\frac{3}{4} \phi$, $Q_{1}$ is discharged by an amount of $\tau_{\text{fall}} I_{\text{f}}$. After $C_{\text{root}}$ has completed its discharge, $Q_{1}$ will then be discharged by $\frac{1}{2} \phi$, which

$$\tau_{\text{rise}} = \tau_{\text{fall}} + \frac{\phi}{4 r_{b}}$$

when $\tau_{\text{fall}}$ and $I_{\text{f}}$ are replaced by their respective value yields:

$$\tau_{\text{rise}} = 2 \tau_{\text{rise}} \frac{r_{b}}{\phi_{\text{source}}} + \frac{3 C_{\text{root}} r_{b} R_{E}}{4 r_{b} + 2 R_{E}}$$

1.3.2.2 Intrinsic delay for rising input step

Considering Figure 1.3 and an input with a rising step from $-\phi$ up to $0$, a similar analysis to that of the falling steps results in exactly the same $\tau_{\text{fall(ideal)}}$. However, $C_{\text{root}}$ now has negligible effect on $\tau_{\text{in}}$, since $C_{\text{source}}$ needs to be charged for a swing of only $\frac{1}{4} \phi$ (as opposed to $\phi$ from the falling input step).

Hence,

$$\tau_{\text{rise}} \approx \tau_{\text{fall}} \approx \tau_{\text{fall(ideal)}}$$

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

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

$$\tau_{\text{sl}} = 0.15 T_{\text{sl}}$$

1.3.4 Extrinsic delay for single-input single-level circuit

Definitions: The extrinsic delay, $T_{\text{ex}}$, 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 $C_{c}$ is a simple sum of all the parasitic capacitances at the base of the emitter follower (node 1 in Figure 1.5) and is given by

$$C_{c} = C_{\text{source}} + 2 C_{e} + C_{\text{load}}$$

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, $I_{\text{ex}}$, flows in to charge $Q_{e}$ of the emitter follower. The resultant excess charge, $Q_{\text{excess}}$, is reflected as excess emitter current used to charge up the load and $Q_{\text{excess}} = \frac{1}{2} I_{\text{ex}} T_{\text{rise}}$, where $I_{\text{ex}}$ is the average excess base current for the period of interest, $T_{\text{rise}}$. From basic charge-control equations of a bipolar transistor [14], we can then write $I_{\text{ex}} = \frac{1}{2} I_{\text{ex}} T_{\text{rise}}$ and $I_{\text{ex}} = \frac{1}{2} I_{\text{ex}} + 1$. Note the following, as the output rises, the current through the loading resistor, $R_{E}$, also increases. However, to simplify the analysis, this increase is assumed to be negligible thus $I_{\text{ex}} \approx I_{\text{ex}} T_{\text{rise}}$, where $I_{\text{ex}}$ stands for the total voltage change at the output. Equating the two expression for $I_{\text{ex}}$, results in a quadratic equation

$$I_{\text{ex}} = \frac{1}{2} I_{\text{ex}} + 1$$

Emitter follower charge-up. $I_{\text{ex}} = I_{\text{ex}} - I_{\text{ex}}$ and $I_{\text{ex}} = \frac{1}{2} I_{\text{ex}} T_{\text{rise}}$. Note that $I_{\text{ex}}$ stands for the voltage change at the base of the emitter follower (node 1 in Figure 1.5). Substituting in the last quadratic equation

$$I_{\text{ex}} = \frac{1}{2} I_{\text{ex}} T_{\text{rise}} - (C_{e} \psi_{0} - I_{\text{ex}}) T_{\text{rise}} - (C_{e} \psi_{0} + C_{\text{load}} \psi_{0}) = 0$$

From the definition of $T_{\text{ex}}$, $\psi_{0}$ is equal to $\frac{1}{2} \phi$ and $\psi_{0}$ is set to $\frac{1}{2} \phi$ to
hand, $i_{\text{Peak}}$ can be computed as an average value $i_{\text{Peak}} = \frac{2}{\phi + \frac{R_c}{3R_c}}$ where $i_{\text{Peak}}$ and $i_{\text{Mid}}$ are the boundary current values during the period of interest. $T_{\text{Time}}$. Therefore, $i_{\text{Peak}} = \frac{2}{\phi}$. Substituting this expression for $i_{\text{Peak}}$ and the values of $\Psi_1$ and $\Psi_2$ into the previous quadratic equation, we can solve the extrinsic delay.

$$T_{\text{Time}} = 0.455R_cC_t - \tau_f$$

+ $\sqrt{0.207(R_cC_t)^2 + 0.911R_cC_t + 1.457\tau_fR_cC_{\text{Load}} + \tau_f^2}$

(7)

The effects of the loading resistance are incorporated by replacing $i_1$ by $i_{\text{Load}} = \frac{1}{4}R_c$ where the second term is an approximation for the average increase of current through $R_c$. A more accurate expression for $T_{\text{Time}}$ is then derived as

$$T_{\text{Time}} = 0.455R_cC_t - \tau_f + f_1$$

+ $\sqrt{0.207(R_cC_t)^2 + 0.911R_cC_t + 1.457\tau_fR_cC_{\text{Load}} + \tau_f^2 + f_2}$

$$f_1 = \frac{0.368R_c}{R_f}$$

(8)

$$f_2 = \frac{\tau_fR_c}{R_f} \left(0.333C_tR_c - 0.727\tau_f + \frac{0.1327R_c}{R_f}\right).$$

1.3.4.2. Emitter follower discharge

Consider the equivalent circuit in Fig. 1.5 with $I_{\text{Sink}}$ 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 $C_t \ll C_{\text{Load}}$: $Q_3$ is quickly discharged by $I_{\text{Sink}}$. The emitter follower is quickly turned off. If $T_{\text{Load}}$ is the time constant at the output, then $T_{\text{Total}} \approx T_{\text{Load}}$. Second case is when $C_t \gg C_{\text{Load}}$: the emitter follower stays on throughout the transition. If $C_t \gg C_{\text{Load}}$, the time constant at the base, then $T_{\text{Total}} - T_{\text{Rf}}$. Finally when $C_t = C_{\text{Load}}$: $T_{\text{Total}}$ depends strongly on both time constants, $T_{\text{Rf}}$ and $T_{\text{Load}}$.

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

$$T_{\text{Total}} = \min(T_{\text{Rf}}, T_{\text{Load}})$$

where $T_{\text{Rf}} = 0.693R_cC_t$ is the discharging time constant seen at the base of the emitter follower, and $T_{\text{Load}} = R_cC_{\text{Load}}\ln(\frac{V_{EE} - \phi}{V_{EE} - 1.5\phi}) = 0.1R_cC_{\text{Load}}$ is the discharging time constant seen at the output of the ECL gate with $V_{EE}$ equals $-5V$, and $\phi = \frac{V_{EE}}{2}$. When $C_t \ll C_{\text{Load}}$: equation (9) yields

$$T_{\text{Total}} = T_{\text{Load}}.$$ When $C_t \gg C_{\text{Load}}$: $T_{\text{Total}} = T_{\text{Rf}}$.

1.3.5. Extensions to Multi-Input/Level ECL Structures

Figure 1.6 shows a logic realization of $(A \lor B \land C) = (D \lor E)$ 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.

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 $\tau_{\text{transit}}$. 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 $\tau_{\text{transit}}$. 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 $t_i$ is in level $i$ if the proper mid voltage swing at the base of $t_i$ is equal to $-(1 - (i - 1))\phi$ (level 1 is denoted by top level).

From the above definition and the valve analogy $\tau_{\text{transit}}(i)$ and $\tau_{\text{Transit}}(i)$ are given by $\tau_{\text{transit}}(i) = \tau_{\text{Transit}}(i)$ and $\tau_{\text{Transit}}(i) = \gamma\tau_f$, where $\tau_{\text{Transit}}(i)$ is the intrinsic delay associated with the $i^{th}$ level, and where $\gamma$ 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 $\tau_{\text{Transit}}$ and can be found from $\tau_{\text{transit}}(i)$ and $\tau_{\text{Transit}}(i)$. If the switching input is at level $i$, then $\tau_{\text{Transit}} = \tau_{\text{transit}}(i) + (i - 1)\tau_{\text{Transit}}$ and is equal to

$$\tau_{\text{Transit}} = \tau_{\text{transit}}(i + 1) + \gamma(1 - 1)\tau_f.$$ (10)

For a stacked circuit, since the extrinsic delay, $\tau_{\text{Transit}}$, is associated only with the top level, the output threshold crossing time is given by $t_2 = t_1 + \tau_{\text{Transit}}$, and the transition at the output starts at $t_1 = \tau_{\text{Transit}}(i)$, where $t_1$, $t_2$, and $t_3$ 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 $t_3(i)$ denote the output threshold crossing time for an intrinsic circuit due to an input change at level $i$. From (10) we define

$$t_3(i) = \tau_{\text{Transit}}(i + 1) + \gamma(1 - 1)\tau_f.$$ (11)

For the stacked circuit is mainly limited by the slowest $t_3(k)$ at the $k^{th}$ level. Hence,

$$t_2 = T_{\text{Transit}} + \max(t_3(k)).$$ (12)

$$t_1 = \max(t_3(i)).$$ (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 $t_3(i)$: Therefore,

$$t_2 = T_{\text{Transit}} + \min(t_3(i)).$$ (14)
and the output will begin to switch as soon as any input begins to switch at

\[ t'_i = \min(t_{th}(i)) \]  

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 kth 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 \( t_{ovhp}(i) \), \( i = 1, 2, ..., L - 1 \), as the overlap time between each individual valve along the branch to valve(k). L is the total number of levels. Note that \( t_{ovhp}(i) \) ranges from 0 to \( t_{in}(k) \). To account for this reduction of delay, we rewrite (40) as:

\[ t_2 = T_{ext} + \min(t_{th}(i)) - \sum_{i=1}^{L-1} t_{ovhp}(i) \]  

Note that (16) is valid only when \( t_{ovhp}(i) \) is decreasing in value for each increment of i. This can be done by simply sorting \( t_{ovhp}(i) \). 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 multiple of \( t_{th} \) (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 \( t_{th}(n) \) denotes the output threshold crossing time for an intrinsic circuit due to an input change at input \( n \). And

\[ t_2 = t_{th}(n) + t_{in}(n). \]  

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 value for each of the \( t_{th}(n) \) 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 \( t'_i \) is given by:

\[ t'_i = T_{ext} + \min(t_{th}(n)) - \sum_{n=1}^{N-1} t_{in}(n + 1) \]  

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

\[ t'_i = \min(t_{th}(n)) \]  

1.3.6.3. Multiple inputs from high to low

Consider the case when multiple inputs change simultaneously from HIGH to LOW. \( t_{in} \) for the multi-input circuit is mainly limited by the slowest intrinsic midpoint crossing time. Therefore,

\[ t'_i = T_{ext} + \max(t_{th}(n)) \]  

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

\[ t'_i = \max(t_{th}(n)) \]  

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

\[ C_{load} = C_{wire} + F_1 \cdot \sum C_{fanout} + \sum C_{loadage}, \]  

where \( C_{wire} \) is the wiring capacitance, \( C_{fanout} \) and \( C_{loadage} \) are the capacitance of the fanout differential transistor. \( F_1 \) is an empirical factor and set to 0.5 from SPICE. \( C_{loadage} \) is the dynamic capacitance for the base leakage current of each fanout. For each fanout whose \( I_{th} \) is switched from one transistor to the other due to the driving signal, we can express \( C_{loadage} \) simply by

\[ C_{loadage} = F_2 \cdot \frac{T_{ext} \cdot T_{in} \cdot C_{loadage}}{\phi/2} \]  

In the above equation, \( I_{loadage} \) is the average base leakage current which starts to flow when the output is approaching its midpoint threshold. Therefore, throughout the entire intrinsic delay of the driver gate, we can assume \( I_{loadage} \) 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 \( I_{loadage} = I_{th} \cdot \frac{T_{in}}{0.15 \cdot T_{load}} \), where \( I_{th} \) is the average current.

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

```
procedure C_load (driver parameter, fanout 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 t_in, I_avg, fanout (Count)
      Evaluate C_loadage (Count) (eq. 22)
      C_load = C_load + C_loadage(Count)
    end if
    C_load = C_load + F_1 \cdot C_p(Count)
  end 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 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 versus the number of transistors. These circuit are based on an ECL implementation of a one
A 5-stage ring oscillator circuit is simulated and the resulting performance and accuracy in circuits containing feedback, a five by feedback capacitor (such as the waveform is shown in Figure 1.9. 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 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, \( \tau_d \) is computed by both EXPRESS-B and SPICE. The result was compared for many circuits. It was found that \( \tau_d \) 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.

### Table 1

<table>
<thead>
<tr>
<th>Circuit</th>
<th>#trans</th>
<th>EXPRESS-B Time (sec)</th>
<th>SPICE Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Bit Full adder</td>
<td>16</td>
<td>0.16</td>
<td>108.93</td>
</tr>
<tr>
<td>2-Bit Full adder</td>
<td>32</td>
<td>1.16</td>
<td>219.33</td>
</tr>
<tr>
<td>3-Bit Full adder</td>
<td>48</td>
<td>1.56</td>
<td>356.23</td>
</tr>
<tr>
<td>4-Bit Full adder</td>
<td>64</td>
<td>1.78</td>
<td>472.42</td>
</tr>
<tr>
<td>3 stage Ring Oscillator</td>
<td>16</td>
<td>0.42</td>
<td>60.17</td>
</tr>
<tr>
<td>5 stage Ring Oscillator</td>
<td>26</td>
<td>0.62</td>
<td>133.45</td>
</tr>
<tr>
<td>4-bit Shift Register</td>
<td>32</td>
<td>0.84</td>
<td>92.07</td>
</tr>
</tbody>
</table>

### References