

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, netlist formats, technology files, library granularity, etc. 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/routers, 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 of

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.

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 such maximal assessment would reveal tools’ suboptimality, and thus guide the improvements of tools’ quality. Such “horizontal” evaluations are usually blocked by gaps between data models and formats of academic benchmark suites, versus those used in industry CAD tool flows.¹ Many benchmarks are constructed for particular technologies with specific library [36] granularity and naming conventions, which limits assessment. Underlying problem formulations may be mismatched to industry use cases, further hampering assessment.

In this work, we pursue the goals of *horizontal benchmarks* and *benchmark extension*, which together seek to maximize “apples-to-apples” assessment at one or more particular design stages, across different benchmarks, technologies, and tools. We use sizing and P&R (placement and routing), which are the topics of recent ISPD contests, to illustrate the challenges of, and our resulting methodologies for, horizontal benchmark enablement. For **benchmarks**, we report transformed sizing-oriented benchmarks (i.e., ISPD12/13 [18] [19]), placement-oriented benchmarks (i.e., ISPD11 [25]) and real designs (from OpenCores [37]). For **technologies**, we show mappings across ISPD12/13 contest and 28/45/65/90nm foundry technologies. Given the resulting horizontal benchmark suite, for **tools** we demonstrate the feasibility of apples-to-apples assessment among two academic sizers and three commercial tools in the sizing domain, and among four academic P&R tools and three commercial tools in the P&R domain. Comparison to commercial tools allows a better assessment of academic tools’ capability. The scope of our efforts is depicted in Figure 1. The website [28] gives all conversion scripts, tool runscripts, and horizontal benchmark datasets that we describe in this paper.

We make several high-level observations. First, our work does not simply convert data formats to be used across different tools. Rather, we address at a number of levels the key challenge of horizontal benchmark enablement, namely, how missing information can be reasonably filled in, and/or which

¹We recognize and applaud initiatives such as OpenAccess [39] and the now-inactive OAGear [40]. These data model and infrastructure projects offer the promise of universal data model and ‘star topology, rather than clique topology’ of interfaces and converters. However, given the long-standing incompleteness of open data models (e.g., with respect to timing flows) as well as the small number of key targets (ISPD benchmark formats, LEF/DEF and Verilog standards) we take a less elegant, and more pragmatic and brute-force, approach to achieving the desired enablement.

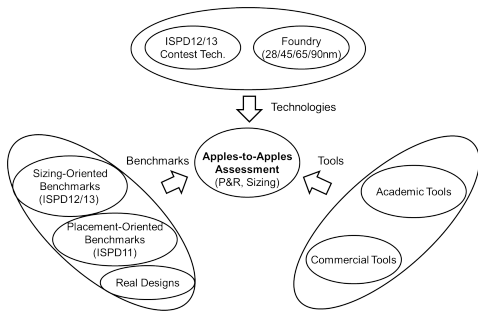


Figure 1: Scope of this work. We enable extensive assessment across different technologies, benchmarks and tools.

information should be simplified or hidden from tools, such that useful studies become possible. Second, the deeper contribution of our work is in enabling new questions to be explored: Can we better assess academic solver quality and scalability, in order to better assess potential gaps between the leading edge of academic research and industry contexts? Third, we emphasize that throughout our paper we use the term “benchmark” as a noun, and not as a verb. Our work is in the same spirit as OAGear [40], the GSRC Bookshelf [46] and works such as [5] – i.e., we hope that horizontal benchmarks will help industry and academia identify the most fruitful targets for academic research, as well as the potential impact of new academic research results.²

Our contributions may be summarized as follows.

- We propose and demonstrate *horizontal benchmarks* 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 related 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 $22\times$ since 1998, while over the same 15-year period the gate count of the largest benchmark netlists has increased by $12\times$; there is currently still a “ $1000\times$ ” gap (indicated by the scale difference

²We do not advocate “benchmarking” (the verb) or any other activity that is in violation of commercial tool licenses.

between two y-axes). More realistically, the gap between academic benchmark and real design complexities can be estimated (based on gate count) at $5\times \sim 20\times$, 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.

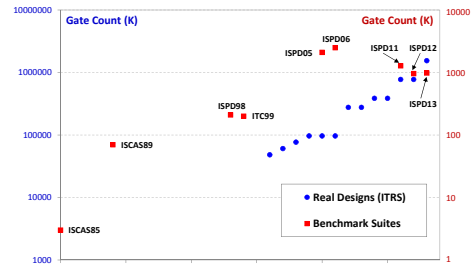


Figure 2: Gate count trajectories of largest MPU products [33] and largest designs in benchmark suites. Data in blue use the left y-axis, and data in red use the right y-axis.

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 ISCAS-89 benchmark suite has $\sim 70K$ gates and $\sim 3K$ flip-flops. The **ISPD98** benchmark suite [2], developed for netlist partitioning applications, includes 18 circuits with module counts up to $\sim 210K$. 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 $\sim 200K$ gates and $\sim 7K$ 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 **ISPD12** benchmark suite [18] adds library timing models (.lib, or Liberty table model format [36]) for a cell library with 11 combinational cells (each with 3 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 *circ/gen* [13], *gnl* [22] and the work of [11]. A valuable class of methods produces instances with *known 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

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 sweetspot) of EDA tools, which has slowed in recent years.

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.⁴ 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 1: Sample issues in horizontal benchmark enablement.

Issue	Summary
A1	Missing logic function information in ISPD11 benchmarks
A2	Need timable benchmarks with parasitic information for sizing
B1	Commercial tools handle richer constraints and design rules
B2	ISPD12/13 technology does not provide LEF file
C1	Commercial sizers require timing-feasible benchmarks
C2	Granularity of libraries varies across different technologies

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 challenge is different formats (see Table 2): 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 2: Data formats/models for ISPD benchmark types

Design stage	Tool	Required file format
Placement	Commercial	.v, .lib, (DEF), LEF
	Academic	.nodes, .nets, .wts, .pl, .scl, .shapes
Global routing	Commercial	.v, .lib, DEF, LEF
	Academic	.gr or .nodes, .nets, .pl, .scl, .shapes, .route
Sizing	Commercial	.v, .lib, DEF, LEF, SPEF, .sdc
	Academic	.v, .lib, SPEF, .sdc

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 stpfd 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

⁴In 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.

⁵Due to space constraints, more complete documentation is given at [28].

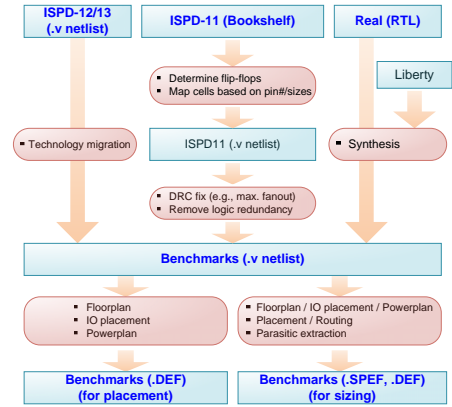


Figure 3: Flow to extend benchmark circuits across technologies.

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.⁶ We then 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 redundancy 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 $\sim 5\times$ 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].

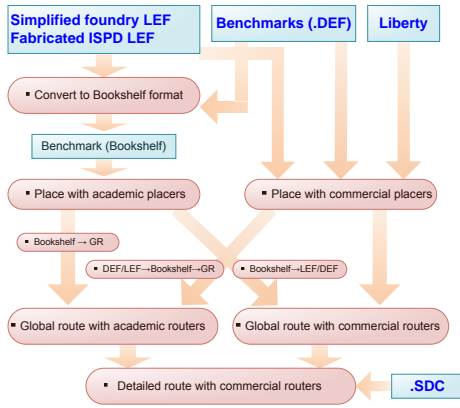


Figure 4: Enablement flows for horizontal P&R assessment.

to evaluate, by removing various extraneous rules that are defined in the LEF file. We use the simplified LEF file in placement and routing with both academic and commercial tools.

Issue B2: No technology (LEF) file is provided with ISPD12/13 benchmarks. To enable P&R of ISPD12/13 benchmarks using commercial tools, **our approach** constructs a new LEF file that incorporates technology information (e.g., metal pitch, width) from the foundry LEF. To generate cell LEF, we extract the pin area of X1 cells in the foundry LEF; based on this, we generate rectangular pins with the same area and height. Currently, we only distribute pins evenly inside each cell – then, based on the generated X1 cells, we scale width, pin area and on-grid pin locations linearly with drive strength to derive the LEF for larger cells.⁷

3.3 Enablement of Gate Sizing Assessments

We also enable horizontal evaluation across academic and industry tools for gate sizing (i.e., post-routing leakage reduction), as depicted in Figure 5. The cell sizing/Vt-swapping optimization reduces leakage *while preserving a timing signoff*. While commercial tools can consider, e.g., an area increase constraint, to achieve a fair assessment we only study tools in a pure leakage minimization use context. Inputs to sizing tools are netlist (.v), interconnect parasitics (SPEF), timing constraints (.sdc) and timing/power Liberty (.lib).

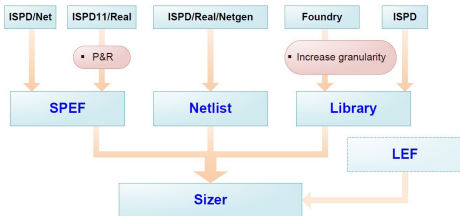


Figure 5: Enablement flows for horizontal sizer assessment.

Issue C1: Academic tools developed for the ISPD12/13 gate sizing contests must perform timing legalization as well as leakage minimization with a fixed set of SPEF parasitics, since no timing-feasible solution is provided. On the other hand, the use model for commercial “post-route leakage recovery” tools is to preserve a timing signoff (with fixed SPEF from a complete detailed route) while minimizing leakage. In other words, the industry tools assume a starting timing-feasible solution. For a fair assessment, we obtain timing-feasible solutions for all testcases. **Our approach** uses the academic tool [17] to perform timing recovery and changes .sdc files to generate timing-feasible solutions.

Issue C2: For assessment across different technologies, we would like to ensure that input netlists and sizing/Vt solution spaces are preserved across technologies. Varying library granularity poses a challenge, e.g., there are 10 sizes of inverters in ISPD12/13

⁷Particularly for mapping to advanced (≤ 28 nm) foundry technologies, we recognize the need to improve awareness of porosity, pin accessibility, and related considerations.

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 3: Benchmark circuits.

Benchmark	Name	Gate Count (P&R)	Gate Count (Sizing)
ISPD13-1	des_perf	113112	113112
ISPD13-2	netcard	982258	982258
ISPD13-3	cordic	42903	42903
ISPD13-4	matrix_mult	156440	156440
ISPD12-1	b19	219268	219268
ISPD11-1	superblue1	817297	651533
ISPD11-2	superblue12	1286948	895309
ISPD11-3	superblue18	467261	385882
Real-1	jpeg_encoder	83241	83241
Real-2	leon3mp	473986	473986

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 vEDI13.1* [30] and *Synopsys IC Compiler vH-2013.03-SP3* [41].

Table 4: Apples-to-apples assessments in P&R domain.

	Benchmark	Tech	Tool
Expt 1	ISPD13-{1-4}, ISPD12-1, ISPD11-{1-3}, Real-{1-2}	28nm	cPlacer1, mPL6
Expt 2	ISPD13-2, ISPD11-2	ISPD, 28/45/65/90nm	cPlacer1, mPL6
Expt 3	ISPD11-2	ISPD, 28/65nm	cPlacer1, cPlacer2, cPlacer3, NTUPlace3, mPL6, FastPlace3.1
Expt 4	ISPD13-2	28nm	cPlacer1, mPL6, cRouter1, BFG-R,

Expt 1 assesses solution quality (HPWL) and runtime of one 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.

Tech	cPlacer1		cPlacer2		NTUPlace3		mPL6		FastPlace3.1	
	HPWL (mm)	Runtime (min)	HPWL (mm)	Runtime (min)	HPWL (mm)	Runtime (min)	HPWL (mm)	Runtime (min)	HPWL (mm)	Runtime (min)
ISPD	50300	263	46100	103	47300	1330	73300	32	40400	88
28nm	36400	328	41500	29	-	-	32000	1212	48200	130
65nm	58800	335	65000	30	-	-	51100	657	73200	141

Table 5: Expt 1. Placer assessment across benchmark circuit types.

Benchmark	cPlacer1		mPL6	
	HPWL (mm)	Runtime (min)	HPWL (mm)	Runtime (min)
ISPD13-1	1040	12	868	7
ISPD13-2	31800	185	35800	313
ISPD13-3	224	3	178	3
ISPD13-4	1478	18	1338	11
ISPD12-1	1535	34	1360	853
ISPD11-1	19660	159	18260	547
ISPD11-2	36400	328	32000	1212
ISPD11-3	13960	94	12140	282
Real-1	954	10	800	6
Real-2	29400	273	17600	268

Table 6: Expt 2. Placer assessment across technologies.

Benchmark	Tech	cPlacer1		mPL6	
		HPWL (mm)	Runtime (min)	HPWL (mm)	Runtime (min)
ISPD13-2	ISPD	47200	149	39900	177
	28nm	31800	185	35800	313
	45nm	33600	221	31200	185
	65nm	50200	273	43400	172
	90nm	67500	327	55400	148
ISPD11-2	ISPD	50300	263	40400	88
	28nm	36400	328	32000	1212
	45nm	42600	307	37100	781
	65nm	58800	335	51100	657
	90nm	78600	449	73800	719

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 starts to show 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 *BlazeMO v2013* [29], and leakage optimization functions in *Synopsys IC Compiler vH-2013.03-SP3* [41] and *Cadence SoC Encounter vED113.1* [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 $1.2 \times$ 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 “somewhat odd” WNS values after leakage optimization, as a result of this methodology.

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

Benchmark	Placer	GlobRouter	WL (mm)	%Overflow
ISPD13-2	mPL6	BFG-R	50.7	68.8
		cRouter1	48.0	71.8
	cPlacer1	BFG-R	47.9	47.1
		cRouter1	44.7	59.3
ISPD13-3	mPL6	BFG-R	0.68	0.0
		cRouter1	0.23	1.0
	cPlacer1	BFG-R	0.75	0.0
		cRouter1	0.27	1.1

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

	Benchmark	Tech	Tool
Expt 5	ISPD13-[1-4], ISPD12-1, ISPD11-[2-3], Real-1	28nm	cSizer1, UFRGS
Expt 6	ISPD13-2, ISPD11-2	ISPD, 28/45/65nm	cSizer1, UFRGS
Expt 7	ISPD13-2	ISPD, 28/65nm	cSizer1, cSizer2, cSizer3, Trident, UFRGS

academic sizers, in general, tend to spend more time and resources (e.g., memory), compared with the commercial sizers.

Table 10: Expt 5. Assessment of sizers on various benchmarks.

Benchmark	cSizer1			UFRGS		
	Leak (mW)	WNS (ns)	Runtime (min)	Leak (mW)	WNS (ns)	Runtime (min)
ISPD13-1	2.5	0.0	6.0	2.5	0.0	9.3
ISPD13-2	27.8	0.5	64.0	27.7	-3.7	73.5
ISPD13-3	0.4	0.0	6.8	0.4	0.0	3.0
ISPD13-4	1.1	0.0	13.5	1.0	-0.1	18.4
ISPD12-1	2.1	0.4	21.9	2.1	-0.2	28.0
ISPD11-2	30.4	2.7	155.0	30.4	-14.8	241.1
ISPD11-3	25.9	118.3	48.5	25.9	136.8	106.0
Real-1	1.4	0.0	4.2	1.3	-0.3	7.2

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 65m 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.⁹

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.¹⁰

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 cSizer3 are certainly unusually poor, but we have double-confirmed the reported numbers.

Table 12: Expt 7. Comparison across sizers. Benchmark: ISPD13-2.

Tech	cSizer1			cSizer2			cSizer3			Trident			UFRGS		
	Leak (mW)	WNS (ns)	Runtime (min)	Leak (mW)	WNS (ns)	Runtime (min)	Leak (mW)	WNS (ns)	Runtime (min)	Leak (mW)	WNS (ns)	Runtime (min)	Leak (mW)	WNS (ns)	Runtime (min)
ISPD	5231.6	-0.01	55.0	5591.5	0.0	31.6	3899.1	-125.4	80.5	5233.1	0.0	179.8	5184.1	-0.2	46.0
28nm	27.8	0.5	64.0	27.8	0.7	35.0	27.5	-851.2	98.0	29.4	1.4	43.7	27.7	-3.7	73.5
65nm	45.8	0.4	49.5	45.9	0.5	34.0	45.0	-283.9	104.5	46.0	1.2	46.8	45.4	-2.6	77.3

Table 11: Expt 6. Assessment of sizers across technologies.

Benchmark	Tech	cSizer1			UFRGS		
		Leak (mW)	WNS (ns)	Runtime (min)	Leak (mW)	WNS (ns)	Runtime (min)
ISPD13-2	ISPD	5231.6	-0.01	55.0	5184.1	-0.2	46.0
	28nm	27.8	0.5	64.0	27.7	-3.7	73.5
	45nm	35.9	1.2	77.5	35.5	-5.8	95.6
	65nm	45.8	0.4	49.5	45.4	-2.6	77.3
ISPD11-2	ISPD	7143.8	14.8	77.0	6341.8	16.6	192.0
	28nm	30.4	2.7	155.0	30.4	-14.8	241.1
	45nm	39.8	96.5	127.2	39.4	302.6	367.0
	65nm	50.2	25.8	67.5	50.1	-56.8	262.9

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 enable 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

- [1] S. N. Adya, M. C. Yildiz, I. L. Markov, P. G. Villarrubia, P. N. Parakh and P. H. Madden, “Benchmarking for Large-Scale Placement and Beyond”, *IEEE TCAD* 23(4) (2004), pp. 472-487.
- [2] C. J. Alpert, “The ISPD98 Circuit Benchmark Suite”, *Proc. ISPD*, 1998, pp. 80-85.
- [3] F. Brglez, D. Bryan and K. Kozmiński, “Combinational Profile of Sequential Benchmark Circuits”, *Proc. ISCAS*, 1989, pp. 1929-1933.
- [4] F. Brglez and H. Fujiwara, “Recent Algorithms for Gate-Level ATPG with Fault Simulation and Their Performance Assessment”, *Proc. ISCAS*, 1985, pp. 663-698.
- [5] A. E. Caldwell, A. B. Kahng, A. A. Kennings and I. L. Markov, “Hypergraph Partitioning for VLSI CAD: Methodology for Heuristic Development, Experimentation and Reporting”, *Proc. DAC*, 1999, pp. 349-354.
- [6] T. F. Chan, J. Cong, J. R. Shinnerl and K. Sze, “mPL6: Enhanced Multilevel Mixed-Size Placement”, *Proc. ISPD*, 2006, pp. 212-214.
- [7] C. C. Chang, J. Cong and M. Xie, “Optimality and Scalability Study of Existing Placement Algorithms”, *Proc. ASP-DAC*, 2003, pp. 621-627.
- [8] T.-C. Chen, T.-C. Hsu, Z.-W. Jiang and Y.-W. Chang, “NTUPlace: A Ratio Partitioning Based Placement Algorithm for Large-Scale Mixed-Size Designs”, *Proc. ISPD*, 2005, pp. 236-238.
- [9] J. Cong, M. Romesis and M. Xie, “Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms”, *Proc. ISPD*, 2003, pp. 88-94.
- [10] F. Corno, M. S. Reorda and G. Squillero, “RT-Level ITC’99 Benchmarks and First ATPG Results”, *IEEE Design & Test of Computers* 17(3) (2000), pp. 44-53.
- [11] J. Darnauer and W. W.-M. Dai, “A Method for Generating Random Circuits and Its Application to Routability Measurement”, *Proc. FPGA*, 1996, pp. 66-72.
- [12] P. Gupta, A. B. Kahng, A. Kasibhatla and P. Sharma, “Eyecharts: Constructive Benchmarking of Gate Sizing Heuristics”, *Proc. DAC*, 2010, pp. 592-602.
- [13] M. D. Hutton, J. Rose, J. P. Grossman and D. Corneil, “Characterization and Parameterized Generation of Synthetic Combinational Circuits”, *IEEE TCAD* 17(10) (1998), pp. 985-996.
- [14] C. Inacio, H. Schmit, D. Nagle, A. Ryan, D. E. Thomas, Y. Tong and B. Klass, “Vertical Benchmarks for CAD”, *Proc. DAC*, 1999, pp. 408-413.
- [15] A. B. Kahng and S. Kang, “Construction of Realistic Gate Sizing Benchmarks With Known Optimal Solutions”, *Proc. ISPD*, 2012, pp. 153-160.
- [16] A. B. Kahng, S. Kang, H. Lee, I. L. Markov and P. Thapar, “High-Performance Gate Sizing with a Signoff Timer”, *Proc. ICCAD*, 2013, pp. 450-457.
- [17] V. S. Livertamento, C. Guth, J. L. Güntzel and M. O. Johann, “Fast and Efficient Lagrangian Relaxation-Based Discrete Gate Sizing”, *Proc. DATE*, 2013, pp. 1855-1860.
- [18] M. M. Ozdal, C. Amin, A. Ayupov, S. M. Burns, G. R. Wilke and C. Zhuo, “ISPD-2012 Discrete Cell Sizing Contest and Benchmark Suite”, *Proc. ISPD*, 2012, pp. 161-164. http://archive.sigda.org/ispd/contests/12/ispd2012_contest.html.
- [19] M. M. Ozdal, C. Amin, A. Ayupov, S. M. Burns, G. R. Wilke and C. Zhuo, “An Improved Benchmark Suite for the ISPD-2013 Discrete Cell Sizing Contest”, *Proc. ISPD*, 2013, pp. 168-170. http://www.ispd.cc/contests/13/ispd2013_contest.html.
- [20] J. A. Roy and I. L. Markov, “High-performance routing at the nanometer scale”, *Proc. ICCAD*, 2007, pp. 496-502.
- [21] L. Srivani and V. Kamakoti, “Synthetic Benchmark Digital Circuits: A Survey”, *IETE Technical Review* 29(6) (2012), pp. 442-448.
- [22] D. Stroobandt, P. Verplaetse and J. V. Campenhout, “Generating Synthetic Benchmark Circuits for Evaluating CAD Tools”, *IEEE TCAD* 19(9) (2000), pp. 1011-1022.
- [23] D. Sylvester and K. Keutzer, “Getting to the Bottom of Deep Submicron”, *Proc. ICCAD*, 1998, pp. 203-211.
- [24] N. Viswanathan, IBM Corporation, *personal communication*, November 2011.
- [25] N. Viswanathan, C. J. Alpert, C. Sze, Z. Li, G.-J. Nam and J. A. Roy, “The ISPD-2011 Routability-Driven Placement Contest and Benchmark Suite”, *Proc. ISPD*, 2011, pp. 141-146. http://www.ispd.cc/contests/11/ispd2011_contest.html.
- [26] N. Viswanathan, M. Pan and C. Chu, “FastPlace 3.0: A Fast Multilevel Quadratic Placement Algorithm with Congestion Control”, *Proc. ASP-DAC*, 2007, pp. 135-140.
- [27] X. Yang, M. Wang, K. Eguro and M. Sarrafzadeh, “A Snap-On Placement Tool”, *Proc. ISPD*, 2000, pp. 153-158.
- [28] Horizontal Benchmarks Project Website. <http://vlsicad.ucsd.edu/A2A>
- [29] Blaze MO. <http://www.tela-inc.com>
- [30] Cadence Design Systems. <http://www.cadence.com>
- [31] ISPD05/06 benchmarks. <http://archive.sigda.org/ispd2005/contest.htm>, <http://archive.sigda.org/ispd2006/contest.html>
- [32] ISPD website. <http://www.ispd.cc>
- [33] ITRS 2011 edition reports. <http://public.itrs.net/reports.html>
- [34] IWLS contest. <http://www.iwls.org/challenge/index.html>
- [35] LEF DEF reference. <http://www.si2.org/openeda.si2.org/projects/lefdef>
- [36] Liberty Technical Advisory Board. <http://www.opensourceliberty.org>
- [37] OpenCores. <http://opencores.org>
- [38] Research Data Alliance. <http://rd-alliance.org>
- [39] Si2 OpenAccess. <http://www.si2.org/?page=69>
- [40] Si2 OpenAccess Gear. http://www.si2.org/openeda.si2.org/help/group_ld.php?group=73
- [41] Synopsys. <http://www.synopsys.com>
- [42] Synopsys Design Compiler User Guide. <http://www.synopsys.com/Tools/Implementation/RTLsynthesis/DCUltra/Pages>
- [43] Synopsys PrimeTime User’s Manual. <http://www.synopsys.com>
- [44] TAU contest. <http://www.tauworkshop.com>
- [45] UCLA/UCSD Sizing Optimizers. <http://vlsicad.ucsd.edu/SIZING/optimizer.html>
- [46] VLSI CAD Bookshelf, A. E. Caldwell, A. B. Kahng and I. L. Markov. <http://vlsicad.eecs.umich.edu/BK>