

# Test des circuits intégrés VLSI

version 1.0

# Plan

---

- **Les enjeux du test**
- **Test Paramétrique**
- **Test fonctionnel et test structurel**
- **Le modèle des collages et la simulation de fautes**
- **La génération automatique de vecteurs de test**
- **Test des circuits séquentiels : le scan-path**
- **Techniques d'auto-test**

# Rendement de fabrication

---

$$\rho = \frac{\text{Nombre de puces sans défauts}}{\text{Nombre de puces total sur une tranche}}$$

Les principales causes de défauts physiques:

- poussières
- défauts cristallins
- désalignements de masques

Le rendement décroît exponentiellement avec la surface de la puce

$$\rho \propto \exp(-kS)$$

# Objectifs du test de production

---

Il s'agit de faire le tri entre les puces fonctionnelles et les puces défectueuses, pour éviter de monter une puce défectueuse dans un équipement :

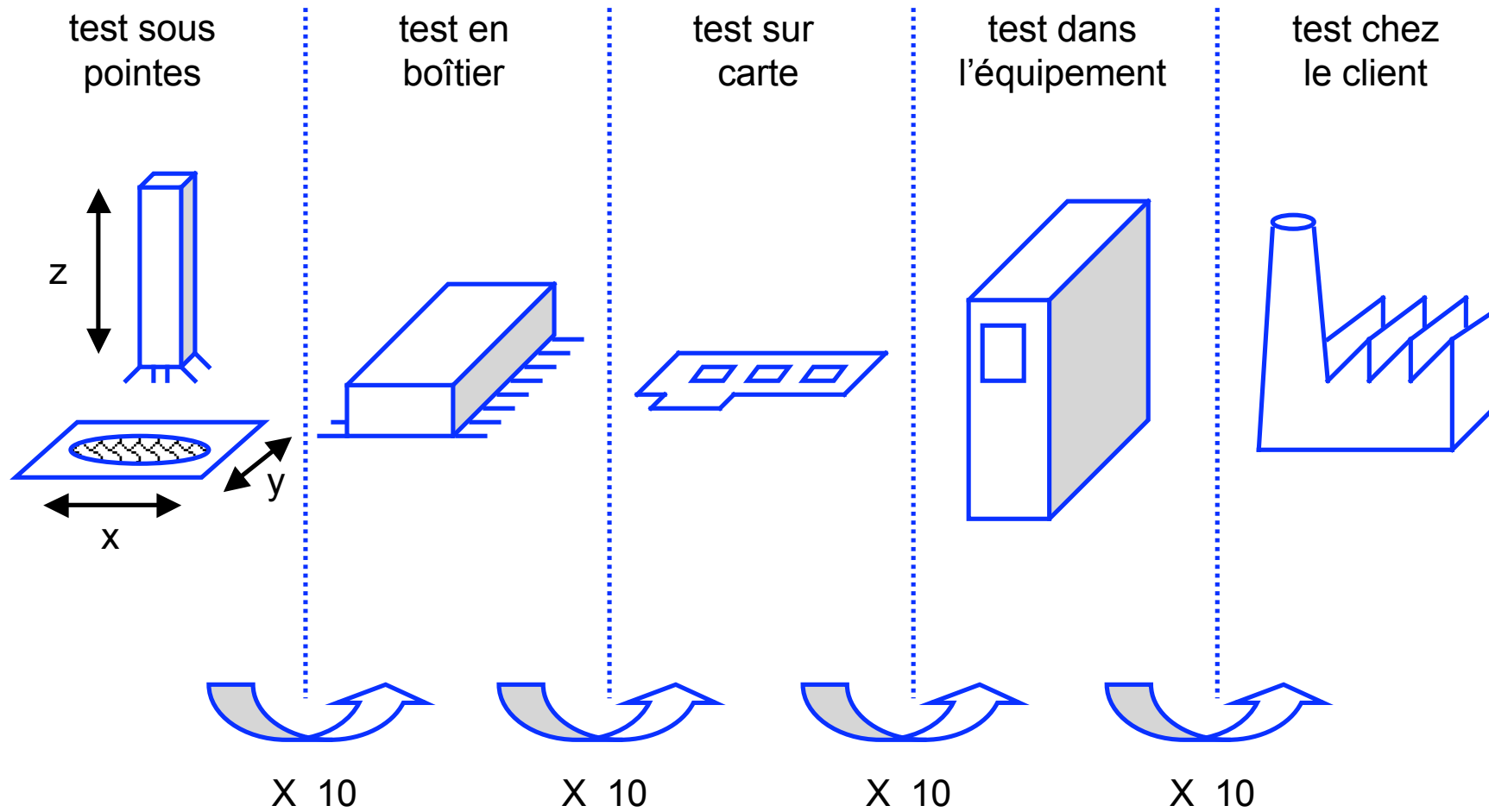
- ⇒ on ne cherche pas à détecter d'éventuelles erreurs de conception
- ⇒ on ne cherche pas à identifier la cause du dysfonctionnement

On parle de test GO / NOGO

Les deux principaux objectifs sont donc :

- ⇒ minimiser le nombre de pièces défectueuses non détectées
- ⇒ minimiser la durée du test (quelques secondes au plus par pièce).

# Le coût du test



# Plan

---

- Les enjeux du test
- **Test Paramétrique**
- Test fonctionnel et test structurel
- Le modèle des collages et la simulation de fautes
- La génération automatique de vecteurs de test
- Test des circuits séquentiels : le scan-path
- Techniques d'auto-test

# Test paramétrique

---

- Le but du test paramétrique est d'éliminer les tranches globalement défectueuses. Il est indépendant de la fonctionnalité des puces se trouvant sur la tranche.

- **Wafer Fab Yield** = 
$$\frac{\text{Nombre de tranches livrées}}{\text{Nombre de tranches lancées en fabrication}}$$

- Les principales causes de tranches perdues
  - Tranches cassées dans un équipement ou lors de transfert entre équipements
  - Tranches non conformes avec des paramètres électriques hors spécification

# Objectifs du test paramétrique

---

- Il s'agit de faire le tri entre les tranches conformes et les tranches non conformes : On cherche à détecter des erreurs lors de la fabrication ou des dérives lentes de certains paramètres
- Les deux objectifs sont donc:
  - Eviter de livrer des tranches et donc des puces « non conformes » qui pourraient ne pas être détectées comme défectueuses au test sous pointes.
  - Obtenir le maximum d'informations sur des paramètres électriques mesurables, pour identifier l'origine du problème et réagir le plus rapidement possible.



# Contenu du test paramétrique

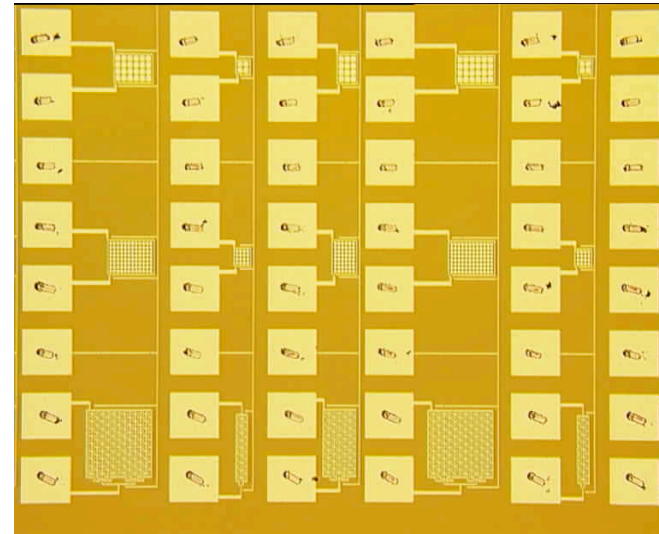
---

- Paramètres élémentaires

- Résistivité des couches Métal, N+, P+, Polysilicium...
- Résistance de contact
- Epaisseur d'oxyde
- Isolation entre niveaux
- Peigne inter-digités

- Structures fonctionnelles

- Transistor
- Diode
- Capacité
- Résistance



# Plan

---

- Les enjeux du test
- Test Paramétrique
- **Test fonctionnel et test structurel**
- Le modèle des collages et la simulation de fautes
- La génération automatique de vecteurs de test
- Test des circuits séquentiels : le scan-path
- Techniques d'auto-test

# Taux de couverture

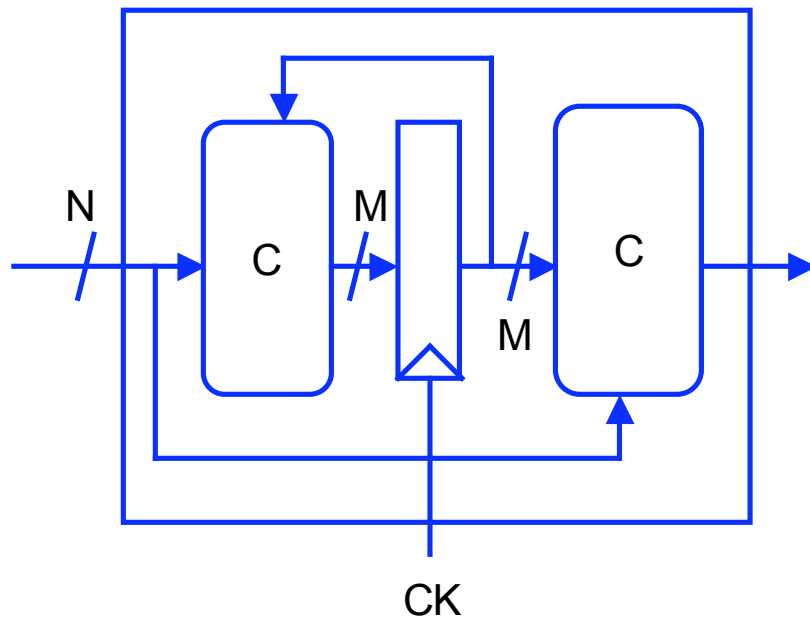
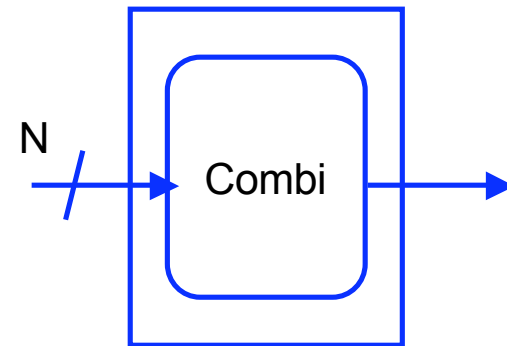
---

- ❑ Un **vecteur de test** est un couple, constitué par :
  - un ensemble de valeurs imposées sur les signaux d'entrée
  - un ensemble de valeurs attendues sur les signaux de sortie
  
- ❑ Un **jeu de vecteurs**  $\{V\}$  définit le filtre qui permet d'éliminer les puces défectueuses.
  
- ❑ Le **taux de couverture** est la métrique qui permet de mesurer l'efficacité d'un jeu de vecteurs particulier pour un circuit particulier.

La définition de cette métrique est un problème difficile !!!

# Test fonctionnel

Pour un circuit **combinatoire** possédant  $N$  entrées, le test fonctionnel exhaustif suppose d'appliquer  $2^N$  valeurs sur les entrées.



Pour un circuit **séquentiel** possédant  $N$  entrées et  $M$  bits mémorisés dans des registres internes, le test fonctionnel exhaustif suppose d'appliquer  $2^{N+M}$  vecteurs de test.

# Test structurel

---

- Le nombre de vecteurs de test augmente de façon **exponentielle** ( $2^{N+M}$ ) avec la complexité des circuits.
- Le nombre de composants défectueux possibles augmente **linéairement** avec la complexité des circuits
- On « oublie » donc la fonctionnalité du circuit, pour définir comme suit le **taux de couverture structurel** d'un jeu de vecteurs  $\{V\}$  :

$$\tau = \frac{\text{nombre de défauts détectés par au moins un vecteur de } \{V\}}{\text{nombre de défauts total du circuit.}}$$

⇒ Il faut donc définir un **modèle de défauts**, permettant d'**énumérer** tous les défauts possibles d'un circuit...

# Plan

---

- Les enjeux du test
- Test Paramétrique
- Test fonctionnel et test structurel
- **Le modèle des collages et la simulation de fautes**
- La génération automatique de vecteurs de test
- Test des circuits séquentiels : le scan-path
- Techniques d'auto-test

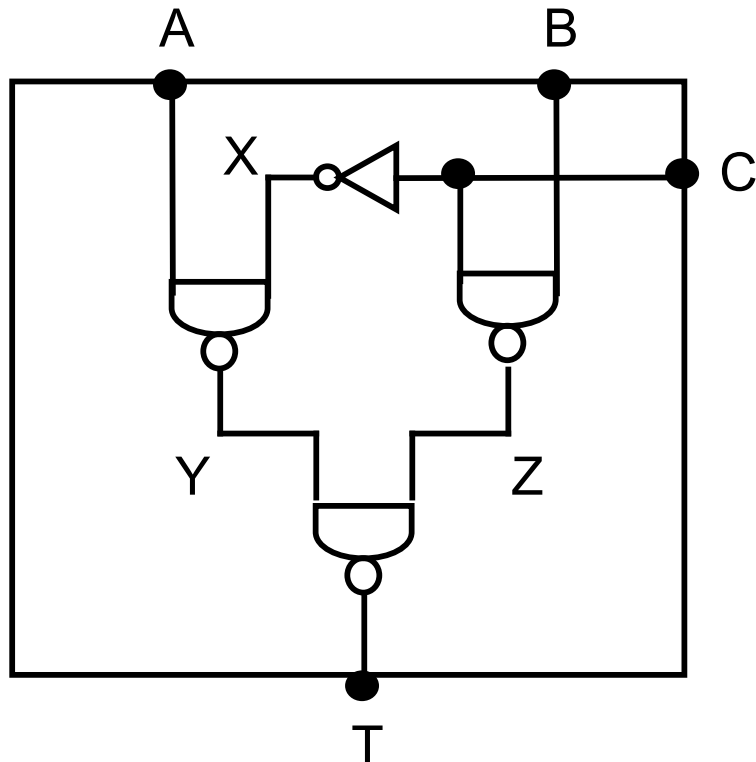
# Modèle des collages

---

- On dispose d'une description du circuit de type « net-list » comportant  $N$  portes logiques »
- On considère que tout défaut physique a pour effet que le circuit se comporte comme si un et un seul des  $N$  signaux du circuit était « collé à 0 » ou « collé à 1 ».

=> Avec cette hypothèse (très violente), le nombre total de circuits fautifs est dénombrable, et égal à  $2N$

## Exemple : multiplexeur 1 bit



Ce circuit possède 3 entrées A, B, C et 7 signaux : A, B, C, X, Y, Z, T

Il y a donc  $2^3 = 8$  vecteurs de test, et 14 collages possibles, (donc 14 circuits fautifs possibles).

Notation:

- On note S0X le circuit fautif dont le signal X est collé à 0
- On note S1X le circuit fautif dont le signal X est collé à 1.

Question : Quelles est le taux de couverture fourni par les trois vecteurs (001) (011) (111) ?



# Simulation de fautes

Vecteur			O	S	S	S	S	S	S	S	S	S	S	S	S	S	
A	B	C	K	0	1	0	1	0	1	0	1	0	1	0	1	0	1
				A	A	B	B	C	C	X	X	Y	Y	Z	Z	T	T
0	0	1	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1
0	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0	0	1
1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1

La simulation de fautes permet de calculer le taux de couverture d'un jeu de vecteurs  $\{V\}$  en simulant explicitement les  $2N$  circuits fautifs, pour déterminer quels circuits fautifs sont détectés par au moins un vecteur de  $\{V\}$ .

Elle est très coûteuse en temps de calcul puisqu'elle demande un nombre de simulations égal au produit du nombre de collages possibles par le nombre de vecteurs de l'ensemble  $\{V\}$ .

# Plan

---

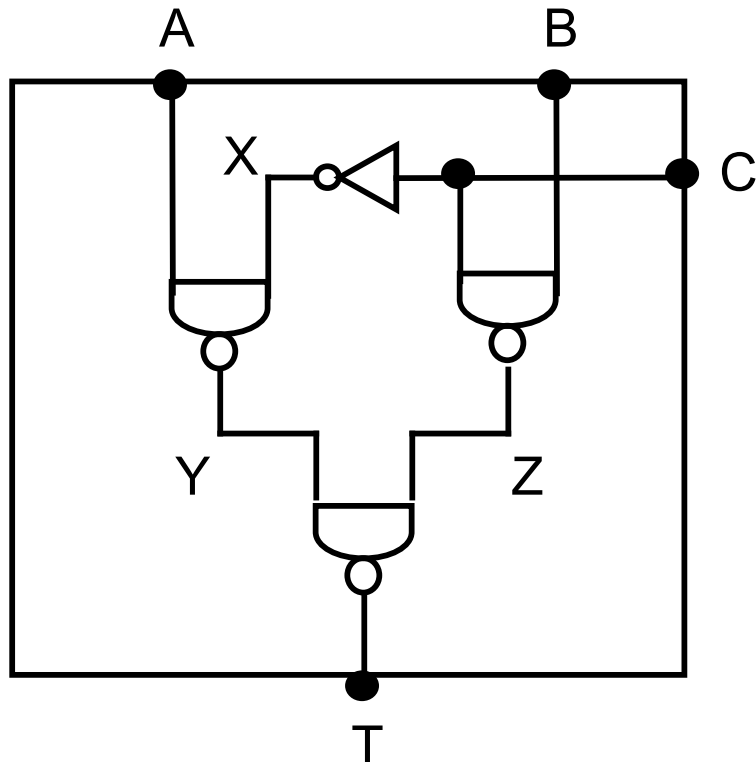
- Les enjeux du test
- Test Paramétrique
- Test fonctionnel et test structurel
- Le modèle des collages et la simulation de fautes
- **La génération automatique de vecteurs de test**
- Test des circuits séquentiels : le scan-path
- Techniques d'auto-test

# Génération automatique

---

- On cherche à résoudre le problème suivant :  
« Trouver un vecteur de test qui détecte un collage particulier d'un signal X interne du circuit »
- Il faut satisfaire deux contraintes :
  - sensibiliser un chemin entre une entrée du circuit et le nœud X, pour pouvoir contrôler X (et forcer une valeur différente de la valeur de collage)
  - sensibiliser un chemin entre le nœud X et une sortie du circuit, pour pouvoir observer la valeur effective du nœud X.

## Exemple : multiplexeur 1 bit



Question : trouver un vecteur de test permettant de détecter le collage à 1 du nœud Y

1/ contrôlabilité :

- pour forcer Y à 0, il faut  $(X=1)$  et  $(A=1)$
- pour cela il faut  $(C = 0)$  et  $(A = 1)$

2/ observabilité

- pour que Y soit observable, il faut  $Z = 1$
- pour cela, il faut  $(B = 0)$  ou  $(C = 0)$

=> Les deux vecteurs  $(A B C) = (1 0 0)$  et  $(A B C) = (1 1 0)$  détectent le collage S1Y

# Principe des ATPG

---

Les générateurs automatiques de vecteurs de test (Automatic Test Pattern Generator) fonctionnent sur le principe suivant :

1. On construit un « dictionnaire » des fautes du circuit, où toutes les fautes sont initialement marquées « non détectées »
2. On génère aléatoirement un premier ensemble  $\{v\}$  de vecteurs de test. On détermine, par simulation de fautes, toutes les fautes détectées par au moins un vecteur de  $\{v\}$ . On les marque « détectées » dans le dictionnaire.
3. On prend la première faute SiX non détectée, et on calcule (par sensibilisation des chemins) un vecteur de test V qui détecte la faute SiX. On détermine, par simulation de fautes, toutes les fautes détectées par V, et on les marque dans le dictionnaire.
4. On recommence le point 3 tant que le taux de couverture est inférieur à un seuil prédéfini (98% par exemple).

# Taux de couverture

---

- Le **taux de couverture** (au sens du modèle des collages) d'un jeu de vecteurs de test  $\{V\}$  n'est jamais égal à 100%, car il existe des fautes non détectables, à cause de la redondance des schémas logiques.
- Le taux de couverture ne permet pas de calculer le **pourcentage de circuits défectueux non détectés** : Un taux de couverture de 95% ne signifie pas que le pourcentage de circuits défectueux non détectés est égal à 5% ! ... Ce pourcentage est heureusement bien inférieur, ...
- La **corrélation** (expérimentalement observée) entre un taux de couverture élevé et un pourcentage de défauts non détectés très faible a permis au modèle des collages de s'imposer comme une métrique fiable dans l'industrie de la micro-électronique.
- Un taux de couverture de **98%** est généralement considéré comme suffisant et acceptable.

# Plan

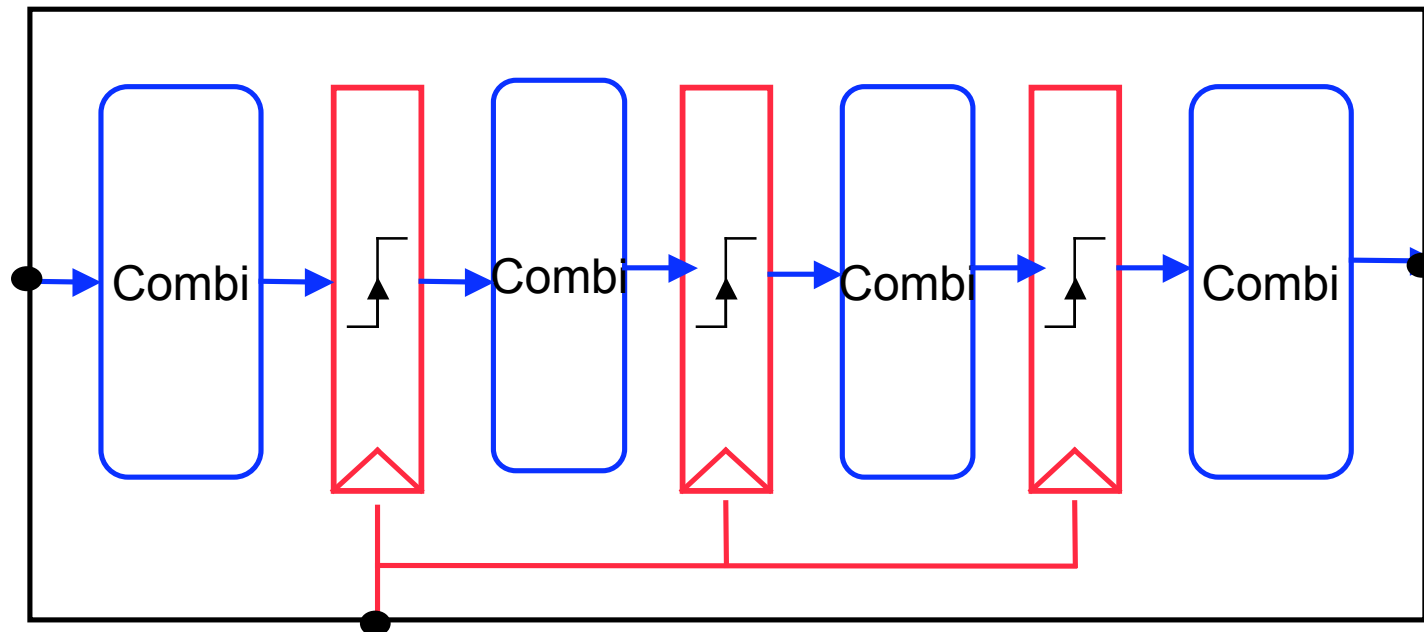
---

- Les enjeux du test
- Test Paramétrique
- Test fonctionnel et test structurel
- Le modèle des collages et la simulation de fautes
- La génération automatique de vecteurs de test
- **Test des circuits séquentiels : le scan-path**
- Techniques d'auto-test

# Observabilité et Contrôlabilité

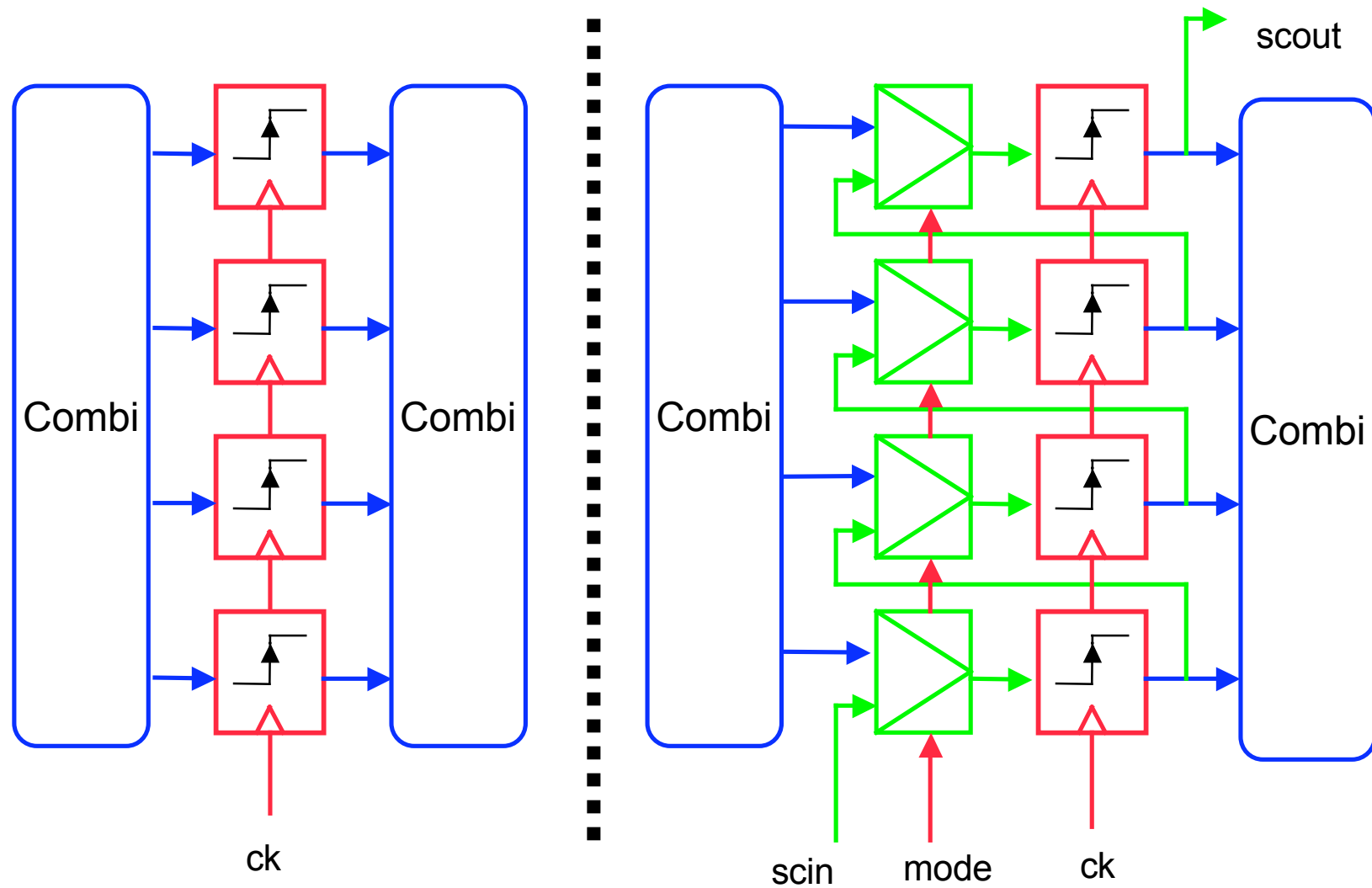
Pour tester un bloc combinatoire, il faut pouvoir :

- **contrôler** les valeurs de ses entrées
- **observer** les valeurs de ses sorties



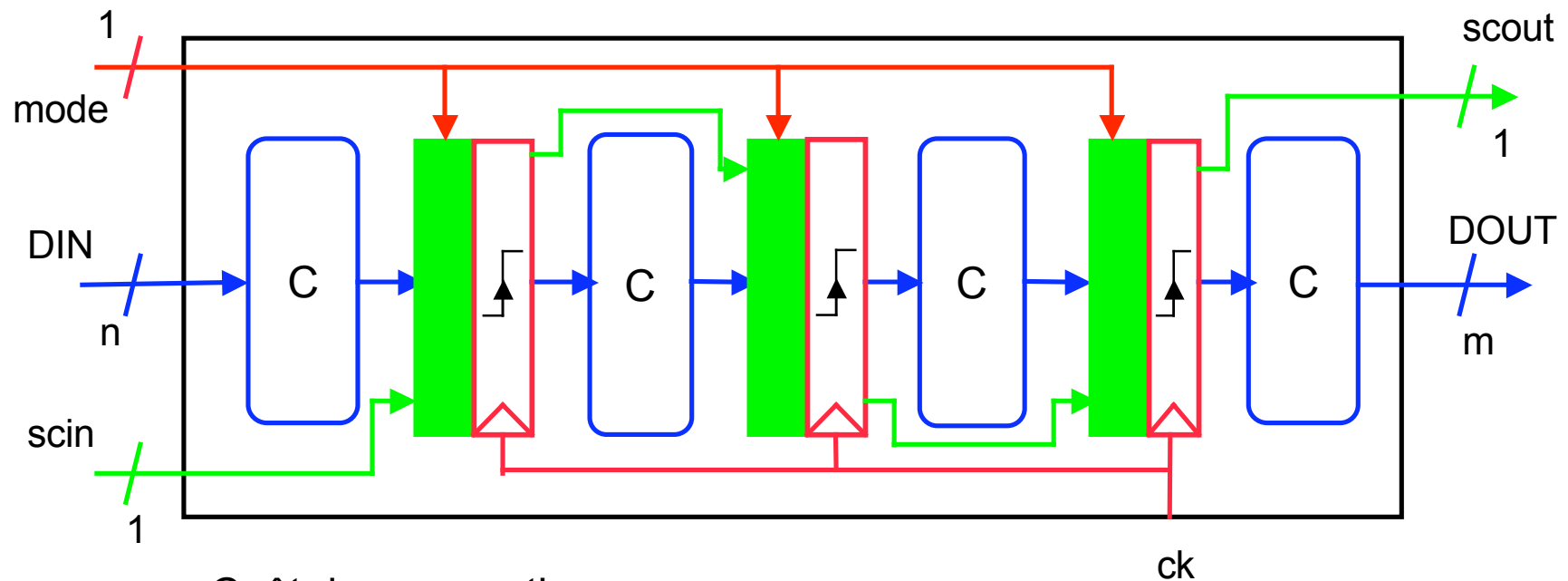


# Technique de « Scan-Path »



# Mode « Scan »

Le mode « scan » permet de transformer l'ensemble des N bascules internes en un (long) registre à décalage : il faut N cycles pour charger une nouvelle configuration dans les N registres...



Coût du scan path :

- un multiplexeur par bit
- trois connecteurs externes : mode, scin, scout

# Pas de test

---

L'application d'un vecteur de test correspond à la séquence suivante :

- 1/ N cycles en mode « scan » pour initialiser le scan-path (chargement série du vecteur de test dans les registres internes en utilisant l'entrée scin)
- 2/ 1 cycle en mode « fonctionnel » pour écrire dans les registres internes le résultat calculé par les blocs combinatoires. (il faut aussi appliquer les valeurs sur les entrées externes, et observer les valeurs obtenues sur les sorties externes)
- 3/ N cycles en mode « scan » pour relire en série, sur la sortie scout, les résultats du vecteur de test, stockés dans les registres.

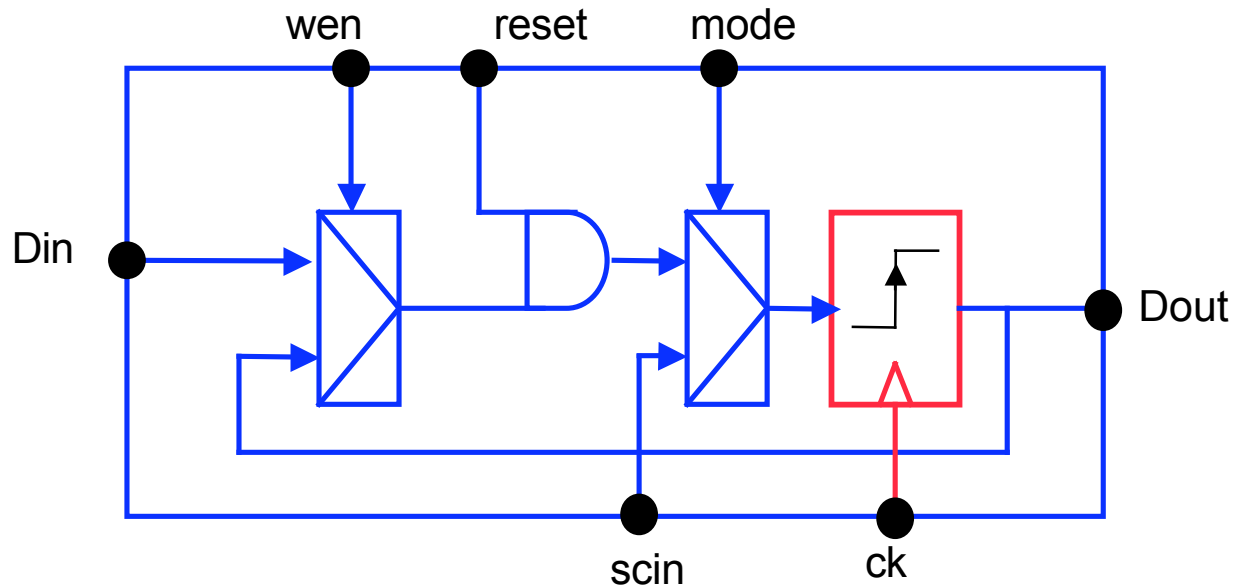
=> L'application d'un seul vecteur de test nécessite  $2N+1$  cycles !

# Registre « habillé »

Dans le cas général, l'écriture dans un registre est contrôlée par trois signaux :

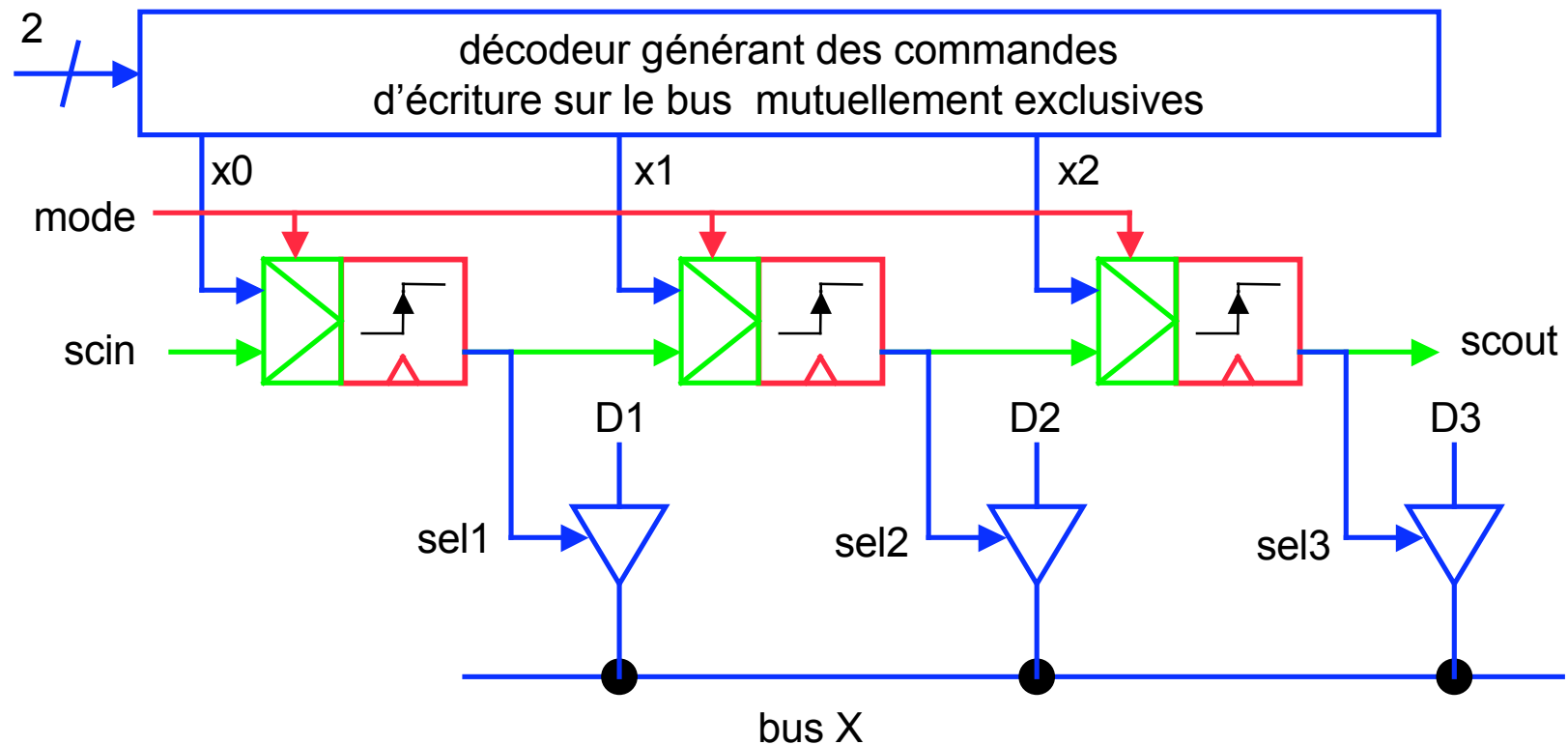
- mode : activation du scan-path
- wen : autorisation d'écriture en mode fonctionnel
- reset : initialisation

La priorité est la suivante : mode > reset > wen



# Danger du scan-path...

L'utilisation du scan-path peut créer des court-circuits sur les bus, lorsque les commandes des émetteurs trois-états sont stockées dans des registres « scanables »...



# Plan

---

- Les enjeux du test
- Test Paramétrique
- Test fonctionnel et test structurel
- Le modèle des collages et la simulation de fautes
- La génération automatique de vecteurs de test
- Test des circuits séquentiels : le scan-path
- **Techniques d'auto-test**

## Limitations du test « externe »

---

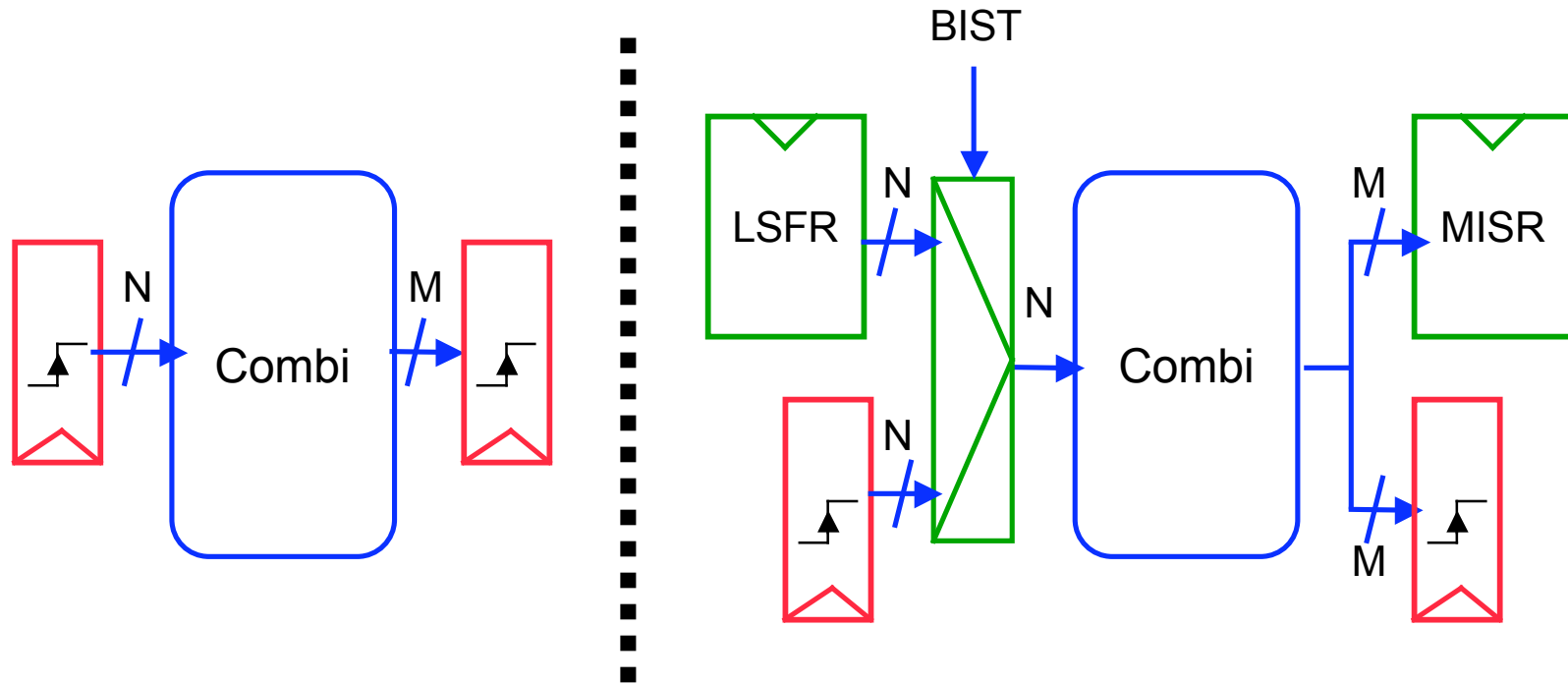
La complexité des systèmes intégrés sur puce devient difficilement compatible avec le test externe :

- **La durée du test**
  - => Le chargement/déchargement série des vecteurs de test par le scan path est peu efficace.
- **Le volume des données de test**
  - => la capacité de stockage des testeurs est un facteur de coût très important.
- **La fréquence de test**
  - => la fréquence du testeur est largement inférieure à la fréquence de fonctionnement des circuits, ce qui interdit la détection des fautes temporelles.

# Techniques d'auto-test (BIST)

Pour chaque bloc combinatoire à tester, on ajoute :

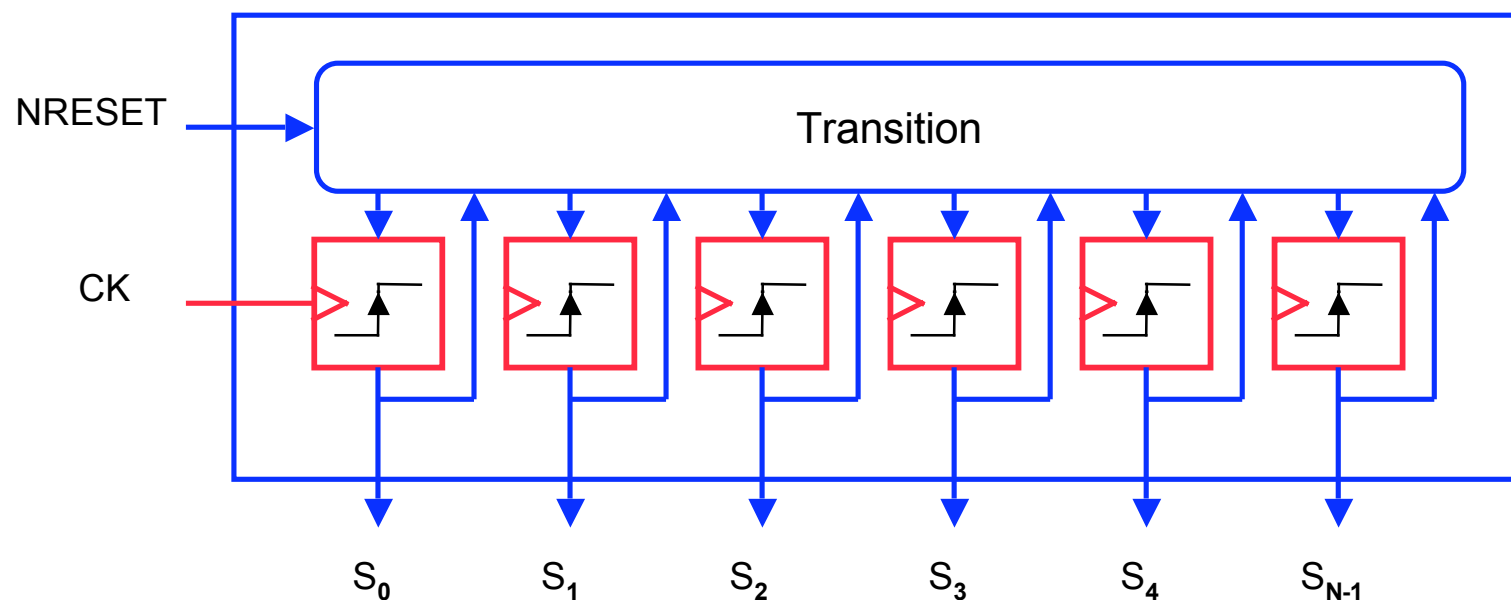
- un générateur de vecteurs de test (LSFR) en entrée
- un analyseur de signature (MISR) en sortie





# Linear Feedback Shift Register (LFSR)

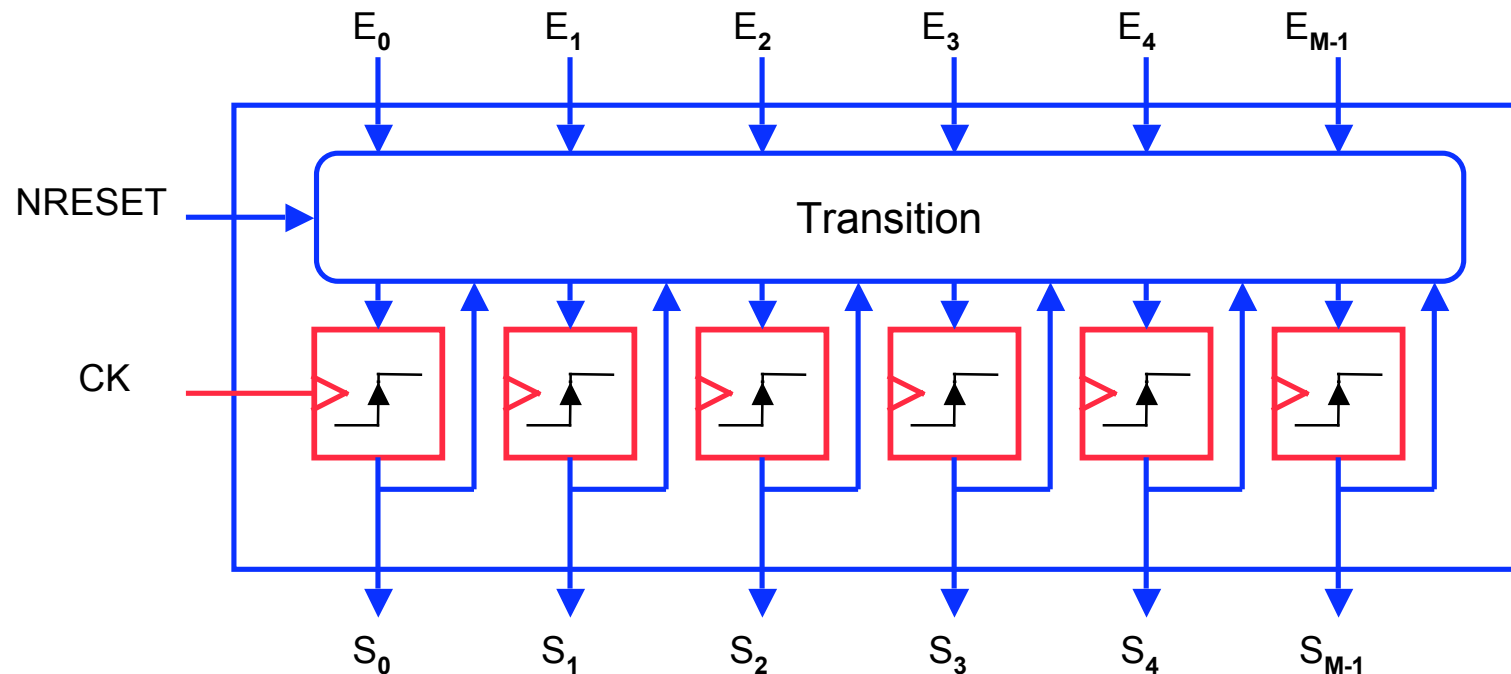
Un LFSR contient un registre N bits, et se comporte comme un automate.  
La fonction de transition permet de parcourir l'ensemble des  $2^N - 1$  états possibles suivant une séquence « pseudo-aléatoire ».



La fonction de transition est réalisée avec des XOR, et l'état « 000000 » n'est généralement pas accessible.

# Multiple Inputs Shift Register (MISR)

Un MISR contient un registre M bits, et se comporte comme un automate.  
La fonction de transition permet d'accumuler dans le registre une « signature ».  
La signature est la valeur contenue dans le registre après K cycles.  
Cette valeur dépend des  $M * K$  bits accumulés pendant K cycles.



La probabilité de ne pas détecter un bit erroné est proche de  $1/2^M$