

Τεχνικές Υψηλής Ποιότητας Ελέγχου Ολοκληρωμένων Κυκλωμάτων

Σπυριδούλα Μπαλατσούκα

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΕΞΕΙΔΙΚΕΥΣΗΣ

— ♦ —

Ιωάννινα, Σεπτέμβριος 2009



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF IOANNINA



Τεχνικές Υψηλής Ποιότητας Ελέγχου Ολοκληρωμένων Κυκλωμάτων

Η
ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΕΞΕΙΔΙΚΕΥΣΗΣ

Υποβάλλεται στην

ορισθείσα από την Γενική Συνέλευση Ειδικής Σύγκλησης
του Τμήματος Πληροφορικής
Εξεταστική Επιτροπή

από την

Μπαλατσούκα Σπυριδούλα

ως μέρος των Υποχρεώσεων

για τη λήψη

του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ

ΜΕ ΕΞΕΙΔΙΚΕΥΣΗ ΣΤΙΣ ΤΕΧΝΟΛΟΓΙΕΣ-ΕΦΑΡΜΟΓΕΣ

Σεπτέμβριος 2009



ΒΙΒΛΙΟΘΗΚΗ
ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΙΩΑΝΝΙΝΩΝ



026000314173

ΕΠΙΣΤΗΜΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑ

ΠΡΟΠΤΗΤΙΑ

ΕΠΙΣΤΗΜΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑ
ΠΡΟΠΤΗΤΙΑ

ΕΠΙΣΤΗΜΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑ

ΕΠΙΣΤΗΜΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑ

ΕΠΙΣΤΗΜΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑ

ΕΠΙΣΤΗΜΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑ



ΑΦΙΕΡΩΣΗ

Η παρούσα διατριβή αφιερώνεται με πολύ αγάπη στους γονείς μου, Κωνσταντίνο και Χρυσούλα.



ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον κ. Καβουσιανό Χρυσοβαλάντη, Επίκουρο Καθηγητή του Τμήματος Πληροφορικής του Πανεπιστημίου Ιωαννίνων, για τη συνεχή και εποικοδομητική επίβλεψη της πορείας του μεταπτυχιακού μου, καθώς και για την ηθική υποστήριξη. Επίσης, ευχαριστώ και τον κ. Βασίλη Τενέντε, διδάκτορα του Τμήματος Πληροφορικής του Πανεπιστημίου Ιωαννίνων, για τη βοήθειά που μου πρόσφερε. Τέλος, ευχαριστώ τους γονείς μου και τα αδέρφια μου για την κατανόησή, βοήθεια και αγάπη τους.



ΠΕΡΙΕΧΟΜΕΝΑ

	Σελ
ΑΦΙΕΡΩΣΗ	ii
ΕΥΧΑΡΙΣΤΙΕΣ	iii
ΠΕΡΙΕΧΟΜΕΝΑ	iv
ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ	ivi
ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ	vii
ΕΠΕΞΗΓΗΣΕΙΣ ΣΥΜΒΟΛΙΣΜΩΝ	ix
ΠΕΡΙΛΗΨΗ	x
EXTENDED ABSTRACT IN ENGLISH	xii
ΚΕΦΑΛΑΙΟ 1. Εισαγωγή	1
1.1. Γενικά	1
1.2. Διάρθρωση της εργασίας	6
ΚΕΦΑΛΑΙΟ 2. Θεωρητικό Υπόβαθρο	7
2.1. Οι τεχνικές ελέγχου καθυστέρησης LOS και LOC	7
2.1.1. Γενικά	7
2.1.2. Σύντομη επισκόπηση των τεχνικών LOS και LOC	7
2.2. Τεχνικές X-Filling	10
2.2.1. Γενικά	10
2.2.2. Σύντομη επισκόπηση της τεχνικής Fill Adjacent	13
2.2.3. Σύντομη επισκόπηση της μεθόδου Preferred Fill	14
2.3. Η έννοια των Output Deviations	16
2.3.1. Γενικά	16
2.3.2. Συνοπτική αναφορά στα Output Deviations	17
2.4. Η μετρική weighted transition	23
ΚΕΦΑΛΑΙΟ 3. Η Προτεινόμενη Μέθοδος	25
3.1. Περιγραφή της προτεινόμενης μεθόδου	25
3.1.1. Θεωρητικό Υπόβαθρο	25
3.1.2. Η παραγωγή των υποψήφιων test vectors	29
3.1.3. Εκτίμηση και επιλογή των test vectors	33
3.2. Οι τεχνικές Fill Adjacent και Random Fill	37
3.3. Εκτίμηση του μέσου όρου των ενεργειών scan-in και scan-out	39
ΚΕΦΑΛΑΙΟ 4. Πειράματα	40
4.1. Πειράματα	40
4.1.1. Πειράματα για την κατανάλωση ενέργειας	41
4.1.2. Πειράματα για την ανίχνευση μη μοντελοποιημένων σφαλμάτων	45
ΚΕΦΑΛΑΙΟ 5. Συμπεράσματα	57
ΑΝΑΦΟΡΕΣ	58



ΠΑΡΑΡΤΗΜΑ Α	61
ΠΑΡΑΡΤΗΜΑ Β	68
ΠΑΡΑΡΤΗΜΑ Γ	77
ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ	92



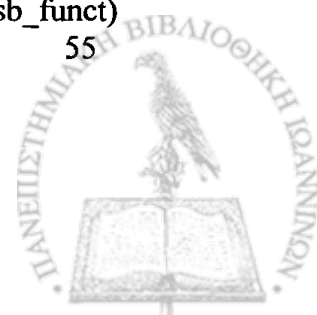
ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

Πίνακας	Σελ
Πίνακας 2.1 Πιθανότητες σημάτων	21
Πίνακας 2.2 Τα γεγονότα σφάλματος για είσοδο abcd=0000	22
Πίνακας 3.1 Συμπλήρωση των X bits	26
Πίνακας 3.2 Test cube T=XXX1XXX0XXX0XXXXX1	27
Πίνακας 4.1 Ποσοστό μείωσης ολικής μέσης ενέργειας σε σύγκριση με την RF (με χρήση του εργαλείου PrimePower of Synopsys)	42
Πίνακας 4.2 Ποσοστό μείωσης scan-in power σε σύγκριση με την RF (με χρήση της μετρικής weighted transition)	43
Πίνακας 4.3 Ποσοστό μείωσης scan-out power σε σύγκριση με την RF (με χρήση της τροποποιημένης μετρικής weighted transition)	44
Πίνακας 4.4 Ποσοστό μείωσης capture power σε σύγκριση με την RF (με χρήση του εργαλείου PrimePower of Synopsys)	44
Πίνακας 4.5 Ποσοστά ανίχνευσης σφαλμάτων (%)	49
Πίνακας A.1 s5378	62
Πίνακας A.2 s9234	62
Πίνακας A.3 s13207	63
Πίνακας A.4 s15850	63
Πίνακας A.5 s38417	64
Πίνακας A.6 s38584	64
Πίνακας A.7 ac97_ctrl	65
Πίνακας A.8 mem_ctrl	65
Πίνακας A.9 pci_bridge32	66
Πίνακας A.10 tv80	66
Πίνακας A.11 usb_funct	67
Πίνακας A.12 ethernet	67



ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

Σχήμα	Σελ
Σχήμα 2.1 Επισκόπηση των LOS και LOC	8
Σχήμα 2.2 Κυματομορφές των LOS και LOC	9
Σχήμα 2.3 Πύλες NAND και NOR	19
Σχήμα 2.4 Διανύσματα CL των πυλών NAND και NOR	19
Σχήμα 2.5 Η σχέση των σφαλμάτων stuck-at και των διανυσμάτων CL	20
Σχήμα 2.6 Παράδειγμα κυκλώματος	21
Σχήμα 3.1 Σχηματικό ελέγχου με c ισοζυγισμένες αλυσίδες scan	30
Σχήμα 3.2 Συνοπτικό διάγραμμα παραγωγής υποψήφιων test vectors (MFA/MFA+P)	33
Σχήμα 3.3 Συνοπτικό διάγραμμα παραγωγής test vectors από την FA*	38
Σχήμα 4.1 Κάλυψη σφαλμάτων τύπου transition delay για διάφορα L (s9234)	46
Σχήμα 4.2 Κάλυψη σφαλμάτων τύπου transition delay για διάφορα L (s5378)	46
Σχήμα 4.3 Κάλυψη σφαλμάτων τύπου transition delay για διάφορα L (s13207)	46
Σχήμα 4.4 Κάλυψη σφαλμάτων τύπου transition delay για διάφορα L (s15850)	47
Σχήμα 4.5 Κάλυψη σφαλμάτων τύπου transition delay για διάφορα L (s38417)	47
Σχήμα 4.6 Κάλυψη σφαλμάτων τύπου transition delay για διάφορα L (s38584)	47
Σχήμα 4.7 Κύκλωμα s38417	48
Σχήμα 4.8 Κύκλωμα s9234	48
Σχήμα 4.9 Ποσοστό ανίχνευσης σφαλμάτων transition-delay - ramp-up (s5378)	50
Σχήμα 4.10 Ποσοστό ανίχνευσης σφαλμάτων transition-delay - ramp-up (s9234)	50
Σχήμα 4.11 Ποσοστό ανίχνευσης σφαλμάτων transition-delay - ramp-up (s13207)	51
Σχήμα 4.12 Ποσοστό ανίχνευσης σφαλμάτων transition-delay - ramp-up (s15850)	51
Σχήμα 4.13 Ποσοστό ανίχνευσης σφαλμάτων transition-delay - ramp-up (s38417)	52
Σχήμα 4.14 Ποσοστό ανίχνευσης σφαλμάτων transition-delay - ramp-up (s38584)	52
Σχήμα 4.15 Ποσοστό ανίχνευσης σφαλμάτων transition-delay - ramp-up (ac97_ctrl)	53
Σχήμα 4.16 Ποσοστό ανίχνευσης σφαλμάτων transition-delay - ramp-up (mem_ctrl)	53
Σχήμα 4.17 Ποσοστό ανίχνευσης σφαλμάτων transition-delay - ramp-up (pci_bridge32)	54
Σχήμα 4.18 Ποσοστό ανίχνευσης σφαλμάτων transition-delay - ramp-up (tv89)	54
Σχήμα 4.19 Ποσοστό ανίχνευσης σφαλμάτων transition-delay - ramp-up (usb_funct)	55



Σχήμα 4.20 Ποσοστό ανίχνευσης σφαλμάτων transition-delay - ramp-up (ethernet)	55
Σχήμα Β.1 Ροή του TetraMax	71
Σχήμα Β.2 Ροή Προσομοίωσης	74
Σχήμα Β.3 Προσομοίωση για μοντέλα σφαλμάτων Transition και Bridging	76
Σχήμα Γ.1 PrimePower	77
Σχήμα Γ.2 Η διαδικασία δημιουργίας δραστηριότητας εναλλαγών	83
Σχήμα Γ.3 Εντολές toggle	84
Σχήμα Γ.4 Πολυπλέκτης	85
Σχήμα Γ.5 Αρχείο SAIF forward-annotation	86
Σχήμα Γ.6 Τύποι αρχείων	88



ΕΠΕΞΗΓΗΣΕΙΣ ΣΥΜΒΟΛΙΣΜΩΝ

Test pattern: ορίζεται μια ακολουθία από τιμές εισόδου και αναμενόμενες τιμές εξόδου, η οποία χρησιμοποιείται για τον έλεγχο μιας συσκευής για πιθανά σφάλματα ή ελαττώματα.

Test cube: ορίζεται ένα test pattern το οποίο περιέχει καθορισμένες τιμές '0', '1' και μη καθορισμένες τιμές 'X'.

Test vector: ορίζεται ένα test pattern το οποίο περιέχει μόνο καθορισμένα bits.

ATPG: η διαδικασία αυτόματης παραγωγής των test patterns.

Capture power: ορίζεται η ενέργεια που καταναλώνεται κατά τη λήψη των αποκρίσεων του κυκλώματος για το δοθέν διάνυσμα ελέγχου στις αλυσίδες scan.

Scan-in power: ορίζεται η ενέργεια που καταναλώνεται κατά την εισαγωγή με ολίσθηση του διανύσματος εισόδου στις αλυσίδες scan.

Scan-out power: ορίζεται η ενέργεια που καταναλώνεται κατά την εξαγωγή με ολίσθηση της απόκρισης του κυκλώματος για το δοθέν διάνυσμα ελέγχου από τις αλυσίδες scan.

Switching activity (ή δραστηριότητα εναλλαγών): ορίζεται ως η δραστηριότητα που παρατηρείται στο κύκλωμα, λόγω φόρτισης και αποφόρτισης των στοιχείων του κυκλώματος.

Stuck-at fault: ορίζεται το σφάλμα που προκύπτει όταν ένας κόμβος έχει μόνιμα είτε τη λογική τιμή 1 (stuck-at-1), είτε τη λογική τιμή 0 (stuck-at-0)

Transition delay fault: ορίζεται το σφάλμα που εμφανίζεται σε κόμβο με την ιδιότητα *slow-to-rise* ή *slow-to-fall*.

Bridging fault: ορίζεται το σφάλμα που προκαλεί δυο ανεξάρτητες γραμμές σε μια συσκευή να συνδεθούν ηλεκτρικά εξαιτίας επιπλέον υλικού ή ελαττωματικής χαρακτηριστικής.



ΠΕΡΙΛΗΨΗ

Μπαλατσούκα Σπυριδούλα του Κωνσταντίνου και της Χρυσούλας. MSc, Τμήμα Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Σεπτέμβριος 2009. Τεχνικές Υψηλής Ποιότητας Ελέγχου Ολοκληρωμένων Κυκλωμάτων. Επιβλέπωντας: Χρυσοβαλάντης Καβουσιανός.

Ο σειριακός έλεγχος των ολοκληρωμένων κυκλωμάτων αποτελεί πλέον επιβεβλημένο στάδιο, προκειμένου να ανιχνευθούν πιθανά ελαττώματα, εξασφαλίζοντας κατ' αυτό τον τρόπο την προώθηση στην αγορά όσο το δυνατόν πιο αξιόπιστων κυκλωμάτων. Ωστόσο, οι σημερινές τάσεις των υπομικρομικών τεχνολογιών (VDSM) επιβάλλουν τη συνεχή σμίκρυνση των συσκευών και κατά συνέπεια όχι μόνο αυξάνεται η πολυπλοκότητα κατασκευής, αλλά ανακύπτουν και καινούργιοι τύποι ελαττωμάτων γεγονός που καθιστά το κόστος ελέγχου ιδιαίτερα υψηλό. Οι παραδοσιακές τεχνικές ελέγχου ολοκληρωμένων κυκλωμάτων καταφέρνουν να μειώσουν το κόστος ελέγχου με το να ανιχνεύουν ταυτόχρονα όσο το δυνατόν περισσότερα ελαττώματα. Όμως, αυτή η τακτική επιφέρει αύξηση στην κατανάλωση ενέργειας κατά τη διάρκεια του ελέγχου. Μάλιστα, μελέτες απέδειξαν ότι η κατανάλωση ενέργειας κατά τη διάρκεια του ελέγχου είναι πολλές φορές μεγαλύτερη από ότι όταν το κύκλωμα βρίσκεται σε κανονική λειτουργία. Συνεπώς, εξαιτίας της αυξημένης κατανάλωσης ενέργειας συνεπάγονται διάφορα προβλήματα, όπως η αυξημένη παραγωγή θερμότητας, η οποία αυξάνει τη θερμοκρασία του ολοκληρωμένου πέρα από τα επιτρεπτά όρια και έτσι η διαδικασία ελέγχου καθίσταται ως αναξιόπιστη. Επιπλέον, παρατηρείται και πτώση τάσης η οποία όχι μόνο επηρεάζει τις καθυστερήσεις των πυλών, αλλά και αυξάνει την πιθανότητα αποτυχίας του διανύσματος ελέγχου όταν εφαρμόζεται έλεγχος at-speed. Είναι αναπόφευκτο λοιπόν μη ελαττωματικά ολοκληρωμένα κυκλώματα να χαρακτηρίζονται εσφαλμένα ως ελαττωματικά.



Προκειμένου να μειωθεί η κατανάλωση ενέργειας κατά τη διάρκεια ελέγχου (shift/capture power) έχουν εδραιωθεί ποικίλες μέθοδοι ελέγχου ολοκληρωμένων κυκλωμάτων, που εντάσσονται στην κατηγορία των δομικών ή των αλγοριθμικών. Η εργασία εστιάζεται στην κατηγορία των αλγοριθμικών και συγκεκριμένα στις διάφορες μεθόδους συμπλήρωσης των μη καθορισμένων bits (X) των διανυσμάτων ελέγχου που εφαρμόζονται στα κυκλώματα. Βασικό μειονέκτημά των ήδη καθιερωμένων μεθόδων είναι ότι μπορεί μεν να επιφέρουν χαμηλή κατανάλωση ισχύος, αλλά ταυτόχρονα επιφέρουν και ποσοστό ανίχνευσης μη μοντελοποιημένων σφαλμάτων χαμηλότερο από τη μέθοδο *Random Fill*. Προκειμένου να αντισταθμιστούν αυτά τα μειονεκτήματα, προτείνεται μια καινούργια μέθοδος, η MFA, μαζί με μια τροποποιημένη έκδοσή της, η MFA+P. Συγκεκριμένα, η προτεινόμενη μέθοδος MFA επιτυγχάνει μείωση της ενέργειας ολίσθησης (*shift power*), με ταυτόχρονη επιβολή περιορισμών στην μέγιστη κατανάλωση ενέργειας κατά τη λήψη των αποκρίσεων του κυκλώματος, και η μείωση αυτή είναι συγκρίσιμη με αυτή που επιφέρει η μέθοδος *Fill Adjacent*. Ταυτόχρονα, η μέθοδος MFA επιφέρει ποσοστά ανίχνευσης ελαττωμάτων που πλησιάζουν και σε πολλές περιπτώσεις υπερβαίνουν αυτά της μεθόδου *Random Fill*, χωρίς βέβαια να επέλθει αύξηση των διανυσμάτων ελέγχου. Σε σύγκριση με την MFA, η τροποποιημένη έκδοσή της MFA+P επιφέρει αξιοσημείωτες βελτιώσεις στα ποσοστά ανίχνευσης σφαλμάτων, ενώ αυξάνει την κατανάλωση ενέργειας ανάλογα με την τιμή του P. Τα προτερήματα των δυο προτεινόμενων μεθόδων MFA και MFA+P επαληθεύονται στα αποτελέσματα των εξομοιώσεων των μεγαλύτερων κυκλωμάτων των κατηγοριών ISCAS και IWLS, αντίστοιχα.



EXTENDED ABSTRACT IN ENGLISH

Balatsouka, Spyridoula. MSc, Computer Science Department, University of Ioannina, Greece. September, 2009. High Quality Techniques of Testing Integrated Circuits. Supervisor: Xrysovalantis Kavousianos

Nowadays, scan testing of integrated circuits is widely used for defect screening and quality assurance. However, in the case of the very-deep submicron (VDSM) technologies, the increased complexity and the new types of defects lead to undesirable increase in the cost of testing. Traditional testing techniques decrease test costs by concurrently targeting as many defects as possible, leading thus to elevated test power consumption, which can be several times higher than that in functional mode. Needless to say, these very high power demands during scan testing increase heat dissipation which elevates the temperature of the chip beyond acceptable limits and undermines the reliability of testing. Another side-effect associated with the troublesome issue of the excessive power dissipation is IR-drop on the current supply which severely affects the gate delays and imposes high probability of failure at speed testing on the test vector. As a consequence, many good chips may be erroneously identified as defective, leading thus to yield loss.

Numerous methods have been proposed in the literature for limiting power consumption during test application, which can be further categorized as being either structural or algorithmic. Among the algorithmic methods, there are various X-filling methods that have been proposed for reducing the shift and/or capture power in scan testing. The major drawback of these methods is that X-filling for low power results in lower defect coverage than random-fill. With the view to overcome these obstacles, we propose a unified low-power and defect-aware X-filling method for scan testing, called MFA, along with an enhanced version of it, called MFA+P. The MFA method reduces shift power under constraints on the peak power during response capture, and



the power reduction is comparable to that of the Fill-Adjacent X-filling method. At the same time, this approach provides high defect coverage, which approaches and in many cases is higher than that achieved by random-fill, without increasing the pattern count. Compared to MFA, the MFA+P method accomplishes further improvements in defect coverage at the cost of a small increase in the power consumption. The advantages of the proposed methods are demonstrated with simulation results for the largest ISCAS and the IWLS benchmark circuits.



ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ

1.1 Γενικά

1.2 Διάρθρωση της εργασίας

1.1. Γενικά

Η ανάγκη για όσο το δυνατόν πιο αξιόπιστα ηλεκτρονικά προϊόντα έχει καταστήσει σημαντική τη διαδικασία ελέγχου των ολοκληρωμένων κυκλωμάτων (IC) με χρήση των αλυσίδων scan (σειριακός έλεγχος). Οι σύγχρονες τεχνολογίες (very-deep submicron technologies, VDSM) έχουν επιφέρει την σμίκρυνση συσκευών και κατά συνέπεια την ενσωμάτωση πολλών ολοκληρωμένων κυκλωμάτων σε μια επιφάνεια, γεγονός που έχει προκαλέσει την εμφάνιση νέων τύπων ελαττωμάτων επιβαρύνοντας κατ' αυτό τον τρόπο το κόστος ελέγχου. Πλέον, τα σύνολα ελέγχου που χρησιμοποιούνται για την ανίχνευση μικρών καθυστερήσεων είναι 2-5 φορές μεγαλύτερα σε μέγεθος από τα σύνολα ελέγχου που χρησιμοποιούνται για την ανίχνευση ελαττωμάτων τύπου stuck-at. Επιπρόσθετα, είναι αναγκαίο να επισημανθεί ότι σύμφωνα με εκτενείς μελέτες, τα ολοκληρωμένα κυκλώματα καταναλώνουν πολύ περισσότερη ενέργεια όταν βρίσκονται στη φάση ελέγχου από ότι όταν βρίσκονται σε κανονική λειτουργία [8]. Μάλιστα, στην τεχνολογία CMOS η κατανάλωση ενέργειας είναι ανάλογη της δραστηριότητας των εναλλαγών, που εξελίσσεται στο εσωτερικό του ολοκληρωμένου κυκλώματος. Έχει αποδειχθεί ότι στη φάση ελέγχου υπάρχει αυξημένη δραστηριότητα εναλλαγών σε σύγκριση με αυτή στη φάση κανονικής λειτουργίας. Αυτή η κατάσταση οφείλεται στο γεγονός ότι τα συνεχόμενα ψευδοτυχαία διανύσματα ελέγχου που εφαρμόζονται στο κύκλωμα κατά τη φάση ελέγχου έχουν χαμηλό βαθμό συσχέτισης μεταξύ τους, σε σύγκριση με τα συνεχόμενα διανύσματα εισόδου που εφαρμόζονται στην κανονική λειτουργία.



Η διαδικασία ελέγχου των ολοκληρωμένων κυκλωμάτων με χρήση των αλυσίδων scan απαρτίζεται γενικά από τρία στάδια. Στο 1^ο στάδιο, εισάγεται με ολίσθηση στις αλυσίδες scan το διάνυσμα ελέγχου, που θα εφαρμοστεί στο ελεγχόμενο κύκλωμα, προκαλώντας με αυτό το τρόπο εναλλαγές στα κύτταρα των αλυσίδων scan. Στο 2^ο στάδιο, που αποτελείται από έναν κύκλο (ή δυο κύκλους στην περίπτωση ελέγχου στην ταχύτητα λειτουργίας), αποθηκεύεται στα κύτταρα των αλυσίδων scan η απόκριση που δίνει το ελεγχόμενο κύκλωμα για το δοθέν διάνυσμα ελέγχου. Έτσι, στις αλυσίδες scan προκαλείται δραστηριότητα εναλλαγών, η οποία βέβαια μπορεί να υπολογιστεί με βάση την απόσταση Hamming ανάμεσα στο δοθέν διάνυσμα ελέγχου και την αντίστοιχη απόκριση. Στο 3^ο στάδιο, εξάγεται με ολίσθηση από τις αλυσίδες scan η απόκριση, προκαλώντας πάλι εναλλαγές στα κύτταρα των αλυσίδων scan.

Συνεπώς, η κατανάλωση ενέργειας κατά τη φάση ελέγχου ορθότητας ενός κυκλώματος διακρίνεται στην κατανάλωση ενέργειας κατά τη φάση ολίσθησης (1^ο και 3^ο στάδιο) και στην κατανάλωση ενέργειας κατά τη φάση της λήψης της απόκρισης του κυκλώματος για το δοθέν διάνυσμα ελέγχου (2^ο στάδιο). Στο 1^ο και 3^ο στάδιο, πέρα από τις εναλλαγές που σημειώνονται στα κύτταρα των αλυσίδων scan, υπογραμμίζεται ότι, λόγω της ολίσθησης, προκαλούνται αναπόφευκτα εναλλαγές και στις τιμές των στοιχείων του συνδυαστικού τμήματος του ελεγχόμενου κυκλώματος. Η συσσωρευμένη δραστηριότητα εναλλαγών, που εντοπίζεται σε αυτά τα δυο στάδια, επιφέρει λοιπόν αυξημένη μέση κατανάλωση ενέργειας και κατ' επέκταση αυξημένη παραγωγή θερμότητας. Ως επακόλουθο, είναι δυνατόν η θερμοκρασία του τσιπ να υπερβεί τα επιτρεπτά όρια και έτσι η διαδικασία ελέγχου να καταστεί ως αναξιόπιστη. Από την άλλη πλευρά, η σχετικά χαμηλή συσχέτιση ανάμεσα στα διανύσματα ελέγχου και τις αντίστοιχες αποκρίσεις τους επιφέρει αυξημένη δραστηριότητα εναλλαγών στο 2^ο στάδιο. Συνήθως, η δραστηριότητα εναλλαγών στο 2^ο στάδιο είναι υψηλότερη κατά τη φάση ελέγχου ορθότητας ενός ολοκληρωμένου κυκλώματος παρά κατά την φάση της κανονικής του λειτουργίας. Οι παρενέργειες που ανιχνεύονται στην προκειμένη περίπτωση, είναι η πτώση τάσης η οποία όχι μόνο επηρεάζει τις καθυστερήσεις των πυλών, αλλά και αυξάνει την πιθανότητα αποτυχίας του διανύσματος ελέγχου όταν εφαρμόζεται ταχύς έλεγχος [19]. Κατά συνέπεια, δεν είναι



λίγες οι περιπτώσεις που μη ελαττωματικά τσιπς χαρακτηρίζονται εσφαλμένα ως ελαττωματικά και έτσι να διακόπτεται η προώθησή τους στο εμπόριο.

Γενικά, έχουν αναπτυχθεί ποικίλες μέθοδοι ελέγχου ορθότητας ολοκληρωμένων κυκλωμάτων που στοχεύουν στον περιορισμό της κατανάλωσης ενέργειας. Μερικές ασχολούνται μόνο με τη δραστηριότητα εναλλαγών κατά τη φάση ολίσθησης [1, 2, 5-7, 10, 12, 15], ενώ άλλες μόνο με τη δραστηριότητα εναλλαγών κατά τη φάση της λήψης των αποκρίσεων του ελεγχόμενου κυκλώματος για τα δοθέντα διανύσματα ελέγχου [4, 13, 17, 23-27]. Ωστόσο, αυτό που παρατηρείται είναι ότι οι περισσότερες μέθοδοι εστιάζονται στον περιορισμό της δραστηριότητας εναλλαγών που εντοπίζεται σε ένα μόνο στάδιο. Με άλλα λόγια, ελάχιστες είναι οι μέθοδοι που επικεντρώνονται στη ταυτόχρονη μείωση των δραστηριοτήτων εναλλαγών που συμβαίνουν και στα τρία προαναφερθέντα στάδια [3, 11, 14, 16, 18].

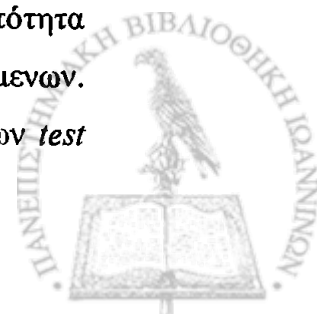
Στη συνέχεια, οι μέθοδοι που χρησιμοποιούνται για τη μείωση της δραστηριότητας εναλλαγών εντάσσονται σε δυο κατηγορίες: τις δομικές και τις αλγοριθμικές. Στην περίπτωση των δομικών μεθόδων, εφαρμόζονται δομικές τροποποιήσεις στις αλυσίδες scan [4, 11] ή στη λογική αποσυμπίεσης [2, 5, 6, 7, 12, 15]. Όμως, αυτές οι τεχνικές μειονεκτούν διότι επιφέρουν αυξημένο όγκο δεδομένων ελέγχου, καθώς και πιο περίπλοκους αλγόριθμους συμπίεσης. Στην περίπτωση των αλγοριθμικών μεθόδων, τροποποιείται η διαδικασία *Automatic Test Pattern Generation (ATPG)* είτε με την παραγωγή πιο αποτελεσματικών *test cubes*, όσον αφορά την ενέργεια, για την ανίχνευση συγκεκριμένων ελαττωμάτων [23, 24], είτε με τη στατική συμπίεση των *test cubes* σύμφωνα με κάποιο κατάφλι ενέργειας [18]. Μάλιστα, στην κατηγορία των αλγοριθμικών μεθόδων εντάσσονται και οι γνωστές ως *X-filling* μέθοδοι που εκμεταλλεύονται τα X bits των *test cubes*. Συγκεκριμένα, αναθέτουν τέτοιες τιμές στα X bits ώστε να σημειώνεται χαμηλή κατανάλωση ενέργειας. Οι τεχνικές *X-filling* δεν επηρεάζουν δραματικά τη διαδικασία *ATPG* και επιπλέον δεν επιβάλλουν τροποποιήσεις τόσο στις αλυσίδες scan, όσο και στο ελεγχόμενο κύκλωμα. Ιδίως αυτές οι μέθοδοι μπορούν να συνδυαστούν και με άλλες τεχνικές παρέχοντας έτσι κάποιες επιθυμητές βελτιώσεις



Περισσότερες από τις τεχνικές για τη μείωση της δραστηριότητας εναλλαγών πετυχαίνουν μεν μια επιθυμητή μείωση της κατανάλωσης ενέργειας, αλλά παράλληλα μειώνουν και το ποσοστό ανίχνευσης σφαλμάτων. Παλιότερα, προκειμένου να παρατηρείται μειωμένη κατανάλωση ενέργειας, είχε προταθεί να υλοποιείται η διαδικασία ελέγχου σε χαμηλότερες συχνότητες από αυτές της κανονικής λειτουργίας. Ωστόσο, αυτή η πρόταση έφερε βελτίωση μόνο στην παραγωγή θερμότητας και ταυτόχρονα ο χρόνος εφαρμογής αυξήθηκε. Όμως, οι σημερινές τεχνολογίες VDSM απαιτούν ελέγχους (at-speed testing) σε συχνότητες ίδιες με τις συχνότητες της κανονικής λειτουργίας για την ανίχνευση αργών κυκλωμάτων. Έτσι, δεν επιτρέπεται η χρήση χαμηλότερων συχνοτήτων στη φάση ελέγχου, γιατί έτσι είναι πιθανόν να συγκαλυφθούν διάφορα σφάλματα καθυστέρησης μειώνοντας κατά αυτό τον τρόπο το ποσοστό ανίχνευσης σφαλμάτων.

Μια αρκετά διαδεδομένη μέθοδος *X-filling* είναι η λεγόμενη μέθοδος *Fill Adjacent* [1]. Η μέθοδος αυτή στοχεύει στη μείωση της ενέργειας που καταναλώνεται κατά την εισαγωγή με ολίσθηση του διανύσματος ελέγχου στις αλυσίδες scan. Παράλληλα όμως, επιτυγχάνει και αισθητή μείωση της ενέργειας κατά την εξαγωγή με ολίσθηση της απόκρισης από τις αλυσίδες scan, λόγω του ότι υπάρχει υψηλή συσχέτιση μεταξύ αυτών των δυο τύπων ενεργειών [3]. Επιπλέον, η μέθοδος *Fill Adjacent* σε συνδυασμό με μεθόδους που στοχεύουν στη μείωση της κατανάλωσης ενέργειας κατά τη λήψη της απόκρισης που δίνει το ελεγχόμενο κύκλωμα για το δοθέν διάνυσμα ελέγχου, όπως η *Preferred Fill* [16, 17], μπορεί να οδηγήσει σε πιο αξιοσημείωτες μειώσεις της κατανάλωσης ενέργειας.

Ωστόσο, μπορεί μεν οι μέθοδοι *X-filling* να οδηγούν σε αισθητή μείωση της ενέργειας, είτε κατά τη φάση ολίσθησης είτε κατά τη φάση της λήψης της απόκρισης του κυκλώματος για το δοθέν διάνυσμα ελέγχου, αλλά σημειώνεται χαμηλό ποσοστό ανίχνευσης μη μοντελοποιημένων σφαλμάτων, λόγω της συμπλήρωσης των μη καθορισμένων τιμών X υπό τον περιορισμό κάποιου κριτηρίου. Από την άλλη πλευρά, συνηθίζεται κατά την παραγωγή διανυσμάτων *ATPG* οι μη καθορισμένες τιμές X να συμπληρώνονται τυχαία, προκειμένου να αυξηθεί η δυνατότητα ανίχνευσης τόσο των αναμενόμενων σφαλμάτων, όσο και των μη αναμενόμενων. Επίσης, αρκετές μέθοδοι *X-filling* καταφεύγουν στην αύξηση του πλήθους των *test*



patterns για να πετύχουν μεγαλύτερα ποσοστά ανίχνευσης μη μοντελοποιημένων σφαλμάτων.

Αυτό που πρόκειται λοιπόν να παρουσιαστεί είναι μια μέθοδος *X-filling* που βασίζεται στη συμπλήρωση των μη καθορισμένων τιμών X ενός *test cube* με διάφορους τρόπους. Συγκεκριμένα, η προτεινόμενη μέθοδος πετυχαίνει τους εξής επιθυμητούς στόχους:

- Υψηλή μείωση της κατανάλωσης ενέργειας κατά την ολίσθηση στη φάση ελέγχου, η οποία προσεγγίζει τη μείωση που επιφέρει η γνωστή μέθοδος *Fill Adjacent*.
- Αποφυγή της υπέρβασης κάποιου προκαθορισμένου ορίου από τη δραστηριότητα εναλλαγών κατά τη φάση της λήψης της απόκρισης του κυκλώματος για το δοθέν διάνυσμα ελέγχου.
- Αύξηση του ποσοστού ανίχνευσης σφαλμάτων, η οποία προσεγγίζει και μερικές φορές υπερβαίνει το ποσοστό που επιτυγχάνεται από τη μέθοδο της τυχαίας συμπλήρωσης των μη καθορισμένων τιμών X , χωρίς βέβαια να επέλθει αύξηση στο πλήθος των *test patterns*.
- Παροχή ενός *tradeoff* ανάμεσα στη μείωση ενέργειας και στην ανίχνευση σφαλμάτων, το οποίο συμβαδίζει με τις απαιτήσεις κάθε κυκλώματος.

Η προτεινόμενη μέθοδος *X-filling* συμπληρώνει τις μη καθορισμένες τιμές X των *test cubes* με διάφορους τρόπους, προκειμένου να κατοχυρωθεί η υψηλή μείωση της κατανάλωσης ενέργειας κατά την ολίσθηση σε συνδυασμό με κάποιες λειτουργικές προδιαγραφές της μέγιστης επιτρεπτής ενέργειας. Έτσι, ανάμεσα στα προκύπτοντα *test vectors* επιλέγεται αυτό που βελτιώνει την ανίχνευση σφαλμάτων. Για την αξιολόγηση των πιο αποτελεσματικών *test vectors* χρησιμοποιείται η έννοια του *output deviation* [21]. Γενικά, η έννοια του *output deviation* σχετίζεται με την πιθανότητα τα *test vectors* να ανιχνεύουν διάφορα σφάλματα από όλα τα μοντέλα σφαλμάτων. Η ποιότητα του ελέγχου των κυκλωμάτων είναι σαφώς καλύτερη όταν οι μέθοδοι ελέγχου δε στηρίζονται σε ένα μόνο μοντέλο σφαλμάτων [22]. Μέσω εκτενών πειραμάτων, που διεξήχθησαν στα κυκλώματα των κατηγοριών ISCAS και IWLS, διαφαίνεται η αποτελεσματικότητα της προτεινόμενης μεθόδου. Η μέθοδος αυτή αποσκοπεί κατ' εξοχήν όχι μόνο στην μείωση της κατανάλωσης ενέργειας, αλλά και σε ένα αξιολόγο ποσοστό ανίχνευσης μη μοντελοποιημένων σφαλμάτων.



1.2. Διάρθρωση της εργασίας

Στην εργασία αυτή θα ασχοληθούμε με δυο προτεινόμενες μεθόδους, που στοχεύουν τόσο στη μείωση της κατανάλωσης ενέργειας, όσο και στην εξασφάλιση ικανοποιητικών ποσοστών ανίχνευσης σφαλμάτων/ελαττωμάτων. Αρχικά, στο Κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο, δηλαδή οι τεχνικές ελέγχου LOC και LOS, κάποιες τεχνικές X-Filling, η έννοια των Output Deviations καθώς και η μέθοδος υπολογισμού της δραστηριότητας σε ένα κύκλωμα. Στο Κεφάλαιο 3 πραγματευόμαστε τις δυο προτεινόμενες μεθόδους, μαζί με μια τροποποίηση της καθιερωμένης τεχνικής *Fill Adjacent* και αναφέρουμε τον τρόπο που ακολουθείται στη παρούσα εργασία για την εκτίμηση των *scan-in power* και *scan-out power* ξεχωριστά. Τα πειράματα και τα σχετικά αποτελέσματα που καθιστούν τις προτεινόμενες μεθόδους ως αξιοποιήσιμες παρουσιάζονται στο Κεφάλαιο 4. Στο Κεφάλαιο 5 περιέχονται απλώς τα συμπεράσματα. Στο Παράρτημα Α παραθέτονται συγκεντρωμένα σε πίνακες τα αποτελέσματα όλων των πειραμάτων για όλα τα κυκλώματα. Τέλος, στα Παραρτήματα Β και Γ παρουσιάζονται αντίστοιχα τα εργαλεία TetraMax και PrimePower του Synopsys, τα οποία χρησιμοποιήθηκαν για τη διεξαγωγή των πειραμάτων σχετικά με την εκτίμηση του ποσοστού ανίχνευσης μη μοντελοποιημένων σφαλμάτων και της κατανάλωσης ενέργειας.



ΚΕΦΑΛΑΙΟ 2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1 Οι τεχνικές ελέγχου καθυστέρησης LOS και LOC

2.2 Τεχνικές X-Filling

2.3 Η έννοια των Output Deviations

2.4 Η μετρική weighted transition

2.1. Οι τεχνικές ελέγχου καθυστέρησης LOS και LOC

2.1.1. Γενικά

Με την ολοένα και αυξανόμενη ανάπτυξη των τεχνολογιών της τάξης του νανόμετρου, τα κατασκευαστικά ελαττώματα που προκαλούν χρονικά σφάλματα έχουν επιστήσει την προσοχή. Πλέον, οι υπάρχουσες τεχνικές ελέγχου stuck-at δεν δύνανται να ελέγξουν εξ' ολοκλήρου τη λειτουργικότητα των ολοκληρωμένων κυκλωμάτων. Επίσης, άλλες παραδοσιακές τεχνικές ελέγχου, εξαιτίας της σημερινής κλίμακας των κυκλωμάτων, αντιμετωπίζουν την αυξημένη οικονομική επιβάρυνση. Επίσης, στα πρότυπα SOC (system-on-chip) η περιορισμένη πρόσβαση στα εσωτερικά cores καθιστά αδύνατο τον λειτουργικό έλεγχο. Η πιο πρακτική προσέγγιση για τον έλεγχο καθυστέρησης σε μεγάλα SOCs, είναι ο έλεγχος βασιζόμενος στις αλυσίδες scan μιας και έτσι αυξάνεται η δυνατότητα ελέγχου και παρατήρησης των εσωτερικών σημάτων στα SOCs.

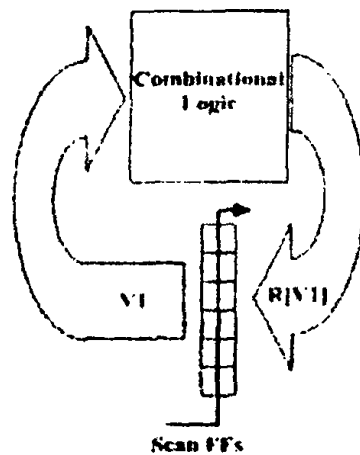
2.1.2. Σύντομη επισκόπηση των τεχνικών ελέγχου καθυστέρησης LOS και LOC

Ο έλεγχος για καθυστέρηση απαιτεί την εφαρμογή δυο test vectors προκειμένου να δημιουργηθούν οι λογικές μεταβάσεις και έτσι να διαδοθούν στο κύκλωμα. Αυτές οι



μεταβάσεις ελέγχονται για εκτεταμένες καθυστερήσεις πέρα από την περίοδο του ρολογιού. Επιπλέον, λόγω των κατασκευαστικών περιορισμών, χρησιμοποιούνται δυο τεχνικές η launch-on-shift (LOS) και η launch-on-capture (LOC). Στην ουσία, μόνο η LOC εφαρμόζεται μιας και η LOS απαιτεί υψηλής ταχύτητας σήμα scan enable, το οποίο είναι μεν ακριβό για να εφαρμοστεί και δεν υποστηρίζεται από διάφορα πρότυπα βασισμένα στις αλυσίδες scan. Πάντως, η κάλυψη σφαλμάτων καθυστέρησης από την τεχνική LOC είναι μέτρια, ενώ η τεχνική LOS επιτυγχάνει υψηλή κάλυψη με αρκετά λιγότερα test vectors.

Στο Σχήμα 2.1 παρουσιάζεται η γενική ιδέα του ελέγχου καθυστέρησης βασισμένου στην αλυσίδα scan, χρησιμοποιώντας δυο test patterns $\{V1, V2\}$.

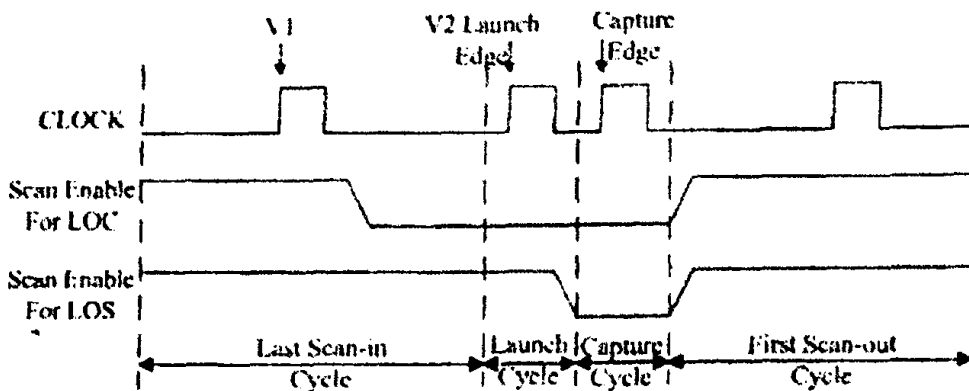


Σχήμα 2.1 Επισκόπηση της τεχνικής scan

Το πρώτο διάνυσμα $V1$ εισάγεται μέσα στην αλυσίδα scan και χρησιμοποιείται για την αρχικοποίηση των λογικών τιμών των εισόδων του συνδυαστικού κυκλώματος (CUT). Το δεύτερο διάνυσμα $V2=R[V1]$ χρησιμοποιείται για να προωθήσει τις μεταβάσεις από τις εισόδους στις εξόδους του κυκλώματος, οι οποίες συλλαμβάνονται πίσω στην αλυσίδα scan. Ανάλογα με το πώς προκύπτει το διάνυσμα $V2$, έχουμε δυο τεχνικές ελέγχου καθυστέρησης την skewed-load και την broad-side. Στην περίπτωση της skewed-load, η οποία είναι ευρέως γνωστή ως LOS, το δεύτερο διάνυσμα $V2$ προκύπτει από το πρώτο διάνυσμα $V1$ εφαρμόζοντας ολίσθηση του ενός bit. Ενώ στην περίπτωση της broad-side, η οποία είναι ευρέως γνωστή ως LOC, το δεύτερο διάνυσμα $V2$ είναι η απόκριση του κυκλώματος στο πρώτο



διάνυσμα V1. Στο Σχήμα 2.2 παρουσιάζονται οι κυματομορφές για τις τεχνικές LOS και LOC, αντίστοιχα.



Σχήμα 2.2 Κυματομορφές των LOS και LOC

Το σήμα scan enable παίρνει την τιμή 0 για την κανονική λειτουργία της αλυσίδας scan και 1 για τη λειτουργία ολίσθησης. Έτσι, όσο το διάνυσμα V1 ολισθαίνει μέσα στην αλυσίδα scan το σήμα scan enable πρέπει να έχει τη τιμή 1. Στο Σχήμα 2.2 εικάζεται ότι τα flip-flops είναι θετικά ακμοπυροδότητα και φαίνεται ο τελευταίος παλμός του ρολογιού με βάση τον οποίον ολοκληρώνεται η είσοδος του V1 στην αλυσίδα scan και έτσι είναι διαθέσιμο στο κύκλωμα στην επόμενη θετική ακμή του ρολογιού.

Για την τεχνική LOC, το σήμα scan enable παίρνει στη συνέχεια τη τιμή 0 και πρέπει να περάσει ένα διάστημα μέχρι να έρθουν οι δυο παλμοί του ρολογιού. Μάλιστα, στον πρώτο παλμό συλλαμβάνεται στην αλυσίδα scan το διάνυσμα V2, που είναι η απόκριση του κυκλώματος για το διάνυσμα εισόδου V1, και στο δεύτερο παλμό συλλαμβάνεται η απόκριση του κυκλώματος για το διάνυσμα εισόδου V2. Ο χρόνος, που μεσολαβεί ανάμεσα σε αυτές τις 2 θετικές ακμές του ρολογιού, πρέπει να συμβαδίζει με τον ρυθμό του λειτουργικού ρολογιού προκειμένου να εξασφαλίζεται ότι το κύκλωμα δίνει τις σωστές αποκρίσεις εντός της περιόδου του κανονικού ρολογιού.



Για την τεχνική LOS, το σήμα scan enable παραμένει στη τιμή 1 για ένα ακόμα παλμό ρολογιού μετά την εισαγωγή του V1 για να γίνει άλλη μια ολίσθηση και να προκύψει το V2, το οποίο περνάει στις εξόδους των flip-flops. Εκείνη τη στιγμή το σήμα scan enable πρέπει γρήγορα να αλλάξει σε 0 για να μπορέσει με την θετική ακμή του ρολογιού η αλυσίδα scan να πάρει την απόκριση του κυκλώματος για το διάστημα V2. Επειδή το σήμα scan enable πρέπει να εναλλάξει τιμές στο διάστημα ανάμεσα στις δυο θετικές ακμές, είναι σημαντικό το σήμα αυτό να φτάσει σε όλα τα flip-flops εντός των χρονικών ορίων. Έτσι, το σήμα scan enable, στην περίπτωση της τεχνικής LOS, καθίσταται χρονικά κρίσιμο και κατ' επέκταση ακριβό. Γι' αυτό και η συνηθέστερη τεχνική που εφαρμόζεται για τους ελέγχους καθυστέρησης είναι η LOC.

2.2. Τεχνικές X-Filling

2.2.1. Γενικά

Η εύρεση κατάλληλων μηχανισμών για τον έλεγχο της ορθότητας των ολοκληρωμένων κυκλωμάτων (IC) αποτελεί το επίκεντρο του ενδιαφέροντος. Αρχικά, θεωρούνταν επαρκής η εφαρμογή ελέγχων με αλυσίδες scan χαμηλής ταχύτητας. Όμως, οι σύγχρονες τεχνολογίες οδήγησαν στην επικράτηση των ελέγχων με αλυσίδες scan υψηλής ταχύτητας, προκειμένου τα αποτελέσματα να είναι πιο ποιοτικά.

Η ενέργεια, που καταναλώνεται λόγω των εναλλαγών που συμβαίνουν κατά τη διάρκεια του ελέγχου ορθότητας του κυκλώματος, συμβάλλει σημαντικά στην όξυνση των προβλημάτων. Όπως έχει προαναφερθεί, η κατανάλωση ενέργειας κατά τη φάση που το κύκλωμα βρίσκεται υπό έλεγχο, μπορεί να υπερβαίνει την κατανάλωση ενέργειας κατά τη φάση που το κύκλωμα βρίσκεται υπό κανονική λειτουργία [8], [19]. Η κατανάλωση ενέργειας απασχολεί όχι μόνο στο στάδιο ολίσθησης των *test vectors* στις αλυσίδες scan, αλλά και στο στάδιο λήψης των αποκρίσεων στις αλυσίδες scan, οι οποίες διακρίνονται για την υψηλή ταχύτητα. Κατά τη διάρκεια του ελέγχου ενός κυκλώματος υπεισέρχονται αρκετές παρενέργειες, όπως η ανεπιθύμητη



απώλεια, η πτώση τάσης (IR-drop), το φαινόμενο crosstalk, οι παρατεταμένες αποτυχίες, εξαιτίας των υψηλών απαιτήσεων σε ενέργεια.

Γενικά, η λειτουργία του ελέγχου ορθότητας του κυκλώματος χωρίζεται σε δυο υπολειτουργίες, την *scan shifting* και τη *scan capture*. Όμως, κατά την φάση της *scan shifting*, όπου πραγματοποιείται η ολίσθηση του *test vector* στην αλυσίδα scan, παρατηρείται υπερβολική κατανάλωση ενέργειας εξαιτίας του μεγάλου αριθμού των ταυτόχρονων εναλλαγών. Έτσι, η ενέργεια μπορεί να διατηρηθεί σε ομαλές τιμές είτε μειώνοντας τη συχνότητα ολίσθησης, είτε τροποποιώντας την αρχιτεκτονική της αλυσίδας scan. Επίσης, τα προβλήματα με την κατανάλωση ενέργειας μπορούν να περιοριστούν αν τροποποιηθούν με κατάλληλη μέθοδο τα *test patterns*. Αντίστοιχα, κατά την φάση της *scan capture*, όπου πραγματοποιείται η λήψη της απόκρισης στην αλυσίδα scan, ιδιαίτερα όταν εφαρμόζονται υψηλές ταχύτητες, ανακύπτουν προβλήματα, όπως η πτώση τάσης (IR-drop), οι παρενέργειες του crosstalk, οι ανεπιθύμητες παρατάσεις του κύκλου ή διαμορφώσεις των κυμάτων που συμβαίνουν στους παλμούς του ρολογιού και επιφέρουν μείωση της συχνότητας του ρολογιού ή την παντελή παύση του. Έτσι, για να επιλυθούν ιδίως τα τελευταία προβλήματα προτείνεται η εισαγωγή της έννοιας του clock gating [1].

Όπως έχει προαναφερθεί, η κατάλληλη τροποποίηση του περιεχομένου των *test patterns*, με άμεσο στόχο τη μείωση των ταυτόχρονων εναλλαγών κατά την φάση της *scan shifting*, είναι δυνατόν να επιφέρει μείωση της κατανάλωσης της ενέργειας. Μάλιστα, σε διάφορες μελέτες έχει προταθεί η τεχνική *pattern scrubbing* [1] σύμφωνα με την οποία αν τα περισσότερα δεδομένα που εισάγονται στην αλυσίδα scan είναι ίδια μεταξύ τους, δηλαδή όλα είναι 0 ή όλα 1, αλλά προτιμότερο να είναι 0, τότε τα δεδομένα που παίρνουμε κατά τη φάση της *scan capture* είναι παρόμοια. Με αυτό τον τρόπο επιτυγχάνεται μείωση της δραστηριότητας των ταυτόχρονων εναλλαγών.

Το μείζον θέμα όμως είναι ποια bit πρέπει να τροποποιηθούν χωρίς να προκύψουν περαιτέρω συνέπειες και δεν είναι βέβαιο ότι η τροποποίηση των *test patterns* είναι η ιδανική λύση σε κάθε περίπτωση. Μια ικανοποιητική λύση στο θέμα των bits είναι η εκμετάλλευση των μη καθορισμένων bits ή αλλιώς των X bits που έχει κάθε *test*



cube. Όταν παράγεται ένα ντετερμινιστικό *test pattern*, ο αλγόριθμος ATPG (Automatic Test Pattern Generation) θα τοποθετήσει στα κύτταρα της αλυσίδας scan και στις εισόδους των κυκλωμάτων τις τιμές 0 και 1, που καλούνται καθορισμένα bits. Στη συνέχεια, με την εφαρμογή τεχνικών συμπίεσης πάνω στα *test patterns*, είναι δυνατόν να ανιχνευθούν πολλαπλά σφάλματα με ένα μόνο *test pattern*, αυξάνοντας έτσι το πλήθος των καθορισμένων bits.

Στον συμβατικό αλγόριθμο ATPG, τα μη καθορισμένα bits συμπληρώνονται τυχαία και το προκύπτον *test vector* προσομοιώνεται για να επιβεβαιωθεί η ανίχνευση των επιθυμητών σφαλμάτων, αλλά και για να υπολογιστεί το ποσοστό των επιπλέον σφαλμάτων που ενδεχομένως να ανιχνεύονται. Είναι επιθυμητό λοιπόν το ποσοστό ανίχνευσης και άλλων σφαλμάτων πέρα από τα αναμενόμενα να είναι μεγάλο. Ωστόσο, ενώ εξελίσσεται η διαδικασία παραγωγής των *test vectors* και το ποσοστό ανίχνευσης των επιθυμητών σφαλμάτων αυξάνεται, το ποσοστό ανίχνευσης των μη αναμενόμενων σφαλμάτων μειώνεται. Επίσης, έχει παρατηρηθεί ότι το ποσοστό των μη καθορισμένων bits σε σχέση με τα συνολικά διαθέσιμα bits είναι μεγάλο. Μάλιστα, η μέση αναλογία των μη καθορισμένων bits στα ντετερμινιστικά *test patterns* είναι γενικά υψηλότερη για τα σφάλματα τύπου *transition* από ότι για τα σφάλματα τύπου *stuck-at*, εξαιτίας της δύσκολης εφαρμογής των τεχνικών συμπίεσης στα patterns για τα σφάλματα τύπου *transition*. Τέλος, έχει παρατηρηθεί ότι τα πρώτα patterns, που εφαρμόζονται, έχουν μεγαλύτερο ποσοστό από καθορισμένα bits συγκριτικά με τα patterns, που έπονται [1]. Συνεπώς, η ύπαρξη αυτού του ποσοστού των μη καθορισμένων bits μπορεί να ληφθεί υπόψη στην προσπάθεια μείωσης της κατανάλωσης της ενέργειας, χρησιμοποιώντας άλλες τεχνικές πέρα από τη τυχαία συμπλήρωση των μη καθορισμένων bits.

Πάντως, η εφαρμογή διαφορετικών τεχνικών συμπλήρωσης των μη καθορισμένων bits οδηγεί στην παραγωγή *test vectors* με διαφορετικές ιδιότητες, σύμφωνα με τις διάφορες μετρικές, όπως την ποσότητα των *test vectors*, την κάλυψη σφαλμάτων και την κατανάλωση ενέργειας. Όσον αφορά την ποσότητα των *test vectors*, οι κακές επιλογές στη συμπλήρωση των δεδομένων μπορεί να περιορίσουν το πλήθος των σφαλμάτων, που δεν αναμενόταν να ανιχνευθούν πέρα από τα αναμενόμενα σφάλματα. Έχοντας λοιπόν μικρό αριθμό τέτοιων ανιχνεύσεων για κάθε *test vector*,

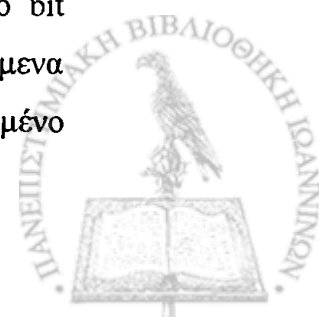


τότε αυξάνεται η ολική ποσότητα των *test vectors*. Επιπλέον, αν η ποσότητα των *test vectors* αυξάνεται τόσο ώστε να επιβάλλονται περικοπές στον ελεγκτή (tester), τότε είναι αναμενόμενο η ανίχνευση των σφαλμάτων να περιοριστεί. Η ανίχνευση των σφαλμάτων περιορίζεται και όταν κάποια σφάλματα, που ενώ στην τυχαία συμπλήρωση των μη καθορισμένων bits μπορούσαν να ανιχνευθούν ακόμα και αν δεν ήταν αναμενόμενα, στην μη τυχαία συμπλήρωση δεν ανιχνεύονται ακόμα και αν είναι αναμενόμενα. Μάλιστα, η εφαρμογή κάποιων τεχνικών μη τυχαίας συμπλήρωσης των μη καθορισμένων bits οδηγεί στη μείωση της δραστηριότητας των εναλλαγών και κατ' επέκταση της ενέργειας, ειδικά όταν ο αριθμός των καθορισμένων bits είναι μικρός. Συμπερασματικά, είναι αναγκαία η εφαρμογή τεχνικής που θα μειώνει μεν την κατανάλωση ενέργειας και την ποσότητα των *test vectors*, αλλά θα αυξάνει το ποσοστό ανίχνευσης σφαλμάτων.

2.2.2. Σύντομη επισκόπηση της τεχνικής *Fill Adjacent*

Η τεχνική *Fill Adjacent* [1] είναι ευρέως γνωστή λόγω του ότι επιτυγχάνει σημαντική μείωση στην κατανάλωση ενέργειας κατά τη διάρκεια της εισαγωγής των *test vectors* με ολίσθηση στις αλυσίδες scan. Όταν δυο συνεχόμενα bits σε ένα *test vector* έχουν διαφορετικές τιμές, τότε κατά την εισαγωγή τους με ολίσθηση στις αλυσίδες scan παράγεται μια δραστηριότητα εναλλαγών. Η τεχνική *Fill Adjacent* εστιάζεται στη μείωση της ενέργειας με το να εκμεταλλεύεται την ύπαρξη των μη καθορισμένων bits σε ένα *test cube*. Συγκεκριμένα, στοχεύει στη μείωση του πλήθους των συνεχόμενων bits με συμπληρωματικές τιμές σε ένα *test vector*, καθώς και στη μείωση της απόστασης που θα διανύσουν τα συνεχόμενα, συμπληρωματικά bits στις αλυσίδες scan.

Πιο αναλυτικά, η τεχνική αυτή χρησιμοποιεί την πληροφορία για τα καθορισμένα bits, προκειμένου να δώσει τιμές στα μη καθορισμένα bits. Αν, για παράδειγμα, τα *test vectors* εισάγονται σε μια αλυσίδα scan από τα αριστερά της, τότε τα bits κάθε *test vector* θα υποβάλλονται σε επεξεργασία ένα-ένα από τα δεξιά προς τα αριστερά από την τεχνική *Fill Adjacent*. Στη συνέχεια, το πιο πρόσφατο καθορισμένο bit χρησιμοποιείται για να δοθεί τιμή στο επόμενο μη καθορισμένο bit ή στα επόμενα κατά σειρά μη καθορισμένα bits και όταν μετά εντοπίζεται καινούργιο καθορισμένο



bit, τότε η τιμή του θα ληφθεί για τη συμπλήρωση των γειτονικών μη καθορισμένων bits. Για παράδειγμα, αν έχουμε το ακόλουθο *test cube*: XX1XX10XXX1XX0XXX0, τότε αυτό με την εφαρμογή της τεχνικής *Fill Adjacent* θα μετατραπεί στο *test vector*: 111111011110000000. Παρατηρείται λοιπόν ότι τα γειτονικά bits τείνουν να έχουν τις ίδιες τιμές. Επιπλέον, όταν είναι αναπόφευκτη η ύπαρξη γειτονικών, συμπληρωματικών bits, τότε επιχειρείται αν είναι δυνατόν η απομάκρυνση τέτοιων ζευγών από την έξοδο της αλυσίδας scan, προκειμένου να μειωθεί η απόσταση που θα διανύσουν αυτά τα ζεύγη κατά την είσοδο τους με ολίσθηση στην αλυσίδα scan. Η συγκεκριμένη τεχνική επιφέρει μείωση στην δραστηριότητα εναλλαγών, διατηρώντας σχετικά χαμηλή την ποσότητα των test vectors. Ωστόσο, εφόσον τα μη καθορισμένα bits δε συμπληρώνονται τυχαία, συνήθως αναμένεται μείωση του ποσοστού ανίχνευσης σφαλμάτων.

Κάποιες άλλες προτεινόμενες τεχνικές μη τυχαίας συμπλήρωσης των μη καθορισμένων bits είναι η τεχνική της συμπλήρωσής τους μόνο με 0 ή μόνο με 1 και η τεχνική της συμπλήρωσής των 1000 πρώτων διανυσμάτων με 0, των 1000 επόμενων με 1, μετά με 0 κοκ. Επίσης, υπάρχει η τεχνική όπου σε κάθε διάνυσμα επιλέγονται τυχαία 4 αλυσίδες, όπου τα X συμπληρώνονται με 1 και οι υπόλοιπες αλυσίδες συμπληρώνονται τυχαία και η τεχνική όπου σε κάθε διάνυσμα επιλέγονται τυχαία 4 αλυσίδες με τα X να συμπληρώνονται σύμφωνα με την τεχνική *Fill Adjacent* και οι υπόλοιπες αλυσίδες να συμπληρώνονται τυχαία.

2.2.3. Σύντομη επισκόπηση της μεθόδου *Preferred Fill*

Η μέθοδος *Preferred Fill* επιτυγχάνει αισθητή μείωση της δραστηριότητας εναλλαγών κατά τη διάρκεια της λήψης της απόκρισης του ελεγχόμενου κυκλώματος για ένα δοθέν test vector στην αλυσίδα scan [16, 17]. Η βασική λειτουργία αυτής της μεθόδου είναι η συμπλήρωση των μη καθορισμένων bits ενός test cube. Έστω ότι έχουμε το ζεύγος διανυσμάτων $\langle V_1, V_2 \rangle$, που έχουν προκύψει σύμφωνα με την τεχνική LOC και επρόκειτο να χρησιμοποιηθούν για την ανίχνευση σφαλμάτων τύπου *transition delay*. Πιο αναλυτικά, τα διανύσματα αυτά μπορούν να οριστούν ως $V_1 = (PI_1, S_1)$ και $V_2 = (PI_2, S_2)$. Συγκεκριμένα, ως PI_1 συμβολίζονται οι τιμές των πρωταρχικών εισόδων (PI) του test vector, ως S_1 συμβολίζονται οι τιμές των



πρωταρχικών ψευδοτυχαίων εισόδων (PPI), ως PI_2 συμβολίζονται οι τιμές των πρωταρχικών εισόδων (PI) του *test vector* και ως S_2 συμβολίζονται οι τιμές των πρωταρχικών ψευδοτυχαίων εξόδων (PPO) που προκύπτουν από την εφαρμογή του V_1 . Απώτερος στόχος λοιπόν, είναι να μειωθεί η απόσταση Hamming ανάμεσα στα διανύσματα V_1 και V_2 . Πιο συγκεκριμένα, να μειωθεί η απόσταση Hamming στα επιμέρους τμήματα αυτών των διανυσμάτων, δηλαδή τόσο ανάμεσα στα PI_1 και PI_2 , όσο και ανάμεσα στα S_1 και S_2 .

Έτσι, σε περίπτωση που οι πρωταρχικές εισοδοί δεν είναι σταθερές καθ' όλη τη διάρκεια των δυο κύκλων της διαδικασίας LOC, είναι δυνατόν οι μη καθορισμένες τιμές στο PI_1 (PI_2) να συμπληρώνονται σύμφωνα με τις καθορισμένες τιμές στο PI_2 (PI_1). Στα προκύπτοντα λοιπόν PI_1 και PI_2 , οι εναπομένουσες μη καθορισμένες τιμές θα βρίσκονται στις ίδιες θέσεις. Για παράδειγμα, αν έχουμε $PI_1 = (1XXX01X)$ και $PI_2 = (01X0XXX)$, τότε οι θέσεις 2 και 4 στο PI_1 θα πάρουν τις τιμές 1 και 0, ενώ οι θέσεις 5 και 6 στο PI_2 θα πάρουν τις τιμές 0 και 1. Συνεπώς, προκύπτουν τα $PI_1 = (11X001X)$ και $PI_2 = (01X001X)$, όπου αν οι μη καθορισμένες τιμές συμπληρωθούν τυχαία με 0 θα έχουμε αντίστοιχα $PI_1 = (1100010)$ και $PI_2 = (0100010)$ με απόσταση Hamming 1. Αντί για τη τυχαία συμπλήρωση των παραπάνω μη καθορισμένων τιμών με 0, μπορεί να χρησιμοποιηθεί η παρακάτω μέθοδος που εφαρμόζεται στα S_1 και S_2 .

Στην περίπτωση των S_1 και S_2 , έστω $S_1 = (s_{11}, s_{12}, s_{13}, \dots, s_{1n})$ και $S_2 = (s_{21}, s_{22}, s_{23}, \dots, s_{2n})$. Αν το s_{1j} δεν είναι καθορισμένο, τότε θα πρέπει να συμπληρωθεί με 1 (0), αν το s_{2j} έχει τη τιμή 1 (0) με πιθανότητα μεγαλύτερη από την πιθανότητα να έχει τη τιμή 0 (1). Δηλαδή, το s_{1j} θα συμπληρωθεί με τη τιμή που είναι πιθανότερο να διατηρηθεί στο j -οστό κύτταρο της αλυσίδας scan. Ως $HP_j(v)$, $v \in \{0, 1\}$, ορίζεται η κατά συνθήκη πιθανότητα το j -οστό κύτταρο σε μία αλυσίδα scan να διατηρεί την τιμή v , ενώ όλες οι άλλες εισοδοί παίρνουν τυχαίες τιμές.

Ορισμός 2.1: Η προτιμητέα τιμή για το j -οστό κύτταρο μιας αλυσίδας scan είναι το 0 (1) αν και μόνο αν $HP_j(0) > HP_j(1)$ ($HP_j(1) > HP_j(0)$).

Για τη συμπλήρωση των μη καθορισμένων τιμών του S_1 , πρώτα σε κάθε θέση με μη καθορισμένη τιμή βάζουμε την καθορισμένη τιμή που ενδεχομένως να υπάρχει στην αντίστοιχη θέση στο S_2 . Εντέλει, στις ίδιες θέσεις που το S_1 θα έχει μη καθορισμένες



τιμές, θα έχει και το S_2 . Για παράδειγμα, αν έχουμε $S_1 = (1XXX01X)$ και $S_2 = (01X0XXX)$, τότε οι θέσεις 2 και 4 στο S_1 θα πάρουν τις τιμές 1 και 0. Συνεπώς, προκύπτουν τα $S_1 = (11X001X)$ και $S_2 = (01X0XXX)$, όπου οι μη καθορισμένες τιμές του S_1 θα συμπληρωθούν με την προτιμητέα τιμή αν υπάρχει ή τυχαία.

Η προτιμητέα τιμή, όπως έχει ήδη αναφερθεί, γνωστοποιείται μέσω των κατά συνθήκη πιθανοτήτων. Αυτές οι πιθανότητες μπορούν να υπολογιστούν με συμβολικές μεθόδους, οι οποίες εντάσσονται στην κατηγορία NP-hard, ή με προσομοιώσεις πολλών τυχαίων διανυσμάτων εισόδου, το οποίο είναι υπολογιστικά πολύ απαιτητικό. Ένας πιο απλοποιημένος τρόπος είναι η χρήση των πιθανοτήτων των τιμών όλων των κυττάρων μιας αλυσίδας scan ($\forall s_{2j}, 1 \leq j \leq n$), αγνοώντας ενδεχόμενες συσχετίσεις ανάμεσα στις εισόδους των πυλών. Μια επιπλέον εικασία που λαμβάνεται υπόψη είναι ότι η πιθανότητα το s_{2j} , δηλαδή το j -οστό μέλος του S_2 , να πάρει τη τιμή 1 ή 0 είναι ανεξάρτητη από την κατάσταση του s_{1j} . Συνεπώς, μπορούμε να έχουμε $HP_j(0) > HP_j(1)$ ($HP_j(1) > HP_j(0)$) αν και μόνον αν $P_j(0) > P_j(1)$ ($P_j(1) > P_j(0)$), όπου $P_j(1)$ ($P_j(0)$) ορίζεται η πιθανότητα το s_{2j} να πάρει την τιμή 1 (0).

2.3. Η έννοια των Output Deviations

2.3.1. Γενικά

Για την ανίχνευση ελαττωμάτων χρησιμοποιούνται ποικίλες τεχνικές, όπως ο λειτουργικός έλεγχος με ταχύτητα, ο έλεγχος καθυστέρησης, τα σύνολα ελέγχου *n*-detection, οι οποίες όμως καταλήγουν σε τεράστιες ποσότητες δεδομένων ελέγχου (*test data*). Μάλιστα, έχει προβλεφθεί ότι ο χρόνος ελέγχου των κυκλωμάτων και το μέγεθος των δεδομένων ελέγχου θα είναι 30 φορές μεγαλύτερος το 2010 συγκριτικά με τη τρέχουσα κατάσταση. Εκτός των άλλων, ανακύπτουν καινούργιοι τύποι ελαττωμάτων εξαιτίας της τάσης της συρρίκνωσης της τεχνολογίας, οι οποίοι δεν είναι δυνατόν να ενταχθούν στα υπάρχοντα μοντέλα σφαλμάτων. Έτσι, είναι αναγκαίο να βρεθεί μια τεχνική, η οποία να αξιολογεί και να επιλέγει τα πιο αποτελεσματικά *test patterns* μέσα από ένα τεράστιο σύνολο διανυσμάτων ελέγχου. Έως τώρα, η επιλογή των *test patterns* βασιζόταν στην κατηγοριοποίηση των



σφαλμάτων, η οποία όχι μόνο ήταν υπολογιστικά ακριβή, αλλά έπρεπε να επαναλαμβάνεται για κάθε μοντέλο σφαλμάτων. Συνεπώς, αντί να χρησιμοποιούνται όλα τα *test patterns* για κάθε μοντέλο σφαλμάτων ξεχωριστά, είναι προτιμότερη η εύρεση της ιδανικής ένωσης όλων των συνόλων διανυσμάτων ελέγχου για όλα τα μοντέλα σφαλμάτων χωρίς να σημειώνεται αισθητή αύξηση του μεγέθους αυτής της ένωσης.

Στη συνέχεια, θα αναλυθεί ένα πιθανολογικό μοντέλο σφαλμάτων για την διεξαγωγή ενός χάρτη πιθανοτήτων πάνω στο κύκλωμα, ο οποίος μπορεί να χρησιμοποιηθεί μετέπειτα για την ταξινόμηση των *test patterns*. Σε αυτόν τον χάρτη βασίζεται και ο υπολογισμός του *output deviation*, το οποίο αποτελεί μετρική μονάδα για την ανακατανομή των *test patterns* μέσα στο σύνολο[21].

Οι μέθοδοι ανακατανομής των *test patterns*, οι οποίες ταξινομούν τα *patterns* και τοποθετούν τα πιο αποτελεσματικά στην αρχή του συνόλου, επιφέρουν μειώσεις τόσο στο χρόνο ελέγχου, όσο και στον όγκο των δεδομένων ελέγχου. Η εφαρμογή των πιο αποτελεσματικών *patterns* στην αρχή, αυξάνει την πιθανότητα να ανιχνευθούν τα κατασκευαστικά ελαττώματα νωρίτερα. Συνεπώς, αν τα αποτελεσματικά *patterns* τοποθετούνται στην αρχή του συνόλου των δεδομένων, τότε τα ελαττωματικά κυκλώματα θα εντοπιστούν νωρίτερα, μειώνοντας έτσι το χρόνο ελέγχου, ο οποίος παίζει σπουδαίο ρόλο σε ορισμένα περιβάλλοντα. Είναι εφικτό τα ανακαταταξινομημένα σύνολα δεδομένων να περικόπτονται προκειμένου να προσαρμόζονται στους χρονικούς και χωρικούς περιορισμούς.

2.3.2. Συνοπτική αναφορά στα *Output deviations*

Το *output deviation* χρησιμοποιείται ως μέτρο στην επιλογή των βέλτιστων *patterns*, για την γρηγορότερη και αποτελεσματικότερη ανίχνευση μη μοντελοποιημένων σφαλμάτων, και στη μέθοδο ανακατανομής των *patterns* [21]. Η μέθοδος που επακολουθεί για την επιλογή των *patterns*, βασιζόμενη στο *output deviation*, μπορεί να συνδυαστεί με τη ροή ATPG για να επιτευχθεί καλύτερη ποιότητα και μειωμένη ποσότητα των *patterns*. Για παράδειγμα, για κάθε μοντέλο σφαλμάτων μπορεί να παραχθεί ένα μεγαλύτερο σύνολο n' -detect ($n' > n$) από το οποίο θα επιλεγθούν τα n



patterns με το μεγαλύτερο deviation, παράγοντας εντέλει το σύνολο n-detect. Επιπλέον, όταν το εργαλείο ATPG παράγει ένα test cube για ένα σφάλμα, τότε τα μη καθοριζόμενα bits (X) συμπληρώνονται τυχαία για να προκύψει ένα test vector. Εκμεταλλευόμενοι αυτό το γεγονός, μπορούμε να έχουμε πολλά test vectors για ένα test cube και έτσι να επιλεγθούν αυτά με τα μεγαλύτερα output deviations.

Για να γίνει κατανοητή η έννοια του *output deviation*, που χρησιμοποιείται στην αξιολόγηση των patterns, προηγείται η επεξήγηση κάποιων ορισμών [21]. Αρχικά, θα αναφερθούμε στο *CL (Confidence Level)* μιας λογικής πύλης μιας εξόδου. Πιο αναλυτικά, το CL συμπεριλαμβάνει όλους τους δυνατούς, διαφορετικούς συνδυασμούς των εισόδων και για ένα δοθέν συνδυασμό παρέχει την πιθανότητα η έξοδος να είναι σωστή σύμφωνα με ένα δοθέν μοντέλο σφαλμάτων. Ωστόσο, η πιθανότητα η έξοδος να είναι σωστή μπορεί να διαφέρει για τους διάφορους συνδυασμούς των εισόδων.

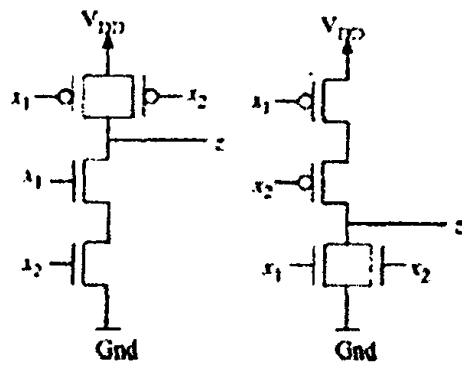
Ορισμός 2.2: Το CL (Confidence Level) R_i μιας πύλης G_i με m εισόδους και μια έξοδο είναι ένα διάνυσμα με 2^m στοιχεία, ορισμένο ως εξής: $R_i = (r_i^{(00, \dots, 00)}, r_i^{(00, \dots, 01)}, r_i^{(00, \dots, 10)}, \dots, r_i^{(11, \dots, 11)})$, όπου κάθε στοιχείο του R_i δηλώνει την πιθανότητα η έξοδος της πύλης να είναι σωστή για τον αντίστοιχο συνδυασμό εισόδων.

Για παράδειγμα, ως $r_i^{(00)}$ ορίζεται η πιθανότητα η έξοδος της πύλης G_i να είναι σωστή όταν οι 2 εισοδοί είναι 00. Αν $m=2$ για μια λογική πύλη, τότε θα έχουμε $R_i = (r_i^{(00)}, r_i^{(01)}, r_i^{(10)}, r_i^{(11)})$.

Ένας πιο γενικός ορισμός του *CL (Confidence Level)* μπορεί να τεθεί για ένα υποκύκλωμα ή για μια υπερπύλη με m εισόδους και $k>1$ εξόδους. Σε αυτή την περίπτωση, το CL είναι ένα σύνολο από k διανύσματα, όπου κάθε στοιχείο αυτού του συνόλου είναι το CL της αντίστοιχης εξόδου.

Το διάνυσμα *CL (Confidence Level)* που σχετίζεται με μια λογική πύλη παράγεται σύμφωνα με τις πιθανότητες αποτυχίας στο επίπεδο των τρανζίστορ. Πιο συγκεκριμένα, ας λάβουμε υπόψη τις πύλες NAND και NOR 2 εισόδων στο Σχήμα 2.3.





Σχήμα 2.3 Πύλες NAND και NOR

Ας υποθέσουμε ότι το τρανζίστορ δεν άγει ποτέ με πιθανότητα a εξαιτίας κάποιου ελαττώματος (σφάλμα stuck-open) και άγει μόνιμα με πιθανότητα β εξαιτίας κάποιου ελαττώματος (σφάλμα stuck-on). Έστω λοιπόν ότι έχουμε τον συνδυασμό εισόδων $x_1x_2 = 00$ και παρουσιάζονται μόνο stuck-open σφάλματα. Τότε, η πύλη NAND θα δώσει τη σωστή έξοδο με πιθανότητα $1-a^2$ γιατί η εσφαλμένη έξοδος θα προκύψει μόνο όταν και τα 2 p-transistors είναι stuck-open. Ομοίως, αν ληφθούν υπόψη μόνο τα σφάλματα stuck-on, τότε η πύλη NAND θα δώσει τη σωστή έξοδο για είσοδο 00 με πιθανότητα $1-\beta^2$. Στο Σχήμα 2.4 παρουσιάζονται τα διανύσματα CL των πυλών NAND και NOR αντίστοιχα για τα σφάλματα stuck-open και stuck-on για όλους τους δυνατούς συνδυασμούς των εισόδων.

x_1x_2	$R_1^{NAND}(\bar{x}_1, \bar{x}_2, a)$	$R_2^{NAND}(\bar{x}_1, \bar{x}_2, b)$
00	$1-a^2$	$1-b^2$
01	$1-a$	$1-b$
10	$1-a$	$1-b$
11	$(1-a)^2$	$(1-b)^2$

x_1x_2	$R_1^{NOR}(\bar{x}_1, \bar{x}_2, a)$	$R_2^{NOR}(\bar{x}_1, \bar{x}_2, b)$
00	$(1-a)^2$	$(1-b)^2$
01	$1-a$	$1-b$
10	$1-a$	$1-b$
11	$1-a^2$	$1-b^2$

Σχήμα 2.4 Διανύσματα CL των πυλών NAND και NOR



Ως $R_i^{\text{NAND}}(0,0,\alpha)$ ορίζεται η πιθανότητα η είσοδος 00 να παράγει την σωστή έξοδο όταν η πιθανότητα σφάλματος stuck-open σε ένα τρανζίστορ είναι α . Επίσης, και τα τυποποιημένα σφάλματα, όπως τα stuck-at, μπορούν να εκφραστούν σύμφωνα με τα CL διανύσματα. Στο Σχήμα 2.5 παρουσιάζεται η περίπτωση μιας πύλης NAND 2 εισόδων x_1, x_2 και μιας εξόδου z για σφάλματα stuck-at.

Stuck-at fault in two-input NAND gate	Corresponding CL vector for the NAND gate
Input x_1 s-a-0	(1 1 1 0)
Input x_1 s-a-1	(1 0 1 1)
Input x_2 s-a-0	(1 1 1 0)
Input x_2 s-a-1	(1 1 0 1)
Output z s-a-0	(0 0 0 1)
Output z s-a-1	(1 1 1 0)

Σχήμα 2.5 Η σχέση των σφαλμάτων Stuck-at και των διανυσμάτων CL

Γενικά, όταν εμφανίζονται ταυτόχρονα τόσο τα σφάλματα stuck-open, όσο και τα stuck-on, τότε θα ισχύει ο ακόλουθος τύπος $R(x_1, x_2, \alpha, \beta) = R(x_1, x_2, \alpha) \cdot R(x_1, x_2, \beta)$. Δηλαδή, για οποιοδήποτε συνδυασμό εισόδων (x_1, x_2) το σύνολο των τρανζίστορ που επηρεάζουν την έξοδο και εμφανίζουν σφάλματα stuck-open είναι ανεξάρτητο από το σύνολο των τρανζίστορ με σφάλματα stuck-on. Τα διανύσματα *CL* (*Confidence Level*), που επεξηγήθηκαν, θα χρησιμοποιηθούν στη συνέχεια για τον υπολογισμό του *output deviation*.

Κατόπιν, ως $p_{i,0}$ και $p_{i,1}$ ορίζονται οι πιθανότητες η γραμμή i να έχει τη λογική τιμή 0 ή 1, αντίστοιχα και για τη γραμμή i θα ισχύει $p_{i,0} + p_{i,1} = 1$. Έχοντας λοιπόν μια λογική πύλη G , η οποία είναι NAND με δυο εισόδους j, k και μια έξοδο i , θα έχουμε τις ακόλουθες πιθανότητες:

$$p_{i,0} = p_{j,1}p_{k,1}r_i^{(11)} + p_{j,0}p_{k,0}(1 - r_i^{(00)}) + p_{j,0}p_{k,1}(1 - r_i^{(01)}) + p_{j,1}p_{k,0}(1 - r_i^{(10)})$$

$$p_{i,1} = p_{j,0}p_{k,0}r_i^{(00)} + p_{j,0}p_{k,1}r_i^{(01)} + p_{j,1}p_{k,0}r_i^{(10)} + p_{j,1}p_{k,1}(1 - r_i^{(11)})$$

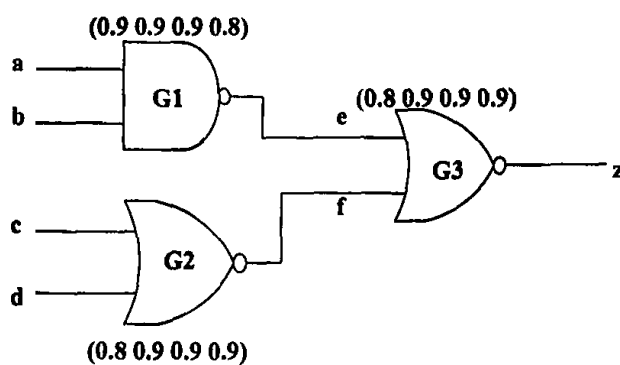
Ας αναφερθεί επίσης και η περίπτωση της λογικής πύλης XOR, για την οποία ισχύουν οι εξής πιθανότητες σημάτων για την έξοδο:



$$p_{i,0} = p_{j,0}p_{k,0}r_i^{(00)} + p_{j,0}p_{k,1}(1 - r_i^{(01)}) + p_{j,1}p_{k,0}(1 - r_i^{(10)}) + p_{j,1}p_{k,1}r_i^{(11)}$$

$$p_{i,1} = p_{j,0}p_{k,0}(1 - r_i^{(00)}) + p_{j,0}p_{k,1}r_i^{(01)} + p_{j,1}p_{k,0}r_i^{(10)} + p_{j,1}p_{k,1}(1 - r_i^{(11)})$$

Στο Σχήμα 2.6 παρουσιάζεται ένα μικρό κύκλωμα με τις πύλες G1, G2 και G3. Το CL κάθε πύλης είναι αντίστοιχα (0.9 0.9 0.9 0.8), (0.8 0.9 0.9 0.9) και (0.8 0.9 0.9 0.9). Για τρεις διαφορετικούς συνδυασμούς εισόδων 0000, 0101 και 1111 παρουσιάζονται στον Πίνακα 2.1 οι προκύπτουσες πιθανότητες σημάτων ενώ στην 1^η στήλη εμφανίζεται και η σωστή τιμή της εξόδου z.



2.6 Παράδειγμα κυκλώματος

Πίνακας 2.1 Πιθανότητες σημάτων

Input pattern, z	$p_{e,0}$	$p_{e,1}$	$p_{f,0}$	$p_{f,1}$	$p_{z,0}$	$p_{z,1}$
0000, 0	0.1	0.9	0.2	0.8	0.886	0.114
0101, 0	0.1	0.9	0.9	0.1	0.837	0.163
1111, 1	0.8	0.2	0.9	0.1	0.396	0.604

Για μια λογική πύλη g ενός κυκλώματος, έστω ότι η σωστή τιμή εξόδου της για ένα οποιοδήποτε πρότυπο εισόδου t_j είναι d, $d \in \{0, 1\}$. Ως *output deviation* $\Delta_{g,j}$ της πύλης g για το πρότυπο εισόδου t_j ορίζεται η πιθανότητα $p_{g,d'}$, όπου d' είναι το συμπλήρωμα του d. Με άλλα λόγια, η απόκλιση (deviation) για ένα πρότυπο εισόδου είναι ουσιαστικά η πιθανότητα η έξοδος να είναι εσφαλμένη. Έτσι, όσο μεγαλύτερη είναι η απόκλιση για ένα πρότυπο εισόδου, τόσο αυξάνεται η κάλυψη σφαλμάτων με



την εφαρμογή αυτού του προτύπου. Για παράδειγμα, το *outrput deviation* για το πρότυπο εισόδου 1111 στον Πίνακα 2.1 είναι η πιθανότητα $p_{z,0} = 0.396$.

Ορισμός 2.3: Ένα συνδυαστικό κύκλωμα C ορίζεται ως $C = \{G, PI, Z, R\}$, όπου $G = \{G_1, G_2, \dots, G_N\}$ είναι το σύνολο των λογικών πυλών του C , PI και Z είναι τα σύνολα των βασικών εισόδων και εξόδων αντίστοιχα και $R = \{R_1, R_2, \dots, R_N\}$ είναι το σύνολο των διανυσμάτων CL (Confidence Level) των πυλών του G .

Ορισμός 2.4: Το πιθανολογικό μοντέλο σφαλμάτων F του κυκλώματος C ορίζεται ως εξής: 1) Κάθε πύλη G_i μπορεί να αποτύχει ανεξάρτητα από τις υπόλοιπες πύλες και 2) η εσφαλμένη συμπεριφορά του κυκλώματος C ορίζεται από το διάνυσμα R_i .

Σύμφωνα με το μοντέλο F το κύκλωμα του Σχήματος 2.6 μπορεί να αποτύχει για διάφορους λόγους, που ορίζονται ως γεγονότα σφάλματος. Στον Πίνακα 2.2 έχουν ομαδοποιηθεί όλα τα πιθανά γεγονότα σφάλματος, που μπορούν να εμφανιστούν στο κύκλωμα, μαζί με τις αντίστοιχες πιθανότητές τους, δοθέντος ότι το πρότυπο εισόδου είναι $abcd = 0000$. Όπως μπορεί να παρατηρηθεί, μόνο τα γεγονότα 1, 4, 5 και 6 είναι ανιχνεύσιμα μιας και η έξοδος που δίνεται είναι 1 και όχι 0. Συνεπώς, αν E είναι ένα γεγονός εξαιτίας του οποίου το πρότυπο εισόδου 0000 ανιχνεύει σφάλμα στο κύκλωμα, τότε η αντίστοιχη πιθανότητα υπολογίζεται από τον τύπο:

$$P[E] = P[E1 \cup E4 \cup E5 \cup E6] = P[E1] + P[E4] + P[E5] + P[E6] = 0.114$$

Είναι αντιληπτό ότι η πιθανότητα το πρότυπο εισόδου t_j να εμφανίσει ένα λάθος στην έξοδο z_i , σύμφωνα με το πιθανολογικό μοντέλο σφαλμάτων F , είναι ανάλογη του *outrput deviation* $\Delta_{z,j}$. Συμπερασματικά λοιπόν, μπορεί να τεθεί ως στόχος η εφαρμογή εκείνων των εισόδων σε ένα κύκλωμα, που θα δίνουν μεγάλες τιμές στο *outrput deviation* στις εξόδους προκειμένου να αυξηθεί η πιθανότητα ανίχνευσης σφαλμάτων στο κύκλωμα.

Πίνακας 2.2 Τα γεγονότα σφάλματος για είσοδο $abcd=0000$

Fault Event	Fault event description	Event probability	Output value
E0	G_1, G_2, G_3 fault-free	$.9 \times .8 \times .9 = .648$	0
E1	G_1, G_2 fault-free, G_3 faulty	$.9 \times .8 \times .1 = .072$	1
E2	G_1, G_3 fault-free, G_2 faulty	$.9 \times .2 \times .9 = .162$	0
E3	G_2, G_3 fault-free, G_1 faulty	$.1 \times .8 \times .9 = .072$	0
E4	G_1, G_2 faulty, G_3 fault-free	$.1 \times .2 \times .8 = .016$	1
E5	G_1, G_3 faulty, G_2 fault-free	$.1 \times .8 \times .1 = .008$	1
E6	G_2, G_3 faulty, G_1 fault-free	$.9 \times .2 \times .1 = .018$	1
E7	G_1, G_2, G_3 faulty	$.1 \times .2 \times .2 = .004$	0



Η έννοια του *output deviation* χρησιμοποιείται γενικά ως κριτήριο για την ταξινόμηση των *test patterns* στο σύνολο δεδομένων. Μάλιστα, σύμφωνα με κάποια πειράματα, ένα σύνολο από *test patterns* χωρίστηκε σε δυο υποσύνολα T_{high} και T_{low} σύμφωνα με τα *output deviations*. Το σύνολο T_{high} περιλάμβανε εκείνα τα *test patterns* που έδιναν μια τουλάχιστον έξοδο με απόκλιση (deviation) μεγαλύτερη από κάποιο προκαθορισμένο κατώφλι, ενώ τα υπόλοιπα *test patterns* εντάσσονταν στο άλλο σύνολο. Το σύνολο T_{high} επέφερε μεγαλύτερη κάλυψη σφαλμάτων για σφάλματα τύπου stuck-at και bridging. Επομένως, εφαρμόζοντας πρώτα τα πιο αποτελεσματικά *test patterns* σε ένα ταξινομημένο σύνολο δεδομένων, εντοπίζονται γρηγορότερα τα ελαττωματικά κυκλώματα μειώνοντας έτσι τον χρόνο ελέγχου.

2.4. Η μετρική weighted transition

Όταν εισάγουμε κάποια *test cubes* στις αλυσίδες scan, έχουμε κατανάλωση ενέργειας η οποία εξαρτάται από τις εναλλαγές που συντελούνται όχι μόνο στα διάφορα μέρη του ελεγχόμενου κυκλώματος, αλλά και στις αλυσίδες scan.

Μια αρκετά γνωστή μέθοδος που χρησιμοποιείται για τον υπολογισμό της δραστηριότητας των εναλλαγών (switching activity) που συντελείται κατά την εισαγωγή με ολίσθηση ενός *test vector* σε μία αλυσίδα scan είναι η *weighted transition* [5][15][18]. Σύμφωνα με αυτή τη μετρική, υπολογίζεται το πλήθος των εναλλαγών που προκαλούνται στα διαδοχικά κύτταρα της αλυσίδας scan, συνεκτιμώντας ταυτόχρονα και τη σχετική θέση της κάθε εναλλαγής. Πιο αναλυτικά, έστω m το μήκος της αλυσίδας scan και $T = b_m \dots b_2 b_1$ το *test vector* με το bit b_k να εισάγεται στην αλυσίδα scan πριν από το b_{k+1} . Η μετρική weighted transition κανονικοποιείται με το να διαιρεθεί με το μέγιστο πλήθος εναλλαγών που μπορούν να συμβούν σε μια αλυσίδα scan μήκους m και ο τελικός τύπος είναι ο ακόλουθος:

$$P_{SI}(T) = 2[m(m-1)]^{-1} \sum_{i=1}^{m-1} (m-i)(b_i \oplus b_{i+1}) \quad \text{Εξ. 2.1}$$



Ο μέσος όρος κατανάλωσης ενέργειας scan κατά τη διάρκεια ελέγχου του κυκλώματος μπορεί να υπολογιστεί αν προστεθούν τα αποτελέσματα του παραπάνω τύπου για όλες τις αλυσίδες scan και για όλα τα test vectors.

Αν το test vector T χωρίζεται σε c τμήματα S_1, \dots, S_c όπου το τμήμα $S_j = x_{j,r} \dots x_{j,2} x_{j,1}$ φορτώνεται στην αλυσίδα scan j (το $x_{j,m}$ φορτώνεται πριν από το $x_{j,m+1}$), τότε η αντίστοιχη κανονικοποιημένη μορφή της μετρικής weighted transition για c αλυσίδες scan μήκους r δίνεται από τον τύπο:

$$P_{SI}(T) = 2[cm(m-1)]^{-1} \sum_{j=1}^c \sum_{i=1}^{r-1} (r-i)(x_{j,i} \oplus x_{j,i+1})$$

Εξ. 2.2



ΚΕΦΑΛΑΙΟ 3. Η ΠΡΟΤΕΙΝΟΜΕΝΗ ΜΕΘΟΔΟΣ

3.1 Περιγραφή της προτεινόμενης μεθόδου

3.2 Οι τεχνικές Fill Adjacent και Random Fill

3.3 Εκτίμηση του μέσου όρου των ενεργειών scan-in και scan-out

Γενικά, η προτεινόμενη μέθοδος παράγει πολλαπλά, υποψήφια *test vectors* για κάθε *test cube* συμπληρώνοντας τις μη καθορισμένες τιμές με διάφορους τρόπους, ούτως ώστε να σημειωθεί μια σχετικά μειωμένη κατανάλωση ενέργειας. Κατόπιν, τα υποψήφια *test vectors* κάθε *test cube* αξιολογούνται στην ανίχνευση σφαλμάτων, σύμφωνα με την μετρική *output deviation*, και επιλέγεται το πιο αποτελεσματικό.

3.1. Περιγραφή της προτεινόμενης μεθόδου

3.1.1. Θεωρητικό Υπόβαθρο

Η βασική ιδέα της προτεινόμενης μεθόδου έχει ως υπόβαθρο την τεχνική *Fill Adjacent*, η οποία θα καλείται για χάρη συντομίας ως FA. Λόγω του ότι η προτεινόμενη μέθοδος είναι στην ουσία μια τροποποιημένη έκδοση της FA θα καλείται ως MFA.

Γενικά, οι μη καθορισμένες τιμές σε ένα *test cube*, δηλαδή τα X bits, συμπληρώνονται σύμφωνα με τις υπάρχουσες καθορισμένες τιμές. Για χάρη απλότητας, είναι αναγκαίο να αναφέρουμε ότι επειδή έχουμε θέσει τη διαδικασία εισόδου των bits σε μία αλυσίδα scan να εφαρμόζεται από τα αριστερά της και τη διαδικασία εξόδου από τα δεξιά της, όλες οι μέθοδοι, για τις οποίες γίνεται λόγος,



εφαρμόζονται πάνω στο *test cube* από τα δεξιά προς τα αριστερά. Σύμφωνα με την MFA, αρχίζουμε να επεξεργαζόμαστε κάθε *test cube* από δεξιά προς τα αριστερά. Η τιμή του πιο πρόσφατου καθορισμένου bit χρησιμοποιείται για τη συμπλήρωση ορισμένων από τα επόμενα μη καθορισμένα bit, ενώ τα υπόλοιπα κατά σειρά μη καθορισμένα bit θα συμπληρωθούν σύμφωνα με τη τιμή του επόμενου καθορισμένου bit που θα αναγνωσθεί. Το πλήθος των συνεχόμενων μη καθορισμένων bits, που θα συμπληρωθούν σύμφωνα με το πιο πρόσφατο bit, ορίζεται τυχαία κάθε φορά. Αν, για παράδειγμα, έχουμε την ακολουθία 1XXX0, τότε σύμφωνα με την προτεινόμενη μέθοδο, μπορεί να μετασχηματιστεί σε μία από τις ακόλουθες περιπτώσεις: 10000, 11000, 11100, 11110. Πιο αναλυτικά, σε κάθε *test cube* απομονώνονται οι ομάδες από συνεχόμενες μη καθορισμένες τιμές. Στη συνέχεια, ελέγχονται αν τα καθορισμένα bits που βρίσκονται στα αριστερά και δεξιά τους διαφέρουν. Αν διαφέρουν, τότε ένας τυχαίος αριθμός από τα δεξιότερα X bits κάθε ομάδας συμπληρώνονται σύμφωνα με το δεξί καθορισμένο bit, ενώ τα υπόλοιπα σύμφωνα με το αριστερό. Αν πάλι είναι ίδια, τότε απλώς όλα τα X bits παίρνουν την τιμή αυτή. Με άλλα λόγια, σε μια ομάδα από μη καθορισμένα bits το σημείο εναλλαγής τίθεται τυχαία στην MFA, ενώ στην FA το σημείο εναλλαγής τίθεται στην πιο αριστερή θέση έτσι ώστε να διανύσει τη μικρότερη δυνατή απόσταση κατά την είσοδο στην αλυσίδα scan. Στον πίνακα 3.1 παρουσιάζονται όλες οι περιπτώσεις ομάδων από μη καθορισμένα bits, που μπορεί να έχει ένα *test cube*, μαζί με τις αντίστοιχες προτάσεις συμπλήρωσής τους από κάθε μέθοδο.

Πίνακας 3.1 Συμπλήρωση των X bits

	Type of Block	FA	MFA
i	0x...x0, 0x...x, x...x0	00...00	00...00
ii	1x...x1, 1x...x, x...x1	11...11	11...11
iii	0xx...x1	011...11	011...11, 001...11, ..., 000...01
iv	1xx...x0	100...00	100...00, 110...00, ..., 111...10

Σε περίπτωση που τα πρώτα δεξιότερα bits ενός *test cube* δεν είναι καθορισμένα, τότε συμπληρώνονται σύμφωνα με το πρώτο καθορισμένο bit που θα διαβαστεί από δεξιά προς τα αριστερά. Ομοίως, αν τα τελευταία αριστερά bits δεν είναι καθορισμένα, τότε συμπληρώνονται σύμφωνα με το πιο πρόσφατο καθορισμένο bit. Παρατηρείται λοιπόν ότι τόσο η FA, όσο και η MFA έχουν το ίδιο ελάχιστο πλήθος από εναλλαγές



σε ένα test cube. Μάλιστα, για την MFA τόσο στην περίπτωση iii, όσο και στην iv του Πίνακα 3.1 μπορεί κανείς να διαπιστώσει ότι από την πρώτη έως την τελευταία πρόταση συμπλήρωσης, η δραστηριότητα εναλλαγών κατά την είσοδο στην αλυσίδα scan αυξάνεται μιας και από πρόταση σε πρόταση το σημείο εναλλαγής θα διανύσει περισσότερη απόσταση.

Συνεπώς, σε κάθε ομάδα από n μη καθορισμένα bits του τύπου iii ή iv του Πίνακα 3.1 αντιστοιχούν $n+1$ προτάσεις συμπλήρωσης σύμφωνα με την MFA. Επομένως, αν ένα test cube απαρτίζεται από m ομάδες με n_1, n_2, \dots, n_m μη καθορισμένα bits αντίστοιχα, τότε θα μπορούν να παραχθούν σύμφωνα με την MFA $(n_1+1) \cdot (n_2+1) \cdot \dots \cdot (n_m+1)$ διαφορετικά test vectors. Ένα ενδεικτικό παράδειγμα είναι να έχουμε το test cube 1XX0X1, από το οποίο παράγονται τα ακόλουθα test vectors: 100011, 110011, 111011, 100001, 110001, 111001.

Στον Πίνακα 3.2 παρουσιάζεται ένα test cube, το οποίο συμπληρώνεται με καθεμία από τις μεθόδους: α) τυχαίας συμπλήρωσης (Random Fill), β) FA και γ) MFA και υπολογίζονται αντίστοιχα τα ποσοστά κατανάλωσης ενέργειας scan-in σύμφωνα με την μετρική weighted transition.

Πίνακας 3.2 Test cube $T=XXX1XXX0XXX0XXXXX1$

	MFA	MFA+20
Random Fill	Moderate SA	Moderate SA
010110100110101001	111111000000001111	101111000000001111
$P_{ST}(T): 75.16\%$	$P_{ST}(T): 13.1\%$	$P_{ST}(T): 15\%$
FA	Worst SA	Worst SA
111100000000111111	111111100000000001	010111100000000001
$P_{ST}(T): 10.5\%$	$P_{ST}(T): 15.7\%$	$P_{ST}(T): 19.6\%$

Στην 1^η στήλη, παρουσιάζονται τα προκύπτοντα test vectors με βάση την Random Fill και την FA. Είναι προφανές ότι η μέθοδος FA επιφέρει λιγότερη δραστηριότητα εναλλαγών συγκριτικά με τη Random Fill. Στη 2^η στήλη παρουσιάζονται δυο διαφορετικές περιπτώσεις συμπλήρωσης του test cube σύμφωνα με την μέθοδο MFA.



Το test vector στην 1^η γραμμή παρουσιάζει μια μέτρια δραστηριότητα εναλλαγών, ενώ αυτό στη 2^η γραμμή τη χειρότερη. Είναι εμφανές ότι ακόμα και η χειρότερη περίπτωση παρουσιάζει δραστηριότητα εναλλαγών λίγο υψηλότερη από αυτή της FA και αρκετά λιγότερη από αυτή της Random Fill.

Διαπιστώνεται γενικά ότι παρόλο που η μέθοδος MFA οδηγεί σε λίγο περισσότερη κατανάλωση ενέργειας scan σε σχέση με την FA, τα παραγόμενα *test vectors* παρουσιάζουν αξιόλογες διακυμάνσεις όσον αφορά την ανίχνευση μη μοντελοποιημένων ελαττωμάτων. Οι διακυμάνσεις αυτές οφείλονται στο πλήθος των διαφορετικών *test vectors* που μπορούν να προκύψουν σύμφωνα με τη μέθοδο MFA. Εκμεταλλευόμενοι λοιπόν τις ιδιότητες της μεθόδου MFA, θα παραθέσουμε και μια τροποποιημένη έκδοσή της. Σε αυτή την τροποποιημένη έκδοση αυτό που συντελείται είναι να εφαρμόζεται η τυχαία συμπλήρωση των μη καθορισμένων bits σε ένα ποσοστό του test cube, ενώ το υπόλοιπο test cube να συμπληρώνεται σύμφωνα με την μέθοδο MFA. Το ποσοστό αυτό δίνεται με μια παράμετρο P από τον χρήστη και αφορά το αριστερό μέρος του *test cube*. Δηλαδή, τα P% αριστερότερα bits του test cube αν είναι μη καθορισμένα συμπληρώνονται τυχαία και τα εναπομείναντα σύμφωνα με την MFA. Επιλέγονται τα αριστερότερα bits, διότι αυτά διανύουν τη μικρότερη απόσταση κατά την είσοδο με ολίσθηση στην αλυσίδα scan και έτσι συμβάλλουν λιγότερο στη δραστηριότητα εναλλαγών που προκαλείται. Καθώς η παράμετρος P αυξάνεται, παρατηρείται παράλληλη αύξηση όχι μόνο στην κατανάλωση ενέργειας scan, αλλά και στο ποσοστό ανίχνευσης σφαλμάτων που επιφέρουν τα προκύπτοντα *test vectors*. Η τροποποιημένη έκδοση της MFA καλείται MFA+P και εύλογα διαπιστώνει κανείς ότι η MFA+0 ταυτίζεται με την MFA.

Στη 3^η στήλη του πίνακα 3.2 παρουσιάζονται δυο διαφορετικές περιπτώσεις συμπλήρωσης του test cube σύμφωνα με την μέθοδο MFA+20. Το test vector στην 1^η γραμμή παρουσιάζει μια μέτρια δραστηριότητα εναλλαγών, ενώ αυτό στη 2^η γραμμή τη χειρότερη. Και στις δυο περιπτώσεις η δραστηριότητα εναλλαγών είναι υψηλότερη συγκριτικά με αυτή της μεθόδου MFA, αλλά είναι αρκετά λιγότερη από αυτή της Random Fill.



3.1.2. Η παραγωγή των υποψήφιων *test vectors*

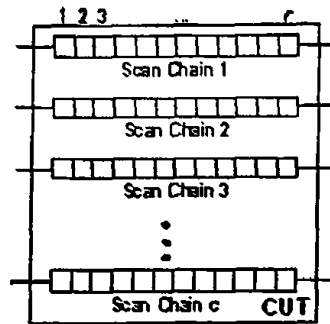
Είναι απαραίτητο να αναφερθεί ότι η μέθοδος FA επηρεάζει δυσμενώς τη μέγιστη *capture power*, η οποία είναι δυνατόν να υπερβεί και τη μέγιστη *capture power* που επιφέρει η τεχνική της τυχαίας συμπλήρωσης (Random Fill) [3]. Αυτό το γεγονός δεν μπορεί να αποφευχθεί και στις δυο προτεινόμενες μεθόδους MFA και MFA+P, διότι δεν είναι τίποτα άλλο παρά τροποποιήσεις της μεθόδου FA. Κατά συνέπεια, προκειμένου να περιοριστεί αυτό το μειονέκτημα και να διατηρηθεί η μέγιστη *capture power* κάτω από ένα αποδεκτό κατώφλι, εισάγεται η τεχνική *Preferred Fill* [17] για τον προσδιορισμό όσο το δυνατόν περισσότερων μη καθορισμένων bits. Γενικά, ο υπολογισμός της *capture power* πραγματοποιείται με χρήση της απόστασης Hamming ανάμεσα στο *test vector* και την πρώτη απόκριση. Αυτό το ζεύγος πάντα παράγει την μέγιστη ενέργεια [17].

Πρώτα απ' όλα, θα πρέπει να επισημάνουμε ότι προηγείται μια διεργασία πάνω στο αρχικό μας σύνολο από *test cubes* μεγέθους N , το οποίο για χάρη απλότητας καλούμε TSI (*test set initial*). Πιο συγκεκριμένα, για κάθε *test cube* του συνόλου TSI, εφαρμόζοντας την μέθοδο *Preferred Fill* [17], προκύπτουν 11 *test cubes*. Πιο αναλυτικά, το 1^ο *test cube* της 11αδας είναι το αρχικό μας, χωρίς καμία επίδραση από τη μέθοδο *Preferred Fill* και το ονομάζουμε για χάρη συντομίας CPF0 (*cube with Preferred Fill*). Το 2^ο *test cube* της 11αδας προκύπτει με την εφαρμογή της μεθόδου *Preferred Fill* στο 10% των μη καθορισμένων bits του αρχικού *test cube*, τα οποία είναι από τα πιο ικανά μεταξύ άλλων στο να διατηρήσουν σχετικά μειωμένη την μέγιστη *capture power* και το ονομάζουμε CPF10. Το 3^ο *test cube* (δηλ. CPF20) προκύπτει με την εφαρμογή της μεθόδου *Preferred Fill* στο 20% των πιο ικανών μη καθορισμένων bits, το 4^ο *test cube* (δηλ. CPF30) με την εφαρμογή της μεθόδου *Preferred Fill* στο 30% των πιο ικανών μη καθορισμένων bits, το 5^ο *test cube* (δηλ. CPF40) με την εφαρμογή της μεθόδου *Preferred Fill* στο 40% των πιο ικανών μη καθορισμένων bits κτλ. Έτσι, αν το αρχικό μας σύνολο TSI είναι μεγέθους N , τότε το προκύπτον σύνολο θα αποτελείται από N ομάδες των 11 *test cubes*, το οποίο για χάρη απλότητας το καλούμε TSF (*test set final*).

Επίσης, πρέπει να τονιστεί ότι κάθε *test cube* του συνόλου έχει τεθεί να φορτώνεται σε c ισοζυγισμένες αλυσίδες scan. Συνεπώς, αν το *test cube* είναι μεγέθους x , τότε



διαμερίζεται σε c τμήματα μέγιστου μεγέθους $r = \lceil x/c \rceil$. Στο Σχήμα 3.1 παρουσιάζεται η προαναφερθείσα διαδικασία.



Σχήμα 3.1 Σχηματικό ελέγχου με c ισοζυγισμένες αλυσίδες scan

Πρώτα απ' όλα, επεξεργαζόμαστε κάθε ομάδα των 11 test cubes του συνόλου TSF ανεξάρτητα. Όπως έχει προαναφερθεί, κάθε test cube μεγέθους x διασπάται σε c τμήματα μέγιστου μεγέθους $r = \lceil x/c \rceil$, προκειμένου να φορτωθεί στις c ισοζυγισμένες αλυσίδες scan. Μάλιστα, κάθε τμήμα συμπληρώνεται ανεξάρτητα, σύμφωνα με την μέθοδο MFA. Σε περίπτωση που από τη διάσπαση προκύψει τμήμα που αποτελείται μόνο από μη καθορισμένα bits, τότε αυτά συμπληρώνονται όλα είτε με 0, είτε με 1.

Για να γίνει πιο κατανοητή η τεχνική που μόλις περιγράφηκε, παρατίθεται στη συνέχεια ένα παράδειγμα. Έστω ότι έχουμε το ακόλουθο test cube $T = \text{XX01XXX010X0XXX1}$ μεγέθους $x=16$ bits και διασπάται σε 2 τμήματα προκειμένου να φορτώσει 2 ισοζυγισμένες αλυσίδες scan μεγέθους $r=8$ bits. Τότε τα προκύπτοντα τμήματα θα είναι τα εξής: $T_1 = \text{XX01XXX0}$ και $T_2 = \text{10X0XXX1}$. Καθένα από αυτά συμπληρώνεται ανεξάρτητα, σύμφωνα με την MFA. Έτσι, το τμήμα T_1 μπορεί να μετατραπεί σε μία από τις ακόλουθες περιπτώσεις: 00010000, 00011000, 00011100, 00011110 και το τμήμα T_2 σε μία από τις ακόλουθες περιπτώσεις: 10001111, 10000111, 10000011, 10000001. Εντέλει, το τελικό test vector, που θα φορτωθεί στις 2 ισοζυγισμένες αλυσίδες μεγέθους $r = 8$ bits, θα είναι ένας τυχαίος συνδυασμός των τμημάτων T_1 και T_2 . Μια πιθανή πρόταση λοιπόν είναι το test vector $T = 0001110010000111$.

Βασικός στόχος για καθεμία από τις N ομάδες των 11 *test cubes* (CPF0 έως CPF100) είναι να δημιουργηθούν CANDIDATES υποψήφια *test vectors*, τα οποία να επιφέρουν *capture power* μικρότερη από ένα δοθέν κατώφλι L . Ως κατώφλι L έχει τεθεί το 30% του πλήθους όλων των εισόδων του κυκλώματος. Αυτό σημαίνει ότι το πλήθος των επιτρεπτών bits, στα οποία μπορούν να διαφέρουν το *test vector* και η πρώτη απόκριση που λαμβάνεται από το κύκλωμα, να είναι το πολύ το 30% του μεγέθους του *test cube*.

Συγκεκριμένα, ξεκινώντας από το 1^ο *test cube* της ομάδας (δηλ. το CPF0), παράγουμε C-CANDIDATES υποψήφια *test vectors* σύμφωνα με τα προαναφερθέντα βήματα, όπου οι τιμές στις παραμέτρους C και CANDIDATES δίνονται από τον σχεδιαστή. Στη συνέχεια, όλα τα παραγόμενα C-CANDIDATES *test vectors* ελέγχονται για το αν παραβιάζουν το κατώφλι L και επιλέγονται αυτά που δεν το παραβιάζουν. Αν το πλήθος των επιλεγμένων *test vectors* είναι πάνω από CANDIDATES, τότε CANDIDATES από αυτά επιλέγονται τυχαία. Από την άλλη, αν το πλήθος είναι λιγότερο από CANDIDATES, τότε επιχειρούμε να το συμπληρώσουμε συνεχίζοντας στο 2^ο κατά σειρά *test cube* της ομάδας (δηλ. το CPF10). Για το 2^ο *test cube* της ομάδας (δηλ. το CPF20) παράγονται πάλι με τον ίδιο τρόπο C-CANDIDATES υποψήφια *test vectors*, εκ των οποίων αυτά που δεν παραβιάζουν το κατώφλι L προστίθενται στα ήδη επιλεγμένα. Αν πάλι τελικά το πλήθος είναι λιγότερο από CANDIDATES, τότε επαναλαμβάνεται η ίδια διαδικασία για το 3^ο *test vector* της ομάδας (δηλ. το CPF20) κτλ. Αν πλέον το πλήθος των επιλεγμένων υποψήφιων *test vectors* είναι ίσο με CANDIDATES ή η ροή έχει προχωρήσει μέχρι το 11^ο *test cube* της ομάδας (δηλ. το CPF100), το οποίο είναι πλήρως καθορισμένο, τότε προχωράμε στην επιλογή των CANDIDATES υποψήφιων *test vectors* της επόμενης ομάδας. Επίσης, πρέπει να αναφερθεί ότι στην περίπτωση που η ροή έχει ολοκληρώσει και με το 11^ο *test cube* της ομάδας (δηλ. το CPF100), χωρίς να έχει επιλεγθεί κανένα μέχρι τώρα *test vector*, γιατί τυχαίνει κανένα να μην ικανοποιεί το κατώφλι, αυτό που γίνεται είναι να αυξηθεί το κατώφλι κατά 5 και να επαναληφθεί η ίδια διαδικασία επιλογής των CANDIDATES υποψήφιων *test vectors* για την ίδια ομάδα. Αν πάλι ικανοποιούν το κατώφλι λιγότερα από CANDIDATES *test vectors*, τότε αντί να αυξηθεί το κατώφλι, αντιγράφεται το τελευταίο *test vector* όσο χρειάζεται για να συμπληρωθούν τα CANDIDATES *test vectors*. Έτσι, από κάθε ομάδα εντέλει θα

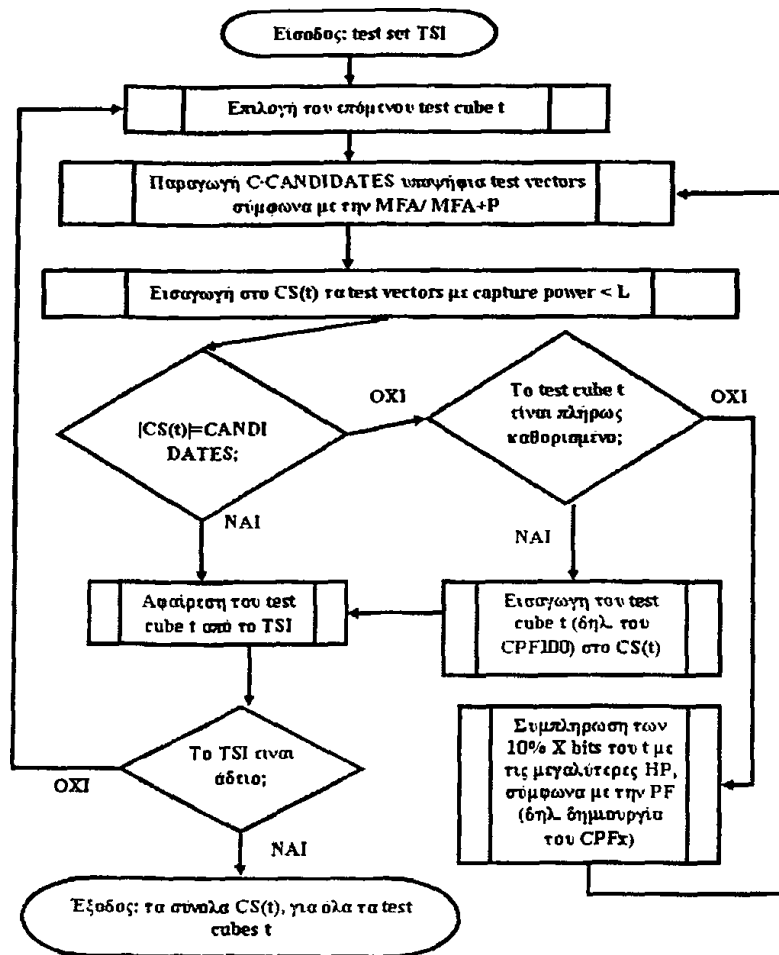


προκύψουν CANDIDATES υποψήφια *test vectors*, εκ των οποίων θα επιλεχθεί το πιο αποτελεσματικό. Ως πιο αποτελεσματικό, θεωρείται αυτό που θα ανιχνεύσει τα περισσότερα ελαττώματα στο δοθέν κύκλωμα όσο το δυνατόν γρηγορότερα, δηλαδή αυτό που έχει το μέγιστο *output deviation*.

Επιπλέον, η παραπάνω διαδικασία επιλογής των CANDIDATES υποψήφιων *test vectors* πάνω στο προκύπτον σύνολο από *test cubes* TSF εφαρμόζεται ομοίως και με την μέθοδο MFA+P. Βασιζόμενοι στο προηγούμενο παράδειγμα, σύμφωνα με τη μέθοδο MFA+10 το 10% αριστερότερο μέρος των τμημάτων $T_1 = \text{XX01XXX0}$ και $T_2 = 10\text{X0XXX1}$ θα συμπληρωθεί τυχαία και το υπόλοιπο σύμφωνα με την MFA. Δηλαδή, το 1^ο αριστερότερο bit σε περίπτωση που δεν είναι καθορισμένο θα πάρει τυχαία τη τιμή 1 ή 0 και το υπόλοιπο μέρος θα συμπληρωθεί ανεξάρτητα σύμφωνα με την μέθοδο MFA. Έτσι, για παράδειγμα το T_1 μπορεί να αναχθεί σε μία από τις εξής περιπτώσεις: 00010000, 00011000, 00011100, 00011110, 10010000, 10011000, 10011100, 10011110. Ομοίως, στη μέθοδο MFA+20 το 20% αριστερότερο μέρος των τμημάτων $T_1 = \text{XX01XXX0}$ και $T_2 = 10\text{X0XXX1}$ θα συμπληρωθεί τυχαία και το υπόλοιπο σύμφωνα με την προτεινόμενη μέθοδο. Δηλαδή, τα δυο πρώτα αριστερότερα bits σε περίπτωση που δεν είναι καθορισμένα θα πάρουν τυχαία τη τιμή 1 ή 0 και το υπόλοιπο μέρος θα συμπληρωθεί ανεξάρτητα σύμφωνα με την μέθοδο MFA. Στη συγκεκριμένη περίπτωση, το T_1 μπορεί να αναχθεί σε μία από τις εξής περιπτώσεις: 00010000, 00011000, 00011100, 00011110, 01010000, 01011000, 01011100, 01011110, 11010000, 11011000, 11011100, 11011110. Εντέλει, το τελικό *test vector*, που θα φορτωθεί στις 2 ισοζυγισμένες αλυσίδες μεγέθους $r = 8$ bits, θα είναι ένας τυχαίος συνδυασμός των τμημάτων T_1 και T_2 .

Στο Σχήμα 3.2 παρουσιάζεται συνοπτικά η προαναφερθείσα διαδικασία παραγωγής των υποψήφιων *test vectors* για κάθε *test cube* εφαρμόζοντας είτε την μέθοδο MFA είτε την MFA+P. Διευκρινίζεται ότι το σύνολο CS(t) θα περιέχει τα υποψήφια *test vectors* για το *test cube* t του συνόλου TSI ή εναλλακτικά για κάθε 11δα που δημιουργείται με βάση το *test cube* t.





Σχήμα 3.2 Συνοπτικό διάγραμμα παραγωγής υποψήφιας test vectors (MFA/MFA+P).

3.1.3. Εκτίμηση και επιλογή των test vectors

Καθένα από τα CANDIDATES υποψήφια test vectors για καθεμία από τις N ομάδες αξιολογούνται με βάση την έννοια του *output deviation* και επιλέγεται το καλύτερο για κάθε αρχικό test cube του συνόλου TSI. Υπογραμμίζεται ότι για την εκτίμηση και επιλογή των βέλτιστων test vectors έχει εφαρμοστεί η τεχνική που περιγράφεται στο [9]. Επιπλέον, υπενθυμίζεται ότι τα *output deviations* είναι πιθανότητες που σχετίζονται με τις εξόδους του κυκλώματος και υποδηλώνουν τη πιθανότητα να ανιχνευθεί κάποιο σφάλμα σε αυτές τις εξόδους. Γενικά, τα διανύσματα εισόδου με υψηλά *output deviations* είναι πιο αποτελεσματικά στην ανίχνευση σφαλμάτων. Τελικά, θα προκύψουν τόσα test vectors όσες οι ομάδες ή αλλιώς όσα τα αρχικά test cubes του συνόλου TSI και θα είναι ταξινομημένα σύμφωνα με την αποτελεσματικότητά τους. Πρώτα δηλαδή θα εφαρμόζονται τα πιο αποτελεσματικά,



ούτως ώστε να ανιχνεύονται όσο το δυνατόν νωρίτερα τα στοχευόμενα σφάλματα σε ένα κύκλωμα.

Στην ουσία, η μετρική που χρησιμοποιείται για την αξιολόγηση των *test vectors* λαμβάνει υπόψη τη δυνατότητα ανίχνευσης σφαλμάτων κάθε *test vector* τόσο στην 1^η απόκρισή του, όσο και στη 2^η. Στην προκειμένη περίπτωση, έχει εφαρμοστεί η τεχνική LOC. Συμπερασματικά, τα στοχευόμενα σφάλματα είναι εξαρτώμενα και μη από το χρόνο.

Η μετρική για την αξιολόγηση των *test vectors* εκμεταλλεύεται τις ακόλουθες ιδιότητες:

A) Για κάθε υποψήφιο *test vector* v , υπολογίζονται τα *output deviations* σε όλες τις εξόδους και για τις δυο αποκρίσεις του. Στη συνέχεια, οι έξοδοι, όπου τα *output deviations* είναι ιδιαίτερα υψηλά ανάμεσα σε όλα τα υποψήφια *test vectors*, είναι οι πιο ιδανικές για την ανίχνευση ελαττωμάτων, ενώ οι υπόλοιπες έξοδοι παραλείπονται. Οι αποτελεσματικές αυτές έξοδοι χωρίζονται για κάθε *test vector* j σε τέσσερα σύνολα. Για την 1^η απόκριση του διανύσματος j έχουμε τα σύνολα $MS_0(j,0)$ και $MS_0(j,1)$, ενώ για την 2^η απόκριση τα σύνολα $MS_1(j,0)$ και $MS_1(j,1)$. Ουσιαστικά, για την m (1^η ή 2^η) απόκριση, το σύνολο $MS_m(j,0)$ περιέχει τις εξόδους με τα μέγιστα *output deviations* και μη εσφαλμένη λογική τιμή 0, ενώ το $MS_m(j,1)$ περιέχει τις εξόδους με τα μέγιστα *output deviations* και μη εσφαλμένη λογική τιμή 1.

Τώρα, για την αναγνώριση εκείνων των εξόδων που έχουν υψηλά *output deviations*, είτε κατά τη φάση λήψης της 1^{ης} απόκρισης είτε κατά τη φάση λήψης της 2^{ης}, ανάμεσα σε όλα τα υποψήφια *test vectors* προηγείται μια βασική διαδικασία. Πιο αναλυτικά, σχετικά με την επιλογή του βέλτιστου *test vector* για ένα *test cube* του συνόλου TSI, εφαρμόζεται αρχικά το σύνολο όλων των υποψήφιων *test vectors* σε ένα κύκλωμα. Αυτά τα *test vectors* διαχωρίζονται σε 4 σύνολα για κάθε έξοδο i . Το ένα σύνολο θα περιέχει τα *test vectors* που δίνουν ως 1^η απόκριση χωρίς σφάλματα τη τιμή 0, το δεύτερο αυτά που δίνουν ως 1^η απόκριση χωρίς σφάλματα τη τιμή 1, το τρίτο αυτά που δίνουν ως 2^η απόκριση χωρίς σφάλματα τη τιμή 0 και το τελευταίο αυτά που δίνουν ως 2^η απόκριση χωρίς σφάλματα τη τιμή 1. Στη συνέχεια, υπολογίζονται τα *output deviations* όλων των εξόδων για όλα τα υποψήφια *test*



vectors και για τις τέσσερις περιπτώσεις ξεχωριστά. Σε κάθε έξοδο i , για κάθε μία από τις τέσσερις περιπτώσεις το μέγιστο *output deviation* θα συνιστά το Maximum Deviation $MD_m(i, v)$, όπου $m, v = 0, 1$. Κατόπιν, έστω ότι η έξοδος i για το j -οστό υποψήφιο *test vector* από τα CANDIDATES υποψήφια *test vectors* του s -οστού αρχικού *test cube* (ή αλλιώς της s -οστής ομάδας) έχει στην m ($1^{\text{η}}$ ή $2^{\text{η}}$) απόκριση την μη εσφαλμένη τιμή v και *output deviation* $D_m(s, j, i, v)$. Η τιμή $D_m(s, j, i, v)$ θεωρείται υψηλή αν προσεγγίζει την $MD_m(i, v)$ ή είναι ίση. Δηλαδή, αν ισχύει ο παρακάτω τύπος:

$$D_m(s, j, i, v) \geq F_1 \cdot MD_m(i, v), \quad v = 0, 1 \text{ και } m=0, 1 \quad \text{Εξ. 3.1}$$

Το F_1 είναι ένας πραγματικός αριθμός που πρέπει να προσεγγίζει το 1 για την επιλογή εκείνων των *test vectors* που έχουν τόσο *output deviation*, όσο το $MD_m(i, v)$. Η προτεινόμενη τιμή της ανήκει στο διάστημα $[0.99 \ 0.995]$ [9].

B) Το βάρος κάθε εξόδου ενός κυκλώματος καθορίζεται από τη πιθανότητα να ανιχνευθούν σε αυτή την έξοδο αρκετά ελαττώματα. Το βάρος αυτό εξαρτάται κυρίως από: α) το μέγεθος του λογικού κώνου της αντίστοιχης εξόδου, μιας και περισσότερα σφάλματα μπορούν να ανιχνευθούν σε εξόδους με μεγάλους λογικούς κώνους, παρά με μικρούς, β) τη μη εσφαλμένη λογική τιμή κάθε εξόδου, διότι διαφορετικά σφάλματα εντοπίζονται σε μία έξοδο όταν έχει τη μη εσφαλμένη λογική τιμή 0 και διαφορετικά σφάλματα όταν έχει τη μη εσφαλμένη λογική τιμή 1, και γ) το πλήθος των ελαττωμάτων σε κάθε κώνο τα οποία δεν ανιχνεύτηκαν από τα προηγούμενα επιλεγμένα *test vectors* για την αντίστοιχη έξοδο. Το πλήθος των ελαττωμάτων σε κάθε κώνο της εξόδου μπορεί να εκτιμηθεί σε σχέση με το πλήθος το προηγούμενων επιλεγμένων *test vectors* που έχουν υψηλά *output deviations* για την αντίστοιχη έξοδο. Δηλαδή, όσο μεγαλύτερο είναι το πλήθος το προηγούμενων επιλεγμένων *test vectors*, τόσο υψηλότερο είναι το ποσοστό των ελαττωμάτων που έχουν ήδη ανιχνευθεί στην αντίστοιχη έξοδο και έτσι περιορίζεται το ποσοστό των ελαττωμάτων που εναπομένουν προς ανίχνευση.

Στη συνέχεια, η αξιολόγηση των CANDIDATES υποψήφιων *test vectors* για κάθε *test cube* του συνόλου TSI διεξάγεται βασισόμενη στις προαναφερθείσες ιδιότητες. Αρχικά, σε κάθε έξοδο i έχει ανατεθεί το ζεύγος βαρών $w_{00}(i, 0)$ και $w_{00}(i, 1)$ για την $1^{\text{η}}$ απόκριση με μη εσφαλμένες τιμές 0, 1 αντίστοιχα και το ζεύγος βαρών $w_{01}(i, 0)$



και $w_{01}(i, 1)$ για την 2^η απόκριση με μη εσφαλμένες τιμές 0, 1 αντίστοιχα. Τα βάρη αυτά έχουν αρχικοποιηθεί στο πλήθος των εισόδων του λογικού κώνου αυτής της εξόδου. Γενικά, αυτά τα βάρη αποτελούν ένδειξη του πλήθους των ελαττωμάτων που δεν έχουν ακόμα εντοπιστεί. Τόσο το σύνολο των βαρών $\{w_{00}(i, 0), w_{00}(i, 1), w_{01}(i, 0), w_{01}(i, 1)\}$, όσο και οι τιμές των *output deviations*, θα χρησιμοποιηθούν στον καθορισμό του βάρους καθενός από τα CANDIDATE υποψήφια *test vectors* κάθε *test cube*.

Στη συνέχεια, ενοποιούνται όλα τα CANDIDATES υποψήφια *test vectors* όλων των ομάδων του συνόλου TSF σε ένα σύνολο που για χάρη απλότητας το ονομάζουμε UTS (unified test set). Για καθένα *test vector* j του συνόλου UTS υπολογίζονται τα *output deviations* στις εξόδους και κατά συνέπεια ορίζονται τα σύνολα $MS_m(j, 0)$ και $MS_m(j, 1)$, όπου $m=0, 1$. Υπενθυμίζεται ότι τα σύνολα αυτά αποτελούνται από εκείνες τις εξόδους i , με $1 \leq i \leq k$, όπου k το πλήθος των εξόδων στο κύκλωμα, για τις οποίες οι τιμές των *output deviations* $D_m(s, j, i, v)$, όπου $m, v = 0, 1$ και s η ομάδα (ή αλλιώς το αρχικό test cube), ικανοποιούν τον τύπο της Εξίσωσης (3.1). Κατόπιν, η ακόλουθη διαδικασία επαναλαμβάνεται και σε κάθε επανάληψη ουσιαστικά επιλέγεται ένα *test vector*. Αρχικά, χρησιμοποιείται ο ακόλουθος τύπος που υπολογίζεται για κάθε *test vector*:

$$WS(j) = \sum_{m=0,1} \sum_{v=0,1} \left[\sum_{i \in MS_m[j,v]} w_{0m}(i, v) \right] \quad \text{Εξ. 3.2}$$

Με άλλα λόγια, το $WS(j)$ του j -οστού *test vector*, είναι το άθροισμα των βαρών εκείνων των εξόδων των οποίων τα *output deviations*, είτε στην 1^η είτε στη 2^η απόκριση, προσεγγίζουν την τιμή του αντίστοιχου μέγιστου deviation. Έτσι, ανάμεσα σε όλα τα υπολογισμένα *test vectors*, επιλέγεται το *test vector* x με τη μέγιστη τιμή στον τύπο της Εξίσωσης (3.2). Ο τύπος αυτός δίνει τη δυνατότητα επιλογής εκείνων των *test vectors*, που αυξάνουν τις τιμές των *output deviations* στις εξόδους μεγάλων κώνων ενός κυκλώματος. Όσο μεγαλύτεροι είναι οι κώνοι, τόσο αυξάνονται οι πιθανότητες ανίχνευσης των ελαττωμάτων. Με την επιλογή του *test vector* x που ανήκει, για παράδειγμα, στο σύνολο από CANDIDATES υποψήφια *test vectors* της p -οστής ομάδας, έχει ουσιαστικά επιλεγεί το βέλτιστο για αυτή την ομάδα. Έτσι τα υπόλοιπα υποψήφια *test vectors* αυτής της ομάδας διαγράφονται από το ενοποιημένο



σύνολο UTS. Η επιλογή του βέλτιστου *test vector* για κάθε ομάδα γίνεται έτσι ώστε να αποφεύγεται η αύξηση των *output deviations* σε ένα μόνο υποσύνολο εξόδων, το οποίο είναι δυνατόν να οδηγήσει σε μείωση των ανιχνεύσιμων σφαλμάτων. Έτσι, κάθε φορά που επιλέγεται κάποιο *test vector* x , όλα τα βάρη $w_m(i, v)$ για όλες τις εξόδους $i \in MS_m(x, v)$, όπου $v=0, 1$ είναι οι μη εσφαλμένες τιμές για τις δυο αποκρίσεις, διαιρούνται με έναν σταθερό παράγοντα F_2 . Με αυτή τη μέθοδο, οι έξοδοι με μειωμένα βάρη ουσιαστικά δεν συνυπολογίζονται στην επιλογή του επόμενου βέλτιστου *test vector* μιας ομάδας. Ο παράγοντας F_2 έχει τιμή 8, καθώς έχει αποδειχθεί πειραματικά ότι το επιτρεπτό εύρος τιμών του είναι [2, 10] προκειμένου να εξασφαλιστεί η μεγιστοποίηση του *deviation* σε όλες τις εξόδους [9]. Κατόπιν, επαναλαμβάνεται ο υπολογισμός του τύπου της Εξίσωσης (3.2) για τα εναπομείναντα *test vectors*, προκειμένου να επιλεγεί το επόμενο βέλτιστο *test vector* κάποιου άλλου αρχικού *test cube*. Με αυτή τη μεθοδολογία, επιλέγονται τα βέλτιστα *test vectors* όλων των N ομάδων του συνόλου TSF (ή αλλιώς όλων των N αρχικών *test cubes* του συνόλου TSI) σε ταξινομημένη σειρά.

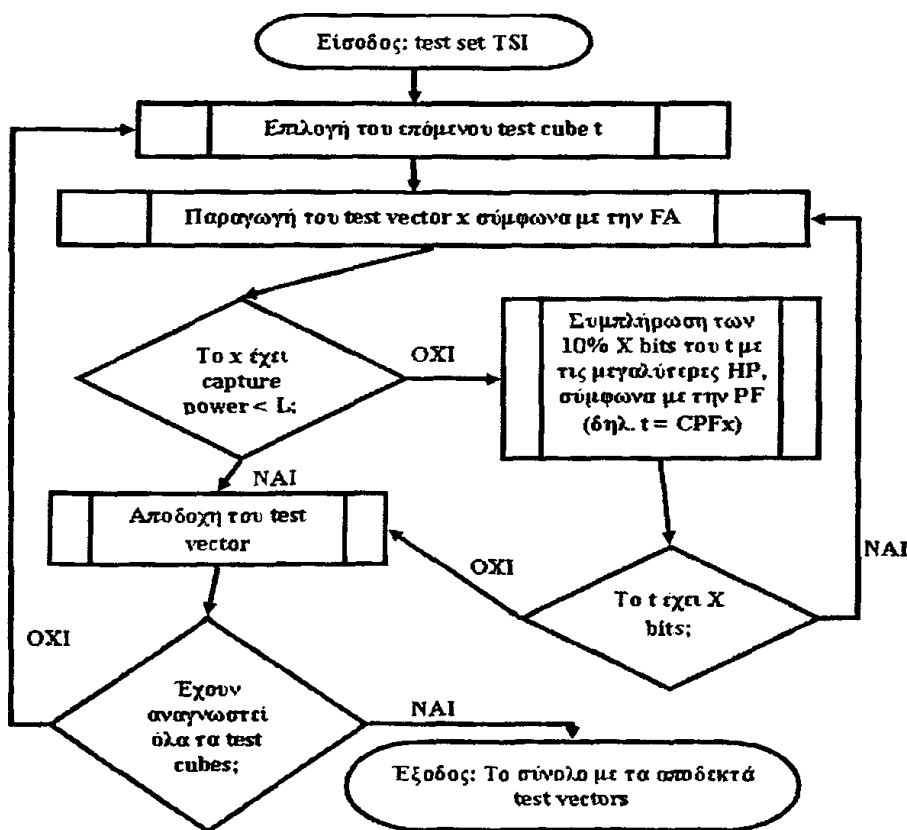
3.2. Οι τεχνικές *Fill Adjacent* και *Random Fill*

Προκειμένου να υπάρξει μια πιο λεπτομερής σύγκριση, έχουν εφαρμοστεί η τεχνική *Fill Adjacent* και η τεχνική της τυχαίας συμπλήρωσης (*Random Fill*). Η τεχνική της τυχαίας συμπλήρωσης εφαρμόζεται απευθείας στο αρχικό σύνολο TSI από N *test cubes*. Από την άλλη μεριά, τεχνική *Fill Adjacent* εφαρμόζεται στο προκύπτον σύνολο TSF των N ομάδων από 11 *test cubes* η κάθε μία. Δηλαδή, προκύπτει μια υβριδική μέθοδος που δεν είναι τίποτα άλλο παρά ένας συνδυασμός των *Fill Adjacent* και *Preferred Fill*, προκειμένου η μέγιστη *capture power* να μην επιφέρει αρκετές δυσμενείς υπερβάσεις και η σύγκριση με τις προτεινόμενες μεθόδους να είναι πιο ισορροπημένη. Αυτή η υβριδική μέθοδος καλείται ως FA*.

Πιο αναλυτικά, για την τεχνική FA*, παρομοίως επεξεργαζόμαστε κάθε μία από τις N ομάδες ανεξάρτητα. Υπενθυμίζεται ότι κάθε *test cube* διαμοιράζεται σε c τμήματα μέγιστου μεγέθους $r = \lceil x/c \rceil$ και κάθε τμήμα συμπληρώνεται ανεξάρτητα σύμφωνα με τη τεχνική *Fill Adjacent*. Ξεκινώντας λοιπόν με το 1^ο *test cube* της ομάδας (δηλ. το CPF0), παράγεται ένα *test vector* συμπληρωμένο σύμφωνα με την τεχνική *Fill*



Adjacent και μετά ελέγχεται αν έχει *capture power* μικρότερη από ένα δοθέν κατώφλι L , το οποίο έχει τεθεί να είναι το 30% του πλήθους όλων των εισόδων του κυκλώματος. Αν δεν παραβιάζεται το κατώφλι, τότε επιλέγεται και συνεχίζουμε στην επόμενη ομάδα. Αν πάλι παραβιάζεται, τότε προχωράμε στο 2^ο *test cube* της ομάδας (δηλ. το CPF10), στο οποίο το 10% των μη καθορισμένων bits έχει ήδη συμπληρωθεί με τη μέθοδο *Preferred Fill*, και αφού εφαρμοστεί η *Fill Adjacent* ελέγχεται η τήρηση του κατωφλίου. Αν πάλι σημειώνεται παράβαση, τότε προχωράμε στο επόμενο κατά σειρά *test cube* της ίδιας ομάδας. Στην περίπτωση που η ροή σε μια ομάδα έχει φθάσει και στο 11^ο *test cube* (δηλ. το CPF100), στο οποίο το 100% των μη καθορισμένων bits έχει ήδη συμπληρωθεί με τη μέθοδο *Preferred Fill*, και σημειώνεται πάλι παραβίαση του κατωφλίου, τότε αυξάνεται το κατώφλι κατά 5 και επαναλαμβάνεται εξ αρχής η ίδια διαδικασία για αυτή την ομάδα. Συνεπώς, κάθε επιλεγμένο *test vector* από κάθε ομάδα απαρτίζει εντέλει το σύνολο των *test vectors* μεγέθους N , που θα χρησιμοποιηθεί στη φάση ελέγχου ενός κυκλώματος. Στο Σχήμα 3.3 παραθέτουμε ένα συνοπτικό διάγραμμα της προαναφερθείσας διαδικασίας.



Σχήμα 3.3 Συνοπτικό διάγραμμα παραγωγής test vectors από την FA*



3.3. Εκτίμηση του μέσου όρου των ενεργειών scan-in και scan-out

Κάθε μέθοδος λοιπόν, θα επιφέρει το δικό της σύνολο από *test vectors*. Μπορούμε πλέον βασιζόμενοι στη μετρική *weighted transition* [5][15][18] να υπολογίσουμε και τον μέσο όρο κατανάλωσης ενέργειας κατά την εισαγωγή με ολίσθηση κάθε *test vector* στις αλυσίδες scan για κάθε μέθοδο ξεχωριστά. Επιπλέον, εφαρμόζοντας την τεχνική LOC (launch-on-capture) για τη λήψη των 2 αποκρίσεων για κάθε *test vector*, μπορούμε να υπολογίσουμε την κατανάλωση ενέργειας κατά την εξαγωγή με ολίσθηση κάθε απόκρισης από την αλυσίδα scan μεγέθους k . Αυτό επιτυγχάνεται με μια τροποποίηση της κανονικοποιημένης μετρικής *weighted transition* η οποία είναι η ακόλουθη για ένα *test cube* T μεγέθους k :

$$P_{so}(T) = 2 \left[k(k-1) \right]^{-1} \sum_{i=1}^{k-1} i(x_i \oplus x_{i+1}) \quad \text{Εξ. 3.3}$$

Ο μέσος όρος κατανάλωσης ενέργειας κατά την εξαγωγή κάθε απόκρισης από τις αλυσίδες scan για κάθε μέθοδο ξεχωριστά μπορεί να υπολογιστεί αν προστεθούν τα αποτελέσματα του παραπάνω τύπου για όλες τις αλυσίδες scan και για όλες τις αποκρίσεις του κυκλώματος για όλα τα *test vectors*.



ΚΕΦΑΛΑΙΟ 4. ΠΕΙΡΑΜΑΤΑ

4.1 Πειράματα

4.2 Συνολικά αποτελέσματα

4.1. Πειράματα

Στο παρόν κεφάλαιο μελετάται και ταυτόχρονα συγκρίνεται η αποτελεσματικότητα των προτεινόμενων μεθόδων MFA και MFA+P. Επισημαίνεται ότι οι μέθοδοι MFA, MFA+P μαζί με τις μεθόδους FA* και την Random Fill έχουν υλοποιηθεί στη γλώσσα προγραμματισμού C. Τα σύνολα δεδομένων, πάνω στα οποία εφαρμόστηκαν όλες οι μέθοδοι, προέκυψαν από το εμπορικό εργαλείο ATPG με δυναμική συμπίεση, και παράλληλα παρέχουν πλήρη κάλυψη σφαλμάτων τύπου stuck-at. Επίσης, βασιζόμενοι στα προκύπτοντα από κάθε μέθοδο σύνολα από *test vectors*, η διεξαγωγή προσομοιώσεων για την εύρεση του ποσοστού ανίχνευσης σφαλμάτων και οι υπολογισμοί της κατανάλωσης ενέργειας επιτυγχάνονται μέσω των εμπορικών εργαλείων TetraMax και PrimePower του Synopsys, αντίστοιχα. Συγκεκριμένα, το πλήθος των υποψήφιων *test vectors* για κάθε *test cube* έχει τεθεί ίσο με 30 (CANDIDATES=30) και η παράμετρος C ίση με 3. Τα πειράματα διεξήχθησαν σε κυκλώματα των κατηγοριών ISCAS'89 και IWLS'05. Από την 1^η κατηγορία μελετώνται τα κυκλώματα s5378, s9234, s13207, s15850, s38417 και s38584 με πλήθος αλυσίδων scan 2, 2, 5, 5, 10 και 10, αντίστοιχα. Από την 2^η κατηγορία μελετώνται τα κυκλώματα ac97_ctrl, mem_ctrl, pci_bridge32, tv80, usb_funct και ethernet με πλήθος αλυσίδων scan 15, 8, 20, 2, 10 και 50, αντίστοιχα.



4.1.1. Πειράματα για την κατανάλωση ενέργειας

Αναφέρεται ότι εξαιτίας της δυναμικής συμπίεσης που εκτελείται από το εργαλείο ATPG, τα πρώτα *test cubes* που προκύπτουν έχουν πολλά καθορισμένα bits. Αυτό έχει ως συνέπεια να μην μπορεί να εφαρμοστεί πλήρως η τεχνική *Preferred Fill*, προκειμένου να μειωθεί η *capture power* κάτω από ένα καθορισμένο κατώφλι. Έτσι, αυτά τα *test cubes* αντικαθίστανται από άλλα με λιγότερα καθορισμένα bits, τα οποία παράγονται ενεργοποιώντας ξανά το εργαλείο ATPG. Όπως έχει αναφερθεί και σε προηγούμενο κεφάλαιο, το άνω όριο στην κατανάλωση της *capture power*, δηλαδή το L , έχει επιλεγεί να είναι το 30% των κυττάρων όλων των αλυσίδων scan, στις οποίες φορτώνονται τα *test vectors*. Πρέπει να επισημανθεί ότι η επιλογή της συγκεκριμένης τιμής του άνω ορίου L δεν έγινε τυχαία. Συγκεκριμένα, οφείλεται στο γεγονός ότι η συμπαγής μορφή των *test cubes* αποτρέπει την πιθανότητα ύπαρξης άλλων άνω ορίων της *capture power* χαμηλότερων από την L ($= 30\%$), χωρίς να συμβούν παραβιάσεις. Συγκεκριμένα, όταν το L παίρνει χαμηλότερες τιμές από 30%, τότε παρατηρείται ότι η τεχνική *Preferred Fill* αδυνατεί να παράγει *test vectors* για κάποια *test cubes*, όπου η *capture power* να μην παραβιάζει το όριο. Επίσης, όπως θα διαπιστωθεί στη συνέχεια, η επιλογή του L δεν επηρεάζει ιδιαίτερα την αποτελεσματικότητα των προτεινόμενων μεθόδων. Για την εκτίμηση της κατανάλωσης ενέργειας, δημιουργήθηκε αρχικά από ένα αρχείο για κάθε μέθοδο. Σε καθένα από αυτά τα αρχεία εμπεριέχονται όλες οι ενδιάμεσες εναλλαγές που συμβαίνουν στις αλυσίδες scan έως ότου εισαχθεί εξ' ολοκλήρου κάθε *test vector* του συνόλου καθώς και οι δυο αποκρίσεις του. Στη συνέχεια, με τη χρήση του PrimePower εκτιμώνται ο μέσος όρος της ολικής κατανάλωσης ενέργειας (scan-in, capture, scan-out), ο μέσος όρος της *capture power* και η μέγιστη ενέργεια σε ένα κύκλωμα. Επίσης, με χρήση της μετρικής *weighted transition* υπολογίζεται η ενέργεια που καταναλώνεται κατά τις ολισθήσεις των bits στις αλυσίδες scan.

Στον Πίνακα 4.1 παρουσιάζονται οι μέθοδοι FA, FA*, MFA, MFA+10 και MFA+20. Υπογραμμίζεται ότι η FA* είναι η υβριδική έκδοση της *Fill Adjacent* και της *Preferred Fill* και η FA είναι η *Fill Adjacent* εφαρμοσμένη πάνω στο αρχικό σύνολο των *test cubes*. Οι πρώτες δυο στήλες παρουσιάζουν τα ονόματα των κυκλωμάτων και το μέγεθος των αρχικών συνόλων από *test cubes* που εφαρμόζονται σε κάθε κύκλωμα για όλες ανεξαιρέτως τις μεθόδους. Οι υπόλοιπες στήλες παρουσιάζουν το



ποσοστό μείωσης της κατανάλωσης της ολικής μέσης ενέργειας που επιτυγχάνεται σε κάθε κύκλωμα από κάθε μέθοδο σε σύγκριση με τη μέθοδο Random Fill (RF). Είναι προφανές ότι η FA επιτυγχάνει τη μεγαλύτερη μείωση, αλλά όπως έχει ήδη αναφερθεί έχει ανεπιθύμητα αποτελέσματα όσον αφορά την *capture power*. Στην περίπτωση της FA*, επειδή έχει περισσότερα καθορισμένα bits λόγω της τεχνικής *Preferred Fill*, αυξάνει την ολική μέση κατανάλωση ενέργειας σε σύγκριση με την FA. Επίσης, οι FA* και MFA προκαλούν παρόμοιες μειώσεις στη ολική μέση κατανάλωση ενέργειας σε σύγκριση με την RF. Οι μέθοδοι MFA+10 και MFA+20 παρέχουν χαμηλότερες μειώσεις, οι οποίες παραμένουν σημαντικές. Παρατηρείται ότι σε ορισμένες περιπτώσεις η MFA είναι λίγο καλύτερη από την FA*. Το γεγονός αυτό οφείλεται κυρίως στο ότι η FA* τείνει να έχει *test vectors*, που προέρχονται από *test cubes* με περισσότερη επιρροή από την τεχνική *Preferred Fill*, σε αντίθεση με την MFA. Η MFA παράγει αρκετά υποψήφια *test vectors* για κάθε *test cube* της κάθε 11αδας, με αποτέλεσμα να αυξάνεται η πιθανότητα επιλογής *test vectors* που πληρούν τον περιορισμό της *capture power* και παράλληλα προέρχονται από τα πρώτα cubes της 11άδας με τη λιγότερη επιρροή από την τεχνική *Preferred Fill*.

Πίνακας 4.1 Ποσοστό μείωσης ολικής μέσης ενέργειας σε σύγκριση με την RF (με χρήση του εργαλείου PrimePower of Synopsys)

circuit	# cubes	FA	FA*	MFA	MFA+10	MFA+20
s5378	134	59.3%	43.42%	47.28%	34.84%	29.07%
s9234	166	41.15%	37.65%	38.49%	34.75%	30%
s13207	269	52.05%	49.61%	50.89%	48.20%	45.20%
s15850	162	56.92%	56.49%	56.57%	51.90%	47.50%
s38417	143	64.87%	64.80%	64.21%	59.86%	56.13%
s38584	185	57.22%	56.66%	56.94%	52.09%	47.17%
ac97_ctrl	66	54.44%	54.69%	53.67%	50.57%	45.94%
mem_ctrl	603	73.59%	73.40%	71.98%	65.61%	58.47%
pci_bridge32	298	67.98%	68.26%	67.69%	62.38%	56.66%
tv80	757	64.14%	64.18%	63.13%	57.84%	53.60%
usb_funct	136	45.37%	42.04%	42.58%	40.26%	37.69%
ethernet	1113	90.00%	90.19%	89.81%	80.83%	71.69%

Επιπρόσθετα, στον Πίνακα 4.2 παρουσιάζονται τα ποσοστά μείωσης της ενέργειας, που καταναλώνεται κατά την εισαγωγή με ολίσθηση των *test vectors* στις αλυσίδες scan (*scan-in power*), για κάθε κύκλωμα από κάθε μέθοδο σε σύγκριση με την RF.



Υπογραμμίζεται ότι για τον υπολογισμό της μέσης *scan-in power* προστέθηκαν τα αποτελέσματα της Εξίσωσης (2.1) για όλες τις αλυσίδες *scan* και για όλα τα *test vectors*. Η δεύτερη στήλη εμπεριέχει το πλήθος των ισοζυγισμένων αλυσίδων *scan* που χρησιμοποιούνται σε κάθε κύκλωμα. Παρομοίως, στον Πίνακα 4.3 παρουσιάζονται τα ποσοστά μείωσης της ενέργειας, που καταναλώνεται κατά την εξαγωγή με ολίσθηση των αποκρίσεων των κυκλωμάτων στα αντίστοιχα *test vectors* από τις αλυσίδες *scan* (*scan-out power*), για κάθε κύκλωμα από κάθε μέθοδο σε σύγκριση με την RF. Ομοίως, για τον υπολογισμό της μέσης *scan-out power* προστέθηκαν τα αποτελέσματα της Εξίσωσης (3.3) για όλες τις αλυσίδες *scan* και για όλα τα *test vectors*. Στην περίπτωση της ενέργειας που καταναλώνεται κατά την εισαγωγή με ολίσθηση των *test vectors* στις αλυσίδες *scan*, όπως είναι αναμενόμενο, η μέθοδος FA* επιτυγχάνει λιγότερη μείωση συγκριτικά με την *Fill Adjacent* (FA), εξαιτίας της ύπαρξης επιπλέον καθορισμένων bits από την εφαρμογή της τεχνικής *Preferred Fill*. Ωστόσο η FA* επιφέρει αξιόλογη μείωση, η οποία προσεγγίζεται και σε ορισμένα κυκλώματα υπερβαίνεται από την προτεινόμενη μέθοδο MFA. Ομοίως, και οι άλλες δυο μέθοδοι MFA+10 και MFA+20. Παρατηρώντας και την περίπτωση της ενέργειας που καταναλώνεται κατά την εξαγωγή με ολίσθηση των αποκρίσεων των κυκλωμάτων στα αντίστοιχα *test vectors* από τις αλυσίδες *scan*, η μέθοδος FA* επιτυγχάνει σε μερικά κυκλώματα αξιόλογη μείωση, όπως και η FA, ενώ σε άλλα όχι και ιδιαίτερα ικανοποιητική. Την πορεία της FA* ακολουθεί και η προτεινόμενη μέθοδος MFA, ενώ οι άλλες δυο επιφέρουν μεν μείωση αλλά όχι με την ίδια απόδοση όσο οι FA* και MFA.

Πίνακας 4.2 Ποσοστό μείωσης *scan-in power* σε σύγκριση με την RF (με χρήση της μετρικής *weighted transition*)

Scan In Power Reduction Compared to Random						
Circuit	#sc	FA	FA*	MFA	MFA+10	MFA+20
s5378	2	75.96%	45.53%	53.13%	51.58%	48.84%
s9234	2	72.68%	64.25%	65.94%	65.84%	63.81%
s13207	5	94.61%	87.49%	89.62%	88.83%	85.83%
s15850	5	87.65%	85.31%	86.13%	85.36%	82.96%
s38417	10	82.99%	83.01%	82.75%	81.94%	79.52%
s38584	10	86.99%	82.74%	83.35%	81.48%	77.61%
ac97_ctl	15	85.04%	84.70%	83.89%	81.31%	64.61%
mem_ctl	8	95.79%	95.73%	95.56%	94.68%	92.01%
pci_bridge32	20	93.98%	93.86%	92.81%	90.04%	70.79%
tv80	2	91.09%	90.85%	90.53%	89.78%	87.44%
usb_funct	10	79.85%	73.13%	73.64%	72.76%	70.70%
ethernet	50	98.89%	98.89%	98.74%	97.77%	94.93%



Πίνακας 4.3 Ποσοστό μείωσης scan-out power σε σύγκριση με την RF (με χρήση της τροποποιημένης μετρικής weighted transition)

Scan Out Power Reduction Compared to Random						
Circuit	#sc	FA	FA*	MFA	MFA+10	MFA+20
s5378	2	48.40%	45.29%	46.94%	21.27%	12.72%
s9234	2	22.73%	23.40%	23.56%	15.96%	7.51%
s13207	5	18.84%	19.41%	19.29%	15.11%	11.01%
s15850	5	24.95%	28.05%	27.40%	19.26%	15.02%
s38417	10	46.87%	46.59%	45.54%	37.11%	31.35%
s38584	10	15.81%	21.29%	20.94%	15.26%	11.35%
ac97_ctrl	15	32.65%	33.15%	32.00%	26.57%	19.86%
mem_ctrl	8	62.73%	62.55%	59.74%	48.84%	37.36%
pci_bridge32	20	52.51%	52.98%	51.95%	42.45%	33.81%
tv80	2	48.57%	48.75%	46.85%	37.47%	32.24%
usb_funct	10	16.84%	17.64%	18.15%	14.20%	10.95%
ethernet	50	85.00%	85.35%	84.83%	69.70%	55.65%

Στον Πίνακα 4.4 με παρόμοιο τρόπο παρουσιάζονται τα ποσοστά μειώσεων που σημειώνονται στην μέση κατανάλωση της *capture power* σε κάθε κύκλωμα από κάθε μέθοδο σε σύγκριση με τη μέθοδο RF. Σε όλα σχεδόν τα κυκλώματα παρατηρείται μια μικρή μείωση κυρίως από την μέθοδο FA*, αλλά και από τις MFA, MFA+10, MFA+20. Επίσης, επιβεβαιώνεται ο ισχυρισμός ότι η FA μειονεκτεί έναντι των υπόλοιπων μεθόδων όσον αφορά την κατανάλωση της *capture power*. Ωστόσο, πρέπει να αναφερθεί ότι στα κυκλώματα ac97_ctrl και pci_bridge32 παρατηρείται μια ανεπαίσθητη αύξηση της ενέργειας από την προτεινόμενη μέθοδο MFA μαζί με τις MFA+10 και MFA+20. Το γεγονός αυτό οφείλεται στο ότι το κατώφλι L για τα σύνολα ελέγχου αυτών των δυο κυκλωμάτων ήταν πολύ μεγάλο και έτσι δεν χρησιμοποιήθηκε η μέθοδος *Preferred Fill*.

Πίνακας 4.4 Ποσοστό μείωσης capture power σε σύγκριση με την RF(με χρήση του εργαλείου PrimePower of Synopsys)

Capture Average Power Reduction Compared to Random						
Circuit	#sc	FA	FA*	MFA	MFA+10	MFA+20
s5378	2	4.42%	26.47%	29.11%	25.36%	25.47%
s9234	2	-5.38%	-0.86%	1.37%	1.72%	4.64%
s13207	5	23.19%	21.07%	15.40%	16.83%	16.26%
s15850	5	4.91%	10.65%	9.30%	4.43%	4.52%
s38417	10	22.61%	23.67%	20.30%	18.68%	15.77%
s38584	10	3.48%	14.98%	12.86%	11.19%	11.07%
ac97_ctrl	15	-0.10%	0.90%	-1.96%	-1.06%	-1.11%
mem_ctrl	8	43.95%	44.07%	39.63%	36.29%	29.31%
pci_bridge32	20	0.48%	1.12%	-2.21%	-2.50%	-4.63%
tv80	2	1.24%	2.01%	-2.65%	2.07%	2.32%
usb_funct	10	6.05%	13.87%	11.62%	12.56%	11.62%
ethernet	50	54.19%	55.67%	52.04%	43.37%	38.41%



4.1.2. Πειράματα για την ανίχνευση μη μοντελοποιημένων σφαλμάτων

Στην περίπτωση της εκτίμησης του ποσοστού ανίχνευσης σφαλμάτων, μελετώνται τα σφάλματα *transition* και *bridging*, εφαρμόζοντας στο υπό μελέτη κύκλωμα τα *test vectors* τύπου *stuck-at* που παράγονται από κάθε μέθοδο ξεχωριστά με τη βοήθεια του εργαλείου TetraMax. Για τα σφάλματα τύπου *transition* χρησιμοποιείται η τεχνική LOC, γνωστή ως *broadside scan*, προκειμένου να εφαρμόζονται στο υπό μελέτη κύκλωμα ζεύγη από *test vectors*. Επιπλέον, για την εκτίμηση των μεθόδων όσον αφορά την ανίχνευση σφαλμάτων τύπου *bridging* εισάγεται και η μετρική BCE^+ [20]. Η μετρική BCE^+ χρησιμοποιείται για την σύγκριση των μεθόδων και ο τύπος είναι ο ακόλουθος:

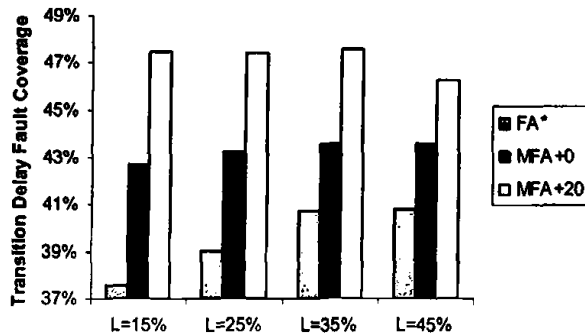
$$BCE^+ = \sum_{i=1}^n \frac{f_i^{sa-v}}{|F|} \cdot \left[\sum_{j=1}^{|S|} \frac{1}{|S|} \left(1 - (1 - p_{j,v}) \right) \right] \quad \text{Εξ. 4.1}$$

όπου $v=0, 1$. Η παράμετρος f_i^{sa-v} αναφέρεται στο πλήθος των σφαλμάτων τύπου *stuck-at-0* (για $v=0$) και *stuck-at-1* (για $v=1$), που έχουν ανιχνευθεί i φορές από τα διανύσματα ελέγχου (n είναι ο μέγιστος αριθμός ανίχνευσης για κάθε σφάλμα τύπου *stuck-at*). Οι παράμετροι $|S|$, $|F|$ και $p_{j,v}$ είναι αντίστοιχα το πλήθος των γραμμών του κυκλώματος, ο συνολικός αριθμός των σφαλμάτων τύπου *stuck-at* και η πιθανότητα το σήμα j να πάρει τη τιμή 0 (για $v=0$) και τη τιμή 1 (για $v=1$). Γενικά, η μετρική BCE^+ δεν είναι ακριβής στην εκτίμηση της πραγματικής κάλυψης σφαλμάτων τύπου *bridging* που επιτυγχάνει μια μέθοδος, αλλά είναι ικανοποιητική στην σύγκριση των μεθόδων. Η μέθοδος με υψηλή τιμή BCE^+ είναι η πιο αποτελεσματική στην ανίχνευση σφαλμάτων. Πιο αναλυτικά, επιλέχθηκαν τυχαία γύρω στα 100K ζεύγη γραμμών για κάθε κύκλωμα, τόσο από πύλες AND όσο και από OR, και προσομοιώθηκαν για κάθε ζεύγος τέσσερα σφάλματα τύπου *bridging*, λαμβάνοντας και τις 2 γραμμές κάθε ζεύγους ως “aggressors” και “victims”, καθώς και τις τιμές 0 και 1 στις γραμμές “aggressors”.

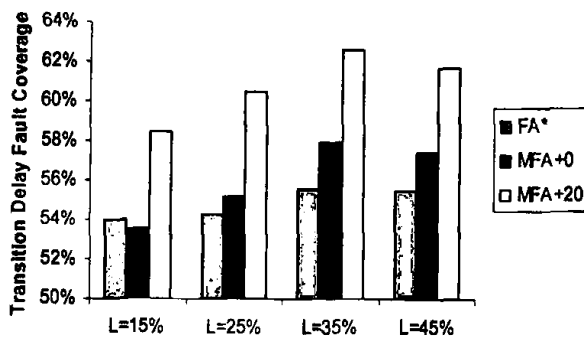
Πρώτα απ’ όλα, παρουσιάζονται τα ποσοστά ανίχνευσης σφαλμάτων τύπου *transition delay* που επιτυγχάνονται από τις μεθόδους FA*, MFA+0, MFA+20 για διάφορες τιμές του ορίου L της *capture power* μέσα στο εύρος [15%, 45%]. Στα Σχήματα 4.1 έως 4.6 παρουσιάζονται τα αποτελέσματα για τα κυκλώματα της κατηγορίας ISCAS. Όπως ήταν αναμενόμενο, για κάθε τιμή L σημειώνονται βελτιώσεις στην ανίχνευση



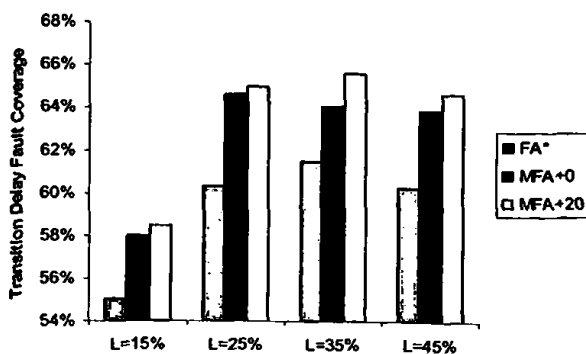
σφαλμάτων τύπου *transition*. Ωστόσο, πρέπει να τονιστεί ότι για $L = 15\%$ η τεχνική *Preferred Fill* δεν δύναται να δημιουργήσει αρκετά *test vectors* με *capture power* λιγότερη από το όριο για αρκετά *test cubes*, εξαιτίας της συμπιεσμένης μορφής των συνόλων δεδομένων.



Σχήμα 4.1 Κάλυψη σφαλμάτων τύπου transition delay για διάφορα L (s9234)

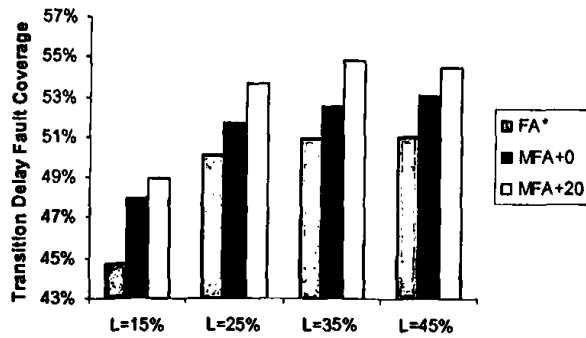


Σχήμα 4.2 Κάλυψη σφαλμάτων τύπου transition delay για διάφορα L (s5378)

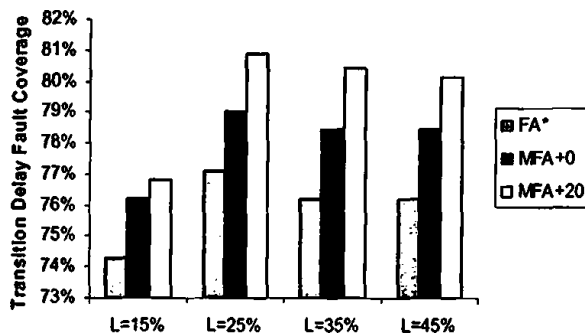


Σχήμα 4.3 Κάλυψη σφαλμάτων τύπου transition delay για διάφορα L (s13207)

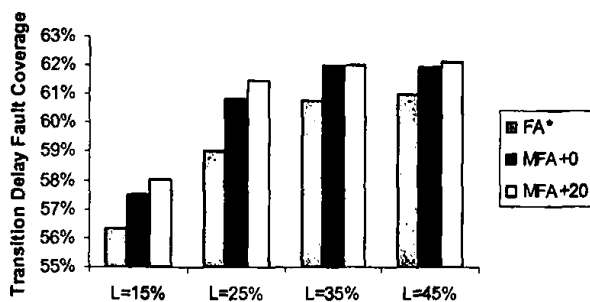




Σχήμα 4.4 Κάλυψη σφαλμάτων τύπου transition delay για διάφορα L (s15850)



Σχήμα 4.5 Κάλυψη σφαλμάτων τύπου transition delay για διάφορα L (s38417)

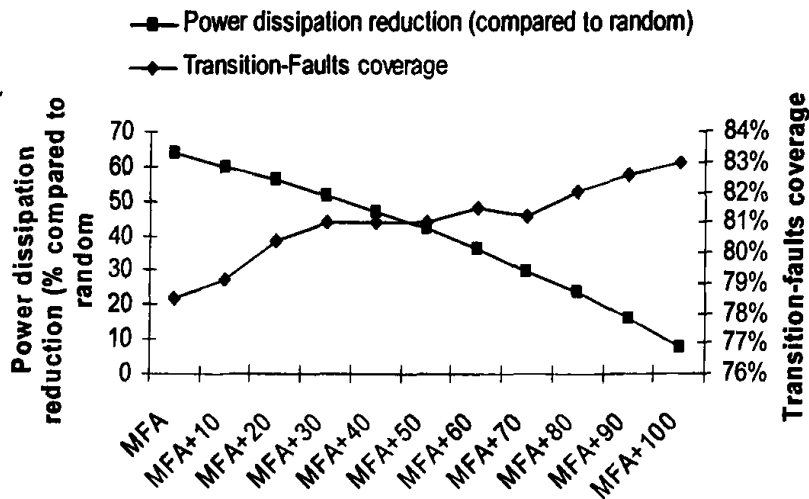


Σχήμα 4.6 Κάλυψη σφαλμάτων τύπου transition delay για διάφορα L (s38584)

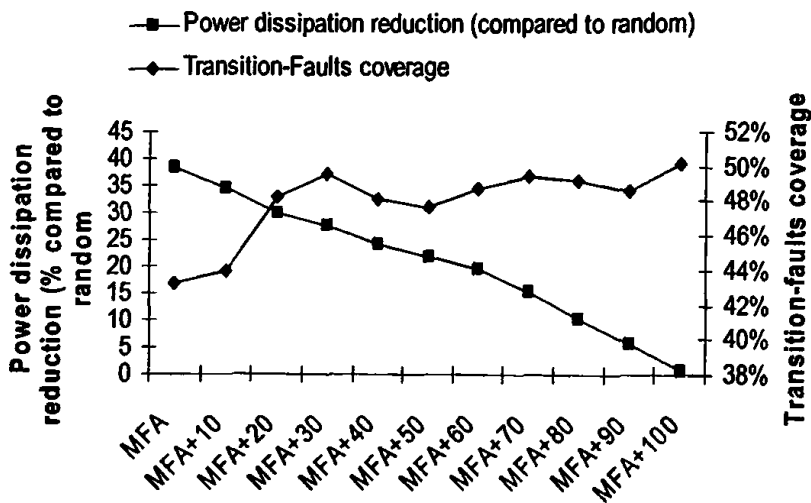
Στα Σχήματα 4.7 και 4.8 παρουσιάζονται τόσο τα ποσοστά μείωσης της κατανάλωσης της ολικής μέσης ενέργειας σε σχέση με την RF, όσο και τα ποσοστά ανίχνευσης σφαλμάτων τύπου *transition-delay* για διάφορες τιμές της παραμέτρου P .



Συγκεκριμένα, παρατίθενται οι συμπεριφορές των κυκλωμάτων s38417 και s9234 για τις μεθόδους MFA, MFA+10, MFA+20, ...,MFA+100. Μπορεί κανείς να παρατηρήσει ότι η αύξηση της παραμέτρου P επιφέρει αύξηση στην ανίχνευση σφαλμάτων, ενώ επιφέρει γραμμική μείωση στο ποσοστό μείωσης της ενέργειας σε σύγκριση με την *Random Fill*, η οποία μάλιστα προσεγγίζεται στο μηδέν για P=100.



Σχήμα 4.7 Κύκλωμα s38417



Σχήμα 4.8 Κύκλωμα s9234

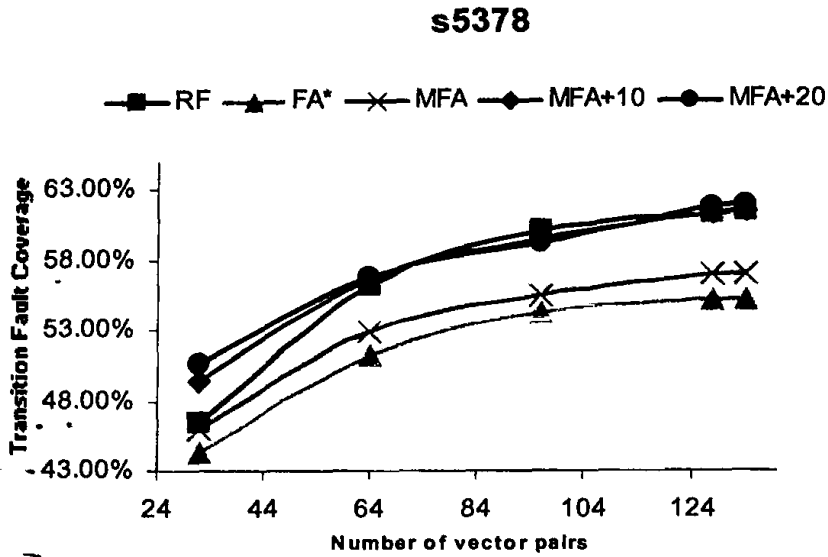


Στον Πίνακα 4.5 παρουσιάζονται τα συνολικά αποτελέσματα ανίχνευσης σφαλμάτων. Η πρώτη στήλη παρουσιάζει τα ονόματα των κυκλωμάτων και οι επόμενες έξι στήλες περιέχουν τα ποσοστά ανίχνευσης σφαλμάτων τύπου *transition-delay* για τις μεθόδους RF, FA, FA*, MFA, MFA+10 και MFA+20, αντίστοιχα. Παρατηρώντας λοιπόν, μπορεί κανείς να συμπεράνει ότι οι FA και FA* σημειώνουν χαμηλά ποσοστά, ενώ οι προτεινόμενες μέθοδοι σημειώνουν υψηλά ποσοστά τα οποία σε αρκετές περιπτώσεις υπερβαίνουν και αυτά της RF. Στα Σχήματα 4.9 έως 4.20 παρουσιάζονται τα γραφήματα ανίχνευσης σφαλμάτων τύπου *transition delay* που επιτυγχάνονται από κάθε μέθοδο για κάθε κύκλωμα της κατηγορίας ISCAS' 89 και IWLS' 05, αντίστοιχα. Σε κάθε διάγραμμα ο άξονας των x σχετίζεται με το πλήθος των ζευγών των *test vectors* που εφαρμόζονται, ενώ ο άξονας των y με το ποσοστό κάλυψης σφαλμάτων τύπου *transition*. Στα περισσότερα κυκλώματα, όπως κανείς μπορεί να παρατηρήσει, η προτεινόμενη μέθοδος, μαζί με τις δυο τροποποιημένες εκδοχές της, επιφέρουν μια αυξανόμενη κλίση που είναι καλύτερη από αυτή της μεθόδου RF και της FA*. Αυτό το γεγονός αποτελεί προτέρημα μιας και είναι δυνατόν τα περισσότερα ελαττώματα να ανιχνευθούν από τα πρώτα πιο αποτελεσματικά ζεύγη διανυσμάτων ελέγχου, μειώνοντας κατά αυτό τον τρόπο το χρόνο που καταναλώνεται κατά τη φάση ελέγχου.

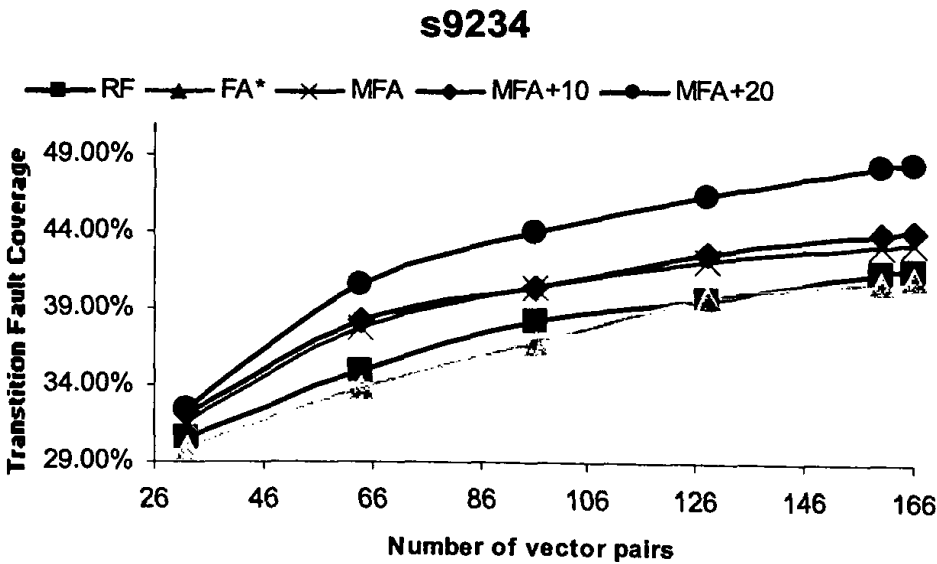
Πίνακας 4.5 Ποσοστά ανίχνευσης σφαλμάτων (%)

Circuit	Transition-Fault Coverage						Bridging-Fault Coverage											
							BCE						400K Random Faults Coverage					
	RF	FA	FA*	MFA	MFA+10	MFA+20	RF	FA	FA*	MFA	MFA+10	MFA+20	RF	FA	FA*	MFA	MFA+10	MFA+20
s5378	61.47	55.48	55.18	56.95	61.34	61.81	95.20	93.81	93.55	94.00	94.14	94.26	94.27	92.19	92.08	92.54	92.79	92.91
s9234	41.47	40.82	41.01	43.26	44.04	48.32	87.51	87.12	87.28	87.28	87.44	87.42	86.38	85.49	85.77	86.00	86.24	86.34
s13207	62.29	60.31	61.00	64.05	65.43	65.90	92.77	92.71	92.16	92.93	93.02	93.11	91.97	91.30	91.14	91.73	91.80	92.06
s15850	51.53	51.05	50.33	52.51	52.56	54.03	94.24	94.09	93.84	93.82	93.91	93.98	93.52	93.11	93.00	93.03	93.15	93.29
s38417	79.53	76.20	76.22	78.48	79.09	80.38	98.20	97.56	97.62	97.75	97.83	97.85	97.16	96.28	96.32	96.60	96.68	96.73
s38584	61.80	61.07	60.83	61.67	62.17	62.08	90.31	90.10	89.53	89.80	89.85	89.91	89.86	89.58	89.29	89.50	89.55	89.56
ac97_ctrl	42.62	42.53	42.48	44.31	44.39	44.96	94.54	94.11	94.09	94.24	94.29	94.38	96.94	96.66	96.65	96.72	96.76	96.83
mem_ctrl	40.96	36.97	36.76	38.18	38.92	40.26	62.32	59.72	59.59	60.09	60.34	60.70	74.56	72.54	72.43	72.84	72.99	73.24
pci_bridge32	64.40	61.74	61.82	65.38	66.68	67.50	95.75	95.41	95.46	95.62	95.67	95.71	96.61	96.33	96.36	96.48	96.52	96.54
tr80	53.47	51.10	51.10	53.26	57.48	58.14	91.49	91.00	90.99	91.11	91.15	91.12	89.30	88.38	88.38	88.87	88.87	88.90
usb_func1	63.94	63.14	62.46	64.41	63.98	64.27	93.74	93.21	93.34	93.39	93.44	93.49	95.14	94.77	94.86	94.96	95.03	95.04
ethernet	47.58	46.74	46.79	48.24	48.58	49.07	88.81	88.57	88.54	88.68	88.74	88.79	90.70	90.35	90.36	90.53	90.52	90.56



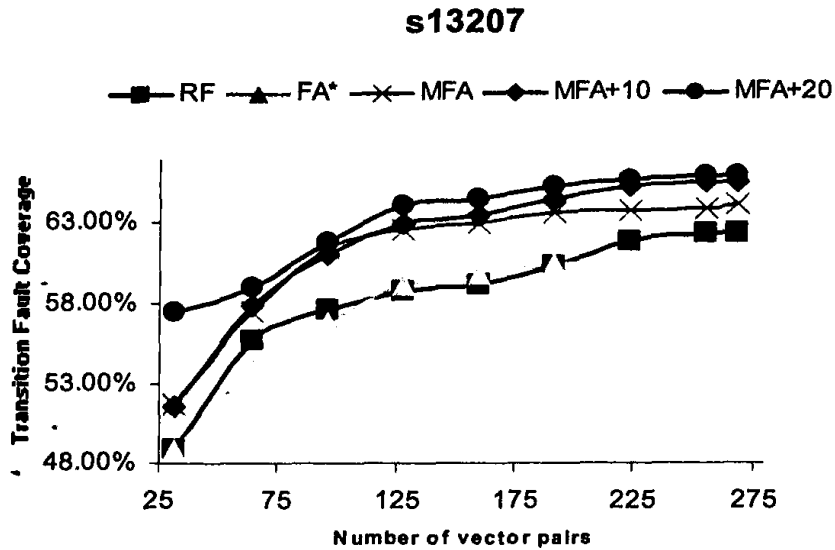


Σχήμα 4.9 Ποσοστό ανίχνευσης σφαλμάτων transition delay - ramp-up (s5378)

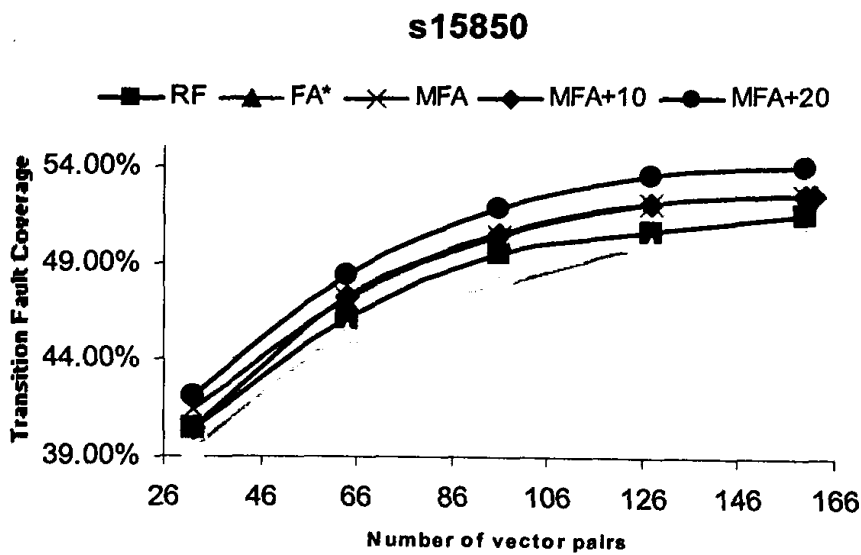


Σχήμα 4.10 Ποσοστό ανίχνευσης σφαλμάτων transition delay - ramp-up (s9234)



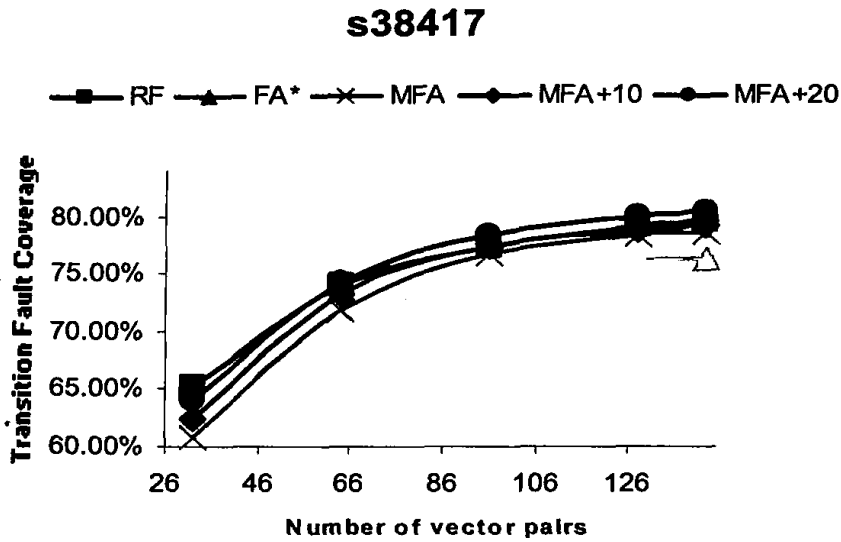


Σχήμα 4.11 Ποσοστό ανίχνευσης σφαλμάτων transition delay - ramp-up (s13207)

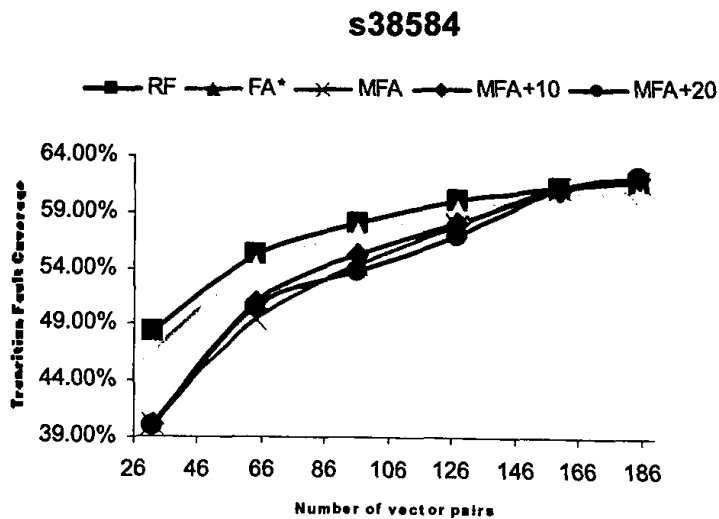


Σχήμα 4.12 Ποσοστό ανίχνευσης σφαλμάτων transition delay - ramp-up (s15850)



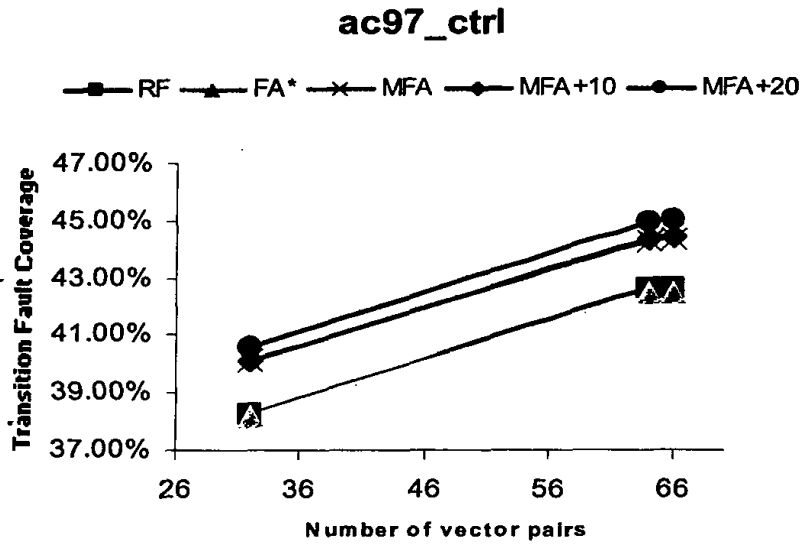


Σχήμα 4.13 Ποσοστό ανίχνευσης σφαλμάτων transition delay - ramp-up (s38417)

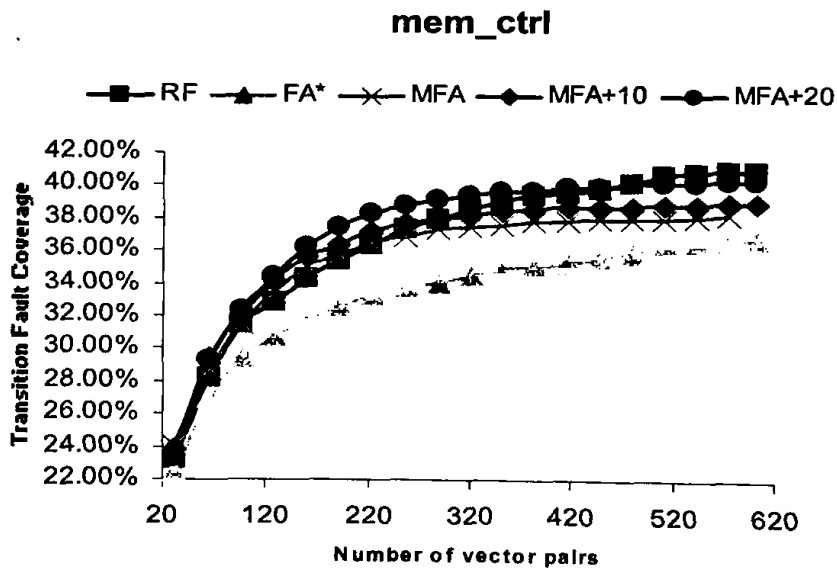


Σχήμα 4.14 Ποσοστό ανίχνευσης σφαλμάτων transition delay - ramp-up (s38584)



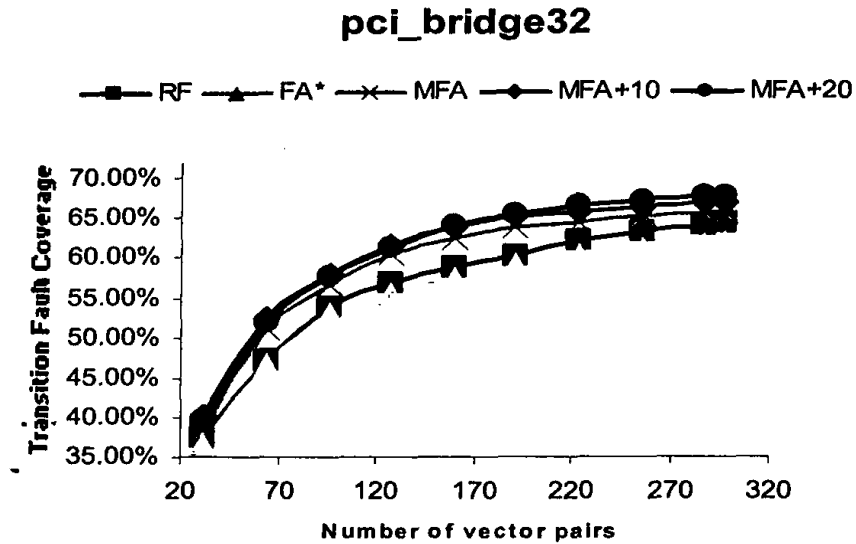


Σχήμα 4.15 Ποσοστό ανίχνευσης σφαλμάτων transition delay - ramp-up (ac97_ctrl)

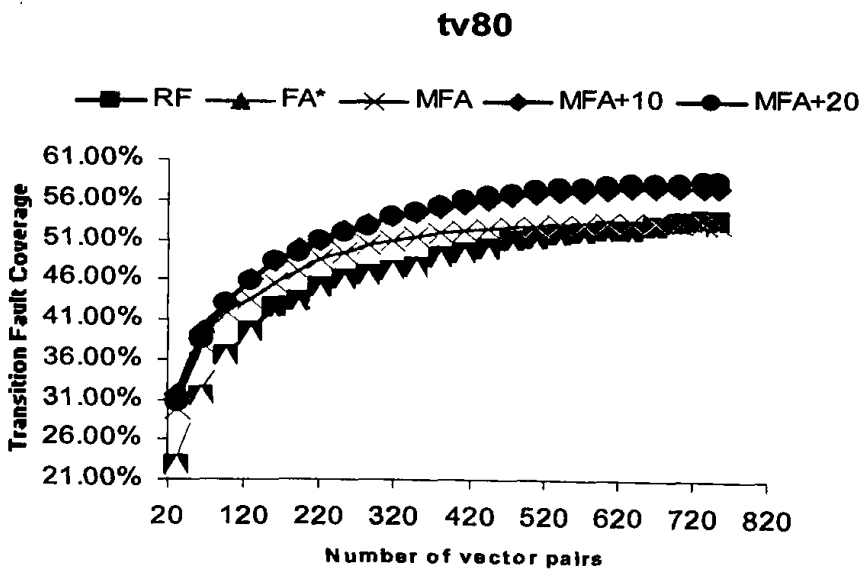


Σχήμα 4.16 Ποσοστό ανίχνευσης σφαλμάτων transition delay - ramp-up (mem_ctrl)



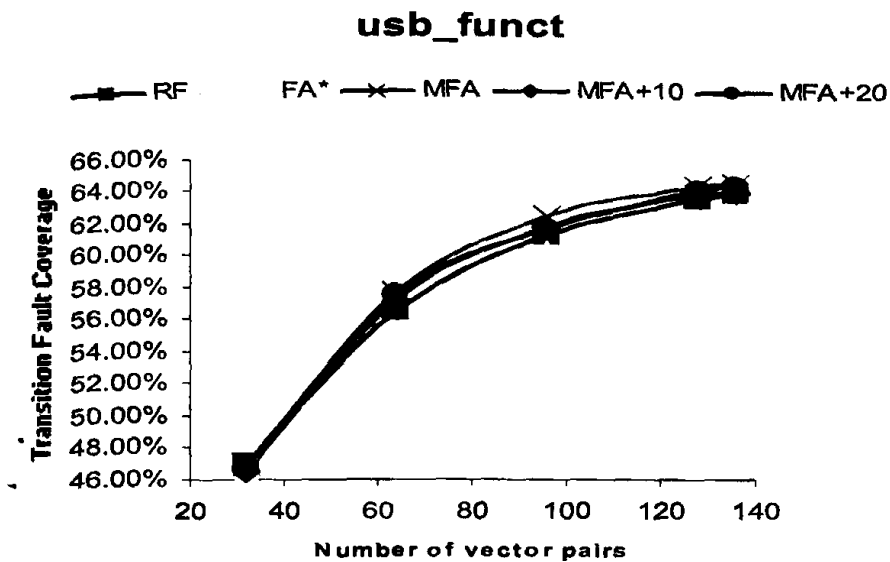


Σχήμα 4.17 Ποσοστό ανίχνευσης σφαλμάτων transition delay - ramp-up (pci_bridge32)

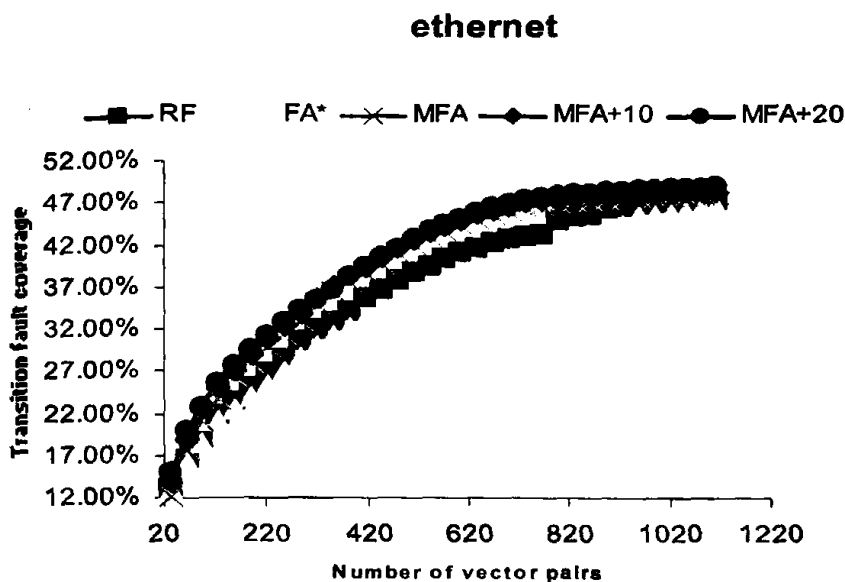


Σχήμα 4.18 Ποσοστό ανίχνευσης σφαλμάτων transition delay - ramp-up (tv80)





Σχήμα 4.19 Ποσοστό ανίχνευσης σφαλμάτων transition delay - ramp-up (usb_funct)



Σχήμα 4.20 Ποσοστό ανίχνευσης σφαλμάτων transition delay - ramp-up (Ethernet)

Στις επόμενες δώδεκα στήλες του Πίνακα 4.5 παρουσιάζονται αντίστοιχα τα ποσοστά ανίχνευσης σφαλμάτων τύπου *bridging*. Οι πρώτες έξι στήλες εμπεριέχουν τα αποτελέσματα BCE⁺ για κάθε μία από τις μεθόδους και οι επόμενες τα ποσοστά ανίχνευσης σφαλμάτων τύπου *bridging*. Η προτεινόμενη μέθοδος, μαζί με τις δυο



τροποποιημένες εκδοχές της, παρουσιάζει τέτοια κάλυψη σφαλμάτων τύπου *bridging*, καθώς και τιμές BCE^+ , που προσεγγίζουν τα αντίστοιχα αποτελέσματα της μεθόδου RF και είναι καλύτερα από αυτά των μεθόδων FA και FA*.

Συμπερασματικά, το κέρδος σε ενέργεια που επιφέρει όχι μόνο η προτεινόμενη μέθοδος MFA, αλλά και οι δυο τροποποιημένες εκδοχές της MFA+10, MFA+20, είναι σημαντικό. Από τη μια το μεγάλο κέρδος σε κατανάλωση ενέργειας και από την άλλη το ικανοποιητικό ποσοστό κάλυψης σφαλμάτων τύπου *transition* και *bridging* καθιστούν τις μεθόδους MFA, MFA+10 και MFA+20 ως ιδιαίτερα αξιοποιήσιμες.



ΚΕΦΑΛΑΙΟ 5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα εργασία προτάθηκαν δυο καινούργιες μέθοδοι, οι MFA και MFA+P, οι οποίες στοχεύουν στην χαμηλή κατανάλωση ενέργειας κατά τη διάρκεια ελέγχου ενός κυκλώματος και στην αύξηση του ποσοστού ανίχνευσης μη μοντελοποιημένων σφαλμάτων. Με τη διεξαγωγή των πειραμάτων στα μεγαλύτερα κυκλώματα των κατηγοριών ISCAS και IWLS, καθώς και με τη σύγκριση αυτών των μεθόδων με τις μεθόδους *Random Fill*, *Fill Adjacent* και την υβριδική FA* διαπιστώνεται εύλογα ότι ο στόχος έχει περατωθεί μιας και προσεγγίζουν τα θετικά χαρακτηριστικά των *Fill Adjacent* και *Random Fill*, αντίστοιχα. Συγκεκριμένα, σε σύγκριση με την καθιερωμένη μέθοδο *Fill Adjacent*, η οποία είναι ευρέως γνωστή για την χαμηλότερη κατανάλωση ενέργειας που εξασφαλίζει, η μέθοδος MFA την προσεγγίζει, όσον αφορά την κατανάλωση ενέργειας, και ταυτόχρονα αυξάνει σημαντικά το ποσοστό ανίχνευσης μη μοντελοποιημένων σφαλμάτων που επιφέρουν τα προκύπτοντα *test vectors*. Επίσης, υπογραμμίζεται ότι με την μέθοδο MFA εξασφαλίζεται η τήρηση των ορίων της μέγιστης ενέργειας κατά τη διάρκεια της λήψης των αποκρίσεων. Η μέθοδος MFA+P, με τη σειρά της, οδηγεί σε περαιτέρω βελτιώσεις στα ποσοστά ανίχνευσης μη μοντελοποιημένων σφαλμάτων, αυξάνοντας λίγο την μέση κατανάλωση ενέργειας.



ΑΝΑΦΟΡΕΣ

- [1] M. Butler, et. all, "Minimizing Power Consumption in Scan Testing: Pattern Generation and DFT Techniques", ITC 2004, pp.355-364.
- [2] A. Chandra, K. Chakrabarty, "Low-power scan testing and test data compression for system-on-a-chip", IEEE Trans. on CAD, Vol. 21, No 5, May 2002 pp.597-604
- [3] Chandra, A.; Kapur, R., "Bounded Adjacent Fill for Low Capture Power Scan Testing", 26th IEEE VTS, April 27-May 1 2008, pp. 131-138.
- [4] Bo-Hua Chen, et. all, "Response Inversion Scan Cell (RISC): A Peak Capture Power Reduction Technique", 16th ATS, 2007, pp. 425-432.
- [5] D. Czysz, G. Mrugalski, J. Rajski, Jerzy Tyszer "Low Power Embedded Deterministic Test", 25th IEEE VTS, 2007: 75-83.
- [6] D. Czysz, G. Mrugalski, J. Rajski, J. Tyszer "Low-Power Test Data Application in EDT Environment Through Decompressor Freeze", IEEE Trans. on CAD, vol. 27, no 7. pp: 1278-1290 (2008)
- [7] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A test vector inhibiting technique for low energy BIST design", in Proc. VLSI Test Symp., 1999, pp. 407-412.
- [8] P. Girard, "Low Power Testing of VLSI Circuits: Problems and Solutions", in Proc. ICQED, 2000, pp. 173-179.
- [9] X. Kabousianos, K. Chakrabarty, "Generation of Compact Test Sets with High Defect Coverage", DATE 2009, pp. 1130-1135.
- [10] S. Kajihara, K. Ishida, K. Miyase, "Test Vector Modification for Power Reduction during Scan Testing", Proc. VTS, pp. 160-165, 2002.
- [11] Ko H. F., Nicolici N., "Automated Scan Chain Division for Reducing Shift and Capture Power During Broadside At-Speed Test", IEEE Trans. on CAD, vol. 27, No 6, Page(s): 2092-2097.
- [12] J. Lee, and N. Touba, "Low Power Test Data Compression Based on LFSR Reseeding", in Proc. ICCD, 2004, pp. 180-185.



- [13]W. Li, S. Reddy, I. Pomeranz, "On Reducing Peak Current and Power During Test", IEEE Annual Symposium on VLSI, 2005.
- [14]J. Li, Q. Xu, Y. Hu and X. Li, "iFill: An Impact-Oriented X-Filling Method for Shift- and Capture-Power Reduction in At-Speed Scan-Based Testing", DATE March 10-14, 2008, p. 1184.
- [15]G. Mrugalski, J. Rajski, D. Czysz, J. Tyszer, "New Test Data Decompressor for Low Power Applications", Design Automation Conference 2007, pp. 539-544.
- [16]S. Remersaro, X. Lin, S. Reddy, I. Pomeranz, J. Rajski, "Scan Based Tests with Low Switching Activity", IEEE Design & Test of Computers, May-June 2007, pp. 268-275.
- [17]S. Remersaro, et. all, "Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs", ITC, 2006, pp. 1-10
- [18]R. Sankaralingam, R. R. Oruganti, and N. A. Touba, "Static compaction techniques to control scan vector power dissipation", in Proc. VLSI Test Symp., 2000, pp. 35-40.
- [19]J. Saxena, et.al. "A Case Study of IR-drop in Structured At-Speed Testing", Proc. ITC 2003, pp 1098 – 1104.
- [20]H. Tang et. al., "Defect Aware Test Patterns", in Proc. DATE Conf., pp. 450-455, 2005
- [21]Z. Wang and K. Chakrabarty, "Test-Quality/Cost Optimization Using Output-Deviation-Based Reordering of Test Patterns", IEEE Trans. CAD, vol. 27, No 2, pp. 352-365, 2008.
- [22]L.-C. Wang, P.R. Mercer, S.W. Kao and T.W. Williams, "On the Decline of Testing Efficiency as Fault Coverage Approaches 100%", in Proc. VTS, pp.74-83, 1995.
- [23]X.Wen, et. all, "A Capture-Safe Test Generation Scheme for At-Speed Scan Testing", 13th IEEE ETS, 25-29 May 2008, pp: 55-60.
- [24]X. Wen, et. all, "A New ATPG Method for Efficient Capture Power Reduction During Scan Testing", 24th IEEE VLSI Test Symposium, 2006.
- [25]X. Wen, et. all, "Low-capture-power test generation for scan-based at-speed testing", ITC, 2005, pp.10 pp.-1028.
- [26]X. Wen, et. all, "A Highly-Guided X-Filling Method for Effective Low-Capture-Power Scan Test Generation", ICCD 2006, pp. 251-258.
- [27]X. Wen, et. all, "Low Capture Switching Activity Test Generation for Reducing IR-Drop in At-Speed Scan Testing ", JETTA, vol. 24, No 4, 2008, pp 379-391.



[28]IWLS'05 bench. Circ., [HTTP://WWW.IWLS.ORG/IWLS2005/BENCHMARKS.HTML](http://www.iwls.org/iwls2005/benchmarks.html).



ΠΑΡΑΡΤΗΜΑ Α

Στο Παράρτημα Α παρουσιάζονται αναλυτικά όλα τα αποτελέσματα όλων των κυκλωμάτων που χρησιμοποιήθηκαν. Πιο αναλυτικά, στους Πίνακες Α.1 έως Α.12 παρουσιάζονται τα ποσοστά ανίχνευσης σφαλμάτων για τα μοντέλα σφαλμάτων *transition*, *bridging* και *stuck-at*, τα οποία έχουν υπολογιστεί χρησιμοποιώντας το εργαλείο TetraMax του Synopsys. Στην περίπτωση των σφαλμάτων τύπου *bridging* παρατίθεται μεταξύ άλλων και η μετρική BCE+. Επίσης, παρουσιάζονται η ολική μέση κατανάλωση ενέργειας (TOTAL AVERAGE POWER), η μέση κατανάλωση της ενέργειας capture (CAPTURE AVERAGE POWER), καθώς και η μέγιστη ενέργεια capture (PEAK POWER IN CAPTURE) σε Watt με τη βοήθεια του εργαλείου PrimePower του Synopsys. Παράλληλα, στην περίπτωση της μέγιστης ενέργειας capture παρέχεται και το αντίστοιχο χρονικό διάστημα. Επιπλέον, παρουσιάζεται η μέση ενέργεια που καταναλώνεται κατά την εισαγωγή με ολίσθηση των *test vectors* στις αλυσίδες scan (SCAN IN POWER) και η μέση ενέργεια που καταναλώνεται κατά την εξαγωγή με ολίσθηση των αντίστοιχων αποκρίσεων από τις αλυσίδες scan (SCAN OUT POWER). Για τον υπολογισμό της *scan-in power* προστέθηκαν τα αποτελέσματα της Εξίσωσης (2.1) για όλες τις αλυσίδες scan και για όλα τα *test vectors*, ενώ για την *scan-out power* προστέθηκαν τα αποτελέσματα της Εξίσωσης (3.3) για όλες τις αλυσίδες scan και για όλα τα *test vectors*.



Πίνακας Α.1 s5378

TRANSITION FAULTS					
patterns	RF	FA'	MFA	MFA+10	MFA+20
32	46.41%	44.38%	45.98%	49.45%	50.70%
64	56.15%	51.30%	52.94%	56.59%	56.70%
96	60.02%	54.16%	55.38%	59.49%	59.16%
128	61.20%	55.13%	56.90%	61.20%	61.70%
134	61.47%	55.18%	56.95%	61.34%	61.81%
BRIDGING FAULTS					
FAULT COVERAGE	94.27%	92.08%	92.54%	92.79%	92.91%
BCE+	95.1989%	93.5499%	93.9951%	94.1405%	94.2578%
STUCK FAULTS					
FAULT COVERAGE	99.66%	99.65%	99.66%	99.66%	99.66%
TOTAL AVERAGE POWER	6.649E-03	3.762E-03	3.505E-03	4.332E-03	4.716E-03
CAPTURE AVERAGE POWER	4.699E-03	3.455E-03	3.331E-03	3.507E-03	3.502E-03
PEAK POWER IN CAPTURE	8.415E-01	7.445E-01	6.921E-01	6.266E-01	5.957E-01
PEAK TIME FROM:	299759.114	163500.259	654000.259	168950.241	59950.250
INTERVAL TO:	299750.115	163500.260	654000.260	168950.242	59950.251
SCAN IN POWER	133.204902	72.548933	62.431494	64.493387	68.138247
SCAN OUT POWER	117.266972	64.148298	62.210192	92.321107	102.347558

Πίνακας Α.2 s9234

TRANSITION FAULTS					
patterns	RF	FA'	MFA	MFA+10	MFA+20
32	30.51%	29.87%	31.65%	32.07%	32.46%
64	34.90%	33.89%	37.79%	38.15%	40.51%
96	38.18%	36.86%	40.32%	40.42%	43.97%
128	39.55%	39.79%	42.15%	42.53%	46.33%
160	41.33%	40.65%	43.10%	43.86%	48.16%
166	41.47%	41.01%	43.26%	44.04%	48.32%
BRIDGING FAULTS					
FAULT COVERAGE	86.38%	85.77%	86.00%	86.24%	86.34%
BCE+	87.5131%	87.2768%	87.2796%	87.4406%	87.4172%
STUCK FAULTS					
FAULT COVERAGE	94.87%	94.87%	94.88%	94.88%	94.88%
TOTAL AVERAGE POWER	1.101E-02	6.864E-03	6.772E-03	7.184E-03	7.707E-03
CAPTURE AVERAGE POWER	6.968E-03	7.028E-03	6.872E-03	6.848E-03	6.644E-03
PEAK POWER IN CAPTURE	1.012E+00	9.123E-01	9.238E-01	9.227E-01	9.022E-01
PEAK TIME FROM:	94500.241	863100.398	1008000.398	617400.399	365400.241
INTERVAL TO:	94500.242	863100.399	1008000.399	617400.400	365400.242
SCAN IN POWER	164.438491	58.772220	56.000112	56.162593	59.504929
SCAN OUT POWER	163.395027	125.156243	124.892561	137.315770	151.121375



Πίνακας Α.3 s13207

TRANSITION FAULTS					
patterns	RF	FA*	MFA	MFA+10	MFA+20
32	48.92%	48.92%	51.75%	51.51%	57.41%
64	55.65%	53.84%	57.48%	57.83%	58.95%
96	57.51%	56.75%	61.39%	61.07%	61.75%
128	58.74%	59.05%	62.49%	62.93%	64.06%
160	59.12%	59.68%	62.88%	63.42%	64.51%
192	60.21%	60.31%	63.51%	64.47%	65.25%
224	61.73%	60.49%	63.73%	65.22%	65.55%
256	62.22%	60.64%	63.81%	65.42%	65.82%
269	62.29%	61.00%	64.05%	65.43%	65.90%
BRIDGING FAULTS					
FAULT COVERAGE	91.97%	91.14%	91.73%	91.80%	92.06%
BCE+	92.7737%	92.1568%	92.9283%	93.0211%	93.1057%
STUCK FAULTS					
FAULT COVERAGE	99.17%	99.17%	99.17%	99.17%	99.17%
TOTAL AVERAGE POWER	1.701E-02	8.570E-03	8.353E-03	8.810E-03	9.320E-03
CAPTURE AVERAGE POWER	1.033E-02	8.153E-03	8.739E-03	8.591E-03	8.650E-03
PEAK POWER IN CAPTURE	1.454E+00	1.374E+00	1.199E+00	1.226E+00	1.244E+00
PEAK TIME FROM:	71000.000	120750.241	191750.000	85200.249	78100.249
INTERVAL TO:	71000.001	120750.242	191750.001	85200.250	78100.250
SCAN IN POWER	672.434841	84.104933	69.784995	75.068859	95.269168
SCAN OUT POWER	668.736280	538.919733	539.675334	567.657348	595.104830

Πίνακας Α.4 s15850

TRANSITION FAULTS					
patterns	RF	FA*	MFA	MFA+10	MFA+20
32	40.46%	39.49%	41.46%	40.47%	42.08%
64	46.20%	45.37%	47.13%	47.31%	48.42%
96	49.50%	47.88%	50.41%	50.49%	51.86%
128	50.67%	49.83%	52.08%	52.08%	53.50%
160	51.53%	50.33%	52.51%	52.54%	54.03%
162				52.56%	
BRIDGING FAULTS					
FAULT COVERAGE	93.52%	93.00%	93.03%	93.15%	93.29%
BCE+	94.2443%	93.8387%	93.8171%	93.9050%	93.9795%
STUCK FAULTS					
FAULT COVERAGE	97.76%	97.76%	97.76%	97.76%	97.76%
TOTAL AVERAGE POWER	2.183E-02	9.498E-03	9.479E-03	1.050E-02	1.146E-02
CAPTURE AVERAGE POWER	8.881E-03	7.935E-03	8.055E-03	8.487E-03	8.479E-03
PEAK POWER IN CAPTURE	1.067E+00	8.583E-01	8.992E-01	8.672E-01	8.254E-01
PEAK TIME FROM:	12500.000	75000.249	662500.249	287500.249	31251.195
INTERVAL TO:	12500.001	75000.250	662500.250	287500.250	31251.196
SCAN IN POWER	405.415069	59.535108	56.212945	59.349725	69.077968
SCAN OUT POWER	386.633018	278.145762	280.657263	312.138882	328.554693



Πίνακας Α.5 s38417

TRANSITION FAULTS					
	RF	FA*	MFA	MFA+10	MFA+20
32	65.25%	62.57%	60.76%	62.41%	64.10%
64	74.11%	71.08%	71.74%	73.18%	74.16%
96	77.25%	74.49%	76.59%	77.33%	78.40%
128	79.02%	75.98%	78.24%	78.79%	80.00%
143	79.53%	76.22%	78.48%	79.09%	80.38%
BRIDGING FAULTS					
FAULT COVERAGE	97.16%	96.32%	96.60%	96.68%	96.73%
BCE*	98.1997%	97.6155%	97.7540%	97.8259%	97.8503%
STUCK FAULTS					
FAULT COVERAGE	99.75%	99.75%	99.75%	99.75%	99.75%
TOTAL AVERAGE POWER	5.524E-02	1.944E-02	1.977E-02	2.217E-02	2.423E-02
CAPTURE AVERAGE POWER	4.443E-02	3.391E-02	3.541E-02	3.613E-02	3.742E-02
PEAK POWER IN CAPTURE	3.233E+00	3.182E+00	3.267E+00	3.240E+00	3.195E+00
PEAK TIME FROM:	67600.445	67600.445	464750.445	371800.446	363350.445
INTERVAL TO:	67600.446	67600.446	464750.446	371800.447	363350.446
SCAN IN POWER	696.797155	118.357060	120.166873	125.641095	142.660334
SCAN OUT POWER	575.655413	307.447519	313.450316	362.027562	395.174753

Πίνακας Α.6 s38584

TRANSITION FAULTS					
patterns	RF	FA*	MFA	MFA+10	MFA+20
32	48.36%	46.67%	40.07%	40.14%	39.87%
64	55.28%	54.08%	49.47%	50.99%	50.65%
96	58.05%	57.18%	54.30%	55.27%	53.79%
128	60.02%	59.14%	57.89%	58.11%	56.95%
160	61.16%	60.46%	61.26%	61.09%	61.19%
185	61.80%	60.83%	61.67%	62.17%	62.08%
BRIDGING FAULTS					
FAULT COVERAGE	89.86%	89.29%	89.50%	89.55%	89.56%
BCE*	90.3126%	89.5307%	89.8008%	89.8486%	89.9055%
STUCK FAULTS					
FAULT COVERAGE	96.49%	96.49%	96.49%	96.49%	96.49%
TOTAL AVERAGE POWER	5.319E-02	2.305E-02	2.290E-02	2.548E-02	2.810E-02
CAPTURE AVERAGE POWER	2.456E-02	2.088E-02	2.140E-02	2.181E-02	2.184E-02
PEAK POWER IN CAPTURE	4.912E+00	3.253E+00	3.259E+00	3.293E+00	3.277E+00
PEAK TIME FROM:	14900.126	96850.132	327800.132	275650.132	245850.132
INTERVAL TO:	14900.127	96850.133	327800.133	275650.133	245850.133
SCAN IN POWER	927.089194	159.949678	154.292508	171.682176	207.527470
SCAN OUT POWER	921.397931	725.191594	728.415733	780.773470	816.772465



Πίνακας Α.7 ac97_ctrl

TRANSITION FAULTS					
patterns	RF	FA*	MFA	MFA+10	MFA+20
32	38.24%	38.22%	40.13%	40.16%	40.57%
64	42.56%	42.46%	44.23%	44.34%	44.89%
66	42.62%	42.48%	44.31%	44.39%	44.96%
BRIDGING FAULTS					
FAULT COVERAGE	96.94%	96.65%	96.72%	96.76%	96.83%
BCE+	94.5350%	94.0861%	94.2393%	94.2863%	94.3799%
STUCK FAULTS					
FAULT COVERAGE	99.90%	99.90%	99.90%	99.90%	99.90%
TOTAL AVERAGE POWER	6.355E-02	2.879E-02	2.944E-02	3.141E-02	3.435E-02
CAPTURE AVERAGE POWER	1.879E-02	1.862E-02	1.916E-02	1.899E-02	1.900E-02
PEAK POWER IN CAPTURE	3.680E+00	2.872E+00	3.686E+00	3.609E+00	3.627E+00
PEAK TIME FROM:	30600.562	130050.762	175950.562	91800.562	45900.562
INTERVAL TO:	30600.563	130050.763	175950.563	91800.563	45900.563
SCAN IN POWER	492.962375	73.724681	75.405582	79.372226	92.090841
SCAN OUT POWER	494.276844	330.389681	336.105342	362.928356	396.111690

Πίνακας Α.8 mem_ctrl

TRANSITION FAULTS					
patterns	RF	FA*	MFA	MFA+10	MFA+20
32	23.16%	22.49%	24.10%	23.87%	23.37%
64	28.16%	26.95%	28.75%	29.47%	29.26%
96	31.47%	29.42%	31.57%	32.17%	32.31%
128	32.89%	30.82%	33.50%	34.16%	34.40%
160	34.33%	31.72%	35.02%	35.60%	36.24%
192	35.45%	32.63%	35.77%	36.44%	37.50%
224	36.40%	33.34%	36.46%	37.26%	38.29%
256	37.36%	33.84%	36.92%	37.70%	38.81%
288	38.03%	34.10%	37.33%	38.02%	39.22%
320	38.59%	34.58%	37.53%	38.23%	39.44%
352	39.11%	34.93%	37.67%	38.48%	39.65%
384	39.42%	35.14%	37.80%	38.59%	39.73%
416	39.55%	35.35%	37.87%	38.69%	39.91%
448	39.76%	35.61%	37.96%	38.72%	39.97%
480	40.20%	35.94%	38.00%	38.75%	40.11%
512	40.68%	36.46%	38.03%	38.80%	40.16%
544	40.78%	36.56%	38.10%	38.86%	40.20%
576	40.85%	36.62%	38.18%	38.91%	40.24%
603	40.96%	36.76%		38.92%	40.26%
BRIDGING FAULTS					
FAULT COVERAGE	74.56%	72.43%	72.84%	72.99%	73.24%
BCE+	62.3191%	59.5858%	60.0855%	60.3362%	60.7021%
STUCK FAULTS					
FAULT COVERAGE	84.01%	84.02%	84.02%	84.02%	84.01%
TOTAL AVERAGE POWER	2.774E-02	7.378E-03	7.771E-03	9.538E-03	1.152E-02
CAPTURE AVERAGE POWER	8.665E-03	4.846E-03	5.231E-03	5.520E-03	6.125E-03
PEAK POWER IN CAPTURE	1.884E+00	1.571E+00	1.748E+00	1.646E+00	1.702E+00
PEAK TIME FROM:	60800.000	60800.000	83600.000	1147600.000	91299.000
INTERVAL TO:	60800.001	60800.001	83600.001	1147600.001	91200.001
SCAN IN POWER	2404.695645	102.450950	106.557156	127.712303	191.939274
SCAN OUT POWER	2316.321578	867.418931	932.523422	1184.838347	1450.899959



Πίνακας A.9 pci_bridge32

TRANSITION FAULTS					
patterns	RF	FA'	MFA	MFA+10	MFA+20
32	37.44%	36.65%	38.35%	40.31%	39.31%
64	47.08%	46.47%	51.15%	52.40%	51.86%
96	53.81%	52.84%	56.58%	57.82%	57.69%
128	56.90%	55.52%	60.51%	61.37%	61.21%
160	58.61%	57.36%	62.28%	63.72%	63.91%
192	60.17%	58.91%	63.65%	65.09%	65.49%
224	62.12%	60.24%	64.33%	65.72%	66.50%
256	63.30%	60.91%	65.05%	66.31%	67.13%
288	64.02%	61.61%	65.33%	66.66%	67.46%
298	64.40%	61.82%	65.38%	66.68%	67.50%
BRIDGING FAULTS					
FAULT COVERAGE	96.61%	96.36%	96.48%	96.52%	96.54%
BCE+	95.7454%	95.4588%	95.6237%	95.6668%	95.7084%
STUCK FAULTS					
FAULT COVERAGE	99.90%	99.90%	99.90%	99.90%	99.90%
TOTAL AVERAGE POWER	8.225E-02	2.610E-02	2.657E-02	3.094E-02	3.564E-02
CAPTURE AVERAGE POWER	2.479E-02	2.451E-02	2.534E-02	2.541E-02	2.594E-02
PEAK POWER IN CAPTURE	3.562E+00	3.170E+00	3.171E+00	3.159E+00	3.147E+00
PEAK TIME FROM:	124601.315	35600.000	1343901.062	258100.000	124600.000
INTERVAL TO:	124601.316	35600.001	1343901.063	258100.001	124600.001
SCAN IN POWER	2978.785951	179.084972	182.656896	214.104872	296.477754
SCAN OUT POWER	2974.603533	1398.476613	1429.043343	1711.712471	1968.598739

Πίνακας A.10 tv80

TRANSITION FAULTS					
patterns	RF	F'	MFA	MFA+10	MFA+20
32	22.68%	22.61%	29.77%	31.51%	30.62%
64	31.41%	31.24%	36.05%	39.37%	36.50%
96	36.50%	36.00%	41.97%	43.06%	43.00%
128	39.56%	38.79%	43.49%	45.94%	45.73%
160	42.46%	40.50%	45.60%	48.26%	48.27%
192	43.40%	42.07%	47.17%	49.98%	49.42%
224	45.05%	43.99%	48.62%	50.92%	50.82%
256	46.06%	45.33%	49.25%	52.15%	52.01%
288	46.55%	45.88%	50.29%	52.80%	52.70%
320	47.06%	46.83%	50.75%	53.86%	53.93%
352	47.57%	47.11%	51.35%	54.38%	54.43%
384	48.74%	48.01%	51.82%	55.00%	55.11%
416	49.35%	48.54%	52.13%	55.44%	55.86%
448	49.94%	48.96%	52.42%	55.94%	56.38%
480	50.86%	49.29%	52.63%	56.48%	56.81%
512	51.03%	49.64%	52.76%	56.66%	57.24%
544	51.51%	49.84%	52.90%	56.92%	57.40%
576	51.85%	49.99%	52.93%	57.04%	57.56%
608	52.13%	50.14%	53.06%	57.22%	57.77%
640	52.25%	50.20%	53.09%	57.33%	57.85%
672	52.63%	50.46%	53.19%	57.38%	57.90%
704	53.11%	50.74%	53.22%	57.45%	58.06%
736	53.46%	51.07%	53.22%	57.47%	58.14%
757	53.47%	51.10%	53.26%	57.48%	58.14%
BRIDGING FAULTS					
FAULT COVERAGE	89.30%	88.38%	88.87%	88.87%	88.90%
BCE+	91.4934%	90.9882%	91.1130%	91.1478%	91.1228%
STUCK FAULTS					
FAULT COVERAGE	97.64%	97.64%	97.64%	97.64%	97.64%
TOTAL AVERAGE POWER	2.856E-02	1.023E-02	1.053E-02	1.204E-02	1.325E-02
CAPTURE AVERAGE POWER	9.584E-03	9.391E-03	9.838E-03	9.385E-03	9.361E-03
PEAK POWER IN CAPTURE	1.168E+00	1.123E+00	1.123E+00	1.124E+00	1.185E+00
PEAK TIME FROM:	611000.550	611000.537	6226400.537	6222800.537	6260400.538
INTERVAL TO:	611000.551	611000.538	6226400.538	6222800.538	6260400.539
SCAN IN POWER	758.420459	69.376751	71.814473	77.601656	95.234118
SCAN OUT POWER	763.462947	386.099971	400.465388	471.139378	510.531938



Πίνακας A.11 usb_func1

TRANSITION FAULTS					
patterns	RF	FA*	MFA	MFA+10	MFA+20
32	47.05%	46.40%	46.85%	46.56%	46.64%
64	56.49%	55.10%	57.60%	57.06%	57.42%
96	61.23%	60.21%	62.37%	61.75%	61.59%
128	63.49%	62.21%	64.30%	63.82%	64.07%
136	63.94%	62.46%	64.41%	63.98%	64.27%
BRIDGING FAULTS					
FAULT COVERAGE	95.14%	94.86%	94.96%	95.03%	95.04%
BCE*	93.7370%	93.3353%	93.3919%	93.4409%	93.4916%
STUCK FAULTS					
FAULT COVERAGE	99.58%	99.59%	99.59%	99.59%	99.58%
TOTAL AVERAGE POWER	5.565E-02	3.225E-02	3.195E-02	3.324E-02	3.467E-02
CAPTURE AVERAGE POWER	2.659E-02	2.290E-02	2.350E-02	2.325E-02	2.350E-02
PEAK POWER IN CAPTURE	3.589E+00	2.490E+00	2.499E+00	2.507E+00	2.521E+00
PEAK TIME FROM:	253800.109	310200.109	601600.000	564000.109	470000.109
INTERVAL TO:	253800.110	310200.110	601600.001	564000.110	470000.110
SCAN IN POWER	678.541075	182.315113	178.806444	184.827024	198.766962
SCAN OUT POWER	668.103346	550.221026	546.808566	573.192970	594.926063

Πίνακας A.12 ethernet

TRANSITION FAULTS					
patterns	RF	FA*	MFA	MFA+10	MFA+20
32	13.28%	13.02%	12.26%	13.86%	15.07%
64	16.45%	16.25%	17.61%	19.07%	20.07%
96	18.63%	18.58%	21.10%	22.59%	22.86%
128	21.87%	21.24%	23.78%	25.14%	25.75%
160	23.49%	22.55%	26.25%	27.13%	27.80%
192	25.11%	24.04%	28.05%	28.93%	29.59%
224	26.85%	25.47%	29.81%	30.84%	31.39%
256	28.37%	27.14%	31.32%	32.51%	32.87%
288	30.65%	28.74%	32.85%	33.91%	34.34%
320	31.90%	30.02%	35.13%	35.87%	35.58%
352	33.04%	31.22%	36.43%	37.15%	36.77%
384	34.01%	32.25%	37.68%	38.27%	38.43%
416	35.82%	33.39%	38.77%	39.41%	39.60%
448	36.73%	34.34%	39.84%	40.43%	40.75%
480	37.85%	35.43%	41.21%	41.66%	41.71%
512	38.82%	36.40%	42.21%	42.66%	42.90%
544	39.59%	37.20%	43.14%	43.48%	43.79%
576	40.45%	38.02%	43.87%	44.20%	44.50%
608	41.15%	38.61%	44.54%	44.92%	45.29%
640	41.74%	39.15%	45.24%	45.67%	45.93%
672	42.30%	39.73%	45.67%	46.10%	46.57%
704	42.89%	40.19%	46.14%	46.61%	46.97%
736	43.07%	40.30%	46.51%	46.96%	47.45%
768	43.35%	40.45%	46.75%	47.20%	47.71%
800	44.88%	41.66%	46.96%	47.35%	47.96%
832	45.32%	42.02%	47.19%	47.55%	48.13%
864	45.70%	42.22%	47.35%	47.67%	48.24%
896	46.40%	43.56%	47.47%	47.81%	48.36%
928	46.67%	44.40%	47.61%	47.97%	48.53%
960	46.88%	45.58%	47.75%	48.07%	48.65%
992	47.01%	45.74%	47.84%	48.16%	48.77%
1024	47.11%	45.86%	47.99%	48.30%	48.85%
1056	47.25%	46.09%	48.11%	48.40%	48.93%
1088	47.43%	46.54%	48.20%	48.50%	49.02%
1113	47.58%	46.79%	48.24%	48.58%	49.07%
BRIDGING FAULTS					
FAULT COVERAGE	90.70%	90.36%	90.53%	90.52%	90.56%
BCE*	88.8128%	88.5388%	88.6846%	88.7362%	88.7930%
STUCK FAULTS					
FAULT COVERAGE	99.93%	99.93%	99.93%	99.93%	99.93%
TOTAL AVERAGE POWER	2.440E-01	2.392E-02	2.486E-02	4.676E-02	6.907E-02
CAPTURE AVERAGE POWER	3.587E-02	1.590E-02	1.720E-02	2.031E-02	2.209E-02
PEAK POWER IN CAPTURE	3.037E+01	3.065E+01	3.189E+01	3.183E+01	3.194E+01
PEAK TIME FROM:	1397551.622	3515251.644	3687251.644	2956251.644	1892051.644
INTERVAL TO:	1397551.623	3515251.624	3687251.645	2956251.645	1892051.645
SCAN IN POWER	27814.363155	308.256896	348.777518	617.669419	1410.143322
SCAN OUT POWER	27800.565615	4071.669793	4216.372812	8422.748457	12327.528118



ΠΑΡΑΡΤΗΜΑ Β

TetraMax

Το TetraMAX είναι ένα γρήγορο και αποτελεσματικό εργαλείο παραγωγής διανυσμάτων ελέγχου (test patterns) (ATPG). Μπορεί να παράγει *test patterns*, τα οποία να αυξάνουν την κάλυψη σφαλμάτων, βασισόμενο σε μικρό αριθμό διανυσμάτων για ένα ευρύ φάσμα σχεδιαζόμενων συστημάτων, των οποίων το μέγεθος ανέρχεται μέχρι και σε εκατομμύρια πυλών.

Το TetraMAX διαθέτει τις ακόλουθες ιδιότητες:

- Μπορεί να διαβάζει αρχεία, που περιγράφουν τα ηλεκτρονικά στοιχεία και τις μεταξύ τους διασυνδέσεις, γραμμένα σε Verilog, VHDL και EDIF, καθώς και πρωτόκολλα ελέγχου σε STIL. Αυτά τα αρχεία ονομάζονται netlists.
- Μπορεί να γράψει αρχεία από διανύσματα ελέγχου (test patterns) σε διάφορους τύπους, όπως WGL, STIL, Verilog, VHDL, Fujitsu TDL, TI TDL91, Toshiba TSTL2.
- Προσφέρει διάφορους τρόπους αυτόματης παραγωγής των διανυσμάτων ελέγχου (ATPG), όπως Basic-Scan, Fast-Sequential και Full-Sequential.
- Μπορεί να παράγει test patterns για τα εξής μοντέλα σφαλμάτων: stuck-at, IDDQ, transition delay, path delay και bridging.
- Υποστηρίζει διάφορα είδη flip-flops, εσωτερικούς και μη αποκωδικοποιημένους διαύλους τριών καταστάσεων, τύπους RAM και ROM, ελεγκτές.
- Παράγει και επαληθεύει διανύσματα ελέγχου που αποφεύγουν διακυμαινόμενες συνθήκες.
- Παρέχει αλληλεπιδραστική ανάλυση και εκσφαλμάτωση με το Graphical Schematic Viewer (GSV) για εύκολη ανάλυση διαφόρων σχεδιαστικών παραβιάσεων.



- Παρέχει συνδέσμους στους προσομοιωτές των Verilog και VHDL.
- Παρέχει ενσωματωμένο προσομοιωτή σφαλμάτων.
- Μπορεί να εκτελέσει απευθείας διάγνωση, επιτρέποντας έτσι τη γρήγορη αντιστοίχιση μιας αποτυχίας στο ελαττωματικό σημείο του σχεδιαζόμενου συστήματος.

Στο σχεδιαζόμενο σύστημα, προκειμένου να λειτουργήσει ορθά το εργαλείο TetraMAX, πρέπει να ορισθούν οι ακόλουθες περιπτώσεις:

- Οι θύρες ρολογιού.
- Οι ασύγχρονες θύρες set και reset.
- Οι θύρες εισόδου και εξόδου των αλυσίδων scan.
- Οι θύρες που θέτουν το σύστημα σε κατάσταση ελέγχου και οι αντίστοιχες καταστάσεις τους.
- Οι θύρες που ενεργοποιούν την ολίσθηση των αλυσίδων scan και οι αντίστοιχες καταστάσεις τους.
- Οι θύρες που ελέγχουν τη δικατευθυντήρια δραστηριότητα και οι αντίστοιχες καταστάσεις τους.

Στη συνέχεια, στο Σχήμα Β.1 παρουσιάζεται ένα σχεδιάγραμμα σχετικά με τα βήματα που ακολουθεί το εργαλείο TetraMAX για την παραγωγή των διανυσμάτων ελέγχου (*test patterns*), καθώς και παραδείγματα εντολών που περατώνουν αυτά τα βήματα.

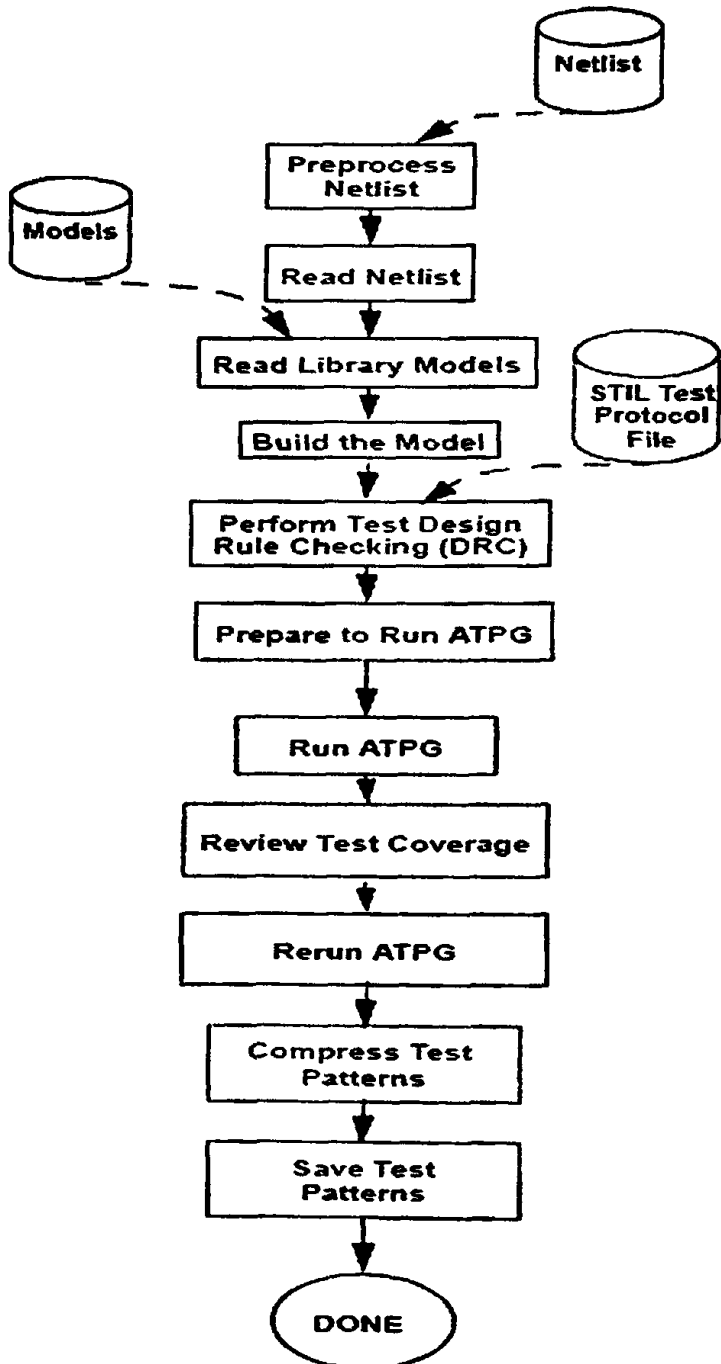
- Για την ανάγνωση ενός netlist: BUILD> read netlist filename.
- Για την ανάγνωση των βιβλιοθηκών που περιέχουν όλα τα μοντέλα του σχεδιαζόμενου συστήματος: BUILD> read netlist \$HOME/Model/*.v.
- Για τη δημιουργία του σχεδιαστικού μοντέλου ATPG: BUILD> run build_model top_module_name.
- Για τον έλεγχο του σχεδιαζόμενου συστήματος: BUILD> run drc. Στη φάση αυτή πραγματοποιούνται έλεγχοι για την εφαρμογή των σχεδιαστικών κανόνων. Πιο απλά, γίνονται οι ακόλουθοι έλεγχοι:
 - Αν οι εισοδοί και έξοδοι των αλυσίδων scan είναι λογικά συνδεδεμένοι.



- Αν μόνο πρωτεύουσες θύρες εισόδου ελέγχουν τα ρολόγια και τους ακροδέκτες (pins) set/reset.
 - Αν τα σήματα clocks/sets/resets είναι απενεργοποιημένα όταν υπάρχει εναλλαγή από την φυσιολογική λειτουργία στη λειτουργία ολίσθησης και αντίστροφα.
 - Αν διάφορες εσωτερικές γραμμές, οδηγούμενες από πολλές εξόδους, είναι υπερφορτωμένες.
- Για την προετοιμασία του ATPG πρέπει να αρχικοποιηθεί η λίστα σφαλμάτων, να επιλεγθεί η πηγή των patterns και να καθοριστεί ο τρόπος παραγωγής των patterns.
 - Για τη δημιουργία μιας λίστας σφαλμάτων, στην οποία να εμπεριέχονται όλα τα πιθανά σφάλματα στο σχεδιαστικό μοντέλο του ATPG, χρησιμοποιείται η ακόλουθη εντολή: TEST> add faults –all.
 - Για τη χρήση μιας ήδη υπάρχουσας λίστας χρησιμοποιείται η εξής εντολή: TEST> read faults fault_list_filename.
 - Το εργαλείο TetraMAX είναι προκαθορισμένο να παράγει εσωτερικά (internal) patterns. Οι πηγές των patterns μπορεί να είναι μία από τις εξής: εσωτερικά (Internal), τυχαία (Random) και εξωτερικά (External). Ωστόσο, μπορεί να τεθεί στο να χρησιμοποιεί εξωτερικά patterns χρησιμοποιώντας την ακόλουθη εντολή: TEST> set patterns external filename.
 - Για την ενεργοποίηση του ATPG: TEST> run atpg –random. Για να υπάρχει συμβιβασμός ανάμεσα στην συμπύκνωση των patterns και στον χρόνο της CPU αρκεί να προστεθεί η παράμετρος –auto_compression. Επίσης, δίνεται η δυνατότητα της N-detect n παραμέτρου, σύμφωνα με την οποία κάθε σφάλμα για να διαγραφεί από τη λίστα σφαλμάτων θα πρέπει να ανιχνευθεί n φορές.
 - Για τη δημιουργία αναφορών για τα σφάλματα χρησιμοποιείται η εξής εντολή: TEST> report summaries.
 - Για να αποθηκευθούν τα παραγόμενα test patterns, για παράδειγμα στη μορφή wgl, αρκεί να γραφτεί η εντολή: TEST> write patterns patterns.wgl –format wgl –replace.



- Δίνεται η δυνατότητα να αποθηκευτούν σε ένα αρχείο τα σφάλματα του σχεδιαζόμενου συστήματος, προκειμένου να αναλυθούν ή να χρησιμοποιηθούν σε καινούργια εκτέλεση του ATPG. Η σχετική εντολή είναι:
TEST> write faults filename -all -replace.



Σχήμα Β.1 Ροή του TetraMax



Το εργαλείο TetraMAX διαθέτει μια λίστα από πιθανά σφάλματα σε ένα σύστημα και αναθέτει κάθε σφάλμα σε μια κλάση σφαλμάτων ανάλογα με το αν μπορεί να ανιχνευθεί ή όχι. Γενικά, υπάρχουν πέντε κατηγορίες σφαλμάτων, καθεμία από τις οποίες περιέχει κλάσεις. Η σχετική οργάνωση των κλάσεων είναι ενδεικτικά η ακόλουθη:

- DT: detected
 - DR: detected robustly
 - DS: detected by simulation
 - DI: detected by implication
- PT: possibly detected
 - AP: ATPG untestable-possibly detected
 - NP: not analyzed-possibly detected
- UD: undetectable
 - UU: undetectable unused
 - UT: undetectable tied
 - UB: undetectable blocked
 - UR: undetectable redundant
- AU: ATPG untestable
 - AN: ATPG untestable- not detected
- ND: not detected
 - NC: not controlled
 - NO: not observed

Η διαδικασία προσομοίωσης σφαλμάτων δίνει τη δυνατότητα να καθοριστεί το ποσοστό ανίχνευσης σφαλμάτων από ένα εξωτερικά παραγόμενο test pattern. Γενικά, είναι απαραίτητα λειτουργικά test patterns, τα οποία δημιουργήθηκαν για τον έλεγχο του συστήματος και περιέχουν και την αναμενόμενη τιμή του συστήματος. Στο Σχήμα Β.2 παρατίθεται ένα σχεδιάγραμμα σχετικά με τα βήματα που εκτελούνται κατά την προσομοίωση σφαλμάτων. Τα test patterns, που επρόκειτο να χρησιμοποιηθούν στον προσομοιωτή σφαλμάτων ATPG του TetraMAX, θα πρέπει να ικανοποιούν τους ακόλουθους περιορισμούς:

- Να είναι αποδεκτά από το ATE (automated test equipment).

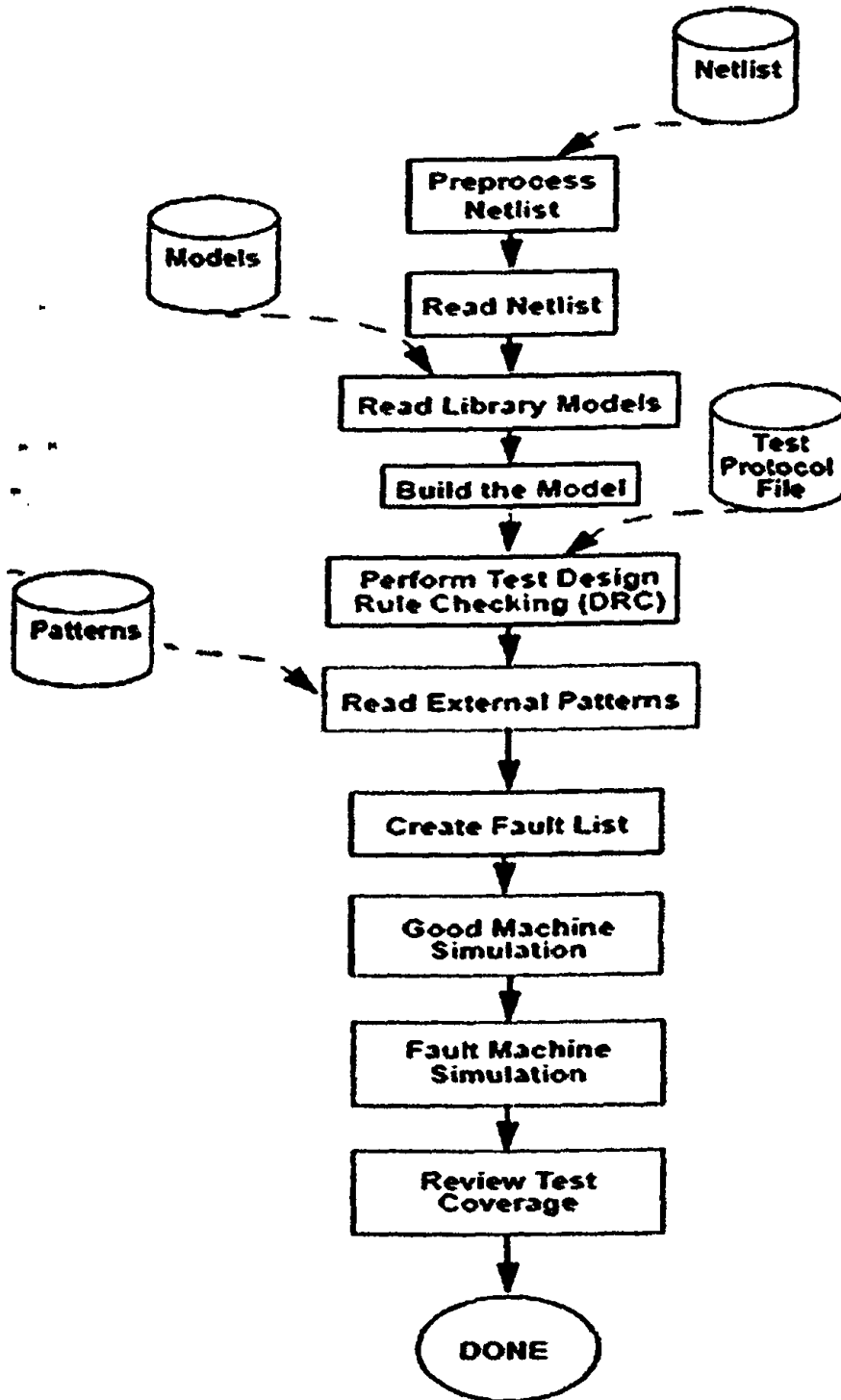


- Να είναι ανεπηρέαστα από το χρόνο.
- Το αρχείο τους να είναι σε αναγνωρίσιμη μορφή.

Παρόμοια με το ATPG, οι εντολές που χρησιμοποιούνται για να ολοκληρωθεί κάθε στάδιο της προσομοίωσης σφαλμάτων είναι οι ακόλουθες:

- Για την ανάγνωση ενός netlist και για την ανάγνωση των βιβλιοθηκών που περιέχουν όλα τα μοντέλα του σχεδιαζόμενου συστήματος έχουμε αντίστοιχα:
 - BUILD> read netlist filename
 - BUILD> read netlist \$HOME/Model/*.v.
- Για τη δημιουργία του σχεδιαστικού μοντέλου ATPG: BUILD> run build_model top_module_name.
- Για τον έλεγχο του σχεδιαζόμενου συστήματος, αν δηλαδή πληρεί τις βασικές προδιαγραφές έχουμε: BUILD> run drc.
- Για την ανάγνωση του αρχείου με τα patterns που θα εφαρμοστούν στην προσομοίωση σφαλμάτων έχουμε την εξής εντολή: TEST> set patterns external patterns.wgl.
- Για τη δημιουργία της λίστας σφαλμάτων, στην οποία να εμπεριέχονται όλα τα πιθανά σφάλματα στο σχεδιαστικό μοντέλο, χρησιμοποιείται η ακόλουθη εντολή: TEST> add faults -all.
- Στη φάση της προσομοίωσης της *good machine* χρησιμοποιούνται τα patterns πριν την προσομοίωση σφαλμάτων για να συγκριθούν οι αποκρίσεις που δίνει η προσομοίωση του TetraMAX με τις αναμενόμενες αποκρίσεις που εμπεριέχονται στο αρχείο με τα patterns. Αν αναφερθούν λάθη, τότε δεν έχει νόημα η εφαρμογή του επόμενου βήματος. Η σχετική εντολή είναι: TEST> run simulation -sequential.
- Για την ενεργοποίηση του προσομοιωτή σφαλμάτων και την λήψη των σχετικών αποτελεσμάτων, η αντίστοιχη εντολή είναι: TEST> run fault_sim -sequential -ndetects 1, όπου με την παράμετρο ndetects ορίζουμε ότι το κάθε σφάλμα αρκεί να ανιχνευθεί 1 φορά για να διαγραφεί από την ενεργή λίστα σφαλμάτων.
- Για την παραγωγή μιας περίληψης της προσομοίωσης σφαλμάτων αρκεί η εντολή: TEST> report summaries.





Σχήμα Β.2 Ροή Προσομοίωσης

Το μοντέλο σφαλμάτων *stuck-at* είναι το καθιερωμένο μοντέλο για την παραγωγή των test patterns. Ως σφάλμα θεωρείται όταν ένας κόμβος έχει κολλήσει είτε στη λογική τιμή 1 (*stuck-at-1*), είτε στη λογική τιμή 0 (*stuck-at-0*). Στόχος λοιπόν είναι να



προωθηθούν αυτές οι εσφαλμένες τιμές στις πρωτεύουσες εξόδους ή στα κύτταρα μιας αλυσίδας scan προκειμένου να παρατηρηθούν.

Το μοντέλο σφαλμάτων *transition delay* είναι παρόμοιο με το *stuck-at*. Πιο συγκεκριμένα, επιχειρεί την ανίχνευση κόμβων που έχουν την ιδιότητα *slow-to-rise* ή *slow-to-fall*. Το σφάλμα *slow-to-rise* σημαίνει ότι η μετάβαση από 0 σε 1 δεν παράγει τα σωστά αποτελέσματα σε μια συσκευή που λειτουργεί με τη μέγιστη ταχύτητα, ομοίως και το σφάλμα *slow-to-fall* που είναι η μετάβαση από το 1 στο 0.

Στο μοντέλο σφαλμάτων *bridging* η παρουσία ελαττώματος προκαλεί δυο ανεξάρτητες γραμμές σε μια συσκευή να συνδεθούν ηλεκτρικά εξαιτίας επιπλέον υλικού ή ελαττωματικής χαρακτηριστικής. Τέτοιου είδους σφάλματα γίνονται αντιληπτά αν μια από τις δυο γραμμές (ο επιδρομέας) προκαλεί την άλλη γραμμή (το θύμα) να πάρει λάθος τιμή, η οποία μπορεί να προωθηθεί σε ένα σημείο που μπορεί να παρατηρηθεί. Οι τύποι των σφαλμάτων *bridging* είναι είτε *ba0*, είτε *ba1*. Το σφάλμα *ba0* είναι ανιχνεύσιμο αν η τιμή 0 στην οποία έχει κολλήσει η γραμμή-θύμα (victim) ανιχνεύεται την ίδια στιγμή που η μη-εσφαλμένη τιμή της γραμμής-επιδρομέα (aggressor) είναι 0. Ομοίως, το σφάλμα *ba1* είναι ανιχνεύσιμο αν η τιμή 1 στην οποία έχει κολλήσει η γραμμή-θύμα (victim) ανιχνεύεται την ίδια στιγμή που η μη-εσφαλμένη τιμή της γραμμής-επιδρομέα (aggressor) είναι 1. Η προσθήκη των σφαλμάτων *bridging*, πέρα από τον καθορισμό του αντίστοιχου μοντέλου με την εντολή `set faults`, επιτυγχάνεται με την εντολή `add faults <-node_file <name>> [-bridge<0 | 1 | 01>][-aggressor_node <first | second | both>]`. Πιο αναλυτικά, με την παράμετρο `node_file` εισάγουμε ένα αρχείο που περιέχει τα ζεύγη των γραμμών (victim - aggressor), με την παράμετρο `bridge` ορίζουμε σε ποιες εσφαλμένες τιμές μπορεί να πάρει η γραμμή-θύμα και με την παράμετρο `aggressor_node` ορίζουμε σε κάθε ζεύγος γραμμών του αρχείου ποιες θα είναι οι επιδρομείς (aggressors).

Προκειμένου να αλλαχθεί το μοντέλο σφαλμάτων σε μια προσομοίωση, αρκεί να χρησιμοποιηθεί η εντολή `set faults -model [stuck | transition | bridging]`. Στη συνέχεια, στο Σχήμα Β.3 παρατίθενται παραδείγματα προσομοίωσης σφαλμάτων για τα μοντέλα σφαλμάτων *transition delay* και *bridging*, αντίστοιχα.



Για το μοντέλο σφαλμάτων transition delay:

1. set workspace size_line 500000
2. read netlist circuit.v
3. read netlist \$HOME/Model/*.v
4. run build_model TOP
5. run drc
6. set faults -model transition
7. add faults -all
8. set patterns external name.wgl
9. run fault_sim -ndetects 1
10. report summaries faults patterns

Για το μοντέλο σφαλμάτων bridging:

1. set workspace size_line 500000
2. read netlist circuit.v
3. read netlist \$HOME/Model/*.v
4. run build_model TOP
5. run drc
6. set faults -model bridging -pt_credit 0
7. add faults -node_file name -bridge 01 -
aggressor_node both
8. set patterns external name.wgl
9. run fault_sim -ndetects 1
10. report summaries faults patterns

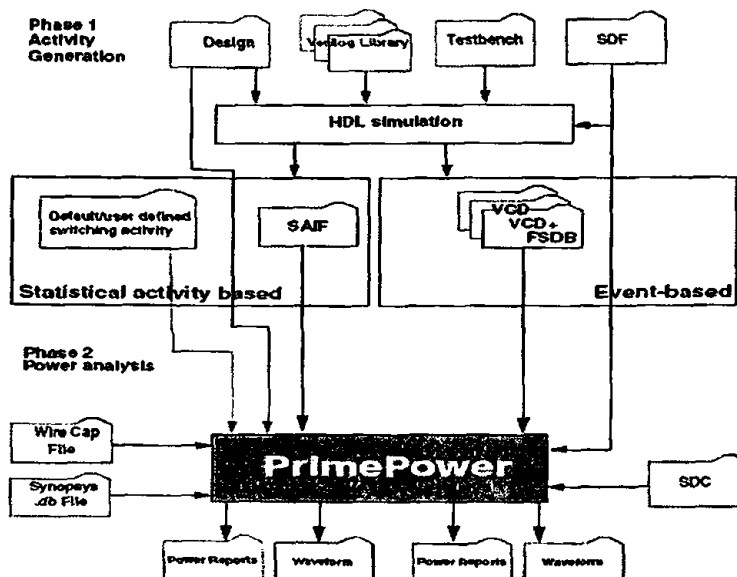
Σχήμα Β.3 Προσομοίωση για μοντέλα σφαλμάτων Transition και Bridging



ΠΑΡΑΡΤΗΜΑ Γ

PrimePower

Το PrimePower είναι ένα εργαλείο ανάλυσης ενέργειας σε επίπεδο πυλών, το οποίο αναλύει με απόλυτη ακρίβεια την ενέργεια που καταναλώνεται από τα διάφορα συστατικά μέρη ενός σχεδιαζόμενου συστήματος. Κατά τη λειτουργία του PrimePower πρώτα παράγεται η δραστηριότητα εναλλαγών μέσα στο σύστημα και έπειτα το προφίλ της ενέργειας που καταναλώθηκε. Αυτές οι δυο φάσεις παρουσιάζονται στο παρακάτω Σχήμα Γ.1.



Σχήμα Γ.1 PrimePower

Για την παραγωγή της δραστηριότητας εναλλαγών υπάρχουν διάφορες επιλογές, μερικές από τις οποίες είναι οι ακόλουθες:

- Αν δεν παρέχεται καμία πληροφορία σχετικά με τις εναλλαγές, που πρέπει να συμβούν στο σύστημα, τότε το εργαλείο PrimePower αναθέτει μια



προκαθορισμένη δραστηριότητα εναλλαγών στις πρωτεύουσες εισόδους του συστήματος και στη συνέχεια υπολογίζονται οι εναλλαγές που συμβαίνουν στο εσωτερικό του συστήματος.

- Οι χρήστες είναι δυνατόν να παρέχουν τουλάχιστον τον ρυθμό εναλλαγών του ρολογιού και άλλων κρίσιμων γραμμών για μια πιο ακριβή διάδοση και ανάλυση.
- Οι χρήστες μπορούν να ενεργοποιήσουν την προσομοίωση σε επίπεδο RTL και να εισάγουν την παραχθείσα πληροφορία των εναλλαγών RTL στο εργαλείο PrimePower.
- Οι χρήστες μπορούν να ενεργοποιήσουν την προσομοίωση σε επίπεδο RTL και να εισάγουν την παραχθείσα πληροφορία των εναλλαγών στο εργαλείο PrimePower με τη μορφή VCD, VCD+ ή FSDB.
- Μπορεί επίσης να χρησιμοποιηθεί και το αρχείο SAIF σε επίπεδο πυλών, το οποίο παράγεται από την προσομοίωση σε επίπεδο πυλών.
- Για μια ακριβή ανάλυση βασισόμενη σε γεγονότα, η δραστηριότητα εναλλαγών βασισόμενη στο επίπεδο πυλών μπορεί να παραχθεί στις μορφές αρχείων VCD/VCD+/FSDB. Τα συγκεκριμένα αρχεία παράγονται μέσω προσομοίωσης σε επίπεδο πυλών.

Το εργαλείο PrimePower εκτελεί να ακόλουθα βήματα για την ακριβή ανάλυση της ενέργειας που καταναλώνεται σε ένα σύστημα:

- Βασισμένο στην συνδεσιμότητα του συστήματος, στους διάφορους περιορισμούς και στη χωρητικότητα των γραμμών, το εργαλείο PrimePower καθορίζει τις εναλλαγές για όλους τους ακροδέκτες του συστήματος.
- Για την ανάλυση του μέσου όρου της ενέργειας, το εργαλείο PrimePower καθορίζει τις εναλλαγές εξαρτώμενες από την κατάσταση και τη διαδρομή σε όλους τους κόμβους του συστήματος. Αν κάποιες γραμμές δεν έχουν ενημερωθεί σχετικά, τότε αναλαμβάνει έργο η μηχανή διάδοσης.
- Το εργαλείο PrimePower έχει πρόσβαση και στις βιβλιοθήκες του Synopsys, που περιέχουν τους πίνακες ενέργειας και με βάση αυτά καθορίζει την κατανάλωση ενέργειας κάθε κυττάρου-φύλλου. Στην συνέχεια, τα



αποτελέσματα αυτά αθροίζονται προκειμένου να προκύψει η συνολική κατανάλωση ενέργειας του σχεδιαζόμενου συστήματος.

Το εργαλείο PrimePower διαθέτει δυο μεθόδους για την ανάλυση ενέργειας:

- *Ανάλυση βασισμένη σε γεγονότα:* Σε αυτή την περίπτωση, το εργαλείο χρησιμοποιεί έναν αλγόριθμο προσομοίωσης βασισμένο σε γεγονότα και υπολογίζει την κατανάλωση ενέργειας που επιτελείται σε κάθε γεγονός. Γενικά, παρέχεται μια ακριβής και λεπτομερέστατη όσον αφορά το χρόνο ανάλυση. Επίσης, μπορεί να δοθεί μια αναφορά τόσο για τη μέση ενέργεια, όσο και την κορυφαία ενέργεια του συστήματος.
- *Ανάλυση βασισμένη σε στατιστικά δεδομένα της δραστηριότητας:* Σε αυτή την περίπτωση, το εργαλείο χρησιμοποιεί έναν αλγόριθμο πιθανολογικής και στατιστικής εκτίμησης για τον υπολογισμό της μέσης ενέργειας και την κατασκευή της κυματομορφής του μέσου κύκλου ενέργειας. Η είσοδος είναι ένα αρχείο SAIF ή μια δραστηριότητα εναλλαγών ορισμένη από τον χρήστη.

Γενικά, οι διάφοροι τύποι ενέργειας που καταναλώνονται από ένα κύκλωμα είναι οι ακόλουθοι:

- Στατική.
- Δυναμική.

Ως *στατική* ορίζεται η ενέργεια που καταναλώνεται από μια πύλη όταν αυτή δεν εναλλάσσει τιμές, δηλαδή όταν είναι ανενεργή ή στατική. Αυτό οφείλεται στην ύπαρξη του χαμηλού κατωφλίου της τάσης που αποτρέπει την πύλη να απενεργοποιηθεί εντελώς. Δηλαδή, η πύλη διαρρέεται από μια ελάχιστη ποσότητα ρεύματος και γι' αυτό η στατική ενέργεια αποκαλείται και ενέργεια διαρροής (leakage).

Ως *δυναμική* ορίζεται η ενέργεια που καταναλώνεται όταν το κύκλωμα είναι ενεργό. Το κύκλωμα είναι ενεργό οποτεδήποτε η τάση (voltage) στη γραμμή αλλάζει. Επειδή η τάση σε μια γραμμή εισόδου μπορεί να αλλάξει χωρίς να επιφέρει αλλαγή και στην αντίστοιχη έξοδο, μπορεί να θεωρηθεί ότι καταναλώνεται δυναμική ενέργεια ακόμα



και αν η λογική τιμή της γραμμής εξόδου δεν αλλάζει. Η δυναμική ενέργεια αναλύεται στους εξής δυο τύπους:

- Ενέργεια εναλλαγών (switching).
- Εσωτερική ενέργεια.

Ως *ενέργεια εναλλαγών* σε ένα κύτταρο ορίζεται η ενέργεια που καταναλώνεται λόγω φόρτισης και αποφόρτισης της εξόδου του κυττάρου. Το συνολικό φορτίο ρεύματος στην έξοδο ενός κυττάρου αποτελείται από το άθροισμα των φορτίων ρεύματος της γραμμής και της αντίστοιχης πύλης. Καθώς, η φόρτιση και αποφόρτιση είναι αποτέλεσμα των εναλλαγών στην έξοδο του κυττάρου, η ενέργεια εναλλαγών αυξάνεται με την αύξηση των λογικών εναλλαγών. Συνεπώς, η ενέργεια εναλλαγών ενός κυττάρου είναι συνάρτηση του συνολικού φορτίου ρεύματος στην έξοδο του κυττάρου και του ρυθμού των λογικών εναλλαγών.

Ως *εσωτερική ενέργεια* ορίζεται η ενέργεια που καταναλώνεται εντός των ορίων του κυττάρου. Κατά την εναλλαγή το κύκλωμα καταναλώνει εσωτερική ενέργεια φορτίζοντας και αποφορτίζοντας το εσωτερικό του κυττάρου. Επίσης, εσωτερική ενέργεια έχουμε και στην περίπτωση κατανάλωσης ενέργειας ανάμεσα στα τρανζίστορς P και N μιας πύλης.

Η ανάλυση ενέργειας βασισμένη στη στατιστική δραστηριότητα στο εργαλείο PrimePower εκτελείται σε πολλαπλά στάδια:

- Ελέγχεται η δραστηριότητα εναλλαγών σε όλες τις γραμμές του συστήματος. Η δραστηριότητα εναλλαγών μπορεί να ληφθεί από μια απλή προσομοίωση ή με χρήση των SAIF αρχείων και σχετικών εντολών.
- Το εργαλείο PrimePower διαδίδει αυτή τη δραστηριότητα εναλλαγών σε όλες τις γραμμές και αναθέτει όπου είναι δυνατόν προκαθορισμένες τιμές μέχρι να ορισθούν οι ρυθμοί εναλλαγών σε κάθε γραμμή.
- Βασισμένο στη δραστηριότητα των εισόδων και εξόδων για κάθε κύτταρο, το PrimePower υπολογίζει τη διαφορά Boolean για κάθε τόξο ενέργειας και καθορίζει τη συνεισφορά κάθε τόξου στην ολική ενέργεια ενός κυττάρου.



Η ενέργεια που καταναλώνει ένα σύστημα εξαρτάται από τη δραστηριότητα εναλλαγών που εντοπίζεται στους ακροδέκτες των κυττάρων και στις γραμμές. Η σχετική πληροφορία των εναλλαγών διαδίδεται στις γραμμές, στις θύρες, στους ακροδέκτες και στα κύτταρα και με τη χρήση της εντολής `calculate_power` πραγματοποιείται η σχετική ανάλυση. Στη συνέχεια, παρατίθενται οι τύποι των δραστηριοτήτων εναλλαγών που μπορούν να διαδοθούν σε ένα σύστημα.

- Η απλή δραστηριότητα εναλλαγών αποτελείται από μια στατική πιθανότητα και έναν ρυθμό εναλλαγών (toggle rate). Ως στατική καλείται η πιθανότητα ένα αντικείμενο να έχει λογική τιμή 1 και ως ρυθμό εναλλαγών εννοούμε τον ρυθμό με τον οποίο ένα αντικείμενο εναλλάσσεται ανάμεσα στις λογικές τιμές 0 και 1.
- Οι ρυθμοί εναλλαγών (toggle rates) στους ακροδέκτες εισόδου των κυττάρων, οι οποίοι εξαρτώνται από την κατάσταση.
- Οι ρυθμοί εναλλαγών (toggle rates) στους ακροδέκτες εισόδου των κυττάρων, οι οποίοι εξαρτώνται όχι μόνο από την κατάσταση, αλλά και από το μονοπάτι.
- Η στατική πιθανότητα στα κύτταρα, η οποία εξαρτάται από την κατάσταση.

Η δραστηριότητα των εναλλαγών μπορεί να διαδοθεί στο σύστημα με τους εξής τρόπους:

- Σε ανεξάρτητα σχεδιαζόμενα αντικείμενα μπορεί να εφαρμοστεί η εντολή `set_switching_activity`.
- Παράγοντας αρχεία SAIF (Switching Activity Interchange Format) χρησιμοποιώντας προσομοίωση είτε επιπέδου πυλών είτε RTL και διαβάζοντας στη συνέχεια αυτά τα αρχεία με τις εντολές `read_saif` ή `merge_saif`.
- Χρησιμοποιώντας προκαθορισμένη δραστηριότητα εναλλαγών.

Χρησιμοποιώντας την διεπαφή PLI 1.0 (Programmable Language Interface 1.0), που υποστηρίζει το Cadence Verilog-XL ή VCS προσομοιωτή. Μέσω της διεπαφής PLI, μπορεί κανείς να διαβάσει ή να γράψει απευθείας αρχεία SAIF forward και backward κατά τη διάρκεια της προσομοίωσης επιπέδου πυλών ή RTL. Η διεπαφή του DesignPower στο Verilog PLI αναγνωρίζει τις εντολές toggle που εισάγονται στο



αρχείο testbench. Κατά τη φάση της προσομοίωσης σε επίπεδο πυλών, μπορεί να χρησιμοποιηθεί η διεπαφή στο PLI για να:

- Διαβαστούν τα αρχεία SAIF forward στην προσομοίωση σε επίπεδο πυλών. Αυτής της κατηγορίας τα αρχεία περιέχουν πληροφορίες σχετικά με εξαρτήσεις μονοπατιών και καταστάσεων που έχουν τα κύτταρα των βιβλιοθηκών.
- Γραφούν τα αρχεία SAIF backward από την προσομοίωση σε επίπεδο πυλών. Αυτής της κατηγορίας τα αρχεία περιέχουν πληροφορίες για τις εναλλαγές μερικών ή όλων των συνδέσεων και τις εναλλαγές των στοιχείων του κυκλώματος μαζί με τις εξαρτήσεις τους.

Η προσομοίωση Verilog ελέγχει όλα τα αντικείμενα του ιεραρχημένου, σχεδιαζόμενου συστήματος και μέσω συγκεκριμένων εντολών δίνεται η δυνατότητα ελέγχου ορισμένων πυλών ή πυλών της υψηλότερης ιεραρχίας.

Για την αναφορά των καθυστερήσεων των πυλών απαιτείται η εξαγωγή ενός αρχείου SDF (Standard Delay Format) back-annotation, το οποίο είναι απαραίτητο για προσομοίωση συμπεριφοράς. Το αρχείο SDF γράφεται με την παρακάτω εντολή:

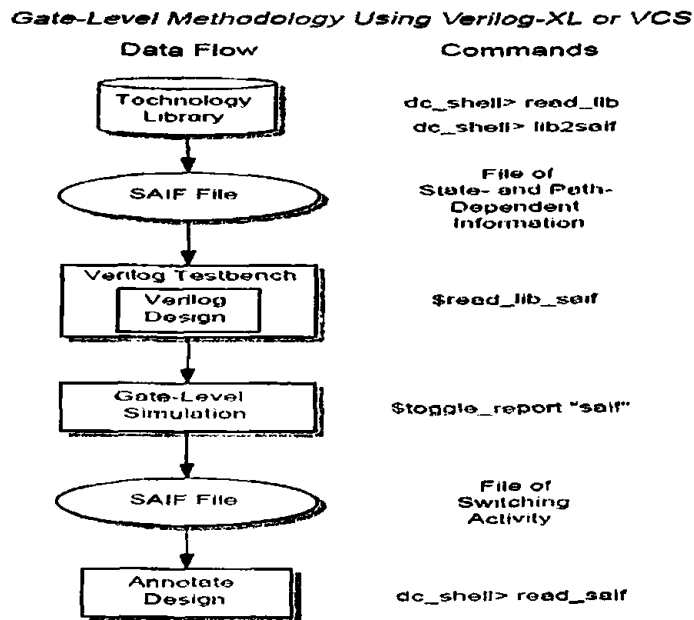
- `write_sdf -version 1.0 s9234_RF.sdf`

Αν εφαρμοστεί η προσομοίωση σε επίπεδο πυλών για την παραγωγή της δραστηριότητας εναλλαγών (SAIF), τότε τα σχετικά βήματα είναι τα ακόλουθα:

- Δημιουργία ενός αρχείου SAIF forward-annotation.
- Ενσωμάτωση του αρχείου SAIF forward-annotation στην προσομοίωση.
- Η προσομοίωση θα κατασκευάσει ένα αρχείο SAIF back-annotation.
- Ανάγνωση του αρχείου SAIF back-annotation για να προστεθεί στο σχέδιο.

Στο Σχήμα Γ.2 παρουσιάζεται η διαδικασία δημιουργίας της δραστηριότητας εναλλαγών μέσω της προσομοίωσης Verilog σε επίπεδο πυλών.





Σχήμα Γ.2 Η διαδικασία δημιουργίας δραστηριότητας εναλλαγών

Με την εντολή `lib2saif` δημιουργείται το αρχείο SAIF forward-annotation, το οποίο εμπεριέχει πληροφορίες από την βιβλιοθήκη σχετικά με τα κύτταρα, που έχουν εξαρτήσεις μονοπατιών και καταστάσεων, και είναι κατάλληλο μόνο για προσομοιώσεις σε επίπεδο πυλών. Γενικά, χάρη στον προσομοιωτή VCS ή Verilog-XL, μπορούν να προστεθούν στο Verilog testbench εντολές εναλλαγών (`toggle`), προκειμένου να διαπιστωθεί η διαδικασία εναλλαγών σε όλη τη διάρκεια της προσομοίωσης. Χρησιμοποιώντας αυτές τις εντολές, μπορεί κανείς να:

- Καθορίσει ένα μπλόκ ή υπομπλόκ για καταμέτρηση των εναλλαγών.
- Καθορίσει ανεξάρτητα αντικείμενα για καταμέτρηση των εναλλαγών.
- Αρχίσει και να τερματίσει την καταμέτρηση εναλλαγών για τα αντικείμενα του κυκλώματος ή μπλόκ.
- Ορίσει συγκεκριμένες χρονικές περιόδους για την καταμέτρηση των εναλλαγών κατά τη διάρκεια της προσομοίωσης.
- Καταγράψει τη δραστηριότητα εναλλαγών σε ένα αρχείο εξόδου.
- Ορίσει τον τύπο του αρχείου εξόδου.

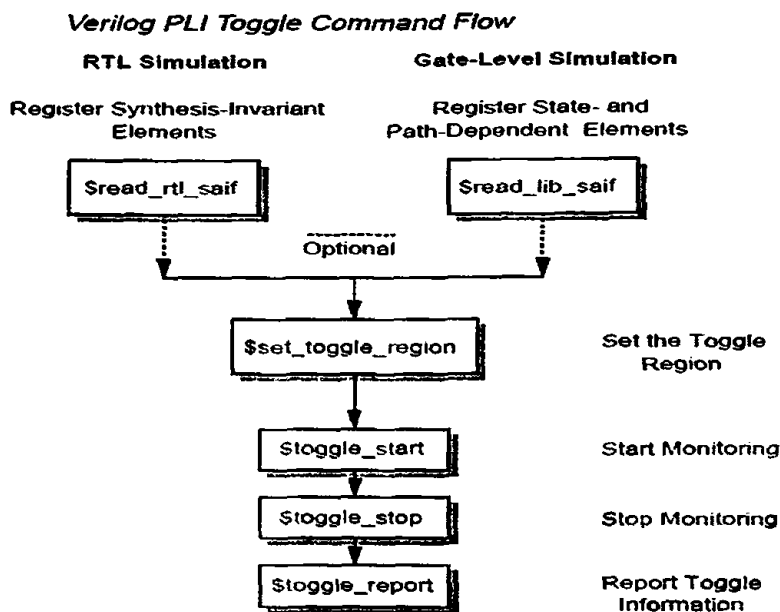
Οι εντολές εναλλαγών PLI είναι ενδεικτικά οι ακόλουθες:

- `$read_rtl_saif`



- \$read_lib_saif
- \$set_gate_level_monitoring
- \$set_toggle_region
- \$toggle_start
- \$toggle_stop
- \$toggle_report
- \$toggle_reset

Παρακάτω παρατίθεται ενδεικτικά ο τρόπος με τον οποίον εισάγονται οι εντολές εναλλαγών σε ένα αρχείο testbench. Συγκεκριμένα στο Σχήμα Γ.3 διαφαίνεται η ροή των εντολών εναλλαγών (toggles).



Σχήμα Γ.3 Εντολές toggle

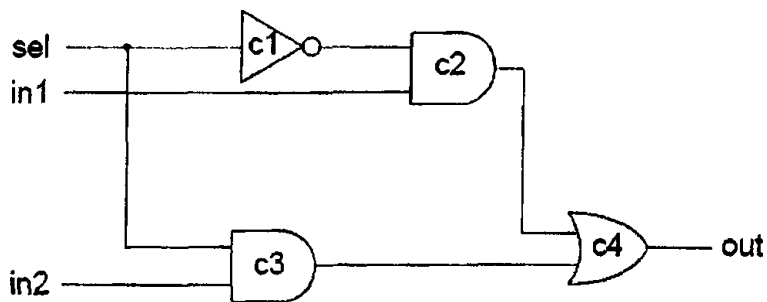
Πριν κληθεί η προσομοίωση, τα αντικείμενα του συστήματος πρέπει να δηλωθούν. Στην προσομοίωση σε επίπεδο πυλών, η καταγραφή των κυττάρων, που έχουν εξαρτήσεις μονοπατιών και καταστάσεων, και όλων των εσωτερικών γραμμών του συστήματος πραγματοποιείται με την εντολή \$read_lib_saif. Η σύνταξη της εντολής είναι \$read_lib_saif("filename"). Στη συνέχεια, με την εντολή \$set_toggle_region ορίζεται η περιοχή των αντικειμένων, που θα ελέγχονται από την προσομοίωση. Με



άλλα λόγια, η ελεγχόμενη από την προσομοίωση περιοχή είναι η τομή των καταγεγραμμένων αντικειμένων και της περιοχής εναλλαγών που ορίζεται μέσω της προαναφερθείσας εντολής. Με την εντολή \$toggle_start, μέσα στο αρχείο testbench, αρχίζει ο έλεγχος των εναλλαγών στα καταγεγραμμένα αντικείμενα και με την εντολή \$toggle_stop τερματίζεται η όλη διαδικασία. Στη συνέχεια, ενσωματώνοντας στο αρχείο testbench την εντολή \$toggle_report, καταγράφεται όλη σχετική πληροφορία των εναλλαγών της ελεγχόμενης περιοχής σε ένα αρχείο εξόδου. Το αρχείο εξόδου μπορεί να είναι της μορφής SAIF ή της μορφής dp.

Στο Σχήμα Γ.4 παρατίθεται το σχηματικό ενός απλού πολυπλέκτη και στη συνέχεια ακολουθεί η περιγραφή του σε Verilog.

Schematic of Multiplexer Circuit: MUX21



Σχήμα Γ.4 Πολυπλέκτης

Η περιγραφή του πολυπλέκτη στη γλώσσα Verilog είναι η ακόλουθη:

```

/* 'timescale 10ps/ 1ps */
module MUX21(out, d1, d2, sel);
input d1, d2, sel;
output out;
    IV c1(.Z(sel_), .A(sel));
    AN2 c2(.Z(d1m), .A(d1), .B(sel_));
    AN2 c3(.Z(d2m), .A(d2), .B(sel));
    OR2 c4(.Z(out), .A(d1m), .B(d2m));
endmodule

```



Αφού διαβαστεί η βιβλιοθήκη που περιέχει πληροφορίες σχετικά με την τεχνολογία, με την εντολή `lib2saif` δημιουργείται το αρχείο SAIF forward-annotation που φαίνεται στο Σχήμα Γ.5.

State- and Path-Dependent SAIF File (cell.saif)

```
(SAIFILE
(SAIFVERSION "2.0")
(DIRECTION "forward")
(SCOPE      "lib")
(DESIGN )
(   DATE "Mon Nov 24 14:04:58 1997")
(   VENDOR "Synopsys, Inc")
(   PROGRAM_NAME "lib2saif")
(   VERSION "1.0")
(DIVIDER / )
(LIBRARY "foo"
  (MODULE "AN2"
    (PORT
      (Z (IOPATH A IOPATH B ))
    )
  )
)
)
```

Σχήμα Γ.5 Αρχείο SAIF forward-annotation

Το αντίστοιχο Verilog testbench είναι το εξής:

```
'timescale 1ns/ 10ps
module top;
  reg in1, in2, sel;
  parameter hazrate = 0.99;
  parameter haztime = 0.23;

  MUX21 m1(out, in1, in2, sel);

  initial
  begin
    // start monitoring
    $monitor($time,, "in1=%b in2=%b sel=%b out=%b", in1,in2,sel,out);
    //read fwd SAIF file of state/path dependent info
    $read_lib_saif(cell.saif);
    //define the monitoring scope
    $set_toggle_region(m1);
    $toggle_start;
    //test first data line passing 0
```



```

sel = 0;
in1 = 0;
in2 = 0;
//test first data line passing 1
#10 in1 = 1;
#10 sel = 1;
$toggle_stop;
$toggle_report ("my_1st", 1.09e-9, "top.m1", hazrate, haztime);
//exit simulation
$finish(2);
end
endmodule

```

Κατόπιν, πραγματοποιείται η προσομοίωση του σχεδιαζόμενου συστήματος καλώντας είτε τον προσομοιωτή VCS, είτε τον Verilog-XL. Έτσι, λόγω της εντολής \$toggle_report δημιουργείται το αρχείο SAIF back-annotation, με βάση το οποίο μέσω της εντολής read_saif ενημερώνεται το σχεδιαζόμενο σύστημα.

Η ανάλυση βασισμένη σε γεγονότα του εργαλείου PrimePower παράγει ακριβείς, χρονικές πληροφορίες και προϋποθέτει τα ακόλουθα βήματα:

- Το εργαλείο PrimePower υπολογίζει το φορτίο για κάθε κύτταρο του σχεδιαζόμενου συστήματος.
- Το εργαλείο PrimePower επεξεργάζεται τα γεγονότα στο αρχείο που προκύπτει από την προσομοίωση και καθορίζει την ολική κατανάλωση ενέργειας για κάθε γεγονός.
- Το εργαλείο PrimePower αθροίζει την ενέργεια για κάθε κυττάρου για να καθορίσει την μέση κατανάλωση ενέργειας, καθώς και την κορυφαία κατανάλωση ενέργειας (peak power).

Τα αρχεία που περιέχουν τις δραστηριότητες του συστήματος είναι είτε της μορφής VCD είτε VPD. Γενικά, το PrimePower υποστηρίζει τους ακόλουθους τύπους αρχείων:



File type	Extension	Utility
VCD	default file type	none
VPD	.vpd	none
compressed VCD	.Z	uncompress
gzipped VCD	.gz	gunzip
FSDB	.fsdb	\$\$SYNOPSIS/bin/fsdb2vcd

Σχήμα Γ.6 Τύποι αρχείων

Τα αρχεία αυτά, ανεξαρτήτου τύπου, καθορίζονται με την ακόλουθη εντολή:

- `read_vcd [-path prefix] -strip_path [path prefix] [-pipe command] [-ignore_scope_type] [-zero_delay] filename`

Για τη δημιουργία των αρχείων VCD (Value Change Dump) πρέπει να εκτελεστεί η προσομοίωση του αρχείου testbench, στο οποίο είναι αναγκαίο να εμπεριέχονται οι σχετικές εντολές PLI. Το αρχείο VCD γενικά εμπεριέχει πληροφορίες σχετικά με τον χρόνο, την εμβέλεια, τους ορισμούς των σημάτων και τις αλλαγές των σημάτων. Οι εντολές PLI χρησιμοποιούνται για την επιλογή των οντοτήτων του συστήματος ή των σημάτων (\$dumpvars) που πρόκειται να ληφθούν υπόψη, την ονομασία του αρχείου VCD (\$dumpfile), την έναρξη και τη λήξη της διαδικασίας ελέγχου (\$dumpon, \$dumpoff) και την παραγωγή σημείων ελέγχου (\$dumpall). Αφού γίνει διαθέσιμο το αρχείο VCD, τότε το εργαλείο PrimePower με την εντολή `calculate_power` και τις κατάλληλες παραμέτρους μπορεί να προβεί στον υπολογισμό της ενέργειας που καταναλώνεται από κάθε οντότητα του συστήματος, καθώς και στην αναφορά της κορυφαίας ενέργειας. Γενικά, η εντολή `calculate_power` παρέχει τη δυνατότητα καθορισμού των χρονικών διαστημάτων, ποιες οντότητες θα είναι υπό έλεγχο και τι πληροφορίες μπορούν να αναφερθούν. Ενδεικτικά, αναφέρονται κάποιες παράμετροι της εντολής `calculate_power`:

- `-instance{list}`: αναφέρονται συγκεκριμένες οντότητες.
- `-time{list}`: ορίζονται τα χρονικά παράθυρα προσομοίωσης για τον υπολογισμό της ενέργειας.
- `-mode{expression}`: η ενέργεια υπολογίζεται όταν ισχύει η έκφραση Boolean.



- `-statistics`: το PrimePower θα αναφέρει στην έξοδο διάφορα στατιστικά δεδομένα, όπως το συνολικό πλήθος καταστάσεων, το πλήθος των καταστάσεων X, το πλήθος των καταστάσεων glitch, το πλήθος ενδεχόμενων παραβιάσεων κτλ.
- `-waveform`: το PrimePower αποθηκεύει την ενέργεια που καταναλώθηκε για κάθε γεγονός σε ένα δοθέν διάστημα (συνήθως του 1ns) σε ένα αρχείο με κυματομορφές, το οποίο μπορεί να ανοιχτεί με το εργαλείο turboWave.

Αν κανείς επιθυμεί να τροποποιήσει τις ιδιότητες σχετικά με τις κυματομορφές, αρκεί να χρησιμοποιήσει πριν την εντολή `calculate_power` την εντολή `set_waveform_options`, της οποίας η γενική σύνταξη είναι:

- `set_waveform_options [-file filename] [-leaf] [-hier level] [-format fsdb | out| rpt] [-interval sampling_period] [-effort high | low]`

Προκειμένου να αναφερθεί και η κορυφαία κατανάλωση ενέργειας (peak power), στην εντολή `calculate_power` θα πρέπει να προστεθεί η παράμετρος `-waveform`. Έτσι, η ενέργεια που καταναλώνει κάθε γεγονός σε ένα δοθέν χρονικό διάστημα (1ns) αποθηκεύεται στο αρχείο με τις κυματομορφές. Η μέγιστη τιμή και η χρονική στιγμή στην οποία έλαβε χώρα το γεγονός με αυτή τη τιμή καταγράφονται στο αρχείο που παράγεται με την εντολή `report_command`. Αν ωστόσο κάποιος ενδιαφέρεται μόνο για την κορυφαία κατανάλωση ενέργειας, μπορεί να μειωθεί η ποσότητα πληροφορίας προσθέτοντας στην επιλογή της κυματομορφής την παράμετρο `-hier 1`. Με αυτό τον τρόπο, μόνο οι πληροφορίες του υψηλότερου επιπέδου του συστήματος αποθηκεύονται, αλλά αναφέρεται η κορυφαία κατανάλωση ενέργειας όλης της ιεραρχίας του συστήματος στην αναφορά. Μια εναλλακτική λύση είναι η προσθήκη της παραμέτρου `-format rpt` στην εντολή `set_waveform_options` και έτσι δεν παράγονται καθόλου πληροφορίες για την κυματομορφή, παρά μόνο στην αναφορά εμπεριέχεται η πληροφορία σχετικά με την κορυφαία κατανάλωση ενέργειας.

Η βασική εντολή για την δημιουργία των αναφορών, μετά τον υπολογισμό της καταναλισκόμενης ενέργειας με την εντολή `calculate_power`, είναι η `report_power`. Η γενική της σύνταξη είναι η ακόλουθη:

- `report_power [-file file_prefix] [-instance list] [-threshold value][-leaf][-hier level_number][-cell][-net][-input_net][-sortby sort_type][-nodisplay]`



Στην προκύπτουσα αναφορά, όπως θα παρατηρήσει κανείς, εμπεριέχονται οχτώ τύποι ενεργειών:

- Η συνολική ενέργεια: η συνολική καταναλωμένη ενέργεια εξαιτίας των γεγονότων και η ενέργεια διαρροής.
- Η ενέργεια βασισμένη σε γεγονότα: η συνολική καταναλωμένη ενέργεια λόγω της εσωτερικής ενέργειας, του ρεύματος, των " glitches " και των μεταβάσεων X.
- Η ενέργεια διαρροής
- Η ενέργεια εναλλαγής: η ενέργεια που καταναλώνεται λόγω φόρτισης και αποφόρτισης των στοιχείων του κυκλώματος.
- Η εσωτερική ενέργεια: η ενέργεια που καταναλώνεται μέσα σε ένα κύτταρο.
- Η ενέργεια λόγω των μεταβάσεων X.
- Η ενέργεια λόγω ανιχνεύσιμων glitches στις γραμμές του κυκλώματος.
- Οι λογικές καταστάσεις για κάθε οντότητα.

Οι παράμετροι της εντολής `report_power` παρέχουν τη δυνατότητα για επιμέρους αναφορές, όπως αναφορές μόνο για τα κύτταρα (`-cell`) ή για τα κύτταρα που ανήκουν στο χαμηλότερο επίπεδο της ιεραρχίας (`-leaf`) ή για τις γραμμές (`-net`) ή για τις κυματομορφές.

Τέλος, παρουσιάζεται ενδεικτικά η διαδικασία που ακολουθείται από την παραγωγή των αναγκαίων αρχείων, που περιέχουν τη δραστηριότητα εναλλαγών, μέχρι την αναφορά με την καταναλισκόμενη ενέργεια του σχεδιαζόμενου συστήματος.

- `dc_shell`: Έναρξη του Design Compiler
- `set search_path "/tools/libraries ."` : Θέτουμε τη μεταβλητή `search_path` να περιέχει το path του technology library.
- `set link_library "umcl18u250t3_tc_180V_25C.db"` : Θέτουμε τη μεταβλητή `search_path` να περιέχει το technology library.
- `read_verilog circuit.v` : Ανάγνωση του αρχείου περιγραφής.
- `write_sdf -version 1.0 circuit.sdf` : Δημιουργία του αρχείου sdf.



- `lib2saif -output lib.saif umcl18u250t3_tc_180V_25C.db` : Δημιουργία του αρχείου SAIF forward-annotation με όνομα `cell.saif`.
- `exit`
- `vcs -P /tools/synopsys/syn_Z-2007.03-SP3/auxx/syn/power/vpower/vcs/vcs.sample/vpower.tab s9234_RF.v tb.v -v Model/*.v /tools/synopsys/syn_Z-2007.03-SP3/auxx/syn/power/vpower/lib-linux/libvpower.a` : Μεταγλώττιση
- `./simv` : Εκτέλεση και προσομοίωση. (Δημιουργία του αρχείου SAIF back-annotation με όνομα `my_1st.saif` και το αρχείου τύπου `vcd`.)
- `pp_shell` : Εναρξη του Prime Power.
- `set search_path "/tools/libraries . "` : Θέτουμε τη μεταβλητή `search_path` να περιέχει το path του `technology library`.
- `set link_library "umcl18u250t3_tc_180V_25C.db"` : Θέτουμε τη μεταβλητή `search_path` να περιέχει το `technology library`.
- `read_verilog circuit.v` : Ανάγνωση του αρχείου περιγραφής.
- `current_design TOP` : Θέτουμε το αντικείμενο TOP ως τρέχον.
- `Link` : Υλοποιείται η σύνδεση.
- `create_clock -period 50 -waveform {0 25} -name clk` : Δημιουργία του εικονικού ρολογιού `clk`.
- `set_input_transition 0.1 [all_inputs]` : Θέτουμε τον χρόνο της εναλλαγής.
- `read_sdf circuit.sdf` : Ανάγνωση του αρχείου SDF.
- `read_vcd -input -strip_path testmodule/mut TESTNAME.vcd` : Ανάγνωση του αρχείου VCD.
- `set_waveform_options -format rpt -effort high`: Ορίζονται οι ιδιότητες του αρχείου κυματομορφής.
- `calculate_power -statistics -waveform` : Υπολογισμός του μέσου όρου της καταναλισκόμενης ενέργειας και της κορυφαίας ενέργειας.
- `report_power > power_report.log` : Αναφορά των αποτελεσμάτων στο αρχείο εξόδου `power_report.log`.
- `exit`



ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ

Η Σπυριδούλα Μπαλατσούκα έλαβε το πτυχίο Πληροφορικής από το Τμήμα Πληροφορικής του Πανεπιστημίου Ιωαννίνων, το 2006. Από το 2007 είναι μεταπτυχιακή φοιτήτρια του Τμήματος Πληροφορικής του Πανεπιστημίου Ιωαννίνων.

