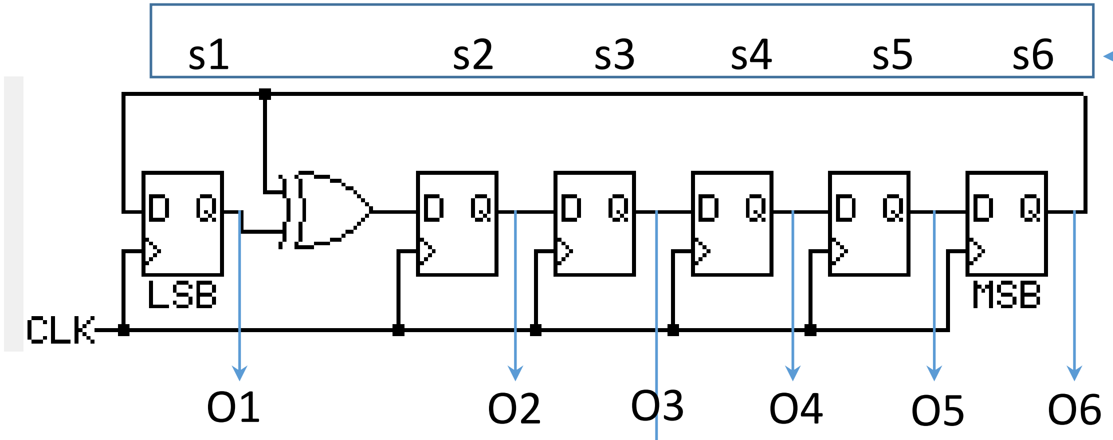
Κωδικοποίηση διανυσμάτων σε LFSR, LFSR Reseeding, MISR παραδείγματα

Βασίλης Τενέντες

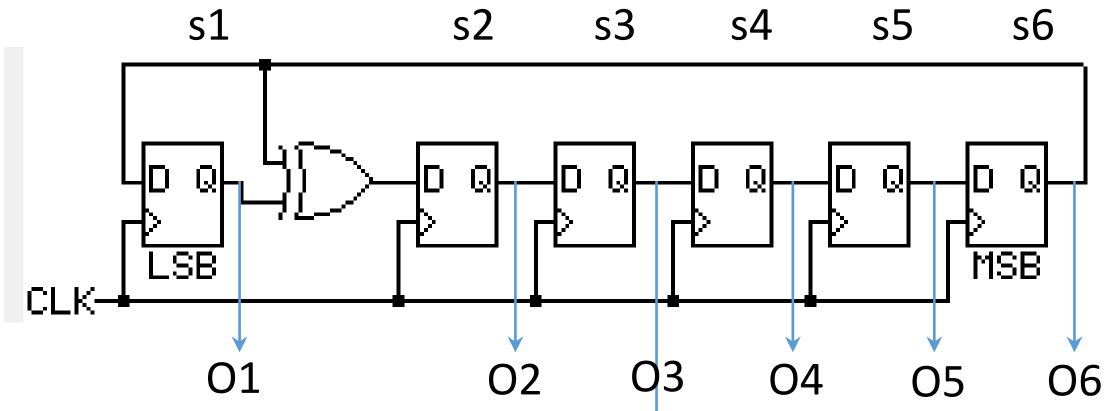
Symbolic simulation of LFSR



Αρχική κατάσταση (seed)

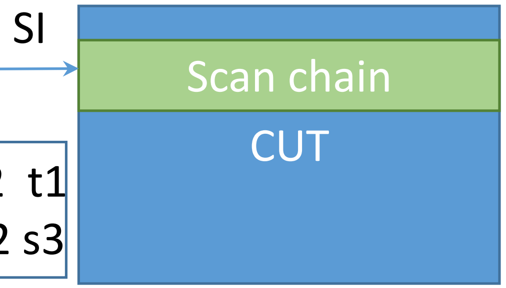
t1:	s1	s2	s3	s4	s5	s6
t2:	s6	s1s6	s2	s3	s4	s5
t3:	s5	s5s6	s1s6	s2	s3	s4
t4:	s4	s4s5	s5s6	s1s6	s2	s3
t5:	s3	s3s4	s4s5	s5s6	s1s6	s2

Σύνδεση εξόδων LFSR με εισόδους scan chains του CUT



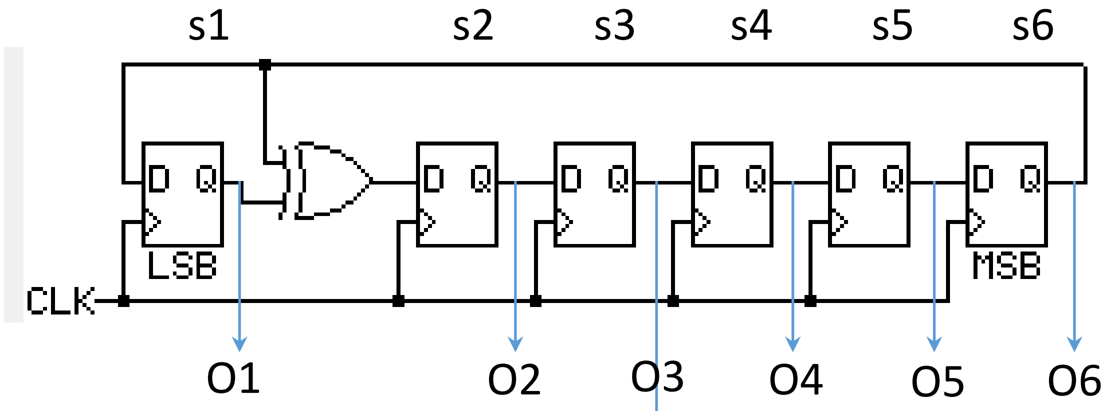
t1:	s1	s2	s3	s4	s5	s6
t2:	s6	s1s6	s2	s3	s4	s5
t3:	s5	s5s6	s1s6	s2	s3	s4
t4:	s4	s4s5	s5s6	s1s6	s2	s3

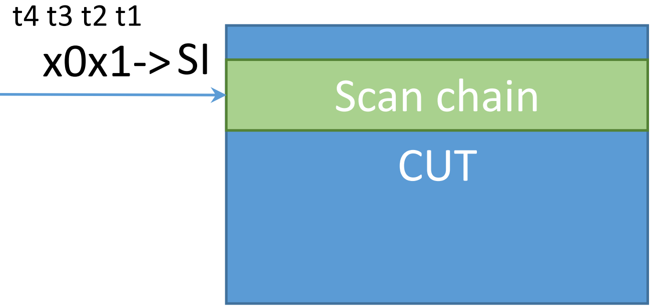
t4	t3	t2	t1
s5s6	s1s6	s2	s3



Symbolic simulation κωδικοποίηση test cube σε έξοδο του LFSR

test cube: x0x1





t1:	s1	s2	s3 =1	s4	s5	s6
t2:	s6	s1s6	s2 =x	s3	s4	s5
t3:	s5	s5s6	s1s6 =0	s2	s3	s4
t4:	s4	s4s5	s5s6 =x	s1s6	s2	s3
t5:	s3	s3s4	s4s5	s5s6	s1s6	s2

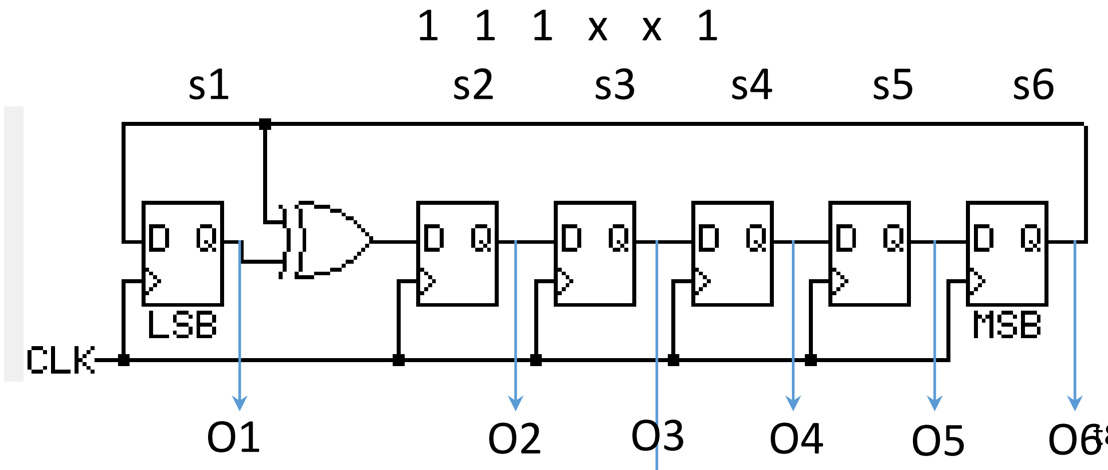
Σύστημα επίλυσης του test cube από την έξοδο O3:

s3=1
s2=x
s1s6=0
s5s6=x

s3=1
s1=s6

s1	s2	s3	s4	s5	s6
1	x	1	x	x	1
0	x	1	x	x	0

Symbolic simulation κωδικοποίηση πολλαπλών test cubes σε έξοδο του LFSR

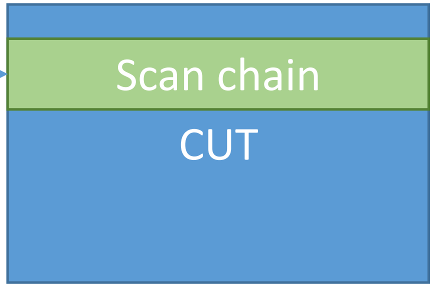


1st test cube: x0x1 0 1 1 0 0 0

2nd test cube: 10xx

t8 t7 t6 t5 | t4 t3 t2 t1

10xx | x0x1->SI



t1:	s1	s2	1= s3	s4	s5	s6
t2:	s6	s1s6	x= s2	s3	s4	s5
t3:	s5	s5s6	0= s1s6	s2	s3	s4
t4:	s4	s4s5	x= s5s6	s1s6	s2	s3
t5:	s3	s3s4	x= s4s5	s5s6	s1s6	s2
t6:	s2	s2s3	x= s3s4	s4s5	s5s6	s1s6
t7:	s1s6	s1s6s2	0= s2s3	s3s4	s4s5	s5s6
t8:	s5s6	s5s1	1= s1s2s6	s2s3	s3s4	s4s5

Σύστημα 1ου test cube από O3:

s3=1
s2=x
s1s6=0
s5s6=x

→

s3=1
s1=s6

λύσεις →

s1	s2	s3	s4	s5	s6
1	x	1	x	x	1
0	x	1	x	x	0

Σύστημα 2ου test cube από O3:

s4s5=x
s3s4=x
s2s3=0
s1s2s6=1

→

s2s3=0
s1s2s6=1

→

Σύστημα 1ου και 2ου test cube

s1=s6
s2=1
s3=1

λύσεις →

s1	s2	s3	s4	s5	s6
1	1	1	x	x	1
0	1	1	x	x	0

Αυτοματοποίηση της Επίλυσης των γραμμικών συστημάτων με την τεχνική Gauss Jordan elimination (GJE) – εφαρμόζεται και σε Boolean άλγεβρα

Εξισώσεις του 1ου test cube από O3:

s3=1, s1s6=0

Εξισώσεις του 2ου test cube από O3:

s2s3=0, s1s2s6=1

Σύστημα προς επίλυση

s3=1

s1s6=0

s2s3=0

s1s2s6=1

Πινακοποίηση του συστήματος

s1	s2	s3	s4	s5	s6	result
0	0	1	0	0	0	1
1	0	0	0	0	1	0
0	1	1	0	0	0	0
1	1	0	0	0	1	1

Ισχύουν κανονικά οι γραμμοπράξεις της GJE

s1	s2	s3	s4	s5	s6	result
1	1	0	0	0	1	1
1	0	0	0	0	1	0
0	1	1	0	0	0	0
0	0	1	0	0	0	1

s1	s2	s3	s4	s5	s6	result
1	1	0	0	0	1	1
1	0	0	0	0	1	0
0	**1**	**1**	**0**	**0**	**0**	**0**
0	**0**	**1**	**0**	**0**	**0**	**1**

xor

s1	s2	s3	s4	s5	s6	result
1	1	0	0	0	1	1
1	0	0	0	0	1	0
0	1	0	0	0	0	1
0	0	1	0	0	0	1

s1	s2	s3	s4	s5	s6	result
1	**1**	**0**	**0**	**0**	**1**	**1**
1	0	0	0	0	1	0
0	**1**	**0**	**0**	**0**	**0**	**1**
0	0	1	0	0	0	1

xor

s1	s2	s3	s4	s5	s6	result
1	0	0	0	0	1	0
1	0	0	0	0	1	0
0	1	0	0	0	0	1
0	0	1	0	0	0	1

xor

s1	s2	s3	s4	s5	s6	result
0	0	0	0	0	0	0
1	0	0	0	0	1	0
0	1	0	0	0	0	1
0	0	1	0	0	0	1

s1	s2	s3	s4	s5	s6	result
1	0	0	0	0	1	0
0	1	0	0	0	0	1
0	0	1	0	0	0	1

s1	s2	s3	s6	result
1	0	0	1	0
0	1	0	0	1
0	0	1	0	1

s1s6=0

s2=1

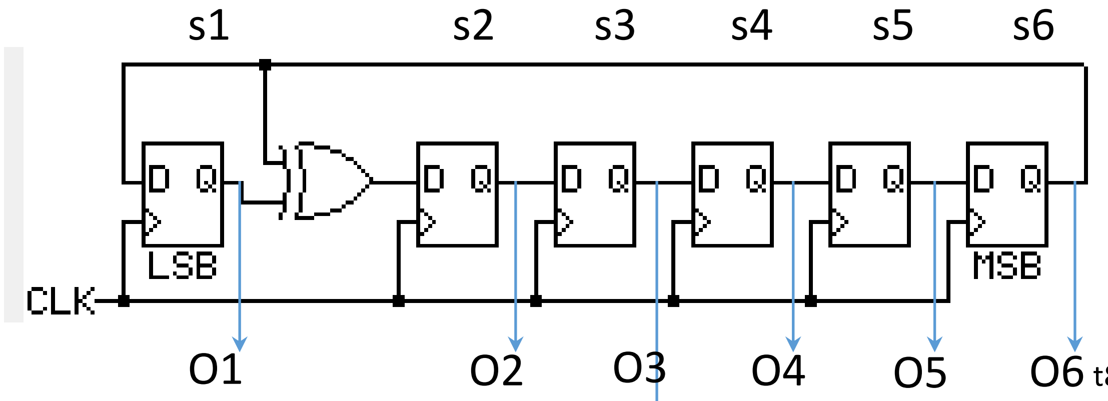
s3=1

s1=s6

s2=1

s3=1

Επαλήθευση της λύσης με Binary simulation



Η λύση
s1 s2 s3 s4 s5 s6
0 1 1 0 0 0

Test cubes που κωδικοποιήσαμε

1st test cube: x0x1

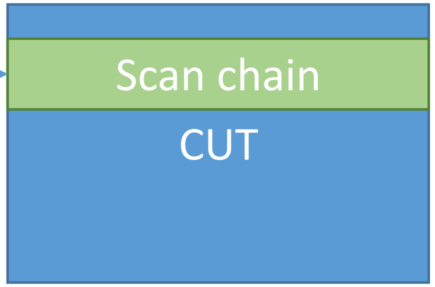
2nd test cube: 10xx

t8 t7 t6 t5 | t4 t3 t2 t1

10xx | x0x1->SI

1010 | 0011->

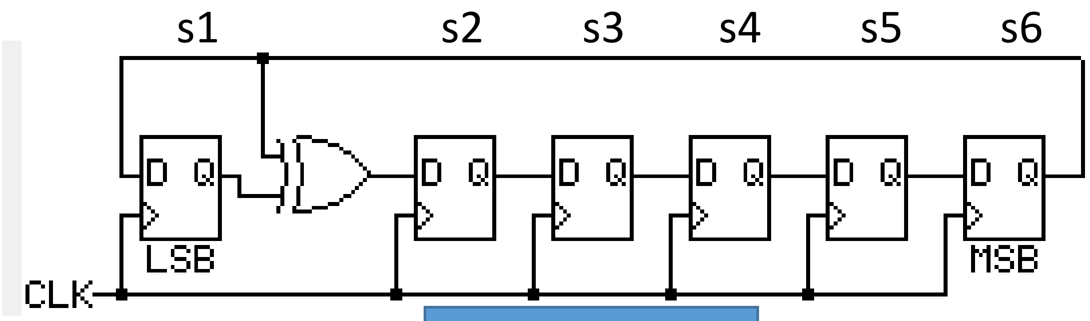
Vector 2 Vector 1

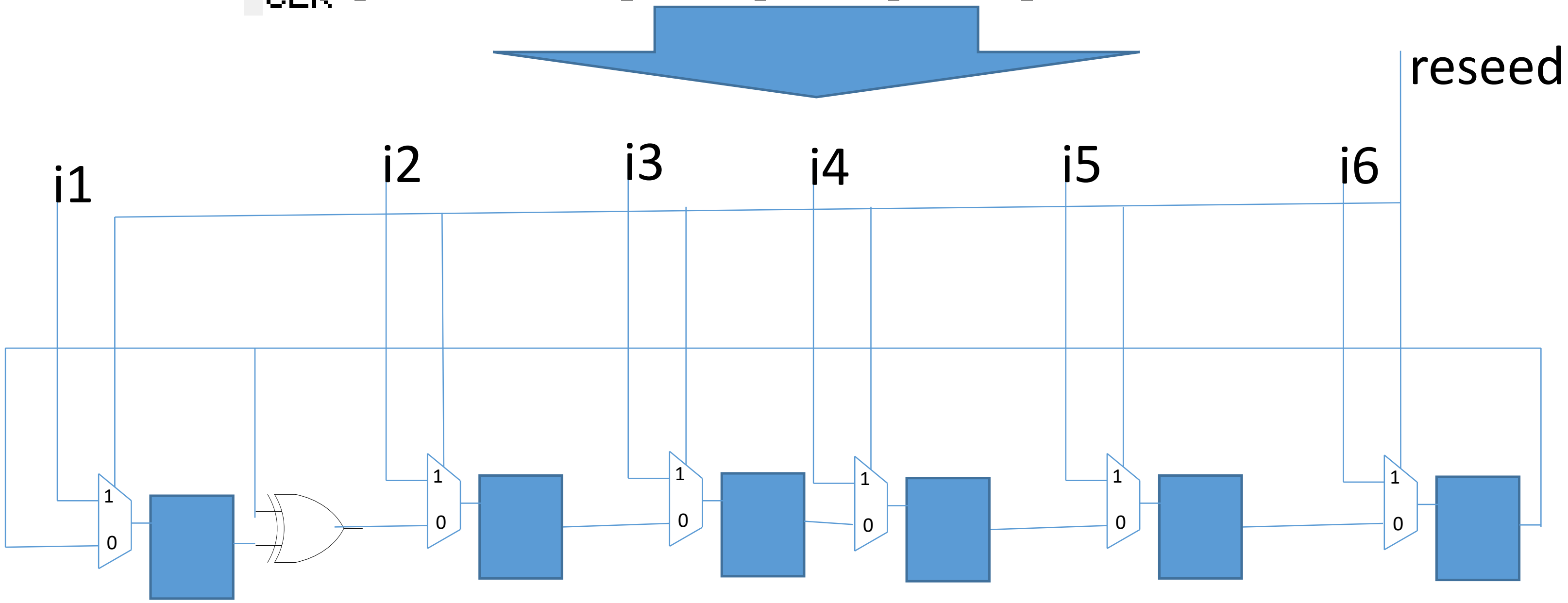


	O1	O2		O3	O4	O5	O6
t1:	0	1	Vector 1	1	0	0	0
t2:	0	0		1	1	0	0
t3:	0	0		0	1	1	0
t4:	0	0		0	0	1	1
t5:	1	1	Vector 2	0	0	0	1
t6:	1	0		1	0	0	0
t7:	0	1		0	1	0	0
t8:	0	0		1	0	1	0

Διανύσματα που παρήχθησαν (αυτά που πέρασαν στις scan chains)

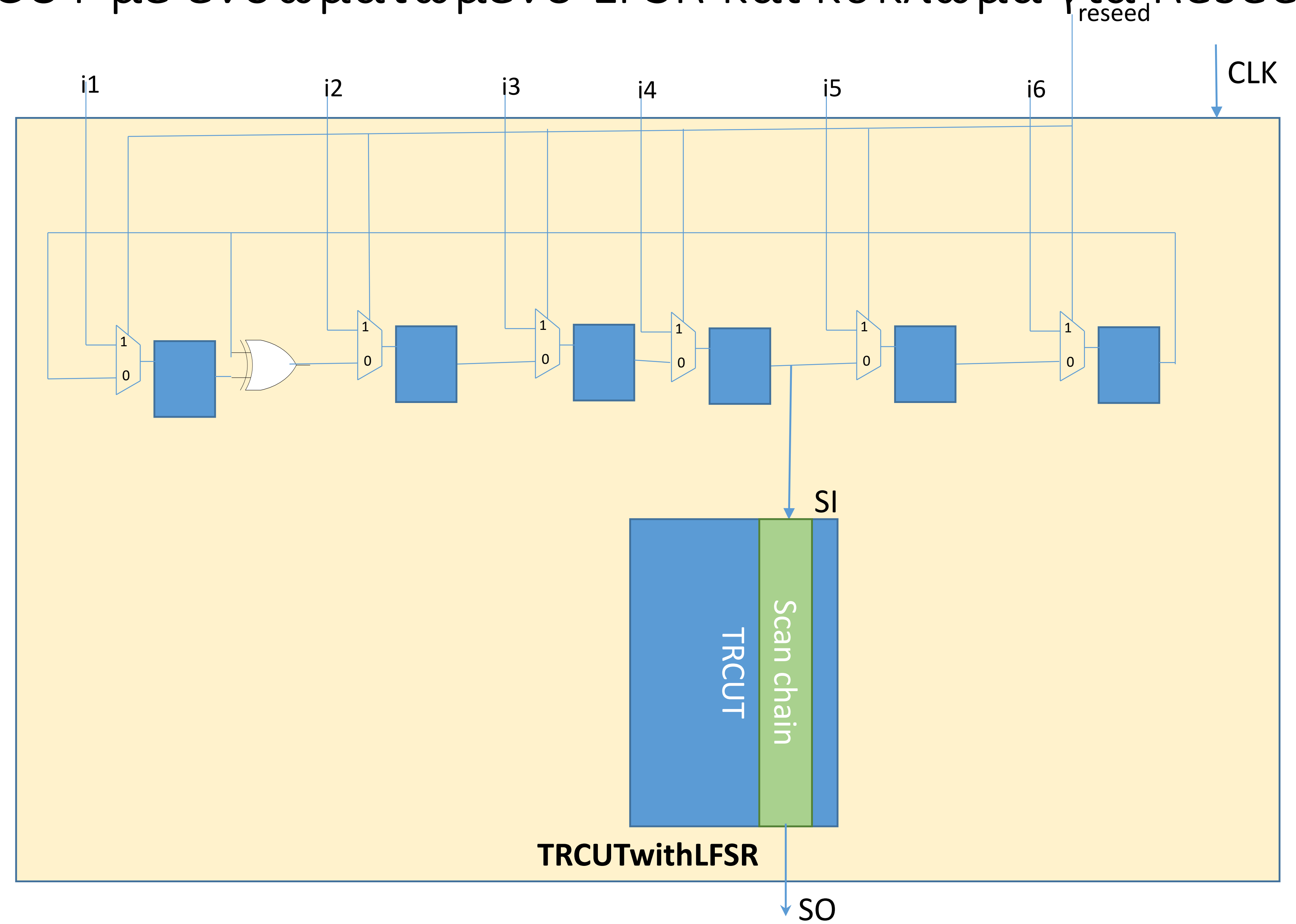
LFSR Reseeding από παράλληλες εισόδους





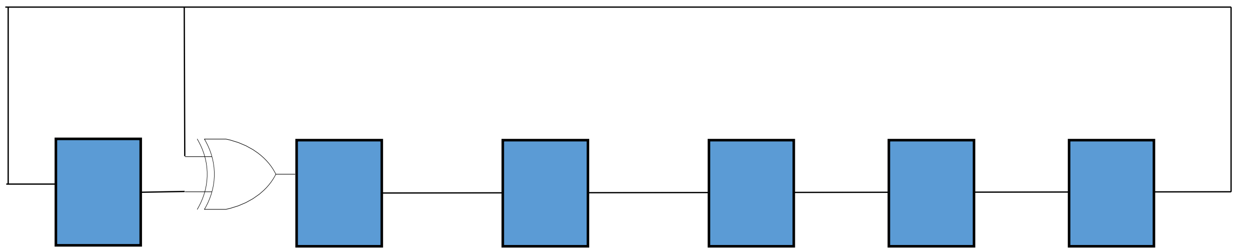
Όταν το σήμα *reseed* είναι στο λογικό-1 τότε τα μπορούμε να φορτώσουμε ένα νέο seed στο LFSR από τις εισόδους i. Προσοχή το LFSR είναι σύγχρονο κύκλωμα, το ρολόι στο σχήμα δεν φαίνεται. Οι είσοδοι θα περάσουν στα flip-flops του LFSR με την ακμή του ρολογιού κατά το reseeding.

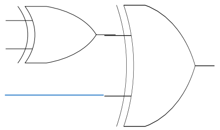
TRCUT με ενσωματωμένο LFSR και κύκλωμα για Reseeding



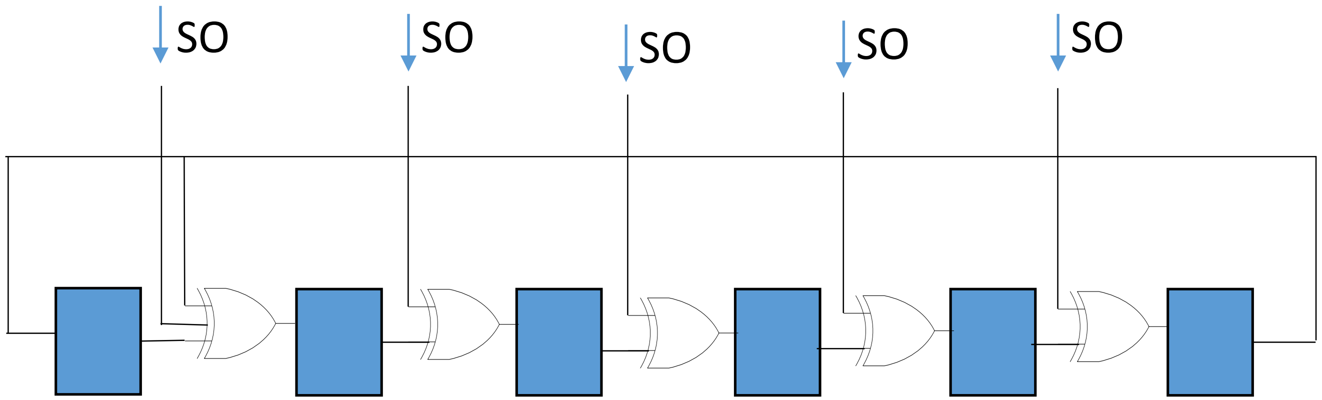
Σχεδίαση MISR

Ξεκινάμε από LFSR

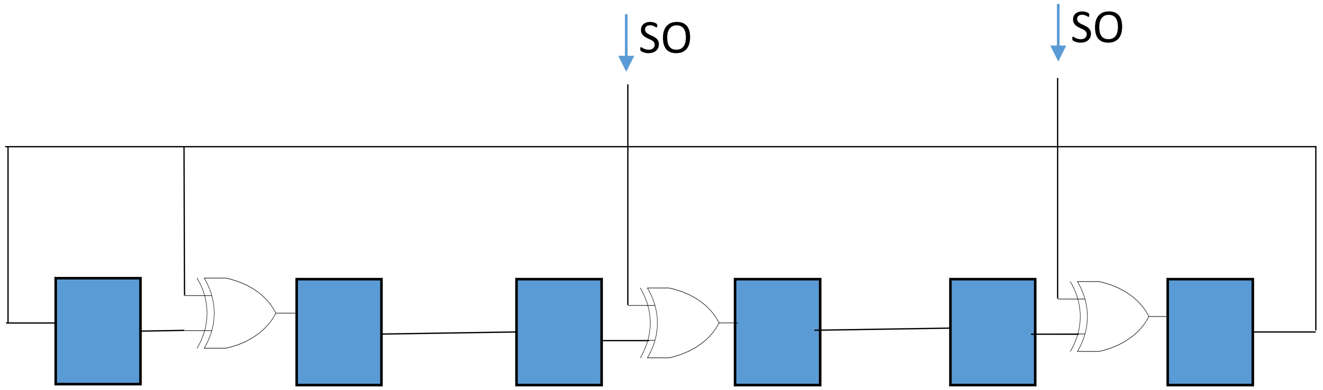




Βάζουμε όσα xor taps εισόδου θέλουμε. Όπου έχει ήδη, τους προσθέτουμε είσοδο.



ΠΡΟΣΟΧΗ: αν έχουμε ένα κύκλωμα με μόνο 2 αλυσίδες τότε δεν χρειάζονται όλα τα xor taps εισόδου πχ:

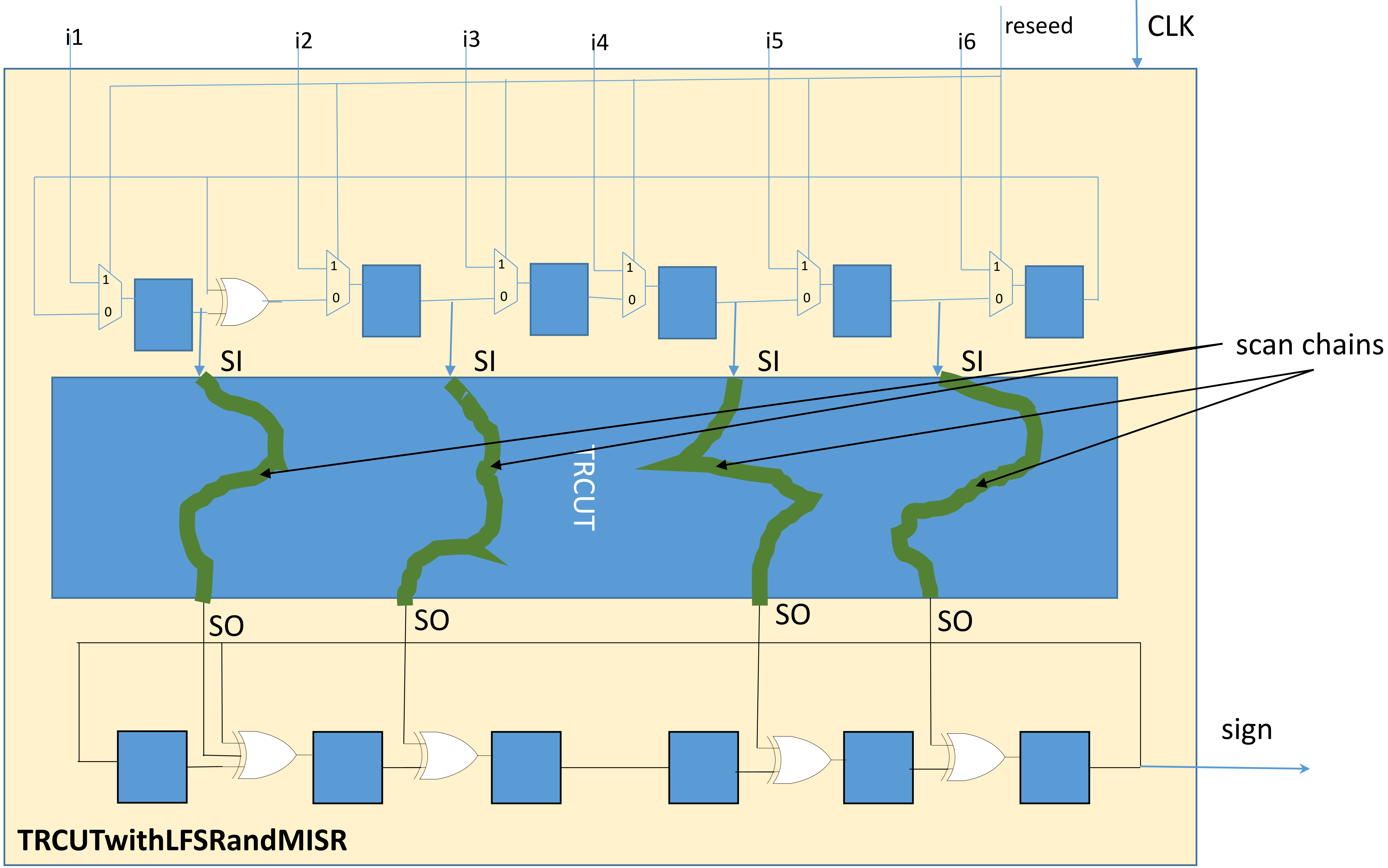


Ομοίως μπορείτε να επιλέξετε όποιο flip flop του MISR για να οδηγήσετε το sign

TRCUT με ενσωματωμένο LFSR με Reseeding και MISR

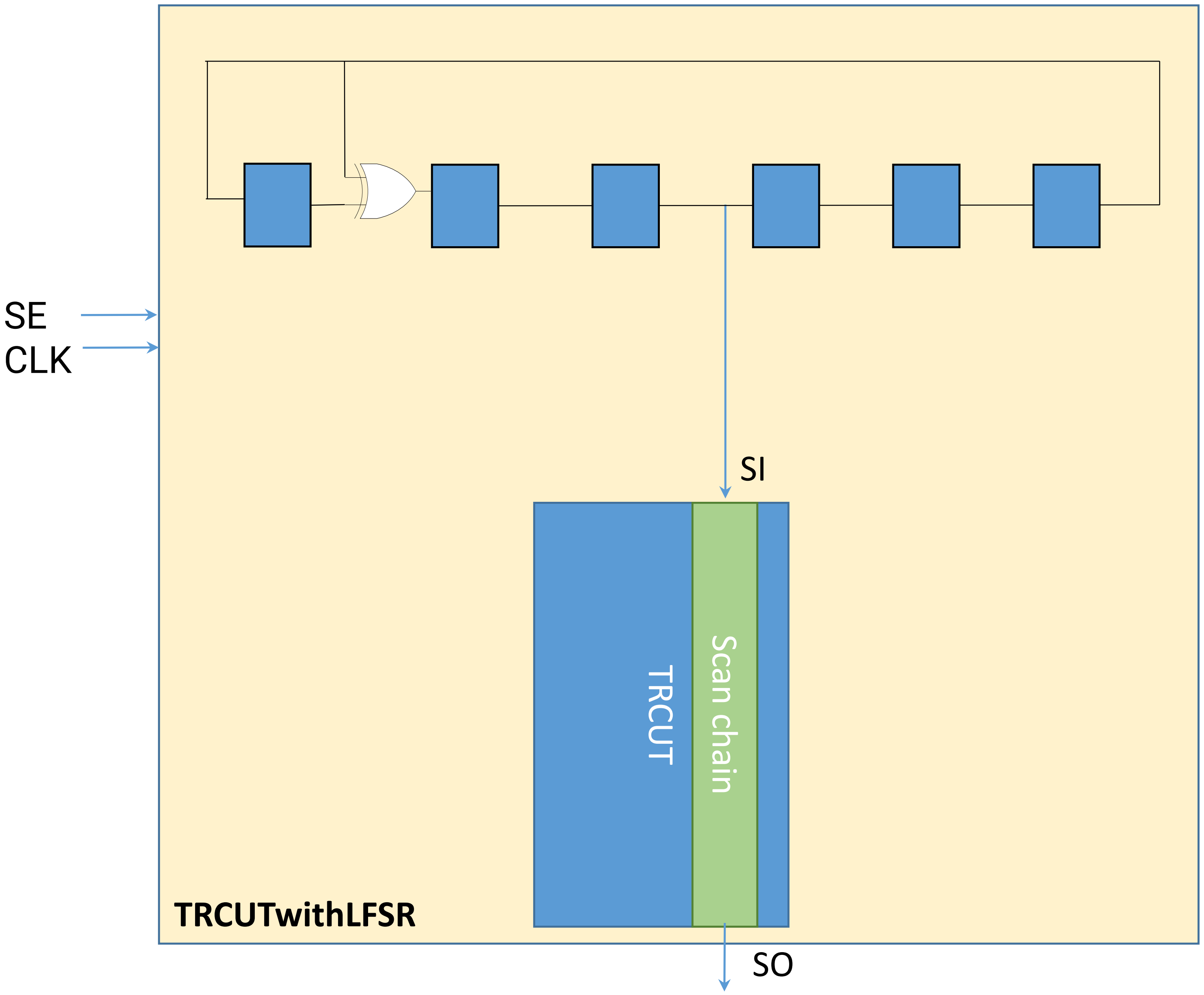
Πόσοι κύκλοι απαιτούνται για να εφαρμόσουμε 10 διανύσματα με Scan chain length=5 και να πάρουμε την υπογραφή με ένα MISR μεγέθους M:

Για να εφαρμόσουμε 10 διανύσματα θέλουμε (5+1)*10+5 κύκλους
65 κύκλους (αυτοί οι +5 κύκλοι είναι για να βγουν τα δεδομένα που έχουν οι scan chains κατά την εφαρμογή του τελευταίου διανύσματος). Αφού περάσουν
αυτοί οι κύκλοι θα το αφήσουμε άλλους M κύκλους (όπου M το μέγεθος του MISR) για να πάρουμε τις τιμές που έχει μέσα του το MISR από το sign.
Συνολικά θα τρέξει για 65+M κύκλους. Τα **M bits** που θα μαζέψουμε από το sign στο τέλος, είναι η **υπογραφή** των tests που εφαρμόσαμε

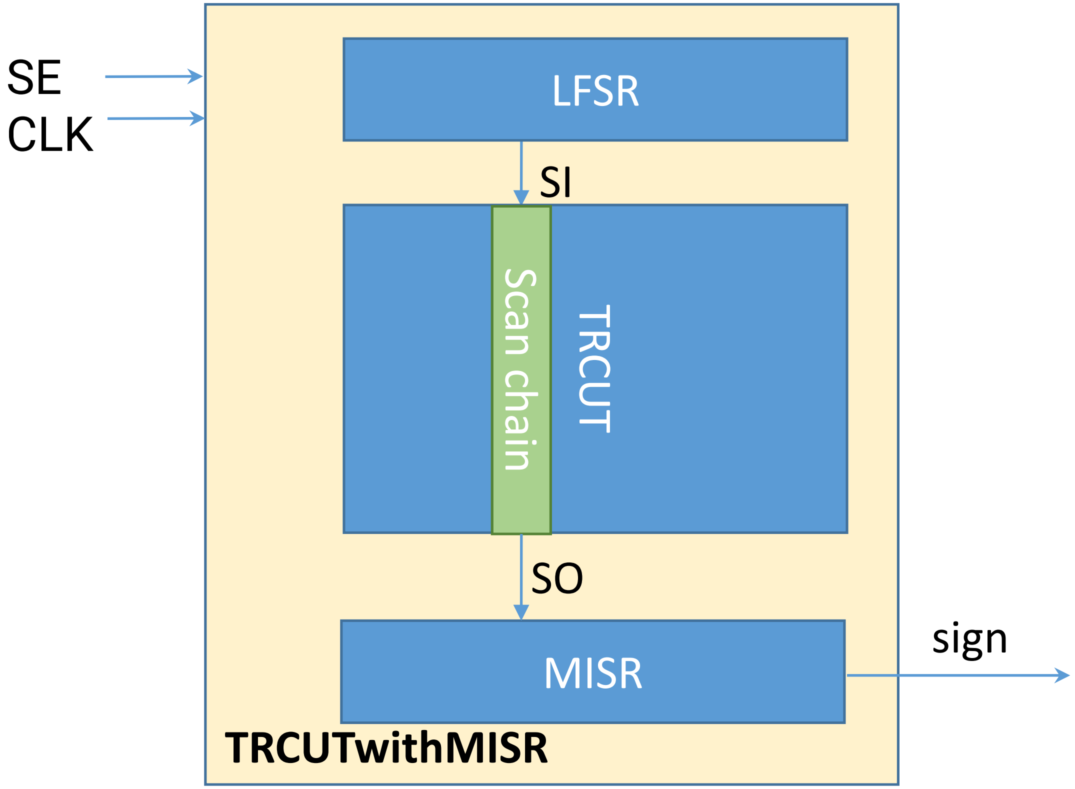


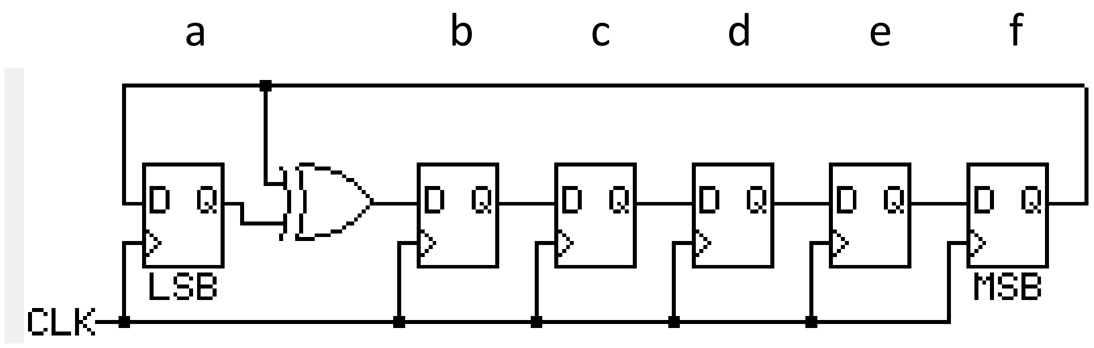
Τέλος

ακολουθούν
κάποια σχήματα
από τις εκφωνήσεις
των ασκήσεων

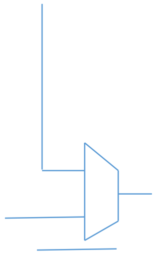

SE
CLK
SI
TRCUT
Scan chain
TRCUTwithLFSR
SO

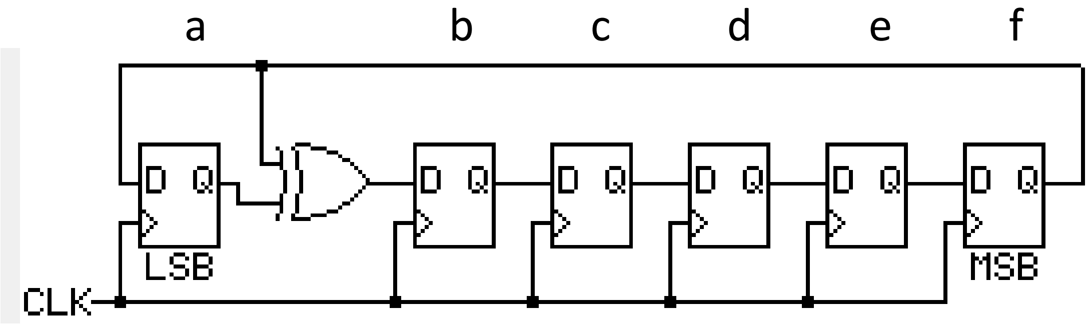
TRCUT με ενσωματωμένο LFSR και MISR





a
b
c
d
e
f
D Q
D Q
D Q
D Q
D Q
D Q
LSB
MSB
CLK




a
b
c
d
e
f
D Q
LSB
MSB
CLK