

MINISTERE DE L'ENSEINEMENT SUPERIEUR ET  
DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS -SETIF-

UFAS (ALGERIE)

**MEMOIRE**

Présenté à la Faculté de Technologie

Département de l'Electronique

Pour obtention de Diplôme de

**MAGISTER**

**Option : Instrumentation**

Présenté par

**Mr : KOUZRIT Daoud**

**THEME**

**Approche de test de circuit analogique par la technique  
BIST**

Soutenu le juin 2013 devant la commission d'examen

MR. BOUZIT NACERDINE Professeur a l'université de Sétif	Président
MR. CHEMALI HAMIMI Maitre de conférence a l'université de Sétif	Examineur
MR. AMARDJIA NOUREDINE Maitre de conférence a l'université de Sétif	Examineur
MR. BOUROUBA NACERDINE Maitre de conférence a l'université de Sétif	Encadreur

# ***Remerciements***

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

Je tiens à exprimer ma reconnaissance à mon directeur de thèse, Monsieur Nacerdine Bourouba Maître de conférences à l'université de Sétif pour tous les conseils précieux qu'il m'a donnés, pour le temps qu'il a consacré pour diriger cette thèse. Il m'a vraiment aidé à mieux comprendre le sujet de test qui fait partie d'un domaine de recherches récemment entamé.

Je remercie chaleureusement Monsieur Bouzit Nacerdine professeur à l'université de Sétif d'avoir accepté de présider le jury de cette thèse. Par ailleurs, je lui dois reconnaissance pour son dévouement en tant que Chef de département pour le bien de la formation.

Mes vives remerciements vont à Monsieur Chemali Hamimi Maître de conférences à l'université de Sétif pour m'avoir fait l'honneur de participer au jury de ce mémoire. Sans oublier sa contribution en tant que mon enseignant pour une formation solide en électronique.

J'adresse mes remerciements les plus sincères à Monsieur Amardjia Nouredine pour avoir accepté de faire partie de ce jury de mémoire.

Je remercie tout particulièrement et sincèrement Monsieur Gamoura Amar et Mohammed Merabet qui m'ont aidé et apporté leurs soutien et connaissances surtout dans la partie de la Simulation.

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.

## Résumé :

En numérique, les techniques de test ont conduit à un soubresaut éminent en matière de qualité de l'opération de test qui se caractérise par sa simplicité, efficacité et rapidité. Parmi ces techniques on cite la fameuse BIST qui fut même implémentée avec le circuit à tester le rendant ainsi auto testable. Par contre en analogique, le test est plus complexe de par sa nature diversifiée en termes de paramètres. Malheureusement cette tâche est délicate et consommatrice de temps. Un souci fort majeur domine les esprits des chercheurs afin de solutionner ces problèmes.

L'objectif fixé à travers le thème de notre travail est de tester un circuit analogique par la technique BIST. L'importance donnée ici en premier est de réussir ce test de ce circuit par cette technique de test tout en préservant sa structure de base qui sert à la génération de vecteur de test, de l'analyse de signature et de prise de décision sur la présence et l'absence de fautes. Le circuit sous test n'est autre que l'ampli-inverseur d'usage commun conçu à base d'un amplificateur opération de type 741. Le type de fautes pris en compte dans cette étude est celui des fautes dures appelées aussi fautes catastrophiques provenant des dégradations physiques fréquentes des transistors de technologie CMOS et bipolaires. Notre circuit soumis à la présence de ce genre de défaillance est simulé par le biais du SIMULINK afin de constituer un récipient de données d'analyse de ses réponses.

Mots clés : technique BIST, circuit analogique, défaillance, transistor bipolaire

# Chapitre 1 : Introduction au test

Introduction générale .....	1
1-1 Introduction sur le Test .....	4
1-2 Étapes de fabrication d'un circuit intégré (IC) .....	4
1.2.1 Le coût de test .....	5
1.2.2 Les phases de test .....	6
1-3 Les différents types de tests .....	6
1.3.1 Test paramétrique .....	7
1.3.2 Test fonctionnel .....	7
1-4 Les défauts globaux et locaux .....	8
1.4.1 Les défauts globaux et leurs effets associés.....	8
1.4.2 Les défauts locaux et leurs effets .....	9
1-5 Concepts de base du test analogique .....	9
1.5.1 Les défauts et les fautes analogiques.....	9
1.5.2 Sources de fautes analogiques .....	9
1.5.3 Détection des pannes et la couverture de fautes .....	13
1.5.4 Contrôlabilité et observabilité .....	13
1.5.5 Les fautes indétectables .....	14
1.5.6 La couverture de fautes .....	14
1-6 Techniques de conception en vue de test (Design for testabilité D.F.T) .....	15
1.6.1 Approche structurale: techniques de Scan-Path (le balayage).....	17
1.6.2 Technique LSSD (Level Sensitive Scan Design) .....	17
1.6.3 Technique de Scan-Path Partial .....	18
1-7 La technique BIST .....	18
1.7.1 Introduction au BIST .....	18
1.7.2 La structure générale de la technique BIST.....	19
1.7.3 Le générateur des signaux de test (TPG).....	20
1.7.4 Analyseur de signaux de réponse (ORA).....	22
1.7.5 Exemple de signature de fautes d'un circuit numérique à l'aide de la technique BIST .....	23
1-8 Conclusion .....	25

## Chapitre 2 : La simulation et son apport dans le test de circuit électrique

2.1 Introduction .....	26
2.2 L'intérêt de la simulation des fautes .....	26
2.3 Test ou diagnostique .....	26
2.3.1 Diagnostic des défauts .....	27
2.3.2 L'origine de défauts dans les circuits intégrés .....	29
2.3.3 Classification et types de défauts .....	30
2.4 Mécanisme de défaillance .....	32
2.4.1 Les défauts des semi-conducteurs .....	32
2.4.2 Modèle de faute au niveau des composants : le court-circuit et le circuit ouvert.....	33
2.4.3 Modélisation des défauts dans les circuits analogiques intégrés .....	34
2.4.4 Modèles des défauts en analogique .....	34
2.4.5 La technique de modélisation employée .....	35
2.5 Macro modèles de fautes .....	35
2.5.1 Macro modélisation numérique .....	36
2.5.2 Modélisation numérique des défauts des circuits analogiques.....	36
2.6 Exemple du modèle numérique pour circuit analogique particulier.....	36
2.7 L'approche D.C. pour la Génération du Dictionnaire de Défauts .....	38
2.8 La simulation des fautes.....	38
2.9 L'emploi de modèle de faute logique (stuck-at) .....	41
2.10 La simulation des fautes par emploi des techniques analogiques .....	42
2.11 le type de modèle employé dans notre travail .....	44
2.12 Conclusion.....	44

## Chapitre 3 : La simulation

3.1 Techniques de simulation .....	45
3.1.1 .Simulation logique.....	45
3.1.2 Simulation analogique .....	46
3.1.3 La simulation mixte logique-analogique .....	47
3.2 Procédure interne d'un simulateur .....	48
3.3 Les trois analyses basiques de la simulation électrique .....	49

3.3.1	L'analyse statique .....	49
3.3.2	L'analyse Transitoire .....	50
3.3.3	L'analyse alternative petit signal .....	50
3.4	Les différents types de Simulateurs .....	50
3.4.1	Simulateurs Mixtes, Langages de modélisation numérique .....	50
3.4.2	Le VHDL .....	51
3.4.3	Langage de modélisation Mixte Multi-Domain (VHDL-AMS) .....	52
3.4.4	VHDL-AMS .....	51
3.5	Introduction à MATLAB .....	53
3.5.1	MATLAB .....	53
3.5.2	Exemples .....	54
3.5.3	Quelques fonctions mathématiques et les opérations arithmétiques.....	55
3.6	La partie SIMULINK .....	56
3.6.1	Introduction .....	56
3.6.2	Démarrage de Simulink .....	56
3.6.3	Création de sous systèmes .....	60
3.6.4	Réglage du temps de simulation .....	60
3.6.5	Exemple d'application .....	61
3.7	Conclusion .....	62

## **Chapitre 4 : application de la technique BIST sur l'Ampli-Op**

4.1	INTRODUCTION .....	63
4.2	Description de la structure sous test.....	63
4.2.1	L'amplificateur opérationnel le $\mu A$ 741.....	63
4.2.2	Amplificateur en mode inverseur.....	65
4.2.3	Structures internes de L'amplificateur $\mu A$ 741 .....	66
4.2.4	Structure des transistors utilisés.....	68
4.3	Création du circuit électronique.....	71
4.3.1	La structure interne de $\mu A$ 741 sous Simulink .....	72
4.3.2	Insertion de caractéristiques FERRANTI.....	73
4.4	Simulation de $\mu A$ 741 sous Simulink.....	74
4.5	Comparaison entre SIMULINK et PSPICE .....	76
4.6	Simulation de la technique BIST.....	78
4.6.1	La configuration numérique de la technique .....	78

4.6.2	Vérification fonctionnels de chaque unité .....	79
-	Générateur de signaux de test (TPG) .....	80
-	Analyseur de vecteur de réponse (ORA) .....	80
-	Les circuits mémoires .....	81
-	Unité de comparaison .....	82
4.6.3	Exemple d'application numérique .....	83
4.6.4	Insertion de faute dans le circuit combinatoire.....	85
4.7	Circuits d'adaptation entre l'Analogique et le Numérique.....	85
4.7.1	Conversion numérique analogique avec Simulation .....	86
4.7.2	Conversion Analogique Numérique .....	87
-	La conversion de la partie entier .....	88
-	La conversion de la partie décimale d'un chiffre .....	89
-	Le signe .....	90
4.8	Contrôleur de test .....	91
a-	Initialisation .....	91
b-	L'horloge CLK et remise à zéro CLR.....	92
c-	Signal de sélection .....	92
4.9	Simulation de processus de test numérique pour un circuit analogique .....	93
4.10	Description de signaux .....	93
4.10.1	Les Signaux de commande .....	94
4.10.2	Les signaux à traiter .....	95
4.10.2.1	Les tensions $V_s$ et $V_e$ .....	95
4.10.2.2	La tension $V_s$ en binaire .....	95
4.10.2.3	La signature .....	96
4.10.3	Comparaison avec la théorie .....	97
4.11	Test par BIST de circuit fautif.....	98
4.11.1	Choix de la faute particulière à injecter .....	98
4.11.1.1	La faute de circuit ouvert (BOQN10).....	98
a-	Simulation par PSPICE .....	98
b-	Simulation par SIMULINK.....	99
4.11.1.2	La faute de court circuit (CCBEQP2).....	99
a)	Simulation par SIMULINK .....	100
b)	Simulation par PSPICE .....	100
4.11.2	Simulation par emploi de la technique BIST.....	100
4.11.2.1	La première faute (BOQN10) .....	101

4.11.2.2	Le cas de la deuxième faute CCBEQP2 .....	103
4.12	Conclusion .....	105
	Conclusion générale .....	106
	Bibliographie .....	108

# ***Introduction générale***

Ces dernières années plusieurs nouveaux domaines émergeant en sciences et technologie sont devenus très importants dans la vie quotidienne, tels que la télécommunication, le multimédia, les applications biomédicales, etc. Ces domaines nécessitent une utilisation importante des circuits intégrés analogiques et mixtes. Tester ces derniers cause un impact considérable sur le coût de production. Ceci est dû à la complexité des circuits par leurs fonctions et leur taille qui ne cessent de s'accroître. En outre, la tendance d'intégrer des systèmes analogiques/numériques complets dans une seule puce pose de très sérieux problèmes d'accès pour le test [1].

Le test des circuits intégrés est devenu une tâche essentielle dans l'industrie des semi-conducteurs. Ces circuits peuvent contenir dans une seule puce de silicium un nombre de 1 million et plus de transistors intégrés, cela rend le test plus compliqué et plus difficile. Il en résulte des mises en garde des affrontements contre différents fronts technologiques à savoir :

- La complexité imposée par cet événement,
- L'apport de l'outillage sophistiqué et les techniques adéquates qui se déploient à travers les différentes phases de fabrication de ces circuits.
- L'unification de technique de test pour les deux types de circuits.

En urgence à ces problèmes, il fut d'abord l'adoption du concept '**diviser pour régner**' pour résoudre le problème de complexité. Il consiste à la répartition des circuits complexes en entités élémentaires et de séparer l'analogique du numérique. Beaucoup de techniques de test apparaissent dans ces deux types de circuits et qui tiennent place au niveau fonctionnel que structurel. L'objectif est de vérifier le bon produit du mauvais, d'améliorer la qualité de celui-ci et bien sûr de rendre lucrative la production des circuits.

En numérique, les techniques de test ont conduit à un soubresaut éminent en matière de qualité de l'opération de test qui se caractérise par simplicité, efficacité et rapidité. En analogique, le test analogique est plus complexe de par sa nature et plus diversifié en termes de paramètres par rapport au précédent. Ceci rend malheureusement la tâche de test délicate et

consommatrice de temps. Un souci fort majeur domine les esprits des chercheurs afin de solutionner ces problèmes

Les travaux de recherche concernant le test des circuits analogiques ont été basés traditionnellement sur le diagnostic des fautes. Mais l'accroissement rapide du niveau d'intégration des puces de nouvelle génération est considéré comme un problème difficile pour non seulement diagnostiquer les fautes mais aussi distinguer les bons circuits des mauvais. Des blocs analogiques intégrés dans des systèmes numériques ne peuvent pas être facilement testés [2].

Par conséquent, de nombreuses techniques ont été proposées afin de faciliter la tâche de test des circuits analogiques et mixtes. Ces techniques visent l'optimisation du test lors de la fabrication (test de production) ou lors de l'application (test en ligne/test hors-ligne). Certaines techniques permettent d'ajouter des circuits dans la puce pour faciliter cette opération ou même réaliser un autotest. Quelques tentatives timides consistent en l'emploi de techniques numériques basées sur des modèles logiques tels que les collages à 0 et à 1 (stuck at 0 and 1) ont été probants pour certains circuits analogiques [46]. Malheureusement, certaines difficultés apparaissent pour certains circuits spécifiques tels que les oscillateurs en anneaux [48] et concernent l'établissement de modèles logiques équivalents pouvant reproduire approximativement les fonctions de ces circuits en absence et en présence de fautes.

Notre objectif fixé à travers le thème de notre travail est de tester un circuit analogique par la technique BIST, qui est largement utilisée dans le domaine des circuits numériques. Elle est incontestablement très efficace dans un processus d'autotest puisqu'elle a permis d'améliorer la détectabilité de fautes dans les circuits numériques. L'importance donnée en premier dans notre travail est de réussir le test de ce circuit par cette technique de test tout en préservant sa structure de base qui sert à la génération de vecteurs de test, de l'analyse de signature et de prise de décision sur la présence et l'absence de fautes. Le circuit sous test n'est autre que l'ampli-inverseur d'usage commun conçu à base d'un amplificateur opération de type 741. Le type de fautes pris en compte dans cette étude est celui des fautes dures appelées aussi fautes catastrophiques. Elles proviennent généralement à des dégradations physiques fréquentes des transistors de technologie CMOS et bipolaires et qui sont modélisées tout simplement par des courts-circuits et circuits ouverts. Notre circuit soumis à la présence de ce genre de défaillance est simulé par le biais du SIMULINK afin de constituer un récipient de

données d'analyse de ses réponses appropriées aux fautes. Cet outil de simulation est utilisé sous trois aspects différents :

- 1- la simulation électrique du circuit sous test ;
- 2- la simulation logique de la structure de test ;
- 3- simulation des accessoires de conversions CAN et CNA par emploi d'opération mathématiques.

### **Organisation des chapitres:**

Le Chapitre 1 est contient deux parties dont l'une donne un aperçu général sur le test, ces différents types et quelques techniques utilisé dans ce domaine. La deuxième partie quant à elle relate la technique BIST (Built in Self Test) adoptée dans notre processus de test dont l'attention est focalisée sur sa structure générale et ses détails, ces différents blocs avec un exemple d'illustration de la structure BIST sur un circuit numérique.

Le chapitre 2 deux porte sur le diagnostic des fautes, les différents types de fautes qui peuvent avoir lieu chez es circuits analogiques et numériques. Les approches de simulations de fautes qui suivent ce processus d'analyse en font partie.

Le troisième chapitre explique les techniques de simulations logique, analogique et mixte qui accompagnent un processus de test ou conception de circuits électroniques. Plus de détails concernant l'utilisation de Matlab/ SIMULINK, la création des blocs, les propriétés de simulation et les différents bibliothèques sont étalés dans cette partie d'étude.

Le quatrième Chapitre est réservé à la simulation et les résultats obtenus sur le circuit sous test, la structure de test BIST, et sur la procédure de test par cette technique sur ce circuit analogique. Une partie de ce chapitre est dédiée au rappel sur les caractéristiques de l'Ampli-Op inverseur de type Ferranti et la simulation de sa fonction de transfert par SIMULINK.

Enfin au terme de ce travail, une conclusion est donnée présentant les aspects les plus marquants de notre travail.

# **Chapitre 1**

## **Introduction au test**

## 1.1 Introduction au Test.

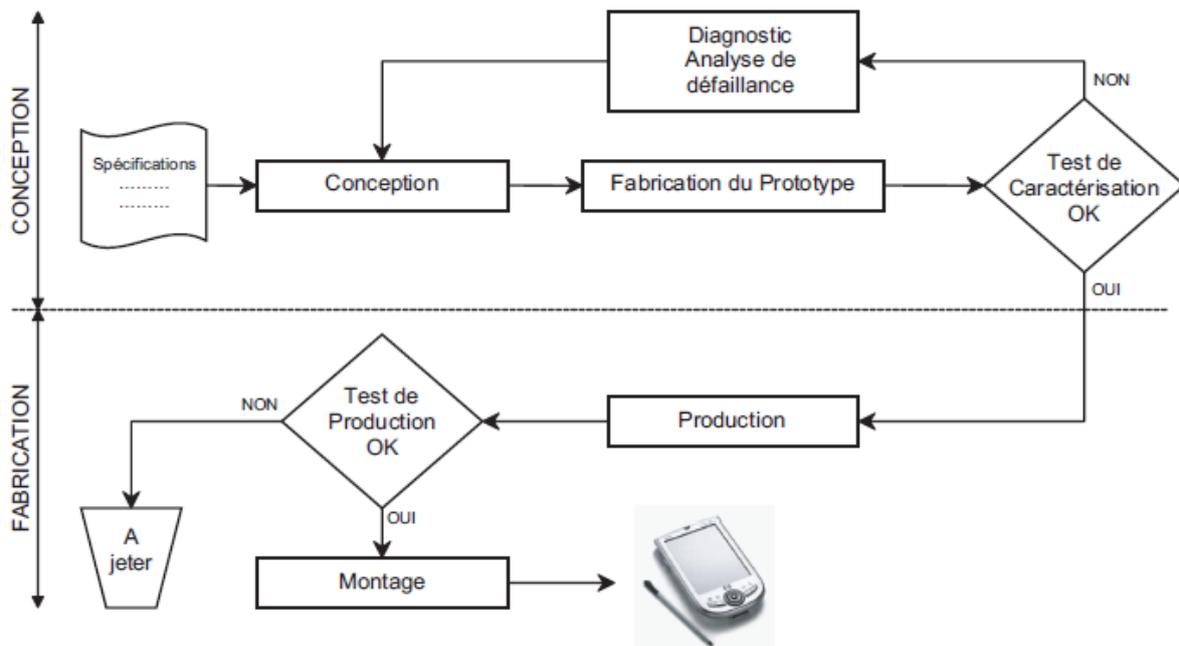
Historiquement, les circuits électroniques étaient simples et par conséquent faciles à tester. Il en est de même pour l'accès à toutes les entrées et sorties pour détecter d'éventuelles fautes, mais avec le développement de la technologie et l'intégration des composants à grande échelle, la taille des circuits est considérablement réduite ainsi que le coût de production. Néanmoins, la détection et la localisation de fautes devient une tâche difficile rendant le test plus onéreux [4], [2].

Il n'est pas surprenant de constater que dans un peu plus de dix ans les coûts de développement de test auront augmenté d'environ 25% à environ 50% du total des coûts de développement de produits. En outre, les machines de test plus coûteuses sont nécessaires pour traiter un plus grand nombre de broches d'E / S, les fréquences de fonctionnement plus élevées, et les grands ensembles de vecteurs de test généralement associés avec les dispositifs VLSI plus complexes.

Les prévisions pour l'avenir indiquent que le problème de test deviendra crucial à l'année 2014 [13] [14]. Par exemple, les machines d'essai pour les principaux dispositifs VLSI coûtera plus de vingt millions de dollars. Le coût associé à l'essai de chaque transistor dans un dispositif VLSI sera plus grand que le coût de fabrication de ce transistor. Certaines des techniques de tests les plus utilisées ne pourront plus répondre aux besoins de test dans un proche avenir. Enfin, en raison de la complexité croissante des dispositifs VLSI et les PCB (circuit imprimés), la capacité de fournir un certain niveau de diagnostic des défauts lors de la fabrication est nécessaire pour faciliter l'analyse des modes de défaillance (FMA : Failure Mode Analysis) et par conséquent l'amélioration du rendement et des procédures de réparation. L'approche BIST est considérée comme l'une des principales solutions à ces problèmes de tests [14].

## 1.2 Étapes de fabrication d'un circuit intégré (IC) [1].

La fabrication d'un circuit intégré (IC) passe par deux étapes principales : l'étape de conception ou de design et l'étape de fabrication ou de manufacture.



**Figure 1.1 : étapes de fabrication de circuit intégré**

Dans chacune des étapes de fabrication (figure 1.1), un test doit être effectué. Dans la première étape, le test à effectuer est appelé test de validation ou test de caractérisation. Il s'agit d'un test fonctionnel devant vérifier si les performances du circuit correspondent à celles prévues (du cahier de charges). Si ce n'est pas le cas, un diagnostic doit être fait pour localiser l'erreur de conception ou de fabrication.

Dans la deuxième étape, le test à effectuer est un test de production. Son but est de séparer les circuits défectueux des circuits fonctionnels. Néanmoins dans cette étape les circuits ne peuvent pas être réparés, ils seront donc jetés. En revanche, le diagnostic est utilisé dans le but d'améliorer le rendement de la chaîne de fabrication [3] [1].

### 1.2.1 Le coût de test.

Le coût total d'un circuit électronique dépend de plusieurs facteurs comme la fabrication, l'emballage, l'assemblage et le test; le coût de test augmente à l'opposé des autres facteurs qui diminuent, le tableau (4) montre la relation entre le coût de détection de la faute en fonction de l'échelle de l'intégration [3] [5].

**Tableau 1. Coût de détection du mauvais fonctionnement des dispositifs en fonction de leur échelle d'intégration**

<b>Device integration level</b>	<b>Cost in \$</b>
wafer	0.01 – 0.1
package	0.1 – 0.3
board	0.3 - 3
system	3 - 30
application	30 - 300

Il est clair que la détection du mauvais fonctionnement du dispositif durant le processus de fabrication avant la phase d'intégration, l'optimisation des essais de production conduisent à une réduction considérable du coût [4].

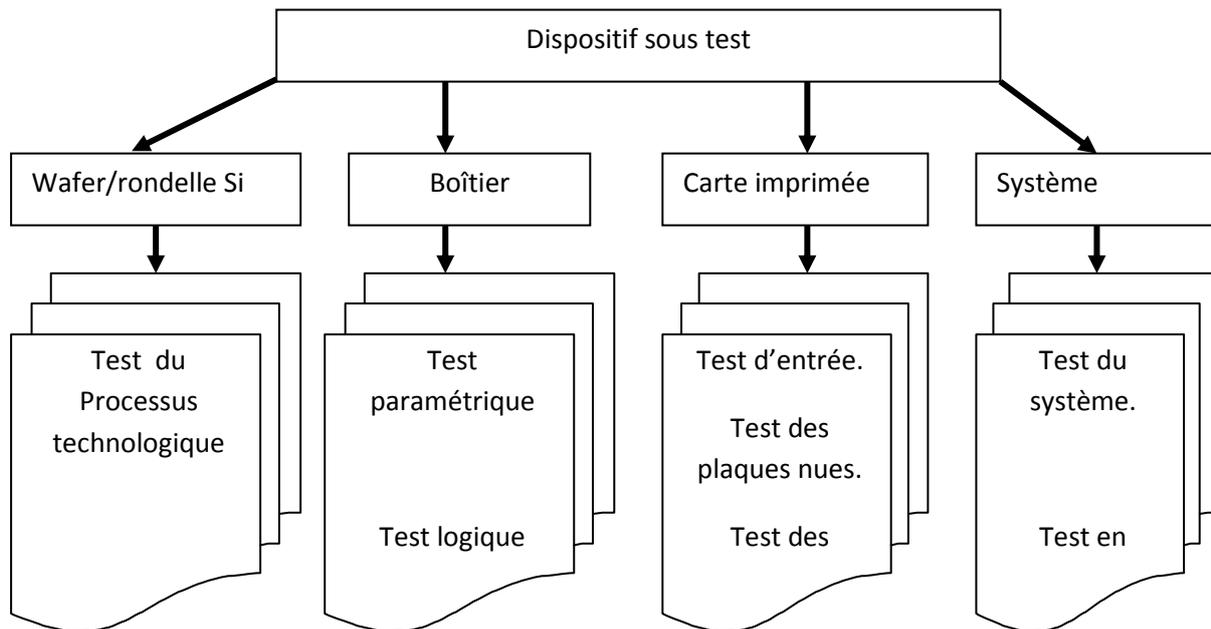
### 1.2.2 Les phases de test [6] .

Il y'a trois phases de test dans un cycle de vie d'un produit où la première est la phase de test de conception visant l'assurance de la fonction prévue du produit. Deuxièmement, il s'agit de la phase de test de fabrication pour détecter s'il y'a des fautes au niveau du circuit intégré, circuit imprimé etc... Finalement, la phase de test de mise en marche qui prend place juste après l'assemblage du système pour déceler les défauts subits lors de cette opération.

La qualité et la quantité de tests au cours du cycle de vie d'un produit dépendent des propriétés du produit, par exemple si le produit est volumineux et moins chère, il n'est pas testé mais systématiquement jeté. Toutefois quand il s'agit d'un système compliqué on doit lui faire subir des tests rigoureux pour s'assurer du bon fonctionnement, et quand la faute est détectée les procédures d'identification et de remplacement des composants défectueux sont entamées.

### 1.2.3 Les différents types de tests .

Les "**puces**" sont testées sur les plaquettes "wafers" durant la fabrication de circuits, lors de leur insertion en boîtiers, après leur encapsulation par le fabricant et avant leur assemblage sur une carte par l'équipementier. Quant aux cartes électroniques, elles subissent un autre type de test avant leur assemblage dans un système qui lui-même fera partie de ce type d'examen une fois fini.



**Figure 1.2 : Différents types de test**

Postérieurement à l'étape où cette tâche est effectuée, les tests appliqués ne seront pas les mêmes puisqu'ils n'ont pas la même finalité et qu'ils s'appliquent dans des conditions tout à fait différentes [6]. En général, chaque puce ou circuit intégré subit deux types de tests [6]:

➤ **Test paramétrique :**

Celui-ci peut se développer en régime continu, et concerne la vérification des courts circuits, des circuits ouverts, le courant de saturation ou le surplus de courant de fuite, et le niveau de seuil (threshold): il s'agit du test paramétrique en mode DC. Un autre test paramétrique touche spécialement le temps de retard de propagation, temps d'établissement, temps de maintien, temps de descente et de montée etc... Les temps d'accès, de pause et de rafraîchissement, temps de montée et de descente : c'est le test paramétrique AC ou dynamique.

Ce genre de test est en général lié à la technologie des composants sous test, par exemple la mesure de tension de sortie d'une structure électronique à base de CMOS n'a pas besoin de courant de charge par contre en technologie bipolaire, le courant de charge est nécessaire [9] [15].

➤ **Test fonctionnel :**

C'est un test ayant rapport avec les vecteurs d'entrée et les réponses d'un circuit ou d'un système électronique. Il permet le contrôle du bon fonctionnement d'une conception vérifiée à travers les différents nœuds internes. Ce test fonctionnel sert en réalité à l'aboutissement d'une

couverture de fautes très élevée. Ces fautes sont en générale modélisées tel que le modèle de collage (stuck- at) et leur génération est primordiale dans un processus de test.

En plus de ce type de contrôle, ce genre de test peut être orienté vers différents objectifs: il cible la vérification des caractéristiques des modules pour des conditions nominales. En outre, il s'applique aussi à des températures élevées afin de garantir les spécifications données dans le cahier de charge: un circuit opérant jusqu'à 70°C (température limite à ne pas dépasser), est porté à 80°C dans un tel test pour garantir la précédente. Il s'agit alors d'une marge de garde. Une autre application mettant à l'épreuve les performances des circuits intégrées est le processus de dégradation accéléré «Speed Binning Process»: ce test agit sur les conditions limites de tensions ou des paramètres temporels tels que fréquence d'horloge, temps de retard etc..... [9] [16]. Ceci ne représente pas l'unique façon de classifier les tests. Un test peut être effectué d'une manière statique ou dynamique. Pour le premier, seulement les caractéristiques de l'état stationnaire, ou DC, du circuit sous test sont intéressantes.

Dans le deuxième, la réponse temporelle est d'une importance capitale pour évaluer une puce. Une sous-classe principale du test dynamique est celle appelée ``**at-speed testing**`` où le circuit est testé à vitesse d'exploitation nominale (ou même maximale).

D'autres classifications peuvent être aussi basées sur la manière dont le vecteur de test est appliqué où nous distinguons le test externe et le test interne (ou autotest). Le premier test nécessite d'utiliser le circuit en dehors de son environnement opérationnel et l'utilisation d'équipements de test externes.

Contrairement au test externe, les techniques de l'autotest sont directement intégrées dans la puce même dans le but d'éviter l'utilisation des équipements de test externes.

Finalement, une autre subdivision permet de distinguer entre les méthodes de test concurrent et non-concurrent dépendant de la manière dont le circuit sous test peut être testé pendant sa phase opérationnelle (ou pendant que le circuit effectue ses fonctions) [6].

### **1.3 Les défauts globaux et locaux :**

#### **1.3.1 Les défauts globaux et leurs effets associés.**

Ce premier type de défauts provient de problèmes liés à la fabrication provoquant une altération au niveau de la topologie ou des spécifications du circuit. Parmi ces problèmes on énumère les conditions instables du processus, la non-homogénéité du substrat, mauvais alignement des masques etc.... Ils sont dits globaux car ils affectent tous les circuits d'une même

puce de silicium et qui peuvent toucher toute la production d'une chaîne de fabrication. Les effets de ce type de défaut se manifestent par des changements plus au moins importants au niveau des valeurs des paramètres. Ces effets sont connus sous le nom de fautes paramétriques [7] [8].

Les fautes paramétriques sont des fautes non catastrophiques (soft fault) comme le définissent certaines bibliographies, entraînant une déviation moins importante sur les paramètres électriques du circuit sous test à savoir le gain, la phase, l'impédance etc.... Ces fautes se caractérisent par des erreurs de tolérance associées aux composants actifs (diodes, transistors, etc..) et composants passifs (résistances, condensateurs, bobines) et elles sont estimées à différents pourcentages (10% et plus) de la valeur nominale [7].

### **1.3.2 Les défauts locaux et leurs effets.**

Ces défauts par contre font leur apparition sur un seul circuit ou une petite région de la puce de silicium et qui proviennent usuellement de la présence de corps étrangers affectant le processus de fabrication. Leur effet apparaît sous une des deux formes suivantes :

- Une altération paramétrique du circuit comme décrit précédemment, il s'agit alors de fautes paramétriques locales comme par exemple la présence de particules dans la structure de transistor à effet de champs entraînant l'élargissement de son canal [6] [9].
- Une altération structurelle affectant les spécifications du circuit qui selon le degré de gravité elle est considérée comme catastrophique ou non.

Les fautes catastrophiques sont généralement les fautes qui provoquent des déviations importantes et en conséquence entraînent un fonctionnement loin des spécifications. Ces fautes de structures sont en général des courts circuits entre les nœuds ou circuit ouverts au niveau de des points de connexion des transistors à effet de champs ou bipolaires ainsi que les résistances, les condensateurs et les bobines. Dans certains cas de figures, ce type de fautes conduit à un fonctionnement du circuit en dehors de la gamme de spécification qui lui est fixée : ils sont appelées fautes non catastrophiques ou paramétriques.

## **1.4 Concepts de base du test analogique**

Il existe quelques concepts qui doivent être définis précisément pour éviter une interprétation confuse du test analogique en théorie et en application. La première définition

devant être considérée avec attention est l'idée du test elle-même, qui est introduite de façon ambiguë dans la littérature.

Une large étude et analyse de ces concepts a été faite. Le test des circuits analogiques est caractérisé par trois tâches principales : la détection, la localisation et le diagnostic. Toutes ces trois tâches nécessitent l'utilisation d'un vecteur de test. Cependant, la détection d'une faute, sa localisation et son diagnostic dépendent de la qualité du vecteur du test.

#### 1.4.1 Les défauts et les fautes analogiques

La mal construction d'un circuit au delà des variations du processus est appelée **défaut** et l'effet des défauts sur les caractéristiques électriques d'un circuit intégré déviant au delà des valeurs spécifiées est appelé **faute**. En d'autres termes, une faute est une conséquence d'un défaut, mais il est possible qu'il n'y ait aucune faute dans un circuit avec défaut. Avant toute tentative de classification de fautes, nous passerons en revue les mécanismes provoquant les fautes analogiques dans les circuits intégrés [4] [8] [11].

#### 1.4.2 Sources des fautes analogiques [4] [12].

A part les fautes résultant du design et qui sont sensées être corrigées après la vérification des prototypes, le processus technologique est le responsable principal de la présence des fautes, que ce soit immédiatement après fabrication ou bien après un temps d'attente qui peut être long .

La fabrication des circuits intégrés est destinée à produire un nombre important de puces identiques. Les compositions ainsi que le layout de n'importe quel couple de circuits (supposés identiques) fabriqués à partir d'une même ligne de production sont supposées être exactement les mêmes, mais les performances de ces deux circuits peuvent être différentes à cause des variations du processus. Les sources majeures de ces variations sont :

- Les erreurs humaines et les défaillances des équipements.
- Instabilité des conditions du processus, en termes de changement de valeurs de n'importe quelle variable physique supposée constante.
- Instabilité du matériel, qui se rapporte à de petites variations dans les compositions chimiques utilisées dans la ligne du processus. Par exemple, la contamination chimique venant des résidus d'un autre processus.

- L'hétérogénéité du substrat, incluant les points défectueux, l'imperfection des surfaces, en particulier celles venant de l'implantation des ions.
- Les non-alignements des masques. Les erreurs dans les translations des alignements sont souvent dominantes parce qu'il y a plusieurs masques qui doivent être parfaitement alignés durant des étapes successives du processus.
- Les points lithographiques, causés par des poussières dans les régions transparentes ou les rayures dans les régions opaques.

Probablement, et à l'exception des erreurs humaines, les effets de toutes ces sources peuvent être modélisés comme étant des phénomènes aléatoires. La majorité de ceux-ci sont globaux, parce qu'ils affectent approximativement de la même manière, tous les dispositifs sur une puce et dans la majorité des cas, toutes les puces dans une plaquette de silicium.

D'autre part, d'autres sources de défauts sont de natures locales et affectent les dispositifs individuellement ou bien une très petite région d'une puce. Ces défauts appelés **spots** sont souvent causés par des particules dans l'environnement de fabrication et affectent soit les couches individuelles, soit les interconnexions entre deux couches. Les sources des spots les plus fréquentes sont :

- Les vides (et extras) dans le poly-silicon ou les lignes métalliques (généralement causés par les spots lithographiques).
- Trou d'oxyde.
- Contacts manquants (causés souvent par une sous-gravure).
- Contamination ionique mobile, qui peut être concentrée dans une région particulière du circuit quand il est polarisé.
- Pipes de conduction locales formées par des imperfections dans la structure du cristal.

#### **Terminologie générale [4] :**

- **Performances** : elles décrivent le fonctionnement du circuit. Connaître les performances d'un circuit permettra de juger s'il est fonctionnel ou bien défaillant.
- **Spécifications** : c'est l'ensemble des valeurs acceptables de chaque performance. Chaque performance possède deux spécifications (la borne inférieure et la borne

supérieure). Si une performance possède uniquement une spécification alors l'autre est fixée à l'infini.

- **Mesures de test ou Observations** : ce sont des mesures qui sont effectuées sur le circuit.
- **Critères de test** : c'est un sous ensemble de mesures de test qui doivent corrélérer avec les spécifications afin de pouvoir décider si le circuit passe le test ou non. L'un des buts de la simulation de fautes est de valider l'efficacité de tels critères.
- **Défaut** : mal construction au niveau d'une des parties du circuit qui le rend défaillant.
- **Défaillance ou panne** : l'effet d'un défaut.
- **Faute** : déviations au-delà des valeurs spécifiées.
- **Erreur** : fonctionnement anormal produit par une faute.
- **Détection** : procédure permettant de confirmer l'existence d'une faute dans un circuit.
- **Localisation** : s'effectue une fois la faute est détectée. C'est une procédure qui permet de déterminer l'élément défectueux qui a causé la défaillance (ou la panne).
- **Diagnostic** : détermination de la cause produisant la faute (ou le dysfonctionnement du circuit).
- **Stimulus** : signal appliqué à l'entrée d'un circuit.
- **Signature de faute** : caractérisation d'une faute résultant d'une défaillance.
- **Points de test** : les différentes connections ou nœuds d'un circuit dans lesquelles il est possible de brancher un instrument de mesure.
- **Équipement Automatique de Test (ATE)** : ensemble intégré de logiciels et matériels utilisé pour réaliser la procédure de test.
- **CUT (Circuit Under Test)** : Circuit Sous Test, représente le circuit dans la phase de test.
- **DUT (Design Under Test)** : Dispositif Sous Test, c'est une autre nomination du CUT.
- **Paramètres du processus** : les paramètres liés au procédé de fabrication et la physique des composants.
- **Paramètres du circuit** : les paramètres liés à la géométrie du design (résistance, capacité, largeur d'un transistor, etc.).
- **Paramètres du design** : appelés aussi les performances. Représentent les paramètres permettant de décider si le CUT est fonctionnel ou non.
- **Paramètres de test** : appelés aussi critères de test, ils peuvent être une partie des paramètres du design ou bien d'autres paramètres pouvant aider à décider si le circuit passe le test ou non.

### 1.4.3 Détection des pannes et la couverture de fautes

La couverture de fautes dans sa forme la plus simple est le rapport entre le nombre de défauts détectés par un ensemble donné de vecteurs de test et le nombre total de défauts possibles envisagés, telle que donnée par l'équation suivant [14] :

$$FC = \frac{D}{T}$$

Où : FC : Couverture de fautes (Fault Coverage)

D : nombre de défauts détectés (Detected Faults)

T : Nombre totale de fautes possibles

Cependant, il ya d'autres facteurs qui influent sur le calcul de la couverture de faute. Il s'agit notamment de défauts indétectables et potentiellement détectés, qui sont susceptibles de réduire la couverture de fautes qui sera obtenu pour la plupart des circuits sauf si on considère spécifiquement ces types de défauts et la façon de les appréhender.

Les portes logiques élémentaires individuelles sont complètement et facilement testables. Par conséquent, la manière dont nous interconnectons ces portes qui mènent à des problèmes de testabilité et de test.

### 1.4.4 Contrôlabilité et observabilité

Afin de détecter un défaut, les stimuli de test appliqués au CUT doivent engendrer des conditions au sein du CUT de telle sorte que le circuit défectueux produise une réponse différente de celle du circuit sans faute. Par conséquent, la détection d'une erreur donnée (quelle que soit le modèle de faute considéré) consiste à établir des stimuli d'entrée qui produisent la valeur logique appropriée ou une condition sur le site présumé de défaut pour créer une altercation entre le circuit sans faute et défectueux. Ensuite, cette altercation doit être propagée vers une sortie principale du CUT de sorte qu'elle puisse être observée. Ainsi, la testabilité d'un site de défaut donné dans un circuit est une fonction de la contrôlabilité de l'emplacement de défaut à partir des entrées primaires, ainsi que l'observabilité de l'emplacement de défaut à partir des sorties primaires. La Contrôlabilité est la facilité avec laquelle nous pouvons contrôler un site de défaut et l'observabilité est la facilité avec laquelle on peut observer un site de défaut [14] [17].

Selon le modèle de faute, chaque entrée et sortie d'une porte, chaque transistor, ou chaque paire de fils adjacents dans le circuit représente un site de défaut potentiel. En conséquence, la

testabilité de l'ensemble du circuit est une fonction de la contrôlabilité et l'observabilité de tous les sites de défauts dans le circuit complet.

### 1.4.5 Les fautes indétectables.

Si aucun vecteur de test ou une séquence de vecteurs de test n'existe pour détecter une panne donnée, alors la faute est dite indétectable [18].

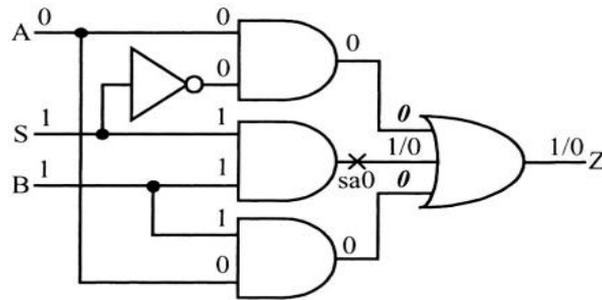


Figure 1.3 : exemple d'une faute indétectable

C'est une situation intéressante, car nous ne pouvons pas dire si la faute existe ou non. À la suite, l'opération du système ne sera pas affectée par la présence de la faute. Les défauts indétectables viennent généralement de la redondance dans la logique et du nombre de fan-out ré-convergeant qui ne se traduit pas toujours par la présence de ce type de fautes. En pratique, ces fautes sont difficiles à vérifier. Ils apparaissent tout simplement comme fautes indétectables et affectent aussi bien le temps de simulation que la détermination de leur couverture résultante. Ces fautes indétectables ont toujours tendance à ralentir le processus de leur simulation puisqu'elles ne sont guère détectées et l'ensemble des vecteurs de test est généré pour elle avant qu'elle ne soit introduite dans la liste de fautes indétectables [14].

### 1.4.6 La couverture de fautes.

La figure 1.4 montre la simulation de défauts d'une façon plus claire.

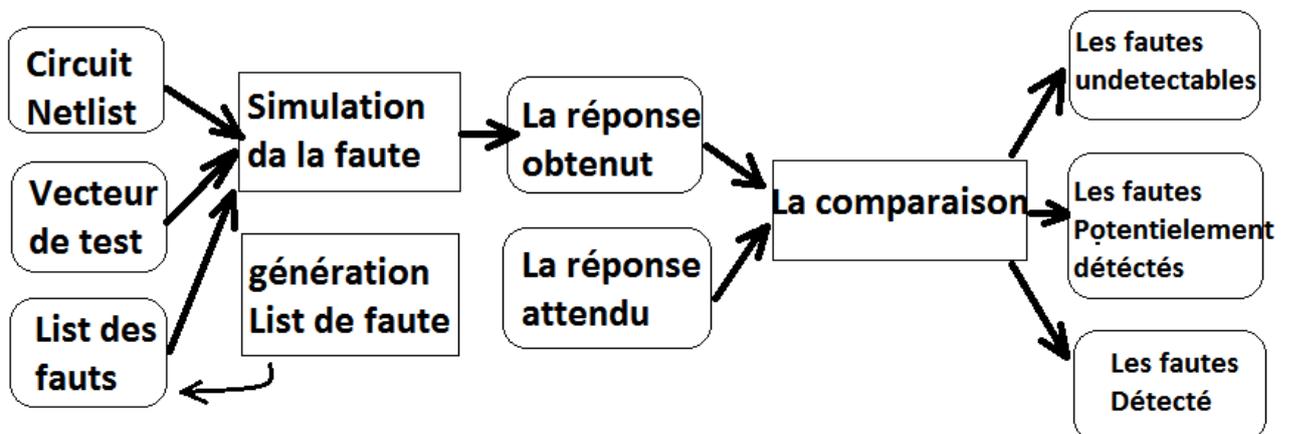


Figure 1.4 : schéma synoptique pour la détection de la faute

Au cours de la simulation de défaut initial, la liste de défauts à simuler est généralement générée par le simulateur de défaut en possession de l'utilisateur afin d'obtenir l'ensemble initial de défauts réduit ou non. Dans les simulations de défauts suivantes, la liste des défauts non-détectés antérieurement est généralement réutilisée en excluant ceux détectés. De cette manière, on minimise le temps de simulation de faute par le fait d'éviter de simuler une deuxième fois les défauts détectés précédemment. La possibilité de spécifier cette liste de fautes est également importante pour la statistique de simulation de pannes où nous avons produit un ensemble de fautes choisies au hasard, et pour combler la simulation de faute, il a été produit un ensemble de défauts extrait du lay-out physique.

Dans le processus de simulation de faute, chaque défaut doit être soit détecté, non-détecté, ou potentiellement détecté par l'ensemble donné de vecteurs de test. Alternativement, la faute elle-même peut être indétectable, et peut ainsi résider dans l'ensemble des défauts non détectés [14]. Cependant, si nous pouvons déterminer qu'un défaut est indétectable (sans recourir à des mesures correctives pour éliminer la source de la faute indétectable) alors on doit retirer la faute indétectable de la liste des défauts pour réduire le temps de simulation de défauts. Alors un calcul simple de couverture de faute a été donné dans l'équation (2.1), un calcul plus réaliste de couverture de faute est donné par:

$$FC = \frac{D + xP}{T - U}$$

où D est le nombre de défauts détectés, P est le nombre de défauts potentiellement détectés (y compris des défauts d'oscillation), T est le nombre total de défauts, U est le nombre de défauts déterminés comme étant indétectable, et x est la probabilité de détection pour les défauts éventuellement détectés. La valeur de x peut être ajustée dans la plage  $0.0 \leq x \leq 1.0$  en se basant sur les données de l'activité du CUT et selon le jugement du concepteur et / ou l'essai de l'ingénieur. Le calcul de la couverture de fautes est conservateur lorsque  $x = 0.5$ . Tandis que le cas de la probabilité la plus défavorable pour la détection de ces défauts est 0.0.

L'hypothèse que le circuit défectueux soit initialisé à une valeur logique 0 ou 1 et reste à cette valeur fait de 0,5 une valeur conservatrice valide [19]. En revanche tout défaut éventuellement détecté ou sans risque être supposé détecté par l'intermédiaire de multiples observations des deux valeurs logiques peut être compris dans la liste des défauts détectés (de P à D) pour le calcul de la couverture fautes [14].

### 1.5 Techniques de conception en vue de test (Design for testabilité D.F.T.).

Les algorithmes de génération de test des circuits combinatoires ne peuvent être utilisés pour les circuits séquentiels que s'il y'a subdivision du circuit en parties combinatoires et éléments mémoires. Le chargement à volonté des états et leur lecture permettent de simplifier l'opération de test. Pour réduire les problèmes de test au niveau des cartes électroniques ou au niveau des circuits hautement intégrés, plusieurs standards de DFT ont été élaborés, et autres sont en court de développement.

Le but essentiel de ces méthodes est de trouver une configuration standardisée pour le test des circuits VLSI. Leur objectif est d'améliorer la testabilité par plus de contrôle et meilleure observation des valeurs des signaux au niveau des nœuds internes des circuits. Ces nouvelles techniques ont non seulement réduits le coût de génération de test (TPG) mais aussi le processus de test [6]:

- En rendant les nœuds internes plus accessibles;
- Par transformation des circuits séquentiels en circuits combinatoires et décomposition des circuits complexes en sous-circuits simples à tester.
- En rendant le circuit auto testable;
- En réduisant le flot de données requis par le processus de test des circuits.
- Les méthodes ad hoc [6] :

Ces méthodes interviennent après conception des circuits pour ramener un plus à la testabilité [20]. Parmi ces méthodes on trouve:

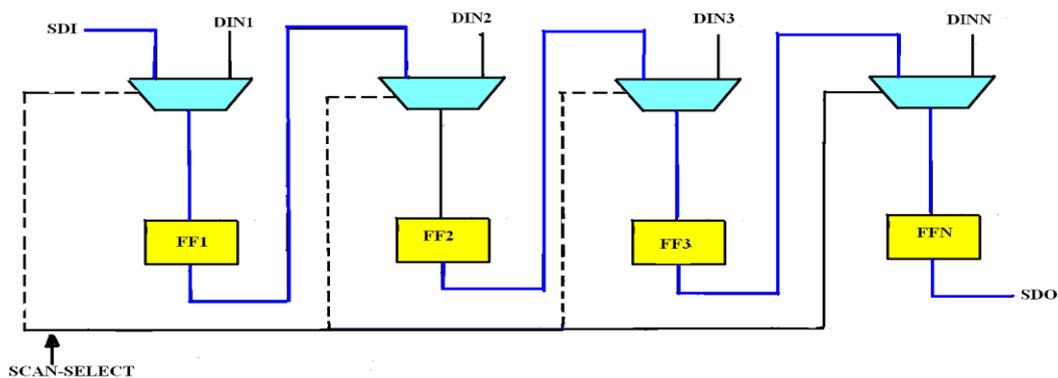
- ❖ L'introduction des points de test par technique de routage afin de faciliter l'accès à certains nœuds pour d'avantage de contrôle et d'observation.
- ❖ Le partitionnement des circuits complexes en sous circuits testables auxquels sont additionnées des portes logiques généralement à deux entrées dont l'une sert pour donnée normale, quant à l'autre elle sert au contrôle. Autres circuits (les multiplexeurs ou démultiplexeurs) peuvent être employés pour développer des fonctions de test additionnelles assurant l'entrée ou la sortie du signal de test [21]. L'inconvénient qui fut remarqué réside dans les retards (delay time) qu'introduisent ces circuits auxiliaires et entraînant une dégradation de la réponse.

### 1.5.1 Approche structurale: techniques de Scan-Path (le balayage) [6] .

Les techniques Scan-Path visent à faciliter l'accès aux variables internes du circuit. L'objectif de cette circuiterie d'aide au test est de rendre un circuit séquentiel testable en le faisant se comporter comme étant un circuit combinatoire. Cette technique donne une meilleure contrôlabilité et observabilité dans les circuits séquentiels. Ces derniers sont conçus de manière à ce que leurs bascules puissent fonctionner aussi bien en mode de chargement parallèle qu' en mode de chargement sériel.

– **Mode de chargement parallèle:** Les différentes bascules sont chargées en parallèle lors du fonctionnement normal du circuit.

\_ **Mode de chargement sériel:** Dans ce mode de test, on charge les différentes bascules d'une façon sérielle par des vecteurs de test désirées (contrôlabilité) par simple décalage. On peut observer leur contenu en sortie (observabilité)



**Figure 1.5 : Structure d'une cellule SP**

La séquence d'entrée de scrutation (Scan Data In) est appliquée à l'une des entrées du premier multiplexeur figure 1.5, Suivant l'état de la ligne de commande SCAN-SELECT, les bascules fonctionnent en mode chargement parallèle à travers les entrées "DINn" qu'en mode de décalage sériel d'où la formation d'un registre à décalage. Le circuit devient alors contrôlable à travers SDI et observable à travers SDO (Scan Data Out).

### 1.5.2 Technique LSSD (Level Sensitive Scan Design) .

La technique LSSD est une technique de conception qui peut remédier aux problèmes spécifiques de Scan-Path. Cette technique utilise des cellules de SP composées de deux bascules maître- esclave dont leur fonctionnement est contrôlé par des horloges multiples pour améliorer la

contrôlabilité et l'observabilité du circuit. Le principe de LSSD est le même que celui du Scan-Path, la seule différence est le fait que la cellule LSSD utilise des horloges multiples donc une circuiterie plus complexe que des circuits Scan-Path.

### **1.5.3 Technique de Scan-Path Partial .**

La technique SP permet une économie de 50% à 80% sur la durée. Cependant, le temps de test est presque entièrement consacré au changement et décalage de la chaîne SP, en plus de la circuiterie supplémentaire qu'elle entraîne dans le cas d'un circuit complexe, ce qui peut réduire la performance du système.

On peut réduire ces problèmes en décomposant le registre afin d'effectuer des opérations de test parallèles. Mais il est suggéré d'utiliser un nombre de bascules limité, on réalise ainsi une réduction du temps de test et de la circuiterie additionnelle. Les séquences de changement et décalage deviennent ainsi plus petites. Les bascules à introduire dans la chaîne SP sont celles qui sont supposées difficiles à contrôler et qui doivent représenter une source d'information et non seulement un point de réception de données. Les FFs contrôlables sont observables à partir des PIs et des POs peuvent être exclus de la chaîne SP.

## **1.6 La technique BIST.**

### **1.6.1 Introduction au BIST.**

L'idée de base de BIST, dans sa forme la plus simple, consiste à concevoir un circuit de sorte que le circuit puisse s'autotester et de déterminer s'il est «bon» ou «mauvais». Cela nécessite généralement des circuits supplémentaires incorporés dans la conception du circuit pour faciliter la fonction d'auto-test. Ces fonctionnalités supplémentaires doivent être capables de générer des signaux de commande, avec un mécanisme pour déterminer si les réponses de sortie du circuit sous test (CUT) sont correctes ou non [22].

Prenons un exemple de mise en œuvre de l'approche BIST et supposons qu'un ensemble de vecteurs de test a déjà été développé pour le CUT, et nous voulons utiliser ces vecteurs de test pour tester le CUT au cours de la séquence de BIST. Les vecteurs de test pourraient être stockés dans une mémoire (ROM) avec un compteur utilisé pour lire ces vecteurs à partir de la ROM pour les appliquer au CUT, Cette combinaison du compteur et de la ROM constituerait la fonction (TPG) dans le circuit BIST.

Une structure similaire peut être utilisée pour construire la fonction ORA où les données du circuit prévues sans défauts sont également stockées dans une mémoire ROM. Comme les

réponses attendues sont lues à partir de la ROM- ORA, elles sont comparées à la réponse de sortie réelle du CUT pour chaque vecteur de test appliqué par le TPG. Toute discordance détectée par le comparateur est verrouillée pour indiquer une défaillance produite pendant la séquence de test BIST.

Un seul compteur, cadencé par le même signal d'horloge que celui d'entrée au CUT, peut être utilisé pour lire les données des deux ROM à la fois, et qu'on le considère comme un contrôleur de test.

### 1.6.2 La structure générale de la technique BIST

Comme le montre le schéma de la figure 1.6, trois blocs principaux constituent la structure de la technique BIST:

- ❖ TPG : générateur des signaux de test (Test Pattern Generator).
- ❖ ORA : analyseur de réponse (Output Response Analyser).
- ❖ CUT: circuit soumis au test (Circuit under Test).

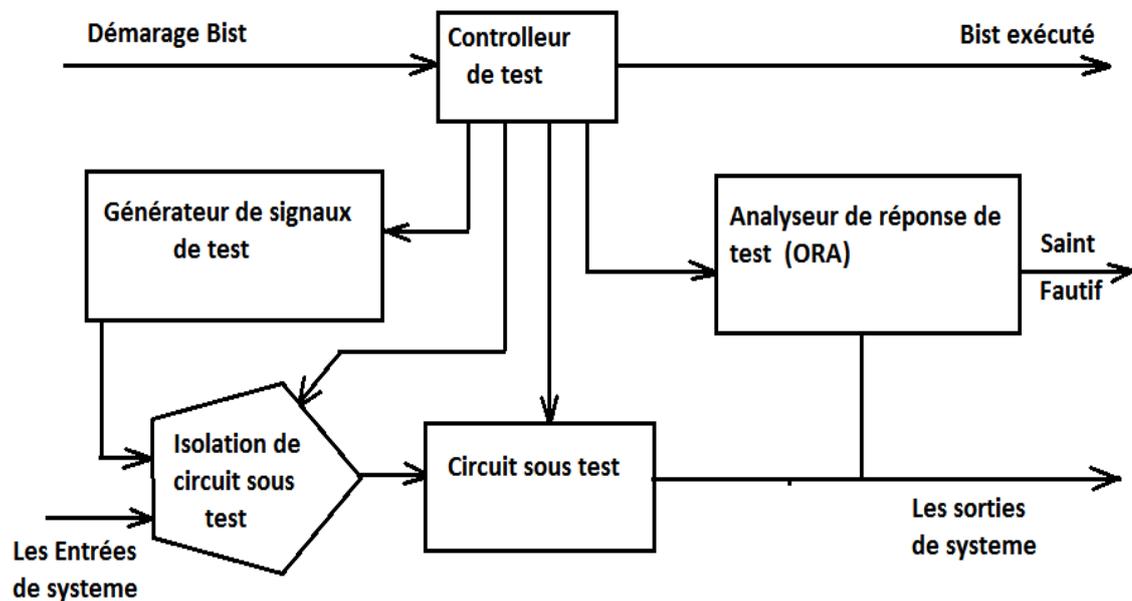


Figure 1.6 : structure générale de BIST

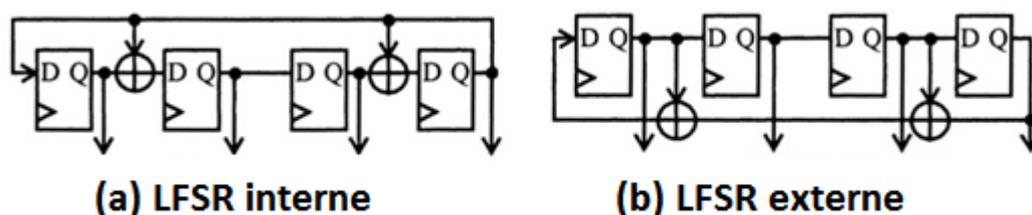
### 1.6.3 Le générateur des signaux de test (TPG) .

Le générateur de signaux de test est le composant le plus important dans la structure BIST [23].

➤ **Le registre à décalage (LFSR: Linear Feedback Shift Register) :**

Le LFSR est l'une des implémentations le plus souvent utilisées dans les applications BIST [26]. Il est plus efficace qu'un compteur, car le LFSR exigent moins de combinaison logique par flip-flop. Il existe deux types de LFSR : LFSR à implémentation interne et externe. L'interne demande une implémentation à une fréquence d'opération très élevée pour assurer une bonne performance. Toutefois , il y'a des cas où l'utilisation de l'externe est plus efficace [24] .

LFSR signifie : un registre à décalage avec les chemins de rétroaction combinés linéairement par des portes OU-exclusif.



**Figure 1.7 : les deux types de LFSR interne et externe**

Le placement des portes OU exclusif est définie par le polynôme caractéristique  $P(X)$  du LFSR - (a) et (b) :  $P(X) = X^4 + X^3 + X + 1$ .

Chaque coefficient non nul dans le polynôme caractéristique représente un OU-exclusif dans le réseau de rétroaction, les exceptions sont les coefficients  $X^n$  et  $X^0$  qui sont toujours non-nul. Le degré du polynôme nous donne le nombre de bascules. Les portes OU-exclusif et registre à décalage agissent pour produire une séquence binaire pseudo-aléatoire (PRBS : Pseudo Random Binary Sequence ) à chacune des sorties des bascules. En choisissant correctement les points sur lesquels nous prenons la rétroaction d'un registre à décalage ; nous pouvons produire une PRBS de longueur  $2^n - 1$  (longueur de la séquence maximale qui comprend tous les motifs possibles ou vecteurs de n bits) [27].

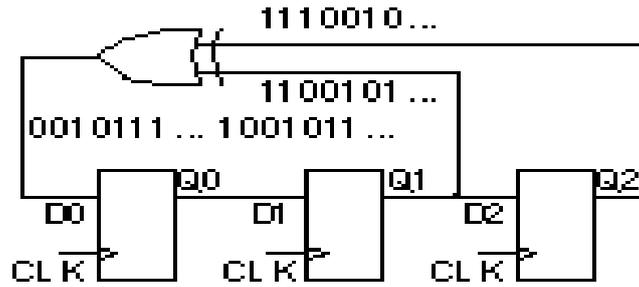


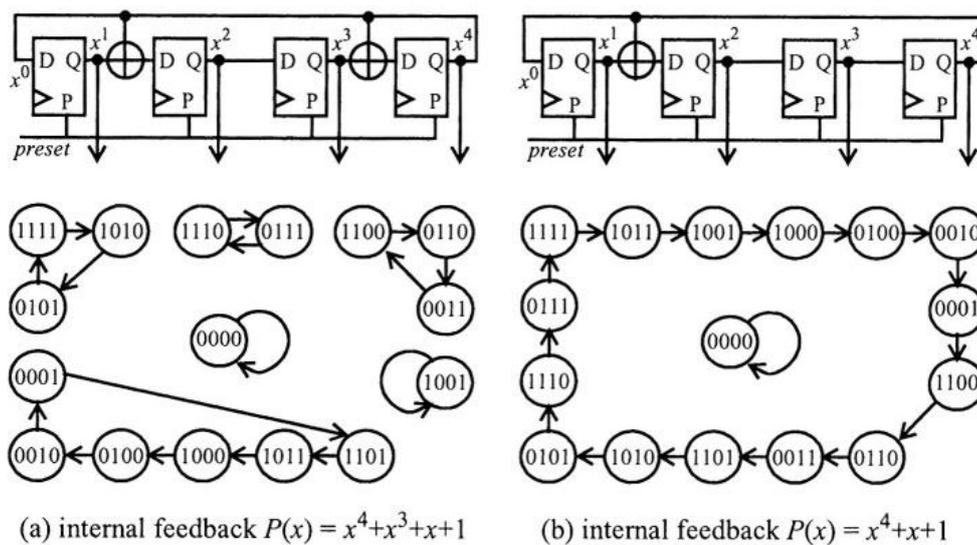
Figure 1.8 : un LFSR à trois bits

La figure 1.8 représente un registre à décalage à rétroaction linéaire (LFSR). Un LFSR 3-bits de longueur maximale produit une chaîne de répétition pseudo-aléatoire de sept nombres: 7, 3, 1, 4, 2, 5, 6 ( $2^3-1=7$ )

➤ **Polynômes primitifs**

Les Polynômes qui aboutissent à une séquence de longueur maximale sont appelés polynômes primitifs et les polynômes qui ne produisent pas une séquence de longueur maximale sont appelés non primitifs

- ❖ Un polynôme primitif est comme un nombre premier; ce qu'il n'est divisible que par lui-même et par 1.
- ❖ Les Polynômes primitifs avec un nombre minimal de coefficients non nuls sont les polynômes caractéristiques les plus désirés pour les LFSR à utiliser pour les TPGS dans des applications BIST [26] [14].



(a) internal feedback  $P(x) = x^4 + x^3 + x + 1$

(b) internal feedback  $P(x) = x^4 + x + 1$

Figure 1.9 : les deux types de TPG non primitive (a) et primitive(b).

(a) : c'est un Polynôme Non primitif.

(b) : c'est un polynôme primitif.

Le tableau 1.2 nous donne les polynômes primitifs avec un minimum de coefficients non-nuls (et par conséquent moins de portes OU exclusives) pour les degrés allant de 2 à 74. Comme on peut le voir sur le tableau, le nombre minimal de portes OU exclusives nécessaires à mettre en œuvre l'un de ces polynômes primitifs est compris entre 1 et 3.

Degree ( $n$ )	Polynomial	Degree ( $n$ )	Polynomial
2, 3, 4, 6, 7, 15, 22, 60, 63	$x^n + x + 1$	12	$x^n + x^6 + x^4 + x^3 + 1$
5, 11, 21, 29, 35	$x^n + x^2 + 1$	33	$x^n + x^{13} + 1$
8, 19, 38, 43	$x^n + x^6 + x^5 + x + 1$	34	$x^n + x^{15} + x^{14} + x + 1$
9, 39	$x^n + x^4 + 1$	36	$x^n + x^{11} + 1$
10, 17, 20, 25, 28, 31, 41, 52	$x^n + x^3 + 1$	37	$x^n + x^{12} + x^{10} + x^2 + 1$
13, 24, 45, 64	$x^n + x^4 + x^3 + x + 1$	40	$x^n + x^{21} + x^{19} + x^2 + 1$
14, 16	$x^n + x^5 + x^4 + x^3 + 1$	42	$x^n + x^{23} + x^{22} + x + 1$
18, 57	$x^n + x^7 + 1$	46	$x^n + x^{21} + x^{20} + x + 1$
23, 47	$x^n + x^5 + 1$	54	$x^n + x^{37} + x^{36} + x + 1$
26, 27	$x^n + x^{12} + x^{11} + x + 1$	55	$x^n + x^{24} + 1$
30, 51, 53, 61, 70	$x^n + x^{16} + x^{15} + x + 1$	58	$x^n + x^{19} + 1$
32, 48	$x^n + x^{28} + x^{27} + x + 1$	65	$x^n + x^{18} + 1$
44, 50	$x^n + x^{27} + x^{26} + x + 1$	69	$x^n + x^{29} + x^{27} + x^2 + 1$
49, 68	$x^n + x^9 + 1$	71	$x^n + x^6 + 1$
56, 59	$x^n + x^{22} + x^{21} + x + 1$	72	$x^n + x^{53} + x^{47} + x^6 + 1$
66, 67, 74	$x^n + x^{10} + x^9 + x + 1$	73	$x^n + x^{25} + 1$

**Tableau 1.2 : les polynômes primitifs de puissance 2 - 74**

#### 1.6.4 Analyseur de signaux de réponse (ORA).

L'analyseur de réponse utilise un registre à décalage LFSR (figure 1.10) comme le TPG; sauf qu'il nécessite une entrée de données contrairement au TPG qui est un system fermé; l'analyseur de réponse est représenté par un polynôme appelé polynôme de données. L'idée principale est de diviser le polynôme de données  $K(x)$  par le polynôme caractéristique de LFSR  $(2) P(x)$ ; le reste de la division  $R(x)$  correspond à la signature de (faute / pas de faute) à la fin de séquence de BIST [28].

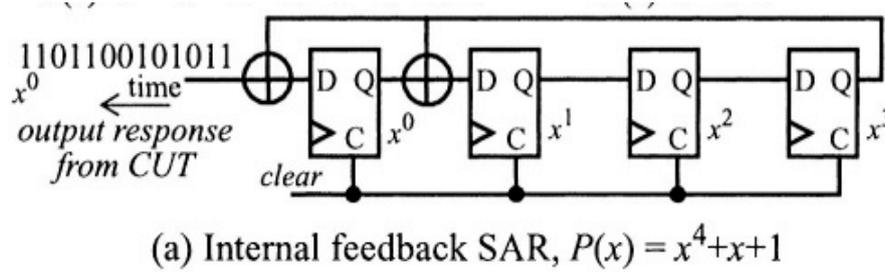


Figure 1.10 : architecture interne de l'ORA

### 1.6.5 Exemple de signature de fautes d'un circuit numérique à l'aide de la technique BIST

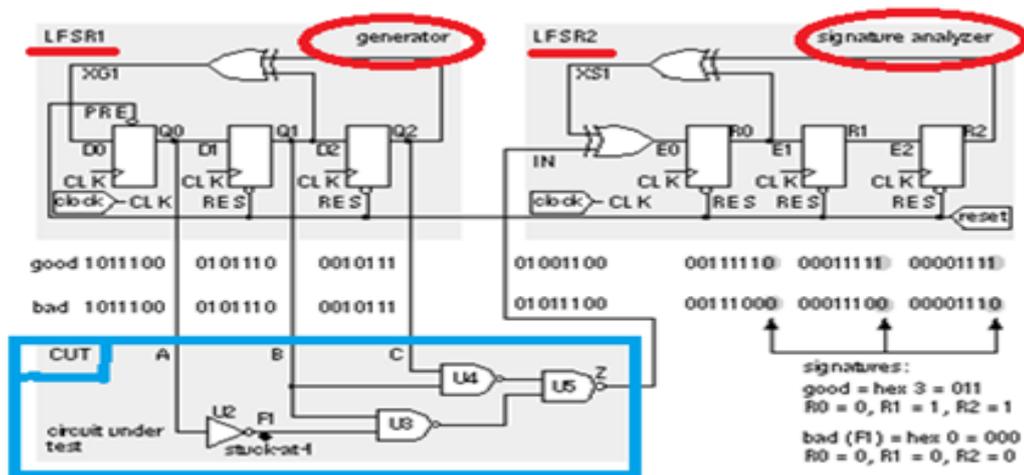


Figure 1.11 : exemple numérique de la technique BIST

Nous pouvons combiner le générateur de séquence pseudo-aléatoire TPG avec le registre de la signature ORA pour former la structure BIST simple illustrée en figure 1.11. Le LFSR1 (TPG) génère un maximum de longueur ( $2^3 - 1 = 7$  cycles) PRBS. Le LFSR2 (ORA) calcule la signature («011» pour le bon circuit) du CUT. Le LFSR1 est initialisé à «100» ( $Q0 = 1, Q1 = 0, Q2 = 0$ ) et le LFSR2 est initialisé à «000».

Le LFSR du TPG comme illustré en figure 1.12, génère un cycle de 15 mots binaires pseudo-aléatoire en attaquant le circuit sous test par ces mots et à la sortie on obtient une séquence de bits (1101100101011).

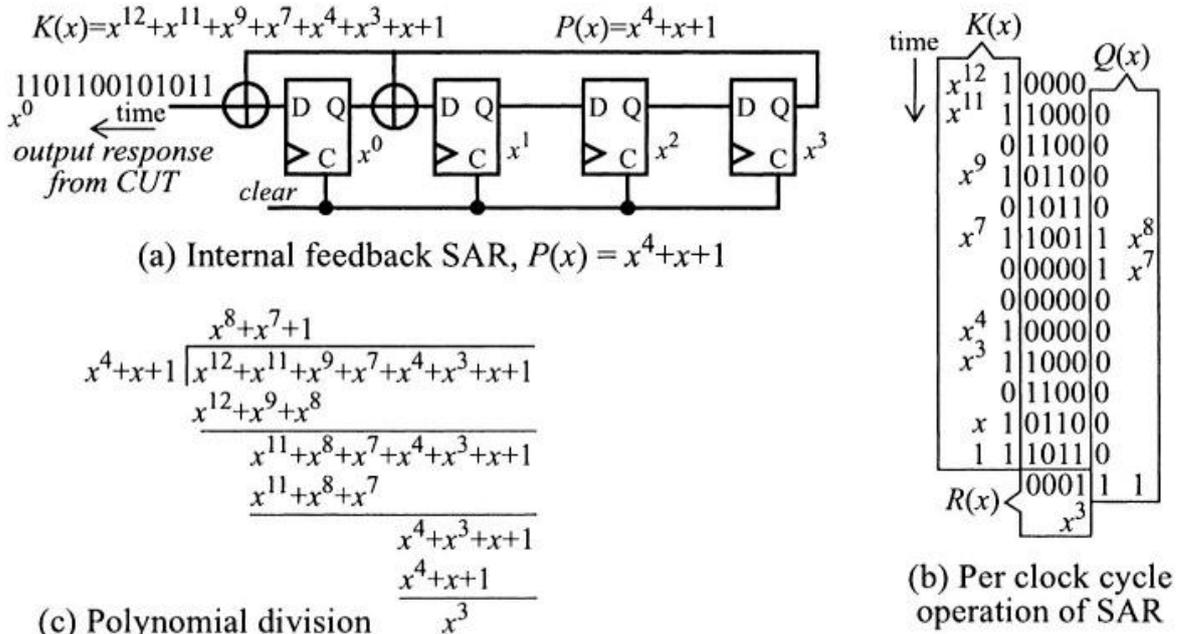


Figure 1.12 L'ORA fait la division polynomiale pour le circuit sain

Les bits binaires sont des coefficients d'un polynôme appelé polynôme de données, «  $1101100101011 = X^{12} + X^{11} + X^9 + X^7 + X^4 + X^3 + X + 1 = K(x)$  » ; qui passera ensuite en phase d'analyse (ORA) à travers un LFSR 2 (figure 12-a) . Ce registre à une entrée de données avec deux portes OU-exclusif ; à la fin de la séquence l'ORA délivre une signature («  $0001 = X^3$  » dans notre exemple pour le bon circuit) figure (b-c). Maintenant s'il y'a une faute (collage à 1 ou collage à 0) dans le CUT, la signature sera différente comme le montre la figure 1.13 :

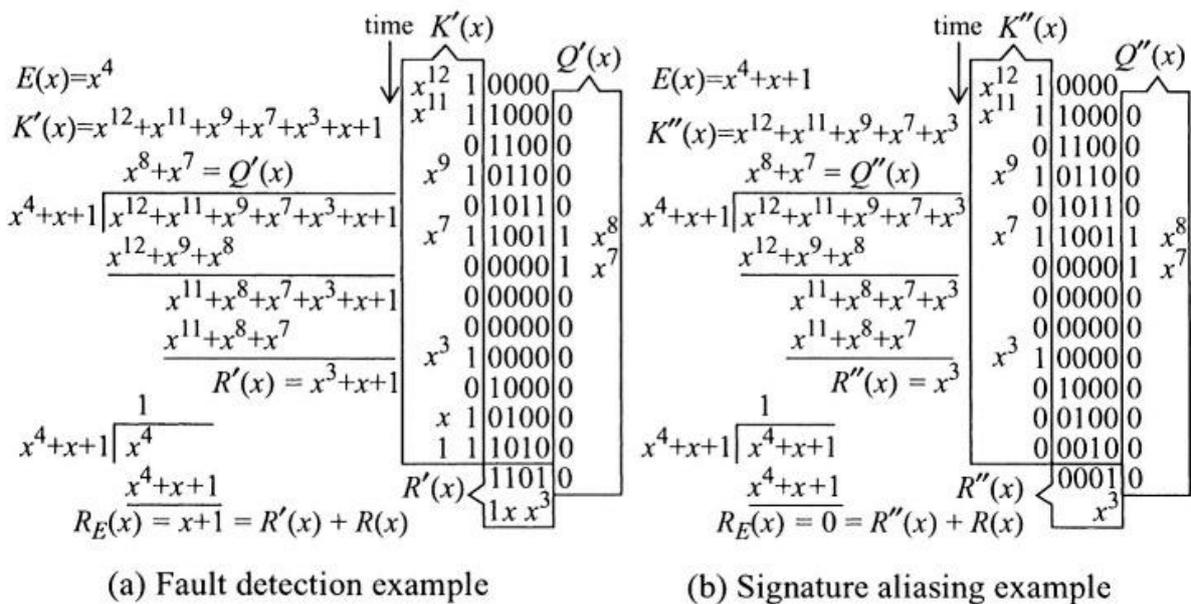


Figure 1.13 : L'ORA fait la division polynomiale pour le circuit fautif

Le polynôme,  $E(x)=X^4$ , signifie que la faute se trouve au niveau du bit à coefficient 4, c.-à-d.

$K(X)=K'(X)+X^4$ , la signature devient  $1101 = X^3+X+1$  figure (1.13-a).

### Remarque :

Il y'a une faible probabilité que la signature d'un circuit mauvais sera la même que celle d'un bon circuit. Ce problème est connu sous le nom aliasing ou masquage d'erreur. La Figure (b) illustre bien ce problème : l'erreur  $E(x)=X^4+X+1$  alors  $K''(x)=X^{12}+X^{11}+X^9+X^7+X^3$ , mais la signature  $R''(x)=X^3=R(x)$ , bien sur le quotient  $Q''(x)=X^8+X^7$  diffère de  $Q(x)=X^8+X^7+1$ , alors la comparaison du quotient sera comme une solution.

## 1.7 Conclusion

A travers ce chapitre on a fait remarquer que le test est une tâche indispensable dans l'industrie de nos jours, la technique BIST est admise comme l'une des meilleurs techniques utilisée récemment car elle efficace et simple à implanter. Seulement deux parties de cette structure de test sont essentiels dans notre travail : la TPG et l'ORA. La première partie génère des vecteurs de test pseudo-aléatoire pour attaquer le circuit sous test. L'autre partie est un autre LFSR qui sert à analyser les réponses de circuit à tester pour nous donner une signature. Néanmoins, son application dans le monde analogique exige d'abord un aperçu sur les types de fautes ,leur modélisation dans un environnement de circuits analogiques et numériques dont les détails seront énoncés dans les Chapitres qui suivent.

# **Chapitre 2**

**la simulation est son apport  
dans le test de circuits  
électroniques**

## 2.1 Introduction.

La simulation est un outil aussi important dans un processus de test que dans le domaine de conception de circuits électroniques. Elle peut servir de moyen de diagnostic de circuits défectueux et prédire les conséquences malencontreuses que peuvent subir les circuits une fois finis. On peut citer ici deux sortes de processus de simulation de fautes : la simulation après (after) test (SAT) et simulation avant (before) test (SBT).

Tout comme en domaine du design, la simulation peut prendre part dans le test de circuits dans différents niveaux d'abstraction : simulation aux niveaux des jonctions ; des composants primitifs, des portes, fonctionnel et comportemental etc.

## 2-2 L'intérêt de la simulation des fautes :

la simulation des fautes est introduite dans le domaine de test pour :

- Déterminer le Taux de couverture d'une séquence de test
- Etablir la liste des fautes non détectées
- Dresser le dictionnaire de fautes, c.-à-d. la liste des fautes détectées, les vecteurs détectant chacune de ces fautes et les réponses du circuit.

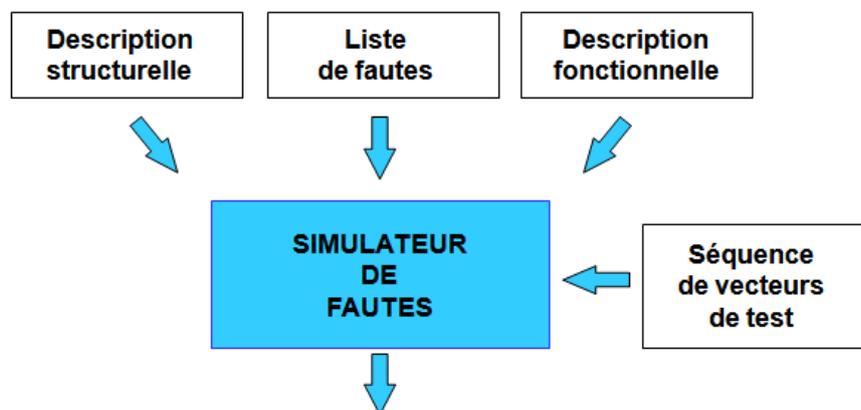


Figure 2.1 : diagramme d'un processus de simulation de faute

## 2-3 Test ou diagnostic :

Réaliser un test de circuit c'est d'arriver à détecter son fonctionnement non conforme aux spécifications précisées dans son cahier de charge (pour des raisons de fonctionnalité logique, de vitesse de fonctionnement ou encore de niveau électrique). Dans la majorité des cas le test sert uniquement à distinguer les **bons circuits (Golden circuits) des mauvais** pour être

livrés au client. Ces derniers circuits sont à éliminer et que la cause exacte du mauvais fonctionnement n'a la plupart du temps pas à être analysée.

Dans certain cas, cependant la cause de mauvais fonctionnement doit être identifiée

- Dans le but d'améliorer la conception du circuit ou le processus de fabrication pour augmenter le rendement de production, une analyse de défaillance est menée en fin de fabrication sur les circuits jugés mauvais.
- lorsque le circuit dispose d'éléments défectueux qui aujourd'hui est devenu chose classique pour la production de circuit mémoire (SRAM et DRAM) .

Lorsqu'une telle identification de la cause est souhaitée ; ce genre de test n'est plus suffisant ; il doit être complété par une phase de diagnostic, qui nécessite une analyse beaucoup plus longue et détaillée. Le diagnostic doit en effet permettre de distinguer les différentes causes internes possibles pour une même erreur externe détectée par le test initial. Une telle tâche est également souvent nécessaire lors du test de circuit prototype, dans un objectif de déverminage.

### 2.3.1 Diagnostic des défauts

La Simulation est très importante dans le diagnostic, pour cela il existe des logiciels sur le marché, comme SPICE et PSPICE, SABER etc ... Ce sont des logiciels qui vérifient le bon fonctionnement du dispositif sous test en lui attribuant toutes les valeurs des éléments qui le constituent, pour confirmer ou nier que le problème est du ou pas au concepteur lui-même. Ensuite on procède à la vérification et aux tests pratiques. Le tableau suivant donne d'une façon explicite le degré d'importance du diagnostic utile pour les défauts dans les différentes phases du développement du produit électronique.

Défaut	Conception	Prototype	Fabrication	Usage
<b>Conception</b>	+++	+	<b>0</b>	<b>0</b>
<b>Fabrication</b>	<b>0</b>	+++	+++	<b>0</b>
<b>Vieillessement</b>	<b>0</b>	<b>0</b>	+++	
<b>Vérification par Simulation</b>	+++	+	<b>0</b>	<b>0</b>
<b>Détection par Test</b>	<b>0</b>	++	+++	+++
<b>Diagnostic utile</b>	<b>0</b>	+++	++	+

**Tableau 2.1 : Type de défauts à détecter (moyen de détection et diagnostic)**

L'objectif du diagnostic des fautes est d'établir une forme d'hypothèse qui permet de détecter la faute et de prédire la localisation physique de la défaillance au niveau de la puce de silicium et de pouvoir l'identifier. Si les résultats de ce processus sont précis, l'analyse de la cause de la défaillance pourra être déterminée et permettra en conséquence l'amélioration de la conception du circuit et même la rentabilité de la production. Cette précision dépend en réalité de la précision du modèle de faute [6].

Les méthodes de diagnostic pour les circuits analogiques peuvent se classer en deux groupes dépendant de l'occurrence de la simulation avant ou après le test. Selon certains travaux de recherches [29], [30], [31], [32], la simulation avant le test appelée SBT (Simulation Before Test) permet la détection des défauts structurels et paramétriques locales. La méthode de simulation après test connue sous l'abréviation SAT (Simulation After Test) intervient en détection des fautes paramétriques globales.

L'approche SBT est conçue autour de trois tâches essentielles à savoir:

- ❖ Un dictionnaire de fautes basé sur les résultats de la simulation des fautes. Les réponses du circuit avec et sans fautes sont stockées sous forme de tableau ainsi que leur vecteurs de test et leurs fautes responsables.

- ❖ A une sélection optimale de mesures dont le nombre est voué à la baisse sans affecter le diagnostic du circuit. D'autre part, une sélection de stimulus doit être menée soigneusement afin d'exciter le circuit et faire propager l'effet de la faute vers la sortie ou à un nœud à observer. Il est à remarquer ici qu'un certains nombres de fautes produisant le même effet au niveau des nœuds d'observation. De telles fautes sont regroupées ensembles pour constituer ce qu'on appelle des groupes d'ambiguïté. Ce genre de groupe constitue une étape indispensable pour la localisation des fautes.

- ❖ l'étape d'identification est complexe et repose sur la reconnaissance de modèle. Les techniques employées ici sont basées sur la distance floue, méthodes adaptatives et réseaux de neurones. Les inconvénients que présente cette approche SBT sont :

- a) Un modèle de faute employé.
- b) Un temps total énorme que requiert l'ensemble de fautes du dictionnaire à simuler.
- c) Un problème de stockage que peut entraîner la taille du dictionnaire.

L'avantage de cette approche est que tout l'effort informatique s'effectue antérieurement au test et en off line. Tout comme, elle est fortement recommandée pour le diagnostic des fautes catastrophiques.

L'approche SAT consiste à l'évaluation des paramètres du circuit à partir d'un système d'équations indépendantes de la connaissance de la topologie de ce circuit et sur un ensemble de mesures. Si ce nombre de mesures est suffisamment grand tous les éléments du circuit peuvent être identifiés. Dans le cas contraire, l'identification se limitera à un nombre trop restreint d'éléments fautifs. Ces éléments sont identifiés à partir des déviations obtenues par comparaison des valeurs mesurées aux valeurs nominales des différents paramètres. Ces déviations peuvent être aussi calculées par résolution des équations et si ces valeurs sont en dehors de la gamme de tolérance, les éléments sont considérés comme défectueux.

Cette méthode de diagnostic s'applique convenablement pour le cas des fautes paramétriques et l'élément affecté peut être identifié même si le circuit fonctionne correctement. L'inconvénient réside dans l'effort informatique considérable qu'exige un tel diagnostic et qui convient moins dans une application de test de production [6].

### **2-3-2 L'origine de défauts dans les circuits intégrés [6] :**

Le plus souvent, les défauts rencontrés dans les circuits intégrés proviennent de deux sources différentes à savoir les défauts provenant des **erreurs de conception** et les **défauts de fabrication**. Tous les deux peuvent entraîner un dysfonctionnement total du circuit ou à des degrés moindres. Dans ce qui va suivre, nous essayons de définir ces deux sources de défauts.

#### **A- Les erreurs de conception.**

L'augmentation de complexité des circuits rend de plus en plus difficile la tâche de test qui est aussi présente durant le moment de conception servant à vérifier l'implémentation convenable des spécifications de haut niveau d'abstraction. Des validations à tous les niveaux de la conception (fonctionnel, comportemental et circuit) peuvent faciliter cette vérification. Ces validations sont en général faites par simulation voire quelquefois par preuve formelle, et sont très consommatrices de temps [32]. Elles sont de plus incomplètes car elles ne peuvent pas être exhaustives pour des raisons économiques (temps de simulation trop importants) mais aussi pour des raisons techniques (paramètres environnementaux trop importants). Ces erreurs de conception sont mises en évidence pendant la phase de caractérisation du produit avant fabrication: cette phase est dite phase de test de prototype. Elles sont résolues par une conception adéquate appuyée par l'emploi d'instruments de précision exigée pour s'assurer de la

performance du circuit: il s'agit d'une opération de débogage de conception [8, 10, 33]. Cette opération est suivie d'une évaluation de la conception (design) par la mesure de cette performance sous des conditions spécifiques (dans le pire des cas) de températures de tensions etc. . Bien que parfois mises en évidence par des procédures de test, ces erreurs ne constituent pas la cible et l'objectif des techniques et méthodes présentées dans nos travaux de recherche.

### **B- Défauts de fabrication [34] :**

Quelles que soient les qualités du processus de validation et donc du niveau de confiance et du crédit à apporter à la conception vis à vis des spécifications, il reste néanmoins que les défauts de fabrication peuvent apparaître à n'importe quelle étape de la fabrication ou post - production. Ils doivent donc être pris en compte :

- En essayant de les minimiser le plus possible par une amélioration du rendement de production,
- Lors de l'élaboration de techniques de test pour le circuit.

Certains de ces défauts peuvent avoir lieu sur toute la ligne ou chaîne de production de circuits. Ils sont appelés défauts globaux. Par contre d'autres défauts ne se produisent que sur de faibles zones de circuits: on les appelle les défauts locaux. Quant à leurs causes d'apparition sont dues à différents mécanismes qui contribuent à l'occurrence de ces défauts de fabrication.

### **2-3-3 Classification et types de défauts [35]**

#### **a- Classes de défauts des Circuits Intégrés.**

Les circuits à très grande échelle d'intégration sont siège de plusieurs défauts physiques. On distingue deux classes fondamentales de ces défauts:

- ❖ **les défauts durs:** dits « Hard Faults » affectent la fonctionnalité du circuit et sont connus comme défauts fonctionnels. Ils peuvent entraîner:
  - Un dysfonctionnement pour une configuration particulière de données qui de ce fait peut s'avérer fort complexe à détecter,
  - une défaillance franche (indépendante des données) dont la détection posera beaucoup moins de problème.

Ces fautes sont aussi connues comme fautes catastrophiques dues souvent à des défauts aléatoires (spot defect), par exemple une particule de poussière sur un masque photo-lithographique entraînant une modification locale de la structure comme les courts-circuits et les circuits ouverts.

- ❖ **Les défauts tendres:** dits «Soft Faults» pour lesquels bien que le dispositif soit apparemment fonctionnel, ne peut atteindre certaines performances comme par exemple le niveau

de tension de sortie ou la fréquence de travail maximale. Il s'agit des déviations de sortie enregistrées en dehors des intervalles de tolérances qui sont dues généralement à des fluctuations des paramètres du processus de fabrication comme mentionnés dans [36, 37, 38,39] dont les plus importants sont l'épaisseur d'oxyde, la mobilité en surface, le dopage en substrat, variation de la géométrie du transistor etc. .

Certains défauts peuvent n'affecter que le comportement paramétrique du circuit. On les appelle aussi les défauts paramétriques. Leur détection peut s'avérer très difficile et demande des tests spécifiques. L'opération ici ne se limite pas à la recherche de bons stimuli pour faire propager la faute mais plutôt trouver les paramètres adéquats fournissant des écarts hors tolérances à la sortie du circuit [3].

### **b- Types de défauts [20] [40]**

Les défauts ou les erreurs pouvant altérer le bon fonctionnement d'un circuit intégré peuvent être issues de l'une des causes suivantes:

#### 1- Défauts matériels

- Défauts cristallins et dislocations.
- Composition de la métallisation.
- La contamination ionique.

#### 2- Propriétés de métallisation

- Etape de couverture.
- L'égratignure ou pontage (bridging).
- L'alignement.
- Largeur de ligne.
- Corrosion.
- Défauts de gravure.
- Epaisseur de ligne.
- Pauvre adhésion.

#### 3- Défauts de conception

- Alignement des masques.
- Compatibilité des matériaux.
- La densité du courant.
- Géométrie des lignes
- Barrières de diffusion.

#### 4- Défauts induit par le processus général

- Défauts de soudure
- Erreurs de masques.
- Fil de liaison oublié.
- Fils de conduction court-circuités.
- Présence de particule.
- Mise en boîtier.

## 2.4 Mécanisme de défaillance [34] [41]

On entend par mécanisme de défaillance tout processus physique ou chimique qui conduit à une défaillance d'un dispositif qu'il s'agit d'une panne ou d'un défaut de fonctionnement (hard fault) ou de défaut paramétrique (soft fault). On distingue les mécanismes de défaillance dans les semi-conducteurs; les mécanismes de défaillance dans les appareils ; les mécanismes de défaillance liés à l'environnement.

### 2.4.1 Les défauts des semi-conducteurs.

On rencontre le plus souvent des circuits électroniques analogiques tels que les diodes, les transistors, les circuits qui tombent en panne même si apparemment ces composants n'ont subi aucune usure. Cela est dû à plusieurs mécanismes qui se manifestent au niveau de leur structure interne tels que :

- 1- **Claquage de couches isolantes (Oxyde break down):** Au fur et à mesure que le degré d'intégration s'élève, les couches isolantes des semi-conducteurs deviennent de plus en plus minces, elles se perforent avec le temps, même si elles ne sont pas soumises à une tension de claquage (Time Dependent Dielectric Break down, TDDB). L'effet de cette défaillance comme le précise certains travaux de recherche [42] est traduit par des décalages de valeurs des niveaux haut et bas de tension (des erreurs de bit) pour des circuits de technologie CMOS.
- 2- **Injection de porteurs chauds (Hot carrier injection):** dans les semi-conducteurs, les porteurs de charges sont accélérés par les champs électriques. L'énergie produite risque d'être suffisante pour traverser les couches isolantes en structures MOS. Ce mécanisme de défaillance provoque souvent des défauts de fonctionnement intermittents.
- 3- **Electro-migration:** Dans les pistes conductrices très minces, des ions métalliques peuvent se déplacer et entraîner dans le cas le plus sévère une coupure d'une liaison

comme le montre la figure3. Dans le cas le moins défavorable, il est à noter une transformation du matériau le rendant beaucoup plus résistif.



**Figure 2.2 Electromigration**

- 4- Migration sous contrainte mécanique (stress migration):** Les conditions thermiques sur les pistes minces peuvent les amener à la rupture comme le justifie bien la figure 2-2. Cet effet entraîne le déplacement du métal dans la direction du flux des charges électriques. Ceci entraîne en conséquence une dilatation beaucoup plus importante de cette piste métallique dont un côté est plus fin qu'un autre la rendre plus fragile vis-à-vis des changements de température: un refroidissement brusque conduit à la rupture de la piste.

#### **2.4.2 Modèle de faute au niveau des composants : le court-circuit et le circuit ouvert.**

Dans la modélisation de défauts à l'aide de simulateur à bas niveau d'abstraction (au niveau de composant) connu aussi sous le nom de simulateur analogique, la représentation de défauts peut être plus précise que celle au niveau de porte ou switcher grâce au type de simulation employée. Un simulateur de circuit possède une large sélection de résistances, de capacités, et de diodes lesquelles vont donner une représentation assez bonne des types de défauts rencontrés.

Les mécanismes de défaillance qui mènent à un mode de défaillance en circuit ouvert peuvent évidemment être représentés par l'introduction d'une haute résistance au circuit, et similairement en court circuit par une faible résistance selon d'autres études [43, 44,45]. Comme alternative, des commutateurs «**switch**» peuvent être utilisés pour introduire l'effet de faute (un switcher fermé pour modéliser un circuit ouvert, et switcher ouvert pour

modéliser un court circuit).

L'automatisation de ce processus de modélisation va entraîner l'introduction des composants défectueux dans la description du circuit existante, l'exécution de la simulation puis de nouveau leur omission. Seulement, l'utilité de ce genre de modélisation est restreinte aux faibles portions de circuits larges. Pour remédier ce problème, autres alternatives de modélisations furent proposées dont nous relaterons quelques unes les plus importantes

### **2.4.3 Modélisation des défauts dans les circuits analogiques intégrés [46].**

Pour générer des vecteurs de test dans un ensemble électronique, il est nécessaire de modéliser les défauts les plus probables, c'est à dire de leur faire correspondre des types de défauts théoriques créant les mêmes anomalies de fonctionnement du circuit que les défauts réels.

Les modèles sont des représentations des défauts physiques qui peuvent être détectés dans un composant quelconque. Ils peuvent prouver et déterminer la qualité des ensembles électroniques sous test, et le fait de les faire correspondre aux modèles originaux, nous permet d'évaluer quelques informations concernant leur comportement avant et durant le test, pendant leur bon ou mauvais fonctionnement. De cette façon le test devient plus simple et moins compliqué surtout coté analogique.

Pour avoir des modèles de défauts, précis sinon parfait, le simulateur nous offre toutes les possibilités, afin de récolter le maximum de modèles. Le court circuit et circuit ouvert sont présentés respectivement par des résistances très petites et par des résistances de valeurs infinies. Les modèles logiques ont été aussi employés dans les circuits analogiques dans le but d'apporter un plus à la testabilité.

Un système calculateur ne contient pas seulement des circuits numériques, mais aussi une variété de circuits analogiques tels que des amplificateurs opérationnels, comparateurs, détecteurs d'enveloppe, échantillonneurs bloqueurs, etc.

### **2.4.4 Modèles des défauts en analogique.**

La modélisation des défauts en analogiques peut s'effectuer à différents niveaux d'abstraction [20], [21]:

a) Niveau des composants : des modifications des paramètres de ces éléments (valeur de la résistance, de la capacité, tension de seuil d'une diode le gain en courant du transistor etc....) ou injection des courts circuits ou des circuits ouverts aux bornes des ces éléments.

b) Niveau comportemental ou fonctionnel : il s'agit de développement de modèle représentant la fonction ou le comportement du circuit sous une faute particulière. Le circuit est considéré comme une boîte noire et seuls les signaux d'entrées et de sorties sont mesurables et permettent l'élaboration de modèle de fautes. L'avantage de celui-ci réside en sa simplicité et augmente d'une façon conséquente la vitesse de simulation s'il est fait usage d'un simulateur approprié. Seulement ce modèle peut ne pas avoir aucune ressemblance avec le circuit originel et entraîne une déperdition de précision sur les résultats de la simulation

#### **2.4.5 ..... La technique de modélisation employée.**

La technique utilisée pour la modélisation des circuits analogiques dépend des simulateurs logiques qui sont largement utilisés dans les systèmes numériques. Ces logiciels offrent une large gamme d'application depuis le niveau des éléments primitives (transistors, diodes etc. ...), des portes logiques jusqu'aux niveaux des blocs fonctionnels. Pendant très longtemps, le modèle de fautes utilisé était le collage à un état (0 et 1) pour les circuits numériques cette modélisation convenait à la technologie TTL mais, la technologie CMOS, a conduit à plus de perfectionnement de modèles de fautes pour rester près des manifestations des défauts physiques comme s'est décrit plus haut.

Dans le cas de test des circuits analogiques complexes, il est fort recommandé de les subdiviser en sous circuits élémentaires afin de contrecarrer le problème de la complexité.

Un modèle ou plus précisément macro modèle est attribué pour chaque sous circuit avec ses paramètres pour caractériser entièrement sa fonction avec une précision fort acceptable assurant ainsi une meilleure performance. Cependant une diversité de modèles (macromodèles) a émergé pour décrire aussi bien le bon comportement du composant/circuit (fault-free model) ou le mauvais comportement (fault model) a été exploité et chacun d'eux présente ses avantages et ses inconvénients qu'on étalera à travers ce chapitre.

### **2.5 Macro modèles de fautes**

Une méthode de simulation de défauts des circuits intégrés a été développée pour fournir la façon la plus simple de décrire un défaut au niveau macro. Cette méthode permet de présenter un bloc avec un défaut particulier par un macromodèle basé sur la fonction de transfert du circuit défectueux. Comme ce macromodèle a la même forme que celui du bon circuit, et il est valable pour toutes les autres fautes pour ce bloc, la substitution du modèle de défaut devient très simple et seulement la forme de la fonction de transfert est altérée. Il existe aussi bien des macromodèles pour les simulateurs analogiques que pour ceux numériques,

### 2.5.1 Macro-modélisation numérique.

La plus importante utilisation de la macro-modélisation dans les applications numériques est la vérification de la fonction et le temps critique du chemin de propagation dans les SSI et les VLSI. La correction des erreurs de conception est directe avec l'utilisation des simulateurs mais un compromis précision-rapidité doit être fait [47]. Le modèle de défaut qui reste incontestablement le plus approprié à tous les niveaux de description des circuits est le collage à (Stuck at model).

### 2.5.2 Modélisation numérique des défauts des circuits analogiques.

La modélisation numérique des circuits analogiques est introduite comme alternative au processus de simulation électrique qui a montré son incapacité à simuler des circuits complexes. Cette approche numérique consiste à établir des modèles numériques à partir des fonctions électriques de chaque portion analogique du circuit de cette envergure.

L'objectif ciblé par ce genre de modélisation est de produire un test adéquat pour des circuits analogiques à partir d'un modèle équivalent numérique où la détection des défauts sera assurée par emploi du modèle de **collage à 1 ou à 0**.

Des travaux dans ce même axe [46, 29, 48] ont été effectués sur différents circuits analogiques à savoir l'amplificateur inverseur, amplificateur différentiel, comparateur etc. et les résultats obtenus du test basé sur ce type de modèle numérique étaient prometteurs puisque la couverture de fautes a atteint pour certains circuits les 76%. Seulement certaines complications ont été soulevées pour différents cas de figures dont on abordera quelques unes dans le prochain exemple. Il est alors indispensable de connaître ce type d'approche de modélisation qui sert d'outil pour le test des circuits analogiques.

## 2.6 Exemple du modèle numérique pour circuit analogique particulier.

Un modèle numérique pour un circuit analogique est une représentation logique (à l'aide de portes logiques) basée sur un choix judicieux des niveaux logiques et les seuils (thresholds) des entrées et des sorties. Une fois ces derniers sont définis et les niveaux logiques sont connus, une table de vérité (table de Karnaugh) s'établit et la configuration à l'aide de portes logiques se déduit. A titre d'exemple, un amplificateur inverseur ayant un gain différent de l'unité et opérant à +/- 5V dispose d'une entrée inverseuse A et une sortie B (voir fig. 3-21-a) a conduit à la construction de la table de vérité donnée en

figure 3-21-c avec sa configuration logique équivalente. Deux ensembles de niveaux de tensions d'entrée ( $V_e$ ) et de niveau de seuil de sortie ( $V_s$ ) peuvent se présenter et dont un seul suffit pour une telle construction. Ses deux ensembles de niveaux sont :

1/  $0V \leq V_e \leq 5V$  et  $V_s = -2V$  ;

2/  $-5V \leq V_s \leq 0V$  et  $V_s = 2V$ .

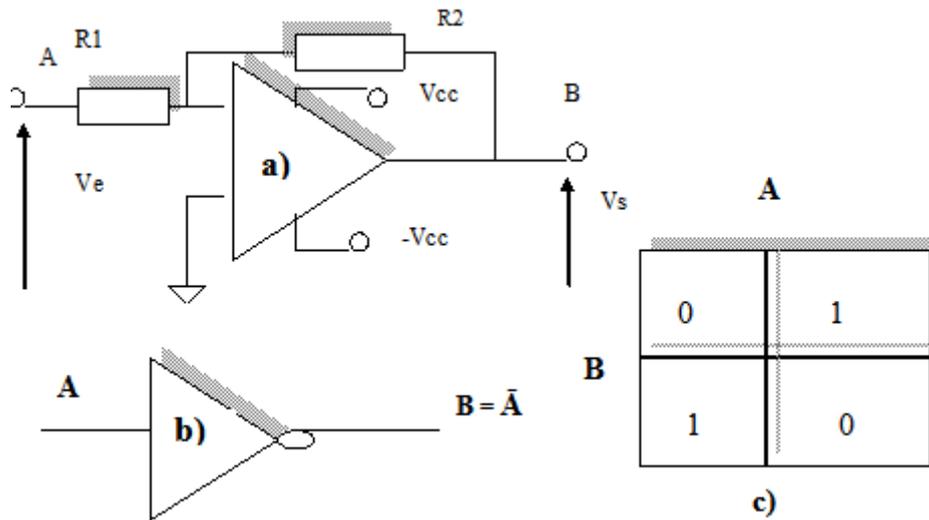


Figure 2.3 :

a) Amplificateur inverseur    b) configuration logique équivalente,    c) table de vérité

L'application du modèle **collage à 1/ 0** aux entrées et sorties pour une telle configuration logique équivalente a contribué à une amélioration de la couverture de fautes par une combinaison de 3 vecteurs de test d'entrée. Ces fautes sont en réalité des fautes dures ou catastrophiques (des courts circuits ou circuits ouverts) dont leur effet à la sortie du circuit est en corrélation avec ce type de modèle comme le montre le tableau 2 ;

Test n°	seuil de tension	niveau logique A	niveau logique B	type de faute collage à
1	-5V	0	1	A/1 ou B/0
2	0V	1	0	A/0 ou B/1
3	+5V	1	0	A/0 ou B/1

Tableau 2.2 : emploi de modèle de « collage à » pour amplificateur inverseur.

A partir de ce résultat, on constate que les tests 2 et 3 détectent le même type de fautes (A/0 ou B/1) seulement ils couvrent différentes fautes internes ce qui augmentera le taux de couverture en conséquence. Dans le même contexte d'idées, la combinaison des tests 1 et 2 a permis une couverture de fautes d'environ de 62% du nombre total égal à 96 fautes.

Dans le but de maximaliser cette couverture, le test n°3 fut employé après un changement de niveau de seuil du signal de sortie et a conduit ensemble avec les 2 tests précédents à une nette amélioration du taux de détection de faute puisque celui-ci est passé à 78%

## **2.7 L'approche D.C. pour la Génération du Dictionnaire de Défauts [6].**

L'approche D.C. pour la génération du dictionnaire de défauts est considérée comme un excellent outil pour le diagnostic et la détection des défauts sévères (le court-circuit et le circuit ouvert) dans les circuits analogiques non linéaires [50].

Un dictionnaire de défaut pour un circuit donné est un ensemble de valeurs de tensions continues pré-calculées, qui servent à détecter un défaut dans un pareil circuit lorsque celui-ci présente une défaillance. Pour cela, l'ingénieur de test doit suivre une procédure adéquate afin d'arriver à ce but.

Dans l'approche D.C., des tensions nodales sont générées grâce à un simulateur de circuit, pour le fonctionnement nominal: le fonctionnement sous des défauts prédéfinies du circuit concerné. Cet ensemble de valeurs de tensions continues forme le dictionnaire de défauts qui sera stocké dans un ATE (Automatic Test Equipment). Le circuit défaillant est alimenté par les mêmes tensions que celles utilisées en simulation et les nouvelles tensions résultantes sont alors mesurées et stockées. Ces dernières sont comparées avec celles du dictionnaire de défauts pour déterminer les défauts du circuit testé. Théoriquement, les entrées appliquées et les sorties mesurées dans un dictionnaire de défaut peuvent être soit continues soit variables. L'approche D.C. se fait selon 2 étapes distinctes l'analyse pré-test et l'analyse post-test. et elle a les avantages de la simplicité de simulations des circuits et l'aisance dans la mesure des sorties.

## **2.8 La simulation des fautes.**

La simulation des fautes est considérée comme un outil nécessaire dans tout processus de génération de vecteur de test où l'objectif visé est l'amélioration de la qualité de test. Ceci sera concrétisée par la qualité du jeux des stimuli assurant un taux de couverture de fautes maximal en se servant de modèles de fautes électriques ou logiques tels que le collage à 1 ou 0 pour les circuits

numériques ou le court circuit ou le circuit ouvert pour les circuits analogiques. Par ailleurs, il constitue un apport fort appréciable pour le diagnostic des défauts de circuits à travers les techniques de localisation et d'isolation basées sur les résultats de la simulation des fautes [6].

La simulation de fautes analogique peut s'effectuer selon différents modes d'opération et selon le besoin d'évaluation de la couverture de fautes et de l'efficacité de l'ensemble des signaux continus ou alternatifs employés dans le test. On envisage ici les différents types de simulation de fautes analogiques et leur degré d'importance dans le test:

- simulation des fautes en mode DC pour les circuits non linéaires.
- Simulation de fautes en régime AC pour les circuits linéaires;
- Simulation de fautes en mode transitoire ou temporel.

Le mode DC doit être considéré en premier lors d'un processus de test [51] car il est simple à développer et que les deux autres modes ne seront introduits que lorsque le premier type de test échoue car ils sont plus difficiles à générer et plus coûteux à appliquer.

Tout comme il existe toute une classe de simulateurs permettant de prédire la réponse à la sortie d'un circuit en exemption de défaillance. Il y a aussi une autre classe de simulateurs appelée classe de simulateurs de fautes. Le rôle de ces derniers est de prédire le comportement d'un tel circuit fonctionnant sous condition de défaillance. L'injection de celle-ci est rendue possible grâce à l'option d'insertion de modèle de faute que peut fournir un de ces outils informatiques (softwares).

En numérique, la simulation de faute fut démarrée par implémentation dans le simulateur à valeur réelle l'habilité de maintenir la sortie d'un circuit logique à un état logique constant (0 ou 1) afin de simuler le collage à 1 ou à 0. Après application du chaque vecteur de test, la réponse augurée du circuit fautif sera comparée avec la réponse de référence (du circuit libre de fautes) et en cas de différence flagrante ce stimulus est efficace pour la détection de cette faute. Seulement, la complexité croissante des circuits a rendu cette technique fastidieuse car elle devient consommatrice de temps. Afin d'apaiser la sévérité de ce problème plusieurs techniques ont été développées entraînant soit une simulation parallèle ou une simulation déductive ou une simulation concurrente. Ces types de simulations peuvent se définir comme suit:

✓ **Simulation parallèle [52]: il s'agit d'une simulation de N copies d'un même circuit dont une copie (copie de référence) est sans défaut et N-1 sont affectés de défauts. Le nombre de circuits**

défaillants à simuler en parallèle dépend de la longueur de la séquence binaire ou chaque bit permet d'identifier une faute particulière dans le circuit. En conséquence, s'il y a  $M$  fautes à simuler, le nombre de simulation à exécuter par vecteur de test sera de  $M/(N-1)$ . Ce nombre pourra être réduit d'avantage si les fautes indépendantes sont simulées ensemble. De même pour les circuits disposant d'un certains nombre de sortie  $P$  et ayant le même nombre de fautes à simuler, le nombre de simulation à effectuer passera à  $M / ((N-1) .P)$ .

✓ **Simulation déductive:** dans ce type de simulation, les copies de circuits fautifs se déduisent à partir de la copie de circuit sans faute au niveau de ses nœuds internes ou ses sorties primaires . Puisque toutes les fautes sont détectables simultanément, il est alors nécessaire de procéder à une seule simulation par vecteur de test ce qui est contraire à la simulation précédente ou  $M/(N-1)$  simulations sont indispensables. En général, bien que le temps nécessaire pour une seule simulation déductive est supérieure à celui d'une seule simulation de la méthode parallèle, il est inférieur à celui nécessaire pour  $M / (N-1)$  simulations exécutées. Seulement cette méthode de simulation déductive exige beaucoup de mémoire de locations des résultats de ce processus et de taille des listes de fautes qui varie d'un type de circuit à un autre [53].

✓ **Simulation concurrente:** ce genre de technique est une combinaison des deux précédentes. Seulement, ce processus de simulation concerne les circuits fautifs qui présentent des valeurs de sortie différentes de celle du circuit sans défaut. Quoique cette technique exige plus d'espace mémoire que le requiert la simulation déductive, elle reste une technique rapide [54]. Ces techniques ont incontestablement prouvé leur efficacité de réduire le temps de simulation seulement elles restent difficilement accessibles par les circuits analogiques [55].

Dans le domaine de l'analogique, il existe peu de simulateurs permettant la prise en charge de fautes et qui sont FSIM, ANAFAULT [56]. Ces simulateurs emploient la même approche, que celle utilisé par SPICE, ELDO tous deux simulateurs électriques incorporant plusieurs options qui permettent la simulation des fautes à un niveau bas d'abstraction. La méthode s'effectue traditionnellement suivant les étapes suivantes:

- ✚ Simulation du circuit sain (en absence de défaut).
- ✚ Injection de faute a partir d'une liste préétablie dans le circuit.
- ✚ - Simulation du circuit fautif .pour un ensemble de vecteurs test
- ✚ - Comparaison des résultats trouver dans les étapes 1 et 3 et chercher les fautes détectables

- ✚ Recherche des bons stimuli mais au nombre restreint et pourvoyant le maximum de fautes possibles.

Cependant ce genre de simulateurs est caractérisé par la lenteur d'exécution d'une part et d'autre part la complexité des circuits croissante à cause de moyenne et haute échelle d'intégration des composants sur une même puce de silicium (circuits de type MSI et LSI etc....) rend finalement la simulation de fautes analogiques très coûteuse en temps CPU. Outre le problème de taille de circuits, la prise en compte des dispersions technologiques des paramètres la capacité de simuler en mode DC, AC et transitoires sont devenues des points gênants pour l'optimisation du processus de simulation des fautes analogiques. Dans de nombreux travaux de recherche, des techniques de simulations de fautes émergentes ont été proposées dans le but d'une optimisation du temps de simulation pour certaines classes de circuit [3] et selon différents aspects dont on citera quelques uns à travers les sections qui viennent.

## 2.9 L'emploi de modèle de faute logique (stuck-at) :

Les modèles logiques ont fait preuve d'utilisation plus ou moins efficaces dans la simulation de fautes des circuits analogiques [29], [48], [46]. Chacune de ses références utilise la technique de modélisation selon différents angles. Ils vont de la modélisation structurelle du circuit analogique en rapprochant sa structure interne par une structure logique équivalente (en terme de portes logiques) jusqu'à la modélisation fonctionnelle ou comportementale ou un circuit logique équivalent est déduit à partir de la fonction analogique. L'approche de modélisation est la même que celle employée en numérique et qui consiste à l'instauration de seuil des tensions délimitant les niveaux logiques.

L'analyse de l'effet d'une faute analogique injectée dans le circuit a conduit à son assimilation à un collage à 0 ou à 1 au niveau de l'entrée ou la sortie du circuit logique équivalent. Les résultats obtenus sont très probants quant à l'amélioration du taux de détectabilité pour certains circuits spécifiques tels que les amplificateurs, comparateurs et autres. Seulement cette méthode présente certains points gênants :

- 1- l'imprécision du modèle puisqu'il fournit à la sortie une réponse maintenue à un seul état logique ignorant toute altération réelle sur ce signal.
- 2- Le choix de la tension de seuil ne s'applique pas de la même façon que les circuits numériques : il varie d'un type de circuit à un autre.

3- Certains circuits analogiques tels que les oscillateurs en anneau sont difficiles à modéliser par cette approche.

## **2.10 La simulation des fautes par emploi des techniques analogiques :**

### **a) simulation par fonction de transfert :**

Cette technique de simulation consiste à la modélisation des circuits analogiques linéaires couvrant une large gamme d'applications dans le domaine de la télécommunication, biomédicale et autres. Le principe consiste à la modélisation des éléments primitifs actifs (transistors bipolaires, ou à effet de champs, diodes etc... ) opérant en régime faibles signaux. Le modèle équivalent de ces éléments est linéaire autour du point de repos (de polarisation). Cette approche a permis l'élaboration d'algorithmes efficaces dans la simulation des fautes dans les circuits analogiques [6].

### **b) Technique de modélisation orientée défauts :**

Plusieurs méthodes ont été utilisées pour réduire le temps résultant de la simulation des fautes d'une liste [57, 58]. Cette réduction de temps a été atteinte par le fait de considérer :

- 1- uniquement les fautes concrètes ou leur occurrence de se produire est fort probable.
- 2- les fautes non redondantes car leur détection entraîne la détection de leur équivalente. Il est possible de procéder à une compression de la liste des défauts en classe de fautes équivalentes.

### **c) Simulation des fautes basée sur les limites de tolérances :**

Dans ce genre de simulation, des méthodes proposées pour certains travaux [3] sont basées sur la recherche des limites extrêmes des intervalles de sortie aussi bien pour le circuit sain que pour le circuit fautif sans passer par la méthode de simulation basée sur le calcul statistique (méthode de Monte Carlo). Dans la première référence, il est supposé que ces limites extrêmes de cet intervalle sont obtenues à partir de la prise en considération des limites extrêmes de tolérances des paramètres ce qui n'est pas toujours vraie pour tous les circuits analogiques. Cette méthode basée sur les intervalles mathématiques comme mentionnée dans la deuxième référence pour remédier au problème soulevé préalablement [6].

### **d) Simulation de fautes en mode DC :**

L'accélération du processus de faute de simulation est rendu possible par emploi d'outils de calcul spécifiques que procure le simulateur électrique en vue de surmonter les difficultés rencontrées lors de la résolution des équations non linéaires qui dans la plus part des temps exigent

beaucoup d'itérations pour converger. Ces outils comme les présentent certains travaux [59], [57], [58], [55], [11], [60] peuvent se résumer ainsi:

- 1- Méthode de calcul du pivot complémentaire employé en mode DC.
- 2- Méthode prédictive de la réponse fautive causée par chaque faute utilisant la première itération Newton-Raphson dans la simulation du circuit. Le résultat constaté était une réduction de nombre d'itération moyen de facteur de 4 par circuit analogique.
- 3- Méthode de réduction du nombre d'itérations Newton-Raphson de la simulation de faute en mode DC et transitoire. L'approche adoptée permet une détection précoce dans le processus itératif des fautes non détectables sans attendre la convergence des résultats de la simulation qui dans la plus part des cas tarde à venir. Il s'agit de l'emploi d'un algorithme de calcul de distance à chaque itération entre le circuit fautif et le circuit sans faute. Le calcul se fait alors pour toutes les itérations précédentes et si la distance calculée est inférieure à un certain seuil la simulation est arrêtée. Les résultats escomptés par une telle approche sont une réduction de temps plus de 50%.

**e) La simulation de fautes au niveau comportemental [6] :**

L'usage d'une simulation de fautes au niveau comportemental vise à réduire d'avantage le temps CPU [61, 62]. Ceci a été possible par l'extension de l'application des simulateurs supportant des langages de description de haut niveau des circuits analogiques AHDL (Analog Hardware Description language). Cependant, des langages similaires tels que le VHDL, Verilog ont fait apparition il y a bien longtemps dont l'application était limitée aux circuits numériques. Ces nouveaux simulateurs comme le précisent certains travaux [63,64] ont été implémentés à un niveau d'équations basé sur les équations d'états.

Des macromodèles comportementaux ont été aussi utilisés pour améliorer d'avantage la rapidité du processus de simulation de fautes. La technique consiste à subdiviser le circuit en sous circuits selon l'approche Top-Down en descendant au niveau des transistors. On procède ensuite à la modélisation comportementale de ses sous circuits en présence et en absence de fautes en utilisant le langage HDL A ou le VHDL-AMS. Cette technique a été appliquée sur deux circuits complexes la PLL et le filtre Leapfrog conçu à partir de six amplificateurs avec des boucles locales et globales et les résultats étaient très probants [65], [66].

### **2.11 le type de modèle employé dans notre travail :**

Dans notre étude de modélisation de défauts dans le circuit amplificateur-inverseur, le choix est porté sur adjonction de 2 types de modèle : électriques et logique.

Le premier est adopté au circuit analogique simulé par le simulateur électrique PSPICE, qui accepte l'implémentation d'un défaut physique par un défaut électrique (court circuit ou circuit ouvert). Le modèle équivalent de l'un d'eux est respectivement une résistance à faible valeur ou très grande valeur.

Le deuxième type de modèle est une représentation binaire de la réponse du circuit sous test. Cette représentation est constitué d'un certain nombre de bits qu'on considère comme une signature binaire (exprimée par des uns et /ou zéros) qui exprime l'état du circuit : sain ou défaillant. Une défaillance du circuit peut être interprétée par un changement d'état de un ou plusieurs bits de a signature. Celle-ci est délibérément choisi pour être exploitée par la technique de test BIST qui de nature de test logique. Plus de détails sur ce type de modélisation est donné au chapitre 4.

### **2.12 Conclusion.**

A travers ce chapitre, une étude générale sur la simulation de fautes est donnée pour mettre en exergue son rôle primordial dans le processus de test. Son exploitation aussi bien avant ou après ce processus lui permet de jouer le premier rôle dans la fiabilité de produits finis : minimisation des défauts et amélioration de la performance avec le moindre coût. Cette procédure de simulation peut intervenir à différents niveaux de la conception ou de la fabrication des circuits et s'effectue de façons distinctes selon le type de simulateur employé. Ceci est dû à cause des défauts qui surviennent aux différentes phases de la conception. Ces défauts peuvent être modélisés en concordance avec la nature de fautes (électrique ou logique) et selon la possibilité d'implémentation de modèle fournie par le logiciel de simulation. Dans notre cas d'étude, un modèle de type mixte est employé car l'environnement où la simulation est effectuée est hybride par l'association de deux logiciels de développement différents : le PSICE et Simulink.

# **Chapitre 3**

## **La simulation**

### 3.1 Techniques de simulation

La simulation est un processus qui intervient pour résoudre certains problèmes apparaissant dans la fabrication de circuits complexes à différents niveaux d'abstractions. Elle contribue beaucoup plus dans leur conception afin de vérifier leur fonctionnalité et éliminer les erreurs qui peuvent l'affecter. En conséquence, la simulation prend part à leurs différents niveaux de fabrication pour fournir d'information utiles aux différents aspects de conception. Par exemple, la simulation à un haut niveau d'abstraction sert à vérifier l'aspect fonctionnel du circuit conçu et elle est rapide. Seulement elle perd de la précision à cause de modèles simplifiés auxquels on fait appel aux modèles mathématiques pour décrire la fonction du circuit. En revanche, la simulation au bas niveau d'abstraction, se consacre beaucoup plus à la performance du design associé aux éléments primitifs du circuit et leur implémentation physique. La précision en est l'arme aiguisée de ce processus qui est malheureusement parfois très lent. Ces situations ont drainé une panoplie de techniques de simulation afin de contrecarrer soient les problèmes de temps, de complexités de circuits ou de leurs natures numériques et analogiques.

#### 3.1.1 .Simulation logique [67] [68] [8].

La simulation logique est une technique de simulation rapide basée sur l'évaluation de fonctions logiques et la propagation d'événements dans le modèle. Les signaux ne peuvent prendre qu'un nombre fini d'états et le temps est représenté par une valeur discrète, un multiple entier d'une unité de base appelée temps de résolution minimum (MRT - minimum resolvable time). Un événement est un changement de valeur sur un signal. La Figure 3-1 donne l'algorithme général de la simulation dirigée par les événements (event-driven simulation).

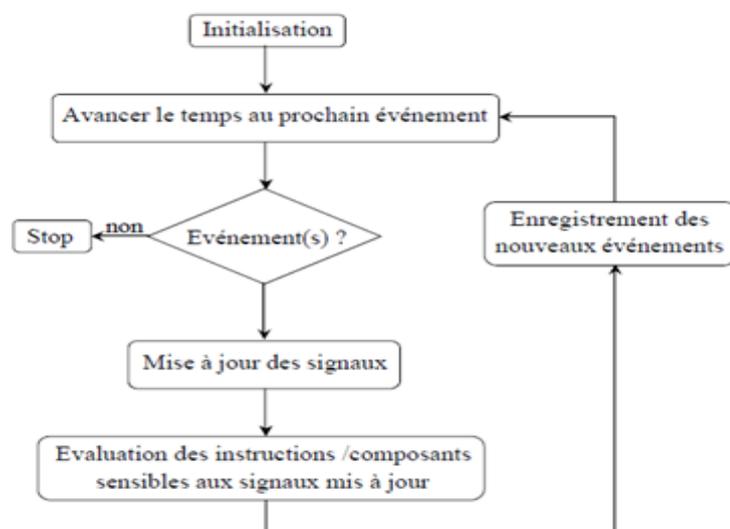


Fig. 3-1 : Algorithme de Simulation dirigée par événements [68].

La simulation démarre par l'affectation de valeurs initiales à tous les signaux. Il faut noter que l'état initial du modèle n'est pas nécessairement un état cohérent, ou stable car il n'y a pas encore de propagation des valeurs aux entrées primaires du modèle. Le temps de simulation est ensuite avancé jusqu'au moment du prochain événement prévu. Les valeurs des signaux ayant un événement à ce moment-là sont mises à jour et toutes les instructions ou les composants du modèle concernés par ces mises à jour sont réévalués. Ceci aboutit potentiellement à de nouveaux événements sur des signaux, au même instant ou à des temps futurs. La boucle se répète ainsi jusqu'à ce qu'il n'y ait plus d'événements à propager dans le modèle.

### 3.1.2 Simulation analogique [49] [67] [68].

La simulation analogique est beaucoup plus complexe que la simulation logique et requiert ainsi plus de ressources (temps de calcul, mémoire). L'archétype du simulateur analogique, ou électrique, est le programme SPICE.

La simulation analogique implique la résolution d'équations différentielles et algébriques linéaires et non linéaires. Les solutions sont des tensions entre les nœuds du circuit et les courants dans les branches de ce dernier. Normalement seulement un sous-ensemble de toutes les tensions et de tous les courants est requis.

La simulation analogique permet plusieurs types d'analyses :

- **L'analyse temporelle** (transient analysis) calcule les réponses temporelles du circuit (tensions et courants en fonction du temps) relativement à un ensemble de stimulus (sources et conditions initiales).
- **L'analyse DC** (direct current) calcule l'état du circuit pour un ensemble de stimulus fixes après un temps infiniment long (steady state). L'analyse DC est utile pour calculer le point de repos, ou de polarisation du circuit, des fonctions de transfert, la résistance d'entrée et de sortie du circuit, les sensibilités de variables de sortie en fonction de paramètres du circuit.
- **L'analyse AC** (alternative current) calcule les réponses fréquentielles du circuit en régime de petits signaux sinusoïdaux appliqués autour du point de repos du circuit.

L'analyse AC est utile pour calculer des fonctions de transfert (p. ex. gain en tension, trans-impédances) en fonction de la fréquence et des conditions de polarisation du circuit. Elle est aussi utile pour analyser l'influence du bruit et déterminer les caractéristiques de distorsion du circuit [49].

### 3.1.3 La simulation mixte logique-analogique [6] [7].

Le tableau 3-1 récapitule les caractéristiques principales de la simulation logique et de la simulation analogique.

**Tableau 3-1 : caractéristiques de la simulation logique**

Caractéristique	Simulation logique	Simulation analogique
Variables/inconnues	Signaux logiques	Tensions, courants, etc.
Valeurs des inconnues	Quantifiées ('0', '1', 'X', 'Z', etc.)	Réelles
Calcul de l'état du circuit/modèle	Evaluation de fonctions logiques	Résolution d'équations différentielles algébriques non linéaires
Etat initial (t=0)	Pas nécessairement un état stable	Etat stable (point de repos DC) requis
Itération à un temps donné	Affectation de signaux avec délais nul (délai delta)	Résolution de systèmes non linéaires
Représentation du temps	Discret, multiple du MRT	Réel
Gestion du temps	Dirigé par événement	Continue avec pas d'intégration variable
Contrôle du temps temporel	Evénements sur les signaux	Erreur de troncature locale ou équivalente
Types d'analyses	temporelle	Temporelle, DC, AC

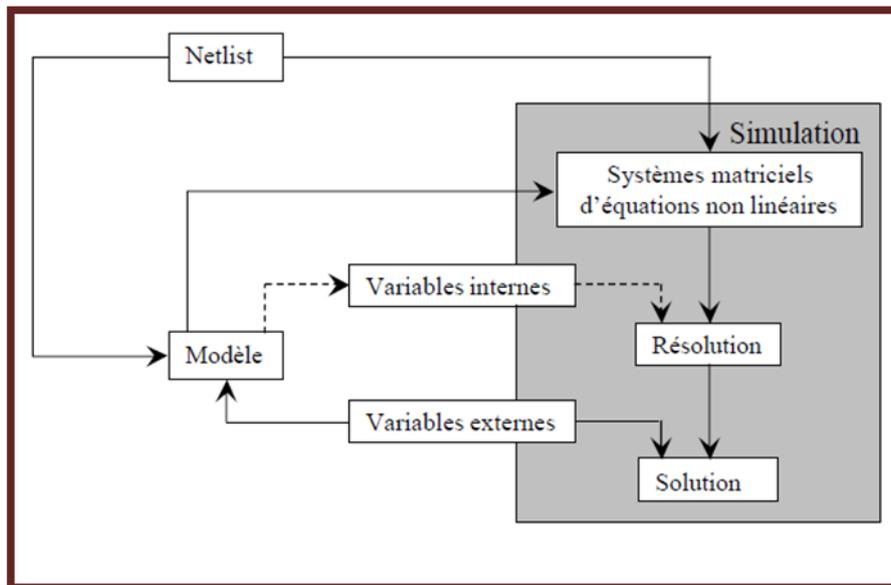
Au vu de cette table, la simulation mixte logique-analogique sert résoudre les problèmes suivants:

- **Conversions entre valeurs logiques et analogiques** : Il s'agit de définir des conversions qui aient un sens physique sans provoquer une perte de précision ou de non-convergence durant la simulation. Il est à noter que ces conversions ne sont que des artefacts de la simulation mixte et ne constituent aucunement des composants physiques dans le circuit.
- **Etat initial (à  $t = 0$ ) du circuit/modèle** : L'analyse temporelle en simulation analogique requiert le calcul d'un point de repos DC correspondant à une solution des équations du circuit. Dans le cas antinomique, la suite de l'analyse a de fortes chances de diverger ou au mieux d'être incorrecte. La simulation logique n'est pas aussi sévère car l'avancement du temps mettra le circuit dans le bon état.
- **Gestion du temps** : Non seulement le temps est représenté différemment en simulation logique et en simulation analogique, mais il est géré selon des critères qui ne sont pas communs. Il s'agit donc de définir des points de synchronisation entre les deux échelles de temps de manière à prendre en compte correctement les interactions logiques-analogiques.
- **Support des analyses possibles en simulation analogique** : Seule l'analyse temporelle est réellement applicable de manière commune aux deux modes. Il s'agit donc de définir l'état de la

partie logique lorsque l'on veut procéder à une analyse DC ou AC de la partie analogique. Une manière simple est de considérer la partie logique comme stable, c'est-à-dire que les signaux logiques agissant à l'interface logique-analogique doivent être considérés comme des sources constantes (après conversion des valeurs logiques en valeurs analogiques) [49].

### 3.2 Procédure interne d'un simulateur

D'une manière générale, les simulateurs se servent de modèles conçus à partir des équations qui représentent le comportement physique du composant. Leur précision dépend du nombre des paramètres et de la complexité des équations [69]. Le principe de ces paramètres est basé sur des méthodes d'extraction mathématique à partir des courbes expérimentales qui nécessitent des mesures précises.



**Fig. 3-2 : organigramme des différentes phases de résolutions d'un simulateur.**

Les simulateurs sont destinés à l'analyse continue, fréquentielle et temporelle des circuits. Ils disposent pour cela l'algorithme de résolutions numériques des équations différentielles. La figure 3-2 décrit la procédure engagée lors d'une simulation, les circuits sont décrits dans un simulateur par une liste des interconnexions (Netlist), qui indique comment sont connectés les éléments. A chaque modèle est associé un système d'équations qui décrivent les lois aux différents nœuds (tension et courant). Le simulateur résout ses systèmes d'équations non

linéaires par des méthodes d'intégration numériques, des techniques itératives et des méthodes de résolution matricielle.

### **3.3 Les trois analyses basiques de la simulation électrique**

Pour caractériser complètement un circuit électrique, au moins trois types d'analyse doivent pouvoir être réalisés par un simulateur électrique: une analyse statique, une analyse transitoire et une analyse alternative pour faibles signaux.

#### **3.3.1 L'analyse statique**

L'analyse statique sert à déterminer les points de fonctionnement (la polarisation et les points de repos correspondant) ainsi que les caractéristiques de transfert des circuits électriques linéaires et non linéaires. Pour ce faire, les composants réactifs sont neutralisés (les condensateurs sont remplacés par des circuits ouverts et les inductances sont remplacées par des courts-circuits) et toutes les sources indépendantes sont considérées comme statiques. Si l'opérateur  $F$  est linéaire la solution est directement déterminée par une méthode numérique de résolution d'un système d'équations linéaires, sinon le système est préalablement linéarisé par une méthode numérique d'analyse non linéaire.

#### **3.3.2 L'analyse Transitoire**

L'analyse transitoire est utilisée pour déterminer la réponse temporelle d'un circuit pour une durée d'observation  $[0, T]$ . Pour ce faire, cette intervalle est discrétisé afin d'obtenir les instants de calcul  $[0, t_1, t_2, \dots, T]$ . Ensuite, partant d'un jeu de valeurs initiales spécifié par l'utilisateur ou par une analyse statique, pour chaque instant de calcul une intégration numérique est effectuée par une méthode numérique d'intégration afin de transformer le système d'équations différentielles en système d'équations algébriques. Enfin, le système est linéarisé par une méthode numérique d'analyse non linéaire pour délivrer la solution par une méthode numérique de résolution d'un système d'équations linéaires.

#### **3.3.3 L'analyse alternative pour faibles signaux.**

L'analyse linéaire alternative pour faibles signaux est utilisée pour déterminer les caractéristiques fréquentielles d'un circuit. Dans ce cas, tous les éléments actifs non linéaires du circuit sont modélisés par un circuit équivalent linéaire faible signal autour d'un point de fonctionnement déterminé par une analyse statique. Tous les stimuli sont sinusoïdaux et de même fréquence mais peuvent avoir des phases relatives différentes. Les impédances ou les

admittances sont mises sous leur forme opérationnelle et évaluées sur l'axe imaginaire du plan de la variable complexe  $P$  ( $P=j\omega$ ). Ainsi, l'impédance d'un condensateur est considérée sous la forme opérationnelle  $Z_C = 1/jC\omega$ , l'impédance d'une inductance est considérée sous la forme  $Z_L = jL\omega$ .

### 3-4 Les différents types de Simulateurs :

#### 3.4.1 Simulateurs Mixtes, Langages de modélisation numérique :

Les langages de modélisation numérique sont les langages de description matérielle de haut niveau communément appelé HDL (Hardware Description Language). Un langage HDL est une instance d'une classe de langage informatique ayant pour but la description formelle d'un système électronique.

Le HDL est alors un langage de description de circuits logiques en électronique, utilisé pour la conception d'ASICs (Application-Specific Integrated Circuits) et de FPGAs (Field-programmable Gate Array), le HDL peut généralement :

- Décrire le fonctionnement du circuit
- Décrire sa structure,
- Assurer la documentation,
- Vérification de Netlist (LVS),
- Tester le circuit et le vérifier par simulation

Il existe un grand nombre de HDL nous pouvant citer quelques exemples des langages les plus connus comme VHDL et Verilog.

#### 3.4.2 Le VHDL :

Le VHDL (abréviation de l'expression anglaise : *Very High Speed Integrated Circuits (VHSIC) Hardware Description Language*) est un langage moderne, lisible et puissant de description du matériel destiné à décrire le comportement et/ou l'architecture d'un système électronique numérique. L'intérêt d'une telle description réside dans son caractère exécutable : une spécification fonctionnelle décrite en VHDL peut être vérifiée par simulation, avant que la conception détaillée ne soit terminée. L'une des particularités du VHDL provient du fait qu'il est possible d'exprimer facilement le parallélisme à l'intérieur d'un circuit. La syntaxe du VHDL est originaire du langage ADA, dont les mots clefs ont été adaptés à la conception de matériel.

VHDL est maintenant le langage de description matérielle majoritairement utilisé par les entreprises européennes alors que Verilog est souvent préféré dans les entreprises américaines.

### 3.4.3 Langage de modélisation Mixte Multi-Domain (VHDL-AMS) :

À l'aide de ces langages de modélisation de haut niveau, les différentes phases de conception peuvent être optimisées.

Ces langages permettent de traiter indifféremment des modélisations logiques, analogiques ou mixtes au sein d'un même composant ou système. Par ailleurs, la philosophie de conception de ces langages et leurs jeux d'instructions en font des langages intrinsèquement multi-domaines qui gèrent les équations implicites liées au fonctionnement d'un circuit. Mais, ils ont des limitations.

Un certain nombre de ces langages de modélisation ont été développés avec succès, comme les langages : VHDL-AMS, Verilog-AMS, MAST, Modelica et Bond Graph.

### 3.4.4 VHDL-AMS

VHDL-AMS inclut toutes les propriétés du VHDL standard, avec en plus, la capacité de décrire les systèmes mixtes (analogiques et numériques) par le biais de modèles multi-abstractions, multidisciplinaires, hiérarchiques à temps continu et à événements discrets à l'aide d'équations différentielles ordinaires (figure 3-3 ).

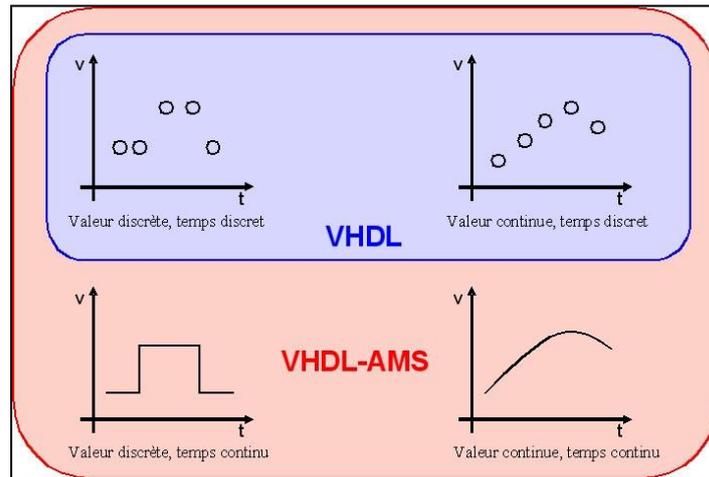
VHDL-AMS est conçu pour le multi domaine : des terminaux de connexion sont liés à des grandeurs physiques qui respectent implicitement les lois de Kirchhoff généralisées. Des quantités analogiques libres peuvent prendre n'importe quelle unité physique.

VHDL-AMS permet ainsi de modéliser et de simuler le système de puissance complet. Il permet de modéliser tout système dont le comportement peut être décrit par un ensemble d'équation différentielles algébriques et/ou ordinaires (ODE, ADE : Ordinary Differential equation, Algebraic Differential Equations) qui ont le temps comme variable indépendante.

Ces équations peuvent être écrites sous la forme :  $F(x, dx/dt, t)=0$ , où  $x$  est le vecteur d'inconnues.

Ce langage non propriétaire présente l'avantage de proposer un langage commun indépendant des fournisseurs et de la technologie. Du point de vue technique, il permet une haute modularité facilitant les descriptions hiérarchiques.

Les systèmes, qui étaient modélisés sous Matlab, VHDL et Spice peuvent aujourd'hui être modélisés sous VHDL-AMS en utilisant un seul langage, puisqu'il offre la possibilité de faire travailler simultanément un simulateur à événements discrets et un solveur d'équations différentielles.



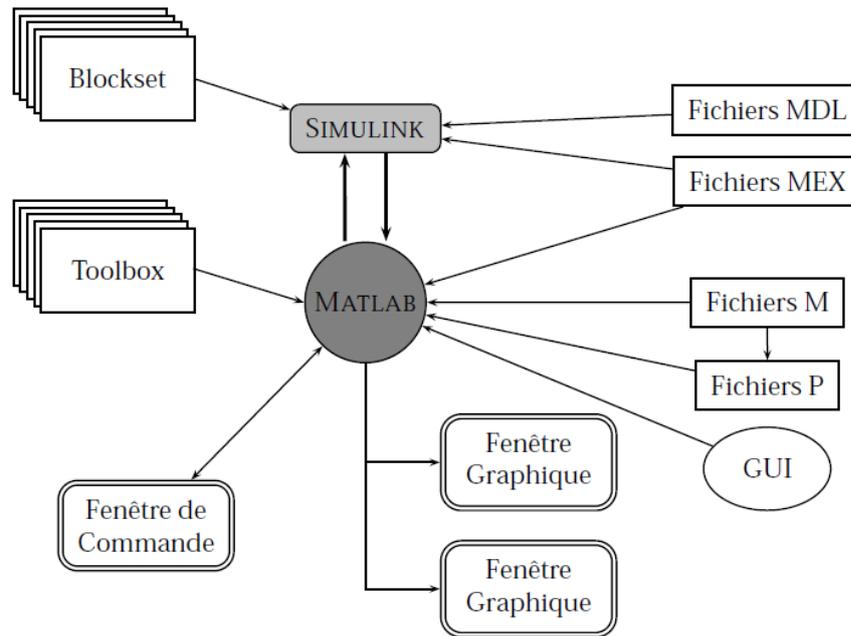
**Figure3-3 : entre VHDL et VHDL-AMS**

### 3.5 Introduction au MATLAB

La première version du Matlab est développée aux universités de New Mexico et de Stanford dans les années 1970, utilisant la théorie des matrices, l'algèbre linéaire et l'analyse numérique ; maintenant il devient le programme le plus puissant pour les ingénieurs.

#### 3.5.1 MATLAB :

- est un logiciel de calcul matriciel à syntaxe simple ;
- peut être considéré comme un langage de programmation adapté pour les problèmes scientifiques, grâce à ses fonctions spécialisées ;
- est un interpréteur, car ses instructions sont interprétées et exécutées ligne par ligne.
- possède des bonnes capacités graphiques pour présenter des résultats ou pour créer des applications
- peut être intégré avec du code C ou FORTRAN.
- fonctionne dans plusieurs environnements tels que UNIX/X-Windows, Windows, Macintosh.
- dispose de moyens comme indiqués en figure 3-4 lui donnant plus de flexibilité et efficacité dans des opérations de simulations et qui sont :



**Fig. 3-4 MATLAB et ses différentes boîtes à outils**

- a- Deux sortes de fenêtres servant à la commande et au graphisme et qui sont :**
- fenêtre de Commande** : dans cette fenêtre, l'utilisateur donne les instructions et MATLAB retourne les résultats ;
  - Fenêtres Graphiques** : MATLAB trace les graphiques dans ces fenêtres ;
- b- Deux sortes de fichiers :** **fichiers M** renfermant des programmes en langage MATLAB (écrits par l'utilisateur) ; et **fichiers P** qui sont en version pré-interprétée des fichiers M;
- c- Toolboxes** (« boîtes à outils») : ce sont des collections de fichiers M développés pour des domaines d'application spécifiques a savoir (signal processing, toolbox, system identification toolbox, control system toolbox, synthesis and analysis toolbox robust control toolbox, robust control toolbox, optimisation toolbox, neural network toolbox, spline toolbox, symbolic math toolbox, fuzzy logic toolbox, etc.) ;
- d- Simulink** : c'est l'extension graphique de MATLAB permettant de travailler avec des schémas en blocs, pour modéliser et simuler des systèmes ; on cite les éléments suivants :
- **Blocksets** : ce sont des collections de blocs SIMULINK développés pour des domaines d'application spécifiques (DSP BLOCKSET, POWER SYSTEM BLOCKSET, etc.)
  - **Fichiers MDL** : ce sont des fichiers représentant des modèles SIMULINK ;
  - **Fichier MEX** : modules exécutables créés à partir de sources en C ou FORTRAN ;
- e- GUI** : interface graphique utilisateur pour créer des applications basées sur MATLAB. En MATLAB, il existe deux modes de fonctionnement :

- **Mode interactif** : MATLAB exécute les instructions au fur et à mesure qu'elles sont données par l'utilisateur.
- **Mode exécutif** : MATLAB exécute ligne par ligne un programme en langage MATLAB écrit dans un fichier M (ou P), ou un fichier exécutable MEX.

### 3.5.2 Exemples :

Dans la fenêtre de commande de la figure 3-5 on écrit les instructions, et le programme l'exécute comme le montrent les exemples suivants :

- Création d'un scalaire

```
>> a = 1.3  
a =  
1.3000
```

- Le point virgule après l'expression supprime l'affichage

```
>> a = 1.3;
```

- Création d'un vecteur:

```
>> b = [1 -2 13 41]  
b =  
1 -2 13 41
```

- >> c = 1:5

```
c =  
1 2 3 4 5
```

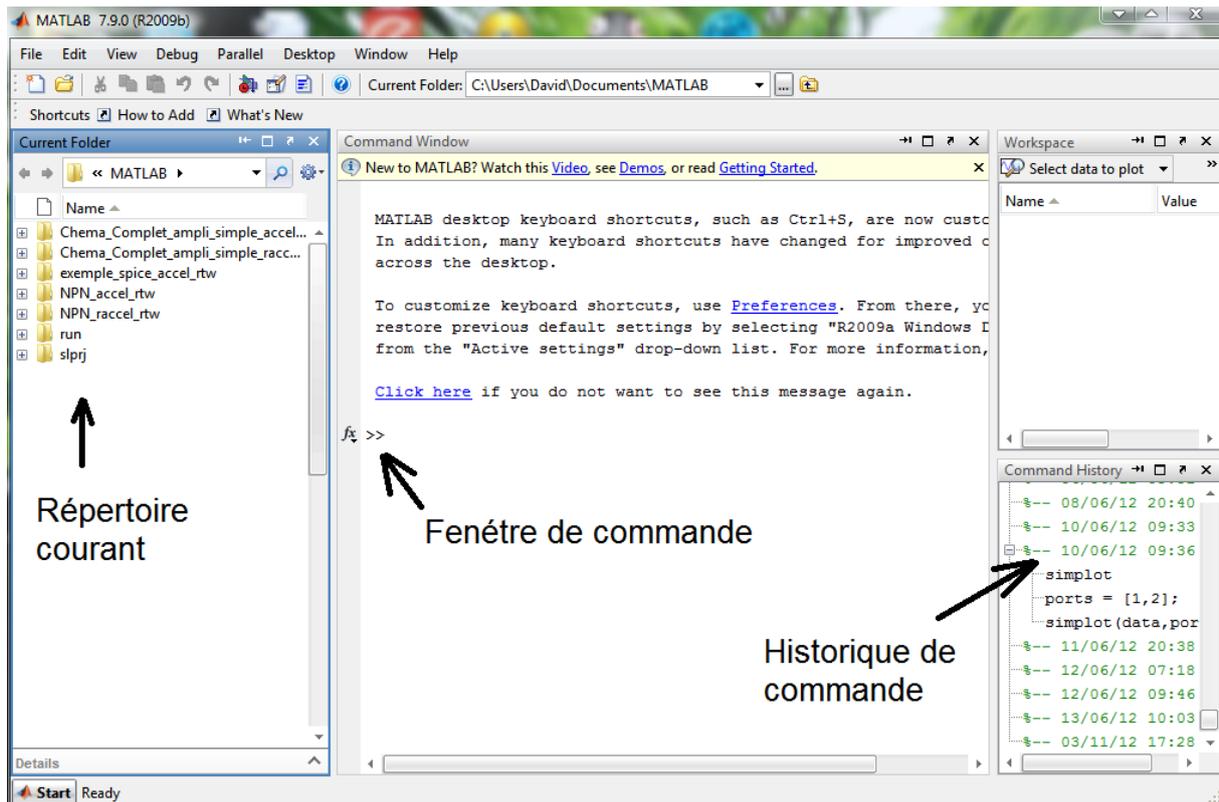


Figure 3-5 : interface du logiciel MATLAB

```
>> d = 4:-0.5:2.5
```

```
d =
```

```
4.0000    3.5000    3.0000    2.5000
```

### 3.5.3 Quelques fonctions mathématiques et les opérations arithmétiques.

En outre les instruction de commande qu'utilise le MATLAB, il dispose aussi d'un pléiade de fonctions mathématiques et opération arithmétiques que les programmeurs peuvent faire usage. A titre d'information, on exhibe les plus importants dans les tableaux 3.6 (a) et (b).

(a)

Operation	Symbol	Example
Addition, a+b	+	5+3
Subtraction, a-b	-	5.05-3.111
Multiplication, a*b	*	0.124*3.14
Left division, a\b	\	5\3
Right division, b/a	/	3/5(=5\3)
Exponentiation, a <sup>b</sup>	^	5^2

(b)

<b>abs</b> valeur absolue module (nb. complexe)	<b>angle</b> argument (nb. complexe)	<b>sqrt</b> racine carrée	<b>real</b> partie réelle	<b>imag</b> partie imaginaire
<b>conj</b> conjuguée (nb. complexe)	<b>round</b> arrondir	<b>fix</b> arrondir (vers zéro)	<b>floor</b> arrondir (vers $-\infty$ )	<b>ceil</b> arrondir (vers $+\infty$ )
<b>sign</b> signe	<b>rem</b> reste	<b>exp</b> exponentielle	<b>log</b> logarithme base $e$	<b>log10</b> logarithme base 10

Tableau 3.2 (a) et (b) : quelques fonctions Mathématiques

### 3.6 La partie SIMULINK

#### 3.6.1 Introduction :

Simulink est un outil de Matlab qui permet d'introduire un modèle à l'aide d'une interface d'usage visuel. Simulink s'intègre ainsi dans Matlab comme un nouvel environnement. Dans cet environnement, plusieurs boîtes à outils sont disponibles. Dans chaque boîte à outils il existe des éléments communément utilisés dans l'automatique déjà prêts à utiliser comme intégrateurs, additionneurs, gains, fonction de transfert, etc. . L'utilisateur n'a qu'à choisir les composants qui vont constituer son modèle puis il les interconnecte. SIMULINK offre aussi des possibilités de communication avec l'environnement Matlab, enregistrement des résultats ou des variables et finalement différentes possibilités d'affichage. Il représente l'extension graphique de MATLAB permettant d'indiquer les fonctions mathématiques et les systèmes dynamiques sous forme de schémas blocs.

#### 3.6.2 Démarrage de Simulink

Pour démarrer Simulink à partir de Matlab, un clic sur l'icône et une fenêtre apparaîtra comme le montre la figure 3-6.

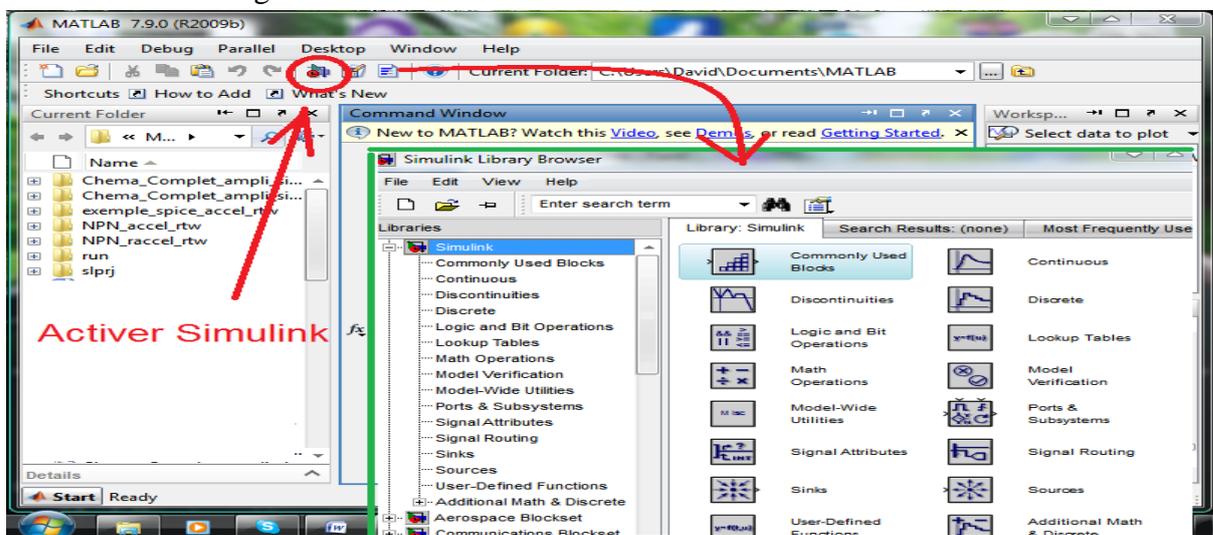


Figure 3-6 : fenêtre avec barre d'outil et de démarrage du simulink.

Il existe plusieurs bibliothèques et sous-bibliothèques pour les différentes opérations à exécuter et contenant les modèles nécessaires pour les tâches de conception et simulation. Nous présentons certaines de ces opérations et modèles dans le tableau 3-3:

**Le tableau 3-3 : quelques bibliothèques de Simulink**

Collection	Contenu	Blocs plus utilisés
Sources	sources de signaux	générateurs de signaux, horloges, chargement de données d'un fichier ou du <i>workspace</i>
Sinks	affichage, stockage	afficheurs de signaux, stockage de données dans un fichier ou dans le <i>workspace</i>
Continuous	blocs continus	intégrateur, fonction de transfert, représentation d'état, retards
Discrete	blocs discrets	intégrateur, fonction de transfert, représentation d'état, bloqueurs, filtres
Math opération	opérateurs mathématiques	fonctions trigonométriques, signe, valeur absolue, gains, somme, produit
Nonlinear	blocs non-linéaires	frottements, saturations, commutateurs
SimElectronics	Electronique	Circuit intégré, composant semiconducteur
Logic and bit operation	Logique	Les Portes logiques, comparateur binaires

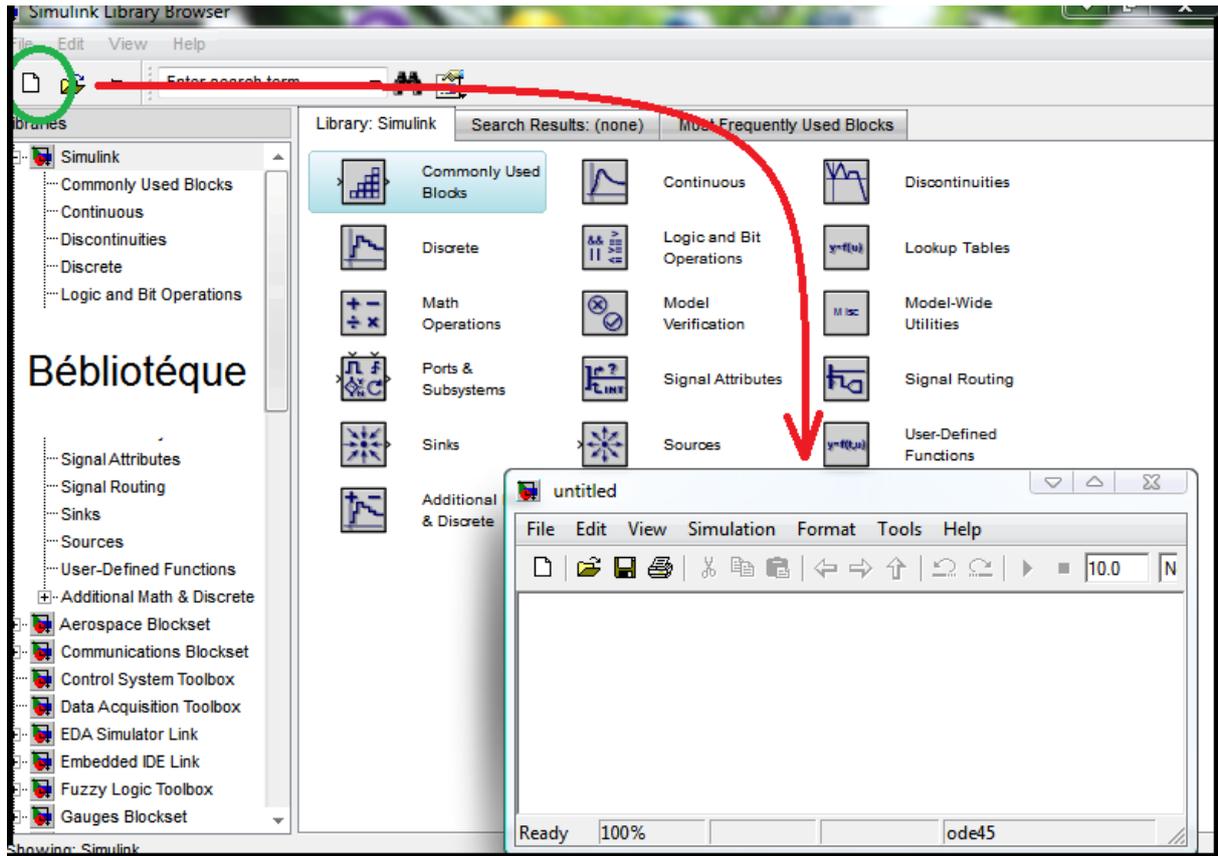


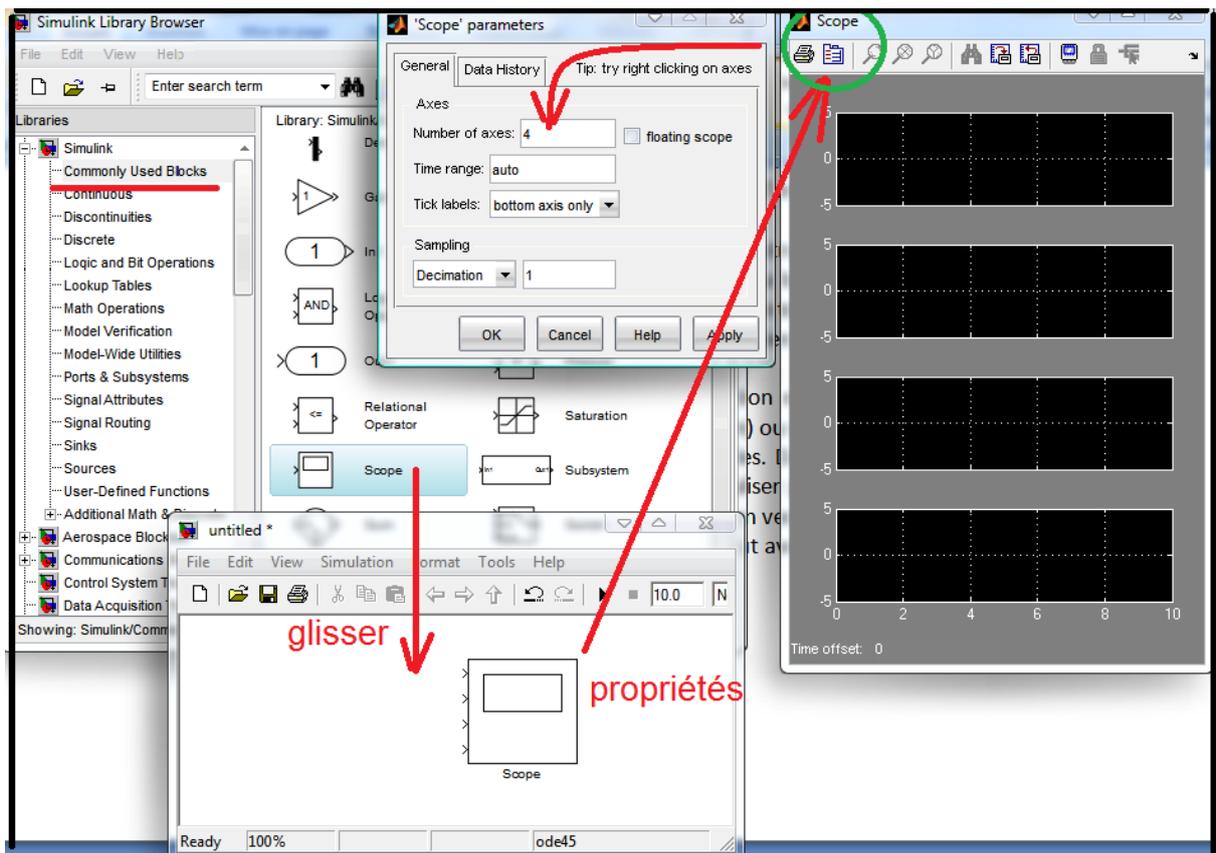
Figure 3-7 : ouverture de fenêtre de travail de simulation.

La construction des modèles se fait, en général, en recopiant des blocs à partir des bibliothèques. Cliquez sur l'icône "**Linear**" –par exemple-, nous permet l'ouverture d'une nouvelle fenêtre de travail présentant les blocs disponibles dans cette boîte à outils : gains, fonctions de transfert, intégrateurs, etc.

Pour ajouter un élément à votre modèle, on clique dessus puis on le glisse vers la fenêtre du modèle. On répète l'opération avec autant d'éléments nécessaires. Une fois l'élément placé dans la fenêtre modèle, on peut modifier ses paramètres en cliquant deux fois sur lui : une boîte dialogue apparaîtra vous permettant la définition de votre élément. On remarque que certains éléments (l'additionneur, par exemple) permettent de définir le nombre d'entrées au besoin.

Dans la boîte "Linear" on trouve les outils de base pour construire notre modèle et la plupart des modèles linéaires indépendamment de leur complexité. La boîte "Sources" nous offre des générateurs des signaux, rampes, échelons et d'autres entrées d'intérêt, ainsi qu'une passerelle pour obtenir des données à partir d'un fichier de données (.mat) ou bien directement à partir de l'environnement Matlab.

Finalement, la boîte “Sinks” contient des éléments de visualisation et d’affichage (scope, display) (voir figure 3-8), ainsi que la passerelle pour envoyer des résultats vers un fichier (.m) ou vers l’espace Matlab. Utilisez l’élément “scope” pour visualiser l’évolution temporelle des variables. Dans les versions précédentes chaque boîte “scope” ne compte qu’une seule entrée. Afin de visualiser plusieurs signaux simultanément dans le même écran, il suffit de les combiner dans un vecteur des signaux à l’aide d’un “multiplexeur”. Mais la version Matlab 7.9.0 qu’on a utilisée peut avoir plusieurs entrées à la fois seulement en modifiant ces propriétés.



**Fig.3-8 : Emploi de l’option scope**

### 3.6.3 Création de sous systèmes :

Pour grouper des blocs en un sous-système afin d’éviter la complexité du système, on choisit les blocs qu’on veut contenir dans le sous système, puis on sélectionne Create sub-system dans le menu Edit.

### 3.6.4 Réglage du temps de simulation :

Pour les montages simples on peut se contenter des paramètres proposés par défaut (figure 3-9), en choisissant juste la durée désirée de la simulation. Cependant, ce menu a une influence énorme sur la qualité de la simulation pour des systèmes à peine compliqués et il faut apprendre

à bien s'en servir. Cette opération se fait par ouverture des paramètres de configuration en sélectionnant « Configuration Paramètres » dans le menu Simulation.

Par la suite, on règle les temps de simulation en utilisant les champs Start time et Stop time dans la section "Simulation Time" de la boîte de dialogue "Configuration Parameters".

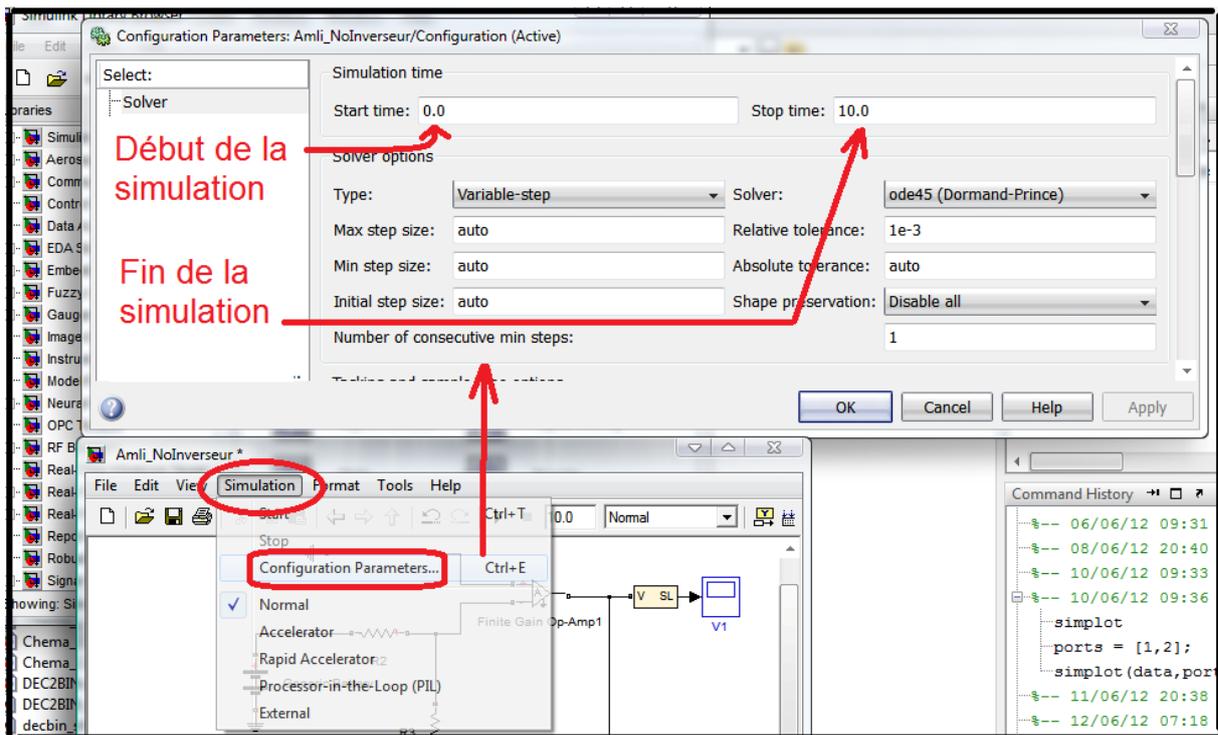


Fig. 3-9 : boîte de dialogue de réglage de temps de simulation.

### 3.6.5 Exemple d'application :

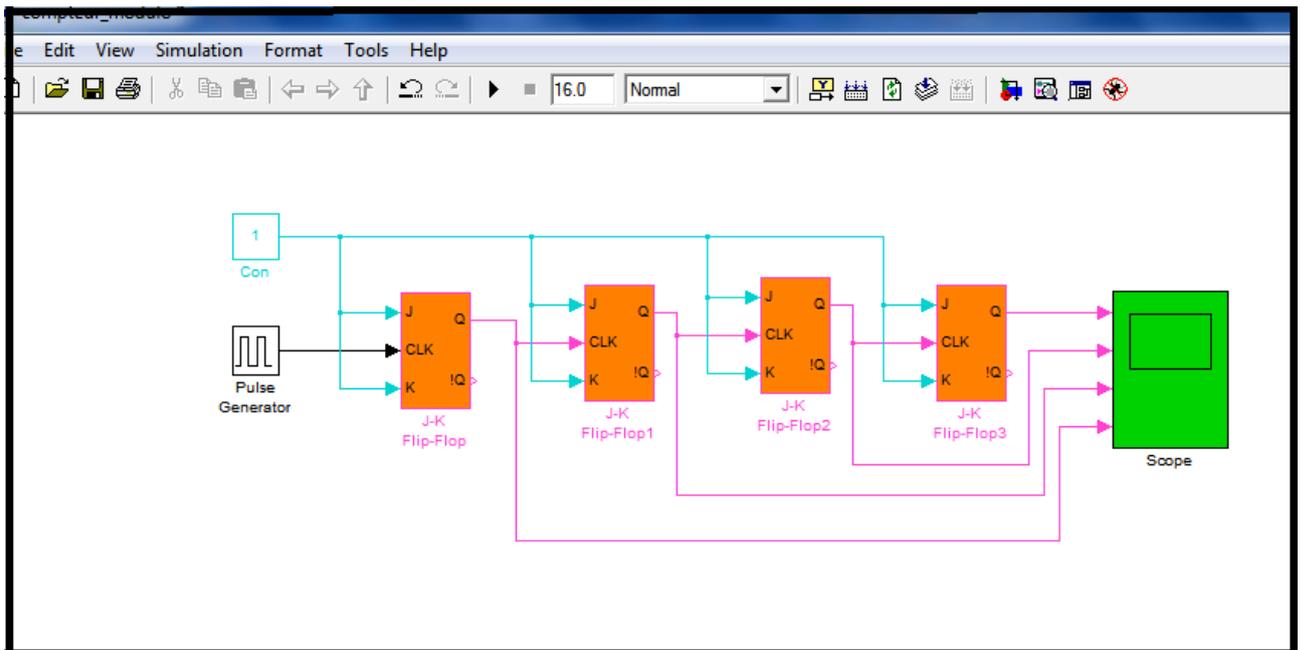
A titre d'exemple de simulation, on prend le cas d'un compteur simple Modulo 16, alors on opte pour un nouveau model en cliquant sur l'icône New ---- > Model ----- > et une fenêtre vide apparaîtra ;

Pour ce compteur Modulo 16 on aura besoin de quatre Bascule JK, un générateur d'impulsion pour l'horloge, et une Constante pour attaquer les bascules JK par des « 1 » .

Les bibliothèques utiles sont :

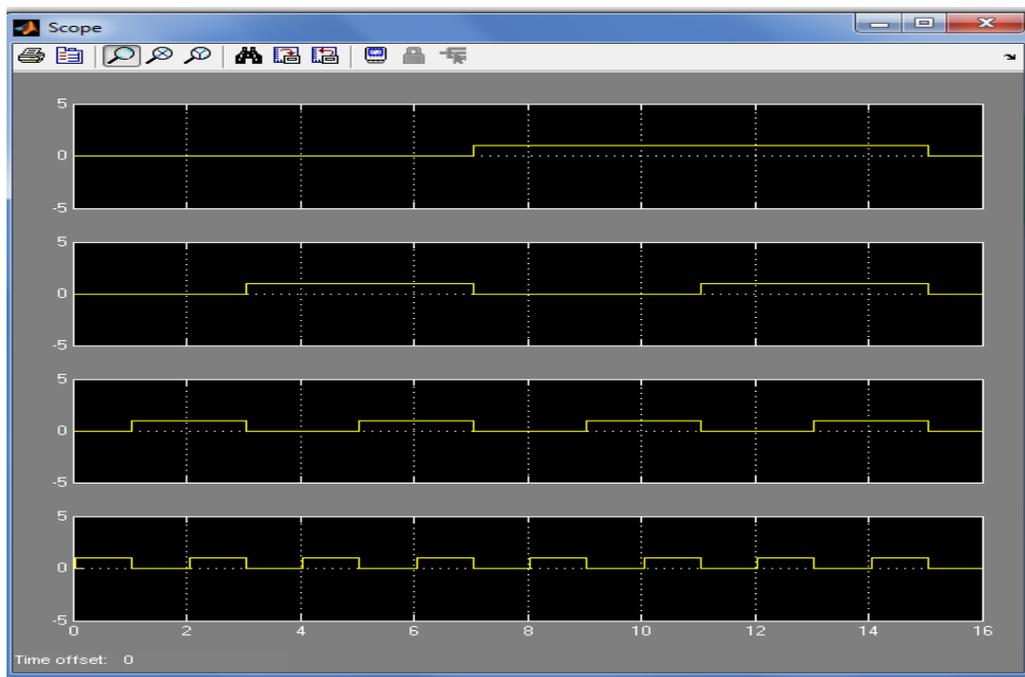
- **Simulink Extra ---- > Flip Flop** : on trouve les Bascules JK et le Clock
- **Simulink ----> Sources** : pour copiée la Constante
- **Simulink ----> Sinks** : pour avoir le Scope

A travers la figure 3-10, on essaie de donner le schéma complet d'un compteur:



**Fig. 3-10 : exemple d’application de simulation d’un compteur.**

Le temps de simulation est de 16 secondes pour voir tout le cycle, après l’exécution le Scope nous montre les résultats selon la figure 3-11



**Fig. 3-11. Chronogramme d’un compteur par simulation**

### **3.7 Conclusion :**

Au cours de ce chapitre, on a donné un aperçu général sur l'outil de simulation et son importance dans la conception. Cet outil se distingue d'un type d'analyse à un autre. Par exemple, l'analyse analogique (électrique) et l'analyse logique d'un circuit combinatoire ne se font pas dans un même simulateur. Chacune d'elles se déroule dans un environnement software approprié : le premier type d'analyse s'effectue par un simulateur électrique par contre le second se fait à l'aide du simulateur logique. L'option MATLAB / Simulink est choisie pour résoudre ce problème de conflit entre l'analogique et la logique binaire des circuits. Son éminence est mise en valeur dans le chapitre qui suit.

# **Chapitre 04**

**la structure BIST et  
son application sur un  
circuit analogique**

## 4.1 INTRODUCTION

Le travail présenté dans ce chapitre consiste à mettre en œuvre la technique BIST de nature numérique par voie de simulation (emploi de Simulink) sur un circuit analogique pourvoyant une fonction électronique élémentaire choisie comme circuit à tester, CUT. Ce dernier est un amplificateur inverseur réalisé selon une configuration d'un prototype de la compagnie *Ferranti Interdesign* en utilisant des transistors bipolaires Monochip selon une technologie planaire et par épitaxie. En conséquence, une simulation précise de circuit exige l'emploi de modèles de transistors propres à cette technologie d'où leur implémentation prépondérante dans ce simulateur .

Le processus de simulation suivi est un peu complexe dans son exploitation car 3 sortes de simulation en font partie. Il s'agit de la simulation numérique de la structure électronique BIST, la simulation électrique de la structure du circuit analogique à tester et la simulation de conversion analogique-numérique et inversement en utilisant les fonctions mathématiques qu'on a implémentées dans ce même simulateur. Cette dernière opération fut introduite comme interface de communication entre la structure de test BIST et celle du CUT.

La validation de ce processus de test passera sans doute d'abord par une réussite de cette technique BIST à fournir des signatures émanant du circuit sous test en l'absence et présence de fautes (défauts électriques). Ces fautes sont généralement des courts-circuits ou circuits- ouverts très fréquents en analogique. Dans cette partie de travail, l'attention est dédiée à la vérification de l'efficacité de ce genre de test pour 2 fautes distinctes pouvant atteindre un des transistors du circuit à tester.

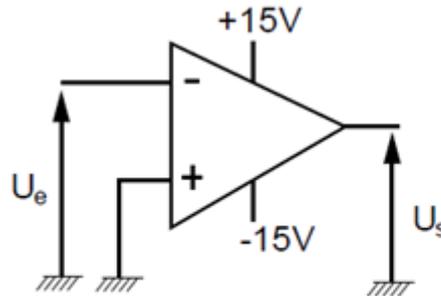
## 4.2 Description de la structure sous test.

L'objectif fixé dans notre étude est de rendre possible le test de circuit analogique intégré par un testeur numérique. Le circuit à soumettre à ce processus de vérification n'est autre que le fameux amplificateur de type 741 monté en amplificateur inverseur .Le choix porté sur ce composant est dû à son ample champs d'utilisation en électronique et une grande diversité de fonctions qu'il offre.

### 4.2.1 L'amplificateur opérationnel $\mu A$ 741

Ce composant électronique analogique constitue une brique de base dans un circuit électronique. Il peut réaliser diverses opérations sur un signal électrique: amplification, comparaison, soustractions, additions, déphasage (décalages dans le temps), filtrages, etc... . Ses

différentes fonctions à réaliser sont définies par des résistances, condensateurs, diodes, etc., auxquels ce composant est branché ainsi que de la topologie du circuit externe.



**Figure 4.1 Représentation schématique de l'Ampli-Op**

Selon sa configuration donnée en figure 4-1, ce composant est muni de 3 bornes de raccordements fonctionnelles: deux entrées + et - et une sortie. Il possède 2 bornes d'alimentation dont la tension de service est en général symétrique  $\pm 5V$ ,  $\pm 10V$ ,  $\pm 12V$ ,  $\pm 15V$ ... Dans certains cas (assurant une fonction de commutation), l'alimentation peut aussi être dissymétrique, par exemple (0V-5V.).

Ces caractéristiques électriques les plus usuelles sont :

- Résistance d'entrée théoriquement infiniment grande ( $R_e \rightarrow \infty$ ). Pratiquement sa valeur est finie est dépendante de la technologie des transistors employée (10 à 20 M $\Omega$  pour la technologie bipolaire jusqu'à 200 M $\Omega$  en technologie CMOS).
- Résistance de sortie très faible pouvant avoir quelques dizaines d' Ohms (théoriquement  $R_s=0 \Omega$ ).
- Le gain en tension  $G_v$ :
  - Théoriquement le gain en boucle ouverte est infiniment grand ( $G_v \rightarrow \infty$ ),
  - Pratiquement, ce gain est de  $6 \cdot 10^5$  à  $10^7$  environ.

Ce paramètre est donc un facteur d'amplification par excellence pour des tensions d'entrée  $U_e$  jugées trop faibles pour le composant. Par contre pour des tensions plus ou moins importantes, l'amplification avec ce gain n'est plus valable.

A titre d'exemple, si on dispose d'un gain de 600000, alors si  $U_e=1\text{mV}$ , la tension de sortie  $U_s$  devrait être de 600 V, ce qui est impossible vu que la tension d'alimentation ne dépasse guère  $\pm 15\text{ V}$ . On dit que l'amplificateur est saturé, son fonctionnement est non-linéaire.

Cette situation pourra être surmontée par ajout de composants externes (des résistances) qui rendent le gain  $G_v$  contrôlable selon le niveau de tension appliquée. Par contre, en fonctionnement linéaire, si  $G_v$  est très grand (vers l'infini), alors  $U_e$  tend vers 0.

#### 4.2.2 Amplificateur en mode inverseur.

En utilisant les caractéristiques propres à l'amplificateur opérationnel définies précédemment, on pourra énoncer les évidences suivantes :

- Résistance d'entrée infinie signifie que le courant qui traverse  $R_1$  et  $R_2$  de l'amplificateur inverseur de la figure 4-2 est le même. Le gain de l'ampli. Op en fonctionnement linéaire est infini, entraînant une différence de potentiel  $U_d$  quasi -nulle.
- Selon l'évidence précédente, on pourra obtenir le gain que l'on désire par le choix d'un rapport des 2 résistances précitées avec plus de commodité car elles sont très bon marché, de bonne qualité et offrent des valeurs extrêmement variées.

$$\left. \begin{array}{l} i = \frac{U_e - u}{R_1} = \frac{U_e}{R_1} \\ i = -\frac{U_s - u}{R_2} = -\frac{U_s}{R_2} \end{array} \right\} \Rightarrow \frac{U_s}{U_e} = G = -\frac{R_2}{R_1}$$

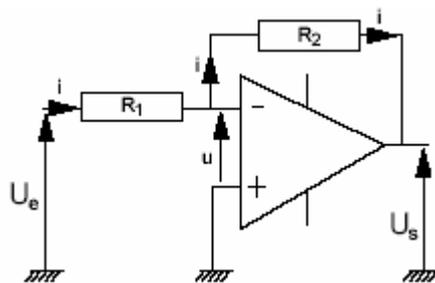


Figure 4.2 : la fonction de transfert de l'Ampli-Op inverseur

### 4.2.3 Structures internes de l'amplificateur $\mu A$ 741

Le test appliqué au circuit analogique choisi vise à détecter les fautes électriques pouvant avoir lieu au niveau des transistors de l'amplificateur opérationnel  $\mu A$  741. Celui-ci peut avoir une structure électronique interne qui diffère d'une compagnie/ maison de fabrication à une autre de par la technologie bipolaire, CMOS ou mixte et du nombre de composants employés. Il est alors impératif de connaître ses deux variantes (technologie et complexité) de tout circuit à tester.

Le type d'ampli-Op utilisé dans notre circuit est propre à la compagnie FERRANTI – INTERDESIGN et d'ont le schéma électrique de la figure 4-3 nous montre sa structure interne.

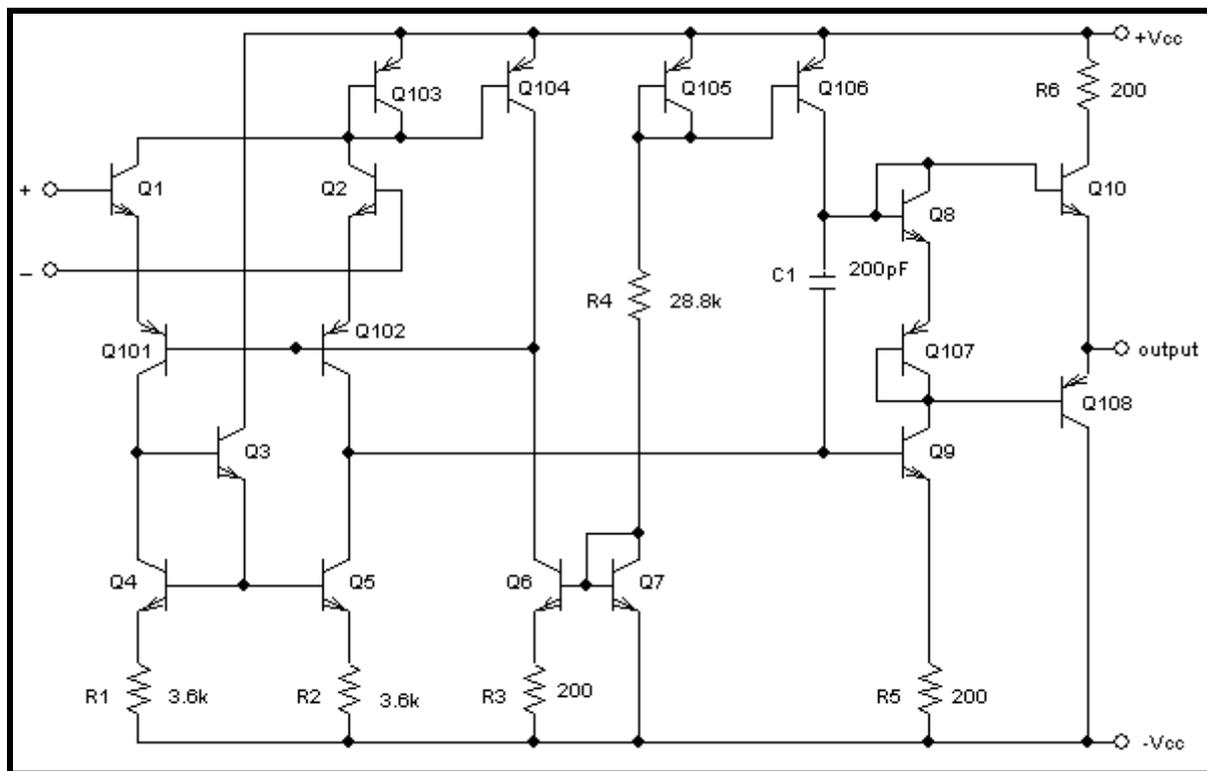


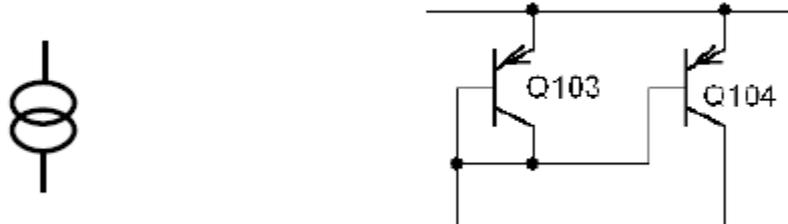
Figure 4.3 : structure interne d'un Amplificateur  $\mu A$ 741.

A partir de ce schéma électrique, on peut distinguer et identifier trois étages associés en cascade :

#### ❖ L'étage d'entrée :

Il est constitué par l'ensemble des transistors Q1, Q2, Q101, Q102, avec un fort gain par la suite de leurs associations, le gain en sortie de l'ensemble est important grâce aux charges

actives constituées par les transistors Q4 et Q5. Comme le montre la figure 4-4, le circuit de polarisation de l'étage d'entrée est réalisé par le transistor monté en diode, Q103, et la source de courant Q104. Le circuit courant miroir fournit un courant maintenu constant (autour de  $180\mu\text{A}$ ), par un réseau polarisé construit avec les transistors Q6, et Q7.



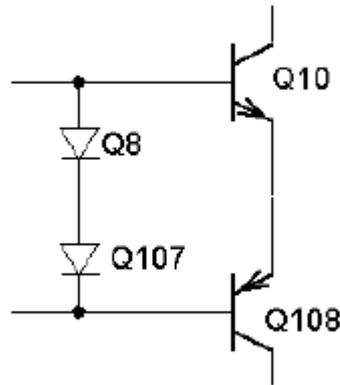
**Figure 4.4 : Polarisation de l'étage d'entrée (Q103 monté en Diode)**

❖ **L'étage intermédiaire (fondamental) :**

L'étage d'entrée est suivi d'un circuit fondamental (émetteur commun) à fort gain en tension, dont la sortie n'est plus différentielle. Il est identifié par la liaison du collecteur du Q102, et la base du Q9. Son courant de polarisation est assuré par le générateur du courant des deux transistors Q105, et Q106, générant ainsi le courant miroir ( $650\mu\text{A}$ ).

❖ **L'étage de sortie :**

Cet étage illustré en figure 4-5, est constitué de deux transistors montés en push-pull Q10 et Q108. Ces deux composants de polarisation inverse sont complémentaires et assurent une amplification de tensions aussi bien négatives que positives tout en fournissant un rendement en puissance meilleur et un taux de distorsions faible.



**Figure 4.5 : Polarisation de push-pull**

Nous remarquons que les deux transistors montés en diodes (Q107 et Q8) sont employés pour polariser les transistors précédemment cités par une tension d'environ de 1.4 volts tout en la maintenant constante et stable à l'entrée de cet étage. En plus des éléments essentiels décrits ci-dessus, on trouve une capacité  $C$  qui régit le comportement de l'amplificateur opérationnel en fonction de la fréquence. Elle permet le couplage entre l'étage d'entrée et celui de sortie tout en assurant la suppression des composantes continues en régime alternatif.

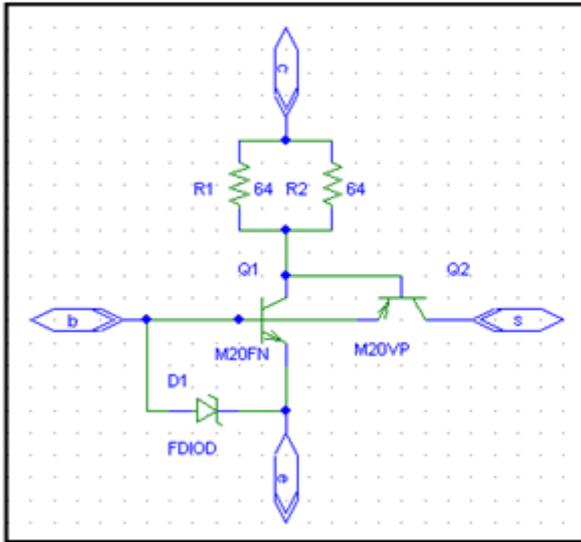
#### 4.2.4 Structure des transistors utilisés [74] [49].

Le schéma électrique de l'amplificateur opérationnel disposé en figure 4-3 est propre à la maison originale FERRANTI, en outre, la réalisation fut possible grâce à la disposition des transistors (voir Fig.4-7) intégrés dans une même puce (Mono-chip) et offrant les mêmes caractéristiques électriques que celles du 741 intégré.

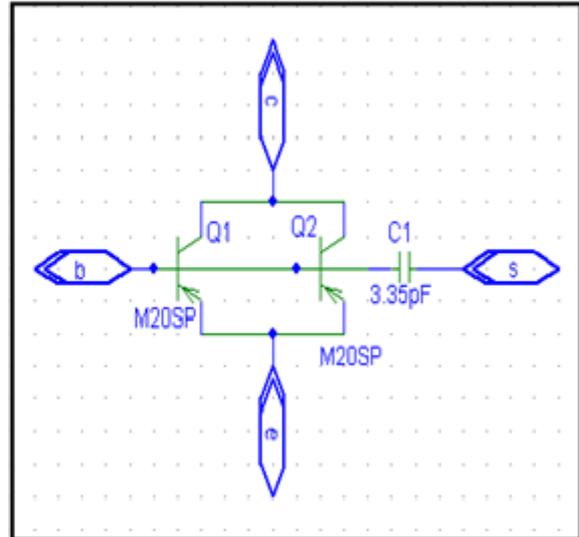
Il y'a deux types de polarisation de transistors employés dans cet amplificateur particulier :

- les transistors de Q1 à Q10. sont de type NPN
- les transistors de Q101 à Q108 sont de type PNP.

La structure interne équivalente à chacun de ces transistors spécifiques à cette compagnie est complexe et émane de la technologie de fabrication par épitaxie servant à leur réalisation : il s'agit de structures planaires latérale et verticale conduisant aux circuits équivalents des transistors comme indiqués dans les figures suivantes :



(a) Transistor NPN



(b) Transistor PNP

**Figure 4.6 structure interne du transistor type FERRANTI**

Ces circuits équivalents de transistor sont au fait un assemblage de composants de transistors élémentaires et diodes, résistances dont les caractéristiques sont citées dans les 2 fiches techniques suivantes :

```

**** paramètres de la Diode *****
FDIOD
IS 45.000000E-21
N 1.75
ISR 100.000000E-12
BV 6.6
IBV 10.000000E-06
RS 75
TT 5.000000E-09
CJO 1.000000E-12
VJ .75
M .3333
    
```

**Fiche technique 1: Paramètres électriques du modèle de diode en Technologie planaire.**

	M2OSP	M20VP	M20FN
	PNP	PNP	NPN
IS	200.000000E-18	1.300000E-15	450.000000E-18
BF	90	20	150
NF	1	1	1
VAF	50	150	150
IKF	500.000000E-06		6.000000E-03
ISE	5.000000E-15		7.000000E-15
NE			1.75
BR	1	.15	.5
NR	1	1	1
VAR	100	100	150
IKR	100.000000E-06		1.000000E-03
ISC	10.000000E-15		1.000000E-15
NC	1.5		1.3
RB	50	760	135
RE	100	7	2.42
RC	100	15	36
CJE	120.000000E-15	2.000000E-12	546.000000E-15
CJC	400.000000E-15	2.700000E-12	1.520000E-12
TF	10.000000E-09	10.000000E-09	338.000000E-12
XTF	10	10	10
VTF	10	10	10
ITF	1	1	1
TR	10.000000E-09	10.000000E-09	33.800000E-09
XTB			4.400000E-03
CN	2.2	2.2	2.42
D	.52	.52	.87

**Fiche technique 2 : les paramètres électriques du modèle de transistors bipolaires en technologie planaire**

### 4.3 Création du circuit électronique.

Le circuit analogique à tester est un amplificateur inverseur et à posteriori aura une configuration finale après simulation par Simulink comme la montre la figure4-7 :

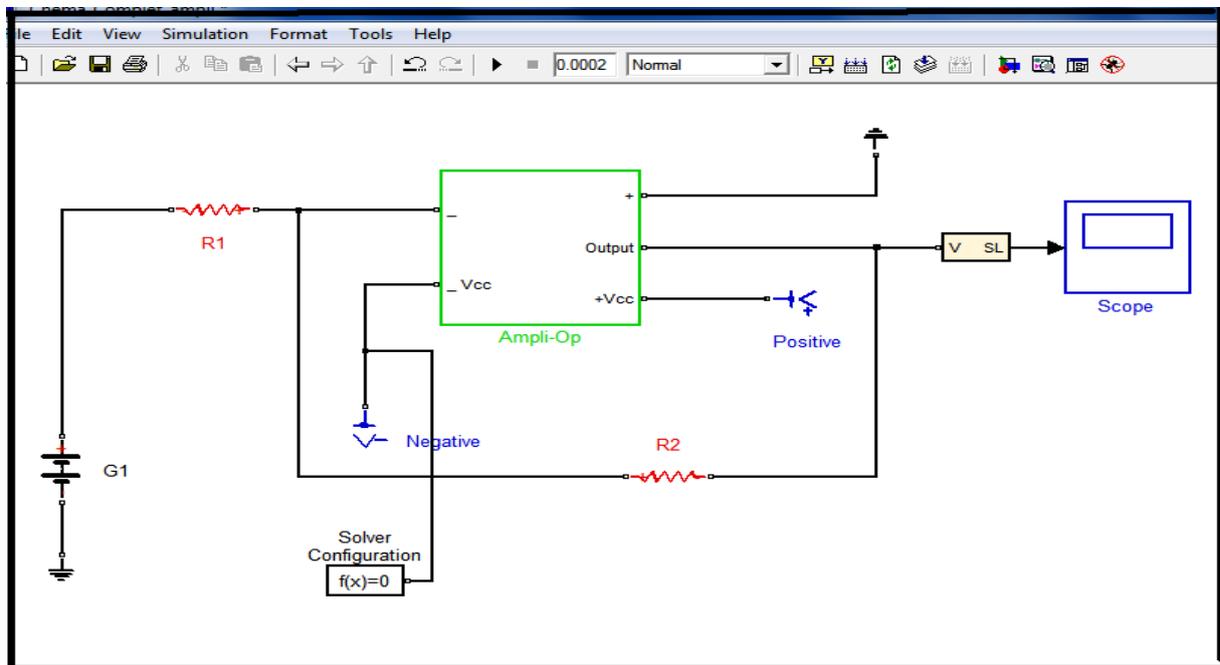


Figure 4.7 : création de  $\mu A$  741 sous Simulink

Le montage de l'Ampli-OP en inverseur exécuté sous Simulink est fait par des composants électroniques compatibles avec PSPICE empruntés de la bibliothèque de composants dont l'accès s'effectue selon l'enchaînement d'actions suivantes:

Simscape --- > SimElectronics --- > Semiconductor Device --- > SPICE Compatible Semiconductor

A travers la figure 4-8, où le schéma électrique de  $\mu A$  741 est présenté sous Simulink, on fait remarquer deux composants nouveaux :

- **Solver Configuration** est indispensable dans les diagrammes qui utilise-les composants Simscape .
- **Une interface (V  $\rightarrow$  SL)** qui sert à transformer le signal physique généré par les composants SPICE en un signal compatible avec le Simulink pour pouvoir le connecter avec un Scope ou autre bloc Simulink , de même pour l'opération inverse, (SL  $\rightarrow$  V) qui est aussi disponible .

La structure interne de l'amplificateur a pour configuration électrique sous Simulink présentée en figure 4-8. Il est à noter que les transistors bipolaires sont assimilés à des blocs (boîtes noires) ou seuls les points d'accès conservent leur appellation habituelle (émetteur, base,

collecteur). L'exception faite ici est la présence d'un 4<sup>ème</sup> point d'accès qui permet la liaison du substrat à la masse. Quant à la fonction du bloc, il assure la tâche d'un transistor équivalent décrit par le modèle propre à la technologie le réalisant.

### 4.3.1 La structure interne de $\mu A741$ sous Simulink

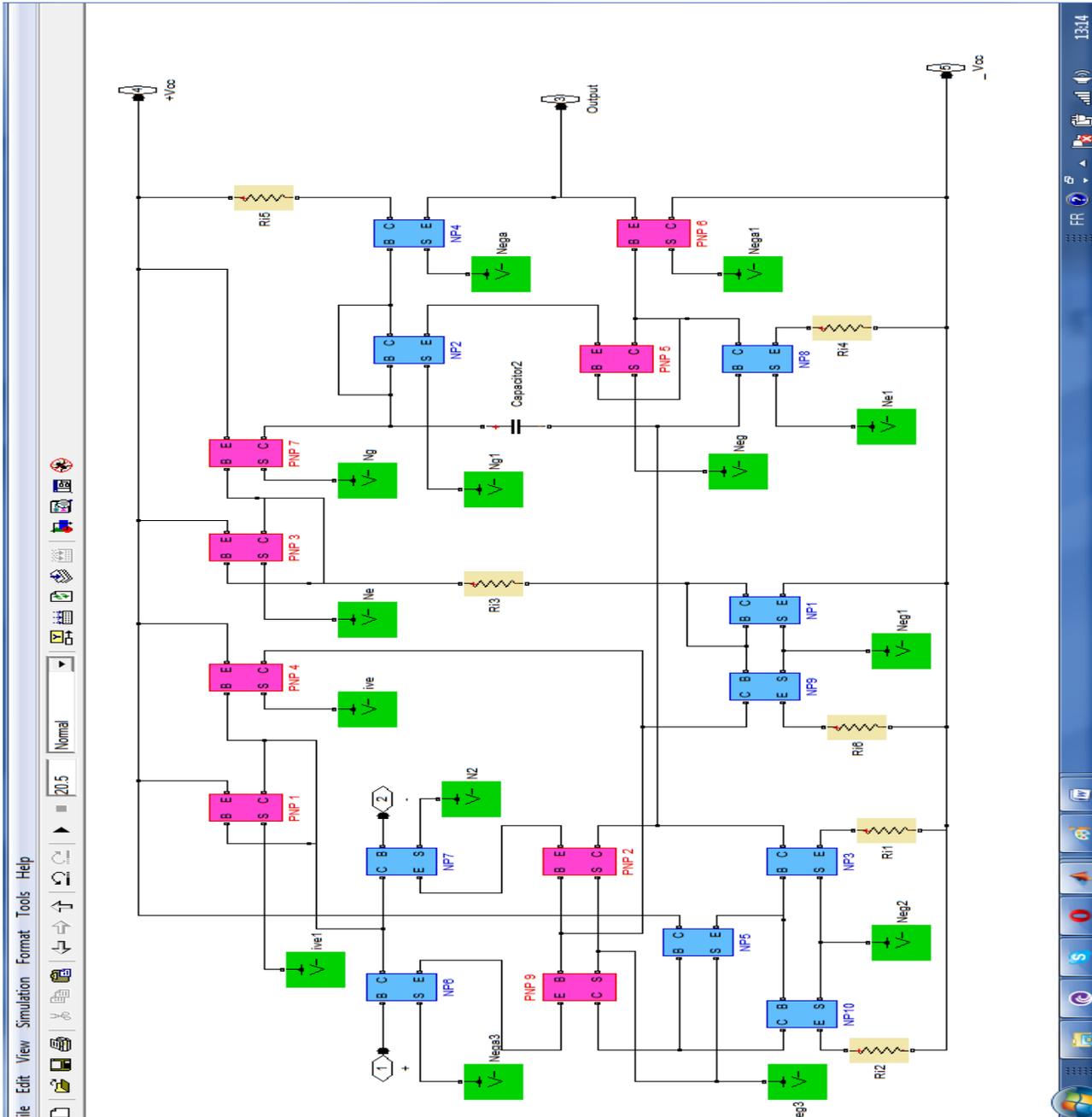
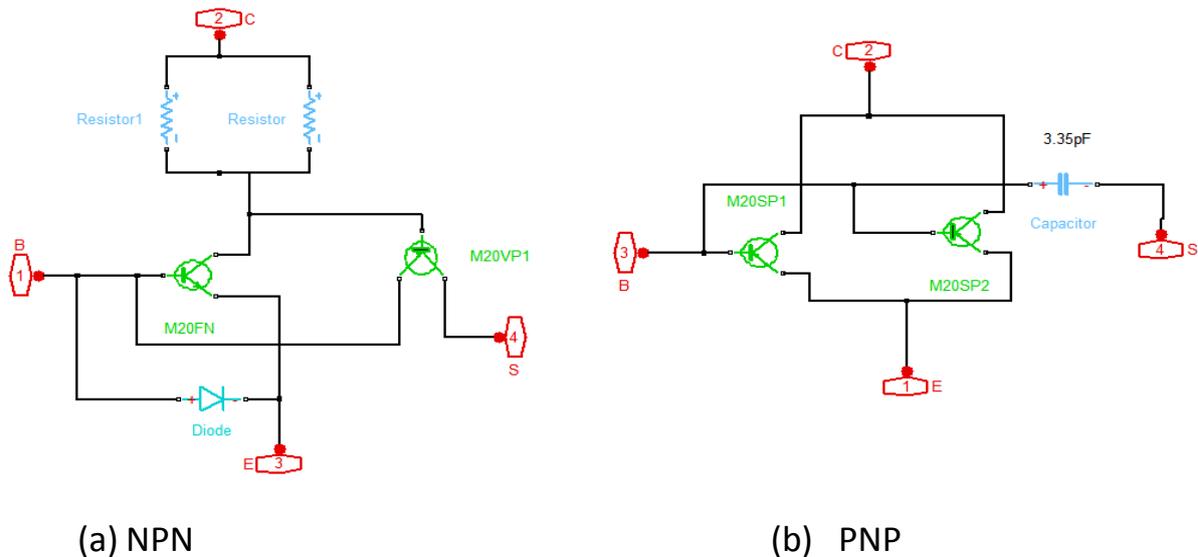


Figure 4.8 : structure interne de  $\mu A741$  selon FERRANTI

Les schémas équivalents de ces transistors sont détaillés en figures 4.9 (a) et (b) :



Figures 4.9 (a) et (b) : schéma équivalent de transistors NPN et PNP sous Simulink

### 4.3.2 Insertion de caractéristiques FERRANTI.

Pour que le circuit sous test reproduise les caractéristiques citées dans le cahier de charge il est indispensable d'injecter au préalable les paramètres électriques du transistor modèle propre à FERRANTI qui renferment dans leurs structures les transistors M20FN, M20VP, M20SP et la diode "FDiode" [49] [74]. Cette procédure s'effectue au niveau du simulateur tout simplement par un double clic sur le transistor choisi de la bibliothèque et qu'on a nommé "M20FN" par exemple puis sur propriétés. En conséquence, une fenêtre indiquée dans la figure 4-10 apparaîtra permettant ainsi de modifier ou d'insérer les valeurs des différents paramètres électriques.

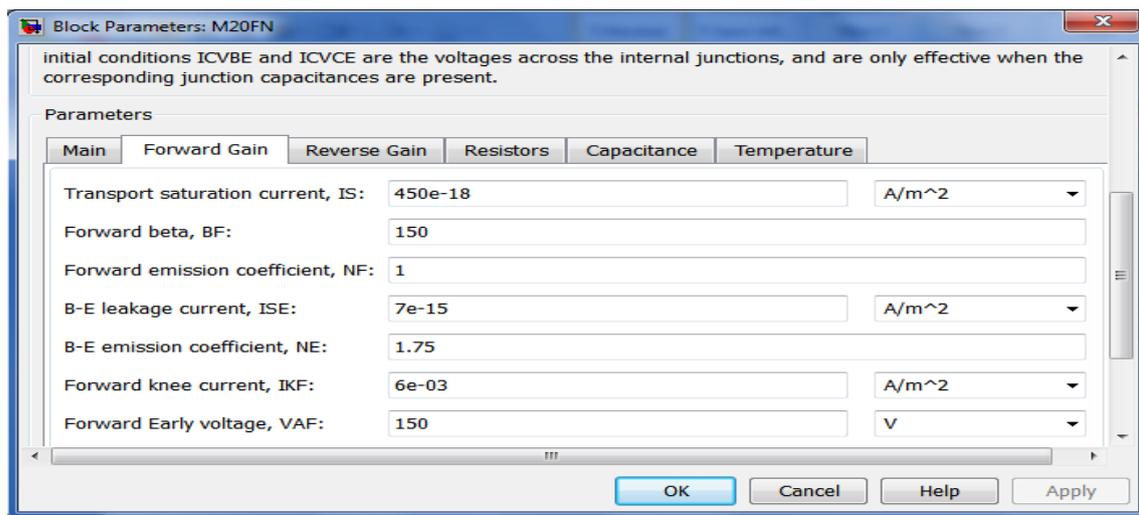
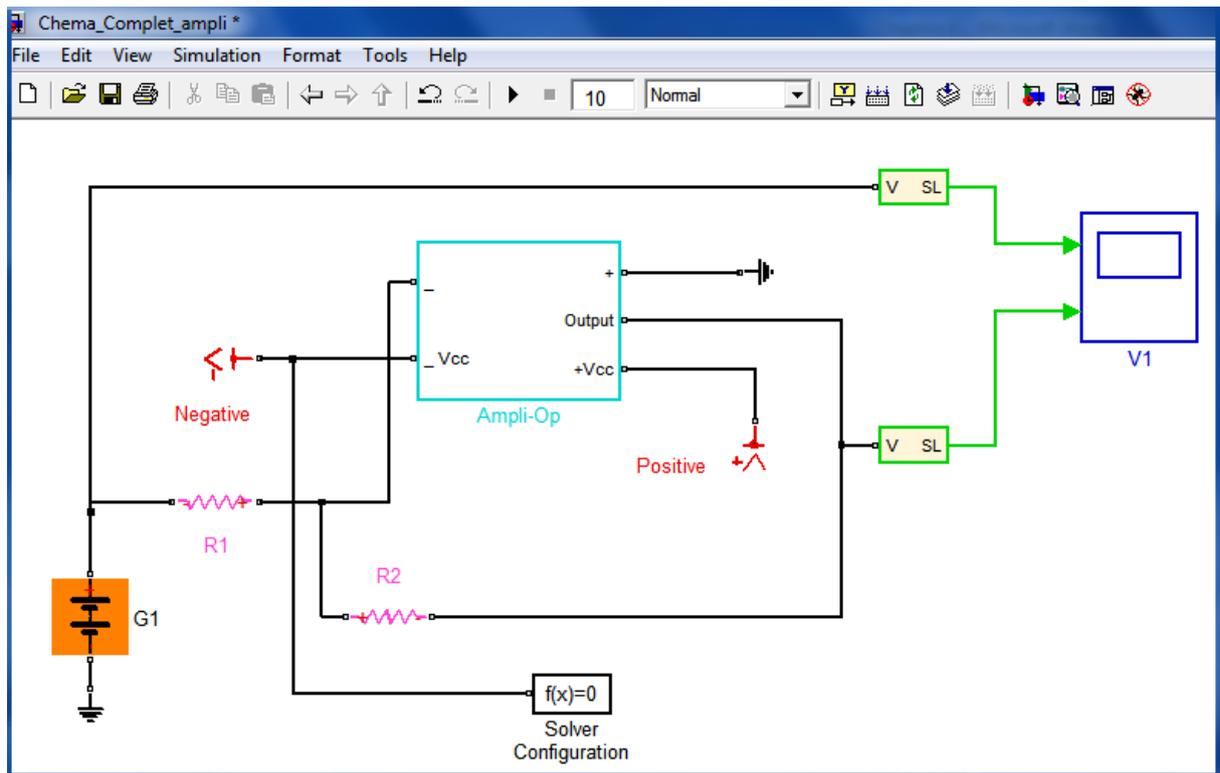


Figure 4.10 : paramètres de transistors

#### 4.4 Simulation de $\mu A741$ sous Simulink.

Il s'agit ici de l'étape finale de simulation du circuit sous test. Elle prend lieu une fois l'amplificateur construit et mis sous forme de bloc accessible par ces terminaux principaux (les entrées et sorties ainsi que les points d'alimentation). Une telle présentation fait de ce composant un élément multifonction.



**Figure 4.11 : Simulation et visualisation a l'aide d'un Scope**

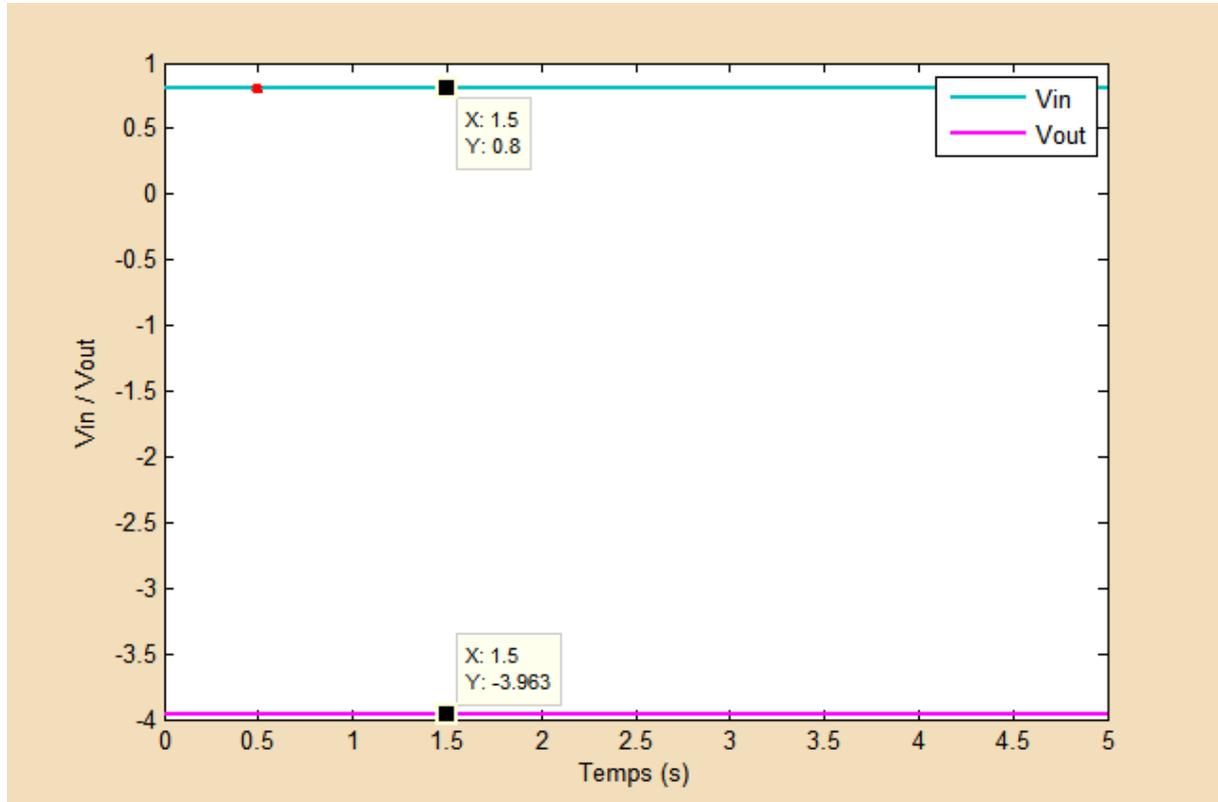
Selon le cahier de charge établi au circuit sous test d'une façon délibérée, les valeurs de composants associés sont:

- $R1 = 1K\Omega$  ;  $R2 = 5 K\Omega$  ; alors le gain  $G = - 5$ .
- $G1$  est un générateur de tension continue = 0.8 Volt (choisie à titre de démonstration).
- $-V_{cc} = -5V$
- $+V_{cc} = 5V$

Comme remarque très importante, le logiciel Simule notre montage après ajustement des paramètres de configuration suivants:

- Type : **variable-step** , Solver : **ode23t**

- Un Double clic sur le bloc **Solver Configuration** [  $f(x)=0$  ] et puis on exécute par un simple clic sur **Start Simulation** (Start Simulation from steady state) et le Scope affiche les résultats, sur la figure suivant :



**Figure 4.12 : le signal d'entrée et celui de sortie de l'Ampli-Op**

Le scope nous montre que l'Ampli-Op fonctionne correctement, par exemple pour une tension  $V_e = +0.8V$ , le circuit nous livre bien une tension de sortie  $V_s$  de  $-3,96$  Volts qui est bien le résultat de :

$$V_s = V_e * G = 0.8 * (-5).$$

L'affichage clair des résultats par le scope se déroule selon les étapes qui suivent [70]:

- ✚ Double clic sur le **scope** ---- > **paramètres** ----- > **Data history**.
- ✚ Cocher **save data to workspace**.
- ✚ Changer **variable name** (sc) par exemple.
- ✚ Changer le **format** vers **array**.
- ✚ Sauvegarder votre model dans un nouveau dossier sous un nom '**projet1**'.

Sur la page principale de MATLAB qu'on atteint après cette série d'actions: **file** -- > **New** -- > **M-File**, on peut prescrire :

&lt;&lt;

```

sim ('projet1')
figure (1)
plot (sc (:,1),sc(:,2),'k',sc (:,1),sc(:,3),'m')

```

&gt;&gt;

Pour une seule courbe, la commande plot sera :

```

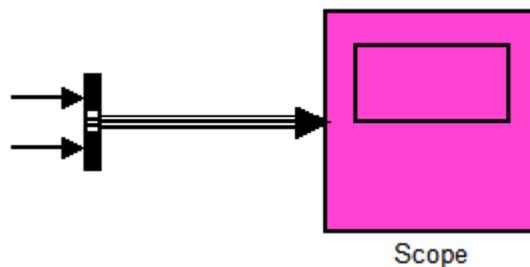
<<
plot (sc (:,1),sc(:,2),'k')

```

&gt;&gt;

On sauvegarde la page M-File dans le même nouveau dossier précédent. et le résultat s'affiche par Clic sur la touche "Run".

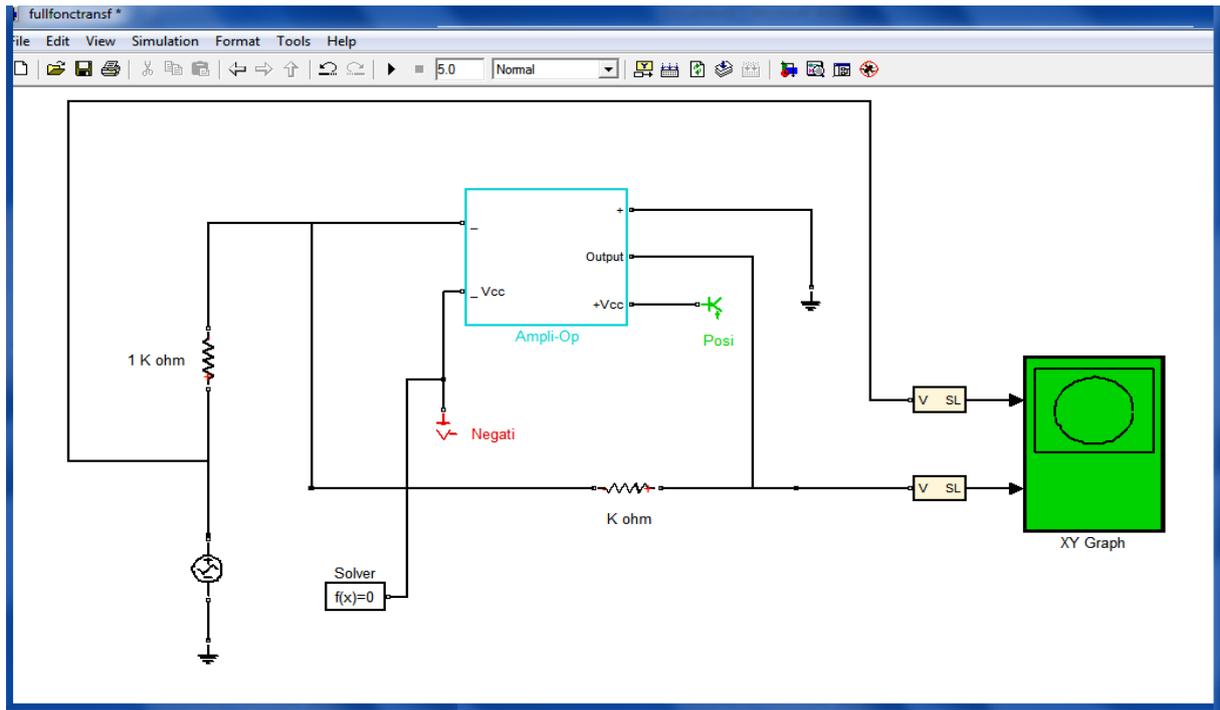
Pour ce type de visualisation nous mentionnons que le scope ne dispose que d'une seule entrée d'où une visualisation d'un seul signal. Dans le cas où plusieurs signaux sont à visualiser simultanément, ils seront liés par un bus de sélection (bus selector / multiplexage) comme le précise la figure 4-13.



**Figure 4.13 : multiplexage de signaux dans le Scope**

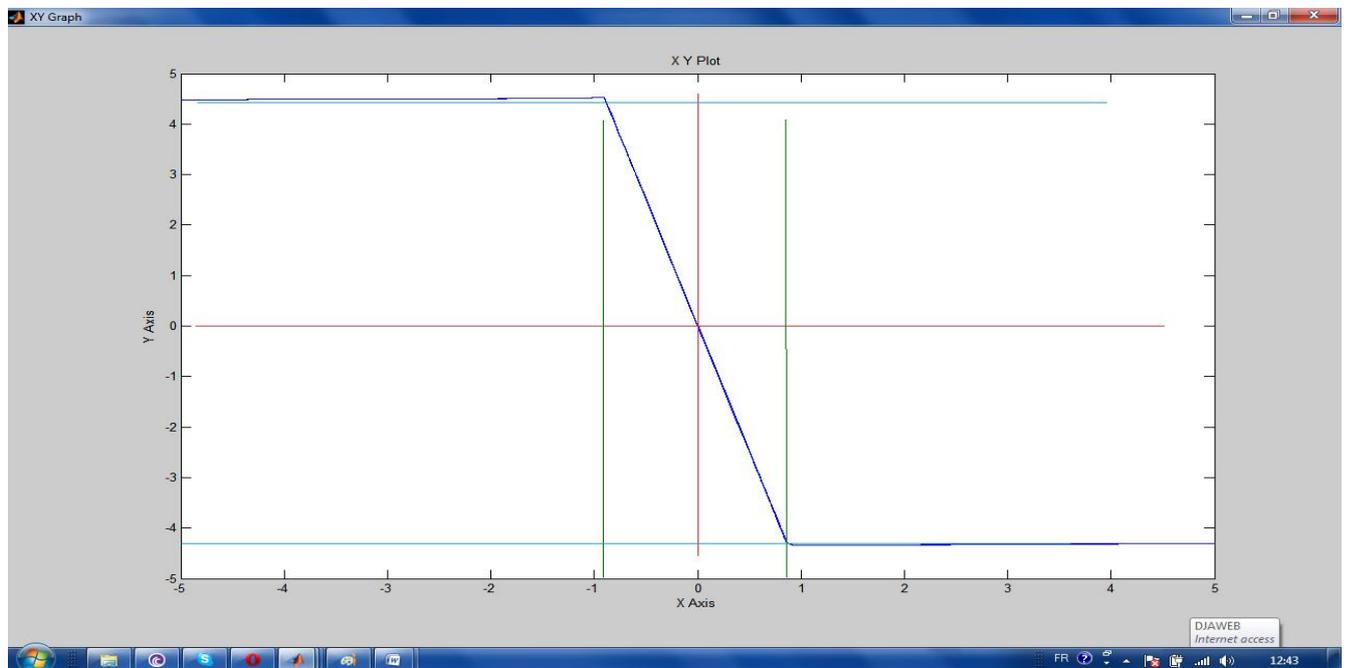
#### 4.5 Comparaison entre Simulink et PSPICE :

Les résultats de la simulation de la fonction de transfert de l'amplificateur –Inverseur à base du  $\mu A$  741 libre de faute par Simulink (figure 4-14) ont été comparés à ceux réalisés par le simulateur PSPICE [49]. Les courbes obtenues par ces simulations et présentées par les figures 4-15, et 4-16 confirment bien la validité de notre travail par l'usage du SIMULINK



**Figure 4.14 : Simulation de la fonction de transfert**

On rappelle que, le bloc (X-Y Graph) donné en figure 4-15 est pris de la **bibliothèque Sink**, pour assurer la représentation graphique des tensions d'entrée et de sortie injectées respectivement sur ses voies X et Y du circuit.



**Figure 4.15 : Caractéristique de transfert sous Simulink pour un Ampli-Op-inverseur avec gain de 5 sans fautes**

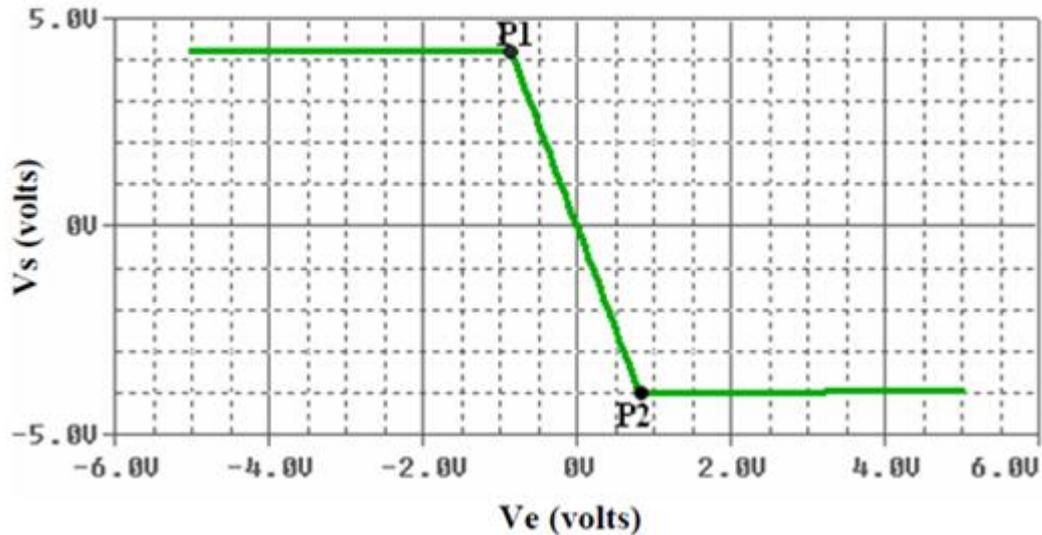


Fig. 4.16 :caractéristique de transfert sous PSPICE d'un ampli-inverseur avec Gain de 5

#### 4.6 Simulation de la technique BIST.

##### 4.6.1 La configuration numérique de la technique :

Cette tâche de simulation a été prise en charge par le simulateur SIMULINK mais dans un environnement numérique dont le but est de vérifier la validité de la technique BIST. A partir de la configuration de la technique BIST présentée au premier chapitre, on a commencé cette phase de simulation par l'insertion d'abord des différentes unités fonctionnelles de la structure du circuit de test, comme le présente la figure 4-17. Il s'en suit subséquemment l'établissement de liaisons entre ses différents éléments qui le constituent.

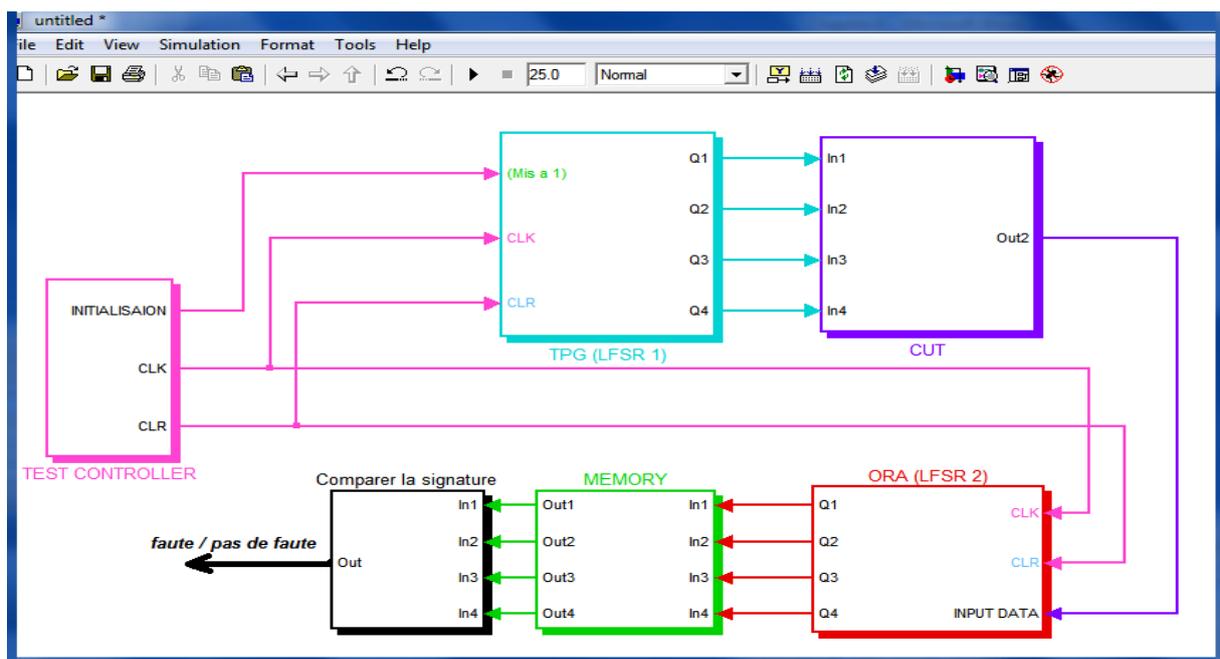


Figure 4.17 : la structure complète de BIST dans SIMULINK

#### 4.6.2 Vérification fonctionnelle de chaque unité :

##### A- Générateur de signaux de test (TPG) :

Le générateur de signaux de test utilisé est un ensemble de registres à décalage de type LFSR de 4 bits, dont le polynôme primitif correspondant est :  $P(x) = X^4 + X + 1$ , et la configuration LFSR équivalente est présentée en figure 4-18. Comme l'ordre du polynôme est de quatre alors il y'a nécessité de quatre bascules, et un seul coefficient non nul (à l'exception de  $X^4$  et  $X^0$ ) représente une porte OU-exclusif. La porte logique OR est employée seulement pour démarrer le cycle par un vecteur différent de (0000) au début de la simulation, à l'aide de la mise à 1 d'une de ses entrées.

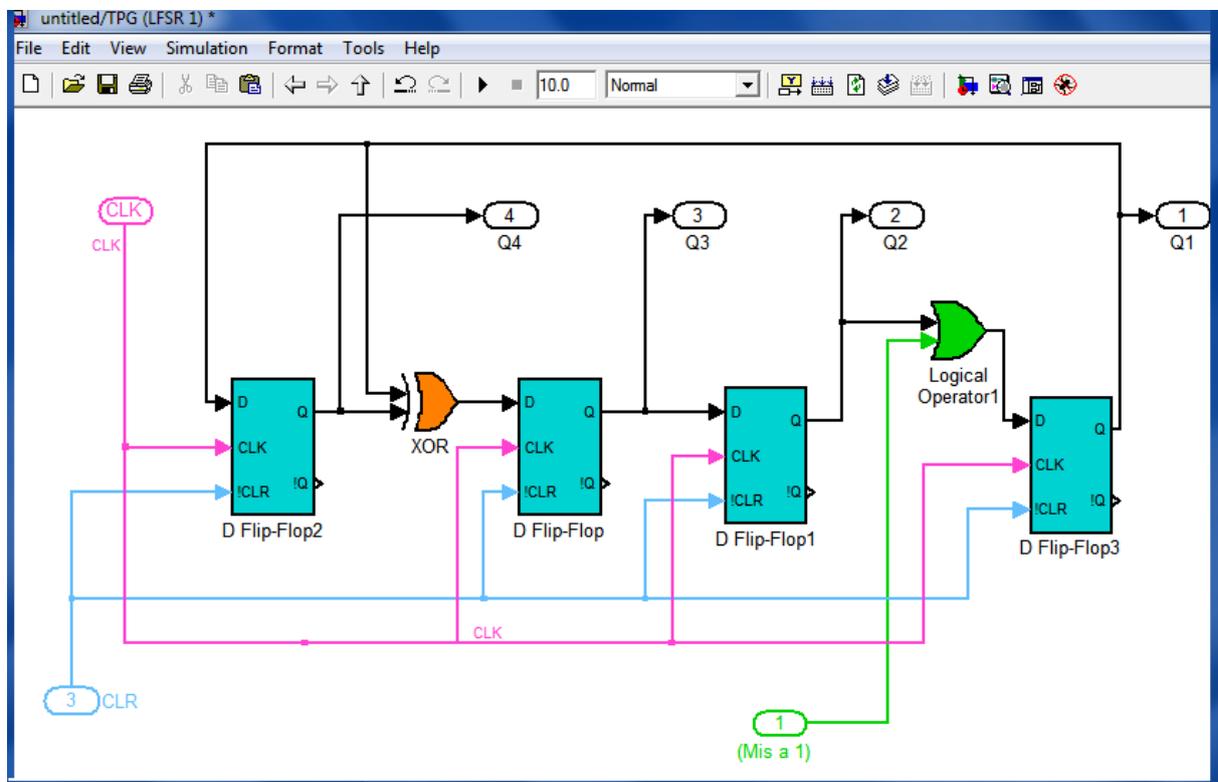
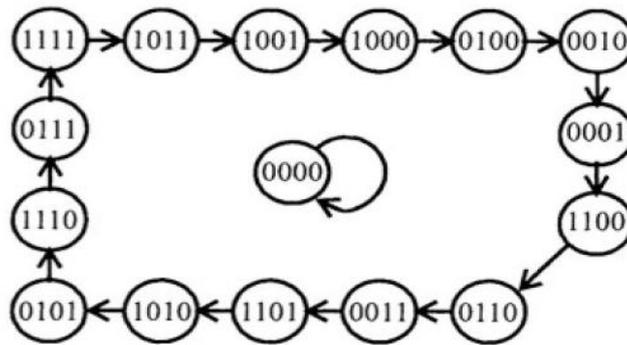


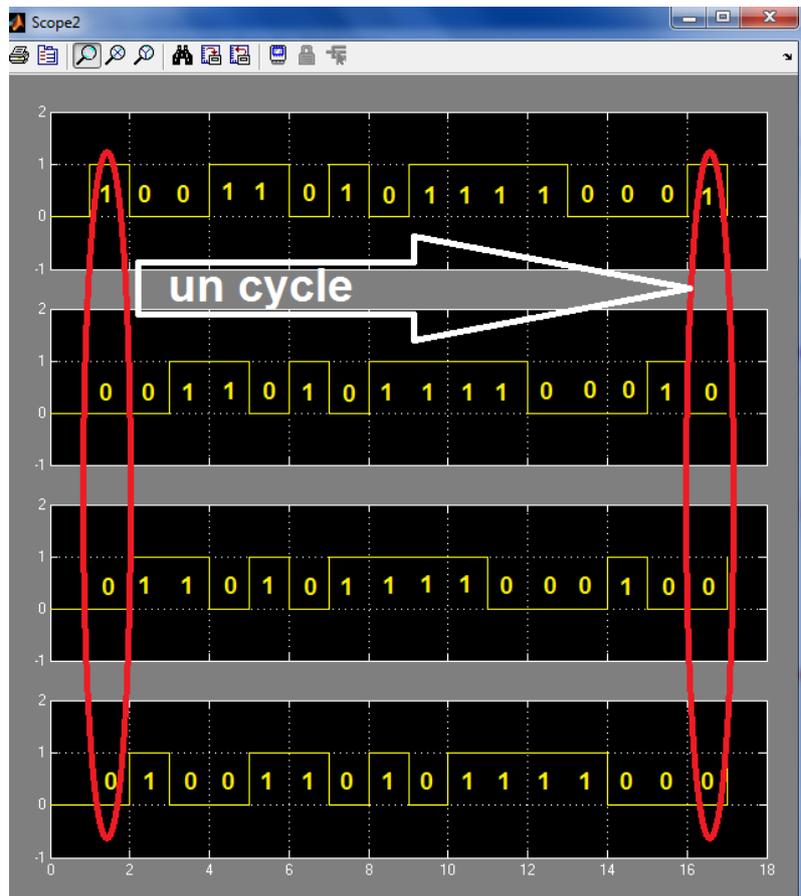
Figure 4.18 : générateur de signaux de test

La visualisation de ces vecteurs de test comme le montre figure 4-19 est rendue possible grâce à l'utilisation d'un Scope, et selon les mêmes séquences du cycle de test citées précédemment en théorie.



**Figure 4.19 : les séquences de vecteurs de test**

Pour pouvoir générer l'ensemble des vecteurs de test et éviter l'état de blocage au vecteur (0000), on force le TPG à démarrer par (0001) c.-à-d. mise à un d'un des bits du vecteur de test de la séquence parmi les 15 vecteurs restant qui se génèrent d'une manière pseudo-aléatoire .



**Figure 4.20 : cycle de vecteur de test avec un Scope**

### B- Analyseur de vecteur de réponse (ORA) :

L'ORA est une unité fonctionnelle indispensable dans le processus de test émanant de la technique BIST. Elle est constituée d'une structure LFSR comme l'indique la figure 4-21, presque semblable à celle de l'unité TPG. Seulement on a besoin d'une entrée pour les données qui proviennent évidemment du circuit sous test, le polynôme primitif correspondant est le même que celui du TPG ;  $P(x) = X^4 + X + 1$ .

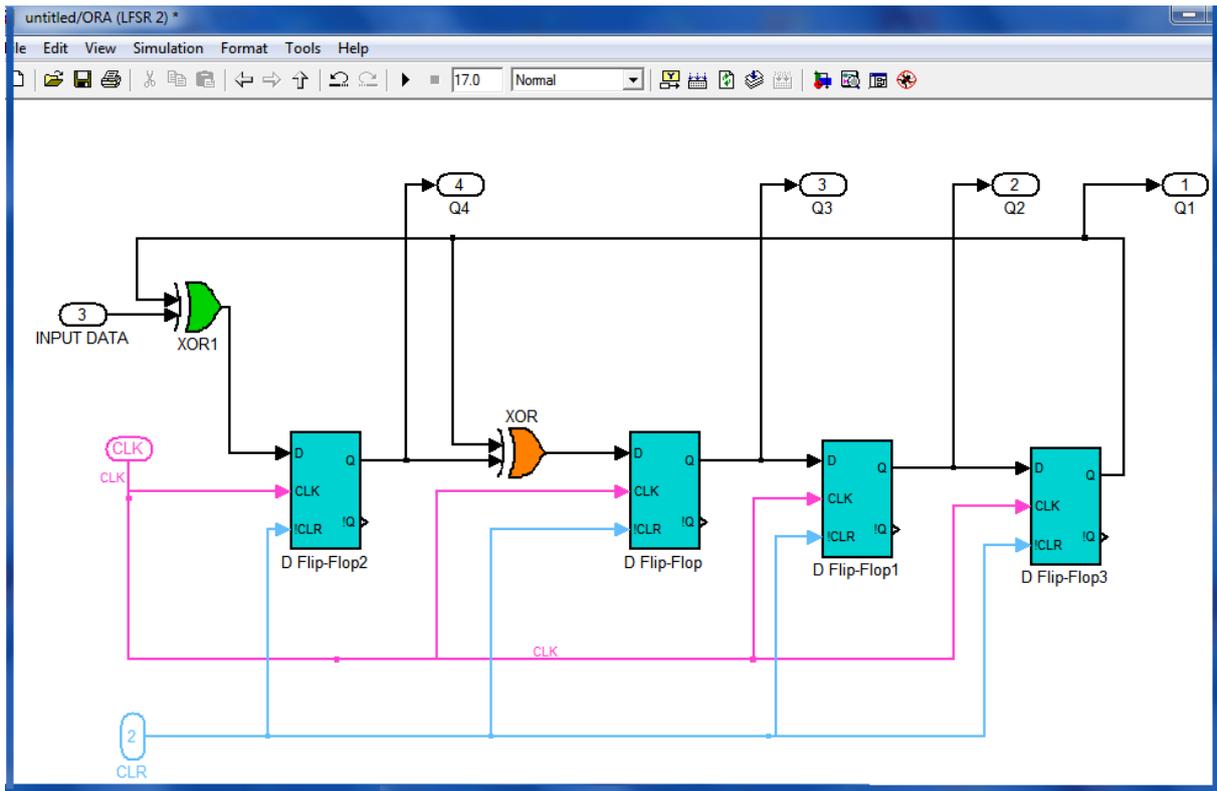


Figure 4.21: structure interne de l'ORA

La porte logique XOR1 est introduite pour un acheminement des données d'entrée venant du circuit sous test vers l'ORA. Celle-ci procède à leur analyse dont les résultats sont récupérés aux sorties Q1, Q2, Q3, et Q4 sous formes de vecteurs de 4 bits, et où on s'intéresse seulement au dernier qui constitue finalement "la signature".

### C- Les circuits mémoires :

Ce sont des cellules Read /Write dont la configuration est donnée en figure 4-22. Elles permettent la sauvegarde et la récupération des données pour des usages ultérieurs. Dans notre cas d'étude, seulement le dernier signal (la signature) est sauvegardé grâce au signal de commande transmis vers notre mémoire.

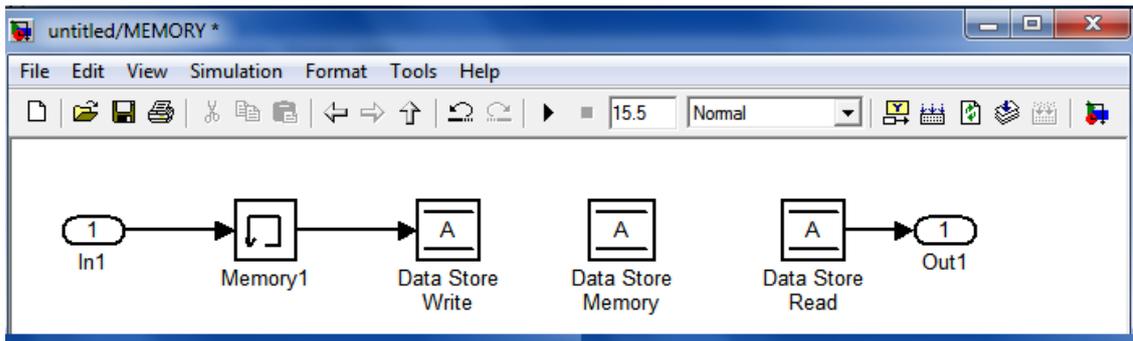


Figure 4.22 : structure interne de la mémoire

**D- Unité de comparaison :**

La signature pour un circuit sain est sauvegardée dans cette unité. La signature est reçue d'un circuit quelconque et est comparée à celle sauvegardée, si elle est la même on trouve (1) à la sortie ce qui signifie circuit sans faute, si non ce sera un (0).

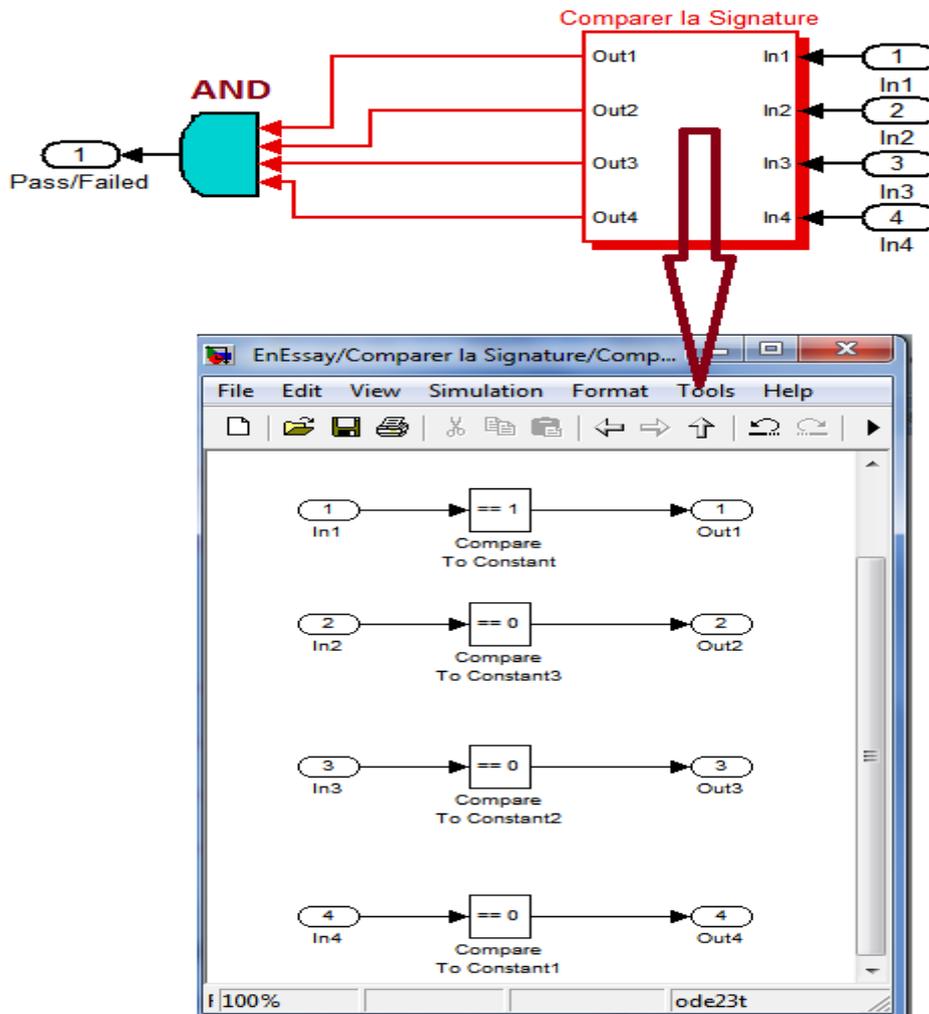


Figure 4.23 : la structure de l'unité de comparaison

### 4.6.3 Exemple d'application numérique :

On suppose un circuit combinatoire constitué d'un ensemble de portes logiques élémentaires comme le montre la figure 4-24 :

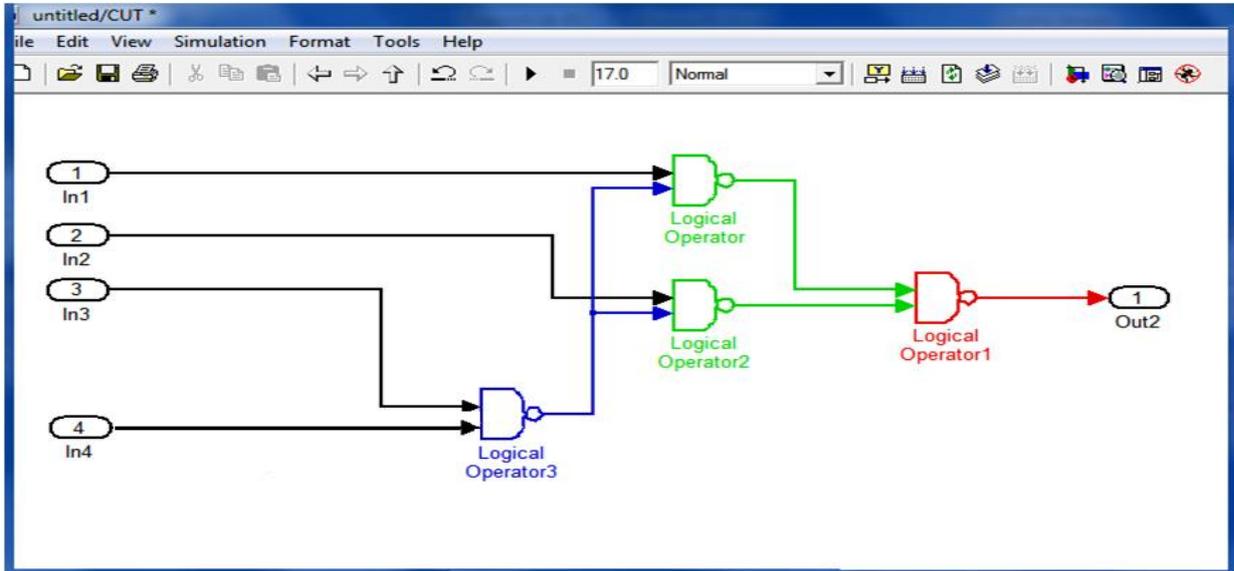


Figure 4.24: structure interne du circuit à testé

Ce circuit reçoit les mots binaires pseudo-aléatoires venant du générateur de vecteur TPG, à la sortie on aura une séquence de bits : 100111010111001

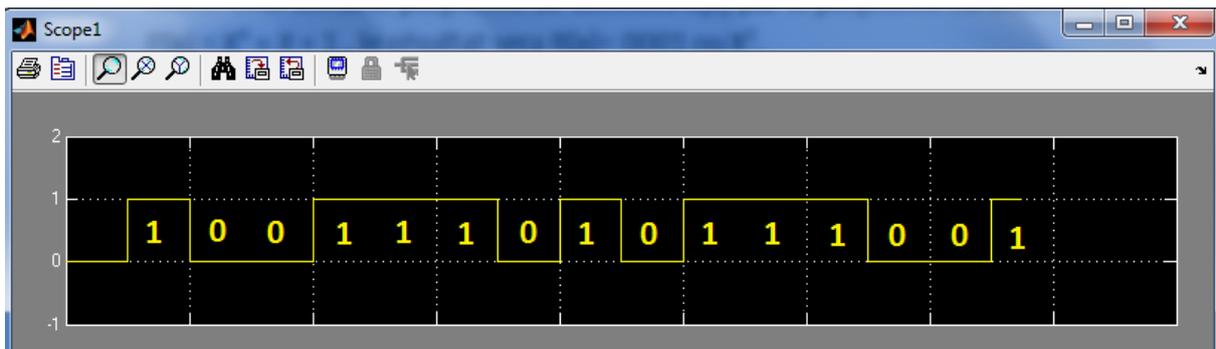


Figure 4.25 : la réponse chronologique du circuit combinatoire sous test

Cette séquence se traduit par un polynôme  $K(x)$  tel que :

$$K(x) = X^{14} + X^{11} + X^{10} + X^9 + X^7 + X^5 + X^4 + X^3 + 1.$$

Une fois appliquée au niveau de l'ORA comme le montre la figure 4-26, celle-ci procède à la division du polynôme de données  $K(x)$  par le polynôme caractéristique  $P(x) = X^4 + X + 1$ ,

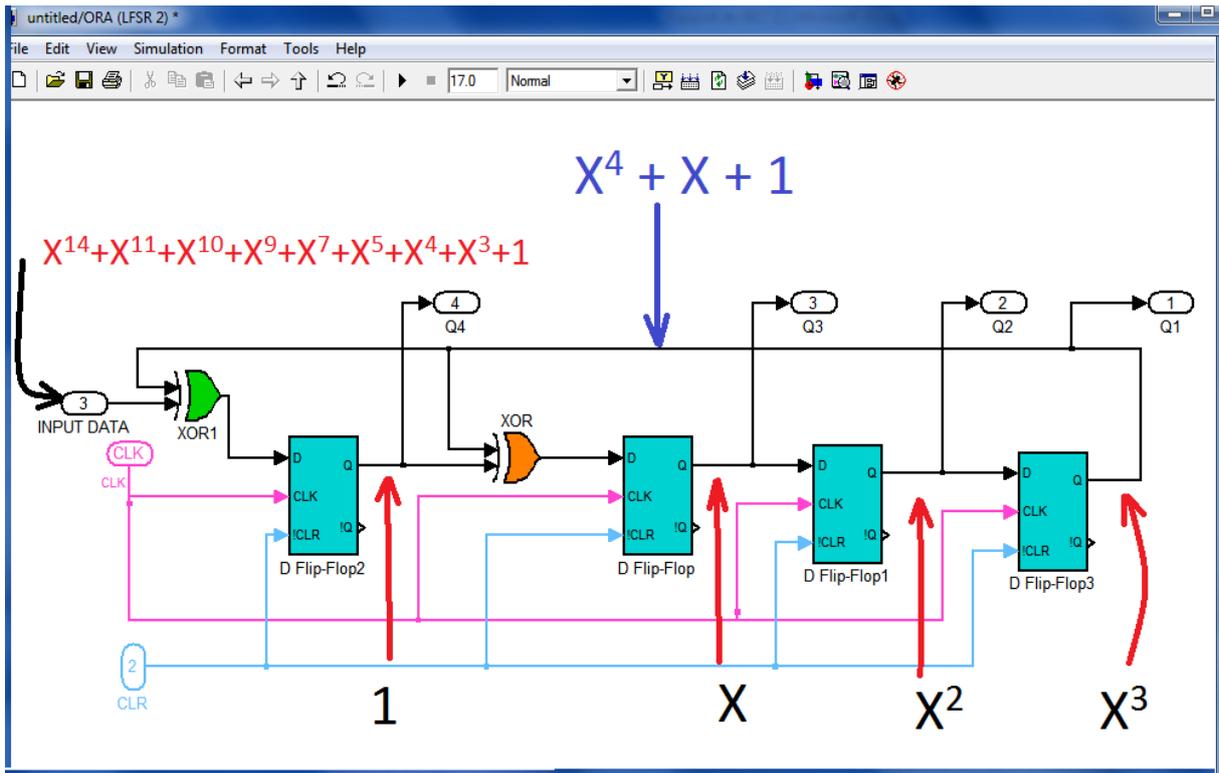
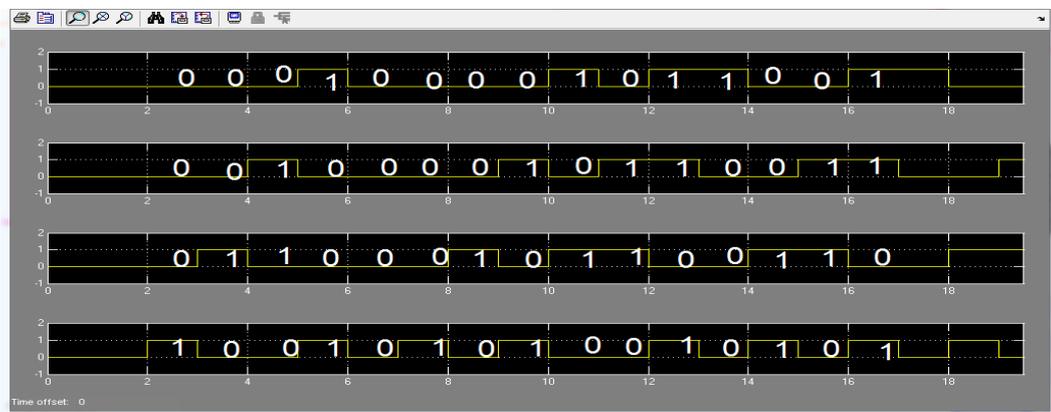
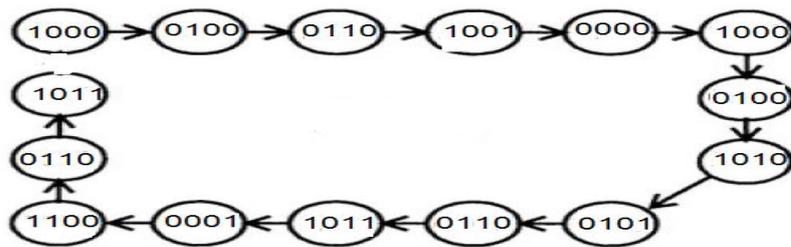


Figure 4.26 : la division polynomiale a l'intérieur de l'ORA

(a)



(b)



Figures 4.27 (a) et (b) : les séquences obtenue à la sortie de ORA

La figure 4-27 (a) illustre bien le cycle de séquence de vecteurs de test. Toutefois, on doit s'intéresser seulement au dernier vecteur qui représente la signature ( $Q^1 Q^2 Q^3 Q^4 = 1011$ ).

Cette signature correspond vraiment au polynôme résultant  $R(x) = 1 + X^2 + X^3$ , de la division du polynôme de donnée :  $K(x) = X^{14} + X^{11} + X^{10} + X^9 + X^7 + X^5 + X^4 + X^3 + 1$  par le polynôme caractéristique :  $P(x) = X^4 + X + 1$ .

Ce reste de la division est alors sauvegardé dans des circuits mémoires pour une éventuelle comparaison avec une signature du même circuit opérant dans d'autres conditions (circuit fautif par exemple).

#### 4.6.4 Insertion de fautes dans le circuit combinatoire:

Pour confirmer la validation de la structure BIST de test, on doit injecter une faute logique communément connue comme collage à 1 (stuck at 1) ou collage à 0 (stuck at 0) dans le circuit logique sain (voir figure 4.24). On procède par la suite à vérifier la nature de la signature. Dans notre exemple, un collage à 0 entre la porte logique OR et NAND à la quatrième entrée, le résultat sera différent conduisant évidemment et la signature change ( $Q_1 Q_2 Q_3 Q_4 = 1101$ ).

#### 4.7 Circuits d'adaptation entre l'Analogique et le Numérique.

L'adaptation du circuit de test (structure BIST) au circuit à tester (CUT) de nature analogique exige la conversion de signaux de données de test ou à tester en signaux de données acceptés par le circuit cible. Il s'ensuit une insertion entre ces 2 structures de deux convertisseurs analogique-numérique (CAN) et numérique-analogique (CNA) comme l'illustre la figure 4-28.

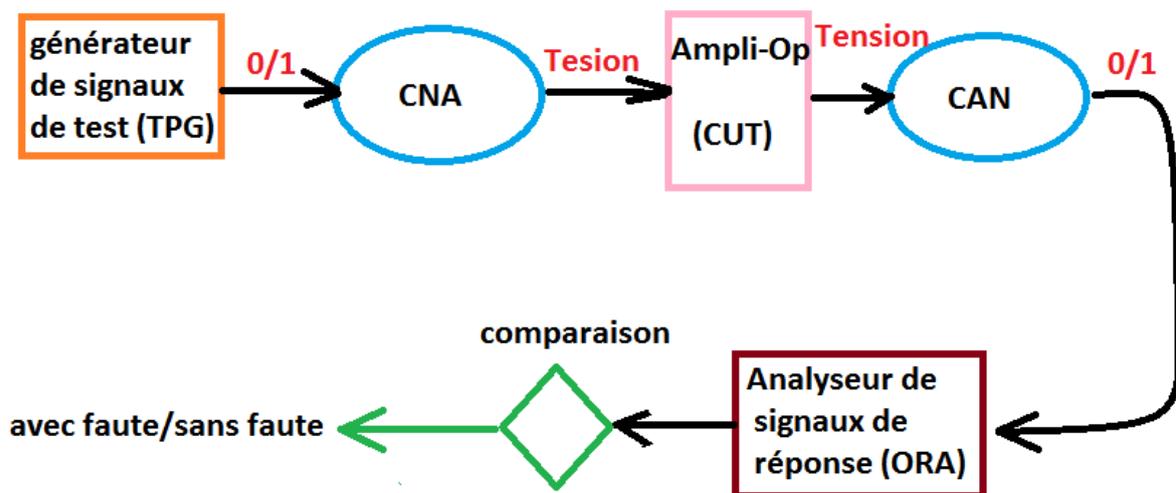
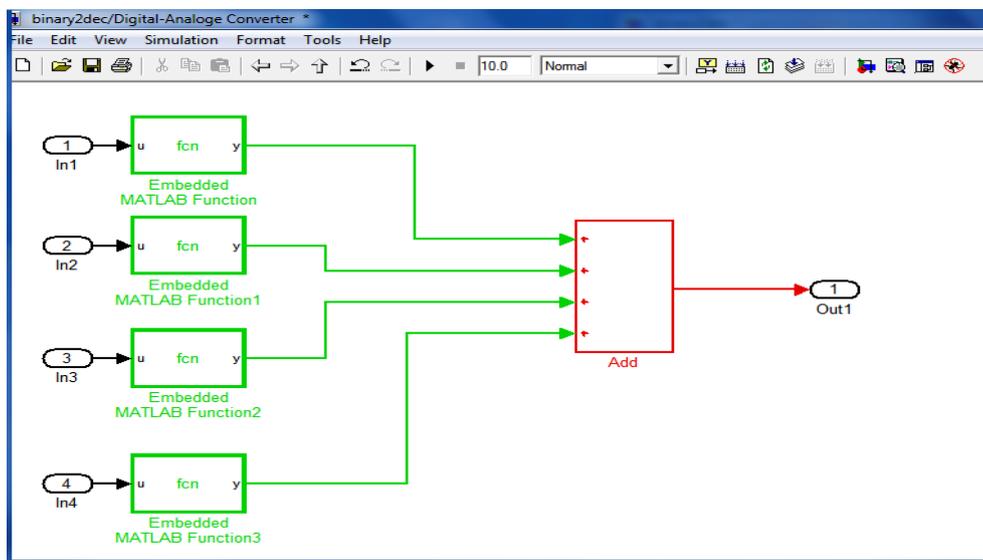


Figure 4.28 : schéma électrique représente la conversion AN-NA

L'objectif à atteindre est d'arriver à tester par la technique numérique de test BIST un circuit analogique, qui dans notre cas est un Ampli-Op inverseur.

#### 4.7.1 Conversion numérique analogique avec Simulation :

Le non disponibilité de CNA et CAN dans la bibliothèque du SIMULINK qui répondent à nos besoins, nous a poussé à créer des modèles mathématiques basés sur le même principe connu de conversion d'une donnée numérique en donnée analogique. Son illustration est donnée en figure 4-29.



**Figure 4.29: conversion numérique analogique**

Le but de ce processus de conversion est de livrer à partir d'un vecteur de 4 bits provenant du TPG des valeurs décimales qu'on attribue aux signaux analogiques. Ces derniers représentent les signaux d'entrée de l'amplificateur –inverseur et peuvent prendre des valeurs aussi bien positives que négatives.

- **Simulation de la conversion numérique analogique**

Pour illustrer cette tâche de conversion, on prend le cas d'un signal d'entrée analogique à générer et qui doit se situer dans la région linéaire de notre Amplificateur -inverseur définie dans l'intervalle  $-0.8 < V_e < +0.8$ . Alors, le vecteur de test sera l'équivalent d'un nombre décimal avec virgule qu'on peut obtenir à partir de l'expression :

$$V_e = a_1 * 2^{-1} + a_2 * 2^{-2} + a_3 * 2^{-3} + a_4 * 2^{-4}$$

Un signal  $V_e$  de 0.5V par exemple sera généré par le vecteur binaire 1000 déduit par l'opération suivante :

Valeur décimale :  $(1000)_{10} = 1 * 2^{-1} + 0 + 0 + 0 = 0,5$ .

Cette opération de conversion de vecteur de test (1000) en un signal électrique 0.5V est simulée sous MATLAB comme le présente la figure 4-30.

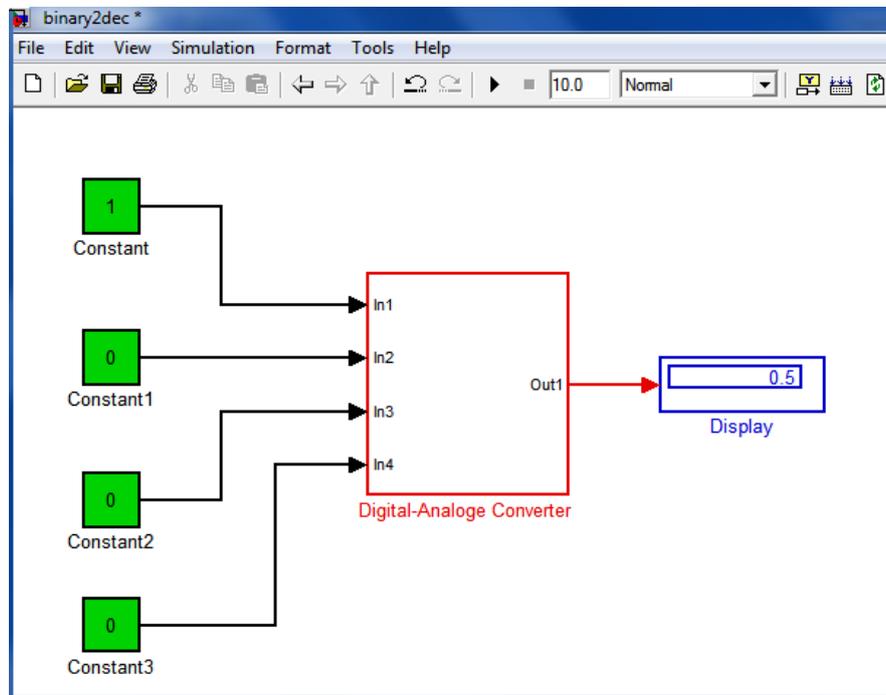


Figure 4.30: simulation du CNA

#### 4.7.2 Conversion Analogique Numérique :

##### A- La conversion de la partie entière

Le signal à la sortie du circuit sous test (ampli-Op) est un signal analogique qui nécessite d'être converti en un signal codé binaire pour subir un traitement logique de la part de la structure BIST. Le principe de ce type de conversion est similaire à celui du CNA mais inversé. Il s'effectue en 2 parties séparées :

- La conversion de la partie entière.
- La conversion de la partie décimale.

\*- Principe de la conversion de la partie entière :

Cette opération s'effectue par la division successive d'un nombre par deux, dont les restes correspondent à des bits 0 et 1. La simulation de cette tâche est réalisée grâce à la disponibilité

des fonctions mathématiques qu'offre le simulateur, telles que les valeurs absolues, les constantes, des opérations de division etc... Une illustration de ce processus est donnée en figure 4-31. On fait remarquer ici les étapes essentielles qui consistent en :

- 1- la division par un bloc renfermant un ensemble d'opérations exécutables selon la configuration indiquée dans cette figure.
- 2- La représentation du LSB ;
- 3- La représentation du signe par le MSB
- 4- Séparation du signe (-) de la partie entière du chiffre dans le cas des valeurs négatives (voir figure 4-32)

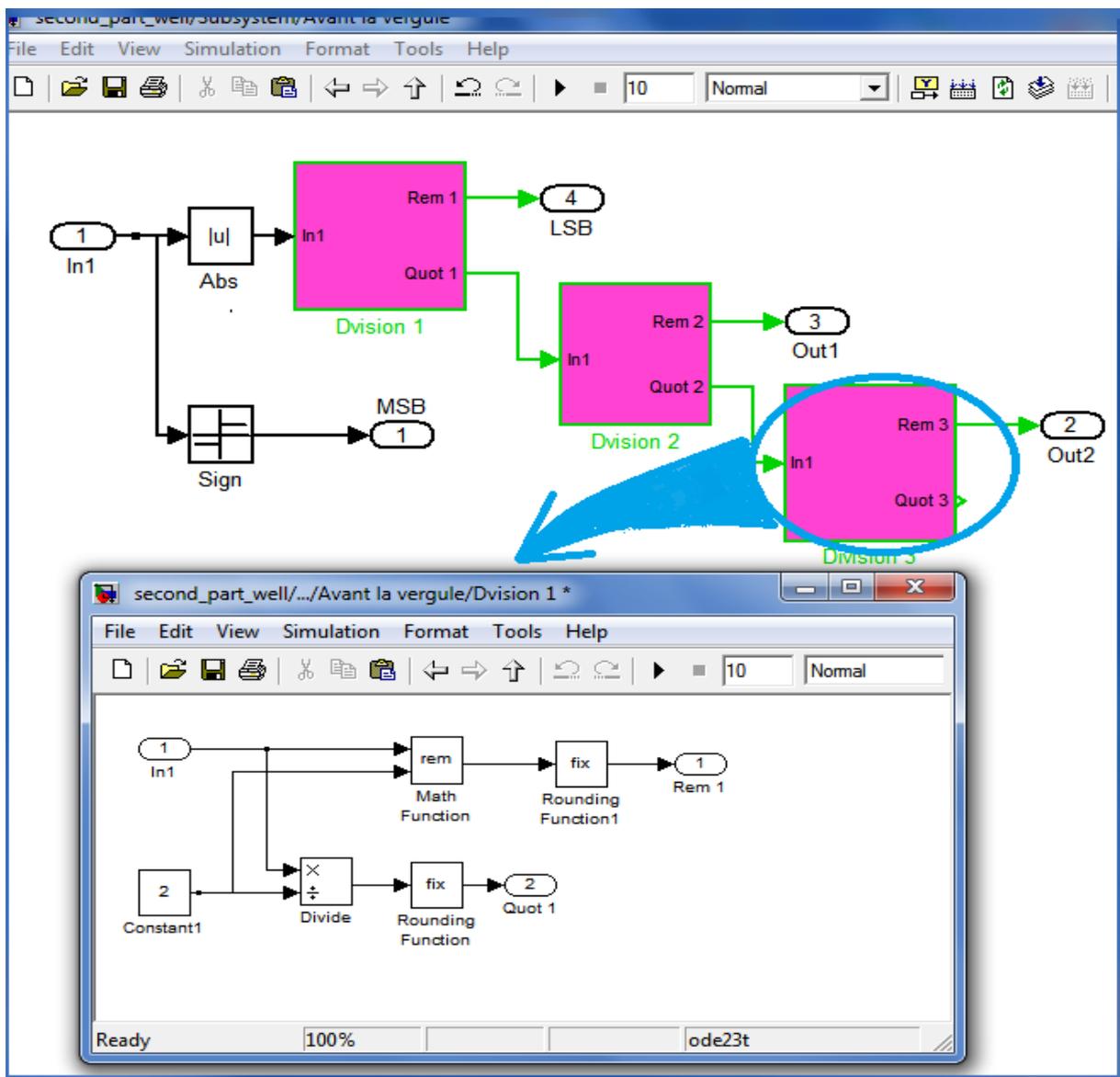


Figure 4.31: Détails du Bloc de conversion

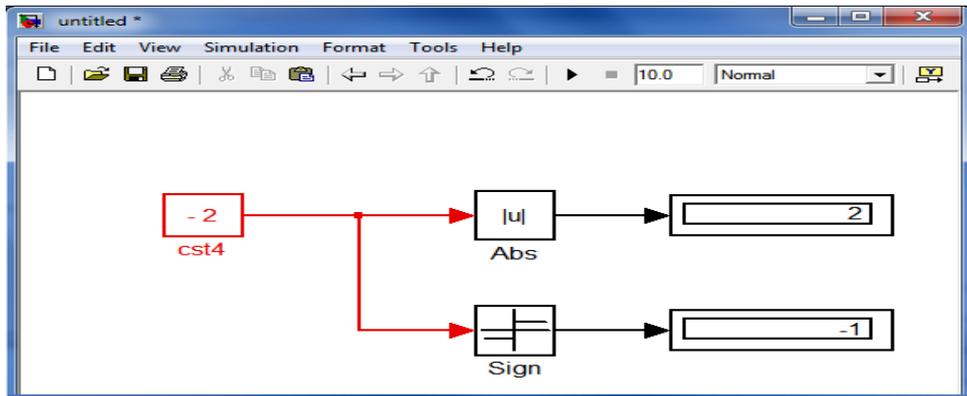


Figure 4.32: présentation de la séparation du signe du chiffre

**B- La conversion de la partie décimale d'un chiffre :**

A l'opposé du premier type de conversion, la partie décimale est convertie en binaire (codé en 4 bits) à partir d'un certain nombre de multiplications par deux et comparaison de leur résultat par rapport à 1. Si le résultat est inférieur à 1 le bit sera égale à 0, dans le cas contraire le bit sera égale à 1. Dans ce cas-ci on lui soustrait par la suite 1 et on refait les mêmes opérations avec le nombre du reste. Cette opération continue jusqu'à l'obtention du 4<sup>ème</sup> bit de la représentation binaire. Cette phase d'opérations est décrite en figure 4-33 ou les bits de la conversion sont obtenus aux sorties des blocs (sous systèmes) effectuant chacun les opérations citées au préalable.

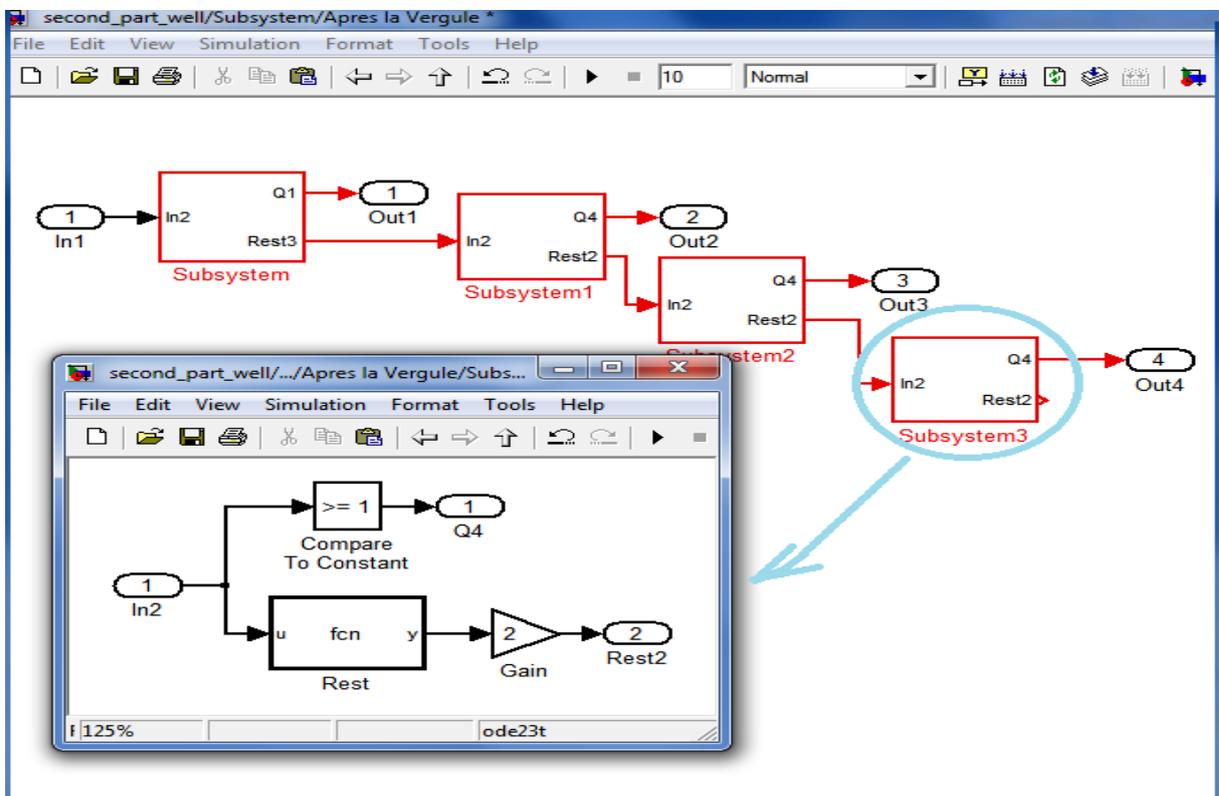


Figure 4.33: conversion de la partie décimale

**C- Le signe :**

Dans ces deux types de conversion CNA et CAN on aura toujours besoin d'un bit pour le signe. Pour la partie entière, seuls les 3 premiers bits de la CAN/CNA sont suffisants pour retrouver la valeur en base de 10 et qui est comprise entre -4 et +4. Cette restriction de valeurs est imposée par les niveaux de saturations positive et négative du circuit sous test. Le quatrième bit (le MSB) est employé pour indiquer le signe positif ou négatif d'un nombre. Au niveau du CNA, les vecteurs binaires proviennent du TPG de la structure BIST comme l'indique la figure 4-34, seulement la représentation du nombre décimal fait restriction pour le signe par rapport à celle de la partie entière : le signe (-) est représenté par le dernier bit à 1.

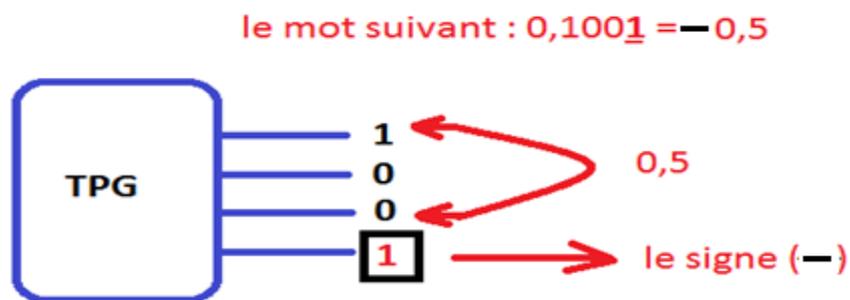


Figure 4.34: le bit qui représente le signe +/-

**D- La simulation de la conversion analogique numérique.**

A travers cette partie, on illustre à partir de la figure 4-35, le processus de simulation analogique numérique appliqué sur une valeur décimale négative.

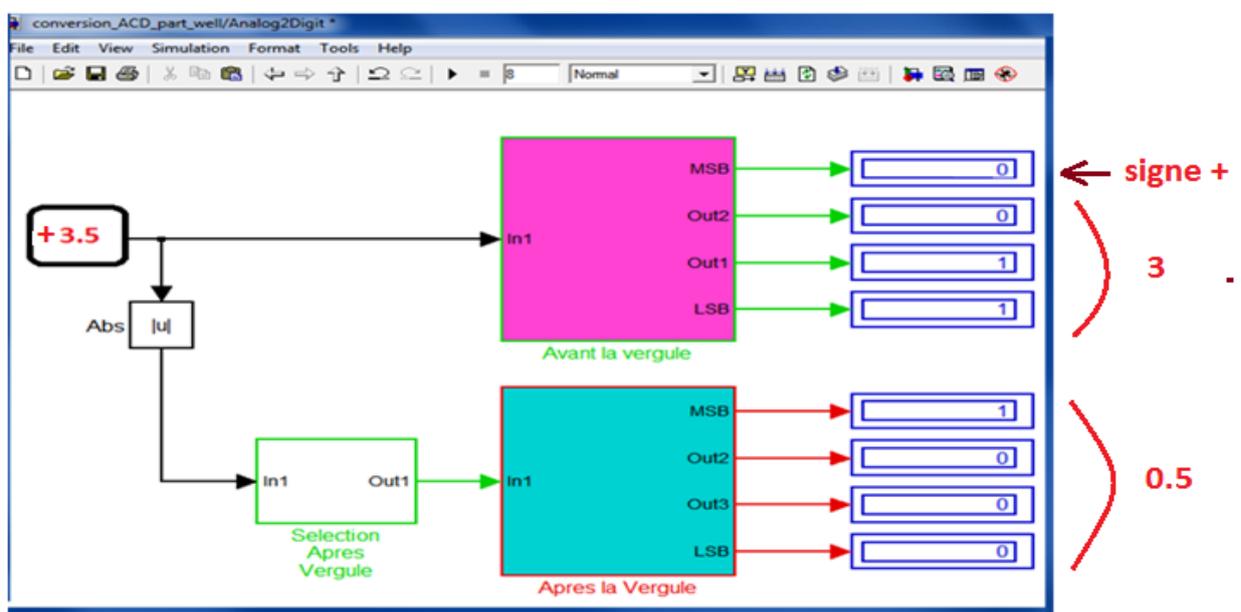


Figure 4.35: conversion d'un nombre décimal avec signe

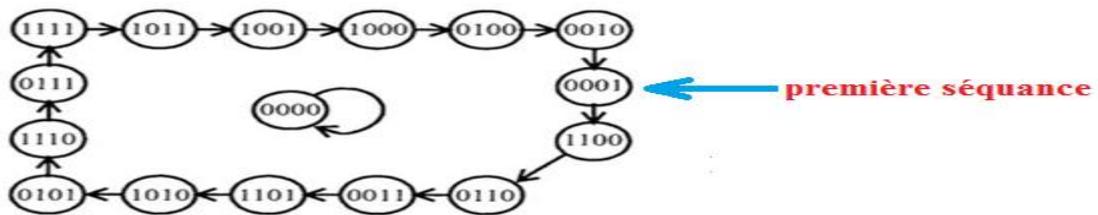
Le bloc le plus haut sert à convertir la partie entière en un mot de 4 bits, où le premier est destiné au signe et les trois autres pour la valeur ( $3 = 011$ ). L'autre bloc de conversion un mot de 4bits également s'occupe de la partie décimale une fois débarrassée du signe par la fonction Abs et séparée de la partie entière par un bloc de sélection.

**4.8 Contrôleur de test :**

Le processus de test de circuits par une telle structure s'effectue par l'accomplissement d'un certain nombre d'actions qui se définissent comme transfert de vecteurs de données à convertir, réception de donnée à analyser, leur mémorisation, génération de la signature, activation des registres LFSR . Ces fonctions sont activées grâce un certains nombres de signaux de commandes qu'on décrit comme suit.

**1- Initialisation :**

Le générateur de vecteur de test TPG est un LFSR de quatre bascules D qui assure la génération des signaux sous forme des séquences pseudo-aléatoires :



**Figure 4.36 : les séquences de TPG**

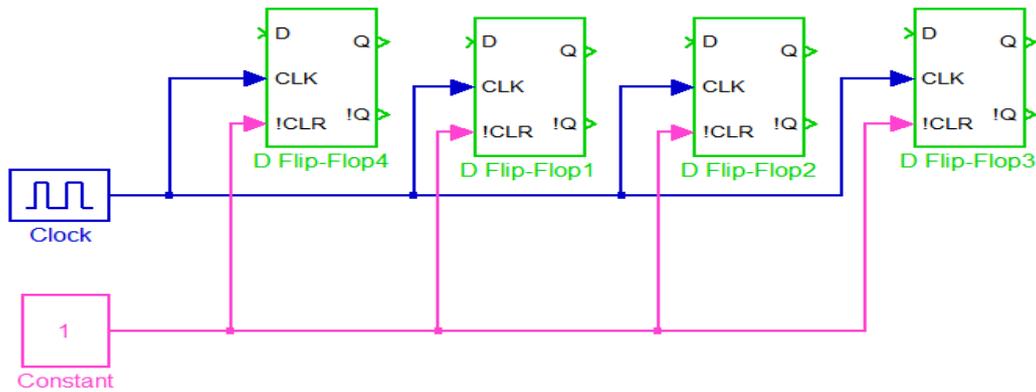
Son initialisation consiste à éviter la séquence (0000), par un forçage d'une bascule D à démarrer par 1 (Mise à un) a l'aide d'une impulsion de durée d'une seconde générée par un Timer qu'on applique à son entrée D (voir figure 4-37).



**Figure 4 .37 : Mise à un de la bascule D**

**2- L'horloge CLK et remise à zéro CLR :**

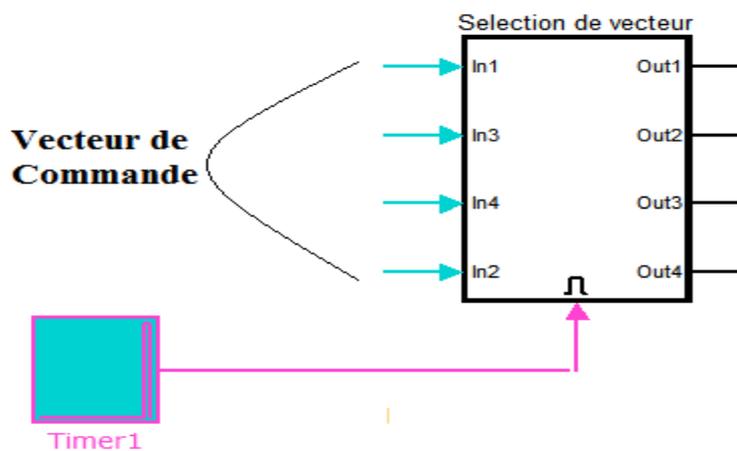
Les LFSR du générateur de vecteur de test TPG et de l'analyseur de signaux de réponse sont composés de bascules D dont l'horloge est indispensable pour leur fonctionnement, et le CLEAR sert à leur remise à zéro pour chaque démarrage de simulation et pour s'assurer de l'inexistence de bit ou d'état antérieure sauvegardé.



**Figure 4.38 : le signal d'horloge et le signal CLEAR**

**3- Signal de sélection :**

Le TPG employé dans notre structure de test génère des séquences de 15 bits, seulement un seul vecteur bien déterminé doit être sélectionné pour vérifier la fonctionnalité du circuit à tester. Un bloc de sélection est créé pour laisser passer seulement le vecteur choisi comme le montre la figure 4-39. La séquence de test est alors sélectionnée par le bloc suite à son activation par un signal du Timer.



**Figure 4.39 : le bloc de sélection du vecteur de commande**

4.9

Simulation du processus de test numérique pour un circuit analogique .

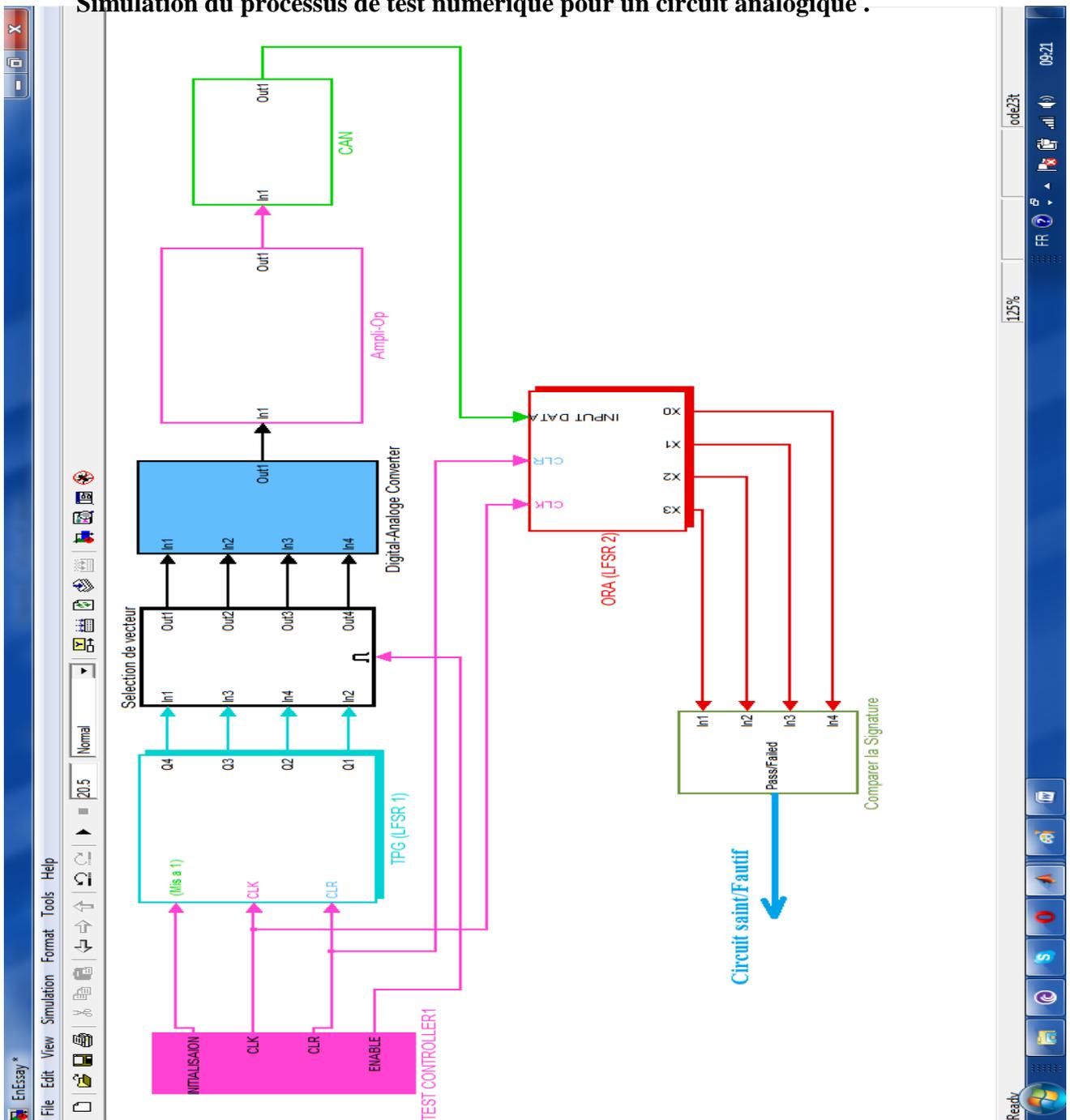


Figure 4.40 : le schéma complet de la technique BIST dans le SIMULINK

4.10 Description des signaux :

Un des points importants dans le fonctionnement correcte d'une structure de test est la bonne gouvernance du système à partir d'un ensemble de signaux chacun orienté vers une fonction bien précise. On distingue 2 sortes de signaux générés par le contrôleur de test : les signaux de commande et les signaux de traitement et d'analyse de données.

#### 4.10.1 Les signaux de commande .

Ces signaux sont destinés à générer les vecteurs de test par l'unité TPG, à activer les différents registres ou bascules de l'unité LFSR, à contrôler le fonctionnement de l'unité ORA. Leur mise en valeur s'illustre à partir de l'exemple d'application suivant :

La vérification du bon fonctionnement de l'amplificateur inverseur par la structure de test BIST s'effectue par l'envoi d'un signal analogique dont la valeur doit se situer dans un intervalle caractérisant l'une des régions opérationnelles de ce circuit à savoir les 2 régions de saturations négative et positive et la région linéaire. Pour ne citer que celle-ci, et eu égard aux caractéristiques de ce circuit (gain en tension  $G_v = -5$ ) le signal à amplifier doit se trouver dans un intervalle de  $-0.85$  à  $+0.85$ .

Si la tension à amplifier est de  $-0.5$  V, le vecteur de test à générer par l'unité TPG qui est choisi parmi les 15 vecteurs de séquences de test doit correspondre au vecteur ayant la représentation binaire (1001) comme l'illustre la figure 4.41.

Le Scope 3 représente les signaux de tests venants de la TPG, et le Scope 5 montrera le signal binaire sélectionné pris à la sortie du bloc de sélection qu'on présente sous forme de vecteur (1001). Celui-ci qui après conversion en valeur analogique est appliqué à l'Ampli-Op monté en inverseur.

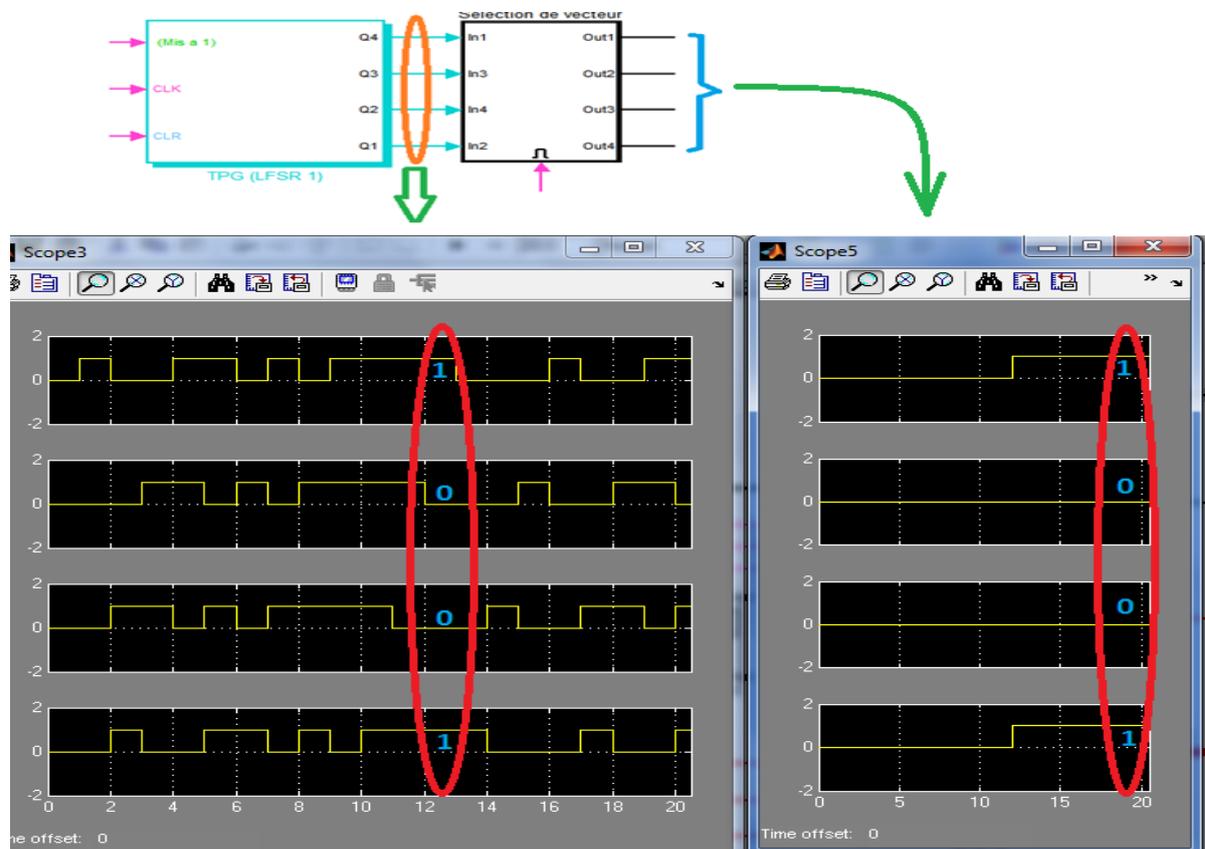
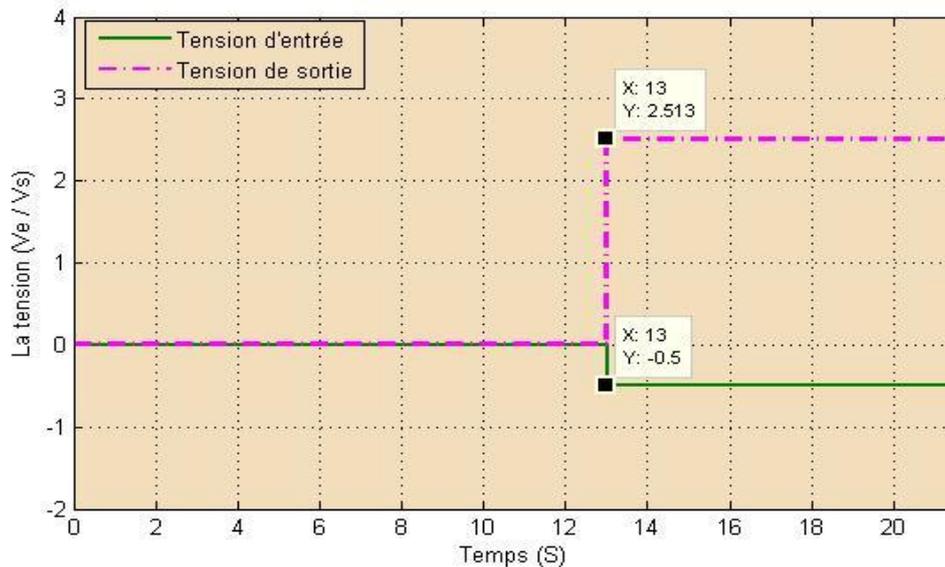


Figure 4.41 : séquences de test généré par le TPG et le vecteurs sélectionné

### 4.10.2 Les signaux à traiter :

#### 4.10.2.1 Les tensions $V_s$ et $V_e$ :

Le convertisseur CNA convertit le vecteur (1001) en une tension  $V_e = -0.5V$ , la tension de sortie sera égale à  $V_s = +2.5V$ . La figure 4-42 présente ces résultats sous Simulink.



**Figure 4.42: la tension d'entrée ( $V_e$ ) et de sortie ( $V_s$ ) de  $\mu A741$**

Le vecteur de test choisi (1001) correspond à celui de la 12<sup>ème</sup> séquence générée par le TPG, alors cette séquence démarrera à l'instant  $T = 12$  s car chaque séquence dure une seconde. A cet instant l'Amplificateur inverseur reçoit la tension d'entrée ( $V_e$ ) de  $-0.5$  V et fournit à sa sortie une tension telle que  $V_s = +2.5$  V (voir figure 4.42).

#### 4.10.2.2 La conversion de la tension $V_s$ en binaire :

Pour que le signal soit analysé par la structure de test il doit être de nature numérique. Dans l'exemple d'étude choisi, le signal de tension de sortie du circuit sous test, de nature analogique, doit subir une conversion ou codification en binaire. Cette tension de sortie ( $V_s$ ) obtenue après amplification ( $+2.513$  V) aura une représentation en un mot à 8 bits qui s'écrira (00101000) dont les 4 premiers bits concernent le signe et la partie entière du chiffre et les 4 autres bits restant pour la partie décimale. Cette représentation peut être illustrée par le chronogramme donné en figure 4.43.

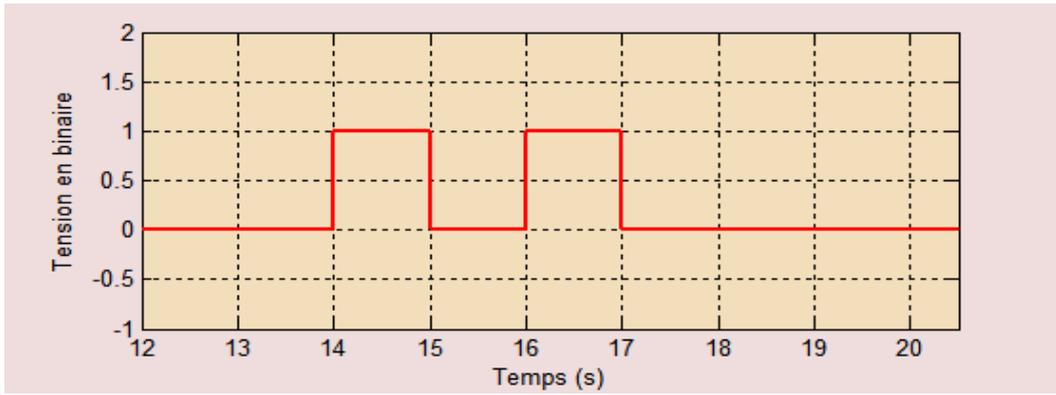


Figure 4.43 : la tension de sortie de l'Ampli-Op convertie en Binaire

Selon la figure précédente, le convertisseur analogique numérique convertit la tension de sortie de l'Ampli-Op (+ 2.513 ) à un signal numérique : 00101000 .



#### 4.10.2.3 La signature :

Ce mot binaire obtenu suite à la conversion précédemment mentionnée est transmis vers l'analyseur de réponse l'ORA pour nous fournir après traitement une signature comme le présente scope 4 de la figure 4.44.

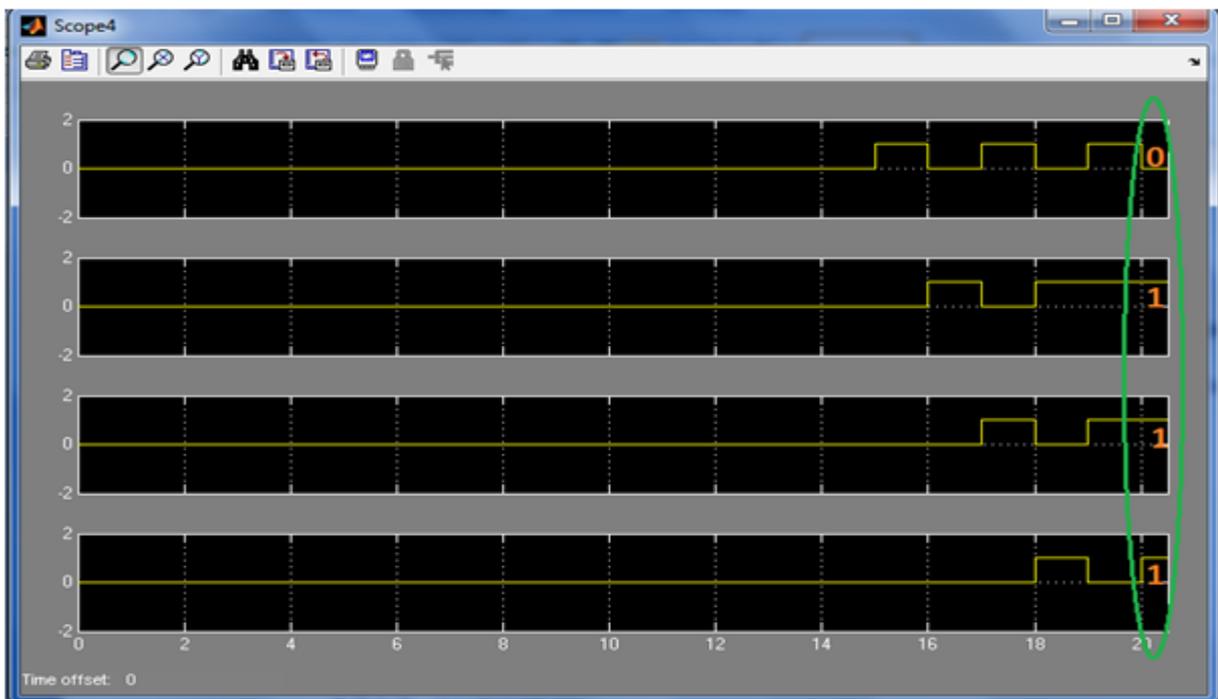


Figure 4.44 : la signature a la sortie de l'ORA

Les signaux récupérés à la sortie de l'ORA au nombre de 4 constituent le résultat de traitement du signal. Chaque signal est en fait une séquence binaire dont le dernier état représente un bit de la signature dont la valeur binaire affichée par les scopes selon l'exemple ci-dessus est de 1110.

#### 4.10.3 Comparaison avec la théorie :

En technique BIST comme décrit au Chapitre 1, le vecteur de test appliqué au circuit sous test (Amplificateur inverseur) conduit à une réponse qui, convertie en un signal numérique, peut se mettre sous forme d'un polynôme  $K(x)$ , reçu par l'ORA,  $K(x)$  subit une division polynomiale par le polynôme primitive de cette unité de traitement  $P(x) = X^4 + X + 1$  (voir le paragraphe de TPG). Cette opération se décrit comme suit :

- Passage d'une représentation binaire en un polynôme dont les termes sont ceux correspondant aux bits égaux à 1 de la séquence binaire

$$\begin{array}{cccccccc} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ | & & & & & & & | \\ X^7 & + & X^6 & + & X^5 & + & X^4 & + & X^3 & + & X^2 & + & X & + & 1 \end{array}$$

Alors le signal binaire se présente sous forme d'un polynôme  $K(x) = X^5 + X^3$ .

- La division polynomiale de  $K(x)$  par  $P(x)$  aboutit au reste  $R(x)$  :
- La division d'un polynôme d'ordre 7 par un autre d'ordre 4 mène à un polynôme d'ordre 3 au maximum représentant le reste  $R(x) = aX^3 + bX^2 + cX + d$  ou les facteurs  $a, b, c, d$  représentent le nombre binaire (1 ou 0). On obtient la signature:

$$\begin{array}{l|l} K(x) = \cancel{X^5} + X^3 & P(x) = X^4 + X + 1 \\ \cdot \cancel{X^5} + X^2 + X & X \\ \hline R(x) = X^3 + X^2 + X & \end{array}$$

Le reste  $R(x) = X^3 + X^2 + X$  est la signature qu'on a trouvée pratiquement à la sortie de l'ORA, ( $X^3 + X^2 + X = 1110$ ). Ce polynôme est sauvegardé dans une mémoire pour le comparer à une signature comme par exemple celle du circuit fautif. Cette mémoire est contenue dans le bloc surnommé **comparer la signature**.

#### 4.11 Test par BIST de circuit fautif.

##### 4.11.1 Choix de la faute particulière à injecter :

L'objectif fixé est de réussir par voie de simulation de l'opération de test par la technique BIST sur le circuit  $\mu A741$  monté en inverseur en l'absence et présence de fautes. Le choix et le nombre limité à 2 fautes sont établis d'une façon aléatoire car l'importance donnée ici concerne l'efficacité de cette opération sur des circuits analogiques. Ces fautes choisies pour le test sont des fautes dures. Il s'agit d'un circuit ouvert à la base du transistor QN10 (voir figure 4.45) qu'on a codé comme (BOQN10) et l'autre faute est un court-circuit entre base-émetteur du transistor QP2 sous le code CCBEQP2

##### 4.11.1.1 La faute de circuit ouvert (BOQN10):

Cette faute a été simulée par SIMULINK au moyen d'une rupture de connexion au niveau de la base du transistor QN10 qui constitue l'amplificateur du circuit à tester. Le résultat de la simulation obtenu graphiquement (figure 4-46) est comparé au résultat obtenu par le simulateur PSPICE (figure 4.45) pris des travaux antérieures développés sur le même circuit analogique d'étude [49].

##### a) Simulation par PSPICE :

La faute BOQN10 (base ouverte de transistor QN10) :

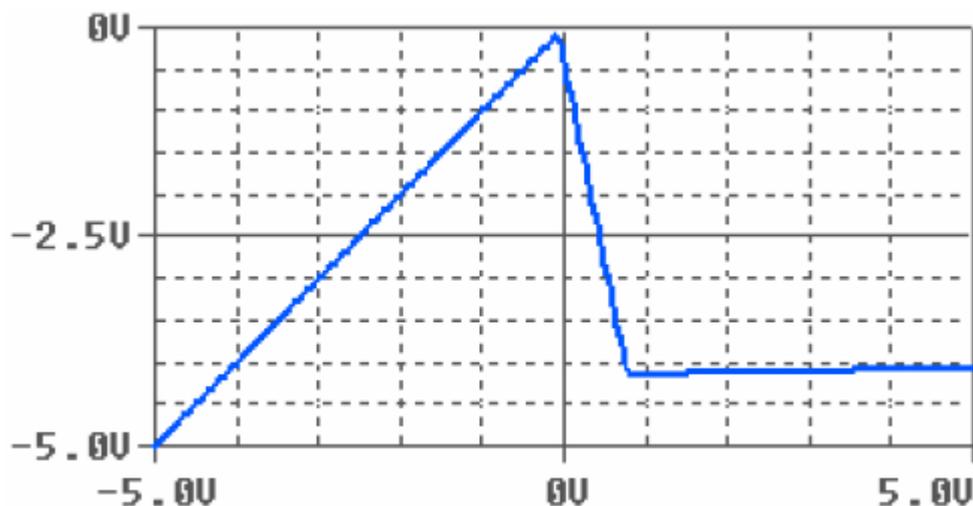
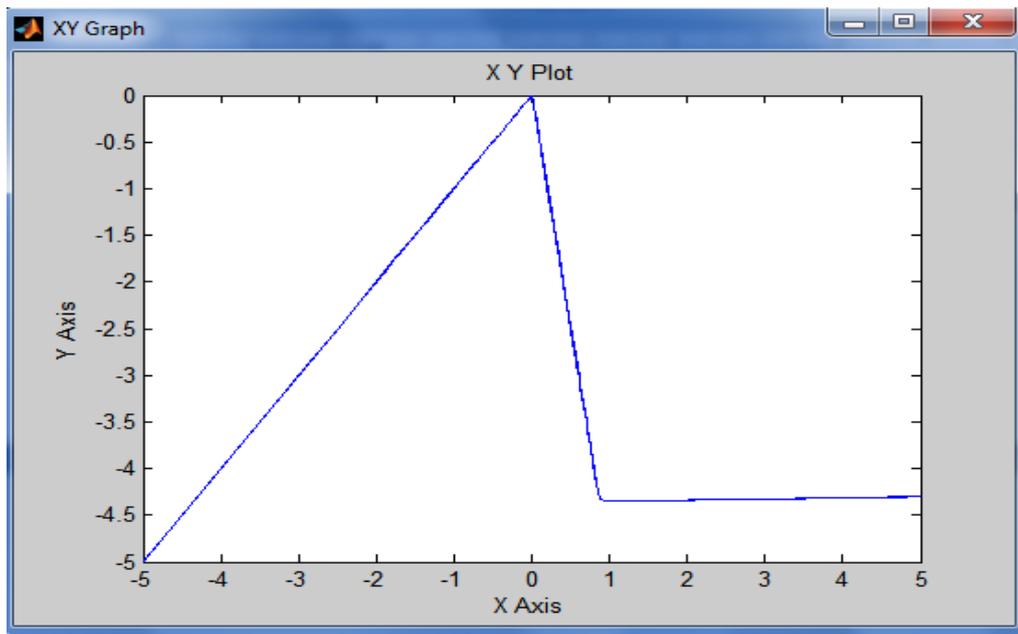


Figure 4.45 : Simulation de la faute BOQN10 par PSPICE [49]

**b) Simulation par SIMULINK :**



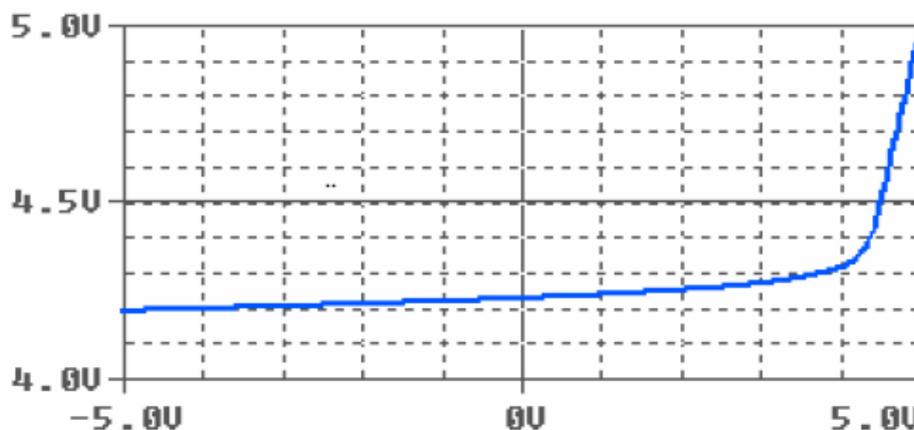
**Figure 4.46 : Simulation de la faute BOQN10 par Simulink**

On remarque que la réponse de l'Amplificateur inverseur sous la faute BOQN10 (Base ouvert de QN10) est identiquement enregistrée par SIMULINK et PSPICE.

**4.11.1.2 La faute de court circuit (CCBEQP2):**

**a) Simulation par PSPICE :**

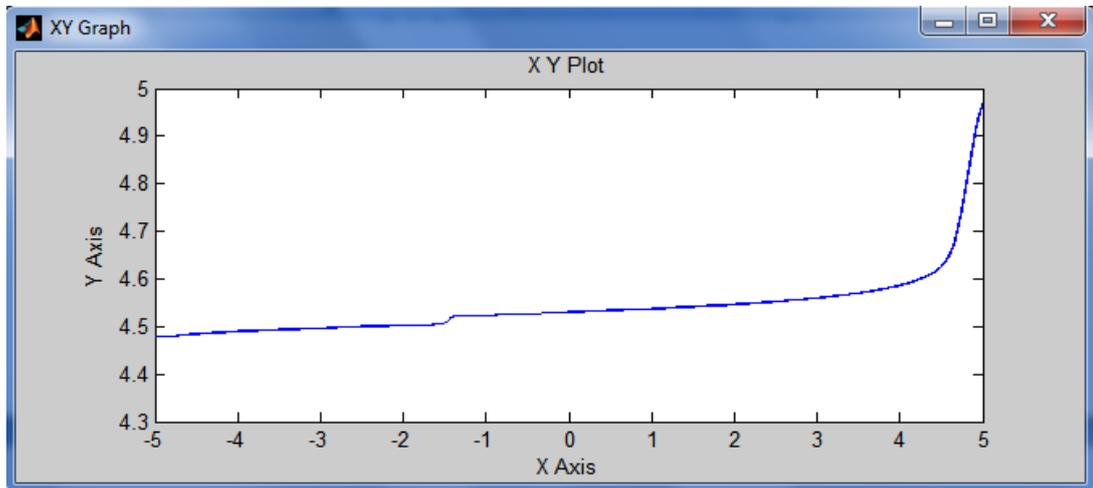
Selon toujours la même référence [49], le court circuit entre l'émetteur et la base du transistor QP2 a conduit à un effet donné par la réponse de sortie du circuit comme indiquée graphiquement selon PSPICE en figure suivante 4.47.



**Figure 4.47 : la faute CCBEQP2 sous PSPICE [14].**

### b) Simulation par SIMULINK :

Cette même faute CCBEQP2 est reprise sous l'environnement SIMULINK et a conduit au même résultat que le précédent dont la similitude est confirmée par le graphe de la figure 4.48.



**Figure 4.48 : la réponse en présence de la faute CCBEQP2**

Selon ces deux exemples de fautes, on remarque que le fonctionnement de l'Amplificateur inverseur est simulé correctement sous **PSPICE** ou le **SIMULINK**. De même avec l'insertion des fautes au niveau du circuit, les mêmes changements affectant la fonction de transfert ont été similairement enregistrés par ces mêmes simulateurs : Ce résultat valide notre bon choix sur l'emploi de SIMULINK dans notre processus de test. En conséquence, les résultats obtenus par ces 2 fautes seront traitées par la technique BIST pour établir leur signature binaire équivalente.

#### 4.11.2 Simulation par emploi de la technique BIST.

Dans cette technique de test numérique BIST, on doit connaître au préalable le vecteur binaire de test qui, une fois convertie en analogique, sera appliqué au circuit sous faute et propagera son effet à sa sortie. Ensuite un traitement numérique de celui-ci s'effectuera par l'unité de conversion analogique-numérique et enfin par l'ORA pour la génération de la signature.

Ce vecteur binaire est généré suite à une identification des bons stimuli à partir des courbes de transfert présentées dans les figures précédentes. Ces stimuli dont on donne une illustration correspondent aux signaux de tensions d'entrée et de sortie qui mettent en évidence la faute. Ces signaux après conversion en binaire serviront respectivement comme éléments d'excitation et de signature de faute.

#### 4.11.2.1 La première faute (BOQN10) :

Selon le graphe (figure 4.49) comportant la courbe de la fonction de transfert du circuit en présence de la faute BOQN10; il est à noter que pour une tension d'entrée  $V_e$  de -0.5 V la sortie du circuit livre une tension  $V_s$  de -0.5 au lieu de +2.48 V (cas de circuit sans faute). Ceci est mis en évidence à partir du graphe par le point d'intersection des lignes horizontale et verticale. Ces valeurs sont aussi affichées au niveau du scope branché à l'entrée et à la sortie du circuit sous test comme le montre la figure 4.50

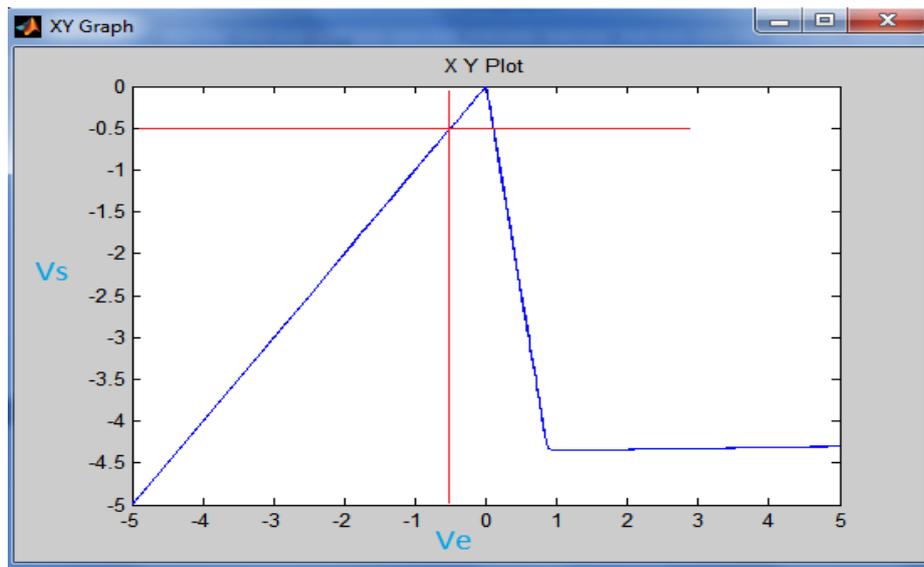


Figure 4.49 :  $V_s / V_e$  dans la fonction de transfert

On rappelle que le scope affiche l'entrée  $V_e$  et la sortie  $V_s$  comme fonction du temps. Il apparait selon la figure ci-dessous que leurs niveaux commutent de 0 à -0.5 V après un temps de 13 s qui représente l'instant correspondant à la 13<sup>ème</sup> séquence nécessaire à la génération du vecteur de test par la TPG.

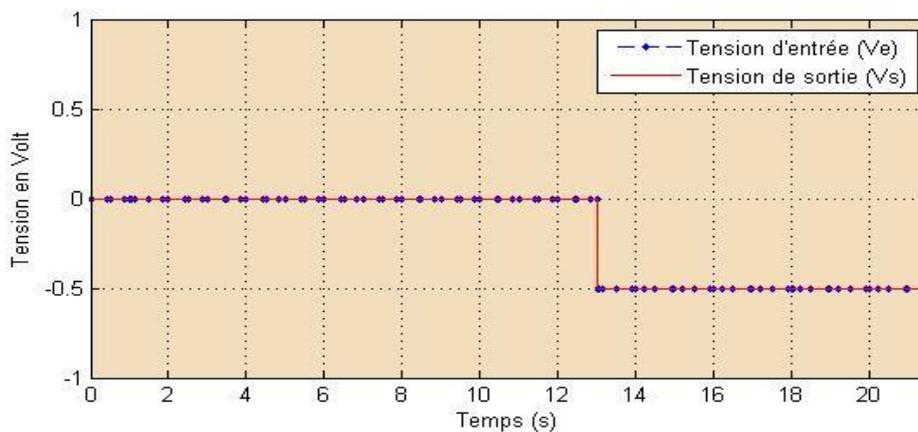


Figure 4.50:  $V_e$  et  $V_s$  en fonction du temps

Cette même valeur de  $V_s$  par SIMULINK sans l'emploi de la technique BIST confirme le bon fonctionnement de l'Amplificateur inverseur sous test.

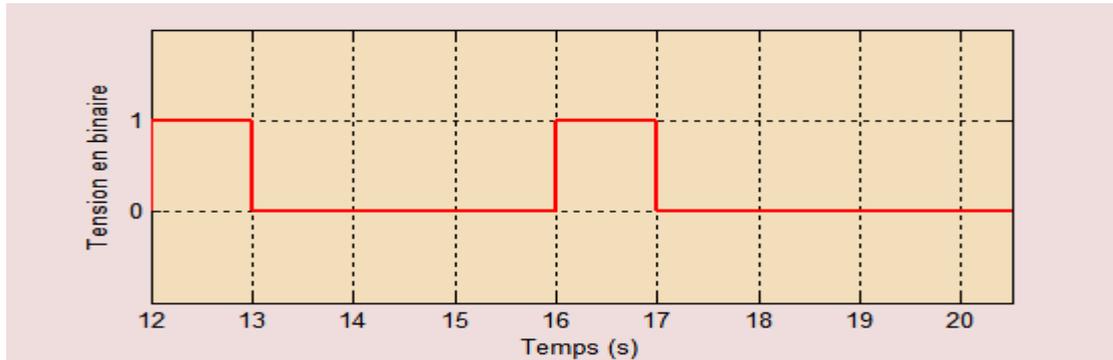


Figure 4.51 : la tension de Sortie  $V_s$  après la conversion analogique numérique

A partir de l'instant  $t = 13$  s, l'Amplificateur inverseur répond au signal d'entrée ayant pour valeur de  $-0.5V$ , et portera sa sortie à une tension  $V_s = -0.5V$ . Celle-ci transmise au convertisseur CAN subira une conversion à un signal binaire caractérisé par un mot de 8 bits où le MSB représente le signe :  $10001000 = K(x) = X^7 + X^3$ .

signe (-) 1 0 0 0 1 0 0 0  
 (0)                      0.5

Cette séquence binaire peut être traduite sous forme de polynôme  $K(x) = X^7 + X^3$

Par application de cette séquence binaire au niveau de L'ORA, on reçoit aux sorties de celle-ci des réponses sous forme d'impulsions visualisées au niveau du scope (voir figure 4.51). La signature correspond alors aux derniers états logiques présents à ces sorties à l'instant  $t$  de 21s

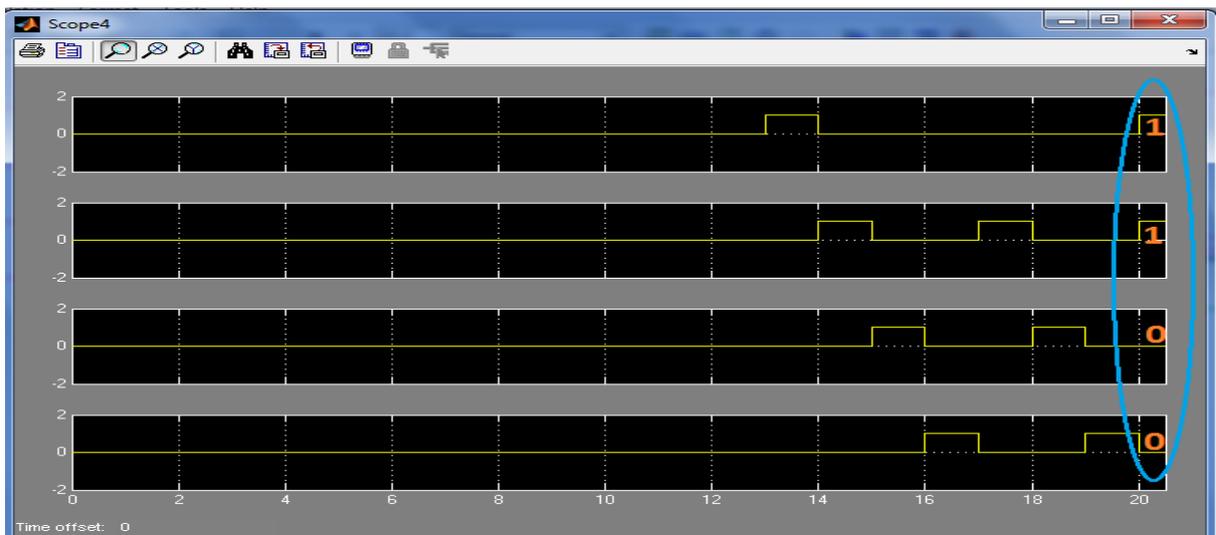


Figure 4.52: la signature a la sortie de l'ORA pour la faute BOQN10

La signature résultante de la faute BOQN10 est 0011, pour la comparer à celle théorique, on procède à la division polynomiale de  $K(x) = X^7 + X^3$  par le polynôme primitive  $P(x) = X^4 + X + 1$ , dont le reste qui représente la signature est  $R(x) = X + 1 = 0011$ .

#### 4.11.2.2 Le cas de la deuxième faute CCBEQP2 :

Similairement au 1<sup>er</sup> cas, l'examen de la courbe de la fonction de transfert du circuit sous cette 2<sup>ème</sup> faute (voir figure 4.52) montre, pour un signal d'entrée  $V_e = -0.5$  V, une sortie portée à une tension  $V_s = +4.53$  V. Ces valeurs sont indiquées clairement au niveau du graphe par le point d'intersection des lignes verticale et horizontale.

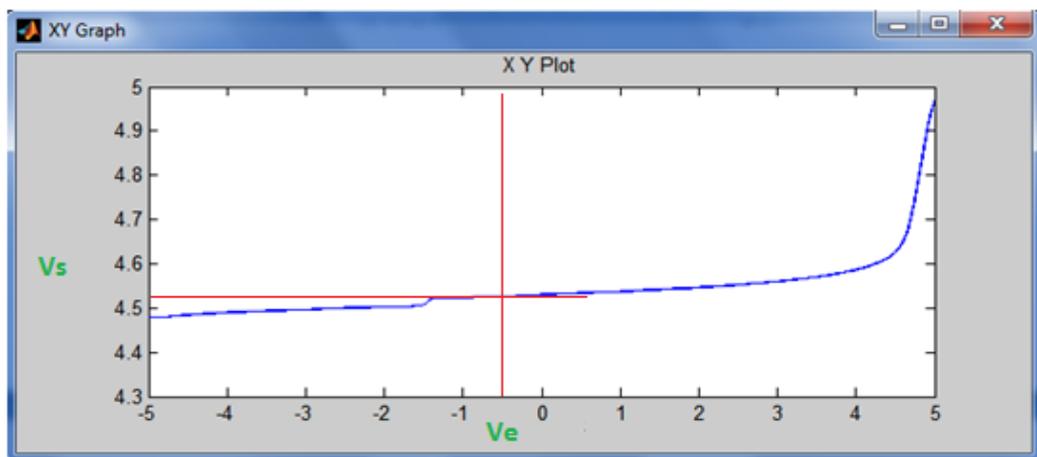


Figure 4.53 : Vs et Ve par la fonction de transfert

Le scope affiche également les mêmes valeurs de tensions des signaux d'entrée et de sortie tout en indiquant leur évolution dans le temps. Il apparaît d'une manière analogue au cas de la première faute, que le niveau du signal d'entrée bascule de 0 à -0.5V après un instant  $t$  de 13s correspondant bien sûr à la 13<sup>ème</sup> séquence de la génération du vecteur de test. Le signal de sortie reste par contre collé au niveau 4.53V quelque soit le niveau de tension à l'entrée (0 ou -0.5V).

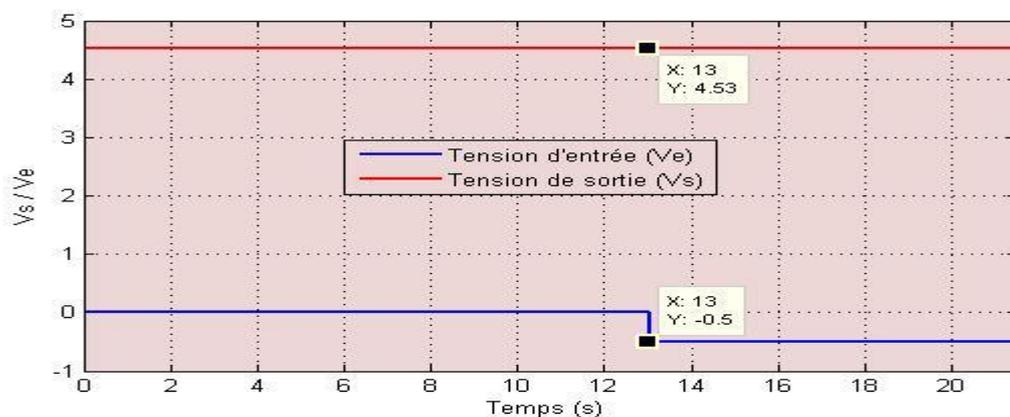
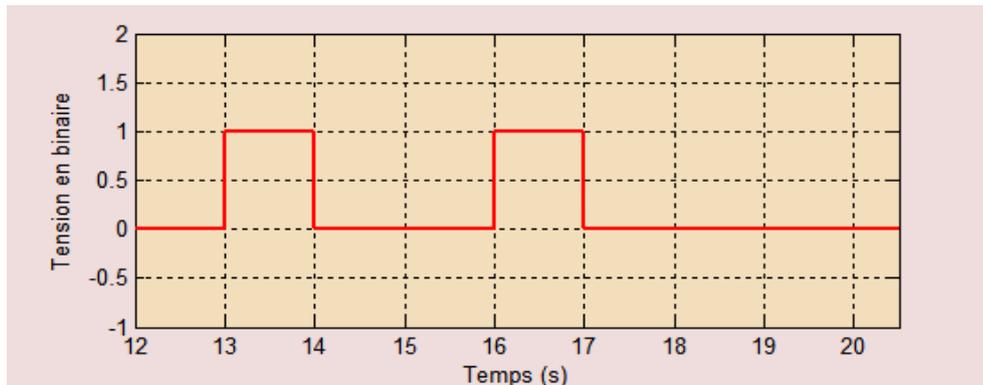


Figure 4.54 : Vs et Ve avec le Scope

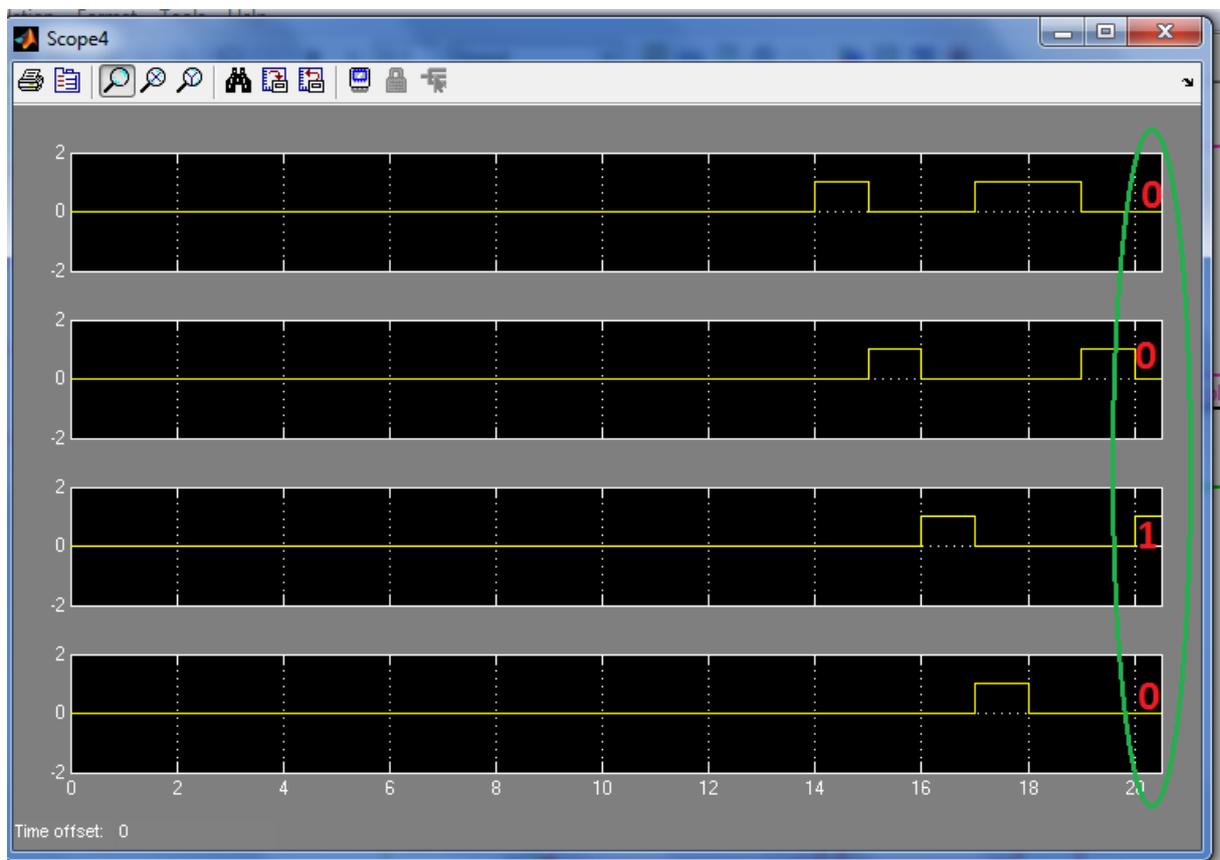
En phase d'analyse de la réponse du circuit fautif par l'ORA, ce signal de sortie  $V_s$  transmis vers le CAN sera converti en binaire dont le résultat est affiché par le scope en figure 4.55.



**Figure 4.55 : la tension  $V_s$  en binaire**

Cette tension de sortie  $V_s$  d'une valeur analogique de +4.5V a une valeur binaire de : **01001000** qui peut être écrite sous forme d'un polynôme  $K(x) = X^6 + X^3$ .

L'application de cette séquence binaire à l'ORA conduira cette dernière à livrer à ses sorties des signaux dont les chronogrammes sont présentés en figure 4.52.



**Figure 4.56 : la signature à la sortie de l'ORA pour la faute CCBEQP2**

La signature résultante de la faute (CCBEQP2) est une valeur binaire **(0100)** dont les bits correspondent aux derniers états logiques se trouvent aux sorties de l'ORA. L'établissement d'une comparaison entre cette signature et la signature théorique passe par le calcul de cette dernière à partir d'une division polynomiale de  $K(x) = X^6 + X^3$  par le polynôme primitive  $P(x) = X^4 + X + 1$ . Le reste de cette opération donnera comme reste  $R(x) = X^2$ , ce polynôme représente bien la valeur binaire **0100**.

On résume la signature pour le circuit saint et fautif dans les deux cas (court circuit et circuit ouvert) dans le tableau suivant :

L'Ampli La Signature	Ampli-Op saint	Ampli-Op avec faute N° 01 (BOQN10)	Ampli-Op avec faute N° 2 (CCBEQP2)
Polynôme R(x)	$X^3 + X^2 + X$	$X + 1$	$X^2$
Mots binaire	<b>1110</b>	<b>0011</b>	<b>0100</b>

Finalement, on fait remarquer que la signature (1110) est une référence, elle reste sauvegardée dans une mémoire. A chaque fois qu'on simule une faute différente, la signature obtenue serait comparée à cette signature mémorisée dont la différence observée signifie présence d'une faute (faute détectée). C'est ainsi que la technique BIST se déroule pour détecter les fautes.

#### 4.12 Conclusion

Dans cette partie de travail, on a pu présenter trois aspects de simulation différents à savoir la simulation électrique, la simulation numérique et la simulation basée sur l'emploi de fonctions mathématiques. Ces simulations sont exploitées simultanément dans un même environnement qui est le SIMULINK. L'objectif fixé est de parvenir à simuler un processus de test numérique s'appuyant sur la fameuse technique BIST et orienté vers les circuits analogiques. L'approche de test consiste à générer une signature du circuit sans faute qui serve de référence. Celle-ci est utilisée à titre comparatif avec celles produites dans 2 cas de circuits fautifs et le résultat de test est concluant.

# Conclusions et perspectives

## Conclusions :

L'approche de test utilisant la technique BIST pour la signature numérique en l'absence et présence de fautes d'un circuit analogique, dans notre cas un inverseur construit autour d'un amplificateur opérationnel, s'est avérée probante quant à l'extraction de la signature numérique. Au cours de ce travail nous nous sommes penchés principalement sur les blocs numériques tels que le TPG et ORA qui ont permis respectivement de générer les vecteurs de test et d'analyser la réponse numérique. Les convertisseurs numérique-analogique et analogique-numérique à 4 bits ont prouvé leur efficacité en terme de précision et fiabilité du test puisque les signatures se révèlent assez pertinentes en les comparant avec les résultats issus des travaux antérieurs.

L'outil de simulation Matlab/Simulink était d'une grande utilité pour la synthèse et utilisation de la technique BIST mettant en œuvre deux environnements numériques et analogiques dans un même processus de test. En effet cette structure a fait préalablement l'objet d'une simulation d'un circuit purement numérique aboutissant à des résultats très intéressants. Ceci nous a conduits à poursuivre le processus pour le circuit mixte associant la partie analogique (Le circuit sous test) et les blocs numériques impliqués dans le processus de la technique BIST.

Ainsi, deux cas de circuits fautifs ont été considérés : il s'agit d'un court circuit et d'un circuit ouvert insérés d'une façon séparée au niveau des transistors composant ce circuit analogique. Les réponses récoltées dans ces 2 cas de figure de circuit ont subi un même traitement que précédemment décrit par la structure de test afin de produire la signature correspondante. Enfin leur comparaison avec la signature référence confirme la présence ou l'absence de faute. Le test a été concluant pour ces deux fautes investiguées.

## Perspectives :

La technique BIST a prouvé son efficacité à tester un circuit analogique. Ceci nous a tracé le chemin dans l'espoir de continuer ce processus de test pour pouvoir couvrir l'ensemble de fautes avec le minimum de vecteurs de test et en un temps relativement petit. D'autres perspectives sont à considérer qui peuvent s'inscrire dans l'emploi de moyens de traitement de données de test, et de gestions automatisées des différentes unités de test.

L'emploi d'un microcontrôleur est à prévoir pour générer tous les signaux de commandes avec un espace mémoires servant à la sauvegarde des signatures.

## Références

- [1] Ahcéne BOUNCEUR, « Plateforme CAO pour le Test de Circuit Mixte » thèse de Doctorat Avril 2007 à l'INPG.
- [2] L.S. Milor, A Tutorial Introduction to Research on Analog and Mixed-Signal Circuit Testing. IEEE Transactions on Circuit and Systems II: Analog and Digital Signal Processing, vol. 45, issue 10, October 1998, pp. 1389-1407
- [3] A. Khouas. « Simulation de Fautes et Optimisation des Tests de Production pour les Circuits Analogiques avec Prise en Compte des Tolérances ». Thèse de doctorat, Université de Paris VI, 2000.
- [4] Asquini Anna , « technique de test pour synthétiseur de fréquence » .thèse de doctorat Janvier 2010 au Grenoble.
- [5] S. Sunter and A. Roy, "Purely Digital BIST for any PLL or DLL". 12th European Test Symposium, May 2007, pp. 185-192
- [6] N. Bourouba , " Développement des techniques de test pour des circuits analogiques de type SSI." Thèse de Doctorat, Novembre 2007
- [7] Abdelhakim Khouas " Simulation de faute et optimisation des test de production pour les circuits analogiques avec prise en compte des tolérances" thèse de doctorat, septembre 2000.
- [8] H. Albustani, " Modeling methods for testability Analysis of Analog Integrated Circuits Based on Pole-Zero analysis, " Thèse de Doctor Ing. August 2004
- [9] A.K.Stevens, "Introduction to Component Testing Addison –Wesley, Reading, MA, 1986.
- [10] Nur Engin, "Linking Mixed –Signal Design and test : Generation and Evaluation of Specification Based Tests, " University of Twente in Enschede, The Netherlands, 2000.
- [11] N. Engin. "Linking Mixed-Signal Design and Test : Generation and Evaluation of Specification-Based Tests". University of Twente in Enschede, the Netherland, 2000.
- [12] L. Milor. "Fault-driven analog testing. Ph.d. dissertation" University of California, Berkeley, Calif., 1992.
- [13] Semiconductor Industry Association, International Technology Roadmap for Semiconductors: 1999 Edition, Austin, TX: SEMATECH, 1999.
- [14] Charles E. Stroud , "A Designer's Guide to Built In Self Test" University of North Carolina at Charlotte
- [15] N. G. Einspruch, " VLSI Handbook, academic Press Orlando, FL. 1985.

- [16] Duhamel, J. C. Rault, "Automatic Test Generation techniques for Analog circuits and Systems: A Review", Transactions on Circuits and Systems, Vol. CAS-26, N°7, July 1979.
- [17] V. Agrawal and S. Seth, Tutorial: Test Generation for VLSI Chips, Los Alamitos, California, IEEE Computer Society Press, 1988.
- [18] E. McCluskey, "Logic Design Principles with Emphasis on Testable Semicustom Circuits", Upper Saddle River, New Jersey: Prentice Hall, 1986.
- [19] E. Rudnick, J. Patel and I. Pomeranz, "On Potential Fault Detection in Sequential Circuits," Proc. IEEE International Test Conf., 1996, pp. 142-149.
- [20] C. Benoît. "Modélisation de fautes de conception en vue de test structurel de microsysteme". Thèse de Doctorat, 12 Mars 2001, Institut polytechnique de Grenoble.
- [21] J. Grason and A.W. Nagle "Digital test generation and design for testability," 17th Design Automation Conference. Proceeding 175-189 (1980).
- [22] H.- J. Wunderlich "BIST for systems-on-a-chip" INTEGRATION, the VLSI journal 26 (1998) 55-78
- [23] S. Hellebrand, B. Reeb, S. Tarnick, "Pattern Generation for a Deterministic BIST Scheme "ACM/IEEE International Conference on CAD 95 (ICCAD-95), San Jose, Ca., November 1995
- [24] J. SMITH "An Automated BIST Architecture for Testing and Diagnosing FPGA Interconnect Faults " JOURNAL OF ELECTRONIC TESTING: Theory and Applications, 22, 239-253, 2006
- [25] P. Fišer , H. Kubátová "Column-matching based mixed-mode test pattern generator design technique for BIST " Microprocessors and Microsystems 32 (2008) 340-350.
- [26] P. Bardell, W. McAnney and J. Savir, "Built-In Self-Test for VLSI: Pseudorandom Sequences, Somerset, New Jersey: John Wiley & Sons, 1987.
- [27] S. Golomb, Shift Register Sequences, Launa Hills, California: Aegean Park Press, 1982.
- [28] B. Johnson, "Design and Analysis of Fault Tolerant Digital Systems", Reading, Massachusetts: Addison-Wesley, 1988.
- [29] R. Ball, "testing electronic Circuits that contain both analogue and digital Elements" M. Phil thesis, January 1986.
- [30] B.R. Wilking "Testing Digital circuits. An Introduction" Ed Van Nostrand Reinhold, UK,; Chap.2, pp 45-49, 1986
- [31] M. Catelani, M. Gori, "On the Application of Neural Networks to Fault Diagnosis of Electronic Analog circuits, " Measurements, Vol. 17, N°2, pp73-80, 1996.

- [32] B.P. Richards , P.H. Footner, "Failure analysis in semiconductor device –rational, methodology and practice " GEC Journal Vol. 1,n°2 pp74-90,1983
- [33] J.L. Huertas, "Test and Design for Testability of Analog and Mixed -signal Integrated Circuits. Theoretical Basis and Pragmatically Approaches." M Kunt, Elsevier Science Publishers, 1993.
- [34] J. Partridge, " Testing for Bipolar Integrated Circuits failures modes," The Charles Stark Draper Laboratory, IEEE Test Conference, pp397-406,1980.
- [35] C.H. Jone, " Fault Investigation of some silicon Integrated Circuits " ,The radio and Electronic Engineer, Vol. 42 ,n°4, pp 185-194. April 1972.
- [36] S. Liu and K. Singhal, " A Statistical Model for MOS FET." In Proc. IEEE. Int. Conference on Computer-Aided Design, pp 78-80- 1985
- [37] L. Milor and V. VisVanathan," Detection on catastrophic faults in Analog Integrated Circuits. " IEEE Trans. Computer-Aided Design , 114-130 Feb 89.
- [38] L. Milor and A.L. Sangiovanni-Vincentelli,"Minimizing Production Test Time to Detect Faults in Analog Circuits, " IEEE Trans. On Computer –Aided Design of Tntegrated Circuits and Systems 13(6)-1994.
- [39] M.W. Lindermeir, Helmut E. Graeb and Kurt L. Antreich, "Analog testing by characteristic observation inference." IEEE. Trans. Computer –Aided Design 18(9) :1368-1368-September 1999.
- [40] D.G. Damper, "Fault Models and Failure Mechanisms" Paper presented at the IEE vacation School on IC testing , Brunel University 1984.
- [41] E.A. Amerasekera, D.S. COMPBELL," Failure mechanisms in semiconductor devices " John Willey & Sons , 1987, ISBN 0471914347
- [42] Jonathan R. Carter, Sule Osev, and Daniel J. Sorin," Circuit-Level Modeling for Concurrent Testing of Operational Défects due to Gate Oxide Breakdown," IEEE Computer Society, Proceedings of the Design , Automation and Test in Europe Conference and Exhibition ,DATE'05, 2005.
- [43] C.-Y. Pan, K. T. Cheng, S. Gupta "A Comprehensive Fault macromodel for Opamps" ITC 1994, pp 344-348
- [44] H. HAO, and E.J.Mc.Cluskey, " Resistive Shorts within CMOS circuits," in Proc.1991, Int; Test Conf. Nashville, T.N, OCT 26-30, 1991 pp292-301.
- [45] W. Maly,P.K. Nag, P. Nigh," Testing Oriented Analysis of CMOS ICs with opens," in Dig. Int'l. Conf. Computer-Aided design, Santa clara CA; Nov-7-1à, 1988, pp 334-347.
- [46] N. Bourouba," Fault macromodelling and fault collapsing of analogue Integrated Circuits," M.phil thesis, University of Lancaster, December 1989.

- [47] E. Berger, E. Lindbloom, "Random –Pattern coverage Enhancement and diagnosis for LSSD Self-Test. " IBM Journal Research Development, pp265-271 Vol. 27n°3 May 1983.
- [48] J. Damianos "Testing Hybrid Circuits Using Digital Techniques," PhD Thesis, Southampton University, 1986.
- [49] A. Abderrezak "Développement de modèles électriques des défauts physiques d'un amplificateur inverseur à partir des caractéristiques électriques en vue de simulation" thèse de Magister a Ferhat Abbas, Setif , 2007.
- [50] W. Hochwald, J.D. Bastian, " A dc Approach for analog fault d dictionary determination", IEEE trans. Circuits and Systems, CAS -26, 523-529,1979.
- [51] A. Lakhlef,"génération automatique de vecteur de test et simulateur logique »- Thèse de magister, Université Ferhat Abbas de Sétif, 1994.
- [52] Szygenda, S.A. and Thompson , E.W." Digital Logic Simulation in a time –based table driven environment: Part II parallel fault Simulation"; Trans. Computer C 8,39-49,1975
- [53] D.B. Armstrong, "A deductive method for simulating faults in large circuits." IEEE Trans. Computer, C21, 469-471, 1972.
- [54] E.G. Ulrich, E.T. Baker, "Concurrent simulation of nearly identical digital networks," Trans. Computer 39-44, April 1974.
- [55] M. Zwolinski, A .D. Brown, , C.D. Chalk," Concurrent Analog Fault Simulation . "In Proc. IEEE International Mixed-Signal Testing Workshop, 1997.
- [56] C. Sebeke, J.P. Teixeira, M.J. Ohletz," Automatic Fault extraction and simulation of layout Realistic Faults of Integrated Circuits," Proc. European Design and Test Conference(EDTC) ,pp604-608,March 1996.
- [57] L. Michael, Bushnell, D. Vishwani, D. Agrawal," Essential of Electronic Testing for digital memory &Mixed signal VLSI circuits," Kluwer Academic Publishers,2000.
- [58] Z.R. Yang, Zwoollinsky, "Fast , Robust DC and Transient Fault Simulation For nonlinear Analog Circuits," DATE, pp244- 248.
- [59]N. Benhamida, K. Saab, D. Marche, R. Kaminska, "A perturbation based fault modeling and simulation for mixed –signal Circuits" in Proc. IEEE Asian Test Symposium pp182-187, 1997.
- [60] N. Engin, Hans. G. Kerkhoff,"Fast-fault simulation for analog circuits "IEEE Design & Test of Computers, March –April 2003

- [61] B. Straube, W. Vermeiren, H. Albustani, V. Spenke, "Multi-level Hierarchical Analogue Fault simulation with a FSIM," IEEE International Mixed signal Testing workshop, pp174\_180, 2000.
- [62] A.J. Perkins, M. Zwolinski, C.D. Chalk, B.R Wilkins, "Fault modeling and simulation using VHDL-AMS" analog Integrated Circuits and signal Processing 16, 141-155, 1998
- [63] N. Nagi, Jacob A Abraham, "Fault simulation of linear Analog circuits," Journal of Electronic Testing: Theory and Application, 4, 345-360, 1993.
- [64] P.N. Variyam, A. Chatterjee, "FLYER: "Fast Fault simulation of Linear Analog Circuits Using Polynomial waveform and perturbed State representation," Int. Conference ON VLSI Design, pp408-412, Jan 1997.
- [65] C.I. Richard Shi, Nihal L. Godambe, "Behavioral Modeling and Simulation of phase-locked loops using VHDL –A like language. In Proc. IEEE International ASIC Conference and Exhibit 1997.
- [66] A.L. Perkins, M. Zwolinski, C.D. Chalk and B.R. Wilkins, "Fault modeling and simulation using VHDL-AMS", analog Integrated Circuits and signal Processing, Kluwer Academic Publishers 16, 53-67, 1998.
- [67] F. Lémery, « Modélisation comportementale des circuits analogiques et mixtes » thèse de Doctorat, institut polytechnique de Grenoble, Décembre 1995.
- [68] A. Vachoux, « Modélisation des systèmes analogiques et mixtes, introduction à VHDLAMS » Laboratoire de systèmes microélectroniques, version 2003.
- [69] S. Belkacem, « Macro modélisation comportementale de circuits analogiques : application au circuit convoyeur de courant » thèse de Magister, université de Batna, 2005.
- [70] B. Hahn, D. valentine "Essential MATLAB for Engineers and Scientists" third edition 2007.
- [71] Damian G, Jerzy R. "fault diagnosis in analog electronic circuits – the SVM approach" Metrol. Meas. Syst. Vol. XVI (2009), No 4, pp. 583-598.
- [72] Zbigniew C. « A simple fault diagnosis Method for analog parts on electronic embedded systems " XIX IMEKO World Congress Fundamental and Applied Metrology September 6-11, 2009, Lisbon, Portugal
- [73] Wimol S. Tachibana M. " An On-Chip Analog Mixed-Signal Testing Compliant with IEEE 1149.4 Standard Using Fault Signature Characterization Technique " Kochi University of Technology 2009 N° 782-8502,
- [74] Manuel Ferranti Interdesign INC. 201 Analog IC designs, 1985.

## موجز :

تقنيات الاختبار الرقمية أدت إلى نقلة نوعية في كفاءة الاختبار الذي يتميز بالبساطة والسرعة. ومن بين هذه التقنيات تقنية BIST المشهور ، التي تعتمد على إدراج دائرة للاختبار ضمن الدائرة الأصلية مما يجعل هذه الأخيرة قابلة للاختبار الذاتي على العكس من الدارات التناظرية، أين يكون الاختبار أكثر تعقيدا ويأخذ وقتا أطول للأسف، هذا ما جعل المهمة صعبة أمام الباحثين إذ يحاولون تجاوز هذا المشكل .

الهدف من عملنا هذا هو اختبار دائرة تناظرية بتقنية BIST. إذ نركز على اختبار هذه الدائرة بتلك التقنية مع الحفاظ على هيكلها الذي يستخدم مولدا لسلسلة إشارات ، ثم يحللها ويقارنها مع النتيجة الصحيحة في حالة وجود أخطاء في الدارة أو عدم وجودها . الدائرة تحت الاختبار هي عبارة عن دائرة 741 تستعمل لتضخيم الإشارة. نوعية الأخطاء المُدرجة في دراستنا هي أخطاء صعبة أو تسمى أخطاء كارثية وهي التي تكون شائعة في الترانزستورات ثنائية القطب من نوع CMOS ، تطرقنا في بحثنا إلى دراسة تأثير هذا النوع من الأخطاء في دارة تناظرية مستعملين البرنامج المحاكي SIMULINK لتحليل البيانات ودراستها .

كلمة المفتاح : تقنية BIST ، الدارة التناظرية ، عجز الدارة الالكترونية ، ترانزستور ثنائي القطب .

## Summary:

In the digital world, the test techniques led to an eminent jump regarding to the test process quality which is characterized by simplicity, efficiency and speed. Among these techniques we quote the famous BIST that has been even implemented with the circuit to be tested so making it self-tested. On the contrary, the test in analog world is more complex due to its nature and more diversified with its test parameters. Unfortunately this task is delicate and time consumer. A very major concern that dominates in mind the researchers is to solve these problems.

The objective fixed in our work is to test an Analog circuit by the technique BIST. The importance given here at the first stage is to pass this circuit test successfully while protecting its basic structure which uses the generation of test vectors. Then the signature analysis and decision-making have to be settled down on the presence and the absence of faults. The circuit under test is one of common use namely the inverter amplifier designed with a 741 type operational amplifier. In this study, the type of faults taken into account are catastrophic ones resulting from frequent physical damages to the CMOS and bipolar transistors. Our circuit subjected to the presence of this kind of failure is simulated by means of the SIMULINK to establish an analysis data collection from the faulty circuit responses.

**Key words:** BIST technique, analog circuit, failure, bipolar transistor