information

*Article*

# Exploiting Net Connectivity in Legalization and Detailed Placement Scenarios

**Antonios Dadaliaris [1,*], George Kranas [1,*], Panagiotis Oikonomou [1], George Floros [2] and Michael Dossis [3]**

[1] Computer Science & Telecommunications Department, University of Thessaly, 35100 Lamia, Greece; paikonom@uth.gr
[2] Electrical and Computer Engineering Department, University of Thessaly, 38334 Volos, Greece; gefloros@e-ce.uth.gr
[3] Computer Science Department, University of Western Macedonia, 52100 Kastoria, Greece; mdossis@uowm.gr
* Correspondence: dadaliaris@uth.gr (A.D.); gekranas@uth.gr (G.K.)

**Abstract:** Standard-cell placement is the fundamental step in a typical VLSI/ASIC design flow. Its result, paired with the outcome of the routing procedure can be the decisive factor in rendering a design manufacturable. Global placement generates an optimized instance of the design targeting a set of metrics, while ignoring rules pertaining its feasibility. Legalization and detailed placement rectify this situation, attempting to attain minimum quality loss by often disregarding the connectivity between cells and making runtime the focal point of these steps. In this article, we present a set of variations on a connectivity-based legalization scheme that can either be applied as a legalizer or a detailed placer. The variations can be applied in the entirety of the chip area or in the confinement of a user-specified bin while they are guided by various optimization goals, e.g., total wire length, displacement and density. We analytically describe our variations and evaluate them through extensive simulations on commonly used benchmarks.

**Keywords:** standard cell placement; legalization; detailed placement; standard cell design



**Citation:** Dadaliaris, A.; Kranas, G.; Oikonomou, P.; Floros, G.; Dossis, M. Exploiting Net Connectivity in Legalization and Detailed Placement Scenarios. *Information* **2022**, *13*, 212. https://doi.org/10.3390/info13050212

Academic Editor: Arkaitz Zubiaga

Received: 14 March 2022
Accepted: 12 April 2022
Published: 20 April 2022



## 1. Introduction

The constant change in chip design requirements, has led to the creation of custom design methodologies based on the available vendor-related tool flows and the expected quality of result. Nevertheless, there is a mature and proven backbone process which can be seen in Figure 1 and includes the following steps:

- **Functional Specification**: Architecture, features and functionalities specification in terms of power consumption, area and delay.
- **HDL Design and Simulation**: Design implementation using a Hardware Description Language (HDL) (primarily VHDL or Verilog) followed by a functional simulation that verifies the logical and/or algorithmic behavior of the design.
- **Synthesis**: Conversion of the HDL design into an optimal technology-dependent gate-level netlist, based on a set of constraints.
- **Floorplanning**: Design partitioning that generates the shape and size of blocks, followed by block, macros and pin placement, and chip area estimation.
- **Placement**: Designation of standard cells' positions in pre-defined rows as to minimize total interconnect wire length, power dissipation and delay. Placement can be further divided into three distinct steps: global placement, legalization and detailed placement. During global placement, the coordinates of each standard cell are computed as the outcome of an overall optimization procedure that focuses on minimizing key metrics in addition to interconnect wire length. The result at hand might, and most certainly will, contain overlapping cells and/or cells that are not properly embedded in the design's predefined rows, a situation which certifies the infeasibility of the manufacturing procedure. The aforementioned issues can be resolved by distancing

and fitting the cells throughout legalization. Moreover, the deterioration caused by the preceding step is leveled by minor and swift modifications that are performed in detailed placement.

- **Routing**: Implementation of the connections between cells, blocks and pins.
- **Verification and Signoff**: Succeeding routing, the design process undergoes three steps of physical verification (commonly referred as signoff):
  - Layout versus schematic (LVS), certifying that the layout matches the schematic.
  - Design rule check (DRC), affirming that the geometry follows the foundry rules.
  - Logical equivalence check(LVC), checking the equivalence between pre and post design layout.

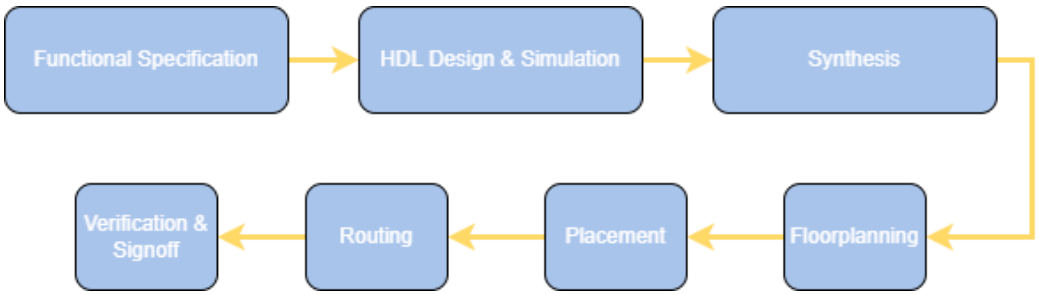



**Figure 1.** Archetypal standard-cell design flow.

The following list reports on the contribution of our work:

- We propose six variations on the legalization scheme described at [1] that can be easily applied on top of any modern standard cell design.
- Each variation can be applied either as a legalizer or as a detailed placer.
- Proposed approaches managed to reduce the total interconnect wire length up to 81% upon the original legalization scheme [1], without significantly affecting the execution time.
- Extensive simulations are conducted to analyze the performance of proposed variations against state-of-the-art algorithms using 22 real-world benchmark circuits

The rest of the article is organized as follows. Section 2 presents related approaches considering legalization and detailed placement, Section 3 introduces a list of variations on a previously published legalization scheme [1], that can be applied either as a legalizer or as a detailed placer, while Section 4 accommodates the experimental results for the selected benchmark suites, followed by a discussion and the conclusions in Section 5.

## 2. Related Work

Tetris [2] and Abacus [3] have laid the foundation for most modern legalizers. A mutual and beneficial feature displayed by both algorithms is their overall execution time, and the ability to minimize total displacement considering the initial locations of the cells. However, both algorithms can be characterised as wire length agnostic, since the minimization of the total interconnect wire length is ignored in placement decisions. Due to their straightforward approach and swift manner, these algorithms provide a good baseline solution, that can be further enhanced with additional optimization approaches and heuristics that mitigate their disadvantages.

In [4], a collection of Tetris-based heuristics, that improve upon major placement-related metrics is presented. Tempering with the directionality regarding each cell's movement and the construction of the chip area grid, restricts abrupt cell movements, thus avoiding a sharp increase in interconnection wire length. A two-step overlap elimination procedure is presented in [5]. Initially, the cells are redeployed vertically, until the row capacity constraints are satisfied, while at later stage, the overlaps within rows are eliminated by modelling the procedure as a topological shortest path calculation problem.

OAL [6] and Domocus [7] can be considered as Abacus-adjacent. OAL presumes upon a linear wire length model and can be applied successfully in designs containing obstacles, while Domocus utilises parallelisation techniques in order to mitigate the high execution time and memory consumption of Abacus. To do that, authors proposed the partition of the entire circuit area into equally-sized vertical zones. Then, the Abacus algorithm is applied in parallel to each zone over the cells contained in the zone. In this way, the need for extra synchronization overhead is avoided (e.g., semaphores) as each zone is processed independently. In addition, in [8] a history-based legalizer is proposed. During each iteration of the algorithm, feasible solutions to the legalization problem at hand are automatically applied, while non-feasible solutions are registered in an appropriate record, forming a sequence of possible moves that should be discouraged in future iterations.

Additionally, in [9], a legalization algorithm targeting mixed-height cell designs that takes into consideration custom region constraints is presented in [10], a swift optimization method for quadratic cell movement that can manage designs containing cells with height equal to that of two standard rows is deployed to tackle the legalization problem, while in [11] the legalization scheme qualifies the minimization of displacement as its target metric.

Finally, in [1], an approach that incorporates the connectivity between cells throughout each iteration of a typical legalization algorithm is presented. Cells are dealt with in an ascending order, according to their lower left corner x-coordinate. For each cell the insertion cost in each row is calculated based on its displacement, and the most cost-effective row is selected to accommodate the cell in question. Upon the completion of the aforementioned move, all cells belonging to the same nets as the preceding cell, are moved by the exactly same amount in the same direction. The only cells excluded from the previous move are those that have been relocated in previous iterations or those that, based on the calculated displacement, are going to be placed outside the boundaries of the core area. The process concludes when all cells have once taken the role of the leader.

Furthermore, considering detailed placement, various approaches have been proposed in academia. We should bear in mind, that these approaches should ideally have minor execution time compared to global placement and yield moderate gains in terms of total half perimeter wire length (HPWL), thus they fall under at least one of the following categories:

- Single Row Optimization: [12–16].
- Cell-to-Slot Matching: [14,17–22].
- Cell Swapping: [14–16,20,23–37].

Single row optimization approaches commonly assume a fixed cell ordering, generated by a previous legalization step and capitalize on the principle of dynamic programming. Unfolding from a pre-computed solution combined with a minimum cost constraint assigned to each cell and related to their current position, an iterative process commences that leads to the computation of the final coordinates.

As for cell-to-slot matching, a subgroup of exchangeable cells is initially identified in a pre-defined bin. A matching problem is subsequently formulated, where cells are to be matched with unoccupied slots inside the aforementioned bin. The legality of the solution might be taken into account (selecting cells with widths less or equal to the slot's at hand) or otherwise a supplementary legalization step is required.

Lastly, cell swapping although the most "simplistic" approach, encompasses a vast amount of possible variations considering the selection of inter-exchangeable cells, the most straightforward approach being taking into account only same size cells therefore retaining the legality of the design. Moreover, the search space can either be expanded by including different size cells or consider empty spaces as void cells, or be restricted in a specific window/bin/row.

## 3. Approaches

There is a plethora of algorithms and heuristics that can be applied throughout physical design [38–41] that achieve feasible solutions upon each of the distinct steps

of the placement procedure. Since standard cell placement can be perceived as an NP problem [42,43], constant research and experimentation is required in order to achieve near optimal solutions without sacrificing execution time.

The most common performance metrics for algorithms belonging in this area are total HPWL and total displacement, both measured in distance units. HPWL is directly related with the routability and the timing behavior of the final outcome, while displacement can be interpreted as a metric showcasing how much we have diverged from an optimal, although illegal, initial solution. A good approximation considering the interconnection wire length of a net, can be calculated based on the half perimeter of the minimum bounding box enclosing its components. Thus, total half perimeter wire length is the summation of each net's wire length. Displacement is calculated as the Manhattan distance between a cell's initial position (after the global placement phase) and the one obtained after any of the subsequent phases (legalization and/or detailed placement). It should be noted that a chip's row density is calculated as the the total area of the cells embedded in the row divided by the row's area.

The work in [1], is used as the yardstick in the proposed methods and as such its functionality requires further analysis. The aforementioned algorithm can be applied either as a legalizer or as a detailed placer in a typical placement flow.

It receives a globally placed design as its input, and proceeds in eliminating overlap and overflow phenomena, meaning overlapping core elements or cells that that extend further than the die area. In the beginning, all core elements are sorted in an ascending order based on their leftmost-x coordinate. Following that, each leading cell is iteratively placed in the die area at the leftmost available position, at a minimum distance from their original position. In an intermediate step, every cell that is directly connected to the leading cell and is a part of the same net, duplicate the movement of its leader, aiming at the deceleration of the deterioration of the total half perimeter wire length.

While this approach succeeds in eradicating manufacturability issues and produces a legal design, it exhibits some drawbacks. Firstly, by stacking all cells to the leftmost available position, congested (highly dense areas) are formed in the left side of the chip and secondly, only half of the available options are considered while checking for the optimal leftmost available positions. Amending these handicaps is the focal point of the variations presented in this work.

The first set of variations focuses on the legalization capabilities of the algorithm:

- **unbounded_bidirectional (ub)**: The main difference from its original counterpart is the utilization of both sides of the chip while considering the position with minimum displacement where the leading cell will be placed. The interconnected cells are relocated in the exact same manner. By considering additional placement slots, cells that are going to be placed in future iterations, will have increased chances of being placed in an optimal position. Moreover, the left-right arrangement reduces the overall density of the chip. Figure 2 depicts an execution example of *ub*, followed by Algorithm 1 describing its functionality.
- **bounded_bidirectional_dens_limit (bd)**: This variation follows the previous bidirectional arrangement but applies a density threshold, in addition, for each row. Upon reaching this threshold, the row at hand is viewed as a macro that cannot be tampered with. The goal is to decongest the globally placed design while simultaneously correcting any illegalities, functioning as a legalizer and a detailed placer at the same time.
- **unbounded_bidirectional_div (ubd)**: Practically, a modified version of *ub* that differentiates in the intermediate step of moving interconnected cells, by recalculating their displacement following the formula:

$$total\_disp = \sum_{0}^{n-1} \frac{orig\_disp}{2^n}, \quad (1)$$

where $n$ is the number of common nets and $orig\_disp$ is the displacement of the leading cell. The goal is to minimize total displacement by limiting the number of moves a cell has to make.

- **bounded_bidirectional_div_dens_limit (bdd)**: An amalgamation of the two previous methods, applying a density threshold on every row of the design while applying the same formula for relocating the interconnected cells.

**Algorithm 1** Unbounded Bidirectional

```
Sort cells in ascending order based on their x-coordinate
for cell = cells[0, 1, ..., N − 1] do
    Find minimum displacement cell position searching in both directions
    Place cell in new position AND render this cell immovable
    for net = cell.nets[0, 1, ..., N − 1] do
        for cell_inter = net.cells[0, 1, ..., N − 1] do
            Move cell_inter the same way as cell
        end for
    end for
end for
```

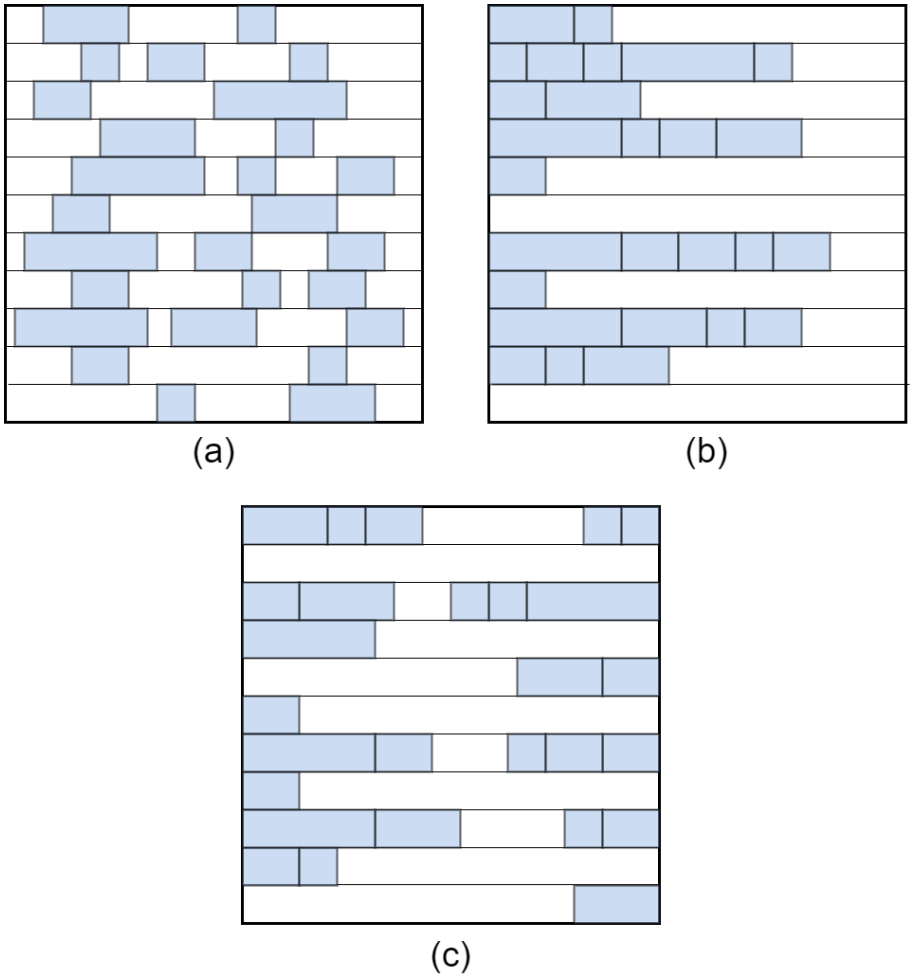



**Figure 2.** Unbounded bidirectional legalization approach, (**a**) initial placement (**b**) connectivity-based legalization (**c**) unbounded bidirectional legalization.

Additional adaptations were explored utilizing the option of dividing the overall chip area in an even number of bins, thus creating a grid, where the aforementioned variations could be applied separately. The number of bins is initially defined based on the user's

input leading to the grid formation (i.e., if the given size is equal to 4, the grid will contain 16 bins). This grid can be considered static, since it remains stable throughout the execution of each approach. The minimum and maximum value of the grid size cannot be strictly defined as it depends upon the design's overall area and core elements.

- **unbounded_bidirectional_grid (ubg)**: The core functionality is implemented, unmodified, and applied upon each bin of the design. Every cell within a bin is placed into an optimal positions following the bidirectional search pattern, generating a layout similar to the one presented in Figure 3. Further displacement reduction is attained, That way we can achieve further reduction upon displacement of every core element and also achieve decongestion of more areas within the design.
- **unbounded_bidirectional_div_grid (ubdg)**: Similar to the corresponding legalization scheme where the minimization of interconnected cells' displacement is taken into account. The generated grid restricts considerably the available search areas.

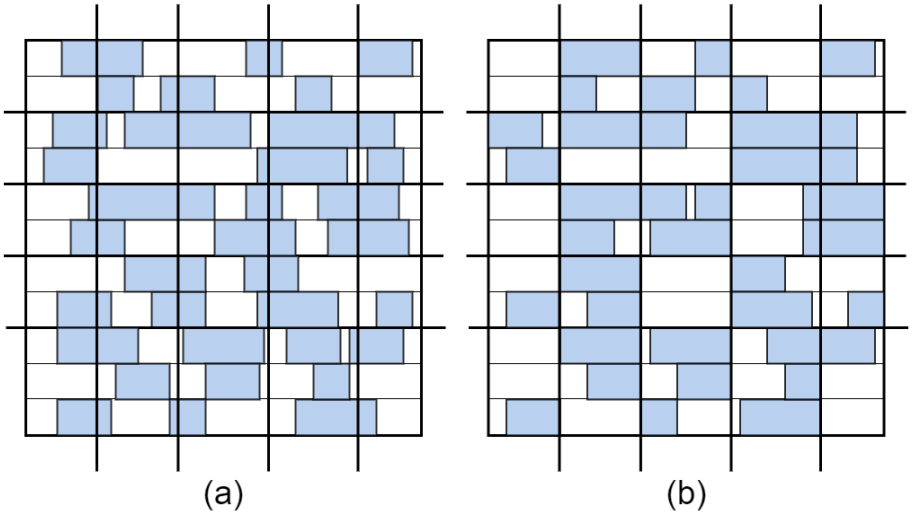



**Figure 3.** Grid layout, (**a**) initial placement, (**b**) final placement.

It should be noted that grid-based versions of the density-driven approaches were also implemented, but as the results demonstrate, there may exist certain cases where grid-based approaches exhibit performance degradation. Finally, a subset of the variations was applied as a detailed placement scheme, in randomly placed designs containing macros to test their effectiveness in a different context.

The proposed variations have an $\mathcal{O}(C(R + C_N C_X))$ time complexity for $C$ cells, $R$ rows. $C_N$ and $C_X$ denote the number of nets where cell $C$ is a part of and the number of cells connected via the net with $C$, respectively.

## 4. Experimental Results

### *4.1. Simulation Setup*

***Standard Cell Designs***

We report on the experimental evaluation of the proposed variation relying on eighteen (18) standard cell designs (benchmarks) as depicted by [44] and on four (4) benchmarks from [45]. These benchmarks are derived from industrial ASIC designs. The number of cells is ranging from 12,506 to 210,341 and from 211,447 to 496,045, respectively. The total number of nets varies from 14,111 to 201,920 and from 221,142 to 515,951, accordingly. The detailed characteristics of ISPD'98 and ISPD'05 benchmarks are presented in Tables 1 and 2, respectively. Regarding [45], in order to produce a feasible all non-movable objects must to placed inside the placement region. The designs were globally placed using mPL6 [46] and ePlace [47,48], and were subsequently legalized by applying the baseline legalizer and the variations described in the previous section.

**Table 1.** ISPD'98 benchmark characteristics.

| Design | #Cells | #I/O Pads | #Nets | #Pins | #Rows |
|---|---|---|---|---|---|
| ibm01 | 12,506 | 246 | 14,111 | 50,566 | 96 |
| ibm02 | 19,342 | 259 | 19,584 | 81,199 | 109 |
| ibm03 | 22,853 | 283 | 27,401 | 93,573 | 121 |
| ibm04 | 27,220 | 287 | 31,970 | 105,859 | 136 |
| ibm05 | 28,146 | 1201 | 28,446 | 126,308 | 139 |
| ibm06 | 32,332 | 166 | 34,826 | 128,182 | 126 |
| ibm07 | 45,639 | 287 | 48,117 | 175,639 | 166 |
| ibm08 | 51,023 | 286 | 50,513 | 204,890 | 170 |
| ibm09 | 53,110 | 285 | 60,902 | 222,088 | 183 |
| ibm10 | 68,685 | 744 | 75,196 | 297,567 | 234 |
| ibm11 | 70,152 | 406 | 81,454 | 280,786 | 208 |
| ibm12 | 70,439 | 637 | 77,240 | 317,760 | 242 |
| ibm13 | 83,709 | 490 | 99,666 | 357,075 | 224 |
| ibm14 | 147,088 | 517 | 152,772 | 546,816 | 305 |
| ibm15 | 161,187 | 383 | 186,608 | 715,823 | 303 |
| ibm16 | 182,980 | 504 | 190,048 | 778,823 | 347 |
| ibm17 | 184,752 | 743 | 189,581 | 860,036 | 379 |
| ibm18 | 210,341 | 272 | 201,920 | 819,697 | 361 |

**Table 2.** ISPD'05 benchmark characteristics.

| Design | #Objects | #Movable Objects | #Fixed Objects | #Nets | #Pins | #Rows |
|---|---|---|---|---|---|---|
| adaptec1 | 211,447 | 210,904 | 543 | 221,142 | 944,053 | 890 |
| adaptec2 | 255,023 | 254,457 | 566 | 266,009 | 1,069,482 | 1170 |
| adaptec3 | 451,650 | 450,927 | 723 | 466,758 | 1,875,039 | 1944 |
| adaptec4 | 496,045 | 494,716 | 1329 | 515,951 | 1,912,420 | 1944 |

***Performance Metrics***

To evaluate the performance of the proposed variations we adopt the following set of metrics: (i) total HPWL, (ii) total displacement and (iii) the overall execution time. For each design, all metrics were measured as the percentage of performance improvement of each approach ($A$) over the connectivity-based legalization scheme ($B$) as follows:

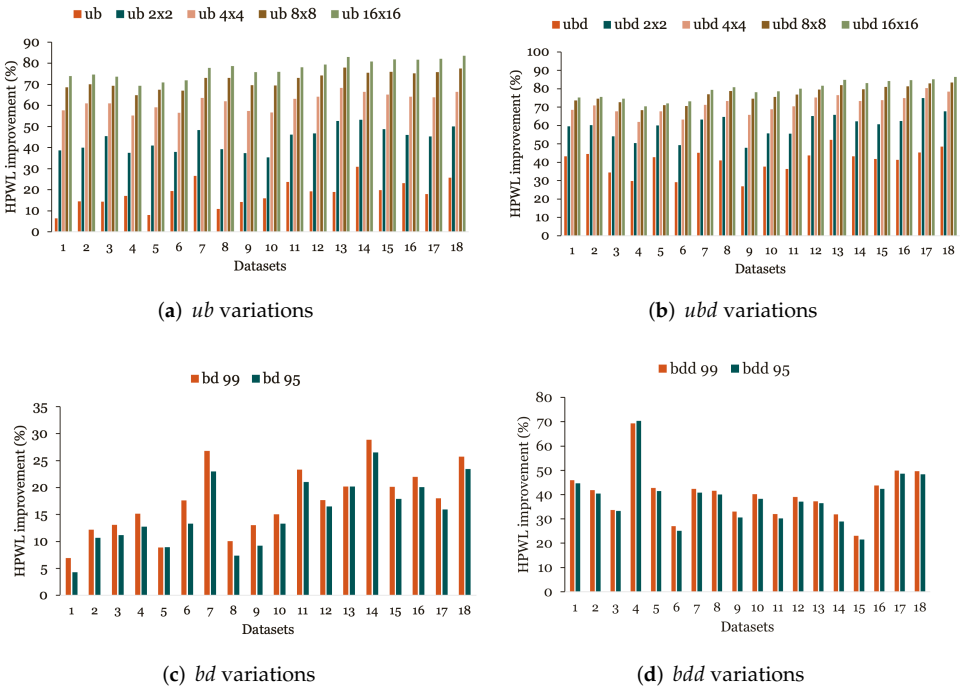
$$improvement = \frac{performance(B) - performance(A)}{performance(B)} \quad (2)$$

All algorithms were implemented in Python, and experiments were performed on a Linux-based server, with two 6-core Intel Xeon E5-2630 CPUs running 2.2 GHz, using the benchmark suit of [44,45].

### *4.2. Performance Assessment*

Firstly, we perform a set of experiments using mPL6 as a global placer over the ISPD'98 benchmarks, whose characteristics are presented in Table 1. Each figure is divided into four sub figures each oh which presents a different proposed variation. In Figure 4, we present the percentage improvement in terms of HPWL. In total all of our variations are performing good in case of HPWL improvement, which is quite encouraging for our variations. In case of $ub$, $bd$, $ubd$ and $bdd$, the improvement is proportional to the size of the design as well as the methodology that each approach is following. This can be justified by the fact that the latter approaches are performing changes to the entire design. To be more precise, $ub$ (Figure 4a), is the first and most similar variation to the original algorithm. Introducing a novel methodology to identify the best position to place a cell clearly achieves significant results in terms of HPWL. Density threshold approaches ($bd$ and $bdd$) are presented in Figure 4c,d. A notable reduction in HWPL (up to 70%) is achieved when we set the threshold down to 95% of the original row density. We have to mention that lower threshold

values will result into non-feasible solutions. This is due to the nature of our benchmarks under testing, as they are smaller in scale and already densely populated. In all the aforementioned cases, we can notice a performance degradation in terms of displacement and execution time. This is expected, as in our approaches, by minimizing HPWL we need to re-place interconnected cells as close as possible in an iterative manner. Regarding the grid based approaches (*ubg* and *ubdg*) our variations demonstrate the best performance. Furthermore, in Figure 6a,b it is shown how the techniques that divide the designs into grid can decrease significantly the execution time. The above performance is reasonable since the best position of each cell can be found by exploring a much smaller-in-scale area compared to the rest of the variations. Additionally, this impacts the HWPL metric as well, because the displacement of each cell from its initial position is far smaller in range (Figure 5a,b).

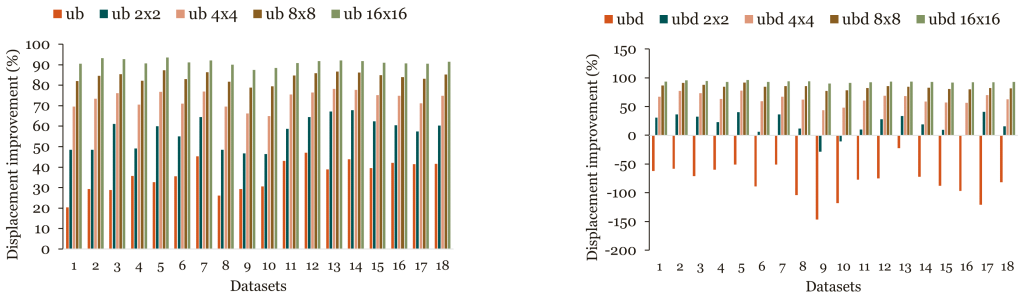



(**a**) *ub* variations (**b**) *ubd* variations

(**c**) *bd* variations (**d**) *bdd* variations

**Figure 4.** HPWL improvement (%) using mPL6 as a global placer.

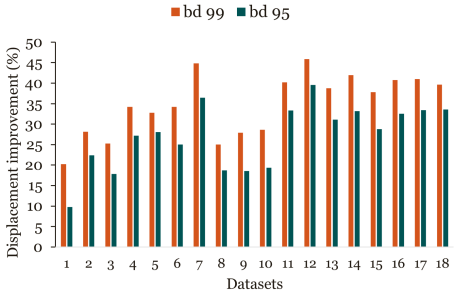
Subsequently, the same set of designs, now globally placed by ePlace, are utilized to enhance our understanding upon the efficiency of the proposed approaches. The results are quite similar and follow the same patterns as those of the first set, thus proving that all approaches are performing better than the original algorithm, as a legalization method, regardless of the global placement algorithms applied in previous steps. The results in their entirety are depicted in Figures 7–9. As can be seen in Table 3, considering the grid-based approaches, whose effectiveness is explained in detail in the previous paragraph, remarkable maximum average improvement can be observed in key metrics (81% for HPWL, 93% for displacement and 99% in execution time) in comparison to the baseline legalizer. Analytic results considering HPWL are depicted in Table 4.



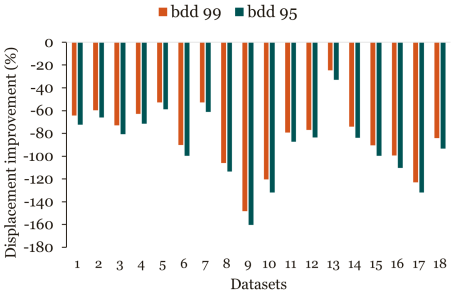

(**a**) *ub* variations




(**b**) *ubd* variations

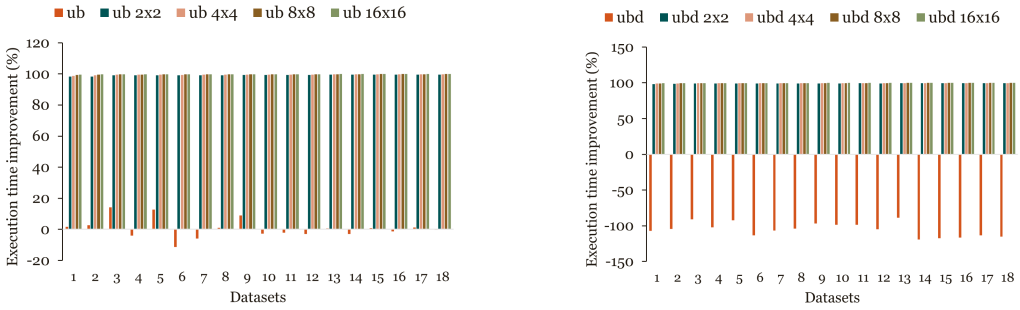



(**c**) *bd* variations




(**d**) *bdd* variations

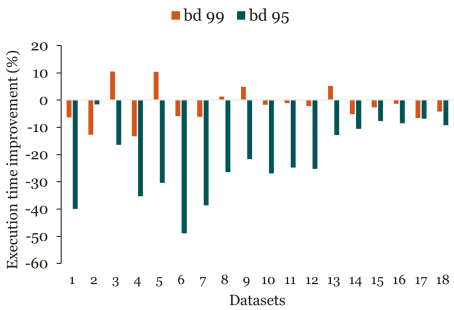
**Figure 5.** Displacement improvement (%) using mPL6 as a global-placer.



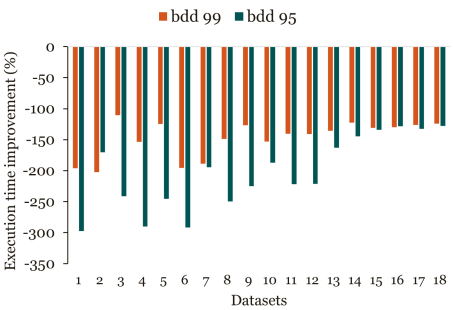

(**a**) *ub* variations




(**b**) *ubd* variations

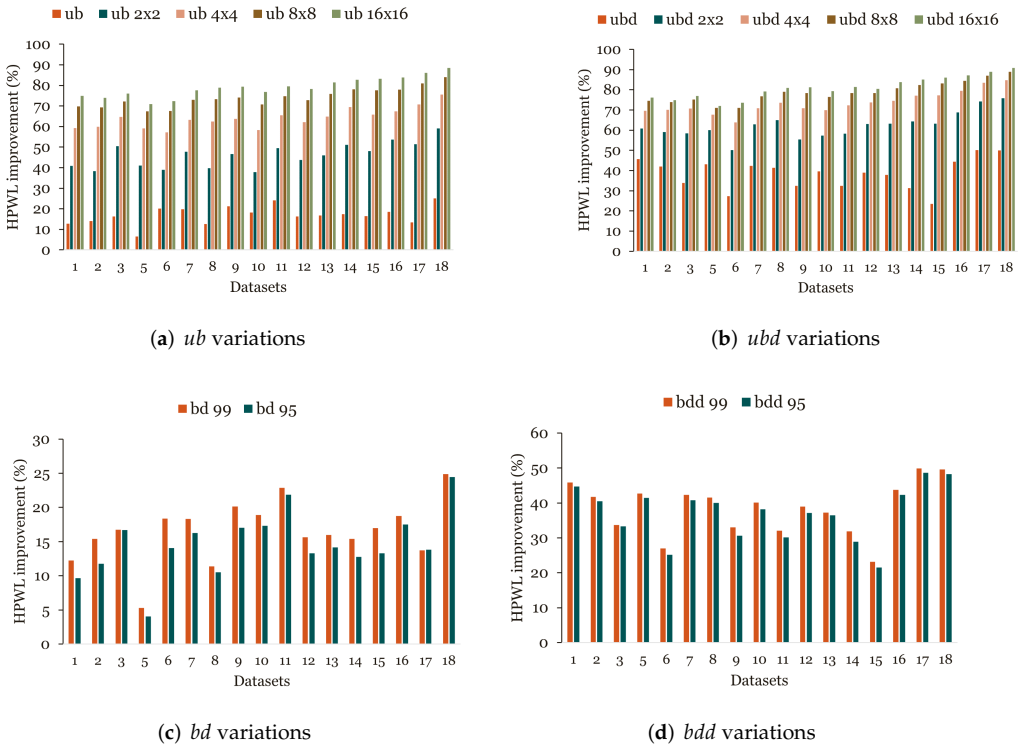



(**c**) *bd* variations




(**d**) *bdd* variations

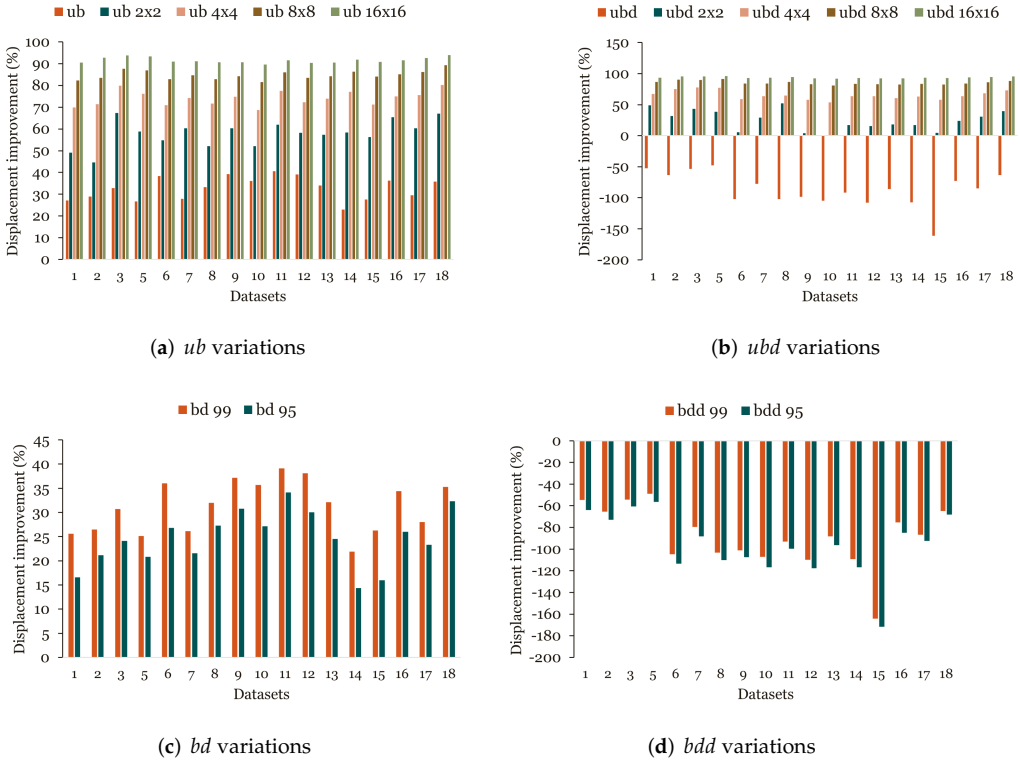
**Figure 6.** Execution time improvement (%) using mPL6 as a global-placer.




(**a**) *ub* variations (**b**) *ubd* variations

(**c**) *bd* variations (**d**) *bdd* variations

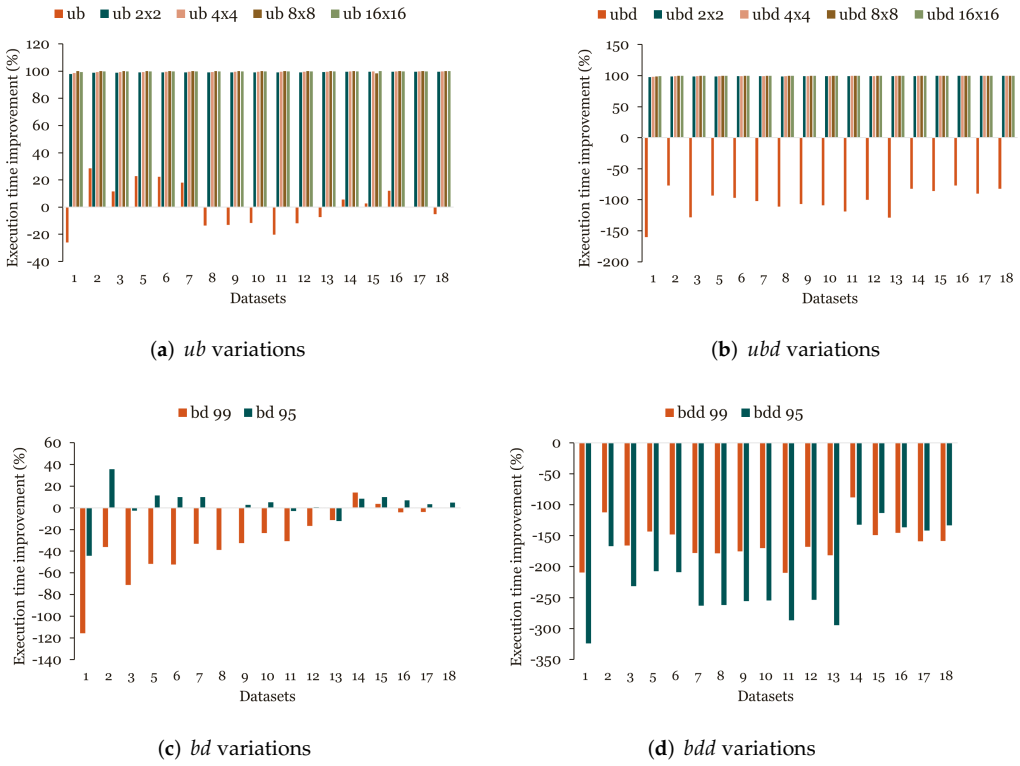
**Figure 7.** HPWL improvement (%) using ePlace as a global placer.




(**a**) *ub* variations (**b**) *ubd* variations

(**c**) *bd* variations (**d**) *bdd* variations

**Figure 8.** Displacement improvement (%) using ePlace as a global-placer.




(**a**) *ub* variations

(**b**) *ubd* variations

(**c**) *bd* variations

(**d**) *bdd* variations

**Figure 9.** Execution time improvement (%) using ePlace as a global placer.

**Table 3.** Average improvement in core placement metrics.

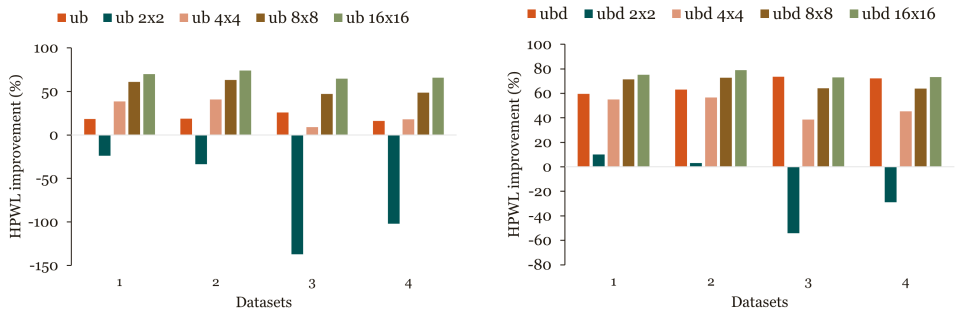
| Approach | Avg. HPWL Improvement (mPL6) | Avg. HPWL Improvement (ePlace) | Avg. Displacement Improvement (mPL6) | Avg. Displacement Improvement (ePlace) | Avg. Runtime Improvement (mPL6) | Avg. Runtime Improvement (ePlace) |
|---|---|---|---|---|---|---|
| ub | 18.13% | 17.01% | 36.18% | 32.72% | 0.53% | 0.86% |
| bd99 | 17.48% | 16.52% | 34.858% | 31.17% | −2.02% | −29.56% |
| bd95 | 15.31% | 14.61% | 27.15% | 24.52% | −21.76% | 2.78% |
| ubd | 40.36% | 38.57% | −80.25% | −86.99% | −105.06% | −102.99% |
| bdd99 | 40.32% | 38.50% | −82.35% | −88.83% | −147.22% | −161.25% |
| bdd95 | 38.74% | 36.95% | −91.09% | −96.33% | −203.66% | −215.67% |
| ub(2 × 2) | 43.87% | 46.13% | 57.05% | 57.88% | 99.22% | 99.12% |
| ubd(2 × 2) | 59.94% | 62.36% | 18.57% | 24.71% | 99.20% | 99.03% |
| ub(4 × 4) | 61.78% | 64.06% | 73.26% | 74.12% | 99.54% | 99.45% |
| ubd(4 × 4) | 71.21% | 73.50% | 63.39% | 65.26% | 99.50% | 99.38% |
| ub(8 × 8) | 72.12% | 74.07% | 83.95% | 84.76% | 99.72% | 99.89% |
| ubd(8 × 8) | 76.84% | 78.78% | 83.97% | 85.28% | 99.67% | 99.60% |
| ub(16 × 16) | 77.41% | 79.09% | 91.08% | 91.56% | 99.81% | 99.78% |
| ubd(16 × 16) | 79.31% | 81.04% | 92.98% | 93.65% | 99.78% | 99.73% |

**Table 4.** Detailed results considering HPWL of the dominant legalization variations.

| Design | GP (mPL6) | Baseline [1] | ub 8 × 8 | ubd 8 × 8 | ub 16 × 16 | ubd 16 × 16 |
|---|---|---|---|---|---|---|
| ibm01 | 2,073,720 | 8,578,749 | 2,687,459 | 2,268,377 | 2,232,947 | 2,127,129 |
| ibm02 | 4,115,535 | 17,088,957 | 5,104,443 | 4,348,663 | 4,328,567 | 4,176,168 |
| ibm03 | 5,618,096 | 22,618,146 | 6,934,248 | 6,196,641 | 5,970,270 | 5,764,097 |
| ibm04 | 6,959,972 | 23,732,424 | 8,328,121 | 7,522,072 | 7,262,231 | 7,030,794 |
| ibm05 | 10,058,599 | 36,390,124 | 11,862,607 | 10,530,664 | 10,582,865 | 10,163,971 |
| ibm06 | 5,735,064 | 22,289,668 | 7,360,543 | 6,551,592 | 6,250,799 | 5,973,455 |
| ibm07 | 9,403,115 | 47,913,647 | 12,863,250 | 11,069,734 | 10,653,858 | 9,894,014 |
| ibm08 | 10,188,507 | 54,872,802 | 14,751,303 | 11,639,903 | 11,703,331 | 1,0572,367 |
| ibm09 | 11,095,230 | 54,440,500 | 16,567,562 | 13,836,911 | 13,147,719 | 11,938,413 |
| ibm10 | 19,841,489 | 98,718,698 | 30,091,176 | 24,160,180 | 23,733,083 | 21,170,214 |
| ibm11 | 16,349,962 | 89,133,528 | 23,991,183 | 20,617,189 | 19,454,785 | 17,756,809 |
| ibm12 | 24,204,836 | 139,103,758 | 35,747,744 | 28,500,840 | 28,719,277 | 25,676,673 |
| ibm13 | 19,099,801 | 141,639,550 | 31,222,644 | 25,607,769 | 24,010,729 | 21,529,955 |
| ibm14 | 34,804,692 | 234,332,753 | 57,311,946 | 47,511,227 | 44,773,614 | 39,711,435 |
| ibm15 | 42,097,139 | 309,725,895 | 74,604,463 | 58,926,602 | 56,302,260 | 48,779,650 |
| ibm16 | 45,342,902 | 353,949,149 | 87,668,119 | 66,058,790 | 64,848,164 | 5,4053,562 |
| ibm17 | 62,242,741 | 483,965,502 | 116,968,604 | 82,558,553 | 86,069,785 | 71,817,907 |
| ibm18 | 44,084,573 | 386,787,994 | 86,903,950 | 64,359,380 | 63,517,563 | 5,2543,606 |
| adaptec1 | 103,334,844 | 491,406,304 | 19,1049,903 | 141,001,589 | 148,124,392 | 121,840,413 |
| adaptec2 | 121,481,139 | 684,333,764 | 251,113,848 | 185,450,686 | 178,073,983 | 143,689,629 |
| adaptec3 | 256,734,372 | 1,314,853,028 | 695,749,638 | 472,768,557 | 466,814,791 | 355,400,251 |
| adaptec4 | 230,334,408 | 126,638,919 | 652,205,290 | 457,445,662 | 434,328,800 | 338,347,283 |

The last set of experiments targeted the application of the aforementioned approaches as detailed placement methods over legalized designs. The benchmarks from [45] were globally placed in a random manner and legalized by the original algorithm. It should be noted that these designs contain immovable core elements that cannot partake in any placement-related procedure and cannot be moved/re-placed in any way. The size of these elements, compared to the available spaces in the die area, render the density-driven approaches presented, impractical, and as a consequence no further experiments were performed.

The non-grid versions of *ub* and *ubd* improve HPWL, as expected. As for their counterpart, although the 2 × 2 grid version seems unable to generate sustainable results (mainly due to the "crude" partitioning of the overall area), the remainder of the approaches outperform this version in an incremental way as the partitioning scale rises to create greater grids, as depicted in Figures 10–12 display the results concerning displacement and execution time, respectively. In both cases, improvements are inversely proportional to the size of the design and the grid constructed.

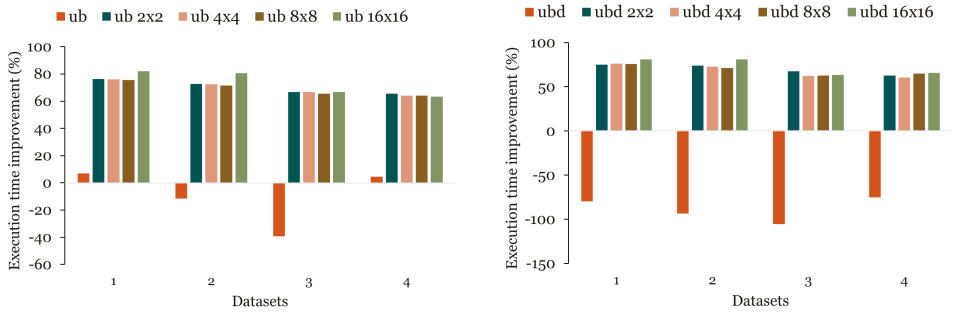



(**a**) *ub* variations (**b**) *ubd* variations

**Figure 10.** HPWL improvement (%) using upon randomly placed design in detailed placement approach.


(**a**) *ub* variations (**b**) *ubd* variations

**Figure 11.** Displacement improvement (%) using upon randomly placed design in detailed placement approach.




(**a**) *ub* variations (**b**) *ubd* variations

**Figure 12.** Execution time improvement (%) using upon randomly placed design in detailed placement approach.

## 5. Discussion and Conclusions

In this article, we introduced new approaches to tackle the VLSI placement problem. We presented legalization and detailed placement variations that consider the entire chip design as well as grid based approaches whereby the entire design area is divided into a variable sized grid. We provide the outcome of an extensive experimentation using 22 real-world benchmark circuits (fixed macro blocks were included) and compared the proposed variations with a sophisticated legalizer/detailed placer and two global placers. Our evaluation exhibits that the proposed variations can outperform the performance of the solutions provided in the respective literature in terms of both HPWL and execution time. Viewing the results in a retrospect we can state that the variations that utilize the grid formation achieve the best performance in all three optimization criteria, i.e., HPWL, displacement and execution time. Generally, in grid variations the total number of cells' moves are fewer compared to the rest approaches as the grid creation generates small areas of unused space, i.e., smaller number of candidate positions to allocate an unplaced cell. The above holds true regarding the type of global placer being used as an initial step. Thus, the proposed variations can be applied after any global placer encountered without jeopardising the anticipated performance.

**Author Contributions:** Conceptualization, A.D., G.K. and P.O.; methodology, A.D. and G.K.; software, G.K.; validation, A.D. and P.O.; investigation, A.D., G.F. and M.D.; resources, A.D. and G.K.; data curation, G.K.; writing—original draft preparation, A.D., G.K. and P.O.; writing—review and editing, G.F. and M.D.; visualization, G.K.; supervision, A.D. and M.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data available on request due to restrictions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References


1. Dadaliaris, A.N.; Oikonomou, P.; Koziri, M.G.; Nerantzaki, E.; Loukopoulos, T.; Stamoulis, G.I. A connectivity-based legalization scheme for standard cell placement. *Circuits Syst.* **2017**, *8*, 191. [CrossRef]
2. Hill, D. Method and System for High Speed Detailed Placement of Cells within an Integrated Circuit Design. U.S. Patent 6,370,673, 9 April 2002.
3. Spindler, P.; Schlichtmann, U.; Johannes, F.M. Abacus: Fast legalization of standard cell circuits with minimal movement. In Proceedings of the 2008 International Symposium on Physical Design, Portland, OR, USA, 13–16 April 2008; pp. 47–53.
4. Dadaliaris, A.N.; Oikonomou, P.; Koziri, M.G.; Nerantzaki, E.; Hatzaras, Y.; Garyfallou, D.; Loukopoulos, T.; Stamoulis, G.I. Heuristics to augment the performance of Tetris legalization: Making a fast but inferior method competitive. *J. Low Power Electron.* **2017**, *13*, 220–230. [CrossRef]
5. He, L.; Kahng, A.B.; Tam, K.H.; Xiong, J. Variability-driven considerations in the design of integrated-circuit global interconnects. In Proceedings of the 21th Intl. VLSI Multilevel Interconnection (VMIC) Conf. Citeseer, Waikoloa, HI, USA, 29 September–2 October 2004; pp. 214–221.
6. Chou, S.; Ho, T.Y. OAL: An obstacle-aware legalization in standard cell placement with displacement minimization. In Proceedings of the IEEE International SOC Conference (SOCC), Belfast, Ireland, 9–11 September 2009; pp. 329–332.
7. Oikonomou, P.; Koziri, M.G.; Dadaliaris, A.N.; Loukopoulos, T.; Stamoulis, G.I. Domocus: Lock free parallel legalization in standard cell placement. In Proceedings of the 6th International Conference on Modern Circuits and Systems Technologies (MOCAST), Thessaloniki, Greece, 4–6 May 2017; pp. 1–4.
8. Cho, M.; Ren, H.; Xiang, H.; Puri, R. History-based VLSI legalization using network flow. In Proceedings of the 47th Design Automation Conference, Anaheim, CA, USA, 13–18 June 2010; pp. 286–291.
9. Zhu, Z.; Chen, J.; Zhu, W.; Chang, Y.W. Mixed-cell-height legalization considering technology and region constraints. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2020**, *39*, 5128–5141. [CrossRef]
10. Hougardy, S.; Neuwohner, M.; Schorr, U. A Fast Optimal Double Row Legalization Algorithm. In Proceedings of the 2021 International Symposium on Physical Design, Online, 22–24 March 2021; pp. 23–30.
11. Ferreira, J.; Butzen, P.F.; Meinhardt, C.; Reis, R.A. FBM: A Simple and Fast Algorithm for Placement Legalization. In Proceedings of the 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Genoa, Italy, 27–29 November 2019; pp. 209–212.
12. Chen, J.; Zhu, W. An analytical placer for VLSI standard cell placement. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2012**, *31*, 1208–1221. [CrossRef]
13. Bredthauer, B.; Olbrich, M.; Barke, E. Stp-a quadratic vlsi placement tool using graphic processing units. In Proceedings of the 17th International Symposium on Parallel and Distributed Computing (ISPDC), Amsterdam, The Netherlands, 3–7 June 2018; pp. 77–84.
14. Lin, Y.; Pan, D.Z.; Ren, H.; Khailany, B. DREAMPlace 2.0: Open-source gpu-accelerated global and detailed placement for large-scale vlsi designs. In Proceedings of the China Semiconductor Technology International Conference (CSTIC), Shanghai, China, 18–19 March 2020; pp. 1–4.
15. Gu, J.; Jiang, Z.; Lin, Y.; Pan, D.Z. DreamPlace 3.0: Multi-electrostatics based robust VLSI placement with region constraints. In Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD), San Diego, CA, USA, 2–5 November 2020; pp. 1–9.
16. Lin, Y.; Li, W.; Gu, J.; Ren, H.; Khailany, B.; Pan, D.Z. ABCDPlace: Accelerated batch-based concurrent detailed placement on multithreaded cpus and GPUs. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2020**, *39*, 5083–5096. [CrossRef]
17. Zhu, W.; Huang, Z.; Chen, J.; Chang, Y.W. Analytical solution of poisson's equation and its application to vlsi global placement. In Proceedings of the International Conference on Computer-Aided Design, San Diego, CA, USA, 5–8 November 2018; pp. 1–8.
18. Lin, T.; Chu, C.; Wu, G. POLAR 3.0: An ultrafast global placement engine. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, USA, 2–6 November 2015; pp. 520–527.
19. Chen, J.; Peng, Z.; Zhu, W. A VLSI global placement solver based on proximal alternating direction method. In Proceedings of the IEEE 11th International Conference on ASIC (ASICON), Chengdu, China, 3–6 November 2015; pp. 1–4.
20. Popovych, S.; Lai, H.H.; Wang, C.M.; Li, Y.L.; Liu, W.H.; Wang, T.C. Density-aware detailed placement with instant legalization. In Proceedings of the 51st ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 1–5 June 2014; pp. 1–6.
21. Cheng, C.K.; Kahng, A.B.; Kang, I.; Wang, L. Replace: Advancing solution quality and routability validation in global placement. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2018**, *38*, 1717–1730. [CrossRef]
22. Chen, T.C.; Jiang, Z.W.; Hsu, T.C.; Chen, H.C.; Chang, Y.W. NTUplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraints. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2008**, *27*, 1228–1240. [CrossRef]

23. Darav, N.K.; Kennings, A.; Westwick, D.; Behjat, L. High performance global placement and legalization accounting for fence regions. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, USA, 2–6 November 2015; pp. 514–519.
24. Hu, B.; Marek-Sadowska, M. FAR: Fixed-points addition & relaxation based placement. In Proceedings of the 2002 International Symposium on Physical Design, Del Mar, CA, USA, 7–10 April 2002; pp. 161–166.
25. Vygen, J. Algorithms for detailed placement of standard cells. In Proceedings of the Conference on Design, Automation and Test in Europe, Paris, France, 23–26 February 1998; pp. 321–324.
26. Chan, T.; Cong, J.; Sze, K. Multilevel generalized force-directed method for circuit placement. In Proceedings of the International Symposium on Physical Design, San Francisco, CA, USA, 3–6 April 2005; pp. 185–192.
27. Yang, X.; Sarrafzadeh, M.; Wang, M. Dragon2000: Standard-cell placement tool for large industry circuits. In Proceedings of the IEEE/ACM International Conference on Computer Aided Design, ICCAD-2000, IEEE/ACM Digest of Technical Papers (Cat. No. 00CH37140), San Jose, CA, USA, 5–9 November 2000; pp. 260–263.
28. Taghavi, T.; Yang, X.; Choi, B.K. Dragon2005: Large-scale mixed-size placement tool. In Proceedings of the 2005 International Symposium on Physical Design, San Francisco, CA, USA, 3–6 April 2005; pp. 245–247.
29. Kim, M.C.; Viswanathan, N.; Alpert, C.J.; Markov, I.L.; Ramji, S. MAPLE: Multilevel adaptive placement for mixed-size designs. In Proceedings of the 2012 ACM International Symposium on International Symposium on Physical Design, Taipei, Taiwan, 29 March–1 April 2012; pp. 193–200.
30. Luo, T.; Pan, D.Z. DPlace2. 0: A stable and efficient analytical placement based on diffusion. In Proceedings of the Asia and South Pacific Design Automation Conference, Seoul, Korea, 21–24 January 2008; pp. 346–351.
31. Yan, J.Z.; Chu, C.; Mak, W.K. Safechoice: A novel approach to hypergraph clustering for wirelength-driven placement. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2011**, *30*, 1020–1033. [CrossRef]
32. He, X.; Huang, T.; Xiao, L.; Tian, H.; Cui, G.; Young, E.F. Ripple: An effective routability-driven placer by iterative cell movement. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Diego, CA, USA, 2–5 November 2011; pp. 74–79.
33. He, X.; Huang, T.; Xiao, L.; Tian, H.; Young, E.F. Ripple: A robust and effective routability-driven placer. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2013**, *32*, 1546–1556. [CrossRef]
34. Kim, M.C.; Lee, D.J.; Markov, I.L. SimPL: An effective placement algorithm. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2011**, *31*, 50–60. [CrossRef]
35. Kim, M.C.; Lee, D.J.; Markov, I.L. Simpl: An algorithm for placing VLSI circuits. *Commun. ACM* **2013**, *56*, 105–113. [CrossRef]
36. Spindler, P.; Schlichtmann, U.; Johannes, F.M. Kraftwerk2—A fast force-directed quadratic placement approach using an accurate net model. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2008**, *27*, 1398–1411. [CrossRef]
37. Chang, C.C.; Cong, J.; Pan, Z.; Yuan, X. Multilevel global placement with congestion control. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2003**, *22*, 395–409. [CrossRef]
38. Sherwani, N.A. *Algorithms for VLSI Physical Design Automation*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
39. Alpert, C.J.; Mehta, D.P.; Sapatnekar, S.S. *Handbook of Algorithms for Physical Design Automation*; CRC Press: Boca Raton, FL, USA, 2008.
40. Sait, S.M.; Youssef, H. *VLSI Physical Design Automation: Theory and Practice*; World Scientific: Singapore, 1999; Volume 6.
41. Kahng, A.B.; Lienig, J.; Markov, I.L.; Hu, J. *VLSI Physical Design: From Graph Partitioning to Timing Closure*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.
42. Garey, M.R.; Johnson, D.S. *Computers and Intractability*; Freeman: San Francisco, CA, USA, 1979; Volume 174.
43. Garey, M.R.; Johnson, D.S.; Stockmeyer, L. Some simplified NP-complete problems. In Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, 30 April–2 May 1974; pp. 47–63.
44. Viswanathan, N.; Chu, C.N. FastPlace: Efficient analytical placement using cell shifting, iterative local refinement, and a hybrid net model. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2005**, *24*, 722–733. [CrossRef]
45. Nam, G.J.; Alpert, C.J.; Villarrubia, P.; Winter, B.; Yildiz, M. The ISPD2005 placement contest and benchmark suite. In Proceedings of the 2005 International Symposium on Physical Design, San Francisco, CA, USA, 3–6 April 2005; pp. 216–220.
46. Chan, T.F.; Cong, J.; Shinnerl, J.R.; Sze, K.; Xie, M. mPL6: Enhanced multilevel mixed-size placement. In Proceedings of the 2006 International Symposium on Physical Design, San Jose, CA, USA, 9–12 April 2006; pp. 212–214.
47. Lu, J.; Chen, P.; Chang, C.C.; Sha, L.; Dennis, J.; Huang, H.; Teng, C.C.; Cheng, C.K. ePlace: Electrostatics based placement using Nesterov's method. In Proceedings of the 51st ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 1–5 June 2014; pp. 1–6.
48. Lu, J.; Chen, P.; Chang, C.C.; Sha, L.; Huang, D.J.H.; Teng, C.C.; Cheng, C.K. ePlace: Electrostatics-based placement using fast fourier transform and Nesterov's method. *ACM Trans. Des. Autom. Electron. Syst. (TODAES)* **2015**, *20*, 1–34. [CrossRef]