Clock Skew Analysis via Vector Fitting in Frequency Domain

Ling Zhang[1], Wenjian Yu[1,2], Haikun Zhu[3], Wanping Zhang[3], Chung-Kuan Cheng[1]

[1]University of California, San Diego, CA 92093
[2]Tsinghua University, Beijing, China, [3] Qualcomm Inc., CA 92121
[1,2,3]{lizhang, ckcheng, hazhu, w7zhang}@cs.ucsd.edu, [2]yu-wj@tsinghua.edu.cn

Abstract

An efficient frequency-based clock analysis method: CSAV is proposed in this paper. It computes the circuit response by first solving the state equation in frequency domain, and derive the rational approximate with the help of vector fitting [9]. There are two aspects that contribute to the time efficiency of the method. One is CSAV solves the state equation only on selected frequency points, which significantly reduce the amount of time for equation solving. The other is CSAV performs vector fitting and waveform recovery only on user specified nodes, which save the unnecessary computation on the nodes which are not concerned by user. The complexity of our method is $O(\lceil \lg f_{max} \rceil N^{\alpha} + \lceil \lg f_{max} \rceil^2 N_a N_{out})$, where f_{max} is proportional to the knee frequency of input signal, N is the node number of the circuit, α is a constant around 1.3, N_a is the order of approximation and N_{out} is the number of output nodes. Our experimental results show that compared with Hspice, CSAV achieves speed-up up to 35 times while the error is only 1%. Moreover, computational saving of CSAV grows with circuit size, which makes this method especially promising for large cases.

1. Introduction

Clock signal is used as a time reference for all on-chip data switching, and is the key for the success of synchronous digital system [8]. As integrated circuit technology scales into deep submicrometer and the clock frequency rapidly increases into Giga-Hertz range, small amount of clock skews on critical paths have ever greater impact on the overall clock network performance. Designers need to know the skew information in early design phase, efficient analysis tools are needed for this purpose.

Hspice simulation is widely used for circuit analysis because of its high accuracy. For large circuits such as clock network and power/ground grid, simulation can be very time consuming, since the time complexity of SPICE is $O(N^p)$, where N is the number of node in the circuit and p is between 1 and 2. The large number of node will significantly slow down the simulation. In [4], Chen et al. proposed a sliding window scheme to speed up the simulation process for clock with mesh structure. The method has little accuracy loss but it is only applicable to clock mesh.

To serve as a fast estimation method of interconnect delay, interconnect models are developed in many previous works [12][10][3]. [3] shows an effective modeling strategy for the derivation of closed form time domain solution of RLC interconnects by adopting the Fourier Series expansion. [12]and [10] formulate clock trees as RLC interconnect trees and use second-order models to analyze the trees. This model based methods are very efficient, however the accuracy is limited.

Model Order Reduction (MOR) techniques can be used to obtain more accurate results [14][13][7]. These methods reduce the dimension of the system matrix so as to significantly fasten the matrix solving process.

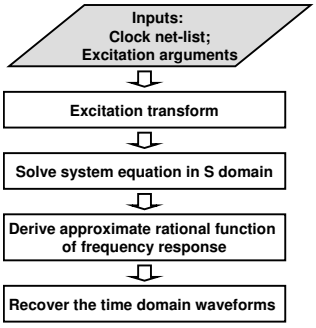

Figure 1. Flow chart of CSAV

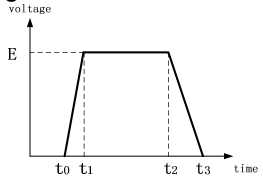

Figure 2. Ramp signal waveform

In this paper, we propose an efficient analysis method via vector fitting approach: CSAV(Clock Skew analysis via Vector Fitting). The complexity of our method is $O(\lceil \lg f_{max} \rceil N^{\alpha} + \lceil \lg f_{max} \rceil^2 N_a N_{out})$, in which f_{max} is proportional to the knee frequency of input signal, N is the node number of the circuit, α is a constant around 1.3, N_a is the order of approximation and N_{out} is the number of output nodes.

Our method offers a different approach from MOR technique to derive an accuracy-guaranteed approximation. We choose the sampling frequency points in log scale, and we fit the frequency response of the system, instead of the system transfer function.

Unlike the model based methods, our method does not make topological constraint upon the network, and it is more accurate. The error is within 1% compared with results from Hspice simulation. CSAV is also efficient compared to Hspice simulation. For a given circuit, the CPU time of our method linearly depends on the number of frequency sampling points, and the number of circuit nodes that users are concerned with.

2. Description of CSAV

Fig. 1 describes the flow of CSAV. The inputs to CSAV includes circuit matrices and excitation signal arguments. Without much loss of generality, it is assumed that the circuit is excited by ramp signals, as shown in Fig. 2. In CSAV, the Laplace transform of excitation is derived analytically, and frequency domain outputs on selected points are solved. Then we use vector fitting to have the con-

0-7695-3117-2/08 $25.00 © 2008 IEEE
DOI 10.1109/ISQED.2008.67

tinuous frequency response and recover the time domain response to have the waveforms. In this section, the flow of CSAV is explained in details. We also present the theoretical reason of why the flow is particularly effective for clock network and therefore huge CPU time can be saved without loss of accuracy.

2.1. Laplace transform of excitation

Under the ramp input assumption, the signal waveform presented in Fig. 2 can be written as Equation 1:

$$v_{in}(t) = \frac{E}{t_1-t_0}(t-t_0)u(t-t_0) - \frac{E}{t_1-t_0}(t-t_1)u(t-t_1) - \frac{E}{t_3-t_2}(t-t_2)u(t-t_2) + \frac{E}{t_3-t_2}(t-t_3)u(t-t_2) \quad (1)$$

in which $u(t)$ is a step function. By the definition of Laplace transform, the ramp signal in s domain has the following format:

$$V_{in}(s) = \frac{E}{(t_1-t_0)s^2}e^{-st_0} - \frac{E}{(t_1-t_0)s^2}e^{-st_1} - \frac{E}{(t_3-t_2)s^2}e^{-st_2} + \frac{E}{(t_3-t_2)s^2}e^{-st_3} \quad (2)$$

which is a simple function of E, t_0, t_1, t_2, t_3 and s. Note that strictly speaking, the clock source input signal is output from buffers, which has similar shape with the piecewise linear function but more smooth. Its Laplace transform can still be derived by numerical method without having the above simple formula.

2.2. Obtain the output frequency response on selected points

2.2.1 The system equation

In general, the system behavior is described by the system equation. For clock network which contains *RLC* components, it is reasonable to assume that all nodes have capacitors connected to ground, and the system equation can be written as [5]:

$$\boldsymbol{M\frac{dx(t)}{dt} = -Gx(t) + Pv_{in}(t)} \quad (3)$$

where

$$\boldsymbol{x(t) = \begin{bmatrix} V_c(t) \\ I_L(t) \end{bmatrix}, M = \begin{bmatrix} C & 0 \\ 0 & L \end{bmatrix}, G = \begin{bmatrix} Y & E \\ -E^T & R \end{bmatrix}} \quad (4)$$

V_c denotes the vector of voltages across the capacitors and I_L denotes a vector of currents through the inductors, and C, L are the capacitance and inductance matrices. The matrices Y and R are the admittance matrix and impedance matrix. $v_{in}(t)$ is the input voltage vector, and P is the incidence matrix.

We can use Laplace transform to rewrite Equation 3 in s domain:

$$\boldsymbol{sMx(s) - Mx_0 = -Gx(s) + PV_{in}(s)} \quad (5)$$

in which x_0 denotes the initial condition of the time domain state vector $x(t)$ at time $t = 0$. For clock network, we can safely assume that $x_0 = 0$. After simple matrix multiplication and inversion, the state vector in s domain can be expressed as:

$$\boldsymbol{x(s) = (sM + G)^{-1}PV_{in}(s)} \quad (6)$$

Matrices G, M, P are obtained from user inputs, excitation signal in s domain $V_{in}(s)$ is derived by the step in Section 2.1, therefore Equation 6 can be solved for given complex frequency. Usually solving Equation 6 has high computational expense due to the large circuit size and matrix inversion operation.

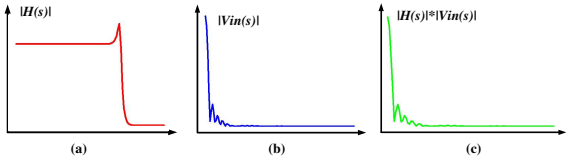

Figure 3. Frequency response of the network transfer function, the input signal and their product

2.2.2 Sampling frequencies selection

Since it is computationally expensive to solve the system equation, we want to minimize the number of matrix operation while obtain the output frequency response accurately. In this subsection we discuss how should we select the sampling frequencies.

The clock signal, simplified as piecewise linear function, has the frequency spectrum depicted in Fig. 3(b), which has large magnitude in low frequency range and attenuate to zero at high frequency range. Usually knee frequency f_{knee} is used to characterize the frequency distribution: $f_{knee} = \frac{0.5}{t_r}$ where t_r refers to the rise/fall time of the signal. 90% of the signal energy is contributed by the frequency components that are lower than f_{knee} [11].

With the purpose of propagating the clock signal, clock network with *RLC* elements can be considered as low-pass filter (Fig. 3(a)), which permits the low frequency (the main energy part of clock signal) to pass but block the high frequency part. Cut-off frequency is introduced to describe the frequency characteristics of a filter: for low-pass filter, cut-off frequency is the frequency where the magnitude of transfer function drops -3dB from unity value. For clock network, the cut-off frequency is a function of the value of R, L and C.

For a clock network and a given input signal, when the network cut-off frequency is higher than the knee frequency of input signal, the network can propagate the input without much change on the shape of waveform. Otherwise, the input is distorted or even can not be used as clock signal. For both of the cases, the output frequency response, that is the product of the network transfer function and input frequency response, has the characteristic shown in Fig 3(c). To capture the output frequency spectrum, only the low frequency components need to be considered and calculated. The frequencies that are far beyond the knee frequency can be ignored.

A very straightforward way of choosing sampling frequencies is to first define a frequency upper-bound f_{max}, and then uniformly select frequency points in logarithmic scale from zero to f_{max}. By doing so, the information in lower frequency components can be captured. The value of f_{max} is determined by the knee frequency and is set to $10f_{knee}$ in our experiments. The number of sampling points N_s, which can be expressed as $\lceil \lg(f_{max}) \rceil$ also affects the computation accuracy, and computation time as well.

2.3. Derive the continuous output frequency response via vector fitting

From the discussion in Section 2.2, we can obtain the frequency responses of the clock network outputs on a set of discrete sampling points $s_1, s_2, ..., s_{N_s}$. It is hard to restore the accurate time domain waveforms from this information, therefore we need to derive the approximated rational function in s domain, which is done by vector fitting [9].

Vector Fitting is a robust numerical method for rational approximation in the frequency domain, and has been used in P/G ground network analysis [18]. It is a pole relocation technique in which the poles are improved by iteratively solving a linear problem until convergence is achieved.

Inputs to Vector fitting include a set of values of $f(s)$ and a group of initial poles $\overline{a_i}, i = 1, 2, ..., N_a$. Vector fitting approximates $f(s)$

with the following rational function:

$$f(s) = \sum_{i=1}^{N_a} \frac{c_i}{s - a_i} + d + sh \tag{7}$$

in which d and h can be set to zero by user specification. Regarding to the value of initial poles, since they affect the performance of the fitting algorithm significantly, the authors recommended that the poles are chosen in conjugate pairs and the real part is 0.01 of the imaginary part:

$$\overline{a_i} = -\alpha + j\beta,\ \overline{a_{i+1}} = -\alpha - j\beta \tag{8}$$

where

$$\alpha = \beta/100 \tag{9}$$

N_a is the order of approximation for vector fitting. Larger N_a gives higher accuracy and require longer computation time.

Vector fitting returns the fitted poles a_i and residues c_i, and d, h as well. The root-mean square error of fitting result is also reported.

In our application, each output node of the clock network has a frequency response $V_{out}(s)$ with respect to input signals. Hence we derive the poles and residues of each of the frequency response by directly adopting vector fitting based on the discrete frequency responses obtained in Section 2.2. The parameter d and h are set to be zero here, since the response of a linear circuit system should not contain impulse function. Therefore the fitting results of vector fitting have the following format:

$$V_{out}(s) = \sum_{i=1}^{N_a} \frac{c_i}{s - a_i} \tag{10}$$

2.4. Time domain waveform recovery

Having the rational approximation results with format of Equation 10, we can readily employ inverse Laplace transform to recover the output expression in time domain:

$$v_{out}(t) = \sum_{i=1}^{N_a} c_i e^{a_i t}, t > 0 \tag{11}$$

To plot the corresponding waveform, we need to evaluate the above function at each time step. Assume the time step is Δt, and the number of time step is N_t. N_t is usually not a small number (10^2, 10^3 or larger), and each evaluation of Equation 11 involves a number of exponential computations, which is quite time consuming.

A better way of generating time domain waveform is as follows. First we define:

$$E_i = e^{a_i \Delta t} \tag{12}$$

Then Equation 11 can be rewritten as:

$$v_{out}(\Delta t) = \sum_{i=1}^{N_a} c_i E_i \tag{13}$$

which is the first point in the time domain waveform. The subsequent points can be derived as:

$$v_{out}(2\Delta t) = \sum_{i=1}^{N_a} c_i E_i^2 \tag{14}$$

$$... \tag{15}$$

$$v_{out}(N_t \Delta t) = \sum_{i=1}^{N_a} c_i E_i^{N_t} \tag{16}$$

By doing so, the number of exponential computation can be reduced from $N_a N_t$ to N_a, while the number of multiplication computation increases from $N_a N_t$ to $N_a N_t + N_a(N_t - 1)$.

Table 1. Test cases logistics

Tree test case	s1423	s5378	s15850	r4	r5
#node	146	356	1192	3804	6200
Mesh test case	m80	m100	m120	m150	m200
#node	6400	10000	14400	22500	40000

3. Complexity analysis

We briefly discuss the complexity of each step in the flow of CSAV.

(1) Having the analytical form expression, the complexity of the Laplace transform of excitations is $O(N_{in}\lceil \lg(f_{max})\rceil)$, in which N_{in} is the number of inputs.

(2) The complexity of matrix solving in S domain is $O(N^{\alpha}\lceil \lg(f_{max})\rceil)$, in which N is the number of nodes in the system, α is found to be around 1.3 by experiments. N^{α} is the complexity for one matrix solving.

(3) In vector fitting, the most expensive computation is Singular Value Decomposition (SVD). For a dense $p \times q$matrix, the complexity of SVD computation is $O(pq^2+p^2q+q^3)$ [2]. In our application, $p = \lceil \lg(f_{max})\rceil, q = N_a$. Usually $\lceil \lg(f_{max})\rceil$ is larger than N_a, therefore the complexity of vector fitting for one rational function is $O(\lceil \lg(f_{max})\rceil^2 N_a)$. When the output number is N_{out}, the complexity becomes $O(\lceil \lg(f_{max})\rceil^2 N_a N_{out})$.

(4) The complexity of doing time domain waveform recovery, as discussed in 2.4, is $O(N_t N_a N_{out})$.

Our experimental results show that step (2) and (3) dominate the total complexity of our method. Thus, the complexity of our method can be written as $O(\lceil \lg f_{max} \rceil N^{\alpha} + \lceil \lg f_{max} \rceil^2 N_a N_{out})$

4. Implementation and experimental results

We use PETSc [1], which is written in C to solve the s domain system equations, and use vector fitting, which is written in Matlab to fit the s domain outputs. Time domain waveform recovery is also implemented in Matlab.

We compare the analysis results of CSAV against Hspice on both clock tree test cases and mesh cases, and Table 1 shows the number of nodes in each circuit. Bounded Skew Clock Tree Routing (BST) algorithm proposed by [6] is used to generate clock trees for ISCAS89 benchmarks [15], and two IBM cases r4 and r5, which have larger scale. The mesh cases are created to have regular structure. From the discussion in Section 2, we can see that the number of sampling frequency points N_s and the approximation order N_a influence the accuracy of CSAV. Fig 4 presents the effects of these two parameters. When N_a is too small, the approximated system has too few orders to capture the high frequency effect. When N_s is too small, the waveform is not very smooth and has small up and down at the very beginning.

N_s and N_a also influence computation speed of CSAV. N_s corresponds to the number of equation solving, and N_a determines fitting complexity. We minimize these two parameters subjected to a certain accuracy requirement. In our experiments, we permit 1% error of output arrival times compared to Hspice.

The CPU time of CSAV have three parts: T_p, T_v and T_w, as explained in Table 2. T_v and T_w linearly depend on the number of interested output nodes N_{out}, which is determined by user input. In Table 2, the total Hspice CPU time for each case is shown in the second column. The values of T_v and T_w are listed in the third and forth column. For example, for the circuit s1423, the total CPU time of Hspice simulation is 0.78 seconds, and the CPU time of CSAV is $0.16 + 0.029N_{out}$. Usually in clock design, skews on critical paths are concerned, and the number of critical paths is limited. To reasonably quantify the efficiency of CSAV, we assume

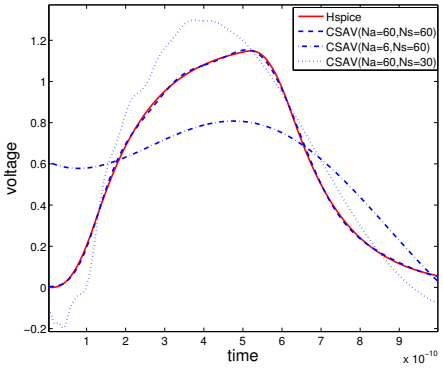

Figure 4. Recovered time domain waveforms

Table 2. CPU time comparison of CSAV and Hspice (unit: second)

Test case	Hspice CPU time	CSAV		Speed-up for $\frac{N_{out}}{N_{total}} = 5\%$[3]
		T_p[1]	$(T_v + T_w)$[2]	
s1423	0.78	0.16	0.029	2.15
s5378	2.08	0.34	0.037	2.07
s15850	22.45	2.39	0.038	4.81
r4	277.15	18.49	0.041	10.55
r5	668.71	46.78	0.067	9.90
m80	338.14	7.55	0.053	13.80
m100	681.17	13.27	0.038	21.10
m120	1342.06	23.67	0.037	26.68
m150	2965.53	43.03	0.037	35.03
m200	9352.85	132.75	0.069	34.54

[1] T_p is the time for PETSc to solve state equations in s domain.
[2] T_v is the time for the vector fitting, and T_w is the time for waveform recovery in the time domain.
[3] The speed up is calculated based on the assumption that the user defined output nodes are 5% of the total node number.

that 5% of nodes in the clock network belong to critical paths and their waveform need to be observed recovered, and the corresponding speed-up is presented in the fifth column of Table 2.

From Table 2 we notice that CSAV has higher speed-up for circuits with larger scale, and the maximum speed-up is around 35 times.

In [13], the run time of Hspice and PRIMA was compared for a R-L mesh ground plane case. Hspice needs over 17,000 seconds while PRIMA only needs no more than 2 seconds. In [17], the author proposed a general passive multi-point moment matching model order reduction algorithm, and compared its performance with SPICE on a clock net. The results show that about 22 times of speed up can be achieved. In [16], a more recent work on model order reduction, the author compared their approach(BVOR) of building circuit matrix to the original MNA (Modified Nodal Analaysis) method, and over 3300 times of speed up was reported.

Since the improvements on CPU time are case dependent, the same cases should be used to compare the speed of the above model order reduction based methods and CSAV. Once we obtain the test cases, we will conduct detailed comparisons and report the results.

5. Conclusions and future work

In this paper, an efficient clock network analysis method, CSAV, is proposed. CSAV computes the circuit response by first solving the state equation in frequency domain, and derive the rational approximate with the help of vector fitting. There are two aspects that contribute to the time efficiency of the method. One is CSAV solves the state equation only on selected frequency points, which significantly reduce the amount of time for equation solving. The other is CSAV performs vector fitting and waveform recovery only on user specified nodes, which save the unnecessary computation on the nodes which are not concerned by user.

Our experimental results show that compared with Hspice, CSAV achieves speed-up up to 35 times while the error is only 1%. Moreover, computational saving of CSAV grows with circuit size, which makes this method especially promising for large cases.

Current implementation of CSAV assumes a ramp input signal, which is not necessarily true in reality. For arbitrary input signal, numerical method can be used to obtain the Laplace transform of excitation, which combined with accurate clock buffer model enables CSAV deal with clock network containing buffers.

6. Acknowledgment

The authors would like to acknowledge the support of NSF CCF-0618163 and California MICRO Program. The work of W. Yu is supported by the the Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList).

References

[1] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc Web page, 2001. http://www.mcs.anl.gov/petsc.

[2] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *European Conference on Computer Vision*, 2002.

[3] G. Chen and E. G. Friedman. An RLC interconnect model based on fourier analysis. *IEEE Trans. Computer-Aided Design of Intergrated Circuits and Systems*, 2005.

[4] H. Chen, C. Yeh, G. Wilke, S. Reddy, H. Nguyen, W. Walker, and R. Murgai. A sliding window scheme for accurate clock mesh analysis. In *Proc. International Conference of CAD*, 1995.

[5] C. K. Cheng, J. Lillis, S. Lin, and N. Chang. *Interconnect Analysis and Synthesis*. Wiley-Interscience Publication, 2000.

[6] J. Cong, A. B. Kahng, C.-K. Koh, and C. W. A. Tsao. Bounded-skew clock and steiner routing. *ACM Trans. on Design Automation of Electronic Systems*, 1998.

[7] P. Feldmann and R. W. Freund. Efficient linear circuit analysis by pade approximation via the lanczos process. *IEEE Trans. on CAD*, 1995.

[8] E. G. Friedman. Clock distribution networks in synchronous digital integrated circuits. *Proc. of IEEE*, 2001.

[9] B. Gustavsen and A. Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Trans. Power Delivery*, 1999.

[10] Y. I. Ismail, E. G. Friedman, and J. L. Neves. Equivalent elmore delay for rlc trees. *IEEE Trans. Computer-Aided Design of Intergrated Circuits and Systems*, 2000.

[11] H. W. Johnson and M. Graham. *High-Speed Digital Design, A Handbook of Black Magic*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[12] A. B. Kahng and S. Muddu. An analytical delay model for rlc interconnects. *IEEE Trans. Computer-Aided Design of Intergrated Circuits and Systems*, 1997.

[13] A. Odabasioglu, M. Celik, and L. T. Pileggi. Prima: Passive reduced-order interconnect macromodeling algorithm. *IEEE Trans. Computer-Aided Design of Intergrated Circuits and Systems*, 1998.

[14] L. T. Pillage and R. A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Trans. Computer-Aided Design of Intergrated Circuits and Systems*, 1990.

[15] J. G. Xi and W. M. Dai. Useful-skew clock routing with gate sizing for low power design. 1996.

[16] H. Yu, Y. Shi, L. He, and D. Smart. A fast block structure preserving model order reduction for inverse inductance circuits. In *International Conference on Computer Aided Design*, 2006.

[17] Q. Yu, J. M. L. Wang, and E. S. Kuh. Passive multipoint moment matching model order reduction algorithm on multiport distributed interconnect networks. *IEEE Trans. on Circuits and Systems–I: Fundamental Theory and Applications*, 1999.

[18] W. Zhang, L. Zhang, R. Shi, H. Peng, Z. Zhu, L. Chua-Eoan, R. Murgai, T. Shibuya, N. Ito, and C. Cheng. Fast power network analysis with multiple clock domains. In *International Conference on Computer Design*, 2007.