

Reporting of Standard Cell Placement Results

Patrick H. Madden SUNY Binghamton Computer Science Department
pmadden@cs.binghamton.edu <http://vlsicad.cs.binghamton.edu>

ABSTRACT

VLSI fabrication technology has advanced rapidly, bringing with it a strong demand for faster and better design automation tools. Accurate reporting of results for placement approaches is crucial to the development of improved automation tools; unfortunately, publicly available placement benchmarks are outdated, and there are wide variations in their interpretation.

At best, poor benchmarks and differences in interpretation result in misunderstandings of the effectiveness of some approaches. At worst, they can motivate research in areas of very little promise, while other areas which have true potential are ignored. In this paper, we describe current standard cell placement benchmarks and illustrate common differences in their interpretation, in the hope that a clear understanding of these issues will allow research to be directed and evaluated more efficiently.

1. INTRODUCTION

Standard cell placement is a fundamental problem in VLSI computer aided design. Objectives such as wire length and area minimization have long been a concern, and with the advent of deep submicron design, the scope of the problem now includes delay optimization, power minimization, and a number of other issues.

Many approaches to the placement problem have been proposed, and a set of well known benchmark circuits are widely available. Unfortunately, there is wide variation in the interpretation of these benchmarks, making comparison of results difficult or impossible.

In this paper, we attempt to classify the common interpretations, and bring together reported results from a number of authors into a unified table. The large variation in results make it clear that there are fundamental differences in how measurements are taken. Contrary to what might be anticipated for a problem this well studied, there are few “common assumptions.”

This paper has a motivation similar to that of [2], which illustrated difficulties in hypergraph partitioning research, due to variation in implementation of “standard” algorithms. The challenges faced with circuit placement are substantially more difficult than those observed with hypergraph partitioning: there is not even agreement on how a result should be measured, to say nothing of how a

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISPD’01, April 1-4, 2001, Sibina, California, USA.
Copyright 2001 ACM 1-58113-347-2/01/0004 ...\$5.00.

“standard” algorithm should be implemented.

As with [2], we observe that the lack of consensus on interpretations hampers placement research. In many published works, there are unintentional comparisons of “apples” to “oranges,” resulting in incorrect or misleading conclusions. If we hope to promote the best algorithms and approaches, we must have a clear understanding of what “best” means. At the most fundamental level, if we hope to derive any benefit from research results, we must know what the results imply.

The remainder of the paper is organized as follows. We first provide a bit of historical perspective on the benchmarks themselves, followed by a discussion of the metrics used to evaluate placement approaches. Our focus is primarily on wire length based metrics, but we consider delay optimization, area minimization, parasitics, and several other issues which can influence evaluations. We then present a table which brings together reported results from a number of earlier works. We conclude this paper with a number of suggestions that should provide for less ambiguity in the evaluation of placement methods, allowing research to be directed more efficiently and effectively.

2. STANDARD CELL PLACEMENT BENCHMARKS

The MCNC benchmark suite was released in the early 1990s, with a number of standard cell circuits being made available in the YAL and VPNR formats. Several translators were available, allowing conversion into EDIF, and academic TimberWolf formats. Later, other translators allowed conversion into a variety of formats, including PROUD, commercial versions of TimberWolf, Cadence LEF/DEF, and the GSRC Bookshelf formats. While not part of this original group of benchmarks, IBM’s *golem3* has also become a staple in placement research. These circuits are by far the most commonly used benchmarks in standard cell placement research.

In [15], results of early placement approaches on the benchmark circuits were summarized. In this work, only *circuit areas* were reported; the circuits were placed and routed, with the figure of merit being total area. While it was suggested that subsequent research should report circuit areas (and in particular, for placements with a variety of aspect ratios), most work reports instead *half perimeter wire lengths*.

The transition from reporting of area to reporting of half perimeter wire length is understandable. Global and detail routing are generally time consuming, and variation in the quality of the routing tools can skew comparisons. Routing is now rarely performed (for reporting of placement results), and half perimeter wire length is widely accepted as a relevant metric.

The MCNC benchmarks were designed utilizing fabrication parameters that were current at the time. The advance of fabrication

has made many of the fundamental assumptions used in these designs inappropriate for modern design.

- In early fabrication processes, only two metal layers were available for routing, requiring additional “channel” space between active circuit elements. With current processes, more metal layers are available, and most routing can occur “over the cell.”
- For the early benchmarks, a simple RC delay model is suggested. This is clearly inadequate; most groups now use either *Elmore*[6] delay, or an approach based on asymptotic waveform evaluation[17].
- The benchmarks are small compared to current circuits, with the larger MCNC benchmarks (*avqsmall*, *avqlarge*) being half the size of IP blocks expected for system-on-a-chip design.
- Modern issues such as power minimization, crosstalk minimization, and signal integrity, are not considered at all.

Industry groups have been reluctant to release new benchmarks, with competitive advantage being a primary concern. As a result, researchers in academia are left with a dilemma. If the MCNC placement benchmarks are used as designed, relevance of results to modern objectives is questionable. If proprietary (industry) benchmark results are reported, other groups will be unable to perform comparable experiments; there is no way of knowing if reported results are good, bad, or indifferent. If the MCNC placement benchmarks are scaled or adapted to modern fabrication technologies, comparison of results to previous work will be hopelessly skewed.

3. REPORTING OF RESULTS

In this section, we illustrate common benchmark interpretations, which are a large factor in the variation in reported results. In some cases, we have been unable to determine what interpretations were used to produce some reported results: the details we are concerned with here are rarely documented in the published work, and due to time constraints, we were unable to contact all authors directly. We consider first those issues which impact wire length estimates, and then address performance optimization concerns.

3.1 Wire Length Metrics

While wire length might seem to be a relatively simple quantity to measure, there are a surprising number of issues where results can be skewed substantially.

3.1.1 Differences in Translation

With each translation, there is the possibility that information can be lost. In most cases, the numbers of cells and nets are preserved. The naming scheme, however, is not always preserved, making determination of the use of some nets unclear. Errors in translation may be subtle, and are therefore difficult to identify. In some reported work, we observe the following.

- The number of cells or nets in a benchmark varies from the specifications of the original circuits. This clearly indicates that the results cannot be compared to previously published work.
- Names of nets and cells are sometimes lost in translation. Clearly, nets named *Vdd*, *Vss*, *Reset*, *Clock*, *Scan*, *Phi1*, or *Phi2*, should receive different treatment than ordinary nets, and may be critical to proper determination of circuit delay.

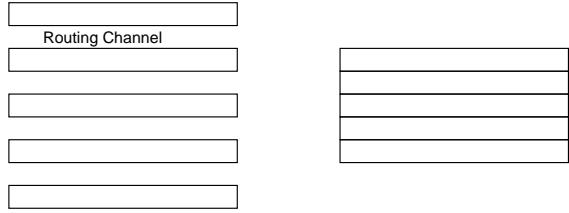


Figure 1: In early fabrication processes, additional space between cell rows is required to complete routing. With modern fabrication, over the cell routing allows the elimination of space between cell rows in most cases. We can expect significantly reduced wire lengths for modern fabrication processes, even if we account for feature size scaling.

3.1.2 Scaling of Benchmark Dimensions

To enable fast cell mirroring, early versions of the TimberWolf placement tool required cell dimensions to be even integers. To handle this requirement, the MCNC benchmarks *fract*, *struct*, and *biomed* have their dimensions doubled in the TimberWolf formats, while *golem3* has dimensions multiplied by four.

The result of this scaling is that in some cases, results reported for these benchmarks may differ by a factor of two or four depending on what input files are used. This has clearly occurred: in [19], the result for *golem3* is reported as 88.98, while [24] reports a result of 19.84, and scales the [19] result to 22.60.

3.1.3 Row Spacing and Numbers of Rows

Perhaps the most significant difference in interpretation has been with row spacings and the numbers of rows used in standard cell placement. When the MCNC benchmarks were first released, relatively few metal layers were available for routing; thus, *channel based design* is appropriate, and spacing between cell rows is required. With current fabrication techniques, *over the cell* routing is possible, and additional spacing can be eliminated in most cases.

Changes to row spacing result in a substantial impact to estimated placement wire lengths. In Figure 1, a simple 5 row placement is shown; by removing spacing between rows, we reduce the lengths of nets which span more than one row. Both interpretations of row spacing are reasonable, and both occur within the literature. In our experiments, we observe that it is quite possible to have as much as a 30% reduction in wire lengths simply by removing inter-row spacing.

If row spacings are not determined precisely by complete routing, many groups assume spacing is equal to standard cell height. In [13][19][12], results reported are with routing between cell rows. In [5], results are obtained with row spacing equal to cell height. In [10][24][11], no space is assumed between rows.

There is also little consensus on the number of rows to use for the benchmarks. This has substantial impact on results: if we consider a circuit which forms an 8 by 8 mesh, a placement into 8 rows is clearly superior to either a 7 or 9 row solution. There is no way to *a priori* determine an appropriate number of rows, or to determine the impact of this decision on total wire length. For the benchmark *primary2*, the number of rows used have included 29[15], 36[14], 28[19][5][7][11][24], and 32[18]. The benchmark *primary2* is not unusual; there are a variety of row numbers used for the other benchmarks as well.

3.1.4 Pad Positioning

A benchmark feature frequently obscured in translations between formats is the placement of input and output pads. In some cases,

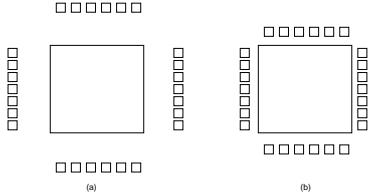


Figure 2: Pad positions are not always preserved during translation from one format to another. If pads are shifted away from standard cell rows, this obviously increases expected wire length. If we assume modern fabrication techniques, placing the pads at the positions suggested in the original VPRN and YAL benchmark files would be extremely pessimistic.

the pads are brought near to the cell rows; in others, positions are determined by the row spacings expected (using routing channels).

As with row spacing, changes to pad positions can impact the reported wire length, particularly for small benchmarks where a large percentage of nets connect to pads. These differences are illustrated in Figure 2.

3.1.5 Pin Positions

Half perimeter wire length is the most common length-based metric used in research on standard cell placement. Like issues already mentioned, there are a number of reasonable methods to estimate half perimeter wire length, and instances of many of these are in use.

Pin positioning is more complex than it might initially appear. In most standard cell libraries, there are multiple *port* locations to allow connection to the transistor inputs and outputs. These ports may be on either the tops or bottoms of the cells, or follow a “center terminal” alignment. Some current detail routers remove pre-determined connection wiring, placing vias at any convenient and design-rule correct location.

The variation in the number of ports, and their locations, results in variations in half perimeter wire length estimation methods. In Figure 3, a number of reasonable methods are shown.

In Figure 3(a), the bounding box contains the pins of a net in their entirety; this is perhaps the most conservative estimate. In 3(b), the bounding box is determined by the centers of the cells, and ignores precise cell positions; this is a frequently used metric, which eliminates the impact of cell mirroring. In 3(c), the bounding box contains pins that might be included with a Steiner or Spanning Tree construction. In 3(d), the bounding box contains the lower left corner of each cell, producing an estimate that would be similar to that of 3(b), but with some differences depending on cell placements and sizes. Some placement tools use the first pin defined in the cell library to determine pin locations, resulting in the possibility that a number of equivalent placements could have slightly different wire length estimations.

Again, these metrics are all reasonable, but can result in differing wire length estimates. The metric shown in (b) might be most common, but the estimate of (c) might be more accurate.

3.1.6 Spanning and Steiner Tree Metrics

In some work ([10], for example), wire length estimates are based on Spanning or Steiner Tree constructions, resulting in an estimate that would be more accurate than half-perimeter. Half perimeter estimates could be considered “optimistic” for nets with more than three pins. If all other factors are equal, one would expect the highest wire lengths to be reported for metrics based on spanning trees,

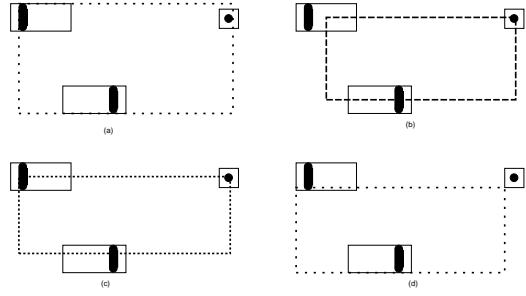


Figure 3: Estimation of half perimeter wire lengths can be done in a number of ways. In most cell libraries, there are multiple ports for the inputs and outputs of a cell; how these locations are considered, or if we only consider cell locations and ignore actual pins, impacts the computed wire length.

with Steiner tree metrics reducing lengths slightly, and half perimeter resulting in the lowest length estimates.

3.2 Other Metrics

While most of our focus is on issues that impact half-perimeter wire length measurement, we note that there is considerable interest in performance optimization (in terms of delay, power consumption, and signal integrity), as well as “practical” metrics such as run time and memory requirements.

3.2.1 Delay, Power, Signal Integrity

The initial MCNC circuits were designed for the technology parameters of the day. We focus in this section on delay optimization, but other modern concerns face similar challenges. Obviously, current performance driven research must consider current fabrication technology, resulting in difficulty in the comparison of results. If current work utilizes the earlier technology parameters, it has little relevance to modern design. If the work instead considers modern parameters, we can expect that wire lengths and delays would be substantially reduced. We note again that this issue has serious implications.

- If the original device dimensions are used, wire length results are comparable, but delay analysis will consider wire lengths greatly in excess of what could be reasonably expected.
- If device dimensions are scaled to modern technology, comparison to previous work is impossible.

As interconnect delay continues to increase in importance[4], these scaling issues become progressively more difficult. If we are to obtain reasonable and realistic evaluations of timing-driven placement approaches, we must scale feature sizes and transistor sizes carefully. The delay model suggested in [15] is clearly inappropriate for modern design; more complex delay models are required. In this area, it is likely that interpretation decisions will have far greater impact than any algorithmic choice we could make.

Beyond these scaling issues, the problem of simply determining the longest critical path is quite difficult. Most modern circuitry utilizes sophisticated clocking schemes, and careful architecture design, to minimize the longest path. Simple methods such as determination of the longest path through a circuit may identify false paths (resulting in optimization for an unrealistic objective). If the functionality of individual nets is not considered, optimization may focus on large nets (such as reset nets, or scan chains), missing the nets that are of true consequence.

3.2.2 Routability

Half perimeter wire length is only an estimate of the routing resources required to complete a design. If the routing is disproportionately horizontal or vertical, or is unevenly distributed, it may be impossible to completely route the circuit. Without successful routing, a placement is of little use[3].

Many modern fabrication processes use *fixed dies*, in which both circuit area, and the spacing between rows, cannot be changed. Most placement research considers only wire length minimization, but it is possible that a solution with low wire length might not “fit” into the space available, or routing of the placement might fail.

3.3 Additional Considerations

Beyond simply placement quality, we have a number of other issues which can influence the evaluation of results.

3.3.1 Run Times and Random Starts

Run times are frequently reported for each benchmark, but comparisons may be difficult or misleading. First, comparisons of processing power on different computing platforms is non-trivial. Second, implementation details such as language choice may impact run times, but reveal little about the underlying algorithmic complexity.

To complicate the issue further, many placement tools can take advantage of increased run times to obtain improved results. The impact of multiple random starts on partitioning results is well known, so placement methods based on partitioning can simply utilize more starts, increasing run time while generally improving solutions. With annealing based methods, additional run time can be utilized for additional moves at a temperature step, or for an elongated cooling schedule.

Ultimately, there can be tradeoff between run time and solution quality, with a circuit designer perhaps being best suited to determine the right mix for a specific design. In published work, authors usually attempt to provide “reasonable” run time results; but what could be considered “reasonable” changes with the advance of computing platforms.

As many placement algorithms utilize randomization, the results of a single run may vary. To obtain a better measure of average case behavior, multiple runs may be required. The number of runs used when reporting the best observed performance, however, varies; when results are compared, differing numbers of runs, or total run times, may cause the comparison to be unfair.

3.3.2 Tuning

Many placement tools have parameters which can be “tuned.” In bisection based methods, for example, different cut sequences can lead to substantially different results. With annealing based methods, the type of moves considered may also have considerable impact.

In general, it is difficult to determine optimum parameter values for an individual benchmark, and each benchmark might require distinctly different parameter settings. When considering new benchmarks, or configurations of row numbers or row spacings in old benchmarks, it is possible that the default configuration of a placement tool will not produce the best possible result. In practice, it is reasonable to assume that a design team would become familiar with a tool, and be able to make adjustments to match design constraints; for reporting of results, either adjustment or use of default parameters might again be considered unfair.

3.3.3 Tool Versions

In many published works, results from “TimberWolf” are re-

| Benchmark | Rows | Cells | Nets | TW Result |
|-----------|------|--------|--------|-----------|
| fract | 6 | 149 | 163 | 0.032 |
| struct | 21 | 1952 | 1920 | 0.383 |
| primary1 | 17 | 833 | 904 | 0.974 |
| primary2 | 22 | 3014 | 3029 | 3.594 |
| biomed | 44 | 6514 | 7052 | 1.690 |
| industry1 | 24 | 3085 | 2594 | 1.540 |
| industry2 | 69 | 12637 | 13419 | 14.064 |
| industry3 | 52 | 15433 | 21967 | 42.264 |
| avqsmall | 79 | 21918 | 30038 | 6.152 |
| avqlarge | 83 | 25178 | 33298 | 6.580 |
| golem3 | 117 | 100312 | 217362 | 25.113 |

Table 1: MCNC placement benchmarks. The number of rows suggested was determined by the TimberWolf placement and routing tools, using row spacing equal to cell height.

ported. We make special note of this, as there have been many versions of this placement tool, and the version numbering sequence causes some confusion. TimberWolf transitioned from an academic tool, with the highest version being “7.0,” to a commercial tool, with a beginning “1.0” version number. Thus, reported results using “TimberWolf 1.x” are likely references to a fairly recent commercial tool, and not to a very early academic effort. An academic version of TimberWolf is frequently integrated with the Berkeley LAGER package, but this is not the best performing version of the tool, with a default configuration that prefers speed to quality of result.

4. REPORTED RESULTS

We first summarize the MCNC benchmarks (and golem3) in Table 1, including half-perimeter wire length results (after routing and insertion of feedthroughs) from TimberWolf 1.2.6, a well known and well respected commercial placement and routing tool. Row spacings are equal to cell height, and pads are placed outside the core region, on the sides indicated by the original MCNC designs. The number of rows indicated allows for a roughly square core area. The TimberWolf results could be considered a “good reference point” for other results.

We now present Table 2, which summarizes wire lengths reported for the MCNC benchmark circuits. Each column corresponds to a tool which has been presented in a competitive conference or journal publication, with the exception of the last column, which was obtained from InternetCAD, a commercial tool vendor. Obviously, this is only a subset of the work done on standard cell placement; we select this set as they provide a cross-section of results, and are all reasonably well known works.

We have contacted many of the authors cited in this table; uniformly, they have been extremely helpful in clarifying their results. Due to time constraints, we have not been able to contact all authors, and this illustrates a central concern of this paper. Ideally, we would hope that the published record would allow a clear understanding of the results of research; in practice, this is seldom the case.

We report the results *verbatim* from the cited works, and in roughly chronological order. The “benchmark units” used has varied, and we include this detail in the second to last row of the table. Units range from meters to microns, resulting in a difference in where a “decimal point” should be placed. The final row of the table includes an indication of row spacing.

Given the length of time that the MCNC benchmarks have been considered, and the importance of the problem, one might expect

| | [13] Gordian | [9] PRC | [22] HALO | [14] POPINS | [19] TW7.0 | [10] QUAD | [18] NRG |
|-----------|-----------------|------------|----------------|----------------|-----------------|-------------------|----------------|
| Fract | | | | 45225 | | 337 | 25602 |
| Struct | 558+362 | | | | | 3780 | 287631 |
| Primary1 | 841+553 | 999.5 | 1.439 | 1128245 | 0.83 | 8972 | 894545 |
| Primary2 | 4761+3153 | 3665.6 | 6.73 | 4128324 | 3.53 | 36824 | 3412195 |
| Biomed | 5232+3123 | | | 4128324 | 3.22 | 23765 | |
| Industry1 | | | | | | | |
| Industry2 | | | | | 13.30 | 332318 | |
| Industry3 | | | | | 41.53 | 938682 | |
| Avqsmall | | | | 23848188 | 5.08 | 62890 | |
| Avqlarge | | | | 28323022 | 5.65 | 65906 | |
| Golem3 | | | | | 88.98 | | |
| Units | X1000 | X1000 | Meter | Micron | Meter | X100 | Micron |
| Spacing | Routed | | | | Routed | None | |
| | [5] FD98 | [7] ARP | [21] Dragon | [24] SPADE | [11] Mongrel | [23] Feng Shui | [12] iTools |
| Fract | 0.034 | | | 0.024 | | 0.032 | |
| Struct | 0.338 | 0.34 | | 0.291 | 0.266 | 0.380 | 0.272 |
| Primary1 | 0.87 | 0.79 | | 0.74 | 0.83 | 1.018 | 0.799 |
| Primary2 | 3.72 | 3.61 | | 3.13 | 2.94 | 3.684 | 3.37 |
| Biomed | 1.78 | 1.83 | | 1.43 | | 1.689 | 2.90 |
| Industry1 | | 1.50 | | | | 1.606 | |
| Industry2 | 14.6 | | 12.88 | 11.90 | 11.89 | 15.408 | 11.4 |
| Industry3 | 45.1 | 48.12 | 42.33 | 35.37 | 34.53 | 44.729 | 39.6 |
| Avqsmall | 4.91 | 6.06 | 5.17 | 5.08 | 4.4 | 5.960 | 4.48 |
| Avqlarge | 5.38 | 6.54 | 5.25 | 6.16 | 4.87 | 6.301 | 4.78 |
| Golem3 | | | 77.56 | 19.84 | | 21.882 | 79.9 |
| Units | Meter | Meter | Meter | Meter | Meter | Meter | Meter |
| Spacing | Row | | Row | None | None | Row | Routed |

Table 2: Reported wire length results. A significant difference in interpretation is in *row spacing*, which varies from area required by an actual routing, to an estimate of channel height equal to row height, to a complete elimination of spacing.

some sort of convergence of the reported results. Clearly, this is not the case; even for the smallest benchmarks, results within the last few years differ substantially.

5. SUMMARY AND CONCLUSION

Circuit placement is a central problem in VLSI design automation. While fabrication technology has advanced at a breakneck pace, design tools have had a difficult time keeping up.

In this paper, we have shown that there is wide variation in the interpretation of placement benchmarks. This has resulted in vast differences in reported results, with many instances of “apples” being inadvertently compared to “oranges.” The placement problem is extremely complex, and as we have shown, many research groups have made assumptions that, while reasonable, differ substantially from the assumptions of others.

Improvement to design automation research is of profound importance; it is difficult to imagine how this will occur if we cannot accurately evaluate or compare approaches. One of the purposes of publishing research is to enable others to understand the merit of an approach, and to make subsequent comparisons. Without a common interpretation of the available benchmarks, the entire research process is undermined.

While we have focused primarily on wire length evaluation here, we note that there are far greater difficulties with performance driven design. Delay models, device parameters, determination of the longest path, and interpretations of clocking schemes, can all make significant impact on results. For example, the delay reported in [16] is a factor of 10 lower than the result of [20]. Other areas of VLSI CAD research are likely to have similar problems: [2] focused on hypergraph partitioning, and makes a number of sugges-

tions which could very easily apply here. As fabrication technology advances, and the size of the “design gap” increases, improvements to the reporting and evaluation of results becomes increasingly important. Our main suggestions are the following.

5.1 Distribution of Placement Results

We would encourage research groups working on placement to make their placement output available (perhaps through the web). This is common practice in many other academic areas, and we expect that this would reduce ambiguity in reported results substantially. Distribution of results requires very little effort.

As part of the GSRC Bookshelf[8] effort, placement results from a number of tools have been made available. In addition, both source code and executable versions of some placement tools are also available, allowing independent verification of results.

We have refrained from presentation of “definitive” placement results in this paper, preferring instead to direct researchers towards a Web-based table of current results. Placement results are large, preventing unambiguous description in the limited space available for a technical paper. Additionally, placement tools are being continually improved, and results that could be reported here would likely become outdated quickly.

5.2 Standard Measurement

We would also encourage the adoption of a standard for row spacing, number of rows, and method of half-perimeter wire length computation. Many current research groups have eliminated row spacing, and this matches current design practice; thus, we suggest that the standard include zero spacing, and caution that this will make results from current placement tools incomparable to some previously published work. The GSRC Bookshelf standard

cell placement slot includes a tool to determine half perimeter wire length, given a properly formatted placement result.

It should be stressed again that half-perimeter wire length should not be considered as the only measure of placement quality. Issues such as signal delay, power consumption, crosstalk, and area are perhaps more important, but are also quite difficult to measure accurately. It is our hope that if some convergence can be obtained on half-perimeter wire length measures, we can extend this success to other metrics as well.

5.3 New Benchmarks

Industry groups have been hesitant to release current designs, citing competitive advantage issues. We would suggest that while some competitive advantage might be lost, guidance and benchmarks from industry groups could improve the state of computer aided design as a whole.

Recently, a number of industrial partitioning benchmarks[1] have been adapted into standard cell placement benchmarks[21]. While these new benchmarks lack information such as signal directions, cell functionality, etc., they are relatively large, and are also free of wide ranging interpretation. We encourage research groups to report results on these benchmarks, using the row spacings, pad placements, and distance metrics provided. These benchmarks are available through the GSRC Bookshelf standard cell placement slot.

Acknowledgements: We would like to thank the anonymous reviewers, and a number of other people, including Prof. Nael Abu-Ghazaleh, Patrika Agarwal, Dr. Charles Alpert, Andrew E. Caldwell, Prof. Jason Cong, Prof. Shantanu Dutt, Dr. Hans Eisenmann, Bill Halpin, Dr. Dennis J.-H. Huang, Prof. Andrew B. Kahng, Prof. George Karypis, Faris Khundakjie, Prof. Cheng-Kok Koh, Prof. John Lillis, Igor L. Markov, Antonis Papadimitriou, Prof. Massoud Pedram, Dr. Bernhard Reiss, Prof. Majid Sarrafzadeh, Prof. Carl Sechen, Ryon Smey, Dr. Bill Swartz, Arvind Vidyarthi, Dr. Dongmin Xu, Xiaojian Yang, and Mehmet Can Yildiz, for their assistance and candor.

6. REFERENCES

- [1] C. J. Alpert. The ispd98 circuit benchmark suite. In *Proc. Int. Symp. on Physical Design*, pages 80–85, 1998.
- [2] Andrew E. Caldwell, Andrew B. Kahng, Andrew A. Kennings, and Igor L. Markov. Hypergraph partitioning for VLSI CAD: Methodology for heuristic development, experimentation and reporting. In *Proc. Design Automation Conf*, pages 349–354, 1999.
- [3] Andrew E. Caldwell, Andrew B. Kahng, and Igor L. Markov. Can recursive bisection alone produce routable placements? In *Proc. Design Automation Conf*, pages 477–482, 2000.
- [4] Jason Cong, Lei He, Cheng-Kok Koh, and Patrick H. Madden. Performance optimization of VLSI interconnect layout. *Integration, the VLSI Journal*, 21:1–94, 1996.
- [5] H. Eisenmann and F. M. Johannes. Generic global placement and floorplanning. In *Proc. Design Automation Conf*, pages 269–274, 1998.
- [6] W. C. Elmore. The transient response of damped linear networks with particular. *Journal of Applied Physics*, 19(1):55–63, January 1948.
- [7] Hussein Etail, Shawki Areibi, and Anthony Vannelli. Attractor-repeller approach for global placement. In *Proc. Int. Conf. on Computer Aided Design*, pages 20–24, 1999.
- [8] GSRC. Bookshelf slot. <http://www.gigascale.org/bookshelf>.
- [9] Takeo Hamada, Chung-Kuan Cheng, and Paul M. Chau. An efficient multilevel placement technique using hierarchical partitioning. *IEEE Trans. on Circuits and Systems*, 39(6):432–439, June 1992.
- [10] D. J.-H. Huang and A. B. Kahng. Partitioning based standard cell global placement with an exact objective. In *Proc. Int. Symp. on Physical Design*, pages 18–25, 1997.
- [11] Sung-Woo Hur and John Lillis. Mongrel: Hybrid techniques for standard cell placement. In *Proc. Int. Conf. on Computer Aided Design*, pages 165–170, 2000.
- [12] InternetCAD Inc. Benchmark results at <http://www.internetcad.com>. Results compare with TimberWolf 7.0.
- [13] J. Kleinhans, G. Sigl, F. Johannes, and K. Antreich. Gordian: Vlsi placement by quadratic programming and slicing optimization. *IEEE Trans. on Computer-Aided Design of Integrated Circuits andSystems*, 10(3):356–365, 1991.
- [14] Tetshushi Koide, Misuhiro Ono, Shin’ichi Wakabayashi, Yutaka Nisimaru, and Noriyoshi Yoshida. A new performance driven placement method with the elmore delay model for row based vlsis. In *Proc. Asia South Pacific Design Automation Conf.*, pages 405–412, 1995.
- [15] K. Koźmiński. Benchmarks for layout synthesis – evolution and current status. In *Proc. Design Automation Conf*, pages 265–270, 1991.
- [16] S.-L. Ou and M. Pedram. Timing-driven placement based on partitioning with dynamic cut-net control. In *Proc. Design Automation Conf*, pages 472–476, 2000.
- [17] L. T. Pillage and R. A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Trans. on Computer-Aided Design of Integrated Circuits andSystems*, 9(4):352–366, April 1990.
- [18] Majid Sarrafzadeh and Maogang Wang. NRG: Global and detailed placement. In *Proc. Int. Conf. on Computer Aided Design*, pages 532–537, 1997.
- [19] W.-J. Sun and C. Sechen. Efficient and effective placement for very large circuits. *IEEE Trans. on Computer-Aided Design of Integrated Circuits andSystems*, 14(3):349–359, 1995.
- [20] W. Swartz and C. Sechen. Timing driven placement for large standard cell circuits. In *Proc. Design Automation Conf*, pages 211–215, 1995.
- [21] Maogang Wang, Xiaojian Yang, and Majid Sarrafzadeh. Dragon2000: Standard-cell placement tool for large industry circuits. In *Proc. Int. Conf. on Computer Aided Design*, pages 260–263, 2000.
- [22] Yeong-Yil Yang and Chong-Min Kyung. HALO: An efficient global placement strategy for standard cells. *IEEE Trans. on Computer-Aided Design of Integrated Circuits andSystems*, page 1024, 1992.
- [23] M. C. Yildiz and P. H. Madden. Global objectives for standard cell placement. In *GLSVLSI*, 2001.
- [24] Ke Zhong and Shantanu Dutt. Effective partition-driven placement with simultaneous level processing and global net views. In *Proc. Int. Conf. on Computer Aided Design*, volume 2000, pages 254–259, 2000.