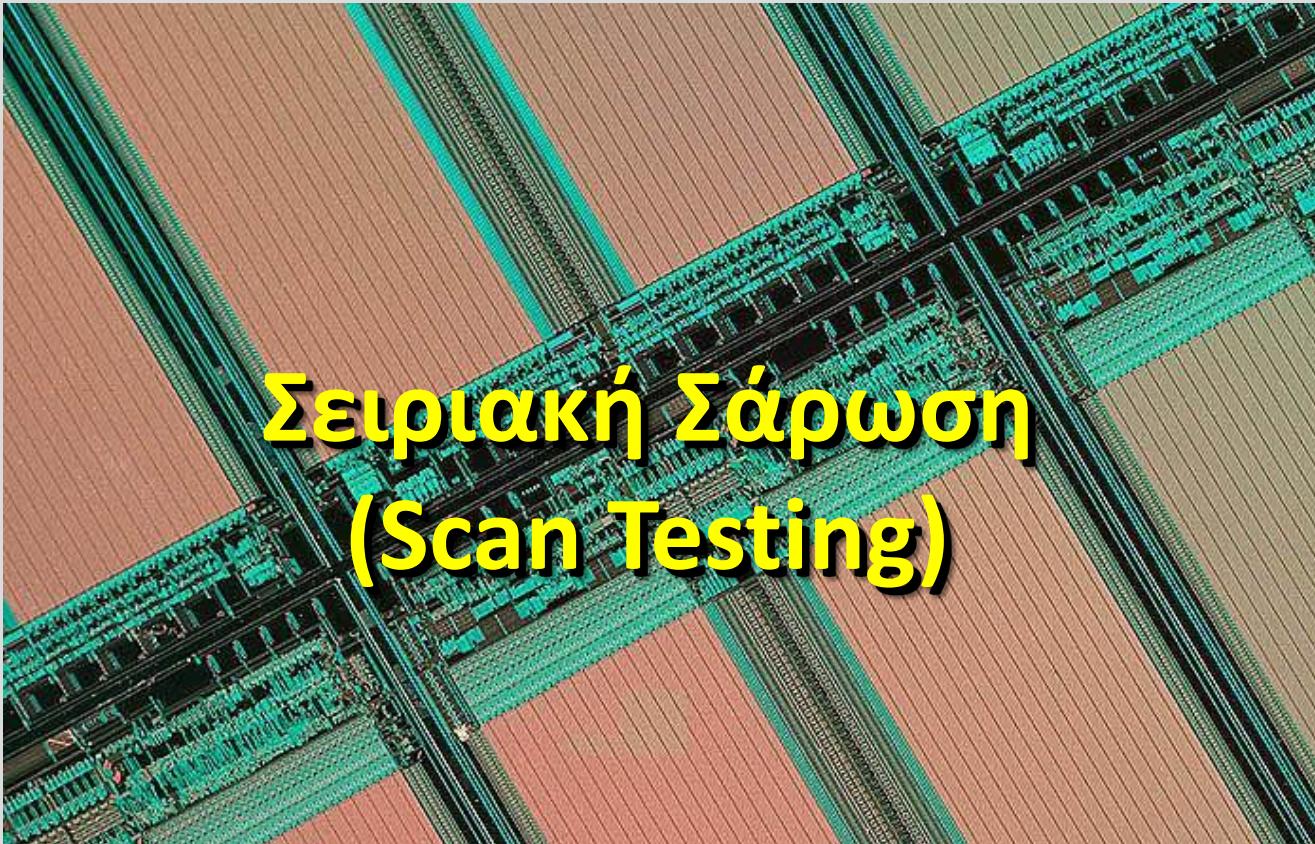


Δοκιμή και Αξιοπιστία Ηλεκτρονικών Συστημάτων

Πανεπιστήμιο Ιωαννίνων

Σειριακή Σάρωση (Scan Testing)



Τμήμα Μηχανικών Η/Υ και Πληροφορικής



CMOS Integrated Circuit Design Techniques

Overview



VLSI Systems
and Computer Architecture Lab

1. *Scan testing: design and application*
2. *At speed testing*
3. *The scan-set design technique*
4. *Scan testing power issues*
5. *The scan-hold design technique*
6. *Level sensitive scan design*
7. *Broadcast and Illinois scan design*

Σχεδιασμός για την αύξηση της ικανότητας δοκιμής

Ο σχεδιασμός για την αύξηση της ικανότητας δοκιμής (*design for testability - DFT*) αποτελεί την αδιαμφισβήτητη τεχνική επίτευξης του ελέγχου της ορθής λειτουργίας στα σύγχρονα ΟΚ.

- Χρήση **τεχνικών σειριακής σάρωσης (scan techniques)**.

Επιτρέπουν την πρόσβαση στις εσωτερικές καταστάσεις του ΟΚ.

- Τεχνικές **ενσωματωμένης αυτοδοκιμής (built-in self test - BIST)**.

Ενσωμάτωση στο ΟΚ τεχνικών και κυκλωμάτων που θα επιτρέψουν την αυτοδοκιμή του. Τα κυκλώματα αυτά παρέχουν διανύσματα εισόδου στο κυρίως κύκλωμα και/ή παρατηρούν τις εξόδους για την ανίχνευση λαθών. Ενδέχεται να παρέχουν κάλυψη σφαλμάτων και κατά τη λειτουργία του κυκλώματος (*on-line testing*).

- Τεχνικές **περιφερειακής σάρωσης (boundary scan)**.

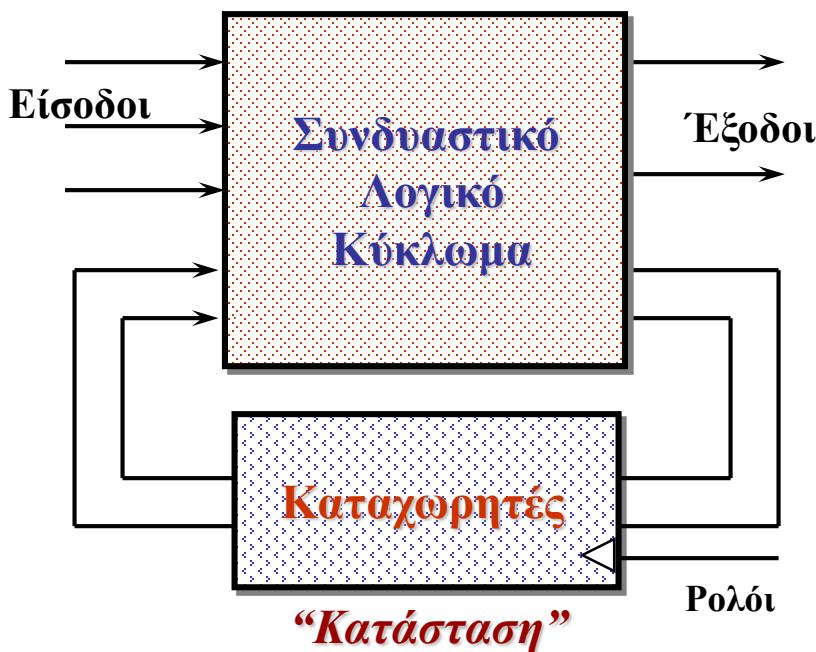
Πρόκειται για τεχνικές σε επίπεδο ΟΚ που επιτρέπουν με τη χρήση ενός μικρού αριθμού από εξειδικευμένους ακροδέκτες σημάτων να πραγματοποιηθεί γρήγορα και πρακτικά η δοκιμή του κυκλώματος (π.χ. πρωτόκολλά IEEE 1149.1 και IEEE 1500).

- Τεχνικές **παρατήρησης ρεύματος (current monitoring – I_{DDQ} testing)**.

Η κατανάλωση ισχύος αποτελεί κριτήριο ορθής ή μη λειτουργίας.

Δοκιμή Ακολουθιακών Κυκλωμάτων

έξοδοι = f (εισόδων, κατάστασης)

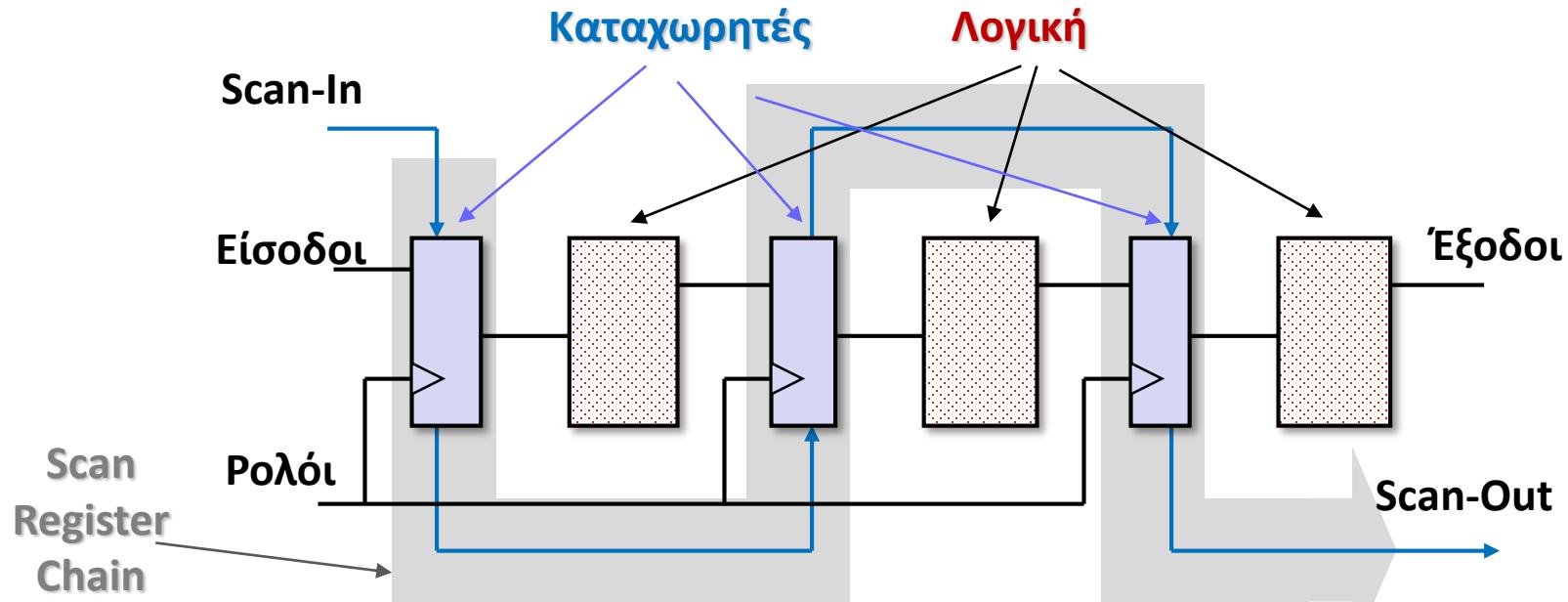


Στα ακολουθιακά κυκλώματα η εσωτερική αρχική κατάσταση του κυκλώματος (τιμές καταχωρητών) είναι συνήθως άγνωστη. Συνεπώς η ενεργοποίηση πιθανών σφαλμάτων και η διάδοση των λανθασμένων αποκρίσεων αποτελεί μια εξαιρετικά δύσκολη διαδικασία.

Η λύση είναι η χρήση τεχνικών αρχικοποίησης της εσωτερικής κατάστασης.

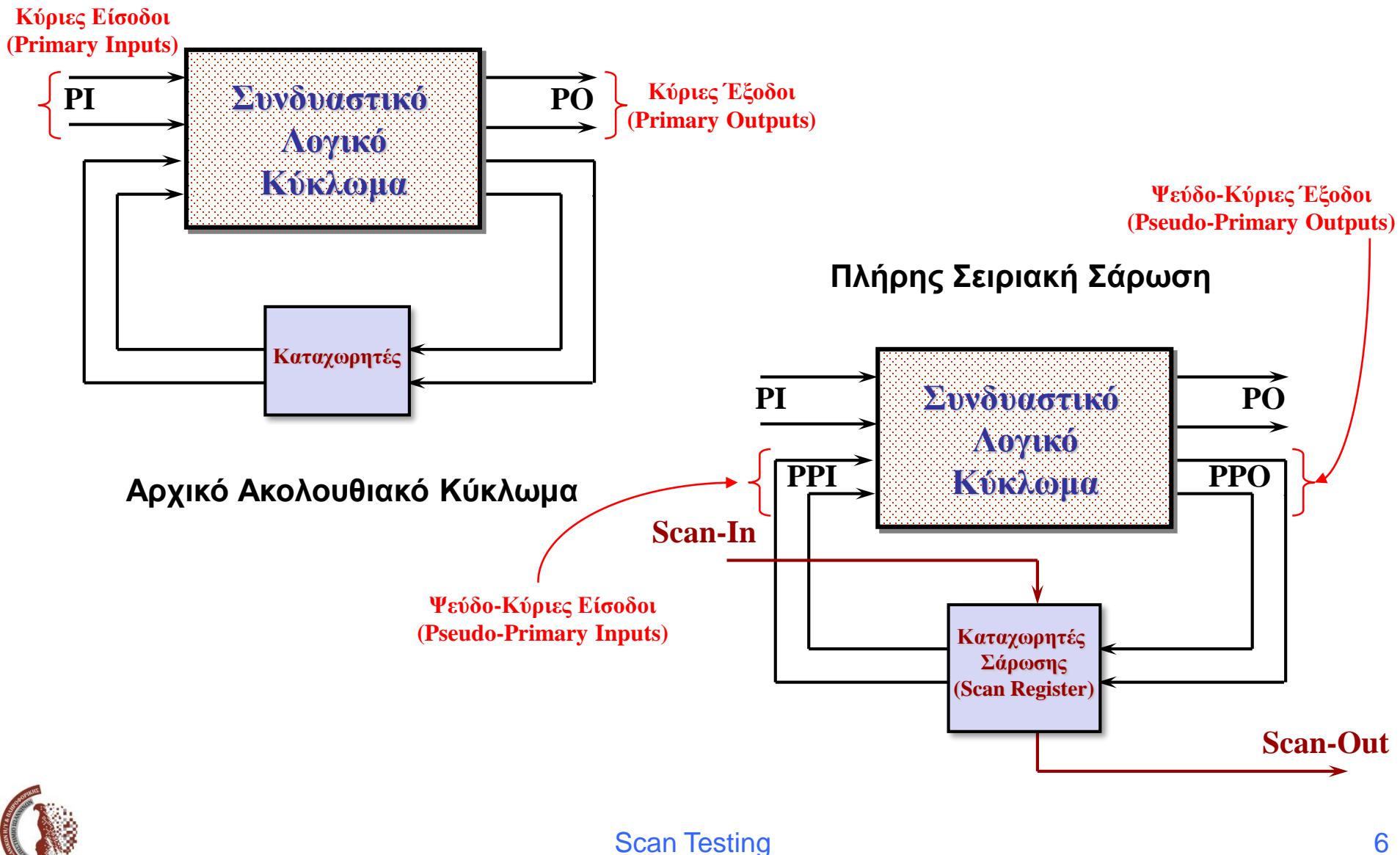
- Εφαρμογή κατάλληλων ακολουθιών διανυσμάτων δοκιμής ή/και χρήση σημάτων Set/Reset.
- Τεχνικές επιβολής αρχικών καταστάσεων από το εξωτερικό περιβάλλον και παρατήρησης των καταστάσεων που οδηγείται το κύκλωμα.

Γενικό Σχήμα Σειριακής Σάρωσης



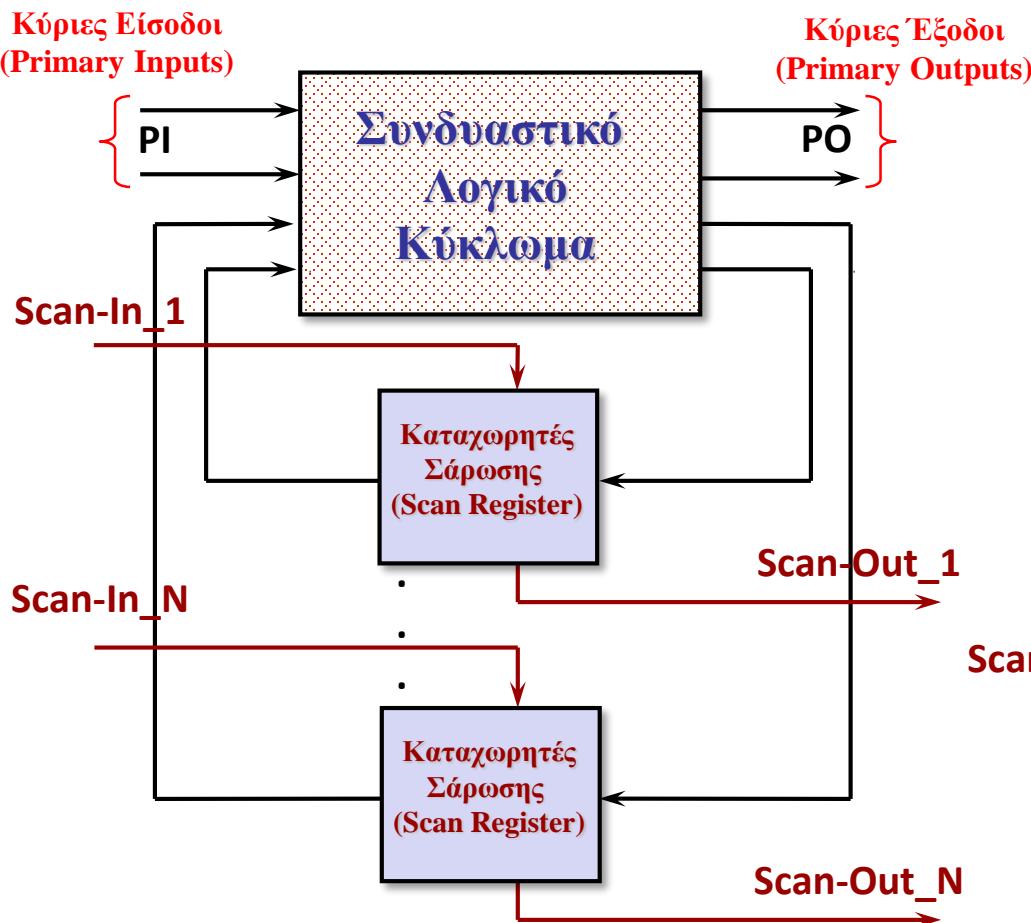
Ενοποίηση των καταχωρητών του κυκλώματος σε έναν ενιαίο **ολισθητή καταχωρητή** γνωστό ως **καταχωρητής (αλυσίδα) σάρωσης (scan register (chain))**. Δηλαδή τροποποιούμε τον καταχωρήτη με τέτοιο τρόπο ώστε εκτός από την κανονική του λειτουργία να μπορεί να λειτουργήσει και σαν **καταχωρητής ολίσθησης**. Κατά αυτόν τον τρόπο η εσωτερική κατάσταση του κυκλώματος μπορεί να καθορίζεται εισάγοντας σειριακά (**scan-in**) δεδομένα στον καταχωρητή σάρωσης. Επιπλέον οποιαδήποτε εσωτερική κατάσταση (που είναι η απόκριση της Λογικής) μπορεί να παρατηρηθεί από το εξωτερικό περιβάλλον εξάγοντας σειριακά (**scan-out**) τα δεδομένα του καταχωρητή σάρωσης.

Σχεδιασμός Σειριακής Σάρωσης (I)

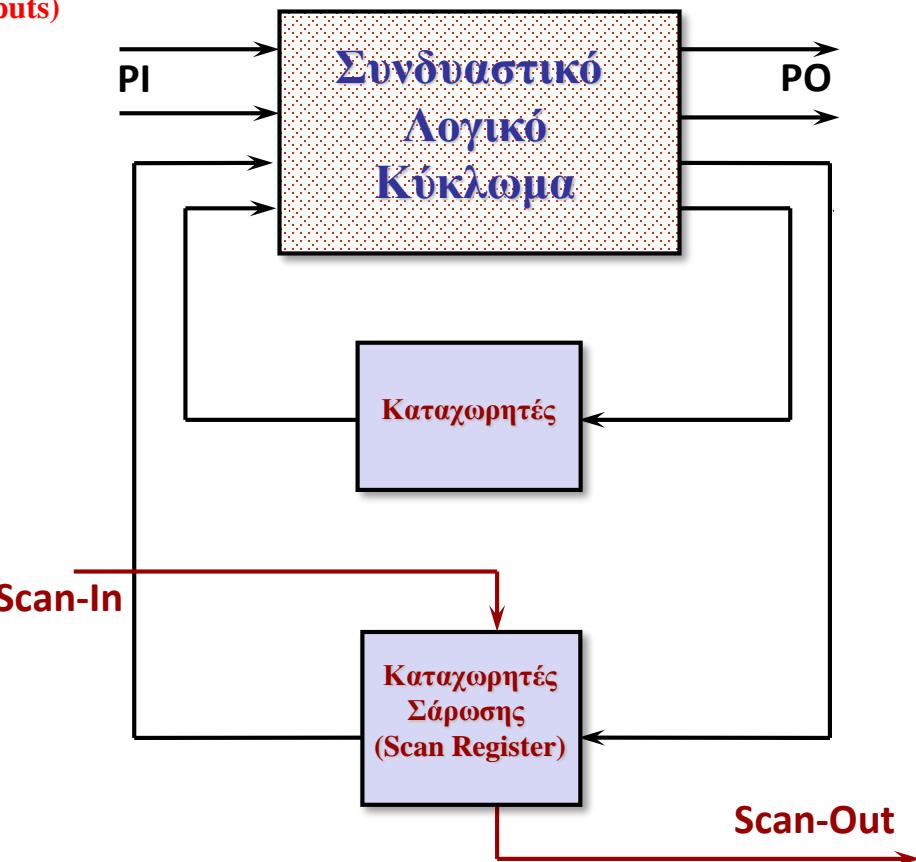


Σχεδιασμός Σειριακής Σάρωσης (II)

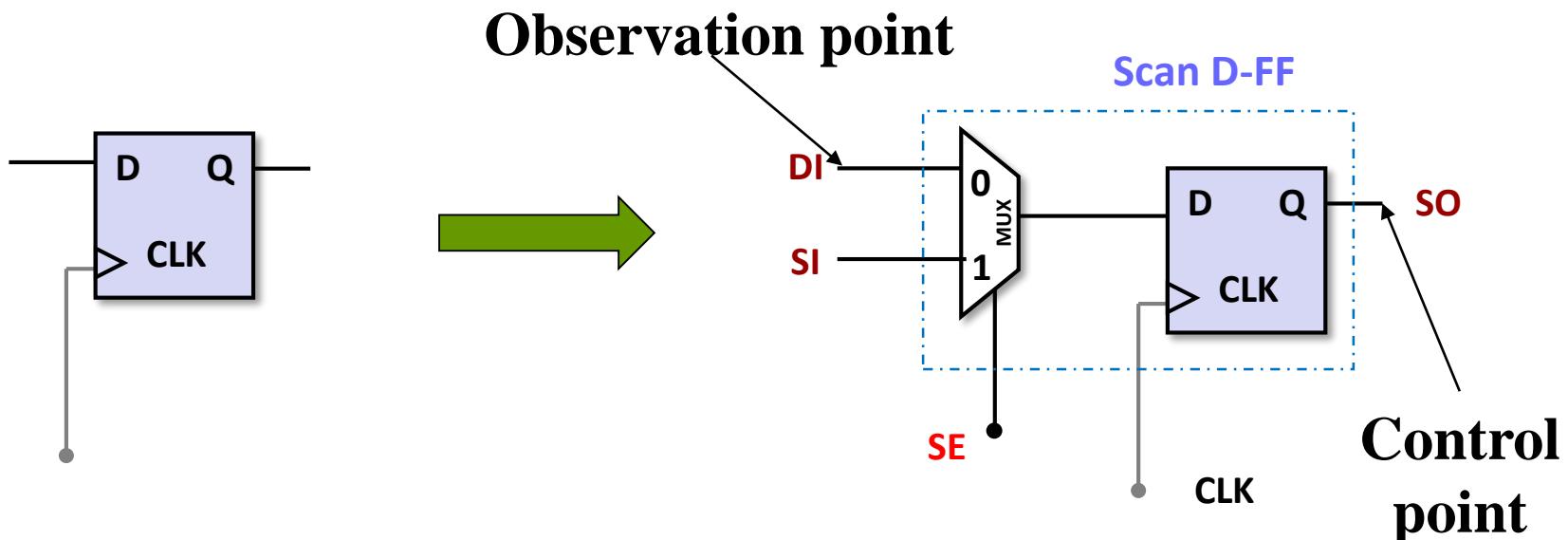
Τμηματική Σειριακή Σάρωση
(Πολλαπλές Αλυσίδες Σάρωσης)



Μερική Σειριακή Σάρωση



Scan D-FF (Muxed-D Scan Cell)



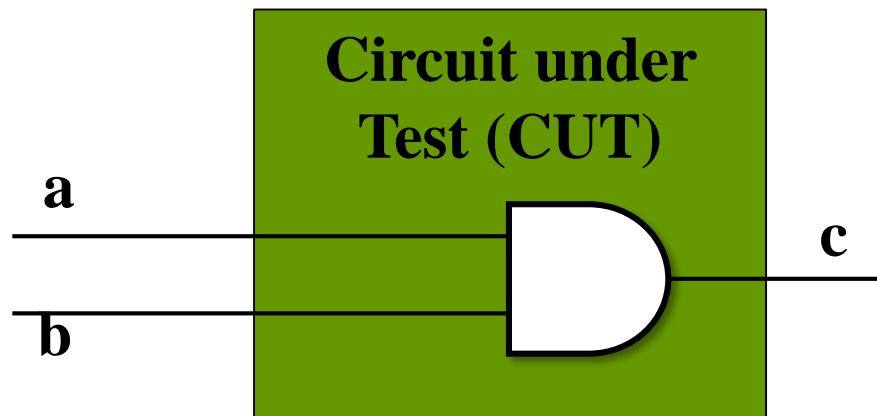
- Με προσθήκη ενός πολυπλέκτη μετατρέπουμε ένα D-Flip Flop σε **Scan D-Flip Flop (Muxed-D Scan Cell)**.
(*Muxed = Multiplexed (Πολυπλεγμένο)*)
- Γενικά τα (αποθηκευτικά) στοιχεία που έχουν δυνατότητα σάρωσης ονομάζονται **στοιχεία σάρωσης (scan cells)**

SE = Scan Enable
DI = Data In
SI = Scan In
SO = Scan Out

Παραδείγματα

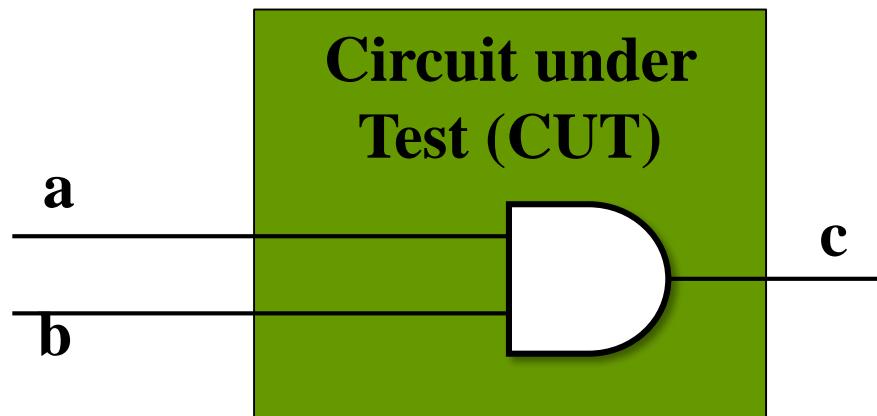


Παράδειγμα testset generation για stuck-at faults και σύνδεσης Scan Chain με ένα Circuit under Test (CUT) Βήμα-βήμα, σύνδεση εισόδων



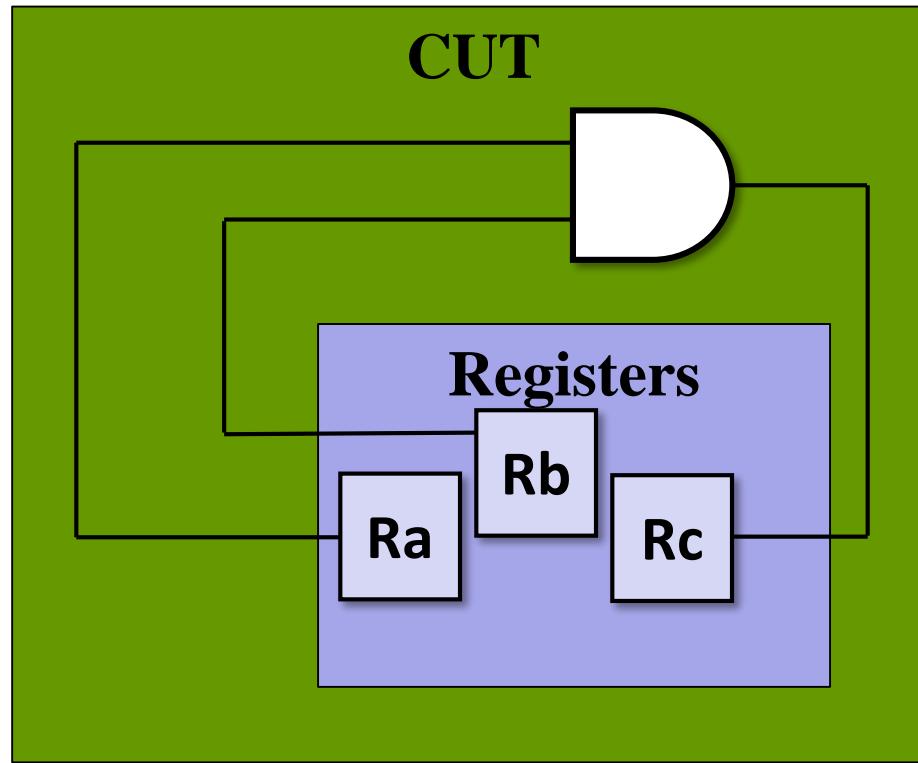
a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

Παράδειγμα testset generation για stuck-at faults και σύνδεσης Scan Chain με ένα Circuit under Test (CUT) Βήμα-βήμα, σύνδεση εισόδων

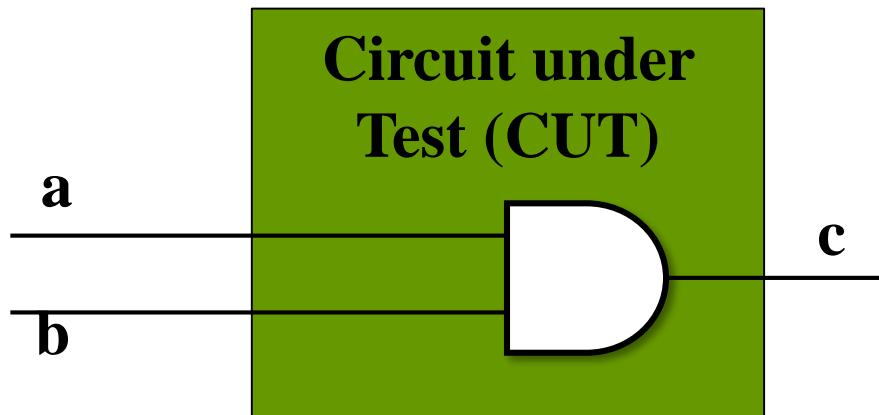


a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

Παράδειγμα: το αρχικό ακολουθιακό κύκλωμα



Παράδειγμα stuck-at faults testset generation και testset compaction



a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

testvector->expected response

a b c -> a b c

Sa0-a: 1 1 x -> 1 1 1

Sa1-a: 0 1 x -> 0 1 0

Sa0-b: 1 1 x -> 1 1 1

Sa1-b: 1 0 x -> 1 0 0

Sa1-c: 0 x x -> 0 x 0

Sa0-c: 1 1 x -> 1 1 1

Compact testset

Sa0-a, Sa0-b, Sa0-c

1 1 x -> 1 1 1

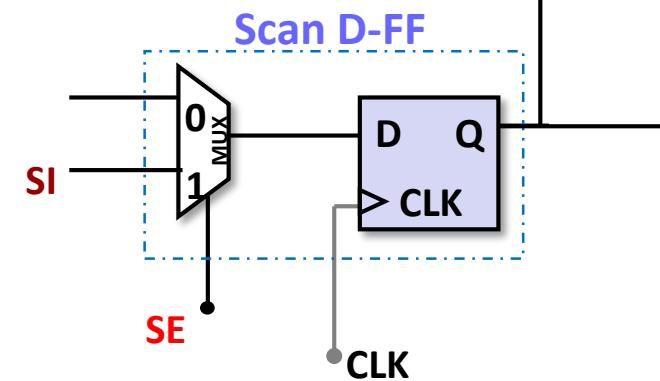
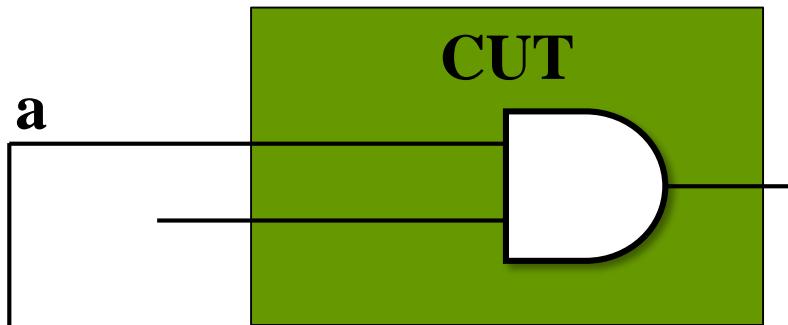
Sa1-a, Sa1-c

0 1 x -> 0 1 0

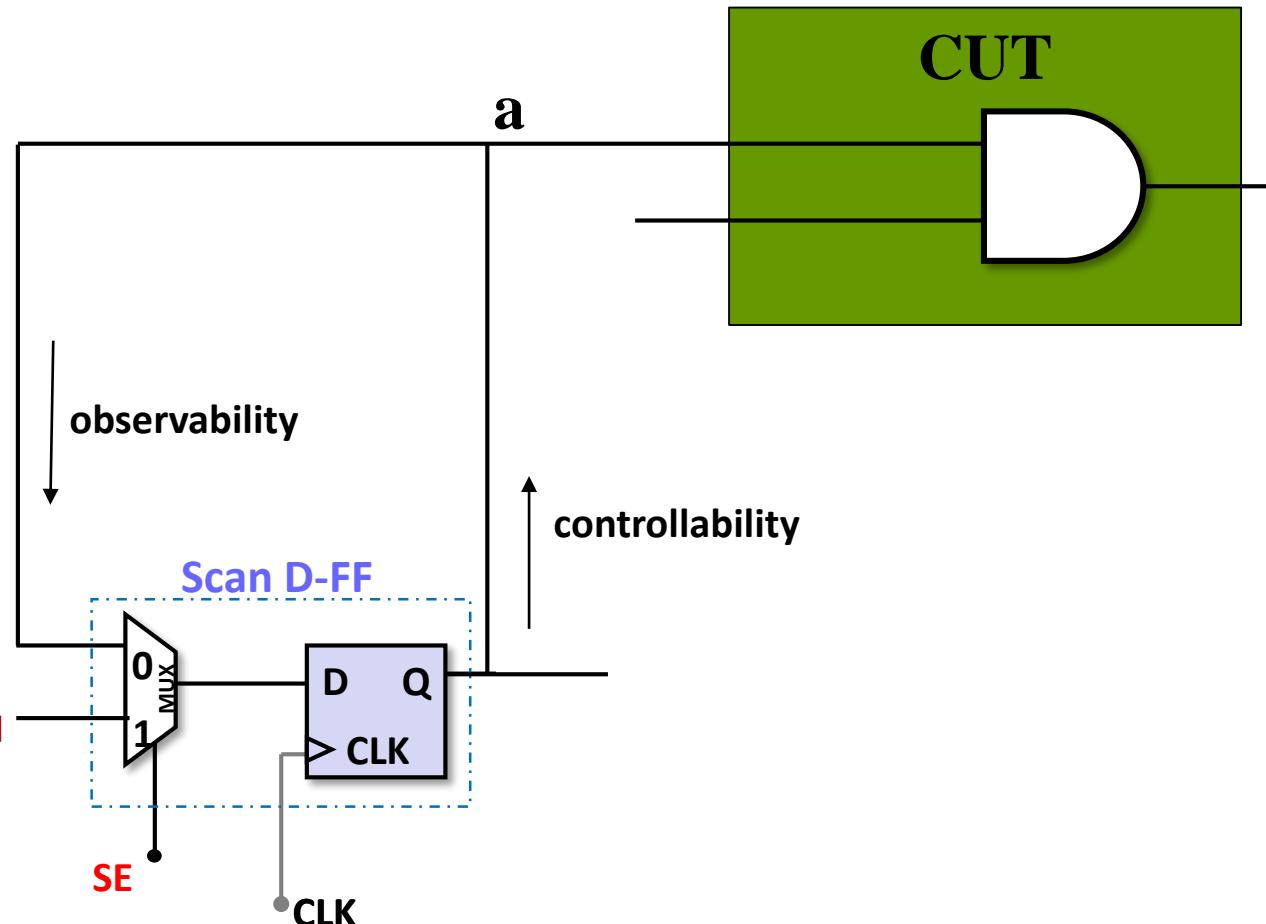
Sa1-b

1 0 x -> 1 0 0

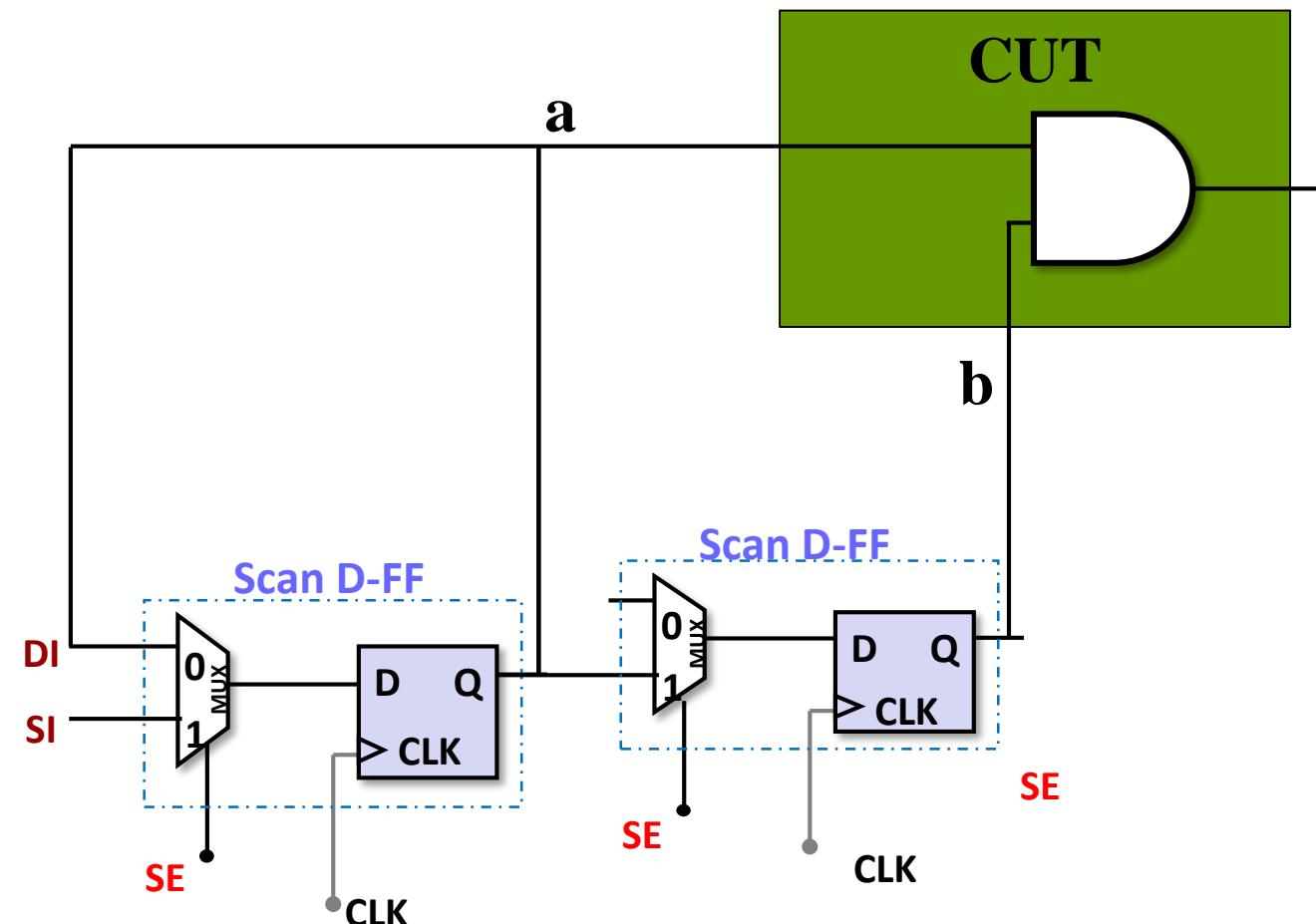
Παράδειγμα σύνδεσης Scan Chain με ένα Circuit under Test (CUT) Βήμα-βήμα, σύνδεση εισόδων



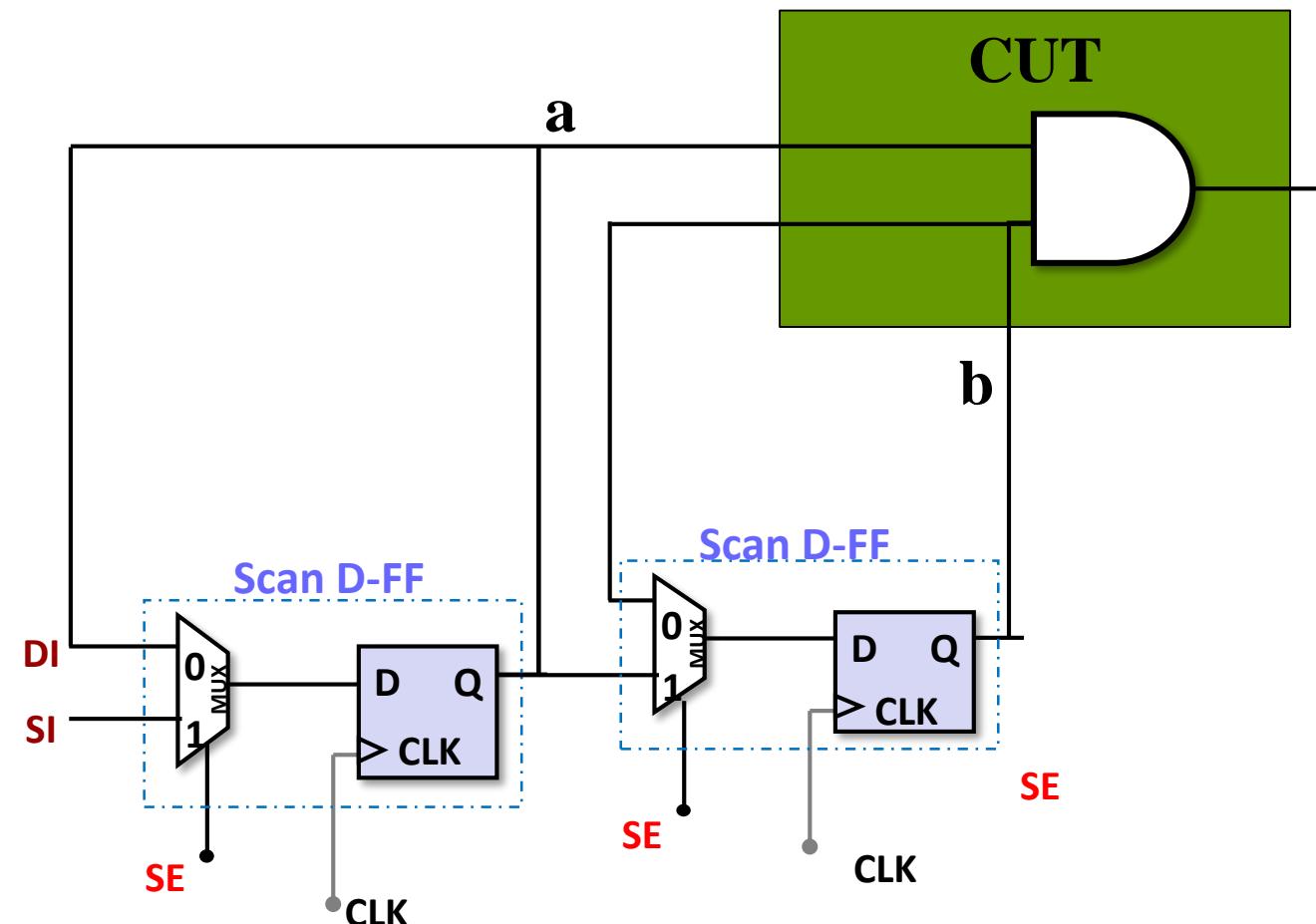
Παράδειγμα σύνδεσης Scan Chain με ένα Circuit under Test (CUT) Βήμα-βήμα, σύνδεση εισόδων



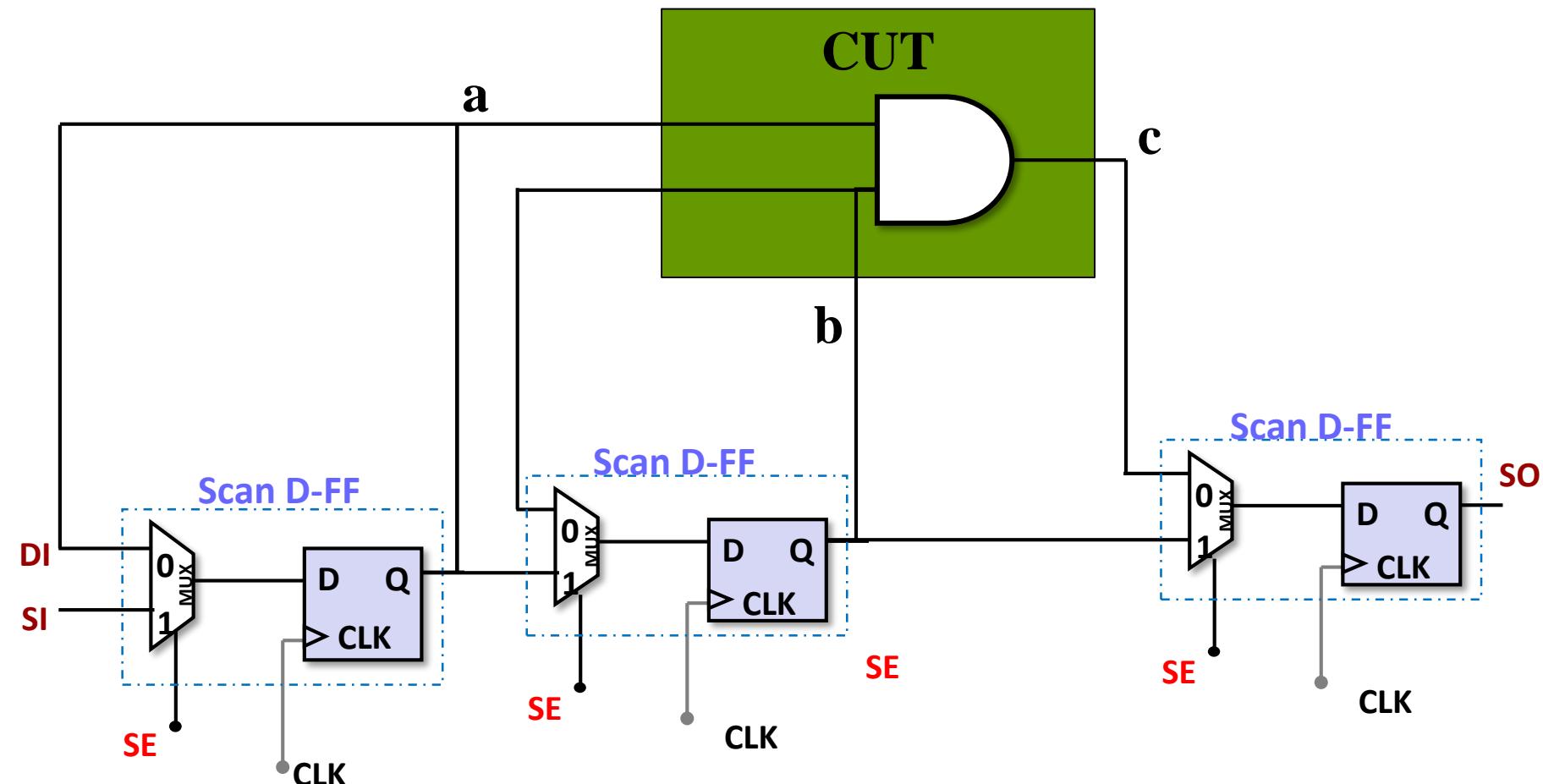
Παράδειγμα σύνδεσης Scan Chain με ένα Circuit under Test (CUT) Βήμα-βήμα, σύνδεση εισόδων



Παράδειγμα σύνδεσης Scan Chain με ένα Circuit under Test (CUT) Βήμα-βήμα, σύνδεση εισόδων

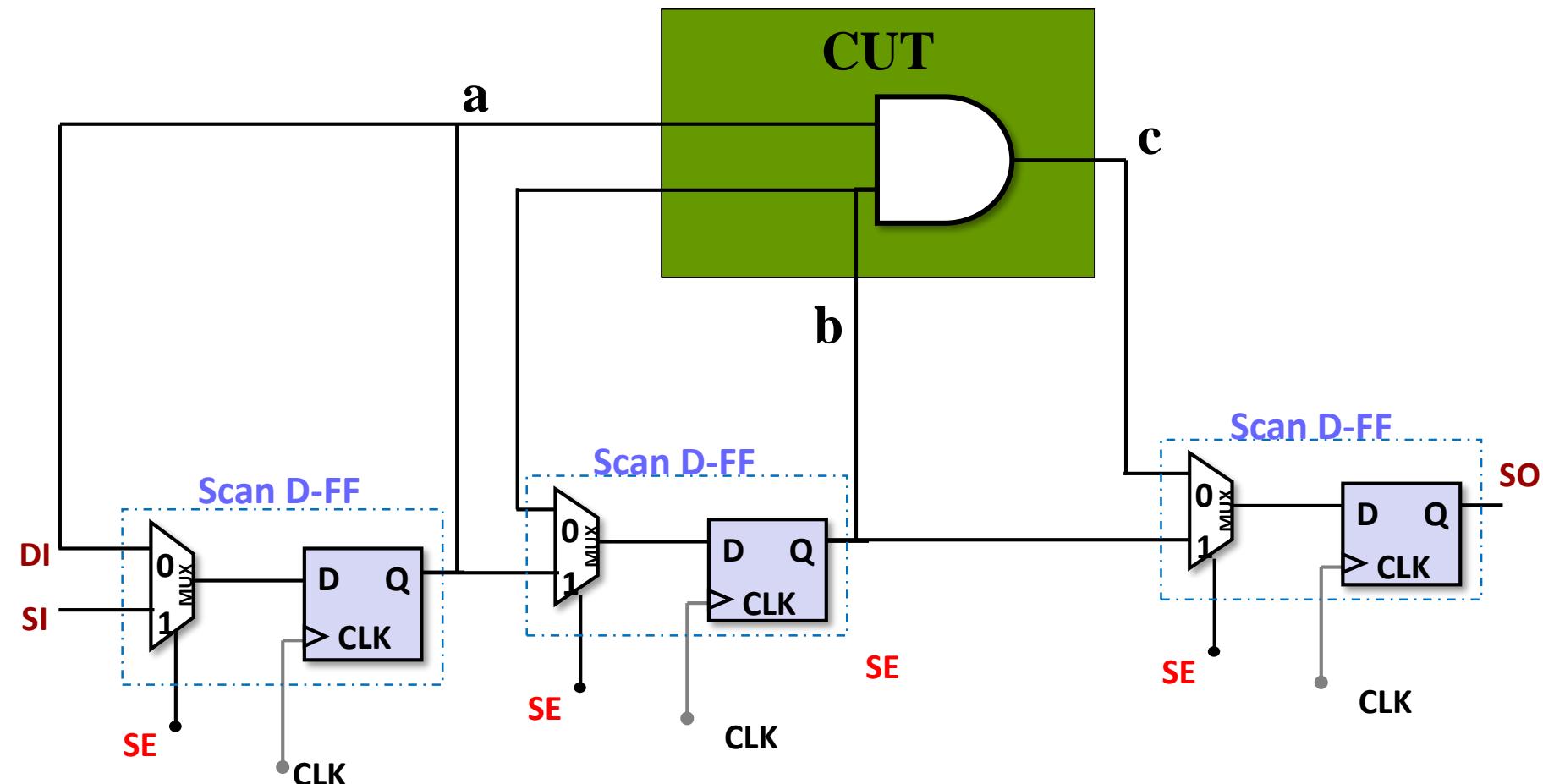


Παράδειγμα σύνδεσης Scan Chain με ένα Circuit under Test (CUT) Βήμα-βήμα, σύνδεση εξόδων



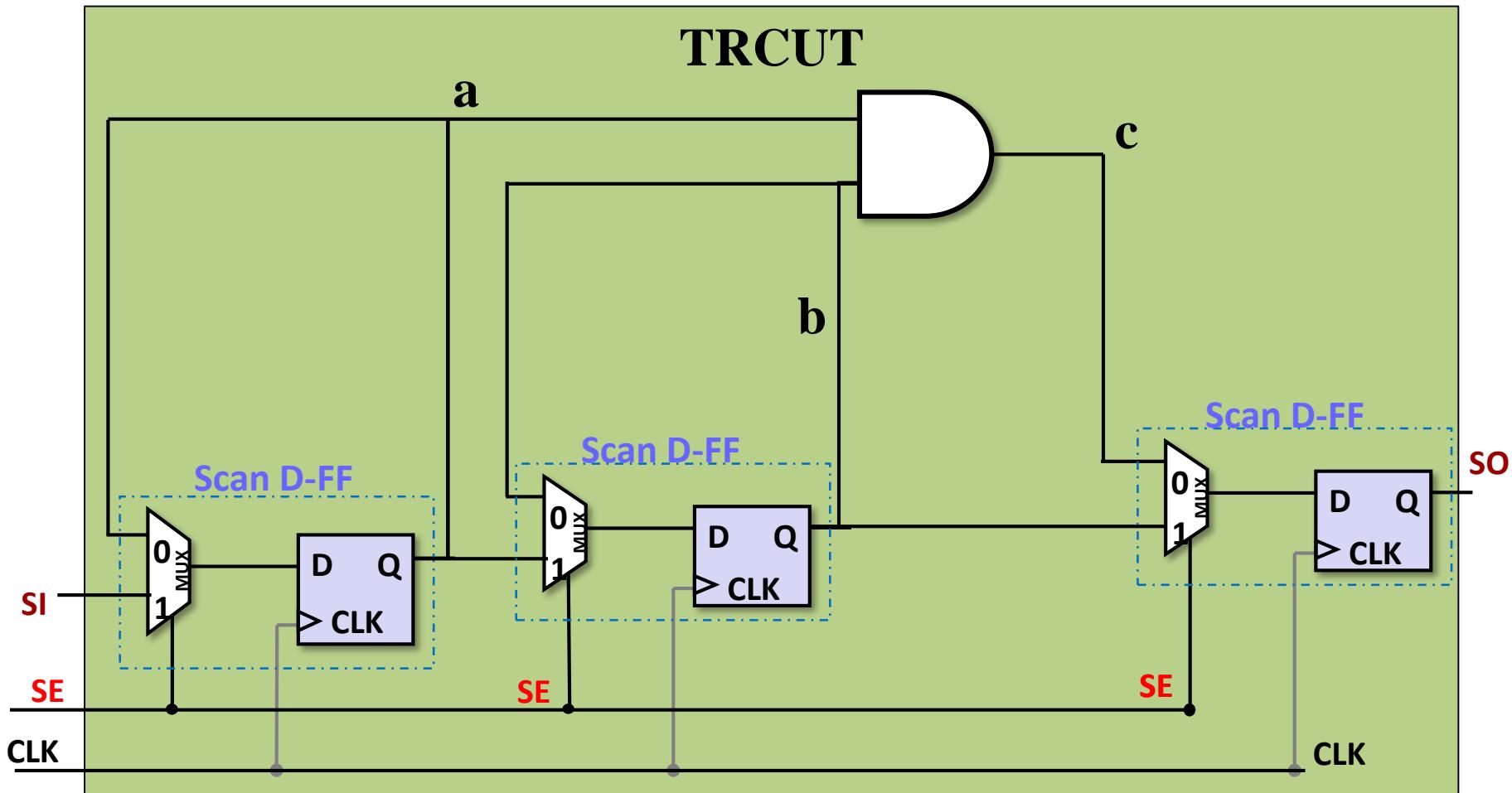
Για λόγους απλότητας θεωρούμε πως κάθε είσοδος/έξοδος έχει το δικό της scan cell

Παράδειγμα σύνδεσης Scan Chain με ένα Circuit under Test (CUT) Βήμα-βήμα, σύνδεση εξόδων



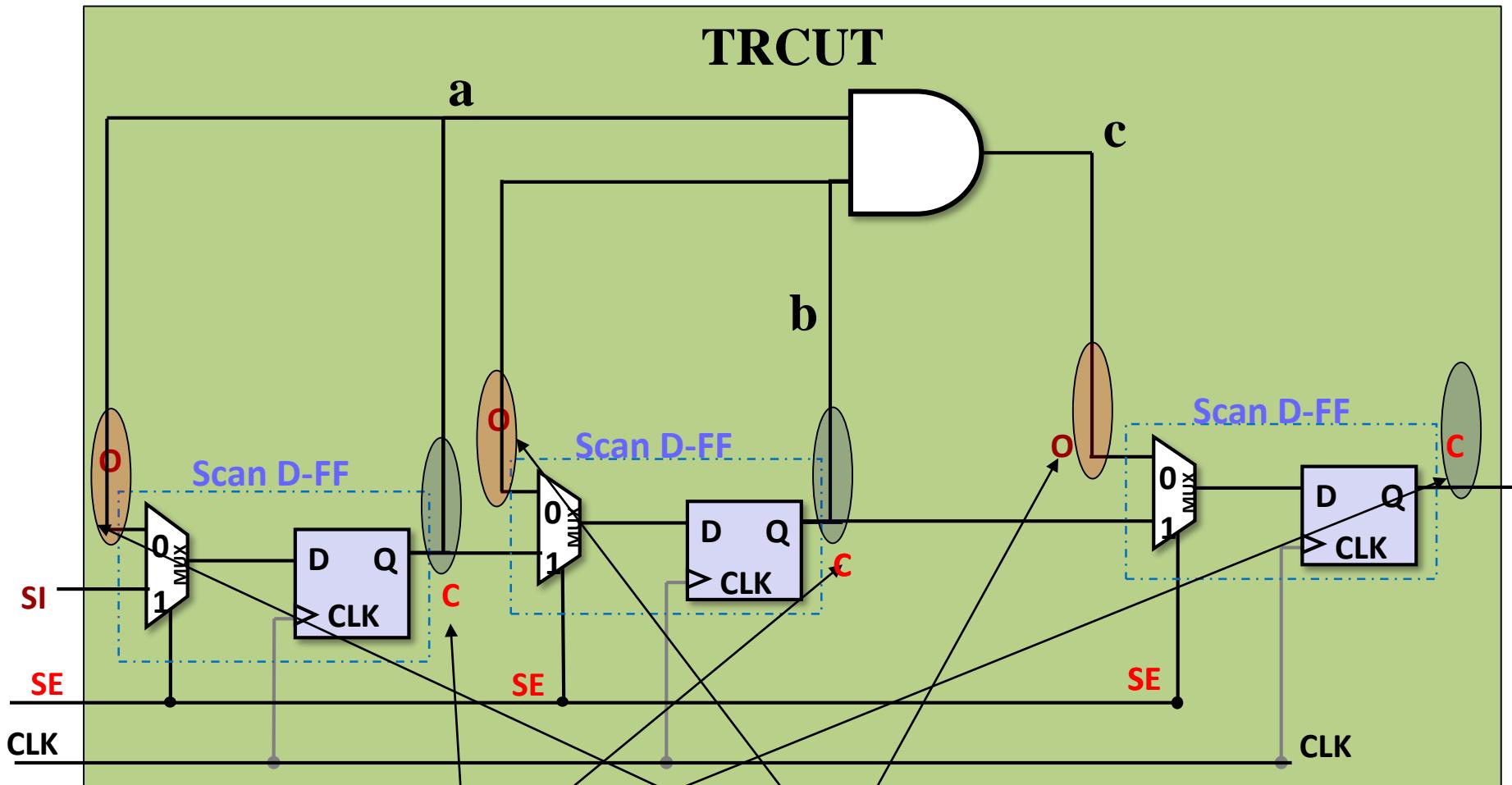
Για λόγους απλότητας θεωρούμε πως κάθε net έχει το δικό του scan cell

Test Ready CUT - TRCUT



Για λόγους απλότητας θεωρούμε πως κάθε net έχει το δικό του can cell

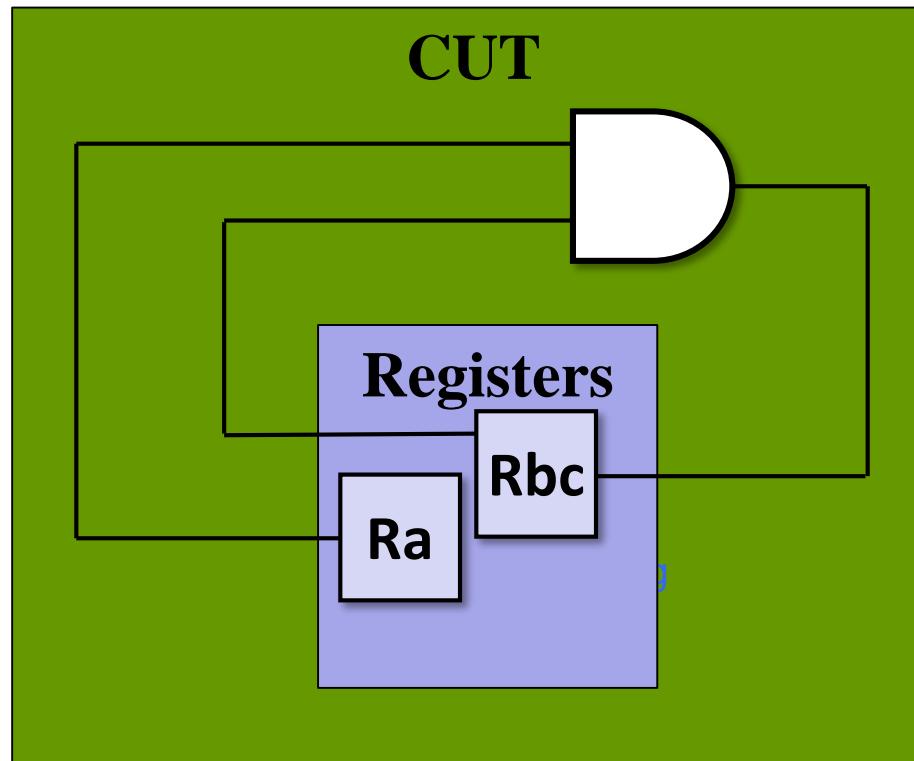
Test Ready CUT - TRCUT



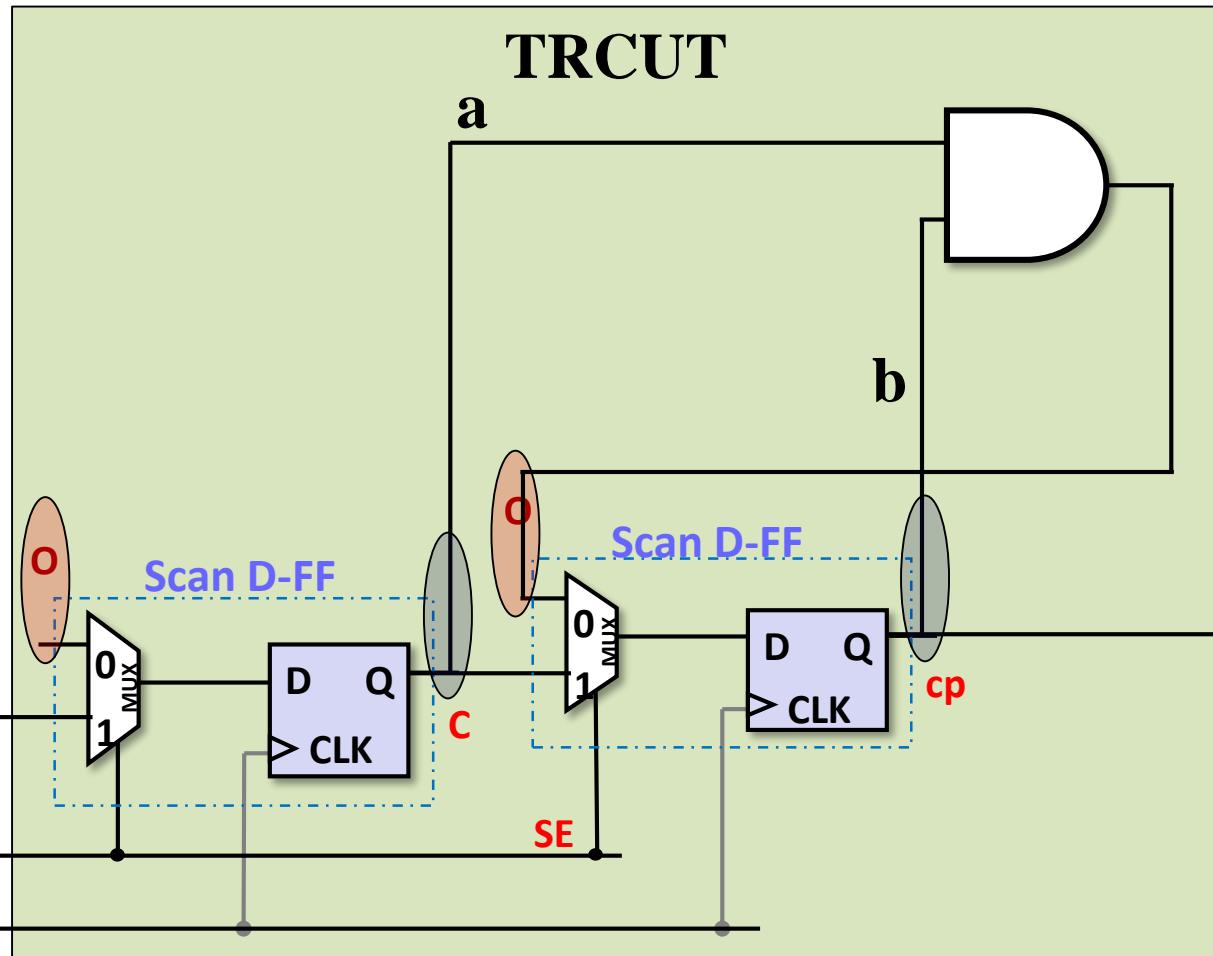
Για λόγους απλότητας θεωρούμε πως κάθε net έχει το δικό του can cell,
C: controllability points, O: observability points

Scan Testing

Παράδειγμα: με διαφορετικό αρχικό ακολουθιακό κύκλωμα



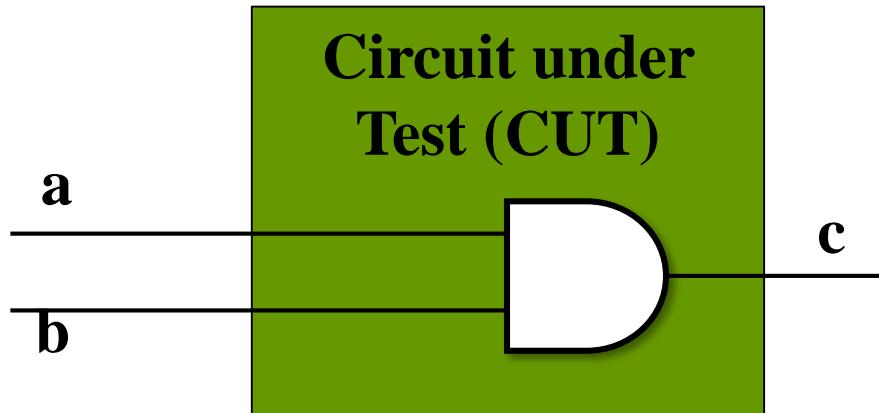
Βελτιστοποίηση αλυσίδας



Ένα scan cell μπορεί να χρησιμοποιείτε τόσο ως controllability point όσο και ως observability point ταυτόχρονα ακόμα κ για διαφορετικά nets – επίσης SO δεν χρειάζονται όλα τα controllability κ observability points για να έχουμε πλήρη ελεγξιμότητα

Παράδειγμα σύνδεσης Scan Chain με ένα Circuit under Test (CUT)

Βήμα-βήμα, σύνδεση εισόδων



a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

testvector expected response

a b c

-> a b c

Sa0-a: 1 1 x

-> 1 1 1

Sa1-a: 0 1 x

-> 0 1 0

Sa0-b: 1 1 x

-> 1 1 1

Sa1-b: 1 0 x

-> 1 0 0

Sa1-c: 0 x x

-> 0 x 0

Sa0-c: 1 1 x

-> 1 1 1

Sa0-a, Sa0-b, Sa0-c

1 1 x -> 1 1 1

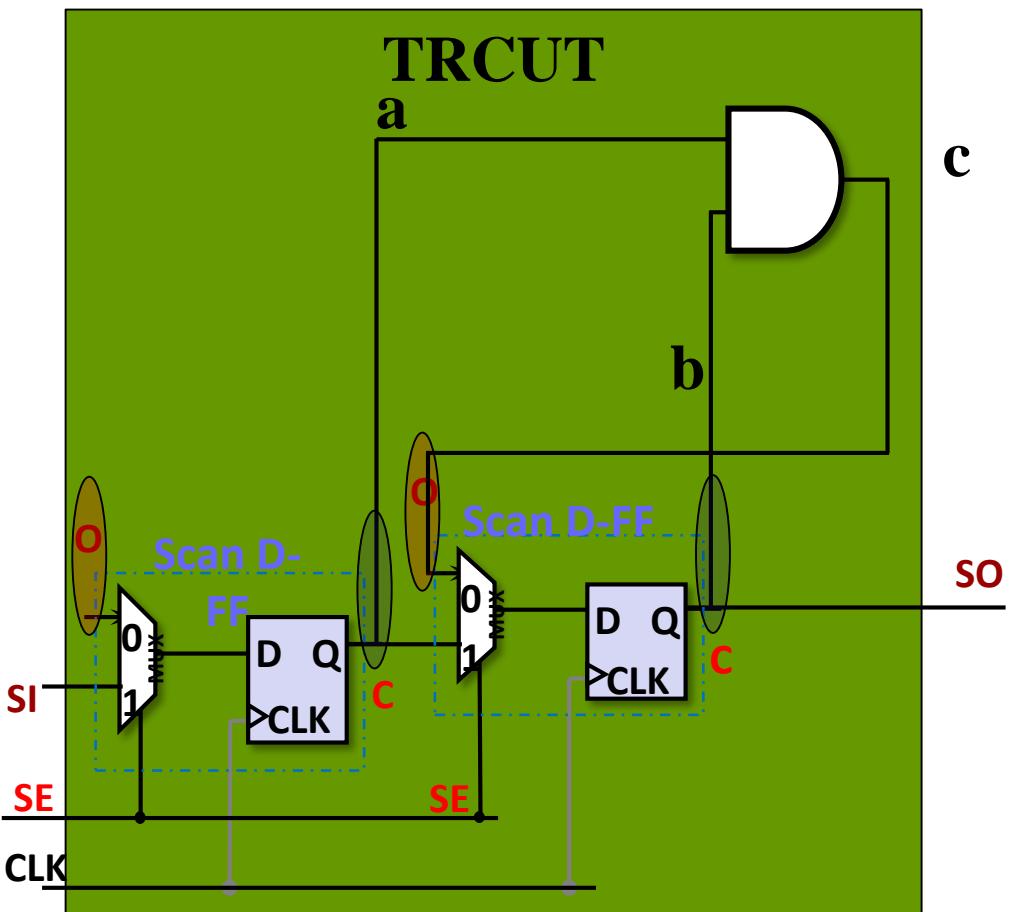
Sa1-a, , sa1-c

0 1 x -> 0 1 0

Sa1-b

1 0 x -> 1 0 0

Βελτιστοποίηση αλυσίδας



Sa0-a, Sa0-b, Sa0-c

1 1 x -> 1 1 1

Sa1-a, , sa1-c

0 1 x -> 0 1 0

Sa1-b

1 0 x -> 1 0 0



Sa0-a, Sa0-b, Sa0-c

1 1 -> x 1

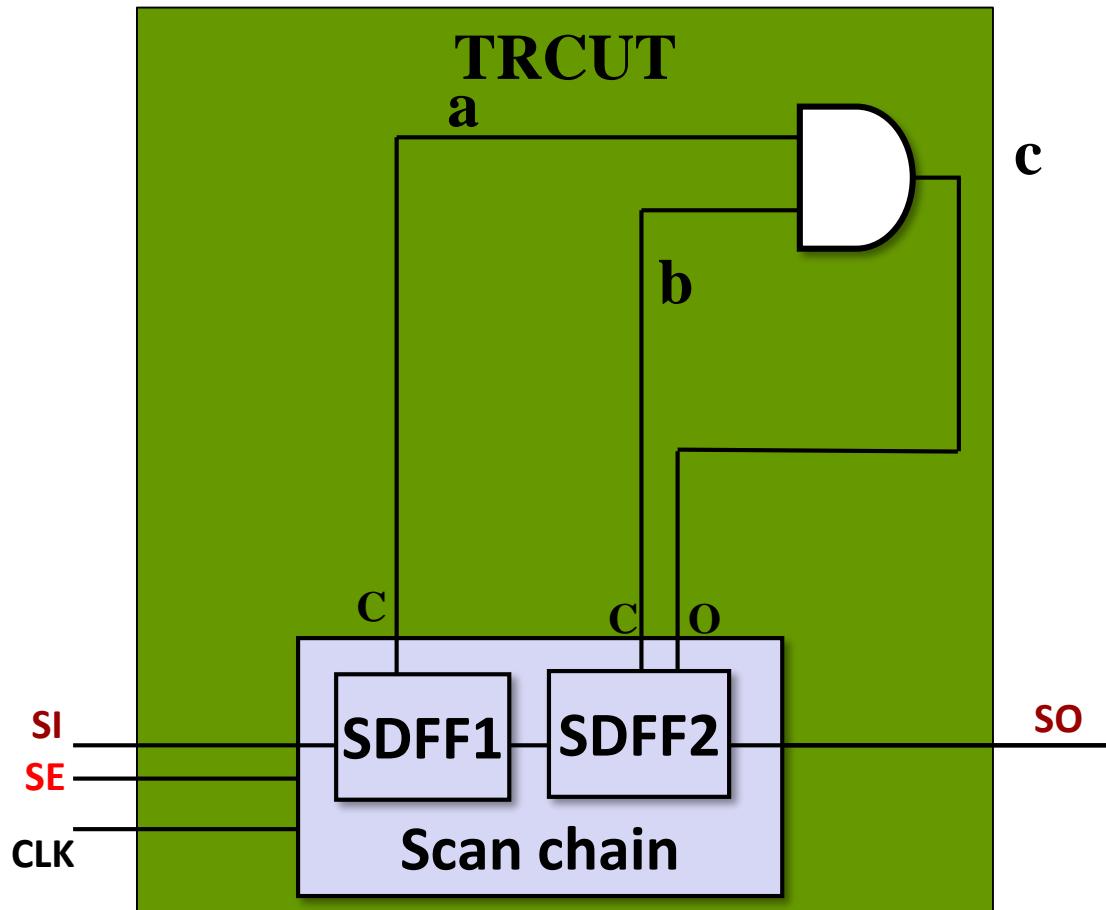
Sa1-a, sa1-c

0 1 -> x 0

Sa1-b

1 0 -> x 0

Βελτιστοποίηση αλυσίδας και επαναπροσδιορισμός του testset



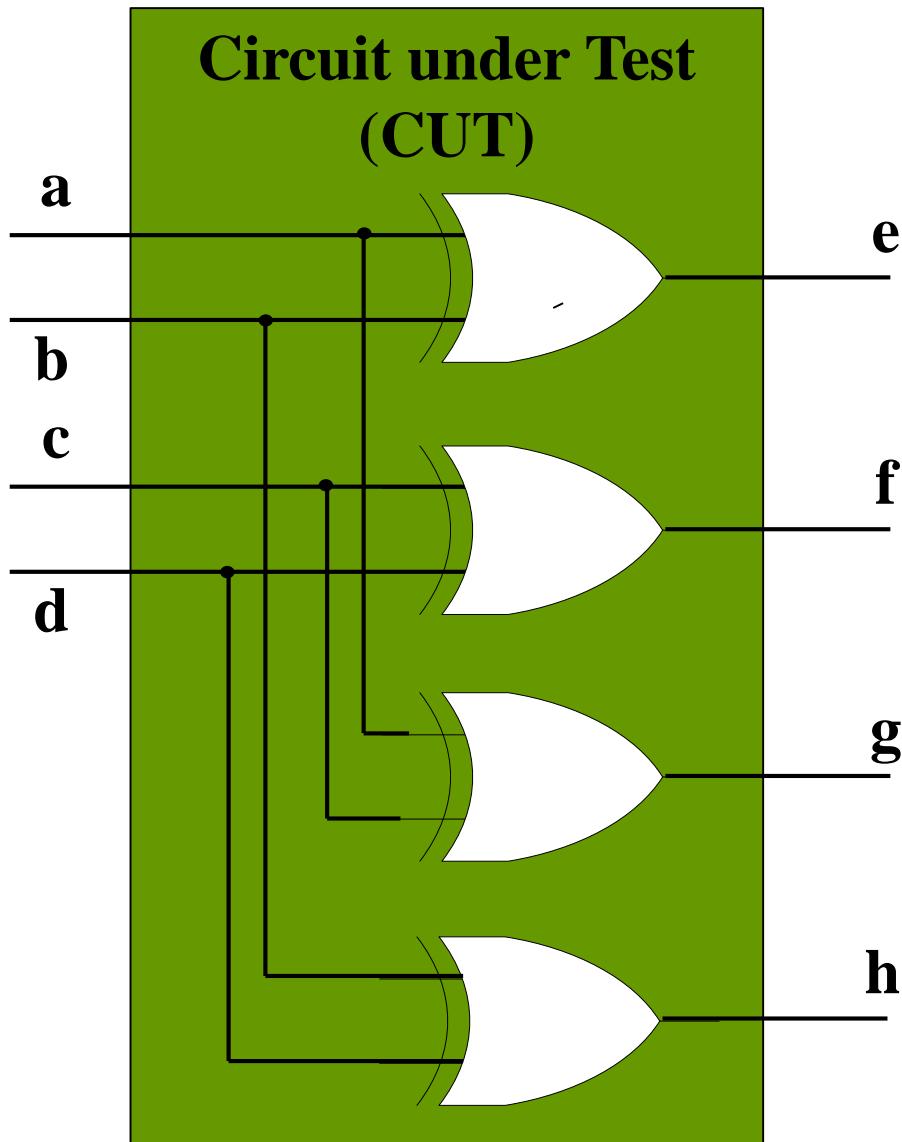
Stuck-at testset

SI : SO
Sa0-a, 1 1 : x 1
Sa0-b,
Sa0-c

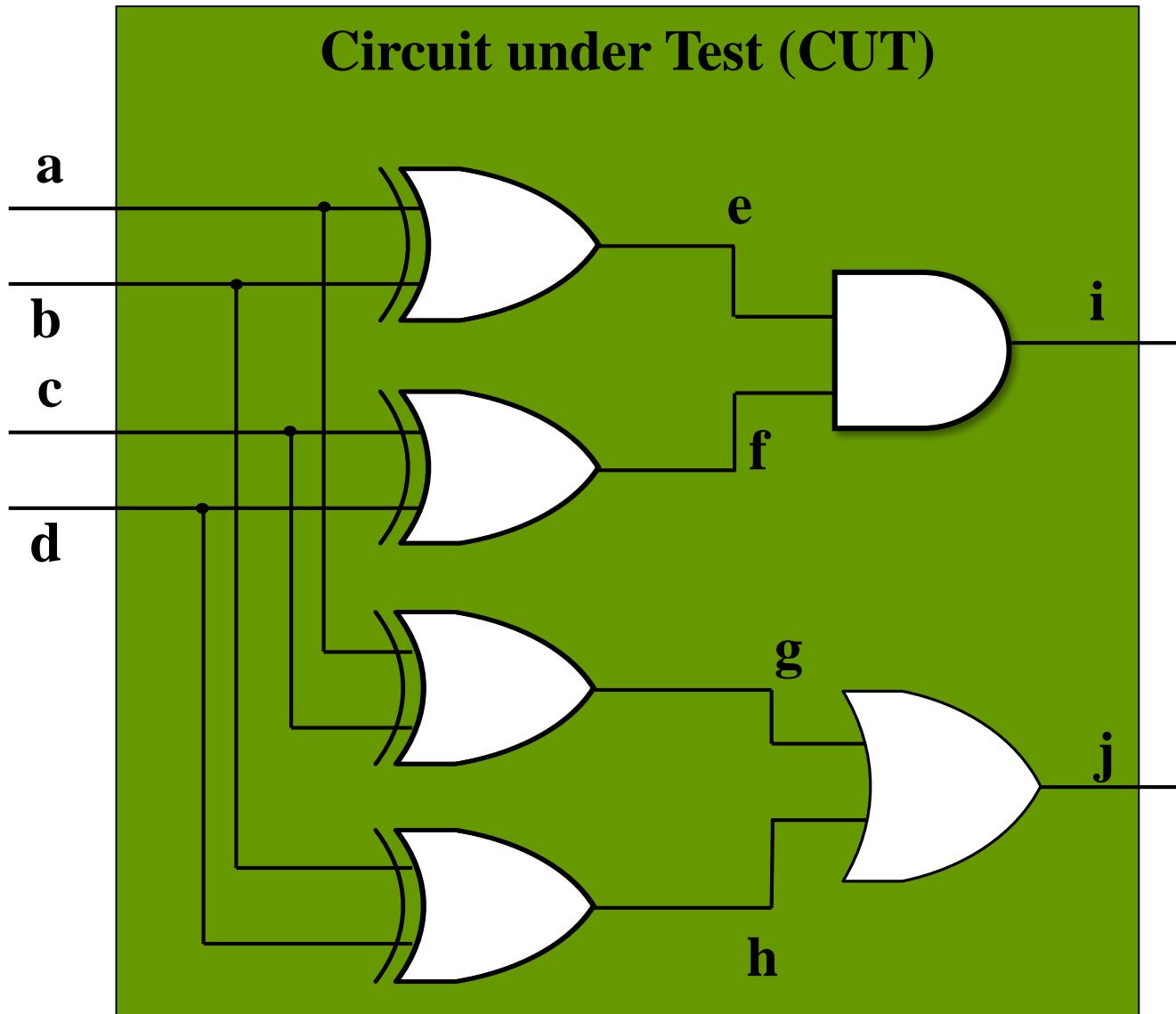
Sa1-a, 0 1 : x 0
sa1-c

Sa1-b, 1 0 : x 0

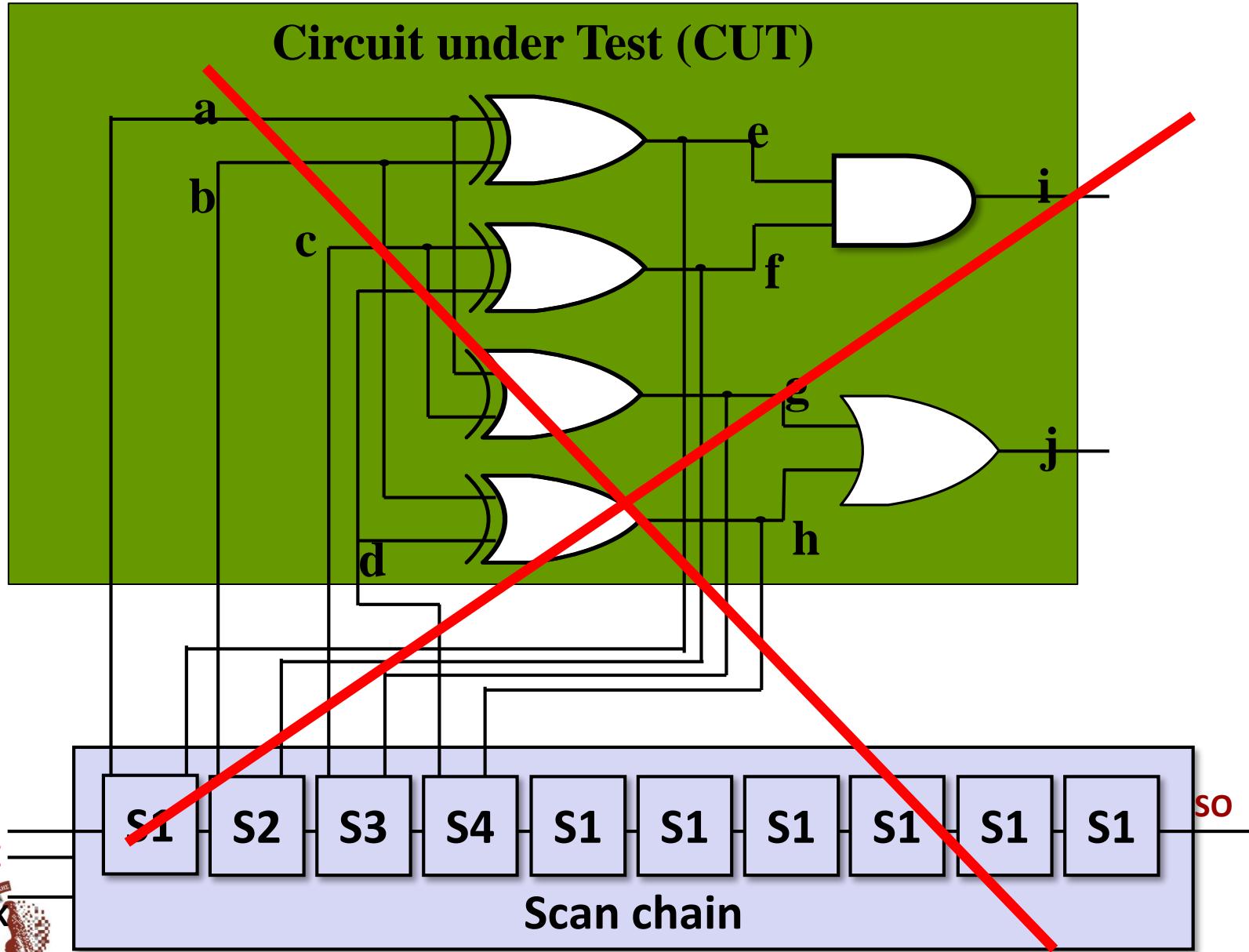
Άσκηση – scan chain για το παρακάτω κύκλωμα



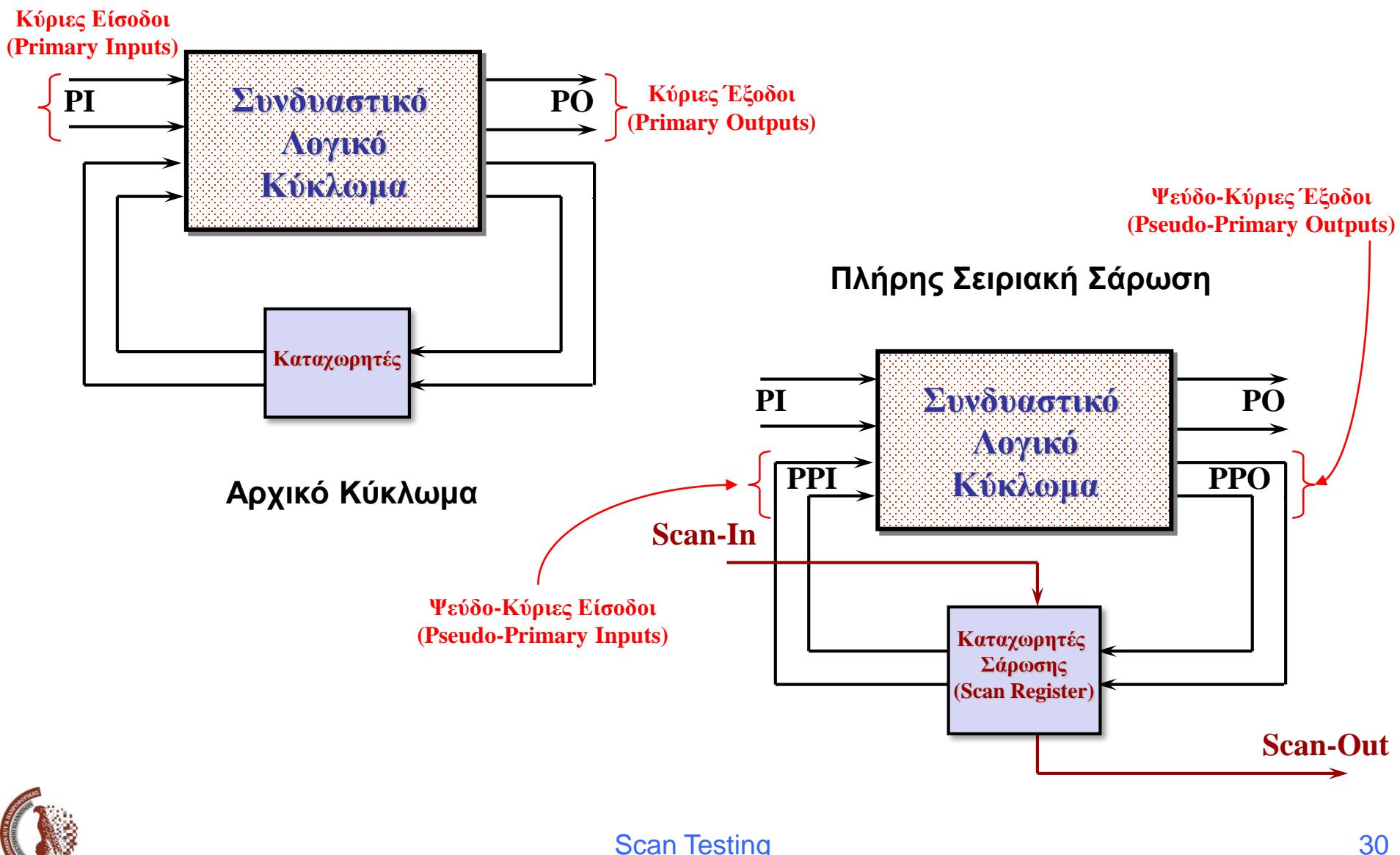
Άσκηση – scan chain για το παρακάτω κύκλωμα



Άσκηση – scan chain για το παρακάτω κύκλωμα

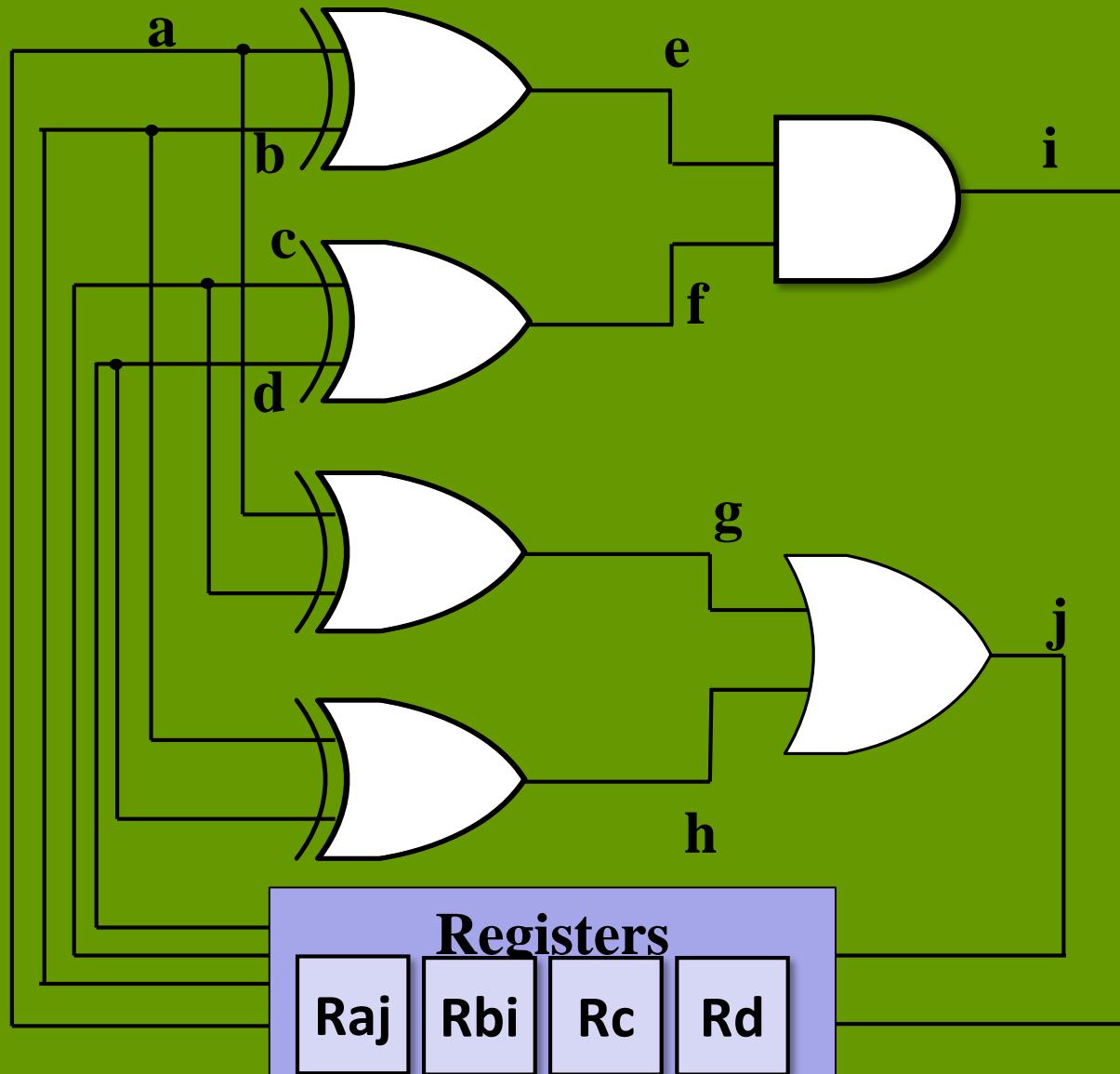


Θυμηθείτε ότι ο Σχεδιασμός Σειριακής Σάρωσης αφορά ακολουθιακά κυκλώματα!

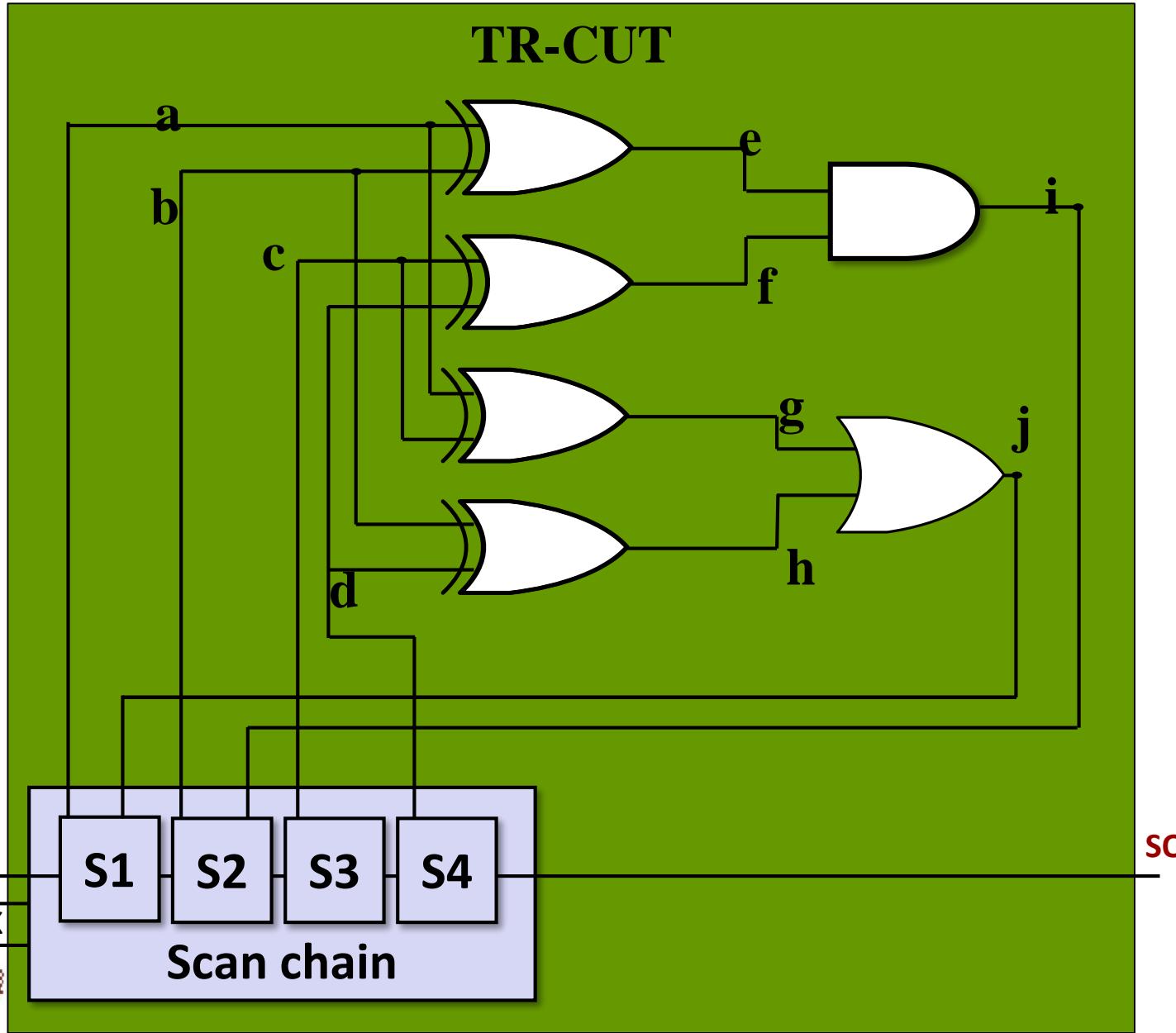


**Άσκηση – το κύκλωμα δηλαδή που το εφαρμόζουμε
Θα είναι ακολουθιακό ήδη:**

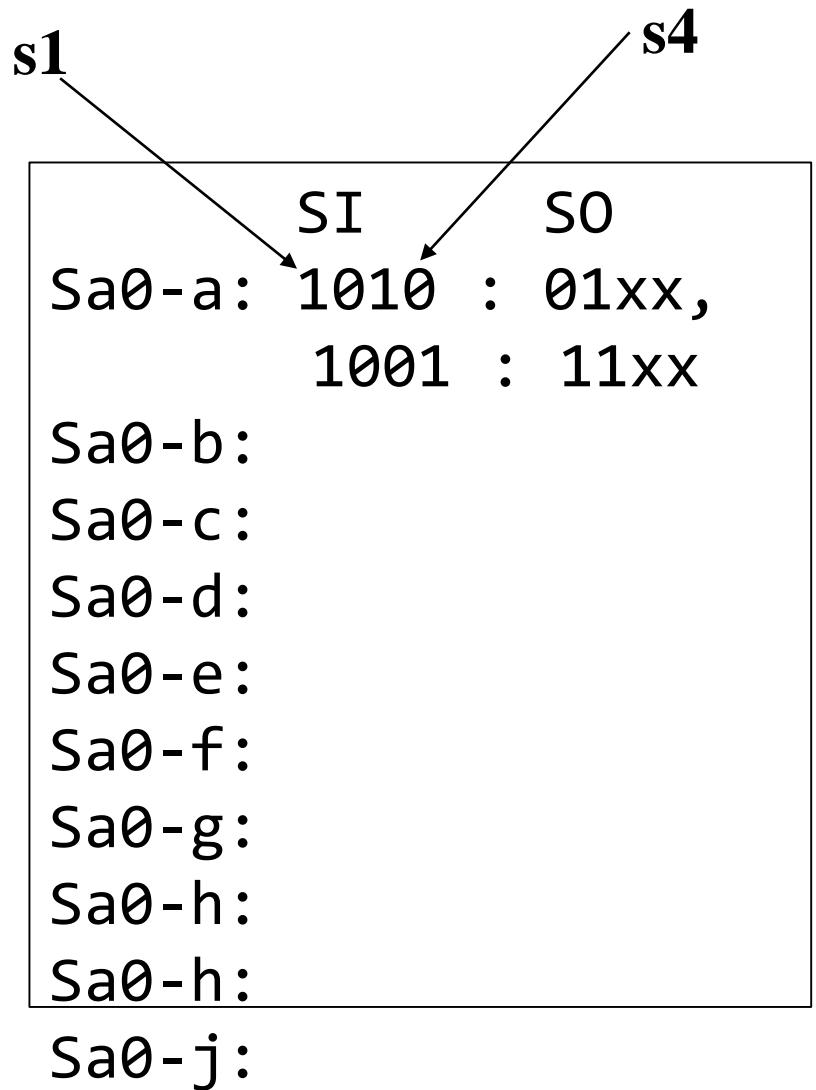
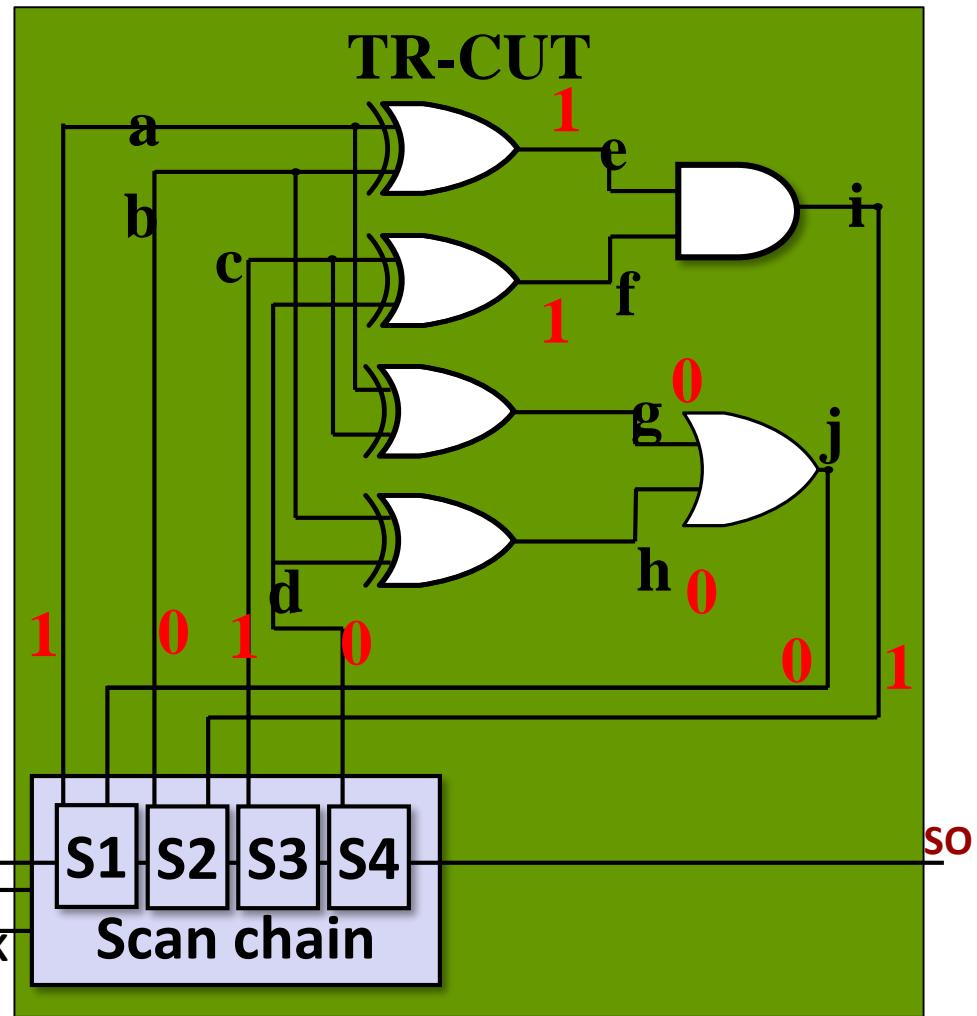
Circuit under Test (CUT)



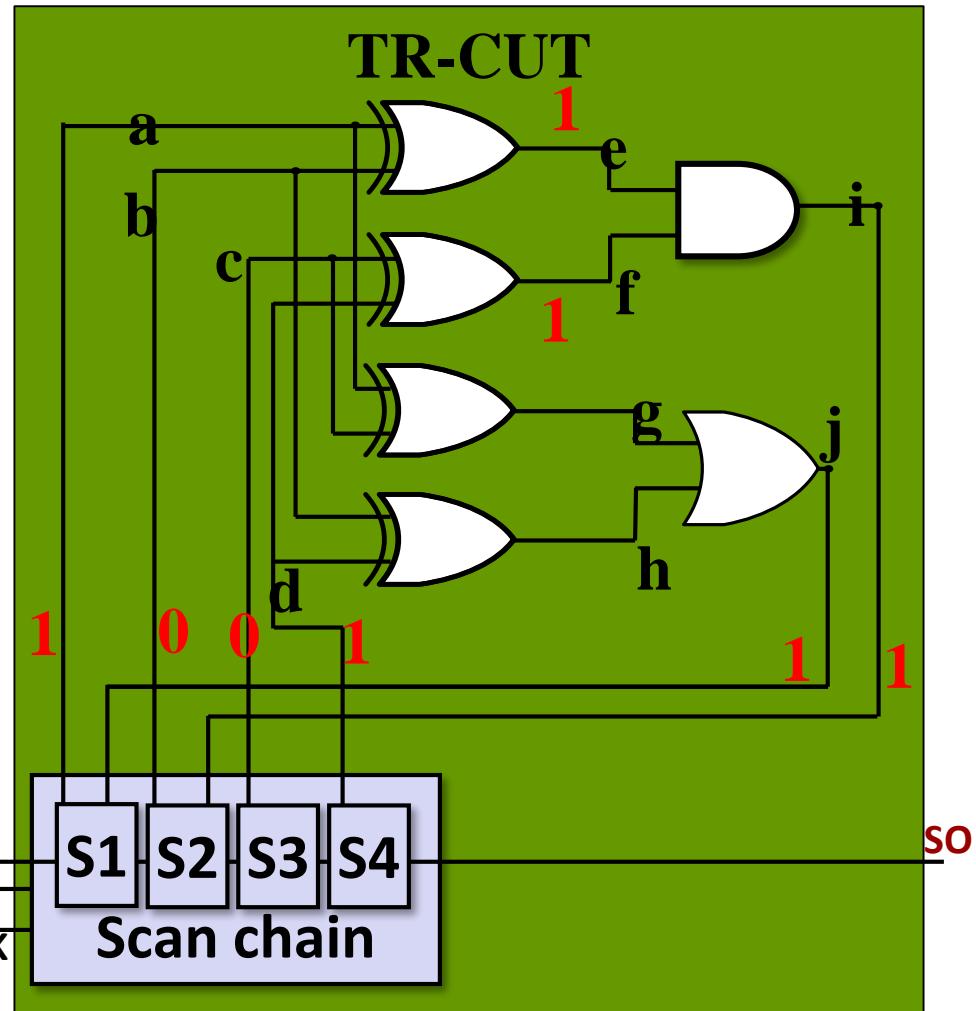
Άσκηση – scan chain για το παρακάτω κύκλωμα



Άσκηση – stuck-at testset για το παρακάτω κύκλωμα

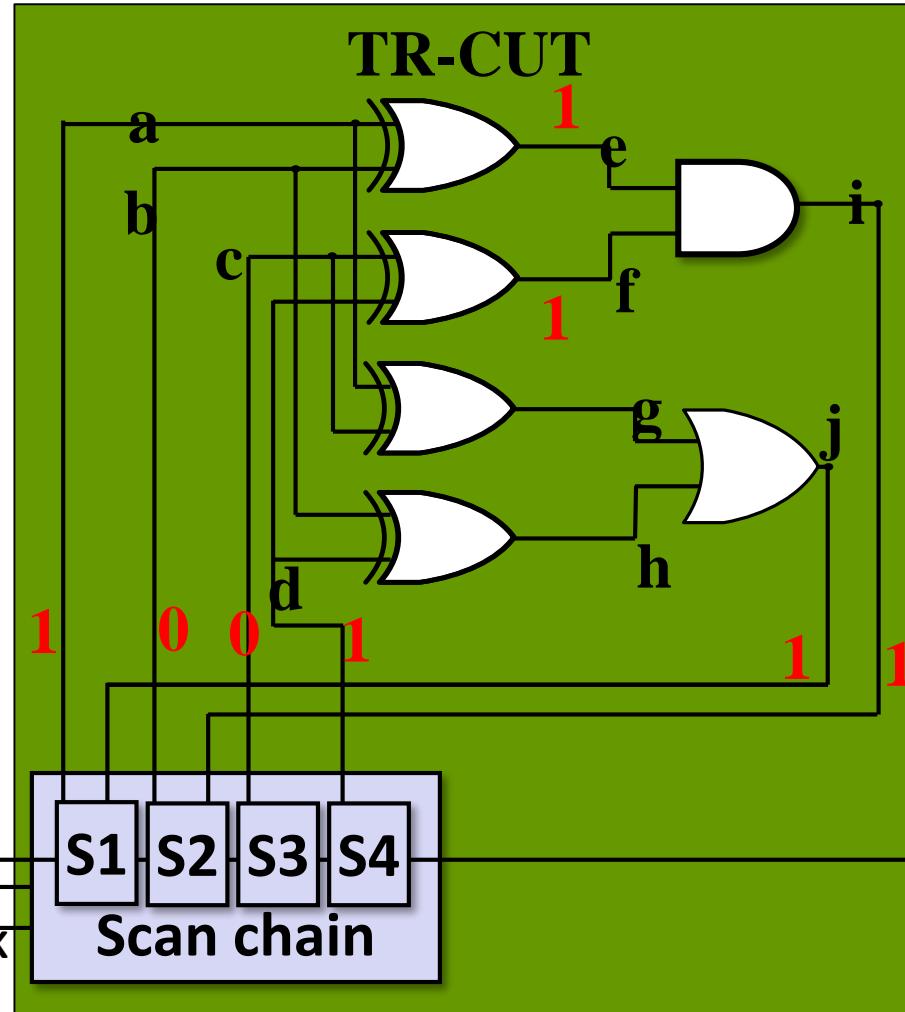


Άσκηση – stuck-at testset για το παρακάτω κύκλωμα



SI	SO
Sa1-a:	
Sa1-b:	
Sa1-c:	
Sa1-d:	
Sa1-e:	
Sa1-f:	
Sa1-g:	
Sa1-h:	
Sa1-i:	
Sa1-j:	

Άσκηση – stuck-at testset για το παρακάτω κύκλωμα

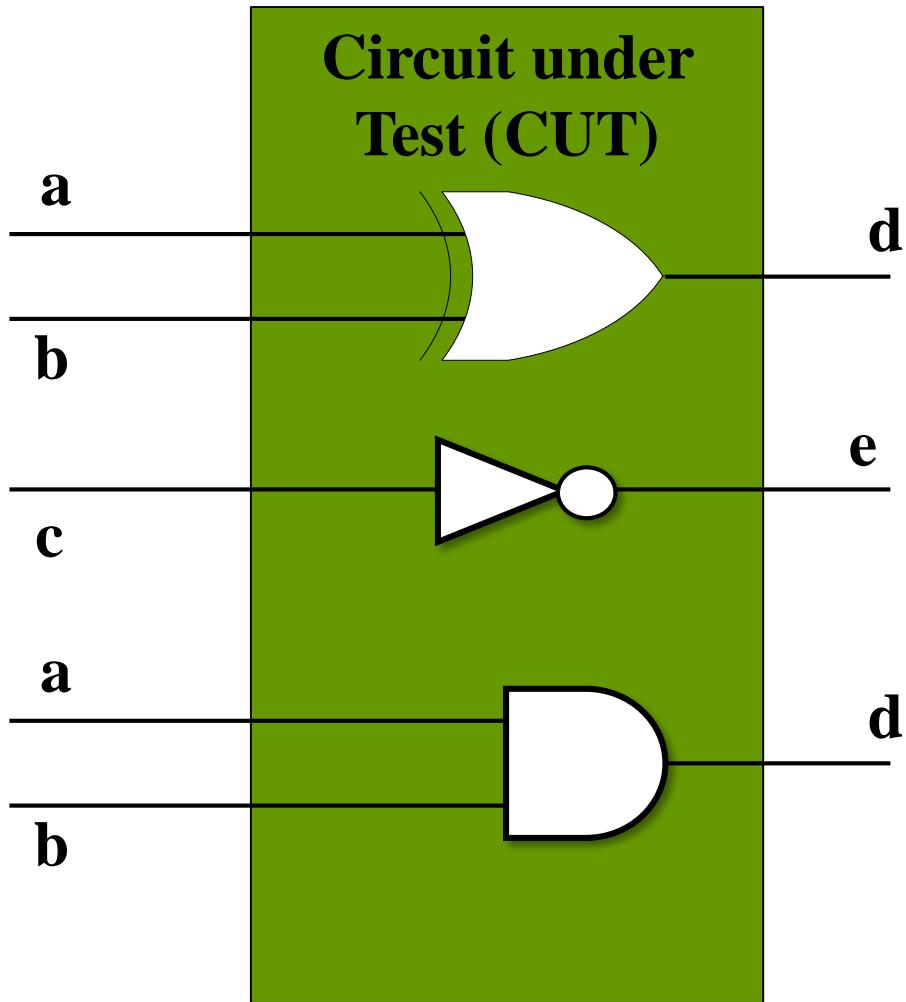
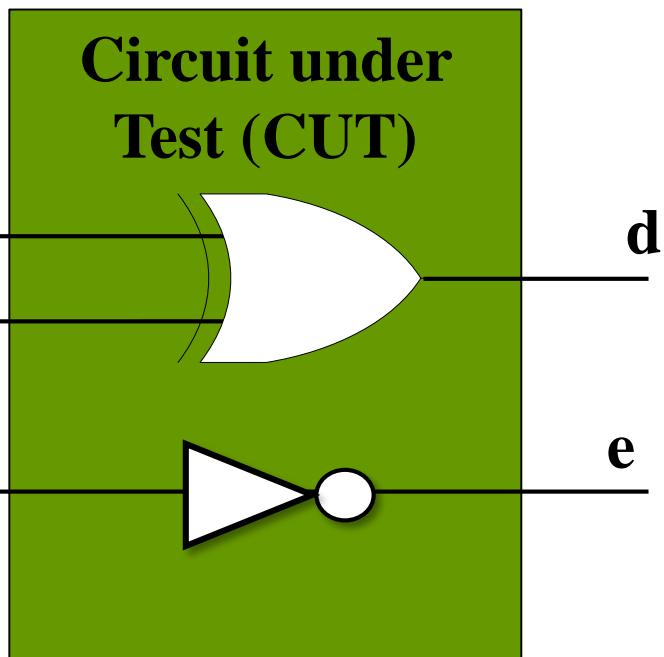


Λογισμικό για σχέδια συστήνω το Visio που μπορείτε να κατεβάσετε δωρεάν εδώ:

<https://azureforeducation.microsoft.com/devtools>

Με τον academic λογαριασμό σας
Κατεβάστε το Visio και σχεδιάστε το
διπλανό σχήμα στο Visio
εγώ έχω την έκδοση 2016, έχει βγει
και νέα αλλά δεν την έχω δοκιμάσει –
καλύτερα βάλτε την 16 μην τυχόν και
υπάρχουν αλλαγές

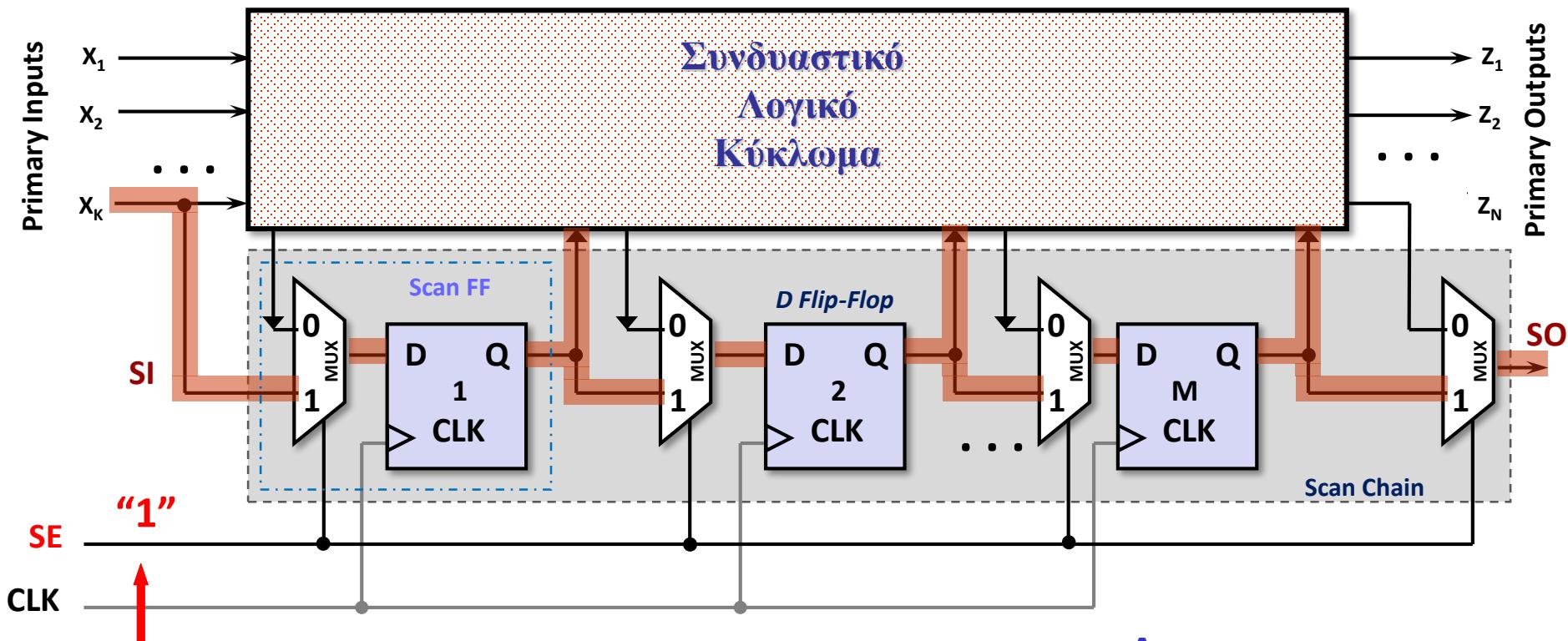
Άσκηση – κυκλώματα για εξάσκηση



Τέλος Παραδειγμάτων

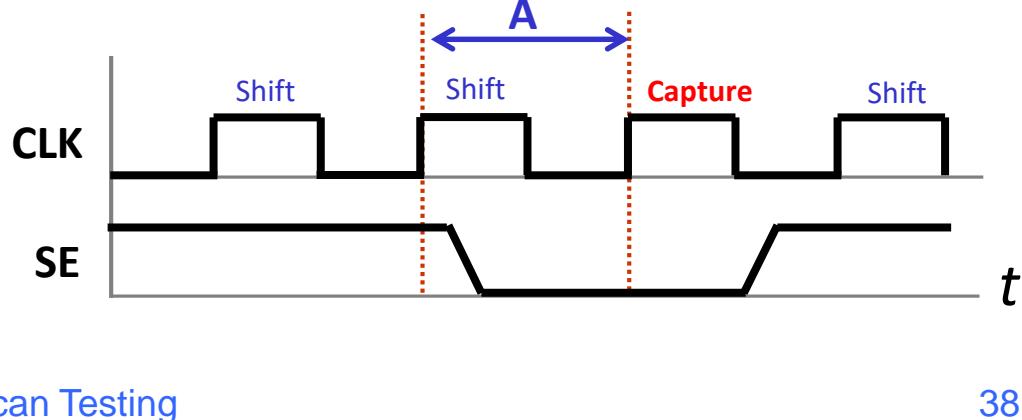


Υλοποίηση Σειριακής Σάρωσης (I)

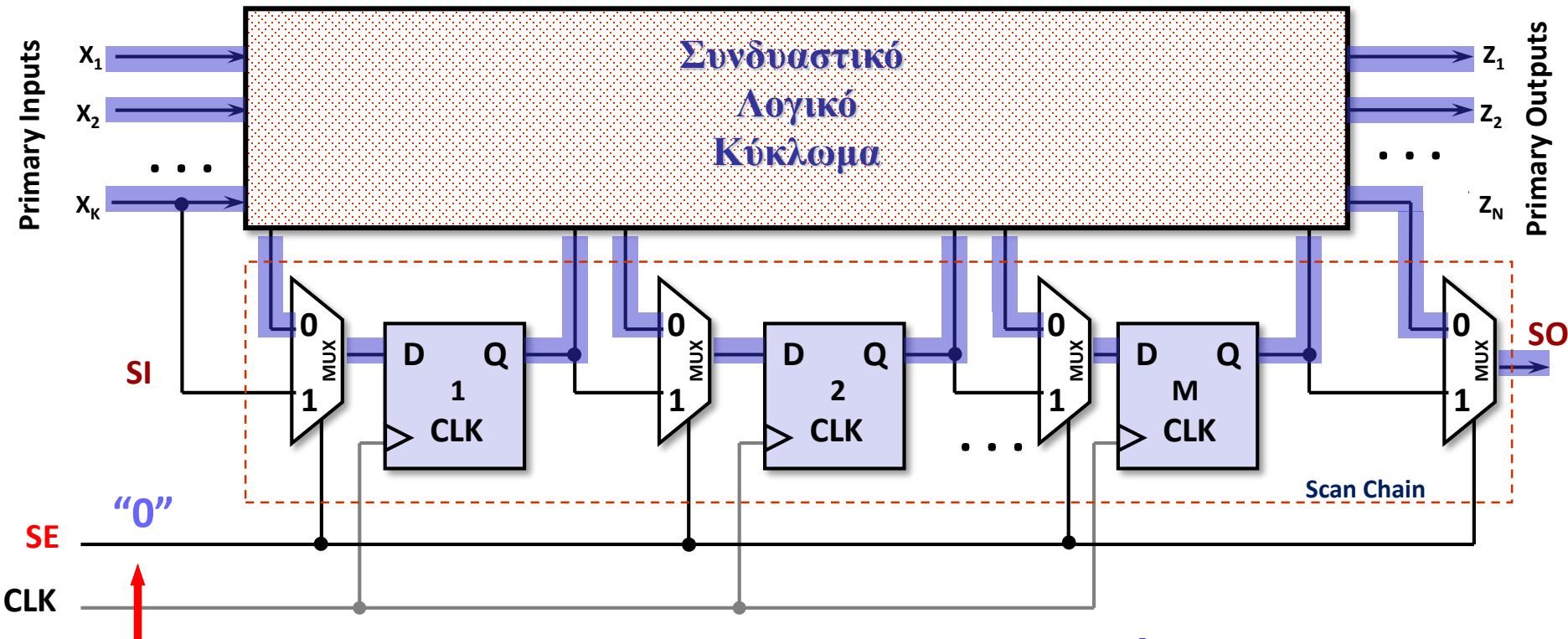


Σειριακή Σάρωση (Scan Mode)

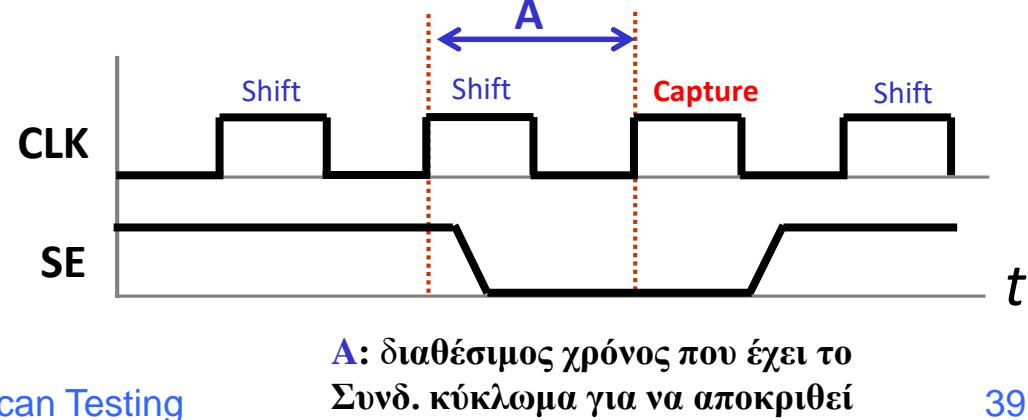
SE=“1” (Scan Enable = 1)



Υλοποίηση Σειριακής Σάρωσης (II)



Κανονική Λειτουργία (Normal Mode)
SE="0" (Scan Enable= 0)

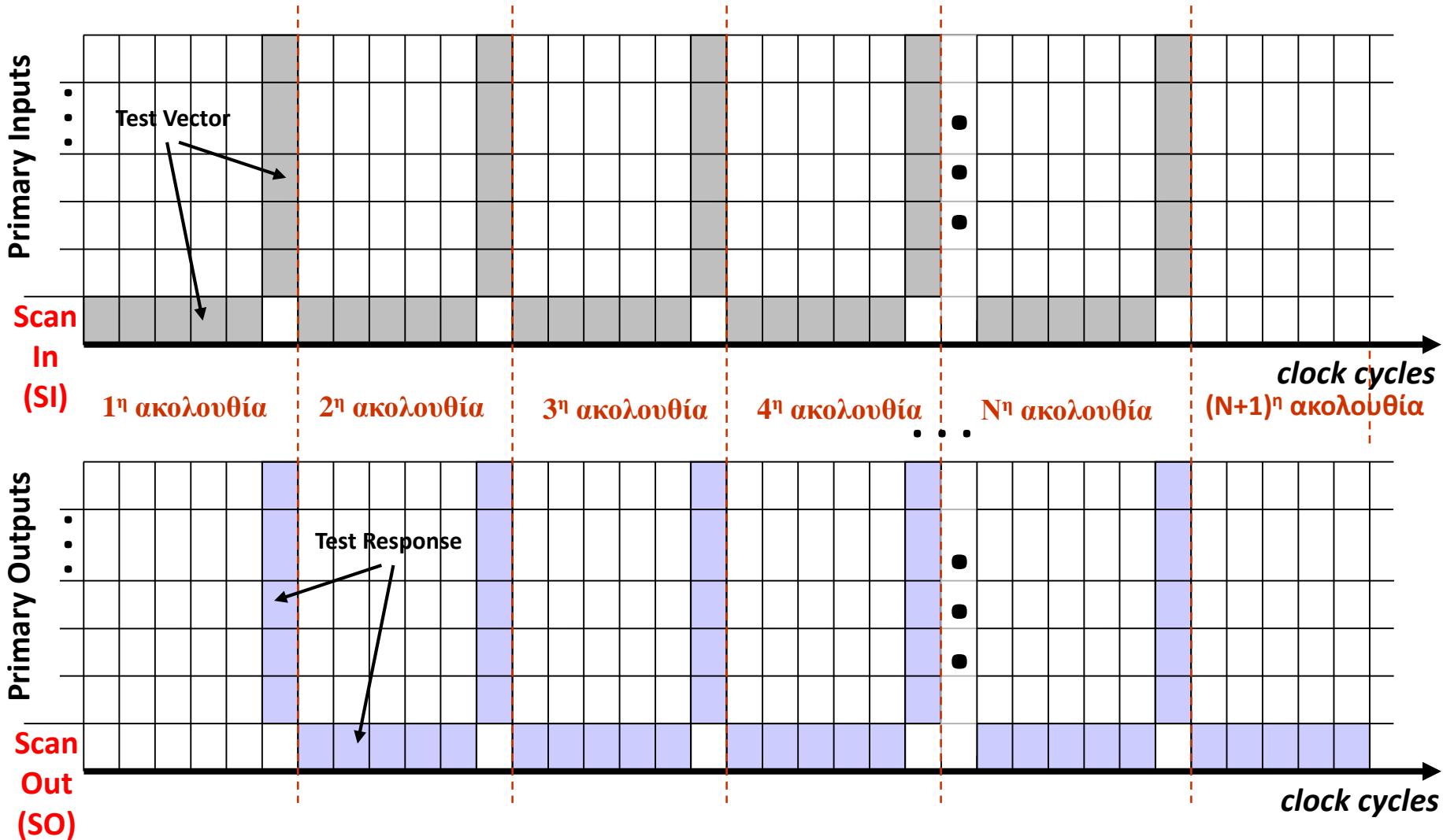


Βήματα Σειριακής Σάρωσης

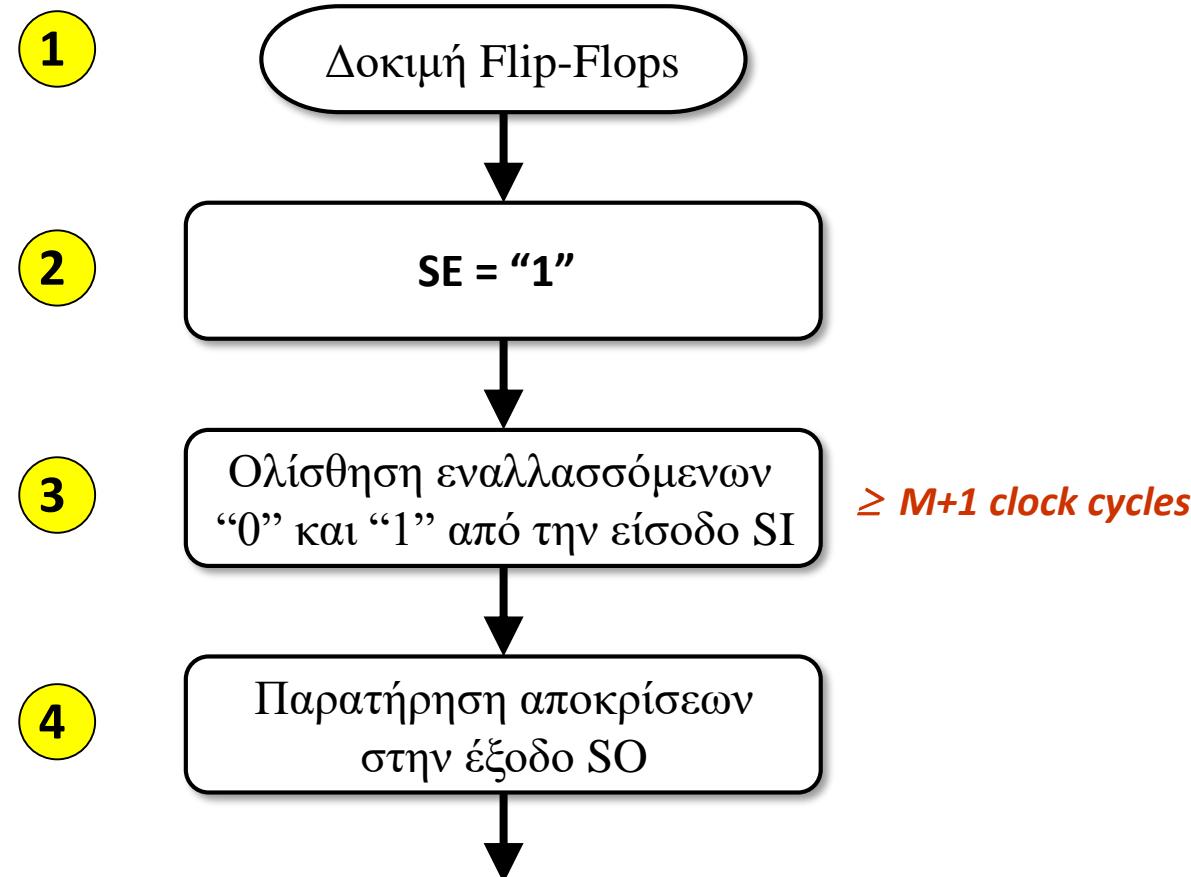
1. Φορτώνουμε σειριακά (με ολίσθηση) το διάνυσμα δοκιμής στην αλυσίδα σάρωσης μέσω της εισόδου **Scan In (SI)**. Το σήμα **Scan Enable (SE)** είναι 1. Χρειάζονται **L κύκλοι ρολογιού**, όπου **L** το μήκος της αλυσίδας.
2. Αφού φορτωθεί το διάνυσμα δοκιμής γίνεται **σύλληψη (capture)** της απόκρισης της λογικής στο διάνυσμα αυτό, δηλαδή η **παράλληλη** φόρτωση των αποκρίσεων στη scan αλυσίδα. Για να γίνει αυτό θέτουμε **SE =0** για **1 παλμό ρολογιού**.
3. Στη συνέχεια θέτουμε **SE = 1** για να φορτωθεί το επόμενο διάνυσμα δοκιμής. Καθώς το νέο διάνυσμα δοκιμής ολισθαίνει από την είσοδο SI, η απόκριση της λογικής που φορτώθηκε στη scan αλυσίδα στο βήμα 2 ολισθαίνει έξω μεσω της εξόδου **Scan Out (SO)** και έτσι μπορούμε να την **παρατηρήσουμε**.
4. Αφού γίνουν τα βήματα 1 και 2 για το **τελευταίο** διάνυσμα δοκιμής, πρέπει να κάνουμε **L** επιπλέον ολισθήσεις για να πάρουμε τις αποκρίσεις (του τελευταίου διανύσματος) στην έξοδο SO.



Ακολουθίες Δοκιμής κατά τη Σάρωση

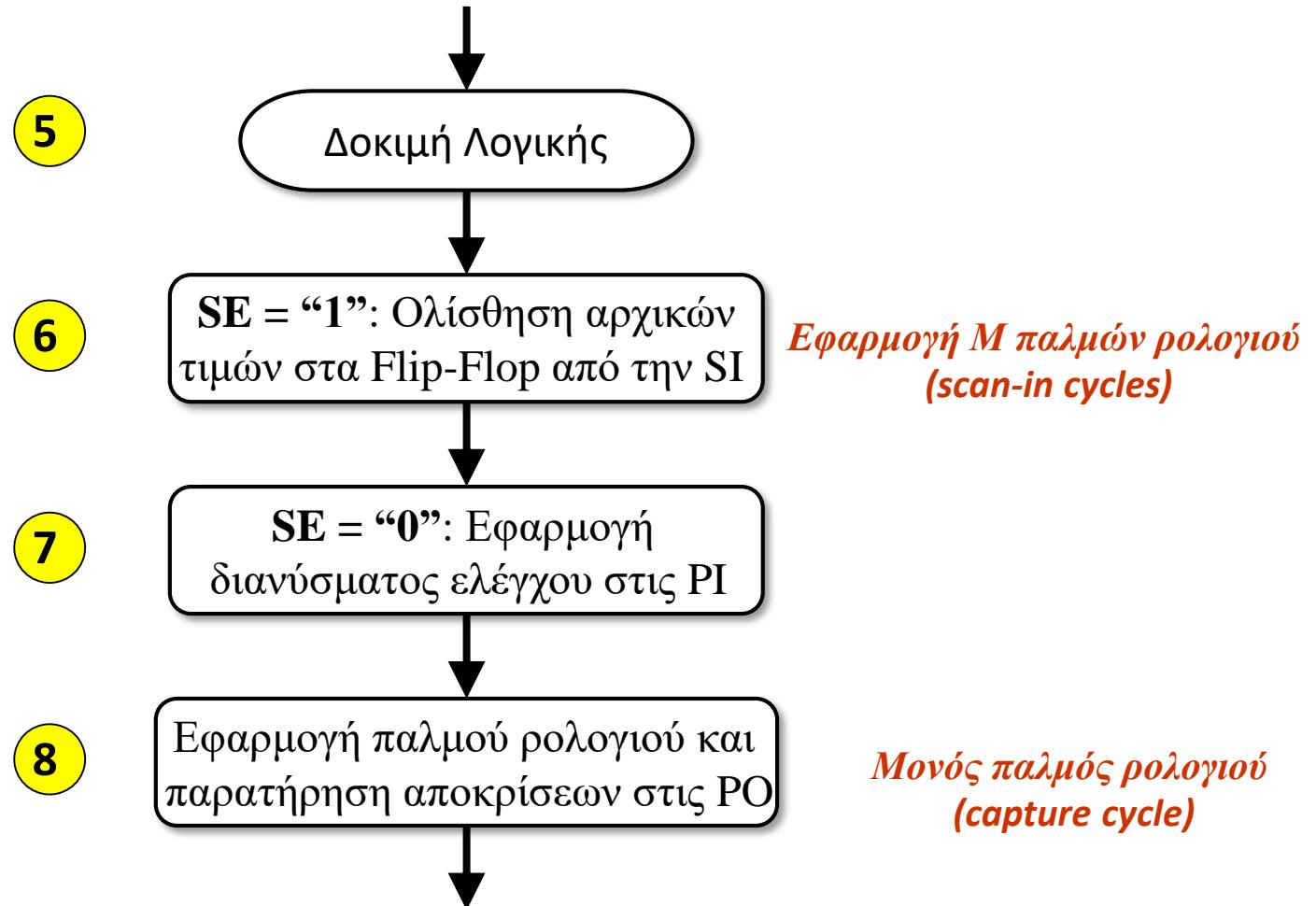


Εφαρμογή Σειριακής Σάρωσης (I)

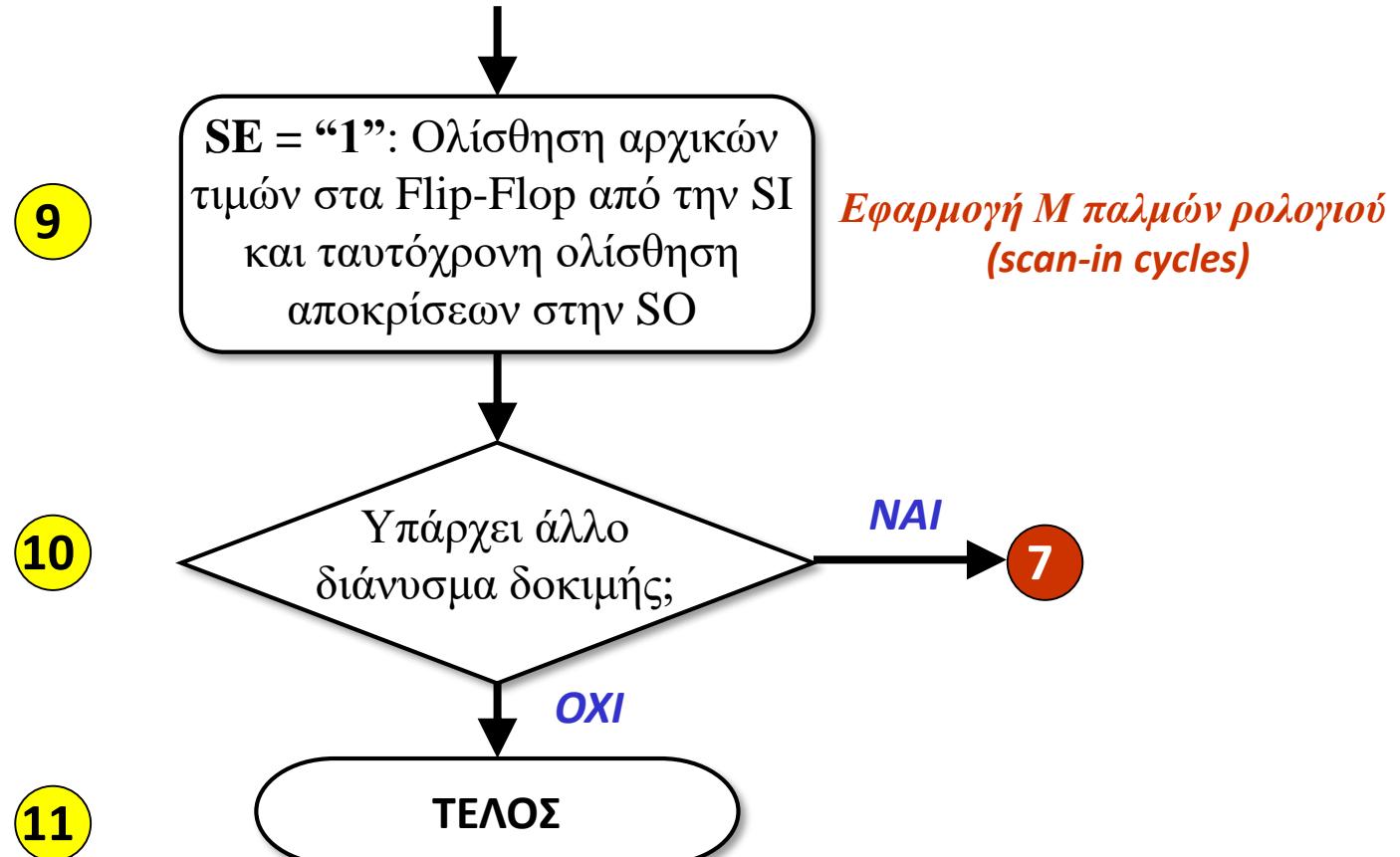


$M = \# \text{ of scan cells}$

Εφαρμογή Σειριακής Σάρωσης (II)



Εφαρμογή Σειριακής Σάρωσης (III)



Άσκηση Σειριακής Σάρωσης

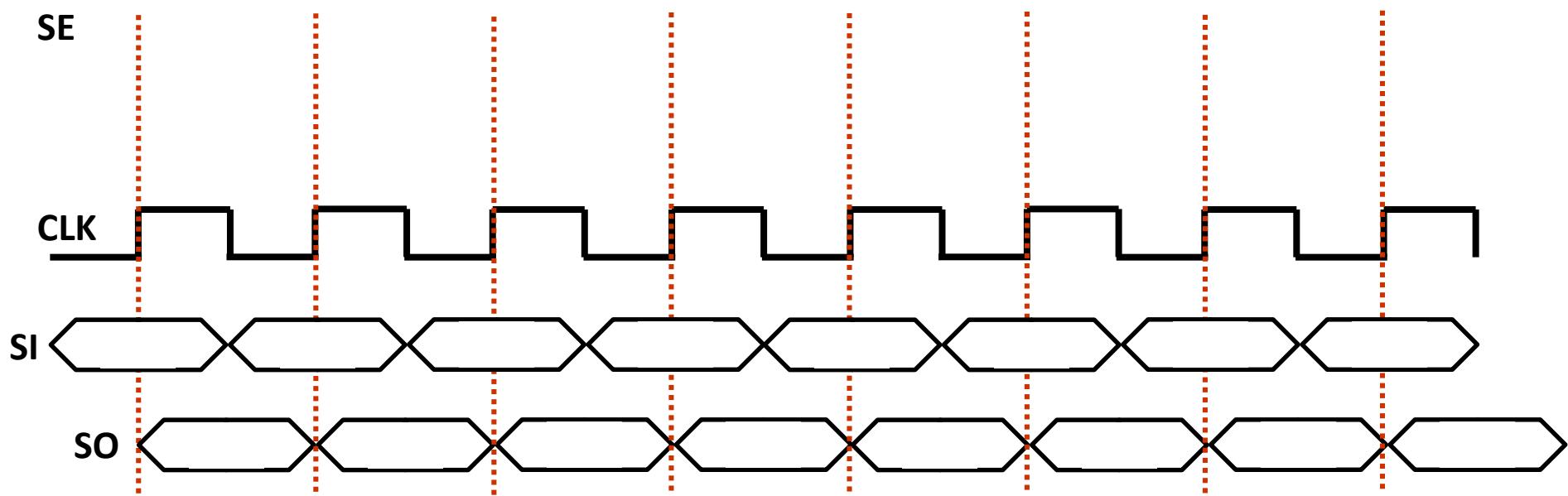
1ο Διανυσμα εισοδου: **a1, a2, a3**

απόκριση: **w1, w2, w3**

ΜΗΚΟΣ ΑΛΥΣΙΔΑΣ = 3

2ο Διανυσμα εισοδου: **b1, b2, b3**

Δείξτε τις τιμές στα SI, SO, σχεδιάστε το SE, δείξτε το διαθέσιμο χρόνο A που έχει η λογική για να αποκριθεί



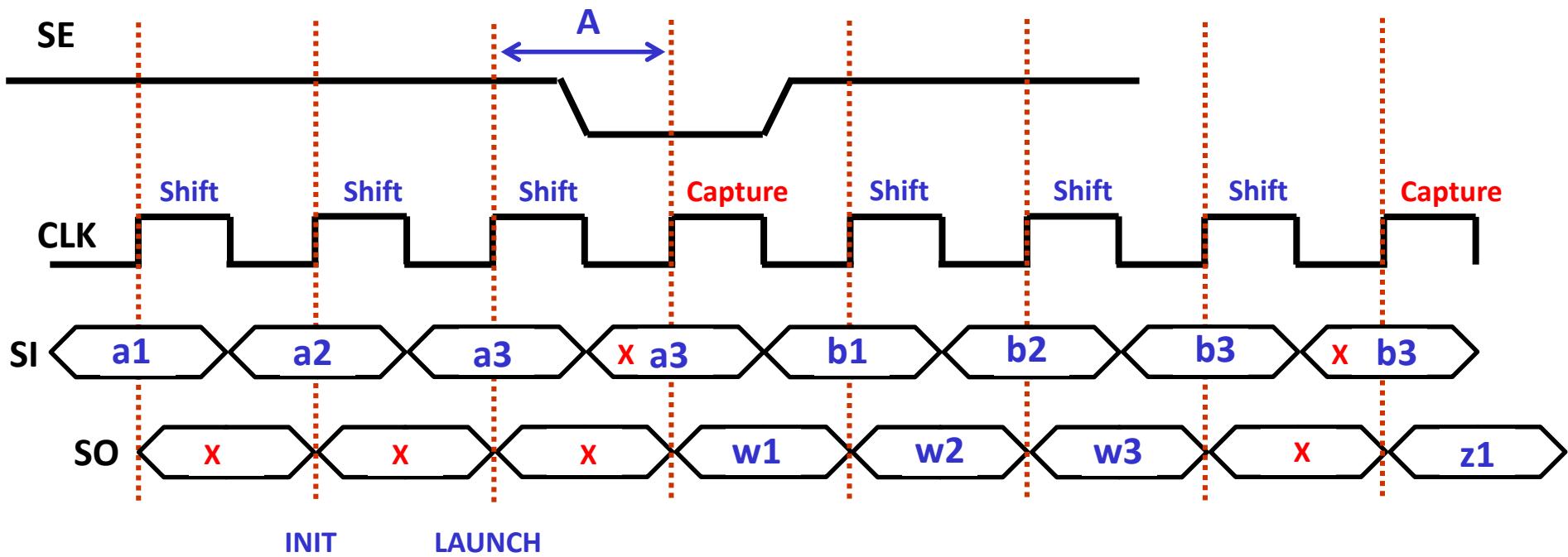
Άσκηση Σειριακής Σάρωσης

1ο Διανυσμα εισοδου: a_1, a_2, a_3 απόκριση: w_1, w_2, w_3

2ο Διανυσμα εισοδου: b_1, b_2, b_3 απόκριση: z_1, z_2, z_3

Δείξτε τις τιμές στα SI, SO, σχεδιάστε το SE, δείξτε το διαθέσιμο χρόνο A που έχει η λογική για να αποκριθεί

Το μήκος της αλυσίδας είναι 3
Τα $a_1 \dots b_1 \dots, w_1 \dots$ μπορεί να είναι διανύσματα (αν έχουμε πολλές αλυσίδες που φορτώνουν παράλληλα) ή απλά bit



X: ΑΔΙΑΦΟΡΗ ΤΙΜΗ

Scan Testing

Α: διαθέσιμος χρόνος που έχει το Συνδ. κύκλωμα για να αποκριθεί

CMOS Integrated Circuit Design Techniques

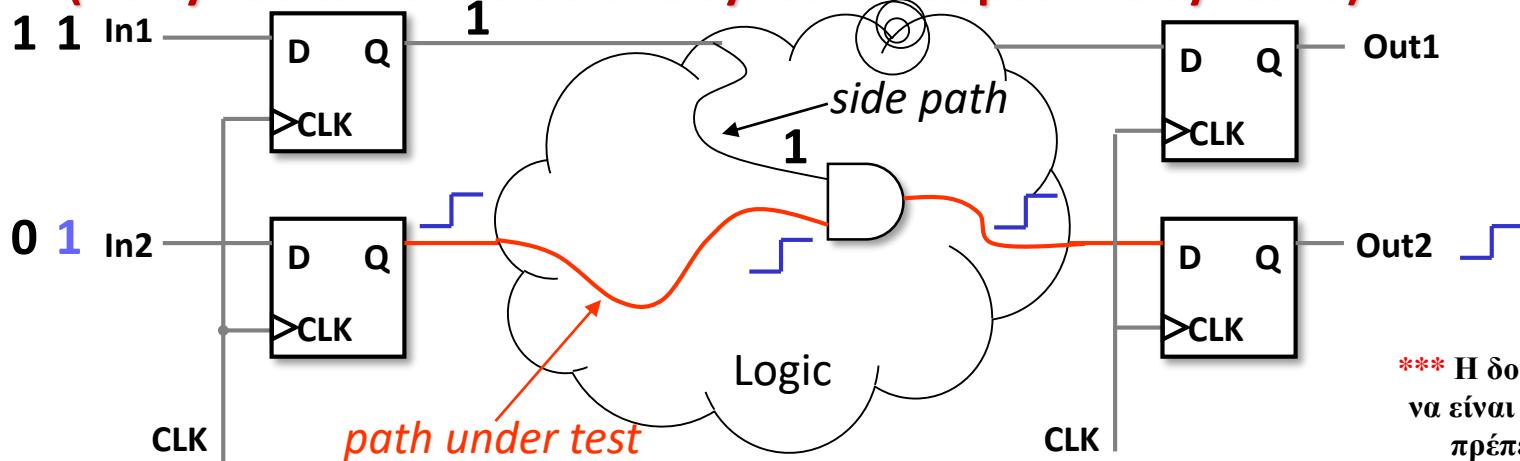
Overview



VLSI Systems
and Computer Architecture Lab

1. *Scan testing: design and application*
2. *At speed testing*
3. *The scan-set design technique*
4. *Scan testing power issues*
5. *The scan-hold design technique*
6. *Level sensitive scan design*
7. *Broadcast and Illinois scan design*

Σφάλματα Καθυστέρησης – Σφάλματα Καθυστέρησης Μετάβασης και Σφάλματα Καθυστέρησης Μονοπατιού (Delay Faults – transition delay faults και path delay faults)



*** Η δοκιμή για να είναι πλήρης πρέπει να περιλάμβάνει και τις 2 μεταβάσεις ($0 \rightarrow 1$ KAI $1 \rightarrow 0$)

$V1 = <10>$ ← διάνυσμα αρχικοποίησης (Initializing test vector)
 $V2 = <11>$ ← διάνυσμα ενεργοποίησης (Activation test vector)

Ένα σφάλμα τύπου καθυστέρησης απαιτεί ένα ζεύγος διαδοχικών διανυσμάτων δοκιμής προκειμένου να το ανιχνεύσουμε. Το πρώτο διάνυσμα αρχικοποιεί το κύκλωμα ενώ το δεύτερο ενεργοποιεί τις εξόδους που παρατηρούμε και προκαλεί την μετάβαση (low to high ή high to low).

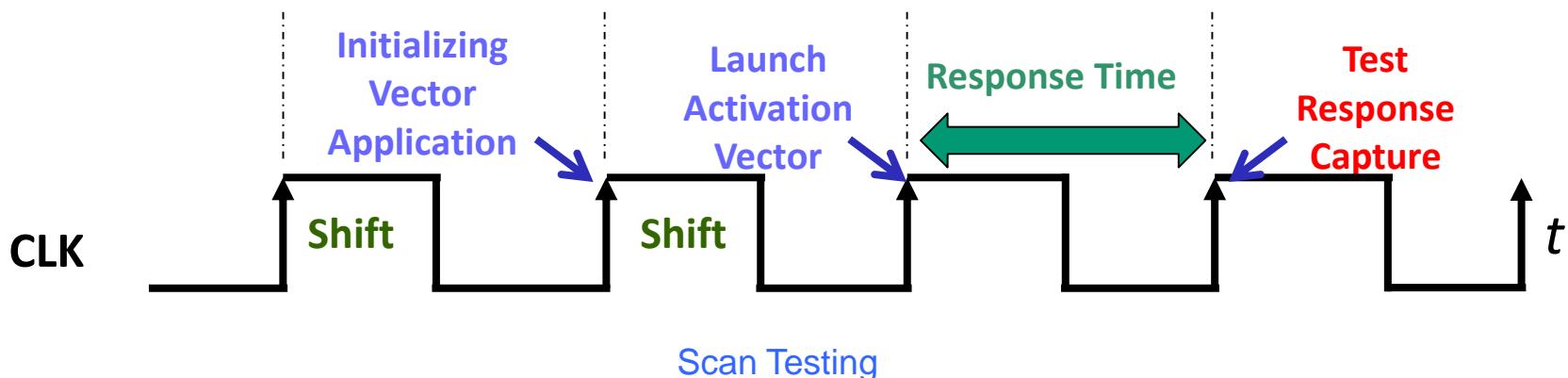
Πως μπορεί η σειριακή σάρωση να αξιοποιηθεί για ανίχνευση σφαλμάτων καθυστέρησης;

Δοκιμή στην Ταχύτητα Λειτουργίας

(At Speed Scan Testing)

Η κλασσική Σειριακή Σάρωση δεν είναι κατάλληλη για σφάλματα καθυστέρησης γιατί:

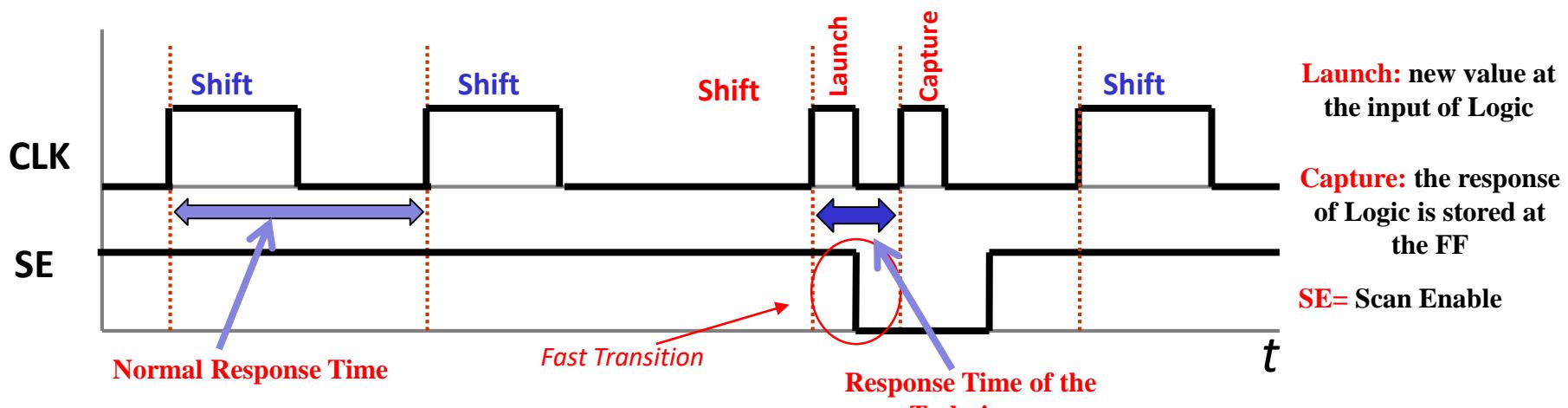
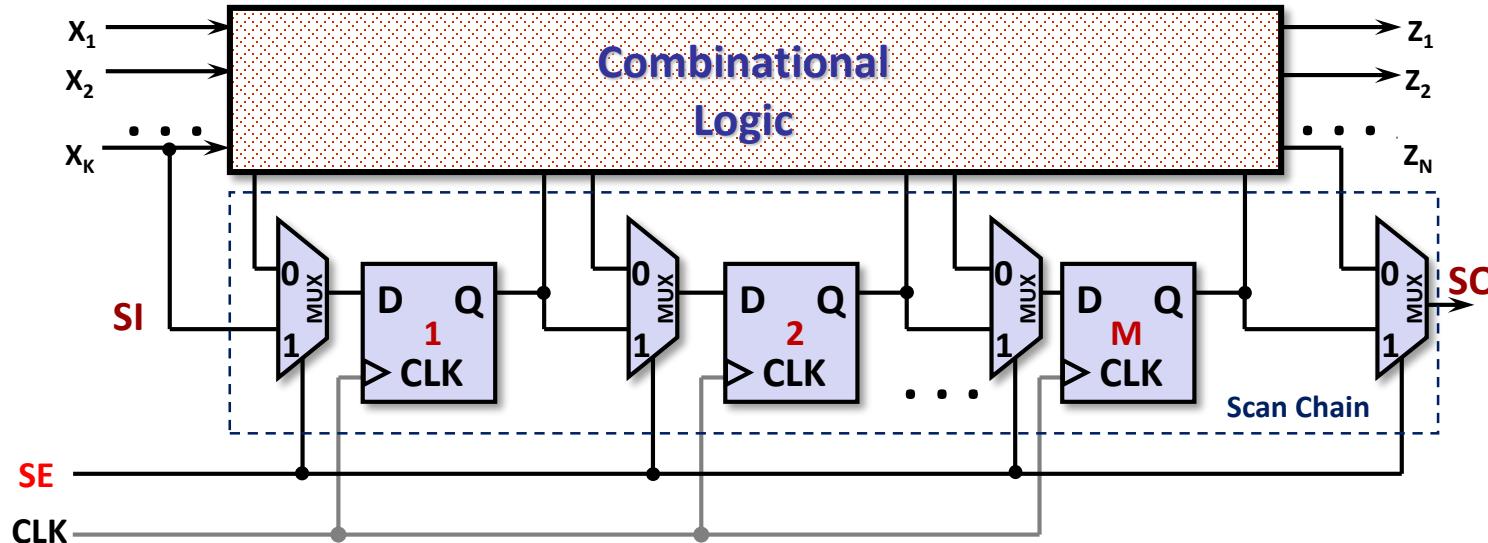
- Δίνει πολύ μεγαλύτερο χρόνο στη λογική του κυκλώματος να αποκριθεί σε σχέση με το χρόνο που αυτό έχει στην κανονική λειτουργία (το ρόλοι που χρησιμοποιούμε για την ολίσθηση είναι αναγκαστικά πιο αργό από το ρόλοι κανονικής λειτουργίας του κυκλώματος λόγω **καθυστέρησης διάδοσης bit κατά την ολίσθηση**).
- Δε μπορούμε να επιλέξουμε οποιοδήποτε ζεύγος διανυσμάτων αρχικοποίησης - ενεργοποίησης (το διάνυσμα αρχικοποίησης είναι το ίδιο με το ενεργοποίησης ολισθημένο κατά 1 bit)
- Χρειαζόμαστε άλλες τεχνικές για δοκιμή της ταχύτητας λειτουργίας.



Δοκιμή στην Ταχύτητα Λειτουργίας (I)

(At Speed Scan Testing)

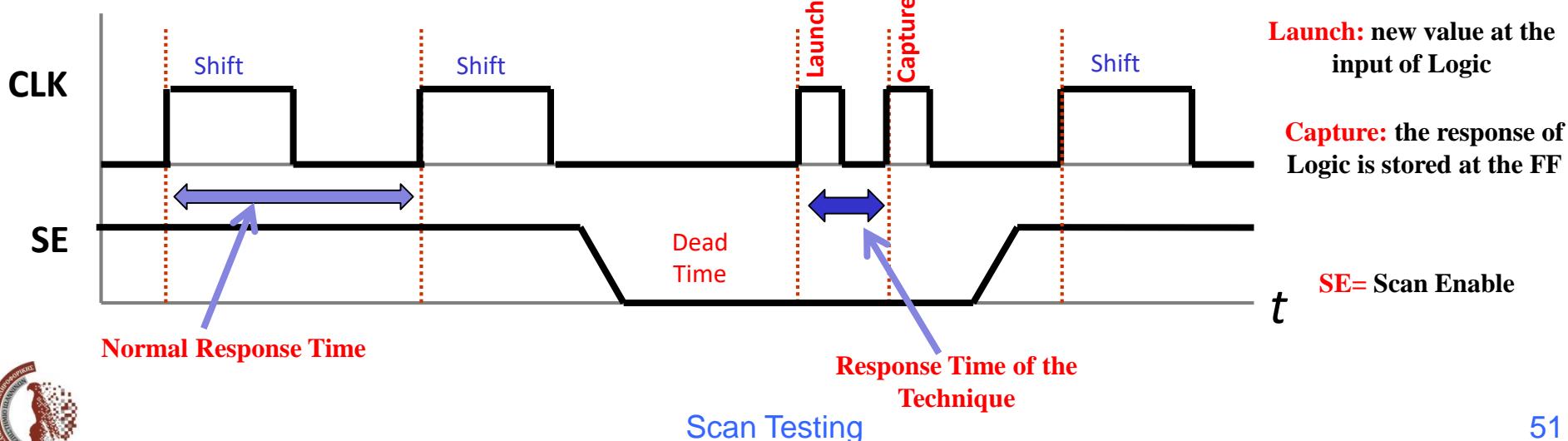
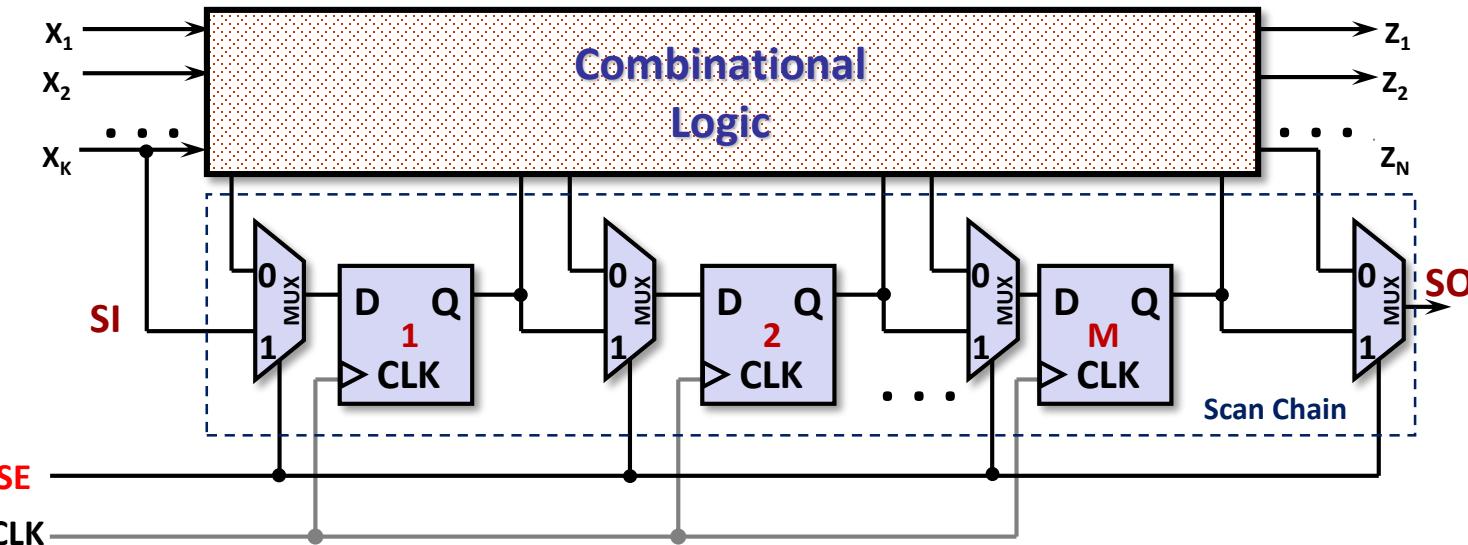
Skewed-load or Launch-on-shift Technique (pg 311)



Δοκιμή στην Ταχύτητα Λειτουργίας (II)

(At Speed Scan Testing)

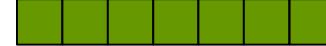
Double Capture or Launch-on-Capture Technique (pg 315)



Δοκιμή στην Ταχύτητα Λειτουργίας (III) 0 0 1 1 0 1 k (At Speed Scan Testing) 0 0 0 1 1 0 1

0 0 0 1 1 0 1 k

ΠΑΡΑΤΗΡΗΣΕΙΣ



- Στην **Skewed-load (Launch-on-shift)** τεχνική το διάνυσμα αρχικοποίησης είναι το τρέχον διάνυσμα ελέγχου ολισθημένο κατά μία θέση αριστερά, ενώ το διάνυσμα ενεργοποίησης είναι το τρέχον διάνυσμα ελέγχου. Στην ουσία εκτελώ το τελευταίο shift και το capture πιο γρήγορα.
- Στην **Double Capture (Launch-on-Capture)** το διάνυσμα αρχικοποίησης είναι το τρέχον διάνυσμα ελέγχου ενώ το διάνυσμα ενεργοποίησης είναι η απόκριση της λογικής στο τρέχον διάνυσμα ελέγχου.
- Και στις 2 περιπτώσεις υπάρχει μία εξάρτηση μεταξύ των 2 διανυσμάτων. Συνεπώς, δεν έχουμε τη δυνατότητα να επιλέξουμε ακριβώς το ζεύγος διανύσματων αρχικοποίησης – ενεργοποίησης που επιθυμούμε.
- Επιπλέον η **Skewed-load (Launch-on-shift)** έχει και τα ακόλουθα μειονεκτήματα: α) απαιτεί τέλειο συγχρονισμό των σημάτων CLK/SE β) απαιτεί πολύ απότομη μετάβαση για το SE (ισχυρό buffering)

0 0 0 1 1
1 1 0 1 0

Δοκιμή στην Ταχύτητα Λειτουργίας (III)

(At Speed Scan Testing)

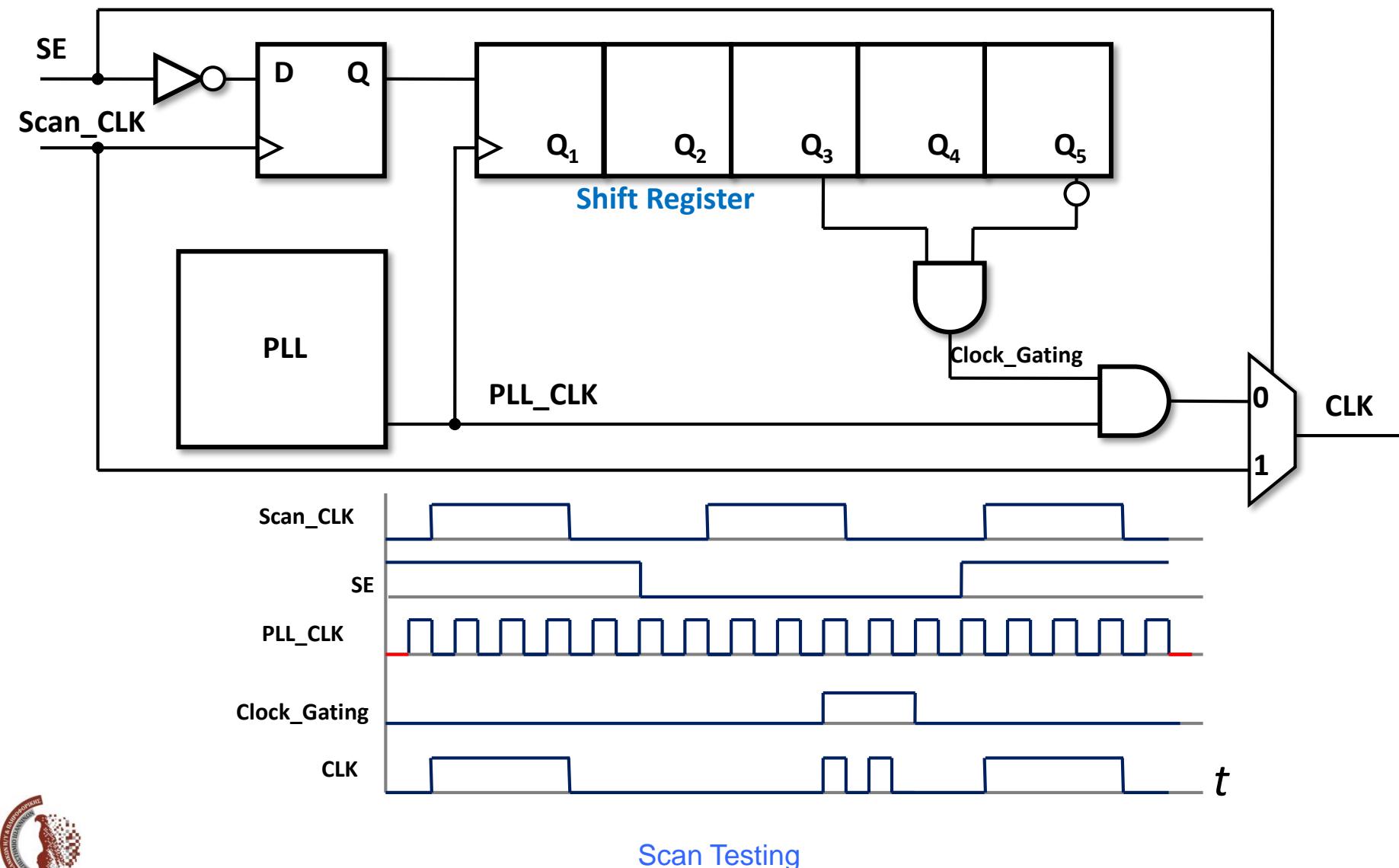
0 0 1 1 0 1 k
0 0 0 1 1 0 1

ΠΑΡΑΤΗΡΗΣΕΙΣ

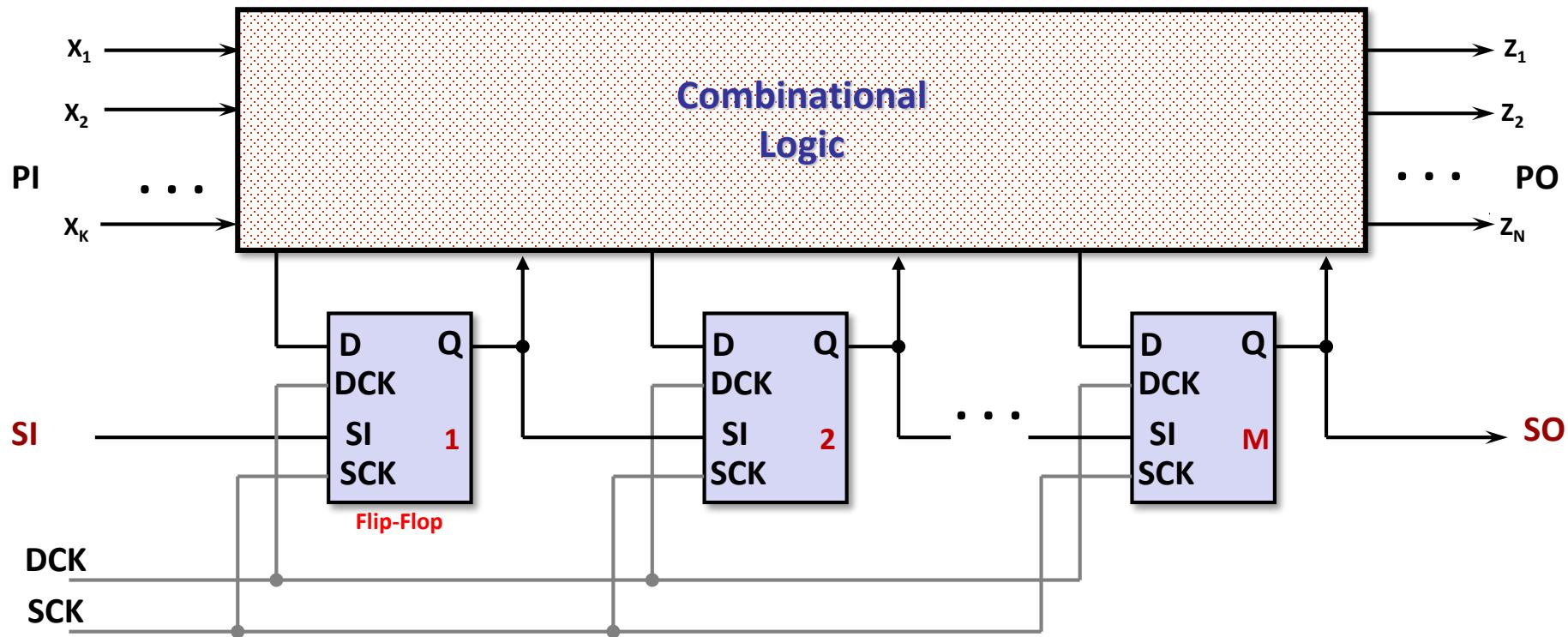


- Στην **Skewed-load (Launch-on-shift)** τεχνική το διάνυσμα αρχικοποίησης είναι το τρέχον διάνυσμα ελέγχου ολισθημένο κατά μία θέση αριστερά, ενώ το διάνυσμα ενεργοποίησης είναι το τρέχον διάνυσμα ελέγχου. Στην ουσία εκτελώ το τελευταίο shift και το capture πιο γρήγορα.
- Στην **Double Capture (Launch-on-Capture)** το διάνυσμα αρχικοποίησης είναι το τρέχον διάνυσμα ελέγχου ενώ το διάνυσμα ενεργοποίησης είναι η απόκριση της λογικής στο τρέχον διάνυσμα ελέγχου.
- Και στις 2 περιπτώσεις υπάρχει μία εξάρτηση μεταξύ των 2 διανυσμάτων. Συνεπώς, δεν έχουμε τη δυνατότητα να επιλέξουμε ακριβώς το ζεύγος διανύσματων αρχικοποίησης – ενεργοποίησης που επιθυμούμε.
- Επιπλέον η **Skewed-load (Launch-on-shift)** έχει και τα ακόλουθα μειονεκτήματα: α) απαιτεί τέλειο συγχρονισμό των σημάτων CLK/SE β) απαιτεί πολύ απότομη μετάβαση για το SE (ισχυρό buffering)

Fast Clock Pulses Generation



The Clocked-Scan Technique



Αντί του **Muxed-D Scan Cell** χρησιμοποιείται το **Clocked-Scan Cell**, το οποίο έχει 2 ρολόγια: **Data Clock (DCK)** και **Scan Clock (SCK)**. Επίσης, ΔΕΝ έχει σήμα SE.

Πλεονέκτημα: Δεν προσθέτει καθυστέρηση στην κανονική λειτουργία.

Μειονέκτημα είναι η απαίτηση υπάρξης δεύτερου ρολογιού γιατί συνεπάγεται κόστος σε υλικό (καλωδίωση – buffers)

The Scan-Set Technique

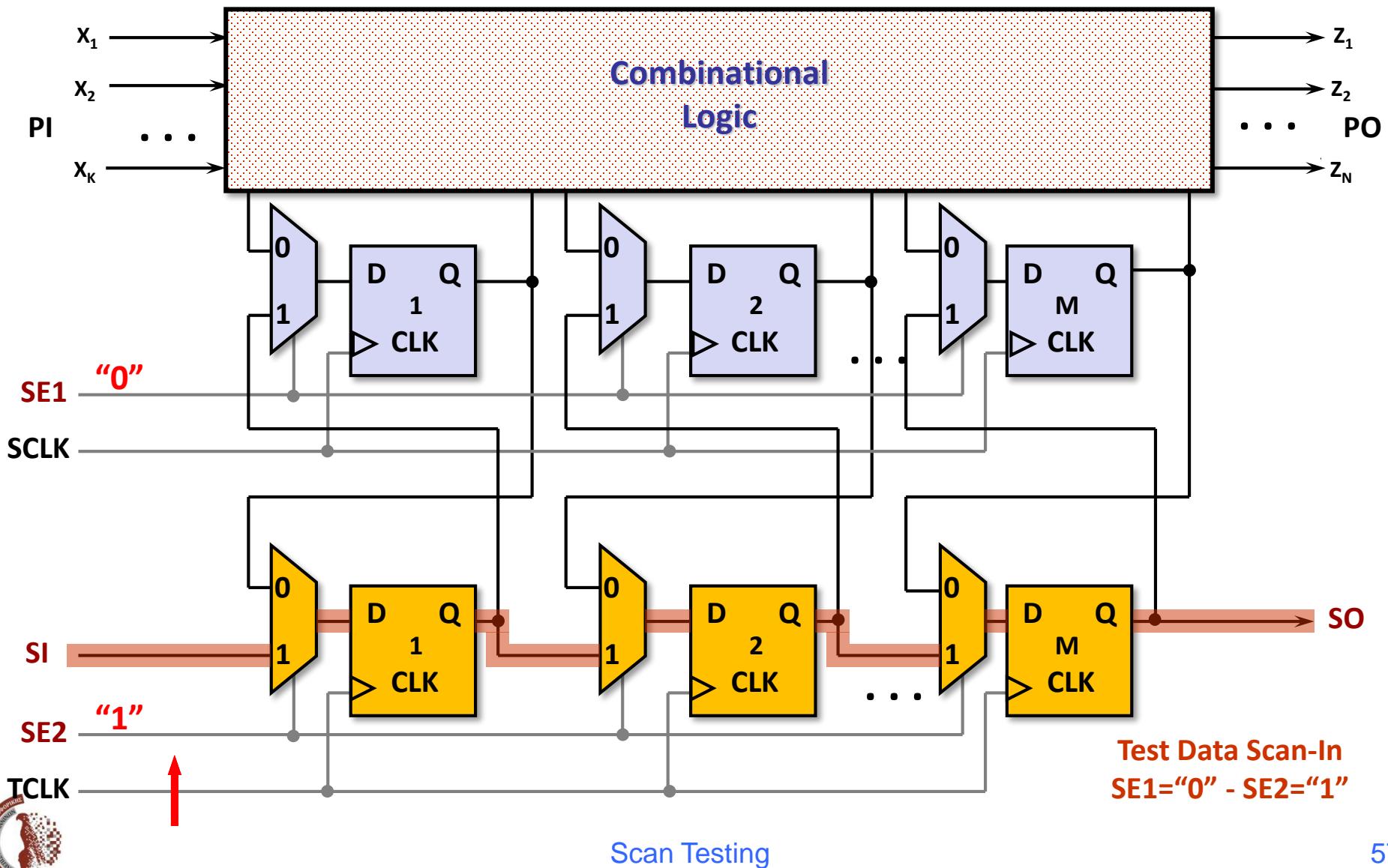
(Snapshot Scan, pg 88)

- Στην τεχνική αυτή προσθέτουμε ένα επιπλέον scan cell για κάθε scan cell του κυκλώματος (βλ επόμενη διαφάνεια). Αυτά τα επιπλέον scan cells δημιουργούν μία **αλυσίδα σάρωσης (scan chain)** και συνδέονται με τα scan cells του κυκλώματος (τα οποία δε σχηματίζουν αλυσίδα)

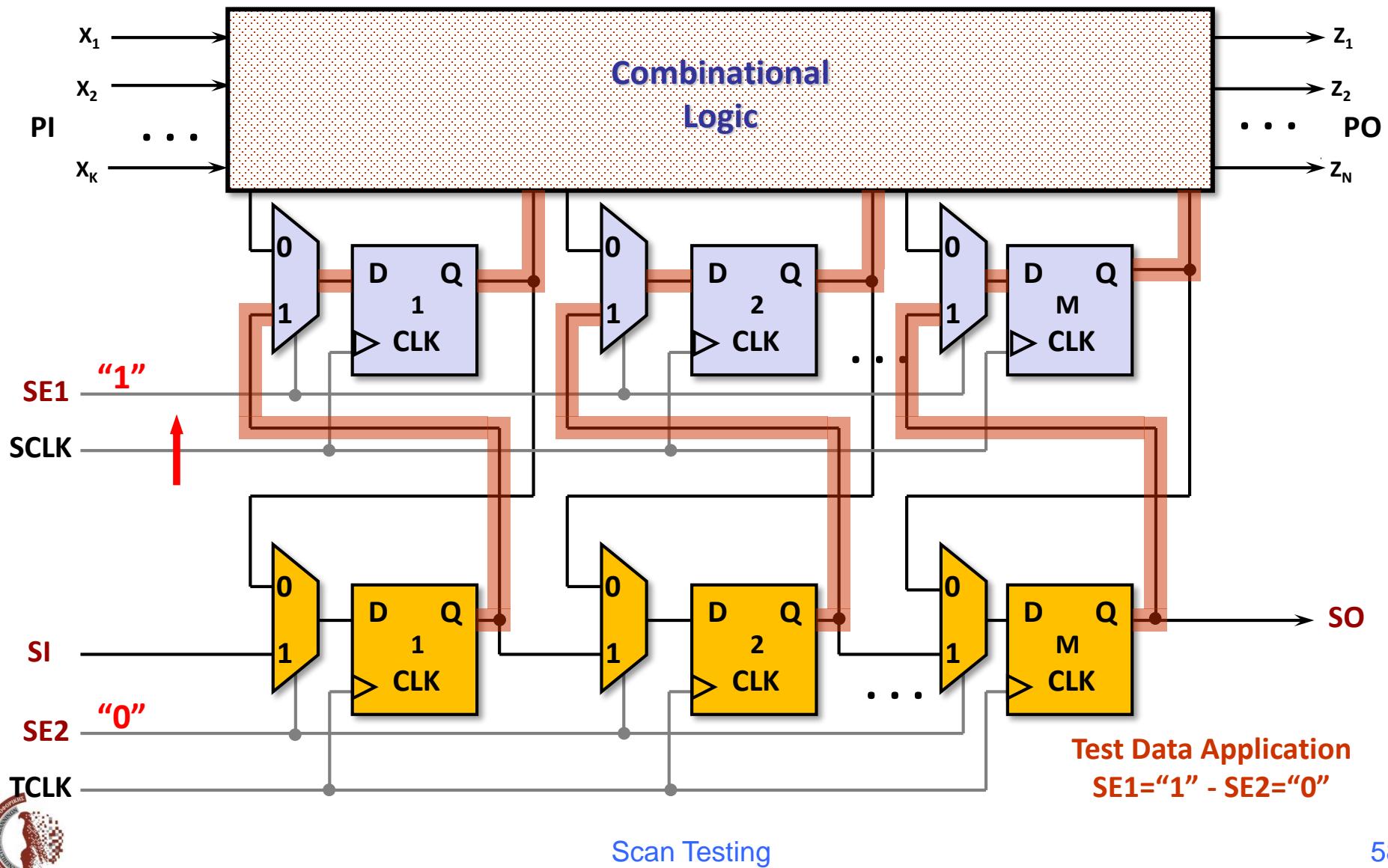
Αυτή η τεχνική μας δίνει 2 δυνατότητες

- Μπορούμε να ‘φωτογραφίσουμε’ και να βγάλουμε στην έξοδο ένα στιγμιότυπο των αποκρίσεων του κυκλώματος χωρίς να διακόψουμε την κανονική του λειτουργία.
- Μπορούμε να φορτώσουμε παράλληλα ένα διάνυσμα δοκιμής (από την δεύτερη αλυσίδα σάρωσης στην κύρια) και επίσης παράλληλα να ‘καταγράψουμε’ τις αποκρίσεις στη δεύτερη αλυσίδα σάρωσης, η οποία στη συνέχεια με ολίσθηση θα βγάλει τις αποκρίσεις έξω. Με αυτό τον τρόπο η κανονική λειτουργία του κυκλώματος διακόπτεται μόνο για λίγους (~3-4) κύκλους ρολογιού.

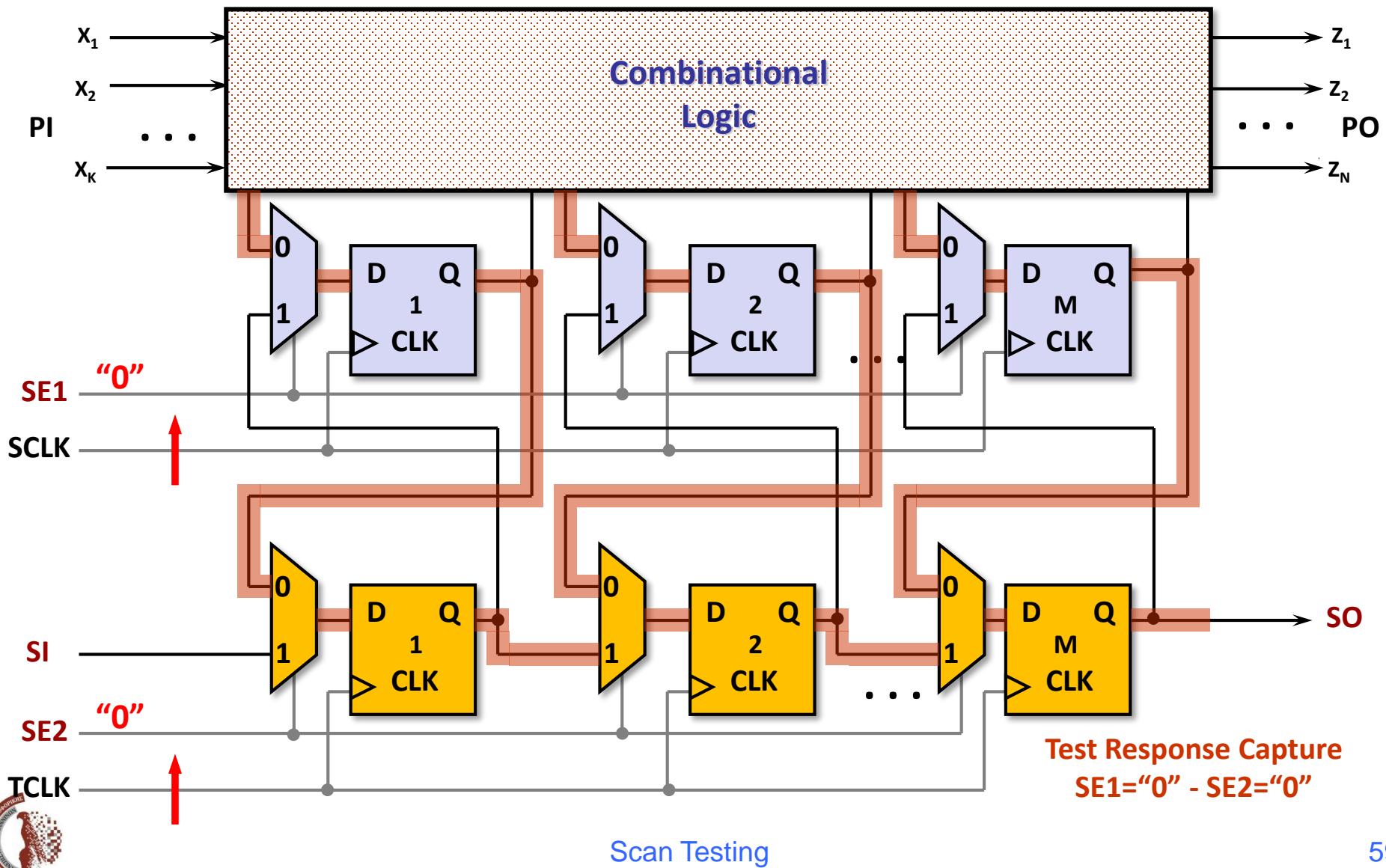
The Scan-Set Technique (I)



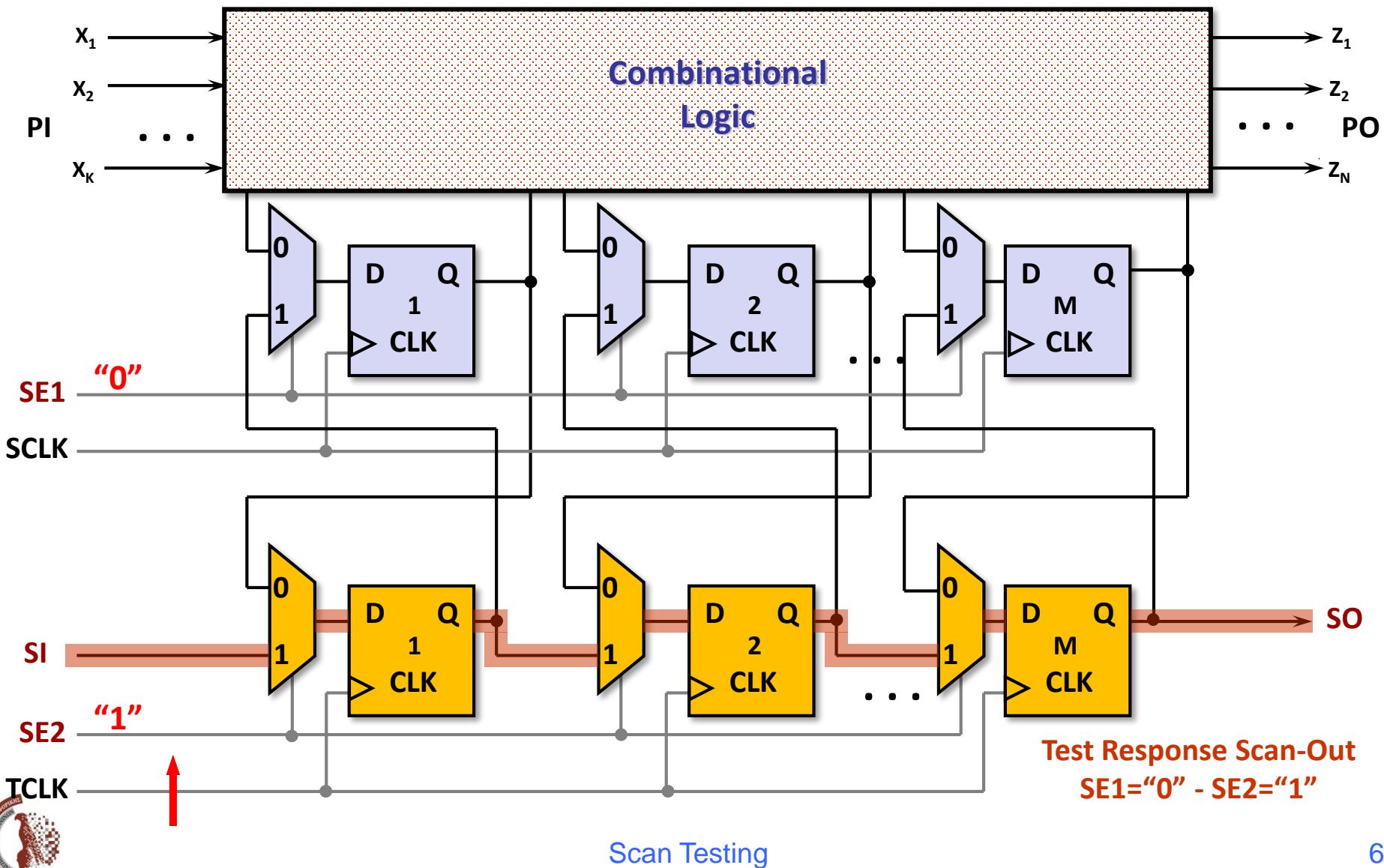
The Scan-Set Technique (II)



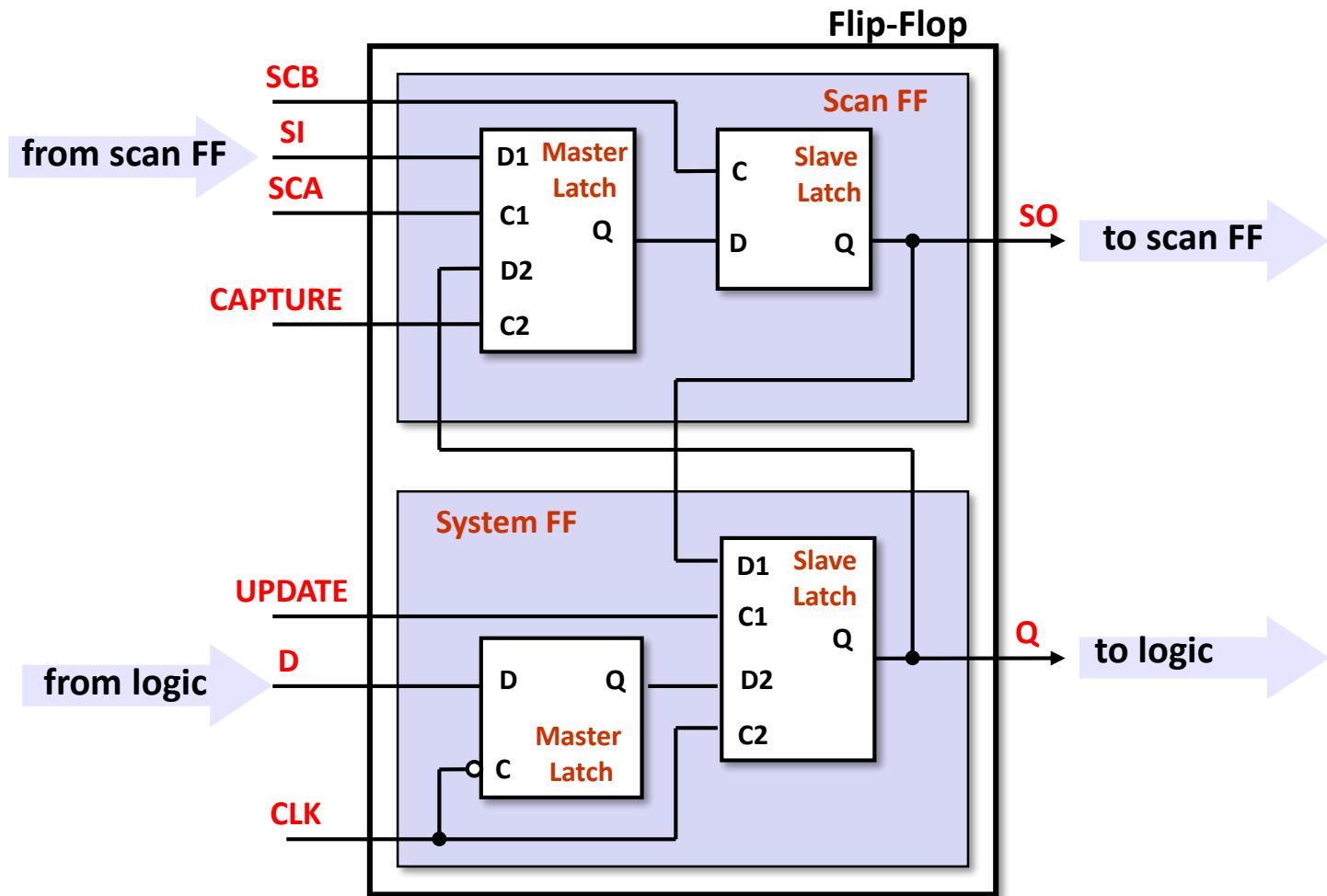
The Scan-Set Technique (III)



The Scan-Set Technique (IV)



Dual Flip-Flops Scan Architecture



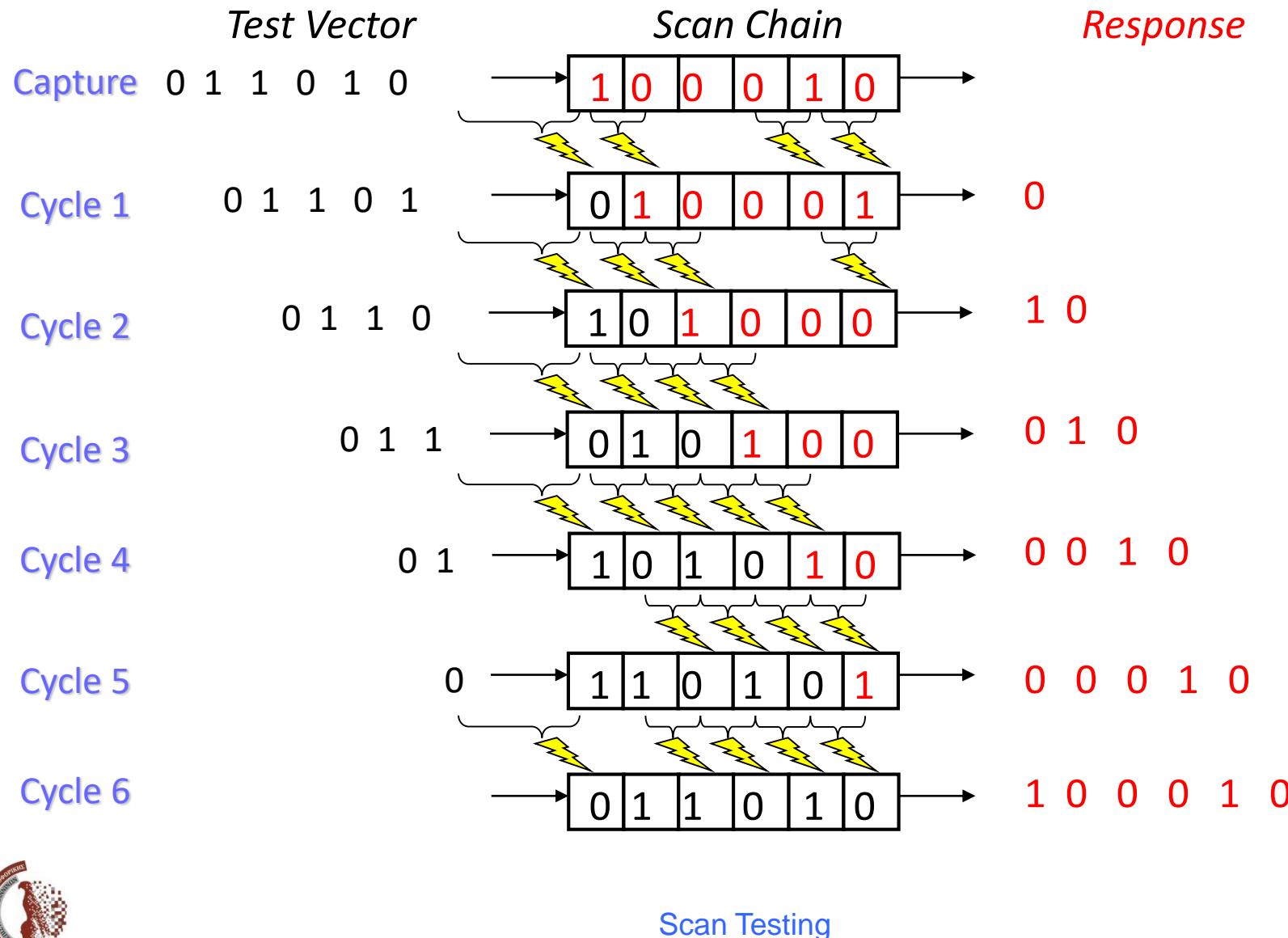
Intel

Επιπτώσεις της Σειριακής Σάρωσης

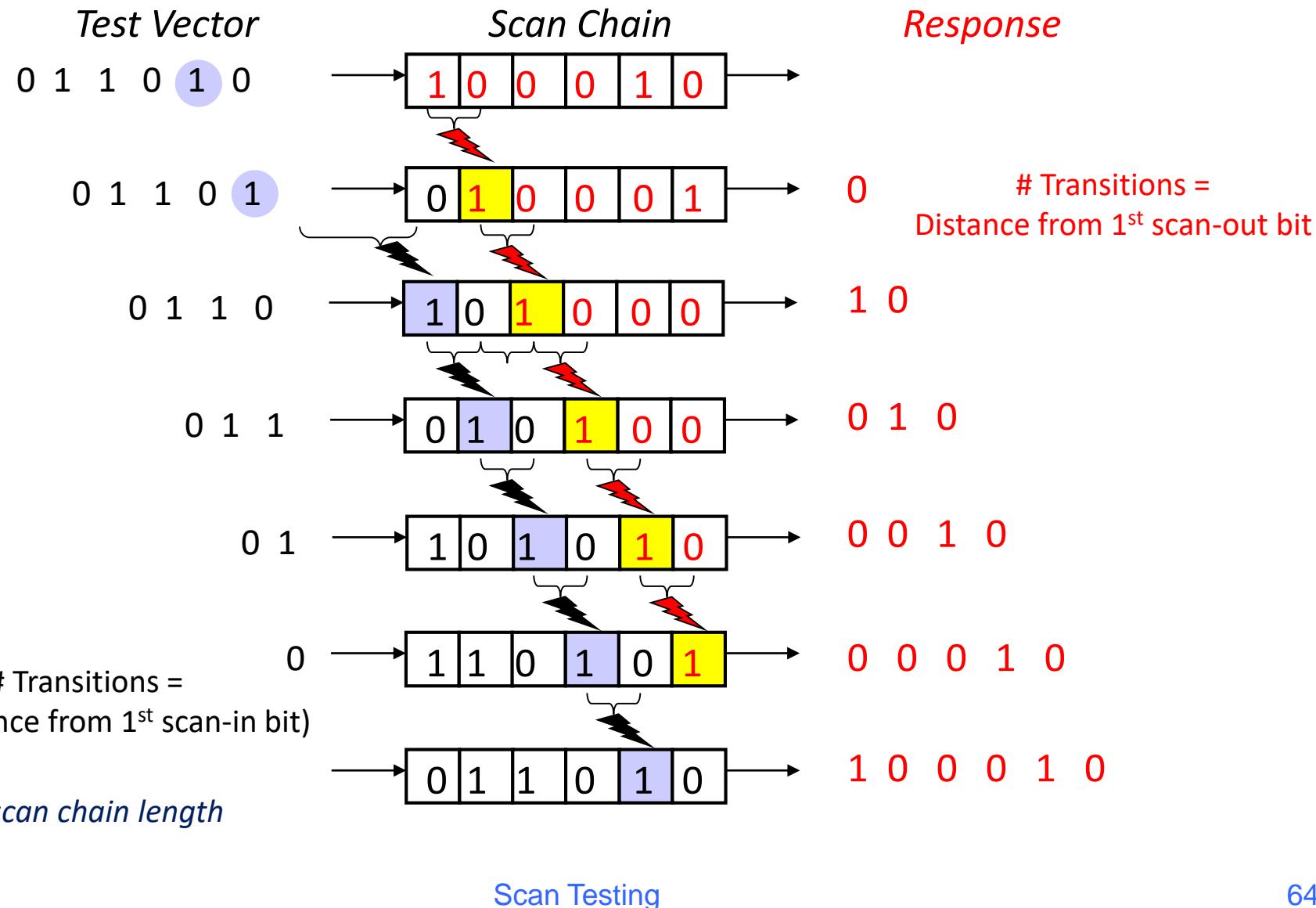
- Κόστος σε επιφάνεια πυριτίου και αριθμό ακροδεκτών.
- Κόστος στην ταχύτητα λειτουργίας του κυκλώματος.
- Κόστος στο χρόνο εφαρμογής της δοκιμής.
- Επιβάρυνση του κυκλώματος με την υπερθέρμανσή του κατά την ολίσθηση δεδομένων στις αλυσίδες σάρωσης (κυρίως λόγω της συνεχούς αλλαγής δεδομένων στις εισόδους της λογικής).



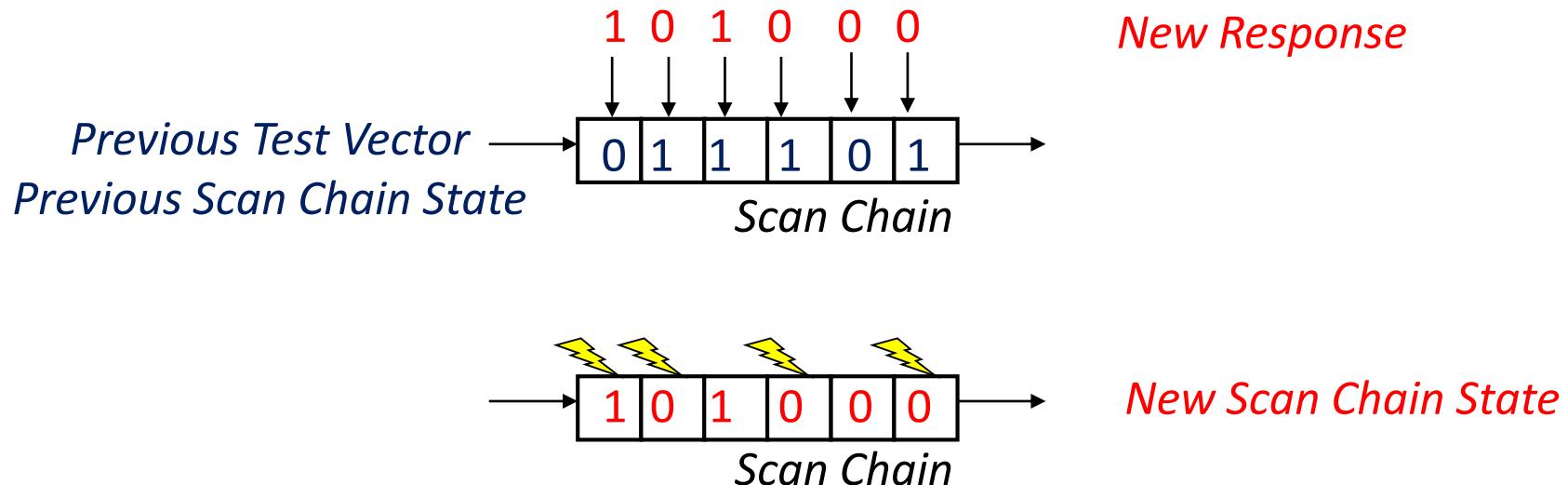
Κατανάλωση ισχύος κατά την ολίσθηση της αλυσίδας σάρωσης (I)



Κατανάλωση ισχύος κατά την ολίσθηση της αλυσίδας σάρωσης (II)



Scan Chain Capture Power Consumption



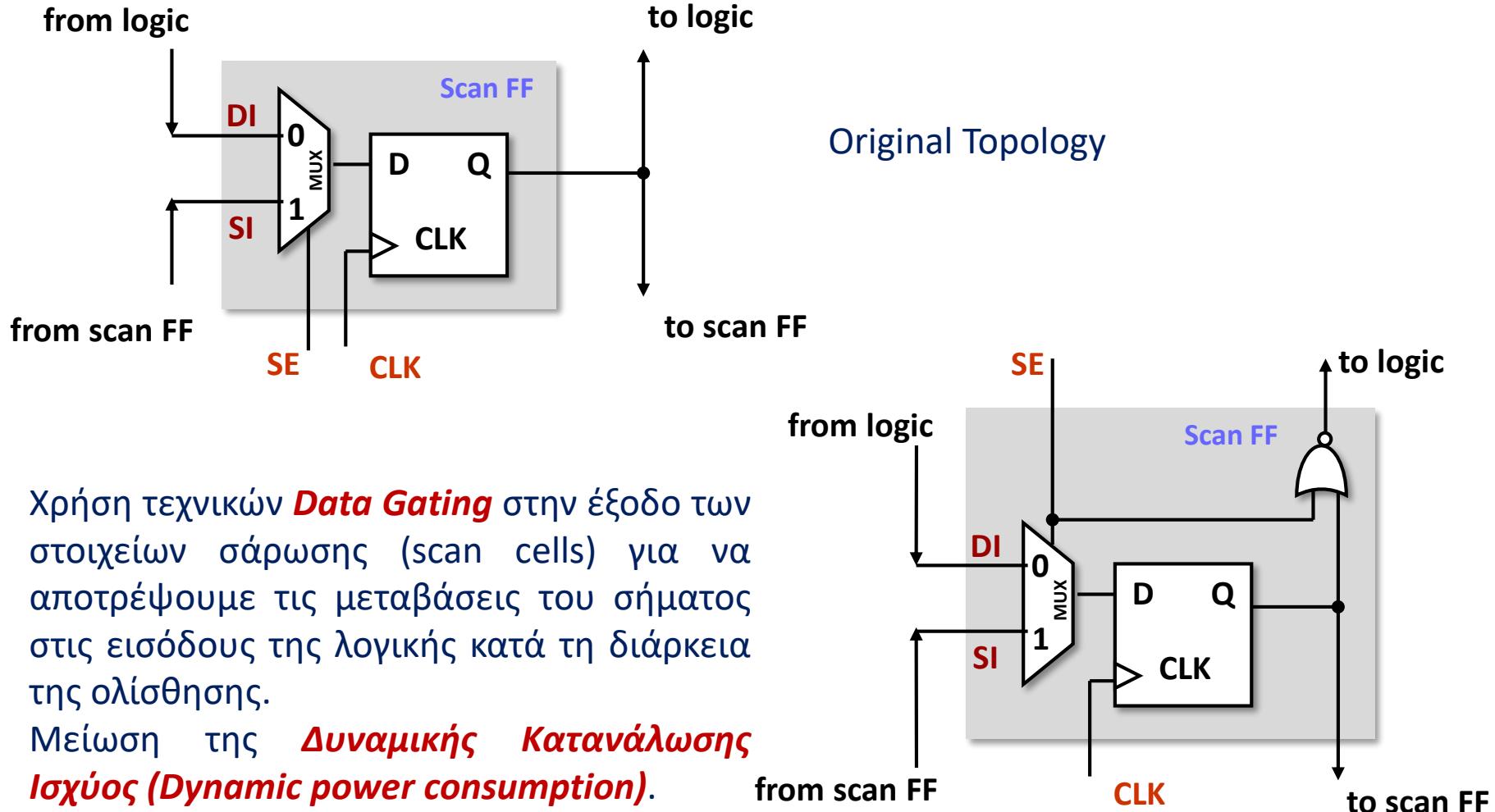
Power consumption during scan testing procedures is a major concern since it can be several times higher than this during the normal mode of operation. This can affect the reliability of the circuit under test (CUT) due to overheat and electromigration phenomena. The excessive switching activity of the CUT during the scan operations may violate the power supply IR and Ldi/dt drop limitations and increase the probability of noise induced test failures. In addition, the elevated temperature can degrade the speed performance of the CUT and result to erroneous test responses that will invalidate the testing process and lead to yield loss.

X-bit Assignment

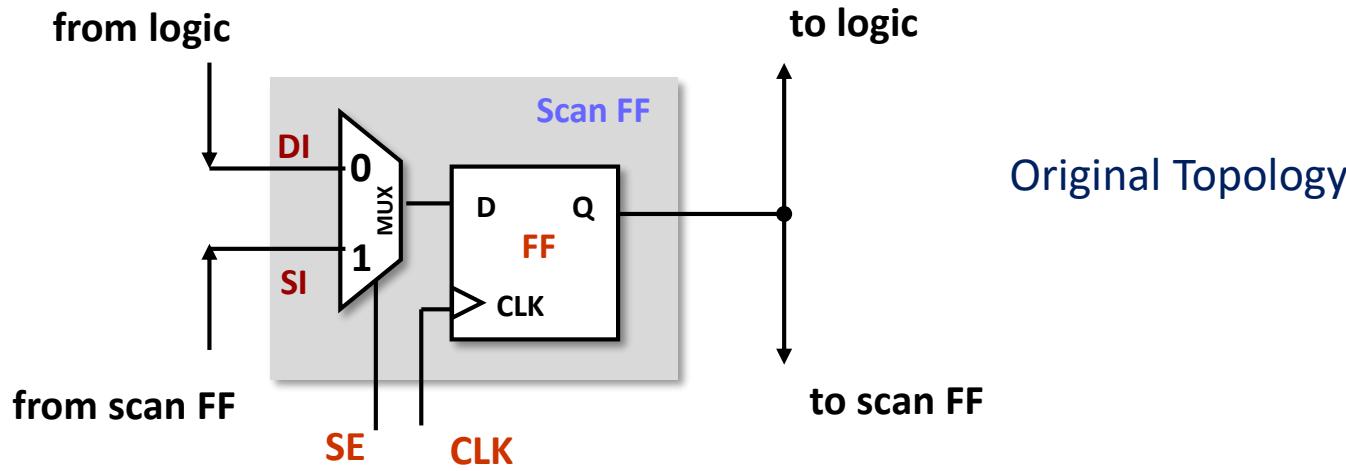
- A large number of bits in a test cube that is generated by an ATPG tool are don't care bits (X-bits).
- In order to apply a test cube for circuit testing, specific values must be assigned to the X-bits (test vector formation). This task is called X-filling.
- The X-filling process can be oriented for shift and/or capture power reduction.



Σαρωση Χαμηλής Κατανάλωσης (Low Power Scan)

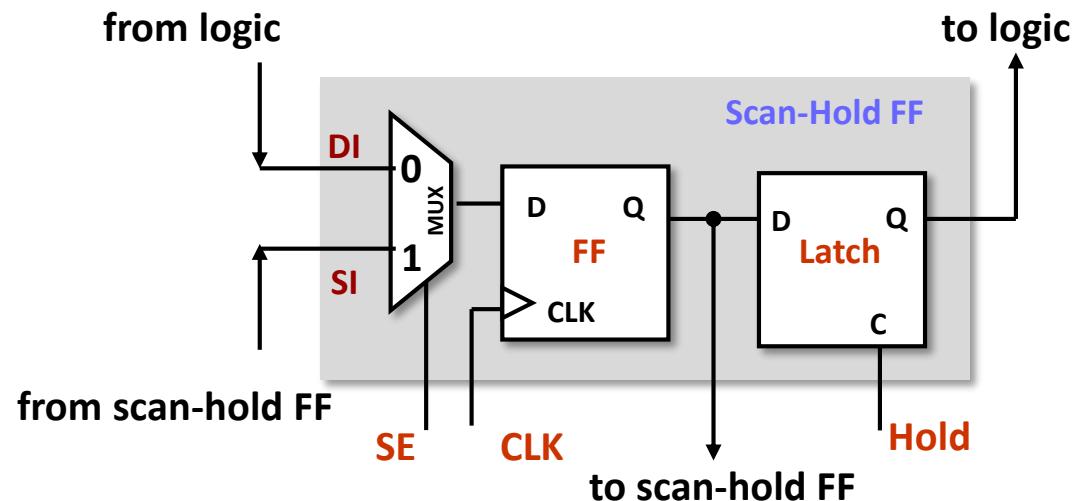


Scan-Hold Flip-Flop



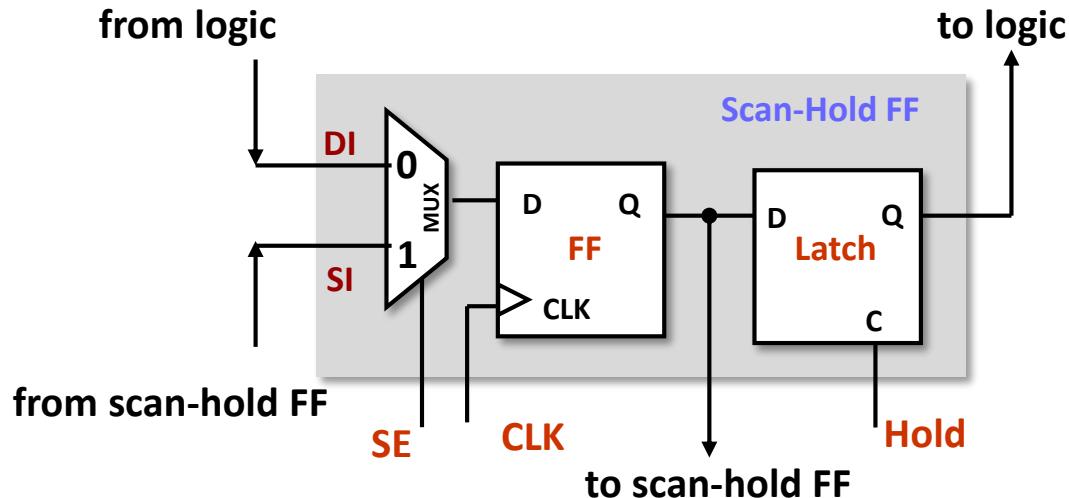
Delay testing oriented Flip-Flop!

+ Dynamic power reduction.



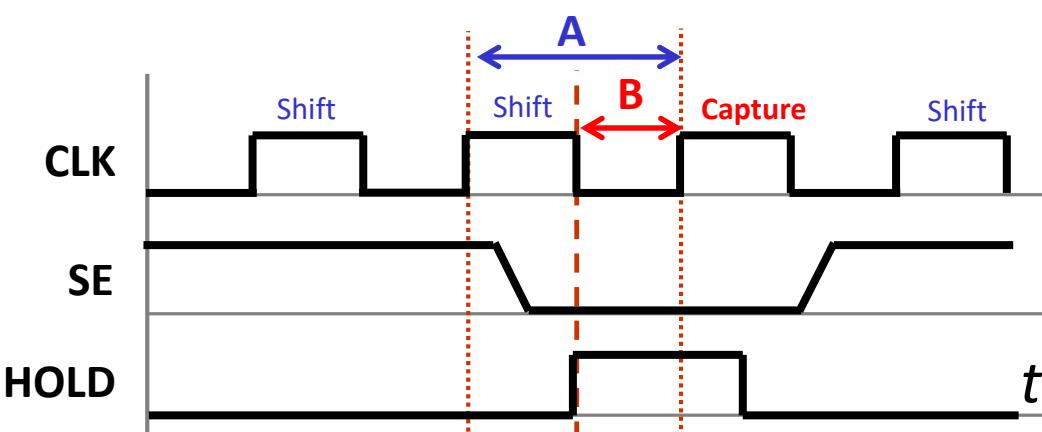
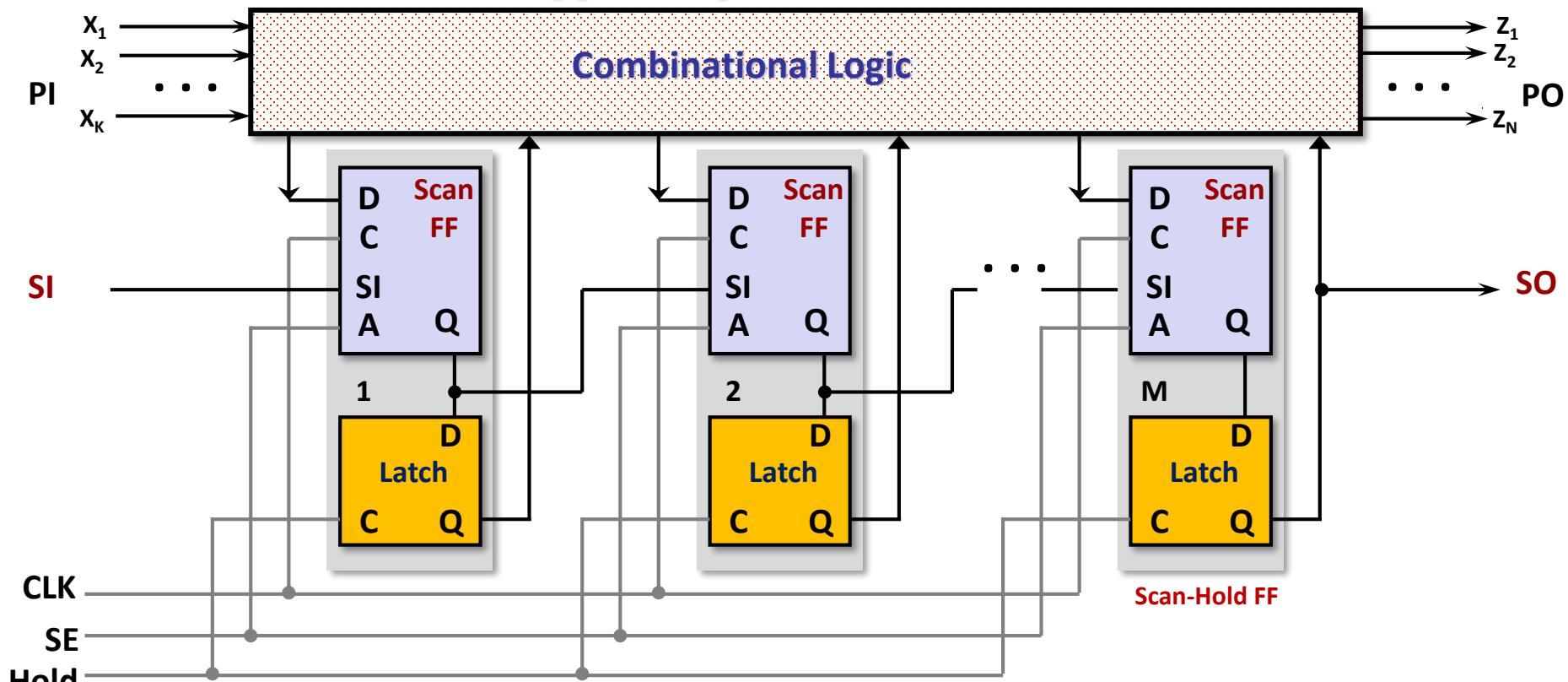
Τεχνική Scan-Hold

(Enhanced Scan pg 88)



- Μπορούμε να εκτελέσουμε έλεγχο σφαλμάτων καθυστέρησης για όλα τα διανύσματα εισόδου, καθορίζοντας πόσο χρόνο πριν τη θετική ακμή του ρολογιού (στην οποία θα γίνει η σύλληψη της απόκρισης) θα είναι διαθέσιμα τα δεδομένα εισόδου.
- Μπορούμε να καθορίσουμε ακριβώς το ζεύγος διανυσμάτων αρχικοποίησης - ενεργοποίησης, το οποίο θα αποτελείται από 2 διαδοχικά διανύσματα ελέγχου.
- Μειώνουμε τη δυναμική κατανάλωση κατά τη διάρκεια της ολίσθησης κρατώντας το σήμα **HOLD** σε κατάσταση low.

Τεχνική Scan-Hold



Διαθέσιμος Χρόνος
απόκρισης Λογικής
A) Χωρις Scan-Hold
B) Με Scan-Hold
(ο χρόνος που φαίνεται είναι ενδεικτικός, είναι ένα παράδειγμα)

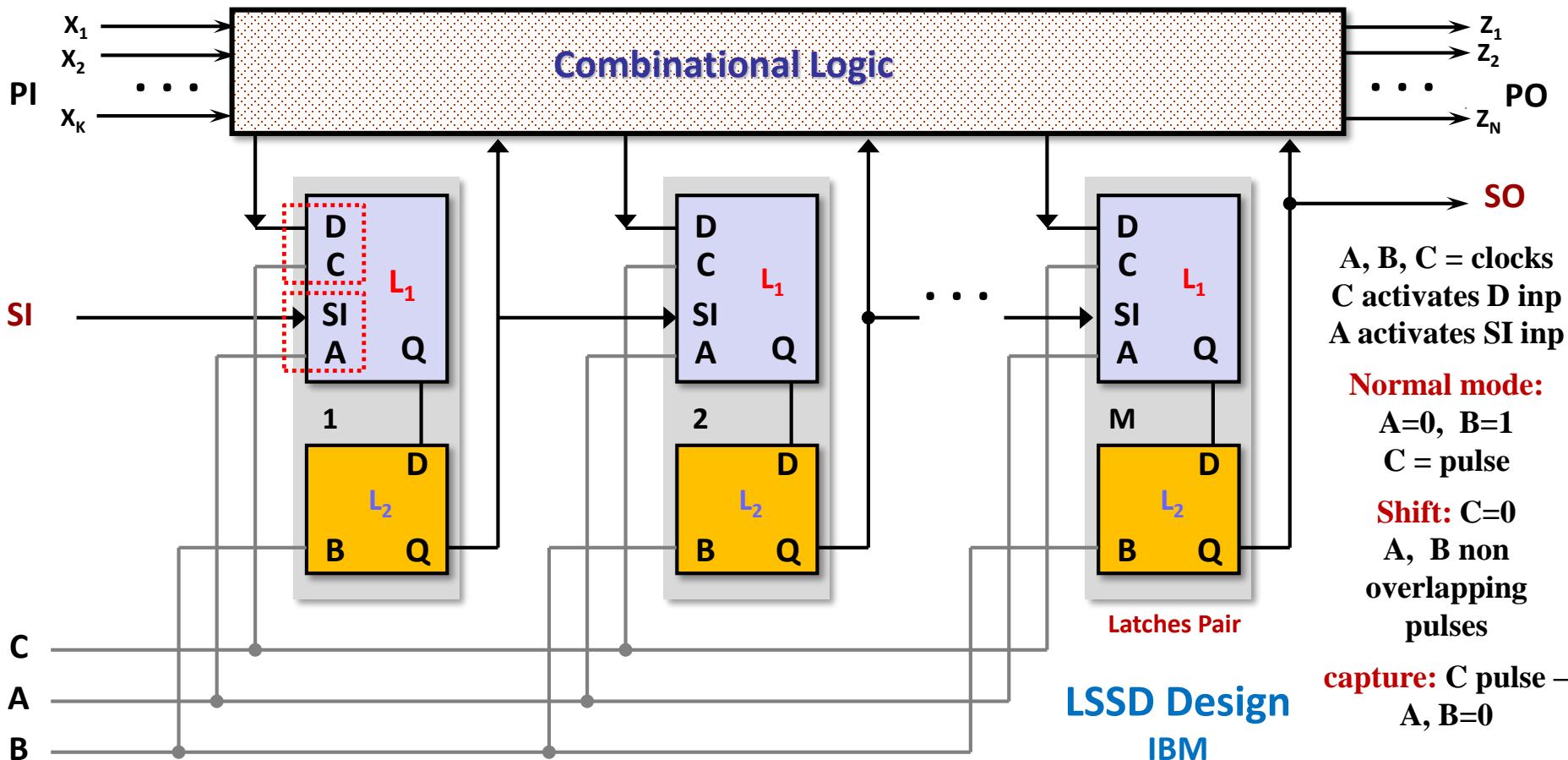
Ανακεφαλαίωση

1. Δομή και λειτουργία αλυσίδων σάρωσης (*Scan testing*)
 1. Παραδείγματα ασκήσεων για *testpatterns generation* και *scanchains insertion*
2. Δοκιμή στην ταχύτητα λειτουργίας (χωρίς αλλαγή στην τοπολογία της αλυσίδας σάρωσης) pg. 311
 - i. Skewed-load or Launch-on-shift Technique
 - ii. Double Capture or Launch-on-Capture Technique
3. Clocked Scan σχεδίαση (δεν επιδρά στην ταχύτητα του κυκλώματος)
4. Scan-Set σχεδίαση (σύλληψη στιγμιοτύπου – *snapshot capture*) pg. 88
5. Προβλήματα κατανάλωσης ισχύος και λύσεις (*Data Gating*)
6. Scan-hold σχεδίαση (χαμηλή κατανάλωση + δοκιμή ταχύτητας λειτουργίας για όλα τα διανύσματα δοκιμής - Enhanced Scan pg 88)



Level Sensitive Scan Design (LSSD) I

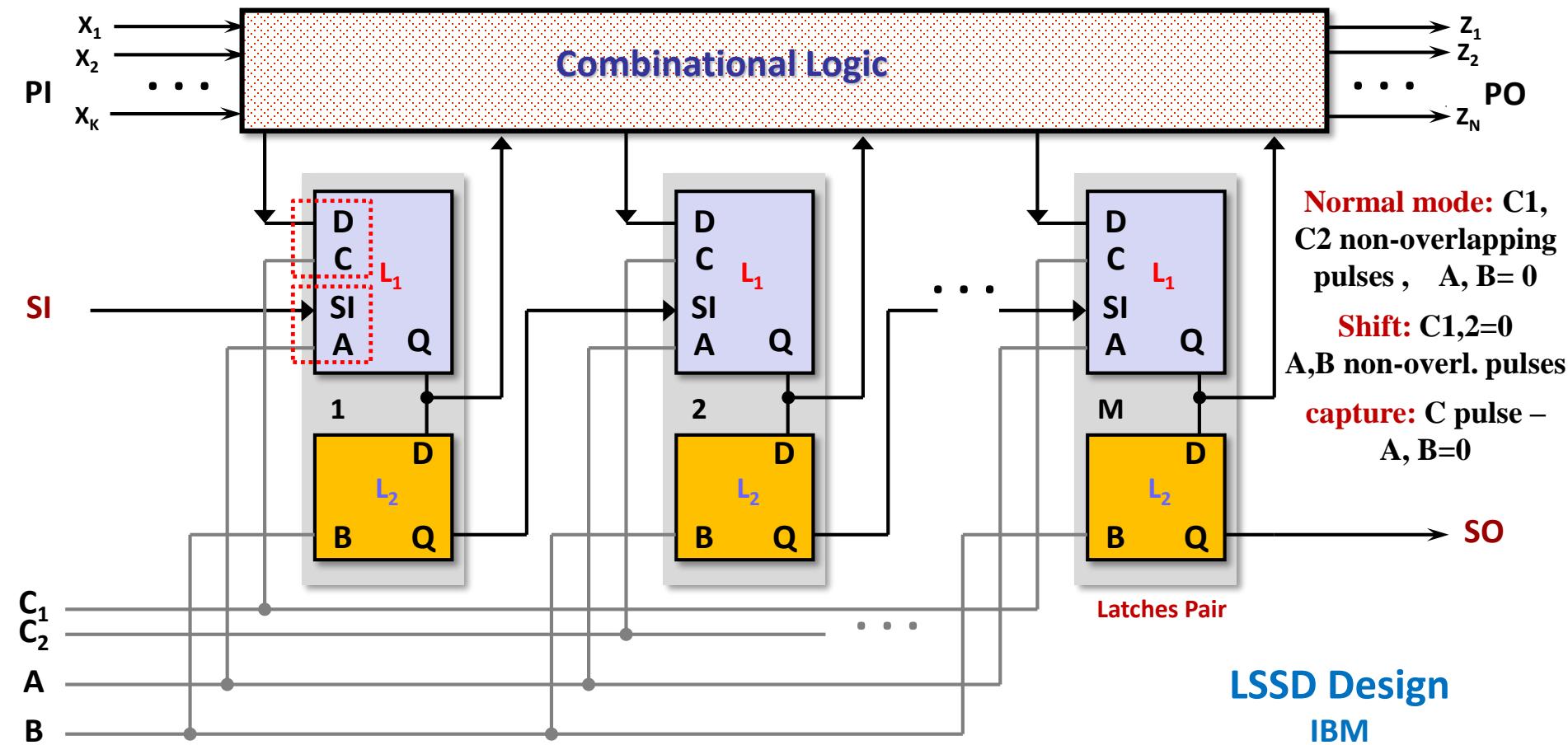
Double Latch Design



Χρησιμοποιείται όταν το αρχικό κύκλωμα έχει LATCH αντί για Flip-Flop

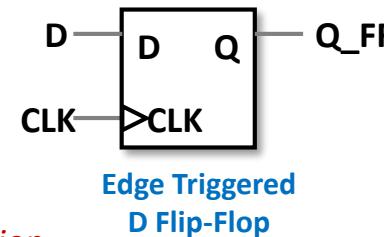
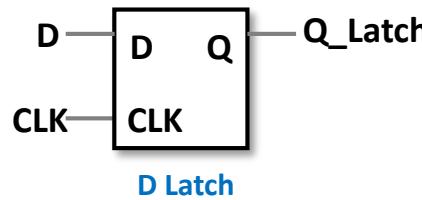
Level Sensitive Scan Design (LSSD) II

Single Latch Design

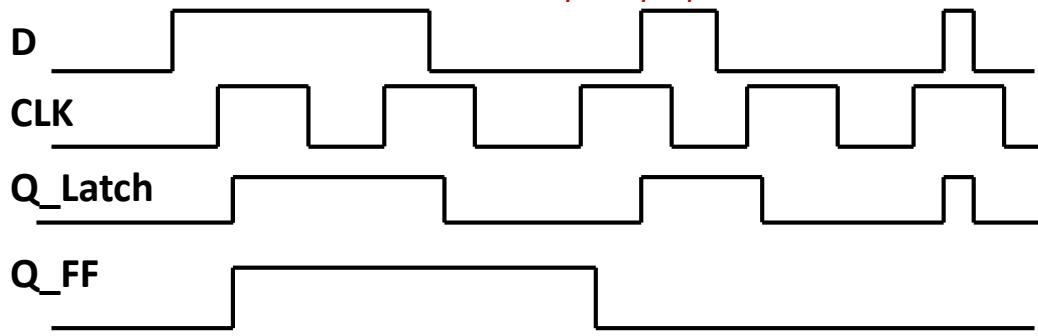


Χρησιμοποιείται όταν το αρχικό κύκλωμα έχει LATCH αντί για Flip-Flop

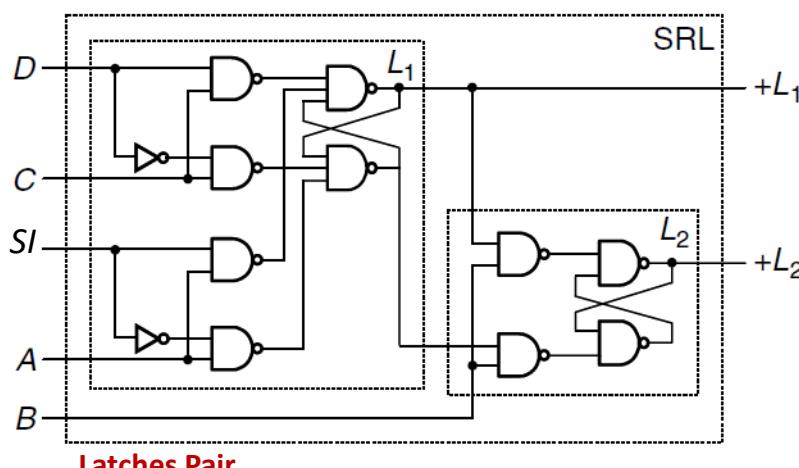
Level Sensitive Scan Design (III)



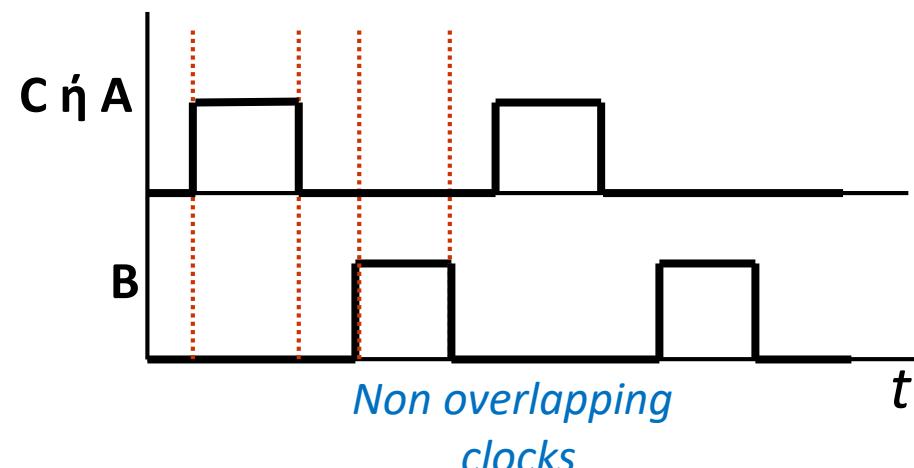
D-Latch and D-Flip-Flop operation



Για να φτιάξω καταχωρητή ολίσθησης με Latch πρέπει πάντα τα διαδοχικά Latch να έχουν διαφορετικά και μή αλληλεπικαλυπτόμενα ρολόγια (Non-overlapping clocks) γιατί αλλιώς δημιουργείται ουσιαστικά συνδυαστικό κύκλωμα.



Scan Testing

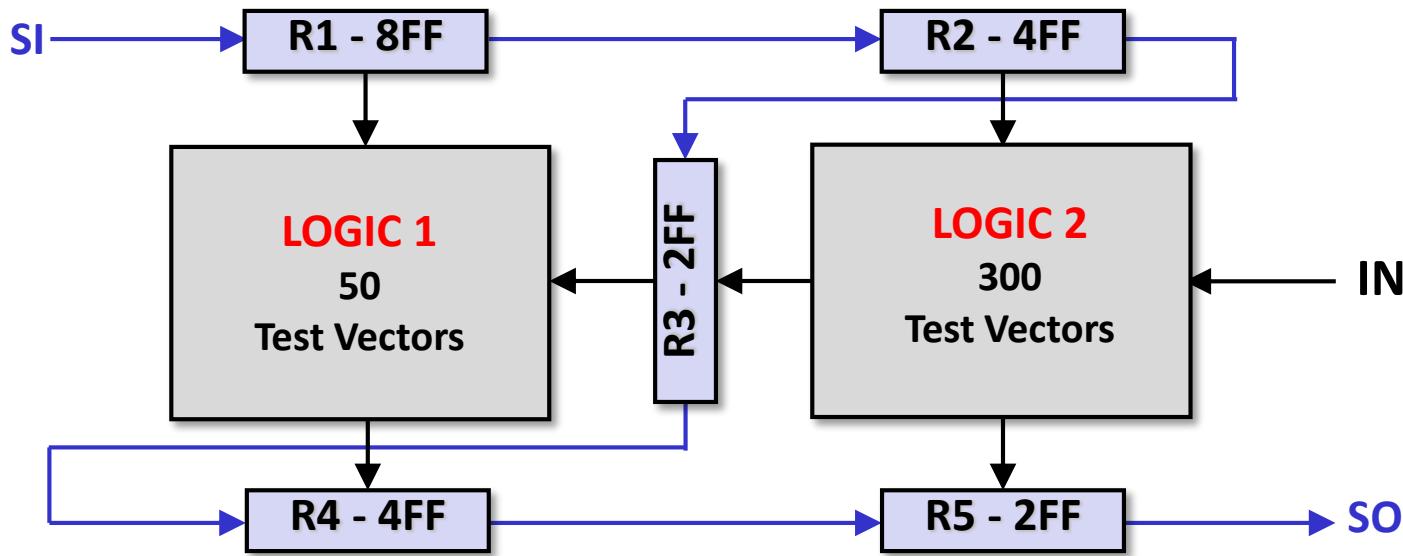


Δυσκολίες στη Σχεδίαση Αλυσίδων Σάρρωσης

- Παρόλο που η αλυσίδα σάρρωσης είναι μια σχετικά απλή και εύκολα κατανοητή ιδέα, η εφαρμογή της στην πράξη κρύβει αρκετές δυσκολίες. Οι δυσκολίες αυτές οφείλονται κυρίως στη χρήση διαφόρων κυκλωματικών στοιχείων και τεχνικών που στοχεύουν στην αύξηση της ταχύτητας και τη μείωση της κατανάλωσης και της επιφάνειας πυριτίου του Ο.Κ. , όπως για παράδειγμα:
 - Tristate Busses / Buffers, Pass Transistors.
 - Bidirectional I/O Ports
 - Gated Clocks
 - Derived Clocks (clock signals generated internally from a storage element, a frequency divider, a pulse generator etc)
 - Combinational Feedback Loops (συνδυαστικά κυκλώματα με βρόχο ανατροφοδότησης)
 - Ασύγχρονα σήματα SET/RESET



Reordering Scan Chain Flip-Flops (I)

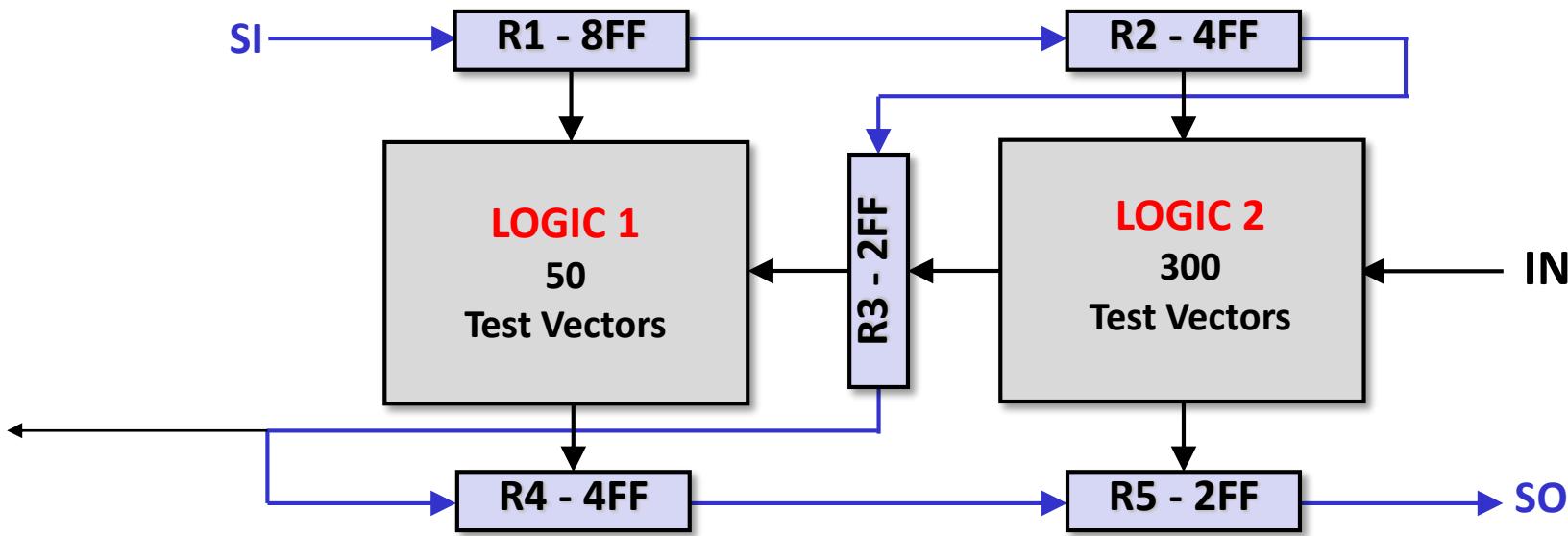


- Υποθέτω ότι **50 Test Vectors** \subset **300 Test Vectors (TV)** και επομένως με τα 300 διανύσματα καλύπτονται και τα 2 υποκυκλώματα λογικής.
- Συνολικό μήκος αλυσίδας $8+4+2+4+2=20$. Για κάθε TV χρειάζονται 20 κύκλοι ρολογιού για ολίσθηση + 1 για σύλληψη (capture). Επιπλέον χρειάζονται άλλοι 20 κύκλοι για την ολίσθηση του τελικού διανύσματος απόκρισης.

Test Application: $300(20+1)+20 = 6320$ clock cycles

“Random” register connection

Reordering Scan Chain Flip-Flops (II)



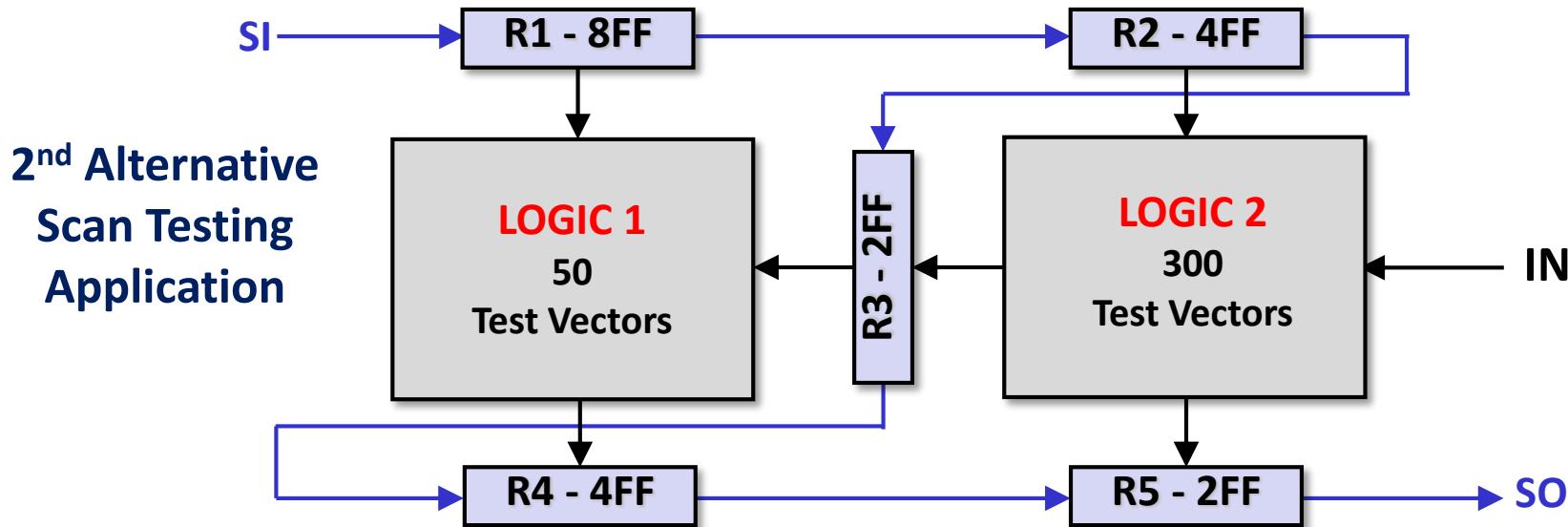
1^η Βελτίωση: λαμβάνω υπόψιν input - output

- Παρατηρώ ότι μόνο τα πρώτα $8+4+2=14$ FFs δίνουν είσοδο στη λογική. Άρα με αυτή την τοπολογία της αλυσίδας σάρωσης το πραγματικό μέγεθος διανύσματος εισόδου είναι 14. (**IN=14, OUT=8, length = MAX(IN, OUT)**)
- Επιπλέον το τελικό διάνυσμα απόκρισης θέλει $2+4+2=8$ ολισθήσεις για να 'βγει έξω'.

1st Alternative Test Application: $300(14+1)+8 = 4508$ clock cycles

Improvement
29% !

Reordering Scan Chain Flip-Flops (III)



- Αφού εφαρμόσουμε πρώτα τα 50 TV που είναι κοινά και για τα 2 υποκυκλώματα, μένει να εφαρμόσουμε τα υπόλοιπα 250 για το LOGIC 2. Αυτό παίρνει είσοδο από R2 και δίνει έξοδο σε R3 και R5, οπότε το μήκος της αλυσίδας είναι: INPUT = 4, OUTPUT = R3+R4+R5 = 8 (ναι, πρέπει να υποστούμε και το R4 που περιττεύει ...) Επομένως τα υπόλοιπα διανύσματα θέλουν 8 ολισθήσεις (length = MAX(IN, OUT)) εκτός από το 51^ο που θέλει 12 λόγω του R1

$$50(14+1) = 750 \text{ clock cycles}$$

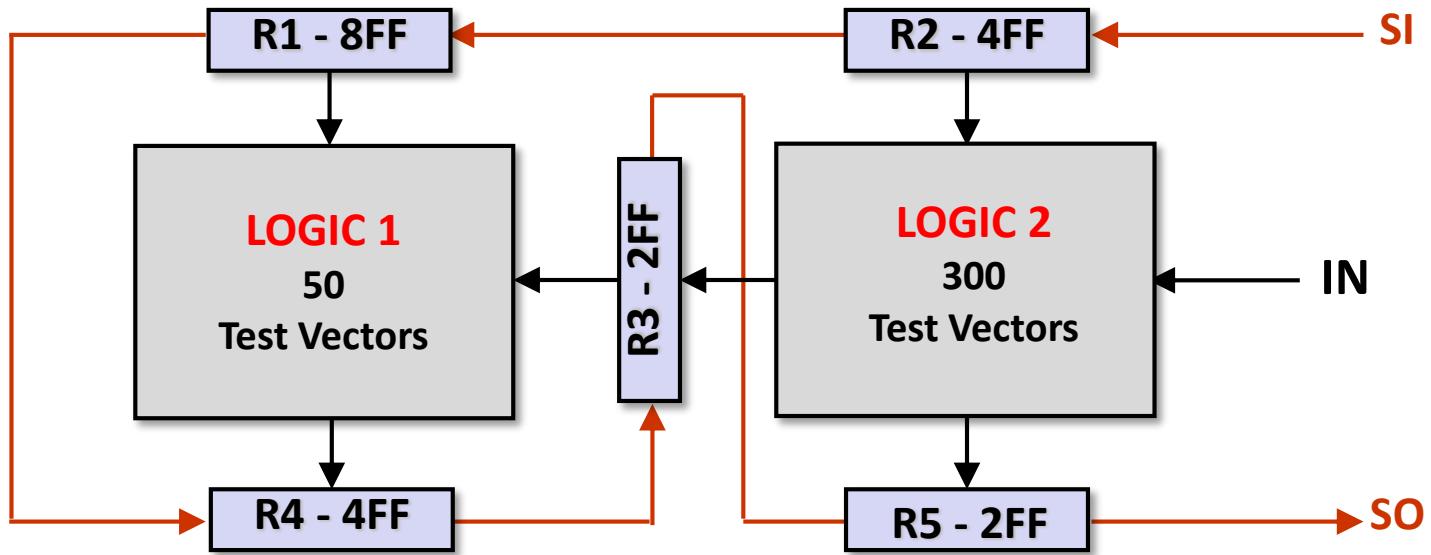
$$12 + 249(8+1) + 8 = 2261 \text{ clock cycles}$$

Improvement
52% !

$$3011 \text{ clock cycles}$$

Reordering Scan Chain Flip-Flops (IV)

3rd Alternative
Scan Testing
Application
with cell
reordering



- Τελική βελτίωση: βάζω τα R3 – R5 συνεχόμενα. Για τα 50 TV που αφορούν όλο το κύκλωμα έχω IN=18, OUT = 8, length = 18. Για τα 250 διανύσματα που αφορούν μόνο το LOGIC 2 το μήκος της αλυσίδας γίνεται 4 (IN=4, OUT=4, length = MAX(IN, OUT)=4). Αυτό ισχύει και για το 51^ο διάνυσμα γιατί το R2 είναι πλέον στην αρχή της αλυσίδας.

In addition, test power reduction is achieved!

$$50(18+1) = 950 \text{ clock cycles}$$

$$250(4+1)+4 = 1254 \text{ clock cycles}$$

$$2204 \text{ clock cycles}$$

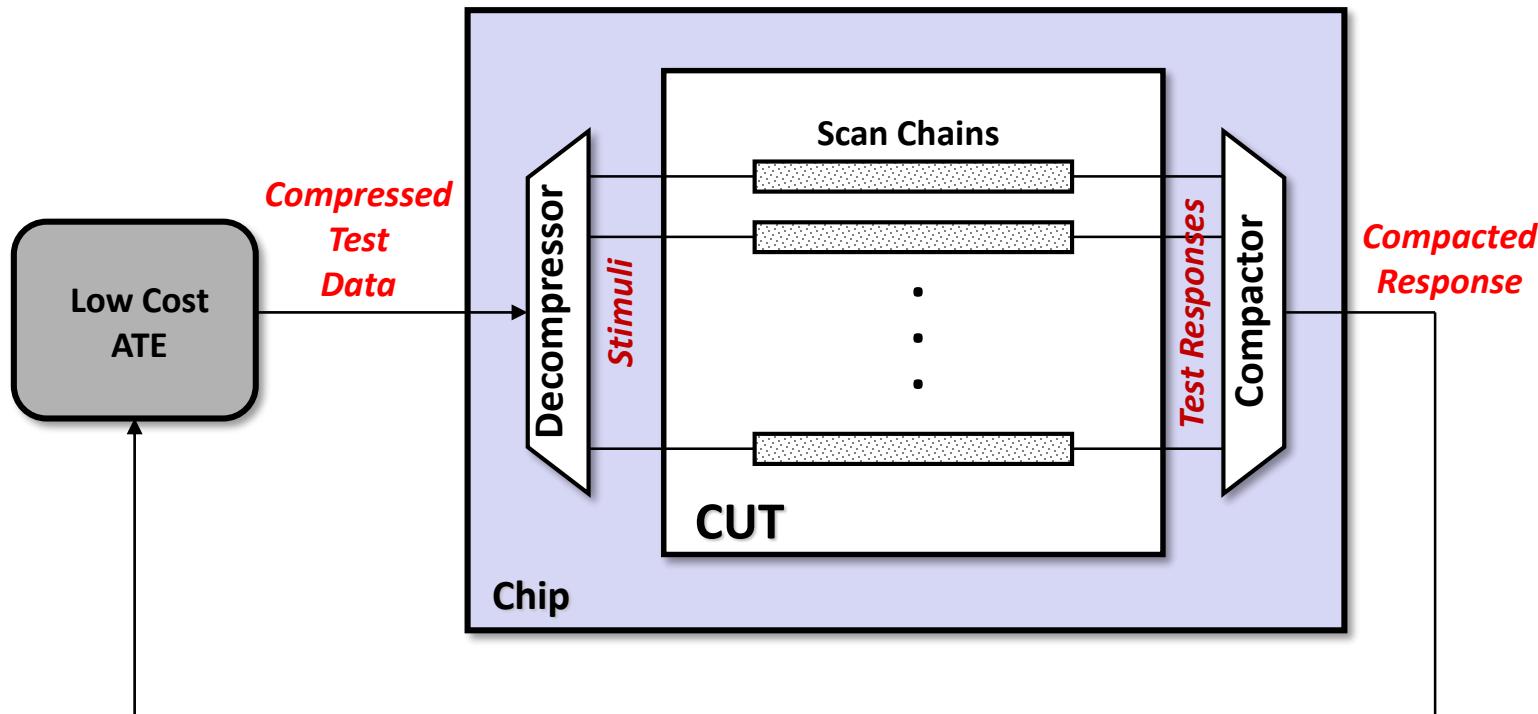
Improvement
65% !

Test Data Compression

- Τα διανύσματα δοκιμής δημιουργούνται από εργαλεία που λέγονται Automatic Test Pattern Generators (ATPGs)
- Συνήθως μόνο το 1-5% των τιμών ενός τέτοιου διανύσματος είναι συγκεκριμένες (οι υπόλοιπες είναι αδιάφορες)
- Ο όγκος δεδομένων των διανυσμάτων δημιουργεί πρόβλημα στην ταχύτητα της δοκιμής (γιατί η μεταφορα δεδομένων από έξω είναι αργή) και στις αποθηκευτικές ικανότητες του εξοπλισμού που διενεργεί τη δοκιμή (Automatic Test Equipment – ATE)
- Λύση: η συμπίεση των δεδομένων καθώς και των αποκρίσεων.

Test Data Compression

- Τα κρίσιμα τμήματα είναι ο αποσυμπιεστής (decompressor) και ο συμπιεστής (compactor) γιατί υλοποιούνται μέσα στο Ο.Κ. με τη χρήση υλικού (σε αντίθεση με τις αντίστροφες διαδικασίες που εκτελούνται εκτός του Ο.Κ. με λογισμικό)

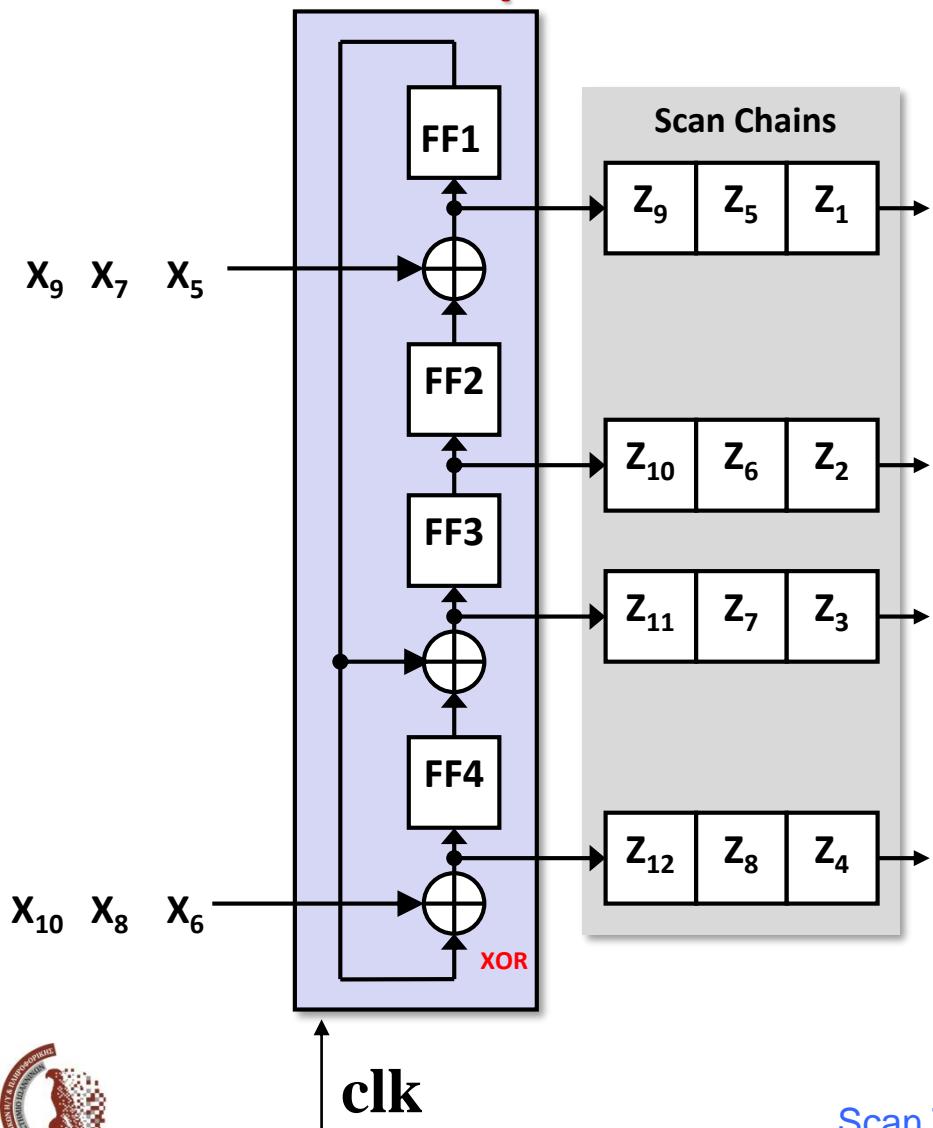


Linear Feedback Shift Register (LFSR)

Γραμμικός Αποσυμπιεστής

(Linear Decompressor pg 351)

Decompressor



Scan Testing

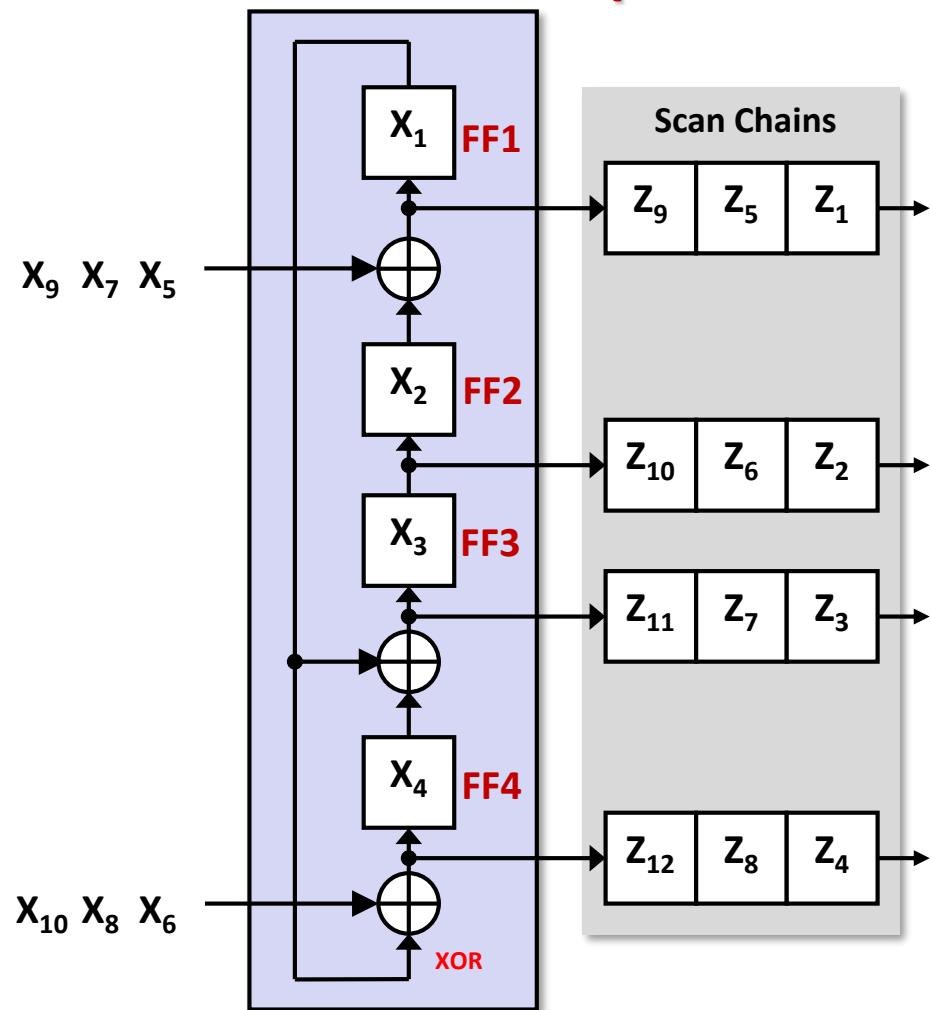
- Γενικά ο Γραμμικός Αποσυμπιεστής είναι μια γραμμική διάταξη από Flip-Flops και XOR πύλες. Φορτώνοντας παράλληλα ένα μικρό αριθμό τιμών (τιμές X) παράγουμε παράλληλα ένα μεγάλο αριθμό τιμών (τιμές Z) οι οποίες φορτώνονται παράλληλα στις αλυσίδες σάρωσης.

- Σε κάθε παλμό ρολογιού εισέρχεται ένα διάνυσμα τιμών εισόδου (πχ (X_5, X_6), (X_7, X_8) κοκ) και παράγεται το αντίστοιχο διάνυσμα τιμών εξόδου (πχ (Z_1, Z_2, Z_3, Z_4), ($Z_5 - Z_8$) κοκ)

- Το διάνυσμα εξόδου είναι συνάρτηση των εισόδων και της τρέχουσας κατάστασης των FFs.

Γραμμικός Αποσυμπιεστής (Linear Decompressor)

Decompressor



- 1) Τα FFs αρχικοποιούνται στις τιμές X_1-X_4 όπως φαίνεται στο σχήμα. Με βάση τις τιμές των FFs και τις τιμές εισόδου παράγονται οι τιμές $Z_1 - Z_4$.
- 2) Στο επόμενο ρολόι οι τιμές των FFs θα αλλάξουν ($FF1 = Z_1, FF2 = Z_2, FF3 = Z_3, FF4 = Z_4$). Με βάση τις νέες τιμές των FFs και τις νέες τιμές εισόδου (X_7, X_8) υπολογίζονται οι νέες τιμές εξόδου $Z_5 - Z_8$.

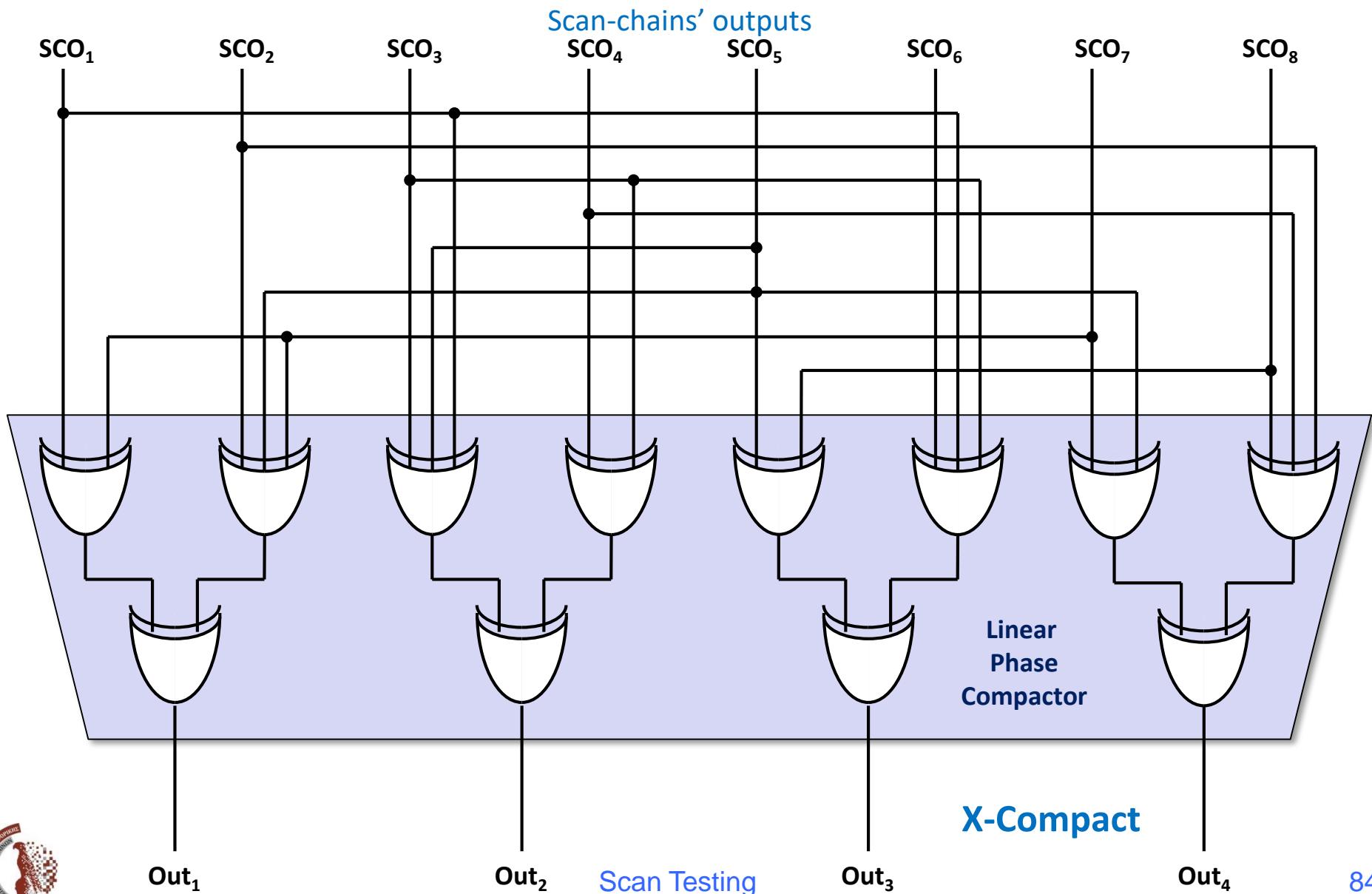
Cycle 3

Cycle 2

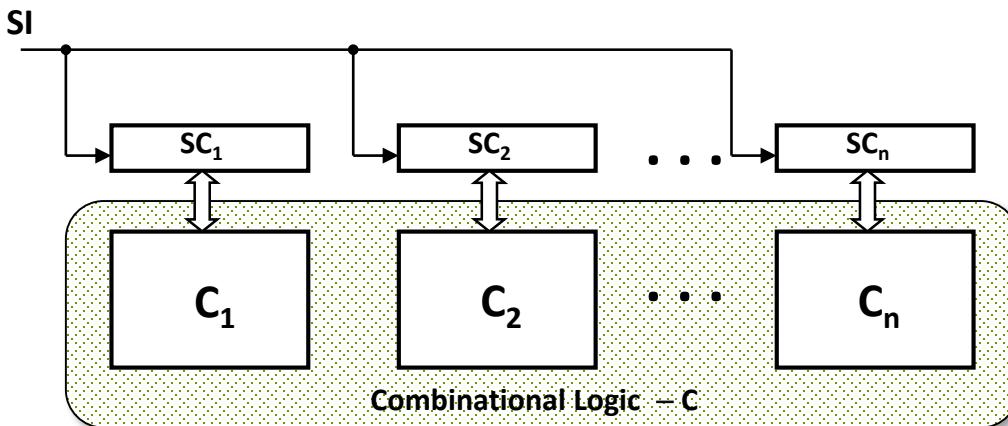
Cycle 1

$Z_9 = X_1 \oplus X_4 \oplus X_9$	$Z_5 = X_3 \oplus X_7$	$Z_1 = X_2 \oplus X_5$
$Z_{10} = X_1 \oplus X_2 \oplus X_5 \oplus X_6$	$Z_6 = X_1 \oplus X_4$	$Z_2 = X_3$
$Z_{11} = X_2 \oplus X_3 \oplus X_5 \oplus X_7 \oplus X_8$	$Z_7 = X_1 \oplus X_2 \oplus X_5 \oplus X_6$	$Z_3 = X_1 \oplus X_4$
$Z_{12} = X_3 \oplus X_7 \oplus X_{10}$	$Z_8 = X_2 \oplus X_5 \oplus X_8$	$Z_4 = X_1 \oplus X_6$

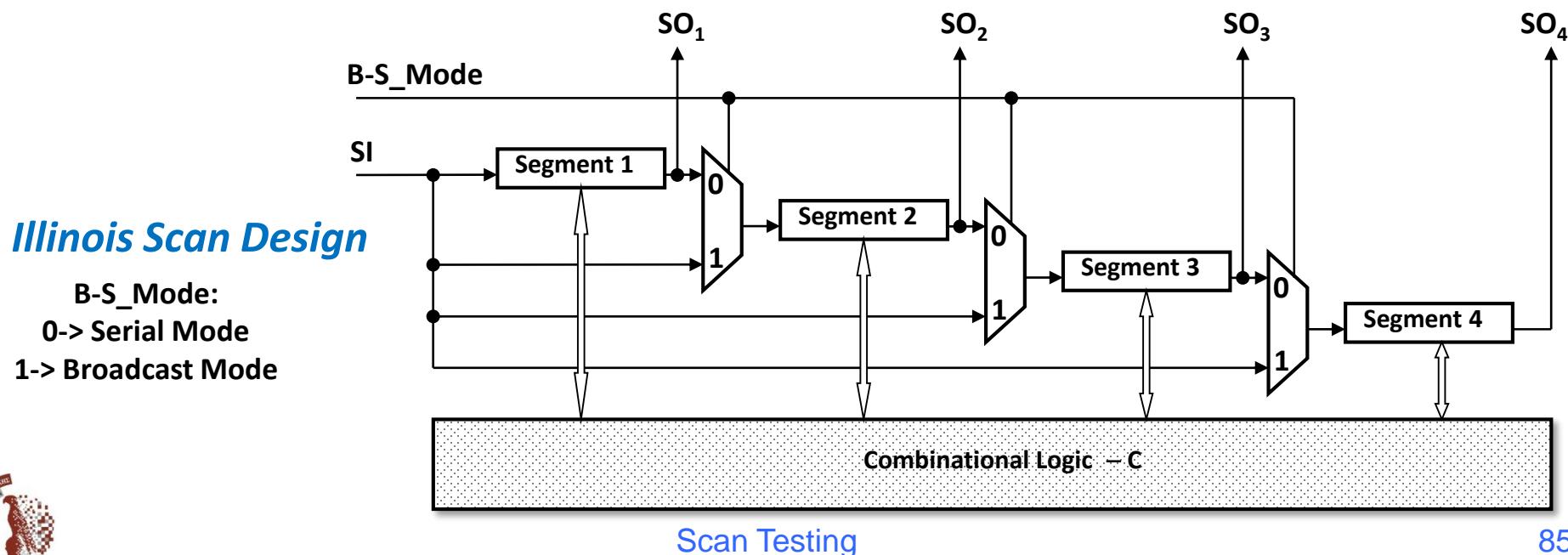
Space Compaction (pg 367)



Broadcast & Illinois Scan Design



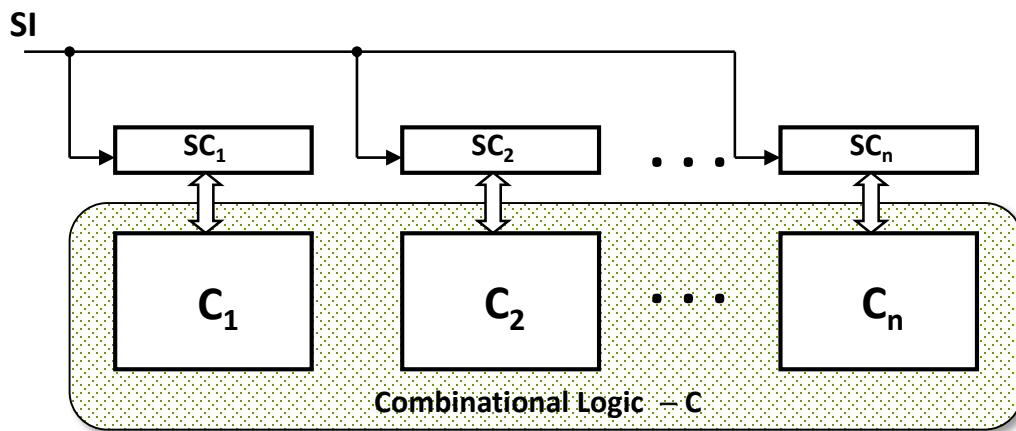
Broadcast Scan Design



Broadcast Scan Design

ΠΡΟΒΛΗΜΑ: Θέλω να τροφοδοτήσω πολλές αλυσίδες έχοντας λίγα bit εισόδου.

Αν τις ενώσω σειριακά όλες θα προκύψει πολύ μεγάλη αλυσίδα που απαιτεί πολλούς κύκλους ρολογιού.



Broadcast Scan Design

Τροφοδοτώ όλες τις αλυσίδες με το ίδιο bit εισόδου

Μειονέκτημα: όλες οι αλυσίδες παίρνουν το ίδιο διάνυσμα δοκιμής.

Χρήση: Μπορώ να εφαρμόσω διανύσματα δοκιμής που καλύπτουν τα «κοινά» ή «εύκολα» σφάλματα που αφορούν όλα τα υποκυκλώματα C_1 , C_2 ... C_n .

Illinois Scan Design

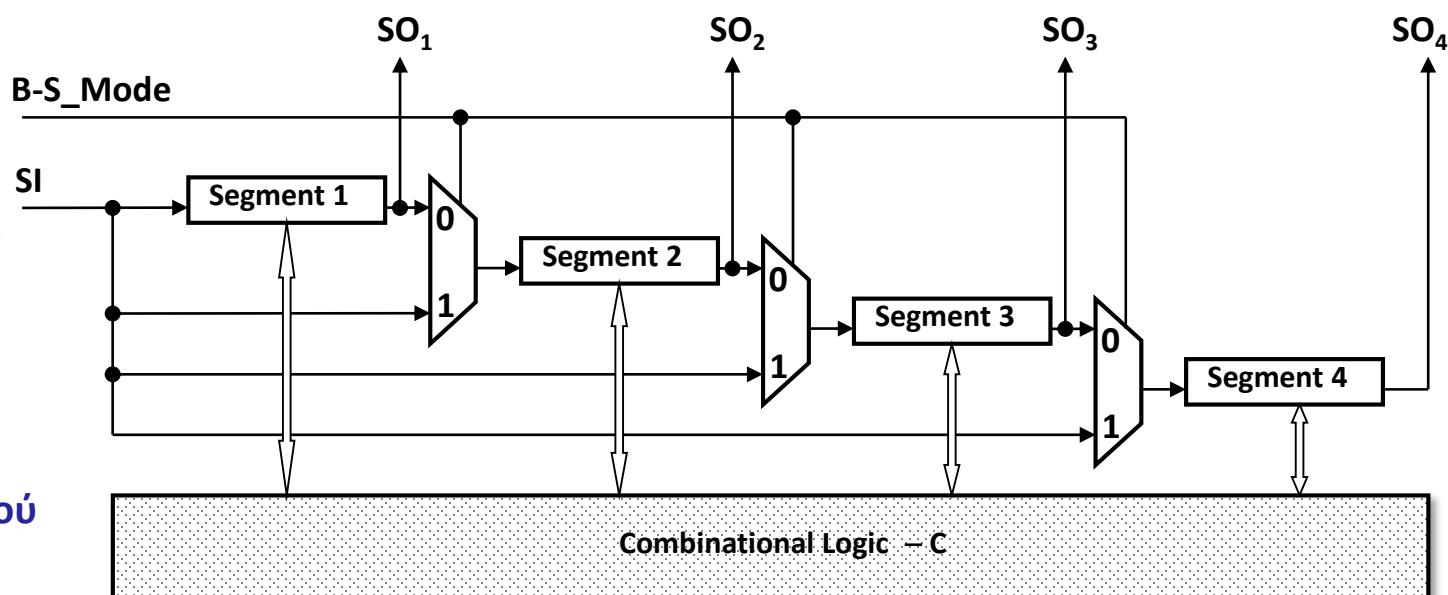
Υπάρχουν όμως σφάλματα που απαιτούν διαφορετικά διανύσματα για κάθε ένα από τα υποκυκλώματα $C_1, C_2 \dots C_n$. Για αυτά αναγκαστικά θα καταφύγουμε στη λύση της συνένωσης όλων των αλυσίδων σε μία μεγάλη αλυσίδα.

ΛΥΣΗ: η σχεδίαση ILLINOIS που μπορεί να εναλάσσεται μεταξύ Broadcast και Serial τοπολογίας.

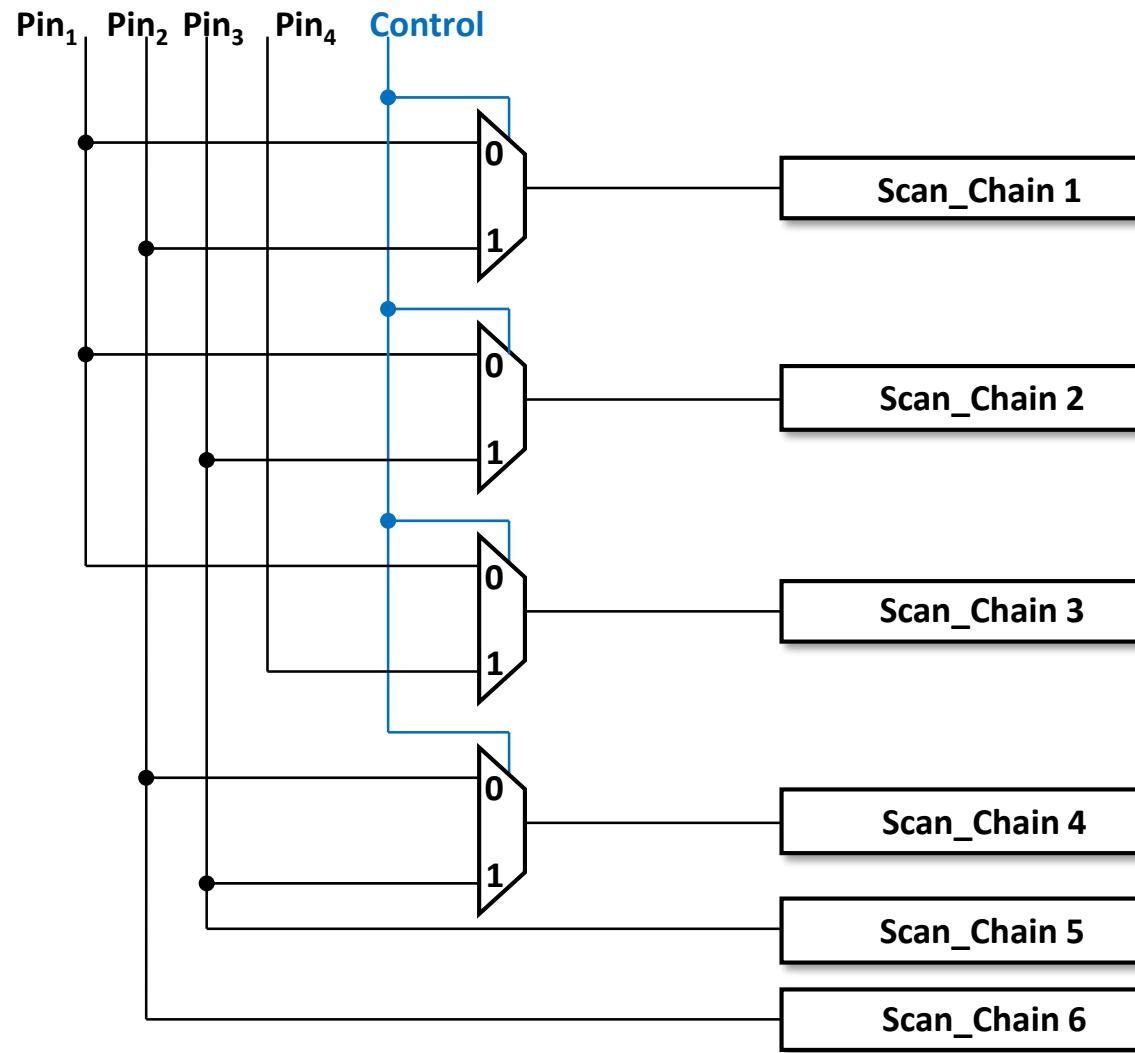
Illinois Scan Design

B-S_Mode:
0-> Serial Mode
1-> Broadcast Mode

Εναλλαγή μεταξύ
Broadcast και σειριακού
τρόπου.

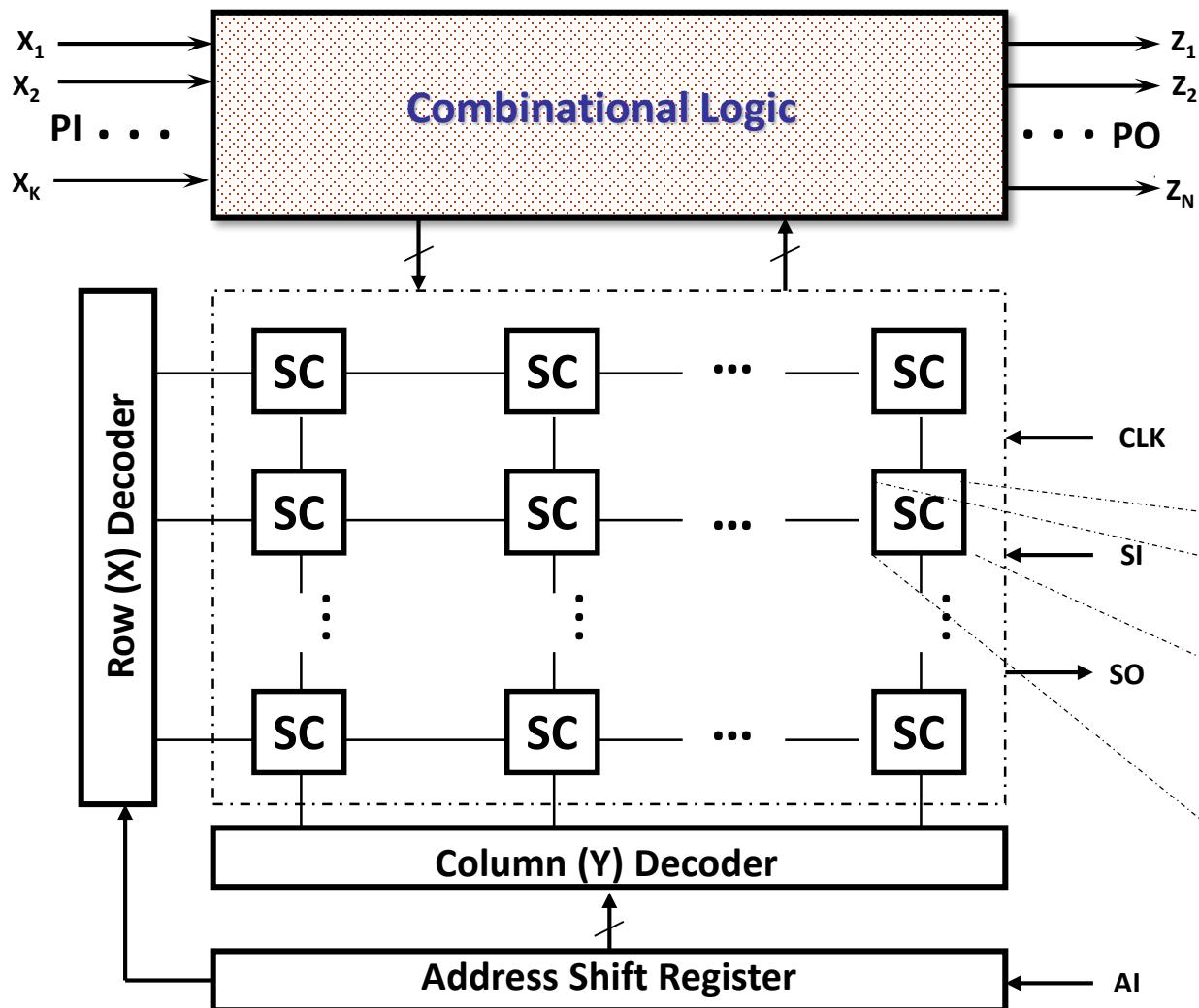


Reconfigurable Broadcast Scan Design



*Dynamic
Reconfiguration*

Random-Access Scan Design



Disadvantages:

+25% routing
+11% silicon area

Advantages:
reduction in power consumption, data volume, test application time

References

- “**Principles of Testing Electronics Systems,**” S. Mourad and Y. Zorian, *John Wiley & Sons*, 2000.
- “**Essentials of Electronic Testing: for Digital, Memory and Mixed-Signal VLSI Circuits,**” M. Bushnell and V. Agrawal, *Kluwer Academic Publishers*, 2000.
- “**Power-Constrained Testing of VLSI Circuits,**” N. Nicolici and B. Al-Hashimi, *Kluwer Academic Publishers*, 2003.
- “**Digital Systems Testing and Testable Design,**” M. Abramovici, M. Breuer and A. Friedman, *Computer Science Press*, 1990.
- “**System-on-Chip Test Architectures,**” L-T Wang, C. Stroud and N. Touba, *Morgan-Kaufmann*, 2008.
- “**VLSI Test Principles and Architectures,**” L-T Wang, C-W. Wu and X. Wen, *Morgan-Kaufmann*, 2006.

