

Static Timing Analysis Considering Power Supply Variations

Sanjay Pant, David Blaauw
University of Michigan, Ann Arbor, MI

Abstract

Power supply integrity verification has become a key concern in high performance designs. In deep submicron technologies, power supply noise can significantly increase the circuit delay and lead to performance failures. Traditional static timing analysis which applies worst-case voltage margins to compute circuit delay leads to a very conservative analysis because the worst-case drop is localized to a small area of the die. In this paper, we propose a new approach for analyzing the impact of power supply variations on circuit delay. The circuit delay maximization problem is formulated as a constrained non-linear optimization problem which takes both IR and Ldi/dt drops into account. The proposed approach does not require apriori knowledge of critical paths in the circuit and can be effectively incorporated in an existing static timing analysis framework. The proposed method has been implemented and tested on ISCAS85 benchmark circuits and compared with the traditional methods for computing worst-case circuit delay under supply variations.

1. Introduction

Power supply networks are essential in providing the devices on a die with a reliable and constant operating voltage. Due to the interconnect resistance and inductance of the on-chip and package supply networks, the supply voltage delivered to various devices on a die is not ideal and exhibits both spatial and temporal fluctuations. These fluctuations in the supplied voltage can result in a reduction in operating frequency and can compromise functional stability. Power supply integrity is, therefore, a critical concern in high-performance designs.

The voltage drop that develops in a supply network can be broadly classified into *IR-drop*, which is the voltage drop due to the parasitic resistances of the interconnects and *Ldi/dt drop*, which is the voltage drop due to the inductance of I/O pads and the parasitic inductance of the supply interconnects. In today's high-end designs, it is not uncommon for the supply network to conduct hundreds of Amperes of total current. As semiconductor technology is scaled and the supply voltage is reduced, the total current that must be supplied by the power network is expected to increase even further, making it more difficult to meet stringent supply integrity constraints. In particular, the Ldi/dt voltage drop is expected to become more prominent as it exacerbates with both increasing current demand and clock frequency [1]. As a result, the effect of IR as well as Ldi/dt drop can no longer be ignored during performance verification.

The voltage fluctuations in a supply network can inject noise in a circuit, leading to functional failures in the design. Extensive work has, therefore, been focused on modeling and efficient analysis of the worst-case voltage drop in a supply network [2-5]. However, with decreasing supply voltages, the gate delay is becoming increasingly sensitive to supply voltage variation as the headroom between V_{dd} and V_t is consistently being reduced. With ever diminishing clock cycle times, accurate analysis of the supply voltage impact on circuit performance has, therefore, become a critical issue.

Power supply variation can impact the circuit delay in two ways. First, a reduced supply voltage lessens the gate drive strength, thereby increasing the gate delay. Second, a difference in the supply voltage between a driver-receiver pair creates an offset in the voltage

with which the driver/receiver gates reference the signal transition. This has the effect of creating either a positive or a negative time shift in perceived signal transition at the receiver gate. This dual nature of the supply voltage impact on circuit delay was observed in [6] and complicates the generation of simulation vectors that maximize the delay along a particular circuit path. Increasing the voltage drop at a particular location may worsen the delay of one gate while improving the delay of another. Therefore, determining the path with worst delay under these often conflicting goals is a complicated task.

Traditionally, the impact of supply noise on delay has been accounted for by reducing the operating voltage of all library cells by the worst-case supply voltage drop during library characterization. This assumes that the expected worst-case voltage drop occurs at all places in the design. This yields a very conservative analysis since, in practice, the worst drop can occur in only a small region at any one point in time. On the other hand, this approach ignores the impact of voltage shifts between driver/receiver pairs, thereby possibly underestimating the worst-case delay in certain situations.

A number of methods have been proposed to estimate supply noise induced worst-case delay. [7] proposed a technique to search vector patterns which produce the worst-case supply noise effects on critical paths in a circuit. An approach to compute the upper bound on circuit delay under voltage variations was presented in [8] Recently, several approaches, [9][10], have been proposed to estimate the effect of supply noise on circuit performance where the delay maximization problem has been formulated as an optimization problem with constraints on currents being drawn from the supply network. However, all these approaches are path-based and require enumeration of all the critical paths of a circuit. It has been observed in [11] that the delay of a path can increase significantly under a 10% drop in V_{dd} and V_{ss} supplies. The increase in delay of a path depends on the placement of the gates constituting the path. Gates located in the region with higher supply drop will, in general, experience a larger increase in delay due to drive strength reduction, as compared to the gates which have access to supply with smaller drop. Thus, in a fairly balanced circuit, a path which is critical under nominal voltage supply may no longer remain critical under supply variations if it is located in the region with lower supply drop. On the other hand, even relatively non-critical paths can become critical if they are located in the worst voltage drop region. To our knowledge, there is no prior work to determine the effect of power supply variations on circuit delay as a whole and does not require apriori knowledge and enumeration of critical paths.

In this paper, we present a new approach for computing the worst-case circuit delay under supply variations, that does not require enumerating all the critical paths and can be effectively incorporated in a traditional static timing analysis framework. The proposed approach is vector-less, allowing for efficient analysis, and addresses both IR-drop and Ldi/dt drop effects. The task of determining the worst-case impact of supply noise on circuit delay is formulated as a constrained non-linear optimization problem where the currents being drawn from the supply network are the optimization variables. The constraints on currents of logic blocks can be extracted from extensive gate level simulation data that is readily available during the design process, thereby avoiding the need for time consuming power grid simulation. The proposed approach has been implemented and tested

on ISCAS85 benchmark circuits and shown to accurately compute the increase in circuit delay.

The remainder of this paper is arranged as follows. Section 2 describes the model used for delay variations with respect to supply voltage fluctuations. Section 3 presents the delay maximization problem under supply variations as a non-linear problem for maximizing the impact of power grid fluctuations on delay. Section 4 presents the results obtained for various benchmarks considering the impact of both static and dynamic voltage drops on circuit delay. In Section 5, we draw our conclusions.

2. Delay Model for Supply Variations

In this section, we present the modeling of the impact of voltage variations on the delay of a circuit. Since the voltage variations in a power grid are typically very slow compared to the transition time of a switching gate [12], we can make the simplifying assumption that the supply voltages are constant during the switching transition of a gate. We are, therefore, concerned with the impact of fixed voltage offsets from the nominal V_{dd} and V_{ss} voltages on the delay of a circuit path. Note, however, that *dynamic* IR-drop and Ldi/dt drop effects will be the cause of these voltage offsets.

Voltage drop at a power supply point can impact the delay of a gate through one of the following two mechanisms.

1. A decrease in the V_{dd} voltage or an increase in the V_{ss} voltage at the gate under consideration decreases the *locally* observed supply voltage of the gate and will reduce its drive strength and hence increase its delay.
2. A relative shift in the V_{dd} or V_{ss} voltages between the driver and receiver gates of a signal net can introduce a voltage offset that will impact the delay of a gate. This relative shift between the driver and receiver gates is likely to be larger if the gates are separated farther apart as compared to when they are located close to each other.

The relative magnitude of the above two mechanisms depends on the input slope, output loading and physical placement of a gate and its driver gate. We now present the model for the dependence of the delay of a single gate on the voltage drops at that gate and at its preceding gate.

2.1 Gate Delay Model

Consider the delay of a gate G , shown in Figure 1, with local supply voltages $V_{dd,g}$, $V_{ss,g}$ and supply voltages $V_{dd,in}$, $V_{ss,in}$ at the preceding driver gate. The propagation delay, τ between the input and output transitions of a gate is measured at 50% points of the switching waveforms and the transition time is measured between 30%-70% of the switching waveform. The delay of the receiver gate depends on the $V_{dd,g}$ and $V_{ss,g}$ voltages at the receiver gate itself, the voltages $V_{dd,in}$, $V_{ss,in}$ at the preceding driver gate, the input transition time and the output load. In general, propagation delay and transition

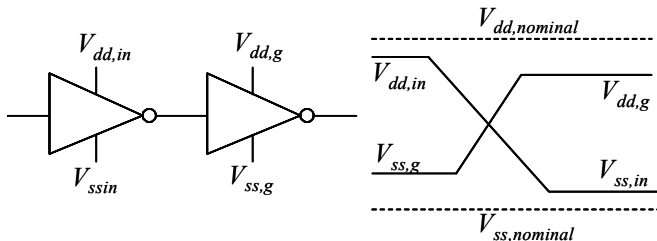


Figure 1. A driver-receiver pair in a non-ideal supply network

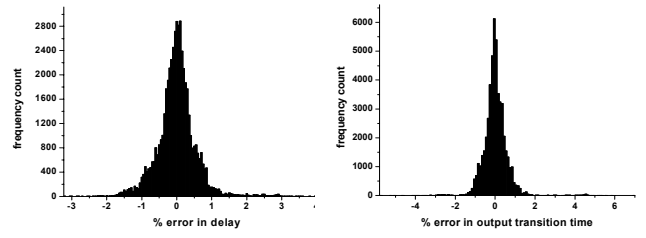


Figure 2. Cell characterization error for delay and transition time

time are nonlinear functions of the supply voltages. However, the voltage drop (ground rise) in a power grid network is restricted and is typically within the range of 10% of $V_{dd,nominal}$ ($V_{ss,nominal}$). Within this range, the delay and output transition times of a gate can be approximated as quadratic functions of supply voltages without inducing a large error. For a fixed output load and input transition time, the propagation delay, τ and output transition time, tr_{out} of a gate (for each transition type) are characterized as quadratic functions of the supply voltages at the gate and its driver.

$$\tau = \sum_i b_i \Delta V_{dd,in}^{\alpha_i} \Delta V_{ss,in}^{\beta_i} \Delta V_{dd,g}^{\gamma_i} \Delta V_{ss,g}^{\delta_i} \quad (1)$$

$$tr_{out} = \sum_i c_i \Delta V_{dd,in}^{\alpha_i} \Delta V_{ss,in}^{\beta_i} \Delta V_{dd,g}^{\gamma_i} \Delta V_{ss,g}^{\delta_i} \quad (2)$$

$$\alpha_i + \beta_i + \gamma_i + \delta_i \leq 2 \quad (3)$$

where $\alpha, \beta, \gamma, \delta \in [0, 1, 2]$ and coefficients $b_i, c_i \in \mathfrak{R}$.

Traditional standard cell libraries are composed of two dimensional tables with table entries representing delay and transition times for different load-transition time combinations. We modify the library to incorporate the coefficients b_i and c_i in place of a fixed delay entry. For a given set of transition time, load and supply voltages at a gate and its driver, the delay is computed appropriately using (1) and (2). Each set of load-transition time combination corresponds to a fixed set of coefficients b_i and c_i which are obtained by performing multiple regression analysis where each gate is simulated over a range of supply voltage variations and rise/fall transition changes. Figure 2 shows the error distribution of delay modeling for 0.13 μ standard cells used in our experiments. The model has maximum error of 4.33% and 6.92% for delay and output transition time respectively. The error in propagation delay and transition have the standard deviation of 0.56% and 0.65% respectively.

The input transition time at a gate G is a function of the supply voltages at the at gates prior to gate G . In other words, a drop in supply voltage at a gate may impact the transition time, and thus, the delay of all the *downstream* gates in the path. How to take into account these changes in transition times in the analysis will be discussed later in Section 3.4

3. Delay Maximization Problem

In this section, we formulate the delay maximization problem as a non-linear optimization problem with constraints on currents drawn from the power grid. Section 3.1 describes the problem expressed in terms of arrival times at the gates. Computing the worst-case delay of a circuit involves determining the worst-case critical path under supply variations in the design and maximizing its delay. The arrival time at the output of a gate itself involves a *max* function. This *max* function complicates the formulation and does not allow for it to be solved directly using an NLP solver. We therefore show how the *max* operations can be eliminated by the introduction of slack variables

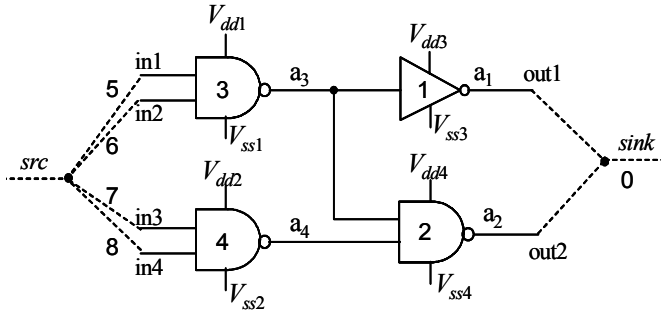


Figure 3. A combinational circuit

with additional constraints, allowing the use of standard NLP optimization methods. Section 3.2 describes the sensitivity computation of voltage drops on currents. Constraints on currents, which are the independent optimization variables, are explained in Section 3.3. Section 3.4 proposes an iterative algorithm to account for changes in input transition time of a gate due to supply drops at earlier preceding gates. Section 3.5 presents a simple and effective circuit pruning approach to reduce the complexity of the non-linear optimization.

3.1 NLP Formulation

Consider a combinational circuit with s primary inputs (PIs), t primary outputs (POs) and n gates or wire segments. The power and ground supplies at a gate i are denoted as Vdd_i and Vss_i respectively. Two fictitious nodes src and $sink$ are added to the circuit. All the primary inputs are connected to the src node and all the primary outputs are joined together to form the $sink$ node. The $sink$ node connecting all the POs is labeled as node 0 and all other gates are numbered in the reverse topological order [13]. The arrival time at the output of a gate i is denoted by a_i . For $0 \leq i \leq n$, let $input(i)$ be the set of indices of gates driving the inputs of gate i . For $1 \leq i \leq n + t$, let $output(i)$ be the set of indices of gates in the fanout of gate i . For example in Figure 3, $s=4$, $n=4$, $t=2$, $input(0)=\{1,2\}$ and $output(3)=\{1,2\}$.

For sake of brevity, we do not differentiate between rising and falling transitions in the discussion although they have been taken into account in our implementation. Let τ_{ij} represent the delay of gate j from one of its inputs i and b_{kj} denote the delay coefficients for gate j . Then the delay maximization problem, given the constraints on gate supply voltages can be stated as follows:

$$\text{maximize } a_0 \quad (4)$$

$$\text{s.t. } a_j = \max\{a_i + \tau_{ij}\} \quad 0 \leq j \leq n \wedge \forall i \in input(j) \quad (5)$$

$$a_j = 0 \quad n + 1 \leq j \leq n + t \quad (6)$$

$$\tau_{ij} = \sum_k b_{kj} \Delta Vdd_i^{\alpha_i} \Delta Vss_i^{\beta_i} \Delta Vdd_j^{\gamma_j} \Delta Vss_j^{\delta_j} \quad 1 \leq j \leq n \wedge \forall i \in input(j) \quad (7)$$

$$\tau_{i0} = 0 \quad \forall i \in input(0) \quad (8)$$

and constraints on voltage drops ΔVdd_i , ΔVss_i at gate i

The objective function is the worst-case arrival time at the output $sink$ node. (5) and (6) set the arrival time at the output of each gate and the $sink$ node. (7) and (8) express gate-delay as a function of the voltage drops at the gate and its driver. The constraints on voltage drops at each gate as a function of currents are incorporated in a set of additional constraints which will be discussed later in Section 3.2.

Eliminating the \max function

The delay maximization problem described above involves the \max function which cannot be used per se for optimization purposes.

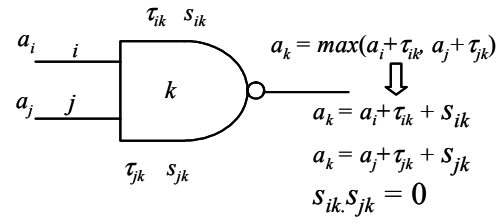


Figure 4. Removal of \max function

A common technique is to replace the \max operation where inequalities are used in place of the \max operator[13]. Although this works for problems such as gate sizing where circuit delay is minimized, it cannot be used in our proposed formulation for maximizing the delay because it makes the objective function unbounded. We, therefore, propose a different method of elimination of the \max function by introduction of slack variables along all the possible input-output pairs in a gate as shown in Figure 4. For instance, in an n input gate, one slack variable is introduced for each possible path from inputs to output. This results in n slack variables for an n input gate. The complexity of the modified formulation depends on the number of multiple input gates in the design. Using these slack variables, the arrival time at the output of an inverting gate j can be stated as follows:

$$a_j = a_i + \tau_{ij} + s_{ij} \quad 0 \leq s_{ij} \leq n \wedge \forall i \in input(j) \quad (9)$$

At least one of the slacks, s_{ij} has to be zero since at least one of the possible paths from inputs to the output will be critical during circuit operation. Therefore, an additional constraint is added for each \max function which ensures that at least one of the slack variables s_{ij} is 0.

$$\prod_{i \in input(j)} s_{ij} = 0 \quad s_{ij} \geq 0, \quad 0 \leq j \leq n \quad (10)$$

This additional constraint places an upper bound on arrival times a_j and the modified maximization problem can be efficiently solved using industrial non-linear solvers.

In the next subsection, the computation of voltage drop sensitivity to currents of logic blocks is discussed.

3.2 Voltage Drop Sensitivity Computation

The power supply network of a chip consists of the ideal supply voltage sources, power and ground wires modeled as a linear RLC network, time-varying current sources representing switching transistors and decoupling capacitances. A design can consist of millions of transistors forming a sea of gates. Simulation of a power supply network at transistor level is not feasible even for moderate sized designs. In block-based power grid simulation, this sea of gates is grouped into large blocks and currents drawn by these blocks are used for power grid simulation. This simplified model of the power supply network consists of RLC elements, ideal time-varying current sources and ideal voltage sources.

We are interested in the sensitivity of voltage variations at the supply points of logic blocks to the block currents. Thus, the simplified linear model of the power supply network can be considered as a linear system with time-varying block currents as inputs and voltage variations at the supply connections as outputs (Figure 5). This system has the impulse response function given by a matrix $H(t)$, whose element at row b and column n denotes the impulse response at node

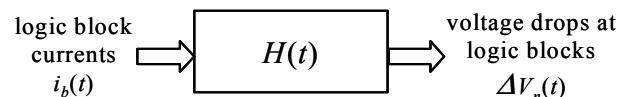


Figure 5. Power grid as a linear system

n due to current block b . Since the system is linear, the voltage response at node n , $V_{nb}(t)$ due to any current waveform of block b , i_b is given by convolution as follows:

$$V_{nb}(t) = \int_0^\infty i_b(t-\tau) \cdot h_{nb}(\tau) \cdot d\tau \quad (11)$$

where, $h_{nb}(\tau)$ is the impulse response at node n due to the excitation at block b .

If the total number of blocks in the design is B , then the voltage response at node n due to all the current blocks acting together, $V_n(t)$ is the superposition of individual responses as shown below.

$$V_n(t) = \sum_{b=0}^{B-1} \int_0^\infty i_b(t-\tau) \cdot h_{nb}(\tau) \cdot d\tau \quad (12)$$

For static or DC-only power grid analysis, the impulse response $h_{nb}(t)$ is a DC value. There are many ways of computing the impulse response $h_{nb}(t)$ at a node n due to a block current i_b . For static analysis, the sensitivity can be computed using the inverse of conductance matrix as follows [14]:

$$GV = I \Rightarrow V = G^{-1}I \quad (13)$$

$$h_{nb} = [G^{-1}]_{nb} \quad (14)$$

where, G is the conductance matrix of the grid and consists of both power and ground grid nodes, I is the vector representing currents between power and ground grid and V is the voltage drop at the power and ground nodes. For large networks, computation of G^{-1} can be computationally intensive and a random walk based sensitivity computation has been proposed recently in [15].

However, since we are also interested in dynamic power grid analysis where block currents can vary with time, we first obtain the step response at node n by applying a unit step current at all the supply nodes of block b and simulating the grid. In this work, we distribute each block's current evenly among all its supply nodes. However, if the block current distribution among its supply nodes is known beforehand, unit step response for the block can be appropriately distributed among its supply points. The unit step response at all the nodes is discretized with a sufficiently small time step T_s and then numerically differentiated to obtain the impulse response for the block. For typical grids, the unit pulse response dampens out quickly (within K time steps) and the grid needs to be simulated only for a small period of time. Given a sufficiently fine grain discretization and sufficient simulation length of the unit step response, arbitrary accuracy can be obtained.

After representing the voltage drops as a linear function of block currents using (12), we need to construct the constraints of block currents. The number of block current related optimization variables and generation of constraints is discussed in the next subsection.

3.3 Constraints on Currents

In this work, we assume that the minimum and maximum currents of logic blocks are known by means of simulation or based on the estimate of a prior manufactured part. Current drawn by a block at every time step is constrained to be within these bounds. We further constrain the sum of all block currents to not exceed the peak current consumption of the chip. For dynamic or Ldi/dt analysis, voltage drop in a time step depends on the currents drawn in K previous cycles, where K is the number of time steps after which the impulse response for all the block currents have reached a steady state value. Any switching activity which has happened more than K time steps

prior to the time step of interest has no effect on the voltage drop and hence, has no effect on the change in delay during the time step of interest. In all, BK optimization variables need to be introduced, K variables for each block current. The constraints on these current variables are listed below:

$$i_{b,min} \leq i_b[n] \leq i_{b,max} \quad \forall n \in \{0, 1 \dots K-1\} \quad (15)$$

$$0 \leq \sum_{b=0}^{B-1} i_b[n] \leq I_{peak} \quad \forall n \in \{0, 1 \dots K-1\} \quad (16)$$

where, $i_b[n]$ is the current of block b in time step n ; $i_{b,min}$, $i_{b,max}$ are the minimum and maximum currents of block b ; I_{peak} is the peak current consumption of the chip and B is the total number of logic blocks in the design. Besides the constraints in (15) and (16), specific current constraints can be obtained by analyzing the current trace of gate-level or transistor-level simulations of blocks.

The overall NLP formulation for delay maximization with constraints on block currents is expressed below:

$$\text{maximize } a_0 \quad (17)$$

$$\text{s.t. } a_j = \max\{a_i + \tau_{ij}\} \quad 0 \leq j \leq n \wedge \forall i \in \text{input}(j)$$

$$a_j = 0 \quad n+1 \leq j \leq n+t$$

$$\tau_{ij} = \sum_k b_{kj} \Delta V dd_i^{\alpha_i} \Delta V ss_i^{\beta_i} \Delta V dd_j^{\gamma_j} \Delta V ss_j^{\delta_j} \quad 1 \leq j \leq n \wedge \forall i \in \text{input}(j)$$

$$\tau_{i0} = 0 \quad \forall i \in \text{input}(0)$$

$$\Delta V_j = \sum_{b=0}^{B-1} \sum_{i=0}^{j-1} i_b[j-i] \cdot h_{nb}[i] \quad 1 \leq j \leq n$$

$$i_{b,min} \leq i_b[n] \leq i_{b,max} \quad \forall n \in \{0, 1 \dots K-1\}$$

$$0 \leq \sum_{b=0}^{B-1} i_b[n] \leq I_{peak} \quad \forall n \in \{0, 1 \dots K-1\}$$

$$\prod_{i \in \text{input}(j)} s_{ij} = 0 \quad 0 \leq s_{ij}, 0 \leq j \leq n$$

3.4 Transition Time Variations

The input transition time at any gate j in a circuit is a function of the supply voltages at its preceding gates. It is possible that the supply voltages at the preceding gates affect input transition time at gate j . The delay and output transition time of the gate are dependent on its input transition time and should, therefore, reflect the changes in input transition time. In other words, the delay coefficients b_j and c_j for delay and output transition time for a gate j in (1) and (2) are a function of input transition time, which can vary depending on the supply at the prior gates. In our experiments, we observed that representing the delay coefficients as a function of input transition time in (17) leads to significant additional complexity of the NLP. Instead of adding this extra complexity to the NLP, we propose a simple iterative approach for delay correction due to changes in transition times.

As a first step, an initial block current distribution i_b which satisfies the current constraints (15) and (16) is identified. The constraints can be satisfied either by setting all block currents in all time steps to zero or by distributing the peak current appropriately among the blocks. These block currents are then used to obtain the voltage drops, ΔV_n at the supply points of the gates in the given circuit. The

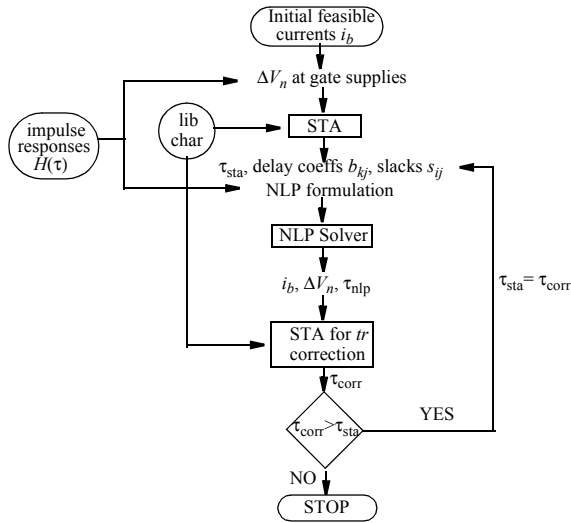


Figure 6. Overall flow of delay maximization problem

static timing analyzer, which has been modified to incorporate supply variations, computes delay coefficients b_{ij} , c_{ij} and slacks s_{ij} for each gate j on the basis of input transition times and output loading. The resulting worst-case delay from the STA, based on the initial current distribution, is denoted by τ_{sta} . The initial guess for current distribution, along with the delay coefficients and slacks obtained from STA, are used to formulate the delay maximization problem as an NLP problem. NLP computes the block current waveforms so as to maximize the worst-case delay of the circuit which is represented as τ_{nlp} in Figure 6. This worst-case NLP delay τ_{nlp} is corrected to account for any changes in the transition times by re-running STA with the new voltage drop values computed by NLP. The delay coefficients computed during corrected STA are used to re-formulate the NLP for the next iteration and the process is repeated until the corrected STA output τ_{corr} is either equal to or less than the pre-NLP delay τ_{sta} in each iteration.

The delay maximization problem involves the maximization of the max function and has a non-convex feasible region and the feasible region of the proposed formulation in (17) is also not convex. Thus, the solution obtained from NLP is a local maximum and is not guar-

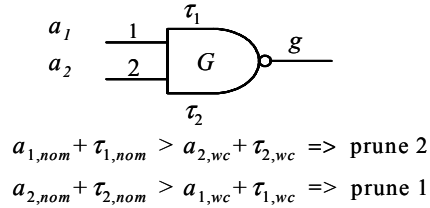


Figure 7. Example illustrating pruning of inputs

anteed to be the global maximum. Hence, we have compared the solution found by NLP against the solutions found through extensive random simulations and have observed that the NLP solution closely matches the solution obtained through 50,000 random runs of STA.

The runtime of the NLP solver depends largely on the number of max functions for multi-input gates in the circuit. The next subsection describes a technique to reduce the number of inputs of a multi-input gate, thus reducing the runtime of the non-linear solver.

3.5 Delay Based Circuit Pruning

We propose a delay based circuit pruning technique to, if possible, eliminate the inputs of multi-input gates in the circuit. Consider a single-output, n -input gate G with inputs numbered as $0,1,\dots,n$. The arrival time at the output of this gate is the maximum of the n arrival times through n possible paths from inputs to the output. As a first step, the arrival times at the inputs of the gate are computed using an ideal voltage supply and the supply with the worst-case voltage margin. These are denoted as $a_{i,nominal}$. If output arrival time from any input under worst case voltage margin is found to be greater than the output arrival time from all other inputs under ideal voltage supply, the input is removed from the gate and the gate is treated as a $n-1$ input gate. Figure 7 shows an example when input 2 of a 2-input nand gate is removed by pruning.

The next section shows the experimental results obtained on ISCAS85 benchmarks for both static IR-only and dynamic IR+Ldi/dt power grid analysis.

4. Experimental Results

The proposed approach for determining the worst-case voltage drop of a given circuit was implemented and tested on ISCAS85

ckt	num. of gates	nom. delay (ns)	traditional approach delay				proposed approach delay				run time	mem. usage
			10% margin		avg. currents		NLP delay(ns)	% incr.	hspice (ns)	% err with hspice		
			delay(ns)	% incr.	delay(ns)	% incr.						
c17	7	0.109	0.154	41.3%	0.122	11.9%	0.131	20.2%	0.129	1.81%	0.24s	1.08M
c432	212	1.224	1.728	41.2%	1.393	13.8%	1.471	20.2%	1.479	0.54%	2.70s	4.83M
c499	553	0.824	1.156	40.3%	0.947	14.9%	0.999	21.2%	1.011	1.20%	25.34s	7.77M
c880	568	1.550	2.190	41.3%	1.771	14.3%	1.857	19.8%	1.916	3.08%	9.39s	7.58M
c1355	654	1.393	1.945	39.6%	1.586	13.9%	1.671	19.9%	1.696	1.45%	19.46s	8.76M
c1908	543	1.912	2.680	40.2%	2.183	14.2%	2.296	20.1%	2.338	1.80%	10.25s	7.61M
c2670	1043	1.512	2.125	40.5%	1.736	14.8%	1.848	22.2%	1.957	5.55%	30.46s	11M
c3540	1492	2.063	2.907	40.9%	2.353	14.1%	2.462	19.3%	2.407	2.29%	1m57s	15M
c5315	2002	1.966	2.798	42.3%	2.238	13.8%	2.379	21.0%	2.492	4.54%	3m12s	20M
c6288	3595	5.175	7.424	43.5%	6.016	16.3%	6.591	27.4%	6.749	2.33%	6m19s	34M
c7552	2360	2.957	4.349	47.1%	3.327	12.5%	3.696	24.5%	3.806	2.89%	7m21s	24M
AVG				41.6%		14.0%		21.5%		2.5%		

Table 1. Experimental results for DC current constraints

ckt	delay (ns) random runs	delay (ns) NLP	%diff
c17	0.131	0.131	0.00%
c432	1.467	1.471	0.27%
c499	0.998	0.999	0.10%
c880	1.854	1.857	0.16%
c1355	1.667	1.671	0.24%
c1908	2.288	2.296	0.35%
c2670	1.832	1.848	0.87%
c3540	2.453	2.462	0.37%
c5315	2.379	2.379	0.00%
c6288	6.572	6.591	0.29%
c7552	3.693	3.696	0.08%
AVG			0.25%

Table 2. Random run comparison with NLP

benchmark circuits synthesized in 0.13 μ technology. A modified static timing analyzer was implemented in C++ and all experiments were run on a 1GHz SUN machine with 4GB of memory. MINOS [16] non-linear solver, which uses the augmented Lagrangian method, was used for optimization and AMPL [17] was used as the programming interface. The power grid for the circuits was constructed in metal layers M2-M8 using pitches and widths of an industrial microprocessor design. A PEEC-based extraction tool was used to extract the RLC parameters of the grid. Package inductance and resistance of 1nH and 1m Ω were attached in series with all the C4 bumps. In all experiments, on-die decap was distributed uniformly among the lowest metal layers across the whole area of the die. The power supply network was assumed to consist of 10 logic blocks and the minimum and maximum currents of the blocks were generated according to their area estimates. The current of each block was assumed to be distributed evenly among all its supply nodes. The gates for the circuits were connected to the supply nodes in the lowermost metal layer. In order to compute voltage sensitivities to block currents, the power grid was simulated in HSPICE with a fast current ramp (resembling a step waveform) applied at the supply nodes constituting each block and the response is numerically differentiated to obtain the impulse responses at all the nodes in the grid.

Table 1 shows the results for static power grid analysis where only DC current constraints are applied. Each block has a minimum and maximum constraint on its current. There is a constraint on the peak current consumption of the chip. Columns 1 and 2 show the circuits and the number of gates in their netlists. Column 3 shows the nominal delay of each circuit obtained from STA. Columns 4 and 5 show circuit delay with minimum supply margin (10%) for both V_{dd} and V_{ss} which is generally done in traditional STA tools to estimate supply noise induced delay increase. The 10% margin is the worst case voltage drop possible in our experiments when all block currents are allowed to switch within their bounds. Circuit delay increases on average by 41.6% for this worst-case voltage margin. Columns 6 and 7 show the case when the peak current is uniformly distributed among all the blocks, which causes an increase in circuit delay by 14% on an average. Columns 8 and 9 show the worst case delay obtained from NLP with constraints on block currents. On an average, the worst-case delay of the circuits increases by 21.52%. This shows that applying 10% voltage drop to all the gates in the circuit is very pessimistic because worst-case voltage drop is typically localized to a small region on the die and most of the die area may observe considerably better supplies. On the other hand, applying average

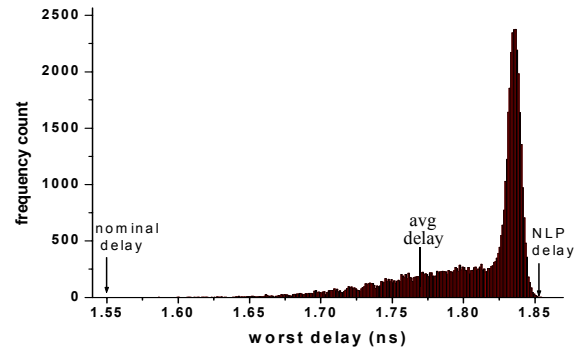


Figure 8. Random run result for circuit c880

block currents underestimates the worst-case delay because it does not allow the blocks with a higher sensitivity to circuit delay to switch with higher currents and may lead to an under-estimation of the impact of supply noise on delay. Column 10 presents the worst case delay of the circuit obtained from HSPICE when the circuit is simulated with voltage supplies obtained from NLP optimization. The approach has an average error of 2.5% (column 11) and a maximum error of 5.6% for the tested circuits. Column 12 and 13 show the run-time and peak memory usage respectively for each test circuit. The runtime includes the total runtime including the time for parsing in the netlist, leveling the circuit and iterative STA-NLP algorithm shown in Figure 6.

To test the quality of the NLP solution, 50,000 random runs were performed on all the test circuits. Currents for all the logic blocks were generated randomly so as to strictly meet the peak current consumption constraint:

$$\sum_{b=0}^{B-1} i_b = I_{peak} \quad (18)$$

Table 2 shows the comparison of worst-case delay obtained from NLP with the delay obtained from random simulations. The difference between the delays has a maximum error of 0.87%. Figure 8 shows the delay distribution curve obtained from 50,000 random runs of circuit c880. Every random run strictly met the peak current consumption constraint. Thus, each random run lies in the feasible region of the NLP. The arrows in Figure 8 show the nominal delay, delay obtained from average uniform block currents and the worst-case delay obtained from optimization. The worst-case NLP delay is

ckt	nom. delay (ns)	current ramped from zero to avg.		NLP		runtime
		delay(ns)	%incr.	delay(ns)	%incr.	
c17	0.109	0.129	18.35%	0.145	33.03%	0.32s
c432	1.224	1.441	17.73%	1.616	32.03%	7.54s
c499	0.824	0.973	18.08%	1.089	32.16%	44.21s
c880	1.550	1.823	17.61%	2.034	31.22%	41.05s
c1355	1.393	1.639	17.66%	1.826	31.08%	70.21s
c1908	1.912	2.268	18.62%	2.524	32.00%	40.95s
c2670	1.512	1.783	17.92%	2.010	32.94%	2m11s
c3540	2.063	2.422	17.40%	2.755	33.54%	5m08s
c5315	1.966	2.333	18.67%	2.627	33.62%	11m05s
c6288	5.175	6.225	20.29%	7.089	36.99%	33m51s
c7552	2.957	3.497	18.26%	3.932	32.97%	26m54s
AVG			18.24%		32.87%	

Table 3. Experimental results for AC (time-varying) current constraints

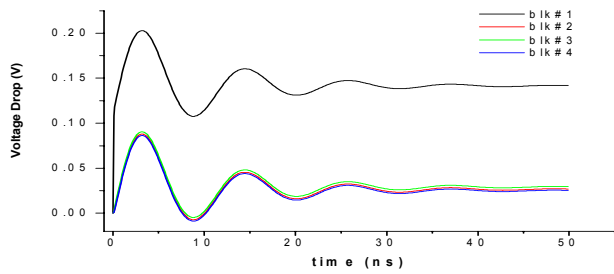


Figure 9. Step response at a node due to different blocks

always greater than the worst case delay obtained from random runs in all the circuits showing the effectiveness of the approach.

The proposed approach was also tested for dynamic case when the currents drawn by the logic blocks vary with time so as to maximize the worst-case circuit delay. In the dynamic scenario, the delay of a circuit in a given cycle depends not only on the values of currents in the current cycle, but also on the history of currents in the previous K cycles/time-steps where K is the total number of cycles/time-steps in the impulse response of the linear power grid system. Figure 9 shows the step response (obtained with very fast current ramps) at a node due to four different block currents. Assuming the operating frequency to be 1GHz, the step response is observed to reach the steady DC state after 50 cycles. In our experiments, the time step was taken to be one clock cycle and the delay of a circuit in a cycle depends on the activity of block currents in prior $K=50$ cycles. Table 3 shows the worst circuit delays, run-time and memory usage for the different benchmarks. Column 1 shows the circuit delay under nominal supply. Column 2 shows the increase in circuit delay when block currents are switched from zero to their respective average values within a cycle. In this case, the delay of the circuits is computed in the cycle of first voltage drop due to the applied current ramps. Column 3 shows the delay of the circuit and percentage increase in delay after NLP optimization. On an average, the delay of the circuits was observed to increase by 32.87%, demonstrating the need to incorporate the Ldi/dt drop while estimating the increase in circuit delay due to supply fluctuations. Columns 4 and 5 present the runtime and memory usage for the implementation. Figure 10 shows the total current obtained after NLP maximizes circuit delay at $t=50$ ns. It is interesting to observe that the total current of the chip oscillates between the minimum possible current to the maximum possible current with a frequency of 75MHz, which is exactly equal to the resonance frequency of the step response as shown in Figure 9. The total current gets distributed among the individual blocks based on the sensitivities of the supply nodes of the circuit and the dependence of circuit delay on the supplies at different nodes in the power grid.

Table 4 shows reduction in run-time due to circuit pruning. A worst-case voltage margin of 10% was assumed for pruning. Column 2 and 3 show the number of nets removed and percentage reduction in the pruned circuits. The run-time of the implementation (including pruning time) of the proposed approach on the pruned circuits is

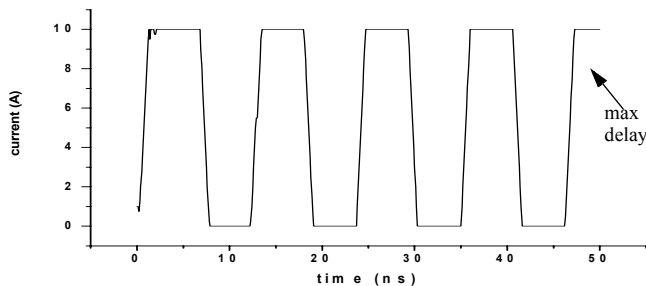


Figure 10. Current waveforms obtained from delay maximization

ckt	# gates initial	# inputs pruned	runtime initial	runtime pruned	runtime reduction
c17	7	2	0.32s	0.30s	6.25%
c432	212	123	7.54s	4.65s	38.33%
c499	553	234	44.21s	20.66s	53.27%
c880	568	287	41.05s	18.46s	55.03%
c1355	654	300	70.21s	30.08s	57.16%
c1908	543	260	40.95s	21.50s	47.50%
c2670	1043	453	2m11s	49.09s	62.53%
c3540	1492	723	5m08s	2m00s	61.04%
c5315	2002	1123	11m05s	4m52s	56.09%
c6288	3595	1048	33m51s	23m06s	31.76%
c7552	2360	1085	26m54s	13m44s	48.95%

Table 4. Runtime reduction due to pruning

shown in column 5. Column 6 shows the reduction in runtime for the test circuits, which is 47% on average. The computed worst-case delay was identical for the original and the pruned circuit.

5. Conclusion

In this paper, we have presented a new approach for computing the worst-case circuit delay due to power supply fluctuations. The analysis takes both IR and Ldi/dt drops into account. The proposed approach does not require enumeration of the critical paths of a given circuit and can be effectively incorporated in the existing STA framework. A delay based pruning technique is proposed in order to reduce the runtime of the implementation. The analysis has been implemented and tested on ISCAS85 benchmark circuits and is shown to accurately compute the worst possible delay under supply variations.

Acknowledgements

The authors would like to thank Vladimir Zolotov at T.J.W Research Center, IBM for helpful discussions. This work has been supported by funding from NSF, GSRC and Intel.

References

- [1] Semiconductor Industry Association. International Technology Roadmap for Semiconductors, 2004.
- [2] H. Qian, S. R. Nassif and S. S. Sapatnekar, "Random walks in a supply network," in *Proc. Design Automation Conference*, 2003.
- [3] J. Kozhaya, S. R. Nassif, and F. N. Najm, "A multigrid-like technique for power grid analysis," in *IEEE Trans. on Computer-Aided Design*, vol. 21, no. 10, pp. 1148-1160, October 2002.
- [4] S. R. Nassif and J. N. Kozhaya, "Fast power grid simulation," in *Proc. Design Automation Conference*, pp. 156-161, 2000.
- [5] M. Zhao, R. V. Panda, S. S. Sapatnekar and D. Blaauw, "Hierarchical analysis of power distribution networks," in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 159-168, 2002.
- [6] L. H. Chen, M. Sadowska and F. Brewer, "Coping with buffer delay change due to power and ground noise," *Proc. DAC*, 2002.
- [7] J.-J. Liou, A. Krstic, Y.-M. Jiang and K.-T. Cheng, "Path selection and pattern generation for dynamic timing analysis considering power supply noise effects," in *Proc. ICCAD*, 2000.
- [8] G. Bai, S. Bobba and I. N. Hajj, "Static timing analysis including power supply noise effect on propagation delay in VLSI circuits," in *Proc. DAC*, 2001.
- [9] D. Kouroussis, R. Ahmadi and F. N. Najm, "Worst-case circuit delay taking into account power supply variations," in *Proc. DAC*, 2004.
- [10] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, R. Panda, "Vectorless analysis of supply noise induced delay variation," in *Proc. ICCAD*, 2003.
- [11] R. Ahmadi and F. N. Najm, "Timing analysis in presence of power supply and ground voltage variations," in *Proc. ICCAD'03*.
- [12] A. Chandrakasan, W. J. Bowhill and F. Fox, Design of high performance microprocessor circuits. NY: IEEE Press, 2001.
- [13] Chung-Ping Chen, C. C. N. Chu, and D. F. Wong, "Fast and Exact Simultaneous Gate and Wire Sizing by Lagrangian Relaxation", in *IEEE Trans on CAD*, vol. 18, no. 7, July 1999.
- [14] D. Kouroussis and F. N. Najm, "A static pattern-independent approach for power grid voltage integrity verification," in *Proc. DAC*, 2003.
- [15] H. Qian, S. R. Nassif and S. S. Sapatnekar, "Early-stage Power Grid Analysis for Uncertain Working Modes", in *Proc. ISPD'04*.
- [16] Stanford Business Software Inc. www.sbsi-sol-optimize.com.
- [17] AMPL: A modeling language for mathematical programming. <http://www.ampl.com>.

