An Efficient Method for Terminal Reduction of Interconnect Circuits Considering Delay Variations

Pu Liu*, Sheldon X.-D. Tan*, Hang Li*, Zhenyu Qi*, Jun Kong†, Bruce McGaughy†, Lei He‡
* Department of Electrical Engineering, University of California, Riverside, CA 92521
† Cadence Design Systems Inc., San Jose, CA 95134
‡ Department of Electrical Engineering, University of California, Los Angeles, CA 90095

***Abstract*— This paper proposes a novel method to efficiently reduce the terminal number of general linear interconnect circuits with a large number of input and/or output terminals considering delay variations. Our new algorithm is motivated by the fact that VLSI interconnect circuits have many similar terminals in terms of their timing and delay metrics due to their closeness in structure or due to mathematic approximation using meshing in finite difference or finite element scheme during the extraction process. By allowing some delay tolerance or variations, we can reduce many similar terminals and keep a small number of representative terminals. After terminal reduction, traditional model order reduction methods can achieve more compact models and improve simulation efficiency. The new method, TERMMERG, is based on the moments of the circuits as the metrics for the timing or delay. It then employs singular value decomposition (SVD) method to determine the optimum number of clusters based on the low-rank approximation. After this, the K-means clustering algorithm is used to cluster the moments of the terminals into different clusters. Experimental results on a number of real industry interconnect circuits demonstrate the effectiveness of the proposed method.**

I. INTRODUCTION

Complexity reduction and compact modeling of interconnect networks have been an intensive research area in the past decade due to increasing signal integrity effects and rising electro and magnetic couplings modeled by parasitic capacitors and inductors. Most of previous research works mainly focus on the reduction of the internal circuitry by various reduction techniques. The most popular one is based on subspace projection [11], [2], [12], [8], [5]. Projection-based method was pioneered by Asymptotic Waveform Evaluation (AWE) algorithm [11] where explicit moment matching was used to compute dominant poles at low frequency. Later more numerical stable techniques are proposed [2], [12], [8], [5] by using implicit moment matching and congruence transformation.

However, nearly all existing model order reduction techniques are restricted to suppress the internal nodes of a circuit. Terminal reduction, however, is less investigated for compact modeling of interconnect circuits. An important observation is that many interface terminals of interconnect networks modeled as RLCM circuits are functionally similar or equivalent. The reason is that those terminals are close to each other structurally or they are extracted by using mathematic approximation by volume or surface meshing in methods like finite difference and finite element scheme. As a result, their electrical characteristics are also similar in a well-designed VLSI system. For instance clock sinks in clock networks, substrate plane, critical interconnects in the memory circuits like word or bit lines are among those interconnects. Recent studies [9], [10], [4], [3] show that there exists a large degree of correlation between the various input and output terminals. Therefore a low-rank approximation was performed on the input and output mapping matrices before the model order reduction process. However, the low-rank approximation in the existing methods are only based on the DC or a specific order of moments of responses. So the port-reduced systems may not correlate well with the original systems in terms of timing or delay.

In this paper we present a novel terminal reduction method called TERMMERG. The new terminal reduction method is based on the observation that if we allow some delay tolerance or variations, which actually can't be avoided in today's VLSI chip manufacture and working environments, some of the terminals with similar timing responses actually can be suppressed or merged into one representative terminal during the reduction without affecting their modeling functionality. In contrast to the existing terminal reduction methods, the new approach use high order moments as timing and delay metrics. Specifically, given some delay tolerances or variations, TERMMERG employs singular value decomposition (SVD) method to determine the number of clusters based on the low-rank approximation. Then the K-means clustering algorithm is used to cluster the moments of the terminals into different clusters. After the clustering, we pick one terminal that could best represent other terminals for each cluster.

The paper is organized as follows. In Section II, we first briefly reviews the concept of moments and then we present how the input and output moment matrices are constructed for terminal reduction. Section III presents a SVD-based method to obtain the optimal number of clusters for the proposed terminal merging algorithm. In section IV, we describe the K-means based clustering algorithm to find the representative terminals for each cluster. In section V, we present the whole terminal reduction flow of TERMMERG and then discuss some practical issues associated with the implementation. The experimental results and conclusions are presented in section VI and section VII, respectively.

II. INPUT AND OUTPUT MOMENT MATRICES

Our task is to find the terminals with similar delay or timing behaviors such that they can be viewed as one terminal if some

This work is funded by NSF CAREER Award CCF-0448534, UC Micro #04-088 via Cadence Design System Inc.

delay variations are allowed. In this paper, we focus on the timing metric of terminals and look at their timing responses due to step or impulse inputs. But our method can be applied to other metrics of interest.

Ideally, the delay or timing information should be represented by waveforms in the time domain . But this does not give us the best representation for our terminal merging method. Because all the waveforms have to be computed first, it is also difficult to compare two waveforms in general. Instead, we use terminal response moments in frequency domain to represent their time-domain response information.

Our algorithm is based on the observation that if two terminals have similar timing or delay, then they should have the similar moments (vectors) numerically. Remember that the moments computed in Eq.(4) represent the impulse responses of outputs due to inputs. It is well known that the 1^{th} moment m_1 represents the first-order delay approximation, or the Elmore delay, of the corresponding output with respect to a specific input. Higher order moments represent more detailed timing and delay information. Thus moment vector is an ideal expression of timing information for our terminal merging algorithm.

Since we need to merge both input and output terminals, we need to present the timing information for both input and output terminal. As a result, we have two moment matrices, the input moment matrix M_I and output moment matrix M_O.

For a single-input and single-output (SISO) system, the system's transfer function $H(s)$ could be expanded into a Taylor series around $s = 0$. The coefficient of s^i in the series expansion is the i^{th} moment of the transfer function:

$$H(s) \quad = m_0 + m_1 s + m_2 s^2 + m_3 s^3 + \dots \tag{1}$$

For a general linear (linearized) time-invariant network with p input and q output terminals, we can describe it as follows.

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} \\ \mathbf{y} &= C\mathbf{x} \end{aligned} \tag{2}$$

Where $\mathbf{x}$ is n-dimensional state vector, $\mathbf{u}$ is p-dimensional input vector, and $\mathbf{y}$ is q-dimensional output vector. The transfer function by Laplace transformation is

$$H(s) = C(sI - A)^{-1}B \tag{3}$$

If we expand the above equation in a Taylor's series at $s = 0$, we get the moments at various terminals:

$$\begin{aligned} \mathbf{m}_0 &= -CA^{-1}B \\ \mathbf{m}_1 &= -CA^{-2}B \\ &\vdots \\ \mathbf{m}_{r-1} &= -CA^{-r}B \\ &\vdots \end{aligned} \tag{4}$$

Each moment $\mathbf{m}_i$ is a $q \times p$ matrix, where each column j in $\mathbf{m}_i$ represents the moment vector of all output terminals due to the input terminal j and each row k in $\mathbf{m}_i$ represents the moment vector at the output terminal k due to all input terminals. Then a moment matrix can be written as

$$M = \begin{bmatrix} \mathbf{m}_0 & \mathbf{m}_1 & \dots & \mathbf{m}_{r-1} & \dots \end{bmatrix} \tag{5}$$

In order to perform terminal reduction for both inputs and outputs, different moment matrices are constructed. For the output terminal reduction, we define the *output moment matrix* M_O as:

$$M_O = \begin{bmatrix} \mathbf{m}_0^T \\ \mathbf{m}_1^T \\ \vdots \\ \mathbf{m}_{r-1}^T \end{bmatrix} \tag{6}$$

where each column j represents a moment series of output node j due to all input's stimuli. Notice that in this way, the output terminal's responses are with respect to all the inputs to make sure they are similar under all the inputs.

Similarly, for input terminal reduction, the *input moment matrix* M_I is defined as:

$$M_I = \begin{bmatrix} \mathbf{m}_0 \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_{r-1} \end{bmatrix} \tag{7}$$

where each column k represents a moment series at all output's nodes due to an input node k.

To determine the order of moments r, we use the following rule: the number of moments from all the inputs should be equal or larger than the number of terminals to be merged. The reason is that in the worst case where no terminals can be merged, we should be able to distinguish all the terminals using the moment information. This will become more clear in the following section. As a result, we have

$$rp \geq q \text{ for } M_O \tag{8}$$

$$p \leq rq \text{ for } M_I \tag{9}$$

Moments can be efficiently computed by recursively solving the given circuits using traditional SPICE-like simulation technique [11]. After we obtain the moment matrices as shown in Eq.(6) or Eq.(7), we proceed to find the optimum number of clusters using singular value decomposition method to be shown in the next section.

III. Determination of Cluster Number By SVD

In this section, we present how to find the best number of independent clusters by using singular value decomposition on the input and output moment matrices discussed in the previous section.

If two terminals have similar timing responses, it means that their moments have very similar values. If we have a number of terminals with similar timing behaviors, their moment matrix, where each moment series is a column or a row, will be a low-rank matrix. Singular value decomposition is very efficient to deal with rank-deficient matrices and it can reveal a great deal about the ranks and structure of a matrix, which motivate us to find the optimal number of clusters based on the moment matrices.

For a $m \times n$ matrix A, the SVD decomposition of A is

$$A = U_{m\times m}\Sigma V_{n\times n}^T \tag{10}$$

where $U_{m\times m}$ and $V_{n\times n}$ are orthogonal matrices, $U^T_{m\times m}U_{m\times m} = I$ and $V^T_{n\times n}V_{n\times n} = I$. $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_{min(m,n)})$, σ_i is called singular values and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{min(m,n)}$.

Before we proceed to present our cluster number determination method, we review the important result from the SVD decomposition [6].

For a SVD decomposition of a matrix A, if $k < r = rank(A)$ and

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T, \tag{11}$$

then

$$\min_{rank(B)=k} \| A - B \|_2 = \| A - A_k \|_2 = \sigma_{k+1} \tag{12}$$

Eq.(12) basically reflects the fact that the rank-k approximation matrix A_k, is just σ_{k+1} away from original matrix A in terms of norm-2 distance.

As a result, by just looking at the singular values and selecting the sufficient small singular value σ_{k+1}, we know how close the corresponding rank-k approximation matrix A_k is to the original matrix A. So the selected rank number k essentially can be viewed as the number of clusters we expect as SVD decomposition essentially reveals the true rank of a matrix in a numerical way. Since the rank of matrix A also reflects the number of independent columns in the given matrix A, so it is naturally for us to use k as the cluster number. We set a threshold ζ to denote the small singular value σ_{k+1}. Typically ζ is set to 10^{-3} in our experiments, which means we regard σ_{k+1} as the sufficient small singular value only if $\sigma_{k+1} \leq \zeta$.

In our method, we select the approximation order not only based on the absolute value of the sufficient small singular value, but also on the relative ratio between two adjacent singular values. If the relative ratio between two adjacent singular values is close to 1, that means there is no big difference if the approximation order is increased by one. In this condition, we will keep increasing the approximation order until the ratio becomes small enough. Thus we first define a threshold ε. Then we compare two adjacent singular values σ_{k+1} and σ_k. If $\sigma_{k+1}/\sigma_k \leq \varepsilon$ and $\sigma_{k+1} \leq \zeta$, then the k is our choice. Typically ε is set to 10^{-3} in our experiments.

IV. K-Means Based Clustering Algorithm

After we determine the number of clusters for the output(or input) terminals, say k, we proceed to group the output(or input) terminals into the k clusters.

In our approach, we apply a so-called K-means algorithm [7] to determine each cluster. K-means is the widely used clustering algorithm for partitioning (or clustering) N data points into k disjoint S_j subsets containing N_j data points to minimize the sum-of-squares criterion:

$$J = \sum_{j=1}^{k} \sum_{i\in S_j} |\mathbf{x}_i - \varphi_j|^2 \tag{13}$$

where $\mathbf{x}_i$ is a vector representing the i^{th} data point, and φ_j is a vector, representing the geometric centroid of the data points in S_j. It has wide applications in data mining and data analysis.

However, the K-means clustering algorithm is very sensitive to the initial choice of cluster centers and the number of clusters [1]. Only the appropriate number of clusters produces the best approximation result. Fortunately in our method, cluster number k can be first obtained from SVD decomposition on the moment matrix as mentioned above.

For a general moment matrix M ($M = M_O, or M = M_I$ for different terminal reductions), it has l terminals to be merged. Let $M = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_l]$.

The proposed clustering algorithm is shown in Fig. 1 based on K-means clustering scheme.

K-MEANSCLUSTER

CONSTRUCT_CLUSTER(k)
1 select k seed vectors out of $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_l$ as k centroids
2 put rest of unselected vectors into nearest cluster S_j
3 compute the centroid φ_j for each cluster S_j
4 **do** re-cluster all $\mathbf{c}_i$ into k clusters according to φ_j of each cluster
5 recompute φ_j for each cluster S_j
6 **until** no change in φ_j
7 **return** $S_1, S_2, \ldots, S_k$ and $\varphi_1, \varphi_2, \ldots, \varphi_k$

SELECT_REP_VECTOR($S_1, S_2, \ldots, S_k$)
1 **for** $i \in S_j$, compute $\|d_i\|^2 = \|\mathbf{c}_i - \varphi_j\|^2$
2 find $R_j = \mathbf{c}_i$ so that $\min\{\|d_i\|^2\}$
3 **return** R
4 end

Fig. 1. K-means based clustering algorithm.

There are two steps (algorithms) in our clustering method. The first step CONSTRUCT_CLUSTER(k) clusters the given l vectors into k clusters ($l > k > 1$). The second step SELECT_REP_VECTOR($S_1, S_2, \ldots, S_k$)) takes the output of the clustering algorithm as the input and finds the representative vector for each cluster. The representative vectors R_j will be kept during the terminal reduction. All the other unselected vector will be suppressed.

The basic idea of CONSTRUCT_CLUSTER(), is to dynamically find the k clusters so that all the vectors in a cluster have the closest distance to their geometric centroid vector φ_j of cluster S_j. CONSTRUCT_CLUSTER() uses an iterative algorithm that minimizes the sum of distances from each vector to its cluster centroid vector φ_j, over all clusters. This algorithm moves vectors among clusters until the sum cannot be decreased any more. In this sense this algorithm is a global optimization method. The result is a set of clusters that are as compact and well-separated as possible.

The second algorithm SELECT_REP_VECTOR () finds the representative vector for each given cluster S_j. This has been achieved by calculating the distance between the centroid and each vector belonging to this cluster to find the terminal with the closest distance to the centroid.

V. TermMerg Algorithm

In this section, we first present the whole terminal reduction flow of the TermMerg algorithm. Then we discuss some practical considerations of the algorithm.

A. TermMerg Algorithm Flow

In this subsection, we present the flow of the TermMerg method.

TermMerg Algorithm

1) Construct the input and/or the output moment matrices.
2) Perform the SVD on the input and/or the output moment matrices to find the best cluster numbers.
3) Invoke K-MeansCluster to find the all the representative terminals for each cluster.

B. Practical Implementation

We have mentioned that we determine the order of moments r based on the fact that the number of moments in the moment series from all the inputs should be equal or large than the number of terminals to be merged as shown in Eq.(8) and Eq.(9). In this way we can get the complete information of the moment matrix after singular value decomposition. However, if we have many outputs and a few inputs, we end up with large r. In other words, we have to use very high order moments. But high order moments actually are not very informative as they contain mainly the dominant pole information numerically [11]. This is also the case for interconnect circuits with many inputs and a few outputs. But on the other hand, large number of outputs or inputs does not imply that the circuit has more independent outputs and inputs, or clusters. Practically the final cluster number may be still very small compared to the number of outputs and inputs. In this case, we do not need many high order moment information to distinguish those terminals. This motivates the following schemes to solve this problem.

We pre-define an effective cluster number q_e, which is smaller than the number of terminals q to be merged but typically is larger than the resulting number of clusters. For example we may define $q_e = \lceil q/i \rceil, i = 2, 3, \ldots$ depending on circuits, where the function $\lceil x \rceil$ means rounding the x to the nearest larger integer. Then the order of moments r would be equal or larger than q_e/p. If the cluster number is qual to q_e after SVD method at some threshold, that means the pre-defined number of clusters is too small to find the optimal cluster number at this threshold. We then either increase the threshold value (at cost of more approximation errors) or increase the pre-defined cluster number q_e to re-select the optimum cluster number by SVD. If we already have had some pre-knowledge about the circuit terminals, it will be very helpful to choice the effective cluster number q_e.

Another way is using a hierarchical terminal reduction method, which is suitable for large number of resulting clusters. First we partition the merging terminals into a number of groups so that their corresponding moment matrix M will not need to use the higher moments. After this we perform SVD on each group and find several representative terminals from each group by K-means clustering method. Then we perform SVD on all the representative terminals to find the best global cluster number. Then we perform K-means algorithm on the top of those selected representative terminals to find the global representative terminals. We may have several hierarchical levels when the number of terminals to be merged is large.

VI. Experimental Results

The proposed method has been implemented in MATLAB. We tested our algorithm on a number of real industry interconnect circuits from our industry partner, Cadence Design System Inc.

The first interconnect circuit *net1026* is an one-bit line circuit from a SRAM circuit in 160nm technology. This network contains 525 resistors, 772 capacitors, 6 drivers and 256 receivers. We perform K-means based TermMerg algorithm to reduce both receiver (output) terminals and driver (output) terminals. Since there are many outputs (256 receivers) comparing with a few inputs (6 drivers), we set effective the cluster number $q_e = 30$. Then we only need $r = q_e/6 = 5$ orders of moments to construct the output matrix M_O.

By using the singular value decomposition, the optimal number of clusters is found to be 5 if we define threshold $\varepsilon = 10^{-3}$. The dominant singular values we find are $\Sigma = diag(1.2 \times 10^6, 2.5 \times 10^3, 27.4, 1.63, 0.12, 5.47 \times 10^{-6}, \ldots)$. It can be seen that there is a big magnitude drop between two singular values 0.12 and 5.47×10^{-6}. So it is naturally to select 5 as the number of clusters.

The final clustering results from TermMerg are shown in Table I. The first column is the cluster index number. The second column is the representative terminal of each cluster. And all the terminals in each cluster are placed in the third column.

TABLE I

The output clustering results for the one-bit lines circuit *net1026*.

Cluster #	Rep. Terminal	Clustered Terminals
1	Rcv206	Rcv151, Rcv152, ..., Rcv256
2	Rcv58	Rcv39, Rcv40, ..., Rcv77
3	Rcv19	Rcv1, Rcv2, ..., Rcv38
4	Rcv98	Rcv78, Rcv79, ..., Rcv119
5	Rcv144	Rcv120, Rcv121, ..., Rcv150

Fig. 2 plots the distribution of terminals (x-axis) with respect to the cluster index number(y-axis).

Then we go back to time domain to validate the effectiveness of our method. We add a voltage source to a driver input to view the step responses at other receivers. Fig. 3 shows the responses of five representative terminals. If we compare the 50% delay time, the delay time difference among them is approximately 10-20ps, which is quite different. The enlarged local waveforms are shown in Fig. 4.

If we plot more responses for all the suppressed terminals in one cluster, for instance *receiver97* and *receiver99*, whose representative terminal is the *receiver98*, we can not tell the difference in responses between these reduced terminals and their representative terminal, *receiver98*, for the delay time as shown in Fig. 5. Detailed analysis shows that the delay time differences among these terminals are only about 1-2ps, which

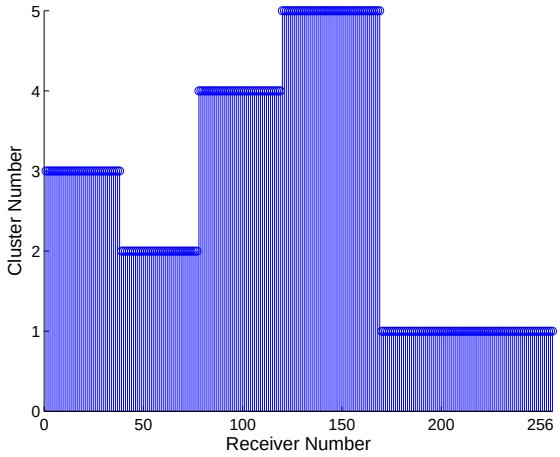

Fig. 2. Output terminal distribution for each cluster for *net1026* circuit.

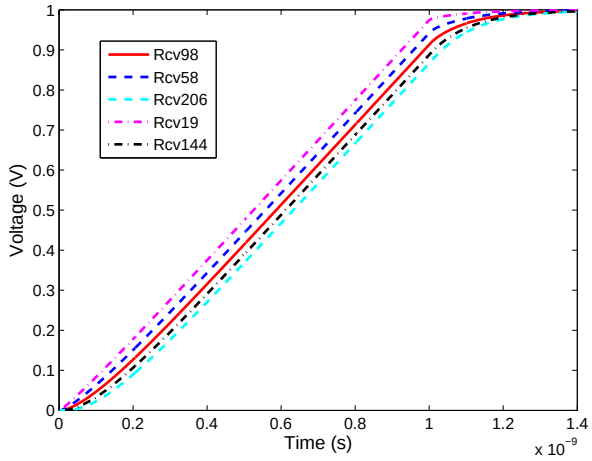

Fig. 3. Step responses of representative output terminals.

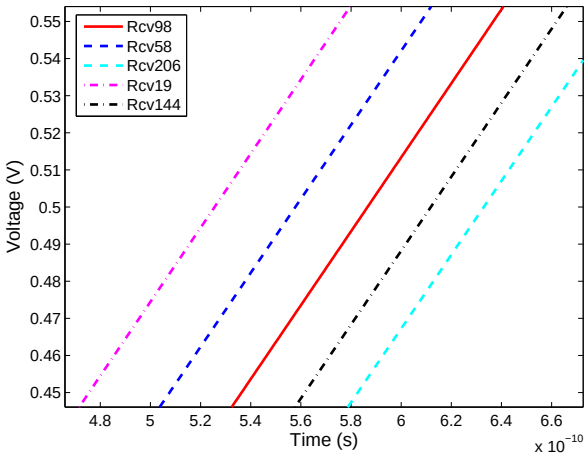

Fig. 4. Comparison of 50% delay time among the representative output terminals.

is clearly shown in Fig. 6. In other words, if we allow 1-2ps delay variations, those terminals can be viewed as the same terminal.

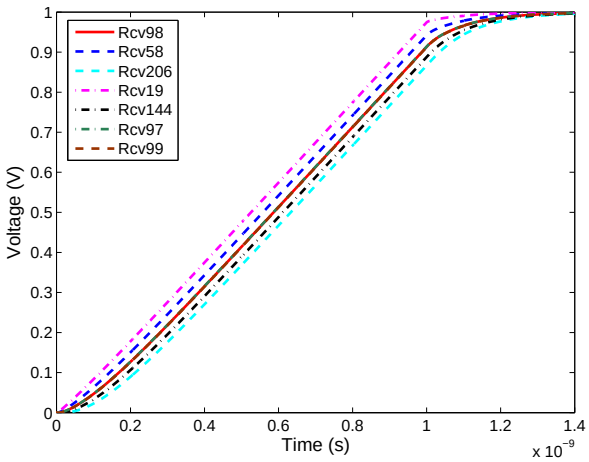

Fig. 5. Step responses of representative output terminals and two suppressed outputs.

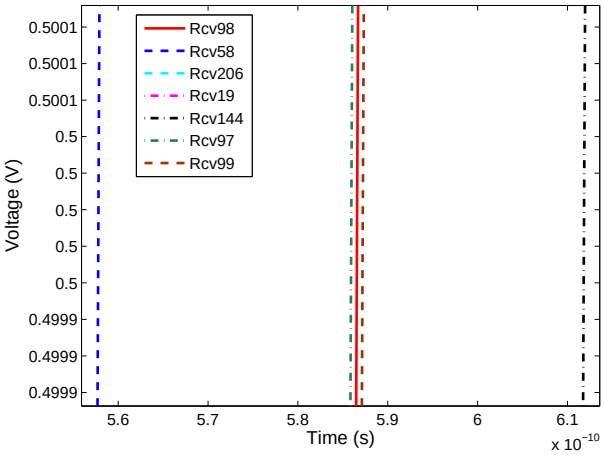

Fig. 6. Comparison of 50% delay time among the representative output terminals and two suppressed outputs.

At this point, we can say that it is reasonable using the response at *receiver98* to represent the responses at the suppressed terminals in its cluster such as *receiver97* and *receiver99*. Considering the process variations and other environmental variations, it is possible that we can combine them into one terminal. Also we can improve the accuracy of this method by relaxing the threshold ε to generate more number of clusters.

For the input terminal merging, we need to cluster the 6 input (6 drivers) terminals. By using the same threshold level $\varepsilon = 10^{-3}$, the cluster number is 5. Only the terminals at *driver3* and *driver4* could be merged together.

The second example, *net27*, is a clock tree circuit also in 160nm technology. It contains 167 resistors, 654 capacitors,

14 drivers and 118 receivers. For the output reduction, we set the effective cluster number $q_e = 118/2 = 59$. Then we only need $r = q_e/14 \approx 4$ orders of moments to format the output matrix M_O. After the SVD step, the output moment matrix M_O has the following singular values: $\Sigma = diag(52.58, 16.85, 3.41, 0.48, 0.024, 5.70 \times 10^{-6}, \ldots)$. Since there is a big magnitude drop between two singular values 0.024 and 5.70×10^{-6}, it is obvious to select cluster number as $k = 5$ at the given $\varepsilon = 1 \times 10^{-3}$.

We also present its distribution of terminals for different clusters in Fig. 7 when we select cluster number $k = 5$. The representative terminals are *receiver98*, *eceiver18*, *receiver110*, *receive36*, *receiver84* corresponding to the cluster from 1 to 5.

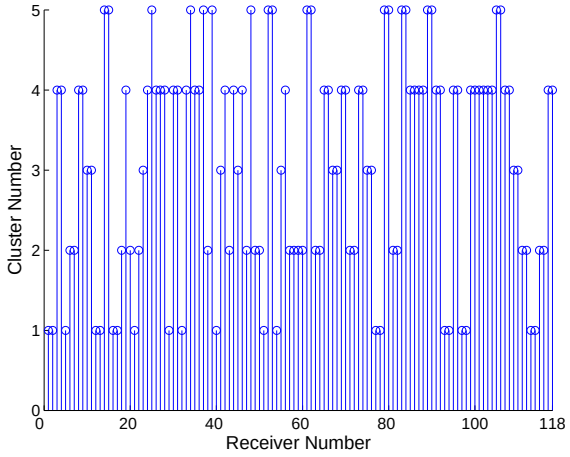

Fig. 7. Output terminal distribution for each cluster for *net27* circuit.

As for the input terminal reduction, the input moment matrix M_I has the following singular values after the SVD: $\Sigma = diag(55.32, 0.091, 1.54 \times 10^{-4}, 8.89 \times 10^{-6}, 8.05 \times 10^{-6}, 6.55 \times 10^{-6}, 6.33 \times 10^{-6}, 6.03 \times 10^{-6}, 5.80 \times 10^{-6}, 6.56 \times 10^{-15}, \ldots)$. If we set the threshold $\varepsilon = 1 \times 10^{-3}$, the number of clusters is $k = 10$. However we notice that there is a big drop between 0.091 and 1.54×10^{-4}. If we relax the threshold to $\varepsilon = 2 \times 10^{-3}$, the cluster number will be only 2. The reduction results will be more efficient but less accuracy. Table II shows the terminal assignment for each cluster when we cluster terminals at $k = 10$ and $k = 2$.

VII. Conclusion

In this paper, we have proposed a novel method named TermMerg to efficiently reduce the terminal number of general linear interconnect circuits considering delay variations. The new method is based on the high order moment responses of terminals in frequency domain as the metrics for the timing or delay. We first applied singular value decomposition method to determine the best number of clusters based on the low-rank approximation. After this, the K-means clustering algorithm was used to cluster the moments of the terminals into the different clusters. Experimental results on a number of real industry interconnect circuits demonstrated the effectiveness of the proposed method.

TABLE II

The iuput clustering results for the clock network circuit *net27* at different thresholds.

Threshold	Cluster #	Rep. Terminal	Clustered Terminals
$\varepsilon = 0.001$	1	Drv2	Drv2, Drv4
	2	Drv11	Drv11
	3	Drv13	Drv13
	4	Drv10	Drv10
	5	Drv8	Drv8, Drv9
	6	Drv6	Drv6, Drv7
	7	Drv14	Drv14
	8	Drv1	Drv1, Drv3
	9	Drv5	Drv5
	10	Drv12	Drv12
$\varepsilon = 0.002$	1	Drv2	Drv2, Drv4, Drv6, Drv7, Drv8, Drv9
	2	Drv12	Drv1, Drv3, Drv5, Drv10, Drv11, Drv12, Drv13, Dvr14

Acknowledge

The authors would like to thank Prof. Yinbo Hua at the Electrical Engineering Department in University of California at Riverside for many insightful discussions on singular value decompositions.

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley and Sons Inc., 2001.

[2] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by pade approximation via the lanczos process," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 5, pp. 639–649, May 1995.

[3] P. Feldmann and F. Liu, "Sparse and efficient reduced order modeling of linear subcircuits with large number of terminals," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 2004, pp. 88–92.

[4] R. W. Freund, "Model order redution techniques for linear systems with large numbers of terminals," in *Proc. European Design and Test Conf. (DATE)*, 2004, pp. 944–947.

[5] ——, "SPRIM: structure-preserving reduced-order interconnect macro-modeling," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 2004, pp. 80–87.

[6] G. H. Golub and C. V. Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[7] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[8] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 645–654, 1998.

[9] J. R. Phillips and L. M. Silveira, "Poor man's TBR: a simple model reduction scheme," in *Proc. European Design and Test Conf. (DATE)*, 2004, pp. 938–943.

[10] ——, "Poor man's tbr: a simple model reduction scheme," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 1, pp. 43– 55, 2005.

[11] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 352–366, April 1990.

[12] M. Silveira, M. Kamon, I. Elfadel, and J. White, "A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 1996, pp. 288–294.