Test-Point Insertion for Power-Safe Testing of Monolithic 3D ICs using Reinforcement Learning*

Shao-Chun Hung†, Arjun Chaudhuri†, and Krishnendu Chakrabarty‡
†Department of Electrical & Computer Engineering, Duke University, USA
‡School of Electrical, Computer and Energy Engineering, Arizona State University, USA

2023 IEEE European Test Symposium (ETS) | 979-8-3503-3634-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ETS56758.2023.10174135

***Abstract*—Monolithic 3D (M3D) integration for integrated circuits (ICs) offers the promise of higher performance and lower power consumption over stacked-3D ICs. However, M3D suffers from large power supply noise (PSN) in the power distribution network due to high current demand and long conduction paths from voltage sources to local receivers. Excessive switching activities during the capture cycles in at-speed delay testing exacerbate the PSN-induced voltage droop problem. Therefore, PSN reduction is necessary for M3D ICs during testing to prevent the failure of good chips on the tester (i.e., yield loss). In this paper, we first develop an analysis flow for M3D designs to compute the PSN-induced voltage droop. Based on the analysis results, we extract the test patterns that are likely to cause yield loss. Next, we propose a reinforcement learning (RL)-based framework to insert test points and generate low-switching patterns that help in mitigating PSN without degrading the test coverage. Simulation results for benchmark M3D designs demonstrate the effectiveness of the proposed power-safe testing approach, compared to baseline cases that utilize commercial tools.**

I. Introduction

Monolithic 3D (M3D) integration is an emerging technology that promises to continue performance improvement when Moore's Law hits physical limits. M3D leverages fine-grained monolithic inter-tier vias (MIVs) to achieve high alignment precision and low power consumption. Compared to through-silicon vias in today's 3D technologies, MIVs are one to two orders of magnitude smaller in size and the induced capacitance is negligible [1]. These advantages enable the use of MIVs in large numbers, leading to a significant reduction in wirelength.

The benefits of M3D integration are accompanied by design and test challenges. One of the major concerns is related to the power supply noise (PSN) in the power delivery network (PDN). Compared to traditional 2D designs, M3D integrated circuits (ICs) suffer more from PSN-induced voltage droop [2]. This problem is more severe in the test mode than in the functional mode due to high switching activities during testing. Excessive switching activities in the capture cycle can lead to circuit timing degradation, which makes good chips fail on the tester and yield loss.

Various test techniques have been proposed in the literature to mitigate switching activities in the capture cycle. [3] provides a survey of power-safe testing strategies used in industrial designs. A don't-care bit-filling (X-filling) algorithm [4] has been developed to assign values to don't-care bits in a partially-specified pattern such that the switching activity can be minimized. However, X-filling algorithms limit the effectiveness of test compaction. Low-power automatic test pattern generation (ATPG) restricts the maximum switching activity of each pattern. Clock-gating switches off parts of the circuit in the capture cycle. However, low-power ATPG and clock-gating can lead to either test data inflation or a reduction in test coverage.

Test point insertion (TPI) is used to improve the testability and diagnosability of the circuit under test [5]. TPI leads to high test coverage and a reduction of pattern count. However, the study of TPI to reduce test power has not received much attention. [6] uses TPs to decrease peak power consumption during scan capture, but the locations for TPI are limited to the outputs of scan elements. In [7], TPs are inserted at the boundary of high-capture-power regions to reduce local switching activities. However, the number of variables in the proposed satisfiability (SAT)-based solution grows with the size of the circuit, which is not scalable for high-density M3D designs. A new TPI methodology is therefore needed for M3D ICs to mitigate the PSN problem during testing.

In this paper, we propose a power-safe testing framework for M3D designs. We leverage reinforcement learning (RL) to find the best locations for TPI to help eliminate the yield loss problem due to the PSN-induced voltage droop. The key contributions of this paper are as follows:

- We develop a detailed M3D power analysis flow to extract the PSN-induced voltage droop at local receivers.
- We describe an RL-based framework that uses the PSN-induced voltage droop data and determines the optimal types and locations of TPs for test power reduction.
- We demonstrate the effectiveness of the proposed approach by presenting evaluation results for M3D designs and comparing our technique with a commercial tool.

The rest of the paper is organized as follows. Section II provides an overview of M3D integration, test points, and RL. Section III presents the proposed power analysis flow for M3D designs. Details of our RL-based TPI framework are provided in Section IV. In Section V, we compare the effectiveness of the proposed framework with baseline cases that insert TPs determined by a commercial ATPG tool. Finally, Section VI concludes the paper.

II. Background

A. Monolithic 3D Integration

All device tiers in an M3D design are fabricated *in situ* on the same wafer. This has been made possible by the

*This research was supported in part by the National Science Foundation under grant CCF-1908045.

979-8-3503-3634-4/23/$31.00 ©2023 IEEE

low-temperature manufacturing processes [8]. Low-temperature processes are essential for upper-tier fabrication to prevent devices and wires underneath from damage. Despite breakthroughs in manufacturing techniques, the PSN in M3D PDNs remains a big challenge. PSN contributes to the difference between the nominal voltage at power supplies and the voltage at local receivers. In M3D designs, the supply current for the bottom-tier devices must flow through the top-tier PDN [2]. The increase in the equivalent resistance along the conduction path between power supply and receivers can lead to a high IR-drop.

Efforts have been devoted in recent years to reducing PSN in M3D designs. [2] proposes cell repositioning and metal scaling to reduce the IR-drop. [9] uses genetic programming to create a reliable PDN for M3D designs. However, such PDN optimization methodologies aim at reducing PSN in the functional mode. The voltage droop caused by PSN during testing has not been addressed in prior work on testing M3D designs. In [4], test-pattern reshaping algorithms are developed to eliminate the PSN-induced yield loss problem in M3D designs. However, the efficiency of the reshaping process is constrained by the ATPG procedure without any design-for-testability structures. Other related work on test power reduction for 2D designs is of limited effectiveness for M3D because additional PSN due to the 3D stacking in PDNs is overlooked. Therefore, there is a need for a new solution that can adequately address the M3D PSN issues during testing.

B. Test Points

Test points are of two types: control points (CPs) and observe points (OPs). CPs are used to assign specific values to certain signals in the design during testing. Typically, CPs are inserted at locations that are difficult to control by existing scan elements. OPs are introduced to capture signals at specific locations, especially where fault effects are hard to propagate through the capture logic.

The general purposes of TPI include pattern count reduction and test coverage improvement. Leveraging TPI for test power reduction has not been fully explored yet. For delay testing, OPs can facilitate the observation of fault effects; CPs can provide constant values in the capture cycles to switch off unnecessary signals. These advantages allow ATPG tools to generate low-switching patterns without an adverse impact on test coverage. Therefore, we aim at developing an efficient framework to find the best locations for TPI to reduce switching activities during testing and eliminate the PSN-induced yield loss.

C. Reinforcement Learning

RL is a class of machine learning algorithms that can learn an optimal decision-making process in an environment [10]. The optimal behavior is learned through the observations of how the environment is changed by the action taken at each time step, guided by reward values. The goal of RL is to find the policy such that the cumulative reward over discrete time steps is optimized.

RL algorithms have been shown to be effective in solving electronic design automation problems. [11] proposes a deep RL solution for macro floorplanning that can achieve significant power, performance, and area (PPA) improvements. In [12], RL models are utilized to efficiently generate standard cell layouts and fix design rule checking errors in the routing process. An RL-based gate sizing algorithm for timing optimization is developed in [13]. The effectiveness of RL in IC design problems motivates us to develop an RL-based framework for TPI. We aim at training an RL agent to find an optimal set of TPs that can reduce switching activities and improve the test coverage for low-power patterns. Such a co-optimization problem typically requires large runtime and considerable human efforts in existing algorithms. In RL, multiple objectives can be encoded into the reward function in the form of a weighted combination of evaluation metrics, which ensures that a solution can be derived more efficiently.

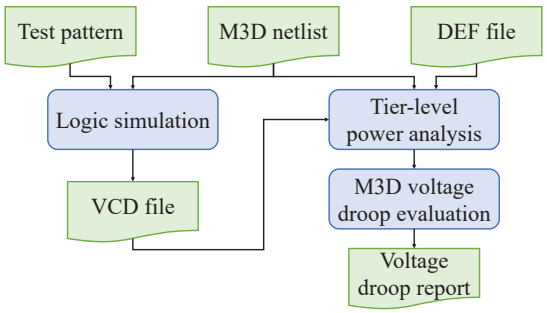


Fig. 1: Power analysis flow for M3D ICs.

III. M3D Power Analysis

In this section, we describe the proposed power analysis flow for M3D ICs to obtain the PSN-induced voltage droop at each gate. We extend the analysis method available for today's commercial tools for conventional 2D designs by considering the 3D structure of M3D PDNs. The analysis results are utilized to identify test patterns that are susceptible to yield loss and to evaluate the effectiveness of the proposed RL-based TPI framework.

A. Overview

We first converted benchmark designs into the M3D version using the method in [14]. The 3D placement and routing results after tier-partitioning are saved in a design exchange format (DEF) file. Next, we generated transition-delay fault (TDF) patterns by Siemens EDA Tessent and wrote out the patterns into Verilog testbenches. Given a pattern and the M3D design netlist, we used Siemens EDA Questa advanced simulator to conduct a gate-level logic simulation and record the switching activities during capture cycles in a value change dump (VCD) file. The M3D netlist, DEF, and VCD files were imported into Cadence Voltus to perform vector-based power analysis of each tier separately using the flow for traditional 2D designs. Finally, we combined the tier-level analysis results to evaluate the PSN-induced voltage droop in M3D. An overview of the proposed power analysis flow is shown in Fig. 1.

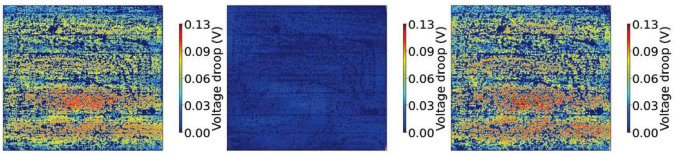
```
Input: M3D netlist N, DEF file f_DEF, Power analysis results f_p
Output: M3D voltage droop V_droop
1  V_droop := ∅
2  foreach logic gate g in N do
3      v_g = VoltageDroopExtraction(g, f_p)
4      if g in top tier then
5          V_droop := V_droop ∪ v_g
6      else
7          v_top = NearestTopTierVoltageDroop(g, f_DEF, f_p)
8          v'_g := v_g + v_top
9          V_droop := V_droop ∪ v'_g
10     end
11 end
12 return V_droop;
```

Fig. 2: Pseudo-code for M3D voltage droop evaluation.

B. M3D Voltage Droop Evaluation

Because commercial tools do not consider the impact of 3D integration, we develop an algorithm to extract the PSN-induced voltage droop in M3D designs. Fig. 2 sketched the steps involved in M3D voltage droop evaluation. Lines 2-11 iterate through every gate g in the input design. Line 3 extracts the voltage droop of g from the tier-level power analysis report generated by Cadence Voltus, denoted as v_g. In Line 5, if g is located at the top tier, we use v_g to represent the voltage droop of g. This is because the top-tier PDN is directly connected to power supplies (i.e., C4 bumps). A problem with commercial tools is that they overlook the following scenario. If g is in the bottom tier, it suffers from additional voltage droop as the supply current needs to flow through the top-tier PDN. Such an effect is ignored by commercial tools during tier-level power analysis. To simulate this scenario, in Line 7, we extract the voltage droop of the top-tier gate that is nearest to the location of g according to placement results. Lines 8-9 superimpose the extracted value on v_g to reflect the PSN effect in 3D integration and used the superimposed value to represent the PSN-induced voltage droop at g. The evaluation process is completed after iterating through all the gates in both tiers.

Fig. 3 shows the voltage droop distributions of the Tate Bilinear Pairing (Tate) benchmark with and without M3D voltage droop evaluation. Note that the values provided by a commercial tool are the difference between receivers and power sources of each tier (i.e., C4 bumps for the top tier and power MIVs for the bottom tier, respectively). From Fig. 3(a) and Fig. 3(b), it is obvious that the voltage droop problem is more severe in the top tier than in the bottom tier. This has been explained in [2]. With the reduction of footprint in M3D designs, the number of C4 bumps is limited by their large size. Excessive current demand and the limited number of power sources lead to high voltage droop in the power rails near the C4 bumps. In contrast, the bottom-tier PDN is supplied by numerous power MIVs, which serve as current sources to prevent a large-magnitude current from flowing through each MIV and surrounding metal wires. Therefore, the IR-drop problem is mitigated. This scenario points out the drawback of using existing 2D analysis flow for M3D designs. Voltage droop hotspots shown in Fig. 3 tend to be overlooked in the report



(a) (b) (c)

Fig. 3: Voltage droop distributions for the Tate benchmark: (a) top-tier distribution from ATPG; (b) bottom-tier distribution from ATPG; (c) distribution with the proposed M3D voltage droop evaluation.

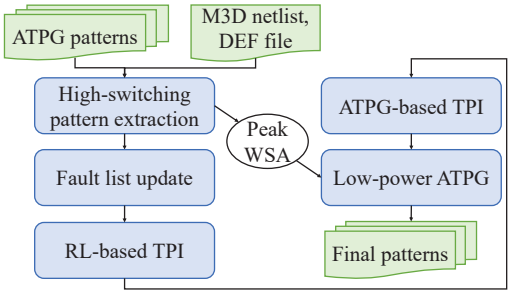


Fig. 4: Flowchart for the proposed RL-based TPI framework.

generated by the commercial tool. The impact of such hotspots on circuit timing may be underestimated, making it difficult to identify the patterns that can cause yield loss. With the proposed evaluation algorithm, the PSN-induced voltage droop in M3D designs can be extracted appropriately. This is important for the proposed RL-based framework for TPI to ensure that the yield-loss problem is fully eliminated with the TP-inserted designs and the resulting low-switching patterns.

IV. Proposed RL-based Framework

Fig. 4 presents the flowchart of the proposed RL-based TPI framework, where WSA is the weighted switching activity [15]. Given an M3D design, the corresponding DEF file, and TDF patterns generated by an ATPG tool, we conduct power simulation to extract high-switching patterns that are susceptible to yield loss. Next, we update the fault list to capture the remaining undetected faults (UDs) after pattern extraction. The updated fault list and the M3D design become inputs to our RL model to find the best set of TPs that can help in detecting the UDs without incurring high switching activities. We also leverage an ATPG-based TPI process for test coverage improvement accompanied by the TPs determined by RL. Finally, low-power ATPG is carried out on the TP-inserted design to generate patterns that can mitigate the PSN-induced yield loss problem without any adverse impact on the test coverage.

A. Extraction of High-switching Patterns

In an ATPG pattern set, only a small proportion of patterns can lead to yield loss; it is not necessary to reduce the switching activity for each pattern. Therefore, the first step in the proposed framework is to identify the high-switching patterns that lead to

```
   Input: M3D netlist N, DEF file f_DEF, ATPG Patterns P
   Output: Patterns P' ∈ P that are susceptible to yield loss
1  P' := ∅
2  P_sampled := RandomSampling(P)
3  foreach pattern p in P_sampled do
4  |   V_droop := M3DPowerAnalysis(p, N, f_DFT)
5  |   if ScaledSlack(V_droop, N) < 0 then
6  |   |   P' := P' ∪ HighSwitchPatterns(p, P)
7  |   end
8  end
9  return P';
```

Fig. 5: Pseudo-code for extracting high-switching patterns.

excessive PSN-induced voltage droop. We extract such patterns from the original pattern set. Fig. 5 sketches the steps in our high-switching pattern extraction algorithm. In Line 2, we randomly sample the input ATPG patterns because it is time-consuming to conduct power analysis on every pattern. In this work, we sample 5% of the ATPG pattern set to identify high-switching patterns. Lines 3-8 iterate through each sampled pattern, denoted as p. Line 4 runs the M3D power analysis to obtain the PSN-induced voltage droop as described in Section III. With the voltage droop values, we scale the slack of the critical path in the M3D design in Line 5. Let v_{max} be the maximum voltage droop values with p assuming the worst-case scenario. The scaled slack is calculated as follows [16]:

$$s_{droop} = t_{clock} - t_{crit} \times \frac{1 - \frac{v_{th}}{v_{DD}}}{1 - \frac{v_{th}}{v_{DD} - v_{max}}} \tag{1}$$

where t_{clock} is the clock period, t_{crit} is the delay of the critical path, v_{DD} and v_{th} are the supply voltage and the threshold voltage according to the standard cell library, respectively. In a fault-free chip without any voltage droop, the slack of the critical path is always positive. However, if the voltage droop induced by high switching activity in p makes s_{droop} become negative, an erroneous response can be captured during testing, leading to the failure of the good chip and yield loss. Therefore, in Line 6, we calculate the WSA of each pattern; all patterns with a WSA larger than the WSA of p are collected. Faults that are uniquely detected by the collected patterns are the main focus of the subsequent TPI process.

B. RL Training Environment

After extracting high-switching patterns, we conduct fault simulation with the remaining patterns in the ATPG pattern set to update the fault list. UDs in the updated fault list are imported into the RL environment to guide the training process. The objective of RL-based TPI is to find the best set of TPs that can help in detecting such UDs with low-switching patterns.

Ideally, a TP can be inserted anywhere in the design. However, this level of flexibility introduces prohibitively high runtime to evaluate every candidate location for TPI, making it impractical for RL training. Therefore, we narrow down the number of candidates in each iteration during the training process by partitioning the design into tiles and choosing one location for each tile at a time. Fig. 6 shows an overview of our RL training environment. We first convert the M3D netlist into a graph, where each fault site (i.e., every pin of a gate) creates a node, and connections between fault sites form the edges. After graph construction, node features are calculated based on the topological dependencies of the circuit and the input fault list. Next, we partition the graph into tiles based on the location of C4 bumps under the assumption that gates in the same tile tend to be supplied by the same bump. When multiple gates in the same tile switch simultaneously, a large-magnitude current is drawn from the bump and flows through the surrounding metal wires, leading to a large voltage droop. Therefore, the proposed tile-partitioning method aims at collecting gates that are highly correlated in a tile and guiding the RL model to minimize the tile-based switching activities.

With the partitioned graph, we utilize the proposed tile-based candidate heuristic to find a candidate for each tile in one training iteration. In tile T_i, we choose the gate with the largest distance to the launch and capture flops to be the candidate. This is because when an OP is inserted, the probability of detecting multiple UDs along the propagation paths from the launch flops to the candidate location is increased, while inserting a CP helps in switching off large activities between the candidate and the capture flops. The node feature of such a candidate is used as the representation of T_i, denoted as $\boldsymbol{h}_{T_i}$. Feature vectors of candidates for n tiles in the design $[\boldsymbol{h}_{T_1}, \boldsymbol{h}_{T_2} \dots \boldsymbol{h}_{T_n}]$ forms an observation of the RL model; the combinations of all tile-based candidate features construct the observation space. The action space is composed of two elements: (i) $\text{TP}_{\text{tile}} \in \{T_1, T_2, \dots, T_n\}$, and (ii) $\text{TP}_{\text{type}} \in \{\text{AND-type CP, OR-type CP, OP}\}$, where TP_{tile} and TP_{type} represent which tile-based candidate and which type of TPs are going to be inserted, respectively. The state of the environment and the reward are updated in every iteration according to the action taken by the RL agent.

C. Reward Function

The primary objective of the proposed framework is to improve the test coverage of UDs while minimizing switching activities. In this case, we define our reward function as a weighted combination of performance metrics. Let s^t and a^t be the state and the action taken at time step t, making the state become s^{t+1}. The reward function is defined as follows:

$$r^t = \alpha \cdot r^t_{UD} + \beta \cdot r^t_{SW} \tag{2}$$

where α and β are hyperparameters, r_{UD} and r_{SW} are rewards for the detection of UDs and the reduction in tile-based switching activities, respectively.

Note that the most straightforward method to evaluate the impact of each action on the test coverage and the switching activity is to re-generate patterns every time a TP is inserted. However, the synthesis of TP logic and the pattern generation with commercial tools lead to large runtime overhead, making it infeasible for RL training. To approximate the impact of each TP, we utilize the signal-transition probability (STP) metric from [17]. The STP of each node is denoted as $[p_{S_0}, p_{S_1}, p_{T_r}, p_{T_f}]$, where p_{S_0} (p_{S_1}) are the probabilities of static-0 (static-1) signals, and p_{T_r} (p_{T_f}) are the probabilities

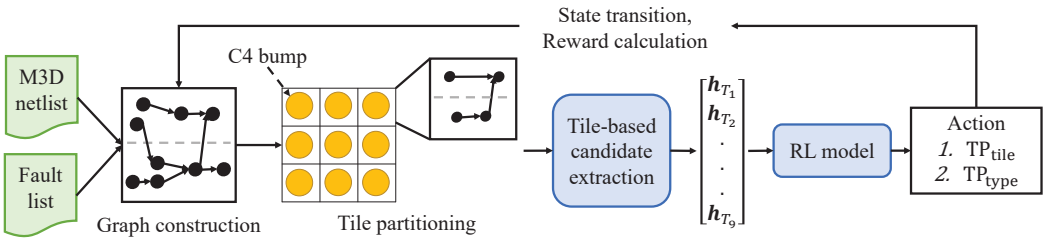


Fig. 6: Overview of the training environment of the proposed RL-based TPI framework.

of rising (falling) transitions. We set the STPs of nodes corresponding to primary inputs and pseudo primary inputs (i.e., output pins of scan elements) to be $[0.5, 0.5, 0, 0]$. Next, to simulate the launch and capture cycles in a TDF pattern, we propagate the STPs throughout the graph twice to create the initial value of each node. If an AND-type CP is inserted at node i at time t, because the signal of node i can be forced to static-1 when TPE is enabled (see Section II-B), we update the STP_i such that $\text{STP}_i^{t+1} = 0.5 \cdot [1, 0, 0, 0] + 0.5 \cdot \text{STP}_i^t$. If an OR-type CP is inserted, STP_i^{t+1} is updated to be $0.5 \cdot [0, 1, 0, 0] + 0.5 \cdot \text{STP}_i^t$. The STP_i^{t+1} is then propagated through the fan-out cone of node i to complete the state transition. The STP of every node remains unchanged if an OP is inserted.

Based on the STPs, we approximate the detection of each UD with $\sum(p_{T_r} + p_{T_f})$ of nodes in its fan-out cone. This value should be maximized to increase the probability of the UD being captured by the scan elements. Note that if an OP is inserted, the UD can be captured without propagating through the existing scan flops; therefore, we calculate the $\sum(p_{T_r}+p_{T_f})$ along the paths through the OP to represent the impact of the action. If such a summation is increased due to the action a^t, the detection reward of UD u, denoted as d_u^t, is 1; otherwise, d_u^t is -1. Let the set of UDs being affected by the action be S_{UD}. Then r_{UD}^t can be calculated as:

$$r_{UD}^t = \frac{\sum_{u \in S_{UD}} d_u^t}{|S_{UD}|} \tag{3}$$

For r_{SW}, we approximate the tile-based switching activities with $\sum(p_{T_r} + p_{T_f})$ for all the nodes in each tile. This value should be reduced to minimize the PSN-induced voltage droop. Therefore, the switching activity reward of tile T_i, denoted as sw_{T_i}, is 1 if $\sum(p_{T_r} + p_{T_f})$ is decreased after a TP is inserted; otherwise, sw_{T_i} is -1. Let the set of tiles in the design be S_T. Then r_{SW}^t is calculated as:

$$r_{SW}^t = \frac{\sum_{T_i \in S_T} sw_{T_i}^t}{|S_T|} \tag{4}$$

Using Eq. (2)-(4), the RL agent is able to learn the impact of each action on both the detection of UDs and the tile-based switching activity. In this work, we set $\alpha = \beta = 0.5$ in Eq. (2) to achieve co-optimization of test coverage improvement and switching activity reduction. An assessment using other values of α and β is left for future work.

V. Experimental Results

We evaluate the proposed RL-based TPI framework on four two-tier benchmark M3D designs, namely LDPC and Tate from OpenCores, and netcard and leon3mp from the ISPD 2012 benchmark suite [18]. We first leverage Siemens EDA Tessent to generate ATPG patterns, followed by the proposed high-switching pattern extraction algorithm to extract the patterns that are susceptible to yield loss. Details of ATPG-generated patterns for the benchmark M3D designs are shown in Table I, where V_{droop} is the PSN-induced voltage droop. Next, we utilize our RL models to find the TPs that help in detecting UDs without causing high switching activities. Our RL framework is implemented in PyTorch with the Stable-Baselines3 package [19]. Finally, we run Siemens EDA Tessent to insert TPs and conduct low-power ATPG procedures to generate the final pattern sets. In this work, the number of TPs is limited to 1% of the number of scan flops, which is the threshold adopted by Siemens EDA Tessent, to prevent large area overhead.

We create two baseline cases, namely No-TP and ATPG-TPI, by generating low-power patterns for benchmark M3D designs without any TP and with TPs determined by ATPG, respectively. Table II shows the results obtained after lower-power pattern generation with the proposed RL-based TPI and the baseline cases. Note that the values in parenthesis are changes from the original pattern sets listed in Table I.

Compared to No-TP, the proposed RL-based TPI framework provides higher test coverage for all benchmarks. This is because some faults in the designs cannot be sensitized, propagated, and captured by patterns with low switching activities without any TP inserted. For the netcard benchmark, the reduction in test coverage is up to 0.99%. Such a reduction results in more than 22K faults being undetected during testing, which significantly increases the probability of test escape. Moreover, the worse-case V_{droop} of LDPC exceeds the critical voltage droop shown in Table I. The PSN-induced yield loss problem in the No-TP case cannot be fully eliminated. The proposed RL-based TPI framework ensures that the PSN-induced voltage droop for all benchmarks is lower than the critical V_{droop}. The inserted TPs also help in improving the test coverage.

The TPI procedure in the ATPG tool mainly focuses on test coverage improvement. It is expected that for the LDPC benchmark, ATPG-TPI can achieve better test coverage than RL-based TPI. However, for netcard and leon3mp, the test coverage reduces after inserting TPs. This is due to the fact that switching activity is not considered during the ATPG-based TPI process. The inserted TPs tend to increase the ability of each pattern to sensitize and capture multiple faults, leading to additional switching activities. Such patterns with high switching activities are discarded during low-power pattern generation; therefore,

TABLE I: Details of the ATPG-generated patterns.

Design	# Patterns	Test coverage	# High-switching patterns	Worst-case V_{droop}	Critical V_{droop} to cause yield loss	Average WSA
LDPC	629	95.43%	121 (19.24%)	0.182 V	0.179 V	33.68%
Tate	366	98.43%	182 (49.73%)	0.130 V	0.120 V	22.49%
netcard	43873	97.81%	4833 (11.02%)	0.179 V	0.160 V	8.87%
leon3mp	18461	99.11%	1620 (8.78%)	0.118 V	0.116 V	6.65%

TABLE II: Comparisons between the proposed RL-based TPI framework and baseline cases.

Design	TPI method	# TPs from ATPG	# TPs from RL	# Patterns	Test coverage	Worst-case V_{droop} of low-power patterns	Average WSA	Yield loss eliminated?
LDPC	No-TP	0	0	608 (-3.34%)	95.42% (-0.01%)	0.181 V	33.09%	No
	ATPG-TPI	20	0	712 (+13.20%)	95.65% (+0.22%)	0.165 V	33.17%	Yes
	RL-based TPI	5	15	656 (+4.29%)	95.47% (+0.04%)	0.173 V	32.93%	Yes
Tate	No-TP	0	0	331 (-9.56%)	98.05% (-0.38%)	0.111 V	21.54%	Yes
	ATPG-TPI	314	0	432 (+18.03%)	98.82% (+0.39%)	0.111 V	21.32%	Yes
	RL-based TPI	78	236	460 (+25.68%)	98.88% (+0.45%)	0.111 V	20.68%	Yes
netcard	No-TP	0	0	44962 (+2.48%)	96.82% (-0.99%)	0.133 V	8.85%	Yes
	ATPG-TPI	674	0	40172 (-7.20%)	97.51% (-0.30%)	0.151 V	8.83%	Yes
	RL-based TPI	168	506	41218 (-6.05%)	97.91% (+0.10%)	0.133 V	8.82%	Yes
leon3mp	No-TP	0	0	19554 (+5.92%)	99.02% (-0.09%)	0.108 V	6.63%	Yes
	ATPG-TPI	1087	0	16862 (-8.66%)	97.74% (-1.37%)	0.113 V	6.64%	Yes
	RL-based TPI	217	870	18051 (-2.22%)	99.25% (+0.14%)	0.115 V	6.51%	Yes

test coverage is decreased. In contrast, the proposed framework includes switching activity in the reward function during RL training. The impact of TPs on test coverage during low-power pattern generation is minimized. Results with the RL-based TPI demonstrate that our framework can eliminate the voltage droop problem without any adverse impact on test coverage.

Note that the proposed RL model requires much fewer TPs for PSN mitigation compared to ATPG-TPI. This advantage allows additional TPs to be inserted to further improve test coverage. In our framework, we carry out TPI procedures for test coverage improvement with the remaining budget after inserting the TPs selected using RL. Table II shows that the proposed solution achieves better test coverage and lower worst-case V_{droop} than APTG-TPI for most designs. For leon3mp, the test coverage improvement helps to detect additional 4.7K faults during testing, leading to a significant reduction in test escapes. Moreover, RL-based TPI achieves the best average WSA for all benchmarks compared to baseline cases. This helps minimize the impact of PSN on circuit delay throughout the testing process, which also reduces the likelihood of good chips failing on the tester under small variations.

VI. Conclusion

We have developed a power analysis flow to obtain the PSN-induced voltage droop in M3D designs. We have created a pattern extraction algorithm to extract high-switching patterns that are susceptible to yield loss. Based on the above methods, we have proposed an RL-based TPI framework to generate the best TPs that help in detecting UDs without incurring high switching activities. Using the OpenCore and ISPD benchmarks, we have demonstrated that our framework is effective in reducing the PSN-induced voltage droop. With the proposed TPI solution, the yield loss problem due to the PSN in M3D designs can be eliminated, without any loss of test coverage and with a negligible increase in pattern count.

References

[1] P. Batude et al. 3D sequential integration: A key enabling technology for heterogeneous co-integration of new function with CMOS. *IEEE JETCAS*, 2(4):714–722, 2012.

[2] K. Chang et al. System-level power delivery network analysis and optimization for monolithic 3D ICs. *IEEE Trans. VLSI Systems*, 27(4):888–898, 2019.

[3] A. Bosio, P. Girard, and A. Virazel. Test of low power circuits: Issues and industrial practices. In *IEEE International Conference on Electronics, Circuits and Systems*, pages 524–527, 2016.

[4] S.-C. Hung et al. Power supply noise-aware at-speed delay fault testing of monolithic 3-D ICs. *IEEE Trans. VLSI Systems*, 2021.

[5] M.J. Geuzebroek et al. Test point insertion that facilitates ATPG in reducing test time and data volume. In *Proc.. Int. Test Conf.*, 2002.

[6] R. Sankaralingam and N.A. Touba. Inserting test points to control peak power during scan testing. In *IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems*, 2002.

[7] S. Eggersglüß et al. Formal test point insertion for region-based low-capture-power compact at-speed scan test. In *IEEE Asian Test Symposium*, 2016.

[8] S. Wong et al. Monolithic 3D integrated circuits. In *Int. Symp. VLSI-TSA*, 2007.

[9] A. Koneru et al. Reliable power delivery and analysis of power-supply noise during testing in monolithic 3D ICs. In *IEEE VLSI Test Symposium*, 2019.

[10] M. A. Wiering and M. Van Otterlo. Reinforcement learning. *Adaptation, Learning, and Optimization*, 2012.

[11] A. Mirhoseini et al. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2021.

[12] H. Ren and M. Fojtik. Invited- NVCell: Standard cell layout in advanced technology nodes with reinforcement learning. In *ACM/IEEE DAC*, 2021.

[13] Y.-C. Lu et al. RL-sizer: VLSI gate sizing for timing optimization using deep reinforcement learning. In *ACM/IEEE DAC*, 2021.

[14] Y.-C. Lu et al. TP-GNN: a graph neural network framework for tier partitioning in monolithic 3D ICs. In *ACM/IEEE DAC*, 2020.

[15] P. Girard. Survey of low-power testing of VLSI circuits. *IEEE Design Test of Computers*, 2002.

[16] A. Ramalingam et al. Robust analytical gate delay modeling for low voltage circuits. In *ASPDAC*, 2006.

[17] M. Yilmaz et al. Test-pattern selection for screening small-delay defects in very-deep submicrometer integrated circuits. *IEEE Trans. CAD*, 29(5):760–773, 2010.

[18] M. Ozdal et al. The ISPD-2012 discrete cell sizing contest and benchmark suite. http://www.ispd.cc/contests/12/ispd2012_contest.html, 2012.

[19] A. Raffin et al. Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.