

# DFT e JTAG

*Design for Testability*

# Perché un circuito va testato?

Livello	Tipo di Malfunzionamento	Costo
Wafer	Diffusioni errate, corti circuiti	1\$
Chip	Taglio, Connessioni	10\$
Scheda	Saldature, Pin incurvati	100\$
Sistema	Cavi, Connessioni	1000\$
Utente Finale	Vibrazioni, Corrosioni	10000\$

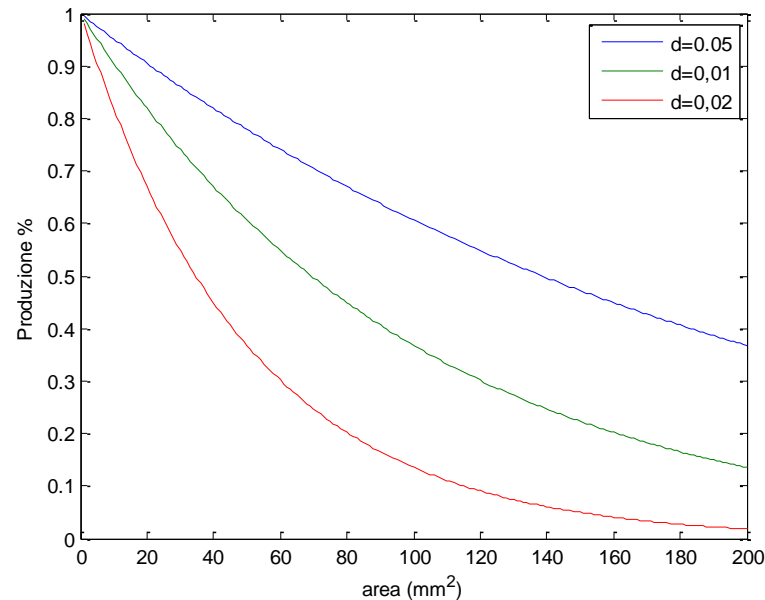
Qualunque malfunzionamento deve venir individuato al più presto

# Produzione

- Più aumenta l'area più aumenta lo scarto
- La percentuale di prodotti malfunzionanti è data da:

$$P = \exp(-AD)$$

- A: Area
- D: Densità di difetti (0.005 – 0.02 difetti/mm<sup>2</sup>)



Nel VLSI è fondamentale poter ridurre D

# Cosa testare:

## ■ Test estensivo:

- Circuiti combinatori a N ingressi :  
 $2^N$  vettori di test
- Circuiti seq. co N ingressi ed M Stati  
 $2^{(N+M)}$  vettori di test

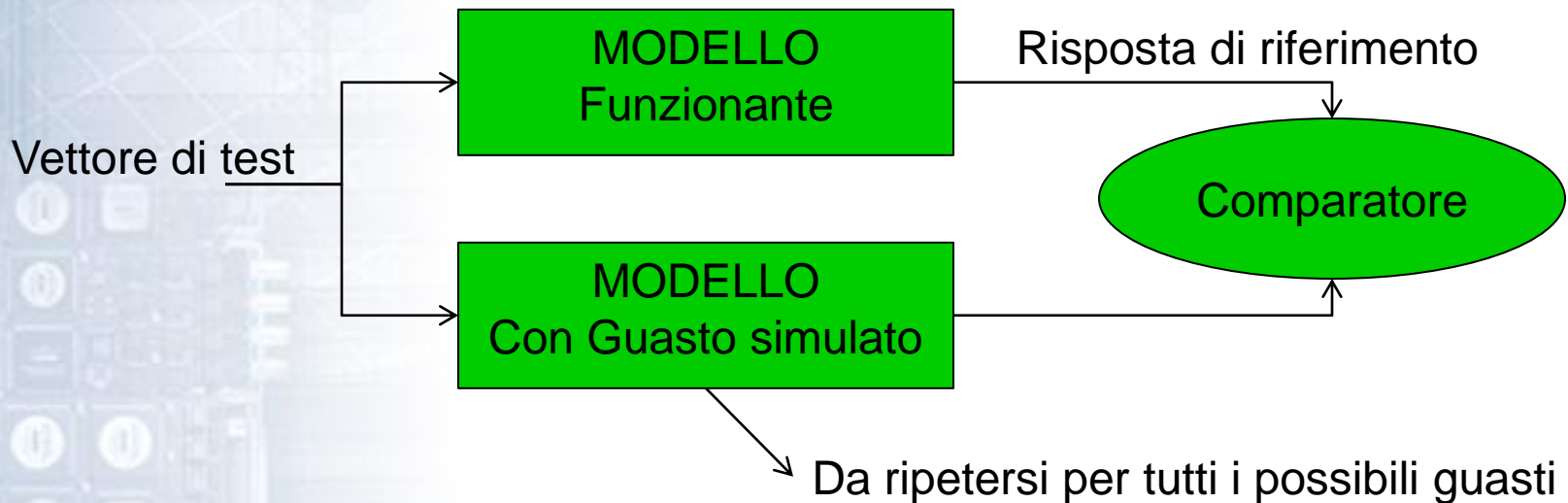
## ■ Esempio

- A 10 MHz per un circuito combinatorio con
  - 32 ingressi → tempo = 7 minuti
  - 64 ingressi → tempo = 60.000 anni

## ■ Bisogna sfruttare la conoscenza a priori del circuito e del suo funzionamento

# Come generare i vettori di test

- Basandosi sulla descrizione comportamentale
- Utilizzando un sottoinsieme dei vettori di test impiegati nelle simulazioni
- Generati da generatori automatici (ATPG)
- Pseudo random
- Attraverso simulazioni di malfunzionamenti



# Copertura degli errori

$$\text{Copertura} = \frac{\text{N.ro di malfunzionamenti evidenziati}}{\text{N.ro di possibili malfunzionamenti}}$$

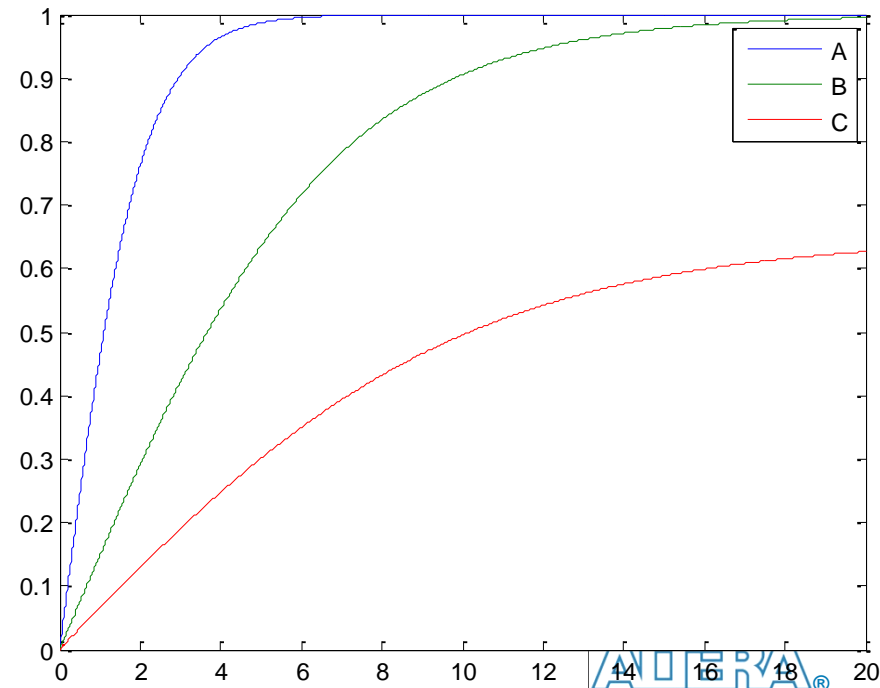
Per ottenere una buona copertura

- Sviluppare un circuito facile da testare
- Generare vettori di test significativi del funzionamento del circuito

A: Disegno sviluppato considerando già in fase di progetto le problematiche di testabilità

B: Progetto sviluppato senza considerazioni legate alla testabilità ma con un certo sforzo si riesce ad avere una buona copertura

C: Progetto difficile da testare anche con un grande numero di vettori



# Concetti Base

## ■ Controllabilità:

- La facilità nel controllare lo stato del circuito

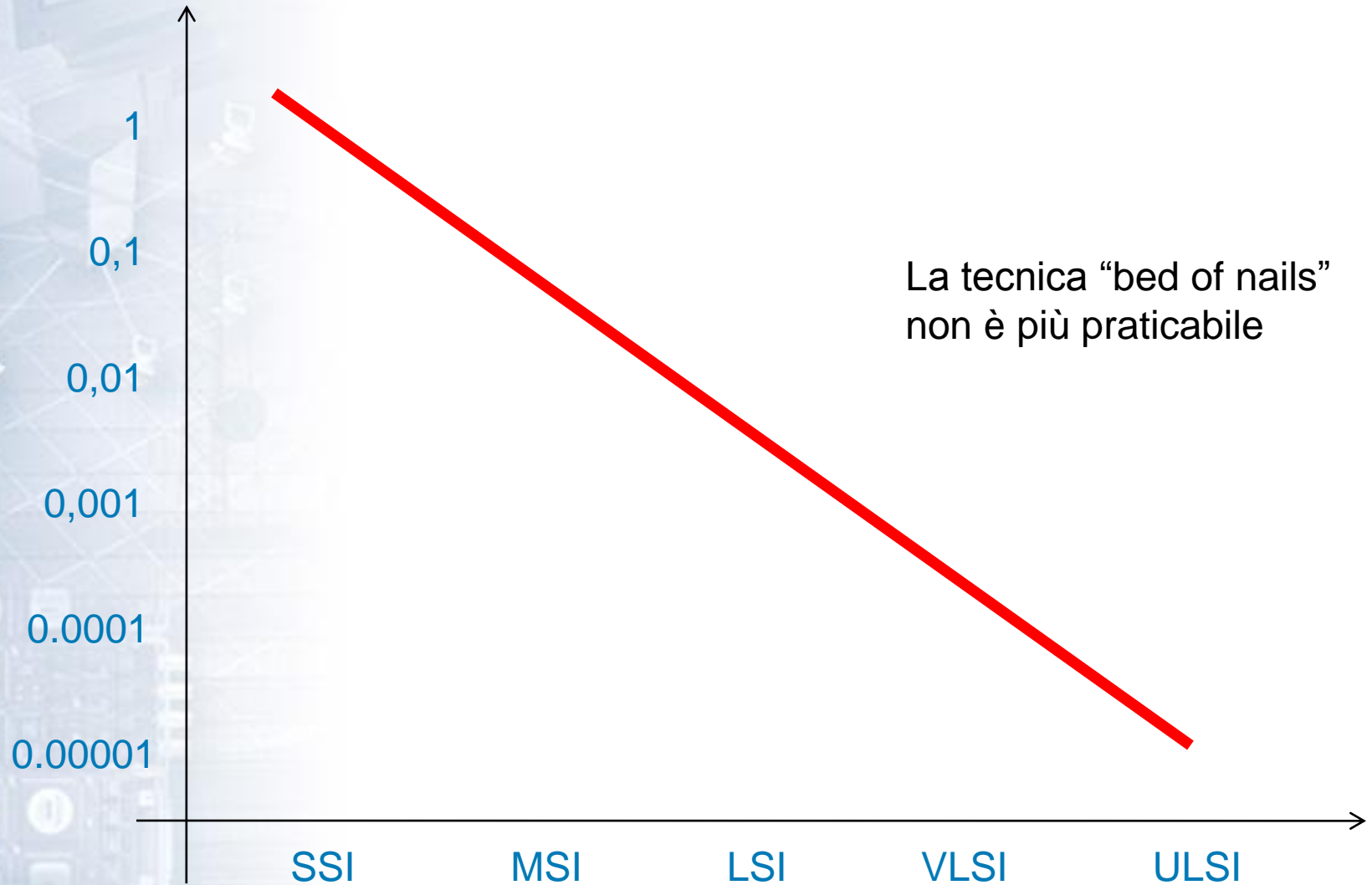
## ■ Osservabilità:

- La facilità nel comprendere in quale stato sia il circuito

Come testare un nodo di un circuito:

- Applicare una sequenza che porti il circuito nello stato voluto
- Applicare una sequenza che modifichi lo stesso
- Applicare una sequenza che consenta di “osservare” il nodo senza però modificare lo stato dello stesso

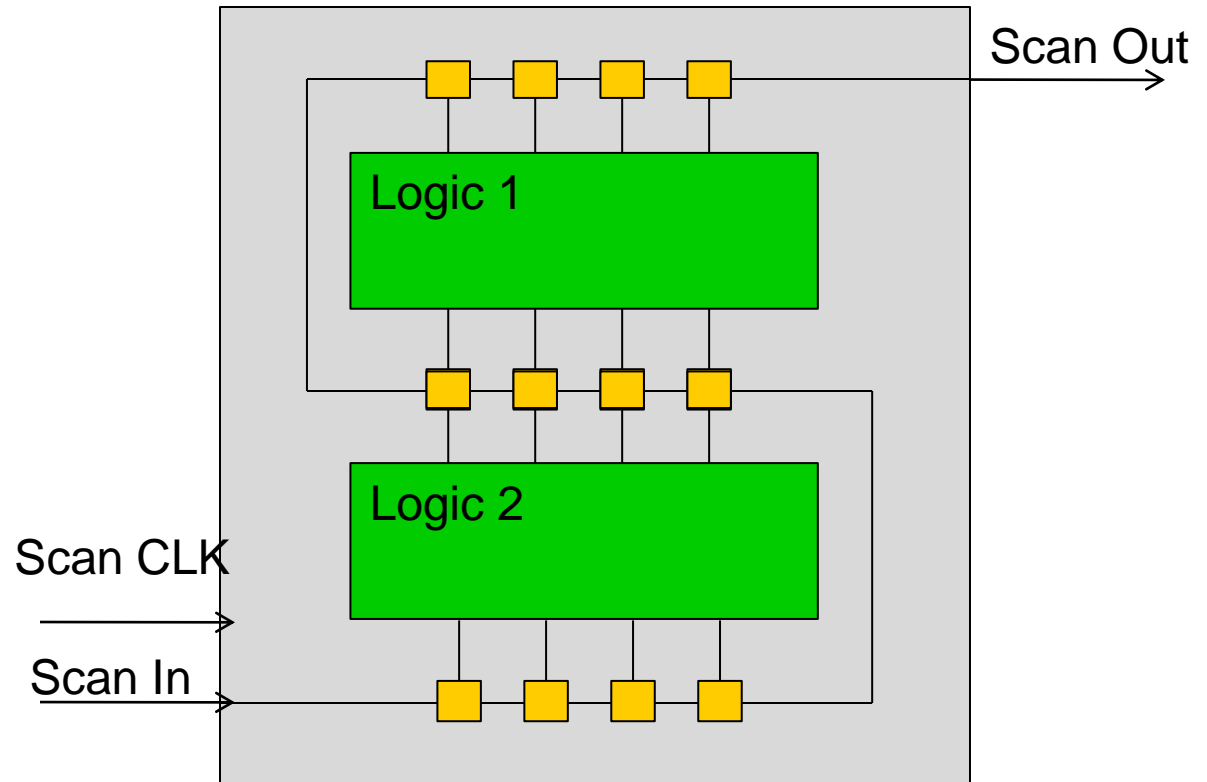
# Pins / Gate





# SCAN Path

- Si abiliti “Scan Mode” e si pre-carichino i dati
- Si disabiliti lo scan Mode e si faccia lavorare il circuito
- Si abiliti lo scan Mode e si analizzino i risultati

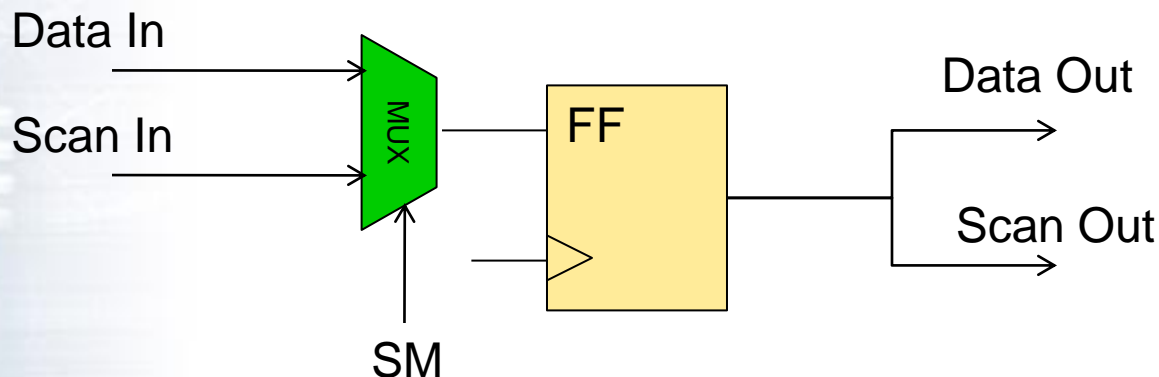


# Scan Path

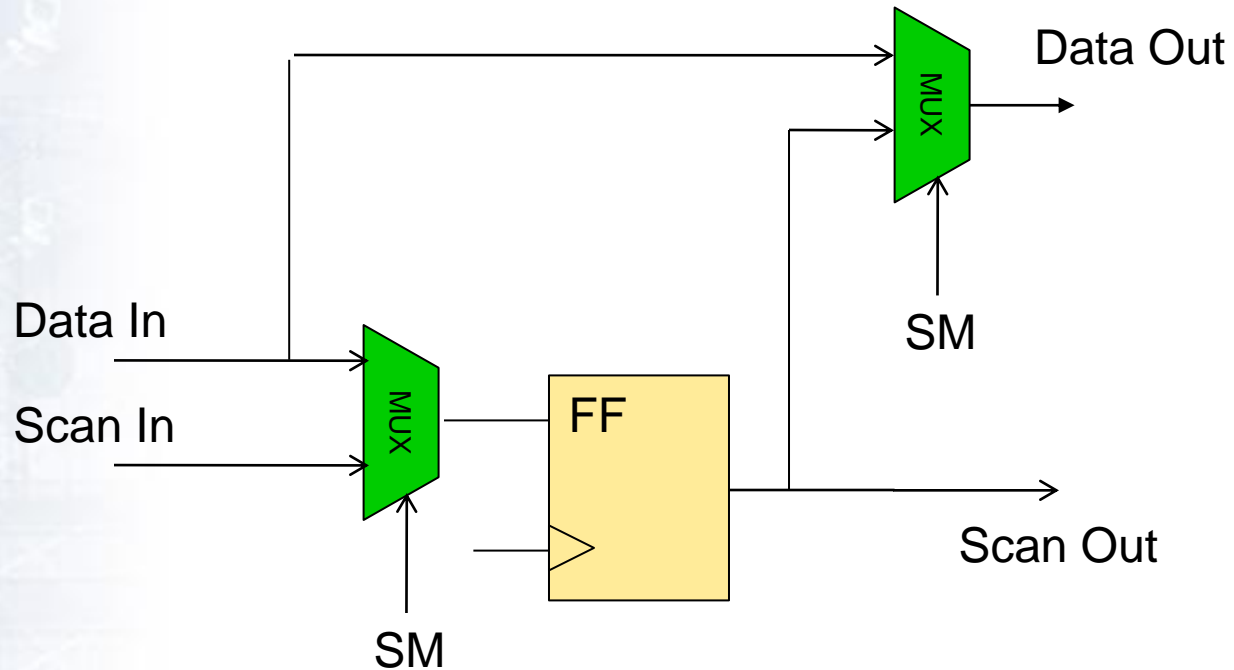
- Aumenta l'osservabilità e la controllabilità
- Si possono generare più facilmente vettori di test
- Aumenta l'Hardware del circuito (ogni nodo va sostituito con una cella SPC – Scan Path Cell)
- Diminuiscono (di poco) le prestazioni

## Nodi di Memoria (Con FF)

Si aggiunge un MUX ed alcune linee



# Scan Path



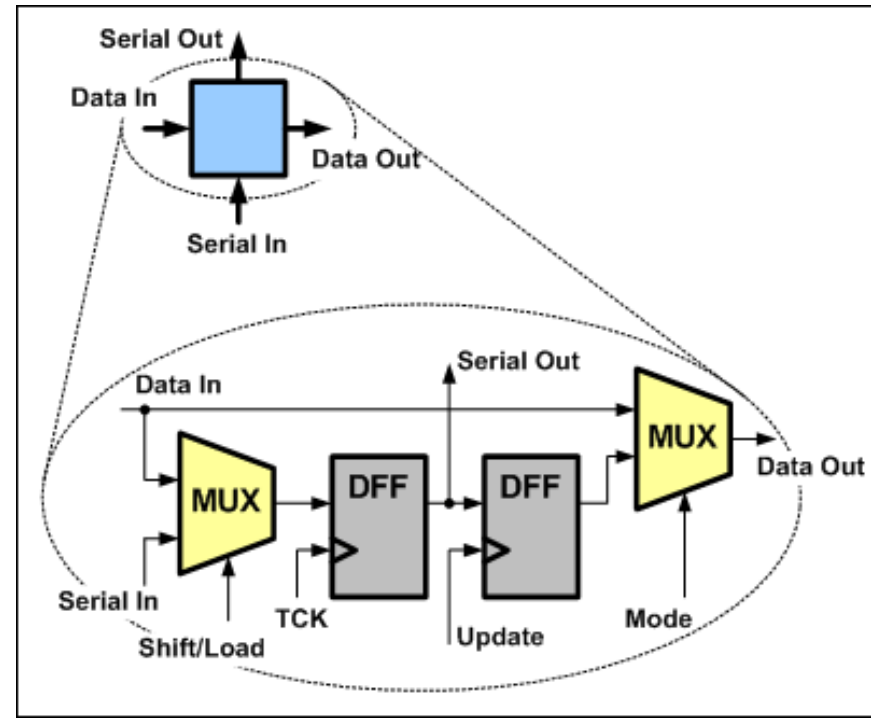
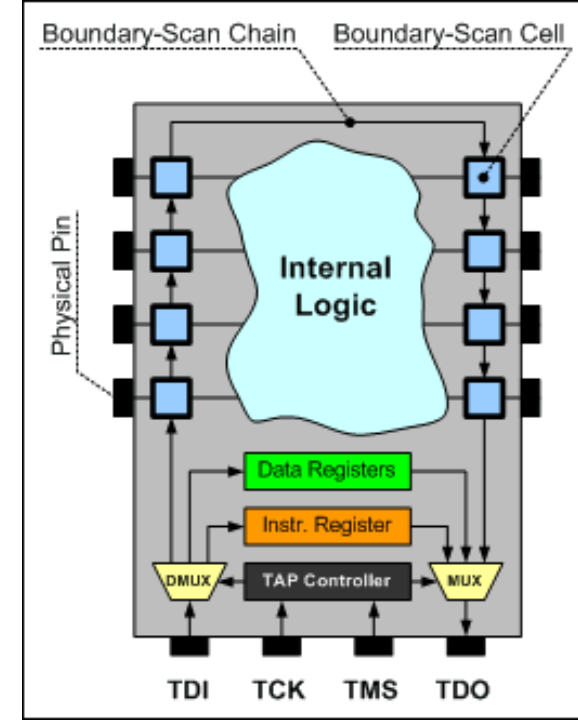
Nodi senza Memoria

Si aggiungono 2 Mux ed un FF

# JTAG



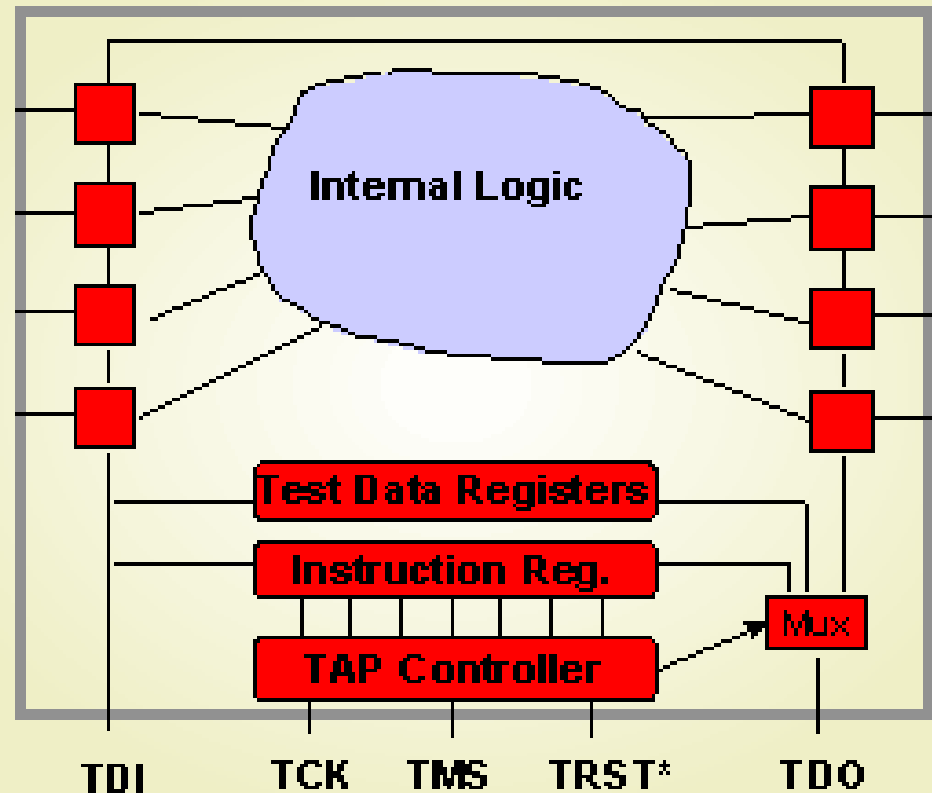
- Nasce lo Standard IEEE 1149.1
  - Testabilità interna (Scan Path)
  - Testabilità esterna (Boundary scan)
  - Testabilità della scheda all'interno del rack
- Definisce i segnali da impiegare
  - TDI,TDO,TCK,TMS,TRS
- Definisce il protocollo di comunicazione
- Definisce le funzioni
  - Funzioni Standard
  - Funzioni Opzionali



# Architettura JTAG

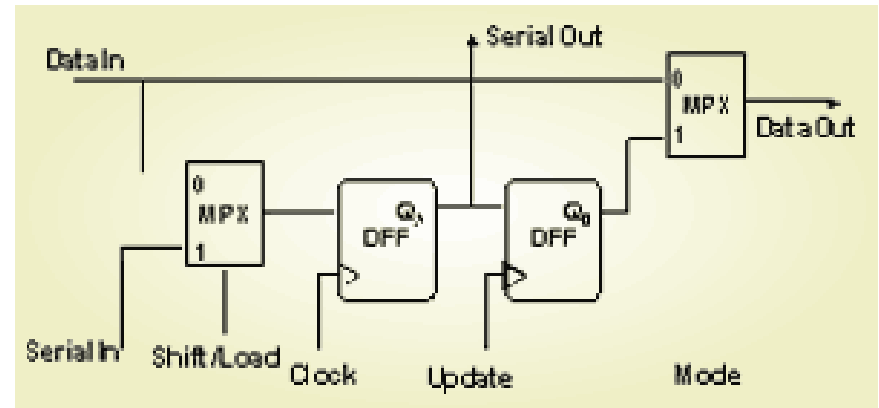
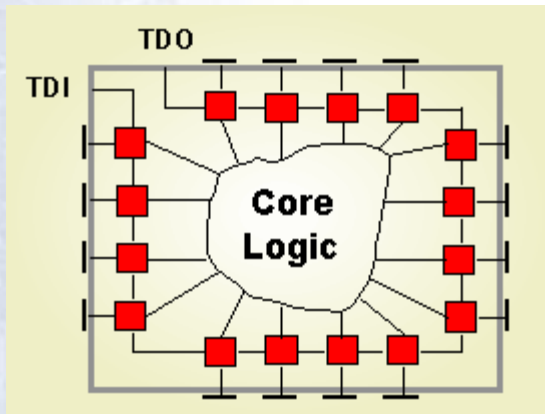
Deve contenere:

- Test Access Port (TAP)
  - Composto da 4/5 segnali TDI, TDO, TCK, TMS, (TRST)
- Instruction Register
- Boundary Scan Registers
- Altri eventuali registri
  - ID register
  - User Register
  - Bypass register
- TAP Controller



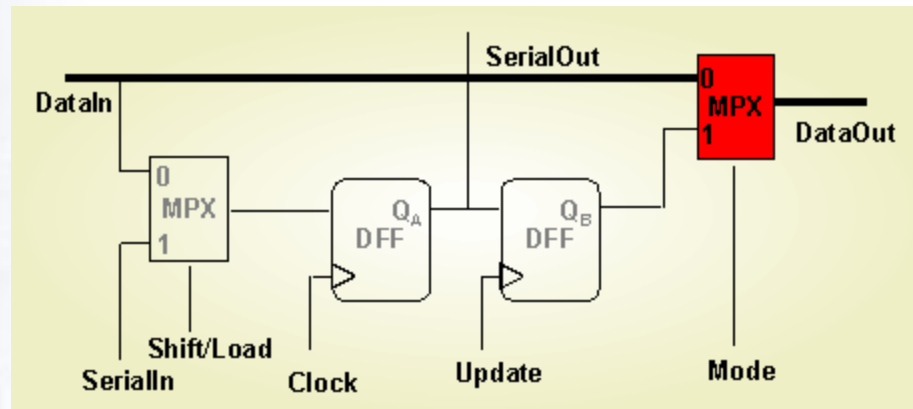
# Boundary Scan Cell

- I segnali di controllo arrivano dal TAP controller
- Vi sono 4 modi di funzionamento:
  1. normal mode
  2. capture mode
  3. scan mode
  4. update mode.

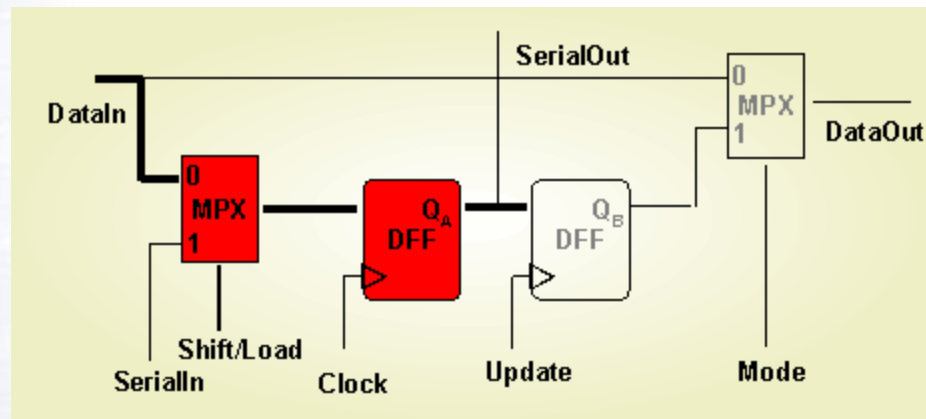


# BSC Modes

- Normal Mode (Mode=0)

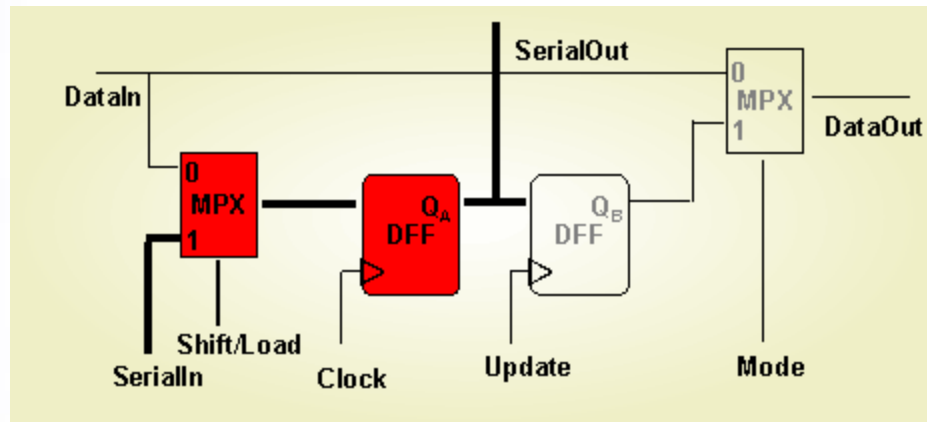


- Capture Mode (Shift/load=0, pulse on CLK)

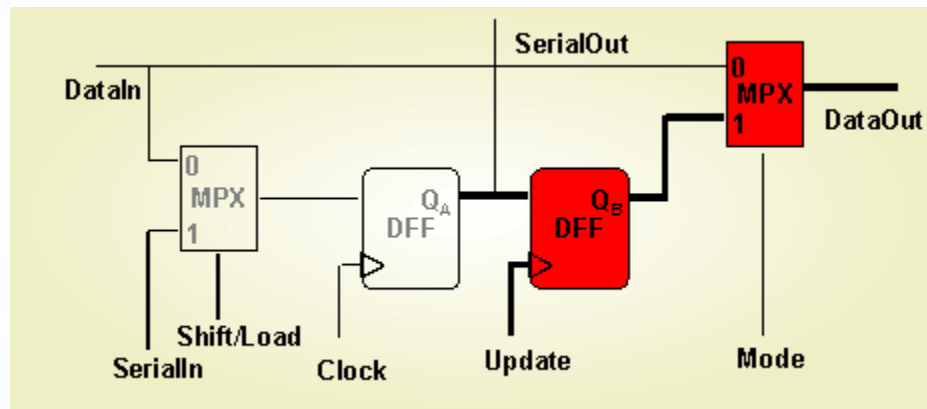


# BSC Modes

- Scan Mode (Shift/Load=1, sequence on clk)



- Update Mode (pulse on Update to acquire data, Mode=1 to drive to output)





# Other Registers

## ■ Bypass Reg.

- A single cell register
  - Se selezionato serve ad abbreviare il path
  - Viene selezionato tramite opportuna istruzione BYPASS (111..1)

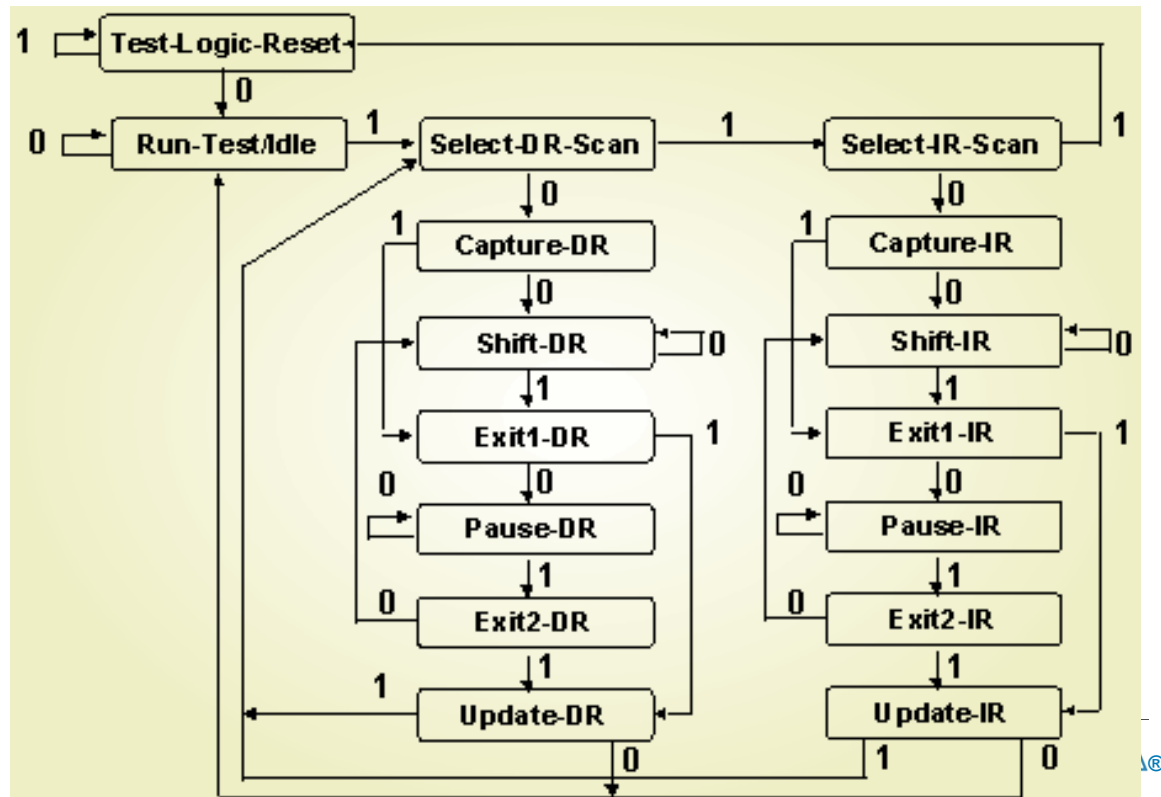
## ■ Device ID Reg

- A 32 bit register
  - Contiene informazioni sul dispositivo in uso
  - Viene selezionato tramite opportuna istruzione IDCODE

## ■ User Regs

# TAP Controller

- E una semplice macchina a stati finiti composta da 16 stati
- E' Pilotata dai segnali TCK e TMS ed opzionalmente da TRS
  - Una sequenza di cinque 1 la riporta comunque allo stato di reset
- Genera i segnali per pilotare i registri (composti da BSC)
- Gli stati sono raggruppabili secondo 4 gruppi:
  - Reset
  - BIST
  - Data Register
  - Instruction Register

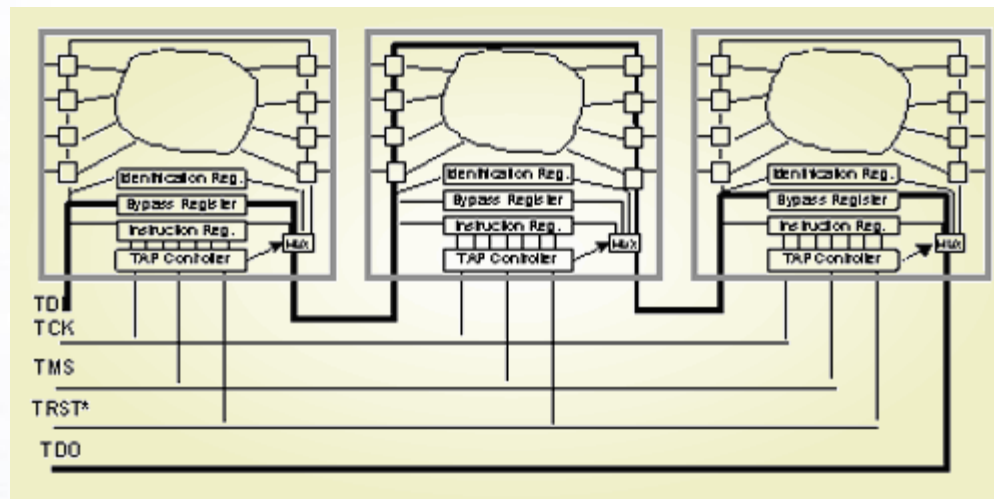


# Istruzioni

- Il funzionamento del JTAG dipende
  - Dall'istruzione introdotta in IR (tamite opportuna procedura da TAP)
  - Dal TAP Controller e dallo stato in cui si trova
- Ha effetto tanto su quale registro coinvolge che sull'operazione effettuata su di esso
- Le istruzioni si dividono in
  - Invasive e NON sul funzionamento normale ovvero se influenzano il normale funzionamento del circuito
  - Istruzioni Standard e NON

# Istr. BYPASS

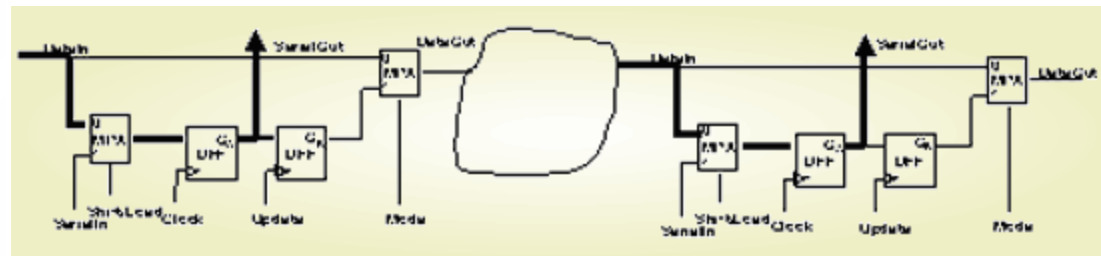
- Standard / NON Invasiva
- Fa riferimento al Reg. Bypass di 1 solo bit
- Serve ad “accorciare” il data path quando il circuito non è sotto test
  - CaptureDR state: A logical 0 is captured in the bypass register.
  - ShiftDR state: Input data shifted in via TDI is shifted out via TDO after a one-clock delay.
  - UpdateDR state: No changes.



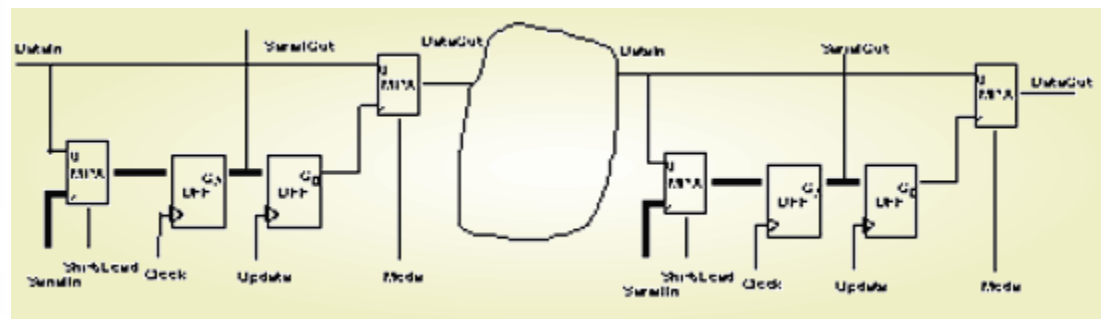
# Istr. SAMPLE/PRELOAD

- Standard / NON Invasiva
- Fa riferimento al Reg. Boundary Scan
  - CaptureDR state: The inputs from the system logic (test vector) are captured.
  - ShiftDR state: The captured test vector is shifted out via TDO output while a new test vector is shifted in via TDI input.
  - UpdateDR state: Preload Operation.

SAMPLE

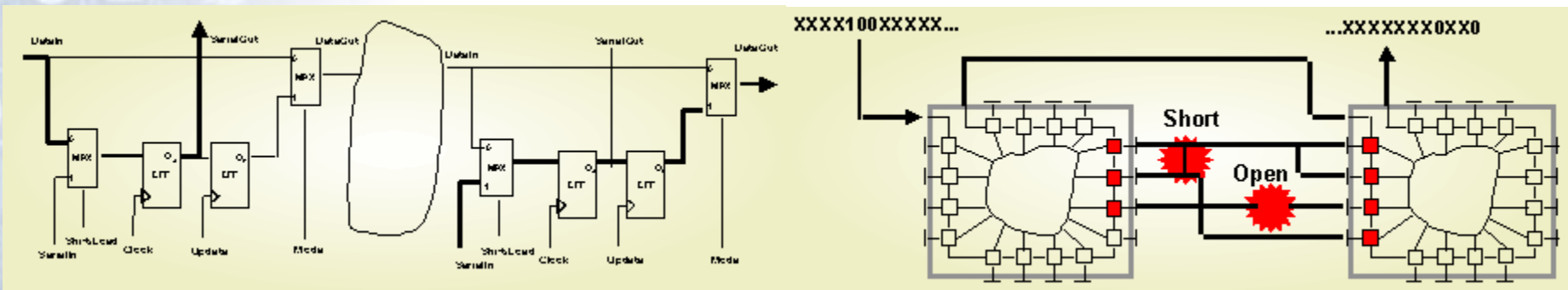


PreLOAD



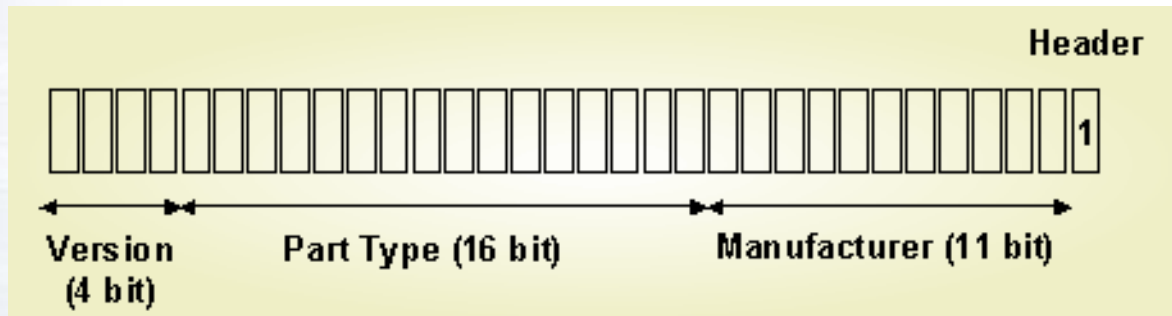
# Istr. EXTEST

- Standard / Invasiva
- Serve a testare collegamenti esterni
  - I dati presenti nel BS vengono inviati in uscita e si memorizzano quelli presenti in ingresso, che potranno essere successivamente resi disponibili
- Fa riferimento al Reg. Boundary Scan
  - UpdateDR state: The data shifted in via TDI is applied to Output and Control cells (output pins are driven with 0, 1, or highZ).
  - ShiftDR state: The captured test vector is shifted out via TDO output while a new test vector is shifted in via the TDI input.
  - CaptureDR state: The outputs from the system logic (test vector) are captured.



# Istr. IDCODE

- Opzionale / NON Invasiva
- Fa riferimento al Reg. IDCODE di 32 bits
- Serve a leggere il codice del circuito
  - CaptureDR state: The ID value is captured from the ID register.
  - ShiftDR state: The captured ID value is shifted out via TDO output.
  - UpdateDR state: The data shifted in via TDI is ignored (ID is a read-only register).





# Altre Istruzioni (Opzionali)

## ■ INTEST: Test Interno del Chip

- I dati precedentemente precaricati nelle BSC vengono forniti come ingressi del Chip e le uscite vengono memorizzate per poi essere rese disponibili all'uscita

## ■ RUNBIST: Autodiagnosi

- Solo per circuiti dotati di sistema di autodiagnosi. I dati resi disponibili in un apposito registro vengono successivamente resi disponibili all'uscita.

## ■ CLAMP :

- Dopo aver “precaricato” il BS, si “congelano” i valori d'uscita mentre il BS è in bypass

## ■ HIGH Z:

- Mette tutte le uscite del Chip in alta impedenza mentre BS è in BYPASS

## ■ USERCODE:

- Durante il normale funzionamento si mette a disposizione un registro utente a 32 bit