ATPG Power Guards: On Limiting the Test Power below Threshold

Rohini Gulve and Virendra Singh
Computer Architecture and Dependable Systems Lab,
Electrical Engineering, Indian Institute of Technology Bombay, India.
Email: rohini.gulve@iitb.ac.in, viren@ee.iitb.ac.in

***Abstract*—Modern circuits with high performance and low power requirements impose strict constraints on manufacturing test generation, particularly on timing test. Delay test is used for performance grading of the circuit. During the application of the test, power consumption has to be less than the functional threshold value, in order to avoid yield loss. This work proposes a new direction to generate power safe test without any changes in DFT (design for testability) structure or existing CAD (computer-aided design) tools. We propose a virtual wrapper circuitry around the circuit under test (CUT), for test generation purpose, which acts as a shield to obtain power safe vectors. The wrapper prohibits the generation of test vector if power consumption exceeds the threshold limits. We consider analytical power models for power analysis of candidate test vector patterns. Experiments performed on benchmark circuits show power safe test generation without coverage loss.**

I. Introduction

Excessive power consumption during manufacturing test application is one of the severe problems in high-performance circuits. Test generation based on only structural information may lead to the application of test inputs which may not correspond to any functional state in the design. Use of non-functional inputs as test vector may stimulate more logic in the circuit than normal operational activity possible. The testing through popular scan based DFT incure high shift power and capture power durring test application. Shift power can be avoided by output gating of scan FF [1]. However high power during launch or capture cycles of test application is more risky. The high activity caused by non-functional test results into voltage droop and trigger false delay fault. These false failures lower down the manufacturing yield. To safely conduct the test application, test power should be minimized. However, reduction in activity through logic reduces the number of faults uncovered per pattern, eventually the pattern count increases. On the other hand, we do not want to trigger the false failures by applying excessive activity causing patterns. Thus power safe test generation should be construed as, *none of the test pattern in test set should exceed the functional power budget, P_{th} of the circuit.*

Weighted Switching Activity (WSA), one of power evaluation metric, is determined as the weighted (multiplied by fan-out) summation of the logic transitions occurring at nets of the circuit. The metric is less efficient to utilize at the test generation process in the following ways: **a)** full logic simulation is required to generate WSA for the applied inputs. Computation becomes tedious during ATPG due to memory exploration. The complexity involved in the logic simulation gets worst when non-zero delay model is considered. **b)** Power consumption may vary for cells/gates performing the identical logical operation but differ in implementation. WSA considers the same power values for both of the cases unless a proper multiplicative coefficient is defined for the corresponding cell. However, storage of information for all library elements is challenging. To overcome the above issues, we have used the analytical model based power analysis to compute power dissipation by applied inputs.

The major categories in which research has been developed are: data manipulation of an available test set, innovative ATPG algorithms and better DFT architectures to produced power safe test set [2]-[13]. The techniques which try to modify the available test set to minimize the power are called data manipulation techniques. To ensure power safety by *X* restoration or filling of don't care bits in patterns is reported by [4]. Authors show significant reduction in WSA. The a mask is constructed for response of unsafe pattern which indicate tester about possible false failure. The [3] utilizes SAT-based optimization approach to limit the WSA under budget. Method in [7] works on careful selection of test vector to compact together such that WSA can be minimized. The [8] is one such approach, which performs compression with consideration of DFT architecture of CUT. The other techniques of data manipulation such as [5][6] propose *X* filling by *bit flip* method of identified critical path FF cells and IR-drop cost function guided filling to achieve acceptable threshold respectively.

The ATPG algorithms have been augmented to take care of power violations during signal assignment process. In [9], authors identify the specific faults which are easy to detect together without much stimulation of logic. Test generation for those fault groups are targeted repeatatively until maximum coverage is achieved. Pomeranz [10] [11] proposed methods to obtain functional broadside tests to reduction of capture and shift power as well. Authors of [12] reported power-aware sensitization of path delay faults. It can incorporate non-zero propagation delay behavior of gates, thus power consumption due to glitches can also be considered. In [12] is able to keep the WSA under power budget. Another optimization based work which cosiders glitch power is reported in [13] propose modification in PODEM (Path Oriented DEcision making) algorithm to enable safe test generation.

978-3-9819263-0-9/DATE18/©2018 EDAA

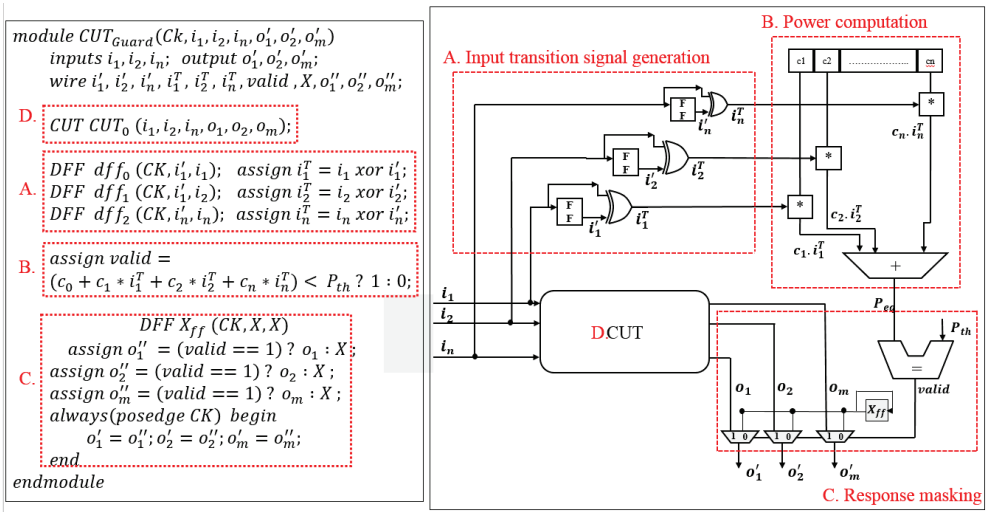

Fig. 1. ATPG Power Guard Design

We find all these strategies have limitations in following sense. The data manipulation methods may not guarantee the safe test, i.e., some pattern's power may exceed the limit. The DFT redesign approaches such as scan segmentation, scan cell reordering etc. affects the cost and performance. Moreover, ATPG process without insights of DFT may generate unsafe vectors. Lastly, implementation of new ATPG algorithms require major changes in EDA software for which industry may be reluctant to adopt.

As discussed above, solution is required without significant changes in EDA or in DFT. To the best of our knowledge, this is the first attempt in this direction. We propose power guard around CUT to prohibit power violating patterns generation. Power guard is designed to estimate the power demand by candidate test pattern and discard it by masking the fault response in case of power limit violations. The key features of the proposed method are

- Unlike pseudo power metric like WSA, we use power information from cell library in the form of analytical power models.
- This technique guarantees power safety by limiting the power values under the specified threshold value, P_{th}.
- This solution can be exercised by all the designers irrespective of third-party ATPG tools used. Advantages of well evolved ATPG algorithms in commercial tools can be employed EDA changes.
- No additional logic augmentation/modification is required to apply test safely.

The components of the proposed power guarding module, i.e., *input-transition signal generation*, *power computation* and *response making* are discussed in sect. IIA, IIB and IIC. The ATPG set up needed to obtain valid delay test is explained in Sect. IV. We report the effectiveness of this work in Sect. V.

II. ATPG Power Guard

The proposed solution develops a virtual hardware augmentation around the CUT to obtain a safe test set. The guarding logic is referred as virtual because it exists only during ATPG process and will not be implemented on silicon. The guarding logic takes the decision to mask or pass the fault response at the output point, based on power demand of candidate test pattern. Three tasks of power guard are: **a)** generation of transition signals, if input net experience a logic change from one to the another. **b)** estimation of power based on analytical model, and **c)** mask or pass the fault response,as in Figure 1.

A. *Input-transition signal generation*

The estimation of power demand from candidate test patterns during ATPG process is done based on the analytical model, which is function of transition signals at input nodes. The ATPG algorithms use time frame expansion techniques to generate V_1 and V_2 vectors of delay test pattern. Desired input values assignment to sensitize the fault and propagate the fault effect to one of the outputs are achieved through backward justification and forward implication processes. One may not have insights about signals values at nets assigned at every frame during test generation process. However, input transitions have to be identified at each cycle to check the possible power violations.

We propose the addition of memory element, FF(flip-flop) to keep track of input signal assignment in the previous clock cycle. As shown in " *input-transition signal generation*" block of Fig. 1, each input net is fed to an input of FF followed by a *XOR* gate. Input and output signals of FF elements can be XORed to see the existence of logic transition at current frame/clock cycle. For example, $V_2 = i_1, i_2, ...i_n$ is input to the circuit at current time frame and input applied at one clock cycle earlier is $V_1 = i'_1, i'_2, ...i'_n$. If the i_m have different values at consecutive clock cycles then signal value i^T_m will be logic 1, indicating logic transition. This input-transition signals will

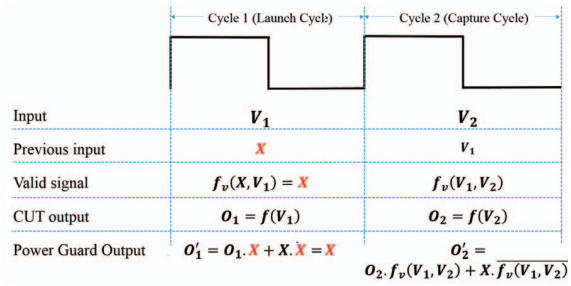

Fig. 2. 1-capture test

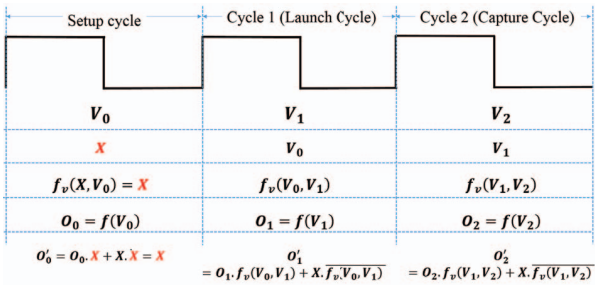

Fig. 3. 2-capture test

be further used to estimate the power consumption at current clock cycle when V_2 is applied after V_1.

B. *Power computation*

One of the important steps towards power safety is the estimation of power demand from test vectors. It will be used for blocking the test generation if, it exceeds the allowable limit. Analytical model as a function of input transitions is used to abstract the power behavior of lower level (gate/cell level).

$$P_{eq} = f_p(V_1, V_2) = c_0 + c_1.i_1^T + c_2.i_2^T ... + c_n.i_n^T \quad (1)$$

In this work we have modeled cycle accurate model given in Eq. 1. Function $f_p(V_1, V_2)$ represents estimated power when input changes from vector V_1 to V_2. The coefficients $c_0, c_1, c_2..$ are obtained by recursive learning of large random input set generated through power simulations. One can incorporate effect of glitches by consideration of power simulations with non-zero delays of gates. The graphical block, *power computation* in Figure 1 shows implementation of equation in HDL and power guard hardware. The output of power computation unit will be available at each clock cycle to compared with threshold power values to make decisions.

C. *Response masking*

The generation of invalid or power unsafe test patterns are avoided by blocking the fault response propagation at the output of the circuit. This makes ATPG to find another test to detect the fault as output is not observable due to masking. The *response masking* logic, compares the estimated power demand P_{eq} from analytical power model at each clock cycle with threshold power limit, P_{th}. If candidate test pattern pair generated by ATPG does not exceed the power limit, then the test pair is considered as a valid test. An output of comparator logic is set to 1 if the pattern is valid, otherwise invalid.

$$O' = O.valid + X.\overline{valid} \quad (2)$$

The output response of CUT is allowed to observe at power guard output if the generated test is valid. In case of invalid test, the output is masked by forwarding an output of uncontrollable FF (called X_{ff}). The X_{ff} input is driven by its own output thus the value of this not controllable through any means, thus ATPG will always see a *X* or don't care value at its output if the valid signal is low/0. Power guard outputs will be equal to *X* values in case of power unsafe test generation and test generation algorithms will not be able to observe the fault effect at output point. Such situations will force ATPG to explore alternate test for detection of the fault with valid patterns. This arrangement will make sure that generated test patterns will be always valid with respect to power violations.

III. 2-CAPTURE TEST GENERATION

The delay test patterns are applied in two functional clock cycles. Figure 2 shows output of CUT as a function of corresponding inputs for cycle 1 and cycle 2 when vector V_1 and vector V_2 are the inputs. As it can be seen at $valid$ signal during n^{th} cycle, it is a function of inputs at clock cycle n and previous cycle $n-1$ i.e. $valid = f_v(V_{n-1}, V_n)$. However, FF outputs prior to cycle 1 are unknown, i.e. (i'_1, i'_2, i'_n) in cycle 1. Unspecified FF output reflects in unresolved transition signals and eventually in the computation of power through the analytical equation. The value of $valid$ signal will be unresolved and interpreted as *X* by ATPG because input values prior to cycle 1 were unknown. An unresolved/unknown valid signal results in unobservable output response. The test generation fails as output effect of input vector V_1 is *X*. This issue can not be addressed unless memory elements are set to non-*X* values before launch cycle.

We propose a simple solution to solve the undefined initial state of power guarding circuit. Total three vector test is generated, i.e., (V_0, V_1, V_2) where is V_0 is setup vector and will not be part of test. As elaborated in Figure 3, the value of the valid signal at cycle 0 will be unknown/*X*, whereas ATPG will generate input V_0 and V_1 such that valid signal is set to logic 1 in clock 1 and output is passed to power guard output. Similarly, generation of vector V_2 without power violations is explored. This allows ATPG to pass responses of V_1 and V_2 to the observation point and ATPG can work to the generation of valid test for targeted delay fault.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed power guard technique is implemented and effectiveness is evaluated. We uses *PrimeTime* and *Tetramax* ATPG by *Synopsys*. The delay test patterns with a very high value of threshold limit in power guard module is generated which is referred as the unguarded test set. Power simulations are performed on unguarded set to identify maximum power

TABLE I
POWER GUARDED TEST GENERATION (POWER VALUES IN e^{-6})

Circuit	*Unguarded*			80% *of* P_{max}				70% *of* P_{max}			
	$\#P$	P_{avg}	$P_{peak} = P_{max}$	$\#P$	P_{avg}	P_{peak}	$\#outliers$	$\#P$	P_{avg}	P_{peak}	$\#outliers$
c1355	75	1.4	1.97	86	1.14	1.7	2	89	0.88	1.43	3
c1908	87	1.57	3.25	101	1.44	2.62	1	109	1.29	2.4	1
c3540	84	2.2	6.82	328	2.07	4.73	0	346	1.57	4.47	0
c5315	244	2.06	4.19	238	1.22	3.31	0	261	0.84	2.71	0
c6288	166	23.7	36.3	181	22.8	32.8	19	190	13.2	30.4	1
c7552	225	3.7	9.8	243	3.62	8.95	6	237	1.08	3.63	0

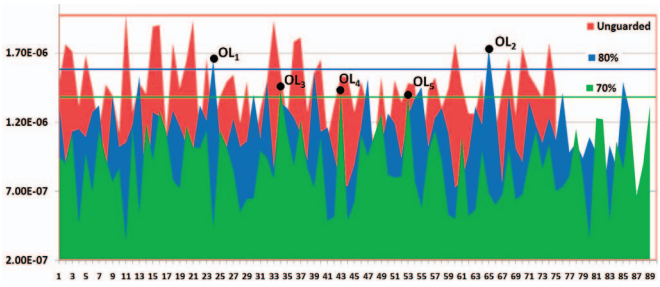

Fig. 4. Power guarded test generation for circuit C1355

consumption, P_{max}. We generate test patterns with threshold power values as 80% and 70% that of P_{max}. The plot shown in Fig. 4, represents power consumption by the circuit under test. Graph indicates power values on *y*-axis with respect to pattern number plotted on *x*-axis. The *red* colored section corresponds to power for circuit *c1355*, subjected to unguarded ATPG. The peak value $1.97e^{-6}$ is considered as maximum power, P_{max}. Thus two power guarded test sets are generated by updating threshold limits as $P_{max} * 0.8 = 1.58e^{-6}$ and $P_{max} * 0.7 = 1.38e^{-6}$ shown by solid *blue* and *green* lines respectively. The solid area of corresponding colors represents the power consumption after ATPG process is limited by given threshold values.

The results obtained by experiments on other benchmark circuit are reported in Table I. Number of patterns in test set $\#P$, average power consumption P_{avg}, and peak values of power among all the patterns in set P_{peak}, corresponding to unguarded ATPG is given in columns 2, 3 and 4 respectively. No fault coverage drop is observed from unguarded to power guarded test set. Similarly test data analysis of pattern count, average and peak values of power corresponding to power threshold of 80% and 70% of P_{max}. Column 10 and 14 report the number of test pattern violating the set threshold limit.

We could report the three important observations from experiments performed. **a)** few vectors in data set are violating the given limits(*black points*). The points OL_1, OL_2 and OL_3, OL_4, OL_5 are the outlines exceeding the set threshold budget. This may happen due to errors during abstraction of the model, i.e. analytical model can not exactly mimic the cell level power behavior. **b)** In some cases number of the test patterns in the set are same as that of higher threshold values. The number of patterns generated when the limit is P_1 is less than the number of patterns obtained with limit P_2 even when $P_1 < P_2$. Refer to circuit *c7552* in Table I, where patterns count corresponding to 70% are less than 80% power limit. It indicated ATPG is performing in the more efficient way such that compacted set is generated. This may happen when number of unique faults detected per vector are increased. **c)** Most of the cases as threshold limit is lowered down, test pattern count is increased. This is intuitive in the sense that due to tight limitations on switching activity in logic, the number of faults detected per vector is decrease.

V. CONCLUSION AND FUTURE WORK

Excessive test power is the serious issue of false failure alarms during manufacturing testing. To avoid yield loss due to power issues, the test patterns should consume less power than the functional power budget of the circuit. We address the issue by proposing a virtual power guard circuit around CUT during ATPG process. This is a new method to prohibit power violating test generation without any DFT changes or EDA modifications. The more investigation can be done for modular level circuits where accurate foundry provided micro-models of library cells can be used.

REFERENCES

[1] Mahshid Mojtabavi Naeini, et al. , "An integrated DFT solution for power reduction in scan test applications by low power gating scan cell." Integration, the VLSI Journal, 2017.
[2] Zhen Chen, et al. , "Low-capture-power at-speed testing using partial launch-on-capture test scheme." VTS, 2010.
[3] Stephan Eggersgl, et al. , "SAT-based post-processing for regional capture power reduction in at-speed scan test generation." ETS, 2016.
[4] Xiaoqing Wen, et al. , "On pinpoint capture power management in at-speed scan test generation." ITC, 2012.
[5] Tengteng Zhang, et al. , "Improved power supply noise control for pseudo functional test." VTS, 2014.
[6] Po-Fan Hou, et al. , "An IR-Drop Aware Test Pattern Generator for Scan-Based At-Speed Testing." ATS, 2016.
[7] Ranganathan Sankaralingam et al. , "Static compaction techniques to control scan vector power dissipation.", VTS, 2000.
[8] Amit Kumar, et al. , "Isometric test compression with low toggling activity.", ITC, 2014.
[9] Wen X., Kajihara S., et al. , " A new ATPG method for efficient capture power reduction during scan testing." VTS, 2006.
[10] Irith Pomeranz, "Scan Shift Power of Functional Broadside Tests." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 30, no. 9, 2011.
[11] Irith Pomeranz, "Signal-transition patterns of functional broadside tests." IEEE Transactions on Computers 62, no. 12, 2013.
[12] Matthias Sauer, et al. , "On Optimal Power-Aware Path Sensitization." ATS, 2016.
[13] V. R. Devanathan , et al. , "Glitch-aware pattern generation and optimization framework for power-safe scan test." VTS, 2007.