Telmo Filipe Pereira Ferraria

Verification of Analog Circuits in Power-Down Mode

Dissertação para obtenção do Grau de Mestre em
Engenharia Eletrotécnica e Computadores

Orientadora : Maria Helena Fino, Doutora, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Co-orientador : Helmut Gräeb, Doutor, Technische Universität München

Júri:

Presidente: Fernando José Vieira do Coito

Arguentes: João Pedro Oliveira

Vogais: Maria Helena Fino
Helmut Gräeb

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2013

Verification of Analog Circuits in Power-Down Mode

"Aprende como se tu fosses viver para sempre. Vive como se tu fosses morrer amanhã".
Mahatma Gandhi

Acknowledgements

Este espaço é dedicado a todos aqueles que me acompanharam durante o percurso académico na FCT e na TUM-EDA. Para todos eles, o meu mais sincero obrigado!

Agradeço á Professora Doutora Maria Helena Fino a amizade orientação e apoio constante ao longo de todo o meu percurso académico assim como na minha dissertação. Gostaria também de lhe agradecer a oportunidade de ter realizado a minha dissertação em Munique, com esta eu cresci muito como pessoa.

I would like to thank to Dipl. Eng. Michael Zwerger and Prof. Dr. Eng. Helmut Gräeb for providing an excellent working environment and for all the technical support and advice that made this work possible all the time that I have been in the TUM-EDA.

Ao Departamento de Engenharia Eletrotécnica da Universidade Nova de Lisboa pelo uso das instalações e por ter sido ao longo destes anos uma segunda casa.

Gostaria também de agradecer aos meus colegas de casa Miguel Duarte, Pedro Cartaxo e Sérgio Pinto por todo o tempo que passamos juntos. Aos colegas Fábio Passos, Rui Lopes, Marisa Amaral e a Filipa Lourenço porque foram a minha segunda família durante estes anos.

Ao Pedro Leitão todos os momentos que passamos desde pequenos como jogadores de hóquei até aos últimos três meses em Munique.

Para os colegas Hugo Viana e José Vieira não tenho palavras para descrever o que passei com eles durante o meu percurso académico. Agradeço todo o apoio e força que me transmitiram sempre nas alturas mais complicadas. Todos os bons momentos passados.

Não menos importante, gostaria de mostrar a gratidão mais gentil e especial para

com a minha família em particular com os meus pais e irmã, por todo o apoio e carinho que me deram ao longo da minha vida e que sem eles seria impossível alcançar esta meta.

Por último, quero agradecer a minha namorada Natacha Duarte por todo o tempo que não estive presente, por o incentivo, orientação e amor que sempre me deu.

Abstract

The energy efficiency and optimization are two important points of analog circuits. With purpose to reduce the power consumption, most of these circuits are equipped with power-down features, which means the circuits are idle when they are not used. In power-down mode internal nodes can have floating states which results in an increase of the transistor degradation.

In this thesis a computer program that checks the node voltage levels and the state of the transistors in power-down mode is presented. The search procedure will ensure that all currents in the paths are safely turned off. The program works just with the structural information of the circuit given into a input file i.e net-list file. No numerical simulation is needed.

Experimental results show the efficacy and efficiency in industrial circuits and also the integration with the CADENCE software.

Resumo

A maximização energética bem como otimização nos circuitos analógicos é uma preocupação fundamental na concepção de circuitos analógicos. De forma a reduzir o seu consumo muitos circuitos incluem sistemas de $Power - Down$ que diminuem a corrente quando o mesmo não é necessário. Contudo ao se encontrar em modo $Power - Down$ algumas das tensões internas podem ter um valor indefinido o que por sua vez se traduz num aumento na degradação dos transístores até a sua falha.

Nesta dissertação um programa que estima corretamente os níveis de tensão e os estados dos transístores quando o circuito se encontra em modo $Power - Down$ é apresentado. O mesmo garante que não existe corrente a fluir nos circuitos em modo $Power - Down$.

Para o funcionamento do programa apenas um ficheiro contendo o esquemático do circuito sem dimensões é necessário.

Os resultados experimentais mostram grande eficácia e rapidez no funcionamento do programa em grandes circuitos bem como a sua integração com o simulador de circuitos CADENCE.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Motivation

Reliability in CMOS is one of the biggest problems for the analog circuits in the world. Reliability, is defined as the probability of a product operating for a given amount of time under specified conditions without failure [Ohr98]. Reliability is highly affected by circuit aging, i.e, the deterioration of the circuit's performance over its lifetime. The lifetime can vary from a few years to a few months under worst-case scenarios.

Circuit deterioration may lead to power consumption increase and in extreme cases, circuit aging may even cause functional failures to occur.

The introduction of new materials in CMOS technologies raised additional mechanisms that may become faulty or generate system failures. Hot carrier injection (HCI) and bias temperature instability (BTI) are two push mechanisms responsible for the degradation of transistors and consequently of an integrated circuit [MG11],
[CMFSL11].

These modifications resulted in a change of CMOS transistors characteristics, such as the threshold voltage, decrease in drain current and transconductance [SBA+03].

Furthermore circuit aging is particularly affected by mismatching. A mismatch may occur due to either process variations or stress-induced degradation during the device operation, such as when large asymmetrical voltages are applied to the transistors [CZT+01], [MG11].

Experimental studies have shown that mismatches in differential amplifiers and current mirrors are reinforced by HCI degradation, which in turn contributes to the degradation of the circuits performance over time [PG11].

The power consumption of analog circuits is always present in the mind of circuit

designers. In order to save battery power and to reduce the chip heat most of the analog circuits are equipped with power-down features, which means the circuit will be idle when not in use.

During power-down mode, the potentials of the internal nodes is determined by the sub-threshold characteristics of the devices, leakage paths and by the signals applied to the inputs [SBA+03].

This nodes can cause asymmetrical stress conditions in structures.

1.2 Goals of the Work

While it is relatively easy to check small circuits manually, larger circuits often have many pitfalls that could be easily missed. The overall goal of this Thesis is to develop a computer program capable of analyzing large circuits, with the ability to estimate node voltage levels and the state of transistors in power-down mode, based on the circuit structure.

This program detects if there is current flowing in paths when the circuit is in power-down mode. A fully automatic checking system of stress-sensitive structures was proposed as a future work [ZG12]. Therefore the program implements this automatic detection.

Furthermore, in order to help the circuit-designer, the program is integrated into CADENCE Software.

1.3 State of the Art

Since the problem addressed is relatively new, very few work has been developed in this area. This section presents the relevant research concerning the development of algorithms to use in analog circuits in power-down mode. The first algorithm was proposed by [BJ96]. They developed two versions of the voltage propagation algorithm, one considered a basic version and another one considered to be a more elaborated version.

These algorithms take under consideration two linked lists which contain all components and all nodes of the circuit. In the basic version three voltage levels are considered, *gnd* level which is equal to the lowest voltage, *vdd* which is equal to the positive supply voltage or *float* when the voltage is unknown or the node has high impedance. The basic version is limited to resistors and MOS devices, the possibility of current being put into forward biased junctions is not considered and diodes components are equally not considered.

In the elaborated version, the limitations of the basic version are overcome by extending the number of the possible voltage levels of a node to six. In this version they consider *gnd*, *vdd*, *float*, *curr*, *pullup* and *pulldown* voltage levels.

The last two voltage levels are used to deal with the diode component, and the *curr*

voltage level is used when current is detected in a component. Yet, even with this definition the node voltage levels are not correctly estimated.

Recently a new version of the voltage propagation algorithm was proposed [ZG12]. Although it is based on the same principles as the ones used by [BJ96], this version includes MOS and bipolar transistors, resistor, capacitor and diodes.

Furthermore, some transistor configurations are considered, such diode configuration. The circuit is transformed into a directed graphical representation, containing all connections and nodes of the circuit.

The proposed algorithm indicates if the node voltages are correctly estimated and floating nodes are detected.

Both versions of voltage propagation algorithm rely on the representation of the circuit from an input file which contains the net-list of the circuit. No electrical simulation or aging simulation is needed. Therefore, algorithms that identify the types of components and nodes existing in the net list file are needed [MGS08].

These algorithms involve a creation of an hierarchical library which contains basic CMOS and bipolar build blocks, like e.g., transistors, current mirrors and a differential stage. Then the algorithm identifies the presence of these elements in the given circuit net-list.

1.4 Outline of the Thesis

This thesis is organized in five chapters. This first chapter gives the outline of the scope of the thesis. In the chapter 2 a problem formulation and the voltage propagation behavior in elementary blocks are presented. Chapter 3 describes the tool for the analysis for verification of analog circuits in power-down mode. This includes the voltage propagation, the short circuit and the stress analysis. Chapter 4 discusses the results obtained with the algorithms that were used and it also incorporates a comparison between the DC analysis and voltage propagation analysis. Finally chapter 5 gives the overall conclusions and offers suggestions with regards to future research.

2

Voltage Propagation in Power-Down

In this chapter an introduction to the voltage propagation behaviour in circuits in power-down mode is presented.

Firstly, a brief formulation of the problem is introduced in section 2.1.

Then, to detect the problems described in section 2.1, each elementary circuit component was tested. These tests are described section 2.2 where a theoretical prevision is presented and the corresponding transient analysis is performed to validate the assumptions made.

2.1 Problem Formulation

The problems of analog circuits in power-down mode are illustrated with the differential stage represented in figure 2.1.

Here, transistor M_1 is a power-down switch, so in order to put the circuit in the idle mode it is necessary to turn the current in M_2 off by connecting the n_{pwd} to a negative supply voltage. As a result of this, n_{bias} will be pulled up to *vdd*. Since M_2 is off there is no tail current, consequently no current flows through in M_3 or M_4. The voltages of n_1 and n_3 nodes depend on the sub-threshold characteristics of the devices, leakage paths and by the signals applied to the inputs. As these voltage levels cannot be defined as *vss* or *vdd* it is considered that nodes n_1 and n_3 are floating.

In power-down mode, if the gate of a transistor is floating, it is impossible to define in which state it's in. Thus, potential current flow in the paths that contain these transistors. In matched structures this kind of node will cause asymmetrical stress conditions.

As this situations will lead to severe aging degradation in circuit performance, a

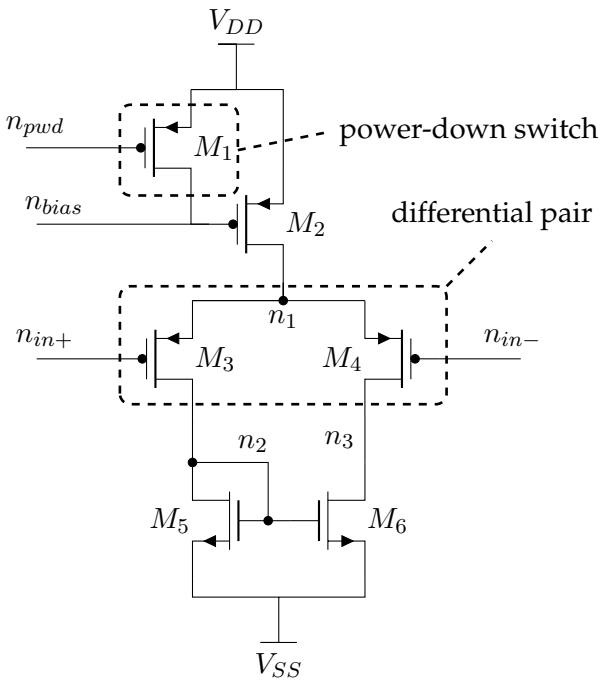


Figure 2.1: Differential Stage With Power-Down Switch [ZG12]

methodology for detecting these situations is of great importance. In order to automatically identify stress conditions, the voltage propagation along the circuits is addressed in the following subsection.

2.2 Voltage Propagation in elementary blocks

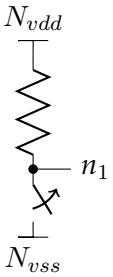
According to previous publication [ZG12], a voltage propagation algorithm can be developed to estimate the node voltages and detect floating nodes in power-down mode.

With the purpose of illustrating how the voltage propagation works in circuits, a first analysis of the propagation in the elementary blocks is addressed.

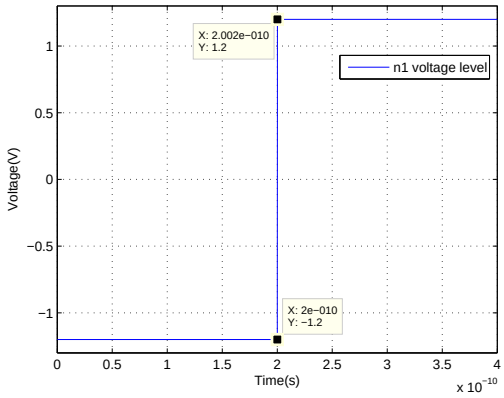
For the voltage propagation to work well it is fundamental to guaranty that the current is interrupted giving rise to an adjustment of the charges in the circuit.

The voltage propagation behaviour in a resistor is shown in Figures 2.2 and 2.3. In Figure 2.2(a), when the switch is open the current becomes zero and the voltage level in n_1 will rise to value of positive supply.

This behavior is illustrated in Figure 2.2(b) where transient simulation of the circuit is represented.



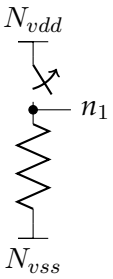
(a) Schematic



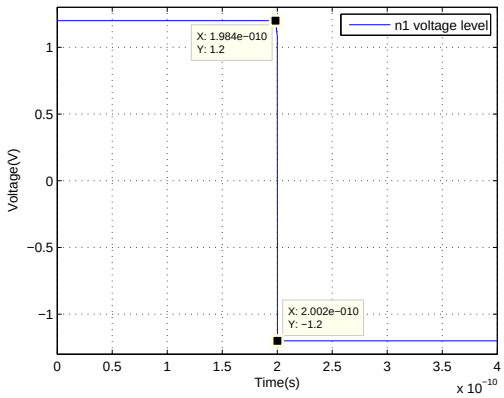
(b) Transient Simulation

Figure 2.2: Propagation of Positive Charge in Resistor

In the circuit represented in Figure 2.3 the voltage value of n_1 becomes equal to the negative supply.



(a) Schematic

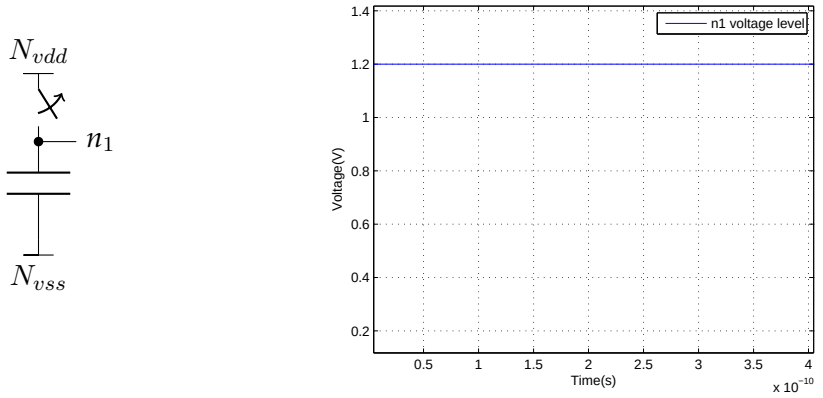


(b) Transient Simulation

Figure 2.3: Propagation of Negative Charge in Resistor

Since resistors have no polarity, the propagation method for the other direction is also to be considered.

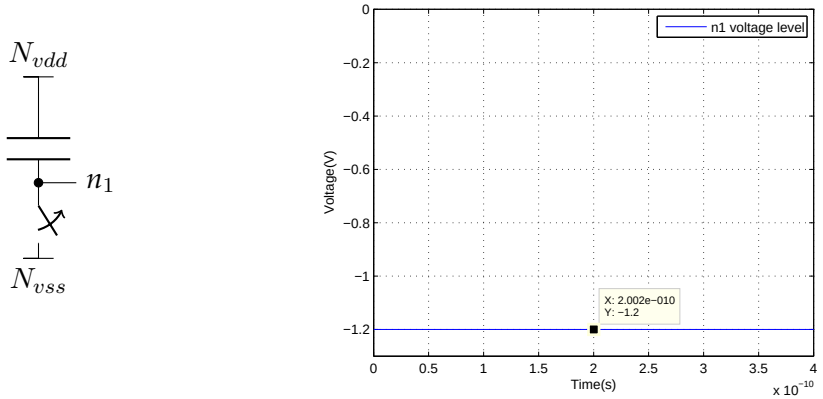
Figures 2.4 and 2.5 represent the voltage propagation in a capacitor. In this situation the charge in the capacitor remains constant. As result n_1 has the same value when the switch is open.



(a) Schematic (b) Transient Simulation

Figure 2.4: Propagation of Positive Charge in Capacitor

While in Figure 2.4(b) the transient analysis shows that the voltage remains in 1.2 V, in Figure 2.5(b) this value remains in -1.2 V before and after of the switch is open.

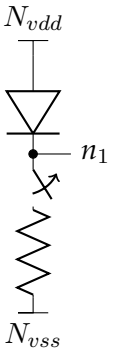


(a) Schematic (b) Transient Simulation

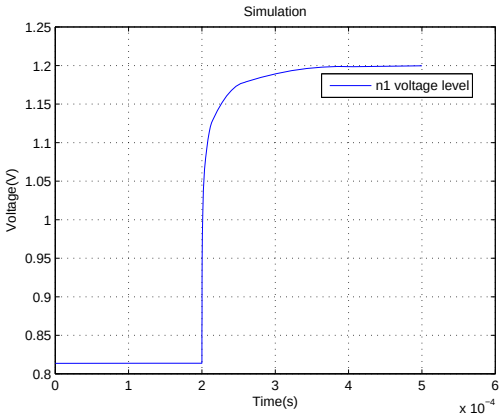
Figure 2.5: Propagation of Negative Charge in Capacitor

In the diode component the voltage propagation can occur only in one direction because the diode only conducts current if the anode has a higher voltage than the cathode.

The diode represented in Figure 2.6(a) was simulated. The charge will flow across the diode and n_1 will reach the positive supply value when the switch is open. The transient behaviour of the circuit illustrated in Figure 2.6(b) shows the expected result.



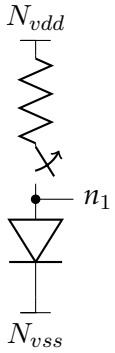
(a) Schematic



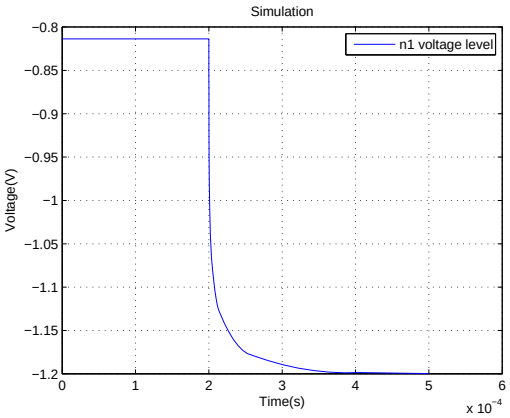
(b) Transient Simulation

Figure 2.6: Propagation of Positive Charge in Diode

For the case where the diode is connected in the configuration represented in Figure 2.7(a), when the switch is open the charge in n_1 will flow across the diode and n_1 will reach the value of the negative supply. The transient analysis results represented in Figure 2.7(b) proves the expected result.



(a) Schematic



(b) Transient Simulation

Figure 2.7: Propagation of Negative Charge in Diode

In transistors the propagation behaviour depends on the transistor configuration.

Figure 2.8 shows the MOS transistor in diode configuration. The propagation in this case is similar to the diode component.

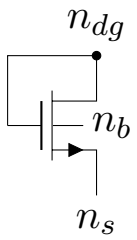


Figure 2.8: Transistor in Diode Configuration

Figure 2.9 shows a transistor in a general configuration. In this case the propagation will depend on the state of the transistor, thus several transistor biasing configuration will be discussed.

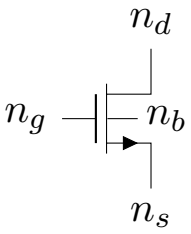
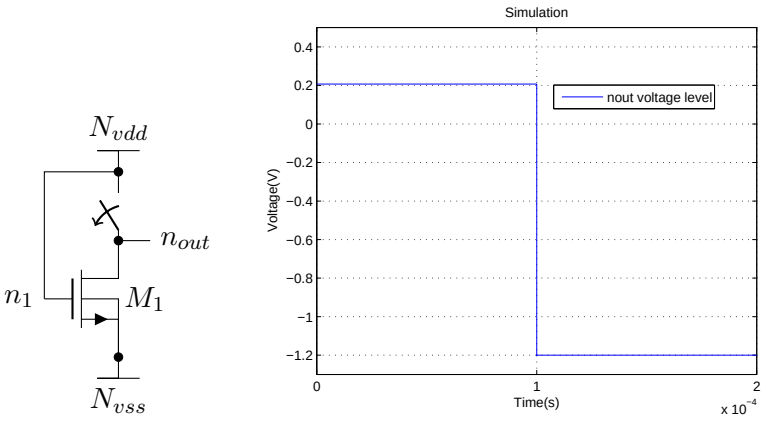


Figure 2.9: Transistor in General Configuration

The operation regions will depend on the gate voltage level, so the gate of the transistor will control the voltage propagation.

Figure 2.10(a) can be used to explain how the propagation works in circuits with transistors. In power-down mode the current in the path should be zero. Therefore, it is necessary open the switch. If the n_1 node was connected to N_{vdd} node then M_1 transistor can propagate, leading the propagation of the charge on N_{vss} to n_{out}. As result n_{out} node voltage becomes equals to the negative supply.

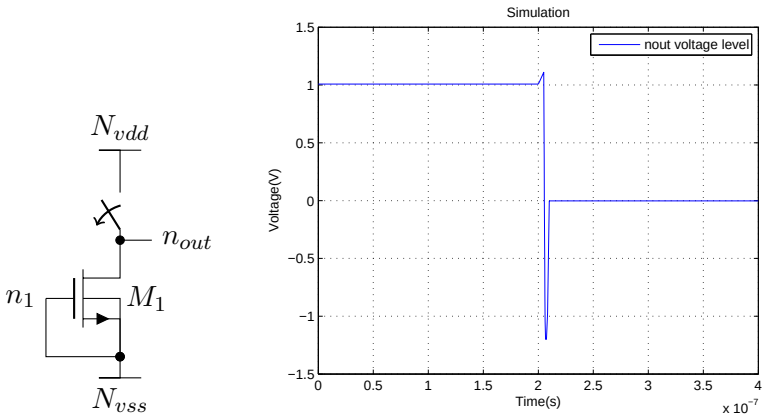


(a) Schematic (b) Transient Simulation

Figure 2.10: Propagation in Transistor with Gate Connected to N_{vdd}

The transient analysis represented in Figure 2.10(b) shows that n_{out} node change the voltage level to -1.2 when the gate of M_1 is connected to N_{vss}.

In the second case the circuit in Figure 2.11(a) with the n_1 node connected to N_{vss} is considered. As result M_1 transistor cannot propagate and n_{out} has floating state. In this situation it's not possible to define whether the voltage level in n_{out} is equal to a positive or negative supply and the transient analysis in Figure 2.11(b) can prove it.



(a) Schematic (b) Transient Simulation

Figure 2.11: Propagation in Transistor with Gate Connected to N_{vss}

The last case is similar to the previous one, if the gate of M_1 is floating, then it's impossible to define if the propagation is possible. Thus n_{out} has again floating state.

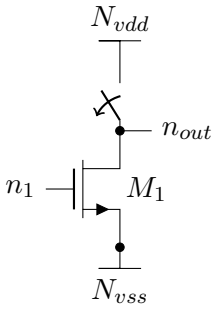


Figure 2.12: Propagation in Transistor with Gate floating

For the complementary type of transistor the same behavior was obtained.

Similar analysis may be directly applied to bipolar transistors, but in this case the current in the transistor base needs to forced to zero. Otherwise the propagations methods cannot be valid.

2.3 Summary

Stress conditions in circuits in power-down mode lead to severe aging problems in the circuits. To avoid stress conditions, voltage propagation in circuit must be performed so that such situations may be identified and duly corrected.

In this chapter stress conditions in differential pair and floating nodes were demonstrated. Then the voltage propagation in elementary circuit elements was presented as a way of introducing the verification tool presented in the next chapter.

3

Verification Tool for Circuits in Power-Down Mode

The analysis performed regarding the verification of power-down mode program will be described throughout this chapter.

The verification tool starts by applying the voltage propagation behavior described in Chapter 2. The methodology adopted for the automatic evaluation of the voltage propagation will be presented in section 3.1. Then, in section 3.2 a short circuit analysis yielding the evaluation of the paths for current flowing is presented.

Section 3.3 relates to a fully automated stress analysis check of stress-sensitive topologies.

Finally, in section 3.4 an abstract integration of all analysis is presented.

3.1 Voltage Propagation Analysis

Figure 3.1 shows the several steps performed during the voltage propagation analysis.

Firstly the *Structure Recognizer* block receives the circuit representation given in the netlist. This block runs the algorithm described in [MGS08], where elementary components are identified.

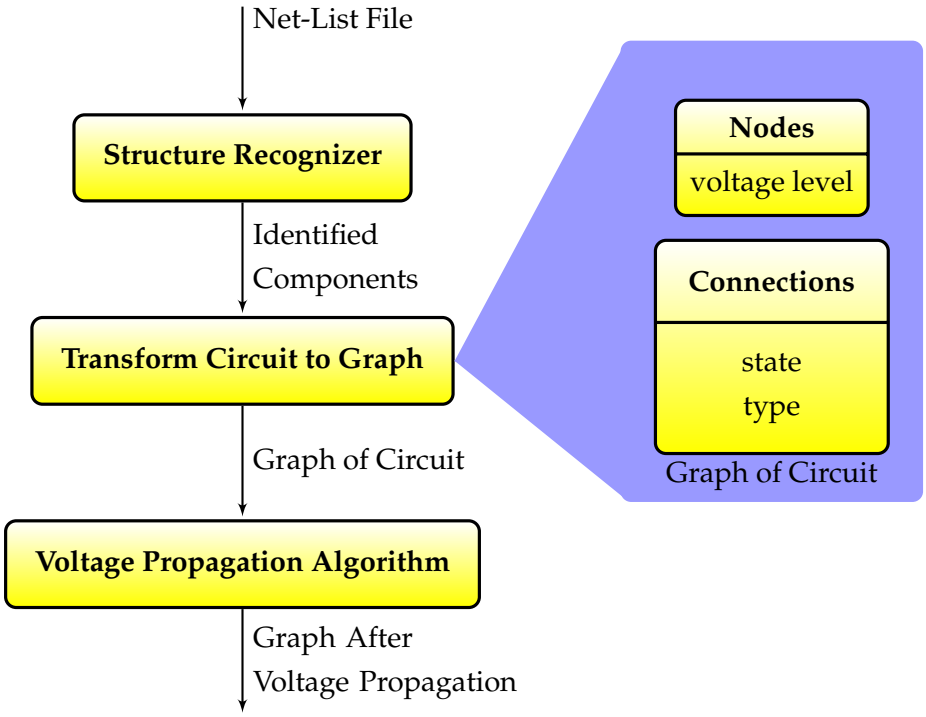


Figure 3.1: Block Diagram of an Automatic Voltage Propagation Analysis

In the *Transform Circuit to Graph* block, a graph representation of the interconnections between the previously identified structures is generated. In this graph, the nodes contain information regarding the corresponding voltage level, whereas the branches contain information relative to the type and the state of the connections between the nodes.

The types for connections considered are described in Section 3.1.1. The description of the states for each connection appears in Section 3.1.2 as well as the description of the voltage levels for the nodes.

Then the graph representation is used by the *Voltage Propagation Algorithm* block, responsible for the execution of the voltage propagation algorithm described in section 3.1.3. The propagation changes the voltage levels in nodes and the state of the connections in graph representation.

After the analysis has been completed, a graph representation containing an estimation of the voltage level in each node and the state of connections is obtained.

The graph representation obtained after voltage propagation can be used for both

short circuit and stress analysis.

3.1.1 Mapping of Circuit Components into Graph Branches

According to the type of component, different types of connections are generated. Each connection contains one of the three types identified below.

- ***diode-type*** (dio)
- ***n-switch*** (nsw)
- ***p-switch*** (psw)

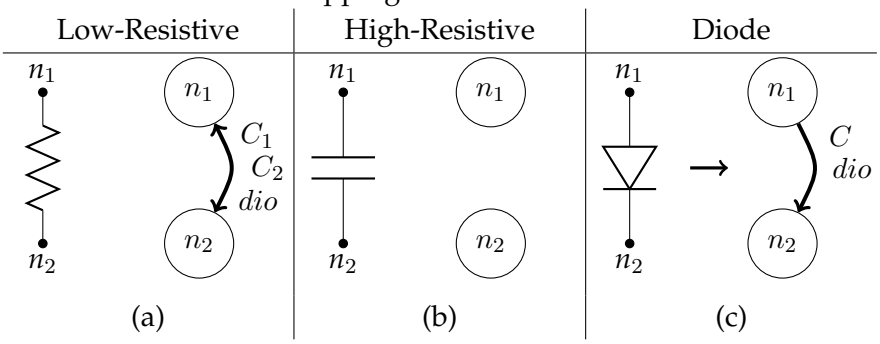
In this work, a component mapping for two terminal devices and transistors were included.

Figure 3.1 shows the two terminal devices considered while the transistors are considered in Figure 3.2.

For the two terminal devices three possible types are accounted for:

- **Resistor**: For this device two branches are generated because the resistor propagates the voltage in both directions. The column in Table 3.1 (a) shows these branches (C_1 and C_2) with bidirectional arrow and the type of them is *diode-type* (dio in the Table 3.1 (a)).
- **Capacitor**: In this case no connection is generated, because no current flows in either direction.
- **Diode**: In diode component one branch from anode to cathode is generated. This is represented in Table 3.1 (c). The type of this connection is diode *diode-type*.

Table 3.1: Mapping of Two Terminal Devices



For bipolar and MOS technologies three possible configurations are considered for the transistors.

- **Diode-Connected**: For NMOS or *npn*, one connection from drain/gate to source or base/collector to emitter is created. For PMOS or *pnp* one connection from source to drain/gate or from emitter to base/collector is created. In both cases the type of connection is *diode-type*. Table 3.2(a) shows a representation of this kind of configuration.

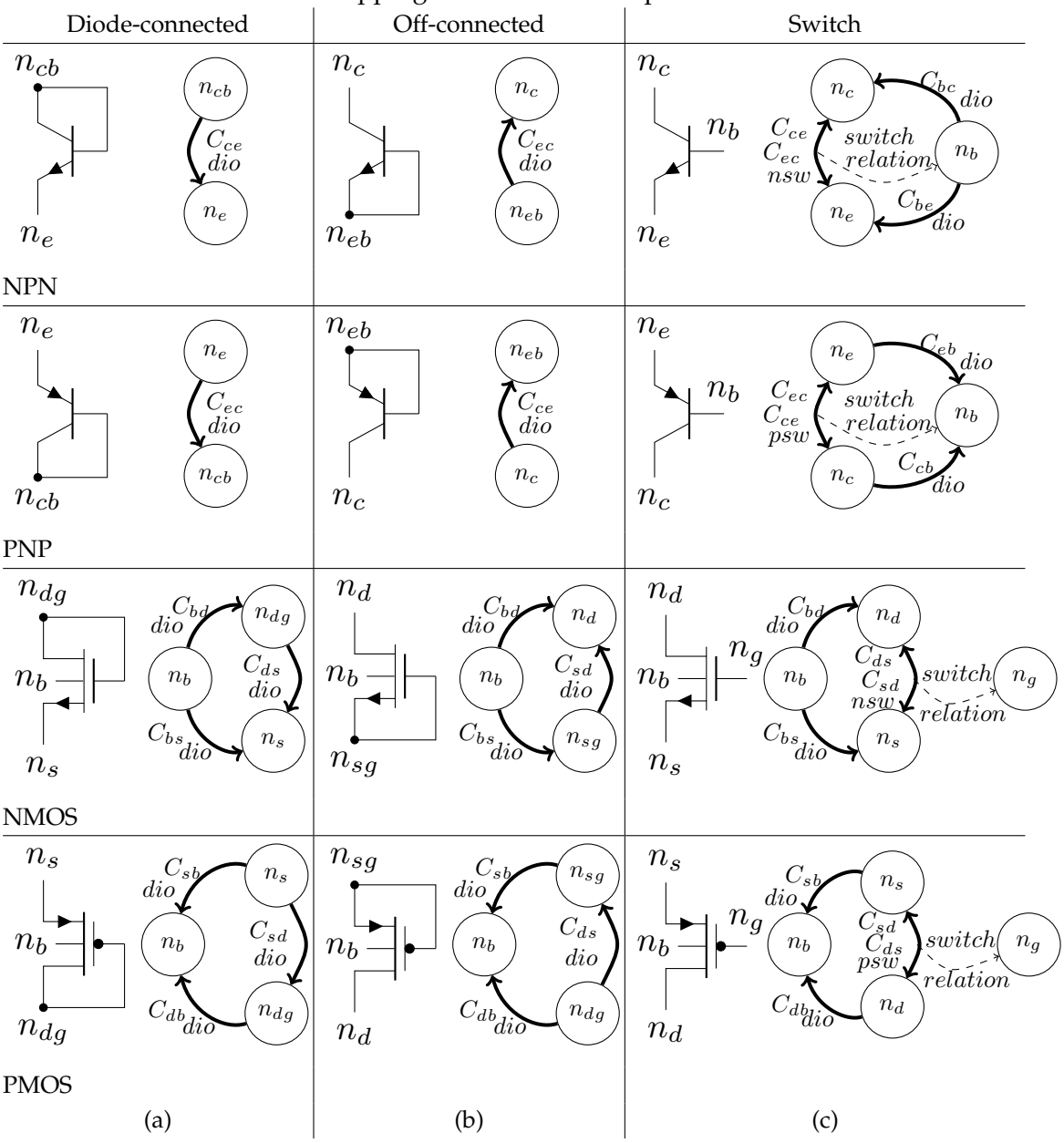
- **Off-Connected**: In this type of configuration the direction of the connections are reversed to diode-connected. For NMOS or *npn* one connection from source/gate to drain or emitter/base to collector is created. For PMOS or *pnp* one connection from drain to source/gate or from collector to emitter/base is created. In both cases the type of connection is *diode-type*, as shown in Table 3.2(b).

- **Switch**: In switch configuration, two connections are generated. These are represented by a bidirectional arrow in Table 3.2(c). The type of the connection will depend on the type of transistor, which means if it's a NMOS or *npn* transistor, it will be a *n-switch* connection. If it is PMOS or *pnp* transistor, then it's a *p-switch* connection. The dotted arrows in Table 3.2(c) represent the switch relation. As we can observe the switch connections change depending if they are connected to the gate or to the base.

For MOS technology the parasites bulk-drain and bulk-source diode is considered. In this case two graph branches are generated and the direction of these connections depends on whether it's a NMOS or PMOS. The branches start in bulk when it's a NMOS transistor and finishes in bulk when it's a PMOS transistor.

When bipolar transistors are in *switch* configuration two branches are generated, which model the base-emitter and base-collector diodes. As in MOS technology the direction of these connections depends if the type of transistor *npn/pnp*. The connections start in the base when it's a *npn* or end in the base when it's a *pnp*.

Table 3.2: Mapping of Transistor Component Devices



3.1.2 State Mapping

Before the state of connections is defined, it is necessary to appraise the node voltage levels. For each node the following three voltage levels are to be considered:

- vdd: when the estimated value in the node is equal to positive supply voltage.
- vss: when the estimated value in the node is equal to negative supply voltage.
- $floating$: when it is impossible estimate the value of node.

In order to ease the comprehension, different colors have been attributed to the different voltage levels. Red represents the vdd voltage level. The node that is expected to be vss is yellow and the blue represents the *floating* voltage level. This is shown in Table 3.3.

Table 3.3: Node Voltage Levels

vdd	*floating*	*vss*
node	*node*	*node*

If the type of connection and node voltage levels are known, then it's possible to ascertain the state of the connection. As previously cited there are three possible states [ZG12].

- ***conducting***: when the state of connection is conducting.
- ***non-conducting***: when the state of connection is not conducting.
- ***unknown***: when it is impossible define the state of connection.

Two versions of state mapping were introduced in voltage propagation analysis.

The *Liberal* version as previously published [ZG12] and the *Conservative* version, which consists of a new approach. The difference between these two versions resides in the switching relation. In *Liberal* only the gate/base voltage level is pondered. In *Conservative* the gate/base and source/emitter voltage levels are weighted.

Figure 3.2 shows the *Liberal* version, where, for *diode-type* connections the state is always considered to be *conducting*.

For *switch* type connections the state depends on the gate (switch relation) voltage level. The connection is considered *non-conducting*, when the type of the connection is *n-switch* and the gate (switch relation) voltage level is *vss* or when the type of the connection is *p-switch* and the gate (switch relation) voltage level is *vdd*.

If the gate voltage level is *floating* then the state of the connection is considered *unknown*.

The connection state is *conducting*, when the connection type is *n-switch* and the gate voltage level is *vdd* or when the connection type is *p-switch* and the gate voltage level is *vss*.

The nodes represented in white may have any of the previously defined voltage levels i.e. *vdd*, *vss* or *floating*. While for the bipolar transistors, the same reasoning can be applied in both versions.

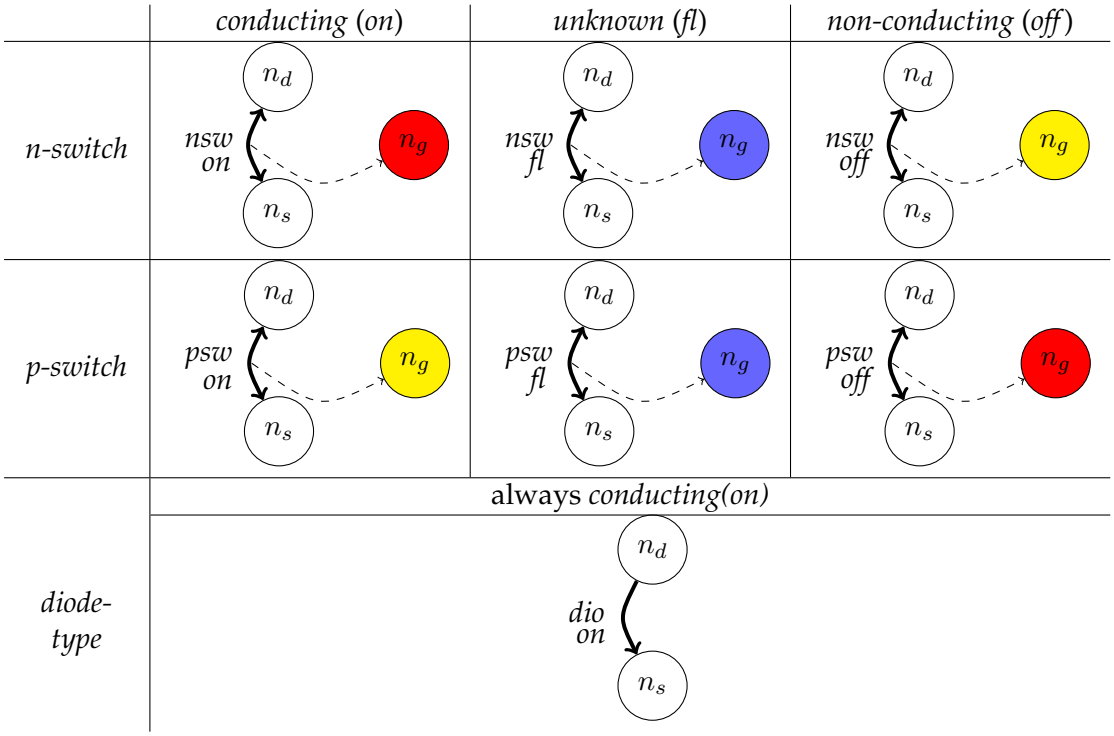


Figure 3.2: Liberal State Mapping for CMOS Transistors

The *Conservative* version was elaborated during the development of this program, aiming at obtaining a more accurate estimation of node voltage levels and state of connections. In this version the difference occur in the *switch* configuration represented in figure 3.3 which is controlled by two switch relations.

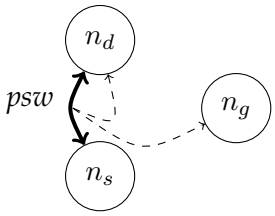


Figure 3.3: Conservative differences

Table 3.4 shows the main differences introduced in the *Conservative* version for bipolar transistors.

For *switch* type connections, the state is considered *conducting* when the type of the connection is *n-switch*, the gate voltage level is *vdd* and the source voltage level is *vss*.

When the type of the connection is *p-switch*, the gate voltage level is *vss* and the source voltage level is *vdd*. This means that there is a positive *source-gate* voltage.

For *n-switch* or *p-switch* types, if the source voltage level is *floating* then the state of the connection is considered as *unknown*.

If the source and the gate voltage levels are equal, then the connection has a *non-conducting* state.

Table 3.4: Conservative State Mapping for CMOS Transistors

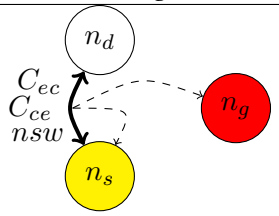
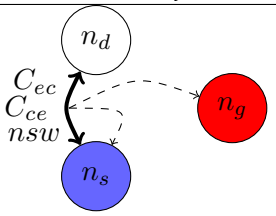
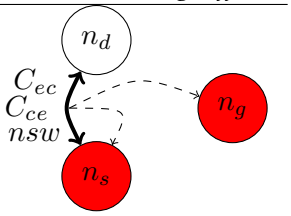
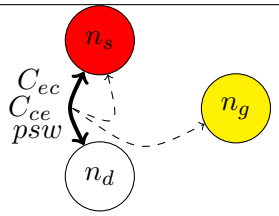
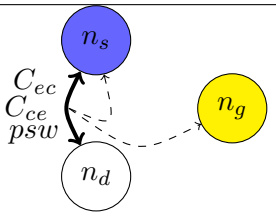
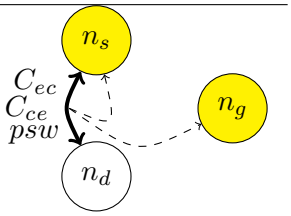
	conducting (*on*)	*unknown* (*fl*)	*non-conducting* (*off*)
$n-switch$			
$p-switch$			

Table 3.5 shows all possible cases in $Conservative$ version for bipolar and MOS transistors.

Table 3.5: Conservative State Mapping

source emitter voltage level	vss	vdd	fl	vss	fl	vdd	vss	fl	vdd
gate base voltage level	vss	vdd	fl	fl	vss	fl	vdd	vdd	vss
n-switch	off	off	fl	fl	off	fl	fl	on	off
p-switch	off	off	fl	fl	fl	fl	off	off	on
diode-type	always $conducting(on)$								

3.1.3 Voltage Propagation Algorithm

After having introduced the different states, types of connections and the voltage levels for the nodes, the voltage propagation algorithm is Figure 3.4 can be formulated.

```
1:  initialize all nodes with floating voltage level.
2:  initialize known nodes with v_ss / v_dd voltage levels
3:  repeat
4:      for all connections in graph do
5:          if it is possible propagate & not propagate before then
6:              save the connection in connection − propagate list
7:              propagate vss voltage level or propagate vdd voltage level
8:          else
9:              next connection
10:         end if
11:     end for
12: until there is no more propagation
```

Figure 3.4: Pseudo-Code for Voltage Propagation Algorithm

This algorithm starts with all nodes on *floating* voltage level (line 1). Additionally, all known voltages are initialized (e.g. the supply nodes or the power-down nodes) with *vss* or *vdd* respectively (line 2).

When the voltage propagation algorithm is running, the voltage is propagated along the connections. In order to optimize the processing speed, the connections are saved in the *connections-propagate* list.

Two main rules for propagation were considered (line 7).

- Propagate vdd voltage level in direction of the connection, if the connection has $conducting$ state (see Figure 3.5).

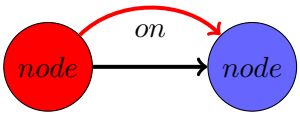


Figure 3.5: Propagation of vdd voltage level

- Propagate vss voltage level in counter direction of the connection, if the connection has $conducting$ state (see Figure 3.6).

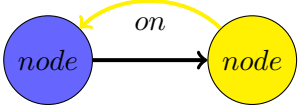


Figure 3.6: Propagation of vss voltage level

In each iteration, for all connections in graphical representation (line 4), it is checked if the propagation is possible and if the connection is not in *connections-propagate* list (line 5). If condition are fulfilled, then propagation vdd or vss takes place according to the previous rules and save this connection in *connections-propagate* list (line 6).

Due to the fact that connections can change their state during propagation, this cycle is repeated until there are no more connections to propagate through (line 12).

3.1.4 Working Example

The voltage propagation analysis is illustrated with the circuit represented in Figure 3.7. It has three transistors (M_1, M_2 and M_3) and four nodes(n_1, n_2, N_{vdd} and N_{vss}).

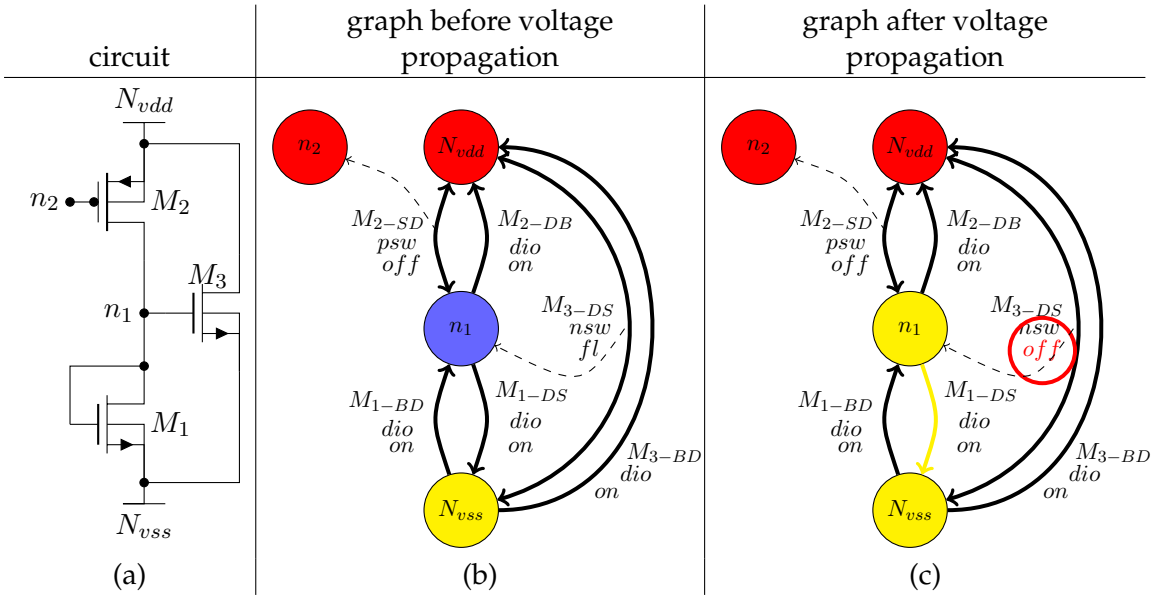


Figure 3.7: Example of Voltage Propagation Analysis

As previously mentioned in section 3.1.1, M_1 is in diode-configuration. The source and bulk pins are connected to N_{vss} node and the drain pin is connected to n_1. Two *diode-type* connections are generated one from drain/gate to source and other from bulk to drain/gate. The bulk to source connection is not taken under consideration since the pins are connected to the same node.

As M_2 transistor is in a switch configuration, the source and bulk pins are connected to N_{vdd} node, drain pin is connected to n_1 node and the gate pin is connected to n_2 node.

As previously stated (section 3.1.1), this transistor is a *p-switch* type. Three connections are considered as well as the switch relation. One connection from N_{vdd} to n_1 and one connection from n_1 to N_{vdd}, as represented in Figure 3.7(b) with a bidirectional arrow. One *diode-type* connection from n_1 to N_{vdd} is also created. The switch relation is connected to n_2 node and it was represented with a dotted arrow.

Similar connections are created for M_3. In this situation the connection has *n-switch* type and bulk is connected to N_{vss}. The switch relation is connected to n_1 node and it's

also represented by a dotted arrow.

In accordance to the voltage propagation algorithm (algorithm 3.4) and after having a graph representation, it is necessary to define all the nodes with *floating* voltage level. It was also considered that n_2 was initialized with *vdd* voltage level. Therefore, the n_2 and N_{vdd} are shown in red, n_1 in blue and N_{vss} in yellow in Figure 3.7(b).

The M_2 switch connection has *non-conducting* state, because the gate has *vdd* voltage level and the M_3 switch connection has an *unknown* state, due to the fact that the gate (switch relation) has a *floating* voltage level.

The remaining connections have *conducting* state, due to their *diode-type* nature.

According to rules in voltage propagation algorithm (pseudo-code in Figure 3.4), vss can propagate in M_1 diode-type connection (yellow connection in Figure 3.1.4(c)). Therefore, n_1 changed its voltage level to *vss*. This change in the voltage level resulted in the M_3 switch connection changing its state to *non-conducting*, as is shown in figure 3.1.4(c) (red circle).

3.2 Short Circuit Analysis

The short circuit analysis is responsible for finding potential short-circuit paths or short-circuit paths when the circuit is in power-down mode. The block diagram in Figure 3.8 shows the several steps compressing this analysis.

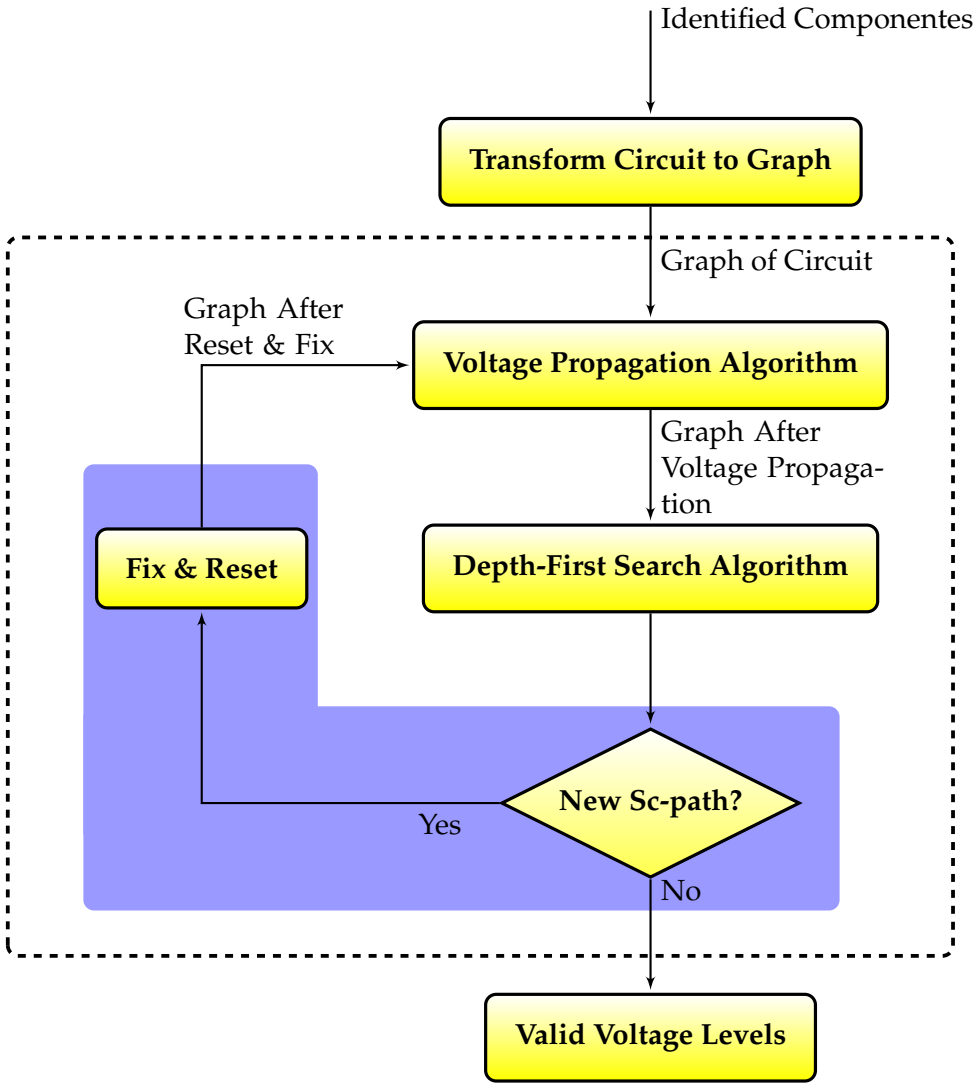


Figure 3.8: Block Diagram of Short Circuit Analysis

The dashed rectangle in the diagram represents the short circuit algorithm described in section 3.2.3, this including the $VoltagePropagationAlgorithm$, $Depth-First\ Search$ and the $FixReset$ algorithms.

The $Depth-FirstSearch$ algorithm described in section 3.2.1 runs on the graph representation after the voltage propagation analysis.

If no short circuit paths and no potential short circuit paths are found, then all currents are safely turned off. Thus, the node voltage calculated during voltage propagation analysis is valid.

If there are short circuit paths or potential short circuit paths, the result of voltage propagation analysis is not valid and in such a case an additional step is performed. The additional step is represented on Figure 3.8 with a blue box which contain the $FixReset$ block described in section 3.2.2.

For this analysis short-circuit paths were considered if a path connects a positive supply node (i.e., a node N_{vdd}) to a negative supply node (i.e., a node N_{vss}) and for all connections in the path the state is $conducting$. Figure 3.9(a) displays an example of a short-circuit path with red connections.

There is a potential short circuit path, when there are paths that connect a positive supply node(i.e., a node N_{vdd}) to a negative supply node (i.e., a node N_{vss}) and the path contains connection with *conducting* and *unknown* state. Figure 3.9(b) shows an example of potential short circuit path with a yellow connection.

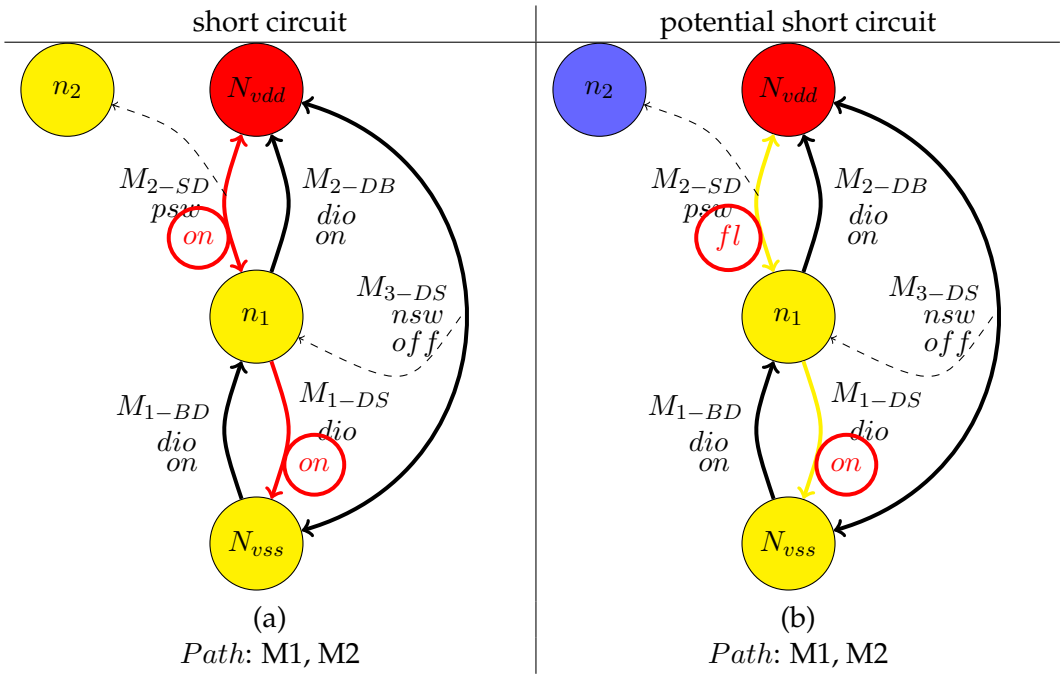


Figure 3.9: Example of Short-Circuit Path and Potential Short Circuit Path

3.2.1 Depth-First Search Algorithm

In order to detect the paths in figure 3.9, a recursive depth-first search algorithm was implemented. Figure 3.10 shows how it works. The search starts from the first node and explores left half of the graph. The node is checked, if it is equal to target, then the search ends, if not the search moves to the child of the current node. If the node is a leaf, then the search tracks-back to an unexplored node.

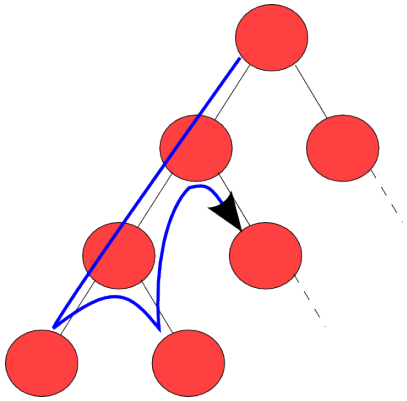


Figure 3.10: Depth-First Search

During the search part of the analysis, it is necessary to save:

- A list containing all connections found in potential short-circuit or short-circuit. This information is kept in the *path* list.
- A list of the paths discovered in graph representation. This information is kept in the *remember-path* list.
- A list containing all nodes visited during the path. This information is kept in the *nodes-visited* list.

The depth-first search algorithm (pseudo-code in Figure 3.11) starts by checking all nodes, that are originally defined as having a *vdd* voltage level (line 1). The recursive function *go to next node* is called (line 2) for each node.

1: **for all** nodes initialized with *vdd* voltage level **do**
2: *go To Next Node (node)*
3: **end for**
4: *Check Connections(remember-path)*

Figure 3.11: Pseudo-Code for $Depth - First\ Search$ algorithm

The pseudo-code for the function *Go-To Next Node* represented in Figure 3.13 and checks whether the node exists in the *nodes-visited* list (line 2). If the condition returns *visited*, then the function ends or returns to the previous node and the last connection is removed from the *path* list (line 19).

```
1: function Go-To Next Node(node)
2:     if the node not in visited − node list then
3:         save node in visited − node list
4:         if the node not initialized with vss voltage level then
5:             for all out connections in node do
6:                 if the state of the connection is fl or on then
7:                     save the connection in path list
8:                     go to next node (node)
9:                 else
10:                    next connection
11:                end if
12:            end for
13:            return one node & remove the last node of node − visited list
14:        else
15:            save path list in remember − path list
16:            return one node & remove last connection of path list & the last node of
    node − visited list
17:        end if
18:    else
19:        return one node & remove last connection of path list
20:    end if
21: end function
```

Figure 3.12: Pseudo-Code for $Go\text{-}To\ Next\ Node$ Function

If the condition returns *not visited* the node is saved in *nodes-visited* list (line 3) and it it gets checked in order to assess if it was initially defined as having a *vss* voltage level (line 4). If the condition returns true then one potential short circuit or short circuit path is found. This is the reason why it is necessary to save the *path* list in *remember-path* list (line 15). If the condition returns false all connections that start on that node will be checked (line 5).

If the connection is defined with a *conducting* or *unknown* (line 6), it is saved in the $path$ list (line 7). The *go To Next Node* function is called and initiates a recursive cycle. If there are no more connections the function ends or returns to the previous node. Finally it is also necessary to remove the node from the *nodes-visited* list (line 13).

The recursive algorithm runs until all connections are traversed and all paths found in *remember-path* are returned.

In order to identify if the paths in the *remember-path* list are potential short-circuit or short-circuit an additional function needs to be run. The corresponding pseudo-code is shown in Figure 3.13

This iterates over all found paths. For each path a check is performed in order to

verify if all connections have *conducting* state (line 3). If the condition is fulfilled, then the path is a short-circuit (line 4), otherwise a potential short-circuit is discovered (line 6).

```
1: function Check Connections(remember − path)
2:     if all connection in path have conducting state then
3:         the path is a short circuit
4:     else
5:         the path is a potential short circuit
6:     end if
7: end function
```

Figure 3.13: Pseudo-Code for *Check Connections* function

3.2.2 Fix and Reset

According to diagram block in Figure 3.8, if no short-circuit or potential short-circuit path is discovered, the result of the voltage propagation analysis is valid and all node voltage levels have been correctly estimated.

If the depth-first search algorithm detects one or more potential short circuit or short circuit paths, then the voltage propagation analysis is not guaranteed to be valid.

The current in path will create a Vds drop in the real circuit and all nodes lying on a potential short circuit or short circuit path won't be able to be determined as being vss or vdd and therefore must be reset to $floating$ voltage level.

Consider the example shown in Figure 3.14(a). If n_2 is initialized with *vss* voltage level, then the switch connection of M_2 has *conducting* state (red circle in figure 3.14(b)).

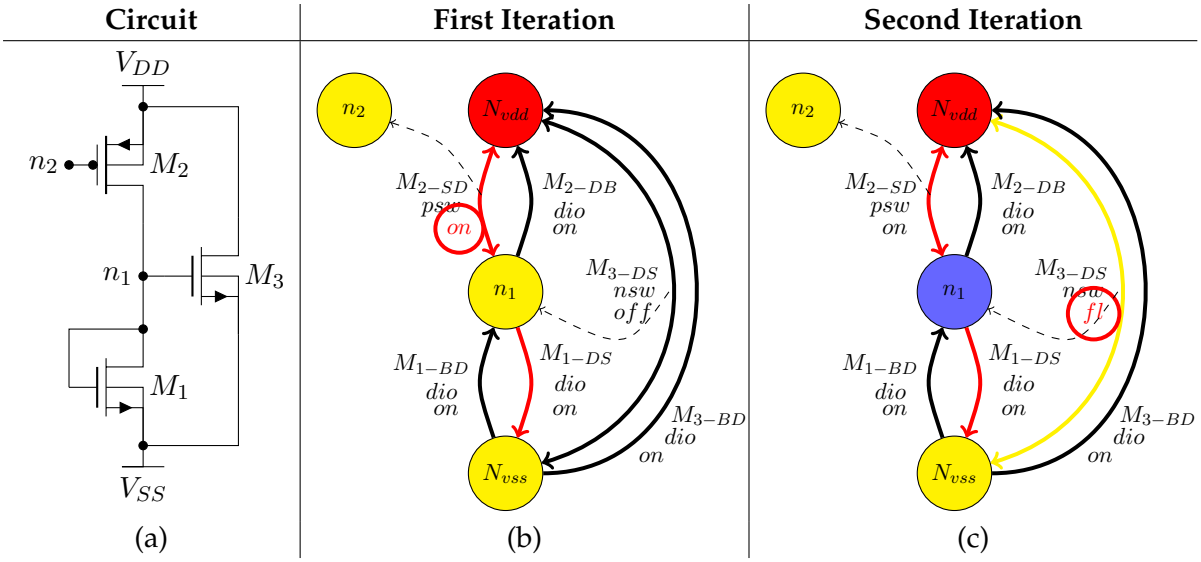


Figure 3.14: Resetting and Fixing

After the voltage propagation algorithm runs, the internal node n_1 will have a *vss* voltage level. This value however will not be correct.

Its still not clear if the n_1 node has *vss* or *vdd* voltage level. According to the rules of propagation in section 3.1.3, both switch connections (M_1 or M_2) can propagate because they have a *conducting* state.

In order to solve this problem, it is necessary to reset n_1 node to *floating* voltage level. The resetting can cause further potential short circuit paths. This is highlighted in yellow connection in Figure 3.14(c). To detect these new paths, it is necessary to run the voltage propagation algorithm again, but to avoid wrong paths, its necessary to fix the nodes in paths. The fixing can change some of the switch relations, as shown on Figure 3.14(c) (red circle). The switch connection of M_3 in the first iteration (Figure 3.14(b)) has *non-conducting* state and in the second iteration (Figure 3.14(c)) has *unknown* state.

3.2.3 Short Circuit Algorithm

After the depth-first search algorithm and the additional rule have been introduced, the short circuit algorithm can be defined. Firstly, the algorithm runs the depth-first search algorithm (line 3). It is checked if new paths were found (line 4). If the condition returns true, then a reset and fix is necessary (line 5). After this step has been completed the voltage propagation algorithm runs again (line 6). If the condition doesn't return true the algorithm ends and the result of voltage propagation analysis is valid. This cycle repeats until no new paths are found (line 8).

```
1: function checkconnections(remember − path)
2:     repeat
3:         Depth-First Search Algorithm
4:         if new paths were found then
5:             Reset and Fix the nodes in path
6:             Voltage Propagation Algorithm
7:         end if
8:     until no new paths were found
9: end function
```

Figure 3.15: Pseudo-Code for $Short\text{-}Circuit$ Algorithm

3.3 Stress Analysis

The stress analysis is responsible for finding asymmetrical stress conditions in structures. The diagram block shown in Figure 3.16 describes the tasks executed by this analysis.

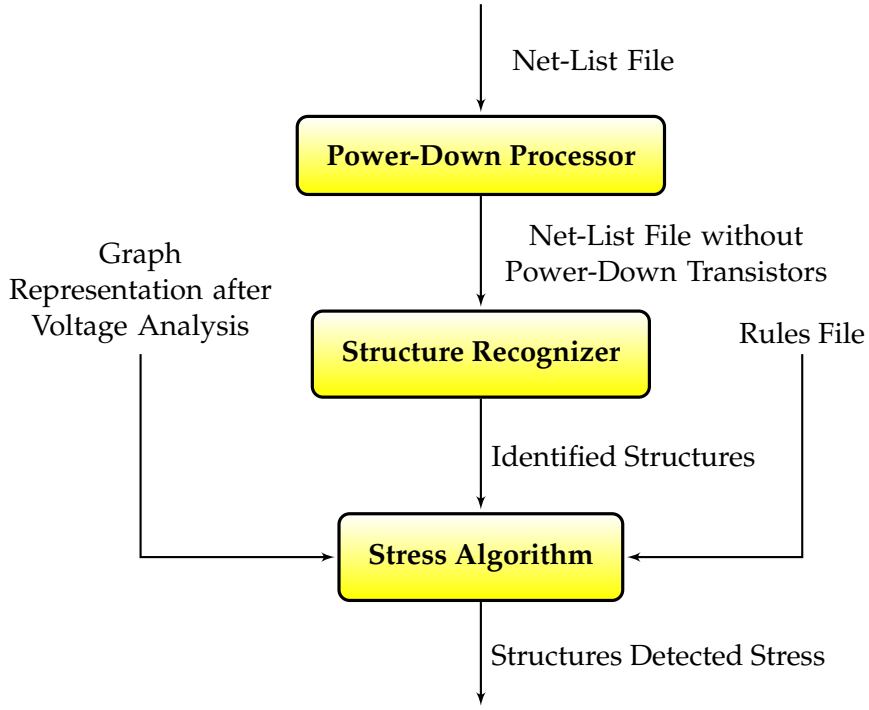


Figure 3.16: Block Diagram of Stress Analysis

The analysis starts by removing the power-down switches in the netlist file. To do this a *Power-Down Processor* is needed. Figure 3.17 shows an example illustrating the need for removing the power-down switches so that the *structure recognizer* block can identify matched structures correctly.

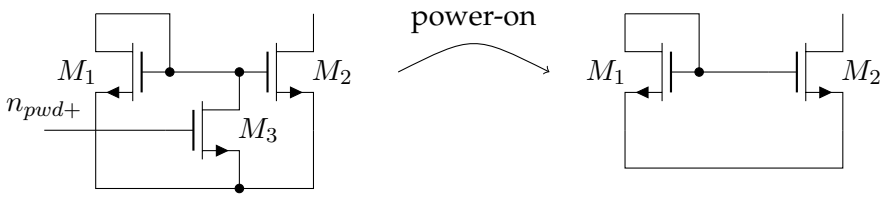


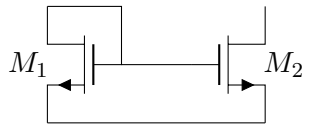
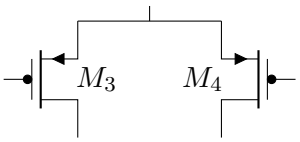
Figure 3.17: Remove Power-down Switches

The algorithm inside of the *structure recognizer* block identifies structures such as current mirrors and differential pars.

In order to detected the problems in these structures a file containing rules will be necessary.

Each rule for structures can be seen on table 3.6.

Table 3.6: Structure Rules File

Structure	Rules
	• drain of M_1 must equal the drain of M_2 • gate of M_1 must equal the gate of M_2 • source of M_1 must equal the source of M_2
	• drain of M_3 must equal the drain of M_4 • gate of M_3 must equal the gate M_4 • source of M_3 must equal the source of M_4

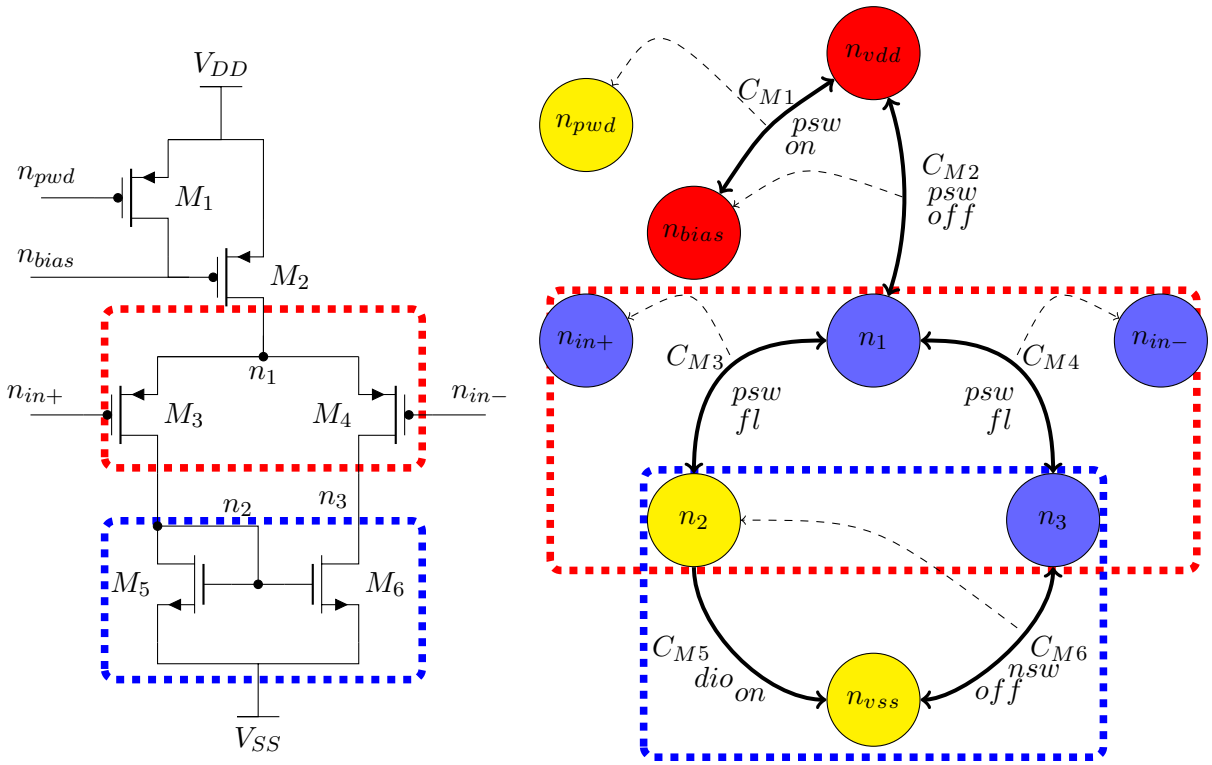
The *Stress Algorithm* block contains the stress algorithm described in section 3.3.1. This stress algorithm compares the voltage levels in each node of graph representation with the rules of structures, thus allowing to detect asymmetrical stress conditions. The analysis returns structures where asymmetrical stress exists.

In Figure 3.18(a) it can be seen that the simple current mirror with red and differential par with blue colors were successfully identified.

Figure 3.18(b) shows the graph representation after voltage propagation analysis.

In this simple current mirror, asymmetrical stress conditions were detected because there is a mismatch at the drains of M_1 and M_2 with different voltage levels in the graph representation.

The same occurs in the differential pair, showing a mismatch at the drains and gates of M_3 and M_4 because their node have different voltages levels.



(a) Schematic (b) Transient Simulation

Figure 3.18: Example of Stress Analysis

3.3.1 Stress Algorithm

The stress algorithm starts by reading the rule file for the structures (line 1). For each structure, a search is performed for each corresponding nodes (line 3).
A comparison between the defined rules and the estimated voltage levels is made. If the comparison returns as all the rules being fulfilled (line 4), then the structure does not have asymmetrical stress conditions (line 5). If one or more rules are not violated, then asymmetrical stress conditions can occur in the structure (line 7).

1: Read Rules File
2: **for all** identified structures **do**
3: find the nodes of structure in graph representation
4: **if** the rules defined for this structure are fulfilled **then**
5: No asymmetrical stress conditions are found
6: **else**
7: Asymmetrical stress condition are found
8: **end if**
9: **end for**

Figure 3.19: Pseudo-Code for *Stress* algorithm

3.4 All Program

The program was designed to ensure that each analysis can be called separately. The implementation was made in C++.

The block diagram shown in Figure 3.20 represents the implementation of the verification tool with all relevant data exchange between analysis.

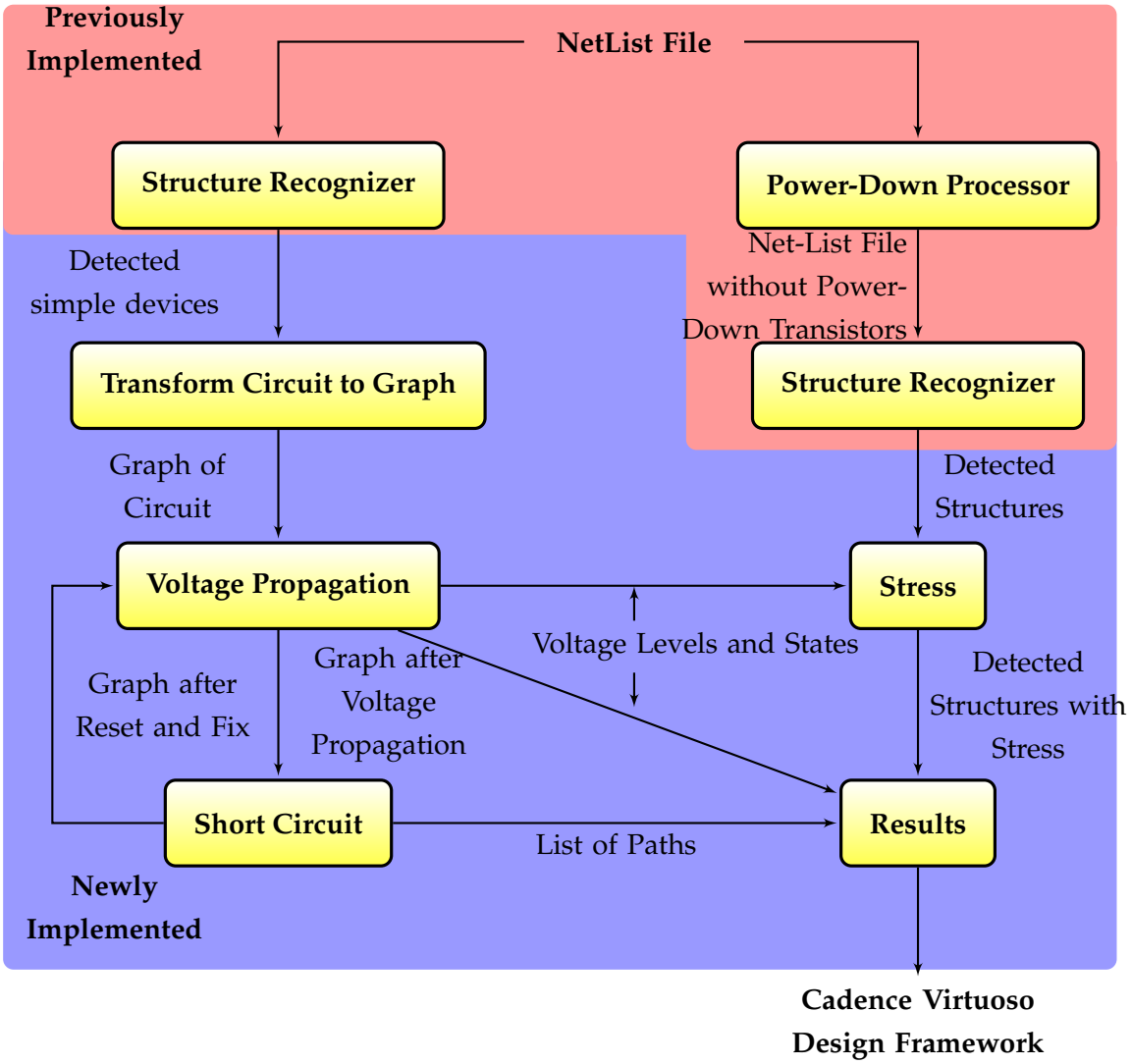


Figure 3.20: Bock Diagram of Verification of Power-Down Mode Program

The blocks with red background were previously implemented in TUM-EDA. Blue represents the newly implemented parts. The verification tool takes as input the circuit representation given in the netlist file. In the left part of the diagram, the *Structure Recognizer* block receives the netlist and detects the simple devices present in it, i.e. MOS and bipolar transistors, resistors. These are used by the *Transform Circuit to Graph* block were a graph representation described in section is created. The *Voltage Propagation* block

runs the voltage propagation algorithm described in section 3.1.3. As result, a graph representation after voltage propagation is obtained. This is used by the *Short-Circuit* block. As previously mentioned in section 3.2.2, if no paths are found then the voltage propagation results are valid. If there are paths, then the *Short-Circuit* block returns the graph representation after resetting and fixing of nodes (see section 3.2.2). The voltage propagation block receives this graph and runs the voltage propagation algorithm (Algorithm 3.4) again.

In the right part of the diagram block, power-down transistors are removed by the *Power-Down Processor* block (see section 3.3) from the net-list file. Thus, structures correctly identified by the *Structure Recognizer* block can be used in *Stress* block. It receives the returned voltage levels from the *Voltage Propagation* block and return the structures where stress conditions can occur.

The *Results* block contains all possible results. It includes the expected node voltage levels, the expected transistor state, paths returned by short-circuit analysis and the structures with stress conditions. The *Results* block was prepared to be integrated into CADENCE enviroment.

3.5 Summary

In this chapter the constituent analysis modules for the verification tool were presented.

These analysis includes the voltage propagation, the short-circuit and the stress analysis. For each of them a block diagrams containing the operation method and examples with graph representation were shown.

In the voltage propagation analysis, the corresponding graph representation as well the algorithm needed to identify the voltage along the connections were described.

After this analysis, it is possible guarantee the estimate node voltage levels with the short-circuit analysis.

In addition the graph representation were used for find asymmetrical stress conditions in the stress analysis.

4

Results

The results obtained with the verification tool in three circuits equipped with power-down mode are presented and in this chapter.

In section 4.1, the circuit is a two-stage amplifier in buffer configuration. For this circuit all results of the analysis are shown with graphical representations.

To ensure that the program works in MOS and bipolar technologies, section 4.2 presents the results for a BiCMOS circuit.

The verification tool was tested on a few industrial circuits. Section 4.3 shows the results of one of these tests.

Sections 4.1-4.3 demonstrate the results obtained from the comparison between the voltage propagation analysis and DC analysis obtained from simulation.

Finally, section 4.4 shows the integration of the verification tool and user interface for CADENCE software.

4.1 Two-stage Amplifier

The first working example is shown in Figure 4.1, where the circuit is equipped with three power-down switches M_1, M_2, M_3 and has 10 components and 9 nodes.

The backgate connections are not drawn in the circuit, but for NMOS devices they are connected to n_{vss} and for PMOS devices they are connected to n_{vdd}.

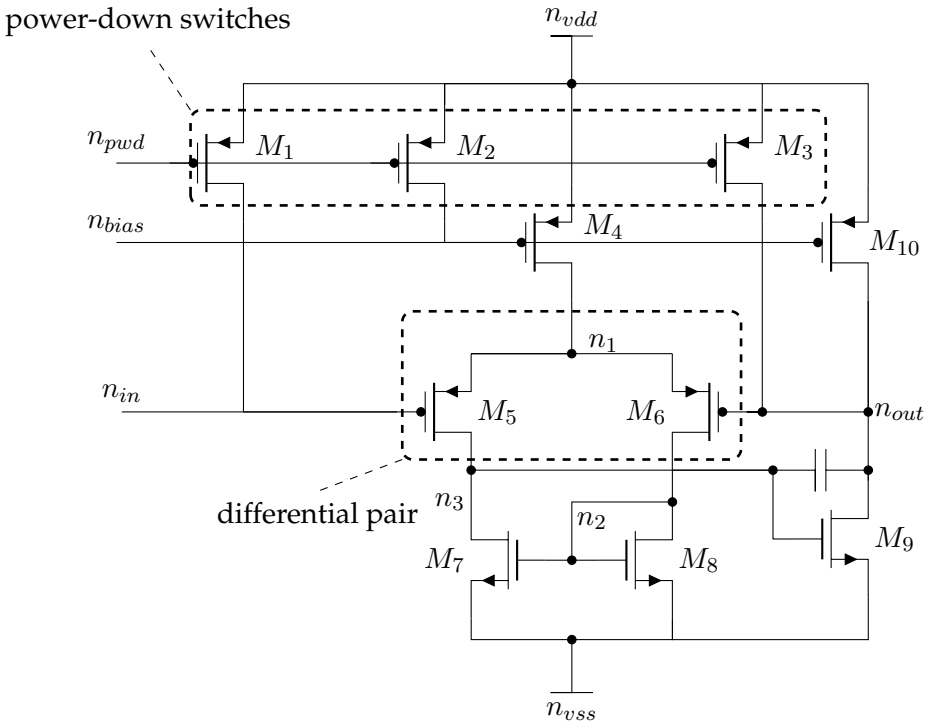


Figure 4.1: Two-Stage Amplifier Schematic

The graph representation of the circuit before voltage propagation analysis is shown in Figure 4.2. Since the corresponding bulk to source and bulk to drain connections never propagate they can be left out of the graph representations.

In order to put the circuit in power-down mode, the n_{pwd} node needs to be defined with vss voltage level. Thus, the n_{vdd} has vdd voltage level are represented in red on Figure 4.2. The nodes n_{vss} and n_{pwd} have vss voltage level and are represented in yellow. The other nodes have $floating$ voltage level and are represented in blue.

M_8 is in diode configuration, which means its in a *conducting* state. Switches M_1, M_2 and M_3 connections have a $conducting$ state because their gates have a vss voltage level. These are represented in Figure 4.2 with red circles. The remaining connections have $unknown$ state before the start of the voltage propagation algorithm because their gates have a *floating* voltage level.

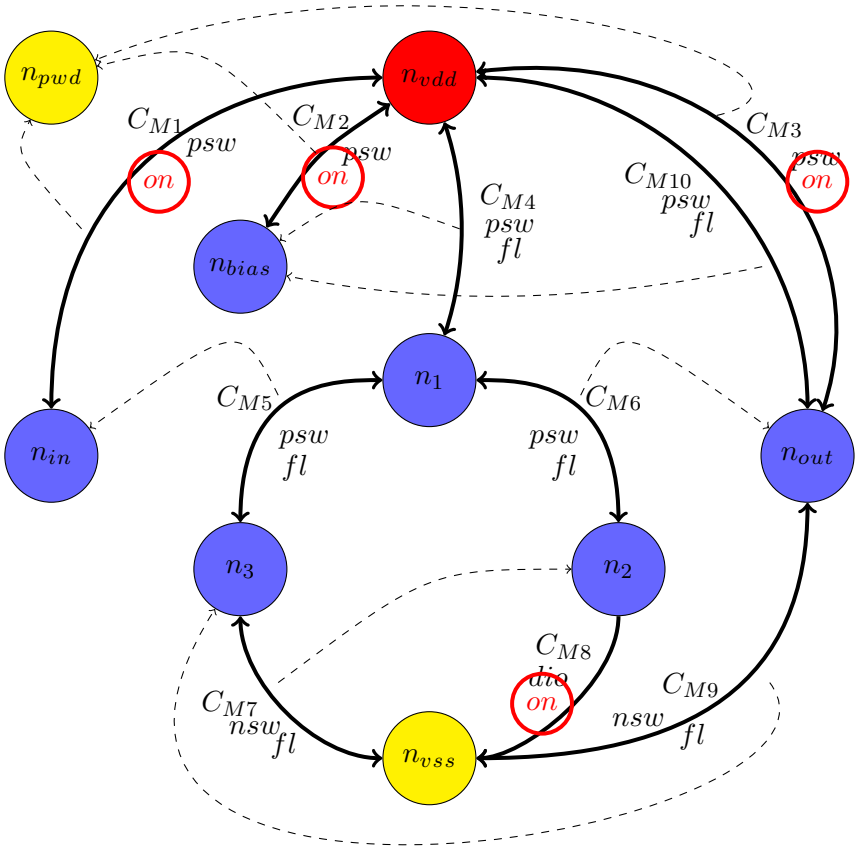


Figure 4.2: Graph Representation of Two-Stage Amplifier before Voltage Propagation Analysis

Figure 4.3 shows the results after the voltage propagation has run. As a result a few voltage levels have to propagate and according to state mapping in section 3.1.2 some connections change their state. This is shown in Figure 4.3 with red circles.

Node n_2 has vss voltage level, while n_{out}, n_{in} and n_{bias} have vdd voltage level.

The switches M_4, M_5, M_6, M_7 and M_{10} connections have *non-conducting* state, while M_9 switch connection has an *unknown* state.

In order to validate the voltage propagation results, a short-circuit analysis needs to be run in order to detect potential short-circuit and/or short-circuit paths.

In power-down mode a potential short circuit may occur with path M_3 and M_9. Yellow connections in Figure 4.3 show the path in graph representation. The M_3 connection has a *conducting* state because its gate has a *vss* voltage level and the state of M_9 connection is *unknown* because its gate voltage level is *floating*.

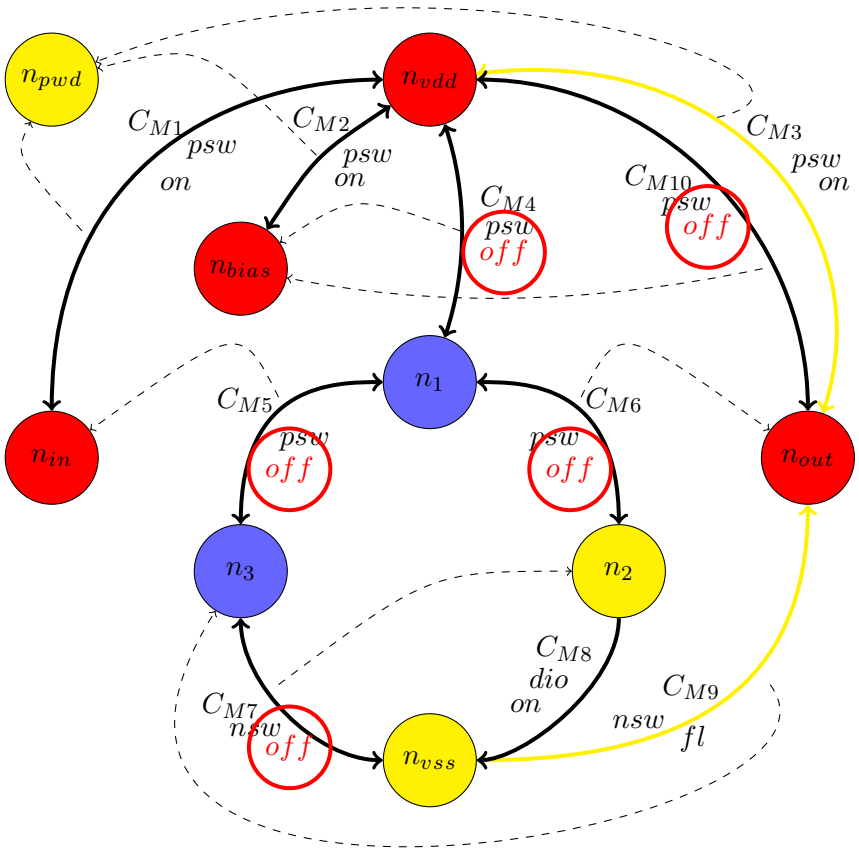


Figure 4.3: Graph Representation of Two-Stage Amplifier after Voltage Propagation Analysis

Then the node voltage are reevaluated according to the procedure described in section 3.2.2. Because of this, n_{out} node was fixed with a *floating* voltage level and the voltage propagation is run again in order return a new graph representation.

The figure 4.4 shows the results after the second run of voltage propagation algorithm. The n_{out} node changes its voltage level to *floating*, the state of M_6 connection changed to *unknown*, but no more potential short circuits were found, thus the voltage propagation estimated in nodes is valid.

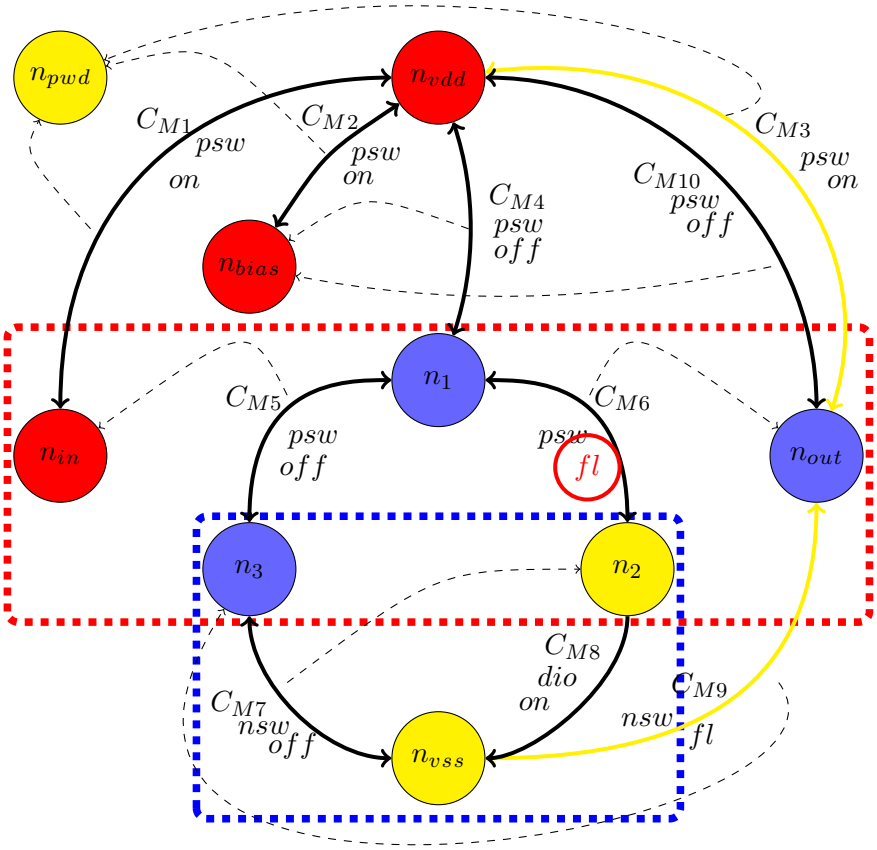


Figure 4.4: Graph Representation of Two-Stage Amplifier after Short Circuit Analysis

The stress analysis identified two structures and they are represented in figure 4.4 with dashed rectangles.

According to the rules of these two structures, in the simple current mirror asymmetrical stress conditions were detected, since there is a mismatch at the drains of M_7 and M_8 with different voltage levels. Similarly, in the differential pair the drains and gates of M_5 and M_6 have different voltage levels and asymmetrical stress conditions can occur.

The results of the verification tool are shown in Figure 4.5. Each transistor contains the estimated state returned by the program, the red line represents the potential short circuit path. The detected floating nodes are shown in blue. The dashed rectangle represents the structures with stress problems.

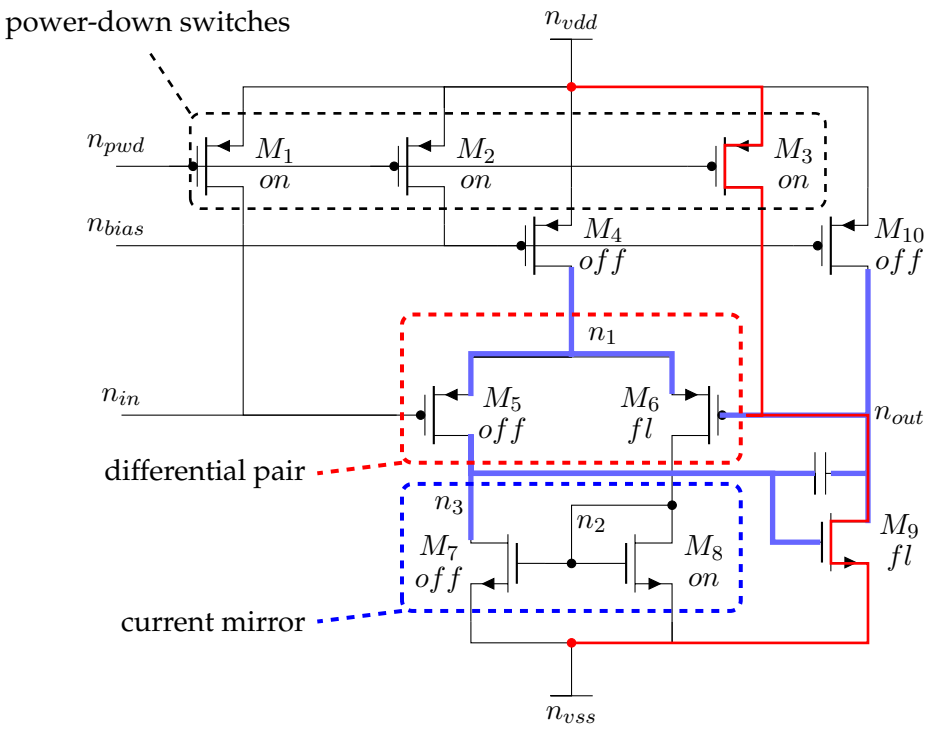


Figure 4.5: Two-Stage Amplifier with Verification Tool

In order to check if the node estimated values are valid, a comparison with the DC analysis of the circuit in power-down was made. *Liberal* and *Conservative* versions of state mapping were used (see section 3.1.2) and both produced the same results. The circuit was implemented in .32 μm technology and using a 3.3 volt supply. Table 4.1 shows the expected *vss* and *vdd* nodes. The DC analysis shows a maximum relative error is 0.2%.

$$Error(\%) = \begin{cases} 100 * (V_{dc}/V_{dd}) & \text{if voltage level is } vss \\ 100 * (V_{dd} - V_{dc})/V_{dd} & \text{if voltage level is } vdd \end{cases}$$

Table 4.1: Nodes expected with *vss* and *vdd* Voltage Level of Two-Stage Amplifier

Node Name	n_{vdd}	n_{bias}	n_{in}	n_{vss}	n_2	n_{pwd}
Voltage Propagation	*vdd*	*vdd*	*vdd*	*vss*	*vss*	*vss*
DC Analysis [V]	3.3	3.3	3.3	0	0.006	0
Error [%]	0	0	0	0	0.2	0

The table 4.2 shows the expected states of transistors and the relevant comparison with the DC analysis. For the purpose of this comparison it was assumed that if the V_{gs} over V_{th} ratio is higher than 1, then the state of transistor is *conducting*. If the ratio is lower

than 1, then the state of transistor in *non-conducting*.

$$State = \begin{cases} V_{gs}/V_{th} > 1 & conducting \\ V_{gs}/V_{th} < 1 & non-conducting \end{cases}$$

As result M_1, M_2 and M_3 transistors after the voltage propagation algorithm were expected to be *conducting*. This is verified by the DC analysis because their V_{gs} voltage drop is 8 times higher than the V_{th} voltage. The *non-conducting* transistors are also correct because V_{gs} over V_{th} is zero.

Table 4.2: Expected State of Transistors of Two Stage Amplifier

State	***conducting***			***non-conducting***			
Transistor	M_1	M_2	M_3	M_4	M_5	M_6	M_{10}
Vgs/Vth	8.05	8.05	8.05	0.00	0.00	0.00	0.00

4.2 BiCMOS

Figure 4.6 represents a BiCMOS circuit combining CMOS and bipolar technology on the same circuit. It contains 7 bipolar, 5 MOS transistors and 9 nodes. In the source 2.1 volt in supply was used. Table 4.3 shows the current consumed of this circuit in normal operation (power on).

In order to prove that the verification tool works with bipolar transistors one power-down switch was added (M_5 in Figure 4.6). This turns off the current through M_3 and M_4, but the current can flow in the red paths in Figure 4.6.

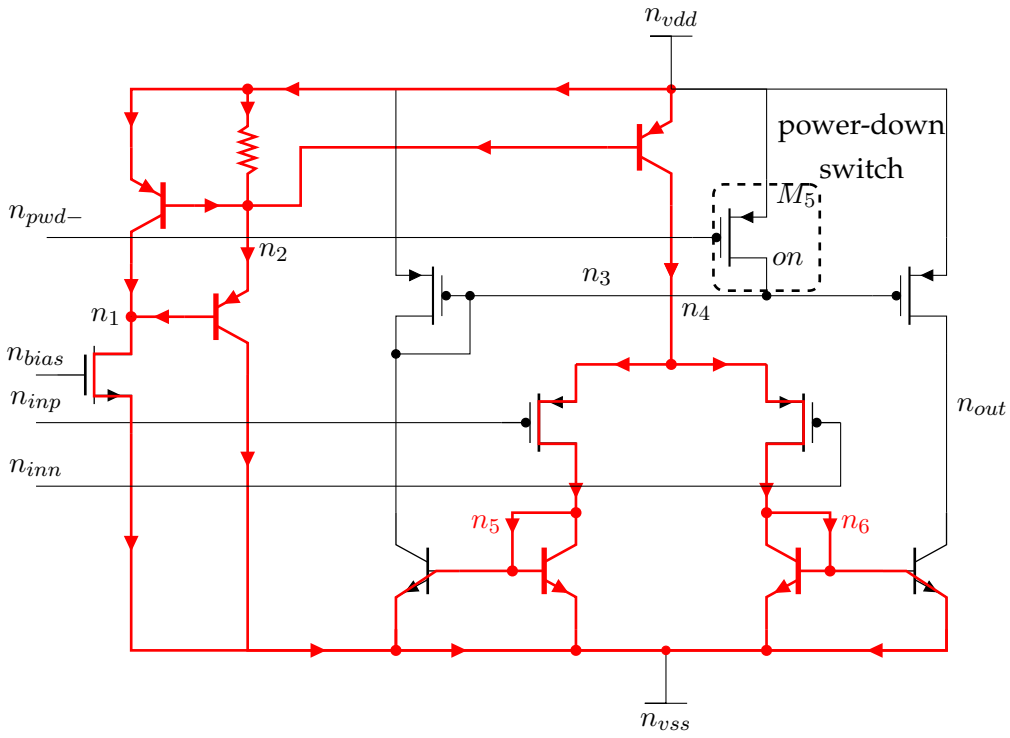


Figure 4.6: BiCMOS Circuit With One Power-Down Switch

Table 4.3: Current Consumption (DC)

	Figure 4.6	Figure 4.8
power-on	61,5 μA	61,5 μA
power-off	47,4 μA	61,5 pA

The graph representation in Figure 4.7 shows that all these paths were successfully identified by the verification tool. As a result all currents are not reliably turned off and the estimated node voltages have errors. The DC analysis shows that the current values changed with the M_5 power-down switch (table 4.3 power-off).

Most of these potential short circuit occurs because the n_{bias} node has $floating$ voltage level. This controls the state of M_6 switch connection which is $unknown$.

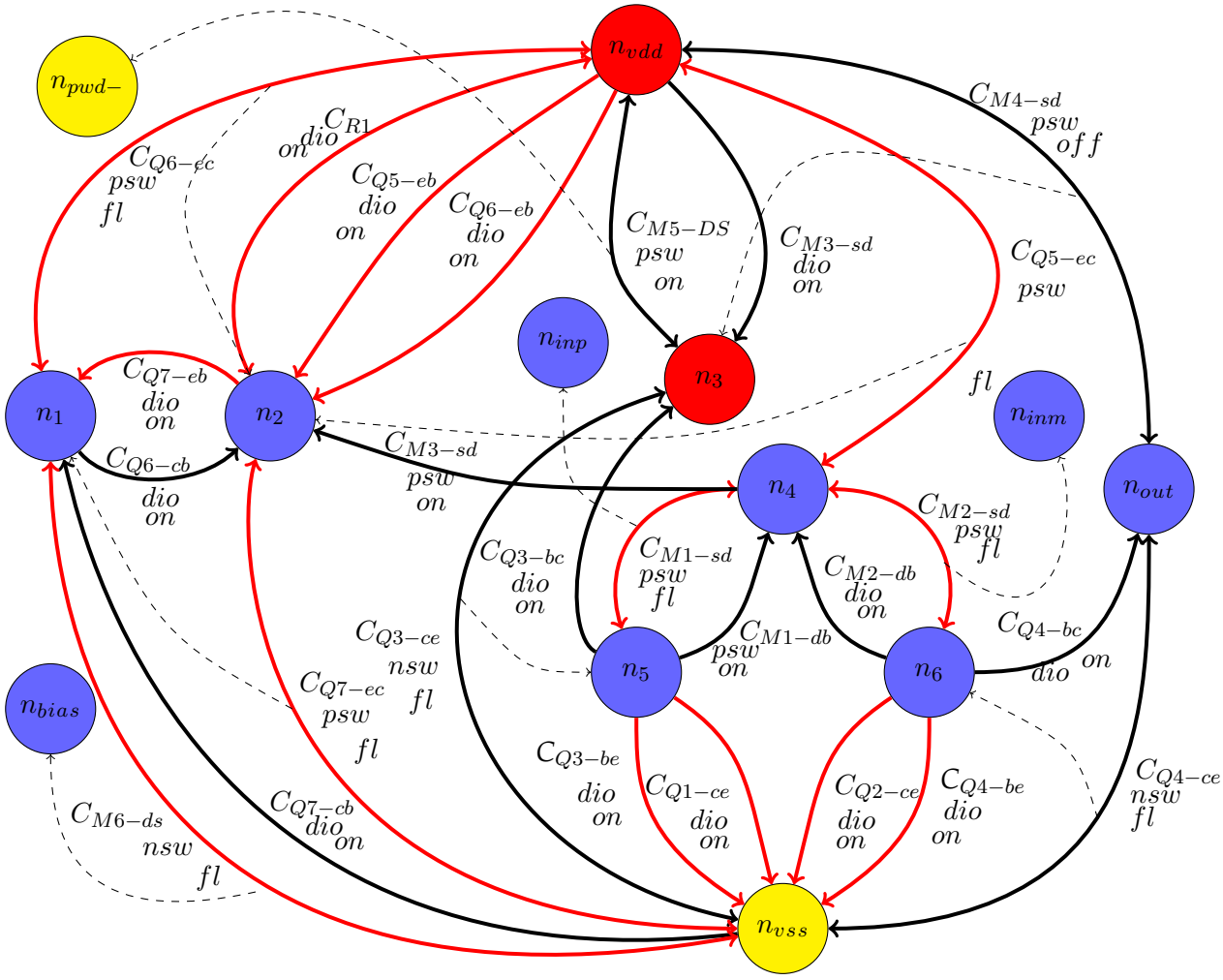


Figure 4.7: Graph Representation of BiCMOS Circuit with One Power-Down Switch After Verification Tool

To overcome this situation one more power-down switch was added onto the BiCMOS circuit (M_7 in Figure 4.8). The circuit now contains two power-down switches indicated by dotted black rectangles. For the new transistor two connections from n_{bias} to n_{vss} are created (yellow bidirectional arrow in Figure 4.9), these connections are controlled by the n_{pwd+} node which should be assigned vdd voltage level for power-down.

For this circuit, no potential short-circuit or short-circuit path is found by the verification tool and the DC analysis shows that the current changed approximately to zero (power-off table 4.3). This can be seen in the graph representation in Figure 4.9. It is possible conclude that all currents are safely turned off and the calculated node voltages are accurate.

The transistors state as well as the detected floating nodes (blue line) returned by the verification tool are shown in Figure 4.8.

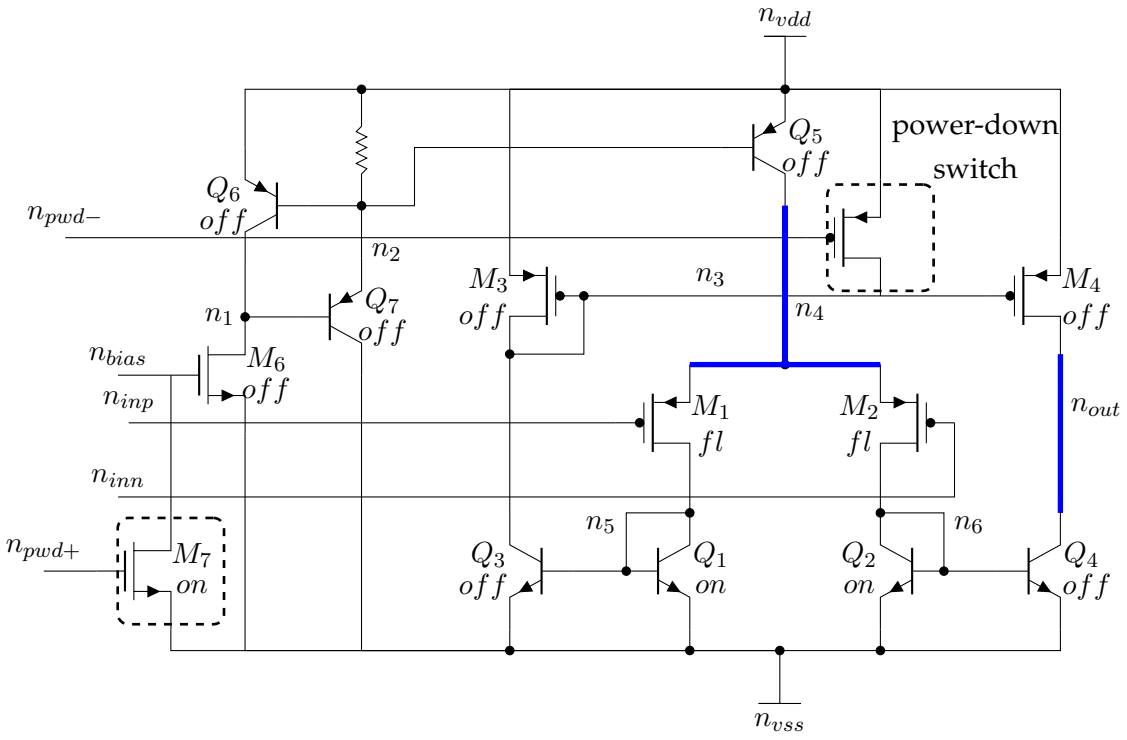


Figure 4.8: BiCMOS Circuit With Two Power-Down Switches

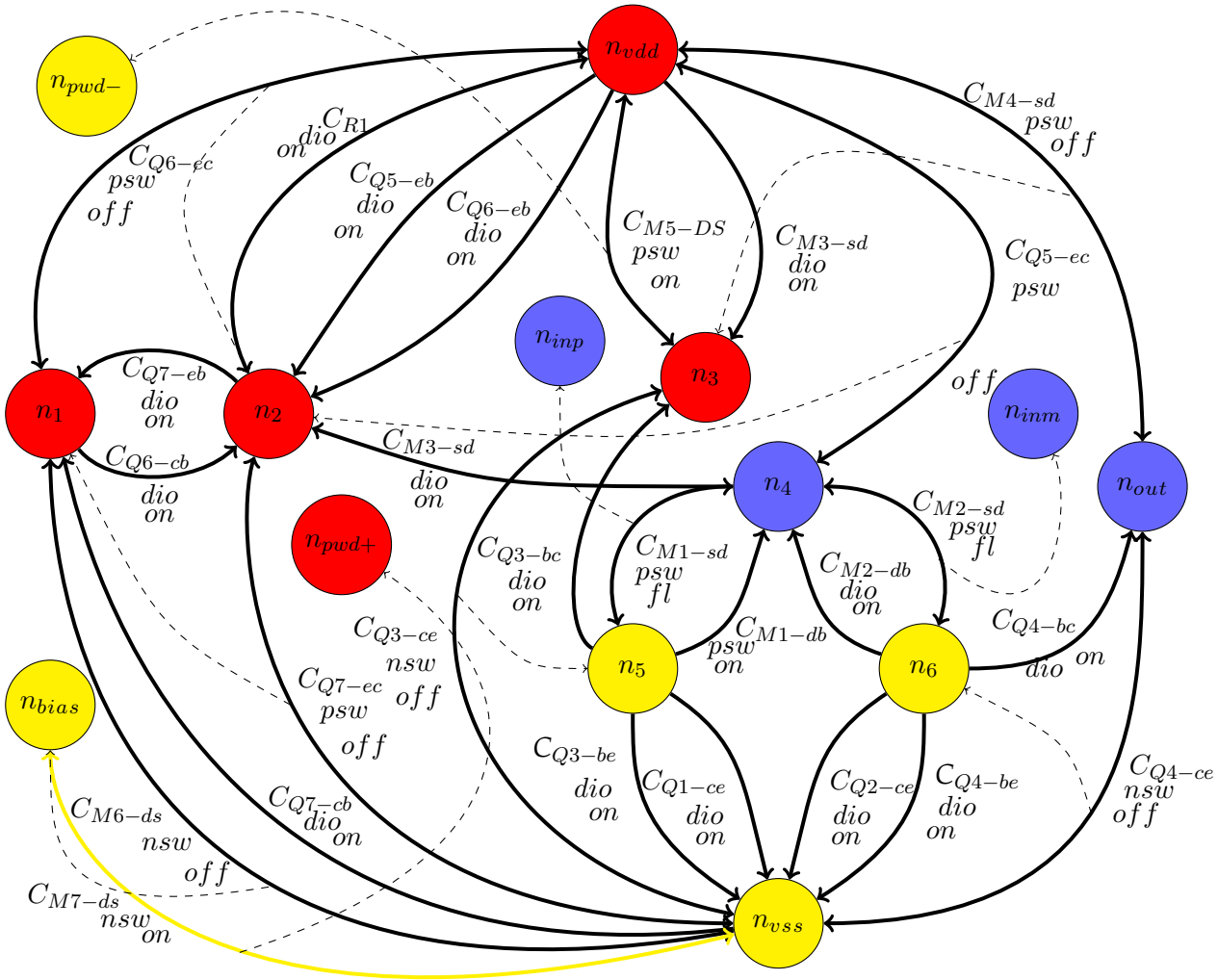


Figure 4.9: Graph Representation of BiCMOS Circuit with Two Power-Down Switches After Verification Tool

Table 4.4 shows the comparison of the node voltages generated by the verification tool and the DC analysis. For BiCMOS circuit it is possible conclude that the maximum relative error is 15,7%.

These results happen since the sizing of the transistors are not considered for the verification tool.

Table 4.4: Expected Voltage Levels in BiCMOS Circuit

Node	n_{vdd}	n_1	n_2	n_3	n_{bias}	n_{vss}	n_5	n_6
Verification Tool	vdd	vdd	vdd	vdd	vdd	vss	vss	vss
DC Analysis [V]	2.1	1.766	2.1	2.1	2.1	0	0.231	0.231
Error [%]	0	15.71	0	0	0	0	11	11

4.3 Industrial Circuit

The verification tool was tested in OTA amplifier. Figure shown the schematic of the circuit which contains 32 transistors, 2 resistors and 25 nodes.

For this circuit the verification tool runs in less then 10 ms.

The OTA amplifier was tested $Liberal$ and $Conservative$ versions of state mapping defined in section 3.1.2.

In power-down mode, the circuit has floating nodes but no potential short circuit or short circuit paths were found by the verification tool. It can be concluded that all currents are safely turned off and the calculated node voltages are valid.

The comparison between DC analysis and the voltage levels returned by the verification tool are represented in tables 4.5 and 4.6.

Table 4.5 shows the expected nodes with vdd voltage level. Equivalent results with $Liberal$ and $Conservative$ versions were obtained and the DC analysis show an error equal to 0%.

Table 4.5: Nodes with Expected vdd Voltage Level in Industrial Circuit

Node Name	n_{vdd}	n_{pwd+}	n_6
DC Analysis [V]	3.3	3.3	3.3
Error [%]	0.00	0.00	0.00

Table 4.6 shows the nodes with expected vss voltage level. Different results were obtained with $Liberal$ and $Conservative$ versions. Three different estimated voltage levels were found. The n_0, n_1 and n_2 nodes were expected to be vss in $Liberal$ version and these two changed to $floating$ in $Conservative$ version. In comparison with the DC analysis the $Liberal$ version shows a maximum relative error is 52.42 % whereas the $Conservative$ version shows a maximum relative error of 2.63%.

Table 4.6: Nodes with Expected vss Voltage Level in Industrial Circuit

Node Name	n_{vss}	n_{pwd-}	n_0	n_1	n_2	n_3	n_4	n_{16}	n_{out}
DC Analysis [V]	0.00	0.00	1.73	1.27	0.72	0.00	0.08	0.00	0.00
$Liberal$**-version**	vss								
Error [%]	0.00	0.00	52.42	38.36	21.70	0.00	2.63	0.00	0.00
$Conservative$**-version**	vss		fl			vss			
Error [%]	0.00	0.00	-	-	-	0.00	2.63	0.00	0.00

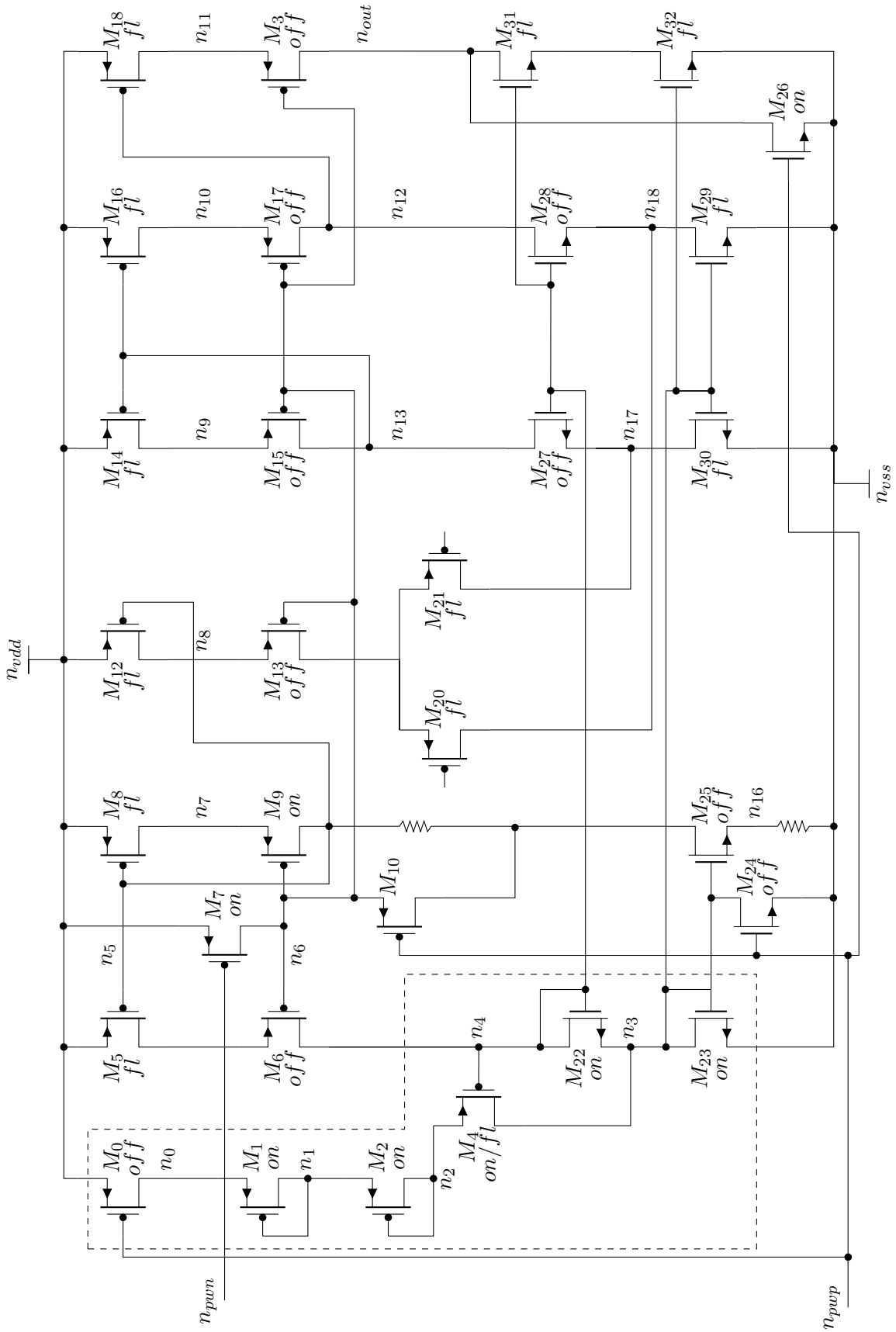


Figure 4.10: OTA Circuit

The dashed rectangle circuit in figure 4.10 represent the path where these difference results were obtained while the explanation for these appears in figure 4.11. According to the state mapping in section 3.1.2, the *Liberal* version propagated the vss voltage because the M_4 switch connection had *conducting* state, in *Conservative* version an *unknown* state is considered for this connection. Thus, vss can not propagate.

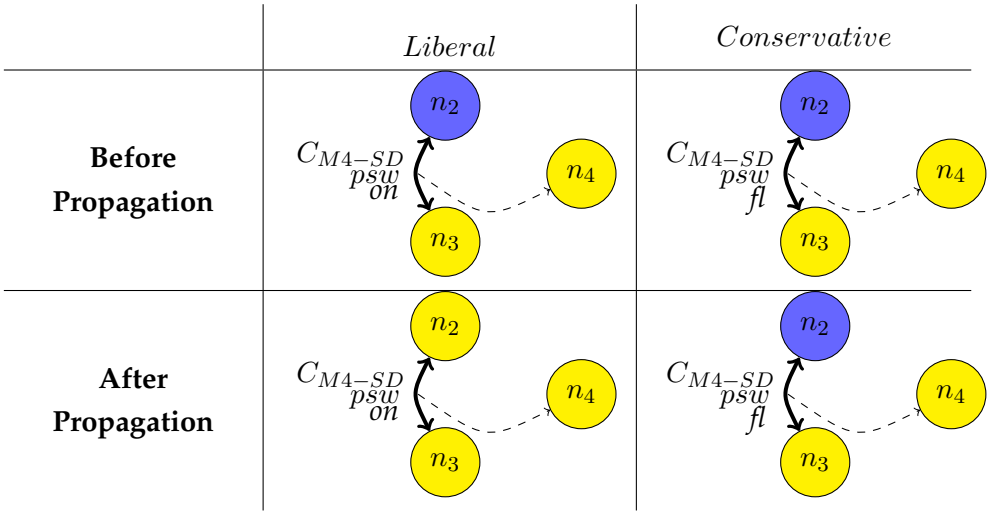


Figure 4.11: Differences in Industrial Circuit

The transistors states were also compared. The *non-conducting* transistors are shown in Table 4.7, *Liberal* and *Conservative* versions are showing equivalent numbers. After the DC analysis all V_{gs} over V_{th} ratio are close to zero.

Table 4.7: Expected *non-conducting* Transistors in Industrial Circuit With *Liberal* and *Conservative* versions

Transistor Name	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
Liberal&*Conservative*-version	*non-conducting*									
V_{gs}/V_{th}	0.00	0.00	0.00	0.00	0.00	0.15	0.28	0.35	0.35	0.35

Table 4.8 shows the expected *conducting* transistors with *Liberal* and *Conservative* versions. Comparing the two, three different states were found. M_{13}, M_{14} and M_{15} were expected conducting in *Liberal* version and on the *Conservative* version they changed their state to *unknown*. The DC analysis shows that these three transistors have V_{gs} over V_{th} less then one.

Table 4.8: Expected *conducting* Transistors in Industrial Circuit with *Liberal* Version

Transistor Name	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}	M_{17}	M_{18}	M_{19}	M_{20}	M_{21}
Liberal-**version**	all transistor expected to be *conducting*										
V_{gs}/V_{th}	0.33	0.074	0.89	7.02	7.02	7.02	7.02	8.05	8.05	8.05	8.05
Conservative-**version**	*unknown*			*conducting*							

4.4 Integration with Cadence Software

The verification tool provides only text-based output presented in a console.

In order to simplify the circuit-designers life, a graphical user interface was integrated with CADENCE.

This interface was developed in parallel with this work in TUM-EDA [Lei12][LEI]. The interface allows the end-user to easily configure all the analysis and show the results in an user-friendly way. The results are shown in a table-view and by highlighting them in the schematics editor.

Joining the two programs (the verification tool and the interface), when the designers are developing a circuit they can quickly detect floating nodes, potential short circuits or short circuits and asymmetrical stress conditions in power-down mode. The designer can also resolve some of these situations these by making some changes.

Figure 4.12 shows the results returned by the verification tool of two stage amplifier in CADENCE software. The potential short circuit path found is highlighted with yellow. The expected voltage levels have different colors, the vss voltage is represented with blue, the vdd voltage with purple and the expected $floating$ nodes are orange.

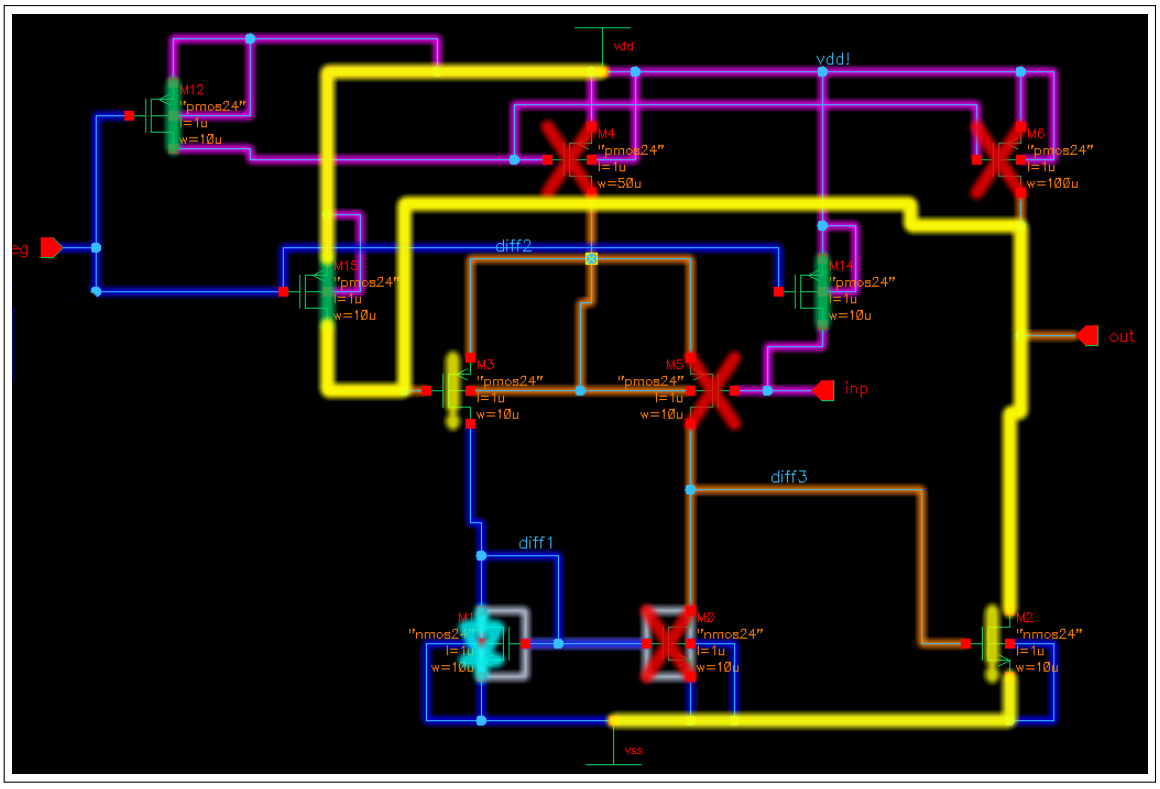


Figure 4.12: Two Stage Amplifier in Cadence With Verification Tool and Interface

In figure 4.13 it is possible see the transistor state returned by the verification tool. The interface draws symbols for each state and also draws a diode component for transistors in diode configuration and triangle for transistors in off configuration.

In summary they are represented as:

- *conducting*: predefined as a green wire connecting the drain/collector to source/emitter
- *diode-connected*: predefined as an diode in the correct polarity between the drain/-collector and the source/emitter
- *non-conducting*: predefined with a red cross on the top of the transistor, symbolizing that it is not conducting
- *unknown*: predefined with an yellow exclamation mark
- *off-connected* : predefined with a triangle over the pins of the instance (not shown in figure).

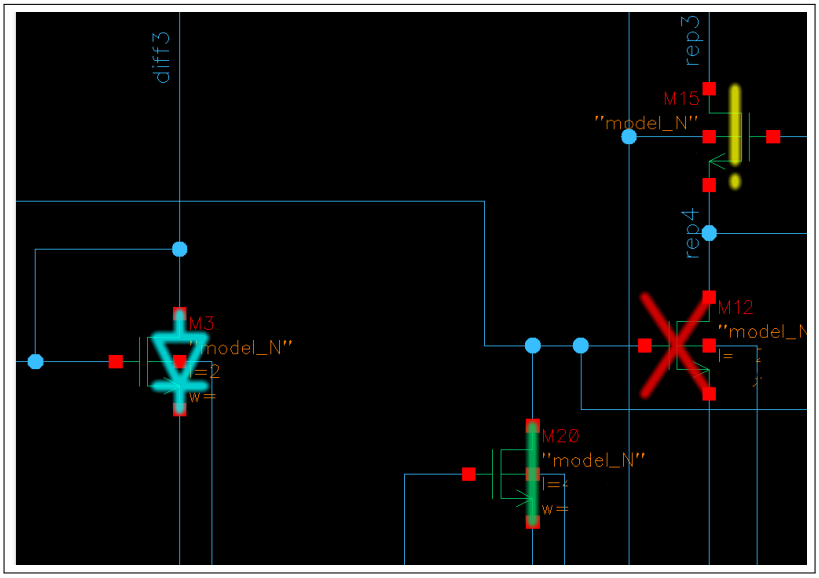


Figure 4.13: Defined States in Cadence

Figure 4.14 displays one example of asymmetrical stress conditions in current mirror detected by the verification, for each one, nodes with white color were designed by the interface.

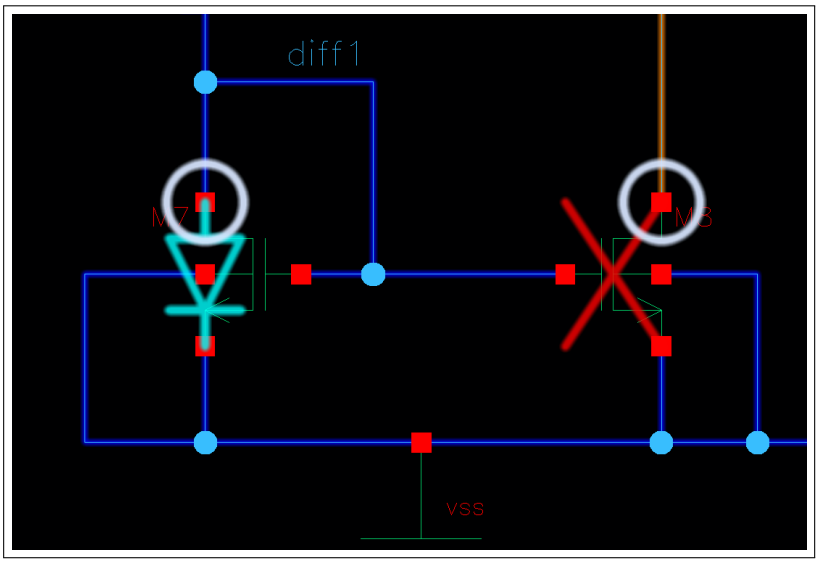


Figure 4.14: Detect Structures with Problems in Cadence

4.5 Summary

In this chapter the results obtained with the verification tool in circuits were presented. The voltage propagation analysis presents a good estimation of node voltage levels in the circuit as well as the state of the transistors. Furthermore floating nodes were detected as illustrated for three different circuits.

Besides having proven that the program works with CMOS and bipolar devices, a comparison with the DC analysis was made. While for small circuit $Liberal$ and $Conservative$ versions show the same results on estimation of the nodes voltages levels, better results were obtained with $Conservative$ version for large circuits.

The presented short-analysis shows which potential short-circuit or short-circuit was detected with good accuracy.

The asymmetrical stress conditions by the stress analysis were correctly detected.

In addition, the results of integration of the verification tool and the interface for the CADENCE were shown.

5

Conclusions

In this thesis, a program for circuits analog circuits equipped with power-down mode features was described.

In the voltage propagation analysis described in section 3.1 the circuit is transformed directly in to a graph representation which contains all connections and nodes of circuit. The algorithm used is responsible to propagate the voltage along of the connections. As results, an estimation of the node voltage levels and the state of the transistors is obtained. Furthermore floating nodes are detected as well.

In addition a new state mapping was developed (*Conservative* version). While for small circuits both (*Liberal* and *Conservative*) versions of state mapping had the same results, for larger circuits best results with the *Conservative* version on the estimation of the voltages and the state of transistors were obtained.

With the short circuit analysis, potential short circuit or short circuit paths in circuit can be found. This analysis runs the depth-first search in the graph representation where to find out if all currents are reliably turned off. If potential or short circuit paths were discovered, then the estimated voltage levels contains errors. In order to solve this problem, a short circuit algorithm was presented. This includes the depth-first search and an iterative fixing and resetting method as described in section 3.2.2. Finally, the node voltage levels and the state of transistors are correctly estimated.

The calculated voltages can be used to inspect the circuit for severe reliability problems due to stress. A fully automated checking remains as future work [ZG12]. The stress analysis described in this thesis makes this detection process automatic. In this analysis structures such as a differential pair or current mirror were successfully identified. For each structure is necessary a file containing the rules. Then the structures where the rules are not fulfilled are returned.

Tests were made in circuits with the verification tool. As results, in two-stage amplifier floating nodes, potential short circuits and asymmetrical stress condition were detected in power-down mode by the verification tool.

In the BiCMOS circuit two different power-down configurations were tested. On one of the configurations with only one power-down switch the DC analysis confirms that the current is flowing along the paths. All these paths were successfully identified by the verification tool. With two power-down switches no paths were found while the DC analysis shows that consume changed approximately to zero.

Tests performed on an industrial circuit show that, the verification tool runs in less than 10ms. Thus, it can be concluded that the program works fast and the time is negligible.

A comparison between the estimated node voltage levels and the DC analysis was made. The obtained results shows a good estimation of node voltage levels. While for CMOS the error is less then 5%, the BiCMOS present an error of less then 16%.

The CADENCE software is one of most commonly used programs in the development of circuits. Therefore, the program developed in this thesis was prepared to work with it. In parallel with this work, a interface was developed in TUM-EDA [LEI][Lei12]. With both (the verification tool and the interface), designers can quickly detect floating nodes, potential short circuits and asymmetrical stress conditions in power-down mode. These situation can then be resolved by making certain changes.

To finalize this work, a program to use in circuit equipped with power-down features was implemented. The results prove the efficacy of it but several improvements can be done.

As future work it is proposed to update this program to work with different supply voltages. Knowing that the CADENCE is one important program for the circuit designers, it is also proposed to increase the program to work with hierarchy blocks.

Bibliography

[BJ96] S. Blieck and Er. Janssens. Software Check for Power-Down Mode of Analog Circuits. In *Proceedings ESSCIRC*, pages 404–407, September 1996.

[CMFSL11] F.R. Chouard, S. More, M. Fulde, and Doris Schmitt-Landsiedel. An analog perspective on device reliability in 32nm high-k metal gate technology. In *Design and Diagnostics of Electronic Circuits Systems (DDECS)*, pages 65–70, April 2011.

[CZT+01] Yuan Chen, Jonathon Zhou, S. Tedja, Frank Hui, and A.S. Oates. Stress-induced MOSFET mismatch for analog circuits. In *Integrated Reliability Workshop Final Report, 2001. 2001 IEEE International*, pages 41–43, 2001.

[LEI] Technical report.

[Lei12] Pedro Leitao. Backannotations for the Verification of Power-Down-Mode (User Documentation). Technical report, September 2012.

[MG11] Elie Maricau and G. Gielen. Transistor aging-induced degradation of analog circuits: Impact analysis and design guidelines. In *European Solid-State Circuits Conference (ESSCIRC)*, pages 243–246, September 2011.

[MGS08] Tobias Massier, Helmut Graeb, and Ulf Schlichtmann. The Sizing Rules Method for CMOS and Bipolar Analog Integrated Circuit Synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(12):2209–2222, December 2008.

[Ohr98] Milton Ohring. *Reliability and Failure of Electronic Materials and Devices*. Academic Press, 1998.

[PG11] Xin Pan and Helmut Graeb. *Lifetime Yield Optimization of Analog Circuits Considering Process Variations and Parameter Degradations*, chapter 6, pages 131–146. InTech, February 2011.

[SBA+03] C. Schlunder, R. Brederlow, B. Ankele, A. Lill, K. Goser, and Roland Thewes. On the degradation of p-MOSFETs in analog and RF circuits under inhomogeneous negative bias temperature stress. In *12th IEEE Internationa Workshop on Cellular Nanoscale Networks and their Applications*, pages 5–10, March 2003.

[ZG12] Michael Zwerger and Helmut Graeb. Short-Circuit-Path and Floating-Node Verification of Analog Circuits in Power-Down Mode. In *Int. Conf. Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, September 2012.