

Article

Module Based Floorplanning Methodology to Satisfy Voltage Island and Fixed Outline Constraints

Srinath Balasubramanian ^{1,*}, Arunapriya Panchanathan ^{1,†}, Bharatiraja Chokkalingam ^{2,*}, Sanjeevikumar Padmanaban ^{3,†} and Zbigniew Leonowicz ^{4,†}

¹ Department of Electronics and Communication, SRM Institute of Science and technology, Chennai 603 203, India; arunapriya.p@ktr.srmuniv.ac.in

² Department of Electrical and Electronics Engineering, SRM Institute of Science and Technology, Chennai 603 203, India

³ Department of Energy Technology, Aalborg University, 6700 Esbjerg, Denmark; san@et.aau.dk

⁴ Faculty of Electrical Engineering, Wroclaw University of Science and Technology, 50370 Wroclaw, Poland; zbigniew.leonowicz@pwr.edu.pl

* Correspondence: srinath.bktr@gmail.com (S.B.); bharatiraja@gmail.com (B.C.)

† These authors contributed equally to this work.

Received: 7 September 2018; Accepted: 5 November 2018; Published: 15 November 2018

Abstract: Multiple supply voltage is the most prevalent method for low power reduction in the design of modern Integrated circuits. Floorplanning process in this design performs positioning of functional blocks in the layout satisfying both fixed outline and voltage island constraints. The floorplans while satisfying these two significant constraints causes significant rise in wirelength and congestion. In this paper, a congestion and wirelength aware floorplanning algorithm is proposed which allows effective placement of functional blocks in the layout to satisfying fixed outline and voltage island constraints simultaneously. To perform voltage island floorplanning, the proposed algorithm uses Skewed binary tree representation scheme to operate the functional blocks in its predefined voltage level. The proposed methodology determines the feasible dimensions of the functional blocks in the representation which aids the placement process for the reduction of congestion and wirelength. With these optimal dimensions of the functional blocks, floorplanning is also performed for the layouts of aspect 1:1, 2:1, and 3:1, to evaluate the ability of proposed algorithm for satisfying the fixed outline constraint. The proposed methodology is implemented in the layout of InternationalWorkshop on Logic and Synthesis (IWLS) benchmarks circuits for experimental purpose. The resulting floorplans were iteratively optimized for optimal reduction of wirelength and congestion. Experimental results show that the proposed methodology outperforms existing state-of-the-art approaches in wirelength reduction by about 18.65% and in congestion reduction by around 63%, while delivering the 30.35% power consumption.

Keywords: low power; multiple supply voltage; fixed outline floorplanning; voltage island floorplanning

1. Introduction

Design of low power integrated circuits (IC) has become a challenging problem due to the continuous technology scaling. In the design of chips for modern applications like solar [1,2], motor control [3,4] and smart energy systems [5], power consumption plays a significant role. Comparatively, the contribution of dynamic power due to the frequent switching action of more number of transistors is more pronounced in these ICs to static power dissipation. Deployment of multiple supply voltages in the ICs is one of the best-known technique for the reduction of dynamic power consumption [6]. In this technique, the modules in the chip operate at different voltages, in addition to the chip level voltage [7]. Even though it helps in power savings, this introduces a lot of new challenges in the

physical design process. Electronic design automation industries follow a unique design flow to reduce the complexities of the design process. The Figure 1 depicts the state-of-art design flow for multiple supply voltages (MSV) designs. It involves three essential phases (a) voltage assignment, (b) insertion of level shifters, and (c) floorplanning [8,9].

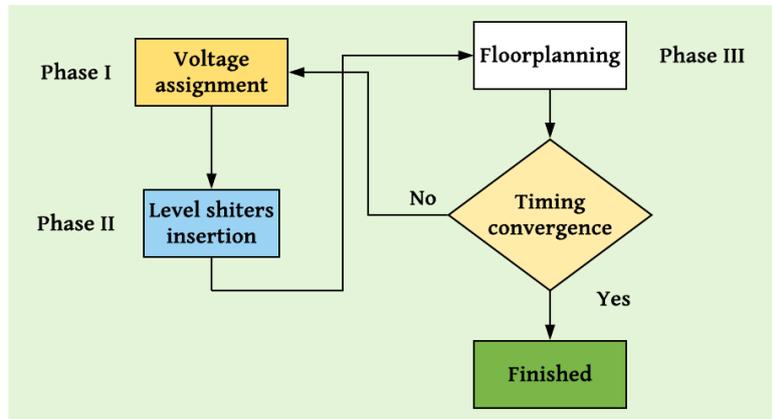


Figure 1. Multiple supply voltage design flow.

To meet the demand on miniature ICs, the researchers focus on fixed-outline floorplanning considering reduction of wirelength [10]. Increase in number of voltage islands increases the routing source and complicates power planning [11]. Among the various methods, voltage island-based floorplanning provides a better solution with minimization of power routing resource in the layout [12]. In this method, the floorplanner places the modules of similar operating voltage in a region, named as voltage island. Usually, this voltage island constraint-based floorplanning appears either in post floorplanning/placement stage. The voltage assignment phase determines the suitable operating voltage of the modules in the layout by examining the delay and power characteristic for the predefined set of voltages [13–24]. In some cases, the decision making for the selection of appropriate becomes tough due to the best performance of modules in a group of voltage levels. Lee et al. presents a method which considers reconvergence fan-out for optimal voltage assignment [8]. As the modification, this work, many papers focus on providing a solution for voltage assignment and floorplanning. Sengupta et al. proposed an algorithm for core-based designs and contributed a power state model approach which aids for voltage assignment based on IDLE, ON, OFF states of the cores in the chip. Some of the papers give a significant contribution to the place of level shifters to satisfy the timing constraint [25].

After the process of voltage assignment, these algorithms perform floorplanning using a data structure available for single supply voltages. The methodologies present in the literature follow the implementation of algorithms either in Gigascale Systems Research Center (GSRC) or Microelectronics Center of North Carolina (MCNC) benchmarks. These algorithms mainly consider wirelength and time complexity as the Figure of merit for evaluating the effectiveness of the methodology. The academic floorplanners use structures like B* tree [26], Normalized polish expression (NPE) [27,28], Transitive closure graph (TCG) [29], Sequence pair (SP) [30], and Skewed binary tree (SKB) [31] to represent the modules in the initial floorplan. Further, upon perturbations and using packaging methodologies for compactness results in a reduction of dead space and wire length. For fast convergence and accurate results, simulated annealing framework is employed with a cost function, consisting of wirelength and deadspace. Some of the other approaches include perturbation of modules in the layout and random positioning and legalization using mathematical formulations. Quadratic and second-cone programming methodologies provide a notable reduction in wirelength in these MSV designs [32]. Even though these academic benchmarks show better results in GSRC and MCNC circuits, their performance becomes typical after implementation in commercial Electronic design automation (EDA) tools like CADENCE, SYNOPSIS, etc.

Some of the notable facts after implementation of academic floorplanners are:

1. In the design of small size chips for various applications, the floorplanning processes require positioning of modules in the layout within the predefined application-specific aspect ratio. After implication of an application specific aspect ratio in an industry tool, the synthesis process it comparatively changes dimensions of the blocks inside the chip from its default area. This transformation is due to the high-level synthesis process involved in the EDA tool and results in the availability of unplaced modules outside the core region of the layout.
2. Increase in geometric violations while performing design rule check.
3. Avoids setup and hold time conditions.
4. Fails to consider the placement of standard cells and other challenges in the physical design after detailed routing.

In addition, conventional design MSV flow shown in Figure 1 is completely different in a EDA tool. The Figure 2 shows design flow for MSV design in Cadence Innovus software for digital designs.

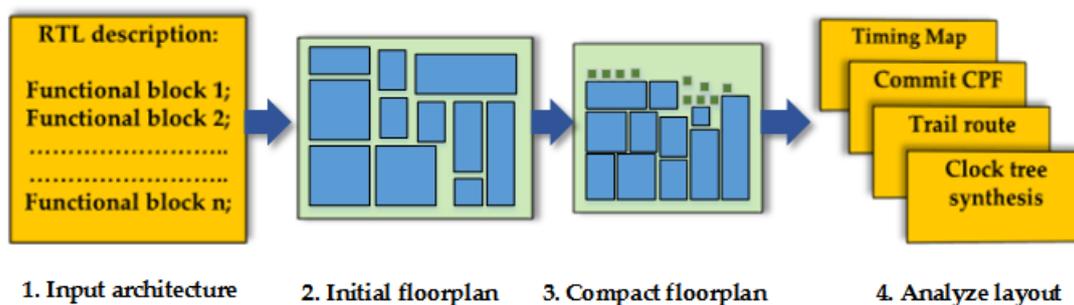


Figure 2. Design flow incorporated in Cadence tool for MSV.

This paper devotes to overcome the challenges as mentioned above while implementing MSV design floorplanning algorithm in an EDA tool. The main contributions in this paper are listed below:

- **Solution to industrial floorplanning:** This paper implements the conventional SKB tree meant for academic benchmarks in industrial EDA environment. While implementation of conventional SKB methodology for different aspect ratios, some of the modules are unable to position inside the core area of chip. This is paper overcomes this problem using the proposed algorithm through finding the optimal dimensions of modules in the design.
- **Elimination of cluster constraint:** While fixed-outline floorplanning for various aspect ratios of the chip, there are more number of unplaced modules outside the core area of the layout. The proposed algorithm places those modules inside the core area of the floorplan with short wirelength and reduces the percentage of congestion.
- **Avoiding re-voltage assignment:** This paper performs iterative improvement in positioning of modules inside the voltage island, this results in reduction of wirelength, congestion and power. The conventional SKB tree performs exchange of modules both intra and inter voltage island modes. This process leads to change of assigned voltage to modules and increases wirelength. Since the proposed algorithm accommodates all its modules inside the voltage island, the perturbations were performed within the voltage island. Through iterative improvement, the proposed algorithm reduces wirelength and congestion.
- **Unoccupied space for post-floorplanning stages:** Unlike the previous works, the proposed algorithm performs concurrent fixed-outline floorplanning and voltage island floorplanning for the minimization of wirelength and power consumption under various aspect ratio. The results show a notable contribution to power saving and wirelength compared to state-of-art methods.

The rest of this paper is organized as follows: Section 2 presents the existing SKB tree methodology for floorplanning with voltage island and fixed outline constraints. Section 3 gives the proposed

algorithm and a placement methodology to formulate the problem while floorplanning in EDA tool. Section 4 reports the experimental reports. Finally, Section 5 concludes this paper.

2. Preliminaries

This paper proposes a algorithmic method which is based on the existing the Skewed Binary (SKB) representation in [31]. Hence this section first reviews floorplanning topology using SKB and Section 3 presents proposed floorplanning methodology to manage fixed outline and voltage island floorplanning.

2.1. Floorplanning Representation

Consider the tree structure shown in Figure 3. Modules operating at similar voltage levels are placed at the parent node, one after another, in the left side of every node in a branch. Thus, they form a voltage island. Each node contains modules on the right side of the parent node. The parent node in the tree structure is placed in the left-corner of the voltage island. Every parent node of the branch is connected to the root node due to its inter connectivity. Thus, each branch will have modules at different operating voltages.

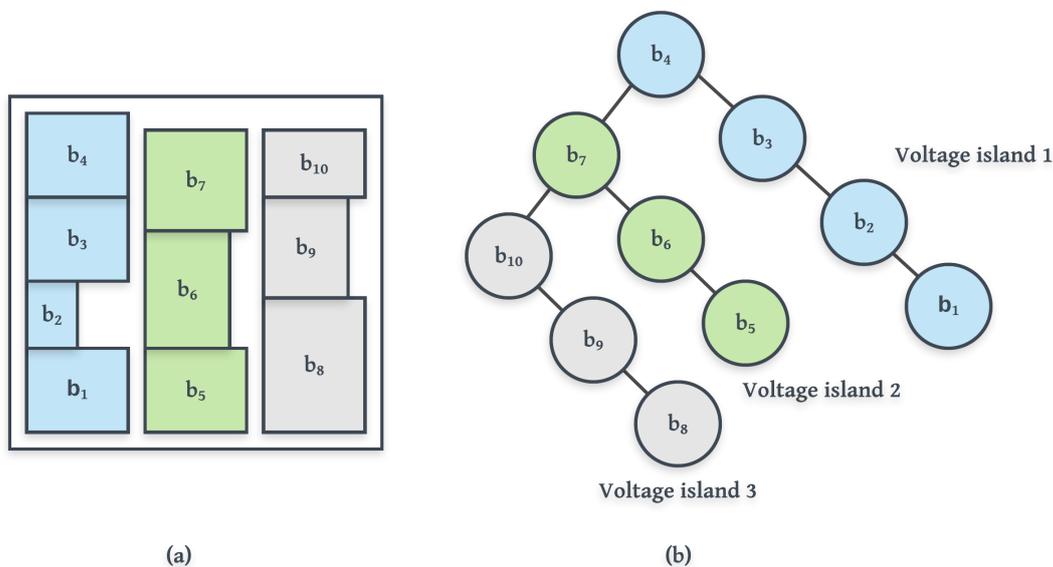


Figure 3. Example floorplan. (a) Floorplan after application of SKB tree (b) SKB tree structure of the floorplan.

Figure 3b illustrates a typical SKB tree representation and Figure 3a shows the floorplan of a circuit consisting of 10 modules. Modules $\{n_1, n_2, n_3, n_4\}$ operate at voltage level-1 and $\{n_5, n_6, n_7\}$ at voltage level-2. Similarly, $\{n_8, n_9, n_{10}\}$ modules work at voltage level-3. n_1 is considered as a parent node at level-0 since it needs to be placed in the left corner of the floorplan. Similarly, modules n_5 and n_8 represents the parent node for levels 1 and 2, and they are placed in the left corner of their power domain.

In this way, SKB tree helps the designer with the placement of modules in the core area of a chip. In the event of tree traversal, the time complexity is reduced for the placement of modules in the core area of the chip.

2.2. Placement Process

Consider a chip, with N number of modules in its design. Let $N = \{n_1, n_2, \dots, n_i\}$, where $n_1, n_2, n_3, \dots, n_i$ are the modules in the design and $M = \{m_1, m_2, m_3, \dots, m_j\}$ is a set containing voltage islands in N . With the dimension of modules in width and height, the tree structure in Figure 3

is used to allocate a space in the floorplan, based upon its operating voltage. Using the conventional depth first search algorithm, the modules are filled in the voltage island region one after another. The width of the power domain is calculated as

$$W_i = \frac{a_i}{a_t} W_c (1 + \gamma) \tag{1}$$

In the above Equation (1), a_i refers to the total area of modules in the power domain, a_t refers to the total area of the chip, W_c refers to the width of the chip and γ refers to the allowable dead-space.

Before a module is placed in a voltage island region, the algorithm determines the fitness of a module in the allotted width of the voltage island W_i . If the module width $n_i(w_i)$ is greater than W_i , the algorithm pushes in the n_i queue and traverses the tree for the insertion of the next module $n(i + 1)$ in the m_j region. Before the placement of module $n(i + 2)$ of m_j , the algorithm prioritizes the modules in the queue. If the module fails to fill the region, several modules would be combined as a merged block to fill up the region. For the successful placement of that merged module, a contour is drawn, unlike in [26], to estimate the feasible placement of the module. This procedure is iterative, until all the modules in the queue and voltage islands are placed in its region of m_j .

Figure 4 depicts the placement process of modules inside the voltage island. In Figure 4, the width of the module b_4 is greater than the width of voltage island W_1 . This can be avoided through merging the modules b_4, b_5 , as illustrated in Figure 5.

To avoid the congestion and aforementioned merging of modules, cluster constraint is introduced which serves as an upper bound on the density of modules in the voltage island. Thus, the algorithm limits the number of modules in voltage islands and produces a compact floorplan.

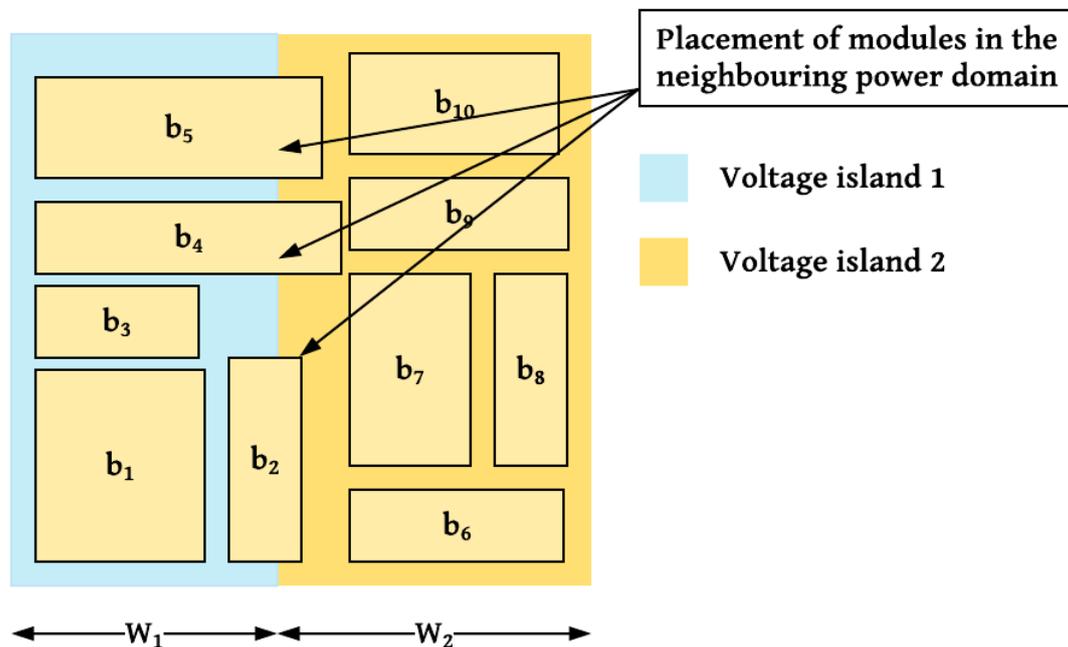


Figure 4. Placement of modules in the neighbouring power domains.

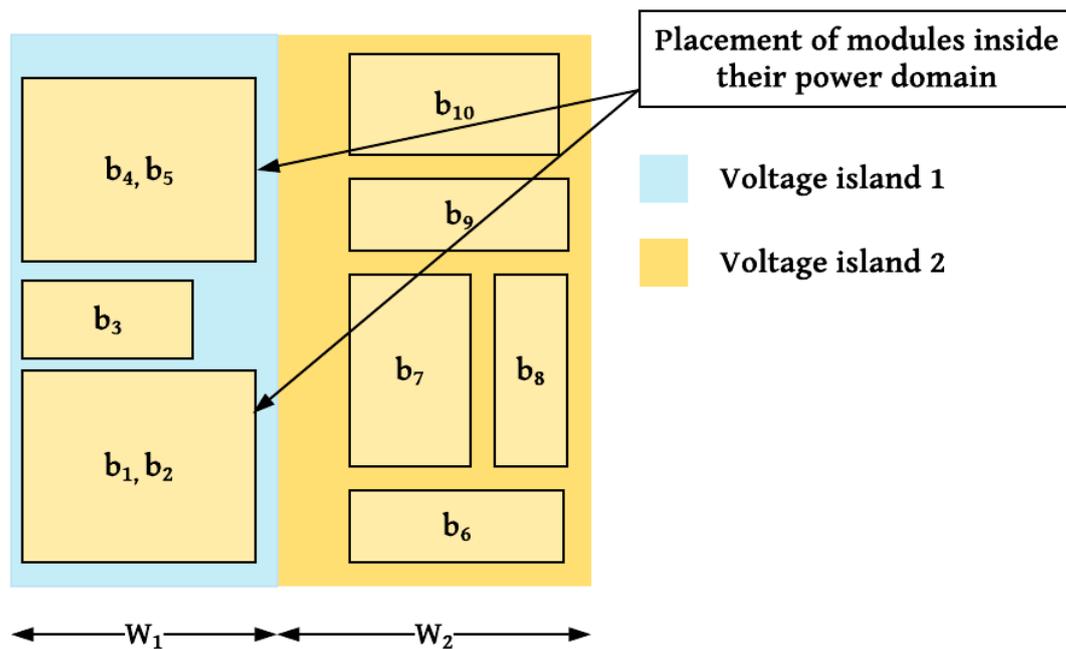


Figure 5. Placement of modules in the power domains.

2.3. Our Contributions

This paper proposes a floorplanning methodology based on SKB tree representation and performs voltage island floorplanning eliminating the cluster constraint in the conventional SKB tree methodology. Unlike the conventional SKB tree, the proposed methodology also predetermines the width of voltage island using the Equation (1). For tight packing of modules within this pre-estimated island width, this paper proposes an algorithm which determines the optimal dimensions of modules. Placement of these modules with their optimal dimensions positions all the modules of voltage island inside this width. However, for the reduction of power, congestion and wirelength, this procedure repeated iteratively. Different from conventional SKB methodology, the proposed methodology is implemented in Cadence EDA tool for the synthesized netlist of IWLS benchmarks to provide solutions to the industrial floorplanning problems. In addition to this voltage island floorplanning, proposed algorithm is also performed for fixed outline floorplanning, and simultaneous fixed and voltage island floorplanning.

Simulations were on both conventional SKB and proposed floorplanning methods using Cadence EDA tool. To compare the effectiveness of the proposed algorithm, four different experiments were performed in this paper. In first experiment for floorplanning with zero deadspace, the proposed methodology gives 18% of reduction in wirelength while satisfying the fixed outline constraint. Second experiment is performed for floorplanning with 15% deadspace on various aspect ratios of chips. The results reveals that the proposed floorplanning method gives average reduction of 29.3% and 63% in wirelength and congestion. In third experiment the proposed algorithm performs voltage island floorplanning with 38.7% reduction of power. Different from conventional SKB method, we perform an experiment to test ability of proposed algorithm to satisfy both fixed and voltage island floorplanning simultaneously. The results show that proposed algorithm provides reduces 22% of power consumption and increases the percentage of power saving compared with conventional SKB tree method.

3. Proposed Methodology

3.1. Step 1: Initial Floorplan

Given the dimensions of modules-in-chip in width and height as well as the operating voltage, we assign the modules in each node of the SKB tree structure. Our proposed floorplanning algorithm overcomes the drawbacks in the SKB tree algorithm for the multiple supply voltage (MSV) design of voltage island floorplanning, without any change in the area of the module. Table 1 presents the parameters used in our proposed Algorithms 1 and 2.

Table 1. Parameter description in Algorithms 1 and 2.

Parameter	Description
T	tree structure as shown in Figure 3b
$level$	levels in T
M	set containing all the modules in the design
m_i, m_j	two consecutive modules in a level
w_i, h_i	width and height of module m_i
w_j, h_j	width and height of module m_j

Given a module-based design, with all its module dimensions in width and height, and its operating voltage, a tree structure is obtained from the initial floorplan as shown in Figure 3. Modules are arranged in tree structure T , as explained in the Section 2.1.

3.2. Step 2: Optimal Dimension of Modules

After the arrangement of modules in the tree structure, the optimal dimensions of modules are estimated to fit all the modules in its voltage island. The optimal dimension is obtained using the Algorithms 1 and 2.

Given a tree structure, T , with nodes representing modules, M , and levels of the tree representing voltage levels, we obtain the optimal dimensions for the modules m_i and m_j , using the procedure $Opt_dimension(m_i, m_j)$, where $m_i, m_j \in M$. Finally, after obtaining the optimal dimensions, it is updated to the existing dimensions of m_i and m_j .

The Algorithm 2 describes the procedure to find the optimal dimensions of the modules. Lines (3–14) in the algorithm aims to find the optimal dimension of m_i and m_j . Stock–Meyer’s equation for vertical bisection used for non-slicing methodology is used to find the optimal dimensions of m_i and m_j . It computes the list of possible dimensions, for a given module size in width and height.

Consider the two modules m_i and m_j with dimensions (w_i, h_i) and (w_j, h_j) present in a voltage island. The proposed algorithm in this paper determines the optimal dimensions of the modules without changing the area of the module. Hence, the algorithm assumes duplicate dimensions of modules m_i and m_j as t_1 and t_2 . The dimension of t_1 is assumed as $t_1(w_i) = h_i$ for width which is height of module m_i while $t_1(h_i) = w_i$ for height which is actually the width of module m_i . Similarly, for module m_j its duplicate is assumed as $t_2(w_j) = h_j$ and $t_2(h_j) = w_j$. The proposed algorithm finds the optimal area required to floorplan the modules m_i and m_j , both width-wise and breath-wise, using the Equations (2)–(7). The Equation (2) helps to determine the optimal dimension required to floorplan two modules with its original dimensions (w_i, h_i) and (w_j, h_j) . In this Equation (2), u denotes the optimal width and v denotes optimal height after merging the modules m_i and m_j .

$$(u, v) = [w_i + w_j, \max\{h_i, h_j\}] \quad (2)$$

For minimum area floorplan, the algorithm compares value of u and v . If the value of u is greater than v , optimal dimension is determined using duplicate dimensions of module m_i . It is given by the Equation (3),

$$(u, v) = [t_1(h_i) + h_j, \max\{t_1(w_i), h_j\}] \quad (3)$$

Algorithm 1 Initialize(T)

```

1 A Tree  $T$ , with nodes representing modules in the design,
2  $level \leftarrow 1$ , where  $level \in T$ 
3 for  $level$  do
4    $M \in T; m_i, m_j \in M;$ 
5   Choose  $m_i$  and  $m_j;$ 
6    $Opt\_dimension(m_i, m_j);$ 
7   update  $m_i$  and  $m_j;$ 
8 end for

```

Algorithm 2 Opt_dimension(m_i, m_j)

```

1  $S = \emptyset; P = \emptyset;$  // initialize the set  $S$  and  $P;$ 
2 top;
3 while  $m_i, m_j$  do // for module  $m_i, m_j$  find the optimal dimensions length-wise;
4    $(u, v) = [w_i + w_j, \max\{h_i, h_j\}];$  // compute the dimension after merging;
5    $S+ = (u, v);$  // store all the resulting dimensions in set  $S;$ 
6   if  $u > v$  then
7      $(u, v) = [t_1(h_i) + h_j, \max\{t_1(w_i), h_j\}];$  // Compute the new dimensions using  $t_1;$ 
8      $S+ = (u, v);$  // store all the resulting dimensions in set  $S;$ 
9   else
10     $(u, v) = [t_2(h_j) + h_i, \max\{t_2(w_j), h_i\}];$  // Compute the new dimensions using  $t_1;$ 
11     $S+ = (u, v);$  // store all the resulting dimensions in set  $S;$ 
12  end if
13 end while
14 find minimum of  $S;$  // Find the minimum dimension in set  $S;$ 
15 while  $m_i, m_j$  do // for module  $m_i, m_j$  find the optimal dimensions width-wise;
16    $(u, v) = [\max\{m_i(w_i), m_j(w_j)\}, m_i(h_i) + m_j(h_j)];$  // compute the dimension after merging;
17    $P+ = (u, v);$  // store all the resulting dimensions in set  $S;$ 
18   if  $u > v$  then
19      $(u, v) = [\max\{t_1(h_i), m_j(h_j)\}, t_1(w_i) + m_j(h_j)];$  // Compute the new dimensions using  $t_1;$ 
20      $P+ = (u, v);$ 
21   else
22      $(u, v) = [\max\{t_2(h_j), m_i(h_i)\}, t_2(w_j) + m_i(h_i)];$  // Compute the new dimensions using  $t_2;$ 
23      $P+ = (u, v);$ 
24   end if
25 end while
26 find minimum of  $P;$  // Find the minimum dimension in set  $P;$ 
27 find minimum ( $S, P$ ); // Find the minimum dimension in set  $S$  and  $P;$ 
28 update minimum area dimension to  $m_i, m_j;$  // update these optimal dimensions in  $m_i, m_j;$ 
29  $i \leftarrow i + 1;$ 
30  $j \leftarrow j + 1;$ 
31 goto top;

```

For u less than v , Equation (4) is used to find the optimal dimensions to occupy the modules m_i and m_j .

$$(u, v) = [t_2(h_j) + h_i, \max\{t_2(w_j), h_i\}]; \tag{4}$$

The resulting dimensions from Equations (2)–(4) are stored in a set S .

In the above Equation (4), the algorithm finds the possibilities for optimal dimension with duplicate dimensions of module m_j . Thus, the algorithm finds the optimal dimension width-wise. In the same way, the algorithm uses the Equations (5)–(7) to find the height-wise optimal dimensions.

$$(u, v) = [\max\{w_i, w_j\}, h_i + h_j] \tag{5}$$

$$(u, v) = [\max\{t_1(h_i), h_j\}, t_1(w_i) + h_j] \tag{6}$$

$$(u, v) = [\max\{t_2(h_j), h_i\}, t_2(w_j) + h_i] \tag{7}$$

the resulting dimensions from Equations (5)–(7) are stored in a set P .

A minimal area dimension area is chosen from the set S and P and we update the dimensions of modules m_i and m_j .

Repeating this process with all the modules in the voltage islands, results in placement of all the modules within pre-estimated width of W_i . This helps the proposed algorithm for the placement of modules in the layout eliminating cluster constraint introduced in the conventional SKB tree methodology.

3.3. Example for Determining Optimal Dimensions of Modules

For better understanding of proposed methodology simple illustration is given below for determining optimal dimension of modules in the first level of the SKB tree shown in Figure 6. It constructed for a design with three voltage islands consisting of eight modules.

The nodes in the tree represents the modules in the layout and the levels of the tree represents voltage domains. The dimensions present above every node are the width and height of the modules. The proposed algorithm in this paper performs the following procedure and determines the optimal dimensions of modules in the tree structure.

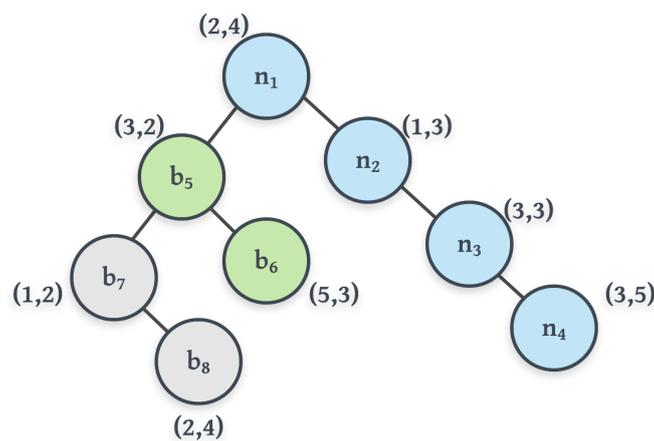


Figure 6. Example SKB tree with three power domains.

From the Figure 6, let $N = \{n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8\}$ are the modules with dimensions (width, height) in micro metres.

The algorithm proceeds with the following steps given below.

1. Taking the vertices n_4 and n_3 . Let $U = \{(3, 5), (5, 3)\}$, $V = \{(3, 3)\}$.

For the vertical orientation,

- (i) join $u_{11} = (3, 5)$ and $v_{11} = (3, 3)$, we get $(3 + 3, \max\{5, 3\}) = (6, 5)$. Since the maximum is from U , we join u_{12} and u_{11} .
- (ii) join $u_{12} = (5, 3)$ and $v_{11} = (3, 3)$, we get $(5 + 3, \max\{3, 3\}) = (8, 3)$

The resulting dimensions are $(6, 5) = 30 \mu\text{m}^2$ and $(8, 3) = 24 \mu\text{m}^2$. The minimum of these is $24 \mu\text{m}^2$.

For the horizontal orientation,

- (i) join $u_{11} = (3, 5)$ and $v_{11} = (3, 3)$, we get $(\max\{3, 3\}, 5 + 3) = (3, 8)$. Since the maximum is from V , we join u_{12} and u_{11} .

The resulting dimensions is $24 \mu\text{m}^2$. As a result, the dimensions either $[(3, 5)$ and $(3, 3)]$ or $[(5, 3)$ and $(3, 3)]$ can be taken for minimum area.

2. Taking the vertices $\{n_4, n_3\}(3, 8)$ and $n_2(1, 3)$, Let $U = \{(3, 8), (8, 3)\}$, $V = \{(1, 3), (3, 1)\}$. For the vertical orientation,

- (i) join $u_{11} = (3, 8)$ and $v_{11} = (1, 3)$, we get $(3 + 1, \max\{8, 3\}) = (4, 8)$. Since the maximum is from V , we join u_{11} and v_{12} .
- (ii) join $u_{11} = (3, 8)$ and $v_{12} = (3, 1)$, we get $(3 + 3, \max\{8, 1\}) = (6, 8)$

The resulting dimensions are $(4, 8) = 32 \mu\text{m}^2$ and $(6, 8) = 48 \mu\text{m}^2$. The minimum of these is $32 \mu\text{m}^2$.

For the horizontal orientation,

- (i) join $u_{11} = (3, 8)$ and $v_{11} = (1, 3)$, we get $(\max\{3, 1\}, 8 + 3) = (3, 11)$. Since the maximum is from V , we join u_{11} and v_{12} .
- (ii) join $u_{11} = (3, 8)$ and $v_{12} = (3, 1)$, we get $(\max\{3, 3\}, 8 + 1) = (3, 9)$.

The resulting dimensions are $33 \mu\text{m}^2$ and $27 \mu\text{m}^2$. The minimum of these dimensions is $27 \mu\text{m}^2$. Finally, the minimum of dimensions horizontal $27 \mu\text{m}^2$ and vertical $32 \mu\text{m}^2$ is taken as the dimensions. The minimum of these is, $27 \mu\text{m}^2$. Its co-ordinates are $(3, 8)$ and $(3, 1)$. The area of the merged modules $\{n_4, n_3\}$ is fixed as $(3, 8)$.

3. Taking the vertices $\{n_4, n_3, n_2\}(3, 9)$ and $n_1(2, 4)$.

Let $U = \{(3, 9), (9, 3)\}$, $V = \{(2, 4), (4, 2)\}$ For the vertical orientation,

- (i) join $u_{11} = (3, 9)$ and $v_{11} = (2, 4)$, we get $(3 + 2, \max\{9, 4\}) = (5, 9)$. Since the maximum is from V , we join u_{11} and v_{12} .
- (ii) join $u_{11} = (3, 9)$ and $v_{12} = (4, 2)$, we get $(3 + 4, \max\{9, 2\}) = (7, 9)$

The resulting dimensions are $(5, 9) = 45 \mu\text{m}^2$ and $(7, 9) = 63 \mu\text{m}^2$. The minimum of these is $45 \mu\text{m}^2$.

For the horizontal orientation,

- (i) join $u_{11} = (3, 9)$ and $v_{11} = (2, 4)$, we get $(\max\{3, 2\}, 9 + 4) = (3, 13)$. Since the maximum is from V , we join u_{11} and v_{12} .
- (ii) join $u_{11} = (3, 9)$ and $v_{12} = (4, 2)$, we get $(\max\{3, 4\}, 9 + 2) = (4, 11)$.

The resulting dimensions are $39 \mu\text{m}^2$ and $44 \mu\text{m}^2$. The minimum of these dimensions is $39 \mu\text{m}^2$.

From the $(3, 13) = 39 \mu\text{m}^2$, if $W_p = 13$, the procedure given below is followed i.e., the update on the dimensions of the modules is performed, else, a new module is taken from the tree structure

and again the procedure for horizontal and vertical orientation is repeated until the resulting merged dimension is equal to W_p .

Finally, the minimum of dimensions horizontal $39 \mu\text{m}^2$ and vertical $45 \mu\text{m}^2$ is taken as the dimensions. The minimum of these is, $39 \mu\text{m}^2$. Its co-ordinates are (3, 9) and (2, 4).

The area of the merged modules $\{n_4, n_3, n_2\}$ is fixed as (3, 9). For the modules $\{n_4, n_3\}$ and n_2 , the updated dimensions are (3, 8) and (3, 1).

Now, for (3, 8), the modules n_4 and n_3 holds the dimensions as (3, 5) and (3, 3).

Thus the optimum dimension for voltage island 1 is found to be $39 \mu\text{m}^2$.

3.4. Step 3: Placement

After updating the dimensions in modules for all voltage islands using depth first search, the modules are placed in the core area of layout from the lower left corner within the pre-estimated width. If a module failed to fit in the width W_i , it is pushed in a temporary queue and priority is given for the next module. Before the placement of any module inside the voltage island, modules in the queue are prioritized to check the feasibility of placement within the size of W_i .

Presence of unoccupied space in the layout support placement of standard cells and offers reduction congestion while routing. However, unoccupied space present after placement and routing increases the silicon cost during manufacturing of chip. Hence, this paper analyses the unoccupied space while floorplanning the modules in the layout. Figure 7 illustrates the proposed methodology for placement of modules with reduction of unoccupied space. In this illustration, the pre-estimated width of voltage island W_i is assumed as 8 units.

Consider the three modules $m_i(3,8)$, $m_j(4,4)$, and $n_i(1,3)$. Figure 7a–d shows the possible placement of modules. Figure 7a shows the placement of module n_i over the module m_i . Even though this placement fits inside the width W_i with total occupied area of 63 square units, it increases the unoccupied space. The Figure 7b shows a placement when n_i is vertically placed with respect to module m_i . This further increases the unoccupied space compared with placement method in Figure 7a with occupied area of 64 square units. The floorplan in Figure 7c also depicts a notable unoccupied space with placement of module n_i adjacent to module m_i and the module m_j on top it. The total occupied area of module including unoccupied space is 56 square units. Figure 7d show the compact floorplan with reduction in occupied area of 48 square units; this floorplan reduces the silicon cost through allocating space for the placement of the new module after m_j . Hence, a unoccupied space analysis is performed in this paper before placement of every module inside voltage island to aid the process in post-floorplanning stages.

In order to obtain optimal reduction in wirelength, congestion and power, a cost function is developed in this paper. Every candidate floorplan is evaluated to the cost function given below in Equation (8), where P denotes power in mW, WL denotes wirelength in μm and congestion in percentage,

$$\text{Cost}(F) = WL + C + P \quad (8)$$

To obtain an optimized floorplan, iterative improvement methodology is incorporated in this paper. For every iteration, we alter the positions of node inside the voltage of SKB tree representation. Then, the proposed methodology is implemented on the tree structure which results in optimal dimensions. Based on this optimal dimension, the floorplan is evaluated for reduction in unoccupied space. The conventional physical design flow given in Figure 2 is followed to obtain the total power, wirelength and congestion. This procedure is repeated until the cost function is minimized.

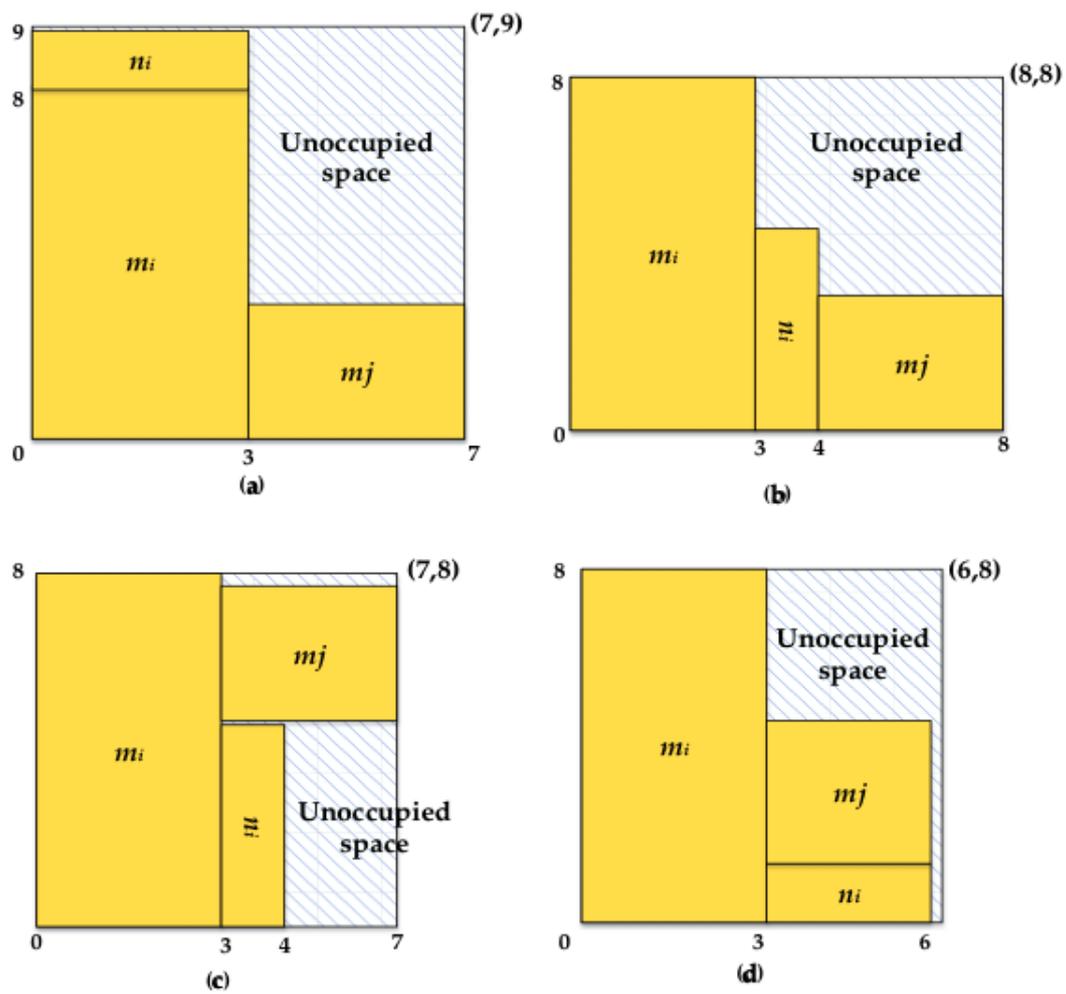


Figure 7. Proposed placement methodology. (a) Positioning of n_i above m_i , (b) Unoccupied space in the floorplan due to placement of m_j adjacent to n_i , (c) Unoccupied space created due to the placement of new module m_j above n_i , (d) Compact placement process incorporated in proposed algorithm.

4. Simulation Results

This section presents four experimental results for fixed-outline floorplanning and voltage island floorplanning using proposed methodology. The proposed methodology was implemented using Cadence Innovus system for 45 nm technology. To prove the efficiency proposed methodology, IWLS benchmark circuits with high logically density are chosen. For better comparison, simulations were performed both proposed algorithm and conventional SKB floorplanning. Table 2 details the logical structures of each data set in IWLS benchmarks. The number of blocks in these circuits ranges from 15 to 28, and the number of nets ranges from 19,000 to 60,000, approximately.

Table 2. IWLS Benchmark circuit description.

# Circuit	# Function	# Sequential	# inverter	# Buffer	# Logic	# Total
<i>AES_core</i>	AES cipher	530	5589	274	14,402	20,795
<i>DMA_core</i>	DMA Controller	2192	2678	253	13,995	19,118
<i>RISC_core</i>	RISC CPU	7599	7370	126	44,872	59,974
<i>256tapFIRfilter</i>	FIR fliter	2520	1789	76	11,175	15,500

4.1. Comparison of Wirelength With Existing Fixed Outline Floorplanners for Zero Deadspace

In this experiment, the proposed floorplanning methodology is implemented with zero dead space as fixed outline constraint. Before implementation of proposed methodology, we first study the performance of conventional SKB tree with results from academic benchmarks circuits towards reduction of wirelength to satisfy fixed outline constraint. The simulations are performed using the design flow given in Figure 8. The Table 3 show comparison results of Analytical floorplanning (A-FP) [33], Parquet 4.0 [34], Partitioning to optimize module arrangement (PATOMA) [35], Simulated Annealing based Fixed-outline Floorplanner with the Optimal Area (SAFFOA) [36] and SKB-tree [31] using GSRC benchmarks with average reduction in wirelength of 41.96%. The results show that SKB tree performs better compared to existing floorplanners. After implementation of conventional SKB tree in EDA environment for circuit netlist of IWLS benchmarks, the results shows that proposed floorplanning methodology gives 18% of reduction in wirelength to satisfy the fixed outline constraint. Even though performance of conventional SKB tree is better with academic benchmarks, results from EDA environment considers placement of standard cells and it requires more routing resources. This increases the total wirelength due to more number of nets. However, simulation results show that proposed methodology gives better results even with this increase in number of nets.

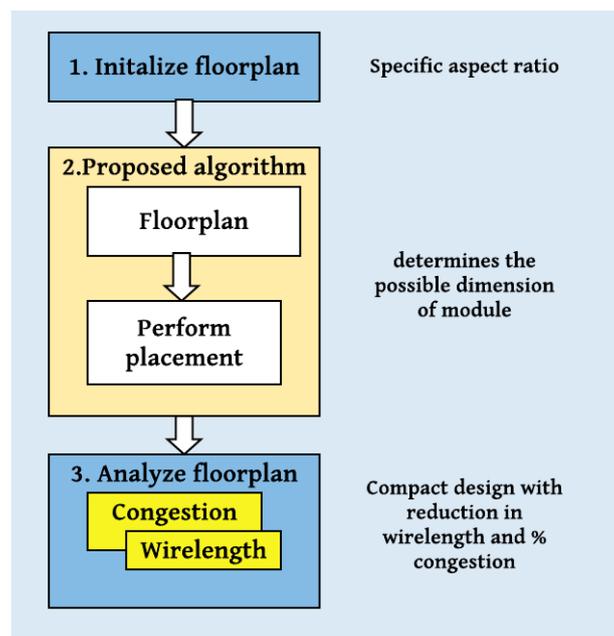


Figure 8. Cadence design flow incorporated to analyze congestion and wirelength.

4.2. Comparisons with Conventional Floorplanner for Fixed Outline Floorplanning with 15% Deadspace on Various Aspect Ratios of Chips

This subsection presents experimental results of proposed method for fixed-outline floorplanning with 15% allowable deadspace on aspect ratios 1:1, 2:1, and 3:1 and the Table 4 shows simulation results. In this experiment, in addition to wirelength, the percentage of congestion in the layout was measured on these aspect ratios. From the Table 4, it is evident that compared with existing SKB floorplanning technique, our proposed method reduces wirelength by 20%, 28%, and 40% when the aspect ratios of the chips are 1:1, 2:1, and 3:1 respectively. For congestion, the proposed methodology can reduce 30%, 17%, and 13% for the layout with aspect ratios 1:1, 2:1, and 3:1. This reduction in wirelength is mainly due to the iterative positioning of blocks in the layout using the proposed method for determining its optimal dimension and reducing unoccupied space. The Figures 9 and 10 shows the layout of *AES_core*, *RISC_core*, and 256 tap FIR filter after proposed algorithm under various aspect ratio.

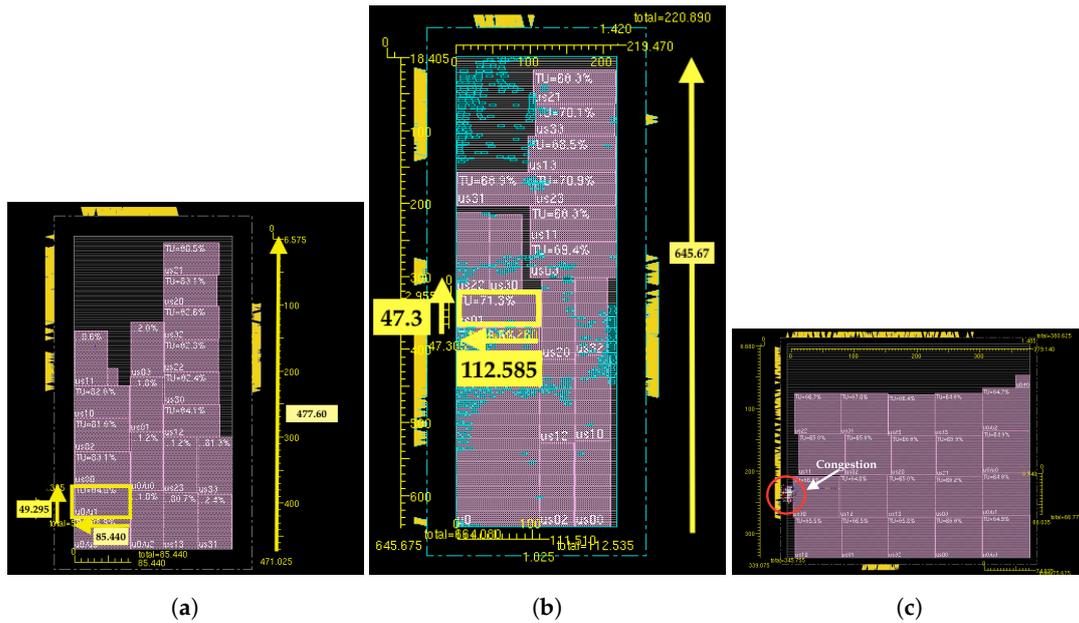


Figure 9. Floorplans of AES_core after proposed algorithm for aspect ratio 2:1, 3:1, and 1:1. (a) 2:1 aspect ratio; (b) 3:1 aspect ratio; (c) 1:1 aspect ratio.

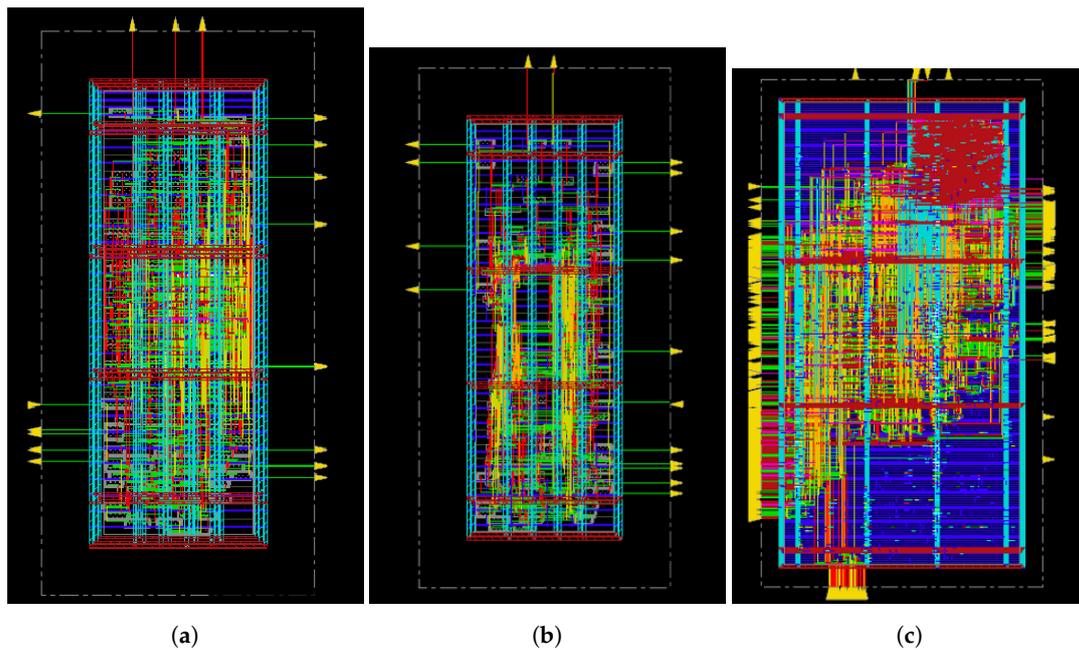


Figure 10. Floorplans of RISC_core and 256 tap FIR filter after proposed algorithm under various aspect ratio. (a) 256 tap FIR with 3:1 with one voltage island; (b) 256 tap FIR filter in 2:1 with one voltage island; (c) 3:1 layout of RISC_core with two voltage islands.

4.3. Comparisons with Existing Works for Voltage Island Floorplanning

The third set of experiments deals with floorplanning to satisfy, voltage island floorplanning. As in the Section 4.1, we first study the results of various existing methodologies in the literature addressing this problem and the results are tabulated in Table 5. The columns (3–6) in this table shows the results of voltage island floorplanning approach and conventional SKB tree techniques with one and two voltage island designs performed on GSRC benchmark circuits. In the Table 5, the columns (9–12) presents the results after implementation of exiting SKB floorplanning and proposed method on IWLS benchmarks. The experiments performed using GSRC benchmarks selects the operating voltage of modules in the

floorplan from a set 1.5, 1.3, 1.2, 1.1, 1.0 V randomly, where 1.5 V is the chip level voltage. However, since the proposed methodology is a technology dependent netlist, the best operating voltage is found using the method described in [19]. Thus voltage assignment is performed for the modules in the layout. In the Table 5, column 9 refers to the number of voltage islands. Here VI = 1 refers to operating voltage of 1.1 V for the whole circuit, and VI = 2 refers to two voltage island in the layout, where voltage domain one is operating at 1.1 V while voltage domain two at 0.9 V, considering a chip voltage of 1.1 V. Since most of the ICs used for various applications operates with only two voltage islands, the experiments were performed for only to two voltage, even though an increase in the number of voltage islands will show significant power reduction. The Figure 11 show the skeleton of the common power format used in Cadence Innovus tool to declare the power domains. In the Table 5, power saving % denotes the amount of power reduction due to the increase in voltage island. The results show that voltage island floorplanning consumes 25% power more than conventional SKB tree method while SKB tree gives 98% of power saving. This high percentage of power saving is mainly due to voltage assignment to the modules in the layout and better reduction in wirelength. After implementation in industrial EDA tool the results in the columns (9–12) shows that conventional SKB method consumes 38.7% percentage of power with the proposed method comparatively. Also, the proposed method better power saving with increase in voltage island compared conventional SKB floorplanning. This is due to the application of dynamic programming during voltage assignment to the modules and conditional floorplanning of modules which results in the reduction of wirelength.

4.4. Simultaneous Voltage Island and Fixed-Outline Floorplanning

In this fourth set of experiment, the proposed algorithm is performed for fixed-outline floorplanning with 2:1 and 3:1 aspect ratios of the chip while satisfying the voltage island floorplanning constraint simultaneously. The Table 6 shows the simulation results of IWLS benchmarks both conventional SKB tree and the proposed algorithm. The voltage assignment is performed to the modules in the layouts as in [19]. As mentioned in previous Section 4.3, the layout has 1.1 V as chip level voltage and 0.9 V and 1.1 V for two power domains. Common power format (CPF) file is written based on the voltage assignment and simulations were performed as given in the design flow shown in the Figure 11 for 15% deadspace. For 2:1 aspect ratio, the proposed algorithm provides reduces 22% of power consumption and increases the percentage of power saving compared with conventional SKB tree method. Similarly, the simulations results for 3:1 aspect ratio, reveals that the proposed algorithm reduces the power by 14% compared with conventional SKB tree methodology.

Table 3. Comparison of other floorplanners for fixed outline floorplanning.

Circuit	#blocks/#nets	Parquet [34]	PATOMA [35]	SAFFOA [36]	SKB [31]	Circuit	SKB	Proposed
n30	30/349	166,336	156,921	138,218	102,579	AES_core (n30/1102)	595,000	500,000
n50	50/485	189,121	180,115	165,366	129,916	DMA_core (n50/4174)	910,000	861,000
n10	10/118	50,690	5228	46,207	36,175	256 tap FIR (n10/666)	119,576	96,457
						RISC_core (n18/9756)	484,300	318,200
Average WL ratio		1.51	1.44	1.3	1	Average WL ratio	1.187	1

Table 4. Comparisons with conventional floorplanner for fixed outline floorplanning with 15% deadspace on various aspect ratios of chips.

Circuit	#VI	1:1 Aspect Ratio				2:1 Aspect Ratio				3:1 Aspect Ratio			
		SKB		Proposed		SKB		Proposed		SKB		Proposed	
		WL(um)	Cong.%	WL(um)	Cong.%	WL(um)	Cong.%	WL(um)	Cong.%	WL(um)	Cong.%	WL(um)	Cong.%
AES_core	1	595,000	3.245	547,000	2.253	495,000	5.9	390,000	4.431	375,000	8.52	347,000	9.305
	2	558,000	4.065	500,000	3.416	582,000	7.266	429,000	5.124	582,000	12.29	456,000	8.945
DMA_core	1	872,000	2.195	861,000	2.101	982,000	2.75	912,000	2.422	1,982,000	3.039	1,021,000	2.047
	2	910,000	3.825	813,000	2.407	790,000	3.331	578,000	2.921	900,000	5.881	546,000	5.329
256 tap FIR	1	975,499	0.712	964,570	0.45	950,900	0.85	878,340	0.883	1,015,100	1.67	996,210	1.485
	2	915,766	0.536	535,860	0.268	882,000	2.8	643,760	1.748	1,003,100	3.68	822,780	2.591
RISC_core	1	784,436	2.001	457,610	1.187	798,321	7.518	483,863	6.223	882,750	10.284	545,012	9.747
	2	545,660	4.784	446,073	2.653	586,538	7.92	398,593	7.786	660,800	12.785	545,068	10.863
Average ratio		1.201	1.302	1	1	1.28	1.17	1	1	1.401	1.133	1	1

Table 5. Comparisons with previous works for Voltage island floorplanning.

Circuit /Nets	#VI	Voltage Island Driven Floorplanning [28]		SKB [31]		Circuit /Nets	#VI	Conventional SKB		Proposed	
		Power (mW)	Power Saving %	Power (mW)	Power Saving %			Power (mW)	Power Saving %	Power (mW)	Power Saving %
n30/349	1	469,330	0	469,330	0	AES_core /(n30/1102)	1	3.79	0	3.13	0
	2	312,071	33.51	312,071	33.51		2	3.29	13.19	2.12	32.33
n50/485	1	446,803	0	446,803	0	DMA_core /(n50/4174)	1	7.26	0	6.17	0
	2	265,290	40.62	263,633	40.99		2	6.705	7.71	4.86	21.25
n10/118	1	498,778	0	498,778	0	256 tap FIR /(n10/666)	1	4.37	0	3.23	0
	2	393,660	21.08	352,637	29.3		2	3.36	23.11	2.233	30.86
						RISC_core /(n18/9756)	1	4.776	0	2.375	0
							2	3.975	16.77	1.924	18.98
Average ratio		1.25	0.98	1	1	Average ratio		1.387		1	

Table 6. Comparison of power for simultaneous fixed outline floorplanning and voltage island floorplanning with conventional SKB floorplanning.

Circuit /Nets	#VI	2:1 Aspect Ratio		3:1 Aspect Ratio	
		Conventional SKB	Proposed	Conventional SKB	Proposed
AES_core (n30/1102)	1	4.97	2.97	5.13	3.59
	2	4.23	2.58	4.027	2.75
DMA_core (n50/4174)	1	6.21	6.138	9.21	8.762
	2	4.86	5.3	5.986	6.86
256 tap FIR filter (n10/666)	1	3.19	2.278	3.37	2.556
	2	3.13	2.229	3.18	2.114
RISC_core (n18/9756)	1	3.573	2.466	4.124	3.46
	2	3.368	2.465	3.787	2.759
Average ratio		1.22	1	1.146	1

```

set_cpf_version 1.1
set_design <design>
set_hierarchy_separator "/"

# creation of power nets
create_power_nets -nets VDD1 -voltage 1.1
create_power_nets -nets VDD2 -voltage 0.9
create_ground_nets -nets VSS

#creation of power domains
create_power_domain -name PD1 -default
update_power_domain -name PD1 -primary_power_net VDD1 -primary_ground_net VSS
create_power_domain -name PD2 -instances update_power_domain -name PD2
                        -primary_power_net VDD2 -primary_ground_net VSS

create_nominal_condition -name high -voltage 1.1
update_nominal_condition -name high -library_set 1v08_max
create_nominal_condition -name low -voltage 0.9
update_nominal_condition -name low -library_set 0v9_max

#power modes
create_power_mode -name PM_Default \
    -domain_conditions {PD1@high PD2@low \
    -default
update_power_mode -name PM_Default -sdc_files sdc_file.sdc

#define global connections
create_global_connection -domain PD1 -net VDD1 -pins VDD
create_global_connection -domain PD1 -net VSS -pins VSS
create_global_connection -domain PD2 -net VDD2 -pins VDDG
create_global_connection -domain PD2 -net VDD3 -pins VDD
create_global_connection -domain PD2 -net VSS -pins VSS

#create operating corners and analysis view
create_operating_corner -name 1v08_max_oc -voltage 1.1 -library_set 1v08_max
    -temperature 125 -process 1
create_operating_corner -name 0v9_max_oc -voltage 0.9 -library_set 0v9_max
    -temperature 125 -process 1
create_analysis_view -name AV_Default -mode PM_Default \
    -domain_corners {PD1@slow_1v0 PD2@slow_1v2}

#level shifters rules
create_level_shifter_rule -name shifter_rule1 -from PD1 -to PD2
update_level_shifter_rules -names shifter_rule1 -location to -prefix ls_PD12
create_level_shifter_rule -name shifter_rule2 -from PD2 -to PD1
update_level_shifter_rules -names shifter_rule2 -location to -prefix ls_PD21

end_design

```

Figure 11. Common power format used in simulation for the declaration of power domains in the layout.

5. Conclusions

In this paper, a new paradigm has been presented for industrial floorplanning to satisfy both fixed outline and voltage island constraint. Since the proposed methodology shows better results towards power saving and reducing wirelength, the experiment is extended for simultaneous optimization of

both constraints. The experimental results demonstrates that the proposed algorithm in this paper is effective in solving industrial floorplanning under various fixed outline and voltage island conditions.

For future SOC designs, three-dimensional structures are preferred due to its ability to place these two-dimensional layouts in stack-based structures and more integration of modules in the chip. Since fixed outline and voltage island floorplanning is challenging problem, there exist few works considering this issue. The proposed methodology requires dedicated voltage assignment and floorplanning, which will be considered as future work.

Author Contributions: S.B. and A.P. has developed the proposed research concept and developed the complete study with implemented results and wrote the article; B.C. and S.P. as contributed his experience in the field of Electronics and development of systems in Integrated circuits and involved in development, verification and providing suggestion to wrote the article; Z.L. involved in the work to set the task in numerical study and assisted with authors to implement and wrote the article as co-authors.

Funding: This research receives no external funding.

Acknowledgments: This work is supported by center for Bio energy and Green Engineering, Department of Energy technology, Aalborg University, Esbjerg, Denmark, for insight technical information received for this research activities.

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

References

1. Chokkalingam, B.; Padmanaban, S.; Siano, P.; Krishnamoorthy, R.; Selvaraj, R. Real-time forecasting of EV charging station scheduling for smart energy systems. *Energies* **2017**, *10*, 377. [[CrossRef](#)]
2. Kamalpathi, K.; Priyadarshi, N.; Padmanaban, S.; Holm-Nielsen, J.; Azam, F.; Umayal, C.; Ramachandaramurthy, V. A Hybrid Moth-Flame Fuzzy Logic Controller Based Integrated Cuk Converter Fed Brushless DC Motor for Power Factor Correction. *Electronics* **2018**, *7*, 288. [[CrossRef](#)]
3. Bharatiraja, C.; Jeevananthan, S.; Latha, R. FPGA based practical implementation of NPC-MLI with SVPWM for an autonomous operation PV system with capacitor balancing. *Int. J. Electr. Power Energy Syst.* **2014**, *61*, 489–509. [[CrossRef](#)]
4. Chokkalingham, B.; Padmanaban, S.; Blaabjerg, F. Investigation and Comparative Analysis of Advanced PWM Techniques for Three-Phase Three-Level NPC-MLI Drives. *Electr. Power Compon. Syst.* **2018**, *46*, 258–269. [[CrossRef](#)]
5. Mathew, M.; Babu, N.R.; Suresh, M.; Sanjeevikumar, P. High Response Photon-Counting for Phase Fraction Measurement Using Compact-RIO with FPGA. In *Advances in Systems, Control and Automation*; Springer: Singapore, 2018; pp. 133–137.
6. Usami, K.; Ishikawa, T.; Kanazawa, M.; Kotani, H. Low-power design technique for ASICs by partially reducing supply voltage. In Proceedings of the Ninth Annual IEEE International ASIC Conference and Exhibit, Rochester, NY, USA, 23–27 September 1996.
7. Cheng, C.H. Using a Voltage Domain Programmable Technique for Low-Power Management Cell-Based Design. *J. Low Power Electron. Appl.* **2011**, *1*, 303–326. [[CrossRef](#)]
8. Lee, W.P.; Liu, H.Y.; Chang, Y.W. Voltage-Island Partitioning and Floorplanning Under Timing Constraints. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.* **2009**, *28*, 690–702. [[CrossRef](#)]
9. Srinath, B.; Arunapriya, P. A Modified Algorithm for Voltage Assignment and Floorplanning of SOC Designs. *Indian J. Sci. Technol.* **2015**, *8*. [[CrossRef](#)]
10. Adya, S.N.; Markov, I.L. Fixed-outline floorplanning through better local search. In Proceedings of the 2001 IEEE International Conference on Computer Design: VLSI in Computers and Processors, Austin, TX, USA, 23–26 September 2001; pp. 328–334. [[CrossRef](#)]
11. Chang, J.M.; Pedram, M. Energy minimization using multiple supply voltages. In Proceedings of the 1996 International Symposium on Low Power Electronics and Design, Monterey, CA, USA, 12–14 August 1996; pp. 157–162. [[CrossRef](#)]
12. Hu, J.; Shin, Y.; Dhanwada, N.; Marculescu, R. Architecting voltage islands in core-based system-on-a-chip designs. In Proceedings of the 2004 International Symposium on Low Power Electronics and Design (IEEE Cat. No.04TH8758), Newport Beach, CA, USA, 9–11 August 2004; pp. 180–185. [[CrossRef](#)]

13. Chen, Z.; Wang, G.; Dong, C. Multiple voltage assignment based on PSO. In Proceedings of the 2011 International Conference on Computer Science and Network Technology, Harbin, China, 24–26 December 2011; Volume 2, pp. 1295–1298. [[CrossRef](#)]
14. Wang, Y.; Cheng, C. A hierarchy multiple-voltage design technique for low-power performance-manageable bio-chips. In Proceedings of the 2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Nantou, Taiwan, 27–29 May 2016; pp. 1–2.
15. Chi, J.C.; Lee, H.H.; Tsai, S.H.; Chi, M.C. Gate Level Multiple Supply Voltage Assignment Algorithm for Power Optimization Under Timing Constraint. *IEEE Trans. Very Large Scale Integr. Syst.* **2007**, *15*, 637–648. [[CrossRef](#)]
16. Wong, M.D.F. Low power design with multi-V_{dd} and voltage islands (Abstract). In Proceedings of the 2007 7th International Conference on ASIC, Guilin, China, 22–25 October 2007; pp. 1325–1325. [[CrossRef](#)]
17. Amelifard, B.; Pedram, M. Optimal Design of the Power-Delivery Network for Multiple Voltage-Island System-on-Chips. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.* **2009**, *28*, 888–900. [[CrossRef](#)]
18. Du, S.; Xia, Y.; Chu, Z.; Qi, L. Generic algorithm based voltage Islands-Aware Multiple Voltage Assignment. In Proceedings of the 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, China, 22–24 October 2010; Volume 5, pp. 623–627. [[CrossRef](#)]
19. Yu, B.; Dong, S.; Goto, S. Multi-voltage and level-shifter assignment driven floorplanning. In Proceedings of the 2009 IEEE 8th International Conference on ASIC, Changsha, China, 20–23 October 2009; pp. 1264–1267. [[CrossRef](#)]
20. Ma, Y.; Qiu, X.; He, X.; Hong, X. Incremental power optimization for multiple supply voltage design. In Proceedings of the 2009 10th International Symposium on Quality Electronic Design, San Jose, CA, USA, 16–18 March 2009; pp. 280–286. [[CrossRef](#)]
21. Muthumala, W.H.; Hariyama, M.; Kameyama, M. GA-Based Assignment of Supply and Threshold Voltages and Interconnection Simplification for Low Power VLSI Design. In Proceedings of the APCCAS 2006-2006 IEEE Asia Pacific Conference on Circuits and Systems, Singapore, 4–7 December 2006; pp. 1264–1267. [[CrossRef](#)]
22. Shimin, D.; Yinshui, X.; Runping, Y.; Xiangsheng, Y. Genetic algorithm-based multiple voltages assignment considering the power network complexity and level shifters. In Proceedings of the 2012 International Conference on Computer Science and Information Processing (CSIP), Xi'an, China, 24–26 August 2012; pp. 652–655. [[CrossRef](#)]
23. Hao, C.; Wang, N.; Yoshimura, T. A Unified Scheduling Approach for Power and Resource Optimization With Multiple V_{dd} or/and V_{th} in High-Level Synthesis. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.* **2017**, *36*, 2030–2043. [[CrossRef](#)]
24. Blaauw, D.; Devgan, A.; Najm, F. Leakage power: Trends, analysis and avoidance. In Proceedings of the ASP-DAC 2005. Asia and South Pacific Design Automation Conference, Shanghai, China, 21 January 2005; Volume 1. [[CrossRef](#)]
25. Sengupta, D.; Saleh, R.A. Application-Driven Voltage-Island Partitioning for Low-Power System-on-Chip Design. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.* **2009**, *28*, 316–326. [[CrossRef](#)]
26. Chang, Y.C.; Chang, Y.W.; Wu, G.M.; Wu, S.W. B*-Trees: A New Representation for Non-Slicing Floorplans. In Proceedings of the Design Automation Conference, Los Angeles, CA, USA, 5–9 June 2000.
27. Ma, Q.; Qian, Z.; Young, E.F.Y.; Zhou, H. MSV-Driven Floorplanning. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.* **2011**, *30*, 1152–1162. [[CrossRef](#)]
28. Ma, Q.; Young, E.F.Y. Multivoltage Floorplan Design. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.* **2010**, *29*, 607–617. [[CrossRef](#)]
29. Lin, J.M.; Chang, Y.W. TCG: A transitive closure graph-based representation for non-slicing floorplans. In Proceedings of the 38th Annual Design Automation Conference, Las Vegas, NV, USA, 18–22 June 2001; pp. 764–769. [[CrossRef](#)]
30. Sengupta, D.; Veneris, A.; Wilton, S.; Ivanov, A.; Saleh, R. Sequence pair based voltage island floorplanning. In Proceedings of the 2011 International Green Computing Conference and Workshops, Orlando, FL, USA, 25–28 July 2011; pp. 1–6. [[CrossRef](#)]
31. Lin, J.M.; Hung, Z.X. SKB-Tree: A Fixed-Outline Driven Representation for Modern Floorplanning Problems. *IEEE Trans. Very Large Scale Integr. Syst.* **2012**, *20*, 473–484. [[CrossRef](#)]

32. Luo, C.; Anjos, M.F.; Vannelli, A. Large-scale fixed-outline floorplanning design using convex optimization techniques. In Proceedings of the 2008 Asia and South Pacific Design Automation Conference, Seoul, Korea, 21–24 January 2008; pp. 198–203. [[CrossRef](#)]
33. Zhan, Y.; Feng, Y.; Sapatnekar, S.S. A fixed-die floorplanning algorithm using an analytical approach. In Proceedings of the Asia and South Pacific Conference on Design Automation, Yokohama, Japan, 24–27 January 2006. [[CrossRef](#)]
34. Adya, S.N.; Markov, I.L. Fixed-outline floorplanning: Enabling hierarchical design. *IEEE Trans. Very Large Scale Integr. Syst.* **2003**, *11*, 1120–1135. [[CrossRef](#)]
35. Cong, J.; Romesis, M.; Shinnerl, J.R. Fast floorplanning by look-ahead enabled recursive bipartitioning. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.* **2006**, *25*, 1719–1732. [[CrossRef](#)]
36. He, O.; Dong, S.; Bian, J.; Goto, S.; Cheng, C.K. A novel fixed-outline floorplanner with zero deadspace for hierarchical design. In Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, 10–13 November 2008; pp. 16–23. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).