

# Modeling of Coplanar Waveguide for Buffered Clock Tree \*

Jun Chen

Electrical Engineering Department  
University of California, Los Angeles  
Los Angeles, CA 90025

Lei He

Electrical Engineering Department  
University of California, Los Angeles  
Los Angeles, CA 90025

**Abstract**—Owing to inductive effect, coplanar waveguide (CPW) is widely used to achieve signal integrity in high performance clock designs. In this paper, we first propose a piece-wise linear (PWL) model for the far-end response of a CPW considering ramp input and capacitive loading. The PWL model has a high accuracy but uses at least 1000x less time compared to SPICE. We then apply the PWL model to synthesize the CPW geometry for clock trees considering constrains of rising time and oscillation at sinks. We obtain a spectrum of solutions with smooth tradeoff between area and power.

## I. INTRODUCTION

The signal integrity in clock trees of GHz+ frequencies gains increasing importance due to inductive effects. Coplanar waveguide (CPW) sandwiches the clock signal line by two AC-grounded shielding wires (see fig.1), and can be used to effectively reduce the oscillation of clock signal [1, 2]. However, there is virtually no existing work on automatic synthesis of CPW structure for buffered clock trees. In this paper, we will develop an efficient yet accurate model for far-end response in a CPW, and use the model to synthesize buffer insertion solution and CPW geometry for a given clock tree topology.

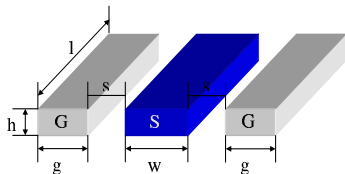


Fig. 1. Coplanar Waveguide Structure

It has been proposed in [2] that a CPW can be modeled by an equivalent transmission line with the following parasitics:

$$R = R_s + R_g/2 \quad (1)$$

$$L = L_s - 2L_{sg} + \frac{L_{gg}}{2} + \frac{L_g}{2} \quad (2)$$

$$C = 2C_{sg} + C_s \quad (3)$$

\*This paper is partially supported by NSF CAREER award CCR-0093273, SRC grant 1100, a UC MICRO grant sponsored by Analog Devices, Fujitsu Laboratories of America, Intel and LSI Logic, and a Faculty Partner Award by IBM. We used computers donated by Intel and SUN Microsystems. Address comments to lhe@ee.ucla.edu.

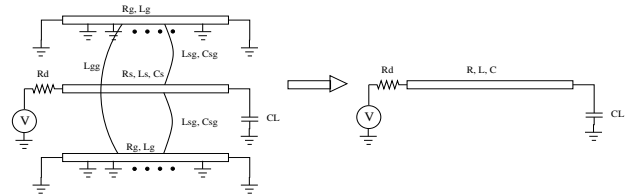


Fig. 2. Circuit models for CPW

where the parameters are shown in fig. 2. Existing works on transmission line model [3, 4, 5, 6] are not able to obtain accurate oscillation and far-end rising time with consideration of both capacitive loading and input rising time. Our first contribution in the paper is to develop a piece-wise linear (PWL) model for computation of waveform at the far-end of a single transmission line with consideration of capacitive loading and ramp input. The model can compute delay, rising time and noise with high accuracy but takes at least 1000X less time when compared to SPICE simulation.

A recent work [2] studied the ranges of geometrical parameters of CPW structure to ensure the minimal transmission delay and no oscillation. However, the tight constraints may lead to over-design and cost unnecessary power and area. Our second contribution of this paper is to apply the newly developed CPW model to synthesize the CPW geometry for clock trees with respect to relaxed constrains of bounded rising time and oscillation. We show that the min-area and min-power solutions are totally different, and obtain a spectrum of solutions for tradeoff between area and power. We also point out that there exists a knee point in the tradeoff curve, which leads to a desired solution with 5% more power but 60% less area compared to the min-power solution.

The rest of the paper is organized as follow: we present the PWL model in section II, and synthesize CPW-based clock trees in section III. We conclude in section IV with discussion of future work.

## II. PIECE-WISE LINEAR MODEL

The piece-wise linear model (PWL) computes the far-end response of a transmission line with capacitive loading for ramp input. It includes three steps: 1. transform the system to a new system without loading capacitance; 2. construct wave-

form for step input; 3. construct waveform for ramp input. We will briefly explain these steps in this section, more detailed derivation can be found in a technical report [7].

For clear explanation, we summarize the notations in table I. Generally, we use subscript ‘‘i’’ for notations related to the input, subscript ‘‘o’’ for those related to the far-end response, subscript ‘‘1’’ for those related to the far-end response resulting from the step input, and subscript ‘‘2’’ for those related to the far-end response resulting from the ramp input.

TABLE I  
NOTATIONS

w	width of clock signal wire
g	width of shielding wire
s	spacing between signal wire and shielding
l	length of CPW segment
$R_d$	driver resistance
$C_L$	loading capacitance
$t_{ri}$	input rising time
$t_f$	flight time of original transmission line
$t'_f$	flight time of the transmission line after mapping
$t_{do}$	delay at far-end
$t_{ro}$	rising time at far-end
$\bar{t}_{ro}$	upper bound of rising time at far-end
$V_i$	input waveform
$V_{o1}$	voltage response at far-end with step input
$V_{o2}$	voltage response at far-end with ramp input
$V_{osc}$	amplitude of oscillation at far-end
$\bar{V}_{osc}$	upper bound of amplitude of oscillation at far-end
$A$	area of CPW
$P$	power consumption of CPW
$O$	penalty function of oscillation violation at the far-end
$T$	penalty function of rising time violation at the far-end
$\alpha$	tradeoff factor between area and power
$\beta$	area of minimal driver
$\lambda$	balance factor between area and power
$k_b$	number of buffers
$C_w$	total capacitance of transmission line
$d$	size of buffer

### A. Consideration of Capacitive Loading

Based on the circuit model in fig.2, the transfer function at the far end of the wire is [8],

$$\begin{aligned} H(s) &= \frac{1}{(1 + sR_sC_L) \cosh(\theta) + (\frac{R_s}{Z_0} + sC_LZ_0) \sinh(\theta)} \\ &= \frac{1}{1 + \sum_{i=1}^{\infty} b_i s^i} \end{aligned} \quad (4)$$

where  $\theta = (R + sL)sC$ . The time of flight of the transmission line is  $t_f = \sqrt{LC}$ . To consider the loading capacitance in the model, we propose to transform the original circuit model with  $C_L$  to a new open-ended transmission line without  $C_L$  by matching their first two moments of the transfer functions. To do this, we modify the wire capacitance  $C$  and wire inductance

$L$  of the transmission line. The new transfer function of the circuit is

$$H'(s) = \frac{1}{\cosh(\theta') + \frac{R_s}{Z'_0} \sinh(\theta')} \quad (5)$$

Note that the  $\theta'$  and  $Z'_0$  in (5) are different from  $\theta$  and  $Z_0$  in (4). By matching the first two moments of (4) and (5), we obtain the new wire capacitance  $C'$  and wire inductance  $L'$  as,

$$C' = \frac{b_1}{R_d + \frac{R}{2}} \quad (6)$$

$$L' = \frac{2 \left( b_2 - \frac{R^2 C'^2}{24} - \frac{R_d R C'^2}{6} \right)}{C'} \quad (7)$$

where  $b_1$  is defined in (4). The time of flight of the mapped line is,

$$\begin{aligned} t'_f &= \sqrt{L'C'} \\ &= \frac{\sqrt{(LC + \frac{R^2 C^2}{12} + R_d R C_L C + \frac{(R_d C + C_L R) R C}{3} + 2C_L L - \frac{R^2 C'^2}{12} - \frac{R_d R C'^2}{3})}}{3} \end{aligned} \quad (8)$$

We will use the  $t'_f$  in our model later on. Normally  $t'_f > t_f$ , but when  $C_L$  and in turn  $C'$  is sufficient large,  $t'_f$  may be smaller than  $t_f$ . In this case,  $t'_f$  is not physically meaningful. However, because of the large capacitive loading, the circuit becomes capacitive dominant in this case. Naturally, we can just match the first moment and obtain,

$$C' = \frac{b_1}{R_d + \frac{R}{2}} \quad (9)$$

$$\Rightarrow t'_f = \sqrt{C'L} \quad (10)$$

$L$  will be the same in this special case. Because  $C' > C$ ,  $t'_f > t_f$  holds.

### B. PWL Model with Step Input

After mapping, the system is an open-ended transmission line, thus it can be solved by the formula from [4]. The formula is based on the series of modified Bessel function and provides a closed-form solution. However, directly applying the algorithm results in steep rising at  $t = (2n + 1)t'_f$ , which is far from true due to the loading capacitance. Furthermore, it is not efficient to compute the entire waveform simply by time stepping. Thus we develop a PWL model to approximate the waveform and efficiently compute delay, rising time, overshoot and undershoot.

Our algorithm works as follows: we first compute the waveform slopes at  $n \cdot t'_f$ ,  $n = 1, 2, \dots$ . Then we draw straight lines passing through these points with the calculated slopes. Finally, we obtain the crossing points of directly adjacent lines, and approximate the waveform by connecting these crossing points. Fig.3 illustrates the process.

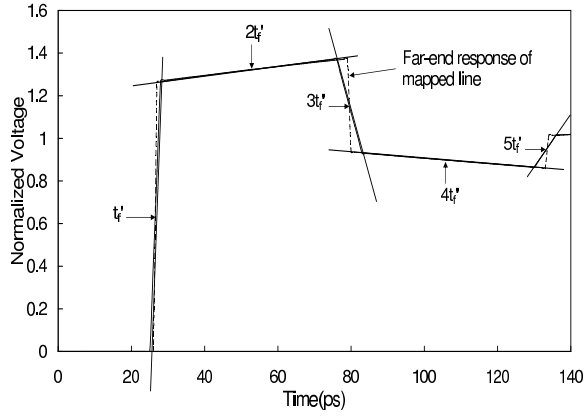


Fig. 3. Illustration of piece-wise linear model.

The above algorithm is justified by the following observations. Owing to the reflection from the far end, the waveform can be divided into regions  $(0, t'_f)$ ,  $(t'_f, 3t'_f)$ ,  $(3t'_f, 5t'_f)$ ,  $\dots$ . The waveform changes quickly only at the boundary of these regions but not inside these regions. Therefore, we can use one line to approximate the waveform at the reflection point  $t'_f$ , and use two lines to approximate the waveform in each region starting from  $(t'_f, 3t'_f)$ . One line passes through the middle point (e.g.,  $2t'_f$ ) in the region, and the other passes through the next reflection time point (e.g.,  $3t'_f$ ).

In the following, we explain how to compute the slopes. Without losing generality, we assume input signal rising from 0 to  $V_{dd}$ . In fig.4, we illustrate the computation of the slope at  $t'_f$ . We approximate the time where the voltage reaches the 50% of the amplitude of this rise at  $t'_f$ , of which the starting point of the rise is at  $t_f$ , the flight time without considering the loading capacitance. From this approximation, we obtain the slope at  $t'_f$  as

$$s_1 = \frac{V_{o1}(t'_f + \delta)}{2(t'_f - t_f)}. \quad (11)$$

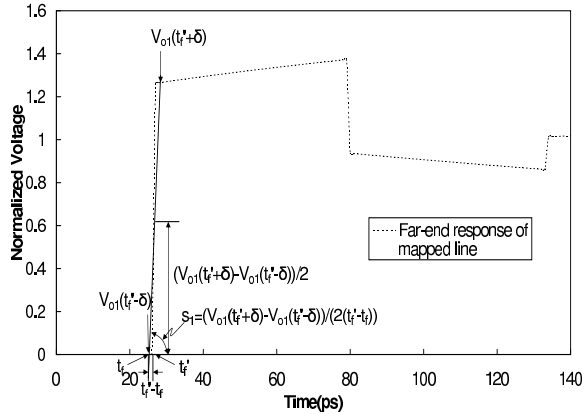


Fig. 4. Construction of piece wise linear model.

We directly solve the slope at  $2t'_f$  for the region  $(t'_f, 3t'_f)$  as

$$s_2 = \frac{dV_{o1}(2t'_f)}{dt} = \frac{V_{o1}(2t'_f + \delta) - V_{o1}(2t'_f - \delta)}{2\delta}. \quad (12)$$

In this case, the approximating line is the tangent line at  $2t'_f$ .

Because at  $3t'_f$  the reflected wave travels twice along the line after  $t'_f$ , we approximate the time for the waveform to reach 50% of the falling by  $2(t'_f - t_f)$ . Therefore the slope at  $3t'_f$  is

$$s_3 = \frac{\frac{V_{o1}(3t'_f + \delta) - V_{o1}(3t'_f - \delta)}{2}}{2(t'_f - t_f)} \quad (13)$$

The rest of the regions are calculated in the similar fashion: Regions  $((2n - 1)t'_f - \delta, (2n - 1)t'_f + \delta)$  are similar to the region  $(3t'_f - \delta, 3t'_f + \delta)$ , where the slope is

$$s_{2n-1} = \frac{\frac{V_{o1}((2n-1)t'_f + \delta) - V_{o1}((2n-1)t'_f - \delta)}{2}}{2(t'_f - t_f)}. \quad (14)$$

Regions  $((2n - 1)t'_f, (2n + 1)t'_f)$  are similar to the region  $(t'_f, 3t'_f)$ , where the slope is

$$s_{2n} = \frac{dV_{o1}((2n)t'_f)}{dt} = \frac{V_{o1}((2n)t'_f + \delta) - V_{o1}((2n)t'_f - \delta)}{2\delta}, \quad (15)$$

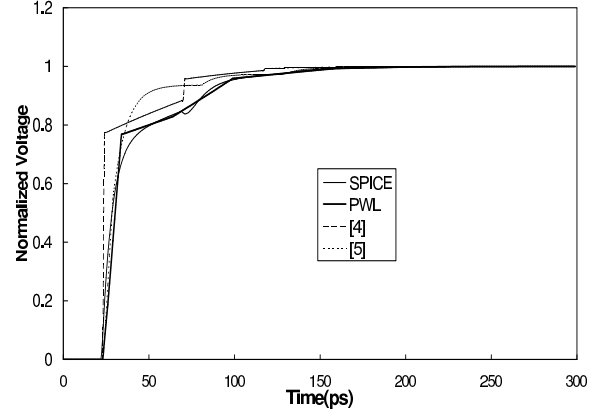


Fig. 5. Overdamped far-end waveform of  $l = 3000\mu m$ ,  $w = 10\mu m$ ,  $g = 8\mu m$ ,  $h = 1\mu m$ ,  $s = 2\mu m$ ,  $R_d = 40\Omega$ ,  $C_L = 0.2pF$ . Input is step input.

In fig.5 and 6, we compare the waveforms from different models and SPICE. Our model obtains results that match SPICE simulations very well in both overdamped (see fig.5) and underdamped (see fig.6) cases. Our model slightly deviates from SPICE simulation around the knee points but the error is small. The waveform from either [4] or [5] can not match the SPICE simulation.

### C. PWL model with ramp input

We now extend our model to consider the ramp input with rising time  $t_{ri}$ . Because of the extra knee point in the ramp input, the regions of the far-end waveform for the step input need

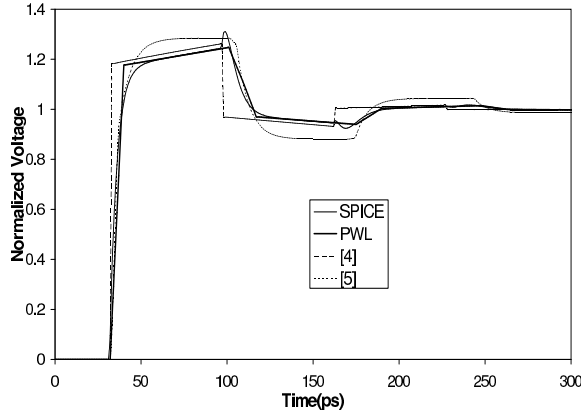


Fig. 6. Underdamped far-end waveforms of  $l = 3000\mu m$ ,  $w = 20\mu m$ ,  $g = 10\mu m$ ,  $h = 1\mu m$ ,  $s = 0.6\mu m$ ,  $R_d = 12\Omega$ ,  $C_L = 0.2pf$ . Input is step input

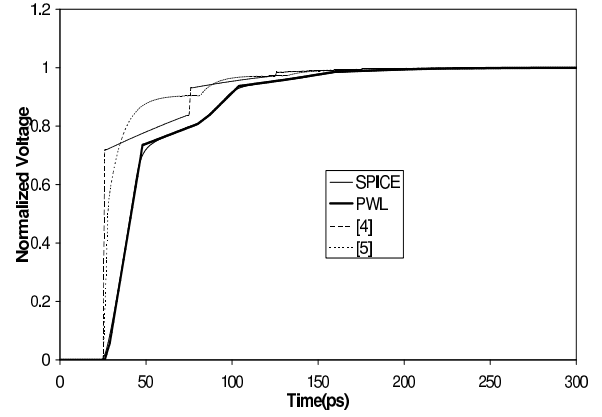


Fig. 7. Overdamped far-end waveforms of  $l = 3000\mu m$ ,  $w = 20\mu m$ ,  $g = 15\mu m$ ,  $h = 1\mu m$ ,  $s = 0.6\mu m$ ,  $R_d = 60\Omega$ ,  $C_L = 0.2pf$ . Input rising time is 20ps.

to be further divided according to  $t_{ri}$ . We find the voltage and slope at  $\frac{t_1+t_2}{2}$ . For each pair of two adjacent time points  $t_1$  and  $t_2$  in the set of  $\{(2n+1)t'_f, (2n+1)t'_f + t_{ri}\}$ , ( $n = 1, 2, \dots$ ), then approximate the waveform by a straight line at  $\frac{t_1+t_2}{2}$  with the computed slope. The entire waveform can be approximated by connecting the crossing points of directly adjacent lines.

Next, we discuss how to compute voltage and slope. From the linear circuit theory[2], the waveform at the far end of the transmission line resulting from the ramp input is

$$\begin{aligned} V_{o2}(t) &= \int_{-\infty}^{\infty} V_{o1}(t) \frac{dV_i(t-\tau)}{dt} dt \\ &= \frac{1}{t_{ri}} \int_{t-t_{ri}}^t V_{o1}(t) dt \end{aligned} \quad (16)$$

Because we have already obtained the PWL waveform  $V_{o1}$  for the step input in section B, we can compute the slope and voltage value efficiently without computation of the series of modified Bessel functions. According to (16) we can compute the slope as

$$\frac{dV_{o2}(t)}{dt} = \frac{V_{o1}(t) - V_{o1}(t - t_{ri})}{t_{ri}} \quad (17)$$

and the voltage value as

$$V_{o2}(t) = \frac{1}{t_{ri}} \sum_{(t_i, t_{i+1}) \subseteq (t-t_{ri}, t)} \frac{V_{o1}(t_i) + V_{o1}(t_{i+1})}{2} (t_{i+1} - t_i) \quad (18)$$

where  $(t_i, t_{i+1})$  is a linear piece in the PWL expression of  $V_1(t)$ . Thus the extension to ramp input is extremely efficient.

We compare waveforms from different models and SPICE simulations in fig.7 and 8. From the figures, we can see that our model again matches SPICE simulation very well in both overdamped case (fig.7) and underdamped case (fig.8). The waveform from [4] and [5] differs a lot from the SPICE simulation results.

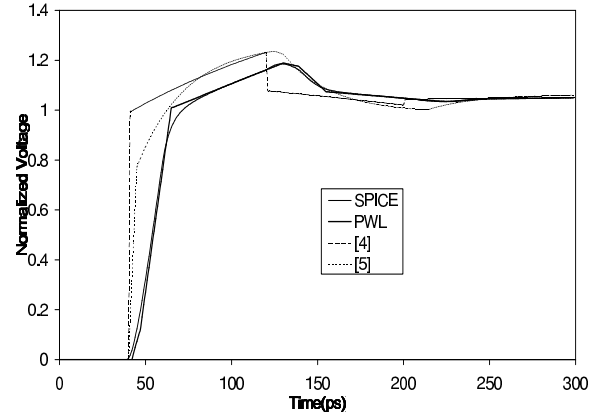


Fig. 8. Underdamped far-end waveforms of  $l = 5000\mu m$ ,  $w = 10\mu m$ ,  $g = 5\mu m$ ,  $h = 1\mu m$ ,  $s = 1\mu m$ ,  $R_d = 15\Omega$ ,  $C_L = 0.2pf$ . Input rising time is 20ps.

#### D. Calculation of delay, rising time and oscillation

Because of the sequential property of the construction procedure of PWL model, calculation of delay, rising time and amplitude of oscillation can be easily implemented in a need-based procedure. A knee point is calculated only if it is needed by the calculation of delay, rising time and oscillation. The maximum overshoot will happen around  $3t'_f$ , and so calculating the knee points up to  $4t'_f$  is needed. Similarly, maximum undershoot will happen around  $5t'_f$ , thus we only need to calculate the regions up to  $6t'_f$ . To estimate the delay  $t_{do}$  and  $t_{ro}$ , we just need to calculate the knee points till the voltage meet the corresponding bound, for example 90% for  $t_{ro}$ .

#### E. Time complexity and accuracy

We present sample CPW structures in table II and summarize the runtime and compare different models in terms of oscillation, delay and rising time in table III. We compare our method with SPICE simulation and the models from [5] and

TABLE III  
 RUNTIME AND RESULTS FROM DIFFERENT MODELS. SPICE AND [4] CALCULATE UP TO 300ps BY TIME STEPPING (1ps/STEP).

Model		runtime (s)				50% delay (ps)				rising time (ps)				amplitude of oscillation (%Vdd)			
setting	type	SPICE	PWL	[5]	[4]	SPICE	PWL	[5]	[4]	SPICE	PWL	[5]	[4]	SPICE	PWL	[5]	[4]
1	underdamped	88.10	0.01	0.01	0.18	24	25	25	24	10	8	9	6	4.6	4.5	9.2	5.1
2	overdamped	148.10	0.01	0.01	0.18	42	42	42	41	83	83	46	80	0	0	0	0
3	underdamped	368.23	0.01	0.01	0.12	83	84	83	80	58	56	48	48	8.6	8.9	10.3	8.8
4	overdamped	23.23	0.01	0.01	0.73	33	33	12	9	47	47	26	26	0	0	0	0
5	underdamped	121.39	0.01	0.01	0.20	55	55	39	38	26	26	10	1	4.6	5.2	11.3	8.0
6	underdamped	344.70	0.01	0.01	0.02	112	113	96	93	28	25	26	1	13.5	14.2	16.7	15.7

[4]. Both our model and [5] are at least 1000× faster than SPICE, and [4] is about 100× faster than SPICE. Our model is accurate compared to SPICE simulation. The error of delay and noise is less than 10%, and the error of rising time is less than 20% in the worst case. The PWL model sometimes obtains smaller rising time compared to SPICE simulation. This is because the time point of 90% Vdd happens to be around the knees. The error is normally less than 20% however. In the contrast, both [5] and [4] can introduce huge errors in delay, rising time and oscillation extraction. The error of [5] can be up to 90% for amplitude of oscillation and 50% for rising time assuming step input. The model is much worse in the case of ramp input. [4] also has up to 40% error for the step input cases and up to 90% error for ramp input.

TABLE II  
 SAMPLE EXPERIMENT SETTINGS (ALL GEOMETRIES ARE IN μm)

setting	l	w	s	g	R <sub>d</sub> (Ω)	C <sub>L</sub> (fF)	t <sub>ri</sub> (ps)
1	3000	6	1	4	30	45	0
2	5000	10	2	5	40	45	0
3	10000	8	2	8	24	90	0
4	1000	8	1	4	60	90	30
5	5000	10	2	10	24	45	30
6	10000	10	1	10	24	90	30

### III. POWER AND AREA OPTIMIZATION FOR CLOCK

The on-chip clock trees consume significant portion of chip area and power. In this section, we use the PWL model to optimize the power and area for the CPW-based clock tree. We define the noise  $V_{osc}$  as the difference between maximal overshoot and maximal undershoot, and rising time  $t_{ro}$  as the time between the moments when voltage reaches 10% Vdd and 90% Vdd respectively. Our clock optimization considers constraints of  $t_{ro}$  and  $V_{osc}$  at clock sinks.

#### A. Objective function

To handle multiple objectives and multiple constraints simultaneously, we choose to minimize a weighted sum of area, power, and penalties of rising time and oscillation violations. With respect to notations in table I, the area of a CPW segment with driver size of  $d$  is,

$$A = l \cdot (w + 2s + 2g) \cdot k_b \cdot \beta \cdot d \quad (19)$$

where  $d$  is the size of buffer, and  $\beta$  is a constant to adjust the relative importance of interconnect area versus device area. Our experiment uses  $\beta=0.01$  as the chip area is mainly decided by the routing area. Because we only consider dynamic power, power is defined as the total capacitance, i.e.,

$$P = k_b \cdot (C_w + C_L) \quad (20)$$

The penalty of the rising time violation is defined as

$$T = \begin{cases} T_{ro} - \overline{T_{ro}} & , T_{ro} > \overline{T_{ro}} \\ 0 & , otherwise \end{cases} \quad (21)$$

Clearly, there is no penalty when there is no violation. Similarly, the penalty of the oscillation violation is

$$O = \begin{cases} V_{osc} - \overline{V_{osc}} & , V_{osc} > \overline{V_{osc}} \\ 0 & , otherwise \end{cases} \quad (22)$$

Then, the objective function is defined as

$$F = \alpha \cdot \lambda \cdot A + (1 - \alpha) \cdot P + \mu \cdot O + \nu \cdot T \quad (23)$$

where  $\alpha$ ,  $\lambda$ ,  $\mu$  and  $\nu$  are weight constants.  $\alpha$  controls the tradeoff between power and area, and is specified by the designer.  $\lambda$  is introduced to balance the different orders of magnitude of  $A$  and  $P$ . It is decided by the ratio of power and area of a sample circuit, and is 0.1 in our experiment. To ensure that the final solution has no rising time and oscillation violations, we use large values for  $\mu$  and  $\nu$ .

#### B. Buffered Tree

We apply our algorithm to optimize the clock tree with fixed buffer placement. The objective function is (23), considering all CPW segments in the clock tree for power and area. We enforce the oscillation constraint at all the buffers, but only enforce the constraint of rising time at the sinks. The input rising time at a driver/buffer is the output rising time of its previous stage. We determine the optimal solution of signal wire width  $w$ , shielding  $g$  and spacing  $s$  of each wire segment, and determine buffer size of each buffer, such that the objective function (23) is minimized.

Our experiment assumes a symmetric H-tree in figure 9. The input rising time is 30ps, and the rising time constraint at the sink is 75ps. The noise constraint at each driver/buffer is 5% Vdd. The receiver at the sink has a fixed size of 25×. The allowed driver/buffer size is [1×, 500×]. Our algorithm adjusts

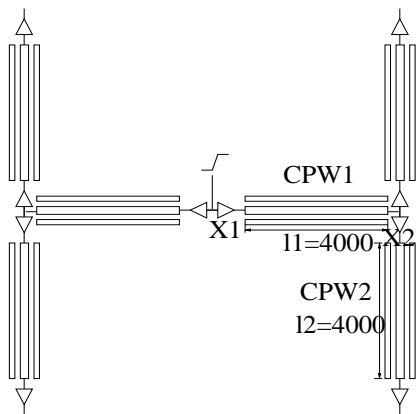


Fig. 9. A simple H-tree.

the driver sizes of  $X1$  and  $X2$ , and geometries of  $CPW1$  and  $CPW2$ . We use a simulated annealing algorithm to optimize the area and power of the H-tree.

Fig.10 presents the tradeoff between the area and power of the H-tree obtained by our algorithm. The min-area solution has 50% more power than the min-power solution, but the min-power solution has 200% more area than the min-area solution. There also exists a knee point around  $\alpha = 0.3$ , which leads to a desired design with 10% more power but 50% less area compared to the min-power solution. We show the geometry optimization results in table IV.

TABLE IV  
EXPERIMENT RESULTS WITH DIFFERENT TRADEOFF FACTORS  
FOR A BALANCED H-TREE (ALL GEOMETRIES ARE IN  $\mu m$ ).

$\alpha$	x1	w1	s1	g1	x2	w2	s2	g2	power
0	254	2.2	9.5	8.6	137	1.2	12.9	2.7	1786fF
0.3	360	2.4	4.5	4.4	153	1.2	6.0	2.4	1938fF
1	500	5.88	2.3	4.0	250	1.8	2.5	1.5	2770fF

The tradeoff in this experiment is mainly decided by the buffer size. Larger buffers enable narrower CPW for satisfying the constraints, which helps reduce area because the chip area is mainly determined by routing area. However, the narrower spacing and larger buffers introduce larger capacitance and in turn higher power.

#### IV. CONCLUSION

In this paper, we have developed an efficient model for the far-end response at a coplanar waveguide (CPW) line with capacitive loading and ramp input. This model is highly accurate compared to SPICE simulation but is at least 1000x faster. We have also applied the model to minimize power and area in a buffered clock tree. We have shown that there exist knee points in the area-power curves, and such knee points lead to the desired solutions with slightly higher power but much reduced area compared to the solutions with the minimum power. In our future, we plan to extend our model to consider the nonlin-

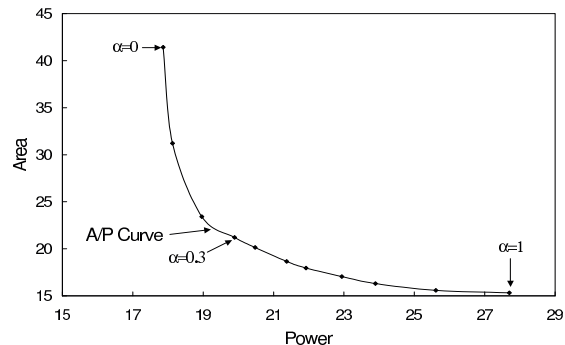


Fig. 10. Tradeoff between area and power of H-tree.

earity of drivers, and develop optimization algorithms to handle more design freedoms in a highly efficient fashion.

#### REFERENCES

- [1] N. Chang, S. Lin, L. He, O. S. Nakagawa, and W. Xie, "Clocktree RLC extraction with efficient inductance modeling," in *Design Automation and Test in Europe*, March 2000.
- [2] R. Escovar and R. Suaya, "Transmission line design of clock trees," in *Proc. Int. Conf. on Computer Aided Design*, 2002.
- [3] A. Kahng and S. Muddu, "An analytical delay model for rlc interconnects," in *Proc. IEEE Int. Symp. on Circuits and Systems*, 1996.
- [4] J. A. Davis and J. D. Meindl, "Compact distributed rlc interconnect models. I. single line transient, time delay, and overshoot expressions," *IEEE Transactions on Electron Devices*, pp. 2068–2077, November 2000.
- [5] Y. Eo, J. Sim, and W. R. Eisenstadt, "A traveling-wave-based waveform approximation technique for the timing verification of single transmission lines," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 6, pp. 723–730, 2002.
- [6] R. Venkatesan, J. A. Davis, and J. D. Meindl, "A physical model for the transient response of capacitively loaded distributed rlc interconnects," in *Proc. Design Automation Conf*, 2002.
- [7] J. Chen and L. He, "Modeling and synthesis of coplanar waveguide for buffered clock tree," Tech. Rep. ENG 03-242, UCLA, Nov. 2003.
- [8] H. You and M. Soma, "Crosstalk analysis of interconnection lines and packages in high-speed integrated circuits," *IEEE Trans. on Circuits and Systems*, pp. 1019–1026, August 1990.