

# Υπολογιστικά Συστήματα Υψηλής Αξιοπιστίας

## Προσομοίωση σφαλμάτων

Δρ. Γκάμας Βασίλειος

Επιστημονικός Συνεργάτης  
vgkamas@uniwa.gr

Πανεπιστήμιο Δυτικής Αττικής  
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

# Σκοπός παρουσίασης

- Να παρουσιάσει θέματα και τεχνικές που σχετίζονται με την προσομοίωση σφαλμάτων σε λογικά κυκλώματα

# Προσομοίωση σφαλμάτων

- Τι είναι η προσομοίωση σφαλμάτων;
- Δεδομένου
  - Ενός λογικού κυκλώματος
  - Ενός συνόλου διανυσμάτων δοκιμής
  - Ενός μοντέλου σφαλμάτων
- Καθορίζουμε
  - Τις εσφαλμένες εξόδους του κυκλώματος
  - Τα μη ανιχνεύσιμα σφάλματα
  - Την κάλυψη σφαλμάτων

# Προσομοίωση σφαλμάτων

- Χρονική πολυπλοκότητα
  - $n$ : μέγεθος κυκλώματος, πλήθος λογικών πυλών
  - $p$ : πλήθος διανυσμάτων δοκιμής
  - $f$ : πλήθος μοντελοποιημένων σφαλμάτων
- Καθώς το  $f$  είναι σχεδόν ανάλογο του  $n$ , η συνολική χρονική πολυπλοκότητα είναι  $O(pn^2)$

# Τεχνικές προσομοίωσης σφαλμάτων

- Σειριακή (Serial)
- Παράλληλη (Parallel)
- Επαγωγική (Deductive)
- Ταυτόχρονη (Concurrent)

# Σειριακή προσομοίωση σφαλμάτων (1/4)

- Προσομοίωση του fault-free κυκλώματος και αποθήκευση των απαντήσεων.
- Για κάθε πιθανό σφάλμα επαναλαμβάνονται τα παρακάτω βήματα:

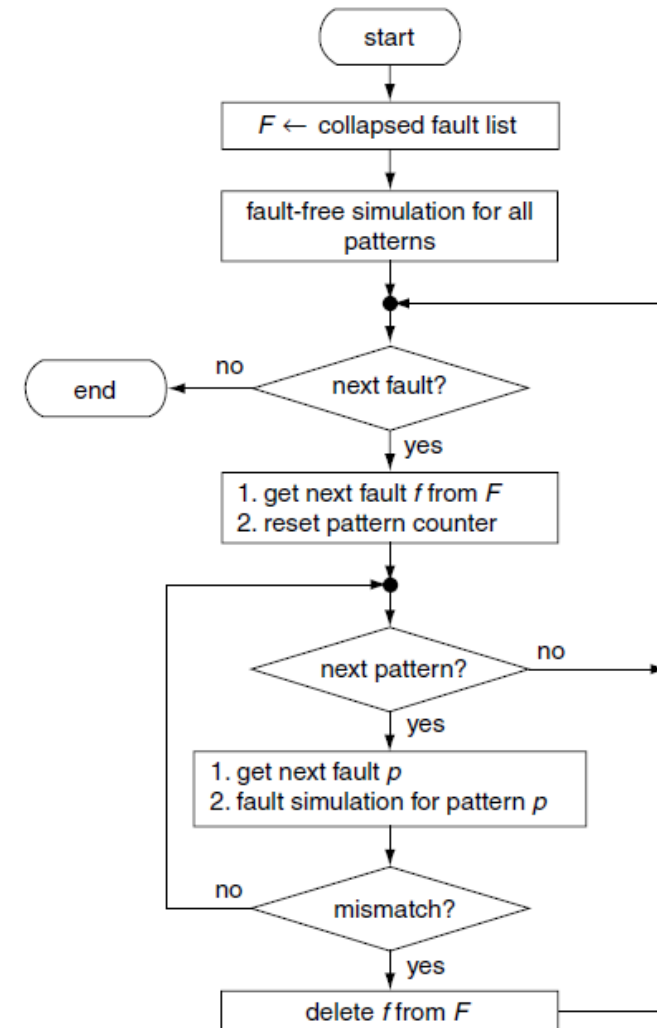
*Modify netlist by injecting one fault*

*Simulate modified netlist, vector by vector, comparing responses with saved responses*

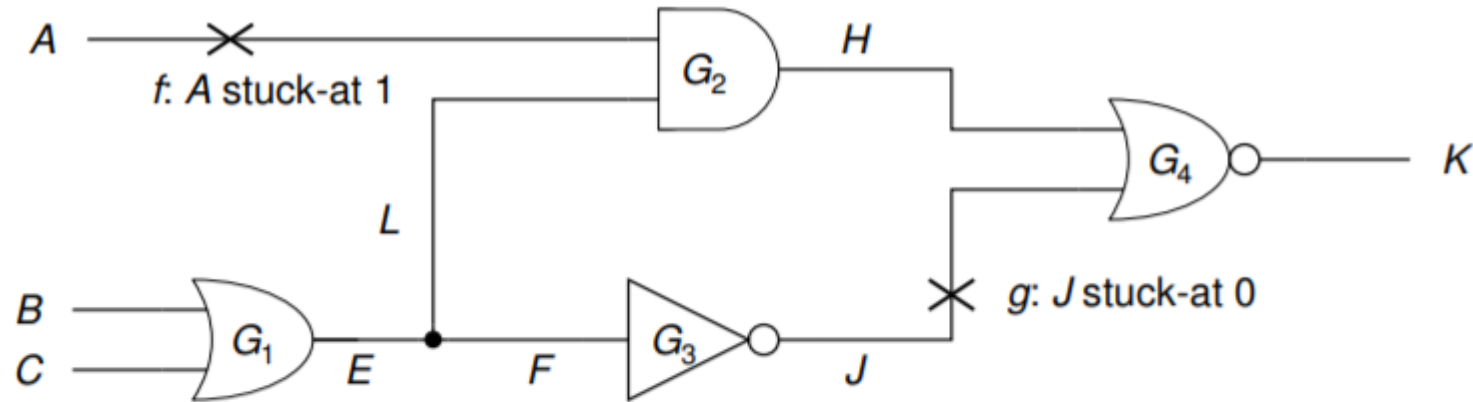
*If response differs, report fault detection and suspend simulation of remaining vectors*

# Σειριακή προσομοίωση σφαλμάτων (2/4)

- Πλεονεκτήματα:
  - Εύκολο στην υλοποίηση, χαμηλές απαιτήσεις σε μνήμη
  - Τα περισσότερα σφάλματα μπορούν να προσομοιωθούν
- Μειονεκτήματα
  - Επαναλαμβανόμενοι υπολογισμοί
  - Ο χρόνος CPU είναι απαγορευτικός για VLSI κυκλώματα



# Σειριακή προσομοίωση σφαλμάτων (3/4)



Pat. #	Input			Internal					Output		
	<i>A</i>	<i>B</i>	<i>C</i>	<i>E</i>	<i>F</i>	<i>L</i>	<i>J</i>	<i>H</i>	<i>K<sub>good</sub></i>	<i>K<sub>f</sub></i>	<i>K<sub>g</sub></i>
<i>P1</i>	0	1	0	1	1	1	0	0	1	0	1
<i>P2</i>	0	0	1	1	1	1	0	0	1	0	1
<i>P3</i>	1	0	0	0	0	0	1	0	0	0	1



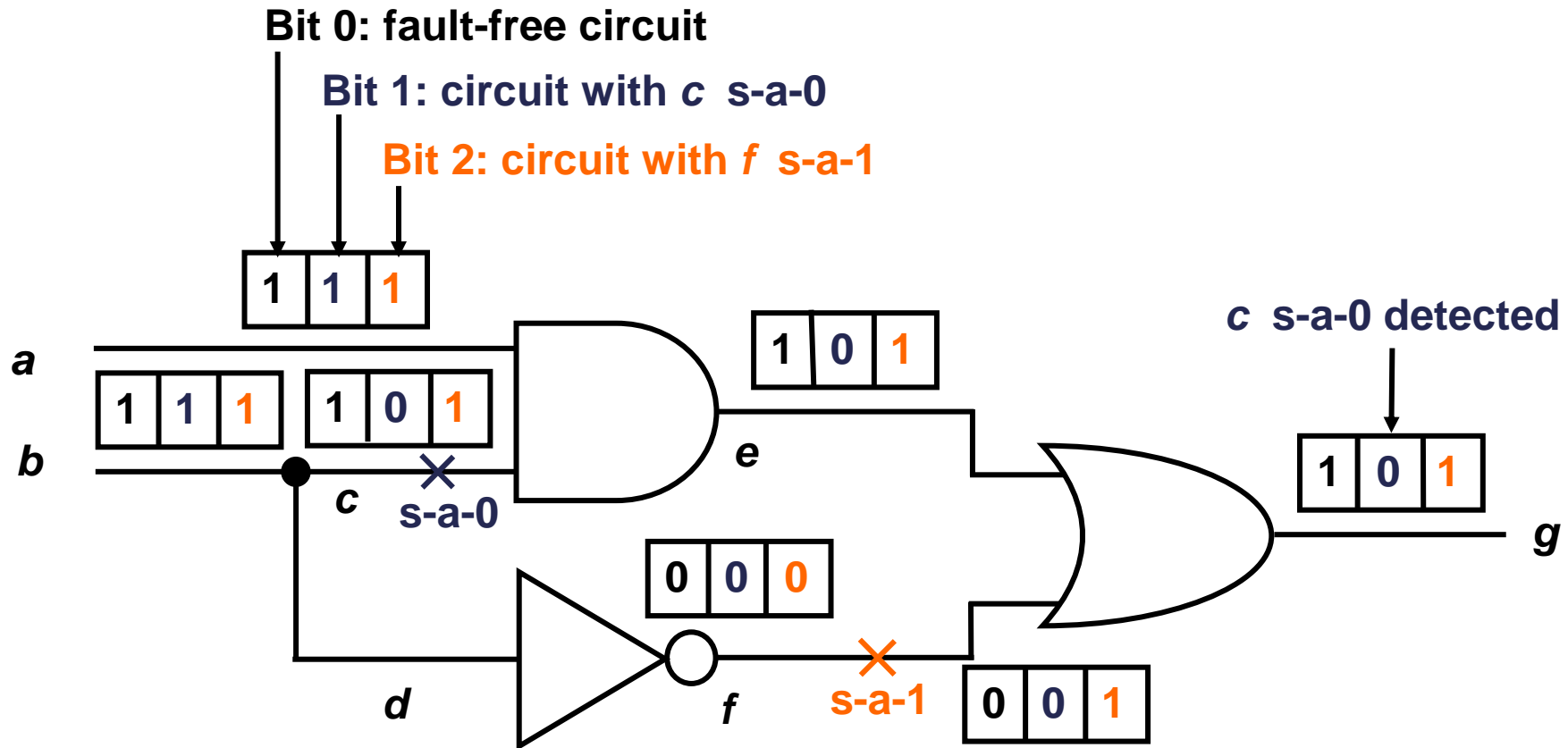
# Σειριακή προσομοίωση σφαλμάτων (4/4)

- Τερματισμός προσομοίωσης σφαλμάτων που έχουν ανιχνευθεί
- Παράδειγμα
  - Ας υποθέσουμε ότι θέλουμε να προσομοιώσουμε ένα σφάλμα  $f$  και χρησιμοποιούμε 3 διανύσματα δοκιμής  $P1, P2, P3$
  - Το σφάλμα  $f$  ανιχνεύεται από το διάνυσμα δοκιμής  $P1$
  - Δεν χρησιμοποιούνται τα  $P2, P3$  για να προσομοιώσουμε το σφάλμα  $f$
- Τα περισσότερα σφάλματα ανιχνεύονται όταν εφαρμόσουμε έναν σχετικά μικρό αριθμό από διανύσματα δοκιμής

# Παράλληλη προσομοίωση σφαλμάτων (1/2)

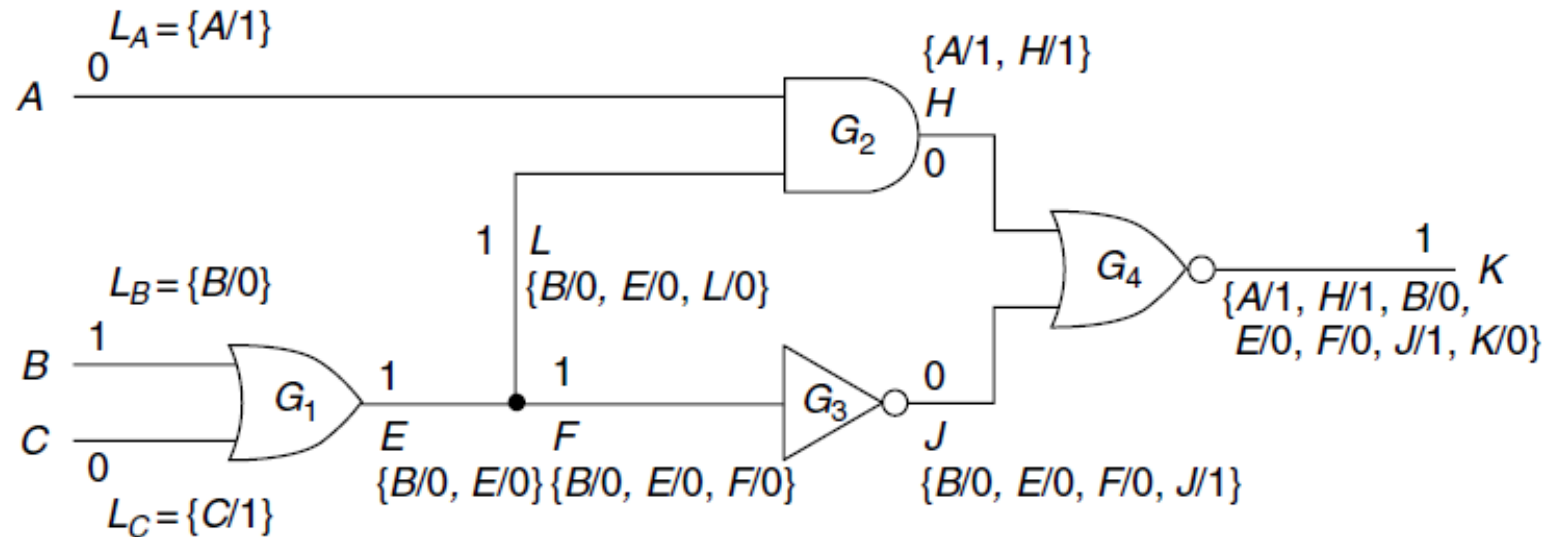
- Χρησιμοποιούνται λέξεις δεδομένων εύρους  $w$ -bits
  - Χρησιμοποιούνται  $w-1$  bits για τα faulty κυκλώματα
  - Χρησιμοποιείται 1 bit για το fault-free κύκλωμα
- Επεξεργάζονται παράλληλα τα faulty and fault-free κυκλώματα χρησιμοποιώντας bitwise λογικές λειτουργίες

# Παράλληλη προσομοίωση σφαλμάτων (2/2)



# Επαγωγική προσομοίωση σφαλμάτων (1/5)

- Καθορίζει ποια σφάλματα είναι ανιχνεύσιμα από ένα συγκεκριμένο διάνυσμα ελέγχου
- Κάθε γραμμή  $k$  του κυκλώματος περιέχει μία λίστα σφαλμάτων  $L_k$  τα οποία κάνουν την γραμμή  $k$  να διαφέρει από την fault-free τιμή της



# Επαγωγική προσομοίωση σφαλμάτων (2/5)

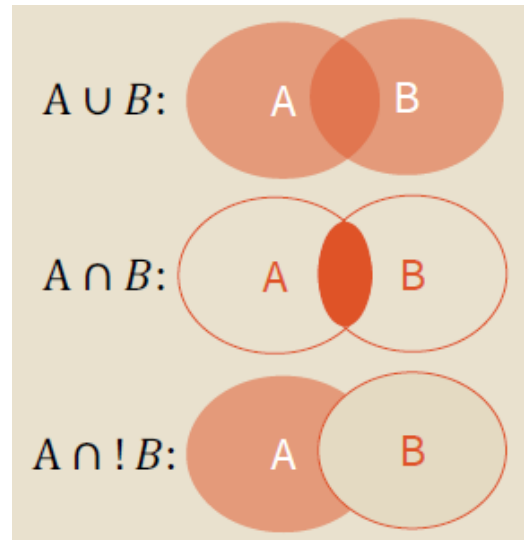
- Στηριζόμενοι στην συλλογιστική της λογικής (logic reasoning), η διαδικασία εξαγωγής της λίστας σφαλμάτων της εξόδου μιας πύλης από τις τιμές των εισόδων της ονομάζεται διάδοση λίστας σφαλμάτων (fault list propagation)
- Για παράδειγμα, η λίστα σφαλμάτων της γραμμής εξόδου  $E$  της πύλης  $G_1$  είναι η ένωση της λίστας σφαλμάτων της γραμμής  $B$  και του σφάλματος  $E/0$
- Προφανώς το σφάλμα  $E/0$  πρέπει να συμπεριληφθεί στην λίστα σφαλμάτων  $L_E$  καθώς η σωστή τιμή της γραμμής  $E$  είναι 1
- Καθώς η fault-free τιμή της γραμμής  $C$  (το 0) αποτελεί μία non-controlling τιμή της πύλης  $G_1$ , το αποτέλεσμα κάθε σφάλματος της λίστας  $L_B$  θα μεταδοθεί στο  $E$
- Οπότε όλα τα σφάλματα της λίστας  $L_B$  μεταδίδονται στην λίστα  $L_E$
- Τα σφάλματα της λίστας  $L_C$  δεν μεταδίδονται στην έξοδο της πύλης, καθώς η είσοδος  $B$  έχει την τιμή ελέγχου της εξόδου της πύλης  $G_1$

# Επαγωγική προσομοίωση σφαλμάτων (3/5)

- Η λίστα σφαλμάτων στην έξοδο κάθε πύλης ενημερώνεται χρησιμοποιώντας κανόνες συνόλων (set-theoretic rules) και την λίστα σφαλμάτων στις εισόδους της πύλης
- Η λίστα σφαλμάτων στην κύρια έξοδο καθορίζει τα σφάλματα που μπορούν να ανιχνευθούν από ένα διάνυσμα δοκιμής που εφαρμόζεται στις κύριες εισόδους του κυκλώματος
- Περιορισμοί:
  - Η χρήση κανόνων συνόλων είναι δύσκολο να εφαρμοστούν σε non-Boolean κυκλώματα
  - Η εφαρμογή της επαγωγικής προσομοίωσης είναι δύσκολη στην περίπτωση σφαλμάτων καθυστέρησης

# Επαγωγική προσομοίωση σφαλμάτων (4/5)

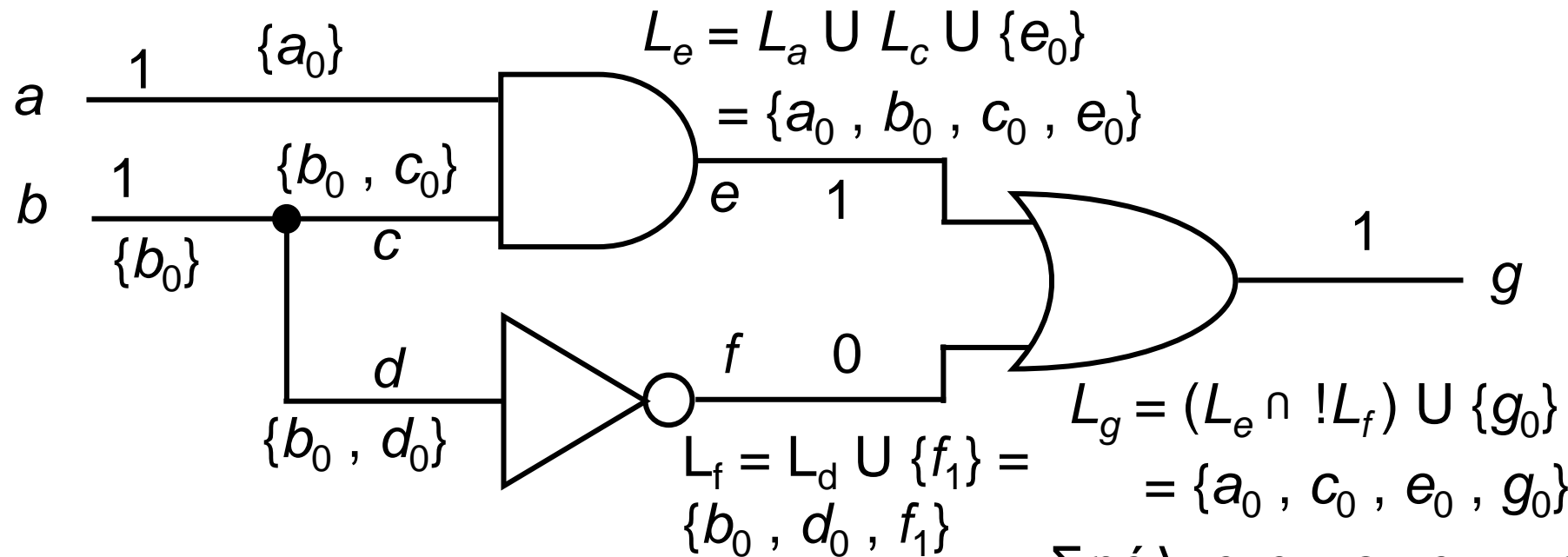
Κανόνες ανίχνευσης σφαλμάτων σε λογικές πύλες



Gate Type	Inputs (In1 and In2)		Output (O)	Deductive Fault list at O
AND	0	0	0	$[L_{In1} \cap L_{In2}] \cup O_1$
	0	1	0	$[L_{In1} \cap \overline{L_{In2}}] \cup O_1$
	1	0	0	$[\overline{L_{In1}} \cap L_{In2}] \cup O_1$
	1	1	1	$[L_{In1} \cup L_{In2}] \cup O_0$
OR (Dual of AND gate)	0	0	0	$[L_{In1} \cup L_{In2}] \cup O_1$
	0	1	1	$[\overline{L_{In1}} \cap L_{In2}] \cup O_0$
	1	0	1	$[L_{In1} \cap \overline{L_{In2}}] \cup O_0$
	1	1	1	$[L_{In1} \cap L_{In2}] \cup O_0$
NOT	0	----	1	$[L_{In1}] \cup O_0$
	1	-----	0	$[L_{In1}] \cup O_1$
Fanout	0	----	0	$[L_{In1}] \cup O_1$
	1	-----	1	$[L_{In1}] \cup O_0$

# Επαγωγική προσομοίωση σφαλμάτων (5/5)

Notation:  $L_k$  is fault list for line  $k$   
 $k_n$  is s-a-n fault on line  $k$



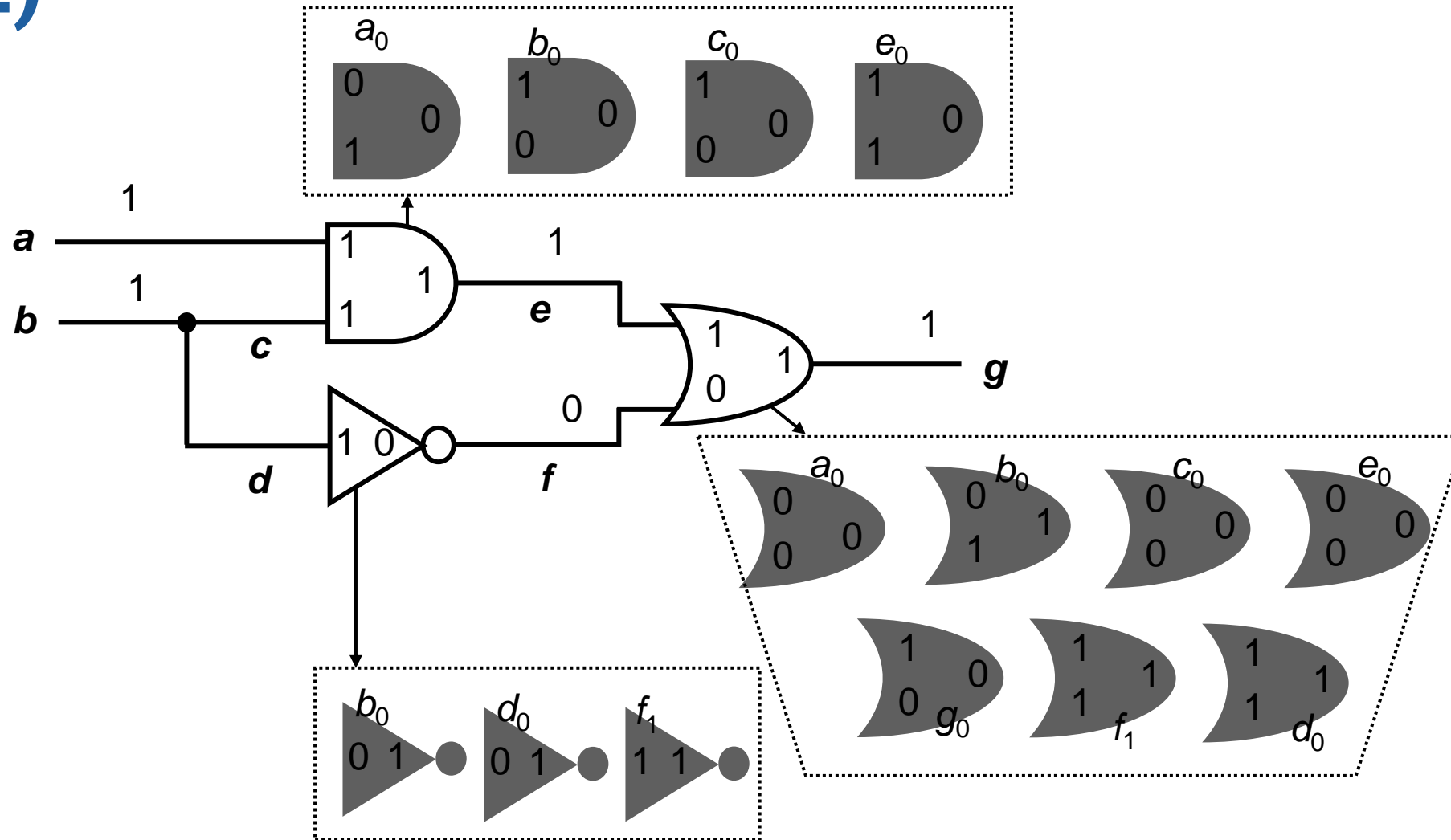
Σφάλματα που ανιχνεύονται  
 από το διάνυσμα δοκιμής 11



# Ταυτόχρονη προσομοίωση σφαλμάτων (1/2)

- Event-driven προσομοίωση του fault-free κυκλώματος και μόνο για εκείνα τα μέρη του faulty κυκλώματος τα οποία διαφέρουν ως προς τις καταστάσεις των σημάτων από το fault-free κύκλωμα
- Για κάθε πύλη δημιουργείται μία λίστα πυλών η οποία επηρεάζεται από κάποιο σφάλμα στην πύλη
- Γρηγορότερη από τις άλλες μεθόδους αλλά χρησιμοποιείται περισσότερη μνήμη

# Ταυτόχρονη προσομοίωση σφαλμάτων (2/2)



# Σύγκριση τεχνικών προσομοίωσης σφαλμάτων

- Ταχύτητα
  - Σειριακή προσομοίωση σφαλμάτων: η πιο αργή
  - Παράλληλη προσομοίωση σφαλμάτων:  $O(n^3)$  όπου  $n$  το πλήθος των πυλών
  - Επαγωγική προσομοίωση σφαλμάτων:  $O(n^2)$  όπου  $n$  το πλήθος των πυλών
  - Ταυτόχρονη προσομοίωση σφαλμάτων: γρηγορότερη από την επαγωγική
- Χρήση μνήμης
  - Σειριακή προσομοίωση σφαλμάτων, παράλληλη προσομοίωση σφαλμάτων: κανένα πρόβλημα
  - Επαγωγική προσομοίωση σφαλμάτων: δυναμική εκχώρηση μνήμης και δυσκολία στην πρόβλεψη του μεγέθους μνήμης που θα απαιτηθεί
  - Ταυτόχρονη προσομοίωση σφαλμάτων: μεγαλύτερες προκλήσεις σε σχέση με την επαγωγική

# Σύγκριση τεχνικών προσομοίωσης σφαλμάτων

- Συνδυαστικά κυκλώματα
  - Η παράλληλη εξομοίωση σφαλμάτων αποτελεί την πιο διαδεδομένη τεχνική
- Ακολουθιακά κυκλώματα
  - Σειριακή, παράλληλη, ταυτόχρονη προσομοίωση σφαλμάτων: κανένα πρόβλημα
  - Επαγωγική προσομοίωση σφαλμάτων: δύσκολη εξαιτίας των άγνωστων καταστάσεων του κυκλώματος

# Εναλλακτικές τεχνικές στην προσομοίωση σφαλμάτων

- Toggle Coverage
- Fault Sampling
- Critical Path Tracing
- Statistical Fault Analysis

# Toggle Coverage

- Δημοφιλής τεχνική για την αξιολόγηση της ποιότητας των προτύπων δοκιμής επειδή απαιτεί μόνο μία fault-free προσομοίωση
- Net: ένα ηλεκτρικό μονοπάτι σε ένα κύκλωμα το οποίο έχει την ίδια τιμή σε όλα τα σημεία του
- A net is toggled if
  - Relaxed definition: its value has been set to zero and one during fault-free simulation
  - Stringent definition: it has both a zero-to-one transition and a one-to-zero transition during fault-free simulation
- Toggle coverage: number of toggled nets / number of total nets in the circuit

# Fault Sampling (1/2)

- Προσομοιώνεται ένα τυχαία επιλεγμένο υποσύνολο σφαλμάτων
- Η κάλυψη σφαλμάτων του δείγματος χρησιμοποιείται για την εκτίμηση της κάλυψης σφαλμάτων σε όλο το κύκλωμα
- Πλεονέκτημα
  - Εξοικονόμηση υπολογιστικών πόρων (χρόνος CPU και μνήμη)
- Μειονέκτημα: Περιορισμένα δεδομένα σχετικά με τα μη ανιχνεύσιμα σφάλματα

# Fault Sampling (2/2)

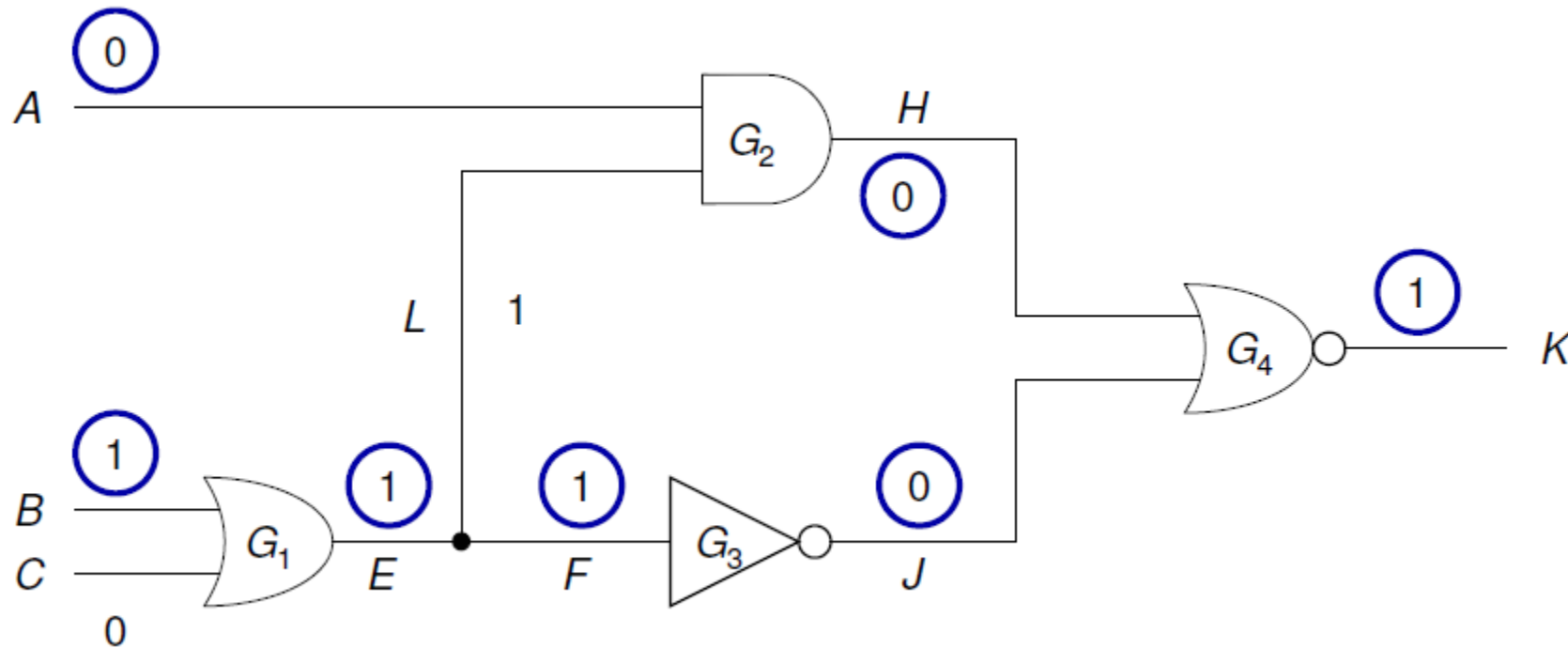
- Η πολυπλοκότητα της προσομοίωσης σφαλμάτων εξαρτάται από:
  - Το πλήθος των λογικών πυλών
  - Το πλήθος των σφαλμάτων
  - Το πλήθος των διανυσμάτων δοκιμής
- Η πολυπλοκότητα της προσομοίωσης σφαλμάτων με δειγματοληψία εξαρτάται από:
  - Το πλήθος των λογικών πυλών
  - Το πλήθος των διανυσμάτων δοκιμής



# Critical Path Tracing (1/2)

- Δεδομένου ενός διανύσματος δοκιμής  $t$ , μία γραμμή  $x$  έχει μία κρίσιμη τιμή  $v$ , αν και μόνο αν το σφάλμα  $x$  stuck at  $v$  ανιχνεύεται από το  $t$
- Κρίσιμο μονοπάτι: Μονοπάτι που περιλαμβάνει κρίσιμες τιμές
- Το κρίσιμο μονοπάτι γίνεται trace από τις κύριες εξόδους προς τις κύριες εισόδους δίνοντας την λίστα των σφαλμάτων που ανιχνεύονται

# Critical Path Tracing (2/2)



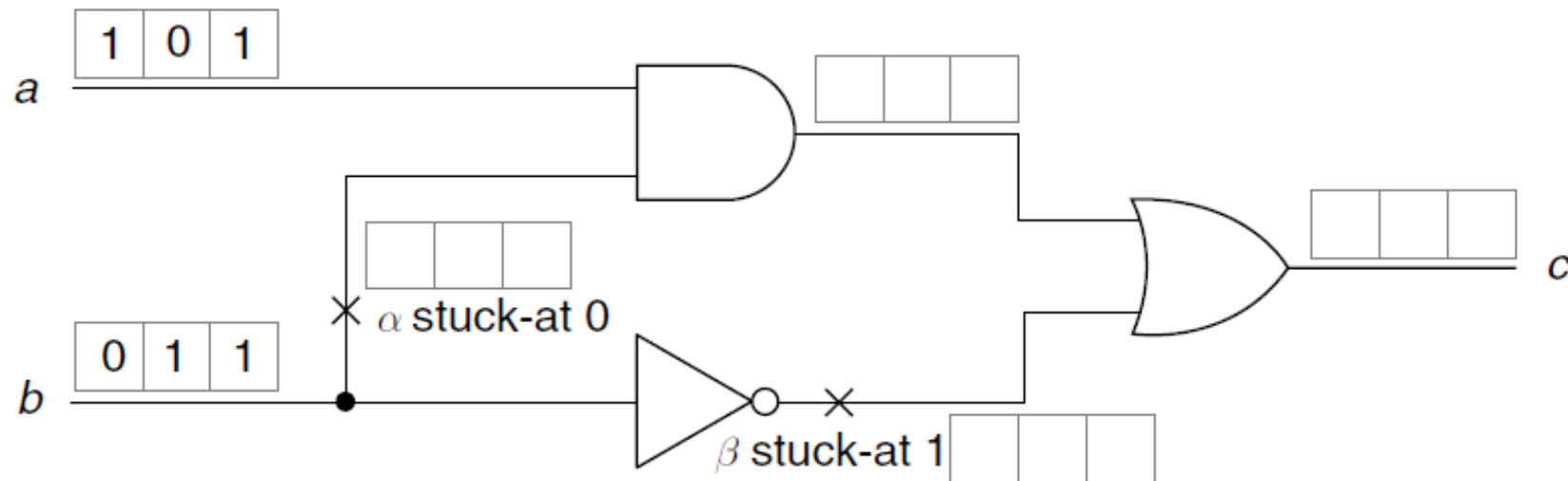
- Τα  $L$  και  $C$  δεν αποτελούν κρίσιμες τιμές καθώς αν αλλάξουν δεν αλλάζει η έξοδος του κυκλώματος
- Ανιχνεύονται τα σφάλματα  $A/1$ ,  $H/1$ ,  $B/0$ ,  $E/0$ ,  $F/0$ ,  $J/1$ ,  $K/0$

# Statistical Fault Analysis

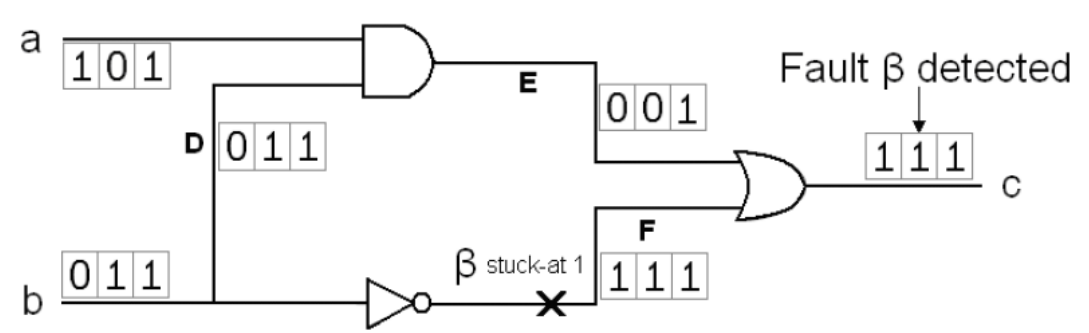
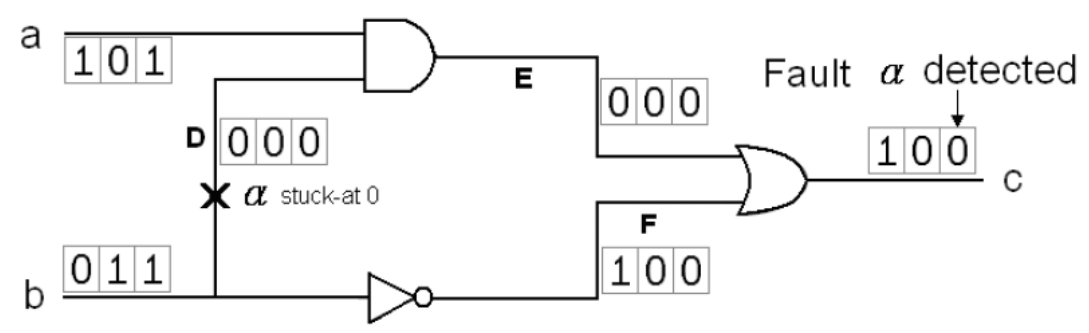
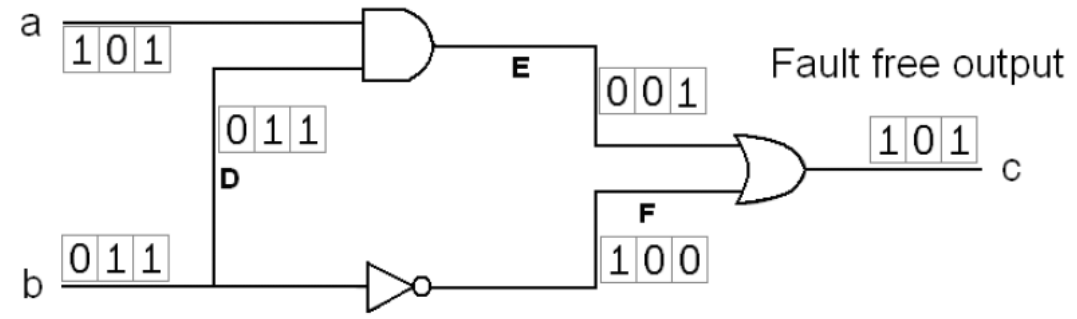
- Χρησιμοποιείται θεωρία πιθανοτήτων για να εκτιμηθεί η αναμενόμενη τιμή της κάλυψης σφαλμάτων
- Ανιχνευσιμότητα ενός σφάλματος  $f$  σε ένα net  $x$ 
  - 1-controllability,  $C1(x)$ : πιθανότητα να τεθεί το net  $x$  στην τιμή 1 από ένα τυχαίο πρότυπο δοκιμής
  - 0-controllability,  $C0(x)$ : πιθανότητα να τεθεί το net  $x$  στην τιμή 0 από ένα τυχαίο πρότυπο δοκιμής
  - Observability,  $O(x)$ : πιθανότητα κάποια δεδομένα πρότυπα δοκιμής να μεταδώσουν το αποτέλεσμα ενός σφάλματος στο net  $x$  στις κύριες εξόδους του κυκλώματος
  - Sensitization probability,  $S(x)$ : sensitization count of gate input  $x$  / number of test patterns

# Άσκηση 1

- Χρησιμοποιήστε την τεχνική παράλληλης προσομοίωσης σφαλμάτων για να καθορίσετε ποια από τα 2 σφάλματα ( $\alpha/0$  και  $\beta/1$ ) ανιχνεύονται από τα πρότυπα δοκιμής στο παρακάτω κύκλωμα



# Άσκηση 1 - Λύση



# Πηγές

- Laung-Terng Wang, Cheng-Wen Wu and Xiaoqing Wen, VLSI Test Principles and Architectures, Springer, 2006.

# Ερωτήσεις

