Current Signature Compression For IR-Drop Analysis

Rajat Chaudhry, David Blaauw, Rajendran Panda, Tim Edwards

Advanced Tools, Motorola Inc., Austin-TX

E-mail rajat.chaudhry@motorola.com

Abstract

We present a novel approach to compress current signatures for IR-drop analysis of large power grids. Our approach divides the original current signature into error bounded compression sets. Each compression set has a representative timepoint. The compressed current signature consists of the representative timepoints. The compression technique exploits the pattern of change of individual currents, time locality and periodicity to achieve very high quality results. We provide error guarantees for the compression. Our results are superior in compression and accuracy in comparison to other compression methods such as the single cycle compression.

1. Introduction

In recent years, the power consumption of microprocessors has increased dramatically although the operating voltages have decreased. Due to these trends, the current that is supplied to the processor through the power grid has increased significantly. Subsequently, IR-drop analysis for power grids has become a critical requirement in the design of high performance microprocessors.

A number of efficient analysis tools [1,2,3] have been developed to analyze chip-level power grids containing millions of nodes, and significant research has been published on modeling the power grid accurately[3, 4]. The difficulty in analyzing the voltages in a power grid stems from the fact that the problem is an inherently global one. The voltage and current at a given point in the grid are dependent on the currents at all other points in the grid, and vice versa. Therefore, the power grid must to be simulated as a whole and cannot be partitioned without significant loss in accuracy. Also, the currents drawn by transistors are time-varying, so a transient analysis using time-varying current sources is required for power grid analysis. The effect of the interconnect capacitance on the power grid is typically less than 1-2%, and can be ignored. For explicitly placed decoupling capacitance, the IR drop analysis must be performed together with an inductive package model. However in practice, inductive noise and resonance frequency of the grid are analyzed at a very early stage in the design process and they can be observed with short synthesized current waveforms. Power grids are simulated with current waveforms from long vector traces only in the later stages of the design cycle and the analysis is performed mainly to study IR-drop problems. In this paper we deal only with IR-drop analysis.

It is infeasible to simulate an entire power grid together with all the circuit devices, since the resulting network is both non-linear and very large (1M - 100M nodes). This complexity is managed by separating the non-linear devices (transistors) from the linear power grid interconnect circuit. The analysis is performed in two steps as illustrated in Figure 1: (i) The non-linear devices are simulated using a fast non-linear simulator such as PowerMill[1]. During this simulation, the supply voltage is fixed at Vdd, and the current drawn by each device connected to the power grid is monitored. (ii) The power grid interconnect is simulated using a linear solver with the currents obtained in the first step modeled as time-varying current sources (called current signatures). Since the actual voltage supplied to the devices is less than Vdd and is thus over-estimated in the device simulation, the analysis will over estimate the current signatures (CS) and thereby over estimate the voltage drop in the grid by a small percentage. To manage the runtime of the non-linear device simulation, the circuit is partitioned by circuit blocks which are simulated in parallel. However, the power grid simulation must be solved on the unpartitioned whole using either direct or iterative linear system solvers. Despite the linear nature of the network, this task quickly becomes the dominant factor in the overall run time of the analysis. Also, the linear network is solved at multiple time points within each clock cycle to obtain a transient analysis, and a large number of clock cycles are simulated to adequately verify the integrity of the power grid. Solving a 10M node grid requires approximately 5 minutes with a highly optimized linear solver[3]. Therefore, the analysis of 1000 clock cycles with 40 time points in each requires approximately 5 months of CPU time, which is clearly unacceptable.

Since this run time is unacceptable, a number of approaches have been proposed to simulate the power grid in a reasonable time with conservative results. In [5], a method called IMAX is presented for statically generating a maximum instantaneous peak current signature for one clock cycle. This method is very fast and dramatically reduces the number of time points that are simulated by the linear solver. However, since IMAX does not logically
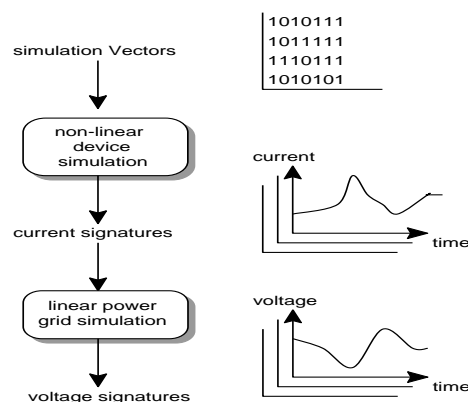
Figure 1. IR-drop analysis flow.

constrain the set of gates that can switch simultaneously, it produces a very loose upper bound on the current signatures and results in a very pessimistic voltage drop analysis. To help address this problem, extensions of IMAX were proposed in [8], using partial input enumeration (PIE), and in [9] using constrained graphs.

Other approaches have automatically synthesized a small number of input patterns using genetic algorithms[7] or ATPG techniques[6]. These methods aim to generate an n-cycle input pattern for the non-linear device simulation which maximizes either the average dynamic power or the peak instantaneous power. A deficiency of these methods is that they do not address the problem of correctly initializing the large state machine under estimation, and thus will generate states which are unreachable. In our experience, using an invalid initial state can significantly increase the peak current predicted by such methods. Also, the produced simulation vectors can be quite large, requiring extensive simulation time for the linear solver.

Some work has been reported on compressing the simulation vectors that are used by the device simulator[11,12,13]. However, these methods typically target preserving the average power of the simulation set, not the peak current. Also, the compressed vectors can still be quite large, and so do not completely alleviate the run time problem.

In this paper, we present a new method that uses current signature compression to dramatically reduce the number of time points to be solved by the linear solver, while producing a conservative analysis result with user controllable error bounds. Using the set of device current signatures produced by the fast non-linear simulation, we generate a new set of current signatures called the *compressed signatures*, such that i) the number of time points in the compressed signatures is greatly reduced from the original signatures, ii) the compressed signatures do not result in a under-estimation of the predicted voltage drop at any point in the power grid, and iii) the error in the predicted voltage drop adhers to a user specified bound.

Using the compressed current signature (CCS), the linear network is simulated with reasonable run time and acceptable error in the predicted voltage drop. The approach can be used to compress current signatures generated from device simulation with either user-generated or automatically generated simulation vectors.

A common and simple approach for compressing the current signatures is to fold them into a single cycle current envelope in which the current at each time point in the single cycle signature corresponds to the maximum current at that timepoint across all clock cycles. Since this approach compresses all clock cycles into one cycle, it achieves very good compression of the time points. However, it does not guarantee any accuracy bound on the obtained results and does not allow the user to trade-off the accuracy of the compression with the compression ratio. Our results in Section 3.0 show that the single cycle compression method is as much as 50% pessimistic. This is due to the fact that the compression is predetermined and does not analyze the current waveforms during compression.

In this paper, we propose a new dynamic method for compressing the current signatures which uses analysis to obtain a maximum compression ratio while minimizing the error of the compression. The error is conservative, and is bound by a user specified limit. It will be shown through experimental results that our proposed scheme typically achieves better quality compression than single-cycle compression, and yet is able to estimate the worst voltage drop with a very small error (15%). This method has been successfully used to verify the global power grid of a PowerPC$_{tm}$ chip with more than 50M nodes, using the tool described in [3].

2.0 Current Signature Compression

Our compression technique takes the set of current vectors in the original current signature and generates a reduced set of current vectors. The reduced set is many times (20X-100X) smaller than the original set and can be simulated in a reasonable amount of time. Due to the reduction, some information is lost and this causes an error in the IR drop values. The error in our technique is always conservative. Our technique also gives an upper bound for the error.

In the proposed compression approach we first scan the current signature and filter out negligible current values. This is done to achieve a higher degree of compression with very little loss in accuracy. After filtering, we start building error bounded compression sets. Each compression set is represented by one simulation timepoint in the final compressed current signature. For each current source, the representative time point has a current value which is the maximum current value for that current source for all time points in the compression set. This guarantees conservatism and introduces an error, which we bound by a user specified bound. We now discuss and step in greater detail.

2.1 Generating Error Bounded Compression Sets

If a circuit has n simulation timepoints and m current sources, we can define its current signature as a set
$$T = \{t_1, t_2......t_n\}$$
where t_i is the vector of values of the m current sources at timepoint i.

Definition: A *Compression Set* CS is a set of one or more timepoint vectors
$$CS = \{t_i\}$$

The compression algorithm creates *y* disjoint *compression sets* the union of which equals *T*. Each *compression set* is converted into one representative timepoint vector. The representative timepoint vector is generated by assigning each current source its max value in the compression set.
If we have a Compression Set
$$CS = \{t_1...t_y\}$$
where *CS* has *y* timepoint vectors, then its representative vector t_{ri} is constructed such that
$$t_{ri} = max(t_{ji})$$
where *i* is the ith current source and *j* is from 1 to *y*. Figure 2. demonstrates the construction of a representative timepoint. If the original current signature *T* is divided into *k* compression sets the degree of compression is *n/k*. The objective of the compression algorithm is to cover all timepoints with compression sets such that the number of compression sets is minimized. When the compression set is converted to a representative timepoint vector, a conservative error is introduced since the representative timepoint vector is generated by taking maximum current values. In order to provide a trade-off between the compression ratio and the error, we need to construct the compression sets,

such that the error introduced for each compression set is bounded by a user specified value K. If each compression set is bounded by K then the overall compressed current signature can be bounded by the K. The obvious way to bound the error is to add timepoints to a compression set only if the difference in the max and min value of each current source is less than K. In this case current values for all the current sources at each timepoint in the compression set are within K of the current values in the representative timepoint. Since the circuit is linear, this guarantees that the error in the voltage drop will be less than or equal to K. This is however, a very conservative approach for IR drop verification. While doing IR drop simulations, we are only interested in the maximum IR drops for each current source, and so it is necessary to bound only the worst case IR drops. The above method bounds all the IR drop values for each current source. Therefore the above method is sufficient but not necessary for bounding the error for the maximum IR drop within a factor of K.

A compression set is valid for an error bound K if one of the timepoint vectors in the compression set satisfies the following relationship for all the current sources.

$$t_{rj} \leq t_{ij}(1+K) \qquad for\ j = 1\ to\ m \qquad (i)$$

where t_{rj} is the value of current source j in the representative timepoint vector t_r, t_{ij} is the value of the current source j in the timepoint vector t_i and K is the error bound.

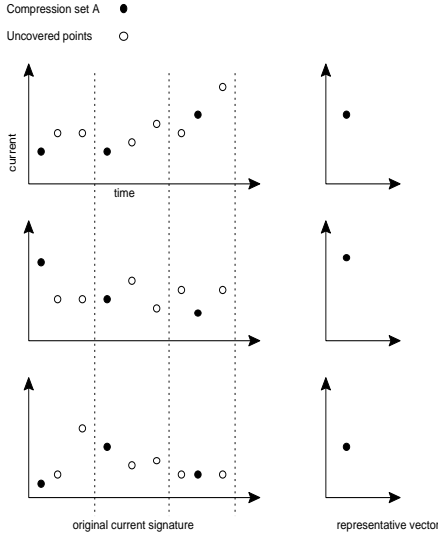Compression set A ●
Uncovered points ○

Figure 2. Representative point creation

The timepoints which satisfy relation (i) are called *guarantee points* because they guarantee the validity of the error bound K. If a compression set has at least one guarantee point, the error introduced on the maximum IR drop for each current source by simulating the representative timepoint vector is bounded by K.

Proof:

Let V_{jmax} be the actual max drop for any current source j. Let V_{jr} be the drop for current source j obtained by simulating the representative timepoint vector. Let V_{jg} be the drop for current source obtained from simulating the guarantee timepoint vector.

From equation (i)

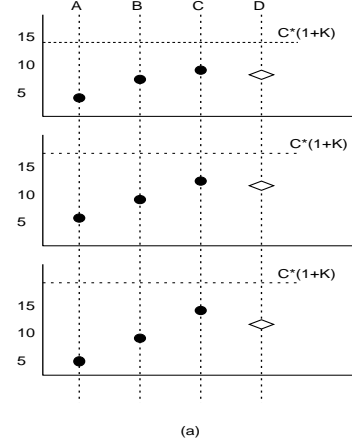$$V_{jr} \leq V_{jg}(1+K) \qquad (ii)$$

Since the guarantee point is an element of the compression set.
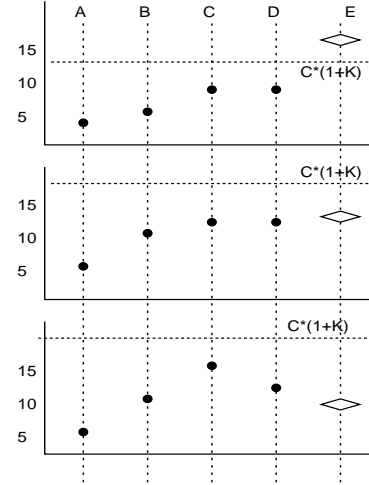
$$V_{jg} \leq V_{jmax} \qquad (iii)$$

From (ii) and (iii)

$$V_{jr} \leq V_{jmax}\,(1+K)$$

Existing compression set ●
Candidate vector ◇   K = 0.4

(a)

(b)

Figure 3. Compression Set Creation

which means that the overestimation in voltage drop is less than or equal to K. Equation (i) shows that the voltage drops for all current sources obtained by simulating the representative vector cannot be greater than K when compared to those obtained by simulating the guarantee point vector. Since the guarantee point vector is one of the actual timepoint vectors from the compression set, its voltage drops can only be less than or equal to the maximum voltage drops. Since the guarantee point drops are bounded by K, the maximum drops must also be bounded by K. In practice, we find

that the actual error is less than *K,* since the representative timepoint vector values are not exactly K greater than the guarantee point values for many current sources. To add a candidate timepoint vector to the compression set, we add the candidate vector and update the maximum current for each current source. We then check if the compression set has any guarantee points. If not the candidate point is removed from the compression set and a new candidate point is selected for adding or we start a new compression set. The addition of candidate timepoint vectors to a compression set is demonstrated by Figure 3. In this example the error bound *K* is 0.4. In figure 3(a) timepoint vector C is the existing guarantee point for compression set "A". Candidate timepoint vector D is tested for adding to the compression set "A". The horizontal dashed line shows (1+K) times the values of vector C. Timepoint vector D is successfully added because after addition of D, C still satisfies equation (i). The maximum values for the set are still below the dashed line. In Figure 3(b) we try to add candidate E but cannot because after E's addition none of the timepoint vectors in the set can satisfy equation (i).

2.2 Compression Set Candidate Selection
Given an error bound *K* and current signature *T,* we need to find the smallest number of valid compression sets which cover all *n* timepoint vectors. The number of ways we can choose the first compression set is equal to the size of the power set of *n* which is 2^n. The number of possible ways to select the second compression set is 2^{n-m}, where *m* is the size of the first compression set. Finding the optimum solution therefore exponential in complexity, forcing us to use a heuristic.

compression set A   →   compression set B   ⇾

(a)
cycle 1 | cycle 2 | cycle 3 | cycle 4 | cycle 5

(b)
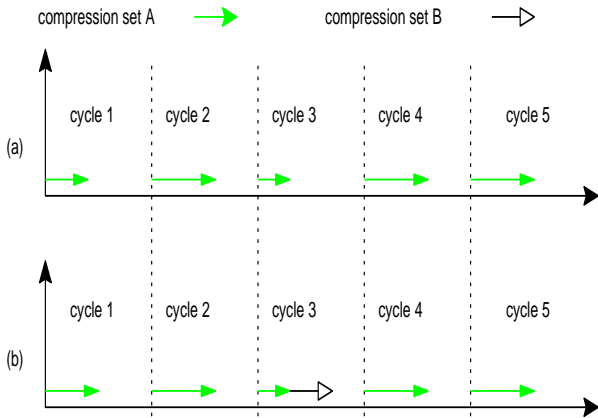cycle 1 | cycle 2 | cycle 3 | cycle 4 | cycle 5

Figure 4. Candidate vector selection

Transistor currents exhibit temporal locality within one clock cycle and are similar between corresponding timepoints of different clock cycles. We have developed a heuristic which captures this temporal locality and periodicity of currents across cycles. We define a *covered* timepoint as a timepoint which has been assigned to a compression set. Our heuristic works in the following way. We start with the first timepoint vector in the current signature and create a compression set. The compression set is grown by selecting the timepoint vectors in the present cycle consecutively until we find a point which cannot be added. We then jump to the first uncovered timepoint vector in the next cycle. This procedure is repeated in each cycle consecutively until we cannot grow the compression set in the last cycle, and we then complete the compression set. This way we complete a

compression set. For the next compression set we choose the starting timepoint from the cycle which has been least covered. This keeps the cycles synchronized, meaning the number of covered timepoint vectors in each cycle remains almost same.

The complexity of successfully adding *n* timepoints to a compression sets is n+m, since the addition of each time point requires one successful comparison with a guarantee point (n operations), and each guarantee point can be eliminated only once (m operations). The overall complexity of adding every time point in the current signature to a compression set is therefore 2*n. The complexity for unsuccessful additions to a compression set is C*g, where g is the number of guarantee points in the compression set and C is the number of clock cycles (since in each clock cycle there will be one time point that is not added). For all compression set, the complexity is C*g*CS = C*n, where CS is the number of compression sets.The overall complexity is therefore 2n + C*n < $O(n^2)$. In practice however, C is much smaller than n, resulting in fast run times.
Figure 4 explains the candidate selection procedure. Figure 4(a) we shows the first completed compression set. The length of the arrow indicates how many points have been covered in each cycle and the dotted lines indicate cycle boundaries. In the first compression set the order for visiting the cycles is 1,2,3,4 and 5. In each cycle candidate points are selected consecutively. For the second compression set we start from cycle 3 because it is the least covered. In the second compression set the order of visiting the cycles is b 3,4,5,2, and 1.

2.3 Signature Filtering
In our compression technique, the first step is to filter out some current sources and current values to improve the degree of compression such that the error added to the error bound is negligible. Since our error bound is a ratio it can be extremely conservative for small currents. Therefore we want to avoid two types of situations.
i) A current source has very small currents compared to other current sources. A big error in the current percentage-wise results in only a small error in absolute units. To avoid the situation where current sources with very small current values hinder the degree of compression we should eliminate such current sources when calculating guarantee points. Therefore while checking for the validity of a compression set we ignore current sources which have small peak current compared to other current sources. The max current value for these eliminated current sources within the compression set is present in the representative timepoint vector, the only difference is that their current values are not used in making decisions while determining whether a timepoint vector is a guarantee point.
ii) A current source may have substantially higher peaks but may also have very small currents in some parts of a cycle. For example a current source may have current of several mA for some parts of the cycle and current of several uA in other parts. Since the worst drop at this current source is determined in all likelihood by the high currents, therefore an increased error during the smaller current intervals will not affect the worst drop. Therefore we ignore current values of a current sources when they are a certain percentage below the peak. The guarantee timepoint vectors do not have to obey relation (i) when t_{ij} is less than certain user-controlled percentage *Y* below the peak current for current source *j*.

2.4 Error Guarantees.

In practice, we have found that filtering increases the degree of compression significantly. However, it introduces a small error in addition to the user specified error *K*. This additional error cannot be theoretically bounded. The most accurate error guarantee can be given when we have the results for the simulation of the original waveform, but that is what we are trying to avoid, instead we give our guarantee by simulating a carefully selected subset of timepoint vectors from the original current signature and then compare the results of this simulation to the results of the compressed signature simulation. This is a conservative guarantee because the set we will simulate may not include the maximum IR drops for all the current sources. Since we are going to give a error guarantee for all current sources we want to select timepoint vectors which have a higher average IR drop for most current sources. In this way we are able to give a reasonable error guarantee for all current sources. Our error guarantee timepoint vector is selected from compression set whose representative timepoint vector gives the maximum voltage drop. This error guarantee timepoint vector has the highest total current within the compression set.

3.0 Results

Table 1 shows the results of our compression algorithm and the single cycle compression algorithm. The original current signature for each circuit had 10,000 timepoints. The first four circuits in the table had 500 simulation timepoints in each cycle and the remaining circuits had 200 timepoints per cycle. These results were obtained for an error bound K=20% and the filtering parameter Y=30%. The table also shows the error for the maximum IR drop current source in our compression and the single cycle compression. One can see that our compression has an error less than 20% in all cases. This shows that the error caused by filtering is small. The results show that in most cases our compression has produced better results both in accuracy and compression. Our compression results are good, and consistent in accuracy in all cases. The error guarantees given by our algorithm are also close to the actual error. For the *processor1* circuit the single cycle compression yields 42% error whereas our approach yields only a 12% error. Our approach also achieved a higher degree of compression than single cycle compression. For some circuits the original maximum drops are not reported because running the complete simulation would have taken more than 10 days of CPU time. Since our error guarantee is conservative, we have assumed that to be the actual error for these circuits. Based on this assumption we have back-calculated the actual drop for these circuits and used those values for calculating the errors for single cycle compression for a reasonable comparison. We have not shown the run times for the compression, which were less than 5 minutes for all cases.

Figure 5 illustrates the results of our compression technique. Fig 5(a) is the original current signature for 4 current sources over a period of 12 clock cycles. figure5(b) shows the results of single cycle compression and figure 5(c) shows the results of our compression. Our compression technique clearly shows much higher efficiency of compression, the number of compressed timepoints is half that of the single cycle compression. In the single cycle compression there are a lot of time points where there is

very little or no current. Our technique eliminates such points since it takes into account the current profile into account while compressing.
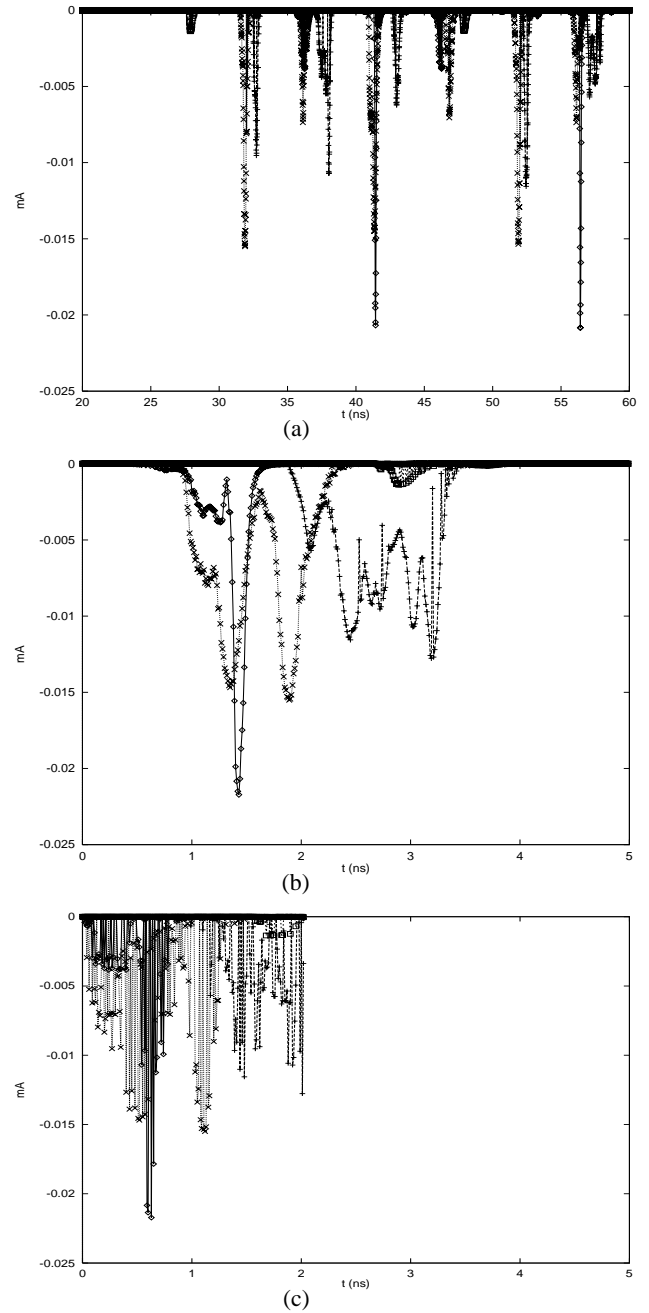

Figure 5. (a) original current signature for 4 current sources (b) single cycle compression of signature in (a). (c) our compression of signature in (a).

4.0 Conclusion

We have developed a compression technique which produces a high degree of compression and also introduces only a small amount of error. Our technique also allows the user to specify the error tolerance. The results show that our

approach produces superior results when compared to other algorithms such as the single cycle compression.

References

[1] RailMill Reference Manual, User Guide, Synopsys Inc, 1997

[2] Lightning: Power Distribution Verification manual,Version 1.1, Simplex Solutions Inc.

[3] "Design and Analysis of Power Distribution Networks in PowerPC Microprocessors"

[4] H.H. Chen and D.D.Ling, "Power Supply Noise Analysis Methodology for Deep-Submicron VLSI Chip Design," DAC-97, pp638-643

[5] H. Kriplani, et. al., "Maximum Current Estimation in CMOS Circuits," DAC-92, pp 2-7.

[6] A. Krstic, et. al., "Vector Generation for Maximum Instantaneous Current through Supply Lines for CMOS Circuits," DAC-97, pp383-388.

[7] M.S. Hsiao, et. al., "K2: An Estimator for Peak Sustainable Power of VLSI Circuits," DAC-97, pp 178-183.

[8] H. Kriplani, et. al, "Resolving Signal Correlations for Estimating Maximum Currents in CMOS combinational Circuits," DAC-93, pp 384-388

[9] S.Bobba, I.N. Hajj, "Estimation of Maximum Current Envelope for Power Bus Analysis and Desig," Int. Symp on Phys Des.(ISPD98), Monterrey, CA, April 1998

[10] Radu et. al, "Hierarchichal Sequence Compaction for Power Estimation", DAC-97, pp570-575

[11] S.Y. Huang et. al., "Compact Vector Generation for Accurate Power Simulation", DAC-96, pp 161-164

[12] R. Radjassamy, and J. D. Carothers, "Vector Compaction for Efficient Simulation Based Power Estimation", Intl. Conf. on Packaging and IC Design Integration, 1998

[13] C. Y. Tsui et al, "Improving the Efficiency of Power Simulators by Input vector Compaction", DAC-96, pp 165-168

Circuit	Our Compression Timepoints	Our Method Compression ratio (single cycle)	Number of Current Sources	Actual Worst Drop	Our Compression Worst Drop(%error)	Our Compression Worst drop Error Guarantee	Single Cycle Worst Drop(%error)
Integer1	269	37X (20X)	912	260mV	295mV(13.5%)	17.8%	365mV(40.4%)
fpu1	353	28X (20X)	1007	91mV	100mV(9.9%)	9.9%	112mV(23.1%)
sequence	168	60X (20X)	1305	126mV	138mV(5.4%)	11.1%	134mV(6.4%)
processor1	353	28X (20X)	4010	260mV	293mV(12%)	18.1%	369mV(41.9%)
processor2	253	40X (50X)	35032	NA	61mV(12.5%)	12.5%	66mV(20%)
fpu2	165	60X (50X)	5233	NA	40mV(0.57%)	0.57%	40mV(0.57%)
integer2	172	58X (50X)	4603	NA	41mV(5.3%)	5.3%	52mV(33%)
circuit3	236	42X (50X)	3606	NA	44mV(8.3%)	8.3%	54mV(34%)
circuit4	257	39X (50X)	3606	NA	43mV(12%)	12%	50mV(31.5%)
circuit5	240	42X (50X)	3606	NA	41mV(12%)	12%	49mV(29%)
circuit6	137	73X (50X)	2249	NA	31mV(5.9%)	5.9%	33mV(10%)
circuit7	42	238X (50X)	2249	NA	62mV(12%)	12%	67mV(21%)

Table 1: Compression Results