Performance-driven Analog Placement Considering Boundary Constraint

Cheng-Wu Lin, Jai-Ming Lin, Chun-Po Huang and Soon-Jyh Chang
Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C.
lcw@sscas.ee.ncku.edu.tw; jmlin@ee.ncku.edu.tw; gppo@sscas.ee.ncku.edu.tw; soon@mail.ncku.edu.tw

ABSTRACT

To reduce parasitic mismatches in analog design, we usually care about the property of symmetric placement for symmetry groups, which would form several symmetry islands in a chip. However, routing is greatly affected by placement results. If modules with input or output ports are placed arbitrarily in a symmetry island, the routing wires, which connect these modules with other modules outside the island, may induce unwanted parasitics coupling to signals, and thus circuit performance is deteriorated. This phenomenon can not be identified by a cost function, which only considers placement area and total wire length. Therefore, we would like to introduce the necessity of considering boundary constraint for the modules with input or output ports in symmetry islands. Based on ASF-B* tree [3], we explore the feasible conditions for 1D and 2D symmetry islands to meet this constraint. Further, a procedure is presented to maintain the feasibility for each ASF-B* tree after perturbation. Experimental results show that our approach guarantees the boundary property for the modules with input or output ports in symmetry islands.

Categories and Subject Descriptors: B.7.2 [Integrated Circuits]: Design Aids – Layout, Placement and Routing

General Terms: Algorithms, Design

Keywords: Analog placement, symmetry, boundary constraint

1. INTRODUCTION

In analog or mixed-signal layout design, circuit performances are sensitive to the parasitic mismatches caused by process variation or thermal gradient. To reduce unwanted parasitic mismatches and improve circuit performances, designers would place matched devices symmetrically in layouts [1, 2]. Besides, in order to obtain better performance, they also place the matched devices belonging to the same sub-circuit close to each other, in which a symmetry group is formed. Thus, Lin and Lin [3] recently introduced the concept of symmetry islands for placement of symmetry groups to ensure better electrical properties, such as parasitic matching and thermal gradient. By their definition, each module should abut at least one of other modules in the same group. Hence the modules of a symmetry group would form a connected placement, i.e. a symmetry island. Figures 1 shows the architecture of a mixed-signal circuit --- pipelined ADC. Depending on the converter resolution, a pipelined ADC usually consists of several conversion stages. Each stage includes several sub-circuits, such as the MDAC and sub-ADC. Because the sub-circuits are sensitive to parasitic mismatches, the matched devices in each sub-circuit should be placed symmetrically. Based on the concept of symmetry islands, the devices belonging to the same sub-circuit would form a symmetry island. Thus, the pipelined ADC contains several symmetry islands in the corresponding placement.

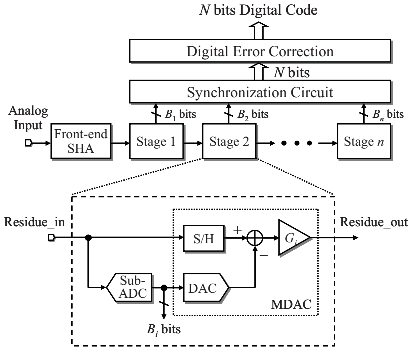


Figure 1. Architecture of a pipelined ADC.

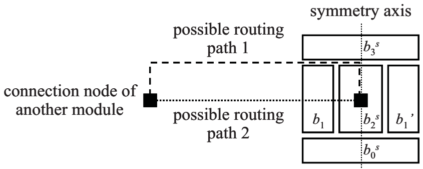


Figure 2. A symmetry island consists of one symmetry pair (b_1, b_1') and three self-symmetry modules b_0^s, b_2^s and b_3^s. The module b_2^s with an input or output port is placed inside the island.

Although several works [3, 5-12, 17] have studied analog placement, they emphasized the importance of symmetrical placement without considering actual routing problems. However, placement has a great impact on routing. Once the locations of devices are assigned, the routing paths are roughly determined. Thus, we need to consider the routing issue during placement to avoid some unwanted parasitics. For example, Figure 2 shows one symmetry island corresponding to a sub-circuit defined in Figure 1. In the figure, b_i^s denotes a self-symmetry module and (b_j, b_j') represents a symmetry pair. These modules are placed symmetrically with respect to the vertical axis. After all the sub-circuits in Figure 1 have been placed, we need routing to connect different symmetry islands, which would be affected by the modules with input or output ports in those islands. For simplicity, we call the modules with input or output ports as in-out modules in this paper. If the in-out modules are placed inside the island, the routing paths should be across or along other modules in the same island, as shown in Figure 2. These paths may induce additional parasitics to the symmetry island. Besides, if in-out modules are placed inside the island, the routing wires would be longer than that when the in-out modules are placed on the boundary of the island. Longer routing wires would induce more

DAC'10, June 13–18, 2010, Anaheim, California, USA
Copyright 2010 ACM 978-1-4503-0002-5 /10/06…$10.00

parasitics coupling to signal paths, and thus worsen circuit performances. Therefore, considering placement area and total wire length is not always sufficient to evaluate that condition. To further reduce the routing parasitics, designers should be better to place in-out modules on the boundaries of symmetry islands. We call it **symmetry-island boundary constraint** and consider it in analog and mixed-signal placement.

1.1 Previous Work

The problem of floorplanning/placement considering symmetry or boundary constraint has been extensively studied for decades. Most of these works used topological representations with simulated annealing algorithm [4] to tackle the problem. For the symmetry constraint, the symmetric-feasible conditions have been derived for several representations, including sequence-pairs [5], O-trees [6], binary trees [7], TCG [8, 9], and B*-trees [3, 10]. Lin and Lin [3] proposed the symmetry-island formulation to model a symmetric placement and introduced the ASF-B*-trees to guarantee symmetric placements after packing. Recently, two works based on CBL [11] and HB*-trees [12] further considered thermal effect with symmetry constraint in analog placement.

For the boundary constraint, several works have explored the feasible conditions based on different representations, such as sequence-pairs [13, 14], CBL [15], and B*-trees [16]. Tayu [14] and Lin et al. [16] applied a procedure to transform an infeasible solution into a feasible one and guarantee a feasible placement after each perturbation. Based on sequence-pairs, Tam et al. [17] used the dummy nodes and additional constraint edges to handle symmetry groups and boundary modules simultaneously in a placement. However, these works all consider the boundary constraint to limit modules placed on the boundary of a chip. Thus, none of existing works have considered boundary constraint for symmetry islands.

1.2 Our Contributions

In this paper, we first demonstrate the importance of the symmetry-island boundary constraint in analog design by implementing an op-amp layout and performing its post-layout simulation. The simulation results show that inappropriate placement of in-out modules in a symmetry island may induce more routing parasitics and cause worse circuit performances. In order to reduce the unwanted routing parasitics, we have to consider boundary constraint for symmetry islands. Thus, we extend the ASF-B*-trees [3] to handle boundary-constrained modules in 1D and 2D symmetric placements. Besides, we also present a method to transform an infeasible solution into feasible one with boundary constraint for each ASF-B*-tree. Experimental results show that our approach can effectively place in-out modules such that both symmetry and boundary constraints are satisfied in a symmetry island.

The remainder of this paper is organized as follows: Section 2 uses an actual design to demonstrate the performance deterioration caused by the locations of in-out modules in a symmetry island. Section 3 introduces how to consider boundary constraint for symmetry islands. Section 4 presents our placement algorithm based on the framework of HB*-trees and describes the approach to guarantee a feasible ASF-B*-tree in each perturbation. Section 5 reports the experimental results. Finally, Section 6 concludes this paper.

2. IN-OUT MODULES IN A SYMMETRY ISLAND

In this section, we would like to show the impact on circuit performance due to the in-out module locations in a symmetry island. We first implement two placements for an op-amp, and then compare their performances. From the comparison results, we conclude that it would induce more parasitics, if in-out modules are placed inside a symmetry island instead of being placed at boundary.

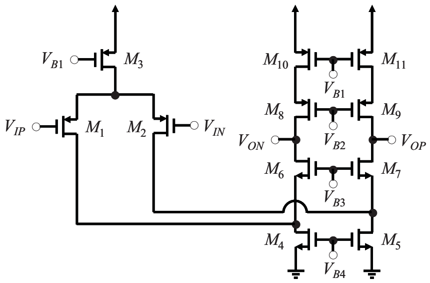


Figure 3. Schematic of a fully-differential folded-cascode op-amp.

Table 1. Component sizing results of the op-amp in Figure 3.

$V_{power\text{-}supply}$	1.80 V		W/L (μm/μm)	Multiplier
$V_{input\text{-}common\text{-}mode}$	0.90 V	M_1 (M_2)	4.35/0.36	34
$V_{output\text{-}common\text{-}mode}$	0.90 V	M_3	4.77/0.36	99
V_{B1}	1.21 V	M_4 (M_5)	3.70/0.36	17
V_{B2}	0.95 V	M_6 (M_7)	3.79/0.36	13
V_{B3}	0.87 V	M_8 (M_9)	4.87/0.36	50
V_{B4}	0.61 V	M_{10} (M_{11})	4.46/0.36	28

To demonstrate the phenomenon, we manually design the physical layout of an op-amp. This op-amp is applied for the front-end SHA or the first stage of an 8-bit 50-MS/s pipelined ADC. The resolution of the first stage is 1.5 bit. According to the ADC specifications, we can derive the op-amp specifications from a design procedure based on [21]. The op-amp specifications are 60-dB DC gain, 450-MHz unity-gain frequency and 60° phase margin with a 0.8-pF load capacitance. We select a fully-differential folded-cascode topology to design this op-amp. Figure 3 shows the core schematic of the op-amp without bias circuit and common-mode feedback circuit for the simplicity of layout. In the figure, M_i represents a MOS transistor and V_{Bi} denotes a bias voltage. V_{IP} (V_{OP}) and V_{IN} (V_{ON}) are biased at input (output) common-mode voltage for DC operation. We designed the op-amp in TSMC 0.18-μm CMOS technology. The component sizing results are listed in Table 1, including the voltage values of biases, the channel widths and lengths of transistors, and the multiples of transistors.

Before conducting the layout, we first identify the matching groups in the op-amp. Let {M_i, M_j} denote a matching group with two devices M_i and M_j, which should be placed symmetrically to each other. In Figure 3, the op-amp consists of five matching groups, {M_1, M_2}, {M_4, M_5}, {M_6, M_7}, {M_8, M_9} and {M_{10}, M_{11}}, and one single device M_3. We generate three self-symmetry modules for the matching groups {M_1, M_2}, {M_4, M_5} and {M_6, M_7} by interdigitated structure based on the stack-based approach [18] or the pattern-based approach [19]. The remaining matching groups are regarded as symmetry-pair modules. Let $b_{i\text{-}j}^{s}$ (b_i^s) denote a self-symmetry module corresponding to a matching group {M_i, M_j} (single device M_i). Let ($b_{i\text{-}j}$, $b_{i\text{-}j}'$) represent a symmetry-pair module corresponding to a matching group {M_i, M_j}. Thus, we have four self-symmetry modules, $b_{1\text{-}2}^{s}$, $b_{4\text{-}5}^{s}$, $b_{6\text{-}7}^{s}$ and b_3^s, and two symmetry-pair modules, ($b_{8\text{-}9}$, $b_{8\text{-}9}'$) and ($b_{10\text{-}11}$, $b_{10\text{-}11}'$), to construct a symmetry island.

As shown in Figure 4, we compare two placement results if an in-out module is placed at the boundary of symmetry island or not. Note that $b_{1\text{-}2}^{s}$ is an in-out module with input nodes connected to external signal source. In Figure 4(a), $b_{1\text{-}2}^{s}$ is placed inside the

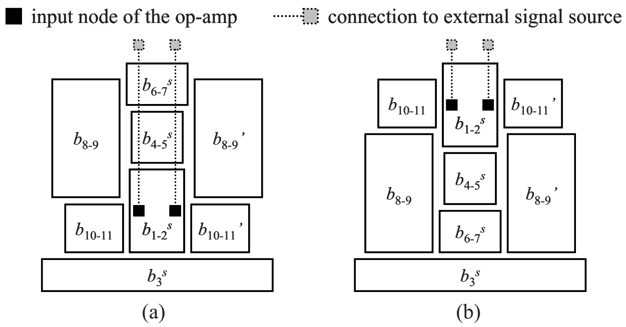


Figure 4. Two symmetric placements of the op-amp. The module with input ports is placed (a) inside the symmetry island (b) on the top boundary of the symmetry island.

Table 2. Circuit performances of the placements in Figure 4.

Post-layout Simulation		DC Gain	Unity-gain Frequency	Phase Margin
Test signal is injected directly into input node	Figure 4(a)	60.3 dB	469.3 MHz	62.8°
	Figure 4(b)	60.3 dB	469.6 MHz	62.8°
Test signal is injected from external source	Figure 4(a)	60.3 dB	465.2 MHz	60.5°
	Figure 4(b)	60.3 dB	467.2 MHz	61.3°

island while it is placed on the top boundary of the island in Figure 4(b). Conventionally, placement problem would apply a cost function, such as Equation (1) defined in Section 4, to select a better-quality placement. However, the two placements in Figure 4 have the same bounding-rectangle area. Besides, although the routing distance between $b_{1\text{-}2}^s$ and b_3^s in Figure 4(a) is shorter than that in Figure 4(b), the routing distance between $b_{1\text{-}2}^s$ and external signal source in Figure 4(a) is longer than that in Figure 4(b). The estimated total wire lengths of the two placements are nearly the same. Thus, it is hard to distinguish which placement is better according to the cost function.

To observe the circuit performances, we perform post-layout simulations by two conditions with adequate routing to the respective placements. In the first condition, the test signal is injected directly into the input nodes of module $b_{1\text{-}2}^s$. This case represents an ideal condition since no parasitics couple to the signal source. The simulation results of the first condition are listed in the upper half of Table 2. The circuit performances of the two placements are almost the same. The results show that different placements would not have much variance in circuit performances while no routing parasitics couple to input or output nodes. In the second condition, the test signal is injected into the external nodes and then passed to the input nodes of module $b_{1\text{-}2}^s$ through the routing wires. This case denotes an actual condition that the signal source comes from another module beyond the op-amp. The simulation results of the second condition are listed in the last two rows of Table 2. Compared to the first condition, the circuit performances are deteriorated because the routing wires between the input nodes and the external nodes would induce parasitics coupling to the signal source. The maximum performance deterioration of Figure 4(a) is 3.7%, but that of Figure 4(b) is only 2.4%. These results indicate that additional routing wires of the input or output nodes would induce parasitics coupling to the signal path and affect circuit performances. More critically, if in-out modules are placed inside a symmetry island, the routing wires, intended to connect the in-out modules with other modules outside the island, would be longer than that if they are placed on the boundary of the island. Longer routing wires would induce more parasitics coupling to signal paths, and thus worsen circuit performances.

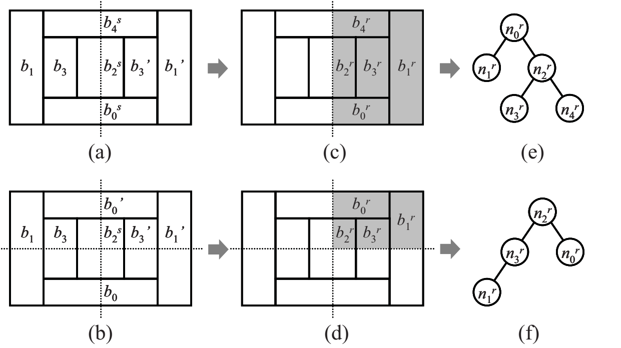


Figure 5. (a-b) Placement examples of a symmetry group in 1D and 2D symmetry respectively. (c-d) Selecting a representative for each symmetry pair and self-symmetry module. (e-f) The ASF-B*-trees representing the placements of the symmetry groups respectively.

These experiments have shown the importance of the in-out module location relative to a symmetry island. To shorten the connections between in-out modules and external modules and reduce the unwanted routing parasitics coupling to signal paths, we should place in-out modules on the boundary of the island. Thus, for analog placement, instead of only considering symmetry constraint, we should also think of **boundary constraint** for the in-out modules in each symmetry island.

3. SYMMETRY ISLAND CONSIDERING BOUNDARY CONSTRAINT

In this section, we would like to show how to consider boundary constraint in a symmetry island. We first review the representation of automatically symmetric-feasible B*-tree (ASF-B*-tree) [3], and then extend it to deal with boundary constraint. Based on the representation, the feasible conditions for boundary constraint in 1D and 2D symmetry islands will be fully explored.

3.1 Review of ASF-B*-tree

In analog layout design, the placement of a symmetry group has three major symmetry types: 1D vertical symmetry, 1D horizontal symmetry, and 2D symmetry. The placement for 1D vertical (horizontal) symmetry has the symmetry axis in the vertical (horizontal) direction, while that for 2D symmetry has symmetry axes in both directions. Figure 5(a) (Figure 5(b)) illustrates a placement example for 1D (2D) symmetry. As defined by [3], the representative b_i^r is the right half (the top-right quarter) of a self-symmetry module b_i^s in a 1D (2D) symmetric placement, or it can be the b_i' (the top half of b_i') of a symmetry pair (b_i, b_i') in a 1D (2D) symmetric placement. For 1D symmetry as shown in Figure 5(a), the b_1' of the symmetry pair (b_1, b_1') is selected as the representative b_1^r in Figure 5(c). For 2D symmetry as shown in Figure 5(b), the top-right quarter of the self-symmetry module b_2^s is served as the representative b_2^r in Figure 5(d). An ASF-B*-tree is a B*-tree in which each node n_i^r corresponds to a representative b_i^r. Figure 5(e) (Figure 5(f)) shows the ASF-B* tree corresponding to the placement in Figure 5(c) (Figure 5(d)). The packing of an ASF-B*-tree can automatically form a symmetry island.

3.2 1D Symmetry Island with Boundary Constraint

To place the in-out modules on the boundary of a symmetry island, we have to add additional constraints to ASF-B*-tree representation. First, we introduce how to consider boundary constraint in the representation for 1D symmetry islands.

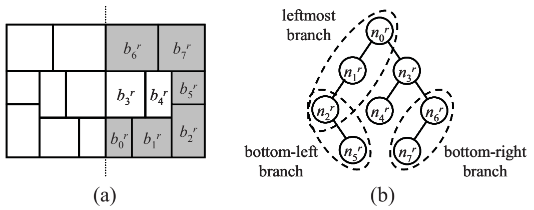


Figure 6. (a) A placement in 1D vertical symmetry. (b) The feasible conditions for a symmetry pair served as an in-out module.

For the packing of a 1D vertical symmetry island, we only need to consider the representatives on the right half of the placement plane, as illustrated in the previous subsection. In the placement plane, the representative of a symmetry pair can be placed anywhere except that represents an in-out module. The representative of an in-out module can only be placed on the bottom, or the right, or the top boundary of the right-half plane. Thus, if an in-out module is a symmetry pair, we have the following choices for the corresponding node in an ASF-B*-tree.

- **In the leftmost branch:** the representative would be placed on the bottom boundary.
- **In the bottom-left branch with the left child of each node in the branch being deleted:** the representative would be placed on the right boundary.
- **In the bottom-right branch with the right child of each node in the branch being deleted:** the representative would be placed on the top boundary.

Figure 6(a) shows the placement of a symmetry group S = {(b_0, b_0'), (b_1, b_1'), (b_2, b_2'), (b_3, b_3'), (b_4, b_4'), (b_5, b_5'), (b_6, b_6'), (b_7, b_7')} in 1D vertical symmetry. Figure 6(b) presents the feasible conditions for a symmetry pair served as an in-out module.

Because of the property of a 1D symmetry island, the representative b_i^r of a self-symmetry module b_i^s must abut to the symmetry axis; thus, it is placed on the left boundary of the placement plane. However, the representative of an in-out module needs to be placed on the boundaries except the left boundary of a placement plane. By considering these conditions, if the in-out module is a self-symmetry module, we have the following choices for the corresponding node in an ASF-B*-tree.

- **Being the root node or the bottom node in the rightmost branch:** the representative would be placed in the bottom-left corner or the top-left corner.
- **Being the node between two hierarchy nodes, and these nodes being arranged in a right-skewed branch with the left child for each node in the branch being deleted:** the representative would be placed on the left boundary and no module could be placed at its right side.

The former constraint provides only two possible positions for placing a self-symmetry module if it is served as an in-out module, while the latter constraint provides additional positions for placing the module. As shown in Figure 7(a), for the placement of a symmetry group S = {b_0^s, (b_1, b_1'), b_2^s, b_3^s, (b_4, b_4')} in 1D vertical symmetry, the feasible conditions for a self-symmetry module served as an in-out module is shown in Figure 7(b). If an in-out module would not be placed in the bottom-left or the top-left corner of a placement plane, the latter constraint provides another choice. In this case, we have to employ the hierarchical B*-tree (HB*-tree) representation [3] to obtain a feasible packing. For example, Figure 8 shows the placement of a symmetry group S = {(b_0, b_0'), (b_1, b_1'), (b_2, b_2'), (b_3, b_3'), b_4^s, (b_5, b_5'), (b_6, b_6'), (b_7, b_7'), (b_8, b_8')}, in which the self-symmetry module b_4^s represents an in-out module. To ensure that the position of b_4^s is on the boundary of the symmetry island, we enclose all the modules except b_4^s as two groups. Each group corresponds to a hierarchy node n_{Si} in a HB*-tree, and the node n_4^s which denotes b_4^s is placed between two hierarchy nodes n_{S0} and n_{S1}. No module will be placed at the right side of b_4^s in the corresponding symmetry island.

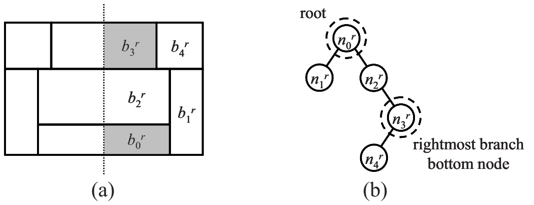


Figure 7. (a) The in-out module is a self-symmetry module placed in the bottom-left or the top-left corner. (b) The feasible conditions.

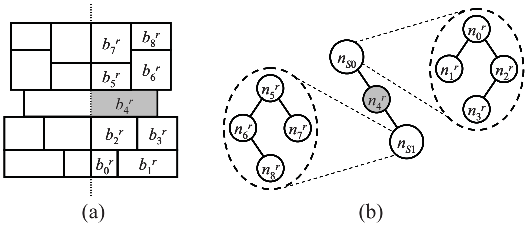


Figure 8. (a) The in-out module b_4^r is a self-symmetry module but not placed in the bottom-left or the top-left corner. (b) A HB*-tree is employed to deal with the feasible condition.

Similarly, we can derive the feasible conditions for a 1D horizontal symmetry island with boundary constraint. In this case, we only consider the top-half placement plane when packing an ASF-B*-tree.

3.3 2D Symmetry Island with Boundary Constraint

Now, we would like to introduce the ASF-B*-tree considering boundary constraint in a 2D symmetry island. For packing this kind of islands, we only need to consider the representatives on the top-right quarter of the placement plane.

Because the representative of a symmetry pair must abut to one of the symmetry axes in a 2D symmetry island, it should be placed on the left boundary or the bottom boundary of the placement plane. However, the representative of an in-out module needs to be placed on the right or the top boundary. Thus, if an in-out module is a symmetry pair, we have the boundary constraint for the corresponding node in an ASF-B*-tree as follows:

- **Being the bottom node in the rightmost branch or in the leftmost branch:** the representative would be placed in the top-left corner or the bottom-right corner.

This constraint provides only two possible positions for placing a symmetry pair served as an in-out module. It implies that there are at most two in-out modules in the form of symmetry pairs for a feasible 2D symmetry island with boundary constraint. Figure 9(a) shows the 2D symmetric placement of a symmetry group S = {b_0^s, (b_1, b_1'), (b_2, b_2'), (b_3, b_3')}. Figure 9(b) presents the feasible conditions for a symmetry pair served as an in-out module.

For a self-symmetry module in a 2D symmetry island, its representative must abut both symmetry axes and thus should be placed in the bottom-left corner of the placement plane. Nevertheless, the representative of an in-out module needs to be placed on the right or the top boundary. Therefore, if the in-out module is a self-symmetry module, we have the boundary constraint for the corresponding node in an ASF-B*-tree as follows:

- **Being the root node in a left-skewed branch or in a right-skewed branch:** the representative would be placed

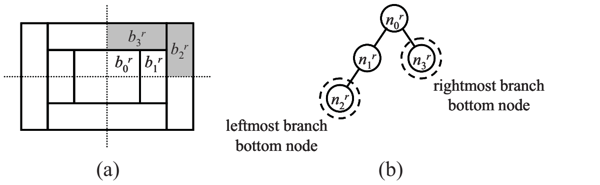


Figure 9. (a) A placement in 2D symmetry. (b) The feasible conditions for a symmetry pair served as an in-out module.

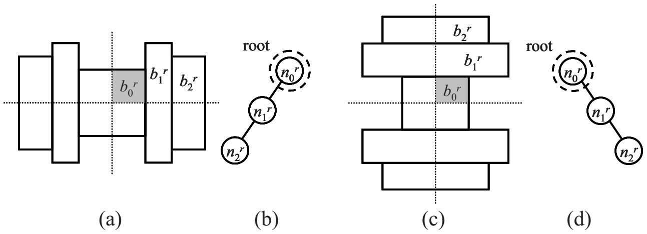


Figure 10. (a) A self-symmetry module without visible module placed at its top side. (b) The corresponding feasible condition. (c) A self-symmetry module without visible module placed at its right side. (d) The corresponding feasible condition.

in the bottom-left corner and no module could be placed at either its top side or its right side.

This constraint provides only one possible position for placing a self-symmetry module if it is an in-out module, which implies that only one in-out module can be a self-symmetry module for a feasible 2D symmetry island with boundary constraint. Figure 10(a) and 10(c) show two 2D symmetric placements of a symmetry group $S = \{b_0^s, (b_1, b_1'), (b_2, b_2')\}$. The corresponding feasible conditions for a self-symmetry module served as an in-out module are shown in Figure 10(b) and 10(d), respectively.

4. MAINTAINING A FEASIBLE ASF-B*-TREE

In this section, we first introduce our algorithm to consider boundary constraint for symmetry islands in analog placement. Then, we present a procedure to maintain a feasible ASF-B*-tree.

Given a set of device modules and symmetry-constrained matching groups, our objective is to obtain a placement P that satisfies the proposed symmetry-island boundary constraint and minimizes the cost function $\Phi(P)$, defined in Equation (1).

$$\Phi(P) = \alpha \times A_P + \beta \times W_P \ , \qquad (1)$$

where α and β are user-specified parameters, A_P is the bounding-rectangle area of the placement, and W_P is the total wire length measured by the half-perimeter estimation. We follow the framework of HB*-trees [3] to handle hierarchical perturbation and packing. After an initial HB*-tree is given, our algorithm perturbs the HB*-tree to get a new solution and then checks the feasible conditions of each ASF-B*-tree. If any boundary constraint is violated, we transform the infeasible ASF-B*-tree into a feasible one. The algorithm repeats until predefined termination conditions are satisfied.

Now, we would like to describe how to transform an infeasible ASF-B*-tree into a feasible one for a 1D symmetry island. Given an ASF-B*-tree, let S_p denotes the set of nodes corresponding to the symmetry pairs on the bottom, right and top boundaries of a placement plane. For the nodes corresponding to symmetry pairs which are served as in-out modules but not in S_p, we record them in the set X_p. If $X_p \neq \varnothing$, each node $n_i^r \in X_p$ will be deleted from its current position and randomly inserted into the leftmost, or the bottom-left, or the bottom-right branch.

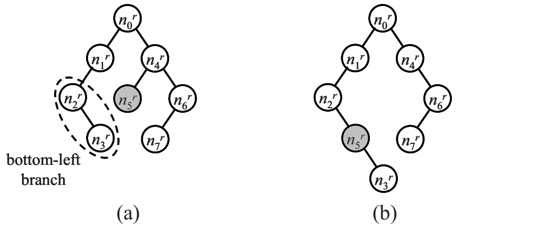


Figure 11. (a) Infeasible ASF-B*-tree if node n_5^r corresponds to a right-boundary module but n_5^r is not in the bottom-left branch. (b) Feasible ASF-B*-tree where n_5^r is inserted into the bottom-left branch.

For example, Figure 11(a) presents an infeasible ASF-B*-tree, where the node n_5^r corresponds to a right-boundary symmetry pair, but $n_5^r \notin S_p$. To get a feasible ASF-B*-tree, we delete n_5^r from the current position and insert it into the bottom-left branch. Figure 11(b) shows the feasible ASF-B*-tree after modification.

Similarly, let S_s denotes the set of nodes corresponding to the self-symmetry modules on the left boundary of a placement plane. For the nodes corresponding to the self-symmetry modules which are served as in-out modules but not in S_s, we record them in the set X_s. If $X_s \neq \varnothing$, we delete each node $n_i^r \in X_s$ from its current position and insert it into the rightmost branch. Further, to guarantee a feasible ASF-B*-tree for a 1D symmetry island during perturbation, we do not move a node to the left (right) child of the node in the bottom-left (bottom-right) branch. For a 2D symmetry island, we can follow the similar procedure to maintain a feasible ASF-B*-tree.

5. EXPERIMENTAL RESULTS

Our placement algorithm was implemented in C++ and run on a 2.8GHz Intel Pentium4 PC with 1GB RAM. We performed two sets of experiments: one is based on two circuits, which include biasynth_2p4g and lnamixbias_2p4g, used by [3, 10, 17, 20], and the other is to combine some symmetry groups in the two circuits and the MCNC benchmarks, which include apte, hp, ami33 and ami49.

In the first set of experiments, since no in-out module was specified in the original benchmarks, we randomly selected some devices from the symmetry groups as in-out modules. Table 3 lists the information of the benchmarks and the placement results of other works, which include sequence-pairs with dummy nodes [17], symmetry islands [3] and Plantage [10]. The fourth column describes the number of in-out modules randomly selected in each symmetry group. Although our work is not the best in the comparison results, we have considered the additional boundary constraint for each symmetry group, while other works did not. Figure 12 illustrates our placement result of lnamixbias_2p4g, in which the in-out modules are all placed at the boundaries of the symmetry islands.

If there exist more and more modules in a symmetry group, the probability that an in-out module is placed inside the symmetry island would be higher. Thus, in the second set of experiments, we randomly combined two symmetry groups into a large group in each benchmark, and randomly selected two devices as in-out modules in each combined symmetry group. These benchmarks are experimented with 10 trials by our placement algorithm with and without considering symmetry-island boundary constraint, respectively. In these experimental results, if any in-out module is not placed at the boundary of a symmetry island, we regard the result as an infeasible placement. Table 4 lists the information of the benchmarks and the infeasibilities of the placement results. In the third column of the table, (a+b) denotes a combined group consisting of two

Table 3. Comparisons of area and runtime for sequence-pairs with dummy nodes (SP w. Dummy) (on Pentinum4 3.2GHz), symmetry islands (Symmetry Is.) (on Pentium4 3.2GHz), Plantage (on Pentium4 3.2GHz), and our work (on Pentium4 2.8GHz).

Circuit Name	# of Mods.	# of Sym. Mods.	# of I/O Mods.	Mod. Area ($10^3 \mu m^2$)	SP w. Dummy [17]*		Symmetry Is. [3]*		Plantage [10]*		Our Work	
					Area (%)	Time (s)	Area (%)	Time (s)	Area (%)	Time (s)	Area (%)	Time (s)
biasynth_2p4g	65	8+12+5	0+2+1	4.70 = 100%	118.51	134	104.68	22	104.96	337	111.44	166
lnamixbias_2p4g	110	16+6+6+12+4	2+0+0+2+0	46.00 = 100%	113.50	227	105.72	43	107.68	387	112.14	272

* The experimental results were reported by the original works, which did not consider boundary constraint for the symmetry groups.

Table 4. Comparisons of infeasibilities for the placements with/without considering symmetry-island boundary constraint.

Circuit Name	# of Mods.	# of Sym. Mods.	# of I/O Mods.	Mod. Area ($10^3 \mu m^2$)	# of Trails	Placement with the Constraint			Placement without the Constraint		
						Avg. Area	Avg. Time	# of Infeas.	Avg. Area	Avg. Time	# of Infeas.
biasynth_2p4g	65	8+(12+5)	0+(2+0)	4.70 = 100%	10	113.92%	151 s	0	113.18%	95 s	4
lnamixbias_2p4g	110	16+6+(6+12)+4	0+0+(0+2)+0	46.00 = 100%	10	114.87%	224 s	0	114.13%	140 s	3
apte + hp + ami33	53	8+(8+6)	0+(2+0)	56546 = 100%	10	106.02%	122 s	0	105.51%	67 s	2
apte + hp + ami49	69	8+(8+4)	0+(2+0)	90840 = 100%	10	111.17%	186 s	0	110.57%	138 s	3

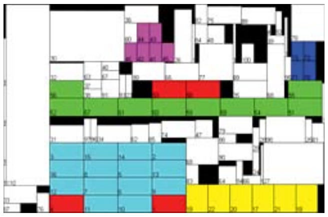
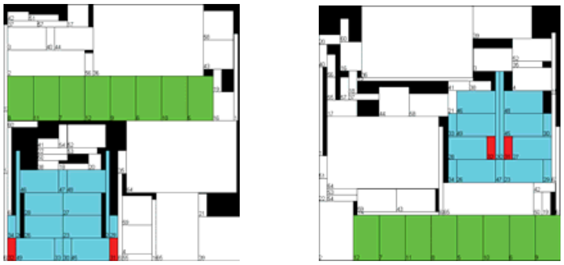


Figure 12. The placement result of lnamixbias_2p4g obtained by our approach considering symmetry-island boundary constraint. The symmetry groups are colored in different colors, and the in-out modules are colored in red. (Area usage: 112.14%)



(a) (b)

Figure 13. The placement results of biasynth_2p4g in the second set of experiments. (a) With the constraint. (Area usage: 112.48%) (b) Without the constraint. (Area usage: 111.82%)

symmetry groups whose numbers of modules are *a* and *b* respectively. The placement results of biasynth_2p4g with and without considering boundary constraint are shown in Figure 13(a) and 13(b), respectively. It demonstrates that our approach, placement considering symmetry-island boundary constraint, can guarantee the boundary properties for the in-out modules in symmetry islands.

6. CONCLUSIONS

We have introduced the issue of performance deterioration caused by in-out module locations and proposed the boundary constraint for symmetry islands to deal with the issue. Based on ASF-B*-tree, we have explored the feasible conditions with boundary constraint and developed an algorithm to meet this property in each perturbation. The experimental results have shown the effectiveness of our approach.

7. ACKNOWLEDGEMENTS

We would like to thank Prof. Po-Hung Lin of National Chung Cheng University for many very helpful suggestions on implementing ASF-B*-tree and HB*-tree. This work was partially supported by National Science Council of Taiwan under Grant No's NSC-98-2221-E-006-156-MY3.

8. REFERENCES

[1] J. M. Cohn, D. J. Garrod, R. A. Rutenbar, and L. R. Charley, *Analog Device-level Layout Automation*, Kluwer Academic Publishers, 1994.

[2] A. Hastings, *The Art of Analog Layout*, 2nd Ed., Prentice Hall, 2006.

[3] P.-H. Lin and S.-C. Lin, "Analog placement based on novel symmetry-island formulation," *Proc. DAC*, 2007, pp. 465-470.

[4] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680, May 1983.

[5] F. Balasa and K. Lampaert, "Module placement for analog layout using the sequence-pair representation," *Proc. DAC*, 1999, pp. 274-279.

[6] Y.-X. Pang, F. Balasa , K. Lampaert, and C.-K. Cheng, "Block placement with symmetry constraints based on the O-tree non-slicing representation," *Proc. DAC*, 2000, pp. 464-467.

[7] F. Balasa, "Modeling non-slicing floorplans with binary trees," *Proc. ICCAD*, 2000, pp. 13-16.

[8] J.-M. Lin, G.-M. Wu, Y.-W. Chang, and J.-H. Chuang, "Placement with symmetry constraints for analog layout design using TCG-S," *Proc. ASPDAC*, 2005, pp. 1135-1138.

[9] L. Zhang, C.-J. R. Shi, and Y. Jiang, "Symmetry-aware placement with transitive closure graphs for analog layout design," *Proc. ASPDAC*, 2008, pp. 180-185.

[10] M. Strasser, M. Eick, H. Graeb, U. Schlichtmann, and F. M. Johannes, "Deterministic analog circuit placement using hierarchically bounded enumeration and enhanced shape functions," *Proc. ICCAD*, 2008, pp. 306-313.

[11] J. Liu, S. Dong, Y. Ma, D. Long, and X. Hong, "Thermal-driven symmetry constraint for analog layout with CBL representation," *Proc. ASPDAC*, 2007, pp. 191-196.

[12] P.-H. Lin, H. Zhang, M. D. F. Wong, and Y.-W. Chang, "Thermal-driven analog placement considering device matching," *Proc. DAC*, 2009, pp. 593-598.

[13] J. Lai, M.-S. Lin, T.-C. Wang, and L.-C. Wang, "Module placement with boundary constraints using the sequence-pair representation," *Proc. ASPDAC*, 2001, pp. 515-520.

[14] S. Tayu, "A simulated annealing approach with sequence-pair encoding using a penalty function for the placement problem with boundary constraints," *Proc. ASPDAC*, 2003, pp. 319-324.

[15] Y. Ma, S. Dong, X. Hong, Y. Cai, C.-K. Cheng, and J. Gu, "VLSI floorplanning with boundary constraints based on corner block list," *Proc. ASPDAC*, 2001, pp. 509-514.

[16] J.-M. Lin, H.-E. Yi, and Y.-W. Chang, "Module placement with boundary constraints using B*-trees," *IEE Proceedings – Circuits, Devices and Systems*, vol. 149, no. 4, pp. 251-256, Aug. 2002.

[17] Y.-C. Tam, E. F. Y. Young, and C. Chu, "Analog placement with symmetry and other placement constraints," *Proc. ICCAD*, 2006, pp. 349-354.

[18] R. Naiknaware and T. S. Fiez, "Automated hierarchical CMOS analog circuit stack generation with intramodule connectivity and matching considerations," IEEE JSSC, vol. 34, no. 3, pp. 304-317, Mar. 1999.

[19] P.-H. Lin, H.-C. Yu, T.-H. Tsai, and S.-C. Lin, "A matching-based placement and routing system for analog design," *Proc. VLSI-DAT*, 2007, pp. 16-19.

[20] F. Balasa, S. C. Maruvada, and K. Krishnamoorthy, "On the exploration of the solution space in analog placement with symmetry constraints," *IEEE TCAD*, vol. 23, no. 2, pp. 177-191, Feb. 2004.

[21] S. H. Lewis and P. R. Gray, "A pipelined 5-Msample/s 9-bit analog-to-digital converter," *IEEE JSSC*, vol. 22, no. 6, pp. 954-961, Dec. 1987.