

# Vector Generation for Maximum Instantaneous Current Through Supply Lines for CMOS Circuits \*

Angela Krstić and Kwang-Ting (Tim) Cheng

*Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106*

## Abstract

We present two new algorithms for generating a small set of patterns for estimating the maximum instantaneous current through the power supply lines for CMOS circuits. The first algorithm is based on timed ATPG, while the second is a probability-based approach. Both algorithms can handle circuits with arbitrary but known delays and they produce a set of 2-vector tests. Experimental results demonstrating that the outcome of applying our algorithms is a small set of patterns producing a current that is a tight lower bound on the maximum instantaneous current are included.

## 1 INTRODUCTION

Continuous shrinking of the device feature sizes introduces many new problems in VLSI design. Large voltage drops caused by high instantaneous current flowing through the power supply lines can affect the reliability as well as the performance of the circuit. High current density is a cause of electromigration which can lead to short or open circuits. To design the power supply lines properly, it is necessary to estimate the maximum instantaneous current flowing through them. However, this estimation is not an easy task for several reasons. First, the maximum instantaneous current through the power supply lines depends on the inputs that are applied to the circuit. Current through the supply lines of CMOS circuits is mainly due to the switching on the signals. To be able to observe switching on the signals, a two pattern test has to be applied at the circuit inputs. Second, the maximum instantaneous current is very dependent on the circuit delays because the maximum current depends on the number of signals that are switching simultaneously or within a small time interval. Therefore, the circuit timing should be considered during the maximum current estimation process.

Several research groups have worked on estimating the power or maximum instantaneous current [1, 3, 4, 6, 7]. Methodologies proposed in [1, 3] are applicable only to small circuits. Kriplani *et al.* [4] have presented a pattern independent algorithm to find an upper bound on the maximum instantaneous current through the power

supply lines of CMOS circuits. Because of their assumption that all signals are independent, the estimated maximum current for most circuits represents a loose upper bound. In [6, 7] a test generation strategy was devised for finding test patterns that would produce the maximum power. The estimated maximum power represented a lower bound. The main problem in this methodology was the use of the zero-delay model for the circuit. It is known that the glitches can significantly contribute to the maximum power and they cannot be considered when a zero-delay model is used.

In this paper, we approach the problem of estimating the maximum instantaneous current through the supply lines of CMOS circuits through automatic test pattern generation (ATPG). Our methodology produces a lower bound on the maximum current and it handles circuits with arbitrary delays. We present two different algorithms for test generation: one based on *timed ATPG* [2] and the other using a probabilistic approach. In general, the timed ATPG approach produces a tighter lower bound on the maximum current at the expense of memory and computation time. The probabilistic approach is more efficient but it produces a looser lower bound on the maximum current. We simulate the produced test patterns using a commercially available event-driven transistor-level power/current simulator [5] and compare the results to the results obtained for a set of randomly generated test patterns.

## 2 PRIOR WORK

We rely on the *iMax algorithm* [4] in the first step of our methodology for estimating the maximum instantaneous current through the power supply lines. It assumes that all inputs to the combinational logic switch simultaneously at time  $t=0$  and that the delays of the gates can take any arbitrary values and are known for each gate. The current drawn from the supply lines during switching of a signal is assumed to be of a triangular form as shown in Fig. 1. The peak current is assumed to coincide with the transitions at the input of the gate. All above assumptions are kept the same in our work.

At any point in time a signal is assumed to have one out of four possible excitations: stable 1 value ( $S1$ ), stable 0 value ( $S0$ ), rising transition ( $r$ ) or falling transition ( $f$ ) [4]. To find an upper bound on the maximum instantaneous current by simultaneously considering all possible 2-vector input patterns, the excitations on the signals are represented using *uncertainty waveforms* [4]. An uncertainty waveform  $U(t)$  captures all possible excitations that a signal can have under any 2-input vector applied to the primary inputs. The following example

\* This work was supported by EPIC, Inc., by California MICRO and by NSF under Grant MIP-9409174.

### Design Automation Conference ©

Permission to make digital or hard copy of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
DAC 97, Anaheim, California  
(c)1997 ACM 0-89791-920-3/97/06 ..\$3.50

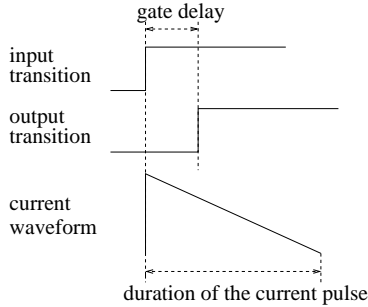
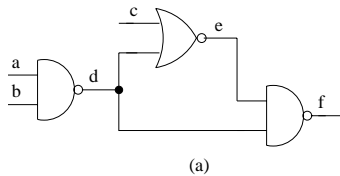


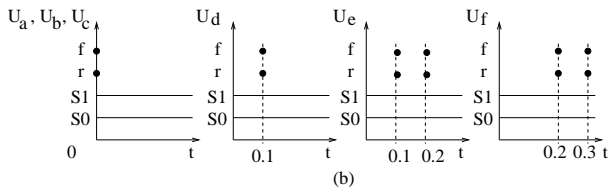
Figure 1: Current model.

illustrates the *iMax* procedure.

**Example 1** Consider the circuit in Fig. 2(a). The rising



(a)



(b)

Figure 2: An example.

and falling delays of all gates are assumed to be 0.1ns. It is assumed that at  $t=0$  any input signal can have any of the four possible excitations. Fig. 2(b) shows the uncertainty waveforms for all signals in this circuit. The transient current is assumed to be triangular with peak value of 3mA and the duration of the current pulse is assumed to be 0.3ns. From the uncertainty waveforms in Fig. 2(b) we can see that at time  $t=0$  only the inputs of gates  $d$  and  $e$  may be switching. Therefore, the total current at  $t=0$  is  $I_{tot}(0)=I_d+I_e=3+3=6\text{mA}$ . Next, to find the total current at time  $t=0.1\text{ns}$  we note that since the inputs to gate  $d$  can only switch at time  $t=0$  the current contribution of gate  $d$  at time  $t=0.1\text{ns}$  is 2mA. The inputs to gate  $e$  can switch either at time  $t=0$  or at time  $t=0.1\text{ns}$  and the maximum current contribution of gate  $e$  at time  $t=0.1\text{ns}$  is  $I_e=\max(2, 3)\text{mA}$ . Also, at time  $t=0.1\text{ns}$  the maximum current contribution of gate  $f$  is 3mA. Therefore, the total current at time  $t=0.1\text{ns}$  is  $I_{tot}(0.1)=I_d+I_e+I_f=2+3+3=8\text{mA}$ . The current contributions at other time points can be found in a similar way:  $I_{tot}(0.2)=I_d+I_e+I_f=1+2+3=6\text{mA}$ ,  $I_{tot}(0.3)=I_e+I_f=1+2=3\text{mA}$ ,  $I_{tot}(0.4)=I_f=1\text{mA}$ ,  $I_{tot}(0.5)=0$ . From the above discussion, the maximum current is found to be 8mA at time 0.1ns.

### 3 MAXIMUM INSTANTANEOUS CURRENT IN POWER/GROUND LINES

Our approach to estimating maximum instantaneous current is through test generation. Our goal is to find a small set of two-vector patterns which would produce high instantaneous current. Since the *iMax* algorithm [4]

can predict the upper bound on the value of the current through supply lines at any given time, we use this procedure as a starting point for our algorithm. The idea is to use the current waveform produced by *iMax* to find a set of time instances when the current is most likely to have a high value. Fig. 3 shows a possible current waveform produced by *iMax* algorithm. Because the signal correlations have been ignored in the *iMax* algo-

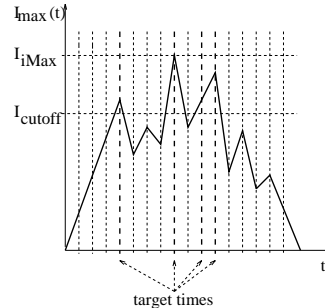


Figure 3: Finding the target time and target gates.

rithm, the predicted maximum current ( $I_{iMax}$ ) might be much higher than the actual one. Our assumption is that the value of the actual maximum instantaneous current is somewhere between the values  $I_{cutoff}$  and  $I_{iMax}$ . The value of  $I_{cutoff}$  represents some percentage of  $I_{iMax}$ . The time instances for which the value of the current is in the interval  $[I_{cutoff}, I_{iMax}]$  we call *target times*. Corresponding to each time instance  $t$  is a set of gates such that their simultaneous switching would cause current  $I_{max}(t)$  to flow through the power supply lines. The set of gates that corresponds to a target time  $T$  is called *set of target gates*  $G$ . According to the assumed current waveform (Fig. 1) the output of a target gate  $g \in G$  is required to switch at time  $T + t_r(g)$  or  $T + t_f(g)$ , where  $t_r(g)$  and  $t_f(g)$  represent the rising and falling delay of gate  $g$ , respectively. If a 2-input test can be found such that all target gates contribute to a transient current at the target time, the maximum instantaneous current would be equal to the current predicted by the *iMax* algorithm. However, often such 2-input vector cannot be found. In our algorithm, for each pair  $(T, G)$  we try to find a two-vector test that maximizes the current produced by switching of the outputs of the target gates. We propose two methods for generating the two-vector test. The first method is based on the *timed ATPG* technique [2]. The second is a probabilistic approach.

#### 3.1 TIMED ATPG BASED APPROACH

**Timed ATPG.** Timed ATPG is ATPG with an additional dimension: time. Timed ATPG was first proposed in [2] for timing analysis. In timed ATPG each signal is characterized by its logic value and the time interval in which this logic value should occur. Therefore, in timed ATPG conflicts on the signals can be twofold: logic or timing conflicts. Logic conflicts occur when a signal is required to simultaneously have two different logic values. Timing conflicts occur when it is required that a signal be assigned the given logic value outside the required time interval. The timed ATPG proposed in [2]

was able to generate a one-vector test. Here, we extend the timed ATPG concept to handle a vector pair. Our algorithm consists of several steps.

**Step 1.** Given the target time  $T$  and its corresponding set of target gates  $G$ , we try to assign transitions to as many target gates as possible. We process the gates in the target set one at a time. The order for processing the target gates depends on the value of the gate’s current contribution, i.e., the value of the current produced when the output of the target gate switches. The current contribution of a gate is a function of the load of the gate (sum of the input capacitances of its fanouts), the gate type (NAND, NOR, etc.), the type of the transition (rising, falling), the number of inputs that are switching. Also, since we assume that the current waveform is triangular, the current contribution of a gate depends on the time. To estimate the value of the peak current and the value of the duration of the current pulse as a function of the above variables, we use a transistor-level simulator [5] to characterize the library cells and create lookup tables for different cells, with different number of inputs.

To decide which type of transition will be assigned to a target gate at the target time, we use the signal uncertainty waveforms derived in the *iMax* algorithm. For example, if at the target time the uncertainty waveform shows that the given signal can only have a rising (falling) transition, we assign a rising (falling) transition to the gate. However, in some cases the uncertainty waveform might indicate that the signal can be assigned either a rising or a falling transition at the target time. For such gates we pick the transition that produces a higher current at the target time.

**Step 2.** After the order of gates for processing has been decided, we assign the required transition to the target gate on the top of the list. Using the information about the target time  $T$ , gate delays ( $t_r$  and  $t_f$ ) and the uncertainty waveforms derived by *iMax* we try to sensitize a path from the given gate  $g$  to a PI. The sensitized path has to be such that the required time for the transition at the given target gate is  $T+t_r(g)$  or  $T+t_f(g)$  (depending on the transition type) and the required time at the primary input is 0. In this process, we use only mandatory assignments and their implications for the on- and off-inputs of the path, and we keep updating the uncertainty waveforms using these assignments.

If there is a conflict in this phase we backtrack to sensitize another path to a PI. The number of paths that can be sensitized using the general delay model usually is not very large. This is because a necessary condition for a path to be sensitized is that the primary input at the source of the sensitized path has to be applied at time  $t=0$ . If for some gate no path to a primary input can be sensitized, we leave the output of this gate unassigned and proceed with the next gate in the list. During the path sensitization phase, we keep track of the gates that require justification of the values in their uncertainty waveforms, i.e., we create a justification list.

**Step 3.** After all gates in the target set have been pro-

cessed, we check if the justification list is empty. Some signals might be in the justification list more than once since it might be necessary to justify their value in more than one time interval. If all signals are successfully justified, there could still be some PIs with unspecified values. From the derived set of uncertainty waveforms at the PIs, we obtain the set of excitations that are possible for each such PI and we randomly assign one of them. On the other hand, if it is impossible to justify all the signals in the justification list for a given target time, we backtrack to the last decision in the path sensitization phase and try to sensitize a different path. If a different path can be sensitized, the justification procedure is again attempted. The procedure ends when either all gates are justified or when all possibilities for sensitizing the paths have been explored. The whole procedure is then repeated for the next  $(T, G)$  pair.

**Example 2** Consider again the circuit in Fig. 2(a). Let us assume that  $I_{cutoff}=0.5I_{iMax}=4mA$ . From Example 1 we get that the target times are 0.1ns, 0.2ns and 0ns. Since the current is the highest at time  $t=0.1ns$  we process this target time first. For  $t=0.1ns$ , the set of target gates contains gates  $f$ ,  $e$  and  $d$ . Their current contributions at the target time are 3mA, 3mA and 2mA, respectively. Either gate  $f$  or gate  $e$  can be processed first. We pick gate  $f$  since it is further from the primary inputs than gate  $e$ . From the uncertainty waveforms in Fig. 2(b) we see that the current contribution of gate  $f$  at the target time could be due to either a rising or a falling transition at  $t=0.2ns$ . In our example, the currents due to the falling or rising transition are the same and we randomly assign a falling transition to signal  $f$ . Next, we have to sensitize a path from  $f$  to some primary input. The only two paths that can satisfy the timing requirements for a falling transition at  $t=0.2ns$  at  $f$  are paths  $\{adf, falling\}$  and  $\{bdf, falling\}$ . Let the chosen path be  $\{adf, falling\}$ . The sensitized path and the requirements on the path on-inputs are shown in Fig. 4(a). Updating uncertainty waveforms:

1. Since the on-input  $d$  must have a rising transition at time  $t=0.1ns$ , the only possible excitations at the off-input  $e$  are S1, rising transition at  $t=0.1ns$  and falling transition at  $t=0.2ns$ . Therefore, the uncertainty waveform at  $e$  is updated and the new uncertainty waveform is implied across gate  $f$  (Fig. 4(a)).
2. In order to have a rising transition at signal  $d$  at  $t=0.1ns$  when input  $a$  has a falling transition at  $t=0$ , signal  $b$  can only be assigned a falling transition or an S1 value. The new uncertainty waveforms for signals  $a$  and  $b$  are shown in Fig. 4(b).
3. A rising transition at  $t=0.1ns$  at  $d$  combined with any excitation at the primary input  $c$  cannot produce a rising transition at  $t=0.1ns$  at the output  $e$ . Therefore, the uncertainty waveform of signal  $e$  has to be further updated and it is shown in Fig. 4(b).

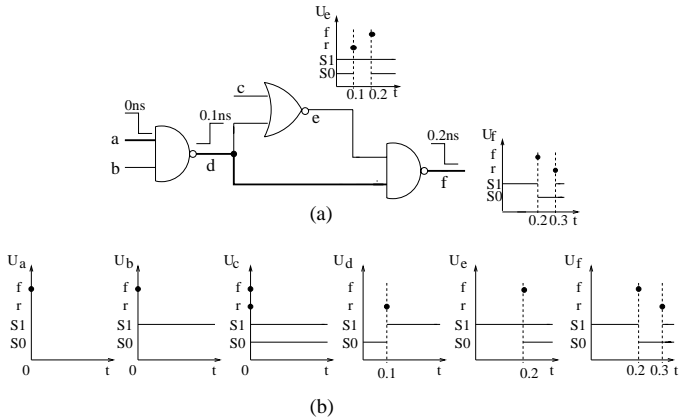


Figure 4: The updated uncertainty waveforms.

After repeating this process for gates  $e$  and  $d$  we get that, for example, assigning a falling transitions to inputs  $a$  and  $b$  and a rising transition to input  $c$  results in the maximum current of 8mA at time 0.1ns.

### 3.2 PROBABILITY BASED APPROACH

Timed ATPG approach can be computationally expensive for large designs. In this subsection we describe our probability based approach to generate test vectors for high instantaneous current given a pair  $(T, G)$ . This method is more practical for large designs than the timed ATPG at the price of a looser lower bound on the maximum instantaneous current.

In this approach, the idea is to derive good weights of switching at the PIs for generating weighted random vector pairs for maximum current. Our algorithm starts from the set of target gates  $G$  and it numerically characterizes each of the four possible excitations at the output of each gate  $g \in G$ . Next, these values are backward propagated to the PIs. Once the PIs are reached, the derived numeric values are used as weights for generating a small set of 2-vector tests which have a high probability to generate a high current at the target time. To be able to handle arbitrary circuit delays, the assigned numeric values have to be associated with time. In the following we describe the details of our probability based approach.

Each gate in the target set is assigned four *excitation lists*:  $L_0$ ,  $L_1$ ,  $L_r$  and  $L_f$ . Each excitation list contains pairs of type  $(w, t)$ . Value  $w$  represents a numerical measure characterizing the preference for the gate to have the given type of excitation at the time  $t$ . For example, if for a gate  $g$ , list  $L_1(g)$ , contains a pair  $(0, t)$  it means that at time  $t$  it is not desirable for  $g$  to have a stable 1 value. A higher value for  $w$  denotes a higher preference for the signal to have the given excitation at time  $t$ .

**Step 1.** Since our goal is to have as many gates as possible switching at the target time  $T$ , each target gate  $g \in G$  is initially assigned the following values:  $L_0(g)=L_1(g)=\{(0, T+t_f(g)), (0, T+t_r(g))\}$ ,  $L_f(g)=\{(w_f, T+t_f(g))\}$  and  $L_r(g)=\{(w_r, T+t_r(g))\}$ . For each target gate  $g$ , the value  $w_f$  ( $w_r$ ) represents the current contribution caused by the falling (rising) transition at the output of  $g$  at time  $T+t_f(g)$  ( $T+t_r(g)$ ).

**Step 2.** After initializing all the excitation lists at the

outputs of target gates, we propagate backward these lists to the PIs. The gates are processed in a topological order starting from the target gates towards the PIs. For each gate, the lists for all four excitations are propagated from the output to each of its inputs by backward propagating each  $(w, t)$  pair in the output list according to the rules explained below.

In the following, the superscript  $i$  is used to denote the values at the gate input, while the superscript  $o$  denotes the values at the gate output. Therefore, pair  $(w^i, t^i)$  is associated with input while pair  $(w^o, t^o)$  is associated with the output of a gate. Given an output pair  $(w^o, t^o)$ , the time component  $t^i$  of each input pair is found using the information about the falling or rising delay of the gate, i.e.,  $t^i = t^o - t_f$  or  $t^i = t^o - t_r$ , depending on the excitation list and gate type. The numerical component  $w^i$  of an input pair is a function of the gate type and the number of inputs to the gate. Finding the value of  $w^i$  for each pair of an excitation list when the values of  $w^o$  for all the excitation lists are given is explained next.

In general, an excitation at the input satisfies the following equation:

$$w_k^i = A_k \times w_f^o + B_k \times w_r^o + C_k \times w_0^o + D_k \times w_1^o$$

where  $k \in \{r, f, 0, 1\}$ . Coefficients  $A_k$ ,  $B_k$ ,  $C_k$  and  $D_k$  for each of the excitations can be found as a function of the gate type and the number of inputs to the gate. As an example, we illustrate finding of these coefficients for  $w_r^i$  for a 2-input NAND gate. The coefficients for other excitations and other gate types can be found in a similar way. Table 1 shows the truth table for a 2-input NAND gate. The first three rows in the left half of the table show all possible input excitations that result in a falling transition at the output of the NAND gate, the next three rows show that for the rising transition, etc. When the output of a 2-input NAND gate has a falling

out	in1	in2	out	in1	in2
f	S1	r	S1	S0	f
	r	S1		S0	r
	r	r		f	S0
r	S1	f		r	S0
	f	S1		S0	S0
	f	f		S1	S0
S0	S1	S1	S0	S1	
			f	r	
			r	f	

Table 1: Truth table for a 2-input NAND gate. transition there are two cases in which an input can have a rising transition and one case in which the same input has an S1 value. Therefore, the coefficient  $A_r$  can be found as  $A_r = \frac{2}{1+2}$ . Because we are interested in finding the maximum current, we prefer to have transitions at the internal signals instead of stable values. Let  $p$  denote a *weight* of a transition with respect to a stable value. In other words, if  $p=1$ , a transition and a stable value are weighted equal, if  $p=2$ , a transition is twice as valuable as a stable value, etc. Then, the expression for  $A_r$  can be written as  $A_r = \frac{2p}{1+2p}$ . If the output of a NAND gate has a rising transition or an S0 value, from Table 1 we see that no input can have a rising transition. Therefore, we have

$B_r=0$  and  $C_r=0$ . When the output of a NAND gate has an S1 value, there are two possibilities for an input to have a rising transition. Also, in this case the input can have a transition in a total of 4 cases. Therefore, using the transition weight  $p$ , we get  $D_r=\frac{2p}{5+4p}$ . Table 2 shows all the coefficients for a 2-input NAND gate.

	$w_f^o$	$w_r^o$	$w_0^o$	$w_1^o$
$w_r^{(i)}$	$A_r = \frac{2p}{1+2p}$	$B_r = 0$	$C_r = 0$	$D_r = \frac{2p}{5+4p}$
$w_f^{(i)}$	$A_f = 0$	$B_f = \frac{2p}{1+2p}$	$C_f = 0$	$D_f = \frac{2p}{5+4p}$
$w_0^{(i)}$	$A_0 = 0$	$B_0 = 0$	$C_0 = 0$	$D_0 = \frac{4}{5+4p}$
$w_1^{(i)}$	$A_1 = \frac{1}{1+2p}$	$B_1 = \frac{1}{1+2p}$	$C_1 = 1$	$D_1 = \frac{1}{5+4p}$

Table 2: Backward propagation for 2-input NAND gate.

The excitation lists for fanout stems are found by combining the excitation lists of the fanout branches. Combining of the excitation lists means that the pairs  $(w, t)$  with the matching time  $t$  are combined by summing up the values of  $w$  for each pair, and then the lists with different time components are concatenated together.

**Step 3.** Once the primary inputs are reached the derived numerical values can be used as weights to generate weighted random tests for estimating the maximum instantaneous current.

**Example 3** Consider again the circuit in Fig. 2(a). From Example 1, the maximum current occurs for target time  $T=0.1ns$  and the target gates are  $f$ ,  $e$  and  $d$ . For each target gate we initialize the excitation lists using the information about the current contribution of the gate at the target time:

gates  $f$  and  $e$ :  $L_r=L_f=\{(3, 0.2)\}$ ,  $L_0=L_1=\{(0, 0.2)\}$ ,  
gate  $d$ :  $L_r=L_f=\{(2, 0.2)\}$ ,  $L_0=L_1=\{(0, 0.2)\}$ .

Next, we need to backward propagate these excitation lists. We illustrate this step for gate  $f$ . The procedure for gates  $e$  and  $d$  is similar. Using the rules from Table 2 for gate  $f$ , we get:

$$w_r^e = w_r^d = \frac{2p}{1+2p}w_f^f + \frac{2p}{5+4p}w_1^f = \frac{6p}{1+2p}$$

$$w_f^e = w_f^d = \frac{2p}{1+2p}w_r^f + \frac{2p}{5+4p}w_1^f = \frac{6p}{1+2p}$$

$$w_0^e = w_0^d = \frac{4}{5+4p}w_1^f = 0$$

$$w_1^e = w_1^d = \frac{1}{1+2p}(w_f^f + w_r^f) + w_0^f + \frac{1}{5+4p}w_1^f = 0$$

Similar to what was said for the fanout stems, for each gate, the pairs with matching time component are combined by adding up the numerical components while the pairs with different time components are just concatenated. After backward propagation for all gates in the circuit, at the primary inputs we get:

$$\text{input } c: L_r=L_f=\left\{\left(\frac{6p}{1+2p}, 0.1\right), \left(\frac{12p(4p^2+6p+1)}{(5+4p)(1+2p)^2}, 0\right)\right\}$$

$$L_0=\left\{\left(\frac{6}{1+2p}, 0.1\right), \left(\frac{24}{(5+4p)(1+2p)}, 0\right)\right\}$$

$$L_1=\left\{\left(0, 0.1\right), \left(\frac{6(8p^2+12p+1)}{(5+4p)(1+2p)^2}, 0\right)\right\}$$

inputs  $a$  and  $b$ :

$$L_r=L_f=\left\{\left(\frac{4p}{1+2p}, 0.1\right), \left(\frac{12p(4p^2+7p+1)}{(5+4p)(1+2p)^2}, 0\right), (s_1, -0.1)\right\}$$

$$L_0=\left\{\left(0, 0.1\right), \left(\frac{24}{(5+4p)(1+2p)}, 0\right), (s_3, -0.1)\right\}$$

$$L_1=\left\{\left(\frac{4}{1+2p}, 0.1\right), \left(\frac{36(4p^2+6p+1)}{(5+4p)(1+2p)^2}, 0\right), (s_4, -0.1)\right\}$$

where  $s_1$ ,  $s_2$ ,  $s_3$  and  $s_4$  are some complicated functions of  $p$  which are irrelevant for our discussion.

From the above expressions we see that all primary inputs have excitations that contain pairs of type  $(w, 0)$ . For example, for  $p=2$  from these pairs we get:

gate  $c$ :  $w_r = w_f = 2.28, w_0 = 0.36, w_1 = 1.05$

gates  $a$  and  $b$ :  $w_r = w_f = 4.20, w_0 = 0.36, w_1 = 3.21$

These values can next be used to decide which excitations to assign to the PIs. For all inputs the maximum values appear for falling/rising transition and for example assigning falling transitions to  $a$  and  $b$  and a rising transition to  $c$  produces the maximum current of 8mA.

Ckt	<i>iMax</i>		timed_ATPG		random		
	(mA)	nor.	pttn	(mA)	nor.	(mA)	nor.
b1	25.6	1.37	3	18.5	1	16.1	0.87
b9	152.2	1.55	8	98.2	1	81.9	0.83
c8	196.2	1.28	8	152.6	1	114.4	0.75
cc	74.1	1.69	7	43.7	1	39.3	0.90
cm150a	112.8	1.36	10	82.8	1	67.2	0.81
cm163a	49.7	1.55	10	31.9	1	19.7	0.61
cm42a	29.5	1.69	11	17.4	1	9.24	0.53
cm85a	43.8	1.21	8	36.0	1	31.6	0.88
cmb	57.3	1.05	7	54.4	1	22.4	0.41
majority	16.6	1.15	7	14.5	1	13.5	0.93
mux	92.0	1.17	7	78.7	1	78.5	0.99
parity	62.5	1.17	10	53.5	1	28.7	0.53
pcler8	109.3	1.75	9	62.2	1	52.5	0.83
average		1.38			1		0.75

Table 3: Results for timed ATPG approach.

The bottleneck of the probability based approach is the *iMax* algorithm used in the first step. The excitation lists can be backward propagated in linear time which makes this approach efficient for larger designs.

#### 4 EXPERIMENTAL RESULTS

We have implemented the two proposed algorithms and tested them on a set of MCNC combinational benchmark circuits. In our experiments, the value of  $I_{cutoff}$  was set as 50% of the maximum current predicted by *iMax* algorithm. For both methodologies we generate a small set of patterns and we simulate them using PowerMill [5]. We also generate a set of 500 random patterns and simulate them using the same tool. We generate the random patterns such that 80% of the inputs are assigned transitions [6]. This is because simulating patterns in which each input has a transition does not have to necessarily produce the maximum current (especially true for circuits with XORs). We compare the maximum instantaneous current reported by PowerMill for these two sets.

Table 3 shows the results for the timed ATPG methodology. Column 2 shows an upper bound on the maximum current as predicted by *iMax* algorithm. The third column shows the normalized value of the current with respect to the value produced by timed ATPG. Column 4 shows the number of generated patterns by the timed ATPG algorithm. The results of simulating this set of patterns are shown in column 5. Column 7 shows the results for simulating the random set of 500 patterns and the last column shows the values of the current normalized with respect to the values produced by timed ATPG. As it can be seen, our methodology produces a small set of patterns which gives on the average a 25%

Ckt	<i>iMax</i>		probab. appr.			random	
	(mA)	nor.	pttn	(mA)	nor.	(mA)	nor.
9symml	271.5	2.97	13	91.2	1	85.8	0.94
C432	217.6	1.51	62	143.9	1	124.7	0.86
C499	382.1	2.19	35	174.3	1	122.6	0.70
apex6	779.5	2.14	13	364.5	1	324.2	0.89
apex7	279.1	2.23	14	125.0	1	109.9	0.88
b1	25.6	1.38	3	18.5	1	16.1	0.87
b9	152.2	1.79	8	84.8	1	81.9	0.96
c8	196.2	1.40	8	139.4	1	114.4	0.82
cc	74.1	1.60	7	46.3	1	39.3	0.85
cht	258.2	1.26	8	204.3	1	162.9	0.79
cm150a	112.8	1.63	10	69.3	1	67.2	0.97
cm163a	49.7	1.92	10	25.8	1	19.7	0.76
cm42a	29.5	1.77	11	16.6	1	9.4	0.56
cm85a	43.8	1.27	8	34.4	1	28.9	0.84
cmb	57.3	1.24	7	46.0	1	22.4	0.48
comp	161.2	1.38	11	116.9	1	59.2	0.50
cordic	123.9	1.27	8	97.3	1	72.8	0.75
f51m	177.2	2.03	12	87.3	1	54.9	0.63
frg2	1216.4	2.35	22	517.4	1	420.4	0.81
majority	16.6	1.24	7	13.4	1	13.5	1.01
mux	92.0	1.22	7	75.4	1	78.5	1.04
my_adder	235.3	1.47	10	160.4	1	117.2	0.73
parity	62.5	1.17	10	53.5	1	28.7	0.53
pc1er8	109.3	1.87	9	58.2	1	44.7	0.77
pm1	56.5	1.87	7	30.1	1	20.7	0.69
term1	423.4	1.71	17	246.0	1	211.8	0.86
tt2	274.7	1.86	10	146.4	1	138.2	0.94
unreg	170.7	1.16	11	146.4	1	127.5	0.87
x2	65.1	1.64	8	39.7	1	34.7	0.87
x3	1036.4	2.12	10	487.0	1	384.6	0.79
x4	521.3	2.31	14	225.8	1	204.5	0.90
z4ml	92.4	2.19	10	42.0	1	36.5	0.87
average		1.72			1		0.79

Table 4: Results for the probabilistic approach.

tighter lower bound on the maximum current than the bound obtained with a larger set of random patterns.

Table 4 shows experimental results for the probability based approach with  $p=2$ . Column 4 shows the number of patterns generated for each circuit. For each set of weights at the PIs we generate one 2-vector pattern. Our experiments have shown that generating a larger set of weighted random patterns does not lead to a significant improvement in the results. In Table 4 the values of the current are normalized with respect to the values produced by our probability based approach. On the average the patterns generated by the probability based approach produce a 21% tighter bound on the maximum current than the much larger set of random patterns.

Table 5 shows the comparison between the two proposed methodologies. Column 2 shows the maximum current produced by simulating the patterns generated by timed ATPG, by probabilistic approach or by a random set of 500 patterns. The values in the table are normalized with respect to this best known lower bound. The timed ATPG for this set of circuits produces on the average a 24% tighter lower bound than the random set of patterns. The improvement of the probabilistic method over the random set is 16% on the average.

## 5 CONCLUSIONS

We approach the problem of estimating the maximum

Ckt	best known lower bound		timed ATPG		probab. appr.		random	
	(mA)	nor.	(mA)	nor.	(mA)	nor.	(mA)	nor.
b1	18.5	1	18.5	1	18.5	1	16.1	0.87
b9	98.2	1	98.2	1	84.8	0.86	81.9	0.83
c8	152.6	1	152.6	1	139.4	0.91	114.4	0.75
cc	46.3	1	43.7	0.94	46.31	1	39.3	0.85
cm150a	82.8	1	82.80	1	69.3	0.83	67.2	0.81
cm163a	31.9	1	31.9	1	25.8	0.81	19.7	0.61
cm42a	17.4	1	17.4	1	16.8	0.95	9.24	0.53
cm85a	36.0	1	36.0	1	34.4	0.95	31.6	0.88
cmb	54.4	1	54.4	1	46.0	0.84	22.4	0.41
majority	14.5	1	14.5	1	13.4	0.92	13.5	0.93
mux	78.7	1	78.7	1	75.4	0.96	78.5	0.99
parity	53.5	1	53.5	1	53.5	1	28.7	0.53
pc1er8	62.2	1	62.2	1	58.2	0.93	52.5	0.83
average		1		0.99		0.91		0.75

Table 5: Timed ATPG vs. probabilistic approach. instantaneous current through the power supply lines of CMOS circuits through test generation. We propose two algorithms (one based on timed ATPG and another probabilistic approach) for generating a small set of test patterns that would produce a high instantaneous current. In general, the timed ATPG methodology produces a tighter lower bound on the maximum current than the probabilistic approach but it also has higher computational requirements. Our experimental results show that in comparison with a random set of patterns, the small sets of patterns generated using our methodologies result in a tighter lower bound on the maximum current. We are currently investigating the use of genetic algorithms for generating test patterns for maximum instantaneous current.

## References

- [1] S. Chowdhury and J. S. Barkatullah. Estimation of Maximum Currents in MOS IC Logic Circuits. *IEEE Transactions on CAD*, 9(6):642–654, June 1990.
- [2] S. Devadas, K. Keutzer, and S. Malik. Computation of Floating Mode Delay in Combinational Circuits: Theory and Algorithms. *IEEE Transactions on CAD*, 12(12):1913–1923, December 1993.
- [3] S. Devadas, K. Keutzer, and J. White. Estimation of Power Dissipation in CMOS Combinational Circuits Using Boolean Function Manipulation. *IEEE Transactions on CAD*, 11(3):373–383, March 1992.
- [4] H. Kriplani, F. N. Najm, and I. N. Hajj. Pattern Independent Maximum Current Estimation in Power and Ground Buses of CMOS VLSI Circuits: Algorithms, Signal Correlations, and Their Resolution. *IEEE Transactions on CAD*, 14(8):998–1012, August 1995.
- [5] EPIC Design Technology. PowerMill Reference Manual. August 1992.
- [6] C.-Y. Wang and K. Roy. Maximum Power Estimation for CMOS Circuits Using Deterministic and Statistic Approaches. *Proceedings of 9th International Conference on VLSI Design*, pages 364–369, January 1995.
- [7] C.-Y. Wang, K. Roy, and T.-L. Chou. Maximum Power Estimation for Sequential Circuits Using a Test Generation Based Technique. *Proceedings of IEEE Custom Integrated Circuits Conference*, pages 229–232, April 1996.