electronics

MDPI

Article

Efficient Circuit Structure Analysis for Automatic Behavioral Model Generation in Mixed-Signal System Simulation

Ling-Yen Song [1,*], Yu-Kang Lou [2], Ching-Ho Lin [2], Chien-Nan Liu [1], Juinn-Dar Huang [1], Jing-Yang Jou [2], Meng-Jung Lee [3] and Yu-Lan Lo [3]

[1] Institute of Electronics, National Yang Ming Chiao Tung University, Hsinchu 300093, Taiwan; jimmyliu@nctu.edu.tw (C.-N.L.); jdhuang@mail.nctu.edu.tw (J.-D.H.)
[2] Department of Electrical Engineering, National Central University, Taoyuan 320317, Taiwan; jameslouhouse@gmail.com (Y.-K.L.); saal601601@gmail.com (C.-H.L.); jyjou@ee.ncu.edu.tw (J.-Y.J.)
[3] Realtek Semiconductor Corporation, Hsinchu 300093, Taiwan; monica_lee@realtek.com (M.-J.L.); jeny@realtek.com (Y.-L.L.)
* Correspondence: audrey1535.ee07g@nctu.edu.tw

Abstract: For mixed-signal systems, identifying the analog and digital circuit blocks in the transistor-level netlist has many benefits for system analysis and verification. However, existing approaches still have difficulty handling large mixed-signal designs with millions of transistors, especially when multiple analog structure patterns are included. In this paper, we propose an efficient structure recognition methodology to support analyzing highly complex designs with various circuit structures and different devices. In order to tackle the complexity of real cases, a hierarchical partition-based analysis methodology and an encoding-based fast screening technique are proposed in this work. To correctly ascertain the boundary of analog and digital structures, we propose an enhanced direct current connection (DCC) partition method and combine it with the analog structure analysis flow. The non-transistor devices, such as resistors and capacitors, are also included in our recognition flow to improve the recognition capability and accuracy. As demonstrated with two industrial cases, the behavioral models generated from the structure recognition results do help to improve the efficiency of the AMS system verification.

Keywords: behavioral model; mixed-signal simulation; model generator

check for updates

Citation: Song, L.-Y.; Lou, Y.-K.; Lin, C.-H.; Liu, C.-N.; Huang, J.-D.; Jou, J.-Y.; Lee, M.-J.; Lo, Y.-L. Efficient Circuit Structure Analysis for Automatic Behavioral Model Generation in Mixed-Signal System Simulation. *Electronics* **2021**, *10*, 1088. https://doi.org/10.3390/electronics10091088

Academic Editor: Paolo Colantonio

Received: 11 April 2021
Accepted: 1 May 2021
Published: 4 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

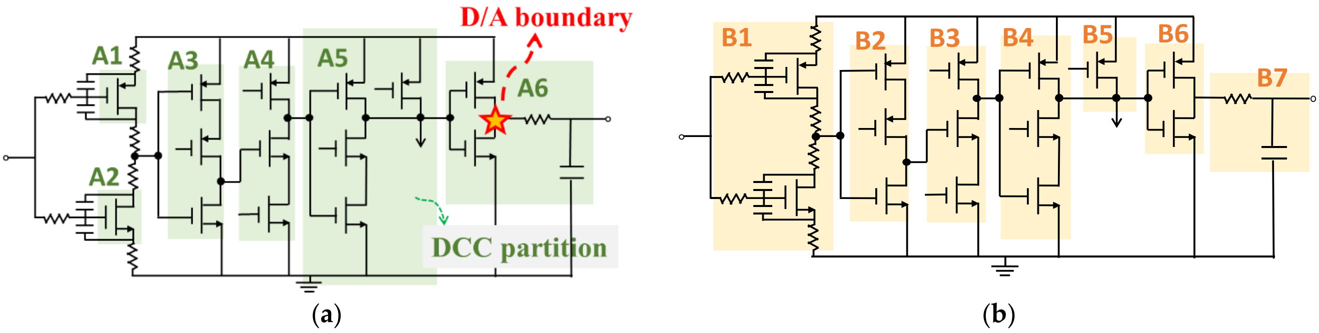
Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In System-on-Chip (SoC) designs, digital circuits and analog circuits are often integrated in the same chip. In traditional design flows, analog and digital parts can be integrated at the layout level only, and it is impractical to perform whole chip simulation after integration. In recent years, a couple of synthesizable approaches have been proposed that try to generate the corresponding hardware for analog circuits so that they can be co-simulated with digital circuits on an FPGA or emulator [1–5]. However, the cost to fix the bugs at such a late design stage is very high, and it would be preferable to have alternate solution that allows for checking the system integration at early stages. Another possible solution is building the behavioral models for both digital and analog circuits and using these behavioral models to perform system simulation [6–11]. This approach greatly reduces the simulation time and enables designers to verify the system integration at early design stages. If the required behavioral models can be automatically generated from the given designs, this could be an attractive approach for designers. Because the behavioral models of analog circuits and digital circuits are built in different ways [12–28], it is necessary to correctly identify different circuit blocks in the transistor-level netlist of the whole system. This can also help to understand the behavior of the flattened transistor-level netlists extracted from layouts in reverse engineering or commercial tools.

At the integration stage, the designs are often given as transistor-level netlist, in which digital and analog structures are mixed together. Therefore, the first difficulty is the high complexity of the circuit netlist. For digital designs, many existing techniques [29–31] are able to extract the sub-circuits efficiently. Although a digital-style approach can deal with high complexity, it can be applied on standard digital circuits only. If some analog-style circuits are mixed together, such as the pass-transistor logic (PTL), the extraction results may not perfectly match the real gate structures. SubGemini [32] is the first work for general sub-circuit identification based on graph isomorphism check, which is usually employed for the transformation from transistor-level to gate/block-level netlist in a reverse engineering procedure [33,34]. Although this approach is able to deal with analog structures, the complexity of an exhaustive search is often too high to be applied on modern million-gate designs. There are some recent works [35–45] that improve the efficiency of sub-circuit identification. However, they can be applied on small designs with tens to hundreds of transistors only. Moreover, most of the previous works cannot support multiple pattern identification. If only one target pattern can be compared at a time, it is not efficient for modern designs with hundreds of cells in the library.

The second difficulty in dealing with modern designs comes from the mixture of digital cells and analog circuits. Most of the identification techniques for digital cells often fail at the boundary of digital and analog parts as shown in Figure 1a. Take the DC connection (DCC) based method [31] as a simple example. Figure 1b is the desired structure composed of 6 digital cells and 1 analog block, along with 1 MOS switch that forces the TSPC (true single-phase clock) flip-flop to go to a reset state and a simple RC filter. In the DCC partition approach, due to the drain connection, both A5 and A6 in Figure 1a are failed outcomes. Of course, these supercells A5 and A6 cannot match any existing cell in the pre-defined library. Furthermore, circuit designers may insert some extra devices other than transistors into the designs, such as the pull-up resistors, the output capacitance and the parasitics extracted from the layouts shown in B1 of Figure 1. These extra devices do not change the main functionality of the sub-circuits, but they will change the circuit pattern and make all existing methods fail to identify the desired blocks.



(**a**) (**b**)

Figure 1. (**a**) Unacceptable (**b**) Expected identification results for TSPC (true single-phase clock) logic with reset control and RC filter.

In this paper, we propose an efficient circuit structure recognition methodology for large mixed-signal designs, as shown in Figure 2, to support automatic behavioral model generation from given transistor-level netlists. In order to deal with real industry cases, a hierarchical recognition methodology is proposed in this work. The first stage compares each sub-circuit in the design hierarchy with the library cells to pick up the digital circuits composed of standard cells and existing analog components. For other complex sub-circuits that are not composed of a single cell, we applied a DCC partition in each sub-circuit to find more matched digital blocks in the partition results. For the blocks that could not be partitioned correctly in DCC approach, an enhanced partition method based on current path analysis is proposed to improve the accuracy. After the digital blocks are extracted, the remaining netlist is dealt with using the analog structure analysis technique. Finally, the behavioral models of every identified circuit blocks in the analog part and digital part can be generated to support system-level behavior simulation. The non-transistor devices, such as resistors, capacitors, and inductors, are also automatically replaced by proper models in the pre-processing step to avoid misleading models and further improve the accuracy of the generated system-level behavioral model. As shown in two real industry cases, the proposed methodology is able to deal with large mixed-signal designs with tens to hundreds or thousands of different devices. Compared to previous works, the key advantages of the proposed structure recognition methodology are briefly summarized as follows.

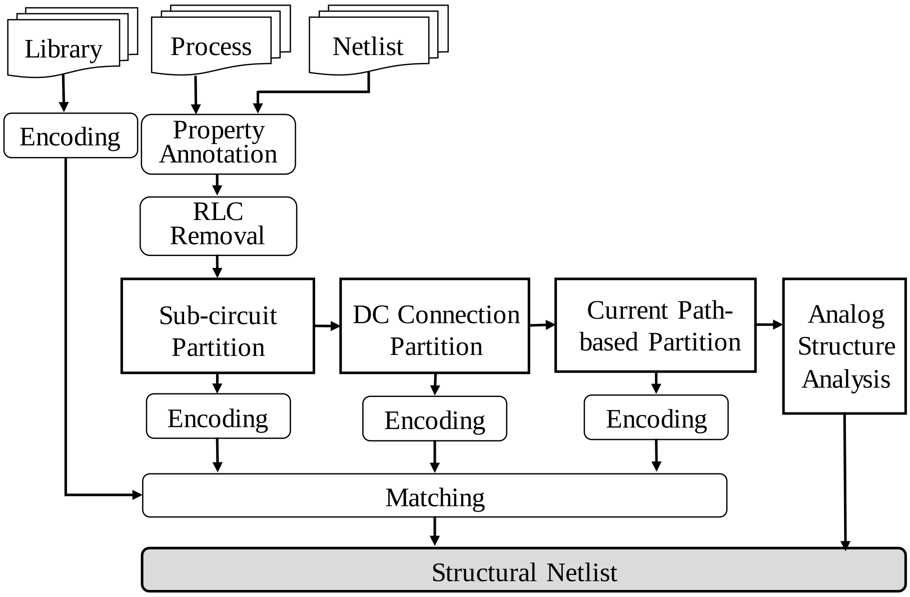


Figure 2. Overall hierarchical structure recognition flow for system-level behavior simulation.

1. In order to deal with the complexity of real industrial mixed-signal designs, a hierarchical structure recognition methodology for both digital and analog circuits is proposed. Because most simple cases are solved in early stages by quick methods, only a few cases are required to be compared carefully in late stages for accuracy. Combined with the proposed circuit encoding scheme for fast screening, this methodology not only significantly reduces the comparison time but also improves the search efficiency by supporting simultaneous comparison with multiple blocks.
2. In order to improve the accuracy for the cases where analog and digital structures are mixed together, this work considers the original design hierarchy and proposes an enhanced DCC partition method based on current path analysis. This is an efficient and accurate approach to recognize both standard CMOS cells and non-standard structures. Non-transistor devices, such as resistors and capacitors, are also properly replaced by an equivalent DC model to avoid misleading results. This feature can help identify more equivalent cells with similar structures.
3. In order to support hundreds of standard cells and various circuit structures in real industrial designs, the proposed methodology leaves enough flexibility for users to add the blocks they like to increase the recognition capability, no matter whether the block is composed of analog or digital circuits. A bottom-up recognition process with redundancy removal is also proposed to support compound custom structures. This feature further increases the feasibility of the proposed methodology on real designs.

The rest of this paper is organized as follows. The evolutions of analog modeling and structure recognition methods are introduced in Section 2. Section 3 explains the proposed circuit encoding scheme and other pre-processing works. The proposed hierarchical structure recognition flow is presented in Section 4. Section 5 briefly explains the behavioral model generation flow based on the structure recognition results. The experimental results on two industrial cases are provided in Section 6 to demonstrate the accuracy and efficiency of the proposed approach. Finally, some conclusions are drawn in Section 7.

2. Background

2.1. Behavioral Modeling for Analog Circuit

Unlike digital circuits that purely operate in simple Boolean functions, analog circuits often exhibit continuous-time, non-linear behavior. Therefore, the behavioral models built by some simple circuit equations may not be able to model the non-linear properties of analog circuits accurately. Building the behavioral models from transistor-level netlist may be a better way to improve the modeling accuracy and obtain meaningful results for system verification. This straight-forward approach treats the given circuit as a black box and uses some mathematical approximation, such as linear or non-linear regression, neural networks, Volterra series, Kriging technique, response surface design, etc., to build the relation equations between its inputs and outputs. These kinds of models, which are known as metamodels and surrogate models [12–16], require many simulation results to perform the approximating process. Due to extensive simulation time, it is impractical for large designs.

In the current design flow, most designers are used to manually abstracting the circuit and build behavioral models based on their design knowledge. Therefore, a couple of customized modeling approaches have been proposed to automatically extract the corresponding behavioral models for specific mixed-signal circuits, such as PLL, ADC, and DAC, etc. [21–26]. With prior knowledge of the circuit behavior, these approaches develop efficient methods to construct the specific behavioral models with good accuracy. However, the same method cannot be directly applied to different circuits, and the abstraction level cannot be changed easily for different verification purposes. As a result, we were motivated to develop an automatic abstraction methodology that could not only be applied to huge system designs or architectures but also support various fidelities depending on different complex applications.

Another approach builds the behavioral models based on the circuit structures, as illustrated in Figure 3. Instead of modeling the relationship between circuit inputs and outputs directly, the whole circuit is divided into several small building blocks, whose behavioral models are built separately. Since each building block is a common simple block, it is much easier to be modeled by the black-box approach. Therefore, the model construction efforts can be greatly reduced without losing the generality for different circuits. In recent years, a structural macromodeling technique has been used to model the building block behaviors for analog circuits, because this hierarchical approach is able to deal with different fidelities and various sizes of circuits [19,20,27,28]. However, accurate structure recognition techniques are the key to enable this kind of modeling approach, which are introduced in next section.

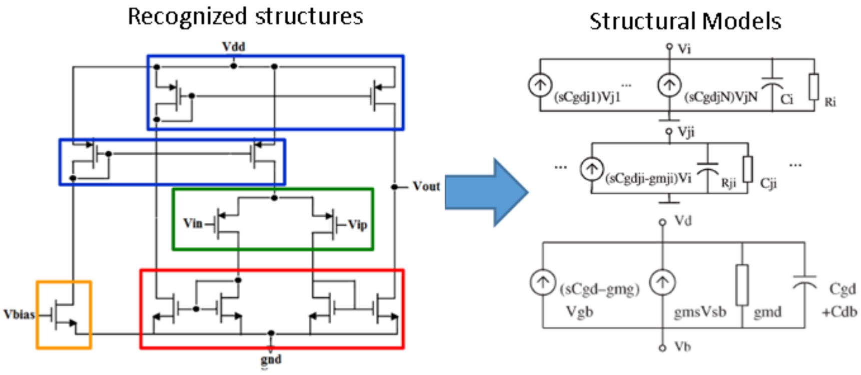


Figure 3. Illustration of macromodeling on building blocks.

2.2. Structure Recognition

Structure recognition is widely used in many computer-aided design fields, such as circuit simulation, synthesis, testing, verification, physical design, and reverse engineering. For circuit simulation, large designs are often partitioned into small structures, to reduce the dimension of equations for simulations [46]. In recent years, a couple of approaches have been proposed for digital circuits to recognize the gate components within a synthesized transistor-level netlist [30,31]. Among these methods, the partition-based approach is able to abstract the design with several hundreds of thousands of transistors in few minutes. For example, the DCC approach [31] begins from the power supply and traverses through the drain and source ports of transistors until reaching the ground. If the connection meets a gate port, the traversal stops at this point, which is the key point to cut down connections between digital gates. All the reached transistors in this traversal will be grouped as a block. Although this kind of approach is successful for purely digital designs, it cannot handle non-typical structures and analog components in the mixed-signal systems, as mentioned in Section 1.

In order to deal with more general structures, a structure recognition problem is usually formulated as a graph isomorphism problem in these approaches [35–41]. For example, SubGemini [32], the first method representing a circuit as a graph, tries to identify the sub-graph by iteratively exploring each selected candidate and relabeling the vertices of the candidate. Thus, two identical structures can be recognized with these given labels. By translating circuits into graph representations, the structure recognition can be solved by the identical sub-graph searching method. However, graph-based structure recognition suffers from the complexity explosion issue. It is difficult to apply these techniques to modern complex systems.

Using similar ideas for digital circuits, another approach treats analog circuits as a combination of some common building blocks [47]. For this purpose, a bottom-up block merging method has been proposed to explore building blocks' structures rapidly [44]. In [44], a unique representation of the circuit, structural signal flow graph (SSFG), is used to represent analog circuits. Rather than performing isomorphism checking directly, the method merges the two sub-circuits that have contributed to layout constraints by a graph union operation. Because building blocks usually accompany analog layout constraints, the overall recognition result is built in a hierarchical rule tree, and each node in the tree is a building block. Figure 4 demonstrates this method with a simple circuit. However, analyzing analog circuits is more challenging than digital circuits because various structures with little modification may be used. Therefore, machine learning-based approaches have been involved to identify the building blocks with some variations [45]. In addition to these large training efforts, this learning-based approach may generate unreasonable building blocks due to mis-identification, which is not acceptable by designers. Most importantly, these approach are proposed for analog circuits only. There is no evidence that they can handle industry-scale complex systems with a mixture of analog and digital blocks.

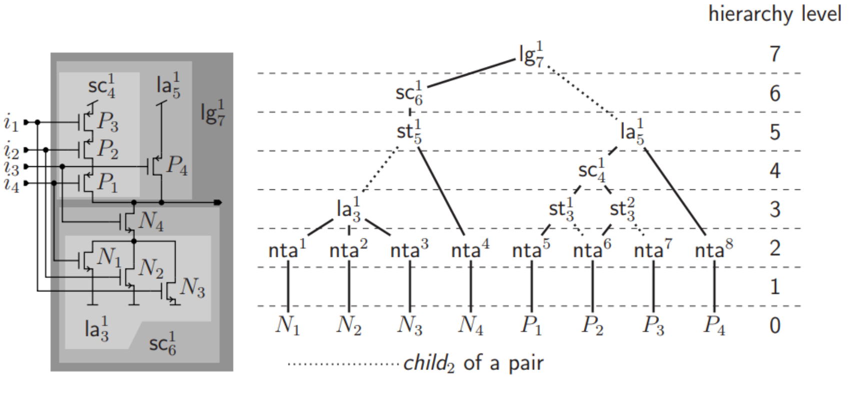


Figure 4. Hierarchical recognized structures for the circuit [44].

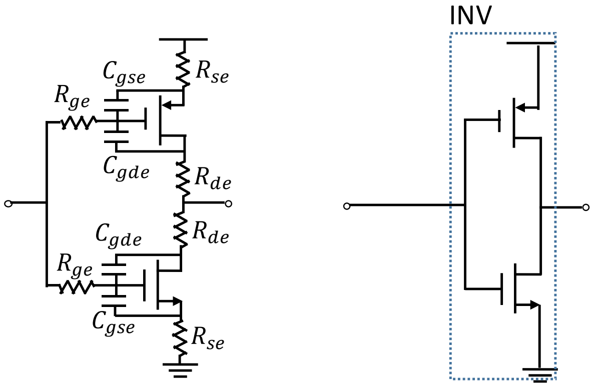
3. Pre-Processing and Circuit Encoding

In industrial design flow, transistor-level netlists are often extracted by commercial tools. With these tools, the circuit topology can be retained well but some design properties are lost. In a post-layout netlist, it becomes more complicated due to extra nodes and devices for the layout induced parasitics. Therefore, some pre-processing work is necessary to automatically handle this kind of netlist efficiently.

3.1. Property Annotation and RLC Removal

In real applications, different designs may have multiple supply voltages and various names for power and ground nodes according to demands. Moreover, in different technology files, transistor models may be defined by different names. Because the power/ground nodes and device types are essential information for structure analysis, the recognizer will first automatically identify power/ground nodes and various kinds of devices through given pin information and model definition. This pre-processing is able to simplify the following recognition process and increases the portability of the proposed flow.

Most sub-circuit identification works are developed to handle the circuits composed of transistors only [31,32]. However, passive components (PCs), e.g., capacitors and resistors, are usually used in real applications that may cause mismatching when comparing to the patterns in the library, as shown in Figure 5a. In this work, we adopt a simple approach to deal with these unnecessary devices as they are in DC conditions. Capacitors are treated as open circuits and removed directly; resistors and inductors are treated as short circuits and replaced by wires. After this, a pure transistor structure is delivered for the following structure analysis procedure, as illustrated in Figure 5b, which can be easily recognized as an inverter structure. PC removal benefits the successful rate of structure identification. In Figure 1, A1 and A2 are able to be recognized as the inverter structure B1, which helps to increase the accuracy of the behavioral model. While concerning the model accuracy loss caused by RLC removal, the contributions of PCs can be recovered by the model calibration process. Certainly, RLC circuits, such as RC filters B7 in Figure 1, are kept and recognized as separate models.



(**a**) pre-removal (**b**) post-removal

Figure 5. The inverter with resistors and capacitors.

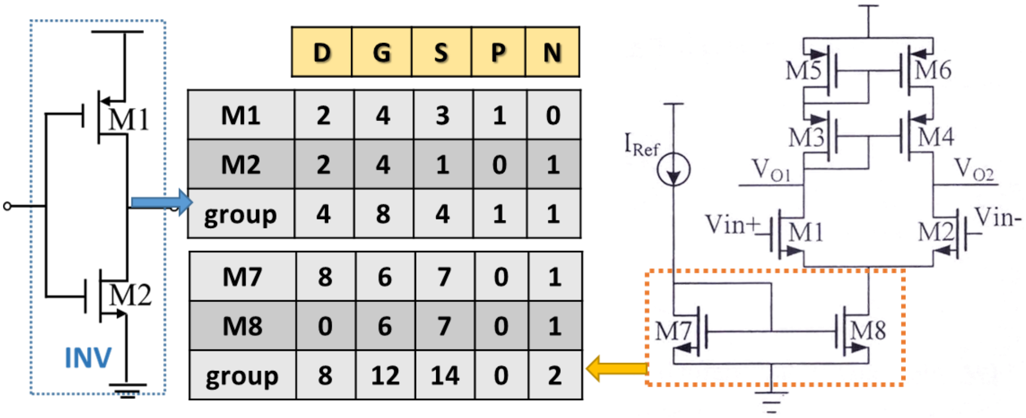
3.2. Circuit Encoding Scheme

Graph isomorphism is a well-known problem with many efficient algorithms. However, while being used in the sub-circuit identification problem, comparing only one target pattern at a time is not efficient for modern designs with hundreds of different cells. In this work, we propose a special circuit encoding scheme to represent each circuit structure as a series of numbers. If the numbers of two groups are different, it is not possible for them to have the same structure. In this way, the graph isomorphism problem can be turned into a simple number comparison problem, which greatly accelerates the matched pattern searching even for a huge library. In this section, the encoding rules and the comparing method are briefly introduced.

In the device encoding procedure, each transistor will be represented as a 5-number code according to the connection and device type. The first three numbers specify the connecting terminal types for the three terminals D(drain), G(gate), and S(source) of a transistor. For example, if a terminal is connected to a drain terminal, it will realize a score 2 at this field. If this terminal goes to Vdd, it will realize a score 3 at this field. Refer to Table 1 for score definitions of all connection types. If this terminal is connected to multiple terminals, the final number is determined by summing all scores together. We review the simple circuit in Figure 6 as an example. M7 will a realize a score 7 (6 + 1) at the S field because it connects to a source terminal (+6) of M8 and the ground node (+1). The last two numbers specify the type of this transistor. PMOS transistors will realize a score 1 at the P field and a score 0 at the N field, and vice versa. Finally, a group code is determined to represent a partitioned circuit. Each field in the group code is the sum of the same field in all devices. As shown in Figure 6, the last rows in code tables are the group codes of the two partitioned circuits, respectively.

Table 1. Definitions of transistor encoding.

Encoding Definitions				
To Ground	To Power	To Drain	To Gate	To Source
+1	+3	+2	+4	+6



	D	G	S	P	N
M1	2	4	3	1	0
M2	2	4	1	0	1
group	4	8	4	1	1
M7	8	6	7	0	1
M8	0	6	7	0	1
group	8	12	14	0	2

Figure 6. Illustration of proposed encoding scheme.

3.3. *Structure Library for Matching*

In modern designs, almost all digital circuits are implemented with pre-defined standard cells. However, for analog circuits, it is difficult to enumerate all possible basic structures, as mentioned in [47]. Circuit designers often invent some special structures to meet different requirements. In this work, we propose an easy-to-extend structure library that allows users to add new structure blocks. The digital standard cells and the common analog structures mentioned in [47] are added into the library as the default structures to support typical designs. For non-typical structures, users can add them into the library to allow the tool to recognize these structures and generate the required behavioral models or design constraints for these blocks automatically. This flexible scheme helps to deal with real industry designs in our experiments. Each block in the structure library has a number following the encoding in Section 2.2. This number will be compared with the code of identified blocks to exclude the structures that are not possible to be the same with the identified block.

4. Hierarchical Structure Recognition

For million-transistor designs, searching whole circuits to identify some specific patterns is difficult. Therefore, we propose a hierarchical structure recognition methodology with special circuit encoding scheme to improve efficiency. This methodology can be divided into four partition stages: sub-circuit partition, DCC partition, current-based partition, and analog block partition. Most simple cases are solved in the early stages by quick partition methods, which maintains good efficiency even in large designs. Where digital and analog structures are mixed together, this hierarchical methodology also helps to improve partition accuracy while maintaining efficiency. In the following sections, the details of each step are introduced.

4.1. Sub-Circuit Identification

In general, large designs are often partitioned into proper hierarchies based on their functionalities to tackle design complexity. Moreover, most of the digital blocks in real cases are standard cells, which appear as individual sub-circuits in the SPICE netlist. Therefore, in the first stage, we directly calculated the group code for each sub-circuit in the netlist based on the original partition and compared the group code to each structure in the library. This approach can quickly identify most of the blocks even in large designs. Note that pre-defined analog structures are also encoded and included in the structure library. Therefore, analog sub-circuits can also be identified at this stage. Furthermore, the proposed number comparison scheme allows for the concurrent identification of multiple cells, which is quite efficient even for commercial libraries with hundreds of cells. However, there are some rare cases that two similar circuit structure have the same group code. In this work, a detail isomorphism check is also proposed to distinguish such cases. Because the number of such cases is small, the time for isomorphism checking is greatly reduced to keep the structure recognition process efficient for large designs.

4.2. DCC Matching and Current Path Analysis

For other complex sub-circuits that are not composed of a single block, further analysis is required to identify the composing blocks. In the second stage, we partition these sub-circuits based on the DCC concept [31]. As mentioned in Section 1, DCC may fail to identify digital blocks at the boundary of digital and analog parts. Moreover, it also fails at the node driven by multiple gates. Therefore, for these unrecognizable groups, such as A5 in Figure 1, a detailed analysis based on the current paths [48] is applied and shown in Figure 7.

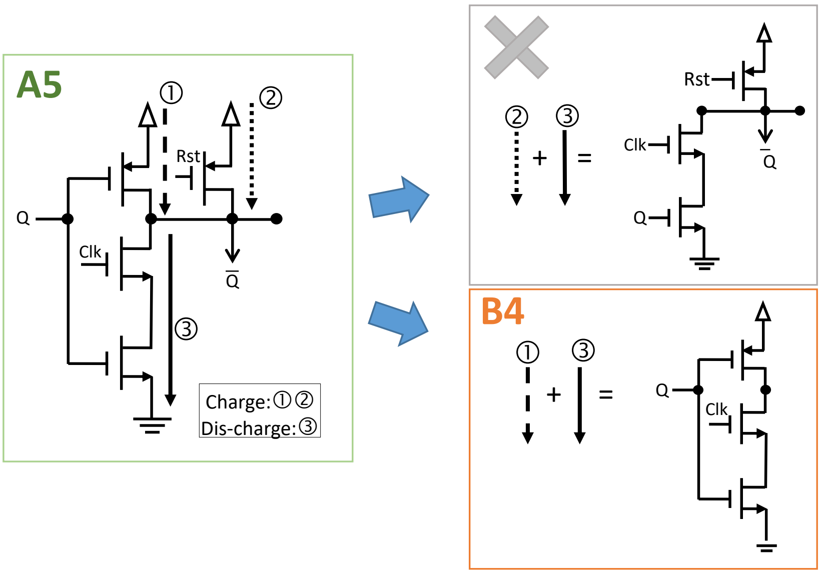
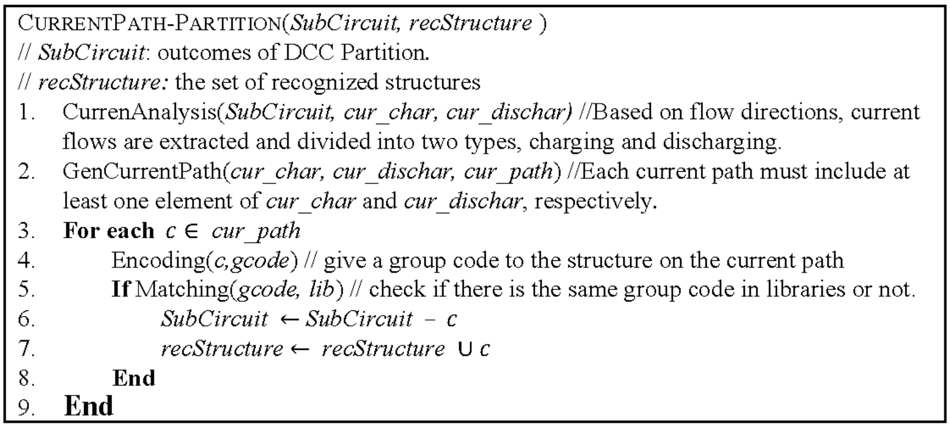


Figure 7. Illustration of current path method.

Current path partition begins by determining the current direction of each transistor based on their types, which are marked as N(φ1①,②), and P(③) in Figure 7. Because most digital circuits are implemented with CMOS technologies, a legal combination should include both charging and discharging paths. For each legal current path combination, it will be checked to see if any one matches a pre-defined structure in the library. After the current path partition, B4 and B5 can be extracted from A5 successfully. The detailed flow is shown in Figure 8.



```
CURRENTPATH-PARTITION(SubCircuit, recStructure )
// SubCircuit: outcomes of DCC Partition.
// recStructure: the set of recognized structures
CurrenAnalysis(SubCircuit, cur_char, cur_dischar) //Based on flow directions, current flows are extracted and divided into two types, charging and discharging.
GenCurrentPath(cur_char, cur_dischar, cur_path) //Each current path must include at least one element of cur_char and cur_dischar, respectively.
For each $c \in$ cur_path
    Encoding(c,gcode) // give a group code to the structure on the current path
    If Matching(gcode, lib) // check if there is the same group code in libraries or not.
        SubCircuit ← SubCircuit − c
        recStructure ← recStructure ∪ c
    End
End
```

Figure 8. Pseudocode of current path-based partitioning.

4.3. Analog Structure Analysis

After the digital blocks are extracted, the remaining analog design is much smaller than the original design, which allows for detail structure analysis. The analog parts were analyzed by the structure-based approach [44,45] with some modifications. Based on the netlist connection between each device, the common building blocks in analog circuits can be recognized by a bottom-up clustering from lower levels to higher levels. In addition to the basic structures in [47], we expanded the library to 80 blocks according to designers' feedback. Figure 9 shows a part of the building blocks at different level libraries. The block library can be enriched by users later to fit the common usage of their applications. In order to improve the simulation speed, we attempted to cluster these composing blocks as much as possible and represented each cluster as one circuit block in the simulation model.

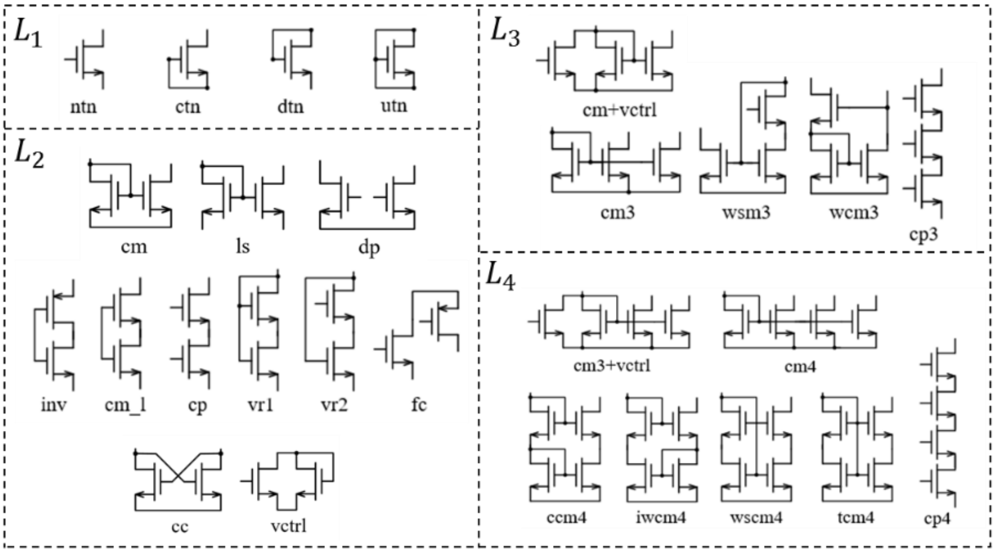


Figure 9. Illustration of building block library.

As illustrated in Figure 10, rather than choosing the two recognized blocks CCM and DP as models, the model CDA, which includes more devices, can improve the performance of circuit simulation better than the smaller models. Therefore, instead of keeping all subsets, i.e., recognized blocks, in the structure library, we modified the bottom-up searching with module set reduction, given a module set $M = \{S_1, S_2\}$, in which both subsets have two elements, i.e., devices, $S_1 = \{e_1, e_2\}$ and $S_2 = \{e_2, e_3\}$. If a new subset $S' = \{e_1, e_2, e_3, e_4\}$ is added to the module set M, we updated M by excluding redundant subsets: $M \leftarrow M \cup S' - \{S_1, S_2\}$ when $S_1, S_2 \in S'$. This approach was able to shrink the database of module clustering and help to reduce the searching time effectively.

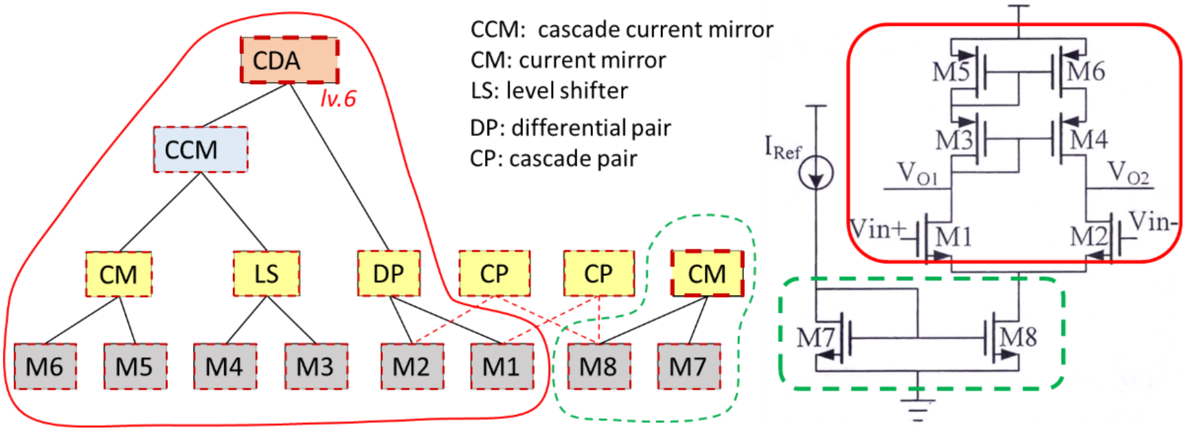


Figure 10. Structure recognition result of CDA with a CM.

If the same device was involved in more than one candidate at the same level, bond strength would be adopted as the decision factor. If more connections existed between two blocks, they were more likely to be in the same cluster. For example, as shown in Figure 11, M1 is the common device of CM and two CPs. However, the bond strength of CM, *bs*(CM), was larger than *bs*(CS). Thus, only the CM was added to the model list. The effectiveness of our priority selection strategy is demonstrated in this general analog circuit topology.

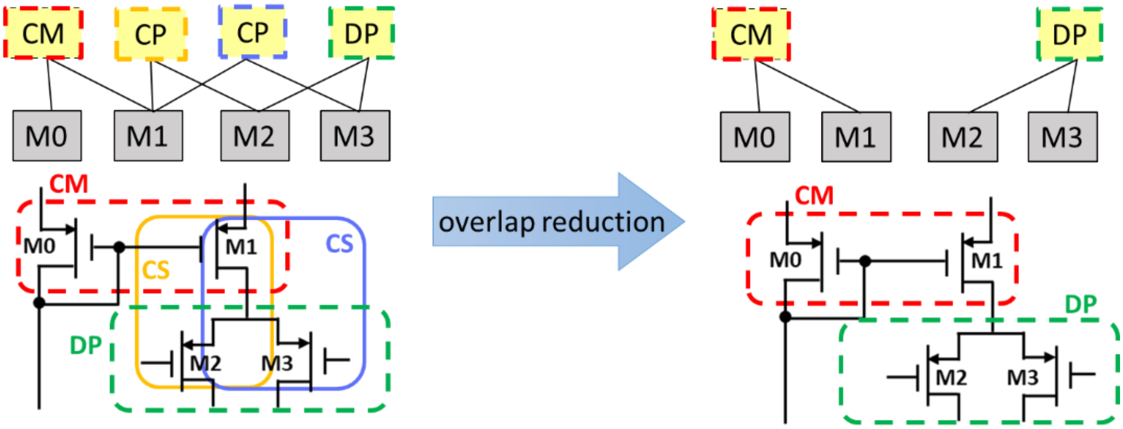


Figure 11. Example of bond strength selection for overlapping problems.

The overall flow of analog structure analysis is shown in Figure 12. The level of the circuit, *lv*(c), is the number of devices included in the largest block of the analog circuit. In the initialization, each device was regarded as a block element at level1, *lv*(b) = 1 and *lv*(c) = 1. In each iteration, the two connected blocks were grouped together and given a group code by the encoding scheme mentioned in Section 2.2. If the group code was matched to the analog library, a new block, which includes more devices, was created at the higher level, and the level of the circuit, *lv*(c), was updated. Moreover, these blocks, which were completely involved in other blocks with larger *lv*(b), were removed from the list for the next iteration grouping after the matching process. The redundant block reduction can reduce the loading of clustering operations in each iteration and prevent complexity explosion. When there is no new block created in an iteration, the bottom-up clustering process was terminated.

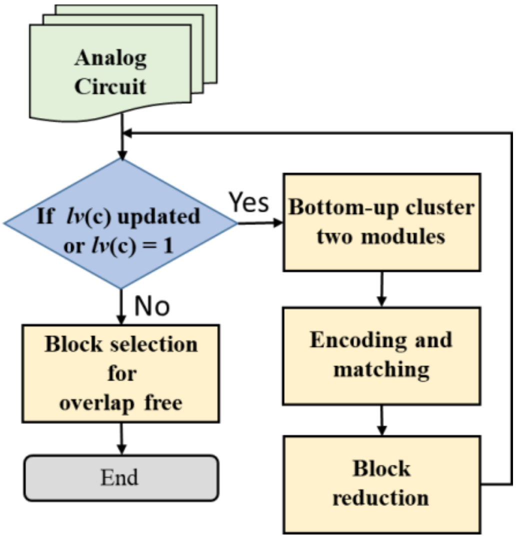

Analog Circuit
If lv(c) updated or lv(c) = 1
Yes
No
Bottom-up cluster two modules
Encoding and matching
Block reduction
Block selection for overlap free
End

Figure 12. Overall flow of analog structure analysis.

5. Application on Behavioral Model Generation

Digital designs are often started from HDL models and then synthesized into gate-level netlist composed of standard cells. Therefore, they already have comprehensive behavioral models at gate-level and/or RTL in the library. Only analog circuits require special handling to generate their corresponding behavioral models. However, building accurate yet efficient behavioral models for analog circuits is not an easy job due to their nonlinear properties. For specific circuits, such as PLL, ADC, DAC, etc., many approaches [25–28] are available to build these behavioral models automatically. In order to deal with different kinds of circuits, some automatic behavioral model generators are developed based on statistical approaches [26] or structure-based approaches [27,28]. Although those previous works are able to extract the required behavioral models from the given netlist, several limitations still exist that prevent their being used in large AMS designs.

In this work, we adopted a structure-based flow [49] to generate the behavioral models from the given netlist. The overall flow is shown in Figure 13. After the proposed structure recognition was applied, we built the relevant behavioral model of each recognized building block in the circuit. Based on the functionality of each block, the model template for each block was built in advance. Figure 14 shows an example of the current mirror block, which treats the output as the behavior of a current source. The description can be implemented in real number mod-el(RNM) of Verilog. During the model construction step, several circuit parameters were extracted by simulation to calibrate the corresponding behavioral model of each sub-block. Because each block has different device size and structure, the model of each block was calibrated separately in the original design to obtain accurate parameters. As illustrated in Figure 15, the five models of the recognized blocks were calibrated independently to have pin accuracy. The final behavioral model for the whole circuit can be obtained by connecting the sub-models of every block, including digital blocks. Then, the behavioral model can be used to speed up the simulation for the verification of a mixed-signal system.

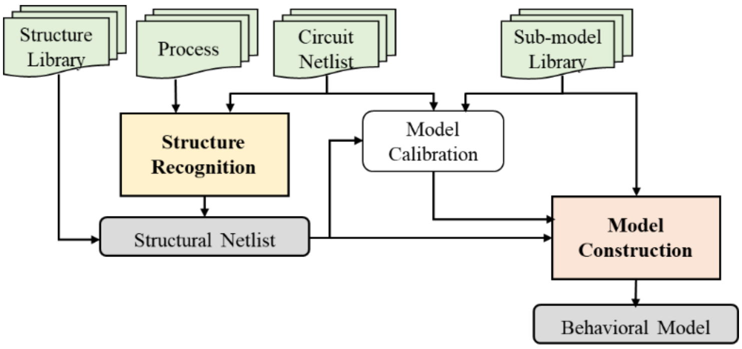
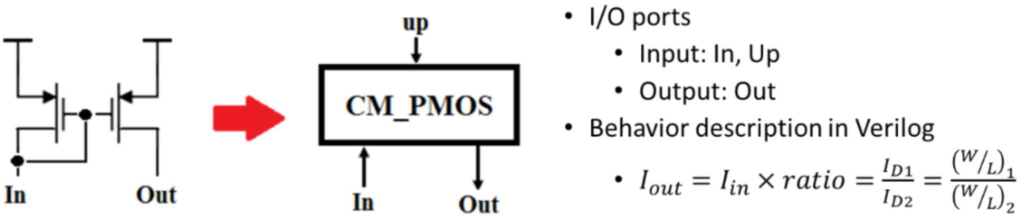
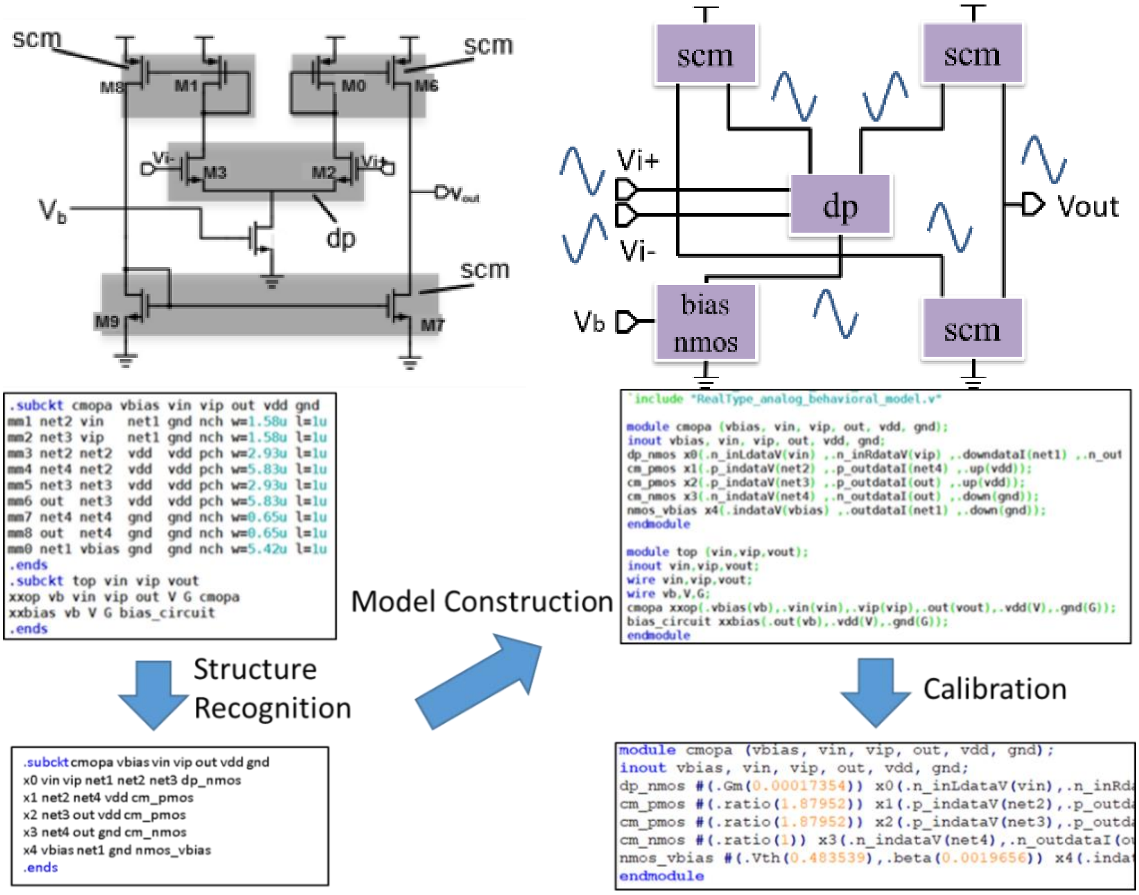


Figure 13. Behavioral model generation flow.



- I/O ports
  - Input: In, Up
  - Output: Out
- Behavior description in Verilog
  - $I_{out} = I_{in} \times ratio = \frac{I_{D1}}{I_{D2}} = \frac{(W/L)_1}{(W/L)_2}$

Figure 14. Real number model of current mirror block in Verilog.



```
.subckt cmopa vbias vin vip out vdd gnd
mm1 net2 vin   net1 gnd nch w=1.58u l=1u
mm2 net3 vip   net1 gnd nch w=1.58u l=1u
mm3 net2 net2  vdd  vdd pch w=2.93u l=1u
mm4 net4 net2  vdd  vdd pch w=5.83u l=1u
mm5 net3 net3  vdd  vdd pch w=2.93u l=1u
mm6 out  net3  vdd  vdd pch w=5.83u l=1u
mm7 net4 net4  gnd  gnd nch w=0.65u l=1u
mm8 out  net4  gnd  gnd nch w=0.65u l=1u
mm0 net1 vbias gnd  gnd nch w=5.42u l=1u
.ends
.subckt top vin vip vout
xxop vb vin vip out V G cmopa
xxbias vb V G bias_circuit
.ends
```

```
.subckt cmopa vbias vin vip out vdd gnd
x0 vin vip net1 net2 net3 dp_nmos
x1 net2 net4 vdd cm_pmos
x2 net3 out vdd cm_pmos
x3 net4 out gnd cm_nmos
x4 vbias net1 gnd nmos_vbias
.ends
```

```
`include "RealType_analog_behavioral_model.v"

module cmopa (vbias, vin, vip, out, vdd, gnd);
inout vbias, vin, vip, out, vdd, gnd;
dp_nmos x0(.n_inLdataV(vin) ,.n_inRdataV(vip) ,.downdataI(net1) ,.n_out
cm_pmos x1(.p_indataV(net2) ,.p_outdataI(net4) ,.up(vdd));
cm_pmos x2(.p_indataV(net3) ,.p_outdataI(out) ,.up(vdd));
cm_nmos x3(.n_indataV(net4) ,.n_outdataI(out) ,.down(gnd));
nmos_vbias x4(.indataV(vbias) ,.outdataI(net1) ,.down(gnd));
endmodule

module top (vin,vip,vout);
inout vin,vip,vout;
wire vin,vip,vout;
wire vb,V,G;
cmopa xxop(.vbias(vb),.vin(vin),.vip(vip),.out(vout),.vdd(V),.gnd(G));
bias_circuit xxbias(.out(vb),.vdd(V),.gnd(G));
endmodule
```

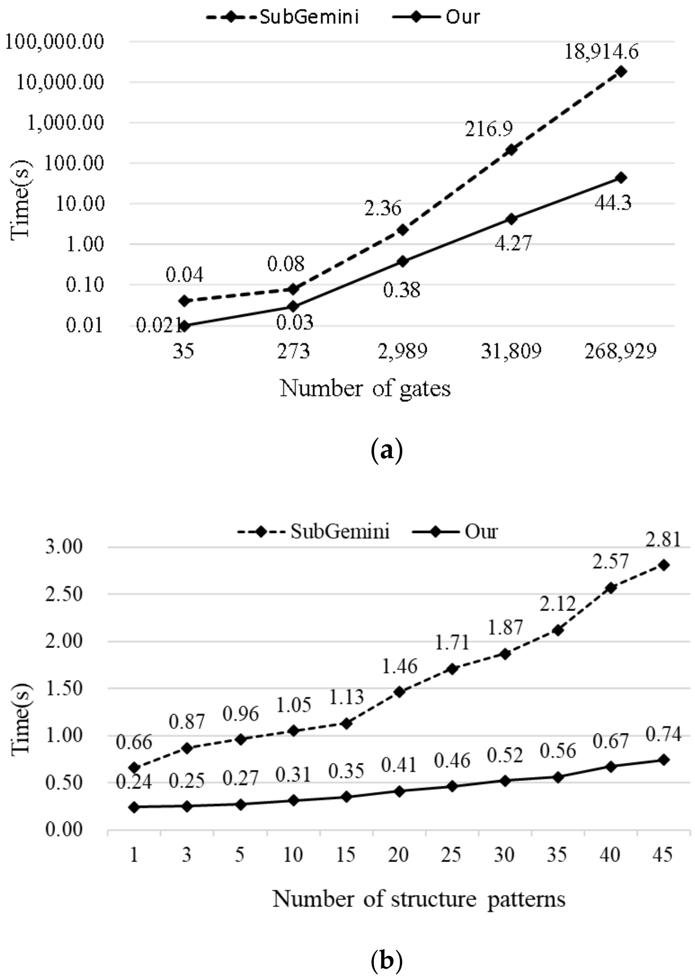
```
module cmopa (vbias, vin, vip, out, vdd, gnd);
inout vbias, vin, vip, out, vdd, gnd;
dp_nmos #(.Gm(0.00017354)) x0(.n_inLdataV(vin),.n_inRda
cm_pmos #(.ratio(1.87952)) x1(.p_indataV(net2),.p_outda
cm_pmos #(.ratio(1.87952)) x2(.p_indataV(net3),.p_outda
cm_nmos #(.ratio(1)) x3(.n_indataV(net4),.n_outdataI(ou
nmos_vbias #(.Vth(0.483539),.beta(0.0019656)) x4(.inda
endmodule
```

Figure 15. Example of the model construction and calibration process.

6. Experimental Results

6.1. Complexity Analysis of the Proposed Methodology

In the first experiment, we would like to demonstrate the efficiency improvement of the proposed structure recognition method compared to the graph-based approach. Classical graph-based identification approaches, such as SubGemini [32], test two graphs for isomorphism with exponential complexity in worst case. In order to demonstrate that the proposed methodology has linear time complexity, we analyzed the complexity in two aspects: design complexity and library complexity. The structure recognition was implemented in C++ and evaluated on a Linux workstation with an Intel Xeon 3.50GHz CPU and 64GB RAM. The analysis of design complexity is shown in Figure 16a; both recognition rates of two approaches are 100%. We use a fixed number of digital cells to generate five random circuits with the desired number of transistors (100, 1000, 10,000, 100,000, 1,000,000). “Number of Gates” shows the total number of blocks in the test files. As the design complexity grows, the proposed approach is able to obtain more speedup, which are 1.9×, 2.67×, 6.2×, 50.8×, and 427× for different design sizes. This result shows that under the same accuracy, the time complexity of the proposed methodology only grows linearly with respect to the design size, versus the complexity of the graph-based approach, which grows exponentially.



(**a**)

(**b**)

Figure 16. Efficiency comparison with graph-based method under growing of (**a**) design size (**b**) library size.

In order to analyze the impacts of library complexity, we generated random test circuits with 2500 gates by using a different number of structure patterns in the library, which are reported as “Number of structure patterns” in Figure 16b. According to the results, both recognition rates of the two approaches are 100%. As the library size increases, the run time of graph-based approach increases linearly to the number of involved library cells because its strategy is identifying each possible structure one by one. This fatal drawback leads to these approaches failing to solve huge designs. However, in the proposed approach, runtime increases very slowly as the amount of structure increases. Due to the parallel recognition of library cells, the proposed approach is able to handle modern industrial designs with thousands of structure patterns in minutes.

6.2. Verification of Industrial Cases

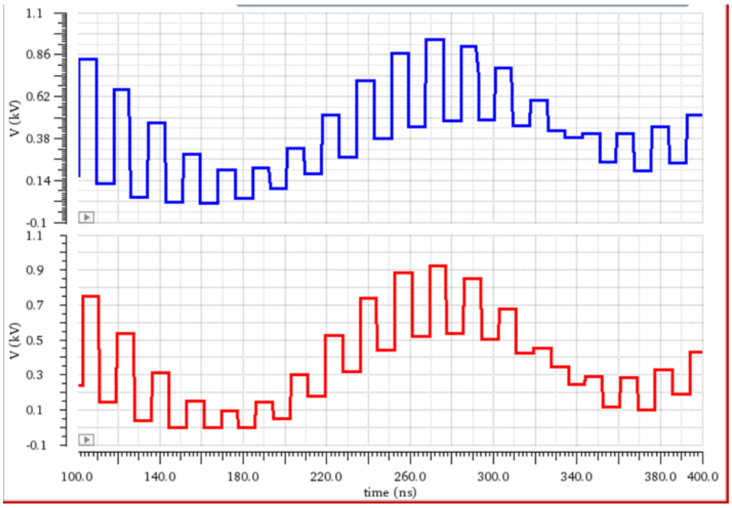
In this section, two real industrial cases, a parallel-in/serial-out (PISO) design (~15K transistors) and a Voltage Differential Analog-to-Digital Converter (VDADC) design (~246K transistors), are used to demonstrate the proposed structure recognition methodology and behavioral model generation environment. For easier integration with digital circuits, all the behavioral models are translated into Verilog format. The simulation tools for the behavioral model and transistor-level netlist are Cadence AMS, which distributes circuits to NCSim 2015, and Spectre 2017, respectively.

The structure analysis results of these cases are shown in Table 2. The PISO design can be divided into 50 analog blocks and 198 digital blocks; the VDADC design can be divided into 461 analog blocks and 98 digital blocks. For the two cases, extracting the behavioral models from transistor-level netlist could be completed in 5 and 75 s respectively, which is shown in the column “Extract”. For such large circuits, the extraction time is quite good because the two cases cannot be processed by the graph-based method at all.

Table 2. Structure recognition results for industrial cases.

Parallel-In/Serial-Out Design (PISO)			
#Trans.	Extract	Beh. Sim.	Spectre
15K	<5 s	00:39:25 (147×)	96:45:00 (1×)
Recognized Digital Blocks			
#RLCD	#TG	#Std Cell	#DCC+current
4	0	198	14
Recognized Analog Blocks			
L1	L2	L3	>L4
35	15	0	0
Voltage Differential Analog-to-Digital Converter (VDADC)			
#Trans.	Extract	Beh. Sim.	Spectre
246K	<75 s	00:5:25 (132×)	11:55:47 (1×)
Recognized Digital Blocks			
#RLCD	#TG	#Std Cell	#DCC+current
371	45	16	37
Recognized Analog Blocks			
L1	L2	L3	>L4
259	146	39	17

After the behavioral models of those basic blocks are built, they are connected together to perform behavioral simulation in AMS mode. The VDADC model is simulated with a Cadence Verilog simulator, NCSim 2015 version, and the transistor-level VDADC circuit is simulated with a Cadence Spectre 2017 version with moderate accuracy default. The stop time of both transient analysis is 400ns. The simulation results of the VDADC are shown in Figure 17; the root-mean-square error (RMSE) of the behavioral model is merely 0.0693, which shows that our behavioral model is highly accurate for capturing the behavior of design. For the PISO design, the simulation time of the behavior-level simulation (Beh. Sim.) and the transistor-level simulation (Spectre) are 2365 s and 348,300 s, respectively, with the same input patterns. In other words, we can obtain 147× speedup for system verification with behavioral models. For the VDADC design, we can obtain 132× speedup, in this case for system verification with behavioral models.



AMS Simulation

Model: generated behavioral model by the proposed methodology
Simulator: Co. sim by Spectre and ncsim
Sim. Time: 5m 25s (132X)
RMSE = 6.93%

Spice-level Simulation

Model: transistor–level model
Simulator: Spectre
Sim. Time: 11hr 55m 47s (1X)

Figure 17. Simulation waveforms of the VDADC (Voltage Differential Analog-to-Digital Converter) design.

7. Conclusions and Future Works

In order to support automatic behavioral model generation from given mixed-signal netlists, this paper proposes a fast approach to identify the pre-defined sub-blocks from the flattened transistor-level netlist. Unlike previous techniques that work for pure digital designs only, the proposed recognition flow is able to deal with real cases in which analog structures, digital structures, and even non-transistor devices are mixed together. The proposed hierarchical structure recognition methodology and the special circuit encoding scheme significantly improve the efficiency of sub-circuit identification, which help this flow to deal with large cases. As demonstrated with two industrial cases, the efficiency of the AMS system simulation is improved by hundreds of times with the automatically extracted behavioral models.

Structure recognition is widely used in many computer-aided design fields. For example, the netlist analysis can help to extract layout constraints, which are necessary information in physical design automation. Programmable electrical rules checking (PERC), which is a popular method for checking reliability issues of the integrated circuit, requires a robust and efficient structure analysis engine to handle large systems with millions of transistors. In the future, the proposed structure recognition also can be extended and applied to efficiently solve these urgent issues.

Author Contributions: Conceptualization, C.-N.L. and M.-J.L.; methodology, L.-Y.S., Y.-K.L. and C.-H.L.; software, L.-Y.S., Y.-K.L. and C.-H.L.; validation, L.-Y.S., Y.-K.L. and C.-H.L.; investigation, L.-Y.S., Y.-K.L. and C.-H.L.; resources, C.-N.L., J.-D.H., J.-Y.J., M.-J.L. and Y.-L.L.; data curation, C.-N.L. and L.-Y.S.; writing—original draft preparation, L.-Y.S.; writing—review and editing, C.-N.L.; visualization, L.-Y.S.; supervision, C.-N.L., J.-D.H. and J.-Y.J.; project administration, C.-N.L. and M.-J.L.; funding acquisition, C.-N.L., M.-J.L. and Y.-L.L. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded in part by Realtek Co., Ltd and MOST of Taiwan under Grant MOST109-2221-E-009-131-MY2.

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

References

1. Fahmy, A.; Liu, J.; Kim, T.; Maghari, N. An All-Digital Scalable and Reconfigurable Wide-Input Range Stochastic ADC Using Only Standard Cells. *IEEE Trans. Circuits Syst. II Express Briefs* **2015**, *62*, 731–735. [CrossRef]
2. Liu, J.; Park, B.; Guzman, M.; Fahmy, A.; Kim, T.; Maghari, N. A Fully Synthesized 77-dB SFDR Reprogrammable SRMC Filter Using Digital Standard Cells. *IEEE Trans. Very Large Scale Integr. Syst.* **2018**, *26*, 1126–1138. [CrossRef]
3. Park, J.-E.; Hwang, Y.-H.; Jeong, D.-K. A 0.5-V Fully Synthesizable SAR ADC for On-Chip Distributed Waveform Monitors. *IEEE Access* **2019**, *7*, 63686–63697. [CrossRef]
4. Aiello, O.; Crovetti, P.; Alioto, M. Fully Synthesizable Low-Area Analogue-to-Digital Converters With Minimal Design Effort Based on the Dyadic Digital Pulse Modulation. *IEEE Access* **2020**, *8*, 70890–70899. [CrossRef]
5. Aiello, O.; Crovetti, P.; Toledo, P.; Alioto, M. Rail-to-Rail Dynamic Voltage Comparator Scalable down to pW-Range Power and 0.15-V Supply. In *IEEE Transactions on Circuits and Systems II: Express Briefs*; IEEE: Piscataway, NJ, USA, 2021. [CrossRef]
6. Hakhamaneshi, K.; Werblun, N.; Abbeel, P.; Stojanovic, V. BagNet: Berkeley Analog Generator with Layout Optimizer Boosted with Deep Neural Networks. In Proceedings of the 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Westminster, CO, USA, 26 December 2019; pp. 1–8.
7. Settaluri, K.; Haj-Ali, A.; Huang, Q.; Hakhamaneshi, K.; Nikolic, B. AutoCkt: Deep Reinforcement Learning of Analog Circuit Designs. In Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 15 June 2020; pp. 490–495.
8. Felt, E.; Zanella, S.; Guardiani, C.; Sangiovanni-Vincentelli, A. Hierarchical statistical characterization of mixed-signal circuits using behavioral modeling. In Proceedings of the International Conference on Computer Aided Design, San Jose, CA, USA, 6 August 2002.
9. Duran, P.A. *A Practical Guide to Analog Behavioral Modeling for IC System Design*; Metzler, J.B., Ed.; Springer Science & Business Media: Berlin, Germany, 1998.
10. Nassif, S.R. Modeling and analysis of manufacturing variations. In Proceedings of the IEEE 2001 Custom Integrated Circuits Conference (Cat. No.01CH37169), San Diego, CA, USA, 7 August 2002.
11. Fraccaroli, E.; Lora, M.; Fummi, F. Automatic Generation of Analog/Mixed Signal Virtual Platforms for Smart Systems. *IEEE Trans. Comput.* **2020**, *69*, 1263–1278. [CrossRef]
12. De Tommasi, L.; Gorissen, D.; Croon, J.A.; Dhaene, T. Surrogate Modeling of RF Circuit Blocks. In *Progress in Indus-Trial Mathematics at ECMI*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 447–452.
13. Khandelwal, S.; Garg, L.; Boolchandani, D. Reliability-Aware Support Vector Machine-Based High-Level Surrogate Mod-el for Analog Circuits. In *IEEE Transactions on Device and Materials Reliability*; IEEE: Piscataway, NJ, USA, 2015; Volume 15, pp. 461–463.
14. Garbaya, A.; Kotti, M.; Fakhfakh, M.; Tlelo-Cuautle, E. On the accurate modeling of analog circuits via the Kriging meta-modeling technique. In Proceedings of the 14th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Ap-plications to Circuit Design (SMACD), Giardini Naxos, Italy, 17 July 2017; pp. 1–4.
15. Rewienski, M.; White, J. A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*; IEEE: Piscataway, NJ, USA, 2003; pp. 155–170.
16. Demir, A.; Roychowdhury, J. A reliable and efficient procedure for oscillator PPV computation, with phase noise macromodeling applications. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2003**, *22*, 188–197. [CrossRef]
17. Yu, Y.S.; Lee, H.S.; Hwang, S.W. Macro-Modeling of Single Electron Transistors for Efficient Circuit Simulation. In *Extended Abstracts of the 1998 International Conference on Solid State Devices and Materials*; JSAP: Hiroshima, Japan, 1998; pp. 196–197. [CrossRef]
18. Biswas, D.; Priyoti, A.T.; Khosru, Q.D.M. Programmable Single Electron Transistor: A modified Macro-model & its Applications. In Proceedings of the 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 2 November 2020; pp. 1106–1109.
19. Wei, Y.; Doboli, A. Systematic development of analog circuit structural macromodels through behavioral model de-coupling. In Proceedings of the 42nd Design Automation Conference, San Diego, CA, USA, 13–17 June 2005; pp. 57–62.
20. Wei, Y.; Doboli, A. Structural Macromodeling of Analog Circuits Through Model Decoupling and Transformation. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2008**, *27*, 712–725. [CrossRef]
21. Georgoulopoulos, N.; Hatzopoulos, A. Real number modeling of a flash ADC using SystemVerilog. In Proceedings of the 2017 Panhellenic Conference on Electronics and Telecommunications (PACET), Xanthi, Greece, 18 January 2018; pp. 1–4.
22. Mounir, A.; Mostafa, A.; Fikry, M. Automatic behavioural model calibration for efficient PLL system verification. In Proceedings of the 2003 Design, Automation and Test in Europe Conference and Exhibition, Munich, Germany, 19 December 2003; pp. 280–285.
23. Vogels, M.; De Smedt, B.; Gielen, G. Modeling and simulation of a sigma-delta digital to analog converter using VHDL-AMS. In Proceedings of the 2000 IEEE/ACM International Workshop on Behavioral Modeling and Simulation, Orlando, FL, USA, 6 August 2002; pp. 5–9.
24. Malcovati, P.; Brigati, S.; Francesconi, F.; Maloberti, F.; Cusinato, P.; Baschirotto, A. Behavioral modeling of switched-capacitor sigma-delta modulators. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **2003**, *50*, 352–364. [CrossRef]
25. Kuo, C.-C.; Wang, Y.-C.; Liu, C.-N.J. An efficient bottom-up extraction approach to build accurate PLL behavioral models for SOC designs. In *Proceedings of the 15th ACM Great Lakes Symposium on VLSI - GLSVLSI '05*; ACM: Chicago, IL, USA, 2005; pp. 286–290. [CrossRef]

26. Wambacq, P.; Fernández, F.; Gielen, G.; Sansen, W.; Rodríguez-Vázquez, A. Efficient Symbolic Computation of Approxi-mated Small-signal Characteristics. *IEEE J. Solid-State Circuits* **1995**, *30*, 327–330. [CrossRef]
27. Chen, J.-Y.; Wang, S.-W.; Lin, C.-H.; Liu, C.-N.; Lin, Y.-J.; Lee, M.-J.; Lo, Y.-L.; Kao, S.-Y. Automatic Behavioral Model Genera-tor for Mixed-Signal Circuits Based on Structure Recognition and Auto-Calibration. In Proceedings of the IEEE International SOC Design Conference, Gyeongju, Korea, 11 February 2016.
28. Song, L.-Y.; Wang, C.; Liu, C.-N.; Lin, Y.-J.; Lee, M.-J.; Lo, Y.-L.; Kao, S.-Y. Non-Regression Approach for the Behavioral Model Generator in Mixed-Signal System Verification. In Proceedings of the IFIP/IEEE International Conference on VLSI-SOC, Abu Dhabi, United Arab Emirates, 14 December 2017; pp. 1–5.
29. Chang, W.-H.; Tzeng, S.-D.; Lee, C.-Y. A novel subcircuit extraction algorithm by recursive identification scheme. In Proceedings of the ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196), Sydney, NSW, Australia, 7 August 2002; pp. 491–494.
30. Ren, Y.; Shi, Y.; Gwee, B.; Ting, C.W. An efficient VLSI circuit extraction algorithm for transistor-level to gate-level ab-straction. In Proceedings of the 2010 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), Shanghai, China, 18 October 2010; pp. 49–52.
31. Huang, L.; Jiang, T.Y.; Jou, J.Y.; Huang, H.L. An Efficient Logic Extraction Algorithm Using Partitioning and Cir-cuit-Encoding. In Proceedings of the IEEE International Symposium on Circuits and Systems, Vancouver, BC, Canada, 3 September 2004.
32. Ohlrich, M.; Ebeling, C.; Ginting, E.; Sather, L. SubGemini: Identifying Subcircuits Using a Fast Subgraph Isomorphism Algorithm. In Proceedings of the IEEE/ACM Design Automation Conference, Dallas, TX, USA, 14–18 June 1993; pp. 31–37.
33. Hansen, M.; Yalçin, H.; Hayes, J. Unveiling the ISCAS-85 benchmarks: A case study in reverse engineering. *IEEE Des. Test Comput.* **1999**, *16*, 72–80. [CrossRef]
34. Bourbakis, N.; Mogzadeh, A.; Mertoguno, S.; Koutsougeras, C. A knowledge-based expert system for automatic visual VLSI reverse-engineering: VLSI layout version. *IEEE Trans. Syst. Man, Cybern. Part A Syst. Humans* **2002**, *32*, 428–436. [CrossRef]
35. Ullman, J. An Algorithm for Subgraph Isomorphism. *J. Assoc. Comput. Mach.* **1976**, *23*, 31–42. [CrossRef]
36. Messmer, B.; Bunke, H. Efficient subgraph isomorphism detection: A decomposition approach. *IEEE Trans. Knowl. Data Eng.* **2000**, *12*, 307–323. [CrossRef]
37. Vijaykrishnan, N.; Ranganathan, N. SUBGEN: A genetic approach for subcircuit extraction. In Proceedings of the 9th International Conference on VLSI Design ICVD-96, Bangalore, India, 6 August 2002.
38. Zhang, N.; Wunsch, D.C. A fuzzy attributed graph approach to subcircuit extraction problem. In Proceedings of the 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ '03., St. Louis, MO, USA, 25 June 2003; pp. 1063–1067.
39. Zhang, N.; Wunsch, D.C. A Novel Subcircuit Extraction Algorithm Using Heuristic Dynamic Programming. In Proceedings of the International Conference on VLSI, Las Vegas, NV, USA, 24–27 June 2002; CSREA Press: London, UK, 2002; pp. 38–44.
40. Rubanov, N. SubIslands: The probabilistic match assignment algorithm for subcircuit recognition. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2003**, *22*, 26–38. [CrossRef]
41. Rubanov, N. A High-Performance Subcircuit Recognition Method Based on the Nonlinear Graph Optimization. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2006**, *25*, 2353–2363. [CrossRef]
42. Su, H.-Y.; Hsu, C.-H.; Li, Y.-L. SubHunter: A High-Performance and Scalable Sub-circuit Recognition Method with Pru-fer-Encoding. In Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 23 April 2015.
43. Liou, G.-H.; Wang, S.-H.; Su, Y.-Y.; Lin, M.P.-H. Classifying Analog and Digital Circuits with Machine Learning Tech-niques toward Mixed-Signal Design Automation. In Proceedings of the IEEE International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design, Prague, Czech Republic, 16 August 2018.
44. Eick, M.; Strasser, M.; Lu, K.; Schlichtmann, U.; Graeb, H.E. Comprehensive Generation of Hierarchical Placement Rules for Analog Integrated Circuits. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*; IEEE: Piscataway, NJ, USA, 2011; Volume 30, pp. 180–193. [CrossRef]
45. Li, H.; Jiao, F.; Doboli, A. Analog Circuit Topological Feature Extraction with Unsupervised Learning of New Sub-Structures. In Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 28 April 2016; pp. 1509–1512.
46. Najm, F.N. Solution of Linear Algebraic Circuit Equations. In *Circuit Simulation*; Wiley-IEEE Press: Piscataway, NJ, USA, 2010; pp. 49–126.
47. Massier, T.; Graeb, H.; Schlichtmann, U. The Sizing Rules Method for CMOS and Bipolar Analog Integrated Circuit Synthesis. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*; IEEE: Piscataway, NJ, USA, 2008; Volume 27, pp. 2209–2222. [CrossRef]
48. Long, D.; Hong, X.; Dong, S. Signal-path driven partition and placement for analog circuit. In Proceedings of the Asia and South Pacific Conference on Design Automation, Yokohama, Japan, 13 March 2006; p. 694.
49. Lin, Y.-J.; Lee, M.-J.; Lo, Y.-L.; Kao, S.-Y. Automatic mixed-signal behavioral model generation environment. In Proceedings of the 2016 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), Hsinchu, Taiwan, 2 June 2016; pp. 1–4.