Maze Routing with Buffer Insertion Under Transition Time Constraints

Li-Da Huang, Minghorng Lai, D. F. Wong

Department of Computer Sciences
University of Texas at Austin
Austin, TX 78712 U.S.A.
{lida, mlai, wong@cs.utexas.edu}

Youxin Gao

Avant! Corporation
46871 Bayside Parkway
Fremont, CA 94538 U.S.A.
gao@avanticorp.com

Abstract

In this paper, we address the problem of simultaneous routing and buffer insertion. Recently in [12, 22], the authors considered simultaneous maze routing and buffer insertion under the Elmore delay model. Their algorithms can take into account both routing obstacles and restrictions on buffer locations. It is well known that Elmore delay is only a first-order approximation of signal delay and hence could be very inaccurate. Moreover, we cannot impose constraints on the transition times of the output signal waveform at the sink or at the buffers on the route. In this paper we extend the algorithm in [12] so that accurate delay models (e.g., transmission line model, delay look-up table from SPICE, etc.) can be used. We show that the problem of finding a minimum-delay buffered routing path can be formulated as a shortest path problem in a specially constructed weighted graph. By including only the vertices with qualifying transition times in the graph, we guarantee that all transition time constraints are satisfied. Our algorithm can be easily extended to handle buffer sizing and wire sizing. It can be applied iteratively to improve any given routing tree solution. Experimental results show that our algorithm performs well.

1 Introduction

As the VLSI technology shrinks down to nanometer range, interconnect delay becomes the bottleneck in achieving high performance circuits. Techniques aiming at reducing interconnect delay are highly desirable. Buffer insertion is an efficient technique in interconnect optimization. It has been an active research area in the past few years, resulting in a large body of literature on the subject [1, 2, 4, 6, 7, 14, 17, 19, 21]. Traditionally, buffer insertion is a post-layout optimization technique, meaning that it is applied to improve the layout after the routing stage. Recently, it was demonstrated in [22] that simultaneous routing and buffer insertion can produce significantly better results, and an algorithm based on dynamic programming was presented. A simpler and faster graph-based algorithm was presented in [12]. Both algorithms in [12, 22] use Elmore delay model and can take into account both routing obstacles and restrictions on buffer locations. Unfortunately, it is well known that Elmore delay is only a first-order approximation of signal delay and hence could be very inaccurate. Moreover, we cannot impose constraints on the transition times of the output waveform at the sink or at the buffers on the route.

Fast transition time is crucial in high speed circuit designs for maintaining signal integrity. A slow transition on a net is more susceptible to a coupling noise caused by a fast-transition signal on nearby parallel nets without shielding. Therefore, by controlling the transition time, we can reduce coupling noise and maintain signal integrity. As a rule of thumb, the transition time should be kept within 10% - 15% of the cycle time in the design. However, the Elmore model based buffer insertion methods cannot capture the transition time violation but only optimize some threshold delay (such as 50% delay). In practice, most often the designs meet the maximum delay criterion but violate the slew rate (which is the inverse of transition time) constraint.

In this paper we extend [12] to obtain a graph-based algorithm for simultaneous maze routing and buffer insertion. We use accurate delay models and consider transition time constraints. Routing obstacles and restrictions on buffer locations are taken into account. In our algorithm, we accurately represent a signal waveform by a finite ramp, which is characterized by two parameters: shift time and transition time. By using this characterization of the waveform, we show that given an input transition time, the output shift time and transition time for any buffer-to-buffer segment of a buffered routing path can be uniquely determined (e.g., by using transmission line model, delay look-up table from SPICE, etc.) As a result, the delay along any buffered routing path in the routing area is the sum of individual shift times of all the buffer-to-buffer segments along the path and 50 % of the transition time at the last buffer. Based on this observation, we construct a weighted graph called Buffer Planning Graph (BPG) where each vertex corresponds to a transition time value associated with a certain possible buffer location. That is, a vertex in the BPG represents not only the buffer location but also the transition time. Each edge in BPG represents a buffer-to-buffer segment in a buffered routing path, with the assigned weight being the shift time for the segment. We show that the problem of finding a minimum-delay buffered routing path in the routing area can be formulated as a shortest path problem in the BPG. By appropriately eliminating some of the vertices in the BPG, we guarantee all transition time constraints are satisfied. Our algorithm can be easily extended to handle buffer sizing and wire sizing. Experimental results show that our algorithm performs well.

The remainder of the paper is organized as follows: In Section 2, we describe the accurate delay model in detail. In Section 3, we review previous methods for finding a minimum-delay buffered routing path and propose a new algorithm that is able to consider transition time constraints. In Section 4, we show our experimental results and provide some concluding remarks.

2 Accurate Delay Model

We now describe an accurate delay model that we use to quickly build up a delay look-up table. Our algorithm can use any delay look-up table constructed by using other delay models (e.g., SPICE simulation). Since a single delay value (e.g., 50% delay) in a general delay model cannot fully characterize the waveform of a signal, we approximate each signal as a finite ramp which is characterized by two parameters: shift time S and transition time T (see Fig. 1). Any arbitrary signal is represented by a finite ramp by connecting two points at 10% and 90% threshold voltages, respectively. The usual 50% delay is thus $S + \frac{1}{2}T$. To simplify the notation, we use a pair (S, T) to represent a finite ramp. In the rest of the section, we show how to use an analytical delay model in which wire delay is based on the transmission line model and buffer delay is characterized by k-factor equations.

In buffer insertion, a typical wire-buffer system is shown in

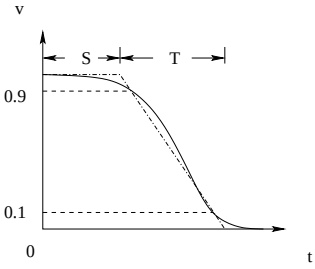

Figure 1: An arbitrary signal is represented by a finite ramp with shift time S and transition time T.

Fig. 2(a), where a wire is connecting two buffers. In Fig. 2(b), the buffer is represented by a circuit which contains an input capacitance C_B and a voltage source V_B. The waveform calculation is cascaded in terms of a pair (S, T) through each wire-buffer system.

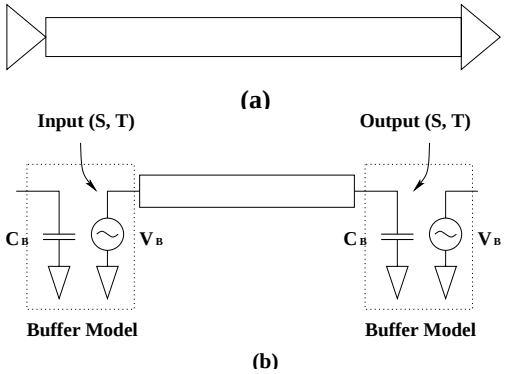

Figure 2: (a) A buffer-to-buffer segment showing a wire connecting two buffers. (b) Signal waveform calculation is cascaded in terms of a pair (S, T).

We use the transmission line model to model the interconnect wire, which is found to be more accurate than methods based on lumped circuit approximation [10, 21]. Under the transmission line model, voltage and current at any position are described by the telegraph equations. In modeling the interconnect wire, both fringing capacitance and inductance, considered important in today's design [3, 9], are taken into consideration. The waveform calculation for a wire consists of the following steps (details can be found in [9, 10]):

1. Use transmission line model to derive $\mathcal{ABCD}$ parameters for an interconnect wire by solving telegraph equations.
2. Use a delay model based on three pole approximation. Obtain analytical forms for calculating the first three terms b_1, b_2 and b_3 in the transfer function $H(s) = \frac{1}{1+b_1 s+b_2 s^2+b_3 s^3+\ldots}$, and derive the time domain response by assuming a finite ramp input.
3. Use analytical delay expressions to calculate delay at any threshold voltage [10, 11], as well as the output waveform (S_{out}, T_{out}).

To calculate the output waveform (S_{out}, T_{out}) for a buffer, we make use of k-factor equations. Delays at threshold voltages 10% and 90% are expressed in terms of the following empirical equations [20]:

$$T_{10} = (k_1 + k_2 C_w) T_{in} + k_3 C_w^2 + k_4 C_w + k_5 \quad (1)$$

$$T_{90} = (k'_1 + k'_2 C_w) T_{in} + k'_3 C_w^2 + k'_4 C_w + k'_5 \quad (2)$$

In the equations, T_{in} is the transition time of an input slope, $C_w = C_L/w$, C_L is the buffer's load capacitance, and w is the buffer's channel width. Therefore, the waveform of the voltage source V_B can be calculated in terms of a pair (S, T) as: $S = (9T_{10} - T_{90})/8$, $T = (T_{90} - T_{10})/0.8$ for a rising ramp; and $S = (9T_{90} - T_{10})/8$, $T = (T_{10} - T_{90})/0.8$ for a falling ramp. Without loss of generality, throughout the rest of the paper we assume that PMOS and NMOS have the same driving capability. Therefore it is not necessary to distinguish between rising transition and falling transition. Because of the resistance shielding, the load capacitance in the k-factor equations is the effective capacitance for the wire [18]. To compute the effective capacitance, we use analytical expressions to calculate the input admittance $Y(s)$. We expand $Y(s)$ into Taylor series and keep first three terms, i.e., $Y(s) = y_1 s + y_2 s^2 + y_3 s^3$. The technique in [16] is applied to determine an equivalent CRC Π-model which matches $Y(s)$. Then, we compute the effective capacitance C_{eff} for this CRC Π-model using the technique in [18].

For our delay model, we note that shift time S is additive. That is, if an input waveform $(0, T_{in})$ causes an output waveform (S_{out}, T_{out}), then another input waveform (S_{in}, T_{in}) will cause an output waveform $(S_{in} + S_{out}, T_{out})$. For the buffer macro-model in equations (1-2), the additivity of shift time is obvious, since delays are not depending on input shift time. In the delay calculation for a wire, suppose the final voltage response is $V_{out}(s) = V_{in}(s)H(s)$, where $V_{in}(s)$ is the input voltage response, and $H(s)$ is the transfer function. Assume that the corresponding voltage response in the time domain is $v_{out}(t)$. If the input signal is shifted by $\mathcal{S}$, the new input signal will be $V'_{in}(s) = e^{-s\mathcal{S}} V_{in}(s)$. The corresponding output voltage is then $V'_{out}(s) = e^{-s\mathcal{S}} V_{in}(s)H(s) = e^{-s\mathcal{S}} V_{out}(s)$. The new voltage response in the time domain is thus the original voltage response $v_{out}(t)$ shifted by $\mathcal{S}$.

A direct result of the additivity of shift time is that, when computing output shift time and transition time for an output waveform, we can specify the input waveform as $(0, T_{in})$ instead of (S_{in}, T_{in}) for either a buffer or a wire, or a system contains both. For the wire-buffer system shown in Fig. 2(a), the output waveform is uniquely determined by the wire length and the input waveform (thus T_{in} only). Specifically, given an input transition time, the output shift time and transition time for any buffer-to-buffer segment in a buffered path can be uniquely determined.

3 Maze Routing with Buffer Insertion Under Transition Time Constraints

In this section we define the simultaneous maze routing and buffer insertion problem and review previously proposed approaches. These methods use Elmore delay model and find a minimum-delay buffered path in the routing area. However, they were incapable of handling transition time constraints. We therefore utilize the accurate delay model in Section 2 and propose a new formulation that performs maze routing with buffer insertion under transition time constraints.

3.1 Problem Definition and Previously Proposed Methods

The goal of maze routing is to find a route between two terminals s and t in a routing area. Some wiring obstacles and restrictions on buffer locations may be present in the routing area. One important objective of maze routing is to minimize the interconnect delay of the routed path. The Maze Routing with Buffer Insertion problem can be formally described as follows:

Problem 1 (Maze Routing with Buffer Insertion) *Given a routing grid graph $G = (V, E)$, a buffer function p with $p(v) = 1$*

indicating buffer insertion allowed at v, two nodes $s, t \in V$, find a buffered path $P = (v_1, v_2, \ldots, v_n)$, with $v_1 = s$, $v_n = t$, $b(v_i) \in 0, 1$ where $b(v_i) = 0$ indicates that no buffer is inserted at v_i, such that the delay for path P is minimized.

We call any buffered path P from s to t with the minimum delay a minimum-delay buffered path. A routing grid graph and a minimum-delay buffered path is shown in Fig. 3. Note that in the graph no wires are allowed in the dark area as it represents a wiring obstacle. Wires are allowed to go through the buffer obstacle areas(gray areas); however, no buffers can be inserted in these areas.

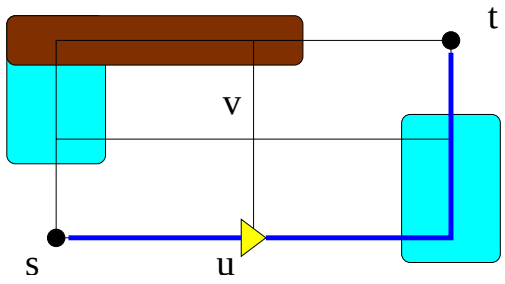

Figure 3: A routing grid graph. Both u and v are possible buffer locations. Wiring obstacle and buffer obstacles are indicated as dark area and gray areas, respectively. A buffered path from s to t is shown with buffer inserted at u.

Previously, in [22] Zhou et al proposed a dynamic programming based approach for finding minimum-delay buffered path in the grid graph with restrictions on buffer locations. The authors used the Elmore delay model. Sub-solutions of the form (capacitance,time) are propagated from t to s. The dynamic programming approach is general. However, it is also storage-intensive and inefficient when wire sizing is considered.

In [12], Lai and Wong formulated the problem as a shortest path problem and improved the performance even when wire sizing is taken into consideration. They made the observation that the delay of any individual buffer-to-buffer segment in a buffered path can be independently determined by the wire length of the segment and the buffers. Based on this observation, a buffer planning graph(BP-Graph) is constructed. Each vertex of the BP-Graph represents a possible buffer location in the grid graph G. An edge (u, v) in BP-Graph represents a buffer-to-buffer wire segment in a buffered path. The weight of edge (u, v) in BP-Graph is the minimum Elmore delay between node u and v in grid graph G. This minimum Elmore delay is a function of the shortest distance between u and v in G and the resistance and capacitance of buffers. Under this formulation, the minimum-delay buffered path in the routing graph G corresponds to a shortest path in the BP-Graph. However, their formulation is also based on the Elmore delay model, and cannot handle transition time constraints.

3.2 New Formulation Considering Transition Time Constraints

Signal integrity has become a major concern as technology scales down and operating frequency rises. As a result, signal transition time should also be considered in routing and buffer insertion. The Maze Routing with Buffer Insertion Under Transition Time Constraint problem can be formally described as follows:

Problem 2 (Maze Routing with Buffer Insertion Under Transition Time Constraint) *Given a routing grid graph $G = (V, E)$, a buffer function p with $p(v) = 1$ indicating buffer insertion allowed at node v, two nodes $s, t \in V$, upper bound on transition time T_U, find a buffered path $P = (v_1, v_2, \ldots, v_n)$, with $v_1 = s$, $v_n = t$, $b(v_i) \in 0, 1$ where $b(v_i) = 0$ indicates that no buffer is inserted at v_i, transition time $T_{tr} < T_U$, such that the delay for path P is minimized.*

Note that the BP-Graph in [12] does not contain any information about transition time; therefore, the proposed method cannot take transition time constraints into consideration.

3.2.1 Buffer Planning Graph (BPG)

We show how to construct a new weighted graph called Buffer Planning Graph (BPG) which contains transition time information. Note that the BP-Graph in [12] does not contain any information about transition time. A buffered path in the routing area can be decomposed into buffer-to-buffer segments joined together at inserted buffers. In the BPG, each vertex T_i^v corresponds to a possible transition time value t_i^v associated with the possible buffer location at node v. That is, vertex T_i^v in the BPG represents not only the buffer location v but also the transition time t_i^v. Given the transition time t_i^u at a vertex T_i^u, we compute the shift time s_j^v and transition time t_j^v. If the transition time satisfies the constraint, we introduce a new edge (T_i^u, T_j^v) with weight s_j^v. Each edge (T_i^u, T_j^v) represents a buffer-to-buffer segment between node u and v with distinct shift time s_j^v and transition time t_j^v uniquely determined by t_i^u. Therefore, for each buffered path in the routing area, there is a corresponding path in the BPG with vertices containing transition times at inserted buffer locations and edge weights being shift times for buffer-to-buffer segments. As a result, the delay along any buffered path in the routing area is the sum of individual shift times of all the edges along the corresponding path in the BPG and 50% of the transition time at the last buffer. We will see that the problem of finding a minimum-delay buffered routing path can be formulated as a shortest path problem in the BPG.

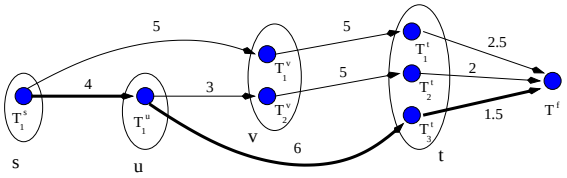

Figure 4: The BPG for finding a minimum-delay buffered path from s to t in the grid graph in Fig. 3. The vertices are signal transition times associated with a buffer location. For every buffered path in the routing area, there is a corresponding path in the BPG with vertices containing transition times at inserted buffer locations and edge weights being shift times for buffer-to-buffer segments. For example, for buffered path (s, v, t) in the routing grid, there is a corresponding path (T_1^s, T_1^v, T_1^t) in the BPG. With transition time information embedded at vertices in the BPG, the transition time constraints are easily enforced. The shortest path $(T_1^s, T_1^u, T_3^t, T^f)$ in the BPG corresponds to the minimum-delay buffered path (s, u, t) in the routing area(Fig. 3).

In Fig. 4, we show how to build the BPG for the routing grid in Fig. 3. Assume that we are trying to find a minimum-delay buffered path from s to t. Nodes u and v are two possible buffer insertion locations on the buffered path from s to t. We introduce a vertex T_1^s representing transition time t_1^s at node s. Then, we compute the shift time s_1^u(=4) and transition time t_1^u for buffer-to-buffer segment (s, u). Vertex T_1^u is added to represent transition

time t_1^u at node u. Edge (T_1^s, T_1^u) is created with weight $s_1^u = 4$. The same procedure is repeated to create other vertices and edges in the BPG. Note that there are two vertices T_1^v and T_2^v associated with the same node v. Vertex T_1^v represents the transition time of the signal arriving along path (s, v), while vertex T_2^v represents the transition time of the signal arriving along path (s, u, v). All possible buffered paths from s to v in Fig. 3 are shown in Fig. 5. Notice that each of these buffered paths from s to t has a corresponding path from T_1^s to $T_i^t, i = 1, 2, 3$, in the BPG in Fig. 4.

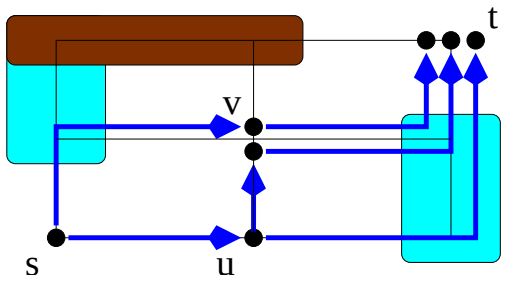

Figure 5: Buffered paths from s to t. Buffers are allowed to be inserted at u and/or v. Each of these buffered paths has a corresponding path from T_1^s to $T_i^t, i = 1, 2, 3$, in the BPG in Fig. 4.

We create an extra vertex T^f and add new edges connecting vertices T_i^t representing different transition times t_i^t at t to T^f. The weights of these edges are 50% of the respective transition times of the vertices at t. It is clear that the minimum-delay buffered path from s to t in the routing area corresponds to the shortest path from T_1^s to T^f in the BPG. From Fig. 4, the shortest path from T_1^s to T^f is $(T_1^s, T_1^u, T_3^t, T^f)$, and the minimum-delay buffered path from s to t has a buffer inserted at u (Fig. 3). The signal delay from s to t is computed as 11.5, with the 50% transition time at the load end being 1.5.

3.2.2 Computation in Shortest Path Formulation

We have shown how to construct the BPG in Fig. 4 for the simple grid graph G in Fig. 3. In simplified Fig. 4, connections between vertices representing transition time values at a pair of buffer locations are strictly from one buffer to the other. In reality, the connections need to be bi-directional because both segments $u \rightarrow v$ and $v \rightarrow u$ could be a component in the minimum-delay buffered path P from s to t.

Fig. 6 is an example showing connections between vertices representing transition time values at two buffer locations. Given transition time t_i^u at vertex T_i^u, the weight of edge (T_i^u, T_j^v) in the BPG is the shift time computed by the accurate delay model for the minimum-distance buffer-to-buffer segment between u and v in G.

For a large grid graph, the number of possible transition time values for a buffer location can be huge. Therefore, we divide the transition time values into k discrete time bins. Our experiments show that with 50 transition time bins the computed delays are highly accurate compared to SPICE results.

The BPG constructed for the grid graph in Fig. 7 is shown in Fig. 8. For each possible buffer location, we introduce three vertices in BPG representing three discrete transition time bins at the buffer location. The shift time for edge (T_i^u, T_j^v) is determined by the transition time t_i^u and the minimum distance $d(u, v)$ between u and v in the routing grid graph G. The 50% transition times are labeled as the weights for the edges connecting T_i^t to T^f. The weight of the shortest path from T_1^s to T^f in the BPG is the minimum delay of the buffered path from s to t in G. In

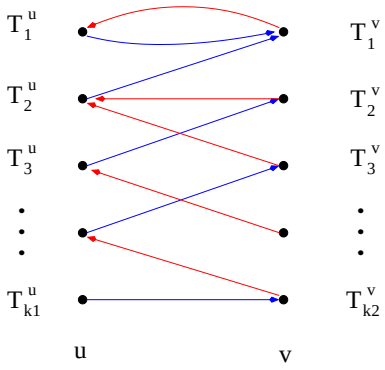

Figure 6: An example showing connections between vertices representing transition time values at a pair of buffer locations. The weight of edge (T_i^u, T_j^v) is the shift time of the output waveform determined by the accurate delay model with input transition time t_i^u and the shortest distance $d(u, v)$ between u and v in grid graph G.

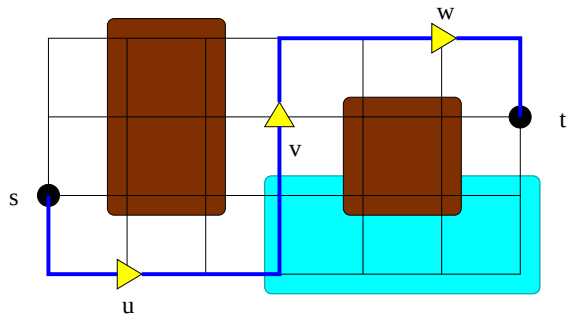

Figure 7: A routing grid graph and the minimum-delay buffered path corresponding to the shortest path in the BPG in Fig. 8.

the BPG(Fig. 8) the shortest path is $(T_1^s, T_2^u, T_1^v, T_3^w, T_1^t, T^f)$. It corresponding minimum-delay buffered path is (s, u, v, w, t) in Fig. 7.

We utilize a look-up table to speed up our algorithm. From Equation (1) and (2), output shift time and transition time of a buffer-to-buffer segment are uniquely determined by the input shift time and the length of the wire. Therefore, for a pair (input transition time, wire length), we can pre-compute the output shift time and transition time of a buffer-to-buffer segment and store the result in a look-up table. Then, the pair (transition time, wire length) is used as an index to check the look-up table.

If there are transition time constraints along the buffered routing path, we can delete the vertices representing transition times violating the constraints. Then, the shortest path in the BPG corresponds to the minimum-delay buffered path that meets the transition time constraints.

The details of the algorithm is in Fig. 9. Notice that not every edge and vertex is needed in finding the minimum delay buffered path under the transition time constraint. The vertices representing transition times larger than the constraint are removed without affecting the result. The transition time constraints can be specified at every possible buffer location.

This algorithm could be easily extended to consider multiple buffer types and simultaneous wire sizing under transition time constraints. For multiple buffer types, each vertex in the BP-Graph in [12] represents one buffer type and is expanded into k vertices in the BPG for k discrete transition time bins of that buffer type. As for wire sizing, the delay and transition time of

test circuit	grid size	Elmore Delay Model [12]			Accurate Delay Model			
		$T_{50\%}$(ns)	T_{tr}(ns)	$T_{tr} < 0.5$ns ?	$T_{50\%}$(ns)	T_{tr}(ns)	$T_{tr} < 0.5$ns ?	runtime (s)
P1	20 x 20	1.069	0.912	No	1.567	0.190	Yes	6.38
P2	20 x 20	1.047	0.976	No	1.395	0.272	Yes	6.65
P3	25 x 25	0.081	0.884	No	1.375	0.272	Yes	8.75
P4	25 x 25	1.037	0.864	No	1.270	0.246	Yes	8.10
P5	50 x 50	1.163	0.786	No	1.567	0.432	Yes	30.13
P6	50 x 50	0.755	1.794	No	1.035	0.434	Yes	30.99

Table 1: Finding a minimum-delay buffered path with transition time constraints $T_{tr} < 0.5$ns. Note that the paths routed by method in [12] do not satisfy the transition time constraints.

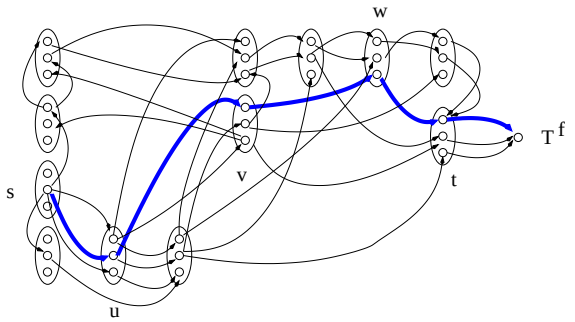

Figure 8: The BPG for the grid graph in Fig. 7. Note that only the grid nodes that are possible buffer locations introduce new vertices in the BPG. For clarity reason, not all edges are shown in this graph. The shortest path from s to t is also indicated in the graph. Its corresponding minimum-delay buffered path is shown in Fig. 7.

the wire library W are pre-calculated and stored in a look-up table and can be easily integrated into our algorithm.

For a grid graph $G = (V, E)$, with k transition time bins, at most $k|V|$ new vertices are created in the BPG. It takes $O(|V|^2 \log(|V|))$ to compute the shortest distances between all pairs of nodes in G. The runtime for computing the shortest path in the BPG is $O(k^2|V|^2)$. The space complexity of the Maze Routing with Buffer Insertion Under Transition Time Constraint Algorithm is $O(k^2|V|^2)$, since the BPG is a dense graph with $O(k|V|)$ vertices.

3.2.3 Fixing Transition Time Violation in Routing Trees

Our algorithm can be applied to improve any given routing tree. Most of the routing trees are constructed based on the Elmore delay model or Rubinstein model [23, 24]. Since both of the delay models are only a first-order approximation of the signal delay, they can not accurately capture the transition time of the signal. The resultant routing tree can contain some transition time violations. In Fig. 10, the buffered routing tree is constructed to minimize the maximum delay, but the input signals at b_3 and at the sink t_3 do not meet the transition time constraints. In this case, we can apply our algorithm on the path b_2 through t_1, and path b_6 through t_3 respectively. Both paths are rerouted and/or re-buffered to meet the transition time constraints with minimum delays as shown in Fig. 11. In general, we can iteratively apply our algorithm to any given routing tree solution for minimizing delay subject to transition time constraints.

Algorithm for Maze Routing with Buffer Insertion under Transition Time Constraint

```
G = (V,E) is the routing grid graph
for each pair of nodes u, v ∈ V do
    compute shortest distance d(u,v) in G
k = number of transition time bins
for each u ∈ V do
    for each 0 ≤ i < k do
        if buffer insertion is allowed at u
          create a new vertex T_i^u in BPG
for each pair of (T_i^u, T_j^v)
    if d(u,v) < ∞ then
        use (t_i^u, d(u,v)) as index to check
        look-up table for output waveform
        in terms of (s_j^v, t_j^v).
        if t_j^v meets transition time constraint
          create edge (T_i^u, T_j^v) with weight s_j^v
create a new vertex T^f
for each vertex T_i^t,
    if t_i^t meets transition time constraint
      create edge (T_i^t, T^f) with edge weight
      equal to 50% of transition time t_i^t
find the shortest path from T_1^s to T^f
in the BPG
find the minimum-delay buffered path
from s to t in G
```

Figure 9: Using shortest path formulation to solve mazing routing with buffer insertion under transition time constraints.

4 Experiment Results and Concluding Remarks

We tested our formulation and algorithm on a set of routing areas. Routing obstacles and macro blocks are randomly generated and placed in these routing areas. The wire parameters are chosen as follows: wire width $w = 1.035\mu m$, unit square resistance $r_0 = 0.092\Omega/\Box$, unit area capacitance $c_0 = 0.03205fF/\mu m$, driver resistance $R_D = 28.3\Omega$, load capacitance $C_L = 0.016pF$, grid unit length is $400\mu m$, unit length fringing capacitance $c_f = 0.0877fF/\mu m$, and unit length self inductance $l_0 = 0.73913pH/\mu m$. For buffers, we choose fixed channel width and length for the NMOS transistor as $5\mu m$ and $0.5\mu m$, respectively. We build our look-up table based on a maximum wire length of 100 grid units. Therefore, it has 100 locations where we can insert buffers on the wire. The table can be expanded to cover more routing area. In order to decide the number of the transition time bins, we use a one-dimensional grid (i.e., a

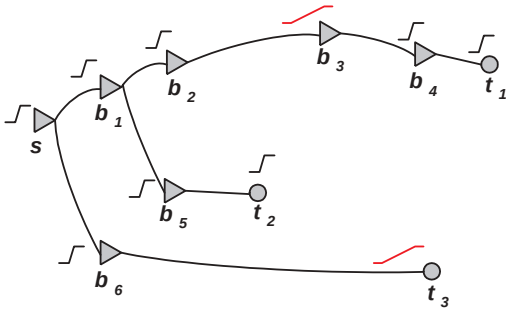

Figure 10: A routing tree based on the Elmore delay mode. The input signals at the buffer b_3 and the sink t_3 do not meet the transition time constraints.

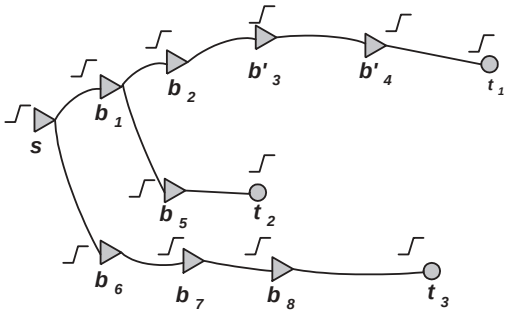

Figure 11: After fixing the transition time violations, the buffer b_3' and b_4' are the new locations of the b_3 and b_4 in Fig. 10. The buffer b_7 and b_8 are introduced to fix the transition time violation at sink t_3.

line) to find the minimum 50% threshold delay. We also increase the number of discrete transition time bins to increase the accuracy of computed delays as compared to SPICE. Our experiments show that when using 50 transition time bins the computed delays are highly accurate. For the input signal, we specify a ramp by choosing $S = 0$, $T_{in} = 1ps$.

In the experiments, we use our algorithm to find minimum-delay buffered paths in the routing area with constraints on the transition time at the load end. Note that the method in [12] cannot take transition time into consideration. It is clear from Table 1 that the transition time constraints are not automatically satisfied by the buffered paths routed by method in [12]. That is, although the method in [12] is capable of minimizing the delay, the routed paths could violate transition time constraints. Transition time constraint violations could cause circuit performance degradation at high frequency and should be avoided. Our method, on the other hand, is capable of finding a minimum-delay buffered path while taking the transition time constraints into consideration.

In Fig. 12 we show the routing area and the minimum-delay buffered path for one test circuit. In order to have a fast transition waveform, the buffers are inserted as close to the target node as possible. The minimum delay of the buffered path increases when transition time constraints are imposed. However, by using our formulation, we control the transition timing, which, in addition to delay, is also an important concern in high speed circuit design.

We have shown in this paper a new formulation for the simultaneous maze routing and buffer insertion problem. By using an accurate delay model and a buffer planning graph (BPG), the transition time constraints are taken into consideration. Our experiment results show that our algorithm performs well.

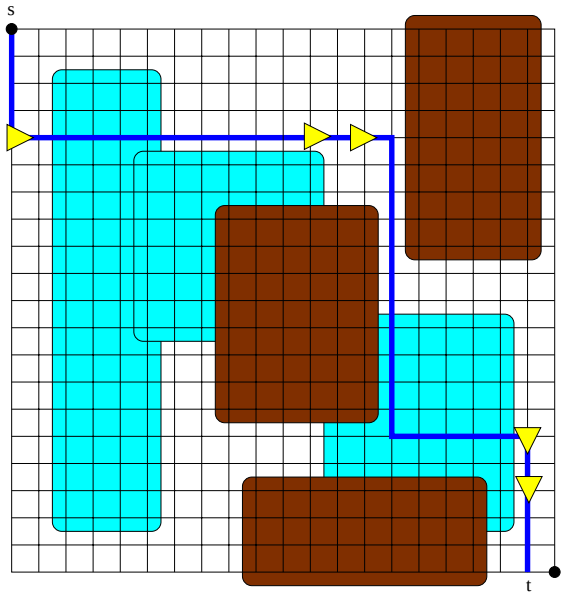

Figure 12: The minimum-delay buffered path under transition time constraint. For the waveform to have a fast transition, two buffers are inserted close to the target node. Transition time constraint is an important concern in maintaining signal integrity.

References

[1] C.J. Alpert, A. Devgan and S.T. Quay, Buffer Insertion with Accurate Gate and Interconnect Delay Computation, *Proc. ACM/IEEE Design Automation Conf.*, pp.479-484, 1999.

[2] C.J. Alpert, A. Devgan and S.T. Quay, Buffer Insertion for Noise and Delay Optimization, *Proc. ACM/IEEE Design Automation Conf.*, pp.362-367, 1997.

[3] C.-P. Chen and D.F. Wong, Optimal Wire-sizing Function with Fringing Capacitance Consideration, *Proc. ACM/IEEE Design Automation Conf.*, pp.604-607, 1997.

[4] C.-P. Chen and N. Menezes, Noise-Aware Repeater Insertion and Wire Sizing for On-Chip Interconnect Using Hierarchical Moment-Matching, *Proc. ACM/IEEE Design Automation Conf.*, pp.502-506, 1999.

[5] J. Culetu, C. Amir and J. MacDonald, A Practical Repeater Insertion Method in High Speed VLSI Circuits, *Proc. ACM/IEEE Design Automation Conf.*, pp.392-395, 1997.

[6] C.C.N. Chu and D.F. Wong, Closed Form Solution to Simultaneously Buffer Insertion/Sizing and Wire Sizing, *Int. Symp. on Physical Design*, pp.192-197, 1997.

[7] C.C.N. Chu and D.F. Wong, A New Approach to Simultaneous Buffer Insertion and Wire Sizing, *Proc. IEEE Int. Conf. on Computer Aided Design*, pp.614-621, 1997.

[8] W.C. Elmore, The Transient Response of Damped Linear Network with Particular Regard to Wide-band Amplifier, *Journal of Applied Physics*, vol.19, pp.55-63, 1948.

[9] Y. Gao and D.F. Wong, Wire-Sizing Optimization with Inductance Consideration Using Transmission Line Model, *IEEE Trans. on Computer-Aided Design*, 1999.

[10] Y. Gao and D.F. Wong, A Fast and Accurate Delay Estimation Model for Buffered Interconnects, *Proc. IEEE Asia and South Pacific Design Automation Conf.*, 2001.

[11] A.B. Kahng and S. Muddu, Delay Models for MCM Interconnects When Response is Non-monotone, *IEEE Multi-Chip Module Conf.*, pp.102-107, 1997.

[12] M. Lai and D. F. Wong, Maze Routing with Buffer Insertion and Wiresizing, *ACM/IEEE Design Automation Conference*, 2000.

[13] D. Li, A. Pua, P. Srivastava and U. Ko, A Repeater Optimization methodology for Deep Sub-Micron, High-Performance Processors, *Proc. IEEE Intl. Conf. on Computer Aided Design*, pp.726-731, 1997.

[14] J. Lillis, C.-K. Cheng and T.-T. Lin, Optimal and Efficient Buffer Insertion and Wire Sizing, *Proc. Custom Integrated Circuits Conf.*, pp.259-262, 1995.

[15] N. Menezes and C.-P. Chen, Spec-Based Repeater Insertion and Wire Sizing for On-Chip Interconnect, *Proc. Intl. Conf. on VLSI Design*, pp.476-483, 1999.

[16] P.R. O'Brien and T.L. Savarino, Modeling the Driving Point Characteristic of Resistive Interconnect for Accurate Delay Estimation, *Proc. IEEE Intl. Conf. on Computer Aided Design*, pp.512-515, 1989.

[17] P. Sarkar and C. Koh, Repeater Block Planning Under Simultaneous Delay and Transition Time Constraints, *IEEE Design, Automation and Test in Europe*, pp.540-545, 2001.

[18] J. Qian, S. Pullela and L. Pillage, Modeling the "Effective Capacitance" for the RC Interconnect of CMOS Gates, *IEEE Trans. on Computer-Aided Design*, Vol.13, No.12, pp.1526-1535, 1994.

[19] L.P.P.P. van Ginneken, Buffer Placement in Distributed RC-tree Networks for Minimal Elmore Delay, *Intl. Symp. Circuits and Systems*, pp.865-868, 1990.

[20] N.H.E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design* , 2nd Edition, Addison-Wesley Publishing Company, 1993.

[21] Q. Yu and E.S. Kuh, Exact Moment Matching Model of Transmission Lines and Application to Interconnect Delay Estimation, *IEEE Trans. on VLSI*, Vol.3, No.2, pp.311-322, 1995.

[22] H. Zhou et al., Simultaneous Routing and Buffer Insertion with Restrictions on Buffer Locations, *ACM/IEEE Design Automation Conf.*, 1999.

[23] J. Cong and X. Yuan, "Routing tree construction under fixed buffer locations", *ACM/IEEE Design Automation Conf.*, 2000.

[24] X. Tang, R. Tian, H. Xiang, and D. F. Wong, "A New Algorithm for Routing Tree Construction with Buffer Insertion and Wire Sizing under Obstacle Constraints", *Proc. IEEE Int. Conf. on Computer Aided Design*, 2001.