

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj

Faculté des Sciences et de la technologie

Département d'Electronique

Mémoire

Présenté pour obtenir

LE DIPLOME DE MASTER

FILIERE : Electronique

Spécialité : Microélectronique

Par

- **Labachi Oussama**
- **Tourche Zakaria**

Intitulé

Etude et applications des simulateurs Gateway et SmartSpice dans le domaine de la conception des circuits intégrés

Évalué le : 27/06/2022

Par la commission d'évaluation composée de :*

<i>Nom & Prénom</i>	<i>Grade</i>	<i>Qualité</i>	<i>Etablissement</i>
<i>Mme.F.kherrat</i>	<i>MAA</i>	<i>Président</i>	<i>Univ-BBA</i>
<i>Dr. Messai Zitouni</i>	<i>MCA</i>	<i>Encadreur</i>	<i>Univ-BBA</i>
<i>Mme.F.Khaled</i>	<i>MCB</i>	<i>Examineur</i>	<i>Univ-BBA</i>
<i>Dr.Touati Zine-eddine</i>	<i>Dr</i>	<i>Co-encadrant</i>	<i>Univ-BBA</i>

Année Universitaire 2021/2022

Remerciements

Ce travail a été effectué dans le laboratoire de recherche au département
d'électronique université de Bordj Bou Arreridj

Je remercie d'abord DIEU le tout puissant. Pour l'aide, la santé, la volonté et
la patience qu'il m'a donnés.

Que tous qui, tout au long de l'élaboration de ce travail, m'ont aidés et
apporté leurs connaissances trouvent ici l'expression de ma profonde gratitude.

J'adresse mes remerciements les plus sincères à :

Monsieur Dr. Messai rapporteur, de ce mémoire qui m'a fait confiance en me
proposant ce thème ainsi que son aide et ses précieux conseils qu'il n'a cessé de
me prodiguer en toutes occasions.

Nous remercions beaucoup les membres du jury qui nous ont fait l'honneur
de participer et de juger notre mémoire.

Je tiens à remercier sincèrement Dr.Zine-eddine Touati, Co-rapporteur, de ce
mémoire pour son aide et son soutien fraternel.

Enfin, je voudrais exprimer toute ma reconnaissance à ma famille qui n'a
cessé de m'encourager et de me soutenir durant toutes les années de formation
suivie à l'université, et que tous mes amis trouvent ici mes sincères
remerciements

Dédicaces

Je dédie ce modeste travail aux deux personnes qui sont les plus chères :
A la plus belle créature sur terre, à ma source de tendance et d'amour, celle
qui s'est toujours sacrifiée pour me voir heureux.

A ma mère !

A la lumière de mes jours, la flamme de mon bonheur, source de générosité et
de patience tout au long de ma carrière scolaire.

A mon père !

Que Dieu vous protège et vous prête bonne santé et longue vie qu'il vous
garde pour moi.

A mes chers frères.

A tous mes amis

A tous ceux qui m'aiment

Résumé :

Dans cette étude, nous présentons les principales fonctions du simulateur « Gateway » ainsi que son langage « SmartSpice ». Ce dernier est un outil important à l'heure actuelle en matière de simulation de circuits analogiques et numériques.

Le Gateway comprend un éditeur de schéma, un éditeur de texte et un module d'affichage des résultats ainsi que d'autres utilitaires qui rendent le logiciel plus agréable pour l'utilisateur.

Cette étude, nous avons montré que l'outil « Gateway » permet de posséder un véritable labo de mesure et d'essai, à moindre coût, et dans un espace restreint. De plus nous sommes arrivés à mettre en évidence la concordance des résultats obtenus par simulation avec les résultats théoriques et pratiques.

A partir de la création du schéma d'un montage dont les composants sont décrits par les modèles fournis en général par les fabricants de semi-conducteurs, « Gateway » permet de simuler le fonctionnement du circuit et de l'analyser finement, tant du point de vue des grandeurs continues que dans le domaine temporel ou fréquentiel.

Mot clé : Gateway, SmartSpice, langage Spice, net liste, Simulation, CAO.

Abstract :

In this study, we present the main functions of the GateWay simulator as well as its language. The latter is an important tool today in the simulation of analog and digital circuits.

The Gateway includes a schematic editor, text editor, and result display module along with other utilities that make the software more user-friendly.

During this study, we showed that the Gateway tool makes it possible to have a real measurement and test lab, at a lower cost, and in a limited space. In addition, we have succeeded in highlighting the concordance of the results obtained by simulation with the theoretical and practical results.

From the creation of the diagram of an assembly whose components are described by the models generally provided by the manufacturers of semiconductors, GATEWAY makes it possible to simulate the operation of the circuit and to analyze it finely, both from the point of view continuous quantities than in the time or frequency domain.

Key word: Gateway, SmartSpice, Spice language, net list, Simulation, CAD.

ملخص

في هذه الدراسة ، نقدم الوظائف الرئيسية لمحاكاة GateWay بالإضافة إلى لغتها. هذا الأخير هو أداة مهمة اليوم في محاكاة الدوائر التناظرية والرقمية.

تتضمن Gateway على محرر تخطيطي ومحرر نصوص ووحدة عرض النتائج جنبًا إلى جنب مع أدوات مساعدة أخرى تجعل البرنامج أكثر سهولة في الاستخدام.

خلال هذه الدراسة ، أوضحنا أن أداة Gateway تتيح إمكانية الحصول على مختبر قياس واختبار حقيقي ، بتكلفة أقل ، وفي مساحة محدودة. بالإضافة إلى ذلك ، نجحنا في إبراز تطابق النتائج التي تم الحصول عليها عن طريق المحاكاة مع النتائج النظرية والعملية.

من خلال إنشاء الرسم التخطيطي للتجميع الذي يتم وصف مكوناته بواسطة النماذج المقدمة بشكل عام من قبل الشركات المصنعة لأشباه الموصلات ، يتيح Gateway محاكاة تشغيل الدائرة وتحليلها بدقة ، من وجهة نظر الكميات المستمرة أكثر في مجال الوقت أو مجال التردد.

الكلمة المفتاحية : GateWay ، SmartSpice ، لغة Spice ، قائمةالعناصر ، المحاكاة ، CAD

Sommaire

Introduction générale :	2
Chapitre I: Simulation électronique sous SPICE	4
I.1. Introduction	5
I.2. Histoire du code SPICE	5
I.3. Pourquoi simuler ?	6
I.3.1. But de la simulation	6
I.3.2. Les avantages de la simulation électronique	6
I.4. Logiciels de simulations	8
I.4.1. PSPICE	8
I.4.2. LTSPICE	9
I.4.3. SmartSpice	10
I.4.4. Gateway	11
I.5. Principe de fonctionnement des logiciels Spice	12
I.5.1. Saisie du schéma	12
I.5.2. Simulation	12
I.5.2.1. Exploitation de la simulation	12
I.6. Conclusion :	13
Chapitre II: Initiation aux simulateurs Gateway/SmartSpice	14
II.1. Introduction :	15
II.2. Gateway :	15
II.2.1. Définition :	15
II.2.2. Fenêtre Gateway :	16
II.2.3. Création d'un nouveau projet :	17
II.2.3.1. Espace de travail workspace :	17
II.2.3.2. Bibliothèques :	19
II.2.3.3. Création d'un fichier schématique :	20
II.2.3.4. Sauvegarde du fichier schématique :	21
II.2.4. Saisie du schéma :	21
II.2.4.1. Placer des composants :	21
II.2.4.2. Création de fils :	22
II.2.4.3. Chargement des fichiers :	23
II.2.4.4. Vérification de la conception :	24

II.2.4.5.	Netlisting et panneau de contrôle :.....	24
II.2.4.6.	Simulation :.....	27
II.2.4.7.	Boutons de la barre d'outils de simulation :.....	27
II.2.4.8.	Affichage de polarisation DC :.....	28
II.2.4.9.	Post-traitement.....	28
II.2.4.10.	Ajouter du texte :.....	29
II.2.4.11.	Filtrage de sélection :.....	29
II.3.	Le choix de simulateur.....	29
II.3.1.	État et informations de SmartSpice :.....	30
II.4.	Les Types d'analyses dans SmartSpice.....	31
II.4.1.	Analyses .OP, .DC, .AC et .TRAN :.....	31
II.4.2.	Analyse .DISTO :.....	31
II.4.2.1.	Exemple :.....	32
II.4.3.	Analyse .PZ :.....	32
II.4.3.1.	Exemple :.....	32
II.4.4.	Analyse de réseau .NET (deux ports) :.....	32
II.4.4.1.	Exemple :.....	33
II.4.5.	Analyse .TF :.....	33
II.4.6.	Analyse .NOISE :.....	34
II.4.6.1.	Exemple :.....	35
II.5.	Conclusion :.....	35
Chapitre III: Simulation des schémas électriques avec Gateway.....		37
III.1.	Introduction :.....	38
III.2.	Schéma du VCO (Oscillateur Contrôlé par Voltage).....	38
III.2.1.	Définition :.....	38
III.2.2.	Principe de fonctionnement du VCO.....	38
III.2.3.	Liste Composant du VCO :.....	39
III.2.4.	Connexion les composant.....	39
III.2.5.	Écriture du modèle de composant :.....	40
III.2.6.	Traitement et affichage des erreurs :.....	41
III.2.6.1.	Activation et désactivation des avertissements et des erreurs :.....	43
III.2.7.	Simulation :.....	44
III.3.	L'amplificateur 1GHz :.....	46
III.3.1.	Liste Composant de schéma l'amplificateur 1GHz :.....	46

III.3.2. Connexion des composants :.....	46
III.3.2.1. Modèle symbole de transistor :.....	47
III.3.2.2. Fichier de contrôle [12]:.....	47
III.3.3. Simulation :.....	48
III.4. Conclusion :	49
Conclusion générale	50
Conclusion générale :	51
Référence : 53	

Liste des figures

- Figure I.1** Fenêtre PSPICE
- Figure I.2** Résultats de simulations PSPICE
- Figure I.3** Fenêtre LTspice
- Figure I.4** Fenêtre SmartSpice.
- Figure I.5** Fenêtre de Gateway
- Figure II.1**Fenêtre Gateway
- Figure II.2**Nouvelle boîte de dialogue Espace de travail
- Figure II.3**Boîte de dialogue Paramètres de l'espace de travail
- Figure II.4**Dialogue de configuration d'espace de travail avec chemin relatif
- Figure II.5**Volet de bibliothèque
- Figure II.6**Affichage des symboles en grandes icônes
- Figure II.7**Nouveau fichier schématique
- Figure II.8**Boîte de dialogue pour enregistrer le schéma
- Figure II.9**Placementdu symbole Vpulse
- Figure II.10** Création de fils
- Figure II.11**Exemple de Gateway current_mirror. Schlr
- Figure II.12**Tableau de simulation
- Figure II.13**Volet d'erreurs
- Figure II.14**Exemple de Netlist sous SmartSpice
- Figure II.15**Fenêtre du fichier de contrôle
- Figure II.16**Barre d'outils Analyses
- Figure II.17**Dialogue d'analyse transitoire
- Figure II.18**Boîte de dialogue de la progression de la simulation
- Figure II.19**Exemple de SmartView
- Figure II.20**Boutons de polarisation DC
- Figure II.21**Exemple des tensions et courants du point de fonctionnement
- Figure II.22**Barre d'outils Cross Probe
- Figure II.23**Sélectionner la barre d'outils du filtre
- Figure II.24**Boîte de dialogue Préférences
- Figure II.25**Barre d'état du simulateur et de l'outil
- Figure II.26**Barre d'état affichant des informations SmartSpice avancées
- Figure III.1** Schéma VCO
- Figure III.2**Symbole
- Figure III.3**Boîte de dialogue Propriétés
- Figure III.4** Affichage des erreurs
- Figure III.5**Mode Fil actif lors de la correction de l'erreur
- Figure III.6**Vérifications de dessin dans les préférences
- Figure III.7** Graphiques de VCO
- Figure III.8**Liste de Properties.
- Figure III.9** Graphiques Après avoir appuyé sur Split All/Selected
- Figure III.10** Schéma de l'amplificateur 1GHz accordé
- Figure III.11** Symbole de transistors
- Figure III.12** Graphiques de l'amplificateur 1GHz

Liste des tableaux

Tableau I.1 Types de bruit

Tableau III.1 Liste des composants

Tableau III.2 Liste des composants

Introduction générale

Introduction générale :

La microélectronique comprend la production miniaturisée de fonctions électroniques de plus en plus complexes sur un seul support « system-on-chip » (SOC) et des ASIC hybrides (circuits intégrés spécifiques à une application). Initialement, l'objectif de la microélectronique était de réduire le poids et le volume des appareils, mais ces deux critères sont devenus des critères secondaires face à la fiabilité accrue et à la réduction du coût (Time_to_market) permis par l'intégration[1]. Par conséquent, les concepteurs de systèmes ont un grand besoin d'outils permettant un développement rapide à moindre coût. Cette attente doit être répondue dans le "Virtual Prototype" (PV) du système, en utilisant le Hardware Description Language (HDL), qui réduit au minimum la phase de test du prototype physique, ce qui est très coûteux en temps et en argent[2].

La famille de simulateurs SPICE apparait dans les années 60. (PSPICE, LTSPICE, SmartSpice, Gateway) fournit de nombreuses composants primitives (résistances, condensateurs, bobines, transistors MOS et bipolaires, etc.) pour décrire les topologies des circuits. Chaque primitive est associée à un modèle plus ou moins précis[3]. Certains modèles, tels que les transistors MOS, sont étroitement liés à la technologie en affichant un certain nombre de paramètres de processus. D'autres, comme les composants passifs, sont essentiellement décrits en termes d'électricité. Cependant, quel que soit le type de modèle, la taille du système à résoudre pour le simulateur sera proportionnelle au nombre de nœuds du circuit. De cela, on peut en conclure que lorsque le nombre de transistors devient plus grand (des centaines à des milliers dans les conceptions analogiques), le temps de simulation devient plus long et des problèmes de convergence peuvent survenir, rendant la simulation des circuits analogiques plus compliquée.

Dans cette étude, nous présentons les principales fonctions du simulateur Gateway de SILVACO ainsi que son langage source. Ce dernier est un outil important à l'heure actuelle en matière de simulation de circuits analogiques et numériques. Il constitue encore aujourd'hui l'outil de simulation en génie électrique.

Le Gateway comprend un éditeur de schéma, un éditeur de texte et un module d'affichage des résultats ainsi que d'autres utilitaires qui rendent le logiciel plus agréable pour l'utilisateur.

Pour une meilleure explication de l'exploitation de ce logiciel, nous avons divisé notre travail en trois chapitres.

Le premier chapitre est consacré à la présentation d'une vue générale sur la simulation Spice, les avantages de simulations et les différents logiciels de simulation Spice.

Le deuxième chapitre consiste l'initiation au Gateway dont l'objectif est de maîtriser les différentes fonctionnalités de l'outil, afin d'éditer un schéma, le simuler sous différentes formes, puis pouvoir analyser le résultat.

Et enfin, dans le troisième chapitre, nous avons modélisé deux circuits :(VCO et un amplificateur travaillant jusqu'à 1Ghz) en utilisant l'éditeur schématique « Gateway » et le simulateur Spice « SmartSpice » en décrivant les différentes étapes de simulations.

Chapitre I :

Simulation électronique

sous SPICE

I.1. Introduction

La fin du XX siècle a connu une évolution considérable des technologies de fabrication des circuits intégrés due principalement à la pression du marché des circuits électroniques de communication[4].

Le début du XXIe connaît la même tendance avec l'apparition des objets connectés, des applications liées à la mobilité, des systèmes embarqués (notamment dans le secteur de l'automobile), du traitement des données en masse ou des dispositifs médicaux. Pour les concepteurs de circuits, il faut réduire le temps de mise sur le marché tout en satisfaisant à la fois aux demandes de performances accrues (consommation, poids, tailles et coûts réduits).

Le temps de mise sur le marché du produit tant un facteur crucial de réussite, ces systèmes de grande complexité doivent être conçus, vérifiés et validés de manière sûre. Les coûts de fabrication étant très élevés, les tentatives multiples ne sauraient être tolérées.

Les acteurs du marché de l'EDA (Electronique Design Automation) fournissent aux fabricants de circuits intégrés des flots de conception, mais aussi de vérification alliant méthodologies, logiciels et matériels appareillés pour respecter au mieux les exigences précédemment citées[5].

L'un des éléments clés de ces deux flots est le simulateur électrique à niveaux d'abstractions mixtes, signaux mixtes et RF (radiofréquences). Nous le désignerons tout simplement : Simulateur AMS/RF (Analog Mixed Signal/RF).

SPICE (Simulation Program with Integrated Circuit Emphasis) est un logiciel libre de simulation généraliste de circuits électroniques analogiques. Il permet la simulation au niveau du composant (résistances, condensateurs, transistors) en utilisant différents types d'analyses :

Point de polarisation (courant continu), analyse fréquentielle pour petits signaux et bruit (courant alternatif linéaire), transitoire...etc.

Les logiciels Easyspice, Gspiceui, LTspice, Pspice ou SmartSpice (Gateway) sont des interfaces graphiques généralistes pour l'utilisation de SPICE.

I.2. Histoire du code SPICE

SPICE est un programme informatique conçu pour simuler des circuits électroniques analogiques et numériques, son objectif initial était le développement de circuits intégrés, d'où il tire son nom : Simulation Program with Integrated Circuit Emphases[6].

L'origine du code SPICE remonte à un autre programme de simulation de circuit appelé CANCER. Développé par le professeur Ronald Rohrer de l'UC Berkeley[7]avec certains de ses étudiants à la fin des années 1960, CANCER a continué à être amélioré au début des années 1970. Lorsque Rohrer a quitté Berkeley, CANCER a été réécrit et renommé SPICE, publié en tant que version 1 dans le domaine public en mai 1972. La version 2 de SPICE a été publiée en 1975 (version 2g6, une révision mineure de cette version de 1975) [8].

Une amélioration majeure est intervenue en mars 1985 avec la version 3 de SPICE (également publiée dans le domaine public). Écrite en langage C plutôt qu'en FORTRAN, la version 3 incorporait des types de transistors supplémentaires (le MESFET, par exemple) et des éléments de commutation. La version 3 permettait également l'utilisation d'étiquettes de nœud alphabétiques plutôt que de chiffres uniquement. Les instructions écrites pour la version 2 de SPICE devraient cependant toujours fonctionner dans la version 3 [2].

Plusieurs programmes similaires sont disponibles aujourd'hui, SPICE étant le plus largement distribué. Il peut être utilisé à la fois sur ordinateur central et personnel [4].

I.3. Pourquoi simuler ?

I.3.1. But de la simulation

Avec l'arrivée de l'informatique, tous les outils de simulation se sont répandus dans tous les secteurs d'activité. L'électronique ne pouvait pas être épargnée. La simulation donne au technicien la possibilité de réaliser puis de tester un prototype virtuel. Les avantages sont évidents : Economies de temps et d'argent.

Pour l'enseignant la simulation s'impose pour les raisons suivantes :

- Cette technique permet d'obtenir des résultats lorsque le matériel nécessaire manque : mesure des courants, calcul de la transformée de Fourier d'un signal...
- A partir de la simulation, les étudiants permettent de résoudre de nombreux phénomènes théoriques trop complexes, la simulation permet aussi de montrer ce que l'on n'a pas le temps d'expliquer.
- La simulation est de plus en plus utilisée dans l'industrie, il faut donc y préparer nos étudiants.

I.3.2. Les avantages de la simulation électronique

L'un des principaux avantages des simulateurs est qu'ils sont en mesure de fournir aux utilisateurs un retour d'information pratique lors de la conception de systèmes réels. Cela

permet au concepteur de déterminer l'exactitude et l'efficacité d'une conception avant la construction effective du système. Par conséquent, l'utilisateur peut explorer les avantages de conceptions alternatives sans construire réellement les systèmes.

En examinant les effets des décisions de conception spécifiques pendant la phase de conception plutôt que pendant la phase de construction, le coût global de la construction du système diminue de manière significative. A titre d'exemple, considérons la conception et la fabrication de circuits intégrés. Au cours de la phase de conception, le concepteur est confronté à une multitude de décisions concernant, par exemple, le placement des composants et le routage des câbles de connexion.

Il serait très coûteux de fabriquer toutes les conceptions potentielles afin d'évaluer leurs performances respectives. Cependant, à l'aide d'un simulateur, l'utilisateur peut étudier la supériorité relative de chaque conception sans fabriquer les circuits eux-mêmes. En simulant le comportement des conceptions, le simulateur de circuit est en mesure de fournir au concepteur des informations relatives à l'exactitude et à l'efficacité de conceptions alternatives. Après avoir soigneusement pesé les ramifications de chaque conception, le meilleur circuit peut alors être fabriqué.

Un autre avantage des simulateurs est qu'ils permettent aux concepteurs de systèmes d'étudier un problème à différents niveaux d'abstraction. En abordant un système à un niveau d'abstraction plus élevé, le concepteur est mieux à même de comprendre les comportements et les interactions de tous les composants de haut niveau au sein du système et est donc mieux armé pour contrecarrer la complexité du système global.

Cette complexité peut simplement submerger le concepteur si le problème a été abordé à un niveau inférieur. Comme le concepteur comprend mieux le fonctionnement des composants de niveau supérieur grâce à l'utilisation du simulateur, les composants de niveau inférieur peuvent ensuite être conçus et ensuite simulés à des fins de vérification et d'évaluation des performances.

L'ensemble du système peut être construit sur la base de cette technique «top-down». Cette approche est souvent appelée décomposition hiérarchique et est essentielle dans tout outil de conception et simulation prenant en charge la construction de systèmes complexes. Par exemple, en ce qui concerne les circuits, il est souvent utile de penser à un microprocesseur en termes de registres, d'unités logiques arithmétiques, de multiplexeurs et de commandes.

Les simulateurs peuvent être utilisés comme un moyen efficace d'enseigner ou de démontrer des concepts aux étudiants. Cela est particulièrement vrai pour des simulateurs qui

utilisent intelligemment l'infographie et l'animation. Ces simulations montrent de manière dynamique le comportement et les relations de tous les composants du système simulé, offrant ainsi à l'utilisateur une compréhension significative de la nature du système.

Considérons à nouveau, par exemple, un simulateur de circuit. En montrant les chemins empruntés par les signaux lorsque les entrées sont consommées par les composants et les sorties générées via leur sortie en éventail respective, l'étudiant peut réellement voir ce qui se passe dans le circuit et a donc une meilleure compréhension de la dynamique du circuit. Un tel simulateur devrait également permettre aux étudiants d'accélérer, de ralentir, d'arrêter ou même d'inverser une simulation pour faciliter la compréhension. Cela est particulièrement vrai lors de la simulation de circuits contenant des boucles de rétroaction ou d'autres opérations qui ne sont pas immédiatement intuitives lors de la conception.

I.4. Logiciels de simulations

I.4.1. PSPICE

Dans les années 70, des étudiants en informatique de l'université de Berkeley en Californie mettent au point un Algorithme destiné à la conception des circuits intégrés analogiques.

La version de SPICE vendue par Microsim devient le standard de l'industrie sous le nom de PSPICE, considéré comme un logiciel de simulation de circuits électronique. Nous révélons les mesures de tous les courants et de toutes les tensions avant même de l'avoir construit. Tout est assemblé et expérimenté sur le pc[9].

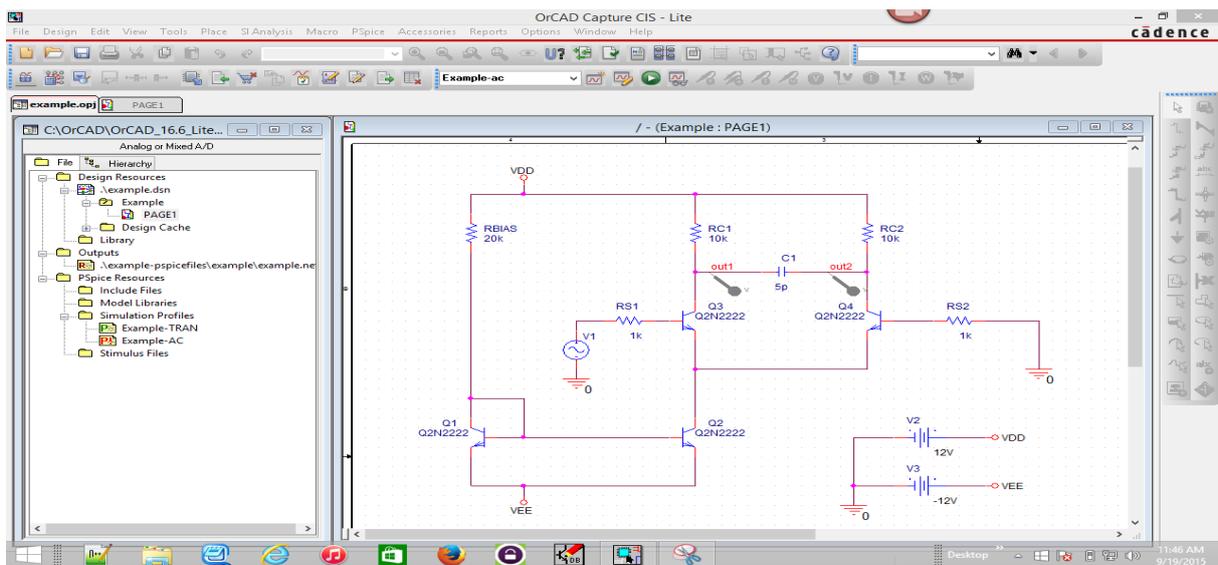


Figure I.1. Fenêtre PSPICE

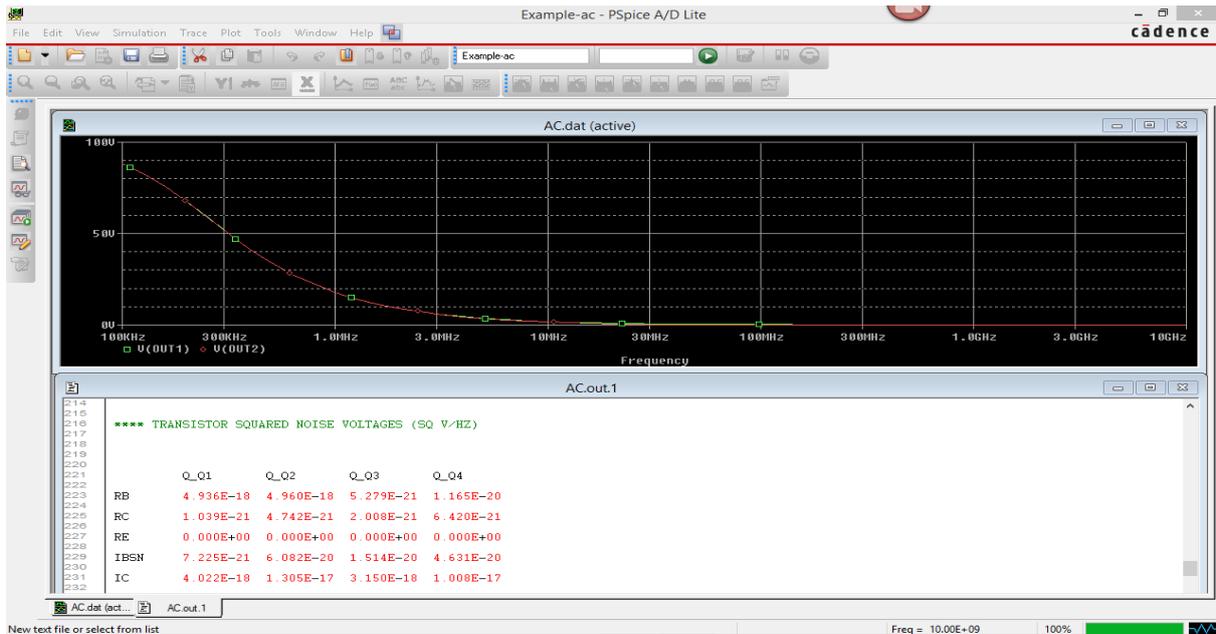


Figure I.2. Résultats de simulations PSPICE

Nous pouvons utiliser tous les composants électroniques connus pour constituer le circuit, résistances, condensateurs, transistors bipolaires et FET. De très nombreux circuits intégrés sont également disponibles en bibliothèque. Mais on peut définir soi-même de nouveaux composants, butinés sur Internet ou achetés. Les résultats de simulations s'accordent généralement très bien avec la pratique, même jusqu'à des fréquences très élevées. À côté des circuits analogiques, les logiciels modernes de simulation permettent aussi de traiter des circuits numériques tels que microcontrôleurs, RAM et logique à portes, mais aussi les antennes et les lignes de transmission [10].

I.4.2. LTSPICE

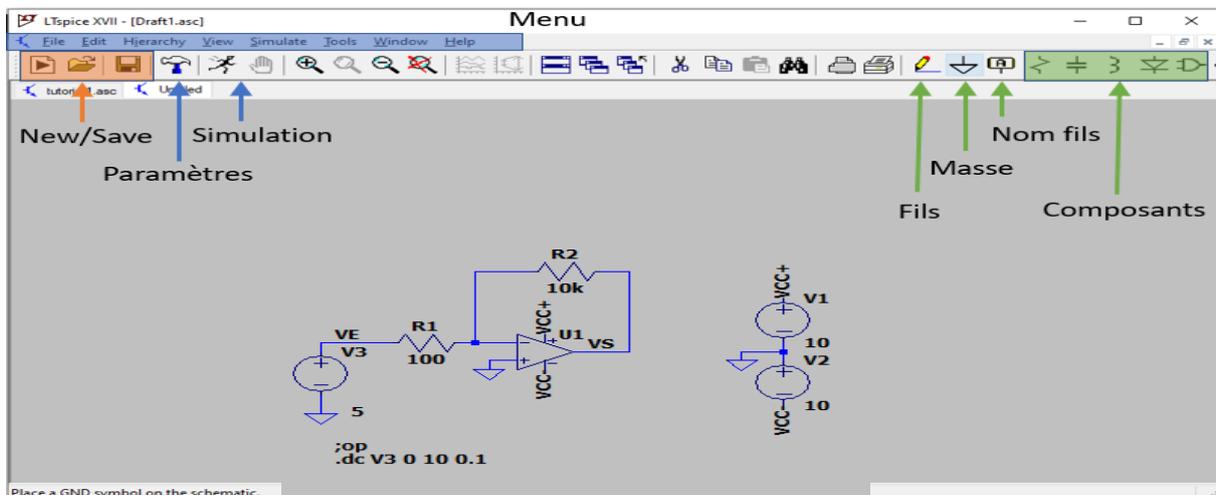


Figure I.1. Fenêtre LTSpice

LTspice est un logiciel de simulation électronique qui permet d'anticiper les caractéristiques et les performances d'un circuit électronique en assemblant à l'écran des composants virtuels[11]. A partir du noyau Spice développé à l'université Berkeley mais très peu convivial, la société Liner Technologie (LT) a développé une version plus visuelle, plus facile d'emploi, et gratuite.

I.4.3. SmartSpice

SmartSpice est un programme de simulation de circuit de haute qualité, de qualité commerciale et à usage général pour les analyses CC non linéaire, transitoire non linéaire et CA linéaire. SmartSpice permet de simuler un circuit en calculant le comportement de tous les composants de circuit simultanément. Grâce à ses nombreux modèles fiables, SmartSpice base sa simulation sur des propriétés physiques, ainsi qu'électriques pour simuler le comportement des circuits plus complexes. Il simule des circuits et sous-circuits composé de résistances, condensateurs, inducteurs, inducteurs mutuels, tension indépendante, courant sources, sources dépendantes, lignes déliées dispositifs à semi-conducteurs : DIMEDE, JIEF, TJE, MOSFET et METAFET[12].

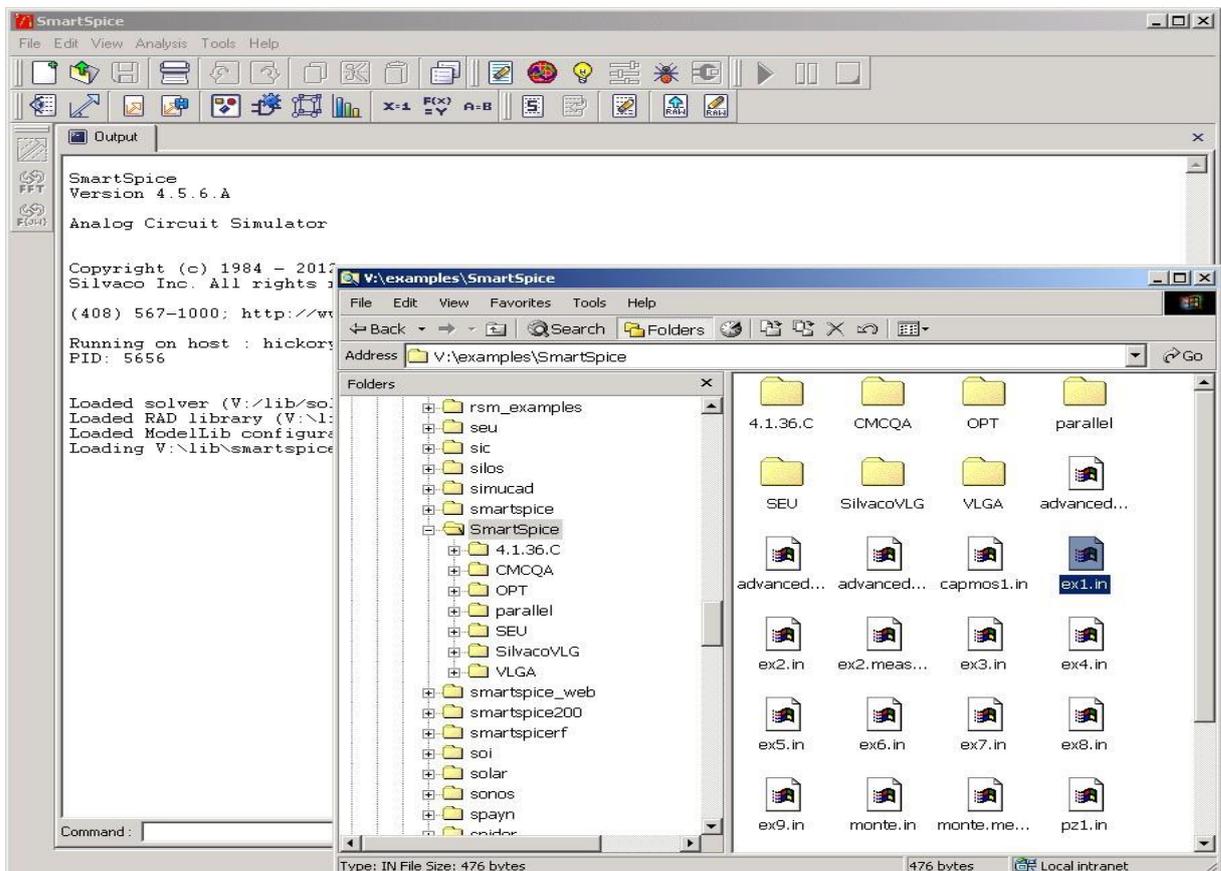


Figure I.2. Fenêtre SmartSpice.

SmartSpice est le résultat d'une recherche et d'un développement intensifs et est supérieur à tous simulateurs de circuit disponibles dans le commerce avec une vitesse compétitive, une convergence supérieure, la précision, et le pré- et post-traitement puissant. SmartSpice est disponible sur les plates-formes Windows et Linux.

SmartSpice peut faire plusieurs analyses tel que :

L'analyse de DC, l'analyse des petits signaux, l'analyse transitoire, Fonction de transfert, Analyse du réseau, Analyse de sensibilité, Analyse du bruit, Analyse des distorsions, Analyse de Fourier, Analyse de Monte Carlo.

I.4.4. Gateway

Gateway est un éditeur schématique, il représente le point d'entrée de la conception car son objectif principal est de générer une NETLIST qui peut être chargée à partir de la représentation schématique du circuit. Les schémas sont dessinés dans Gateway puis transformés en un format de netlist spécifié. Cette netlist est un fichier ASCII décrivant la connectivité du circuit[13].

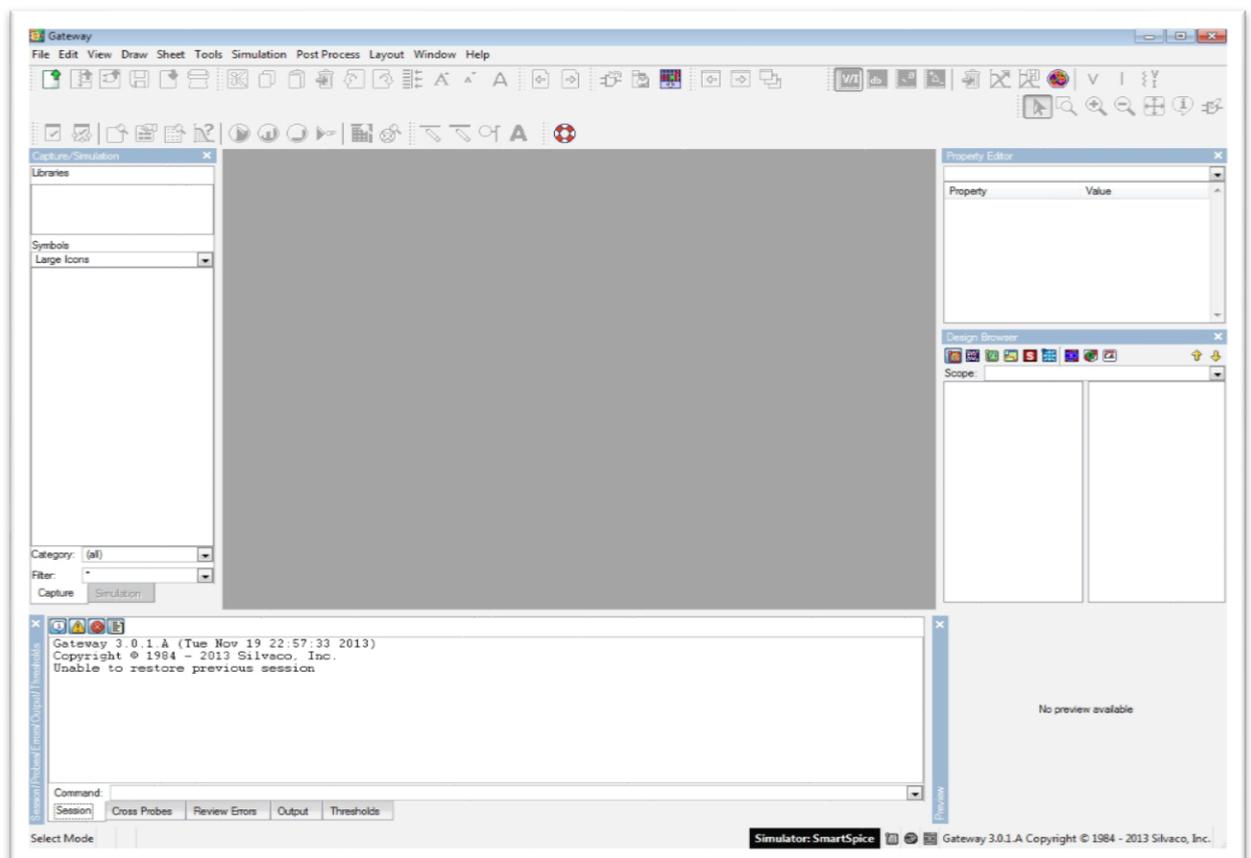


Figure I.3. Fenêtre de Gateway

I.5. Principe de fonctionnement des logiciels Spice

I.5.1. Saisie du schéma

La première opération consiste à saisir le schéma du dispositif qu'on veut simuler ; les programmes commerciaux proposent en général un module de saisie graphique comme dans notre cas Gateway. Sinon il faut utiliser un éditeur compatible SPICE comme ORCAD. L'utilisateur dispose pour cette saisie une bibliothèque contenant des différents types de composants et de sources prédéfinies. Sous Windows, il n'a pas besoin de mémoriser des touches de fonction : tout se fait à la souris et avec des menus déroulants. Il peut aussi définir des conditions initiales (comme la tension aux bornes d'un condensateur à $t = 0$) et choisir la ou les sources (toutes les formes de signal sont possibles : trains d'ondes, impulsions de temps de montée réglables, modulation de fréquence...).

Les caractéristiques de tous les composants électroniques sont stockées dans des fichiers par les fabricants (on trouve par exemple une dizaine de paramètres pour une simple diode)[14].

Une fois le schéma sous Gateway est complété et sauvegardé sous l'extension. Schlr, il est transcrit à travers SmartSpice sous forme d'un fichier .sp ou CIRCUIT aux normes PSPICE ; ce fichier de type texte peut être lu et modifié avec un éditeur de texte standard, et ne dépend donc pas de l'environnement graphique utilisé (DOS, WINDOWS, UNIX, MACINTOSH...).

La description d'un réseau par SPICE consiste à numéroter toutes les connexions, en spécifiant les dipôles présents entre deux nœuds [14].

I.5.2. Simulation

L'utilisateur peut choisir le type d'analyse à effectuer :

Étude en régime transitoire, analyse spectrale, mode sinusoïdal établi ; recherche de fonction de transfert, des déphasages, amplitudes, recherche de point de fonctionnement en continu.

Il lance alors la simulation, dont les résultats sont stockés dans un fichier .DAT [14].

I.5.2.1. Exploitation de la simulation

Un module spécial (SmartView dans notre cas) est utilisé pour traduire le fichier résultat en fichier. Rawsous forme graphiques. On peut obtenir en couleur les tracés de toutes les intensités, potentiel, d.d.p. désirées, ou des expressions de ces grandeurs (dont le module et la phase)[14].

I.6. Conclusion :

Appliquée à l'électronique analogique, la simulation est un outil efficace pour vérifier la fonctionnalité d'une conception ou d'une étude théorique. Elle permet aussi de déceler les erreurs liées à la conception elle-même, à des fonctionnements marginaux non mesurables sur composants réels ou aux tolérances sur les composants. Ainsi, la technique de simulation permet de résoudre tous ces problèmes en intervenant soit sur la conception du composant, soit sur la conception du circuit avant de passer l'étape de la réalisation d'un prototype. Avec les différents logiciels SPICE comme Easyspice, Gspiceui, LTspice, Pspice ou SmartSpice (Gateway), on peut étendre un simple travail vers un projet.

Dans ce chapitre on va présenter une vue générale sur la simulation Spice, les avantages de simulations et les différents logiciels de simulation Spice.

Chapitre II :

Initiation aux simulateurs

Gateway/SmartSpice

II.1. Introduction :

Afin de faciliter la conception des systèmes électroniques performants et plus complexes, dans des temps toujours plus courts, de nombreux outils de CAO (Conception Assistée par Ordinateur) ont vu le jour ces 20 dernières années. Ils permettent de concevoir puis faire la synthèse en mode virtuel des circuits ou des systèmes sous forme de carte électronique ou des ASICs en passant par les étapes de vérification des règles et de la testabilité. L'avantage premier de la CAO est qu'elle permet d'éviter la phase de prototypage, ce qui permet en outre une diminution du coût de fabrication. La simulation donne à l'ingénieur ou au technicien la possibilité de réaliser, puis de tester un prototype virtuel. Les avantages sont évidents : économies de temps et d'argent.

Dans ce chapitre, nous présentons les principales fonctions des simulateurs Gateway et SmartSpice ainsi que leur langage intermédiaire Spice. Ce sont des outils parmi d'autres qui sont important à l'heure actuelle en matière de simulation de circuits analogiques et numériques. Gateway et SmartSpice de la firme Silvaco peuvent être considérés actuellement comme des principaux simulateurs analogique/numérique.

Le simulateur Gateway comprend un éditeur de schéma, un fichier de création de netlist (Create Netlist), un fichier d'éditeur de contrôle (Edit Control file) et un module d'affichage des résultats (SmartViewer) ainsi que d'autres utilitaires qui rendent le logiciel plus agréable pour l'utilisateur.

II.2. Gateway :

II.2.1. Définition :

Gateway est un outil de capture schématique et constitue le point d'entrée du flux de conception de circuits intégrés. Gateway se compose d'une interface graphique pour la saisie schématique et il est intégré aux simulateurs suivants :

- Silvaco Smart Spice (analogique).
- Silvaco SmartSpice RF (RF).
- Silos Silvaco (Verilog).
- Silvaco Atlas (TCAD Mixed Mode).
- Synopsys HSPICE (analogique)

Le but principal du Gateway est de générer une liste de réseau qui peut être chargée dans les simulateurs de circuit. Les schémas sont saisis dans Gateway et ensuite transformés en un format netlist spécifié. Cette netlist est un fichier ASCII décrivant la connectivité du circuit.

C'est le premier fichier nécessaire pour le flux de simulation. Le fichier suivant utilisé dans le processus de conception est le fichier de contrôle, qui peut être créé pour chaque simulateur que Gateway exécutera. Ce fichier de contrôle contient toutes les options et paramètres du modèle et de la simulation. Lorsque vous créez une liste de réseau et un fichier de contrôle, ils fusionnent pour former la partie principale du fichier d'entrée.

II.2.2. Fenêtre Gateway :

Gateway dispose de deux modes de fonctionnement : Capture et Simulation. Le mode Capture affiche les bibliothèques de symboles et sert à la création de schémas. Le mode de simulation dispose de paramètres pour tracer les options et les préreglages de simulation. La Figure II-1 montre Gateway lors de son premier lancement. En plus de la fenêtre principale de dessin schématique, il y a plusieurs zones de panneau que vous pouvez redimensionner, ancrer, défaire ou masquer. Ces zones comprennent :

- Capture/simulation affiche les bibliothèques de symboles en mode capture et les options de simulation en mode simulation.
- L'aperçu montre la feuille entière indiquant la fenêtre de visualisation actuelle.
- Session/cross Probes/review Errors/output :
 - La session affiche des informations en temps réel issues de la capture schématique, de l'édition et des actions lors de l'exécution d'une simulation.
 - Cross Probes affiche une liste de sondes à enregistrer, défiler ou tracer.

Review Errors affiche une liste d'avertissements ou d'erreurs trouvés après une vérification de dessin.

- Output affiche la sortie d'exécution des simulations.
 - Thres holds affichent les paramètres d'affichage des courants et des tensions du dispositif à afficher entre les seuils spécifiés en mode polarisation CC.
 - Design Browser est l'arborescence de la structure hiérarchique du schéma.
 - Le volet propriété affiche les attributs de l'objet sélectionné sur le dessin aux fins de visualisation et d'édition.

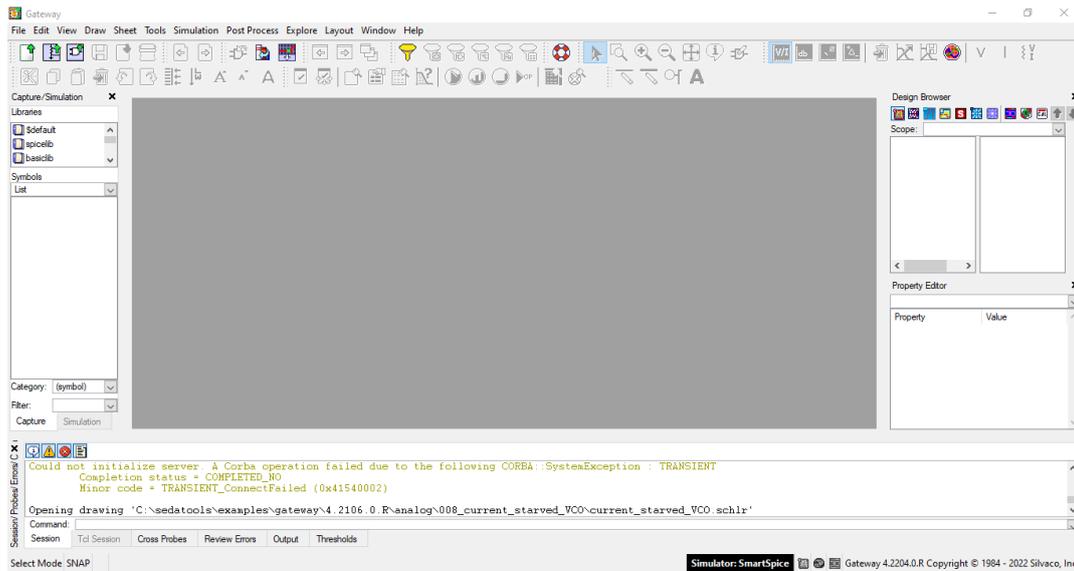


Figure II.1. Fenêtre Gateway

II.2.3. Création d'un nouveau projet :

II.2.3.1. Espace de travail workspace :

L'espace de travail est un fichier contenant les bibliothèques qui seront utilisées pour créer un schéma. Le fichier de l'espace de travail dans Gateway a une extension **.workspace**. Vous devez créer ou charger un espace de travail avant de créer un schéma ou un dessin de symbole. Pour créer un nouvel espace de travail, sélectionnez **File**New**Workspace** et la boîte de dialogue Nouvel espace de travail s'affiche (Figure II.2).

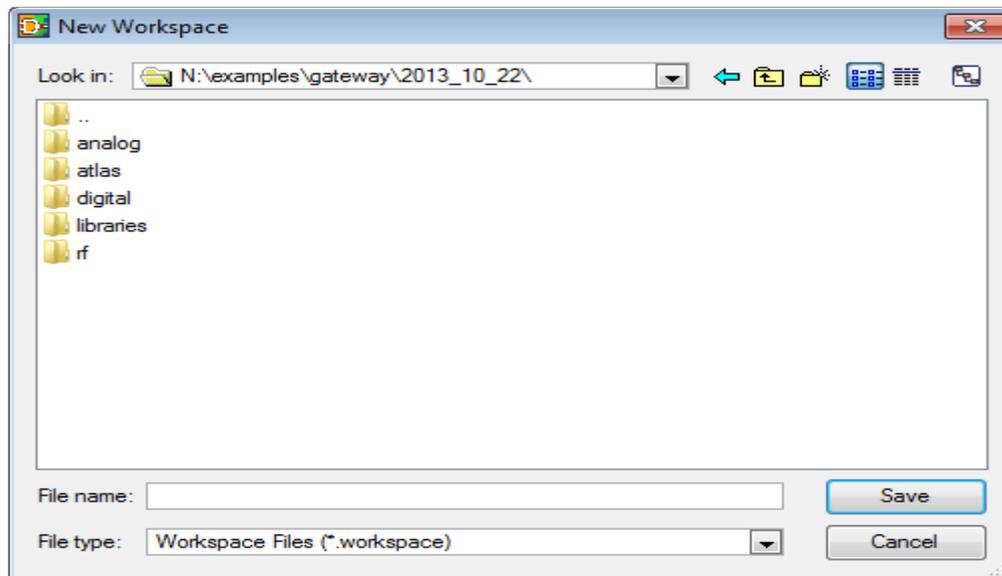


Figure II.2. Nouvelle boîte de dialogue Espace de travail

Saisissez un nom de fichier et cliquez sur OK. La boîte de dialogue Paramètres de l'espace de travail s'affiche alors (Figure II.3).

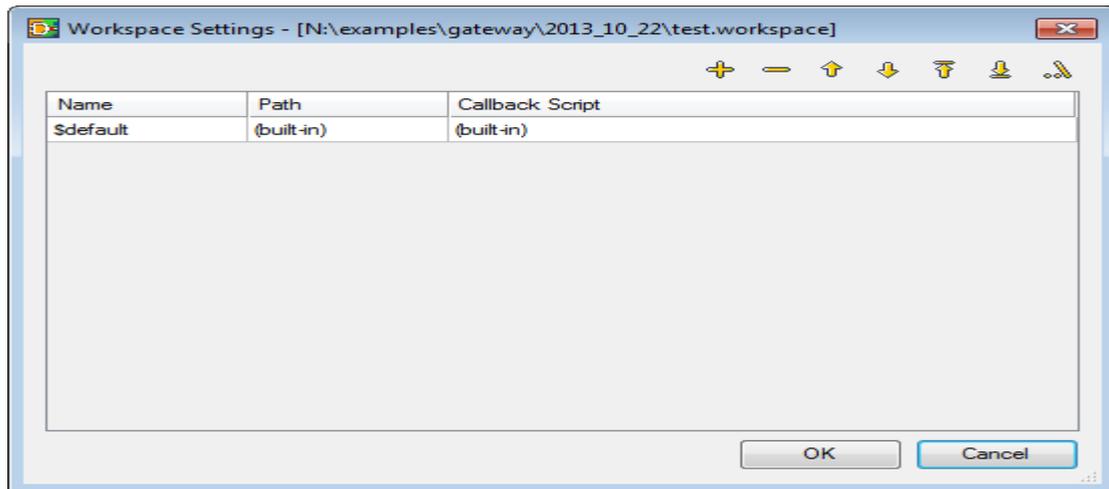


Figure II.3. Boîte de dialogue Paramètres de l'espace de travail

Lorsque cette boîte de dialogue s'affiche, appuyez sur Ajouter (+) pour ajouter la première bibliothèque.

Une ligne vide est insérée dans la boîte de dialogue. Pour nommer la bibliothèque, cliquez sur la bibliothèque avec le bouton gauche de la souris et saisissez un autre nom. Une fois la bibliothèque nommée, le chemin d'accès au dossier de la bibliothèque doit être spécifié. Si les fichiers de symboles dans la bibliothèque font référence à un fichier callback, saisissez le nom du fichier dans **Callback Script**. Pour spécifier un chemin, tapez le chemin dans le champ Chemin ou appuyez sur Parcourir pour trouver le chemin où réside le dossier.

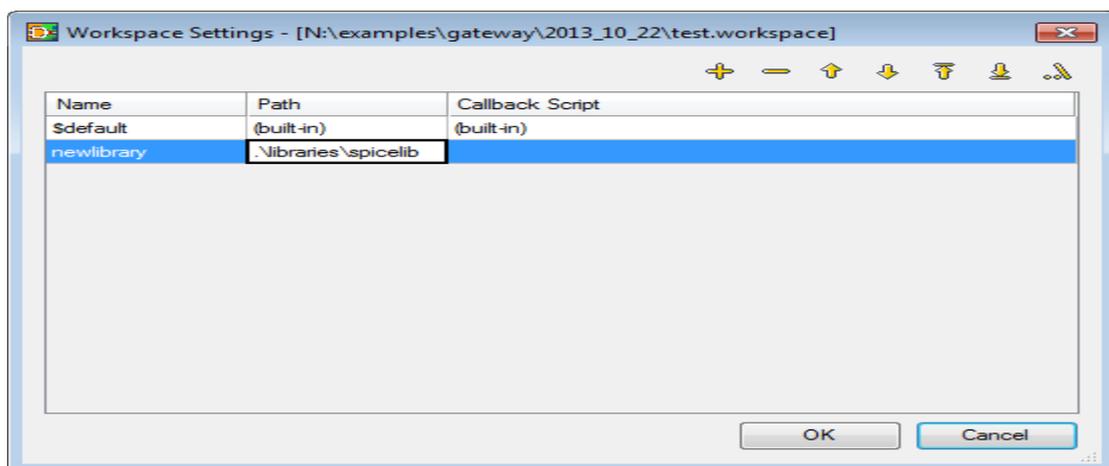


Figure II.4. Dialogue de configuration d'espace de travail avec chemin relatif

Lorsque vous utilisez le bouton Parcourir, Gateway spécifie ce chemin de fichier comme chemin relatif (voir Figure II.4). Le chemin peut également contenir des variables d'environnement. Ceci est réalisé en ajoutant des chevrons autour de la variable d'environnement.

Pour supprimer une bibliothèque, cliquez sur la bibliothèque ou le chemin et appuyez sur Supprimer (-). Les bibliothèques peuvent être organisées dans n'importe quel ordre à l'aide des boutons Déplacer vers le haut (↑) et Déplacer vers le bas (↓) dans la boîte de dialogue.

Lorsque les bibliothèques ont été ajoutées avec leurs chemins respectifs, appuyez sur OK pour accepter les modifications et fermer la boîte de dialogue.

II.2.3.2. Bibliothèques :

Une bibliothèque dans Gateway est un répertoire sur un système d'exploitation qui contient des schémas ou des symboles ou les deux. Les schémas ont un l'extension de fichier schlr et les fichiers de symboles ont une extension de fichier symbol.

La structure de bibliothèque dans Gateway est plate, ce qui signifie que chaque répertoire doit être mappé individuellement dans l'espace de travail pour être valide. Les sous-répertoires sont ignorés s'ils ne sont pas mappés avec leurs propres chemins dans l'espace de travail. Les symboles dans le volet Bibliothèque/Symboles sont classés par ordre alphabétique. Vous pouvez choisir parmi l'une des quatre vues disponibles pour afficher les symboles disponibles en utilisant la boîte bascule dans le panneau de symboles.

Spicelib est la bibliothèque par défaut dans Gateway. Il contient les symboles SmartSpice actifs et passifs nécessaires pour simuler n'importe quelle conception simple utilisant SmartSpice. La Spicelib contient des cellules de sous-circuit pour les exemples. Bien que les schémas soient contenus dans les bibliothèques, ils ne sont pas visibles dans les zones de bibliothèque ou de panneau de symboles. Seuls les symboles apparaissent dans le panneau de symboles. S'il y a un symbole et un schéma portant le même nom dans une même bibliothèque, tels que inverter.symbol et inverter.schlr, le symbole de l'onduleur apparaît dans le panneau des symboles de cette bibliothèque. Cela décrit également une paire descendante, ou un fichier symbole qui descend dans sa vue schématique, car les fichiers ont le même nom de base et sont dans la même bibliothèque.

La Figure II.5 montre le volet bibliothèque de la fenêtre principale Symbole/Options. Les bibliothèques doivent être plates (non hiérarchiques) et ceci est indiqué par la disposition du volet bibliothèque. Lorsqu'une bibliothèque est sélectionnée en cliquant dessus, les symboles de cette bibliothèque sont affichés dans le panneau de symboles ci-dessous. Gateway génère automatiquement des icônes de symboles pour refléter l'empreinte réelle du symbole. Tous les symboles fournis avec Gateway ou créés par un concepteur s'affichent dans ce panneau. La Figure II.6 montre la vue des grandes icônes pour les symboles de bibliothèque,

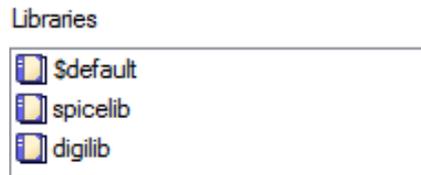


Figure II.5. Volet de bibliothèque

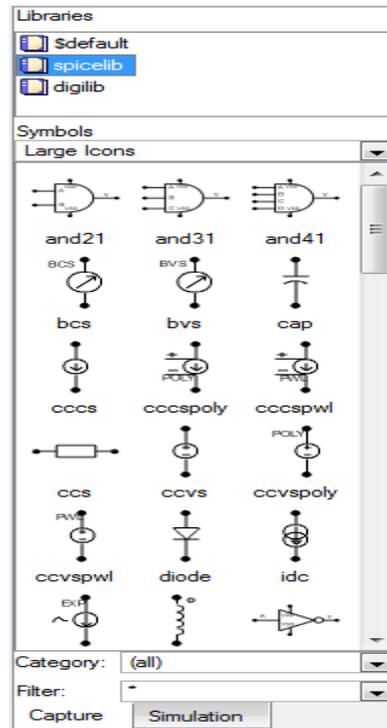


Figure II.6. Affichage des symboles en grandes icônes

II.2.3.3. Création d'un fichier schématique :

Un espace de travail doit être créé avant qu'un schéma ou un symbole puisse être ouvert. Pour créer un nouveau fichier schématique, on clique sur ctrl+N ou bien faire : File → New → schematic.

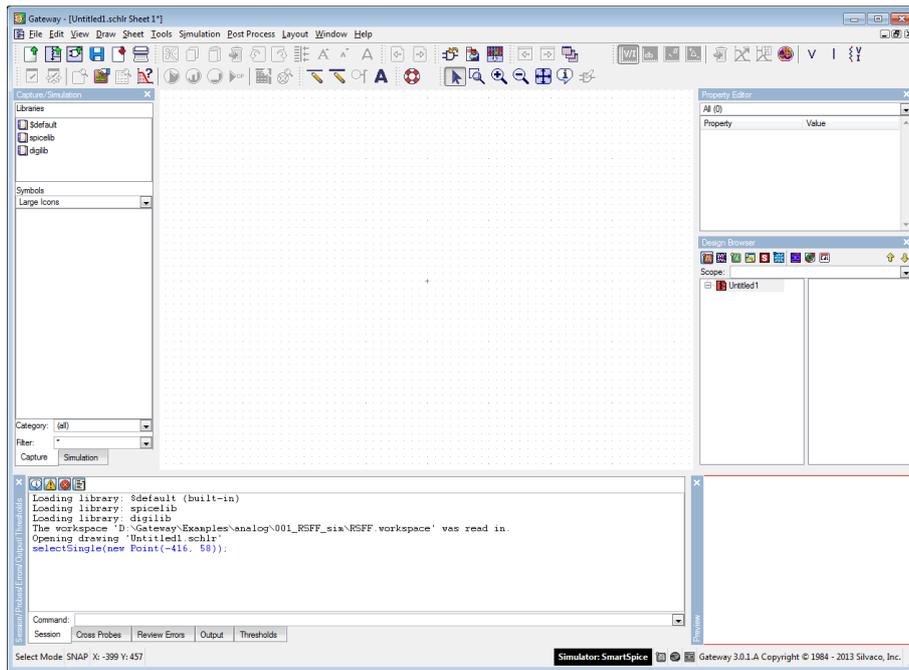


Figure II.7. Nouveau fichier schématique

II.2.3.4. Sauvegarde du fichier schématique :

La sauvegarde du schéma s'effectue en cliquant sur l'icône  ou en choisissant File → save ou raccourci de clavier CTRL+S.

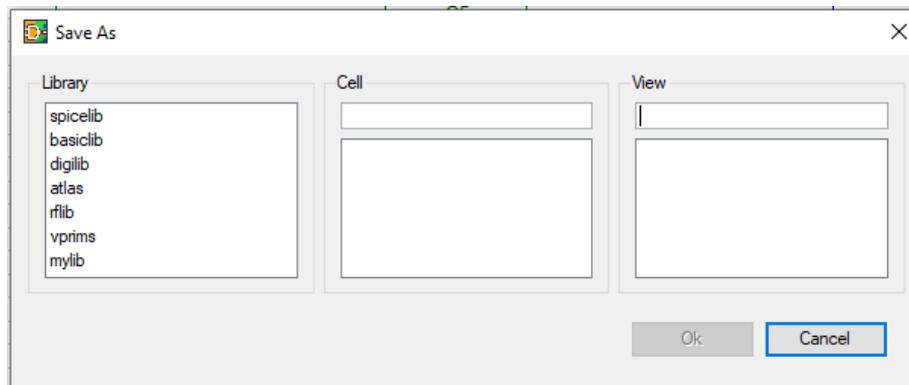


Figure II.8. Boîte de dialogue pour enregistrer le schéma

Choisissez un dossier parmi les bibliothèques existantes dans lequel le schéma sera enregistré, puis donnez un nom au Cell, puis un nom au View, puis appuyez sur OK.

II.2.4. Saisie du schéma :

II.2.4.1. Placer des composants :

Pour placer un composant sur un schéma :

- Sélectionnez une bibliothèque. Pour cet exemple, utilisez **spicelib**.

- Cliquez sur le nom du symbole dans la liste de la fenêtre Symbole. Dans cet exemple, sélectionnez **vpulse**. Le nom du symbole sera mis en évidence et le symbole sera attaché au curseur, flottant autour de la zone schématique jusqu'à ce qu'elle soit placée.

Le symbole peut pivoter ou être mis en miroir lorsqu'il se déplace en cliquant avec le bouton droit de la souris pour utiliser le menu d'orientation des symboles. Tournez-le jusqu'à ce que vous ayez l'orientation souhaitée du symbole.

- Cliquez avec le bouton gauche de la souris pour déposer le symbole sur le schéma. Le symbole aura points d'interrogation (?) à côté. Cela signifie que ces attributs doivent être définis.

Une fois le symbole placé, un autre symbole, qui sera du même type, sera attaché au curseur. Pour placer un autre symbole, cliquez à nouveau sur le bouton gauche de la souris. Pour placer un autre type de symbole, cliquez sur un autre nom de symbole dans la fenêtre Symbole et répétez les étapes décrites ci-dessus.

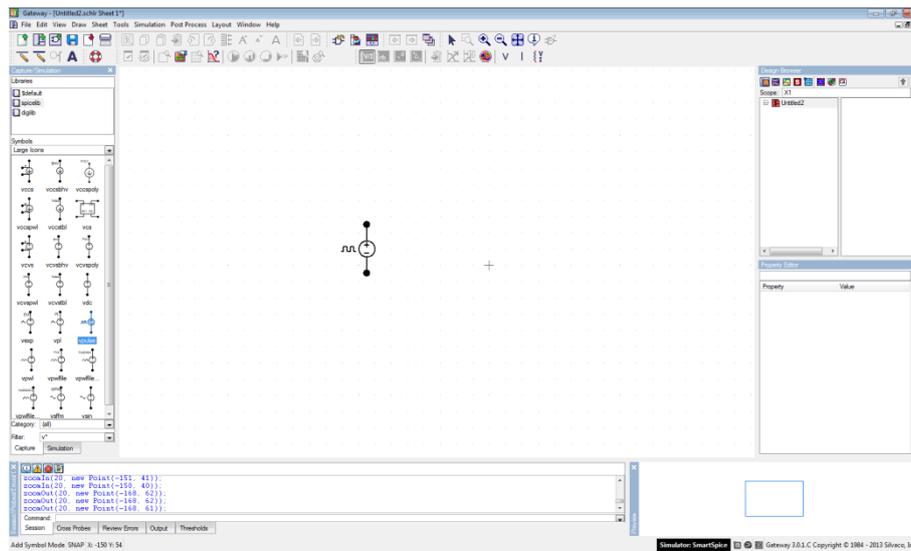


Figure II.9. Placement du symbole Vpulse

II.2.4.2. Création de fils :

Une fois les symboles placés, ils devront être reliés entre eux. Pour créer un fil :

- Presse  ou sélectionnez Tracer un fil.
- Commencez le fil en plaçant le curseur à l'endroit où le fil commencera et cliquez avec le bouton gauche de la souris.
- Déplacez la souris dans la direction du point d'arrivée du fil et double-cliquez sur le bouton gauche pour terminer le fil.

- Cliquez avec le bouton droit de la souris pour quitter le mode fil.



Figure II.10. Création de fils

Toute simulation nécessite obligatoirement une référence de tension : le symbole de masse

dont le nom est 0. Symbole :



II.2.4.3. Chargement des fichiers :

Cette section quickstart (démarrage rapide) décrit les étapes de base de l'utilisation de Gateway en passant par l'exemple miroir de courant actuel qui est livré avec le logiciel. Tout d'abord, lancez Gateway, puis ouvrez l'espace de travail du miroir actuel. Pour ce faire, sélectionnez **File**Open**Workspace**, D'abord Puis, naviguez vers le répertoire suivant :

<installation_directory>\examples\gateway\<date>\analog\006_current_mirror

Sélectionnez current_mirror.workspace et cliquez sur OK. Lorsque l'espace de travail se charge, les bibliothèques disponibles apparaissent dans le volet bibliothèque. Cliquer sur l'une de ces bibliothèques dans le panneau affiche la liste des symboles disponibles ci-dessous dans le panneau des symboles. Vous pouvez modifier la vue d'affichage des symboles en cliquant sur le menu déroulant et en choisissant une vue. La vue des grandes icônes est la vue par défaut.

Sélectionnez maintenant **File**Open**Schematic** et sélectionnez current_mirror SCHL dans la boîte de dialogue, puis cliquez sur OK. La fenêtre Gateway apparaîtra comme le montre la Figure II-11.

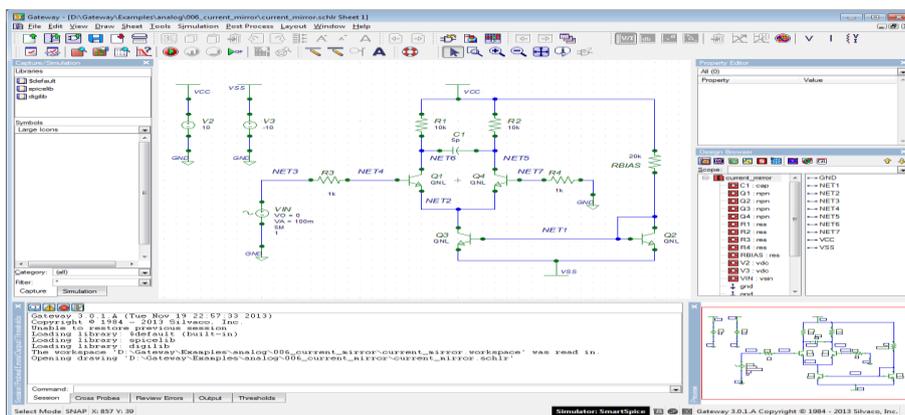


Figure II.11. Exemple de Gateway current_mirror. Schlr



Figure II.12. Tableau de simulation

II.2.4.4. Vérification de la conception :

Vous devriez vérifier la conception avant Netlisting. Un contrôle de dessin propre est susceptible de générer une netlist sans erreurs. Pour Vérifier le schéma électrique cliquez sur

l'icône  . Une liste d'erreurs apparaîtra ci-dessous (figure II.13), corrigez les erreurs une par une.

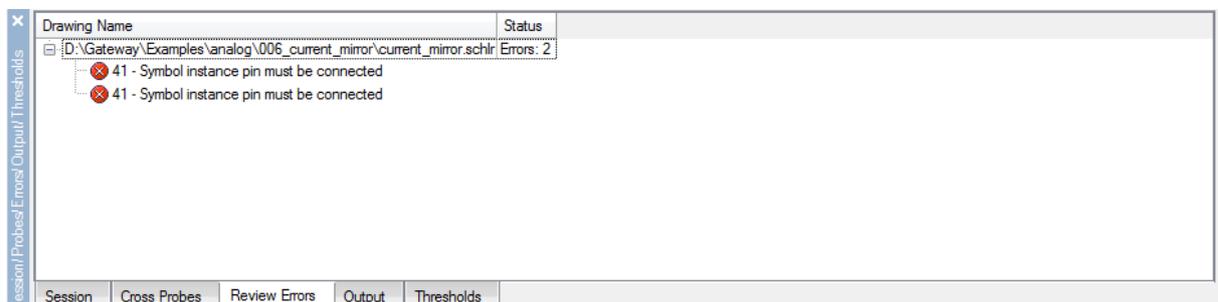


Figure II.13. Volet d'erreurs

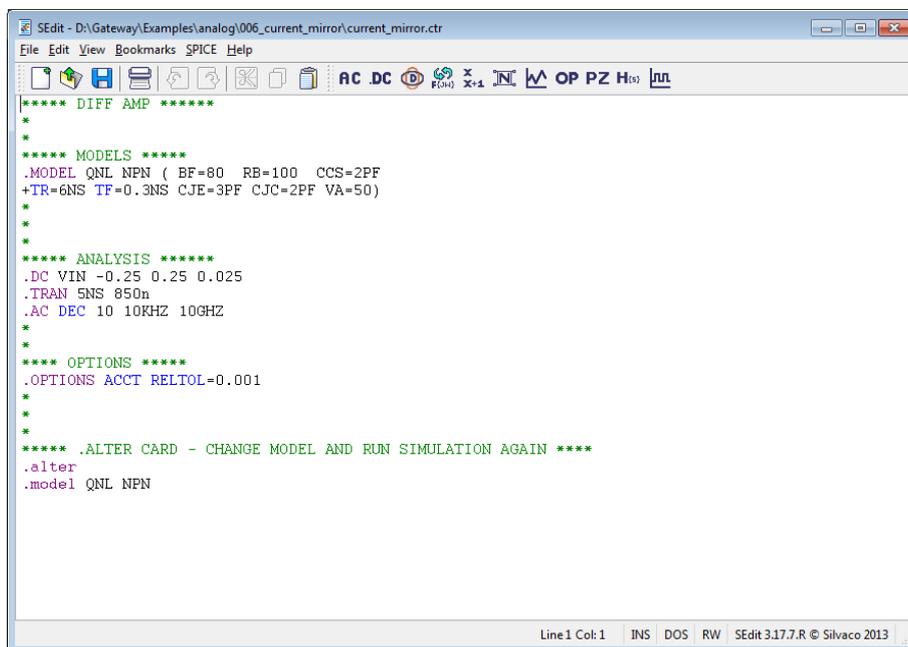
II.2.4.5. Netlisting et panneau de contrôle :

Une netlist est un fichier créé par Gateway contenant des informations de connectivité entre les symboles, les signaux, les fils et les broches d'un schéma. Les types de netlist que Gateway peut générer sont les suivants :

- **SmartSpice** : La netlist SmartSpice est la netlist de simulation pour exécuter le simulateur de circuit analogique SmartSpice. Par exemple, si la conception schématique est nommée RSFF.schlr, la netlist SmartSpice est RSFF.net.
- **SmartSpice Pro** : La netlist SmartSpice 200 est destinée à la simulation à l'aide de SmartSpice 200.
- **SmartSpice RF** : la netlist SmartSpice RF est utilisée pour la simulation avec SmartSpice RF.
- **Silos** : la netlist Silos est une netlist Verilog et est utilisée pour exécuter des simulations Verilog dans Silos.
- **HSPICE** : Le format netlist HSPICE est utilisé pour exécuter des conceptions analogiques dans HSPICE.

- **Atlas** : La netlist Atlas est générée pour exécuter des simulations TCAD Atlas Mixed Mode.
- **NDL** : Le NDL (Netlist Driven Layout) est la netlist qui a des directives pour une plus grande productivité dans la pré-mise en page.
- **Guardian** : La netlist Guardian reflète le layout physique de LVS. En utilisant l'exemple RSFF, la netlist Guardian est RSFF_lvs.net.
- **CDL** : La liste d'interconnexions CDL est l'option de liste d'interconnexions Cadence CDL.

La conception est maintenant prête pour netlist. Cliquez sur l'icône Create Netlist  et la netlist apparaîtra.



```

SEdit - D:\Gateway\Examples\analog\006_current_mirror\current_mirror.ctr
File Edit View Bookmarks SPICE Help
AC DC P(S) X+1 JN OP PZ H(s) IIN
***** DIFF AMP *****
*
*
***** MODELS *****
.MODEL QNL NPN ( BF=80 RB=100 CCS=2PF
+TR=6NS TF=0.3NS CJE=3PF CJC=2PF VA=50)
*
*
***** ANALYSIS *****
.DC VIN -0.25 0.25 0.025
.TRAN SWS 850n
.AC DEC 10 10KHZ 10GHZ
*
*
**** OPTIONS ****
.OPTIONS ACCT RELTOL=0.001
*
*
***** .ALTER CARD - CHANGE MODEL AND RUN SIMULATION AGAIN *****
.alter
.model QNL NPN
Line 1 Col: 1 INS DOS RW SEdit 3.17.7.R © Silvaco 2013

```

Figure II.14. Exemple de Netlist sous SmartSpice

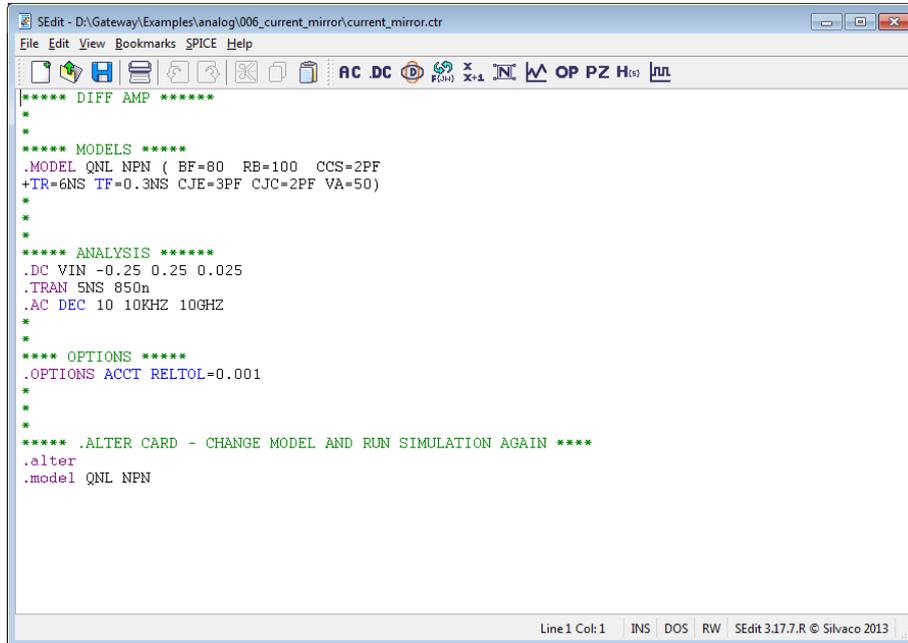
La netlist a été créée avec succès.

Cliquez maintenant sur l'icône Edit Control File  pour voir les instructions de contrôle et la fenêtre Control File apparaîtra (Figure II.14).

Le fichier de contrôle (*. Ctr) est un fichier contenant des informations et des instructions sur ce qui doit être exécuté dans le simulateur cible par exemple, si le simulateur choisi est SmartSpice, alors le fichier de contrôle devient un fichier SPICE. Vous pouvez créer le fichier en tapant dans les instructions ou en utilisant des boîtes de dialogue pour aider à écrire certaines instructions d'analyse. Habituellement, au moins un énoncé d'analyse est requis. Vous pouvez utiliser un certain nombre des options ou d'autres énoncés SPICE pour créer un

profil de simulation. De plus, tous les fichiers de bibliothèque ou autres fichiers nécessaires à la simulation sont ajoutés ici en utilisant les instructions `.LIB` ou `.INCLUDE`.

La barre d'outils d'analyse (Figure II.14) lance des boîtes de dialogue pour la plupart des déclarations d'analyse SPICE.



```

SEdit - D:\Gateway\Examples\analog\006_current_mirror\current_mirror.ctr
File Edit View Bookmarks SPICE Help
AC DC [Distortion] [Fourier] [Modif] [Network] [Noise] [OP PZ H{x} [X+1] [F{JW}]

***** DIFF AMP *****
*
***** MODELS *****
.MODEL QNL NPN ( BF=80 RB=100 CCS=2PF
+TR=6NS TF=0.3NS CJE=3PF CJC=2PF VA=50)
*
*
***** ANALYSIS *****
.DC VIN -0.25 0.25 0.025
.TRAN 5NS 850n
.AC DEC 10 10KHZ 10GHZ
*
*
**** OPTIONS ****
.OPTIONS ACCT RELTOL=0.001
*
*
***** .ALTER CARD - CHANGE MODEL AND RUN SIMULATION AGAIN ****
.alter
.model QNL NPN
  
```

Figure II.15. Fenêtre du fichier de contrôle



Figure II.16. Barre d'outils Analyses

Par exemple, pour générer une instruction `.TRAN`, cliquez sur l'icône transitoire  et saisissez les paramètres dans la boîte de dialogue de l'analyse Transitoire (Figure II.15).

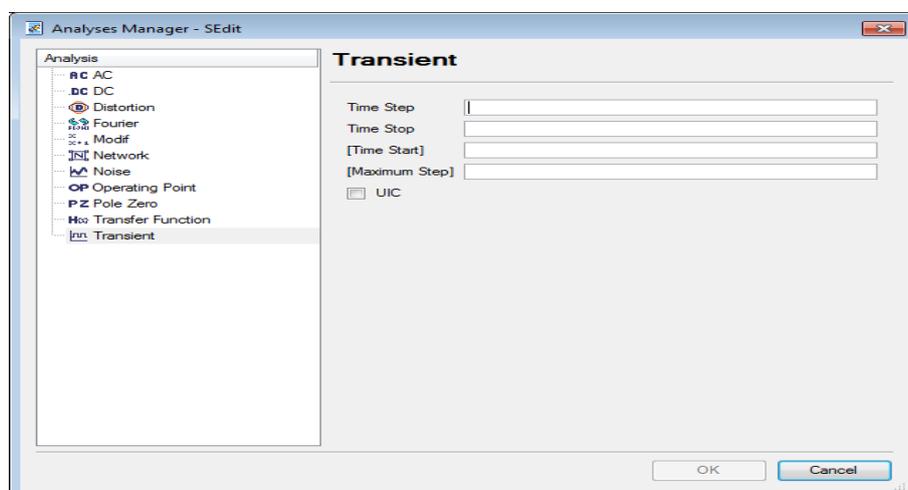


Figure II.17. Dialogue d'analyse transitoire

II.2.4.6. Simulation :

Pour simuler, appuyez sur le bouton Exécuter.  Lorsque la simulation commence, la boîte de dialogue simulation apparaît avec les statistiques d'exécution (Figure II.18). Après la simulation, SmartView apparaît avec les tracés de forme d'onde. Dans Cette simulation, il existe deux vecteurs marqués : V(NET5) et V(NET6). Il y a une déclaration ALTER dans la présentation qui exécute la simulation deux fois. En outre, il y a trois analyses : .TRAN, .DC et .AC. Par conséquent, il y a un total de $(2*2*3) = 12$ vecteurs tracés, 4 pour le transitoire, 4 pour le DC et 4 pour le AC. Voir la figure II.19 pour un exemple.

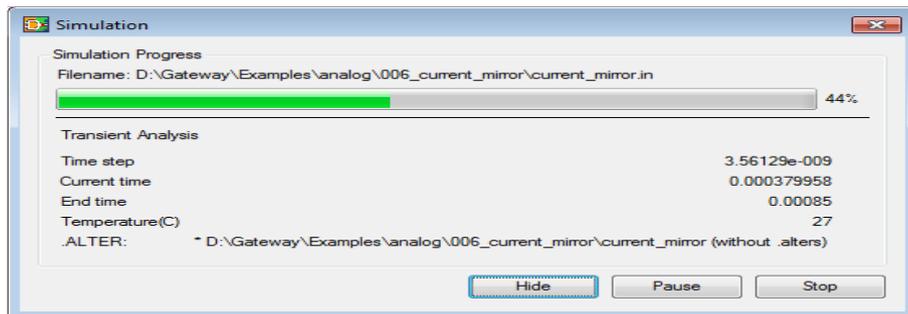


Figure II.18. Boite de dialogue de la progression de la simulation

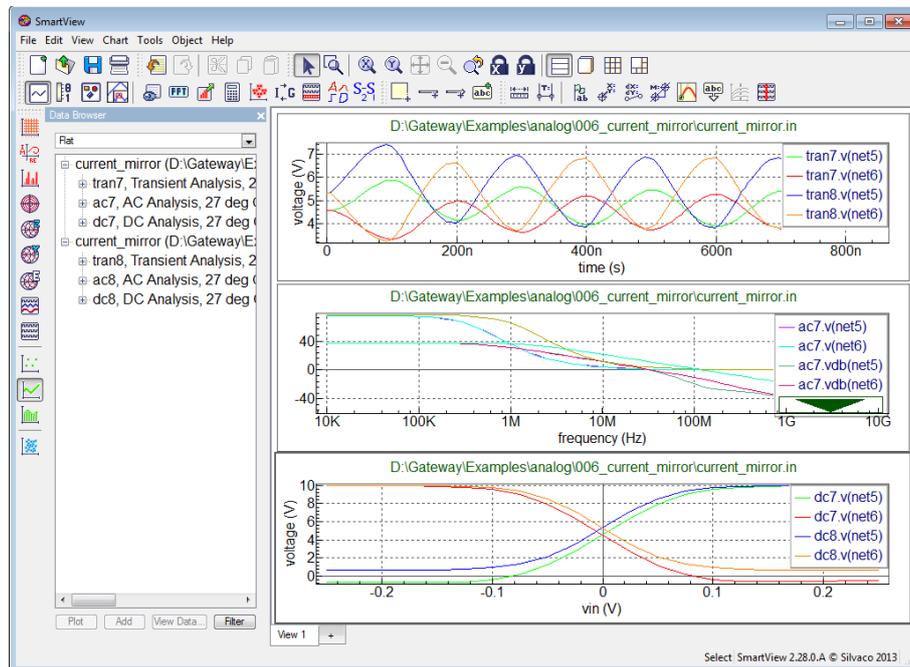


Figure II.19. Exemple de SmartView

II.2.4.7. Boutons de la barre d'outils de simulation :

- Pour arrêter la simulation, appuyez sur 

- Pour interrompre la simulation, appuyez sur 

- Pour reprendre une simulation, appuyer sur 
- Calcule des points de polarisation DC sans effectuer de simulation 
- Affiche le fichier d'erreur de simulation 
- Affiche l'erreur de simulation. 

II.2.4.8. Affichage de polarisation DC :

Gateway affiche les tensions et les courants du point de fonctionnement DC une fois les simulations DCOP terminées. Pour afficher ces valeurs sur le schéma, cliquez sur l'un des boutons de la barre d'outils illustrée dans la Figure II.20.



Figure II.20. Boutons de polarisation DC

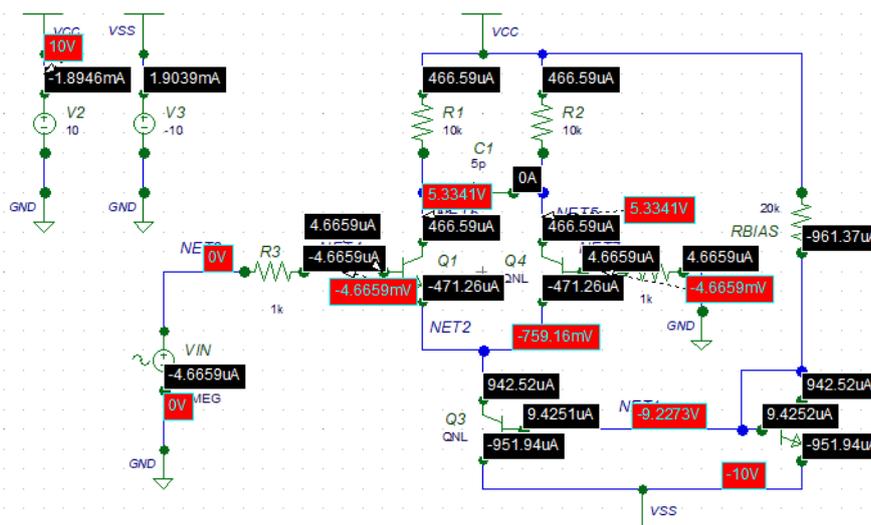


Figure II.21. Exemple des tensions et courants du point de fonctionnement

II.2.4.9. Post-traitement

La barre d'outils cross probe (Figure II.22) affiche les icônes qui traitent les données.



Figure II.22. Barre d'outils Cross Probe

Pour tracer les données après une simulation, marquez les nœuds ou les broches sur le schéma et cliquez sur . Pour supprimer des marqueurs de la pile dans le volet Sonde

croisée, cliquez sur  et tous les marqueurs cochés seront supprimés. Pour lancer SmartView, cliquez sur .

Le bouton Tracé à partir de l'archive  ouvre le dernier fichier brut enregistré pour la simulation de schéma de niveau supérieur chargée et vous permet de tracer les données vectorielles de cette analyse ou de ces analyses. Notez que lorsque les simulations sont exécutées en séquence, bien que vous puissiez voir toutes les simulations exécutées dans la session en cours, le fichier rawfile ne conserve que l'analyse finale pour être visible dans la session suivante. Si vous souhaitez accéder à toutes les données de la session précédente, vous devez enregistrer le fichier rawfile avant de quitter SmartView.

II.2.4.10. Ajouter du texte :

Du texte et des étiquettes peuvent être ajoutés au schéma. Pour ajouter du texte au schéma :

- Presse  ou sélectionnez **Draw**Texte** la fenêtre Entrer du texte apparaîtra.
- Saisissez le texte et appuyez sur **OK**. Le texte flottera maintenant avec le curseur.
- Cliquez avec le bouton gauche de la souris pour le placer à l'emplacement souhaité.

II.2.4.11. Filtrage de sélection :

La barre d'outils Sélectionner un filtre vous permet de choisir un sous-ensemble spécifique d'éléments graphiques lorsque vous cliquez ou faites glisser la sélection sur la zone de dessin.



Figure II.23. Sélectionner la barre d'outils du filtre

 Le filtrage de sélection est désactivé par défaut, c'est-à-dire que tous les éléments sont sélectionnés. Pour activer le filtrage, cliquez sur le bouton Sélection du filtre.

 Lorsque le filtrage est activé pour la première fois, le bouton All On est automatiquement activé et tous les éléments sont conservés lorsqu'une sélection est effectuée.

 Vous pouvez faire une sélection plus finement avec le bouton Tous les paramètres.

II.3. Le choix de simulateur

Les paramètres du simulateur dans les Préférences permettent de choisir le simulateur cible, la version, le chemin et diverses options de démarrage pour les simulateurs suivants :

- Smartspice, Smartspice RF et smartspice PRO
- Atlas (DeckBuild)
- HSPICE
- Silos
- VictoryDevice

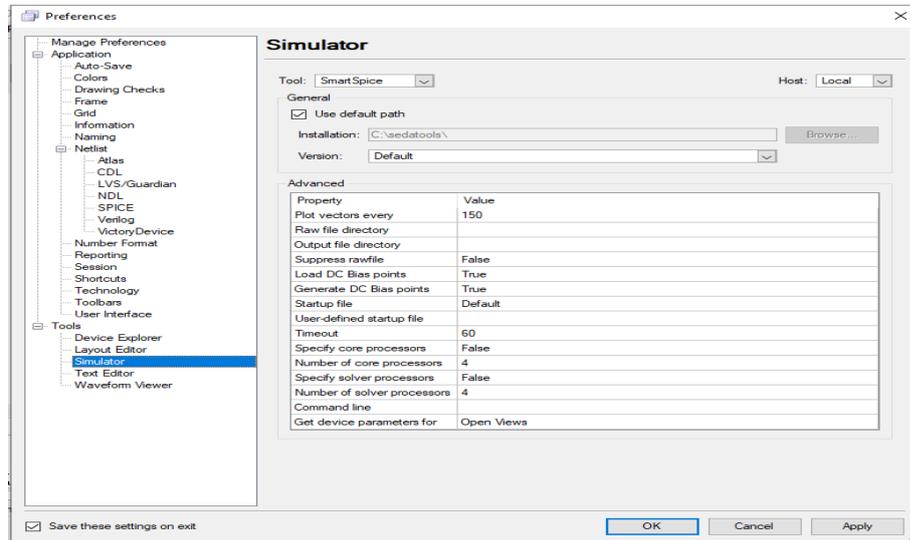


Figure II.24. Boîte de dialogue Préférences

II.3.1. État et informations de SmartSpice :

SmartSpice est défini comme simulateur par défaut. Lorsqu'elle est cochée, la case utiliser le chemin par défaut définit le chemin vers le même emplacement où Gateway est installé. Si cette case n'est pas cochée, vous pouvez définir le chemin vers un autre emplacement pour pointer vers le répertoire où le simulateur est installé. La combo-box permet de sélectionner la version. La valeur par défaut est la dernière version. La liste déroulante Hôte vous permet de choisir d'exécuter SmartSpice localement ou à distance. En cas d'exécution à distance, vous devrez entrer l'hôte, l'utilisateur, le mot de passe et l'emplacement de l'exécutable sur la machine distante.

Dans la partie inférieure droite de l'image de base de la fenêtre Gateway, il y a une série d'affichages d'état qui fournissent des informations de connexion sur le simulateur et d'autres applications connectées à Gateway. La Figure II.25 montre cette zone de la fenêtre Gateway, qui inclut le numéro de version de la passerelle en cours d'exécution.



Figure II.25. Barre d'état du simulateur et de l'outil

Le Simulateur est réglé sur SmartSpice. La première icône de la série de trois icônes est l'icône du simulateur. La deuxième icône est l'icône du visualisation de formes d'onde. Dans ce cas, SmartView est affiché en tant que visualisation de formes d'onde choisi pour cette session. La troisième icône est pour l'éditeur de layout, et dans cette session, il est défini à Expert. Les trois icônes sont allumées et indiquent que Gateway est connecté aux trois applications. Si l'un d'eux est déconnecté ou fermé, l'icône devient grise montre la passerelle déconnectée d'Expert.

Pour afficher les détails de connexion avancés, passez la souris sur l'une des icônes et l'infobulle dont plus d'informations. La Figure II.26 montre des informations avancées pour la connexion SmartSpice connections, y compris l'état et la version de SmartSpice en cours d'exécution.

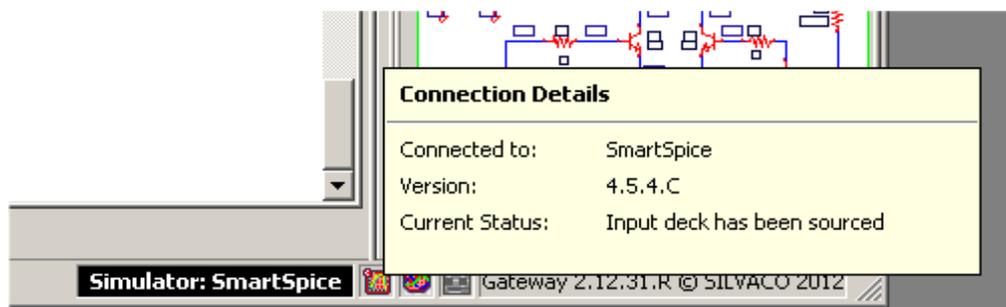


Figure II.26. Barre d'état affichant des informations SmartSpice avancées

II.4. Les Types d'analyses dans SmartSpice

II.4.1. Analyses .OP, .DC, .AC et .TRAN :

Les analyses OP, DC, AC et transitoires génèrent les types de sortie suivants :

- L'analyse OP génère des vecteurs réels (vecteurs de longueur un(1)).
- L'analyse DC génère des vecteurs réels avec une dimension qui dépend des arguments de l'instruction .DC et avec une variable spécifiée sur l'instruction .DC comme échelle.
- Les analyses AC et réseau génèrent des vecteurs complexes dont la dimension dépend d'une gamme de fréquences spécifiée, avec la fréquence comme échelle.
- L'analyse transitoire génère des vecteurs réels dont la dimension dépend d'une plage de temps spécifiée, avec le temps comme échelle.

II.4.2. Analyse .DISTO :

L'analyse de distorsion avec l'instruction .DISTO effectue une analyse harmonique et produit deux tracés DISTO si f2overf1 n'est pas spécifié :

Un pour la fréquence harmonique $2f_1$ et un pour la fréquence harmonique $3f_1$.

Les deux tracés contiennent des tensions et des courants de nœud AC via des sources de tension indépendantes avec une fréquence comme échelle. Les vecteurs sont complexes et leur dimension dépend de gammes de fréquences spécifiées.

Si f_2 over f_1 est spécifié, l'instruction `.DISTO` effectue une analyse spectrale et produit trois tracés pour les fréquences des produits d'intermodulation : f_1+f_2 , f_1-f_2 et $2f_1-f_2$.

Les trois tracés contiennent des tensions et des courants de nœud via des sources de tension indépendantes avec une fréquence comme échelle, et tous les vecteurs sont complexes.

II.4.2.1. Exemple :

Si l'analyse de la distorsion des petits signaux est spécifiée à l'aide de l'instruction :
`.DISTO DEC 10 1K 10MEG ; f2overf1 non spécifié`

Puis l'énoncé : `.PRINT DISTO V(4)`

Imprime la tension au nœud 4 aux fréquences harmoniques $2f_1$ et $3f_1$ (deux impressions), où f_1 est la fréquence a balayé de 1K à 10MEG, 10 points par décade.

Si l'analyse de la distorsion des petits signaux est spécifiée à l'aide de l'instruction :

`.DISTO DEC 10 1K 10MEG 0.9 ; f2overf1 spécifié`

Puis l'énoncé : `.PRINT DISTO V(4)`

Imprime la tension au nœud 4 aux fréquences de produit d'intermodulation f_1+f_2 , f_1-f_2 et $2f_1-f_2$ (trois impressions), où f_1 est la fréquence balayée de 1K à 10MEG, 10 points par décade, et f_2 est maintenue fixe à la valeur $f_2=f_2$ over f_1 $f_{start}=0.91K=0.9kHz$.

II.4.3. Analyse .PZ :

L'analyse Pôle-zéro génère des pôles et/ou des zéros de fonctions de transfert comme variables de sortie. Toutes les sorties sont des scalaires complexes (vecteurs de longueur un) avec des types pôle pour les pôles et zéro pour les zéros. Ils sont nommés pole(1), pole(2), ..., et zero(1), zero(2),... .

II.4.3.1. Exemple :

La déclaration : `pz 1 0 3 0 vol pz`

Calcule les pôles et les zéros de la fonction de transfert $v(3)/v(1)$. Pour afficher tous les pôles et zéros, émettez l'instruction : `.print pz all`

Ou, pour afficher les deux premiers pôles et les deux premiers zéros, l'instruction suivante est utilisée : `.print pz pole(1) pole(2) zero(1) zero(2)`

II.4.4. Analyse de réseau .NET (deux ports) :

L'analyse de réseau à petits signaux génère 16 variables vectorielles de sortie complexes :

$S11, S12, s21, S22$ (*S-parameters, type: S-param*)

$h11, h12, h21, h22$ (*h-parameters, type: h-param*)

$Z11, Z12, Z21, Z22$ (*z-parameters, type: z-param*)

$y11, y12, y21, y22$ (*y-parameters, type: y-param*)

Avec la fréquence comme échelle. Les dimensions des vecteurs dépendent de la gamme de fréquences spécifiée.

II.4.4.1. Exemple :

Pour afficher/tracer les résultats du calcul des paramètres réseau spécifiés par l'instruction `:.net v(9,11) vin dec 10 1k 1G`

Les déclarations suivantes peuvent être utilisées :

```
.print net all
```

```
.print net S11 h12 Z22
```

```
.print net mag(S12) ph(y22)
```

```
.plot net mag(Z12) mag(z22)
```

II.4.5. Analyse .TF :

L'analyse de la fonction de transfert génère trois variables de sortie : la fonction de transfert, la résistance d'entrée et la résistance de sortie[12]. Toutes les variables sont des vecteurs réels (vecteurs de longueur 1) du type `no type`.

L'analyse `tf` spécifiée par `:.tfv(5,3) vin`

Génère les variables de sortie suivantes :

- `tf` (fonction de transfert)
- `zin_at_vin` (résistance d'entrée à vin)
- `zout_at_5_3` (résistance de sortie aux nœuds 5 et 3)

Pour afficher les résultats, l'une des deux instructions suivantes peut être utilisée :

```
.printtf all
```

```
.printtfzfzin_at_vin zout_at_5_3
```

L'analyse `TF` spécifiée par l'instruction :

```
.tfv(4) iin
```

Génère les variables de sortie suivantes :

- `tf` (fonction de transfert)
- `zin_at_iin` (résistance d'entrée à iin)
- `zout_at_4` (résistance de sortie au nœud 4)

L'instruction suivante peut être utilisée pour afficher les variables de sortie :

.printtfz in_at_iiin zout_at_4

II.4.6. Analyse .NOISE :

L'instruction *.NOISE* produit deux tracés :

- *NOISES* : Pour les courbes de densité spectrale de bruit.
- *NOISET* : Pour le bruit intégré total sur la plage de fréquences spécifiée.

Le tracé *BRUITS* contient au moins trois vecteurs réels :

- *frequency* : de type fréquence pour une échelle.
- *noise_s* : de type *noise-spectrum* pour les densités spectrales de bruit total en sortie.
- *inoise_s* : de type *inoise-spectrum* pour un bruit d'entrée équivalent.

La longueur des vecteurs dépend de la plage de fréquences spécifiée et du paramètre *INTER*

Le tracé *NOISET* contient au moins trois scalaires (vecteurs de longueur 1) :

- *number* de type *notype* pour l'échelle.
- *noise_t* de type *noise-integrated* pour le bruit de sortie intégré total.
- *inoise_t* de type *noise-integrated* pour le bruit d'entrée équivalent intégré.

Si le paramètre *INTER* est spécifié dans l'instruction *.NOISE*, les deux tracés contiennent un certain nombre de vecteurs et de scalaires supplémentaires, représentant la contribution de chaque source de bruit de périphérique à la sortie totale ou au bruit intégré. Les noms vectoriels/scalaires des dispositifs passifs (résistances, interrupteurs commandés en tension et interrupteurs commandés en courant) sont créés par :

noise_s.devicename, noise_t.devicename, inoise_t.devicename

Le nom vectoriel/scalaire des appareils actifs est composé du suffixe supplémentaire *noisetype* :

noise_s.devicename.noisetype,

noise_t.devicename.noisetype,

inoise_t.devicename.noisetype.

Si le paramètre *RX <=val>* est spécifié dans l'instruction *.NOISE*, les tracés *NOISES* contiennent un certain nombre de vecteurs supplémentaires, représentant les coefficients de la fonction de transfert vers la sortie pour la source de bruit correspondante dans le circuit. Les noms de vecteurs pour tous les appareils sont créés par la même règle, mais le préfixe *RX.* est utilisé à la place de *noise_s.* : *RX.devicename, RX.devicename.noisetype.*

Les valeurs des *noisetype* est définie dans le tableau ci-dessous.

Dispositif	Type de bruit	Sens	RX = val
BJT	ib	Bruit dû à ib	1
	1overf	Flicker noise (1/f)	1
	ic	Bruit dû à ic	2
	rb	Bruit dû à rb	3
	rc	Bruit dû à rc	4
	re	Bruit dû à re	5
DIOD	id	Bruit dû à id	1
	1overf	Flicker noise (1/f)	1
	rs	Bruit dû à rs	2
JFET MESFET MOSFET	id	Bruit dû à id	1
	1overf	Flicker noise (1/f)	1
	rd	Bruit dû à rd	2
	rs	Bruit dû à rs	3

Tableau II.1. Types de bruit

II.4.6.1. Exemple :

La déclaration : `.NOISE v(5,3) iindec 10 1 10g`

Ne spécifie pas INTER. Les tracés générés contiendront les vecteurs et scalaires suivants :

`NOISES: frequency, inoise_s_i, onoise_s`

`NOISET: inoise_t_i, number, onoise_t`

Si INTER est spécifié dans l'instruction :

`.NOISE v(2) v1 dec 10 1k 100Meg 10`

Alors les deux tracés générés contiendront de nombreux vecteurs. Le nombre de vecteurs dépend du nombre de dispositifs passifs et actifs dans le circuit.

II.5. Conclusion :

Pour simuler un circuit sous l'environnement Gateway, nous avons trois étapes de base à suivre : Saisie du circuit, simulation et calcul, présentation des résultats.

➤ 1^{ère} étape : Il existe deux solutions pour décrire et éditer le circuit à simuler :

- Description par un schéma : dans ce cas, on exploite l'éditeur graphique « Gateway ».

Une fois le circuit est saisi, et avant d'entamer la simulation, ce simulateur créera un

Fichier « netlist », c'est-à-dire une description du circuit sous forme d'un texte, et un fichier de sauvegarde schématique portant l'extension « .Schlr »

- Description par un fichier texte : Dans ce cas, on passera à travers un éditeur de texte

«SmartSpice». Le fichier doit porter l'extension « .sp » (ne pas mettre l'extension .Txt ou .doc).

➤ 2eme étape : « Smartspice » ou « SmartSpicePro » lisent le fichier d'entrée « .sp » : Soit celui généré par l'éditeur graphique ou celui qui est écrit par l'utilisateur sous forme de texte, puis exécutent les calculs demandés et donne les informations obtenues sur un fichier « .OUT » et stocke les résultats dans un fichier « .raw », destinés au logiciel de tracés de courbes « SmartViewer ».

➤ 3eme étape : Les deux simulateurs « Gateway » et « Smartspice » exploitent les fichiers (.raw) et présentent les courbes demandées sous un interface (SmartViewer).

Chapitre III :

Simulation des schémas

électriques avec Gateway

III.1. Introduction :

Dans ce chapitre, nous abordons un travail d'application pour ce que nous avons expliqué dans le deuxième chapitre, de sorte que nous fournirons deux exemples appliqués. Schéma du VCO (Oscillateur Contrôlé par Voltage) et un amplificateur 1GHz accordé, avec une explication du fonctionnement de chacun des deux circuits, ainsi que la manière de les simuler sur le Gateway.

III.2. Schéma du VCO (Oscillateur Contrôlé par Voltage)

III.2.1. Définition :

Un oscillateur commandé en tension est un oscillateur électronique largement utilisé où la tension d'accord d'entrée détermine la fréquence d'oscillation. La fréquence de sortie de l'oscillateur commandé en tension est soit sinusoïdale soit en dents de scie. Les oscillateurs commandés en tension sont des composants importants dans de nombreuses applications telles que les équipements de brouillage électronique, les générateurs de fonctions et les boucles à verrouillage de phase.

III.2.2. Principe de fonctionnement du VCO

La conception de VCO en 5 étapes a été effectuée avec l'utilisation d'un modèle SmartSpice de niveau 54B. Sim4 (voir Figure III.1). Le circuit peut être utilisé pour générer des signaux d'horloge pour conduire les LPL et d'autres circuits. En utilisant cette technique de conception (manque de courant), le courant est contrôlé à chaque étape de l'oscillateur de boucle, et atteint une fréquence de fonctionnement large. Cette conception fonctionne jusqu'à 350 Mhz et la sortie est installée à 1.4V.

Le fichier de contrôle indique que le circuit fonctionnera deux fois. La première fois, l'entrée de poche de 4 MHz est connectée. En outre, les instructions : **ARAM VDD = 2.2V GND = 0.0V t = 1.0us** est défini de sorte que l'un de ces paramètres peut être modifié tout au long de la conception en modifiant la valeur dans **.PARAM. FAQ**. Puisque $t = 1.0\mu s$ a été défini, les paramètres de temps dans les instructions peuvent être une fonction t plutôt que d'utiliser des temps séparés. Dans l'exemple, les instructions transitoires sont configurées pour fonctionner à une heure de t ou de 1.0.

Après le premier tour, les entrées de poche VIN sont remplacées par une source de tension linéaire pour chaque coupure spécifique de PWL dans les paires de tension temporelle (t, v). Les valeurs sont définies en termes de t et de VDD, respectivement, et une fois la simulation terminée, les courbes sont affichées.

III.2.3. Liste Composent du VCO :

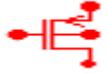
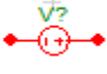
Composent	Symbole
11 Transistors Pmos4.	
11 Transistors nmos4.	
2 résistances.	
2 diodes.	
2 générateurs de tension	
Input vco	
Output vco	
Gnd.	

Tableau III.1. Liste des composants

III.2.4. Connexion les composant

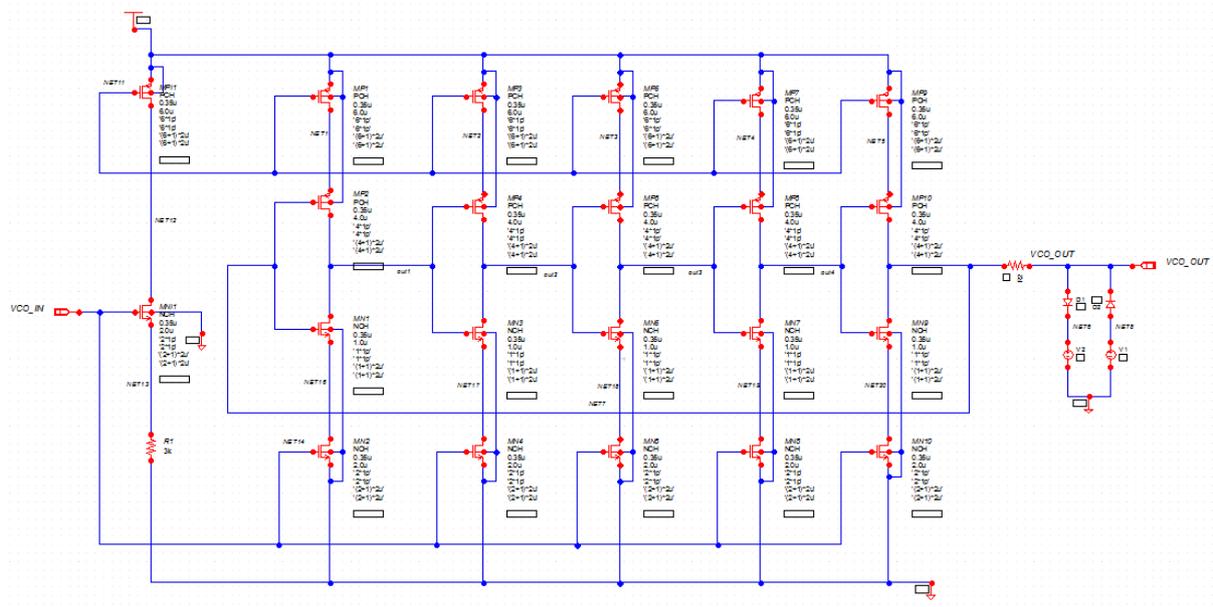
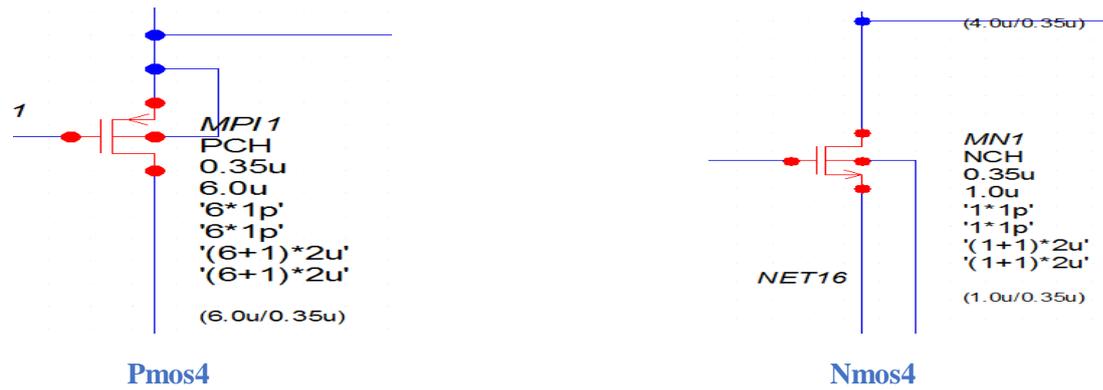


Figure III.1 Schéma VCO



III.2.5. Écriture du modèle de composant :

Il existe plusieurs approches pour créer des symboles. Vous pouvez en créer un à partir de zéro et ajouter des épines, des graphiques et des attributs manuellement. Une deuxième approche consiste à créer un symbole à partir d'un autre en ouvrant un symbole existant, puis en sélectionnant **File**Save As** en enregistrant sous un autre, puis en apportant des modifications. Une troisième approche consiste à créer un nouveau symbole, puis choisir **Tools**Generate Symbol** pour utiliser une interface graphique pour ajouter des graphiques et dimensionner l'empreinte automatiquement. Pour créer un symbole à partir d'un schéma, ouvrez le schéma contenant les broches et cliquez sur **Tools**Generate Symbol**. Enfin, les bibliothèques de symboles peuvent être créées automatiquement à partir des sous-circuits SPICE ou Verilog RTL et des modules comportementaux en utilisant les menus Import SPICE et Import Verilog respectivement.

Pour ajouter, supprimer ou modifier des attributs de symboles, cliquez sur **Edit**Properties...** lorsque le fichier de symboles est ouvert[13].

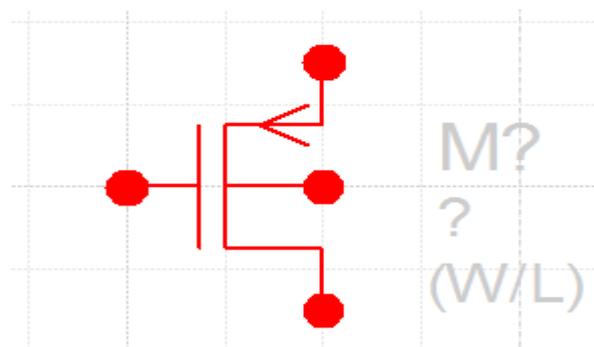


Figure III.2Symbole

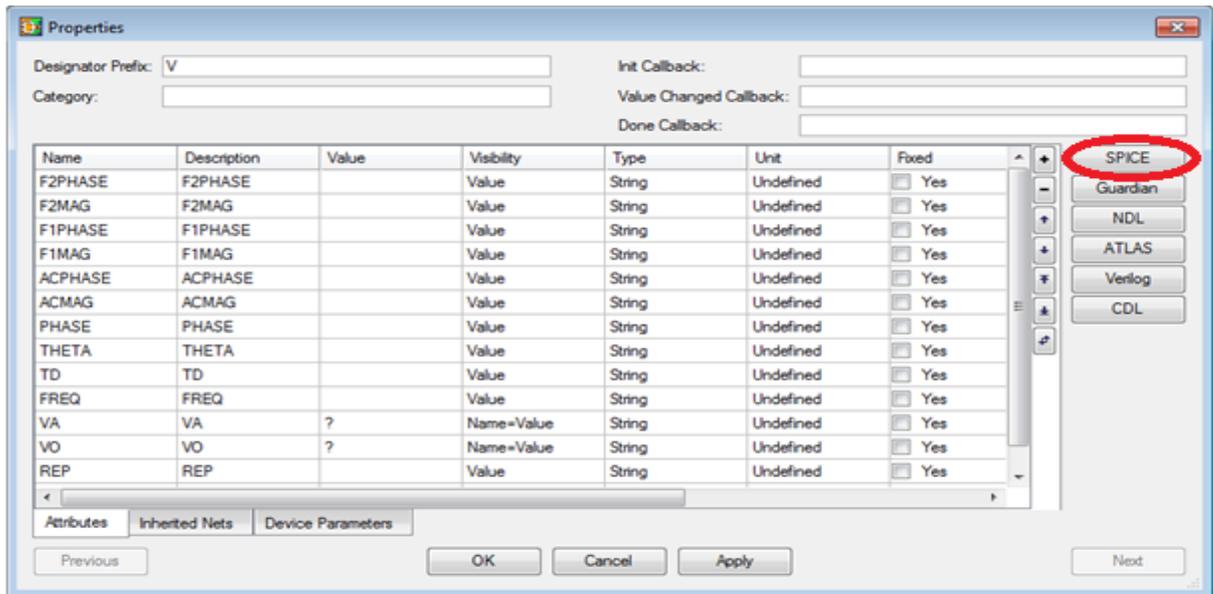


Figure III.3 Boîte de dialogue Propriétés

La boîte de dialogue Propriétés du fichier Symbole comporte cinq champs en haut :

- **Designator Prefix** - Spécifie le préfixe du symbole qui représente son type pour le simulateur
- **Category** - Ceci peut être utilisé optionnellement pour assigner le symbole à une catégorie de symboles, comme MOS, par exemple. Ce champ est ensuite utilisé dans le panneau de symboles de l'éditeur pour le filtrage
- **Init Callback** - Exécute le callback lorsque la boîte de dialogue des propriétés du symbole ou de l'instance est initialisée (remplie).
- **Value Changed Callback** - Exécute le rappel lorsqu'une valeur change dans la boîte de dialogue.
- **Done Callback** - Exécute le callback lorsque la boîte de dialogue/propriété est terminée en appuyant sur **OK** dans la boîte de dialogue.

III.2.6. Traitement et affichage des erreurs :

L'onglet Erreurs de revue affiche les opérations illégales dans l'ordre où Gateway les a trouvées. Pour vérifier les erreurs, cliquez sur l'une des erreurs indiquées dans la fenêtre (voir la Figure III.4).

La Gateway zoomera ensuite sur l'emplacement de l'erreur (si disponible) et mettra en surbrillance le fil, la broche ou le symbole avec l'erreur. La Figure III.4 montre uniquement les erreurs de broche non connectées sans avertissement de l'appareil. Cliquer sur l'une d'entre elles affiche l'erreur sur le dessin. Vous pouvez utiliser les touches fléchées sur le clavier pour monter et descendre à travers les erreurs comme Gateway les affiche sur le

dessin lorsqu'il est sélectionné. Cela permet de corriger les erreurs au fur et à mesure que vous les examinez, y compris en entrant en mode fil et en corrigeant les problèmes de câblage.

La Figure III.5 montre le mode de fil actif lorsque vous corrigez la deuxième erreur. Une fois que vous avez parcouru les erreurs et les corrigé toutes, exécutez une autre vérification de dessin pour voir s'il y a encore des problèmes.

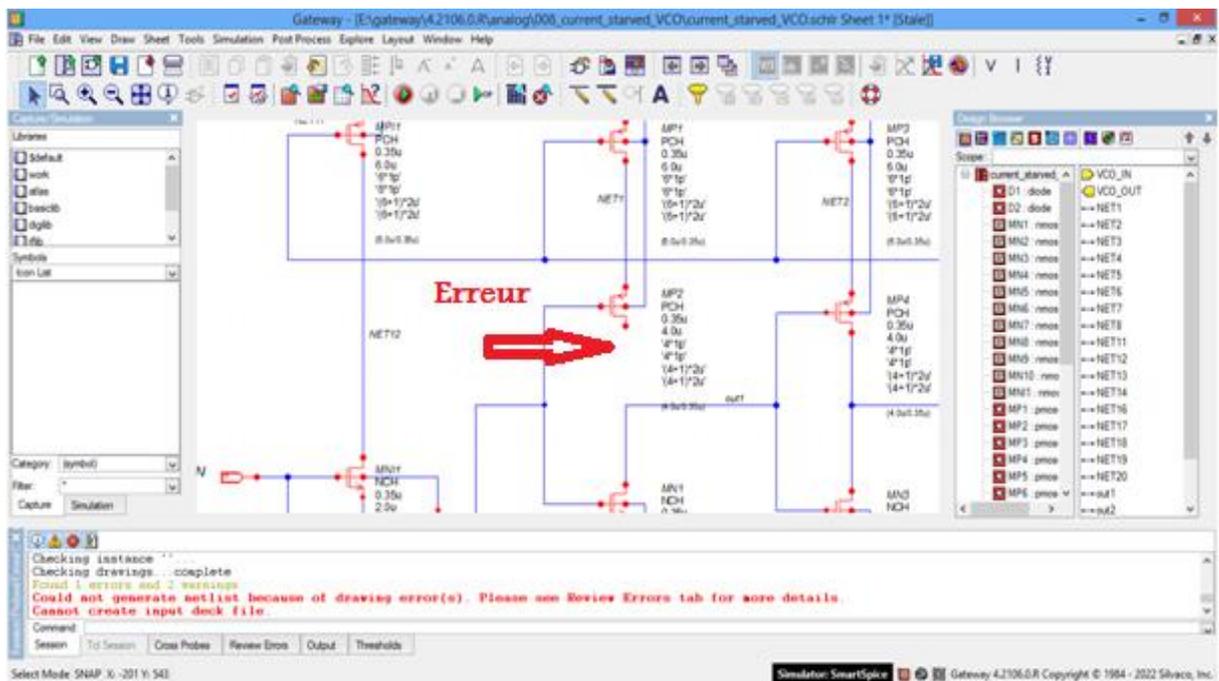


Figure III.4 Affichage des erreurs

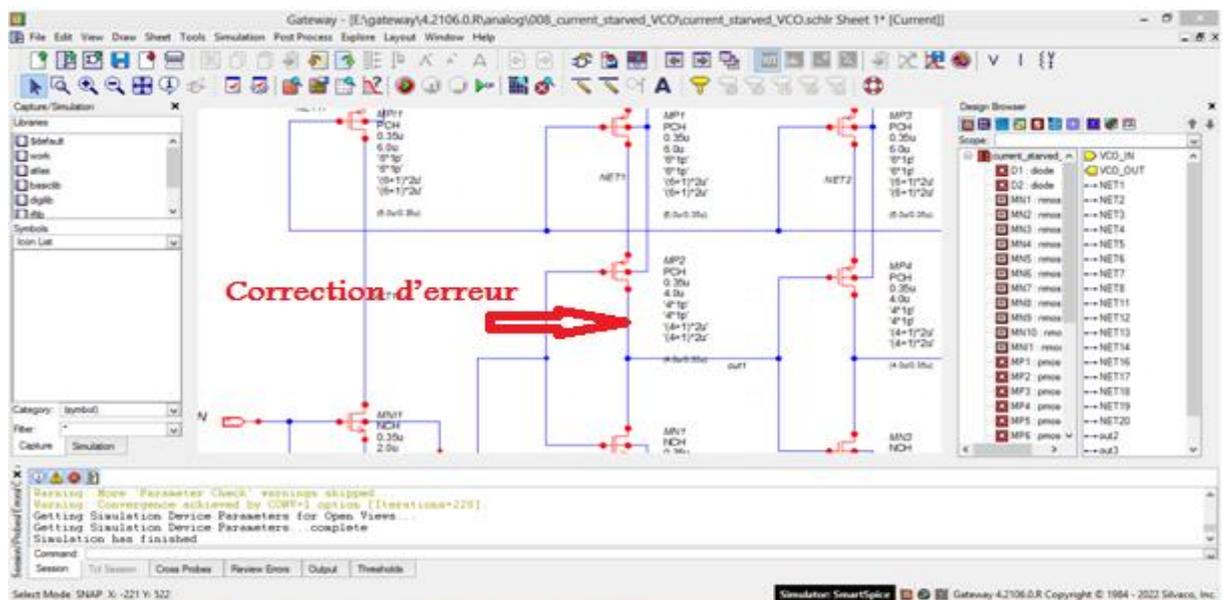


Figure III.5 Mode Fil actif lors de la correction de l'erreur

Il existe deux types d'opérations illégales qui apparaîtront dans la boîte de dialogue Erreurs de revue : Avertissements et Erreurs.

Avertissements (⚠) Les avertissements n'affecteront pas la capacité de netlist le schéma ou la simulation mais montreront des erreurs potentielles dans netlist.

Erreurs (✖) Causera très probablement un dysfonctionnement de Smart Spice, comme empêcher un schéma de simuler ou empêchera la génération de la netlist.

Vous pouvez mettre à niveau les avertissements en erreurs pour permettre à certaines vérifications de conception de signaler les erreurs afin d'éviter les erreurs courantes dans le schéma.

III.2.6.1. Activation et désactivation des avertissements et des erreurs :

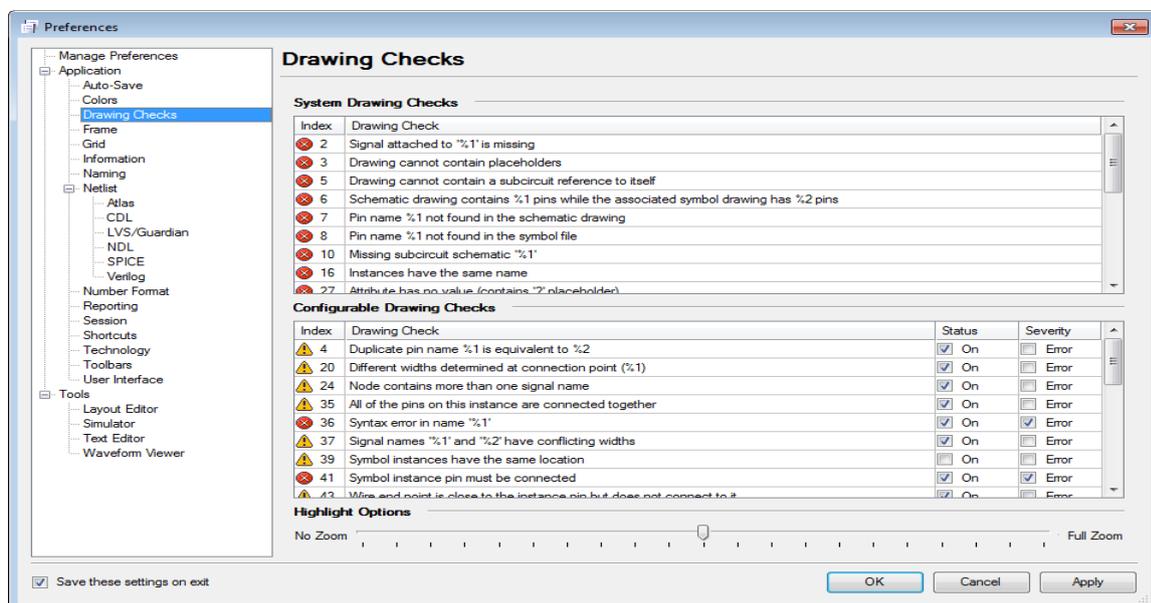


Figure III.6 Vérifications de dessin dans les préférences

La Figure III.6 montre les paramètres des avertissements et des erreurs. La boîte de dialogue comporte deux sections : Vérifications de dessin du système et Vérifications de dessin configurables.

Les vérifications de dessin du système consistent uniquement en des erreurs et ne peuvent pas être rétrogradées en avertissements. Les vérifications de dessin du système sont définies par Gateway et leurs que état est On à temps plein.

Marqué par une icône rouge (✖) Et chacun a un numéro d'index pour référencer l'erreur. Les erreurs empêchent la génération d'une netlist.

Les vérifications de dessin configurables peuvent être des avertissements ou des erreurs, selon la sévérité de chaque vérification de dessin.

Les avertissements sont marqués par () les icônes jaunes. Pour activer ou désactiver un dessin spécifique, cochez la case dans la colonne État. Dans la colonne Gravité, toutes les vérifications sont définies comme des avertissements, à moins que la case ne soit cochée et que l'état de l'erreur ne soit défini. Vous pouvez également désactiver l'état des vérifications configurables à l'aide du volet Erreurs de revue en cliquant avec le bouton droit de la souris sur une vérification de dessin donnée et en sélectionnant Désactiver[13].

III.2.7. Simulation :

Une fois les simulations terminées, les formes d'onde sont affichées (voir Figure III.7).

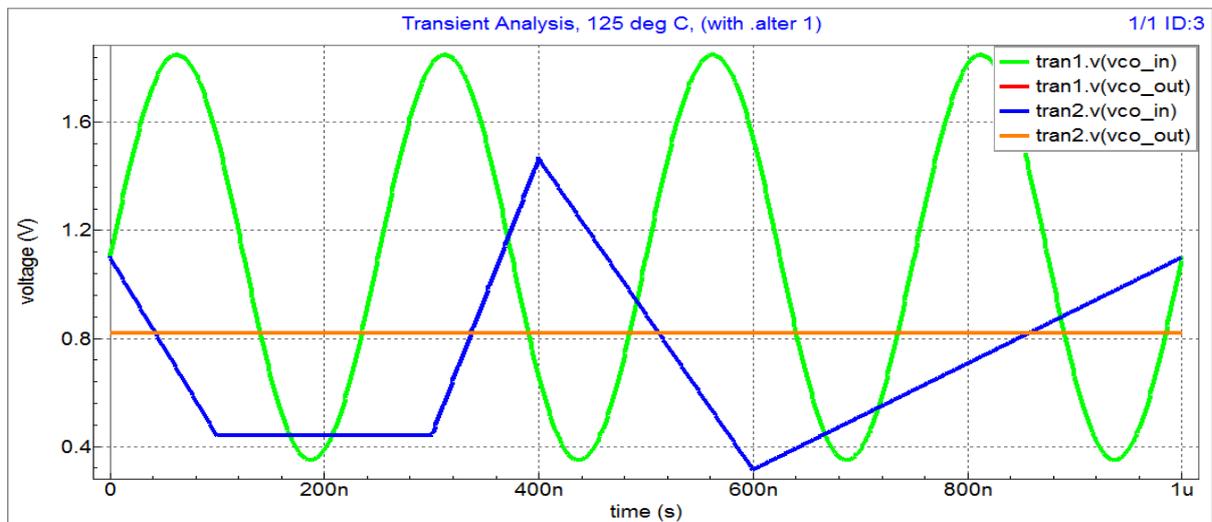


Figure III.7 Graphiques de VCO

L'exécution de la simulation produit les sorties de gain et de phase, comme indiqué. Pour fractionner les formes d'onde afin qu'elles apparaissent sur des graphiques distincts, cliquez avec le bouton droit de la souris et choisissez **Split All/Selected** dans SmartView...

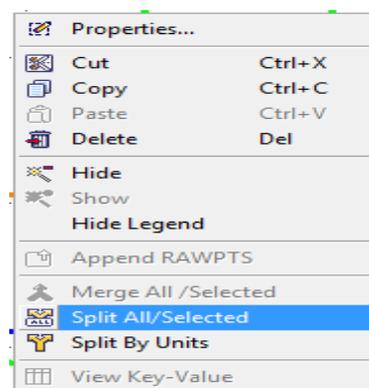


Figure III.8 Liste de Properties.

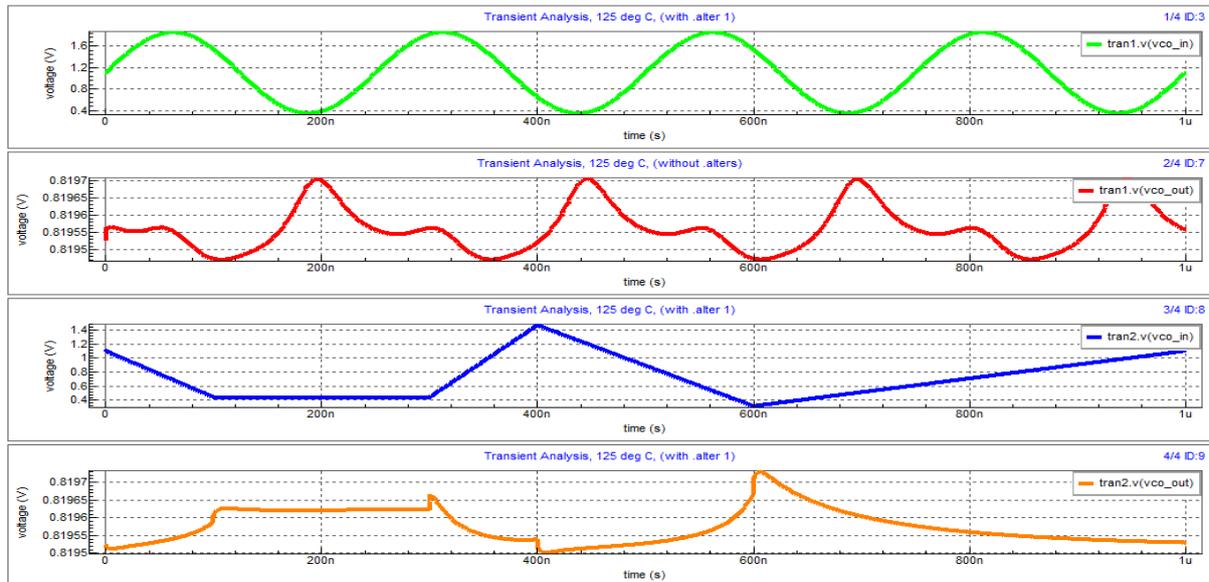


Figure III.9 Graphiques Après avoir appuyé sur Split All/Selected

La fréquence de résonance dans l'oscillateur commandé en tension est contrôlée par la tension d'entrée. Il produit un signal périodique, le contrôle de la tension d'entrée étant lié au niveau de la fréquence du signal. Un élément d'accord appelé diode varactor est utilisé dans les oscillateurs commandés en tension. À l'aide d'une tension de courant continu propre appliquée à la diode varactor, l'oscillateur est réglé pour faire varier la capacité nette fournie au circuit.

Selon la forme d'onde, les oscillateurs commandés en tension sont classés en oscillateurs harmoniques et oscillateurs de relaxation. Les oscillateurs harmoniques ou linéaires sont composés d'un résonateur et d'un amplificateur et produisent une forme d'onde sinusoïdale. Un oscillateur LC-tank est un oscillateur harmonique. Les oscillateurs de relaxation produisent une forme d'onde en dents de scie et ne nécessitent que des composants externes minimes pour produire une large gamme de fréquences opérationnelles. Les oscillateurs harmoniques présentent une meilleure stabilité de fréquence par rapport aux facteurs externes tels que le bruit, la température et la puissance. Il y a aussi une meilleure précision en ce qui concerne le contrôle de fréquence dans les oscillateurs harmoniques. Cependant, ils sont moins adaptés à la technologie des circuits intégrés monolithiques, mais bien mieux adaptés aux oscillateurs de relaxation. De plus, les oscillateurs de relaxation peuvent fournir une large gamme de fréquences opérationnelles.

III.3. L'amplificateur 1GHz :

L'amplificateur électronique se compose de plusieurs parties dont au moins un transistor ou une vanne électronique qui traite un signal analogique de sorte qu'il s'agrandit dans ses qualités et en ressort grossissant. L'une des qualités de la sortie est qu'elle donne une plus grande capacité que celle entrée.

III.3.1. Liste Composent de schéma l'amplificateur 1GHz :

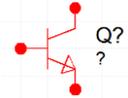
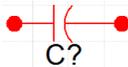
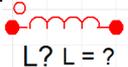
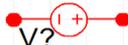
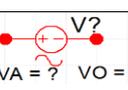
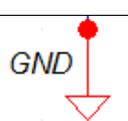
Composent	Symbole
Transistors NPN	
2 Résistances	
4 Condensateur.	
La bobine	
Générateurs de tension VDD	
Générateurs de tension alternatif	
Output	
Gnd.	

Tableau III.2. Liste des composants

III.3.2. Connexion des composants :

Ce schéma (Figure III.10) montre un amplificateur accordé à 1 GHz conçu à l'aide du modèle bipolaire VBIC niveau5 type NPN.

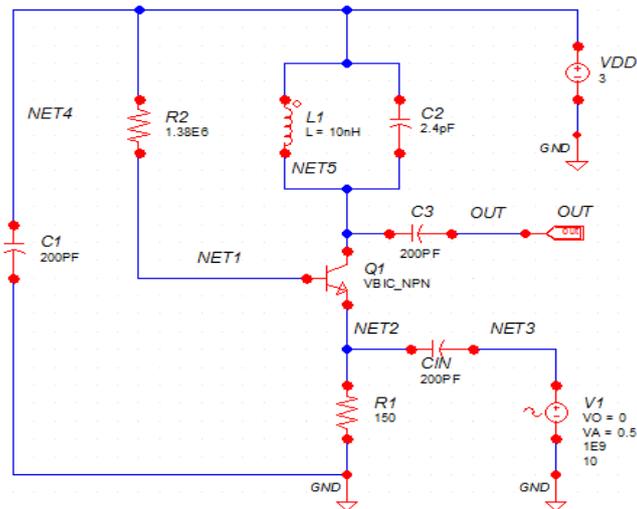


Figure III.10 Schéma de l'amplificateur 1GHz accordé

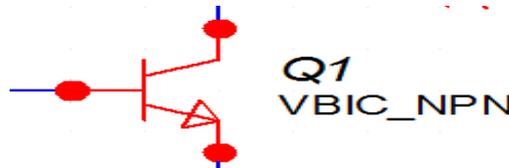


Figure III.11 Symbole de transistors

III.3.2.1. Modèle symbole de transistor :

Le fichier de contrôle affiche les spécificités de l'instruction **AC** et du modèle VBIC. Le nœud OUT est marqué à la fois par des marqueurs de phase et de gain, qui spécifient ces macros de sauvegarde dans le fichier d'entrée ;

III.3.2.2. Fichier de contrôle [12]:

```
.MODEL vbic_npn NPN (
+LEVEL = 5      TNOM = 27
+RCI = 100.4118061  RCX = 21.0073718
+VO = 10      GAMM = 1.612176E-12
+HRCF = 1     RBX = 250
+RBI = 20     RE = 3
+RS = 0      RBP = 0
+IS = 1.019031E-17  NF = 1
+NR = 1.005   FC = 0.5
+CJE = 8E-15  PE = 0.6
+ME = 0.2    CJC = 3.3E-15
+CJEP = 1.54E-15  PC = 0.6
+MC = 0.05   CJCP = 1.540104E-15
+PS = 0.9837837  MS = 0.5893953
+IBEI = 7E-20  WBE = 1
+NEI = 1     IBEN = 1E-16
+NEN = 1.5543186  IBCI = 3.594252E-19
+NCI = 0.996   IBCN = 1.717776E-17
```

```

+NCN = 1.202521   AVC1 = 1E-3
+AVC2 = 1E-5     ISP = 1.332E-19
+WSP = 1         NFP = 1
+IBEIP = 0       IBENP = 0
+IBCIP = 0       NCIP = 1
+IBCNP = 0       NCNP = 2
+VEF = 55.9455555 VER = 2.2052435
+IKF = 0.0350195 IKR = 1.807895E-4
+IKP = 2.908576E-5 TF = 1.001E-16
+XTF = 0.03      VTF = 14.985
+ITF = 0.0999   TR = 2.23542E-9 )
.OP
.AC DEC 100 100MEGHZ 10GHZ
.GLOBAL GND
.save all(i) all(v)
.OPTIONS ACCT=2
.control
set usedegrees=true
.endc
*

```

III.3.3. Simulation :

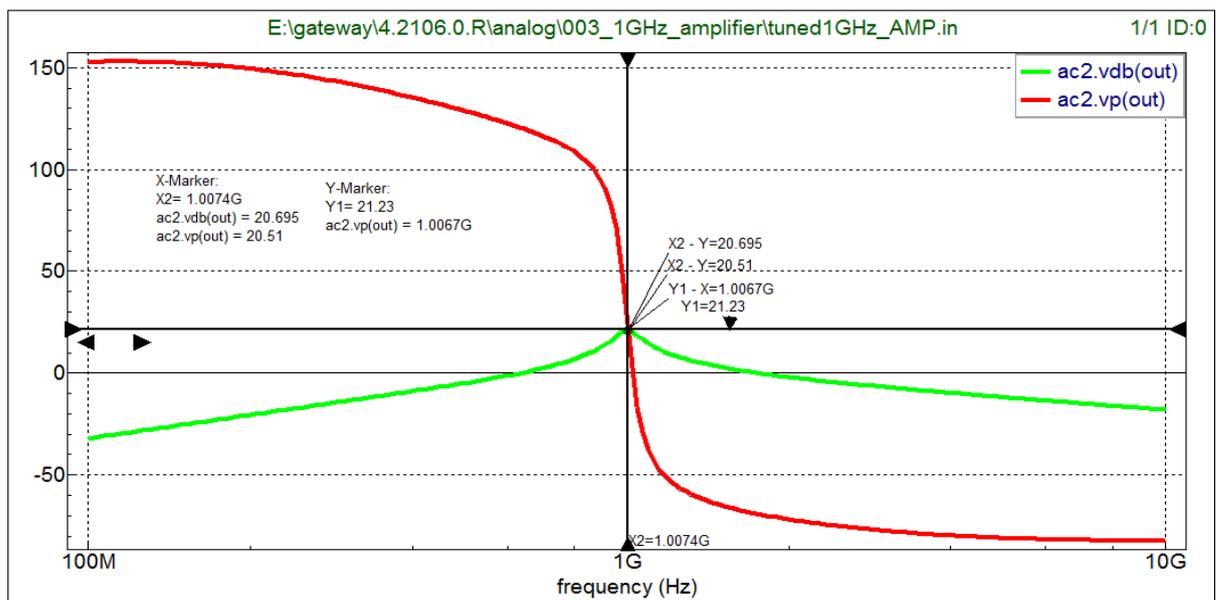


Figure III.12 Graphiques de l'amplificateur 1GHz

Les simulations en cours permettent de progresser et d'obtenir des résultats, comme le montre la Figure III.12 pour observer l'intersection des courbes vdp (OUT) et vp (OUT) au point 1GHz

III.4. Conclusion :

Pendant les simulations, Gateway s'appuie sur SmartView pour afficher les courbes et vous permet de les étudier clairement et de prendre les résultats de manière correcte. Gateway est l'une des dernières versions de Spice afin qu'il soit facile pour les ingénieurs et les étudiants de travailler dessus.

Conclusion générale

Conclusion générale :

Face à la réduction permanente des temps de gestation pour les projets industriels, la miniaturisation et la complication galopante associée, les outils de CAO multi-domaines, multi-signaux et multi-abstraction sont devenus une nécessité pour améliorer la productivité et réduire les coûts de développement grâce au prototype virtuel.

Les outils de CAO analogiques ont besoin évoluer pour rattraper le temps perdu. La conception numérique est maintenant grandement automatisée. Cette automatisation est rendue possible par le développement de langages de description du matériel et au custom Integrated circuit (IC). Cette étude, nous a permis de:

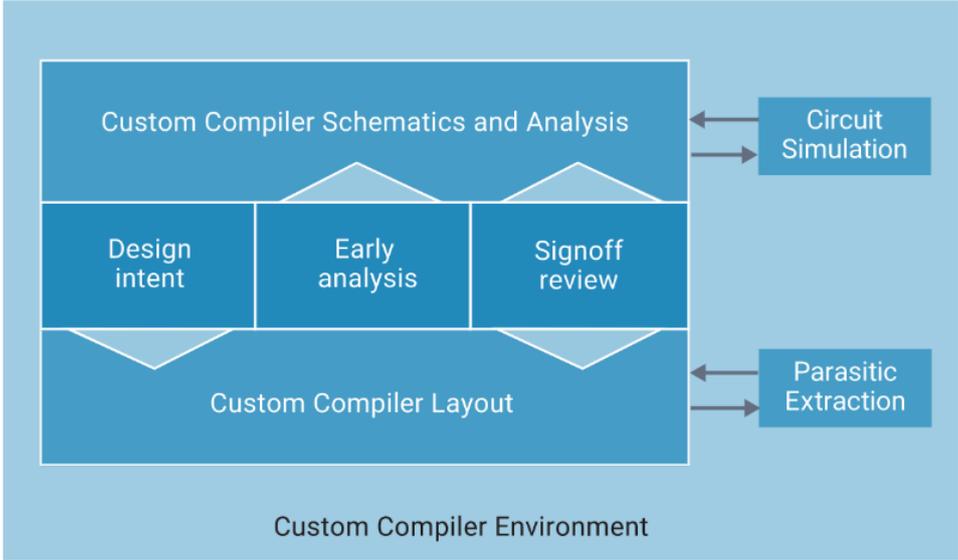
- Savoir dessiner un schéma et configurer une simulation sous l'outil d'éditeur schématique

« Gateway » et le simulateur Spice « SmartSpice ».

- Savoir simuler en utilisant le programme Spice par le biais du simulateur « SmartSpice ».

Lors de cette étude, nous avons montré que l'outil Gateway et SmartSpice de Silvaco permet de posséder un véritable labo de mesure et d'essai virtuel, à moindre coût, et dans un espace restreint. De plus nous sommes arrivés à mettre en évidence la concordance des résultats obtenus par simulation avec les résultats théoriques et pratiques. A partir de la création du schéma d'un montage dont les composants sont décrits par les modèles fournis en général par les fabricants de semi-conducteurs, Gateway permet de simuler le fonctionnement du circuit et de l'analyser finement, tant du point de vue des grandeurs continues que dans le domaine temporel ou fréquentiel. On peut dire que l'outil regroupe en même temps les fonctionnalités de mesure du multimètre, de l'oscilloscope et de l'analyseur de spectre. Sa puissance ne s'arrête pas là, on peut réaliser des analyses paramétriques, c'est -à-dire représenter la variation d'une grandeur du circuit en fonction de la variation d'une autre, pas nécessairement le temps ou la fréquence. Il travaille en alternance et en interaction avec d'autres outils qui le complète pour enfin réaliser le dessin physique de masque (Layout) comme le montre cette figure ci-dessous.

Enfin nous pouvons dire que ce travail nous a permis de comprendre et de savoir exploiter ce logiciel puissant qui nous a émerveillés.



Environnement et le flow de conception

Référence :

1. Defrance, N., *Caractérisation et modélisation de dispositifs de la filière nitrure pour la conception de circuits intégrés de puissance hyperfréquences*. 2007, Lille 1.
2. *Chapter 7 - Using The spice Circuit Simulation Program*. 2022.
3. Berthiau, G., *La méthode du recuit simulé pour la conception des circuits électroniques: adaptation et comparaison avec d'autres méthodes d'optimisation*. 1994, Châtenay-Malabry, Ecole centrale de Paris.
4. Tap, H., et al., *De la conception à la fabrication de circuits intégrés en technologie CMOS*. 2019. **18**: p. 1019.
5. AHMED LAMINE, M., *SIMULATION DU TRANSISTOR MOS PAR PSPICE*. 2016, UNIVERSITE DE MOHAMED BOUDIAF M'SILA FACULTE DE TECHNOLOGIE.
6. Chekour, M., M. Laafou, and R.J.E. Janati-Idrissi, *Vers l'introduction du simulateur Pspice dans l'enseignement de l'électricité: Cas du Tronc commun Sciences*. 2015. **175**.
7. Nagel, L.R., Ronald %J IEEE Journal of Solid-State Circuits, *Computer analysis of nonlinear circuits, excluding radiation (CANCER)*. 1971. **6**(4): p. 166-182.
8. Nagel, L. and D.O. Pederson, *SPICE (simulation program with integrated circuit emphasis)*. 1973.
9. Logvinov, V.V. and S.M. Smolskiy, *The MicroCap12 System of Circuit Modeling*, in *Radio Receivers for Systems of Fixed and Mobile Communications*. 2022, Springer. p. 285-324.
10. Roza, C. and B. Kahina, *Etude du logiciel de simulation pspice et ses application en électronique*. 2010, Université Mouloud Mammeri.
11. González, M.L.J.S.L.d.C., *LTspice*. 2018.
12. Silvaco, *SmartSpice User Manual*. 2022.
13. Silvaco, *Gateway Users Manual*. 2022.
14. GAZAIX, S.J.B.d.I.U.d.P., *La simulation électronique avec SPICE*. 1995(771): p. 373-385.