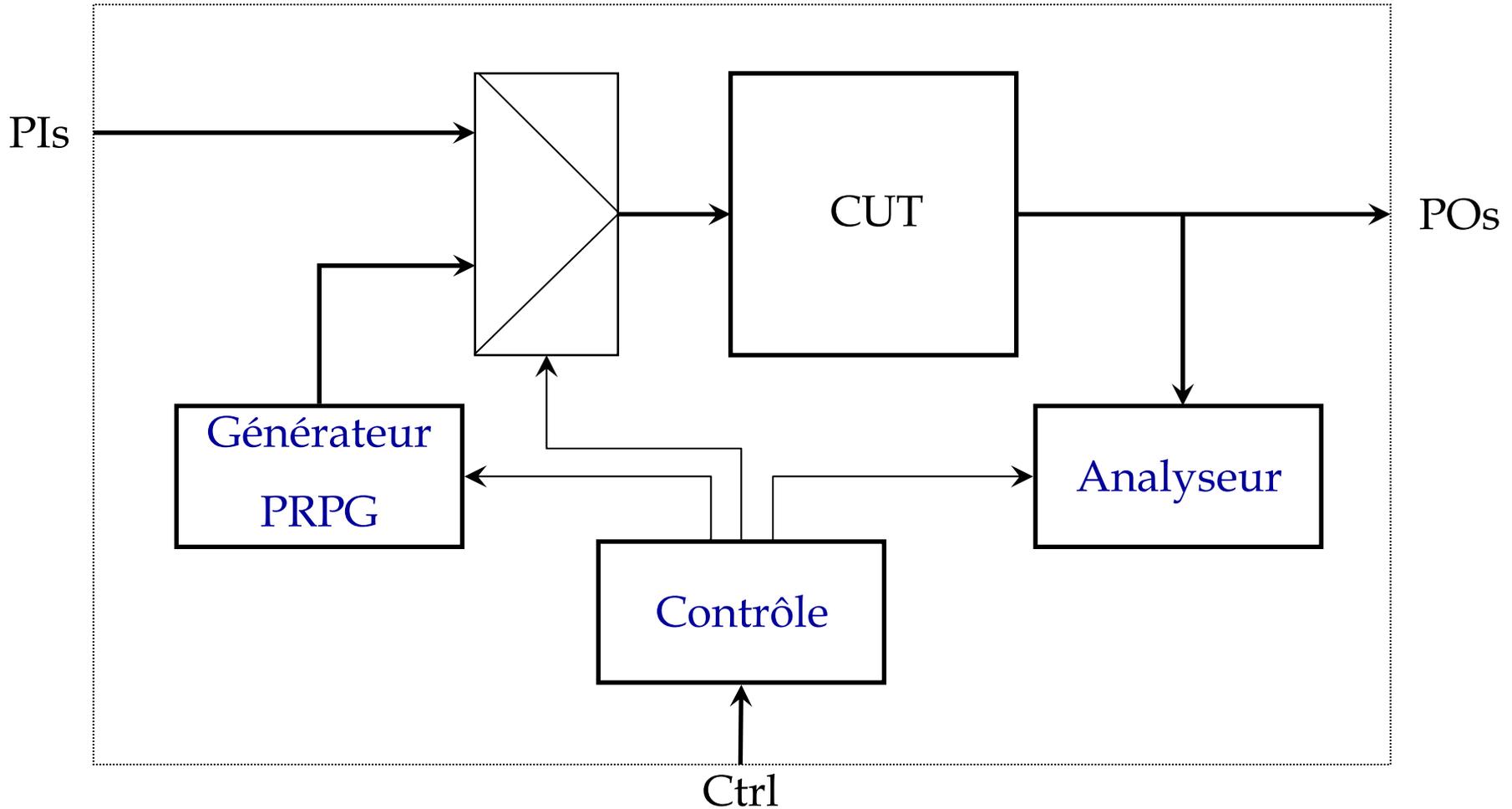


# Architecture générale du BIST



# « Linear Feedback Shift Register » LFSR

☞ Les registres à décalage à rebouclage linéaire « LFSR » sont des circuits logiques contenant les composants suivants :

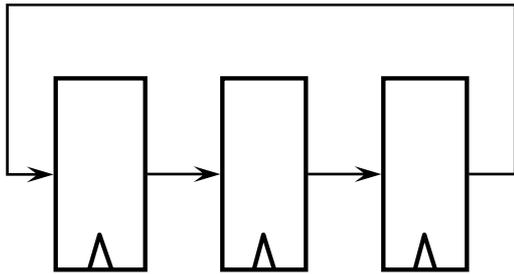
- 1- Bascules D « D Flip-Flops »
- 2- Additionneurs modulo-2 « XOR »

☞ Propriétés du LFSR :

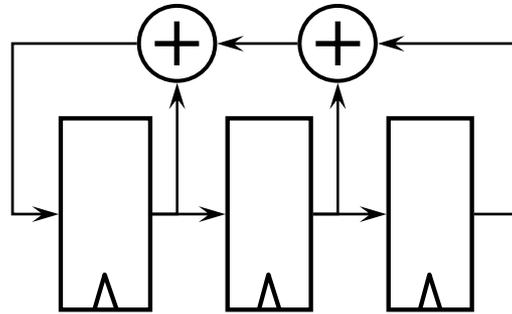
- Cyclique
  - ☞ Le circuit revient à son état initial pour un nombre suffisant de coups d'horloge
- Période
  - ☞ La période du LFSR est comprise entre 1 et  $2^n - 1$ ,  $n$  est le nombre de bascules D (longueur du LFSR ou nombre d'étages)

# Exemples de registres à décalage

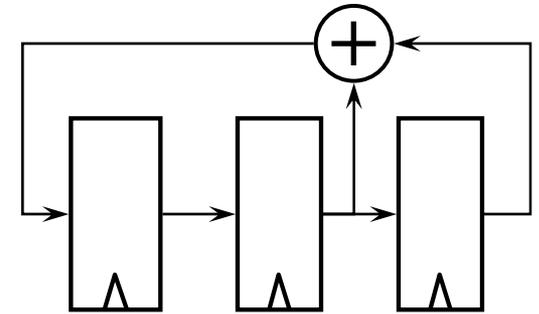
Cyclic SR



Feedback Shift Register



Maximal-Length FSR



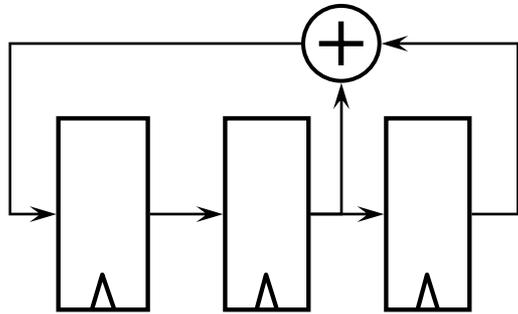
S0 :	0	1	1
S1 :	1	0	1
S2 :	1	1	0
-----			
S0 :	0	1	1

S0 :	0	1	1
S1 :	0	0	1
S2 :	1	0	0
S3 :	1	1	0
-----			
S0 :	0	1	1

S0 :	0	1	1
S1 :	0	0	1
S2 :	1	0	0
S3 :	0	1	0
S4 :	1	0	1
S5 :	1	1	0
S6 :	1	1	1
-----			
S0 :	0	1	1

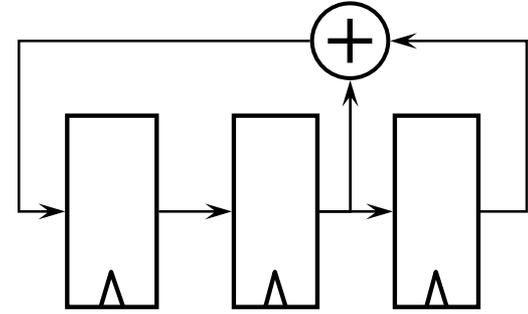
# Exemples de registres à décalage (suite)

Maximal-Length FSR



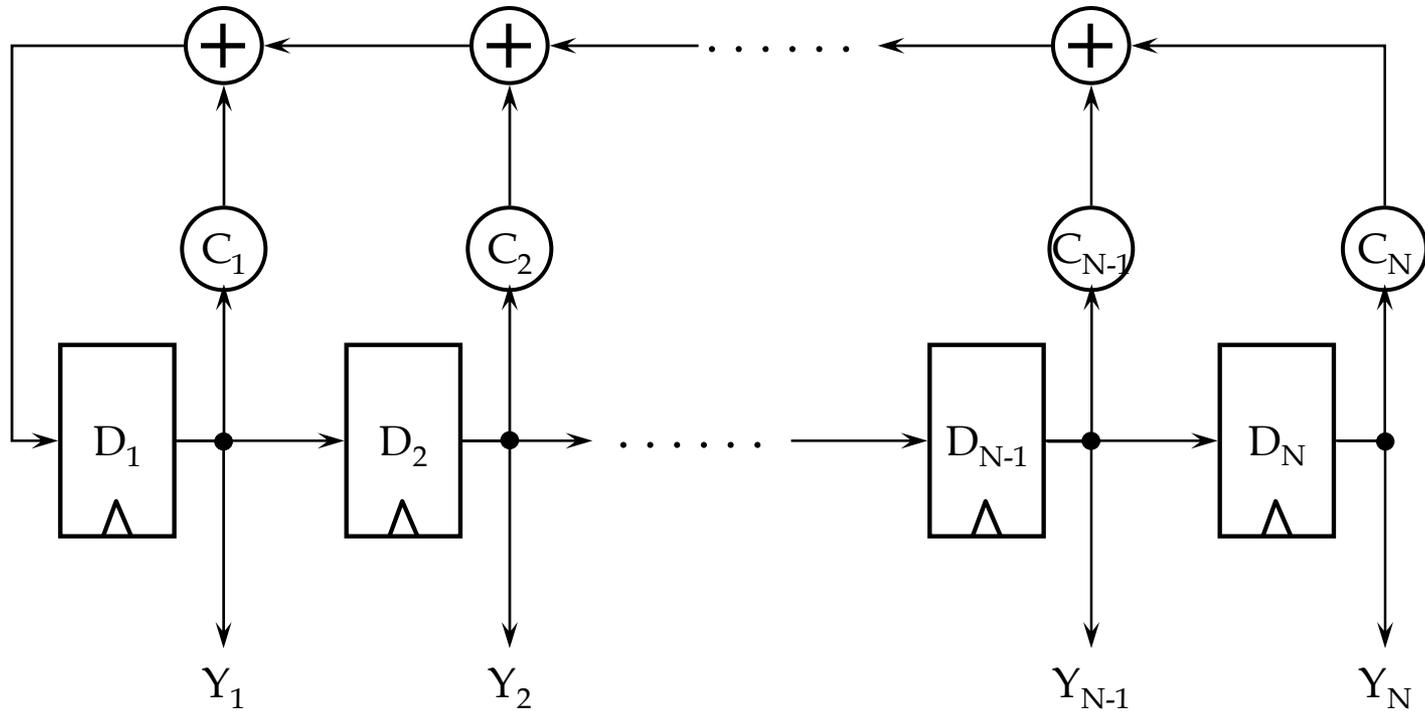
S0 :	1	1	1	}	<i>Cyclique</i>			
S1 :	0	1	1			<i>de</i>		
S2 :	0	0	1				<i>longueur</i>	
S3 :	1	0	0					$2^3 - 1$
S4 :	0	1	0					
S5 :	1	0	1					
S6 :	1	1	0					
S0 :				1	1			

Maximal-Length FSR



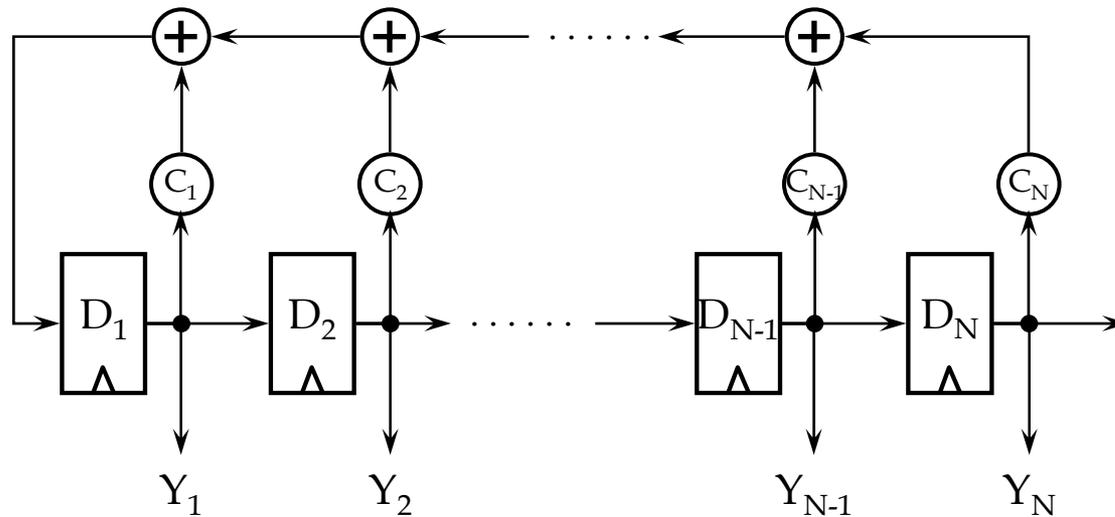
S0 :	0	0	0			
S0 :				0	0	0

# Architecture d'un LFSR standard



Architecture générale d'un LFSR standard

# Polynôme caractéristique

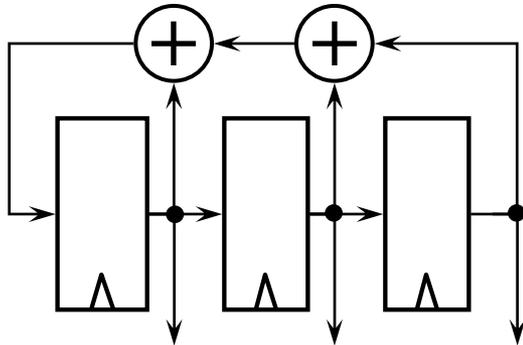


➡ On appelle polynôme caractéristique du LFSR

$$P(X) = 1 + \sum_{i=1}^n C_i X^i$$
$$= 1 + C_1 X + C_2 X^2 + \dots + C_{n-1} X^{n-1} + C_n X^n$$

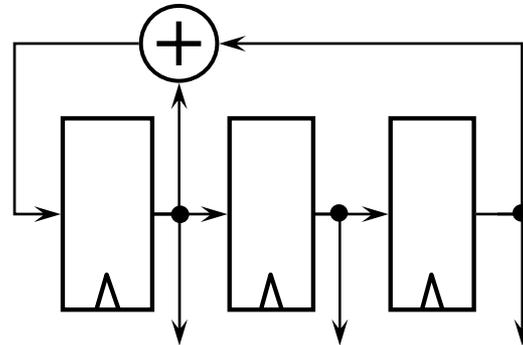
# Exemples de LFSR

$$P(X) = 1 + X + X^2 + X^3$$



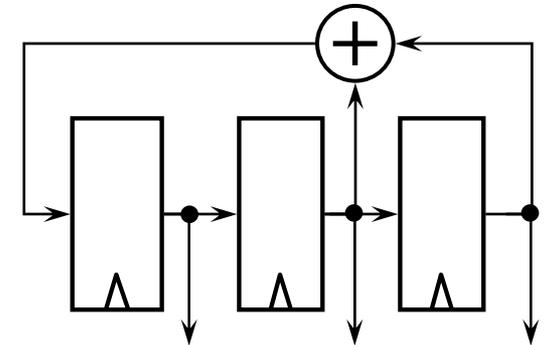
S0 :	0	1	1
S1 :	0	0	1
S2 :	1	0	0
S3 :	1	1	0
-----			
S0 :	0	1	1

$$P(X) = 1 + X + X^3$$



S0 :	0	1	1
S1 :	1	0	1
S2 :	0	1	0
S3 :	0	0	1
S4 :	1	0	0
S5 :	1	1	0
S6 :	1	1	1
-----			
S0 :	0	1	1

$$P(X) = 1 + X^2 + X^3$$



S0 :	0	1	1
S1 :	0	0	1
S2 :	1	0	0
S3 :	0	1	0
S4 :	1	0	1
S5 :	1	1	0
S6 :	1	1	1
-----			
S0 :	0	1	1

# Polynômes primitifs (suite)

Valeurs de N	Polynômes primitifs
1,2,3,4,6,7,15,22	$1 + X + X^n$
5,11, 21, 29	$1 + X^2 + X^n$
10,17,20,25,28,31	$1 + X^3 + X^n$
9	$1 + X^4 + X^n$
23	$1 + X^5 + X^n$
18	$1 + X^7 + X^n$
8	$1 + X^2 + X^3 + X^4 + X^n$
12,14,16	$1 + X + X^3 + X^4 + X^n$
13	$1 + X + X^4 + X^6 + X^n$

# Polynômes primitifs (suite)

☞ Les polynômes de degré 3 sont :

➤  $P(X)=1+X^3$

➤  $P(X)=1+X+X^3$  primitif

➤  $P(X)=1+X^2+X^3$  primitif

➤  $P(X)=1+X+X^2+X^3$

❖  $(1+X^3) = (1+X)(1+X+X^2)$  et  $(1+X+X^2+X^3) = (1+X)(1+X^2)$

☞ Il existe plusieurs polynômes primitifs de degré  $n$ , mais il est avantageux d'utiliser celui qui a le moins de termes car il nécessite moins de portes xor.

# Propriétés des générateurs LFSR

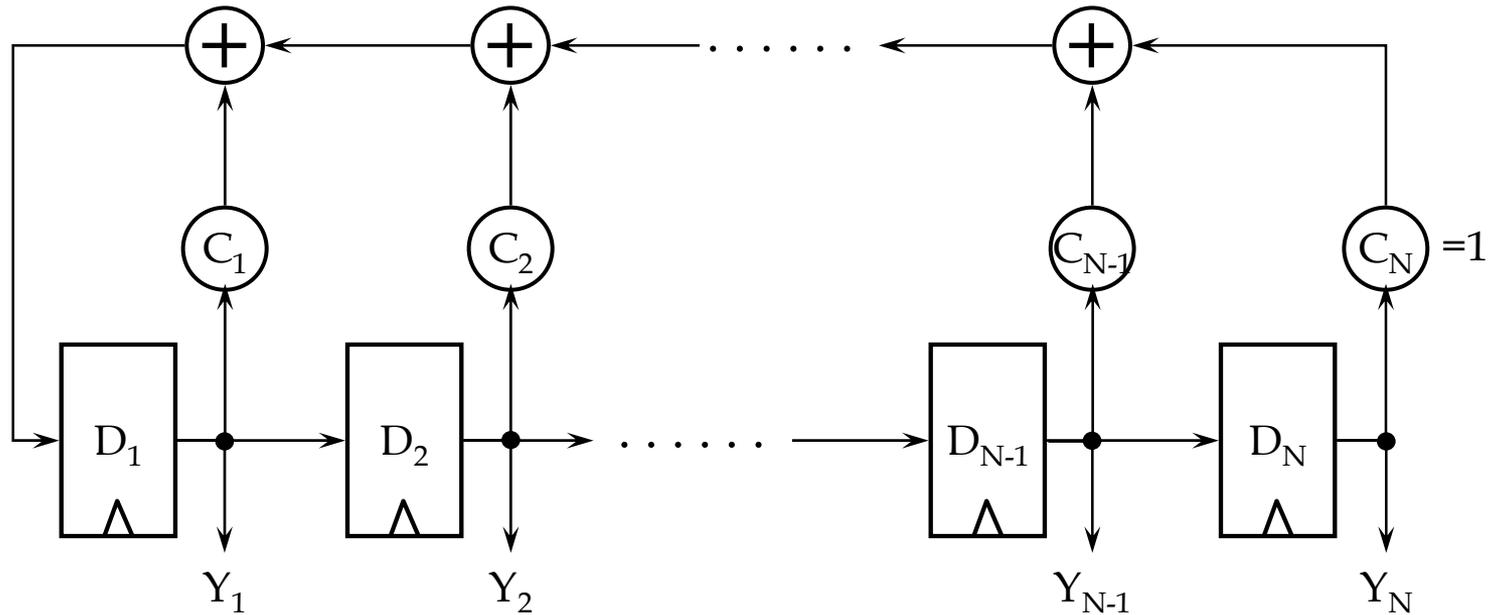
☞ La séquence des vecteurs générés par un LFSR ayant un polynôme primitif d'ordre  $n$  à les propriétés suivantes :

- ❖ La période de la séquence est  $2^n - 1$
- ❖ Si on initialise le LFSR à une valeur différente de 0, le LFSR génère tous les vecteurs possibles (sauf 0) avant de revenir au vecteur initial
- ❖ La séquence de longueur  $(2^n - 1)$  a  $2^{n-1}$  1s et  $(2^{n-1} - 1)$  0s
- ❖ La séquence de longueur  $(2^n - 1)$  a  $(2^{n-1} - 1)$  transitions

☞ Pour la génération pseudo-aléatoire, le LFSR est l'un des meilleurs générateurs

# Types de LFSR

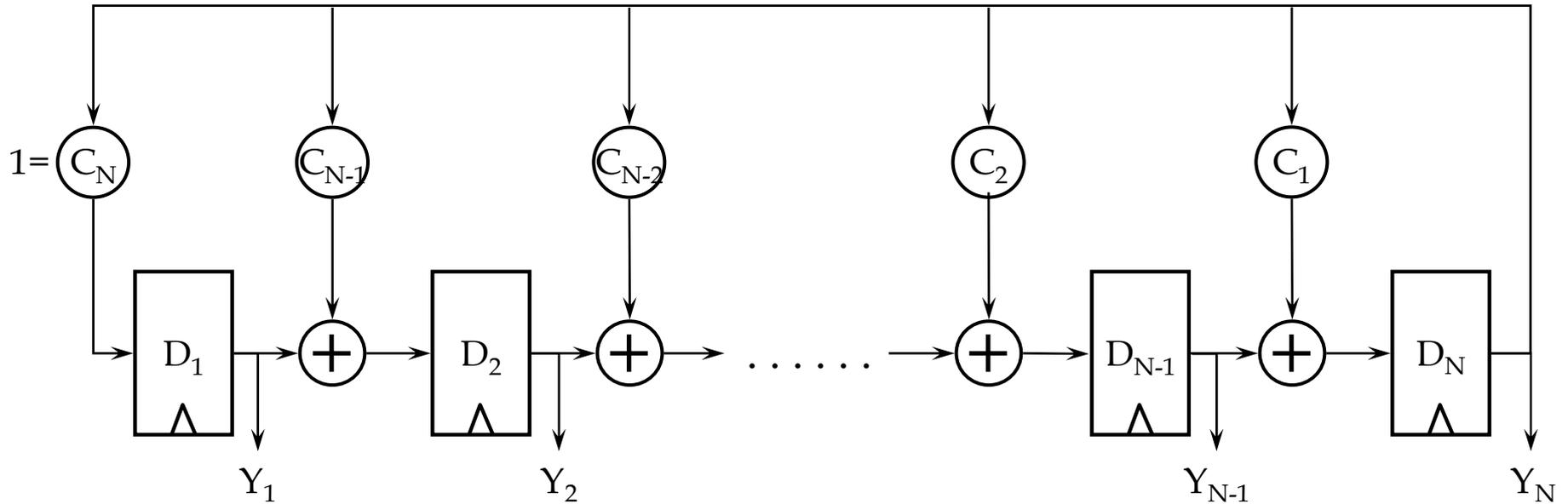
☞ Type 1 : LFSR standard « xor externes »



$$P(X) = 1 + \sum_{i=1}^n C_i X^i = 1 + C_1 X + C_2 X^2 + \dots + C_{n-1} X^{n-1} + C_n X^n$$

# Types de LFSR (suite)

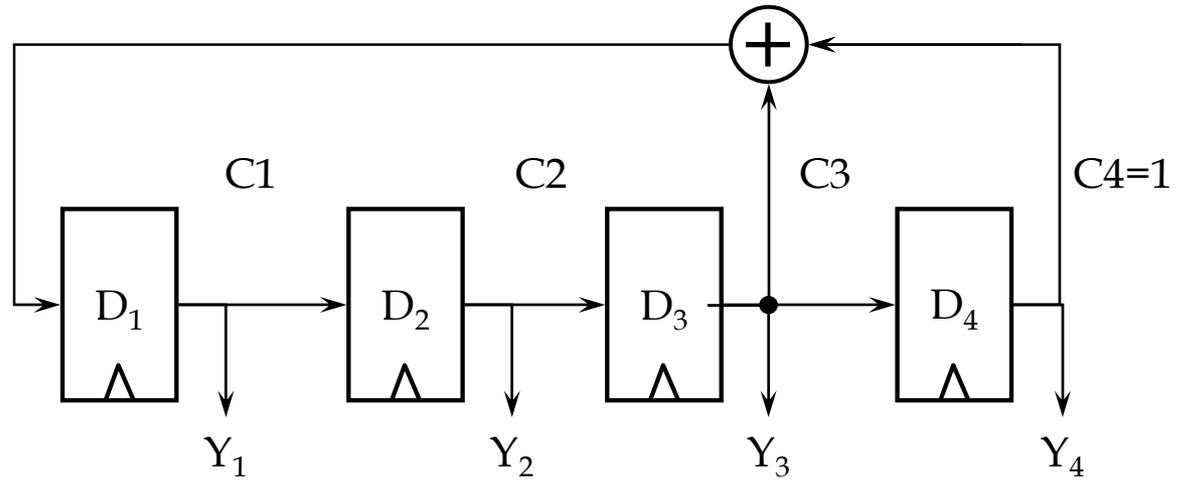
☞ Type 2 : LFSR dual « xor internes »



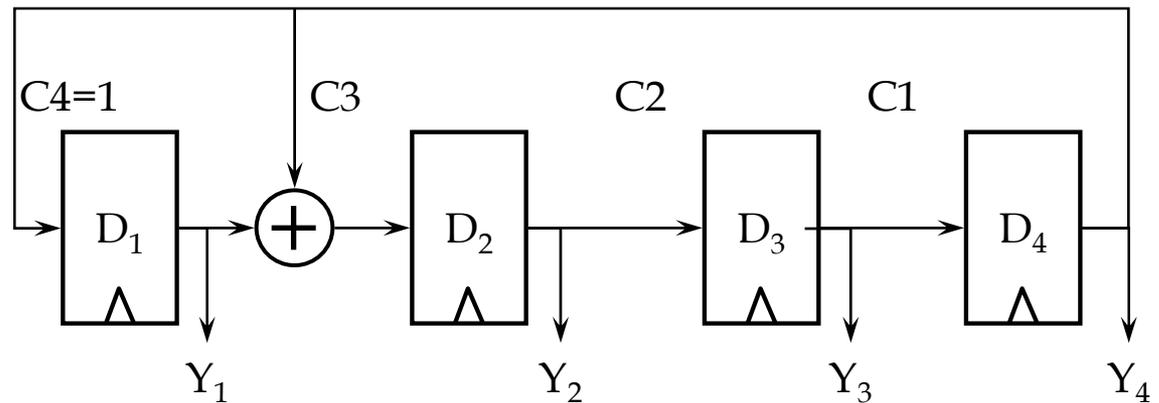
$$P(X) = 1 + C_1X + C_2X^2 + \dots + C_{n-1}X^{n-1} + C_nX^n$$

# Types de LFSR (suite)

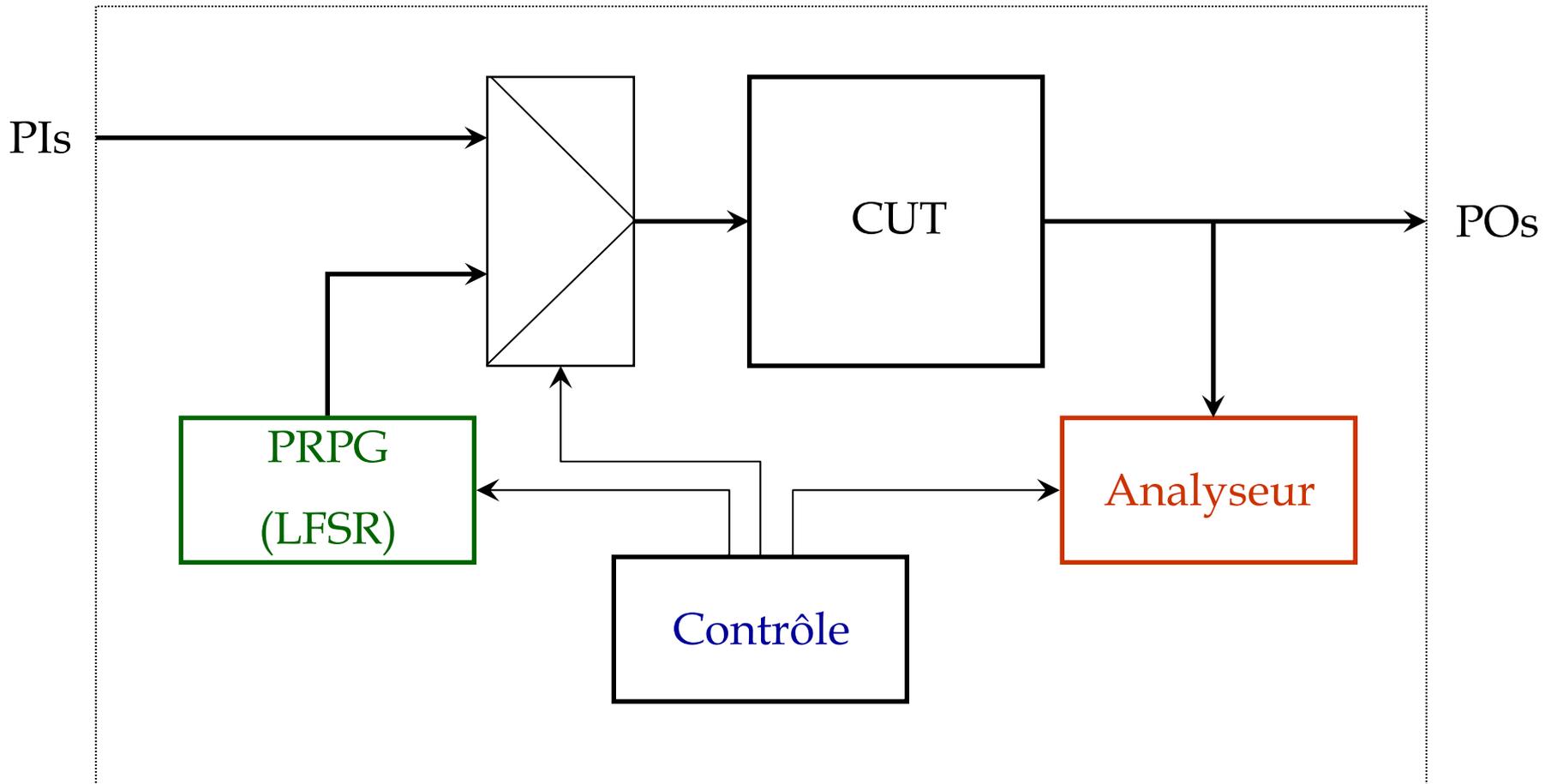
☞ LFSR  
standard



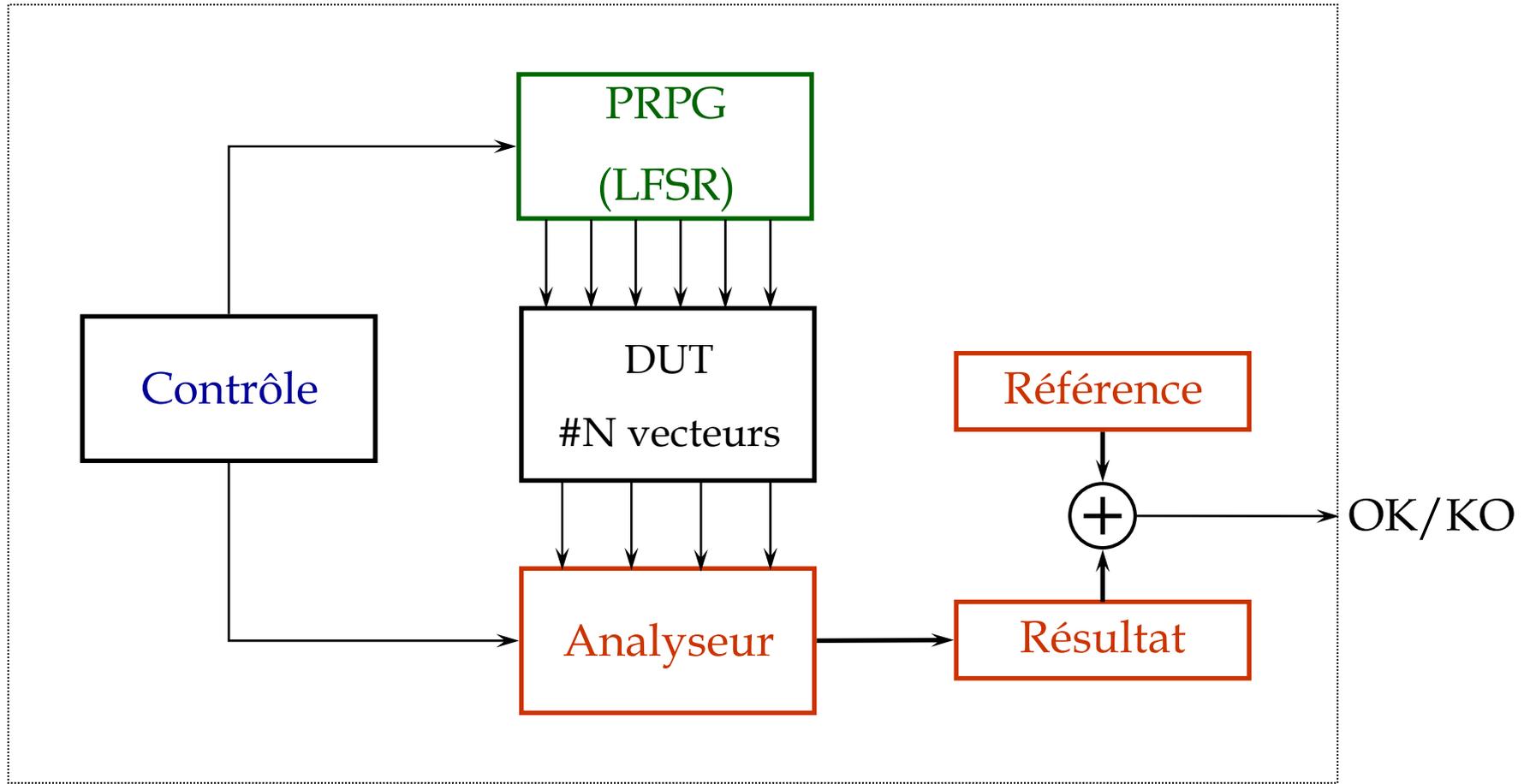
☞ LFSR dual



# Architecture générale du BIST



# Analyse des réponses



# Analyse des réponses (suite)

## 👉 But

- Trouver un moyen de comparer les réponses du circuit sous test avec les réponses du circuit correct pour pouvoir détecter les circuits défectueux. Les paramètres importants à considérer sont :
  - ❖ L'algorithme utilisé doit être facile à implémenter
  - ❖ Le temps d'analyse des résultats doit être très rapide
  - ❖ Minimiser la longueur du résultat à stocker
  - ❖ Minimiser la perte d'information

## 👉 Problématique de l'analyse des réponses

- Compression (réversible) et stockage trop coûteux
- Compaction (irréversible) engendre le problème de masquage de fautes

# Analyse des réponses (suite)

## ☞ Masquage de fautes

- Soit  $C(m)$  la fonction de compression (compaction) utilisée pour  $n$  vecteurs
- On appelle « signature » le résultat de la fonction  $C(m)$
- Une faute est masquée si la fonction  $C(m)$  donne la même signature pour le circuit fautif et le circuit correct. Conséquences :
  - ❖ Pour le test, il est impossible de détecter la faute malgré que la séquence de test détecte la faute
  - ❖ Pour le diagnostic, il est impossible de différencier deux fautes qui ont la même signature

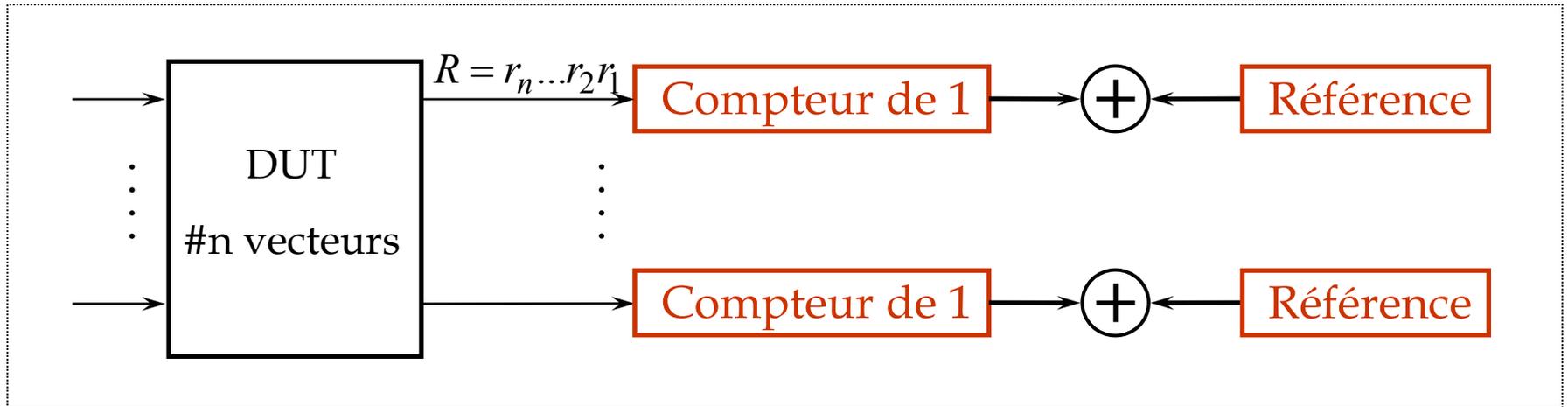
☞ Une des mesures importantes de la qualité d'une technique de compression est la probabilité de masquage de fautes

# Analyse des réponses (suite)

☞ Il existe plusieurs techniques d'analyse des réponses qui sont :

- Contage de un « Ones counting »
- Contage de transitions « Transitions counting »
- Vérification de la parité « Parity checking »
- Compression cyclique « Cyclic code compression »
- Analyse de signatures « Signature analysis »
  - ↳ en utilisant un registre LFSR
  - ↳ en utilisant un registre MISR
  - ↳ en utilisant un registre BILBO

# Contage de un

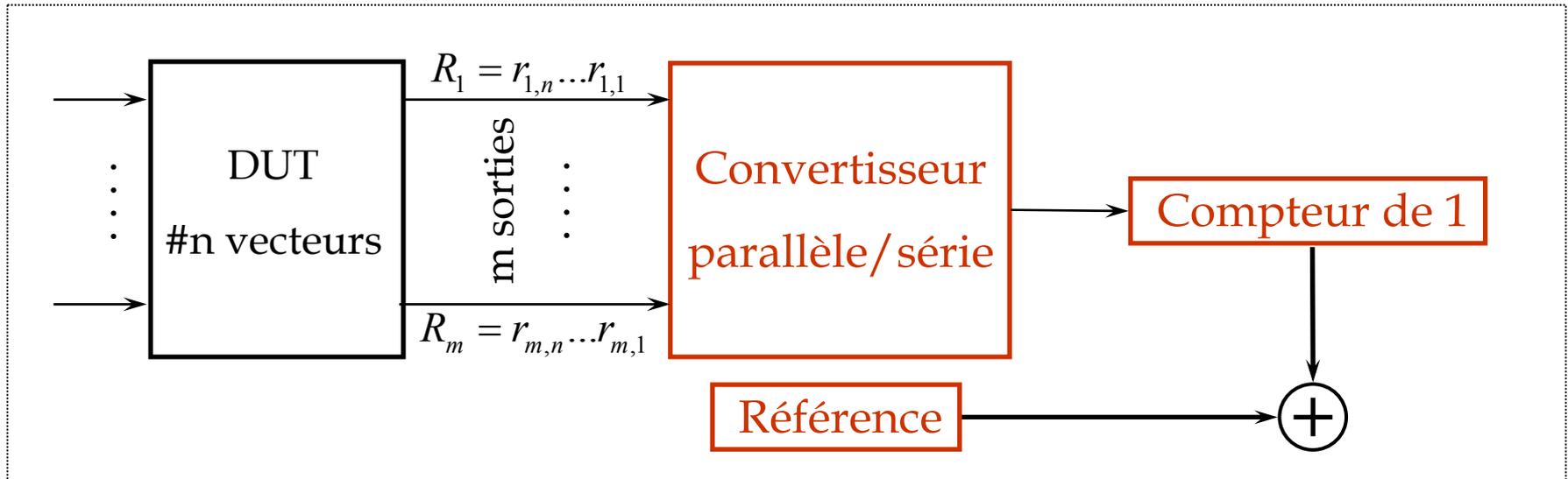


La fonction de compression est :  $1C(R) = \sum_{i=1}^n r_i = r$  on a  $0 \leq r \leq n$

Le nombre de séquence  $R'$  de  $n$  bits ayant  $1C(R') = r$  est :  $\binom{n}{r} = \frac{n!}{r!(n-r)!}$

La probabilité de masquage de fautes est :  $P(n/r) = \frac{\binom{n}{r} - 1}{2^n - 1}$

# Contage de un (suite)



La fonction de compression est :  $1C(R) = \sum_{j=1}^m \sum_{i=1}^n r_{j,i} = r$  on a  $0 \leq r \leq n.m$

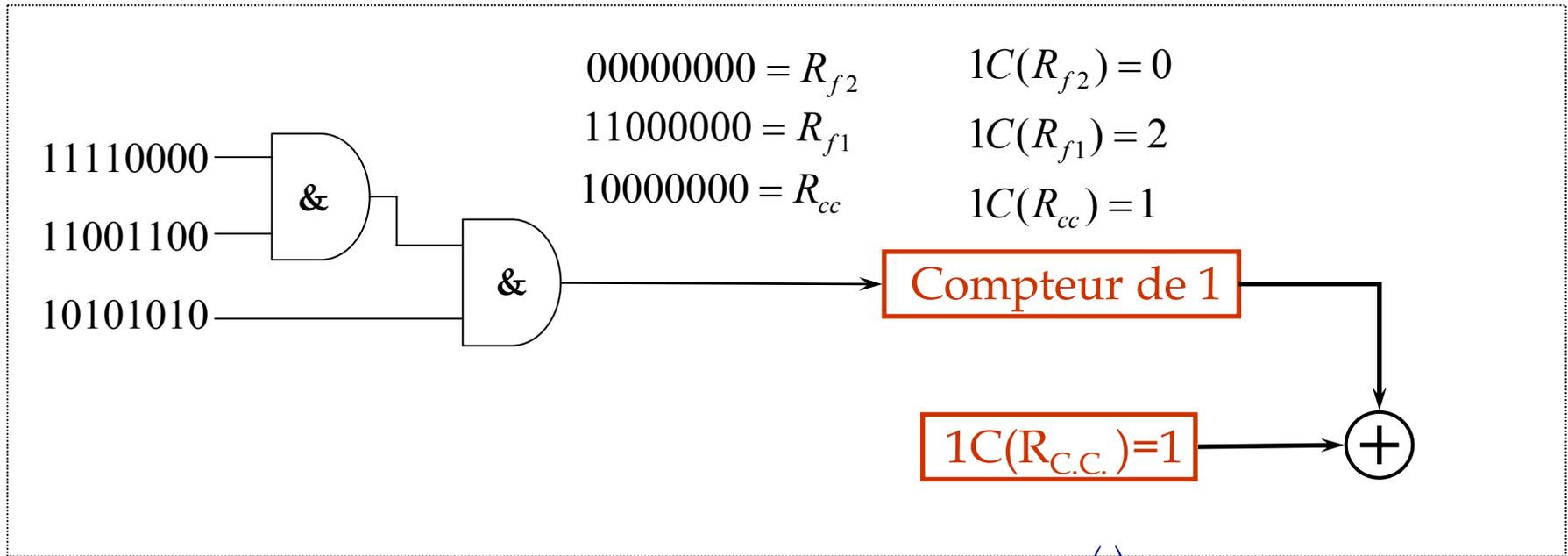
## 👉 Avantage :

- Diminue la surface additionnelle

## 👉 Inconvénients :

- Augmente le temps d'application des tests
- Augmente la probabilité de masquage de fautes

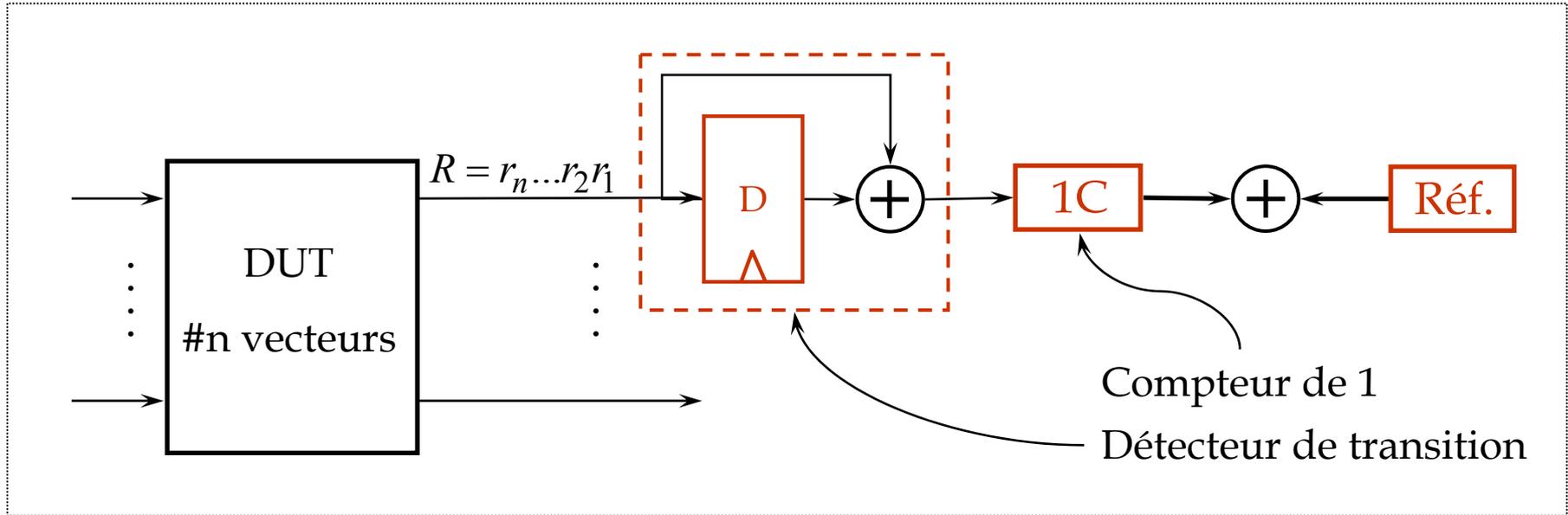
# Contage de un (suite)



La probabilité de masquage de fautes est :  $P(8/1) = \frac{\binom{8}{1} - 1}{2^8 - 1} = \frac{7}{255} = 0,02745$

On remarque que la probabilité de masquage est plus faible pour les valeurs extrêmes de  $1C(R_{C.C.})$ . Si  $1C(R_{C.C.})=0$  ou  $8$  alors  $P=0$

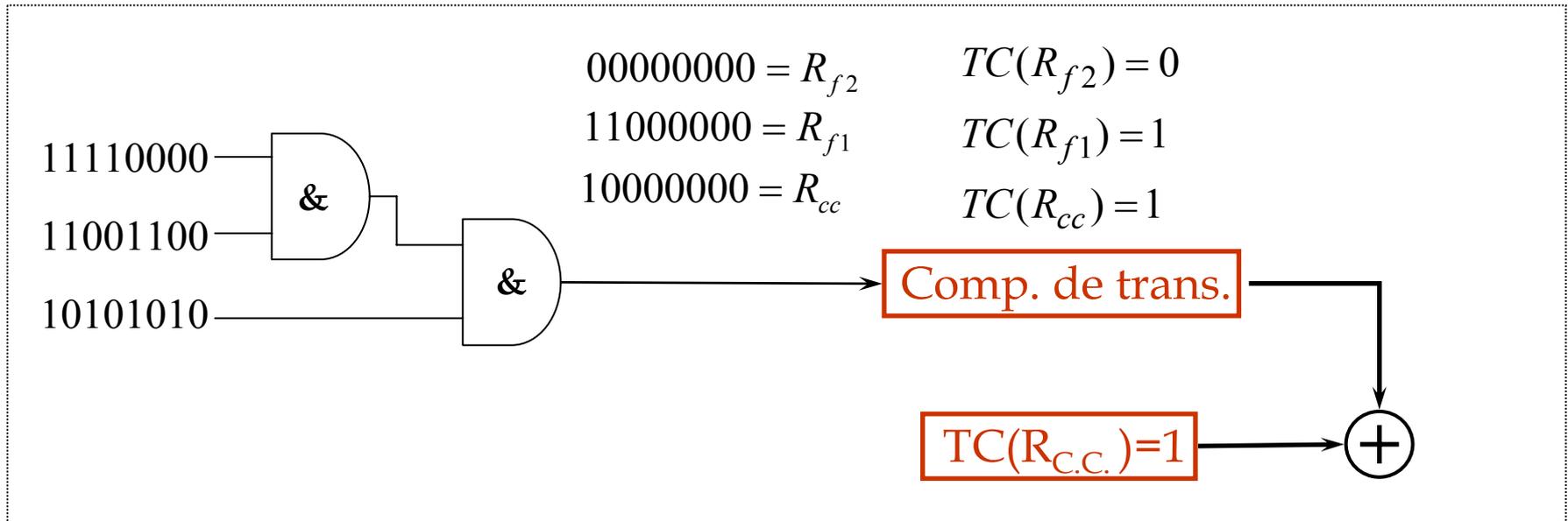
# Contage de transitions



La fonction de compression est :  $TC(R) = \sum_{i=1}^{n-1} (r_i \oplus r_{i+1}) = r$  on a  $0 \leq r \leq (n-1)$

La probabilité de masquage de fautes est :  $P(n/r) = \frac{2^{\binom{n-1}{r}} - 1}{2^n - 1}$

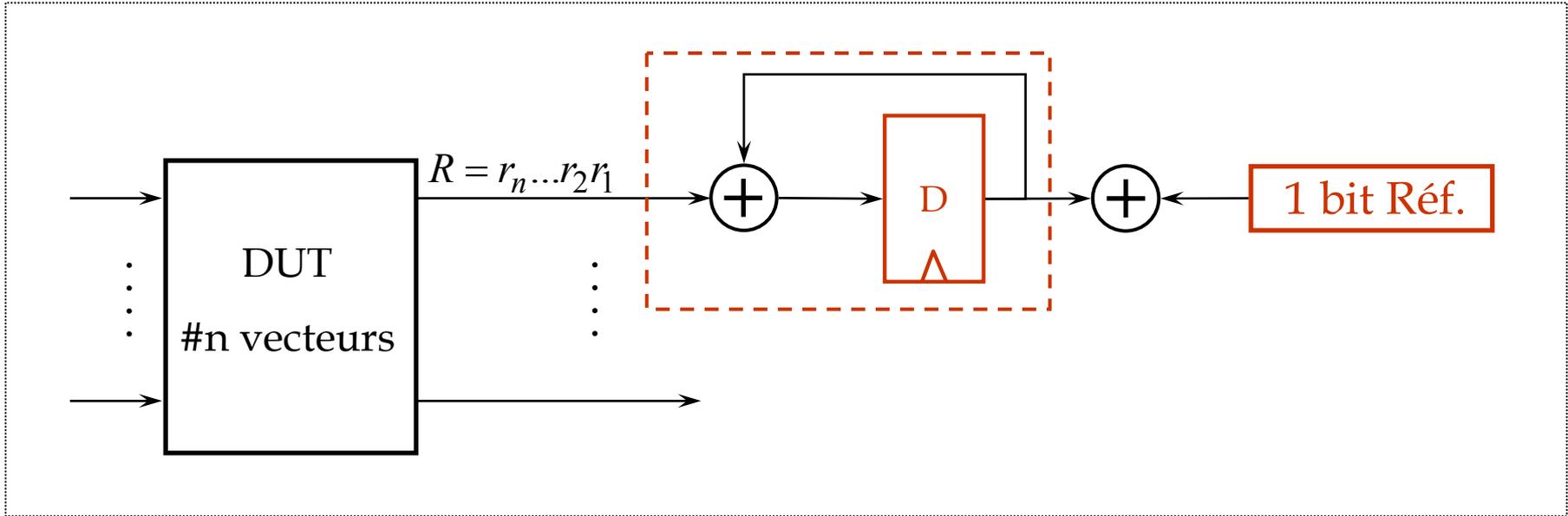
# Contage de transitions (suite)



La probabilité de masquage de fautes est :  $P(8/1) = \frac{2^{\binom{7}{1}} - 1}{2^8 - 1} = \frac{14}{255} = 0,055$

- ☞ On remarque que, dans ce cas, la probabilité de masquage est plus grande que pour la technique de comptage de 1.
- ☞ On ne peut pas détecter la faute f1

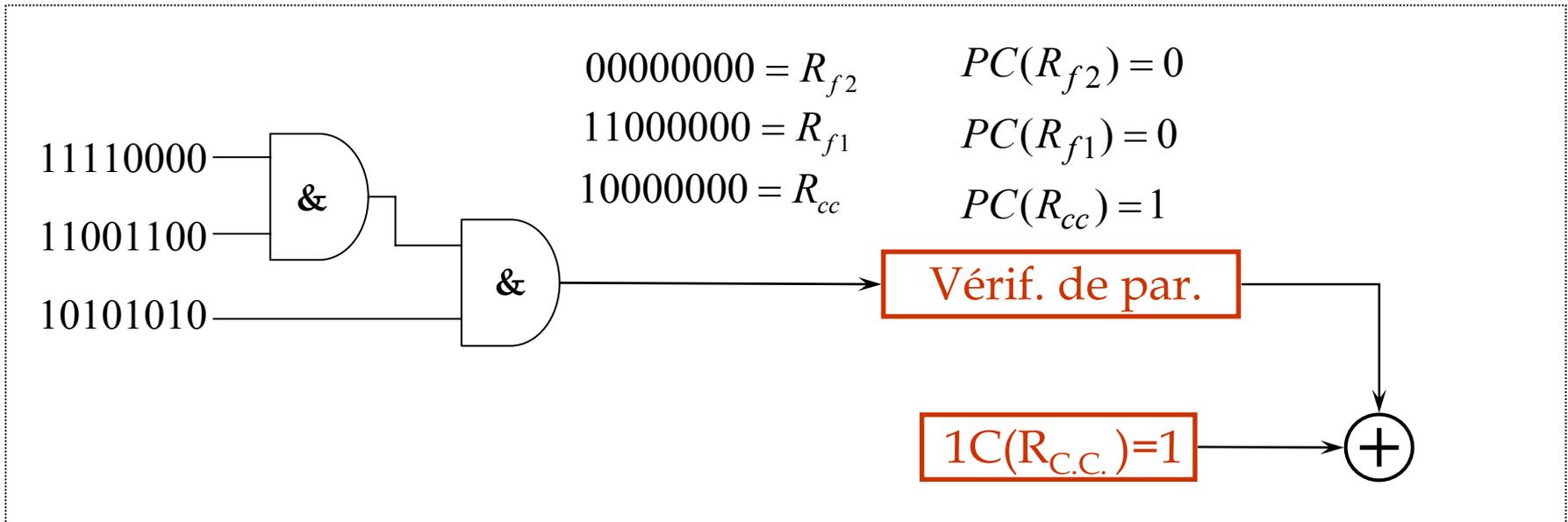
# Vérification de la parité



La fonction de compression est :  $PC(R) = r_0 \oplus r_1 \oplus \dots \oplus r_i$        $PC(R) = 0$  ou  $1$

La probabilité de masquage de fautes est :  $P(n) = \frac{2^{n-1} - 1}{2^n - 1} \approx 0,5$

# Vérification de la parité (suite)



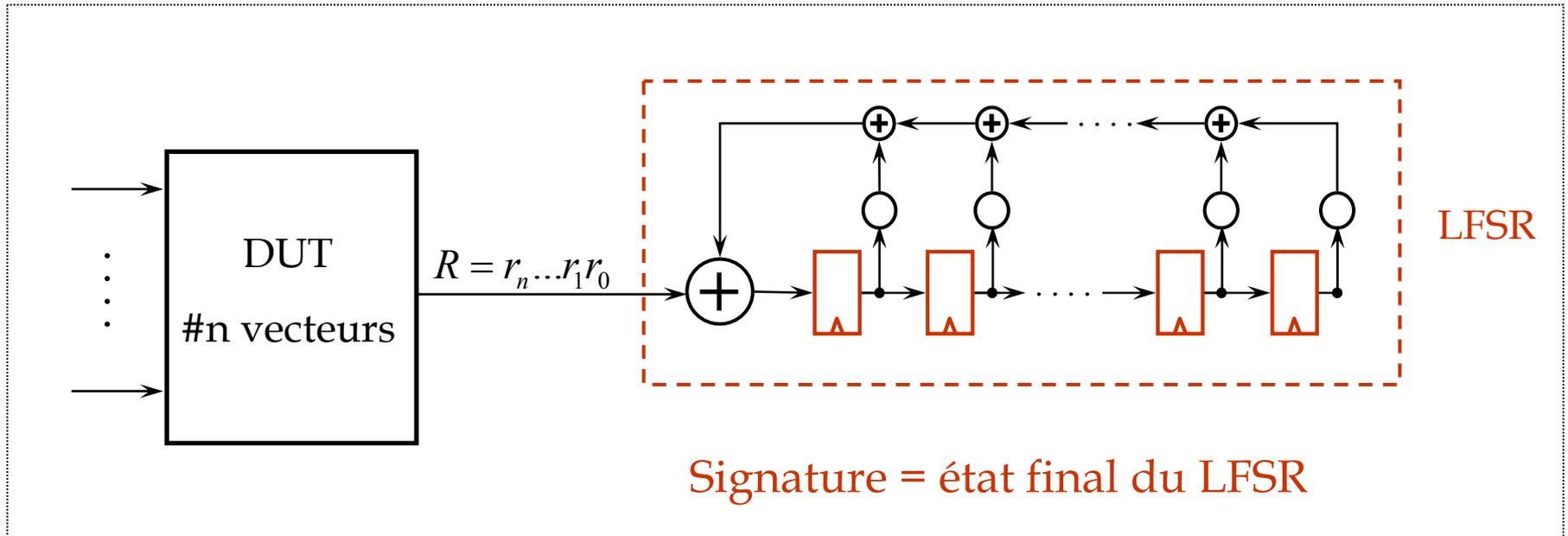
La probabilité de masquage de fautes est :  $P(8) = \frac{2^7-1}{2^8-1} = \frac{14}{255} = 0,498$

☞ On remarque que la probabilité de masquage est plus grande que pour la technique de comptage de transition, mais on détecte les deux fautes de l'exemple.

# Analyse de signatures

- ☞ L'analyse de signature est une technique de compression basée sur l'utilisation des registres à décalage
  - La forme la plus simple de cette méthode consiste à utiliser un LFSR à une entrée
  - La signature est le contenu du LFSR après avoir entré le dernier bit de la séquence à analyser

# Analyse de signatures avec LFSR



La signature de la séquence R est :  $R(X) = \text{état final du registre}$

La probabilité de masquage de fautes est :  $P(n) = \frac{2^{n-m} - 1}{2^n - 1} \approx \frac{1}{2^m}$

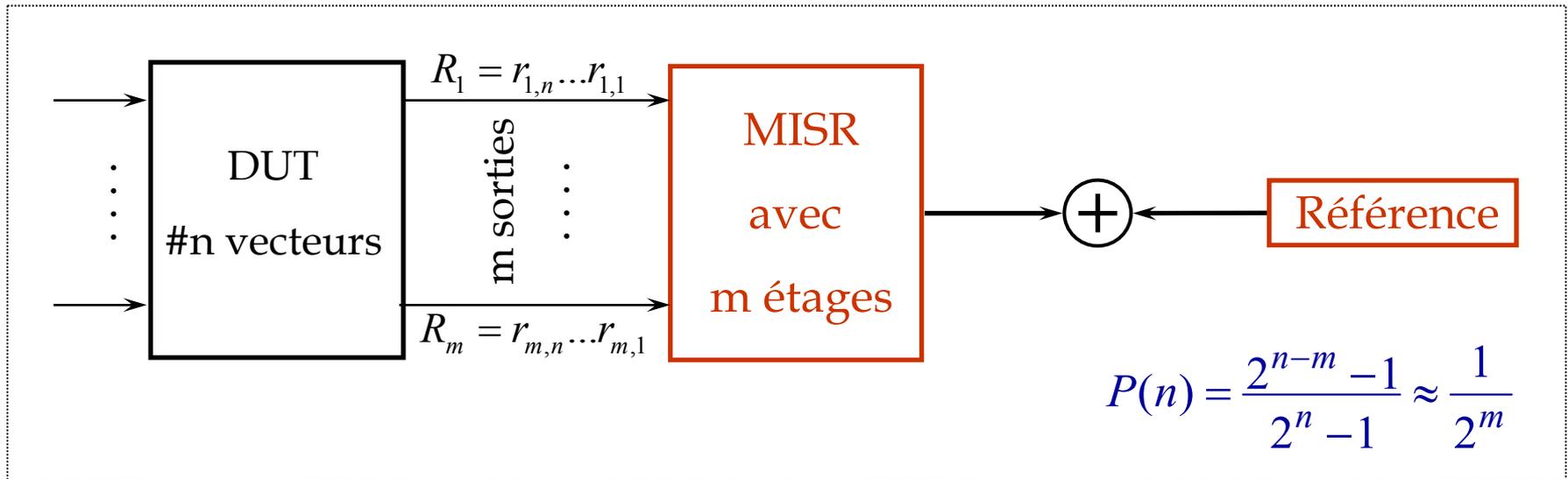
avec  $m = \text{longueur du LFSR}$

# Analyse de signatures avec LFSR (suite)

➡ Pour un circuit à plusieurs sorties  $R_1 \dots R_m$  il existe plusieurs adaptations de la technique d'analyse de signatures avec un LFSR qui sont :

- 1- LFSR avec un multiplexeur
- 2- LFSR avec un XOR
- 3- LFSR à entrées multiples MISR

# Analyse de signature avec MISR



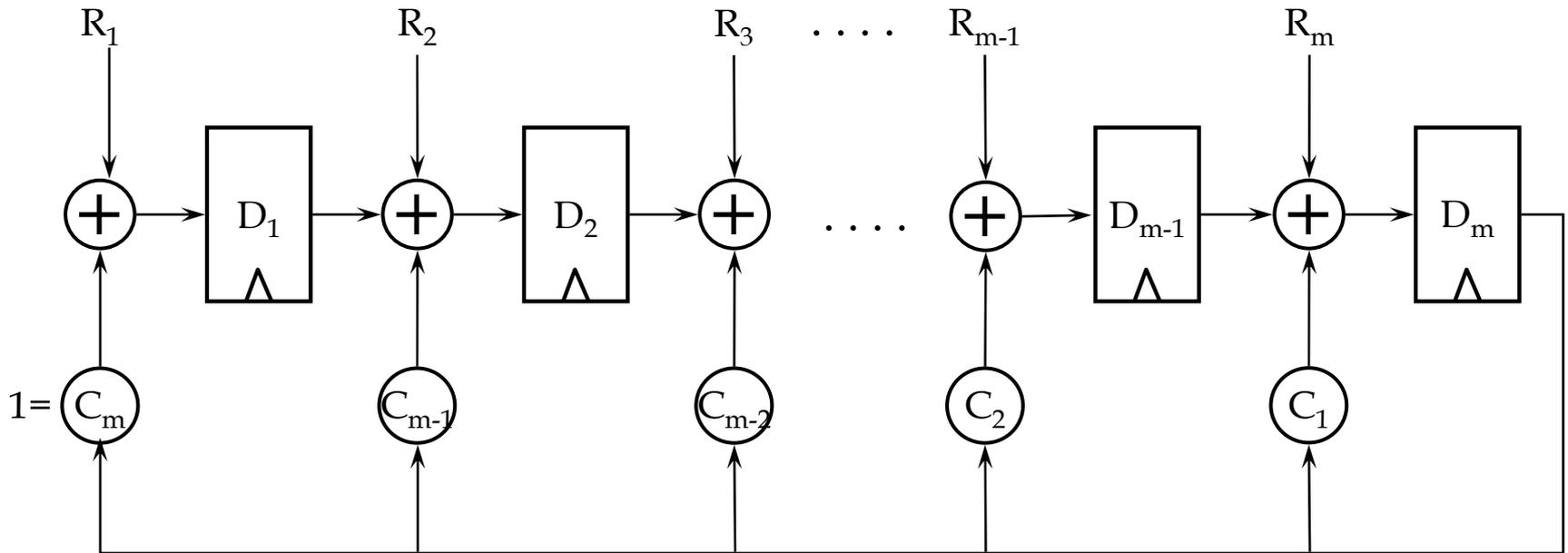
## 👉 Multiple Input Shift Register « MISR »

- On utilise un MISR pour analyser plusieurs sorties à la fois

## 👉 Avantages

- Diminue la surface additionnelles (par rapport à plusieurs LFSRs)
- Diminue le temps de test (par rapport au LFSR avec multiplexeur)

# Multiple Input Shift Register MISR



- Le polynôme caractéristique du MISR peut être non primitif, mais des études ont montré que les MISRs avec des polynômes caractéristiques primitifs ou complexes sont meilleurs

# Built-In Logic Block Observer « BILBO »

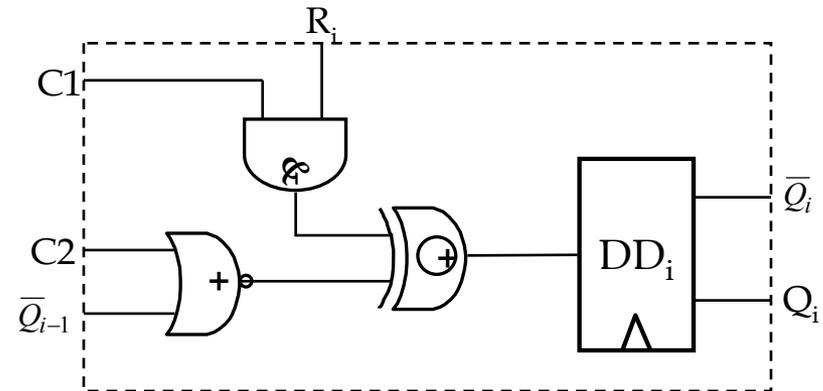
☞ Combine les registres avec scan, le générateur de vecteurs de test et l'analyseur de signature

☞ Quatre modes de fonctionnement

- Normal
- Générateur LFSR et analyseur de signature MISR
- SCAN et Reset

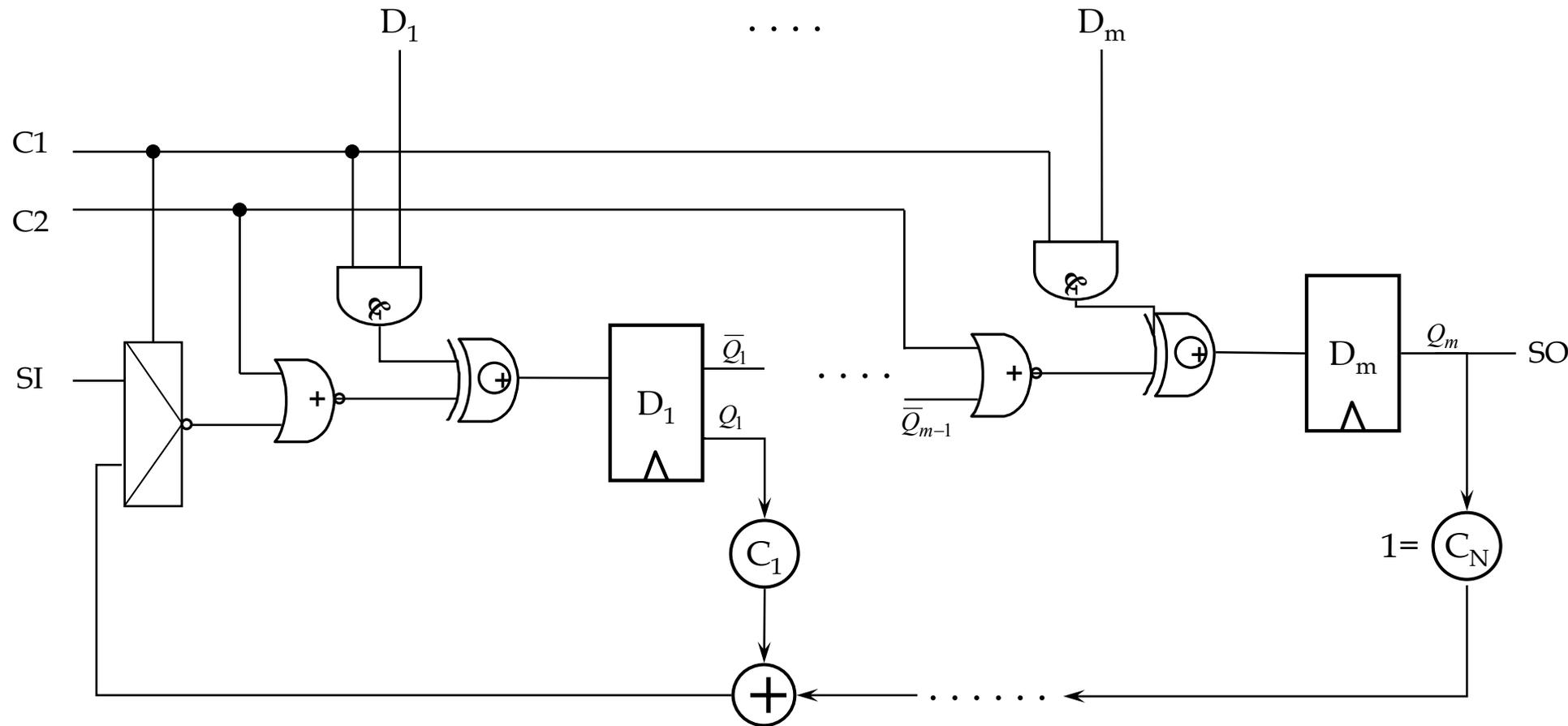
Mode		Fonction
C1	C2	
0	0	SCAN
0	1	Reset
1	0	LFSR/MISR
1	1	Normal

Modes de fonctionnement



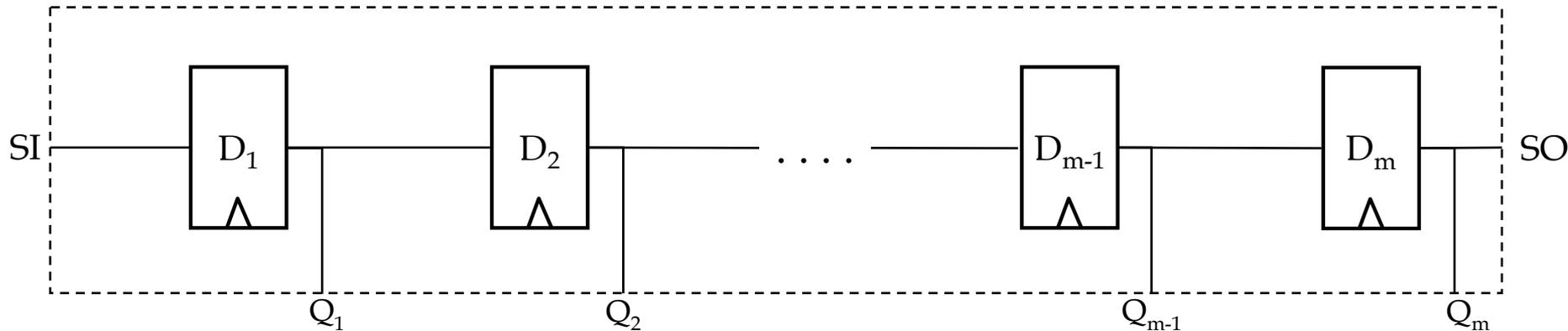
Cellule BILBO

# Built-In Logic Block Observer (suite)

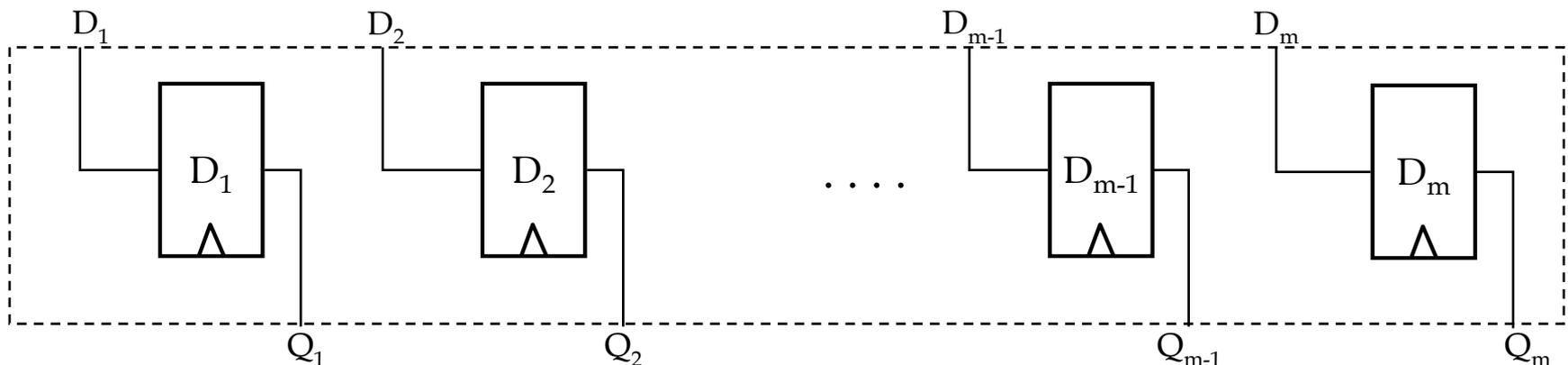


# Built-In Logic Block Observer (suite)

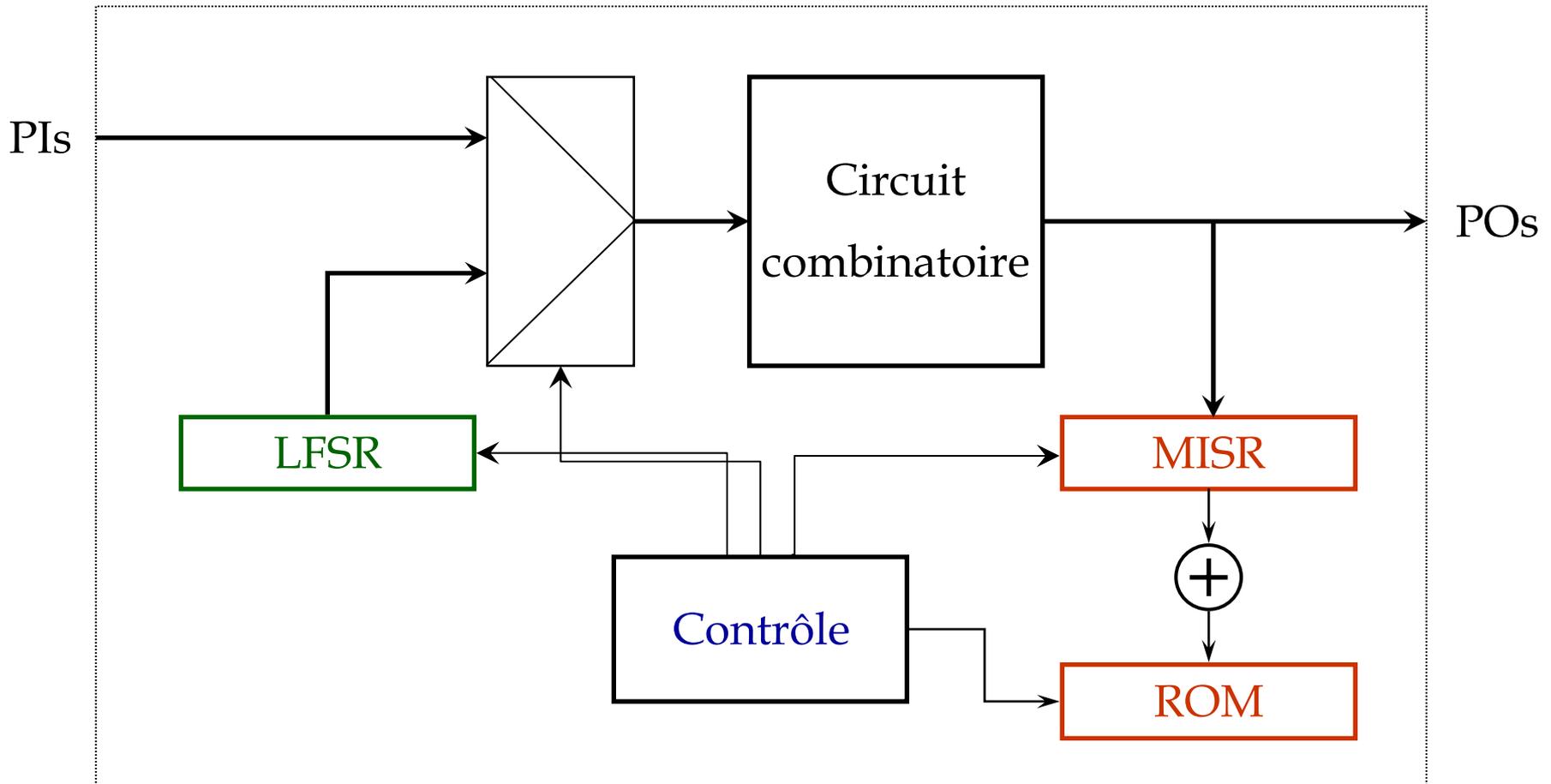
☞  $C1=C2=0$  (Scan)



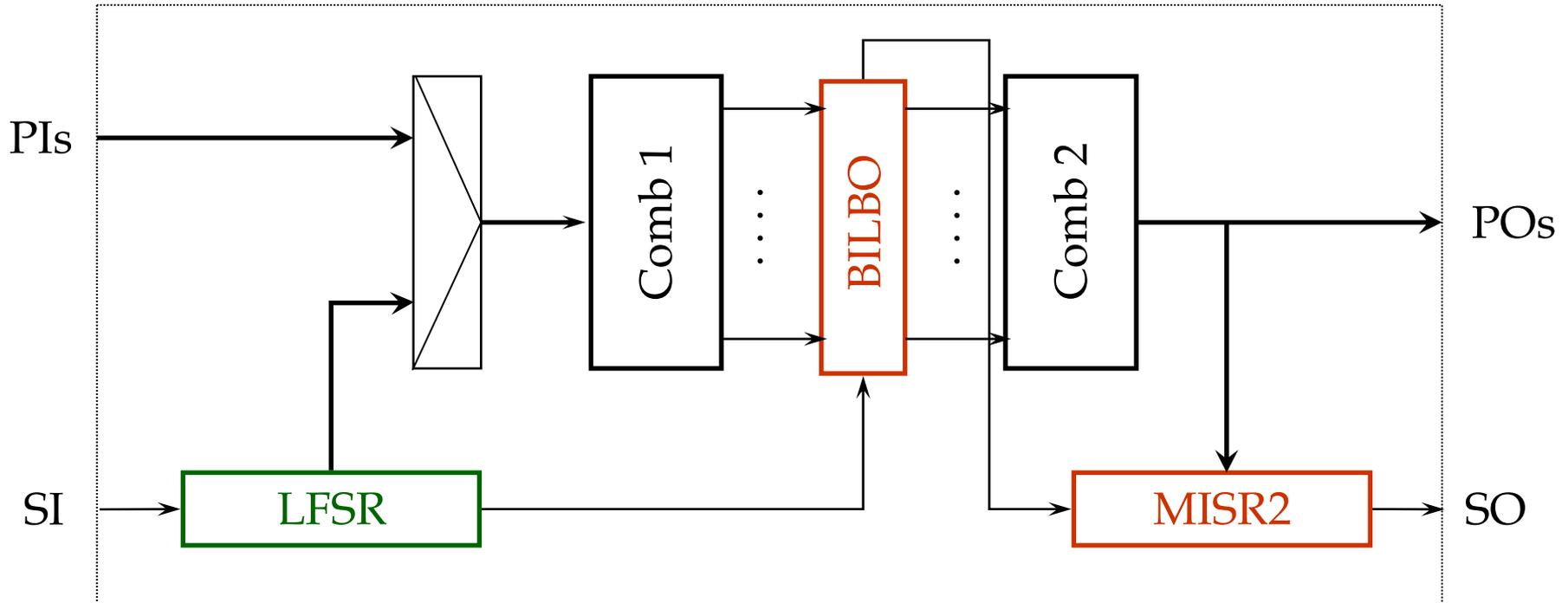
☞  $C1=C2=1$  (normal)



# BIST avec LFSR et MISR



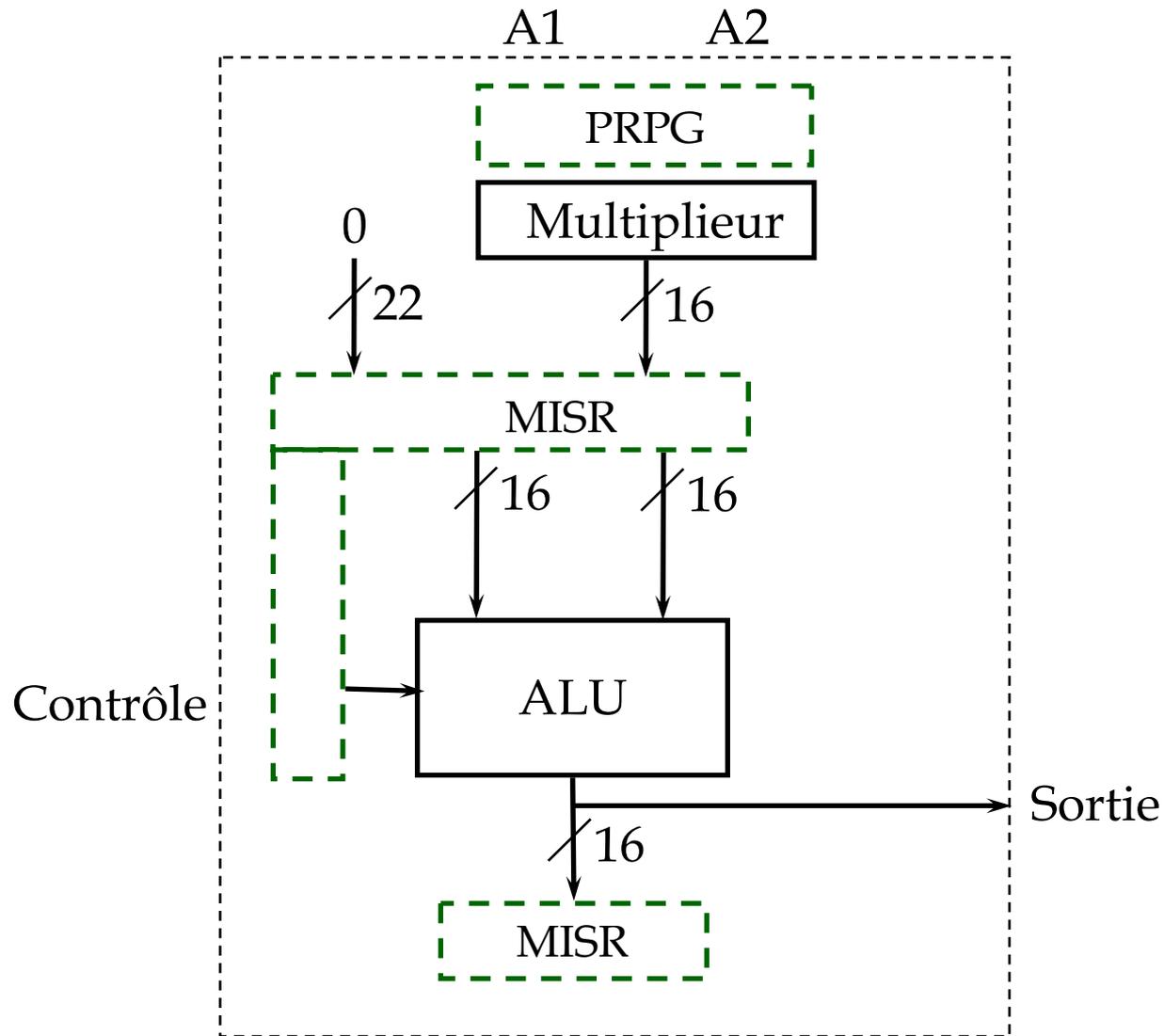
# Test intégré avec BILBO



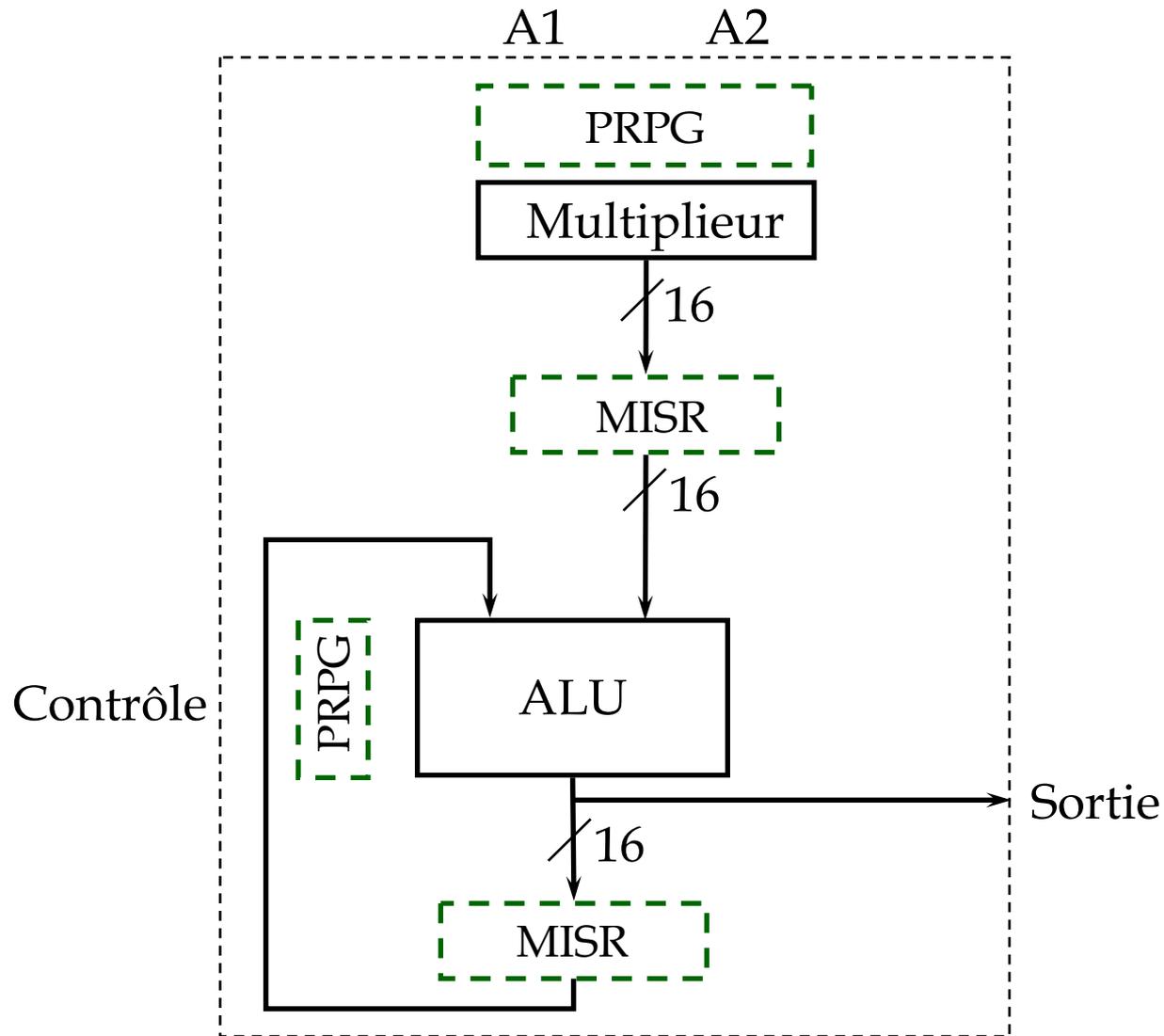
Test intégré avec BILBO pour les circuits séquentiels

- Le BILBO remplace les registres du circuit séquentiel, et est utilisé comme, générateur de vecteur de test LFSR et analyseur de signature MISR

# Exemple d'architecture de BIST : solution 3



# Exemple d'architecture de BIST : solution 4



# Exemple d'architecture de BIST : comparaison

Solution	Nombre de vecteurs			Taux de couverture
	minimum	maximum	moyenne	
1	830	3619	2177	100%
2			>3000	64,5%
3	634	2531	1457	100%
4	721	2121	1378	100%

# Conclusion

## ☞ Définition du BIST

- Avantages et inconvénients
- Architecture

## ☞ Les méthodes et techniques de génération des vecteurs de test

## ☞ Générateur LFSR

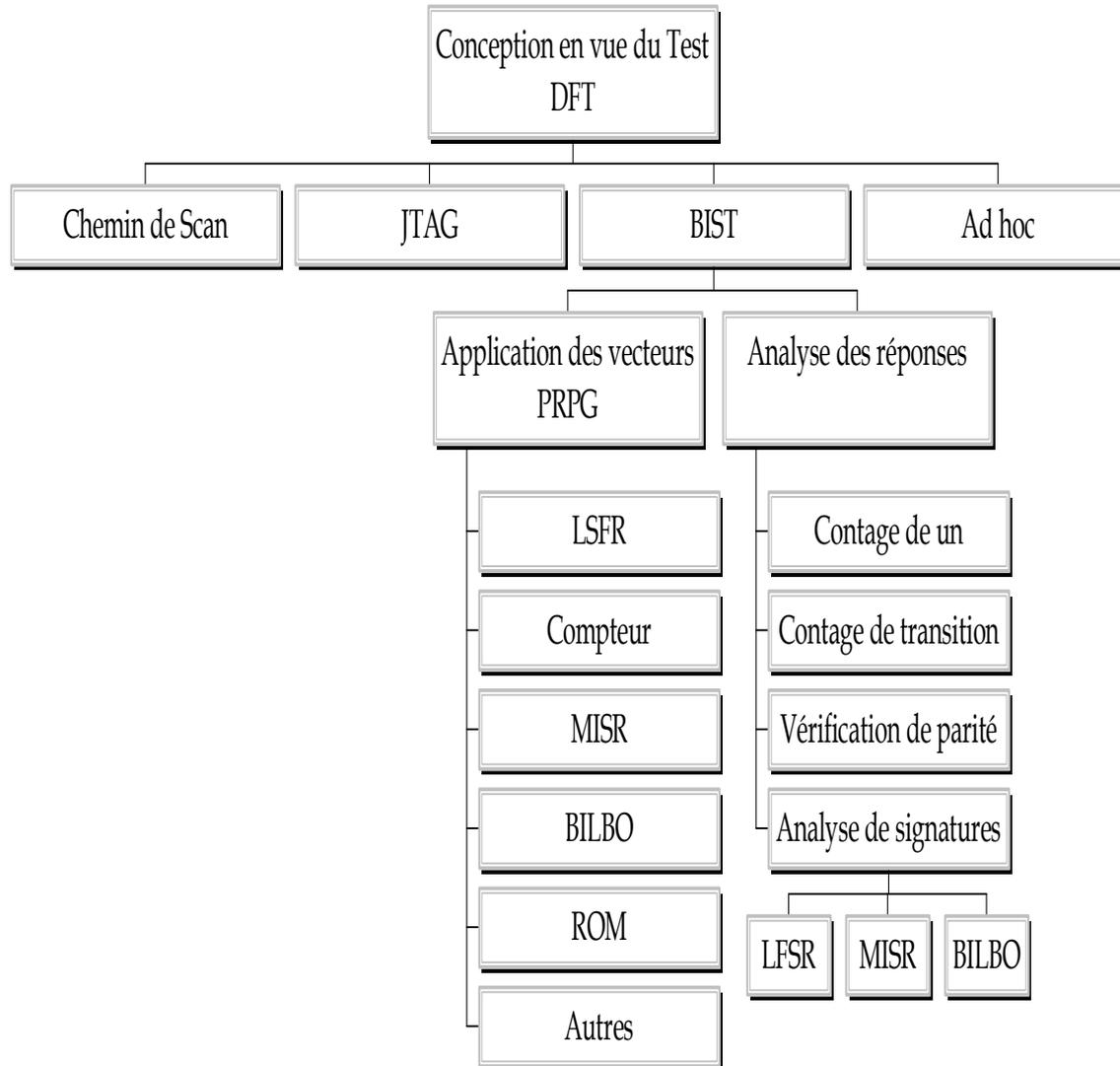
- Théorie et architecture

## ☞ Les techniques d'analyse des réponses

## ☞ Analyse de signature avec MISR

## ☞ Technique BILBO

# Conclusion (suite)



# Documentation

## ATPG and BIST

- Kagaris, D.;
- The VLSI Handbook ; Ed. Wai-Kai Chen, CRC Press LLC, 2000.

## CAD Tools for BIST/DFT and Delay Faults

- Tragoudas, S. ;
- The VLSI Handbook. Ed. Wai-Kai Chen, CRC Press LLC, 2000.

## DS-LFSR: a BIST TPG for low switching activity

- Wang, S.; Gupta, S.K.;
- IEEE Transactions on Computer-Aided Design of Integrated
- Circuits and Systems, July 2002 Page(s): 842 -851

## An analysis of the probabilistic behavior of linear feedback signature registers

- Ivanov, A.; Agarwal, V.K.;
- IEEE Transactions on Computer-Aided Design of Integrated
- Circuits and Systems, Oct. 1989, pp. 1074 -1088

## Linear Feedback Shift Register v3.0

- Xilinx.

# Questions

