

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOU MAMMERI DE TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE  
DEPARTEMENT D'ELECTRONIQUE

# Test des circuits intégrés

Cours pour master M1 académique

Spécialité : Electronique

Option : Microélectronique

Présenté par :

M<sup>me</sup> Ghania AIT ABDELMALEK

Spécialité : Microélectronique

Introduction générale .....	1
-----------------------------	---

## **Chapitre 1 : Notions de base du test des circuits intégrés**

1.1-Introduction.....	3
1.2-Terminologie et définitions générales .....	3
1.3- Définition du test.....	5
1.4- Les différents types de test.....	5
1.4.1- Test fonctionnel.....	5
1.4.2- Test de production.....	6
1.5- Le test industriel dans les phases de fabrication de circuits intégrés.....	7
1.6-Les différentes phases de test.....	7
1.7- Motivation du test.....	9
1.8- Techniques de test utilisées.....	10
1.9- Les aspects économiques du test de production.....	11
1.9.1 Coût du test de production.....	11
1.9.2 Rendement de fabrication.....	11
1.9.3 Niveau de qualité d'un ensemble de test.....	12
1.9.4- La couverture de fautes.....	13

## **Chapitre 2 : Test industriel et Équipements de Test « ATE »**

2.1- Introduction.....	14
2.2- Conception des circuits VLSI.....	14
2.2.1- La modélisation du système.....	15
2.2.1.1- Description comportementale.....	15
2.2.1.2- Description RTL.....	16
2.2.1.3- Description au niveau porte.....	16
2.2.1.4- Description au niveau transistor.....	16
2.2.1.5- Description au niveau layout.....	17
2.2.2- la vérification du design.....	17
2.2.2.1- La vérification dynamique.....	17
2.2.2.2- La vérification statique.....	17
2.2.2.3- La vérification physique du dessin de masques.....	18
2. 2.3- La synthèse.....	18
2.3- Test industriel des circuits VLSI.....	18
2.3.1- Différentes phases de test.....	18
2.3.2- Test de production.....	19
2.3.3- Test pire-cas.....	19
2.3.3.1- Dessin « Shmoo ».....	20

2.3.3.2- La data sheet.....	20
2.4- Paramètres de test.....	20
2.5- Inverseur CMOS.....	21
2.5.1- Comportement statique (ou DC).....	21
2.5.2- Comportement dynamique.....	26
2.5.3- Temps de stabilisation (setup time, $t_{SU}$ ) et de temps de maintien (hold time, $t_H$ )....	29
2.6- Plan de test.....	29
2. 7- Les testeurs industriels et équipements de test.....	29
2.7.1- Testeurs de production.....	30
2.7.2- Testeurs DFT.....	30
2.7.3- Équipements d'un testeur ATE.....	31
2.8 Analyse des données de test.....	31

### **Chapitre 3 : Modélisation de défauts**

3.1- Introduction.....	32
3.2-Défauts et fautes.....	33
3.3- Types de défauts physiques « fabrication ».....	33
3.3.1 Les défauts catastrophiques.....	33
3.3.1.1-Défaut de court-circuit.....	33
3.3.1.2- Défaut de circuit-ouvert.....	34
3.3.1.3-Défauts paramétriques.....	35
3.3.1.4-Défauts permanents.....	35
3.3.1.5-Défauts temporaires.....	35
3.4- Sources de défauts.....	35
3.5- Classification et répartition de défauts.....	36
3.6-Modèles de fautes « modèles de pannes ».....	37
3.6.1-Le modèle de collage.....	37
3.6.2- Modèle des collages multiples.....	38
3.6.3 -Modèle des courts- circuits « Bridging Faults ».....	39
3.6.4 - Modèle des circuits-ouverts « Stuck Open Faults ».....	39
3.6.5 Modèle de fautes de délai.....	40
3.6.5.1-Modèle des fautes de transition.....	40
3.6.5.2-Modèle de fautes de retard de porte.....	41
3.6.5.3-Modèle de fautes de délai de chemin.....	41
3.7-Détection de fautes.....	41

### **Chapitre 4 : Simulation de fautes**

4.1- Introduction.....	42
4.2 - Simulation de fautes.....	42

4.3- Méthodologie de la simulation de fautes.....	43
4.4- Algorithmes de simulation de fautes.....	44
4.4.1-Simulation de fautes série.....	44
4.4.2-Simulation de fautes parallèle.....	44
4.4.3-Simulation de fautes déductive « DFS ».....	46
4.4.4-Simulation de fautes concurrente.....	47

## **Chapitre 5 : Génération Automatique des Vecteurs de Test (ATPG)**

5.1- Introduction.....	48
5.2- La génération de vecteurs de test.....	48
5.3- Les types de génération de vecteurs de test.....	49
5.3.1- La génération manuelle des vecteurs de test.....	49
5.3.2- La génération pseudo aléatoire des vecteurs de test.....	49
5.3.3- Génération exhaustive des vecteurs de test.....	49
5.3.4- Génération automatique ATPG « Automatic Test Pattern Generation ».....	49
5.4- Détection de fautes.....	50
5.5- algorithmes de génération automatique.....	51
5.5.1- Algorithme D.....	51
5.5.1.1-Phase de « setup ».....	51
5.5.1.2-Phase de propagation.....	52
5.5.1.3-Phase de justification.....	52
5.5.2- Algorithme PODEM.....	53
5.5.3- Algorithme FAN.....	55
Conclusion générale.....	56
Références bibliographiques.....	57

# **Introduction générale**

Ce cours est une unité découverte destiné aux étudiants de Master 1 de spécialité « microélectronique ». Son volume horaire est de 1h30 de cours hebdomadaire durant le 2<sup>ème</sup> semestre. Le but de ce cours est d'initier les étudiants aux différentes techniques et outils de test des circuits intégrés.

Plusieurs domaines sont devenus très importants dans la vie quotidienne ces dernières années, tels que la télécommunication, le multimédia, les applications biomédicales, etc. Ces domaines nécessitent une utilisation importante des circuits intégrés. Avec l'arrivée des technologies fortement submicroniques, certains phénomènes physiques qui auparavant étaient négligeables deviennent prépondérants. Ces phénomènes pouvant induire des défaillances ont une répercussion directe sur la fiabilité et le rendement de fabrication. Tester ces circuits cause un impact considérable sur le coût de production.

Le test d'un circuit intégré consiste à mettre en évidence un mauvais fonctionnement éventuel dû à une défaillance physique. Compte tenu du grand nombre de défaillances physiques possibles dans un circuit, il est impossible de développer des outils rapides et efficaces capables de manipuler directement ces défaillances. Des modèles de fautes ont donc été développés pour représenter les défaillances susceptibles de se manifester dans un circuit, et la plupart des techniques de test s'appuient sur ces modèles de fautes.

Ce cours est composé de 5 chapitres. Dans le premier chapitre, on traite, après un rappel sur les définitions générales indispensables pour la compréhension du domaine de test des circuits intégrés, le test industriel dans les phases de fabrication et sa caractérisation, la motivation du test ainsi que les techniques de test utilisées et enfin les aspects économiques du test de production.

Le deuxième chapitre : Test industriel et Équipements de Test « ATE » est consacré aux définitions et propriétés du test industriel et ses équipements de test utilisés. La théorie est expliquée et une application essentielle, les paramètres de test, est présentée et commentée à l'aide d'un exemple d'un circuit intégré de base à savoir l'inverseur CMOS. Enfin, certaines applications utiles dans le test des circuits intégrés, telles que le plan de test et l'analyse des données de test sont étudiées. Cette analyse se fait à l'aide des outils statistiques qui sont très utiles dans le test des circuits où ils peuvent servir à la vérification de la qualité du circuit sous test et l'amélioration du rendement de fabrication.

Le troisième chapitre : modélisation de défauts, présente les différents types de défaut rencontrés dans les circuits intégrés. Les sources de ces défauts sont définies. Les modèles de fautes représentatifs des défauts réels sont détaillés en vue de développer une stratégie de test efficace en termes d'outils de simulation de fautes, de génération de vecteurs de test, ou de diagnostic. Enfin pour conclure ce chapitre, un exemple de détection de fautes est étudié.

Le quatrième chapitre : simulation de fautes, traite la méthodologie de la simulation de fautes et ses différents algorithmes utilisés. Il évoque avec détail quatre aspects fondamentaux

rencontrés en algorithmes de simulation de fautes : la simulation de fautes série, la simulation de fautes parallèle, la simulation de fautes déductive et la simulation de fautes concurrente.

Le cinquième chapitre : Génération Automatique des Vecteurs de Test (ATPG), clôt ce cours. Il représente le moyen permettant de passer du domaine théorique au domaine pratique et où la forme de base des vecteurs de test, discrète, est obtenue par l'application des algorithmes de génération de vecteurs automatique. Des algorithmes D, PODEM et FAN, sont décrits et leur grande utilité en tant que traitement de base est précisée.

# **Chapitre 1. Notions de base du test des circuits intégrés**

## 1.1- Introduction

Un circuit intégré est dit performant s'il est apte à réaliser sa fonction voulue avec un niveau de qualité et de fiabilité des plus élevés. Pour garantir cette qualité de bon fonctionnement, il faut tester et vérifier le circuit dès les premières étapes de la conception.

Ce chapitre a pour objectif de donner les notions de base du test des circuits VLSI, sans chercher à prétendre à l'exhaustivité.

## 1.2-Terminologie et définitions générales

Il existe quelques termes qui doivent être définis avec précision pour éviter une fausse interprétation pour la théorie et l'application du test des circuits intégrés. Une description de ces termes est donnée dans ce qui suit.

- Performances : elles décrivent le fonctionnement du circuit. Connaître les performances d'un circuit permettra de décider s'il est fonctionnel ou bien défaillant.
- Spécifications : ce sont l'ensemble des valeurs acceptables de chaque performance. Chaque performance possède deux spécifications (la borne inférieure et la borne supérieure).
- Mesures de test ou observations : ce sont des mesures qui sont effectuées sur le circuit.
- Critères de test : 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 : mauvaise construction au niveau d'une des parties du circuit qui le rend défaillant.
- Défaillance ou panne : l'effet d'un défaut.
- Faute : représentation des défauts au niveau logique selon un modèle de fautes.
- Erreur : fonctionnement anormal produit par une faute.
- Détection : procédure permettant de confirmer l'existence d'une faute dans un circuit.
- Localisation : elle s'effectue une fois la faute 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).
- Vecteur de test : c'est une série de pulsations. Chaque pulsation  $i$  est caractérisée par sa largeur  $PW_i$  et son temps d'échantillonnage  $t_i$ .
- ATPG : (Automatic Test Pattern Generation) : génération automatique de vecteurs de test
- DFT : (Design For Test) : Conception en vue du test
- Circuits VLSI (Very Large Scale Integration): circuits à très grande échelle d'intégration.
- 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 process : 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.
- Test fonctionnel : vérification des spécifications du circuit.
- Test structurel : il permet de détecter les défauts de fabrication qui peuvent affecter le fonctionnement des circuits.
- Test paramétrique : il est utilisé pour vérifier si certains paramètres du circuit sous test sont dans la plage des valeurs requises.
- Test alternatif : test qui permet de calculer (ou prédire) les performances du circuit à partir de tests plus simples.
- Test statique : vérification des états stationnaires ou logique du système sous test.
- Test dynamique : vérification des caractéristiques dynamiques du système sous test.
- Test exhaustif : vérification de tous les modes d'opération pour tous les types de fautes.
- Test on-line : (test en-ligne) le test est effectué pendant le fonctionnement normal du circuit.
- Test off-line : (test hors-ligne) le test est effectué en mode test.
- Test concurrent : le test est effectué pendant l'opération.
- Test semi-concurrent : le test est effectué pendant l'opération mais pendant le temps de repos.
- Test non-concurrent : le test est effectué hors-ligne.
- Test wafer : le test est effectué au niveau plaquettes (Wafer) pendant la fabrication des circuits intégrés.
- Test de production : test qui se fait durant la phase de production des circuits intégrés.
- Test sur-puce (on-chip) : un test qui s'effectue dans la puce et à l'aide d'un ensemble de circuits qui se trouvent à l'intérieur de la même puce.
- Test in-field : un test qui s'effectue une fois la puce et dans son application finale.

### **1.3- Définition du test :**

C'est le processus qui permet de déterminer si le circuit est sain ou défectueux. Le test des circuits intégrés se fait en trois étapes principales : la détection, la localisation et le diagnostic. Si le circuit est défectueux, c'est soit parce que le circuit conçu ne répond pas aux spécifications du cahier des charges et par conséquent il contient une ou plusieurs erreurs de conception et nécessite donc un test fonctionnel, soit à cause d'un ou de plusieurs défauts qui affectent le circuit fabriqué, ce qui le rend incompatible au circuit conçu, d'où la nécessité du test structurel.

### **1.4- Les différents types de test**

Le test tente de répondre aux trois questions suivantes :

- Est-ce que le circuit fonctionne ?
- Est-ce que le circuit cache un problème important ?
- Le circuit a-t-il sa capacité complète ?

Il existe trois types de test. Le premier type représente le test fonctionnel, puisqu'il est destiné à vérifier que le circuit sous test possède les caractéristiques fonctionnelles attendues. Le deuxième type représente le test structurel, qui est destiné à vérifier s'il y a présence d'une faute dans le circuit sous test. Ce type de test peut être facile comme il peut être difficile, car il dépend du but recherché, détection ou localisation de la faute. Enfin, le troisième type de test est le test paramétrique. Celui-ci est utilisé pour vérifier si certains paramètres du circuit sont dans la plage des valeurs requises.

#### **1.4.1- Test fonctionnel**

Comme le montre la figure 1, le but d'un test fonctionnel est de vérifier toutes les fonctionnalités du circuit vu comme une boîte noire, avant de l'envoyer en fabrication. Autrement dit, le test fonctionnel est effectué pour déterminer si le circuit réalise bien toutes les spécifications décrites dans le cahier de charges et faire le diagnostic en cas d'erreurs pour modifier sa conception. Il représente une approche de base pour les petites puces analogiques et mixtes. Puisque la fonctionnalité des circuits numériques devient de plus en plus complexe, il apparaît qu'il est pratiquement impossible de vérifier les fonctionnalités d'une puce, en particulier les composants numériques larges tels que les microprocesseurs. Ainsi, dans les circuits VLSI le test fonctionnel tend à être remplacé par le test structurel. Ce test améliore le taux de couverture de fautes.

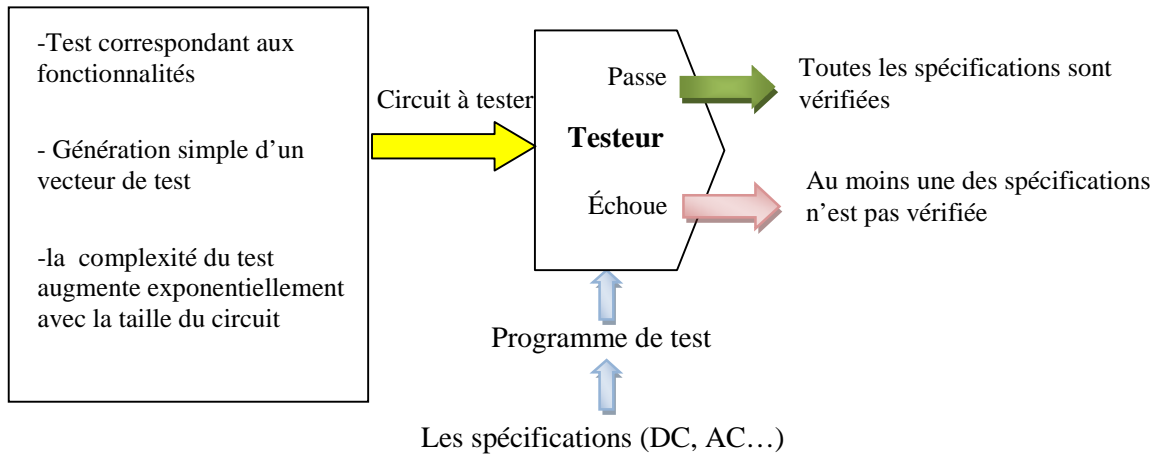


Figure 1. Principe d'un test fonctionnel

### 1.3.2- Test de production

Il permet de déterminer si le circuit fabriqué ne contient pas de défauts physiques et ainsi de séparer les circuits défectueux des circuits sains. Le test de production comprend principalement trois types de test :

- Test alternatif récemment proposé pour les circuits analogiques, mixtes et radiofréquences RF
- Test structurel (test logique), le circuit est vu comme une boîte blanche par ce type de test (Figure 2). Il permet d'utiliser un ensemble optimal de vecteurs de test et nécessite un minimum de temps de test, et par conséquent réduit efficacement le coût du test.
- Test paramétrique ou de caractérisation sur plusieurs lots de circuits et prototypes pour déterminer les limites de fonctionnement du circuit. Ce type de test s'effectue à chaque nouvelle conception ou nouveau processus de fabrication sous différentes conditions

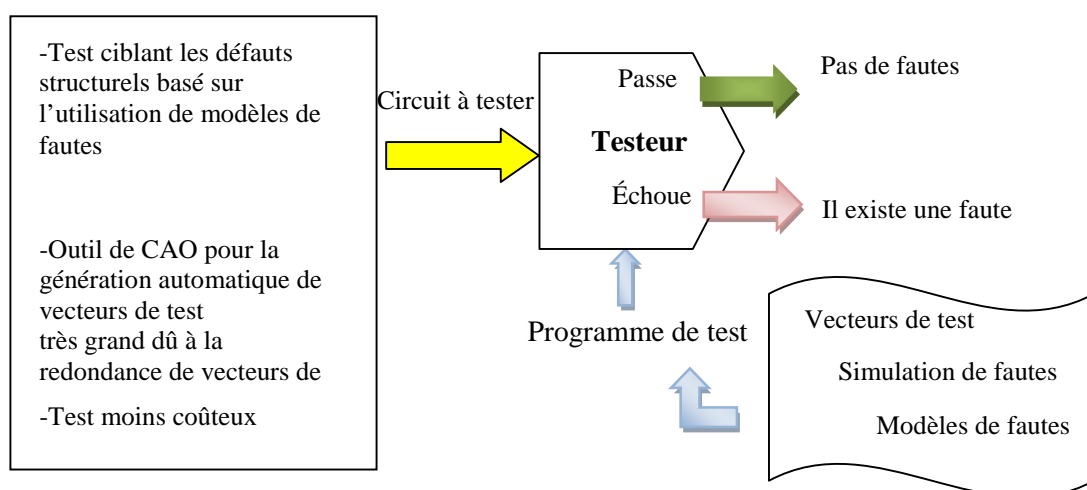


Figure 2.Principe d'un test structurel

#### 1.4- Le test industriel dans les phases de fabrication de circuits intégrés

La fabrication d'un circuit intégré (IC) se fait en deux étapes principales : l'étape de conception (design) et l'étape de fabrication (manufacturing) (Figure 3). Dans chacune des étapes, un test doit être réalisé. Dans l'étape de conception, le test fonctionnel est effectué pour vérifier si les performances du circuit correspondent à celles définies dans le cahier des charges. Si ce n'est pas le cas, un diagnostic doit être fait pour localiser où se trouve l'erreur en vue de modifier la conception du circuit. Dans l'étape de fabrication, le test à effectuer est un test structurel ou de production. Dans cette étape les circuits ne peuvent pas être réparés, ils seront donc jetés. Par contre un diagnostic se fait pour améliorer le rendement de la chaîne de fabrication.

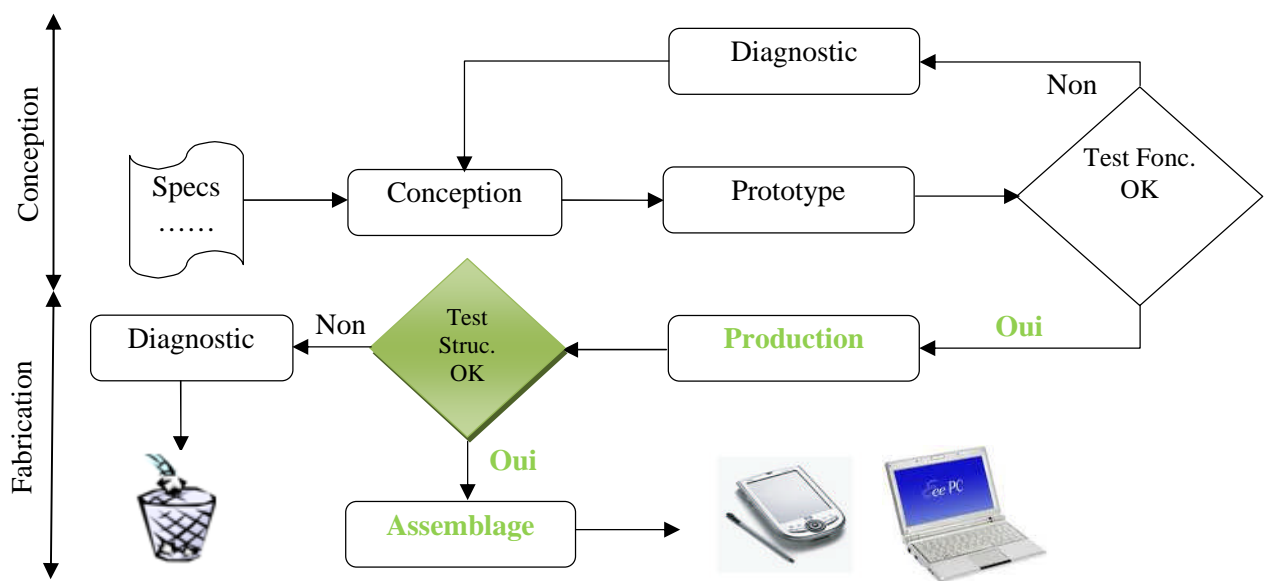


Figure 3. Le test dans les phases de fabrication

#### 1.5- Les différentes phases de test

Le circuit VLSI subit différentes phases de test illustrées à la figure 4, pendant son procédé de fabrication (Figure 5), dont la durée appelée temps de cycle est d'environ deux mois jusqu'à la mise en boîtier. La fabrication d'un circuit intégré passe par deux étapes principales. Dans chacune des étapes de conception et de fabrication les phases de test s'appliquent de la manière suivante :

- Test sur wafer «  $\varnothing=300\text{mm}$  » : réalisé par la machine « wafer prober » pour détecter les défauts de fabrication et éviter le montage en boîtier des circuits défectueux;
- Test du circuit encapsulé (Packaged) : détecter les défauts dus au processus d'encapsulation une fois les dices découpées et les circuits montés en boîtier ;
- Test du circuit sur la carte ;
- Test de la carte dans le système.

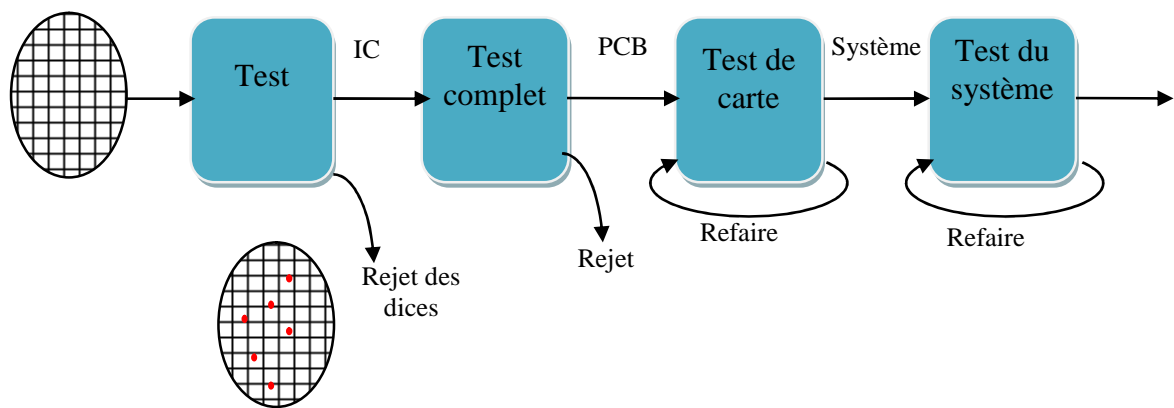


Figure 4. Phases de test

Les principales étapes des deux dernières phases sont:

- Le test de fonctions aux conditions nominales puis aux conditions limites d'environnement ;
- Le test des performances dynamiques ;
- Le test paramétrique (essentiellement les caractéristiques électriques des broches d'entrée/sortie) qui est souvent appliqué aux circuits pour mesurer par exemple, le seuil du courant de fuite  $I_{DDQ}$  consommé par le circuit au repos. Si le circuit sous test consomme moins de ce courant de seuil, il est accepté, sinon, il est rejeté. Ce test détecte certaines défaillances que les tests classiques ne perçoivent pas. En effets, certains défauts, tels qu'un court circuit entre polysilicium et oxyde de grille ou encore un défaut de pont entre deux pistes provoquent une surconsommation du courant  $I_{DDQ}$ . La détection de ce dernier nécessite une minuterie de capture pour effectuer la comparaison requise. Des solutions sont utilisées aujourd'hui et qui s'étendent des capteurs de courants hors puce (off-chip sensors) aux capteurs de courant intégrés (built-in current sensor BICS) pour des circuits à haute fiabilité.

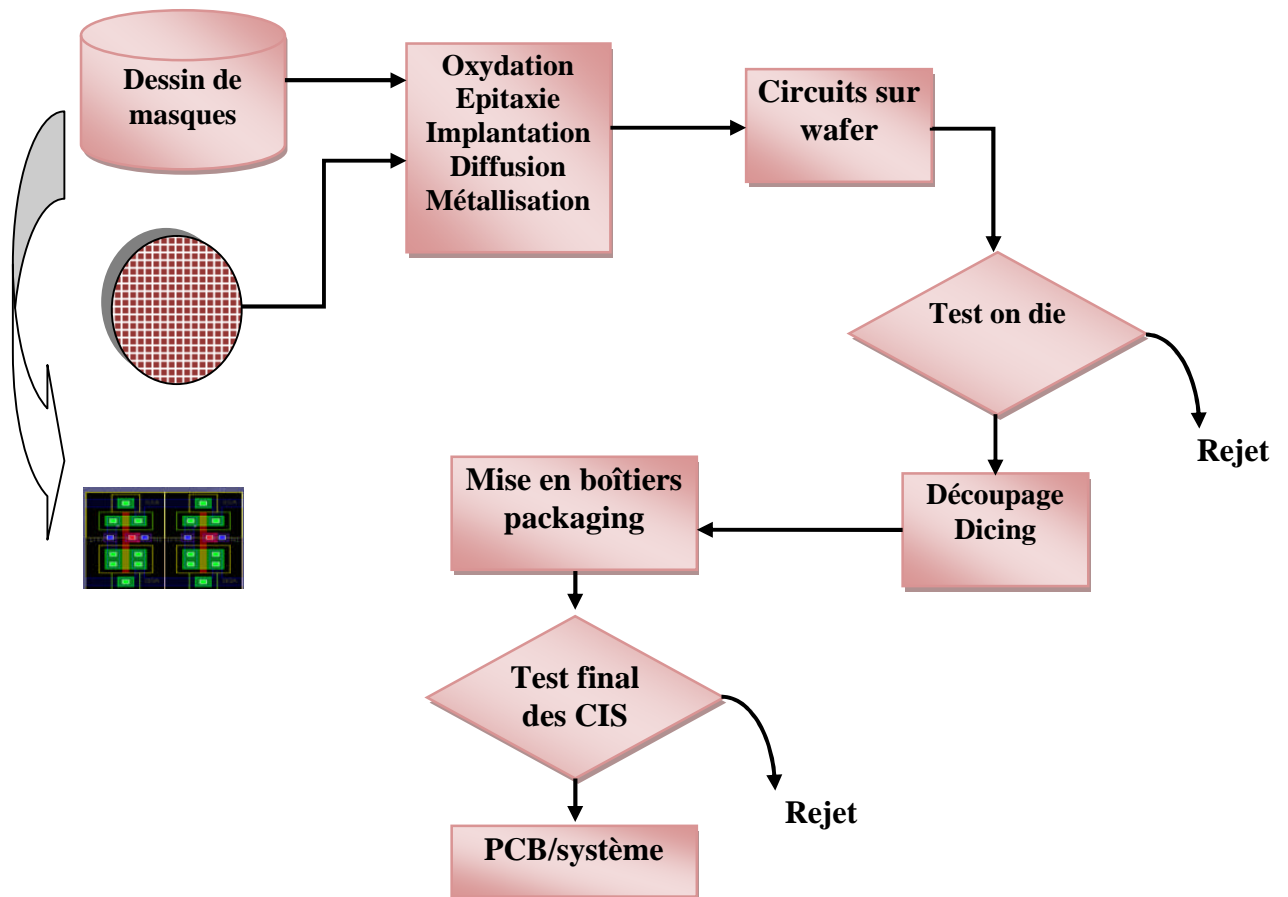


Figure 5. Processus de fabrication des circuits intégrés.

### 1.6- Motivation du test

Au niveau de leur coût, les tests suivent la règle de dix (10). Plus un défaut est détecté tôt dans le processus de fabrication, moins le coût qu'il va induire est élevé. Par exemple, si un circuit intégré est détecté défectueux, le coût du boîtier, de son intégration dans un système et ensuite de la localisation de l'erreur dans ce système sont évités. La figure 6 présente le coût d'un défaut en fonction du moment où il est repéré.

En résumé, le test constitue plus de **70%** de l'effort du développement de circuits (systèmes) électroniques.

**Règle de dix :**

- Rondelle (Wafer) \$ 0,01-\$0,1
- Puce, circuit encapsulé \$ 0,1-\$1
- Carte PCB \$1-\$10
- Système \$10-\$100
- Domaine d'application \$100-\$1000

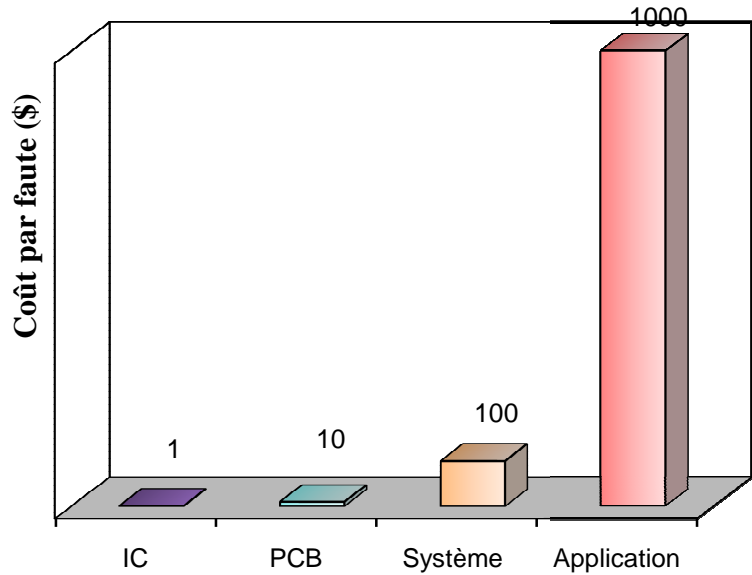


Figure 6. Coût de détection des défauts (règle des X 10)

**1.7- Techniques de test utilisées**

Les différentes techniques de test (Figure 7) utilisées sont : l'utilisation de la simulation de fautes avec les vecteurs de test fonctionnel pour déterminer la liste des fautes non détectées, l'utilisation d'un générateur automatique de vecteurs de test (ATPG) pour détecter les fautes non détectées, et s'il y a encore beaucoup de fautes non détectées, l'utilisation des techniques de conception en vue du test (DFT) est requise pour améliorer la testabilité.

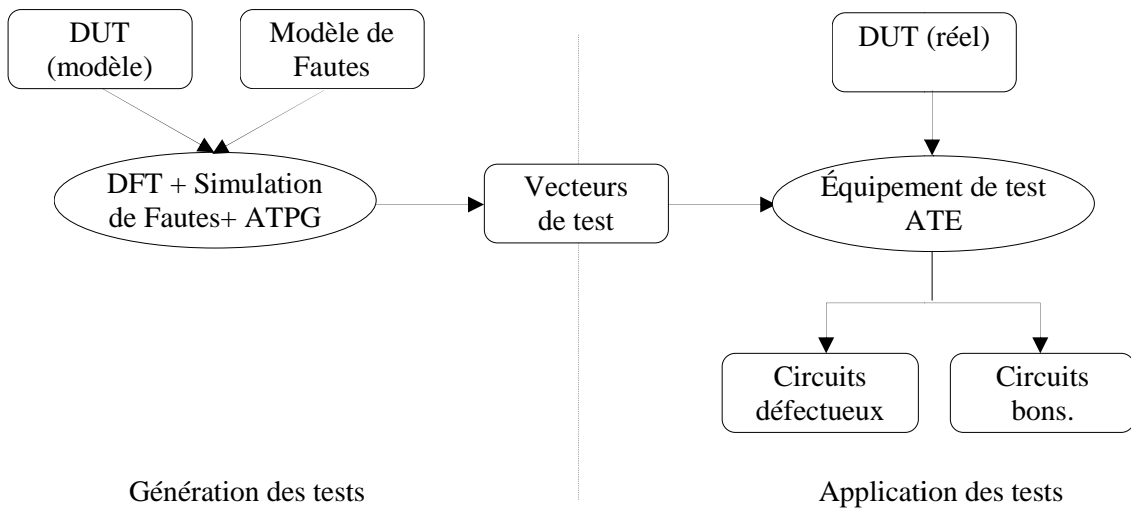


Figure 7. Test des circuits intégrés.

## 1.8 Les aspects économiques du test de production

Les différents aspects économiques du test de production sont : le coût du test, le rendement, le niveau de qualité et le taux de couverture.

### 1.8.1 Coût du test de production

L'objectif du test de production est de réduire le coût du test par seconde ( $C_S$ ) et le temps moyen de test par circuit ( $T_m$ ) et cela en diminuant le nombre de vecteurs de test et en appliquant les tests qui détectent le plus de défauts mais aussi les plus probables. Le coût du test de production est donné par la formule suivante :

Coût =  $N_C * C_S * T_m$ , où  $N_C$  est le nombre de circuits.

**Exemple :**  $N_C = 4$  millions,  $C_S = 0.03$  \$/s et  $T_m = 4$  s

Coût du test = 480 000 \$

### 1.8.2 Rendement de fabrication

Noté couramment  $Y$  (Yield), le rendement exprime la rentabilité d'un processus de fabrication, il dépend :

- Du processus de fabrication utilisé ;
- La propreté de la salle blanche ;
- Des variations de température ;
- Des vibrations ;
- De la précision des machines ;
- De la pureté des produits utilisés.
- Essentiellement de la surface du circuit fabriqué.

$$\text{Rendement : } Y = \frac{\text{nombre de circuits qui passent le test}}{\text{nombre total de circuits}}$$

#### Exemple

Matrice 9x9

Nombre circuits = 45

Nombre circuits défectueux « marqués » = 6

Nombre de bons circuits = 39

$Y = 39/45 = 87 \%$

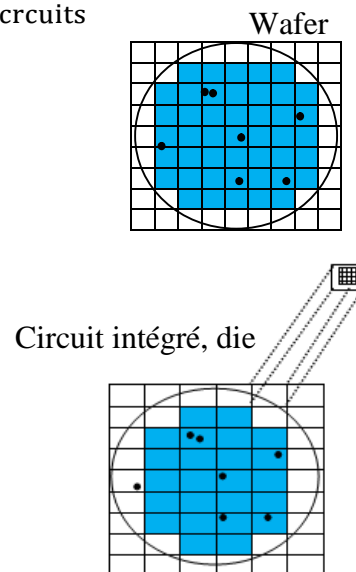
Matrice 6x9

Nombre circuits = 24

Nombre circuits défectueux « marqués » = 5

Nb de bons circuits = 19

$Y = 19/24 = 79 \%$



### 1.8.3 Niveau de qualité d'un ensemble de test

La figure 8 comment évaluer la qualité d'un test. Les différentes métriques de test peuvent être directement calculées. Le rendement du test est le pourcentage des dispositifs qui passent le test. Le niveau de qualité d'un test correspond au nombre de circuits bons parmi les circuits qui passent le test. D'une part, parmi les dispositifs qui ne passeront pas le test (ou échouent) certains peuvent être bons. Il s'agit dans ce cas là d'une perte de rendement. D'autre part, parmi les dispositifs qui passeront le test certains contiennent toujours des défauts et peuvent être défectueux. On parle ici d'un niveau de défautuosité ou taux de défauts. Dans les deux cas, les décisions prises sont fausses et sont dues à l'imprécision des tests. Plus le test est précis plus le nombre des circuits qui passeront le test est petit.

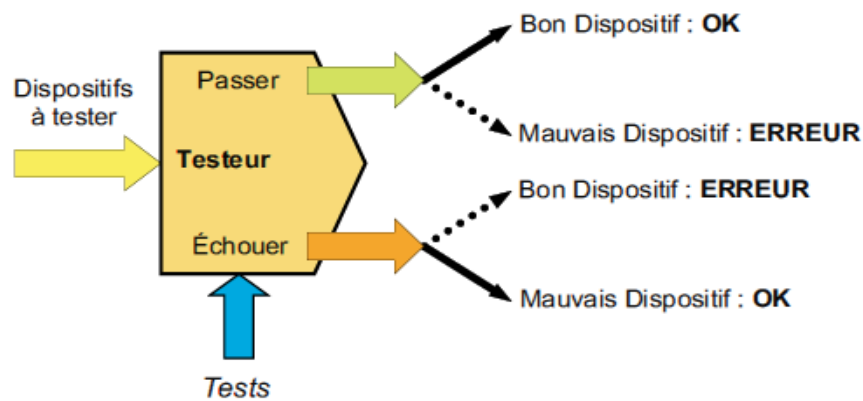
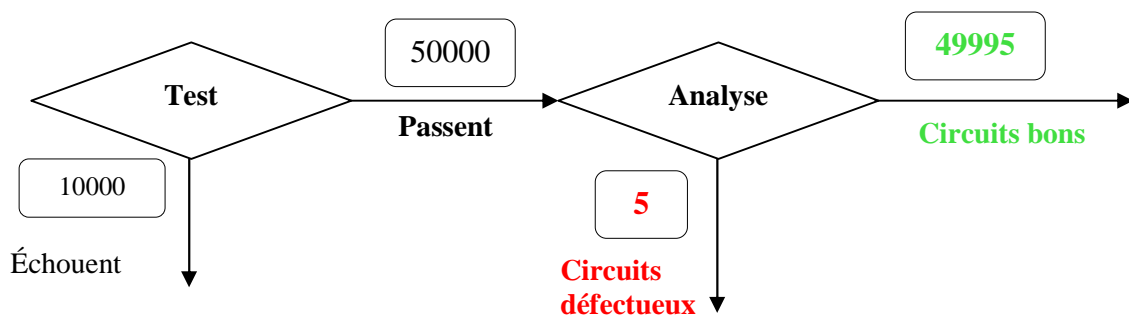


Figure 8. La qualité d'un test

**Exemple :**



Niveau de qualité « Quality level » :  $QL = \frac{\text{Le nombre de circuits bons}}{\text{Le nombre de circuits qui passent le test}}$

$$QL = \frac{49995}{50000} = 0.9999 = 99,99 \%$$

Niveau de défauts « Defect Level » :  $DL = \frac{\text{Le nombre de circuits défectueux}}{\text{Le nombre de circuits qui passent le test}}$

$$= \frac{5}{50000} = 0.0001 = 0.01 \%$$

$$= 1 - QL$$

• **Remarque : Niveau de qualité d'une carte électronique  $QL_{carte}$**

Le niveau de qualité d'une carte contenant N circuits correspond à la probabilité  $p$  que la carte ne contient aucun circuit défectueux. Il est donné par la formule suivante :

$$P = QL_{carte} = \prod_{i=1}^N QL_i = \prod_{i=1}^N (1 - DL_i)$$

Comme le montre la figure 9, le niveau de qualité de la carte diminue avec l'augmentation du nombre de circuits N.

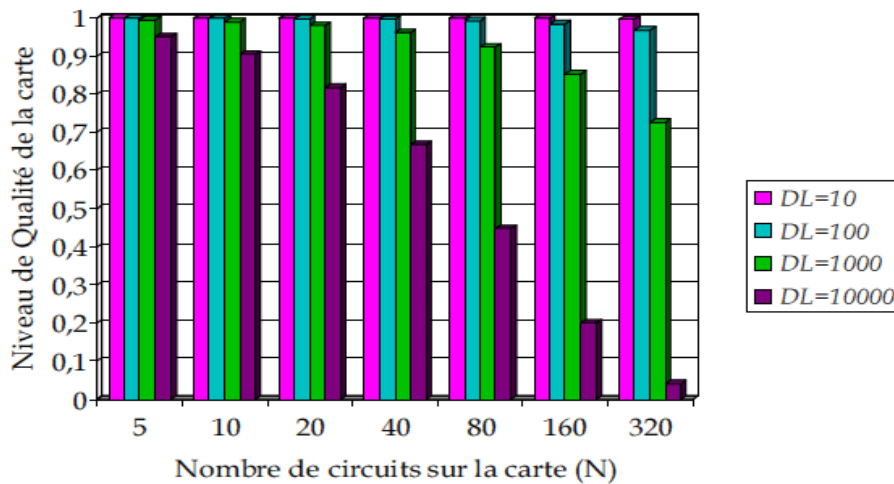


Figure 9. Niveau de qualité de la carte en fonction du nombre de circuits N

**1.8.4- La couverture de fautes**

Le paramètre le plus utilisé pour évaluer la qualité d'un test dans le cas des fautes simples et des circuits numériques est la couverture de fautes F qui correspond à la probabilité de détection de fautes dans le circuit. Cette probabilité est estimée comme suit:

$$F = \frac{\text{le nombre de fautes détectées}}{\text{le nombre total de fautes}}$$

Une première relation entre le Taux de défauts et la Couverture de fautes, appelée formule de Williams et Brown, a été définie comme suit :

$$DL = 1 - Y^{1-F}$$

où DL est le Taux de défauts, Y est le Rendement et F la Couverture de fautes.

## **Chapitre 2**

# **Test industriel et Équipements de Test « ATE »**

## **2.1- Introduction**

Aujourd'hui, les circuits intégrés sont de véritables systèmes intégrés sur une même puce. Ces systèmes « SOC » (System On Chip) peuvent intégrer divers éléments tels que des composants numériques, analogiques ou encore des mémoires. Pour répondre à la problématique du test de ces SOC, les testeurs souvent désignés par l'acronyme ATE (Automatic Test Equipment) sont de plus en plus complexes et performants.

## **2.2- Conception des circuits VLSI**

Elle consiste à décrire le circuit du niveau comportemental jusqu'au niveau layout (dessin des masques) en vue de fabriquer correctement et sans erreur un circuit intégré (ou puce électronique). Passer d'un niveau au suivant et vérifier le résultat obtenu se fait en utilisant des outils logiciels. Comme le montre la Figure 1, la conception d'un circuit procède en plusieurs étapes, et notamment :

- La spécification fonctionnelle décrivant la fonctionnalité voulue de la puce, ainsi que des contraintes non fonctionnelles (surface, coût, consommation...),
- La modélisation du circuit à très haut niveau,
- La vérification,
- La synthèse,
- La représentation sous forme d'un fichier informatique des dessins de masques (layout) de la puce.

Ce fichier permet la fabrication des masques qui serviront lors de la fabrication de la puce dans les unités de fabrication de semi-conducteur pendant les étapes de photolithographie.

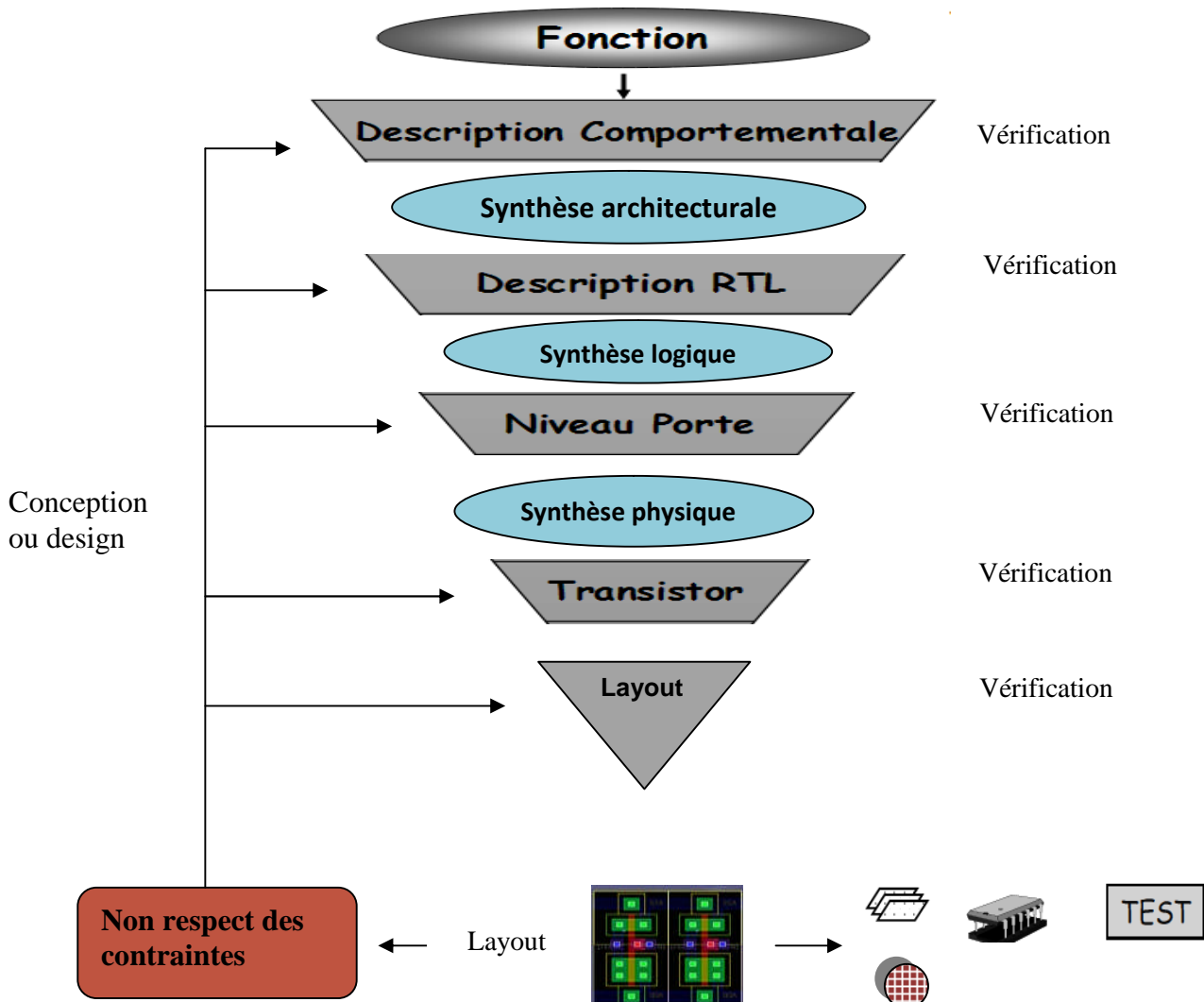


Figure 1. Flot de conception de circuits intégrés

### 2.2.1- La modélisation du système

Le système complet (matériel et logiciel) est modélisé à très haut niveau afin de valider en termes de performance l'architecture choisie par rapport aux besoins de l'application. Cette modélisation se fait à l'aide des langages et des simulateurs tel que C++, Verilog ou VHDL,...etc. Quatre types de descriptions sont utilisés : la description au niveau comportementale, la description au niveau transfert de registres RTL (Register Transfer Level), la description au niveau porte logique et la description au bas niveau ou au niveau transistor.

#### 2.2.1.1- Description comportementale

Elle définit la fonction du circuit sans le partitionner en plusieurs blocs et sans définir les connections entre les éléments.

- **Exemple** : la description comportementale de la porte Nand avec le langage VHDL

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity nandgate is
    Port ( a : in  STD_LOGIC;
          b : in  STD_LOGIC;
          y : out STD_LOGIC);
end nandgate;

architecture Behavioral of nandgate is
begin
y <= a nand b;
end Behavioral;

```

### 2.2.1.2- Description RTL

Le circuit est modélisé au niveau transfert de registres. Cette modélisation consiste à décrire l'implémentation sous forme de blocs séquentiels (registres ou bascules) et de blocs logiques. Elle est codée à l'aide d'un langage de description matériel HDL (Hardware Description Language). Cela se fait soit en VHDL, soit en Verilog. Chaque bloc est représenté par son code Verilog ou VHDL.

### 2.2.1.3- Description au niveau porte

Elle définit la liste d'interconnexions ou bien la netlist des opérateurs logiques et des éléments mémoire.

### 2.2.1.4- Description au niveau transistor

Elle définit chaque porte par son réseau de transistor. La Figure 2 représente un exemple de description au niveau transistor de la porte NAND.

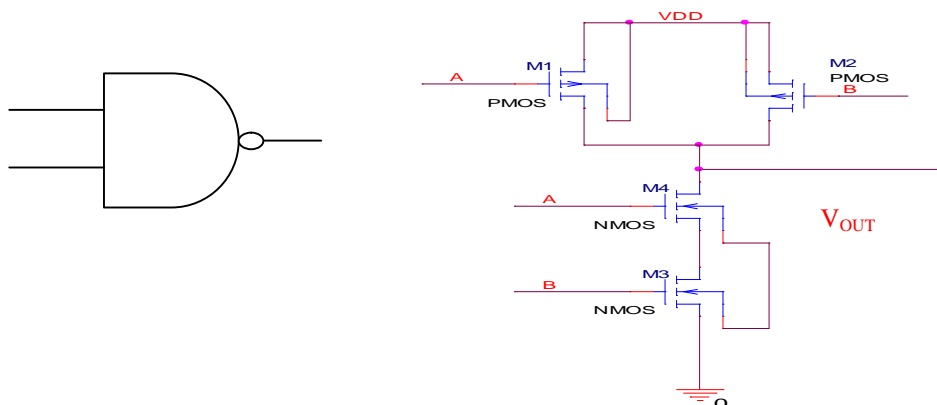


Figure 2. Exemple de description au niveau transistor de la porte NAND

### 2.2.1.5- Description au niveau layout

La Figure 3 représente un exemple d'une porte nand placée et routée. Elle est remplacée par son layout (dessin des masques).

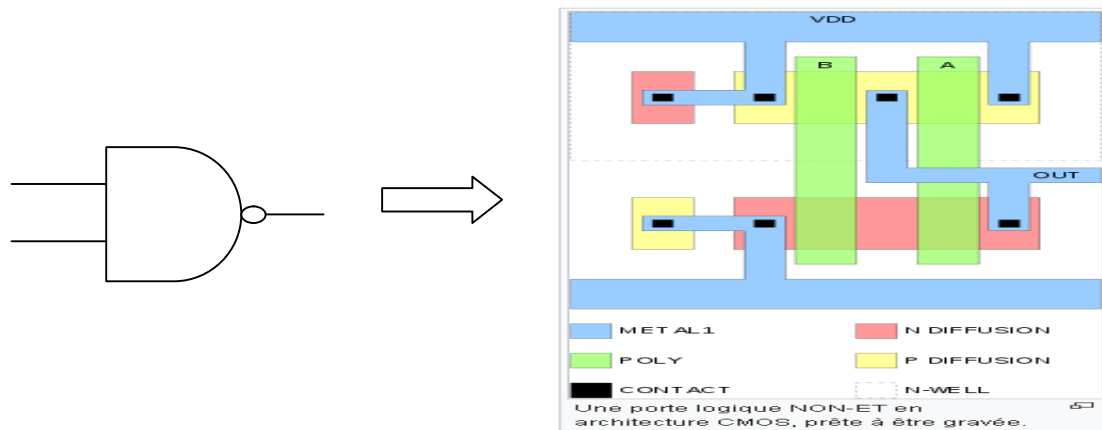


Figure 3. Exemple de description au niveau layout de NAND à 4 entrées

### 2.2.2- la vérification du design

Le circuit est vérifié à chaque niveau (comportemental, RTL, porte, transistor) pour s'assurer du bon fonctionnement du circuit lorsqu'il sera fabriqué: Il existe deux types de vérification : la vérification dynamique et la vérification statique.

#### 2.2.2.1- La vérification dynamique

Le circuit complet peut être simulé uniquement aux niveaux de description les plus élevés (simulateur Verilog/VHDL). Le but est de simuler le code RTL écrit en VHDL ou Verilog pour garantir qu'il n'y a pas d'erreur de conception.

Pour les systèmes complexes actuels l'émulation ou prototypage à base de FPGA est utilisée. Le but est de réaliser un prototype fonctionnel à base de composants à logique programmable de type FPGA ou EPLD comme ceux proposés par les sociétés Xilinx ou Altera. Le prototype peut alors être stimulé dans les conditions réelles de son environnement.

#### 2.2.2.2- La vérification statique

Son objectif est d'assurer de la bonne fonctionnalité du circuit et d'analyser la fréquence de fonctionnement de circuit. L'analyse statique de timing (STA) est réalisée pour s'assurer que les délais dans le design sont respectés dans les différentes conditions d'utilisations prévues (la température de fonctionnement, le voltage, etc). Elle se fait généralement dans le pire cas (Worst Case) et le meilleur cas (Best Case). L'outil Prime-time (Synopsys) est l'un des outils de STA utilisant des générateurs aléatoires de stimuli avec une vérification automatique.

### 2.2.2.3- La vérification physique du dessin de masques (Design Rule Check/DRC).

Elle permet d'assurer que le dessin (layout) correspond au circuit électrique qui a été spécifié, et qu'il peut être fabriqué sans erreurs. La comparaison du layout au schéma électrique se fait par la recherche des courts-circuits ou des coupures dans le circuit, mais aussi par le calcul des caractéristiques électriques, tels que les valeurs des résistances et des capacités du circuit, mais aussi toutes les résistances et capacités parasites.

### 2. 2.3- La synthèse

La synthèse d'un circuit synchrone se fait en trois étapes. La première étape est la synthèse architecturale qui transforme la description comportementale en code RTL. La deuxième étape est la synthèse logique qui utilise des algorithmes mathématiques pour transformer la description RTL en éléments logiques combinatoires et séquentiels. Le résultat de cette étape est un fichier informatique représentant la copie des portes de la bibliothèque cible et leur interconnexion, représentant le circuit électronique (gate netlist). La troisième étape est la synthèse physique qui consiste à transformer la description au niveau porte en une description au niveau layout du circuit.

### 2.3- Test industriel des circuits VLSI

La Figure 4 montre le principe du test industriel d'un circuit VLSI, il consiste à appliquer des vecteurs de test sur les entrées du circuit sous test (DUT) et analyser les sorties.

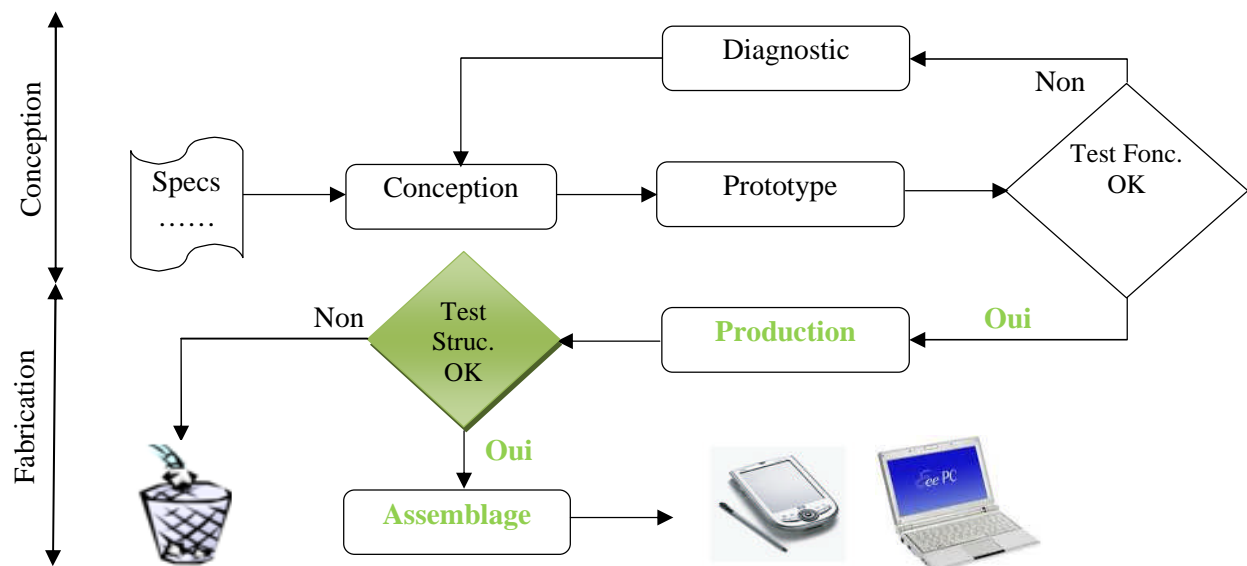
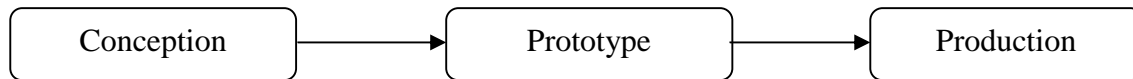


Figure 4. Le test industriel

#### 2.3.1- Différentes phases de test

Le tableau 1 résume les différentes phases de test industriel. Dans un premier temps les tests s'appliquent tout d'abord sur le prototype conçu sont le test de caractérisation et le test

fonctionnel pour vérifier les limites de fonctionnement du circuit. Ce dernier est effectué à chaque nouvelle conception ou nouveau processus de fabrication et sous différentes conditions (pire-cas, etc...). Dans un second temps, le test à effectuer est un test de production ou de fabrication. Le test de production se fait en plusieurs phases: Le test de continuité, le test structurel et le test de caractérisation qui se fait sur plusieurs circuits et plusieurs lots, le test sur wafer et le test après encapsulation.



Modélisation Simulation Synthèse Analyse des délais Placement routage Génération des tests de production	Test des spécifications (test fonctionnel) Test de caractérisation Test pire-cas	Test de continuité Test structurel Test de caractérisation (des échantillons) Test sur wafer Test après encapsulation
---	--	---

**Tableau 1. Phases de test**

### 2.3.2- Test de production

Comme le montre la Figure 5, le test de production est réalisé sur chaque circuit sous différentes conditions (pire-cas). Ce type de test permet de détecter les défauts de fabrication pour séparer les circuits défectueux des circuits fonctionnels. Son but est de couvrir un grand nombre de fautes et de minimiser le temps de test (coût de test).

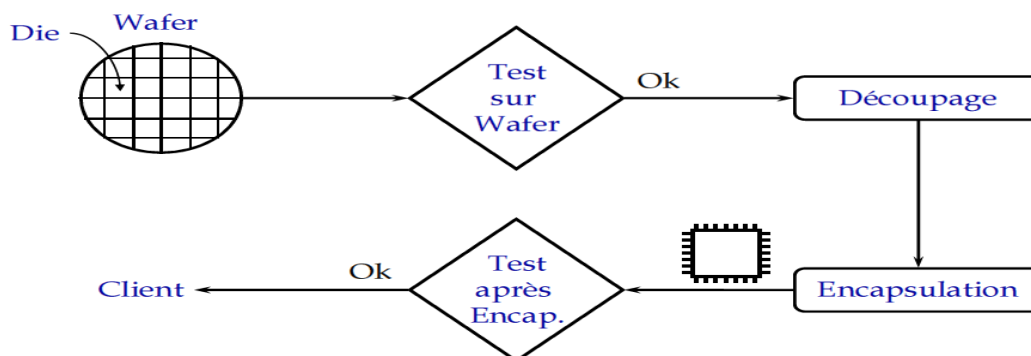


Figure 5. Le test de production.

### 2.3.3- Test pire-cas

Avec l'utilisation des outils statistiques le test pire-cas est effectué sur des circuits aléatoires du processus de fabrication pour déterminer les limites de fonctionnement du

circuit. En utilisant le dessin « shmoo plot » les régions des paramètres où le circuit fonctionne correctement sont identifiées. Ce type de test permet aussi de valider la « Data sheet ».

### 2.3.3.1- Dessin « Shmoo »

Comme le représente la Figure 6 le dessin ou la courbe « Shmoo » est l'ensemble des résultats obtenus en faisant varier simultanément plusieurs paramètres et indiquant les régions où le circuit fonctionne « pass » et les régions où le circuit ne fonctionne pas correctement « fail ». Tous les testeurs industriels disposent d'un mode « Shmoo ».

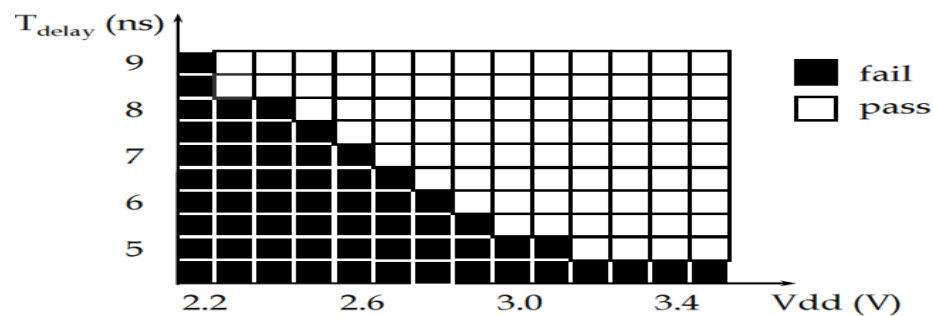


Figure 6. Le dessin « Shmoo ».

### 2.3.3.2- La data sheet

La « data sheet » est un document qui contient toutes les informations sur le circuit, et notamment la description du circuit et son principe de fonctionnement, les valeurs min, max et typiques de tous les paramètres électriques du circuit et les digrammes d'analyse de timing. Elle est aussi l'élément de base pour créer un plan de test de caractérisation d'un circuit.

## 2.4- Paramètres de test

A partir des caractéristiques électriques des circuits, différents paramètres de test sur lesquels se base un testeur industriel peuvent être définis. Parmi ces paramètres, nous pouvons citer :

- Les tensions de seuil  $V_{OH}$  « Output high voltage »,  $V_{OL}$  « Output low voltage »,  $V_{IL}$  « Input low voltage » et  $V_{IH}$  « Input high voltage ».
- Les marges de bruit
- Délais de propagation
- Temps de setup  $T_{su}$
- Temps de hold  $T_h$

## 2.5- Inverseur CMOS

L'inverseur logique est la base des circuits numériques. Le fonctionnement et les caractéristiques de l'inverseur CMOS peuvent être extrapolés aux autres circuits numériques CMOS plus complexes. La fonction de l'inverseur consiste à transformer un niveau logique en son opposé (haut en bas ou bas en haut). La Figure 7 illustre son symbole et sa structure électrique. L'inverseur CMOS est composé de deux transistors MOSFET de polarité opposée (NMOS et PMOS). Son fonctionnement peut être expliqué par l'étude de son comportement statique et dynamique. On suppose que les deux transistors de l'inverseur sont appariés.

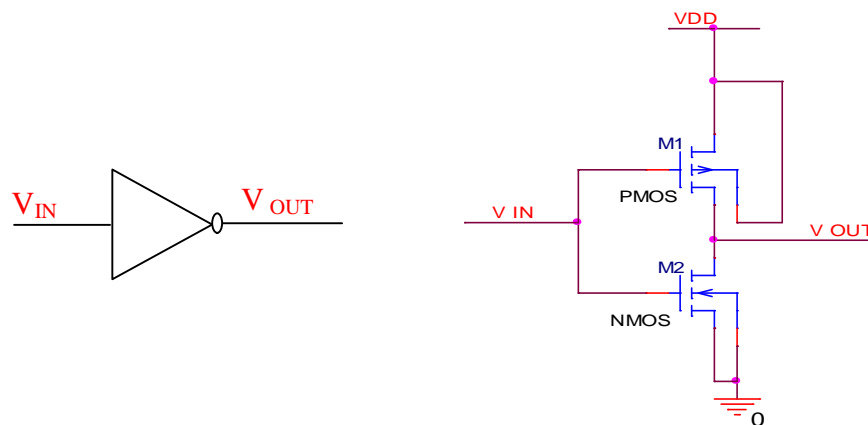


Figure 7. Symbole et structure électrique d'un inverseur CMOS.

### 2.5.1- Comportement statique (ou DC)

Le comportement de l'inverseur CMOS est représenté par sa caractéristique de transfert  $V_{OUT} = f(V_{IN})$ , illustrée à la Figure 8. Trois différentes étapes de son fonctionnement statique peuvent être définies :

- Région Z-A : lorsque la tension d'entrée ( $V_{IN}$ ) est inférieure à la tension de seuil du transistor NMOS ( $V_{THN}$ ), celui-ci est bloqué et le transistor PMOS est en mode saturé (il conduit). Dans ce cas, la tension d'alimentation est reliée à la sortie de l'inverseur par le transistor PMOS en régime de saturation. Autrement dit, la tension de sortie de l'inverseur ( $V_{OUT}$ ) est au niveau logique HAUT proche de la tension d'alimentation VDD (Figure 9.a).
- Région A-D : Lorsque la tension d'entrée  $V_{IN}$  est supérieure à  $V_{THN}$  et inférieure à la tension de seuil du transistor PMOS ( $V_{DD} - V_{THP}$ ), la tension de sortie  $V_{OUT}$  est dans un état indéterminé. L'augmentation progressive de  $V_{IN}$  provoque des changements de régime de fonctionnement des deux transistors. Le transistor NMOS passe du régime de saturation au régime linéaire tandis que le transistor PMOS passe du régime linéaire au régime saturation. Il existe cependant un point de fonctionnement ( $V_{th}$ ) où les transistors sont tous deux en régime de saturation. Ce point, appelé point de basculement ou tension de seuil de l'inverseur correspond à la condition où  $V_{IN}$  est

égal à  $V_{OUT}$ .

- Région D-E: Lorsque  $V_{IN}$  est supérieure à  $V_{DD} - V_{THP}$ , le transistor NMOS reste en régime de saturation et le transistor PMOS devient bloqué. La sortie est reliée à la masse par le transistor NMOS.  $V_{OUT}$  est au niveau logique BAS, proche de zéro (Figure. 9.b).

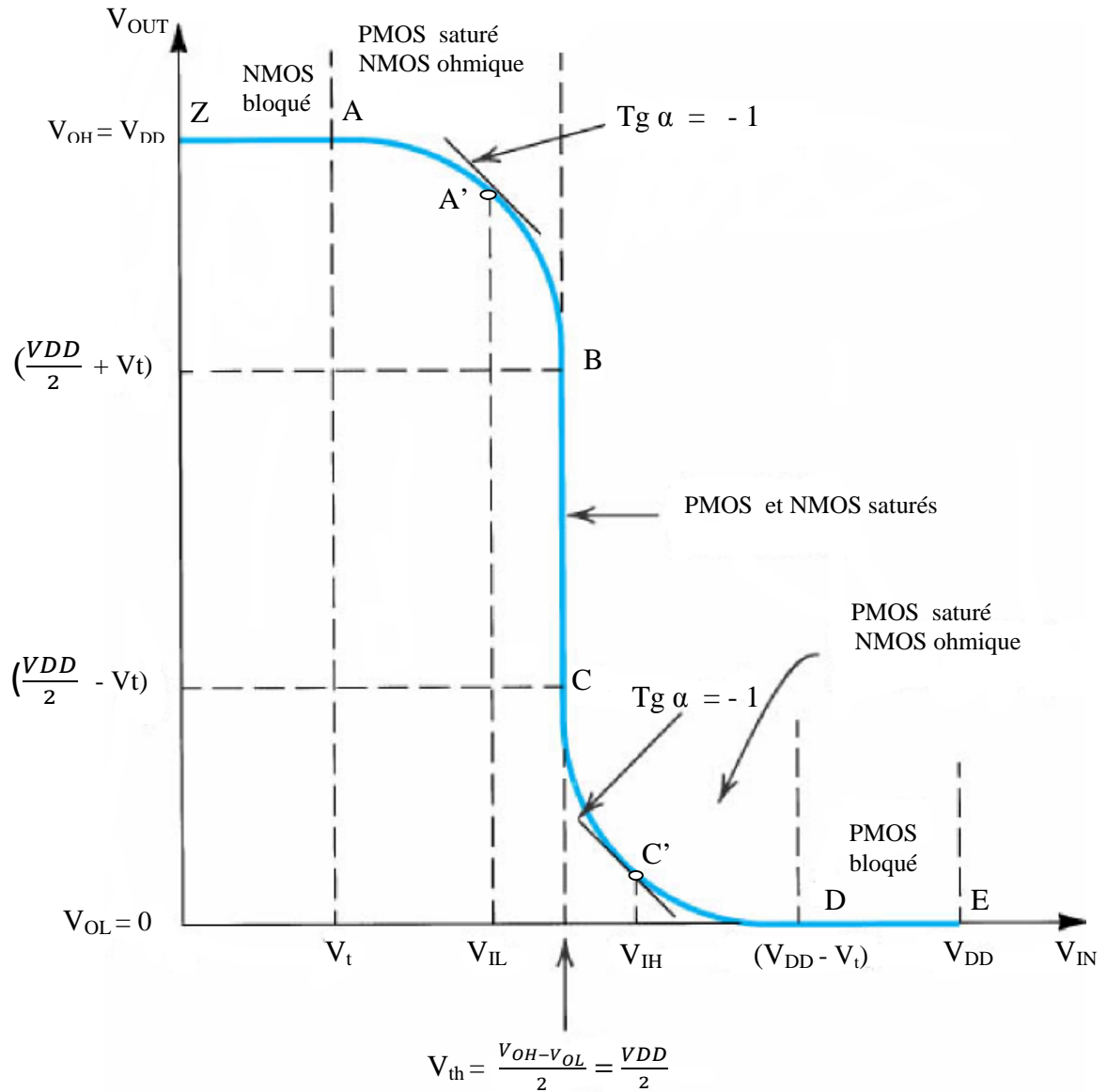


Figure 8. Fonction de transfert d'un inverseur CMOS.

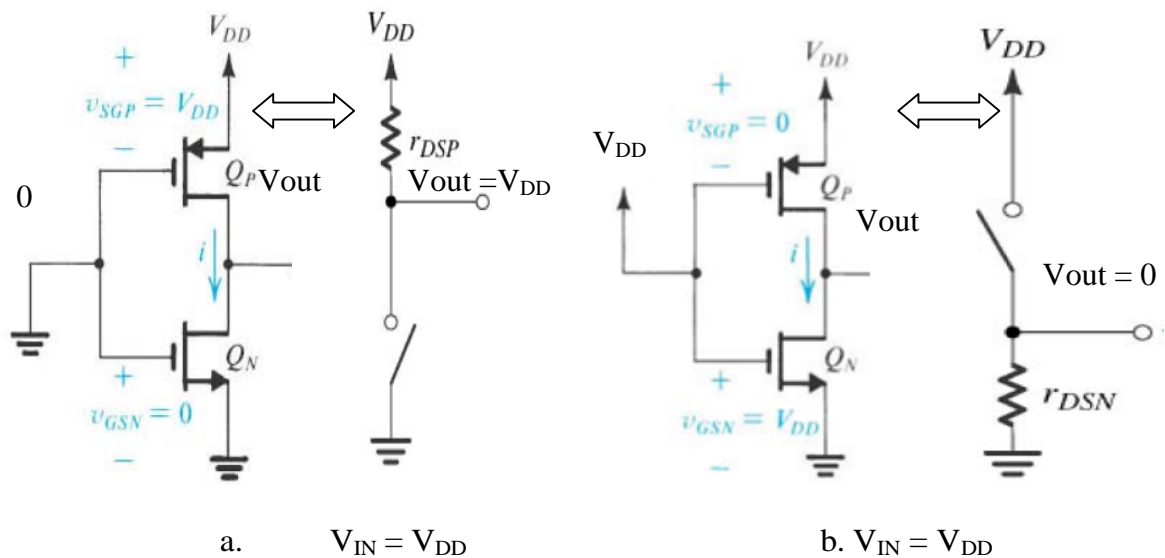


Figure 9. Fonctionnement d'un inverseur CMOS

Cette caractéristique de transfert aide aussi à définir les seuils logiques de l'inverseur CMOS. Parmi ces seuils, nous pouvons citer :

- Le seuil d'entrée définissant le niveau logique BAS ( $V_{IL}$ , pour INPUT LOW), ce niveau de seuil désigne la limite maximale de la tension d'entrée interprétée comme un 0 logique. Cette limite correspond à l'abscisse du point où celle-ci possède une pente de -1 (Point A' sur la Figure 8) tout en ayant une tension sortie de l'inverseur CMOS proche de la tension d'alimentation  $V_{DD}$ .
- Le seuil d'entrée définissant le niveau logique HAUT ( $V_{IH}$  pour INPUT HIGH) : ce niveau de seuil désigne la limite minimale de la tension d'entrée interprétée comme un 1 logique.  $V_{IH}$  peut être également déterminé à partir de la caractéristique de transfert et correspond à l'abscisse du point où la pente de celle-ci est de -1 (Point C' sur la Figure 8) avec une tension de sortie de l'inverseur CMOS proche de la masse.
- Le seuil de sortie définissant le niveau logique BAS ( $V_{OL}$  pour OUTPUT LOW) : ce niveau de seuil désigne la limite maximale de la tension de sortie à partir de laquelle l'état de la sortie de l'inverseur CMOS est au niveau BAS. Elle peut être également évaluée à partir de la caractéristique de transfert et correspond à l'ordonnée du point C' de la Figure 8. Sa valeur idéale est  $V_{OL} = 0$ .
- Le seuil de sortie définissant le niveau logique HAUT ( $V_{OH}$  pour OUTPUT HIGH) : ce niveau de seuil désigne la limite minimale de la tension de sortie à partir de laquelle l'état de la sortie de l'inverseur CMOS est au niveau HAUT. Elle peut être également évaluée à partir de la caractéristique de transfert et correspond à l'ordonnée du point A' (sur la Figure 8). Sa valeur idéale est  $V_{OH} = V_{DD}$ .

Ces paramètres sont liés entre eux par les marges de bruit dont les valeurs idéales ou bien maximales pour un inverseur idéal sont de  $NM_H = NM_L = V_{DD} / 2$

- La marge de bruit HAUT ( $NM_H$  pour NOISE MARGIN HIGH) est égale à la différence entre  $V_{OH}$  et  $V_{IH}$  ( $NM_H = V_{OH} - V_{IH}$ ).
- La marge de bruit BAS ( $NM_L$  pour NOISE MARGIN LOW) est égale à la différence entre  $V_{IL}$  et  $V_{OL}$  ( $NM_L = V_{IL} - V_{OL}$ ).

**Exemple : calcul des tensions de seuil de l'inverseur à transistors appariés**

On analyse le comportement de chacun des transistors en fonction de leur tension de grille.

$V_G = V_{IN}$ .

Le montage en série implique :

$$\left\{ \begin{array}{l} V_{GSn} = V_{IN} \\ V_{DSn} = V_{OUT} \end{array} \right. \quad \begin{array}{l} V_{GSp} = -(V_{DD} - V_{IN}) \\ V_{DSp} = -(V_{DD} - V_{OUT}) \end{array} \quad I_{Dn} = I_{Dp}$$

On suppose que les deux transistors de l'inverseur sont appariés. Autrement dit,  $V_{Tn} = -V_{Tp}$  on notera  $V_T$  la valeur positive de  $V_{Tn}$  qui vaut :

$$V_{Tn} = |V_{Tp}| = V_T \quad \text{et} \quad \frac{W_P}{W_n} = \frac{\mu_n}{\mu_p}$$

On rappelle les équations statiques du transistor NMOS ( $V_{DS} > 0$  et  $V_T > 0$ ) :

- Régime bloqué pour  $V_{GS} \leq V_T$  :  $I_D = 0$ .....(2.1)
- Régime résistif ou linéaire pour  $V_{DS} < V_{GS} - V_T$  :  $I_D = \mu_n C_{OX} \frac{W}{L} \left[ (V_{GS} - V_T) V_{DS} - \frac{1}{2} V_{DS}^2 \right]$ .....(2.2)
- Régime saturé pour  $V_{DS} \geq V_{GS} - V_T$  :  $I_D = \frac{1}{2} \mu_n C_{OX} \frac{W}{L} (V_{GS} - V_T)^2$ .....(2.3)

Et les équations statiques du transistor PMOS ( $V_{DS} < 0$  et  $V_T < 0$ ) :

- Régime bloqué pour  $V_{GS} > V_T$  :  $I_D = 0$ .....(2.4)
- Régime résistif pour  $V_{DS} > V_{GS} - V_T$  :  $I_D = \mu_p C_{OX} \frac{W}{L} \left[ (V_{GS} - V_T) V_{DS} - \frac{1}{2} V_{DS}^2 \right]$ .....(2.5)
- Régime saturé pour  $V_{DS} \leq V_{GS} - V_T$  :  $I_D = \frac{1}{2} \mu_p C_{OX} \frac{W}{L} (V_{GS} - V_T)^2$ .....(2.6)

• **Calcul de  $V_{OH}$**

- $V_{IN} < V_T \Rightarrow$  le NMOS est bloqué
- $V_{GSp} < -(V_{DD} - V_T)$  le PMOS est donc conducteur leur courant commun est  $I_D \approx 0 \Rightarrow V_{DSp} = 0$  (mode résistif) et  $V_{OUT} = V_{DD}$ . Autrement dit, le point d'opération (Fig 10.a) représente le point d'intersection des deux caractéristiques  $I - V_{OUT} \Rightarrow I = 0$  et  $V_{OUT} = V_{DD}$

Donc  $V_{OH}$ , qui est la tension de sortie lorsque  $V_{IN} = 0$ , vaut :  $V_{OH} = V_{DD}$

• **Calcul de  $V_{OL}$**

- $V_{IN} > (V_{DD} - V_T) \Rightarrow$  le NMOS est conducteur (résistif)

- $V_{GSp} > -V_T$  le PMOS est bloqué leur courant commun est  $I_D \approx 0 \Rightarrow V_{DSn} = 0$  (mode résistif) et  $V_{OUT} = 0$ . Autrement dit, l'intersection des deux caractéristiques  $I-V_{OUT} \Rightarrow I=0$  et  $V_{OUT} = V_{DD}$  (Fig.10.b).

Donc  $V_{OL}$ , qui est la tension de sortie lorsque  $V_{IN} = V_{DD}$ , vaut :  $V_{OL} = 0$

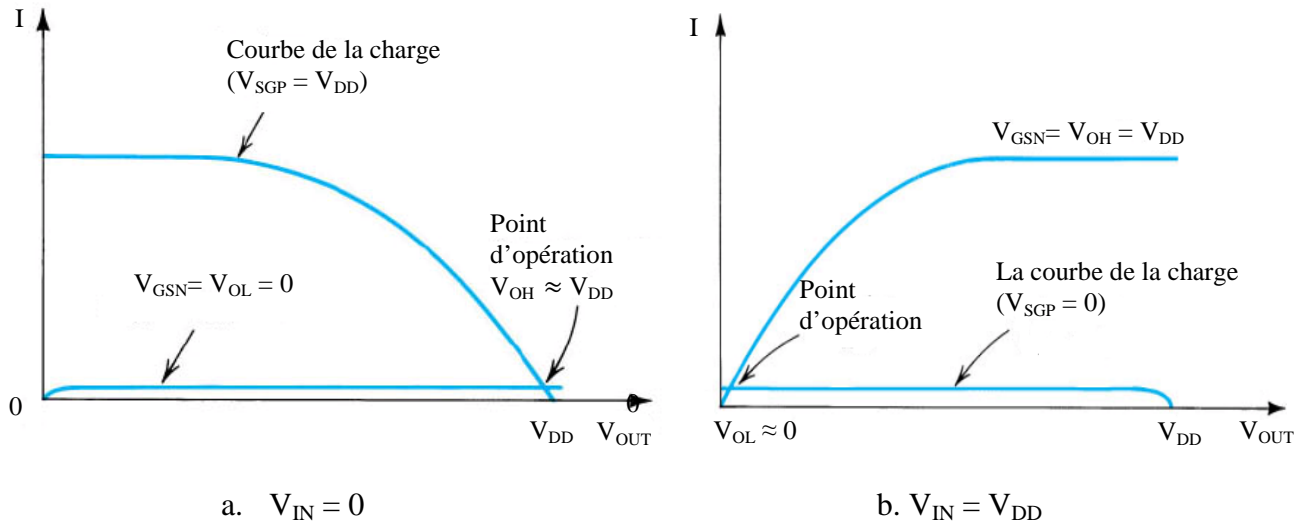


Figure 10. les deux cas extrêmes de l'inverseur CMOS

• **Calcul de  $V_{IL}$  et  $V_{IH}$**

Les points  $V_{IL}$  et  $V_{IH}$  correspondent à la relation  $\left(\frac{dV_{OUT}}{dV_{IN}}\right)_{VIL} = -1$ .

Pour  $V_{IL}$  (région A-B) le NMOS est en saturation, et le PMOS est en régime résistif.

$V_{IL}$  est obtenue en résolvant le système d'équations :

$$\begin{cases} I_{DN}(\text{sat}) = I_{DP}(\text{lin}) \\ \left(\frac{dV_{OUT}}{dV_{IN}}\right)_{VIL} = -1 \end{cases}$$

La relation  $I_{Dn} = I_{Dp}$  se traduit par :

$$I_{DN} = \frac{1}{2} \mu_n C_{OX} \frac{W}{L} (V_{GSn} - V_{Tn})^2 = I_{DP} = \mu_p C_{OX} \frac{W}{L} [(V_{GSp} - V_{Tp}) V_{DSp} - \frac{1}{2} V_{DSp}^2] \dots (2.7)$$

Avec  $V_{GSn} = V_{IN}$ ,  $V_{DSn} = V_{OUT}$ ,  $V_{GSp} = -(V_{DD} - V_{IN})$  et  $V_{DSp} = -(V_{DD} - V_{OUT})$

$$(1) \Leftrightarrow (V_{IN} - V_T)^2 = 2(V_{IN} - V_{DD} + V_T)(V_{OUT} - V_{DD}) - (V_{OUT} - V_{DD})^2 \dots (2.8)$$

En dérivant par rapport à  $V_{IN}$ , on obtient :

$$V_{IN} - V_T = (V_{IN} - V_{DD} + V_T) \frac{dV_{OUT}}{dV_{IN}} + (V_{OUT} - V_{DD}) - (V_{OUT} - V_{DD}) \frac{dV_{OUT}}{dV_{IN}}$$

En remplaçant  $\frac{dV_{OUT}}{dV_{IN}} = -1$  correspondant au point d'abscisse  $V_{IN} = V_{IL}$ , on obtient :

$$V_{IN} - V_T = 2V_{OUT} - V_{IL} - V_T - V_{DD} \Rightarrow V_{IN} = V_{OUT} - \frac{V_{DD}}{2} \dots \dots \dots (2.9)$$

En résolvant le système d'équations (2) et (3), on obtient :

$$V_{IL} = \frac{1}{8} (3V_{DD} + 2V_T) \dots \dots \dots (2.10)$$

Pour le point  $V_{IH}$  (région C-D) le NMOS est en régime résistif et le PMOS en saturation.

$$I_{DN} = I_{DP} \Leftrightarrow \mu_n C_{OX} \frac{W}{L} \left[ (V_{GSn} - V_{Tn}) V_{DSn} - \frac{1}{2} V_{DSn}^2 \right] = \frac{1}{2} \mu_p C_{OX} \frac{W}{L} (V_{GSp} - V_{Tp})^2 \dots (2.11)$$

$$(4) \Rightarrow (V_{IN} + V_T - V_{DD})^2 = 2V_{OUT} (V_{IN} - V_T) - V_{OUT}^2 \dots \dots \dots (2.12)$$

En dérivant par rapport à  $V_{IN}$ , on aura :

$$V_{IN} + V_T - V_{DD} = \frac{dV_{OUT}}{dV_{IN}} (V_{IN} - V_T) + V_{OUT} - V_{OUT} \frac{dV_{OUT}}{dV_{IN}} \dots \dots \dots (2.13)$$

En remplaçant  $\frac{dV_{OUT}}{dV_{IN}} = -1$  correspondant au point d'abscisse  $V_{IN} = V_{IH}$ , on obtient :

$$V_{OUT} = V_{IN} - \frac{V_{DD}}{2} \dots \dots \dots (2.14)$$

En résolvant le système des équations (5) et (6), on obtient :

$$V_{IH} = \frac{1}{8} (5V_{DD} - 2V_T) \dots \dots \dots (2.15)$$

### 2.5.2- Comportement dynamique

Il est caractérisé par la forme de son signal de sortie lorsqu'il commute, et dépend des délais de propagation à travers l'inverseur. Ces délais dépendent des propriétés du transistor fonctionnant comme un interrupteur mais aussi des capacités, qu'elles soient parasites ou non, présentent entre la sortie de l'inverseur et la masse (C et Cdb1). L'inverseur est représenté par le schéma de la Figure 11.

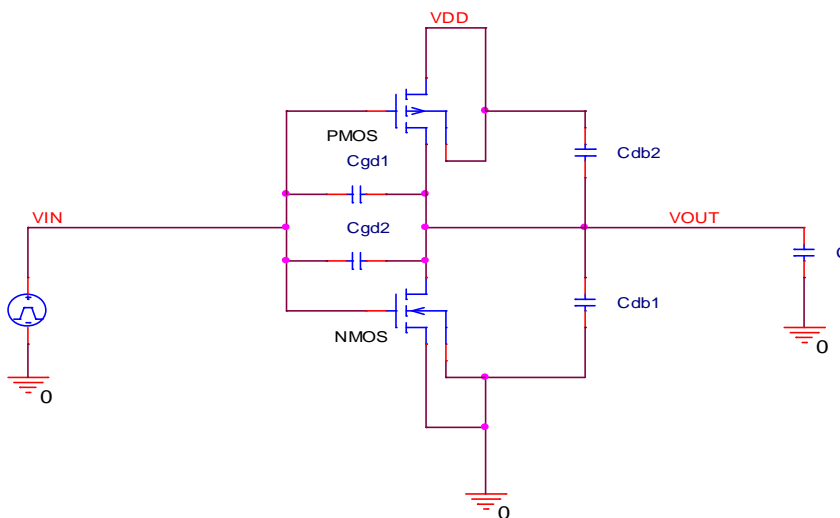


Figure 11. Différentes capacités intervenant dans le calcul des délais de propagation

La vitesse de fonctionnement ou bien la fréquence de fonctionnement des circuits intégrés dépend des délais de propagation à travers les portes logiques et des délais des interconnexions. Ces délais peuvent être évalués à partir des signaux logiques d'entrée et de sortie de l'inverseur CMOS, comme le montre la Figure 12. Cette dernière montre quatre types de délais de propagation :

- $t_{PHL}$ : c'est le temps de propagation d'un niveau haut «high» en entrée à un niveau bas «low» en sortie. Pour l'inverseur CMOS,  $t_{PHL}$  = temps de propagation de  $V_{DD}$  à  $V_{DD}/2$
- $t_{PLH}$ : c'est le temps de propagation d'un niveau bas «low» en entrée à un niveau haut «high» en sortie. Pour l'inverseur CMOS,  $t_{PLH}$  = temps de propagation de 0 à  $V_{DD}/2$ .
- Temps de montée  $t_{THL}$  (de 90% à 10%): c'est le temps de propagation de la sortie de l'inverseur de  $V_{OH}$  «high» à  $V_{OL}$  «low»
- Temps de descente  $t_{TLH}$  (de 10% à 90%) : c'est le temps de propagation de la sortie de l'inverseur de  $V_{OL}$  «low» à  $V_{OH}$  «high».
- 

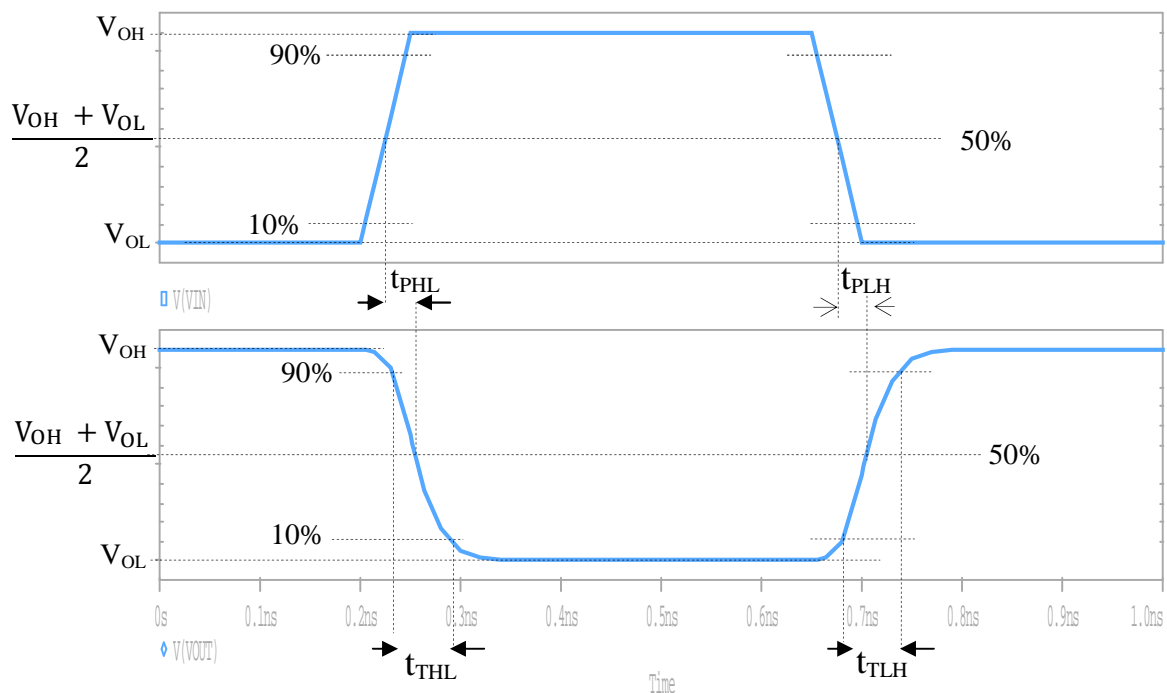
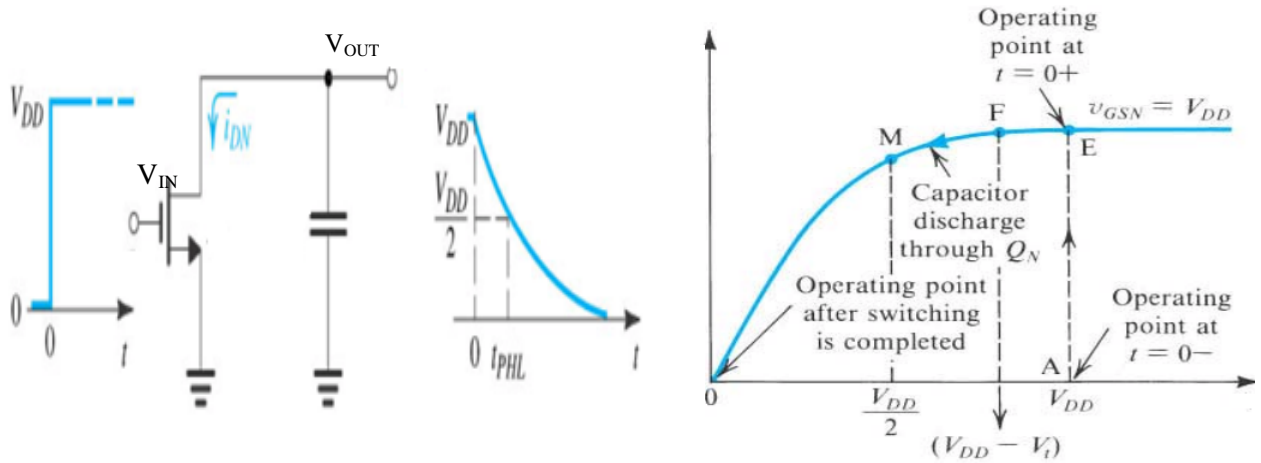


Figure 12. Evaluation des délais de propagation et des temps de transition d'un inverseur logique CMOS ( $V_{DD} = 1.1$  V).

• Exemple : calcul des délais de propagation d'un inverseur CMOS

Parmi les méthodes les plus couramment utilisées pour le calcul des délais on trouve la méthode représentée sur la Figure 13. Cette dernière représente le circuit équivalent pour le calcul du délai  $t_{PHL}$  de l'inverseur ainsi que la trajectoire du point d'opération.



a. Circuit équivalent pour le calcul du délai  $t_{PHL}$

b. Trajectoire du point d'opération

Figure 13. Méthode de calcul de délais de propagation

A  $t = 0$ , le transistor NMOS est en saturation et on a :

$$I_{DN}(0) = \frac{1}{2} \mu_n C_{OX} \frac{W}{L} (V_{DD} - V_T)^2 \dots \dots \dots (2.16)$$

A  $t = t_{PHL}$ , le transistor NMOS est résistif et on a :

$$I_{DN}(t_{PHL}) = \mu_n C_{OX} \frac{W}{L} \left[ (V_{GS} - V_T) V_{DS} - \frac{1}{2} V_{DS}^2 \right] \dots \dots \dots (2.17)$$

$$I_{DN-moy} = \frac{I_{DN}(0) + I_{DN}(t_{PHL})}{2} \dots \dots \dots (2.18)$$

$$t_{PHL} = \frac{C \Delta V}{I_{DN-moy}} = \frac{C V_{DD}}{2 I_{DN-moy}} \dots \dots \dots (2.19)$$

Si on suppose  $V_T = 0.2 V_{DD}$ , on obtient :

$$I_{DN}(0) = 0.32 \mu_n C_{OX} \frac{W}{L} V_{DD}^2 \text{ et } I_{DN}(t_{PHL}) = 0.275 \mu_n C_{OX} \frac{W}{L} V_{DD}^2 \dots \dots \dots (2.20)$$

$$t_{PHL} = \frac{C V_{DD}}{2 * 0.3 \mu_n C_{OX} \frac{W}{L} V_{DD}^2} = \frac{1.66 C}{\mu_n C_{OX} \frac{W_n}{L_n} V_{DD}} \dots \dots \dots (2.21)$$

$$t_{PLH} = \frac{1.66 C}{\mu_p C_{OX} \frac{W_p}{L_p} V_{DD}} \dots \dots \dots (2.22)$$

### 2.5.3- Temps de stabilisation (setup time, $t_{SU}$ ) et de temps de maintien (hold time, $t_H$ )

Deux temps de synchronisation doivent être respectés pour qu'un circuit réponde correctement à ses entrées de commande lorsqu'arrive le front déclencheur de l'horloge. Ces temps correspondent aux positions temporelles de deux entrées d'un même circuit. Un exemple de la position de l'entrée D et de l'horloge d'une bascule D synchrone, qui réagit aux fronts montants de son horloge est présenté sur la Figure 14. Nous définirons ces temps sur cet exemple simple, mais ils se généralisent à toutes les entrées d'un circuit qui provoquent une action conjointe.

- Le temps de stabilisation,  $t_{SU}$ , est l'intervalle qui précède immédiatement le front déclencheur du signal d'horloge, pendant lequel l'entrée de commande doit être stable (inchangée). Si on ne respecte pas ce temps, il n'est pas garanti que la bascule réponde correctement à l'arrivée du front.
- Le temps de maintien,  $t_H$ , est l'intervalle qui suit immédiatement le front déclencheur du signal d'horloge pendant lequel l'entrée synchrone doit être stable. Les fabricants de CI spécifient généralement le temps minimal acceptable. Si on ne respecte pas ce temps, la bascule ne sera pas déclenchée correctement.

### 2.6- Plan de test

Il représente la liste des tests et des procédures à effectuer pour vérifier la qualité du circuit sous test « DUT ». Le plan de test permet de valider la « Data sheet ».

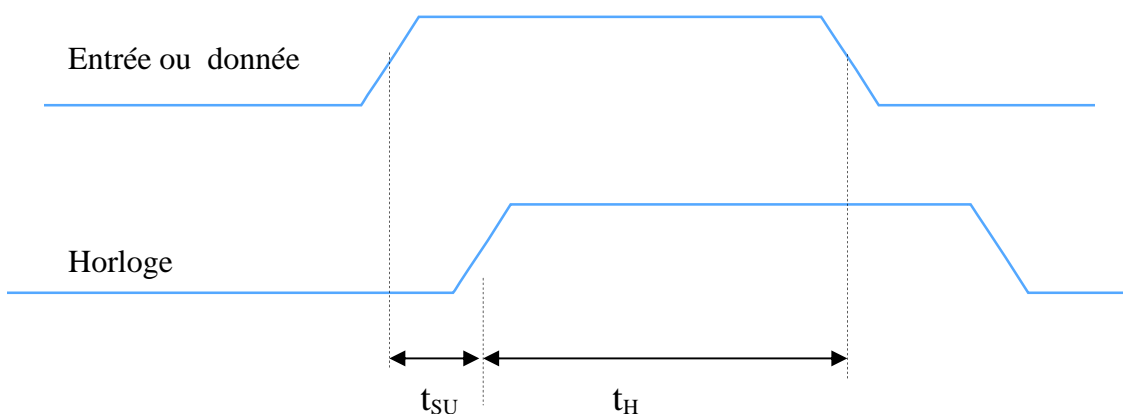


Figure 14. Les entrées de commande doivent rester stables pendant a) un temps  $t_{SU}$  avant la transition active de l'horloge, et b) un temps  $t_H$  après la transition active de l'horloge.

### 2.7- Les testeurs industriels et équipements de test

Le test met en œuvre des matériels coûteux. En production, les ATE sont utilisés en fin de fabrication des circuits dans trois domaines suivants:

- Le test des motifs de surveillance du procédé de fabrication (chaînes de vias, transistors,...). Ces motifs appelés Test Element Group (TEG) permettent de contrôler que les étapes technologiques du procédé de fabrication se sont déroulées

correctement et que les dispositifs intégrés (transistors, interconnexions,...) ont les caractéristiques attendues. Il s'agit donc essentiellement d'un test électrique de caractérisation (mesure de résistances, caractéristiques courant-tension,...).

- Le test des circuits avant découpe : une fois les étapes de fabrication terminées, un premier test des circuits est réalisé sur la plaquette de silicium « wafer » avant sa découpe. Le test sur wafer est appelé test EWS (Electrical Wafer Sort) ou test sous pointes car il est réalisé à l'aide d'une carte à pointes « wafer prober ». Ce test a pour but d'éviter le montage en boîtier des circuits défectueux.
- Le test des circuits en boîtier. Une fois le wafer découpé et les circuits mis en boîtier, les mêmes tests que ceux pratiqués sur wafer sont réalisés mais avec des spécifications plus élevées garantissant la fonctionnalité des circuits avant la livraison chez le client.

### 2.7.1- Testeurs de production

Les testeurs de production sont utilisés à l'étape d'EWS. Ils sont associés à un « prober » pour tester les circuits au niveau wafer. Ce prober a pour rôle d'aligner le wafer avec la carte à pointe et d'indexer le wafer dans le but de se déplacer d'une pièce à une autre ou d'un site de pièce à un autre site de pièce. Il permet également l'encrage des pièces défectueuses qui ne seront pas assemblées en boîtier. Le prober est couplé à la tête du testeur grâce à une interface mécanique spécifique. Les pins électroniques du testeur sont connectées aux « pads » (broches) du DUT à travers des cartes à pointe pour assurer la connexion électrique entre le testeur et le DUT.

### 2.7.2- Testeurs DFT

Les testeurs DFT (Design For Test) sont conçus pour le test des. Ils ne nécessitent pas toutes les fonctionnalités coûteuses d'un ATE traditionnel tel que le test fonctionnel à très haute fréquence. Ces testeurs peuvent aussi bien être utilisés en production (exemple : le testeur « Ocelot » de la société Inovys) présenté par la Figure 15 pour le développement des patterns de test et l'analyse de défaillance.

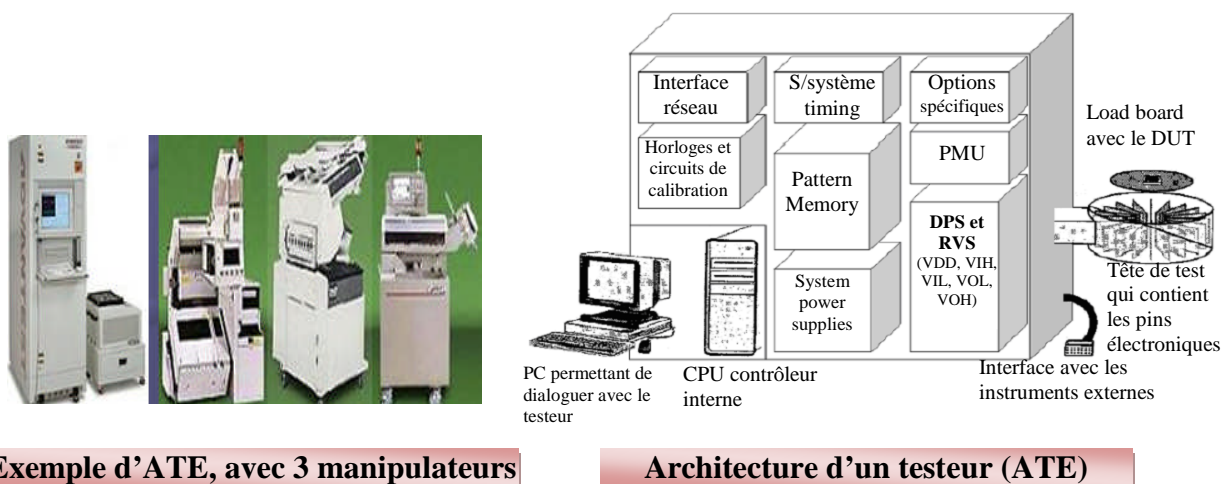


Figure 15. Testeur DFT de production Ocelot de la société Inovys.

### 2.7.3- Équipements d'un testeur ATE

Le schéma de la Figure 16 représente l'architecture d'un ATE. La description des principaux blocs d'un ATE est la suivante:

- Le contrôleur interne permet la gestion des éléments du testeur en fournissant le moyen de transfert des instructions ;
- Le sous système DC contient les éléments permettant de fournir au circuit les tensions d'alimentation « DPS ». Il contient également les tensions de référence « RVS » nécessaires pour fournir les niveaux logiques haut et bas. Ces tensions sont généralement nommées par les abréviations  $V_{IH}$ ,  $V_{IL}$ ,  $V_{OH}$  et  $V_{OL}$ .
- Les mémoires de vecteurs de test qui est une mémoire à accès rapide destinée à stocker les groupes de vecteurs de test « patterns ».
- Le sous système temporel « timing » reçoit les patterns de test provenant de la mémoire vecteur de test et génère les signaux avec les données temporelles décrites dans le pattern de test. Ce dernier est un fichier informatique contenant plusieurs vecteurs de test. le sous système temporel peut être inclus dans les pins électroniques.
- La tête du testeur qui contient les cartes auxquelles appartiennent les pins électroniques du testeur et les cartes d'interface avec le DUT. Chaque pads du circuit utilisé lors du test est connecté à un pin électronique. Les pins électroniques permettent de comparer la réponse mesurée avec la réponse théorique contenue dans le pattern de test et de donner la première validation de la « data sheet ».



**Exemple d'ATE, avec 3 manipulateurs**

**Architecture d'un testeur (ATE)**

Figure 16. L'équipement de test ATE.

### 2.8 Analyse des données de test

L'analyse des données est le processus qui consiste à examiner les résultats de test en vue de diagnostiquer les circuits défectueux et d'améliorer le rendement de fabrication et le plan de test. Cette analyse se fait à l'aide des outils statistiques.

# **Chapitre 3**

## **Modélisation de défauts**

### 3.1- Introduction

La loi empirique de Gordon Moore « cofondateur d'Intel » prédit que la densité d'intégration des circuits intégrés double tous les 18 mois. Cependant cette intégration croissante qui permet de réaliser des systèmes intégrés sur une seule puce, systèmes SOC « systèmes on chip », systèmes SIP « systèmes In Package », a un impact négatif sur la fiabilité des circuits. En effet, à cause de la miniaturisation croissante des procédés de fabrication « 90 nm, 65 nm, 45 nm,... » Il est de plus en plus difficile de réaliser un circuit intégré sans aucun défaut de fabrication.

Pour développer une stratégie de test efficace, il est très important de définir des modèles de défaut (Fig.1) les plus représentatifs des défauts réels des technologies CMOS actuelles et à venir.

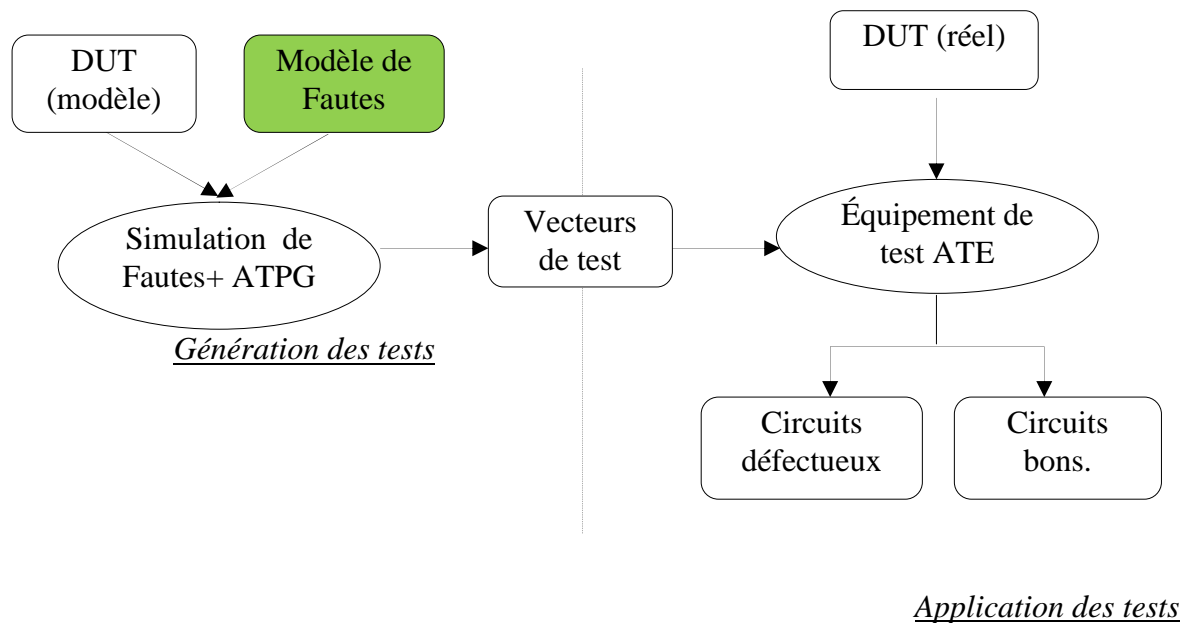


Figure 1. Test des circuits intégrés

### 3.2-Défauts et fautes

Les défauts sont des pannes physiques qui affectent le layout d'un circuit. L'impact des défauts sur les caractéristiques électriques d'un circuit intégré déviant au-delà des valeurs spécifiées est appelée faute. La faute est donc la conséquence d'un défaut mais aussi la représentation des défauts au niveau logique selon un modèle de fautes.

### 3.3- Types de défauts physiques « fabrication »

Selon la surface atteinte par le mécanisme de défaillance, on distingue deux classes de défauts, les défauts globaux et les défauts locaux, et ce sont:

#### 3.3.1 Les défauts catastrophiques

L'ensemble de défauts catastrophiques inclut les circuits-ouverts, les courts-circuits ainsi que d'autres changements topologiques dans le circuit. Les défauts catastrophiques engendrent la déviation de toutes les spécifications et font que le circuit ne fonctionne pas. Ce type de défauts peut facilement être détecté par un test statique.

##### 3.3.1.1-Défaut de court-circuit

Les défauts de court-circuit comprennent tous les défauts et les mécanismes de défaillance qui provoquent des connexions électriques non souhaitées entre deux ou plusieurs nœuds du circuit. Les défauts de court-circuit non linéaires comprennent les courts circuits de l'oxyde de grille (fig 2a) et les courts circuits entre poly-silicium et oxyde de grille (fig. 2b). Les courts circuits résistifs se produisent dans les motifs défectueux des circuits intégrés qui laissent des ponts de métal ou de poly-silicium (fig 2b), dans une contamination ionique mobile, qui peut être concentrée dans une région particulière du circuit quand il est déformé, et dans certaines formes de court-circuit de l'oxyde de grille. Il existe plusieurs classes de défauts de court-circuit : court-circuit entre les nœuds du transistor (grille, drain et source), court-circuit à l'intérieur d'une porte logique et court-circuit entre deux portes logiques.

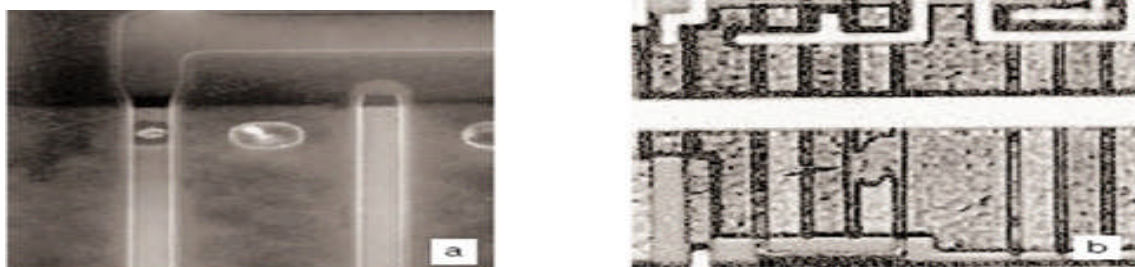


Figure 2. Exemples de défauts de court-circuit causant une surconsommation d' $I_{DDQ}$ , (a) Court-circuit d'oxyde de grille causant  $360\mu A$  d' $I_{DDQ}$  additionné et (b) Court-circuit entre poly-silicium et oxyde de grille causant  $5mA$  d' $I_{DDQ}$  additionné.

### 3.3.1.2- Défaut de circuit-ouvert

Les défauts de circuits ouverts sont des discontinuités électriques involontaires. Ils provoquent des comportements qui peuvent être difficiles à prévoir. Ces défauts comprennent des fils minces, des contacts mal formés (via), des fissures dans les siliciures, des contacts absents souvent causés par une gravure incorrecte, des trous d'oxyde, des vides dans les lignes métalliques (Figure 3) ou le poly-silicium résultant des spots lithographiques.

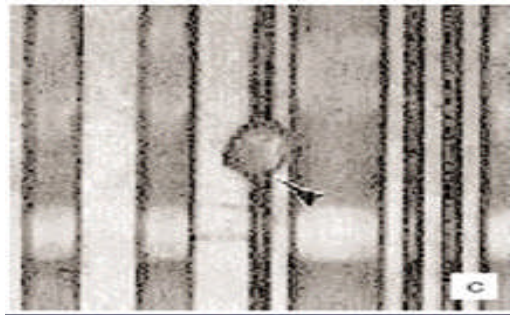


Figure 3. Exemple de circuit-ouvert du métal causant une surconsommation de  $5 \mu\text{A}$  d' $I_{DDQ}$ .

Les propriétés du défaut de circuit-ouvert dépendent principalement de la taille du défaut, de l'emplacement du défaut, de la structure locale électrique et de la variabilité du processus de fabrication. La figure 4 montre l'effet de l'emplacement du défaut lorsque le circuit-ouvert est dans la grille d'un seul transistor. Un circuit-ouvert dans un seul transistor permet une forte capacité de couplage entre le drain, la grille et la source. Il est à noter que le circuit-ouvert sur un seul transistor de la figure 4 induit une augmentation du courant de consommation ( $I_{DDQ}$ ) et est à 100 % détectable par la méthode de test  $I_{DDQ}$ . Le test en courant  $I_{DDQ}$  est basé sur la mesure du courant d'alimentation du circuit. Cependant, le circuit-ouvert sur le drain et le circuit-ouvert sur la source ont une probabilité de détection plutôt faible en utilisant des tests  $I_{DDQ}$ .

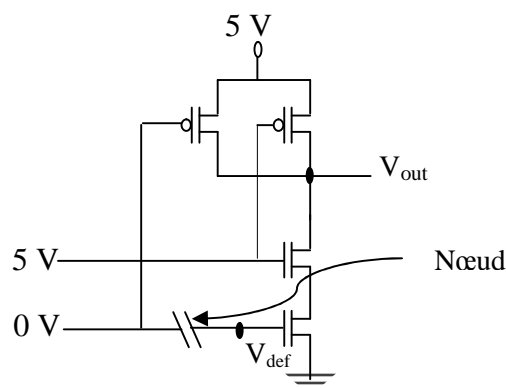


Figure 4. Circuit-ouvert sur la grille d'un transistor.

### **3.3.1.3-Défauts paramétriques**

Les défauts paramétriques représentent les fautes qui ne changent pas la topologie du circuit mais par contre elles engendrent la déviation des paramètres en dehors des spécifications (par exemple, une déviation de 10% de la valeur d'une résistance ou d'une capacité).

### **3.3.1.4-Défauts permanents**

Ce sont des défauts toujours présents tels que les défauts des collages.

### **3.3.1.5-Défauts temporaires**

Ce sont des défauts présents dans certaines conditions « température, tension,...etc. ».

## **3.4- Sources de défauts**

Le procédé de fabrication est généralement la cause principale de la présence des fautes, que ce soit après fabrication ou bien après une durée de vie du circuit. Les fautes peuvent être aussi provoquées par le design et sont sensées être corrigées après la vérification des prototypes. Les sources qui peuvent être à l'origine de l'apparition de certains défauts sont:

- 1 Les erreurs humaines
  - Une partie d'un métal peut être discontinu provoquant un circuit ouvert (fig.5a) ;
  - Les dopants peuvent ne pas être diffusés avec la bonne concentration (non uniformité) ou aux zones appropriées altérant ainsi les caractéristiques des dispositifs ;
  - Mauvais alignement des masques ou contamination de ces derniers avec des particules de poussières, cheveux ...etc.
  - Équipement défectueux
- 2 L'instabilité dans les conditions du procédé de fabrication (, en termes de changement de valeurs de n'importe quelle variable physique supposée constante.
- 3 L'instabilité du matériel, qui se rapporte à de petites variations dans les compositions chimiques utilisée dans la ligne du procédé de fabrication, comme par exemple, la contamination chimique venant des résidus d'un autre procédé de fabrication.
- 4 L'hétérogénéité du substrat, incluant les points défectueux, l'imperfection des surfaces, en particulier celles venant de l'implantation des ions.
- 5 Les points lithographiques, causés par des poussières dans les régions transparentes ou les rayures dans les régions opaques.
  - Les contacts et les via résistifs (fig.5b);
  - Deux fils métalliques séparés pouvant être court-circuités (fig.5c);
  - L'imperfection et le claquage de la couche SiO<sub>2</sub> pouvant causer une augmentation du bruit de la tension ou du courant;

- Les défauts d'oxyde de grille et les défauts dans les couches d'interconnexions.

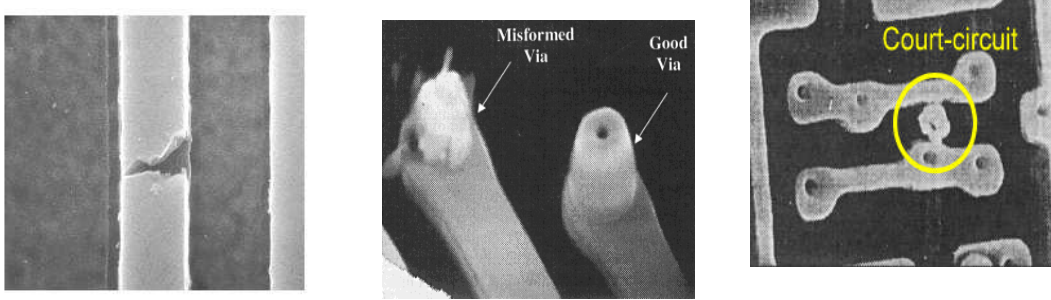


Figure 5 a)-b)-c). Défauts de fabrication

### 3.5- Classification et répartition de défauts

Les fréquences d'apparition de défauts dans certains circuits intégrés sont données dans les tableaux 1, 2 et 3.

Type de défautuosité	Fréquences d'apparition (%)
Court circuit	39
Circuit ouvert	14
Court circuit diffusion	14
Circuit ouvert diffusion	6
Court circuit Alu-Substrat	2
Pas de défautuosité observable	10

Tableau 1. Répartition des défautuosités dans les circuits nMos

Type de défautuosité	Fréquences d'apparition (%)
Court circuit	51
Circuit ouvert	1
Composant oublié	6
Mauvais composant	13
Composant mal placé	6
Déformation d'un conducteur	8
Mauvaises spécifications analogiques	5
Circuit logique défectueux	5
Autres	5

Tableau 2. Répartition des défautuosités dans les cartes imprimées (PCB)

Classes	Types de défectuosité
Très probable	Court circuit entre la grille et le drain Court circuit entre la grille et la source
Probable	Circuit ouvert sur le drain Circuit ouvert sur la source
Peu probable	Court circuit entre la grille et le substrat Grille flottante

Tableau 3. Classification des défauts catastrophiques pour les transistors MOS

### 3.6-Modèles de fautes « modèles de pannes »

Les modèles de fautes sont des circuits qui représentent le comportement électrique d'un défaut réel durant une simulation, permettant de réduire le nombre de défauts à simuler, la complexité du circuit à simuler et d'effectuer le test très tôt dans la phase de conception d'un circuit (réduire le TTM (Time To Market)). Pour développer une stratégie de test efficace en termes d'outils de simulation de fautes, de génération de vecteurs de test, ou de diagnostic, des modèles de fautes représentatifs des défauts réels sont nécessaires.

Les modèles couramment utilisés sont :

- Modèle des collages « stuck-at » le plus utilisé pour les circuits numériques.
- Modèle des collages au niveau transistor, toujours fermé/ouvert « stuck-on / open faults ».
- Modèle des court- circuits « Bridging Faults ».
- Modèle des circuits ouverts.
- Modèles pour les fautes de délai « « delay faults »
- Modèle de fautes pour les mémoires et pour les circuits configurables (FPGA, PLA...etc.).

#### 3.6.1-Le modèle de collage

Dans le cas du modèle des collages (fig.6) le défaut physique se traduit par un collage d'un signal du circuit soit à 0 (Vss) soit à 1 (Vdd). Ainsi, au niveau porte logique une ligne L avec un collage à 0(1) aura toujours l'état logique 0(1) quelle que soit la valeur logique imposée par la sortie de la porte liée à cette ligne. La condition de sensibilisation d'une telle panne est l'application d'une valeur logique opposée au collage sur le site de la panne.

Pour un signal X :  $X@0 = \text{collage à } 0 = Ca0$ ,  $X@1 = \text{collage à } 1 = Ca1$ .

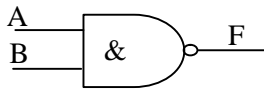
Autres notations :  $X_{S-a-0}$  et  $X_{S-a-1}$  ou  $X/0$  et  $X/1$ .

#### ✓ Avantages

- Indépendant de la technologie et facile à utiliser (simulation de fautes et génération de vecteurs de test) ;

- Peut être utilisé au niveau logique et au niveau module ;
- Un nombre de fautes raisonnable ( $2n$  fautes),  $n$  est le nombre de signaux dans le circuit,
- Il détecte de nombreux types de défauts fabrication (90% en technologie CMOS).
- ✓ **Inconvénient** : ce type de test ne détecte pas tous les défauts de fabrication.

**Porte logique : Nand**



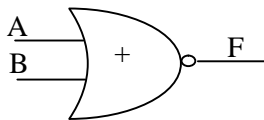
**Circuit Correct**

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

**Faute A@0**

A	B	F
0	0	1
0	1	1
1	0	1
1	1	1

**Porte logique : nor**



**Circuit Correct**

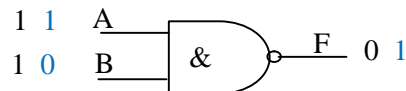
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

**Faute A@0**

A	B	F
0	0	1
0	1	0
1	0	1
1	1	0

Faute détectée

**Porte Nand sans fautes**



**Porte Nand avec collage A@0**

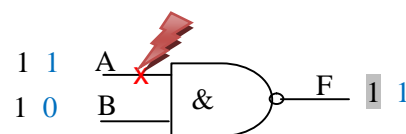


Figure 6. Modèle de collage.

**3.6.2- Modèle des collages multiples**

Le modèle de collage multiple est une extension du modèle de collage simple, dans ce modèle, plusieurs lignes peuvent être colées en même temps à 0 ou à 1.

✓ **Avantages**

- Ce type de modèle permet de détecter plus de fautes que le modèle du collage simple. Aussi, le modèle de collage multiple présente un intérêt théorique primordial pour les applications à très haute sûreté.

✓ **Inconvénients**

- Très grand nombre de fautes ( $3^n - 1$ ,  $n$ =nombre de signaux), mais aussi algorithmes de génération très complexes.

### 3.6.3 -Modèle des courts- circuits « Bridging Faults »

Il se traduit par un contact entre deux nœuds E et F (fig.7). Dans le cas de technologies TTL ou ECL, la fonction  $Z(x, y)$  réalisée par ce court-circuit peut être modélisé soit par un ET logique soit par un OU logique (fig.7) des valeurs logiques portées initialement par les deux équipotentielles. Par contre, la plupart des courts circuits dans les circuits actuels représentent une résistance significative entre les nœuds défectueux. Le modèle de défaut de court circuit résistif insère une résistance ( $R_{CC}$ ) entre les nœuds défaillants pour modéliser le court circuit de façon plus réaliste (fig.8).

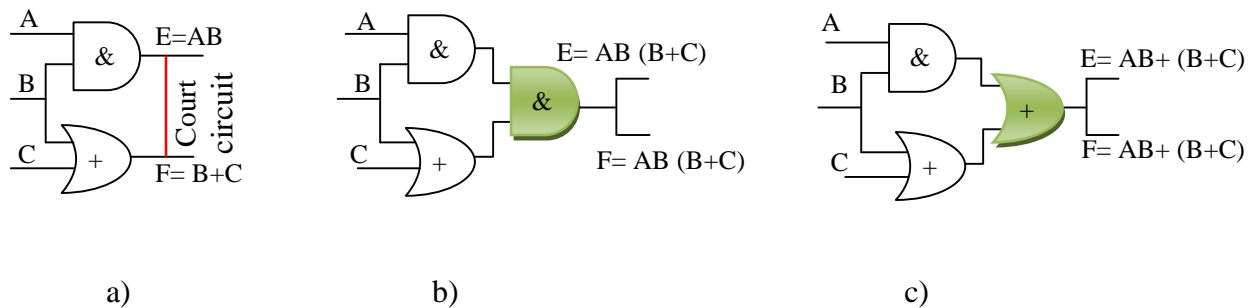


Figure 7. Modèles des courts- circuits. a) Court-circuit au niveau porte, b) Modèle logique (AND) et c) Modèle logique (OR)

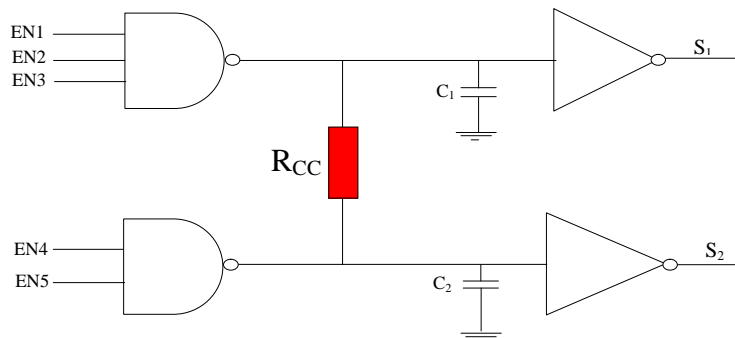


Figure 8. Modèle de court-circuit résistif

### 3.6.4 - Modèle des circuits-ouverts « Stuck Open Faults »

Le défaut physique se traduit par un circuit ouvert sur un nœud du circuit, le nœud coupé n'est connecté ni à Vdd ni à Vss « effet mémoire ». Des recherches ont classé le circuit-ouvert en forte ouverture ( $> 10 \text{ M}\Omega$ ) et en faible ouverture ( $\leq 10 \text{ M}\Omega$ ). Les fortes ouvertures causent les fautes de collage (stuck-at) qui peuvent être détectées par le modèle de collage. Les faibles

ouvertures engendrent les fautes de délais. Ces fautes ne peuvent pas être détectées par les modèles traditionnels de collage. Dans la technologie moderne submicronique, le pourcentage des ouvertures faibles est largement élevé. La figure 10 montre une ligne défectueuse qui peut être modélisée par un circuit-ouvert résistif ( $R_{CO}$ ).

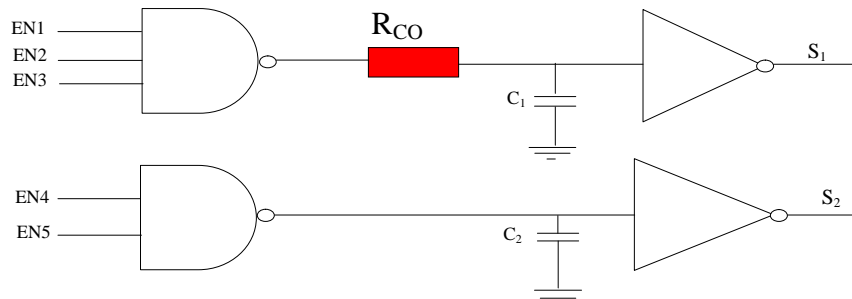


Figure 10. Modèle de circuit-ouvert résistif.

### 3.6.5 Modèle de fautes de délai

Dans le cas des fautes de délai, le défaut physique se traduit par un rallongement du temps de réponse du circuit. Les modèles traditionnels de défaut de court-circuit comprennent des modèles de fautes fonctionnelles ou des modèles de fautes de délai.

Les modèles les plus utilisés pour les fautes de délai sont les modèles des fautes de transition, les modèles de fautes de retard de porte et les modèles des fautes de délai sur les chemins.

#### 3.6.5.1-Modèle des fautes de transition

La faute de transition est engendrée lorsque le temps de transition est plus lent que les spécifications. Il existe deux temps de transitions, le temps de montée et le temps de descente. Pour chaque signal de sortie on distingue deux fautes: temps de montée plus lent « Slow-to-rise » et temps de descente plus lent « Slow-to-fall ».

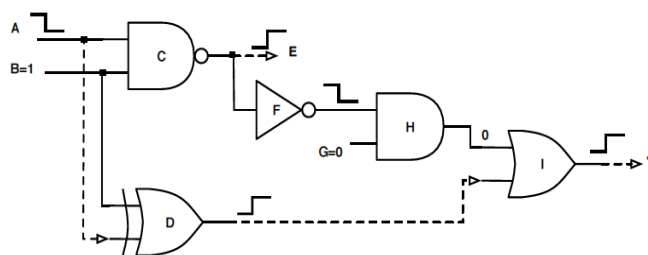


Figure 11. Faute de retard de transition. A-C-E: chemin court et A-D-I-J: chemin long.

### 3.6.5.2-Modèle de fautes de retard de porte

Dans le modèle de faute de retard de porte, le retard de propagation à travers la porte défectueuse est plus long que prévu. Cette défaillance est modélisée par une faute de type Lent à monter ou de type Lent à descendre suivant le type de commutation sur les entrées ou les sorties d'une porte du circuit. Une faute de retard de type temps de montée plus lent ou temps de descente plus lent affectant une porte logique caractérise un retard de propagation d'une transition montante ou descendante supérieure à la valeur spécifique.

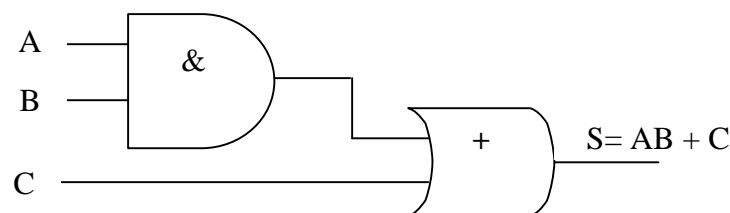
### 3.6.5.3-Modèle de fautes de délai de chemin

Une faute de retard de chemin se produit lorsque le temps de propagation entre les entrées et les sorties primaires est plus long que prévu. Une paire de vecteurs de test est nécessaire pour détecter les défauts de retard. Pour tester un défaut de retard il faudra lancer une transition au niveau des entrées qui va exciter la faute de collage, et de trouver un chemin de propagation entre l'endroit défectueux et une sortie primaire du circuit, et enfin de mesurer le temps de propagation de cette transition à la sortie. Le modèle de faute de retard de chemin détecte plus de fautes que le modèle des fautes de transition. Par contre, il test un nombre élevé de chemins et utilise des algorithmes de génération très complexes.

### 3.7-Détection de fautes

Le principe de la détection de fautes est de comparer la réponse en sortie du circuit fautif avec celle du circuit sain. En d'autres termes, soit un circuit combinatoire  $C$  avec la sortie  $S$ , une faute  $f$  engendrant un nouveau circuit  $C_f$  avec la sortie  $S_f$  et Un test  $t$ . On dit que le test  $t$  détecte la faute  $f$  si et seulement si la sortie  $S$  correspondant au test  $t$  (noté  $S(t)$ ) a des valeurs différentes dans le cas du circuit correct et du circuit fautif. Autrement dit,  $S(t) \neq S_f(t) \Leftrightarrow S(t) \oplus S_f(t) = 1$

#### Exemple :



L'entrée A est affectée par la faute  $f$  de collage à 0  $\Rightarrow f = A@0$

$$S = AB + C$$

$$S_f = 0 + C = C$$

$$S(t) \oplus S_f(t) = 1 \Leftrightarrow \bar{S} \cdot S_f + \overline{S_f} \cdot S = ABC = 1$$

$$\Leftrightarrow A = 1, B = 1 \text{ et } C = 0$$

Le vecteur de test qui permet de détecter la faute de collage est  $[ABC] = [110]$ .

# **Chapitre 4**

## **Simulation de fautes**

## 4.1- Introduction

Afin de déterminer la qualité d'une séquence de test T la simulation de fautes est utilisée (Figure 1). Cette qualité s'exprime sous la forme d'un taux de couverture de fautes. Ce dernier n'est directement significatif que pour l'ensemble des fautes prises en compte par le simulateur.

L'objectif de ce chapitre est d'introduire :

- La simulation de fautes
- Les algorithmes de simulation de fautes
  - Simulation série
  - Simulation parallèle
  - Simulation déductive
  - Simulation concurrente

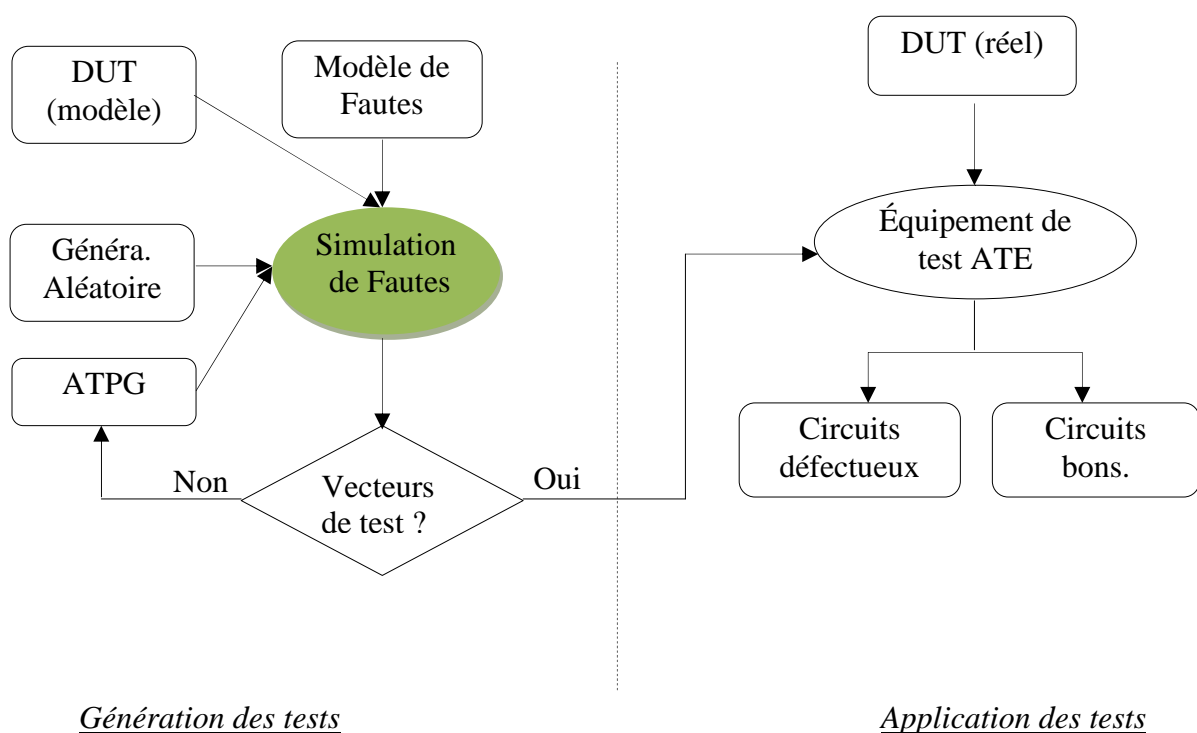


Figure 1. Test des circuits intégrés.

## 4.2 - Simulation de fautes

Elle permet de modéliser et de simuler le circuit à tester (DUT) en vue de vérifier les spécifications du circuit, d'évaluer d'autres alternatives de conception et de réduire le temps

de conception. La simulation joue un rôle important dans le processus de génération de vecteurs de test. Le premier type d'utilisation est donné à la figure 2. Si le résultat n'est pas satisfaisant on modifie la séquence T, soit en rajoutant des vecteurs pour améliorer le taux de couverture soit en supprimant des vecteurs qui ne contribuent plus à améliorer la séquence.

La simulation est aussi utilisée du fait que l'objectif principal de nombreux générateurs automatiques de vecteurs de test est de détecter une faute donnée. Une fois trouvé le vecteur de test correspondant à cette faute particulière, on effectue une simulation de fautes afin de supprimer de la liste des fautes à traiter celles détectées par ce nouveau vecteur.

Une dernière utilisation courante de la simulation de fautes consiste à analyser le comportement du circuit en présence de défauts

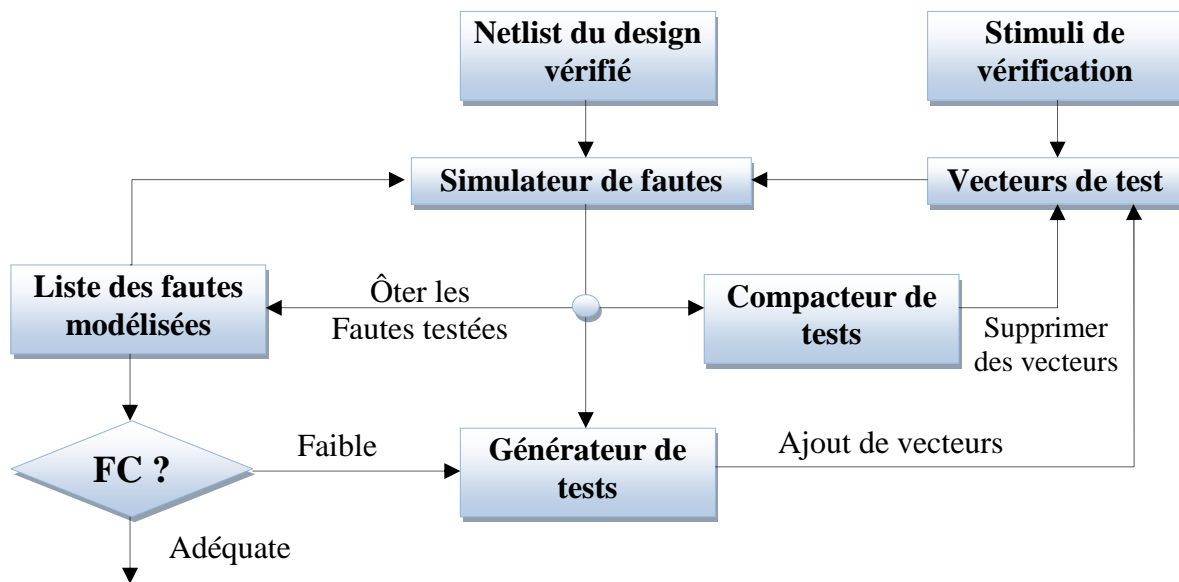


Figure 2. Principe de la simulation de fautes.

#### 4.3- Méthodologie de la simulation de fautes

La simulation se fait avec des simulateurs de circuits analogiques de type SPICE, SPECTRE, etc (fig.3). Elle comprend une phase de simulation du circuit sans fautes, une phase d'injection de fautes dans le circuit, une phase de simulation du circuit avec des fautes et enfin une phase de comparaison des résultats des deux simulations. Un simulateur se caractérise par son efficacité, son temps d'exécution, et par la mémoire utilisée.

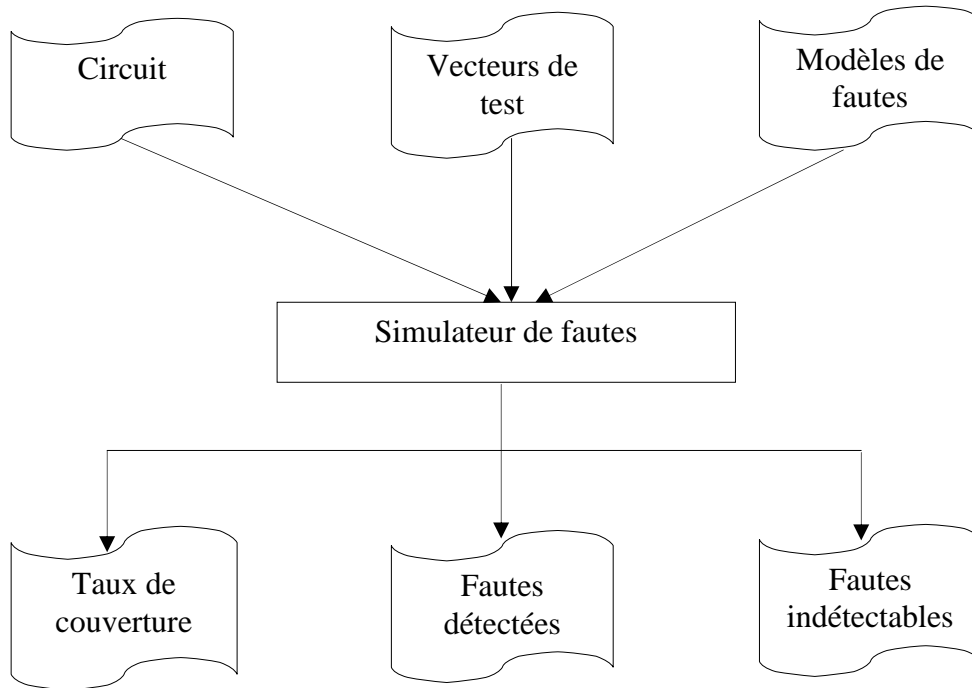


Figure 3. Méthodologie de la simulation de fautes

#### 4.4- Algorithmes de simulation de fautes

Leur but est de déterminer toutes les fautes détectables par chaque vecteur de test. Le principe est de simuler le circuit correct et les circuits fautifs et comparer les résultats sur les sorties primaires du circuit, autrement dit la simulation du circuit correct, la sauvegarde des résultats obtenus, injection de la faute, puis simulation du circuit fautif et enfin comparaison des résultats des deux simulations.

Il existe plusieurs algorithmes de simulation de fautes à savoir :

##### 4.4.1- Simulation de fautes série

C'est la méthode de simulation de fautes la plus simple, elle consiste à simuler le circuit correct  $C$ , transformer le circuit correct  $C$  en  $C_f$  en injectant la faute  $f$  et simuler le circuit fautif  $C_f$  avec le simulateur logique. Cependant, elle nécessite des temps de traitement considérables « CPU » (simulation d'une seule faute à la fois).

##### 4.4.2- Simulation de fautes parallèle

Dans la simulation de fautes parallèle, on simule le circuit correct et  $N$  fautes simultanément.

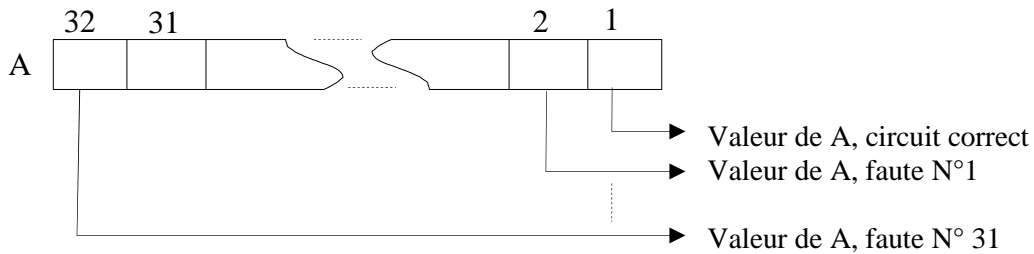
- Les valeurs des signaux du circuit correct et des  $N$  circuits fautifs sont stockées en mémoire de l'ordinateur.
- Si  $W$  est la taille de la mémoire, alors  $N=W-1$

- Si  $N_f$  est le nombre de fautes, il faut  $N_f / (W-1)$  simulations
- L'insertion des fautes est effectuée avec des masques.

Pour chaque signal S du circuit on définit deux masques :

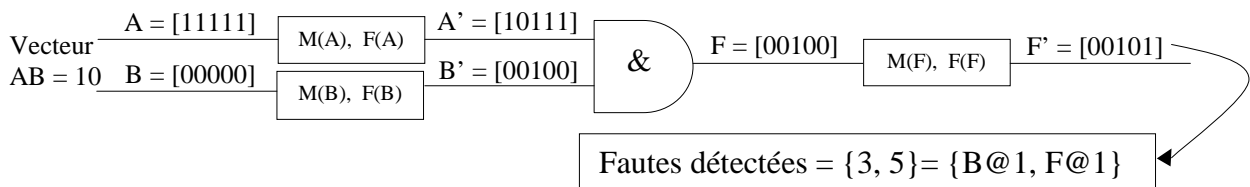
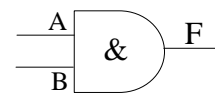
- Un masque  $M(S)$  qui indique la position du bit modélisant la faute  
 $M(S)_i = 1$  si le  $i^{\text{ème}}$  bit représente une faute du signal S, 0 Sinon
- Un masque  $F(S)$  qui indique le type de collage  
 $F(S)_i = 1$  si la faute est  $S@1$ , 0 si la faute est  $S@0$ .

Le nouveau signal :  $S' = S.\overline{M(S)} + M(S).F(S)$



Soit l'exemple d'injection de fautes appliqué pour la porte AND suivant :

Position	Fautes	Signal S	M (S)	F (S)
1(MSB)	C.Correct	A	01000	00000
2	A@0	B	00100	00100
3	B@1	F	00011	00001
4	F@0			
5(LSB)	F@1			

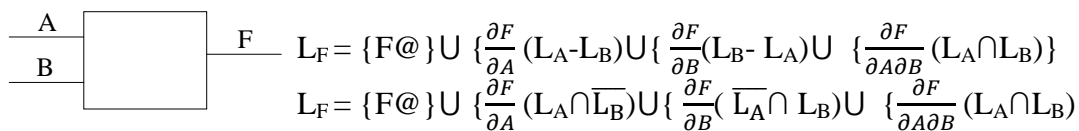


### 4.4.3- Simulation de fautes déductive « DFS »

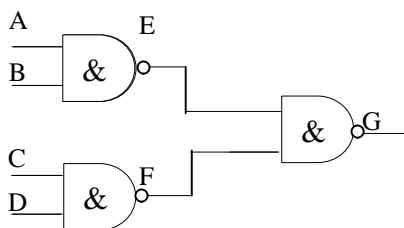
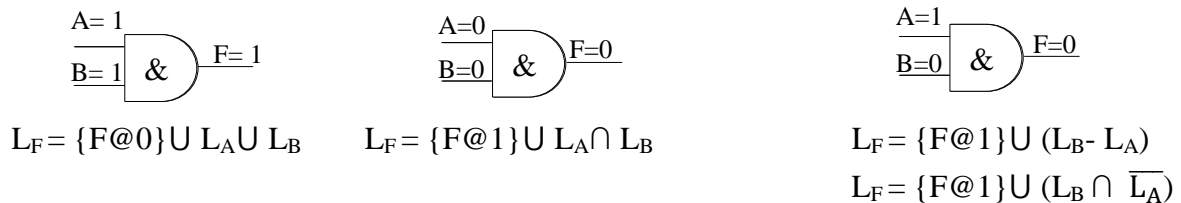
Elle consiste à simuler uniquement le comportement du circuit correct et en déduire toutes les fautes détectables en même temps et donc on a besoin d'effectuer uniquement une seule passe de simulation pour chaque vecteur de test appliqué. On associe à chaque signal S, une valeur logique, une liste de fautes notée  $L_S$ . Puis on effectue une seule simulation pour chaque vecteur de test (propagation des valeurs logiques « simulation logique » et propagation des listes de fautes). Si le signal S est une entrée primaire, alors on a :  $L_S = \{S@0\}$  si  $S = 1$ , et  $L_S = \{S@1\}$  si  $S = 0$ . Les listes de fautes des sorties primaires contiennent les fautes détectées par le vecteur de test.

Des travaux ont prouvé expérimentalement que la simulation déductive est plus rapide que celle parallèle seulement pour les circuits dont le nombre de portes est supérieur à 500.

#### -Propagation des listes de fautes



#### -Exemples :



8 fautes réduites					
A@0	A@1	E@0	×	G@0	×
×	B@1				
C@0	C@1	×	×	G@0	×
×	D@1				

1- ABCD = 1100 E=0 F=1 G=1

$$L_A = \{A@0\} ; L_B = \{ \}$$

$$L_C = \{C@1\} ; L_D = \{D@1\}$$

$$LE = \{\} \cup LA \cup LB = \{A@0\}$$

$$LF = \{\} \cup (LC \cup LD) = \{\}$$

$$LG = \{G@0\} \cup (LE - LF) = \{G@0, A@0\}$$

## 2- ABCD = 1010 E=1 F=1 G=0

$$LA = \{A@0\} ; LB = \{B@1\}$$

$$LC = \{C@0\} ; LD = \{D@1\}$$

$$LE = \{E@0\} \cup (LB - LA) = \{E@0, B@1\}$$

$$LF = \{\} \cup (LD - LC) = \{D@1\}$$

$$LG = \{\} \cup LE \cup LF = \{E@0, B@1, D@1\}$$

### 4.4.4- Simulation de fautes concurrente

C'est la méthode de simulation la plus générale, elle est rapide car elle est basée sur la simulation simultanée de plusieurs portes fautives. Cependant elle est complexe et demande beaucoup d'espace mémoire que la simulation déductive en début de simulation.

# **Chapitre 5**

## **Génération Automatique des Vecteurs de Test (ATPG)**

## 5.1- Introduction

Une fois le modèle de fautes choisi, il faut déterminer les vecteurs d'entrée qui rendent une faute contrôlable et observable.

L'objectif de ce chapitre est d'introduire :

- La génération de vecteurs de test ;
- Les types de génération de vecteurs de test.

## 5.2- La génération de vecteurs de test

Un vecteur de test est une combinaison des entrées du circuit sous test DUT qui permet de contrôler les fautes à partir des entrées primaires et de les observer sur les sorties primaires. La génération de vecteurs de test consiste à définir un programme qui permet de décrire les vecteurs de test spécifiques pour détecter l'ensemble des fautes données dans le but d'avoir une qualité du test et un coût d'application du test optimums. La Figure 1 montre un schéma général résumant la génération et l'application des vecteurs de test.

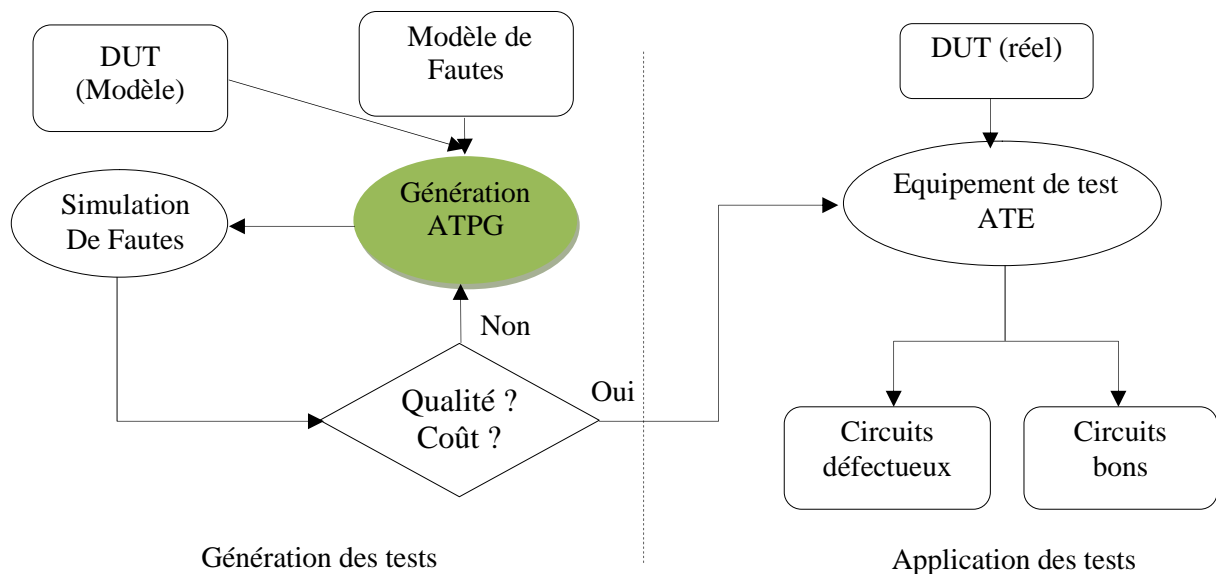


Figure 1. Génération et application des vecteurs de test

### **5.3- Les types de génération de vecteurs de test**

Il existe quatre types de génération de vecteurs de test :

#### **5.3.1- La génération manuelle des vecteurs de test**

Les vecteurs de test sont écrits manuellement par l'ingénieur de test. Elle utilise un simulateur de fautes et ne nécessite pas d'algorithme de génération, cependant, il est très difficile d'atteindre un taux de couverture élevé de cette façon.

#### **5.3.2- La génération pseudo aléatoire des vecteurs de test**

Dans ce cas là, les vecteurs de test sont choisis aléatoirement, en utilisant un simulateur de fautes pour déterminer le taux de couverture, mais elle ne détecte pas les fautes difficiles, ce qui fait que ces techniques de génération ne trouvent pas encore un intérêt industriel important.

#### **5.3.3- Génération exhaustive des vecteurs de test**

Ce type de génération, applique tous les vecteurs de test possibles, sans avoir à utiliser la simulation de fautes (pas de modèle de fautes). Elle détecte toutes les fautes détectables de tous les modèles de fautes. Cependant elle présente un test très couteux (grand nombre de vecteurs) et s'avère aussi être inutilisable pour les circuits avec un grand nombre d'entrées. A noter que, des techniques de partitionnement du circuit pour réduire le nombre de vecteurs et la génération localement exhaustive peuvent améliorer une telle génération.

#### **5.3.4- Génération automatique ATPG « Automatic Test Pattern Generation ».**

Dans ce cas là, l'ATPG utilise un simulateur de fautes et des vecteurs de test déterministes, pour chaque faute non détectée un vecteur de test qui la détecte est généré. Il permet d'atteindre un taux de couverture maximum avec un coût minimum. Prenons l'exemple, du circuit ALU 74181-14 entrées. L'ATPG détecte 100% des fautes modélisées, en nécessitant seulement 47 vecteurs de test. Cependant, malgré les avantages que présente l'ATPG, il utilise des algorithmes très complexes qui nécessitent donc des temps de traitement considérables.

La figure 2 montre le principe de base de la génération automatique des vecteurs de test.

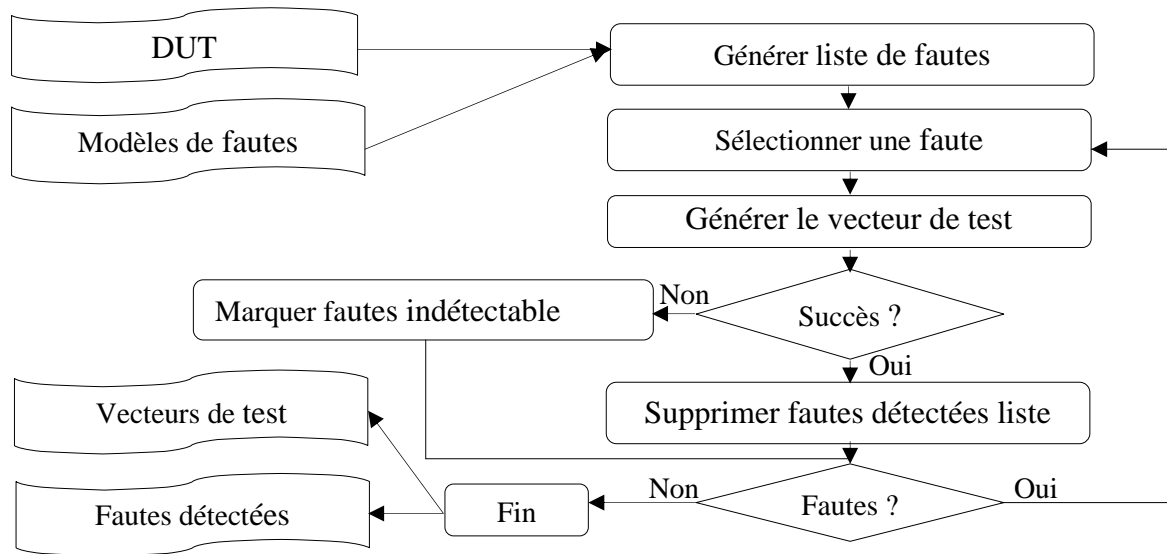


Figure 2. Génération automatique « ATPG ».

#### 5.4- Détection de fautes

La recherche d'un vecteur de test T qui détecte la faute F revient à :

- Trouver un vecteur de test T qui :
  - 1- Active la faute F à partir des entrées primaires (PIs).
  - 2- Propage la faute F jusqu'aux sorties primaires (POs).

#### Exemple : F= S @0

Le vecteur qui détecte la faute S @0 doit :

- Imposer la valeur 1 au signal S
- Propager la faute X =1 jusqu'aux sorties primaires (POs).

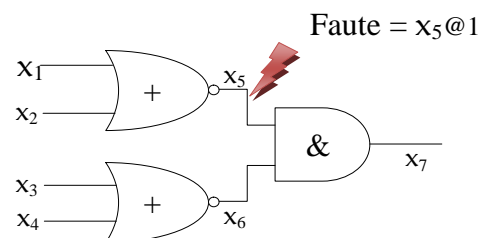
Le changement de la valeur du signal S de 1 vers 0 doit être visible sur au moins une des sorties primaires POs du circuit.

#### Activation des fautes

Son principe est de ramener le circuit dans l'état ou la faute produit une erreur dans le circuit. Pour le modèle de collage, c'est ramener la valeur opposée du collage sur le signal en question.

#### Exemple : F = X<sub>5</sub>@1

Activation :  $(X_5 = 0) \Rightarrow (x_1x_2 = 11)$   
 Ou  $(x_1x_2 = 01)$  Ou  $(x_1x_2 = 10)$



**Propagation des fautes :** c'est le processus qui permet de propager la faute à travers toutes les portes du circuit jusqu'aux sorties primaires.

$$F = X_5@1$$

$$\text{Faute} = x_5@1 \quad (x_5 = 0)$$

$$\text{Propagation} \Rightarrow (x_6 = 1) \Rightarrow (x_3x_4 = 00).$$

**Détection de fautes = Activation + propagation**

$$F = X_5@1$$

$$\text{Détection} \Rightarrow x_1x_2x_3x_4 = 1100$$

$$\text{Ou } x_1x_2x_3x_4 = 0100$$

$$\text{Ou } x_1x_2x_3x_4 = 1000$$

### 5.5- algorithmes de génération automatique

Les algorithmes de génération automatique de vecteurs de test (ATPG) les plus utilisés sont :

- Algorithme D
- Algorithme PODEM
- Algorithme FAN
- Autres algorithmes

#### 5.5.1- Algorithme D

L'algorithme D est le premier algorithme complet développé par IBM en 1966, il génère un test pour détecter une faute présente dans le circuit. Pour un signal S on :

$$S = D \Leftrightarrow S = 1 \text{ dans le bon circuit et } S = 0 \text{ dans le circuit défectueux.}$$

$$S = \bar{D} \Leftrightarrow S = 0 \text{ dans le bon circuit et } S = 1 \text{ dans le circuit défectueux.}$$

L'algorithme D procède en trois phases :

##### 5.5.1.1-Phase de « setup »

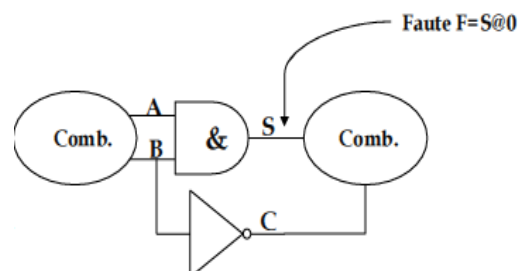
C'est la phase pendant laquelle la faute (signal D) est créée. Elle ramène le signal fautif à la valeur opposée du collage en assignant les entrées de la porte.

**Exemple : F = S@0**

**Setup :**

Affectation :

$$\Rightarrow A = 1 ; B = 1$$



Implication :

$\Rightarrow S = D ; C = 0 ; \dots$

### 5.5.1.2-Phase de propagation

C'est le processus qui permet de propager la faute (signal D) jusqu'aux sorties primaires.

#### **Exemple : $F = S @ 0$**

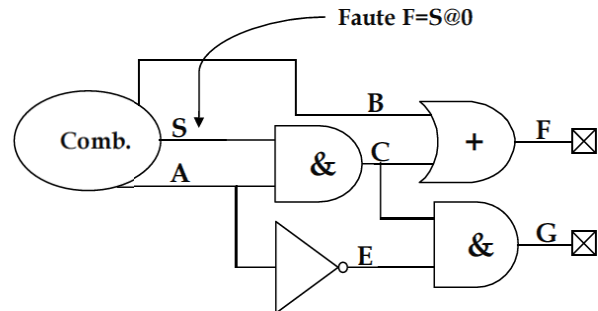
Propagation :  $(S = D)$

Affec. 1 :  $A = 1$

Impl.1 :  $C = D ; E = 0 ; G = 0$

Affec.2 :  $B = 0$

Impl.2 :  $F = D$  (fin de propagation)



#### **Algorithme de propagation:**

Procédure propager (S: signal; V= D ou D')

Begin

if (S est une sortie primaire)

return ;

else

trouver une porte P qui a le signal S comme entrée et U en sortie

trouver une combinaison des entrées de P propageant V

simuler le circuit // phase d'implication

propager (P, V= D ou D')

pour chaque signal d'entrée  $S_i$  de la porte P

justifier ( $S_i, V_i$ )

end if

### 5.5.1.3-Phase de justification

C'est le processus qui permet de remonter jusqu'aux entrées primaires à partir des signaux imposés par les deux premières phases.

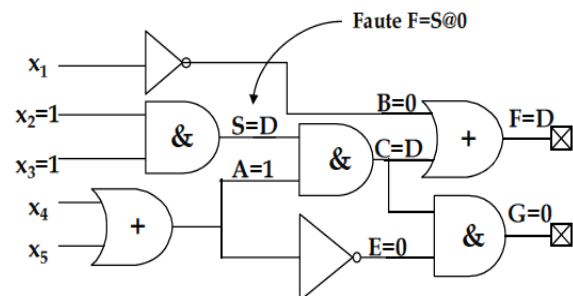
#### **Exemple : $F = S @ 0$**

Justification

$A = 1 \Rightarrow x_4 = 1$  et  $x_5 = x$

ou  $x_5 = 1$  et  $x_4 = x$

$B = 0 \Rightarrow x_1 = 1$



#### **Algorithme de justification**

Procédure justifier (S : signal ; V= 0 ou 1)

```

Begin
  If (S est une entrée primaire)
    return ;
  else
    trouver la porte P qui a le signal S comme sortie
    trouver une combinaison des entrées de P permettant d'avoir la valeur V en sortie de la
    porte
    simuler le circuit // phase d'implication
    pour chaque signal d'entrée Si de la porte
      justifier (Si, Vi)

  end if
end

```

- **Reconvergence des signaux**

Les conflits dans les phases de propagation et de justification sont créés lorsque la reconvergence des signaux de sortie des portes ont un fanout > 1.

**Exemple**

Setup:

$S = 1 \Rightarrow X_2 = 1$  et  $X_3 = 1$

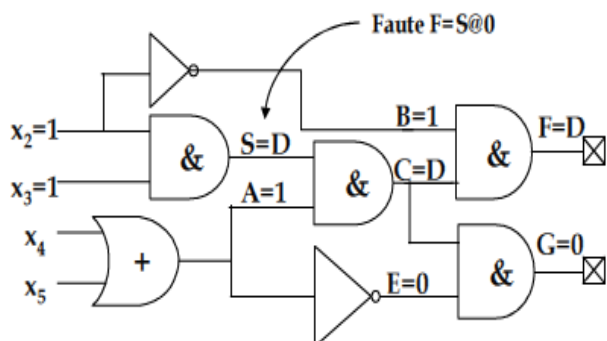
Propagation

$A = 1 ; B = 1$

Justification

$A = 1 \Rightarrow X_4 = 1$  ou  $X_5 = 1$

$B = 1 \Rightarrow X_2 = 0$  (conflit)



**5.5.2- Algorithme PODEM**

Il a été développé pour améliorer l'algorithme D. Il permet d'éviter le nombre très élevé de justification créant des problèmes de reconvergence des signaux dans l'algorithme D. Il procède en trois phases :

**Phase 1 : détermination de l'objectif (procédure objectif())**

Elle représente le processus qui sert à ramener d'abord la valeur de test (D ou D') à la sortie de porte fautive, et enfin propager la valeur de test de l'entrée à la sortie de la porte.

**Procédure objectif ()**

Algorithme de la procédure Objectif () :

Procédure objectif () /\* Faute = S@v)\*/

begin

```

if (la valeur de S est x)
    return (S, v') ;
else
    trouver toutes les portes qui ont D ou D' en entrée et x en sortie
    choisir la porte P la plus proche des sorties primaires
    choisir une entrée  $E_i = x$  de la porte P
        soit  $c_i$  la valeur qu'on veut assigner à l'entrée  $E_i$ 
    return (E, c)
end if

end

```

**Phase 2 :** Remontée (Backtrace) jusqu'aux entrées primaires qui remplissent l'objectif défini.

**Procédure Backtrace ()**

Algorithme de la procédure Backtrace () :

```

Procédure Backtrace (E, c) /*Faute =S@v*/
Begin
    If (E est une entrée primaire)
        return (E, c);
    else
        choisir une combinaison des entrées de la porte E permettant
        d'avoir la valeur c en sortie
        pour la combinaison choisie, choisir une entrée  $E_i$  de la porte E
        soit  $c_i$  la valeur qu'on veut assigner à l'entrée  $E_i$ 
        return (Backtrace ( $E_i, c_i$ ))
    end if
end

```

**Phase 3 :** Simulation (implication) de la valeur de l'entrée assignée.

Durant cette phase s'il y a toujours un chemin possible, on continue avec un autre objectif sinon, on simule la valeur opposée de l'entrée.

**Procédure PODEM ()**

Algorithme simulation ()

```

Begin
    If (une des sorties primaires est à D ou D')
        return succès ;

```

```

while (il existe encore un chemin)
    (S, v) = Objectif ()
    (E, c) = backtrace (S, v)
    simuler la valeur c sur l'entrée primaire E
if (PODEM() = succès) return success
simuler la valeur c' sur l'entrée primaire E
if (PODEM() = succès) return succès
simuler la valeur x sur l'entrée primaire E
return echec
end while
end

```

### Procédure PODEM

**Exemple : Faute = S@0**

Obj 1 : S= 1

Bac 1 : x2= 1

Sim : S= X ; F= X ; G= X

Bac 2: x3= 1

Sim: S=1; F=X; G= X

Obj2: C= D/D'

Bac 1: A= 1

Bac 2: x4= 1

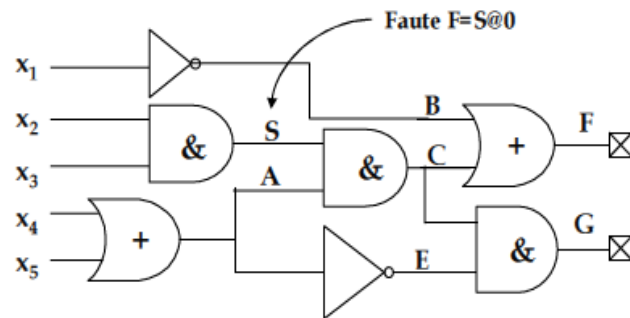
Sim: C= D; F= X; G= 0

Obj3: F= D/D'

Bac 1: B: 0

Bac 2: x1= 1

Sim: F= D; G=0



### 5.5.3- Algorithme FAN

Il a été développé en vue d'améliorer les algorithmes D et PODEM qui nécessitent un nombre élevé de backtraces dû au problème de reconvergence dans les circuits.

# Conclusion générale

Ce cours est la version améliorée de ce qui a été présenté et mis à la disposition des étudiants Master 1 de spécialité « microélectronique » durant ces deux dernières années successives au département d'électronique de l'Université Mouloud Mammeri.

Le test des circuits intégrés est une discipline des sciences modernes. Il repose sur les sciences électroniques et informatiques sur le plan théorique. Mais aussi sur l'utilisation des moyens pratiques performants rendus possibles par l'évolution d'un ensemble important d'outils de simulation et d'injection de fautes commercialisés.

Les étudiants trouveront les outils nécessaires à la maîtrise de divers concepts utilisés dans le test des circuits intégrés. La mise en pratique de ces derniers est présentée par une série d'exemples.

## Références bibliographiques

- [Arm72] D.B. Armstrong, "A Deductive Method for Simulating Faults in Logic Circuits IEEE Trans. On Computers, Vol. C-21, no 5, May 1972, pp. 464-471.
- [Cha74] Chang et al, "Comparison of parallel and deductive fault simulation methods"
- [Haw94] C.F. Hawkins et al, "Defect classes - an overdue paradigm for CMOS IC testing", ITC, p. 413, 1994.
- [Her94] K. Heragu, V.D Agrawal, M.L Bushnell, "FACTS: fault coverage estimation by test vector sampling", 12th IEEE VLSI Test Symposium, 25-28 April 1994, pp. 266 –271.
- [Hit82] R. B. Hitchcock, G. L. Smith, and D. D. Cheng, "Timing analysis of computer hardware", IBM Journal Research and Development, vol. 26, no. 1, pp. 100–105, January 1982.
- [Kho07] A. Khouas, "Test de systèmes électroniques" fascicule de cours, École polytechnique Montréal, Automne 2007.
- [Kir88] T. Kirkland, M.R. Mercer, "Algorithms for Automatic Test-Pattern Generation" IEEE Design & Test of Computers, Vol. 5, Issue 3, June 1988, pp.: 43-55.
- [Li89] W. N. Li, S. M. Reddy, and S. K. Sahni, "On path selection in combinational logic circuits," IEEE Trans. on Computer-Aided Design, vol. 8, no. 1, pp. 56–63, January 1989.
- [Li03] Z. Li, X. Lu, W. Qiu, W. Shi, and D. M. H. Walker, "A circuit level fault model for resistive opens and bridges," VLSI Test Symposium, pp. 379–384, 2003.
- [Li03a] Z. Li, X. Lu, W. Qiu, W. Shi, and D.M.H. Walker, "A Circuit Level Fault Model for Resistive Bridges", ACM Transactions on Design Automation of Electronic Systems, Vol. 8, No. 4, October 2003.
- [Mac08] A.Machouat, "Développement et application d'une méthode d'analyse de défaillances fonctionnelles et contribution à l'amélioration de l'utilisation des techniques optiques statiques et dynamiques ", thèse de doctorat, Université Bordeaux I, 2008.
- [Mon02] R.Rodriguez-Montanes and J. P. de Gyvez, "Resistance characterization for weak open defects", IEEE Design and Test, Sept.2002, pp: 18-25.
- [Moo00] W. Moore, G. Gronthoud, K. Baker and M.Lousberg, Delay-fault testing and defects in sub-micron Ics-does critical resistance really mean, anything, ITC,2000,95-104.
- [lau05] M. Laurent Capocchi, "Simulation concurrente de fautes comportementales pour des systèmes à événements discrets", thèse de doctorat, Université de Corse, Pasquale Paoli, 2005
- [Mew78] J.Mewon and Chapell, "Deductive fault simulation with functional blocks", IEEE Trans Comput vol.c27, pp 689-695, 1978.
- [Nor89] F.E. Norrad and H. Raskard, "An automatic test generation algorithm for hardware description languages", 26 th ACM/IEEE design autom. conf. 1989, pp 429-434.
- [Ulr74] E.G. Ulrich ; T. Baker, "Concurrent Simulation of Nearly Identical Digital Networks Computers", Vol. 7, April 1974, pp. 39-44.
- [Wil86] T.W. Williams, "VLSI Testing", North-Holland, 1986.