Fast Low Power Rule Checking for Multiple Power Domain Design

Chien-Pang Lu
MediaTek Inc.
Hsinchu 30078, Taiwan
Knuth.Lu@mediatek.com

Iris Hui-Ru Jiang
Dept. of Electronics Eng., National Chiao Tung Univ.
Hsinchu 30010, Taiwan
huiru.jiang@gmail.com

***Abstract*—Power management via multiple power domains can effectively save power by dynamically turning off idle domains. To control domains of a design, introducing low power intent complicates the physical implementation and verification process. During the physical implementation stage, the optimization or manual ECO could be tedious, and error-prone on power/ground signal connections. Therefore, in this paper, we focus on low power rule checking at the physical implementation stage for multiple power domain design. Existing methods adopt an iterative approach, which identifies one error at a time, thus possibly requiring multiple iterations. Different from them, we propose a fast low power rule checking approach to detect all errors at one time. To do so, we separate all paths into inner-domain and cross-domain paths and extract cross-domain net topology before power rule verification. Based on the global topology, we can verify the correctness of connections and detect all errors at the same time. Experimental results show the effectiveness and efficiency of our approach, achieving 3.62X speedups to detect all errors compared with the iterative approach. Moreover, our approach can identify complicated bugs to facilitate subsequent bug fixing.**

Keywords—Power management; multiple power domains; power rules; IEEE standard 1801.

I. Introduction

Application requirement has been shifting to wearable devices and mobile designs. As semiconductor technology advances, the growth rate of battery capacity, however, is much smaller than that of chip power consumption. Hence, power management via multiple power domains has become the design mainstream, where a design is divided into multiple power domains, and each domain is dynamically powered on (when it is busy) or shut down (when it is idle) according to power states. The low power intent is described as low power rules [1]. Along with this trend, design complexity and the number of power rules are also constantly growing.

Recently, power intent verification has drained a great deal of research endeavor. For example, in [2], a general power intent and rule checking methodology is proposed. In [3-5], the formal method verifies the correctness of power intent insertion and power state sequence of each power domain. In [6], test patterns are generated to test the correctness of power connection in schematic with layout. In [7], five verification models are proposed to check mixed-signal designs in the integration stage. In [8], simulation-based power verification via automated test case generation is presented. In [9], a solution for power grid verification at design-time is proposed. In [10-14], power analysis is used to refine the power intent on each domain. Most of aforementioned approaches, however, target for power intent verification at architectural level or RTL.

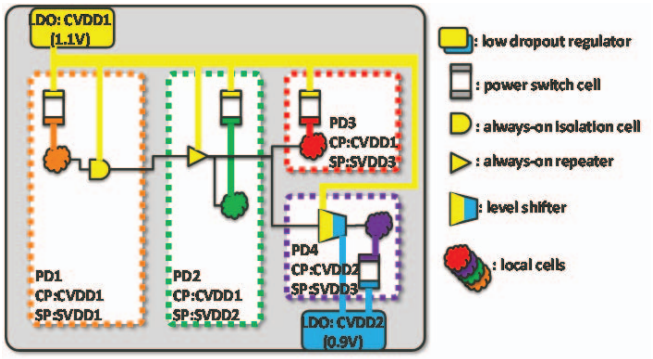

Fig. 1. Power management architecure and signal interconnection of a multi-domain design. (a) LDO: low drop regulator, power supply. (b) Power switch cell: a device turns on/off domain power. (c) Always-on isolation cell: a device pulls signal to high/low when domain is shut down. (d) Always-on repeater: a device propagates signal cross a shut down domain. (e) Level shifter cell: a device as a voltage bridge when source/sink voltage is different. (f) Local cells: devices inside a domain, which can be shut down.

Introducing low power intent into a design indeed complicates the physical implementation and verification process. During the physical implementation stage, the optimization or manual ECO (engineering change order) could be tedious, and error-prone on power/ground signal connections. Therefore, in this paper, we focus on power intent verification at the physical implementation stage for multiple power domain designs. Fig. 1 shows a multiple power domain design with four power domains, PD1, PD2, PD3, and PD4. The power/ground signal of each domain is controlled by a power switch, dynamically enabled/disabled according to power states. Sometimes a cross-domain path between two working devices at different power domains may pass through some idle domains, e.g., PD1 → PD2 → PD3 and PD1 → PD2 → PD4. We should prevent cross-domain paths from input floating and from insufficient signal threshold swing. Thus, circuit components on cross-domain paths are usually implemented by always-on isolation cells, always-on repeaters, and level shifters.

Existing methods adopt an iterative approach for physical verification, which identifies one error at a time, thus possibly requiring multiple iterations. Different from them, in this paper, we propose a fast low power rule checking approach to detect all errors at one time. To do so, we first separate all paths into inner-domain and cross-domain paths, and extract cross-domain net topology. Then, based on the global topology, we verify the correctness of power/ground connections and detect all errors simultaneously. Moreover, our approach can report complicated bugs that cannot be handled by existing methods.

978-3-9815370-8-6/17/$31.00 ©2017 IEEE

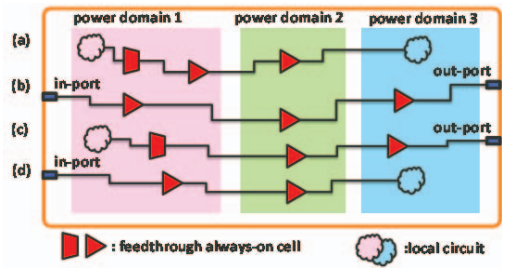

Fig. 2. Multiple power domain designs require special handling on feedthrough paths, e.g., always-on cells. Because each domain can be turned on or off individually, cross-domain paths can be classified into 4 types, (a) domain to domain. (b) external feedthrough. (c) domain to external. (d) external to domain.

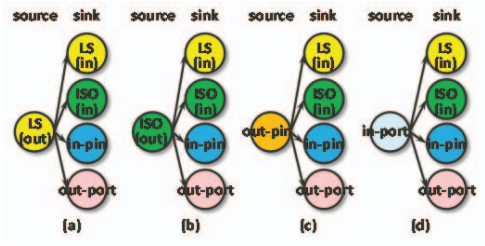

Fig. 3. Source and sink combinations for cross-domain paths. (a) Source is a level shifter (LS). (b) Source is an isolation cell (ISO). (c) Source is an output pin (out-pin). (d) Source is a block input port (in-port).

Our experiments are conducted on six 28 nm industrial designs. Experimental results show the effectiveness and efficiency of our approach, achieving 3.62X speedups to detect all errors.

The remainder of this paper is organized as follows. Section II briefly introduces low power rules and gives the problem formulation. Section III details our fast low power rule checking approach, including binding power rule constraints, cross-domain topology extraction, and power rule checking. Section IV reports experimental results. Finally, Section V concludes this work.

II. Preliminaries

In this section, we briefly introduce low power rules and describe the problem formulation.

A. Low Power Rules

References [1, 15] state the industry specification on power management architecture and multiple domain design requirement. The power intent of a design is defined by power rules, describing power domains and power states. Fig. 1 shows the power management architecture of an example design. This design contains four power domains, PD1, PD2, PD3, and PD4. All constant powers (CP) come from low dropout regulators (LDO). The functionality of each power domain is implemented by local cells driven by a switched power (SP). Switched powers are enabled/disabled at alive/idle domains by power switch cells according to power states. Sometimes a cross-domain path between two alive devices at different power domains may pass through some idle domains. Hence, for cross-domain paths, we check the following rules:

- *Electrical rule* guarantees no input floating when the source domain is shut down.
- *Stable voltage rule* ensures full signal threshold swing while source and sink domains are at different voltage levels.

When PD1 and PD3 are alive, and PD2 is off, the electrical rule checks if path PD1 $\rightarrow$ PD2 $\rightarrow$ PD3 works correctly. Hence, components on this kind of cross-domain paths are implemented by always-on isolation cells and always-on repeaters to propagate signal from source to sink domain correctly.

When the source and sink domains are at different voltage levels, the stable voltage rule tests whether the signal swing is aligned to the voltage level at sink domain, e.g., path PD1 (1.1V) $\rightarrow$ PD2 (1.1V) $\rightarrow$ PD4 (0.9V). Hence, components on this kind of cross-domain paths are implemented by always-on isolation cells, always-on repeaters, and level shifters.

B. Problem Formulation

As mentioned earlier, during the physical implementation stage, the optimization or manual ECO could be tedious, and error-prone on power/ground signal connections. Therefore, in this paper, we focus on power intent verification at the physical implementation stage for multiple power domain designs. Thus, the power rule verification (PRV) problem at the physical implementation stage can be thus formulated as follows:

The PRV Problem: Given a multiple power domain design netlist and power rules, verify if the design satisfies these power rules on power/ground connection for all devices.

For modern multiple power domain design, a circuit is partitioned into blocks during physical implementation. Each block may contain one or more power domains. Power rule verification can be performed on each block individually. To do so, power rules for a block also include block boundary constraints. For each block, all signal connections can be separated into inner-domain paths and cross-domain paths. In an inner-domain path, the source and sink devices belong to the same domain, while in a cross-domain path, the source and sink devices are located in different domains.

Usually, most of power intent (in terms of power rules) is created for cross-domain paths to ensure devices on cross-domain paths are driven by correct supplies (power/ground connections). As shown in Fig. 2, cross-domain paths can be classified into four types, including domain to domain, external feedthrough, domain to external, and external to domain. For devices on an inner-domain path, power should be provided by local supply. For devices on a cross-domain path, power should be provided by a correct supply defined by power rule constraints and power state conditions. Besides, we shall guarantee that the power/ground connection of devices on a cross-domain path satisfies the electrical rule and the stable voltage rule.

Furthermore, for a cross-domain path, we shall check rules at a level shifter output, an isolation cell output, a domain output pin (out-pin), or a block input port (in-port) for the source, while checking rules at a level shifter input, an isolation cell input, a domain input pin (in-pin), or a block output port (out-port) for the sink. Then, we have four cases for the source and sink relationship of cross-domain paths as shown in Fig. 3.

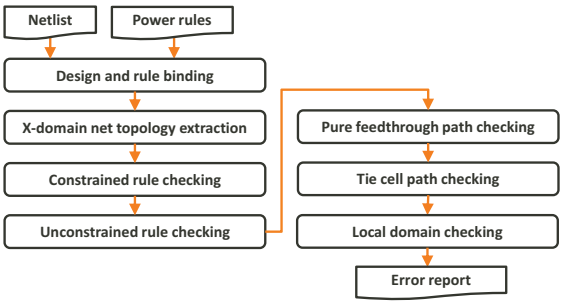

Fig. 4. The overview of our low power rule checking approach.

If source is a level shifter (see Fig. 3(a)), the supply of the cross-domain path refers to sink domain power. If source is an isolation cell (see Fig. 3(b)), the supply of the cross-domain path follows the isolation cell rule. If source is an output pin (see Fig. 3(c)), the supply of the cross-domain path follows the source domain. If source is a block input port (see Fig. 3(d)), the cross-domain path is constrained by a block boundary constraint.

III. Low Power Rule Checking

In this section, we present our fast low power rule checking approach at the physical implementation stage for multiple power domain design.

A. Overview

Fig. 4 shows the overview of our low power rule checking approach. In order to detect all errors at one iteration, we extract cross-domain net topology before power rule verification. Based on the global topology, we devise several scenarios for cross-domain paths. Every device belongs to only one scenario, and we check it accordingly.

As mentioned in Section II-B, a design is partitioned into blocks, and each block can be verified individually. First, given a block design netlist and corresponding power rules, we bind rules to the design. Second, we extract cross-domain net topology, thus separating inner-domain and cross-domain paths. Third, based on the binding relation and the global topology, we check power rule constraints. Fourth, we check all remaining unconstrained cross-domain paths. Fifth, we verify pure feedthrough paths, which are constrained by block

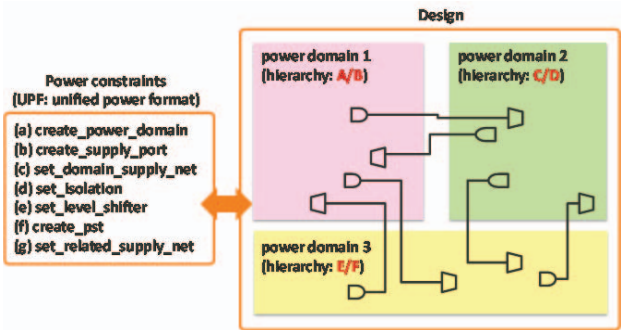

Fig. 5. Building up the relation between power constraints and a design. (a) Assigning power domain hierarchy. (b) Creating design supply port, corresponding to constant power. (c) Assigning domain supply net. (d) Creating an isolation cell rule. (e) Creating a level shifter rule. (f) Describing a power state table. (g) A block boundary constraint (assigning special power/ground net connection).

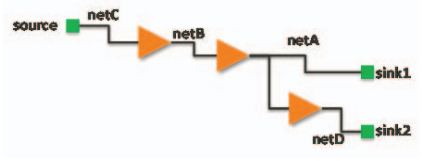

Fig. 6. Cross-domain net topology extraction. Cross-domain paths starting from the same source can be extracted by BFS/DFS.

boundary constraints. Sixth, we check all tie signal paths. Seventh, we check inner-domain paths. Finally, we generate the error report.

B. Design and Rule Binding

Power rules are described in the industry standard, unified power format (UPF) [1]. Before checking the correctness of the power/ground connections of devices, we construct the relations between power constraints and a design.

As shown in Fig 5, each power domain corresponds to several design hierarchies, e.g., hierarchies A/B, C/D and E/F are individual domains. The constant power and ground is defined. For each power domain, the supply net is also defined. The supply power or ground can be constant or switched. If a domain is supplied by some constant power, it is an always-on domain; otherwise, it belongs to a switchable domain. Each switched power comes from a constant power via a power switch cell. The cross-domain path interface is controlled by isolation cell rules and level shifter rules. If both source and sink belong to switchable domains and their constant power is at the same voltage level, then there is some isolation cell rule on the domain boundary. If the supplied voltage levels of source and sink domains are different, then there is some level shifter constraint for the cross-domain path. The on/off state sequence of each domain is recorded by a power state table. If a cross-domain path passes through some block, then a block boundary constraint is applied.

C. Cross-Domain Net Topology Extraction

Before power rule verification, we extract cross-domain net topology first. All paths can thus be separated into inner-domain paths and cross-domain paths, and we verify them in the following Sections (Sections III-D~III-H).

Before topology extraction, we can break all cross-domain paths round multiple-input gates (e.g., AND, OR, etc.). Then, cross-domain paths form a forest. Cross-domain paths starting from the same source can be extracted by breadth-first search (BFS) and depth-first search (DFS). Moreover, to facilitate retrieving the detailed path information, each device is associated with a prefix, which records the path information from the source to the investigated device. The prefix recording is performed along with the BFS/DFS traversal. The extraction starts from block input ports and domain output pins. Cross-domain net topology extraction can be done in O(N), where N represents the number of nets in a design netlist. (see Fig. 6.)

Cross-domain paths starting from the same source will be verified by the same rule. Moreover, after cross-domain net topology extraction, all devices on a path ending with output floating can be identified. These devices can be removed for area saving.

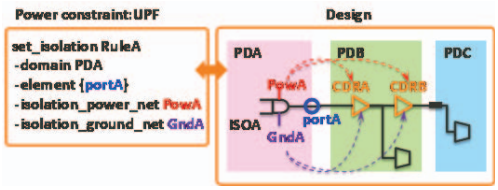

Fig. 7. Constrained rule checking on power/ground connections of isolation cells: Isolation RuleA is defined on power domain PDA, through port portA, with isolation power net PowA and ground net GndA. The cross-domain repeaters are CDRA and CDRB. For repeaters CDRA and CDRB, power follows PowA, while ground follows GndA.

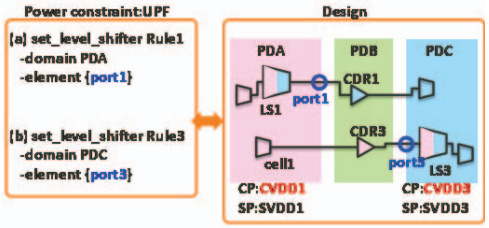

Fig. 8. Constrained rule checking on power/ground connections of level shifters: Level shifter Rule1 is defined on power domain PDA, and level shifter Rule3 is defined on power domain PDC. (a) For LS1, repeater CDR1 follows LS1's output domain voltage (i.e., PDC). (b) For LS3, repeater CDR3 follows LS3's input domain voltage (i.e., PDA).

D. Constrained Rule Checking

After cross-domain net topology extraction, inner-domain paths and cross-domain paths are separated. All cross-domain paths can be divided into two disjoint sets, constrained paths and unconstrained paths. In this step, constrained paths are checked; constrained power rules are checked on the last device in the source domain on a cross-domain path, and the device could be a repeater, an isolation cell, or a level shifter. Besides, repeaters on the cross-domain path are also checked.

During topology extraction, constraint check points are traversed and recorded. During verification, the traced common source devices can be verified and applied with the same rule simultaneously.

As shown in Fig. 7, a cross-domain path starts from power domain PDA, through PDB, to PDC, with an isolation cell ISOA. Along this path, repeaters CDRA and CDRB follow the power and ground connections of ISOA, i.e., PowA and GndA.

If the source and sink domains of a cross-domain path are at different voltage levels, a level shifter is required as shown in Fig. 8. After path extraction, cross-domain repeaters CDR1 and CDR3 are traced, and their sources, LS1 and cell1, are in power domain PDA. For Fig. 8(a), the level shifter is located in the source domain, so the power of CDR1 follows the sink domain PDC (i.e., CVDD3). For Fig. 8(b), the level shifter is located in the sink domain, so the power of CDR3 follows the source domain PDA (i.e., CVDD1).

E. Unconstrained Rule Checking

After constrained cross-domain paths are verified, the remaining unconstrained paths are checked. The devices on an unconstrained path have undefined supplies, thus leading to unstable signal propagation. Sometimes, these devices are introduced by manual ECO or unexpected constraint loss. Similar to constrained rule checking, devices on these unconstrained paths can be checked, and additional rules can be added accordingly.

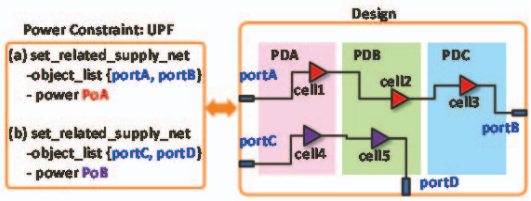

Fig. 9. Pure feedthrough path checking.

F. Pure Feedthrough Path Checking

A pure feedthrough path means a path which starts at an input port and ends at an output port. The source supply of this path is constrained by an input port, as shown in Fig. 9. By topology extraction, the repeaters on feedthrough paths are recorded. These repeaters are checked by corresponding rules. As shown in Fig. 9(a), ports portA and portB are constrained with power PoA, and repeaters cell1, cell2 and cell3 follow power PoA. As shown in Fig. 9(b), ports portC and portD are constrained with power PoB, and repeaters cell4 and cell5 follow power PoB.

G. Tie Cell Path Checking

A tie cell is used to provide a constant signal, fed into some gate inputs. For multiple power domain design, the power/ground of a tie cell cannot be shut down before its driving devices are turned off. During topology extraction, repeaters on the paths from a tie cell to its sinks are extracted. Then, these repeaters cannot be turned off before the sinks are shut down.

As shown in Fig. 10(a)(b)(c), the tie path connections are safe, because the power of the tie cells and repeaters is synchronized with the sink power. As shown in Fig. 10(d), a tie device is supplied by a local power, SVDD2, and thus the connection could cause input floating on the repeater when power domain PDB is turned off earlier than domain PDC.

H. Local Domain Checking

As shown in Fig. 4, in our approach, devices on cross-domain paths are checked in the following order: constrained rule checking, unconstrained rule checking, pure feedthrough path checking, and tie path checking. After devices on cross-domain paths are checked, the rest of unchecked devices belong to local domains. The power and ground connections on these local devices follow their local domain supply. For example, the unverified cells indicated in Fig. 11 are verified

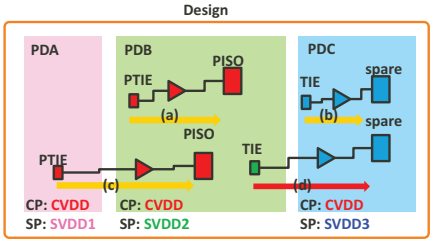

Fig. 10. Tie cell path checking. (a) A correct inner-domain always-on tie path. (b) A correct inner-domain local tie path. (c) A correct cross-domain always-on tie path. (d) An incorrect cross-domain local tie path. PTIE and PISO are always-on devices.

whether their supplies are connected to local power/ground of domain PDC.

IV. Experimental Results

We implemented our approach in C++ and Tcl languages and executed our program on a 2.9 GHz Intel Xeon Linux workstation with 128 GB memory. We conducted two sets of experiments to show the effectiveness and efficiency of our approach.

A. Comparison with the Iterative Approach

The first set of experiments were conducted on six 28 nm industrial designs as listed in Table 1, where Ckt1 is a multimedia encoder, Ckt2 is a Wi-Fi and Bluetooth connectivity, Ckt3 is an inter-block communication, Ckt4 is a modem subsystem, Ckt5 is a video decoder, and Ckt6 is a LTE subsystem. "#domain" means the number of power domains, "#inst" means the number of instances, "#rule" means the number of power rules, and "#Xpath" means the number of cross-domain paths.

Table 2 lists the percentage of accumulated verified devices after each checking step. "cons. check" means the percentage after constrained rule checking, "uncons. check" means the accumulated percentage after unconstrained rule checking, "pure FT check" means the accumulated percentage after pure feedthrough path checking, "tie path check" means accumulated percentage after tie path checking, and "rem. check" means accumulated percentage after local domain checking. It can be seen that all paths are successfully verified by our approach. In addition, in most of designs, less than 5% paths belong to cross-domain paths.

Table 3 shows the comparison with the iterative approach adopted by existing methods. "version" gives the circuit status when we performed power rule verification: eco1 means the circuit after the first run of ECO is applied, and beco means the circuit before ECO. "#rule" gives the number power rules. "#det_error" gives the number of detected errors at the first iteration. "#miss" gives the number of missed errors at the first iteration. "#ite" gives the number of iterations. "#rm cell" gives the number of removable cells with output floating (#miss does not count #rm_cell and #uncons). "#uncons" gives the number of unconstrained devices (spare cells are also included).

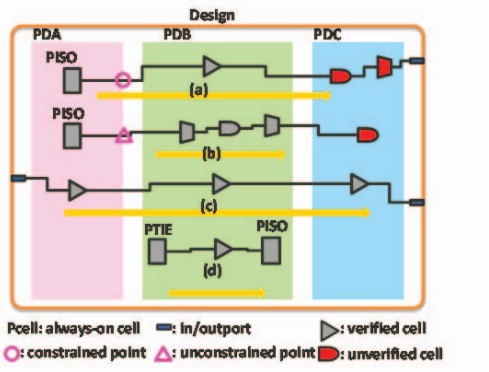

Fig. 11. Devices on cross-domain paths are checked in the following order: (a) Constrained rule checking. (b) Unconstrained rule checking. (c) Pure feedthrough path checking. (d) Tie path checking. PTIE and PISO are always-on devices.

Table I. Benchmark Statistics.

circuit	#domain	#inst	#rule	#Xpath
Ckt1	5	3323K	24	2604
Ckt2	3	1377K	10	4172
Ckt3	2	1407K	16	9864
Ckt4	7	1792K	40	9490
Ckt5	3	1112K	20	2370
Ckt6	8	2422K	28	5282

Table II. Verified Percentage after Each Step.

circuit	cons. check	uncons. check	pure FT check	Tie path check	rem. check
Ckt1	2%	2%	3%	4%	100%
Ckt2	3%	3%	4%	4%	100%
Ckt3	4%	4%	5%	5%	100%
Ckt4	4%	4%	5%	5%	100%
Ckt5	2%	2%	3%	3%	100%
Ckt6	4%	4%	5%	5%	100%

"runtime" gives the total runtime.

Different from existing methods, our approach is seamlessly integrated with placement and routing tool environment. The iterative approach reports one error for each path at a time. Hence, the iterative approach requires multiple iterations when more than one error occurs on a cross-domain path. In contrast, our approach can detect all errors simultaneously. It can be seen that our approach outperforms the iterative approach for every case and achieves 3.62X speedups on average. (Please note that the netlist preparation time is not counted for the iterative approach.) Moreover, because of the extracted global topology, we can identify removable cells due to output floating and recognize unconstrained devices, which cannot be found by the iterative approach.

B. Discussion on Complicated Bugs

The iterative approach may miss some bugs as listed in Table 3.

As shown in Fig. 12, a path starts from an isolation cell ISO1 and ends with output floating. By Rule1, ISO1 is supplied by a constant power CVDD. In this case, inst2 is output floating, and inst1 and inst2 are supplied by switched power SVDD. The iterative approach passes this case because there is no electrical or stable voltage rule violation. In contrast, our approach notifies user that ISO1, inst1, and inst2 can be removed for area saving.

Compared with the iterative approach, our approach can report all power and ground connection errors at one time. As shown in Fig. 13, three repeaters on a cross-domain path have wrong power connections. By Rule2, an isolation cell ISO2 is supplied by a constant power CVDD, and all devices on the cross-domain path should be supplied by CVDD, too. However, inst3, inst4 and inst5 are supplied by switched power SVDD. This wrong power connection causes input floating of cell2 in domain PD2 while SVDD is shut down. The iterative approach requires three iterations to report all errors. In contrast, based on cross-domain topology extraction, our approach can verify the supplies of inst3, inst4 and inst5 simultaneously.

In a layout with always-on devices, the primary power pin DVDD must be aligned with power rail, and the secondary power pin RVDD can be connected to constant power. As

TABLE III. COMPARISON WITH THE ITERATIVE APPROACH.

Benchmark statistics			Iterative				Ours			
circuit	version	#rule	#det_error	#miss	#ite	runtime*	#det_error	#rm_cell	#uncons	runtime
Ckt1	eco1	24	163	25	3	42m	250	62	0	6m
Ckt2	eco1	10	1	18	6	65m	19	0	0	3m
Ckt3	beco	16	3	3	1	13m	72	0	69	11m
Ckt4	eco1	40	5	0	1	14m	369	323	41	13m
Ckt5	beco	20	0	0	1	14m	46	34	12	5m
Ckt6	beco	28	0	5**	1	15m	91	86	0	7m
Ratio						3.62				1.00

* The netlist preparation time is not counted for the iterative approach.
** The iterative approach cannot detect these five errors.
*** Iterative(#det_error + #miss) = Ours(#det_error - #rm_cell - #uncons).

shown in Fig. 14, the repeaters on a cross-domain path are implemented by always-on repeaters and located in some shut down domains. The primary and secondary power pins should be both connected to constant power CVDD. Logically, connecting CVDD to both power pins does not cause any rule violations, and thus the iterative approach does not report errors for this case. However, physically the primary power pin is aligned with local power, SVDD, so the current connection is not allowed and can be reported by our approach. The correct power connections are that DVDD connects to SVDD, while RVDD connects to CVDD.

V. CONCLUSION

Power management via multiple power domains can effectively save power, but introducing low power intent complicates the physical implementation and verification process. In this paper, we have proposed a fast low power rule checking approach at the physical implementation stage for multiple power domain design. Because of cross-domain net topology extraction, we can verify the correctness of connections and simultaneously detect all errors. Experimental results have shown the effectiveness and efficiency of our approach. Compared with the iterative approach, we have achieved 3.62X speedups to detect all errors. Moreover, our approach can identify complicated bugs to facilitate subsequent fixing.

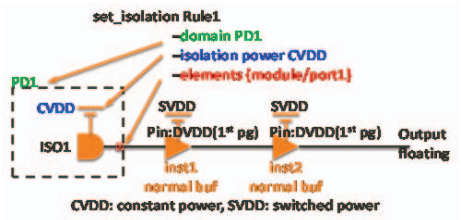

Fig. 12. Output floating: all repeaters on the fanout path can be removed.

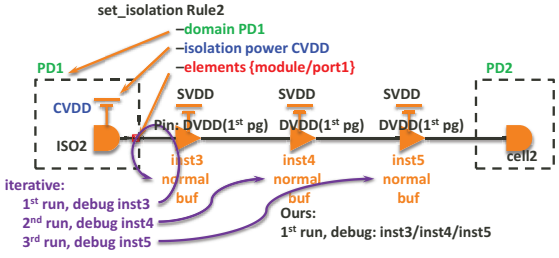

Fig. 13. The iterative approach requires multiple iterations to report all errors.

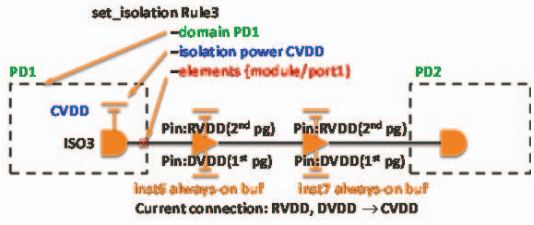

Fig. 14. The iterative approach cannot identify physical connection error on always-on repeaters. An always-on cell has the primary power pin DVDD and the secondary power pin RVDD.

REFERENCES

[1] IEEE standard for design and verification of low-power integrated circuits. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6521327&filter=AND(p_Publication_Number:6521325).

[2] F. Bembaron, S. Kakkar, R. Mukherjee, and A. Srivastava, "Low power verification methodology using UPF," in Proc. DVCon, pp. 228--233, 2009.

[3] A. Hazra, S. Goyal, P. Dasgupta and A. Pal, "Formal verification of architectural power intent", IEEE Trans. VLSI, 21(1), pp. 78--91, Jan. 2013.

[4] A. Hazra, S. Mitra, P. Dasgupta, A. Pal, Bagchi, Debabrata and K. Guha, "Leveraging UPF-extracted assertions for modeling and formal verification of architectural power intent," in Proc. DAC, pp. 773--776, 2010.

[5] R. Mukherjee, A. Srivastava, and S. Bailey, "Static and formal verification of power aware designs at the RTL using UPF," in Proc. DVCon, pp. 42--47, 2008.

[6] A. Mohy and M. A. Makarem, "A robust and automated methodology for LVS quality assurance," in Proc. Design and Test Workshop, pp. 1--3, 2009.

[7] C. Liang, "Mixed-signal verification methods for multi-power mixed-signal System-on-Chip (SoC) design," in Proc. ASICON, pp. 1--4, 2013.

[8] C. Trummer, C.M. Kirchsteiger, C. Steger, R. Weiss, D. Dalton, and M. Pistauer, "Simulation-based verification of power aware system-on-chip designs using UPF IEEE 1801," in Proc. NORCHIP, pp. 1--4, 2009.

[9] I. A. Ferzli, F.N. Najm and L. Kruse, "A geometric approach for early power grid verification using current constraints," in Proc. ICCAD, pp. 40--47, 2007.

[10] N. Khan and W. Winkeler, "Power assertions and coverage for improving quality of low power verification and closure of power intent," in Proc. DVCon, pp. 53--58, 2008.

[11] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in Proc. ISCA, pp. 83--94, 2000.

[12] D. Chaver, L. Piuel, M. Prieto, F. Tirado and M. C. Huang, "Branch prediction on demand: an energy-efficient solution," in Proc. ISLPED, pp. 390--395, 2003.

[13] V. Viswanath, J. A. Abraham and W.A. Hunt, "Automatic insertion of low power annotations in RTL for pipelined microprocessors," in Proc. DATE, pp. 1--6, 2006.

[14] V. Viswanath, S. Vasudevan and J. A. Abraham, "Dedicated rewriting automatic verification of low power transformations in RTL," in Proc. VLSI Design, pp. 77--82, 2009.

[15] Advanced configuration and power interface (ACPI). Available at: http://www.acpi.info.