Title

Fast Transient Simulation of Lossy Transmission Lines

Authors

Peng, He
Cheng, Chung-Kuan

Publication Date

2006-12-13

Peer reviewed

Fast Transient Simulation of Lossy Transmission Lines

He (Vincent) Peng, Chung-Kuan Cheng
Dept. of Computer Science and Engineering
Univeristy of California, San Diego
La Jolla, CA 92037-0404
Email: hepeng, kuan@cs.ucsd.edu

Abstract

In this paper, an efficient approach is proposed for the problem of transient simulation of lossy transmission lines. The complexity of the conventional convolution approach for lossy transmission lines is reduced from $O(N^2)$ to $O(Nlog^2N)$ by utilizing a multilevel FFT convolution method, where N is the total number of time points. Numerical convolution formula that exploits both the analytical forms of the lossy transmission line impulse responses and adaptive time steps are developed for the multilevel FFT convolution method. A new breakpoint control scheme is also proposed to adaptively control the time step. Experimental results show the proposed approach is over 100 times faster than berkeley SPICE3 [1, 4] while remains the same accuracy.

1. INTRODUCTION

With increasing design complexity, gigahertz level operating frequencies and increasing interconnect length, interconnects dominate the performance and noise immunity in high-speed designs. Electrical length of interconnects, both on-chip and off-chip, become significant fraction of the signal wavelength and, hence, interconnects can no longer be modeled as lumped circuits. Instead, distributed transmission line models should be used. Moreover, interconnects at chip, package, and board level become the most critical parts for the signal integrity of the whole system. Interconnect effects such as ringing, signal delay, distortion, reflections, and crosstalk could distort an analog signal such that it fails to meet specifications or they could cause logic glitches that may fail a design [3]. System-level performance issues such as transmission lines and crosstalk are getting more and more important. Hence it is extremely important to efficiently and accurately simulate circuits that contains transmission lines.

The main difficulty of transient simulation of transmission lines is the mixed time/frequency domain problem: when transmission line circuits contain nonlinear devices such as transistors and diodes, these nonlinear devices must be characterized in time domain while transmission lines are best characterized in frequency domain.

The first group of methods for transmission line simulation is the direct convolution approach [4–7]. First, frequency domain description of the transmission line is transformed into time domain using inverse Fast Fourier Transform (FFT) [6], Numerical Inversion of Laplace Transform (NILT) [7], or analytical approach [4, 5]. Then convolution and nonlinear iterations are performed in time domain to solve the circuit. This group of methods has the highest accuracy. However, these approaches are computationally expensive because the convolution operation needs to extend over the entire history, which leads to a total computation time of $O(N^2)$.

Another category of methods for transmission line simulation approximates the characteristics of transmission lines by rational functions [10–17]. This kind of methods has the advantages that it eliminates the use of FFT or NILT and allows an efficient evaluation of the time domain convolution. Once the rational function approximation is obtained, recursive convolution [11, 13–17] is used to efficiently evaluate the time-domain convolution in linear time. For this category of methods, higher order rational function approximations are needed when accuracy is the main concern, which would make these methods less efficient or even impractical.

The third category is the discrete lumped model methods which use modal order reduction techniques [20, 21] to approximate the infinite order of the distributed transmission line network. This kind of approaches is the fastest, but has the lowest accuracy.

Between the tradeoff of accuracy and efficiency, recent trend in VLSI industry has made accuracy the main concern for the simulaiton of transmission line circuits. Thus the first category of methods, the direct convolution methods, is prefered. However, the quardratic complexity of the direct convolution methods needs to be improved.

In [2], the first category of methods is improved by using an efficient convolution algorithm that based on the combination of FFT and conventional convolution to reduce the complexity of the convolution operation from $O(N^2)$ to $O(Nlog^2N)$. However, this approach requires a uniform time step to perform the FFT convolution operation, which means that the smallest time step must be used when this approach is integrated in a circuit simulator, such as SPICE3 [1, 4]. The smallest time step used by a simulator would be orders of magnitude smaller than the average time step used, which would make this approach impractical.

In this paper, we present a fast multilevel FFT convolution algorithm for the transient simulation of lossy transmission lines which extends [2]. The main contribution of this paper is the use of our new average based numerical convolution formula combined with the FFT convolution method proposed in [2]. Our average based numerical convolution formula exploits both the analytical forms of the transmission line impulse responses [4, 5] and adaptive time steps. Which makes it possible to use adaptive time steps with the FFT convolution method in [2]. The step size mismatch between the time step used by the circuit simulator and sampling interval of the FFT convolution is handled accurately by using average values of the input function within the FFT sampling interval instead of the input function itself. The sharp slope region of the transfer function is treated specially to ensure numerical stability and accuracy. A new breakpoint control scheme is also presented to adaptively select simulation time steps.

The rest of this paper is organized as follows. Backgrounds of conventional convolution simulation of simple lossy transmission line are introduced in section 2. Section 3 presents our multilevel FFT convolution algorithm. Section 4 discusses dynamic time step control. Experimental results along with comparisons with SPICE3 [4] are presented in section 5. Finally, the paper concludes with conclusions and possible future directions in section 6.

2. CONVENTIONAL CONVOLUTION SIMULATION OF LOSSY INTERCONNECTS

2.1 Derivation of the Convolution Equations

The telegrapher equations that describe the transient behavior of a lossy line are

$$\frac{\partial v}{\partial x} = -(L\frac{\partial i}{\partial t} + Ri) \quad (1)$$

$$\frac{\partial i}{\partial x} = -(C\frac{\partial v}{\partial t} + Gv) \quad (2)$$

with x varying between 0 and l, where l is the length of the transmission line. Initial conditions are assumed to be zero. In conventional two-port notation, the port variables are: $v_1(t) = v(0,t), i_1(t) = i(0,t), v_2(t) = v(l,t), i_2(t) = -i(l,t)$.

By taking Laplace transforms on (1) and (2), we have

$$\frac{\partial V}{\partial x} = -(sL + R)I \quad (3)$$

$$\frac{\partial I}{\partial x} = -(sC + G)V \quad (4)$$

The solution of (3) and (4) is rearranged [5] to arrive:

$$V_2(s)Y_0\frac{Y(s)}{Y_0} - I_2(s) = V_1(s)Y_0\frac{Y(s)}{Y_0}e^{-\lambda(s)l} + I_1(s)e^{-\lambda(s)l} \quad (5)$$

$$V_1(s)Y_0\frac{Y(s)}{Y_0} - I_1(s) = V_2(s)Y_0\frac{Y(s)}{Y_0}e^{-\lambda(s)l} + I_2(s)e^{-\lambda(s)l} \quad (6)$$

where $Y(s) = \sqrt{\frac{sC+G}{sL+R}}$ and $Y_0 = \sqrt{\frac{C}{L}}$. Apply inverse Laplace Transform to (5) and (6) and let:

$$h_1(s) = \mathscr{L}^{-1}\{\frac{Y(s)}{Y_0}\} \quad (7)$$

$$h_2(s) = \mathscr{L}^{-1}\{e^{-\lambda(s)l}\} \quad (8)$$

$$h_3 = \mathscr{L}^{-1}\{\frac{Y(s)}{Y_0}e^{-\lambda(s)l}\} \quad (9)$$

we get the following two equations:

$$Y_0v_2(t) * h_1(t) - i_2(t) = Y_0v_1(t) * h_3(t) + i_1(t) * h_2(t) \quad (10)$$

$$Y_0v_1(t) * h_1(t) - i_1(t) = Y_0v_2(t) * h_3(t) + i_2(t) * h_2(t) \quad (11)$$

where * denotes convolution. Analytical solutions of transfer functions $h_1(t)$, $h_2(t)$, and $h_3(t)$ were derived in [5]. These two equations characterize the transmission line two-port network.

2.2 Numerical Convolution

In general, the convolution to be calculated is:

$$y(t) = \int_0^t x(\tau)h(t-\tau)d\tau \quad (12)$$

where $y(t)$ is the output, $h(t)$ is the transfer function ($h_1(t)$, $h_2(t)$, or $h_3(t)$), and x(t) is the input ($v_1(t)$, $v_2(t)$, $i_1(t)$, or $i_2(t)$), which is only available in previous times. At each simulation time point, we need to calculate this convolution integral and each integral needs to extend over all previous time points. For the sake of simplicity, we use the Backward Euler rule to calculate this integral numerically although high-order method is used in practice. For instance, a generalized trapezoidal method is used in [5]. If the simulation time points are (t_0,t_1,...,t_N), then at time point t_k, the convolution integral is approximated by:

$$y(t_k) = \int_0^{t_k} x(\tau)h(t_k-\tau)d\tau = \sum_{i=0}^{k-1}\int_{t_i}^{t_{i+1}} x(t_i)h(t_k-\tau)d\tau$$

$$\approx \sum_{i=1}^{k-1} x(t_i)h(t_k-t_i)\triangle_i + x(t_k)h(0)\triangle_k \quad (13)$$

where $\triangle_i = t_{i+1} - t_i$. Equation (13) can also be written as $y(t_k) \approx C_1 + C_2x(t_k)$, where C_1 and C_2 can be calculated from (13) since the only unknown in (13) is $x(t_k)$. As the simulation proceeds, k progresses from 1 to N, if we calculate the in Equation (13) for each k directly, the total computation time is $O(N^2)$. Because $x(t_k)$ is unknown at time point t_k, we can not use FFT to calculate the convolution for the reason that FFT requires the entire sequence of $\{x(t_i)\}$.

At a given time point t_k, the only unknowns in equations (10) and (11) are $v_1(t_k)$, $v_2(t_k)$, $i_1(t_k)$, and $i_2(t_k)$. To solve for these four unknowns for this time point, convolutions in (10)(11) are calculated and Modified Nodal Analysis (MNA) is used to obtain two more (KCL) equations at both the near end and the far end of the transmission line. Then these four equations, together with the MNA equations from other components in the circuit, are loaded into the circuit simulator and solved by the matrix solver in the simulator to get $v_1(t_k)$, $v_2(t_k)$, $i_1(t_k)$, and $i_2(t_k)$. Thus the computation of the convolutions in equations (10)(11) dominates the execution time. For this reason, we propose our fast multilevel FFT convolution algorithm in section 3.

3. THE MULTILEVEL FFT CONVOLUTION ALGORITHM

3.1 Multilevel FFT Convolution

A multilevel FFT convolution method is proposed in [2] to efficiently calculate the convolution in $Nlog^2N$ time.

Suppose the time duration of $x(t)$ and $h(t)$ are T. First break $x(t)$ and $h(t)$ into halves, define:

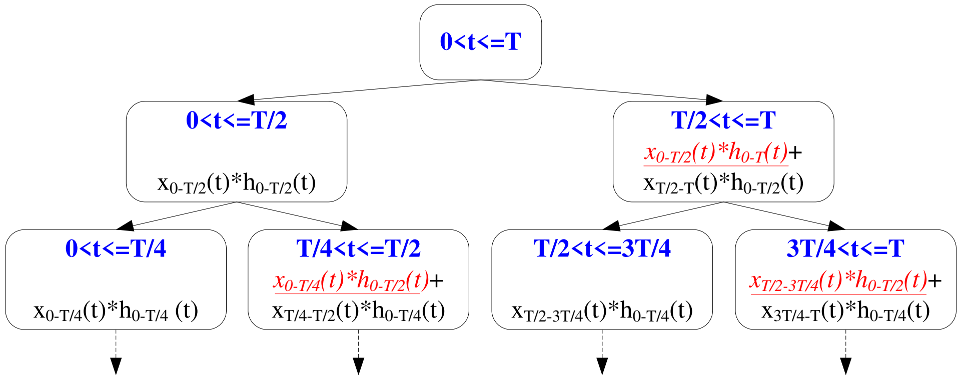

Figure 1: An example of the multilevel FFT convolution algorithm

$$x_a(t) = \begin{cases} x(t) & t \in (0, \frac{T}{2}] \\ 0 & t \in (\frac{T}{2}, T] \end{cases}, x_b(t) = \begin{cases} 0 & t \in (0, \frac{T}{2}] \\ x(t) & t \in (\frac{T}{2}, T] \end{cases}$$

and

$$h_a(t) = \begin{cases} h(t) & t \in (0, \frac{T}{2}] \\ 0 & t \in (\frac{T}{2}, T] \end{cases}, h_b(t) = \begin{cases} 0 & t \in (0, \frac{T}{2}] \\ h(t) & t \in (\frac{T}{2}, T] \end{cases}$$

Then the convolution could be calculated as:

$$y(t) = x(t) * h(t) = [x_a(t) + x_b(t)] * [h_a(t) + h_b(t)]$$

$$= x_a(t)*h_a(t)+x_a(t)*h_b(t)+x_b(t)*h_a(t)+x_b(t)*h_b(t) \quad (14)$$

1. For $0 < t \leq \frac{T}{2}$, only $x_a * h_a$ is nonzero and we have
$$f(t) = x_a(t) * h_a(t) \quad (15)$$
2. For $\frac{T}{2} < t \leq T$, $x_b * h_b = 0$, we have
$$f(t) = x_a(t) * h(t) + x_b(t) * h_a(t) \quad (16)$$
where $x_a(t)*h(t)$ could be calculated by FFT in $O(NlogN)$ time since all values of $x_a(t)$ and $h(t)$ are known.
3. For $t > T$, the convolution $x(t) * h(t)$ can be calculated by FFT.

To calculate $x_a(t) * h_a(t)$ and $x_b(t) * h_a(t)$, we can further divide them into time sequences of duration $\frac{T}{4}$, recursively apply the same idea whenever possible. Thus the complexity of the above method is $O(Nlog^2N)$:

$$O(Nlog^2N) = O(NlogN + 2(\frac{N}{2}log\frac{N}{2}) + 4(\frac{N}{4}log\frac{N}{4}) + ...) \quad (17)$$

An example of this multilevel convolution algorithm is shown in Fig. 5 where the convolutions underlined can be calculated by FFT while other convolutions can be further divided. FFT requires the step size used to sample the waveforms of $x(t)$ and $h(t)$ to be uniform.

3.2 Average Based Numerical Convolution Formula

In this section we will show how to use FFT to calculate the convolution when both $x(t)$ and $h(t)$ are known. A new average based numerical convolution formula is derived for this FFT convolution approach.

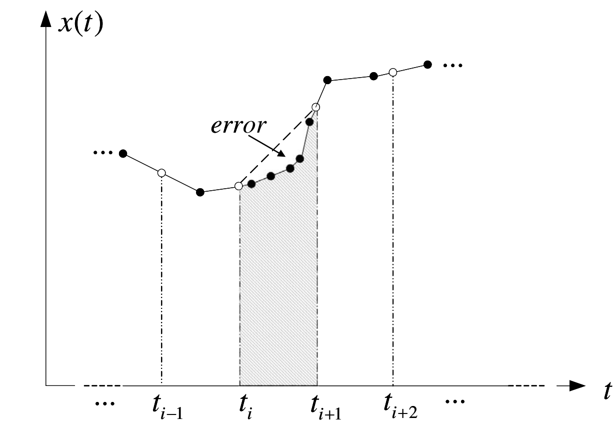

Figure 2: Mismatch between the FFT Sampling Points and Simulation Time Points

In order to use FFT to calculate the convolution, first we need to discretize $x(t)$ and $h(t)$. FFT requires an uniform step size to sample both $x(t)$ and $h(t)$, as shown in Fig. 2 where the time interval $[0, t_k]$ is discretized using a uniform step size $\triangle$ such that $t_{i+1} - t_i = \triangle, i = 0, ..., k-1$. These sampling time points are shown as circles in the figure.

The $x(t)$ values are calculated by the circuit simulator at time points $t'_0, t'_1,...,t'_j$ with non-uniform time step sizes, which are different from $\triangle$. These time points are shown as black dots in Fig. 2 and they may not match the sampling time points of FFT. If we assume $h(t)$ is 1 the convolution integral will be reduced to $y(t_k) = \int_0^{t_k} x(\tau)d\tau = \sum_{i=0}^{k-1}\int_{t_i}^{t_{i+1}} x(t_i)d\tau$. Take the integration $\int_{t_i}^{t_{i+1}} x(t_i)d\tau$ as an example and suppose Trapezoidal Rule is used to calculate the integration. The integral calculated by the uniform FFT sampling is $\int_{t_i}^{t_{i+1}} x(t)dt = \frac{1}{2}\triangle(x(t_i) + x(t_{i+1}))$, which is shown as the area under the dotted line inside the time interval (t_i, t_{i+1}) in Fig. 2. However, the actual value of the integration is the shaded area shown in the same figure. Thus the step size mismatch will cause numerical integration error, as shown in Fig. 2.

In order to solve this step size mismatch problem and utilize the analytical forms of the transfer functions, we propose a new average based numerical convolution method that is based on the average value and slope of $x(t)$ in order to fully utilize the $x(t)$ information inside a FFT sampling interval. we calculate three average values $\tilde{x}_i$, $\tilde{x}_{i1}$, and $\tilde{x}_{i2}$ for the interval $t \in (t_i, t_{i+1})$:

$$\tilde{x}_i = \frac{\int_{t_i}^{t_{i+1}} x(t)dt}{\triangle} \tag{18}$$

$$\tilde{x}_{i1} = \frac{\int_{t_i}^{t_{i+1}-\frac{\triangle}{2}} x(t)dt}{\triangle/2} \tag{19}$$

$$\tilde{x}_{i2} = \frac{\int_{t_i+\frac{\triangle}{2}}^{t_{i+1}} x(t)dt}{\triangle/2} \tag{20}$$

Let the slope $s_i = \frac{\tilde{x}_{i2}-\tilde{x}_{i1}}{\triangle/2}$, we use the piecewise linear function

$$\tilde{x}(t) = \tilde{x}_i + s_i[t-(t_i+\frac{\triangle}{2})], t \in [t_i, t_{i+1}), i = 0, ..., k-1 \tag{21}$$

to approximate $x(t)$.

Thus the convolution $y(t_k) = x(t_k) * h(t_k)$ can be re-written as:

$$y(t_k) = \int_0^{t_k} \tilde{x}(\tau)h(t_k-\tau)d\tau =$$

$$\sum_{i=0}^{k-1}\int_{t_i}^{t_{i+1}} (\tilde{x}_i + s_i[\tau-(t_i+\frac{\triangle}{2})])h(t_k-\tau)d\tau \tag{22}$$

Equation (22) can be evaluated by parts and algebraically manipulated to arrive:

$$y(t_k) = \sum_{i=0}^{k-1}\tilde{x}_i h_{a,k-i} + \sum_{i=0}^{k-1} s_i h_{b,k-i} \tag{23}$$

where

$$h_{a,k-i} = \int_{t_k-t_{i+1}}^{t_k-t_i} h(\tau)d\tau \tag{24}$$

$$h_{b,k-i} = -\frac{\triangle}{2}\int_{t_k-t_{i+1}}^{t_k-ti} h(\tau)d\tau + \int_{t_k-t_{i+1}}^{t_k-t_i}\int_{t_k-t_{i+1}}^{\tau} h(\tau')d\tau' d\tau \tag{25}$$

Equation (23) is evaluated by two FFTs in $O(NlogN)$ time, one FFT for the sum $\sum_{i=0}^{k-1}\tilde{x}_i h_{a,k-i}$ and the other for the sum $\sum_{i=0}^{k-1} s_i h_{b,k-i}$. Then the values of the convolution $y(t)$ at time points $t_0, t_1, ..., t_k$ are obtained by combining these two FFT results. Since our simulation time points may not match time points $t_0, t_1, ..., t_k$, the convolution values for those mismatched time points are calculated by interpolation.

Analytical expressions of $\int_0^t h_1(\tau)d\tau$, $\int_0^t\int_0^{\tau'} h_1(\tau')d\tau' d\tau$, and $\int_0^t h_3(\tau)d\tau$ are derived for the special case of G=0 in [5]. For the cases when analytical expressions are not available, these integrals are calculated numerically.

3.3 Large Slope Control

We observe that the transfer functions $h_1(t)$, $h_2(t)$, or $h_3(t)$ may have a very large slope during a small time interval. For instance, a transfer function $h_1(t)$ is shown in Fig. 3. This transfer

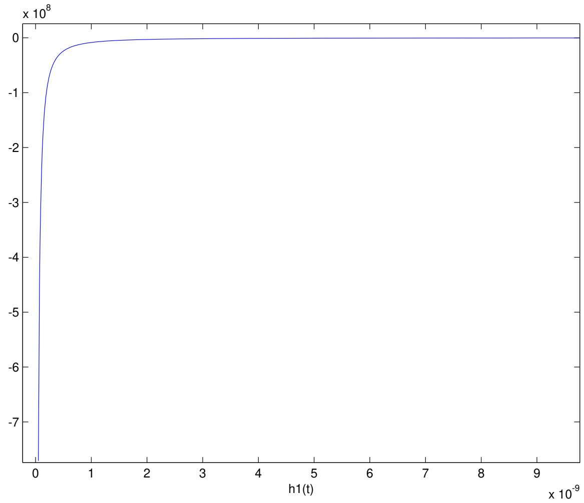

Figure 3: A Transfer function that shows a sharp slope region at the beginning of the simulation

function changes rapidly in the time interval (0,0.4ns). This will cause the corresponding FFT convolution calculated also have a sharp region. Interpolation inside the sharp region of the FFT convolution result will cause large errors. Fortunately, the sharper the transfer function, the smaller the time interval of the sharp region will be. Thus in order to efficiently and accurately catch these fast changing regions of the transfer functions we'll identify the sharp region of each transfer function by pre-processing the slope of the transfer functions and then identify the sharp region and divide each of the transfer functions into two parts:

$$h_s(t) = \begin{cases} h(t) & \text{if } t \in [t_{SharpStart}, t_{SharpEnd}] \\ 0 & \text{otherwise} \end{cases}$$

$$h_n(t) = \begin{cases} 0 & \text{if } t \in [t_{SharpStart}, t_{SharpEnd}] \\ h(t) & \text{otherwise} \end{cases}$$

Thus $h(t) = h_s(t) + h_n(t)$ and the convolution $y(t) = x(t) * h(t) = x(t) * h_s(t) + x(t) * h_n(t)$ Since $h_s(t)$ is the identified sharp slope region of the transfer function h(t), we use the generalized trapezoidal method [5] instead of our multilevel FFT convolution algorithm to calculate $x(t) * h_s(t)$ because interpolation error could be large for this region. For $x(t) * h_n(t)$ the proposed multilevel FFT convolution algorithm is used.

3.4 Algorithm Description

Now we are ready to present the whole picture of the proposed multilevel FFT convolution algorithm. Generally, we follow the recursive procedure described in section 3.1. For a given time point, several FFT convolutions need to be calculated according to Fig. 5. If the current simulation time point is t_k, FFT convolution is calculated using the numerical convolution formula presented in section 3.2 and stored if it hasn't been calculated yet. However, the convolution could be already calculated by a previous time point. In this case we just load the corresponding result for this time point t_k, interpolation is used if needed. The calculated convolution by the proposed multilevel FFT convolution method will be in the form of $Cx(t_k) + C'$, where C and C' are two constants calculated and $x(t_k)$ is the unknown. After all convolutions in equation

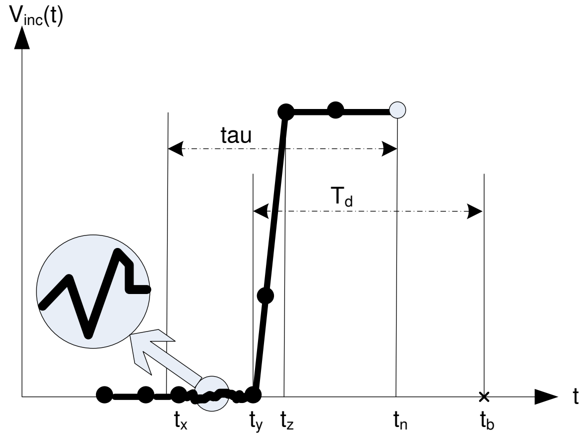

Figure 4: An example of breakpoint control scheme

10 and 11 are calculated, equation 10 and 11 can be written as:

$$c_1 v_2(t_k) - i_1(t_k) = c_2 v_1(t_k) + c_3 i_2(t_k) + c_4 \tag{26}$$

$$c_5 v_1(t_k) - i_2(t_k) = c_6 v_2(t_k) + c_7 i_1(t_k) + c_8 \tag{27}$$

We can write KCL equations for both the near end and the far end of the transmission line. This two KCL equations and equations (26) (27) are loaded into the circuit simulator and then $v_1(t_k)$, $v_2(t_k)$, $i_1(t_k)$, and $i_2(t_k)$ are solved by the simulator.

4. ADAPTIVE TIME STEP CONTROL

A new breakpoint control scheme is presented in this section to effectively capture waveform discontinuity during the simulation. Breakpoints are points in time where simulation is forced. Signals propagate on a transmission line with a delay T_d. If there is a discontinuity in the incident wave at the one end of the line, this discontinuity will propagate to the other end after the delay T_d. Same thing for the reflected wave. Thus it is important for the simulator to catch this discontinuity in order to ensure accuracy.

The breakpoint control scheme is given as an example shown in Fig. 4, where $v_{inc}(t)$ denotes a incident wave at the near end of the line and t_n is the current simulation timepoint. Solid dots represent previous timepoints. We need to detect sigular points in previous timepoints of this near end incident wave in order to capture sigularity at the far end. For example, if we found a sigular point at time point t_y as shown in the figure, we need to add a breakpoint at $t_y + T_d$. The problem is how to detect this sigularity.

Grivet-Talocia et. al. [11] proposed to use second derivative to check the discontinuity. However, if we have noise in the waveform, as shown in Fig.4 from time t_x to t_y, a lot of unnecessary breakpoints will be added because the noise is not ignored. Furthermore, singularity is checked during the time interval $(t_n - T_d, t_n)$ in [11], which would be too large for a long line.

We propose to use a two stage approach to check the discontinuity. Instead of checking the time interval $(t_n - T_d, t_n)$, we check the interval $(t_n - tau, t_n)$ where tau is no larger than T_d and number of time points inside the interval $(t_n - tau, t_n)$ should be less than a threshold, say 10. Once the time interval is determined, we first smooth the waveform using a moving average filter. All high frequency noise should be removed by the filter. Then we check the second derivative of the smoothed waveform. Once the first sigular point t_y is found, we add a breakpoint at $t_y + T_d$ and finish the breakpoint control. The next time point will be determined by the nearest breakpoint in the furture and Local Truncation Error calculated from other devices such as inductors and capacitors in the circuit.

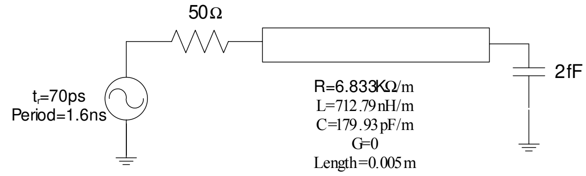

Figure 5: The test circuit, an on-chip transmission line

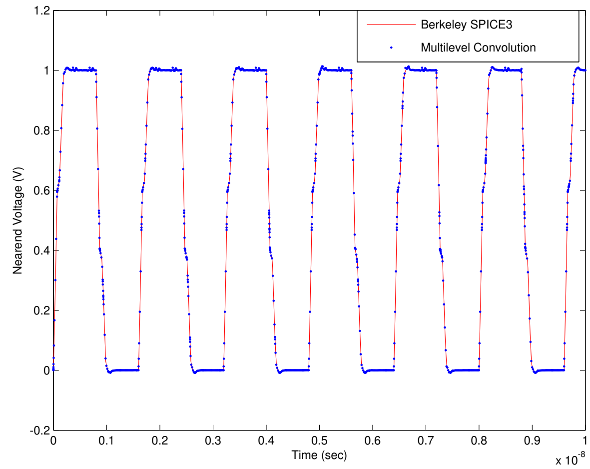

Figure 6: Voltage response at near end

5. EXPERIMENTAL RESULTS

The proposed approach is implemented in a modified version of Berkeley SPICE3f5. The experiments are executed on a Linux machine with 3.4GHz CPU and 2GB memory. For the test circuit shown in Fig. 5, Fig. 6 shows the voltage response at near end of transmission line, where our results exactly matches SPICE3 result. We ran the same circuit for 10ns, 10ns and 100ns. table 1 shows the CPU time and speedup. It can be seen from table 1 that the proposed method is much more efficient than SPICE3. The speedup increases from 2.5x to 126.6x when we increase the simulation length from 10ns to 200ns.

6. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, the proposed multilevel FFT convolution method reduces complexity of the conventional convolution approach for

Table 1: CPU Times For the Circuit Shown in Fig. 5

Experiment	SPICE3	Proposed Method	Speedup
Simulate 10ns	0.4s	0.16s	2.5
Simulate 100ns	34s	1.43s	23.7
Simulate 200ns	366s	2.89s	126.6

lossy transmission lines from $O(N^2)$ to $O(NLog^2N)$ by utilizing a multilevel FFT convolution method. Average based numerical convolution formula that exploits both the analytical forms of the lossy transmission line impulse responses and adaptive time steps are developed for the proposed multilevel FFT convolution method. Effective breakpoint control scheme is developed to adaptively select simulation timesteps. Experimental results demonstrate accurate waveform match with SPICE3f5 while the proposed approach is over 100 times faster than SPICE3f5. The proposed multilevel FFT convolution method can also be used for simulation of multiconductor Transmission Lines with frequency dependance per unit length parameters and transient simulation of lossy interconnects based on frequency domain S-Parameter data.

7. REFERENCES

[1] T. Quarles, A.R. Newton, D.O. Pederson, A. Sangiovanni-Vincentelli, "SPICE3 Version 3f3 User's Manual," Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, May, 1993

[2] I.-T. Chiang, and W. C. Chew, "Fast Real-Time Convolution Algorithm for Transients of Nonlinearly-Terminated Microwave Multiport Circuits," Microwave and Optical Technology Letters, Vol.39, No.4, Nov 2003.

[3] R. Achar and M. S. Nakhla, "Simulation of High-Speed Interconnects," Proceedings of the IEEE, 2001

[4] J. S. Roychowdhury, and D. O. Pederson, "Efficient Transient Simulation of Lossy Interconnect," IEEE/ACM Design Automation Conf., pp. 740-745, June 1991.

[5] J. S. Roychowdhury, A. R. Newton, and D. O. Pederson, "Algorithms for the Transient Simulation of Lossy Interconnect," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, VOL. 13. No.1, January 1994

[6] A. R. Djordjevic, T. K. Sarkar and R. F. Harrington, "Time domain response of multiconductor transmission lines," Proc. of the IEEE, vol. 75, pp.743-764, June 1987.

[7] R. Griffith and M. Nakhla, "Time domain analysis of lossy coupled trasmission lines," IEEE Trans. Microwave Theory Tech., vol. MTT-38, pp. 1480-1487, Oct. 1990

[8] F.H. Branin, Jr., "Transient analysis of loseless transmission line," Proc. IEEE, vol. 55, pp. 2012-2013, Nov. 1967

[9] F.Y. Chang, "Waveform relaxation analysis of RLCG transmission lines," IEEE Trans. Circuit Syst., vol. 37, pp.1394-1415, Nov. 1990.

[10] F.Y. Chang, "Waveform relaxation analysis of nonuniform lossy transmission lines characterized with frequency-dependent parameters," IEEE Trans. circuit Syst., vol. 38, pp. 1484-1500, Dec. 1991

[11] S. Grivet-Talocia, H.-M. Huang, A.E. Ruehli, F. Canavero and I.M. Elfadel, "Transient Analysis of Lossy Transmission Lines: An Efficient Approach Based on the Method of Characteristics," IEEE Trans. Advanced Packaging, vol. 27, No. 1, Feb 2004

[12] E.C. Chang and S.-M. Kang, "Transient simulation of lossy coupled transmission lines using iterative linear least square fitting and piecewise recursive convolution", IEEE Trans. circuit and systems I, vol. 43, no. 11, Nov 1996

[13] J.E. Bracken, V. Raghavan and R.A. Rohrer, "Interconnect simulation with Asymptotic Waveform Evaluation," IEEE Trans. Circuits and Systems, vol. 39, pp. 869-878, Nov 1992.

[14] S. Lin and E. S. Kuh, "Transient Simulation of Lossy Interconnects Based on the Recursive Convolution Formulation," IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications,Vol. 39, No. 11, Nov 1992.

[15] C. Gordon, t. Blazeck, and R. Mittra, "Time-Domain Simulation of multiconductor transmission lines with frequency dependent losses," IEEE Trans. Computer-Aided Design, vol. 11, pp. 1372-1387, Nov. 1992

[16] M. Celik and A.c. Cangellaris, "Simulation of dispersive multiconductor transmission lines by Pade approximation via the Lanczos process," IEEE Trans. Microwave Theory Tech., vol. 44, pp. 2525-2535, Dec, 1996

[17] D.B. Kuznetsov and J.E. Schutt-Aine, "Optimal transient simulation of transmission lines," IEEE Trans. Circuits Syst., vol. 43, pp. 110-121, Feb 1996

[18] T.V. Nguyen, "Efficient simulation of lossy and dispersive transmission lines," 31st IEEE Design Automation Conference, 1994.

[19] M. Celik and L.T. Pileggi, "Simulation of lossy multiconductor transmission lines using backward euler integration," IEEE Tran. Circuits Syst., vol. 45, No. 3 Mar 1998

[20] Q. Yu, J.M. Wang and E.S. Kuh, "Passive Multipoint Moment Matching Model Order Reduction Algorithm on Multiport Distributed Interconnect Networks," IEEE Trans. Circuits Syst., vol. 46, No. 1, Jan 1999

[21] A.C. Cangellaris, S. Pasha, J.L. Prince and M. Celik, "A New Discrete Transmission Line Model for Passive Model Order Reduction and Macromodeling of High-Speed Interconnections," IEEE Trans. Advanced Packaging, vol. 22, No. 3, Aug 1999

[22] S. Ramo, J.R. Whinnery, and T. Van Duzer, Fields and Waves in Communication Electronics. New York: Wiley, 1965

[23] K.S. Oh, "Accurate Transient Simulation of Transmission Lines with the Skin Effect," IEEE Trans. Computer-Aided Desing of Integrated Circuits and Systems, vol. 19 No. 3, Mar. 2000