

# Υπολογιστικά Συστήματα Υψηλής Αξιοπιστίας

## Έλεγχος Ορθής Λειτουργίας Ολοκληρωμένων Κυκλωμάτων

Δρ. Γκάμας Βασίλειος

Επιστημονικός Συνεργάτης  
vgkamas@uniwa.gr

Πανεπιστήμιο Δυτικής Αττικής  
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

# Σκοπός παρουσίασης

- Να παρουσιάσει θέματα που σχετίζονται με τον έλεγχο ορθής λειτουργίας ολοκληρωμένων κυκλωμάτων
- Προκλήσεις στην παραγωγή των δοκιμών (test generation) και στην μοντελοποίηση των σφαλμάτων (fault modelling)

# Ανασκόπηση βασικών εννοιών ψηφιακής σχεδίασης

# Άλγεβρα Boole

$$(a) \quad x + 0 = x$$

$$(a) \quad x + x' = 1$$

$$(a) \quad x + x = x$$

$$(a) \quad x + 1 = 1$$

$$(x')' = x$$

$$(a) \quad x + y = y + x$$

$$(a) \quad x + (y + z) = (x + y) + z$$

$$(a) \quad x(y + z) = xy + xz$$

$$(a) \quad (x + y)' = x'y'$$

$$(a) \quad x + xy = x$$

$$(b) \quad x \cdot 1 = x$$

$$(b) \quad x \cdot x' = 0$$

$$(b) \quad x \cdot x = x$$

$$(b) \quad x \cdot 0 = 0$$

$$(b) \quad xy = yx$$

$$(b) \quad x(yz) = (xy)z$$

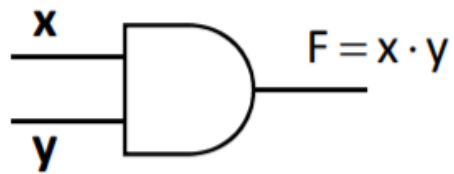
$$(b) \quad x + yz = (x + y)(x + z)$$

$$(b) \quad (xy)' = x' + y'$$

$$(b) \quad x(x + y) = x$$

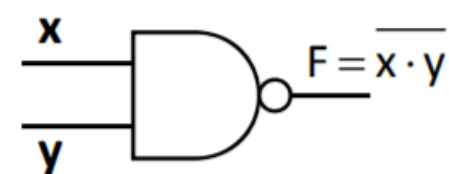
# Λογικές πύλες

**AND**



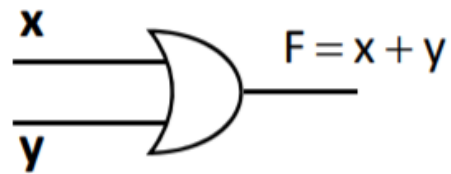
x	y	F
0	0	0
0	1	0
1	0	0
1	1	1

**NAND**



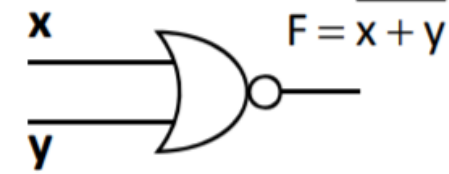
x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

**OR**



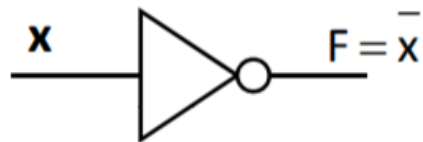
x	y	F
0	0	0
0	1	1
1	0	1
1	1	1

**NOR**



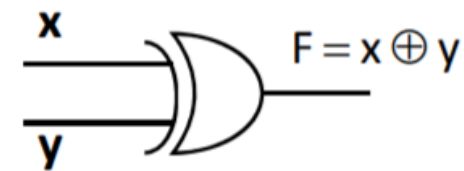
x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

**NOT**



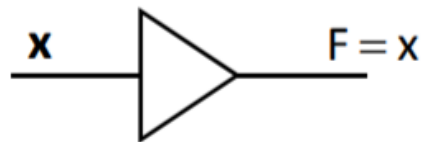
x	F
0	1
1	0

**XOR**



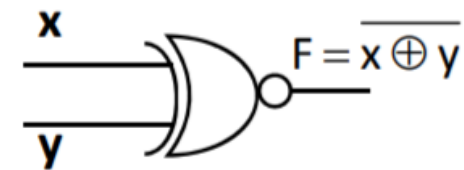
x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

**BUFFER**



x	F
0	0
1	1

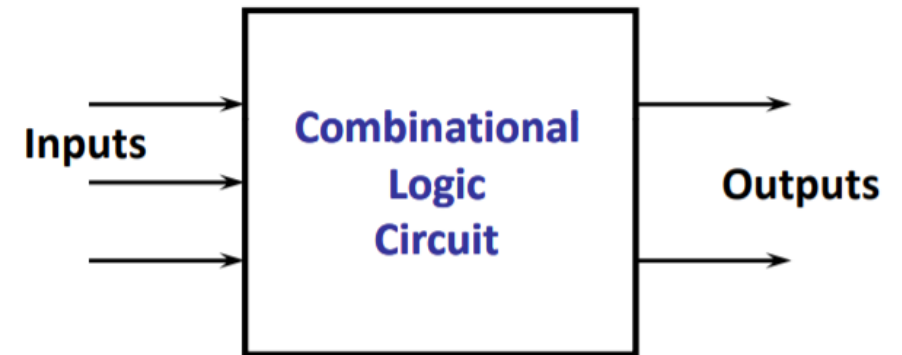
**XNOR**



x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

# Συνδυαστικά κυκλώματα

- Συντίθεται από λογικές πύλες
- Ανά πάσα χρονική στιγμή, οι έξοδοι καθορίζονται μόνο από τον τρέχοντα συνδυασμό εισόδων
- Εκτελεί μία λειτουργία που μπορεί να προσδιοριστεί λογικά από ένα σύνολο συναρτήσεων Boole

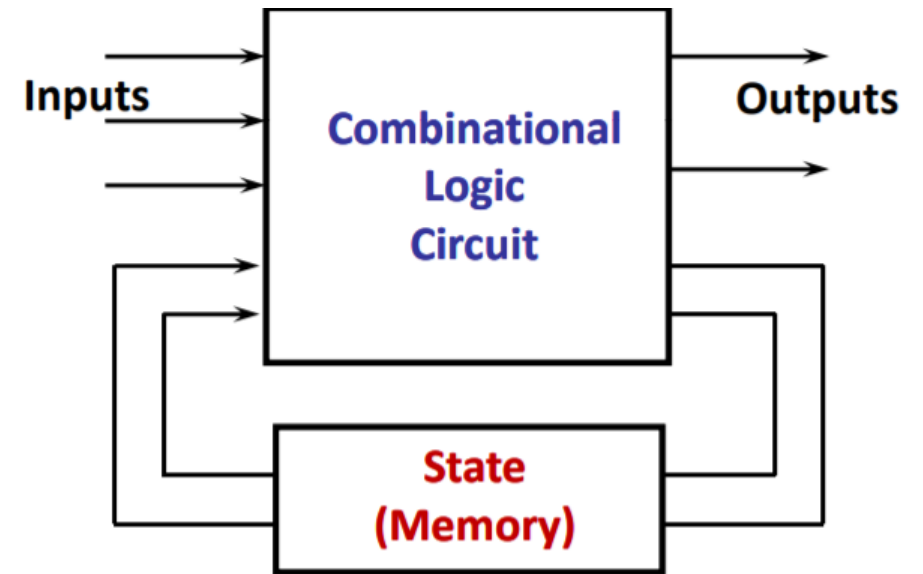


***Combinational Logic***

$$\text{outputs} = f(\text{inputs})$$

# Ακολουθιακά κυκλώματα

- Επιπρόσθετα των λογικών πυλών υπάρχουν και στοιχεία μνήμης
- Οι έξοδοι εξαρτώνται όχι μόνο από τις εισόδους, αλλά και από την κατάσταση των στοιχείων μνήμης
- Η κατάσταση των στοιχείων μνήμης εξαρτάται από προηγούμενες εισόδους

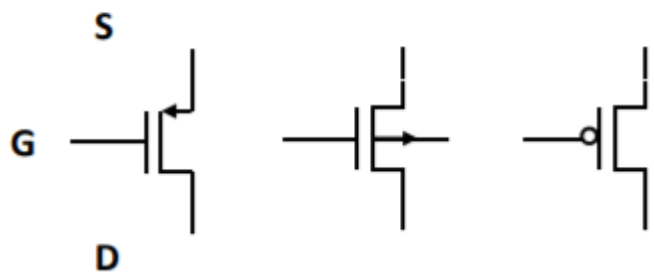


***Sequential Logic***

$$\text{outputs} = f(\text{inputs}, \text{state})$$

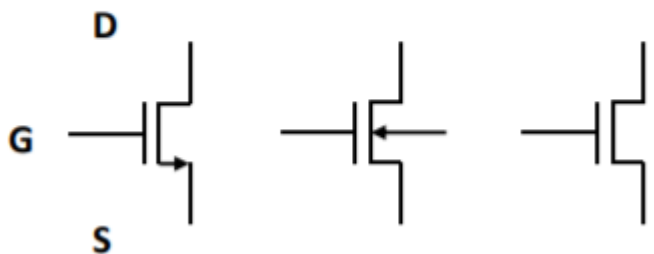
# Λογική CMOS

## pMOS transistor



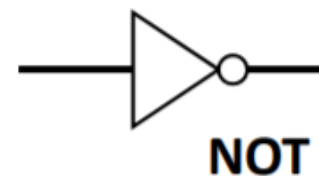
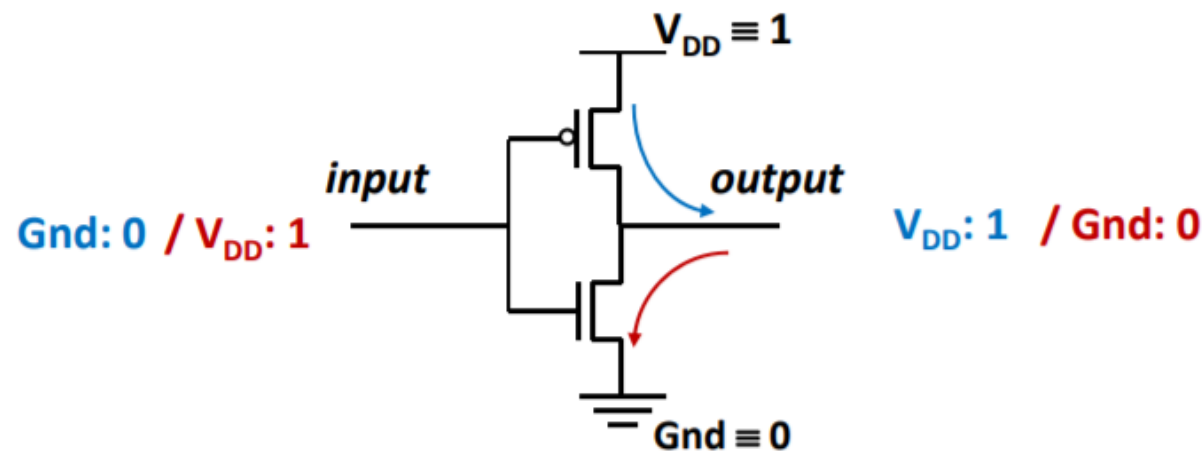
0 → άγει  
1 → δεν άγει

## nMOS transistor



0 → δεν άγει  
1 → άγει

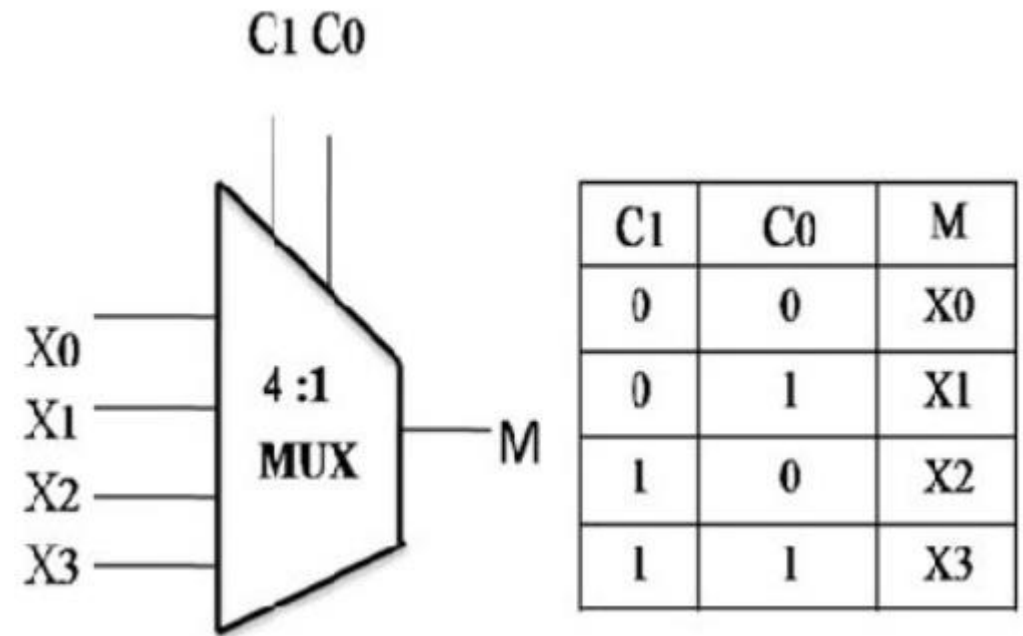
## Inverter – NOT Gate





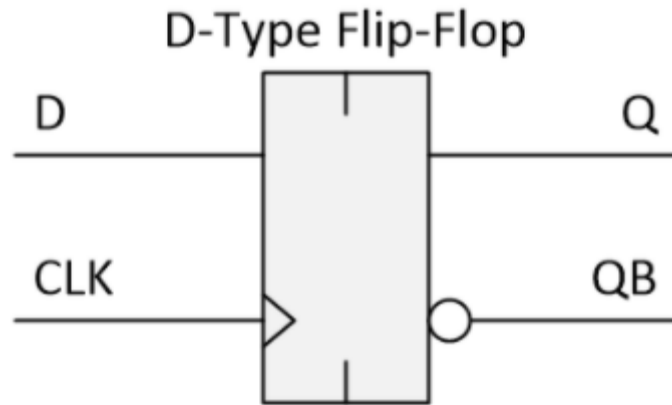
# Πολυπλέκτης

- Συνδυαστικό κύκλωμα το οποίο επιλέγει τη δυαδική πληροφορία μίας από πολλές γραμμές εισόδου και την κατευθύνει σε μία μοναδική γραμμή εξόδου.
- Η επιλογή της συγκεκριμένης γραμμής εξόδου ελέγχεται από ειδικές γραμμές επιλογής.
- Κανονικά, υπάρχουν  $2^n$  γραμμές εισόδου και  $n$  γραμμές επιλογής, των οποίων οι τιμές καθορίζουν ποια είσοδος επιλέγεται.



# Flip Flops

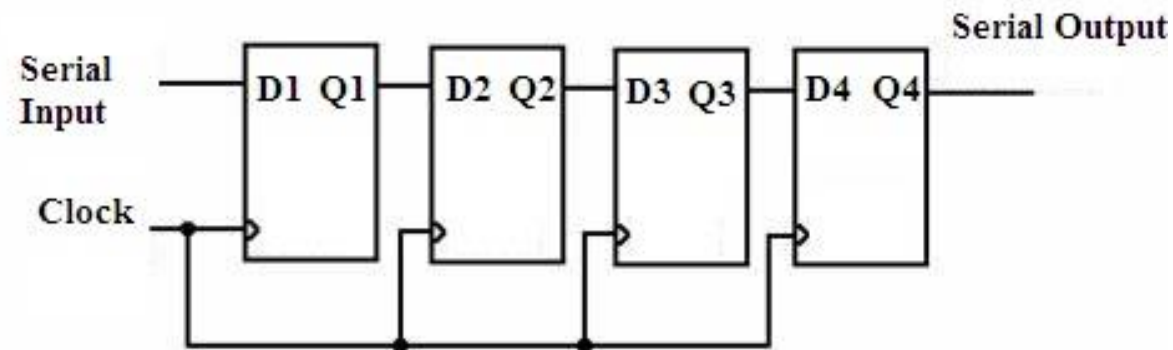
- Στοιχεία μνήμης μπορεί να είναι του ενός ψηφίου όπως είναι το flip-flop ή ακολουθία από flips-flops όπως εμφανίζονται σε έναν καταχωρητή ολίσθησης (shift register).
- Το βασικό χαρακτηριστικό σε κάθε flip-flop είναι ότι έχει δύο σταθερές καταστάσεις που παρουσιάζονται με το λογικό '0' ή '1' αντίστοιχα.



CLK	D	$Q_{(new)}$	$QB_{(new)}$
↑	0	0	1
↑	1	1	0
0	X	$Q_{(old)}$	$QB_{(old)}$
1	X	$Q_{(old)}$	$QB_{(old)}$
↓	X	$Q_{(old)}$	$QB_{(old)}$

# Καταχωρητής ολίσθησης

- Αλληλουχία από flip-flops που μοιράζονται το ίδιο ρολόι χρονισμού. Η έξοδος του κάθε flip flop συνδέεται στην είσοδο του επόμενου στην ακολουθία αυτή
- Το κύκλωμα που προκύπτει, μετατοπίζει κατά μία θέση, σε κάθε παλμό ρολογιού, την ακολουθία από bits που αποθηκεύτηκε αρχικά στα αντίστοιχα flip-flops της αλυσίδας αυτής



Serial in serial out 4-bit Shift Register

# Έλεγχος Ορθής Λειτουργίας Ολοκληρωμένων Κυκλωμάτων

# Εισαγωγή

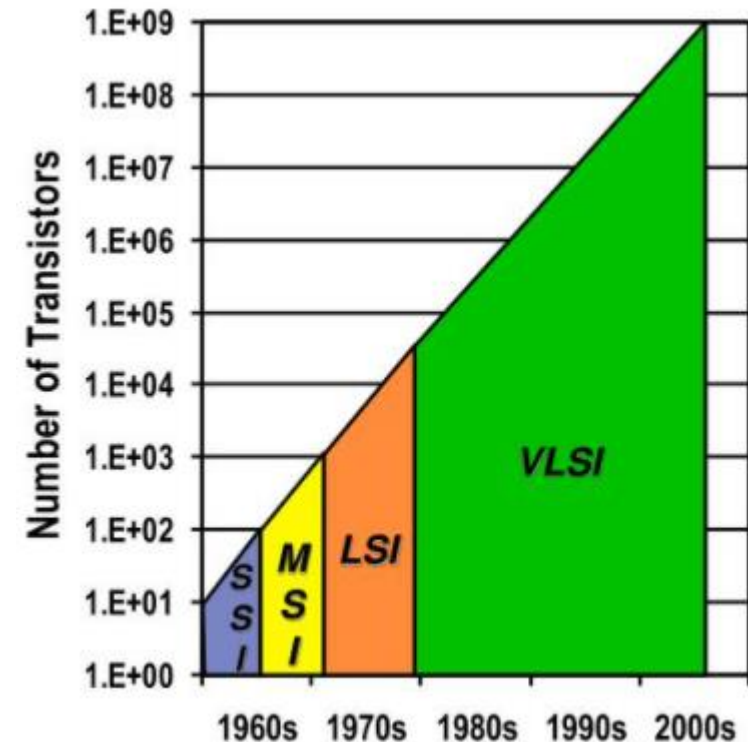
- Οποιαδήποτε ανθρώπινη κατασκευή από την πιο απλή ως την πλέον πολυσύνθετη είναι δυνατόν να παρουσιάσει εσφαλμένη λειτουργία η οποία μπορεί να οφείλεται σε διάφορους παράγοντες
- Κάποιες συσκευές παρουσιάζουν ελαττώματα που εισήχθησαν κατά τη διάρκεια της κατασκευαστικής διαδικασίας ενώ βλάβες μπορεί να παρουσιαστούν και κατά τη διάρκεια χρήσης τους

# Εισαγωγή

- Για να καθοριστεί αν μια συσκευή έχει κατασκευαστεί σωστά ή ότι συνεχίζει να λειτουργεί σύμφωνα με τον επιθυμητό τρόπο πρέπει να περάσει από δοκιμή (test) δηλαδή την διαδικασία επαλήθευσης της σωστής λειτουργίας της που ονομάζεται έλεγχος ορθής λειτουργίας
- Ανάλογα με την πολυπλοκότητα της συσκευής ο έλεγχος ορθής λειτουργίας μπορεί να είναι πολύ απλός και γρήγορος ή μια χρονοβόρα διαδικασία επαλήθευσης ενός πολύπλοκου συστήματος, ώστε να εξασφαλιστεί ότι το σύστημα πληροί τα καθορισμένα κριτήρια απόδοσης, αξιοπιστίας και ασφάλειας

# Εισαγωγή

- Από τα τέλη της δεκαετίας του '50, τα ολοκληρωμένα κυκλώματα (ICs) έχουν αναπτυχθεί ως προς το μέγεθος και την πολυπλοκότητά τους
- Σύμφωνα με τον νόμο του Moore, ο βαθμός ολοκλήρωσης των ICs διπλασιάζεται κάθε 18 μήνες
- Η αύξηση του μεγέθους και της πολυπλοκότητας δημιουργεί πολλές και νέες προκλήσεις αναφορικά με τον έλεγχο ορθής λειτουργίας



# Σημασία της δοκιμής σε IC (1/2)

- Ο Νόμος του Moore είναι αποτέλεσμα της μείωσης των διαστάσεων
  - Οι διαστάσεις των τρανζίστορ και των αγωγών διασύνδεσης έχουν μειωθεί από δεκάδες  $\mu m$  σε δεκάδες  $nm$
- Οι συχνότητες λειτουργίας έχουν αυξηθεί από 100kHz σε αρκετά GHz
- Η μείωση των διαστάσεων οδηγεί στην αύξηση της πιθανότητας εκδήλωσης ελαττωμάτων κατά τη διάρκεια της κατασκευής
  - Ένα και μόνο ελαττωματικό τρανζίστορ οδηγεί σε ελαττωματικό IC
  - Ο έλεγχος ορθής λειτουργίας είναι απαραίτητος για την εξασφάλιση προϊόντων χωρίς βλάβες

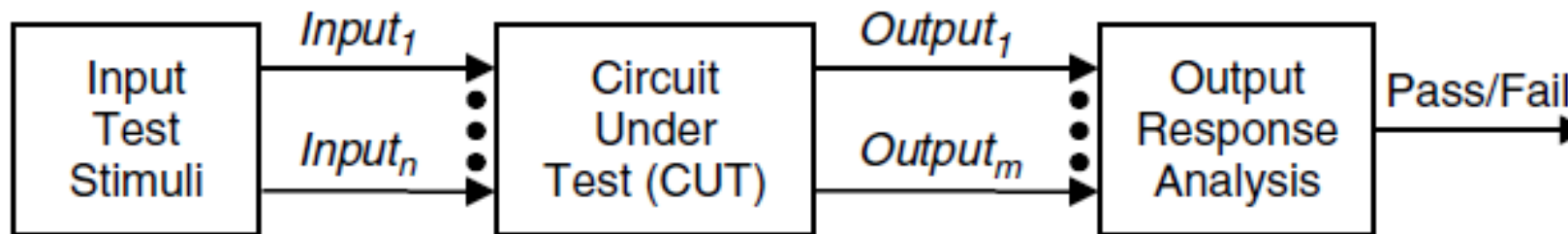


# Σημασία της δοκιμής σε IC (2/2)

- Κανόνας του Δέκα: Το κόστος ανίχνευσης ενός ελαττωματικού IC αυξάνεται κατά μία τάξη μεγέθους καθώς κινούμαστε ως εξής:
  - Διάταξη → Printed Circuit Board → Σύστημα → Λειτουργία
  - Σε όλα τα πιο πάνω στάδια πραγματοποιείται έλεγχος ορθής λειτουργίας
- Έλεγχος ορθής λειτουργίας πραγματοποιείται και κατά τη διάρκεια:
  - Της παραγωγής για τη βελτίωση της κατασκευαστικής απόδοσης
  - Της λειτουργίας για τη διασφάλιση ενός συστήματος απαλλαγμένου από σφάλματα
- Όταν ανιχνεύονται σφάλματα ενεργοποιείται η διαδικασία επιδιόρθωσης

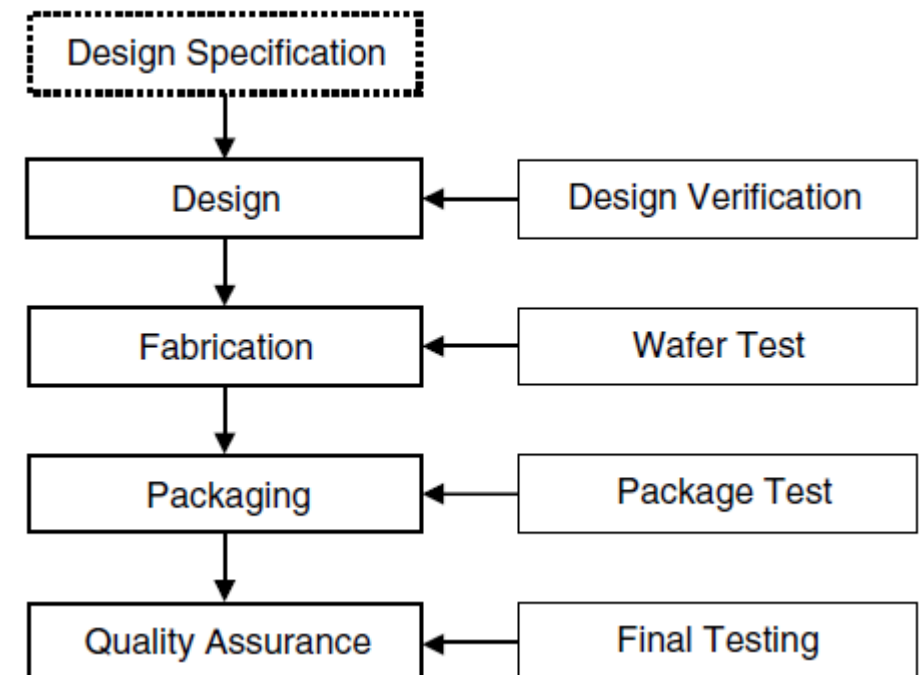
# Δοκιμή κατά τη διάρκεια του κύκλου ζωής

- Η δοκιμή συνίσταται από τα παρακάτω βήματα:
  - Εφαρμογή διεγέρσεων δοκιμής (test stimuli) στις εισόδους του υπό έλεγχο κυκλώματος
  - Ανάλυση των αποκρίσεων εξόδου
    - Αν οι αποκρίσεις είναι μη αναμενόμενες (fail), το υπό έλεγχο κύκλωμα θεωρείται ελαττωματικό
    - Αν οι αποκρίσεις είναι αναμενόμενες (pass), το υπό έλεγχο κύκλωμα θεωρείται απαλλαγμένο από σφάλματα



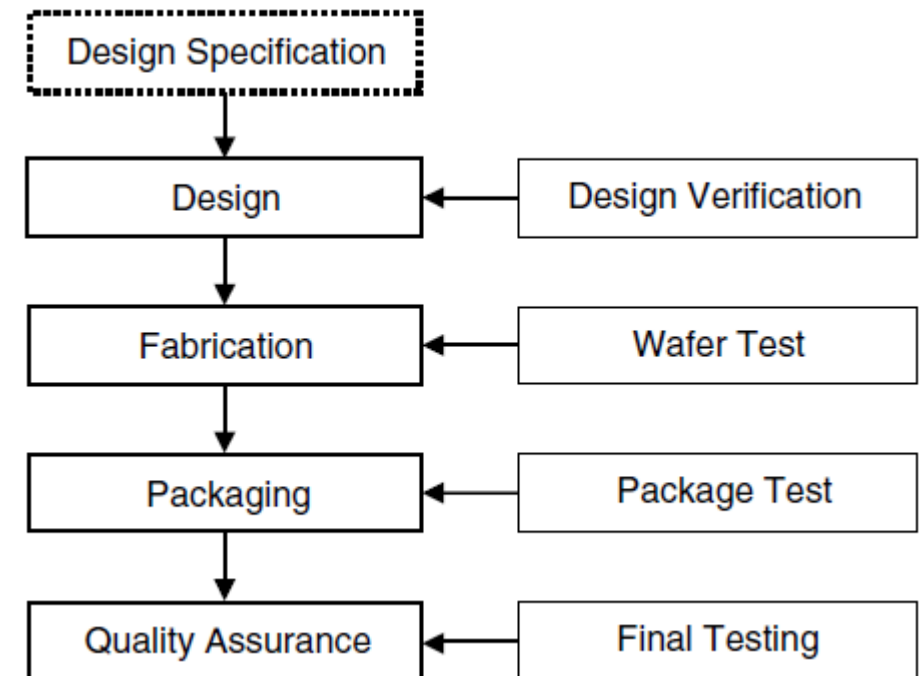
# Δοκιμή κατά τη διάρκεια της ανάπτυξης (1/5)

- Σκοπός της επιβεβαίωσης σχεδιασμού είναι να ανιχνευθούν λάθη που έγιναν κατά τη διαδικασία σχεδίασης ενός προϊόντος, γεγονός που θα είχε ως συνέπεια, αν αυτό κατασκευαστεί να μην λειτουργεί σύμφωνα με τις προδιαγραφές του.
- Σκοπός του ελέγχου μετά την κατασκευή είναι να ανιχνεύσει ελαττώματα που προέκυψαν κατά τη διαδικασία της κατασκευής, με συνέπεια αυτό να μη λειτουργεί σύμφωνα με τις προδιαγραφές του



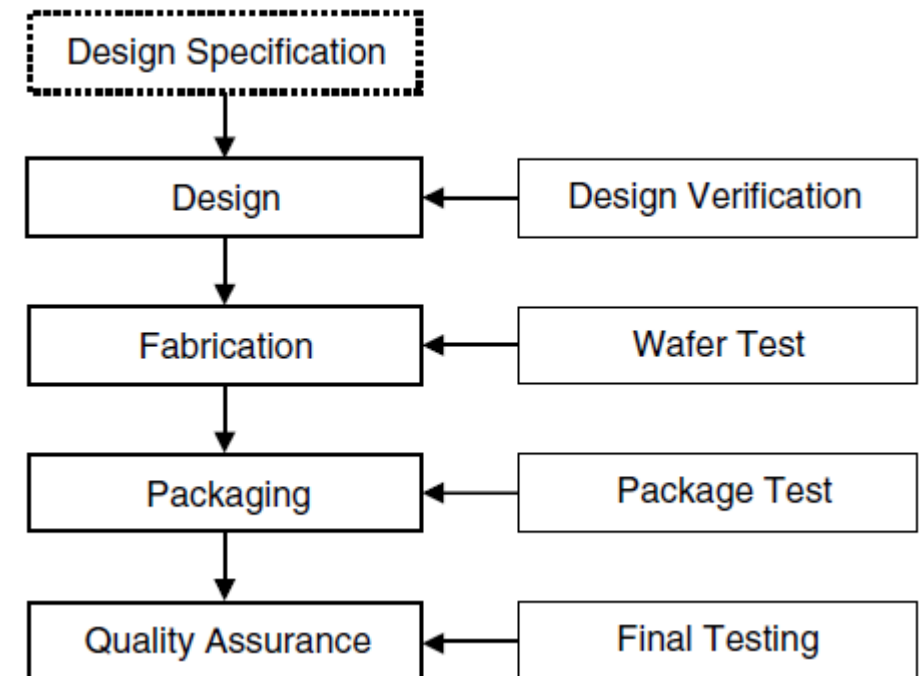
# Δοκιμή κατά τη διάρκεια της ανάπτυξης (2/5)

- Η επιβεβαίωση σχεδιασμού γίνεται πριν την κατασκευή του IC
- Δεν περιμένουμε να ολοκληρωθεί η σχεδίαση ενός IC και στη συνέχεια να κάνουμε την επιβεβαίωση σχεδιασμού
- Για κάθε υπομονάδα του συστήματος η οποία σχεδιάζεται θα πρέπει να επιβεβαιώνεται και η ορθή της λειτουργία, δηλαδή ότι λειτουργεί σύμφωνα με τις προδιαγραφές.
- Όσο πιο γρήγορα διαπιστώνεται το σχεδιαστικό λάθος τόσο μικρότερο είναι το κόστος διόρθωσής του.



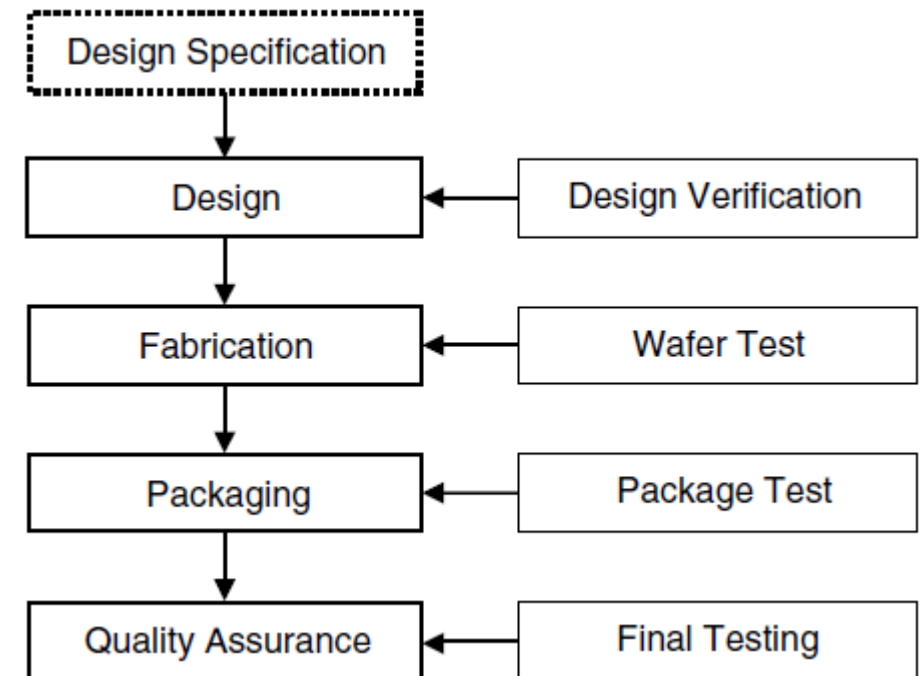
# Δοκιμή κατά τη διάρκεια της ανάπτυξης (3/5)

- Ο έλεγχος της ορθής λειτουργίας μετά την κατασκευή αποτελείται από διάφορους ελέγχους, οι οποίοι λαμβάνουν χώρα σε διάφορα βήματα της κατασκευαστικής διαδικασίας.
- Ο πρώτος έλεγχος λαμβάνει χώρα αμέσως μετά την κατασκευή των ολοκληρωμένων κυκλωμάτων πάνω στο δίσκο πυριτίου (wafer test).



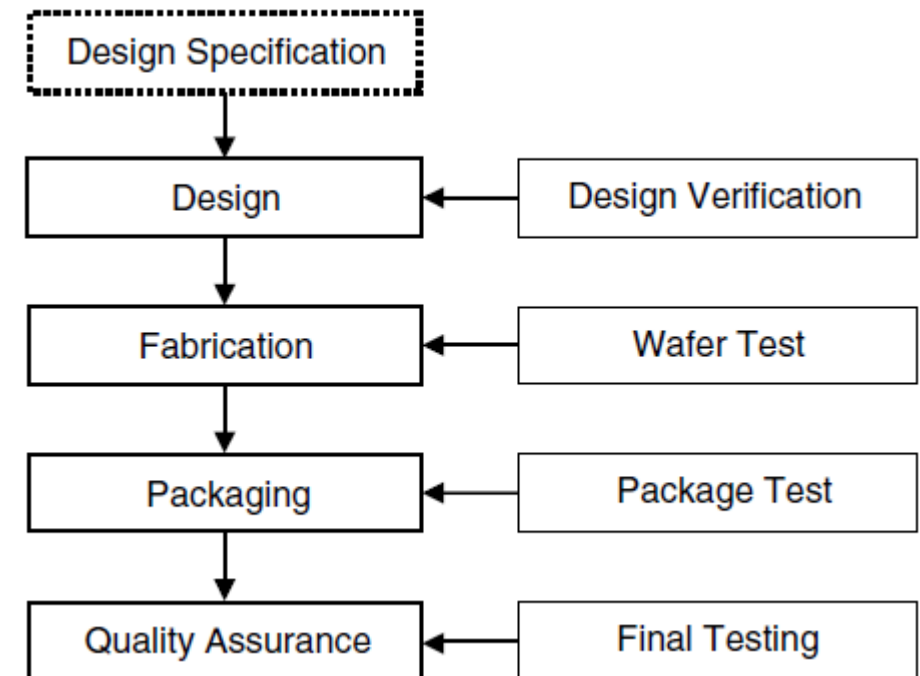
# Δοκιμή κατά τη διάρκεια της ανάπτυξης (4/5)

- Τα ελαττωματικά IC σημειώνονται και απομακρύνονται μετά το κόψιμο του δίσκου πυριτίου σε τμήματα εκ των οποίων κάθε ένα περιέχει και ένα IC.
- Κάποια από τα ελαττωματικά IC οδηγούνται στο τμήμα διαγνωστικού ελέγχου της εταιρίας, για να βρεθούν οι αιτίες που προκάλεσαν τα ελαττώματα κατά τη διαδικασία κατασκευής και να γίνουν οι απαιτούμενες διορθώσεις στη διαδικασία κατασκευής ώστε να μειωθεί ο αριθμός των ελαττωματικών IC που κατασκευάζονται.



# Δοκιμή κατά τη διάρκεια της ανάπτυξης (5/5)

- Κάθε ένα από τα IC που πέρασαν τον παραπάνω έλεγχο τοποθετείται μέσα σε ένα πλαστικό ή κεραμικό πακέτο και συνδέεται με τον απαιτούμενο αριθμό ακροδεκτών.
- Κατά τη διαδικασία αυτή είναι δυνατόν να καταστραφεί το IC ή να μη γίνει σωστή σύνδεση των ακροδεκτών του με τους ακροδέκτες του πακέτου.
- Γι' αυτό ακολουθεί έλεγχος της ορθής λειτουργίας του πακεταρισμένου ολοκληρωμένου κυκλώματος (chip).



# Βασικοί δείκτες - Yield

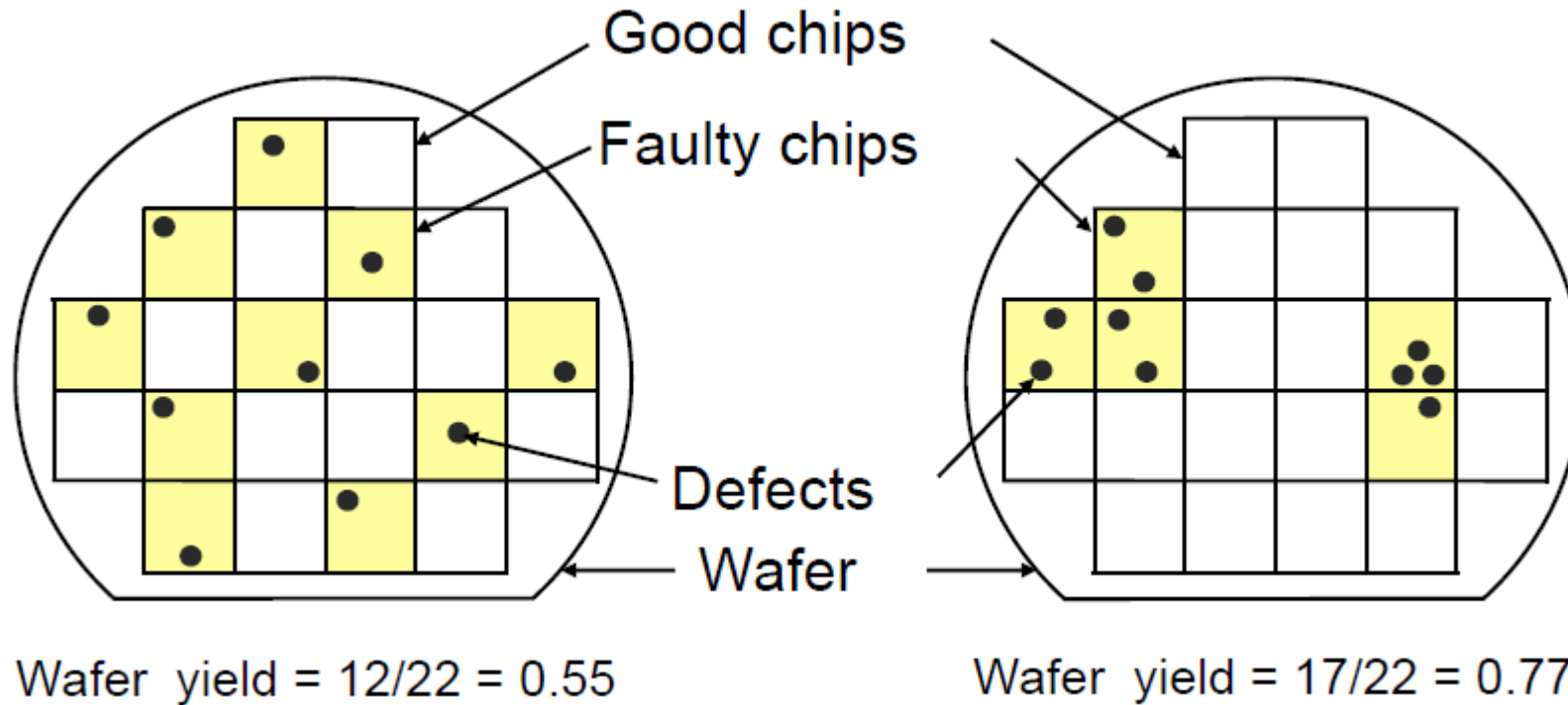
- Chips με σφάλματα λόγω κατασκευαστικών ελαττωμάτων
- Κατασκευαστική απόδοση (yield)

$$yield = \frac{\textit{number of acceptable parts}}{\textit{total number of parts fabricated}}$$

- Απώλεια κατασκευαστικής απόδοσης (yield loss)
  - Καταστροφική (catastrophic): εξαιτίας τυχαίων ελαττωμάτων
  - Παραμετρική (parametric): εξαιτίας διακυμάνσεων διεργασίας



# Βασικοί δείκτες - Yield



# Βασικοί δείκτες - Reject rate

- Chips με σφάλματα εμφανίζονται να μην έχουν πρόβλημα (περνούν τον έλεγχο ορθής λειτουργίας)
- Λόγος απόρριψης (reject rate)

$$\text{reject rate} = \frac{\text{number of faulty parts passing final test}}{\text{total number of parts passing final test}}$$

# Παραγωγή Δοκιμής (1/3)

- Για την πραγματοποίηση της δοκιμής χρησιμοποιείται μια ακολουθία προτύπων δοκιμής (test patterns) τα οποία ονομάζονται διανύσματα δοκιμής (test vectors).
- Τα διανύσματα δοκιμής εφαρμόζονται στο υπό έλεγχο κύκλωμα του οποίου οι έξοδοι παρακολουθούνται και αναλύονται για τη σωστή απόκριση
- Η κάλυψη σφαλμάτων (fault coverage) είναι μία ποσοτική παράμετρος μέτρησης της ποιότητας των διανυσμάτων δοκιμής

## Παραγωγή Δοκιμής (2/3)

- Για ένα δεδομένο σύνολο διανυσμάτων δοκιμής, η κάλυψη σφαλμάτων ορίζεται ως εξής

$$\text{fault coverage} = \frac{\text{number of detected faults}}{\text{total number of faults}}$$

- Η κάλυψη σφαλμάτων κατά 100% είναι αδύνατη, λόγω της ύπαρξης μη ανιχνεύσιμων σφαλμάτων

$$\text{fault detection efficiency} = \frac{\text{number of detected faults}}{\text{total number of faults} - \text{number of undetectable faults}}$$

- Ισχύει η σχέση

$$\text{Reject rate} = 1 - \text{yield}^{(1 - \text{fault coverage})}$$

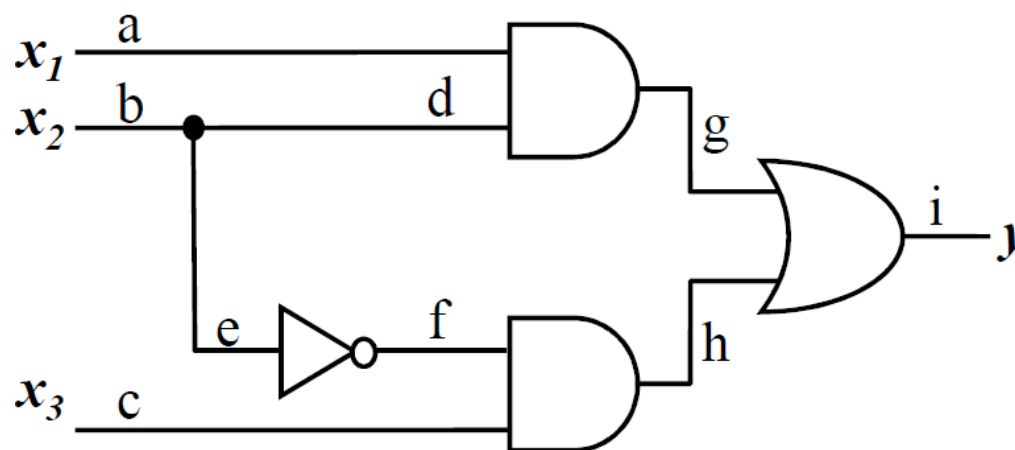
# Παραγωγή Δοκιμής (3/3)

- Στόχος της παραγωγής δοκιμής (test generation) είναι η εύρεση ενός αποδοτικού συνόλου διανυσμάτων δοκιμής τα οποία εξασφαλίζουν μέγιστη κάλυψη σφαλμάτων
- Για τον προσδιορισμό της κάλυψης σφαλμάτων χρησιμοποιείται προσομοίωση σφαλμάτων (fault simulation)
  - Απαιτούνται μοντέλα σφαλμάτων (fault models) για την προσομοίωση της συμπεριφοράς των ελαττωμάτων
- Ένα καλό μοντέλο σφαλμάτων:
  - είναι αποδοτικό για την εκτέλεση προσομοίωσης σε υπολογιστή
  - περιγράφει με ακρίβεια τη συμπεριφορά των ελαττωμάτων
- Κανένα μοντέλο σφαλμάτων δεν μπορεί να καλύψει όλα τα πιθανά ελαττώματα

# Μοντέλα Σφαλμάτων

# Μοντέλα σφαλμάτων (1/2)

- Μια υπόθεση για τα σφάλματα που μπορούν να εμφανιστούν.
- Παραδείγματα:
  - Ένα καλώδιο να είναι «κολλημένο» στο 0
  - Ένα καλώδιο να καθυστερήσει να πάει από το 0 στο 1



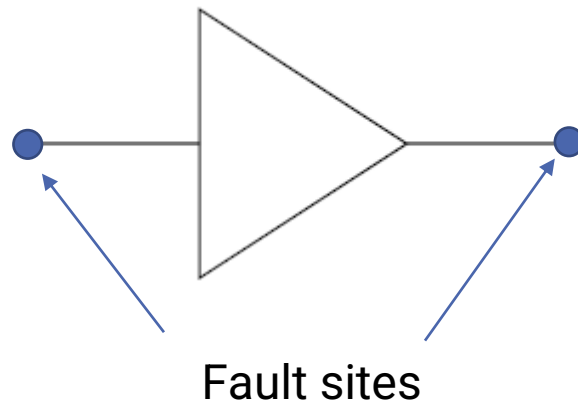
# Μοντέλα σφαλμάτων (2/2)

- Κάθε μοντέλο σφαλμάτων περιλαμβάνει  $k$  τύπους σφαλμάτων
  - $k = 2$  για τα περισσότερα μοντέλα σφαλμάτων
    - Π.χ. καλώδιο κολλημένο στο 0 ή κολλημένο στο 1
- Κάθε κύκλωμα έχει  $n$  πιθανές θέσεις σφάλματος (fault sites)
- Διακρίνουμε δύο μοντέλα σφαλμάτων
  - Μοντέλο απλού σφάλματος (Single faults)
  - Μοντέλο πολλαπλών σφαλμάτων (Multiple Faults)



# Μοντέλο απλού σφάλματος

- Ένα κύκλωμα μπορεί να έχει μόνο ένα σφάλμα
- Αριθμός απλών σφαλμάτων =  $k \times n$
- Καλή κάλυψη απλού σφάλματος συνεπάγεται γενικά και καλή κάλυψη πολλαπλών σφαλμάτων



Πόσα απλά σφάλματα μπορούν να εμφανιστούν;

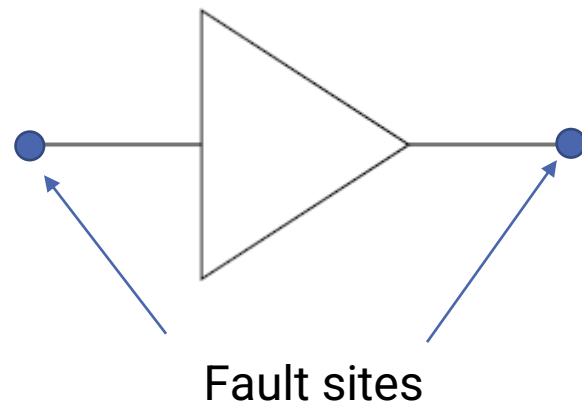
$2 \times 2 = 4$  απλά σφάλματα

Ποια είναι αυτά τα σφάλματα;

# Μοντέλο πολλαπλών σφαλμάτων (1/2)

- Ένα κύκλωμα μπορεί να έχει πολλαπλά σφάλματα συμπεριλαμβανομένων και των απλών
- Αριθμός πολλαπλών σφαλμάτων =  $(k + 1)^n - 1$ 
  - Κάθε θέση σφάλματος μπορεί να έχει έναν από τους  $k$  τύπους σφαλμάτων, ή να είναι ελεύθερο σφάλματος (fault-free)
  - Το “-1” αντιπροσωπεύει την περίπτωση απουσίας σφαλμάτων (fault-free κύκλωμα)
- Μη πρακτικό, εκτός από την περίπτωση πολύ μικρών κυκλωμάτων

# Μοντέλο πολλαπλών σφαλμάτων (2/2)



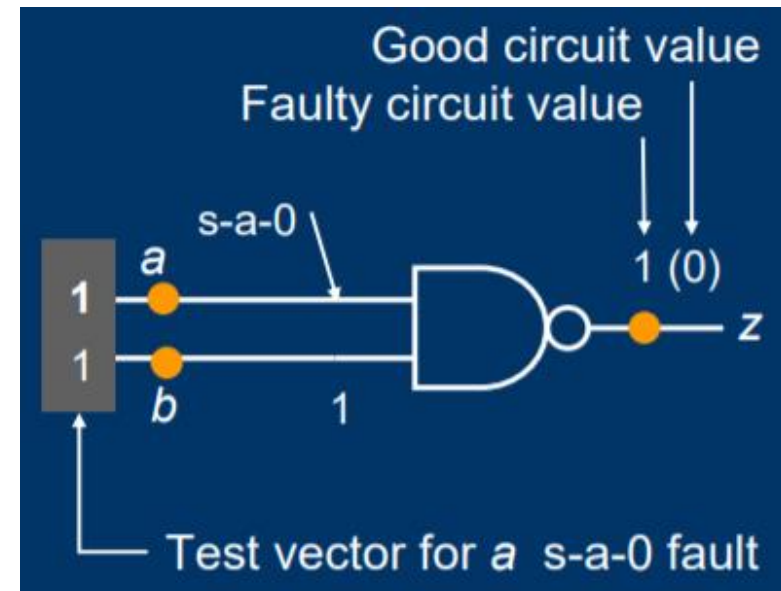
Πόσα πολλαπλά σφάλματα μπορούν να εμφανιστούν;

$3^2 - 1 = 8$  πολλαπλά σφάλματα

Ποια είναι αυτά τα σφάλματα;

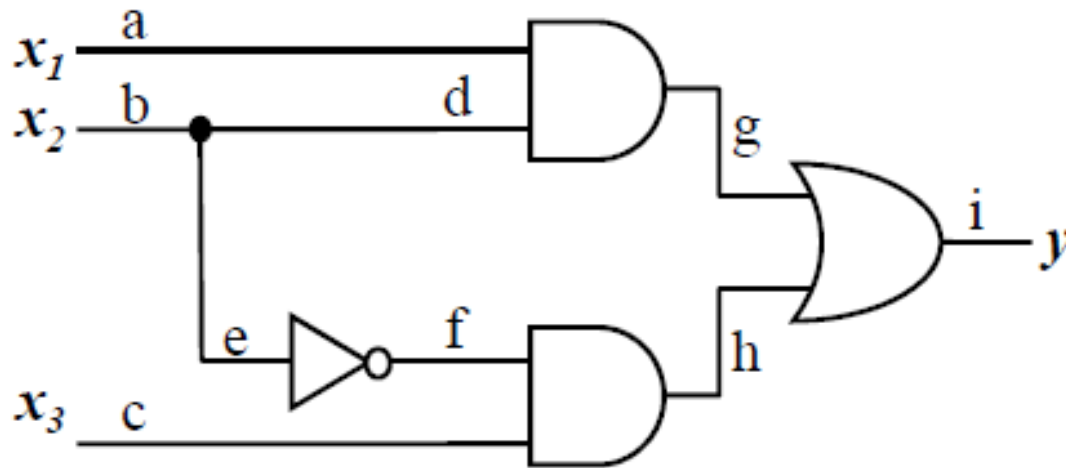
# Σφάλματα μόνιμης τιμής (stuck-at faults)

- Τρεις παράμετροι ορίζουν ένα σφάλμα μόνιμης τιμής
  - Μόνο μία γραμμή εμφανίζει σφάλμα
  - Η γραμμή κολλάει μόνιμα στο 0 ή στο 1
  - Το σφάλμα μπορεί να εμφανιστεί στην είσοδο ή έξοδο μιας πύλης
- Παράδειγμα: Πύλη NAND η οποία έχει 3 fault sites και 6 single stuck-at faults



# Σφάλματα μόνιμης τιμής (stuck-at faults)

- Πως επηρεάζει ένα σφάλμα;
  - a-SA0?
  - a-SA1?

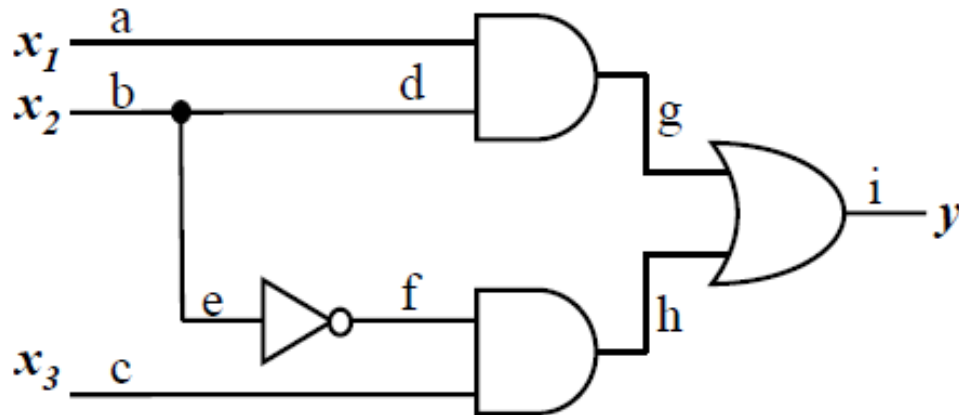


*Truth table for fault-free behavior*

$x_1x_2x_3$	000	001	010	011	100	101	110	111
$y$	0	1	0	0	0	1	1	1
a SA0	0	1	0	0	0	1	0	0
a SA1	0	1	1	1	0	1	1	1

# Σφάλματα μόνιμης τιμής (stuck-at faults)

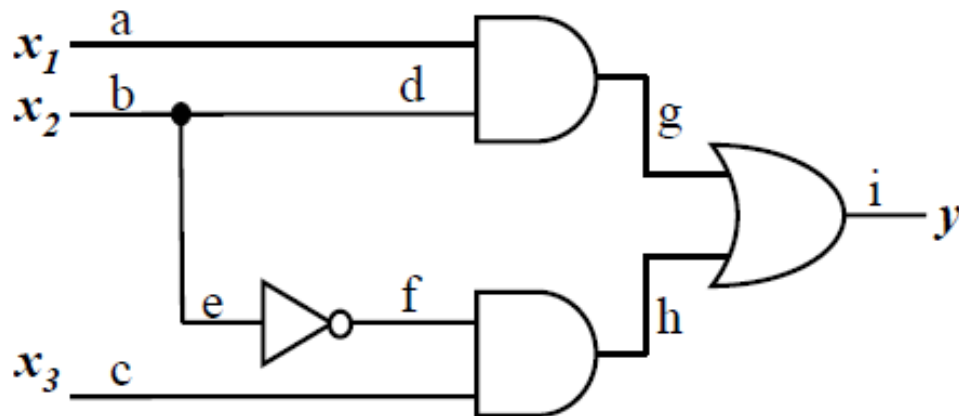
- Παράδειγμα:
  - Πλήθος θέσεων σφάλματος (fault sites)
    - $n = 9$
  - Πλήθος απλών σφαλμάτων
    - $2 \times 9 = 18$



$x_1x_2x_3$	000	001	010	011	100	101	110	111
$y$	0	1	0	0	0	1	1	1
a SA0	0	1	0	0	0	1	0	0
a SA1	0	1	1	1	0	1	1	1
b SA0	0	1	0	1	0	1	0	1
b SA1	0	0	0	0	1	1	1	1
c SA0	0	0	0	0	0	0	1	1
c SA1	1	1	0	0	1	1	1	1
d SA0	0	1	0	0	0	1	0	0
d SA1	0	1	0	0	1	1	1	1
e SA0	0	1	0	1	0	1	1	1
e SA1	0	0	0	0	0	0	1	1
f SA0	0	0	0	0	0	0	1	1
f SA1	0	1	0	1	0	1	1	1
g SA0	0	1	0	0	0	1	0	0
g SA1	1	1	1	1	1	1	1	1
h SA0	0	0	0	0	0	0	1	1
h SA1	1	1	1	1	1	1	1	1
i SA0	0	0	0	0	0	0	0	0
i SA1	1	1	1	1	1	1	1	1

# Σφάλματα μόνιμης τιμής (stuck-at faults)

- Έγκυρα διανύσματα ελέγχου
  - Ένα κύκλωμα με σφάλμα διαφέρει από ένα “καλό” κύκλωμα
- Δύσκολα και εύκολα ανιχνεύσιμα σφάλματα
  - Hard faults: f-SA1, e-SA0, d-SA1
  - Easy faults: g-SA1, h-SA1, i-SA0, i-SA1
- Απαραίτητα διανύσματα:
  - το 011 ανιχνεύει τα f-SA1 και e-SA0
  - το 100 ανιχνεύει το d-SA1
  - τα 011 και 100 ανιχνεύουν 10 σφάλματα, συνολικά
  - τα 001 και 110 ανιχνεύουν τα υπόλοιπα 8 σφάλματα



Truth table for fault-free behavior and behavior of all possible stuck-at faults

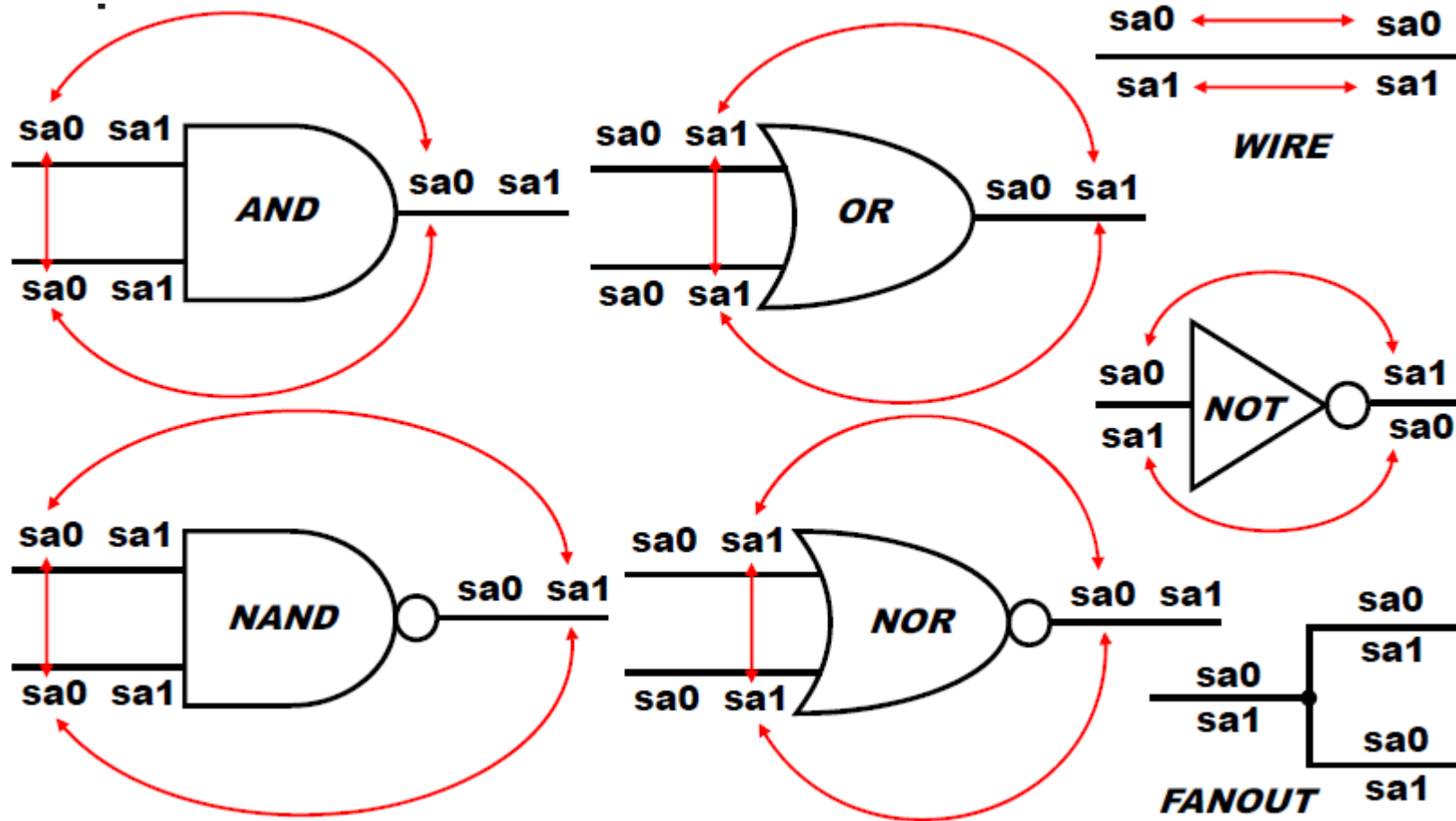
$x_1x_2x_3$	000	001	010	011	100	101	110	111
$y$	0	1	0	0	0	1	1	1
a SA0	0	1	0	0	0	1	0	0
a SA1	0	1	1	1	0	1	1	1
b SA0	0	1	0	1	0	1	0	1
b SA1	0	0	0	0	1	1	1	1
c SA0	0	0	0	0	0	0	1	1
c SA1	1	1	0	0	1	1	1	1
d SA0	0	1	0	0	0	1	0	0
d SA1	0	1	0	0	1	1	1	1
e SA0	0	1	0	1	0	1	1	1
e SA1	0	0	0	0	0	0	1	1
f SA0	0	0	0	0	0	0	1	1
f SA1	0	1	0	1	0	1	1	1
g SA0	0	1	0	0	0	1	0	0
g SA1	1	1	1	1	1	1	1	1
h SA0	0	0	0	0	0	0	1	1
h SA1	1	1	1	1	1	1	1	1
i SA0	0	0	0	0	0	0	0	0
i SA1	1	1	1	1	1	1	1	1

# Ισοδύναμα σφάλματα και περιορισμός σφαλμάτων

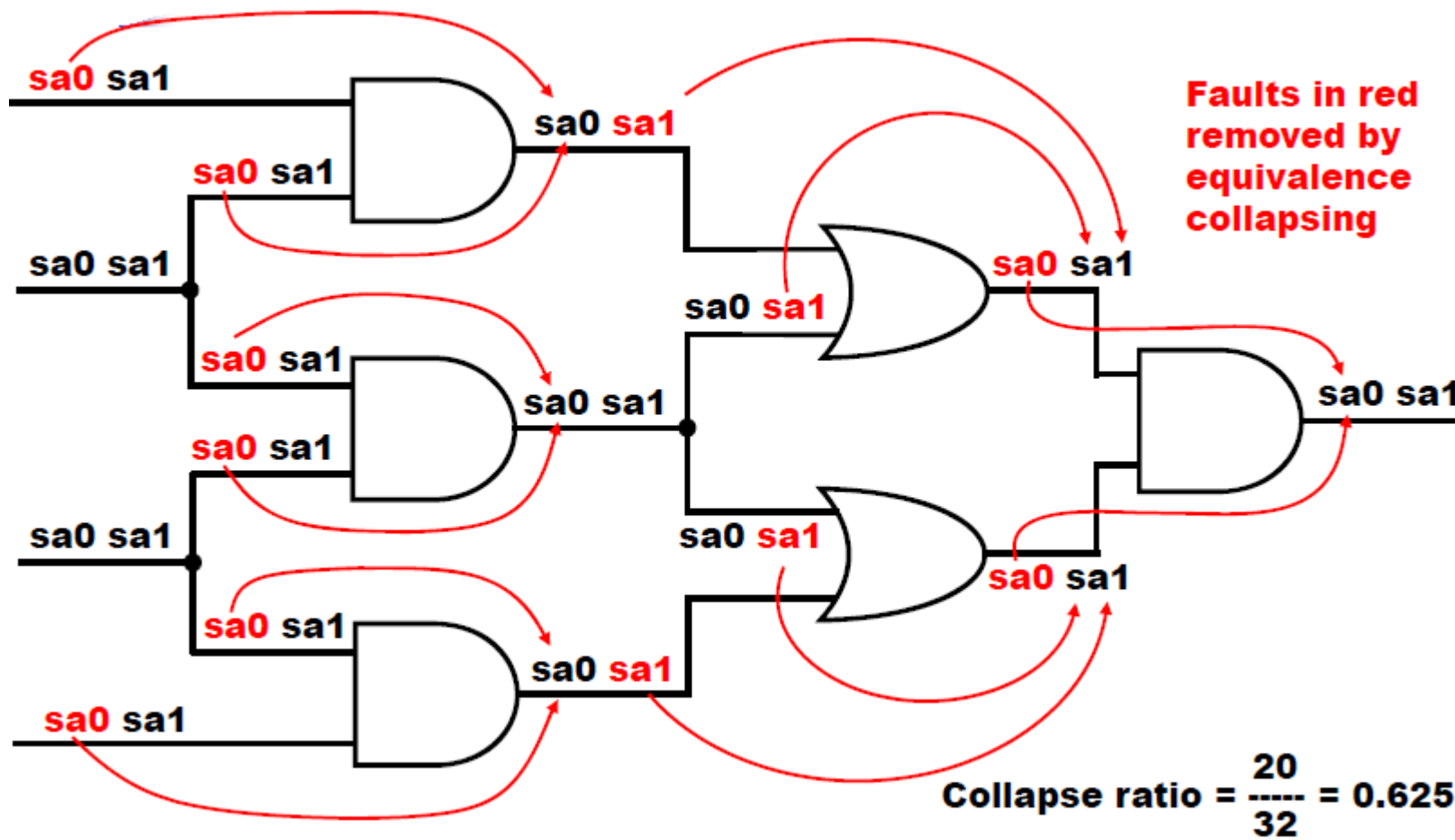
- Ισοδύναμα σφάλματα (equivalent faults)
  - Δύο σφάλματα  $f_1$  and  $f_2$  είναι ισοδύναμα αν όλες οι δοκιμές που ανιχνεύουν το  $f_1$  ανιχνεύουν και το  $f_2$  και αντίστροφα.
  - Ένα ή περισσότερα απλά σφάλματα τα οποία έχουν ίδια συμπεριφορά για όλα τα δυνατά πρότυπα εισόδου
  - Από ένα σύνολο ισοδύναμων σφαλμάτων, μόνο το ένα είναι αναγκαίο να προσομοιωθεί
- Περιορισμός σφαλμάτων (fault collapsing)
  - Αφαίρεση ισοδύναμων σφαλμάτων, εκτός ενός το οποίο θα προσομοιωθεί
  - Περιορίζει τον συνολικό αριθμό σφαλμάτων
    - Μείωση του χρόνου προσομοίωσης σφαλμάτων
    - Μείωση του χρόνου παραγωγής προτύπων δοκιμής



# Ισοδύναμα σφάλματα (1/2)

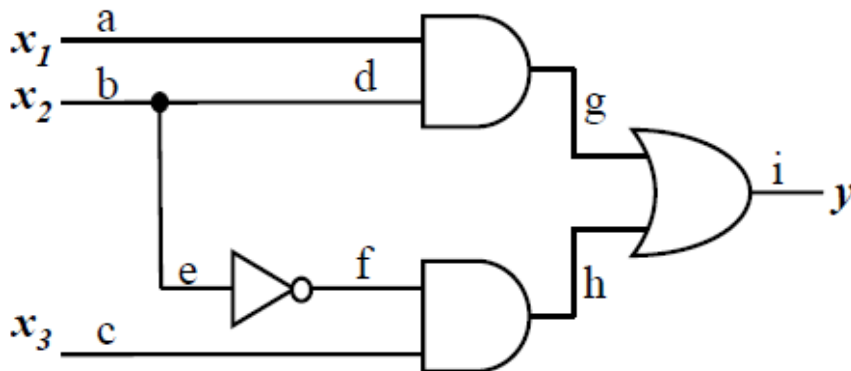


# Ισοδύναμα σφάλματα (2/2)



# Περιορισμός σφαλμάτων

- # collapsed faults =  $2 \times (P_0 + F_0) + G_i - N_i$ 
  - $P_0$ : primary outputs,  $F_0$ : fanout stems,  $G_i$ : gate inputs (για όλες τις πύλες, συμπεριλαμβανομένων των αναστροφέων),  $N_i$ : inverters
- Για το παράδειγμα του σχήματος
  - # collapsed faults = 10
  - $P_0 = 1, F_0 = 1, G_i = 7, N_i = 1$
- Ο περιορισμός των σφαλμάτων (fault collapsing) τυπικά οδηγεί σε μείωση του αριθμού των σφαλμάτων μόνιμης τιμής κατά 50% – 60%

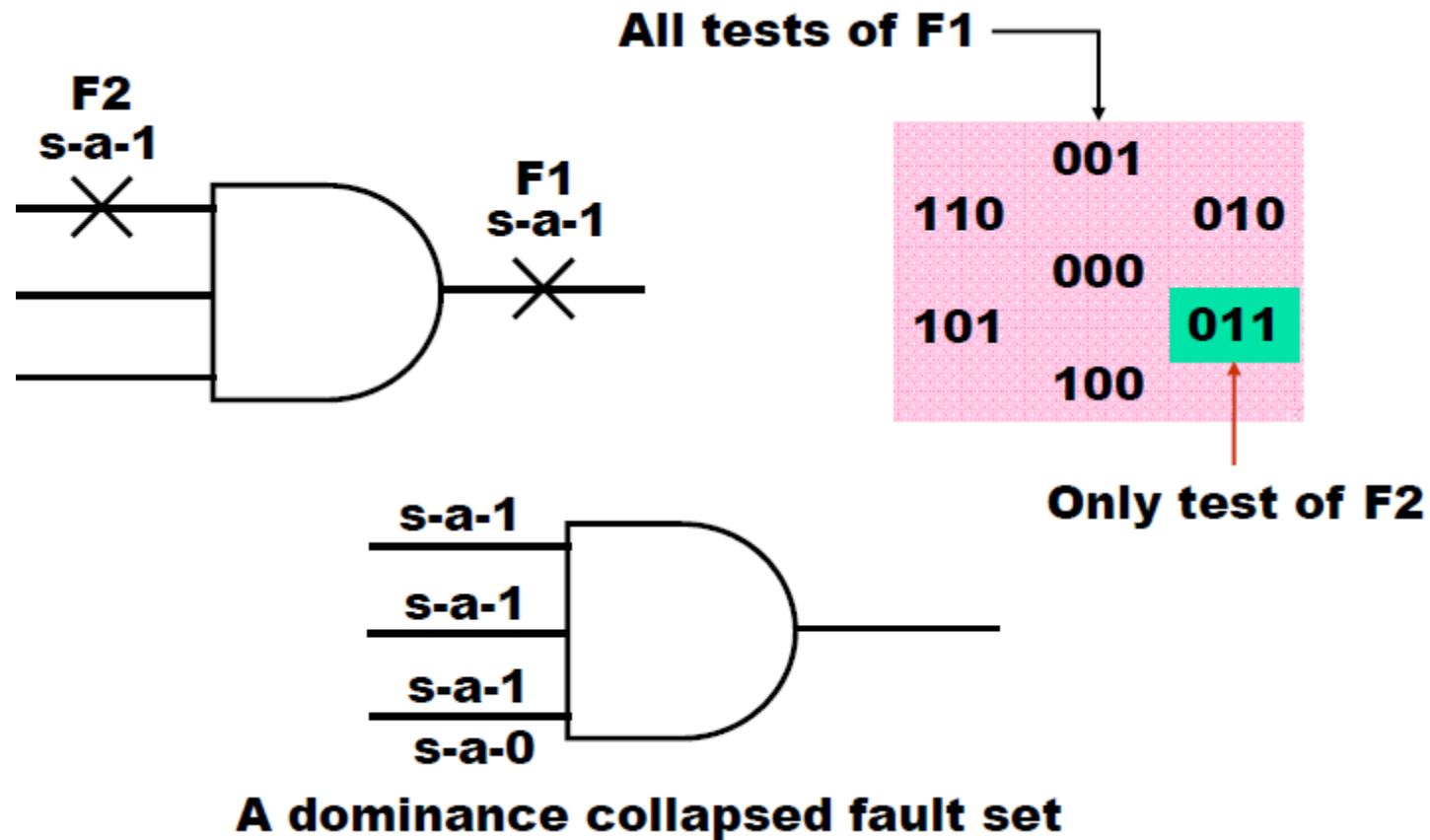


$x_1 x_2 x_3$	000	001	010	011	100	101	110	111
$y$	0	1	0	0	0	1	1	1
a SA0	0	1	0	0	0	1	0	0
a SA1	0	1	1	1	0	1	1	1
b SA0	0	1	0	1	0	1	0	1
b SA1	0	0	0	0	1	1	1	1
c SA0	0	0	0	0	0	0	1	1
c SA1	1	1	0	0	1	1	1	1
d SA0	0	1	0	0	0	1	0	0
d SA1	0	1	0	0	1	1	1	1
e SA0	0	1	0	1	0	1	1	1
e SA1	0	0	0	0	0	0	1	1
f SA0	0	0	0	0	0	0	1	1
f SA1	0	1	0	1	0	1	1	1
g SA0	0	1	0	0	0	1	0	0
g SA1	1	1	1	1	1	1	1	1
h SA0	0	0	0	0	0	0	1	1
h SA1	1	1	1	1	1	1	1	1
i SA0	0	0	0	0	0	0	0	0
i SA1	1	1	1	1	1	1	1	1

# Κυριαρχία σφαλμάτων

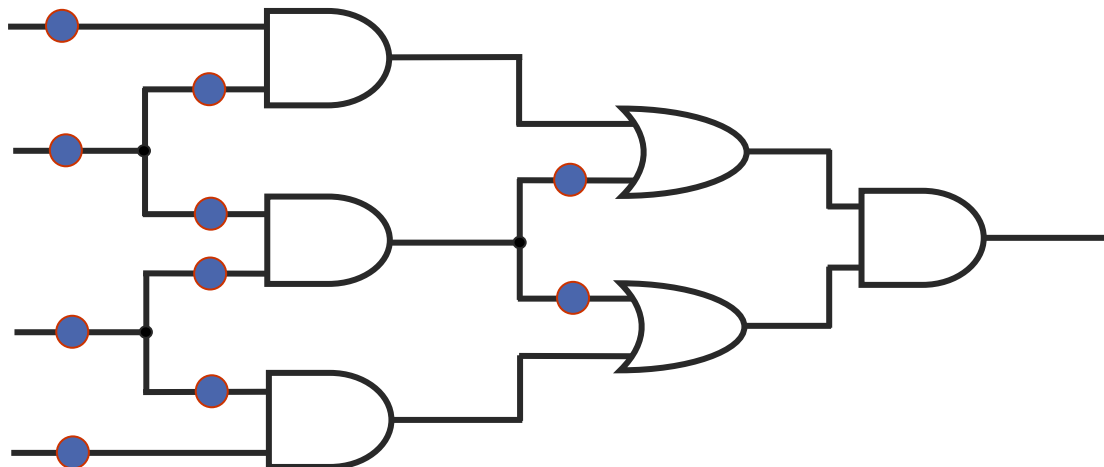
- Έστα τα σφάλματα  $f1$  και  $f2$  με σύνολα δοκιμής  $T1$  και  $T2$  αντίστοιχα
- Το σφάλμα  $f1$  κυριαρχεί το σφάλμα  $f2$ , εάν το σύνολο δοκιμής  $T2$  είναι υποσύνολο του συνόλου δοκιμής  $T1$
- Σ' αυτή την περίπτωση ένα διάνυσμα δοκιμής το οποίο ανιχνεύει το σφάλμα  $f2$ , θα ανιχνεύει επίσης το σφάλμα  $f1$
- Η σχέση δεν είναι συμμετρική

# Κυριαρχία σφαλμάτων



# Checkpoints

- Οι κύριοι είσοδοι και τα fanout branches ενός συνδυαστικού κυκλώματος ονομάζονται checkpoints.
- Ένα διάνυσμα δοκιμής το οποίο ανιχνεύει όλα τα single (multiple) stuck-at faults σε όλα τα checkpoints ενός συνδυαστικού κυκλώματος, ανιχνεύει επίσης και όλα τα single (multiple) stuck-at faults στο κύκλωμα.

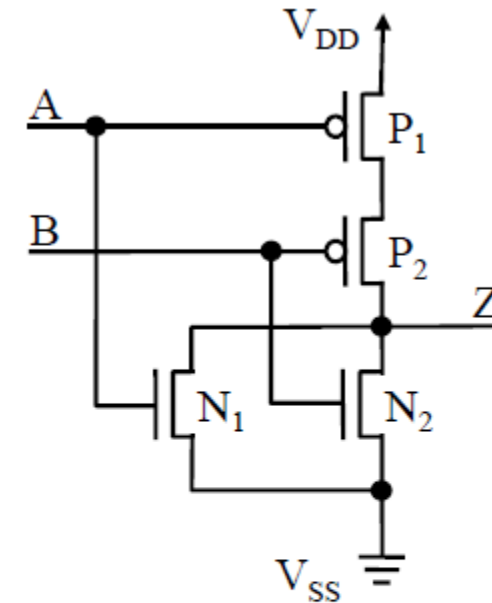


Total fault sites = 16

Checkpoints (●) = 10

# Σφάλματα σε τρανζίστορ (1/2)

- Ένα transistor μπορεί να έχει τα εξής σφάλματα:
  - Stuck-short ή stuck-on: άγει συνεχώς
  - Stuck-open ή stuck-off: δεν άγει ποτέ
- pMOS τρανζίστορ
  - 0 → άγει
  - 1 → δεν άγει
- nMOS τρανζίστορ
  - 0 → δεν άγει
  - 1 → άγει



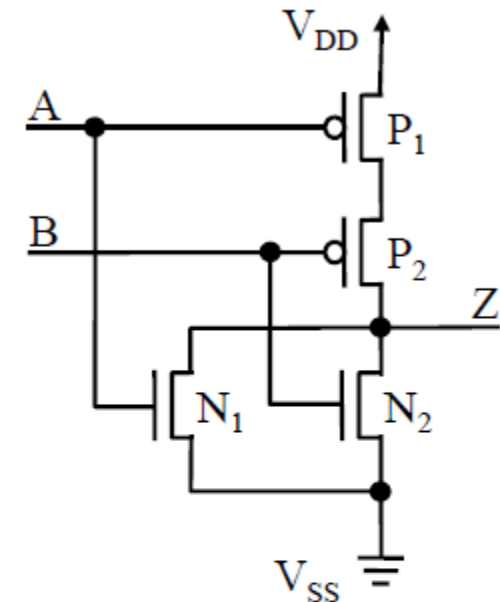
*Truth table for fault-free circuit*

<i>AB</i>	00	01	10	11
<i>Z</i>	1	0	0	0

Πύλη NOR 2 εισόδων (A και B)  
 $V_{DD}$ : Θετική τάση τροφοδοσίας  
 $V_{SS}$ : γείωση

# Σφάλματα σε τρανζίστορ (2/2)

- Stuck-short σφάλματα
  - Δημιουργούν αγώγιμο μονοπάτι μεταξύ του  $V_{DD}$  και του  $V_{SS}$ .
  - Μπορεί να ανιχνευθεί με την παρατήρηση του ρεύματος τροφοδοσίας σταθερής κατάστασης  $I_{DDQ}$ .
- Stuck-open σφάλματα
  - Αναγκάζουν έναν κόμβο εξόδου να αποθηκεύσει την τελευταία τιμή της τάσης



*Truth table for fault-free circuit*

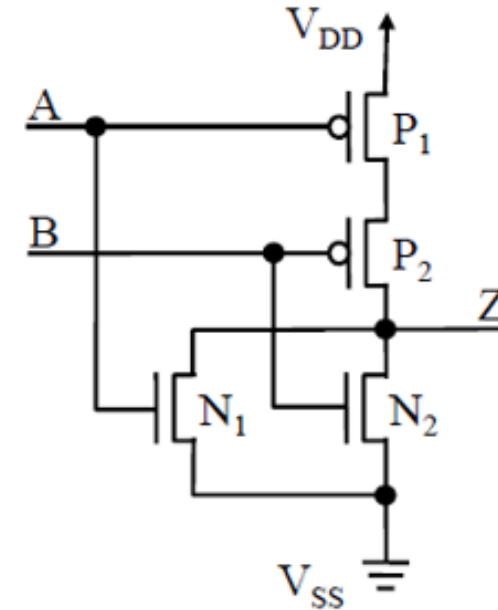
$AB$	00	01	10	11
$Z$	1	0	0	0



# Σφάλματα σε τρανζίστορ

## Stuck-short σφάλμα

- $N_1$  stuck-short σφάλμα με εισόδους 00
  - Το τρανζίστορ  $P_1$  άγει
  - Το τρανζίστορ  $P_2$  άγει
  - Το τρανζίστορ  $N_1$  άγει
  - Δημιουργείται αγώγιμο μονοπάτι μεταξύ του  $V_{DD}$  και του  $V_{SS}$ .
  - Ανιχνεύεται μέσω της τιμής του  $I_{DDQ}$ .



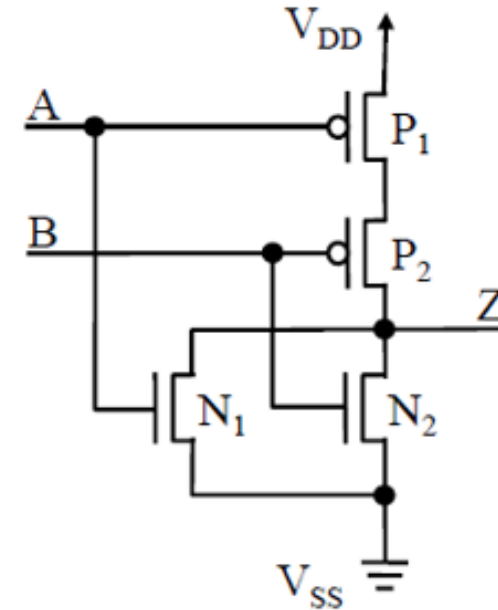
*Truth table for fault-free circuit and  $N_1$  stuck-short transistor fault*

$AB$	00	01	10	11
$Z$	1	0	0	0
$N_1$ stuck-short	$I_{DDQ}$	0	0	0

# Σφάλματα σε τρανζίστορ

## Stuck-open σφάλμα

- $N_1$  stuck-open σφάλμα με εισόδους 10
  - Το τρανζίστορ  $P_1$  δεν άγει
  - Το τρανζίστορ  $P_2$  άγει
  - Το τρανζίστορ  $N_1$  δεν άγει
  - Το τρανζίστορ  $N_2$  δεν άγει
  - Το  $Z$  απομονώνεται από την γείωση  $V_{SS}$ .
  - Το  $Z$  διατηρεί την προηγούμενη κατάσταση του (0 ή 1).



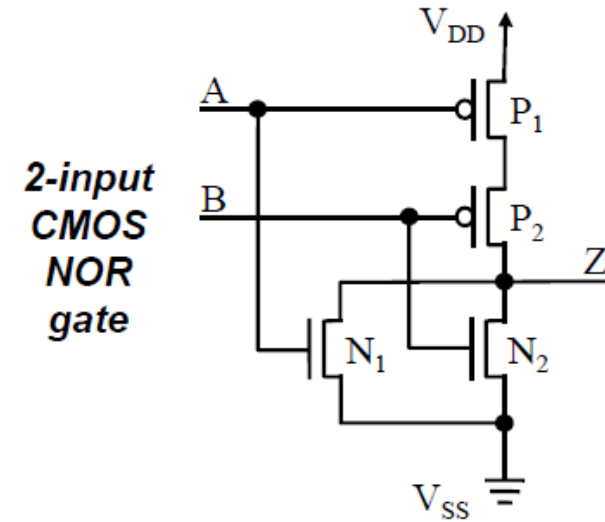
*Truth table for fault-free circuit and  $N_1$  stuck-open transistor fault*

$AB$	00	01	10	11
$Z$	1	0	0	0
$N_1$ stuck-open	1	0	last Z	0

# Σφάλματα σε τρανζίστορ

## Παράδειγμα

- Πόσα απλά σφάλματα μπορούμε να έχουμε;
  - 4 fault sites
  - $2 \times 4 = 8$  απλά σφάλματα
- Τα stuck open απλά σφάλματα για την ανίχνευσή τους απαιτούν την εφαρμογή μιας ακολουθίας από 2 διανύσματα ελέγχου
  - Π.χ. η διαδοχική εφαρμογή των διανυσμάτων 00 και 01 επιτρέπει την ανίχνευση του  $N_2$  stuck-open



*Truth table for fault-free circuit and all possible transistor faults*

$AB$	00	01	10	11
$Z$	1	0	0	0
$N_1$ stuck-open	1	0	last Z	0
$N_1$ stuck-short	$I_{DDQ}$	0	0	0
$N_2$ stuck-open	1	last Z	0	0
$N_2$ stuck-short	$I_{DDQ}$	0	0	0
$P_1$ stuck-open	last Z	0	0	0
$P_1$ stuck-short	1	0	$I_{DDQ}$	0
$P_2$ stuck-open	last Z	0	0	0
$P_2$ stuck-short	1	$I_{DDQ}$	0	0

# Σφάλματα σε τρανζίστορ

## Περιορισμένα σφάλματα

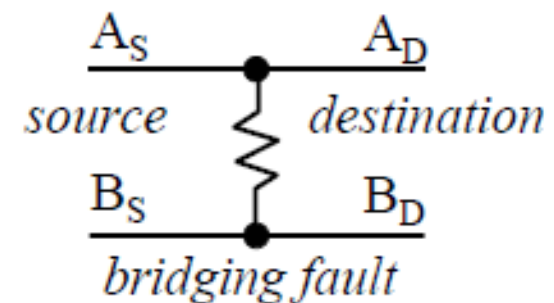
- Πλήθος περιορισμένων σφαλμάτων =  $2 \times T - T_S + G_S - T_P + G_P$ 
  - $T$ : αριθμός τρανζίστορ
  - $T_S$ : αριθμός τρανζίστορ σε σειρά
  - $G_S$ : αριθμός ομάδων με τρανζίστορ σε σειρά
  - $T_P$ : αριθμός παράλληλων τρανζίστορ
  - $G_P$ : αριθμός ομάδων με παράλληλα τρανζίστορ
- Για το παράδειγμα του σχήματος, έχουμε # collapsed faults = 6
  - $T = 4, T_S = 2, G_S = 1, T_P = 2, G_P = 1$
- Ο περιορισμός των σφαλμάτων (fault collapsing) τυπικά οδηγεί σε μείωση του αριθμού των σφαλμάτων τρανζίστορ κατά 25 – 35%

# Σφάλματα σε γραμμές

- Οι γραμμές (wires) μπορεί να είναι:
  - Ανοικτές
    - ασυνέχειες (opens) σε γραμμές που διασυνδέουν τρανζίστορ για το σχηματισμό λογικών πυλών
      - συμπεριφέρονται σαν stuck-open faults σε τρανζίστορ
    - ασυνέχειες (opens) σε γραμμές που διασυνδέουν πύλες για το σχηματισμό κυκλωμάτων
      - συμπεριφέρονται σαν σφάλματα μόνιμης τιμής (stuck-at faults).
    - οι ασυνέχειες (opens) ανιχνεύονται με διανύσματα ελέγχου τα οποία είναι κατάλληλα για την ανίχνευση σφαλμάτων τρανζίστορ και μόνιμης τιμής
  - Βραχυκυκλωμένες με κάποια γειτονική γραμμή (σφάλμα γεφύρωσης – bridging fault)

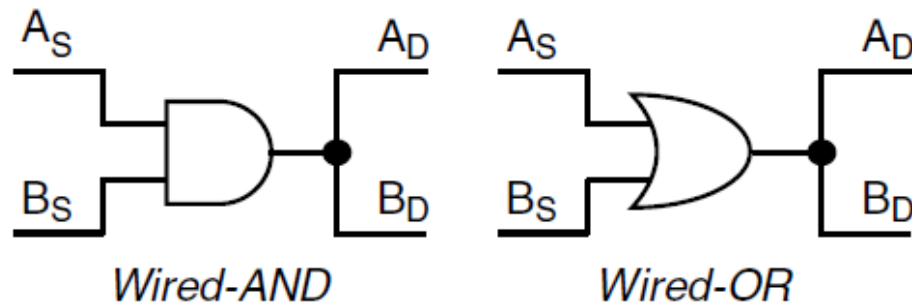
# Σφάλματα γεφύρωσης

- Η περίπτωση που ένα στοιχείο είναι βραχυκυκλωμένο στην τάση τροφοδοσίας ή στην γείωση ισοδυναμεί με το μοντέλο stuck-at fault.
- Η περίπτωση που βραχυκυκλώνονται δύο καλώδια είναι διαφορετική και απαιτούνται μοντέλα σφαλμάτων γεφύρωσης.
- Μοντέλα σφαλμάτων γεφύρωσης:
  - καλωδιωμένη AND/OR (wired AND/OR)
  - υπερισχύον (dominant)
  - υπερισχύον-AND/OR (dominant -AND/OR)
- Ανιχνεύονται με έλεγχο IDDQ



# Σφάλματα γεφύρωσης wired AND/OR

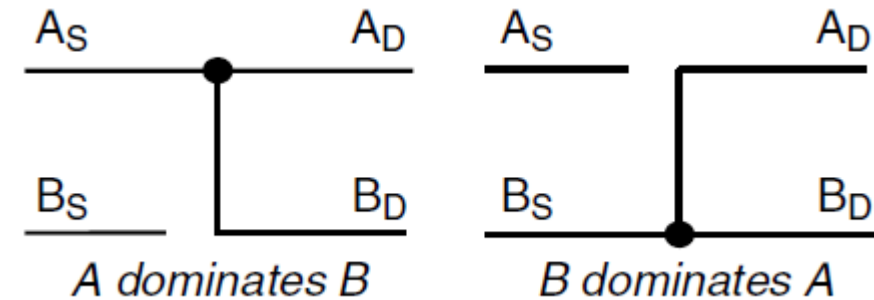
- wired-AND: Το σήμα που προκύπτει από τις βραχυκυκλωμένες γραμμές παίρνει το λογικό 0 αν οποιαδήποτε από τις βραχυκυκλωμένες γραμμές παίρνει στην είσοδό της το λογικό 0
- wired-OR: Το σήμα που προκύπτει από τις βραχυκυκλωμένες γραμμές παίρνει το λογικό 1 αν οποιαδήποτε από τις βραχυκυκλωμένες γραμμές παίρνει στην είσοδό της το λογικό 1



$A_S B_S$	0	0	0	1	1	0	1	1
$A_D B_D$	0	0	0	1	1	0	1	1
Wired-AND	0	0	0	0	0	0	1	1
Wired-OR	0	0	1	1	1	1	1	1

# Σφάλματα γεφύρωσης dominant

- Μία είσοδος καθορίζει την τιμή και στα 2 βραχυκυκλωμένα κυκλώματα
- Ένα σύνολο από διανύσματα δοκιμής το οποίο ανιχνεύει όλα τα dominant σφάλματα γεφύρωσης μπορεί να ανιχνεύσει και όλα τα wired-AND και wired-OR σφάλματα γεφύρωσης.

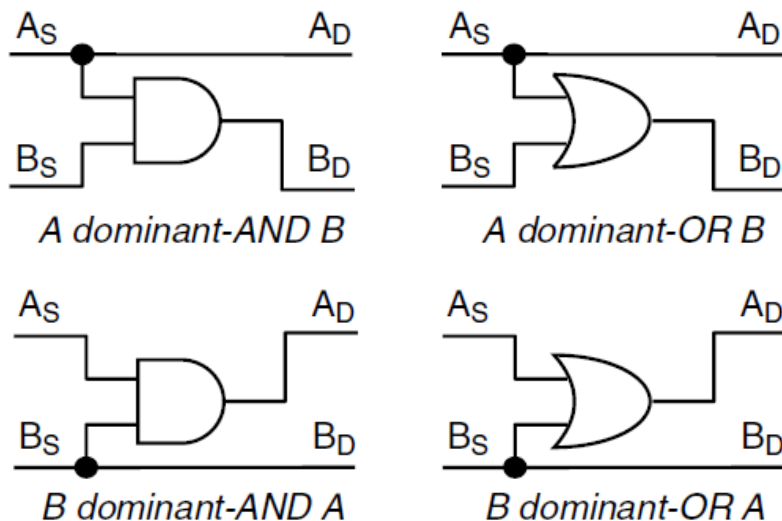


$A_S B_S$	0	0	0	1	1	0	1	1
$A_D B_D$	0	0	0	1	1	0	1	1
Wired-AND	0	0	0	0	0	0	1	1
Wired-OR	0	0	1	1	1	1	1	1
A dominates B	0	0	0	0	1	1	1	1
B dominates A	0	0	1	1	0	0	1	1



# Σφάλματα γεφύρωσης dominant-AND/OR

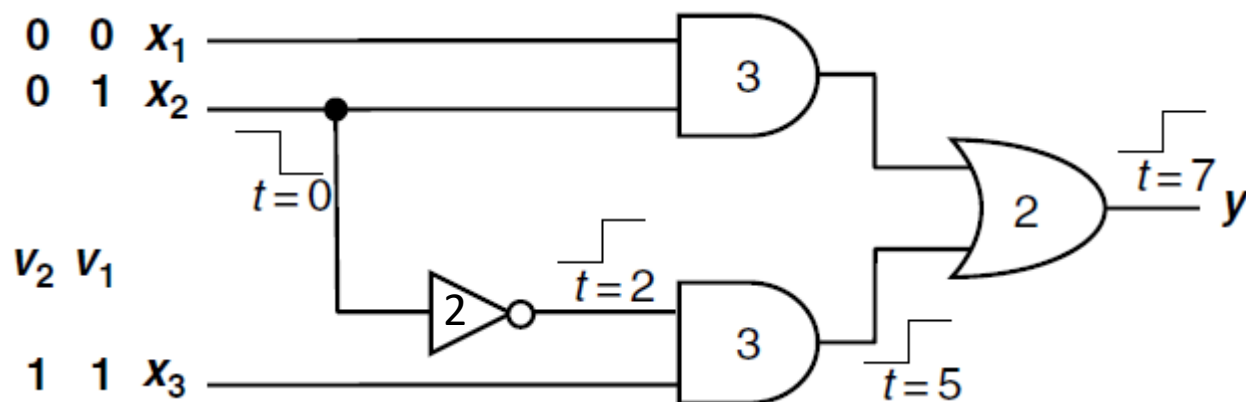
- Μία είσοδος καθορίζει την τιμή και στα 2 βραχυκυκλωμένα κυκλώματα αλλά μόνο για μία δεδομένη λογική τιμή



$A_S B_S$	0	0	0	1	1	0	1	1
$A_D B_D$	0	0	0	1	1	0	1	1
Wired-AND	0	0	0	0	0	0	1	1
Wired-OR	0	0	1	1	1	1	1	1
A dominates B	0	0	0	0	1	1	1	1
B dominates A	0	0	1	1	0	0	1	1
A dominant-AND B	0	0	0	0	1	0	1	1
B dominant-AND A	0	0	0	1	0	0	1	1
A dominant-OR B	0	0	0	1	1	1	1	1
<b>B dominant-OR A</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>

# Σφάλματα καθυστέρησης

- Τα μοντέλα σφάλματος καθυστέρησης διαδρομής (path-delay fault models) λαμβάνουν υπόψη την αθροιστική καθυστέρηση διάδοσης (propagation delay) δια μέσου του CUT
  - 2 διανύσματα ελέγχου δημιουργούν μια μετάβαση κατά μήκος μιας διαδρομής
  - τα ελαττωματικά κυκλώματα παρουσιάζουν υπερβολική καθυστέρηση
- Καθυστερήσεις και στιγμιαία σφάλματα (glitches) μπορεί να οφείλονται στη διαφωνία (crosstalk)



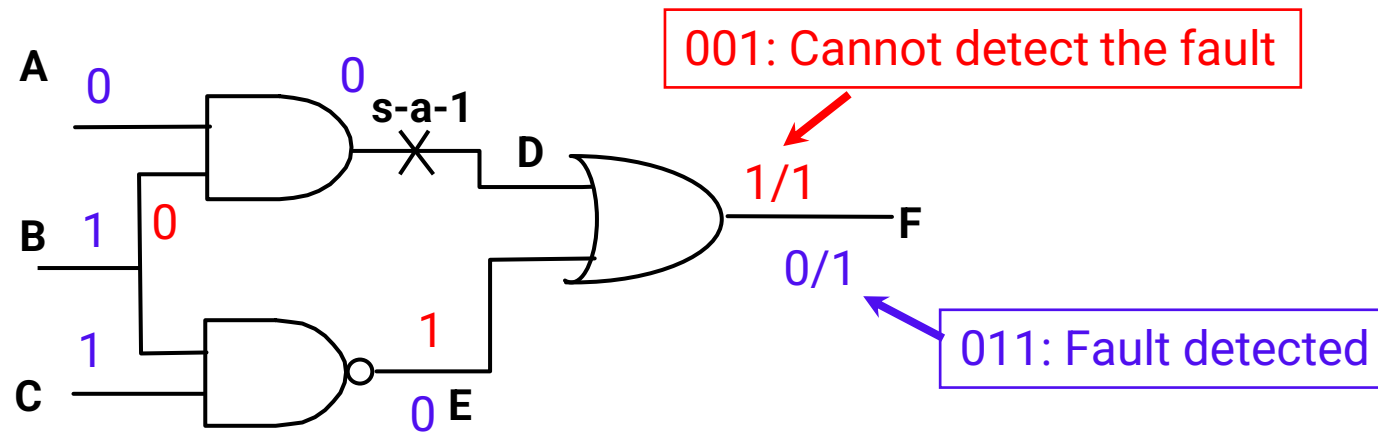
$v_1$  και  $v_2$  διανύσματα ελέγχου για την δοκιμή της διαδρομής από το  $x_2$  στο  $y$ , μέσω του inverter

# Σφάλματα σε μνήμη RAM

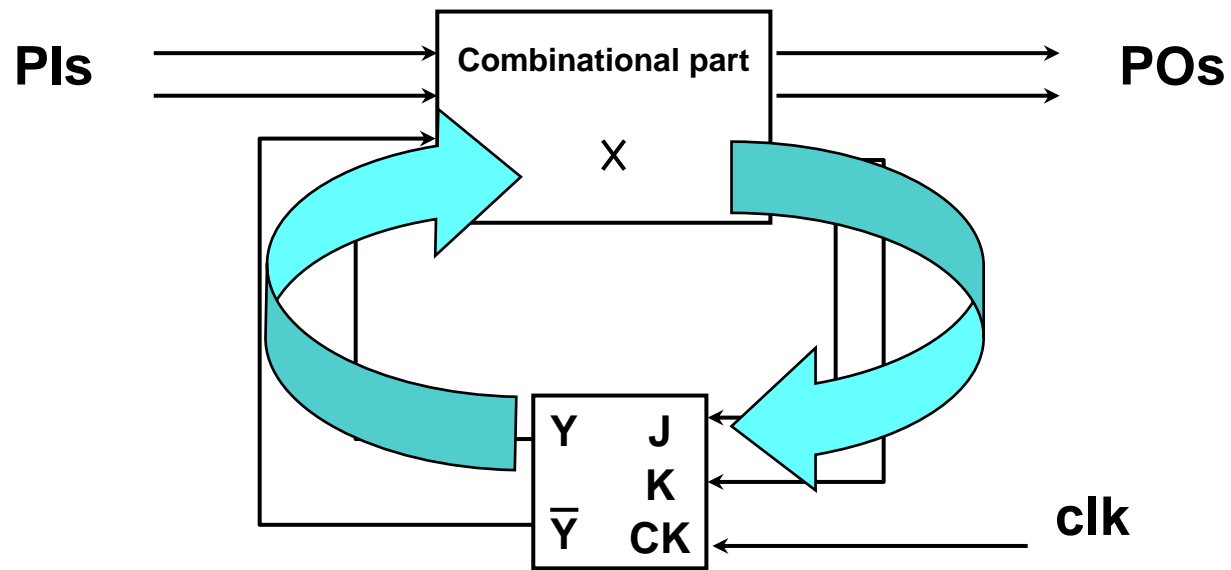
- Pattern sensitivity faults
  - Τα περιεχόμενα ενός κελιού μνήμης (memory cell) επηρεάζονται από τα περιεχόμενα γειτονικών κελιών
- Coupling faults
  - Η αλλαγή του περιεχομένου ενός κελιού μνήμης προκαλεί αλλαγή περιεχομένου ενός άλλου κελιού

# Επισκόπηση τεχνολογίας ελέγχου VLSI κυκλωμάτων

# Δυσκολίες στην δημιουργία δοκιμής (test generation)



# Δυσκολίες στην δημιουργία δοκιμής (test generation)



# Automatic Test Equipment (ATE)

- Αποτελείται από:
  - Υπολογιστή για την κεντρική διαχείριση και τον ευέλικτο έλεγχο και μέτρηση διαφορετικών προϊόντων
  - Κυκλώματα ηλεκτρονικών διεπαφών για την εφαρμογή των προτύπων δοκιμής και τη λήψη των αποκρίσεων
  - Πρόγραμμα ελέγχου για τον έλεγχο του συγχρονισμού των προτύπων δοκιμής και τη σύγκριση των αποκρίσεων με τις αναμενόμενες

# Automatic Test Pattern Generation και Fault Simulation

- Automatic Test Pattern Generation (ATPG)
  - Αλγόριθμοι οι οποίοι, για δεδομένο κύκλωμα παράγουν μια ακολουθία διανυσμάτων δοκιμής με βάση συγκεκριμένα μοντέλα σφαλμάτων
- Fault Simulation
  - Προσομοιώνει μοντέλα σφαλμάτων στο υπό εξέταση κύκλωμα και εφαρμόζει διανύσματα δοκιμής με σκοπό τον προσδιορισμό της κάλυψης σφαλμάτων
  - Ο μεγάλος χρόνος προσομοίωσης, εξαιτίας του μεγάλου αριθμού σφαλμάτων που πρέπει να προσομοιωθούν, μπορεί να μειωθεί με τη χρήση παράλληλης, επαγωγικής (deductive) και ταυτόχρονης (concurrent) προσομοίωσης σφαλμάτων



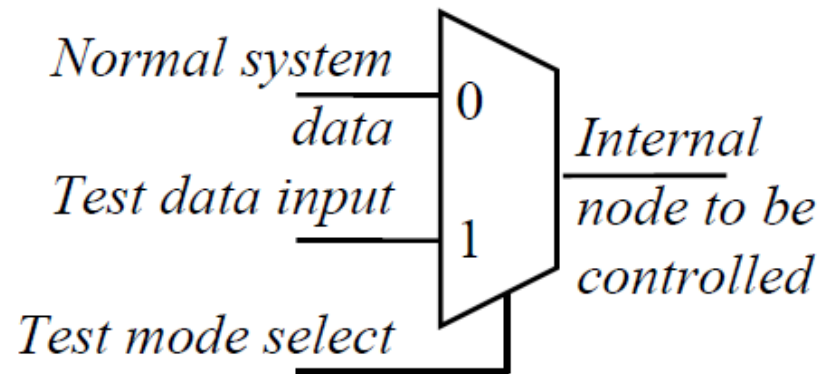
# Σχεδίαση για ελεγχιμότητα

- Για την δοκιμή κυκλωμάτων απαιτείται ο έλεγχος και η παρατήρηση των λογικών τιμών των εσωτερικών γραμμών του κυκλώματος.
  - Ορισμένα στοιχεία σε ένα ακολουθιακό κύκλωμα είναι πολύ δύσκολο να ελεγχθούν και παρατηρηθούν
    - Π.χ. η τιμή του MSB σε ένα n-bit counter μπορεί να παρατηρηθεί μετά από  $2^{n-1}$  κύκλους ρολογιού
- Σχεδίαση για ελεγχιμότητα – Design For Testability (DFT)
  - Ενσωματώνεται στην σχεδίαση των κυκλωμάτων
  - Σκοπός της η βελτίωση της ελεγχιμότητας (controllability) ή/και της παρατηρησιμότητας (observability) των εσωτερικών κόμβων ενός κυκλώματος
- Υπάρχουν τρεις βασικές προσεγγίσεις:
  - Ad-hoc τεχνικές – εξειδικευμένη σχεδίαση κατά περίπτωση
  - Τεχνικές σάρωσης (scan design)
  - Ενσωματωμένος αυτοέλεγχος – (built-in self test, BIST)

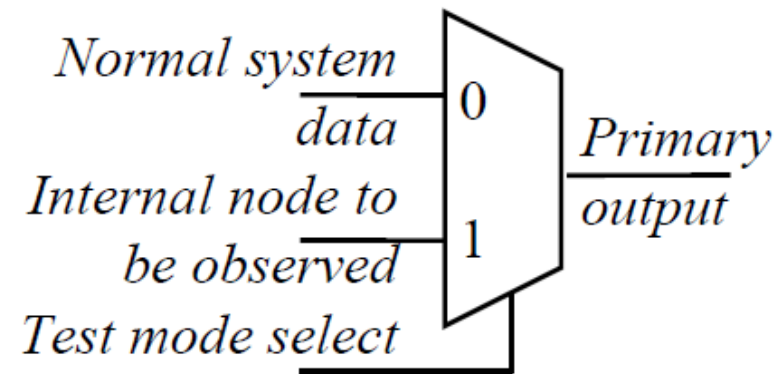
# Σχεδίαση για ελεγχσιμότητα

## Ad hoc τεχνικές

- Προσθήκη εσωτερικών σημείων ελέγχου (test points), συνήθως πολυπλεκτών, για ελεγχσιμότητα και παρατηρησιμότητα
- Χρήση “κατά περίπτωση”, ιδιαίτερα για τα μέρη ενός κυκλώματος τα οποία είναι δύσκολο να ελεγχθούν



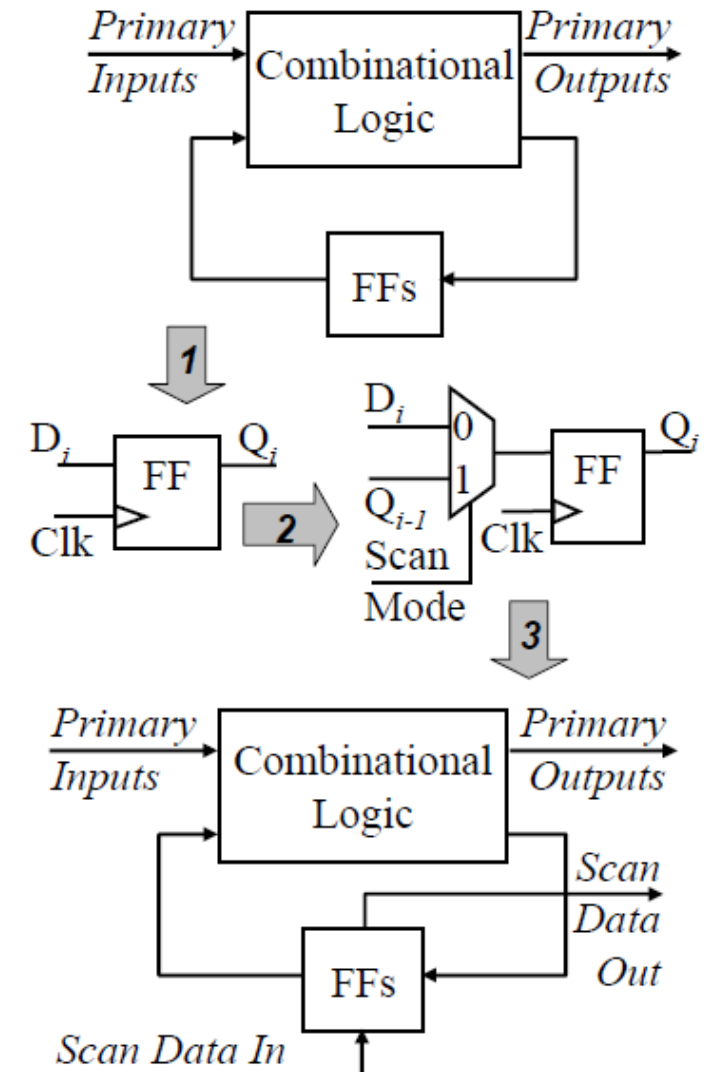
controllability test point



observability test point

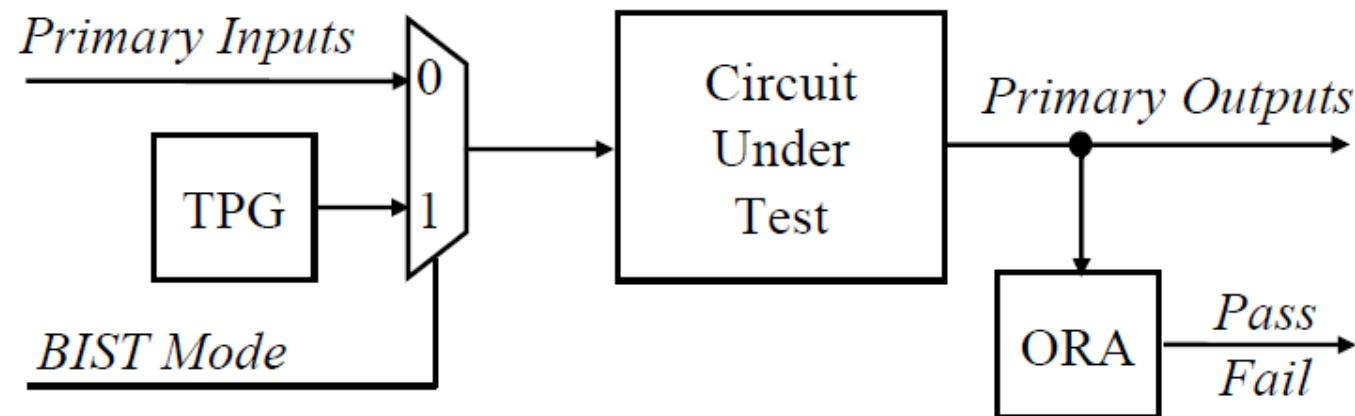
# Σχεδίαση για ελεγχιμότητα Τεχνικές Σάρωσης

- Σχεδίαση σάρωσης (scan design)
  - Μετατρέπει τα flip-flops ενός chip σε έναν καταχωρητή ολίσθησης
  - Σε λειτουργία σάρωσης (scan mode) επιτρέπεται:
    - Η σειριακή εισαγωγή των διανυσμάτων ελέγχου
    - Η σειριακή εξαγωγή των αποκρίσεων



# Σχεδίαση για ελεγχιμότητα Built-In Self-Test

- Ενσωματώνει γεννήτρια προτύπων δοκιμής (test pattern generator, TPG) και αναλυτή απόκρισης εξόδου (Output Response Analyzer)
  - Το κύκλωμα μπορεί να ελέγξει τον εαυτό του
- Αξιοποιείται σε όλες τις φάσεις ελέγχου
  - Διάταξη → PCB → σύστημα → λειτουργία πεδίου



**Δείτε επίσης**

# Ασκήσεις

# Άσκηση 1

- Έστω ένα PCB το οποίο περιλαμβάνει 40 ολοκληρωμένα κυκλώματα, το καθένα από τα οποία έχει κάλυψη σφαλμάτων 90% και κατασκευαστική απόδοση 90%. Να βρεθεί το reject rate και ο αριθμός των ελαττωματικών στοιχείων του PCB ανά εκατομμύριο.

# Άσκηση 1 - Λύση

- Το reject rate για ένα ολοκληρωμένο κύκλωμα δίνεται από την σχέση:

$$Reject\ rate = 1 - yield^{(1 - fault\ coverage)} = 1 - 0,9^{(1 - 0,9)} = 0,01048$$

Το reject rate για το PCB είναι  $40 * 0,01048 = 41,9\%$

Οπότε ο αριθμός των ελαττωματικών στοιχείων του PCB ανά εκατομμύριο είναι 419.000.

## Άσκηση 2

- Κατά τη διαδικασία παραγωγής ολοκληρωμένων κυκλωμάτων, παράγεται μια παρτίδα από 2000 κυκλώματα τα οποία είναι έτοιμα για διάθεση στην αγορά. Πόσα από αυτά τα κυκλώματα θα είναι ελαττωματικά, αν η κατασκευαστική απόδοση είναι 50% και η κάλυψη σφαλμάτων είναι 90% για το σύνολο των διανυσμάτων ελέγχου που χρησιμοποιούνται για τον έλεγχο ορθής λειτουργίας των κυκλωμάτων;



## Άσκηση 2 - Λύση

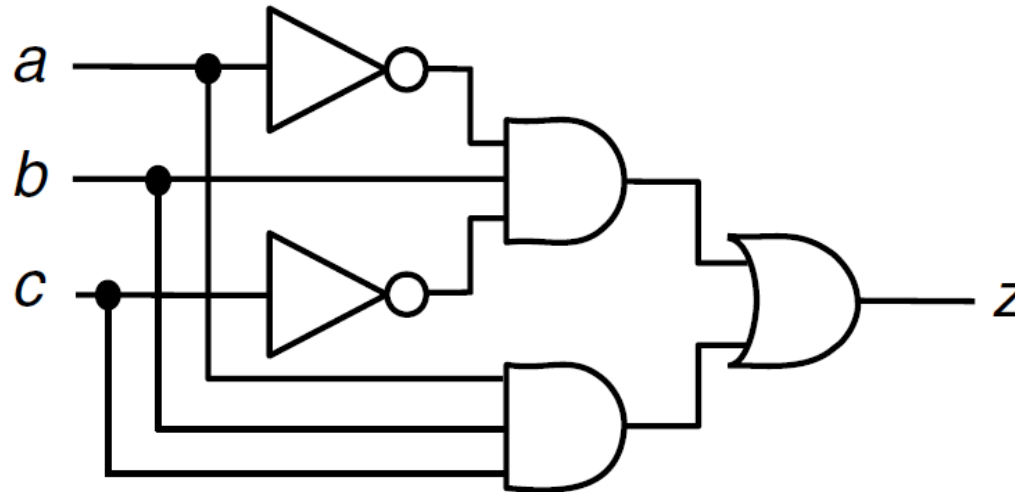
- Το reject rate δίνεται από την παρακάτω σχέση:

$$\text{Reject rate} = 1 - \text{yield}^{(1 - \text{fault coverage})} = 1 - 0,5^{(1 - 0,9)} = 6.7\%$$

Άρα θα είναι ελαττωματικά 134 κυκλώματα

# Άσκηση 3

- Δίνεται το συνδυαστικό κύκλωμα του σχήματος.
  - Ποιος ο αριθμός των πιθανών απλών σφαλμάτων μόνιμης τιμής;
  - Ποιος ο αριθμός των πιθανών πολλαπλών σφαλμάτων μόνιμης τιμής;
  - Ποιος ο αριθμός των περιορισμένων (collapsed) σφαλμάτων μόνιμης τιμής;

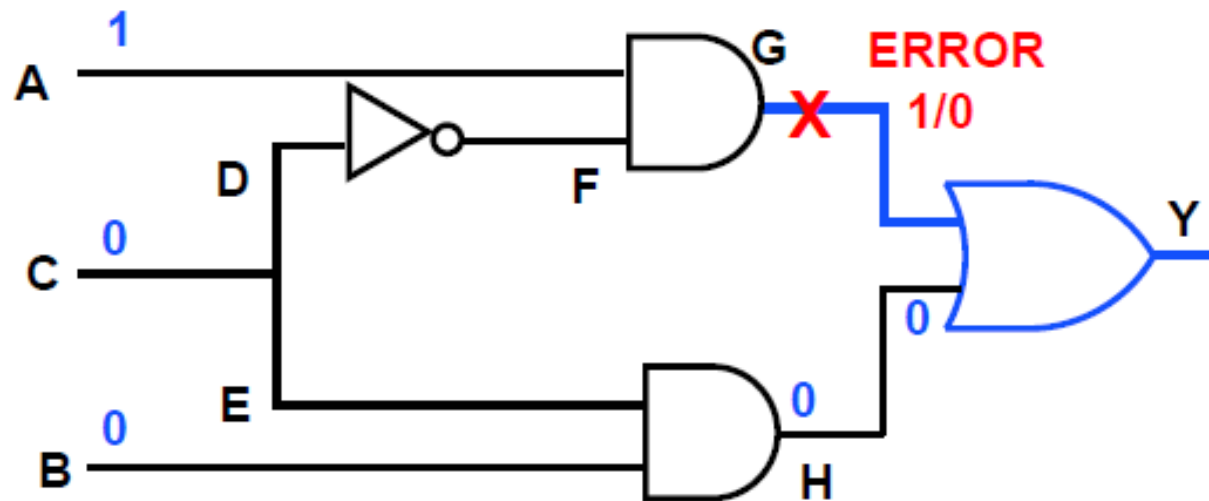


## Άσκηση 3 - Λύση

- Υπάρχουν 14 fault sites στο κύκλωμα. Οπότε υπάρχουν  $14 \times 2 = 28$  απλά σφάλματα μόνιμης τιμής.
- Πολλαπλά σφάλματα μόνιμης τιμής:  $(2 + 1)^{14} - 1 = 4782968$ .
- Περιορισμένα σφάλματα μόνιμης τιμής:  $2 \times (P_0 + F_0) + G_l - N_l$ 
  - $P_0 = 1, F_0 = 3, G_l = 10, N_l = 2$ .
  - Επομένως το πλήθος των περιορισμένων σφαλμάτων μόνιμης τιμής είναι:  $2 \times (1 + 3) + 10 - 2 = 16$ .

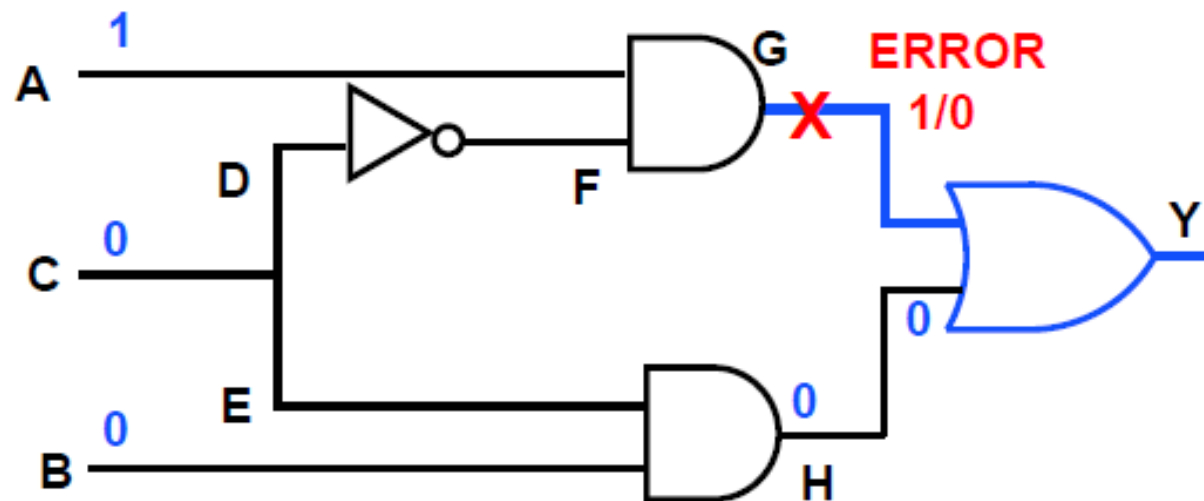
# Άσκηση 4

- Έστω το συνδυαστικό κύκλωμα της παρακάτω εικόνας. Το διάνυσμα δοκιμής 100 θα ανιχνεύσει το σφάλμα G stuck-at-0;



# Άσκηση 4 - Λύση

- Στην περίπτωση του fault-free κυκλώματος η έξοδος Y θα πάρει την τιμή 1.
- Στην περίπτωση του faulty κυκλώματος η έξοδος Y θα πάρει την τιμή 0
- Άρα το διάνυσμα δοκιμής 100 θα ανιχνεύσει το σφάλμα G stuck-at-0



# Άσκηση 5

- Έστω μια πύλη NAND 2 εισόδων A και B και εξόδου Z.
  - Ποιος είναι ο αριθμός των σφαλμάτων μόνιμης τιμής που μπορεί να εμφανιστούν;
  - Καταγράψτε σε έναν πίνακα όλα τα πιθανά σφάλματα μόνιμης τιμής που μπορεί να εμφανιστούν, την απόκριση της fault-free πύλης και την απόκριση της πύλης σε περίπτωση εμφάνισης κάθε σφάλματος
  - Ποια διανύσματα δοκιμής θα επιλέξετε για να καταστεί εφικτή η ανίχνευση όλων των σφαλμάτων;
  - Ποιο διάνυσμα δοκιμής έχει την υψηλότερη κάλυψη σφαλμάτων;

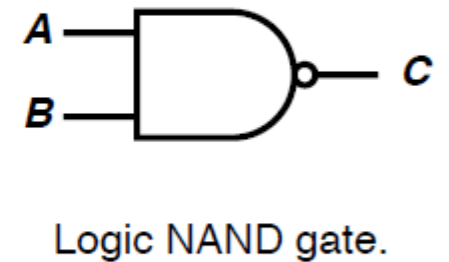
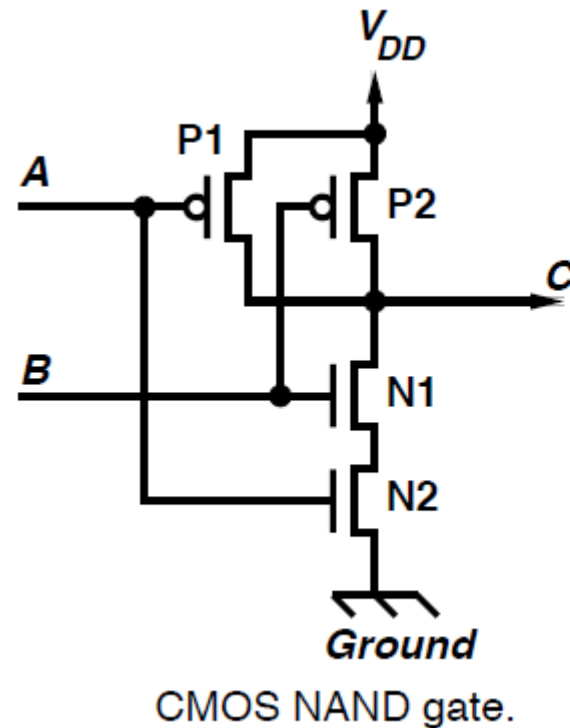
# Άσκηση 5 - Λύση

- Μπορεί να εμφανιστούν 2 x 3 σφάλματα μόνιμης τιμής.
- Θα απαιτηθούν τα διανύσματα δοκιμής 01, 10 και 11
- Το διάνυσμα δοκιμής 11 έχει την υψηλότερη κάλυψη σφαλμάτων ( $3/6 = 50\%$ )

Inputs	Fault-Free	Faulty Response					
AB	Response	A/ 0	B/ 0	Z/ 0	A/ 1	B/ 1	Z/ 1
00	1	1	1	0	1	1	1
01	1	1	1	0	0	1	1
10	1	1	1	0	1	0	1
11	0	1	1	0	0	0	1

# Άσκηση 6

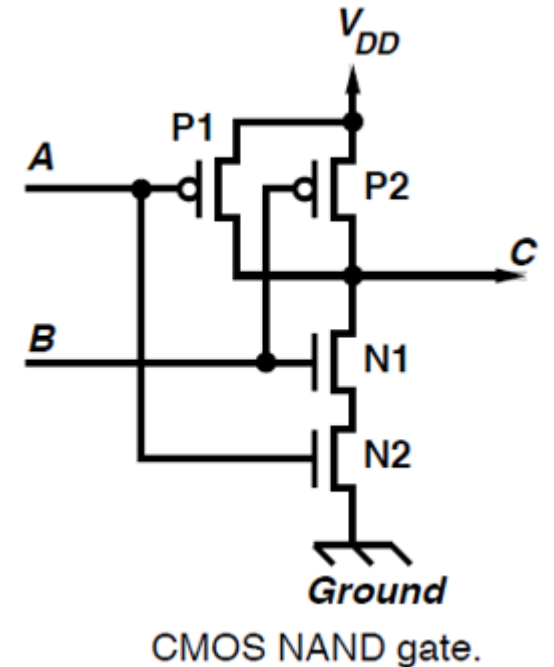
- Έστω μια πύλη CMOS NAND 2 εισόδων
- Βρείτε 2 διανύσματα δοκιμής για κάθε ένα stuck-open σφάλμα που μπορεί να εμφανιστεί σε κάθε τρανζίστορ.
- Υπάρχουν ισοδύναμα σφάλματα;





# Άσκηση 6 - Λύση

Σφάλμα	Διανύσματα Δοκιμής
P1 stuck open	11, 01
P2 stuck open	11, 10
N1 stuck open	01, 11 ή 10, 11 ή 00, 11
N2 stuck open	01, 11 ή 10, 11 ή 00, 11



Τα stuck open σφάλματα στα τρανζίστορ N1 και N2 είναι ισοδύναμα

# Πηγές

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

# Ερωτήσεις

