

# Chapitre 8

## Conception en Vue du Test : Boundary-Scan (JTAG)

*Abdelhakim Khouas*

Département de Génie Électrique  
École Polytechnique de Montréal



*le génie  
sans frontières*

*A. Khouas*

# Plan

## ☞ C'est quoi le Boundary-Scan (JTAG) ?

- Problèmes de test des cartes
- Norme IEEE 1149.1
- Avantages et coût

## ☞ Architecture du Boundary-Scan

- TAP contrôleur
- Registres Boundary-Scan
- Cellules Boundary-Scan

## ☞ Modes de test

- BYPASS
- EXTEST
- INTEST

## ☞ Langage de description du Boundary-Scan BSDL

# Test des cartes

## 👉 Problématique :

- Pour le test et le diagnostic des cartes, il est important de pouvoir tester séparément les composants et les interconnexions

## 👉 Solution existante :

- Technique « Bed of nails » lit de clous

## 👉 Limitations :

- Routage multiple
- Packages avec plus de 200 pins
- Séparation des pins  $< 0.3$  mm
- Largeur de routage  $< 0.1$  mm

## 👉 Solution :

- Scan-Path des plots d'entrée/sortie des circuits intégrés

# Test des cartes (suite)

***Table 1-2. Time to Develop Test Programs (in Man-Months)†***

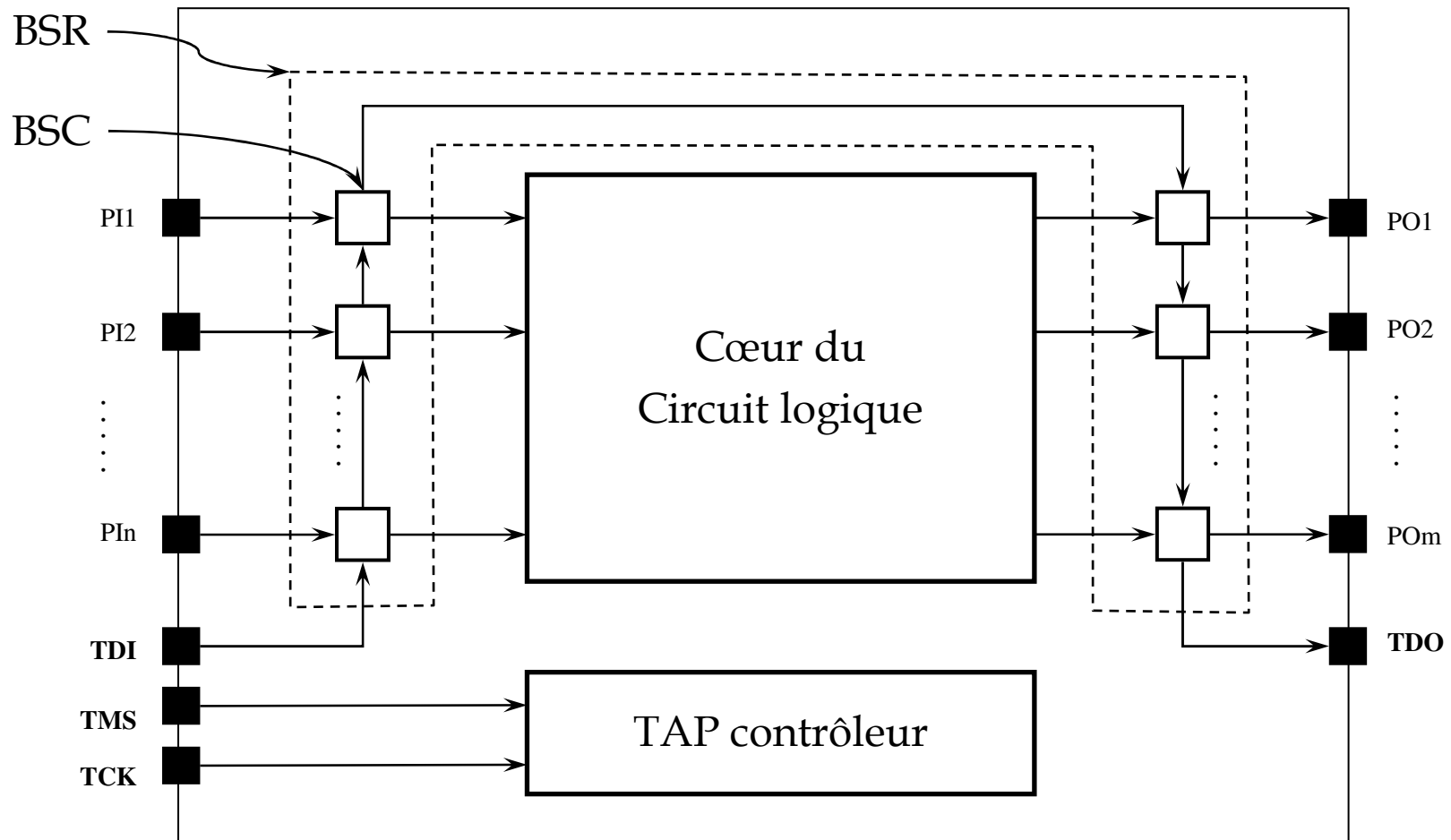
1987–1980 :	3–6 months
1981–1983 :	6–12 months
1984–1986 :	9–18 months
1987–1990 :	12–24 months

†Source: Texas Instruments

# C'est quoi le Boundary-Scan « JTAG »

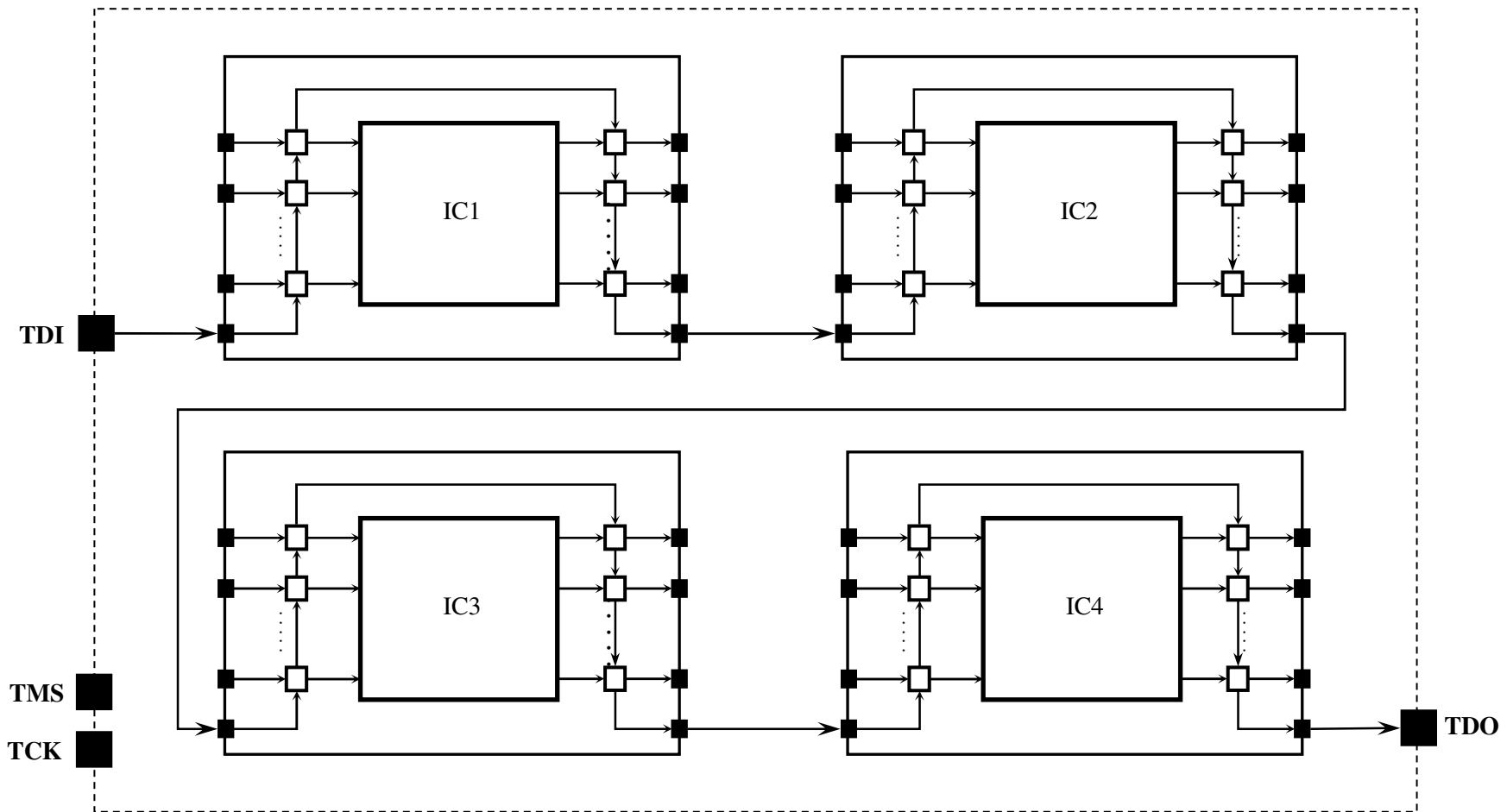
- ☞ C'est une technique de conception en vue du test pour le test des cartes
- ☞ Chaque signal primaire d'entrée et de sortie est complété avec un élément de mémoire appelé cellule Boundary-Scan « BSC »
- ☞ Les cellules BSC sont chaînées entre elles pour former le registre Boundary-Scan « BSR »
- ☞ Technique indépendante du Scan-Path

# C'est quoi le Boundary-Scan (suite)



Circuit avec Boundary-Scan

# C'est quoi le Boundary-Scan (suite)



Carte avec Boundary-Scan

# C'est quoi le Boundary-Scan (suite)

## ☞ Avantages du Boundary-Scan :

- Initialement le Boundary-Scan est prévu pour tester les circuits et les interconnexions sur une carte
  - ☞ Contrôlabilité et observabilité totale de tous les plots d'entrée/sotie de tous les circuits de la carte
  - ☞ Amélioration de la testabilité de la carte
- Utilisé aussi pour le test des circuits intégrés ICS
  - ☞ Permet de faire du diagnostic (test) des ICs sans équipement de test
  - ☞ Facilite l'application du test intégré BIST
- Programmation des systèmes programmables (FPGA)
  - ☞ In-System Programming (ISP)
- Test des systèmes on chip « SOC »
  - ☞ Même technique que pour le test des cartes avec des normes plus adaptées au SOC « Norme IEEE P1500 »



# C'est quoi le Boundary-Scan (suite)

## ☞ Inconvénients du Boundary-Scan :

- Ajout de matériels
  - ↳ Cellules (0.3 % pour le 68040 de Motorola)
  - ↳ Plots (4 plots obligatoires)
  - ↳ Routage
- Perte de rendement du processus de fabrication
- Dégradation des délais (multiplexeur du BSC)
- Manque d'outils de CAO incluant le JTAG
- Rallongement du TTM « Time To Market »

# Histoire du Boundary-Scan

☞ 1985 :

- EJTAG « European Joint Test Action Group » regroupant les fabricants européens de circuits intégrés

☞ 1986 :

- JTAG « Joint Test Action Group »

☞ 1990 : Norme

- IEEE standard test access port and boundary - scan architecture  
IEEE Std 1149.1-1990, 21 May 1990

☞ 1999 :

- IEEE standard for a mixed-signal test bus  
IEEE Std 1149.4-1999 , 28 March 2000

☞ 2000 : Révision

- IEEE standard test access port and boundary-scan architecture  
IEEE Std 1149.1-2001

# Norme IEEE P1149.1

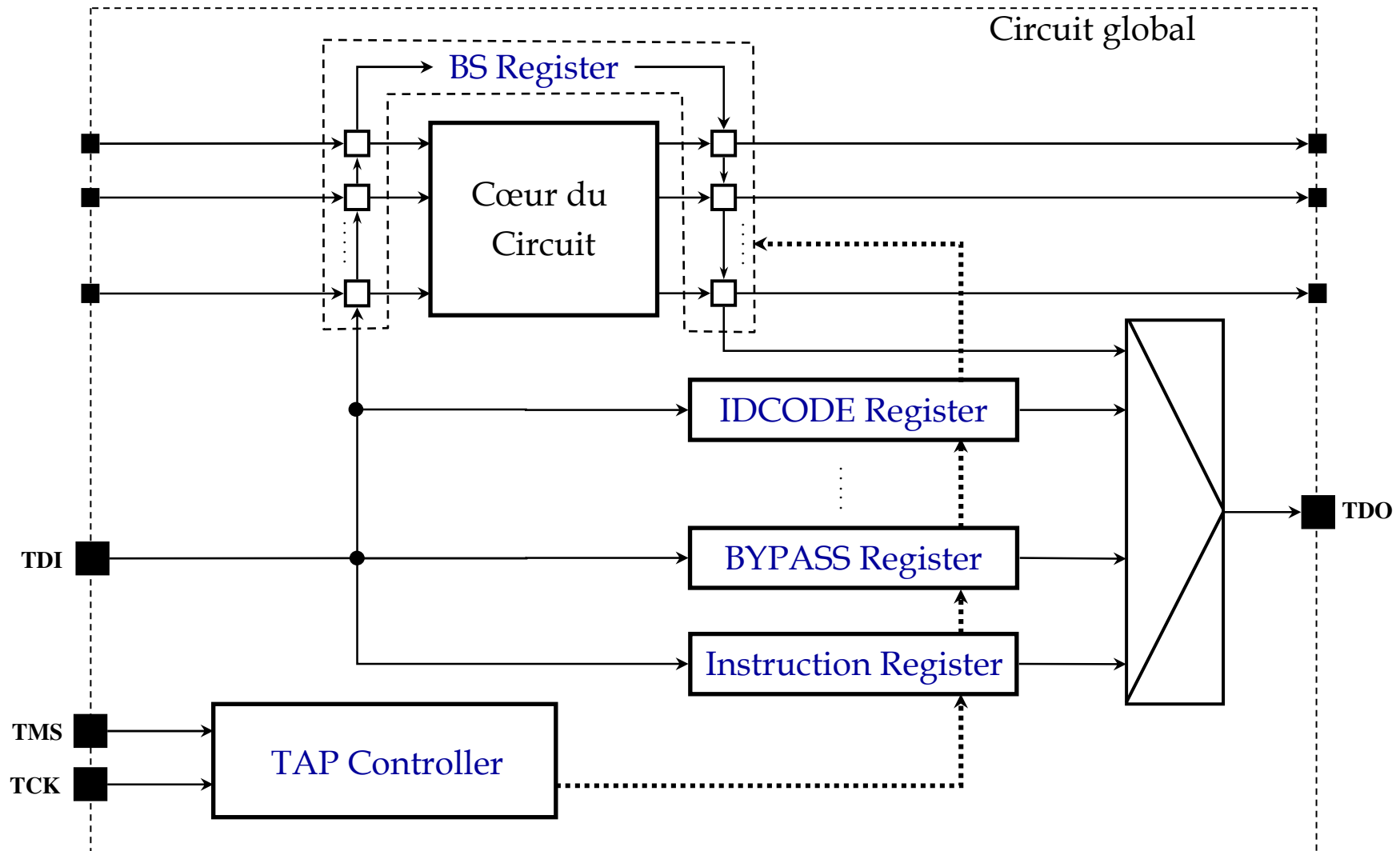
- ☞ Utilisation des circuits de plusieurs fabricants sur la même carte
  - Nécessite l'utilisation de la même norme par tous les fabricants
- ☞ La norme détermine les différents modes de fonctionnement et de communication du Boundary-Scan et non les méthodes d'implémentation

# Architecture de base du BS

Le Boundary-Scan est composé de trois modules :

- ☞ TAP contrôleur
- ☞ Registre instruction IR « Instruction Register »
- ☞ Registres de données DR « Data Register »
  - Registre Bypass
  - Registre Boundary-Scan
  - Registre d'identification Idcode
  - Registre Scan-Path
  - Registre BIST

# Architecture de base du BS (suite)

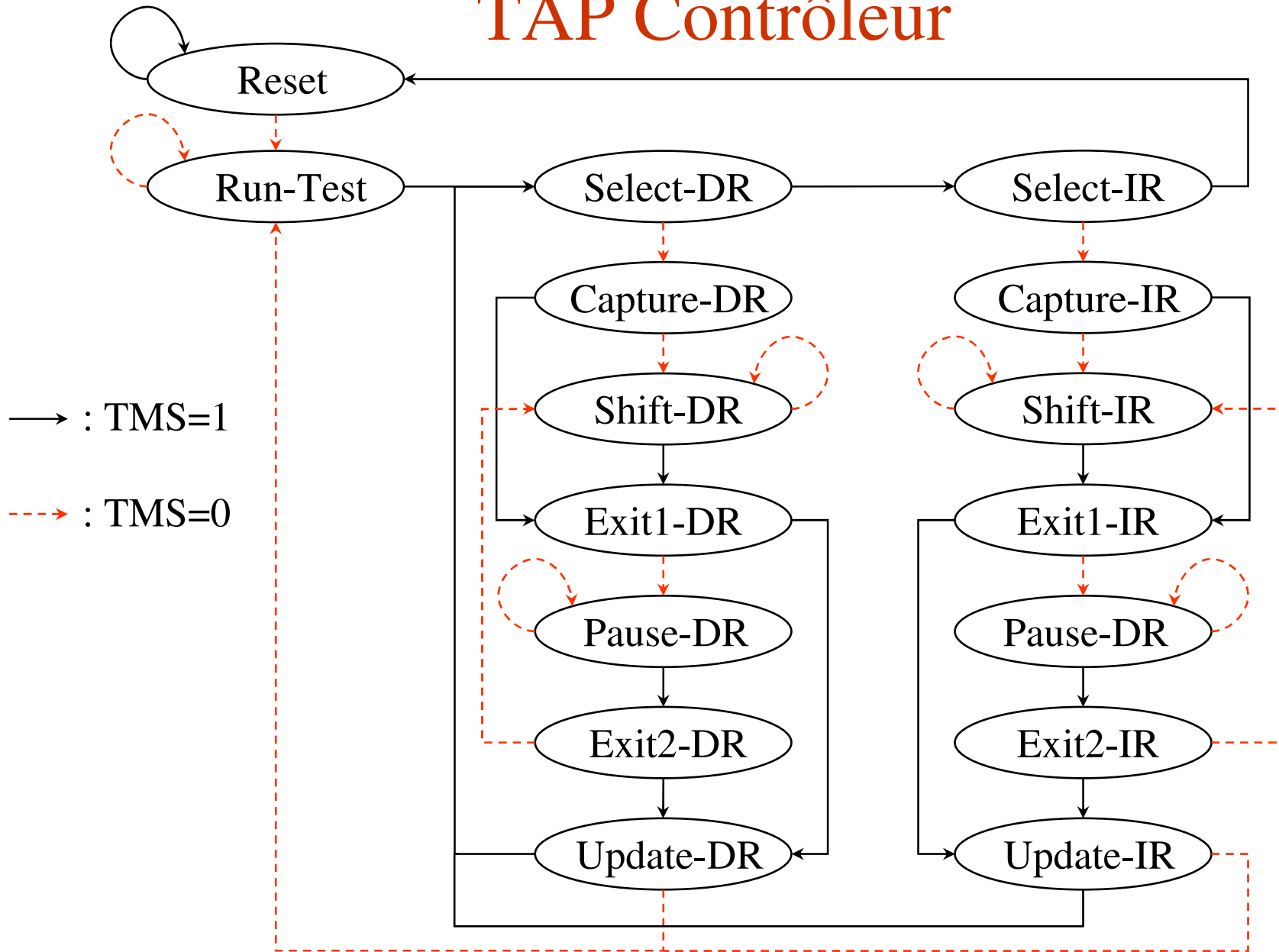


# TAP « Test Access Port »

☞ Il existe 5 plots (pins) d'entrée/sortie (dont un optionnel) qui forment le TAP « Test Access Port » :

- TCK : Signal d'horloge de test « Test Clock »
- TMS : Signal de contrôle « Test Mode Select »
- TDI : Signal d'entrée série « Test Data Input »
- TDO : Signal de sortie série « Test Data Output »
- TRST (optionnel) : Signal de reset « Test Reset »

# TAP Contrôleur



# TAP Contrôleur (suite)

☞ Dans le diagramme de la machine d'état du TAP contrôleur il existe deux colonnes identiques qui sont :

- Colonne d'instruction IR « Instruction Register »
  - ↳ Cette partie du diagramme permet de configurer le registre instruction qui permet de choisir le mode de fonctionnement du Boundary-Scan
- Colonne de données DR « Data Register »
  - ↳ Cette partie du diagramme permet de charger les registres de données selon le mode de fonctionnement choisi, ce mode est déterminé par le registre instruction



# TAP Contrôleur (suite)

☞ Les états du TAP contrôleur sont :

- ❖ Test-Logic-Reset : État de reset
- ❖ Run-Test-Idle : État de mise en attente
- ❖ Select-DR/IR : État temporaire pour choisir le registre instruction/donnée ou le reset
- ❖ Capture-IR/DR : Le registre à décalage IR/DR est chargé en parallèle sur front montant de TCK
- ❖ Shift-IR/DR : Les valeurs sont décalées de l'entrée TDI vers la sortie TDO dans le registre à décalage IR/DR sur front montant de TCK

# TAP Contrôleur (suite)

☞ Les états du TAP contrôleur sont (suite) :

- ❖ Exit1-IR/DR : État temporaire
- ❖ Pause-IR/DR : État de pause (attente de valeurs sur TDI)
- ❖ Exit2-IR/DR : État temporaire
- ❖ Update-IR/DR : Les valeurs contenues dans le registre à décalage IR/DR sont chargées dans le registre parallèle IR/DR sur front descendant de TCK. Prise en compte des nouvelles données par le circuit

# TAP Contrôleur (suite)

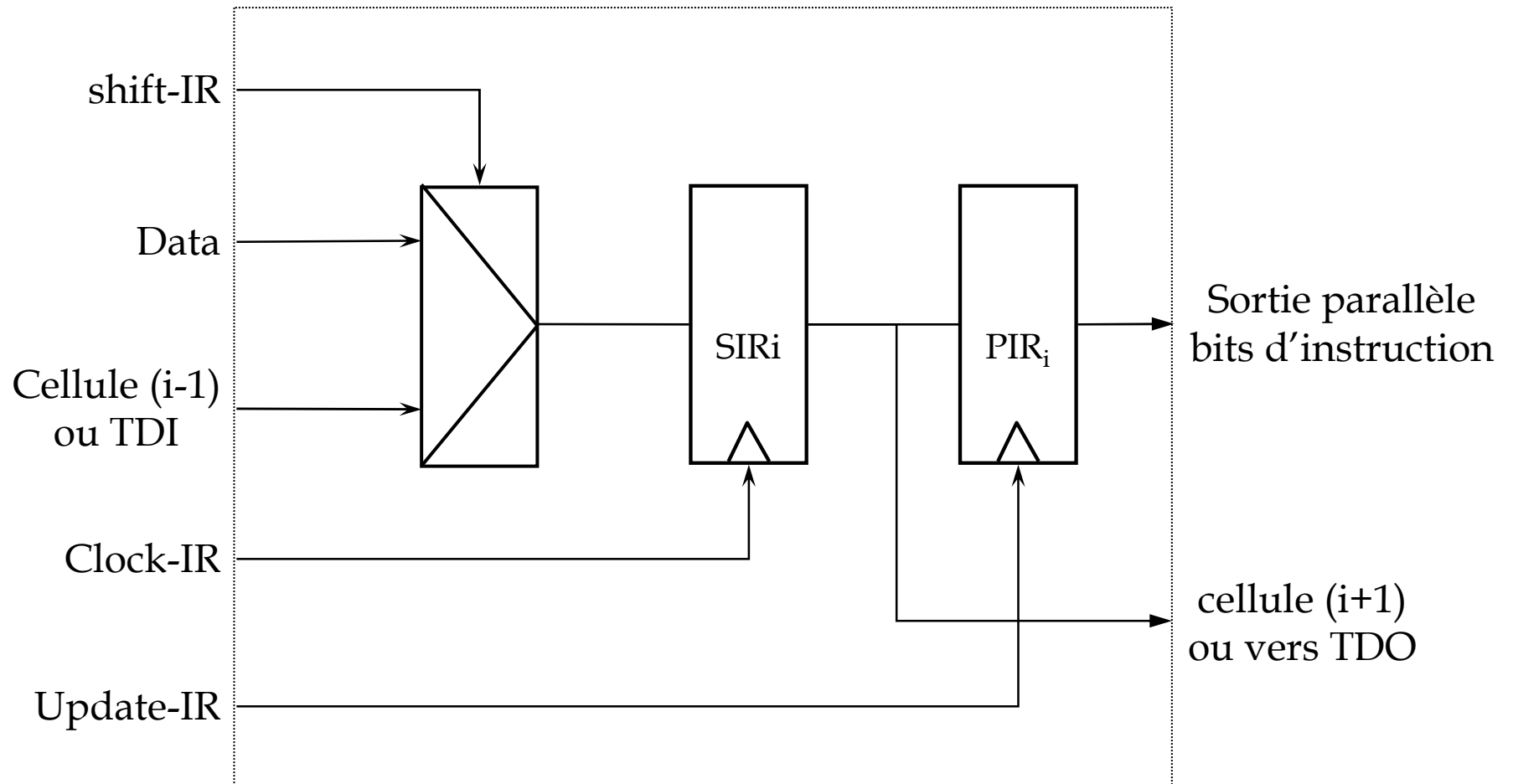
## ☞ Remarques :

- ❖ Les états Shift-IR et Shift-DR activent le driver pour le pin TDO, tous les autres états mettent le driver de TDO en haute impédance
- ❖ Les états Update-IR et Update-DR transfèrent les données du registre de décalage IR/DR vers le registre parallèle IR/DR sur front descendant de TCK
- ❖ Les données sur TDI sont décalées sur front montant de TCK, les données sont décalées à la fin de l'état Shift-IR/DR
- ❖ Les données sur TDO sont décalées sur front descendant de TCK, la donnée est présente sur TDO 1/2 cycle avant que la donnée sur TDI ne soit lue

# Registre instruction « IR »

- ☞ Le registre instruction détermine le mode de fonctionnement du Boundary-Scan. Comme pour les autres registres de la norme 1149.1, il est composé de deux registres :
  - Un registre à décalage :
    - ☞ Ce registre est initialisé à l'état de Capture-IR et décalé de TDI vers TDO à l'état de Shift-IR
    - ☞ Les deux registres les moins significatifs doivent être chargés avec la valeur '01' à l'état Capture-IR.
    - ☞ Ces deux bits seront utilisés pour tester la logique du 1149.1
  - Un registre parallèle :
    - ☞ Ce registre est chargé en parallèle par le contenu du registre à décalage à l'état de Update-IR

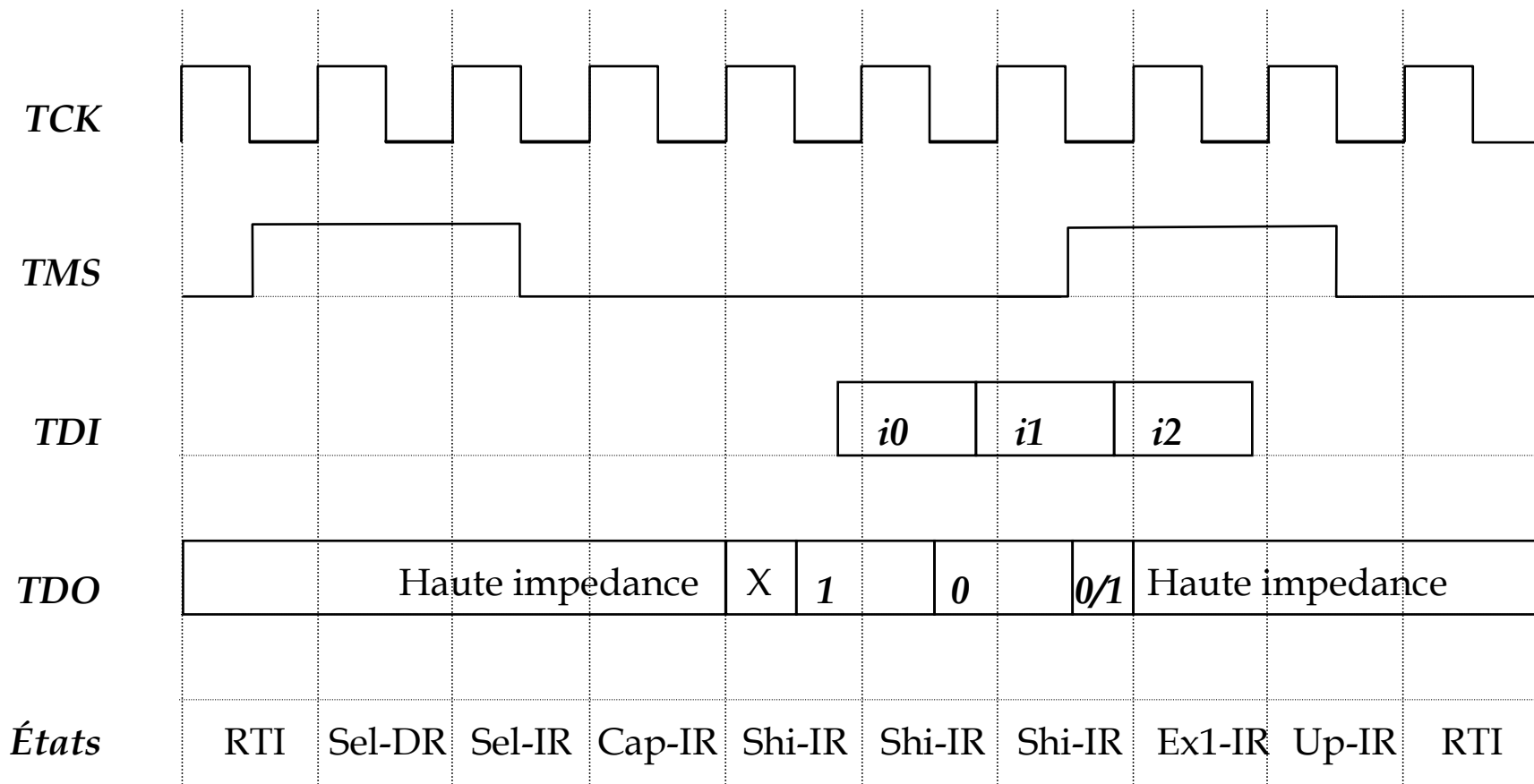
# Cellule du registre instruction



# Opérations du registre instruction

État du TAP Contrôleur	Registre a décalage	Registre Parallèle
Test-Logic-Reset	Indéfini	IDCODE s'il existe, BYPASS sinon
Capture-IR	Chargement de "01" dans LSBs	État précédant
Shift-IR	Décalage des bits d'instruction	État précédant
Exit1-IR	État précédant	État précédant
Pause-IR		
Exit2-IR		
Update-IR	État précédant	Chargement des instructions du registre a décalage

# Chargement d'une instruction



Chronogramme de chargement d'une instruction (3 bits)

# Chargement d'une instruction (suite)

☞ Séquence de chargement d'une instruction de 3 bits :

↵ N° TAP (TMS , TDI , TDO?)	Nouveau état
↵ --- -----	-----
↵ 0 :	RTI
↵ 1 : TAP(1, X, Z)	Select-DR
↵ 2 : TAP(1, X, Z)	Select-IR
↵ 3 : TAP(0, X, Z)	Capture-IR
↵ 4 : TAP(0, X, 1)	Shift-IR
↵ 5 : TAP(0, i0, 0)	Shift-IR
↵ 6 : TAP(0, i1, 0/1)	Shift-IR
↵ 7 : TAP(1, i2, Z)	Exit1-IR
↵ 8 : TAP(1, X, Z)	Update-IR
↵ 9 : TAP(0, X, Z)	RTI



# Registres de données « DR »

☞ Les registres de données représentent l'entité fondamentale du Boundary-Scan. Plusieurs registres peuvent être utilisés

➤ Registre Boundary-Scan :

☞ C'est le registre le plus important, il permet de connecter tous les plots du circuit (sauf les plots du TAP). Ce registre est obligatoire

➤ Registre Bypass :

☞ Ce registre contient une seule cellule Boundary-Scan « BSC » (1 bit).

☞ Ce registre réduit le registre scan à un registre de longueur 1. Ce registre est aussi obligatoire

➤ Registre Idcode :

☞ C'est un registre d'identification, il contient le code d'identification du circuit et il est optionnelle

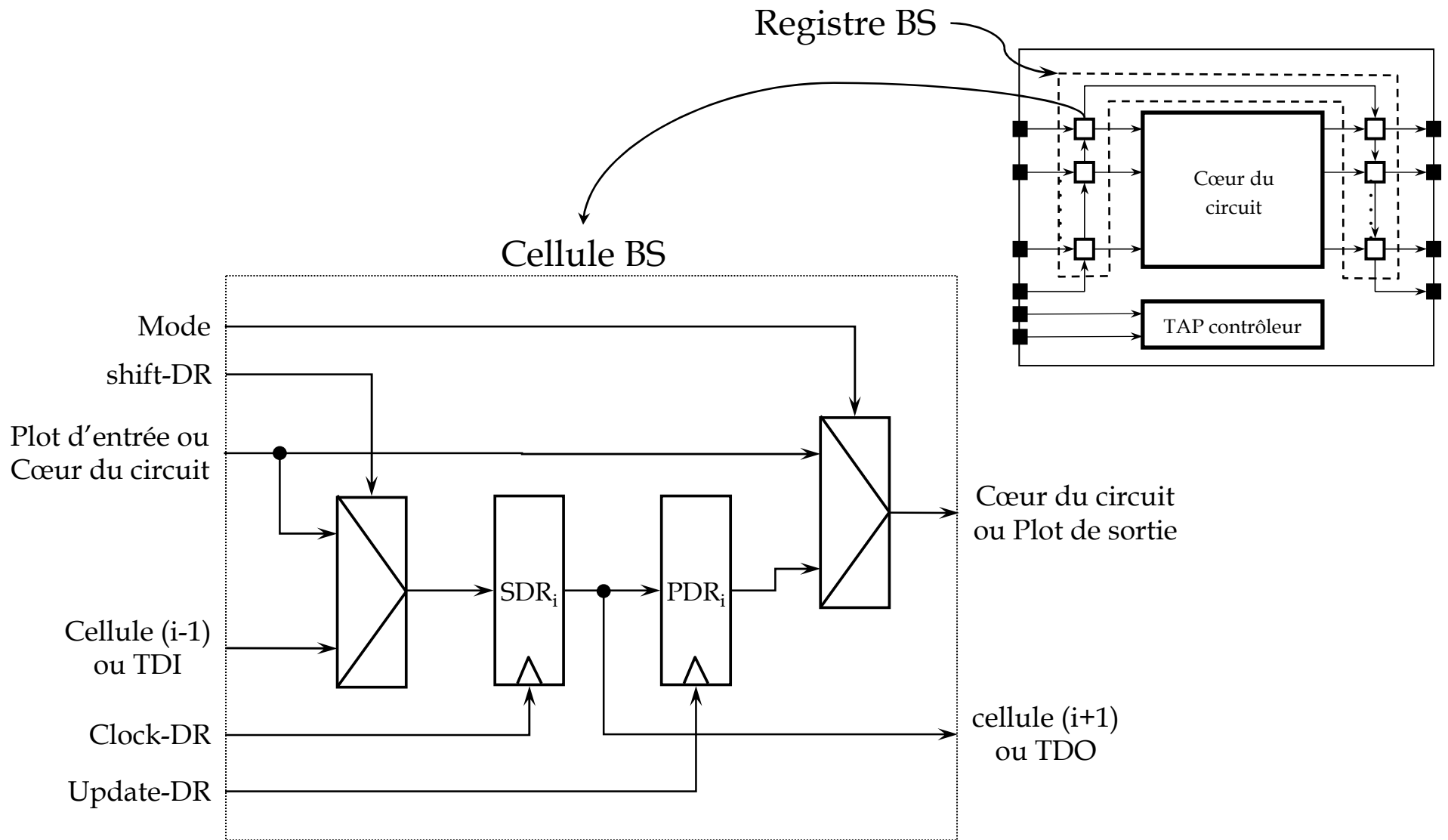
➤ Autres registres

☞ Scan-Path, BIST ...

# Registre Boundary-Scan

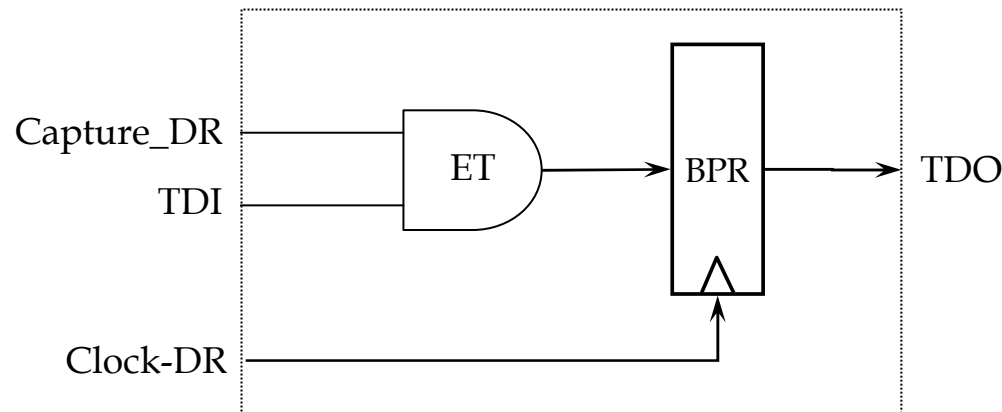
- ☞ Comme pour les autres registres de la norme 1149.1, ce registre est composé de deux registres :
  - Un registre à décalage :
    - ☞ Ce registre est chargé en parallèle à l'état de Capture-DR par :
      - ❖ Les données présentes sur les plots d'entrée pour les cellules d'entrée
      - ❖ Les données présentes sur le cœur du circuit pour les cellules de sortie
    - ☞ Les valeurs du registres sont décalées de TDI vers TDO à l'état de Shift-DR
  - Un registre parallèle :
    - ☞ Ce registre est chargé en parallèle par le contenu du registre à décalage à l'état de Update-DR sur front descendant de TCK

# Cellule du registre Boundary-Scan



# Registre Bypass

- Le registre Bypass est composé d'une seule cellule
- Il permet de traverser le circuits sans le perturber
- Il permet aussi d'atteindre rapidement le circuit cible sur la carte
- A l'état de capture la valeur '0' est chargée dans le registre



# Les modes obligatoires

☞ La norme IEEE 1149.1 impose trois instructions :

➤ 1- BYPASS :

☞ L'instruction Bypass permet de passer au travers les composants avec un bit de décalage. Son code instruction est 11..1

➤ 2- EXTEST :

☞ L'instruction Extest sélectionne le registre Boundary-Scan en déconnectant les plots du cœur du circuit. Elle prépare au test des interconnexions. Son code instruction est 00...0

➤ 3- SAMPLE/PRELOAD :

☞ L'instruction Sample/Preload sélectionne le registre Boundary-Scan mais sans déconnecter les plots du cœur du circuit

☞ La taille minimale du registre instruction est de 2 bits

# Les modes optionnels

☞ Le standard IEEE 1149.1 autorise un certain nombre d'instructions optionnelles, exemples :

➤ INTEST :

☞ Elle sélectionne le registre Boundary-Scan pour le préparer au test du noyau logique interne

➤ IDCODE :

☞ Elle sélectionne le registre d'identification placé entre TDI et TDO pour le préparer au chargement ou déchargement du code Idcode

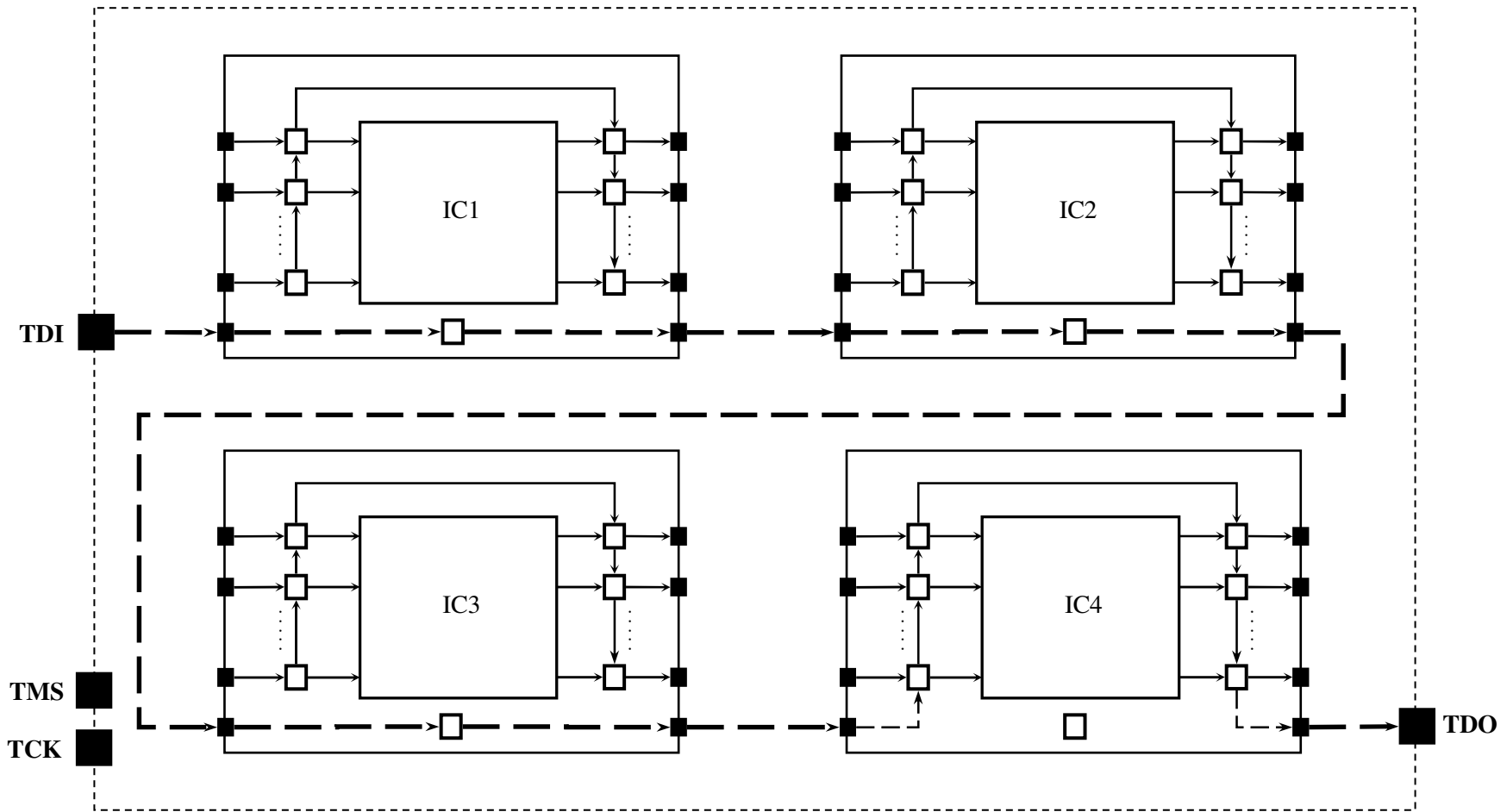
➤ RUNBIST :

☞ C'est l'instruction qui initialise une routine de test interne et charge le résultat dans un registre spécifique placé entre le TDI et le TDO

# Modes d'opération du Boundary-Scan

- ☞ Le TAP contrôleur et ces plots peuvent fonctionner indépendamment du circuit pour les modes suivants :
  - BYPASS
  - IDCODE
  - SAMPLE/PRELOAD
  
- ☞ Les autres modes sont dépendants du circuit :
  - EXTEST
  - INTEST
  - RUNBIST

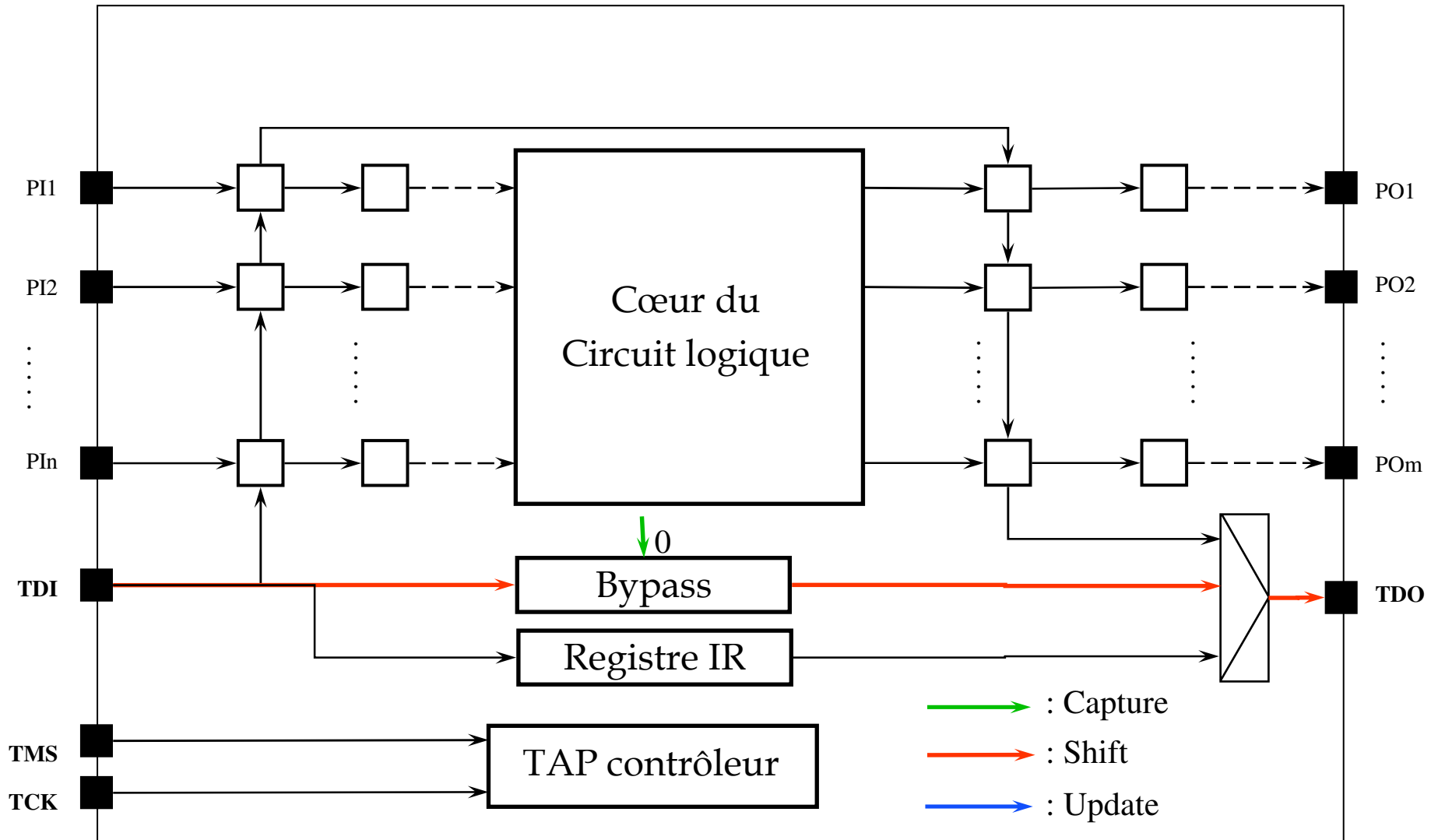
# Mode BYPASS



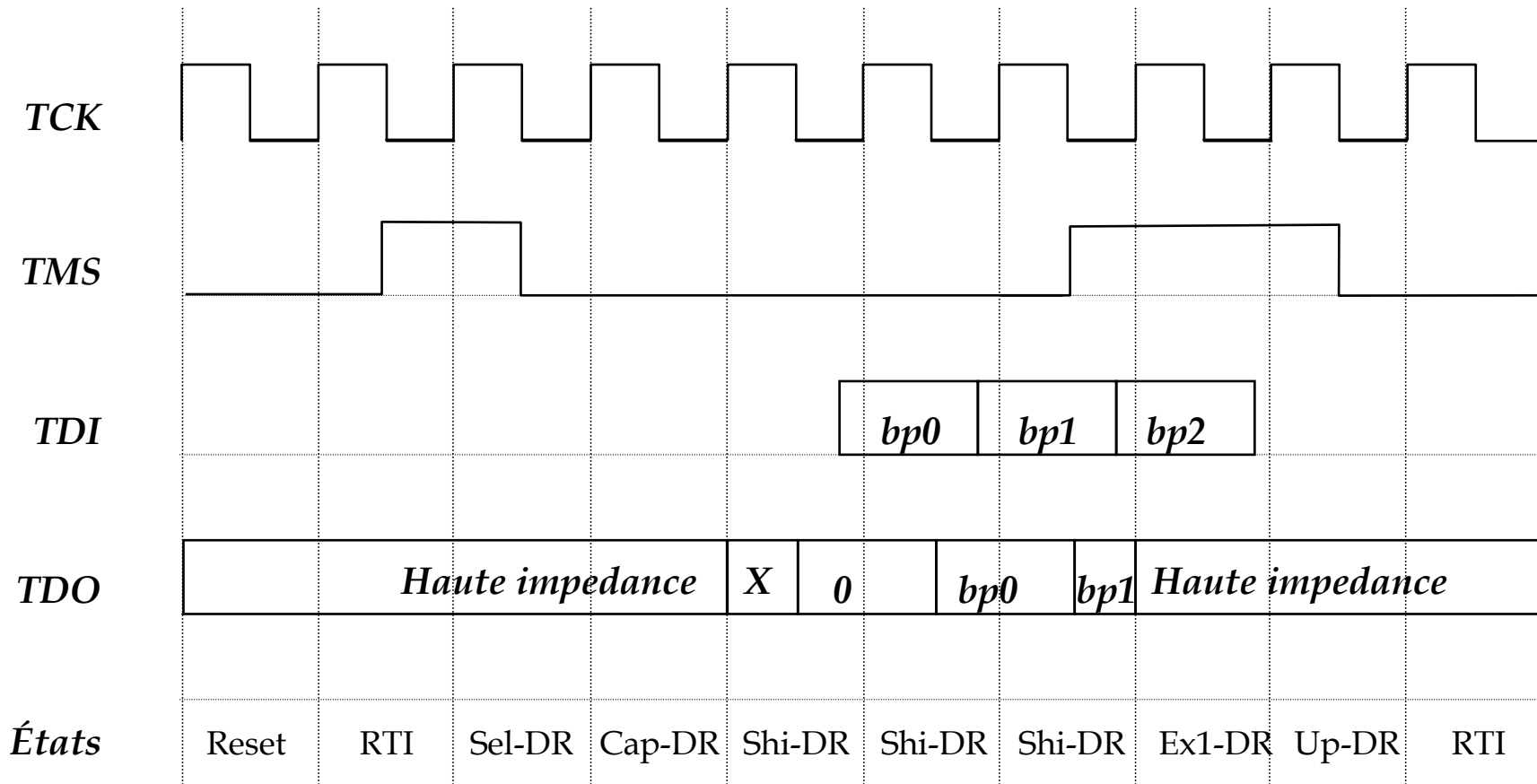
Mode Bypass des circuit IC1, IC2 et IC3



# Mode BYPASS (suite)



# Mode BYPASS (suite)



Chronogramme d'un mode BYPASS

# Mode BYPASS (suite)

## 👉 Séquence d'un mode BYPASS :

↵ N° TAP (TMS , TDI , TDO?)	Nouveau état
↵ --- -----	-----
↵ 0 :	Reset
↵ 1 : TAP(0, X, Z)	RTI
↵ 2 : TAP(1, X, Z)	Select-DR
↵ 3 : TAP(0, X, Z)	Capture-DR
↵ 4 : TAP(0, X, 0)	Shift-DR
↵ 5 : TAP(0, bp0, bp0)	Shift-DR
↵ 6 : TAP(0, bp1, bp1)	Shift-DR
↵ 7 : TAP(1, bp2, Z)	Exit1-DR
↵ 8 : TAP(1, X, Z)	Update-DR
↵ 9 : TAP(0, X, Z)	RTI

# Mode EXTEST

## ☞ Définitions :

- PTV : Vecteur de test parallèle « Parallel Test Vector »
- STV : Vecteur de test séquentiel « Sequential Test Vector »
- PRV : Vecteur de réponse parallèle « Parallel Response Vector »
- SRV : Vecteur de réponse séquentiel « Sequential Response Vector »

☞ Pour le test en mode Boundary-Scan, on utilise le mode séquentiel pour appliquer les vecteurs de test et analyser les réponses des circuits

# Mode EXTEST (suite)

- Step 1 : Initialiser le TAP à Test-Logic-Reset
- Step 2 : Charger l'instruction SAMPLE/PRELOAD
- Step 3 : « Shift » le premier vecteur « STV »
- Step 4 : Charger l'instruction EXTEST
- Step 5 : « Capture » la réponse du vecteur précédant dans le registre BS
- Step 6 : « Shift » la réponse «SRV», en chargeant le vecteur suivant «STV»
- Step 7 : « Update » le nouveau vecteur
- Step 8 : Répéter les étapes 5,6 et 7 jusqu'au dernier vecteur
- Step 9 : « Capture » la réponse du dernier vecteur
- Step 10 : « Shift » la réponse « SRV » du dernier vecteur
- Step 11 : Revenir à l'état Test-Logic-Reset

# Mode EXTEST (suite)

☞ Dans le mode EXTEST :

➤ L'étape de Capture-DR :

☞ permet de copier les données des plots d'entrée (provenant des interconnexions) vers le registre Boundary-Scan

➤ L'étape de Shift\_DR :

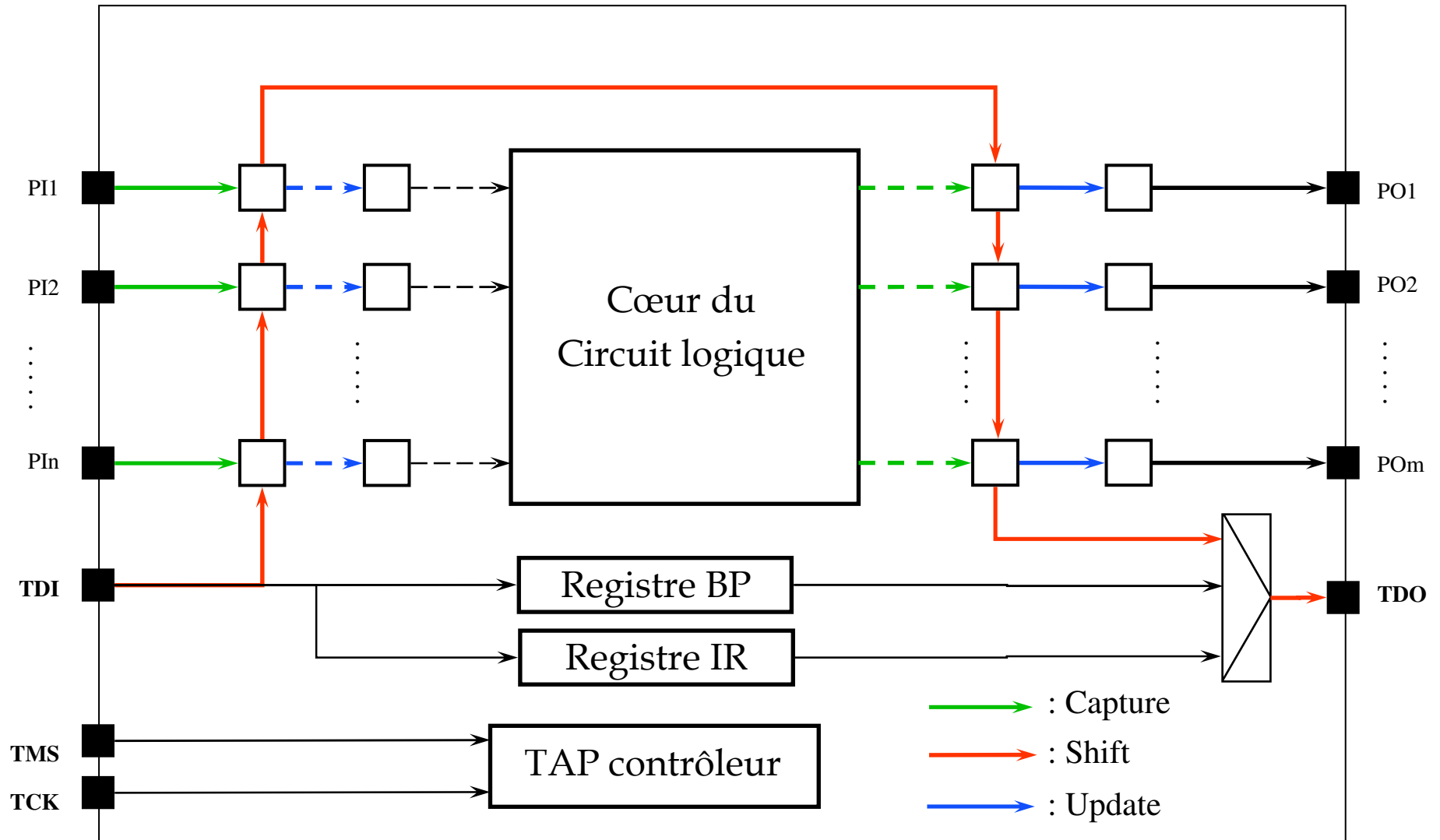
☞ permet de décaler les données copiées pour les analyser tout en entrant de nouvelles données pour les plots de sortie

➤ L'étape Update\_DR :

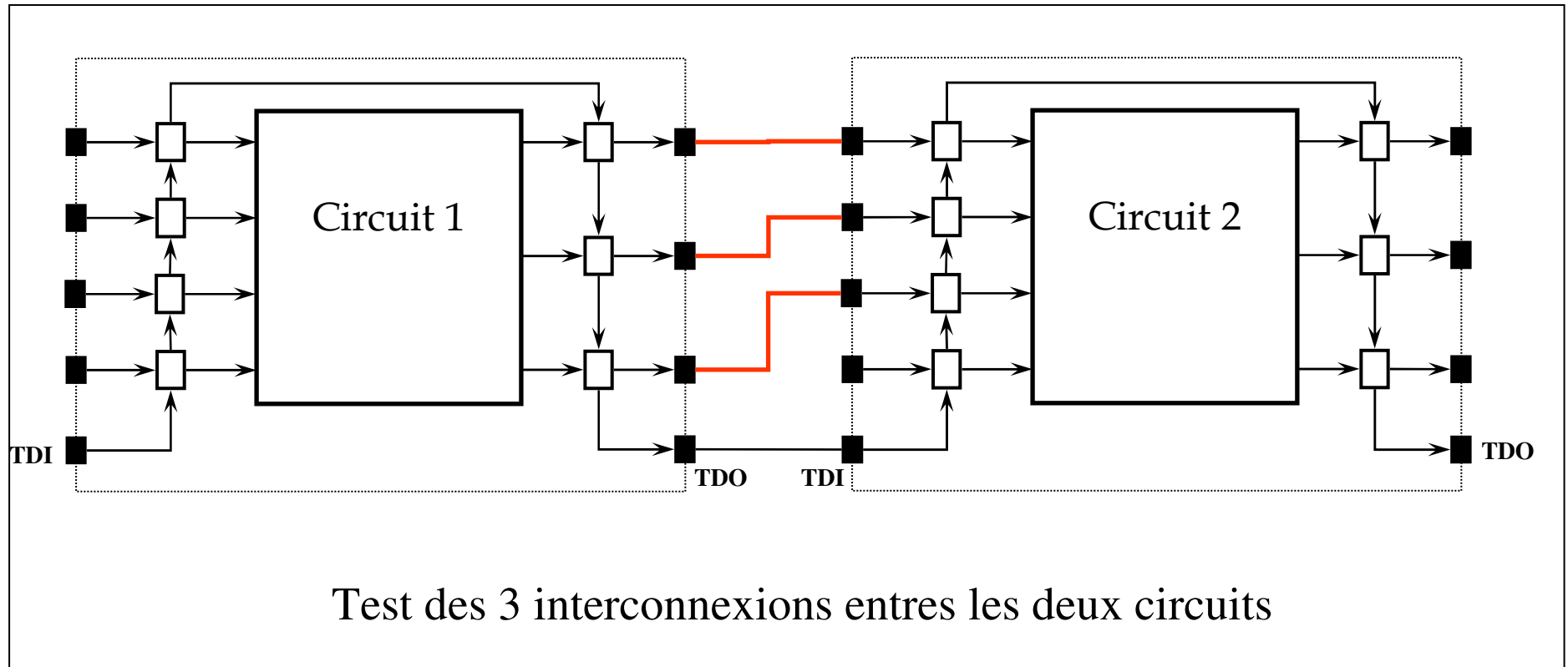
☞ permet de copier les nouvelles données entrées pendant l'étape de shift vers les plots de sorties (donc vers les interconnexions)

☞ Ce mécanisme permet de tester les interconnexions entre les circuits d'une carte

# Mode EXTEST (suite)



# Mode EXTEST (suite)

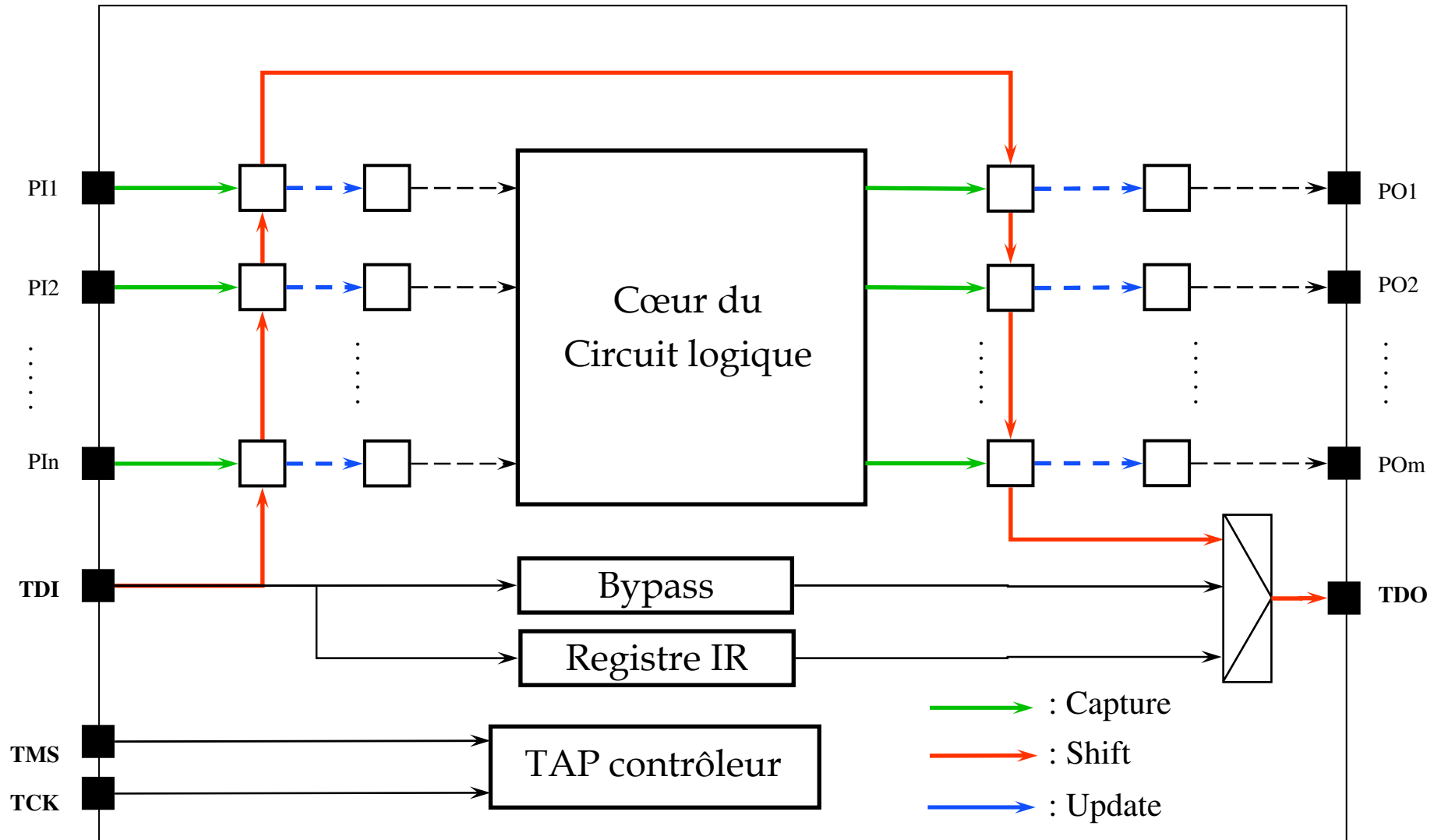


Input1(TDI) : xxxx 101 xxxx xxx  
Output1(TDO) : xxxx xxx x101 xxx

Input2(TDI) : xxxx 011 xxxx xxx  
Output2(TDO) : xxxx xxx x110 xxx



# Mode SAMPLE/PRELOAD



# Mode INTEST

- Appliquer les mêmes étapes que pour le mode EXTEST avec les modification suivantes :
  - Sélectionner le mode INTEST au lieu de EXTEST
  - Contrairement au mode EXTEST, les vecteurs sont à appliquer aux plots d'entrée du circuit et non de sortie
  - Les lectures sont à faire sur les plots de sortie du circuit et non d'entrée

# Mode INTEST (suite)

☞ Dans le mode INTEST :

➤ L'étape de Capture-DR :

☞ permet de copier les données provenant du cœur du circuit logique vers le registre Boundary-Scan

➤ L'étape de Shift\_DR :

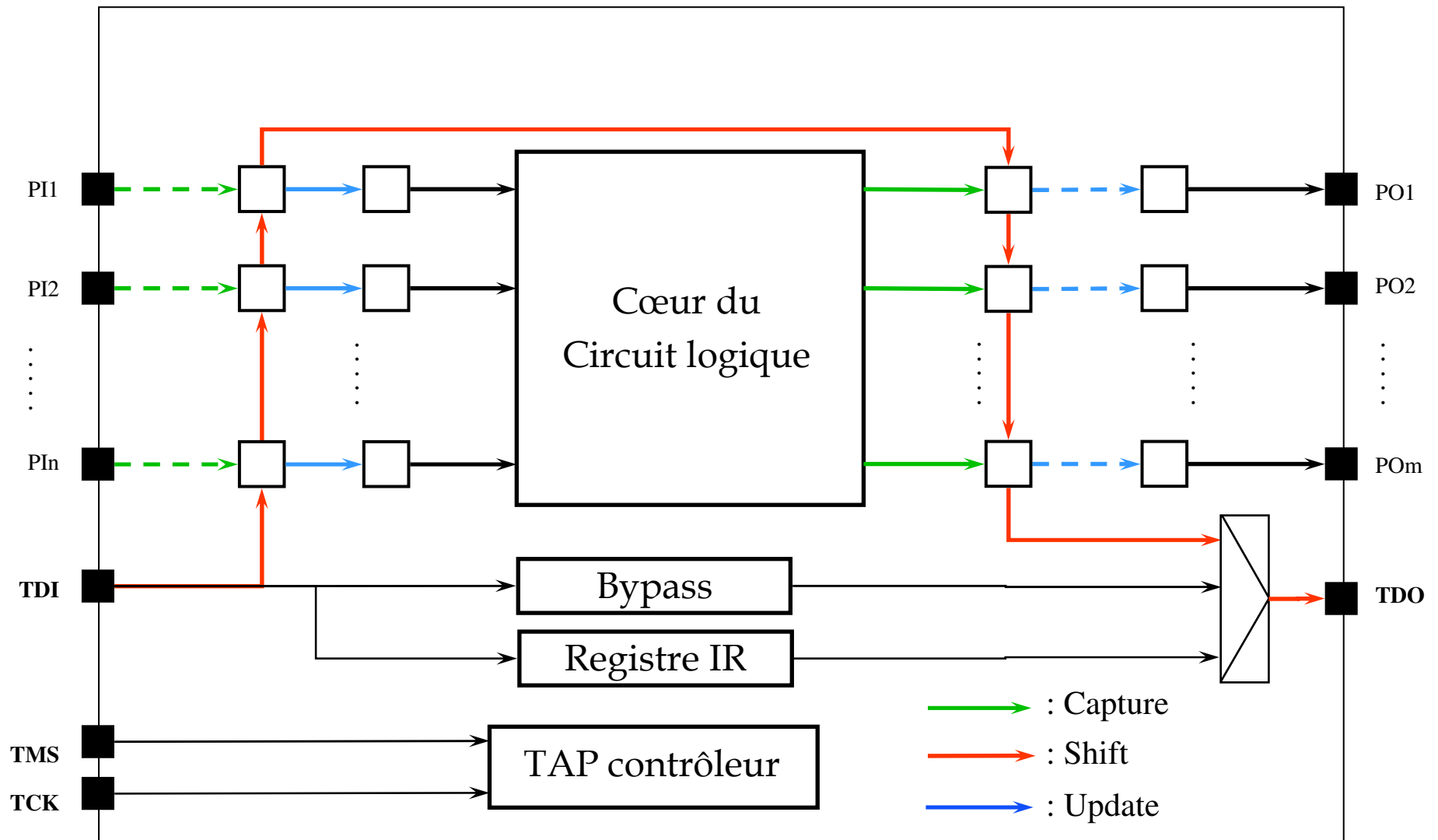
☞ permet de décaler les données copiées pour les analyser tout en entrant de nouvelles données à appliquer au cœur du circuit

➤ L'étape Update\_DR :

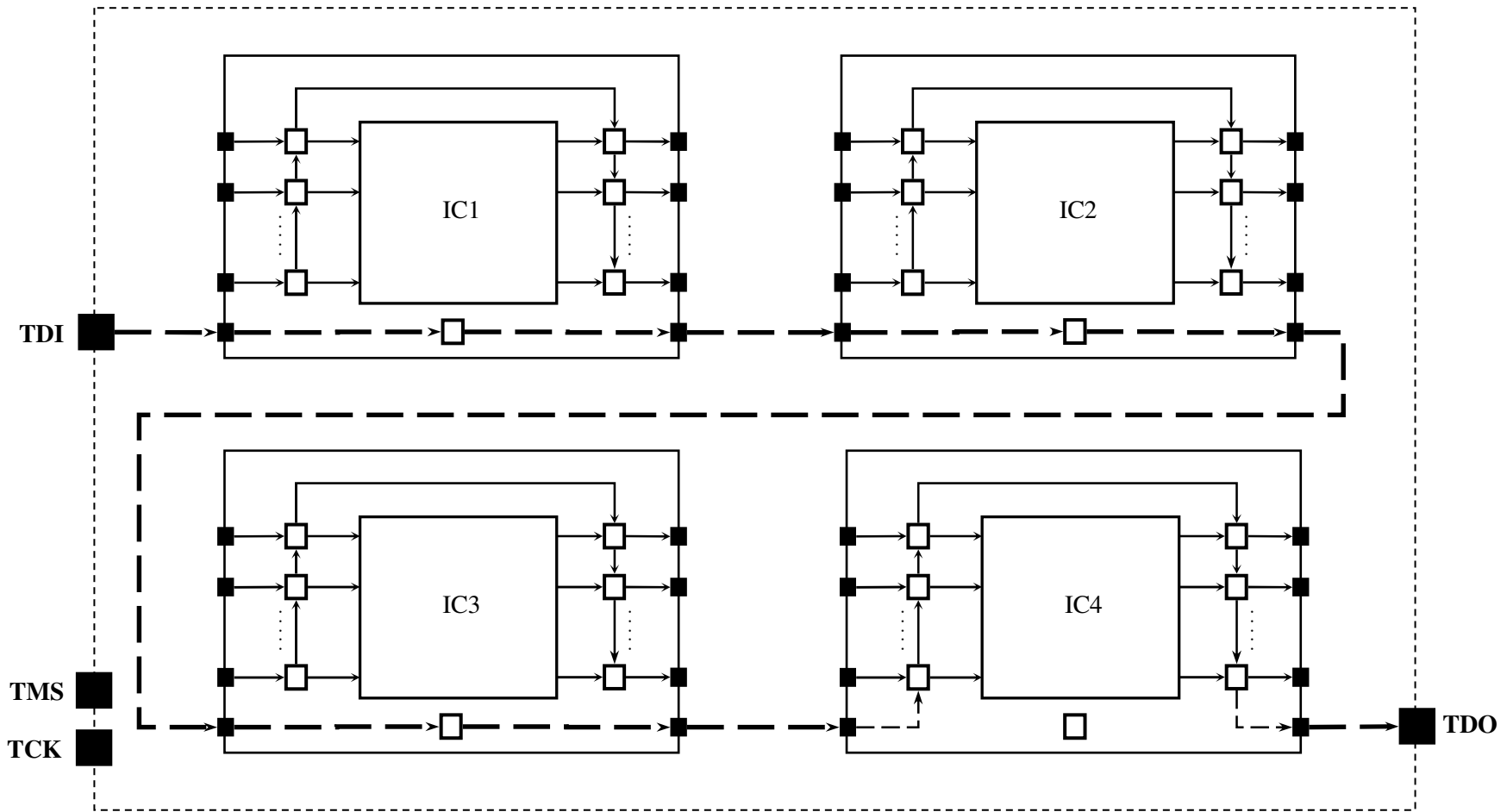
☞ permet d'appliquer les nouvelles données entrées pendant l'étape de shift vers le cœur du circuit

☞ Ce mécanisme permet de tester séparément les circuits logiques d'une carte

# Mode INTEST (suite)

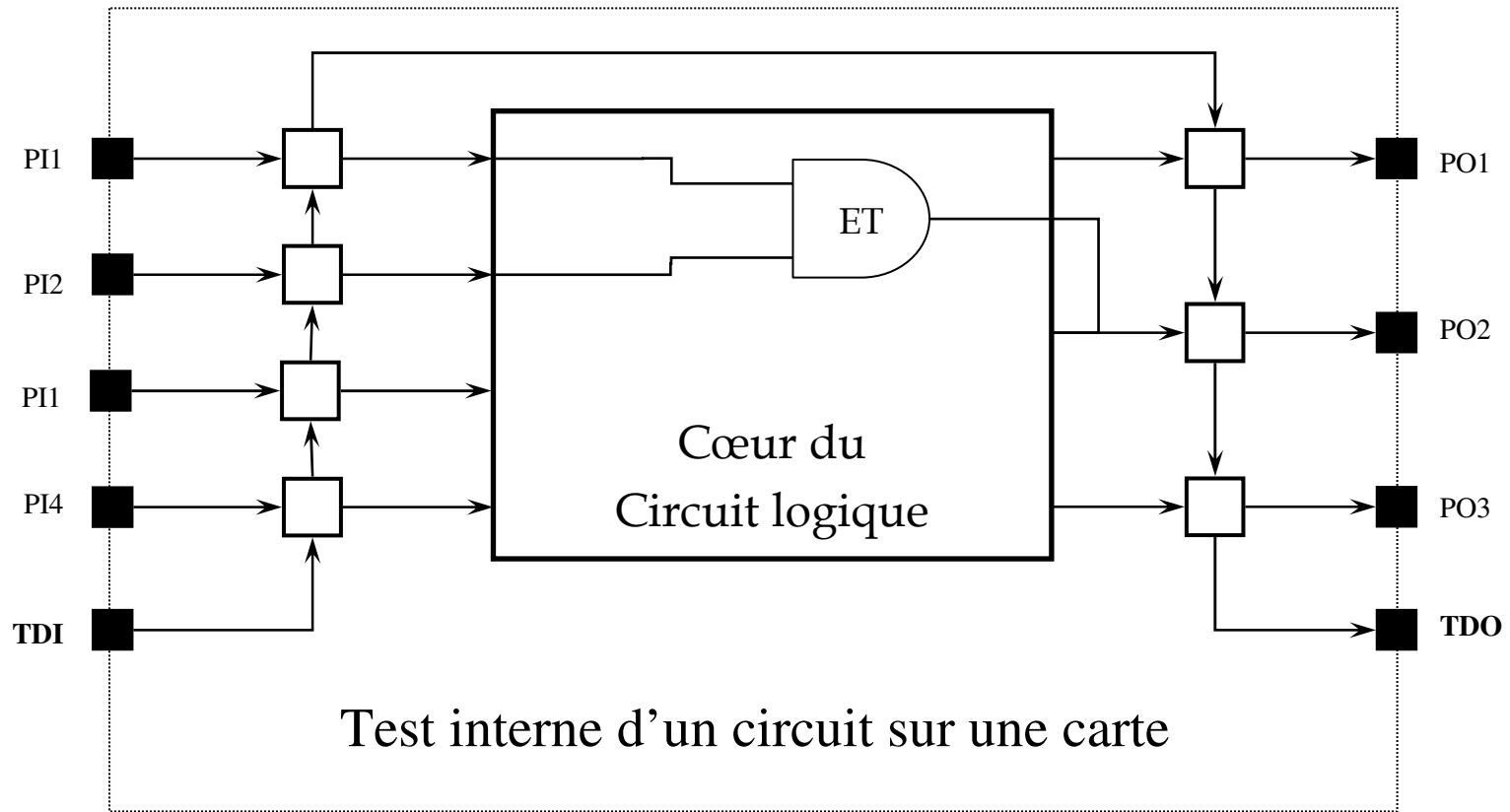


# Mode INTEST (suite)



Mode INTEST du circuit IC4

# Mode INTEST (suite)



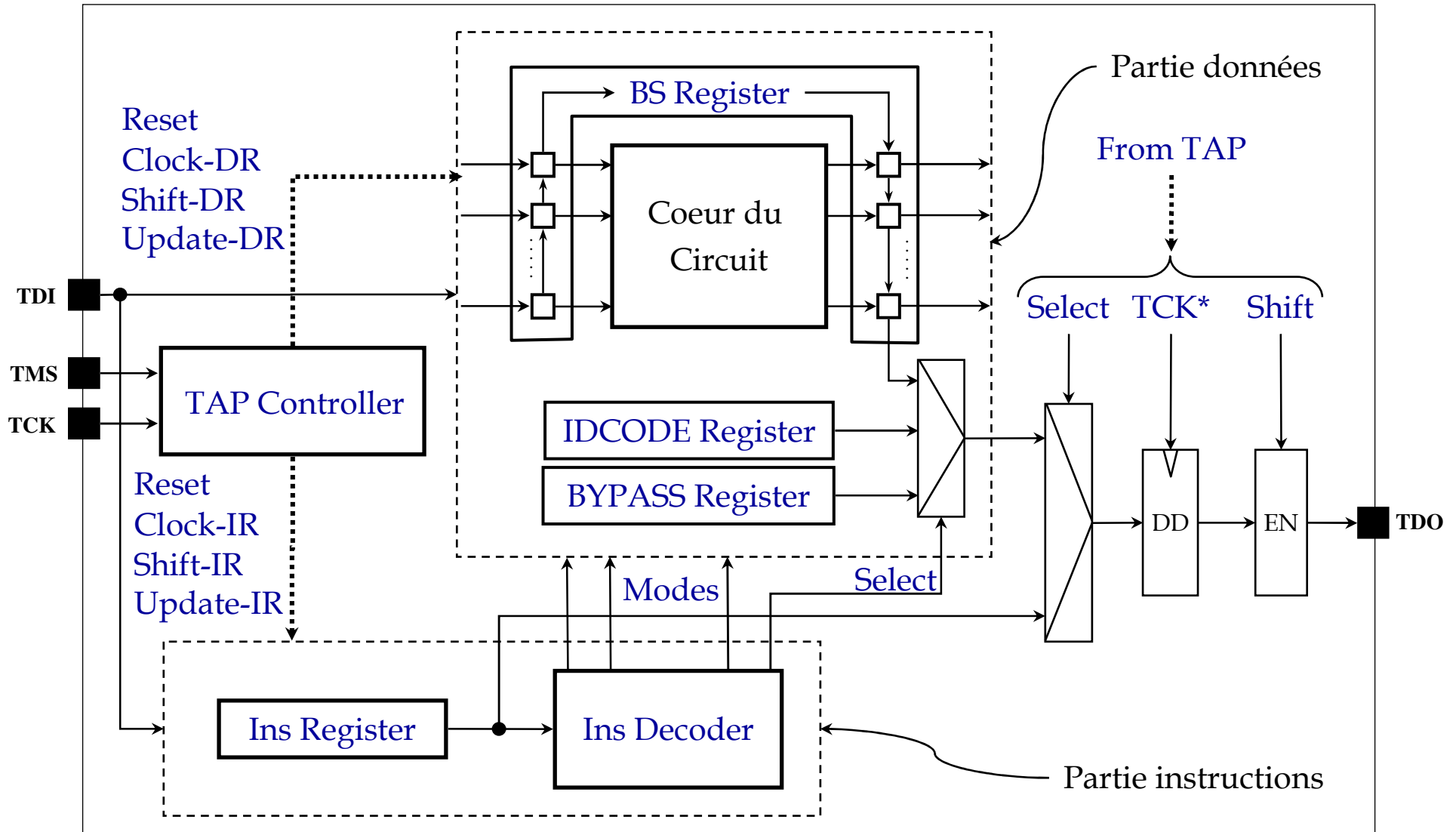
Input1(TDI) : xx11 xxx  
Output1(TDO) : xxxx x1x

Input2(TDI) : xx00 xxx  
Output2(TDO) : xxxx x0x

# MODE RUNBIST

- Step 1 : Initialiser le TAP à Test-Logic-Reset
- Step 2 : Charger l'instruction SAMPLE/PRELOAD
- Step 3 : « Shift » le vecteur de sécurité pour protéger le circuit des conflits
- Step 4 : Charger l'instruction RUNBIST, ce qui va sélectionner le registre de sauvegarde de la signature
- Step 5 : Aller à l'étape Run-Test-Idle
- Step 6 : Effectuer le nombre de coups d'horloge nécessaires pour le BIST
- Step 7 : « Capture » le registre de signature
- Step 8 : « Shift » la signature pour vérification
- Step 9 : Revenir à l'état Test-Logic-Reset

# Architecture globale du Boundary-Scan





# Langage BSDL Std. IEEE 1149.1b

- ☞ Nécessité d'un langage commun pour tous les fabricants de circuits intégrés et de testeurs
- ☞ Initié par Hewlett Packard en 1990
- ☞ Basé sur le VHDL
  - Sous ensemble du VHDL IEEE std 1076-1993
- ☞ Le but du langage est de faciliter la communication entre les outils de CAO pour échanger les informations sur le test des circuits contenant le Boundary-Scan
  - Utilisé pour vérifier la validité de l'implémentation du BS
  - Utilisé pour la synthèse en vue du test

# Langage BSDL Std. IEEE 1149.1b (suite)

## ☞ Std. VHDL package STD\_1149\_1\_1994

- Entity, Generic, Port, ....Constant

- Attributes

  - ↳ TAP ports

  - ↳ TAP instructions

    - ❖ Taille des instructions

    - ❖ Codes des instructions

  - ↳ Cellules Boundary-Scan

  - ↳ Registre Boundary-Scan

    - ❖ Taille du registre

    - ❖ Correspondance entre les cellules BS et les plots d'entrée/sortie

# Test d'intégrité du Boundary-Scan

- ☞ Il faut vérifier la chaîne du Boundary-Scan avant de commencer le test ou le diagnostic du système
- ☞ Plusieurs défauts peuvent affecter le Boundary-Scan :
  - Composant défectueux
  - Une connexion des plots du TAP coupée
  - Cour-circuit sur les connexion de la chaîne TDI-TDO
- ☞ La norme 1149.1 facilite le test de l'intégrité du BS :
  - Utilisation du mode BYPASS
  - Utilisation de l'état Capture\_IR qui charge la valeur '01'
  - Utilisation du mode IDCODE

# Test des interconnexions

- ☞ Le test des interconnexions est effectué en mode EXTEST
  - C'est la fonction principale du Boundary-Scan
  
- ☞ Il permet de détecter deux types de défauts :
  - Les court-circuits entre les connexions
    - ❖ Défauts très destructeur et difficiles à diagnostiquer
  - Les circuits ouverts sur les connexions
    - ❖ Défauts moins dangereux et plus faciles à diagnostiquer
  
- ☞ Des études sur le test des interconnexions existent avant l'apparition du Boundary-Scan

# Test des court-circuits

## 👉 Algorithme de test des court-circuits :

- Pour chaque nœud de la carte on identifie tous les plots drivers (sorties) et les plots récepteurs (entrées)
  - ↪ Choisir un driver pour chaque nœud de la carte
  - ↪ Tester les connexions entre les drivers choisis en utilisant :
    - ❖ La méthode de test « Walking 1's and 0's »
  - ↪ Vérifier tous les récepteurs pour chaque vecteur de test
  
- Exemple : 4 drivers à tester A,B, C et D

Nœud	Vecteurs de test							
A	1	0	0	0	0	1	1	1
B	0	1	0	0	1	0	1	1
C	0	0	1	0	1	1	0	1
D	0	0	0	1	1	1	1	0

# Test des circuits ouverts

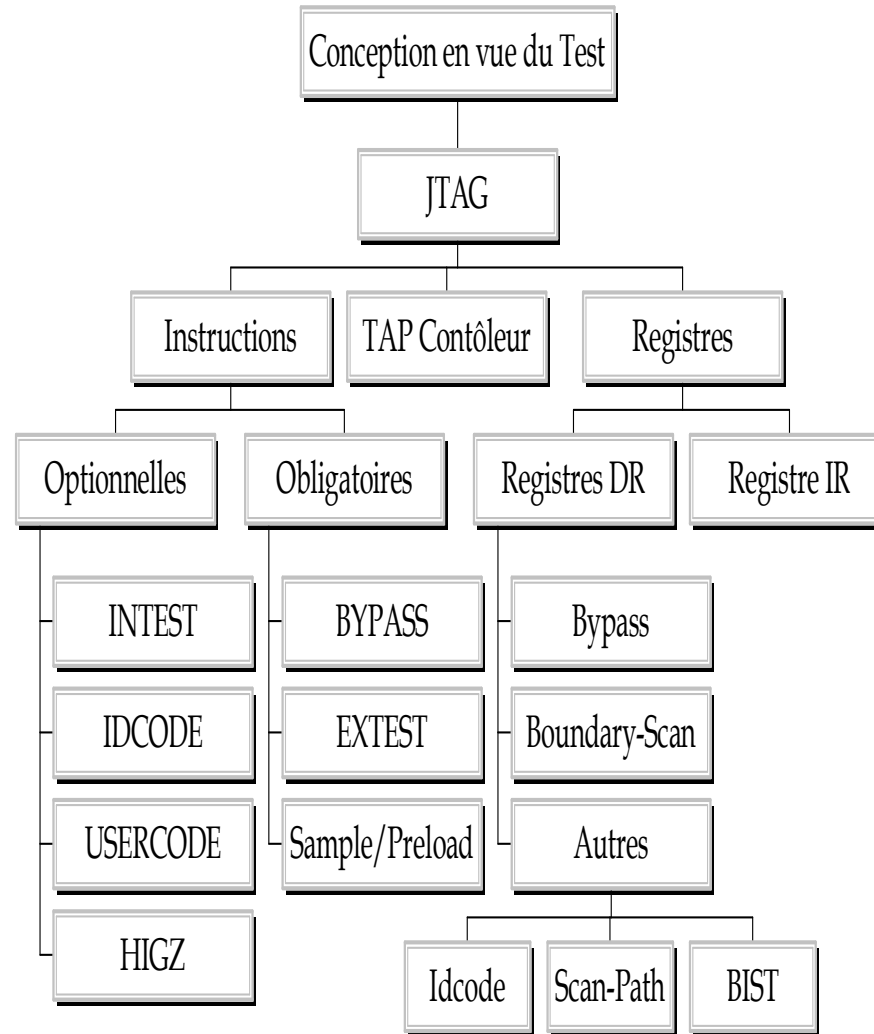
## 👉 Algorithme de test des circuits ouverts :

- Pour chaque nœud de la carte on identifie tous les plots drivers (sorties) et les plots récepteurs (entrées)
  - ↳ Choisir un driver du nœud à tester :
    - ❖ Appliquer les valeurs 0 et 1 en mettant tous les autres drivers à haute impédance (Z)
    - ❖ Pour chaque récepteur du nœud, vérifier la présence des valeurs 0 et 1
  - ↳ Répéter les deux étapes précédentes pour tous les drivers du nœud à tester
- Répéter les étapes précédentes pour tous les nœuds de la carte

# Conclusion

- ☞ Définition du JTAG
- ☞ Architecture du Boundary-Scan
- ☞ Les registres du Boundary-Scan
  - Le registre instruction
  - Les registres de données
- ☞ Modes de fonctionnement
- ☞ Langage de description BSDL
- ☞ Test des circuits JTAG
  - Test d'intégrité
  - Test des interconnexions

# Conclusion (suite)





# Pour bien comprendre le JTAG

Scan Educator from Texas Instruments (TI) :

☞ boundary-scan simulation

☞ PC-based tutorial with interactive

☞ FREE download at :

➤ <http://www.ti.com/sc/data/jtag/scanedu.exe>

# Documentation

## 👉 Lecture recommandée

- [1] [Design for Test ; The Electronic Packaging Handbook ; Ed. Blackwell, G.R. ; Boca Raton: CRC Press LLC, 2000](#)
- [2] [IEEE 1149.1 \(JTAG\) Boundary-Scan Testing in Altera Devices](#)

## 👉 Autres lectures

- [1] [IEEE Std 1149.1-2001 IEEE standard test access port and boundary-scan architecture](#)
- [2] [IEEE Std 1149.1b-1994 Supplement to IEEE Std 1149.1-1990, IEEE standard test access port and boundary-scan architecture](#)
- [3] [IEEE Std 1149.4-1999 IEEE standard for a mixed-signal test bus](#)
- [4] [IEEE Std 1149.6-2003. IEEE Standard for Boundary-Scan Testing of Advanced Digital Networks](#)
- [5] [JTAG Primer, Texas Instruments](#)

# Questions

