Horizontal Benchmark Extension for Improved Assessment of Physical CAD Research

Andrew B. Kahng†, Hyein Lee‡ and Jiajia Li†
†ECE and ‡CSE Departments, University of California at San Diego
La Jolla, CA, 92093
abk@ucsd.edu, hyeinlee@ucsd.edu, jil150@ucsd.edu

ABSTRACT
The rapid growth in complexity and diversity of IC designs, design flows and methodologies has resulted in a benchmark-centric culture for evaluation of performance and scalability in physical-design algorithm research. Landmark papers in the literature present vertical benchmarks that can be used across multiple design flow stages; artificial benchmarks with characteristics that mimic those of real designs; artificial benchmarks with known optimal solutions; as well as benchmark suites created by major companies from internal designs and/or open-source RTL. However, to our knowledge, there has been no work on horizontal benchmark creation, i.e., the creation of benchmarks that enable maximal, comprehensive assessments across commercial and academic tools at one or more specific design stages. Typically, the creation of horizontal benchmarks is limited by mismatches in data models, naming conventions, which limits assessment. Underlying problem is that tools do not simply convert data formats to be used across different tools, technologies, and benchmark suites. In this paper, we describe methodology and robust infrastructure for “horizontal benchmark extension” that permits maximal leverage of benchmark suites and technologies in “apples-to-apples” assessment of both industry and academic optimizers. We demonstrate horizontal benchmark extensions, and the assessments that are thus enabled, in two well-studied domains: place-and-route (four combinations of academic placers/routes, and two commercial P&R tools) and gate sizing (two academic sizers, and three commercial tools). We also point out several issues and precepts for horizontal benchmark enablement.

1. INTRODUCTION
Scaling of integrated system complexities, along with rapid changes in both SOC architectures and underlying process technologies, continue to demand improvements of VLSI CAD algorithms and tool capabilities. Particularly in the academic research context, benchmarks have been widely adopted as the basis for evaluation and comparison of VLSI CAD algorithms and optimizations [1] [21]. Evaluations mainly focus on solution quality and runtime; optimization domains include synthesis, partitioning, placement, clock tree synthesis, global routing, gate sizing, and other aspects of IC implementation. Since the mid-1980s, various benchmark suites and methods for artificial benchmark generation have been published, as reviewed in Section 2 below [4] [3] [2] [7] [9] [15].

At a high level, benchmarks in VLSI CAD (and, specifically, physical design) may be classified as real (derived from actual designs), artificial (intended to mimic aspects of real designs, and often the product of parameterizable generators), and artificial with known optimal solutions (realistic, but with optimal solutions embedded in the benchmark construction). On the other hand, vertical benchmarks [14] explicitly seek to enable evaluation ofPermission to make digital or hard copies of all or parts 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 and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GLSVLSI ’14, May 21–23, 2014, Houston, Texas, USA.
Copyright 2014 ACM 978-1-4503-2816-6/14/05 ...$15.00.
http://dx.doi.org/10.1145/2591513.2591540.

CAD tool performance across a span of several flow stages, via representations at multiple levels of abstraction.

For nearly three decades, VLSI CAD benchmarks, and their use, have faced the same quandary. Essentially, “leading-edge”, “real” designs embody high-value intellectual property of their creators, and cannot be easily released; “old” or “artificial” benchmarks potentially drive CAD research in stale or wrong directions. Thus, when “real” benchmarks are released to the academic research community, their influence can be enormous, as was seen with the ISPD98 partitioning benchmark suite from IBM [2]. Further, the difficulty of obtaining real, leading-edge designs as open drivers for research raises an obvious challenge: How can we maximally leverage available benchmarks as enablers of (physical) CAD research?

To our knowledge, no previous work pursues the maximal assessment of academic research and its prevailing industry context (i.e., across various process/library technologies, benchmark circuits, and tools), at one or more particular flow stages, while still enabling assessment of maximum potential use cases. In this work, we present high-level observations on the potential of horizontal benchmarking as enablers of research. An important question to consider is how we can most effectively capture the key characteristics of “real” designs in our benchmarks. To answer this question, we adopt a methodology which involves a set of high-level observations.

Permission to make digital or hard copies 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 and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

We make several high-level observations. First, we observe that our benchmarking enablement approach allows for improved and more comprehensive assessment of academic and commercial CAD tools. Second, we observe that our approach allows for more comprehensive evaluation of key design challenges. Finally, we observe that our approach allows for improved and more comprehensive assessment of key design challenges. Overall, our approach allows for improved and more comprehensive assessment of academic and commercial CAD tools.
We describe issues and challenges that allow maximum leverage of industry-provided benchmark data, and maximal “apples-to-apples” assessment of academic research tools in industry contexts (hence, technology evaluation and transfer) across benchmarks, technologies and tools, which will provide indications to designers on how to improve their tools’ robustness/performance.

We enumerate a number of challenges in horizontal benchmark creation, along with our solution approaches.

We demonstrate the feasibility of apples-to-apples assessments in the P&R and sizing domains, using a rich mix of academic benchmark and real design data, four distinct process technologies, and a number of academic and commercial optimizers.

Our infrastructure for horizontal benchmark extension and enablement (conversion scripts, tool runscripts, mapped benchmarks) is available on the web [28] for use by industry and academia.

The rest of this paper is organized as follows. Section 2 briefly reviews relevant work on academic benchmark suites and generators. In Section 3, we describe issues and challenges of horizontal benchmark enablement – both general issues, and issues specific to P&R or sizing – along with our solution approaches. Section 4 describes our experimental setup and results that demonstrate the feasibility of horizontal assessment in the P&R and sizing domains. Section 5 gives our conclusions and some perspectives on broader issues pertaining to horizontal benchmark enablement.

2. RELATED WORKS

Previous literature on benchmark generation (a recent review is given in [21]) addresses two main categories of benchmarks. Real benchmarks are derived from actual (but not too recent, for IP protection reasons) industrial designs. Figure 2 shows gate count over time in largest MPU products (per the 2011 ITRS [33]) and in largest circuits of notable benchmark suites. Superficially, gate counts in real designs have increased by 22× since 1998, while over the same 15-year period the gate count of the largest benchmark netlists has increased by 12×; there is currently still a “1000×” gap (indicated by the scale difference between two y-axes). More realistically, the gap between academic benchmark and real design complexities can be estimated (based on gate count) at 5× ~ 20× when we calibrate to individual hard macros and top-level netlists in modern SOCs, or flat ASIC designs.3 Artificial benchmarks are algorithmically generated, typically for a specific field or problem domain such as row-based placement or power grid analysis. The primary concern in artificial benchmark generation has been to capture salient attributes of real designs, such that academic CAD research is appropriately driven to intercept future industry needs. Thus, artificial benchmarks have attempted to match such parameters of real designs as Rent exponent, fanin/fanout distribution, path depth, etc. Important directions have included randomization techniques, and methods to generate artificial benchmarks with known optimal solutions. We briefly review examples of each benchmark type.

3There may be a chicken-egg dynamic here: growth of hard macro gate counts in SOC designs is limited by scaling of capacity (i.e., QoR/runtime sweet spot) of EDA tools, which has slowed in recent years.

Benchmark Suites Based on Real Designs. The highly influential MCNC benchmark suites [3] [4], published in the 1980s, have been used in various CAD applications such as automatic test pattern generation (ATPG), logic synthesis, netlist partitioning, and placement. The largest instance in the ISCAS89 benchmark suite has ~70K gates and ~3K flip-flops. The ISPDB9 benchmark suite [2], developed for netlist partitioning applications, includes 18 circuits with module counts up to ~210K. Since the benchmark circuits are generated from IBM internal designs, functionality, timing and technology information is removed. The ITC99 benchmark suite [10] from the same time frame contains both RTL and gate-level benchmarks, the largest of which has ~200K gates and ~7K flip-flops, targeted at ATPG algorithm evaluation.

Over recent ISPD contests, the ISPD05 and ISPD06 benchmarks [31] respectively afford up to 2.1M and 2.5M placeable modules in the mixed-size placement context. The ISPD11 suite [25] is derived from industrial ASIC designs and aims at routability-driven placement; it goes beyond earlier placement benchmarks by introducing non-rectangular fixed macros and associated pins that reside on metal layers, with up to 1200K modules (standard cells, macros and IO pins). For gate sizing and Vt-swapping, the ISPDB12 benchmark suite [18] adds library timing models (.lib, or Liberty table model format [36]) for a cell library with 11 combinational cells (each with 5 Vt variants and 10 sizes) and one sequential cell, along with a simplified SPEF with a single lumped capacitance for each net. The ISPD13 suite [19] adds more detailed RC modeling and incorporates an industry timer in the evaluation. Instance complexity reaches 982K instances.

Artificial Benchmark Suites. Previous artificial benchmark generation approaches include circegen [13], gpl [22] and the work of [11]. A valuable class of methods produces instances with unknown optimal solutions. The PEKO placement benchmark generator [7] achieves a net-degree distribution similar to (ISPD98) IBM netlists as well as a constructive placement solution with known minimum wirelength. To improve realism (PEKO benchmarks have a single cell size, and all nets are local), PEKU [9] generates instances with known upper bounds on optimal wirelength. Nets in PEKU instances are long; a hybrid of PEKO and PEKU allows users to specify the percentage of short nets in the benchmarks. Generation...
of artificial instances with known optimal solutions has also been achieved for gate sizing optimizations [12]; an extension in [15] produces instances that resemble real designs in terms of gate count, path depth, fanin/fanout distribution and Rent parameter.

3. CHALLENGES

We now discuss challenges of horizontal benchmark extension, focusing on recent ISPD suites and actual designs. The most obvious challenge in benchmark extension, arising from IP protection and limited scope of target problem formulations, is that benchmarks typically omit information. Partitioning instances (ISPD98) omit cell sizes and signal directions; placement instances (ISPD06/11) omit/obfuscate cell functions and combinational-sequential distinctions; global routing instances (ISPD07/08) omit cell functions and pin locations; etc. Thus, we must make a number of judgment calls as to how to best fill in missing information to achieve “benchmark extension”. To (i) enable academic and industry optimizers to be run on the same testcases, and (ii) extend placement benchmarks to sizing benchmarks, we are faced with many options. These include, for example, criteria for mapping a placeable cell in a placement benchmark to a timable cell in a sizing benchmark; setting of timing, max fanout and other constraints; creation of interconnect parasitics; etc. The exemplary issues shown in Table 1 are addressed in the next three subsections.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Missing logic function information in ISPD11 benchmarks</td>
</tr>
<tr>
<td>A2</td>
<td>Need timable benchmarks with parasitic information for sizing</td>
</tr>
<tr>
<td>B1</td>
<td>Commercial tools handle richer constraints and design rules</td>
</tr>
<tr>
<td>B2</td>
<td>ISPD213 technology does not provide LEF file</td>
</tr>
<tr>
<td>C1</td>
<td>Commercial users require timing-feasible benchmarks</td>
</tr>
<tr>
<td>C2</td>
<td>Granularity of libraries varies across different technologies</td>
</tr>
</tbody>
</table>

3.1 Formats, Data Models, and Libraries

We illustrate horizontal benchmark extension using selected instances from the ISPD11, ISPD12 and ISPD13 benchmark suites, along with two designs from the OpenCores website [37]. The first sequential is different; the ISPD98 benchmark is shown in Table 1. ISPD11 benchmarks are in Bookshelf format [46]. ISPD12/13 benchmarks are .v netlists (i.e., structural Verilog), and real designs are described as RTL. Further, cell function information is removed from ISPD11 benchmarks. To enable horizontal assessment, our solution maps all benchmarks to .v netlists, which enables us to synthesize real implementations in arbitrary technology libraries; for ISPD11 benchmark circuits, we map nodes to cells in a given target technology. Apples-to-apples assessment in the P&R domain then requires us to also generate DEF [35] by performing floorplanning, power planning and placement of primary inputs and outputs.

<table>
<thead>
<tr>
<th>Design stage</th>
<th>Tool</th>
<th>Required file format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Placement</td>
<td>Commercial</td>
<td>v. lib (DEF), LEF</td>
</tr>
<tr>
<td></td>
<td>Academic</td>
<td>nodes, nets, wts, pt. scl. shapes</td>
</tr>
<tr>
<td>Global routing</td>
<td>Commercial</td>
<td>v. lib, DRC (LEF, DEF)</td>
</tr>
<tr>
<td></td>
<td>Academic</td>
<td>gr or nodes, nets, pt. scl. shapes, route</td>
</tr>
<tr>
<td>Sizing</td>
<td>Commercial</td>
<td>v. lib, DEF, LEF, SPEF, sdc</td>
</tr>
<tr>
<td></td>
<td>Academic</td>
<td>v. lib, SPEF, sdc</td>
</tr>
</tbody>
</table>

A second basic challenge in horizontal extension is that many academic tools are “hard-wired” to particular technology definitions. When assessing “legacy” tools that are no longer under active development, extra stp of enablement are required to migrate benchmarks across multiple technologies. For example, different cell libraries might vary in granularity (number of cell sizes, number of Vt flavors), available logic functions, or naming conventions, and this makes technology migrations not so straightforward. Figure 3 depicts our flow to extend benchmarks horizontally across multiple technologies. Explanations of sample issues (shown in Table 1), and our corresponding approaches, are as follows.5

5In our experience, horizontal extension of artificial, as opposed to real, netlists does not bring any fundamentally different challenges. Thus, while our discussion below focuses on real instances, it is largely orthogonal to the real vs. artificial dichotomy.

Figure 3: Flow to extend benchmark circuits across technologies.

**Issue A1:** In ISPD11 benchmarks, logic function information is removed and only node (i.e., cell, macro, pin) sizes and connectivity information are provided. To address this issue, our approach maps nodes of a placement benchmark to cells in a given Liberty/LEF pair, based on cell pin count and cell width. We first determine sequential cells. Then we map other nodes to combinational cells in the given LEF based on cell width and pin count. We normalize widths of nodes with the same pin count in the benchmarks to a particular range (e.g., [0, 1]). Then, we normalize cells with the corresponding pin count to the same range. Based on the normalized width values, we randomly assign cells from Liberty to nodes of the ISPD11 benchmark. Since we do not consider design functionality during cell mapping, logic redundant can result, and we therefore use Synopsys DC Compiler [42] to simplify the netlist with Boolean transforms. When we migrate a resulting benchmark to another technology, we preserve functionality but scale footprint accordingly.

**Issue A2:** Timing paths are not considered in placement benchmarks. For instance, there are many floating nets (i.e., driving cell information is missing), notably in ISPD11 benchmarks, which lead to unconstrained timing paths. In addition, parasitic information is missing in placement benchmarks. Our approach adds additional primary inputs, to which we connect the floating nets. We determine the number of additional primary inputs based on Rent’s rule (we use a Rent exponent value of 0.55 in the implementations reported below), and distribute floating nets evenly to the additional primary inputs. Further, we perform low-effort placement and routing and extract parasitic information from the routed designs.

3.2 Enablement of P&R Assessments

Figure 4 shows our enablement of P&R assessments. The inputs of the standard industry flow are LEF, DEF (or .v) and Liberty files. Conversion between LEF/DEF and Bookshelf formats enables assessment across commercial and academic tools. We implement placement with both commercial and academic placers; we then perform global routing on the resultant placement solutions using both commercial and academic tools. Detailed routing is feasible only with commercial tools. To enable apples-to-apples assessments across academic and commercial tools, we modify technology files and apply conversions between different formats. Explanations of sample issues (shown in Table 1), and our corresponding approaches, are as follows.

**Issue B1:** Commercial tools have multiple objectives and need to satisfy many design rules (e.g., antenna and maximum current density rules) and constraints (e.g., multi-mode/multi-corner timing, maximum fanout, etc.) while academic tools have only a specific objective. Our goal is to compare performance in terms of the specific objective, not to compare overall tool quality. In light of this goal, our approach intentionally drives the commercial tools to optimize for a specific objective that we want

6Given that area of a flip-flop is typically ∼5× the area of a NAND gate of similar driving strength, we bucket nodes having width of 25-32 units as flip-flops in ISPD11 benchmarks. Our identification of sequential cells has been confirmed by checking against a golden list of sequential cells provided by contest organizers [25].
Liberty, but a different number of sizes in a foundry Liberty. This would lead to less consistent results across technologies due to the changed solution space; thus, it is difficult to assess tools’ quality across technologies. To match the number of cell variants, our approach increases library granularity so that all different technologies have the same sizing solution space. We generate new cells by interpolating/extrapolating based on timing information (cell delay, output transition time) of existing cells, exploiting logical effort analysis for cells of each given type. Last, we approximate leakage power and pin capacitance values by fitting second-order models to the values of existing cells.

4. EXPERIMENTAL RESULTS

We experimentally validate our horizontal benchmark enablements in two ways: (i) P&R studies; and (ii) sizing studies. In each way, we first assess tools’ performance on different benchmarks, then, in different technologies. Last, we select the largest benchmark in each domain and perform maximal comparison, where we compare among different technologies and tools. Our studies use benchmark circuits with multiple sources and original purposes, as listed in Table 3. We use five distinct: ISPD12/13, and foundry 28nm FDSOI, 45GS, 65GP, and 90LP.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Name</th>
<th>Gate Count (P&amp;R)</th>
<th>Gate Count (Sizing)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPD11-1</td>
<td>des_gen</td>
<td>113112</td>
<td>113112</td>
</tr>
<tr>
<td>ISPD11-2</td>
<td>netcard</td>
<td>298228</td>
<td>596458</td>
</tr>
<tr>
<td>ISPD13-2</td>
<td>cordc</td>
<td>429093</td>
<td>429093</td>
</tr>
<tr>
<td>ISPD13-4</td>
<td>matrix_mult</td>
<td>156440</td>
<td>156440</td>
</tr>
<tr>
<td>ISPD12-1</td>
<td>b19</td>
<td>255268</td>
<td>255268</td>
</tr>
<tr>
<td>ISPD11-1</td>
<td>superblue1</td>
<td>871297</td>
<td>871297</td>
</tr>
<tr>
<td>ISPD11-2</td>
<td>superblue1</td>
<td>128084</td>
<td>128084</td>
</tr>
<tr>
<td>ISPD11-3</td>
<td>superblue18</td>
<td>467261</td>
<td>467261</td>
</tr>
<tr>
<td>Real-1</td>
<td>jpege_encoder</td>
<td>83241</td>
<td>83241</td>
</tr>
<tr>
<td>Real-2</td>
<td>ion5mp</td>
<td>479566</td>
<td>479566</td>
</tr>
</tbody>
</table>

4.1 P&R

We perform horizontal assessments using both academic and commercial P&R tools, across five technologies. Table 4 lists our experiments, where cPlacer1, cPlacer2 and cRouter1 are P&R functions (mapping not given here) in Cadence SoC Encounter vEDI11.1 [30] and Synopsys IC Compiler vH-2013.03-SP3 [41].

Expt 1 illustrates horizontal assessment across two commercial placer and one academic placer, using circuits from ISPD11/12/13 benchmark suites and real designs, with foundry 28nm technology. Results in Table 5 show that the academic tool achieves better HPWL, but consumes more runtime especially on large benchmarks. For a “fair comparison”, awareness of timing and electrical design constraints is disabled in the commercial tool, where these issues (timing, DRVs) are not yet well-considered by any academic placers.

Expt 2 assesses placement solutions of one commercial placer and one academic placer across five different technologies. Results in Table 6 again show the academic tool in most cases achieving less HPWL with larger runtime, consistently across technologies. Expt 3 illustrates horizontal assessment across two commercial and three academic placers, and across three distinct technologies, using the ISPD11-2 benchmark. Results in Table 7 show that solution quality is fairly consistent in the commercial tools, but varies more widely across the academic tools. More critically, the tool rankings that might be inferred using the ISPD technology are quite different from those that might be inferred in 28nm and 65nm technologies, which raises the possibility of greater suboptimality for academic tools in industry technologies.
Table 7: Expt 3. Comparison across placers. Benchmark: ISPD11-2. “-” indicates that no feasible solution is obtained within 48 CPU-hours.

<table>
<thead>
<tr>
<th>Tech</th>
<th>cPlacer1</th>
<th>mPl6</th>
<th>cPlacer2</th>
<th>NTUPlace3</th>
<th>mPl6</th>
<th>FastPlace3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime (min)</td>
<td>1330</td>
<td>1330</td>
<td>73500</td>
<td>73500</td>
<td>67500</td>
<td>72000</td>
</tr>
<tr>
<td>HPWL (mm)</td>
<td>50300</td>
<td>50300</td>
<td>48400</td>
<td>48400</td>
<td>48400</td>
<td>48400</td>
</tr>
<tr>
<td>Runtime (min)</td>
<td>328</td>
<td>328</td>
<td>32000</td>
<td>32000</td>
<td>32000</td>
<td>32000</td>
</tr>
</tbody>
</table>

Table 8: Expt 4. Integration of routers in assessment (at 28nm).

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Placer</th>
<th>router</th>
<th>Tech</th>
<th>Wl (min)</th>
<th>tOverview</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPD13-2</td>
<td>mPl6</td>
<td>cTower1</td>
<td>28nm</td>
<td>73.5</td>
<td>58.8</td>
</tr>
<tr>
<td></td>
<td>cPlacer1</td>
<td>BFC-R</td>
<td></td>
<td>47.9</td>
<td>47.7</td>
</tr>
<tr>
<td>ISPD13-3</td>
<td>mPl6</td>
<td>BFC-R</td>
<td>65nm</td>
<td>42.6</td>
<td>42.6</td>
</tr>
<tr>
<td></td>
<td>cPlacer1</td>
<td>CSEET1</td>
<td></td>
<td>31.2</td>
<td>31.2</td>
</tr>
</tbody>
</table>

Table 9: Apples-to-apples assessments in sizing domain.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Tool</th>
<th>Tech</th>
<th>Wl (min)</th>
<th>tOverview</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPD13-2</td>
<td>cSizer1</td>
<td>28nm</td>
<td>118.3</td>
<td>118.3</td>
</tr>
<tr>
<td>ISPD13-3</td>
<td>cSizer2</td>
<td>28nm</td>
<td>139.6</td>
<td>139.6</td>
</tr>
<tr>
<td></td>
<td>cSizer3, Trident, UFRGS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Last, to further exercise the horizontal benchmark enablement of Figure 4, and to incorporate global routers into our assessments, we run global routing (with identical ggrid definitions) using both commercial and academic tools. Inputs are placement solutions for the ISPD13-2 and ISPD13-3 testcases obtained using commercial and academic placers. Results in Table 8 show that global routing solutions have wirelength roughly consistent with HPWL of placement solutions. At the same time, we notice in the academic tool BFG-R some possible effects of a contest-induced focus on reduction of overflows: a de-emphasis of the wirelength metric might be the cause of longer wirelength (e.g., on the ISPD13-3 testcase) compared to the commercial router. Perhaps a more interesting aspect of this study is that it shows the wide-ranging possibilities from “maximal horizontal benchmark enablement”: a gate sizing testcase is mapped to a production 28nm FDSOI library, placed with both commercial and academic placers, and global-routed with identical ggrid structure by both commercial and academic global routers (!). Potential additional studies abound – e.g., in a future study we will vary the number of routing layers for both placement and global routing comparison.

4.2 Sizing

Table 9 shows our setup of sizing assessments. As with P&R, we enable apples-to-apples assessment of commercial and academic sizers across multiple benchmarks and technologies. cSizer1, cSizer2 and cSizer3 are the leakage optimization tool BizeMO v2013 [29], and leakage optimization functions in Synopsys IC Compiler vH-2013.03-SP3 [41] and Cadence SoC Encounter vED15.1 [30] (mapping not given here).

Expt 5 compares final leakage and runtime of one commercial and one academic sizer on a range of benchmark types (sizing-oriented benchmarks, placement-oriented benchmarks, and real designs) with 28nm foundry technology. Interconnect RC parasitics (SPEF) are generated after P&R, and the clock period constraint is 1.2 x the longest combinational path delay in the extracted and timed netlist.8 We have observed in results that

8The Real-1 benchmark, and the ISPD11-2 and ISPD11-3 instances derived from placement-oriented benchmarks, have “somehow odd” WNS values after leakage optimization, as a result of this methodology.

Expt 6 assesses a commercial (cSizer1) and an academic (UFRGS) sizer with four foundry technologies. Results in Table 11 show that cSizer1 is worse than UFRGS in both solution quality and runtime, when evaluated using the ISPD contest technology. On the other hand, with 28nm, 45nm and 65nm foundry technologies, cSizer1 achieves better solution quality with smaller runtime. The change in tool superiority across technologies, despite our enablement of identical sizing and multi-Vt solution space across technologies (recall issue C1 in Section 3.3), raises the possibility that the academic sizer is somehow specialized to the ISPD technology.

Expt 7 illustrates the horizontal assessment across three commercial and two academic sizers, and across three distinct technologies. Results in Table 12 show differences in ranking between the ISPD technology and industry technologies, which may indicate the potential for improvement of academic tools’ robustness.

5. CONCLUSIONS

In this work, we have proposed and implemented “horizontal benchmark extensions” to maximally leverage available benchmark testcases across multiple optimization domains. We enable new assessments of academic research at one or more design stages, within industrial tool/flow contexts, across multiple technologies, and across multiple types of benchmarks.

9Anecdotally, participants in the 2013 Gate Sizing Contest observed that the ISPD technology was unusual in many respects, notably the non-monotonicity of delay and leakage benefits across sizes such as X3 gates.

10We make two comments. (i) The version of UFRGS that we study, obtained from the tool’s authors, has a known inability to handle interconnect delay correctly; this can result in negative WNS values. The relative tool performance is similar across technologies, which suggests that testcases generated by our methodology are not biased to any particular technology; on the other hand, our SPEF generation may be especially challenging to the UFRGS binary. (ii) The results for cSizer1 are certainly unusually poor, but we have double-confirmed the reported numbers.
In the domains of P&R and gate sizing, we describe several challenges to horizontal benchmark enablement as well as our proposed solution approaches and methodologies. We demonstrate benchmark constructions that are mapped to five technologies and consumed by academic and commercial tools for placement, routing (both global and detailed) and sizing. Experimental results suggest that academic tools can outperform industry tools on very specific objectives, but that over-focusing on a single objective can incur penalties in the multi-objective, highly constrained optimizations that arise in practical VLSI physical design contexts. Our results also point out that (i) academic tools can scale more poorly than commercial tools, and that (ii) the rank-ordering of tools by benchmark outcomes can be highly sensitive to choice of testcases and technology.

Our ongoing work pursues further horizontal benchmark constructions, e.g., to encompass clock network synthesis (ISPD09/10) and routability-driven placement (ICCAD13) benchmark suites while preserving their relevant characteristics. Connecting legacy methods for artificial testcase generation to current tool flows and formats is also of interest. Moreover, we seek benchmark constructions that can create more challenging, realistic benchmarks (e.g., benchmarks that explicitly test the ability to handle multiple objective and constraint types). Last, we believe that horizontal benchmark enablement can bring better exploration of the gaps between academic optimizers and real-world design contexts: certainly, improved understanding of "where things break" (cell counts, obstacles, aspect ratios, utilizations, library density, design rules, RC and signoff corners, etc.) can only help guide academic research.

6. ACKNOWLEDGMENTS

We are grateful to the authors of the academic tools studied [6] [8] [16] [17] [20] [26] for providing binaries of their optimizers for use in our study.

7. REFERENCES

[38] Research Data Alliance, http://rd-alliance.org